

明 細 書

符号化装置および復号化装置

技 術 分 野

本発明は、無線伝送路のような誤り率の高い媒体を介して情報を伝送／蓄積するシステムに係り、高能率圧縮符号化によって得られた圧縮符号列を誤り訂正／検出符号化して伝送／蓄積するのに適した符号化／復号化装置に関する。

背 景 技 術

例えば、無線ＴＶ電話、携帯情報端末、デジタルＴＶ放送システムのように、画像や音声の情報をできるだけ少ない情報量となるように高能率に圧縮符号化して無線伝送路等を介して伝送するシステムにおいては、伝送路の誤り率が高いため、得られた符号列をいかに高品質に伝送するかが重要な課題となる。

このように誤り率の高い媒体を介して符号列の伝送／蓄積を行なう場合に、誤り率を低減する手段として、ＢＣＨ（Bose-Chaudhuri-Hocquenghem）符号、ＲＳ（Recommended Standard）符号、畳み込み等の誤り訂正符号が多く用いられる。また、受信側で誤り検出を可能にする手段としては、チェックサムやＣＲＣ（cyclic redundancy check）等の誤り検出符号が用いられる。これらの誤り訂正／誤り検出は、伝送／蓄積する情報に余分なビット（冗長性）を一定の規則に従って付加し、復号の際に伝送／蓄積された符号列がこの規則に従っているかどうかを調べ、その結果によって誤りの訂正や検出を行なうものである。

しかし、このように高能率圧縮符号化により得られた符号列を誤り訂正／検出符号に符号化して伝送／蓄積する方法は、伝送路／媒体での符号語誤りによって生じる同期外れを回復する同期回復手法との組み合わせが難しくなることが欠点である。同期回復手法としては、一意に復号可能な同期符号と呼ばれる符号を挿

入し、同期外れが生じたときは同期符号を検出した時点から復号を再開する方法が多く用いられている。

同期符号を一意に復号可能な符号語とするためには、他の符号語の組み合わせで同期符号と同一のビットパターンが生じないように符号語を構成する必要がある。しかし、一般に誤り訂正／検出符号化では、ある特定のビットパターンの出現を避けるように符号語を構成することは困難であり、同期符号と同一ビットパターンが出現した場合には、同期符号の誤検出により擬似同期を引き起こしてしまうことになる。

この問題を避けるために、誤り訂正／検出符号化を行なった後に同期符号と同一のビットパターンが符号列中に存在するか否かを判定し、同一ビットパターンが存在する場合には、そのパターン中にある規則に従ってダミービットを挿入し、復号化装置において同一の規則でダミービットを削除することにより擬似同期を防ぐ方法が用いられている。しかし、誤りが生じやすい媒体を介して符号列を伝送／蓄積する場合には、この挿入ビットにも誤りが生じる可能性があり、新たな同期外れや擬似同期を引き起こすという問題点がある。

また、符号列を誤り訂正／検出符号化し同期符号を挿入する場合、従来では同期符号と同期符号とで挟まれた同期区間の最後の部分で、誤り訂正／検出符号化すべき情報ビットの剰余を補うために多くの挿入ビットを符号列に付加する必要があるため、符号化効率が低下するという問題点もある。

一方、誤り訂正／検出能力を高めるためには、伝送／蓄積する情報の冗長度を高めればよいが、冗長度を高くすると同じ情報を送るのに必要なビット数は増える。このため、誤り訂正／検出能力を高めると、より伝送レートの高い伝送路が必要となったり、蓄積しなければならない情報のビット数が増加することになる。また、伝送レートや蓄積容量が同一ならば、冗長度が高いほど少ない情報しか伝送／蓄積できないことになる。画像や音声を高能率に圧縮符号化して伝送／蓄積

する場合、誤り耐性を向上させるために冗長性を付加することは、伝送／蓄積レートが同一ならばより少ない情報量に圧縮符号化を行なわなければならないことになり、画質や音質の低下を招くことになる。

そこで、より少ない冗長度で高い誤り耐性を持たせる方法として、階層符号化と呼ばれる方法がある。これは高能率圧縮符号化された情報を誤りが画質や音質に与える大きさに応じて分類し、誤りの影響が大きい情報には冗長度が高くともより誤り訂正／検出能力の高い誤り訂正／検出符号を用い、誤りの影響があまり大きくない情報には誤り訂正／検出能力はそれほど高くないが冗長度が少ない誤り訂正／検出符号を用いることにより、全体に同一の誤り訂正／検出符号を用いる場合より同一の平均冗長度でより誤り耐性を高める方法である。

例えば、動画像の高能率圧縮符号化に多く用いられている動き補償予測と直交交換を組み合わせた符号化方式、すなわち、入力される動画像信号に対して動き補償予測を行ない、その予測残差をDCT（離散コサイン変換 — Discrete Cosine Transform — ）などにより直交変換する方式においては、誤りが生じると大きな画質劣化を生じる動きベクトル情報や、予測残差信号の直交変換係数のうちの低次係数には、誤り訂正／検出能力の強い誤り訂正／検出符号を用い、誤りの影響が小さい予測残差信号の直交変換係数のうちの高次係数には、誤り訂正／検出能力の弱い誤り訂正／検出符号を用いる。

このような階層符号化を実現するためには、出力される符号列の途中で、誤り訂正／検出能力の異なる誤り訂正／検出符号を切り替える必要がある。誤り訂正／検出能力の異なる誤り訂正／検出符号を切り替える方法としては、誤り訂正／検出符号の種類を示すヘッダ情報を符号列に付加する方法がある。図11は、ヘッダ情報を付加して誤り訂正／検出符号を切り替えた符号列の例を示したものであり、この例では2種類の誤り訂正／検出符号FEC1およびFEC2を切り替えている。ヘッダ1101～1104には、誤り訂正／検出符号の種類や符号語

の数を示すヘッダ情報が入っている。符号化装置では、これらのヘッダ情報に続けて誤り訂正／検出符号化された符号語を並べ、復号化装置ではヘッダ情報を復号し、これに従って誤り訂正／検出符号の復号を行なう。

しかし、このようなヘッダ情報を付加することによって誤り訂正／検出符号を切り替える方法では、ヘッダ情報の付加によって伝送／蓄積しなければならない符号列のビット数が増加するという問題がある。画像や音声を高能率圧縮符号化して伝送／蓄積する場合、ヘッダ情報にビット数が割かれることは、画像や音声の高能率圧縮符号化に用いられるビット数が減少することになり、結果的に画質や音質の劣化を招くことになる。

上述したように、動画像信号などを高能率圧縮符号化した符号列に対して誤り訂正／検出符号化を行なうと、任意のビットパターンが発生するため、誤り訂正／検出符号化と、一意に復号可能な同期符号を用いる同期回復手法とを組み合わせた場合には、同期符号の誤検出による擬似同期という問題点があり、ダミービットの挿入により擬似同期を避ける操作を行なった場合にも挿入ビットの誤りによって新たな同期外れや擬似同期を生じてしまうという問題点があった。

また、符号列を誤り訂正／検出符号化し同期符号を挿入する場合、従来では同期区間の最後の部分で、誤り訂正／検出符号化すべき情報ビットの剰余を補うために多くの挿入ビットを用いる必要があるため、符号化効率が低下するという問題点があった。

さらに、ヘッダ情報を付加することにより誤り訂正／検出能力の異なる誤り訂正／検出符号を切り替える符号化／復号化装置では、ヘッダ情報の付加により伝送／蓄積しなければならないビット数が増加するため、画像や音声を高能率圧縮符号化して伝送／蓄積する場合には、画像や音声の情報に割り当てられる情報量が減少して画質や音質の低下を招くという問題点があった。

発 明 の 開 示

本発明の主たる目的は、同期符号の誤検出による擬似同期や同期外れの問題を解決できる符号化装置および復号化装置を提供することにある。

本発明のより具体的な目的は、誤り訂正／検出符号化と同期符号を用いる同期回復手法を組み合わせた場合において、同期符号の誤検出による擬似同期や同期外れの問題を解決できる符号化装置および復号化装置を提供することにある。

本発明の他の目的は、誤り訂正／検出符号化と同期符号を用いる同期回復手法を組み合わせた場合において、同期区間の最後の部分で用いる挿入ビットを少なくして符号化効率を高めることができる符号化装置および復号化装置を提供することにある。

本発明のさらに別の目的は、画像や音声などの情報を高能率圧縮符号化した符号列を複数種類の誤り訂正／検出符号に切り替えて符号化して伝送／蓄積する場合において、誤り訂正／検出符号の種類を示すヘッダ情報の付加を必要として伝送／蓄積される符号列のビット数を減少させ情報の品質を向上させることができる符号化装置および復号化装置を提供することにある。

(1) 本発明に係る第1の符号化装置は、入力される符号列を情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化する符号化手段と、出力符号列中の予め定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、前記情報ビットを該符号列中の任意の位置に配置し、前記検査ビットを該符号列中の前記同期符号挿入位置以外の位置に配置して出力符号列を組み立てる符号列組立手段とを有することを特徴とする。

本発明に係る第1の復号化装置は、情報ビットと検査ビットからなる誤り訂正／検出符号に符号化された符号列から予め定められた複数の同期符号挿入位置で同期符号の検出を行なう同期符号検出手段と、前記符号列を分解して誤り訂正／検出符号の情報ビットおよび前記同期符号挿入位置以外の位置に配置されて該誤

り訂正／検出符号の検査ビットを抽出する符号列分解手段と、前記符号列分解手段により抽出された情報ビットおよび検査ビットを受けて前記誤り訂正／検出符号を復号化する復号化手段とを有することを特徴とする。

このように第1の符号化／復号化装置では、同期符号は出力符号列の予め定められた同期符号挿入位置にのみ存在し、誤り訂正／検出符号の検査ビットは同期符号挿入位置以外に存在するため、検査ビットに同期符号と同一のビットパターンが含まれていても、これを同期符号と誤検出する可能性はない。従って、特定のビットパターンの発生を防ぐ特殊な誤り訂正／検出符号を用いたり、誤り訂正／検出符号化後に同期パターンを防ぐためのビット挿入等を行なう必要がなくなるため、用いる誤り訂正／検出符号の自由度が増すだけでなく、挿入ビットに誤りが混入することによる新たな同期検出誤りを生じる可能性が無いために、誤りに対する耐性が向上する。

(2) 本発明に係る第2の符号化装置は、第1の符号化装置に、出力符号列中の予め定められた複数の同期符号挿入位置にある同期符号以外の入力符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段を追加し、この符号列変換手段により変換が行なわれた後の符号列を符号化手段に入力して情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化することを特徴とする。

本発明に係る第2の復号化装置は、第1の復号化装置において、復号化手段により復号化された符号列中に同期符号とのハミング距離が予め定められた値以上となるように変換された同期符号挿入位置に存在する該同期符号以外の符号列を元の符号列に変換する符号列変換手段を追加したものである。

このように第2の符号化／復号化装置では、符号化装置において同期符号挿入位置にあるビット列に対し同期符号とのハミング距離が予め定められた値以上となるように変換処理を行ない、復号化装置ではこれと逆の変換を行なうことによ

り、ビット列中に同期符号と同一のビットパターンが含まれなくなるため、同期符号の誤検出がなくなる。また、このように同期符号とそれ以外の符号列とのハミング距離が大きくなるように変換を行えば、符号列に誤りが混入しても同期符号とそれ以外の符号列を区別することが可能になるため、誤りによる同期符号の誤検出の確率が減少する。

この変換／逆変換処理は同期符号挿入位置でのみ行なわれるため、符号列全体に変換／逆変換処理を行なう従来法に比べてオーバーヘッドが少ない。また、符号化装置への入力の符号列では、同期符号と同一のビットパターンを生じないように変換処理を行ったり、特別な符号語を用いる必要はない。特に、異なる符号語テーブルを切り替えて用いる可変長符号化装置を本符号化装置の入力としている場合、可変長符号化装置で同期符号と同一のパターンが生じないように符号語テーブルを作成すると、符号化効率の低下を招いてしまうという問題点があったが、第2の符号化／復号化装置を用いればこのような問題は生じない。

(3) 本発明に係る第3の符号化装置は、入力される符号列を誤り訂正／検出符号に符号化する符号化手段と、前記符号列中に同期符号を挿入する同期符号挿入手段と、前記符号列の前記同期符号の直前の誤り訂正／検出符号に符号化されるべき情報のビット数を判定する判定手段とを有し、前記符号化手段において前記同期符号の直前の誤り訂正／検出符号を前記判定手段により判定されたビット数に基づいて適応的に縮退させた縮退符号とすることを特徴とする。

本発明に係る第3の復号化装置は、誤り訂正／検出符号に符号化されかつ同期符号が挿入された符号列を復号化する復号化手段と、前記符号列中の同期符号を検出する同期符号検出手段と、前記符号列中の同期符号検出手段により検出された同期符号の直前の誤り訂正／検出符号に符号化された情報のビット数を判定する判定手段とを有し、前記復号化手段において前記判定手段の判定結果に基づいて前記同期符号の直前の誤り訂正／検出符号が縮退符号であるか否かを識別して

復号化を行なうことを特徴とする。

このように第3の符号化／復号化装置では、同期符号の直前の誤り訂正／検出符号として、1同期区間の最後に残された情報ビットを符号化するのに必要なビット数まで縮退された縮退符号を用いるため、同期区間の最後の部分で情報ビットの剰除を補うために多くの挿入ビットを用いる必要が無く、符号化効率が向上する。

(4) 本発明に係る第4の符号化装置は、入力される複数種類の情報を含む符号列を誤り訂正／検出符号に符号化する符号化手段と、前記誤り訂正／検出符号の種類を前記符号列中の情報の種類に応じて切り替える切り替え手段とを有することを特徴とする。

本発明に係る第4の復号化装置は、情報の種類に応じて異なる種類の誤り訂正／検出符号に符号化された符号列を復号化して元の情報を生成する復号化手段と、前記復号化手段で生成された情報の種類から前記誤り訂正／検出符号の種類を判定して前記復号化手段に通知する手段とを有することを特徴とする。

このように第4の符号化／復号化装置では、情報の種類に応じて誤り訂正／検出符号を切り替えて符号化／復号化を行なう場合、符号化装置においては入力する符号列の情報の種類に応じて誤り訂正／検出符号の切り替えを行なうとともに、復号化装置では復号された情報を基に誤り訂正／検出符号の種類を判定して符号化装置と同一の切り替えを行なうことによって、従来のように誤り訂正／検出符号の種類を示すヘッダ情報を用いる必要はなく、ヘッダ情報によるオーバーヘッドをなくすることができる。

(5) 本発明に係る第5の符号化装置は、出力符号中の予め定められた複数の同期符号挿入位置とその前後の予め定められたビット数の区間にある同期符号以外を入力符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段と、前記符号列変換手段により変換が行なわれた後の

符号列を情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化する符号化手段と、前記出力符号列中の予め定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に前記情報ビットを該出力符号列中の任意の位置に配置し、前記検査ビットを該符号列中の前記同期符号挿入位置以外の位置に配置して該出力符号列を組み立てる符号列組立手段とを有することを特徴とする。

本発明に係る第5の復号化装置は、情報ビットと検査ビットからなる誤り訂正／検出符号に符号化されかつ同期符号が挿入された符号列から、予め定められた同期符号挿入位置とその前後の予め定められたビット数の区間で同期符号の検出を行なう同期符号検出手段と、前記符号列を分解して誤り訂正／検出符号の情報ビットおよび前記同期符号挿入位置以外の位置に配置された該誤り訂正／検出符号の検査ビットを抽出する符号列分解手段と、前記符号列分解手段により抽出された情報ビットおよび検査ビットを受けて前記誤り訂正／検出符号を復号化する復号化手段と、前記復号化手段により復号化された符号列中の同期符号とのハミング距離が予め定められた値以上となるように変換された前記同期符号挿入位置およびその前後の予め定められたビット数区間に存在する該同期符号以外の符号列を元の符号列に変換する符号列変換手段とを有することを特徴とする。

このように第5の符号化／復号化装置では、入力符号列について同期符号挿入区間のみでなく、その前後の予め定められたビット数の区間で同期符号とのハミング距離が予め定められた値以上となるような変換を行ない、復号化装置では入力された符号列に対してこれと逆の変換を行なうことにより、この区間で同期符号と同一のビットパターンが含まれなくなるため、一部のビットが消失するビット消失や、余分のビットが付加されるビット付加が起こる伝送路／蓄積媒体を介して符号化されたビット列の伝送／蓄積を行なっても、消失／付加のビット数が予め定められたビット数以下ならば、同期符号とそれ以外の符号列を区別することが可能であり、復号化装置で正しく同期検出を行なうことができる。

(6) 本発明に係る第6の符号化装置は、入力信号を圧縮符号化して得られた複数種類の圧縮符号を多重化して多重化符号列を生成する多重化手段と、前記多重化符号列を入力として出力符号列を組み立てる符号列組立手段とを有し、前記符号列組立手段は、前記出力符号列中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入することを特徴とする。

この場合、前記符号列組立手段は、前記出力符号列中の予め周期的に定められた複数の同期符号挿入位置にいずれに同期符号を挿入するために、該出力符号列中にスタッフィングビットの挿入を行なうか、または前記出力符号列中に予め周期的に複数の同期符号挿入位置を定めると共に、前記多重化符号列の境界を指し示す情報（ポインタ情報）を挿入し、前記複数の同期符号挿入位置のいずれかに同期符号を挿入する。このようにすることにより、同期符号を周期的に定められた複数の同期符号挿入位置のいずれかにのみ挿入することができる。

ここで、スタッフィングビットは出力符号列の逆方向から一意に復号可能な符号であることが望ましい。このようにすると、復号化装置においてスタッフィングビットの直前の符号列の復号終了位置とスタッフィングビットの開始位置とを比較することにより、入力符号列中の誤りを容易に検出することができる。

さらに、このスタッフィングビットは同期符号およびその一部との間のハミング距離が予め定められた値以上であることが望ましい。このようにすると、擬似同期の発生確率が低くなるという利点がある。

本発明に係る第6の復号化装置は、入力符号列中の同期符号を検出する同期符号検出手段と、前記入力符号列から前記同期符号検出手段により検出された同期符号の位置を基準として逆多重化を行なって圧縮符号を生成する逆多重化手段と、前記圧縮符号を復号化して再生信号を出力する復号化手段とを有し、前記同期符号検出手段は、前記入力符号列中の予め周期的に定められた複数の同期符号挿入位置で前記同期符号を検出することを特徴とする。

このように第6の符号化／復号化装置では、符号化装置において複数種類の圧縮符号を多重化した多重化符号列中に予め周期的に定められた同期符号挿入位置にのみ同期符号を挿入するため、復号化装置においては、この同期符号挿入位置でのみ同期検出を行えば良く、符号列中の任意の位置に同期符号が挿入される従来方式に比べて同期符号検出の回数が削減され、またビット誤りによって復号化装置に入力されるビット列が同期符号と同一のビットパターンに変化してしまうことによる擬似同期の発生確率は、この同期符号検出回数の削減に伴って減少するため、本発明によると擬似同期の発生を少なくすることができ、同期符号検出に伴う演算処理量も削減される。

(7) 本発明に係る第7の符号化装置は、第6の符号化装置の構成に加えて、前記出力符号列中の前記同期符号挿入位置にある前記同期符号以外の符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段をさらに有することを特徴とする。

本発明に係る第7の復号化装置は、前記入力符号列の前記同期符号挿入位置において前記同期符号とのハミング距離が予め定められた値以上となるように変換された該同期符号以外の符号列を元の符号列に変換する符号列変換手段をさらに有することを特徴とする。

このように第7の符号化／復号化装置では、第6の符号化／復号化装置の構成に加えて、同期符号の誤りを考慮したビット挿入、すなわち同期符号挿入位置にあるビット列に対して同期符号とのハミング距離の変換を行なうことにより、ビット列中に同期符号と同一のビットパターンが含まれなくなるため、想定したビット数以下の誤りに対しては同期符号の誤検出が生じないことが保証され、同期符号誤検出の確率が減少する。

なお、本発明において同期符号とは符号列中に同期回復のために挿入される一意に復号可能な符号であり、例えば同期符号を挿入する符号列が、フレーム単位

で入力される画像信号を圧縮符号化した複数種類の圧縮符号を多重化した多重化符号列であれば、符号化フレームの区切りや、複数種類の圧縮符号の区切りその他の区切りを示す符号である。

以上説明したように、本発明によると次の効果が得られる。

(1) 第1の符号化／復号化装置によれば、同期符号を一定間隔おきの同期符号挿入位置にのみ挿入し、誤り訂正／検出符号の検査ビットについては位置ずらしを行なうことによって同期符号挿入位置以外の位置に存在するようにしているため、検査ビットに同期符号と同一のビットパターンが生じても、同期符号検出を行なう同期符号挿入位置には同期符号と同一のパターンが生じることはなく、誤まって同期検出が行なわれる可能性は原理的に全くない。

また、同期符号挿入位置にある符号列に対しては擬似同期符号が生じないようにビット挿入を行なえば、従来のように同期ビットと同一のビットパターンが生じないように符号語を構成するという困難も解消される。

(2) 第2の符号化／復号化装置によれば、符号化装置において第1の符号化装置の構成に加えて同期符号の誤りを考慮したビット挿入、すなわち同期符号挿入位置にあるビット列に対し同期符号とのハミング距離が予め定められた値以上になるような変換処理を行ない、復号化装置ではこれと逆の変換を行なうことにより、ビット列中に同期符号と同一のビットパターンが含まれなくなるため、想定したビット数以下の誤りに対しては同期符号の誤検出が生じないことが保証され、同期符号の誤検出の確率が減少する。また、このような変換を行なうと符号列に誤りが混入しても同期符号とそれ以外の符号列を区別することが可能になるため、誤りによる同期符号の誤検出の確率は減少する。

さらに、誤り訂正／検出符号化はビット挿入による符号語の変換後に行なわれるため、挿入ビットにも誤り保護が施されることになる。このため、誤り訂正／検出符号化後にビット挿入を行なう従来の方式に比較して、挿入ビットの誤りが

生じる確率が低くなる。しかも、ビット挿入は同期符号挿入位置のみ行なわれるため、従来の符号列全体でビット挿入を行なう場合より、ビット挿入による符号量増加は少なく、符号化効率が向上する。

(3) 第3の符号化／復号化装置によれば、同期符号の直前の誤り訂正／検出符号を縮退符号としているため、従来の符号化装置に比較して同期符号の直前部分で情報ビットの剰除を補うためのビット挿入数を少なくすることができ、符号化効率が向上する。

(4) 第4の符号化／復号化装置によれば、符号化装置において画像や音声等の高能率圧縮符号化装置における多重化の規則を利用して、入力する符号列の情報の種類に応じて誤り訂正／検出符号の切り替えを行ない、復号化装置では復号された情報を基に誤り訂正／検出符号の種類を判定して符号化装置と同一の切り替えを行なうことによって、誤り訂正／検出符号の種類を示すヘッダ情報の付加は不要となり、その分のビット数を画像や音声の高能率圧縮符号化に割り当てることができるため、結果的に画質や音質などの情報品質を向上させることができる。

(5) 第5の符号化／復号化装置によれば、符号化装置において入力符号列について同期符号挿入区間のみでなく、その前後の予め定められたビット数の区間で同期符号とのハミング距離が予め定められた値以上となるような変換を行ない、復号化装置では入力された符号列に対してこれと逆の変換を行なうことにより、この区間で同期符号と同一のビットパターンが含まれなくなるようにすることができる。このため、一部のビットが消失するビット消失や、余分のビットが付加されるビット付加が起こる伝送路／蓄積媒体を介して符号化されたビット列の伝送／蓄積を行なっても、消失／付加のビット数が予め定められたビット数以下ならば、同期符号とそれ以外の符号列を区別することが可能であり、復号化装置で正しく同期検出を行なうことが可能となる。

(6) 第6の符号化／復号化装置によれば、符号化装置において複数種類の圧縮

符号を多重化した多重化符号列中に予め周期的に定められた同期符号挿入位置にのみ同期符号を挿入するため、復号化装置においては、この同期符号挿入位置でのみ同期検出を行えば良く、符号列中の任意の位置に同期符号が挿入される従来方式に比べて同期符号検出の回数が削減される。

また、ビット誤りによって復号化装置に入力されるビット列が同期符号と同一のビットパターンに変化してしまうことによる擬似同期の発生の確率は、この同期符号検出回数の削減に伴って減少するため、本発明によると擬似同期の発生を少なくすることができ、同期符号検出に伴う演算処理量も削減される。

(7) 第7の符号化／復号化装置によれば、第6の符号化／復号化装置の構成に加えて、同期符号の誤りを考慮したビット挿入、すなわち同期符号挿入位置にあるビット列に対して同期符号とのハミング距離の変換を行なうことにより、ビット列中に同期符号と同一のビットパターンが含まれなくなるため、想定したビット数以下の誤りに対しては同期符号の誤検出が生じないことが保証され、同期符号誤検出の確率が減少する。

図面の簡単な説明

図1は、本発明の第1および第2実施形態に係る動画像符号化装置の構成を示すブロック図である。

図2は、図1の動画像符号化装置における多重化器での多重化の規則を示す図である。

図3は、図1の動画像符号化装置における出力符号化装置の構成を示すブロック図である。

図4は、図1の動画像符号化装置からの出力符号列の例を示す図である。

図5は、同期符号の例を示す図である。

図6は、図3の出力符号化装置における誤り訂正／検出切替符号化部の構成を示すブロック図である。

図7は、図3の出力符号化装置における符号列組立器の構成を示すブロック図である。

図8は、本発明の第1および第2実施形態に係る動画像復号化装置の構成を示すブロック図である。

図9は、図8の動画像復号化装置における入力復号化装置の構成を示すブロック図である。

図10は、図9の入力復号化装置における符号列分解器の構成を示すブロック図である。

図11は、従来の誤り訂正／検出符号切替符号化装置で得られる符号列の例を示す図である。

図12は、本発明の第2実施形態を説明するための伝送路のビット付加／消失によって誤りが生じた同期符号の例を示す図である。

図13は、第2実施形態における図3中のビット挿入器の動作を説明するための図である。

図14は、第2実施形態における図9中の同期検出器および挿入ビット除去器の動作を説明するための図である。

図15は、第1および第2実施形態においてフレーム長情報を用いて同期保護を行なうようにした符号列の例を示す図である。

図16は、第1および第2実施形態においてフレーム長情報を用いて同期保護を行なうようにした符号列の他の例を示す図である。

図17は、第1および第2実施形態においてフレーム長情報を用いて同期保護を行なうようにした符号列のさらに別の例を示す図である。

図18は、本発明の第3および第4実施形態に係る動画像符号化装置の構成を示すブロック図である。

図19は、本発明の第3実施形態に係る動画像符号化装置の出力符号化装置の

構成を示すブロック図である。

図20は、本発明の第3実施形態に係る動画像符号化装置からの出力符号列の例を示すブロック図である。

図21は、本発明の第3および第4実施形態に係る動画像復号化装置の構成を示すブロック図である。

図22は、本発明の第3実施形態に係る動画像復号化装置の入力復号化装置の構成を示すブロック図である。

図23は、本発明の第4実施形態に係る動画像符号化装置の出力符号化装置の構成を示すブロック図である。

図24は、本発明の第4実施形態に係る動画像復号化装置の入力復号化装置の構成を示すブロック図である。

図25は、本発明の第4実施形態に係る動画像符号化装置からの出力符号列の例を示すブロック図である。

図26は、動画像符号化装置における多重化器での多重化の規則を示す図である。

図27は、本発明の第5実施形態に係る動画像符号化装置からの出力符号列の例を示すブロック図である。

図28は、動画像符号化装置における多重化器での多重化の他の例を示す図である。

図29は、図28で示される多重化を行なった場合の各同期符号に対して処理を行なった出力符号列を示す図である。

図30は、本発明で用いるスタッフィングビットの例を説明するための符号表を示す図である。

図31は、図30のスタッフィングビットを用いた場合の復号化装置の処理を説明するための図である。

図32は、図30のスタッフィングビットの特徴を説明する図である。

図33は、同期符号挿入間隔を同期符号より短くした場合の出力符号列の例を示す図である。

図34は、異なる長さの同期符号を用いる例を示す図である。

図35は、本発明の第6実施形態に係る動画像符号化装置からの出力符号列を示す図である。

発明を実施するための最良の形態

以下、図面を参照して本発明の実施形態を説明する。

(第1実施形態)

図1は、動き補償適応予測と直交変換符号化の一種である離散コサイン変換符号化を用いた高能率圧縮符号化装置に、本発明による誤り訂正／検出符号切り替え機能を有する符号化装置を組み合わせた動画像符号化装置の第1実施形態を示すブロック図である。動き補償適応予測と離散コサイン変換符号化を組み合わせた符号化方式については、例えば文献1：安田浩編著、“マルチメディア符号化の国際標準”、丸善、(平成3年6月)等に詳しいので、動作の概略のみを説明する。また、本実施形態で用いられる誤り訂正／検出符号は、BCH符号のように情報ビットと検出ビットが分離されているものとする。

図1において、フレーム単位で入力される符号化対象の入力動画像信号131は、まずマクロブロック等の小領域単位で動き補償適応予測が行なわれる。すなわち、動き補償適応予測器101において、入力動画像信号131とフレームメモリ102中に蓄えられている既に符号化／局部復号化が行なわれた画像信号との間の動きベクトルが検出され、この動きベクトルに基づいて動き補償予測により予測信号132が作成される。この動き補償予測器101においては、動き補償予測符号化と、入力動画像信号131をそのまま符号化するフレーム内符号化(予測信号=0)のうち、符号化に用いて好適な方の予測モードが選択され、対

応する予測信号132が出力される。

予測信号132は減算器103に inputs され、入力動画像信号131から予測信号132が減算されることによって予測残差信号133が出力される。予測残差信号133は、離散コサイン変換器104において一定の大きさのブロック単位で離散コサイン変換(DCT)され、DCT係数が生成される。このDCT係数は量子化器105で量子化される。量子化器105からの量子化されたDCT係数データは2分岐され、一方は第1の可変長符号化器106で可変長符号化され、他方は逆量子化器107で逆量子化された後、逆離散コサイン変換器108で逆離散コサイン変換(逆DCT)される。逆離散コサイン変換器108からの出力は加算器109において予測信号132と加算され、局部復号信号が生成される。この局部復号信号は、フレームメモリ102に記憶される。

一方、動き補償適応予測器101において決定された予測モードおよび動きベクトルの情報は、第2の可変長符号化器110で可変長符号化される。第1、第2の可変長符号化器106、110から出力される可変長符号(圧縮符号)は、多重化器111において多重化され、多重化符号列201となる。

多重化器111からは、多重化符号列201と、これに対応する誤り/検出符号の種類を示すFEC種類識別信号202および同期符号の挿入を要求する同期符号挿入要求信号203が出力される。

これらの符号列202、FEC種類識別信号202および同期符号挿入要求信号203は、符号列202を誤り訂正/検出能力の異なる複数種類の誤り訂正/検出符号に切り替えて符号化する出力符号化装置200に inputs され、最終的な出力符号列205が生成される。本実施形態においては、出力符号化装置200が本発明に係る符号化装置に相当する。

図2は、多重化器111における多重化の流れを示す図である。多重化は符号化フレームを単位として行なわれる。まず、同期符号301が多重化される。同

期符号301が多重化されたときには、多重化器111から同期符号挿入要求信号203が出力され、多重化された符号語が同期符号であることが符号化装置200に知らされる。次に、多重化符号列201にその符号化フレームの各種符号化モードを示すピクチャヘッダ203が多重化される。次に、各領域における動き補償適応予測器MCにおける予測モードを示す予測モード情報303が多重化され、さらに動きベクトル情報304および予測残差信号のDCT係数（以下、これを残差DCT係数という）305が多重化される。ピクチャヘッダ302、予測モード情報303、動きベクトル情報304および残差DCT係数305を多重化するときには、それぞれに対応する誤り訂正／検出符号の種類を示すFEC種類識別信号202が出力される。

誤りが混入すると画質が大きく劣化してしまうピクチャヘッダ302、予測モード情報303および動きベクトル情報304には、訂正／検出能力の高い誤り訂正／検出符号を用いる。一方、残差DCT係数305は、誤りが混入してもそれを検出して残差を0とすることで大きな画質劣化を防ぐことができるため、誤り訂正能力はそれほど高くする必要はなく、単に誤り検出を行なうことができるようにするだけでもよい。

図3は、図1中の出力符号化装置200の構成を示すブロック図である。この出力符号化装置200は、ビット挿入器211、誤り訂正／検出切替符号化器212および符号列組立器213から構成される。また、図4は出力符号化装置200で生成される出力符号列205の一例を示す図である。図4において、PSCは同期符号、PHはピクチャヘッダ、MODEは予測モード情報、MVは動きベクトル情報、CHKは誤り訂正／検出符号の検査ビット、COEFは残差DCT係数、STUFFはスタッフィングビット（挿入ビット）である。この出力符号列205は、以下のような特徴を持っている。

(1) 同期符号PSCは、一定間隔（sync__periodビットおき）に配置された

矢印で示される同期符号挿入位置のいずれかにのみ挿入される。sync__periodの長さは、同期符号P S Cの長さで検査ビットCHKの最大長よりも大きくとる。検査ビットCHKは、同期符号挿入位置の直前に来るように位置ずらしが行なわれている。

(2) 1フレーム、つまり同期符号P S Cと次のP S Cとで挟まれた1同期期間の最後の部分の誤り訂正／検出符号は、最後に残った情報ビットだけを符号化する縮退符号とし、検査ビットCHK (図4の例ではCHK 6) の位置ずらしを行なうために必要なビット数だけスタッフィングビットSTUFFのビット挿入を行なう。

(3) 誤り訂正／検出符号の種類や数を示すF E C種類識別信号は、図4の出力符号列205中には存在しない。

この出力符号列205では、上記(1)のように検査ビットCHKの位置ずらしを行なっているため、矢印で示される同期符号挿入位置には検査ビットCHKが入ることが無く、従って検査ビットCHKによって擬似同期が発生する可能性は全く無い。また、上記(2)のようにフレームの最後の誤り訂正／検出符号化を行なう場合、従来の技術では多くの挿入ビット(ダミービット)を挿入することが必要であったが、本実施形態ではフレームの最後が縮退符号となっているため、挿入ビット数は少なくすむ。また、上記(3)のように誤り訂正／検出符号の種類や数を示すヘッダ情報を出力符号列205中に含ませていないため、これによる符号量の増加は無い。

以下、このような出力符号列205を生成する図3の出力符号化装置200の構成と動作について、多重化器111から出力される図2の多重化符号列201および図4の出力符号列205と対応させて詳細に説明する。

多重化器111で同期符号301が多重化されたとき、前述のように同期符号挿入要求信号203が出力される。同期符号301は、例えば図5のようにsync

__0__len ビットの“0”、1ビットの“1”および同期符号301の種類を示すsync_nb__len ビットの“×××××”からなる。出力符号化装置200は、多重化器111からの同期符号301および同期符号挿入要求信号203を受けると、符号列組立器213から出力符号列205として同期符号(PSC)を出力する。

ここで、同期符号301は図4に示したように出力符号列205中のsync__periodビットおきに配置された同期符号挿入位置のみにしか挿入することができないようになっているため、それまでに生成された出力符号列205の最後尾が同期符号挿入位置に無い場合には、後述のようにスタッフィングビットSTUFFを挿入して、同期符号301が同期符号挿入位置に来るようにする。

同期符号301が出力符号列205に出力されると、これ以降ピクチャヘッダ302、予測モード情報303、動きベクトル情報304、残差DCT係数305が以下のように符号化される。まず、多重化器111から出力された多重化符号列201に対しては、ビット挿入器211において擬似同期の発生を防ぐためのビット挿入が行なわれる。すなわち、出力符号列205中に同期符号301の符号語と同一のビットパターンがあると同期符号301を一意に復号できなくなることを防ぐために、必要に応じてビット挿入を行なう。例えば、同期符号301が図5のようにsync__0__len ビットの“0”が連続する符号語ならば、同期符号301以外の符号列中にsync__0__len ビット以上“0”が連続しないように“1”を挿入すれば、擬似同期の発生を防ぐことができる。

前述のように同期符号301は同期符号挿入位置にしか挿入されないために、この擬似同期の発生防止のためのビット挿入操作は同期符号挿入位置だけで行なえばよい。そこで、これまでに生成された出力符号列205の総ビット数を示す計数値221を符号列組立器213から出力して、ビット挿入器211で計数値221に基づいてビット挿入が必要であるか否かを判定する。計数値221、つ

まり既に生成された出力符号列205の総ビット数をtotal __len とすれば、

$$0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len}$$

となる区間で多重化符号列201中の“1”の数を計数し、この区間に“1”が全くなければ1ビットの“1”を挿入する。ここで、 $A \bmod B$ はAをBで除したときの剰余を表す。

また、誤りによる同期符号301の誤検出の確率を低くするには、以下のようにビット挿入を行えばよい。

同期符号301にnビット誤りが混入した場合にも、同期符号301を検出するためには、後述する動画像復号化装置の入力復号化装置で真の同期符号とハミング距離がn以下の符号語は同期符号であるものと判定する必要がある。しかし、同期符号301以外の符号列をそのままにしてこのような判定を行なうと、同期符号

301以外の符号列中にも同期符号とハミング距離がn以下のビットパターンが存在することがあり、これが同期符号挿入位置にあると誤って同期符号301と判定されてしまう。

そこで、ビット挿入器211で多重化符号列201に以下のようにビット挿入を行なうことによって、多重化符号列201中の同期符号挿入位置にある同期符号以外の符号列を同期符号301とのハミング距離が $2 * n + 1$ 以上離れた値となるように変換する。具体的には、

$$0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len} - (2 * N + 1)$$

となる区間で“1”の数(=n0とする)を計数し、n0が $2 * n + 1$ 以下であつたら $2 * n + 1 - n0$ ビットの“1”を多重化符号列201に挿入する。

こうしてビット挿入器211でビット挿入が行なわれた後の符号化列222は、誤り訂正／検出符号の種類を示すFEC種類識別信号202と共に、誤り訂正／検出符号切替符号化部212に入力される。

図6は、図3中に誤り訂正／検出符号切替符号化部212の構成を示すブロック図である。ラッチ回路603は、FEC種類識別信号202をラッチする回路であり、多重化器111から多重化符号列201への同期符号の出力が終了して同期符号挿入要求信号203が停止すると、FEC種類識別信号202をラッチし、ラッチした信号623を誤り訂正／検出符号化器604に供給する。

誤り訂正／検出符号化器604は、ラッチされた信号623に従ってビット挿入器211からの符号列222を誤り訂正／検出符号化し、情報ビット631および検査ビット632を生成して出力する。また、誤り訂正／検出符号化器604は1ブロックの誤り訂正／検出符号化が終了すると、ラッチ回路603に次のFEC種類識別信号202のラッチを指示するラッチ指示信号625を出力する。ラッチ回路603は、このラッチ指示信号625に従ってラッチを行ない、ラッチした信号623を誤り訂正／検出符号化器604に供給する。

出力符号化装置200では以上の動作が繰り返され、ビット挿入器211からのビット挿入後の符号列222は、誤り訂正／検出切替符号化器212で多重化器111から示されたFEC種類識別信号202に従って誤り訂正／検出符号の切り替えがなされながら、誤り訂正／検出符号化が行なわれる。FEC種類識別信号202は、1ブロックの誤り訂正／検出符号の符号化の終了時点でのみラッチ回路603によりラッチされるため、この切り替え点までは同一の誤り訂正／検出符号が適用される。例えば、ピクチャヘッダ302はFEC1、予測モード情報303はFEC2という誤り訂正／検出符号を用いる場合、ピクチャヘッダ302のビット数がFEC1の1ブロックの情報ビット数より短い場合は、後続の予測モード情報303の誤り訂正／検出符号としては、FEC1の情報ビット数に達するまではFEC1が用いられる。

図7は、図3中の符号列組立器213の構成を示すブロック図である。この符号列組立器213は、出力符号列205のビット数を計数するカウンタ701、

検査ビット632とそのビット数を一時的に記憶するバッファ702、出力符号列205を切り替える切替器703、および切替器703を制御する切替制御器704から構成される。

カウンタ701は、同期符号挿入要求信号203が入力すると同期符号長sync__len の値にリセットされ、同期符号の次のビットから順に次の同期符号が入力するまでカウントアップしていく。同期符号入力後、最初の検査ビット632が入力するまでは情報ビット631が出力されるように切替器703が動作する。検査ビット632が入力すると、バッファ702に記憶されると共に、そのビット数（検査ビット数）711がバッファ702から切替制御器704に出力される。

切替制御器704は、検査ビット数711とカウンタ701の計数値221を基に、前述のように検査ビット632が同期符号挿入位置に出力されないようにするための検査ビットずらしを行なうように切替器703を制御する。例えば計数値221をbit __count、検査ビット数711をcheck __len とすれば、

$$\text{bit_count} \bmod \text{sync_period} < \text{sync_period} - \text{check_len}$$

の時は情報ビット631を出力し、

$$\text{sync_period} - \text{check_len} \leq \text{total_bits} \bmod \text{sync_period} < \text{sync_period}$$

の時には、バッファ702に記憶された検査ビット713を出力する。以降、情報ビット631と検査ビット632を入力しながら、以上の処理を繰り返す。

出力符号化装置200は、前述のように1フレームの最後の部分では誤り訂正／検出符号として縮退符号を用いることと、検査ビットの位置ずらしを行なうためにビット挿入を行なうため、それ以外の部分の通常とは異なった動作を行なう。すなわち、多重化器111は1フレームの多重化符号列201の出力が終了すると、次のフレームの同期符号挿入要求信号203を出力する。これに対応して、

図6の誤り訂正／検出切替符号化部212内の誤り訂正／検出符号604は、誤り訂正／検出符号の情報ビット631の不足分は挿入ビット発生器705から出力される予め定められたビットパターンであるとみなして、縮退符号を用いた誤り訂正／検出符号化を行なう。このビットパターンは、全ビットを“1”としてもよいし、“0”としてもよいし、“0101…”のような特定のパターンの繰り返しでもよい。情報ビット631には、この補われた挿入ビットは出力しない。

図7の符号列組立器213では、情報ビット631を最後まで出力した後、切替器703をビット発生器705から入力に切り替え、バッファ702に記憶されている検査ビット713が次の同期符号の直前に配置されるように挿入ビットを挿入する。この挿入ビットの数stuffing_lenは、1フレームの最後の情報ビット631を出力したときのカウンタ701の計数値221をtotal_lenとし、最後に出力する検査ビット632のビット数をlast_check_lenとすれば、

$$\text{stuffing_len} = \text{sync_period} - \text{last_check_len} - (\text{total_len} \bmod \text{sync_period})$$

となる。なお、縮退符号を用いない場合は、最後の情報ビットlast_into_lenの通常の情報ビットinto_lenからの不足分($\text{into_len} - \text{last_into_len}$)ビットのビット挿入を行ない、さらに検査ビットの位置ずらしを行なうためのビット挿入を行なう必要がある。このため、縮退符号を用いる場合に比べ $\text{into_len} - \text{last_into_len} + (\text{into_len} - \text{last_into_len}) \bmod \text{sync_period}$ ビット多くの挿入ビットが必要となってしまう。

符号列組立器213は、こうして切替器703を介して情報ビット631および挿入ビットを出力符号列205に出力した後、最後に検査ビット713に切り替えて出力符号列205に出力する。

次に、本実施形態に係る動画像復号化装置について説明する。

図8は、図1の動画像符号化装置に対応した動画像復号化装置の構成を示すブ

ロック図である。図1の動画像復号化装置から出力された出力符号列205は伝送／蓄積系を経た後、入力符号列205'として入力復号化装置800に入力される。本実施形態においては、入力復号化装置800が本発明に係る復号化装置に相当する。

入力復号化装置800においては、後続の逆多重化器811からの誤り訂正／検出符号の種類を示すFEC種類識別信号802に基づいて、誤り訂正／検出符号を切り替えながら、誤り訂正／検出復号化された符号列801、同期符号検出信号803および誤り検出信号804が出力される。逆多重化器811は、これらの符号列801、同期符号検出信号803および誤り検出信号804を入力して、予測残差符号841と動き補償適応予測情報符号842を分離して出力する。

予測残差符号841および動き補償適応予測情報符号842は、それぞれ第1および第2の変長復号器806および810に入力される。第1の変長復号器806で復号された残差DCT係数831は、逆量子化器807で逆量子化、逆DCT器808で逆離散コサイン変換という一連の処理が施された後、加算器809で動き補償適応予測器801からの出力である動き補償適応予測信号832と加算されて再生画像信号850となる。再生画像信号850は、装置外へ出力されると共に、フレームメモリ820へ記録される。第2の変長復号器810で復号された動き補償適応予測情報は、動き補償適応予測器801に入力され、動き補償予測信号832が生成される。

以上の処理は、図1の動画像符号化装置に対応して動画像を再生する処理であり、逆量子化器807、逆DCT器808、加算器809およびフレームメモリ820が行なう処理は、それぞれ図1における逆量子化器107、逆DCT器108、加算器109およびフレームメモリ102のそれと実現手段は異なる場合もあるが、基本的に同一である。また、第1および第2の変長復号器806、810、逆多重化器811および入力復号化装置800の処理は、符号列に誤り

が混入した場合を除いて、それぞれ図1における可変長符号化器106、110、多重化器111および出力符号化装置200のそれと逆の処理である。

図9は、入力復号化装置800の構成を示すブロック図である。この入力復号化装置800は、入力符号列205'中の同期符号を検出する同期検出器901、入力符号列205'のビット数を計数するカウンタ902、入力符号列205'を情報ビット912と検査ビット913に分解して出力する符号列分解器903、誤り訂正/検出復号化器904および挿入ビット除去器905から構成される。

同期検出器901では、カウンタ902の計数値911に基づいて同期符号挿入位置でのみ同期符号の検出を行なう。例えば、同期符号挿入位置の間隔をsync__period、計数値911をbit __count とし、同期符号の長さをsync__len とすれば、

$$0 < \text{bit_count} \% \text{sync_period} \leq \text{sync_len}$$

の時のみ同期検出を行なう。

ここで、同期符号の誤りを考慮した同期符号の検出を行なっても良い。

図3中の出力符号化装置200内のビット挿入器211において、nビット以下の誤りを考慮して同期符号とのハミング距離が $2 * n + 1$ 離れるようにビット挿入による符号列変換を行なっていれば、真の同期符号とのハミング距離がn以下のものを同期符号と判定しても、nビット以下の誤りの混入であれば誤同期検出は起こらない。

図10は、符号列分解器903の構成を示すブロック図である。入力符号列205'は後述する制御器1001で制御される第1の切替器1002で情報ビット1021と検査ビット913とに切り替えられる。第1の切替器1002から情報ビット1021が出力された場合、この情報ビット1021は第2の切替器1003を経てバッファ1004に情報ビット長だけ記憶される。カウンタ1005は、第2の切替器1003からの出力ビット数を計数する。このカウン

タ1005の計数値1023は、比較器1006で誤り訂正／検出符号情報出力器1007から出力された情報ビット長1024と比較され、両者が一致したときにカウンタ1005がリセットされると共に、誤り訂正／検出符号の種類を示すFEC種類識別信号802がラッチ回路1008でラッチされ、さらにバッファ1004から情報ビット912が出力される。ラッチ回路1008の出力914は、誤り訂正／検出符号情報出力回路1007へ入力されると共に、図9中に示す誤り訂正／検出復号化器904に出力される。

前述のように、誤り訂正／検出符号の検査ビットは位置ずらしが施され、符号列205中後方の誤り訂正／検出符号の情報ビットの間に入っている。制御器1001は、この位置ずらしが施された検査ビットと情報ビットが分離されるように制御を行なう。1ブロックの誤り訂正／検出符号の情報ビットの入力が終了すると、比較器1006において計数値1023と情報ビット長1024が一致する。制御器1001は、この一致信号を受けて誤り訂正／検出情報出力器1007から検査ビット長1025を取り込み、次の情報ビットの間に入っている検査ビットの位置を計算する。比較器1006で一致と判定されたときの符号列205'の入力ビット数の計数値911をbit __count、検査ビット長をcheck __len とすれば、検査ビット開始位置check __start は

$$\text{check_start} = (\text{bit_count} / \text{sync_period} + 1) * \text{sync_period} - \text{check_len}$$

となり、検査ビット終了位置check __end は、

$$\text{check_end} = (\text{bit_count} / \text{sync_period} + 1) * \text{sync_period}$$

となる。制御器1001は、計数値911がcheck __start からcheck __end までの間、切替器1002を検査ビット913が出力されるように制御される。

1フレームの最後は縮退符号で誤り訂正／検出符号化が行なわれているので、特別な処理が行なわれる。1フレームの最後に達すると、同期検出器901から

次のフレームの同期符号を検出したことを示す信号803が出力される。制御器1001は、この信号803を受けてフレームの最後の誤り訂正/検出符号の検査ビットの位置および情報ビットの不足ビット数を計算する。1フレームの最後の誤り訂正/検出符号を入力し始めたときの入力される符号列205'のビット数の計数値911をpre __last__count、1フレームの符号列205'の入力が終了した時点の計数値911をtotal __count、処理時点での計数値911をbit __count、1フレーム最後の誤り訂正/検出符号の検査ビット長をlast__check __len、それより一つ前の誤り訂正/検出符号の検査ビット長をpre __last__check __len とする。まず、誤り訂正符号が縮退符号であることおよびビット挿入が行なわれていることによる情報ビットの過不足分を計算する。1フレーム最後の誤り訂正/検出符号の情報ビットのうち、出力符号列205中に含まれているビット数last__info__len は、

$$\text{last_info_len} = \text{total_count} - \text{last_check_len} - \text{pre_last_count} - \text{pre_last_check_len}$$

である。last__info__len が誤り訂正符号の情報長info__len より短いときには縮退符号であると判定し、計数値1023がlast__info__len からinfo__len までは切替器1021を挿入ビット発生器1015から出力されるビットパターンに切り替えて、縮退による情報ビットの不足分を補う。この挿入ビット発生器1015からの出力ビットパターンは、符号化器の図7中の挿入ビット発生器705と同一のビットパターンを発生する。

一方、last__info__len がinfo__len より長いときには、ビット挿入分と判定し計数値1023がinfo__len 以上となる部分については情報ビット912を出力しないようにする。検査ビットについては、

$$\text{total_count} - \text{check_len} < \text{bit_count} \leq \text{total_count}$$

の時の出力符号列205を検査ビットとして出力するように切替器1002を制

御する。

誤り訂正／検出復号器904は、符号列分解器903から出力された情報ビット912および検査ビット913を入力し、図10のラッチ回路1008でラッチされた誤り訂正／検出符号の種類を示すFEC種類識別信号914に基づいて誤り訂正／検出符号の復号化を行ない、誤り訂正された符号列915と誤り検出信号804を出力する。

誤り訂正された符号列915は、挿入ビット除去器905に入力される。挿入ビット除去器905は、出力符号化装置200のビット挿入器211で挿入された擬似同期信号を防ぐための挿入ビットを取り除く処理を行なう。前述のように、ビット挿入は同期挿入位置でのみ行なわれているため、計数器902の計数値911に基づいて同期挿入位置を判定する。

例えば、同期符号語が図5に示すものとし、ビット挿入器211において同期符号とのハミング距離が $2 * n + 1$ 以上離れるように同期符号の最初のsync__len ビットの“0000…”の部分にビット挿入が行なわれている場合には、同期符号挿入位置からsync__0__len - ($2 * n + 1$) ビットの“1”の数(=n0とする)を計数し、n0が $2 * n + 1$ 以下であったら $2 * n + 1 - n0$ ビット削除する。ただし、挿入ビットは“1”であると定められているため、挿入ビット除去器905で挿入ビットと判定されたビットが“0”となっていたときは、同期符号挿入区間に誤りが混入したと考えられ、この場合には誤り検出信号804を出力する。

以上のようにして入力復号化装置800で復号された符号列801は、逆多重化器811で逆多重化が行なわれる。これは図2に示したような多重化された符号語を分離して出力する操作である。この逆多重化器811は、第1および第2の可変長復号器806および810と連動して動作する。

まず、出力復号化装置800から同期符号検出信号803が入力されると、逆

多重化器811はフレーム処理の初期状態に移る。次に、誤り訂正／検出符号の種類を示すFEC種類識別信号802としてピクチャヘッダに対する誤り訂正／検出符号の種類を出力し、符号列801を入力してピクチャヘッダ302を復号しピクチャヘッダに誤りが無いか判定する。誤りが無い場合には、FEC種類識別信号802として、予測モード情報303に対する誤り訂正／検出符号の種類を出力し、符号列801を入力して予測モード情報を逆多重化し、第2の可変長復号化器810へ出力する。

第2の可変長復号化器810は、全ての予測モード情報を復号すると、その旨を示す信号を逆多重化器811に出力する。これを受けて、逆多重化器811は動きベクトル情報304に対する誤り訂正／検出符号の種類を示すFEC種類識別信号を出力し、動きベクトル情報304の逆多重化を開始する。逆多重化された動きベクトル情報は第2の可変長復号器810に出力され、復号化される。全ての動きベクトル情報の復号が終了すると、第2の可変長復号器810から逆多重化器811にその旨を示す信号が出力され、これを受けて逆多重化器811は残差DCT係数305に対する誤り訂正／検出符号の種類を示すFEC種類識別信号を出力すると共に、残差DCT係数305を逆多重化して第1の可変長復号器806に出力する。この残差DCT係数305を第1の可変長復号器806で復号する。

以上のように、誤り訂正／検出符号の種類は、逆多重化器811において出力符号化装置200と同一に定められた多重化の規則に基づいて判定される。このため、誤り訂正／検出符号の種類を示すヘッダ情報等を出力符号列205に含める必要はない。

誤り訂正／検出復号化器904において、誤り検出符号により入力符号列205'に誤りが混入していることが検出されることがある。また、前述のように挿入ビット除去器905で挿入ビットの誤りが検出されることがある。これら

の場合には、入力復号化装置800から誤り検出符号804が出力される。さらには、可変長復号化の処理において可変長符号語テーブルに無い符号語が検出された場合にも、誤りが混入したと判定される。また、逆多重化器811における逆多重化処理において多重化の規則に違反する部分があると判定されたときも、誤りが混入したと判定される。これらの場合、入力復号化装置800および逆多重化器811は再生画像に大きな劣化が生じないように、以下のような処理を行なう。

(1) 残差DCT係数で誤りが検出された場合は、当該部分の残差を0とする。予測モードとしてイントラ符号化モードが選択されていた場合は、既再生フレームや周囲の領域の再生画像信号から当該領域の再生画像信号を予測してもよい。

(2) 予測モード情報や動きベクトルで誤りが検出された場合は、周囲の領域の予測モード情報や動きベクトル情報から当該領域の予測モード情報や動きベクトル情報が推定できるときにはそれらを用い、それが不可能ならば既再生フレームや周囲の領域の再生画像信号から当該領域の再生画像信号を予測する。

(3) ピクチャヘッダに誤りが検出された場合は、それをそのまま復号すると非常に大きな画質劣化を生じる可能性があるため、前フレームの再生画像をそのまま現フレームの再生画像とする。

以上の(1)(2)(3)処理において、可変長符号化を用いているために誤りが次の同期符号までの後続の符号にも波及している場合には、その部分についても同様の処理を行なう。

上述の説明では、同期符号検出器901が同期符号挿入位置(sync_periodビットおき)でのみ同期符号の検出を行なう例を示したが、伝送/蓄積媒体によってはビットの消失や誤ビットの挿入が起こる場合もある。このような場合は、同期符号挿入位置以外でも同期符号の検出を行ない、同期符号が検出された位置を同期符号挿入位置と判定すればよい。

(第2実施形態)

次に、図12～図14を参照して本発明に係る第2実施形態について説明する。本実施形態による動画像符号化装置および動画像復号化装置は、ビット列の一部が消失してビット数が減少したり、余分なビットが付加されてビット数が増加するような伝送路／蓄積媒体で符号列の伝送／蓄積を行なっても、同期検出を確実に行なうことができるようにしたものである。

図12は、このようなビット付加／消失のある場合の同期検出の処理の原理を示すブロック図である。ここで、正しい同期符号は図12(a)に示すように、sync__0__len ビットの“0”と、1ビットの“1”からなるとする。なお、図12中の“×”は同期符号以外のビットである。

図12(b)～(e)は、ビット付加／消失によって同期符号がどのように変化するかを示したものである。ここで、付加／消失するビット数(N i d)は最大1ビットとしている。(b)は同期符号より前のビット列中で1ビット削除が起きた場合であり、同期符号全体が1ビット前にシフトしている。(c)は同期符号より前のビット列中で1ビット付加が起きた場合であり、同期符号全体が1ビット後ろにシフトしている。(d)は同期符号中でビット削除が起きた場合であり、図中矢印のビット抜け位置から後ろが1ビット前にシフトしている。さらに、(e)は同期符号中で1ビット付加が起きた場合であり、図中矢印のビット付加位置に1ビット付加が起こり、それより後ろが1ビット後ろにシフトしている。

ビット付加／消失が起きても正しく同期を検出するためには、図12(b)～(d)に示したビット列も同期符号と判定する必要がある。図12から分かるように、正しい同期符号挿入位置の±N i dビットの範囲に含まれる“1”の数は最大

$$\text{sync_0_len} - 3 * N i d$$

ビットである。従って、復号側で同期符号挿入位置の±N i dビットの範囲で同期検出を行ない、この区間に含まれる“1”の数が上記の値以下であったならば同期符号と判定すればよい。また、符号化装置では図12(b)～(d)のビットパターンが生じないように符号列の変換を行なう。

以下、このような符号化装置／復号化装置を第1の実施形態との差異を中心に説明する。

第2実施形態に係る動画像符号化装置は、第1の実施形態に係る動画像符号化装置と全体構成は同じであるが、図3中のビット挿入器211の動作が異なっている。図13に、ビット挿入器211の動作を示す。すなわち、第1の実施形態のビット挿入器211では、同期符号挿入区間でのみビット挿入操作を行っていたが、第2の実施形態におけるビット挿入器211では、最大N i dビットのビット付加／消失が起きても同期符号と同一のビットパターンが生じないようにするため、同期符号挿入区間±N i dビットの区間でビット挿入を行なう。

図3における計数值221をtotal __len、同期符号挿入位置の間隔をsync__periodとすれば、ビット挿入器221は

$$\text{total_len mod sync_period} = \text{sync_period} - N i d \text{ (mod : 剰余演算)}$$

から、

$$\text{total_len mod sync_period} = \text{sync_0_len} - 1 - 3 * N i d$$

の区間での“1”の数(=n 0とする)を計数し、n 0が3 * N i d + 1未満であったら3 * N i d + 1 - n 0ビットの“1”を挿入する。

図13では、sync__period=12、sync__0__len = 9、N i d = 1の場合のビット挿入器211の動作例を示している。この例ではn 0 = 2であるので、3 * N i d + 1 - n 0 = 2ビットの“1”を挿入する。

このようにビット挿入を行なうことにより、同期符号挿入区間±N i dビットの“0”の数は3 * N i dビット以上であることが保証され、同期符号と一意に

識別することが可能である。

一方、第2の実施形態に係る動画復号化装置は、第1の実施形態と全体構成は同一であるが、図9中の同期検出器901および挿入ビット除去器905の動作が異なっている。図14に、挿入ビット除去器905の動作を示す。

すなわち、同期検出器901では最大Nidビットのビット付加／消失が起きても同期を検出するために、同期符号挿入位置の前後±Nidビットの範囲で同期符号の検出を行なう。

まず、同期符号挿入位置毎に同期符号が存在するかを判定する。すなわち、カウンタ902の計数値911をbit __count すると、

$$\text{bit_count} \bmod \text{sync_period} = \text{sync_period} - \text{Nid}$$

から、

$$\text{bit_count} \bmod \text{sync_period} = \text{sync_0_len} - 1 + \text{Nid}$$

の間の“0”の数(=ns0とする)をカウントし、ns0が3*Nid以下であったら、この区間に同期符号があると判定する。

図14の動作では、sync__period=12、sync__0__len = 9、Nid=1の場合の例を示している。この例では(bit __count mod sync__period)が“1”から“8”の間で“0”の数をカウントする。図14の例ではns0=2であるので、同期符号と判定する。

次に、同期符号があると判定された同期符号挿入区間で、ビット付加／消失によって符号列が何ビットシフトしているかを判定する。図14のようなsync__0__len ビットの場合、最後の“1”の位置からシフト量を判定する。具体的には、同期符号判定区間の最初からsync__0__len + 1 ビット目から見て最初にある“1”を探索し、この同期符号判定区間の最初から何ビット目にあるか(=first __1__pos ビットとする)を基に、

$$\text{シフトビット数} = \text{first_1_pos} - (\text{sync_0_len} + 1 + \text{Nid})$$

(負の場合は前方シフト、正の場合は後方シフト)

として求める。図14の例では $\text{first_1_pos} = 10$ であるから、

$$\text{シフトビット数} = 10 - (9 + 1 + 1) = -1$$

となり、前方に1ビットシフトしていることが分かる。

挿入ビット除去器905では、第1の実施形態と異なり、同期符号挿入位置± Nid ビットの区間で挿入ビット除去処理を行なう。すなわち、

$$\text{bit_count mod sync_period} = \text{sync_period} - Nid$$

から、

$$\text{bit_count mod sync_period} = \text{sync_0_len} - 1 - 3 * Nid$$

の区間での“1”の数(=n0とする)を計数し、n0が $3 * Nid + 1$ 以下であつたら、 $3 * Nid + 1 - n0$ ビットの“1”を除去する。

第2の実施形態において、伝送路や蓄積媒体でのビット付加/消失が起きた区間を何らかの形で判定できる場合、ビット付加/消失を考慮した同期検出処理やビット挿入処理、ビット除去処理は、その区間でのみ行なうようにしてもよい。

なお、前述した第1の実施形態に係る動画像復号化装置においても、伝送路や蓄積媒体でのビット付加/消失に対応した同期検出を行なうために、同期検出器901では第2の実施形態と同様に同期符号挿入区間± Nid ビットの区間で同期検出を行なうようにしてもよい。この場合、同期符号以外の部分を誤って同期符号と判定する擬似同期が生じる可能性があるが、ビット付加/消失の起きやすい伝送路/蓄積媒体では同期検出ミスによる復号画像の品質劣化を抑えることができ、画質が向上する。

また、伝送路や蓄積媒体でのビット付加/消失が起きた区間を何らかの形で判定できる場合は、この処理はその区間でのみ行ない、それ以外の区間では通常の同期検出を行なうようにしてもよい。

さらに、上述した第1および第2の実施形態において、フレームの長さを示す

情報（以下、フレーム長情報という）を用いて同期をさらに保護するようにしてもよい。図15、図16および図17は、フレーム長情報POINTERを用いた場合の符号列の例を示したものである。

図15の例では、同期符号PSCの直後にフレーム長情報POINTERと、このフレーム長情報POINTERを保護する誤り訂正／検出符号のチェックビットCHKPが続いている。フレーム長情報POINTERには前フレームのビット数、すなわち前フレームの同期符号から現フレームの同期符号までのビット数を示す情報が記録されている。

符号化装置では、1フレームの符号列ビット数をカウントして、これをフレーム長POINTERに変換し、さらに誤り訂正／検出符号化を行なってチェックビットCHKPを生成する。そして、これらを図15に示すように次フレームの同期符号の直後に続けて符号列を生成する。

一方、復号化装置では、第1および第2の実施形態に述べたのと同様の方法で同期符号の検出を行なった後、それに続くフレーム長情報POINTERとチェックビットCHKPを符号列から取り出し、誤り訂正／検出復号を行なってフレーム長情報POINTERを復号する。そして、復号されたフレーム長情報POINTERと、1つ前に検出された同期符号から現在の同期符号の位置までのビット数をカウントして求めた値（フレーム長カウント値）を比較し、同期符号の誤検出が無いかをチェックする。

もし、フレーム長カウント値がフレーム長情報POINTERに示された前フレームの符号長と異なる場合には、同期符号の誤検出が起こった可能性があるため、フレーム長情報POINTERを用いて誤検出された同期符号の再検出を行なう。すなわち、現在の同期符号からフレーム長情報POINTERで示されたビット数だけ前方に検出できなかった同期符号があると見なすこととする。この場合、1つ前に検出された同期符号から現在の同期符号までの区間は、1つ前の

同期符号からフレーム長情報 P O I N T E R で示される位置までの区間と、そこから現在の同期符号までの区間の二つのフレームに分割して復号処理を行なう。

ただし、フレーム長情報 P O I N T E R で示されるビット数で、1つ前に検出された同期符号から現在の同期符号の位置までのビット数よりも多い場合には、フレーム長情報 P O I N T E R が誤っているものと見なし、上記の同期再検出処理は行なわない。

フレーム長情報 P O I N T E R およびチェックビット C H K P のビット数が多い場合には、図 1 6 に示すように同期符号 P S C、フレーム長情報 P O I N T E R、チェックビット C H K P が複数の同期区間にわたるようにしてもよい。この場合、同期符号以外の符号列が同期符号と一定のハミング距離を保つようにするための符号化装置でのビット挿入処理および復号化装置でのビット削除処理は、フレーム長情報 P O I N T E R およびチェックビット C H K P が存在する区間では行なわないようにしてもよい。

図 1 5 および図 1 6 の例において、同期符号 P S C の後半にその同期符号の種類を示す情報（フレーム同期符号、G O B 同期符号の区別等）が含まれている場合、フレーム長情報 P O I N T E R だけでなく、同期符号 P S C の後半ビットを誤り訂正符号で保護するようにしてもよい。これにより、同期符号の位置だけでなくその種類も正確に検出できるため、誤りに対する耐性がさらに向上する。

図 1 7 の例では、フレーム長情報 P O I N T E R およびチェックビット C H K P をフレームの最後（次のフレームの同期符号の直前）に置く例である。この場合、復号化装置では、次のフレームの同期符号を検出した後にその直前のフレーム長情報 P O I N T E R およびチェックビット C H K P を取り出して誤り訂正／検出復号を行ない、図 1 5 および図 1 6 と同様の処理により同期符号の再検出を行なう。

図 1 5 の例においては、同期符号は同期符号挿入位置にのみ存在するので、フ

フレーム長情報 P O I N T E R はフレームのビット数を同期符号挿入間隔 (=sync _period ビット) で割った値を記録してもよい。これにより、フレーム長を少ないビット数で表すことができる。

また、第1および第2の実施形態では、誤り訂正/検出符号を符号化する情報の重要度に応じて変化させる階層符号化を行なう例を示したが、フレーム内で同一の誤り訂正/検出符号を用いるようにしてもよいし、誤り訂正/検出符号を用いなくともよい。そのような場合でも、本実施形態で示したような同期符号以外の符号列が同期符号と予め定められた値以上のハミング距離を保つためのビット挿入処理、およびこれに対応した同期符号検出処理によって、同期を検出する能力は従来方式より向上する。

また、以上の各実施形態の説明では動画像信号を高能率圧縮符号化して伝送/蓄積する例を示したが、本発明は静止画像や音声、データ等の伝送/蓄積にも応用することが可能である。例えば、静止画像信号を直交変換を用いて高能率圧縮符号化する場合には、変換係数の低域成分をより強く誤り保護するように誤り訂正/検出符号を切り替えて用いればよい。音声を駆動源と声道フィルタにモデル化して符号化する方法では、ピッチ周期や声道パラメータ等を強く誤り保護するように誤り訂正/検出符号を切り替えればよい。

(第3実施形態)

次に、本発明に係る第3実施形態について説明する。本実施形態は、誤り訂正/検出符号を用いないことが第1および第2実施形態と異なる。

図18は、本実施形態に係る動画像符号化装置のブロック図である。図1と対応する部分に同一符号を付して第1の実施形態との相違点を中心に説明すると、本実施形態では出力符号化装置200の構成および動作が異なる。また、多重化器111は、基本動作は図1の多重化器111と同一であるが、誤り訂正/検出符号を用いないことに伴って、多重化符号列201および同期符号挿入要求信号

203のみを出力する。

図19は、図18中の出力符号化装置200の構成を示すブロック図である。この出力符号化装置200は、出力符号列205のビット数を計数するカウンタ1701、出力符号列205を切り替える切替器1703、切替器1703を制御する切替制御器1704、およびスタッフィングビットを発生するスタッフィングビット発生器1705から構成される。

図20は、図19の出力符号化装置200で生成される出力符号列205の一例を示す図である。図4の出力符号列と相対応する符号語には同一記号を用いて表記してある。図4と同様に同期符号PSCは周期的に、つまり一定区間(sync__periodビット)おきに配置された矢印で示される同期符号挿入位置の何れかのみに挿入されている。図20では、誤り訂正/検出符号の検査ビットCHKが含まれないところが図4と異なる。出力符号列205の1フレームの最後の部分には、同期符号PSCが同期符号挿入位置に挿入されるように、スタッフィングビットSTUFFが挿入される。スタッフィングビットSTUFFのビット数は、sync__periodビット以下となる。

以下、このような出力符号列205を生成する図19の出力符号化装置200の構成と動作について詳細に説明する。

カウンタ1701は、多重化器111から同期符号挿入要求信号203が入力され、多重化符号列201として同期符号301の最初のビットが入力されると“1”にセットされ、同期符号301の全ビットが入力されると同期符号長sync__lenにセットされる。その後、カウンタ1701は同期符号301の次のビットから順に次の同期符号の直前のビットが出力されるまでカウントアップしてゆく。

切替制御器1704は、多重化符号列201として同期符号の最初のビットから次の同期符号の前のビットまでが入力されている時は、切替器1703を多重

化符号列201の側に切り替え、多重化符号列201が出力符号列205として出力されるように制御される。

そして、1フレームの最後の部分では、次の同期符号が同期符号挿入位置に挿入されるようにするためのビット挿入（ビットスタッフィング）を行なう。多重化器111は、1フレームの多重化符号列201の出力が終了すると、次のフレームの同期符号挿入要求信号203を出力する。これを受けて切替制御器1704は、切替器1703をスタッフィングビット発生器1705の側に切り替え、スタッフィングビット1223を出力符号列205として出力する。このスタッフィングビット1223は全ビットを“1”としてもよいし、“0”としてもよいし、“0101…”のような特定のパターンとしてもよい。

次に、本実施形態に係る動画像復号化装置について説明する。

図21は、図18の動画像符号化装置に対応した動画像復号化装置の構成を示すブロック図である。図8と相対応する部分に同一符号を付して第1の実施形態との相違点を中心に説明すると、本実施形態では入力符号化装置800の構成および動作が異なる。また、入力復号化装置800から逆多重化器811に入力される信号は符号列801および同期符号検出信号803のみであり、逆多重化器811から入力復号化装置800に入力される信号が無い。

図22は、入力復号化装置800の構成を示すブロック図である。この入力復号化装置800は、入力符号列205'中の同期符号を検出する同期検出器1901と、入力符号列205'のビット数を計数するカウンタ1902から構成される。

カウンタ1902は、復号化の最初の段階では“0”にリセットされており、入力符号列205'が1ビット入力される度に計数値1911を“1”ずつカウントアップしていく。

同期検出器1901では、カウンタ1902の計数値1911に基づいて同期

符号挿入位置でのみ同期符号の検出を行なう。例えば、同期符号挿入間隔をsync__period、計数值1911をbit __count とし、同期符号の長さをsync__len とすれば、

$$0 < \text{bit_count} \bmod \text{sync_period} \leq \text{sync_len}$$

の時のみ同期検出を行なう。ここで、 $A \bmod B$ はAをBで除したときの剰余を表す。同期検出器1901は、同期符号を検出すると同期符号検出信号803を出力する。

入力復号化装置からの符号列801は、入力符号列205'がそのまま出力され、逆多重化器811に入力される。以降、図21の動画像復号化装置と同様に逆多重化、復号化の処理が行なわれる。

フレームの最後のスタッフィングビットSTUFFを予め定められたビットパターンとした場合、逆多重化器811においてこのスタッフィングビットSTUFFが予め定められたパターンと一致するか判定し、一致しなかった場合には入力符号列205'中に誤りがあったと判定して、第1の実施形態で示した動画像符号化装置で述べたような画質劣化が大きくなるようにするための処理を行なってもよい。

(第4実施形態)

次に、本発明に係る第4実施形態について説明する。

本実施形態に係る動画像符号化装置はその全体構成は図18の動画像符号化装置と同一であり、出力符号化装置の動作が第3実施形態と異なる。

図23は、図18中の出力符号化装置200の構成を示すブロック図である。図19の出力符号化装置と相対応する部分に同一の符号を付してその差異を中心に説明すると、擬似同期符号を防止するためのビットスタッフィング処理を行なうビット挿入器1211が追加されている。

ビット挿入器1211において、多重化符号列201に対して擬似同期の発生

を防ぐためのビット挿入が行なわれる。これは、出力符号列205中に同期符号と同一のビットパターンがあると同期符号を一意に復号できなくなるため、これを防ぐための処理である。例えば、同期符号が図5に示すようにsync__0__lenビットの“0”、1ビットの“1”および同期符号の種類を示すsync__nb__lenビットの“×××××”からなる場合、同期符号以外の符号列中にsync__0__lenビット以上“0”が連続しないように“1”を挿入すれば擬似同期の発生を防ぐことができる。

同期符号は同期符号挿入装置にしか挿入されない。従って、擬似同期発生防止のためのビット挿入操作は同期符号挿入位置だけで行なえばよい。そこで、出力符号列205の総ビット数を示す計数値1221に基づいてビット挿入が必要であるかを判定する。計数値1221をtotal__lenとすれば、

$$0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len}$$

となる区間で多重化符号列201中の“1”の数を計数し、この区間に“1”が全く無くなければ1ビットの“1”を挿入する。ここで、 $A \bmod B$ はAをBで除したときの剰余を表す。

また、誤りによる同期符号の誤検出の確率を低くするためには、以下のようにビット挿入を行なえばよい。

同期符号にnビット誤りが混入した場合にも同期符号を検出するためには、後述する動画像符号化装置の入力符号化装置で真の同期符号とハミング距離がn以下の符号語は同期符号と判定する必要がある。しかし、同期符号以外の符号列をそのままにしてこのような判定を行なうと、同期符号以外の符号列中にも同期符号とハミング距離がn以下のビットパターンが存在することがあり、これが同期符号挿入位置にあると誤って同期符号と判定されてしまう。

そこで、ビット挿入器211で多重化符号列201に以下のようにビット挿入を行なうことによって、多重化符号列201中の同期符号挿入位置にある同期符

号以外の符号列を同期符号とのハミング距離が $2 \times n + 1$ 以上離れるように変換する。具体的には、

$$0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len} - (2 \times n + 1)$$

となる区間での“1”の数(=n0とする)を計数し、n0が $2 \times n + 1$ 未満であつたら $2 \times n + 1 - n0$ ビットの“1”を多重化符号列201に挿入する。

こうしてビット挿入が行なわれた後の符号列1222は、図19の出力符号化装置と同様にフレームの最後の区間のビット挿入(図20中STUFF)が行なわれ、出力符号列205として出力される。

次に、本実施形態に係る動画像復号化装置について説明する。動画像復号化装置の全体構成は図21の動画像復号化装置と同一であり、入力符号化装置800の動作が第3実施形態と異なる。

図24は入力復号化装置800の構成を示すブロック図である。図22の入力復号化装置と相対応する部分に同一の符号を付して第3の実施形態との差異を中心に説明するビット除去器1905が追加されている。

入力符号列205'は、挿入ビット除去器1905に入力され、図23の出力符号化装置のビット挿入器1211で挿入された擬似同期符号を防ぐための挿入ビットを取り除く処理を行なう。前述のように、ビット挿入は同期符号挿入位置でのみ行なわれているため、計数器1902の計数値1911に基づいて同期符号挿入位置を判定する。

例えば、同期符号が図5に示す符号語であるものとし、ビット挿入器1211において同期符号とのハミング距離が $2 \times n + 1$ 以上離れるように同期符号の最初の“0000…”の部分にビット挿入が行なわれている場合には、同期符号挿入位置から $\text{sync_0_len} - (2 \times n + 1)$ ビットの“1”の数を計数し、n0が $2 \times n + 1$ 未満であつたら $2 \times n + 1 - n0$ ビット削除する。

ここで、挿入ビットは“1”と定められているため、挿入ビットと判定された

ビットが“0”となっていたときは、同期符号挿入区間に誤りが混入したと考えられる。この場合は、図示しない誤り検出信号を逆多重化器811に出力し、第1の実施形態で述べたのと同様に、再生画像に大きな劣化が生じないように処理を行なってもよい。

図23のビット挿入器1211でのビット挿入処理は、同期符号以外の全ての同期符号挿入区間にあらかじめ定められたビット数の挿入ビットを挿入するようにしてもよい。25はこのようなビット挿入処理を行なったときの出力符号列205の一例を示したものである。図中SBが挿入されたビットを示している。

例えば、同期符号が図5に示すようにsync_0_len ビットの“0”、1ビットの“1”および同期符号の種類を示すsync_nb_len ビットの“×××××”からなる場合、同期符号挿入区間の最初からsync_0_len ビットの区間の予め定められた位置に1ビットの挿入ビットSBを挿入する。

挿入ビットSBは、常に“1”としてもよい。また、同期符号挿入区間の最初からsync_0_len ビットの区間のビットパターンに応じて、その区間の“1”の数が1個以上となるように適応的に挿入ビットSBを決定してもよい。

さらに、挿入ビットSBを同期符号挿入区間の最初からsync_0_len ビットの区間の奇パリティとすることにより、同期符号と同一のビットパターンの出現を避けると共に、このビットパターンに混入した誤りの検出をも行なうことが可能である。

図25(b)は、このようなビット挿入処理を行なった出力符号列を示している。この例では、同期符号挿入位置から最初の部分に1ビットの挿入ビットSBを挿入している。この挿入ビットSBは、これと次のビットからsync_0_len - 1ビットを合わせた区間の“1”の数が必ず奇数となるように決定する。例えば、図25(b)の左側の例では、挿入ビットSBは“1”となる。また、図25(b)の右側の例では、挿入ビットSBの次のビットからsync_0_len -

1ビットの区間が全て“0”であっても挿入ビットSBが“1”になるため、同期符号挿入区間に必ず1ビット以上の“1”が入ることになり、同期符号と同一のビットパターンは生じない。また、挿入ビットSBはパリティチェックとしても働くため、この区間に混入したビット誤りを検出することが可能である。

また、挿入ビットSBを次の同期符号挿入位置までの全ビットの奇パリティチェックビットとしてもよい。但し、挿入ビットのSB次のビットからsync_0__len - 1ビットが全て“0”の場合のみ同期符号と同一のビットパターンの出現を避けるために、挿入ビットSBを必ず“1”とする。これにより、全ビットのパリティチェックによる誤り検出を行なうことが可能である。

誤りによる同期符号の誤検出の確率を低くするためには、より多くのビット数を挿入することが望ましい。例えば、nビットの誤りが入っても同期を正しく検出するためには、この区間の予め定められた位置に $2 \times n + 1$ ビットの“1”を挿入する。

本実施形態では、上述のビット挿入器1211の動作に対応して、図24のビット除去器1905の動作も異なったものになる。すなわち、ビット除去器1905ではビット挿入器1211でビット挿入を行なった予め定められた位置の挿入ビットSBを削除する処理を行なう。

ここで、挿入ビットSBを常に“1”とした場合には、入力符号列205'中のビット挿入位置にあるビットが“0”であったときビット誤りと判定して、誤り検出信号（図示しない）を逆多重化器811に出力し、復号画像に大きな劣化を生じないようにするための処理を行なってもよい。

第1ないし第4実施形態においては、多重化器111における多重化は図2のように予測モード情報303、動きベクトル情報304、および、残差DCT計数305を符号化フレームを単位としてまとめて多重化する例を示したが、図26のように符号化領域（例えばマクロブロック、GOB等）単位に予測モード情

報303、動きベクトル情報304、残差DCT計数305をまとめるように多重化してもよい。この場合、ピクチャヘッダ302と、それ以外の情報とで異なる誤り訂正／検出符号を用いてもよいし、同一の誤り訂正／検出符号を用いてもよい。あるいは、ピクチャヘッダのみに誤り訂正／検出符号を用いてもよいし、各フレームの予め定められたビット数の一部の符号列に誤り訂正／検出符号を用いてもよいし、誤り訂正／検出符号を全く用いなくともよい。

また、フレーム（ピクチャ）単位だけでなく、フレーム内の部分領域や、複数のフレームをまとめたレイヤ単位で多重化を行ない、これらの多重化単位（レイヤ単位）毎に同期符号を挿入してもよい。

図28は、このような多重化の例を示したものである。この図28の例では、複数の符号化ブロックをまとめたマクロブロック、複数のマクロブロックをまとめたGOB、ピクチャ（フレーム）、複数のピクチャをまとめたセッションの4つのレイヤ毎に多重化を行なう。これらのうち、セッション、ピクチャ、GOBの各レイヤは、それぞれの同期符号（図中SSC, SEC, PSC, GSC）を用いる。SSC, SEC, PSC, GSCにはそれぞれ異なる符号を用い、どのレイヤの同期符号が検出されたかを区別できるようにする。図5のような同期符号を用いる場合、同期符号の種類を示すsync_nb_len ビットの部分でこれらの同期符号の区別を行なえばよい。

このような多重化を行なった場合も、セッション、ピクチャ、GOBの同期符号の一部ないしは全てに対して前述の実施形態におけるフレーム同期符号と同様の処理を行なってもよい。図29は、このような処理を行なった出力符号列の一例を示したものである。同図に示されるように、PSC, GSCの前にスタッフィングビットSTUFFが挿入されており、SSC, PSC, GSCは図中矢印の同期符号挿入位置に挿入されている。このため、前述の実施形態でフレーム同期符号PSCについて説明したのと同様に、各同期符号の検出精度が向上する。

図15、図16および図17のフレーム長情報POINTERと同様の長さ情報をセッション、ピクチャ、GOBの各同期符号に付加することも可能である。図15および図16のように、フレーム長情報POINTERに対して誤り訂正／検出符号による保護を行なっている場合、フレーム長情報POINTERのみでなく、同期符号の種類を示すsync_nb_len ビットの部分に対しても併せて誤り訂正／検出符号を用いることにより、同期符号の位置のみでなく、その種類も正しく検出できる確率が向上する。また、セッション、ピクチャ、GOBのヘッダ情報（図中SH, PH, GF）の一部または全部も併せて誤り訂正／検出符号を用いて保護を行なってもよく、これによって各ヘッダ情報の誤り耐性も向上する。

本実施形態のように擬似同期符号を防止するためのスタッフィング処理を行なう場合、以下のような処理を行なって同期符号挿入間隔sync_periodを同期符号の長さ以下にすることも可能である。

まず、動画像符号化装置の出力符号化装置における処理について説明する。ここで、同期符号は図5に示すようにsync_0_len ビットの“0”と1ビットの“1”からなる符号語とする。図23の出力符号化装置において、ビット挿入器1211から出力したビット数を示す計数值1221をtotal_len とすれば、total_len を同期符号挿入間隔sync_periodで除した剰余が、同期符号の最初の“0”のビット数sync_0_len から1を引いた値をsync_periodで除した剰余と一致したとき、すなわち、

$$\begin{aligned} & \text{total_len} \bmod \text{sync_period} \\ &= (\text{sync_0_len} - 1) \bmod \text{sync_period} \end{aligned} \quad (1)$$

となったとき、そのときの出力ビットから(sync_0_len - 1)ビット前までの出力ビット中の“1”の数(n1とする)を計数し、“1”が全くなかったら(すなわち、n1 = 0であったら)1ビットの“1”を挿入する。

図33(a)に、このような処理を行なった出力符号列の例を示す。図中、下向き矢印が同期符号挿入位置を示しており、同期符号は23ビットの“0”（すなわち、 $\text{sync_0_len} = 23$ ）と、1ビットの“1”からなる。図の例では同期符号挿入間隔 sync_period は同期符号の長さ（=24ビット）よりも短い8である。

図中、区間1～4は前述の n_1 を計数する区間を示している。各区間で順次“1”の数 n_1 を計数し、 $n_1 = 0$ であったらば、その区間の次のビットにスタフピングビットを挿入する。区間1では $n_1 > 0$ なので、スタフピングの必要はない。区間2では $n_1 = 0$ となるので、この区間の次に1ビットのスタフピングビット3301を挿入する。区間3ではスタフピングビット3301が挿入されたことにより、 $n_1 = 1$ となるため、スタフピングビットを挿入する必要はない。

以上のようなビットスタフピング処理を行なうことにより、出力符号列中の同期符号以外の部分で同期符号と同一のビットパターンが無くなるため、擬似同期を生じることが無くなる。

一方、伝送路誤りによる同期符号の誤検出の確率を低くするためには、以下のようにビット挿入を行なえばよい。

同期符号に n ビット誤りが混入した場合にも、同期符号を正しく検出するためには、ビット挿入1211において、出力符号列中の同期符号以外の部分と同期符号とのハミング距離が $2 \times n + 1$ 以上になるようにビット挿入処理を行なえばよい。

これは、図23における出力符号列205の総ビット数を示す計数值1221 total_len を同期符号挿入間隔 sync_period で除した剰余が、同期符号の最初の“0”のビット数 sync_0_len から $2 \times n + 1$ を引いた値を sync_period で除した剰余と一致したとき、すなわち、

$$\text{total_len mod sync_period}$$

$$= (\text{sync_0_len} - (2 \times n + 1)) \text{ mod sync_period} \quad (2)$$

となったとき、そのときの出力ビットから $(\text{sync_0_len} - (2 \times n + 1))$ ビット前までの出力ビット中の“1”の数 (n_1 とする) を計数し、“1”の数が $(2 \times n + 1)$ 未満、すなわち

$$n_1 < 2 \times n + 1$$

であったら、 $(2 \times n + 1 - n_1)$ ビット“1”を挿入する。

図5に示したような複数ビットの“0”から始まる同期符号を用いる場合、同期符号の直前のビット列中の“1”の数が少ないとこの部分の同期誤検出が起こることがある。これを防ぐために、同期符号からその直前の同期符号挿入位置までの sync_period ビットの区間の“1”のビット数が $2 \times n + 1$ ビット以上になるように、フレームの最後の区間のビット挿入 (図20中、STUFF) を出力するようにしても良い。

これには、必ず $2 \times n + 1$ ビット以上の“1”が含まれる STUFF を用いてもよいし、あるいは出力符号列に応じて STUFF を決定するようにしてもよい。すなわち、STUFF を含めた出力符号列中の、同期符号の直前の sync_period ビットの“1”のビット数が $2 \times n + 1$ ビット以上になるように、STUFF を決定してもよい。

図33 (b) に、このような処理を行なった出力符号列の例を示す。図中、区間1～4は前述の n_1 を計数する区間を示している。各区間で順次“1”の数 n_1 を計数し、 $n_1 < 2 \times n + 1$ であったらその区間の次のビットにスタッフィングビットを挿入する。区間2では $n_1 = 1$ となるので、この区間の次に $(2 \times n + 1) - 1 = 2$ ビットのスタッフィングビット 3311 を挿入する。区間3ではスタッフィング 3311 が挿入されたことにより $n_1 = 3$ となるため、スタッフィングビットを挿入する必要は無い。

さらに、同期符号の直前の部分での同期符号誤検出を防ぐために、以下のよう
にSTUFFを決定する。STUFFの直前のビットを3312とする。このビ
ット3312の直前の同期符号挿入位置から直後の同期符号挿入位置までのsync
__periodビット間（区間5）の“1”の数は1ビットしかないため、図5のよう
な複数の“0”が連続する同期符号を用いる場合にはこの部分で同期誤検出を生
じる可能性がある。このため、同期符号を挿入する位置を次の同期符号挿入位置
までずらし、多くの“1”を含むSTUFF3313を出力する。これにより、
同期符号の直前のsync__periodビット区間（区間6）には $2 \times n + 1$ ビット以上
の“1”が含まれるようになるため、同期誤検出を防止することができる。

以上のようなビットスタフティング処理を行なうことにより、出力符号列中の
同期符号以外の部分で同期符号とのハミング距離を $2 \times n + 1$ 以上にすることが
できるため、同期誤検出の確率が減少する。

次に、動画像復号化装置の入力復号化装置における処理について説明する。図
24のビット除去器1905において、入力した符号列のビット数を示す計数値
1905をtotal __len とすれば、total __len が1の条件を満たす値になった
時点で、そのときの入力ビットから(sync__0__len - 1)ビット前までの入力
ビット中の“1”の数(n1とする)を計数し、“1”が全くなかったら、すな
わち、 $n1 = 0$ であったら、1ビット削除する。

同期符号にnビット誤りが混入した場合にも同期符号を検出するために、ビッ
ト挿入器1211で出力符号列中の同期符号以外の部分と同期符号とのハミング
距離が $2 \times n + 1$ 以上になるようにビット挿入処理を行なった場合には、次のよ
うな処理を行なえばよい。total __len が式(2)を満たす値になった時点で、
そのときの入力ビットから(sync__0__len - ($2 \times n + 1$))ビット前までの
出力ビット中の“1”の数(n1とする)を計数し、“1”の数が($2 \times n + 1$)
未満、すなわち

$$n1 < (2 \times n + 1)$$

であったら $(2 \times n + 1 - n1)$ ビット削除する。

出力符号化装置および入力復号化装置において以上のような処理を行なって、同期符号挿入間隔 `sync__period` を同期符号の長さ以下の短いビット数とすることにより、スタッフィングビット `STUFF` のビット数が削減され、符号化効率力が向上する。特に、同期符号の長さが長い場合や、多くの同期符号が挿入される場合には、このスタッフィングビット `STUFF` のビット数削減による符号化効率向上の度合いが大きい。例えば、動画像符号化における `GOB` / スライスのように、画面を1ないし複数のマクロブロック、あるいはマクロブロックライン毎に区切り、その単位毎に同期符号を挿入する方式においては、多くの同期符号が挿入されるため、`STUFF` のビット数削減による符号化効率の度合いがより大きくなる。

また、図28のように複数のレイヤ構造の多重化を行なう場合には、レイヤに依じて異なる長さの同期符号を用いるようにしても良い。

図34(a)は、このような同期符号の例を示したものである。4種類の同期符号のうち、`SSC`、`SEC`、`PSC`はいずれも23ビットの“0”、1ビットの“1”、および同期符号の種類を示す8ビットの合計32ビットからなる。一方、`GOB`レイヤの同期符号`GSC`は、16ビットの“0”と1ビットの“1”からなる17ビットの同期符号であり、他の同期符号より短い符号語である。

このように`GSC`のみを短い符号語としたのは、`GOB`は1ないし複数のマクロブロック (`MB`) からなる、画面内を小領域に区切った符号化単位であることから、一般に`GOB`レイヤの同期符号は他の同期符号に比べて数が多く、同期符号語長を短くすることで、出力符号列の符号量を削減することができるためである。また、このようにすると、同じ符号量ならばより多くの`GSC`を出力することができ、画面内をより細かな`GOB`領域に区切って符号化することが可能にな

るため、伝送路誤りが発生したときの再生画像の品質が向上する。

第4実施形態で説明したような擬似同期を防止するための処理、すなわち、同期符号以外の符号列中に同期符号と同一のビットパターンを生じないようにするビットスタッフィング処理を行なうようにしてもよい。伝送路誤りによる同期符号の誤検出の確率を低くするためのビットスタッフィング処理、例えば、ビット長の長い同期符号（図34（a）の例ではSSC、SEC、PSC）と同一のビットパターンが生じないことだけが保証されているビット列に対しても、最も短い同期符号（図34（a）の例ではGSC）と同一のパターンが生じないようにビットスタッフィング処理を行えば、全ての同期符号と同一のビットパターンが生じないようにすることができる。この処理は、全レイヤの符号列に対して行なってもよいし、最も短い符号を用いるレイヤより下位のレイヤ（図の例ではGOBレイヤ、マクロブロックレイヤ）の符号列に対して行なってもよいし、そのレイヤよりも一つ上のレイヤ以下のレイヤ（ピクチャレイヤ、GOBレイヤ、マクロブロックレイヤ）の符号列に対して行なってもよい。さらに、この処理を予め定められたレイヤの符号列に対してのみ行なってもよい。

伝送路誤りが生じても長さの異なる同期符号同士を識別しやすくするために、同期符号語およびその前後の処理を以下のようにしてもよい。

(i) 複数ビットの“0”と、それに続く“1”からなる同期符号を用いる場合には、長い符号語の短い符号語で、この“1”の同期符号挿入位置から見た相対位置が異なるようにしてもよい。図34（b）の例では、PSCの“1”3411と、GSCの“1”3412は異なる位置にあり、別の同期符号中の同位置のビット（3411に対する3413、および3412に対する3414）はいずれも“0”である。このようにする事により同期符号およびその部分列のハミング距離が大きくなるため、伝送路誤りが生じても異なる同期符号同士を識別しやすくなる。

(ii) 長さの短い同期符号の前にスタッフィングビットを挿入しても良い。例えば、図中3401のように長さの短いGSCの前に1ないし複数の“1”からなるスタッフィングビット3401を挿入することにより、OSCと他の同期符号の部分列とのハミング距離を大きくすることができる。

(iii) 長さの短い同期符号の後ろにスタッフィングビットを挿入してもよい。例えば、長い同期符号中の同期符号の種類を識別する部分とのハミング距離が大きくなるように、GSCの後ろにビット挿入3402を行なってもよい。

(第5実施形態)

次に、本発明に係る第5実施形態について説明する。

本実施形態による動画像符号化装置および動画像復号化装置の全体の構成は第1の実施形態と同様であり、出力符号化装置200および入力復号化装置800の1同期区間の最初および最後の部分での処理がこれまでの実施形態と異なっている。

図27(a)(b)(c)は、本実施形態における動画像符号化装置からの出力符号列205の一例を示す図である。この出力符号列205は、同期符号PSCの後に前フレーム(フレーム $n-1$)の符号列の一部2701が入っており、この符号列2701と現フレーム(n フレーム)の符号列との境界2703(現フレームの符号列のスタートポイント)、言い換えれば多重化符号列の境界を示すポイント情報2702(SA)があり、これに伴い1フレームの最後のスタッフィングビット(図4中STUFF)が無くなっていることが図4の出力符号列と異なっている。

動画像符号化装置内の出力符号化装置200においては、各同期符号挿入位置においてそのフレームの残りの符号列のビット数 $resid_bit$ をチェックする。 $resid_bit$ と同期符号PSCとポイント情報SAのビット数の和が同期符号挿入間隔 $sync_period$ ビットよりも少ない場合には、そのフレームの残りの符号列

を出力符号列205中に出力する前に、同期符号PSCを出力する。次に、ポインタ情報SA（この場合、resid_bitを表す）を出力し、その後に残りの符号列2701を出力する。その後、次のフレームの符号列を出力していく。

動画復号化装置内の入力符号化装置800においては、各同期符号挿入位置において同期符号の検出を行ない、同期符号が検出された場合にはその後ろにポインタ情報SAとフレームの残りの情報が続いているものとして処理を行なう。

図27中のフレームn-1とフレームnの境界部を例に説明すると、同期符号PSCの直前2704までの復号処理を終えた後、その後ろの同期符号挿入区間で同期符号を検出し、同期符号が検出された場合には、次にポインタ情報2702を復号化し、フレームn-1の符号列があと何ビットあるかを求める。これに基づいて、ポインタ情報の直後の符号列からポインタ情報で示されるビット数を取り出し（図27中2703まで）、これが2704に続くものとして符号列801を出力する。その後ろの符号列（図27中2703から）は次のフレーム、すなわちフレームnとして処理を行なう。

本実施形態においては、図27（a）に示すように出力符号列の一部または全てに対して誤り訂正／検出符号化を行なってもよい。この場合、誤り訂正／検出符号の種類は全て同じでもよいし、異なるものを用いてもよい。

また、図27（b）に示すように、誤り訂正／検出符号化を行なわないようにしてもよい。

さらに、図27（c）のように、図15や図16に示したような1フレームの符号列ビット数を示すフレーム長情報POINTERを挿入してもよい。この場合、フレーム長情報POINTERは、そのフレームの同期符号PSCから次のフレームの同期符号PSCまでのビット数を表すようにしてもよい。

図27（a）のように誤り訂正／検出符号化を行なう場合、同期符号PSCから次の検査ビットCHKまでの部分の誤り訂正／検出符号化は、ポインタ情報S

A、フレーム $n-1$ の残りの符号列2701および、2703以降のフレーム n の符号列をまとめてひとつの情報ビットとして誤り訂正／検出符号を行なう。

ポインタ情報SAは誤り訂正／検出符号化を行なった情報としてもよい。この場合、同期符号PSC（あるいはその一部）、フレーム長情報POINTERとポインタ情報を合わせて誤り訂正／検出符号化を行なうようにしてもよい。

次に、スタッフィングビットSTUFFの具体例を説明する。

図30(a)(b)は、上述したスタッフィングビットSTUFFの具体例としてのスタッフィングビットSTUFFの符号表の例を示した図である。図30

(a)(b)のいずれも、出力符号例の逆方向から一意に復号できることが特徴であり、これによってスタッフィングビットSTUFFの開始位置を一意に特定できる。従って、スタッフィングビットSTUFFの直前の符号列の復号終了位置と、スタッフィングビットSTUFFの開始位置を比較することにより、符号列中に混入した誤りを検出できると共に、同期符号から逆方向に復号を行なう符号化方式に用いた場合に逆方向復号の開始地点を特定することができる。

さらに、図30(a)(b)の符号表に示すスタッフィングビットSTUFFは、その最初のビットが必ず“0”になっており、後述のような簡易的な復号による誤り検出が可能である。

図31は、図30(a)(b)の符号表に示したスタッフィングビットSTUFFを含む符号列の復号処理の例を示したものである。図31では、同期符号挿入位置の直前のスタッフィングビットの例を示しているが、これ以外の任意の同期符号挿入位置の直前にスタッフィングビットを挿入して同様の処理を行なうことも可能である。図31中、矢印3101～3103は順方向に復号化を行なった場合のスタッフィングビットSTUFFの直前の符号列（“×××…”で示す）の復号終了位置の例であり、矢印の右端が復号終了位置を表す。符号列に誤りが混入しておらず、正常に復号化が行なわれた場合には、矢印3101のようにス

タッピングビットSTUFFの直前の符号列の復号終了位置とタッピングビットSTUFFの開始位置は一致する。

一方、符号列に誤りが混入した場合には、タッピングビットSTUFFの直前の符号列の復号終了位置は矢印3102, 3103のようにタッピングビットSTUFFの開始位置とずれる。このような場合には、符号列に誤りがあったと判定する。

復号化装置においては、タッピングビットSTUFFの直前の符号列の復号化が終了すると、次の同期符号挿入位置までタッピングビットSTUFFを読み込み、それが図30(a)(b)に示した符号表の符号に合致するかどうかを判定する。もし、タッピングビットSTUFFが符号表のいずれの符号にも合致しない場合は、誤りがあったと判定する。

タッピングビットSTUFFと符号表との合致を判定する場合、少数のビットの誤りを許容するようにしてもよい。こうすることにより、タッピングビットSTUFFそのものに誤りが入った場合の誤りの誤検出を低減することができる。

図30(a)の符号表は、必ず“0”で始まり後続のビットが“1”である。従って、タッピングビットSTUFFの直前の符号列の復号終了位置の次のビットが“0”であるか否かだけを判定して誤り検出を行なってもよいし、あるいは、最初の“0”と後続のいくつかの“1”のみから誤り検出を行なってもよい。このようにすると、誤り検出精度は若干低下するものの、復号化に要する処理量が削減される。このように全てのタッピングビットSTUFFが特定のビットないしは複数ビットからなる特定のビットパターンから始まる符号表を用いる場合には、復号化処理の簡略化を図ることができる。

さらに、図30(a)(b)の符号表に示されるタッピングビットSTUFFは、“1”のビットを多く含んでおり、図5のように“0”が多く含まれる

同期符号およびその一部分とのハミング距離が離れているため、擬似同期が発生する確率が低いという利点がある。具体的には、図30(a)の符号表ではスタフピングビットSTUFFの最初のビットのみが全て“0”で、それ以外のビットは全て“0”であるため、オール“0”である同期符号およびその一部分とのハミング距離は(スタフピングビットSTUFFの長さ-1)となる。また、図30(b)の符号表ではスタフピングビットSTUFFの最初と最後のビットのみが“0”で、それ以外のビットは全て“1”であるため、同期符号およびその一部分とのハミング距離は(スタフピングビットSTUFFの長さ-2)となる。このようにスタフピングビットSTUFFと同期符号およびその一部分とのハミング距離を所定値以上、例えば(スタフピングビットSTUFFの長さ-2)以上に選ぶことにより、符号列に誤りが混入しても擬似同期符号が発生しにくい。

この効果について、図32を用いて説明する。図32(a-0)、(b-0)はそれぞれ通常のスタフピングビット(全てのビットが“0”)および図30(a)の符号表に示すスタフピングビットSTUFFを用いたときの符号列の例を示したものであり、(a-1)、(b-1)はそれぞれ(a-0)、(b-0)に1ビットの誤りが混入したときの例を示したものである。図32(a-1)から分かるように、全てのビットが“0”である通常のスタフピングビットでは1ビットの誤りが混入しただけで、(a-1)の破線で示すように同期符号と同一のビットパターンが生じるため、擬似同期が発生してしまう。これに対し、図30(a)の符号表に示すスタフピングビットSTUFFは、図32(b-2)に示すように誤りが混入しても同期符号と同一のパターンにならないため、擬似同期が発生しない。

このように本実施形態によるスタフピングビットは、符号列の誤りを容易に検出できると共に、符号列に誤りが混入しても擬似同期符号が発生しにくく、強

い誤り耐性を持つという利点がある。

また、本実施形態によるスタッフィングビットは逆方向から一意に復号が可能であり、その開始位置すなわちスタッフィングビットSTUFFの直前の符号列の終了位置を特定できるため、情報符号列が順方向からも逆方向からも復号可能な符号後の場合、図31の矢印3104で示すようにSTUFFの直前の符号列を逆方向から復号することができる。

上述の実施形態において、スタッフィングビットSTUFFは、以下のように定めてもよい。

(1) 同期符号に図5に示したようにsync__0__len ビットの“0”が含まれる場合、スタッフィングビットSTUFFの全てないしは少なくとも同期符号挿入位置の全ビットを“1”とすることにより、同期符号の“0”の部分とスタッフィングビットSTUFFのハミング距離を離すことができる。従って、このようにすることにより、スタッフィングビットSTUFFに誤りが混入して擬似同期が発生する確率を減少させることができる。

(2) スタッフィングビットSTUFFは、その長さを表す符号語としてもよい。復号化装置において、スタッフィングビットSTUFF以外の符号列の復号が終了した地点より、STUFFの長さを判定すると共に、STUFFの復号を行なってSTUFFの長さ情報を復号する。この場合、もし両者が一致しなかった場合には符号列中に誤りが混入したと判定することができる。

また、スタッフィングビットSTUFFの符号語は、その長さを2進数表現したものでもよい。例えば、STUFFが5ビットの場合、“5”を2進数表現した“00101”としてもよい。あるいは、2進数表現したものの“1”の補数や“2”の補数をとった値をスタッフィングビットSTUFFの符号語としてもよく、それによりSTUFF中の“0”のビットの数が減少するため、(1)で述べたのと同様に擬似同期の発生を抑えることができる。

(3) 順方向だけでなく、逆方向からも復号可能な符号語を用いて符号化を行なった場合、復号装置においてフレームの終了地点からスタッフィングビットSTUFFを逆方向に復号して、その開始地点(STUFFと他の符号語との境界地点)を知る必要がある。そのような場合には、例えばSTUFFを“01111111”のように、1ビットないしは複数ビットの“0”で始まり残りが“1”の符号語として定めればよい。これにより、STUFFを逆方向から復号して“0”がある位置を探せば、その地点をSTUFFの開始地点と一意に判定することができる。また、この例ではスタッフィングビットSTUFFの最初の部分以外のビットは“1”であり、(1)で述べたと同様に擬似同期が発生する確率が減少する。

(4) スタッフィングビットSTUFFは出力符号列の一部または全てのビットの誤り訂正/検出符号の検査ビットや、パリティチェックビット等としてもよい。これにより、出力符号列中に混入したビット誤りの訂正/検出を行なうことが可能である。

以上の例のように、スタッフィングビットSTUFFを予め定められた規則に従って生成することにより、復号装置において入力符号列中のスタッフィングビットSTUFFをその生成規則と照合し、もしその生成規則に反すると判定された場合には、入力符号列中に誤りが混入したと判定することが可能である。これにより、動画像復号化装置において再生画像に大きな劣化が生じないように処理を行なって、入力符号列中に誤りが混入したときの再生画像の品質を向上させることができる。

さらに、上記実施形態において同期符号挿入間隔sync__periodは、以下のよう

に定めてもよい。

(1) 誤り訂正/検出符号を用いる場合、同期符号挿入間隔sync__periodは復号化装置で同期検出を行なう最小のビット数、すなわち同期符号の長さと誤り訂正

／検出符号の検査ビットの最大値の和以上にとれば十分である。フレームの最後のスタッフィングビットSTUFFのビット数の平均値は $\text{sync_period}/2$ であるので、 sync_period をこの同期検出可能な最小ビットにとることにより、スタッフィングビットSTUFFのビット数が削減され、符号化効率が向上する。

(2) 誤り訂正／検出符号を用いない場合、同期符号挿入間隔 sync_period は復号化装置で同期検出を行なう最小のビット数、すなわち同期符号の長さ以上あれば十分である。フレームの最後のスタッフィングビットSTUFFのビット数の平均は $\text{sync_period}/2$ であるので、 sync_period をこの同期検出可能な最小ビットにとることにより、スタッフィングビットSTUFFのビット数が削減され、符号化効率が向上する。

(3) 図15、図16、図17および図27に示されるように、フレーム長情報POINTERを用いる場合には、同期符号挿入間隔 sync_period を同期符号の長さより短くしてもよい。これにより、スタッフィングビットSTUFFのビット数が削減され、符号化効率が向上する。

(4) 伝送路や蓄積媒体において定められた間隔のパケットやセルに区切って伝送／蓄積を行なう場合、同期符号挿入間隔 sync_period をパケットやセルの間隔に合わせるか、その間隔の約数になるようにしてもよい。これにより、パケットやセルの先頭は必ず同期符号挿入位置になるため、パケットロスやセルロスによりパケットやセルが起きた場合にも同期符号を検出することが可能である。

(5) 同期符号挿入間隔 sync_period は、1フレームの最小必要ビット数より短くすることが好ましい。これにより、スタッフィングビットSTUFFのビット数が削減され、符号化効率が向上する。

(第6実施形態)

次に、本発明に係る第6実施形態について説明する。

図35は、本実施形態における動画像符号化装置の出力符号列の一例を示す図

である。この出力符号列では、誤りによる同期符号の誤検出確率を低くするために、前述の実施例で述べたようなビット挿入処理を行なっている。また、予め定められた位置、あるいは同期符号から相対的にみて予め定められた位置に、ヘッダ情報等の情報が入っている。

図35の(a)はビット挿入処理を行なう前の符号列であり、(b)はビット挿入処理を行なった後の出力符号列である。図中、斜線の部分3201, 3202, 3261, 3262が定められた位置(同期符号から相対的に見て定められた位置)に入れる情報であり、白抜きの矢印3211, 3212がその情報を入れる位置である。図35(b)中の符号列の情報3261, 3262が(a)の符号列の情報3201, 3202にそれぞれ相当し、場合によっては符号列(a)から符号列(b)への変換に際して、これらの情報に対しても変換(すなわち、情報3201から情報3261への変換および情報3202から情報3262への変換)が施されることがある。

図35(b)中の3203は、ビット挿入処理によって挿入されたビットである。このビット挿入処理により、挿入ビットに後続するビット列は後方にずれてしまうため、定められた位置に入れる情報が定位置に入るように、その情報直前の符号列の一部を後方に移す処理を行なう。例えば、情報3201の一つ前の同期符号3205からの挿入ビットの数の合計をNs1とすれば、情報3201の直前の図35(a)中の記号3221に示すNs1ビットを情報3201の直後の図35(b)中の記号3231の部分に移せばよい。

情報3201, 3202中に、符号列中の特定の位置を示すポインタ等の情報が含まれている場合には、これを変換する処理を行なってもよい。具体的には、例えば情報3201中に矢印3241で示す位置を示す情報が含まれている場合、この位置よりも挿入ビットの数Ns1だけ後ろの矢印3251で示す位置を指すように、情報3261中の位置を示す情報を変換するようにする。

請 求 の 範 囲

1. 入力される符号列を情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化する符号化手段と、

出力符号列中の予め定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、前記情報ビットを該出力符号列中の任意の位置に配置し、前記検査ビットを該出力符号列中の前記同期符号挿入位置以外の位置に配置して該出力符号列を組み立てる符号列組立手段と、

を備えることを特徴とする符号化装置。

2. 情報ビットと検出ビットからなる誤り訂正／検出符号に符号化された符号列から予め定められた複数の同期符号挿入位置で同期符号の検出を行なう同期符号検出手段と、

前記符号列を分解して誤り訂正／検出符号の情報ビットおよび前記同期符号挿入位置以外の位置に配置された該誤り訂正／検出符号の検査ビットを抽出する符号列分解手段と、

前記符号列分解手段により抽出された情報ビットおよび検査ビットを受けて前記誤り訂正／検出符号を復号化する復号化手段と、

を備えることを特徴とする復号化装置。

3. 出力符号列中の予め定められた複数の同期符号挿入位置にある同期符号以外の入力符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段と、

前記符号列変換手段により変換が行なわれた後の符号列を情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化する符号化手段と、

前記出力符号列中の予め定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、前記情報ビットを該出力符号列中の任意の位置に配置し、



前記検査ビットを該出力符号列中の前記同期符号挿入位置以外の位置に配置して該出力符号列を組み立てる符号列組立手段と、

を備えることを特徴とする符号化装置。

4. 情報ビットと検査ビットからなる誤り訂正／検出符号に符号化されかつ同期符号が挿入された符号列から、予め定められた同期符号挿入位置で同期符号の検出を行なう同期符号検出手段と、

前記符号列を分解して誤り訂正／検出符号の情報ビットおよび前記同期符号挿入位置以外の位置に配置された該誤り訂正／検出符号の検査ビットを抽出する符号列分解手段と、

前記符号列分解手段により抽出された情報ビットおよび検査ビットを受けて前記誤り訂正／検出符号を復号化する復号化手段と、

前記復号化手段により復号化された符号列中の同期符号とのハミング距離が予め定められた値以上となるように変換された前記同期符号挿入位置に存在する該同期符号以外の符号列を元の符号列に変換する符号列変換手段と、

を備えることを特徴とする復号化装置。

5. 入力される符号列を誤り訂正／検出符号に符号化する符号化手段と、

前記符号列中に同期符号を挿入する同期符号挿入手段と、

前記符号列の前記同期符号の直前の誤り訂正／検出符号に符号化されるべき情報のビット数を判定する判定手段と、を備えると共に、

前記符号化手段は、前記同期符号の直前の誤り訂正／検出符号を前記判定手段により判定されたビット数に基づいて適応的に縮退させた縮退符号により符号化することを特徴とする符号化装置。

6. 誤り訂正／検出符号に符号化され、かつ、同期符号が挿入された符号列を復号化する復号化手段と、

前記符号列中の同期符号を検出する同期符号検出手段と、



前記符号列中の同期符号検出手段により検出された同期符号の直前の誤り訂正／検出符号に符号化された情報のビット数を判定する判定手段と、を備えると共に、

前記復号化手段は、前記判定手段の判定結果に基づいて前記同期符号の直前の誤り訂正／検出符号が縮退符号であるか否かを識別して復号化を行なうことを特徴とする復号化装置。

7. 入力される複数種類の情報を含む符号列を誤り訂正／検出符号に符号化する符号化手段と、

前記誤り訂正／検出符号の種類を前記符号列中の情報の種類に応じて切り替える切り替え手段と、

を備えることを特徴とする符号化装置。

8. 情報の種類に応じて異なる種類の誤り訂正／検出符号に符号化された符号列を復号化して元の情報を生成する復号化手段と、

前記復号化手段で生成された情報の種類から前記誤り訂正／検出符号の種類を判定して前記復号化手段に通知する手段と、

を備えることを特徴とする復号化装置。

9. 出力符号中の予め定められた複数の同期符号挿入位置とその前後の予め定められたビット数の区間にある同期符号以外の入力符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段と、

前記符号列変換手段により変換が行なわれた後の符号列を情報ビットおよび検査ビットからなる誤り訂正／検出符号に符号化する符号化手段と、

前記出力符号列中の予め定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、前記情報ビットを該出力符号列中の任意の位置に配置し、前記検査ビットを該符号列中の前記同期符号挿入位置以外の位置に配置して該出力符号列を組み立てる符号列組立手段と、



を備えることを特徴とする符号化装置。

10. 情報ビットと検査ビットからなる誤り訂正／検出符号に符号化されかつ同期符号が挿入された符号列から、予め定められた同期符号挿入位置とその前後の予め定められたビット数の区間で同期符号の検出を行なう同期符号検出手段と、

前記符号列を分解して誤り訂正／検出符号の情報ビットおよび前記同期符号挿入位置以外の位置に配置された該誤り訂正／検出符号の検査ビットを抽出する符号列分解手段と、

前記符号列分解手段により抽出された情報ビットおよび検査ビットを受けて前記誤り訂正／検出符号を復号化する復号化手段と、

前記復号化手段により復号化された符号列中の同期符号とのハミング距離が予め定められた値以上となるように変換された前記同期符号挿入位置およびその前後の予め定められたビット数区間に存在する該同期符号以外の符号列を元の符号列に変換する符号列変換手段と、

を備えることを特徴とする復号化装置。

11. 入力信号を圧縮符号化して得られた複数種類の圧縮符号を多重化して多重化符号列を生成する多重化手段と、

前記多重化符号列を入力として出力符号列を組み立てる符号列組立手段と、を備えると共に、

前記符号列組立手段は、前記出力符号列中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入することを特徴とする符号化装置。

12. 入力信号を圧縮符号化して得られた複数種類の圧縮符号を多重化して多重化符号列を生成する多重化手段と、

前記多重化符号列を入力として出力符号列を組み立てる符号列組立手段と、を備え、



前記符号列組立手段は、前記出力符号列中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、該出力符号列中にスタッフィングビットを適宜挿入することを特徴とする符号化装置。

13. 入力信号を圧縮符号化して得られた複数種類の圧縮符号を多重化して多重化符号列を生成する多重化手段と、

前記多重化符号列を入力として出力符号列を組み立てる符号列組立手段と、を備え、

前記符号列組立手段は、前記出力符号列中の予め周期的に複数の同期符号挿入位置を定めると共に、前記多重化符号列の境界を指し示す情報を挿入し、前記複数の同期符号挿入位置のいずれかに同期符号を挿入することを特徴とする符号化装置。

14. 前記多重化手段は、入力画像信号のフレームまたは該フレームの部分領域単位で圧縮符号化された圧縮符号を前記フレームまたは該フレームの部分領域単位で多重化し、

前記符号列組立手段は、前記多重化符号列の前記フレームまたは該フレームの部分領域単位で多重化された各多重化単位の終りの部分の直前または直後に位置する同期符号挿入位置に同期符号を挿入することを特徴とする請求項11ないし13のいずれか1項に記載の符号化装置。

15. 前記出力符号列中の前記同期符号挿入位置にある前記同期符号以外の符号列を該同期符号とのハミング距離が予め定められた値以上となるように変換する符号列変換手段をさらに有することを特徴とする請求項11ないし13のいずれか1項に記載の符号化装置。

16. 入力符号列中の同期符号を検出する同期符号検出手段と、

前記入力符号列から前記同期符号検出手段により検出された同期符号の位置を基準として逆多重化を行なって圧縮符号を生成する逆多重化手段と、



前記圧縮符号を復号化して再生信号を出力する復号化手段とを有し、

前記同期符号検出手段は、前記入力符号列中の予め周期的に定められた複数の同期符号挿入位置で前記同期符号を検出することを特徴とする復号化装置。

17. 前記入力符号列の前記同期符号挿入位置で前記同期符号とのハミング距離が予め定められた値以上となるように変換された該同期符号以外の符号列を元の符号列に変換する符号列変換手段をさらに備えることを特徴とする請求項16に記載の復号化装置。

18. 出力符号列中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、該出力符号列中に該出力符号列の逆方向から一意に復号可能なスタッフィングビットを適宜挿入する符号列組立手段を備えたことを特徴とする符号化装置。

19. 前記符号列組立手段は、前記出力符号列中の前記複数の同期符号挿入位置のいずれかの直前に前記スタッフィングビットを挿入することを特徴とする請求項18に記載の符号化装置。

20. 予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号が挿入されると共に、逆方向から一意に復号可能なスタッフィングビットが適宜挿入された入力符号列を復号化する復号化手段と、

前記復号化手段による前記スタッフィングビットの直前の符号列の復号終了位置と前記スタッフィングビットの開始位置とを比較することにより、前記入力符号列中の誤りを検出する誤り検出手段と、

を備えることを特徴とする復号化装置。

21. 前記スタッフィングビットは、前記入力符号列中の前記複数の同期符号挿入位置のいずれかの直前に挿入されていることを特徴とする請求項20に記載の復号化装置。

22. 前記復号化手段は、前記スタッフィングビットの直前の符号列を前記



スタッフィングビットの開始位置から逆方向に復号することを特徴とする請求項 19 に記載の復号化装置。

23. 出力符号列中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、該出力符号列中に該出力符号列の逆方向から一意に復号可能で、かつ前記同期符号およびその一部との間のハミング距離が予め定められた値以上であるスタッフィングビットを適宜挿入する符号列組立手段を備えたことを特徴とする符号化装置。

24. 前記スタッフィングビットを前記同期符号の直前に配置することを特徴とする請求項 18 または 23 に記載の符号化装置。

25. 予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号が挿入されると共に、逆方向から一意に復号可能で、かつ前記同期符号およびその一部との間のハミング距離が予め定められた値以上であるスタッフィングビットが適宜挿入された入力符号列を復号化する復号化手段と、

前記復号化手段による前記スタッフィングビットの直前の符号列の復号終了位置と前記スタッフィングビットの開始位置とを比較することにより、前記入力符号列中の誤りを検出する誤り検出手段と、

を備えることを特徴とする復号化装置。

26. 前記入力符号列は、前記スタッフィングビットが前記同期符号の直前に配置されていることを特徴とする請求項 20、22 または 25 のいずれか 1 項に記載の復号化装置。

27. 前記多重化装置は、前記圧縮符号を複数のレイヤ毎に多重化を行なって多重化符号列を生成し、

前記符号列組立手段は、前記多重化符号列を入力として前記複数のレイヤの少なくとも一部のレイヤに同期符号を付加して出力符号列を生成することを特徴とする請求項 11 ないし 13 のいずれか 1 項に記載の符号化装置。



28. 前記符号列組立手段は、前記同期符号の長さを一部または全てのレイヤで異ならせたことを特徴とする請求項27に記載の符号化装置。

29. 前記出力符号列中の前記同期符号が付加されたレイヤのうちの一部または全てのレイヤにおける前記同期符号以外の符号列を前記同期符号とのハミング距離が予め定められた値以上となるように変換する符号化変換手段をさらに備えることを特徴とする請求項27または28に記載の符号化装置。

30. 前記出力符号列中の前記同期符号が付加されたレイヤのうちの一部または全てのレイヤにおける前記同期符号以外の符号列を前記長さの異なる同期符号のうち最も長さが短い同期符号とのハミング距離が予め定められた値以上となるように変換する符号化変換手段をさらに備えることを特徴とする請求項29に記載の符号化装置。

31. 前記入力符号列は、複数のレイヤ毎に多重化され、かつ該複数のレイヤのうちの少なくとも一部のレイヤに同期信号が付加された多重化符号列からなることを特徴とする請求項16に記載の復号化装置。

32. 前記入力符号列中の前記同期符号が付加されたレイヤのうちの一部または全てのレイヤにおける長さの異なる同期符号のうち最も長さが短い同期符号とのハミング距離が予め定められた値以上となるように変換された前記同期符号以外の符号列を元の符号列に変換する符号列変換手段をさらに備えることを特徴とする請求項31に記載の復号化装置。



2/33

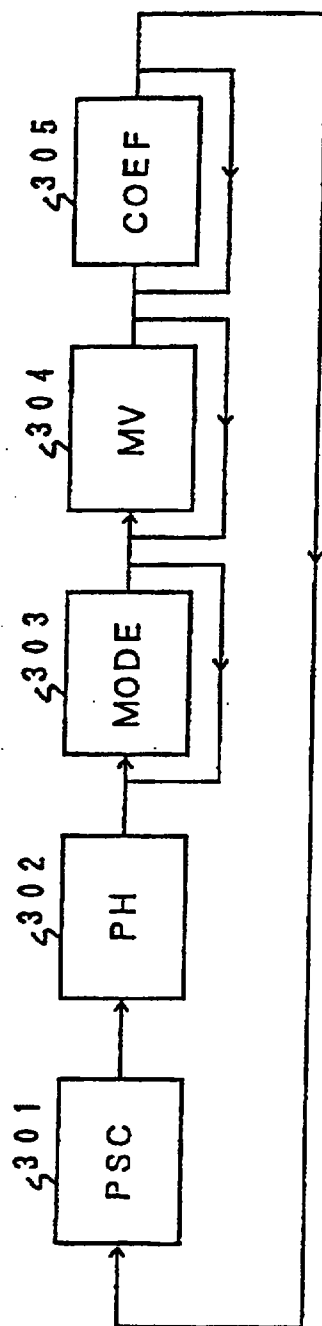


FIG. 2

3 / 33

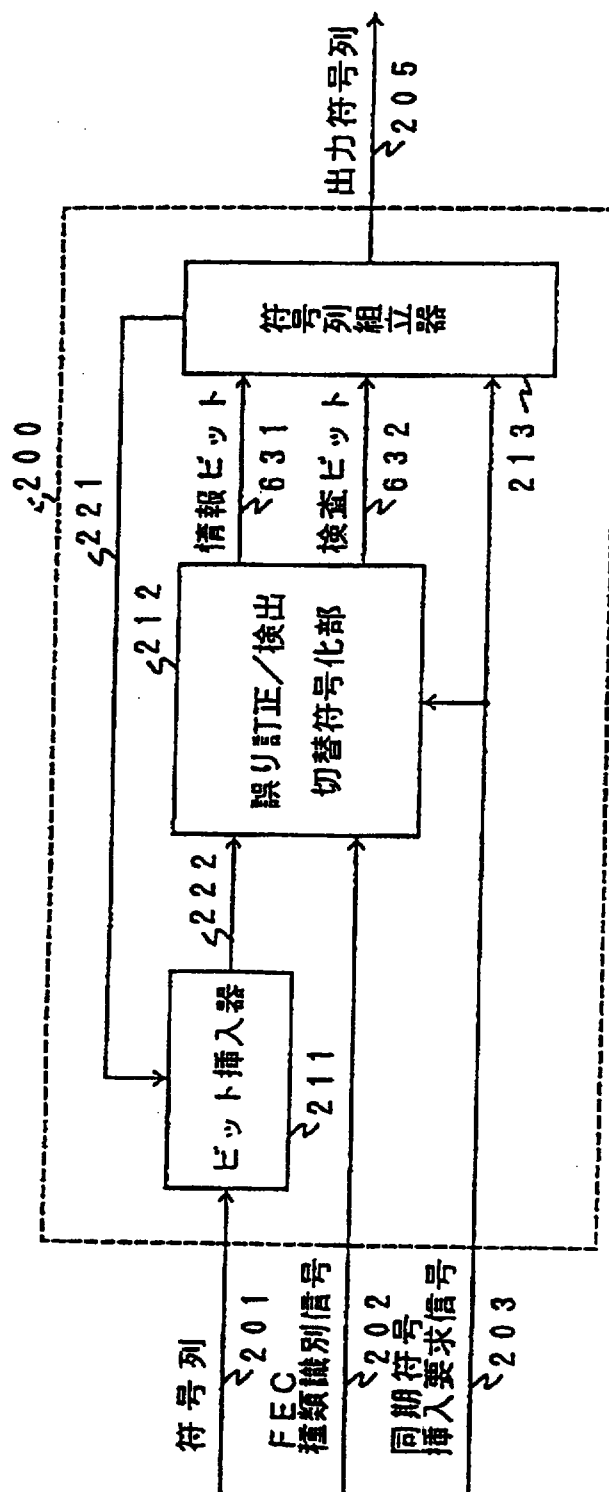


FIG. 3

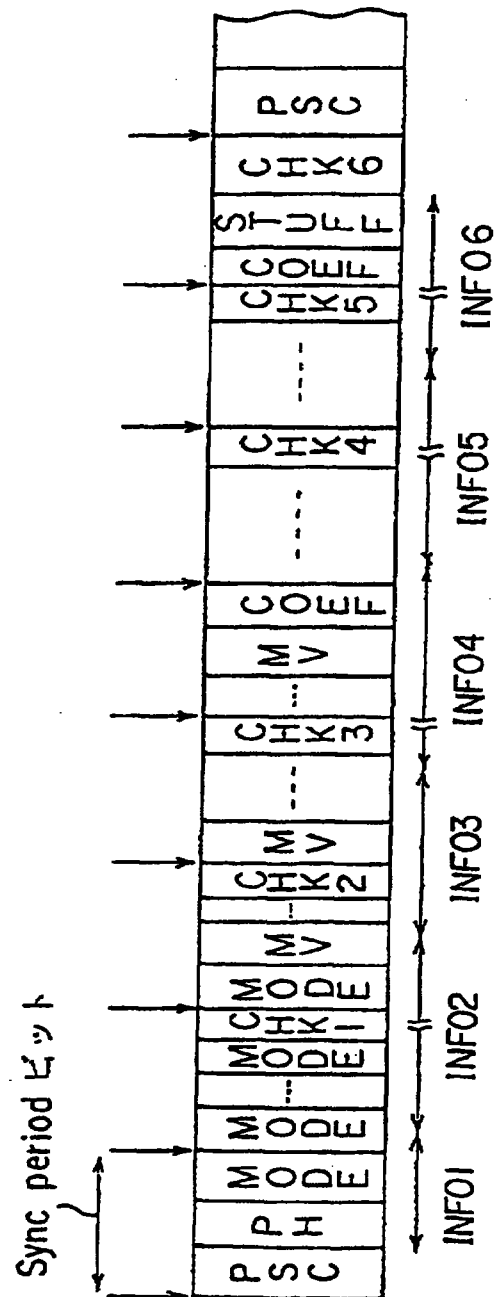


FIG. 4

5 / 33

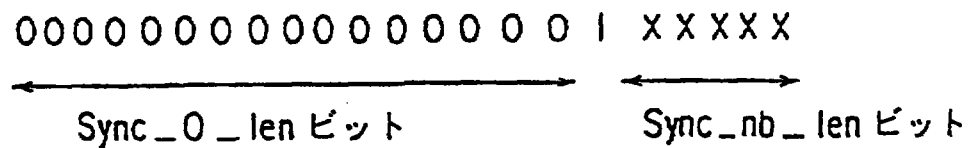


FIG. 5

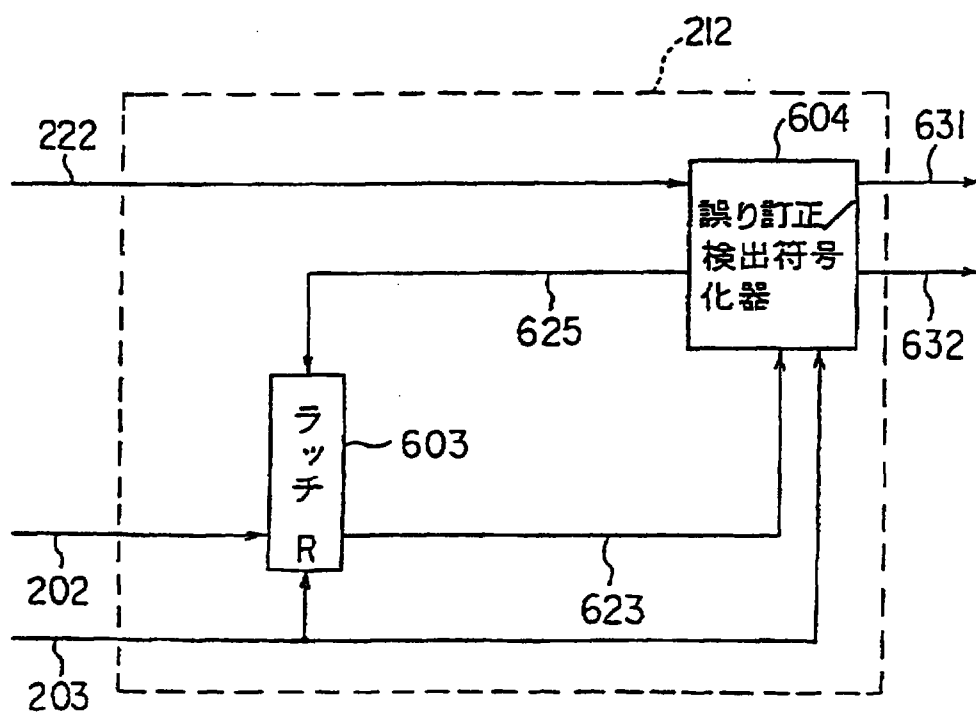


FIG. 6

6 / 33

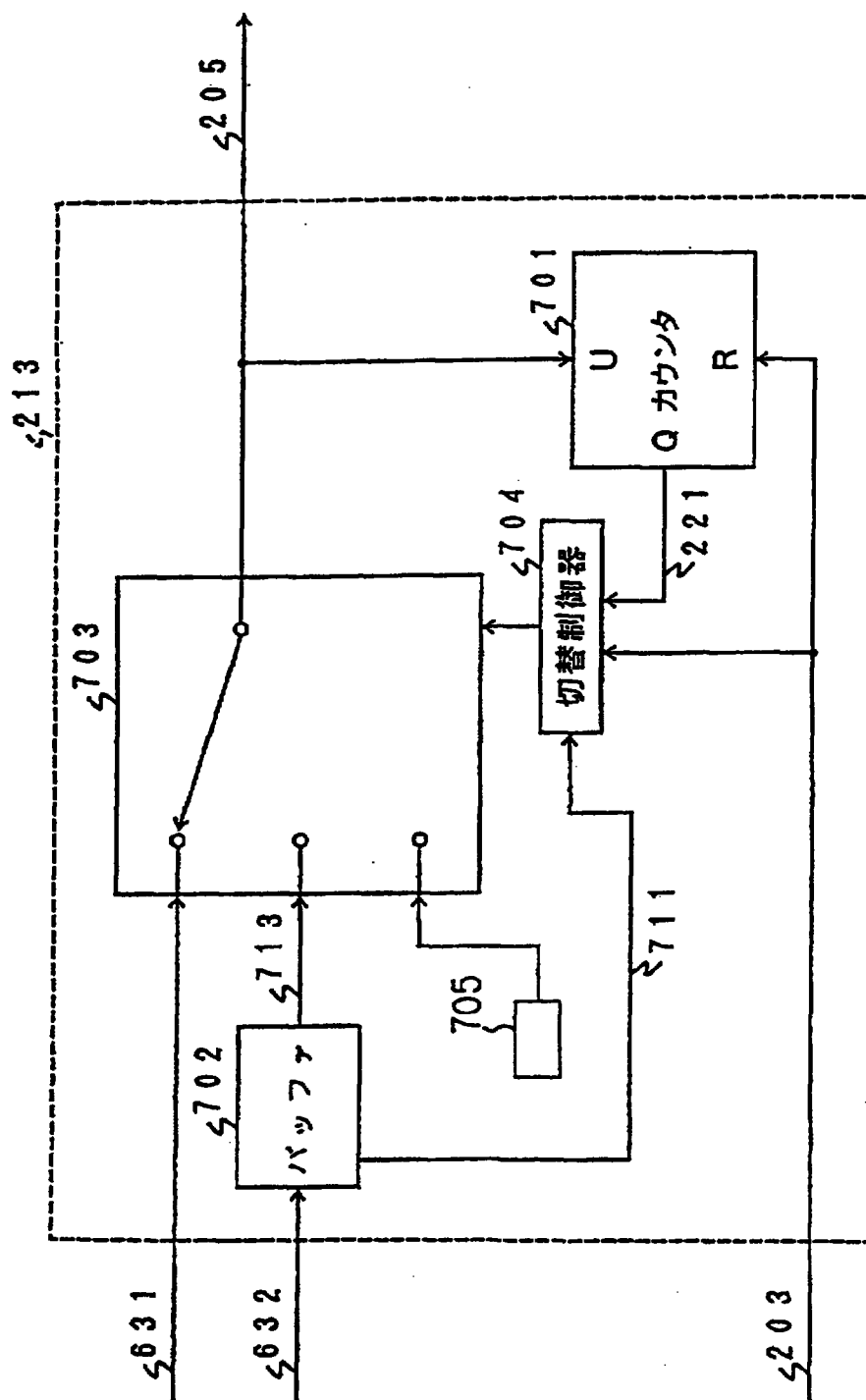


FIG. 7

7/33

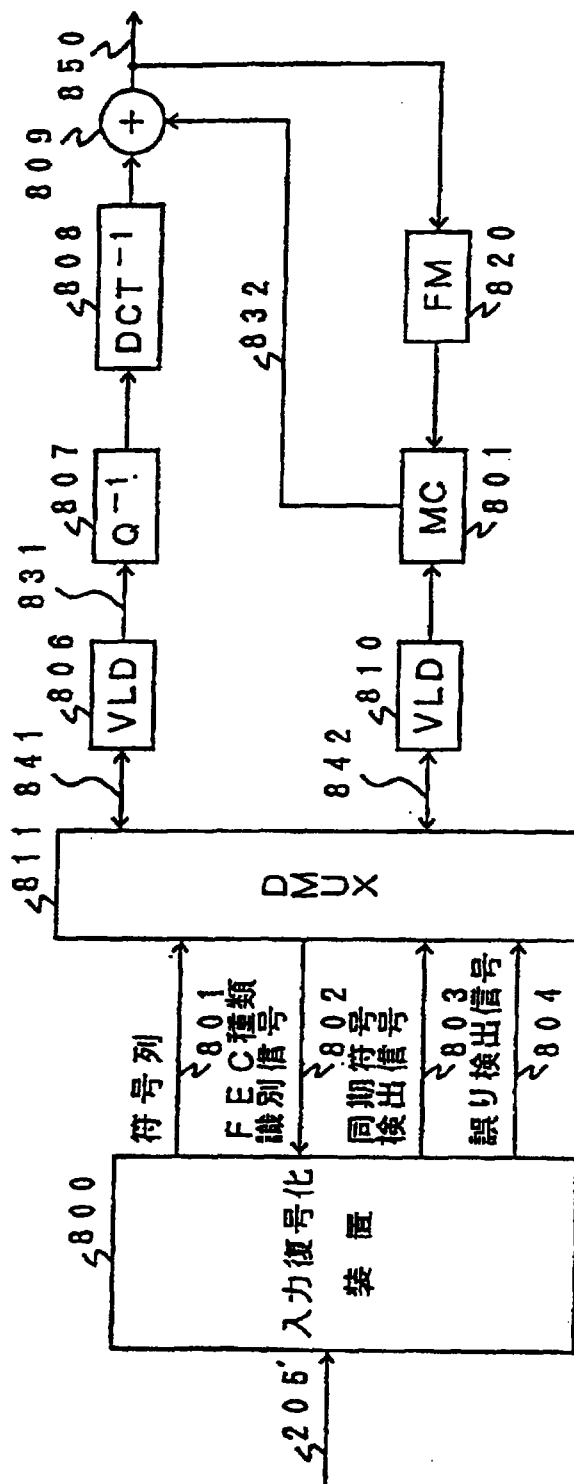


FIG. 8

8/33

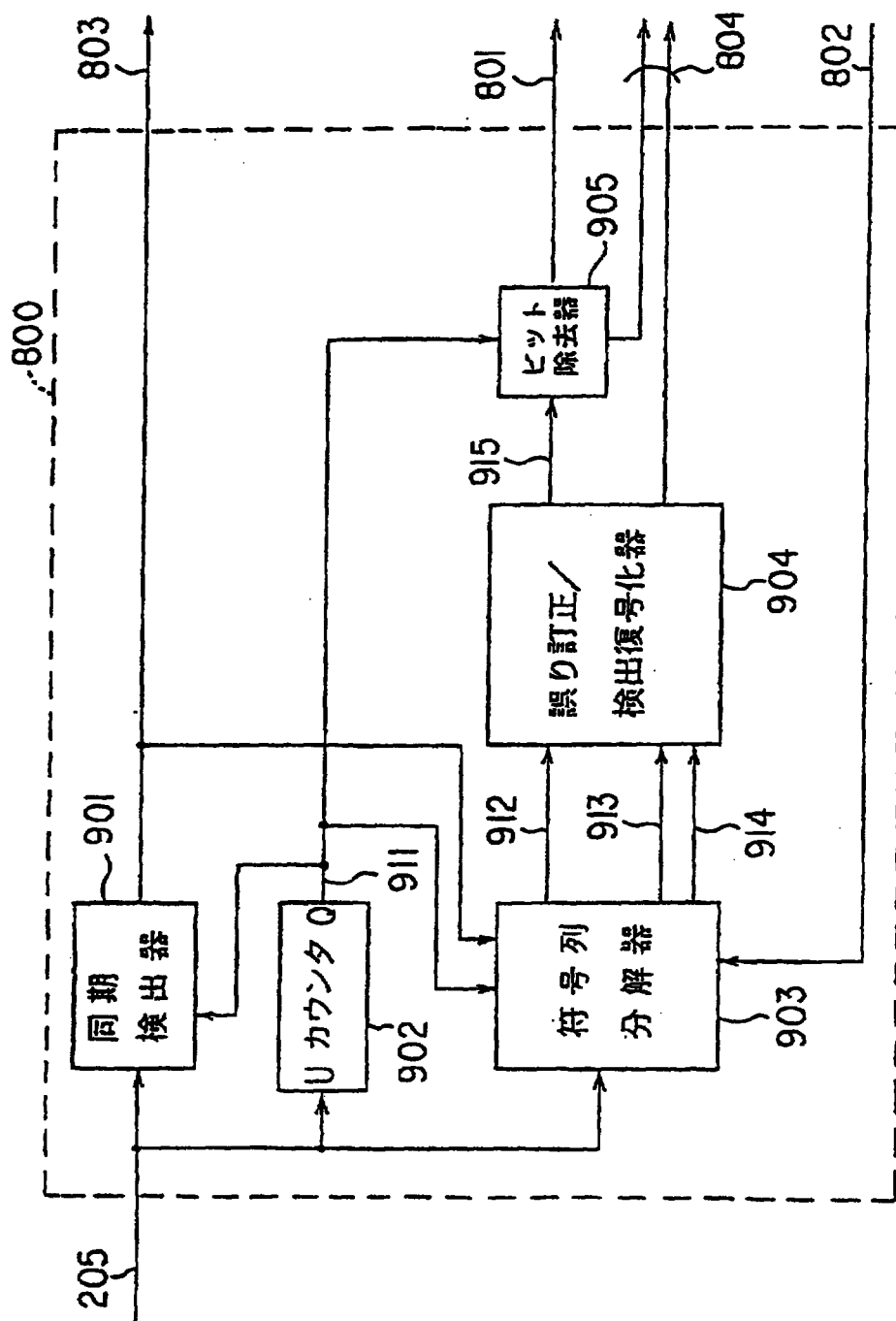


FIG. 9

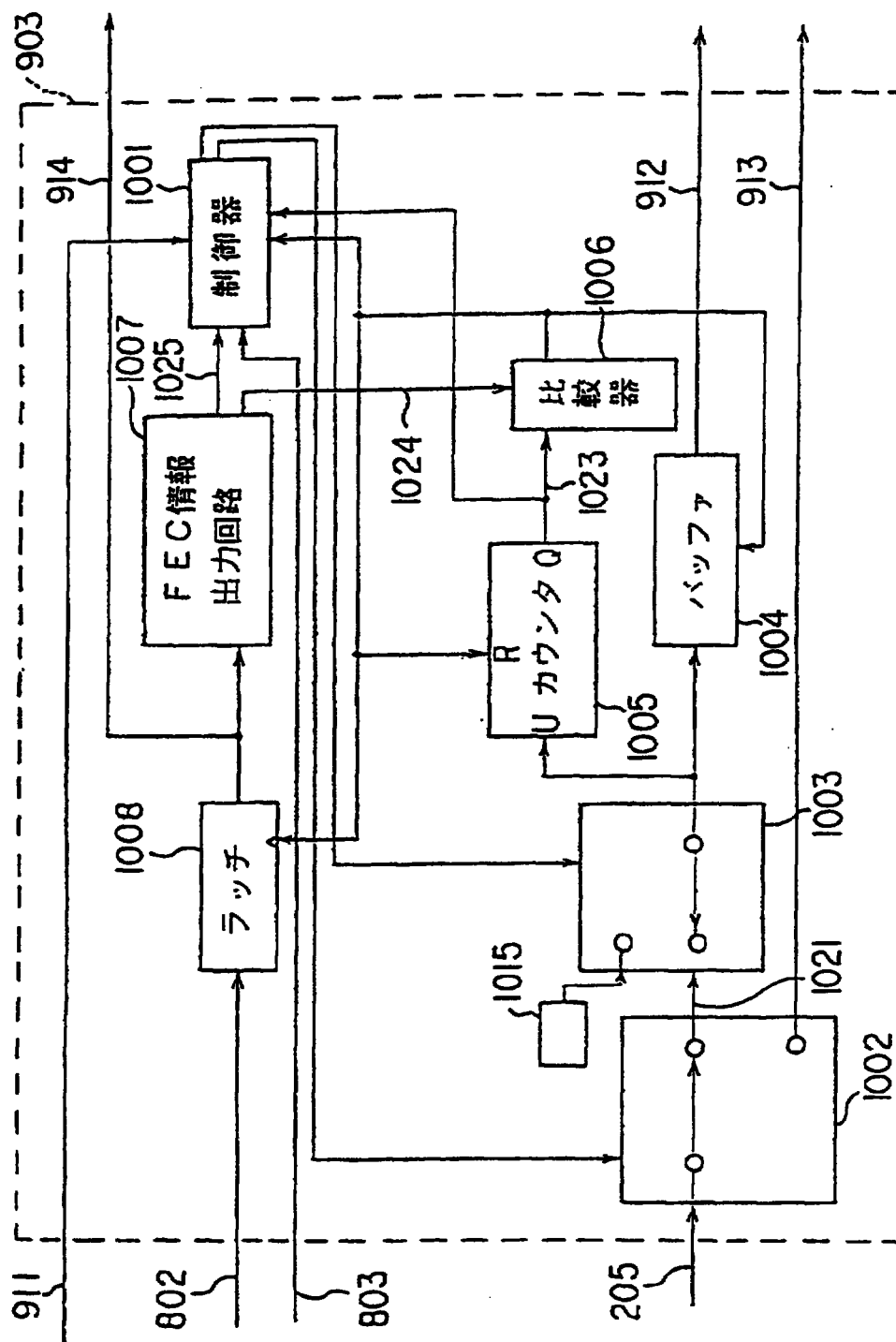


FIG. 10

10/33

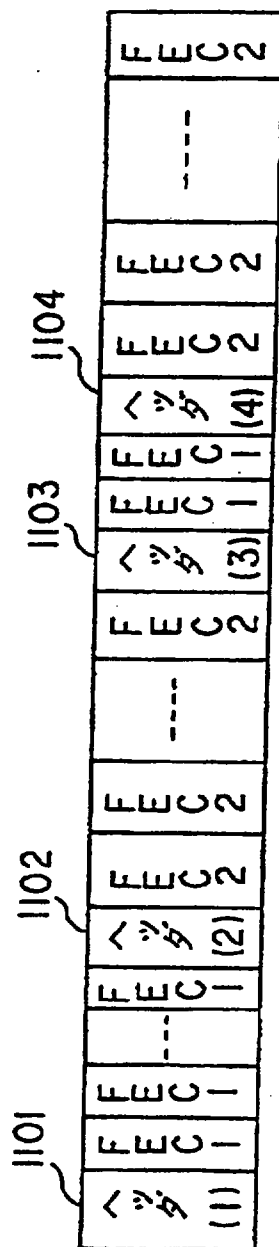


FIG. 11

11/33

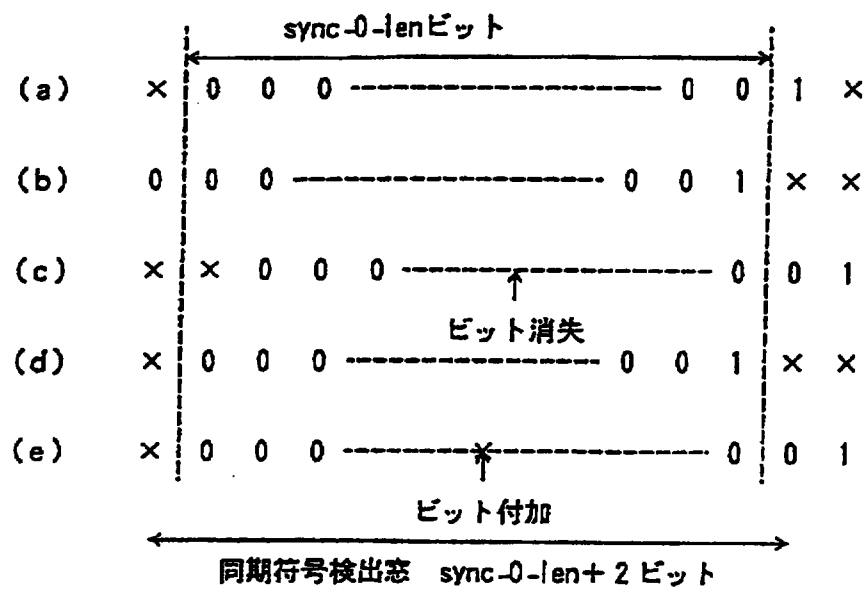


FIG. 12

12 / 33

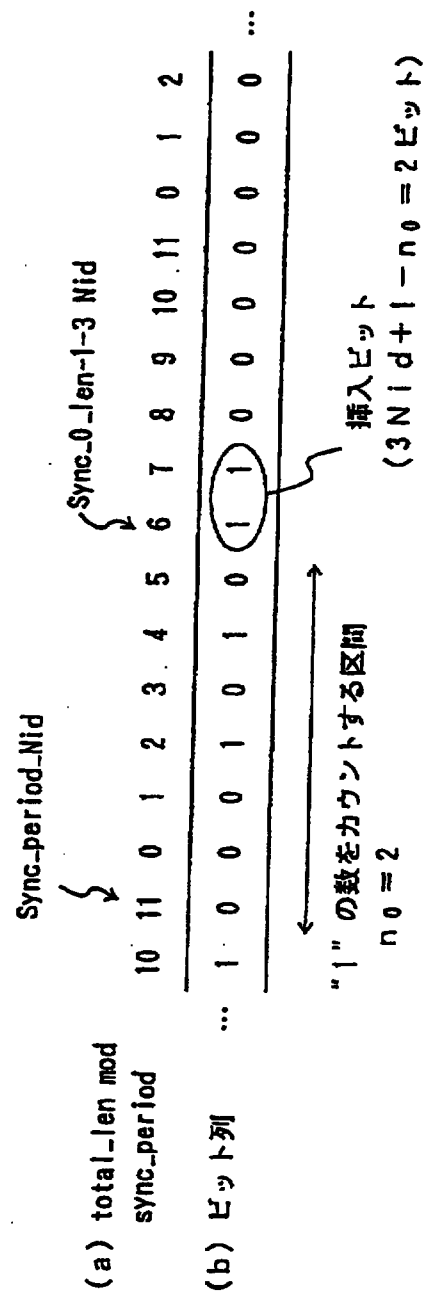


FIG. 13

14 / 33

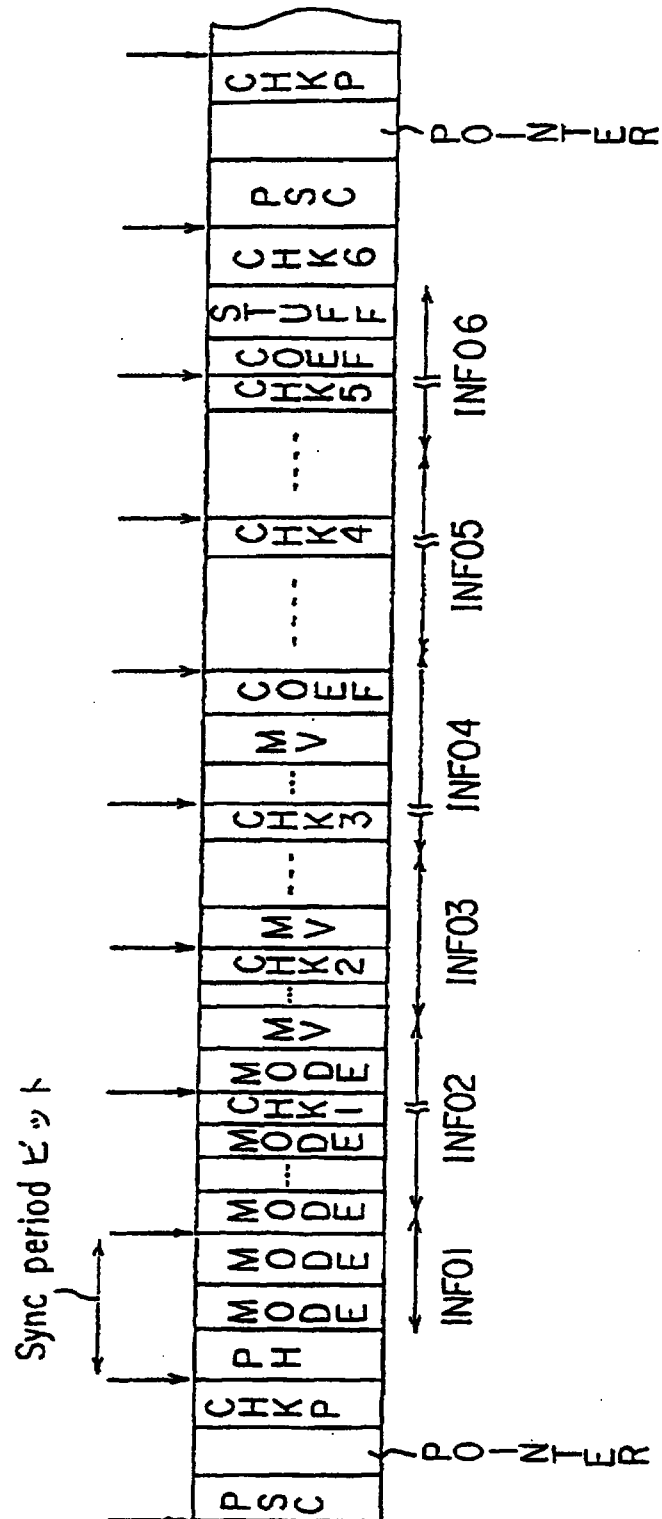


FIG. 15

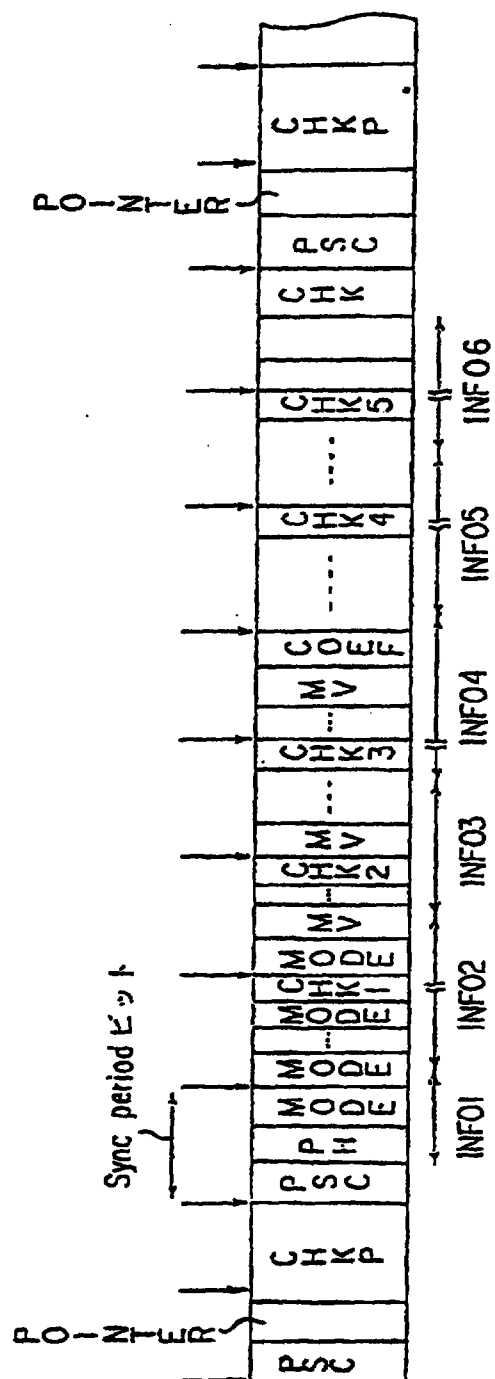


FIG. 16

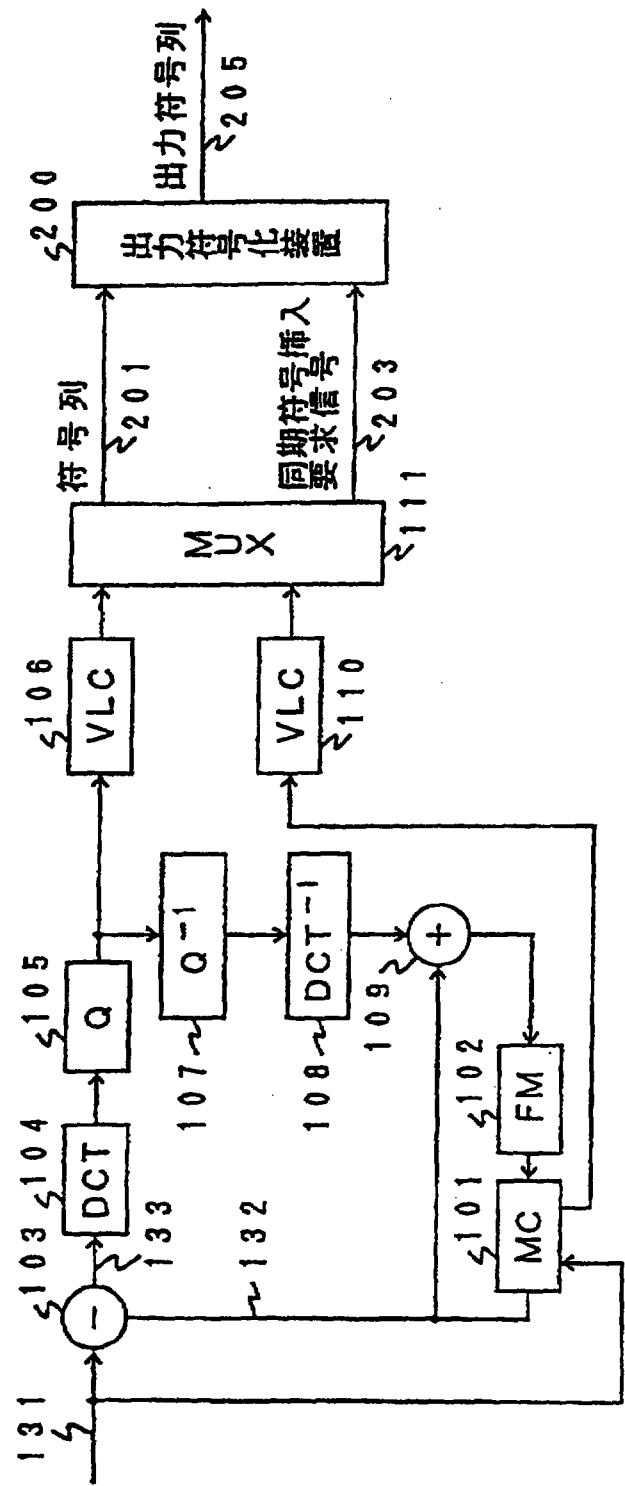


FIG. 18

18 / 33

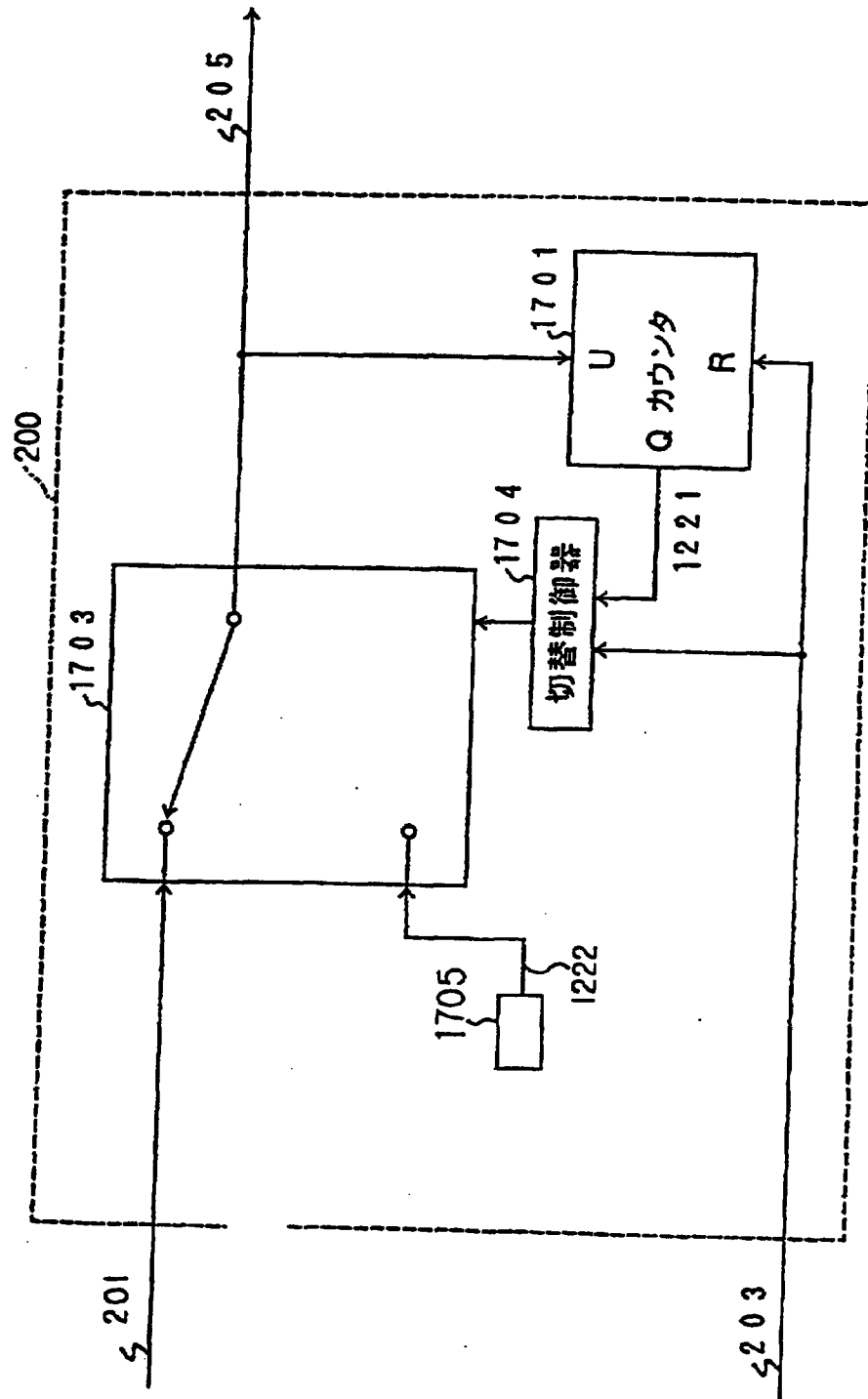


FIG. 19

19 / 33

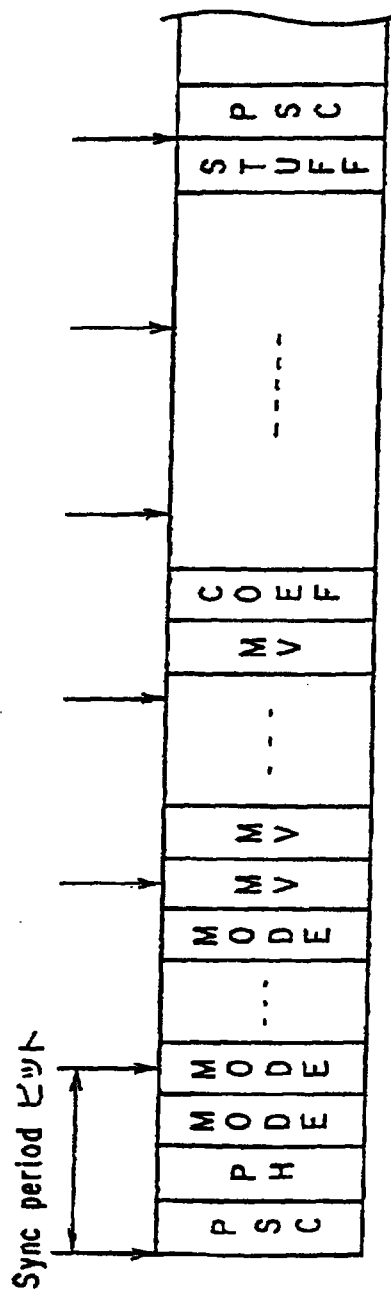


FIG. 20

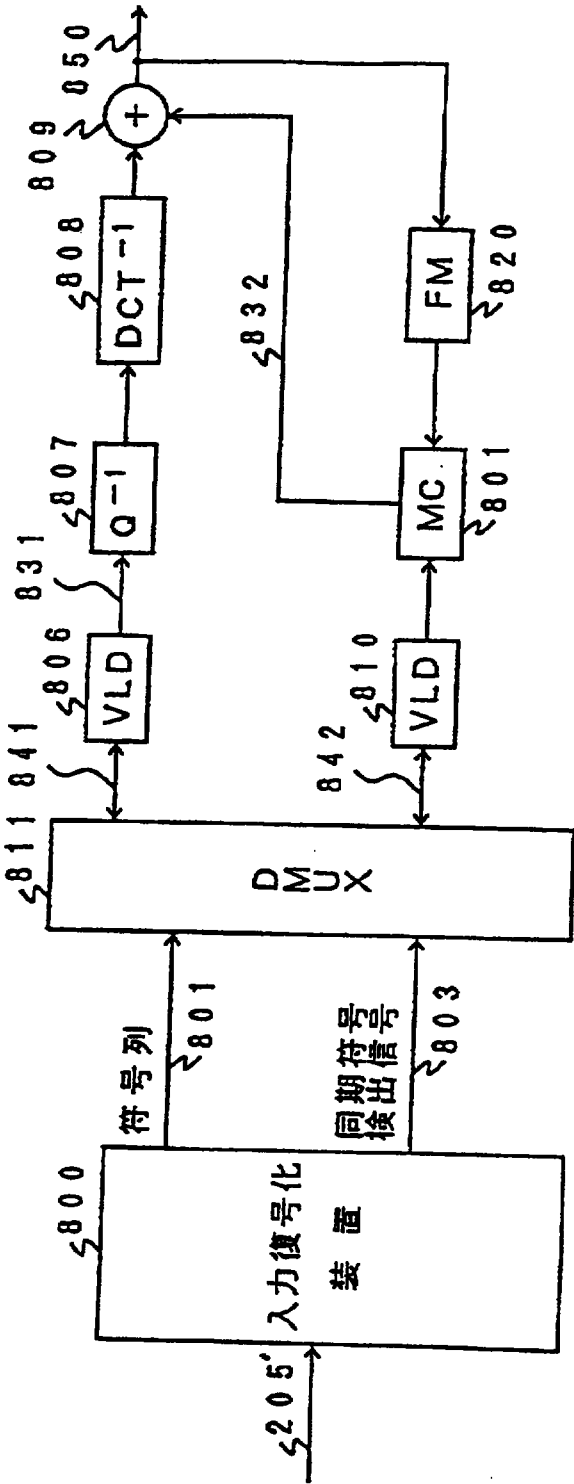


FIG. 21

21/33

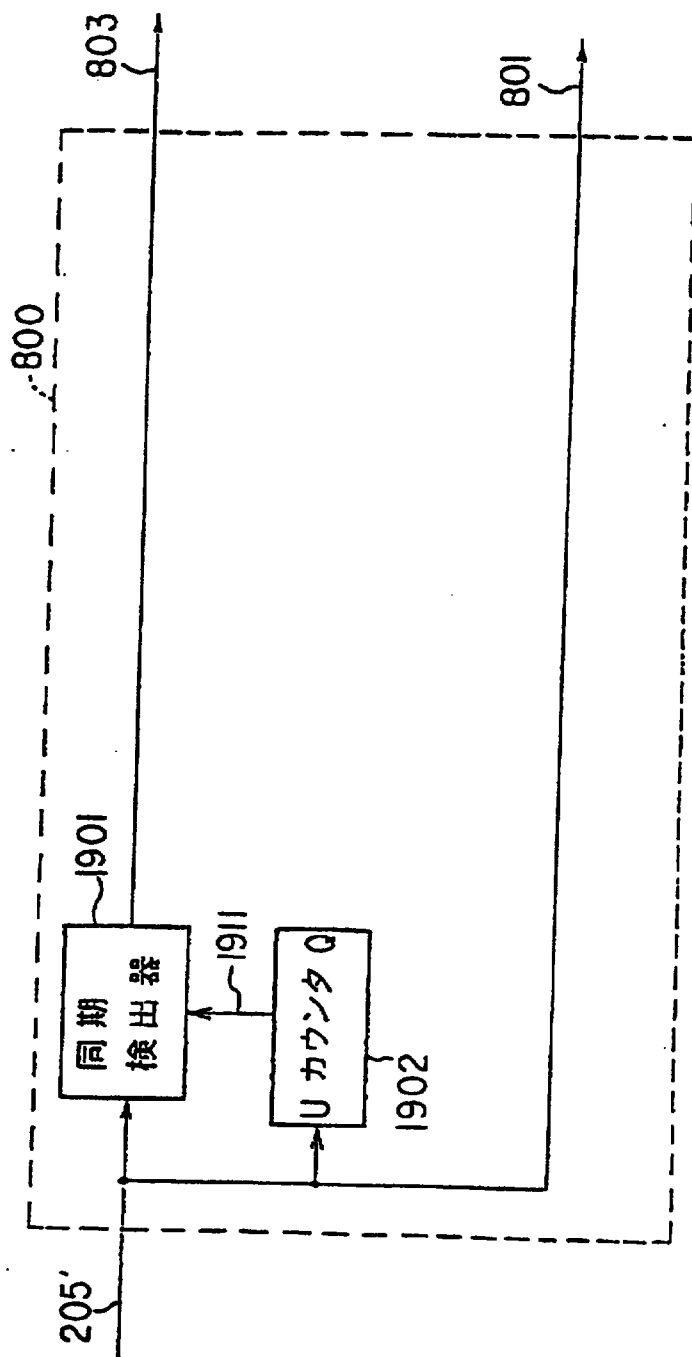


FIG. 22

22 / 33

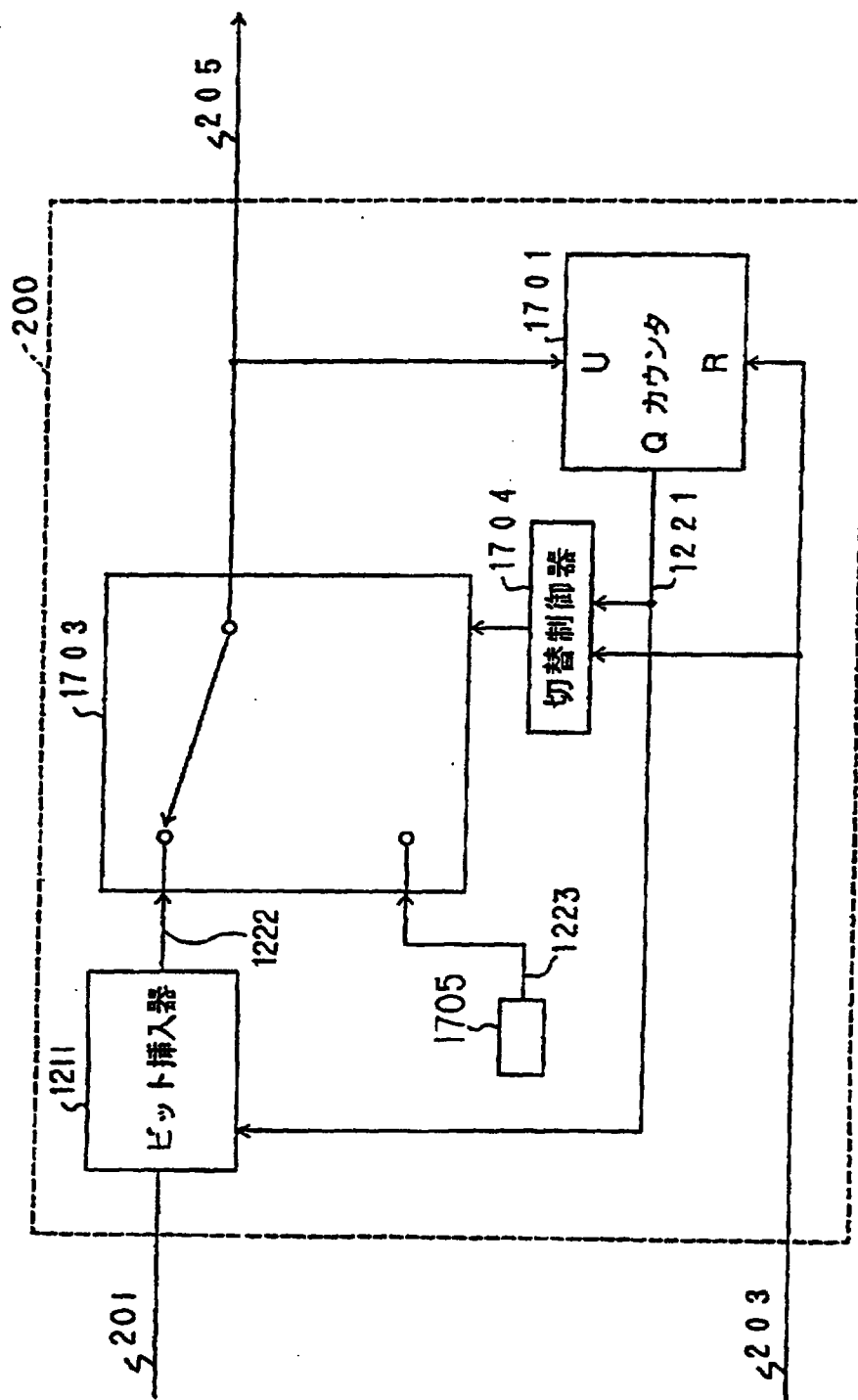


FIG. 23

23 / 33

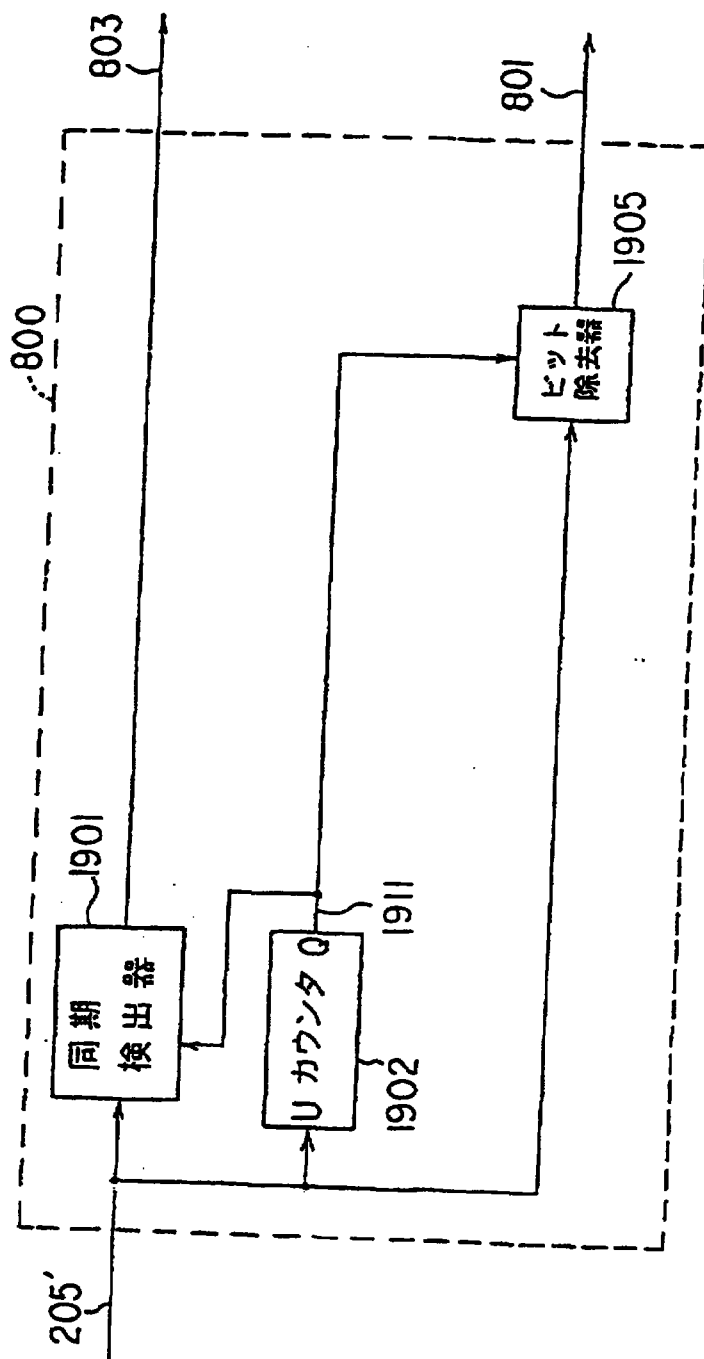


FIG. 24

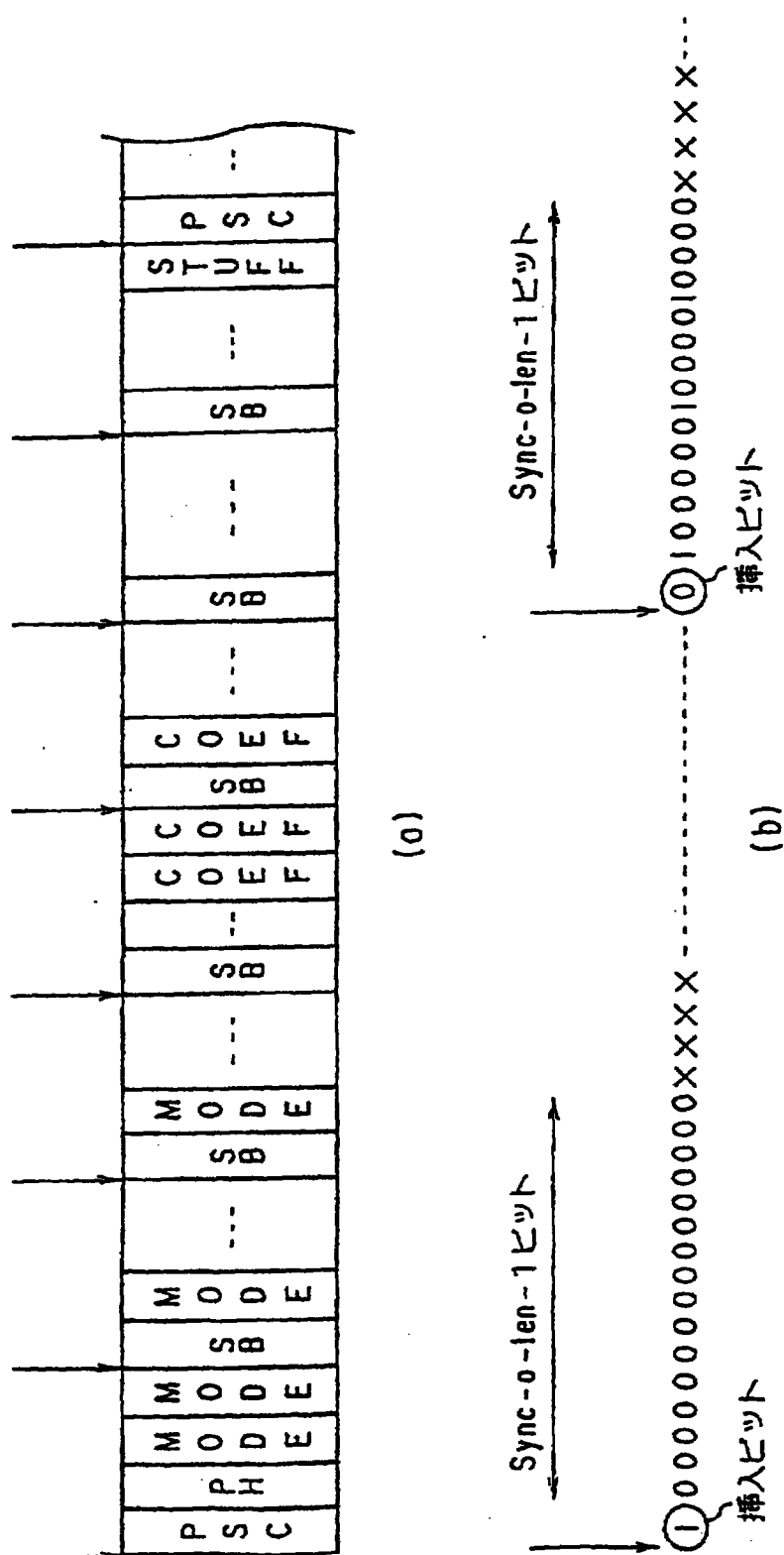


FIG. 25

25 / 33

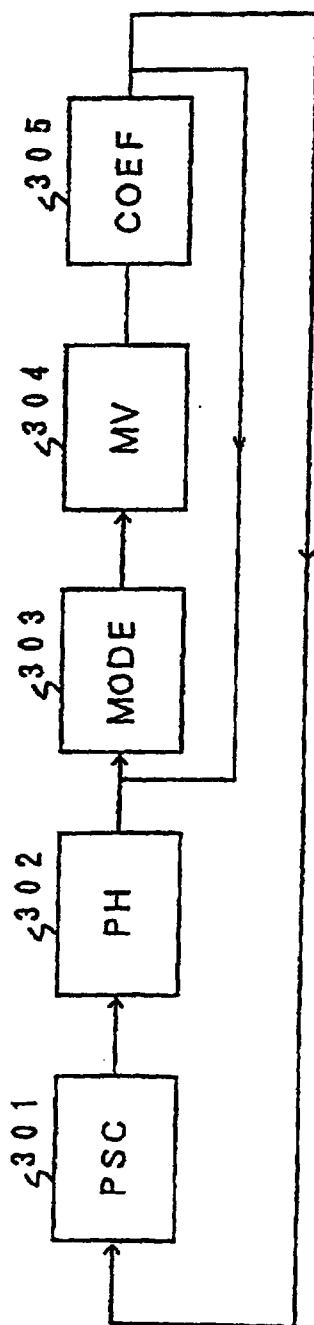


FIG. 26

26 / 33

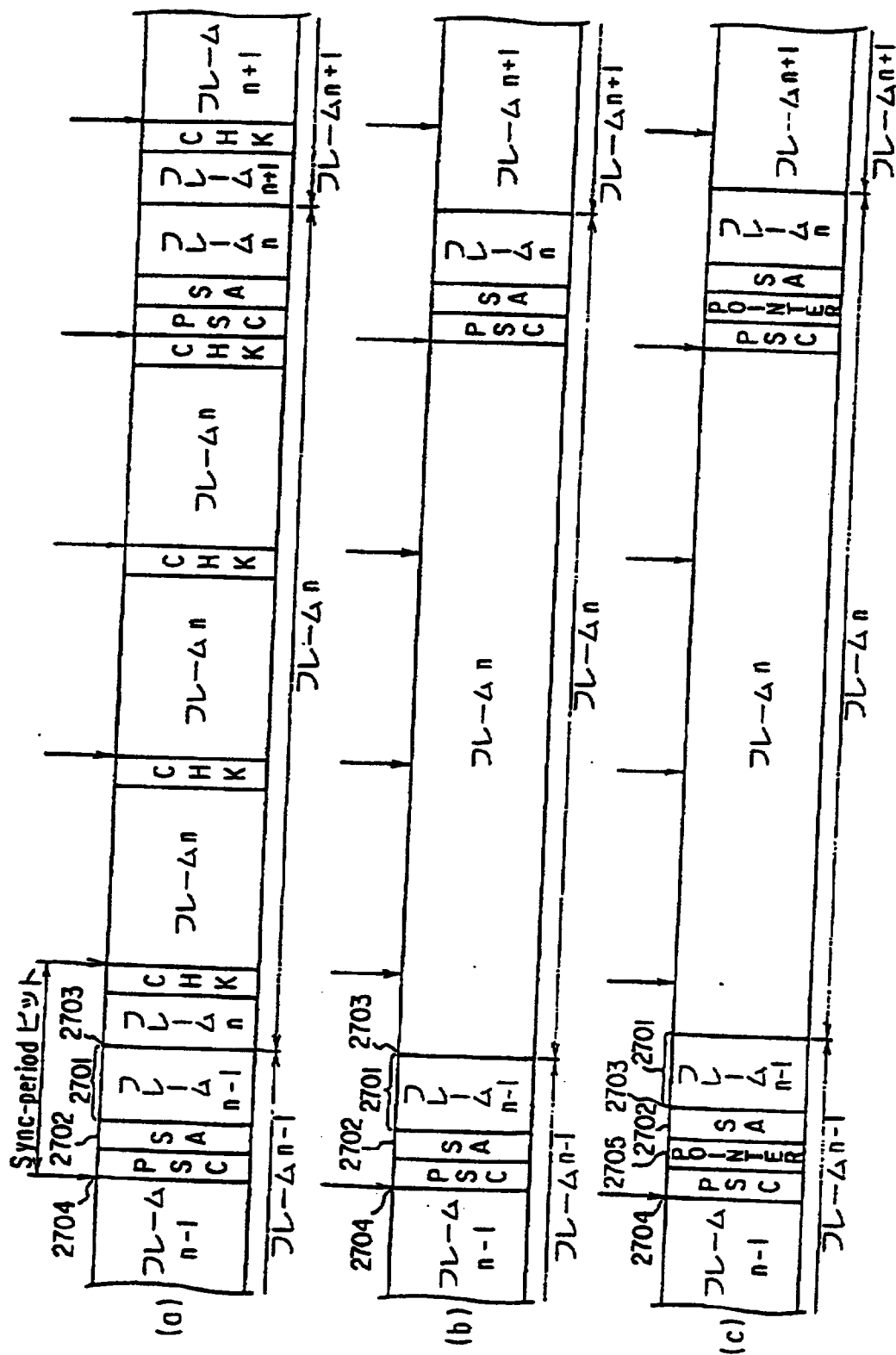
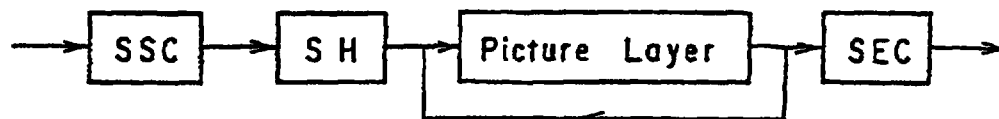
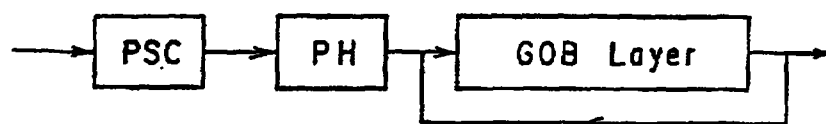


FIG. 27

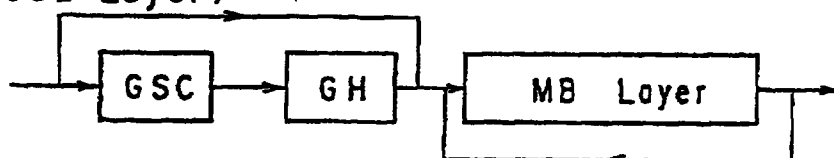
27 / 33

セッションレイヤ
(Session Layer)

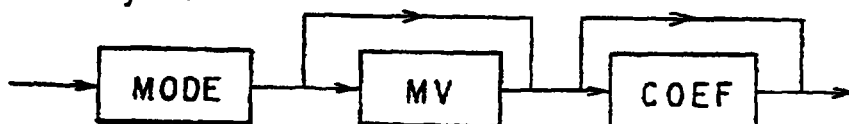
(a)

ピクチャレイヤ
(Picture Layer)

(b)

GOBレイヤ
(GOB Layer)

(c)

マクロブロックレイヤ
(MB Layer)

(d)

FIG. 28

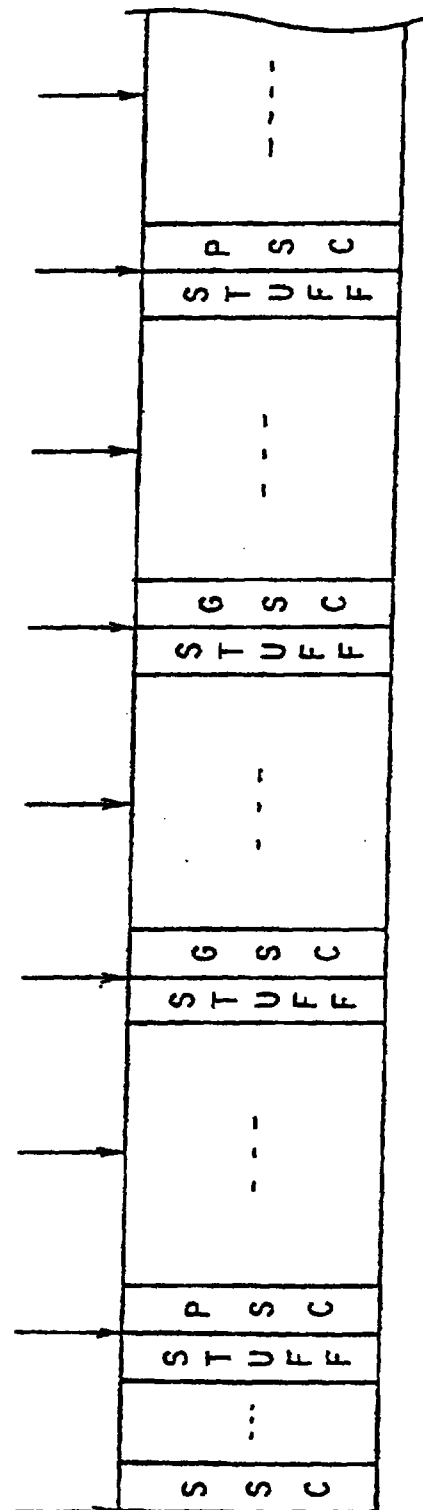


FIG. 29

29 / 33

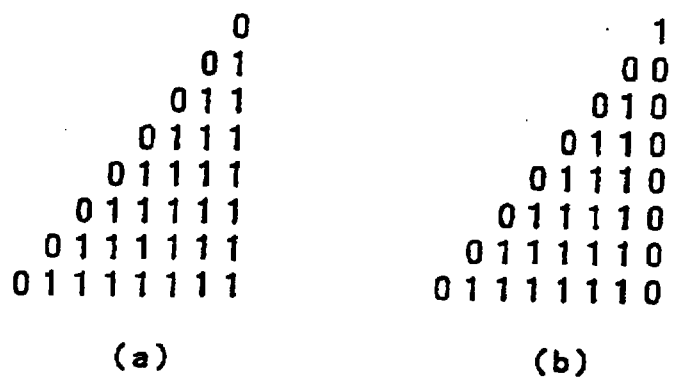


FIG. 30

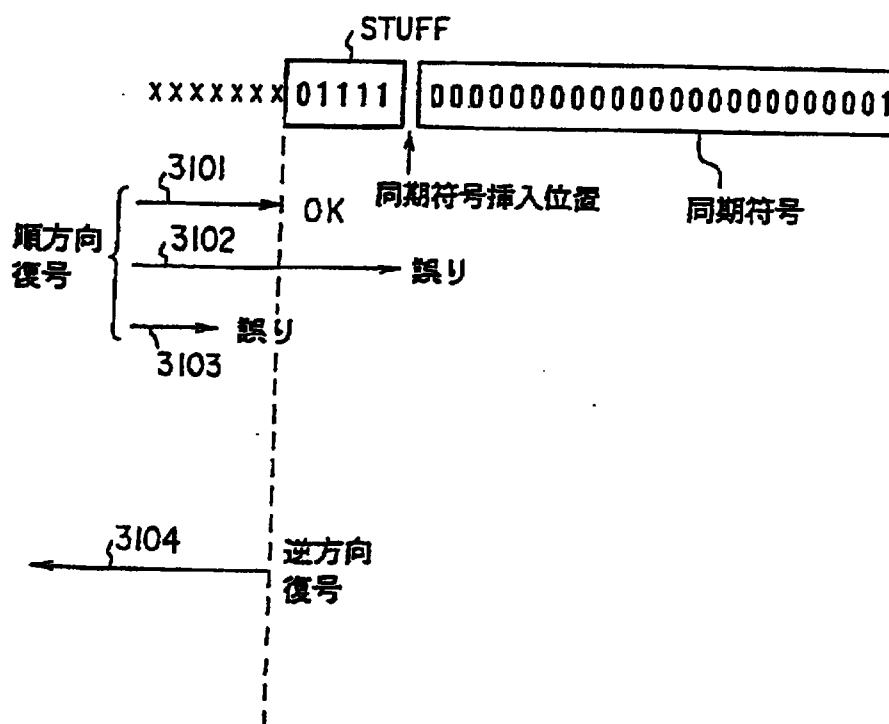


FIG. 31

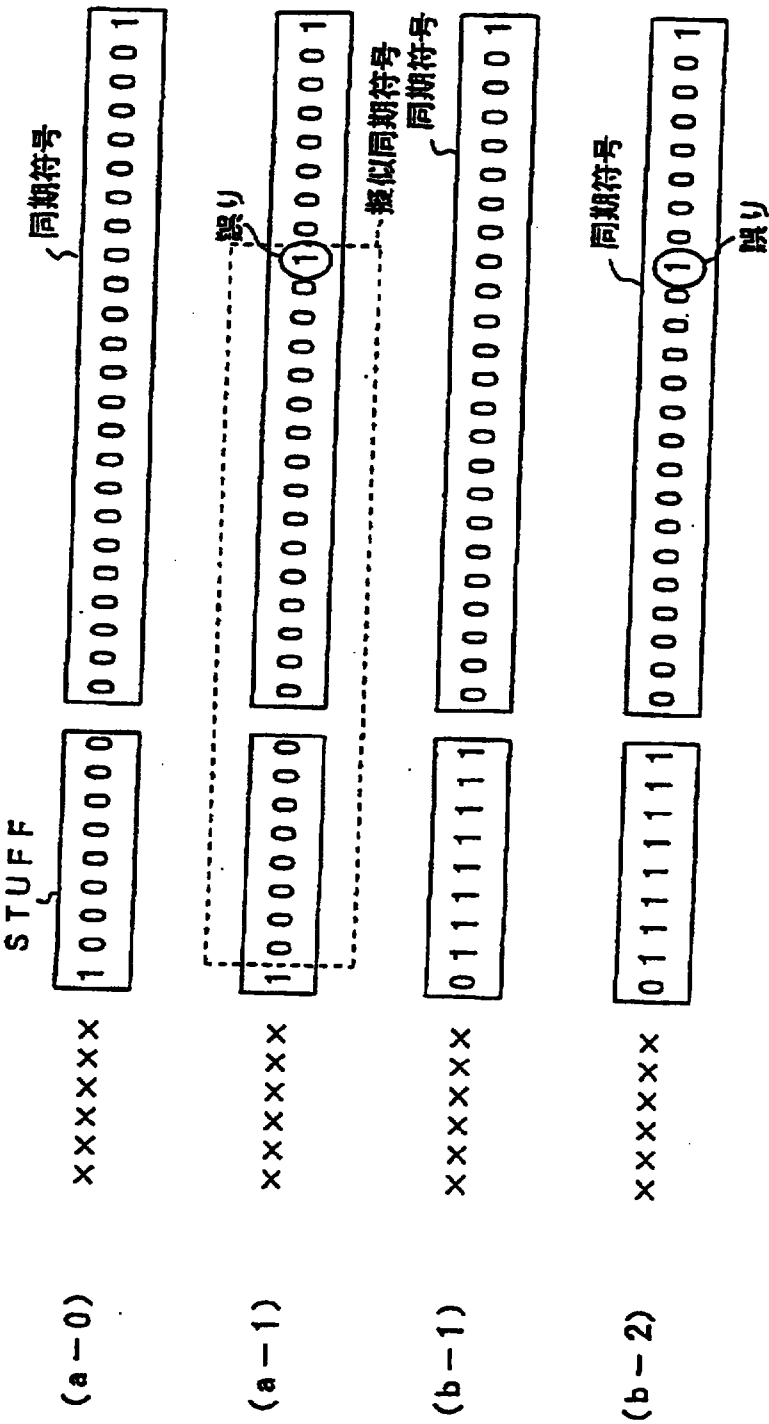


FIG. 32

31 / 33

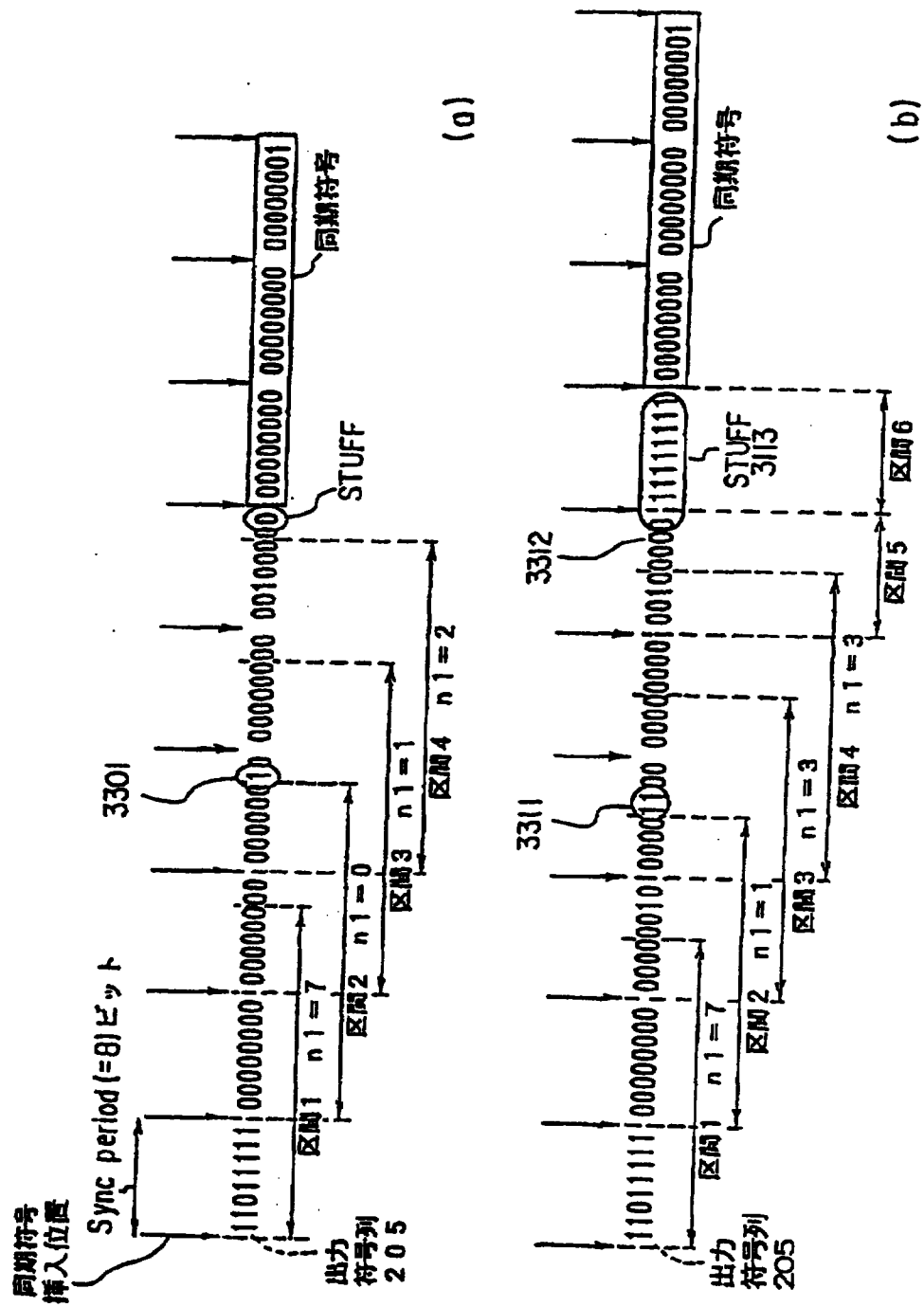


FIG. 33

32 / 33

S S C 00000000 00000000 00000001 00010101
 S E C 00000000 00000000 00000001 00101010
 P S C 00000000 00000000 00000001 00000000
 G S C 00000000 00000000 1

(a)

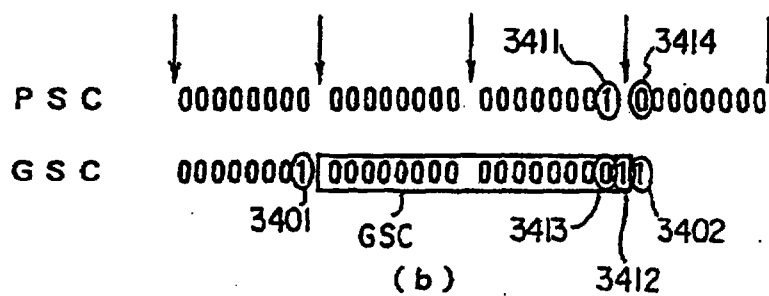


FIG. 34

33 / 33

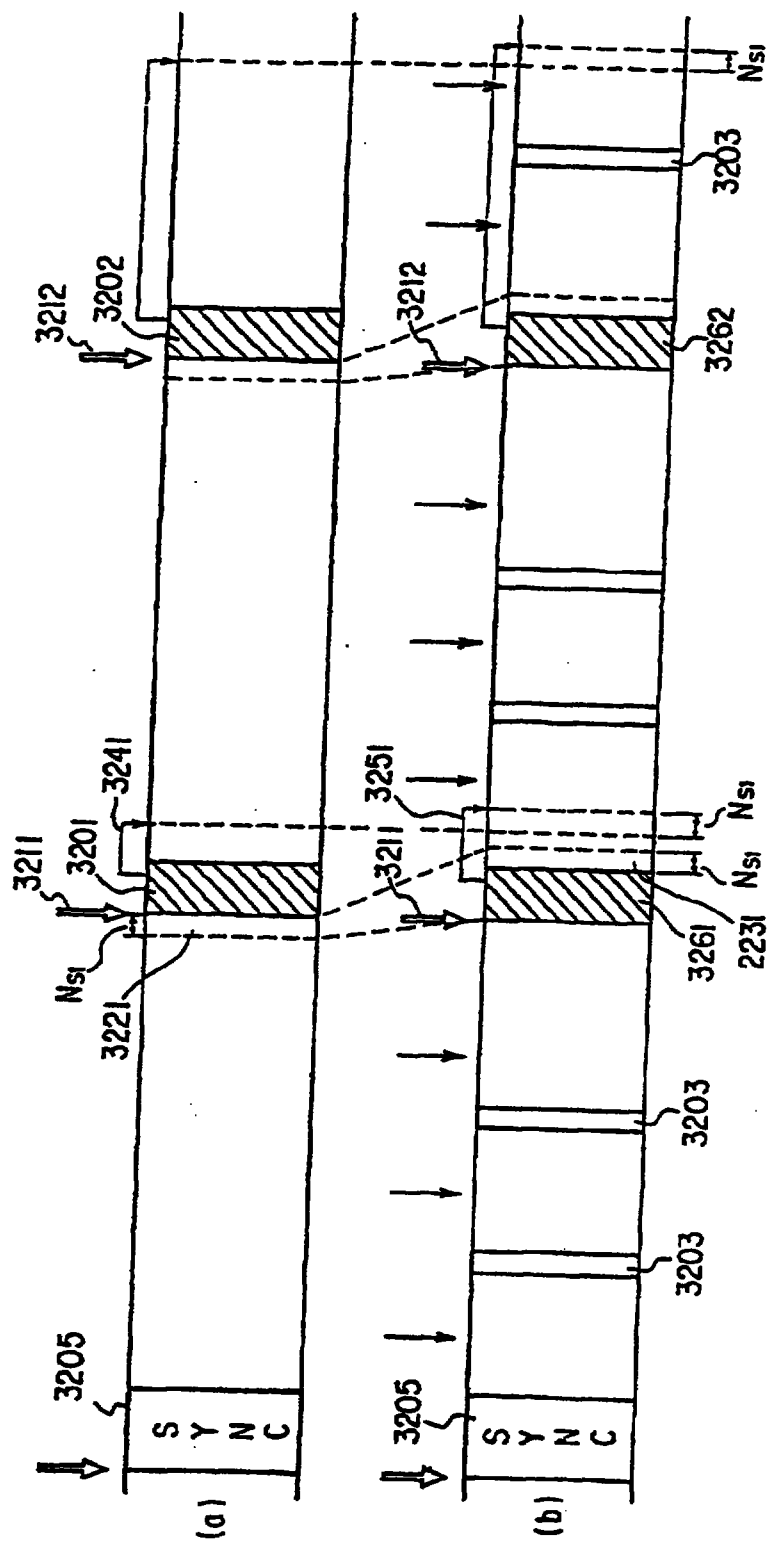


FIG. 35

SPECIFICATION
CODING SYSTEM AND DECODING SYSTEM

TECHNICAL FIELD

5 The present invention relates generally to a system
for transmitting and/or storing information via a medium
of a high error rate, such as a radio transmission line.
More specifically, the invention relates to a coding and/or
10 decoding system suited to carry out the error
correction/detection coding of a compressed code string
obtained by the high efficiency compression coding to
transmit and/or store the compressed code string.

BACKGROUND ART

15 For example, in a system for carrying out the
high-efficient compression coding of a picture and/or voice
information so as to have a small information content to
transmit the compression-coded picture and/or voice
information via a radio transmission line, such as a radio
20 visual telephone, a portable information terminal and a
digital television broadcast system, it is important how
to transmit the obtained code string with a high quality
since the transmission line has a high error rate.

 In a case where a code string is transmitted and/or
25 stored via such a medium of a high error rate, an error
correcting code, such as a Bose-Chaudhuri-Hocquenghem (BCH)
code, a Recommended Standard (RS) code and a convolutional
code, is often used as means for reducing the error rate.
In addition, an error correcting code, such as a check sum
30 and a cyclic redundancy check (CRC), is used as means for
enabling the error correction on the receiving side. The
aforementioned error correction and/or error detection adds
excessive bits (redundancy) to the transmitted and/or
stored information in accordance with a predetermined rule
35 to examine whether the transmitted and/or stored code
string obeys the rule during decoding, to carry out the
error correction/detection on the basis of the results.

However, in such a method for coding a code string obtained by a high-efficiency compression coding into error correcting/detecting codes to transmit and/or store the codes, there is a disadvantage in that it is difficult to
5 be combined with a synchronous recovering technique for recovering a step out caused by a code word error in the transmission line/medium. As a synchronous recovering technique, a method for inserting a uniformly decodable code called a synchronization code to resume the decoding
10 immediately after the synchronization code is detected when a step out occurs is often used.

In order to make a uniformly decodable code word of the synchronization code, the code word must be combined with another code word so as not to form the same bit
15 pattern as the synchronization code. However, in an error correction-/detection coding, it is generally difficult to form a code word so as to prevent a certain bit pattern from occurring. When the same bit pattern as the synchronization code occurs, a pseudo synchronization may
20 be caused by the error detection of the synchronization code.

In order to avoid this problem, there is used a method for preventing the pseudo synchronization by determining whether the same bit pattern as the
25 synchronization code exists in the code string after performing the error correction/detection coding, inserting a dummy bit into the bit pattern in accordance with a certain rule when the same bit pattern exists, and deleting the dummy bit in the same rule in a decoding system.
30 However, in a case where the code string is transmitted and/or stored via a medium in which errors are easy to occur, errors may occur in the inserted bit, so that there is a problem in that a new step out or a new pseudo synchronization may occur.

35 In addition, in a case where the error correction/detection coding of a code string is carried out to insert a synchronization code, there is also a problem

in that the coding efficiency is lowered since it is required to add many inserted bits to the code string in order to compensate an excess of information bits, for which the error detection and/or detection coding is to be carried out, at the end portion of a synchronization interval between the adjacent synchronization codes.

On the other hand, in order to enhance the error correcting/detecting capability, the redundancy of the transmitted and/or stored information may be enhanced. However, if the redundancy is enhanced, the number of bits required to transmit the same information is increased. Therefore, if the error correcting/detecting capability is enhanced, it is required to provide a transmission line of a higher transmission rate, or the number of bits of information to be stored is increased. In addition, if the transmission rate and the stored capacity are the same, the amount of information, which can be transmitted and/or stored, is decreased as the redundancy is enhanced. In a case where a picture and/or voice information is high-efficiently compression-coded to be transmitted and/or stored, in order to add the redundancy to improve the error resistance, the compression coding into a smaller amount of information must be carried out if the transmission and/or storage rate is the same, so that the picture quality and sound quality are lowered.

Therefore, as a method for providing a smaller redundancy and a high error resistance, there is a method called hierarchical coding. This is a method for enhancing the error resistance using the same means redundancy in comparison with when the same error correcting/detecting code is used, by classifying the high-efficiency compression-coded information in accordance with the magnitude of errors influencing the picture quality and the sound quality, using an error correcting/detecting code of a higher error correction/detection capacity while having a high redundancy, for information under a great influence of the error, and using an error correcting/detecting code

of a less redundancy while having not so high error correction/detection capacity, for information which is not so greatly influenced by the error.

For example, in a coding system, which is formed by
5 combining the motion-compensated prediction with the
orthogonal transform and which is often used for the
high-efficiency compression coding of a dynamic image,
i.e., in a system for motion-compensation predicting an
input dynamic image signal to orthogonal transform its
10 prediction residue by a DCT (discrete cosine transform) or
the like, the error correcting/detecting codes of a high
error correcting/detecting capability are used for motion
vector information, which greatly deteriorate the picture
quality if an error occurs, and for lower coefficients of
15 the orthogonal transformation coefficients of the
prediction residual signal, and the error
correcting/detecting codes of a low error correcting-
/detecting capability are used for higher coefficients of
the orthogonal transformation coefficients of the
20 prediction residual signal, which are under a small
influence of the error.

In order to achieve such a hierarchical coding, it
is required to switch the error correcting/detecting codes
of different error correcting/detecting capabilities in the
25 middle of the output code string. As a method for
switching the error correcting/detecting codes of different
error correcting/detecting capabilities, there is a method
for adding a header information representative of the kind
of the error correcting/detecting codes to a code string.
30 FIG. 11 shows an example of a code string, to which the
header information is added to switch the error
correcting/detecting codes. In this example, two kinds of
error correcting/detecting codes FEC1 and FEC2 are
switched. The headers 1101 to 1104 have a header
35 information representative of the kind of the error
correcting/detecting codes and the number of the code
words. In a coding system, code words, which are error

correction/detection coded, are arranged after the header information. In a decoding system, the header information is decoded, and the error correcting/detecting codes are decoded in accordance with the decoding of the header information.

However, in the method for switching the error correcting/detecting codes by adding such a header information, there is a problem in that the number of bits of the code string to be transmitted and/or stored by adding the header information is increased. In a case where the picture and/or voice information is high-efficiency compression coded to be transmitted and/or stored, if the number of bits is occupied by the header information, the number of bits used for the high-efficiency compression coding of the picture and/or voice information is decreased, so that the picture quality and sound quality are deteriorated.

As mentioned above, if the error correction/detection coding of the code string, in which the high-efficiency compression coding of dynamic image signals and so forth is performed, is carried out, an optional bit pattern is produced. Therefore, in a case where the error correction-/detection coding is combined with the synchronous recovering technique using a uniformly decodable synchronization code, there is a problem in that a pseudo synchronization is caused by the error detection of the synchronization code. Also in a case where the dummy bit is inserted to prevent the pseudo synchronization, there is a problem in that a new step out or a new pseudo synchroni-zation is caused by the error of the inserted bit.

In addition, in a case where the error correction-/detection coding of the code string is performed and the synchronization code is inserted, it is conventionally required to use many inserted bits to compensate an excess of information bits, for which the error detection and/or detection coding is to be carried out, at the end portion

of a synchronization interval between the adjacent synchronization codes, so that there is a problem in that the coding efficiency is lowered.

Moreover, in a coding/decoding system for switching
5 error correcting/detecting codes of different error
correcting/detecting capabilities by adding a header
information, the number of bits to be transmitted and/or
stored by adding the header information is increased.
Therefore, in a case where a picture and/or voice
10 information is high-efficiency compression coded to be
transmitted and/or stored, there is a problem in that the
information content assigned to the picture and/or voice
information is decreased to lower the picture quality and
the sound quality.

15

DISCLOSURE OF INVENTION

It is therefore a principal object of the present
invention to eliminate the aforementioned problems and to
provide coding and decoding systems, which can prevent a
20 pseudo synchronization and a step out due to the error
detection of a synchronization code.

It is another object of the present invention to
provide coding and decoding systems, which prevent a pseudo
synchronization and a step out due to the error detection
25 when an error correction/detection coding is combined with
a synchronization recovering technique using a
synchronization code.

It is further object of the present invention to
provide coding and decoding systems, which can decrease the
30 number of inserted bits used at the end portion in a
synchronization interval to enhance the coding efficiency,
when an error correction/detection coding is combined with
a synchronization recovering technique using a
synchronization code.

35 It is still further object of the present invention
to provide coding and decoding systems which can decrease
the number of bits of a code string, to which a header

information representative of the kind of error correction/detection coding must be added and which is transmitted and/or stored, to improve the quality of information.

5 (1) A first coding system, according to the present invention, comprises: coding means for coding an input code string to an error correcting/detecting code comprising an information bit and a check bit; and code string assembling means for inserting a synchronization code into any one of
10 a plurality of predetermined synchronization code inserting positions in an output code string, for arranging the information bit at an optional position in the output code string, and for arranging the check bit at a position other than the synchronization code inserting positions in the
15 output code string to assemble the output code string.

 A first decoding system, according to the present invention, comprises: synchronization code detecting means for detecting a synchronization code at a plurality of predetermined synchronization code inserting positions on
20 the basis of a code string, which is coded to an error correcting/detecting code comprising an information bit and a check bit; code string resolving means for resolving the code string to extract the information bit of the error correcting/detecting code and the check bit of the error
25 correcting/detecting code arranged at a position other than the synchronization code inserting positions; and decoding means for receiving the information bit and the check bit extracted by the code string resolving means to decode the error correcting/detecting code.

30 Thus, in the first coding/decoding system, the synchronization code exists only at the predetermined synchronization code inserting position in the output code string, and the check bit of the error correcting/detecting code exists at a position other than the synchronization
35 code inserting position. Therefore, even if the same bit pattern as the synchronization code is contained in the check bit, there is no probability that it is erroneously

detected that the same bit pattern as the synchronization code is a synchronization code. Accordingly, since it is not required to use a special error correcting/detecting code for preventing a particular bit pattern and to insert
5 a bit for preventing a synchronization code after the error correction/detection coding, the degree of freedom for the used error correcting/detecting code can be increased. In addition, since there is no probability that a new synchronization detection error may occur due to the
10 insertion of an error into an inserted bit, it is possible to improve the resistance to errors.

(2) In a second coding system according to the present invention, code string transforming means is added to the first coding system. The code string transforming
15 means transforms an input code string other than synchronization codes arranged at a plurality of predetermined synchronization code inserting positions in an output code string so that a humming distance from the synchronization code is equal to or greater than a
20 predetermined value. The code string transformed by the code string transforming means is inputted to coding means to be coded to an error correcting/detecting code comprising an information bit and a check bit.

In a second decoding system according to the present invention, code string transforming means is added
25 to the first decoding system. The code string transforming means transforms a code string other than the synchronization code, which exists at a synchronization code inserting position and which is transformed so that
30 a humming distance from the synchronization code in the code string decoded by the decoding means is equal to or greater than a predetermined value, into the original code string.

Thus, in the second coding/decoding system, the
35 coding system carries out the transforming process so that the humming distance from the synchronization code with respect to a bit string arranged at the synchronization

code inserting position is equal to or greater than a predetermined value, and the decoding system carries out the reverse transformation. Therefore, since the same bit pattern as the synchronization code is not contained in the bit string, it is possible to prevent the error detection of a synchronization code. In addition, if the transform is carried out so that the humming distances between the synchronization code and the other code strings are great, the synchronization code can be distinguished from the other code strings even if an error is mixed into the code strings, so that the probability of the error detection of the synchronization code due to errors can be decreased.

Since this transforming/inverse transforming process is carried out only at the synchronization code inserting position, the overhead is less than those in conventional methods for carrying out transforming/inverse transforming process over the whole code string. In addition, in the code string inputted to the coding system, it is not required to carry out the transforming process so as not to produce the same bit pattern as the synchronization code, and it is not required to use a special code string. In particular, in a case where a variable length coding system for switching and using different code word tables is used as an input of this coding system, if the code word table is prepared so as not to produce the same pattern as the synchronization code in the variable length coding system, there is a problem in that the coding efficiency is lowered. However, the second coding/decoding system can eliminate such a problem.

(3) A third coding system, according to the present invention, comprises: coding means for coding an input code string to an error correcting/detecting code; synchronization code inserting means for inserting a synchronization code into the code string; and determining means for determining the number of bits an information to be coded to an error correcting/detecting code immediately before the synchronization code in the code string, wherein the

coding means causes the error correcting/detecting code immediately before the synchronization code, to be a degenerate code, which adaptively degenerated on the basis of the number of bits determined by the determining means.

5 A third decoding system, according to the present invention, comprises: decoding means for decoding a code string, which is coded to an error correcting/detecting code and into which a synchronization code is inserted; synchronization code detecting means for detecting the
10 synchronization code in the code string; and determining means for determining the number of bits of an information, which is coded to an error correcting/detecting code immediately before the synchronization code in the code string detected by the synchronization code detecting
15 means, wherein the decoding means decodes by identifying whether the error correcting/detecting code immediately before the synchronization code is a degenerate code on the basis of the determined result of the determining means.

 Thus, in the third coding/decoding system, since a
20 degenerate code degenerated to the number of bits required to code the information bit remaining at the end portion of a synchronization interval is used as the error correcting-/detecting code immediately before the synchronization code, it is not required to use many
25 inserted bits in order to compensate the remainder of the information bit at the end portion of the synchronization interval, so that the coding efficiency can be improved.

 (4) A fourth coding system, according to the present invention, comprises: coding means for coding a
30 code string containing kinds of input information to an error correct-ing/detecting code; and switching means for switching the kind of the error correcting/detecting code in accordance with the kinds of the input information in the code string.

35 A fourth decoding system, according to the present invention, comprises: decoding means for decoding a code string, which is coded to a different kind of error

correcting/detecting code in accordance of the kind of information, to generate the original information; and means for determining the kind of the error correcting-/detecting code on the basis of the kind of information generated by the decoding means, to inform the decoding means.

Thus, in the fourth coding/decoding system, in a case where the coding/decoding is carried out by switching the error correcting/detecting code in accordance with the kind of information, the coding system switches the error correcting/detecting code in accordance with the kind of information of the input code string, and the decoding system determines the kind of the error correcting-/detecting code on the basis of the decoded information to carry out the same switching as that of the coding system. Therefore, it is not required to use a header information representative of the kind of the error correcting/detecting code unlike conventional systems, so that it is possible to remove the overhead due to the header information.

(5) A fifth coding system, according to the present invention, comprises: code string transforming means for transforming an input code string other than synchronization codes, which are arranged at a plurality of predetermined synchronization code inserting positions in an output code string and in intervals of a predetermined number of bits before and after the predetermined synchronization code inserting positions in an output code string, so that a humming distance from the synchronization code is equal to or greater than a predetermined value; coding means for coding a code string transformed by the code string transforming means, to an error correcting-/detecting code comprising an information bit and a check bit; and code string assembling means for inserting a synchronization code into any one of a plurality of predetermined synchronization code inserting positions in the output code string, for arranging the information bit

at an optional position in the output code string, and for arranging the check bit at a position other than the synchronization code inserting positions in the output code string to assemble the output code string.

5 A fifth decoding system, according to the present invention, comprises: synchronization code detecting means for detecting a synchronization code at a predetermined synchronization code inserting position and in an interval of a predetermined number of bits before and after the
10 predetermined synchronization code inserting position, on the basis of a code string, which is coded to an error correcting/detecting code comprising an information bit and a check bit and into which the synchronization code; code string resolving means for resolving the code string to
15 extract the information bit of the error correcting-/detecting code and the check bit of the error correcting/detecting code arranged at a position other than the synchronization code inserting position; decoding means for receiving the information bit and the check bit
20 extracted by the code string resolving means to decode the error correcting/detecting code; and code string transforming means for transforming a code string other than the synchronization code, which is transformed so that a humming distance from the synchronization code in the
25 code string decoded by the decoding means is equal to or greater than a predetermined value and which exists at the synchronization code inserting position and in an interval of a predetermined number of bits before and after the synchronization code inserting position, into the original
30 code string.

 Thus, in the fifth coding/decoding system, the input code string is transformed in the synchronization code inserting interval as well as in the interval of the predetermined number of bits before and after the
35 synchronization code inserting interval so that the humming distance is equal to or greater than the predetermined value, and the decoding system carries out the reverse

transformation of the input code string, so that the same bit pattern as the synchronization code is not contained in this interval. Therefore, even if the transmission/storage of a bit string coded through a transmission line or a storage medium for causing a bit loss, in which a part of bits is lost, and a bit addition, in which an excess bit is added, is carried out, the synchronization code can be distinguished from the other code strings if the number of lost/added bits is equal to or less than a predetermined number of bits, so that the decoding system can correctly carry out the synchronization detection.

(6) A sixth coding system, according to the present invention, comprises: multiplexing means for multiplexing kinds of compressed codes, which are obtained by compression coding an input signal, to produce a multiplexed code string; and code string assembling means for inputting the multiplexed code string to assemble an output code string, wherein the code string assembling means inserts a synchronization code into any one of a plurality of periodically predetermined synchronization code inserting positions in the output code string.

In this case, in order to insert the synchronization code into any one of the plurality of periodically predetermined synchronization code inserting positions in the output code string, the code string assembling means inserts a stuffing bit into the output code string, or periodically determines the plurality of synchronization code inserting positions in the output code string and inserts an information (a pointer information) representative of the boundary of the multiplexed code string, to insert the synchronization code into any one of the plurality of synchronization code inserting positions. Thus, the synchronization code can be inserted only into any one of the plurality of periodically predetermined synchronization code inserting positions.

The stuffing bit is preferably a code, which can be

uniformly decoded in a backward direction of the output code string. Thus, if the decoding system compares the decoding end position of the code string immediately before the stuffing bit with the starting position of the stuffing
5 bit, it is possible to easily detect an error in the input code string.

Moreover, the humming distances of the stuffing bit from the synchronization code and the part thereof are preferably equal to or greater than a predetermined value.
10 Thus, there is an advantage in that the probability of the occurrence of a pseudo synchronization.

A sixth decoding system, according to the present invention, comprises: synchronization code detecting means for detecting a synchronization code in an output code
15 string; demultiplexing means for demultiplexing on the basis of the position of the synchronization code detected by the synchronization code detecting means from the input code string, to produce a compressed code string; and
20 decoding means for decoding the compressed code to output a reconstructed signal, wherein the synchronization code detecting means detects the synchronization code at a plurality of periodically predetermined synchronization code inserting positions in the input code string.

Thus, in the case of the sixth coding/decoding
25 system, since the coding system inserts the synchronization code into the multiplexed code string, which is obtained by multiplexing kinds of compressed codes, only at the periodically predetermined synchronization code inserting positions, the decoding system may carry out the
30 synchronization detection only at the synchronization code inserting positions, so that the number of synchronization code detecting processes can be decreased in comparison with conventional systems for inserting a synchronization code into a code string at an optional position. In
35 addition, in accordance with the decrease of the number of the synchronization code detecting processes, it is possible to decrease the probability that a pseudo

synchronization occurs because of the bit string inputted to the decoding system changed to the same bit pattern as the synchronization code due to bit error. Therefore, according to the present invention, it is possible to decrease the number of the occurrences of the pseudo synchronization, so that the processing quantity in the synchronization code detection can be decreased.

(7) In a seventh coding system according to the present invention, code string transforming means is added to the sixth coding system. The code string transforming means transforms a code string other than said synchronization code, which is arranged at said synchronization code inserting position in said output code string, so that a humming distance from said synchronization code is equal to or greater than a predetermined value.

A seventh decoding system, according to the present invention, further comprises: code string transforming means for transforming a code string other than the synchronization code, which is transformed so that a humming distance from the synchronization code is equal to or greater than a predetermined value at the synchronization code inserting position of the input code string, into the original code string.

Thus, in the seventh coding/decoding system, in addition to the construction of the sixth coding/decoding system, the bit insertion in view of the error of the synchronization code, i.e., the transformation of the humming distance of the bit string arranged at the synchronization code inserting position from the synchronization code, is carried out, so that the same bit pattern as the synchronization code is contained in the bit code string. Therefore, it is possible to ensure that the error detection of the synchronization code does not occur with respect to an error of a supposed number of bits or less, so that it is possible to decrease the probability of the error detection of the synchronization code.

Furthermore, in the present invention, the synchroniza-tion code is a code string, which is inserted into a code string for synchronization recovery and which can be uniformly decoded. For example, if a code string
5 for inserting a synchronization code is a multiplexed code string, which is obtained by multiplexing a plurality of kinds of compressed codes obtained by compression coding an image signal inputted for each frame, the synchronization code is a code representative of a
10 partition of a coding frame, partitions of the plurality of kinds of compressed codes, and other partitions.

As described above, according to the present invention, the following advantages can be obtained.

(1) According to the first coding/decoding system,
15 the synchronization codes are inserted only into the synchroni-zation code inserting positions arranged at regular intervals, and the check bits of the error correcting-/detecting code are shifted so as to exist at positions other than the synchronization code inserting
20 positions. Therefore, even if the same bit pattern as the synchroni-zation code occurs in the check bit, the same pattern as the synchronization code does not occur at the synchroni-zation code inserting position, and, in theory, there is no probability that synchronization is detected.

25 In addition, if the bit insertion into a code string at the synchronization code inserting position is carried out so as not to produce a pseudo synchronization code, it is possible to eliminate the difficulty in forming a code word so as not to produce the same bit pattern as
30 the synchroni-zation bit.

(2) According to the second coding/decoding system, in addition to the construction of the first coding system, the bit insertion is carried out in view of the synchroni-zation code, i.e., the transforming process is carried out
35 so that the humming distance from the synchronization code is equal to or greater than the predetermined value with respect to the bit string arranged at the synchronization

code inserting position, and the decoding system carries out the backward transformation, so that the bit code does not contain the same bit pattern as the synchronization code. Therefore, it is possible to ensure to prevent the error detection of the synchronization code from occurring due to the errors of the supposed number of bits or less, so that the probability of the error detection of the synchronization code is decreased. In addition, if such transformation is carried out, it is possible to distinguish the synchronization code from the other code strings even if an error is mixed into the code string, so that the probability of the error detection of the synchronization code due to errors can be decreased.

Moreover, since the error correction/detection coding is carried out after the code word is transformed by the bit insertion, the error protection is carried out for the inserted bit. Therefore, in comparison with conventional systems for carrying out the bit insertion after the error correction/detection coding, the probability that an error occurs in the inserted bit is decreased. In addition, since the bit insertion is carried out only into the synchronization code inserting position, the increase of the quantity of cods due to the bit insertion is smaller than that in conventional systems for carrying out the bit insertion over the whole code string, so that the coding efficiency is improved.

(3) According to the third coding/decoding system, since the error correcting/detecting code immediately before the synchronization code is a degenerate code, it is possible to decrease the number of inserted bits for compensating the remainder of the information bit at the portion immediately before the synchronization code, to improve the coding efficiency, in comparison with conventional coding systems.

(4) According to the fourth coding/decoding system, the coding system utilizes the multiplexing rule in the high-efficiency compression coding system, such as image

and voice, and switches the error correcting/detecting code in accordance with the kind of the information of an input code string, and the decoding system determines the kind of the error correcting/detecting code on the basis of the
5 decoded information to carry out the same switching as that of the coding system. Therefore, it is not required to add the header information representative of the kind of the error correcting/detecting code, and it is possible to
10 assign the number of bits to high efficiency compression coding, such as image and voice, so that it is possible to improve the information quality, such as picture quality and sound quality.

(5) According to fifth coding/decoding system, the coding system transforms an input code string so that a
15 humming distance in a synchronization code inserting interval as well as a humming distance in an interval of a predetermined number of bits before and after the synchronization code inserting interval are equal to or greater than a predetermined value, and the decoding system
20 carries out the inverse transformation of the input code string, so that it is possible to prevent the same bit pattern as the synchronization code from being contained in this interval. Therefore, even if a bit string coded through a transmission line and/or a storage medium, which
25 may cause a bit loss for losing a part of bit is lost and a bit addition for adding an excess bit, is transmitted/stored, if the number of lost/added bits is equal to or less than a predetermined number of bits, it is possible to distinguish a synchronization code from
30 other code strings, so that the decoding system can correctly carry out the synchronization detection.

(6) According to the sixth coding/decoding system, since the coding system inserts synchronization codes only into synchronization code inserting positions periodically
35 predetermined in a multiplexed code string, which is obtained by multiplexing kinds of compressed codes, the decoding system may carry out the synchronization detection

only at the synchronization code inserting positions, so that the number of the synchronization code detecting processes can be decreased in comparison with conventional systems for inserting synchronization codes into a code string at optional positions.

In addition, since the probability that a pseudo synchronization occurs by the variation of the bit string inputted to the decoding system into the same bit pattern as the synchronization code, is decreased in accordance with the decrease of the number of the synchronization code detecting processes, it is possible to prevent the pseudo synchronization according to the present invention, so that the processing quantity for the synchronization code detection decreases.

(7) According to the seventh coding/decoding system, in addition to the construction of the sixth coding/decoding system, the bit insertion in view of the error of the synchronization code, i.e., the transformation of the humming distance of the bit string at the synchronization code inserting position from the synchronization code is carried out, so that the same bit pattern as the synchronization code is not contained in the bit string. Therefore, it is ensured that the error detection of the synchronization code does not occur with respect to errors of the supposed number of bits, so that the probability of error detection of synchronization codes can be decreased.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of the first and second preferred embodiments of a dynamic image coding system according to the present invention.

FIG. 2 is a diagram illustrating a multiplexing rule in a multiplexer of the dynamic image coding system of FIG. 1.

FIG. 3 is a block diagram of an output coding unit of the dynamic image coding system of FIG. 1.

FIG. 4 is a diagram illustrating an example of an

output code string of the dynamic image coding system of FIG. 1.

FIG. 5 is a diagram illustrating an example of a synchronization code.

5 FIG. 6 is a block diagram of an error correction-/detection switching coding part in the output coding unit of FIG. 3.

FIG. 7 is a block diagram of a code string assembling unit of the output coding unit of FIG. 3.

10 FIG. 8 is a block diagram of the first and second preferred embodiments of a dynamic image decoding system according to the present invention.

FIG. 9 is a block diagram of an input decoding unit of the dynamic image decoding system of FIG. 8.

15 FIG. 10 is a block diagram of a code string resolver of the input decoding unit of FIG. 9.

FIG. 11 is a diagram illustrating an example of a code string obtained by a conventional error correction-/detection switching coding system.

20 FIG. 12 is a diagram illustrating an example of a synchronization code, in which an error is caused by the addition/loss of a bit(s) in a transmission line, for explaining the second preferred embodiment of the present invention.

25 FIG. 13 is a diagram explaining the operation of a bit inserting unit of FIG. 3 in the second preferred embodiment.

FIG. 14 is a diagram explaining the operations of a synchronization detector and an inserted bit removing unit of FIG. 9 in the second preferred embodiment.

30 FIG. 15 is a diagram illustrating an example of a code string, in which a synchronization protection is carried out using a frame length information in the first and second preferred embodiments.

35 FIG. 16 is diagram illustrating another example of a code string, in which a synchronization protection is carried out using a frame length information in the first

and second preferred embodiments.

FIG. 17 is diagram illustrating further example of a code string, in which a synchronization protection is carried out using a frame length information in the first and second preferred embodiments.

FIG. 18 is a block diagram of the third and fourth preferred embodiments of a dynamic image coding system according to the present invention.

FIG. 19 is a block diagram of an output coding unit of the third preferred embodiment of a dynamic image coding system according to the present invention.

FIG. 20 is a diagram illustrating an example of an output code string of the third preferred embodiment of a dynamic image coding system according to the present invention.

FIG. 21 is a block diagram of the third and fourth preferred embodiments of a dynamic image decoding system according to the present invention.

FIG. 22 is a block diagram of an input decoding unit of the third preferred embodiment of a dynamic image decoding system according to the present invention.

FIG. 23 is a block diagram of an output coding unit of the fourth preferred embodiment of a dynamic image coding system according to the present invention.

FIG. 24 is a block diagram of an input decoding unit of the fourth preferred embodiment of a dynamic image decoding system according to the present invention.

FIG. 25 is a block diagram of an example of an output code string of the fourth preferred embodiment of a dynamic image coding system according to the present invention.

FIG. 26 is a diagram illustrating a multiplexing rule in a multiplexer of a dynamic image coding system.

FIG. 27 is a block diagram of examples of output code strings of the fifth preferred embodiment of a dynamic image coding system according to the present invention.

FIG. 28 is a diagram illustrating other examples of

multiplexing rules in a multiplexer of a dynamic image coding system.

FIG. 29 is a diagram of an output code string, in which synchronization codes are processed when the multiplexing shown in FIG. 28 is carried out.

FIG. 30 is a code table for explaining examples of stuffing bits used for the present invention.

FIG. 31 is a diagram explaining a process in a decoding system when the stuffing bits of FIG. 30 are used.

FIG. 32 is a diagram explaining the characteristics of the stuffing bits of FIG. 30.

FIG. 33 is a diagram of examples of output code strings when a synchronization code inserting interval is shorter than a synchronization code.

FIG. 34 is a diagram of examples of the use of synchronization codes of different lengths.

FIG. 35 is a diagram of examples of output code strings of the sixth preferred embodiment of a dynamic image coding system according to the present invention.

BEST MODE FOR CARRYING OUT THE INVENTION

Referring now to the accompanying drawings, the preferred embodiments of the present invention will be described below.

(First Preferred Embodiment)

FIG. 1 is a block diagram of the first preferred embodiment of a dynamic image coding system, according to the present invention, wherein a coding system having an error correcting/detecting code switching function of the present invention is combined with a high-efficiency compression coding system which uses a motion-compensated adaptive prediction and a discrete cosine transform coding serving as a kind of an orthogonal transform coding. A coding system comprising the combination of a motion-compensated adaptive prediction and a discrete cosine coding is detailed in, e.g., Literature 1 "International Standard of Multimedia Coding" by Hiroshi

Yasuda, Maruzen (June 1991). Therefore, only the operation of the coding system will be schematically described. In addition, it is assumed that information bits are separated from detection bits in an error correcting/detecting code similar to a BCH code.

5 In FIG. 1, with respect to an input dynamic image signal 131 serving as an object to be coded, which is inputted for each frame, a motion-compensated adaptive prediction is first carried out for each small region, such as a macro block. That is, in a motion-compensated adaptive predictor 101, a motion vector between an input dynamic image signal 131 and an image signal, which is stored in a frame memory 102 and which has been encoded and/or locally decoded, is detected, and a prediction signal 132 is produced by the motion-compensated prediction on the basis of the motion vector. In this motion-compensated predictor 101, a prediction mode suitable for coding is selected from the motion-compensated prediction coding and the intraframe coding (prediction signal = 0), which directly encodes the input dynamic image signal 131, to produce the corresponding prediction signal 132.

10 The prediction signal 132 is inputted to a subtracter 103, wherein the prediction signal 132 is subtracted from the input dynamic image signal 131 to output a prediction residual signal 133. The prediction residual signal 133 is discrete-cosine-transformed (DCT) in a discrete cosine transformer 104 for each block having a predetermined size to produce a DCT coefficient. The DCT coefficient is quantized by means of a quantizer 105. The DCT coefficient data quantized by the quantizer 105 are divided into two parts, one of which is variable-length-encoded by means of a first variable length encoder 106, and the other of which is inverse-quantized by means of an inverse quantizer 107 to be inverse-discrete-cosine-transformed (inverse DCT) by means of an inverse discrete cosine transformer 108. The output of the inverse discrete cosine transformer 108 is added to

the prediction signal 132 in an adder 109 to produce a locally decoded signal. This locally decoded signal is stored in the frame memory 102.

5 On the other hand, the information on the prediction mode and the motion vector, which have been determined by the motion-compensated adaptive predictor 101, is variable-length-encoded by means of a second variable length encoder 110. The variable length codes (compressed codes) outputted from the first and second
10 variable length encoders 106 and 110 are multiplexed by means of a multiplexer 111 to be outputted as a multiplexed code string 201.

The multiplexer 11 outputs the multiplexed code string 201, a FEC kind identifying signal 202
15 representative of the kind of an error correcting/detecting code corresponding thereto, and a synchronization code insertion request signal for requesting to insert a synchronization code. The code string 202, the FEC kind identifying signal 202 and the synchronization code
20 insertion request signal 203 are inputted to an output coding unit 200 for switching and encoding the code string 202 to kinds of error correcting/detecting codes of different error correcting-/detecting capabilities, to produce a final output code string 205. In this preferred
25 embodiment, the output coding unit 200 corresponds to a coding system according to the present invention.

FIG. 2 is a diagram illustrating a multiplexing flow in the multiplexer 111. The multiplexing is carried out for each coding frame. First, a synchronization code
30 301 is multiplexed. When the synchronization code 301 is multiplexed, the synchronization code insertion request signal 203 is outputted from the multiplexer 111, and the coding system 200 has notice that the multiplexed code word is a synchronization code. Then, picture headers 203
35 representative of various coding modes of the coding frame are multiplexed to be the multiplexed code string 201. Then, a prediction mode information 303 representative of

a prediction mode in a motion-compensated adaptive predictor MC in each region is multiplexed, and a motion vector information 304 and a DCT coefficient 305 of a predictive residual signal (hereinafter referred to as a "residual DCT coefficient") are multiplexed. When the picture header 302, the prediction mode information 303, the motion vector information 304 and the residual DCT coefficient 305 are multiplexed, the FEC kind identifying signals 202 representative of the kind of error correcting/detecting codes corresponding thereto are outputted.

The error correcting/detecting codes of high correct-ing/detecting capabilities are used for the picture header 302, the prediction mode information 303 and the motion vector information 304, which greatly deteriorate the picture quality if an error is mixed. On the other hand, if an error is mixed into the residual DCT coefficient 305, it is possible to prevent the picture quality from greatly deteriorating by detecting the error and setting the residue to be zero. Therefore, the error correcting capacity is not required to be high, and only the error detection may be carried out.

FIG. 3 is a block diagram of an output coding unit of FIG. 1. The output coding unit 200 comprises a bit inserting unit 211, an error correction/detection switching encoder 212 and a code string assembling unit 213. FIG. 4 shows an example of an output code string 205 produced by the output coding unit 200. In FIG. 4, PSC denotes a synchronization signal, PH denotes a picture header, MODE denotes a prediction mode information, MV denotes a motion vector, CHK denotes a check bit of an error correcting/detecting code, DOEF is a residual DCT coefficient, and STUFF denotes a stuffing bit (an inserted bit). This output code string 205 has the following characteristics.

(1) The synchronization codes PSC are inserted into only synchronization code inserting positions indicated by arrows, which are arranged at regular intervals (every sync_period bits). The length of the sync_period is set

to be greater than the length of the synchronization code PSC and the maximum length of the check bit CHK. The check bit CHK is shifted so as to be arranged immediately before the synchronization code inserting position.

5 (2) The error correcting/detecting code at the end part of a frame, i.e., a synchronization period between a synchronization code PSC and the next PSC, is a degenerate code for encoding only the finally remaining information bit, and the stuffing bits STUFF having the number of bits
10 required to displace the check bit CHK (CHK6 in the example of FIG. 4) are inserted.

 (3) The FEC kind identifying signal representative of the kind and number of the error correcting/detecting codes does not exist in the output code string 205 of FIG.
15 4.

 In this output code string 205, since the check bit CHK is displaced as described in (1), no check bit CHK enters the synchronization code inserting positions indicated by the arrows, so that there is no probability
20 that a pseudo synchronization is produced by the check bit CHK. In addition, in a case where the error correction/detection coding of the end of the frame is carried out as described in (2), it has been required to insert many inserted bits (dummy bits) in the prior art.
25 However, in this preferred embodiment, since the end of the frame has the degenerate code, the number of inserted bits may be small. Moreover, as described in (3), since the header information representative of the kind and number of the error correcting/detecting codes are contained in
30 the output code string 205, the amount of codes is not increased therefor.

 Comparing the multiplexed code string 201 of FIG. 2 outputted from the multiplexer 111 with the output code string 205 of FIG. 4, the construction and operation of the
35 output coding unit 200 of FIG. 3 for producing such an output code string 205 will be described in detail below.

 When the synchronization code 301 is multiplexed in

the multiplexer 111, the synchronization code insertion request signal 203 is outputted as described above. For example, as shown in FIG. 5, the synchronization code 301 comprises "0"s of sync_0_len bits, a "1" of one bit, and a "xxxxx" of sync_nb_len bits representative of the kind of the synchronization code 301. The output coding unit 200 outputs a synchronization code (PSC) serving as an output code string 205 from the code string assembling unit 213 when receiving the synchronization code 301 and the synchronization code insertion request signal 203 from the multiplexer 111.

As shown in FIG. 4, since the synchronization code 301 can be inserted only at the synchronization code inserting positions arranged at intervals of sync_period bits, when the end of the output code string 205 produced is not arranged at the synchronization code inserting position, a stuffing bit STUFF is inserted so that the synchronization code 301 is arranged at the synchronization code inserting position as described later.

After the synchronization code 301 is outputted to the output code string 205, the picture header 302, the prediction mode information 303, the motion vector information 304 and the residual DCT coefficient 305 are encoded as follows. The bit insertion into the multiplexed code string 201 outputted from the multiplexer 111 is carried out to prevent a pseudo synchronization from occurring in the bit inserting unit 211. That is, if the same bit pattern as the code word of the synchronization code 301 exists in the output code string 201, the bit insertion is carried out, if necessary, in order to prevent the synchronization code 301 from being not able to be uniformly decoded. For example, in a case where the synchronization code 301 is a code word having continuous "0"s of sync_0_len bits as shown in FIG. 5, if a "1" is inserted so that "0"s of the sync_0_len bits or more do not continue in code strings other than the synchronization code 301, it is possible to prevent a pseudo

synchronization from being produced.

As described above, since the synchronization codes 301 are inserted only at the synchronization code inserting positions, the bit inserting operations for preventing the pseudo synchronization from occurring may be carried out only at the synchronization code inserting positions. Therefore, a counted value 221 representative of the total number of bits of the output code strings 205 produced is outputted from the code string assembling unit 213, and it is determined by the bit inserting unit 211 on the basis of the counted value 221 whether the bit insertion is required. Assuming that the counted value 221, i.e., the total number of bits of the output code strings 205 produced, is `total_len`, the number of "1"s in the multiplexed code string 201 is counted in an interval wherein $0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len}$. If no "1" exists in this interval, a "1" of one bit is inserted. A $\bmod B$ denotes a remainder when A is divided by B.

In addition, in order to decrease the probability of the error detection of the synchronization code 301, the bit insertion may be carried out as follows.

In order to detect the synchronization code 301 even if an error of n bits is mixed into the synchronization code 301, it is required to determine that a code word having a hamming distance of n or less from a true synchronization code in an input decoding unit of a dynamic image decoding system as described later is a synchronization code. However, if such a determination is carried out while the code strings other than the synchronization code 301 are as they are, bit patterns having a hamming distance of n or less from the synchronization code may exist even in the code strings other than the synchronization code 301. Therefore, if this is arranged at the synchronization code inserting position, it may be erroneously determined to be the synchronization code 301.

Therefore, the bit insertion into the multiplexed

code string 201 is carried out by means of the bit inserting unit 211, so that the code strings other than the synchronization codes arranged at the synchronization code inserting positions in the multiplexed code string 201 are transformed so as to have a humming distance of $2 * n + 1$ or more from the synchronization code 301. Specifically, the number of "1"s (assumed to be n_0) is counted in an interval wherein $0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len} - (2 * N + 1)$. If n_0 is equal to or less than $2 * n + 1$, "1"s of $2 * n + 1 - n_0$ bits are inserted into the multiplexed code string 201.

Thus, a code string 222, in which the bit insertion is carried out by means of the bit inserting unit 211, together with the FEC kind identifying signal 202 representative of the kind of an error correcting/detecting code, is inputted to the error correcting/detecting code switching encoding part 212.

FIG. 6 is a block diagram of an error correcting-/detecting code switching encoding part 212. A latch circuit 603 is a circuit for latching a FEC kind identifying signal 202. When the output of a synchronization code from the multiplexer 111 to the multiplexed code string 201 is completed to stop the output of the synchronization code insertion request signal 203, the latch circuit 603 latches the FEC kind identifying signal 202 to supply a latched signal 623 to an error correction/detection encoder 604.

An error correction/detection encoder 604 carries out the error correction/detection coding of a code string 222, which is outputted from a bit inserting unit 211, on the basis of the latched signal 623 to output an information bit 631 and a check bit 632. In addition, when the error correction/detection coding of one block is completed, the error correction/detection encoder 604 outputs a latch indicating signal 625, which indicates the latch of the next FEC kind identifying signal 202, to the latch circuit 603. The latch circuit 603 latches in accordance with the latch indicating signal 625 to supply

the latched signal 623 to the error correction/detection encoding 604.

5 The aforementioned operation is repeated in the output coding unit 200, so that the error correction/detection coding of the bit-inserted code string 222 outputted from the bit inserting unit 211 is carried out while switching the error correcting/detecting code in the error correct-ion/detection switching encoder 212 on the basis of the FEC kind identifying signal 202 outputted from the multiplexer 111. Since the FEC kind identifying signal 202 is latched by the latch circuit 603 only at the time that the coding of the error correcting/detecting code of one block is completed, the same error correcting/detecting codes are applied before this switching point. For example, in a case where the picture header 302 uses an error correcting/detecting code FEC1 and the prediction mode information 303 uses an error correcting/detecting code FEC2, if the number of bits of the picture header 302 is smaller than the number of information bits of one block of the FEC1, the FEC1 is used as the error correcting/detect-ing code of the subsequent prediction mode information 303 until the number of bits of the picture header 302 reaches the number of information bits of the FEC1.

25 FIG. 7 is a block diagram of a code string assembling unit 213 of FIG. 3. The code string assembling unit 213 comprises a counter 701 for counting the number of bits of the output code string 205, a buffer 702 for temporarily storing the check bit 632 and the number of bits thereof, a switch 703 for switching the output code string 205, and a switch controller 704 for controlling the switch 703.

35 The counter 701 is reset to be a value of a synchronization code length sync_len when the synchronization code insertion request signal 203 is inputted, and counts up sequentially from the next bit of the synchronization code until the next synchronization

code is inputted. The switch 703 is operated so that the information bit 631 is outputted before the initial check bit 632 is inputted after the synchronization code is inputted. When the check bit 632 is inputted, it is stored
5 in the buffer 702 and the number of bits thereof (the number of check bits) 711 is outputted from the buffer 702 to the switch controller 704.

The switch controller 704 controls the switch 703 on the basis of the number 711 of check bits and the
10 counted value 221 of the counter 701 so that the check bit is displaced in order to prevent the check bit 632 from being outputted to the synchronization code inserting position as described above. For example, assuming the counted value 221 is `bit_count` and the number 711 of check
15 bits is `check_len`, the information bit 631 is outputted when $\text{bit_count} \bmod \text{sync_period} < \text{sync_period} - \text{check_len}$, and a check bit 713 stored in the buffer 702 is outputted when $\text{sync_period} - \text{check_len} \leq \text{total_bits} \bmod \text{sync_period} < \text{sync_period}$. Thereafter, the
20 aforementioned process is repeated while the information bit 631 and the check bit 632 are inputted.

As described above, since the output coding unit 200 uses a degenerate code as an error correcting/detecting code at the end portion of one frame and performs the bit
25 insertion to displace the check bit, it performs operation different from usual operations of portions other than the end portion. That is, when the output of the multiplexed code string 201 of one frame is completed, the multiplexer 111 outputs a synchronization code insertion request signal
30 203 for the next frame. In response thereto, the error correcting/detecting code 604 in the error correction-/detection switching encoding part 212 of FIG. 6 performs the error correction/detection coding using a degenerate code, assuming that a shortage of the information bits 631
35 of the error correcting/detecting code is a predetermined bit pattern outputted from an inserted bit generator 705. This bit pattern may be a bit pattern wherein all bits are

"1"s or "0"s, or may be a specific repeated pattern such as "0101...". This compensated inserted bits are not outputted to the information bit 631.

5 In the code string assembling unit 213 of FIG. 7, after the information bit 613 is outputted to the end, the switch 713 is switched from the bit generator 705 to the input, and the inserted bits are inserted so that the check bit 713 stored in the buffer 702 is arranged immediately before the next synchronization code. Assuming that the
10 counted value 221 of the counter 701 is `total_len` when the last information bit 631 of one frame is outputted and that the number of the finally outputted check bits 632 is `last_check_len`, the number of inserted bits `stuffing_len` is $\text{stuffing_len} = \text{sync_period} - \text{last_check_len} -$
15 $(\text{total_len} \bmod \text{sync_period})$. Furthermore, in a case where no degenerate code is used, it is required to insert $(\text{into_len} - \text{last_into_len})$ bits, which is a shortage of the last information bit `last_into_len` from the usual information bit `into_len`, and to insert a shortage of bits
20 for displacing the check bit. Therefore, it is required to insert more inserted bits than those in the use of the degenerate code, by $\text{into_len} - \text{last_into_len} + (\text{into_len} - \text{last_into_len}) \bmod \text{sync_period}$.

25 Thus, after the code string assembling unit 213 outputs the information bits 631 and the inserted bits to the output code string 205 via the switch 703, and it finally outputs the check bit 713 to the output code string 205.

30 The first preferred embodiment of a dynamic image decoding system, according to the present invention, will be described below.

FIG. 8 is a block diagram of a dynamic image decoding system corresponding to the dynamic image coding system of FIG. 1. The output code string 205 outputted
35 from the dynamic image decoding system of FIG. 1 passes through a transmission/storage system to be inputted to an input decoding unit 800 as an input code string 205'. In

this preferred embodiment, the input decoding unit 800 corresponds to a decoding system of the present invention.

5 The input decoding unit 800 switches error correcting/detecting codes on the basis of a FEC kind identifying signal 802 representative of the kind of an error correcting/detecting code outputted from a downstream demultiplexer 811, to output a code string 801, which has been error correction/detection encoded, a synchronization code detection signal 803 and an error detection signal 10 804. The demultiplexer 811 receives the code string 801, the synchronization code detection signal 803 and the error detection signal 804 to separately output a prediction residual signal 841 and a motion-compensated adaptive prediction information code 842.

15 The prediction residual signal 841 and the motion-compensated adaptive prediction information code 842 are inputted to first and second variable length decoders 806 and 810, respectively. For a residual DCT coefficient 831 decoded by the first variable length decoder 806, a series of processes are carried out, i.e., the inverse 20 quantization is carried out by means of an inverse quantizer 807 and the inverse discrete cosine transform is carried out by means of an inverse DCT unit 808. Then, the processed residual DCT coefficient is added to a motion-compensated adaptive prediction signal 832, which 25 is an output of the motion-compensated adaptive predictor 801, in an adder 809 to be outputted as a reconstructed image signal 850. The reconstructed image signal 850 is outputted to the outside of the system and recorded in a frame memory 820. The motion-compensated adaptive prediction information decoded by the second variable 30 length decoder 810 is inputted to the motion-compensated adaptive predictor 801 to produce the motion-compensated adaptive prediction signal 832.

35 The aforementioned processes are processes for producing a dynamic image so as to correspond to the dynamic image coding system of FIG. 1. The processes of

the inverse quantizer 807, the inverse DCT unit 808, the adder 809 and the frame memory 820 are basically the same as those the inverse quantizer 107, the inverse DCT unit 108, the adder 109 and the frame memory 102 although the realizing means may be different. In addition, the processes in the first and second variable length decoders 806 and 810, the demultiplexer 811 and the input decoding unit 800 are inverse processes to those in the variable length encoders 106 and 110, the multiplexer 111 and the output coding unit 200, except for the case that an error is mixed into the code string.

FIG. 9 is a block diagram of the input decoding unit 800. The input decoding unit 800 comprises a synchroni-zation detector 901 for detecting a synchronization code in the input code string 205', a counter 902 for counting the number of bits of the input code string 205', a code string resolver 903 for resolving a code string into information bits 912 and check bits 913, an error correction/detection decoder 904, and an inserted bit remover 905.

The synchronization detector 901 detects synchronization codes only at the synchronization code inserting positions on the basis of the counted value 911 of the counter 902. For example, assuming that the interval between the adjacent synchronization code inserting positions is `sync_period`, the counted value 911 is `bit_count`, and the length of the synchronization code is `sync_len`, the synchronization detection is carried out only when $0 < \text{bit_count} \% \text{sync_period} \leq \text{sync_len}$.

Furthermore, the synchronization code may be detected in view of errors in the synchronization code.

In the bit inserting unit 211 of the output coding unit of FIG. 3, if the code string is transformed by inserting bits so as to have a humming distance of $2 * n + 1$ or more from the synchronization code in view of errors of n bits or less, even if it is determined that the code string having a humming distance of n or less from a true

synchronization signal is a synchronization code, error synchronization detection does not occur when the error of n bits or less is mixed.

FIG. 10 is a block diagram of the code string resolver 903. The input code string 205 is switched into information bits 1021 and check bits 913 by means of a first switch 1002 controlled by a controller 1001 which will be described later. When the information bits 1021 are outputted from the first switch 1002, an information bit length of the information bits 1021 are transmitted to a buffer 1004 via a second switch 1003 to be stored therein. A counter 1005 counts the number of output bits from the second switch 1003. A comparator 1006 compares a counted value 1023 of the counter 1005 with an information bit length 1024 outputted from an error correct-ing/detecting code information output unit 1007. When both are the same, the counter 1005 is reset, the FEC kind identifying signal 802 representative of the kind of an error detecting/correcting code is latched by means of a latch circuit 1008, and the buffer 1004 outputs the information bit 904. The output 914 of the latch circuit 1008 is inputted to the error correcting/detecting code information output circuit 1007, and outputted to the error correction/detection decoder 904 shown in FIG. 9.

As described above, the check bits of the error correcting/detecting code are displaced to be arranged between the information bits of the error correct-ing/detecting code after the code string 205. The controller 1001 controls so that the displaced check bits are separated from the information bits. When the input of the information bits of the error correcting/detecting code of one block is completed, the counted value 1023 corresponds to the information bit length 1024 in the comparator 1006. In response thereto, the controller 1001 receives a check bit length 1025 from the error correction/detection information output unit 1007, and calculates the positions of the check bits arranged between the next

information bits. When it is determined by the comparator 1006 that both correspond to each other, assuming that the counted value 911 of the number of input bits of the code string 205' is `bit_count` and the check bit length is `check_len`, the check bit starting position `check_start` is
5 `check_start = (bit_count / sync_period + 1) * sync_per`
`iod - check_len`, and the check bit end position `check_end`
is `check_end = (bit_count / sync_period + 1) * sync_per`
`iod`. The controller 1001 controls the switch 1002 so
10 that the check bits 913 are outputted from the `check_start`
to the `check_end` of the counted value 911.

Since the end of one frame is a degenerate code and the error correction/detection encoding thereof is carried out, it is specially processed. When reaching the end of
15 one frame, a signal 803 representative of the detection of the next frame is outputted. In response to this signal 803, the controller 1001 calculates the positions of the check bits of the last error correcting/detecting code of the frame and the number of insufficient bits of the
20 information bits. It is assumed that the counted value 911 of the number of bits of the code string 205' inputted when starting to input the last error correcting/detecting code of one frame is `pre_last_count`, the counted value 911 when the input of the code string 205' of one frame is completed
25 is `total_count`, the counted value 911 during the process is `bit_count`, the check bit length of the last error correcting/detecting code of one frame is `last_check_len`, and the check bit length of the error correcting/detecting code immediately before of the last error
30 correcting/detecting code is `pre_last_check_len`. First, the excess and deficiency of information bits due to the error correcting code being a degenerate code and due to the insertion of bits is calculated. Among the information bits of the last error correcting/detecting code of one
35 frame, the number of bits contained in the output code string 205, `last_info_len`, is `last_info_len = total_co`
`unt - last_check_len - pre_last_count - pre_last_chec`

k_len. When last_info_len is shorter than the information length of the error correcting code info_len, it is determined that the error correcting code is a degenerate code. In addition, when the counted value 1023 is in the range of from last_info_len to info_len, the switch 1021 is switched to a bit pattern outputted from an inserted bit generator 1015, to make up for the deficiency of information bits due to degeneracy. The output bit pattern from the inserted bit generator 1015 generates the same bit pattern as that of the inserted bit generator 705 of FIG. 7 of the encoder.

On the other hand, when last_info_len is longer than info_len, it is determined that the error correcting code is the inserted bit part, so that the information bits 912 are not outputted for the portion wherein the counted value 1023 is equal to or greater than info_len. With respect to the check bits, the switch 1002 is controlled so as to output, as check bits, the output code string 205 when $\text{total_count} - \text{check_len} < \text{bit_count} \leq \text{total_count}$.

The error correction/detection decoder 904 receives the information bits 912 and the check bits 913, which are outputted from the code string resolver 903, and decode the error correcting/detecting code on the basis of the FEC kind identifying signal 914 representative of the kind of the error correcting/detecting code latched by the latch circuit 1008 of FIG. 10, to output an error corrected code string 915 and an error detection signal 804.

The error corrected code string 915 is inputted to the inputted bit remover 905. The inserted bit remover 905 removes the inserted bits for preventing the pseudo synchronization signal inserted by the bit inserting unit 211 of the output coding unit 200. As described above, since the bit insertion is carried out only at the synchronization inserting position, the synchronization inserting position is determined on the basis of the counted value 911 of the counter 902.

For example, when the synchronization code word is

that shown in FIG. 5 and when the bit inserting unit 211 inserts bits into the portion "0000..." of the first sync_len bits of the synchronization code so that the humming distance from the synchronization code is greater than $2 * n + 1$, the number of "1"s (= n0) of "sync_0_len - (2 * n + 1)" bits from the synchronization code inserting position is counted. When the n0 is equal to or less than $2 * n + 1$, $2 * n + 1 - n0$ bits are removed. However, since it is determined that the inserted bit is "1", when the bit determined as the inserted bit by the inserted bit removing unit 905 is "0", it is considered that an error is mixed in the synchronization code inserting interval. In this case, the error detection signal 804 is outputted.

Thus, the code string 801 decoded by the input decoding unit 800 is inverse multiplexed by means of the demulti-plexer 811. This is operation for separating and output-ting the multiplexed code word as shown in FIG. 2. This inverse multiplexer 811 works with the first and second variable length decoders 806 and 810.

First, when the synchronization code detection signal 803 is inputted from the output decoding system 800, the demultiplexer 811 returns to the initial state of the frame process. Then, the kind of the error correcting/detecting code to the picture header is outputted as the FEC kind identifying signal 802 representative of the kind of the error correcting/detecting code, and the code string 801 is inputted to decode the picture header 302, so that it is determined whether there is no error in the picture header. When there is no error, the kind of the error correcting-/detecting code to the prediction mode information 303 is outputted as the FEC kind identifying signal 802, and the code string 801 is inputted to demultiplex the prediction mode information to output it to the second variable length decoder 810.

When all the prediction mode information is decoded, the second variable length decoder 810 outputs a

signal representative thereof to the demultiplexer 811. In response thereto, the demultiplexer 811 outputs a FEC kind identifying signal representative of the kind of an error correcting/detecting code to the motion vector
5 information 304, and starts the demultiplexing of the motion vector information 304. The demultiplexed motion vector information is outputted to the second variable length decoder 810 to be decoded. When the decoding of all the motion vector information is completed, a signal
10 representative thereof is outputted from the second variable length decoder 810 to the demultiplexer 811. In response thereto, the demultiplexer 811 outputs a FEC kind identifying signal representative of the kind of an error correcting/detecting code to the residual DCT coefficient,
15 and demultiplexes the residual DCT coefficient 305 which is outputted to the first variable length decoder 806.

As described above, the kind of the error correcting-/detecting code is determined on the basis of the multiplexing rule which is defined in the demultiplexer
20 811 so as to be the same as that of the output coding unit 200. Therefore, the output code string 205 is not required to contain the header information representative of the error correcting/detecting code and so forth.

In the error correction/detection decoder 904, it
25 may be detected by the error detection code that an error is mixed in the input code string 205'. In addition, as described above, an error of the inserted bits may be detected by the inserted bit removing unit 905. In these cases, the error detection code 804 is outputted from the
30 input decoding unit 800. Moreover, when a code word, which does not exist in the variable length code word table, is detected in the variable length decoding process, it is determined that an error is mixed. In addition, when it is determined that a part is against the multiplexing rule
35 in the demultiplexing process in the demultiplexer 811, it is determined that an error is mixed. In these cases, the input decoding unit 800 and the demultiplexer 811 carry out

the following processes so as not to greatly deteriorate the reconstructed image.

5 (1) When an error is detected in the residual DCT coefficient, the residue of the corresponding part is set to be 0. When an intra coding mode is selected as a prediction mode, the reconstructed image signal in the corresponding region may be predicted on the basis of the reconstructed image signal in the reconstructed frame and the surrounding region.

10 (2) In a case where an error is detected in the prediction mode information and the motion vector, when the prediction mode information and the motion vector information in the region, in which the error is detected, can be presumed on the basis of the prediction mode
15 information and the motion vector information in the surrounding region, the presumed information are used. When it is impossible, the reconstructed image signal in the corresponding region is predicted on the basis of the reconstructed image signal in the reconstructed frame and
20 the surrounding region.

(3) When an error is detected in a picture header, if the picture header is used as it is, there is probability that the picture quality is very greatly deteriorated. Therefore, the reconstructed image of the
25 last frame is used as the reconstructed image of the current frame.

In the aforementioned processes (1), (2) and (3), when the error has the influence on the subsequent codes to the next synchronization code because of the use of the
30 variable length coding, the same processes are carried out for that portion.

While it has been described that the synchronization code detector 901 detects synchronization codes only at the synchronization code inserting positions
35 (at intervals of sync_period bits), the loss of bits and/or the insertion of error bits may occur in a transmission/storage medium. In such a case, the

synchronization codes may be detected at positions other than the synchronization code inserting positions, and it may be determined that the positions, at which the synchronization codes are detected, are synchronization code inserting positions.

(Second Preferred Embodiment)

Referring to FIGS. 12 through 14, the second preferred embodiment of the present invention will be described below.

In this preferred embodiment, a dynamic image coding system and a dynamic image decoding system can surely detect synchronization even if a code string is transmit-ted/stored in a transmission line/storage medium in which the number of bits is decreased due to the loss of a part of a bit string or the number of bits is increased due to the addition of excessive bits.

FIG. 12 is a block diagram showing the principle of a process for detecting synchronization when such addition-/loss of bits occurs. It is assumed herein that the right synchronization code comprises "0"s of sync_0_len bits and a "1" of one bit as shown in FIG. 12(a). Furthermore, in FIG. 12, "x" denotes a bit other than a synchronization code.

FIGS. 12(b) through 12(e) show how the synchronization code is changed by the addition/loss of bits. It is assumed herein that the number of added/lost bits (Nid) is one bit at the maximum. FIG. 12(b) shows that one bit is eliminated from a bit string before a synchronization code, so that the whole synchronization code is shifted forwards by the one bit. FIG. 12(c) shows that one bit is added to a bit string before a synchronization code, so that the whole synchronization code is shifted backward by the one bit. FIG. 12(d) shows that a bit is eliminated from a synchronization code, so that the bits after the bit eliminating position indicated by the arrow are shifted forward by one bit. Moreover, FIG. 12(e) shows that one bit is added to a synchronization

code at the bit adding position indicated by the arrow, so that the bits after the bit adding position are shifted backwards by the one bit.

5 In order to correctly detect synchronization even
if the addition/loss of bits occurs, it is required to
determine that the bit strings shown in FIGS. 12(b) through
12(e) are synchronization codes. As can be seen from FIG.
12, the number of "1"s contained in the interval of $\pm N_{id}$
bits at the right synchronization code inserting positions
10 is $sync_0_len - 3*N_{id}$ bits at the maximum. Therefore,
synchronization may be detected in the interval of $\pm N_{id}$
bits at the synchronization code inserting positions on the
decoding side, and if the number of "1"s contained in this
range is less than the aforementioned value, it may be
15 determined to be a synchronization code. In addition, in
the coding system, code strings are transformed so that bit
patterns of FIGS. 12(b) through 12(d) do not occur.

20 Mainly with respect to the difference from the
first preferred embodiment, such coding/decoding systems
will be described below.

The whole construction of the second preferred
embodiment of a dynamic image coding system, according to
the present invention, is the same as that of the first
preferred embodiment of a dynamic image coding system,
25 except for the operation of the bit inserting unit 211 of
FIG. 3. FIG. 13 shows the operation of the bit inserting
unit 211. That is, in the bit inserting unit 211 in the
first preferred embodiment, the bit inserting operation is
carried out only at the synchronization code inserting
interval. On the other hand, in the bit inserting unit 211
30 in the second preferred embodiment, the bit insertion is
carried out in a synchronization code inserting interval
of $\pm N_{id}$ bits in order to prevent the same bit pattern as
the synchronization code from being produced even if the
addition/loss of N_{id} bits occurs at the maximum.
35

Assuming that the counted value 221 in FIG. 3 is
 $total_len$ and the interval between the adjacent

synchronization code inserting positions is `sync_period`, the bit inserting unit 221 counts the number of "1"s (assumed to be `=n0`) in the region of `total_len mod sync_period = sync_0_len - 1 - 3 * Nid` on the basis of
5 `total_len mod sync_period = sync_period - Nid` (mod: excess operation), and inserts "1"s of "`3 * Nid + 1 - n0`" if `n0` is less than `3 * Nid + 1`.

FIG. 13 shows an example of the operation of the bit inserting unit 211 when `sync_period = 12`, `sync_0_len = 9`, and `Nid = 1`. In this example, since `n0 = 2`, "1"s of
10 `3 * Nid + 1 - n0 = 2` bits are inserted.

By such bit insertion, it is ensured that the number of "0"s of $\pm Nid$ bits in the synchronization code inserting interval is equal to or greater than `3 * Nid`
15 bits, so that it is possible to uniformly identify synchronization codes.

On the other hand, the whole construction of the second preferred embodiment of a dynamic image decoding system, according to the present invention, is the same as
20 that of the first preferred embodiment, except for the operations of the synchronization detector 901 and the inserted bit remover 905 in FIG. 9. FIG. 14 shows the operation of an inserted bit remover 905.

That is, the synchronization detector 901 detects
25 synchronization codes in the range of $\pm Nid$ bits before and after the synchronization code inserting position, in order to detect synchronization even if the addition/loss of `Nid` bits occurs at the maximum.

First, it is determined whether synchronization
30 codes exist at every synchronization code inserting position. That is, assuming that the counted value 911 of the counter 902 is `bit_count`, the number of "0"s (assumed to be `= ns0`) in the range of `bit_count mod sync_period = sync_0_len - 1 + Nid` is counted on the basis of `bit_count mod sync_period = sync_period - Nid`. When `n0` is below `3 * Nid`, it is determined that a synchronization code exists
35 in this region.

FIG. 14 shows operation when $\text{sync_period} = 12$, $\text{sync_0_len} = 9$, and $\text{Nid} = 1$. In this example, the number of "0"s is counted in the range that $(\text{bit_count} \bmod \text{sync_period})$ is from "1" to "8". In the example of FIG. 14, since $\text{ns0} = 2$, it is determined that a synchronization code exists.

Then, it is determined what bits the code string is shifted by the addition/loss of bits in the synchronization code inserting interval, in which it is determined that a synchronization code exists. In the case of sync_0_len bits shown in FIG. 14, the shifted amount from the position of the last "1" is determined. Specifically, after searching a "1" which first appears from the $(\text{sync_0_len} + 1)$ th from the head of the synchronization code determining region, it is derived which bit the searched "1" is from the head of this synchronization code determining region (assuming that this is first_1_position bit). On the basis of the derived results, the shifted amount is derived from "the number of shifted bits = $\text{first_1_pos} - (\text{sync_0_len} + 1 + \text{Nid})$ " (forward shift in the case of negative, and backward shift in the case of positive). In the example of FIG. 14, since $\text{first_1_pos} = 10$, the number of shifted bits = $10 - (9 + 1 + 1) = -1$, so that it is found that the code string is shifted forward by one bit.

Unlike the first preferred embodiment, in the inserted bit remover 905, the inserted bit removing process is carried out in the interval of $\pm \text{Nid}$ bits before and after the synchronization code inserting position. That is, on the basis of $\text{bit_count} \bmod \text{sync_period} = \text{sync_period} - \text{Nid}$, the number of "1"s (assumed to be n0) is counted in the region of $\text{bit_count} \bmod \text{sync_period} = \text{sync_0_len} - 1 - 3 * \text{Nid}$. When the n0 is equal to or less than $3 * \text{Nid} + 1$, the "1"s of $3 * \text{Nid} + 1 - \text{n0}$ bits are removed.

In the second preferred embodiment, if the region, in which the addition/loss of bits occurs in a transmission

line or a storage medium, can be determined by some method, the synchronization detecting process, the bit inserting process and the bit removing process, which consider the addition/loss of bits, may be carried out only in that region.

Furthermore, also in the dynamic image decoding system in the first preferred embodiment as described above, the synchronization detection corresponding to the addition-/loss of bits in a transmission line and/or a storage medium is carried out. Therefore, the synchronization detector 901 may detect synchronization in the synchroni-zation code inserting interval, i.e., in the interval of $\pm N_{id}$ bits, similar to the second preferred embodiment. In this case, although a pseudo synchronization, which is erroneously determined to be a synchronization code while being a portion other than the synchronization code, may be produced, it is possible to inhibit the quality of a reproduced image from being deteriorated due to synchronization detection error in a transmission line and/or a storage medium which tends to cause the addition/loss of bits, so that the picture quality can be improved.

In addition, if the region, in which the addition/loss of bits occurs in a transmission line and/or a storage medium, can be determined by some process, that process may be carried out only in that region, and a usual synchronization detection may be carried out in other regions.

Moreover, in the first and second preferred embodiments as described above, the synchronization may further protected using information representative of the length of a frame (which will be hereinafter referred to as a "frame length information"). FIGS. 15, 16 and 17 show examples of code strings using a frame length information POINTER.

In the example of FIG. 15, a frame length information POINTER and a check bit CHKP of an error

correcting/detect-ing code for protecting the frame length information POINTER are arranged immediately after a synchronization code PSC. The number of bits of the last frame, i.e., information representative of the number of bits from the synchronization code of the last frame to the synchroni-zation code of the current frame, is stored in the frame length information POINTER.

The coding system counts the number of bits a code string of one frame to transform the counted number into a frame length POINTER, and carries out the error correction-/detection encoding to produce a check bit CHKP. Then, as shown in FIG. 15, the coding system produces a code string immediately after the synchronization code of the next frame.

On the other hand, after the decoding system detects synchronization codes by the same method as those in the first and second preferred embodiments, it takes the subsequent frame length information POINTER and the check bit CHKP out of the code string, and carries out the error correction/detection decoding to decode the frame length information POINTER. Then, the decoded frame length information POINTER is compared with a value derived by counting the number of bits from the last detected synchronization code to the current synchronization code (frame length counted value), to check whether there is no error detection of synchronization codes.

If the frame length counted value is different from the code length of the last frame indicated in the frame length information POINTER, there is probability that a synchroni-zation code is erroneously detected, so that the erroneously detected synchronization code is detected again using the frame length information POINTER. That is, it is assumed that a synchronization code, which could not be detected, exists before the current synchronization code by the number of bits indicated by the frame length information POINTER. In this case, the interval between the last detected synchronization code and the current

synchronization code is divided into two frames, i.e., the interval between the last synchronization code and the position indicated by the frame length information POINTER, and the interval between the position and the current synchronization code, to carry out the decoding process.

However, if the number of bits indicated by the frame length information POINTER is greater than the number of bits from the last detected synchronization code to the current synchronization code, it is assumed that the frame length information POINTER is erroneous, so that the aforementioned synchronization re-detecting process is not carried out.

If the numbers of bits of the frame length information POINTER and the check bit CHKP are great, the synchronization code PSC, the frame length information POINTER, and the check bit CHKP may extend in a plurality of synchronization intervals as shown in FIG. 16. In this case, the bit inserting process in the coding system and the bit removing process in the decoding system, which are performed in order to maintain a constant humming distance between a code string other than a synchronization code and the synchronization code, may not be carried out in an interval in which the frame length information POINTER and the check bit CHKP exist.

In the examples of FIGS. 15 and 16, in a case where the end half of a synchronization code PSC contains information representative of the kind of the synchronization code (the distinction between a frame synchronization code, a GOB synchronization code and so forth), the frame length information POINTER is not only protected, but the end half of the synchronization code PSC may be also protected by an error correcting code. Thus, since the kind of a synchronization code can be accurately detected in addition to the position of the synchronization code, the resistance to errors can be further improved.

In the example of FIG. 17, a frame length information POINTER and a check bit CHKP are arranged at

the end of a frame (immediately before a synchronization code in the next frame). In this case, after the decoding system detects the synchronization code in the next frame, it takes out a frame length information POINTER and a check bit CHKP immediately before the detected synchronization code to carry out the error correction/detection decoding, to carry out the re-detection of the synchronization code by the same processes as those of FIGS. 15 and 16.

In the example of FIG. 15, since the synchronization codes exist only at the synchronization code inserting positions, the frame length information POINTER to be recorded may be a value derived by dividing the number of bits of the frame by a synchronization code inserting interval (= sync_period bits). Thus, the frame length can be indicated by a smaller number of bits.

In addition, in the first and second preferred embodiments, while there has been shown examples of a hierarchical coding for changing an error correcting-/detecting code in accordance with the importance of the coded information, the same error correcting/detecting code may be used in the frame, or no error correcting/detecting code may be used. In such cases, it is possible to improve the capability to detect synchronization in comparison with conventional systems, by using the bit inserting process for maintaining a code string other than a synchronization code at a humming distance greater than a predetermined value from the synchronization code, and the synchronization code detecting process corresponding to the bit inserting process, as described in this preferred embodiment.

In addition, in the aforementioned preferred embodiments, while it has been shown that a dynamic image signal is high-efficiency compression encoded to be transmitted/stored, the present invention may be applied to the transmission/storage of a static image, voice, data or the like. For example, when the high-efficiency compression coding of a static image signal is carried out

using the orthogonal transform, the error correcting-
/detecting code may be switched so as to more strongly
carry out the error protection of a low component of a
transform coefficient. In the method for modeling and
5 encoding voice into a driving source and a vocal tract
filter, the error correcting/detecting code may be switched
so as to more strongly carry out the error protection of
a pitch cycle, a vocal tract parameter and so forth.

(Third Preferred Embodiment)

10 The third preferred embodiment of the present
invention will be described below. In this preferred
embodiment, the error correcting/detecting code is not
used. At this point, this preferred embodiment is
different from the first and second preferred embodiments.

15 FIG. 18 is a block diagram of a dynamic image
coding system in this preferred embodiment. Using the same
reference numbers as those in FIG. 1 for the parts
corresponding to those in FIG. 1, the different points from
the first preferred embodiment will be mainly described.
20 In this preferred embodiment, the construction and
operation of an output coding unit 200 are different. In
addition, although the basic operation of a multiplexer 111
is the same as the multiplexer 111 of FIG. 1, the
multiplexer 111 in this preferred embodiment outputs only
25 a multiplexed code string 201 and a synchronization code
insertion request signal 203 since the error correcting-
/detecting code is not used.

FIG. 19 is a block diagram of the output coding
unit 200 of FIG. 18. The output coding unit 200 comprises
30 a counter 1701 for counting the number of bits of an output
code string 205, a switch 1703 for switching the output
code string 205, a switch controller 1704 for controlling
the switch 1703, and a stuffing bit generator 1705 for
generating stuffing bits.

35 FIG. 20 shows an example of an output code string
205 generated by the output coding unit 200 of FIG. 19.
The same signs as those in FIG. 4 are used for code words

corresponding to those of the output code string of FIG. 4. Similar to FIG. 4, each of synchronization codes PSC is inserted only at any one of synchronization code inserting positions, which are arranged periodically, i.e.,
5 at regular intervals (sync_period bits) and which are indicated by arrows. FIG. 20 does not contain the check bit CHK for the error correcting/detecting code. At this point, FIG. 20 is different from FIG. 4. A stuffing bit STUFF is inserted into the end part of one frame of the
10 output code string 205 so that a synchronization code PSC is inserted at the synchronization code inserting position. The number of bits of the stuffing bit STUFF is equal to or less than sync_period.

The construction and operation of the output coding
15 unit 200 of FIG. 19 for producing such an output code string 205 will be described in detail below.

The counter 1701 is set to be "1" when a synchronization code insertion request signal 203 is inputted from a multiplexer 111 thereto and an initial bit
20 of a synchronization code 301 is inputted thereto as a multiplexed code string 201, and it is set to be a synchronization code length sync_len when all the bits of the synchronization code 301 are inputted thereto. Thereafter, the counter 1701 counts up sequentially from
25 the next bit of the synchronization code 301 until a bit immediately before the next synchronization code is outputted.

When bits from the initial bit of the synchronization code to the bit before the next
30 synchronization code are inputted as a multiplexed code string 201, The switch controller 1704 controls the switch 1703 so as to switch the switch 1703 to the multiplexed code string 201 to output the multiplexed code string 201 as an output code string 205.

35 Then, at the last part of one frame, the bit insertion (bit stuffing) is carried out so that the next synchronization code is inserted at the synchronization

code inserting position. The multiplexer 111 output a synchronization code insertion request signal 203 for the next frame when the output of the multiplexed code string 201 of one frame is completed. In response thereto, the switch controller 1704 switches the switch 1703 to the stuffing bit generator 1705 to output a stuffing bit 1223 as an output code string 205. The stuffing bit 1223 may have all the bits of "1" or "0", or a specific pattern such as "0101...".

This preferred embodiment of a dynamic image decoding system, according to the present invention, will be described below.

FIG. 21 is a block diagram of a dynamic image decoding system corresponding to the dynamic image coding system of FIG. 18. Using the same reference numbers as those in FIG. 8 for parts corresponding to those in FIG. 8, the different points from the first preferred embodiment will be mainly described. In this preferred embodiment, the construction and operation of an input coding system 800 are different from those in the first preferred embodiment. In addition, signals inputted from the input decoding unit 800 to a demultiplexer 811 are only a code string 801 and a synchronization code detection signal 803, and no signal is inputted from the demultiplexer 811 to the input decoding unit 800.

FIG. 22 is a block diagram of the input decoding unit 800. The input decoding unit 800 comprises a synchronization detector 1901 for detecting a synchronization code in an input code string 205', and a counter 1902 for counting the number of bits of the input code string 205'.

The counter 1902 is reset to be "0" at the initial stage of decoding, and counts up a counted value 1911 for each "1" every time one bit of the input code string 205' is inputted.

The synchronization detector 1901 detects synchronization codes only at the synchronization code

inserting positions on the basis of the counted value 1991 of the counter 1902. For example, assuming that the synchroni-zation code inserting interval is `sync_period`, the counted value 1911 is `bit_count`, and the length of the
5 synchroni-zation code is `sync_len`, the synchronization detection is carried out only when $0 < \text{bit_count} \bmod \text{sync_period} \leq \text{sync_len}$, wherein $A \bmod B$ denotes a remainder when A is divided by B. The synchronization detector 1901 outputs a synchronization code detection
10 signal 803 when a synchroni-zation code is detected.

The code string 801 from the input decoding unit is inputted to the multiplexer 811 while the input code string 205' is outputted as it is. Thereafter, similar to the dynamic image decoding system of FIG. 21, the
15 demultiplexing and decoding processes are carried out.

In a case where the last stuffing bit STUFF of the frame is a predetermined bit pattern, it is determined whether the stuffing bit STUFF corresponds to a predetermined pattern in the demultiplexer 811. When it
20 does not correspond thereto, it is determined that there is an error in the input code string 205', so that the process for preventing the picture quality from being greatly deteriorated, which has been described with respect to the dynamic image coding system in the first preferred
25 embodiment, may be carried out.

(Fourth Preferred Embodiment)

The fourth preferred embodiment of the present invention will be described below.

In this preferred embodiment, the whole
30 construction of a dynamic image coding system is the same as that of the dynamic image coding system of FIG. 18, and the operation of an output coding unit is different from that in the third preferred embodiment.

FIG. 23 is a block diagram of an output coding unit
35 200 in FIG. 18. Using the same reference numbers for parts corresponding to those of the output coding unit of FIG. 19, the difference from the output coding unit of FIG. 19

is that a bit inserting unit 1211 for carrying out the bit stuffing process for preventing a pseudo synchronization code is added.

5 In the bit inserting unit 1211, the bit insertion for preventing a pseudo synchronization for occurring is carried out for the multiplexed code string 201. Since it is not possible to uniformly decode a synchronization code if the same bit pattern as the synchronization code is contained in the output code string 205, the bit insertion is carried out in order to prevent this. For example, as shown in FIG. 5, when the synchronization code comprises 10 "0"s of `sync_0_len` bits, a "1" of one bit and "xxxxx" of `sync_nb_len` bits representative of the kind of the synchronization code, it is possible to prevent a pseudo synchronization from occurring if a "1" is inserted so that 15 "0"s of `sync_0_len` bits or more are not continuously arranged in the code string other than the synchronization code.

The synchronization code is inserted only into the 20 synchronization code inserting system. Therefore, the bit inserting operation for preventing the pseudo synchronization from occurring may be carried out only at the synchronization code inserting positions. Thus, it is determined whether it is required to carry out the bit insertion on 25 the basis of a counted value 1221 representative of the total number of bits of an output code string 205. Assuming that the counted value 1221 is `total_len`, the number of "1"s in the multiplexed code string 201 is counted in an interval of $0 < \text{total_len_mod_sync_period} \leq \text{sync_0_len}$. If no "1" exists in this interval, a "1" 30 of one bit is inserted. Herein, $A \bmod B$ denotes a remainder when A is divided by B.

In addition, in order to decrease the probability of the error detection of a synchronization code due to 35 error, the bit insertion may be carried out as follows.

In order to detect a synchronization codes even if an error of n bits is mixed into the synchronization code,

it is required to determine that a code word having a hamming distance of n or less from a true synchronization code in an input decoding unit of a dynamic image decoding system as described later is a synchronization code.

5 However, if such a determination is carried out while the code strings other than the synchronization code are as they are, bit patterns having a hamming distance of n or less from the synchronization code may exist even in the code strings other than the synchronization code. If this

10 is arranged at the synchronization code inserting position, it may be erroneously determined to be the synchronization code.

Therefore, the following bit insertion into a multiplexed code string 201 is carried out by means of a

15 bit inserting unit 211, so that the code strings other than the synchronization codes arranged at the synchronization code inserting positions in the multiplexed code string 201 are transformed so as to have a humming distance of $2 \times n + 1$ or more from the synchronization code. Specifically,

20 the number of "1"s (assumed to be n_0) is counted in an interval in which $0 < \text{total_len} \bmod \text{sync_period} \leq \text{sync_0_len} - (2 \times N + 1)$. If n_0 is less than $2 \times n + 1$, "1"s of $2 \times n + 1 - n_0$ bits are inserted into the multiplexed code string 201.

25 With respect to a code string 1222, in which the bit insertion has been thus carried out, the bit insertion (STUFF in FIG. 20) is carried out in the last interval of the frame similar to the output coding unit of FIG. 19, to be outputted as an output code string 205.

30 This preferred embodiment of a dynamic image decoding system, according to the present invention, will be described below. The whole construction of this dynamic image decoding system is the same as that of the dynamic image decoding system of FIG. 21, and the operation of an

35 input coding system 800 thereof is different from that in the third preferred embodiment.

FIG. 24 is a block diagram of an input decoding

unit 800. Using the same reference numbers for parts corresponding to those of an input decoding unit of FIG. 22, the difference between this preferred embodiment and the third preferred embodiment will be mainly described. In this preferred embodiment, a bit removing unit 1905 is added.

5 An input code string 205 is inputted to an inserted bit remover 1905, and a process for removing inserted bits is carried out in order to prevent a pseudo synchronization code inserted by a bit inserting unit 1211 of the output coding unit of FIG. 23. As described above, since the bit insertion is carried out only at the synchronization code inserting positions, the synchronization code inserting positions are determined on the basis of a counted value 1911 of a counter 1902.

10 For example, assuming that the synchronization code is a code word shown in FIG. 5 and when the bit inserting unit 1211 carries out the bit insertion into the initial portion "0000..." of the synchronization code so that the humming distance from the synchronization code is greater than $2 \times n + 1$, the number of "1"s of sync_0_len - ($2 \times n + 1$) bits from the synchronization code inserting position is counted. When the n_0 is less than $2 \times n + 1$, $2 \times n + 1 - n_0$ bits are removed.

15 25 Since it is determined that the inserted bit is "1", when the bit determined to be the inserted bit is "0", it is considered that an error is mixed in the synchronization code inserting interval. In this case, an error detection signal (not shown) may be outputted to a demultiplexer 811, and the same process as that in the first preferred embodiment may be carried out so as to prevent a reproduced image from being greatly deteriorated.

30 35 The bit inserting process by the bit inserting unit 1211 of FIG. 23 may be carried out so that the inserted bits of a predetermined number of bits are inserted into all the synchronization code inserting intervals other than the synchronization code. FIG. 25 shows an example of an

output code string 205 when such a bit inserting process is carried out. In the drawing, SB denotes an inserted bit.

5 For example, as shown in FIG. 5, when a synchronization code comprises "0"s of `sync_0_len` bits, a "1" of one bit and "xxxxx" of `sync_nb_len` bits representative of the kind of the synchronization code, an inserted bit SB of one bit is inserted at a predetermined position of an interval of `sync_0_len` bits from the head of the synchronization code inserting interval.
10

The inserted bit SB may be always "1". In addition, in accordance with a bit pattern in an interval of `sync_0_len` bits from the head of the synchronization code inserting interval, the inserted bit SB may be suitably determined so that the number of "1"s in the interval is equal to or greater than 1.
15

Moreover, the inserted bit SB may be an odd parity in an interval of `sync_0_len` bits from the head of the synchronization code inserting interval, to prevent the same bit pattern as the synchronization code from occurring and to detect an error mixed into this bit pattern.
20

FIG. 25(b) shows an example of an output code string, in which such a bit inserting process is carried out. In this example, an inserted bit SB of 1 bit is inserted into the initial portion from the synchronization code inserting position. This inserted bit SB is determined so that the number of "1"s in an interval of the inserted bit SB and `sync_0_len - 1` bits from the next bit is always an odd number. For example, in the left example of FIG. 25(b), the inserted bit SB is "1". In addition, in the right example of FIG. 25(b), since the inserted bit SB is "1" even if all the bits are "0" in the interval of `sync_0_len - 1` bits from the next bit to the inserted bit SB, "1"s of 1 bit or more always exist in the synchronization code inserting interval, so that the same bit pattern as the synchronization code is not produced.
25
30
35 In addition, since the inserted bit SB serves as a parity

check, it is possible to detect a bit error mixed into this interval.

5 In addition, the inserted bit SB may be an odd parity check bit for all the bits before the next synchronization code inserting position. However, in order to prevent the same bit pattern as the synchronization code from occurring only when all of sync_0_len - 1 bits from the next bit to the inserted bit SB are "0", the inserted bit SB is always set to be "1". Thus, it is possible to carry out the error detection by the parity check of all bits.

10 In order to decrease the probability of the error detection of a synchronization code due to error, it is desired to insert more bits. For example, in order to correctly detect synchronization even if an error of n bits is mixed, "1"s of $2 \times n + 1$ bits are inserted into a predetermined position in this interval.

15 In this preferred embodiment, the operation of the bit removing unit 1905 of FIG. 24 is different so as to correspond to the operation of the aforementioned bit inserting unit 1211. That is, the bit removing unit 1905 removes the inserted bit SB at a predetermined position, at which the bit insertion has been carried out by the bit inserting unit 1211.

20 In a case where the inserted bit SB is always "1", it is determined that a bit error exists when a bit at the bit inserting position in the input code string 205 is "0", so that an error detection signal (not shown) may be outputted to the inverse multiplexer 811 so as to prevent a reproduced image from being greatly deteriorated.

25 In the first through fourth preferred embodiments, while the prediction mode information 303, the motion vector information 304 and the residual DCT coefficient 305 have been multiplexed in the multiplexer 111 for each coding frame as shown in FIG. 2, the predication mode information 303, the motion vector information 304 and the residual DCT coefficient 305 may be multiplexed for each

coding region (e.g., a macro block, GOB) as shown in FIG. 26. In this case, the picture header 302 and the other information may have different error correcting/detecting code, or the same error correcting/detecting code.

5 Alternatively, an error correcting/detecting code may be used for only the picture header or for a part of a code string of a predetermined number of bits of each frame, or no error correcting/detecting code may be used.

In addition, the multiplexing may not be carried out only for each frame (picture), but it may be also carried out for each part of a frame or for each layer of a plurality of frames, so that a synchronization code may be inserted for each of these multiplexing units (layer unit).

10

FIG. 28 shows examples of such multiplexing. In the examples of FIG. 28, the multiplexing processes are carried out for each of four layers, i.e., each macro block layer of a plurality of coding blocks, each GOB layer of a plurality of macro blocks, each picture (frame) layer, and each session layer of a plurality of pictures. Among these layers, the session, picture and GOB layers use their synchronization signals (SSC, SEC, PSC and GSC in the drawing), respectively. Different codes are used for the SSC, SEC, PSC and GSC so that it is possible to identify which layer the synchronization code is detected. When the synchronization code shown in FIG. 5 is used, these synchronization codes may be distinguished by the portion of sync_nb_len bits representative of the kind of the synchronization code.

15

20

25

Also when such multiplexing is carried out, the same processes as those for the frame synchronization code in the preceding preferred embodiments may be carried out for a part or all of the synchronization codes of the session, picture or GOB. FIG. 29 shows an example of an output code string, in which such a process is carried out. As shown in FIG. 29, stuffing bits STUFF are inserted before the PSC and the GSC, and the SSC, PSC and GSC are

30

35

inserted at the synchronization code inserting positions indicated by arrows in the drawing. Therefore, similar to the afore-mentioned preferred embodiment which shows the frame synchronization code PSC, the detection accuracy of each synchronization code can be improved.

The same length information as the frame length information POINTER of FIGS. 15, 16 and 17 may be added to each synchronization code of the session, picture and GOB. In a case where the frame length information POINTER is protected by the error correcting/detecting code as shown in FIGS. 15 and 16, if the error correcting/detecting code is not used only for the frame length information POINTER, but it is also used for a portion of sync_nb_len bits representative of the kind of a synchronization code, it is possible to improve the probability that the kind of the synchronization code, in addition to the position thereof, can be correctly detected. In addition, a part or all of the header information (SH, PH and GF in the drawing) of the session, picture and GOB may be protected using the error correcting/detecting code, so that it is possible to improve the error resistance of each header information.

In a case where the stuffing process for preventing a pseudo synchronization code is carried out as this preferred embodiment, the following process may be carried out so that the synchronization code inserting interval sync_period is equal to or less than the length of the synchronization code.

First, the process in the output coding unit of the dynamic image coding system will be described. It is assumed herein that the synchronization code is a code word comprising "0"s of sync_0_len bits and a "1" of 1 bit as shown in FIG. 5. In the output coding unit of FIG. 23, assuming that the counted value 1221 representative of the number of bits outputted from the bit inserting unit 1211 is total_len, when a remainder of division of total_len by the synchronization code inserting interval sync_period is equal to a remainder of division of a value, which is

derived by subtracting 1 from the number of bits `sync_0__len` of the initial "0" of the synchronization code, by `sync_period`, i.e., when

$$\begin{aligned} & \text{total_len mod sync_period} \\ 5 \quad & = (\text{sync_0_len} - 1) \text{ mod sync_period} \quad (4) \end{aligned}$$

the number of "1"s (assumed to be `n1`) in the output bits before $(\text{sync_0_len} - 1)$ bits from the output bit at that time is counted, and a "1" of 1 bit is inserted if no "1" exists (i.e., if `n1` = 0).

FIG. 33(a) shows an example of an output code string, for which such a process has been carried out. In the drawing, each of downward arrows indicates a synchronization code inserting position, and a synchronization code comprises "0"s of 23 bits (i.e., `sync_0__len` = 23), and a "1" of 1 bit. In the shown example, a synchronization code inserting interval `sync_period` is 8, which is shorter than the length of the synchronization code (= 24 bits).

In the drawing, intervals 1 through 4 denote intervals for counting the aforementioned `n1`. In each interval, the number of "1"s `n1` is sequentially counted. If `n1` = 0, a stuffing bit is inserted into the next bit of the interval. Since `n1` > 0 in interval 1, it is not required to insert the stuffing bit. Since `n1` = 0 in interval 2, a stuffing bit 3301 of 1 bit is inserted into the next interval. In interval 3, `n1` = 1 due to the inserted stuffing bit 3301, so that it is not required to insert a stuffing bit.

If such a bit stuffing process is carried out, the same bit pattern as a synchronization code does not exist at a portion other than the synchronization code in an output code string, so that no pseudo synchronization occurs.

On the other hand, in order to decrease the probability that a synchronization code is erroneously detected due to transmission line error, a bit insertion

may be carried out as follows.

Even if an error of n bits enters a synchronization code, the synchronization code can be correctly detected by carrying out a bit inserting process so that a humming distance between a portion other than the synchronization code and the synchronization code in an output code string in a bit insertion 1211 is equal to or greater than $2 \times n + 1$.

In this process, when a remainder derived by dividing a counted value 1221 $total_len$ representative of the total number of bits of the output code string 205 of FIG. 23 by a synchronization code inserting interval $sync_period$ is coincident with a remainder derived by dividing a value, which is derived by subtracting $2 \times n + 1$ from the initial number of bits of "0"s of a synchronization code, by $sync_period$, i.e., when

$$\begin{aligned} & total_len \bmod sync_period \\ &= (sync_0_len - (2 \times n + 1)) \bmod sync_period \\ &(2) \end{aligned}$$

the number of "1"s (assumed to be n_1) in output bits of $(sync_0_len - (2 \times n + 1))$ bits from the output bit at that time. If the number of "1"s is less than $(2 \times n + 1)$ bits, i.e., if $n_1 < 2 \times n + 1$, "1"s of $(2 \times n + 1 - n_1)$ bits are inserted.

As shown in FIG. 5, when a synchronization code starting from "0"s of a plurality of bits is used, if the number of "1"s in a bit string immediately before the synchronization code is insufficient, a synchronization error detection may occur at that portion. In order to prevent this, a bit insertion (STUFF in FIG. 20) in the last interval of a frame may be outputted so that the number of bits of "1"s in an interval of $sync_period$ bits from the synchronization code to the synchronization code inserting position immediately before the synchronization code is equal to or greater than $2 \times n + 1$ bits.

In order to do this, a STUFF always containing "1"s of $2 \times n + 1$ bits may be used, or the STUFF may be

determined in accordance with an output code string. That is, the STUFF may be determined so that the number of bits of "1"s of sync_period bits immediately before the synchronization code in the output code string containing the STUFF is equal to or greater than $2 \times n + 1$ bits.

FIG. 33(b) shows an example of an output code string, for which such a process has been carried out. In the drawing, intervals 1 through 4 denote intervals for counting the aforementioned n_1 . In each interval, the number of "1"s n_1 is sequentially counted. If $n_1 < 2 \times n + 1$, a stuffing bit is inserted into the next bit of that interval. Since $n_1 = 1$ in interval 2, stuffing bits 3311 of $(2 \times n + 1) - 1 = 2$ bits are inserted into the next interval. In interval 3, $n_1 = 3$ due to the inserted stuffing bits 3311, so that it is not required to insert stuffing bits.

Moreover, in order to prevent a synchronization code error detection immediately before a synchronization code, a STUFF is determined as follows. It is assumed that a bit immediately before the STUFF is 3311. Since the number of "1"s in an interval (interval 5) of sync_period bits from a synchronization code inserting position immediately before the bit 3312 to a synchronization code inserting position immediately after the bit 3312 is only 1 bit, an synchronization error detection may occur in this portion if the synchronization code has a plurality of "0" continuously arranged as shown in FIG. 5. Therefore, the position, at which the synchronization code is inserted, is shifted to the next synchronization code inserting position, and a STUFF 3313 containing many "1"s is outputted. Thus, since "1"s of $2 \times n + 1$ or more are contained in an interval (interval 6) of sync_period bits immediately before the synchronization code, it is possible to prevent the synchronization error detection.

If such a bit stuffing process is carried out, the humming distance from the synchronization code can be $2 \times n + 1$ or more at a portion other than the synchronization

code in the output code string, so that the probability of the synchronization error detection can be decreased.

Processes in an output decoding system of a dynamic image decoding system will be described below. In the bit remover 1905 of FIG. 24, assuming that the counted value 1905 representative of the number of bits of an input code string is `total_len`, and when the `total_len` meets the conditions in which it is 1, the number of "1"s (assumed to be `nl`) in input bits of (`sync_0_len` - 1) bits from the input bit at that time is counted. If no "1" exists, i.e., if `nl` = 0, 1 bit is removed.

In order to detect a synchronization code even if an error of n bits is mixed into the synchronization code, in a case where a bit inserting process is carried out by the bit inserting unit 1211 so that the humming distance between a portion other than the synchronization code and the synchronization code in an output code string is equal to or greater than $2 \times n + 1$, the following process may be carried out. When the `total_len` reaches a value meeting formula (2), the number of "1"s (assumed to be `nl`) of (`sync_0_len` - ($2 \times n + 1$)) bits from the input bit at that time in the output bits is counted. When the number of "1"s is less than ($2 \times n + 1$), i.e., when $nl < (2 \times n + 1)$, ($2 \times n + 1 - nl$) bits are removed.

In the output coding unit and the input decoding unit, if the aforementioned processes are carried out so that the synchronization code inserting interval `sync_period` has a smaller number of bits than the length of the synchronization code, the number of bits of the stuffing bits STUFF can be decreased, so that the coding efficiency can be improved. In particular, when a synchronization code is long, or when many synchronization codes are inserted, the degree of the enhanced coding efficiency due to the decrease of the number of bits of the stuffing bits STUFF is great. For example, in a system for dividing a screen into one or a plurality of macro blocks or macro block lines to insert synchronization codes into each unit,

such as a GOB/slice in a dynamic image coding, many dynamic image patterns are inserted, so that the degree of the coding efficiency due to the decrease of the number of bits of the STUFF is increased.

5 In addition, when a structure having a plurality of layers is multiplexed as shown in FIG. 28, synchronization codes having different lengths in accordance with layers may be used.

10 FIG. 34(a) shows examples of such synchronization codes. Among four kinds of synchronization codes, each of SSC, SEC and PSC has 32 bits, which comprises "0"s of 23 bits, a "1" of 1 bit and 8 bits representative of the kind of the synchronization code. On the other hand, a synchronization code GSC of a GOB layer is a synchronization code
15 of 17 bits, which comprises "0"s of 16 bits and a "1" of 1 bit, and is a shorter code word than other synchronization codes.

 The reason why only the GSC is such a shorter code word is as follows. The GOB is a coded unit, which
20 comprises one or a plurality of macro blocks (MB) and which is formed by dividing a screen into small regions, so that there are generally more synchronization codes of the GOB layer than other synchronization codes. Therefore, if the synchroni-zation code length is decreased, the code amount
25 of the output code string can be decreased. In addition, it is possible to output more GSCs if the code amount is the same, and it is possible to divide the screen into small GOB regions for encoding, so that the quality of a reproduced image can be improved when a transmission line
30 error occurs.

 A process for preventing a pseudo synchronization as described in the fourth preferred embodiment, i.e., a bit stuffing process for preventing the same bit pattern as a synchronization code from being produced in a code
35 string other than the synchronization code, may be carried out. If a bit stuffing process for decreasing the probability of error detection of a synchronization code

due to a transmission line error, e.g., a bit stuffing process for preventing the same bit pattern as a shortest synchroniza-tion code (GSC in the example of FIG. 34(a)) from being produced in a bit string, for which it is ensured only that the same bit pattern as a long bit length of synchronization code (SSC, SEC, PSC in the example of FIG. 34(a)) is not produced, is carried out, it is possible to prevent the same bit patterns as all the synchronization codes from being produced. This process may be carried out for the code strings of all the layers, or for code strings of lower layers (GOB layer, macro block layer in the shown example) than a layer using the shortest code, or for code strings of layers (picture layer, GOB layer, macro block layer) below a layer immediately above that layer. Alternatively, this process may be carried out only for code strings of a predetermined layer.

In order to easily identify synchronization codes of different lengths even if a transmission line error occurs, the process of a synchronization code or before and after the synchronization code may be carried out as follows.

(i) In a case where a synchronization code comprising a plurality of bits of "0"s and the subsequent "1" is used, the relative positions of a long code word and a short code word from the synchronization code inserting position of the "1" may be different. In the example of FIG. 34(b), the "1" 3211 of the PSC and the "1" 3412 of the GSC are arranged at different position, and all of bits (3413 to 3411, 3414 to 3412) arranged at the same positions in other synchronization codes are "0". Thus, since the humming distance of the synchronization code and a partial code string thereof is increased, it is possible to easily identify different synchronization codes even if a transmission line error occurs.

(ii) A stuffing bit may be inserted before a short synchronization code. For example, if a stuffing bit 3401 comprising one or a plurality of "1"s is inserted before

a short GSC, the humming distance between the OSC and a partial code string of another synchronization code can be increased.

5 (iii) A stuffing bit may be inserted after a short synchronization code. For example, a bit insertion 3402 may be carried out after a GSC so as to increase the humming distance from a portion identifying the kind of a synchronization code in long synchronization codes.

(Fifth Preferred Embodiment)

10 The fifth preferred embodiment of the present invention will be described below.

In this preferred embodiment, the whole constructions of a dynamic image coding system and a dynamic image decoding system are the same as those in the
15 first preferred embodiment, except for the processes carried out at the head and end portions of a synchronization interval by an output coding unit 200 and an input decoding unit 800.

FIGS. 27(a), 27(b) and 27(c) are examples of an
20 output code string 205 of a dynamic image coding system in this preferred embodiment. In this output code string 205, a part 2701 of a code string of the last frame (frame n-1) is arranged after a synchronization code PSC, and a pointer information 2702 (SA) representative of a boundary 2703 (a
25 start point of a code string of the current frame) between the code string 2701 and the current frame (n frame), i.e., the boundary between multiplexed code strings, is arranged, so that the stuffing bit (STUFF in FIG. 4) does not exist at the last of one frame. At this point, this output code
30 string 205 is different from the output code string of FIG. 4.

In the output coding unit 200 of the dynamic image coding system, the number of bits resid_bit of the residual code strings of the frame is checked at each
35 synchronization code inserting position. When the sum of the resid_bit and the numbers of bits of a synchronization code PSC and a pointer information SA is less than synchro-

nization code inserting interval sync_period bits, the synchronization code PSC is outputted before the residual code strings of the frame are outputted to the output code string 205. Then, the pointer information SA (representative of the resid_bit in this case) is outputted, and then, the residual code strings 2701 are outputted. Thereafter, the code strings of the next frame are outputted.

In the input coding system 800 of the dynamic image decoding system, a synchronization code is detected at each of synchronization code inserting positions. When the synchronization code is detected, it is determined that the pointer information SA and the residual information of the frame are arranged after the detected synchronization code, and the subsequent process is carried out.

For example, the boundary between a frame n-1 and a frame n in FIG. 27 will be described. After the decoding process immediately before 2704 the synchronization code PSC is completed, a synchronization code is detected in the subsequent synchronization code inserting interval. When the synchronization code is detected, the pointer information 2702 is decoded, and it is derived which bits the code strings of the frame n-1 have. On the basis of this, the bits of the number of bits indicated by the pointer information are taken out of the code string immediately after the pointer information (to 2703 in FIG. 27), and the code string 801 is outputted so that these are arranged after 2704.

In this preferred embodiment, the error correction-/detection coding of a part or all of the output code strings may be carried out as shown in FIG. 27(a). In this case, all the kinds of the error correcting/detecting codes may be the same or different.

In addition, as shown in FIG. 27(b), the error correction/detection coding may not be carried out.

Moreover, as shown in FIG. 27(c), the frame length information POINTER representative of the number of bits of code strings of one frame as shown in FIGS. 15 and 16

may be inserted. In this case, the frame length information POINTER may be representative of the number of bits from the synchronization code PSC of the frame to a synchronization code PSC of the next frame.

5 In a case where the error correction/detection coding is carried out as shown in FIG. 27(a), the pointer information SA, the residual code strings 2701 of the frame n-1, and the code strings of the frame n after 2703 are combined to be one information bit, and the error
10 correction/detection coding of the combined information bit is carried out.

The pointer information SA may be information, for which the error correction/detection coding has been carried out. In this case, the synchronization code PSC
15 (or a part thereof), the frame length information POINTER and the pointer information may be combined to carry out the error correction/detection coding.

Examples of stuffing bits STUFF will be described below.

20 FIGS. 30(a) and 30(b) shows examples of code tables of stuffing bits STUFF as examples of the aforementioned stuffing bits STUFF. Both of FIGS. 30a) and 30(b) are characterized in that the decoding can be uniformly carried out in the backward direction to an output code string, so
25 that the starting position of the stuffing bits STUFF can be uniformly identified. Therefore, an error mixed into a code string can be detected by comparing the decoding end position of a code string immediately before the stuffing bit STUFF with the starting position of the stuffing bit
30 STUFF, and the starting point of the backward decoding can be identified when a coding system for decoding in a backward direction from a synchronization code is used.

Moreover, in the stuffing bits STUFF shown in the code tables of FIGS. 30(a) and 30(b), the first bit is
35 always "0", so that the error detection can be carried out by a simplified decoding as described later.

FIG. 31 shows an example of a decoding process of

a code string, which contains the stuffing bits STUFF shown in the code tables of FIGS. 30(a) and 30(b). Although FIG. 31 shows an example of the stuffing bit immediately before the synchronization code inserting position, the same process can be carried out by inserting a stuffing bit immediately before another optional synchronization code inserting position. In FIG. 31, arrows 3101 through 3103 denote examples of decoding end positions of a code string (indicated by "xxx...") immediately before the stuffing bit STUFF when the decoding is carried out in a forward direction, and the right end of each of the arrows indicates the decoding end position. When no error is mixed into a code string and the decoding is normally carried out, the decoding end position of a code string immediately before the stuffing bit STUFF is coincident with the starting position of the stuffing bit STUFF as shown by the arrow 3101.

On the other hand, when an error is mixed into a code string, the decoding end position of a code string immediately before the stuffing bit STUFF is shifted from the starting position of the stuffing bit STUFF as shown by the arrows 3102 and 3103. In such a case, it is determined that an error exists in the code string.

In the decoding system, when the decoding of a code string immediately before the stuffing bit STUFF is completed, the stuffing bit STUFF before the next synchronization code inserting position is read, and it is determined whether the read stuffing bit STUFF is coincident with the codes in the code table shown in FIGS. 30(a) and 30(b). If the stuffing bit STUFF is not coincident with any codes, it is determined that an error exist.

When it is determined whether the stuffing bit STUFF is coincident with the code table, a small bit error may be allowable. Thus, it is possible to decrease the error detection of an error when an error is mixed into the stuffing bit STUFF itself.

The code table of FIG. 30(a) always starts from "0" and has the subsequent bits of "1"s. Therefore, in order to carry out the error detection, it may be determined only whether the next bit to the decoding end position of the code string immediately before the stuffing bit STUFF is "0", or the error detection may be carried out only by the first "0" and the subsequent some "1"s. Thus, although the error detection accuracy is slightly lowered, the throughput required to decoding can be decreased. Thus, in a case where a code table starting from a specific bit pattern, in which all the stuffing bits STUFF comprise a specific bit or a plurality of bits, is used, the decoding process can be simplified.

Moreover, the stuffing bits STUFF shown in the code tables of FIG. 30(a) and 30(b) contain a lot of bits of "1", and the humming distances from the synchronization code containing a lot of "0"s and a part thereof are great as shown in FIG. 5, so that there is an advantage in that the probability that a pseudo synchronization occurs is low. Specifically, in the code table of FIG. 30(a), all of only the first bits of the stuffing bits STUFF are "0", and all of the other bits are "0", so that the humming distance between the synchronization codes, all of which are "0", and a part thereof is (the length of the stuffing bit STUFF is -1). In addition, in the code table of FIG. 30(b), only the first and last bits of the stuffing bits STUFF are "0", and all the other bits are "1", so that the humming distances from the synchronization code and a part thereof are (the length of the stuffing bit STUFF is -2). Thus, if the humming distances between the stuffing bit STUFF, and the synchronization code and a part thereof are selected so as to be greater than a predetermined value, e.g., (the length of the stuffing bit STUFF -2), so that it is difficult to produce a pseudo synchronization code even if an error is mixed into a code string.

Referring to FIG. 32, this effect will be described. FIGS. 32(a-0) and 32(b-0) show examples of code

strings when usual stuffing bits (all the bits are "0") and the stuffing bits STUFF shown in the code table of FIG. 30(a) are used, and FIGS. 32(a-1) and 32(b-1) show examples when an error of 1 bit is mixed in FIGS. 32(a-0) and 32(b-0), respectively. As can be seen from FIG. 32(a-1), if only an error of 1 bit is mixed into the usual stuffing bits wherein all the bits are "0", the same bit pattern as the synchronization code is produced as shown by the broken line in FIG. 32(a-1), so that a pseudo synchronization occurs. On the other hand, the stuffing bit STUFF representative of the code table of FIG. 30(b) is not the same pattern as the synchronization code even if an error is mixed as shown in FIG. 32(b-2), so that no pseudo synchronization occurs.

Thus, the stuffing bit in this preferred embodiment has advantages in that it is possible to easily detect an error of a code string, and it is difficult to produce a pseudo synchronization code even if an error is mixed into a code string, so that it is possible to provide a strong error resistance.

In addition, the stuffing bit in this preferred embodiment can uniformly decode in the backward direction, and identify the starting position thereof, i.e., the end position of a code string immediately before the stuffing bit STUFF. Therefore, after the coding, by which an information code string can be decoded in both of forward and backward directions, is carried out, the code string immediately before the STUFF can be decoded in the backward direction as shown by the arrow 3104 in FIG. 31.

In the aforementioned preferred embodiment, the stuffing bit STUFF may be determined as follows.

(1) In a case where a synchronization code contains "0"s of sync_0_len bits as shown in FIG. 5, if all the bits of the stuffing bits STUFF or at least at the synchronization code inserting positions are set to be "1", the humming distances between the portions of "0" of the synchronization code and the stuffing bits STUFF can be

increased. Therefore, it is possible to decrease the probability that an error is mixed into the stuffing bits STUFF to produce a pseudo synchronization.

5 (2) The stuffing bit STUFF may be a code word representative of the length thereof. In the decoding system, the length of the STUFF from the point, at which the decoding of a code string other than the stuffing bit STUFF is completed, is determined, and the decoding of the STUFF is decoded to decode the length information of the
10 STUFF. In this case, if both are coincident with each other, it can be determined that an error is mixed into the code string.

In addition, the length of the code string of the stuffing bits STUFF may be indicated by binary numbers.
15 For example, if the STUFFs are 5 bits, "5" may be indicated by binary numbers so as to be "00101". Alternatively, a value derived by taking a complement of "1" or "2" from a value indicated by binary numbers may be used as a code word of the stuffing bits STUFF. Thus, the number of bits
20 of "0"s in the STUFFs is decreased, so that it is possible to inhibit a pseudo synchronization from occurring similar to the aforementioned (1).

(3) In a case where the coding is carried out using a code word which can be decoded in both of forward and
25 backward directions, it is required to decode the stuffing bits STUFFs in the backward direction from the end point of the frame in the decoding system, to find the starting point (the boundary point between the STUFF and another code word). In such a case, the STUFFs may be determined
30 so as to be a code word, which starts from "0"(s) of 1 bit or a plurality of bits and has the residuals of "1"s, such as "01111111". Thus, if the STUFFs are decoded in the backward direction to seek the position of "0", it can be uniformly determined that the sought position is the
35 starting point of the STUFFs. In addition, in this example, the bits other than the initial portion of the stuffing bits STUFFs are "1", so that it is possible to

decrease the probability that a pseudo synchronization occurs similar to the aforementioned (1).

(4) The stuffing bit STUFF may be a check bit, a parity check bit or the like for an error correcting/detecting code of a part or all of bits of an output code string. Thus, the error correction/detection of a bit error mixed into an output code string can be carried out.

As described in the above examples, stuffing bits STUFFs are produced in accordance with a predetermined rule, and the stuffing bits STUFFs in an input code string are checked with the producing rule in a decoding system. If it is determined that the stuffing bits STUFFs are against the producing rule, it can be determined that an error is mixed into the input code string. Thus, if a process for preventing a reproduced image from being greatly deteriorated is carried out in a dynamic image decoding system, it is possible to improve the quality of a reproduced image when an error is mixed into the input code string.

Moreover, in the aforementioned preferred embodiment, the synchronization code inserting interval `sync_period` may be determined as follows.

(1) In a case where an error correcting/detecting code is used, a synchronization code inserting interval `sync_period` may be greater than the minimum number of bits required to carry out the synchronization detection by means of a decoding system, i.e., the sum of the length of a synchronization code and the maximum value of a check bit for an error correcting/detecting code. Since the mean value of the numbers of bits of the last stuffing bits STUFFs of a frame is $\text{sync_period} / 2$, if the `sync_period` has the minimum bits, by which the synchronization detection can be carried out, it is possible to decrease the number of bits of the stuffing bits STUFFs to improve the coding efficiency.

(2) In a case where no error correcting/detecting

code is used, a synchronization code inserting interval
sync_period may be greater than the minimum number of bits
required to carry out the synchronization detection by
means of a decoding system, i.e., the length of a
5 synchronization code. Since the mean value of the numbers
of bits of the last stuffing bits STUFFs of a frame is
sync_period / 2, if the sync_period has the minimum bits,
by which the synchronization detection can be carried out,
it is possible to decrease the number of bits of the
10 stuffing bits STUFFs to improve the coding efficiency.

(3) In a case where a frame length information
POINTER is used as shown in FIGS. 15, 16, 17 and 27, a
synchronization code inserting interval sync_period may be
shorter than the length of a synchronization code. Thus,
15 it is possible to decrease the number of bits of the
stuffing bits STUFFs to improve the coding efficiency.

(4) In a case where the transmission/storage is
carried out by dividing into packets or cells at intervals
determined in a transmission line or a storage medium, a
20 synchronization code inserting interval sync_period may be
matched with the interval of the packets or cells, or a
divisor thereof. Thus, since the head of the packets or
cells is always arranged at the synchronization code
inserting position, it is possible to detect a
25 synchronization code even if a packet or cell is produced
due to packet loss or cell loss.

(5) The synchronization code inserting interval
sync_period is preferably shorter than the required
minimum number of bits of a frame. Thus, it is possible
30 to decrease the number of bits of the stuffing bits STUFFs
to improve the coding efficiency.

(Sixth Preferred Embodiment)

The sixth preferred embodiment of the present
invention will be described.

35 FIG. 35 shows examples of output code strings of a
dynamic image coding system in this preferred embodiment.
In these output code strings, in order to decrease the

probability of the error detection of a synchronization code due to errors, the bit inserting process as described in the aforementioned preferred embodiments is carried out. In addition, information, such as heater information, is arranged at predetermined positions, or at predetermined positions based on a synchronization code.

FIG. 35(a) is a code string before a bit inserting process is carried out, and FIG. 35(b) is a code string after a bit inserting process is carried out. In the drawings, each of the portions 3201, 3202, 3261 and 3262 shown by the slanting lines shows information arranged at the predetermined position (the predetermined position based on a synchronization code), and each of void arrows 3211 and 3212 indicates the position, into which the information is inserted. The information 3261 and information 3262 of the code string in FIG. 35(b) corresponds to the information 3201 and information 3202 of the code string in FIG. 35(a), respectively. In some cases, when the code string (a) is transformed to the code string (b), these information may be transformed (i.e., the transformation from information 3201 to information 3261 and the transformation from information 3202 to information 3262).

In FIG. 35(b), 3203 denotes a bit inserted by the bit inserting process. Since the bit string subsequently to the inserted bit is shifted backward by the bit inserting process, a part of a code string immediately before the information to be inserted at a predetermined position is shifted so that the information is inserted at the predetermined position. For example, assuming that the total of the numbers of inserted bits from the synchronization code 3205 immediately before the information 3201 is $Ns1$, $Ns1$ bits indicated by sign 3221 in FIG. 35(a) immediately before the information 3201 may be shifted to the portion of sign 3231 in FIG. 35(b) immediately after the information 3201.

If information such as a pointer representative of

a specific position in a code string is contained in information 3201 and/or 3202, this may be transformed. Specifically, for example, if information representative of the position indicated by arrow 3241 is contained in
5 information 3201, the information representative of the position in information 3261 is transformed so as to indicate the position representative of arrow 3251 after the position by the number Ns1 of the inserted bits.

CLAIMS

1. A coding system comprising:

coding means for coding an input code string to an error correcting/detecting code comprising an information bit and a check bit; and

code string assembling means for inserting a synchroni-zation code into any one of a plurality of predetermined synchronization code inserting positions in an output code string, for arranging said information bit at an optional position in said output code string, and for arranging said check bit at a position other than said synchronization code inserting positions in said output code string to assemble said output code string.

2. A decoding system comprising:

synchronization code detecting means for detecting a synchronization code at a plurality of predetermined synchronization code inserting positions on the basis of a code string, which is coded to an error correcting/detect-ing code comprising an information bit and a check bit;

code string resolving means for resolving said code string to extract said information bit of said error correcting/detecting code and said check bit of said error correcting/detecting code arranged at a position other than said synchronization code inserting positions; and

decoding means for receiving said information bit and said check bit extracted by said code string resolving means to decode said error correcting/detecting code.

3. A coding system comprising:

code string transforming means for transforming an input code string other than synchronization codes arranged at a plurality of predetermined synchronization code inserting positions in an output code string so that a humming distance from said synchronization code is equal to or greater than a predetermined value;



coding means for coding a code string transformed by said code string transforming means, to an error correcting/detecting code comprising an information bit and a check bit; and

code string assembling means for inserting a synchro-nization code into any one of a plurality of predetermined synchronization code inserting positions in said output code string, for arranging said information bit at an optional position in an output code string and for arranging said check bit at a position other than said synchronization code inserting positions in said output code string to assemble said output code string.

4. A decoding system comprising:

synchronization code detecting means for detecting a synchronization code at a predetermined synchronization code inserting position on the basis of a code string, which is coded to an error correcting/detecting code comprising an information bit and a check bit and into which the synchronization code;

code string resolving means for resolving said code string to extract said information bit of said error correcting/detecting code and said check bit of said error correcting/detecting code arranged at a position other than said synchronization code inserting position;

decoding means for receiving said information bit and said check bit extracted by said code string resolving means to decode said error correcting/detecting code; and

code string transforming means for transforming a code string other than said synchronization code, which is transformed so that a humming distance from said synchronization code in said code string decoded by said decoding means is equal to or greater than a predetermined value and which exists at said synchronization code inserting position, into the original code string.

5. A coding system comprising:



coding means for coding an input code string to an error correcting/detecting code;

synchronization code inserting means for inserting a synchronization code into said code string; and

determining means for determining the number of bits an information to be coded to an error correcting/detecting code immediately before said synchronization code in said code string,

said coding means coding said error correcting/detecting code immediately before said synchronization code, using a degenerate code, which is adaptively degenerated on the basis of the number of bits determined by said determining means.

6. A decoding system comprising:

decoding means for decoding a code string, which is coded to an error correcting/detecting code and into which a synchronization code is inserted;

synchronization code detecting means for detecting said synchronization code in said code string; and

determining means for determining the number of bits of an information, which is coded to an error correcting-/detecting code immediately before said synchronization code in said code string detected by said synchronization code detecting means,

said decoding means decoding by identifying whether said error correcting/detecting code immediately before said synchronization code is a degenerate code on the basis of the determined result of said determining means.

7. A coding system comprising:

coding means for coding a code string containing kinds of input information to an error correcting/detecting code; and

switching means for switching the kind of said error correcting/detecting code in accordance with the kinds of said input information in said code string.



8. A decoding system comprising:

decoding means for decoding a code string, which is coded to a different kind of error correcting/detecting code in accordance of the kind of information, to generate the original information; and

means for determining the kind of said error correct-ing/detecting code on the basis of the kind of information generated by said decoding means, to inform said decoding means.

9. A coding system comprising:

code string transforming means for transforming an input code string other than synchronization codes, which are arranged at a plurality of predetermined synchronization code inserting positions in an output code string and in intervals of a predetermined number of bits before and after the predetermined synchronization code inserting positions in an output code string, so that a humming distance from said synchronization code is equal to or greater than a predetermined value;

coding means for coding a code string transformed by said code string transforming means, to an error correct-ing/detecting code comprising an information bit and a check bit; and

code string assembling means for inserting a synchronization code into any one of a plurality of predetermined synchronization code inserting positions in said output code string, for arranging said information bit at an optional position in said output code string, and for arranging said check bit at a position other than said synchronization code inserting positions in said output code string to assemble said output code string.

10. A decoding system comprising:

synchronization code detecting means for detecting a synchronization code at a predetermined synchronization code inserting position and in an interval of a



predetermined number of bits before and after the predetermined synchronization code inserting position, on the basis of a code string, which is coded to an error correcting/detecting code comprising an information bit and a check bit and into which the synchronization code;

code string resolving means for resolving said code string to extract said information bit of said error correcting/detecting code and said check bit of said error correcting/detecting code arranged at a position other than said synchronization code inserting position;

decoding means for receiving said information bit and said check bit extracted by said code string resolving means to decode said error correcting/detecting code; and

code string transforming means for transforming a code string other than said synchronization code, which is transformed so that a humming distance from said synchronization code in said code string decoded by said decoding means is equal to or greater than a predetermined value and which exists at said synchronization code inserting position and in an interval of a predetermined number of bits before and after said synchronization code inserting position, into the original code string.

11. A coding system comprising:

multiplexing means for multiplexing kinds of compressed codes, which are obtained by compression coding an input signal, to produce a multiplexed code string; and

code string assembling means for inputting said multiplexed code string to assemble an output code string,

said code string assembling means inserting a synchronization code into any one of a plurality of periodically predetermined synchronization code inserting positions in said output code string.

12. A coding system comprising:

multiplexing means for multiplexing kinds of compressed codes, which are obtained by compression coding



an input signal, to produce a multiplexed code string; and
code string assembling means for inputting said
multi-plexed code string to assemble an output code string,
said code string assembling means inserting a
synchro-nization code into any one of a plurality of
periodically predetermined synchronization code inserting
positions in said output code string, and suitably
inserting a stuffing bit into said output code string.

13. A coding system comprising:

multiplexing means for multiplexing kinds of
compressed codes, which are obtained by compression coding
an input signal, to produce a multiplexed code string; and
code string assembling means for inputting said
multi-plexed code string to assemble an output code string,
said code string assembling means predetermining a
plurality of synchronization code inserting positions
arranged periodically in said output code string, inserting
an information indicating a boundary of said multiplexed
code string, and inserting a synchronization code into any
one of said plurality of synchronization code inserting
positions.

14. A coding system as set forth in any one of claims 11
through 13, wherein said multiplexing means multiplexes a
compressed code, which is compression coded for each of
frames or segments of the frames of an input image signal,
for each of said frames or said segments of the frame, and
wherein said code string assembling means inserts a
synchronization code into a synchronization code inserting
position immediately before or after the end portion of
each synchronizing unit multiplexed for each of said frames
or said segments of the frames of said multiplexed code
string.

15. A coding system as set forth in any one of claims 11
through 13, which further comprises code string



transforming means for transforming a code string other than said synchronization code, which is arranged at said synchronization code inserting position in said output code string, so that a humming distance from said synchronization code is equal to or greater than a predetermined value.

16. A decoding system comprising:

synchronization code detecting means for detecting a synchronization code in an output code string;

demultiplexing means for demultiplexing on the basis of the position of the synchronization code detected by said synchronization code detecting means from said input code string, to produce a compressed code string; and

decoding means for decoding said compressed code to output a reconstructed signal,

said synchronization code detecting means detecting said synchronization code at a plurality of periodically predetermined synchronization code inserting positions in said input code string.

17. A decoding system as set forth in claim 16, which further comprises code string transforming means for transforming a code string other than said synchronization code, which is transformed so that a humming distance from said synchronization code is equal to or greater than a predetermined value at said synchronization code inserting position of said input code string, into the original code string.

18. A coding system comprising:

code string assembling means for inserting a synchronization code into any one of a plurality of periodically predetermined synchronization code inserting positions in an output code string, and for suitably inserting, into said code string, a stuffing bit, which is able to be uniformly decoded in a backward direction of



said output code string.

19. A coding system as set forth in claim 18, wherein said code string assembling means inserts said stuffing bit immediately before any one of said plurality of synchronization code inserting positions in said output code string.

20. A decoding system comprising:

decoding means for decoding an input code string, in which a synchronization code is inserted into any one of a plurality of periodically predetermined synchronization code inserting positions and into which a stuffing bit, which is able to be uniformly decoded in a backward direction, is suitably inserted; and

error detecting means for detecting an error in said input code string by comparing a decoding end position of a code string immediately before said stuffing bit, which is decoded by said decoding means, with a starting position of said stuffing bit.

21. A decoding system as set forth in claim 20, wherein said stuffing bit is inserted immediately before any one of said plurality of synchronization code inserting positions in said input code string.

22. A decoding system as set forth in claim 19, wherein said decoding means decodes a code string immediately before said stuffing bit in a backward direction from a starting position of said stuffing bit.

23. A coding system comprising:

code string assembling means for inserting a synchro-nization code into any one of a plurality of periodically predetermined synchronization code inserting positions in an output code string, and for suitably inserting, into said code string, a stuffing bit, which is able to be uniformly decoded in a backward direction of



said output code string and whose humming distances from said synchronization code and a part thereof are equal to or greater than a predetermined value.

24. A coding system as set forth in any one of claims 18 through 23, wherein said stuffing bit is arranged immediately before said synchronization code.

25. A decoding system comprising:

decoding means for decoding an input code string, in which a synchronization code is inserted into any one of a plurality of periodically predetermined synchronization code inserting positions and into which a stuffing bit is suitably inserted, said stuffing bit being able to be uniformly decoded in a backward direction, and the humming distances from said synchronization code and a part thereof being equal to or greater than a predetermined value; and

error detecting means for detecting an error in said input code string by comparing a decoding end position of a code string immediately before said stuffing bit, which is decoded by said decoding means, with a starting position of said stuffing bit.

26. A decoding system as set forth in any one of claims 20, 22 and 25, wherein said stuffing bit of said input code string is arranged immediately before said synchronization code.

27. A coding system as set forth in any one of claims 11 through 13, wherein said multiplexing means produces a multiplexed code string by multiplexing said compressed code string for each of a plurality of layers, and said code string assembling means inputs said multiplexed code string to add a synchronization code to at least a part of said plurality of layers to produce an output code.



28. A coding system as set forth in claim 27, wherein said code string assembling means causes said synchronization codes to have different lengths in a part or all of said plurality of layers.

29. A coding system as set forth in claim 27 or 28, which further comprises code string transforming means for transforming a code string other than said synchronization code in a part or all of layers, to which said synchronization code of said output code string is added, so that a humming distance from said synchronization code is equal to or greater than a predetermined value.

30. A coding system as set forth in claim 29, which further comprises code string transforming means for transforming a code string other than said synchronization code in a part or all of layers, to which said synchronization code of said output code string is added, so that a humming distance from a synchronization code, which has a shortest length of synchronization codes having different lengths, is equal to or greater than a predetermined value.

31. A decoding system as set forth in claim 16, wherein said input code string comprises a multiplexed code string, which is multiplexed for each of a plurality of layers, a synchronization code being added to at least a part of said plurality of layers.

32. A decoding system as set forth in claim 31, which further comprises code string transforming means for transforming a code string other than said synchronization code, which is transformed so that a humming distance from a synchronization code, which has a shortest length of synchronization codes having different lengths in a part or all of added layers, is equal to or greater than a predetermined value.



ABSTRACT

5 In a coding system wherein an error
correction/detect-ion coding is combined with a
synchronization recovering technique using a
synchronization code, the problems of a pseudo
synchronization and a step out due to error detect-ion are
solved. There is provided a coding part 212 for coding an
input multiplexed code string 201 to an error correcting-
/detecting code comprising an information bit and a check
10 bit, and code string assembling part 213 for inserting a
synchronization code into any one of a plurality of
periodically predetermined synchronization code inserting
positions in the code string 201, for arranging the
information bit at an optional position in the code string,
15 and for arranging the check bit at a position other than
the synchronization code inserting positions in the code
string 201 to assemble an output code string 205.

20

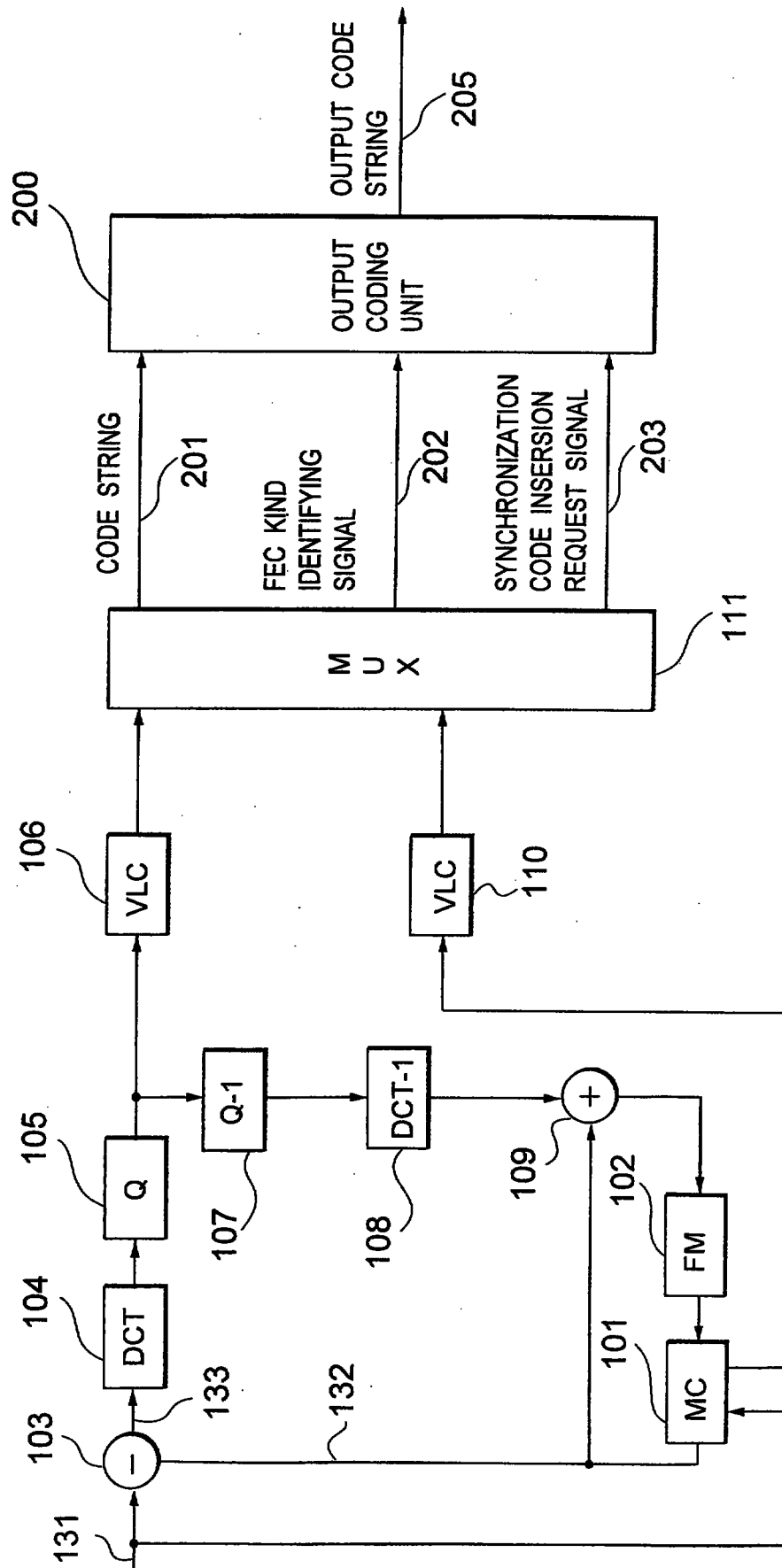


FIG. 1

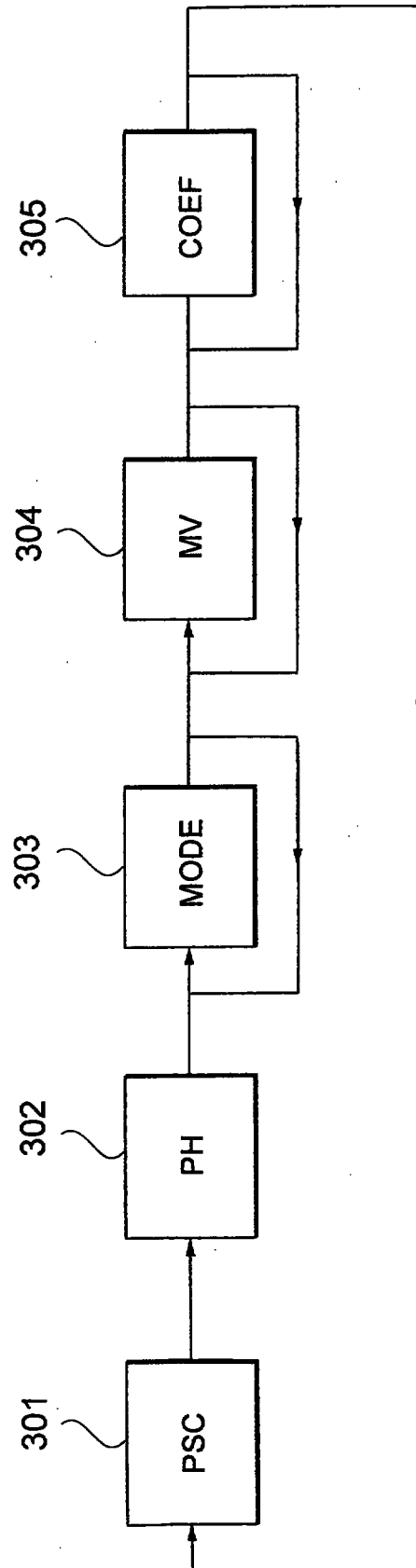


FIG. 2

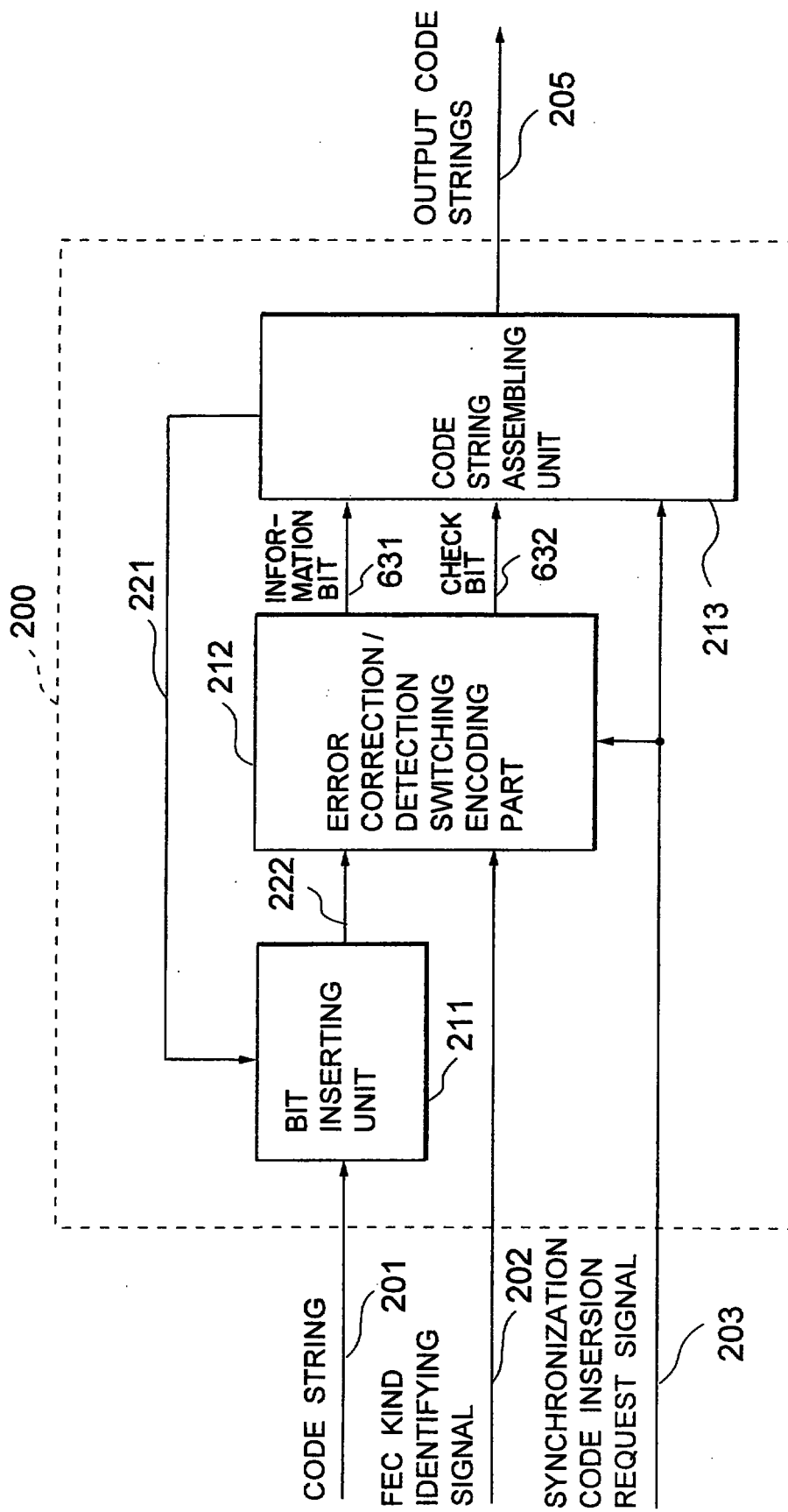


FIG. 3

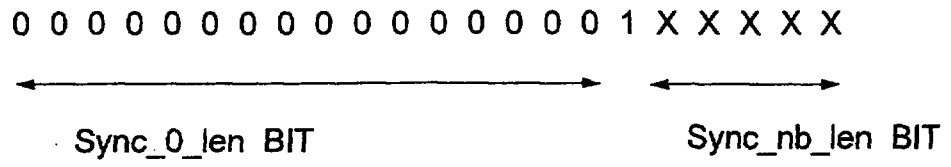


FIG.5

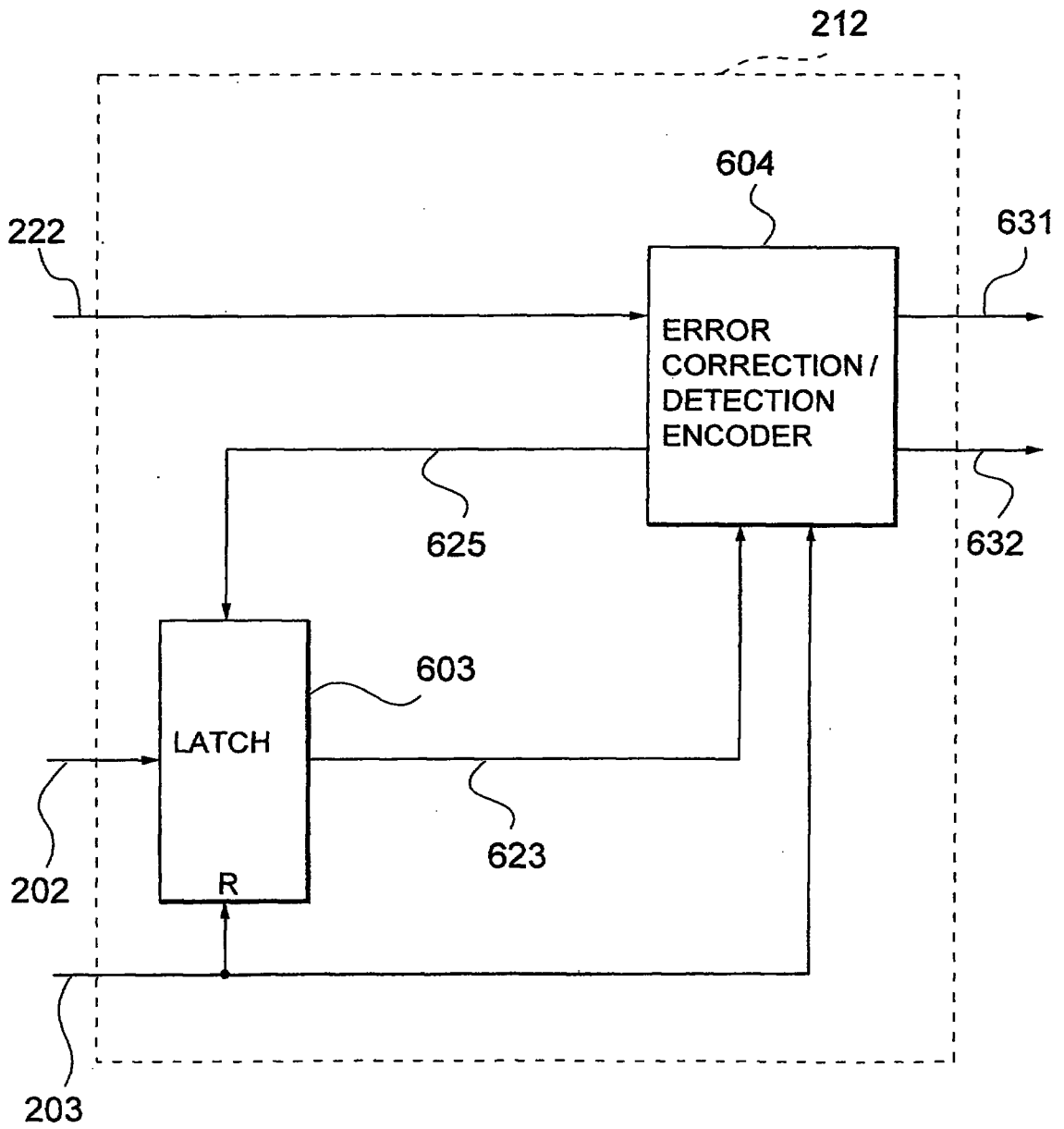


FIG.6

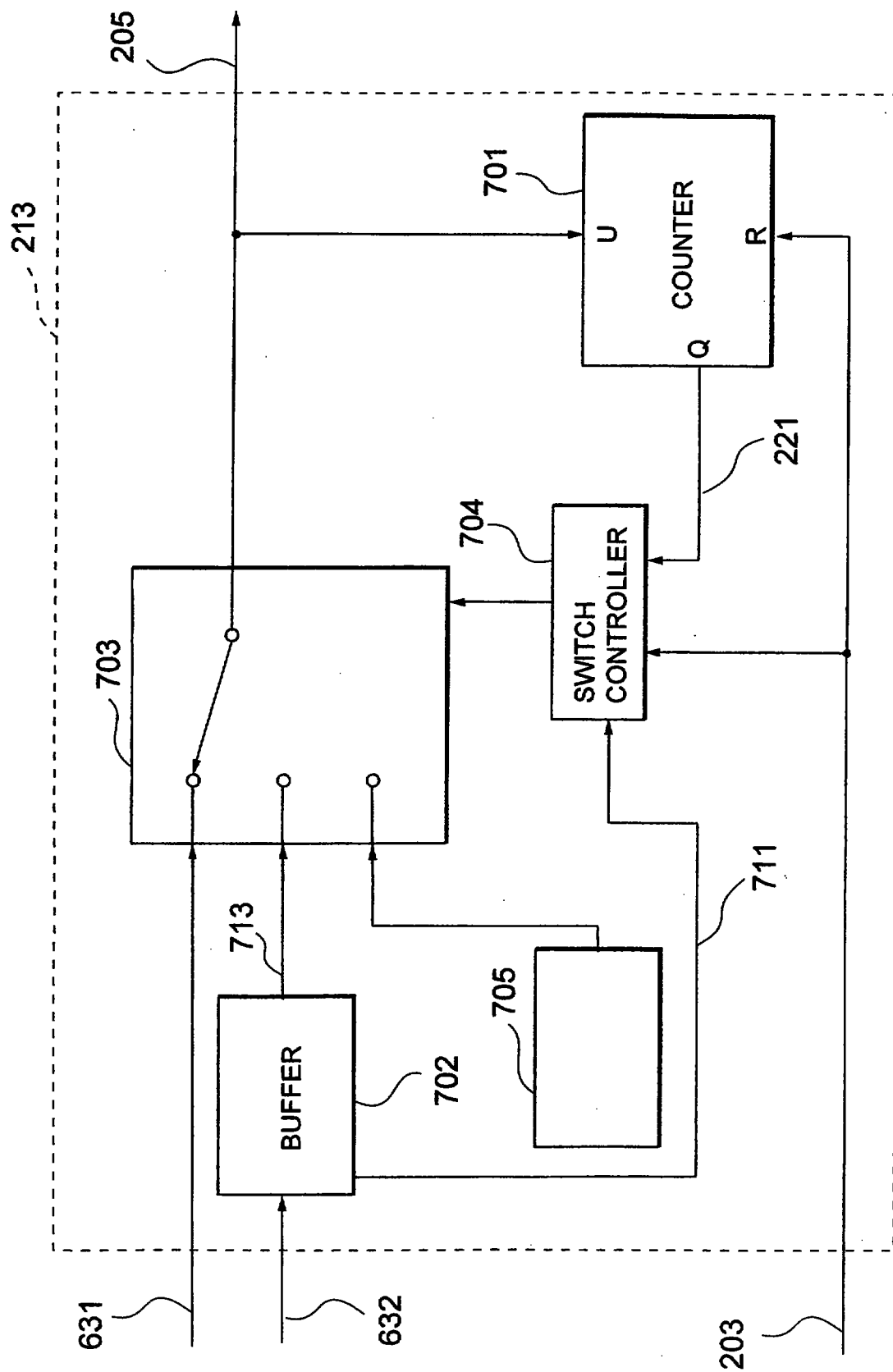


FIG. 7

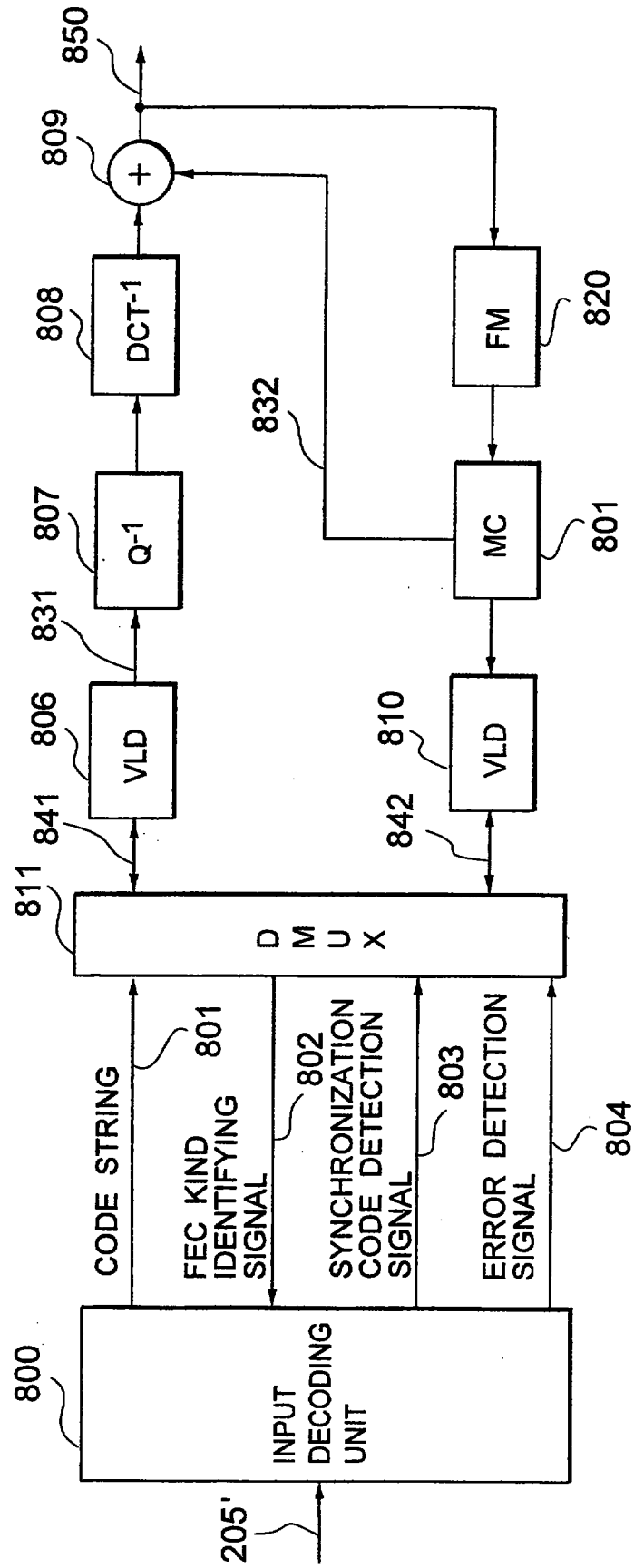


FIG. 8

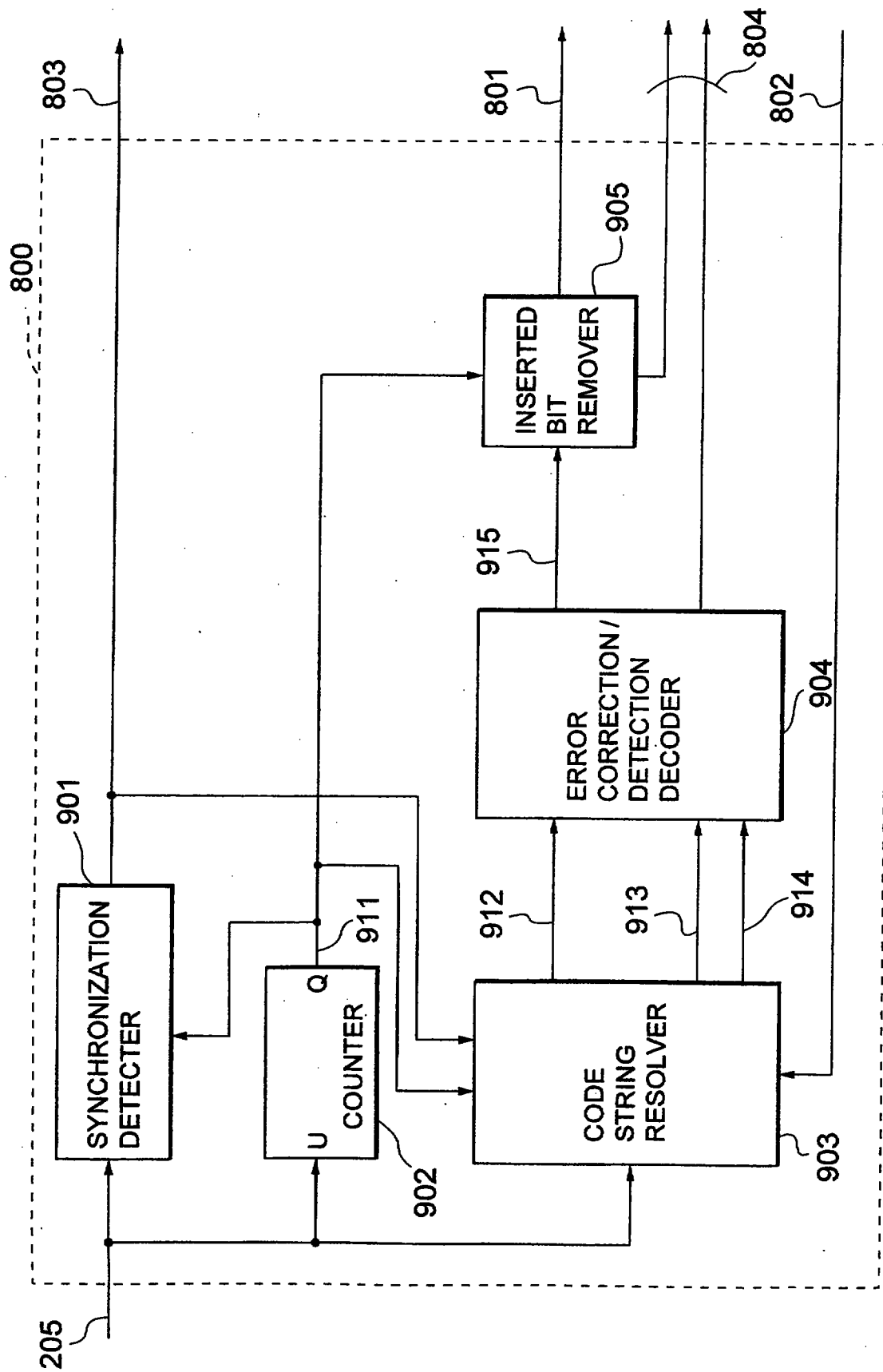


FIG. 9

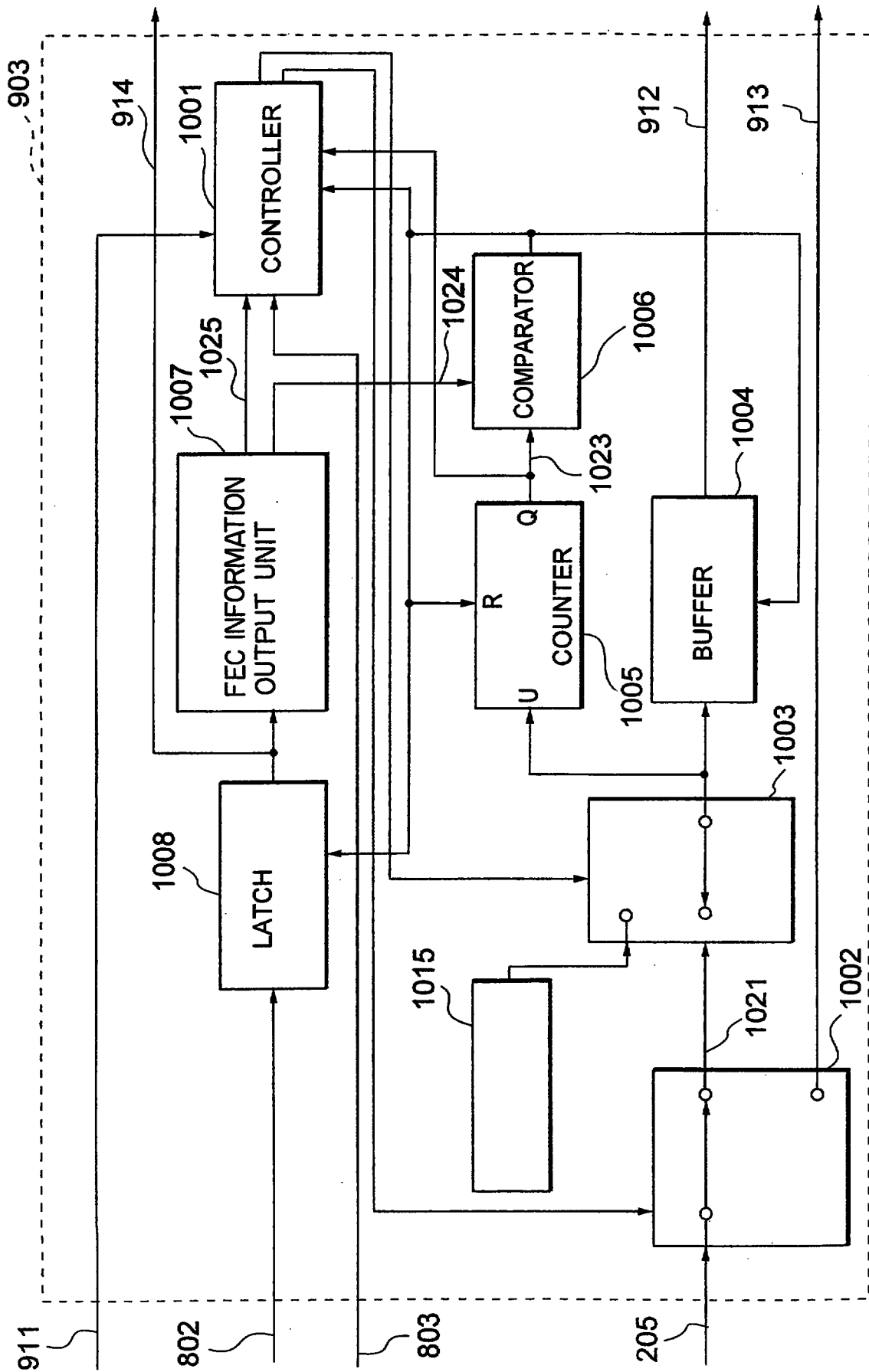


FIG. 10

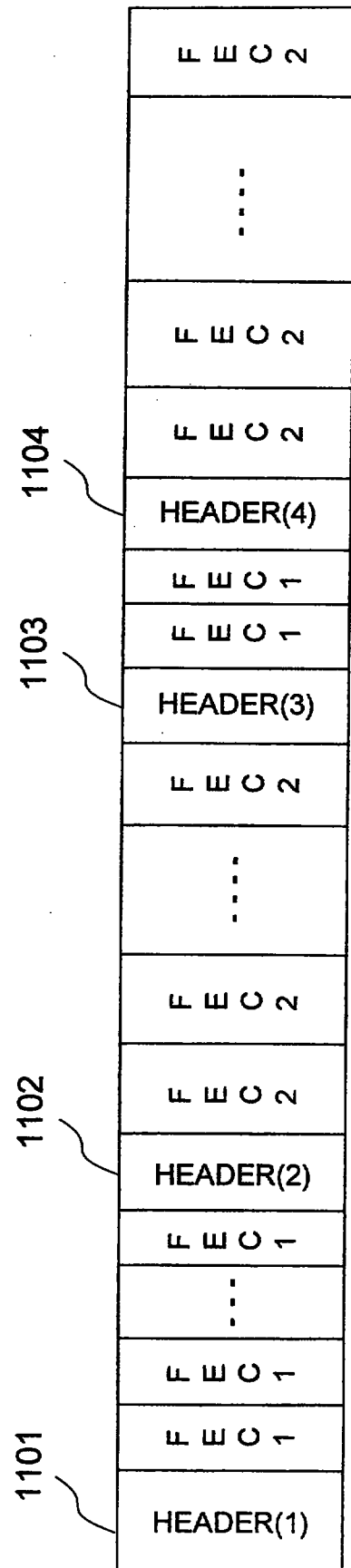
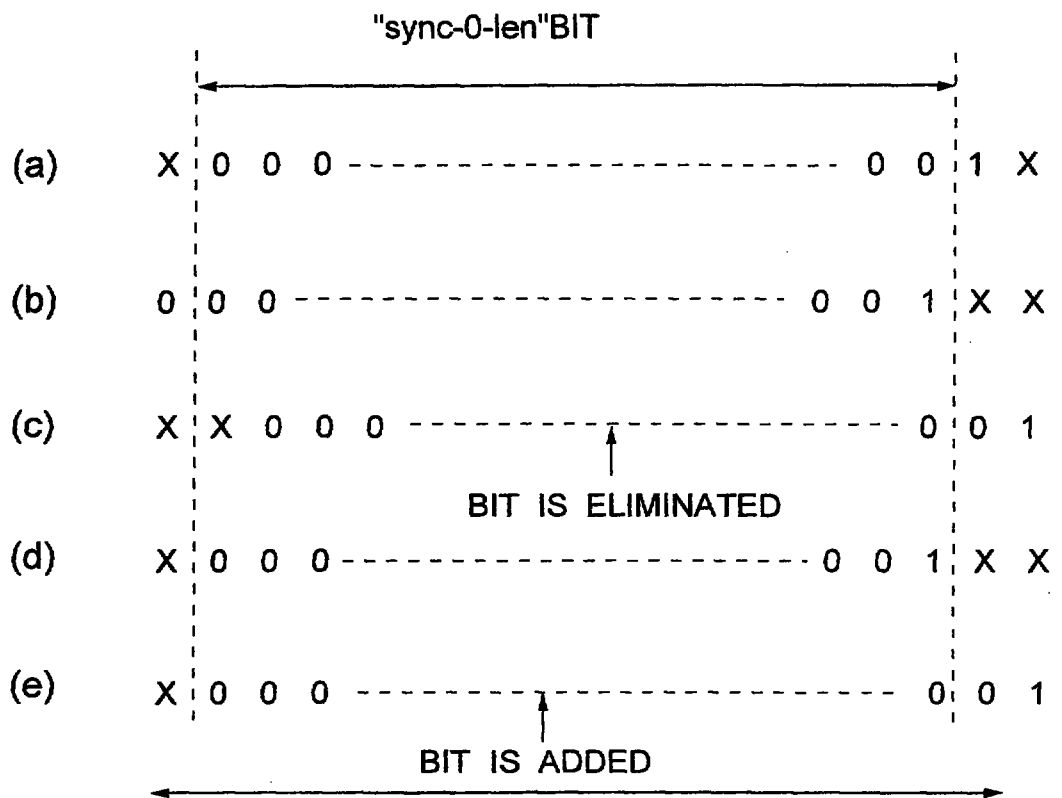


FIG. 11



SYNCHRONIZATION CODE DETECTION
WINDOW OF $\text{"sync-0-len"} + 2$ BIT

FIG.12

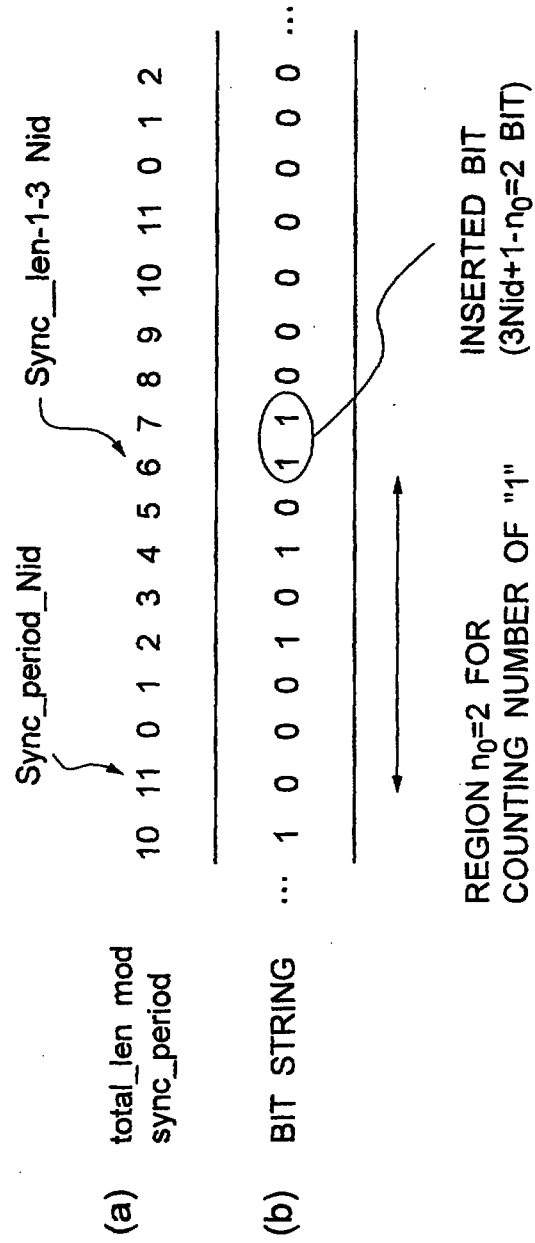


FIG. 13

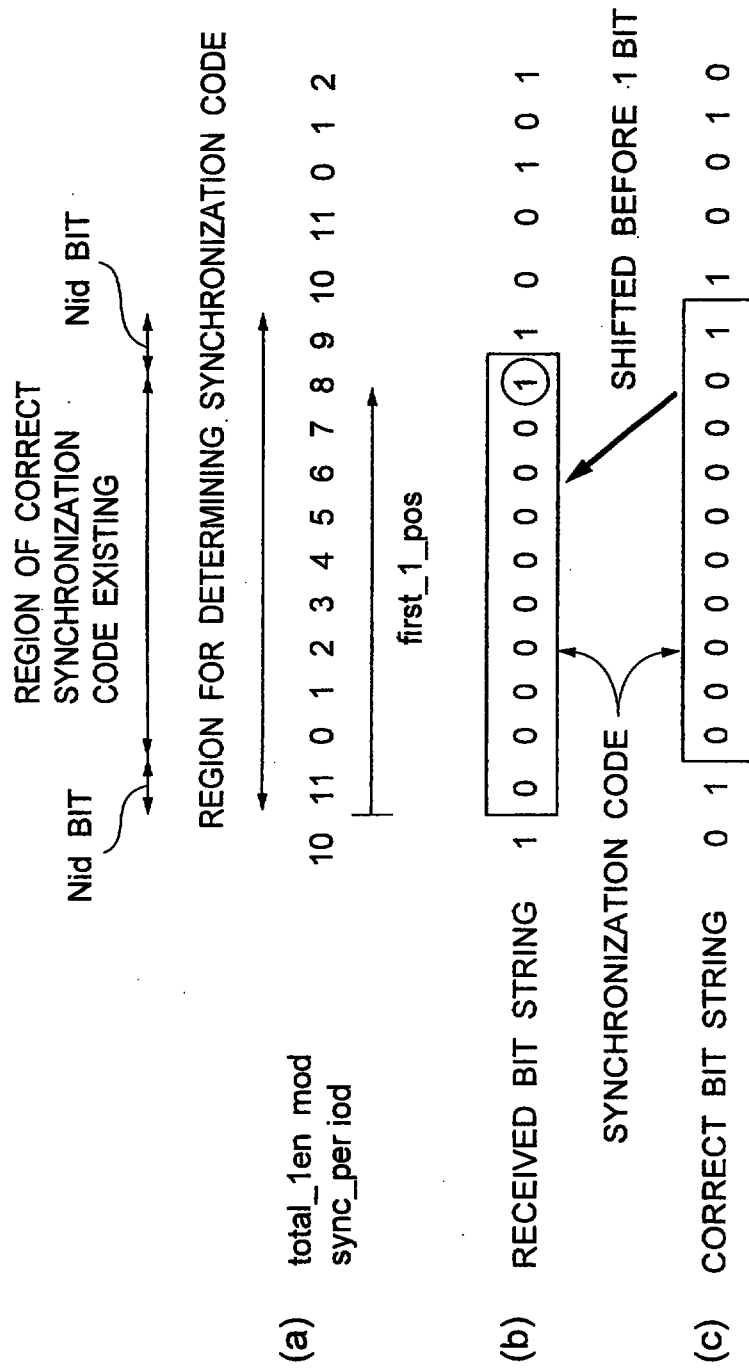


FIG. 14

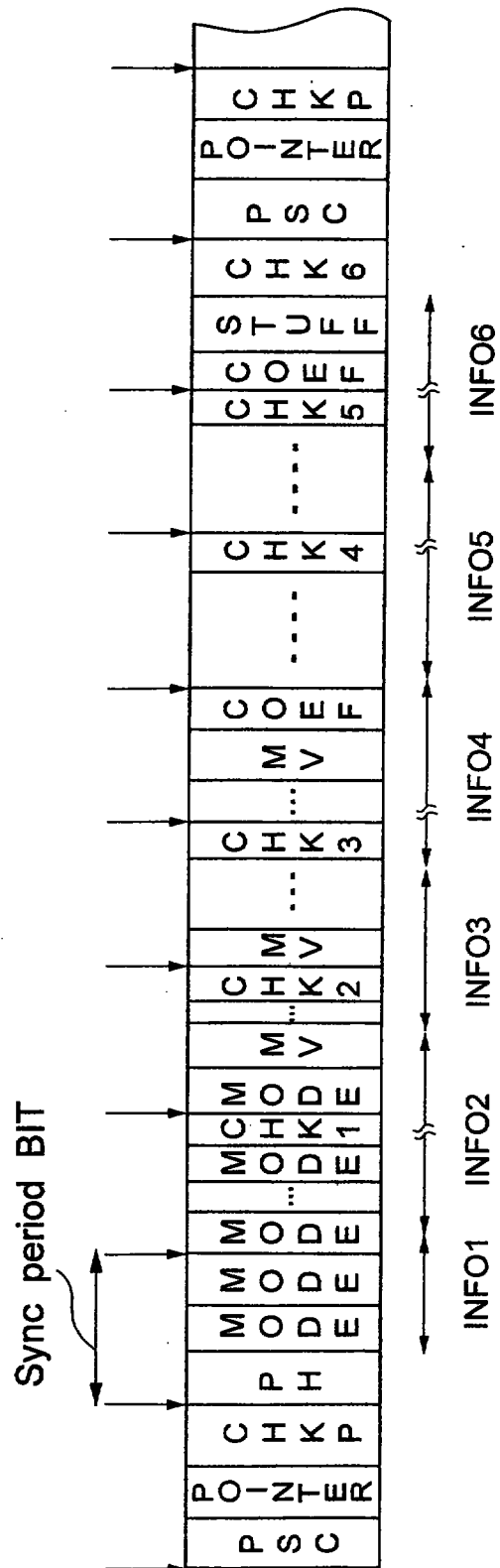


FIG. 15

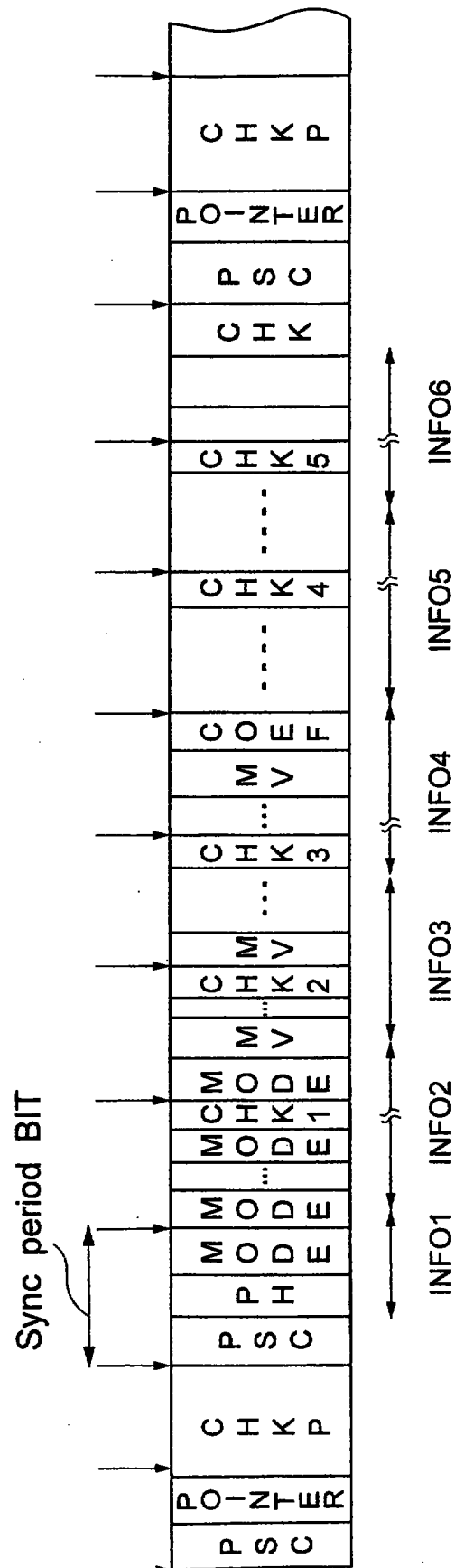


FIG. 16

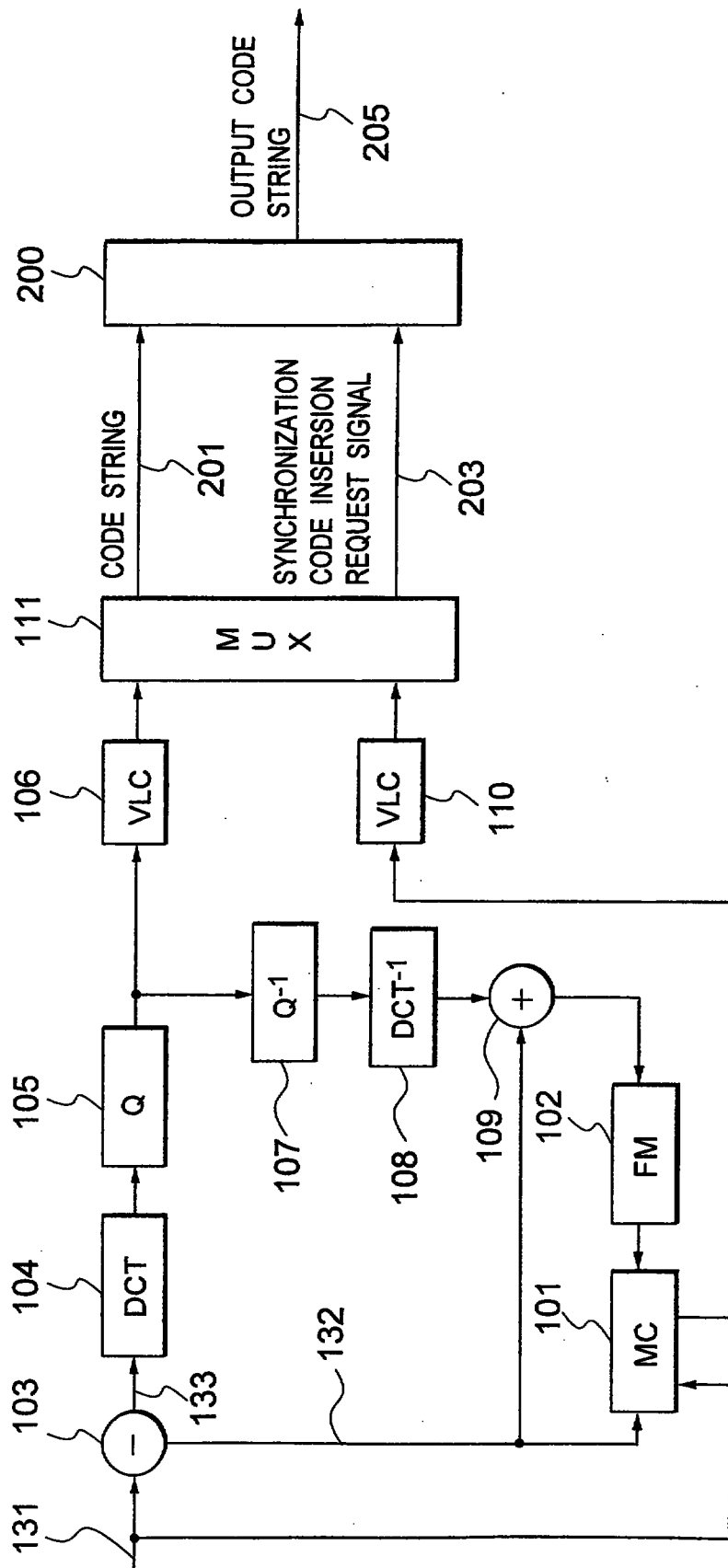


FIG. 18

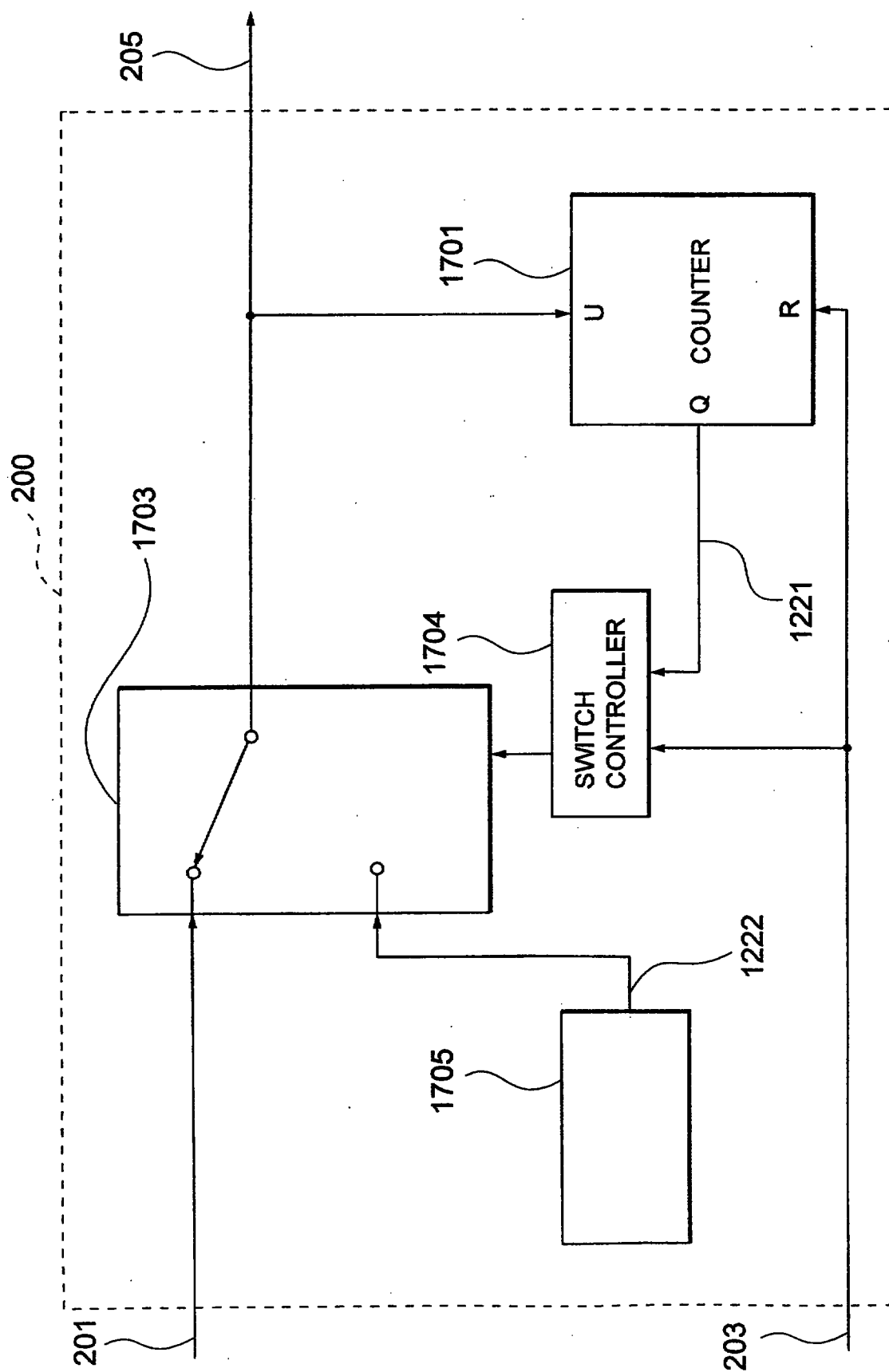


FIG. 19

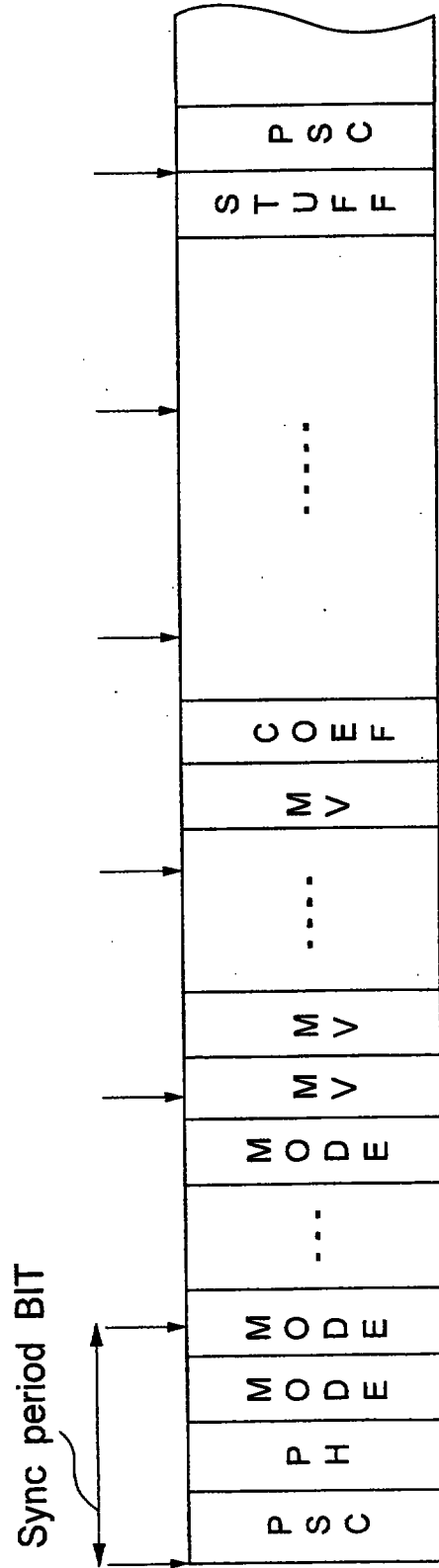


FIG. 20

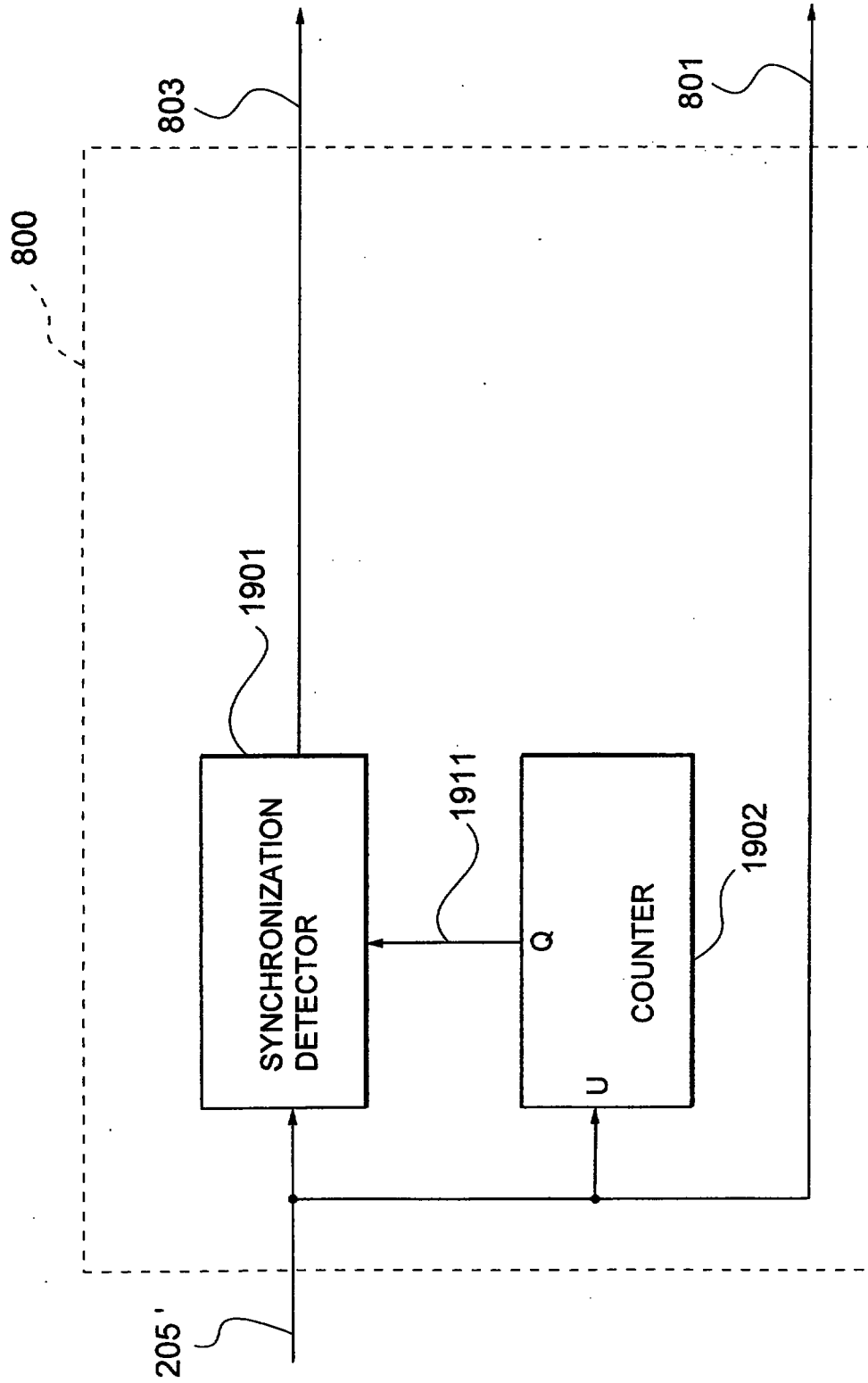


FIG. 22

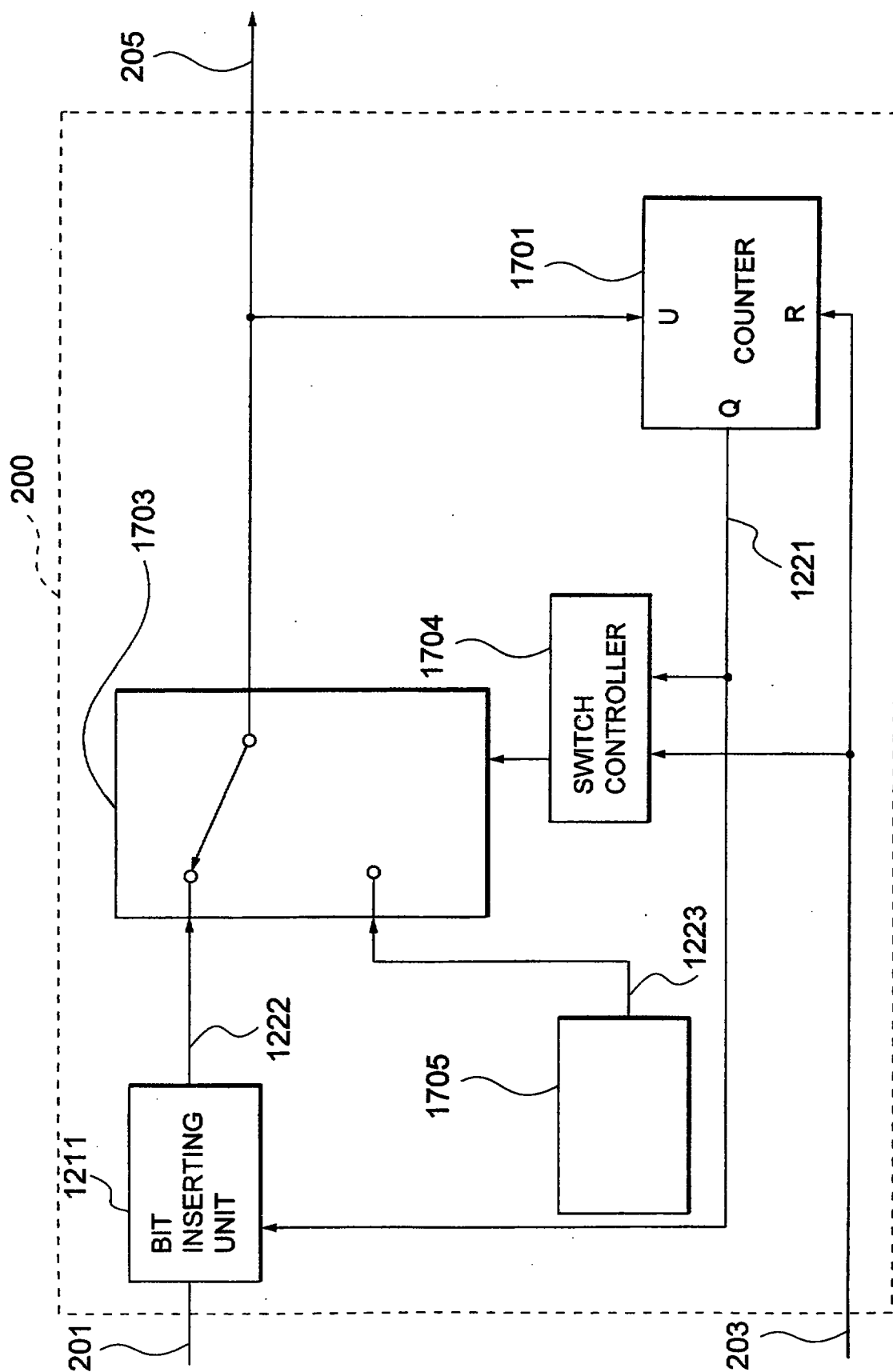


FIG. 23

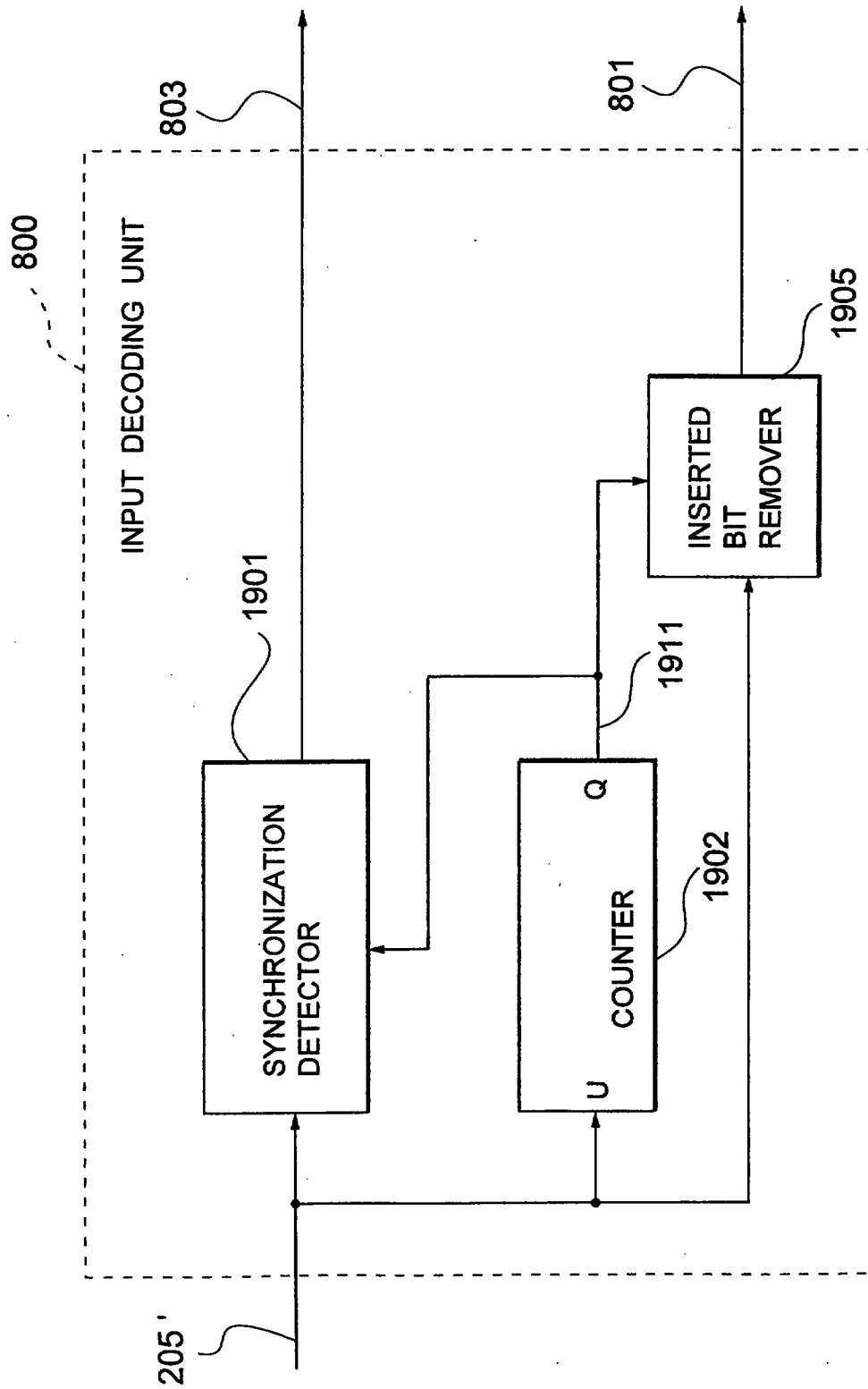
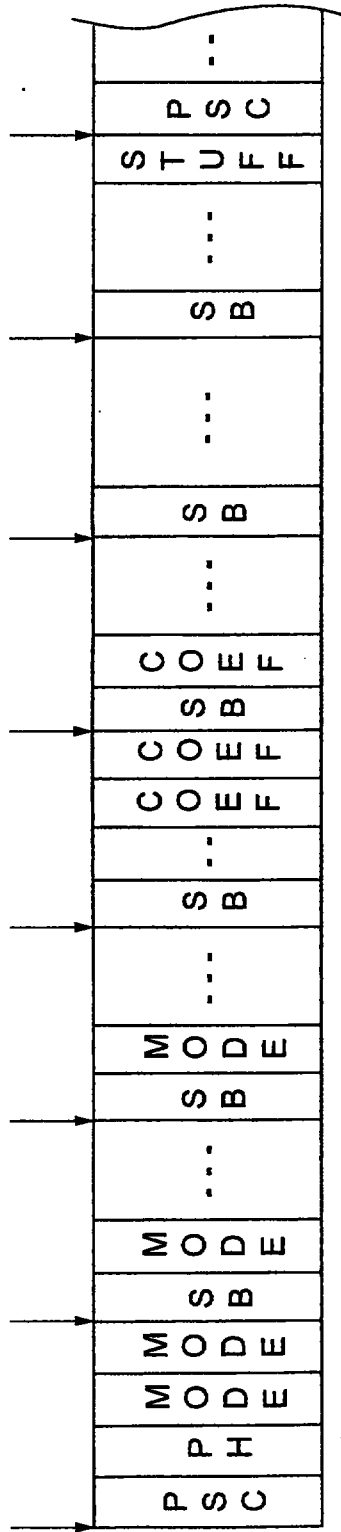
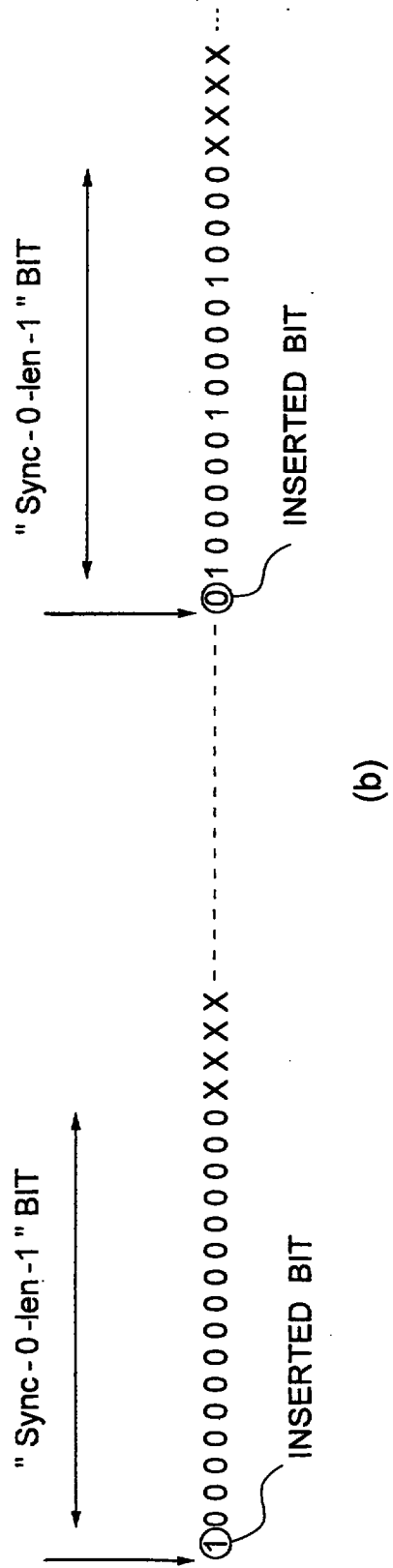


FIG. 24



(३)



(9)

FIG. 25

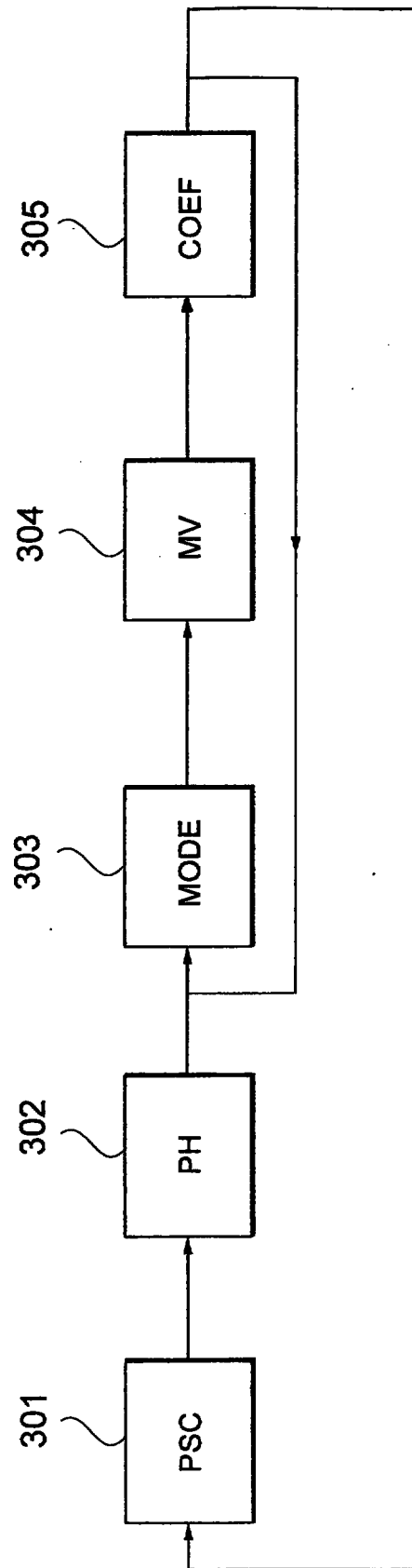


FIG. 26

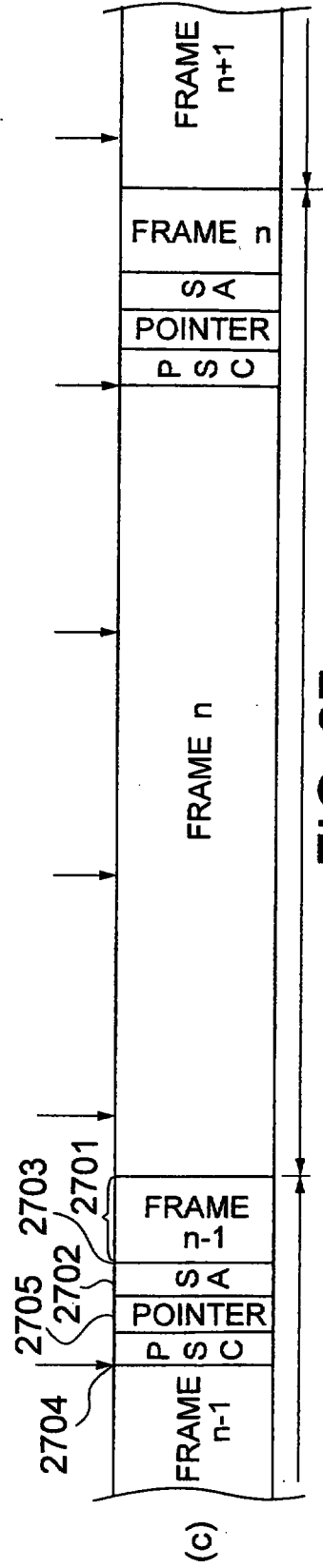
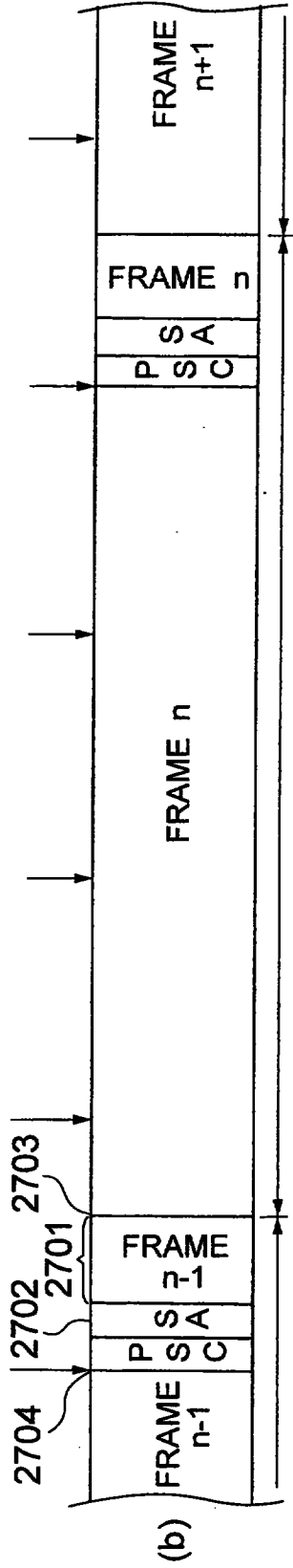
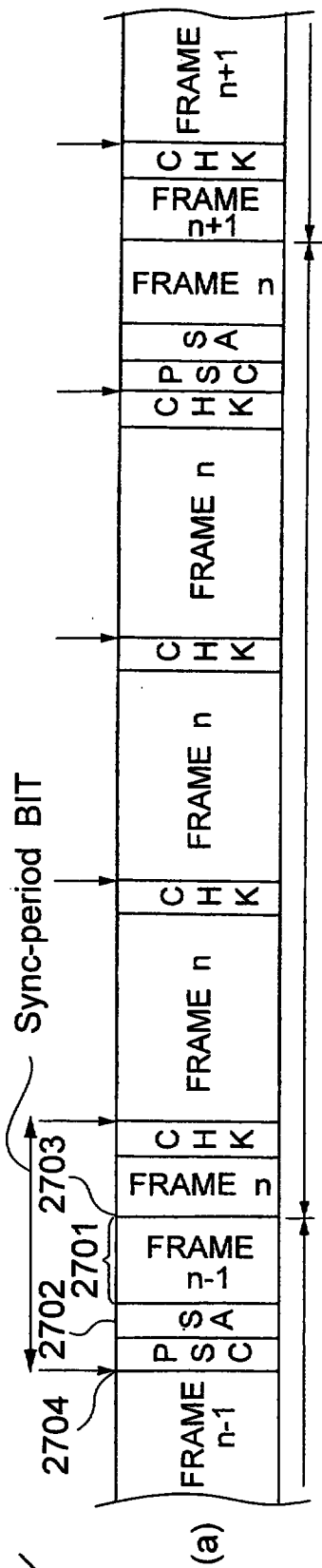
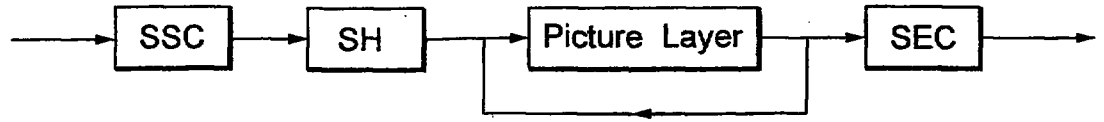


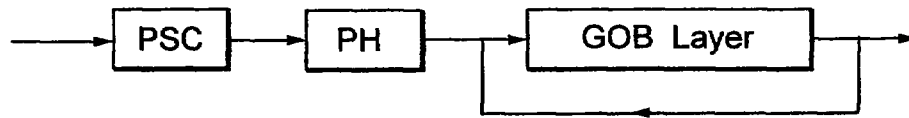
FIG. 27

SESSION LAYER
(Session Layer)



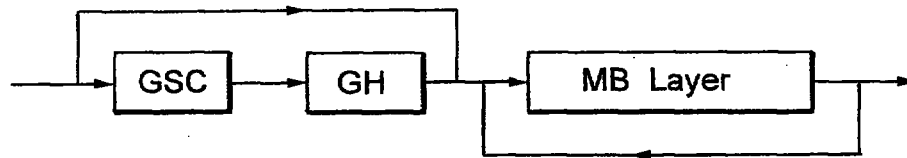
(a)

PICTURE LAYER
(Picture Layer)



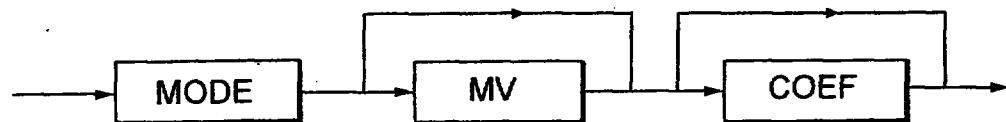
(b)

GOB LAYER
(GOB Layer)



(c)

MACRO-BLOCK LAYER
(MB Layer)



(d)

FIG.28

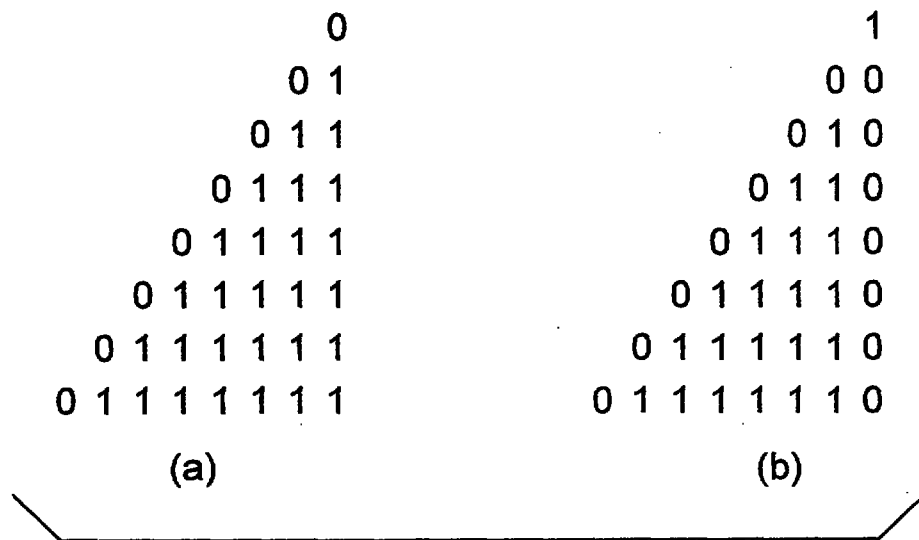


FIG.30

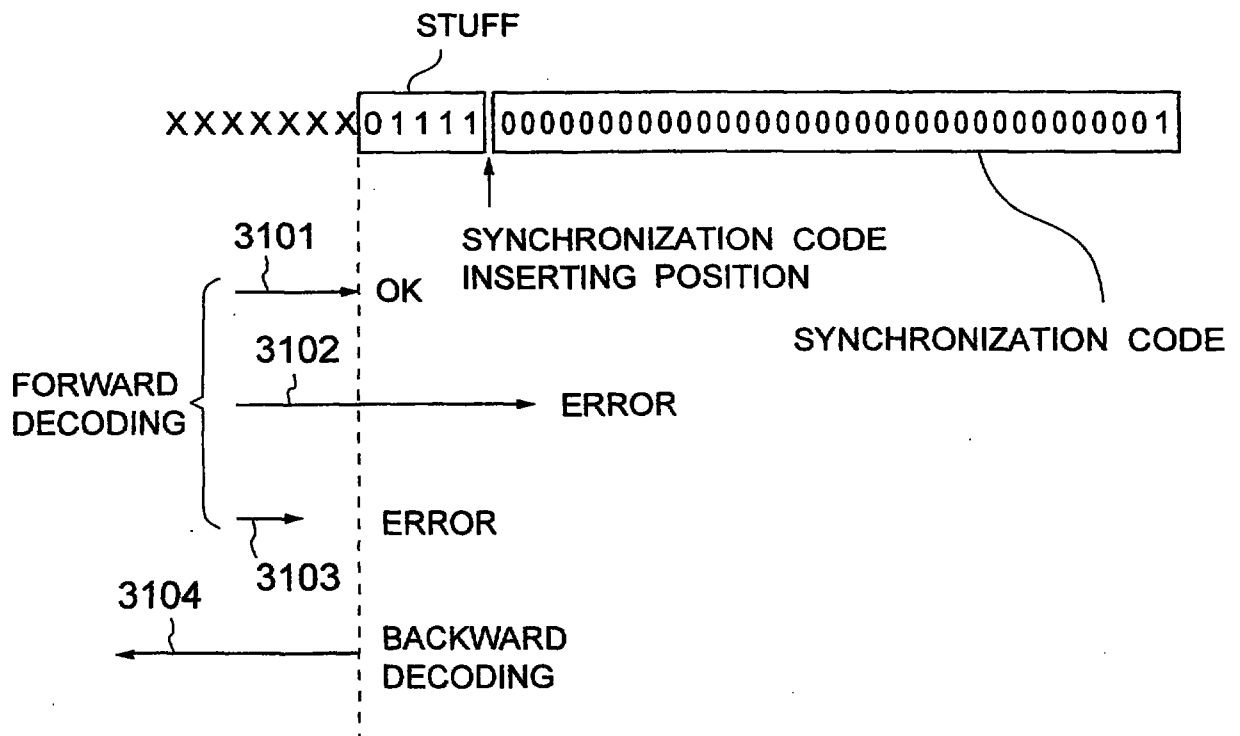


FIG.31

FIG. 32

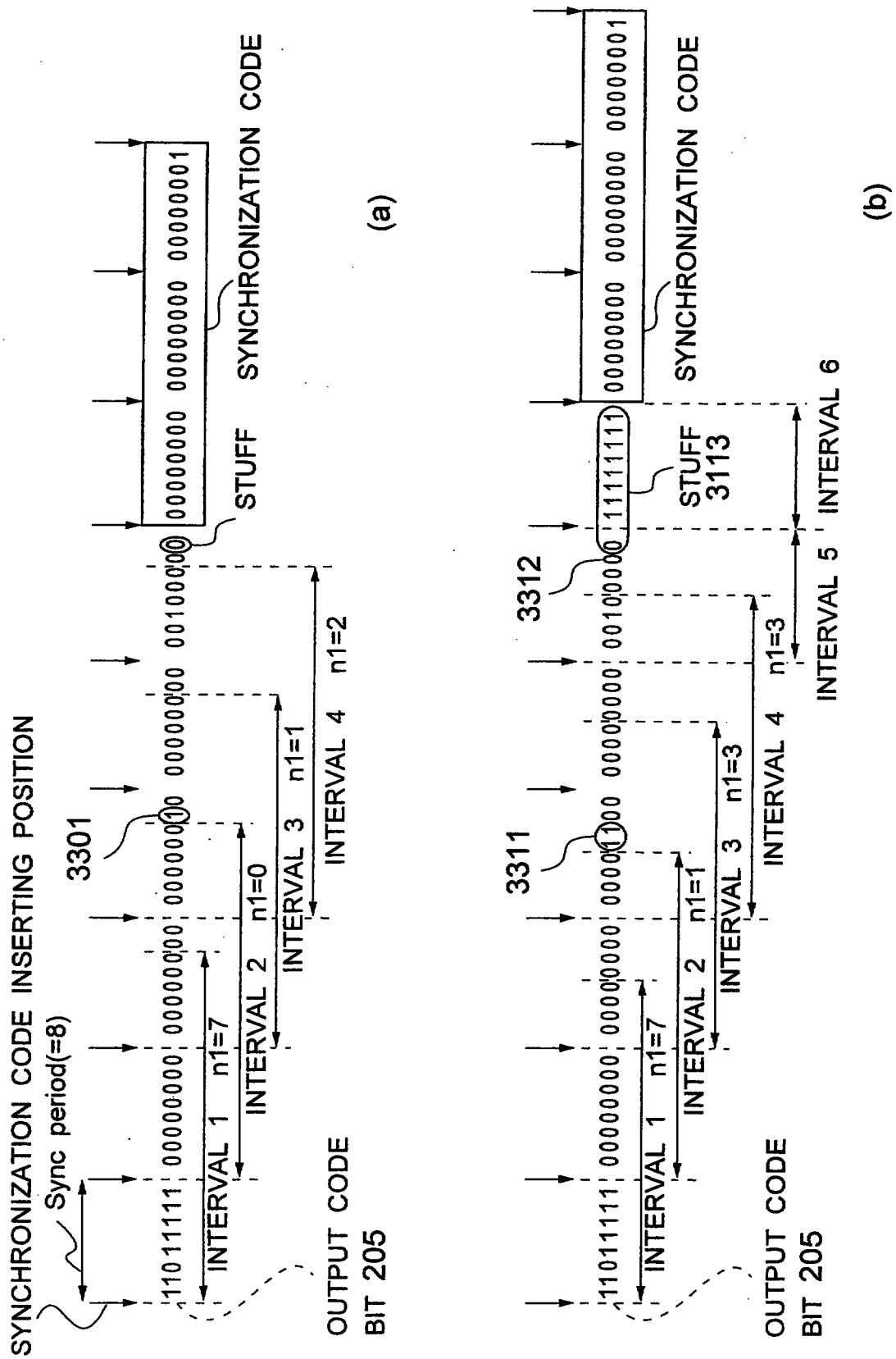


FIG. 33

SSC 00000000 00000000 00000001 00010101

SEC 00000000 00000000 00000001 00101010

PSC 00000000 00000000 00000001 00000000

GSC 00000000 00000000 1

(a)

PSC 00000000 00000000 00000000 1 00000000

GSC 00000000 1 00000000 00000000 1 1

3401 GSC 3411 3413 3412 3414 3402

(b)

FIG.34

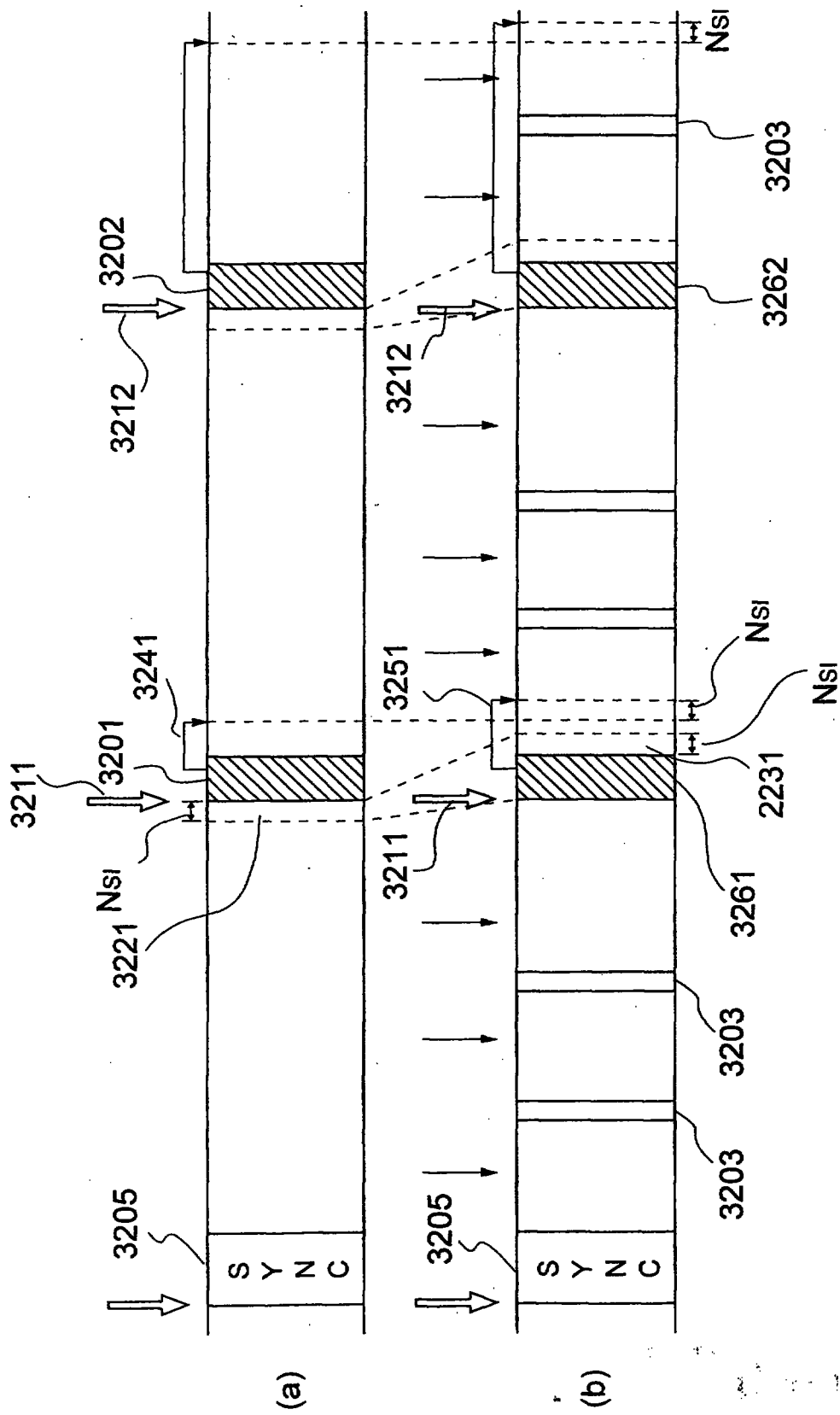


FIG. 35

世界知的所有権機関
国際事務局

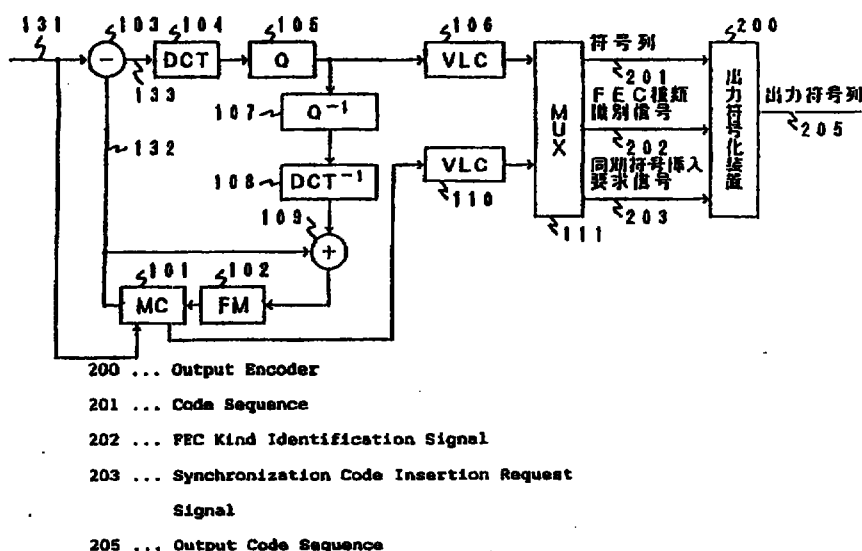
PCT

特許協力条約に基づいて公開された国際出願

(51) 国際特許分類6 H04L 7/08, H04N 7/64, H04L 1/00, H03M 7/30		A1	(11) 国際公開番号 WO97/35401
			(43) 国際公開日 1997年9月25日(25.09.97)
(21) 国際出願番号 PCT/JP97/00864		(71) 渡邊敏明(WATANABE, Toshiaki)[JP/JP] 〒245 神奈川県横浜市泉区白百合3-3-5 Kanagawa, (JP)	
(22) 国際出願日 1997年3月18日(18.03.97)		(72) 駄竹健志(DACHIKU, Kenshi)[JP/JP] 〒213 神奈川県川崎市高津区宇奈根590 ファール東芝101 Kanagawa, (JP)	
(30) 優先権データ		(74) 代理人 弁理士 佐藤一雄, 外(SATO, Kazuo et al.) 〒100 東京都千代田区丸の内三丁目2番3号 富士ビル323号 協和特許法律事務所 Tokyo, (JP)	
特願平8/61450 ✓ 1996年3月18日(18.03.96) JP			
特願平8/163082 ✓ 1996年6月24日(24.06.96) JP			
特願平8/232362 ✓ 1996年9月2日(02.09.96) JP			
特願平8/243883 ✓ 1996年9月13日(13.09.96) JP			
(60) 法律上関連する他の国内特許文献の表示		(81) 指定国 AU, BR, CA, CN, KR, MX, NO, SG, US, VN, 欧州特許 (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(63) 継続による関係		添付公開書類 国際調査報告書	
US 08/720,067(一部継続)			
出願日 1996年6月27日(27.06.96)			
(71) 出願人 (米国を除くすべての指定国について) 株式会社 東芝(KABUSHIKI KAISHA TOSHIBA)[JP/JP] 〒210 神奈川県川崎市幸区堀川町72 Kanagawa, (JP)			
(72) 発明者; および			
(75) 発明者/出願人 (米国についてのみ) 菊地義浩(KIKUCHI, Yoshihiro)[JP/JP] 〒232 神奈川県横浜市南区高砂町1-10-1 クレアーレ東芝吉野町604号 Kanagawa, (JP)			

(54) Title: ENCODER AND DECODER

(54) 発明の名称 符号化装置および復号化装置



(57) Abstract

The problems of pseudo-synchronization and out-of-synchronization caused by erroneous detection of synchronization codes of an encoder in which an error correction/detection coding and a synchronism restoring method using synchronization codes. There are provided a coding unit (212) for encoding a sequence of inputted multiplex codes (201) into error correction/detection codes comprising information bits and inspection bits; and a code sequence assembling unit (213) for assembling a sequence of output codes (205) by inserting a synchronization code into any of a plurality of periodically predetermined synchronization code insertion positions in the code sequence (201) and by arranging the information bits in arbitrary positions in the code sequence and the inspection bits in positions other than the synchronization code insertion positions in the code sequence (201).

(57) 要約

謝り訂正／検出符号化と同期符号を用いる同期回復手法とを組み合わせた符号化装置における同期符号の誤検出に起因する疑似同期や同期外れの問題を解決する。入力される多重化符号列201を情報ビットおよび検査ビットよりなる誤り訂正／検出符号に符号化する符号化部212と、符号列201中の予め周期的に定められた複数の同期符号挿入位置のいずれかに同期符号を挿入すると共に、情報ビットを符号列中の任意の位置に配置し、検査ビットを符号列201中の同期符号挿入位置以外の位置に配置して出力符号列205を組み立てる符号列組立部213を備える。

情報としての用途のみ

PCTに基づいて公開される国際出願をパンフレット第一頁にPCT加盟国を特定するために使用されるコード

AL	アルバニア	EE	エストニア	LR	リベリア	RU	ロシア連邦
AM	アルメニア	ES	スペイン	LS	レソト	SE	スウェーデン
AU	オーストラリア	FI	フィンランド	LT	リトアニア	SG	シンガポール
AZ	アゼルバイジャン	FR	フランス	LU	ルクセンブルグ	SI	スロベニア共和国
BB	バルバドス	GB	イギリス	LV	ラトヴィア	SK	スロバキア
BE	ベルギー	GE	グルジア	MC	モナコ	SN	セネガル
BG	ブルガリア	GH	ガーナ	MD	モルドバ	SZ	スワジランド
BR	ブラジル	GN	ギニア	MG	マダガスカル	TG	トーゴ
BY	ベラルーシ	GR	ギリシャ	MK	マケドニア	TH	タイ
CA	カナダ	HU	ハンガリー	UA	ウクライナ	TM	トルクメニスタン
CC	中絶	IE	アイルランド	US	アメリカ合衆国	TR	トルコ
CF	中央アフリカ共和国	IT	イタリア	UY	ウルグアイ	TT	トリニダード・トバゴ
CG	コンゴ	JP	日本	VE	ベネズエラ	UA	ウクライナ
CH	スイス	KE	ケニア	VI	ヴァチカン	UG	ウガンダ
CI	コート・ジボワール	KZ	キルギスタン	VG	ヴァージン諸島	US	アメリカ合衆国
CM	カメルーン	KR	朝鮮民主主義人民共和国	VI	ヴァージン諸島	UZ	ウズベキスタン共和国
CN	中国	KG	キルギスタン	VG	ヴァージン諸島	VN	ベトナム
CO	コロンビア	KH	カンボジア	VG	ヴァージン諸島	YU	ユーゴスラビア
DE	ドイツ	LI	リヒテンシュタイン	VG	ヴァージン諸島		
DK	デンマーク	LU	ルクセンブルグ	VG	ヴァージン諸島		