

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4545250号
(P4545250)

(45) 発行日 平成22年9月15日 (2010. 9. 15)

(24) 登録日 平成22年7月9日 (2010. 7. 9)

(51) Int. Cl.

H04N 7/32 (2006.01)

F I

H04N 7/137

Z

請求項の数 5 (全 16 頁)

(21) 出願番号 特願平11-197286
 (22) 出願日 平成11年7月12日 (1999. 7. 12)
 (65) 公開番号 特開2000-50284 (P2000-50284A)
 (43) 公開日 平成12年2月18日 (2000. 2. 18)
 審査請求日 平成18年7月11日 (2006. 7. 11)
 (31) 優先権主張番号 114568
 (32) 優先日 平成10年7月13日 (1998. 7. 13)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 504199127
 フリースケール セミコンダクター イン
 コーポレイテッド
 アメリカ合衆国 78729 テキサス州
 オースチン ウェスト パーマー レー
 ン 7700
 (74) 代理人 100116322
 弁理士 桑垣 衛
 (72) 発明者 デビッド・ジェイ・ハウキンス
 アメリカ合衆国テキサス州オースチン、ス
 モーキー・ヒル・ロード6904
 (72) 発明者 ブルース・エー・オウガスティン
 アメリカ合衆国イリノイ州レイク・イン・
 ザ・ヒルズ、バーハーバー・ドライブ46
 35

最終頁に続く

(54) 【発明の名称】 動きベクトルの格納および読出を用いることによりビデオ信号を符号化および解読するための方法および装置

(57) 【特許請求の範囲】

【請求項 1】

ビデオ信号の解読および符号化に用いられる動きベクトルを格納および読出する装置であって：

複数の格納要素を有するメモリであって、前記複数の格納要素の第1要素がビデオ・フレームのマクロブロックに関して第1動きベクトルを格納し、前記マクロブロックはビデオ・フレームを表す複数のマクロブロックの1つであり、さらに前記複数の格納要素の第2要素が前記マクロブロックに関する第2動きベクトルを格納し、前記第1動きベクトルおよび第2動きベクトルが同一のマクロブロックに関して格納されており、前記第1動きベクトルおよび前記第2動きベクトルが、ビデオ・アーチファクト除去のための動きベクトル予測子計算およびオーバーラップブロック動き補償の両方の間で共有される、前記メモリ；および

前記メモリに結合され、前記メモリ内で動きベクトルが格納される場所を制御するアドレス発生回路構成；

を具備し、前記メモリおよび前記アドレス発生回路構成が集積回路形式でハードウェア部品として実現され、

前記メモリは第1集合の動きベクトルと第2集合の動きベクトルを格納し、

前記第1集合の動きベクトルは第1の複数のマクロブロックの上部分に対応する複数の第1動きベクトルを含み、前記複数の第1動きベクトルは、現行のマクロブロックの処理に必要な第1動きベクトルのみを含むものであり、

10

20

前記第 2 集合の動きベクトルは、前記第 1 の複数のマクロブロックの下部分と、第 2 の複数のマクロブロックの下部分とに対応する複数の第 2 動きベクトルを含み、

前記第 1 集合の動きベクトルが前記第 2 集合の動きベクトルよりも少ない格納要素をメモリ内で占有することを特徴とする装置。

【請求項 2】

前記メモリに応答する予測子計算機をさらに具備することを特徴とする請求項 1 記載の装置。

【請求項 3】

ビデオ信号の解読および符号化に用いられる動きベクトルを格納および読出する方法であって：

メモリを設ける段階；

複数のマクロブロックに対応する第 1 集合の動きベクトルと第 2 集合の動きベクトルを受信する段階であって、前記第 1 集合の動きベクトルは、第 1 の複数のマクロブロックの上部分を表す複数の第 1 および第 2 動きベクトルを含み、前記複数の第 1 動きベクトルは、現行のマクロブロックの処理に必要な第 1 および第 2 動きベクトルのみを含むものであり、前記第 2 集合の動きベクトルは、前記第 1 の複数のマクロブロックの下部分と、第 2 の複数のマクロブロックの下部分とを表す複数の第 3 および第 4 動きベクトルを含み、各第 1 動きベクトルは、各マクロブロック内の左上ブロックであるブロック 1 を表し、各第 2 動きベクトルは各マクロブロック内の右上ブロックであるブロック 2 を表し、各第 3 動きベクトルは各マクロブロック内の左下ブロックであるブロック 3 を表し、各第 4 動きベクトルは各マクロブロック内の右下ブロックであるブロック 4 を表し、前記複数のマクロブロックがビデオ・フレームの一部を表す、前記複数の動きベクトルを受信する段階；および

前記第 1 集合の動きベクトルを前記メモリの第 1 部分に格納し、前記第 2 集合の動きベクトルを前記メモリの第 2 部分に格納する段階；

を具備し、前記第 1 集合の動きベクトルが前記第 2 集合の動きベクトルよりも少ない格納要素をメモリ内で占有することを特徴とする方法。

【請求項 4】

ビデオ信号をデコードするための装置であって：

ビデオ信号を受けるビットストリーム・デコーダ；

前記ビットストリーム・デコーダに応答する動きベクトル格納および読出モジュールであって、該動きベクトル格納および読出モジュールは複数のマクロブロックを表す第 1 集合の動きベクトル及び第 2 集合の動きベクトルを格納するメモリを含み、前記複数のマクロブロックは前記ビデオ信号の一部を表し、前記メモリにおける前記第 1 及び第 2 の集合の動きベクトルの一部は少なくとも 1 つの所定のマクロブロック内の複数のブロックを表し、前記少なくとも 1 つの所定のマクロブロックは前記ビデオ信号の一部を表し、前記第 1 集合の動きベクトルは、第 1 の複数のマクロブロックの上部分に対応する複数の動きベクトルを含み、前記上部分に対応する複数の動きベクトルは、現行のマクロブロックの処理に必要な上部分に対応する動きベクトルのみを含むものであり、前記第 2 集合の動きベクトルは、前記第 1 の複数のマクロブロックの下部分と、第 2 の複数のマクロブロックの下部分とに対応する複数の動きベクトルを含み、前記第 1 集合の動きベクトルが前記第 2 集合の動きベクトルよりも少ないメモリを必要とする、前記動きベクトル格納および読出モジュール；

前記ビットストリーム・デコーダに応答する変換をベースとした圧縮モジュール；

前記動きベクトル格納および読出モジュールに応答する動き補償ユニット；および

前記動き補償ユニットおよび変換をベースとした圧縮モジュールに応答する合成器；

を具備することを特徴とするビデオ信号をデコードするための装置。

【請求項 5】

ビデオ信号をエンコードするための装置であって：

動き推定モジュール；

前記動き推定モジュールに回答する動きベクトル格納および読出モジュールであって、該動きベクトル格納および読出モジュールはマクロブロックに対応する第1集合の動きベクトルと第2集合の動きベクトルとを格納するメモリを含み、前記マクロブロックは前記ビデオ信号の一部を表し、前記メモリにおける前記第1及び第2集合の動きベクトルの一部は少なくとも1つの所定のマクロブロック内の複数のブロックを表し、前記少なくとも1つの所定のマクロブロックは前記ビデオ信号の一部を表し、前記第1集合の動きベクトルは、第1の複数のマクロブロックの上部分に対応する複数の動きベクトルを含み、前記上部分に対応する複数の動きベクトルは、現行のマクロブロックの処理に必要な上部分に対応する動きベクトルのみを含むものであり、前記第2集合の動きベクトルは、前記第1の複数のマクロブロックの下部分と、第2の複数のマクロブロックの下部分とに対応する複数の動きベクトルを含み、前記第1集合の動きベクトルは前記第2集合の動きベクトルよりも少ないメモリを必要とする、前記動きベクトル格納および読出モジュール；

10

前記動きベクトル格納および読出モジュールに回答する動き補償ユニット；

前記動き補償ユニットに回答する変換をベースとした圧縮モジュール；および

前記変換をベースとした圧縮モジュールに回答するビットストリーム・エンコーダ；

を具備することを特徴とするビデオ信号をエンコードするための装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、一般にビデオ通信に関し、さらに詳しくは、ビデオ画像の圧縮および圧縮解除に関する。

20

【0002】

【従来の技術および発明が解決しようとする課題】

事業用および家庭用に用いられる多くのビデオ電話およびビデオ会議装置は、現在の電話網に固有の帯域幅上の制約により高い圧縮比を必要とする。この要望に応じて、低ビット速度ビデオ通信に関するH.263国際規格(H.263 International Standard for Low Bit Rate Video Communications)が開発された。この規格は、直ちにビデオ産業に採用された。H.263国際規格には、高度予測モード(APM: Advanced Prediction Mode)と呼ばれる任意の技術の実行が含まれる。APMは、動きの推定中に導入されるアーチファクトを排除することによりビデオ画像が大幅に改善する。APMの欠点は、ハードウェア内に組み込むのが極めて困難で非効率的なことである。その結果、高度予測モードが最初に実現されたのはソフトウェア内のみであった。

30

【0003】

ビデオを圧縮するH.263技術の1つは、動き推定(motion estimation)を用いることである。動き推定は、実際の画素の代わりに動きベクトルをデコーダに送ることによりビデオを処理する。デコーダは、この動きベクトルを取り込み、それを用いて前に解読されたフレームから動きベクトルが指示するデータのブロックにアクセスすることで現行のフレームを再生する。H.263 APMモードにおいては、それぞれ8×8ブロックの画素の代わりに1つの動きベクトルを送ればよい。APMオプションは、それぞれ8×8ブロックを表す動きベクトルを周辺ブロックの動きベクトルで平均化することを求める。これにより、動き推定プロセス中に導入されるブロック化アーチファクトが軽減される。残念ながら、周辺のマクロブロックから動きベクトルを平均化することは、いくつかの理由から、メモリにおいてもハードウェアにおいても集約的になることがわかっている。第1に、周辺ブロックから動きベクトルを読出するには、RAM内に多数の動きベクトルを格納しなければならないことになる。現行のブロック上に動きベクトルを読出しなければならないということは、動きベクトルの行全体を格納することが必要となり、これは極めてメモリ集約的になる。第2に、H.263規格は、2つの別々の関数に関して動きベクトルを格納および読出することを必要とするものとAPMモードを定義する：すなわち、オーバーラップブロック動き補償(OBMC: Overlapped

40

50

Block Motion Compensation)および予測子計算(predictor calculation)である。これらの関数は、容易に合成されない。なぜなら、OBMC関数は解読時間の1マクロブロック分だけ予測子関数より遅延するためである。これにより、2つの関数は異なる時間フレーム内で動作することを強いられる。第3に、OBMCも予測子計算も、それぞれ4つの異なるアドレス・パターンを必要とする。従って、必要なアドレス化スキームは8個になる。しかし、アドレス化スキームのうち線形順序のものは1つもなく、容易にアドレス化可能な他のパターンのものもない。

【0004】

上記の問題のために、H.263規格で定義されるAPMモードのハードウェアを効率的に実行することは極めて困難である。APMモードをソフトウェアで実行することは、この機能を実現するために必要とされる大量のメモリおよびシステム資源の使用可能性のために有利である。しかし、ソフトウェアの実行には、実行速度がはるかに遅く、ハードウェア実行により得られるフレーム速度と画像の品質が犠牲になるという欠点がある。

【0005】

【実施例】

図1には、本発明の好適な実施例によるビデオ・コーデック(エンコーダ/デコーダ10)装置のブロック図を示す。本発明は、上記の問題点を克服し、APMを効率的にシリコン内に実現することを可能にする。ビデオA-D変換器およびデコーダ12は、アナログ・ビデオ信号を受信する入力と、4:2:2のビデオ・フォーマットの信号を提供する出力とを有する。4:2:2信号は、ビデオ・プレプロセッサ14の入力に接続され、プレプロセッサ14は4:2:0のビデオ・フォーマットの信号を提供する。ビデオ・プレプロセッサ14の出力はビデオ・エンコーダ16の入力に接続される。エンコーダ16については図2で詳細に説明する。ビデオ・エンコーダ16は、圧縮されたビデオ・ビットストリームを入力/出力(I/O)インタフェース回路18に提供する。I/Oインタフェース18は、制御、アドレスおよびデータ・バスを介して信号プロセッサ20に結合される。I/Oインタフェース18は、被圧縮ビデオ・ビットストリームを信号プロセッサ20で利用される特定種類の信号プロセッサとインタフェースする。I/Oインタフェース18は、被圧縮ビデオ・ビットストリームをビデオ・デコーダ22に与えるビデオ・デコーダ22の入力にも結合される。ビデオ・デコーダ22の出力は、ポスト・プロセッサ24の入力に接続される。ビデオ・デコーダ22の出力は4:2:0のビデオ・フォーマットである。ポスト・プロセッサ24の出力は、4:2:2のビデオ・フォーマットで、ビデオD-A変換器およびエンコーダ26の入力に接続される。ビデオD-A変換器およびエンコーダ26の出力は、アナログ・ビデオ出力を提供する。またさらに、直接メモリ・アクセス(DMA: direct memory access)/ダイナミック・ランダム・アクセス・メモリ(DRAM: dynamic random access memory)インタフェース回路28が図示される。DMA/DRAMインタフェース回路28は、内部バス32を介して、ビデオ・プレプロセッサ14、ビデオ・エンコーダ16、ビデオ・デコーダ22およびポスト・プロセッサ24の各々に結合される。DMA/DRAMインタフェース回路28は、制御、アドレスおよびデータ・バスを介してDRAM30に結合される。ビデオ・コーデック10は、アナログ・ビデオ信号を受信し、それをビデオA-D変換器およびデコーダ12とビデオ・プレプロセッサ14とを介して2つの異なるビデオ・フォーマットに変換し、4:2:0信号をビデオ・エンコーダに与えるよう機能し、ビデオ・エンコーダが信号を被圧縮ビデオ・ビットストリームに圧縮する働きをする。被圧縮ビデオ・ビットストリームは、信号プロセッサ20により利用され、信号プロセッサ20はホストとして機能して被圧縮ビデオ・ビットストリームをチャンネルに与えて送信する。どのタイプのチャンネルが用いられるかにより、信号プロセッサ20は被圧縮ビデオ・ビットストリームをアナログ・フォーマットに変換するか、あるいはデジタル・フォーマットのままにする。I/Oインタフェース18も被圧縮ビデオ・ビットストリームを提供する機能を果たすが、この被圧縮ビデオ・ビットストリームは信号プロセッサ20を介してチャンネル上で、ビデオ・デコーダ22により受信される。ビデオ・デコー

10

20

30

40

50

ダ 2 2 は、ビデオ・ビットストリームを、ポスト・プロセッサ 2 4 が用いる 4 : 2 : 0 のビデオ・フォーマットに圧縮解除する。ポスト・プロセッサ 2 4 は、4 : 2 : 0 のフォーマットを、ビデオ D / A 変換器 2 6 により用いられる 4 : 2 : 2 のフォーマットに変換する。ビデオ D / A 変換器 2 6 は、従来のテレビジョンまたはビデオ・モニタが受け入れることのできるアナログ出力を提供する。ビデオ・コーデック 1 0 の形式は、図示される如く実質的には従来通りである。言い換えると、既存の集積回路製品を用いて、ビデオ A - D 変換器およびデコーダ 1 2 , ビデオ・プレプロセッサ 1 4 , ポスト・プロセッサ 2 4 およびビデオ D - A 変換器およびエンコーダ 2 6 を実現することができる。しかし、ビデオ・エンコーダ 1 6 およびビデオ・デコーダ 2 2 の実行例と効率性は、用途により変わる。本発明が扱うのは、ビデオ・エンコーダ 1 6 およびビデオ・デコーダ 2 2 の特定の執行例である。

10

【 0 0 0 6 】

図 2 には、図 1 のビデオ・エンコーダ 1 6 の詳細なブロック図を示す。ビデオ・エンコーダ 1 6 は、一般に、4 : 2 : 0 のビデオ・フォーマットでビデオ信号を受信する入力を有する動き推定器回路 3 6 によって構成される。動き推定器 3 6 の出力は、ベクトル格納読出モジュール 3 8 に複数の動きベクトルを提供する。モジュール 3 8 は、動き補償および変位フレーム差回路 4 2 の入力に接続される出力を有する。動き補償および変位フレーム差回路 4 2 の出力は、離散コサイン変換および逆離散コサイン変換 (D C T : D i s c r e t e C o s i n e T r a n s f o r m / I D T : I n v e r s e D i s c r e t e C o s i n e T r a n s f o r m) 回路 4 4 またはモジュールの入力に接続される。一般に、I D T / D C T 回路 4 4 は、ウェーブレットまたは離散コサイン変換などの圧縮 / 圧縮解除アルゴリズムに基づき変換を実行する。離散コサイン変換および逆 D C T 回路 4 4 の出力は、被圧縮ビデオ出力ビデオストリームを提供するビットストリーム・エンコーダ 4 6 の入力に接続される。速度制御回路 4 8 は、これも図 1 に図示される内部バス 3 2 を介して、動き推定器 3 6 , 動き補償および変位フレーム差回路 4 2 および離散コサイン変換および逆 D C T 回路 4 4 の各々に接続される。

20

【 0 0 0 7 】

動作中は、ビデオ・エンコーダ 1 6 は、前回のフレームの画素情報から現行フレームの動き推定を実行することにより機能する。動き推定器 3 6 が、複数の動きベクトルを提供する。この動きベクトルは、前回のフレームと現行のフレームとの間の 8 x 8 画素ブロックの変位を表す。動きベクトル格納読出モジュール 3 8 が動きベクトルを受信する。図 3 に関連してさらに説明されるが、モジュール 3 8 は H . 2 6 3 国際規格などのビデオ規格において高度予測モードを実現するために必要とされる動きベクトルを読出しおよび格納する働きをする。モジュール 3 8 がこのような動きベクトルを読出し格納すると、モジュール 3 8 はその出力において 2 種類の動きベクトルを動き補償および変位フレーム差回路 4 2 に提供する。提供される第 1 種類の動きベクトルがオーバーラップブロック動き補償に用いられる。提供される第 2 種類の動きベクトルは、前回のフレーム再構築および動き補償に用いられる動きベクトルである。動き補償および変位差フレーム回路 4 2 は、これら 2 種類の動きベクトルを取り入れ、2 つの関数を実行する：すなわち、変位フレーム差および動き補償である。動き補償関数は、上記の 2 種類の動きベクトルを用いて、遠隔のデコーダの解読を複製する。動き補償関数は、バス 3 2 を介して動き推定器 3 6 に関して再構築される基準フレームを提供する。このフレームは、前フレームまたは再構築フレームとも呼ばれ、遠隔デコーダにより解読されるものの正確な複製である。これがビデオ・エンコーダ 1 6 が次のフレームを符号化する際に動き推定の基準フレームとして動き推定器 3 6 により用いられる。変位フレーム差関数は、当初の (基準) フレームから動き推定されるフレームを差し引き、その差を離散コサイン変換および逆 D C T 回路 4 4 に与える。離散コサイン変換および逆 D C T 回路 4 4 は、圧縮関数に基づき損失の多い変換を行う。D C T / I D C T 4 4 の出力と動き推定器 3 6 からの動きベクトルとがビットストリーム・エンコーダ 4 6 に与えられる。ビットストリーム・エンコーダ 4 6 は、H . 2 6 3 に準じてデータを配列し、圧縮されたビデオ・ビットストリーム出力をチャンネルに提供する。チ

30

40

50

チャンネル上に送付されるだけでなく、DCT/I DCT回路44の圧縮出力は、同じDCT/I DCT回路44の逆変換関数を用いて圧縮解除される。この圧縮解除された出力は、次にバス32を介して、同じ動き補償および変位フレーム差回路42に送られる。この情報は、変位差フレーム関数により用いられ、その前の減算を取り消し、動き推定器36の基準フレームを再度作成する。これはチャンネルの他端で遠隔デコーダにより解読されるものの正確な複製である。

【0008】

図3には、本発明による図2の動きベクトル格納読出モジュール38を示す。動きベクトル・レジスタ50は、現在の動きベクトルを動きベクトルSRAMアレイ52に提供する出力を有する。この動きベクトルSRAMアレイ52は、より一般的には動きベクトルを格納する複数の格納要素と呼ばれる。ベース・アドレスを生成するベース・アドレス・カウンタ54がアドレス発生器56に接続される。アドレス発生器56は、ルックアップ・テーブル58からオフセット値を受信する第2入力を有する。ルックアップ・テーブル58は、ブロック番号を受信する第1入力を有する。ブロック番号は、多くのビデオ規格で定義されるマクロブロック内の4つの可能なブロックのうちの1つを表す。ルックアップ・テーブル58は、H.263国際規格に必要なOBMCまたは予測子計算モードのいずれか一方を実現するためのOBMC/予測子モード信号を受信する第2入力を有する。ルックアップ・テーブル58は、所望の動きベクトル(左, 上, 右上, 右, 下)を選択する第3入力を有する。これについては図5および図6において詳細に説明する。状態装置は4つの可能な選択肢(左, 上, 右上, 右下)を循環し、新しいマクロブロックが符号化されると最初に戻ってリセットされる。モジュール加算器57は、ルックアップ・テーブル58からオフセット値を受信する第1入力と、ベース・アドレス・カウンタ54からベース・アドレスを受信する第2入力と、動きベクトルのSRAMアレイ52に接続されるアドレスを提供する出力とを有する。SRAMアレイ52は、選択される動きベクトルを乗算器回路60に与える出力を有する。乗算器60は、予測子計算機回路62の入力に接続される第1出力を有する。乗算器60は、OBMC/予測モード制御信号に応答して機能する。この信号は、ルックアップ・テーブル58において適切なオフセット値を選択する際にも用いられる。乗算器60は、動き補償ユニット42の入力に接続される第2出力を有する。動き補償ユニット42は、図2に図示されるDCT/I DCT回路44に信号を提供する出力を有する。動きベクトル格納読出モジュール38の動作を説明する前に、図4ないし図6に関して説明する。

【0009】

図4に、タイプ共通中間フォーマット(CIF)のビデオ・フレーム内の1行のマクロブロック・レイアウトを示す。マクロブロック・レイアウトは複数の行のマクロブロックを有する。図面の便宜上、1行のマクロブロックのみを詳細に示す。各行は22個のマクロブロックを有し、それぞれにC0~C-21とラベルが付けられる。図示される形式では、マクロブロックC0は現行のマクロブロックを時間的に示す。マクロブロックC0のすぐ上にはマクロブロックC-22があり、これは時間領域において、過去の22個のマクロブロックからのマクロブロックである。ハードウェアおよびメモリを最大に節約するためには、OBMCおよび予測子関数のために用いられるメモリを共有および合成することが必要である。OBMC関数は、予測子関数に関して、時間にして1マクロブロック分だけ遅延しなければならない。この事実により、OBMC関数はマクロブロックC-1で動作(動きベクトルの格納および読出)を行うことになり、一方予測子関数はマクロブロックC0で動作する。OBMC関数をマクロブロックC-1まで遅延させることが、OBMC関数が動作するマクロブロックのすぐ右側(図6に関して詳細に説明する)の動きベクトルを読出する必要があるOBMC要件のために必要である。OBMC関数がマクロブロックC0で動作すると、「右側」の動きベクトルをマクロブロックC+1から読出しなければならない。これは1ブロックだけ将来のマクロブロックであり、まだ使用可能になっていない。図5および図6に関してさらに説明するように、OBMCおよび予測子関数の「上の」マクロブロックから動きベクトルを読出することも必要である。「上」の動き

10

20

30

40

50

ベクトルは、OBMC関数に関しては23マクロブロック分だけ過去において処理されたものであり、予測子関数に関しては22マクロブロック分だけ過去に処理されたものであることに注目することが重要である。このため、24個(C0~C-23)のマクロブロックの動きベクトルを格納することが必要になる。

【0010】

図5および図6においては次のような命名法を採用する：16×16画素マクロブロックを表す大きな正方形が4個のより小さな8×8画素ブロックに分割される。8×8画素ブロックの各々は、それに関連する自身の動きベクトルを有する。4個の8×8画素ブロックは、当技術では通常「ブロック」と呼ばれるが、ブロック1, 2, 3, 4と番号が振られる。ブロック1はマクロブロックの左上隅であり、ブロック2が右上隅、ブロック3が左下隅、ブロック4が右下隅である。マクロブロックの処理とは、そのマクロブロックの符号化または解読を指す。図5および図6における「C-1」の名前は、1マクロブロック分だけ過去に処理されたマクロブロックを指す。同様に、マクロブロック「C-21」は、21マクロブロック分だけ過去に処理されたマクロブロックを指す。「C0」は現在処理中のマクロブロックを指す。図5においては、マクロブロックC0内でブロック1について読出すべき動きベクトルは左, 上および右上であることに注目することが重要である。左動きベクトルはマクロブロックC-1内のブロック2のものである。上動きベクトルはマクロブロックC-22内のブロック3のものである。右上動きベクトルはマクロブロックC-21内のブロック3のものである。現在処理中のマクロブロック内のブロック2に関する読出パターンは次のようになる：左動きベクトルが現在処理中のブロック(C0)のブロック1である。上動きベクトルはマクロブロックC-22のブロック4である。右上動きベクトルはマクロブロックC-21のブロック3である。現在処理中のマクロブロック内のブロック3に関する読出パターンもブロック1, 2, 3, 4とは異なる。現行ブロック内のブロック3の読出パターンは次のようになる：左動きベクトルがマクロブロックC-1のブロック4である。上動きベクトルは現在処理中のマクロブロック(C0)のブロック1であり、右上動きベクトルは現在処理中のマクロブロック(C0)のブロック2である。現在処理中のマクロブロック内のブロック4については、左動きベクトルが現行マクロブロックC0内のブロック3である。上動きベクトルは現行マクロブロックC0のブロック1であり、右上動きベクトルは現行マクロブロックC0内のブロック2である。この場合も、ブロック1, 2, 3, 4に関する読出パターンが互いに異なることに注目することが重要である。これらのパターンは、H.263規格において定義される。

【0011】

本発明は、OBMC関数の動きベクトルの読出にも対応する。この関数も同様にマクロブロック内の各ブロックについて別々の読出パターンを有する。OBMC関数は、マクロブロック内の各ブロックについて左, 上, 右および下動きベクトルを読出する必要がある。この場合も、右に動きベクトルを読出する要件は現行のマクロブロックの右の動きベクトルがまだ処理されていないために困難であることに注目することが重要である。右動きベクトルを読出するためには、OBMC関数を1マクロブロック分だけ時間的に遅延させることが必要である。従って、OBMC関数は、予測子関数より1マクロブロック分だけ後にくるマクロブロックC-1で動作する。OBMC関数の読出パターンは次のようになる：ブロック1については、左動きベクトルがマクロブロックC-2のブロック2である。上動きベクトルがマクロブロックC-23のブロック3である。右動きベクトルはマクロブロックC-1のブロック2であり、下動きベクトルはマクロブロックC-1のブロック3である。処理中の同じOBMC動きベクトル内のブロック2については、左動きベクトルがマクロブロックC-1内のブロック1である。上動きベクトルはマクロブロックC-23のブロック4である。右動きベクトルはマクロブロックC0のブロック1であり、下動きベクトルはマクロブロックC-1のブロック4である。OBMC関数のブロック3については、左動きベクトルがマクロブロックC-2のブロック4である。上動きベクトルはマクロブロックC-1のブロック1である。右動きベクトルはマクロブロックC-1

のブロック4であり、下動きベクトルはマクロブロックC - 1のブロック3である。OBMCマクロブロックC - 1のブロック4については、左動きベクトルがマクロブロックC - 1のブロック3である。上動きベクトルはマクロブロックC - 1のブロック2である。右動きベクトルはマクロブロックC 0のブロック3であり、下動きベクトルはマクロブロックC - 1のブロック4である。

【0012】

図3に戻り、動きベクトル格納読出モジュール38は次のように機能する。図5および図6に関して上述されるOBMCおよび予測子モードの両方の8個のアドレス化スキームのオフセットがルックアップ・テーブル58を介して提供される。ルックアップ・テーブル58は、その入力としてマクロブロック内のブロック番号(1, 2, 3, 4)と、必要な動きベクトル(右, 左など)と、動きベクトル読出の適切なオフセットを選択するための関数モード(OBMC / 予測子)とを取り入れる。

【0013】

実行される最初のモードは予測モードである。予測モードを実行するには、図5を参照する。図5は、H.263国際規格により定義される4つのマクロブロック構造を示す。各マクロブロックはマクロブロック内に8x8ブロックを表す4つの動きベクトルを有する。予測モードについては、H.263国際規格はマクロブロックに含まれる4つのブロックの各々に関して左, 上および右上動きベクトルの読出を必要とする。各ブロックの左, 上および右上動きベクトルの読出パターンは、マクロブロック内の個々のブロック1, 2, 3, 4に関して異なる。パターンを実現するために、カウンタ54がゼロから53までの間の数値をアドレス発生器56に提供する。アドレス発生器56は、カウンタの出力に2を掛けてベース・アドレスを得る。この実行例で2を掛ける乗算が必要とされるのは、2つの動きベクトルが各計数において格納されるためである。ルックアップ・テーブル58はブロック番号、モードおよび動きベクトル選択に基づきオフセット値を出力する。モジュール加算器57は、ルックアップ・テーブル58が与えるオフセット値をカウンタ54が与えるベース・アドレスに加算する。その和がSRAMアレイ52との通信においてポインタとして用いられるアドレスとなる。このアドレスは、H.263規格の読出パターンにより適切な左, 上および右上動きベクトルを読出するために用いられる。SRAMアレイ52内でアドレス指定される左, 上および右上動きベクトルが、SRAMアレイ52により乗算器60(Mux)を介して選択的に出力される。選択される動きベクトルは予測子計算機62に与えられる。予測子計算機62は、選択される上, 右上および左動きベクトルのメジアン(中央値)を計算するよう機能する。それに由来するメジアンは、通常「予測子」と呼ばれ、ビットストリーム・エンコーダ46に提供される。ここで予測子を動き推定器36により提供される動きベクトルから減じる。その結果、当技術では「差」と呼ばれるが、差がチャンネル上に送付される。予測子計算機62の実行例は当技術では周知である。

【0014】

OBMCモードにおいては、ルックアップ・テーブル58がH.263規格に準拠するOBMC用動きベクトル読出パターン(図6)に従うオフセット値を提供する。アドレス発生器56は、予測子計算に関する上記の説明と同様に機能して、アドレスがSRAMアレイ52にアドレスが送られ、アレイ52は、選択した動きベクトルを乗算器60に送る。乗算器60から、OBMC動きベクトルが動き補償ユニット42に乗算器60を介して与えられる。動き補償ユニットはOBMC動きベクトルを利用して動き推定器のために再構築フレームを構築する。

【0015】

マクロブロックの上部の動きベクトル、すなわちブロック1, 2はマクロブロックC - 2を超えてセーブする必要がないことに注目すると、RAMにおいて大幅な節約を実現することができることを理解頂きたい。これを理解するには、図6のブロック1(OBMC関数)を参照されたい。図6は、上半分または部分(T-x)の動きベクトルの集合が必要とされる最後のときがマクロブロックC - 1内のブロック1の左動きベクトルの読出の

ためであることを示す。従って、下半分または部分動きベクトル ($B - x$) の集合のみがマクロブロック $C - 2$ の後は必要になる。これは、「上」動きベクトルを得るためには、処理中のマクロブロック上のマクロブロックからの下半分または部分動きベクトルのみが必要となるからである。

【 0 0 1 6 】

図 7 には、S R A M アレイ 5 2 の 1 つの形態に関するメモリ割当の例が示される。図示される形態においては、27 個の半マクロブロックのうち複数のものが提供され、各半マクロブロックまたは部分にはマクロブロックの下または上動きベクトルが含まれる。メモリ・レイアウトにおいては、24 個の下半分のマクロブロックと、3 個の上半分のマクロブロックとがある。マクロブロックの上半分はマクロブロックのブロック 1, 2 の動きベクトルを格納する。マクロブロックの下半分は、マクロブロックのブロック 3, 4 の動きベクトルを格納する。さらに詳しくは、48 個のアドレス位置が下動きベクトルに割り付けられ、6 個のアドレス位置が上動きベクトルに割り付けられる。この割付は、下動きベクトルは 23 マクロブロック分の時間格納する必要がある、上動きベクトルは 3 マクロブロック分の時間しか格納する必要があるという事実に対応する。図 7 で用いられる命名法は次の通りである：B 0 は現在処理中のマクロブロックを表し、現在解読中の下動きベクトル (ブロック 3, 4) が格納される場所である。B - 1 は、1 マクロブロック分だけ過去に処理されたマクロブロックの下動きベクトルが格納される場所である。同様に、T 0 は現行のマクロブロックの上動きベクトル (ブロック 1, 2) が格納される場所であり、T - 1 は 1 マクロブロック分だけ過去に処理されたマクロブロックの上動きベクトルが格納される場所である。これはスライディング / 相対アドレス指定スキームであり、絶対的なアドレス指定ではないことに留意することが重要である。B 0, B - 1 マクロブロックは絶対アドレス・スキームの場合のように絶対アドレス 0 および絶対アドレス 2 にコード化されることはない。ベース・アドレスは常に B 0 の左動きベクトル (現在処理中のマクロブロックの下動きベクトル) を指示することに留意されたい。これは、2 に図 3 のベース・アドレス・カウンタ 54 内の値を掛けることにより計算されるのと同じベース・アドレスである。カウンタが進み、新しいマクロブロックの処理を開始すると、上動きベクトル T 0 が古くなり上動きベクトル T - 1 になることに留意することが重要である。新しい上動きベクトル T 0 が下動きベクトル B - 23 を書き換える。同様に、新しい下動きベクトル B 0 は古い上動きベクトル T - 2 を書き換える。このように、下動きベクトルは常に上動きベクトルを書き換え、上動きベクトルは常に下動きベクトルを書き換える。上および下の動きベクトルに割り付けられる格納場所が 2 つの異なるタイムラインを形成することにも注目されたい。

【 0 0 1 7 】

上記の概念の例を以下に示す。この例では、カウンタの現在値が 5 のとき S R A M アレイ 5 2 のベース・アドレス 10 は現行マクロブロックの下動きベクトルが格納される場所であることを示す。現行マクロブロックの上動きベクトルは常にカウンタより 6 アドレス分だけ先に、この例ではアドレス 16 に格納されることになる。その理由は、図 7 のメモリ割付を注意深く見るとわかる。現行の計数は常に B 0 を指示し、T 0 のスライディング・アドレス空間は 6 をモジュロ加算して T 0 に統合すると得られる。次のマクロブロックを処理して、計数が 6 に進むと、全部が 1 マクロブロックだけ「古くなる」。ベース・アドレス 10 は過去 1 マクロブロックからの下動きベクトルを含み、ベース・アドレス 12 は新しい現行のマクロブロックの下動きベクトルを含む。ベース・アドレス 12 がマクロブロック $C - 2$ ($C - 2$ は上動きベクトル T - 2 および下動きベクトル B - 2 を含むマクロブロックである) の上動きベクトルを以前は含んだことに留意することが重要である。カウンタが 6 に進んだときに、3 マクロブロック時間の間しか格納する必要のない T - 2 動きベクトルが終了し、下動きベクトル B 0 が代わりに書き込まれた。

【表 1】

例:

カウンタ値 5 = ベース・アドレス 10							
ベース・アドレス							
0 ... 4	6	8	10	12	14	16	18 ... 52
B-2 B-1 B0 T-2 T-1 T0 B-23							
カウンタ値 6 = ベース・アドレス 12							
ベース・アドレス							
0 ... 4	6	8	10	12	14	16	18 ... 52
B-3 B-2 B-1 B0 T-2 T-1 T0							

要するに、動きベクトル格納読出モジュール 3 8 全体の動作が次の例に基づき説明される：ベース・アドレス・カウンタ 5 4 が 6 の値を有するとすると、それが指す実際のアドレスは 1 2 になる。これは、各ベース・アドレス位置に 2 つの動きベクトルが含まれるためである。従って、実際のアドレス 1 2 が、B 0 の左動きベクトルが格納される場所となる。ルックアップ・テーブル 5 8 の入力が：ブロック番号 = 1 , O B M C / 予測モード = O B M C であるとする。するとブロック C - 1 の左, 上, 右および下 O B M C 動きベクトルは次のように読出される：

【表 2】

MV選択	位置	LUTアウト	ベース・アドレス	最終アドレス
左	T-2	+3	12	15
上	B-23	+8	12	20
右	T-1	+5	12	17
下	B-1	-2	12	10

上記の例においては、図 6 を用いて、ブロック 1 の O B M C 左動きベクトルがマクロブロック C - 2 の右上ブロック、メモリ位置 T - 2 にあることがわかる。図 7 または上記の例を用いて、オフセットを決定することができる。この場合、T - 2 の右動きベクトルは、3 をモジュール加算することにより得られる。同様に、図 6 では、右動きベクトルがメモリ・セグメント T - 1 の右側にあり、上動きベクトルはメモリ・セグメント B - 2 3 の左側にあり、下動きベクトルはメモリ・セグメント B - 1 の左側にあることがわかる。

【0 0 1 8】

本発明の装置および方法が符号化動作に関して説明されたので、解読動作における本発明の使用を説明する。図 1 のデコーダ部分は、ビデオ D - A 変換器およびエンコーダ 2 6 , ポスト・プロセッサ 2 4 およびビデオ・デコーダ 2 2 を含む。

【0 0 1 9】

図 8 には、ビデオ・デコーダ 2 2 の詳細が図示される。ビデオ・ビットストリームがビットストリーム・デコーダ 7 0 に入力として与えられる。ビットストリーム・デコーダ 7 0 は、解読用動きベクトル格納読出モジュール 7 2 の入力に接続される第 1 出力を有する。ビットストリーム・デコーダ 7 0 の第 2 出力は、図 2 に関連して前述された離散コサイン変換および逆 D C T 4 4 の入力に接続される。解読用動きベクトル格納読出モジュール 7 2 の出力が、図 2 に関して前述された動き補償回路 4 2 に接続される。動き補償回路 4 2 の出力は加算器 7 6 と呼ばれる合成器回路の第 1 入力に接続される。離散コサイン変換および逆 D C T 4 4 の出力は加算器 7 6 の第 2 入力に接続される。加算器 7 6 の出力は、ビデオ・フォーマット 4 : 2 : 0 で信号を提供する。ポスト・プロセッサ 2 4 の出力は、

ビデオD/Aエンコーダ26の入力に接続される。ポスト・プロセッサ24はビデオ・フォーマット4:2:2で信号を提供する。ビデオD/Aエンコーダ26の出力は、図1に図示されるのと同じ信号であるビデオ・アナログ信号を提供する。

【0020】

動作中、デジタル・ビデオ信号がビットストリーム・デコーダ70に入力され、デコーダ70は動きベクトル差と画素データとをビデオ・ビットストリームから分離する。動きベクトル差は解読用動きベクトル格納読出モジュール回路72に送られる。これについては図9にさらに説明する。解読回路72の出力動きベクトルが動き補償回路42に送られる。動き補償が実行されると、次に、それに由来する画素が離散コサイン変換および逆DCT回路44からの出力と加算器76により加算される。ビットストリーム・デコーダ70により提供される画素データが離散コサイン変換および逆DCT44により処理され、変換ベース圧縮解除を実行する。離散コサイン変換および逆DCT回路44の出力は、第2入力として加算器76に与えられる。加算器76は4:2:0のビデオ・フォーマットで信号をポスト・プロセッサ24に送り、ポスト・プロセッサ24がデジタル・アナログ変換に先立って後処理を実行する。D/A変換が実行され、ビデオ・アナログ出力信号を提供する。

【0021】

図9には、本発明による図8の解読用動きベクトル格納読出モジュール回路72が図示される。図3に関連する同じ要素に対する同じ参照番号が図9でも用いられるが、図示されるハードウェア実行例に関して容易に複製される類似の回路構成を示すためにダッシュ()が付けられる。言い換えると、動き補償ユニット42などのように同一番号を持つ要素にダッシュが用いられない場合は、同一の要素を図3と図9の機能の間で容易に時間的に共有することができる。図3と図9との間の類似性の結果として、注目すべき主要な差は加算器回路80が追加されたことである。回路80は動きベクトルレジスタ50'から出力を受け取る第1入力と、予測子計算機62'から出力を受け取る第2入力とを有する。加算器80の出力は、完全な動きベクトル(予測子および差の和)を与え、これは動きベクトルのSRAMアレイ52'により格納されるために用いられる。解読用動きベクトル格納読出モジュール回路72の動作は、図2の動きベクトル格納読出モジュール38の動作と類似であり、前述された。従って、解読用動きベクトル格納読出モジュール回路72の動作の詳細説明は繰り返さない。類似の動作に対する顕著な違いは、解読モードにある動きベクトル差レジスタ50'が、図3のレジスタ50が格納する完全な動きベクトルではなく、実際には動きベクトル差を格納することである。さらに、動きベクトルは図3と同様に動きベクトルのSRAMアレイ52'に格納しなければならない。SRAMアレイ52'に格納するために動きベクトルを得るには、予測子計算機62'からの予測子に、動きベクトルレジスタ50'からの動きベクトル差を加えて、動きベクトルのSRAMアレイ52'内に格納する動きベクトルを得なければならない。これらの動きベクトルの格納が、第2マクロブロックに対応する第2群の動きベクトルを形成する。

【0022】

以上、動きベクトルを効率的に読出および格納する改善されたビデオ動画システムおよび方法が提供されたことは明らかである。メモリに動きベクトルを格納する方法は、各々がビデオ信号の部分を表す第1マクロブロックに対応する4つの動きベクトルを受信する段階によって構成される。第1および第2動きベクトルは第1マクロブロックの上部分に対応し、第3および第4動きベクトルはマクロブロックの下部分に対応する。第1および第2動きベクトルはメモリの第1部分に格納され、第3および第4動きベクトルがメモリの第2部分に格納される。本発明は、動きベクトルのSRAMアレイ52, 52'を実現するために必要なメモリ量を大幅に軽減する。たとえば、1つの動きベクトルにつき2バイトがあるとすると、ビデオ・コーデックのデコーダ部分がOBMCおよび予測子関数を実行するには192バイトのRAM格納部が必要になる。192は次のように計算される：

$$[(24 \text{ マクロブロック}) (4 \text{ 動きベクトル / マクロブロック}) (2 \text{ バイト / 動きベクトル})] = 192$$

192バイトのRAMは必要とされる最小RAMサイズのように見えるが、本発明により大幅な節約がなされる。特に、本発明は図4に示されるマクロブロックC-2を超えてセーブする必要のないマクロブロックの上の動きベクトルをセーブまたは格納すること避ける。言い換えると、各マクロブロックC-3, C-4, C-5~C-23の2つの上ブロックがなくなる(格納されない)。この空間節約方法により、メモリ・サイズ要件は次のようになる:

[(下動きベクトル用の24マクロブロック) (2動きベクトル/マクロブロック)] = 48下動きベクトル

(上動きベクトル用の3マクロブロック) (2動きベクトル/マクロブロック) = 6上動きベクトル

総バイト = [(48 + 6) 動きベクトル] [動きベクトル毎2バイト] = 108バイト
 予測子およびOBMC関数はビデオ・コーデックのエンコーダおよびデコーダ側で実行しなければならないので、総合的な節約分は2(108)対2(192)となる。この例では、本発明を用いることにより、上記の以前の計算の384バイトに対し216バイトのサイズの節約(約44%の節約)となる。本発明をHDTV用MPEG-4と共に用いると、水平解像度がさらに増すために節約分ははるかに大きくなる。本件に開示される本発明のメモリ割付法を用いることにより、マクロブロックC-2後のブロック1, 2を無くすために複雑なアドレス指定法は必要ない。本発明により、H.263規格のOBMCおよび予測子関数が追加のアドレス指定用ハードウェアを追加せずに同じRAMを共有することができる。必要なハードウェアが軽減されるので、シリコン領域、コストおよび消費電力がすべて削減され、結果としてソフトウェアによる実行に匹敵し、さらにはそれよりも有利なビデオ・コーデックのハードウェアによる実行が可能になる。

【0023】

当業者には、本発明の範囲から逸脱せずに修正および変形が可能であることを理解頂けよう。たとえば、「メモリ」という用語が説明される部分で種々の電子的メモリを実現することができることが理解頂けよう。本発明の方法を実現するために種々の論理ゲート構造を利用することができる。従って、本発明は添付の請求項の範囲に入るこれらすべての変形および修正を包含するものである。

【図面の簡単な説明】

【図1】本発明で用いられるビデオ通信システムをブロック図に示す。

【図2】本発明によるビデオ・エンコーダをブロック図に示す。

【図3】本発明による図2の動きベクトル格納読出回路構成をブロック図に示す。

【図4】共通中間フォーマット(CIF: Common Intermediate Format)サイズ・フレーム内の1行のマクロブロック・レイアウトを等角図に示す。

【図5】H.263規格による予測子計算のための動きベクトル読出パターンを示す。

【図6】H.263規格によるOBMC関数のための動きベクトル読出パターンを示す。

【図7】図3のSRAMアレイに関するメモリの割付を示す。

【図8】本発明による図1のシステムで用いるビデオ・デコーダ・ブロック図である。

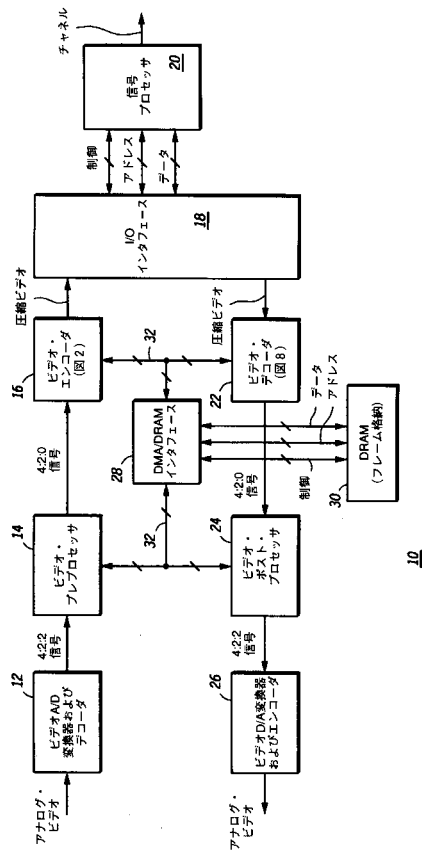
【図9】図8の解読機能用動きベクトル格納読出回路構成をブロック図に示す。

【符号の説明】

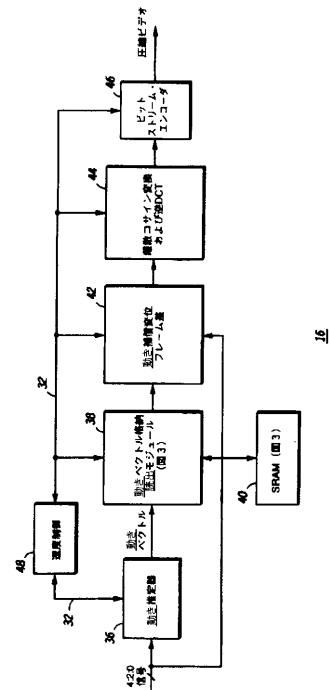
- 10 ビデオ・コーデック
- 12 ビデオA/D変換器およびデコーダ
- 14 ビデオ・プレプロセッサ
- 16 ビデオ・エンコーダ(図2)
- 18 I/Oインタフェース
- 20 信号プロセッサ
- 22 ビデオ・デコーダ(図8)
- 24 ビデオ・ポスト・プロセッサ
- 26 ビデオD/A変換器およびエンコーダ

28 DMA / DRAMインタフェース
30 DRAM (フレーム格納)

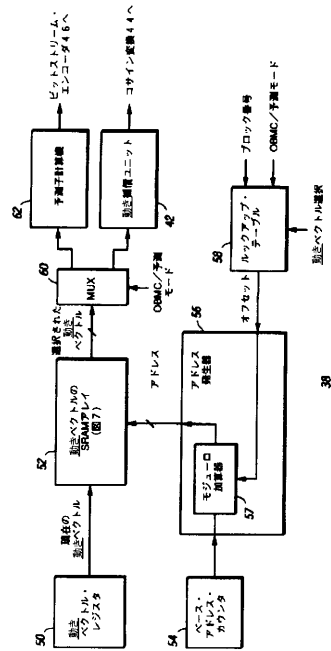
【図1】



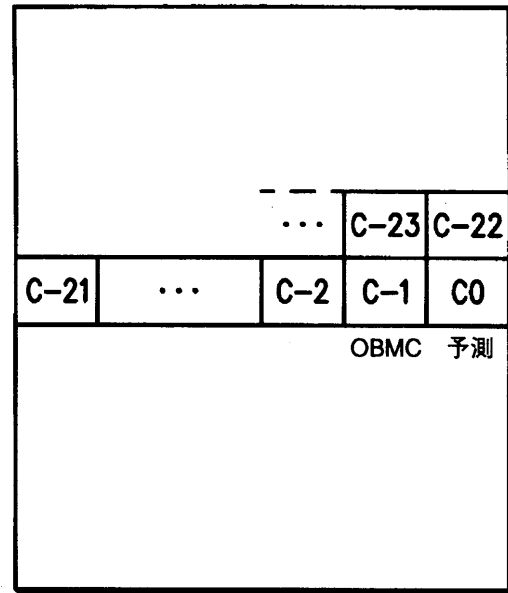
【図2】



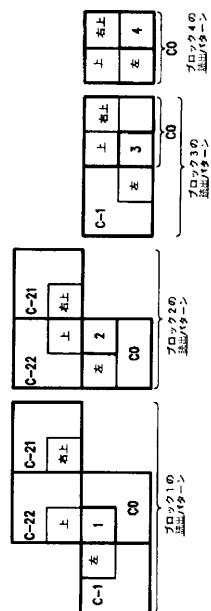
【 図 3 】



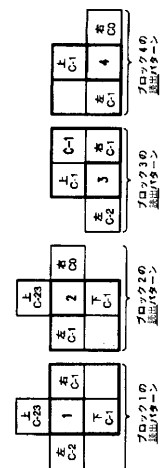
【 図 4 】



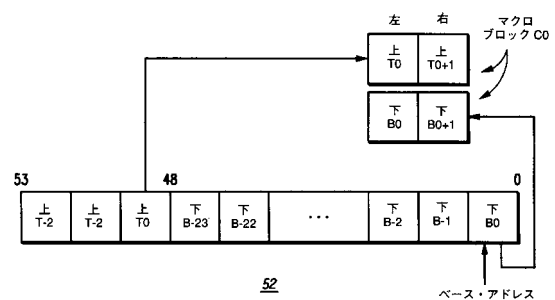
【圖 5】



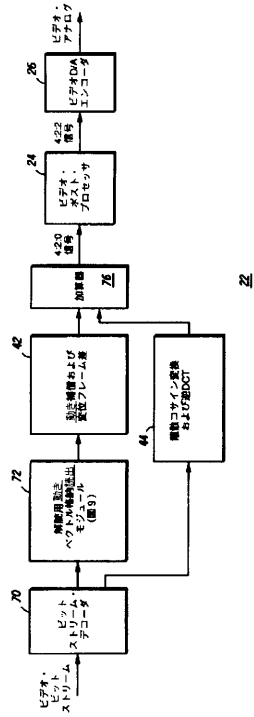
【 図 6 】



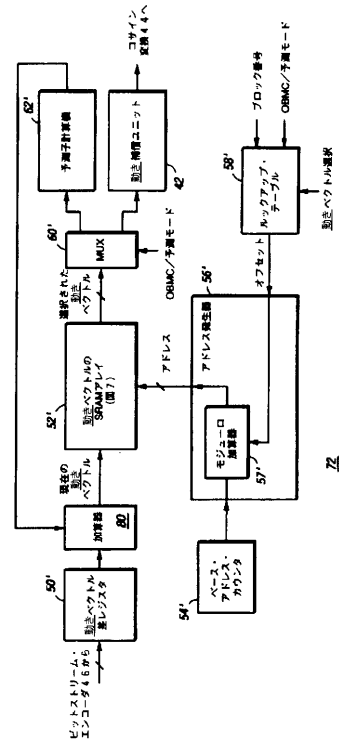
【圖 7】



【圖 8】



【 図 9 】



フロントページの続き

審査官 川崎 優

(56)参考文献 欧州特許出願公開第00838956(E P , A 1)

Nachtergaele, L., Catthoor, F., Kapoor, B., Janssens, S., Moolenaar, D. , Low-power data transfer and storage exploration for H.263 videodecoder system , IEEE Journal on Selected Areas in Communications , 1998年 1月 , Vol.16, No. 1 , P.120-129

(58)調査した分野(Int.Cl. , D B 名)

H04N 7/26-50