

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4583445号
(P4583445)

(45) 発行日 平成22年11月17日 (2010.11.17)

(24) 登録日 平成22年9月10日 (2010.9.10)

(51) Int. Cl. F I
H04N 7/32 (2006.01) H04N 7/137 Z

請求項の数 10 (全 41 頁)

(21) 出願番号	特願2007-513287 (P2007-513287)	(73) 特許権者	501261300
(86) (22) 出願日	平成17年5月10日 (2005.5.10)		エヌヴィディア コーポレイション
(65) 公表番号	特表2007-538431 (P2007-538431A)		アメリカ合衆国, カリフォルニア 950
(43) 公表日	平成19年12月27日 (2007.12.27)		50, サンタ クララ, サン トーマス
(86) 国際出願番号	PCT/US2005/016367		エクスプレスウェイ 2701
(87) 国際公開番号	W02005/110049	(74) 代理人	100094318
(87) 国際公開日	平成17年11月24日 (2005.11.24)		弁理士 山田 行一
審査請求日	平成19年1月10日 (2007.1.10)	(74) 代理人	100107456
(31) 優先権主張番号	60/570,087		弁理士 池田 成人
(32) 優先日	平成16年5月10日 (2004.5.10)	(74) 代理人	100155044
(33) 優先権主張国	米国 (US)		弁理士 田村 誠
(31) 優先権主張番号	11/125,852	(74) 代理人	100123995
(32) 優先日	平成17年5月9日 (2005.5.9)		弁理士 野田 雅一
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ビデオデータのプロセッサ

(57) 【特許請求の範囲】

【請求項 1】

一連のフレームにおいて動き推定を実行するように構成されており、動き推定の実行が、現在のフレームに対応するペルと呼ばれる画素ブロックを現在のフレームの前のフレームに対応する画素ブロックまたは現在のフレームの次のフレームに対応する画素ブロックと比較して変形が最小である画素ブロックを特定し、前記比較は、SAD演算と呼ばれる差の絶対値の合計の複数の演算を含む、参照画素のメモリ、現在の画素のメモリ、マルチプレクサ、動き推定プロセッサ、補間プロセッサ、データ書式設定器の少なくとも1つを再構成可能なデジタルビデオプロセッサと、

前記再構成可能なデジタルプロセッサに命令を与えて、第1のセットの属性に対応する入力ビデオデータを受け取るように、また第2のセットの属性に対応する出力ビデオデータを生成するように、前記再構成可能なデジタルプロセッサを動的に構成するように構成されており、前記命令は、入力ビデオデータに対応する属性と出力ビデオデータに対応する属性に関する情報を含む、制御プロセッサと、
を備え、

前記デジタルビデオプロセッサは、アイドル、セットアップ、実行、テスト、待機、続行を含む複数の状態のうちの1つをとる有限状態機械として構成されたものであり、

前記デジタルビデオプロセッサは、制御プロセッサからの前記命令を受け取った場合に、デジタルビデオプロセッサ内のメモリに、制御プロセッサからの前記命令に対応するタスクパラメータリストおよび命令シーケンスであって、デジタルビデオプロセッサの状態

10

20

を前記複数の状態から選択して切り替えるためのタスクパラメータリストおよび命令シーケンスを格納する、装置。

【請求項 2】

各クロックサイクルの間に複数の S A D 演算を実行可能とするために、各フレームに対応する画素ブロックを格納する複数のメモリを更に含む、請求項 1 に記載の装置。

【請求項 3】

前記複数のメモリが、各クロックサイクルの間に 1 6 回の S A D 演算を実行可能とするために、8 個のノードメモリを含む、請求項 2 に記載の装置。

【請求項 4】

前記 8 個のノードメモリが、重複したデータの入力および結果の出力のためのダブルバッファリングを可能とするように構成されている、請求項 3 に記載の装置。

10

【請求項 5】

前記再構成可能なデジタルビデオプロセッサは、画素位置に応じて一度ずつ、クロック周期ごとに S A D 演算を実行して、各画素ブロックを評価するように更に構成されており、評価のそれぞれは、第 1 の画素ブロックと第 2 の画素ブロック（第 1 の画素ブロックを一部とするより大きな参照ブロックに含まれる第 2 の画素ブロック）との間の S A D 指標値の計算と、最良の変位および当該最良の変位に対応する第 2 の画素ブロックの原点の保存とを含む、請求項 4 に記載の装置。

【請求項 6】

前記再構成可能なデジタルビデオプロセッサは、最良の変形の指標値に対応する第 2 の画素ブロックを取り囲むペル配列の双一次補間を実行するように更に構成されている、請求項 5 に記載の装置。

20

【請求項 7】

前記再構成可能なデジタルビデオプロセッサは、前記双一次補間を実行した後に、複数のハーフペルポジションのそれぞれの S A D 指標値を計算するように更に構成されている、請求項 6 に記載の装置。

【請求項 8】

前記属性は形式および解像度を含み、前記命令は、入力ビデオデータの第 1 の形式および第 1 の解像度に関する情報、および出力ビデオデータの第 2 の形式および第 2 の解像度の情報を含む、請求項 1 に記載の装置。

30

【請求項 9】

前記属性は、規格・標準の情報、符号化の情報、または圧縮の情報を更に含む、請求項 8 に記載の装置。

【請求項 10】

前記制御プロセッサが、ビデオデータ内の領域のセット内の領域の特定の原点と、前記領域のセット内の領域の相対的な変位とを用いて動き推定を実行するように、前記再構成可能なデジタルプロセッサを動的に構成するように更に構成された、請求項 1 に記載の装置。

【発明の詳細な説明】

【関連出願の参照】

40

【0001】

[01] 本出願は、米国仮特許出願第 60 / 570, 087 号明細書「適応コンピューティングマシンのデジタルビデオノード (Digital Video Node for An Adaptive Computing Machine)」(Master ら)、米国特許出願第 (未記載) 号明細書「ビデオデータのプロセッサ (Processor for Video Data)」(Master ら、出願日: 2005 年 5 月 9 日)、米国特許出願公開第 2003 / 0115553 号明細書「制御コードの実行に有限状態機械を選択的に使用するコンピュータプロセッサアーキテクチャ (Computer Processor Architecture Selectively Using Finite-State-Machine for Control Code Ex

50

ecution)」(Masterら)、米国特許出願第2004/0093601号明細書「適応集積回路及び対応する制御情報又は構成情報を開発及びスケジューリングする方法、システム、及びプログラム(Method, System and Program for Developing And Scheduling Adaptive Integrated Circuitry and Corresponding Control or Configuration Information)」(Masterら)、米国特許出願公開第2004/0177225号明細書「外部メモリコントローラノード(External Memory Controller Node)」(Masterら)、米国特許出願公開第2004/0181614号「適応コンピューティング環境における入出力コントローラノード(Input/Output Controller Node in an Adaptable Computing Environment)」(Masterら)からの優先権を請求するものである。

10

【0002】

[02]これらの出願と、米国特許第6,836,839号明細書「アプリケーションに固有の固定的なコンピューティング要素を有する、異種の再構成可能な様々な適応計算ユニットのマトリクスを備えた適応集積回路(Adaptive Integrated Circuitry with Heterogeneous and Reconfigurable Matrices of Diverse and Adaptive Computational Units Having Fixed, Application Specific Computational Elements)」(Masterら)は、あらゆる目的に、本文書に参考とすることにより組み込まれる。

20

【発明の背景】

【0003】

[03]今日の世界では、ビデオ(Video:動画像)情報はあらゆる場所に存在する。子供は、映像用として作成されたテレビ番組や教材から学習する。大人は、娯楽のため、或いは現在の出来事から遅れないように映像を使用する。DVD(Digital Versatile Disc)、デジタルケーブルテレビ、及び衛星テレビでは、映像情報を記録及び配信するのに、古いアナログ機構とは対照的にデジタルビデオデータを使用する。デジタルビデオデータは、今日のホーム環境及びオフィス環境においてますます普及しつつある。

30

【0004】

[04]デジタルビデオデータの処理に関与する数値計算量には、膨大な計算パワーが要求される。一般的な画質の映像1秒分のデジタルビデオデータを生成するためには、数千万~10億回の算術計算を実行する必要がある。

【0005】

[05]ビデオの計算は、ハードウェアを使用することにより、デジタルビデオデータ用のソフトウェアの符号器、復号器、トランスコーダと比較して高速化することができる。しかしながら、ハードウェアの一般的な設計方法は、1つの特定の解像度における1つの特定の形式のビデオデータに対して機能するのみである。従って、様々な解像度、標準・規格、形式のビデオデータに対して機能するハードウェアのニーズが存在している。

40

【発明の概要】

【0006】

[06]本発明の様々な実施形態は、ビデオデータを処理するシステム及び方法を提供する。好ましい実施形態においては、ビデオプロセッサは、様々な属性を有するビデオデータに対して動作するように実行時に構成される。これらのストリームは、例えば、形式、規格・標準、解像度、符号化、圧縮などの属性の様々な組合せを有することができる。本ビデオプロセッサは、動的な制御機構、例えば、プログラム、或いはレジスタに保持されている動的に構成可能な値(ただしこれらに限定されない)に従って、様々なビデオデータストリームに対して動作する。

【0007】

50

[07]本発明のいくつかの実施形態は、命令シーケンスを通じて動的に構成することのできるビデオプロセッサであって、命令が、現在のビデオデータの属性に関する情報を含んでいる、ビデオプロセッサを提供する。ビデオプロセッサは、指定されている属性を有するビデオデータを受け取るように、又は、指定されている属性を有するビデオデータを生成するように、又は、その両方を行うように、この情報によって構成される。

【0008】

[08]ビデオプロセッサの動作は、少なくとも1つの命令シーケンスによって制御される。複数のシーケンスを同時に使用することができ、それによって、プロセッサは、属性の異なるビデオストリームを同時に生成する、若しくは同時に受け取る、又はその両方を行うことができる。

10

【0009】

[09]本発明のいくつかの実施形態においては、1つ以上の制御プロセッサが、命令シーケンスをビデオプロセッサに提供する。制御プロセッサ及びビデオプロセッサは、コプロセッサとして協働するようにされている。これに代えて、又はこれに加えて、命令シーケンスを、ビデオプロセッサに関連付けられている1つ以上の命令メモリ又はキューに格納することができる。

【0010】

[10]本発明のいくつかの実施形態は、制御機構、例えば、命令、レジスタ値、制御信号、プログラマブルリンク（ただしこれらに限定されない）を通じて動的に構成することのできる加算器の配列を提供する。プロセッサによって加算器の配列にビデオデータが送られると、配列は、ビデオデータを処理するのに必要な数値計算、論理計算、又は逐次計算の結果を生成する。

20

【0011】

[11]本発明のいくつかの実施形態においては、加算器の配列を使用して様々な機能を計算することによって、誤りベクトルを生成する。誤りベクトルは、時間的に関連している一連のビデオデータの間に動きを検出するために使用することができる。更に、誤りベクトルは、出力されるビデオに符号化される未処理データの計算にも使用することができる。

【0012】

[12]本発明のいくつかの実施形態においては、加算器の配列は、ビデオデータに適用されるフィルタリング機能の計算にも使用されている。そのようなフィルタリング機能としては、動き検出前の入力ビデオの補間又は間引き（*d e c i m a t i o n*）が挙げられる（ただしこれらに限定されない）。

30

【0013】

[13]間引き又は補間されたペルを使用して、入力ビデオとは異なる属性を有する出力ビデオを生成することができる。これに代えて、又はこれに加えて、符号化プロセス又は圧縮プロセスの間のみ存続する間引きされたペル、又は補間されたペルを使用して、出力ビデオの精度或いは視聴上の品質を高めることができる。そのような用途としては、最初のトップレベルの動きスキャン（*t o p - l e v e l m o t i o n s c a n*）時に入力ビデオの解像度を下げることに基づいて動き検出の性能を高める階層的な手法や、動き検出の精度を高める補間手法が挙げられる（ただしこれらに限定されない）。

40

【0014】

[14]本発明のいくつかの実施形態は、動的に構成可能なハードウェア探索動作を提供する。そのような動作は、ビデオを符号化するときに、様々な目的で、例えば、時間的に関連している一連のビデオデータの間に動きを検出する目的で（ただしこれに限定されない）、使用することができる。1回のハードウェア動作においては、1セットのビデオデータの中の参照ブロックを、同じセット又は異なるセットのビデオデータの中の一連の領域と比較する。ビデオデータの属性を動的に構成できることに加えて、セット内の領域の数と、複数の異なる領域の間の相対的なオフセットとを動的に構成することができる。

【0015】

50

[15]本発明のいくつかの実施形態は、適応コンピューティングマシン（ACM）を使用して実施されている。ACMは、それぞれが相互接続ネットワークに結合されている複数の異種の計算要素を含んでいる。

【0016】

[16]本明細書の説明を添付の図面を参照しながら読むことによって、本発明の様々な実施形態の目的、特徴、及び利点が明らかになるであろう。図面には、本発明に関連する技術、本発明の例としての実施形態、本発明の例としての使用法を図解してある。

【詳細な説明】

【0017】

[60]本明細書における説明及び図は、本発明に関連する技術を解説しており、本発明の例及び本発明の使用例を示している。公知の方法、手順、システム、回路、要素については、本発明の原理があいまいになることを避けるため、概略的に図解及び説明してある。これに対して、本発明の具体的な実施形態については、細部が本発明の他の実施形態とは異なる場合にも、詳細に説明してある。

【0018】

[61]本明細書の説明においては、抽象的な用語又は一般的な用語、例えば、受け取る、提示する、促進する、生成する、はい、いいえ（ただしこれらに限定されない）を使用している。そのような用語は、当業者には、コンピュータやデジタルデバイス、電気機械システムにおけるコンポーネント、データ、或いは動作を表す便利な術語として使用されている。そのようなコンポーネント、データ、及び動作は、実際の物体の物理特性、例えば、電子電圧（electronic voltage）、磁界、光の反射率（ただしこれらに限定されない）に具体化されている。同様に、知覚的或いは感覚的な用語、例えば、比較する、決定する、計算する、制御する（ただしこれらに限定されない）を使用して、そのようなコンポーネントやデータ、動作、或いはそのような物理的操作を言い表すこともできる。

【0019】

[62]図1は、本発明の実施形態によるビデオプロセッサの高レベル機能のブロック図を示している。ビデオプロセッサ100は、ビデオデータへのネットワークアクセスを維持又は提供する1つ以上のビデオデータデバイス110と、1つ以上のビデオ計算エンジン120と、現在のステータス情報を保持する1つ以上のデバイス130と、1つ以上の制御モジュール140と、制御・構成データへのネットワークアクセスを維持又は提供する1つ以上の制御・構成データデバイス150とを含んでいる。

【0020】

[63]ビデオデータ110は、入力ビデオデータ112と、出力ビデオデータ114と、中間結果116とを含んでいる（ただしこれらに限定する必要はない）。本発明の様々な実施形態においては、ビデオデータ110は、ビデオプロセッサ100の他のデバイスと同じ集積回路内のメモリモジュール、その集積回路の外部のメモリデバイス、レジスタファイル、又はデータを保持するその他の回路、のうちの1つ以上に保持されている。

【0021】

[64]ビデオ計算エンジン120は、入力マルチプレクサ122と、フィルタ機能回路124と、動き探索回路126と、誤差、例えば差の絶対値の合計（SAD：sum of absolute differences）計算回路127と、出力書式設定回路128とを含んでいる（ただしこれらに限定する必要はない）。動き推定エンジン120は、入力ビデオデータ112及び中間結果116を受け取り、中間結果116及び出力ビデオ114を生成する。

【0022】

[65]結果116、114を生成する目的でビデオ計算エンジン120が受け取ったデータに適用される処理機能及び演算は、制御信号146によって制御される。エンジン120の様々な実施形態は、信号146の制御下で、様々な機能、例えば、符号化、復号化、トランスコーディング、動き推定を実行する（ただしこれらに限定されない）。

【0023】

[66]ビデオ入力112、ビデオ出力114、及び中間結果116の属性は、制御信号146によって指定される。これらの属性としては、形式、規格・標準、解像度、符号化、圧縮度が挙げられる（ただしこれらに限定する必要はない）。

【0024】

[67]制御信号146は、制御・構成データ150と現在のステータスデータ130とに基づいて制御モジュール140によって生成される。制御モジュール140は、ハードウェアタスクマネージャ142と、有限状態機械144と、コマンド復号器146とを含んでいる（ただしこれらに限定する必要はない）。

【0025】

[68]現在のステータス130としては、ビデオデータ110の中の様々なデータ項目が処理可能な状態にあることを示すステータスと、制御・構成データ150の中の様々なデータ項目が、ビデオデータ110に適用される処理動作を制御できる状態にあることを示すステータスが挙げられる（ただしこれらに限定する必要はない）。様々な処理はビデオデータ及び制御・構成データが適用可能な状態にあるときに行われるため、現在のステータス130によって計算のデータフローモデルが導入される。

【0026】

[69]制御・構成データ150としては、コマンドシーケンス152と、タスクパラメタリスト154と、アドレス及びポインタ156とが挙げられる（ただしこれらに限定する必要はない）。アドレス及びポインタ156は、ビデオデータ110又は制御・構成データ150、或いはその両方のデータの中の様々なデータ項目を指す。本発明の様々な実施形態においては、ビデオデータ110は、ビデオプロセッサ100の他のデバイスと同じ集積回路内のメモリモジュール、その集積回路の外部のメモリデバイス、レジスタファイル、又はデータを保持するその他の回路或いはデバイス、のうちの1つ以上に保持されている。

【0027】

[70]ビデオデータ110の様々な構成要素は、様々な属性、例えば、形式、規格・標準、解像度を有する（ただしこれらに限定する必要はない）。ビデオデータ110の中の構成要素の形式は、MPEG-2、MPEG-4の1つのバージョン、Windows Media Video (WMV)の1つのバージョン、ITU-T（国際電気通信連合）規格X.264の1つのバージョンのうちの少なくとも1つを含んでいることができる。X.264は、この規格の開発を先導しているグループの後、JVT (Joint Video Team: ジョイントビデオチーム)としても知られている。また、X.264は、ISO/IEC（国際標準化機構/国際電気標準会議）規格であるMPEG-4 (Motion Picture Experts Group Layer 4) Part 10としても知られている。ビデオデータ110の中の構成要素は、約1/4 CEL（例えば携帯電話によって撮影される静止画像に使用される）と、高精細テレビ (HDTV)の1つのバージョンによって定義される解像度との間の任意の解像度を有することができる。

【0028】

[71]適応コンピューティングマシン (ACM) は、同種のネットワークによって接続されている異種のノードを含んでいる。そのような異種のノードとしては、ドメインビデオノード (DVN) が挙げられる（ただしこれに限定されない）。そのような同種のネットワークとしては、マトリクス相互接続ネットワーク (MIN: matrix interconnect network) が挙げられる（ただしこれに限定されない）。本発明のいくつかの実施形態のいくつかの側面、例えばMIN、或いは適応コンピューティングマシン (ACM) の他の側面については、関連出願として先に引用した特許明細書及び特許出願明細書に更に説明してある。

【0029】

[72]ドメインビデオノード (DVN) は、ACMにおける1つのノード型として実施す

10

20

30

40

50

ることができ、又は、任意の適切なハードウェア設計手法若しくはソフトウェア設計手法、又はその両方において実施することができる。例えば、本発明の実施形態の機能は、個別の回路として、或いは集積回路（例：カスタム、セミカスタム、その他）、プログラマブルなゲートアレイ、ASIC（特定用途向け集積回路）、又はその他として設計することができる。機能は、必要に応じて、ハードウェアによって、若しくはソフトウェアによって、又はその両方の組合せによって実行することができる。本明細書に説明した様々な機能は、都合よく単独で使用する、又は別の機能（例：先行技術において公知の機能、今後開発される機能）との組合せ或いはサブコンビネーションにおいて使用することができる。DVNは、ASICに匹敵するパワーエリア性能指標値を達成する再構成可能なデジタルビデオプロセッサである。

10

【0030】

[73] 1つの実施形態においては、DVNはACMに含まれており、再構成可能であるという付加的な利点を有し、これによってDVNは、一般に使用されている動き推定アルゴリズムを複数実行することができる（同じ機能を達成するためには場合によっては複数のASICが必要となる）。しかしながら、別の実施形態においては、DVN、又はDVNの一部或いは機能を、任意の適切な方法（例：構成可能ではない設計、部分的に構成可能な設計）において実施することができる。

【0031】

[74] ビデオの処理には、特にフレーム解像度が高い場合には相当な計算パワーが要求される動き推定ステップが含まれる。ドメインビデオノード(DVN)は、ACMのノード型のうち、動き推定機能及びその他のビデオ処理機能の実行を提供するノードである。

20

【0032】

[75] 図2は、DVNの簡略化したブロック図を示している。動き推定は、現在のフレームのピクセル（すなわちペル）のブロックを、前のフレーム又は次のフレーム、又はその両方のブロックと比較して、変形(d i s t o r t i o n)が最小である1つ以上のブロックを特定するステップから成る。そのような比較は、例えば、選択基準として一般に使用されている「差の絶対値の合計」(SAD)指標値(m e t r i c)を含んでいることができる。

【0033】

[76] DVNは、クロックサイクルごとに16回のSAD演算を可能とする8個の2KBのノードメモリを含んでいる。更に、これにより、各次元における±16ペルの探索領域について、 16×16 の現在のブロック（マクロブロック）と 48×48 の参照ブロックに対して、重複したデータの入力及び計算と結果の出力とを行うためのダブルバッファリングも可能である。

30

【0034】

[77] DVNの動作には、1) フルペル(f u l l P E L)探索、2) 18×18 フルペルから 33×33 ハーフペル(h a l f P E L)への双一次補間、3) ハーフペル探索、4) 最良の指標値及びそれに対応する 16×16 マクロブロックの原点を指定されているノードに出力し、 16×16 マクロブロック自体を復号器の動き補償エンジンに出力する、が含まれている。

40

【0035】

[78] フルペル探索：DVNは、フルペル探索の2つの動作モードをサポートする。一方のモードでは、 16×16 ポジションのそれぞれを一度に評価し、クロック周期ごとに16回のSAD演算を実行する。第2のモードでは、隣り合う40個のポジションを同時に評価し、クロック周期ごとに160回のSAD演算を実行する。いずれのモードの場合も、評価は、現在の 16×16 マクロブロックと、 48×48 の参照ブロックの1089の 16×16 マクロブロックのうちの1つとの間の「差の絶対値の合計」(SAD)指標値を計算し、最良の指標値及びそれに対応する 16×16 マクロブロックの原点とをつねに保存するステップから成る。

【0036】

50

[79]いずれのモードにおいても、探索パターンは、複数のハードワイヤードパターン (hardwired pattern)、又は、ノードによって生成される x, y 座標 (原点) のリスト (DVN内に配置されているキューに格納されている) によって指定される複数のポジションとすることができる。様々なキューデータ構造を使用することができ、様々な早期完了方式をサポートすることができる。

【0037】

[80] 48×48 の参照ブロックの中には、1089個の候補探索ポジションが存在する。図3には、相対座標 $H(0)$, $V(0)$ を有する「中央」ポジションを示してある。図4には、5つの探索ポジション、すなわち、中央ポジション ($H(0)$, $V(0)$)、左上ポジション ($H(-16)$, $V(-16)$)、右上ポジション ($H(+16)$, $V(-16)$)、左下ポジション ($H(-16)$, $V(+16)$)、右下ポジション ($H(+16)$, $V(+16)$) を強調して示してある。

10

【0038】

[81] 図5は、40個のポジション、すなわち 48×48 の参照ブロックの中央における 8×5 のポジションを同時に探索する例を示している。

【0039】

[82] 図6は、参照ブロックの中央における9セットの40個のポジションを探索する例を示している。水平探索範囲は $-12 \sim +11$ であり、垂直探索範囲は $-7 \sim +7$ であり、探索されるポジションの合計数は360である。

【0040】

20

[83] ハーフペル双一次補間：変形指標値が最良である 16×16 マクロブロックを求めるフルペル探索の後、周囲の 18×18 ペルの配列に対して双一次補間を実行し、図7に示したように 33×33 ハーフペル配列を得る。図8は、ハーフペル双一次補間において使用する計算についてまとめている。

【0041】

[84] ハーフペル探索：双一次補間を実行した後、DVNは、9個の候補ハーフペルポジションのそれぞれのSAD指標値を計算する。図9は、ハーフペルデータの番号付け規則を示している。この図では、フルペルポジション (ポジション「5」) の指標値はすでに計算されている。従って、ハーフペル探索では、それ以外の8個のハーフペルポジションのそれぞれのSAD指標値を生成し、9個の候補のうち指標値が最良である候補を求める。

30

【0042】

[85] 結果の出力： 33×33 ハーフペル配列の中の最良の指標値及びそれに対応する 16×16 マクロブロックの原点とを、指定されているノードに出力する。 16×16 マクロブロック自体は、復号器の動き補償エンジンに出力する。

【0043】

[86] DVNの動作の要約：DVNは、QSTノードラッパー (QST Node Wrapper) と、8個の2KBのノードメモリ (それぞれ32ビットの512ワードとして編成されている) と、動き推定エンジン (ME) とを含んでいる。動き推定エンジンは、上に概説したフルペル探索と、双一次補間と、ハーフペル探索とを実行する。動き推定エンジンは、最良の指標値及びそれに対応する 16×16 マクロブロックの原点を、指定されているノードに出力し、 16×16 マクロブロック自体を復号器の動き補償エンジンに出力する。

40

【0044】

[87] 8個のノードメモリのそれぞれは、完全に重複したデータ入力及び計算と結果出力とが可能であるように、ピンポンバッファ対 (ping pong buffer pair) に分割されている。指定されているメモリから8個のDVNメモリに1つのデータセットが転送された後、DVNの動き推定エンジンは、その計算を実行して結果を出力する。この間に、次のデータセットが別のシステムメモリから8個のDVNメモリに転送され、以下同様である。

50

【0045】

[88]各データセットは、 48×48 ペルの配列（参照ブロック）と、 16×16 ペルの配列（現在のブロック）とから成る。これらのデータは、動き推定エンジンからデータへのアクセスが最適化されるような方式で8個のノードメモリの間で分散されている。動き推定エンジンからデータへのアクセスには、一般的には、クロック周期ごとに8個のノードメモリのそれぞれからの4バイトデータへのアクセスが含まれる。これにより、一度に1セットの 16×16 探索ポジションを評価する「高速探索」モード時に、クロック周期ごとに16回のSAD演算を実行することができる。更に、「40ポジション同時探索」モードの場合、クロック周期ごとに、参照ブロックの4行のデータと、現在のブロックの4行のデータとにアクセスすることができる。

10

【0046】

[89]ノードメモリアクセスの要約：各データセットは、指定されているノードによって制御される適応コンピューティングマシン（ACM）アーキテクチャのマトリクス相互接続ネットワーク（MIN）を介して、指定されているメモリから8個のDVNノードメモリに転送される。 48×48 ペル参照ブロックの場合、行0, 4, 8, . . . , 44はノードメモリ1の中の連続する位置に書き込まれ、行1, 5, 9, . . . , 45はノードメモリ2の中の連続する位置に書き込まれ、行2, 6, 10, . . . , 46はノードメモリ3の中の連続する位置に書き込まれ、行3, 7, 11, . . . , 47はノードメモリ4の中の連続する位置に書き込まれる。 16×16 ペルの現在のブロックの場合、行0, 4, 8, 12はノードメモリ1及び5に書き込まれ、行1, 5, 9, 13はノードメモリ2及び6に書き込まれ、行2, 6, 10, 14はノードメモリ3及び7に書き込まれ、行3, 7, 11, 15はノードメモリ4及び8に書き込まれる。

20

【0047】

[90]フルペル探索時には、動き推定エンジンは、メモリ1～4から参照ブロックのデータを読み取り、メモリ5～8から現在のブロックのデータを読み取る。双一次補間時には、動き推定エンジンは、メモリ1～4から参照ブロックのフルペルデータを読み取り、ハーフペル補間した参照ブロックデータを、メモリ5～8に書き込む。ハーフペル探索時には、動き推定エンジンは、ハーフペル補間した参照ブロックデータをメモリ5～8から読み取り、メモリ1～4から現在のブロックのデータを読み取る。最後に、 16×16 マクロブロックを復号器の動き補償エンジンに出力するため、動き推定エンジンは、メモリ5～8から読み取ってMINに書き込む。

30

【0048】

[91]ノードメモリの編成（ノードメモリへのデータ入力時）：ノードメモリ1～4においては、参照ブロックのデータは位置 $0 \times 00 \sim 0 \times 8F$ に順次格納され、現在のブロックのデータは位置 $0 \times C0 \sim 0 \times CF$ に順次格納される。ノードメモリ5～8においては、現在のブロックのデータが位置 $0 \times C0 \sim 0 \times CF$ に順次格納される。

【0049】

[92]ノードメモリの編成（双一次補間時）：ノードメモリ5～8においては、候補ポジション1（7）のハーフペルデータが位置 $0 \times 00 \sim 0 \times 10$ に順次格納される。候補ポジション2（8）のハーフペルデータは位置 $0 \times 20 \sim 0 \times 30$ に順次格納される。候補ポジション3（9）のハーフペルデータは位置 $0 \times 40 \sim 0 \times 50$ に順次格納される。候補ポジション4のハーフペルデータは位置 $0 \times 80 \sim 0 \times 8F$ に順次格納される。候補ポジション5のハーフペルデータは位置 $0 \times 90 \sim 0 \times 9F$ に順次格納される。候補ポジション6のハーフペルデータは位置 $0 \times A0 \sim 0 \times AF$ に順次格納される。

40

【0050】

[93]DVNプログラミングモデル：DVNの構成情報及び制御情報のすべては、DVNのタスクパラメータリスト（TPL）に格納されている。条件付き動作では、探索キューに対して、ラッパーの64個のカウンタのうちの適切な数のカウンタと、適切なデータ構造とを使用する。フルペル探索モードの両方において、適切な数及び特性の、ハードワイヤード探索パターンを使用することができる。

50

【0051】

[94] 将来的なDVNの能力としては、例えば、更に高速なフルペル探索のための更に高いSAD強度（SAD strength）、クォーターペル（quarter PEL）処理、双三次補間が挙げられる（ただしこれらに限定されない）。

【0052】

[95] 適応コンピューティングマシン（ACM）は、同種のネットワークによって接続されている異種のノードを含んでいる。各ノードは、図12に示したように3つの要素、すなわちラッパーと、メモリと、実行ユニットとを含んでいる。

【0053】

[96] ドメインビデオノード（DVN）は、ASIC（特定用途向け集積回路）に匹敵するパワーエリア性能指標値を達成する再構成可能なデジタルビデオプロセッサであり、再構成可能であるという付加的な利点によって、DVNは、一般に使用されている動き推定アルゴリズムを複数実行することができる（同じ機能を達成するためには場合によっては複数のASICが必要となる）。

【0054】

[97] 図13は、DVNプログラミング環境を描いたブロック図を示している。DVNの高レベル制御は、DVNノードのラッパー内のハードウェアタスクマネージャ（HTM）と、DVNの実行ユニット内の有限状態機械（FSM）とによって提供される。このインタフェースは、ラッパーから実行ユニット（EU）への、ACMに標準的なEU_RUN信号、EU_CONTINUE信号、及びEU_TEARDOWN信号と、実行ユニットからラッパーへのACK+テスト応答及びEU_DONE応答とを含んでいる。

【0055】

[98] 図14は、FSMの遷移表を示している。FSMは、6つの状態、すなわち、アイドル、セットアップ、実行、ACK+テスト、待機、続行のうちの1つをとる。ノードが有効になっていない、又は実行ユニット（EU）が有効になっていないときには、FSMはアイドル状態に入り、その状態を維持する。ノードラッパーが「eu_abort」信号をアサートしたときにも、FSMはアイドル状態に遷移する。更に、ノードラッパーが「eu_tear down」信号をアサートすると、FSMはそれに応答して待機状態を脱してアイドル状態に遷移する。

【0056】

[99] FSMの一般的な動作シーケンスには、以下の遷移が含まれる。

- ・ ノードラッパーが「eu_run」信号をアサートしたことに応答しての、アイドルからセットアップ：セットアップの間は、バイトコードによって実行ユニットの初期化が制御され、ノードメモリに配置されている「タスクパラメータリスト（TPL）」に格納されているパラメータが構成レジスタにロードされる。
- ・ セットアップのバイトコード「done」に応答しての、セットアップから実行：実行状態の間は、実行ユニットがコマンドキューからの命令を処理して、動き推定動作として、フルペル探索、フラクショナルペル補間、フラクショナルペル探索、マクロブロックの符号付き差、結果の出力を実行する。
- ・ コマンドキューからDONE命令が読み取られたときの、実行からACK+テスト：ACK+テストの間は、動き推定段階中に生成及び消費されたデータ量を示すACKがネットワークに送られる。一般には、最後のACKには、ノードラッパーのハードウェアタスクマネージャ（HTM）に「次の状態（what's next）」を尋ねる「テスト」指示が含まれる。
- ・ ACK+テストバイトコードに応答しての、ACK+テストから待機：ACK+テストから待機した後、FSMは、「次の状態」への応答をノードラッパーから受け取るまで待機状態のままである。
- ・ ノードラッパーが「eu_tear down」信号をアサートしたことに応答しての、待機からアイドル：FSMは「eu_done」をアサートする。
- ・ ノードラッパーが「eu_continue」信号をアサートしたことに応答しての

10

20

30

40

50

、待機から続行：続行の間は、バイトコードによる制御下で、ノードメモリに配置されているタスクパラメータリストからのパラメータが特定の実行ユニットレジスタに再ロードされる。

- ・ 続行のバイトコード「done」に応答しての、続行から実行

【0057】

[100]上記の動作シーケンスでは、ACK+テスト後に、FSMの制御がノードラッパーのハードウェアタスクマネージャに戻される。「ACK処理」後に制御が保持されるようにDVNをプログラムすることもできる。オプションとしては、(「eu_done」のアサート後に)条件付き又は無条件で続行又はアイドルに遷移することが挙げられる。テスト条件としては、ノードラッパーの64個のうちの任意のカウンタの符号(MSB：最上位ビット)、特定のタスクパラメータリストエントリの符号(MSB)、バイトコードによって制御される5ビット・ダウンカウンタの符号(MSB)が挙げられる。

【0058】

[101]タスクを実行するうえで必要なすべての情報は、ノードメモリに配置されている「タスクパラメータリスト」(TPL)に含まれている。セットアップの間、DVNのハードウェアは、タスクパラメータリストへのポインタをノードメモリから取得する。ノードメモリには、最大31個のタスクのポインタが、ノードラッパー信号{tp_t_br[8:0], active_task[4:0]}によって指定される位置において31個の連続するワードに格納される。

【0059】

[102]DVNの各タスクパラメータリストには、ノードメモリの64個のロングワードが割り当てられている。タスクパラメータリストの最初の8個のエントリは、特定の情報用に予約されている。残りの56個のロングワードが、セットアップ、ACK+テスト、及び続行のパラメータを格納するのに利用可能である。

【0060】

[103]タスクパラメータリストの位置0x00には、セットアップの2つのパラメータ、すなわち、セットアップのバイトコードの、ノードメモリにおける開始アドレスと、TPLポインタからパラメータまでのオフセットとが含まれている。TPL位置0x01には、ACK+テストの2つのパラメータ、すなわち、ACK+テストのバイトコードの、ノードメモリにおける開始アドレスと、TPLポインタからパラメータまでのオフセットとが含まれている。TPL位置0x02には、破棄(teardown)の1つのパラメータ、すなわち、破棄のバイトコードの、ノードメモリにおける開始アドレスが含まれている(現時点では「破棄する」ものはないため、このパラメータは将来的に必要なが生じた場合におけるプレースホルダである)。TPL位置0x03には、続行の3つのパラメータ、すなわち、続行のバイトコードの、ノードメモリにおける開始アドレスと、TPLポインタから続行の2セットのパラメータまでのオフセットとが含まれている。TPL位置0x04及び0x05は、バイトコードが操作することのできる一対のセマフォに使用され、位置0x06及び0x07は予約されている。TPL位置0x08~0x3Fは、他のすべてのタスクパラメータを格納するのに利用可能である。

【0061】

[104]図15は、DVNのタスクパラメータリストのレイアウトを示している。タスクパラメータリストには、3つの状態、すなわち、セットアップ、ACK+テスト、及び続行のそれぞれのパラメータセットが格納される。続行に対しては、それぞれパラメータ数は同じであるがTPLポインタオフセットが異なる2つのパラメータセットが存在することができる。これにより、2セットのグローバルパラメータ(例：1つ以上の消費側ノード宛先、フラクショナルペル補間フィルタ)の間で条件付き選択を行うことができる。特定の状態のパラメータセットは、タスクパラメータリストにおける連続する位置に格納されなければならない。バイトコードのLOAD命令が実行されると、DVNのハードウェアは、セットにおける次のパラメータを選択する。その状態のTPLポインタから第1のパラメータの位置までのオフセットは、0x08~0x3Fの範囲内でなければならない

。

【0062】

[105] DVNのTPLメモリ及びレジスタのフォーマット：図16は、バイトコードの開始アドレスとTPLポインタオフセットとが含まれているTPL位置0x00～0x03のフォーマットを示している。図17は、TPL位置0x04及び0x05における2つのセマフォのフォーマットを示している。図18は、TPL位置（TPLポインタとACK+テストオフセットの合計）から始まるACK+テストパラメータのフォーマットを示している。図19は、セットアップ及び続行時にタスクパラメータリストパラメータがロードされるレジスタのフォーマットを示している。

【0063】

[106]制御及びステータスレジスタ（CSR）：DVNのステータスレジスタには、使用する補間フィルタ、参照ブロックの端部における補間方法、64個のフラグのうち「コマンドキューリダイレクト」を示すフラグ、使用するピンポンバッファ対、参照ブロックの場合の1行あたりのロングワード数を指定するフィールドが含まれている。

【0064】

[107]表1及び表2は、ビット[6：5]及びビット[4：0]を説明している。

【表1】

ビット 43210	フィルタ
00001	双一次補間
00010	MPEG4
00100	H.264
01000	WMV
10000	予約

表1：フラクショナルペル補間フィルタの選択：ビット[4：0]

【表2】

ビット 65	方法
00	必要なペルすべてがDVN ノードメモリ内
01	ミラー
10	複製
11	予約

表2：参照ブロックの端部における補間方法：ビット[6：5]

【0065】

[108]ビット[13：8]は、コマンドキューリダイレクトフラグである。コマンドキューの処理中は、DVNは、64個のラッパーカウンタ符号ビットのうち、ビット[13：8]によって示される符号ビットを絶え間なく監視している。カウンタ符号ビットが1'b1にセットされると、DVNは、そのビットをクリアし、「切替」指示をコントローラに送り、コマンドキュー（A／B）からの処理を中断し、コマンドキュー（B／A）からの処理を再開する。

【0066】

[109]ビット[20：16]は、バッファセクタである。5個のビットのそれぞれは、一对のピンポンバッファの間で選択する。ビット16は、「参照ブロックのバッファの選択」であり、0は参照ブロックAを選択し、1は参照ブロックBを選択する。ビット1

10

20

30

40

50

7は、「現在のブロックのバッファの選択」であり、0は現在のブロックAを選択し、1は現在のブロックBを選択する。ビット18は、「コマンドキューのバッファの選択」であり、0はコマンドキューAを選択し、1はコマンドキューBを選択する。ビット19は、「水平コスト関数のバッファの選択」であり、0は水平コスト関数表Aを選択し、1は水平コスト関数表Bを選択する。ビット20は、「垂直コスト関数のバッファの選択」であり、0は垂直コスト関数表Aを選択し、1は垂直コスト関数表Bを選択する。

【0067】

[110]ビット[28:24]は、「参照ブロックの1行あたりのロングワード」である。参照ブロックの行を構成しているロングワード数は、行(k)から行(k+4)までの幅をサポートするのにノードメモリアドレスユニットによって必要とされる。

10

【0068】

[111]コントローラノードロケータレジスタ(CNLR):DVNは、PTPプロトコルを使用してコントローラと通信し、この通信にはポート番号(すなわちビット[5:0])及びノードID(すなわちビット[15:8])が必要である。

【0069】

[112]宛先ノードロケータレジスタ(DNLR):宛先(消費側)ノードへの出力動作が含まれているコマンドキュー命令を処理するとき、DVNは、1ビットの論理出力ポートフィールドを使用して2つの宛先ノードのうちの1つを選択する。このビットが1'b0にセットされていると、その出力はノードID(すなわちビット[15:8])、ポート番号(すなわちビット[5:0])に送られ、このビットが1'b1にセットされていると、その出力はノードID(すなわちビット[31:24])、ポート番号(すなわちビット[21:16])に送られる。

20

【0070】

[113]DVNのバイトコード:DVNのオーバーヘッドプロセッサ(OHP)は、2つの個別のバイトコードセット、すなわち、セットアップ及び続行のセットと、ACK+テストのセットとを解釈する。バイトコードインタープリタがどちらのセットを使用するかは、FSM状態によって示される。セットアップ及び続行に対しては、図20に示したように1つのバイトコードLOADのみが存在する。ACK+テストに対しては、より包括的なバイトコードセットが存在する。これらは図21A～図40Eにまとめてある。ACK+テストのバイトコード0x80～0xFFは予約されており、NOP、すなわち「無演算(no operation)」として扱うことができる。

30

【0071】

[114]DVNの場合、プログラミングステップの1つにおいて、バイトコードを書き込んでタスクパラメータリストを生成することが要求される。バイトコードは、必ず「インラインで」実行され、また、ロングワード境界から始まらなければならない。バイトコードの実行は、「リトルエンディアン」である、すなわち、最初に実行されるバイトコードが第1のロングワードのバイト[7:0]であり、次いでバイト[15:8]、次いでバイト[23:16]、次いでバイト[31:24]、次いで第2のロングワードのバイト[7:0]であり、以下同様である。

【0072】

40

[115]タスクパラメータリスト及びバイトコードのプログラミングの例:図22は、「一般的な」タスクのタスクパラメータリスト及びバイトコードを示している。タスクパラメータリストは、21個のロングワードから成り、バイトコードのロングワードは5個存在する。セットアップでは、タスクパラメータリストからレジスタをロードするのに3クロックサイクルが必要であり、3サイクルの最後にDONE(完了)が示される。ACK+テストでは、少なくとも10クロックサイクルが必要であり、そのうち2クロックサイクルで提供側のACK、2クロックサイクルで消費側のACK、2クロックサイクル(又はそれ以上)で入力バッファの状態をテスト(及び必要であれば待機)し、2クロックサイクル(又はそれ以上)で出力バッファの状態をテスト(及び必要であれば待機)し、2クロックサイクルで、2セットの構成レジスタのうちの1セットを選択するためのカウン

50

タ符号をテストする。続行では、構成レジスタ、すなわちステータスレジスタ、コントローラノードロケータレジスタ、及び宛先ノードロケータレジスタを更新するのに3クロックサイクルが必要である。

【0073】

[116]コマンドキューの概要：DVNのプログラミングモデルには、DVNノードメモリ内のコマンドキューに格納される命令シーケンスを作成するステップが含まれる。これらの命令は、動き推定動作、すなわち、フルベル探索、フラクショナルベル補間、フラクショナルベル探索、（おそらくはマクロブロックの符号付き差）、及び結果の出力を制御する。

【0074】

[117]図23は、DVNのメモリマップを示している。各マクロブロックを処理するため、DVNは、タスクパラメータリスト(TPL)内のエントリによって、DVNノードメモリ内の2つのバッファ、すなわちコマンドキューA又はコマンドキューBのうちの1つから命令をフェッチするように指示される。2つのバッファのデータ構造は同じである。構造が同じである目的は、一方（又は両方）の静的コマンドキューに不変の命令セットを格納するため、又は、一方（又は両方）の動的コマンドキューの中に対話式のコマンドシーケンスを構築するため、又は、マクロブロックごとに2つのバッファの間で選択するためである。第2のコマンドキューを含めることにより、1つの命令シーケンスを終了させて新しい命令シーケンスで続行し、その一方で、前の命令シーケンス全体を後から再び使用できるように保存しておくことができる。

【0075】

[118]2つのコマンドキューのそれぞれは、標準のACM PTP（又はDMA）プロトコルを使用して、DVNノードメモリに転送される（利用可能な32個の入力ポートのうちの1つ又は2つを消費する）。コマンドキューAは、ノードメモリ6の位置0x00~0x3Fにおける64個のロングワード（1ロングワードあたり32ビット）から成り、コマンドキューBは、ノードメモリ7の位置0x00~0x3Fにおける64個のロングワードから成る。

【0076】

[119]DVNのコントローラノードは、DVNのラッパーのポート変換表(PTT)のうち、割り当てられているDVN入力ポート番号に対応する位置に0x6C00を書き込むことによって、コマンドキューA(CQA)を初期化する。これにより、コマンドキューAバッファが、「メモリ=6、物理アドレス=0x000」を開始アドレスとする256バイト（64個のロングワード）のサイズに設定される。コマンドキューB(CQB)（0x6E00）の初期化値は、「メモリ=7、物理アドレス=0x000」である。コントローラは、コマンドキューA及びコマンドキューB（64ロングワードの循環バッファ）をPTP書込みを使用してロードする。コントローラは、0x6C00及び0x6E00をDVNのポート変換表における適切な位置に再び書き込むことによって、これらのバッファを再初期化する。DVNは、（0に初期化されている）6ビットカウンタを使用して、コマンドキューA又はコマンドキューBの命令をフェッチする。

【0077】

[120]図24は、コマンドキューのデータ構造を示している。ビット31は、「有効な」命令を示すために使用され、コントローラがバッファに書き込む命令ごとに1'b1にセットされていなくてはならない。DVNは、ビット31が1'b0にセットされているキューエントリを読み取ると、そのビットが1'b1にセットされる、又はリダイレクトされるまで動作を停止する。

【0078】

[121]コマンドキューA又はコマンドキューBからフェッチされた命令シーケンスの処理中、コントローラは、DVNに対して、1）一方のキューからの命令の処理を終了する、2）他方のキューへのポインタをクリアする、又は、3）他方のキューの位置0から処理を再開する、ように命令することができる。コントローラからDVNへのこの「切替」

10

20

30

40

50

コマンドでは、DVNノードラッパー内の64個のカウンタ符号の1つを利用する。コントローラは、MSBが1'b1にセットされている適切なPCT/CCTカウンタを書き込む（この場合、このビットは以前に1'b0に初期化されていたと想定する）。DVNは、処理の間、このビットを絶え間なく監視する。DVNのタスクのセットアップ時、タスクパラメータリストから6ビットレジスタをロードして、64個のカウンタ符号のうち監視する適切な1つを選択する。DVNは、「切替」ルーチンの一部として、「自己アック、カウンタ初期化（self-ack, initialize counter）」のMINワードを送ることによってこのビットをクリアし、「切替完了」メッセージをコントローラに送る（図25を参照）。

【0079】

10

[122] DVNのコマンドキューデータ構造の説明：以下では、DVNのコマンドキューにおいて使用するデータ構造について、図24を参照しながら説明する。

【0080】

[123] 1つの16×16マクロブロックの指標値の評価 — コマンド0x0：1つのポジション（X，Y）の指標値を求める。この場合、（0，0）は参照ブロックの左上のペルを示し、Xは0～79の範囲内の正の整数であり、Yは0～49の範囲内の正の整数である。これらの正の整数は、左上ペルからの変位を示している。DVNは、指標値、すなわち、SAD256、水平コスト関数、及び垂直コスト関数の合計値を評価する。コスト関数は、ノードメモリに格納されている。

【0081】

20

[124] 5×5ポジションの1つ以上のセットのうちの最良指標値を求める — コマンド0x1：ポジション（X，Y）から始まるm（水平）×n（垂直）個のセットの5×5ポジションのうち最良の指標値を求める。Xは0～79の範囲内の正の整数であり、Yは0～49の範囲内の正の整数であり、mは1～16の範囲内の正の整数であり、nは1～10の範囲内の正の整数である（図26を参照）。

【0082】

[125] ハーフペル補間 — コマンド0x2：16×16マクロブロックに対して、フルペルからハーフペルへの補間を実行する。ビット[24]は「評価」であり、すなわち、0は補間のみを意味し、1は、8（9）セットの16×16ハーフペルマクロブロックの補間及び評価を意味する。ビット[25]は「モード」であり、すなわち、0は、「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1は、（X，Y）基準（すなわちビット[14：8]及びビット[5：0]）を使用してマクロブロックを選択することを意味する。

30

【0083】

[126] クォーターペル補間 — コマンド0x3：クォーターペル補間を実行する。ビット[24]は「評価」であり、すなわち、0は補間のみを意味し、1は、8（9）セットの16×16クォーターペルマクロブロックの補間及び評価を意味する。ビット[25]は「モード」であり、すなわち、0は、ハーフペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1は、ハーフペルバッファ位置（すなわちビット[11：0]）を使用してマクロブロックを選択することを意味する。

40

【0084】

[127] フルペル符号付き差 — コマンド0x4：16×16の現在のブロックと、参照ブロックから選択される16×16フルペルマクロブロックとの間の符号付き差を求める。— 255～255の範囲内の2の補数の16ビット整数としての256個の値を、128個のロングワードにパックして、選択されている宛先に出力する。ビット[25]は「モード」であり、すなわち、0は、フルペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1は、（X，Y）基準（すなわちビット[14：8]及びビット[5：0]）を使用してマクロブロックを選択することを意味する。ビット[26]は「宛先」であり、2つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているル

50

ーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

【0085】

[128] ハーフペル符号付き差 — コマンド 0x5: 16×16 の現在のブロックと、ハーフペルバッファから選択される 16×16 のハーフペルマクロブロックとの間の符号付き差を求める。— 255～255 の範囲内の 2 の補数の 16 ビット整数としての 256 個の値を、128 個のロングワードにパックして、選択されている宛先に出力する。ビット [25] は「モード」であり、すなわち、0 は、ハーフペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1 は、ハーフペルバッファ位置（すなわちビット [11:0]）を使用してマクロブロックを選択することを意味する。ビット [26] は「宛先」であり、2 つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

10

【0086】

[129] クォーターペル符号付き差 — コマンド 0x6: 16×16 の現在のブロックと、クォーターペルバッファから選択される 16×16 のクォーターペルマクロブロックとの間の符号付き差を求める。— 255～255 の範囲内の 2 の補数の 16 ビット整数としての 256 個の値を、128 個のロングワードにパックして、選択されている宛先に出力する。ビット [25] は「モード」であり、すなわち、0 は、クォーターペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1 は、ハーフペルバッファ位置（すなわちビット [11:0]）を使用してマクロブロックを選択することを意味する。ビット [26] は「宛先」であり、2 つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

20

【0087】

[130] 16×16 フルペルマクロブロックの出力 — コマンド 0x7: DVN ノードメモリにおける参照ブロックから選択された 16×16 フルペルマクロブロックを、指定されている宛先に転送する。ビット [25] は「モード」であり、すなわち、0 は、フルペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1 は、(X, Y) 基準（すなわちビット [14:8] 及びビット [5:0]）を使用してマクロブロックを選択することを意味する。ビット [26] は「宛先」であり、2 つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

30

【0088】

[131] 16×16 ハーフペルマクロブロックの出力 — コマンド 0x8: 選択された 16×16 ハーフペルマクロブロックを、DVN ノードメモリにおけるハーフペルバッファから、指定されている宛先に転送する。ビット [25] は「モード」であり、すなわち、0 は、ハーフペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1 は、ハーフペルバッファ位置（すなわちビット [11:0]）を使用してマクロブロックを選択することを意味する。ビット [26] は「宛先」であり、2 つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

40

【0089】

[132] 16×16 クォーターペルマクロブロックの出力 — コマンド 0x9: 選択された 16×16 クォーターペルマクロブロックを、DVN ノードメモリにおけるクォーターペルバッファから、指定されている宛先に転送する。ビット [25] は「モード」であり、すなわち、0 は、クォーターペル探索からの「最良指標値」ベクトルを使用してマクロブロックを選択することを意味し、1 は、クォーターペルバッファ位置（すなわちビッ

50

ト [1 1 : 0]) を使用してマクロブロックを選択することを意味する。ビット [2 6] は「宛先」であり、2つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

【0 0 9 0】

[133]最良指標値の出力 — コマンド 0 x A : 最良指標値 (飽和した符号なし 1 6 ビット整数) を、指定されている 1 つ以上の宛先に転送する。ビット [2 5] は「モード」であり、すなわち、0 は、コントローラのみを送ることを意味し、1 は、コントローラに送り、更に、指定されている宛先に送ることを意味する。ビット [2 6] は「宛先」であり、2つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

10

【0 0 9 1】

[134]動きベクトルの出力 — コマンド 0 x B : クォーターペル解像度を有する動きベクトルを、指定されている 1 つ以上の宛先に転送する。ビット [2 5] は「モード」であり、すなわち、0 は、コントローラのみを送ることを意味し、1 は、コントローラに送り、更に、指定されている宛先に送ることを意味する。ビット [2 6] は「宛先」であり、2つの論理出力ポートのうちの一方を選択する。各ポートについて、タスクパラメータリストには、選択される宛先に関連付けられているルーティングと、入力ポート番号と、メモリモードの指示とが含まれている。

20

【0 0 9 2】

[135]完了 — コマンド 0 x C : DONE コマンドは、「現在のブロック」に対して実行すべき処理がそれ以上存在しないことを DVN に知らせる。DVN は、キューからこのコマンドを読み取ると、DONE ステータスワードをコントローラに送る。DVN の FSM は ACK + テスト状態に遷移する。

【0 0 9 3】

[136]エコー — コマンド 0 x F : ECHO コマンドは、コマンド自体であるステータスワードを戻すように DVN に命令する。これは、診断の支援を目的としている。

【0 0 9 4】

[137]DVN からコントローラへのステータスワード : DVN は、各コマンドを実行した後、ACM PTP プロトコルを使用してステータスワードをコントローラに送る。DVN のタスクパラメータリストには、適切なルーティングフィールドと、入力ポート番号と、この動作をサポートするのに必要なメモリモード指示とが含まれる。この情報は、「セットアップ」動作中及び「続行」動作中に、DVN のコントローラノードロケータレジスタに転送される。図 2 5 は、ステータスワードについてまとめてある。

30

【0 0 9 5】

[138]コマンド 0 x 0 及びコマンド 0 x 1 の場合には、ステータスワードには、その特定の現在のブロックの、その時点までの最良のフルペル指標値が含まれる。更に、ステータスワードには、水平次元については (フルペル単位で) 0 ~ 7 9 の範囲、垂直次元については 0 ~ 4 9 の範囲の対応するベクトルも含まれる。コマンド 0 x 2、すなわちフルペルからハーフペルへの補間の場合には、コマンドのビット [2 4] が 1 ' b 1 (補間及び評価) にセットされているときには、ステータスワードビット [1 5 : 0] は最良のハーフペル指標値の値を示し、ビット [1 9 : 1 6] は、9 個のハーフペル候補のうち、最良の指標値が関連付けられている候補を示す (ビット [1 9 : 1 6] のコーディングには、図 4 0 に示したハーフペル番号付け規則が使用される)。コマンドのビット [2 4] が 1 ' b 0 (補間のみに) にセットされているときには、ビット [1 5 : 0] は、以前に評価された最良のフルペル指標値の値を示し、ビット [1 9 : 1 6] は、補間の「完了」を示すため 0 x F にセットされる。

40

【0 0 9 6】

[139]コマンド 0 x 3、すなわちハーフペルからクォーターペルへの補間の場合には、

50

コマンドのビット [24] が 1 ' b 1 (補間及び評価) にセットされているときには、ステータスワードビット [15 : 0] は最良のクォーターペル指標値の値を示し、ビット [19 : 16] は、9 個のクォーターペル候補のうち、最良の指標値が関連付けられている候補を示す (ビット [19 : 16] のコーディングには、図 4 2 に示したクォーターペル番号付け規則が使用される)。コマンドのビット [24] が 1 ' b 0 (補間のみ) にセットされているときには、ビット [15 : 0] は、以前に評価された最良の指標値 (フルペル又はハーフペル (評価されていた場合) のいずれか) の値を示し、ビット [19 : 16] は、補間の「完了」を示すため 0 x F にセットされる。

【0097】

[140] コマンド 0 x 4、0 x 5、0 x 6、0 x 7、0 x 8、及び 0 x 9 の場合には、ステータスワードは、それぞれ「完了」を示す、0 x F F 0 1 0 0 0 0、0 x F F 0 2 0 0 0 0、0 x F F 0 4 0 0 0 0、0 x F F 0 8 0 0 0 0、0 x F F 1 0 0 0 0 0、及び 0 x F F 2 0 0 0 0 0 である。コマンド 0 x A の場合には、ステータスワードは、最良の指標値の飽和した符号なし 16 ビット整数である。コマンド 0 x B の場合には、ステータスワードは、クォーターペル解像度を有する動きベクトルである。コマンド 0 x C の場合には、ステータスワードは、「完了」を示す 0 x F F 4 0 0 0 0 0 である。コマンド 0 x F の場合には、ステータスワードは、コマンド自体をエコーする。これは、診断の支援を目的としている。「コマンドキューリダイレクト」フラグへの応答においては、ステータスワードは、「完了」を示す 0 x F F 8 0 0 0 0 0 である。

【0098】

[141] DVN ノードメモリ : DVN は、それぞれが 1 ワードあたり 32 ビットの 512 ワードとして編成された、8 個の 2 KB の物理メモリを含んでおり、ノードメモリの総容量は 16 KB である。これにより、クロックサイクルごとに 32 バイトのビデオデータを読み取って、これらのデータの 16 回の「差の絶対値の合計」を処理することができる。更に、16 KB の容量によって、完全に重複したデータ入力及び出力と計算のためのダブルバッファリングが可能である。

【0099】

[142] 任意の ACM は、DVN のラッパーにおけるポート変換表と、3 つのサービス (PTP、DMA、RTI) のいずれかを使用して、DVN ノードメモリに書き込むことができる。更に、K ノード (K node) が、DVN ノードメモリに対して PEEK 処理及び POKE 処理を行うことができる。

【0100】

[143] 図 2 3 は、DVN ノードのメモリマップを示している。表 3 は、割り当てられるバッファの説明である。

【表 3】

バッファ	容量	アドレス範囲 (ロングワード)
参照 ブロック	3840バイト	0x000～0x0EF 0x200～0x2EF 0x400～0x4EF 0x600～0x6EF
参照 ブロックB	3840バイト	0x100～0x1EF 0x300～0x3EF 0x500～0x5EF 0x700～0x7EF
現在の ブロックA	256バイト	0x0F0～0x0FF 0x2F0～0x2FF 0x4F0～0x4FF 0x6F0～0x6FF
現在の ブロックB	256バイト	0x1F0～0x1FF 0x3F0～0x3FF 0x5F0～0x5FF 0x7F0～0x7FF
現在の ブロックA	256バイト	0x8F0～0x8FF 0xAF0～0xAFF 0xCF0～0xCFF 0xEF0～0 EFF
現在の ブロックB	256バイト	0x9F0～0x9FF 0xBF0～0xBFF 0xDF0～0xDFF 0xFF0～0xFFF
タスクパラメータリスト	960バイト	0x800～0x8EF
バイトコード	960バイト	0xA00～0xAEF
コマンドキューA	256バイト	0xC00～0xC3F
コマンドキューB	256バイト	0xE00～0xE3F
水平コスト関数表A	256バイト	0xC40～0xC7F
水平コスト関数表B	256バイト	0xC80～0xCBF
垂直コスト関数表A	128バイト	0xE40～0xE5F
垂直コスト関数表B	128バイト	0xE80～0xE9F
フラクショナルペル コスト関数表	128バイト	0xEA0～0xEBF
ハーフペル バッファ	1280バイト	0x980～0x9CF 0xB80～0xBCF 0xD80～0xDCF 0xF80～0xFCF
クォーターペル バッファ	2048 バイト	0x900～0x97F 0xB00～0xB7F 0xD00～0xD7F 0xF00～0xF7F

表 3 : DVNノードのメモリマップ

【0101】

[144]この場合、参照ブロック及び現在のブロックが「データムーバ (Data Mover)」からMINを介して8個のDVNノードメモリに転送されることを想定している。更に、「データムーバ」がPSNノードによって制御されることを想定している。参照ブロックは、Mペル (行あたり) × Nペル (列あたり) として編成されている。M及び

10

20

30

40

50

Nは5の倍数の整数であり、これは、一度に 5×5 ポジションを探索する方式に矛盾しない。Mの範囲は20～95（20，25，30，．．．，90，95）であり、Nの範囲は20～65（20，25，30，．．．，60，65）である。

【0102】

[145]参照ブロックのオプション：図26には、参照ブロックの容量である3840バイトを超えることのない、MとNの許容される組合せと、網羅的な探索処理サイクルに伴う数とをまとめている。

【0103】

[146]Mペル×Nペルのそれぞれの場合、データムーバからDVNノードメモリへの参照ブロックの転送は、次のように行われる。すなわち、行0，4，8，．．．， $[M - (M) \bmod 4]$ がノードメモリ0における連続する位置に書き込まれ、行1，5，9，．．．， $[M - (M - 1) \bmod 4]$ がノードメモリ1における連続する位置に書き込まれ、行2，6，10，．．．， $[M - (M - 2) \bmod 4]$ がノードメモリ2における連続する位置に書き込まれ、行3，7，11，．．．， $[M - (M - 3) \bmod 4]$ がノードメモリ3における連続する位置に書き込まれる。

【0104】

[147]16ペル×16ペルの現在のブロックそれぞれについては、次のようにデータムーバからDVNノードメモリに転送される。すなわち、行0，4，8，12がノードメモリ0及び4に書き込まれ、行1，5，9，13がノードメモリ1及び5に書き込まれ、行2，6，10，14がノードメモリ2及び6に書き込まれ、行3，7，11，15がノードメモリ3及び7に書き込まれる。

【0105】

[148]フルペル探索時には、DVNの動き推定エンジン（ME）は、参照ブロックのデータをメモリ0～3から読み取り、現在のブロックのデータをメモリ4～7から読み取る。ハーフペル補間時には、動き推定エンジンは、フルペル参照ブロックのデータをメモリ0～3から読み取り、ハーフペル補間したデータをメモリ4～7に書き込む。ハーフペル探索がハーフペル補間と同時に実行されるときには、動き推定エンジンは、現在のブロックのデータをメモリ4～7から読み取る。クォーターペル補間時には、動き推定エンジンは、ハーフペル補間したデータをメモリ4～7から読み取り、クォーターペル補間したデータをメモリ4～7に書き込む。クォーターペル探索がクォーターペル補間と同時に実行されるときには、動き推定エンジンは、現在のブロックのデータをメモリ0～3から読み取る。出力時には、動き推定エンジンは、フルペルマクロブロックをメモリ0～3から読み取り、フラクショナルペルマクロブロックをメモリ4～7から読み取り、マクロブロックの符号付き差の場合には、現在のブロックをメモリ0～3から、フルペル及びフラクショナルペルマクロブロックをメモリ4～7から読み取る。

【0106】

[149]コスト関数表：フルペル探索ポジションのそれぞれについて、DVNは、指標値、すなわち、SAD256、水平コスト関数、及び垂直コスト関数の合計を評価する。コスト関数はノードメモリに格納されている。コスト関数は、符号なし16ビット整数である。

【0107】

[150]M列×N行の参照ブロックに対して、M個の水平コスト関数とN個の垂直コスト関数が存在する。水平コスト関数表のそれぞれには、2つの16ビット整数として一对の水平コスト関数が、表における32ビットのロングワードのそれぞれにパックされている。列0及び列1のコスト関数がテーブル位置0に格納されており、列2及び列3のコスト関数がテーブル位置1に格納されており、以下同様である。偶数番号の列のコスト関数はロングワードのビット[15：0]に格納されており、奇数番号の列のコスト関数はロングワードのビット[31：16]に格納されている。

【0108】

[151]垂直コスト関数表のそれぞれには、2つの16ビット整数として一对の垂直コス

10

20

30

40

50

ト関数が、表における32ビットのロングワードのそれぞれにパックされている。行0及び行1のコスト関数がテーブル位置0に格納されており、行2及び行3のコスト関数がテーブル位置1に格納されており、以下同様である。偶数番号の行のコスト関数はロングワードのビット[15:0]に格納されており、奇数番号の行のコスト関数はロングワードのビット[31:16]に格納されている。図27は、フラクショナルペルのコスト関数と、コスト関数表A及びBとを示している。

【0109】

[152]参照ブロックの概要：図23に示したように、DVNノードメモリには、3840バイトの2つの参照ブロックバッファが割り当てられている。クロック周期ごとに16ペルにアクセスできるように、各バッファは4個の物理メモリに格納されている。本発明においては、左上ペルを基準とする番号付け規則を採用しており、左上ペルの(H)水平座標及び(V)垂直座標を(H:0, V:0)と表す。参照ブロックAの左上ペルは、メモリ位置0x0000の上位バイトに格納され、参照ブロックBの左上ペルは、メモリ位置0x1000の上位バイトに格納される。参照ブロックの行0, 4, 8, . . . がメモリ0に格納され、行1, 5, 9, . . . がメモリ1に格納され、行2, 6, 10, . . . がメモリ2に格納され、行3, 7, 11, . . . がメモリ3に格納される。

【0110】

[153]M=50列×N=50行の参照ブロックは、図28に示したようにノードメモリ0~3に格納される。16ペル×16ペルの現在のブロックのそれぞれは、図29に示したように、ノードメモリ0~3とノードメモリ4~7とに格納される。

【0111】

[154]図30に示したように、50ペル×50ペルの参照ブロックでは、([H:17, V:17]における中央の探索ポジションを基準として)各次元において±17ペルの探索範囲が可能である。50ペル×50ペルの参照ブロックの中には、1225個の候補探索ポジションが存在する。図30には、相対座標H(17), V(17)の「中央」ポジションを示してある。図30には、5個の探索ポジション、すなわち中央ポジションH(17), V(17)、左上ポジションH(0), V(0)、右上ポジションH(34), V(0)、左下ポジションH(0), V(34)、右下ポジションH(34), V(34)を、強調して示してある。

【0112】

[155]図31は、そのような参照ブロック内の中央の5×5ポジションを同時に探索する状況を示している。この場合、探索されるポジションの合計数は25であり、探索範囲は各次元において±2ペルである。図32は、そのような参照ブロック内の中央の9セットの5×5ポジションを探索する状況を示している。この場合、探索されるポジションの合計数は、9×5×5=225であり、探索範囲は各次元において±7ペルである。図33は、そのような参照ブロック内の49セットの5×5ポジションすべてを網羅的に探索する状況を示している。この場合、探索されるポジションの合計数は、49×5×5=1225であり、探索範囲は各次元において±17ペルである。

【0113】

[156]フラクショナルペル処理が含まれるときには、プログラマは次の2つの方式から選択することができる。

- ・ あらゆる探索結果に対する補間がサポートされるような十分な数のペルを有する「オーバーサイズの」参照ブロックをロードする。そのような「オーバーサイズの」参照ブロックのサイズは、現在のマクロブロックのサイズ(m)と、探索範囲(h, v)と、ハーフペル補間フィルタのタップ数(n)とによって決定することができる。具体的には、このサイズは、[m+h+n-1]水平ペル×[m+v+n-1]垂直ペルである。例えば、図34に示したように、16×16ペルマクロブロック、水平方向及び垂直方向に±17ペルの探索範囲、H. 264の6タップ補間フィルタの場合、参照ブロックは56ペル×56ペル(3136バイト)でなければならない。

- ・ 望ましい探索範囲がサポートされる最小数のペルを有する参照ブロックをロードする

。そのような参照ブロックのサイズは、現在のマクロブロックのサイズ (m) と探索範囲 (h, v) とによって決定することができる。具体的には、このサイズは、 $[m+h-1]$ 水平ペル $\times$ $[m+v-1]$ 垂直ペルである。例えば、 16×16 ペルマクロブロック、水平方向及び垂直方向に ± 17 ペルの探索範囲の場合、参照ブロックは 50 ペル $\times$ 50 ペル (2500 バイト) であり、上の方式1) の場合に要求される 3136 バイトよりもかなり小さい。次いで、探索結果において必要であれば、補間ステップに進む前に、 $[m+n]$ 水平ペル $\times$ $[m+n]$ 垂直ペルを有する参照ブロックバッファを再ロードする。この方式は、図35にまとめてある。

【0114】

[157]フルペル探索：DVNは、フルペル探索の2つの動作モードをサポートしている。一方のモードでは、参照ブロック内の任意の 16×16 ポジションを評価し、クロック周期ごとに16回のSAD演算を実行する。第2のモードでは、 $m \times n$ セットの ($5 \times 5 = 25$) ポジションを同時に評価し、クロック周期ごとに最大200回 (平均160回) のSAD演算を実行する。いずれのモードの場合も、「評価」は、現在の 16×16 マクロブロックと、参照ブロック内の任意の 16×16 マクロブロックとの間の「差の絶対値の合計」(SAD) 指標値と、水平コスト関数及び垂直コスト関数との合計を計算し、最良の指標値及びそれに対応する 16×16 マクロブロックの原点をつねに保存するステップから成る。探索パターンは、DVNメモリ内の2つのコマンドキューの1つに制御PSNノードによって書き込まれる命令シーケンスによって制御される。原点 (X, Y) に対して1つのポジションを評価するのか $m \times n$ セットの 5×5 ポジションを評価するのかは、各キューエントリの一意のコマンドコードによって指定される。

【0115】

[158]ハーフペル補間：変形指標値が最良の 16×16 マクロブロックを求めるフルペル探索の後、周囲の ($[16+n] \times [16+n]$) ペルの配列に対してハーフペル補間を実行する (nはフィルタ次数)。双一次補間、WMV、H. 264、MPEG4の場合、それぞれ、 $n=2, 4, 6, 8$ である。図36は、ハーフペル双一次補間に必要な計算についてまとめてある。図37は、「最良指標値」の 16×16 マクロブロックに対応する 18×18 フルペル配列に対して双一次補間を行う結果としての 33×33 ハーフペル配列を示している。図40は、H. 264の6タップフィルタ補間の結果としての 33×33 ハーフペル配列を示している。図41は、サポートされるハーフペル補間FIRフィルタについてまとめてある。水平方向及び垂直方向に $1/2$ ペルだけシフトしたハーフペルポジションの場合、フルペル及びハーフペル補間された値に対して同じフィルタが使用され、これらの値のフル精度が維持される。

【0116】

[159]ハーフペル探索：4個のフィルタリングオプションのうちの1つを使用してのハーフペル補間の後、9個の候補ハーフペルポジションのそれぞれについてハーフペル指標値を計算する。この指標値は、SAD256指標値とハーフペルコスト関数との合計である。図42は、9個の候補ハーフペルポジションの番号付け規則を示している。この図において、フルペルポジション (ポジション「5」) のフルペル指標値は、以前にすでに計算されている。しかしながら、ポジション「5」のハーフペル指標値は、そのフルペル指標値からフルペル水平コスト関数及びフルペル垂直コスト関数を減じ、そのポジションのハーフペルコスト関数を加えた値である (図27を参照)。ハーフペル探索は、9個の候補のそれぞれのSADの指標値+ハーフペルコスト関数を生成するステップと、指標値が最良の候補を選択するステップとから成る。

【0117】

[160]クォーターペル補間：変形指標値が最良の 16×16 マクロブロックを求めるハーフペル探索の後、周囲の ($[16+n] \times [16+n]$) ペルの配列に対してクォーターペル補間を実行する (nはフィルタ次数)。WMVの場合には $n=4$ であり、H. 264及びMPEG4 (双一次補間を使用する) の場合には、 $n=2$ である。図42は、クォーターペルポジション、ハーフペルポジション、及び $3/4$ ペルポジションに対するWM

Vサブペル4タップFIRフィルタの要約を示している。

【0118】

[161]クォーターペル探索：クォーターペル補間の後、9個の候補クォーターペルポジションのそれぞれについてクォーターペル指標値を計算する。この指標値は、SAD256指標値とクォーターペルコスト関数との合計である。図43は、9個の候補クォーターペルポジションの番号付け規則を示している。この図においては、16個のクォーターペルポジションのうち9個を計算するのみでよい。更に、9個の候補クォーターペルポジションのうちの1つ（ポジション「5」）の指標値は、以前にすでに計算されている。しかしながら、ポジション「5」のクォーターペル指標値は、そのハーフペル指標値からハーフペルコスト関数を減じ、そのポジションのクォーターペルコスト関数を加えた値である（図27を参照）。クォーターペル探索は、9個の候補のそれぞれのSAD指標値＋クォーターペルコスト関数を生成するステップと、指標値が最良の候補を選択するステップとから成る。

10

【0119】

[162]マクロブロックの符号付き差：DVNは、DVNノードメモリ内の任意の16ペル×16ペルマクロブロック（フルペル、ハーフペル、又はクォーターペル）と、DVNノードメモリ内のいずれかの16ペル×16ペルの現在のブロックとの間の、9ビット符号付き差の256要素の配列を計算及び出力することもある。この配列は、128個のロングワードとして宛先ノードに出力され、この場合に各ロングワードは、16×16マクロブロックの符号拡張された9ビット符号付き差を表す、パックされた一対の16ビット符号付き整数である。

20

【0120】

[163]DVNに関連するプログラミング要件の要約：図13を参照して、DVNのプログラミングに要求されるステップは次のとおりである。すなわち、タスクパラメタリスト（TPL）を構築してDVNノードメモリにロードする、バイトコードを構築してDVNノードメモリにロードする、DVNハードウェアタスクマネージャ（HTM）を構成する、必要な参照ブロック、現在のブロック、及びコスト関数表と、適切なデータフローACKとを構築してDVNノードメモリに転送するように、データムーバに指示する、必要なコマンドキュー命令シーケンスを構築してDVNノードメモリに転送する、DVNからのメッセージの処理がサポートされるようにDVNのコントローラノード内のハードウェアタスクマネージャを構成する、DVNからの出力を受信し且つ適切なデータフローACKをDVNに送るようにDVN宛先ノード（消費者）を構成する。

30

【0121】

[164]DVNハードウェアの概要：図43は、DVN実行ユニットの簡略化したブロック図を示している。DVNは、QSTノードラッパーと、8個の2KBのノードメモリ（それぞれ32ビットの512ワードとして編成されている）と、実行ユニット（EU）とを含んでおり、実行ユニット（EU）は、オーバーヘッドプロセッサ（OHP）と動き推定エンジン（ME）とから成る。動き推定エンジンは、フルペル探索、ハーフペル探索、及びクォーターペル探索と、ハーフペル補間及びクォーターペル補間と、マクロブロックの符号付き差とを実行する。動き推定エンジンは、最良の指標値及びそれに対応する16×16マクロブロックの原点を、コントローラノードと宛先ノードとに出力し、対応するその16×16マクロブロックを、復号器の動き補償機能を実行する宛先ノードに出力する。動き推定エンジンは、マクロブロックの符号付き差を宛先ノードに出力することができる。

40

【0122】

[165]DVNの特徴としては、効率を最大にするため動き推定エンジンとオーバーヘッドプロセッサとに個別の制御ユニットが設けられている、バイトコードがコンパクトである、セットアップ、ACK+テスト、及び続行のオーバーヘッドが最小になるようにタスクパラメタリストのロード動作及び格納動作が効率的である、シーケンサ（sequence）及び制御メモリに基づく実施形態と比較して面積あたりパワーの指標値が高く

50

なるように、動き推定エンジンに対して有限状態機械（F S M）に基づく P L A 制御が行われる、パフォーマンスが高まりパワーが低減する革新的なアルゴリズムの実施形態、大規模な並行処理が可能である 8 個の物理メモリブロックアーキテクチャ、「書込みアドレス及び読取りアドレスが同一」方式（r e s o l u t i o n）の高性能のメモリインタフェース、が挙げられる。

【0123】

[166]動き推定エンジンは、いくつかの機能要素、すなわち、制御プログラムユニット（C P U）、メモリアドレスジェネレータユニット（A G U）、データパスユニット（D P U）、差の絶対値の合計（S A D）、符号付き差、複数形式のサブペル補間、ラッパーインタフェースユニット（W I U）、メモリインタフェースユニット（M I U）を含んでいる。

10

【0124】

[167] D V N は、以下の動作を行う。1）（最大）80 ペル（水平次元）及び（最大）50 ペル（垂直次元）にわたるフルペル探索、2）4 つ（おそらくはそれ以上）のフィルタのうちの 1 つを使用してのフルペルから 35 × 35 ハーフペルへの補間、3）ハーフペル探索、4）35 × 35 ハーフペルから 9 × 16 × 16 クォーターペルへの補間、5）クォーターペル探索、6）最良の指標値及びそれに対応する 16 × 16 マクロブロックを制御 P S N ノードに出力し、16 × 16 マクロブロック自体を、復号器動き補償を実行するノードに出力する、7）選択された 16 × 16 マクロブロック（フルペル、ハーフペル、又はクォーターペル）と現在の 16 × 16 マクロブロックとの間の 256 個の符号付き差を計算して出力する。

20

【0125】

[168] 8 個のノードメモリのそれぞれは、完全に重複したデータ入力と計算及び結果出力が可能であるように、ピンポンバッファ対に分割されている。データムーバから 8 個の D V N メモリに 1 つのデータセットが転送された後、D V N の動き推定エンジンは、そのデータセットの計算を実行して結果を出力する。この間、次のデータセットがデータムーバから 8 個の D V N メモリに転送される。以下同様である。

【0126】

[169] 各データセットは、可変サイズの参照ブロックと、16 × 16 ペルの配列（現在のブロック）と、フラクショナルペルコスト関数表とを含んでいる。これらのデータは、動き推定エンジンからデータへのアクセスが最適化されるような方式で 8 個のノードメモリの間で分散されている。動き推定エンジンからデータへのアクセスには、一般的には、クロック周期ごとに 8 個のノードメモリのそれぞれから 4 バイトデータへのアクセスが含まれる。これにより、一度に 1 セットの 16 × 16 探索ポジションが評価されるときに、クロック周期ごとに 16 回の S A D 演算を実行することができる。更に、「25 ポジション同時探索」モードの場合、クロック周期ごとに、参照ブロックの 4 行からのデータと、現在のブロックの 4 行からのデータとにアクセスすることができる。

30

【0127】

[170] 本発明のルーチンは、任意の適切なプログラミング言語、例えば、C 言語、C ++ 言語、J a v a 言語、アセンブリ言語などを使用して実装することができる。プロシージャ指向やオブジェクト指向など、様々なプログラミング手法を採用することができる。ルーチンは、1 つの処理デバイス或いは複数のプロセッサにおいて実行することができる。ステップ、動作、或いは計算は、特定の順序で提示したが、異なる実施形態においてはこの順序を変更することができる。本明細書において連続するものとして示した複数のステップは、いくつかの実施形態においては、同時に実行することができる。本明細書に説明した一連の動作には、別のプロセス（例：オペレーティングシステム、カーネル）による割込み、中断、又はそれ以外の制御を行うことができる。ルーチンは、オペレーティングシステム環境において動作する、或いは、システム処理のすべて又は相当な部分を占有するスタンドアロンのルーチンとして動作することができる。機能は、ハードウェア、ソフトウェア、又はその両方の組合せにおいて実行することができる。特に明記しない限り

40

50

、機能は、全体又は一部を手操作によって実行することもできる。

【0128】

[171]本明細書の説明では、本発明の実施形態が完全に理解されるように、コンポーネントの例や方法の例など具体的な詳細を豊富に記載した。しかしながら、これら具体的な細部の少なくとも1つを使用せずに、或いは別の装置、システム、アセンブリ、方法、コンポーネント、材料、部品、その他を使用して、本発明の実施形態を実施できることが、当業者には認識されるであろう。別の例においては、本発明の実施形態の側面があいまいになることを避けるため、周知の構造、材料、動作については、具体的な図示或いは詳細な説明は省いた。

【0129】

[172]本発明の実施形態のための「コンピュータ可読媒体」は、命令を実行するシステム、装置、システム、又はデバイスにおいて使用するプログラム、又は一緒に使用するプログラムを含む、格納する、通信する、伝える、又は転送することのできる任意の媒体とすることができる。コンピュータ可読媒体は、一例として、電子、磁気、光学、電磁気、赤外線、又は半導体に基づくシステム、装置、デバイス、伝播媒体、又はコンピュータメモリとすることができる（ただしこれらに限定されない）。

【0130】

[173]「プロセッサ」又は「プロセス」には、データ、信号、又はその他の情報を処理する、ヒューマン方式、ハードウェア方式、ソフトウェア方式の、任意のシステム、機構、コンポーネントの1つ以上が含まれる。プロセッサとしては、汎用の中央処理装置、複数の処理ユニット、機能を達成する専用回路、又はその他のシステムを備えたシステムが挙げられる。処理は、空間的な位置に限定されたり、時間的に制約される必要はない。例えば、プロセッサは、「リアルタイム」で、「オフライン」で、或いは「バッチモード」において機能を実行することができる。処理の一部を、別の時間に、別の位置において、別の（又は同じ）処理システムによって実行することができる。

【0131】

[174]本明細書全体を通じて、「1つの実施形態」、「実施形態」、又は「具体的な実施形態」という記述は、実施形態に関連して説明されている特定の特徴、構造、又は特性が、本発明の少なくとも1つの実施形態に含まれていることを意味し、これらが必ずしもすべての実施形態に含まれているわけではない。従って、本明細書の中の様々な場所で使われている「1つの実施形態において」、「実施形態において」、又は「具体的な実施形態において」という句は、必ずしも同じ実施形態を指しているわけではない。更に、本発明の任意の具体的な実施形態の特定の特徴、構造、又は特性を、任意の好適な方式において1つ以上の別の実施形態と組み合わせることができる。なお、本明細書における教えに鑑み、本明細書に説明及び図解した本発明の実施形態の別の変形形態及び変更形態が可能であり、これらは本発明の概念及び範囲の一部であるとみなされることを理解されたい。

【0132】

[175]本発明の実施形態は、プログラムされている汎用のデジタルコンピュータを使用することによって実施することができ、この場合、ASIC（特定用途向け集積回路）、PLD（プログラマブルロジックデバイス）、FPGA（フィールドプログラマブルゲートアレイ）、或いは、光学的、化学的、生物学的、若しくは量子工学、又はナノ工学に基づくシステム、コンポーネント、機構を使用することができる。一般的には、本発明の機能は、この技術分野において公知である任意の手段によって達成することができる。分散型又はネットワーク型のシステム、コンポーネント、及び回路を使用することができる。データの伝達又は転送は、有線、無線、又はその他の任意の手段によることができる。

【0133】

[176]また、図面に描いた要素の1つ以上は、特定の用途において有用であるとき、更に分割して、又は統合して、或いは一部の要素を省いて、或いは特定の場合には動作不能なように、実施できることも理解されるであろう。また、上述した方法をコンピュータが実行することができるように、機械可読媒体に格納することのできるプログラム又はコー

10

20

30

40

50

ドを実装することも、本発明の概念及び範囲内である。

【0134】

[177]更には、図面における信号矢印は、特に明記しない限りは、制限ではなく例示のみを目的とするものとみなされたい。更に、本明細書において使用している語「又は、或いは、若しくは」は、特に明記しない限り、一般には「そのうちの少なくとも1つ」を意味するものとする。コンポーネント又はステップの組合せも、記載されているものとみなされ、その場合、分割又は組合せに関して不明確であるため専門用語は予測されるものとする。

【0135】

[178]本明細書の説明及び請求項全体を通じて使用されているとき、「a」、「an」、「the」は、単数であることが文脈から明らかでない限りは、複数の存在も含まれる。また、本明細書の説明及び請求項全体を通じて使用されているとき、「における、の中の」の意味には、文脈によって特に明らかでない限りは、「の」も含まれる。

【0136】

[179]本発明の図解した実施形態の上記の説明（要約書に記載されている内容も含む）は、本発明を網羅的に説明すること、或いは、本明細書に開示した正確な形態に本発明を限定することを目的とするものではない。本明細書においては、本発明の具体的な実施形態及び例を例示のみを目的として説明しているが、関連する技術分野における当業者には認識、理解されるように、本発明の概念及び範囲内で様々な同等の変更形態が可能である。上述したように、これらの変更は、本発明の図解した実施形態の上記の説明に照らして本発明に行うことができるものであり、本発明の概念及び範囲に含まれるものとする。

【0137】

[180]従って、本明細書においては、本発明についてその具体的な実施形態を参照しながら説明したが、上記の開示においては様々な修正、変更及び置換えが意図されており、場合によっては、記載した本発明の範囲及び概念から逸脱することなく、本発明の実施形態のいくつかの特徴が、それ以外の特徴を対応して使用せずに採用されることが理解されるであろう。従って、特定の状況又は材料に適合するように、本発明の本質的な範囲及び概念に多数の変更を行うことができる。本発明は、請求項、若しくは、本発明を実施するうえで考えられる最良の形態として開示した具体的な実施形態、又はその両方において使用している特定の用語に制限されることはなく、本発明には、添付の請求項の範囲内に該当するあらゆる実施形態及び同等の形態が含まれるものとする。

【図面の簡単な説明】

【0138】

【図1】本発明の実施形態によるビデオプロセッサの高レベル機能のブロック図を示している。

【図2】本発明の実施形態によるドメインビデオノード（DVN）を示している。

【図3】本発明の実施形態による、中央の探索ポジションの定義方式を示している。

【図4】本発明の実施形態による、 48×48 の参照ブロック内の探索領域を示している。

【図5】本発明の実施形態による、中央の 8×5 のポジションを同時に探索するプロセスを示している。

【図6】本発明の実施形態による、中央の9セットの 8×5 のポジションを探索するプロセスを示している。

【図7】本発明の実施形態による、ハーフペル双一次補間のプロセスを示している。

【図8】本発明の実施形態による、フルペルからハーフペルへの双一次補間のプロセスを示している。

【図9】本発明の実施形態による、ハーフペルの番号付け規則を示している。

【図10】本発明の実施形態による、6タップフィルタに基づくH. 264のハーフペル補間のプロセスを示している。

【図11】本発明の実施形態による、6タップフィルタに基づくH. 264のハーフペル

補間のプロセスを示している。

【図 1 2】本発明の実施形態による、適応コンピューティングマシン（ACM）のノードの要素のいくつかを示している。

【図 1 3】本発明の実施形態による、ドメインビデオノード（DVN）のプログラミングモデル内のハードウェア要素のいくつかを示している。

【図 1 4】本発明の実施形態による、有限状態機械（FSM）の状態遷移表を示している。

【図 1 5】本発明の実施形態による、DVNのタスクパラメータリスト（TPL）を示している。

【図 1 6】本発明の実施形態による、タスクパラメータリスト内の位置 0 x 0 0 ~ 0 x 0 3 のフォーマットを示している。 10

【図 1 7】本発明の実施形態による、タスクパラメータリスト内の位置 0 x 0 4 ~ 0 x 0 5 のフォーマットを示している。

【図 1 8】本発明の実施形態による、タスクパラメータリスト内のACK+テストのパラメータのフォーマットを示している。

【図 1 9】本発明の実施形態による、セットアップ及び続行のレジスタのフォーマットを示している。

【図 2 0】本発明の実施形態による、セットアップのバイトコード及び続行のバイトコードのフォーマットを示している。

【図 2 1 A】本発明の実施形態による、ACK+テストのバイトコードのフォーマットを示している。 20

【図 2 1 B】本発明の実施形態による、ACK+テストのバイトコードのフォーマットを示している。

【図 2 1 C】本発明の実施形態による、ACK+テストのバイトコードのフォーマットを示している。

【図 2 1 D】本発明の実施形態による、ACK+テストのバイトコードのフォーマットを示している。

【図 2 1 E】本発明の実施形態による、ACK+テストのバイトコードのフォーマットを示している。

【図 2 2】本発明の実施形態による、「一般的な」タスクのタスクパラメータリストエントリ及びバイトコードのフォーマットを示している。 30

【図 2 3】本発明の実施形態による、32ビットの512ワードを有する8バンクとして編成されているDVNメモリのメモリマップを示している。

【図 2 4】本発明の実施形態による、コマンドバッファのデータ構造のフォーマットを示している。

【図 2 5】本発明の実施形態による、「コントローラへの」ステータスワードのフォーマットを示している。

【図 2 6】本発明の実施形態による、探索範囲がM×Nセットの5×5である場合の、メモリ及びクロックサイクルに従っての参照ブロックのロングワードのフォーマットを示している。 40

【図 2 7】本発明の実施形態による、フラクショナルペルコスト関数及びテーブルの形式を示している。

【図 2 8】本発明の実施形態による、M=50列及びN=50行を有する参照ブロックのメモリレイアウトを示している。

【図 2 9】本発明の実施形態による、サイズが16ペル×16ペルである現在のブロックのメモリレイアウトを示している。

【図 3 0】本発明の実施形態による、50×50の参照ブロック内の探索領域を示している。

【図 3 1】本発明の実施形態による、中央の5×5のポジションを同時に探索するプロセスを示している。 50

【図 3 2】本発明の実施形態による、中央の 9 セットの 5×5 ポジションを同時に探索するプロセスを示している。

【図 3 3】本発明の実施形態による、49 セットの 5×5 ポジションを網羅的に探索するプロセスを示している。

【図 3 4】本発明の実施形態による、事前補間バッファを再ロードする必要のない、探索範囲 ± 17 の場合の参照ブロックのサイズを示している。

【図 3 5】本発明の実施形態による、事前補間バッファを再ロードする必要のあるフルペル探索領域を示している。

【図 3 6】本発明の実施形態による、フルペルからハーフペルへの双一次補間を示している。

10

【図 3 7】本発明の実施形態による、ハーフペル双一次補間を行った配列を示している。

【図 3 8】本発明の実施形態による、6 タップフィルタに基づく H. 264 ビデオのハーフペル補間を示している。

【図 3 9】本発明の実施形態による、ハーフペル補間の有限インパルス応答 (FIR) フィルタの要約を示している。

【図 4 0】本発明の実施形態による、ハーフペルの番号付け規則を示している。

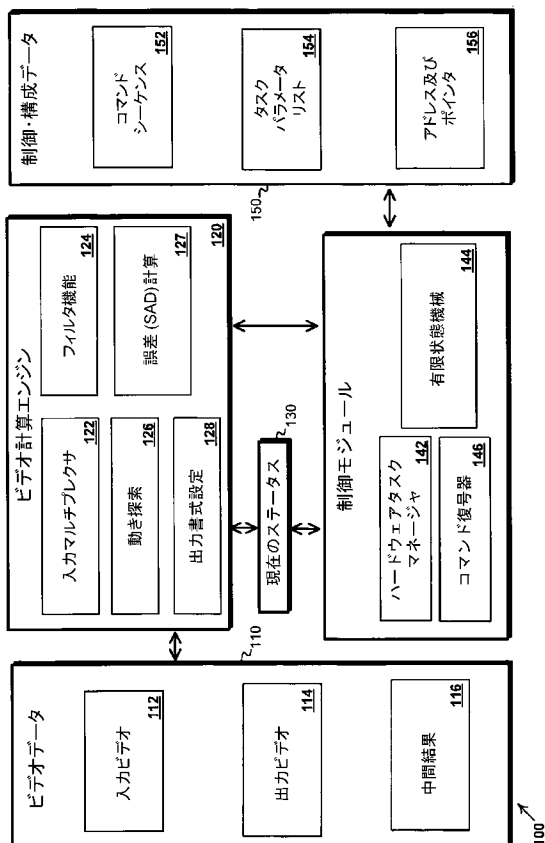
【図 4 1】本発明の実施形態による、Microsoft Windows(R) Media Video (WMV) の場合のサブペル補間 FIR フィルタの要約を示している。

【図 4 2】本発明の実施形態による、クォーターペルの番号付け規則を示している。

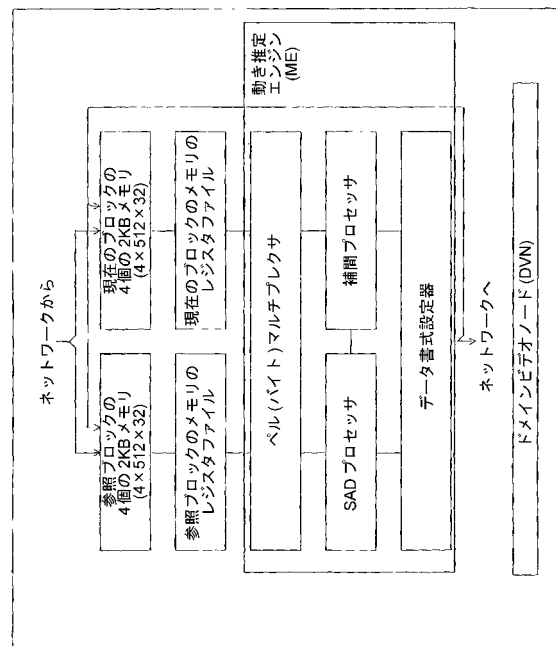
【図 4 3】本発明の実施形態による、DVN の実行ユニットを示している。

20

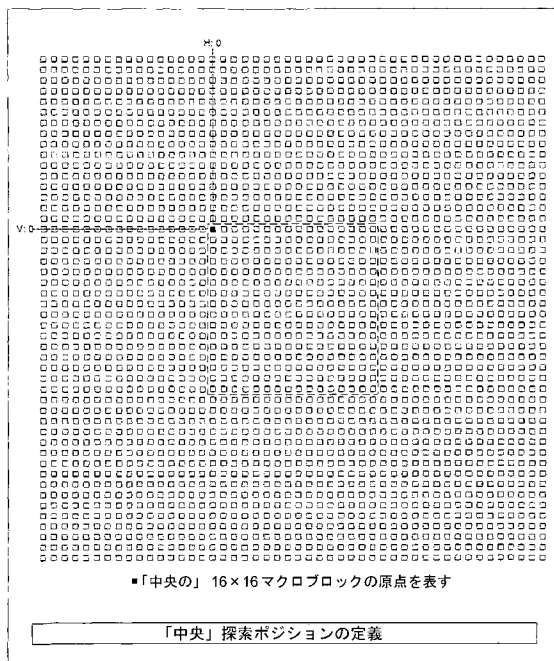
【図 1】



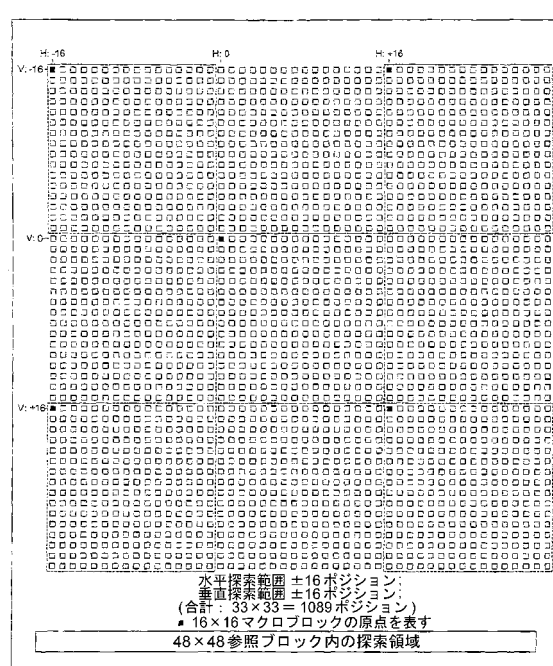
【図 2】



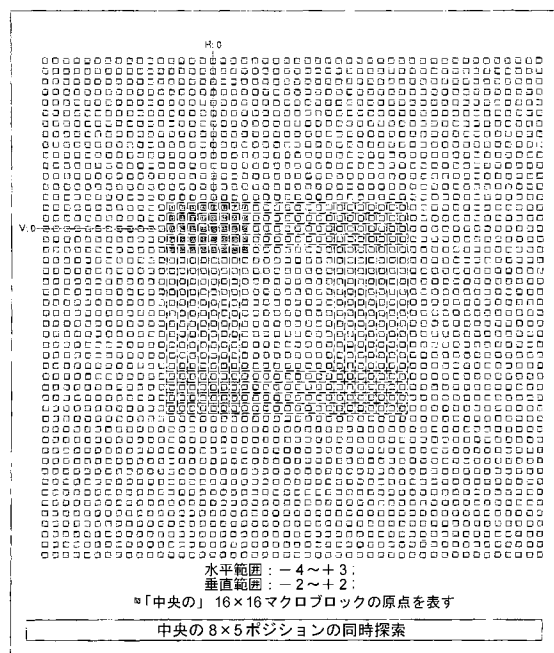
【図 3】



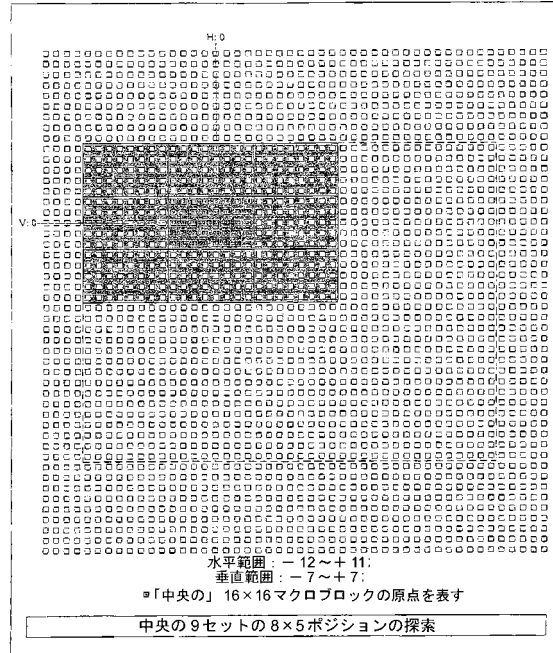
【図 4】



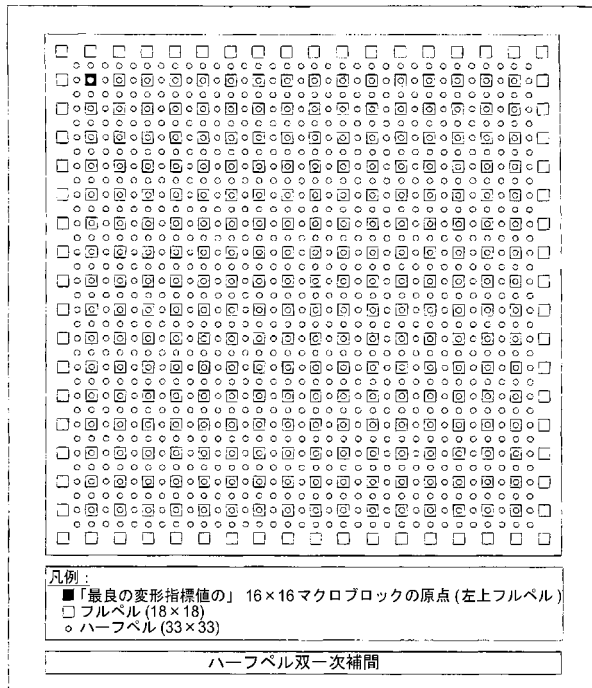
【図 5】



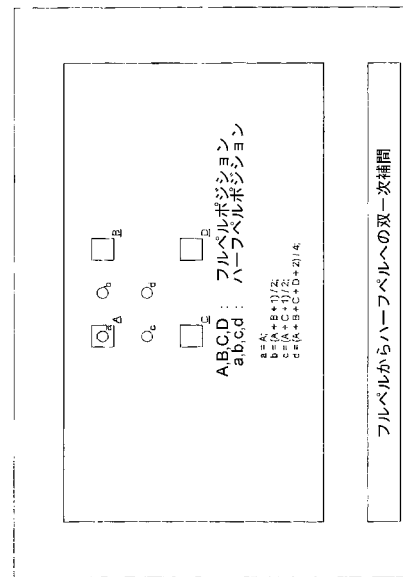
【図 6】



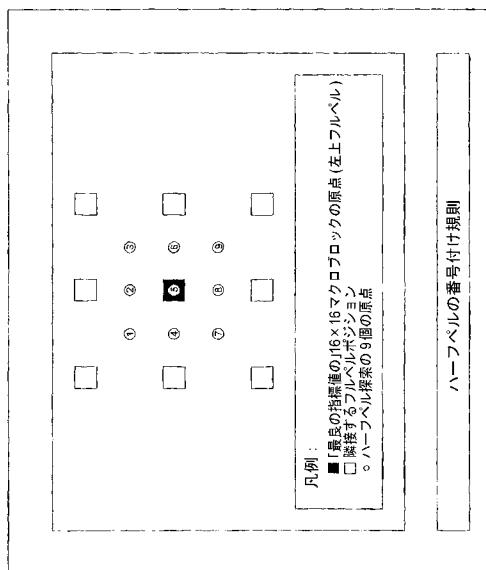
【図 7】



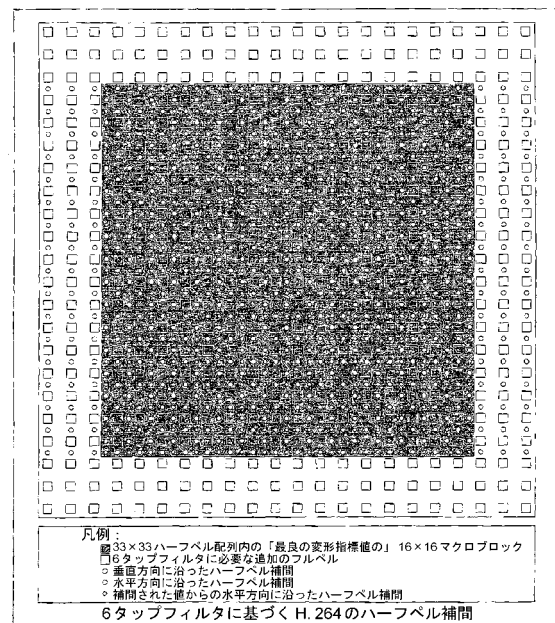
【図 8】



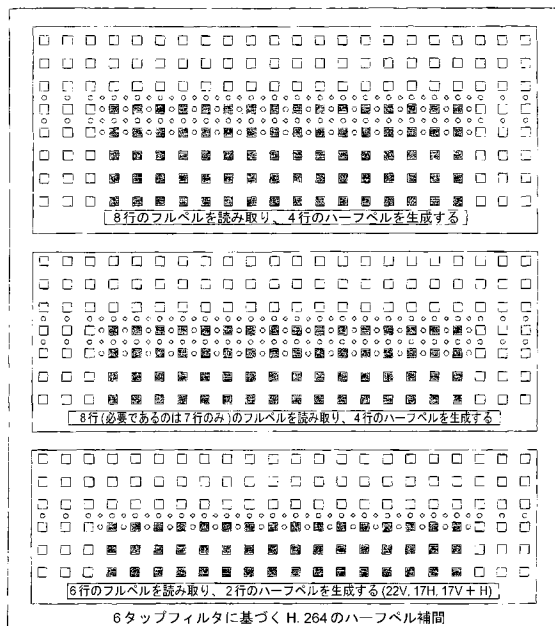
【図 9】



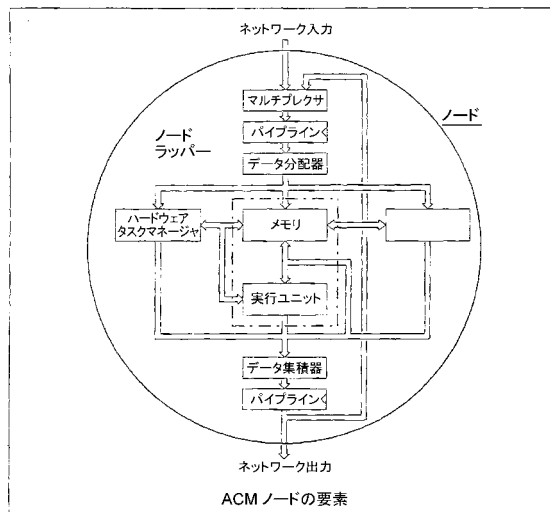
【図 10】



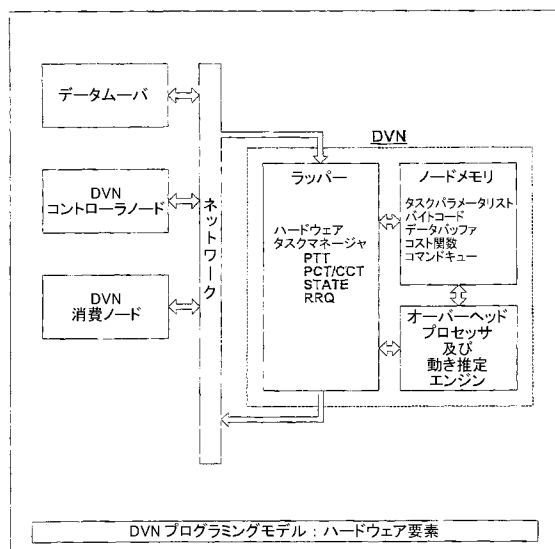
【図 1 1】



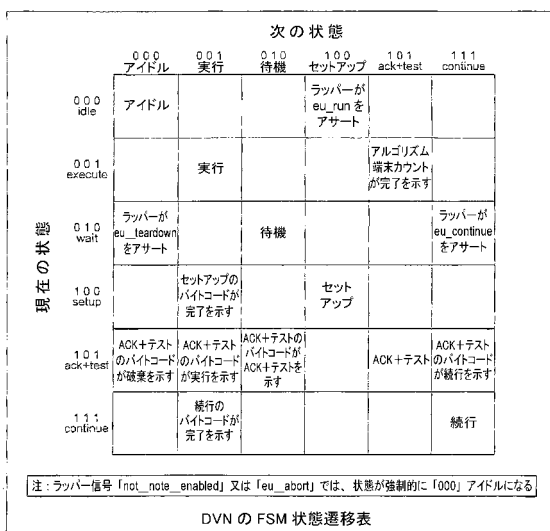
【図 1 2】



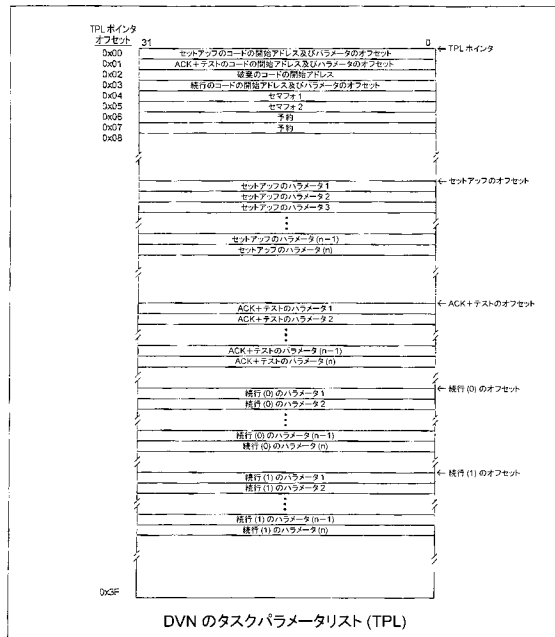
【図 1 3】



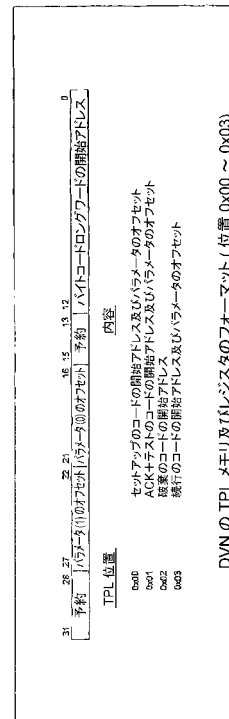
【図 1 4】



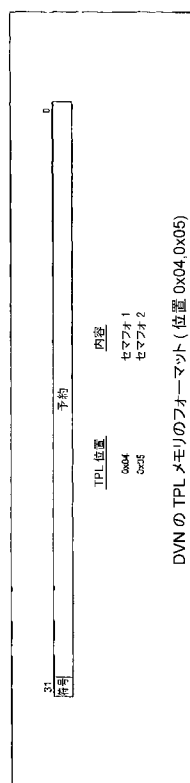
【図 15】



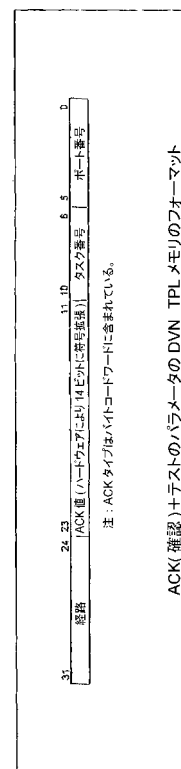
【図 16】



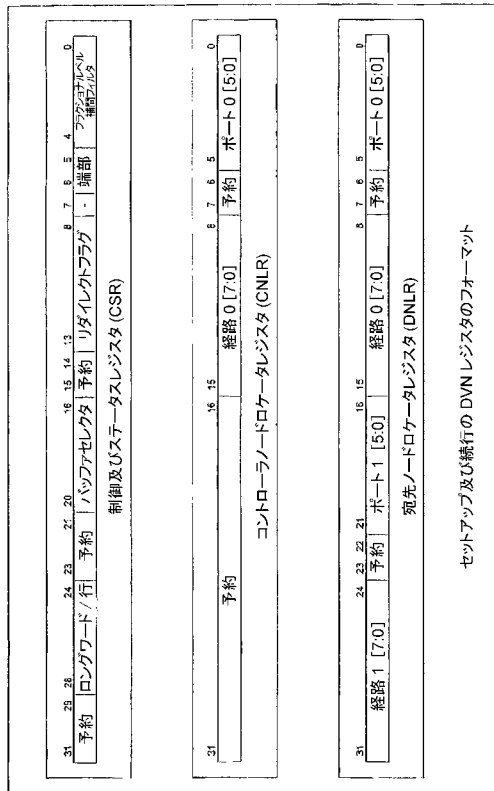
【図 17】



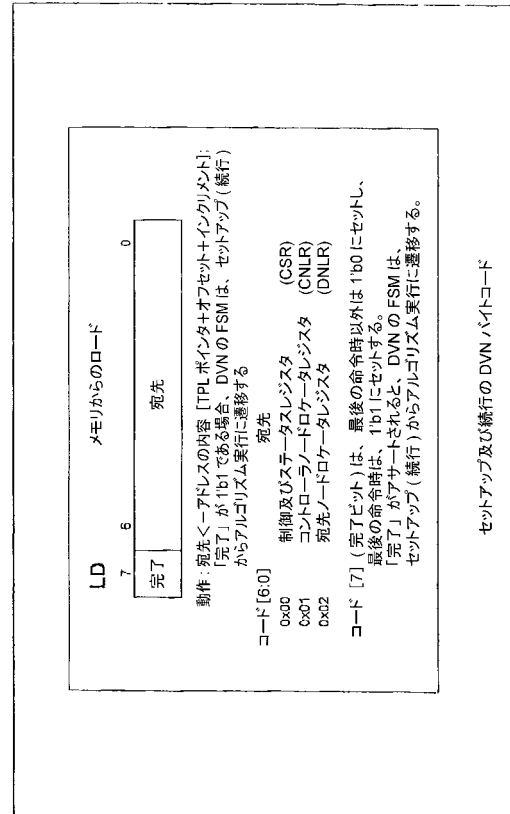
【図 18】



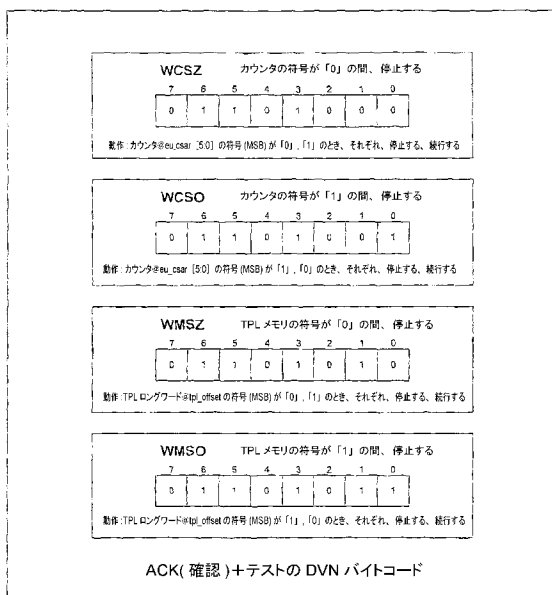
【図 19】



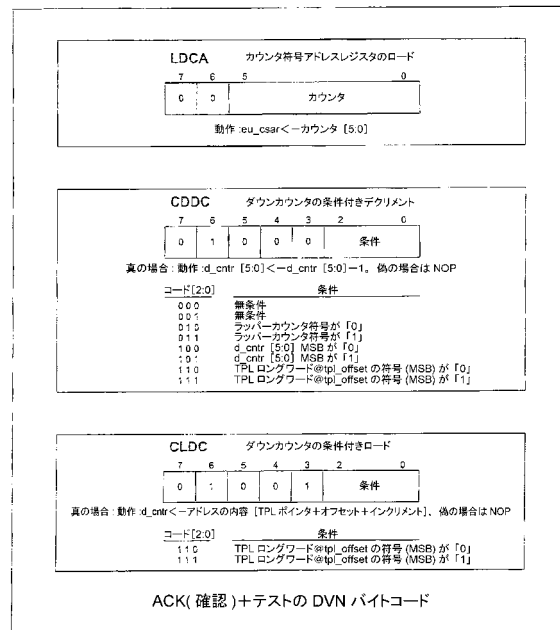
【図 20】



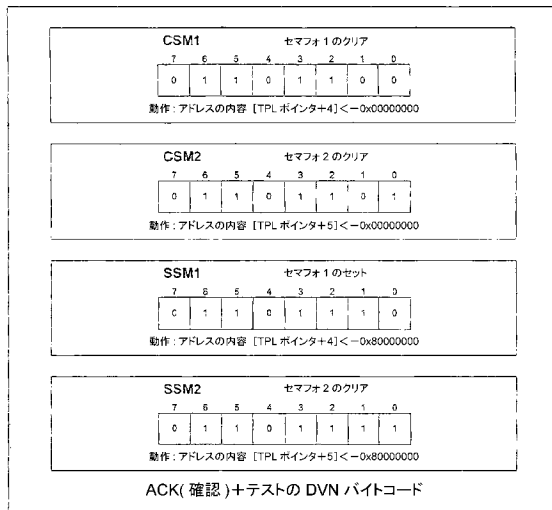
【図 21 A】



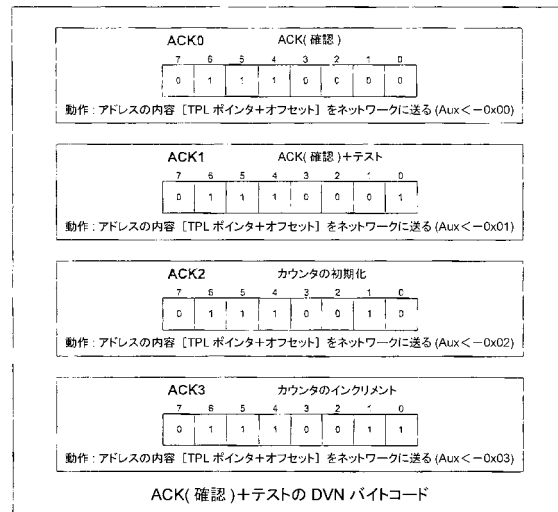
【図 21 B】



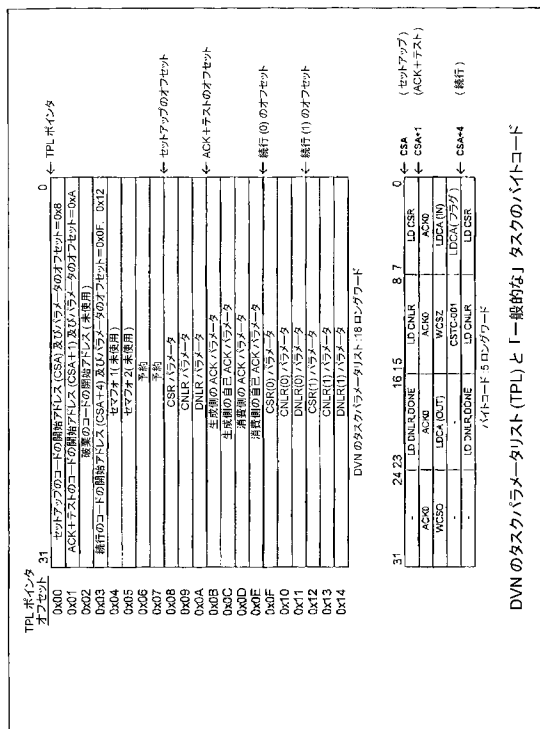
【図 2 1 D】



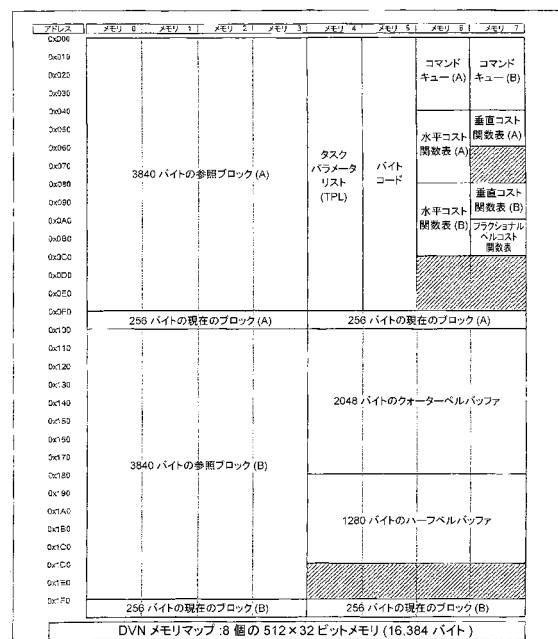
【図 2 1 E】



【図 2 2】



【図 2 3】



【図 24】

31	30	27 26										パラメータ										0										
V コマンド																						DVN コマンドバッファのフォーマット										
31	30	27 26										15 14										8 7 6 5	0									
1	0	0	0	0	予約						X 位置						-	Y 位置														
コマンド 0x0: 1つのポジション (X,Y) の指標値を求める																																
31	30	27 26										24 23	19 18										15 14	8 7 6 5	0							
1	0	0	0	1	予約						n (M)						X 位置						-	Y 位置								
コマンド 0x1: m x n セットの 5 x 5 のポジション (X,Y) の最良の指標値を求める																																
31	30	27 26										23 22	18 15										14	8 7 6 5	0							
1	0	0	1	0	予約						X 位置						-	Y 位置														
コマンド 0x2: フルベルからハーフベルへの補間																																
31	30	27 26										25 24	23	12 11											0							
1	0	0	1	1	予約						ハーフベルバッファ位置																					
コマンド 0x3: ハーフベルからクォーターベルへの補間																																
31	30	27 26										25 24	18 15										14	8 7 6 5	0							
1	0	0	0	0	予約						X 位置						-	Y 位置														
コマンド 0x4: フルベル符号付き差																																
31	30	27 26										25 24	12 11											0								
1	0	1	0	予約						ハーフベルバッファ位置																						
コマンド 0x5: ハーフベル符号付き差																																
31	30	27 26										25 24	12 11											0								
1	0	1	1	0	予約						クォーターベルバッファ位置																					
コマンド 0x6: クォーターベル符号付き差																																
31	30	27 26										25 24	18 15										14	8 7 6 5	0							
1	0	1	1	1	予約						X 位置						-	Y 位置														
コマンド 0x7: フルベルマクロブロックの出力																																
31	30	27 26										25 24	12 11											0								
1	1	0	0	予約						ハーフベルバッファ位置																						
コマンド 0x8: ハーフベルマクロブロックの出力																																
31	30	27 26										25 24	12 11											0								
1	1	0	0	予約						クォーターベルバッファ位置																						
コマンド 0x9: クォーターベルマクロブロックの出力																																
31	30	27 26										25	予約											0								
1	1	0	1	予約						コマンド 0xA: 最良指標値の出力																						
コマンド 0xA: 最良指標値の出力																																
31	30	27 26										25	予約											0								
1	1	0	1	予約						コマンド 0xB: 動きベクトルの出力																						
コマンド 0xB: 動きベクトルの出力																																
31	30	27 26										予約											0									
1	1	1	0	予約						コマンド 0xC: 完了																						
コマンド 0xC: 完了																																
31	30	27 26										任意のビットパターン											0									
1	1	1	1	予約						コマンド 0xF: エコー																						
コマンド 0xF: エコー																																

DVN コマンドバッファのデータ構造

【図 25】

0	ステータス				0	
DVN からコントローラへのステータスワードのフォーマット						
31	24 23		16 15		0	
	X 位置	Y 位置	フルベル指標値			
コマンド 0x0 のステータスワード：1 つのポジション (X, Y) の指標値を求める						
31	24 23		16 15		0	
	X 位置	Y 位置	フルベル優良指標値			
コマンド 0x1 のステータスワード：m 個のセグの 5×5 のポジション (X, Y) の数値の指標値を求める						
31	20 19		16 15		0	
	候補		指標値			
コマンド 0x2 のステータスワード：フルベルからハーフベルへの補間						
31	20 19		16 15		0	
	候補		指標値			
コマンド 0x3 のステータスワード：ハーフベルからクォーターベルへの補間						
31	0xF0190000					0
コマンド 0x4 のステータスワード：フルベル符号付き差						
31	0xFF020000					0
コマンド 0x5 のステータスワード：ハーフベル符号付き差						
31	0xFF040000					0
コマンド 0x6 のステータスワード：クォーターベル符号付き差						
31	0xFF080000					0
コマンド 0x7 のステータスワード：フルベルマクロブロックの出力						
31	0xFF100000					0
コマンド 0x8 のステータスワード：ハーフベルマクロブロックの出力						
31	0xFF200000					0
コマンド 0x9 のステータスワード：クォーターベルマクロブロックの出力						
31	15					0
	指標値					
コマンド 0xA のステータスワード：優良指標値の出力						
31	25 24	16 15		8 7	0	
	水準：0 ~ 79(3/4)				画速：0 ~ 49(3/4)	
コマンド 0xB のステータスワード：クォーターベル解像度の動きベクトルの出力						
31	0xFF400000					0
コマンド 0xC のステータスワード：完了						
31	ステータス [31:0] = コマンド [31:0]					0
コマンド 0xF のステータスワード：エラー						
31	0xFF800000					0
コマンドキューリダイレクトへの応答におけるステータスワード						
DVN からコントローラへのステータスワードのフォーマット						

【図 2 6】

[illegible]

【図 27】

a) ハーバードコスト関数

水平		0	1/2
0	A	B	
1/2	B	C	

垂直

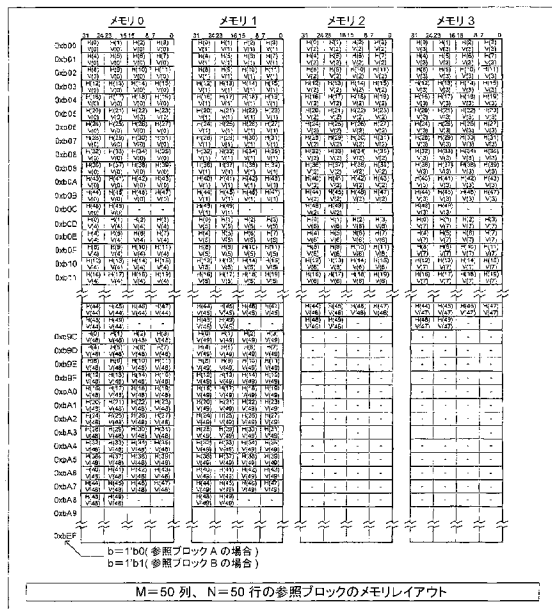
b) ケンターベルコスト関数

水平		0	1/2	1/4
0	A	B	D	
1/2	B	C	E	
1/4	D	E	F	

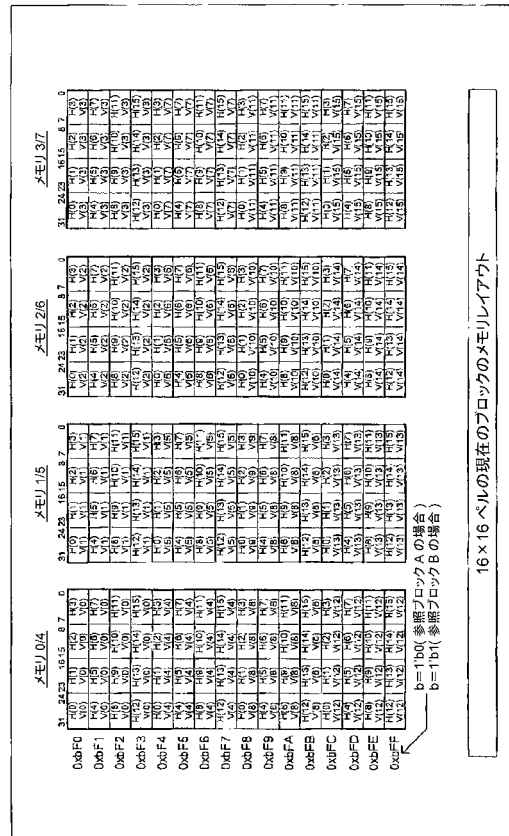
フランクショナルベルコスト関数表 A

フランクショナルベルコスト関数表 B

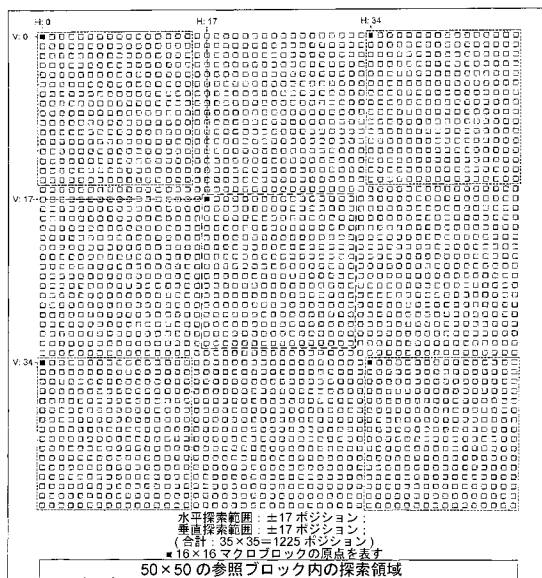
【図 28】



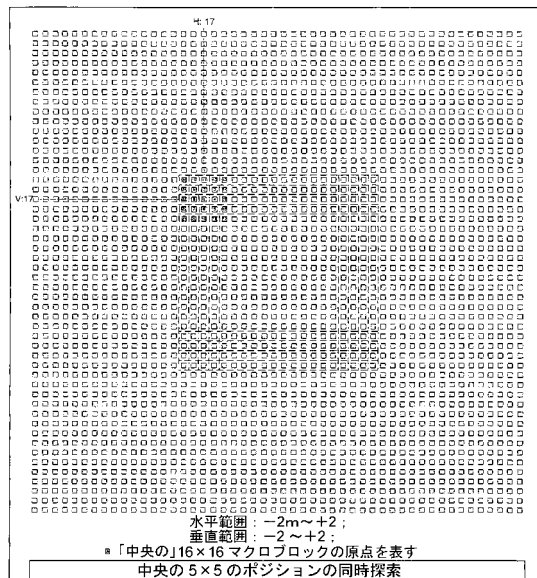
【図 29】



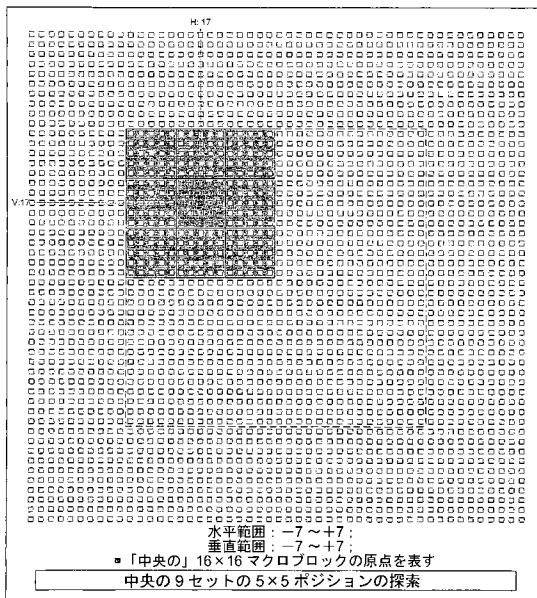
【図 30】



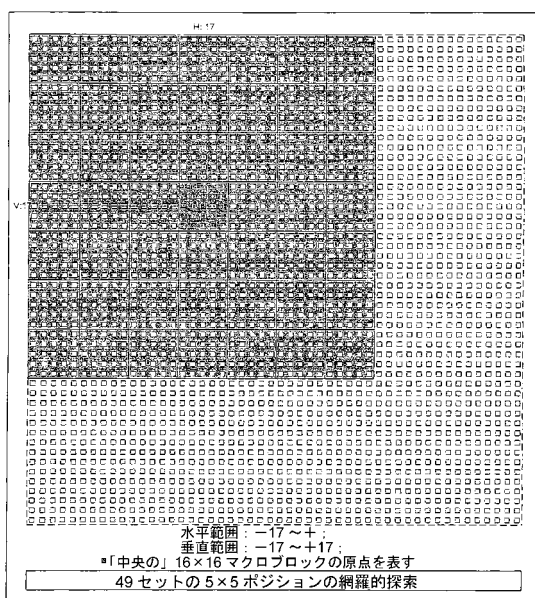
【図 3 1】



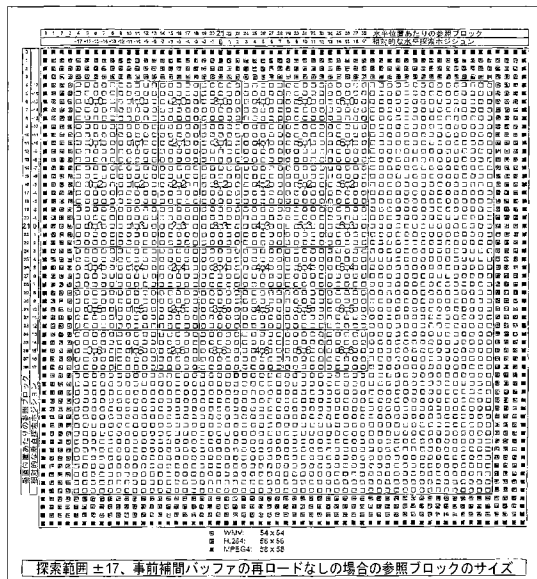
【図 3 2】



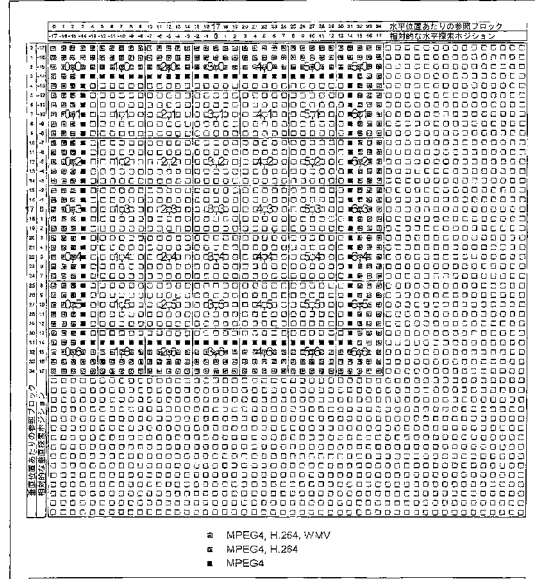
【図 3 3】



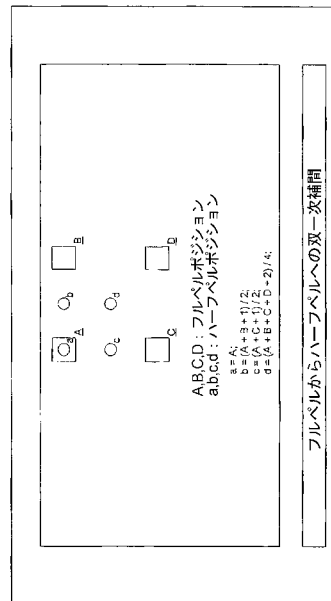
【図 3 4】



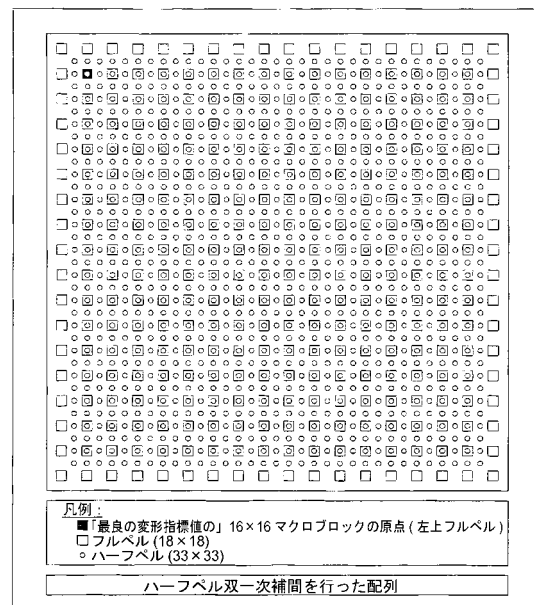
【図 3 5】



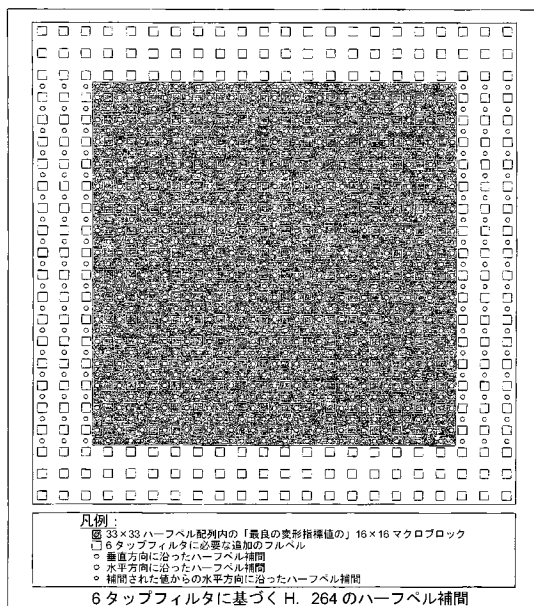
【図 36】



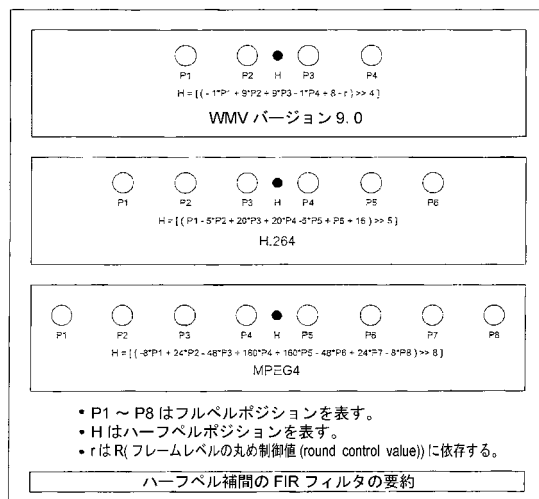
【図 37】



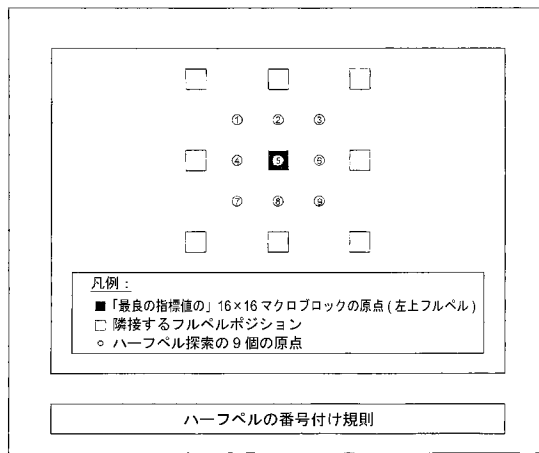
【図 38】



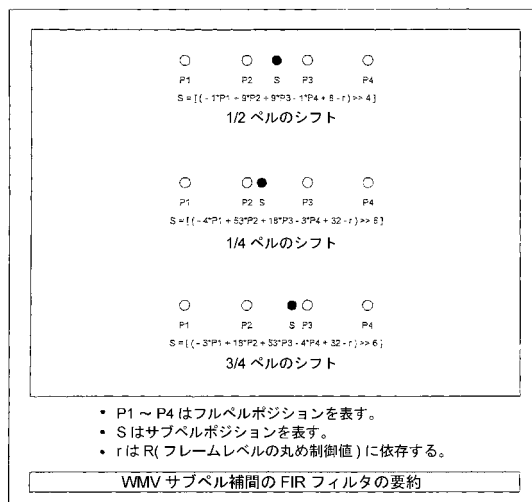
【図 39】



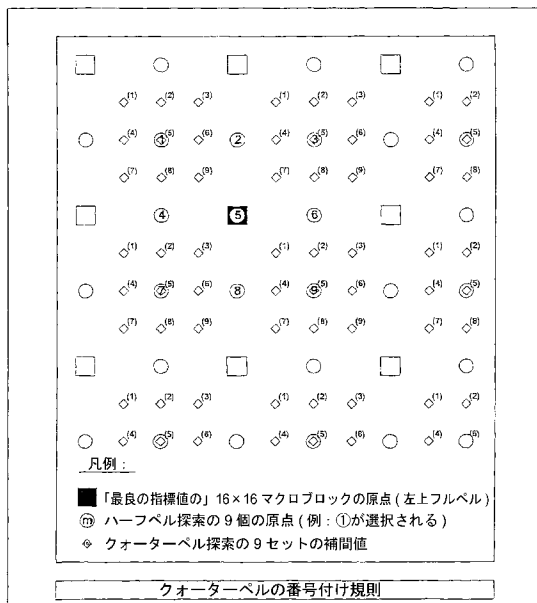
【図 40】



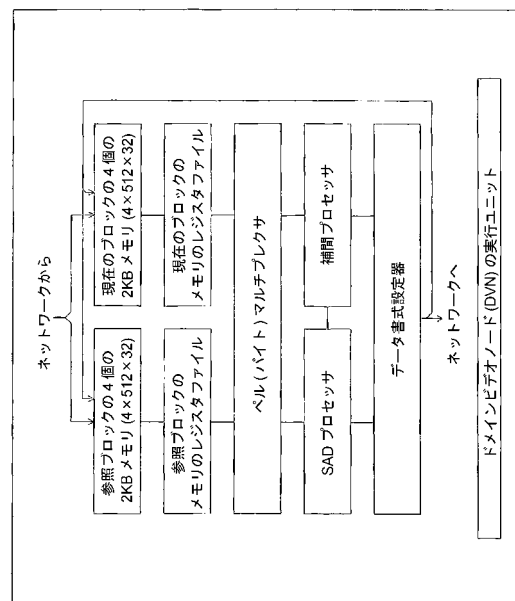
【図 41】



【図 42】



【図 43】



【図 2 1 C】

CSTC 続行への条件付き状態遷移						
7	6	5	4	3	2	0
0	1	0	1	0	条件	

真の場合：動作：FSM 状態が続行になる。偽の場合は NOP

コード[2:0]	条件
000	無条件
001	無条件 (ラッパカウンタ符号によってパラメータオフセットが選択される)
010	ラッパカウンタ符号が「0」
011	ラッパカウンタ符号が「1」
100	d_cntr [5:0] MSB が「0」
101	d_cntr [5:0] MSB が「1」
110	TPL ロングワード@tpl_offset の符号 (MSB) が「0」
111	TPL ロングワード@tpl_offset の符号 (MSB) が「1」

CSTI 条件付きでアイドルに状態遷移し、完了をセットする						
7	6	5	4	3	2	0
0	1	0	1	1	条件	

真の場合：動作：FSM 状態がアイドルになり、「eu_done」アサート。偽の場合は NOP

コード[2:0]	条件
000	無条件
001	無条件
010	ラッパカウンタ符号が「0」
011	ラッパカウンタ符号が「1」
100	d_cntr [5:0] MSB が「0」
101	d_cntr [5:0] MSB が「1」
110	TPL ロングワード@tpl_offset の符号 (MSB) が「0」
111	TPL ロングワード@tpl_offset の符号 (MSB) が「1」

ACK(確認) + テストの DVN バイトコード

フロントページの続き

(72)発明者 シューアマン, ダブリュー, ジェームズ
アメリカ合衆国, カリフォルニア州, サラトガ, サラトガ ヒルズ ロード 21485

審査官 坂本 聡生

(56)参考文献 特開2003-009149 (JP, A)
特開平06-276510 (JP, A)
特開2004-046499 (JP, A)

(58)調査した分野(Int.Cl., DB名)
H04N 7/24 - 7/68