

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2014-501009

(P2014-501009A)

(43) 公表日 平成26年1月16日(2014.1.16)

(51) Int.Cl.		F I			テーマコード (参考)
G06F 9/30 (2006.01)		G06F 9/30	350B		5B033
G06F 15/173 (2006.01)		G06F 15/173	650S		5B045

審査請求 未請求 予備審査請求 未請求 (全 30 頁)

(21) 出願番号	特願2013-540074 (P2013-540074)	(71) 出願人	390020248
(86) (22) 出願日	平成23年11月18日 (2011.11.18)		日本テキサス・インスツルメンツ株式会社
(85) 翻訳文提出日	平成25年7月16日 (2013.7.16)		東京都新宿区西新宿六丁目24番1号
(86) 国際出願番号	PCT/US2011/061487	(71) 出願人	507107291
(87) 国際公開番号	W02012/068513		テキサス インスツルメンツ インコーポ
(87) 国際公開日	平成24年5月24日 (2012.5.24)		レイテッド
(31) 優先権主張番号	61/415, 210		アメリカ合衆国 テキサス州 75265
(32) 優先日	平成22年11月18日 (2010.11.18)		-5474 ダラス メール ステイショ
(33) 優先権主張国	米国 (US)		ン 3999 ピーオーボックス 655
(31) 優先権主張番号	61/415, 205		474
(32) 優先日	平成22年11月18日 (2010.11.18)	(74) 上記1名の代理人	100098497
(33) 優先権主張国	米国 (US)		弁理士 片寄 恭三
(31) 優先権主張番号	13/232, 774		
(32) 優先日	平成23年9月14日 (2011.9.14)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 データを移動させるための方法及び装置

(57) 【要約】

計算ユニット(4308-1~4308-M、7607-1~7607-P)内の第1のレジスタファイル(4358-1~4358-8、7902)からプロセッサ(4322、7614)内の第2のレジスタファイル(5206)へデータを移動するための方法が提供される。データ移動リード(risc_is_mvvr)上の信号の状態が、計算ユニット内の第1のレジスタファイルからプロセッサ内の第2のレジスタファイルへのデータ移動命令を示すように変更される。プロセッサから計算ユニットに第1のアドレスリード(risc_is_ra)でレーンアドレスが提供される。プロセッサから計算ユニットに第2のアドレスリード(risc_is_ra)で読み出しアドレスが提供され、計算ユニット内の第1のレジスタファイルからプロセッサ内の第2のレジスタファイルにデータインタフェースリード(node_regf_rd)でデータが伝送される。

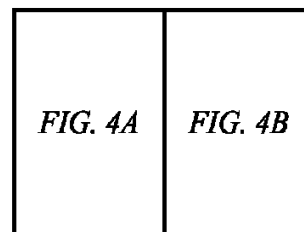


FIG. 4

【特許請求の範囲】

【請求項 1】

装置であって、

第 1 のレジスタファイル (4 3 5 8 - 1 ~ 4 3 5 8 - 8、7 9 0 2) を有する計算ユニット (4 3 0 8 - 1 ~ 4 3 0 8 - M、7 6 0 7 - 1 ~ 7 6 0 7 - P)、及び

前記計算ユニットに結合されるプロセッサ (4 3 2 2、7 6 1 4) であって、前記第 1 のレジスタファイルからのデータ移動命令 (M F V R C) を有する命令セットを含み、且つ圧縮する、前記プロセッサ、

を特徴とし、

前記プロセッサが、

第 2 のレジスタファイル (5 2 0 6) と、

前記第 1 のレジスタファイルのための書き込みアドレスを示すためのアドレスリード (r i s c _ i s _ r a) と、

データを伝送するためのデータインタフェースリード (n o d e _ r e g f _ r d) と

、
データ移動リード (r i s c _ i s _ m f v r e) であって、前記データ移動リード上の信号の状態が変更されるとき、前記第 1 のレジスタファイルから前記第 2 のレジスタファイルへの前記データ移動命令を示すための、前記データ移動リードと、

を含む、装置。

【請求項 2】

請求項 1 に記載の装置であって、前記アドレスリード (r i s c _ i s _ r a) が複数の第 2 のアドレスリード (r i s c _ i s _ r a) を更に特徴とする、装置。

【請求項 3】

請求項 2 に記載の装置であって、前記複数の第 2 のアドレスリード (r i s c _ i s _ r a) が 5 ビット幅である、装置。

【請求項 4】

請求項 1、2、又は 3 に記載の装置であって、前記プロセッサが、上位ハーフ書き込み、下位ハーフ書き込み、フル書き込み、又は読み出し、のいずれを実行するかを示すためのハーフワードリード (r i s c _ i s _ h w z) を含む装置。

【請求項 5】

請求項 1、2、3、又は 4 に記載の装置であって、前記ハーフワードリード (r i s c _ i s _ h w z) が複数のハーフワードリード (r i s c _ i s _ h w z) を更に特徴とする装置。

【請求項 6】

請求項 5 に記載の装置であって、前記複数のハーフワードリードが 2 ビット幅である装置。

【請求項 7】

請求項 1、2、3、4、5、又は 6 に記載の装置であって、前記データインタフェースリード (n o d e _ r e g f _ r d) が、複数のデータインタフェースリード (n o d e _ r e g f _ r d) を更に特徴とする装置。

【請求項 8】

請求項 1、2、3、4、5、6、又は 7 に記載の装置であって、前記計算ユニットが、複数の単一入力多重データ (S I M D) 機能ユニット (4 3 0 8 - 1 ~ 4 3 0 8 - M) を更に特徴とする装置。

【請求項 9】

請求項 1、2、3、4、5、6、又は 7 に記載の装置であって、前記計算ユニットが、複数のベクトルユニット (7 6 0 7 - 1 ~ 7 6 0 7 - P) を更に特徴とする装置。

【請求項 10】

方法であって、

計算ユニット (4 3 0 8 - 1 ~ 4 3 0 8 - M、7 6 0 7 - 1 ~ 7 6 0 7 - P) 内の第 1

10

20

30

40

50

のレジスタファイル(4358-1~4358-8、7902)からプロセッサ(4322、7614)内の第2のレジスタファイル(5206)へのデータ移動命令(MFVRC)を示し、且つ前記第2のレジスタファイルへ圧縮するように、データ移動リード(risc__is__mfvre)上の信号の状態を変更することと、

前記プロセッサから前記計算ユニットにアドレスリード(risc__is__ra)で書き込みアドレスを提供することと、

前記計算ユニット内の前記第1のレジスタファイルから前記プロセッサ内の前記第2のレジスタファイルにデータインタフェースリード(node__regf__rd)でデータを伝送することと、

を特徴とする方法。

10

【請求項11】

請求項10に記載の方法であって、前記アドレスリード(risc__is__ra)が複数の第2のアドレスリード(risc__is__ra)を更に特徴とする方法。

【請求項12】

請求項10又は11に記載の方法であって、ハーフワードリード(risc__is__hwz)で、上位ハーフ書き込み、下位ハーフ書き込み、フル書き込み、又は読み出しのいずれを実行するかを示すことを更に特徴とする方法。

【請求項13】

請求項10、11、又は12に記載の方法であって、前記ハーフワードリード(risc__is__hwz)が、複数のハーフワードリード(risc__is__hwz)を更に特徴とする方法。

20

【請求項14】

請求項10、11、12、又は13に記載の方法であって、前記データインタフェースリード(node__regf__rd)が複数のデータインタフェースリード(node__regf__rd)を更に特徴とする方法。

【請求項15】

システムであって、

計算ユニット(4308-1~4308-M、7607-1~7607-P)内の第1のレジスタファイル(4358-1~4358-8、7902)からプロセッサ(4322、7614)内の第2のレジスタファイル(5206)へのデータ移動命令(MFVRC)を示し、且つ前記第2のレジスタファイルへ圧縮するように、データ移動リード(risc__is__mfvre)上の信号の状態を変更するための手段と、

30

前記プロセッサから前記計算ユニットにアドレスリード(risc__is__ra)で書き込みアドレスを提供するための手段と、

前記計算ユニット内の前記第1のレジスタファイルから前記プロセッサ内の前記第2のレジスタファイルにデータインタフェースリード(node__regf__rd)でデータを伝送するための手段と、

を特徴とするシステム。

【請求項16】

請求項15に記載のシステムであって、前記アドレスリード(risc__is__ra)が複数の第2のアドレスリード(risc__is__ra)を更に特徴とするシステム。

40

【請求項17】

請求項15又は16に記載のシステムであって、ハーフワードリード(risc__is__hwz)上で、上位ハーフ書き込み、下位ハーフ書き込み、フル書き込み、又は読み出しのいずれを実行するかを示すための手段を更に特徴とするシステム。

【請求項18】

請求項15、16、又は17に記載のシステムであって、前記ハーフワードリード(risc__is__hwz)が、複数のハーフワードリード(risc__is__hwz)を更に特徴とするシステム。

【請求項19】

50

請求項 15、16、17、又は 18 に記載のシステムであって、前記データインタフェースリード (node__regf__rd) が、複数のデータインタフェースリード (node__regf__rd) を更に特徴とするシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、全般的にプロセッサに関し、より具体的には処理クラスタに関する。

【背景技術】

【0002】

図 1 はマルチコアシステム (2 ~ 16 コアの範囲) についての実行速度のスピードアップ対並列オーバーヘッドを示すグラフである。スピードアップとは、単一プロセッサの実行時間を並列プロセッサの実行時間で除したものである。図からわかるように、多数のコアから有意な利益を得るために、並列オーバーヘッドはゼロに近くなければならない。しかし並列プログラム間に何らかの相互作用が存在する場合、オーバーヘッドは極めて高くなる傾向があるため、完全に分離されたプログラムでなければ 2 又は 3 以上のプロセッサを効率的に使用するのは通常極めて難しい。従って、改善された処理クラスタが必要とされている。

10

【発明の概要】

【0003】

従って、本開示の実施形態は方法を提供する。その方法は下記を特徴とする。すなわち、計算ユニット (4308 - 1 ~ 4308 - M、7607 - 1 ~ 7607 - P) 内の第 1 のレジスタファイル (4358 - 1 ~ 4358 - 8、7902) からプロセッサ (4322、7614) 内の第 2 のレジスタファイル (5206) へのデータ移動命令を示すように、データ移動リード (risc__is__mfvvr) 上の信号の状態を変更することと、プロセッサ (4322、7614) から計算ユニット (4308 - 1 ~ 4308 - M、7607 - 1 ~ 7607 - P) に第 1 のアドレスリード (risc__is__ua) でレーンアドレスを提供することと、プロセッサ (4322、7614) から計算ユニット (4308 - 1 ~ 4308 - M、7607 - 1 ~ 7607 - P) に第 2 のアドレスリード (risc__is__ra) で読み出しアドレスを提供することと、計算ユニット (4308 - 1 ~ 4308 - M、7607 - 1 ~ 7607 - P) 内の第 1 のレジスタファイル (4358 - 1 ~ 4358 - 8、7902) からプロセッサ (4322、7614) 内の第 2 のレジスタファイル (5206) にデータインタフェースリード (node__regf__rd) でデータを伝送すること、である。

20

30

【図面の簡単な説明】

【0004】

【図 1】マルチコアのスピードアップパラメータのグラフである。

【0005】

【図 2】本開示の実施形態に従ったシステムの図である。

【0006】

【図 3】本開示の実施形態に従った SOC の図である。

40

【0007】

【図 4 A】本開示の実施形態に従った並列処理クラスタの図である。

【図 4 B】本開示の実施形態に従った並列処理クラスタの図である。

【0008】

【図 5】処理クラスタ内のノード又は計算要素の一部分の図である。

【図 6 A】処理クラスタ内のノード又は計算要素の一部分の図である。

【図 6 B】処理クラスタ内のノード又は計算要素の一部分の図である。

【0009】

【図 7】共有機能メモリのブロック図である。

【0010】

50

【図 8】共有機能メモリのための SIMD データパスの図である。

【0011】

【図 9】1つの SIMD データパスの一部分の図である。

【0012】

【図 10】ノードプロセッサ又は RISC プロセッサのより詳細な図である。

【0013】

【図 11】ノードプロセッサ又は RISC プロセッサのためのパイプラインの一部分の例の図である。

【図 12】ノードプロセッサ又は RISC プロセッサのためのパイプラインの一部分の例の図である。

10

【発明を実施するための形態】

【0014】

図 2 では、並列処理を実行する SOC 用アプリケーションの例が見られる。この例では、撮像デバイス 1250 が示される。この（例えば携帯電話又はカメラであり得る）撮像デバイス 1250 は、概して、画像センサ 1252、SOC 1300、ダイナミックランダムアクセスメモリ（DRAM）1315、フラッシュメモリ 1314、ディスプレイ 1254、及び電力管理集積回路（PMIC）1256 を含む。動作では、画像センサ 1252 は、（静止画像又はビデオであり得る）画像情報を捕捉することができ、この画像情報は SOC 1300 及び DRAM 1315 によって処理され得、不揮発性メモリ（即ち、フラッシュメモリ 1314）に保存され得る。また、フラッシュメモリ 1314 に保存される画像情報は、SOC 1300 及び DRAM 1315 の使用によって、ディスプレイ 1254 上で使用するために表示され得る。また、撮像デバイス 1250 は、可搬型であることが多く、電源としてバッテリーを含む。（SOC 1300 によって制御され得る）PMIC 1256 は、バッテリー寿命を長持ちさせるために電力使用量の調整を補助し得る。

20

【0015】

図 3 では、本開示の実施形態に従ったシステムオンチップ又は SOC 1300 の例が図示されている。この（典型的には、OMAP（登録商標）等の集積回路又は IC である）SOC 1300 は、（概して上述の並列処理を実行する）処理クラスタ 1400、及び、（上で説明及び参照された）ホスト環境を提供するホストプロセッサ 1316 を概して含む。ホストプロセッサ 1316 は、ワイド（即ち、32ビット、64ビット等）RISC プロセッサ（例えば ARM Cortex-A9 等）であり得、バスアービトラータ 1310、バッファ 1306、（ホストプロセッサ 1316 がインタフェースバス又は Iバス 1330 上で周辺インタフェース 1324 にアクセスすることを許可する）バスブリッジ 1320、ハードウェアアプリケーションプログラミングインタフェース（API）1308、及び割り込みコントローラ 1322 と、ホストプロセッサバス又は HPバス 1328 上で通信する。処理クラスタ 1400 は、典型的に、（例えば、荷電結合デバイス、又は CCD インタフェースであり得、オフチップデバイスと通信し得る）機能回路要素 1302、バッファ 1306、バスアービトラータ 1310、及び周辺インタフェース 1324 と、処理クラスタバス又は PCバス 1326 上で、通信する。この構成を用いて、ホストプロセッサ 1316 は、API 1308 を介して情報を提供する（即ち、所望の並列実装に適合するように処理クラスタ 1400 を構成する）ことができ、一方、処理クラスタ 1400 及びホストプロセッサ 1316 はいずれも、（フラッシュインタフェース 1312 を介して）フラッシュメモリ 1314 に、（メモリコントローラ 1304 を介して）DRAM 1315 に、直接アクセスできる。また、Joint Test Action Group（JTAG）インタフェース 1318 を介して、テスト及びバウンダリスキャンが実行され得る。

30

40

【0016】

図 4 を参照すると、本開示の実施形態に従った並列処理クラスタ 1400 の例が示されている。典型的には、処理クラスタ 1400 はハードウェア 722 に対応する。処理クラスタ 1400 は、概して、パーティション 1402-1 ~ 1402-R を含む。これらは

50

、ノード 808 - 1 ~ 808 - N、ノードラッパー 810 - 1 ~ 810 - N、命令メモリ 1404 - 1 ~ 1404 - R、及び（以下で詳しく説明する）バスインタフェースユニット又は（BIU）4710 - 1 ~ 4710 - Rを含む。ノード 808 - 1 ~ 808 - Nは、各々データインターコネクト 814 に（各々の BIU 4710 - 1 ~ 4710 - R 及びデータバス 1422 を介して）結合され、パーティション 1402 - 1 ~ 1402 - R のための制御及びメッセージが制御ノード 1406 からメッセージ 1420 を介して提供される。また、グローバルロード/ストア（GLS）ユニット 1408 及び共有機能メモリ 1410 は、（後述のように）データ移動のための付加的な機能を提供する。それに加えて、レベル 3 又は L3 キャッシュ 1412、（概して、IC 内には含まれない）周辺装置 1414、（典型的にはフラッシュメモリ 1314 及び/又は DRAM 1315、並びに SOC 1300 内には含まれないその他のメモリである）メモリ 1416、及びハードウェアアクセラレータ（HWA）ユニット 1418 が処理クラスタ 1400 と共に用いられる。また、データ及びアドレスを制御ノード 1406 に通信するように、インタフェース 1405 が提供される。

10

20

30

40

50

【0017】

処理クラスタ 1400 は、概して、データ伝送のために「プッシュ」モデルを使用する。データ伝送は要求応答型のアクセスではなく、概してポステッドライトとして現れる。これは、データ伝送が一方方向であるため要求応答アクセスに比べてグローバルインターコネクト（即ち、データインターコネクト 814）の占有を 2 分の 1 に減らすという利点を有する。概して、インターコネクト 814 を介して要求をルーティングし、その後、応答が要求元へルーティングされ、その結果インターコネクト 814 上で 2 つの遷移が生成されることは望まれない。プッシュモデルは単一伝送を生成する。これは、ネットワークサイズが増大するとネットワークレイテンシが増大するため、またこのことが要求応答型トランザクションのパフォーマンスを低下させることは避けられないことであるため、スケラビリティに関して重要である。

【0018】

プッシュモデルは、データフロープロトコル（即ち、812 - 1 ~ 812 - N）と同様に、グローバルデータトラフィックを、正確さのために用いられるものまで概して最小化する一方、ローカルノードの利用率に対するグローバルデータフローの影響も概して最小化する。大量のグローバルトラフィックであってもノード（即ち、808 - i）のパフォーマンスに対する影響は、通常、皆無に近い。ソースはデータを（後述する）グローバル出力バッファに書き込み、伝送成功の確認を要求することなく継続する。データフロープロトコル（即ち、812 - 1 ~ 812 - N）は、概して、インターコネクト 814 で単一伝送を用い、データをあて先へ移動する最初の試みでの伝送が成功することを確実にする。（後述する）グローバル出力バッファは（例えば）最大 16 出力まで保持することができるため、出力のための瞬時グローバル帯域幅が不十分になることに起因するノード（即ち、808 - i）のストールの可能性が非常に低くなる。更に、瞬時帯域幅は、要求応答トランザクション又は伝送失敗の繰り返しによる影響を受けない。

【0019】

最後に、プッシュモデルはプログラミングモデルに一層密接に適合する。言い換えるとプログラムは自己データを「フェッチ」せずに、その代わりに、プログラムの入力変数及び/又はパラメータは呼び出される前に書き込まれる。プログラミング環境では、入力変数の初期化は、ソースプログラムによるメモリへの書き込みとして行われる。処理クラスタ 1400 内では、これらの書き込みがポステッドライトに変換され、変数の値をノードコンテキストにポピュレートさせる。

【0020】

（後述する）グローバル入力バッファは、ソースノードからデータを受け取るために用いられる。各ノード 808 - 1 ~ 808 - N のためのデータメモリが単一ポートであるため、入力データの書き込みが、ローカルの単一入力多重データ（SIMD）による読み出しとコンフリクトすることがあり得る。入力データをグローバル入力バッファへ受け入れ

、そこで入力データが空きのデータメモリサイクルを待つことができることによって、この競合は回避される（即ち、SIMDアクセスとのバンクコンフリクトはない）。データメモリは、（例えば）32バンクを有し得るため、直ちにバッファがフリーになる可能性が非常に高い。しかしながら、伝送を確認するためのハンドシェイキングがないので、ノード（即ち、808-i）はフリーのバッファエントリを持つはずである。所望とされる場合は、グローバル入力バッファは、バッファ位置をフリーにするために、ローカルノード（即ち、808-i）をストールさせてデータメモリに強制的に書き込みを行うことができるが、このイベントは極めて希であるべきである。典型的には、グローバル入力バッファは2つの別々のランダムアクセスメモリ（RAM）として実装されて、一方がデータメモリへ読み出されるべき状態にある間、他方がグローバルデータを書き込むための状態になり得るようにする。メッセージングインターコネクトは、グローバルデータインターコネクトとは分かれているが、同様にプッシュモデルを使用する。

10

【0021】

システムレベルでは、所望のスループットにスケーリングされた多数のノードを備えるSMP又は対称型多重処理のように、ノード808-1~808-Nが処理クラスタ1400内で複製される。処理クラスタ1400は極めて多数のノードにまでスケーリングし得る。ノード808-1~808-Nはパーティション1402-1~1402-Rにグループ分けされ、各パーティションは1つ又は複数のノードを有する。パーティション1402-1~1402-Rは、ノード間のローカル通信を増大させることによって及びより大きなプログラムで一層大量の出力データを計算させることによってスケラビリティを促進し、その結果、所望のスループット要件を達成する可能性を更に高める。パーティション（即ち、1402-i）内では、ノードはローカルインターコネクトを用いて通信し、グローバルリソースを必要としない。また、パーティション（即ち、1404-i）内のノードは、排他的命令メモリを用いる各ノードから共通命令メモリを用いる全てのノードまで、任意の粒度で、命令メモリ（即ち、1404-i）を共有することができる。例えば、3つのノードが命令メモリの3つのバンクを共有し、第4のノードが命令メモリの排他的バンクを有することができる。ノードが命令メモリ（即ち、1404-i）を共有するとき、それらのノードは、概して、同じプログラムを同期的に実行する。

20

【0022】

また、処理クラスタ1400は非常に多数のノード（即ち、808-i）及びパーティション（即ち、1402-i）をサポートし得る。しかしながら、1つのパーティションについて4以上のノードを持つと概してノンユニフォームメモリアccess（NUMA）アーキテクチャに類似するため、パーティション毎のノードの数は通常は4つに限定されている。この例では、パーティションは、（後でインターコネクト814に関連して説明する）1つ（又は複数）のクロスバーを介して接続される。クロスバーは概して横断帯域幅が一定している。処理クラスタ1400は、現在、サイクル毎に1ノード幅のデータ（例えば、64、16ビットピクセル）を伝送するように設計されており、4サイクルに亘り、1サイクルにつき16ピクセルの4伝送に区分される。処理クラスタ1400は、概して、レイテンシトレラントであり、インターコネクト814がほぼ飽和（この状態を達成するのは合成プログラム以外では極めて難しいことに留意されたい）であっても、ノードバッファリングが、概して、ノードストールを防止する。

30

40

【0023】

典型的には、処理クラスタ1400はパーティション間で共有する下記のグローバルリソースを含む。

（1） 制御ノード1406。これは（メッセージバス1420で）システムワイドのメッセージングインターコネクト、イベント処理及びスケジューリング、及びホストプロセッサ及びデバッグ（これらは全て後で詳しく説明する）へのインタフェースを提供する。

（2） GLSユニット1408。これはプログラマブル縮小命令セット（RISC）プロセッサを含み、システムデータ移動を可能にする。システムデータ移動は、GLSデ

50

ータ移動スレッドとして直接コンパイルされ得るC++プログラムによって記述され得る。これによって、ソースコードを修正することなく、クロスホスト環境でのシステムコードの実行が可能になり、また、システム又は(後述する)SIMDデータメモリ内の任意のアドレス(変数)のセットから別の任意のアドレス(変数)のセットに移動できるため、ダイレクトメモリアクセスよりもより一般的である。GLSユニット1408は、(例えば)0-サイクルのコンテキストスイッチを備え、マルチスレッド化され、例えば、最大16スレッドまでサポートする。

(3) 共有機能メモリ1410。これは、一般のルックアップテーブル(LUT)及び統計収集機能(ヒストグラム)を提供する大型共有メモリである。また、これは大型共有メモリを使用して、リサンプリング及び歪補正等のノードSIMDにより(コストの理由で)充分サポートされていないピクセル処理をサポートし得る。この処理はネイティブタイプとして、スカラ、ベクトル、及び2Dアレイを実装する(例えば)6発行命令RISCプロセッサ(即ち、後で詳しく説明するSFMプロセッサ7614)を用いる。

(4) ハードウェアアクセラレータ1418。これは、プログラマビリティを必要としない機能のため、或いは電力及び/又は面積を最適化するために組み込まれ得る。アクセラレータは、サブシステムにはシステム内の他のノードとして現れ、制御及びデータフローに参加し、イベントを作成可能であり、スケジューリング可能である。またデバッガにとっては可視的である。(ハードウェアアクセラレータは、適用可能であるときは、専用のLUT及び統計収集を有し得る。)

(5) データインターコネクト814及びシステムオープンコアプロトコル(OCP)L3接続1412。これらは、ノードパーティション、ハードウェアアクセラレータ、及びシステムメモリ、及び、データバス1422上の周辺装置の間のデータ移動を管理する。(ハードウェアアクセラレータは、L3へのプライベート接続も有し得る。)

(6) デバッグインタフェース。これらは、図には示されていないが、本明細書中に記載される。

【0024】

図5を参照すると、ノード808-iの例の更なる詳細が見られる。ノード808-iは、処理クラスタ1400内の計算要素であり、アドレス指定及びプログラムフロー制御のための基本要素はRISCプロセッサ又はノードプロセッサ4322である。典型的には、このノードプロセッサ4322は、(40ビット命令内の20ビットイミディエート(immediate)フィールドの可能性のある)20ビット命令を備える、32ビットのデータバスを有することができる。ピクセル操作は、例えば32ピクセル機能ユニットのセットで、SIMD構成で、SIMDレジスタとSIMDデータメモリとの間で(例えば)4つのロードと(例えば)2つのストアを用いて並列に実行される(ノードプロセッサ4322の命令セットは以下のセクション7で説明する)。命令パケットは、すべてのSIMD機能ユニット4308-1~4308-Mによって実行される3発行SIMD命令と並列に、(例えば)1つのRISCプロセッサコア命令、4つのSIMDロード、及び2つのSIMDストアを記述する。

【0025】

典型的には、(ロードストアユニット4318-iからの)ロード及びストアは、SIMDデータメモリ位置と、例えば、最大64、16ビットピクセルまで表すことができる、SIMDローカルレジスタとの間でデータを移動する。SIMDロード及びストアは間接アドレス指定(直接アドレス指定もサポートされている)に共有レジスタ4320-iを用いるが、SIMDアドレス指定処理はこれらのレジスタを読み出し、アドレス指定コンテキストはコア4320によって管理される。コア4320は、レジスタのスピル/フィル、アドレス指定コンテキスト、及び入力パラメータのためのローカルメモリ4328を有する。ノード毎にパーティション命令メモリ1404-iが提供され、そこでは、多数のノードに及ぶデータセット上で、より大きなプログラムを実行するために、多数のノードがパーティション命令メモリ1404-iを共有することも可能である。

【0026】

10

20

30

40

50

また、ノード808-iは、並列処理をサポートするための幾つかの機能を組み込む。(Lf及びRtバッファ4314-i及び4312-iに関連し、概してノード808-iのための入力/出力(I/O)回路要素を含む)グローバル入力バッファ4316-i及びグローバル出力バッファ4310-iは、ノード808-i入力及び出力を命令実行から切り離し、システムIOに起因してノードがストールする可能性を極めて低くする。入力は、通常、(SIMDデータメモリ4306-1~4306-M及び機能ユニット4308-1~4308-Mによる)処理よりも、充分前に受け取られ、空きサイクルを用いてSIMDデータメモリ4306-1~4306-M内に保存される(これらは非常に一般的である)。SIMD出力データは、グローバル出力バッファ4210-iに書き込まれ、そこから処理クラスタ1400を介してルーティングされ、たとえ、システムのパフォーマンスがその限界に近づいた場合(これも可能性が低い)でも、ノード(即ち、808-i)がストールする可能性を低くする。SIMDデータメモリ4308-1~4306-M及び対応するSIMD機能ユニット4306-1~4306-Mは、各々、集散的に「SIMDユニット」と称される。

10

20

30

40

50

【0027】

SIMDデータメモリ4306-1~4306-Mは、重複しないコンテキスト内に構成され、可変サイズであり、関連又は非関連タスクのいずれかへ割り振られる。コンテキストは、水平及び垂直の両方向で十分に共有可能である。水平方向での共有はリードオンリーメモリ4330-i及び4332-iを使用し、それらは、典型的には、プログラムについてはリードオンリーであるが、書き込みバッファ4302-i及び4304-i、ロード/ストア(LS)ユニット4318-i、又は他のハードウェアによって書き込み可能である。また、これらのメモリ4330-i及び4332-iのサイズは、約512×2ビットである。概してこれらのメモリ4330-i及び4332-iはその上で操作される中央ピクセル位置に対して、左方向及び右方向へのピクセル位置に対応する。これらのメモリ4330-i及び4332-iは、書き込みをスケジューリングするために、書き込み-バッファリング機構(即ち、書き込みバッファ4302-i及び4304-i)を使用し、そこでは、サイド-コンテキスト書き込みは、通常、ローカルアクセスとは同期されていない。バッファ4302-iは、概して、同時に動作する(例えば)隣接するピクセルコンテキストとのコヒーレンスを維持する。垂直方向の共有はSIMDデータメモリ4306-1~4306-M内のサーキュラーバッファを用いる。サーキュラーアドレス指定は、LSユニット4318-iによって適用されるロード及びストア命令によってサポートされているモードである。共有データは、概して、上述のシステムレベル依存性プロトコルを用いてコヒーレントに保たれる。

【0028】

コンテキスト割り振り及び共有は、SIMDデータメモリ4306-1~4306-Mコンテキスト記述子によって、ノードプロセッサ4322に関連付けられるコンテキスト状態メモリ4326内に特定される。このメモリ4326は、例えば、16×16×32ビット又は2×16×256ビットRAMであり得る。また、これらの記述子は、コンテキスト間でデータがどのように共有されるかを、十分に一般的な方式で特定し、コンテキスト間のデータ依存性を取り扱うための情報を保持する。コンテキスト保存/復元メモリ4324は、レジスタ4320-iを並列に保存及び復元させることによって、(後で説明する)0-サイクルタスク切り替えをサポートするように使用される。SIMDデータメモリ4306-1~4306-M、及びプロセッサデータメモリ4328コンテキストは、各々のタスクのための非依存コンテキストエリアを用いて保存される。

【0029】

SIMDデータメモリ4306-1~4306-M、及びプロセッサデータメモリ4328は、可変サイズの可変数コンテキストに区分される。垂直フレーム方向のデータは、そのコンテキスト自体の中で保持及び再使用される。水平フレーム方向のデータは、コンテキストを共に水平グループにリンクさせることによって共有される。なお、コンテキスト構成は、計算に関係するノード数及びそれらが互いにどのように相関するかとはほぼ無

関係であることに留意することが重要である。コンテキストの主目的は、画像データを、このデータを操作するノードの構成に関係なく、保持、共有、及び再使用することである。

【0030】

典型的には、SIMDデータメモリ4306-1~4306-Mは、機能ユニット4308-1~4308-Mによって操作される（例えば）ピクセル及び中間コンテキストを含む。SIMDデータメモリ4306-1~4306-Mは、概して、（例えば）最大16の分離コンテキストエリアに区分される。各分離コンテキストエリアは、プログラマブルベースアドレスを備え、コンパイラによってレジスタのスピル/フィルに使用される全てのコンテキストからアクセス可能な共通エリアを備える。プロセッサデータメモリ4328は、入力パラメータ、アドレス指定コンテキスト、及びレジスタ4320-iのためのスピル/フィルエリアを含む。プロセッサデータメモリ4328は、各々プログラマブルベースアドレスを備える、SIMDデータメモリ4306-1~4306-Mコンテキストに対応する（例えば）最大16の分離ローカルコンテキストエリアを有し得る。

10

【0031】

典型的には、ノード（即ち、ノード808-i）は、8個のSIMDレジスタ（第1の構成）、32個のSIMDレジスタ（第2の構成）、及び32個のSIMDレジスタと、より小さい機能ユニットの各々に3つの予備実行ユニット（第3の構成）の例えば3つの構成を有する。

20

【0032】

例として、図6では、SIMDユニット（即ち、SIMDデータメモリ4306-1及びSIMD機能ユニット4308-1）、ノードプロセッサ4322、及びLSユニット4318-iの例がより詳しく示されている。この例に示されるように、SIMD機能ユニット4308-iは、概して、8個のより小さい機能ユニット4338-1~4338-8で構成され、第3の構成を用いている。

【0033】

先ず、プロセッサコアを見ると、ノードプロセッサ4322は、概して、全ての制御関連命令を実行し、レジスタファイル4340及び4342（各々）に示されるSIMDユニットのための全てのアドレスレジスタ値及び特殊レジスタ値を保持する。（例えば）最大6個のメモリ命令が1サイクルで計算され得る。アドレスレジスタ値の場合、示されたSIMDユニットからノードプロセッサ4322にアドレスソースオペランドが送られ、ノードプロセッサ4322がレジスタ値を送り返し、次にそのレジスタ値がSIMDユニットによってアドレス計算のために使用される。同様に、特殊レジスタ値の場合、示されたSIMDユニットからノードプロセッサ4322に特殊レジスタソースオペランドが送られ、ノードプロセッサ4322はレジスタ値を送り返す。

30

【0034】

ノードプロセッサ4322は、SIMDのための（例えば）15個の読み出しポート及び6個の書き込みポートを有し得る。典型的には、15個の読み出しポートは、6個のメモリ命令の各々のための2つのオペランド（即ち、ls src及びls src 2）を収容する（例えば）12個の読み出しポート、及び特殊レジスタファイル4312のための3つのポートを含む。典型的には、特殊レジスタファイル4342は、RCLIPMIN及びRCLIPMAXという名称の2つのレジスタを含み、これらのレジスタは共に提供されるべきものであって、概して、16エン트리レジスタファイル4342の下位の4つのレジスタに限定される。次にRCLIPMAX及びRCLIPMINレジスタは、命令の中に直接特定される。他の特殊レジスタRND及びSCLは、4ビットレジスタ識別子によって特定され、16エン트리レジスタファイル4342内の任意の場所に配置され得る。また、ノードプロセッサ4322は、命令メモリ1404-iを更新し得るプログラムカウンタ実行ユニット4344を含む。

40

【0035】

ここで、LSユニット4318-i及びSIMDユニットを参照すると、各々の一般的

50

構造が図6に見られる。図示されるように、LSユニット4318-iは、概して、LSデコーダ4334、LS実行ユニット4336、論理ユニット4346、乗算ユニット4348、右実行ユニット4350、及びLSデータメモリ4339を含む。しかしながら、LSユニット4318-iのためのデータバスに関する詳細は後で記載する。より小さい機能ユニット4338-1~4338-8の各々は、概して(且つ各々が)、(例えばそれぞれ32個のレジスタを有し得る)SIMDレジスタファイル4358-1~4358-8、左論理ユニット4352-1~4352-8、乗算ユニット4354-1~4354-8、及び右論理ユニット4356-1~4356-8を含む。これらの左論理ユニット4352-1~4352-8、乗算ユニット4354-1~4354-8、及び右論理ユニット4356-1~4356-8は、概して、それぞれ、左、中央、及び右ユニット4346、4348、及び4350、の複製である。また、LSユニット4318-iと同様、各機能ユニット4338-1~4338-8のためのデータバスは、後で記載する。

10

20

30

40

50

【0036】

また、ノード(即ち、ノード808-i)のための3つの例示の構成では、幾つかの構成要素(即ち、論理ユニット4352-1)のサイズ、又は対応する命令は変化してよいが、他は同じままであり得る。LSデータメモリ4339、ルックアップテーブル、及びヒストグラムは、相対的に同じままとなる。好ましくは、LSデータメモリ4339は、最初の16個の位置がコンテキストベースアドレスを保持し、残りの位置がコンテキストによってアクセス可能であるような、約512x32ビットであり得る。(概して、PC実行ユニット4344内にある)ルックアップテーブル又はLUTは、メモリサイズが16Kbの最大12個のテーブルを有し得る。ここで、4ビットがテーブルを選択するために用いられ得、14ビットがアドレス指定のために用いられ得る。(概して、PC実行ユニット4344内に配置される)ヒストグラムは、4個のテーブルを有することができる。ここで、ヒストグラムはテーブルを選択するために4ビットIDをLUTと共有し、アドレス指定のために8ビットを使用する。次の表1では、例示の3つの構成各々の命令サイズが示され、それらは種々の構成要素のサイズに対応し得る。

【表1】

構成要素	第1の構成	第2の構成	第3の構成
4つのノード(即ち、808-i)と共有されると仮定される命令メモリ(即ち、1404-i)	1024x182ビットの4セット	1024x252ビットの4セット	1024x318ビットの4セット
丸めユニット(即ち、3450)命令	16ビット	22ビット	22ビット
乗算ユニット(即ち、4348)命令	16ビット	24ビット	24ビット
論理ユニット(即ち、4346)命令	16ビット	24ビット	24ビット
LSユニット命令	132ビット	160ビット	156ビット
ノードプロセッサ4322命令	0ビット	20ビット	20ビット
コンテキストスイッチ表示	2ビット	2ビット	2ビット
命令ラインの配置(命令パケット形式)	コンテキスト: C:LS1:LS 2:LS3:LS 4:LS5:LS 6:LU:MU: RU	コンテキスト: C:LS1:T2 0:LS2:LS 3:LS4:LS 5:LS6:LU: MU:RU	コンテキスト: C:LS1:T2 0:LS2:LS 3:LS4:LS 5:LS6:LU: MU:RU

【0037】

図7を参照すると、共有機能メモリ1410が見られる。共有機能メモリ1410は、概して、ノードにより(コストの理由で)充分サポートされない操作をサポートする、大型の集中メモリである。共有機能メモリ1410の主な構成要素は、(各々が、例えば48~1024Kバイトの間で構成可能なサイズ及び構成を有する)2つの大型メモリ、機能メモリ7602及びベクトルメモリ7603である。この機能メモリ7602は、高帯

域、ベクトルベースのルックアップテーブル (LUT)、及びヒストグラム、同期、命令駆動型の実装を提供する。ベクトルメモリ 7603 は、(上記のセクション 8 で説明したように) ベクトル命令を暗示する、(例えば) 6 発行命令プロセッサ (即ち、SFM プロセッサ 7614) による操作をサポートし得る。ベクトル命令は、例えば、ブロックベースのピクセル処理のために用いられ得る。典型的には、この SFM プロセッサ 7614 は、メッセージングインタフェース 1420 及びデータバス 1422 を用いてアクセスされ得る。SFM プロセッサ 7614 は、例えば、ノード内の SIMD データメモリに比べて、より一般的な構成、及びより大きな総メモリサイズを有し、より一般的な処理がデータに適用される得る、ワイドピクセルコンテキスト (64 ピクセル) 上で動作し得る。それは、標準 C++ 整数データタイプ上で、スカラ、ベクトル、及びアレイ操作、並びに、各種のデータタイプと適合性のある、パックされたピクセル上の操作をサポートする。例えば、図示されるように、ベクトルメモリ 7603 及び機能メモリ 7602 に関連する SIMD データバスは、概して、ポート 7605-1 ~ 7605-Q 及び機能ユニット 7607-1 ~ 7607-P を含む。

10

20

30

40

50

【0038】

全ての処理ノード (即ち、808-i) が機能メモリ 7602 及びベクトルメモリ 7603 にアクセスし得るという意味で、機能メモリ 7602 及びベクトルメモリ 7603 は、全般的に「共有」されている。機能メモリ 7602 に提供されるデータは、SFM ラッパーを介して (典型的にはライトオンリーの方式で) アクセスされ得る。また、この共有は、全般的に、ノード (即ち、808-i) を処理するための上述のコンテキスト管理と一貫性がある。また、処理ノードと共有機能メモリ 1410 との間のデータ I/O もデータフロープロトコルを使用し、処理ノードは、典型的には、ベクトルメモリ 7603 に直接アクセスできない。また、共有機能メモリ 1410 は、機能メモリ 7602 に書き込むことができるが、処理ノードによってアクセスされている間は、書き込むことができない。処理ノード (即ち、808-i) は、機能メモリ 7602 内の共通位置を読み出し及び書き込みできるが、(通常は) リードオンリー LUT 操作、又はライトオンリーヒストグラム操作のいずれかとしてである。また、処理ノードが機能メモリ 7602 領域への読み出し - 書き込みアクセスを有することも可能であるが、これは所定のプログラムによるアクセスに限定されるべきである。

【0039】

図 8 を参照すると、共有機能メモリ 1410 のための SIMD データバス 7800 の例が見られる。例えば、8 個の SIMD データバス (これらは、16 ビットパックデータを操作できるので、2 つの 16 ビットハーフに区分され得る) が使用され得る。図示されるように、これらの SIMD データバスは、全般的に、バンクのセット 7802-1 ~ 7802-L、関連するレジスタ 7804-1 ~ 7804-L、及び関連する機能ユニットのセット 7806-1 ~ 7806-L を含む。

【0040】

図 9 では、SIMD データバス (即ち及び例えば、レジスタ 7804-1 ~ 7804-L の 1 つの一部、及び機能ユニット 7806-1 ~ 7806-L の 1 つの一部) の例が見られる。図示されるように、例えば、この SIMD データバスは、16 - エントリ、32 ビットレジスタファイル 7902、2 つの 16 ビット乗算器 7904 及び 7906、及び、同様に、1 サイクル中に 2 つの 16 ビットパック操作を実行し得る、単一の 32 ビット算術 / 論理ユニット 7908 を含み得る。また、例として、各 SIMD データバスは、2 つの、独立した 16 ビット演算、又は組み合わせた 32 ビット演算を実行し得る。例えば、これは、32 ビットの加算器と組み合わせた 16 ビット乗算器を用いて 32 ビットの乗算を形成し得る。また、算術 / 論理ユニット 7908 は、加算、減算、論理演算 (即ち、AND)、比較、及び条件移動を実行することが可能である。

【0041】

図 8 に戻ると、SIMD データバスレジスタ 7804-1 ~ 7804-L は、ベクトルメモリ 7603 へのロード / ストアインタフェースを使用し得る。これらのロード及びス

トアは、ノード（即ち、808 - i）による並列LUT及びヒストグラムアクセスのために提供されるベクトルメモリ7603の特徴を使用し得る。ノードのために各SIMDデータパスハーフは機能メモリ7602内へのインデックスを提供し得る。同様に、SFMプロセッサ7614内の各SIMDデータパスハーフは、独立ベクトルメモリ7603アドレスを提供し得る。アドレス指定は、概して、隣接するデータパスが（例えば）スカラ、ベクトル、及び8、16、又は32ビットデータのアレイなど、データタイプの多数のインスタンス上で同じ操作を実行できるように構成される。これらは、ベクトル暗示アドレス指定モードと称される（ベクトルが、リニアのベクトルメモリ7603アドレス指定を用いて、SIMDによって暗示される）。或いは、各データパスはバンク7608 - 1 ~ 7608 - J内のフレームの領域からのパックされたピクセル上で操作し得る。これらは、ベクトルパック化アドレス指定モードと称される（パックされたピクセルのベクトルは、二次元ベクトルメモリ7603アドレス指定を用いて、SIMDによって暗示される）。両方の場合において、ノードプロセッサ4322と同じように、プログラミングモデルがSIMDの幅を隠すことができ、プログラムはあたかもそれらが単一ピクセル又は他のデータタイプのエレメント上で演算したかのように書き込まれる。

10

20

30

40

50

【0042】

ベクトル暗示データタイプは、概して、各SIMDデータパスによって個別に演算される8ビットchar、16ビットハーフワード、又は32ビットint、のいずれかのSIMD実装ベクトルである（即ち、図9）。これらのベクトルは、概して、プログラム内では明示的でなく、ハードウェア演算によって暗示される。また、これらのデータタイプは、明示的プログラムベクトル又はアレイ内のエレメントとして構成され得る。SIMDは、隠された2次元、又は3次元を、これらのプログラムベクトル又はアレイに、効果的に加算する。実際には、プログラミングビューは専用の32ビットデータメモリを備える単一のSIMDデータパスであり得る。このメモリは従来のアドレス指定モードを用いてアクセスされる。ハードウェアでは、このビューは、32のSIMDデータパスの各々がプライベートデータメモリの外観を有するような方式でマッピングされるが、この機能性を共有機能メモリ1410に実装するために、ベクトルメモリ7603のワイドなバンクされた構成の利点を実装に利用する。

【0043】

SFMプロセッサ7614SIMDは、概して、記述子を用いて、ノードプロセッサ4322コンテキストに類似するベクトルメモリ7603コンテキスト内で動作する。記述子はバンクのセット7802 - 1に整列され、全体のベクトルメモリ7603にアクセスするのに十分に大きい（即ち、1024kBのサイズの場合、13ビット）ベースアドレスを有する。SIMDデータパスの各ハーフは、一番左のデータパスのための0から始まる6ビット識別子（POSN）で番号付けされる。ベクトル暗示アドレス指定の場合、この値のLSBは、概して無視され、残りの5ビットは、データパスによって生成されたベクトルメモリ7603アドレスをベクトルメモリ7603内のそれぞれのワードに整列させるために用いられる。

【0044】

処理クラスタ1400内で、汎用RISCプロセッサは様々な目的を果たす。例えば、（RISCプロセッサであり得る）ノードプロセッサ4322はプログラムフロー制御のために用いられ得る。RISCアーキテクチャの例を以下に説明する。

【0045】

図10を参照すると、RISCプロセッサ5200（即ち、ノードプロセッサ4322）の更に詳細な例が見られる。プロセッサ5200によって使用されるパイプラインは、概して、処理クラスタ1400内で一般のハイレベル言語（即ち、C/C++）を実行するためのサポートを提供する。動作においては、プロセッサ5200は、フェッチ、デコード、及び実行の3段のパイプラインを用いる。典型的には、コンテキストインタフェース5214及びLSポート5212が命令をプログラムキャッシュ508に提供し、その命令は命令フェッチ5204によってプログラムキャッシュ5208からフェッチされ得

る。命令フェッチ5204とプログラムキャッシュ5208との間のバスは、例えば、40ビット幅であり得、プロセッサ5200がデュアル発行命令（即ち、命令が40ビット又は20ビット幅であり得る）をサポートすることを可能にする。概して、（処理ユニット5202内の）「A側」及び「B側」の機能ユニットはより小さい命令（即ち、20ビット命令）を実行し、一方、「B側」機能ユニットは、より大きな命令（即ち、40ビット命令）を実行する。提供された命令を実行するために、処理ユニットは、レジスタファイル5206を「スクラッチパッド」として使用し得る。このレジスタファイル5206は、「A側」と「B側」との間で共有される（例えば）16 - エントリ、32ビットレジスタファイルであり得る。また、プロセッサ5200は、制御レジスタファイル5216及びプログラムカウンタ5218を含む。また、プロセッサ5200はバウンダリピン又はリードを介してアクセスされ得る。各例を、表2で説明する（「z」は、アクティブローピンを示す）。

10

20

30

40

【表2】

Pin名	幅	方向	目的
フラグ及びストラッピング			
risc_is_mtv	1	出力	MTV命令がデコードされる時デコード段5308でアサートされる。（複製を用いて、プロセッサ5200からベクトル又はSIMDレジスタへ移動する。）
risc_is_mtvvr	1	出力	MTVVR命令がデコードされる時デコード段5308でアサートされる。（プロセッサ5200からベクトル又はSIMDレジスタへ移動する）
risc_is_mfvvr	1	出力	MFVVR命令がデコードされる時デコード段5308でアサートされる。（ベクトル又はSIMDレジスタからプロセッサ5200へ移動する）
risc_is_mfvre	1	出力	MFVRC命令がデコードされる時デコード段5308でアサートされる。（コラプスを用いて、プロセッサ5200からベクトル又はSIMDレジスタへ移動する。）
risc_is_mtvre	1	出力	MTVRE命令がデコードされる時デコード段5308でアサートされる。（拡張を用いて、プロセッサ5200からベクトル又はSIMDレジスタへ移動する。）

汎用レジスタからベクトル/SIMDレジスタ転送インタフェース			
risc_vec_ua	5	出力	MTVVR及びMFVVR命令のためのベクトル（又はSIMD）ユニット（aka「レーン」）アドレス。実行段5310で駆動される。
risc_vec_wa	5	出力	MTV、MTVRE、MTVVR命令の場合、ベクトル（又はSIMD）レジスタファイル書き込みアドレス MFVVR及びMFVRC命令の場合、要求されたベクトルデータを受け取るためのGPRのアドレスを含む。 実行段5310で駆動される。
risc_vec_wd	32	出力	ベクトル（又はSIMD）レジスタファイル書き込みデータ 実行段5310で駆動される。
risc_vec_hwz	2	出力	ベクトル（又はSIMD）レジスタファイル書き込みハーフワード選択 00 = 両方書き込む 10 = 下位を書き込む 01 = 上位を書き込む 11 = 読み出す vec_regf_enzのアサートでゲーティングされる。 実行段5310で駆動される。
risc_vec_ra	5	出力	ベクトル（又はSIMD）レジスタファイル読み出しアドレス。 実行段5310で駆動される。
vec_risc_wrz	1	入力	レジスタファイル書き込みイネーブル。MFVVR又はMFVRC命令の結果として書き込みデータを戻すとき、ベクトル（又はSIMD）によって駆動される。
vec_risc_wd	32	出力	ベクトル（又はSIMD）レジスタファイル書き込みデータ。 実行段5310で駆動される。
vec_risc_wa	4	入力	汎用レジスタファイル5206アドレス。MFVVR又はMFVRC命令の結果として戻るベクトルデータの宛て先である。

ノードインタフェース			
node_regf_wrf[0:5]r	1 b × 6	入力	レジスタファイル書き込みポート書き込みイネーブル
node_regf_wa[0:5]	4 b × 6	入力	レジスタファイル書き込みポートアドレス。 ノードサポートのため、汎用レジスタファイル5 2 0 6へ6個の書き込みポートがある。
node_regf_wd[0:5]	3 2 b × 6	入力	レジスタファイル書き込みポートデータ。
node_regf_rd	5 1 2	出力	レジスタファイル読み出しデータ。
node_regf_rdz	1	入力	汎用レジスタファイル5 2 0 6コンテンツ読み出しイネーブル。

共有機能メモリアンタフェース (共有機能メモリ1 4 1 0を備えるプロセッサのために使用され得る)			
vmem_rdy	1	入力	ベクトルメモリレディ。 通常存在し、不使用時はハイにストラップされる。
risc_vec_valid	1	出力	SFM命令レーンが有効であることを示す。 これは通常アサートされるが、プロセッサ5 2 0 0が非並列2 0ビット命令対の第2のハーフを実行しているとき、ディアサートされる。
risc_fimn_addr	2 0	出力	ベクトル暗示ロード/ストアアドレスバス。
risc_fimn_bez	4	出力	ベクトル暗示ロード/ストアバイトイネーブル。
risc_vec_opr	4	出力	このバスは、ベクトル暗示ストアのためのベクトルユニットソースレジスタ、又はベクトル暗示ロードのためのベクトルユニット宛て先レジスタを表す。
risc_is_vild	1	出力	ベクトル暗示符号付きロードフラグ。
risc_is_vildu	1	出力	ベクトル暗示符号無しロードフラグ。
risc_is_vist	1	出力	ベクトル暗示ストアフラグ。
risc_hg_posn	8	出力	プロセッサ5 2 0 0HG_POSN制御レジスタのカレントのコンテンツを反映する。
risc_regf_ra[1:0]	4 b × 2	入力	レジスタファイル読み出しアドレスポート。2つのポートがある。これらのピンは、レーン0 (一番左) ベクトルユニットによって駆動される。ベクトルユニットが、GPRファイルの下位の4レジスタの1つを読み出すことを許可する。
risc_regf_rdl[1:0]r	1 b × 2	入力	ディアサート時は、risc_regf_rdata01バス上のスイッチングをゲートオフする。risc_regf_rdata上の有効データを読み出すために、ローに駆動されるべきである。
risc_regf_rdata[1:0]	3 2 b × 2	出力	レジスタファイル読み出しデータポート。2つのポートがある。これらのピンは、レーン0 (一番左) ベクトルユニットによって駆動される。これらは、risc_regf_raに関連する読み出しデータバスである。
risc_inc_hg_posn	1	出力	BHGN E命令がデコードされるときDOでアサートされる。
wrp_hgposn_nchgsz	1	入力	SFMラッパーによってアサートされる。HG_POSN及びHG_S I Z Eのラッパーコピーが等しくないか否かを示す。

10

20

【 0 0 4 6 】

図 1 1 を参照すると、プロセッサ 5 2 0 0 が、パイプライン 5 3 0 0 を備えてより詳細に示しているのが見られる。ここでは、(フェッチ段 5 3 0 6 に対応する) 命令フェッチ 5 2 0 4 が A 側及び B 側に分割される。ここで、A 側は、(1つの 40 ビット命令又は 2つの 20 ビット命令を有する 40 ビット幅の命令ワードであり得る) 「フェッチパケット」の最初の 20 ビット(即ち、[1 9 : 0])を受け取り、B 側はフェッチパケットの最後の 20 ビット(即ち、[3 9 : 2 0])を受け取る。典型的には、命令フェッチ 5 2 0 4 はフェッチパケット内の命令の構造及びサイズを決定し、それに応じて命令をディスパッチする(以下のセクション 7 . 3 で説明する)。

30

【 0 0 4 7 】

(デコード段 5 3 0 8 及び処理ユニット 5 2 0 2 の一部である) デコーダ 5 2 2 1 は命令フェッチ 5 2 0 4 からの命令をデコードする。デコーダ 5 2 2 1 は、概して、(インターミディエイトを生成するための) 演算子フォーマット回路 5 2 2 3 - 1 及び 5 2 2 3 - 2 及びそれぞれ B 側及び A 側のためのデコード回路 5 2 2 5 - 1 及び 5 2 2 5 - 2 を含む。デコーダ 5 2 2 1 からの出力は、次に、(デコード段 5 3 0 8 及び処理ユニット 5 2 0 2 の一部である) デコードトゥーエグゼキューションユニット (d e c o d e - t o - e x e c u t i o n u n i t) 5 2 2 0 によって受け取られる。デコードトゥーエグゼキューションユニット 5 2 2 0 は、フェッチパケットを介して受け取る命令に対応する、実行ユニット 5 2 2 7 のためのコマンドを生成する。

40

【 0 0 4 8 】

実行ユニット 5 2 2 7 の A 側及び B 側も細分されている。実行ユニット 5 2 2 7 の B 側及び A 側の各々は、それぞれ、乗算ユニット 5 2 2 2 - 1 / 5 2 2 2 - 2、プーランユニ

50

ット5 2 2 6 - 1 / 5 2 2 6 - 2、加算/減算ユニット5 2 2 8 - 1 / 5 2 2 8 - 2、及び移動ユニット5 3 3 0 - 1 / 5 3 3 0 - 2を含む。また、実行ユニット5 2 2 7のB側は、ロード/ストアユニット5 2 2 4及びブランチユニット5 2 3 2を含む。乗算ユニット5 2 2 2 - 1 / 5 2 2 2 - 2、ブーランユニット5 2 2 6 - 1 / 5 2 2 6 - 2、加算/減算ユニット5 2 2 8 - 1 / 5 2 2 8 - 2、及び移動ユニット5 3 3 0 - 1 / 5 3 3 0 - 2は、それぞれ、(A側及びB側の各々のための、読み出しアドレスを含む)汎用レジスタファイル5 2 0 6にロードされたデータ上で、乗算演算、論理ブーラン演算、加算/減算演算、及びデータ移動演算を実行する。制御レジスタファイル5 2 1 6内で移動演算も実行され得る。

【0049】

ベクトル処理モジュールを備えるRISCプロセッサが、概して共有機能メモリ1 4 1 0と共に用いられる。このRISCプロセッサは、プロセッサ5 2 0 0のために用いられるRISCプロセッサと大体同じであるが、計算及びロード/ストア帯域幅を拡張するために、ベクトル処理モジュールを含む。このモジュールは、各々が1サイクルに4 - 演算実行パケットを実行する能力のある、16個のベクトルユニットを含み得る。典型的な実行パケットは、概して、ベクトルメモリアレイからのデータロード、2つのレジスタウェーレジスタ演算、及び、ベクトルメモリアレイへの結果のストアを含む。このタイプのRISCプロセッサは、80ビット幅又は120ビット幅の命令ワードを一般に使用する。この命令ワードは、概して「フェッチパケット」を構成して、整列されない命令を含んでもよい。フェッチパケットは、プロセッサ5 2 0 0に使用されるものと同様の、ベクトル

【0050】

次に「実行パケット」が1つ又は複数のフェッチパケットから形成され得る。部分実行パケットは完了まで命令キューの中に保持される。典型的には、実行段(即ち、5 3 1 0)へ完全実行パケットが提出される。単一サイクル中に、(例えば)4つのベクトルユニット命令、(例えば)2つのスカラ命令、又は(例えば)20ビット及び40ビット命令の組合せが実行されてもよい。また、連続する20ビット命令がシリアルに実行されてもよい。カレントの20ビット命令のビット19が設定される場合、これはカレントの命令及び後続の20ビット命令が実行パケットを形成することを示す。ビット19は、概して、Pビット又は並列ビットと称され得る。Pビットが設定されない場合、これは実行パケットの終了を示す。Pビットが設定されない連続する20ビット命令は、20ビット命令のシリアル実行を引き起こす。なお、この(ベクトル処理モジュールを備える)RISCプロセッサは、以下の制約の任意のものを含んでもよいことにも留意されたい。

(1) (例えば)40ビット命令で、Pビットが1に設定されることは違反である。

(2) ロード又はストア命令は、命令フェッチバスのB側に現れるべきである(即ち、40ビットのロード及びストアの場合、ビット79:40、20ビットのロード及びストアの場合、フェッチバスのビット79:60)。

(3) 単一のスカラロード又はストアは、違反ではない。

(4) ベクトルユニットでは、1つのフェッチパケット内に単一のロード及び単一のストアの両方が存在し得る。

(5) 40ビット命令が、Pビットが1に等しい20ビット命令に先行されることは違反である。

(6) これらの違反状態を検出するためのハードウェアが適所に存在しない。これらの

10

20

30

40

50

制約はシステムプログラムツール 7 1 8 によって実施されることが予期されている。

【 0 0 5 1 】

図 1 2 を参照すると、ベクトルモジュールの例が見られる。ベクトルモジュールは、検出器デコーダ 5 2 4 6、デコードトゥーエグゼキューションユニット 5 2 5 0、及び実行ユニット 5 2 5 1 を含む。また、ベクトルデコーダは、命令フェッチ 5 2 0 4 から命令を受け取るスロットデコーダ 5 2 4 8 - 1 ~ 5 2 4 8 - 4 を含む。典型的には、スロットデコーダ 5 2 4 8 - 1 及び 5 2 4 8 - 2 は互いに類似した方式で動作し、スロットデコーダ 5 2 4 8 - 3 及び 5 2 4 8 - 4 はロード/ストアデコーディング回路要素を含む。次にデコードトゥーエグゼキューションユニット 5 2 5 0 は、ベクトルデコーダ 5 2 4 6 のデコードされた出力に基づいて、実行ユニット 5 2 5 1 のための命令を生成し得る。スロットデコーダの各々は、(各々が汎用レジスタ 5 2 0 6 内のデータ及びアドレスを使用する) 乗算ユニット 5 2 5 2、加算/減算ユニット 5 2 5 4、移動ユニット 5 2 5 6、及びブランチユニット 5 2 5 8 によって使用され得る命令を生成し得る。また、スロットデコーダ 5 2 4 8 - 3 及び 5 2 4 8 - 4 は、ロード/ストアユニット 5 2 6 0 及び 5 2 6 2 のためのロード及びストア命令を生成し得る。

10

【 0 0 5 2 】

汎用レジスタファイル 5 2 0 6 は、3 2 ビット汎用レジスタファイルによる 1 6 - エントリであり得る。汎用レジスタ (G P R) の幅はパラメータ化され得る。概して、プロセッサ 5 2 0 0 がノード (即ち、8 0 8 - i) のために用いられる場合、4 + 1 5 (1 5 はバウンダリピンによって制御される) の読み出しポート及び 4 + 6 (6 はバウンダリピンによって制御される) の書き込みポートがあり、一方、 G L S ユニット 1 4 0 8 のために用いられるプロセッサ 5 2 0 0 は、4 個の読み出しポート及び 4 個の書き込みポートを有する。

20

【 0 0 5 3 】

ノードプロセッサ 4 3 2 2 と S I M D (即ち、S I M D データメモリ 4 3 0 6 - 1 及び機能ユニット 4 3 0 8 - 1 を含む S I M D ユニット) との間でデータを移動し得る命令を表 3 に示す。

【 表 3 】

命令	説明
MTV	全ての機能ユニット (即ち、4 3 3 8 - 1) 内でノードプロセッサ 4 3 2 2 レジスタから S I M D レジスタ (即ち、S I M D レジスタファイル 4 3 1 8 - 1) にデータを移動する
MFVVR	一番左の S I M D 機能ユニット (即ち、4 3 3 8 - 1) からノードプロセッサ 4 3 2 2 内のレジスタファイルにデータを移動する。
MTVRE	ノードプロセッサ 4 3 2 2 内のレジスタを、機能ユニット (即ち、4 3 3 8 - 1) へ拡張する。 レジスタを 3 2 の機能ユニットへ拡張する。
MFVRC	S I M D 内の機能ユニットレジスタを、1 つの 3 2 ビット (例えば) へ圧縮する。

30

【 0 0 5 4 】

次の表 4 は、プロセッサ 5 2 0 0 のための命令セットアーキテクチャの例を示す。ここで、

40

- (1) ユニット表示、. S A 及び . S B は、どちらの発行スロットが 2 0 ビット命令が実行するかを識別するために用いられる。
- (2) 4 0 ビット命令は、規則により、B 側 (. S B) で実行される。
- (3) 基本形式は < ニーモニク > < ユニット > < カンマで区切られたオペランドリスト > である。
- (4) 擬似コードは、C + + シンタックスを有し、適切なライブラリを用いて、シミュレータ又は他のゴールデンモデルに直接含まれ得る。

【表 4】

Table 4

シンタックス／擬似コード	説明
MFVRC .(SB) s1(R5),s2(R4) void ISA::OPC_MFVRC_40b_266 (Vreg &s1, Gpr &s2) { Event initiate,complete; Reg s2Save; rise_is_mfvrc_assert(1); vec_regf_enz_assert(0); vec_regf_hwz_assert(0x3); vec_regf_ra_assert(s1); s2Save = s2.address(); initiate.live(true); complete.live(vec_wdata_wrz.is(0)); }	MOVE VREG TO GPR, COLLAPSE VREGをGPRに 移動する。 コラプス
MFVVR .(SB) s1(R5), s2(R5), s3(R4) void ISA::OPC_MFVVR_40b_264 (Vunit &s1, Vreg &s2,Gpr &s3) { Event initiate,complete; Reg s3Save; rise_is_mfvvr_assert(1); vec_regf_ua_assert(s1); vec_regf_hwz_assert(0x3); }	MOVE UNIT/VREG TO GPR UNIT/VREGを GPRに移動する

10

シンタックス／擬似コード	説明
vec_regf_enz_assert(0); vec_regf_ra_assert(s2); s3Save = s3.address(); initiate.live(true); complete.live(vec_wdata_wrz.is(0)); }	
MFVVR .SB s1(R5), s2(R5), s3(R4) void ISA::OPC_MFVVR_40b_264 (Vunit &s1, Vreg &s2,Gpr &s3) { Reg s3Save; rise_is_mfvvr_assert(1); rise_vec_ua_assert(s1); rise_vec_ra_assert(s2); s3Save = s3.address(); initiate.live(true); vec_risc_wa_assert(s3); vec_risc_wd gets value of Vreg(risc_vec_ra); complete.live(vec_rise_wrz.is(0)); //ditto }	MOVE VUNIT/VREG TO GPR VUNIT/VREGを GPRに移動する
MTV .(SA,SB) s1(R4), s2(R5) void ISA::OPC_MTV_20b_164 (Gpr &s1, Vreg &s2) { Result r1; r1.clear(); r1 = s1.range(0,15); rise_is_mtv_assert(1); }	MOVE GPR TO VREG. REPLICATED (LOW VREG) GPRをVREGに 移動する。 複製 (下位VREG)

20

30

40

シンタックス／擬似コード	説明
<pre>vec_regf_enz_assert(0); vec_regf_wa_assert(s2); vec_regf_wd_assert(r1); vec_regf_hwz_assert(0x0); //active low, write both halves }</pre>	
<pre>MTV (SA,SB) s1(R4), s2(R5) void ISA::OPC_MTV_20b_165 (Gpr &s1, Vreg &s2) { Result r1; r1.clear(); r1.range(16,31) = s1.range(16,31); risc_is_mtv_assert(1); vec_regf_enz_assert(0); vec_regf_wa_assert(s2); vec_regf_wd_assert(r1); vec_regf_hwz_assert(0x0); //active low, write both halves }</pre>	<p>MOVE GPR TO VREG, REPLICATED (HIGH VREG)</p> <p>GPRをVREGに移動する。 複製（上位VREG）</p>
<pre>MTVRE (SB) s1(R4),s2(R5) void ISA::OPC_MTVRE_40b_265 (Gpr &s1, Vreg &s2) { risc_is_mtvre_assert(1); vec_regf_enz_assert(0); vec_regf_wa_assert(s2); vec_regf_wd_assert(s1); vec_regf_hwz_assert(0x0); //active low, both halves }</pre>	<p>MOVE GPR TO VREG, EXPAND</p> <p>GPRをVREGに移動する。拡張</p>
<pre>MTVVR (SB) s1(R4), s2(R5), s3(R5) void ISA::OPC_MTVVR_40b_263 (Gpr &s1, Vunit &s2, Vreg &s3) {</pre>	<p>MOVE GPR TO VUNIT/VREG GPRをVUNIT/VREGへ移動する</p>

10

20

シンタックス／擬似コード	説明
<pre>risc_is_mtvvr_assert(1); vec_regf_ua_assert(s2); vec_regf_enz_assert(0); vec_regf_wa_assert(s3); vec_regf_wd_assert(s1); vec_regf_hwz_assert(0x0); //active low, both halves }</pre>	
<pre>MVKVRHU (SB) s1(U32), s2(R5), s3(R5) void ISA::OPC_MVKVRHU_40b_268 (U16 &s1, Vunit &s2, Vreg &s3) { Result r1; r1 = _unsigned(s1.range(16,31)); risc_is_mtvvr_assert(1); vec_regf_ua_assert(s2); vec_regf_enz_assert(0); vec_regf_wa_assert(s3); vec_regf_wd_assert(r1); vec_regf_hwz_assert(0x1); //active low, high half }</pre>	<p>MOVE U16 TO VUNIT/VREG, HIGH HALF</p> <p>U16をVUNIT/VREGに移動する。 上位ハーフ</p>
<pre>MVKVRLU (SB) s1(U32), s2(R5), s3(R5) void ISA::OPC_MVKVRLU_40b_267 (U16 &s1, Vunit &s2, Vreg &s3) { Result r1; r1.clear(); r1 = _unsigned(s1); risc_is_mtvvr_assert(1); vec_regf_ua_assert(s2); vec_regf_enz_assert(0); vec_regf_wa_assert(s3); vec_regf_wd_assert(r1); vec_regf_hwz_assert(0x0); //active low, both halves }</pre>	<p>MOVE U16 TO VUNIT/VREG, LOW HALF</p> <p>U16をVUNIT/VREGに移動する。 下位ハーフ</p>

30

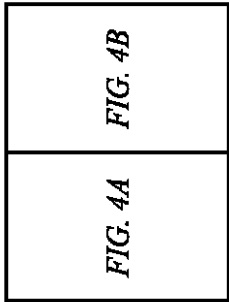
40

【 0 0 5 5 】

本発明に関連する分野の当業者であれば、記載された実施形態及び実現された付加的な実施形態に本発明の請求の範囲内から逸脱することなく変更が行われることが理解されるであろう。

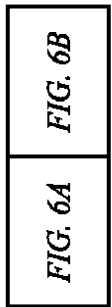
【 図 4 】

FIG. 4



【 図 6 】

FIG. 6



【 図 1 】

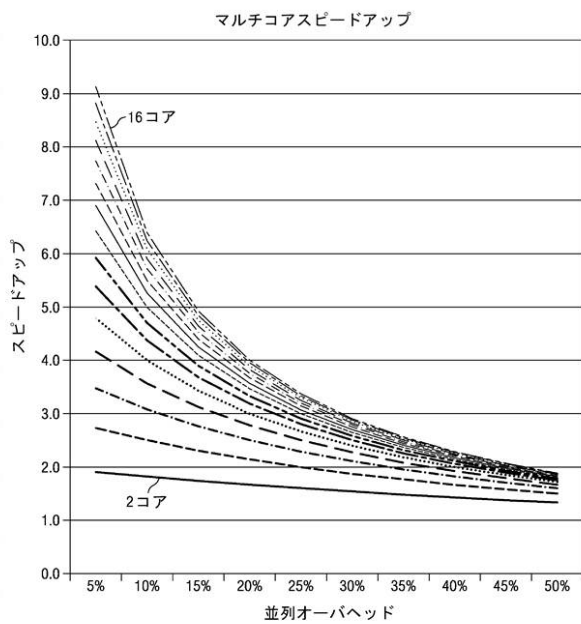
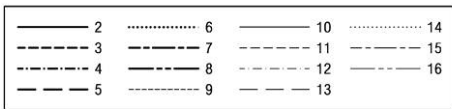


FIG. 1
(従来技術)

【 図 2 】

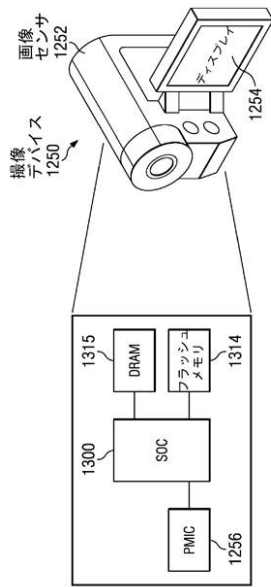


FIG. 2

【 図 3 】

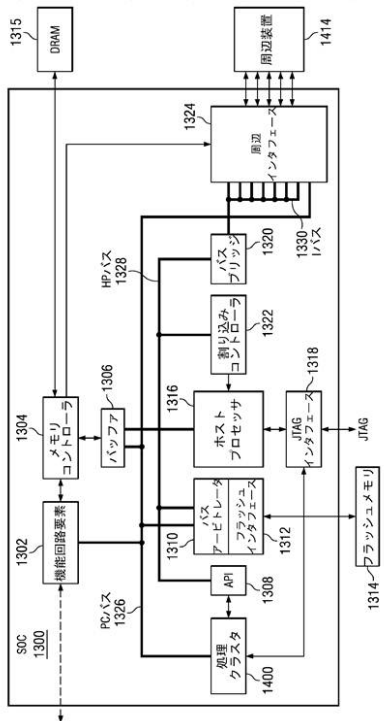
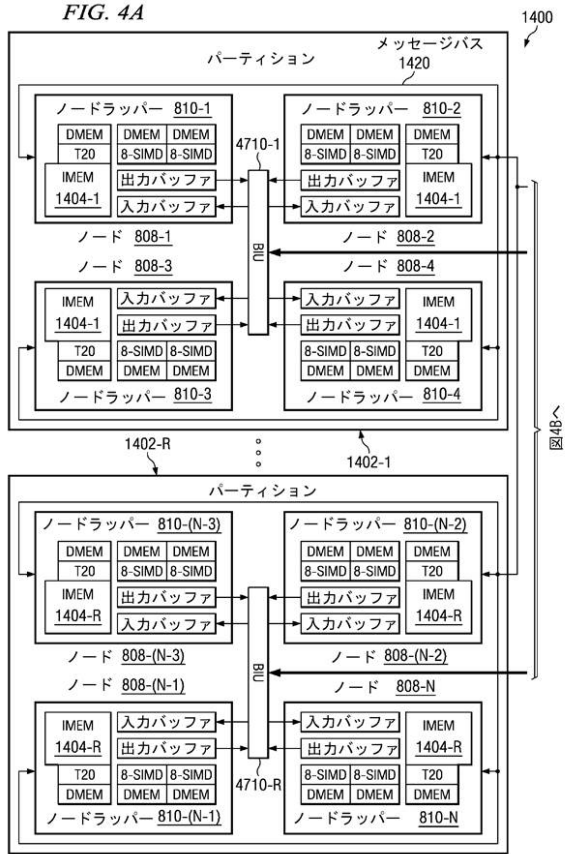


FIG. 3

【図 4 A】



【図 4 B】

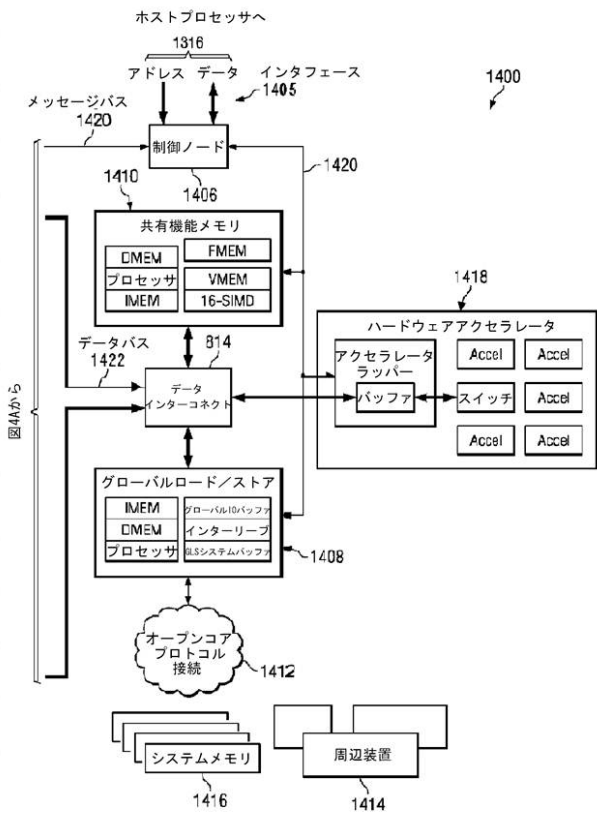


FIG. 4B

【 図 5 】

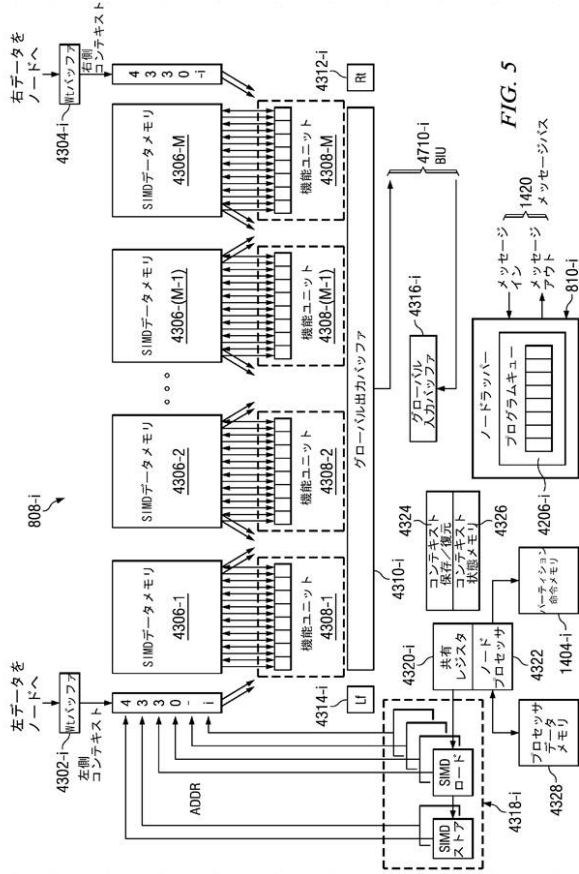


FIG. 5

【 図 6 A 】

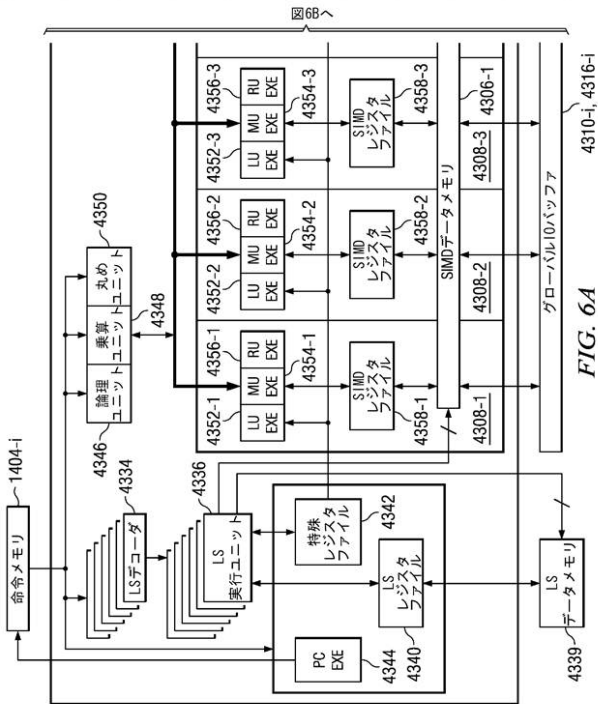
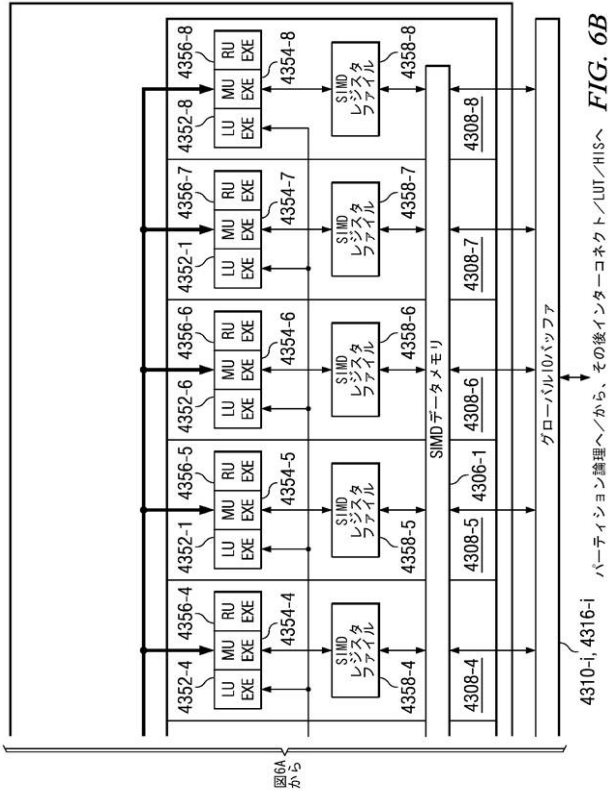


FIG. 6A

【 図 6 B 】



【 図 7 】

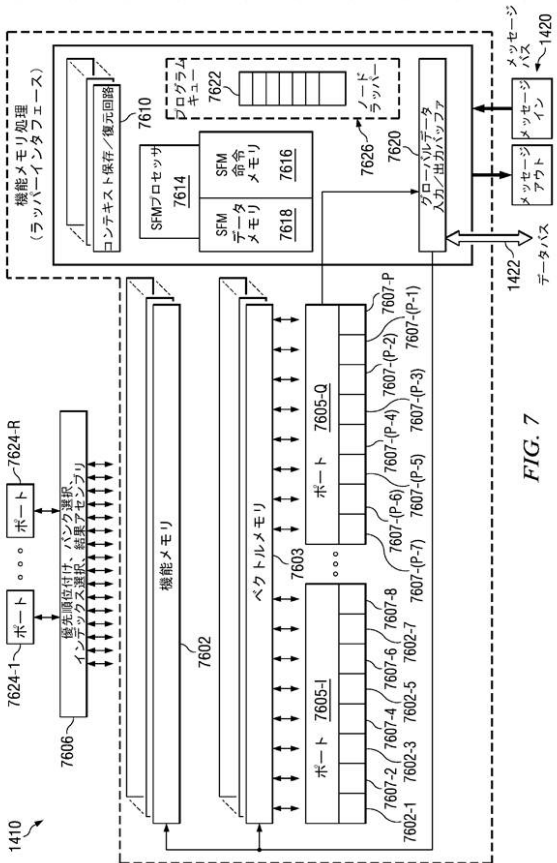


FIG. 7

【 図 8 】

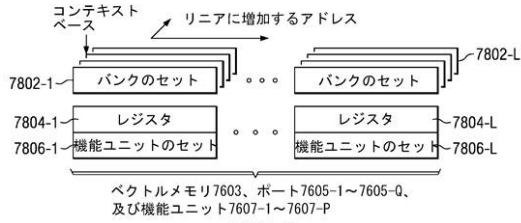


FIG. 8

【 図 9 】

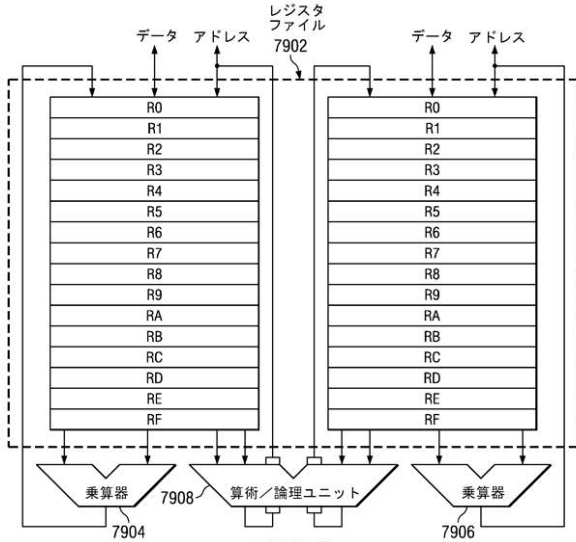


FIG. 9

【 図 10 】

RISC_IS_MTV
RISC_IS_MTVVR
RISC_IS_MFVVR
RISC_IS_MFVRC
RISC_IS_MTVRE
RISC_VEC_UA
RISC_VEC_WA
RISC_VEC_WD
RISC_VEC_HWZ
RISC_VEC_RA
VEC_RISC_WRZ
VEC_RISC_WD
VEC_RISC_WA
NODE_REGF_WA[0:5]Z
NODE_REGF_WA[0:5]
NODE_REGF_WD[0:5]Z
NODE_REGF_WD[0:5]
NODE_REGF_RD
NODE_REGF_RDZ
VMEM_RDY
RISC_VEC_VALID
RISC_FMEM_ADDR
RISC_FMEM_BEZ
RISC_VEC_OPR
RISC_IS_VILD
RISC_IS_VILDU
RISC_IS_VIST
RISC_HG_POSN
RISC_REGF_RA[1:0]
RISC_REGF_RD[1:0]Z
RISC_REGF_RDATA[1:0]
RISC_INC_HG_POSN
WRP_HGPOSN_NE_HGSIZE

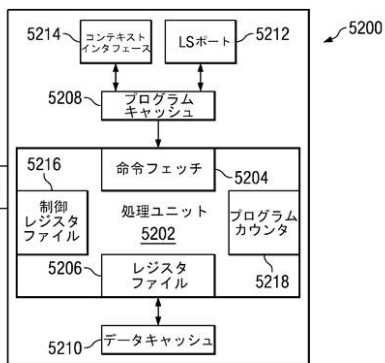
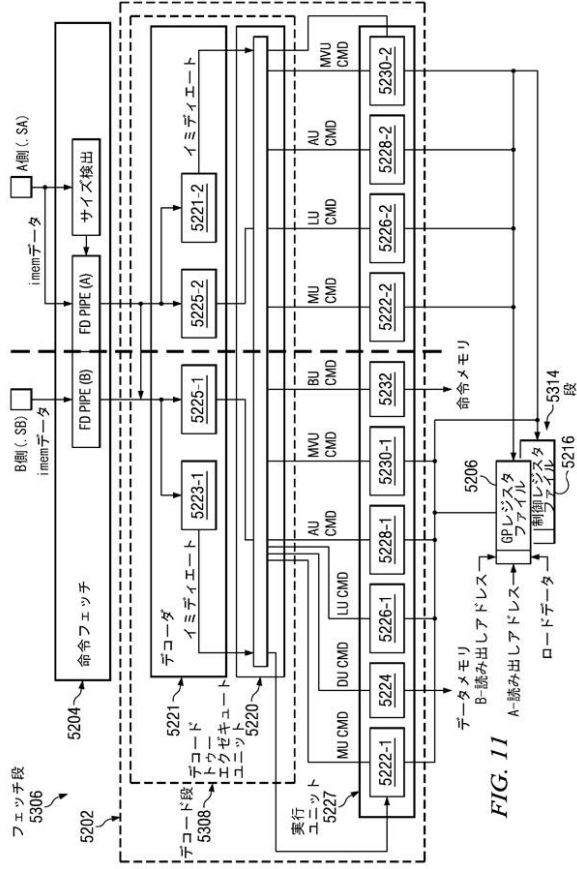
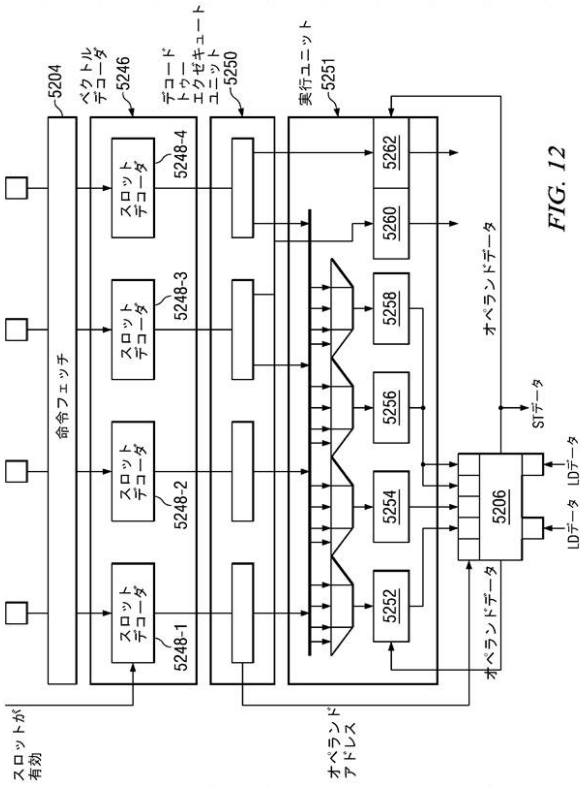


FIG. 10



【 図 1 1 】



【 図 1 2 】



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2011/061487
A. CLASSIFICATION OF SUBJECT MATTER		
<i>G06F 13/14(2006.01)i, G06F 13/38(2006.01)i</i>		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) IPC : G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: SIMD*,vector*,register*,memory*		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2002-0052868 A1 (SANJEEV MOHINDRA et al.) 02 May 2002 See Paragraphs [0033]-[0047]; figures 3-4.	1-19
A	US 2008-0133874 A1 (CAPEK PETER G. et al.) 05 June 2008 See Paragraphs [0059]-[0061] and [0073]-[0089]; figures 3 and 10-12.	1-19
A	US 2005-0055534 A1 (WILLIAM, C. MOYER) 10 March 2005 See Paragraphs [0017]-[0024] and [0135]-[0164]; figure 1.	1-19
A	US 2008-0034185 A1 (DAVE STUTTARD et al.) 07 February 2008 See Paragraphs [0040]-[0054]; figure 3.	1-19
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 12 JULY 2012 (12.07.2012)		Date of mailing of the international search report 16 JULY 2012 (16.07.2012)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer Ko Jae Yong Telephone No. 82-42-481-8212 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/061487

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002-0052868 A1	02.05.2002	None	
US 2008-0133874 A1	05.06.2008	CN 101030192 B JP 2007-234011 A JP 2008-306915 A TW 200821919 A US 2007-0226466 A1 US 2008-0303467 A1 US 7360063 B2 US 7631167 B2 US 7944162 B2	26.05.2010 13.09.2007 18.12.2008 16.05.2008 27.09.2007 11.12.2008 15.04.2008 08.12.2009 17.05.2011
US 2005-0055534 A1	10.03.2005	US 7107436 B2	12.09.2006
US 2008-0034185 A1	07.02.2008	AU 2000-38295 A1 EP 1181648 A1 GB 2348971 A GB 2348972 A GB 2348973 A GB 2348974 A GB 2348975 A GB 2348976 A GB 2348977 A GB 2348978 A GB 2348979 A GB 2348980 A GB 2348981 A GB 2348982 A GB 2348983 A GB 2348984 A GB 2349484 A GB 2352306 A JP 2002-541586 A JP 2011-023036 A JP 2011-023037 A US 2002-0174318 A1 US 2007-0226458 A1 US 2007-0242074 A1 US 2007-0245123 A1 US 2007-0245130 A1 US 2007-0245132 A1 US 2007-0294510 A1 US 2008-0007562 A1 US 2008-0008393 A1 US 2008-0010436 A1 US 2008-0016318 A1 US 2008-0028184 A1 US 2008-0034186 A1 US 2008-0040575 A1	14.11.2000 27.02.2002 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 18.10.2000 01.11.2000 24.01.2001 03.12.2002 03.02.2011 03.02.2011 21.11.2002 27.09.2007 18.10.2007 18.10.2007 18.10.2007 18.10.2007 20.12.2007 10.01.2008 10.01.2008 10.01.2008 17.01.2008 31.01.2008 07.02.2008 14.02.2008

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/061487

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2008-0052492 A1	28.02.2008
		US 2008-0098201 A1	24.04.2008
		US 2008-0162874 A1	03.07.2008
		US 2008-0162875 A1	03.07.2008
		US 2008-0184017 A1	31.07.2008
		US 2009-0198898 A1	06.08.2009
		US 2009-0228683 A1	10.09.2009
		US 7363472 B2	22.04.2008
		US 7506136 B2	17.03.2009
		US 7526630 B2	28.04.2009
		US 7627736 B2	01.12.2009
		US 7802079 B2	21.09.2010
		US 7925861 B2	12.04.2011
		US 7958332 B2	07.06.2011
		US 7966475 B2	21.06.2011
		US 8169440 B2	01.05.2012
		US 8171263 B2	01.05.2012
		US 8174530 B2	08.05.2012
		WO 00-62182 A3	19.10.2000

フロントページの続き

(81) 指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, T, J, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, R, O, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, H, U, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN

(72) 発明者 ウィリアム ジョンソン

アメリカ合衆国 78746 テキサス州 オースティン, コキナ レーン 606

(72) 発明者 ジョン ダブリュー グロツバック

アメリカ合衆国 75013 テキサス州 アレン, インペリアル ドライブ 401

(72) 発明者 ハミッド シェイク

アメリカ合衆国 75013 テキサス州 アレン, ライトハウス レーン 1311

(72) 発明者 アジェイ ジャヤライ

アメリカ合衆国 77479 テキサス州 シュガーランド, サンタ チェース レーン 5430

(72) 発明者 スティーブン ブッシュ

フランス共和国 グラス エフ 06130 シェマン デ シェンヌ 62

(72) 発明者 ミュラリ チナコンダ

アメリカ合衆国 78730 テキサス州 オースティン, リバー プレース ブールバード 4700-7

(72) 発明者 ジェフェリー エル ナイ

アメリカ合衆国 78729 テキサス州 オースティン, サー クリストファー コープ 12545

(72) 発明者 永田 敏雄

アメリカ合衆国 75093 テキサス州 プラノ, オーク ノール ドライブ 4416

(72) 発明者 シャリニ グプタ

アメリカ合衆国 94114 カリフォルニア州 サン フランシスコ, エイピーティー 2 チャーチ ストリート 340

(72) 発明者 ロバート ジェイ ニチカ

アメリカ合衆国 75103 テキサス州 カントン, 400 ブイゼットシーアール 2405

(72) 発明者 デビッド エイチ パートレイ

アメリカ合衆国 75243 テキサス州 ダラス, エコー リッジ コート 10235

(72) 発明者 ガネーシャ サンドララジャン

アメリカ合衆国 75025 テキサス州 プラノ, ホームステッド レーン 2209

Fターム(参考) 5B033 BE01

5B045 AA01