

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4657455号

(P4657455)

(45) 発行日 平成23年3月23日(2011.3.23)

(24) 登録日 平成23年1月7日(2011.1.7)

(51) Int.Cl. F I
G 0 6 F 9/38 (2006.01) G O 6 F 9/38 3 7 O X
G 0 6 F 9/34 (2006.01) G O 6 F 9/34 3 3 O

請求項の数 8 (全 25 頁)

(21) 出願番号	特願2000-604303 (P2000-604303)	(73) 特許権者	597154922
(86) (22) 出願日	平成12年3月9日(2000.3.9)		アルテラ コーポレーション
(65) 公表番号	特表2002-539519 (P2002-539519A)		Altera Corporation
(43) 公表日	平成14年11月19日(2002.11.19)		アメリカ合衆国 95134 カリフォル
(86) 国際出願番号	PCT/US2000/006334		ニア州 サン ホセ イノベーション ド
(87) 国際公開番号	W02000/054144		ライヴ 101
(87) 国際公開日	平成12年9月14日(2000.9.14)	(74) 代理人	100076428
審査請求日	平成19年3月2日(2007.3.2)		弁理士 大塚 康徳
(31) 優先権主張番号	09/267,570	(74) 代理人	100112508
(32) 優先日	平成11年3月12日(1999.3.12)		弁理士 高柳 司郎
(33) 優先権主張国	米国 (US)	(74) 代理人	100115071
前置審査			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二

最終頁に続く

(54) 【発明の名称】 データプロセッサ

(57) 【特許請求の範囲】

【請求項 1】

レジスタファイル索引付けを有するデータプロセッサであって、

前記データプロセッサは、複数のプロセッサ要素 (P E s)、命令メモリ (105) 及び命令シーケンサ (103) を有し、前記複数のプロセッサ要素 (P E s) のそれぞれは N (Nは複数) 個の実行ユニット (A L U, D S U, M A U, L U, S U)、レジスタファイル、超長命令語制御部 (107) 及び超長命令語メモリ (109) を備え、

前記命令シーケンサは、前記複数のプロセッサ要素 (P E s) に接続され、前記命令メモリ (105) から読み出した超長命令語実行命令 (X V 命令) 中で指定された、各プロセッサ要素 (P E) の前記超長命令語メモリ (109) のオフセットアドレスを前記複数のプロセッサ要素 (P E s) に送信することにより、前記 X V 命令が前記超長命令語制御部 (107) に前記超長命令語メモリから超長命令語をフェッチして前記 N 個の実行ユニットで実行するように指示し、

前記 N 個の実行ユニットはそれぞれ各実行ユニットに関連付けられた読み取りポート及び書込みポートを有し、前記読み取りポート及び書込みポートはレジスタファイル索引 (R F I) 制御レジスタ、ルックアヘッドレジスタ及び制御回路を備え、

前記制御回路は、

(i) 最初に受信したレジスタファイル間接指定付きの超長命令実行語命令 (R F I X V 命令) で指示された最初の超長命令語の実行においては、該超長命令語のフィールドが直接前記レジスタファイル中の 1 つのレジスタをアドレスしてアクセスし、並行して、

10

20

前記レジスタファイル索引（R F I）制御レジスタの内容に基づいて該アドレス値を更新して前記ルックアヘッドレジスタにロードするように制御し、

（i i）2 回目に受信したレジスタファイル間接指定付きの超長命令実行語命令（R F I X V 命令）で指示された超長命令語の実行においては、前記ルックアヘッドレジスタの更新されたアドレスが前記レジスタファイル中の 1 つのレジスタをアドレスしてアクセスし、並行して、前記レジスタファイル索引（R F I）制御レジスタの内容に基づいて該アドレス値を更新して前記ルックアヘッドレジスタにロードするように制御することを特徴とするデータプロセッサ。

【請求項 2】

前記プロセッサ要素が超長命令語プロセッサであることを特徴とする請求項 1 に記載のデータプロセッサ。

10

【請求項 3】

前記データプロセッサが、実行するための超長命令語を、前記超長命令語実行命令（X V 命令）の前記オフセットアドレスを使用して前記超長命令語メモリから間接的にアクセスする間接超長命令語プロセッサであることを特徴とする請求項 1 に記載のデータプロセッサ。

【請求項 4】

前記プロセッサ要素が、アレイプロセッサアーキテクチャに含まれる、それぞれ同じ構成の複数のプロセッサ要素の 1 つであることを特徴とする請求項 1 に記載のデータプロセッサ。

20

【請求項 5】

前記制御回路は、実行するためのデータオペランドを記憶している前記レジスタファイルの更新したアドレスの供給を指示するために、1 つ又は複数のレジスタファイル索引制御レジスタを使用することを特徴とする請求項 1 に記載のデータプロセッサ。

【請求項 6】

前記制御回路は、前記ルックアヘッドレジスタが指定され、定数を現在のアドレスに加算或いは減算することにより前記現在のアドレスを更新して、別のレジスタアドレスを選択することを特徴とする請求項 1 に記載のデータプロセッサ。

【請求項 7】

前記制御回路による前記更新により更に、多数のプログラム可能な前記レジスタのセットの特定の 1 つから開始して選択するレジスタアドレスを前記レジスタのセット内で巡回することを特徴とする請求項 6 に記載のデータプロセッサ。

30

【請求項 8】

各ポートの前記制御回路を、更新方法及びレジスタのアドレスのセットについて独立して構成することができ、或いは、レジスタファイル索引付けを選択的に使用不可にできる制御機構をさらに含むことを特徴とする請求項 1 に記載のデータプロセッサ。

【発明の詳細な説明】

【0001】

（関連出願の相互参照）

本願出願は、1998 年 3 月 12 日出願の「Register File Indexing Method and Apparatus for Providing Indirect Control of Register in a VLIW Processor」という名称の米国仮特許出願第 60/077,766 号の利点を主張する。

40

【0002】

（発明の分野）

本発明は、一般に、超長命令語（VLIW）処理における改良に関し、より詳細には、VLIW プロセッサにおけるレジスタ・アドレッシングの間接制御を提供するための有利なレジスタファイル索引付け（R F I）技術に関する。

【0003】

（発明の背景）

1 つの重要なプロセッサ・モデルは、ベクトル処理のものである。このモデルは、従来の

50

スーパーコンピュータにおいて長年使用されてきた。このモデルの典型的な特徴は、専用ベクトル命令、専用ベクトル・ハードウェア、及びデータのブロックに効率的に操作を加える能力の使用である。まさにこの、典型的にはベクトル・データ・タイプのみに操作を加える能力により、このモデルには柔軟性がなく、多様な処理要件に効率的に対処できない。加えて、従来のベクトル・プロセッサでは、制御スカラ処理のサポートが典型的には、分離したハードウェアにおいて、或いは分離した制御プロセッサにおいて行われた。もう1つのプロセッサモデルは、従来の超長命令語（V L I W）プロセッサモデルであり、これは、標準の単一プロセッサタイプの単一機能動作を長い命令語に、専用のマルチサイクル・ベクトル処理機能なしに連結することに基づいた、並列処理モデルに相当する。ブロックデータ・ベクトル・パイプラインを効率的に動作するには、個別のベクトル要素を引き渡すための効率的なインターフェイスを有することが重要である。このため、成功したクラスの従来技術のベクトルマシンは、レジスタベースであった。このレジスタベースのベクトルプロセッサは、ベクトル要素用の高性能レジスタを提供し、要素の効率的なアクセスを機能実行ユニットによって可能にする。実施特有のベクトル長の値に結合された単一のベクトル命令が、ブロック・データ・マルチサイクル動作を引き起こす。加えて、多数のベクトルマシンが連鎖機能を提供しており、個別のベクトル要素への操作が直接、他のベクトル関数ユニットへ経路指定されて、性能が向上される。これらの従前の特徴及び能力が、本発明のための背景を提供する。本発明の一目的は、スカラ、V L I W、及び柔軟性のあるベクトル処理能力を、間接的なV L I Wプロセッサにおいて効率的に組み込むことである。

【0004】

典型的な限定命令セット・コンピュータ（R I S C）及びV L I Wプロセッサでは、レジスタオペランドのアクセスが、レジスタファイルに格納されたオペランドのレジスタアドレスを表す短命令語（S I W）ビット・フィールドから決定される。レジスタベースのベクトルプロセッサでは、専用のハードウェアが使用される。このハードウェアは単一ベクトル命令によって開始され、専用ベクトルレジスタからのベクトル要素（オペランドデータ）のアクセスを自動化する。データのブロックにおけるマルチサイクル実行も自動化される。

【0005】

従来は、レジスタ・オペランド・データの自動アクセスをサポートするために使用された、専用のハードウェア技術もあった。例えば、米国特許第5,680,600号は、レジスタファイル・アドレッシングを自動化するためにループ又は繰り返し命令を使用して、レジスタファイルにアクセスするための技術を記載している。この手法は、レジスタ・アドレッシングをループ又は繰り返し命令に結合し、これにより、レジスタアドレスに、レジスタファイルのアドレス空間を介して増分するように指示する間に、ロード又は格納命令を繰り返させる。電子回路が、多数の逐次命令のためのコントローラメモリの必要量を低減するために指定される。従って、この従来技術の手法は、特殊なループ又は繰り返し命令によって呼び出されたロード及び格納タイプの動作にのみ適用されるように見える。このように、これは、以下で更に対象とされるような間接V L I W M a n A r r a y（マンアレイ）プロセッサに容易に適用可能ではない。

【0006】

（発明の概要）

M a n A r r a yファミリのプロセッサは適切に、多数の「間接V L I W」（i V L I W）プロセッサ及びプロセッサ要素（P E）からなることができ、これらが32ビットの固定長の短命令語（S I W）を利用する。S I Wを、プロセッサにつき最大8つの実行ユニットの1つによって個別に、かつ、S I M Dモードの動作において、多数のP Eを同期させて実行することができる。もう1つのタイプのS I Wは、V L I Wを間接的に参照して、最大8つのS I W命令の発行を、各プロセッサにおいて並行して、多数のP Eを同期させて、並行して実行させることができる。

【0007】

オペランドがレジスタファイルに格納され、各実行ユニットが1つ又は複数の読取り及び書込みポートを有し、これがレジスタファイル或いは複数ファイルに接続される。大抵のプロセッサでは、各ポート用に選択されたレジスタが、命令におけるビットフィールドを使用してアドレス指定される。Man Arrayプロセッサにおいて使用された間接VLIW技術により、VLIWを作成するSIWがVLIWメモリに格納される。各SIWがレジスタオペランド・フィールドを、レジスタによりアクセスされたオペランドデータにおける単一の動作のための定義によって固定するので、単一のオペランドフィールドが処理アルゴリズムによって必要とされるように異ならなければならないときはいつでも、多数のVLIWが必要とされる。従って、このようなプロセッサ、及び、より一般には並列アレイプロセッサに拡張可能なものと共に使用するための、データのブロックにおける動作のための適切なレジスタファイル索引付け技術が、大変有利となる。

10

【0008】

このオペランド・データ固定レジスタ指定の問題が、本発明によって、間接VLIW命令を使用して、データのブロックにおけるパイプライン計算を達成する小型の手段を提供することにより解決される。レジスタファイルにおけるデータのブロックにアクセスする二重間接方法が使用されて、専用のベクトル処理ハードウェアを使用しない効率的な実施が可能となる。加えて、レジスタ・アドレッシングの自動修正が、単一ベクトル命令にも、繰り返し又はループ命令にも結合されない。むしろこの技術は、レジスタファイル索引付け(RFI)と呼ばれ、ブロックデータ操作機能の制御において十分なプログラムの柔軟性を可能にし、非RFI命令をRFI命令と混合するための能力を提供する。ブロックデータ操作機能がiVLIW Man Arrayアーキテクチャに埋め込まれ、命令セット・アーキテクチャにわたるその汎用の使用を、専用のベクトル命令なしに、かつ、繰り返し又はループ命令と使用するのみに限定されることなく可能にする。本発明を利用すると、連鎖動作が、機能ユニットの間のいかなる直接経路指定もなしに、固有に使用可能であり、さらに実施を簡素化する。加えて、このレジスタファイル索引付けアーキテクチャが、コード化されるアルゴリズムのタイプに応じて特に著しくなる可能性のあるVLIWメモリの必要量を低減する。

20

【0009】

更に、VLIW命令のアンロールされたループ(unrolled loop)として表現されたとき、多数の計算が明確なレジスタ使用パターンを示す。これらのパターンは計算パイプラインの特性であり、本明細書で更に記載され、適合されるように、間接VLIWプロセッサに埋め込まれたMan Array間接ベクトル処理と共に利用することができる。

30

【0010】

その他の態様の中で、本発明は、オペランドレジスタ・アドレスを生成するためのユニークな初期化方法、ユニークな二重間接実行機構、ユニークな制御方法を提供し、レジスタファイルを、独立した循環バッファに区分することができる。これはまた、RFI及び非RFI命令を混合すること、及び、VLIW処理要素の多数のアレイ編成に適用可能なスケーラブルな設計も可能にする。以下で更に詳細に対象とするように、本発明は、iVLIWアレイプロセッサにおける並列命令実行のための、VLIWメモリ、及び、結果として、SIWメモリの必要量を低減する。

40

【0011】

【発明の実施の形態】

本発明のこれら及び他の特徴、態様及び利点は、添付の図面と共に取られた以下の詳細な説明から、当業者には明らかになるであろう。

【0012】

【発明の実施の形態】

(詳細な説明)

本発明と共に使用するために現在好ましいMan Arrayアーキテクチャのさらなる詳細は、1997年6月30日出願の米国特許出願第08/885,310号、1997年10月10日出願の米国特許出願第08/949,122号、1998年10月9日出願

50

の米国特許出願第09/169, 255号、1998年10月9日出願の米国特許出願第09/169, 256号、1998年10月9日出願の米国特許出願第09/169, 072号、1998年11月6日出願の米国特許出願第09/187, 539号、1998年12月4日出願の米国特許出願第09/205, 558号、1998年12月18日出願の米国特許出願第09/215, 081号、1999年1月12日出願の米国特許出願第09/228, 374号、及び1999年1月28日出願の米国特許出願第09/238, 446号、ならびに、1998年7月9日出願の「Methods and Apparatus for Instruction Addressing in Indirect VLIW Processors」という名称の仮特許出願第60/092, 130号、1998年10月9日出願の「Efficient Complex Multiplication and Fast Fourier Transform (FFT) Implementation on the ManArray」という名称の仮特許出願第60/103, 712号、1998年11月3日出願の「Methods and Apparatus for Improved Motion Estimation for Video Encoding」という名称の仮特許出願第60/106, 867号、1998年12月23日出願の「Methods and Apparatus for Providing Direct Memory Access (DMA) Engine」という名称の仮特許出願第60/113, 637号、及び1998年12月23日出願の「Methods and Apparatus Providing Transfer Control」という名称の仮特許出願第60/113, 555号においてそれぞれ判明し、全体として、参照により本明細書に組み込まれる。

【0013】

本発明の現在の好ましい実施形態では、図1Aに示されたManArrayの 2×2 i VLIW単一命令多重データ・ストリーム (SIMD) プロセッサ100が、処理要素0 (PE0) SP/PE0 101に結合されたコントローラ・シーケンス・プロセッサ (SP) を含み、これは「Methods and Apparatus for Dynamically Merging an Array Controller with an Array Processing Element」という名称の米国特許出願第09/169, 072号において、更に詳細に記載されている。本発明によれば、3つの追加のPE151、153及び155も、レジスタファイル索引付け及びそのスケーラブルな性質を示すために利用される。PEに、それらの行列位置により、PE0 (PE00) 101、PE1 (PE01) 151、PE2 (PE10) 153、及びPE3 (PE11) 155について括弧において示されたように、ラベルを付けることもできることに留意されたい。SP/PE0 101はフェッチコントローラ103を含み、32ビット命令メモリ105からの短命令語 (SIW) のフェッチを可能にする。このフェッチコントローラ103が、プログラムカウンタ (PC)、分岐機能、デジタル信号処理ループ動作、割り込みのサポートなど、プログラム可能プロセッサにおいて必要とされた典型的な機能を提供し、命令メモリ管理制御を提供し、これは、アプリケーションによって必要とされた場合、命令キャッシュを含むことができる。加えて、SIW 1-フェッチ・コントローラ103は、32ビットのSIWをシステムにおける他のPEに、32ビット命令バス102によってディスパッチする。

【0014】

この例示的システムでは、説明を簡素化するために共通の要素が全体を通して使用されるが、実際の使用はそのように限定されない。例えば、結合されたSP/PE0 101における実行ユニット131を、制御機能に合わせて最適化された1組の実行ユニット、例えば、固定小数点実行ユニットに分離することができ、PE0、ならびに他のPE151、153及び155を、浮動小数点の応用例に合わせて最適化することができる。この記載のために、実行ユニット131が、SP/PE0及び他のPEにおける同じタイプのものであると仮定される。類似の方法において、SP/PE0及び他のPEが5命令スロット i VLIWアーキテクチャを使用し、これは、超長命令語メモリ (VIM) メモリ109、及び、命令復号化及びVIMコントローラ機能ユニット107を含み、これがSP/PE0の1-フェッチ・ユニット103からディスパッチされたような命令を受信してVIMアドレス及び制御信号108を生成し、これがi VLIWにアクセスするために必要とされ、109における文字SLAMDによって識別され、VIMに格納される。ManArrayパイプライン設計は、分岐待ち時間を増大させることなく、間接VLIWメモリ

10

20

30

40

50

・アクセス機構を提供し、これは、間接実行LVIW(XV)命令のための動的に再構成可能な命令パイプラインを提供することによって行い、これは「Methods and Apparatus to Dynamically Reconfigure the Instruction Pipeline of an Indirect Very Long Instruction Word Scalable Processor」という名称の米国特許出願第09/228,374号において、さらに詳細に記載されている。iVLIWのローディングは、「Methods and Apparatus for Efficient Synchronous MIMD Operations with iVLIW PE-to-PE Communication」という名称の米国特許出願第09/187,539号において、さらに詳細に記載されている。また、SP/PE0及び他のPEに含まれているものは、共通のPE構成可能レジスタファイル127であり、これは、「Methods and Apparatus for Dynamic Instruction Controlled Reconfiguration Register File with Extended Precision」という名称の米国特許出願第09/169,255号において、更に詳細に記載されている。

10

【0015】

SP/PE0の結合された特性により、データメモリ・インターフェイス・コントローラ125が、メモリ121におけるSPデータによりSPコントローラの、及び、メモリ123におけるPE0データによりPE0の、データ処理必要性に対処しなければならない。SP/PE0コントローラ125は、32ビット・ブロードキャスト・データバス126を介して送信されるデータのソースでもある。他のPE151、153及び155は、共通の物理データメモリ・ユニット123'、123"及び123'''を含むが、それらに格納されたデータは一般に、各PEにおいて行われるローカル処理によって必要とされるように異なる。これらのPEデータ・メモリへのインターフェイスも、PE1、2及び3における共通の設計であり、PEローカルメモリ及びデータバス・インターフェイス論理157、157'及び157"によって示される。データ転送通信のためにPEを相互接続するものは、クラスタスイッチ171であり、「Manifold Array Processor」という名称の米国特許出願第08/885,310号、「Methods and Apparatus for Manifold Array Processing」という名称の米国特許出願第09/949,122号、及び「Methods and Apparatus for ManArray PE-to-PE Switch Control」という名称の米国特許出願第09/169,256号において、より完全に記載されている。ホストプロセッサへのインターフェイス、他の周辺デバイス、及び/又は外部メモリを、多数の方法において行うことができる。完全性のために示された主要な機構が、DMA制御装置181に含まれ、これは、スケーラブルなManArrayデータバス183を提供し、これがManArrayコアの外部のデバイス及びインターフェイスユニットに接続される。DMA制御装置181が、これらの外部デバイスがバス185を介してManArrayコアメモリとインターフェイスを取るために必要とされた、データフロー及びバス裁定機構を提供する。

20

30

【0016】

上に示された全ての特許は本発明の譲受人に譲渡され、全体として、参照により本明細書に組み込まれる。

【0017】

このとき、本発明に適合されるようなManArrayプロセッサ装置の特定の詳細を参照すると、この手法は、二重間接機構を介した、効率的かつ柔軟性のあるブロックデータ操作能力を提供するので、有利である。

40

(レジスタファイル索引付けプログラミング考察)

本発明の一態様によるレジスタファイル索引付け(RFI)は、各処理要素における、かつ、オペランドレジスタファイルを、直接SIWのフィールドを介するのではなく、二重間接機構を介して、或いは、専用ベクトル命令及びベクトルハードウェアを介して、或いは、要求された繰り返し又はループ命令により、アドレス指定するためのアレイコントローラにおける、方法及び装置を指す。各実行ユニットが、1つ又は複数のレジスタファイルの読取り及び書込みポートを動作する。読取り又は書込みポートは、レジスタファイルに供給されたレジスタ選択アドレス及び制御線、レジスタファイルから読み取り中であるレジスタデータのための読取りポート用のデータバス、及び、レジスタファイルへ書き込

50

み中であるレジスタデータのための書込みポート用のデータバスからなる。これらのポートのレジスタ選択論理への入力は、典型的には図1Bの従来技術の装置において図示されたように、実行中である命令のビットフィールドからのみ来たものである。図1Bでは、プロセッサの命令レジスタ10において受信された命令が、典型的にはレジスタファイル・アドレスを含み、これらが典型的には、レジスタ12、14及び16など、ポートアドレス・レジスタにおいてラッチされ、次いで、レジスタファイル20など、レジスタファイルをアドレス指定するために直接使用されて、図1Bの格納ユニット22、ロードユニット24、ALU26、MAU28及びDSU30など、ユニットによる命令実行がサポートされた。

【0018】

この典型的なレジスタ選択のための方法に加えて、本発明によるRFI動作により、各実行ユニットの各レジスタファイルポートを、以下で更に対象とされるような簡素な制御回路を使用して、二重間接機構を介して独立して制御することもできる。

【0019】

(RFI動作)

RFI動作を、ManArray iVLIWアーキテクチャに埋め込み、二重間接機構によって呼び出すことができるので、有利である。32ビット符号化フォーマット201を有する例示的実行VLIW(XV)命令200が、図2Aに示される。命令構文、パラメータ又はオペランド、命令200によって実行された動作を要約する構文／動作テーブル203が図2Bに示される。ManArray RFI動作が、ビット20及び21であるRFI動作ビット202を、実行VLIW(XV)命令200において、図2Aに示されたように使用して、RFI動作を使用可能にする。

【0020】

更なる詳細においては、XV命令200が使用されて、指定されたSP又はPE VLIMメモリ(VIM)の個々の命令スロットが間接的に実行させられる。VIMアドレスが、ベースVIMアドレス・レジスタVb(V0又はV1)に、符号なし8ビットのオフセットVIMOFFSを加えた和として計算される。個々の命令スロットのいかなる組み合わせも、実行スロット・パラメータ「E={SLAMD}」を介して実行することができ、但し、S=格納ユニット(SU)、L=ロードユニット(LU)、A=論理演算装置(ALU)、M=乗算累算ユニット(MAU)、及びD=データ選択ユニット(DSU)である。空の「E=」パラメータは、いかなるスロットも実行しない。ユニット影響フラグ(UAF)パラメータ「F=[AMDN]」が、VLIWについて指定されたUAFを、それがロードVLIW(LV)命令を介してロードされたとき、オーバーライドする。オーバーライドは、どの算術命令スロット(A=ALU、M=MAU、D=DSU)又は、なし(N=NONE)が、VLIWのこの実行について条件フラグを設定するために可能とされるかを選択する。オーバーライドは、LV命令を介して指定されたUAF設定に影響を与えない。空の「F=」は、VLIWがロードされたときに、指定されたUAFを選択する。レジスタファイル索引付け(RFI)パラメータ「R=[01N]」が使用されて、このXVの命令スロットの間接実行についてRFIが使用可能或いは使用不可にされる。「R=0」(図2AにおけるRFI動作ビット202=00)であれば、RFI動作が使用可能にされ、RFI制御レジスタグループ0が選択される。「R=1」(ビット202=01)であれば、RFI動作が使用可能にされ、RFI制御レジスタグループ1が選択される。「R=N」(ビット202=11)であれば、RFI動作が使用不可にされる。

【0021】

RFIが使用可能にされたXV命令により、第2の間接動作が開始される。第2の間接動作は、実行される次のXV命令において活動し始め、レジスタポート・アドレスが、分離したRFI制御パラメータによって指定された方法において制御されたハードウェアを自動的に増分することを介して、間接的に指定される。RFI動作が以下で、ManArrayパイプラインに関連して、主として、パイプラインの復号化及び実行段階に関して記

10

20

30

40

50

載される。R F I 制御は、4つの部分、即ち、1) R F I 制御指定、2) R F I 初期化制御、3) R F I 更新制御、及び、4) R F I 命令実行からなる。

【0022】

(R F I 制御指定)

R F I 制御指定は、R F I 制御レジスタを介して実行されることが好ましい。各制御レジスタが、特定の実行ユニットによって使用されたレジスタポートについての全てのR F I 制御情報を指定する。制御フィールドが各ポート用の制御レジスタにあり、このフィールドが、R F I 動作がこの特定のポートについて使用可能にされるかどうかを指定し、使用可能にされた場合、R F I レジスタ更新ポリシーを指定する。

【0023】

R F I 制御レジスタは、図3Aにおいて例示されたM a n A r r a y 雑レジスタファイル(M R F) 300を介してアクセスされる。このレジスタファイルは、追加のレジスタを、制限されたM R F アドレス空間内に、追加のレジスタを単一のM R F アドレスにアドレス対応することによって追加できることにおいて、ユニークである。M R F 拡張レジスタ305及び315は、図3B及び図3Cにおいてそれぞれ示され、M R F 拡張アドレスレジスタ(M R F X A R) 301及びM R F 拡張データレジスタ(M R F X D R) 302及び303を使用してアクセスされる。2つのM R F 拡張データレジスタ302及び303が提供されて、実施が簡素化され、各組の拡張レジスタの所期の使用が分離される。レジスタアドレスが、図4の32ビットM R F X A R レジスタ410のハーフワードH1又はH0部分に、図7A及び図7Bにおいて例示されたようなロード即値命令を使用して書き込まれる。図3A乃至図3C、及び図4A及び図4Bの各部分の関係が、以下でより十分に述べられる。

【0024】

【表1】

10

20

MRFX Addr1 4 0 2 (図 4 A)	MRF 拡張レジスタアドレス-1。このフィールドは、図 3 B の MRF 拡張レジスタ・グループ-1 内のレジスタのアドレスを含む。図 3 A の MRF XDR1 3 0 2 が読み取られるか、あるいは書き込まれたとき、このアドレスによって指定された図 3 B における MRF X1 レジスタが、読取りまたは書込み動作の目標である。
MRFX Addr2 4 0 6 (図 4 A)	MRF 拡張レジスタアドレス-2。このフィールドは、図 3 C の MRF 拡張レジスタ・グループ-2 内のレジスタのアドレスを含む。図 3 A の MRF XDR2 3 0 3 が読み取られるか、あるいは書き込まれたとき、このアドレスによって指定された図 3 C における MRF X2 レジスタが、読取りまたは書込み動作の目標である。
AutoIncrement (AI1 または AI2) 4 0 4 または 4 0 8 (図 4 A)	設定されたとき、このビットも、図 4 A の MRF X アドレス・フィールド1 4 0 2 またはフィールド2 4 0 6 を、図 3 A の MRF XDR1 3 0 2 または MRF XDR2 3 0 3 への各読取りまたは書込みアクセスの後に 1 ずつ増分させる。
MRFX Data (MRFX1 ま たは MRFX2) 4 2 0 (図 4 B)	図 3 A の MRF XDR1 3 0 2 または MRF XDR2 3 0 3 を目標とするロード／格納または DSU 動作 (COPY、BIT op) が、図 4 A の MRF XAR1 4 0 2 のビット [2 : 0] または MRF XAR2 4 0 6 のビット「8 : 6」に含まれるアドレスの MRF X レジスタにアクセスする。選択された MRF XAR の自動増分ビット 4 0 4 または 4 0 8 が設定された場合、このアクセスが、MRF XAR1 または MRF XAR2 におけるアドレスも、アクセス後に増分させる。

10

20

30

【0025】

現在好ましい実施形態では、5つの実行ユニットがRFI制御を有する。図3Cは、例示的なセットのRFI制御レジスタの要約を示す。これらのMRFXレジスタ510、520、530、540、550、560、570及び580が、図5において更に詳細に示され、各制御レジスタが、指定された実行ユニット用の読取り／書込みポートに割り当てられる。これらの実行ユニットには、論理演算装置 (ALU)、乗算累算ユニット (MAU)、データ選択ユニット (DSU)、ロードユニット及び格納ユニットが含まれる。

【0026】

レジスタが2つの制御グループ (510～540)、2つの保存及び復元コンテキストレジスタ (550及び560)、及び1つのレジスタ580において使用されて、各制御グループのためのRFI制御の初期化が制御される。予約済レジスタ570も示される。第1の制御グループ0が、図3CにおけるRFIDL S0 310及びRFIAM0 320を含む。さらなる詳細が、図5のレジスタ510及び520において示される。第2の制御グループ1が、RFIDL S1 330及びRFIAM1 340を含み、さらなる詳細がレジスタ530及び540において示される。

【0027】

iVLIWが実行されたとき、制御グループの1つがXV命令において、ビット21及び20という、図2の命令200のRFIビット202を介して指定されて、このVLIW

40

50

における命令によって使用された、いかなるポートのRFI制御も可能にされる。本発明が、制御情報又は制御情報の部分集合を、直接命令においてなど、指定するための別の機構を使用することを除外しないことは、理解されよう。

【0028】

具体的には、制御グループ0において、図5におけるRFIDLS0510が、単一のロードRtポート501、単一の格納Rsポート502、DSURy503、Rx504及びRt505用の3つのオペランド・ポートのためのポート制御情報を含む。制御グループ0における第2のレジスタ、RFIAM0520が、MAURy521、Rx522、Rt523用の3つのオペランドポート、及び、ALURy524、Rx525及びRt526用の3つのオペランドポートのための、ポート制御情報を含む。2つの制御グループに關連付けられたものは、初期化開始ビットであり、これらは、図3CのRFIStartレジスタ380における、より詳細には図5のレジスタ580における、制御グループ0及び1について含まれる。制御グループ0では、初期化開始ビットがH0ハーフワード581に、以下のように実行ユニットにつき単一のビットと共に位置する。即ち、格納ポート開始0ビットー4 587、ロード・ポート開始0ビットー3 586、ALUポート開始0ビットー2 585、MAUポート開始0ビットー1 584、及び、DSUポート開始0ビットー0 583である。類似の方法において、第2の制御グループ1のための制御レジスタRFIDLS1530、RFIAM1540が、図5に示されるようにセット・アップされる。制御グループ1のための初期化開始ビットは、RFIStart580のH1ハーフワード582に位置する。他の2つのRFIレジスタRFIDLSI550及びRFIAMI560が、コンテキスト保存及び復元動作のサポートにおける割り込みにおいて、ポートアドレス値を格納してポートアドレスの値を保存する。

【0029】

制御パラメータが、必要とされたセットの制御情報を表現することができる、いかなるフォーマットを有する可能性もあり、これは、本発明が特定のフォーマットを必要としないからであることに留意されたい。レジスタファイル・ポートのための例示的フォーマット600が、より詳細に図6に示される。RFIパラメータが4ビットに、列601及び602において示されたように符号化される。この制御情報が、次のRFI命令実行において選択される次のレジスタのアドレスを生成するために適用される、更新のタイプを指定する。現時点の好ましい実施形態では、制御パラメータが使用されて、レジスタアドレスに加算される更新増分値603が選択され、選択することができる最大順次（1ずつ増分する）レジスタファイルアドレス範囲（RFBSS）604が指定される。以下で更に記載されるように、開始レジスタがこれらのパラメータと共に、索引によって選択することができる実際のレジスタセットを決定する。列605～611が使用されて、図8及び図9に示された間接ベクトル装置の動作が記載される。これらの列605～611では、「x」が「配慮しない」状態を表す。列612は、適用可能ユニット列であり、どの実行ユニットに制御パラメータが適用されるかを指定する。

【0030】

（RFI初期化制御）

RFI初期化は2つのステップにおいて行われ、これらは図8及び図9を参照して最適に理解される。図8は、算術演算装置におけるポート論理のための例示的RFI装置800を示す。図9は、ロード及び格納ユニットにおけるポート論理のための例示的RFI装置900を示す。この例示的記載は低コストの構成を表し、これは、ALU、MAU及びDSUユニット用の制御グループ0、及び、ロード及び格納ユニット用の制御グループ0及び1を使用する。これは、図5において概説されたアーキテクチャの記述の部分集合であり、プログラマの制約を表し、全てのオプションが、制御グループ0における全ての実行ユニットについて使用可能であるが、制御グループ1は主としてブロック移動、保存及び復元動作について使用される。RFIXV命令が、第2の制御グループ1を実施において選択し、算術演算装置上の制御グループ0のみを可能とし、算術演算装置は、制御グループ1が指定されたときでも、制御グループ0指定に省略時指定される。この部分集合が

実施の費用を最小化し、以下のようにより詳細に記載される。

【0031】

最初に、各レジスタファイルポートについての、図6に例示されたような制御情報が、R F I 制御レジスタ810及び910に、ロード即値(L I M) 命令700の使用によって書き込まれ、この符号化フォーマットが図7Aに示され、この構文/動作710が図7Bに示される。L I M命令700が最初に使用されて、図4のロードM R F X A R ハーフワードH1 410がロードされて、図3AにおけるM R F X D R 2 303へマップされる所望の拡張R F I 制御レジスタがセットアップされる。次いで、L I M命令がデータ値を所望の制御レジスタへ、M R F X D R 2用のアドレスを使用することによってロードする。制御レジスタの各ハーフワード区間が別々に、L I M命令の定義によってロードされる。

10

【0032】

明瞭にするため、命令レジスタ814 H0ハーフワードビット(15-0)からのL I Mデータパスが図示されない。このデータパスが選択的に制御されて、L I M命令のH0ハーフワードが、図5において列挙されたM R F 拡張レジスタのいずれかの低又は高ハーフワード部分にロードされる。例えば、L I M命令が、図5のR F I A M 0レジスタ520のH1部分への、そのH0ハーフワードのローディングを引き起こすことができる。図8の共通の算術R F I ポート制御論理の参照において、R F I A M 0の3つの制御部分のうち1つが、810など、このポート用の更新制御レジスタ0において、この場合はA L U 852についてロードされる。類似の方法において、他の2つのポート制御値が、それら自体のR F I ポート制御論理に含まれたそれら自体のポート更新制御レジスタ0にロードされる。他のM a n A r r a y 命令がR F I 制御レジスタを、M R F データバス809の使用を介してロードすることができる。M R F データバス809は、R F I ポートレジスタを、例えば、文脈切換え動作中に保存するためにも使用される。特定のL I M命令の記載は、以下の通りである。L I M命令のハーフワード形式が、16ビットの即値(イミディエイト値)をS P 又はP E 目標レジスタR t の上位ハーフワード(H1)又は下位ハーフワード(H0)にロードする。16ビット即値が、符号「中間」値として解釈され、「-32768」から「65535」の範囲におけるいかなる値も受け入れられることを意味する。これは、「-32768」から「+32767」の2の補数符号付き値範囲、及び、「0」から「65535」の符号なし値範囲を包含する。

20

【0033】

L I M命令のワード形式が、符号付き拡張17ビット即値を目標レジスタにロードする。17ビット符号付き値を、「-65536」から「65535」の範囲におけるいかなる値にすることもできる。L I Mのワード形式のための符号化が、値の絶対値をI M M 16フィールドに置き、符号ビットが図7Aに示されたL O C フィールドビット23及び22である。L O C フィールドが、上位ハーフワードが全て1のビットで満たされるか、全て0のビットで満たされるかを決定する。

30

【0034】

R F I 初期化の第2のステップでは、開始ビット、例えば、D S U 854のためのビット583が、R F I 開始レジスタ、図5のR F I S t a r t において設定され、これが、各算術実行ユニットのポートのための開始ビット及びm u x 制御ブロック812、及び、ロード又は格納ユニットのポートのためのブロック912に位置する。各開始ビットが、実行ユニットに属する全てのポートのための初期化を制御する。これが現時点での好ましいフォーマットであるが、本発明はこのフォーマットに制限されない。このビットを設定する動作が、このレジスタに書き込むことができるいかなる命令によっても実行される。このタイプの少なくとも1つの命令が使用可能である。このビットの設定の後、この特定のグループ及び実行ユニットのためのR F I 制御を呼び出す次の命令が、以下で「R F I 命令」と呼ばれ、命令語におけるフィールドによって最初に選択されたその実行ユニットのオペランドレジスタを有し、次いで、このグループ及び実行ユニットのための次のR F I 命令により、図8及び図9に示されたR F I 論理の制御下で選択されたその実行ユニットのオペランドレジスタを有する。R F I X V 命令により、図2A及び図2Bにおいて記

40

50

載されたように、S I WのV L I WセットがローカルのV I M（本明細書で図1 AにおけるようなM a n A r r a y実施のために記載されたような5つのS I W）から間接的に検索される。例えば、5つのS I Wのセットの1つが、図8に示されたような命令レジスタ8 1 4にロードされる。フェッチされたS I W R tのポートのためのポートR F I 論理も、図8に示される。フェッチされた命令の最初の実行では、R tポートアドレス8 1 6がR F I ブロック動作のための開始アドレスである。R tポート・アドレス8 1 6がマルチプレクサ8 1 4を介して、開始ビット及びm u x制御ブロック8 1 2によって制御信号8 1 3を介して制御されるように、ポートアドレス・レジスタ8 1 6へ、マルチプレクサ出力8 1 1を介して渡される。R tポートアドレスは、このとき出力8 1 1において含まれ、復号化パイプライン段階の最後でポートアドレス・レジスタ8 1 6にラッチされる。ポートアドレス・レジスタ8 1 6の出力が直接、レジスタファイル8 1 8を、信号パス8 1 7を介してアドレス指定する。オペランドがレジスタファイル8 1 8から選択され、S I W動作が、指定された実行ユニットにおいて実行される。

10

【0035】

R F I X V命令の次の発行において、オペランドが間接的にR F I 論理から指定される。これは、動作シーケンスにおける第2の間接指定である。第1の間接指定は、R F I X V命令を介したものであり、これは間接的にS I Wを指定し、第2の間接指定は、R F I 制御パラメータを介してセット・アップされたようなR F I 論理を介したものである。これを実施するため、動作更新制御レジスタ0 8 1 0、更新加算器論理8 3 0、索引付きポート・ルック・アヘッド・レジスタ8 2 0、マルチプレクサ8 1 4及び8 2 2、及び

20

【0036】

基本概念は、マルチプレクサ8 1 4のアドレス出力8 1 1が、復号化サイクルにおいて十分早期に使用可能であり、更新加算器論理8 3 0が更新制御論理8 2 4信号に基づいてアドレスを更新できるようにすることである。更新されたアドレス8 1 9がm u x制御信号8 1 5によって選択されてマルチプレクサ8 2 2を通過し、索引ポート・ルック・アヘッド・レジスタ8 2 0に、復号化の最後に、現在のポートアドレス8 1 1がポートアドレス・レジスタ8 1 6にロードされると同時にロードされる。次のR F I 命令において、ルック・アヘッド・レジスタ値8 2 1が、フェッチされたS I Wオペランド・ポート・アドレス値の代りに使用され、次の実行サイクルのためのポートアドレス・レジスタ8 1 6にラッチされ、その間に、更新加算器論理が再度、使用される次のポートアドレスを準備中である。R F I 開始ビットの設定の後に続く最初のR F I 命令の後、開始ビットがクリアされ、後続のR F I 命令に、それらのS I Wオペランドレジスタを、対応する索引付きポート・ルック・アヘッド・レジスタによって選択させるようにする。開始ビット及びm u x制御ブロック8 1 2が、命令のレジスタが命令フィールドによって選択されるか、R F I 索引付きポート・ルック・アヘッド・レジスタによって選択されるかを決定するための制御を提供する。その入力は、命令演算コード8 0 7、更新制御レジスタ0 8 1 0、及びR F I イネーブル信号8 2 5から来る。これらの信号が、パイプラインにおける命令の進行を指示するパイプライン制御信号（図示せず）と共に、レジスタ選択ソースを、マルチプレクサ8 1 4を介して決定する。

30

40

【0037】

索引付きポート・ルック・アヘッド・レジスタ8 2 0の使用により、非R F I 命令を、R F I レジスタアドレス・シーケンスに影響を与えることなく、R F I 動作の間で混合することができる。非R F I 命令が検出されたとき、R F I 論理が、非R F I 命令の実行中に、必要とされたR F I 状態を保持する。

【0038】

（R F I 更新制御）

R F I 動作が呼び出されたとき、レジスタファイル8 1 8における1つ又は複数のレジスタのアドレスが、R F I 論理によって供給される。この論理が、次のサイクルのためのレ

50

ジスタアドレスを、復号化サイクルの早期の段階において使用可能なアドレスから定数を加算或いは減算し、特定のセットのレジスタアドレス内で生成されたポートアドレスを維持することによって、更新する。現在の好ましい実施形態では、これが、増分値、及び、図6に示されたようなレジスタファイル・ブロック・サイズ(RFBS)604を、制御される各ポートについて指定することによって行われる。好ましい実施形態では、RFBS値が、1、2、4、8など、2の整数の累乗であり、論理的に、レジスタファイルを、ブロックにつきRFBS順次アドレス指定されたレジスタを有するレジスタのブロックに区分させる。開始レジスタRs(Rcurrent=最初の更新におけるRs)、RFBS値M、下限の商 $Q = \lfloor Rs/M \rfloor$ 、及び正更新増分kと仮定すると、シーケンスにおける次のレジスタ番号、Rnextが、以下によって与えられる。

【0039】

$R_{next} = ((R_{current} + k) \bmod M) + Q \times M$

Rs/Mの余りが、下限演算により無視されるので、 $Q \times M$ の値 $\neq$ Rsである。

【0040】

一例として、開始レジスタポート・アドレスが「5」であり、即ち、最初の演算のためのRcurrentとも等しいRs=R5であると仮定する。また、更新増分がk=2であり、RFBSがM=8であると仮定する。図6において、この例示的設定が行620に対応し、これは、図8で対応する信号値を以下のように列挙する。即ち、G3=x606及び831、G2=0607及び832、G1=1608及び833、G0=1609及び834、X1=1610及び835、及びX0=0611及び836である。信号X1及びX0が、更新加算器論理830への2入力による増分を提供する。ゲート信号G3、G2、G1、及びG0が、任意の開始レジスタが与えられたブロックサイズを維持する。更新加算器論理830は、5つの標準全加算器861、862、863、864及び865からなり、図10において更に詳細に示される。図10の全加算器1000の実行信号Ci+11005が、更新加算器830の各段階からの実行信号841~844に対応する。これらの実行信号が、ANDゲート845~848、及び、図6の指定された制御記述によって必要とされたモジュロ加算器を効果的に作成するゲート制御信号831~834によってゲートされる。これらの仮定では、RFIを使用してこのポートを指定する連続命令が、レジスタに、以下の順序でアクセスする。即ち、R5、R7、R1、R3、R5、R7などである。開始レジスタがR8であった場合、シーケンスは、R8、R10、R12、R14、R8、R10などである。本発明は、2の累乗でない増分及び/又はRFBSを使用することを除外せず、その内部で動作するためのレジスタアドレス・シーケンスを指定する別の機構を使用することも除外しない。例えば、読取り専用メモリを使用して、更新制御論理824及び更新加算器論理830を置き換えて、望まれたいかなる所望のレジスタポート・アドレス・シーケンスも提供することができる。メモリ・ブロックを使用することで実施配線問題を引き起こす可能性があるので、離散論理において更新機能を実施できるようにすることが、現在好ましい方法である。

【0041】

図9は、ロード及び格納ユニットのためのRFI論理900を示し、これらは2つの制御レジスタグループ910及び950をそれぞれ使用するよう識別されている。XV命令が、どのグループが使用されるかを、図2のビット21~20202を介して指定する。例示的システムでは、制御レジスタグループ1が間接的に指定されたとき、VIMからフェッチされたロード及び格納SIWが、mux制御信号951を介してマルチプレクサ952を通じて選択されたような更新制御レジスタ1950を使用するが、算術演算装置は制御レジスタグループ0を使用することに省略時指定される。代替実施では、図9のRFIポート論理を、各算術実行ユニットについて使用することができ、全ての実行ユニットについて2つのRFIコンテキストを提供する。

【0042】

VLIWプロセッサでは、レジスタファイルの全てのポートを、現在記載されたXV命令など、単一の命令のためのRFI制御下で有することが可能である。RFIポート論理は

10

20

30

40

50

実行ユニットの間で独立しているので、ポートを個別に、S I W実行ユニット特有の命令によって制御することができる。これは、別の命令又は命令のグループが、X V命令に加えて、独立R F I制御（即ち、異なる組の制御パラメータ）を必要とした場合、別の制御レジスタのグループを割り当てることができるという意味である。R F Iセット・アップ待ち時間が相対的に短いので、図5において記載されたような制御レジスタ・セットを、容易に他のR F I命令と共有することができる。

【0043】

別のレジスタファイル索引付け装置1100が、図11に示される。このR F I機構はなお、図8及び図9に関して論じられた他のR F I手法において概説された二重間接機構を使用する。しかし、図11の手法では、プログラミングの制約が強制され、処理中のデータのブロックについて、R F I動作を非R F I動作と混合できないことが必要である。この手法は、図8及び図9において使用された、R F I及び非R F I命令を混合することができる手法とは異なる。いくつかの製品定義では、これは問題ではなく、図11の簡素化されたハードウェア手法を使用することができる。

10

【0044】

図11の装置1100の動作は、先のR F I手法の動作と類似している。例えば、R F I初期化のための開始ビットが、先に記載されたように使用される。図11における主な違いは、図8のレジスタ820のような、索引付きポート・ルック・アヘッド・レジスタが使用されないことである。むしろ、ポート・アドレス・レジスタ1116がなおレジスタファイル1118をアドレス指定するが、更新加算器論理1130動作が時間において変位され、図8において使用された手法と比較されるように、ラッチされたポート・アドレス・レジスタ出力1117において実行サイクル中に動作する。次の実行サイクルのための準備において、更新加算器論理1130がポート・アドレス・レジスタ1116の出力1117を、このポートのためのR F I更新制御レジスタ1110によって指定されたように更新する。この実行サイクルの終了まで、マルチプレクサ1114が制御入力1113を介して制御されて、更新加算器論理出力1119が選択されて、マルチプレクサ1114を介して出力1111に渡される。次いで、マルチプレクサ1114出力1111がポート・アドレス・レジスタ1116において、次の実行サイクルの開始のときにラッチされ、それにより、先にR F I制御セット・アップによって指定されたようなレジスタファイル・ポート・アドレスを更新する。

20

30

【0045】

X V R F I使用可能装置に加えて、R F Iを使用可能にする他の手段が使用される。この追加の機構の目的は、R F I順序付けを、V L I W (X V)プログラミング・モデルにおいてのみ使用されることから切り離すことである。ブロック・ロード、ブロック格納、及びブロック移動動作を単一の命令実行によりサポートし、これを、S Pにおいて独立で、或いは、P Eにおいて同時に行うことができることが望ましい。追加のビットをS I Wにおいて使用してこの動作を指定するよりもむしろ、これは本発明によって除外されないが、R F Iを使用可能にする代替間接機構が使用される。このS I Wにおけるビットの節約は、標準動作符号化のための命令フォーマットのよりよい使用を可能にするが、本発明によって提供されたR F I機能性を達成するための能力を除外しない。この代替機構は、M R Fにおいて特定の場所をアドレス指定することができるいかなるS I Wによっても動作する。M R Fにおける多数の場所をこのために提供することができるが、これを除外する可能性のある、特定の実施における他の使用がある。この代替R F I使用可能機能を記載するため、図3AにおけるR F I L S D 304について示されたような、M R Fにおける1つの場所が使用される。

40

【0046】

このR F I使用可能機構を使用するため、ハードウェア復号化論理が拡張されて、R F Iイネーブル信号が、X V R F I命令が受信されたときだけでなく、ロード、格納又はD S U命令がS P又はP E命令レジスタにおいて受信されたときにも生成され、これがR F I L S Dアドレスを、ロードR t、格納R s又はD S U R t又はR sオペランドとして指定

50

する。この代替 R F I 使用可能機構を使用する前に、R F I 制御レジスタがセットアップされ、ブロック・ロード、格納又は D S U 動作において使用される初期レジスタを指定することが必要とされる。開始ビットは、この代替 R F I 使用可能機構において使用されず、これは、ブロック・シーケンスの開始アドレスがポート制御レジスタに格納されるからである。ロード、格納又は D S U 命令を受信すると、これは R F I L D S ビットをオペランド・アドレスとして使用し、R F I モードが使用可能にされ、各レジスタ・オペランド・アドレスが、事前にセットアップされたポート（オペランド）アドレスで、図 8、図 9 及び図 11 の各 R F I 論理において示されたような R F I ポート論理によって、置換される。R F I 及び非 R F I 動作は、図 8 及び図 9 のハードウェアを使用するときには混合することができる。実際に、ロード、格納及び D S U 制御レジスタ、図 9 に示されたようなグループ 0 及び 1 のために 2 つのコンテキストを使用することによって、次いで、第 1 のデータのブロックにおける R F I X V 動作、第 2 のデータのブロックにおける R F I L S D を使用する R F I 動作、及び非 R F I 動作を混合することができる。算術ポート・レジスタ・オペランド・アドレス範囲の適切な拡張によって、算術命令が、R F I L S D アドレスを参照することによって、R F I を算術命令実行のために呼び出させることができることを理解されたい。

【0047】

（R F I 命令実行）

R F I 動作は、命令語に含まれた制御情報を介して使用可能にされる。この制御情報が使用されて、従来のレジスタアドレス選択フィールド（命令に含まれたオペランドアドレス・フィールド）が使用されるかどうか、或いは、レジスタの R F I 選択が使用されるかどうか指定される。現在好ましい実施形態では、命令における制御情報、図 2 の間接 V L I W X V 命令ビット 21 及び 20 202 が、R F I 動作を制御するために使用される制御レジスタ又はレジスタのセットを間接的に指定する。これらの制御レジスタ・グループの 1 つ又は複数が、図 5 においてみられるように、R F I 制御のために使用可能である。X V R F I 命令が、R F I モードを使用可能にし、R F I 動作を制御するための制御レジスタ・グループを選択する。図 5 に示された R F I 制御レジスタのグループ 510 ～ 580 が、すべてのレジスタ・ポートが R F I 制御されることを可能にし、あらゆる実行ユニットが R F I モードで同時に動作できることを意味する。

【0048】

M a n A r r a y プロセッサがそのパイプラインの実行段階を、レジスタファイルへのライト・バックにより終了することに留意されたい。この手法は、ライト・バック・サイクルの後の次のサイクルが、この結果を次の動作において使用できるようにする。懸命なプログラミングによって、次いで、ベクトル演算の連鎖がアーキテクチャにおいて固有である。分離した迂回パスを、実行ユニットにおいて連鎖をサポートするために設ける必要はない。

【0049】

本発明に係る R F I の例示的使用に関する考察がこのとき提示されて、本発明のいくつかの有利な態様が例示される。増分値 1、R F B S 値（M）2 の累乗、開始レジスタ R 2 と仮定すると、レジスタアドレスが、偶数レジスタ R 2 及びその対応する奇数レジスタ（アドレス+1）R 3 という、2 つのレジスタの間で交替する。R F B S = 4 では、レジスタがサイクルを、増分 1 で 4 つの値の中でアドレス指定する。以下の表は、いくつかのアドレス・シーケンスを示す。

【0050】

【表 2】

開始レジスタ	増分	レジスタファイル・ブロック・サイズ	シーケンス
R 2	1	2	R 2, R 3, R 2, R 3, ...
R 2	1	4	R 2, R 3, R 0, R 1, R 2, ...
R 5	1	4	R 5, R 6, R 7, R 4, R 5, ...
R 5	2	4	R 5, R 7, R 5, R 7, ...
R 5	2	8	R 5, R 7, R 1, R 3, R 5, ...
R 6	2	8	R 6, R 0, R 2, R 4, R 6, ...
R 0	1	1	非ロード／格納ユニットでは、R 0, R 1, R 2, R 3, ..., R 31, R 0, R 1, ... ロード／格納ユニットでは、R 0, R 1, R 2, R 3, ..., R 63 (すべてのレジスタを巡回する)

10

【0051】

20

簡素な行列ベクトル乗算を、図1Aのプロセッサ100など、4-PE SIMD VLIW ManArrayプロセッサにおいて計算することが望まれると仮定する。さらに、以下の命令タイプが使用可能であると仮定する。

【0052】

【表3】

疑似命令	動作
LDB $R_N, P_J +$	ロード・ブロードキャスト：SPメモリにおけるアドレス・レジスタ P_J によって指定された記憶場所からロードし、値を各PEのレジスタ R_N に格納する（全てが同じ値を受信する。 P_J は後に1だけ増分される。）
MAC R_T, R_X, R_Y	乗算累算：全てのPEがSIMD様式において演算 $R_T = R_T + (R_X \times R_Y)$ を実行する
ST $R_S, P_J +$	格納：全てのPEが、ソースレジスタ R_S を、 P_J によって指定されたローカルのPE記憶場所に格納する。 P_J は後に1だけ増分される。
REP N, M	後に続くN個の命令をM回実行する

30

40

【0053】

また、 4×4 行列Aが4つのPEであるPE0、PE1、PE2及びPE3に分配され、以下の表に示されたように、各PEが行列の行を、レジスタR4、R5、R6及びR7において含むようにする（PE0が行0を得る、PE1が行1を得る、など）と仮定する。

【0054】

【表4】

レジスタ	R 4	R 5	R 6	R 7
P E 0	a 00	a 01	a 02	a 03
P E 1	a 10	a 11	a 12	a 13
P E 2	a 20	a 21	a 22	a 23
P E 3	a 30	a 31	a 32	a 33

【0055】

4×1ベクトルのシーケンスがメイン（SP）メモリ105から読み込まれ、行列によって乗算され、結果がローカルのPEメモリ123、123'、123"及び123'''に格納された場合、適切な順序アルゴリズムが、R2が最初にゼロであったと仮定された場合、以下のようになる可能性がある。

10

【0056】

【表5】

LDB R0, P0+	入力ベクトルの第1の要素をロードする、x ₀
MAC R2, R4, R0	積を累算する：a _{i0} *x ₀ （Iは行索引およびPE IDである）
LDB R0, P0+	入力ベクトルの第2の要素をロードする、x ₁
MAC R2, R5, R0	積を累算する：a _{i1} *x ₁
LDB R0, P0+	入力ベクトルの第3の要素をロードする、x ₂
MAC R2, R6, R0	積を累算する：a _{i2} *x ₂
LDB R0, P0+	入力ベクトルの最後の要素をロードする、x ₃
MAC R2, R7, R0	積を累算する：a _{i3} *x ₃
ST R2, P1+	結果を格納する：各ローカル・メモリがその要素を得る；出力ベクトル

20

30

【0057】

このアルゴリズムをVLW命令により実行すると、以下を生じる。

【0058】

【表6】

V L I W	S I W	S I W	実行アクション
	L D B R 0, P 0 +		ロード
1	L D B R 0, P 0 +	M A C R 2, R 4, R 0	P Eをロードし、 $x0 * a[i][0]$ をM A Cする
2	L D B R 0, P 0 +	M A C R 2, R 5, R 0	P Eをロードし、 $x1 * a[i][1]$ をM A Cする
3	L D B R 0, P 0 +	M A C R 2, R 6, R 0	P Eをロードし、 $x2 * a[i][2]$ をM A Cする
4	L D B R 0, P 0 +	M A C R 2, R 7, R 0	P Eをロードし、 $x3 * a[i][3]$ をM A Cする
	S T R 2, P 1 +		全てのP Eが格納結果を格納する

10

20

【0059】

これらのV L I W命令の間の唯一の違いがM A C命令の第2のレジスタ指定であっても、これは、4 V L I Wタイプの命令に加えて、単一のロードL D B及び単一の格納S T命令を必要とする。

【0060】

このとき、この例がR F Iを使用して実行された場合、処理は以下のようになる。すなわち、R 2及びR 0が共にゼロに初期化され、レジスタファイル索引付けが、X V命令によって間接的に実行されたV L I Wに関連付けられた以下のパラメータと共に使用されると仮定する。

30

【0061】

【表7】

実行ユニット・レジスタ・ポート	増分	R F B S
ロード書込みポート	0	1
M A U R x 読取りポート	1	4

【0062】

このとき、コードを小型のV L I W形式において書くことができ、第2のレジスタR F I
シーケンスがR 7→R 4→R 5→R 6→R 7などで開始する。

40

【0063】

【表8】

V L I W	L D R F I C, P 1, ctrl		MAUレジスタ・ポートの ためのR F I 制御を初期化 する
	R E P 1, 5		1つの命令を5回繰り返す
1	L D B R 0, P 0 +	M A C R 2, R 7, R 0	ロードかつM A C する：最 初のM A C が0であり、最 後のロードが次のベクトル (またはガーベッジ) に読 み込む
	S T	R 2, P 1 +	結果を格納する

10

【0064】

最終的な効果は、9個の命令を4個の命令に減らすことである。より少ないV L I Wが使用される事実が、実行されるi V L I Wの数、及び、M a n A r r a yアーキテクチャにおいてロードされなければならないV L I Wの数も減らす。これらの節約は間接的であるが、微々たるものではなく、これは、V L I Wメモリ (V I M) がチップの資源において高価なものに相当するからである。R F I動作が、必要とされるV L I Wメモリの量を減らし、従って、より高価でないチップを可能にする。

20

【0065】

本発明を、現在好ましい実施形態の様々な態様に関連して開示したが、本発明を適切に、後に続く特許請求の範囲に適合する他の環境及び応用例に適用できることは、理解されよう。

【図面の簡単な説明】

【図1A】 本発明と共に使用するために適した、2×2のマンアレイi V L I Wプロセッサを例示する図である。

【図1B】 典型的な従来技術のレジスタ・アドレッシング機構を例示する図である。

【図2A】 本発明に係るR F I使用可能ビットによるX V命令符号化を例示する図である。

30

【図2B】 本発明における使用のために適したX V構文／動作の記述を例示する図である。

【図3A】 R F I制御レジスタの場所を識別するM a n A r r a y 雑レジスタファイル(M R F)を例示する図である。

【図3B】 M R F X 1拡張レジスタを例示する図である。

【図3C】 M R F X 2拡張レジスタを例示し、シーケンスプロセッサ(S P)及び処理要素(P E)において使用されたR F Iレジスタを識別する図である。

【図4A】 拡張レジスタの選択を制御するM R F X A Rレジスタを例示する図である。

【図4B】 M R F X D R 1及びM R F X D R 2のためのデータフォーマットを例示し、R F I制御レジスタが、図4AのM R F X A Rレジスタ値によって指定されたようにマッ
プされる図である。

40

【図5】 本発明と共に使用するための好ましいR F I制御レジスタを例示する図である。

【図6】 各R F Iポートについて使用された例示的な特定の制御符号化を例示する図である。

【図7A】 本発明のR F I制御値をロードするために使用することができる、適切なロード即値(L I M)命令符号化を例示する図である。

【図7B】 L I M構文／動作の記述を例示する図である。

【図8】 本発明に係る算術実行ユニットのための例示的R F I制御ブロック図を例示する図である。

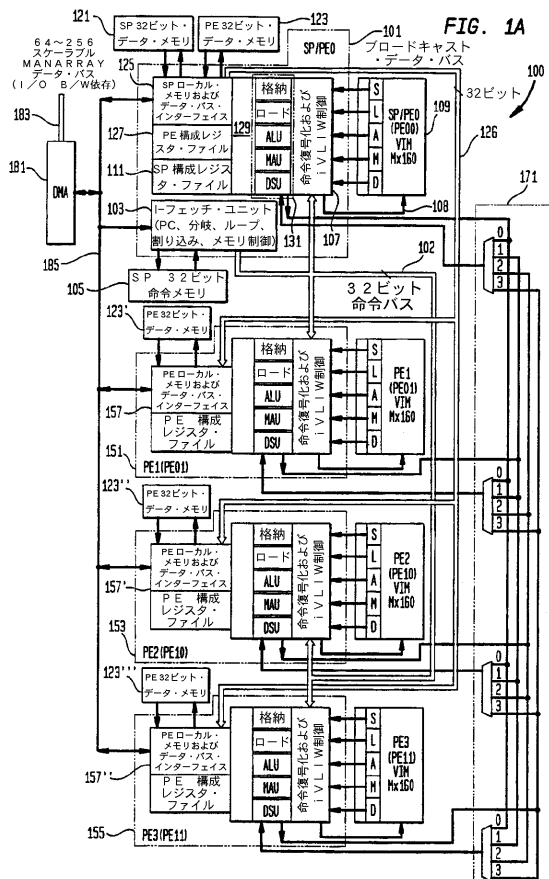
50

【図 9】 本発明に係るロード及び格納実行ユニットのための例示的 R F I 制御ブロック図を例示する図である。

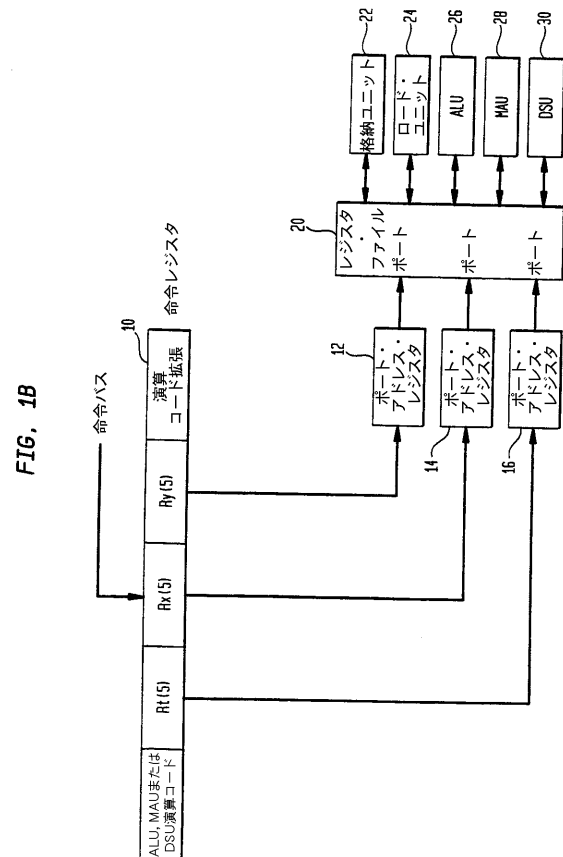
【図 10】 本発明に係る一実施形態における、各 R F I ポート論理における更新加算器論理ユニットにおける使用のための、従来の全加算器を例示する図である。

【図 11】 本発明に係る一実施形態における、算術実行ユニットのためのコスト削減型 R F I 制御ブロック図を例示する図である。

【図 1 A】



【図 1 B】



【図 2 A】

FIG. 2A



【図 2 B】

FIG. 2B

構文／動作	命令	オペランド	動作
XV.[SP]		V[01], VIMOFFS, E=(SLAND), F=(ADRN),[R]	(C=S) の場合、(V [01] + VIMOFFS) [SU] を実行する (C=L) の場合、(V [01] + VIMOFFS) [LU] を実行する (C=A) の場合、(V [01] + VIMOFFS) [ALU] を実行する (C=M) の場合、(V [01] + VIMOFFS) [MAU] を実行する (C=D) の場合、(V [01] + VIMOFFS) [DSU] を実行する (F=またはF=A) の場合、(V [01]+VIMOFFS) [UAF]ALU (F=M) の場合、(V [01] + VIMOFFS) [UAF] MAU (F=D) の場合、(V [01] + VIMOFFS) [UAF] DSU
			T / F 条件が F 0 において満たされた場合にのみ、 動作を行う
[TF], XV.[SP]		V[01], VIMOFFS, E=(SLAND), F=N, [R]	

【図 3 A】

FIG. 3A

SPおよびPEのためのMRFは以下の通り

アドレス (SP(ニーマニック))	PE(ニーマニック)	レジスタ名
10.000	予約済	予約済
10.001	予約済	予約済
11.010	予約済	予約済
11.011	予約済	予約済
11.100	MRFXR0	MRF拡張アドレス・レジスタ(デフォルトは0x00000000)
11.101	MRFXR1	MRF拡張データ・レジスタ(MRFレジスタがここにマップされる)
11.110	MFXR2	MRF拡張データ・レジスタ(MRFレジスタがここにマップされる)
11.111	PF.LSD	ロード、格納、およびDSUブロック動作のためのRFI使用可能アドレス

【図 3 B】

FIG. 3B

MRF拡張(MRFX1)レジスタは以下の通り

MRFX1 アドレス	SP (ニーマニック)	PE (ニーマニック)	レジスタ名
000	予約済	予約済	予約済
001	予約済	予約済	予約済
111	予約済	予約済	予約済

【図 3 C】

FIG. 3C

MRF拡張2(MRFX2)レジスタは以下の通り

MRFX2 アドレス	SP (ニアレジスタ)	レジスタ名
310	RFIDLS0	第1のRFI DSU(HI)、ロード(B0)、および格納(B1)制御レジスタ
320	RFIAM0	第1のRFI ALU(HI)およびMAU(HO)制御レジスタ
330	RFIDLS1	第2のRFI DSU(HI)、ロード(B0)、および格納(B1)制御レジスタ
340	RFIAM1	第2のRFI ALU(HI)およびMAU(HO)制御レジスタ
350	RFIDLS1	DSU(HI)、ロード(B0)、および格納(B1)ポート索引保存/復元コンテキスト
360	RFIAM1	ALU(HI)およびMAU(HO)ポート索引保存/復元コンテキスト
370	予約済	
380	RFI開始	第2の制御グループ開始ビット(HI)および第1の制御グループ開始ビット(HO)

【図 4 B】

FIG. 4B

MRFXDR1またはMRFXDR2

リセット値：未定義

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRFXデータ																															

【図 4 A】

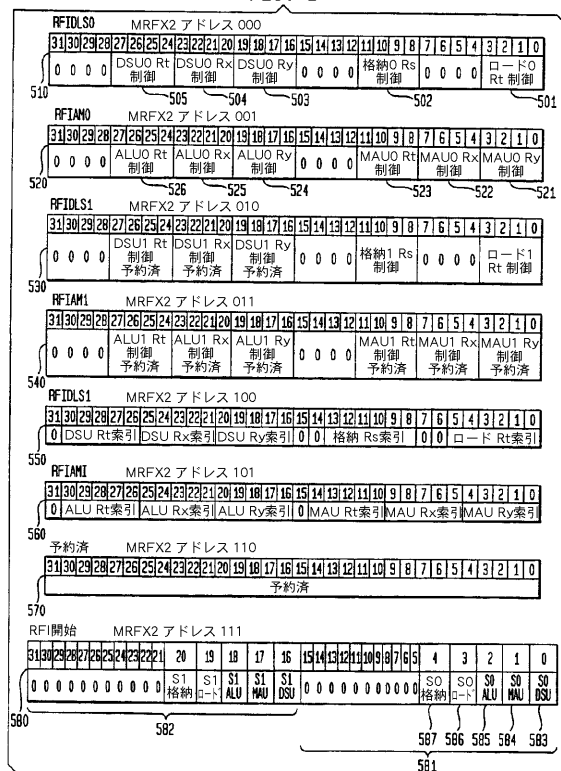
FIG. 4A

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRFXAR																MRFX															
AI																Addr2															
予約済																予約済															
408																404															
																Addr1															
																402															

【図 5】

FIG. 5



【図 6】

FIG. 6

構文/動作		オペランド		動作		ワード	
LIM[SP], W		Rt, IMM17	(MSB(IMM17=1)の場合, RCH1 0xFFFF (MSB(IMM17=0)の場合, RCH1 0x0000 Rt, HO → IMM16				
T.LIM[SP], W		Rt, IMM17	T条件がF0において満たされた場合にのみ、動作を行う				
LIM[SP], H1		Rt, IMM16	Rt, H1 → IMM16				
LIM[SP], HO		Rt, IMM16	Rt, HO → IMM16				
T.LIM[SP], HOH1		Rt, IMM16	T条件がF0において満たされた場合にのみ、動作を行う				

影響を受けた算術フラグ
なし
サイクル: 1

【図 7 B】

FIG. 7B

710

【図 7 A】

FIG. 7A

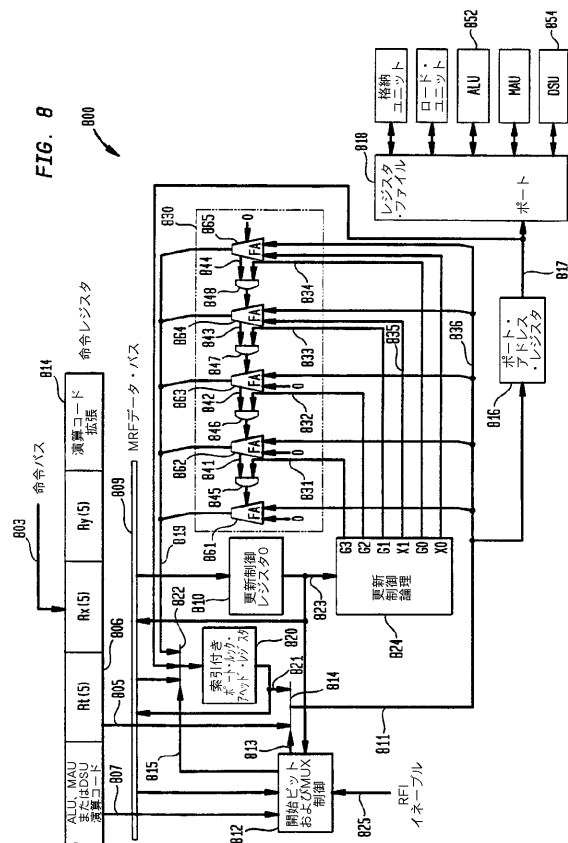
LIM-ロード即値																符号化																LIM命令のためのLOC位置																																																																																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																
データ																ISPL/S																000																PE																LOC																RS																0																IMM16															

700

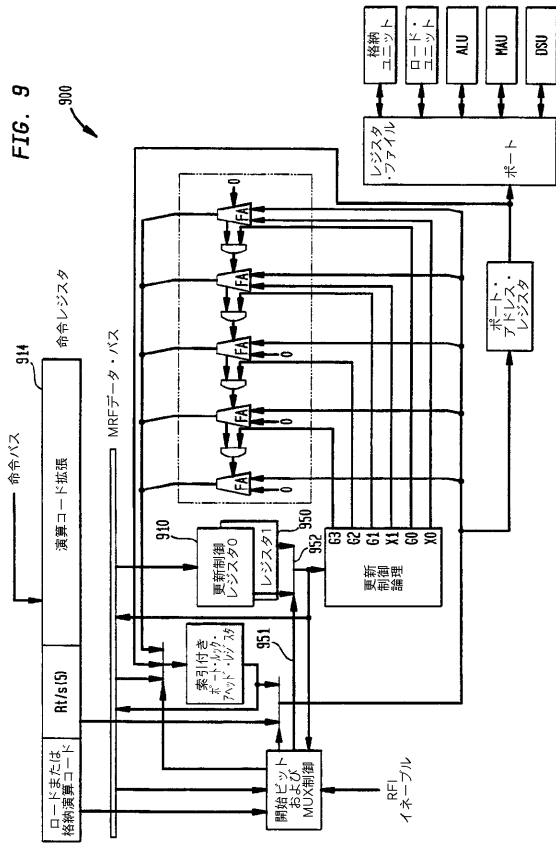
00=H1をロードし、H0に影響を与えない
01=H0をロードし、H1に影響を与えない
10=H0をロードし、H1=0x0000
11=H0をロードし、H1=0xFFFF

【図 8】

FIG. 8

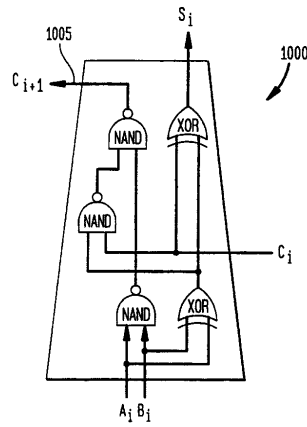


【図 9】

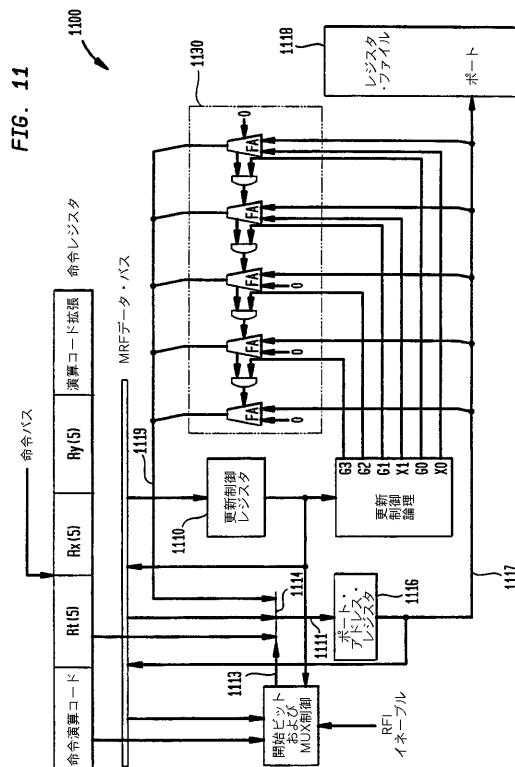


【図 10】

FIG. 10



【図 11】



フロントページの続き

- (72)発明者 バリー, エドウィン, フランク
アメリカ合衆国 ノースカロライナ州 27511, キャリー, ラークホール コート 12
08
- (72)発明者 ペカネック, ジェラルド, ジー.
アメリカ合衆国 ノースカロライナ州 27511, キャリー, ストーンライヒ ドライブ
107
- (72)発明者 マーチャンド, パトリック, アール.
アメリカ合衆国 ノースカロライナ州 27502, エイペックス, キャッスルバーグ ドラ
イブ 2439

審査官 三坂 敏夫

- (56)参考文献 特開平06-259581 (JP, A)
特開平08-249293 (JP, A)
特開昭51-078150 (JP, A)
特開昭58-40668 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06F 9/34