

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3915842号

(P3915842)

(45) 発行日 平成19年5月16日(2007.5.16)

(24) 登録日 平成19年2月16日(2007.2.16)

(51) Int. Cl.	F I
<b>G06F 9/38 (2006.01)</b>	G06F 9/38 310F
	G06F 9/38 310A
	G06F 9/38 370B

請求項の数 2 (全 63 頁)

(21) 出願番号	特願2006-162315 (P2006-162315)	(73) 特許権者	000002369
(22) 出願日	平成18年6月12日(2006.6.12)		セイコーエプソン株式会社
(62) 分割の表示	特願2004-299657 (P2004-299657) の分割	(74) 代理人	110000028 特許業務法人明成国際特許事務所
原出願日	平成4年7月7日(1992.7.7)	(72) 発明者	グエン, レ トロン
(65) 公開番号	特開2006-236396 (P2006-236396A)		アメリカ合衆国 95030 カリフォル ニア州 モンテ セレノ ダニエル プレ イス 15096
(43) 公開日	平成18年9月7日(2006.9.7)	(72) 発明者	レンツ, デレク ジェイ.
審査請求日	平成18年7月12日(2006.7.12)		アメリカ合衆国 95032 カリフォル ニア州 ロス ゲイトス フィリップス アヴェニュー 17400
(31) 優先権主張番号	727, 006		
(32) 優先日	平成3年7月8日(1991.7.8)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 スーパースカラーマイクロプロセッサおよびデータ処理装置

## (57) 【特許請求の範囲】

## 【請求項1】

命令を同時に実行可能な複数の機能ユニットと該機能ユニットにおける前記命令の実行時に使用可能なレジスタとを有するスーパースカラーマイクロプロセッサであって、

少なくともキャッシュメモリを含む読み出し速度の異なる複数種類のメモリに対して、予め定めたプログラム順序に従って、命令のフェッチの要求を出力するフェッチ制御ユニットと、

前記命令のフェッチの要求に伴って、前記少なくともキャッシュメモリを含む読み出し速度の異なる複数種類のメモリに対して、該フェッチの要求に対して識別子を付与して、データのロードまたはストアを出力するデータ制御ユニットと、

前記命令のフェッチの要求に対して、前記読み出し速度の異なる各メモリから読み出される命令を、前記識別子に対応付けて受け取ることにより、該命令の受け取りの順序によらず、該複数の命令を前記プログラム順序に従った命令ストリームとして保持する多段のプリフェッチバッファと、

少なくとも2以上の前記命令を同時にデコードするデコーダと、

該デコードされた複数の命令を、前記複数の機能ユニットに順序外での実行を割り当てるディスパッチ手段と、

該命令の前記機能ユニットに対する割り当てに伴って、該機能ユニットが必要とするデータを、前記データ制御ユニットを介して、前記機能ユニットにロードあるいはストアするロードストアユニットと

10

20

を備えたスーパースカラーマイクロプロセッサ。

【請求項 2】

請求項 1 記載のスーパースカラーマイクロプロセッサと、  
該スーパースカラーマイクロプロセッサが、前記命令の実行に伴って、システムメモリバスを介してデータをアクセスするメモリと  
を備えたデータ処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、RISC型マイクロプロセッサ・アーキテクチャの設計に関し、具体的には 10  
、複数の命令を同時平行に実行することのできるRISCマイクロプロセッサ・アーキテクチャに関する。

なお、以下に列挙した米国特許出願は本件特許出願と同時に米国特許出願され、係属中のものであるが、これらの米国特許出願に開示されており、かつそれぞれ対応して出願された日本での特許出願に開示されている事項は、その出願番号を本明細書で引用することにより本明細書の一部を構成するものとする。

(1) 発明の名称「拡張可能RISCマイクロプロセッサ・アーキテクチャ」(Extensible RISC Microprocessor Architecture) SMOS 7985 MCF/GBR, 米国特許出願第07/727,058号) 1991年7月8日出願、発明者Le T. Nguyen他、およびこれに対応する特願平5-502153号(特表平6-501124号公報)。 20

(2) 発明の名称「アーキテクチャ上の依存関係を隔離したRISCマイクロプロセッサ・アーキテクチャ」(RISC Microprocessor Architecture with isolated Architectural Dependencies) SMOS 7987 MCF/GBR, 米国特許出願第07/726,744号、1991年7月8日出願、発明者Le T. Nguyen他、及びこれに対応する特願平5-502152号(特表平6-502034号公報)。

(3) 発明の名称「複数型レジスタ・セットを採用したRISCマイクロプロセッサ・アーキテクチャ」(RISC Microprocessor Architecture Implementing Multiple Typed Register Sets) SMOS 7988 MCF/GBR/RCC, 米国特許出願第07/726,773号、1991年7月8日出願、発明者Sanjiv Garg他、及びこれに対応する特願平5-502403号(特表平6-501805号公報)。 30

(4) 発明の名称「高速トラップと例外状態をインプリメントしたRISCマイクロプロセッサ・アーキテクチャ」(RISC Microprocessor Architecture Implementing Fast Trap and Exception State) SMOS 7989 MCF/GBR/WSW, 米国特許出願第07/726,942号、1991年7月8日出願、発明者Le T. Nguyen他、及びこれに対応する特願平5-502154号(特表平6-502035号公報)。 40

(5) 発明の名称「シングル・チップ・ページ・プリンタ・コントローラ」(Single Chip Page Printer Controller) SMOS 7991 MCF/GBR, 米国特許出願第07/726,929号、1991年7月8日出願、発明者Derek J. Lentz他、及びこれに対応する特願平5-502149号(特表平6-501586号公報)。

(6) 発明の名称「複数の異種プロセッサをサポートすることのできるマイクロプロセッサ・アーキテクチャ」(Microprocessor Architecture Capable of Supporting Multiple Heterogeneous Processors) SMOS 7992 MCF/WMB, 米国特許出願第07/726,893号、1991年7月8日出願、 50

発明者 Derek J. Lentz 他、及びこれに対応する特願平5-502151号(特表平6-501123号公報)。

【0002】

本明細書の記述は本件出願の優先権の基礎たる米国特許出願07/727,066号の明細書の記載に基づくものであって、当該米国特許出願の番号を参照することによって、当該米国特許出願の明細書の記載内容が本明細書の一部を構成するものとする。

【背景技術】

【0003】

近年、マイクロプロセッサ・アーキテクチャの設計は複合命令セット・コンピュータ(CISC-Complex Instruction Set Computer)アーキテクチャを採用したものから、より単純化された縮小命令セット・コンピュータ(RISC-Reduced Instruction Set Computer)アーキテクチャを採用したものまでに発達している。CISCアーキテクチャは大部分がハードウェアで命令実行パイプラインを実現し、サポートしていることを特徴としている。従来のパイプライン構造の代表的なものは、命令フェッチ、命令デコード、データ・ロード、命令実行、データ・ストアのステージからなり、これらの順序は固定している。命令の組の異なる部分をパイプラインのそれぞれのステージを通して同時平行に実行すると、パフォーマンス上の利点を得られる。パイプラインを長くすると、利用できる実行ステージの数が増加し、同時平行に実行できる命令数が増加する。

【0004】

CISCパイプライン・アーキテクチャの効率を制約する一般的問題として、2つある。最初の問題は、先に実行される条件コード設定命令がパイプラインを通して実質的に実行を完了するまで、条件付ブランチ命令が正しく評価できないことである。従って、そのあとに続く条件付命令の実行が遅延または停止(stall)されるので、いくつかのパイプライン・ステージが数プロセッサ・サイクルの間インアクティブ(inactive)、すなわち不動作のままになっている。代表例として条件コードは実行ステージを通してある命令の処理が完了したときだけ、プロセッサ状況レジスタ(PSR)とも呼ばれる条件コード・レジスタに書かれている。そのために、ブランチ条件コードが判断されるまでの数プロセッサ・サイクルの間、条件付きブランチ命令をデコード・ステージにおいたままパイプラインを停止させなければならない。パイプラインが停止すると、スループットの損失が大になる。さらにコンピュータの平均スループットはプログラム命令ストリームの中で条件付きブランチ命令が条件コード設定命令のあとに近接して現れる頻度によって左右される。

【0005】

第2の問題は、プログラム命令ストリームの中で近接して置かれている命令がプロセッサ・レジスタ・ファイルの同じレジスタを参照する傾向があることから起こる問題である。データ・レジスタは、連続する命令のストア・ステージとロード・ステージにおいてデータの宛先、またはソースとして頻繁に使用されている。一般的にデータをレジスタ・ファイルにストアする命令が少なくとも1つの実行ステージを通して処理を完了してからでなければ、後続命令のロード・ステージ処理でレジスタ・ファイルをアクセスすることができないようになっている。多数の命令を実行するには、ストア・データを得るために1実行ステージで複数のプロセッサ・サイクルを必要とするので、実行ステージのオペレーションが続いている間、パイプライン全体が停止しているのが代表的である。その結果、コンピュータの実行スループットは、命令ストリームが実行される順序に左右されることになる。

【0006】

第3の問題は、命令自体の実行から起こる問題ではなく、マイクロプロセッサ自体のハードウェア・サポートによる命令実行環境の維持、すなわち、マシンの状態(state-of-machine)から起こる問題である。現在のCISCマイクロプロセッサのハードウェア・サブシステムは、命令の実行中にトラップ条件が現れると、それを検出で

10

20

30

40

50

きるようになっている。各トラップを処理するには、対応するトラップ処理ルーチンをプロセッサに実行させる必要がある。トラップが検出されたとき、実行パイプラインをクリアして、トラップ処理ルーチンが即時に実行できるようにする必要がある。同時にトラップが現れた正確な個所で、そのときのマシンの状態を設定しなければならない。この正確な個所は、そのとき実行中の最初の命令が割り込みとトラップのために完了したときと、例外のために実行されなかった命令の直前に現れる。そのあと、マシンの状態と、この場合も、トラップの内容に応じて実行中の命令自体を処理ルーチンの完了時に復元しなければならない。その結果、各トラップまたは関連事象（イベント）が起これば、処理ルーチンの開始時と終了時にパイプラインをクリアし、正確なマシンの状態をセーブし、復元するために待ち時間が生じ、プロセッサのスループットがその分だけ減少することになる。

10

## 【 0 0 0 7 】

CISCアーキテクチャが潜在的にもつスループットを向上するために、これらの問題に対する解決方法がいろいろと試みられている。条件付きブランチ命令が正しく実行されると想定すれば、ブランチ条件コードの最終的判断に先立って、パイプライン実行を試行的に進めることが可能である。また、レジスタが修正されると想定すれば、後続の命令を試行的に実行することが可能である。処理ルーチンの実行を必要とするような例外の発生を最小にすることによって、プログラム命令ストリームの処理に割り込みをかける例外の発生頻度を少なくすることを、別のハードウェアで行うことが可能である。

## 【 0 0 0 8 】

これらの解決方法は、明らかに追加ハードウェアを複雑化するものではあるが、その方法自身にも別の問題がある。つまり、ブランチ条件コードの最終的判断またはレジスタ・ファイルのストア・アクセスに先立って命令の実行を続けるためには、条件付きブランチのロケーションを含むプログラム命令ストリーム内の複数の個所のいずれかにレジスタ・ファイルの各々の修正内容に、及び例外が発生した場合には、最後の複数の命令の実行が完了した以前の個所にマシンの状態が復元可能であることが必要である。その結果、さらに別のサポート・ハードウェアが必要になり、しかも、いずれかのパイプライン・ステージのサイクル・タイムが大幅に増加しないように、ハードウェアを設計しなければならない。

20

## 【 0 0 0 9 】

RISCアーキテクチャでは、マイクロプロセッサ・アーキテクチャのハードウェアによる実現を大幅に単純化することによって、上記の問題の多くを解決することを試みている。極端な場合には、各RISC命令はロード・サイクル、実行サイクル、及びストア・サイクルからなる3つのパイプライン化プログラム・サイクルだけで実行される。ロード及びストア・データをバイパスすることによって、従来のRISCアーキテクチャは3ステージ・パイプラインにおいてサイクルあたり1命令の実行を可能にすることを基本にしている。

30

## 【 0 0 1 0 】

可能な限り、RISCアーキテクチャにおけるハードウェア・サポートは最小化され、必要とする機能はソフトウェア・ルーチンで実行するようにしている。その結果、RISCアーキテクチャは、最適に適合したパイプラインで実行される単純なロード/ストア命令セットの使用により大幅な柔軟性と高速化が期待できる。また、実際にはRISCアーキテクチャは短い高性能パイプラインと増加した命令数を実行する必要性との調和を図ると、必要とするすべての機能を実現できることが判明している。

40

## 【 0 0 1 1 】

RISCアーキテクチャの設計は一般的に、ブランチ、レジスタ参照及び例外に関してCISCアーキテクチャに起こっている問題を回避し、あるいは最小化するようになっている。RISCアーキテクチャにおけるパイプラインは短く、スピードが最適化されている。パイプラインを短くすると、パイプライン停止（stall）またはクリアによって生じる結果を最小化するとともに、マシンの状態を以前の実行個所に復元する際に起こる問題を最小化することができる。

50

## 【 0 0 1 2 】

しかし、一般に認識されている現水準を大幅に超えたスループット・パフォーマンスの向上は、従来のRISCアーキテクチャによっては容易に達成することができない。その結果、これに変わるスーパースcaler (super-scaler) と呼ばれるアーキテクチャが種々提案されている。これらのアーキテクチャは、一般的には、複数の命令を同時並行に実行することによって、プロセッサのスループットを比例的に向上させることを試みている。残念ながら、この種のアーキテクチャの場合もCISCアーキテクチャの問題と同じでないとしても、条件付きブランチ、レジスタ参照、及び例外処理に同じような問題が起こっている。

## 【 発明の開示 】

## 【 発明が解決しようとする課題 】

## 【 0 0 1 3 】

したがって、本発明の一般的目的は従来のCISCアーキテクチャとRISCアーキテクチャに対して大幅なパフォーマンス向上が得られ、さらにマイクロプロセッサで実現するのに適した高性能の、RISCベースのスーパースcaler型プロセッサ・アーキテクチャを提供することである。

## 【 課題を解決するための手段 】

## 【 0 0 1 4 】

この目的を達成するために、本発明によるマイクロプロセッサ・アーキテクチャは命令ストアからフェッチした命令を同時並行に実行することを可能にしている。このマイクロ  
20  
プロセッサ・アーキテクチャは命令ストアから命令セットをフェッチするための命令プリフェッチ・ユニットを備えている。各命令セットは複数の固定長命令から構成されている。命令FIFOは第1バッファと第2バッファを含む複数の命令セット・バッファに命令セットを置いておくバッファリングのために用意されたものである。命令実行ユニットは、レジスタ・ファイルと複数の機能ユニット (functional unit) から構成され、第1バッファと第2バッファに置かれている命令セットを調べ、使用可能な機能  
30  
ユニットに実行させるためにこれらの命令の1つを出す機能を持つ命令制御ユニットを備えている。機能ユニットとレジスタ・ファイル間は複数のデータ経路 (data

path) で結ばれているので、それぞれの命令を並行実行するために必要とされるレジスタ・ファイルへの複数の独立アクセスが可能になっている。

## 【 0 0 1 5 】

レジスタ・ファイルはレジスタ・データを一時的にストアしておくために使用される別  
35  
セットのデータ・レジスタを含んでいる。これらの一時データ・レジスタは、命令が順序外 (out-of-order) で実行される際に機能ユニットによって処理されたデータを受け入れるために命令実行ユニットによって利用される。一時データ・レジスタにストアされたデータは選択的に保持され、そのあと先行するすべての順序内 (in-order) 命令の実行が完了して退避された、命令ストリーム内の命令ロケーションまで正確なマシン状態が進んだときクリアされるか、レジスタ・ファイルに退避される。

## 【 0 0 1 6 】

最後にメモリからの命令の組のプリフェッチは、メイン・プログラム命令ストリーム、  
40  
ターゲット条件付きブランチ命令ストリーム及びプロシージャ命令ストリームのプリフェッチを可能にする複数のプリフェッチ経路によって容易化されている。ターゲット条件付きブランチ・プリフェッチ経路を利用すると、条件付きブランチ命令となり得る両方の命令ストリーム、つまりメインとターゲットを同時にプリフェッチすることが可能である。プロシージャ命令プリフェッチ経路を利用すると、メインまたはターゲット命令ストリームにある一つの命令を実行する拡張プロシージャの実行を可能にする上で効果的な補足的命令ストリームを可能にする。また、プロシージャ・プリフェッチ経路によると、少なくともメイン・プリフェッチ・バッファをクリアすることなく、これらの拡張プロシージャをフェッチして実行することができる。

## 【 0 0 1 7 】

10

20

30

40

50

以上のとおり、本発明の利点は、基本的にRISC型のコア・アーキテクチャを利用して非常に高性能なスループットを実現するアーキテクチャを提供することにある。

本発明の別の利点は、サイクルごとに複数の命令の実行を可能にしたことにある。

さらに、本発明の利点は、複数の命令を同時並行に実行することを最適化するために必要な機能ユニットを動的に(ダイナミック)に選択して、利用することを可能にしたことである。さらに本発明の別の利点は、正確なマシン状態復帰機能をサポートするメカニズムと一体化したレジスタ・ファイル・ユニットを設けたことにある。

#### 【0018】

さらに、本発明の別の利点は、レジスタ・ファイル・ユニット内に複数のレジスタ・ファイルの内蔵し、これらのレジスタ・ファイルは汎用化され、タイプ化され、複数の独立並列整数レジスタ・ファイルとしてのオペレーション、浮動小数点ファイルと整数ファイルとしてのレジスタ・ファイルのオペレーションおよび専用ブール・レジスタ・ファイルのオペレーションを含む、複数のレジスタ・ファイル機能を備えていることにある。

#### 【0019】

さらに、本発明の別の利点は、ロードとストア・オペレーション及び例外と割り込みの処理を効率的な命令キャンセル・メカニズムやロード/ストア順序シンクロナイザを含む、正確なマシン状態復帰機能の使用により正確に実行できるようにしたことである。さらに、本発明の別の利点は、専用レジスタ・ファイル・ユニットによってトラップ状態をサポートして待ち時間を最小化し、処理スループットを向上したことにある。

#### 【0020】

さらに、本発明の別の利点は、メイン及びターゲット・ブランチ・プリフェッチ待ち行列を設け、それによって正確でないターゲット・ブランチ・ストリームが先に実行されても、本発明によって得られる全体的な処理スループットに与える影響を最小化したことにある。さらに、プロシージャ命令プリフェッチ待ち行列が設けられているので、メインまたはターゲット・ブランチ命令ストリームの実行に効率的に割り込んで、プロシージャ・ルーチンの実行を通して新規命令を実現することを可能にし、特にプロシージャ・ルーチンを外部から修正することによって、組み込みプロシージャ命令を実現することを可能にしたことである。

#### 【発明を実施するための最良の形態】

#### 【0021】

以下、本発明の実施の形態について説明する。なお、以下の目次に従って順次説明する。

#### 【0022】

#### 目次

#### I . マイクロプロセッサ・アーキテクチャの概要

#### II . 命令フェッチユニット

##### A) IFUデータ経路

##### B) IFU制御経路

##### C) IFU/IEU制御インタフェース

##### D) PCロジック・ユニットの詳細

##### 1) PFおよびExPC制御/データ・ユニットの詳細

##### 2) PC制御アルゴリズムの詳細

##### E) 割り込みおよび例外の処理

##### 1) 概要

##### 2) 非同期割り込み

##### 3) 同期例外

##### 4) ハンドラ・ディスパッチとリターン

##### 5) ネスト

##### 6) トラップ一覧表

#### III . 命令実行ユニット

10

20

30

40

50

A) I E Uデータ経路の詳細	
1) レジスタ・ファイルの詳細	
2) 整数データ経路の詳細	
3) 浮動小数点データ経路の詳細	
4) ブール・レジスタ・データ経路の詳細	
B) ロード/ストア制御ユニット	
C) I E U制御経路の詳細	
1) Eデコード・ユニットの詳細	
2) キャリヤ・チェッカ・ユニットの詳細	
3) データ依存関係チェッカ・ユニットの詳細	10
4) レジスタ改名ユニットの詳細	
5) 命令発行ユニットの詳細	
6) 完了制御ユニットの詳細	
7) リタイア制御ユニットの詳細	
8) 制御フロー制御ユニットの詳細	
9) バイパス制御ユニットの詳細	
I V . 仮想メモリ制御ユニット	
V . キャッシュ制御ユニット	
V I . 要約及び結論	
以下に目次に従って説明する。	20
【 0 0 2 3 】	
I . マイクロプロセッサ・アーキテクチャの概要	
図 1 は本発明のアーキテクチャ 1 0 0 の概要を示すものである。命令フェッチ・ユニット ( I F U ) 1 0 2 と命令実行ユニット ( I E U ) 1 0 4 はアーキテクチャ 1 0 0 の中心となる機能要素である。仮想メモリ・ユニット ( V M U ) 1 0 8 、キャッシュ制御ユニット ( C U U ) 1 0 6 、およびメモリ制御ユニット ( M C U ) 1 1 0 は、 I F U 1 0 2 と I E U 1 0 4 の機能を直接にサポートするためのものである。また、メモリ・アレイ・ユニット ( M A U ) 1 1 2 は基本的要素として、アーキテクチャ 1 0 0 を動作させるためのものである。もっとも M A U 1 1 2 はアーキテクチャ 1 0 0 の一つの一体的なコンポーネントとして直接的に存在しない。つまり、本発明の好適実施例では I F U 1 0 2 、 I E U 1 0 4 、 V M U 1 0 8 、 C C U 1 0 6 および M C U 1 1 0 は従来の 0 . 8 ミクロン設計ルールの低電力 C M O S プロセスを利用してシングル・シリコン・チップ上に実装され、約 1 , 2 0 0 , 0 0 0 個のトランジスタから構成されている。アーキテクチャ 1 0 0 の標準プロセッサまたはシステムのクロック速度は 4 0 M H z である。しかし、本発明の好適実施例によれば、プロセッサの内部クロック速度は 1 6 0 M H z である。	30
【 0 0 2 4 】	
I F U 1 0 2 の基本的役割は命令をフェッチし、 I E U 1 0 4 による実行が保留されている間命令をバッファに置いておき、一般的には次の命令をフェッチするとき使用される次の仮想アドレスを計算することである。	
【 0 0 2 5 】	40
本発明の好適実施例では、各命令は長さが 3 2 ビットに固定されている。命令の組、つまり、4 個の命令からなる「バケット」 ( b u c k e t ) は C C U 1 0 6 内の命令用キャッシュ 1 3 2 から 1 2 8 ビット幅の命令バス 1 1 4 を経由して I F U 1 0 2 によって同時にフェッチされる。命令の組の転送は、制御ライン 1 1 6 経由で送られてきた制御信号によって調整されて、 I F U 1 0 2 と C C U 1 0 6 間で行われる。フェッチされる命令の組の仮想アドレスは I F U 仲裁、制御およびアドレスを兼ねたバス 1 1 8 経由で I F U 1 0 2 から出力され、さらに I E U 1 0 4 と V M U 1 0 8 間を結合する仲裁、制御およびアドレス共用バス 1 2 0 上に送出される。 V M U 1 0 8 へのアクセスの仲裁 ( a r b i t r a t i o n ) は I F U 1 0 2 と I E U 1 0 4 の両方が V M U 1 0 8 を共通の共用資源として利用することから行われる。本発明の好適実施例では、仮想アドレスの物理ページ内のア	50

ドレスを定義する下位ビットは、IFU102から制御ライン116を経由して直接にキャッシュ制御ユニット106へ転送される。IFU102から与えられる仮想アドレスの仮想上位ビットは、バス118、120のアドレス部分によってVMU108へ送られ、そこで対応する物理ページ・アドレスに変換される。IFU102では、この物理ページ・アドレスは変換要求がVMU108に出されたあと内部プロセッサ・クロック・サイクルの1/2の間に、VMU108からアドレス制御ライン122経由で直接にキャッシュ制御ユニット106へ転送される。

#### 【0026】

IFU102によってフェッチされた命令ストリームの方は命令ストリーム・バス124経由でIEU104に渡される。制御信号は、制御ライン126を介してIFU102とIEU104間でやりとりされる。さらに、ある種の命令フェッチ・アドレス、例えばIEU104内に存在するレジスタ・ファイルへアクセスを必要とするアドレスは、制御ライン126内のターゲット・アドレス・リターン・バスを経由してIFU102へ送り返される。

10

#### 【0027】

IEU104は、CCU106内に設けられたデータ用キャッシュ134との間で80ビット幅双方向データ・バス130を通してデータをストアし、データを取り出す。IEU104がデータ・アクセスするときの物理アドレス全体は制御バス128のアドレス部分によってCCU106へ渡される。また、制御バス128を通して、データ転送を管理するための制御信号をIEU104とCCU106との間でやりとりすることもできる。IEU104は、仮想データ・アドレスをCCU106へ渡すのに適した物理データ・アドレスに変更するための資源としてVMU108を使用する。データ・アドレスの仮想化部分は、仲裁、制御およびアドレス・バス120を経由してVMU108へ渡される。IFU102に対するオペレーションと異なり、VMU108は対応する物理アドレスをバス120経由でIEU104へ返却する。アーキテクチャ100の好適実施例では、IEU104は物理アドレスを使用して、ロード/ストア・オペレーションが正しいプログラム・ストリーム順序で行われていることを確かめている。

20

#### 【0028】

CCU106は、物理アドレスで定義したデータ要求を命令用キャッシュ132とデータ用キャッシュ134のどちらか該当する方から満足できるかどうかを判断する従来のハイレベル機能を備えている。アクセス要求が命令用キャッシュ132またはデータ用キャッシュ134へアクセスすることで正しく満足できる場合は、CCU106はデータ・バス114、128を経由するデータ転送を調整して、その転送を行う。

30

#### 【0029】

データ・アクセス要求が命令用キャッシュ132またはデータ用キャッシュ134から満足できない場合は、CCU106は対応する物理アドレスをMCU110へ渡し、MAU112が、要求しているのは読取りアクセスであるか書込みアクセスであるかを判別し、各要求ごとにCCU106のソースまたは宛先キャッシュ132、134を識別するのに十分な制御情報および要求オペレーションをIFU102またはIEU104から出された最終的データ要求と関係づけるための追加識別情報も一緒にMCU110へ渡される。

40

#### 【0030】

MCU110は、好ましくはポート・スイッチ・ユニット142を備えており、このユニットは単方向データ・バス136によってCCU106との命令用キャッシュ132に接続され、双方向データ・バス138によってデータ用キャッシュ134に接続されている。ポート・スイッチ142は基本的には大きなマルチプレクサであり、制御バス140から得た物理アドレスを複数のポートP0 Pn 146<sub>0..n</sub>のいずれかへ送ることを可能にし、また、ポートからデータ・バス136、138へのデータの双方向転送を可能にする。MCU110によって処理される各メモリ・アクセス要求は、MAU112をアクセスするとき要求されるメイン・システム・メモリ・バス162へのアクセスを仲裁する目的

50

でポート146<sub>0-n</sub>の1つと関連づけられる。データ転送の接続が確立されると、MCUは制御情報を制御バス140経由でCCU106に渡して、ポート141およびポート146<sub>0-n</sub>のうち対応する1つを経由して命令用キャッシュ132またはデータ用キャッシュ134とMAU112との間でデータを転送することを開始する。アーキテクチャ100の好適実施例では、MCU110は、実際にはCCU106とMAU112間を転送する途中にあるデータをストアまたはラッチしない。このようにしたのは、転送の待ち時間を最小にし、MCU110に一つだけ存在するデータを追跡または管理しないですむようにするためである。

#### 【0031】

##### II. 命令フェッチ・ユニット

命令フェッチ・ユニット102の主要エレメントを図2に示す。これらのエレメントのオペレーションおよび相互関係を理解しやすくするために、以下ではこれらのエレメントがIFUデータ経路と制御経路に参与する場合を考慮して説明する。

#### 【0032】

##### A) IFUデータ経路

IFUデータ経路は、命令の組を受け取ってプリフェッチ・バッファ260に一時的にストアしておく命令バス114から始まる。プリフェッチ・バッファ260からの命令の組はIデコード・ユニット262を通してIFIFOユニット264へ渡される。命令IFIFO264の最後の2ステージにストアされた命令の組は、データ・バス278、280を通してIEU104に連続的に取り出して利用することができる。

#### 【0033】

プリフェッチ・バッファ・ユニット260は一度に1つの命令の組を命令バス114から受け取る。完全な128ビット幅の命令の組は、一般にプリフェッチ・バッファ260のメイン・バッファ(MBUF)188部分の4つの128ビット幅プリフェッチ・バッファ・ロケーションの1つに並列に書き込まれる。追加の命令の組は最高4つまで同じように、2つの128ビット幅ターゲット・バッファ(TBUF)190のプリフェッチ・バッファ・ロケーションに、または2つの128ビット幅プロシージャ・バッファ(EBUF)192のプリフェッチ・バッファ・ロケーションに書き込むことが可能である。好適アーキテクチャ100では、MBUF188、TBUF190またはEBUF192内のプリフェッチ・バッファ・ロケーションのいずれかに置かれている命令の組は、プリフェッチ・バッファ出力バス196へ転送することが可能である。さらに、直接フォールスルー(fall through)命令セット・バス194は、命令バス114をプリフェッチ・バッファ出力バス196と直接に接続することによってMBUF188、TBUF190およびEBUF192をバイパスするためのものである。

#### 【0034】

好適アーキテクチャ100では、MBUF188は名目的またはメイン命令ストリーム中の命令の組をバッファするために利用される。TBUF190は、試行的なターゲット・ブランチ命令ストリームからプリフェッチした命令の組をバッファするために利用される。その結果、プリフェッチ・バッファ・ユニット260を通して、条件付きブランチ命令のあとに置かれている可能性のある両方の命令ストリームをプリフェッチすることができる。この機能により、MAU112の待ち時間は長くなるとしても、少なくともCCU112への以後のアクセス待ち時間がなくなるので、条件付きブランチ命令の解決時にどの命令ストリームが最終的に選択されるかに関係なく、条件付きブランチ命令のあとに置かれた正しい次の命令の組を得て、実行することができる。本発明の好適アーキテクチャ100では、MBUF188とTBUF190があるために、命令フェッチ・ユニット102は現れる可能性のある両方の命令ストリームをプリフェッチすることができ、命令実行ユニット104に関連して以下に説明するようにただしいと想定された命令ストリームを引き続き実行することができる。条件付きブランチ命令が解決されたとき、正しい命令ストリームがプリフェッチされて、MBUF188に入れられた場合は、TBUF190に残っている命令の組は無効にされるだけである。他方、正しい命令ストリームの命令の

10

20

30

40

50

組が T B U F 1 9 0 に存在する場合は、命令プリフェッチ・バッファ・ユニット 2 6 0 を通して、これらの命令の組が T B U F 1 9 0 から直接に並行に M B U F 1 8 8 内のそれぞれのバッファ・ロケーションへ転送される。それ以前に M B U F 1 8 8 にストアされた命令の組は、T B U F 1 9 0 から転送された命令の組を重ね書きすることによって事実上無効にされる。M B U F ロケーションへ転送する T B U F 命令の組が無ければ、そのロケーションには無効の印が付けられるだけである。

**【 0 0 3 5 】**

同様に、E B U F 1 9 2 はプリフェッチ・バッファ 2 6 0 を経由する別の代替プリフェッチ経路となるものである。E B U F 1 9 2 は好ましくは M B U F 1 8 8 命令ストリームに現れた単一の命令、つまり、「プロシージャ」命令で指定されたオペレーションを実現するために使用される代替命令ストリームをプリフェッチする際に利用される。このようにすると、複雑な命令や拡張された命令はソフトウェア・ルーチンまたはプロシージャを通して実現することができ、すでにプリフェッチされて M B U F 1 8 8 に入れられた命令ストリームを乱すことなくプリフェッチ・バッファ・ユニット 2 6 0 を通して処理することができる。一般的には、本発明によれば、最初に T B U F 1 9 0 に現れたプロシージャ命令を処理することができるが、プロシージャ命令ストリームのプリフェッチは保留され、以前に現れた保留中の条件付きブランチ命令ストリームが全て解決される。これにより、プロシージャ命令ストリームに現れた条件付きブランチ命令は、T B U F 1 9 0 の使用を通して矛盾なく処理されることになる。従って、プロシージャ・ストリームでブランチが行われる場合は、ターゲットとなる命令の組はすでにプリフェッチされて T B U F 1 9 0 に入れられているので、E B U F 1 9 2 へ並列に転送することができる。

**【 0 0 3 6 】**

最後に M B U F 1 8 8、T B U F 1 9 0 および E B U F 1 9 2 の各々はプリフェッチ・バッファ出力バス 1 9 6 に接続され、プリフェッチ・ユニットによってストアされた命令の組を出力バス 1 9 6 上に送出的ようになっている。さらに、バス 1 9 4 を通過するフローは命令の組を命令バス 1 1 4 から直接に出力バス 1 9 6 へ転送するためのものである。

**【 0 0 3 7 】**

好適アーキテクチャ 1 0 0 では M B U F 1 8 8、T B U F 1 9 0、E B U F 1 9 2 内のプリフェッチ・バッファは直接的には F I F O 構造を構成していない。その代わりにどのバッファ・ロケーションも出力バス 1 9 6 に接続されているので、命令用キャッシュ 1 3 2 から取り出された命令の組のプリフェッチ順序に大幅な自由度をもたせることができる。つまり、命令フェッチ・ユニット 1 0 2 は命令ストリームに一定順序で並んだ命令順に命令の組を判断して要求するのが一般的になっている。しかし、命令の組が I F U 1 0 2 へ返されるときの順序は、要求したある命令の組が使用可能で、C C U 1 0 6 だけからアクセス可能であり、他の命令の組は M A U 1 0 2 のアクセスを必要とするような場合に合わせて順序外に現れることも可能である。

**【 0 0 3 8 】**

命令の組は一定順序でプリフェッチ・バッファ・ユニット 2 6 0 へ返されない場合があっても、出力バス 1 9 6 上に出力される命令の組の列は、一般的に I F U 1 0 2 から出された命令の組要求の順序に従っていなければならない。順序内 ( i n - o r d e r ) の命令ストリーム列は、例えばターゲット・ブランチ・ストリームの試行的実行に影響されるためである。

**【 0 0 3 9 】**

I デコード・ユニット 2 6 2 は、I F I F O ユニット 2 6 4 のスペースが許す限り、プリフェッチ・バッファ出力バス 1 9 6 から命令の組を、普通は 1 サイクルに 1 つの割合で受け取る。一つの命令の組を構成する 4 個の命令からなる各セットは I デコード・ユニット 2 6 2 によって並行にデコードされる。関係の制御フロー情報が I F U 1 0 2 の制御経路部分のためにライン 3 1 8 から抜き出されている間は、命令の組の内容は I デコード・ユニット 2 6 2 によって変更されない。

10

20

30

40

50

## 【 0 0 4 0 】

Iデコード・ユニット262からの命令の組はI F I F Oユニット264の128ビット幅入力バス198上に送出される。内部的には、I F I F Oユニット264はマスタ/スレーブ・レジスタ200、204、208、212、216、220、224の列から構成されている。各レジスタはその後続レジスタに接続され、マスタ・レジスタ200、208、216の内容がF I F Oオペレーションの内部プロセッサ・サイクルの前半時にスレーブ・レジスタ204、212、220へ転送され、そのあとオペレーションの後半サイクル時に次の後続マスタ・レジスタ208、216、224へ転送されるようになっている。入力バス198はマスタ・レジスタ200、208、216、224の各々の入力に接続され、F I F Oオペレーションの後半サイクル時に命令の組がIデコード・ユニ

10

## 【 0 0 4 1 】

マスタ/スレーブ・レジスタ200、204、208、212、216、220、224の各々は、128ビット幅の命令の組の全ビットを並列にストアできるほかに、制御情報のいくつかのビットをそれぞれの制御レジスタ202、206、210、214、218、222、226にストアすることもできる。好ましくは、制御ビットのセットは、例外不一致(exception miss)と例外修正(exception modify)(VMU)、メモリなし(MCU)、プランチ・パイアス、ストリーム、およびオフセット(IFU)からなっている。この制御情報はI F I F Oマスタ・レジスタに入力バス198から新しい命令の組をロードすると同時に、I F U 1 0 2の制御経路部分から発生する。そのあと、制御レジスタ情報は命令の組と並行してI F I F Oユニット264内で並列にシフトされる。

20

## 【 0 0 4 2 】

最後に好適アーキテクチャ100ではI F I F Oユニット264からの命令の組の出力は最後の2マスタ・レジスタ216、224から同時に得られて、I\_\_B u c k e t\_\_0とI\_\_B u c k e t\_\_1命令セット出力バス278、280上に送出される。さらに、対応する制御レジスタ情報がI B A S V 0とI B A S B 1制御フィールドバス282、284上に送出される。これらの出力バス278、282、280、284は全てI E U 1 0 4へ通じる命令ストリーム・バス124となるものである。

30

## 【 0 0 4 3 】

## B) I F U 制御経路

I F U 1 0 2 制御経路は、プリフェッチ・バッファ・ユニット260、Iデコード・ユニット262およびI F I F Oユニット264のオペレーションを直接にサポートする。プリフェッチ制御ロジック・ユニット266は主にプリフェッチ・バッファ・ユニット260のオペレーションを管理する。プリフェッチ制御ロジック・ユニット266とI F U 1 0 2 は一般的にはクロック・ライン290からシステム・クロック信号を受信して、I F U のオペレーションとI E U 1 0 4、C C U 1 0 6 およびV M U 1 0 8 のオペレーションとの同期をとるようにしている。命令の組を選択して、M B U F 1 8 8、T B U F 1 9 0 およびE B U F 1 9 2 に書き込むための制御信号は制御ライン304上に送出される。

40

## 【 0 0 4 4 】

多数の制御信号は、制御ライン316上に送出されて、プリフェッチ制御ロジック・ユニット266へ送られる。具体的には、フェッチ要求制御信号はプリフェッチ・オペレーションを開始するために送出される。制御ライン316上に送出される他の制御信号は要求したプリフェッチ・オペレーションが目標とする宛先かM B U F 1 8 8 であるか、T B U F 1 9 0 であるか、E B U F 1 9 2 であるかを指定している。プリフェッチ要求を受け

50

て、プリフェッチ制御ロジック・ユニット266はID値を生成し、プリフェッチ要求をCCU106に通知できるかどうかを判断する。ID値の生成は、循環4ビット・カウンタを使用して行われる。

【0045】

4ビット・カウンタの使用は次の3つの点で重要である。第1は最大9個までの命令の組をプリフェッチ・バッファ・ユニット260で一度にアクティブにできることである。すなわち、MBUF188での4つの命令の組、TBUF190での2つの命令の組、EBUF192での2つの命令の組、およびフロー・スルー・バス194経由で直接にIDコード・ユニット262に渡される1つの命令の組である。第2は、命令の組が各々4バイトの4個の命令からなることである。その結果、フェッチする命令を選択するどのアドレスも、その最下位4ビットは余分になっている。最後は、プリフェッチ要求アドレスの最下位4ビットとして挿入することで、プリフェッチ要求IDをプリフェッチ要求と容易に関連づけることができることである。これにより、CCU106とのインターフェースとなるために必要な総アドレス数が減少することになる。

【0046】

IFU102から出されたプリフェッチ要求の順序に対して順序外で命令の組がCCU106から返却されるようにするために、アーキテクチャ100ではCCU106からの命令の組の返却と一緒にID要求値が返されるようになっている。しかし、順序外の命令の組返却機能によると、16個の固有IDが使いつくされるおそれがある。条件付き命令の組合せが順序外で実行されると、要求されたがまだ返却されていない追加のプリフェッチと命令の組があるので、ID値を再使用することが可能になる。したがって、4ビット・カウンタは保持しておくのが好ましく、それ以降の命令の組のプリフェッチ要求が出されないことになり、その場合には次のID値は、未処理のまま残っているフェッチ要求やそのときプリフェッチ・バッファ260に保留されている別の命令の組に関連づけられたものとなる。

【0047】

プリフェッチ制御ロジック・ユニット266はプリフェッチ状況配列(アレイ)268を直接に管理し、この配列はMBUF188、TBUF190およびEBUF192内の各命令セット・プリフェッチ・バッファ・ロケーションに論理的に対応する状況記憶ロケーションからなっている。プリフェッチ制御ロジック・ユニット266は選択およびデータ・ライン306を通して、データをスキャンし、読み取って、状況レジスタ配列268に書き込むことができる。配列268内では、メイン・バッファ・レジスタ308は4個の4ビットID値(MB ID)、4個の1ビット予約フラグ(MB RES)および4個の1ビット有効フラグ(MB VAL)をストアしておくためのものであり、これらの各々は論理ビット位置別にMBUF180内のそれぞれの命令の組の記憶ロケーションに対応づけられている。同様に、ターゲット・バッファ・レジスタ310と拡張バッファ・レジスタ312は、それぞれ2個の4ビットID値(TB ID、EB

ID)、2個の1ビット予約フラグ(TB RES、EB RES)および2個の1ビット有効フラグ(TB VAL、EB VAL)をストアしておくためのものである。最後にフロー・スルー状況レジスタ314は1個の4ビットID値(FT ID)、1個の予約フラグ・ビット(FT RES)および1個の有効フラグ・ビット(FT VAL)をストアしておくためのものである。

【0048】

状況レジスタ配列268が最初にスキャンされ、該当するときは、プリフェッチ要求がCCU106に出されるたびにプリフェッチ制御ロジック・ユニット266によって更新され、そのあとは命令の組が返されるたびにスキャンされ、更新される。具体的に説明すると、制御ライン316からプリフェッチ要求信号を受け取ると、プリフェッチ制御ロジック・ユニット266は現在の循環カウンタ生成ID値をインクリメントし、状況レジスタ配列268をスキャンして、使用可能なID値があるかどうか、プリフェッチ要求信号で指定されたタイプのプリフェッチ・バッファ・ロケーションが使用可能であるかどうか

10

20

30

40

50

を判断し、CCU I B U S Y制御ライン300の状態を調べてCCU106がプリフェッチ要求を受け付けることができるかどうかを判断し、受付可能ならば、制御ライン298上のCCU I R E A D制御信号を肯定し、インクリメントされたID値をCCU106と結ばれたCCU I D出力バス294上に出送する。プリフェッチ記憶ロケーションは、対応する予約状況フラグと有効状況フラグが共に偽である場合に使用が可能である。

【0049】

プリフェッチIDは要求がCCU106に出されるのと並行して、M B U F 1 8 8、T B U F 1 9 0、またはE B U F 1 9 2内の目標とする記憶ロケーションに対応する状況レジスタ配列268内のID記憶ロケーションに書き込まれる。さらに、対応する予約状況フラグが真にセットされる。

10

【0050】

CCU106が以前に要求された命令の組をI F U 1 0 2へ返却できるときは、CCU I R E A D Y信号が制御ライン302上で肯定され、対応する命令の組のIDがCCU I D制御ライン296上に出送される。プリフェッチ制御ロジック・ユニット266は状況レジスタ配列268内のID値と予約フラグをスキャンして、プリフェッチ・バッファ・ユニット260内の命令の組の目標とする宛先を判別する。一致するものは一つだけが可能である。判別されると、命令の組はバス114を経由してプリフェッチ・バッファ・ユニット260内の該当ロケーションに書き込まれ、フロー・スルー要求と判別されたときは、直接にIデコード・ユニット262に渡される。どちらの場合も、対応する状況レジスタ配列に入っている有効状況フラグは真にセットされる。

20

【0051】

PCロジック・ユニット270は、以下で詳しく説明するように、I F U 1 0 2全体を調べてM B U F 1 8 8、T B U F 1 9 0およびE B U F 1 9 2命令ストリームの仮想アドレスを探し出す。この機能を実行する際、PCロジック・ブロック270はIデコード・ユニット262を制御すると同時に、そこから動作する。具体的には、Iデコード・ユニット262によってデコードされ、プログラムの命令ストリームのフローの変化と関わりがある可能性のある命令部分はバス318を経由して制御フロー検出ユニット274へ送られると共に、直接にPCロジック・ブロック270へ送られる。制御フロー検出ユニット274は条件付きブランチ命令と無条件ブランチ命令、コール型命令、ソフトウェア・トラップ・プロシージャ命令および種々のリターン命令を含む制御フロー命令を構成する各命令をデコードされた命令の組の中から判別する。制御フロー検出ユニット274は制御信号をライン322を経由して、PCロジック・ユニット270へ送る。この制御信号は、Iデコード・ユニット262に存在する命令の組の内の制御フロー命令のロケーションと種類を示している。これを受けて、PCロジック・ユニット270は、一般的には命令に入れられて、ライン318経由でPCロジック・ユニットへ転送されたデータから制御フロー命令のターゲット・アドレスを判断する。例えば、条件付きブランチ命令に対して先に実行するためにブランチ・ロジック・バイアスが選択された場合には、PCロジック・ユニット270は条件付きブランチ命令ターゲット・アドレスから命令の組をプリフェッチすることを指示し、別々に追跡することを開始する。したがって、制御ライン316上のプリフェッチ要求を次に肯定すると、PCロジック・ユニット270はさらにライン316を経由する制御信号を肯定し、先行してプリフェッチされた命令の組がM B U F 1 8 8またはE B U F 1 9 2へ送られたものと想定すると、プリフェッチの宛先をT B U F 1 9 0として選択する。プリフェッチ要求をCCU106へ渡すことができるとプリフェッチ制御ロジック・ユニット266が判断すると、プリフェッチ制御ロジック・ユニット266は、この場合もライン316を経由してイネーブル(許可)信号をPCロジック・ユニット270へ送って、ターゲット・アドレスのページ・オフセット部分(C C U P A D D R [ 1 3 : 4 ])をアドレス・ライン324を経由して直接にCCU106へ渡すことを可能にする。これと同時に、PCロジック・ユニット270は新しい仮想ページから物理ページへの変換が必要な場合には、さらに、V M U要求信号を制御ライン328を経由して、ターゲット・アドレスの仮想化部分(V M U V A D D R [ 1 3 : 1 4 ])

30

40

50

をアドレス・ライン326を經由してVMU108へ渡して、物理アドレスに変換する。ページ変換が必要でない場合は、VMU108によるオペレーションは必要でない。その代わりに、以前の変換結果がバス122に接続された出力ラッチに保存されているので、CCU106によって即時に使用される。

**【0052】**

PCロジック・ユニット270が要求した仮想から物理への変換時にVMU108にオペレーション・エラーが起これば、VMU例外およびVMU不一致制御(miss control)ライン332、334を通して報告される。VMU不一致制御ライン334は変換索引緩衝機構(Translation lookaside buffer: TLB)の不一致を報告する。VMU例外ライン332上のVMU例外制御信号は他の例外が起これば発生する。いずれの場合も、PCロジック・ユニットは命令ストリーム中の現在の実行個所をストアしておき、そのあと無条件ブランチが行われたのと同じように、それを受けてエラー条件を診断し処理するための専用例外処理ルーチン命令ストリームをプリフェッチすることによって、エラー条件を処理する。VMU例外および不一致制御信号は、発生した例外の種類を示しているので、PCロジック・ユニット270は対応する例外処理ルーチンのプリフェッチ・アドレスを判別することができる。

10

**【0053】**

IFIFO制御ロジック・ユニット272はIFIFOユニット264を直接にサポートするためのものである。具体的には、PCロジック・ユニット270は制御ライン336を經由して制御信号を出力し、命令の組がIデコード・ユニット262から入力バス198を經由して使用可能であることをIFIFO制御ロジック・ユニット272に通知する。IFIFO制御ユニット272は命令の組を受け取るために、最も奥の使用可能なマスタ・レジスタ200、208、216、224を選択する役割をもっている。マスタ・レジスタ202、210、218、226の各々の出力は制御バス338を經由してIFIFO制御ユニット272へ渡される。各マスタ制御レジスタによってストアされる制御ビットは2ビット・バッファ・アドレス(IF\_\_Bx\_\_ADR)、単一ストリーム・インジケータ・ビット(IF\_\_Bx\_\_STRM)、および単一有効ビット(IF\_\_Bx\_\_VLD)からなっている。2ビット・バッファ・アドレスは対応する命令の組の内の最初の有効命令を指定している。つまり、CCU106から返された命令の組は、例えば、ブランチ・オペレーションのターゲット命令が命令の組の内の最初の命令ロケーションに置かれるように境界合わせされていないことがある。したがって、バッファ・アドレス値は、実行の対象として考慮される、命令の組の内の最初の命令を一意的に示すために与えられる。

20

30

**【0054】**

ストリーム・ビットは、条件付き制御フロー命令を含んでいる命令の組のロケーションを示し、IFIFOユニット264を通る命令のストリームに潜在的制御フローの変更を引き起こすマーカとして使用されることを基本としている。メイン命令ストリームは一般にストリーム・ビット値が0のときMBUF188を通して処理される。例えば、相対条件付きブランチ命令が現れるとすると、対応する命令の組はマークが付けられ、ストリーム・ビット値が1となる。条件付き命令の組はIデコード・ユニット262によって検出される。条件付き制御フロー命令は最高4つまで命令の組に存在することができる。そのあと、命令の組はIFIFOユニット264の最も奥の使用可能なマスタ・レジスタにストアされる。

40

**【0055】**

条件付きブランチ命令のターゲット・アドレスを判断するために、現在のIEU104の実行点アドレス(DPC)、ストリーム・ビットで指定された条件付き命令が入っている命令の組の相対ロケーション、制御フロー検出ユニット274から得られた命令の組の内の条件付き命令ロケーション・オフセットは、制御ライン318を通して対応するブランチ命令フィールドから得られた相対ブランチ・オフセット値と結合される。その結果はブランチ・ターゲットの仮想アドレスとなり、PCロジック・ユニット270によってストアされる。ターゲット命令ストリームの最初の命令の組は、このアドレスを使用してプ

50

リフェッチしてT B U F 1 9 0に入れることができる。P C ロジック・ユニット2 7 0のために事前に選択されたブランチ・バイアスに応じてI F I F O ユニット2 6 4 はM B U F 1 8 8 またはT B U F 1 9 0 からロードが続けられる。1 つまたは2 つ以上の条件付フロー命令を含んでいる2 番目の命令の組が現れると、その命令の組はストリーム・ビット値に0 のマークが付けられる。2 番目のターゲット・ストリームはフェッチできないので、ターゲット・アドレスはP C ロジック・ユニット2 7 0 によって計算されてストアされるが、プリフェッチは行われぬ。さらに、それ以降の命令の組はI デコード・ユニット2 6 2 を通して処理することができない。少なくとも、条件付きフロー制御命令を含んでいることが分かった命令の組は1 つも処理されない。

**【0056】**

本発明の好適実施例では、P C ロジック・ユニット2 7 0 は最高2 個までの命令の組に現れた条件付きフロー命令を最高8 個まで管理することができる。ストリーム・ビットの変化でマークが付けられた2 命令の組の各々のターゲット・アドレスは4 つのアドレス・レジスタの配列にストアされ、ターゲット・アドレスは命令の組の内の対応する条件付きフロー命令のロケーションに対して論理的位置に置かれる。

**【0057】**

最初の順序内条件付きフロー命令のブランチ結果が解決されると、P C ロジック・ユニット2 7 0 は、ブランチが行われる場合にはT B U F 1 9 0 の内容をM V U F 1 8 8 に転送し、T B U F 1 9 0 の内容に無効のマークを付けるように、ライン3 1 6 上の制御信号によってプリフェッチ制御ユニット2 6 6 に指示する。正しくない命令ストリーム、つまりブランチが行われない場合はターゲット・ストリームからの、ブランチが行われる場合はメイン・ストリームからの命令の組が、I F I F O ユニット2 6 4 にあるとI F I F O ユニット2 6 4 からクリアされる。2 番目またはそれ以降の条件付きフロー制御命令が第1 ストリーム・ビットのマークが付けられた命令の組に存在すると、その命令は統一された方法で処理される。すなわち、ターゲット・ストリームからの命令の組はプリフェッチされ、M B U F 1 8 8 またはT B U F 1 9 0 からの命令の組はブランチ・バイアスに応じてI デコード・ユニット2 6 2 を通して処理され、条件付きフロー命令が最終的に解決されると、正しくないストリーム命令の組がI F I F O ユニット2 6 4 からクリアされる。

**【0058】**

I F I F O ユニット2 6 4 から正しくないストリーム命令がクリアされたとき、2 番目の条件付きフロー命令がI F I F O ユニット2 6 4 に残っていて、最初の条件付きフロー命令の組にそれ以降の条件付きフロー命令が含まれていないと、第2 ストリーム・ビットのマークが付いた命令の組のターゲット・アドレスはアドレス・レジスタの最初の配列にプロモートされる。いずれの場合も、条件付きフロー命令を含んでいる次の命令の組はI デコード・ユニット2 6 2 を通して評価することが可能になる。したがって、ストリーム・ビットをトグルとして使用すると、ブランチ・ターゲット・アドレスを計算する目的のために、また、ブランチ・バイアスが特定の条件付きフロー制御命令では正しくなかったと、後で判断された場合に、それより上をクリアすべき命令の組のロケーションにマークをつける目的のために、潜在的制御フローの変化にマークをつけておき、I F I F O ユニット2 6 4 を通して追跡することができる。

**【0059】**

命令の組をマスタ・レジスタから実際にクリアするのではなく、I F I F O 制御ロジック・ユニット2 7 2 はI F I F O ユニット2 6 4 の対応するマスタ・レジスタの制御レジスタに入っている有効ビット・フラグをリセットするだけである。このクリア・オペレーションはライン3 3 6 に送出される制御信号でP C ロジック・ユニット2 7 0 によって開始される。マスタ制御レジスタ2 0 2、2 1 0、2 1 8、2 2 6 の各々の入力状況バス2 3 0 を通してI F I F O 制御ロジック・ユニット2 7 2 が直接にアクセスすることができる。好適実施例のアーキテクチャ1 0 0 ではこれらのマスタ制御レジスタ2 0 2、2 1 0、2 1 8、2 2 6 内のビットはI F I F O ユニット2 6 4 によるデータ・シフト・オペレーションと並行して、または独立してI F I F O 制御ユニット2 7 2 によってセットす

10

20

30

40

50

ることが可能である。この機能により、IEU104のオペレーションと非同期に、命令の組をマスタ・レジスタ200、208、216、224のいずれかに書込み、対応する状況情報をマスタ制御レジスタ202、210、218、226に書き込むことができる。

#### 【0060】

最後に、制御および状況バス230上の追加の制御ラインはIFIFOユニット264のIFIFOオペレーションを可能にし、指示する。IFIFOシフトは制御ライン336を通してPCロジック・ユニット270から出力されたシフト要求制御信号を受けてIFIFOユニット264によって行われる。IFIFO制御ユニット272は、命令の組を受け入れるマスタ・レジスタ200、208、216、224が使用可能であると、制御信号をライン316を経由してプリフェッチ制御ユニット266に送ってプリフェッチ・バッファ260から次の該当命令の組を転送することを要求する。命令の組が転送されると、配列268内の対応する有効ビットがリセットされる。

10

#### 【0061】

##### C) IFU/IEU制御インタフェース

IFU102とIEU104とを結ぶ制御インタフェースは制御バス126によって提供される。この制御バス126はPCロジック・ユニット270に接続され、複数の制御、アドレスおよび特殊データ・ラインから構成されている。割り込み要求と受信確認制御信号を制御ライン340を経由して渡すことにより、IFU102は割り込みオペレーションを通知し、IEU104との同期をとることができる。外部で発生した割り込み信号はライン292経由でロジック・ユニット270へ送られる。これを受けて、割り込み要求制御信号がライン340上に送出されると、IEU104は試行的に実行された命令をキャンセルする。割り込みの内容に関する情報は、割り込み情報ライン341を通してやりとりされる。IEU104がPCロジック・ユニット270によって判断された割り込みサービス・ルーチンのアドレスからプリフェッチされた命令の受信を開始する準備状態になると、IEU104はライン340上の割り込み受信確認制御信号を肯定する。IFU102によってプリフェッチされた割り込みサービス・ルーチンがそのあと開始される。

20

#### 【0062】

IFIFO読取り(IFIFORD)制御信号はIEU104から出力され、最も奥のマスタ・レジスタ224に存在する命令の組が実行を完了したことおよび次の命令の組が必要であることを通知する。この制御信号を受けると、PCロジック・ユニット270はIFIFOユニット264でIFIFOシフト・オペレーションを実行するようにIFIFO制御ロジック・ユニット272に指示する。

30

#### 【0063】

PCインクリメント要求とサイズ値(PCINC/SIZE)は制御ライン344上に送出されて、現在のプログラム・カウンタ値を命令の対応するサイズ数だけ更新するようにPCロジック・ユニット270に指示する。これによりPCロジック・ユニット270は、現在のプログラム命令ストリーム中の最初の順序内実行命令のロケーションを正確に指した個所に実行プログラム・カウンタ(DPC)を維持することができる。

40

#### 【0064】

ターゲット・アドレス(TARGET ADDR)はアドレス・ライン346を経由してPCロジック・ユニット270に返される。このターゲット・アドレスはIEU104のレジスタ・ファイルにストアされているデータによって決まるブランチ命令の仮想ターゲット・アドレスである。したがって、ターゲット・アドレスを計算するためにIEU104のオペレーションが必要である。

#### 【0065】

制御フロー結果(CFRESULT)制御信号は制御ライン348を経由して、PCロジック・ユニット270へ送られて、現在保留されている条件付きブランチ命令が解決されたかどうか、その結果がブランチによるものなのか、ブランチによらないものなのか

50

を示している。これらの制御信号に基づいて、PCロジック・ユニット270は、条件付きフロー命令の実行の結果として、プリフェッチ・バッファ260とI F I F Oユニット264に置かれている命令の組のどれをキャンセルする必要があるかを判断することができる。

#### 【0066】

いくつかのI E U命令リターン型制御信号（I E Uリターン）が制御ライン350上を送出されて、I E U104によってある命令が実行されたことをI F U102に通知する。これらの命令には、プロシージャ命令からのリターン、トラップからのリターンおよびサブルーチン・コールからのリターンがある。トラップからのリターン命令はハードウェア割り込み処理ルーチンとソフトウェア・トラップ処理ルーチンで同じように使用される。サブルーチン・コールからのリターンもジャンプとリンク型コールと併用される。どの場合も、リターン制御信号は以前に割り込みがかけられた命令ストリームに対して命令フェッチ・オペレーションを再開するようにI F U102に通知するために送られる。これらの信号をI E U104から出すことにより、システム100の正確なオペレーションを維持することができる。「割り込みがかけられた」命令ストリームの再開はリターン命令の実行個所から行われる。

10

#### 【0067】

現命令実行PCアドレス（現I F \_\_ P C）はアドレス・バス352を経由してI E U104へ送られる。このアドレス値（D P C）はI E U104によって実行される正確な命令を指定している。つまり、I E U104が現在のI F \_\_ P Cアドレスを通過した命令を先に試行的に実行している間は、このアドレスは割り込み、例外その他に正確なマシンの状態が分かっていることが必要な事象の発生に対してアーキテクチャ100を正確に制御するために保持されていなければならない。現在実行中の命令ストリームの中の正確なマシンの状態を進めることが可能であるとI E U104が判断すると、P C I n c / S i z e信号がI F U102に送られ、即時に現在のI F \_\_ P Cアドレス値に反映される。

20

#### 【0068】

最後に、アドレスおよび双方向データ・バス354は特殊レジスタのデータを転送するためのものである。このデータはI E U104によってI F U102内の特殊レジスタに入れられ、あるいはそこから読み取られるようにプログラムすることが可能である。特殊レジスタのデータは一般にI F U102が使用できるようにI E U104によってロードされ、あるいは計算される。

30

#### 【0069】

##### D) PCロジック・ユニットの詳細

PC制御ユニット362、割り込み制御ユニット363、プリフェッチPC制御ユニット364および実行PC制御ユニット366を含むPCロジック・ユニット270の詳細図は図3に示されている。

#### 【0070】

PC制御ユニット362はインタフェース・バス126を通してプリフェッチ制御ユニット266、I F I F O制御ロジック・ユニット272、およびI E U104から制御信号を受けて、プリフェッチおよび実行PC制御ユニット364、366に対してタイミング制御を行う。割り込み制御ユニット363はプリフェッチ・トラップ・アドレス・オフセットを判断してそれぞれのトラップ・タイプを処理する該当処理ルーチンを選択することを含めて、割り込みと例外の正確な管理を担当する。プリフェッチPC制御ユニット364は、特にトラップ処理とプロシージャ・ルーチン命令のフローのためのリターン・アドレスをストアすることを含めて、プリフェッチ・バッファ188、190、192をサポートするために必要なプログラム・カウンタの管理を担当する。このオペレーションをサポートするために、プリフェッチPC制御ユニット364は物理アドレス・バス・ライン324上のC C U P A D D E Rアドレスとアドレスライン326上のV M U V M A D D Rアドレスを含むプリフェッチ仮想アドレスを生成することを担当する。その結果、プリフェッチPC制御ユニット364は現在のプリフェッチPC仮想アドレス値を保持す

40

50

ることを担当する。

【0071】

プリフェッチ・オペレーションは一般に制御ライン316上を送出された制御信号を通してIFIFO制御ロジック・ユニット272によって開始される。これを受けて、PC制御ユニット362はいくつかの制御信号を生成して制御ライン372上に出力し、プリフェッチPC制御ユニットを動作させてアドレス・ライン324、326上にPADDRアドレスと、必要に応じてVMADDRアドレスを生成する。値が0から4までのインクリメント信号も制御ライン374上に出送される場合もあるが、これはPC制御ユニット362が現在のプリフェッチ・アドレスから命令の組のフェッチを再実行しているか、一連のプリフェッチ要求の中の2番目の要求に対して位置合わせを行っているか、プリフェッチのために次の全順次命令の組を選択しているか、によって決まる。最後に現在のプリフェッチ・アドレスPF\_PCがバス370上に出送され、実行PC制御ユニット366へ渡される。

10

【0072】

新しいプリフェッチ・アドレスはいくつかのソースから発生する。アドレスの主要なソースはバス352経由で実行PC制御ユニット366から送られた現在のIF\_PCアドレスである。原理的には、IF\_PCアドレスからはリターン・アドレスが得られ、これは、初期コール、トラップまたはプロシージャ命令が現れたとき、プリフェッチPC制御ユニットによってあとで使用されるものである。IF\_PCアドレスは、これらの命令が現れるたびに、プリフェッチPC制御ユニット364内のレジスタにストアされる。このようにして、PC制御ユニット362は制御ライン350を通してIEUリターン信号を受けたとき、プリフェッチPC制御ユニット364内のリターン・アドレス・レジスタを選択して新しいプリフェッチ仮想アドレスを取り出すだけでよく、これによって元のプログラム命令ストリームを再開する。

20

【0073】

プリフェッチ・アドレスのもう一つのソースは実行PC制御ユニット366から相対ターゲット・アドレス・バス382を経由して、あるいはIEU104から絶対ターゲット・アドレス・バス346を経由して送られたターゲット・アドレス値である。相対ターゲット・アドレスとは、実行PC制御ユニット366によって直接に計算できるアドレスである。絶対ターゲット・アドレスは、これらのターゲット・アドレスがIEUレジスタ・ファイルに入っているデータに依存するので、IEU104に生成させる必要がある。ターゲット・アドレスはターゲット・アドレス・バス384を通してプリフェッチPC制御ユニット364へ送られ、プリフェッチ仮想アドレスとして使用される。相対ターゲット・アドレスを計算する際、対応するブランチ命令のオペランド部分もIデコード・ユニット262からバス318のオペランド変位部分を経由して送られる。

30

【0074】

プリフェッチ仮想アドレスのもう一つのソースは、実行PC制御ユニット366である。リターン・アドレス・バス352'は、現在のIF\_PC値(DPC)をプリフェッチPC制御ユニット364へ転送するためのものである。

【0075】

このアドレスは、割り込み、トラップその他コールなどの制御フロー命令が命令ストリーム内に現れた個所でリターン・アドレスとして使用される。プリフェッチPC制御ユニット364は、新しい命令ストリームをプリフェッチするために解放される。PC制御ユニット362は、対応する割り込みまたはトラップ処理ルーチンまたはサブルーチンが実行されると、IEU104からライン350を経由してIEUリターン信号を受け取る。他方、PC制御ユニット362はライン372上のPF\_PC信号の一つを通して、およびライン350経由で送られてきて実行されたリターン命令のIDに基づいて、現在のリターン仮想アドレスを収めているレジスタを選択する。そのあと、このアドレスが使用されてPCロジック・ユニット270によるプリフェッチ・オペレーションを続行する。

40

【0076】

50

最後に、プリフェッチ仮想アドレスが取り出されるもう一つのソースは、特殊レジスタ・アドレスおよびデータ・バス354である。IEU104によって計算またはロードされたアドレス値、または少なくともベース・アドレス値はデータとしてバス354を経由してプリフェッチPC制御ユニット364へ転送される。ベース・アドレスはトラップ・アドレス・テーブル、高速トラップ・テーブル、およびベース・プロシージャ命令ディスパッチ・テーブルのアドレスを含んでいる。バス354を通して、プリフェッチおよびPC制御ユニット364、366内のレジスタの多くを読み取ることもできるので、マシンの状態の対応する側面をIEU104を通して処理することが可能である。

#### 【0077】

実行PC制御ユニット366は、PC制御ユニット362の制御を受けて、現在のIF\_\_PCアドレス値を計算することを主な役割としている。この役割において、実行PC制御ユニット366は、PC制御ユニット362からExPC制御ライン378を経由して送られてきた制御信号と、制御ライン380を経由して送られてきたインクリメント/サイズ制御信号を受けて、IF\_\_PCアドレスを調整する。これらの制御信号は主に、ライン342経由で送られてきたIFIFO読取り制御信号とIEU104から制御ライン344経由で送られてきたPCインクリメント/サイズ値を受け取ると生成される。

#### 1) PFおよびExPC制御/データ・ユニットの詳細

図4はプリフェッチおよび実行PC制御ユニット364、366の詳細ブロック図である。

これらのユニットは主にレジスタ、インクリメンタ(増分器)その他の類似部品、セレクタおよび加算器ブロックから構成されている。これらのブロック間のデータ転送を管理する制御は、PFPC制御ライン372、ExPC制御ライン378およびインクリメント制御ライン374、380を通してPC制御ユニット362によって行われる。説明を分かりやすくするために、図4のブロック図には、これらの個々の制御ラインは示されていない。しかし、これらの制御信号が以下に説明するように、これらのブロックへ送られることは勿論である。

#### 【0078】

プリフェッチPC制御ユニット364の中心となるものはプリフェッチ・セレクタ(PF\_\_PC\_\_SEL)390であり、これは現プリフェッチ仮想アドレスの中央セレクタとして動作する。この現プリフェッチ・アドレスはプリフェッチ・セレクタ390から出力バス392を通過してインクリメンタ・ユニット394へ送られて、次のプリフェッチ・アドレスを生成する。この次のプリフェッチ・アドレスはインクリメンタ出力バス396を通過してレジスタMBUF\_\_PFnPC398、TBUF\_\_PFnPC400、およびEBUF\_\_PFnPC402の並列配列へ送られる。これらのレジスタ398、400、402は実効的には次の命令プリフェッチ・アドレスをストアしているが、本発明の好適実施例によれば別々のプリフェッチ・アドレスがMBUF188、TBUF190およびEBUF192に保持されている。MBUF、TBUFおよびEBUF\_\_PFnPCレジスタ398、400、402にストアされたプリフェッチ・アドレスはアドレス・バス404、408、410からプリフェッチ・セレクタ390へ渡される。したがって、PC制御ユニット362はプリフェッチ・レジスタ398、400、402の別の1つをプリフェッチ・セレクタが選択することを指示することだけでプリフェッチ命令ストリームの即時切り換えを指示することができる。ストリームの中の次の命令の組をプリフェッチするために、そのアドレス値がインクリメンタ394によってインクリメントされると、その値がプリフェッチ・アドレス398、400、402のうち該当するレジスタへ返却される。もう1つの並列レジスタ配列は簡略化のため単一の特殊レジスタ・ブロック412と示されているが、この配列はいくつかの特殊アドレスをストアするためのものである。レジスタ・ブロック412はトラップ・リターン・アドレス・レジスタ、プロシージャ命令リターン・アドレス・レジスタ、プロシージャ命令ディスパッチ・テーブル・ベース・アドレス・レジスタ、トラップ・ルーチン・ディスパッチ・テーブル・ベース・アドレス・レジスタ、および高速トラップ・ルーチン・ベース・アドレス・レジスタから構成されてい

10

20

30

40

50

る。PC制御ユニット362の制御を受けて、これらのリターン・アドレス・レジスタはバス352'を通して現IF\_PC実行アドレスを受け入れることができる。レジスタ・ブロック412内のリターンおよびベース・アドレス・レジスタにストアされたアドレス値はIEU104から独立して読み書きすることができる。レジスタが選択され、値が特殊レジスタ・アドレスおよびデータ・バス354を経由して転送される。

#### 【0079】

特殊レジスタ・ブロック412内のセクタはPC制御ユニット362によって制御され、レジスタ・ブロック412のレジスタにストアされたアドレスを特殊レジスタ出力バス416上に送出してプリフェッチ・セクタ390へ渡すことができる。リターン・アドレスは直接にプリフェッチ・セクタ390へ渡される。ベース・アドレス値は割り込み制御ユニット363から割り込みオフセット・バス373経由で送られてきたオフセット値と結合される。ソースからバス373'経由でプリフェッチ・セクタ390へ渡された特殊アドレスは、新しいプリフェッチ命令ストリームの初期アドレスとして使用され、そのあとインクリメンタ394とプリフェッチ・レジスタ398、400、402の1つを通るアドレスのインクリメント・ループを続行することができる。

#### 【0080】

プリフェッチ・セクタ390へ送られるアドレスのもう1つのソースは、ターゲット・アドレス・レジスタ・ブロック414内のレジスタ配列である。ブロック414内のターゲット・レジスタには好適実施例によれば8つの潜在的ブランチ・ターゲット・アドレスがストアされる。これらの8つの記憶ロケーションはIFIUユニット264の最下位の2マスタ・レジスタ216、224に保持されている8つの潜在的に実行可能な命令に論理的に対応している。これらの命令のどれでもが、および潜在的にはすべてが条件付きブランチ命令となり得るので、ターゲット・レジスタ・ブロック414は予め計算されたターゲット・アドレスをストアしておくので、TBUF190を通してターゲット命令ストリームをプリフェッチするために使用することを待たせることができる。特に、PC制御ユニット362がターゲット命令ストリームのプリフェッチを即時に開始するように条件付きブランチ・バイアスがセットされると、ターゲット・アドレスはターゲット・レジスタ・ブロック414からアドレス・バス418を経由してプリフェッチ・セクタ390へ送られる。インクリメンタ394によってインクリメントされたあと、アドレスはTBUF\_PFnPC400へ戻されてストアされ、ターゲット命令ストリームをあとでプリフェッチするオペレーションで使用される。別のブランチ命令がターゲット命令ストリームに現れると、その2番目のブランチのターゲット・アドレスが計算され、最初の条件付きブランチ命令が解決されて使用されるまでの間、ターゲット・レジスタ配列414にストアされている。

#### 【0081】

ターゲット・レジスタ・ブロック414にストアされた計算で求めたターゲット・アドレスは実行PC制御ユニット366内のターゲット・アドレス計算ユニットからアドレス・ライン382を経由して、あるいはIEU104から絶対ターゲット・アドレス・バス346を経由して転送される。

#### 【0082】

プリフェッチPF\_PCセクタ390を通過して転送されるアドレス値は、完全な32ビット仮想アドレス値である。ページ・サイズは本発明の好適実施例では16Kバイトに固定されており、最大ページ・オフセット・アドレス値〔13:0〕に対応している。したがって、現プリフェッチ仮想ページ・アドレス〔27:14〕に変化がなければVMUページの変換は不要である。プリフェッチ・セクタ390内のコンパレータはそのことを検出する。VMU変換要求信号(VMXLAT)は、インクリメントがページ境界をこえて行われたか、制御のフローが別のページ・アドレスへブランチしたために、仮想ページ・アドレスが変化したとき、ライン372'を経由してPC制御ユニット362へ送られる。他方、PC制御ユニット362はライン324上のCCU\_PADDRのほか、VM\_VADDRアドレスをバッファ・ユニット420からライン326上に送出し、該

10

20

30

40

50

当の制御信号をVMU制御ライン326、328、330上に送出して、VMU仮想ページから物理ページへの変換を得るように指示する。ページ変換が必要でない場合は、現物理ページ・アドレス〔31：14〕はバス122上のVMUユニット108の出力側のラッチによって保持される。

**【0083】**

バス370上に送出された仮想アドレスはインクリメント制御ライン374から送られてきた信号を受けて、インクリメンタ394によってインクリメントされる。インクリメンタ394は、次の命令の組を選択するために、命令の組を表す値（4命令または16バイト）だけインクリメントする。CCUユニット106へ渡されるプリフェッチ・アドレスの下位4ビットはゼロになっている。したがって、最初のブランチ・ターゲット命令の組の中の実際のターゲット・アドレス命令は最初の命令ロケーションに置かれていない場合がある。しかしアドレスの下位4ビットはPC制御ユニット362へ送られるので、最初のブランチ命令のロケーションをIFU102が判別することができる。ターゲット・アドレスの下位ビット〔3：2〕を2ビット・バッファ・アドレスとして返して、位置合わせされていないターゲット命令の組から実行すべき正しい最初の命令を選択するための検出と処理は、新しい命令ストリーム、つまり、命令ストリームの中の最初の非順次命令の組のアドレスの最初のプリフェッチのときだけ行われる。命令の組の最初の命令のアドレスと命令の組をプリフェッチする際に使用されるプリフェッチ・アドレスとの間の非位置合わせの関係は、現順次命令ストリームが存続している間無視することができ、そのあとも無視される。

**【0084】**

図4に示した機能ブロックの残りの部分は実行PC制御ユニット366を構成している。本発明の好適実施例によれば、実行PC制御ユニット366は独立に機能するプログラム・カウンタ・インクリメンタを独自に備えている。この機能の中心となるのは実行セレクトタ(DPC\_SEL)430である。実行セレクトタ430からアドレス・バス352'上に出力されるアドレスはアーキテクチャ100の現在の実行アドレス(DPC)である。この実行アドレスは加算ユニット434へ送られる。ライン380上に送出されたインクリメント/サイズ制御信号は1から4までの命令インクリメント値を指定しており、この値は加算ユニット434によってセレクトタ430から得たアドレスに加えられる。加算器432が出力ラッチ機能を実行するたびに、インクリメントされた次の実行アドレスがアドレス・ライン436を経て直接に実行セレクトタ430に返され、次の命令インクリメント・サイクルで使用される。

**【0085】**

初期実行アドレスとその後の全ての新しいストリーム・アドレスは、アドレスライン440を經由して新ストリーム・レジスタ・ユニット438から得られる。新ストリーム・レジスタ・ユニット438はプリフェッチ・セレクトタ390からPFPCアドレス・バス370を經由して送られてきた新しい現プリフェッチ・アドレスを直接にアドレス・バス440に渡すことも、後で使用するためにストアしておくこともできる。つまり、プリフェッチPC制御ユニット364が新しい仮想アドレスからプリフェッチを開始することを判断した場合は、新しいストリーム・アドレスは新ストリーム・レジスタ・ユニット438によって一時的にストアされる。PC制御ユニット362は、プリフェッチと実行インクリメントの両サイクルに関与することによって、実行アドレスが新命令ストリームを開始した制御フロー命令に対応するプログラム実行箇所までに達するまで新ストリーム・アドレスを新ストリーム・レジスタ438に置いておく。新ストリーム・アドレスはそのあと新ストリーム・レジスタ・ユニット438から出力されて、実行セレクトタ430へ送られ、新命令ストリーム内の実行アドレスを独立して生成することを開始する。

**【0086】**

本発明の好適実施例によれば、新ストリーム・レジスタ・ユニット438は2つの制御フロー命令ターゲット・アドレスをバッファリングする機能を備えている。新ストリーム・アドレスを即時に取り出すことにより、殆ど待ち時間がなく実行PC制御ユニット36

10

20

30

40

50

6を現実行アドレス列の生成から新実行アドレス・ストリーム列の生成に切り換えることができる。

【0087】

最後にIF\_\_PCセクタ(IF\_\_PC\_\_SEL)442は最終的に現IF\_\_PCアドレスをアドレス・バス352上に送出してIEU104へ送るためのものである。IF\_\_PCセクタ442への入力の実行セクタ430または新ストリーム・レジスタ・ユニット438から得た出力アドレスである。ほとんどの場合、IF\_\_PCセクタ442はPC制御ユニット362の指示を受けて、実行セクタ430から出力された実行アドレスを選択する。しかし、新命令ストリームの実行開始のために使用される新仮想アドレスへ切り替える際の待ち時間をさらに短縮するために、新ストリーム・レジスタ・ユニット438からの選択したアドレスをバイパスして、バス440経由で直接にIF\_\_PCセクタ442へ送り、現IF\_\_PC実行アドレスとして得ることができる。

10

【0088】

実行PC制御ユニット366は、全ての相対ブランチ・ターゲット・アドレスを計算する機能を備えている。現実行点アドレスと新ストリーム・レジスタ・ユニット438から得たアドレスは、アドレス・バス352'、440を経由して制御フロー・セクタ(CF\_\_PC)446に渡される。その結果、PC制御ユニット362は大幅な柔軟性を持ってターゲット・アドレス計算の基となる正確な初期アドレスを選択することができる。

【0089】

この初期アドレス、つまり、ベースアドレスはアドレス・バス454を経由してターゲット・アドレスALU450へ送られる。ターゲットALU450への入力となるもう一つの値は、制御フロー変位計算ユニット452からバス458経由で送られてくる。相対ブランチ命令は、アーキテクチャ100の好適実施例によれば新相対ターゲット・アドレスを指定した即値モード定数の形態をした変位値を含んでいる。制御フロー変位計算ユニット452はイデコード・ユニットのオペランド出力バス318から初めて得たオペランド変位値を受け取る。最後に、オフセット・レジスタ値はライン456を経由してターゲット・アドレスALU450へ送られる。オフセット・レジスタ448はPC制御ユニット362から制御ライン378'を経由してオフセット値を受け取る。オフセット値の大きさはアドレス・ライン454上を送られるベース・アドレスから相対ターゲット・アドレスを計算するときの現ブランチ命令のアドレスまでのアドレス・オフセットに基づいてPC制御ユニット362によって判断される。つまり、PC制御ユニット362は、IFIFO制御ロジック・ユニット272を制御することによって、現実行点アドレスの命令(CP\_\_PCによって要求された)とイデコード・ユニット262によって現在処理中の、従ってPCロジック・ユニット270によって処理中の命令を分離している命令の個数を追跡して、その命令のターゲット・アドレスを判断する。

20

30

【0090】

相対ターゲット・アドレスがターゲット・アドレスALU450によって計算されると、そのターゲット・アドレスはアドレス・バス382を通して、対応するターゲット・レジスタ414に書き込まれる。

【0091】

2) PC制御アルゴリズムの詳細

1. メイン命令ストリームの処理: MBUF\_\_PFnPC

- 1.1 次のメイン・フロープリフェッチ命令のアドレスはMBUF\_\_PFnPCにストアされる
- 1.2 制御フロー命令がないときは、32ビット・インクリメンタはMBUF\_\_PFnPCに入っているアドレス値を各プリフェッチ・サイクルごとに16バイト(x16)だけ調整する
- 1.3 無条件制御フロー命令がイデコードされると、命令の組に続いてフェッチされた全てのプリフェッチ・データはフラッシュされ、MBUF\_\_PFnPCにはターゲット・レジスタ・ユニット、PF\_\_PCセ

40

50

レクタおよびインクリメンタを通して、新しいメイン命令ストリーム・アドレスがロードされる。新しいアドレスは新ストリーム・レジスタにもストアされる

- 1.3.1 相対無条件制御フローのターゲット・アドレスはIFUが保持しているレジスタ・データからと制御フロー命令の後に置かれたオペランド・データからIFUによって計算される
- 1.3.2 絶対無条件制御フローのターゲット・アドレスはレジスタ基準値、ベース・レジスタ値、及びインデックス・レジスタ値からIEUによって最終的に計算される 10
- 1.3.2.1 命令プリフェッチ・サイクルは絶対アドレス制御フロー命令に対してターゲット・アドレスがIEUから返されるまで停止する。命令実行サイクルは続行される
- 1.4 無条件制御フロー命令から得た次のメイン・フロー・プリフェッチ命令のアドレスはバイパスされて、ターゲット・アドレス・レジスタ・ユニット、PF\_\_PCセクタおよびインクリメンタを經由して送られ、最終的にMBUF\_\_PFnPCにストアされ、プリフェッチは1.2から続けられる
2. プロシージャ命令ストリームの処理：EBUF\_\_PFnPC 20
- 2.1 プロシージャ命令はメインまたはブランチ・ターゲット命令ストリームの中でプリフェッチされる。ターゲット・ストリームの中でフェッチされた場合は条件付き制御フェッチ命令が解決され、プロシージャ命令がMBUFへ転送されるまでプロシージャ・ストリームのプリフェッチを停止する。これにより、プロシージャ命令ストリームに現れた条件付き制御フローを処理する際にTBUFを使用できる
- 2.1.1 プロシージャ命令はプロシージャ命令ストリームの中においてはならない。つまり、プロシージャ命令はネストしてはならない。プロシージャ命令からリターンすると、実行は主命令ストリームに戻る。ネストを可能にするためには、ネストしたプロシージャ命令から別の専用リターンが必要である。アーキテクチャはこの種の命令を容易にサポートできるが、プロシージャ命令をネストする機能があっても、アーキテクチャの性能が向上する見込みはない 30
- 2.1.2 メイン命令ストリームにおいては、第1及び第2条件付き制御フロー命令を含む命令の組を含んでいるプロシージャ命令ストリームは第1命令の組の中の条件付き制御フロー命令が解決し、第2条件付き制御フロー命令セットがMBUFへ転送されるまで第2条件付き制御フロー命令の組に対してプリフェッチを停止する 40
- 2.2 プロシージャ命令は、命令の即値モード・オペランド・フィールドとして含まれている相対オフセットによって、プロシージャ・ルーチンの開始アドレスを示している
- 2.2.1 プロシージャ命令から得られたオフセット値はIFUに維持されているプロシージャ・ベース・アドレス(PBR)レジスタに入っている値と結合される。このPBRレジスタは、特殊レジスタの移動命令が実行されると、特殊アドレスおよびデータバスを通して読み書き可能である 50

- 2.3 プロシージャ命令が現れると、次のメイン命令ストリーム  $IF\_PC$  アドレスは  $DPCL$  リターン・アドレス・レジスタにストアされ、プロセッサ・ステータス・レジスタ ( $PSR$ ) 内のプロシージャ進行中ビット ( $procedure-in-progress\ bit$ ) がセットされる
- 2.4 プロシージャ・ストリームの開始アドレスは、 $PBR$  レジスタ (プロシージャ命令オペランド・オフセット値を加えて) から  $PF\_PC$  セレクタへ送られる
- 2.5 プロシージャ・ストリームの開始アドレスは、新ストリーム・レジスタ・ユニットとインクリメントへ同時に送られ、 $(\times 16)$  だけインクリメントする。インクリメントされたアドレスはそのあと  $EBUF\_PFnPC$  にストアされる 10
- 2.6 制御フロー命令がないと、 $32$  ビット・インクリメントは各プロシージャ命令プリフェッチ・サイクルごとに  $EBUF\_PFnPC$  に入っているアドレス値を、 $(\times 16)$  だけ調整する
- 2.7 無条件制御フロー命令がイデコードされると、ブランチ命令のあとにフェッチされた全てのプリフェッチ・データはフラッシュされ、 $EBUF\_PFnPC$  には新しいプロシージャ命令ストリーム・アドレスがロードされる
- 2.7.1 相対無条件制御フロー命令のターゲット・アドレスは  $IFU$  に保持されているレジスタデータからと制御フロー命令の即値モード・オペランド・フィールド内に入っているオペランド・データとから  $IFU$  によって計算される 20
- 2.7.2 絶対無条件ブランチのターゲット・アドレスはレジスタ基準値、ベース・レジスタ値およびインデックス・レジスタ値から  $IEU$  によって計算される
- 2.7.2.1 命令プリフェッチ・サイクルは絶対アドレス・ブランチに対してターゲット・アドレスが  $IEU$  から返されるまで停止する。実行サイクルは続行される 30
- 2.8 次にプリフェッチされたプロシージャの命令の組のアドレスは  $EBUF\_PFnPC$  にストアされプリフェッチは 1.2 から続けられる
- 2.9 プロシージャ命令からのリターンがイデコードされると、プリフェッチが  $uPC$  レジスタにストアされているアドレスから続けられ、そのあと  $(\times 16)$  だけインクリメントされ、あとでプリフェッチするために  $MBUF\_PFnPC$  レジスタに返される
- 3 ブランチ命令ストリームの処理:  $TBUF\_PFnPC$
- 3.1  $MBUF$  命令ストリームの中の最初の命令の組に現れた条件つき制御フロー命令がイデコードされると、ターゲット・アドレスはそのターゲット・アドレスが現アドレスに対する相対アドレスならば  $IFU$  によって絶対アドレスならば  $IEU$  によって判断される 40
- 3.2 「ブランチを行うバイアス」の場合:
- 3.2.1 ブランチが絶対アドレスに行われる場合はターゲット・アドレスが  $IEU$  から返されるまで命令プリフェッチ・サイクルを停止する。実行サイクルは続行される
- 3.2.2  $PF\_PC$  セレクタとインクリメントを経由して転送することによってブランチ・ターゲット・アドレスを  $TBUF\_PFnPC$  にロードする
- 3.2.3 ターゲット命令ストリームがプリフェッチされて  $TBU$  50

- Fに入れられたあとで実行するためにI F I F Oに送られる。I F I F OとT B U Fが一杯になると、プリフェッチを停止する
- 3.2.4 32ビット・インクリメンタは各プリフェッチ・サイクルごとにT B U F \_ P F n P Cに入っているアドレス値を(x 1 6)だけ調整する
- 3.2.5 ターゲット命令ストリーム内の2番目の命令の組に現れた条件付き制御フロー命令がIデコードされるとプリフェッチ・オペレーションを、第1の(主)セット内の全ての条件付きブランチ命令が解決されるまで停止する(しかし、先に進んで、相対ターゲット・アドレスを計算しターゲット・レジスタにストアする) 10
- 3.2.6 最初の命令の組の内の条件付きブランチを「行う」と解釈された場合：
- 3.2.6.1 ブランチのソースがプロシージャ進行中ビットから判断されたE B U F命令の組であったときはM B U FまたはE B U Fに入っている最初の条件付きフロー命令の組のあとに置かれた命令セットをフラッシュする
- 3.2.6.2 プロシージャ進行中ビットの状態に基づいて、T B U F \_ P F n P C値をM B U F \_ P F n P CまたはE B U Fへ転送する 20
- 3.2.6.3 プロシージャ進行中ビットの状態に基づいて、プリフェッチしたT B U F命令をM B U FまたはE B U Fへ転送する
- 3.2.6.4 2番目の条件付きブランチ命令の組がIデコードされていなければ、プロシージャ進行中ビットの状態に基づいて、M B U FまたはE B U Fプリフェッチ・オペレーションを続行する
- 3.2.6.5 2番目の条件付きブランチ命令がIデコードされていれば、その命令の処理を開始する(ステップ3.3.1へ進む) 30
- 3.2.7 最初の条件付き命令の組の中の命令に対する条件付き制御を「行わない」と解釈された場合：
- 3.2.7.1 ターゲット命令ストリームからの命令の組と命令のI F I F OとI E Uをフラッシュする
- 3.2.7.2 M B U FまたはE B U Fプリフェッチ・オペレーションを続行する
- 3.3 「ブランチが行われないバイアス」の場合：
- 3.3.1 命令をプリフェッチしてM B U Fに入れることを停止する。実行サイクルを続ける 40
- 3.3.1.1 最初の条件付き命令の組の中の条件付き制御フロー命令が相対ならばターゲット・アドレスを計算し、ターゲット・レジスタにストアする
- 3.3.1.2 最初の条件付き命令の組の中の条件付き制御フロー命令が絶対ならば、I E Uがターゲット・アドレスを計算して、そのアドレスをターゲット・レジスタに返すまで待つ
- 3.3.1.3 2番目の命令の組の中の条件付き制御フロー命令のIデコードが行われると、最初の条件付き命 50

- 令セットの中の条件付き制御フロー命令が解決されるまでプリフェッチ・オペレーションを停止する
- 3.3.2 最初の条件付きジャンプのターゲット・アドレスが計算されると、TBUF\_PFNPCにロードし、メイン命令ストリームの実行と並行して命令をプリフェッチしてTBUFに入れることを開始する。ターゲット命令セットはロードされない(したがって、ジャンプ・ターゲット命令は最初の命令の組の中の各条件付き制御フロー命令が解決されたとき用意されている) 10
- 3.3.3 最初のセットの中の条件付き制御フロー命令が「行われる」と解釈された場合:
- 3.3.3.1 ジャンプのソースがEBUF命令ストリームであるとプロシージャ進行中ビットの状態から判断されると、MBUFまたはEBUFをフラッシュし、最初の条件付きジャンプ命令の組のあとに置かれたメイン・ストリームからの命令のIFIFOとIEUをフラッシュする
- 3.3.3.2 プロシージャ進行中ビットの状態から判断した通りに、TBUF\_PFNPC値をMBUF\_PFNPCまたはEBUFへ転送する 20
- 3.3.3.3 プロシージャ進行中ビットの状態から判断した通りに、プリフェッチしたTBUF命令をMBUFまたはEBUFへ転送する
- 3.3.3.4 プロシージャ進行中ビットの状態から判断した通りに、MBUFまたはEBUFプリフェッチ・オペレーションを続行する
- 3.3.4 最初のセット内の条件付き制御フロー命令が「行われな
- 3.3.4.1 ターゲット命令ストリームからの命令の組のTBUFをフラッシュする 30
- 3.3.4.2 2番目の条件付きジャンプ命令がIデコードされなかった場合は、プロシージャ進行中ビットの状態から判断した通りに、MBUFまたはEBUFプリフェッチ・オペレーションを続ける
- 3.3.4.3 2番目の条件付きジャンプ命令がIデコードされた場合は、その命令の処理を開始する(ステップ3.4.1へ進む)
- 4 割り込み、例外およびトラップ命令 40
- 4.1 トラップは広義には次のものからなる
- 4.1.1 ハードウェア割り込み
- 4.1.1.1 非同期(外部)発生事象、内部または外部
- 4.1.1.2 いつでも発生し、持続する
- 4.1.1.3 アトミック(通常)命令間で優先順にサービスを受け、プロシージャ命令を一時中止する
- 4.1.1.4 割り込みハンドラの開始アドレスはトラップ・ハンドラ入り口点の事前定義テーブルまでのベクトル番号オフセットとして判断される
- 4.1.2 ソフトウェア・トラップ命令
- 4.1.2.1 非同期(外部)発生命令 50

- 4 . 1 . 2 . 2 例外として実行されるソフトウェア命令
- 4 . 1 . 2 . 3 トラップ・ハンドラの開始アドレスは、T B R または F T B レジスタにストアされたベース・アドレス値と結合されたトラップ番号オフセットから判断される
- 4 . 1 . 3 例外
  - 4 . 1 . 3 . 1 命令と同期して発生する事象
  - 4 . 1 . 3 . 2 命令の実行時に処理される
  - 4 . 1 . 3 . 3 例外の結果により、期待された命令とすべての後続実行命令はキャンセルされる 10
  - 4 . 1 . 3 . 4 例外ハンドラの開始アドレスは、トラップ・ハンドラ入り口点の事前定義テーブルまでのトラップ番号オフセットから判断される
- 4 . 2 トラップ命令ストリーム・オペレーションはそのとき実行中の命令ストリームとインラインで実行される
- 4 . 3 トラップ処理ルーチンが次の割り込み可能トラップの前に x P C アドレスをセーブすることを条件に、トラップはネストが可能である。そうしないと、現トラップ・オペレーションの完了前にトラップが現れると、マシンの状態が壊れることになる
- 5 トラップ命令ストリームの処理：x P C 20
  - 5 . 1 トラップが現れた時：
    - 5 . 1 . 1 非同期割り込みが起こると、そのとき実行中の命令は一時中断される
    - 5 . 1 . 2 同期例外が起こると、例外を起こした命令が実行されるとトラップが処理される
  - 5 . 2 トラップが処理されたとき：
    - 5 . 2 . 1 割り込みは禁止される
    - 5 . 2 . 2 現在の I F \_ P C アドレスは x P C トラップ状態リターン・アドレス・レジスタにストアされる
    - 5 . 2 . 3 I F \_ P C アドレスとそのあとのアドレスにある I F I F O と M B U F プリフェッチ・バッファはフラッシュされる 30
    - 5 . 2 . 4 アドレス I F \_ P C と、そのあとのアドレスの実行された命令と、その命令の結果は I E U からフラッシュされる
    - 5 . 2 . 5 M B U F \_ P F n P C に、トラップ・ハンドラ・ルーチンのアドレスがロードされる
      - 5 . 2 . 5 . 1 トラップのソースは特殊レジスタ群に入っているトラップ番号によって判断されたトラップ・タイプに応じて T B R または F T B レジスタをアドレス指定している 40
    - 5 . 2 . 6 命令がプリフェッチされ、通常通りに実行するために I F I F O に入れられる
    - 5 . 2 . 7 トラップ・ルーチンの命令がそのあと実行される
      - 5 . 2 . 7 . 1 トラップ処理ルーチンは x P C アドレスを所定のロケーションにセーブする機能を備え、割り込みを再び可能にする。x P C レジスタは特殊レジスタ移動命令で、および特殊レジスタ・アドレスとデータ・バスを通して読み書きされる
    - 5 . 2 . 8 トラップ命令からのリターンを実行することによってト 50

ラップ状態から抜け出る必要がある

- 5.2.8.1 以前にセーブしていた時はxPCアドレスをその事前定義ロケーションから復元してからトラップ命令からのリターンを実行する必要がある
- 5.3 トラップ命令からのリターンが実行されたとき：
  - 5.3.1 割り込みが可能にされる
  - 5.3.2 プロシージャ進行中ビットの状態から判断したとおりに、xPCアドレスが現在の命令ストリーム・レジスタMBUFまたはEBUF\_PFnPCに戻され、プリフェッチがそのアドレスから続行される
  - 5.3.3 xPCアドレスが新ストリーム・レジスタを通してIFPCレジスタに復元される。

10

## 【0092】

E) 割り込みおよび例外の処理

### 1) 概要

割り込みと例外は、それらが可能にされている限り、プロセッサがメイン命令ストリームから実行されているか、プロシージャ命令ストリームから実行されているか、プロシージャ命令ストリームから実行されているかに関係なく処理される。割り込みと例外は優先順にサービスが受けられ、クリアされるまで持続している。トラップ・ハンドラの開始アドレスは下述するように、トラップ・ハンドラの事前定義テーブルまでのベクトル番号オフセットとして判断される。

20

## 【0093】

割り込みと例外は、本実施例では基本的に2つのタイプがある。すなわち、命令ストリームの中の特定の命令と同期して引き起こされるものと、命令ストリームの中の特定の命令と非同期に引き起こされるものである。割り込み、例外、トラップおよびフォールト(fault)の用語は、本明細書では相互に使い分けて用いられている。非同期割り込みは、命令ストリームと同期して動作していない、オン・チップまたはオフ・チップのハードウェアによって引き起こされる。例えば、オン・チップ・タイマ/カウンタによって引き起こされる割り込みは、オフ・チップから引き起こされるハードウェア割り込みやマスク不能割り込み(non-maskable interrupt)(NMI)と同じように、非同期である。非同期割り込みが引き起こされると、プロセッサ・コンテキストが凍結され(frozen)、すべてのトラップが割り込み禁止され、ある種のプロセッサ状況情報がストアされ、プロセッサは受け取った特定の割り込みに対応する割り込みハンドラにベクトルを向ける。割り込みハンドラがその処理を完了すると、プログラム実行は割り込み発生時に実行中であったストリームの中の最後に完了した命令のあとに置かれた命令から続けられる。

30

## 【0094】

同期例外とは、命令ストリームの中の命令と同期して引き起こされる例外である。これらの例外は特定の命令に関連して引き起こされ、問題の命令が実行されるまで保留される。好適実施例では、同期例外はプリフェッチ時、命令デコード時、または命令実行時に引き起こされる。プリフェッチ例外には、例えばTLB不一致、その他のVMU例外がある。デコード例外は、例えばデコード中の命令が違法命令であるか、プロセッサの現特権レベル(privilege level)に一致していないと引き起こされる。実行例外は、例えばゼロによる除算といった算術演算エラーが原因で引き起こされる。これらの例外が起こると、好適実施例では例外を引き起こした特定命令と例外とを対応づけ、その命令がリタイア(retire)されるときまでその状態が維持される。その時点で、以前に完了した命令がすべてリタイアされ、例外を引き起こした命令からの試行的結果があれば、試行的に実行された後続の命令の試行的結果と同じようにフラッシュされる。そのあと、その命令で引き起こされた最高優先度例外に対応する例外ハンドラに制御が渡される。

40

。

50

## 【 0 0 9 5 】

ソフトウェア・トラップ命令はCF\_DET274 (図2) によってIデコード・ステージで検出され、無条件コール命令その他の同期トラップと同じように処理される。つまり、ターゲット・アドレスが計算され、プリフェッチはそのときのプリフェッチ待ち行列 (EBUFまたはMBUF) まで続けられる。これと同時に、その例外は命令と対応づけられて記録され、命令がリタイアされるとき処理される。他のタイプの同期例外はすべて、例外を引き起こした特定命令と対応づけられて記録され、累積されるだけで実行時に処理される。

## 【 0 0 9 6 】

## 2) 非同期割り込み:

非同期割り込みは、割り込みライン292を通してPCロジック・ユニット270に通知される。図3に示すように、これらのラインはPCロジック・ユニット270内の割り込みロジック・ユニット363に通知するためのもので、NMIライン、IRQラインおよび1組の割り込みレベル・ライン (LVL) からなっている。NMIラインはマスク不能割り込みを通知し、外部ソースを起点としている。これは、ハードウェア・リセットを除き最高優先度割り込みである。IRQラインも外部ソースを起点としており、外部デバイスがハードウェア割り込みをいつ要求したかを通知する。好適実施例では、外部から起こされるハードウェア割り込みは最高32個までユーザが定義することができ、割り込みを要求した特定外部デバイスは割り込みレベル・ライン (LVL) 上に割り込み番号 (0-31) を送出する。メモリ・エラー・ラインはMCU110によってアクティベート (活性化) され、様々な種類のメモリ・エラーを通知する。その他の非同期割り込みライン (図示せず) も割り込みロジック・ユニット363に通知するために設けられている。これらには、タイマ/カウンタ割り込み、メモリ入出力 (I/O) エラー割り込み、マシン・チェック割り込み、およびパフォーマンス・モニタ割り込みを要求するためのラインがある。

## 【 0 0 9 7 】

非同期割り込みの各々は下述する同期例外と同様に対応する事前定義トラップ番号が関連付けられている。これらのトラップ番号は32個が32個のハードウェア割り込みレベルと関連付けられている。これらのトラップ番号のテーブルは割り込みロジック・ユニット363に維持されている。一般にトラップ番号が大きくなると、トラップの優先度が高くなる。

## 【 0 0 9 8 】

非同期割り込みの一つが割り込みロジック・ユニット363に通知されると、割り込み制御ユニット363は割り込み要求をINT\_REQ/ACKライン340を経由してIEU104へ送出する。また、割り込み制御ユニット363はプリフェッチ一時中止信号をライン343を経由してPC制御ユニット362へ送信し、PC制御ユニット362に命令をプリフェッチすることを中止させる。IEU104はそのときの実行中の命令を全てキャンセルし、すべての試行的結果を中止するか、一部またはすべての命令を完了させる。好適実施例では、そのとき実行中の命令をすべてキャンセルすることによって、非同期割り込みに対する応答を高速化している。いずれの場合も、実行PC制御ユニット366内のDPCは、IEU104が割り込みの受信を確認する前に、最後に完了し、リタイアされた命令に対応するように更新される。プリフェッチされて、MBUF、EBUF、TBUFおよびIFIFO264に置かれている他の命令もすべてキャンセルされる。

## 【 0 0 9 9 】

IEU104は割り込みハンドラから割り込みを受け取る準備状態にあるときだけ、割り込み受信確認信号をINT\_REQ/ACKライン340を経由して、割り込み制御ユニット363へ送り返す。この信号を受け取ると、割り込み制御ユニット363は、以下で説明するように、該当するトラップ・ハンドラにディスパッチする。

## 【 0 1 0 0 】

## 3) 同期例外

10

20

30

40

50

同期例外の場合は、割り込み制御ユニット 3 6 3 は各命令の組ごとに 4 個が 1 組の内部例外ビット（図示せず）をもっており、各ビットはセット内の各命令に対応づけられている。割り込み制御ユニット 3 6 3 は各命令で見つかったとき、通知するトラップ番号も維持している。

#### 【 0 1 0 1 】

特定の命令の組がプリフェッチされている途中で、VMUがTLB不一致または別のVMU例外を通知するとこの情報はPCロジック・ユニット 2 7 0 へ、特に割り込み制御ユニット 3 6 3 へVMU制御ライン 3 3 2、3 3 4 を経由して送られる。割り込み制御ユニット 3 6 3 は、この信号を受信すると、以後のプリフェッチを一時中止するようにライン 3 4 3 を経由して、PC制御ユニット 3 6 2 に通知する。これと同時に、割り込み制御ユニット 3 6 3 は、命令の組が送られる先のプリフェッチ・バッファに関連するVM\_\_MissまたはVM\_\_Exc pビットのどちらか該当する方をセットする。そのあと、割り込み制御ユニット 3 6 3 は命令の組の中の命令のどれも有効でないので、その命令の組に対応する 4 個の内部例外標識ビットを全部セットし、問題を起こした命令の組の中の 4 命令の各々に対応して受信した特定例外のトラップ番号をストアする。問題のある命令より前の命令のシフトと実行は、問題の命令の組がIFIFO 2 6 4 内で最低レベルに達するまで通常通りに続行される。

10

#### 【 0 1 0 2 】

同様に、プリフェッチ・バッファ 2 6 0、Iデコード・ユニット 2 6 2 またはIFIFO 2 6 4 を通して命令をシフトしている途中で他の同期例外が検出されると、この情報も割り込み制御ユニット 3 6 3 へ送られ、ユニット 3 6 3 は例外を引き起こした命令に対応する内部例外標識ビットをセットし、その例外に対応するトラップ番号をストアする。プリフェッチ同期例外の場合と同じように、問題を起こした命令より前の命令のシフトと実行は、問題の命令の組がIFIFO 2 6 4 内で最低レベルに達するまで通常通りに続行される。

20

#### 【 0 1 0 3 】

好適実施例では、プリフェッチ・バッファ 2 6 0、Iデコード・ユニット 2 6 2 またはIFIFO 2 6 4 を通して命令をシフトしている途中で検出される例外は、ソフトウェア・トラップ命令の 1 タイプだけである。ソフトウェア・トラップ命令は、CF\_\_DETユニット 2 7 4 によってIデコード・ステージで検出される。一部の実施例では他の形態の同期例外がIデコード・ステージで検出されるが、他の同期例外の検出は、命令が実行ユニット 1 0 4 に到着するまで待つようにするのが好ましい。このようにすれば、特権命令を処理すると起こるようなある種の例外が、命令が実効的に順序内で実行される前に変化する恐れのあるプロセッサ状態に基づいて通知されるのが防止される。違法命令のように、プロセッサ状態に左右されない例外はIデコード・ステージで検出可能であるが、すべての実行前同期例外（VMU例外は別として）を同じロジックで検出するようにすれば、最低限のハードウェアですむことになる。また、そのような例外の処理は時間を重視することがめったにないので、命令が実行ユニット 1 0 4 に届くまでの待ちによる時間浪費もない。

30

#### 【 0 1 0 4 】

上述したように、ソフトウェア・トラップ命令は、CF\_\_DETユニット 2 7 4 によってIデコード・ステージで検出される。割り込みロジック・ユニット 3 6 3 内のその命令に対応する内部例外標識ビットはセットされ、0 から 1 2 7 までの番号で、ソフトウェア・トラップ命令の即値モード・フィールドに指定できるソフトウェア・トラップ番号はトラップ命令に対応付けられてストアされる。しかし、プリフェッチ同期例外と異なり、ソフトウェア・トラップは制御フロー命令だけでなく、同期例外としても扱われるので、割り込み制御ユニット 3 6 3 は、ソフトウェア・トラップ命令が検出されたときプリフェッチを一時中止するようにPC制御ユニット 3 6 2 に通知しない。その代わりに、命令がIFIFO 2 6 4 を通してシフトされると同時にIFU 1 0 2 はトラップ・ハンドラをプリフェッチしてMBUF命令ストリーム・バッファに入れる。

40

50

## 【 0 1 0 5 】

命令の組が I F I F O 2 6 4 の最低レベルまで達すると、割り込みロジック・ユニット 3 6 3 はその命令の組の例外標識ビットを 4 ビット・ベクトルとして S Y N C H \_ I N T \_ I N F O ライン 3 4 1 経由で I E U 1 0 4 へ送り、命令の組の中で同期例外の発生源とすでに判断されていた命令があれば、どの命令であるかを通知する。I E U 1 0 4 は即時に応答しないで、命令の組の中の全ての命令が通常の方法でスケジュールされるようにする。整数算術演算例外と言った別の例外は、実行時に引き起こされる場合がある。特権命令が実行されたために起こる例外のように、マシンの現在状態に左右される例外もこの時点で検出され、マシンの状態が命令ストリーム内の以前の全ての命令に対して最新となるようにするために、P S R に影響を与える可能性のあるすべての命令（特殊な移動やトラップ命令からのリターンなど）は強制的に順序内で実行される。なんらかの同期例外の発生源となった命令がリタイアされる直前にあるときだけ、例外が起こったことが割り込みロジック・ユニット 3 6 3 に通知される。

10

## 【 0 1 0 6 】

I E U 1 0 4 は試行的に実行され、同期例外を引き起こした最初の命令に先行する命令に現れた全ての命令をリタイアし、試行的に実行され、命令ストリームにそのあとに現れた命令からの試行的結果をフラッシュする。例外を引き起こした特定の命令は、トラップから戻ると再実行されるのが普通であるので、この命令もフラッシュされる。そのあと、実行 P C 制御ユニット 3 6 6 内の I F \_ P C は実際にリタイアされた最後の命令に対応するように更新され、例外が割り込み制御ユニット 3 6 3 に通知される。

20

## 【 0 1 0 7 】

例外の発生源である命令がリタイアされると、I E U 1 0 4 はリタイアされる命令の組（レジスタ 2 2 4）の中に同期例外を起こした命令があれば、どの命令であるかを示した新しい 4 ビット・ベクトルを命令の組内の最初の例外の発生源を示した情報と一緒に、S Y N C H \_ I N T \_ I N F O ライン 3 4 1 を経由して割り込みロジック・ユニット 3 6 3 に返却する。I E U 1 0 4 から返却される 4 ビット例外ベクトルに入っている情報は、割り込みロジック・ユニット 3 6 3 から I E U 1 0 4 に渡された 4 ビット例外ベクトルと I E U 1 0 4 で引き起こされた例外を累積したものである。プリフェッチまたはイデコード時に検出された例外が原因で割り込み制御ユニット 3 6 3 にすでにストアされている情報があれば、その情報と一緒に I E U 1 0 4 から割り込み制御ユニット 3 6 3 に返却される情報の残余部分は、割り込み制御ユニット 3 6 3 が最高優先度同期例外の内容とそのトラップ番号を判断するのに十分である。

30

## 4) ハンドラ・ディスパッチとリターン：

割り込み受信確認信号がライン 3 4 0 経由で I E U から受信されるか、あるいはゼロ以外の例外ベクトルがライン 3 4 1 経由で受信された後、現 D P C がリターン・アドレスとして特殊レジスタ 4 1 2 ( 図 4 ) の一つである x P C レジスタに一時的にストアされる。現プロセッサ状態レジスタ ( P S R ) は先の P S R ( P P S R ) レジスタにもストアされ、そして現状態比較レジスタ ( C S R ) が特殊レジスタ 4 1 2 の中の旧状態比較レジスタ ( P C S R ) にセーブされる。

## 【 0 1 0 8 】

トラップ・ハンドラのアドレスはトラップ・ベース・レジスタ・アドレスにオフセットを加えたものとして計算される。P C ロジック・ユニット 2 7 0 はトラップ用に 2 つのベースレジスタを持ち、これらは共に特殊レジスタ 4 1 2 ( 図 4 ) の一部であり、以前に実行された特殊移動命令によって初期化される。大部分のトラップでは、ハンドラのアドレスを計算するために使用されるベース・レジスタはトラップ・ベース・レジスタ T B R である。

40

## 【 0 1 0 9 】

割り込み制御ユニット 3 6 3 は、現在保留中の最高優先度割り込みまたは例外を判断し、索引 ( l o o k - u p ) テーブルを通して、それに関連付けられたトラップ番号を判断する。これは、選択したベース・レジスタまでのオフセットとして、1 組の I N T \_ O F

50

F S E Tライン373を經由してプリフェッチPC制御ユニット364へ渡される。ベクトル・アドレスは、オフセット・ビットを下位ビットとしてTBRレジスタから得た上位ビットに連結するだけで求められるという利点がある。このため、加算器の遅延が防止される。(本明細書では $2^i$ ビットとは最下位を0番目としてi番目のビットのことである。)例えばトラップが8ビットの値により0から255までの番号が与えられるなら、ハンドラの開始アドレスは8ビットのトラップ番号を、22ビットのTBRにストアされた値の最後に連結することにより求められる。トラップ番号の更に下位に2桁の0ビットを付加すると、トラップ・ハンドラ・アドレスは常にワード境界上に置かれることになる。このようにして作られた連結ハンドラ・アドレスは入力373の一つとしてプリフェッチ・セクタPF\_\_PC Sel390(図4)へ送られ、次のアドレスとして選択され、そこから命令がプリフェッチされる。TBRレジスタを使用したトラップのベクトル・ハンドラ・アドレスは全て1ワードだけ離れている。したがって、トラップ・ハンドラ・アドレスにある命令は、長くなったトラップ処理ルーチンへの予備的ブランチ命令でなければならない。しかし、トラップにはシステム・パフォーマンスの低下を防止するために、その扱いに注意が必要なものがいくつかある。例えばTLBトラップは高速に実行させる必要がある。そのような理由から、好適実施例では、予備的ブランチの費用を払わないで、小型のトラップ・ハンドラを呼び出せるようにした高速トラップ・メカニズムが組み込まれている。さらに、高速トラップ・ハンドラはメモリにも、例えば、オン・チップROMにも独立に配置させることができるので、RAMの位置(ロケーション)に関連するメモリ・システム問題がなくなる。

10

20

#### 【0110】

好適実施例では、高速トラップとなるトラップは上述したVMU例外だけである。高速トラップの番号は他のトラップと区別され、0~7の範囲になっている。しかし、優先度はMMU例外と同じである。割り込み制御ユニット363は、高速トラップがそのとき保留中の最高優先度であると認めると、特殊レジスタ(FTB)から高速トラップ・ベース・レジスタ(FTB)を選択し、トラップ・オフセットと結合するためにライン416上に送出する。ライン373'經由でプリフェッチ・セクタPF\_\_PC\_\_Sel390へ送られた結果のベクトル・アドレスはFTBレジスタからの上位22ビットを連結したもので、そのあとに高速トラップ番号を表した3ビットが続き、そのあとに7個のゼロ・ビットが続いている。したがって、各高速トラップ・アドレスは128バイト、つまり32ワードだけ離れている。呼び出されると、プロセッサは開始ワードへブランチし、ブロックまたはそこからでたブランチ内でプログラムを実行させる。

30

#### 【0111】

32個またはそれ以下の命令で実現できる標準的なTLB処理ルーチンのような小さなプログラムの実行は、実際の実行処理ルーチンへの予備的ブランチが回避されるので、通常のトラップよりも高速化される。

#### 【0112】

好適実施例では、すべての命令は同じ4バイト長になっているが(つまり、4つのアドレス・ロケーションを占有する)、注目すべきことは命令が可変長になったマイクロプロセッサでも高速トラップ・メカニズムが利用できることである。この場合、高速トラップ・ベクトル・アドレス間には、マイクロプロセッサで使用できる最短長の命令を少なくとも2つ、好ましくは32個の平均サイズ命令を受け入れるだけの十分なスペースが設けられることは勿論である。勿論、マイクロプロセッサがトラップからのリターン命令を備えている場合には、ベクトル・アドレス間にはハンドラ内の少なくとも1つの他の命令をその命令におけるだけの十分なスペースを設けておく必要がある。

40

#### 【0113】

また、トラップ・ハンドラヘディスパッチすると、プロセッサはカーネル・モードと割り込み状態に入る。これと並行して状態比較レジスタ(CSR)のコピーが以前のキャリア状態レジスタ(PCSR)に置かれ、PSRのコピーが以前のPSR(PPSR)にストアされる。カーネルと割り込み状態モードはプロセッサ状況レジスタ(PSR)内のビ

50

ットで表される。現 P S R の割り込み状態ビットがセットされるとシャドウ・レジスタまたはトラップ・レジスタ R T [ 2 4 ] ~ R T [ 3 1 ] が上述および図 7 ( b ) に示すように見えるようになる。割り込みハンドラは新しいモードを P S R に書込むだけでカーネル・モードから出ることができるが、割り込み状態から出るためには、トラップからのリターン ( R T T ) 命令を実行する方法だけが唯一の方法である。

#### 【 0 1 1 4 】

I E U 1 0 4 が R T T 命令を実行すると、 P C S R は、 C S R レジスタに復元され、 P P S R レジスタは P S R レジスタに復元されるので、 P S R 内の割り込み状態ビットは自動的にクリアされる。 P F \_ P C \_ S E L セクタ 3 9 0 は特殊レジスタ・セット 4 1 2 中の特殊レジスタ x P C を次にそこからプリフェッチするアドレスとして選択する。 x P C はインクリメンタ 3 9 4 とバス 3 9 6 を通して M B U F \_ P F n P C または E B U F \_ P F n P C のどちらか該当する方に復元される。 x P C を E B U F \_ P F n P C に復元すべきか、または M B U F

10

P F n P C に復元すべきかの判断は、復元された後の P S R の「プロシージャ進行中」ビットにしたがって行われる。

#### 【 0 1 1 5 】

注目すべきことは、プロセッサはトラップとプロシージャ命令の両方のリターン・アドレスをストアするのに同じ特殊レジスタ x P C を使用しないことである。トラップのリターン・アドレスは上述したように特殊レジスタ x P C にストアされるが、プロシージャ命令のあとリターンする先のアドレスは別の特殊レジスタ u P C にストアされる。したがって、割り込み状態は、プロセッサが、プロシージャ命令で呼び出されたエミュレーション・ストリームを実行している間でも、使用可能のままになっている。他方、例外処理ルーチンはエミュレーション・ストリームが完了した後、例外ハンドラへ戻るためのアドレスをストアする特殊レジスタがないので、いずれのプロシージャ命令をも含んでいてはならない。

20

#### 5 ) ネスト :

ある種のプロセッサ状況情報は、トラップ・ハンドラ、特に C S R、P S R、リターン P C およびある意味では“ A ”レジスタ・セット r a [ 2 4 ] ~ r a [ 3 1 ] ヘディスパッチするとき自動的にバックアップが取られるが、他のコンテキスト情報は保護されていない。例えば浮動小数点状況レジスタ ( F S R ) の内容は自動的にバックアップがとられない。トラップ・ハンドラがこれらのレジスタを変更するためには、独自のバックアップを実行しなければならない。

30

#### 【 0 1 1 6 】

トラップ・ハンドラヘディスパッチするとき自動的に行われるバックアップが制限されているために、トラップのネストは自動的に行われぬ。トラップ・ハンドラは必要とするレジスタのバックアップをとり、割り込み条件をクリアし、トラップ処理のために必要な情報をシステム・レジスタから読取り、その情報を適当に処理する必要がある。割り込みは、トラップ・ハンドラヘディスパッチすると自動的に禁止される。処理を終えると、ハンドラはバックアップをとったレジスタを復元し、再び割り込みを可能にし、 R T T 命令を実行して割り込みから戻ることができる。

40

#### 【 0 1 1 7 】

ネストされたトラップを可能にするには、トラップ・ハンドラを第 1 部分と第 2 部分に分割する必要がある。第 1 部分では、割り込みが禁止されている間に、特殊レジスタ移動命令を使用して x P C をコピーし、トラップ・ハンドラが維持しているスタック上にプッシュしておく必要がある。次に、特殊レジスタ移動命令を使用して、トラップ・ハンドラの第 2 部分の先頭のアドレスを x P C に移し、トラップからのリターン命令 ( R T T ) を実行する必要がある。 R T T は割り込み状態を取り除き ( P P S R を P S R に復元することによって)、制御を x P C 内のアドレスに移す。 x P C には、ハンドラの第 2 部分のアドレスが入っている。第 2 部分はこの時点で割り込みを可能にして、割り込み可能モードで例外の処理を続けることができる。注目すべきことは、シャドウ・レジスタ R T [ 2 4

50

〕～RT〔31〕はこのハンドラの第1部分でのみ見ることができ、第2部分では見えないことである。したがって、第2部分ではハンドラは“ A ”レジスタ値がハンドラによって変更される可能性がある場合には、その値を予約しておく必要がある。トラップ処理ルーチンは終わったときは、バックアップにとったレジスタを全て復元し、元のxPCをトラップ・ハンドラ・スタップからポップし、それを特殊レジスタ移動命令を使用してxPC特殊レジスタに戻して、別のRTTを実行する必要がある。これにより、制御はメインまたはエミュレーション命令ストリームの中の該当命令に返される。

6) トラップ一覧表:

次の表Iは、好適実施例で認識されるトラップのトラップ番号、優先度および処理モードを示すものである。

10

【0118】

表 I

トラップ番号	処理モード	同期	トラップ名
0 - 1 2 7	通常	同期	トラップ命令
1 2 8	通常	同期	F P 例外
1 2 9	通常	同期	整数算術演算例外
1 3 0	通常	同期	MMU ( T L B 不一致または修正を除く )
1 3 5	通常	同期	不整列メモリアドレス
1 3 6	通常	同期	違法命令
1 3 7	通常	同期	特権命令
1 3 8	通常	同期	デバッグ例外
1 4 4	通常	非同期	性能モニタ
1 4 5	通常	非同期	タイマ/カウンタ
1 4 6	通常	非同期	メモリア/Oエラー
1 6 0 - 1 9 1	通常	非同期	ハードウェア割込み
1 9 2 - 2 5 3	予約		
2 5 4	通常	非同期	マシン・チェック
2 5 5	通常	非同期	N M I
0	高速トラップ	同期	高速MMU T L B 不一致
1	高速トラップ	同期	高速MMU T L B 修正
2 - 3	高速トラップ	同期	高速 ( 予約 )
4 - 7	高速トラップ	同期	高速 ( 予約 )

20

30

III. 命令実行ユニット

図5はIEU104の制御経路部分とデータ経路部分を示したものである。主要データ経路は、IFU102からの命令/オペランド・データ・バスを始端としている。データ・バスとして、即値オペランドはオペランド位置合わせユニット470へ送られ、レジスタ・ファイル( REG ARRAY ) 472に渡される。レジスタ・データはレジスタ・ファイル472からバイパス・ユニット474を通過して、レジスタ・ファイル出力バス476を経由し、分配バス480を経由して機能計算エレメント( F U<sub>0..n</sub> )の並列配列へ送られる。機能ユニット478<sub>0..n</sub>によって生成されたデータは出力バス482を経由して、バイパス・ユニット474またはレジスタ配列472または両方へ送り返される。

40

【0119】

ロード/ストア・ユニット484によってIEU104のデータ経路部分が完成される。ロード/ストア・ユニット484はIEU104とCCU106間のデータ転送の管理を担当する。具体的には、CCU106のデータ用キャッシュ134から取り出したロード・データはロード/ストア・ユニット484によってロード・データ・バス486を経由してレジスタ配列472へ転送される。CCU106のデータ用キャッシュにストアさ

50

れるデータは機能ユニットの分配バス480から受信される。

【0120】

IEU104の制御経路部分はIEUデータ経路を通る情報の送付、管理、およびその処理を行うことを担当する。本発明の好適実施例では、IEU制御経路は複数の命令の並行実行を管理する機能を備え、IEUデータ経路はIEU104のほぼすべてのデータ経路エレメント間の複数のデータ転送を独立して行う機能を備えている。IEU制御経路は命令/オペランド・バス124を経由して命令を受信すると、それに応じて動作する。具体的には命令の組はEデコード・ユニット490によって受信される。本発明の好適実施例では、Eデコード・ユニット490はIFIFOマスタ・レジスタ216、224に保持されている両方の命令の組を受信して、デコードする。8命令すべてのデコードの結果は、キャリア・チェッカ(CRYCHKR)ユニット492、依存性チェッカ(DEPCHKR)ユニット494、レジスタ改名ユニット(REGRENAME)496、命令発行(ISSUER)ユニット498、およびリタイア制御ユニット(RETIRETCLT)500へ送られる。

10

【0121】

キャリア・チェッカ・ユニット492はEデコード・ユニット490から制御ライン502を経由して、係属中の保留されている8命令に関するデコード化情報を受信する。キャリア・チェッカ492の機能は、保留されている命令のうち、プロセッサ状況ワードのキャリア・ビットに影響を与える、あるいはキャリア・ビットの状態に左右される命令を識別することである。この制御情報は、制御ライン504を経由して命令発行ユニット498へ送られる。

20

【0122】

保留状態の8命令によって使用されているレジスタ・ファイル472のレジスタを示しているデコード化情報は、制御ライン506を経由して直接にレジスタ改名ユニット496へ送られる。この情報は、依存関係チェッカ・ユニット494へも送られる。依存関係チェッカ・ユニット494の機能は、保留状態の命令のどれがレジスタをデータの宛先として参照しているか、もしあればどの命令がこれらの宛先レジスタのいずれかに依存しているかを判断することである。レジスタに依存する命令は、制御ライン508を経由してレジスタ改名ユニット496へ送られる制御信号によって識別される。

30

【0123】

最後にEデコード・ユニット490は保留状態の8命令の各々の特定の内容と機能を識別した制御情報を制御ライン510を経由して命令発行ユニット498へ送る。命令発行ユニット498はデータ経路資源、特に、保留状態の命令の実行のためにどの機能ユニットが使用できるかを判断することを担当する。アーキテクチャ100の好適実施例によれば、命令発行ユニット498はデータ経路資源が使用可能であること、キャリアとレジスタ依存関係の制約を条件として8個の保留状態命令のいずれかを順序外で実行できるようにする。レジスタ改名ユニット496は、実行できるよう適当に制約が解除された命令のビット・マップを制御ライン512を経由して、命令発行ユニット498へ送る。すでに実行された(完了した)命令およびレジスタまたはキャリアに依存する命令は論理的にビット・マップから除かれる。

40

【0124】

必要とする機能ユニット478<sub>0..n</sub>が使用可能であるかどうかに応じて、命令発行ユニット498は各システム・クロック・サイクルに複数の命令の実行を開始することができる。機能ユニット478<sub>0..n</sub>の状況は状況バス514を経由して、命令発行ユニット498へ送られる。命令の実行を開始し、開始後の実行管理を行うための制御信号は命令発行ユニット498から制御ライン516を経由してレジスタ改名ユニット496に送られ、また選択的に機能ユニット478<sub>0..n</sub>へ送られる。制御信号を受けると、レジスタ改名ユニット496はレジスタ選択信号をレジスタ・ファイル・アクセス制御バス518上に送出する。バス518上に送出された制御信号でどのレジスタが割り込み可能にされたかは、実行中の命令を選択することによって、およびレジスタ改名ユニット496がその特定

50

命令によって参照されたレジスタを判断することによって判断される。

【0125】

バイパス制御ユニット(BYPASS CTL)520は、一般的には制御ライン524上の制御信号を通してバイパス・データ・ルーティング・ユニット474の動作を制御する。バイパス制御ユニット520は機能ユニット478<sub>0-n</sub>の各々の状況をモニタし、制御ライン522を経由して、レジスタ改名ユニット496から送られてきたレジスタ参照に関連して、データをレジスタ・ファイル472から機能ユニット478<sub>0-n</sub>へ送るべきかどうか、あるいは機能ユニット478<sub>0-n</sub>から出力されるデータをバイパス・ユニット474経由で機能ユニット宛先バス480へ即時に送って、命令発行ユニット498によって選択された新発行の命令の実行のために使用できるかどうかを判断する。どちらの場合も、命令発行ユニット498は機能ユニット478<sub>0-n</sub>の各々への特定レジスタ・データを選択的に使用可能にすることによって、宛先バス480から機能ユニット478<sub>0-n</sub>へデータを送ることを直接に制御する。

10

【0126】

IEU制御経路の残りのユニットには、リタイア制御ユニット500、制御フロー制御(CF CTL)ユニット528、および完了制御(DONE CTL)ユニット536がある。リタイア制御ユニット500は順序外で実行された命令の実行を無効または確認するように動作する。ある命令が順序外で実行されると、先行命令も全てリタイアされたならば、その命令は、確認またはリタイアされることができる。現セット中の保留状態の8命令のどれが実行されたかの識別情報が制御ライン532上に送出されると、その識別情報に基づいて、リタイア制御ユニット500はバス518に接続された制御ライン534上に制御信号を送出して、レジスタ配列472にストアされた結果データを順序外で実行された命令の先行実行の結果として実効的に確認する。

20

【0127】

リタイア制御ユニット500は、各命令をリタイアするとき、PCインクリメント/サイズ制御信号を制御ライン344を経由してIFU102へ送る。複数の命令を順序外で実行でき、したがって、同時にリタイアする準備状態におくことができるので、リタイア制御ユニット500は同時にリタイアされた命令数に基づいてサイズ値を判断する。最後に、IFIFOマスタ・レジスタ224のすべての命令が実行され、リタイアされた場合は、リタイア制御ユニット500はIFIFO読み取り制御信号を制御ライン342を経由してIFU102へ送ってIFIFOユニット264のシフト・オペレーションを開始することにより、Eデコード・ユニット490に追加の4命令を実行保留命令として与える。

30

【0128】

制御フロー制御ユニット528は各条件付きブランチ命令の論理的ブランチ結果を検出するという特定化された機能を備えている。制御フロー制御ユニット528は現在保留中の条件付きブランチ命令の8ビット・ベクトルIDをEデコード・ユニット490から制御ライン510を経由して受信する。8ビット・ベクトル命令完了制御信号は、同じように完了制御ユニット540から制御ライン532を経由して受信される。この完了制御信号によって、制御フロー制御ユニット528は、条件付きブランチ命令が、条件付き制御フロー状況を判断するのに十分な箇所まで完了すると、それを判別することができる。保留中の条件付きブランチ命令の制御フロー状況結果は、その実行時に制御フロー制御ユニット528によってストアされる。条件付き制御フロー命令の結果を判断するために必要なデータは、レジスタ配列472内の一時状況レジスタから制御ライン530を経由して得られる。各条件付き制御フロー命令が実行されると、制御フロー制御ユニットは新しい制御フロー結果信号を制御ライン348を経由してIFU102に送る。好適実施例では、この制御フロー結果信号は2個の8ビット・ベクトルを含んでおり、このベクトルは、保留されている可能性のある8個の制御フロー命令のそれぞれのビット位置別の状況結果が分かっているかどうか、また、ビット位置の対応づけによって得られる対応する状況結果状態を定義している。

40

50

## 【 0 1 2 9 】

最後に、完了制御ユニット 5 4 0 は機能ユニット 4 7 8<sub>0..n</sub> の各々のオペレーションに関する実行状況をモニタするためのものである。機能ユニット 4 7 8<sub>0..n</sub> のいずれかが命令実行オペレーションの完了を通知すると、完了制御ユニット 5 4 0 は対応する完了制御信号を制御ライン 5 4 2 上に送出して、レジスタ改名ユニット 4 9 6、命令発行ユニット 4 9 8、リタイア制御ユニット 5 0 0 およびバイパス制御ユニット 5 2 0 にアラート（警告）する。

## 【 0 1 3 0 】

機能ユニット 4 7 8<sub>0..n</sub> を並列配列構成にすることにより、I E U 1 0 4 の制御の一貫性を向上している。命令を正しく認識して、実行のためのスケジュールするためには、個々の機能ユニット 4 7 8<sub>0..n</sub> の特性を命令発行ユニット 4 9 8 に知らせる必要がある。機能ユニット 4 7 8<sub>0..n</sub> は必要とする機能を実行するために必要な特定制御フロー・オペレーションを判別し、実行することを担当する。従って、命令発行ユニット 4 9 8 以外は I E U 制御ユニットには、命令の制御フロー処理を独立して知らせる必要はない。命令発行ユニット 4 9 8 と機能ユニット 4 7 8<sub>0..n</sub> は共同して、残りの制御フロー管理ユニット 4 9 6、5 0 0、5 2 0、5 2 8、5 4 0 に実行させる機能を必要な制御信号のプロンプトで知らせる。従って、機能ユニット 4 7 8<sub>0..n</sub> の特定の制御フロー・オペレーションの変更は I E U 1 0 4 の制御オペレーションに影響しない。さらに、既存の機能ユニット 4 7 8<sub>0..n</sub> の機能を強化する場合や、拡張精度浮動小数点乗算ユニットや拡張精度浮動小数点 A L U、高速フーリエ計算機能ユニット、三角関数計算ユニットなどの別の機能ユニット 4 7 8<sub>0..n</sub> を 1 つまたは 2 つ以上を追加する場合でも、命令発行ユニット 4 9 8 を若干変更するだけで済む。必要なる変更を行うには、E デコード・ユニット 4 9 0 によって隔離された対応する命令フィールドに基づいて、特定の命令を認識し、その命令を必要とする機能ユニット 4 7 8<sub>0..n</sub> とを関係づける必要がある。レジスタ・データの選択の制御、データのルーチング、命令完了とリタイアは、機能ユニット 4 7 8<sub>0..n</sub> の他の機能ユニットすべてに対して実行される他のすべての命令の処理と矛盾がないようになっている。

## 【 0 1 3 1 】

## A) I E U データ経路の詳細

I E U データ経路の中心となるエレメントはレジスタ・ファイル 4 7 2 である。しかし、本発明によれば I E U データ経路内には、個々の機能用に最適化された並列データ経路がいくつか用意されている。主要データ経路は整数と浮動少数点の 2 つである。各並列データ経路内では、レジスタ・ファイル 4 7 2 の一部がそのデータ経路内で行われるデータ操作をサポートするようになっている。

## 【 0 1 3 2 】

## 1) レジスタ・ファイルの詳細

図 6 ( a ) は、データ経路レジスタ・ファイル 5 5 0 の好適アーキテクチャの概要図である。データ経路レジスタ・ファイル 5 5 0 は一時バッファ 5 5 2、レジスタ・ファイル配列 5 5 4、入力セクタ 5 5 9、および出力セクタ 5 5 6 を含んでいる。最終的にレジスタ配列 5 5 4 へ送られるデータは結合データ入力バス 5 5 8 ' を経由して一時バッファ 5 5 2 によって最初に受信されるのが代表例である。つまり、データ経路レジスタ・ファイル 5 5 0 へ送られるデータはすべて入力セクタ 5 5 9 によって多重化されて、複数の入力バス 5 5 8 ( 好ましくは 2 つの ) から入力バス 5 5 8 ' 上に送出される。制御バス 5 1 8 上に送出されたレジスタ選択およびイネーブル制御信号は一時バッファ 5 5 2 内の受信データのレジスタ・ロケーションを選択する。一時バッファにストアされるデータを生成した命令がリタイアされると、再び制御バス 5 1 8 上に送出された制御信号は一時バッファ 5 5 2 からレジスタ・ファイル配列 5 5 4 内の論理的に対応づけられたレジスタへデータ・バス 5 6 0 を経路してデータを転送することを許可する。しかし、命令がリタイアされる前は、一時バッファ 5 5 2 にストアされたデータは一時バッファにストアされたデータをデータ・バス 5 6 0 のバイパス部分を経由して出力データ・セクタ 5 5 6 へ送ることにより、後続の命令の実行時に使用することが可能である。制御バス 5 1 8 経由で

送られる制御信号によって制御されるセクタ556は、一時バッファ552のレジスタからのデータとレジスタ・ファイル配列554のレジスタからのデータのどちらかを選択する。結果のデータはレジスタ・ファイル出力バス564上に送出される。また、実行中の命令が完了と同時にリタイアされる場合は、つまり、その命令が順序内で実行された場合は、結果データをバイパス延長部分558"を經由して直接にレジスタ配列554へ送るように指示することができる。

#### 【0133】

本発明の好適実施例によれば、各データ経路レジスタ・ファイル550は2つのレジスタ操作を同時に行なえるようになっている。したがって、入力バス558を通して2つの全レジスタ幅データ値を一時バッファ552に書き込むことができる。内部的には、一時バッファ552はマルチプレクサ配列になっているので、入力データを一時バッファ552内の任意の2レジスタへ同時に送ることができる。同様に、内部マルチプレクサにより一時バッファ552の任意の5レジスタを選択して、データをバス560上に出力することができる。レジスタ・ファイル配列554は同じように入出力マルチプレクサを備えているので、2つのレジスタを選択して、それぞれのデータを同時にバス560から受信することも、5つのレジスタを選択してバス562経由で送ることもできる。最後に、レジスタ・ファイル出力セクタ556は、バス560、562から受信した10レジスタ・データ値のうち、任意の5つがレジスタ・ファイル出力バス564上に同時に出力されるように実現するのが好ましい。

#### 【0134】

一時バッファ内のレジスタ・セットは図6(b)にその概要が示されている。レジスタ・セット552'は8個のシングル・ワード(32ビット)レジスタI0RD、I1RD、...、I7RDから構成されている。レジスタ・セット552'は4個のダブル・ワード・レジスタI0RD、I0RD+1(I4RD)、I1RD、I1RD+1(I5RD)、...、I3RD、I3RD+1(I7RD)のセットとして使用することも可能である。

#### 【0135】

本発明の好適実施例によれば、レジスタ・ファイル配列554内の各レジスタを重複して設ける代わりに、一時バッファ・レジスタ・セット552内のレジスタは2個のIFFIFOマスタ・レジスタ216、224内のそれぞれの命令の相対ロケーションに基づいて、レジスタ改名ユニット496によって参照される。本アーキテクチャ100で実現される各命令は、最高2つまでのレジスタまたは1つのダブル・ワード・レジスタを出力として参照して、命令の実行によって生成されたデータの宛先とすることができる。代表例として、命令は1つの出力レジスタだけを参照する。したがって、その位置を図6(c)に示しているように、8個の保留中命令のうち1つの出力レジスタを参照する命令2(I2)の場合は、データ宛先レジスタI2RDが選択されて、命令の実行によって生成されたデータを受け入れる。命令I2によって生成されたデータが後続の命令、例えばI5によって使用される場合は、I2RDレジスタにストアされたデータはバス560を經由して転送され、結果のデータは一時バッファ552に送り返されて、I5RDで示したレジスタにストアされる。特に、命令I5は命令I2によって決まるので、命令I5は、I2からの結果データが得られるまでは実行することができない。しかし、理解されるように、命令I5は必要とする入力データを一時バッファ552'の命令I2のデータ・ロケーションから得れば、命令I2のリタイア前に実行することが可能である。

#### 【0136】

最後に、命令I2がリタイアされると、レジスタI2RDからのデータは、リタイア箇所の命令の論理位置から判断されて、レジスタ・ファイル配列554内のレジスタ・ロケーションに書かれる。すなわち、リタイア制御ユニット500は、制御ライン510経由でEデコード・ユニット490から与えられたレジスタ参照フィールド・データからレジスタ・ファイル配列内の宛先レジスタのアドレスを判断する。命令I0-3がリタイアされると、I4RD-I7RDに入っている値は、IFFIFOユニット264のシフトと同時にシフトされて、I0RD-I3RDに移される。

## 【0137】

命令 I<sub>2</sub> からダブル・ワード結果値が得られる場合は、さらに複雑になる。本発明の好適実施例によれば、ロケーション I<sub>2RD</sub> と I<sub>6RD</sub> の組合せが、命令 I<sub>2</sub> がリタイアされるか、さもなければキャンセルまで、その命令から得た結果データをストアしておくために使用される。好適実施例では、命令 I<sub>4.7</sub> の実行は命令 I<sub>0.3</sub> のいずれかによるダブル・ワード出力の参照がレジスタ改名ユニット 496 によって検出された場合には、保留される。これにより、一時バッファ 552' 全体をダブル・ワード・レジスタのシングル・ランクとして使用することが可能になる。命令 I<sub>0.3</sub> がリタイアされると、一時バッファ 552' はシングル・ワード・レジスタの 2 ランクとして再び使用することができる。さらに、いずれかの命令 I<sub>4.7</sub> の実行は、ダブル・ワード出力レジスタが必要な場合には、命令が対応する I<sub>0.3</sub> にシフトされるまで保留される。 10

## 【0138】

レジスタ・ファイル配列 554 の論理変性は図 7 (a) ~ 図 7 (b) に示されている。本発明の好適実施例によれば、整数データ経路用のレジスタ・ファイル配列 554 は 40 個の 32 ビット幅レジスタから構成されている。このレジスタ・セットはレジスタ・セット "A" を構成し、ベース・レジスタ・セット ra [ 0 . . 23 ] 565、汎用レジスタ ra [ 24 . . 31 ] 566 からなるトップ・セット、および 8 個の汎用トラップ・レジスタ ra [ 24 . . 31 ] からなるシャドウ・レジスタ・セットとして編成されている。通常の実行では、汎用レジスタ ra [ 0 . . 31 ] 565、566 は整数データ経路用のレジスタ・ファイル配列のアクティブ "A" レジスタ・セットを構成している。 20

## 【0139】

図 7 (b) に示すように、トラップ・レジスタ ra [ 24 . . 31 ] 567 をスワップしてアクティブ・レジスタ・セット "A" に移しておけば、レジスタ ra [ 0 . . 23 ] 565 のアクティブ・ベース・セットと一緒にアクセスすることが可能である。"A" レジスタ・セットのこの構成は、割り込みの受信が確認されるか、例外トラップ処理ルーチンが実行されると、選択される。レジスタ・セット "A" のこの状態は、割り込み許可命令の実行またはトラップからのリターン命令の実行によって図 7 (a) に示す状態に明示によって戻るまで維持される。

## 【0140】

アーキテクチャ 100 によって実現された本発明の好適実施例では、浮動小数点データ経路は図 8 にその概要を示すように拡張精度レジスタ・ファイル配列 572 を使用する。レジスタ・ファイル配列 572 は、各々が 64 ビット幅の 32 個のレジスタ rf [ 0 . . 31 ] から構成されている。浮動小数点レジスタ・ファイル 572 は整数レジスタ rb [ 0 . . 31 ] の "B" セットとして論理的に参照することも可能である。アーキテクチャ 100 では、この "B" セットのレジスタは浮動小数点レジスタ rf [ 0 . . 31 ] の各々の下位 32 ビットに相当している。

## 【0141】

第 3 のデータ経路を表すものとして、ブール演算子レジスタ・セット 574 が図 9 に示すように設けられている。これは、ブール演算の論理結果をストアするこの "C" レジスタ・セット 574 は 32 個の 1 ビット・レジスタ rc [ 0 . . 31 ] から構成されている。ブール・レジスタ・セット 574 のオペレーションは、ブール演算の結果をブール・レジスタ・セット 574 の任意の命令選択レジスタへ送ることができる点でユニークである。これは、等しい、等しくない、より大、その他単純なブール状況値などの条件を表す 1 ビット・フラグをストアするシングル・プロセッサ状況ワード・レジスタを使用するのと対照的である。 40

## 【0142】

浮動小数点レジスタ・セット 572 とブール・レジスタ・セット 574 は双方共、図 6 (b) に示す整数一時バッファ 552 と同じアーキテクチャの一時バッファによって補数 50

がとられる。基本的違いは、一時バッファ・レジスタの幅が補数をとるレジスタ・ファイル配列 5 7 2、5 7 4 の幅と同じになるように定義されていることである。好適実施例では、幅はそれぞれ 6 4 ビットと 1 ビットになっている。

【 0 1 4 3 】

多数の追加の特殊レジスタが、レジスタ配列 4 7 2 に少なくとも論理的に存在している。図 7 ( c ) に示すように、レジスタ配列 4 7 2 に物理的に存在するレジスタはカーネル・スタック・ポインタ ( k e r n e l s t a c k p o i n t e r ) 5 6 8、プロセッサ状態レジスタ ( P S R ) 5 6 9、旧プロセッサ状態レジスタ ( P P S R ) 5 7 0 および 8 個の一時プロセッサ状態レジスタの配列 ( t P S R [ 0 . . 7 ] ) 5 7 1 からなっている。残りの特殊レジスタはアーキテクチャ 1 0 0 の各所に分散している。特殊アドレスおよびデータ・バス 3 5 4 はデータを選択して、特殊レジスタおよび “ A ” と “ B ” レジスタ・セット間で転送するためのものである。特殊レジスタ移動命令は “ A ” または “ B ” レジスタ・セットからレジスタを選択し、転送の方向を選択し、特殊レジスタのアドレス I D を指定するためのものである。

10

【 0 1 4 4 】

カーネル・スタック・ポインタ・レジスタとプロセッサ状態レジスタは、他の特殊レジスタとは異なっている。カーネル・スタック・ポインタはカーネル状態にあるとき、標準のレジスタ間移動命令を実行することによってアクセス可能である。一時プロセッサ状態レジスタは直接にアクセスすることはできない。その代わりに、このレジスタ配列はプロセッサ状態レジスタの値を伝播して、順序外で実行される命令で使用できるようにする継承メカニズム ( i n h e r i t a n c e m e c h a n i s m ) を実現するために使用される。初期伝播値はプロセッサ状態レジスタの値である。つまり、最後にリタイアされた命令から得た値である。この初期値は一時プロセッサ状態レジスタから前方向に伝播され、順序外で実行される命令が対応する位置にある一時プロセッサ状態レジスタ内の値をアクセスできるようにする。命令が依存し、変更できる条件コード・ビットは、その命令がもつ特性によって定義される。命令が依存関係、レジスタまたは条件コードによって制約されないことが、レジスタ依存関係チェッカ・ユニット 4 9 4 とキャリー依存関係チェッカ 4 9 2 によって判断された場合は、命令は順序外で実行することができる。

20

【 0 1 4 5 】

プロセッサ状態レジスタの条件コード・ビットの変更は論理的に対応する一時プロセッサ状態レジスタに指示される。具体的には、変更の可能性のあるビットだけが一時プロセッサ状態レジスタに入っている値に適用され、上位のすべての一時プロセッサ状態レジスタに伝播される。その結果、順序外で実行されるすべての命令は介在する P S R 変更命令によって適切に変更されたプロセッサ状態レジスタ値から実行される。命令がリタイアされたときは、対応する一時プロセッサ状態レジスタ値だけが P S R レジスタ 5 6 9 に転送される。

30

その他の特殊レジスタは表 I I に説明されている。

【 0 1 4 6 】

表 I I

40

特殊レジスタ

特殊移動		
レジスタ	R / W	説明
P C	R	プログラム・カウンタ：一般的には P C は現在実行中のプログラム命令ストリームの次のアドレスを格納している
I F __ P C	R / W	I F U プログラム・カウンタ： I F __ P C は正確な次の実行アドレスを格納している
P F n P C	R	プリフェッチ・プログラム・カウンタ：M B U F

50

			、 T B U F および E B U F _ P F n P C はそれぞれのプリフェッチ命令ストリームの次のプリフェッチ命令アドレスを格納している	
u P C	R / W		マイクロ・プログラム・カウンタ：プロシージャ命令のあとに続く命令のアドレスを格納している。これはプロシージャ命令がリターンしたとき最初に実行される命令のアドレスである	
x P C	R / W		割り込み / 例外プログラム・カウンタ：割り込みまたは例外（または両方）のリターン・アドレスを格納している。リターン・アドレスはトラップ発生時の I F _ P C アドレスである	10
T B R	W		トラップ・ベース・アドレス：トラップ処理ルーチンヘディスパッチするとき使用されるベクトル・テーブルのベース・アドレス。各エントリは1ワード長である。割り込みロジック・ユニット363から与えられるトラップ番号は、このアドレスが指しているテーブルまでのインデックスとして使用される	
F T B	W		高速トラップ・ベース・レジスタ：即時トラップ処理ルーチン・テーブルのベース・レジスタ。各テーブル・エントリは32ワードであり、トラップ処理ルーチンを直接に実行するために使用される。割り込みロジック・ユニット363から与えられるトラップ番号を32倍したものは、このアドレスが指しているテーブルまでのオフセットとして使用される	20
P B R	W		プロシージャ・ベース・レジスタ：プロシージャ・ルーチンヘディスパッチするとき使用されるベクトル・テーブルのベース・アドレス。各エントリは1ワード長であり、4ワード境界に位置合わせされている。プロシージャ命令フィールドとして与えられるプロシージャ番号はこのアドレスが指しているテーブルまでのインデックスとして使用される	30
P S R	R / W		プロセッサ状態レジスタ：プロセッサ状況ワードを格納している。状況データ・ビットは、キャリー、オーバーフロー、ゼロ、負、プロセッサ・モード、現割り込みレベル、実行中のプロシージャ・ルーチン、0による除算、オーバフロー例外、ハードウェア機能割り込み可能、プロシージャ割り込み可能、割り込み可能などのビットがある。	40
P P S R	R / W		旧プロセッサ状態レジスタ：命令が正しく完了するか、割り込みまたはトラップが引き起こされると、P S R からロードされる	
C S R	R / W		状態比較（ブール）レジスタ：シングル・ワードとしてアクセス可能なブール・レジスタ・セット	
P C S R	R / W		旧状態比較レジスタ：命令が正しく完了するか、割り込みまたはトラップが引き起こされると、C S R からロードされる。	

## 2) 整数データ経路の詳細

本発明の好適実施例に従って構築される I E U 1 0 4 の整数データ経路は、図 1 0 に示されている。説明の便宜上、整数データ経路 5 8 0 と結ばれる多数の制御経路は、図には示していない。これらの接続関係は図 5 を参照して説明したとおりである。

### 【 0 1 4 8 】

データ経路 5 8 0 の入力データは、位置合わせユニット 5 8 2、5 8 4 および整数ロード/ストア・ユニット 5 8 6 から得られる。整数即値 ( i n t e g e r i m m e d i a t e ) データ値は、最初は命令埋め込み ( e m b e d d e d ) データ・フィールドとして与えられ、バス 5 8 8 経由でオペランド・ユニット 4 7 0 から得られる。位置合わせユニット 5 8 2 は整数データ値を隔離し、その結果値が出力バス 5 9 0 を経由してマルチプレクサ 5 9 2 へ送られる。マルチプレクサ 5 9 2 への別の入力は、特殊レジスタ・アドレスとデータ・バス 3 5 4 である。

10

### 【 0 1 4 9 】

命令ストリームから得られる即値 ( i m m e d i a t e ) オペランドも、データ・バス 5 9 4 経由でオペランド・ユニット 4 7 0 から得られる。これらの値は、出力バス 5 9 6 上に送出される前に、位置合わせユニット 5 8 4 によって再度右寄せされる。

### 【 0 1 5 0 】

整数ロード/ストア・ユニット 5 8 6 は外部データ・バス 5 9 8 を通して C C U 1 0 6 と双方向でやりとりする。I E U 1 0 4 へのインバウンド・データは整数ロード/ストア・ユニット 5 8 6 から入力データ・バス 6 0 0 を経由して入力ラッチ 6 0 2 へ転送される。マルチプレクサ 5 9 2 とラッチ 6 0 2 からの出力データは、マルチプレクサ 6 0 8 のマルチプレクサ入力バス 6 0 4、6 0 6 上に送出される。機能ユニット出力バス 4 8 2 ' からのデータもマルチプレクサ 6 0 8 に送られる。このマルチプレクサ 6 0 8 はアーキテクチャ 1 0 0 の好適実施例では、データを同時に出力マルチプレクサ・バス 6 1 0 へ送る 2 つの通路を備えている。さらに、マルチプレクサ 6 0 8 を通るデータ転送は、システム・クロックの各半サイクル以内に完了することができる。本アーキテクチャ 1 0 0 で実現される大部分の命令は、1 つの宛先レジスタを利用するので、最大 4 つまでの命令によって各システム・クロック・サイクルの間データを一時バッファ 6 1 2 へ送ることができる。

20

### 【 0 1 5 1 】

一時バッファ 6 1 2 からのデータは一時レジスタ出力バス 6 1 6 を経由して整数レジスタ・ファイル配列 6 1 4 へ、あるいは代替一時バッファ・レジスタ・バス 6 1 8 を経由して出力マルチプレクサ 6 2 0 へ転送することができる。整数レジスタ配列出力バス 6 2 2 は整数レジスタ・データをマルチプレクサ 6 2 0 へ転送することができる。一時バッファ 6 1 2 と整数レジスタ・ファイル配列 6 1 4 に接続された出力バスは、それぞれ 5 個のレジスタ値を同時に出力することを可能にする。つまり、合計 5 個までのソース・レジスタを参照する 2 つの命令を同時に出すことができる。一時バッファ 6 1 2、レジスタ・ファイル配列 6 1 4 およびマルチプレクサ 6 2 0 はアウトバウンド・レジスタ・データの転送を半システム・クロック・サイクルごとに行うことを可能にする。したがって、最高 4 個までの整数および浮動小数点命令を各クロック・サイクルの間に出すことができる。

30

### 【 0 1 5 2 】

マルチプレクサ 6 2 0 はアウトバウンド・レジスタ・データ値をレジスタ・ファイル配列 6 1 4 からあるいは一時バッファ 6 1 2 から直接に選択する働きをする。これにより、以前に順序外で実行された命令に依存する順序外実行命令を I E U 1 0 4 によって実行させることができる。これにより、保留状態の命令を順序外で実行することによって I E U 整数データ経路の実行スループット能力を最大化すると共に、順序外のデータ結果を、実行されリタイアされた命令から得たデータ結果から正確に分離するという 2 目標を容易に達成することができる。マシンの正確な状態を復元する必要があるような割り込みや他の例外条件が起これば、本発明によれば一時バッファ 6 1 2 に存在するデータ値を簡単にクリアすることができる。したがって、レジスタ・ファイル配列 6 1 4 は、割り込みまたは他の例外条件が発生する以前に完了し、リタイアされた命令の実行によってのみ得られた

40

50

データ値を正確に収めたままになっている。

【 0 1 5 3 】

マルチプレクサ 6 2 0 の各半システム・サイクル・オペレーション時に選択されたレジスタ・データ値は最高 5 つまでがマルチプレクサ出力バス 6 2 4 を経由して整数バイパス・ユニット 6 2 6 へ送られる。このバイパス・ユニット 6 2 6 は基本的にマルチプレクサが並列の配列からなり、その入力のいずれかに現れたデータをその出力のいずれかへ送ることができる。バイパス・ユニット 6 2 6 の入力は、マルチプレクサ 5 9 2 から出力バス 6 0 4 を経由する特殊レジスタ・アドレス指定データ値または即値の整数値、バス 6 2 4 上に送出される最高 5 つまでのレジスタ・データ値、整数ロード/ストア・ユニット 5 8 6 からダブル整数バス 6 0 0 を経由するロード・オペランド・データ、その出力バス 5 9 6 を経由して位置合わせユニット 5 8 4 から得た即値オペランド値、最後に機能ユニット出力バス 4 8 2 からのバイパス・データ経路からなっている。このバイパス経由とデータ・バス 4 8 2 はシステム・クロック・サイクルごとに 4 個のレジスタ値を同時に転送することができる。

10

【 0 1 5 4 】

データはバイパス・ユニット 6 2 6 から浮動小数点データ・バスに接続された整数バイパス・バス 6 2 8 上に出力されて、最高 5 つまでのレジスタ・データ値を同時に転送する機能をもつ 2 つのオペランド・データ・バスと、整数ロード/ストア・ユニット 5 8 6 へデータを送るために使用されるストア・データ・バス 6 3 2 へ送られる。

【 0 1 5 5 】

機能ユニット分配バス 4 8 0 はルータ・ユニット 6 3 4 のオペレーションを通して実現されている。また、ルータ・ユニット 6 3 4 はその入力から受信された 5 個のレジスタ値を整数データ通路に設けられた機能ユニットへ送ることを可能にする並列のマルチプレクサ配列によって実現される。具体的には、ルータ・ユニット 6 3 4 はバイパス・ユニット 6 2 6 からバス 6 3 0 を経由して送られてきた 5 個のレジスタ・データ値、アドレス・バス 3 5 2 を経由して送られてきた現 I F \_ P C アドレス値、P C 制御ユニット 3 6 2 によって判断され、ライン 3 7 8 ' 上に送出された制御フロー・オフセット値を受信する。ルータ・ユニット 6 3 4 は、浮動小数点データ経路内に設けられたバイパス・ユニットから取り出されたオペランド・データ値をデータバス 6 3 6 を経由して受信することもできる ( オプション ) 。

20

30

【 0 1 5 6 】

ルータ・ユニット 6 3 4 によって受信されたレジスタ・データ値は、特殊レジスタ・アドレスおよびデータ・バス 3 5 4 上を転送されて、機能ユニット 6 4 0、6 4 2、6 4 4 へ送られる。具体的には、ルータ・ユニット 6 3 4 は最高 3 つまでのレジスタ・オペランド値をルータ出力バス 6 4 6、6 4 8、6 5 0 を経由して機能ユニット 6 4 0、6 4 2、6 4 4 の各々へ送る機能を備えている。本アーキテクチャ 1 0 0 の一般的アーキテクチャによれば、最高 2 つまでの命令を同時に機能ユニット 6 4 0、6 4 2、6 4 4 に対して出すことが可能である。本発明の好適実施例によれば、3 つの専用整数機能ユニットに、それぞれプログラマブル・シフト機能と 2 つの算術演算ロジック・ユニット機能を持たせることができる。

40

【 0 1 5 7 】

A L U 0 機能ユニット 6 4 4、A L U 1 機能ユニット 6 4 2 及びシフタ機能ユニット 6 4 0 はそれぞれの出力レジスタ・データを機能ユニット・バス 4 8 2 ' 上に送出する。A L U 0 とシフタ機能ユニット 6 4 4、6 4 0 から得た出力データも浮動小数点データ経路に接続された共用整数機能ユニット・バス 6 5 0 上に送出される。類似の浮動小数点機能ユニット出力値データ・バス 6 5 2 が、浮動少数点データ経路から機能ユニット出力バス 4 8 2 ' へ設けられている。

【 0 1 5 8 】

A L U 0 機能ユニット 6 4 4 は I F U 1 0 2 のプリフェッチ操作と整数ロード/ストア・ユニット 5 8 6 のデータ操作の両方をサポートするために仮想アドレス値を生成する場

50

合にも使用される。ALU0機能ユニット644によって計算された仮想アドレス値はIFU102のターゲット・アドレス・バス346とCCU106の両方に接続された出力バス654上に送られ、実行ユニットの物理アドレス(EXPADDR)が得られる。ラッチ656は、ALU0機能ユニット644によって生成されたアドレスの仮想化部分をストアするためのものである。アドレスのこの仮想化部分は出力バス658上に送られて、VMU108へ送られる。

### 3) 浮動小数点データ経路の詳細

次に、図11は浮動小数点データ経路を示したものである。初期データは、この場合も、即値整数オペランド・バス588、即値オペランド・バス594および特殊レジスタ・アドレス・データ・バス354を含む複数のソースから受信される。外部データの最終的  
10  
ソースは外部データバス598を通してCCU106に接続された浮動小数点ロード/ストア・ユニット622である。

#### 【0159】

即値整数オペランドは、位置合わせ出力データ・バス668を経由してマルチプレクサ666に渡す前に整数データ・フィールドを右寄せする働きをする位置合わせユニット664によって受信される。マルチプレクサ666は特殊レジスタ・アドレス・データ・バス354も受信する。即値オペランドは第2の位置合わせユニット670へ送られ、右寄せされてから出力バス672上に送られる。浮動少数点ロード/ストア・ユニット662からのインバウンド・データ(inbound data)は、ロード・データ・バス676からラッチ674によって受信される。マルチプレクサ666、ラッチ674および機能ユニット・データ・リターン・バス482からのデータはマルチプレクサ678  
20  
の入力から受信される。マルチプレクサ678は選択可能なデータ経路を備え、2つのレジスタ・データ値がシステム・クロックの半サイクルごとに、マルチプレクサ出力バス682を経由して一時バッファ680に書き込まれることを可能にする。一時バッファ680は図6(b)に示す一時バッファ552'と論理的に同じレジスタ・セットを備えている。一時バッファ680はさらに最高5個までのレジスタ・データ値を一時バッファ680から読み取って、データ・バス686を経由して浮動小数点レジスタ・ファイル配列684と、出力データ・バス690を経由して出力マルチプレクサ688へ送ることができる。マルチプレクサ688は、データ・バス692を経由して、浮動小数点ファイル配列684から最高5個までのレジスタ・データ値も同時に受信する。マルチプレクサ688  
30  
は最高5個までのレジスタ・データ値を選択して、データ・バス696を経由してバイパス・ユニット694へ同時に転送する働きをする。バイパス・ユニット694は、データ・バス672、マルチプレクサ666からの出力データ・バス698、ロード・データ・バス676および機能ユニット・データ・リターン・バス482のバイパス延長部分を経由して、位置合わせユニット670から与えられた即値オペランド値も受信する。バイパス・ユニット694は最高5個までのレジスタ・オペランドデータ値を同時に選択して、バイパス・ユニット・出力バス700、浮動小数点ロード/ストア・ユニット662に接続されたストア・データ・バス702、および整数データ経路580のルータ・ユニット634に接続された浮動小数点バイパス・バス636上に出力するように働く。

#### 【0160】

浮動小数点ルータ・ユニット704は、バイパス・ユニット出力バス700と整数データ経路バイパス・バス628とそれぞれの機能ユニット712、714、716に接続された機能ユニット入力バス706、708、710との間で同時にデータ経路を選択できる機能を備えている。  
40

#### 【0161】

アーキテクチャ100の好適実施例による入力バス706、708、710の各々は最高3個までのレジスタ・オペランド・データ値を機能ユニット712、714、716の各々へ同時に転送することが可能である。これらの機能ユニット712、714、716の出力バスは機能ユニット・データ・リターン・バス482に結合され、データをレジスタ・ファイル入力マルチプレクサ678へ戻すようになっている。整数データ経路機能  
50

ユニット出力バス650を機能ユニット・データ・リターン・バス482"に接続するために設けることも可能である。本発明のアーキテクチャ100によれば、マルチプレクサ機能ユニット712と浮動小数点ALU714の機能ユニット出力バスを浮動小数点データ経路機能ユニット・バス652を経由して整数データ経路500の機能ユニット・データ・リターン・バス482'に接続することが可能である。

#### 【0162】

##### 4) ブール・レジスタ・データ経路の詳細

ブール演算データ経路720は図12に示されている。このデータ経路720は基本的に2種類の命令の実行をサポートするために利用される。最初のタイプはオペランド比較命令であり、この命令では、整数レジスタ・セットと浮動小数点レジスタ・セットから選択された、あるいは即値オペランドとして与えられた2つのオペランドが、ALU機能ユニットの1つで整数と浮動小数点データ経路を減算することによって比較される。この比較は、ALU機能ユニット642、644、714、716のいずれかによる減算によって行われ、その結果の符号とゼロ状況ビットは入力セレクタと比較演算子結合ユニット722へ送られる。このユニット722は、制御信号を指定した命令をEデコード・ユニット490から受け取るとALU機能ユニット642、644、714、716の出力を選択し、符号およびゼロ・ビットを結合し、ブール比較結果値を抽出する。出力バス723を通して比較演算の結果を入力マルチプレクサ726とバイパス・ユニット742へ同時に転送することができる。整数および浮動小数点データ経路と同じように、バイパスユニット742は並列のマルチプレクサ配列として実現され、バイパス・ユニット742の入力間で複数のデータ経路を選択して、複数の出力と結ぶことができる。バイパス・ユニット742の他の入力にはブール演算結果リターン・データ・バス724とデータ・バス744上の2つのブール・オペランドからなっている。バイパス・ユニット742は最高2つまでの同時に実行中のブール命令を表したブール・オペランドを、オペランド・バス748を経由してブール演算機能ユニット746へ転送することができる。また、バイパス・ユニット746は最高2個までのシングル・ビット・ブール・オペランド・ビット(CF0、CF1)を制御フロー結果制御ライン750、752を経由して同時に転送することができる。

#### 【0163】

ブール演算データ経路の残り部分は、比較結果バス723とブール結果バス724上に送出された比較およびブール演算結果値を、その入力として受信する入力マルチプレクサ726を含んでいる。このバス724は最高2個までのブール結果ビットを同時にマルチプレクサ726へ転送することができる。さらに、最高2個までの比較結果ビットをバス723を経由してマルチプレクサ726へ転送することができる。マルチプレクサ726はマルチプレクサの入力端に現れた任意の2個の信号ビットをマルチプレクサの出力端を経由して、システム・クロックの各半サイクル時にブール演算一時バッファ728へ転送することができる。一時バッファ728は2つの重要な点が異なることを除けば、図6(b)に示した一時バッファ752'と論理的に同じである。第1の相違点は、一時バッファ728内の各レジスタ・エントリがシングル・ビットからなることである。第2の相違点は8個の保留中命令スロットの各々に一つのレジスタだけが設けられていることである。これは、ブール演算の結果全部が定義によって1つの結果ビットによって定義されるためである。

#### 【0164】

一時バッファ728は最高4個までの出力オペランド値を同時に出力する。これにより、各々2つのソース・レジスタへのアクセスを必要とする2個のブール命令を同時に実行させることができる。4個のブール・レジスタ値はシステム・クロックの各半サイクルごとにオペランド・バス736上に送出し、マルチプレクサ738へあるいはブール・オペランド・データバス734を経由してブール・レジスタ・ファイル配列732へ転送することができる。ブール・レジスタ・ファイル配列732は、図9に論理的に示すように、1個の32ビット幅データ・レジスタであり、任意に組合せた最高4個までのシングル・

10

20

30

40

50

ビット・ロケーションを一時バッファ728からのデータで修正し、システム・クロックの各半サイクルごとにブール・レジスタ・ファイル配列732から読み取って出力バス740上に送出することができる。マルチプレクサ738はバス736、740経由でその出力端から受信したブール・オペランドの任意のペアをオペランド出力バス744上に送出してバイパス・ユニット742へ転送する。

【0165】

ブール演算機能ユニット746は2個のソース値についてブール演算を幅広く実行する機能を備えている。比較命令の場合には、ソース値は整数および浮動小数点レジスタ・セットのいずれかから得たペアのオペランドとIEU104へ送られる任意の即値オペランドであり、ブール命令の場合は、ブール・レジスタ・オペランドの任意の2つである。表IIIと表IVは、本発明のアーキテクチャ100の好適実施例における論理比較演算を示すものである。表Vは本発明のアーキテクチャ100の好適実施例における直接ブール演算を示すものである。表III-Vに示されている命令条件コードと機能コードは対応する命令のセグメントを表している。また、命令はペアのソース・オペランド・レジスタと対応するブール演算結果をストアするための宛先ブール・レジスタを指定する。

10

【0166】

表 III

-----

整数の比較

条件*	記号	命令 条件コード
rs1はrs2より大	>	0000
rs1はrs2より大か等しい	>=	0001
rs1はrs2より小	<	0010
rs1はrs2より小か等しい	<=	0011
rs1はrs2と等しくない	?	0100
rs1はrs2と等しい	=	0101
予備		0110
無条件		1111

20

-----

\*rs = レジスタ・ソース

30

表 IV

-----

浮動小数点の比較

条件	記号	命令 条件コード
rs1はrs2より大	>	0000
rs1はrs2より大か等しい	>=	0001
rs1はrs2より小	<	0010
rs1はrs2より小か等しい	<=	0011
rs1はrs2と等しくない	?	0100
rs1はrs2と等しい	=	0101
無順序	?	1000
無順序またはrs1はrs2より大	?>	1001
無順序、rs1はrs2より大か等しい	?>=	1010
無順序またはrs1はrs2より小	?<	1011
無順序、rs1はrs2より小か等しい	?<=	1100
無順序またはrs1はrs2と等しい	?=	1101
予備		1110 - 1111

40

-----

50

表 V

ブール演算		
演算 *	記号	命令 機能コード
0	Z e r o	0 0 0 0
b s 1 & b s 2	A N D	0 0 0 1
b s 1 & - b s 2	A N N 2	0 0 1 0
b s 1	b s 1	0 0 1 1
- b s 1 & b s 2	A N N 1	0 1 0 0
b s 2	b s 2	0 1 0 1
b s 1 - b s 2	X O R	0 1 1 0
b s 1 b s 2	O R	0 1 1 1
- b s 1 & - b s 2	N O R	1 0 0 0
- b s 1 - b s 2	X N O R	1 0 0 1
- b s 2	N O T 2	1 0 1 0
b s 1 - b s 2	O R N 2	1 0 1 1
- b s 1	N O T 1	1 1 0 0
- b s 1 b s 2	O R N 1	1 1 0 1
- b s 1 - b s 2	N A N D	1 1 1 0
1	O N E	1 1 1 1

\* b s = ブール・ソース・レジスタ。

#### 【 0 1 6 7 】

##### B) ロード/ストア制御ユニット

図 1 3 はロード/ストア・ユニット 7 6 0 の例を示したものである。データ経路 5 8 0、6 6 0 に別々に示されているが、ロード/ストア・ユニット 5 8 6、6 6 2 は一つの共用ロード/ストア・ユニット 7 6 0 として実現するのが好ましい。それぞれのデータ経路 5 8 0、6 6 0 からのインタフェースはアドレス・バス 7 6 2 およびロードとストア・データ・バス 7 6 4 ( 6 0 0、6 7 6 )、7 6 6 ( 6 3 2、7 0 2 ) を経由している。

#### 【 0 1 6 8 】

ロード/ストア・ユニット 7 6 0 によって使用されるアドレスは、I F U 1 0 2 および I E U 1 0 4 の残り部分で使用される仮想アドレスとは対照的に、物理アドレスである。I F U 1 0 2 は仮想アドレスで動作し、C C U 1 0 6 と V M U 1 0 8 間の調整に依存して物理アドレスを生成するのに対し、I E U 1 0 4 ではロード/ストア・ユニット 7 6 0 を物理アドレス・モードで直接に動作させる必要がある。この要件が必要になるのは、順序外で実行されるために、物理アドレス・データとストア・オペレーションがオーバーラップするような命令が存在する場合、および C C U 1 0 6 からロード/ストア・ユニット 7 6 0 への順序外のデータ・リターンが存在する場合に、データ保全性を保つためである。データ保全性を保つためにロード/ストア・ユニット 7 6 0 はストア命令が I E U 1 0 4 によってリタイアされるまで、ストア命令から得たデータをバッファに置いておく。その結果、ロード/ストア・ユニット 7 6 0 によってバッファに置かれたストア・データはロード/ストア・ユニット 7 6 0 に一つだけ存在することができる。実行されたがリタイアされていないストア命令と同じ物理アドレスを参照するロード命令は、ストア命令が実際にリタイアされるまで実行が遅延される。その時点で、ストア・データをロード/ストア・ユニット 7 6 0 から C C U 1 0 6 へ転送し、C C U データ・ロード・オペレーションの実行によって即時にロード・バックすることができる。

#### 【 0 1 6 9 】

具体的には、物理アドレス全体が V M U 1 0 8 からロード/ストア・アドレス・バス 7 6 2 上に送出される。ロード・アドレスは一般的にはロード・アドレス・レジスタ 7 6 8

0-3 にストアされる。ストア・アドレスはストア・アドレス・レジスタ770<sub>3-0</sub>にラッチされる。ロード/ストア制御ユニット774は命令発行ユニット498から受信した制御信号を受けて動作し、ロード・アドレスとストア・アドレスをレジスタ768<sub>3-0</sub>、770<sub>3-0</sub>にラッチすることを調整する。ロード/ストア制御ユニット774は、ロード・アドレスをラッチするための制御信号を制御ライン778上に送出し、ストア・アドレスをラッチするための制御信号を制御ライン780上に送出する。ストア・データはストア・データ・レジスタ・セット782<sub>3-0</sub>の論理的に対応するスロットにストア・アドレスをラッチすると同時にラッチされる。4×4×32ビット幅アドレス比較ユニット772には、ロードおよびストア・アドレス・レジスタ768<sub>3-0</sub>、770<sub>3-0</sub>に入っているアドレスの各々が同時に入力される。システム・クロックの各半サイクル時の完全マトリックス・アドレス比較の実行は、制御ライン776を介してロード/ストア制御ユニット774によって制御される。ストア・アドレスに一致するロード・アドレスの存在と論理ロケーションは、制御ライン776を経由してロード/ストア制御ユニット774へ送られる。

10

## 【0170】

ロード・アドレスがVMU108から与えられ、保留中のストアがない場合は、ロード・アドレスは、CCUロード・オペレーションの開始と同時に、バス762から直接にアドレス・セレクタ786へバイパスされる。しかし、ストア・データが保留されている場合は、ロード・アドレスは使用可能なロード・アドレス・ラッチ768<sub>0-3</sub>にラッチされる。対応するストア・データ命令がリタイアされると制御信号をリタイア制御ユニット500から受信すると、ロード/ストア制御ユニット774はCCUデータ転送操作を開始し、制御ライン784を通してCCU106へのアクセスの仲裁を行う。CCU106がレディ(ready)を通知すると、ロード/ストア制御ユニット774はCCU物理アドレスをCCU PADDRアドレス・バス788上に送出するようにセレクタ786に指示する。このアドレスはアドレス・バス790を経由して対応するストア・レジスタ770<sub>3-0</sub>から得られる。対応するストア・データ・レジスタ782<sub>3-0</sub>からのデータはCCUデータ・バス792上に送出される。

20

## 【0171】

ロード命令が命令発行ユニット498から出されると、ロード/ストア制御ユニット774はロード・アドレス・ラッチ768<sub>3-0</sub>の一つが要求されたロード・アドレスをラッチすることを許可する。選択された特定のラッチ768<sub>0-3</sub>は関係命令の組の中のロード命令の位置に論理的に対応している。命令発行ユニット498は保留中の可能性のある2命令の組のどちらかの中のロード命令を示している5ビット・ベクトルをロード/ストア制御ユニット774へ渡す。コンパレータ772が一致するストア・アドレスを示していない場合は、ロード・アドレスはアドレス・バス794を経由してセレクタ786へ送られ、CCU PADDRアドレス・バス788上に出力される。アドレスの提供は、ロード/ストア制御ユニット774とCCU106間でやりとりされるCCU要求とレディ制御信号に従って行われる。実行ID値(ExID値)もロード/ストア制御ユニット774によって準備されてCCU106に対して出され、CCU106がExID値を含む要求データをそのあとで返却するときロード要求を識別する。このID値は4ビット・ベクトルからなり、現ロード要求を出したそれぞれのロード・アドレス・ラッチ768<sub>0-3</sub>を固有ビットで指定している。5番目のビットはロード命令を収めている命令の組を識別するために使用される。このID値は、したがって命令発行ユニット498からロード要求と一緒に送られるビット・ベクトルと同じである。

30

40

## 【0172】

先行する要求ロード・データが使用可能であることがCCU106からロード/ストア制御ユニット774へ通知されると、ロード/ストア制御ユニット774は位置合わせユニットがデータを受信し、それをロード・データ・バス764上に送出することを許可する。位置合わせユニット798はロード・データを右寄せする働きをする。

## 【0173】

50

CCU106からデータが返却されると同時に、ロード/ストア制御ユニット774は、CCU106からEXID値を受信する。他方、ロード/ストア制御ユニット774はロード・データがロード・データ・バス764上に送出されることを知らせる制御信号を命令発行ユニット498へ送り、さらに、どのロード命令に対してロード・データが返却されるのかを示したビット・ベクトルを返却する。

【0174】

C) IEU制御経路の詳細

再び、図5を参照して、IEU制御経路のオペレーションを図14に示したタイミング図と関連づけて説明する。図14に示した命令の実行タイミングは、本発明のオペレーションを例示したもので、種々態様に変更可能であることは勿論である。

10

【0175】

図14のタイミング図は、プロセッサ・システム・クロック・サイクルP<sub>0.6</sub>のシーケンスを示している。各プロセッサ・サイクルは内部TサイクルT<sub>0</sub>から始まる。本発明の好適実施例によるアーキテクチャ100では、各プロセッサ・サイクルは2つのTサイクルからなっている。

【0176】

プロセッサ・サイクル0のときは、IFU102とVMU108は物理アドレスを生成するように動作する。この物理アドレスはCCU106へ送られ、命令用キャッシュ・アクセス・オペレーションが開始される。要求された命令の組が命令用キャッシュ132にあると、命令の組はプロセッサ・サイクル1のほぼ中間でIFU102へ戻される。その後、IFU102はプリフェッチ・ユニット260とIFIFO264を経由する命令の組の転送を管理し、転送された命令の組はまずIEU104へ実行のために渡される。

20

【0177】

1) Eデコード・ユニットの詳細

Eデコード・ユニット490は全命令の組を並列に受け取って、プロセッサ・サイクル1が完了する前にデコードする。Eデコード・ユニット490は好適アーキテクチャ100では、バス124を経由して受け取ったすべての有効命令を並列に直接デコードする機能を備えた順列組合せ理論に基づくロジック・ブロックとして実現されている。アーキテクチャ100によって認識される命令は、各タイプ別に、命令、レジスタ要件および必要な資源の仕様とともに表VIに示されている。

30

【0178】

表VI

命令	命令 / 仕様 制御とオペランド情報*
レジスタ間移動	論理 / 算術演算機能コード： 加算、減算、乗算、シフトその他の指定 宛先レジスタ PSRのみセット ソース・レジスタ1 ソース・レジスタ2または即値定数値 レジスタセットA / B選択
即値からレジスタへ移動	宛先レジスタ 即値整数または浮動小数点定数値 レジスタ・セットA / B選択
ロード/ストア・レジスタ	オペレーション機能コード： ロードまたはストアの指定、即値、ベースと即値、またはベースとオフセットの使用 ソース / 宛先レジスタ ベース・レジスタ

40

50

即値コール 制御フロー	インデックス・レジスタまたは即値定数値 レジスタ・セット A / B 選択 符号付き即値変位 オペレーション機能コード： ブランチ・タイプとトリガ条件の指定 ベース・レジスタ インデックス・レジスタ、即値定数変位値、または はトラップ番号 レジスタ・セット A / B 選択 オペレーション機能コード：	
特殊レジスタ移動	特殊 / 整数レジスタとの間の移動の指定 特殊レジスタ・アドレス識別子ソース / 宛先レジ スタ レジスタ・セット A / B 選択 オペレーション機能コード：	10
整数変換移動	浮動小数点から整数への変換タイプの指定 ソース / 宛先レジスタ レジスタ・セット A / B 選択 オペレーション機能コード：	
ブール関数	ブール関数コード：AND、ORなどの指定 宛先ブール・レジスタ ソース・レジスタ 1 ソース・レジスタ 2 レジスタ・セット A / B 選択	20
拡張プロシージャ	プロシージャ指定子：プロシージャ・ベース値か らのアドレス・オフセットの指定 オペレーション：値をプロシージャ・ルーチンへ 渡す	
アトミック・プロシージャ	プロシージャ指定子：アドレス値の指定	

-----  
\* - 命令は、デコードされて命令を識別するフィールドのほかにこれらのフィールドを含  
んでいる。 30

【 0 1 7 9 】  
E デコード・ユニット 4 9 0 は命令の組の各命令を並列にデコードする。その結果の命  
令の識別、命令機能、レジスタ参照および機能要件はE デコード・ユニット 4 9 0 の出力  
から得られる。この情報は再生成され、命令の組の内のすべての命令がリタイアされるま  
でプロセッサ・サイクルの各半サイクル期間、E デコード・ユニット 4 9 0 によってラッ  
チされる。したがって、保留状態の 8 命令すべてに関する情報がE デコード・ユニット 4  
9 0 の出力から絶えず得られるようになっている。この情報は 8 エレメント・ビット・ベ  
クトルの形式で表示され、各ベクトルのビットまたはサブフィールドは 2 つの保留中命令  
の組の内の対応する命令の物理ロケーションに論理的に対応している。したがって、 8 個  
のベクトルが制御ライン 5 0 2 を経由してキャリア・チェッカ 4 9 2 へ送られる。この場  
合、各ベクトルは対応する命令がプロセッサ状況ワードのキャリア・ビットに作用を及ぼ  
しているか、あるいはそれに依存しているかを指定している。 8 個のベクトルが各命令の  
特定の内容と機能ユニット要件を示すために制御ライン 5 1 0 を経由して送られる。 8 個  
のベクトルが制御ライン 5 0 6 を経由して送られ、 8 個の保留中命令の各々によって使用  
されたレジスタ参照を指定している。これらのベクトルはプロセッサ・サイクル 1 が終了  
する前に送られる。 40

【 0 1 8 0 】  
2 ) キャリー・チェッカ・ユニットの詳細  
キャリア・チェッカ・ユニット 4 9 2 は図 1 4 に示すオペレーションのデータ依存関係 50

フェーズ期間の間に依存関係検査ユニット494と並列に動作する。キャリア・チェッカ・ユニット492は好適アーキテクチャ100では順列組合せ理論に基づくロジックとして実現されている。したがって、キャリア・チェッカ・ユニット492によるオペレーションの各繰り返し時に、命令がプロセッサ状態レジスタのキャリア・フラグを変更したかどうかについて8個の命令すべてが考慮される。これが必要とされるのは、その前の命令によって設定されたキャリア・ビットの状況に依存する命令を順序外で実行することを可能にするためである。制御ライン504上に送出された制御信号により、キャリア・チェッカ・ユニット492は、キャリア・フラグに対する先行命令の実行に依存する特定の命令を識別することができる。

#### 【0181】

さらに、キャリア・チェッカ・ユニット492は8個の保留中命令の各々についてキャリア・ビットの一時的コピーをもっている。キャリア・ビットを変更していない命令については、キャリア・チェッカ・ユニット492はプログラム命令ストリームの順序でキャリア・ビットを次の命令に伝える。したがって、順序外で実行され、キャリア・ビットを変更する命令を実行させることが可能であり、さらに、その順序外で実行される命令に依存する後続の命令も、キャリア・ビットを変更する命令のあとに置かれていても、実行することが可能である。さらに、キャリア・ビットがキャリア・チェッカ・ユニット492によって維持されているので、これらの命令のリタイア以前に例外が起こったとき、キャリア・チェッカ・ユニットは内部一時キャリア・ビット・レジスタをクリアするだけでよいことから、順序外で実行することが容易になる。その結果、プロセッサ状況レジスタは、順序外で実行される命令の実行による影響を受けない。キャリア・チェッカ・ユニット492が維持している一時キャリア・ビット・レジスタは、順序外で実行される各命令が完了すると更新される。順序外で実行される命令がリタイアされると、プログラム命令ストリームの中で最後にリタイアされた命令に対応するキャリア・ビットはプロセッサ状況レジスタのキャリア・ビット・ロケーションへ転送される。

#### 【0182】

##### 3) データ依存関係チェッカ・ユニットの詳細

データ依存関係チェッカ・ユニット494はEデコード・ユニット490から制御ライン506を經由して8個のレジスタ参照識別ベクトルを受け取る。各レジスタの参照は32個のレジスタを一度に1つを識別するのに適した5ビット値と“ A ” “ B ”またはブール・レジスタ・セット内に置かれているレジスタ・バンクを識別する2ビット値によって示されている。浮動小数点レジスタ・セットは“ B ”レジスタ・セットとも呼ばれる。各命令は最高3つまでのレジスタ参照フィールドを持つことができる。2つのソース・レジスタ・フィールドと1つの宛先レジスタ・フィールドである。ある種の命令、特にレジスタ間移動命令は、宛先レジスタを指定している場合があっても、Eデコード・ユニット490によって認識される命令ビット・フィールドは実際に作製される出力データがないことを意味している場合がある。むしろ、命令の実行は、プロセッサ状況レジスタの値の変更を判断することだけを目的としている。

#### 【0183】

データ依存関係チェッカ494も好適アーキテクチャ100において純然たる組合せロジック ( pure combinatorial logic ) で実現されているが、これはプログラム命令ストリーム内に後に現れる命令のソース・レジスタ参照と相対的に前に置かれた命令の宛先レジスタ参照との間の依存関係を同時に判断するように動作する。ビット配列は、どの命令が他の命令に依存するかを識別するだけでなく、各依存関係がどのレジスタに基づいて生じたかを識別するデータ依存関係チェッカ494によって作られる。

キャリアとレジスタ・データの依存関係は、第2プロセッサ・サイクルの開始直後に判別される。

#### 【0184】

##### 4) レジスタ改名ユニットの詳細

レジスタ改名ユニット496は8個の保留中の命令すべてのレジスタ参照のIDを制御ライン506を經由してレジスタ依存関係を制御ライン508を經由して受け取る。8個のエLEMENTからのマトリックスも制御ライン542を經由して受け取る。これらのELEMENTは保留中命令の現セットの中でどの命令が実行されたか(完了したか)を示している。この情報からレジスタ改名ユニット496は制御信号の8ELEMENT配列を制御ライン512を經由して命令発行ユニット498へ送る。このようにして送られた制御情報は、現セットのデータ依存関係が判別された場合に、まだ実行されていない現在保留中の命令のうちどの命令の実行が可能になったかについてレジスタ改名ユニット496が行った判断を反映している。レジスタ改名ユニット496は実行のために同時に出される最高6個までの命令を識別した選択制御信号をライン516を經由して受信する。つまり、2個の整数命令、2個の浮動小数点命令および2個のブール命令である。

10

**【0185】**

レジスタ改名ユニット496はバス518を經由してレジスタ・ファイル配列472へ送られた制御信号を通して、識別された命令を実行する際にアクセスするソース・レジスタを選択するというもう一つの機能を備えている。順序外で実行される命令の宛先レジスタは、対応するデータ経路の一時バッファ612、680、728に置かれているものとして選択される。順序内で実行される命令は完了するとリタイアされ、その結果データはレジスタ・ファイル614、684、732にストアされていく。ソース・レジスタの選択は、レジスタが以前に宛先として選択され、対応する以前の命令がまだリタイアされていないかどうかによって決まる。そのような場合には、ソース・レジスタは対応する一時バッファ612、680、728から選択される。以前の命令がリタイアされていた場合は、対応するレジスタ・ファイル614、684、732のレジスタが選択される。その結果、レジスタ改名ユニット496は順序外で実行される命令の場合には、レジスタ・ファイル・レジスタの参照を一時バッファ・レジスタの参照に実効的に置き換えるように動作する。

20

**【0186】**

アーキテクチャ100によれば、一時バッファ612、680、728は対応するレジスタ・ファイル配列のレジスタ構造と重複していない。むしろ、8個の保留命令の各々に対して1つの宛先レジスタ・スロットが用意されている。その結果、一時バッファ宛先レジスタ参照の置換は、保留レジスタ・セット内の対応する命令のロケーションによって判断される。そのあとのソース・レジスタ参照はソース依存関係が発生した命令に対してデータ依存関係チェッカ494によって識別される。したがって、一時バッファ・レジスタ内の宛先スロットはレジスタ改名ユニット496によって容易に判断することが可能である。

30

**【0187】****5) 命令発行ユニットの詳細**

命令発行ユニット498は、発行できる命令のセットをレジスタ改名ユニット496の出力とEデコード・ユニット490によって識別された命令の機能要件に基づいて判断する。命令発行ユニット498は制御ライン514を經由して報告された機能ユニット478<sub>0..n</sub>の各々の状況に基づいてこの判断を行う。したがって、命令発行ユニット498は発行すべき使用可能な命令の組をレジスタ改名ユニット496から受信すると、オペレーションを開始する。各命令を実行するためにレジスタ・ファイルへのアクセスが必要であるとすると、命令発行ユニット498は現在命令を実行中の機能ユニット498<sub>0..n</sub>が使用可能であることを予想する。レジスタ改名ユニット496へ発行すべき命令を判別する際の遅延を最小にするために、命令発行ユニット498は専用の組合せロジックで実現されている。

40

**【0188】**

発行すべき命令を判別すると、レジスタ改名ユニット496はレジスタ・ファイルへのアクセスを開始し、このアクセスは第3プロセッサ・サイクルP2が終了するまで続けられる。プロセッサ・サイクルP3が開始すると、命令発行ユニット498は「Execu

50

te 0」で示すように1つまたは2つ以上の機能ユニット478<sub>0..n</sub>によるオペレーションを開始し、レジスタ・ファイル配列472から送られてきたソース・データを受信して処理する。

#### 【0189】

代表例として、アーキテクチャ100で処理される大部分の命令は1プロセッサ・サイクルで機能ユニットを通して実行される。しかし、一部の命令は、「Execute 1」で示すように、同時に出力された命令を完了するのに複数のプロセッサ・サイクルを必要とする。Execute 0命令とExecute 1命令は、例えばそれぞれALUと浮動小数点乗算機能ユニットに実行させることができる。ALU機能ユニットは図14に示すように、1プロセッサ・サイクル内で出力データを発生し、この出力データはラッチしておき、第5プロセッサ・サイクルP4時に別の命令を実行する際に使用することができる。浮動小数点乗算機能ユニットは内部パイプライン化機能ユニットにすることが好ましい。したがって、次のプロセッサ・サイクルで別の浮動小数点命令を出すことができる。しかし、最初の命令の結果はデータに依存するプロセッサ・サイクル数の間使用することができない。図14に示す命令は、機能ユニットでの処理を完了するためには、3プロセッサ・サイクルを必要とする。

10

#### 【0190】

各プロセッサ・サイクルの間に、命令発行ユニット498の機能は繰り返される。その結果、現在の保留中の命令の組の状況と機能ユニット478<sub>0..n</sub>の全セットの使用可能状況は各プロセッサ・サイクルの間に再評価される。したがって、最適条件のとき、最適アーキテクチャ100はプロセッサ・サイクルごとに最高6個までの命令を実行することができる。しかし、代表的な命令ミックスから得られる総平均実行命令数は1プロセッサ・サイクル当たり1.5個ないし2.0個である。

20

#### 【0191】

命令発行ユニット498の機能で最後に考慮すべきことは、このユニットがトラップ条件の処理と特定命令の実行に関与することである。トラップ条件を発生するためには、まだリタイアされていないすべての命令をIEU104からクリアする必要がある。このような事態は、算術演算エラーに回答して、機能ユニット478<sub>0..n</sub>のいずれからか、あるいは例えば違法命令をデコードしたときにEデコード・ユニット490から外部割り込みを受信し、それが割り込み要求/受信確認制御ライン340を経由して、IEU104へ中継されたのに回答して、起こることがある。トラップ条件が発生しとき、命令発行ユニット498は現在IEU104で保留されているすべての非リタイア命令を中止または無効にすることを受け持つ。同時にリタイアできない命令はすべて無効にされる。この結果は、プログラム命令ストリームを順序内で実行する従来方式に対して割り込みを正確に発生させるために不可欠である。IEU104がトラップ処理プログラム・ルーチンの実行を開始する準備状態になると、命令発行ユニット498は制御ライン340を経由するリターン制御信号によって割り込みの受信を確認する。また、従来純然たる順序内ルーチンにおいてある命令が実行される前に変更されたプロセッサ状態ビットに基づいてその命令に対する例外条件が認識される可能性を防止するために命令発行ユニット498は、PSRを変更する可能性のあるすべての命令(特殊移動やトラップからのリターンなど)が厳格に順序内で実行されるようにすることを受け持つ。

30

40

#### 【0192】

プログラム制御の流れを変更するある種の命令は、Eデコード・ユニット262によって判別されない。この種の命令には、サブルーチン・リターン、プロシージャ命令からのリターン、トラップからのリターンがある。命令発行ユニット498は判別制御信号をIEUリターン制御ライン350を経由してIFU102へ送る。特殊レジスタ412のうち対応するものが選択されて、コール命令の実行時、トラップの発生時またはプロシージャ命令の出現時に存在していたIF

PC実行アドレスを出力する。

#### 【0193】

50

## 6) 完了制御ユニットの詳細

完了制御ユニット540は機能ユニット478<sub>0-n</sub>をモニタして、現在のオペレーションの完了状況を調べる。好適アーキテクチャ100では、完了制御ユニット540は各機能ユニットによるオペレーションの完了を予想して、現在保留中の命令の組の中の各命令の実行状況を示した完了ベクトルを機能ユニット478<sub>0-n</sub>による命令の実行完了よりも約半プロセッサ・サイクル前にレジスタ改名ユニット496、バイパス制御ユニット520およびリタイア制御ユニット500へ送る。これにより、命令発行ユニット498はレジスタ改名ユニット496を通して、実行を完了する機能ユニットを次の命令発行サイクルに対して使用可能な資源として考慮することができる。バイパス制御ユニット520は、機能ユニットから出力されたデータをバイパス・ユニット474を通るようにバイパスする準備を行うことができる。最後に、リタイア制御ユニット500は機能ユニット478<sub>0-n</sub>からレジスタ・ファイル配列472へデータを転送するのと同時間に対応する命令をリタイアするように動作する。

10

【0194】

## 7) リタイア制御ユニットの詳細

完了制御ユニット540から送られた命令完了ベクトルの他に、リタイア制御ユニット500はEデコード・ユニット490から出力された最も古い命令の組をモニタする。命令ストリーム順序の中の各命令に完了制御ユニット540によって完了の印(マーク)が付けられると、リタイア制御ユニット500は、制御ライン534上に送出された制御信号を通して一時バッファ・スロットからレジスタ・ファイル配列472内の対応する命令が指定したファイル・レジスタ・ロケーションヘデータを転送することを指示する。1つまたは複数の命令が同時にリタイアされると、PC Inc/size制御信号が制御ライン344上に送出される。各プロセッサ・サイクルごとに最高4個までの命令をリタイアすることが可能である。命令の組の全体がリタイアされると、IFIFO読取り制御信号が制御ライン342上に送出されてIFIFO264を前進させる。

20

【0195】

## 8) 制御フロー制御ユニットの詳細

制御フロー制御ユニット528は、現在の保留中命令の組の中の制御フロー命令が解決されたかどうか、さらに、その結果ブランチが行われたかどうかを指定した情報をIFU102に絶えず与えるように動作する。制御フロー制御ユニット528はEデコード・ユニット490による制御フロー・ブランチ命令の識別情報を制御ライン510を經由して取得する。現在のレジスタ依存関係のセットは、制御ライン536を經由してデータ依存関係チェッカ・ユニット494から制御フロー制御ユニット528へ送られるので、制御フロー制御ユニット528はブランチ命令の結果が依存関係に拘束されているかどうか、あるいは判明しているかどうかを判断することができる。レジスタ改名ユニット496からバス518を經由して、送られたレジスタの参照は制御フロー制御ユニット528によってモニタされ、ブランチ決定を定義するプール・レジスタが判別される。したがって、ブランチ決定は制御フロー命令の順序外の実行以前でも判断することが可能である。

30

【0196】

制御フロー命令の実行と同時に、バイパス・ユニット474は、制御フロー1と制御フロー2の制御ライン750、752からなる制御ライン530を經由して制御フローの結果を制御フロー制御ユニット528へ送るように指示される。最後に、制御フロー制御ユニット528は各々が8ビットの2個のベクトルを制御ライン348を經由して、IFU102へ連続して送る。これらのベクトルは、ベクトル内のビットに対応する論理ロケーションに置かれた命令が解決されたか否かおよびその結果ブランチが行われたか否かを定義している。

40

好適アーキテクチャ100では、制御フロー制御ユニット528は制御ユニット528への入力制御信号を受けて連続的に動作する組合せロジックとして実現されている。

【0197】

## 9) バイパス制御ユニットの詳細

50

命令発行ユニット498は、バイパス制御ユニット520と厳密に協働して、レジスタ・ファイル配列472と機能ユニット478<sub>0..n</sub>間のデータのルーチング(経路指定)を制御する。バイパス制御ユニット520は図14に示すオペレーションのレジスタ・ファイル・アクセス、出力およびストア・フェーズと関連して動作する。レジスタ・ファイル・アクセスの間にはバイパス制御ユニット520は命令の実行の出力フェーズの間に書き込まれている途中にあるレジスタ・ファイル配列472内の宛先レジスタのアクセスを制御ライン522を通して認識することができる。この場合、バイパス制御ユニット520はバイパスして機能ユニット配布バス480に返すように、機能ユニット出力バス482上に送出されたデータを選択することを指示する。バイパス・ユニット520に対する制御は、制御ライン542を通して命令発行ユニット498によって行われる。

10

## 【0198】

## IV. 仮想メモリ制御ユニット

VMU108のインタフェース定義は図15に示されている。VMU108は主にVMU制御ロジック・ユニット800と内容アドレス(content addressable)メモリ(CAM)802から構成されている。VMU108の一般的機能は図16にブロック図で示してある。

同図において、仮想アドレスの表現はスペースID(sID[31:28])、仮想ページ番号(VADDR[27:14])、ページ・オフセット(PADDR[13:4])、および要求ID(rID[3:0])に分割されている。物理アドレスを生成するためのアルゴリズムでは、スペースIDを使用して、スペース・テーブル842内の16個のレジスタから1つを選択するようになっている。選択したスペース・レジスタの内容と仮想ページ番号とを組み合わせて、テーブル索引バッファ(TLB)844をアクセスするときのアドレスとして使用される。34ビット・アドレスは内容アドレス・タグの働きをし、バッファ844内の対応するバッファ・レジスタを指定するために使用される。タグに一致するものが見つかり、18ビット幅レジスタ値が物理アドレス846の上位18ビットとして得られる。ページ・オフセットと要求IDは物理アドレス846の下位14ビットとして得られる。

20

## 【0199】

タグに一致するものがテーブル索引バッファ844に見つからないと、VMU不一致が通知される。この場合は、MAU112に維持されている完全ページ・テーブル・データ構造をアクセスする従来のハッシュ・アルゴリズム848を採用したVMU高速トラップ処理ルーチンを実行させる必要がある。このページ・テーブル850はアーキテクチャ100によって現在使用中のすべてのメモリ・ページのエントリを含んでいる。ハッシュ・アルゴリズム848は現在の仮想ページ変換操作を満たすために必要なページ・テーブル・エントリを判別する。これらのページ・テーブル・エントリはMAU112からレジスタ・セット“A”のトラップ・レジスタへロードされ、そのあと特殊レジスタ移動命令によってテーブル索引バッファ844へ転送される。例外処理ルーチンから戻ると、VMU不一致例外を引き起こした命令はIEU104によって再実行される。仮想アドレスから物理アドレスへの変換操作は例外を引き起こさないで完了するはずである。

30

## 【0200】

VMU制御ロジック800はIFU102およびIEU104とのデュアル・インタフェースとなる。準備信号は制御ライン822を経由して、IEU104へ送られ、VMU108がアドレス変換のために使用可能であることを通知する。好適実施例では、VMU108は常にIFU102の変換要求を受け付ける準備状態にある。IFU102およびIEU104は共に、制御ライン328および804を経由して要求を提示することができる。好適アーキテクチャ100では、IFUは優先してVMU108をアクセスすることができる。その結果、ビジー(使用中)制御ライン820は1つだけがIEU104に出力される。

40

## 【0201】

IFU102およびIEU104は共にスペースIDと仮想ページ番号フィールドをそ

50

れぞれ制御ライン 326 および 808 を経由して、VMU 制御ロジック 800 へ送る。さらに IEU 104 は読み書き制御信号を制御信号 806 で出力する。この制御信号は、参照された仮想メモリのメモリ・アクセス保護属性を変更するために、そのアドレスをロード・オペレーションに使用すべきか、ストア・オペレーションに使用すべきかを必要に応じて定義している。仮想アドレスのスペース ID と仮想ページ・フィールドは、CAM ユニット 802 に渡されて、実際の変換操作が行われる。ページ・オフセットと E x ID フィールドは最終的に IEU 104 から直接に CCU 106 へ送られる。物理ページと要求 ID フィールドは、アドレス・ライン 836 を経由して CAM ユニット 802 へ送られる。テーブル索引バッファに一致するものが見つかったら、ヒット・ラインと制御出力ライン 830 を経由して VMU 制御ロジック・ユニット 800 に通知される。その結果の 18 ビット長の物理アドレスはアドレス出力ライン 824 上に出力される。

10

#### 【0202】

VMU 制御ロジック・ユニット 800 は、ライン 830 からヒットおよび制御出力制御信号を受けると、仮想メモリ不一致と仮想メモリ例外制御信号をライン 334、332 上に出力する。仮想メモリ変換不一致とは、テーブル索引バッファ 844 内のページ・テーブル ID と一致しなかったことを意味する。その他の変換エラーはすべて仮想メモリ例外として報告される。

#### 【0203】

最後に CAM ユニット 802 内のデータ・テーブルは特殊レジスタ間移動命令を IEU 104 が実行することによって変更することができる。読み書き、レジスタ選択、リセット、ロードおよびクリア制御信号は IEU 104 から制御ライン 810、812、814、816、818 を経由して出力される。CAM ユニット・レジスタに書くべきデータは、特殊アドレス・データ・バス 354 に接続されたアドレス・バス 808 を経由して IEU 104 から VMU 制御ロジック・ユニット 800 によって受信される。このデータは初期設定、レジスタ選択、および読み書き制御信号を制御する制御信号と同時にバス 836 を経由して CAM ユニット 802 へ転送される。その結果、CAM ユニット 802 内のデータ・レジスタは、より高レベルのオペレーティング・システムで定義されているコンテキスト・スイッチを処理するとき必要になるストアのための読出しを含めて、アーキテクチャ 100 の動的オペレーションの間に必要に応じて即座に書き出すことができる。

20

#### 【0204】

##### V. キャッシュ制御ユニット

CCU 106 のデータ・インタフェースに対する制御は、図 17 に示されている。この場合も、IFU 102 と IEU 104 用に別々のインタフェースが設けられている。さらに論理的に別個のインタフェースが CCU 106 に用意され、命令とデータ転送のために MCU 110 と結ばれている。

IFU インタフェースはアドレス・ライン 324 上に送られる物理ページ・アドレス、アドレス・ライン 824 上に送られる VMU 変換ページ・アドレス、および制御ライン 294、296 上を別々に転送される要求 ID からなっている。単方向データ転送バス 114 は命令の組の全体を IFU 102 と並列に転送するためのものである。最後に、読取り/使用中および準備制御信号は制御ライン 298、300、302 を経由して、CCU 106 へ送られる。

30

40

#### 【0205】

同様に、物理アドレス全体は物理アドレス・バス 788 を経由して、IEU 104 へ送られる。要求 E x ID は制御ライン 796 を経由して、IEU 104 のロード/ストア・ユニットとの間で別々に受渡される。80 ビット幅単方向データ・バスは CCU 106 から IEU 104 に出力される。

しかし、アーキテクチャ 100 の好適実施例では、下位の 64 ビットだけが IEU 104 によって使用される。全 80 ビット・データ転送バスを CCU 106 内で使用できるようにし、かつ、サポートしているのは、本アーキテクチャ 100 の引き続いての実行をサポートするためであり、浮動少数点データ経路 660 を変更することによって、IEEE 標

50

準 7 5 4 に準拠する浮動小数点のオペレーションをサポートする。

【 0 2 0 6 】

I E U 制御インターフェースは、要求、使用中、準備、読み書きを通して、および制御信号 7 8 4 を通して確立され、実質的には I F U 1 0 2 によって使用される対応する制御信号と同じである。例外は、ロード・オペレーションとストア・オペレーションを区別するための読み書き制御信号が設けられていることである。幅制御信号は I E U 1 0 4 による各 C C U 1 0 6 へのアクセス時に転送されるバイト数を指定している。これに対して、命令用キャッシュ 1 3 2 の全てのアクセスは固定した 1 2 8 ビット幅データ・フェッチ・オペレーションになっている。

【 0 2 0 7 】

C C U 1 0 6 は命令用キャッシュ 1 3 2 とデータ用キャッシュ 1 3 4 に対して従来とほぼ同じキャッシュ制御機能を備えている。好適アーキテクチャ 1 0 0 では、命令用キャッシュ 1 3 2 は 2 5 6 個の 1 2 8 ビット幅命令の組をストアする機能を備えた高速メモリになっている。データ用キャッシュ 1 3 4 は 1 0 2 4 個の 3 2 ビット幅ワードのデータをストアする機能を備えている。命令用キャッシュ 1 3 2 とデータ用キャッシュ 1 3 4 の内容から即時に満足できない命令要求とデータ要求は M C U 1 1 0 に渡される。命令用キャッシュがミスした場合は、2 8 ビット幅物理アドレスがアドレス・バス 8 6 0 を経由して M C U 1 1 0 に渡される。要求 I D および C C U 1 0 6 と M C U 1 1 0 のオペレーションを調整するための追加制御信号は制御ライン 8 6 2 上に送出される。M C U 1 1 0 が M A U 1 1 2 の必要な読取りアクセスを調整すると、2 つの連続する 6 4 ビット幅データ転送が直接に M A U 1 1 2 から命令用キャッシュ 1 3 2 へ行われる。2 つの転送が必要になるのは、データ・バス 1 3 6 が好適アーキテクチャ 1 0 0 では、6 4 ビット幅バスになっているためである。要求したデータが M C U 1 1 0 を通して返却されると、要求オペレーションの保留中に維持されていた要求 I D も制御ライン 8 6 2 を経由して C C U 1 0 6 へ返却される。

【 0 2 0 8 】

データ用キャッシュ 1 3 4 と M C U 1 1 0 との間のデータ転送オペレーションは、命令用キャッシュの転送オペレーションとほぼ同じである。データ・ロードとストア・オペレーションは単一バイトを参照できるので、全 3 2 ビット幅の物理アドレスがアドレス・バス 8 6 4 を経由して、M C U 1 1 0 へ送られる。インタフェース制御信号と要求 E x I D は制御ライン 8 6 6 を経由して、転送される。双方向の 6 4 ビット幅データ転送はデータ用キャッシュ・バス 1 3 8 を経由して行われる。

【 0 2 0 9 】

発明の効果：

高性能 R I S C をベースとしたマイクロプロセッサ・アーキテクチャは以上に説明した通りである。本発明のアーキテクチャによれば、命令を順序外に実行することができ、メインとターゲット命令ストリームのプリフェッチ命令転送経路を別々に設け、およびプロシージャ命令認識と専用プリフェッチ経路を設けることができる。命令実行ユニットは最適化されているので、最適化された複数のデータ処理経路で整数、浮動小数点およびブール演算をサポートすることができ、また、それぞれの一時レジスタ・ファイルが設けられているので、容易に設定されるマシン状態の状況を正確に維持しながら、順序外の実行と命令取消しを容易に行うことができる。

したがって、上述した説明では、本発明の好適実施例を開示しているが、当業者にとって本発明の範囲内で種々変更および改良することが可能であることは勿論である。

【 図面の簡単な説明 】

【 0 2 1 0 】

【 図 1 】 本発明を実現するマイクロプロセッサ・アーキテクチャを示す簡略図である。

【 図 2 】 本発明の命令フェッチ・ユニットを示す詳細ブロック図である。

【 図 3 】 本発明のプログラム・カウンタ・ロジック・ユニットを示すブロック図である。

【 図 4 】 プログラム・カウンタ・データと制御経路ロジックを示す別の詳細ブロック図で

10

20

30

40

50

ある。

【図5】本発明の命令実行ユニットを示す簡略ブロック図である。

【図6】レジスタ・ファイル・アーキテクチャ及び1次及び2次命令の組を図形で示す図である。

【図7】1次整数レジスタの再構成ステージを図形で示す図である。

【図8】本発明の再構成可能浮動小数点及び2次整数レジスタ・セットを図形で示す図である。

【図9】本発明の3次ブール・レジスタ・セットを図形で示す図である。

【図10】本発明の命令実行ユニットの1次整数処理データ経路部分を示す詳細ブロック図である。

10

【図11】本発明の命令実行ユニットの1次浮動小数点データ経路部分を示す詳細ブロック図である。

【図12】本発明の命令実行ユニットのブール演算データ経路部分を示す詳細ブロック図である。

【図13】本発明のロード/ストア・ユニットを示すブロック図である。

【図14】本発明のオペレーション順序を示すタイミング図である。

【図15】本発明の仮想メモリ制御ユニットを示す簡略ブロック図である。

【図16】本発明の仮想メモリ制御アルゴリズムを示すブロック図である。

【図17】本発明のキャッシュ制御ユニットを示す簡略ブロック図である。

【符号の説明】

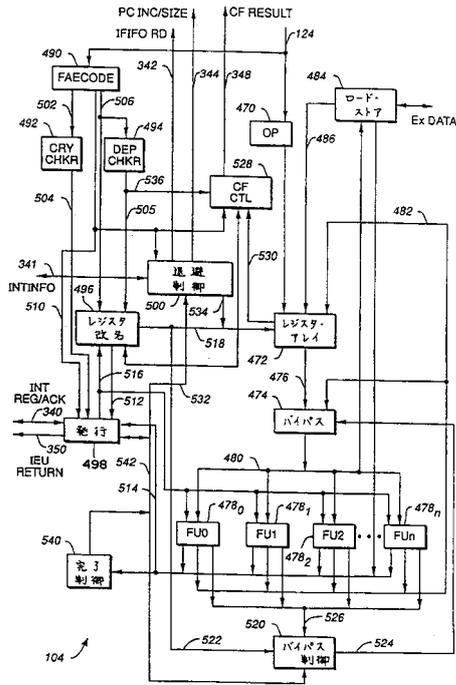
20

【0211】

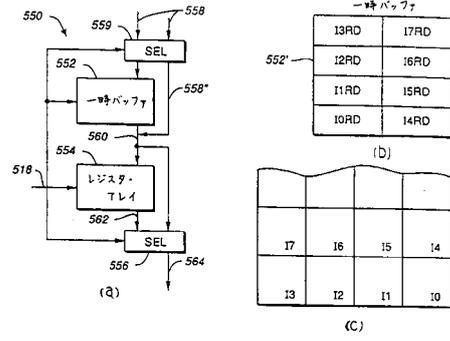
- 100 ... アーキテクチャの概要
- 102 ... 命令フェッチ・ユニット (IFU)
- 104 ... 命令実行ユニット (IEU)
- 106 ... キャッシュ制御ユニット (CUU)
- 108 ... 仮想メモリ・ユニット (VMU)
- 110 ... メモリ制御ユニット (MCU)
- 112 ... メモリ・アレイ・ユニット (MAU)



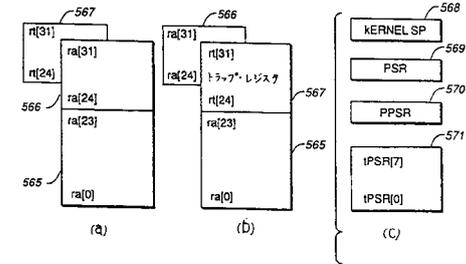
【図5】



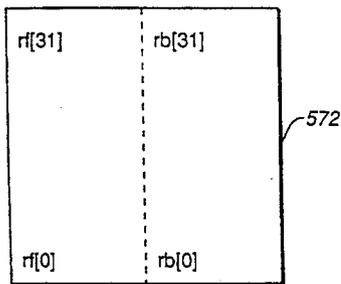
【図6】



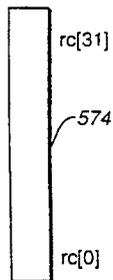
【図7】



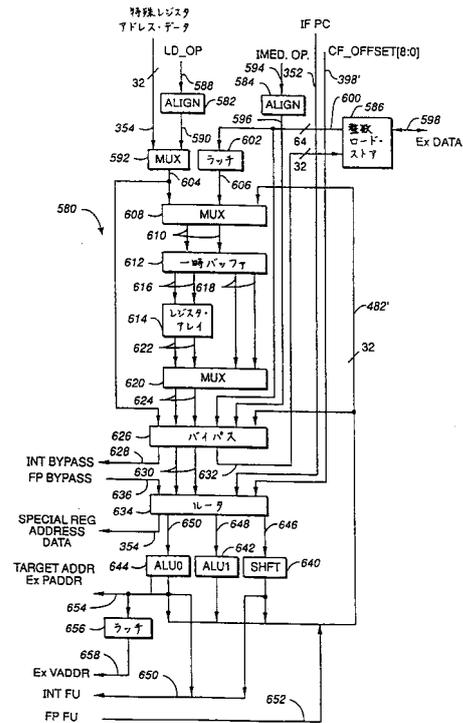
【図8】



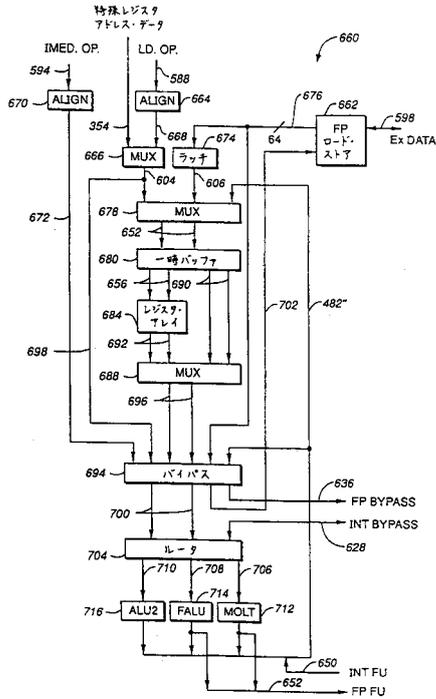
【図9】



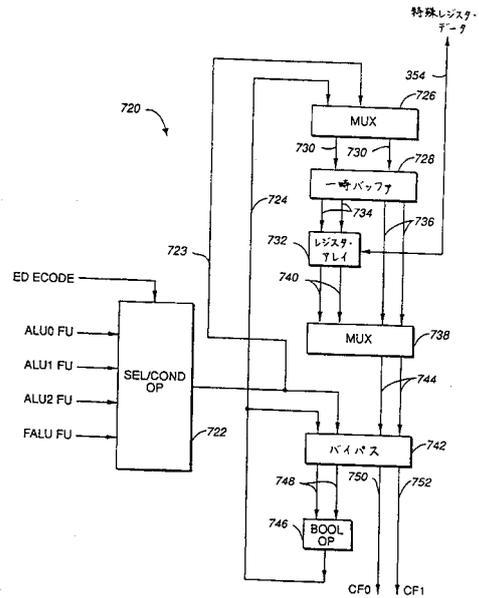
【図10】



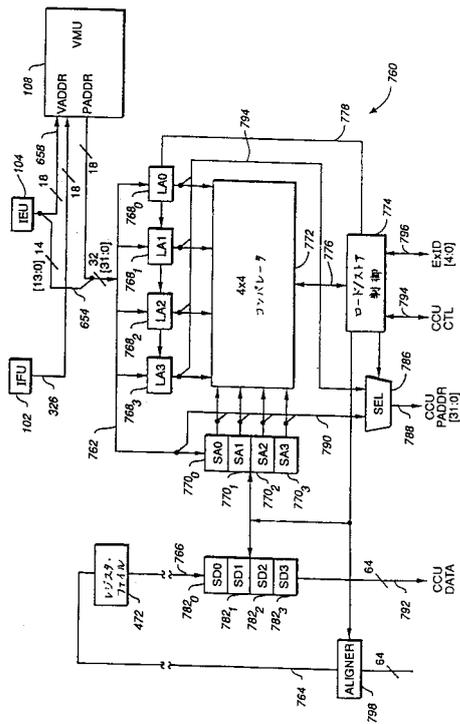
【図11】



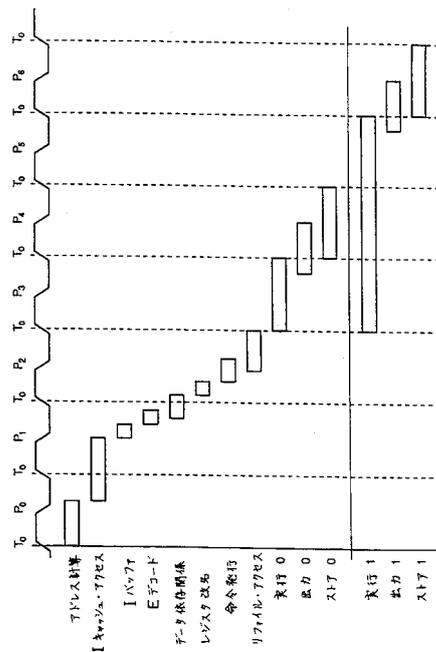
【図12】



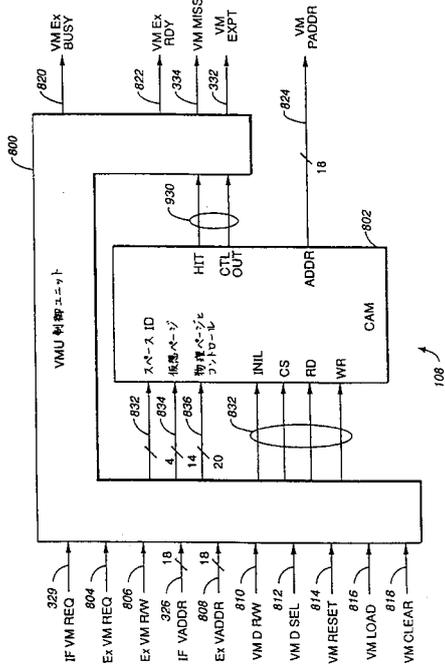
【図13】



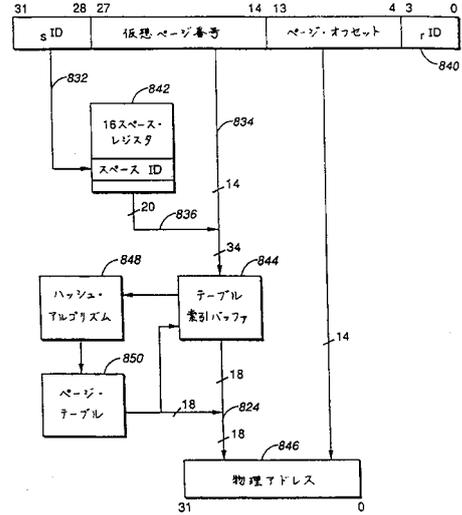
【図14】



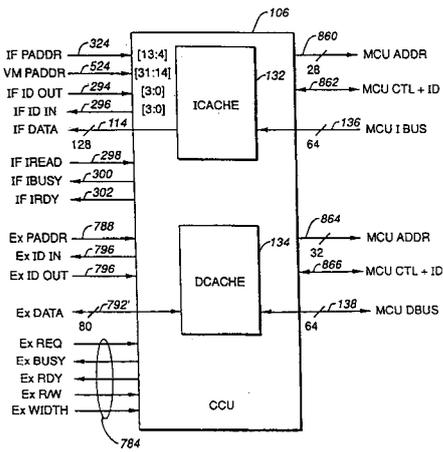
【 図 15 】



【 図 16 】



【 図 17 】



## フロントページの続き

- (72)発明者 ミヤママ, ヨシユキ  
アメリカ合衆国 95050 カリフォルニア州 サンタ クララ ランチョ マコーミック ブ  
ールバード 2171
- (72)発明者 ガルグ, サンジブ  
アメリカ合衆国 94539 カリフォルニア州 フリーモント センティネル ドライブ 46  
820
- (72)発明者 ハギワラ, ヤスアキ  
アメリカ合衆国 95050 カリフォルニア州 サンタ クララ モンロー ストリート 22  
50 アパート 274
- (72)発明者 ワン, ジョハネス  
アメリカ合衆国 94062 カリフォルニア州 レッドウッド シティ キング ストリート  
25
- (72)発明者 ラウ, テーリ  
アメリカ合衆国 94306 カリフォルニア州 パロアルト カレッジ アヴェニュー 411  
アパート イー
- (72)発明者 トラン, クワン エイチ.  
アメリカ合衆国 95130 カリフォルニア州 サンノゼ メイフィールド アヴェニュー 2  
045

審査官 石川 正二

- (56)参考文献 特公昭50-021821(JP, B1)  
特開昭63-280355(JP, A)  
中島雅逸 外8名, "OHMEGA: 数値計算用スーパー scaler・マイクロプロセッサのアーキ  
テクチャ-ハードウェア構成とパイプライン構造-", 情報処理学会研究報告 91-ARC-  
87, 1991年 3月11日, 第91巻, 第23号, p. 1-7

- (58)調査した分野(Int.Cl., DB名)  
G06F 9/38