

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3724497号

(P3724497)

(45) 発行日 平成17年12月7日(2005. 12. 7)

(24) 登録日 平成17年9月30日(2005. 9. 30)

(51) Int. Cl.⁷

G06F 9/38

F I

G06F 9/38 310F

請求項の数 2 (全 61 頁)

(21) 出願番号	特願2004-265342 (P2004-265342)	(73) 特許権者	000002369
(22) 出願日	平成16年9月13日(2004. 9. 13)		セイコーエプソン株式会社
(62) 分割の表示	特願2002-267999 (P2002-267999)		東京都新宿区西新宿2丁目4番1号
	の分割	(74) 代理人	110000028
原出願日	平成4年7月7日(1992. 7. 7)		特許業務法人明成国際特許事務所
(65) 公開番号	特開2005-50369 (P2005-50369A)	(72) 発明者	グエン, レ トロン
(43) 公開日	平成17年2月24日(2005. 2. 24)		アメリカ合衆国 95030 カリフォル
審査請求日	平成16年10月13日(2004. 10. 13)		ニア州 モンテ セレノ ダニエル プレ
(31) 優先権主張番号	727, 058		イス 15096
(32) 優先日	平成3年7月8日(1991. 7. 8)	(72) 発明者	レンツ, デレク ジェイ.
(33) 優先権主張国	米国 (US)		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

最終頁に続く

(54) 【発明の名称】 スーパースカラープロセッサ及びデータ処理装置

(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プロセッサであって、

複数の命令を、前記機能ユニットでの実行に先立って、予め定めたプログラム順序でメモリからプリフェッチするプリフェッチユニットと、

該プリフェッチユニットの後段に設けられ、複数の命令を保持する命令バッファと、

前記命令バッファの後段に設けられ、前記命令バッファに保持された命令のうちの複数の命令を同時にデコードするデコーダと、

該デコードされた命令を前記機能ユニットにより実行する際に用いられる複数のレジスタと、複数の命令を順序外で実行した結果を蓄える実行結果バッファとを備えたレジスタファイルと、

前記デコーダから出力された前記複数の命令の依存関係を、少なくとも前記レジスタの使用条件に基づいて検出する依存関係チェックユニットと、

前記依存関係チェックユニットにより依存関係に基づく実行上の制約がないと判断された命令については、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命令発行ユニットと、

前記複数の命令を前記順序外での実行が完了した際に前記実行結果バッファに実行結果を格納し、前記順序外で実行された命令をリタイアする際に前記実行結果バッファ内の実行結果を前記対応するレジスタに書き込むリタイアユニットと、

10

20

を備え、

前記デコードされた命令であってリタイアされていない命令のロケーションは、該命令によって用いられる前記実行結果バッファのロケーションに対応づけられており、

前記命令がリタイアするとき、該リタイアされる命令のロケーションに対応づけられていた前記実行結果バッファのロケーションが開放される、スーパースカラープロセッサ。

【請求項 2】

請求項 1 記載のスーパースカラープロセッサと、

外部メモリと、

前記スーパースカラープロセッサと前記外部メモリとを接続するバスと、
を備えるデータ処理装置。

10

【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、特に、特定の計算機能に合わせてチューン（調整）したものを含めて、機能的な計算エレメントをアーキテクチャに追加することによって計算スループットを向上するために容易に拡張することのできる RISC マイクロプロセッサ・アーキテクチャに関する。

【0002】

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。

20

(1) 発明の名称「高性能 RISC マイクロプロセッサ・アーキテクチャ」(High-Performance RISC Microprocessor Architecture) SMOS 7984 MCF/GBR, 米国特許出願第 07/727,006 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 5-502150 号（特表平 6-501122 号公報）。

(2) 「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）。

30

(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）。

(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、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）。

50

【0003】

なお、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07/727,058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分を構成するものとする。

【背景技術】

【0004】

最近は、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(Complex Instruction Set Computer -CISC)の使用からより単純化された縮小命令セット・コンピュータ(Reduced Instruction Set Computer-RISC)アーキテクチャまでに成熟している。CISCアーキテクチャの特徴は、命令実行パイプラインを実現し、サポート
10
が大部分ハードウェアで行われていることである。従来の代表的パイプライン構造は、命令フェッチ・ステージ、命令デコード・ステージ、データ・ロード・ステージ、命令実行ステージおよびデータ・ストア・ステージを一定の順序で含んでいる。命令セット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並行に実行させると、パフォーマンスの向上が得られる。パイプラインが長くなると、利用できる実行ステージの数がそれだけ多くなり、かつ同時並行に実行できる命令数が多くなる。

【0005】

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、先行す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きブランチ命令を評価できないことである。従って、条件付き命令の以後の
20
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・ステージは複数のプロセッサ・サイクルの間インアクティブのままになっている。代表例として、条件コードがプロセッサ・ステータス(状況)レジスタ(processor status register-PSR)とも呼ばれる条件コード・レジスタに書かれるのは、実行ステージを通して命令の処理が完了したときだけである。そのため、ブランチ条件コードが判断されるまで、条件付きブランチ命令を複数のプロセッサ・サイクルの間デコード・ステージに残し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、スループットが大幅に損失することになる。さらに、コンピュータの平均的スループットは、条件付きブランチ命令がプログラム命令ストリームの中で条件コード設定命令のあとに接近して何回現れるかによって左右される。
30

【0006】

もう1つの問題は、プログラム命令ストリームの中で近接して現れ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にあるという事実から起こる。データ・レジスタは順次の命令のストア・ステージとロード・ステージでデータの宛先または
40
ソースとしてよく使用される。一般的に、データをレジスタ・ファイルにストアする命令は、次の命令のロード・ステージ処理がレジスタ・ファイルにアクセスできるようにする前に、少なくとも実行ステージでの処理を完了していなければならない。多くの命令は、ストア・データを得るためにその実行に1つの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持続している間、パイプライン全体が停止され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、実行される命令ストリームの内部順序に実質的に左右される。

【0007】

第3の問題は命令自体の実行が原因で起こるのではなく、マイクロプロセッサ自体のハードウェアがサポートする命令実行環境、つまり、マシンの状態(state-of-the machine)の維持から起こる問題である。現在のCISCマイクロプロセッサ・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たことを検出することができる。トラップには、ハードウェア割込み、ソフトウェア・トラップおよび例外がある。各トラップが現れたときは、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は、トラップ処理ルーチンの即時実行を可能にするために実行パイプラインをクリアする必要がある。それと同時に、トラップが生起した正確な点で、つまり、
50

そのとき実行中の最初の命令が割込みとトラップのために終了したときであり、かつ、例外が原因で失敗した命令の直前に生じた正確な点でその時点のマシンの状態を設定する必要がある。引続き、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する必要がある。その結果、各トラップまたは関連事象が発生すると、処理ルーチンの開始時と終了時、および正確なマシンの状態のストアと返却時の双方においてパイプラインをクリアすることにより待ち時間が生じ、プロセッサのスループットがそれに応じて減少することになる。

【0008】

これらの問題を解決するために、CISCアーキテクチャの滞在的スループットを向上するための種々の試みが行われている。条件付きブランチ命令が正しく実行されたことを想定すれば、ブランチ条件コードが最終的に判断される前にパイプライン実行を暫定的に進めることができる。また、レジスタが変更されるかどうかについても想定を行うことにより、そのあとに続く命令を暫定的に実行させることもできる。最後に、ハードウェアを実質的に追加すれば、処理ルーチンの実行を必要とする例外の発生を最小にすることができるので、プログラム命令ストリームの処理に割込みをかける（中断させる）例外の発生回数を少なくすることができる。

10

【0009】

これらの解決方法は、ハードウェアをさらに実質的に複雑化することは明らかであるが、その解決方法自身にもそれぞれの問題がある。ブランチ条件またはレジスタ・ファイル・ストア・アクセスのいずれかが最終的に解決する前に命令の実行を継続させるためには、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点のいずれか、レジスタ・ファイルの各変更、および例外が何か発生した場合には、最後のいくつかの命令の実行が完全に終了する以前の点にマシンの状態を復元可能にする必要がある。その結果、これをサポートするためのハードウェアが別に必要になり、しかも、どのパイプライン・ステージのサイクル・タイムも、大幅に増加しないように特別に設計しなければならない。

20

【0010】

RISCアーキテクチャでは、上述した問題の多くを回避するために、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現実を大幅に簡略化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、およびストア・サイクルを含む3つのパイプライン化プログラム・サイクルにおいてのみ実行される。公知のRISCアーキテクチャでは、ロードとストア・データをバイパスする手法を用いることによって、3ステージ・パイプラインでサイクルごとに1つの命令を実行することを可能にしている。

30

【0011】

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、必要とする機能を実行するためのソフトウェア・ルーチンに有利になるように最小化されている。その結果、RISCアーキテクチャは、最適に適合されたパイプラインによって実行される単純なロード／ストア命令セットを使用することにより大幅な柔軟性と高速化が得られるという希望を与えている。しかも、実際には、RISCアーキテクチャは、高性能パイプラインを短くすることと、必要とするすべての機能を実現する命令数を実質的に増加して実行する必要性とをバランスよく調和させると、利点が得られることが判明している。

40

【0012】

RISCアーキテクチャの設計は、一般的に、ブランチ、レジスタ参照および例外の面で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。RISCアーキテクチャに関係するパイプラインは短く、スピードの面で最適化されている。パイプラインを短くすると、パイプライン停止(pipeline stall)またはクリアが起こったときの結果が最小になり、マシンの状態を以前の実行個所に復元する際の問題が最小になる。

【0013】

しかし、一般的に認識されている現在のレベル以上にスループット・パフォーマンスを

50

大幅に向上させることは、公知のRISCアーキテクチャでは容易に達成することができない。その結果、いわゆるスーパースカラー(super-scaler)と呼ばれる代替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、複数の命令を同時並行に実行することにより、プロセッサのスループットを比例的に増加させることを試みている。残念ながら、このようなアーキテクチャの場合も、CISCアーキテクチャに起こっている問題と同一ではないが、類似した条件ブランチ、レジスタ参照、および例外処理の問題が起こっている。

【0014】

従来のスーパースカラー型アーキテクチャに特に起こっている問題は、一般的に、アーキテクチャ自体に複雑性が内在しているため、アーキテクチャの基礎面を大幅に設計し直さない限り、アーキテクチャの変更ができないことである。同時並行に実行される複数の命令の実行を処理する場合は、命令ストリームの実行の正確さを確実に保つために、アーキテクチャに実質的な制御上の制約がある。事実、ある種の命令は、プログラム命令ストリームの中で先に置かれている命令の実行前に、その実行が完了することがある場合がある。その結果、命令実行の基礎面を管理する制御ロジックさえも設計し直さなければ、特定の命令の実行フローに影響を与えるアーキテクチャ上の変更ができない場合がよくある。

【発明の開示】

【発明が解決しようとする課題】

【0015】

従って、本発明の一般的目的は、計算を補強する機能ユニットを追加し、変更することによりアーキテクチャ機能の強化が容易なRISCベースの高性能スーパースカラー型プロセッサ・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】

【0016】

本発明によるマイクロプロセッサは、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プロセッサであって、

複数の命令を、前記機能ユニットでの実行に先立って、予め定めたプログラム順序でメモリからプリフェッチするプリフェッチユニットと、

該プリフェッチユニットの後段に設けられ、複数の命令を保持する命令バッファと、

前記命令バッファの後段に設けられ、前記命令バッファに保持された命令のうちの複数の命令を同時にデコードするデコーダと、

該デコードされた命令を前記機能ユニットにより実行する際に用いられる複数のレジスタと、複数の命令を順序外で実行した結果を蓄える実行結果バッファとを備えたレジスタファイルと、

前記デコーダから出力された前記複数の命令の依存関係を、少なくとも前記レジスタの使用条件に基づいて検出する依存関係チェックユニットと、

前記依存関係チェックユニットにより依存関係に基づく実行上の制約がないと判断された命令については、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命令発行ユニットと、

前記複数の命令を前記順序外での実行が完了した際に前記実行結果バッファに実行結果を格納し、前記順序外で実行された命令をリタイアする際に前記実行結果バッファ内の実行結果を前記対応するレジスタに書き込むリタイアユニットと、
を備え、

前記デコードされた命令であってリタイアされていない命令のロケーションは、該命令によって用いられる前記実行結果バッファのロケーションに対応づけられており、

前記命令がリタイアするとき、該リタイアされる命令のロケーションに対応づけられていた前記実行結果バッファのロケーションが開放される。

【0017】

本発明のスーパースカラプロセッサによれば、デコード済みでリタイア前の命令のロケーションに、その命令によって用いられる実行結果バッファのロケーションを対応づけているので、実行中の命令と実行結果バッファのロケーションとの対応付けが容易であり、命令の実行結果を実行結果バッファ内の適切なロケーションに容易に格納することが可能である。また、命令がリタイアするとき、リタイアされる命令のロケーションに対応づけられていた実行結果バッファを開放するので、その実行結果バッファのロケーションを他の命令のために直ちに利用することができる。

【発明を実施するための最良の形態】

【0020】

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 10

【0021】

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	20
2)	PC制御アルゴリズムの詳細	
E)	割込みおよび例外の処理	
1)	概要	
2)	非同期割込み	
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	
A)	IEUデータ経路の詳細	30
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)	IEU制御経路の詳細	
1)	Eデコード・ユニットの詳細	
2)	キャリー・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	
4)	レジスタ改名ユニットの詳細	40
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	
IV.	仮想メモリ制御ユニット	
V.	キャッシュ制御ユニット	
VI.	要約及び結論	

以下に目次に従って説明する。

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100の概要を示すものである。命令フェッチ・ユニット(IFU)102と命令実行ユニット(IEU)104は、アーキテクチャ100の中心となる機能要素である。仮想メモリ・ユニット(VMU)108、キャッシュ制御ユニット(CUU)106、およびメモリ制御ユニット(MCU)110は、IFU102とIEU104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(MAU)112は基本的要素として、アーキテクチャ100を動作させるためのものである。もっとも、MAU112はアーキテクチャ100の1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、IFU102、IEU104、VMU108、CCU106、およびMCU110は従来の0.8ミクロン設計ルール of 低電力CMOS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1,200,000個のトランジスタから構成されている。アーキテクチャ100の標準プロセッサまたはシステムのクロック速度は40MHz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160MHzである。

10

IFU102の基本的役割は命令をフェッチし、IEU104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。

【0022】

本発明の好適実施例では、各命令は長さが32ビットに固定されている。命令セット、つまり、4個の命令からなる「バケット」(bucket)は、CCU106内の命令用キャッシュ132から128ビット幅の命令バス114を経由してIFU102によって同時にフェッチされる。命令セットの転送は、制御ライン116経由で送られてきた制御信号によって調整されて、IFU102とCCU106間で行われる。フェッチされる命令セットの仮想アドレスは、IFU仲裁、制御およびアドレスを兼ねたバス118経由でIFU102から出力され、さらにIEU104とVMU108間を結合する仲裁、制御およびアドレス共用バス120上に送出される。VMU108へのアクセスの仲裁(arbitration)は、IFU102とIEU104の両方がVMU108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。

20

30

【0023】

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・ファイルへの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IFUへ送り返される。

40

【0024】

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。IEU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

50

は、IEU 140 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。

【0025】

CCU 106 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、CCU 106 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。

【0026】

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、CCU 106 は対応する物理アドレスをMCU 110 へ渡し、MAU 112 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CCU 106 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IFU 102 またはIEU 104 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渡される。

【0027】

MCU 110 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 CCU 106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 PoPn 146o-n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。MCU 110 によって処理される各メモリ・アクセス要求は、MAU 112 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でポート146o-n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146o-n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MAU 112 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、MCU 110 は、実際には、CCU 106 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、MCU 110 に1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。

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関与する場合を考慮して説明する。

A) IFU データ経路

IFU データ経路は、命令セット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 からの命令セットは I デコード・ユニット262 を通ってIFIFO ユニット264 へ渡される。命令FIFO 264の最後の2ステージにストアされた命令セットは、データ・バス278、280 を通してIEU 104 に連続的に取り出して利用することができる。

【0028】

プリフェッチ・バッファ・ユニット260 は、一度に1つの命令セットを命令バス114 から受け取る。完全な128 ビット幅命令セットは、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。追加の命令セットは最高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、MBUF 188、TBUF 190またはEBUF 192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セットは、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、MBUF 188、TBUF 190およびEBUF 192をバイパスするためのものである。

【0029】

好適アーキテクチャ100では、MBUF 188は名目的またはメイン命令ストリーム中の命令セットをバッファするために利用される。TBUF 190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セット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方向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、MAU 112の待ち時間は長くなるとしても、少なくともCCU 112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セットを得て、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、MBUF 188とMBUF 190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、正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、MBUF 188に入れられた場合は、TBUF 190に残っている命令セット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セットがTBUF 190に存在する場合は、命令プリフェッチ・バッファ・ユニット260を通して、これらの命令セットがTBUF 190から直接に、並行にMBUF 188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MBUF 188にストアされた命令セットは、TBUF 190から転送された命令セットを重ね書きすることによって、事実上無効にされる。MBUFロケーションへ転送するTBUF命令セットがなければ、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。

【0030】

同様に、EBUF 192は、プリフェッチ・バッファ260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。EBUF 192は、好ましくは、MBUF 188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MBUF 188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260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TBUF 190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すべて解決される。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、TBUF 190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、ターゲット命令セットはすでにプリフェッチされてTBUF 190に入れられているので、EBUF 192へ並列に転送することができる。

【0031】

最後に、MBUF 188、TBUF 190およびEBUF 192の各々はプリフェッチ・バッファ出力バス196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セットを出力バス196上に送出するようになっている。さらに、バス194を通過するフローは、命令セットを命令バス114から直接に出力バス196へ転送するためのものである。

好適アーキテクチャ100では、MBUF 188、TBUF 190、EBUF 192内のプリフェッチ・バッファは直接的にはFIFO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

10

20

30

40

50

ンも出力バス196 に接続されているので、命令用キャッシュ132 から取り出された命令セット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102 は命令ストリームに一定順序で並んだ命令順に命令セットを判断して、要求するのが一般的になっている。しかし、命令セットがIFU 102 へ返されるときの順序は、要求したある命令セットが使用可能で、CCU 106 だけからアクセス可能であり、他の命令セットはMAU 102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。

【0032】

命令セットは一定順序でプリフェッチ・バッファ・ユニット260 へ返されない場合があっても、出力バス196 上に出力される命令セットの列は、一般的に、IFU 102 から出された命令セット要求の順序に従っていなければならない。順序内 (in-order) 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。

【0033】

I デコード・ユニット262 は、IFIFO ユニット264 のスペースが許すかぎり、プリフェッチ・バッファ出力バス196 から命令セットを、普通は1サイクルに1つの割合で受け取る。1つの命令セットを構成する4個の命令からなる各セットはI デコード・ユニット262 によって並列にデコードされる。関係の制御フロー情報がIFU 102 の制御経路部分のためにライン318 から抜き出されている間は、命令セットの内容はI デコード・ユニット262によって変更されない。

I デコード・ユニット 162からの命令セットはIFIFO 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、IFIFO 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FIFO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、FIFOオペレーションの後半サイクル時に命令セットがI デコード・ユニット262 からマスタ・レジスタに直接にロードされるようになっている。しかし、マスタ・レジスタを入力バス198 からロードすることは、IFIFO ユニット264 内でデータをFIFOシフトすることと同時に行う必要はない。その結果、命令FIFOユニット264 内にストアされた命令セットの現在の深さに関係なく、さらに、IFIFO ユニット264 内でデータをFIFOシフトすることから独立して、入力バス198 から連続的にIFIFO ユニット264 に入れていくことができる。

【0034】

マスタ／スレーブ・レジスタ200、204、208、212、216、224 の各々は、128 ビット幅命令セット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)、ブランチ・バイアス、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、IFIFO マスタ・レジスタに入力バス198 から新しい命令セットをロードするのと同様に、IFU 102 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セットと並行してIFIFO ユニット263 内で並列にシフトされる。

【0035】

最後に、好適アーキテクチャ100 では、IFIFO ユニット264 からの命令セットの出力は最後の2マスタ・レジスタ216、224 から同時に得られて、I Bucket 0とI Bucket 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IBASV0とIVASV1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

10

20

30

40

50

ス278、282、280、284はすべてIEU 104へ通じる命令ストリーム・バス124となるものである。

B)IFU 制御経路

IFU 102 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IFIFO 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IFU 102 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、IFU のオペレーションとIEU 104、CCU 106およびVMU 108 のオペレーションとの同期をとるようにしている。命令セットを選択して、MBUF 188、TBUF 190 およびEBUF 192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。

10

【0036】

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がMBUF 188であるか、TBUF 190であるか、EBUF 192であるかを指定している。プリフェッチ要求を受けて、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CCU 106 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。

20

【0037】

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個までの命令セットをプリフェッチ・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、MBUF 188での4命令セット、TBUF 190での2命令セット、EBUF 192での命令セット、およびフロー・スルー・バス194 経由で直接にIデコード・ユニット262 に渡される1命令セットである。第2は、命令セットが各々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、CCU 106 とのインタ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。

30

【0038】

IFU 102 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セットがCCU 106 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、CCU 106 からの命令セットの返却と一緒にID要求値が返されるようになっている。しかし、順序外の命令セット返却機能によると、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セットがあるので、ID値を再使用することが可能になる。従って、4ビット・カウンタは保持しておくのが好ましく、それ以降の命令セット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セットに関連づけられたものとなる。

40

【0039】

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MBUF 188、TBUF 190およびEBUF 192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MBUF 180内のそれぞれの命令セ

50

ット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 TD)、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。

【0040】

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 CCU 266 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、命令セット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 216 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を判断し、CCU IBUSY 制御ライン 300 の状態を調べて CCU 106 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、受け付け可能ならば、制御ライン 298 上の CCU IREAD 制御信号を肯定し、インクリメントされた ID 値を CCU 106 と結ばれた CCU ID 出力バス 294 上に送出する。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。

プリフェッチ ID は、要求が CCU 106 に出されるのと並行して、MBUF 188、TBUF 190、または EBUF 192 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。

【0041】

CCU 106 が以前に要求された命令セットを IFU 102 へ返却できるときは、CCU IREADY 信号が制御ライン 302 上で肯定され、対応する命令セット ID が CCU ID 制御ライン 296 上に送出される。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 内の命令セットの目標とする宛先を判別する。一致するものは 1 つだけが可能である。判別されると、命令セットはバス 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。

【0042】

PC ロジック・ユニット 270 は、以下で詳しく説明するように、IFU 102 全体を調べて、MBUF 188、TBUF 190 および EBUF 192 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係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セットの中から判別する。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 に存在する命令セット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・アドレスから命令セット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。従って、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、先行するプリフェッチ命令セットがMBUF 188またはEBUF 192へ送られたものと想定すると、プリフェッチの宛先をTBUF 190として選択する。プリフェッチ要求をCCU 106 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(CCU PADDR[13:4]) をアドレス・ライン324 を経由して直接にCCU 106 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、VMU 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(VMU VADDR [13:14]) をアドレス・ライン326 を経由してVMU 108 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、VMU 108 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るので、CCU 106 によって即時に使用される。

【0043】

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VMU 108 にオペレーション・エラーが起こると、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、他の例外が起こると発生する。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。

【0044】

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 を経由して制御信号を出力し、命令セットがI デコード・ユニット262 から入力バス198 経由で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 は命令セットを受け取るために、最も奥の使用可能なマスタ・レジスタ200、208、216、224 を選択する役割を持っている。マスタ・レジスタ202、210、218、226 の各々の出力は制御バス338 を経由して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セット内の最初の有効命令セットを指定している。つまり、CCU 106 から返された命令セットは、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セット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。従って、バッファ・アドレス値は、実行の対象として考慮される、命令セット内の最初の命令を一意的に示すために与えられる。

【0045】

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セット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滞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MBUF 188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、対応する命令セットはマークがつけられ、ストリーム・ビット値が1 となる。条件付命令セットはI デコード・ユニット262 によって検出される。条件付制

10

20

30

40

50

御フロー命令は最高4つまで命令セットに存在することができる。そのあと、命令セットは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。

【0046】

条件付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IEU 104 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命令が入っている命令セットの相対ロケーション、制御フロー検出ユニット274 から得られた命令セット内の条件付命令ロケーション・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セットは、このアドレスを使用してプリフェッチしてTBUF 190 に入れることができる。PCロジック・ユニット270 のために事前に選択されたブランチ・バイアスに応じて、IFIFO ユニット264 はMBUF 188またはTBUF 190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番目の命令セットが現れると、その命令セットはストリーム・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PCロジック・ユニット270 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ない。さらに、それ以降の命令セットはIデコード・ユニット262 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セットは1つも処理されない。

【0047】

本発明の好適実施例では、PCロジック・ユニット270 は、最高2個までの命令セット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命令セット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・アドレスは命令セット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。

【0048】

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、PCロジック・ユニット270 はブランチが行われる場合は、TBUF 190の内容をMBUF 188に転送し、TBUF 190の内容に無効のマークを付けるように、ライン316 上の制御信号によってプリフェッチ制御ユニット260 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セットがIFIFO ユニット264 にあると、IFIFO ユニット264 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セットに存在すると、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セットはプリフェッチされ、MBUF 188またはTBUF 190からの命令セットはブランチ・バイアスに応じてIデコード・ユニット262 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セットがIFIFO ユニット264 からクリアされる。

【0049】

IFIFO ユニット264 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IFIFO ユニット264に残っていて、最初の条件付きフロー命令セット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セットのターゲット・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セットはIデコード・ユニット262 を通して評価することが可能になる。従って、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あとで判断された場合に、それより上をクリアすべき命令セット・ロケーションにマークを付ける目的のために、潜在的制御フローの変化にマークを付けておき、IFIFO ユニット264 を通して追跡することができる。

10

20

30

40

50

【0050】

命令セットをマスタ・レジスタから実際にクリアするのではなく、IFIFO 制御ロジック・ユニット272 はIFIFO ユニット264 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・プ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336 に送出される制御信号でPCロジック・ユニット270 によって開始される。マスタ制御レジスタ202、210、218、226の各々の入力状況バス230を通してIFIFO 制御ロジック・ユニット272 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100では、これらのマスタ制御レジスタ202、210、218、226内のビットは、IFIFO ユニット264 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IFIFO 制御ユニット272 によってセットすることが可能である。この機能により、IEU 104 のオペレーションと非同期に、命令セットをマスタ・レジスタ200、208、216、224のいずれかに書き込み、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。

10

【0051】

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、命令セット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から次の該当命令セットを転送することを要求する。命令セットが転送されると、配列266内の対応する有効ビットがリセットされる。

20

C) IFU/IEU 制御インタフェース

IFU 102とIEU 104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、IFU 102は割込みオペレーションを通知し、IEU 104との同期をとることができる。外部で発生した割込み信号はライン292経由でロジック・ユニット270へ送られる。これを受けて、割込み要求制御信号がライン340上に送出されると、IEU 104は試行的に実行された命令をキャンセルする。割込みの内容に関する情報は、割込み情報ライン341を通してやりとりされる。IEU 104がPCロジック・ユニット270によって判断された割込みサービス・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、IEU 104はライン340上の割込み受信確認制御信号を肯定する。IFU 102によってプリフェッチされた割込みサービス・ルーチンがそのあと開始される。

30

【0052】

IFIFO 読取り(IFIFO RD)制御信号はIEU 104から出力され、最も奥のマスタ・レジスタ224に存在する命令セットが実行を完了したことおよび次の命令セット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。

40

【0053】

PCインクリメント要求とサイズ値(PC INC/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。

【0054】

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・

50

ユニット270 に返される。このターゲット・アドレスは、IEU 104 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きまるブランチ命令の仮想ターゲット・アドレスである。従って、ターゲット・アドレスを計算するためにIEU 104 のオペレーションが必要である。

制御フロー結果(CF RESULT) 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IFIFO ユニット264 に置かれている命令セット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。

10

【0055】

いくつかのIEU 命令リターン型制御信号(IEUリターン) が制御ライン350 上を送出されて、IEU 104 によってある命令が実行されたことをIEU 102 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IFU 102 に通知するために送られる。これらの信号をIEU 104 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。

20

【0056】

現命令実行PCアドレス(現IF PC) はアドレス・バス352 を経由してIEU 104 へ送られる。このアドレス値(DPC) はIEU 104 によって実行される正確な命令を指定している。つまり、IEU 104 が現在のIF PC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IEU 104 が判断すると、PC Inc/Size 信号がIFU 102 に送られ、即時に現在のIF PCアドレス値に反映される。

30

【0057】

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IEU 104 によってIFU 102 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IFU 102 が使用できるように、IEU 104 によってロードされ、あるいは計算される。

D)PCロジック・ユニットの詳細

PC制御ユニット362、割込み制御ユニット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。

40

PC制御ユニット362 はインターフェース・バス126 を通してプリフェッチ制御ユニット266、IFIFO 制御ロジック・ユニット272、およびIEU 104 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

50

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CCU PADDRアドレスとアドレス・ライン326上のVMU VMADDR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ることを担当する。

【0058】

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上に送られる場合もあるが、これは、PC制御ユニット362が現在のプリフェッチ・アドレスから命令セット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セットを選択しているか、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上に送られ、実行PC制御ユニット366へ渡される。

【0059】

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。

【0060】

プリフェッチ・アドレスのもう1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IEU 104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、IEU 104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。

【0061】

プリフェッチ仮想アドレスのもう1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。このアドレスは、割込み、トラップ、その他に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、IEU 104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PFPC信号の1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、

10

20

30

40

50

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。

【0062】

最後に、プリフェッチ仮想アドレスが取り出されるもう1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。IEU 104 によって計算またはロードされたアドレス値、またはすくなくともベース・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、プロシージャおよび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IEU 104 を通して処理することが可能である。

10

【0063】

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IEU 104 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

20

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。

【0064】

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・セクタ(PF PC SEL) であり、これは現プリフェッチ仮想アドレスの中央セ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・セクタから出力バス392 を通ってインクリメンタ・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 を通ってレジスタMBUF PFnPC 398、TBUF PFnPC 400、およびEBUF PFn PC402 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MBUF 188、TBUF190、およびEBUF 192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・セクタ390 へ渡される。従って、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替えを指示することができる。ストリームの中の次の命令セットをプリフェッチするために、そのアドレス値はインクリメンタ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 として示されているが、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・ルーティン・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る。PC制御ユニット362 の制御を受けて、これらのリターン

30

40

50

・アドレス・レジスタはバス35'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IEU 104 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。

【0065】

特殊レジスタ・ブロック412 内のセレクトは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・セレクト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・セレクト390 へ渡される。ベース・アドレス値は割込み制御ユニット363 から割込みオフセット・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・セレクト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。

10

【0066】

プリフェッチ・セレクト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IFIFO 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、あらかじ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、TBUF 190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の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・セレクト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 PFnPC 400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。

20

30

【0067】

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IEU 104 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。

【0068】

プリフェッチPF PCセレクト390 を通っ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。従って、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 ページ変換は不要である。プリフェッチ・セレクト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 のほかに、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

40

50

持される。

【0069】

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、次の命令セットを選択するために、命令セットを表す値（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。従って、最初のブランチ・ターゲット命令セット内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IFU 102 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セット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セット・アドレスの最初のプリフェッチのときだけ行われる。命令セットの最初の命令のアドレスと命令セット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。

10

【0070】

図4に示した機能ブランチの残り部分は実行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クタ(DPC SEL)430である。実行セクタ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 によってセクタ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クタ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。

20

【0071】

初期実行アドレスとその後のすべての新しいストリーム・アドレスは、アドレス・ライン440 を経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・セクタ390 からPFPCアドレス・バス370 を経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、あと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363 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個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 においておく。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クタ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。

30

40

【0072】

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366 を現実行アドレス列の生成から新実行ユニット・ストリーム列の生成に切り替えることができる。

【0073】

最後に、IF PCセクタ(IF PC SEL) 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IEU 104 へ送るためのものである。IF PCセクタ442 への入力の実行

50

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。殆どの場合、IF PCセクタ442 はPC制御ユニット262 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。

【0074】

実行PC制御ユニット366 は、すべての相対ブランチ・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'、340 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。

10

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ALU 450 へ送られる。ターゲットALU 450 への入力となるもう1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置を含んでいる。制御フロー変位計算ユニット452 はI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置を受け取る。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、現実行点アドレスに命令(CP PCによって要求された)とI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

【0075】

相対ターゲット・アドレスがターゲット・アドレスALU 450 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。

30

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Iデコードされると、命令セット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。

40

【0076】

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

50

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 I E U によって最終的に計算される。
 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 I E U 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、P F _ P C セレクタおよびインクリメンタを經由して送られ、最終的に M B U F P F n P C にストアされ、プリフェッチは 1. 2 から続けられる。 10
2. プロシージャ命令ストリームの処理：E B U F P F n P C
 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 M B U F 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 T B U F を使用できる。 20
 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セットを含んでいるプロシージャ命令ストリームは第 1 命令セット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 M B U F へ転送されるまで第 2 条件付き制御フロー命令セットに対してプリフェッチを停止する。
 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 I F U に維持されているプロシージャ・ベース・アドレス (P B R) レジスタに入っている値と結合される。この P B R 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。 40
 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 I F P C アドレスは D P C 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 (P S R) 内のプロシージャ進行中ビット (p r o c e d u r e - i n - p r o g r e s s b i t) がセットされる。 50

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タへ同時に送られ、(x16)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 PF n PCにストアされる。
2. 6 制御フロー命令がないと、32ビット・インクリメンタは各プロシージャ命令プリフェッチ・サイクルごとにEBUF PF n PCに入っているアドレス値を、(x16)だけ調整する。 10
2. 7 無条件制御フロー命令がI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 PF n PC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。
2. 8 次のプロシージャ・プリフェッチ命令セットのアドレスはEBUF PF n PC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I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(x16)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 PF n PCレジスタに返される。 30
- 3 ブランチ命令ストリームの処理：TBUF PF n PC
3. 1 MBUF命令ストリームの中の最初の命令セットに現れた条件つき制御フロー命令がI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。
3. 2 「ブランチを行うバイパス」の場合： 40
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セクタとインクリメンタ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 PF n PC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TBUFに入れられたあとで実行するためにIFI FOに送られる。IFI FOとTBUF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 50

- 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 番目の命令セット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)。
3. 2. 6 最初の命令セット内の条件付きブランチを「行う」と解釈された場合： 10
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 命令セットであったときは 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セット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。
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。 20
3. 2. 6. 4 2 番目の条件付きブランチ命令セットが 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)。
3. 2. 7 最初の条件付き命令セット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： 30
3. 2. 7. 1 ターゲット命令ストリームからの命令セットと命令の 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 「ブランチが行われえないバイアス」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。
3. 3. 1. 1 最初の条件付き命令セット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。 40
3. 3. 1. 2 最初の条件付き命令セット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 番目の命令セット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 最初の条件付きブランチのターゲット・アドレスが計算 50

されると、T B U F P F n P C 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T B U F 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、ブランチ・ターゲット命令は最初の命令セット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）。

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 ブランチのソースがE B U F 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、M B U F またはE B U F をフラッシュし、最初の条件付きブランチ命令セットのあとに置かれたメイン・ストリームからの命令のI F I F O とI E U をフラッシュする。

10

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、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。

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T B U F 命令をM B U F またはE B U F へ転送する。

20

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、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. 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

い」と解析された場合：

3. 3. 4. 1 ターゲット命令ストリームからの命令セットのT B U F をフラッシュする。

3. 3. 4. 2 2 番目の条件付きブランチ命令が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、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ける。

30

3. 3. 4. 3 2 番目の条件付きブランチ命令が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

4. 1 トラップは広義には次のものからなる。

【0077】

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。

40

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。

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

50

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。
 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。 10
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
 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。 20
 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。
 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。 30
 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。
 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。 40
 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 x P C 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 50

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

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。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。

【0078】

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。非登記割り込みは、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。

20

30

【0079】

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

【0080】

ソフトウェア・トラップ命令は CF_DET_274 (図2) によって I デコード・ページで検出され、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

50

た特定命令と対応付けられて記録され、累積されるだけで、実行時に処理される。

2) 非同期割込み

非同期割込みは、割込みライン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込みは最高32までユーザが定義することができ、割込みを要求した特定外部デバイスは割込みレベル・ライン(LVL) 上に割込み番号(0-31) を送出する。メモリ・エラー・ラインはMCU 110 によってアクティベート(活動化)され、様々な種類のメモリ・エラーを通知する。その他の非同期割込みライン(図示せず)も割込みロジック・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/カウンタ割込み、メモリ入出力(I/O)エラー割込み、マシン・チェック割込み、およびパフォーマンス・モニタ割込みを要求するためのラインがある。

10

非同期割込みの各々は、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に関連づけられている。これらのトラップ番号は32個が32個のハードウェア割込みレベルと関連づけられている。これらのトラップ番号のテーブルは割込みロジック・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。

20

【0081】

非同期割込みの1つが割込みロジック・ユニット363 に通知されると、割込み制御ユニット363 は割込み要求をINT REQ/ACK ライン340 を経由してIEU 104 へ送出する。また、割込み制御ユニット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262 へ送信し、PC制御ユニット262 に命令をプリフェッチすることを中止させる。IEU 104 はそのとき実行中の命令をすべてキャンセルし、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 内にDPC は、IEU 104 は割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IFIFO 264 に置かれている他の命令もすべてキャンセルされる。

30

【0082】

IEU 104 は、割込みハンドラから割込みを受け取る準備状態にあるときだけ、割込み受信確認信号をINT REQ/ACK ライン340 を経由して割込み制御ユニット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。

40

3)同期例外

同期例外の場合は、割込み制御ユニット363 は各命令セットごとに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。割込み制御ユニット363 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。

【0083】

特定の命令セットがプリフェッチれている途中で、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270 へ、特に割込み制御ユニット334 へVMU 制御ライン332、334 を経由して送られる。割込み制御ユニット363 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343 を経由してPC制御ユ

50

ニット362 に通知する。これと同時に、割込み制御ユニット363 は、命令セット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 Excpビットのどちらか該当する方をセットする。そのあと、割込み制御ユニット363 は、命令セットの中の命令のどれも有効でないので、その命令セットに対応する4個の内部例外標識ビット全部をセットし、問題を起こした命令セットの中の4命令の各々に対応して受信した特定の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、問題の命令セットがIFIFO 264 内で最低レベルに達するまで通常通りに続行される。

【0084】

同様に、プリフェッチ・バッファ260 、 I デコード・ユニット262 またはIFIFO 264 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、この情報も割込み制御ユニット363 へ送られ、ユニット363 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、問題の命令セットがIFIFO 264 内で最低レベルに達するまで、通常通りに続行される。

【0085】

好適実施例では、プリフェッチ・バッファ260 、 I デコード・ユニット262 またはIFIFO 264 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タイプだけでる。ソフトウェア・トラップ命令は、CF DET ユニット274 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104 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おそ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104 に届くまでの待ちによる時間浪費もない。

【0086】

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274 によってI デコード・ステージで検出される。割込みロジック・ユニット363 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127 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づ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、割込み制御ユニット363 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362 に通知しない。その代わりに、命令がIFIFO 264 を通知してシフトされると同時に、IFU 102 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。

【0087】

命令セットがIFIFO 264 の最低レベルまで達すると、割込みロジック・ユニット363 はその命令セットの例外標識ビットを4ビット・ベクトルとしてSYNCH INT INFOライン341 経由でIEU 104 へ送り、命令セット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。IEU 104 は即時に回答しないで、命令セットの中のすべ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いった別の例外は、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すべての命令に対して最新となるようにするために、PSR 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。なんらかの同期命令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込みロジック・ユニット363

10

20

30

40

50

に通知される。

【0088】

IEU 104 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ストリームに現れたすべ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PC制御ユニット366 内のIF PC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、例外が割込み制御ユニット363 に通知される。

【0089】

例外の発生源である命令がリタイアされると、IEU 104 は、リタイアされる命令セット (レジスタ224)の中に同期例外を起こした命令があれば、どの命令であることを示した新しい4ビット・ベクトルを、命令セット内の最初の例外の発生源を示した情報と一緒に、SYNCH INT INFOライン341 を経由して割込みロジック・ユニット363 に返却する。IEU 104 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、割込みロジック・ユニット363 からIEU 104 に渡された4ビット例外ベクトルとIEU 104 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Iデコード時に検出された例外が原因で割込み制御ユニット363 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IEU 104 から割込み制御ユニット363 に返却される情報の残余部分は、割込み制御ユニット363 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。

10

20

4)ハンドラ・ディスパッチとリターン：

割込み受信確認信号がライン340 経由でIEU 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341 経由で受信された後、現DPC がリターン・アドレスとして特殊レジスタ412 (図4) の1つであるxPC 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(PSR) は先のPSR(PPSR) レジスタにもストアされ、そして現状態比較レジスタ(CSR) が特殊レジスタ412 の中の旧状態比較レジスタ(PCSR)にセーブされる。

【0090】

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。PCロジック・ユニット270 はトラップ用に2つのベース・レジスタをもち、これらは共に特殊レジスタ412 (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TBR である。

30

【0091】

割込み制御ユニット363 は現在保留中の最高優先度割込みまたは例外を判断し、索引(lookup) テーブルを通して、それに関連づ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INT OFFSETライン373 を経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'ビットとはi'番目のビットのことである。) 例えば、トラップの番号が0 から255 までで、これを8ビット値で表すと、ハンドラ・アドレスは8ビット・トラップ番号を22ビットのTBR ストア値の最後に連結すると求められる。トラップ番号に2桁に下位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1つとしてプリフェッチ・セクタPF PC Sel 390 (図4) へ送られ、次のアドレスとして選択されて、そこから命令がプリフェッチされる。TBR レジスタを使用してトラップのベクトル・ハンドラ・アドレスはすべて1ワードだけ離れている。従って、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

40

50

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、ROM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。

【0092】

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上に出す。ライン373'経由でプリフェッチ・セクタPF PC Sel 390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ビットが続く、そのあとに7個のゼロ・ビットが続いている。従って、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出たブランチ内でプログラムを実行させる。

10

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。

20

【0093】

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利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置けるだけの十分なスペースを設けておく必要がある。

30

【0094】

また、トラップ・ハンドラへディスパッチすると、プロセッサはカーネル・モードと割込み状態に入る。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。カーネルと割込み状態モードはプロセッサ状況レジスタ(PSR)内のビットで表される。現PSRの割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RT[24]～RT[31]が上述および図7(b)に示すように、見えるようになる。割込みハンドラは新しいモードをPSRに書き込むだけだカーネル・モードから出ることができるが、割込み状態から出するためには、トラップからのリターン(RTT)命令を実行する方法だけが唯一の方法である。

40

【0095】

IEU 104がRTT命令を実行すると、PCSRはCSRレジスタに復元され、PPSRレジスタはPSRレジスタに復元されるので、PSR内の割込み状態ビットは自動的にクリアされる。PF PC SELセクタ390は特殊レジスタ・セット412のなかの特殊レジスタxPC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。xPCはインクリメンタ394とバス396を通して、MBUF PFnPCまたはEBUF PFnPCのどちらか該当する方に復元される。xPCをEBUF PFnPCに復元すべきか、またはMBUF PFnPCに復元すべきかの判断は、復元されたあとのPSRの「プロシージャ進行中」ビットに従って行われる。

【0096】

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

50

レスをストアするのに同じ特殊レジスタxPC 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xPC 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uPC にストアされる。従って、割込み状態は、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あと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。

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、CSR、PSR、リタンPC、およびある意味では"A" レジスタ・セットre[24]～re[31]へディスパッチするとき自動的にバックアップがとられるが、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(FSR)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。

【0097】

トラップ・ハンドラへ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ない。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・レジスタから読み取り、その情報を適当に処理する必要がある。割込みは、トラップ・ハンドラへ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、再び割込みを可能にし、RTT 命令を実行して割込みから戻ることができる。

【0098】

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xPC 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xPC に移し、トラップからのリターン命令(RTT) を実行する必要がある。RTT は割込み状態を取り除き(PSR をPSR に復元することによって)、制御をxPC 内のアドレスを移す。xPC 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込みを可能にして、割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RT[24]～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。従って、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すべて復元し、元のxPC をトラップ・ハンドラ・スタックからポップし、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。

【0099】

表 I

トラップ番号	処理モード	同期	トラップ名
0-127	通常	同期	トラップ命令
128	通常	同期	F P 例外

10

20

30

40

50

1 2 9	通常	同期	整数算術演算例外	
1 3 0	通常	同期	MMU (TLB 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 I / O エラー	10
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	NMI	
0	高速トラップ	同期	高速MMU TLB 不一致	
1	高速トラップ	同期	高速MMU TLB 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)	

III. 命令実行ユニット

20

図 5 は、IEU 104 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、IFU 102 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 を通って、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(FUo-n) の並列配列へ送られる。機能ユニット478o-n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。

【0100】

ロード／ストア・ユニット484 によってIEU 104 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IEU 104 とCCU 106 間のデータ転送の管理を担当する。具体的には、CCU 106 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。CCU 106 のデータ用キャッシュにストアさ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。

30

IEU 104 の制御経路部分はIEU データ経路を通る情報の送出、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、IEU 104 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、それを応じて動作する。具体的には、命令セットは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セットを受信して、デコードする。8 命令すべてのデコードの結果は、キャリー・チェッカ(CRY CHKR)ユニット492 、依存性チェッカ(DEP CHKR)ユニット494 、レジスタ改名ユニット(REG RENAME)496、命令発行(ISSUEUR) ユニット498 およびリタイア制御ユニット(RETIRE CLT)500 へ送られる。

40

【0101】

キャリー・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、係統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。キャリー

50

・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ー・ビットに影響を与える、あるいはキャリー・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。

【0102】

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。

10

【0103】

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、キャリー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、実行できるように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ーに依存する命令は論理的にビット・マップから除かれる。

20

【0104】

必要とする機能ユニット478o-n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478o-n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へ送られ、また選択的に機能ユニット478o-n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。

30

【0105】

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・ルーチング・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478o-n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478o-nへ送るべきかどうか、あるいは機能ユニット478o-n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478o-nの各々への特殊レジスタ・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478o-nへデータを送ることを直接に制御する。

40

【0106】

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、先行命令もすべてリタイアされたならば、その命令は、確認ま

50

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上に送られると、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。

【0107】

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IFU 102へ送る。複数の命令を順序外で実行でき、従って、同時にリタイアする準備状態に置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 読取り制御信号を制御ライン342を経由してIFU 102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。

【0108】

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538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個所まで完了すると、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520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IFU 102へ送る。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。

【0109】

最後に、完了制御ユニット540は機能ユニット478o-n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478o-n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540は対応する完了制御信号を制御ライン542上に送出して、レジスタ改名ユニット496、命令発行ユニット498、リタイア制御ユニット500およびバイパス制御ユニット520にアラート（警告）する。

【0110】

機能ユニット478o-nを並列配列構成にすることにより、IEU 104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478o-nの特性を命令発行ユニット498に知らせる必要がある。機能ユニット478o-n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498以外は、IEU 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498と機能ユニット478o-nは共同して、残りの制御フロー管理ユニット496、500、520、528、540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478o-nの特定の制御フロー・オペレーションの変更は、IEU 104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478o-n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ALU、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478o-n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498を若干変更するだけですむ。必要なる変更を行うには、Eデコー

ド・ユニット490 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、その命令と必要とする機能ユニット478o-n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478o-n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。

A) IEU データ経路の詳細

IEU 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472 である。しかし、本発明によれば、IEU 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用意されている。主要データ経路は整数と浮動小数点の2つである。各並列データ経路内では、レジスタ・ファイル472 の一部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。

10

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550 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550 は一時バッファ552 、レジスタ・ファイル配列564 、入力セクタ559 、および出力セクタ556 を含んでいる。最終的にレジスタ配列564 へ送られるデータは、統合データ入力バス558'を経由して一時バッファ552 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550 へ送られるデータはすべて入力セクタ559 によって多重化されて、複数の入力バス55 (好ましくは2つの) から入力バス558'上に送出される。制御バス518 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552 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518 上に送出された制御信号は一時バッファ552 からレジスタ・ファイル配列564 内の論理的に対応づけられたレジスタへデータ・バス560 を経由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552 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560 のバイパス部分を経由して出力データ・セクタ556 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518 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564 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"を経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。

20

30

【0111】

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えるようになっている。従って、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564 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。

40

【0112】

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このシングル・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(I

50

ORD4)、I1RD、I1RD+1(ISRD)...I3RD、I3RD+1(I7RD)のセットとして使用することも可能である。

【0113】

本発明の好適実施例によれば、レジスタ・ファイル配列564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IFIFO 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。従って、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を経由して転送され、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。

10

【0114】

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、リタイア個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564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560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、IFIFO 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。

20

【0115】

命令I2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I2RDとI6RDの組合せが、命令I2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I4+7の実行は、命令I0-3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I0-3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I4+7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I0-3にシフトされるまで保留される。

30

【0116】

レジスタ・ファイル配列564の論理的編成は図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564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re[0..23]565、汎用レジスタre[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re[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。通常のオペレーションでは、汎用レジスタre[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。

40

【0117】

図7(b)に示すように、トラップ・レジスタ re[24..31]567をスワップしたアクティブ・レジスタ・セット"A"に移しておけば、レジスタre[0..23] 565のアクティブ・ベ

50

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に実行によって図7 (a) に示す状態に明示によって戻るまで維持される。

【0118】

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]5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]5の"B" セットとして論理的に参照することの可能である。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。

10

【0119】

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。

【0120】

20

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572、574 の幅と同じ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64ビットと1ビットになっている。

【0121】

多数の追加の特殊レジスタが、レジスタ配列472 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472 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(Kernel stack pointer)568、プロセッサ状態レジスタ(PSR)569、旧プロセッサ状態レジスタ(PPSR)570および8個の一時プロセッサ状態レジスタの配列(tPSR[0..7])571 からなっている。残りの特殊レジスタはアーキテクチャ100 の各所に分散している。特殊アドレスおよびデータ・バス354 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IDを指定するためのものである。

30

【0122】

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、そ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できるようにする継承メカニズム(inheritance mechanism)を現実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全方向に伝播され、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494 とキャリー依存関係チェッカ492 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。

40

50

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PSR 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PSR レジスタ569 に転送される。

その他の特殊レジスタは表 I I に説明されている。

【0 1 2 3】

表 I I

10

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。	20
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、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。	30
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。	40
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。	
P B R	W	プロシージャ・ベース・レジスタ：プロシージャ	50

		・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。	
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、割り込み可能ななどのビットがある。	10
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。	20
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したがって構成されるIEU 104の整数データ経路は図10に示されている。説明の便宜上、整数データ経路580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。

【0124】

データ経路580の入力データは位置合わせユニット582、584および整数ロード／ストア・ユニット586から得られる。整数即値(inteere immediate)データ値は、最初は命令埋込み(embedded)データ・フィールドとして与えられ、バス588経由でオペランド・ユニット470から得られる。位置合わせユニット582は整数データ値を隔離し、その結果値を出力バス590を経由してマルチプレクサ592へ送られる。マルチプレクサ592への別の入力

30

は特殊レジスタ・アドレスとデータ・バス354である。

命令ストリームから得られる即値(immediate)オペランドも、データ・バス594経由でオペランド・ユニット570から得られる。これらの値は、出力バス596上に送出される前に、位置合わせユニット584によって再度右寄せされる。

【0125】

【0126】

整数ロード／ストア・ユニット586は外部データ・バス598を通してCCU 106と双方向でやりとりする。IEU 104へのインバウンド・データは整数ロード／ストア・ユニット586から入力データ・バス600を経由して入力ラッチ602へ転送される。マルチプレクサ592とラッチ602からの出力データは、マルチプレクサ608のマルチプレクサ入力バス604、606上に送出される。機能ユニット出力バス482'からのデータもマルチプレクサ608に送られる。このマルチプレクサ608はアーキテクチャ100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610へ送る2つの経路を備えている。されに、マルチプレクサ608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100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

40

50

時バッファ612へ送ることができる。

【0127】

一時バッファ612からのデータは一時レジスタ出力バス616を経由して整数レジスタ・ファイル配列614へ、あるいは代替一時バッファ・レジスタ・バス618を経由して出力マルチプレクサ620へ転送することができる。整数レジスタ配列出力バス622は整数レジスタ・データをマルチプレクサ620へ転送することができる。一時バッファ612と整数レジスタ・ファイル配列614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612、レジスタ・ファイル配列614およびマルチプレクサ620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。従って、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。

10

【0128】

マルチプレクサ620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614から、あるいは一時バッファ612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IEU 104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IEU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込みや他の例外条件が起

20

【0129】

マルチプレクサ620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624を経由して整数バイパス・ユニット626へ送られる。このバイパス・ユニット626は、基本的に、マルチプレクサが並列の配列からなり、その入力いづれかに現れたデータをその出力のいづれかへ送ることができる。バイパス・ユニット626の入力は、マルチプレクサ592から出力バス604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624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586からダブル整数バス600を経由するロード・オペランド・データ、その出力バス596を経由して位置合わせユニット584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482からのバイパス・データ経路からなっている。このバイパス経路とデータ・バス482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。

30

【0130】

データはバイパス・ユニット626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628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586へデータを送るために使用されるストア・データ・バス632へ送られる。

40

【0131】

機能ユニット分配バス480はルータ・ユニット634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634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経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634はバイパス・ユニット626からバス630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352を経由して送られてきた現IF PCアドレス値、PC制御ユニット362によって判断され、ライン378'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634は、浮動小数点データ経路内に設けられたバイパス・ユニットからとり出されたオペランド・デー

50

タ値をデータ・バス636 を経由して受信することもできる（オプション）。

【0132】

ルータ・ユニット634 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354 上を転送されて、機能ユニット640、642、644 へ送られる。具体的には、ルータ・ユニット634 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646、648、640 を経由して機能ユニット640、642、644 の各々へ送る機能を備えている。本アーキテクチャ100 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640、642、644 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・ユニット機能をもたせることができる。

10

【0133】

ALU0機能ユニット644、ALU1機能ユニット642 およびシフト機能ユニット640 はそれぞれの出力レジスタ・出力を機能ユニット・バス482'上に出送する。ALU0とシフト機能ユニット644、640 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650 上に出送される。類似の浮動小数点機能ユニット出力値データ・バス652 が浮動小数点データ経路から機能ユニット出力バス482'へ設けられている。

【0134】

ALU0機能ユニット644 はIFU 102 のプリフェッチ操作と整数ロード／ストア・ユニット586 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IFU 102 のターゲット・アドレス・バス346 とCCU 106 の両方に接続された出力バス654 上に出送され、実行ユニットの物理アドレス(EX PADDR)が得られる。ラッチ646 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 上に出送されて、VMU 108 へ送られる。

20

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ソースは外部データ・バス598 を通してCCU 106 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。

30

【0135】

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 上に出送される。浮動小数点ロード／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 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 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67

40

50

6 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 に出力するように動く。

【0136】

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。

10

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 とファイルALU 714 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

20

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。

30

40

【0137】

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724 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

50

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 1 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。

【0138】

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、任意に組み合わせた最高 4 個までのシングル・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 上に送送することができる。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 上に送送してバイパス・ユニット 742 へ転送する。

【0139】

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 IEU 104 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。

【0140】

表 I I I

整数の比較		
条件*	記号	命令 条件コード
r s 1 は r s 2 より大	>	0 0 0 0
r s 1 は r s 2 より大か等しい	>=	0 0 0 1
r s 1 は r s 2 より小	<	0 0 1 0
r s 1 は r s 2 より小か等しい	<=	0 0 1 1
r s 1 は r s 2 と等しくない	≠	0 1 0 0
r s 1 は r s 2 と等しい	=	0 1 0 1
予備		0 1 1 0
無条件		1 1 1 1

* r s = レジスタ・ソース

表 I V

浮動小数点の比較

10

20

30

40

50

条件	記号	命令 条件コード	
r s 1 は r s 2 より大	>	0 0 0 0	
r s 1 は r s 2 より大か等しい	> =	0 0 0 1	
r s 1 は r s 2 より小	<	0 0 1 0	
r s 1 は r s 2 より小か等しい	< =	0 0 1 1	
r s 1 は r s 2 と等しくない	≠	0 1 0 0	
r s 1 は r s 2 と等しい	= =	0 1 0 1	
無順序	?	1 0 0 0	
無順序または r s 1 は r s 2 より大	? >	1 0 0 1	10
無順序、r s 1 は r s 2 より大か等しい	? > =	1 0 1 0	
無順序または r s 1 は r s 2 より小	? <	1 0 1 1	
無順序、r s 1 は r s 2 より小か等しい	? < =	1 1 0 0	
無順序または r s 1 は r s 2 と等しい	? =	1 1 0 1	
予備		1 1 1 0 - 1 1 1 1	

表 V

ブール演算		命令	
演算*	記号	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	30
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	40

* b s = ブール・ソース・レジスタ

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 760 の例を示したものである。データ経路 580、660 に別々に示されているが、ロード／ストア・ユニット 586、662 は 1 つの共用ロード／ストア・ユニット 760 として実現するのが好ましい。それぞれのデータ経路 580、660 からのインタフェースはアドレス・バス 762 およびロードとストア・データ・バス 764 (600、676)、766 (632、702) を経由している。

【0 1 4 1】

ロード／ストア・ユニット 760 によって使用されるアドレスは、IFU 102 および IEU 10 50

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。IFU 102 は仮想IFU 102 で動作し、CCU 106 とVMU 108 間の調整に依存して物理アドレスを生成するのに対し、IEU 104 ではロード／ストア・ユニット760 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・データをストア・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CCU 106 からロード／ストア・ユニット760 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760 はストア命令がIEU 104 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760 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760 に1つだけ存在することができ、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760 からCCU 106 へ転送し、CCU 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。

10

【0142】

具体的には、物理アドレス全体がVMU 108 からロード／ストア・アドレス・バス762 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7680-3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7703-0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作動し、ロード・アドレスとストア・アドレスをレジスタ7683-0、7703-0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7823-0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。4x4x32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7683-0、7703-0に入っているアドレスの各々が同時に力される。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。

20

30

【0143】

ロード・アドレスがVMU 108 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7680-3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の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CCU 106 へのアクセスの仲裁を行う。CCU 106 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7703-0から得られる。対応するストア・データ・レジスタ7823-0からのデータはCCU データ・バス792 上に送出される。

40

【0144】

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7683-0の1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7680-3は関係命令セット内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命令セット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 を経由してセクタ786 へ送られ、CCU PADDRアドレス

50

・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CCU 106 間でやりとれされる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CCU 106 に対して出され、CCU 106 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7680-3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セットを識別するために使用される。このID値は、従って、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。

【0145】

先行する要求ロード・データが使用可能であることがCCU 106 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。

【0146】

CCU 106 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CCU 106 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。

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。

【0147】

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P0-6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T0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。

【0148】

プロセッサ・サイクル0のときは、IFU 102 とVMU 108 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CCU 106 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。要求された命令セットが命令用キャッシュ132にあると、命令セットはプロセッサ・サイクル1のほぼ中間でIFU 102 へ戻される。そのあと、IFU 102 はプリフェッチ・ユニット260 とIFIF0 264 を経由する命令セットの転送を管理し、転送された命令セットはまずIEU 104 へ実行のために渡される。

1)Eデコード・ユニットの詳細

Eデコード・ユニット490 は全命令セット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共に表VIに示されている。

【0149】

表 V I

命令／仕様	
命令	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：

10

20

30

40

50

	加算、減算、乗算、シフトその他の指定	
	宛先レジスタ	
	P S R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：	10
	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	
	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	
即値コール 制御フロー	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	20
	インデックス・レジスタ、即値定数変位値、また はトラップ番号	
	レジスタ・セット A / B 選択	
特殊レジスタ移動	オペレーション機能コード：	
	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	
整数変換移動	オペレーション機能コード：	
	浮動小数点から整数への変換タイプの指定	30
	ソース / 宛先レジスタ	
	レジスタ・セット A / B 選択	
ブール関数	ブール関数コード：A N D、O R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	
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	40
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。

【 0 1 5 0 】

E デコード・ユニット 490 は命令セット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 490 の出力から得られる。この情報は再生成され、命令セット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 490 によってラッチ

される。従って、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490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つの保留中命令セット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。従って、8個のベクトルが制御ライン502を経由してキャリア・チェッカ492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510を経由して送られる。8個のベクトルが制御ライン506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。

10

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492は図14に示すオペレーションのデータ依存関係フェーズ期間の間に依存関係検査ユニット494を並列に動作する。キャリア・チェッカ・ユニット492は好適アーキテクチャ100では順列組合せ論理に基づくロジックをして実現されている。従って、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。

20

【0151】

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。従って、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。

30

40

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を経由して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もつことができる。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、実際に作成される出力データがないことを意味し

50

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。

【0152】

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。

10

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 を経由して、レジスタ依存関係を制御ライン508 を経由して受け取る。8 個のエレメントからのマトリックスも制御ライン542 を経由して受け取る。これらのエレメントは、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8 エレメント配列を制御ライン512 を経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 を経由して受信する。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。

20

【0153】

レジスタ改名ユニット496 はバス418 を経由してレジスタ・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1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。

30

【0154】

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。従って、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。

40

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 の出力を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 を経由して報告された機能ユニット4780-nの各々の状況に基づいてこの判断を行う。従って、命令発行ユニット498 は発行すべき使用可能な命

50

令セット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4980-n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断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。

【0155】

発行すべき命令を判断すると、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、「Execute 0」で示すように1つまたは2つ以上の機能ユニット4780-n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。

10

【0156】

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くだけで、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。従って、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。

20

【0157】

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セットの状況と機能ユニット4780-n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。従って、最適条件のとき、好適アーキテクチャ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

30

【0158】

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IEU 104 からクリアする必要がある。このような事態は、算術演算エラーに応答して機能ユニット4780-n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、外部割込みを受信し、それが割込み要求／受信確認制御ライン340 を経由してIEU 104 へ中継されたのに応答して、起こることがある。トラップ条件が発生したとき、命令発行ユニット498 は現在IEU 104 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の方式に対して割込みを正確に発生させるために不可欠である。IEU 104 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込みの受信を確認する。また、従来の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グラム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。

40

【0159】

50

プログラム制御の流れを変更するある種の命令は、I 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IFU 102 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、コール命令の時効時、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。

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478o-n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、現在保留中の命令セット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478o-n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出力をバイパス・ストリーム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478o-nからレジスタ・ファイル配列472 ヘ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

20

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ほかに、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セットをモニタする。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）がつけられると、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。命令セット全体がリタイアされると、IFIF0 読取り制御信号が制御ライン342 上に送出されてIFIF0 264 を前進させる。

30

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、現在の保留中命令セット内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IFU 102 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 を経由して取得する。現在のレジスタ依存関係のセットは、制御ライン536 を経由してデータ依存関係チェック・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 を経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bool・レジスタが判別される。従って、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。

40

【0160】

制御フロー命令の実行と同時に、バイパス・ユニット472 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 を経由して制御フローの結果を制御フロー制御ユニット538 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 を経由してIFU 102 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

50

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。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。

9) バイパス制御ユニットの詳細

命令発行ユニット498はバイパス制御ユニット520と緊密に協働して、レジスタ・ファイル配列472と機能ユニット478o-n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を通して確認することができる。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上に送出された出力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

IV. 仮想メモリ制御ユニット

VMU 108のインタフェース定義は図15に示されている。VMU 108はおもに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。VMU 108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、仮想アドレスの表示は、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、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ない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

30

【0161】

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、MAU 112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MAU 112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IEU 104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。

40

【0162】

VMU制御ロジック800はIFU 102およびIEU 104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IEU 104へ送られ、VMU 108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、VMU 108は常にIFU 102の変換要求を受け付ける準備状態にある。IFU 102およびIEU 104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VMU 108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だけがIEU 104に出力される。

50

【0163】

IFU 102 および IEU 104 は共に、スペースIDと仮想ページ番号フィールドを、それぞれ、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ExID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り、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

【0164】

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。

【0165】

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出力は特殊レジスタ・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

30

V. キャッシュ制御ユニット

CCU 106 用のデータ・インタフェース上のコントロールを図17に示す。この場合も、インタフェースはIFU 102 用とIEU 104 用に別個に設けている。さらに、論理的に別個になったインタフェースをCCU 106 に設けて、MCU 110 と命令とデータの転送が行われる。IFU インタフェースはアドレス・ライン324 上に送出される物理ページ・アドレスと、アドレス・ライン824 上に送出されるVMU 変換ページ・アドレスと、制御ライン294、296 上に別個に転送される要求IDとからなっている。単方向データ転送バス114 は命令セット全体を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(read/busy) 制御信号と準備(ready) 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。

40

【0166】

同様に、完全な物理アドレスはIEU 102 から物理アドレス・バス788 を経由して送られる。要求ExIDはIEU 104 のロード／ストア・ユニットとの間で制御ライン796 を経由して別個に送受される。80ビット幅の単方向データ・バスをCCU 106 に設けて、IEU 104 と結んでいる。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 内に用意し、サポートしたのは、浮動小数点データ経路660 を変更することにより、IEEE標準754 に準拠する浮動小数点オペレーションをサポートするアーキテクチャ100 の将来の実現をサポートするた

50

めである。

【0167】

IEU 制御インタフェースは、要求、使用中、準備、読み書きおよび制御信号784 を通して確立され、IFU 102 によって使用される対応する制御信号とほぼ同じである。ただし、ロード・オペレーション・ストア・オペレーションを区別するために読み書き制御信号が用意されている点異なる。幅制御信号はIEU 104 がCCU 106 をアクセスする都度、その間に転送されるバイト数を指定している。これに対して、命令キャッシュ132 のアクセスは、いずれも、固定128 ビット幅データ・フェッチ・オペレーションである。

【0168】

CCU 106 は、命令用キャッシュ132 とデータ用キャッシュ134 に対してほぼ従来のキャッシュ制御機能を実装している。好適アーキテクチャ100 では、命令用キャッシュ132 は256 個の128 ビット幅命令セットをストアできる高速メモリになっている。データ用キャッシュ134 は1024個の32ビット幅データ・ワードをストアすることができる。命令用キャッシュ132 およびデータ用キャッシュ134 の内容から即時に満たすことができない命令とデータ要求はMCU 110 に引き渡される。命令用キャッシュが不一致（ミス）の場合は、28 ビット幅の物理アドレスがMCU 110 ヘアドレス・バス860 を経由して送られる。要求IDおよびCCU 106 とMCU 100 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862 上に送出される。MCU 110 がMAU 112 の必要な読取りアクセスを調整すると、2つの連続する64ビット幅データ転送がMAU 112 から直接に命令用キャッシュ132 へ行われる。データ・バス136 が好適アーキテクチャ100 において64ビット幅であるとする、2転送が必要である。要求したデータがMCU 110 を経由して返送されるとき、要求オペレーションが保留されていた期間に保持されていた要求IDも、制御ライン862 を経由してCCU 106 へ返送される。

【0169】

データ用キャッシュ134 とMCU 110 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場合の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32ビット幅物理アドレスがアドレス・バス864 を経由してMCU 110 へ送られる。インタフェース制御信号と要求ExIDは制御ライン866 を経由して転送される。双方向64ビット幅データ転送はデータ用キャッシュ・バス138 を経由して行われる。

【0170】

高性能RISC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とおりである。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確認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。

【0171】

従って、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】

【0172】

【図1】 本発明のマイクロプロセッサ・アーキテクチャを示す簡略ブロック図である。

【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・ファイルと命令セットを説明する図である。

【図 7】1 次整数レジスタ・セットの再構成可能状態を示す図である。

【図 8】浮動小数点および 2 次整数レジスタ・セットを示す図である。

【図 9】3 次ブール・レジスタ・セットを示す図である。

【図 10】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。

。

【図 11】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】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】ロード／ストア・ユニットを示す詳細ブロック図である。

10

【図 14】複数の命令を実行する際のオペレーション・シーケンスを示すタイミング図である。

【図 15】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】仮想メモリ制御アルゴリズムを図形で示す図である。

【図 17】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】

【0173】

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)

20

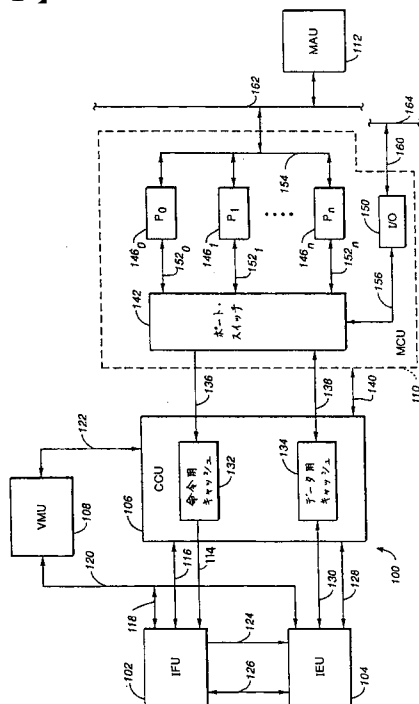
106...キャッシュ制御ユニット (CCU)

108...仮想メモリ・ユニット (VMU)

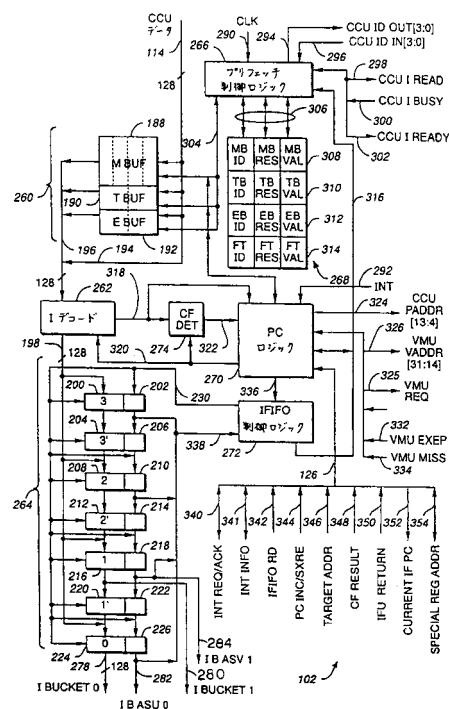
110...メモリ制御ユニット (MCU)

112...メモリ・アレイ・ユニット (MAU)

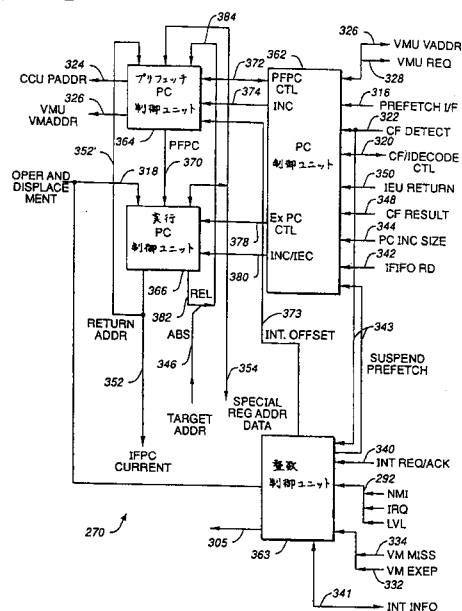
【図 1】



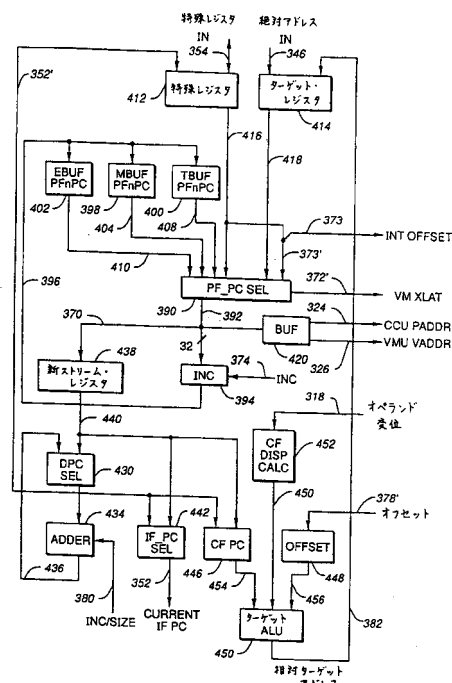
【図 2】



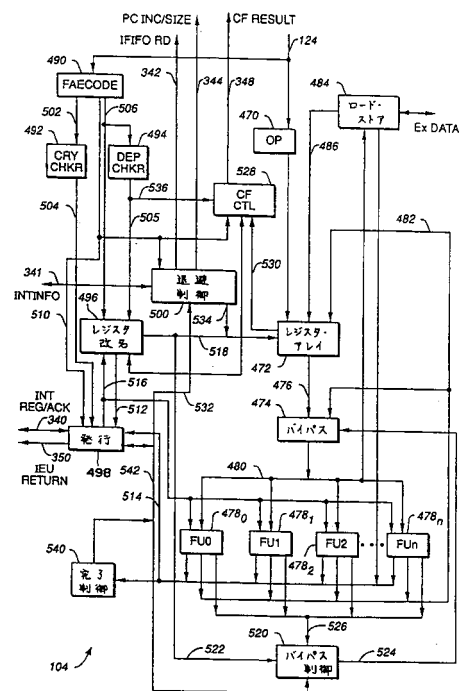
【图 3】



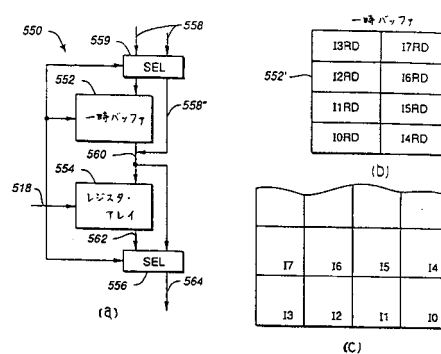
【图 4】



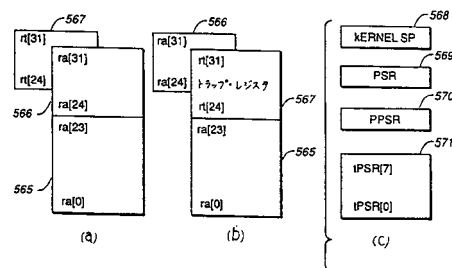
【図 5】



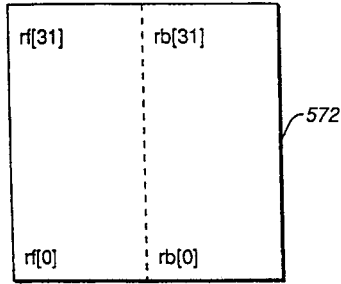
【图 6】



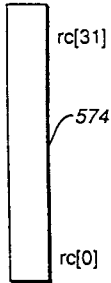
【图 7】



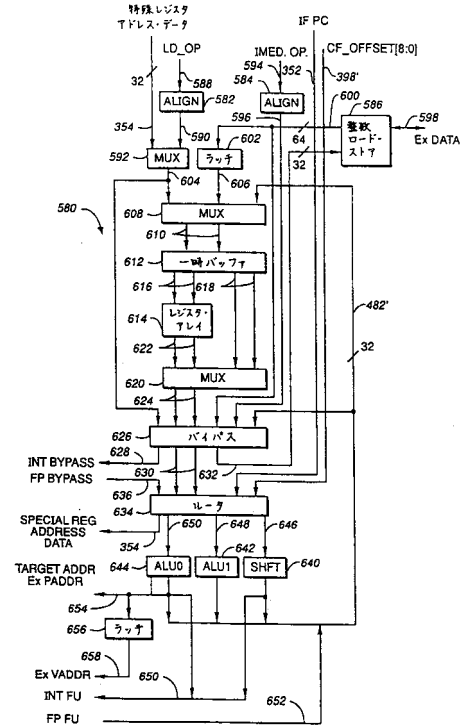
【図 8】



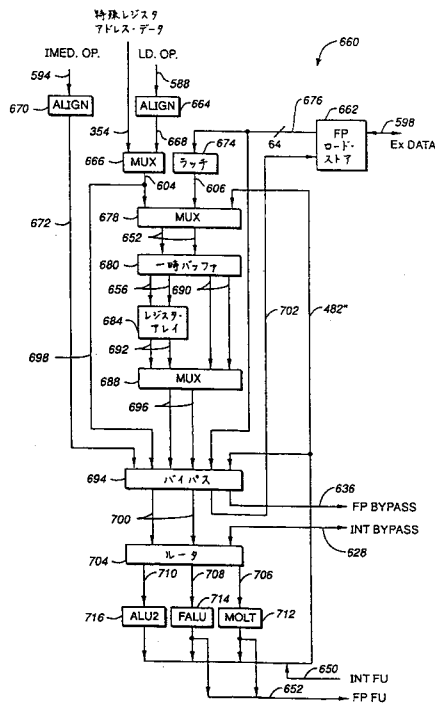
【図 9】



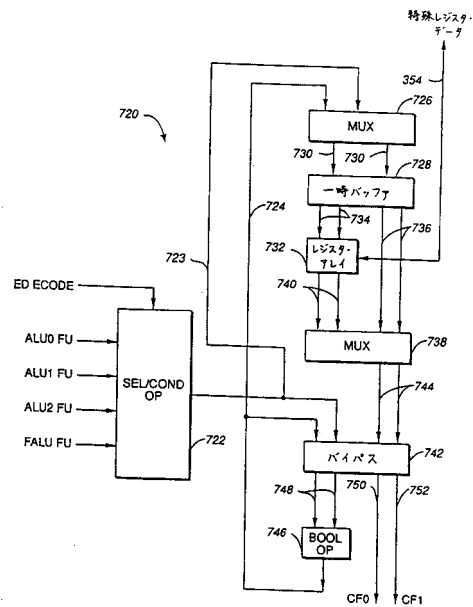
【図 10】



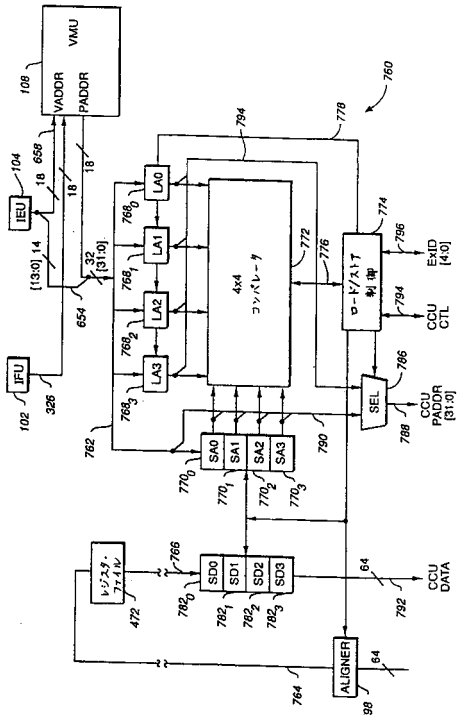
【図 11】



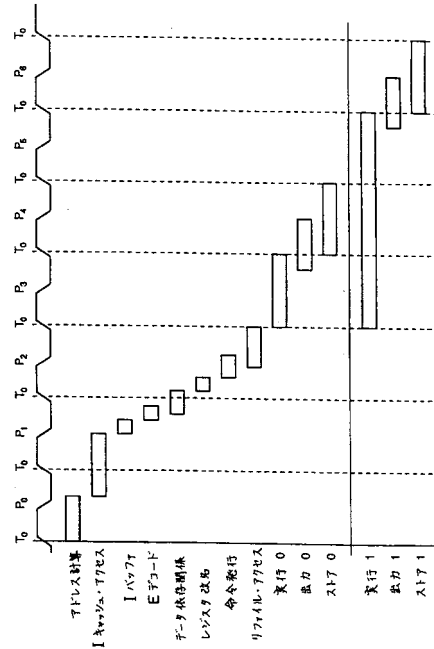
【図 12】



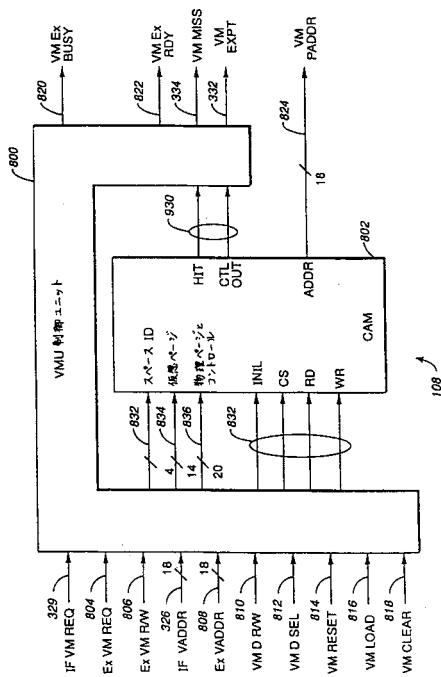
【 図 1 3 】



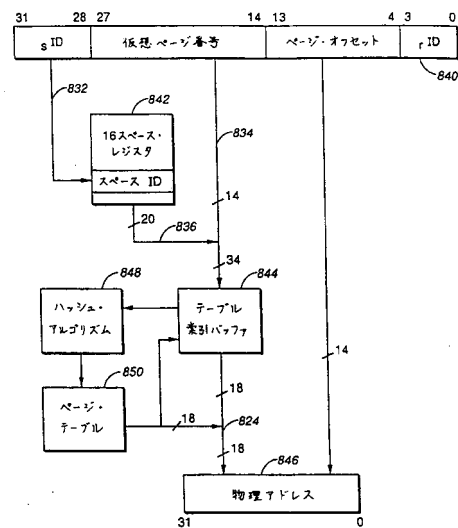
【図 14】



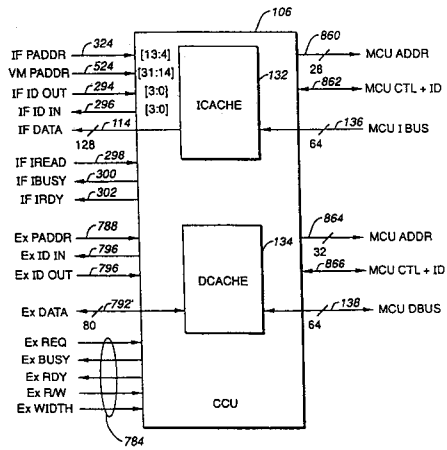
【图 1 5】



【图 16】



【図 17】



フロントページの続き

- (72)発明者 ミヤヤマ, 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, ティーリ
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 後藤 彰

- (56)参考文献 特開平3-42724 (JP, A)
特開昭64-36336 (JP, A)
特開昭58-178464 (JP, A)

- (58)調査した分野(Int.Cl.⁷, DB名)
G06F 9/38