

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3714992号

(P3714992)

(45) 発行日 平成17年11月9日(2005.11.9)

(24) 登録日 平成17年9月2日(2005.9.2)

(51) Int. Cl.⁷

G06F 9/38

F I

G06F 9/38 330A

請求項の数 27 (全 41 頁)

(21) 出願番号	特願平7-134012	(73) 特許権者	591016172
(22) 出願日	平成7年5月31日(1995.5.31)		アドバンスト・マイクロ・デバイス・
(65) 公開番号	特開平7-334362		インコーポレイテッド
(43) 公開日	平成7年12月22日(1995.12.22)		ADVANCED MICRO DEVI
審査請求日	平成14年4月11日(2002.4.11)		CES INCORPORATED
(31) 優先権主張番号	252303		アメリカ合衆国、94088-3453
(32) 優先日	平成6年6月1日(1994.6.1)		カリフォルニア州、サニペイル、ピー・
(33) 優先権主張国	米国 (US)		オウ・ボックス・3453、ワン・エイ・
			エム・ディ・プレイス、メイル・ストップ
			・68 (番地なし)
		(74) 代理人	100064746
			弁理士 深見 久郎
		(74) 代理人	100085132
			弁理士 森田 俊雄

最終頁に続く

(54) 【発明の名称】 複数の動作を同時に行なうためのプロセッサ、その中のスタック、およびスタック制御方法

(57) 【特許請求の範囲】

【請求項1】

複数の動作を同時に行なうためのプロセッサであって、前記動作は、浮動小数点計算命令と、浮動小数点スタック交換と、浮動小数点スタックをプッシュまたはポップする命令とを含む命令セットから選択され、前記プロセッサは、

(a) 前記浮動小数点スタック交換命令を含む前記命令セットにおいて、命令をデコードするための命令デコーダと、

(b) 前記命令デコーダに結合され、前記命令セットにおいて命令の推論的実行を判断するための分岐ユニットとを含み、前記分岐ユニットは、前記推論的実行に従って前記浮動小数点スタック交換命令を実行するためのものであり、前記プロセッサは、

(c) 前記命令デコーダに結合され、浮動小数点計算命令を実行するための浮動小数点機能ユニットと、

(d) 前記浮動小数点機能ユニットに結合され、前記浮動小数点機能ユニットから受取られる計算結果をストアするための浮動小数点スタックアレイと、

前記浮動小数点スタックアレイに結合され、アレイエレメントを特定するための浮動小数点スタックポインタと、

前記浮動小数点スタックポインタを前記浮動小数点スタックアレイに結合し、前記スタックポインタによってアドレス指定される浮動小数点スタックアレイエレメントを再順序づけするための浮動小数点スタックリマップアレイとを含む浮動小数点スタックとを含

10

20

む、プロセッサ。

【請求項 2】

前記浮動小数点スタックは、浮動小数点スタック交換命令が浮動小数点計算と同時に実行できるように、前記浮動小数点機能ユニットから独立して動作可能である、請求項 1 に記載のプロセッサ。

【請求項 3】

前記命令セットは分岐命令をさらに含み、前記浮動小数点スタックは、アレイエレメントを交換することによって浮動小数点スタック交換命令に応答するルックアヘッドリマップアレイと、

前記浮動小数点スタックアレイのエレメントを特定するように前記ルックアヘッドリマップアレイに結合され、前記浮動小数点スタックをポップまたはプッシュする命令に応答して前記ポインタを調節するルックアヘッドスタックポインタとを含み、

前記分岐ユニットは、前記ルックアヘッドリマップアレイおよび前記ルックアヘッドスタックポインタに結合され、前記分岐ユニットはさらに、

分岐命令に応答して前記ルックアヘッドリマップアレイをセーブするためのメモリと、

分岐が発生されるかどうかを予測するための分岐予測器と、

前記分岐予測器に結合され、分岐が予測されるかまたは誤予測されるかを判断する分岐比較器と、

前記メモリと前記ルックアヘッドリマップアレイとの間に結合され、誤予測された分岐に応答して、セーブされた値をアレイに送る第 1 の制御ラインと、

前記ルックアヘッドリマップアレイおよび前記ルックアヘッドスタックポインタをそれぞれ前記浮動小数点リマップアレイおよび前記浮動小数点スタックポインタに結合し、それぞれ浮動小数点の値をルックアヘッドの値で置き換えることによってスタック交換命令の実行に応答する第 2 の制御ラインとを含む、請求項 1 に記載のプロセッサ。

【請求項 4】

前記浮動小数点スタックは、

前記浮動小数点リマップアレイに結合され、浮動小数点スタックエレメントが空であるかといった状態を指定する浮動小数点スタックフル/エンプティアレイと、

前記ルックアヘッドリマップアレイに結合され、前記浮動小数点スタックフル/エンプティアレイのルックアヘッド状態をモニタするルックアヘッドスタックフル/エンプティアレイとをさらに含む、請求項 1 に記載のプロセッサ。

【請求項 5】

スタック交換命令と、スタックをプッシュまたはポップする命令と、スタックにアクセスする命令とを含む命令セットを行なうプロセッサにおいてスタックを制御するための方法であって、

命令デコーダにおいて命令をデコードしてその指示を判断するステップと、

前記指示がスタック交換命令であるとき、推論的実行を制御するための分岐ユニットにおいてスタックリマップアレイのエレメントの交換を指示するステップと、

前記指示がスタックプッシュであるとき、スタックポインタを前記指示デコーダにおける 1 つの方向に調節するステップと、

前記指示がスタックポップであるとき、前記スタックポインタを前記指示デコーダにおける別の方向に調節するステップと、

前記指示がスタックアクセスであるとき前記スタックポインタによって指定されかつ前記スタックリマップアレイによって再順序づけされたスタックアレイエレメントを用いるステップとを含む、方法。

【請求項 6】

スタック交換命令、スタックをプッシュまたはポップする命令、スタックにアクセスする命令、および条件つき分岐命令を含む命令セットを行なうプロセッサにおいてスタックを制御するための方法であって、

(a) 命令デコーダにおいて命令をデコードしてその指示を判断するステップと、

10

20

30

40

50

(b) スタック交換命令に回答して、分岐ユニットにおいてルックアヘッドスタックリマップアレイの要素の交換を指示し、推論的実行の制御をするためのステップと、
(c) スタックをプッシュまたはポップする命令に回答して、指示により命令デコードにおいてルックアヘッドスタックポインタを調節するステップと、
(d) スタックにアクセスする命令に回答して、前記ルックアヘッドスタックポインタによって指定されかつ前記ルックアヘッドスタックリマップアレイによって再順序づけされたスタックアレイ要素を用いるステップと、
(e) 分岐ユニットにおいて条件つき分岐命令に回答するステップとを含み、前記条件つき分岐命令に回答するステップは、
前記ルックアヘッドリマップアレイをセーブするステップと、
分岐が発生されるかどうかを予測するステップと、
分岐が正しく予測されたか誤予測されたかを判断するステップと、
分岐命令が誤予測されたときに前記ルックアヘッドリマップアレイをセーブされた値に復元するステップとを含み、前記方法は、
(f) 命令をそのプログラムの順序で、リオーダバッファ回収理論の指示により回収するステップをさらに含み、前記回収するステップは、
回収するスタック交換命令に回答してスタックリマップアレイを前記ルックアヘッドリマップアレイと置き換えるステップと、
回収するスタックをプッシュまたはポップする命令に回答してスタックポインタを調節するステップとを含む、方法。

10

20

【請求項 7】

スタックをプッシュおよびポップする命令に回答してルックアヘッドフル/エンブティアレイにおけるエントリを設定およびクリアするステップと、
スタックをプッシュまたはポップする命令の回収に回答してルックアヘッドフル/エンブティアレイにおけるエントリを設定およびクリアするステップとをさらに含む、請求項 6 に記載の方法。

【請求項 8】

(a) スタック実行エラーを検出するステップをさらに含み、前記ステップは、
前記ルックアヘッドフル/エンブティアレイのすべての要素がいっぱいであるかどうかを判断するステップと、
前記ルックアヘッドフル/エンブティアレイのすべての要素が空であるかどうかを判断するステップと、
前記ルックアヘッドフル/エンブティアレイのすべての要素が空であるとき、スタックをポップする命令に回答してスタックアンダフローエラーを検出するステップと、
前記ルックアヘッドフル/エンブティアレイのすべての要素がいっぱいであるとき、スタックをプッシュする命令に回答してスタックオーバフローエラーを検出するステップとを含み、
(b) 検出されたスタック実行エラーに回答して再同期化応答を開始するステップをさらに含む、請求項 7 に記載の方法。

30

【請求項 9】

例外状態を検出するステップと、
検出された例外状態に回答して、前記スタックリマップアレイを前記ルックアヘッドスタックリマップアレイにコピーし、かつ前記スタックポインタを前記ルックアヘッドスタックポインタにコピーするステップとをさらに含む、請求項 7 に記載の方法。

40

【請求項 10】

スタック交換命令とスタックをプッシュまたはポップする命令とを含む命令セットを行なうプロセッサにおけるスタックであって、前記プロセッサは、
前記スタック交換命令を含む前記命令セットにおいて、命令をデコードするための命令デコーダと、
前記命令デコーダに結合され、前記命令セットにおいて命令の推論的実行を判断するた

50

めの分岐ユニットとを含み、前記分岐ユニットは前記推論的実行に従って前記スタック交換命令を実行するためのものであり、前記スタックは、

前記命令デコーダと前記分岐ユニットとに結合されたスタックメモリアレイと、

前記スタックメモリアレイに結合され、スタックメモリアレイエレメントを指定するスタックポインタとを含み、前記スタックポインタは、スタックをプッシュまたはポップする命令にตอบสนองして調節され、前記スタックはさらに、

前記スタックポインタに結合されかつ前記スタックメモリアレイに結合されてスタックアレイエレメントを、前記スタックポインタによってアドレス指定されるように再順序づけするポインタのスタックリマップアレイを含み、前記ポインタはスタック交換命令にตอบสนองして交換される、スタック。

10

【請求項 11】

前記スタックリマップアレイに結合され、前記スタックリマップアレイを介して前記スタックポインタによってアドレス指定され、スタックアレイエレメントを、前記スタックポインタによってアドレス指定されるように再順序づけするスタックフル/エンブティアレイをさらに含み、前記スタックフル/エンブティアレイのエレメントはそれぞれ、スタックアレイエントリの加算または削除を表わす、スタックをプッシュおよびポップする命令にตอบสนองしてセットおよびクリアされる、請求項 10 に記載のスタック。

【請求項 12】

分岐命令、スタック交換命令、スタックをプッシュする命令、およびスタックをポップする命令を含む命令セットの複数の命令を同時にフェッチし、デコードし、実行し、かつ回収するプロセッサにおけるスタックであって、前記プロセッサは、条件つき分岐命令にตอบสนองして分岐が発生されるかどうかを予測する分岐予測器と、前記分岐予測器に結合され、その後条件つき分岐が予測されたかまたは誤予測されたかを判断する分岐テストとを有する分岐ユニットを含み、前記スタックは、

20

スタックメモリアレイと、

前記スタックメモリアレイに結合され、スタックメモリアレイエレメントを指定するためのスタックポインタとを含み、前記スタックポインタは、スタックをプッシュまたはポップする命令にตอบสนองして調節され、

前記スタックポインタに結合され、かつ前記スタックメモリアレイに結合されて前記スタックポインタによってアドレス指定されるスタックアレイエレメントを再順序づけするポインタのスタックリマップアレイをさらに含み、前記ポインタは、スタック交換命令にตอบสนองして交換され、

30

前記スタックポインタのルックアヘッド状態を指定するルックアヘッドスタックポインタをさらに含み、前記ルックアヘッドスタックポインタはスタックをプッシュまたはポップする命令にตอบสนองして調節され、

前記ルックアヘッドスタックポインタに結合され前記スタックポインタによってアドレス指定されるスタックアレイエレメントを再順序づけするためのルックアヘッドリマップアレイをさらに含み、前記ルックアヘッドリマップアレイのエレメントはスタック交換命令にตอบสนองして交換され、

前記分岐ユニットに結合され、スタック交換命令にตอบสนองして前記ルックアヘッドリマップアレイをセーブするためのメモリと、

40

前記分岐ユニットに結合され、スタックをプッシュまたはポップする命令にตอบสนองして前記ルックアヘッドスタックポインタをセーブするためのメモリと、

前記分岐テストに結合され、誤予測された分岐にตอบสนองして前記ルックアヘッドリマップアレイおよび前記ルックアヘッドスタックポインタをセーブされた値に復元するための手段と、

スタック交換命令の回収にตอบสนองして、前記スタックリマップアレイを前記ルックアヘッドリマップアレイで置き換えるための手段とをさらに含む、スタック。

【請求項 13】

前記ルックアヘッドリマップアレイに結合され、スタックアレイエレメントを再順序づけ

50

するためにルックアヘッドスタックポイントによってルックアヘッドリマップアレイを介してアドレス指定されるルックアヘッドフル/エンブティアレイをさらに含み、前記ルックアヘッドフル/エンブティアレイの要素は、それぞれスタックアレイエントリの加算または削除を表わすスタックをプッシュまたはポップする命令に回答して設定およびクリアされる、請求項 1 2 に記載のスタック。

【請求項 1 4】

前記ルックアヘッドスタックポイントと、ルックアヘッドリマップアレイと、ルックアヘッドフル/エンブティアレイとに結合される命令デコーダをさらに含むプロセッサにおけるスタックであって、前記スタックは、前記命令デコーダに結合され、スタック交換命令に回答して前記ルックアヘッドリマップアレイのポイントを交換するための手段と、前記命令デコーダに結合され、スタックをプッシュする命令およびスタックをポップする命令に回答して、前記ルックアヘッドスタックポイントを調整し、かつそれぞれ前記ルックアヘッドフル/エンブティアレイにおいてエントリを加えるおよび取除くための手段とを含む、請求項 1 3 に記載のスタック。

10

【請求項 1 5】

前記命令デコーダに結合され、スタック実行エラーを検出するための手段と、前記検出手段のスタックエラー検出に回答して、再同期化応答を開始するための手段とをさらに含む、請求項 1 4 に記載のスタック。

【請求項 1 6】

前記スタックポイントおよび前記リマップアレイに結合されるリオーダバッファと、前記スタックメモリアレイに結合されるレジスタファイルとをさらに含むプロセッサにおけるスタックであって、前記スタックは、前記リオーダバッファに結合され、かつ前記スタックリマップアレイを介して前記スタックポイントに結合されるスタックフル/エンブティアレイをさらに含む、請求項 1 3 に記載のスタック。

20

【請求項 1 7】

前記プロセッサは例外状態検出器を含み、前記命令デコーダに結合され、例外状態に回答して、前記ルックアヘッドリマップアレイを前記スタックリマップアレイで置き換え、前記ルックアヘッドスタックフル/エンブティアレイを前記スタックフル/エンブティアレイで置き換え、前記ルックアヘッドスタックポイントを前記スタックポイントで置き換えるための手段をさらに含む、請求項 1 6 に記載のスタック。

30

【請求項 1 8】

前記プロセッサは浮動小数点機能ユニットをさらに含み、前記浮動小数点機能ユニットは前記スタックメモリアレイ内に含まれるデータに基づいて動作する、請求項 1 3 に記載のスタック。

【請求項 1 9】

分岐命令と、スタック要素交換命令と、スタックをプッシュおよびポップする命令とを含む命令セットからの複数の命令を同時に実行するプロセッサであって、

前記スタックは、

40

(a) スタックメモリアレイを含むレジスタファイルと、

(b) 前記レジスタファイルに結合されるリオーダバッファとを含み、前記リオーダバッファは、

スタックメモリアレイ要素を特定するたのスタックポイントと、

前記スタックポイントを前記スタックメモリアレイに結合し、前記スタックポイントによってアドレス指定されるスタックメモリアレイ要素を再順序づけするためのスタックリマップアレイとを含み、

(c) 前記レジスタファイルおよび前記リオーダバッファに結合される命令デコーダをさらに含み、前記命令デコーダは、

ルックアヘッドスタックポイントと、

50

前記ルックアヘッドスタックポインタに結合されるルックアヘッドリマップアレイと、
前記ルックアヘッドスタックポインタおよび前記ルックアヘッドリマップアレイに結合されるデコーダ回路と、

スタックエレメント交換命令のデコードにตอบสนองして、ルックアヘッドリマップアレイエレメントを交換するための手段と、

スタックをプッシュまたはポップする命令のデコードにตอบสนองして、前記ルックアヘッドスタックポインタを調節するための手段とを含み、

(d) 前記命令デコーダに結合され、前記ルックアヘッドリマップアレイおよび前記ルックアヘッドスタックポインタのコピーを受取るための分岐ユニットをさらに含み、前記分岐ユニットは、

分岐命令にตอบสนองして、前記ルックアヘッドリマップアレイおよび前記ルックアヘッドスタックポインタのコピーをその中にセーブするためのメモリと、

条件つき分岐命令にตอบสนองして、分岐が発生されるかどうかを予測するための分岐予測器と、

前記分岐予測器に結合され、前記条件つき分岐が予測されたか誤予測されたかをその後に判断するための分岐テストと、

誤予測された分岐にตอบสนองして、前記ルックアヘッドリマップアレイを前記セーブされたルックアヘッドリマップアレイで置き換え、かつ前記ルックアヘッドスタックポインタを前記セーブされたルックアヘッドスタックポインタで置き換えるための手段と、

スタックエレメント交換命令の実行の終了にตอบสนองして、前記スタックリマップアレイを前記セーブされたルックアヘッドリマップアレイで置き換えるための手段とを含む、プロセッサ。

【請求項 20】

前記命令デコーダに結合され、スタック実行エラーを検出するための手段と、

前記検出手段のスタックエラー検出にตอบสนองして、再同期化応答を開始するための手段とをさらに含む、請求項 19 に記載のプロセッサ。

【請求項 21】

例外状態を検出するための手段と、

前記命令デコーダに結合され、例外状態にตอบสนองして、前記ルックアヘッドリマップアレイを前記スタックリマップアレイで置き換え、前記ルックアヘッドスタックフル/エンプティアレイを前記スタックフル/エンプティアレイで置き換え、前記ルックアヘッドスタックポインタを前記スタックポインタで置き換えるための手段とをさらに含む、請求項 19 に記載のプロセッサ。

【請求項 22】

浮動小数点機能ユニットをさらに含み、前記浮動小数点機能ユニットは、前記スタックメモリアレイに含まれるデータに基づいて動作する、請求項 19 に記載のプロセッサ。

【請求項 23】

スタックプッシュ、スタックポップ、および分岐命令を含む命令セットからの命令を行なうプロセッサにおけるスタックであって、前記プロセッサは、命令をディスパッチするデコーダと、前記デコーダに結合され、条件つき分岐命令を予測しかつ誤予測を検出する分岐ユニットと、前記デコーダに結合され、予測に応じて命令を推論的に実行する複数の機能ユニットと、前記デコーダおよび前記機能ユニットに結合され、回収された命令の結果を受取るメモリとを含み、前記スタックは、

(a) リオーダーバッファに結合された複数の記憶エレメントと、

(b) 前記記憶エレメントに結合され、さらに前記リオーダーバッファに結合されて前記複数の記憶エレメントのうちの記憶エレメントにアドレス指定するためのポインタと、

(c) 前記デコーダに結合され、前記機能ユニットが命令を推論的に実行しているときに前記複数の記憶エレメントのうちの記憶エレメントをアドレス指定するためのルックアヘッドポインタと、

(d) 前記記憶エレメントと、前記ポインタと、前記ルックアヘッドポインタと、前記

10

20

30

40

50

デコーダとに結合されるスタックコントローラとを含み、前記スタックコントローラは、スタックプッシュおよびスタックポップのディスパッチにตอบสนองして、前記ルックアヘッドポインタを更新するための手段と、スタックプッシュおよびスタックポップの回収にตอบสนองして、前記ルックアヘッドポインタを前記ポインタにコピーするための手段と、条件つき分岐命令のディスパッチにตอบสนองして、前記ルックアヘッドポインタをセーブするための手段と、分岐の誤予測にตอบสนองして、前記ルックアヘッドポインタを前記セーブされた値に復元するための手段とを含む、スタック。

【請求項 2 4】

前記ポインタに結合され、前記複数の記憶エレメントのうちの記憶エレメントに対応するエレメントを有し、前記記憶エレメントがいっぱいであるか空であることを示すためのフル/エンptyアレイをさらに含み、前記スタックコントローラは、スタックプッシュおよびスタックポップのディスパッチにตอบสนองして前記フル/エンptyアレイを調節するための手段と、スタックプッシュおよびスタックポップの回収にตอบสนองして、前記フル/エンptyアレイを調節するための手段と、条件つき分岐命令予測にตอบสนองして、前記フル/エンptyアレイをセーブするための手段と、分岐の誤予測にตอบสนองして、前記フル/エンptyアレイを前記セーブされた値に復元するための手段とをさらに含む、請求項 2 3 に記載のスタック。

【請求項 2 5】

スタックプッシュ、スタックポップ、および分岐命令を含む命令セットからの命令を行なうプロセッサにおけるスタックであって、前記プロセッサは、命令をディスパッチするデコーダと、前記デコーダに結合され、条件つき分岐命令を予測しかつ誤予測を検出する分岐ユニットと、前記デコーダに結合され、予測に応じて命令を推論的に実行するための複数の機能ユニットと、前記デコーダおよび前記機能ユニットに結合され、回収された命令の結果を受取るためのメモリとを含み、前記スタックは、

(a) 前記リオーダーバッファに結合された複数の記憶エレメントと、
 (b) 前記記憶エレメントと前記リオーダーバッファとに結合され、前記複数の記憶エレメントのうちの記憶エレメントを再順序づけするためのリマップアレイと、
 (c) 前記リマップアレイを介して前記記憶エレメントに結合され、前記複数の記憶エレメントのうちの記憶エレメントをアドレス指定するためのポインタと、
 (d) 前記デコーダに結合され、前記機能ユニットが推論的に命令を実行しているときに前記複数の記憶エレメントのうちの記憶エレメントをアドレス指定するルックアヘッドポインタと、

(e) 前記記憶エレメントと前記デコーダとに結合され、前記機能ユニットがディスパッチされたスタック交換命令に関して命令を推論的に実行しているときに前記複数の記憶エレメントのうちの記憶エレメントを再順序づけするためのルックアヘッドリマップアレイと、

(f) 前記記憶エレメント、前記リマップアレイ、前記ポインタ、前記ルックアヘッドポインタ、前記推論的な分岐ルックアヘッドポインタ、および前記ルックアヘッドリマップアレイに接続されるスタックコントローラとを含み、前記スタックコントローラは、前記デコーダに結合され、スタックプッシュおよびスタックポップのディスパッチにตอบสนองして、前記ルックアヘッドポインタを更新するための手段と、前記リオーダーバッファに結合され、スタックプッシュおよびスタックポップの回収にตอบสนองして、前記ルックアヘッドポインタを前記ポインタにコピーするための手段と、前記デコーダに結合され、条件つき分岐命令のディスパッチにตอบสนองして、前記ルックアヘッドポインタをセーブするための手段と、前記分岐ユニットに結合され、分岐の誤予測にตอบสนองして、前記ルックアヘッドポインタを

10

20

30

40

50

前記セーブされた値に復元するための手段と、
 前記分岐ユニットに結合され、スタック交換命令のディスパッチにตอบสนองして、前記ルックアヘッドリマップアレイの要素を交換するための手段と、
 前記分岐ユニットに結合され、条件つき分岐命令予測のディスパッチにตอบสนองして、前記ルックアヘッドリマップアレイを前記推論的分岐ルックアヘッドリマップアレイにセーブするための手段と、

前記分岐ユニットに結合され、分岐の誤予測にตอบสนองして、前記ルックアヘッドリマップアレイを前記セーブされた値に復元するための手段と、

前記リオーダバッファに結合され、スタック交換命令の回収にตอบสนองして、前記ルックアヘッドリマップアレイを前記リマップアレイにコピーするための手段とを含む、スタック。

10

【請求項 26】

プロセッサのスタックアレイを動作させるための方法であって、

(a) スタックポインタを用いて前記スタックアレイの要素を選択するステップと、

(b) リマップアレイを用いて、前記スタックポインタによって指定されるスタックアレイ要素を再順序づけするステップと、

(c) 前記スタックのルックアヘッド状態を、前記スタックポインタに対応するルックアヘッドスタックポインタメモリと、前記リマップアレイに対応するルックアヘッドリマップアレイとにストアするステップと、

(d) スタック要素交換命令にตอบสนองして、前記分岐ユニットの推論的状态の指示により、前記ルックアヘッドリマップアレイの要素を交換するステップと、

20

(e) スタックをプッシュまたはポップする命令にตอบสนองして前記ルックアヘッドスタックポインタを調節するステップと、

(f) 分岐命令、スタック交換命令、およびスタックをプッシュまたはポップする命令を含む命令を分岐ユニットにおいて推論的に実行するステップと、

(g) 推論的に実行された分岐命令に対応して、前記ルックアヘッドスタックポインタおよび前記ルックアヘッドリマップアレイをセーブするステップと、

(h) 推論的に実行された命令を回収するステップとを含み、前記回収ステップは、回収するスタック交換命令にตอบสนองして、前記リマップアレイを前記ルックアヘッドリマップアレイで置き換えるステップと、

30

回収するスタックプッシュまたはポップ命令にตอบสนองして前記スタックポインタを調節するステップとを含み、

(i) 誤予測された分岐にตอบสนองして、前記ルックアヘッドスタックポインタおよび前記ルックアヘッドリマップアレイをセーブされた値に復元するステップをさらに含む、方法。

【請求項 27】

前記ポインタに結合されるフル/エンブティアレイを用いて前記スタックアレイ要素が空であるかいっぱいであるかをモニタするステップをさらに含み、前記フル/エンブティアレイは、前記スタックアレイの要素に対応する要素を有し、スタックコントローラは、

40

スタックをプッシュおよびポップする命令のディスパッチにตอบสนองして指示によりそれぞれエントリを加えるおよび取除くことによって前記フル/エンブティアレイのルックアヘッドコピーを調節するステップと、

スタックをプッシュおよびポップする命令の回収にตอบสนองして、必要に応じて、それぞれエントリを加えるおよび取除くことによって前記フル/エンブティアレイの永久的なコピーを調節するステップと、

条件つき分岐命令予測にตอบสนองして、前記ルックアヘッドフル/エンブティアレイをセーブするステップと、

分岐の誤予測の検出にตอบสนองして前記ルックアヘッドフル/エンブティアレイをセーブされた推論的分岐ルックアヘッドフル/エンブティアレイに復元するステップとを実行する、

50

請求項 26 に記載の方法。

【発明の詳細な説明】

【0001】

【発明の分野】

本発明はプロセッサスタックに関し、より特定的には、命令の推論的実行に関わるプロセッサのためのスタックおよびスタック動作方法に関する。

【0002】

【関連技術の説明】

プロセッサは一般に命令セットのうちの1つの命令をいくつかのステップで処理する。初期の技術によるプロセッサは、これらのステップをシリアルに行なっていた。技術の進歩により、多くの命令の異なるステップを同時に行なうスカラプロセッサと呼ばれるパイプライン方式のプロセッサが開発された。「スーパースカラ」プロセッサは、スカラ命令の同時実行をサポートすることにより性能をさらに向上する。スーパースカラプロセッサでは、データまたは資源が利用可能でないため発行された命令を実行することができない従属性の条件や命令の競合が生じる。たとえば、発行された命令は、その入力オペランドがまだ実行を終了していない他の命令によって計算されるデータに依存する場合は、実行することができない。

【0003】

スーパースカラプロセッサの性能は、命令をすぐ実行する能力にかかわらず命令をデコードし続けることにより向上される。命令のデコードと命令の実行とを切離すためには、命令を実行する機能ユニットと呼ばれる回路によって用いられるディスパッチされた命令情報をストアするためのルックアヘッドバッファと呼ばれるバッファが必要である。

【0004】

このバッファによっても、散在する分岐命令を含む命令シーケンスに対するプロセッサの性能が向上される。分岐に従う命令は通常、状態がわかるまで待たなければならない、その後になって初めて実行を進めることができるため、分岐命令はプロセッサの性能を損なう。スーパースカラプロセッサでは、「推論的に」命令を実行することによって、分岐条件の結果を予測しかつその予測に従ってその次の命令に進むことに関わる分岐能力が向上される。バッファは、プロセッサの推論状態を維持するように実現される。予測が間違っている場合には、間違っただけで予測された分岐に従う命令によって生じた結果は放棄される。分岐を間違っただけで予測した場合に迅速に回復し、適切な命令シーケンスを再び開始することにより、スーパースカラプロセッサの性能はかなり向上される。回復方法により、不適切に行なわれた命令による影響が取消される。再開の手順により、正しい命令シーケンスが再び確立される。

【0005】

『スーパースカラプロセッサ設計 (Superscalar Processor Design)』, Englewood Cliffs, N.J., Prentice Hall, 1991, p.92-97においてマイク・ジョンソン (Mike Johnson) によって教示される1つの回復および再開方法では、リオーダバッファおよびレジスタファイルが用いられる。レジスタファイルは、回収された動作、すなわちもう推論的でない動作によって発生したレジスタ値を保持する。リオーダバッファは、動作の推論的結果、すなわち予測されてはいるが確証はされていない分岐に従ったシーケンスで実行される動作の結果を保持する。リオーダバッファは、先入れ先出しの待ち行列として動作する。命令がデコードされると、リオーダバッファの末尾でエントリが割当てられる。エントリは、命令および命令の結果に関する情報が利用可能になればそれを保持する。その結果値を受取ったエントリがリオーダバッファの先頭に達すると、その結果をレジスタファイルに書込むことによりその動作は回収される。リオーダバッファは、誤予測された分岐に従う命令によって生じたレジスタ値を放棄するために、分岐の誤予測の後の回復の間にプロセッサによって用いられる。リオーダバッファは誤予測された分岐に従うレジスタを復元するが、他のプロセッサのレジスタも復元される必要があるかもしれない。たとえば、データを管理するためにスタックを用いるプロセッサでは、スタックは復元を必要とする。

10

20

30

40

50

スタックの復元には、アレイエレメントおよびポインタを含むすべてのスタックエレメントの回復が必要である。

【 0 0 0 6 】

スタックの一例は、カリフォルニア州サンタクララ (Santa Clara) のインテル・コーポレーション (Intel Corporation) から入手可能な商標ペンチウム (Pentium) マイクロプロセッサの浮動小数点ユニット (F P U) レジスタスタックである。F P U レジスタスタックは、拡張された実データをストアする、8つのマルチビット数値レジスタのアレイである。F P U の命令により、スタックの頂部 (T O S) に関するデータレジスタがアドレス指定される。商標ペンチウムマイクロプロセッサにおける浮動小数点交換 (F X C H) 命令により、スタックの頂部の内容が、特定のスタックエレメント、たとえばT O S に関するスタックの終わりから2番目の位置にあるデフォルトエレメントの内容と交換する。ペンチウム (商標) 浮動小数点命令は一般に、スタックのトップ位置に配置されるべき1つのソースオペランドを必要とし、かつF P U 命令の結果がT O S に残されることがよくあるため、F X C H 命令は有用である。ほとんどのF P U 命令がT O S へのアクセスを必要とするため、F X C H 命令を用いてスタック内のデータ位置を操作することが望ましい。

10

【 0 0 0 7 】

スタックのトップ位置はT O S ポインタによって識別される。スタックエントリは、いくつかの浮動小数点命令とデータのロードおよびストア命令とを実行することによりプッシュされかつポップされる。これらの命令がプロセッサのプログラミングに依存するため、浮動小数点のオーバーフローおよびアンダフローが生じ、これらはトラップされなければならない。これにより例外条件が発生する。誤予測された分岐のような例外条件には、プロセッサの推論的状态の復元が必要である。

20

【 0 0 0 8 】

このF X C H 命令により生じる1つの結果は、スタックエレメントの順序が変わりやすくなってしまうことであり、これにより誤予測された分岐または例外の後に生じるスタックの復元が複雑になってしまう。

【 0 0 0 9 】

スーパースカラプロセッサでは、誤予測された分岐および例外が生じても、効果的な回復および再開の手順を所望のように行なう。スタック、およびスタックの状態を簡単にかつ迅速に復元するためのスタックの動作方法が求められている。

30

【 0 0 1 0 】

【 発明の概要 】

本発明の一実施例は、浮動小数点計算命令、浮動小数点スタック交換、浮動小数点スタックをプッシュまたはポップする命令等の複数個の動作を同時に行なうためのプロセッサである。このプロセッサは、計算を実行するための浮動小数点機能ユニットと、浮動小数点機能ユニットから得られた計算結果を扱うための浮動小数点スタックとを含む。スタックは、浮動小数点機能ユニットから得られた計算結果をストアするための浮動小数点スタックアレイと、浮動小数点スタックアレイのエレメントを特定するための浮動小数点スタックポインタと、スタックポインタによってアドレス指定された浮動小数点スタックアレイエレメントを順序づけるための浮動小数点スタックリマップアレイとを含む。

40

【 0 0 1 1 】

本発明の別の実施例は、スタックを制御するための方法である。スタックは、スタック交換命令およびスタックをプッシュまたはポップする命令を含む命令を実行するためのプロセッサにおいてスタックメモリアレイおよびスタックポインタを含む。この方法は、スタックポインタをスタックのトップ位置のメモリアレイに設定し、かつシーケンシャルな順序でスタックメモリアレイエレメントをアドレス指定するようにスタックリマップアレイを設定することによってスタックを初期化するステップを含む。この方法は、実行するための命令をデコードしかつディスパッチするステップと、スタック交換命令に回答してスタックリマップアレイのエレメントを交換するステップと、スタックをプッシュまたはポ

50

ップする命令にตอบสนองしてスタックポインタを調節するステップとをさらに含む。

【0012】

本発明のさらに他の実施例は、プロセッサスタックを制御するための方法である。スタックは、メモリアレイおよびスタックポインタを含む。この方法は、スタックを初期化するステップを含み、この初期化ステップは、スタックポインタおよびルックアヘッドスタックポインタをスタックのトップ位置のメモリアレイに設定し、シーケンシャルな順序でスタックメモリアレイエレメントをアドレス指定するようにスタックリマップアレイおよびルックアヘッドリマップアレイを設定するサブステップを含む。この方法は、実行するための命令をデコードしかつディスパッチするステップと、ディスパッチされたスタック交換命令にตอบสนองしてルックアヘッドリマップアレイのエレメントを交換するステップと、ディスパッチされたスタックをプッシュまたはポップする命令にตอบสนองしてルックアヘッドスタックポインタを調節するステップとをさらに含む。ディスパッチされた分岐命令にตอบสนองして、この方法は、ルックアヘッドリマップアレイをセーブするステップと、分岐が発生されるかどうかを予測するステップと、分岐が正しく予測されたかどうかを判断するステップと、分岐命令が誤って予測されたときにルックアヘッドリマップアレイをセーブされた値に復元するステップとを含む。この方法は、命令をそのプログラムの順序で回収するステップをさらに含み、この回収ステップは、回収するスタック交換命令にตอบสนองしてスタックリマップアレイをルックアヘッドリマップアレイで置き換え、回収するスタックをプッシュまたはポップする命令にตอบสนองしてスタックポインタを調節するサブステップを含む。

10

20

【0013】

本発明の種々の実施例は、プロセッサが誤予測された分岐または例外に遭遇した場合に単純でかつ迅速な回復および再始動の手順を達成するデータスタックを動作させるための方法および装置を含む。

【0014】

本発明の特定の応用は、単純でかつ迅速な回復および再始動の手順を達成する浮動小数点データスタックを動作させるための方法および装置である。本発明により、浮動小数点演算命令とパラレルに浮動小数点交換命令を実行することができるという有利な能力が得られる。

【0015】

添付の図面を参照して以下に示す詳細な説明を読めば本発明がよりよく理解され、本発明の利点、目的および特徴がより明らかになるであろう。図中、同一の参照番号は同一のエレメントを示す。

30

【0016】

【好ましい実施例の詳細な説明】

図2および図3は、種々の機能ブロックの間でアドレス、データおよび制御の転送のやり取りを行なう内部アドレスおよびデータバス111を含むスーパースカラプロセッサ110と、外部メモリ114とを示している。命令キャッシュ116は、CISC命令を解析しかつプリデコードする。バイトキュー135は、プリデコードされた命令を命令デコーダ118に転送し、これはCISC命令をそれぞれRISCのような動作(「ROP」)のための命令のシーケンスにマップする。

40

【0017】

適切な命令キャッシュ116は、1993年10月29日出願の米国特許出願連続番号第08/145,905号により詳細に記載されている(デイビッド・ビー・ウィット(David B. Witt)およびマイケル・ディ・ゴダード(Michael D. Goddard)「可変バイト長命令に特に適切なプリデコード命令キャッシュおよびその方法(“Pre-Decode Instruction Cache and Method Therefor Particularly Suitable for Variable Byte-Length Instructions”)」; 1994年10月25日、日本出願第260701号の「可変バイト長命令フォーマットを有するタイプのプロセッサのための命令キャッシュ」)。適切なバイトキュー135は、1993年10月29日出願の米国特許出願連続番号第08/145

50

、902号に詳細に記載されている(デイビッド・ビー・ウィット(David B. Witt)「可変バイト長命令に特に適切な推論的命題キューおよびその方法(“Speculative Instruction Queue and Method Therefor Particularly Suitable for Variable Byte-Length Instructions”)」; 1994年10月25日、日本出願第260700号の「可変バイト長命令フォーマットを有するタイプのプロセッサのための推論的命題キュー」)。適切な命令デコーダ118は、1993年10月29日出願の米国特許出願連続番号第08/146,383号により詳細に記載されている(デイビッド・ビー・ウィット(David B. Witt)およびマイケル・ディ・ゴダード(Michael D. Goddard)「スーパースカラ命令デコーダ(Superscalar Instruction Dcode)」; 1994年10月26日、日本出願第262437号の「スーパースカラ命令デコード/発行装置」)。これらの出願全体を引用をここに援用する。 10

命令デコーダ118は、種々のバスを介してプロセッサ110内の機能ブロックにROPをディスパッチする。プロセッサ110は、マイクロプロセッサのサイクルにおいて、4個以下のROPの発行、5個以下のROP結果のやり取り、および16個以下の推論的に実行されたROPのキューへの登録をサポートする。AおよびBソースオペランドと宛先レジスタとに対する4組以下のポインタが、命令デコーダ118によって、それぞれのAオペランドポインタ136、Bオペランドポインタ137、および宛先レジスタポインタ143を介してレジスタファイル124およびリオーダバッファ126に与えられる。レジスタファイル124およびリオーダバッファ126により、4対のAオペランドバス130およびBオペランドバス131上の種々の機能ユニットに適切なソースオペランドAおよびBが与えられる。4対のAオペランドタグバス148およびBオペランドタグバス149を含むオペランドタグバスは、Aオペランドバス130およびBオペランドバス131に関連する。データをオペランドバス上に配置するのに利用できない場合、利用可能になったときにデータを受取るためのリオーダバッファ126におけるエントリを識別するタグは、対応するオペランドタグバス上にロードされる。オペランドバスおよびタグバスは、4つのROPディスパッチ位置に対応する。命令デコーダは、リオーダバッファ126と協働して、ROPが実行された後に機能ユニットから結果を受取るために、リオーダバッファ126におけるエントリを識別するための4つの宛先タグバス140を特定する。機能ユニットは、ROPを実行し、宛先タグを5本の結果タグバス139のうちの1つにコピーし、結果が利用可能である場合にはその結果を5本の結果バス132のうちの対応する結果バスに配置する。結果タグバス139における対応するタグが結果を待っているROPのオペランドタグと一致すれば、機能ユニットは結果バス132上の結果にアクセスする。 20

【0018】

命令デコーダ118は、4本の操作コード/タイプバス150を介してAおよびBソースオペランド情報に付随する操作コード情報をディスパッチする。操作コード情報は、機能ユニットのうちの適切な1つを選択するタイプフィールドと、RISC操作コードを識別する操作コードフィールドとを含む。 30

【0019】

プロセッサ110は、分岐ユニット120、整数機能ユニット121、浮動小数点機能ユニット122、ロード/ストア機能ユニット180等のいくつかの機能ユニットを含む。整数機能ユニット121は一般的な意味で与えられたものであって、種々のタイプの演算論理ユニットまたはシフトユニットを表わす。分岐ユニット120は、分岐がある場合に適切な命令フェッチ速度を可能にする分岐予測機能を果たし、複数の命令が発行された場合の性能を達成するために必要とされる。分岐ユニット120および命令デコーダ118を含む適切な分岐予測システムは、ジョンソン(Johnson)による『スーパースカラマイクロプロセッサ設計(“Superscalar Microprocessor Design”)』, Prentice Hall, 1990、および米国特許番号第5,136,697号(ウィリアム・エム・ジョンソン(William M. Johnson)「キャッシュの命令ブロックの各々でストアされるフェッチ情報を用いて、正しく予測された分岐命令の後の実行に関して遅延を低減するためのシステム(“Sy 40

10

20

30

40

50

stem for Reducing Delay for Execution Subsequent to Correctly Predicted Branch Instruction Using Fetch Information Stored with each Block of Instructions in Cache”)」)により詳細に示されており、これは引用によりここに援用される。プロセッサ 110は、複雑になりすぎないようにするために単純な1組の機能ユニットを有するように示されている。必要に応じて、整数ユニットおよび浮動小数点ユニットの他の組合せを実現することも可能である。

【0020】

レジスタファイル124は、中間の計算結果を保持するためのマップされたCISC整数レジスタ、浮動小数点レジスタ、一次レジスタを含む物理記憶メモリである。レジスタファイル124は、4個以下の同時にディスパッチされたROPの各々に関してAオペランドポインタ136およびBオペランドポインタ137の2個以下のレジスタポインタによってアドレス指定され、選択されたエントリの値を8個の読取ポートを介してAオペランドバス130およびBオペランドバス131上に与える。整数はレジスタファイル124の32ビットレジスタにストアされ、浮動小数点の数はレジスタファイル124の82ビットレジスタにストアされる。レジスタファイル124は、回収結果として既知であるプロセスにおいて、リオーダバッファ126から4本のライトバックバス134を介して実行された動作および非推論的動作の結果を受取る。

【0021】

リオーダバッファ126は、推論的に実行されたROPの相対的な順序を追跡するための環状FIFOである。記憶位置は、先頭キューポインタおよび末尾キューポインタを用いて、結果をレジスタファイル124に回収しかつ機能ユニットから結果を受取るように動的に割当てられる。命令がデコードされると、そのROPには、利用可能になった場合には結果値と、結果が書込まれるべきレジスタファイル124の宛先レジスタの番号とを含むROP情報をストアするためのリオーダバッファ126における位置が割当てられる。従属性を持たないROPに関しては、Aオペランドバス130およびBオペランドバス131は、レジスタファイル124から駆動される。浮動小数点データは、従属性を持たない浮動小数点ROPに関してAオペランド、Bオペランドおよび宛先レジスタが整数ROPの態様で直接アドレス指定されるのではなくスタックポインタおよびリマップレジスタによって指定されるように、スタックを用いてアクセスされる。スタックポインタおよびリマップレジスタは組合せられて、レジスタファイル124の浮動小数点レジスタを指す。しかしながら、ROPが従属性を持ち、そこにストアされていると考えられる値を得るために名前が変えられた宛先レジスタを参照すると、エントリがリオーダバッファ126内でアクセスされる。そこで結果が利用可能であれば、この結果はオペランドバス上に置かれる。結果が利用不可能であれば、リオーダバッファエントリを識別するタグがAオペランドタグバス148およびBオペランドタグバス149のうち的一方の上に与えられる。結果またはタグは、それぞれオペランドバス130、131またはオペランドタグバス148、149を介して機能ユニットに与えられる。浮動小数点ROPに関しては、データ依存性オペランドがリオーダバッファ124からアクセスされるか、またはスタックポインタおよびリマップレジスタに従ってタグがつけられる。

【0022】

機能ユニット120、121、122、180において実行が終了し結果が得られると、それらの結果およびそれぞれの結果タグは5本のバス幅結果バス132および結果タグバス139を介してリオーダバッファ126と機能ユニットの保存局に与えられる。5本の結果バス、結果タグおよび状態バスのうち、4本は、整数および浮動小数点の結果をリオーダバッファに送るための汎用バスである。送られた結果以外の情報を機能ユニットのうちのいくつかからリオーダバッファに送るために、付加的な第5の結果バス、結果タグおよび状態バスが用いられる。たとえば、分岐ユニット120による分岐動作から生じる状態情報は、この付加的なバスに置かれる。特定の機能ユニットは、5本の結果バス132および対応する結果タグバス139のサブセットにのみ相互接続し得る。

【0023】

レジスタファイル、リオーダバッファ、およびバスを含む適切なRISCコアは、1993年10月29日出願の米国特許出願連続番号第08/146,382号(デイビッド・ビィ・ウィット(David B. Witt)およびウィリアム・エム・ジョンソン(William M. Johnson)「高性能スーパースカラマイクロプロセッサ(High Performance Superscalar Microprocessor)」; 1994年10月27日、日本出願第263317号の「スーパースカラマイクロプロセッサ」)に記載されており、これを引用によりここに援用する。

【0024】

図4は、3つのパイプラインを用いて算術計算を行なう浮動小数点ユニット122の概略ブロック図である。第1のパイプラインは、2つの加算器段242、243と正規化シフト段253とを含む加算/減算パイプラインである。第2のパイプラインは、2つの乗算段244、245を含む乗算パイプラインである。第3のパイプラインは、検出ブロック252を含む。浮動小数点機能ユニット122はまた、共有浮動小数点ラウンダ247とFPU結果ドライバ251とを含む。浮動小数点保存局241は、操作コード/タイプバス150、Aオペランドバス130、Bオペランドバス131、結果バス132、結果タグバス139、Aオペランドタグバス148、Bオペランドタグバス149、および宛先タグバス140からの入力を受取るように接続される。保存局241は、2つのエントリを保持し、これらのエントリの各々は82ビットAオペランドおよび82ビットBオペランドのための記憶装置と、宛先結果タグと、8ビット操作コードと、4ビットAオペランドタグと、4ビットBオペランドタグと、浮動小数点スタックのオーバフローおよびアンダフローの状態を示すための状態ビットとを含む。保存局241は、クロックサイクルごとに、2つのROPの形の1つの浮動小数点動作を受入れることができる。保存局241は、その各々が82ビットオペランドと3つの浮動小数点計算制御ビットとを含む85ビット浮動小数点Aオペランドバス254および85ビット浮動小数点Bオペランドバス255を駆動する。

【0025】

検出252は、浮動小数点ユニット122への入力が規定された無効性のある条件を満たす場合例外信号を発生する。浮動小数点スタックオーバフローまたはアンダフロー信号が設定されるか、除算動作において分母オペランドが0であるか、またはソースオペランドの値が命令により発生された結果が0またはにされるような値を有する場合に、無効状態が生じる。浮動小数点機能ユニット122への入力のために例外が発生されると、ユニットが動作の残りの段をキャンセルし、リオーダバッファ126がプロセッサ110中にわたって例外応答を開始するように結果バス132上に例外信号を配置する。

【0026】

浮動小数点ラウンダ247は、浮動小数点ROPの実行により生じる例外を検出する。これらの例外は、浮動小数点の指数値のオーバフローもしくはアンダフロー、または丸めている間の不正確な誤差を含む。これらの誤差は信号で保存局141に送られる。

【0027】

浮動小数点スタックは、浮動小数点命令により用いられる。浮動小数点命令は、スタックからそのオペランドを取る。なお、浮動小数点スタックはプロセッサ110においていくぶんか分配されており、浮動小数点機能ユニット122内にはなく、一般的には、浮動小数点機能ユニット122から構造的に分離されている。

【0028】

リオーダバッファ126は、浮動小数点スタックにあるデータを含む推論データが、プロセッサ110の種々のブロックが協働することにより一貫した態様で、しかしながら浮動小数点機能ユニット122の動作から一般に独立して扱われるようにデータの管理を制御する。リオーダバッファ126において従属性分析を含むデータフロー制御を与えることによって、FPU122を含む他のプロセッサブロックが簡略化される。浮動小数点ユニット122によって用いられる制御情報は、スタックのオーバフローまたはアンダフローの状態を示すビット等のスタック状態ビットに制限される。この情報は命令デコーダ118によって発生され、ROPがディスパッチされると浮動小数点ユニット122に送られ

10

20

30

40

50

る。FPU122は、オーバフローまたはアンダフロートラップを受取ると、例外信号を発生する。

【0029】

図5は、種々のレジスタと、スタックを制御しかつスタックを動作させるためのデータ通信経路を相互接続するためのアレイとを含む、浮動小数点スタックを組込むプロセッサ110のエレメントを示している。図5は、分岐の予測および誤った予測によりスタック機能性のいくつかの局面が左右されるため、分岐機能を実現するエレメントを示している。浮動小数点スタックは、命令デコーダ118、分岐ユニット120、リオーダバッファ126およびレジスタファイル124内に記憶および制御回路を含む。なお、本実施例のプロセッサ110では、浮動小数点機能ユニット122は、浮動小数点スタックの構造のうち

10

のいずれも含んでおらず、それにより浮動小数点命令および浮動小数点スタック交換命令を同時に実行することができる。

【0030】

スタックに影響を及ぼす命令のタイプは2つある。スタックに影響を及ぼす命令の第1のタイプは、浮動小数点命令である。これらの命令は、スタック上のデータを用い、結果をスタックに戻す。このスタックに影響を及ぼす命令の第1のタイプは、浮動小数点ユニット122において実行される。スタックに影響を及ぼす命令の第2のタイプは、スタックのエレメントを交換する浮動小数点スタック交換(FXCH)命令である。種々の理由のため、FXCH命令は分岐ユニット120において実行される。

【0031】

FXCH命令が分岐ユニット120において実行される1つの理由は、データオペランドの値が推論的であるのと同様にスタックエレメントの順序が推論的であることである。条件つき分岐は誤予測され得るため、誤予測された分岐に従うFXCH命令によって変更されるスタックエレメントの順序を復元しなければならない。分岐がディスパッチされるときにルックアヘッドスタックエレメントの順序をセーブするために、FXCH命令は分岐ユニット120にディスパッチされる。FXCH命令を分岐ユニット120において実行する第2の理由は、プロセッサ110が、分岐ユニット120によって開始される再同期化動作を介してスタックアンダフロー状態等のスタックエラーに応答することである。

20

【0032】

命令キャッシュ116と分岐ユニット120とが協働することにより、ターゲットPCバス322および分岐フラグ310を介する通信を用いて分岐予測能力が得られる。命令キャッシュ116は、バイトキューバス348を介して命令デコーダ118に命令を与える。分岐ユニット120は、スタックのルックアヘッド状態を特定の分岐命令と関連させるデータをストアするレジスタを含む。FXCHおよび浮動小数点命令を同時に実行できるように、FXCH命令を浮動小数点ユニット122にではなく分岐ユニット120にディスパッチすることが有利である。

30

【0033】

命令デコーダ118は、与えられた命令に対応するROPを種々のバスを介して種々の機能ユニットにディスパッチし、この種々の機能ユニットのうちの一つのは分岐ユニット120である。命令デコーダ118は、ROPをディスパッチすると、ROPのソースオペランドおよび宛先レジスタを識別するために、Aオペランドポインタ136、Bオペランドポインタ137、および宛先ポインタ143をレジスタファイル124およびリオーダバッファ126に駆動する。命令デコーダ118は、デコード(DPC)バス313を介して分岐ユニット120にデコードプログラムカウンタ(PC)を送る。命令デコーダ118は、スタックのルックアヘッド状態をストアするレジスタおよびアレイを含む。浮動小数点ROPに関しては、ルックアヘッドのスタックのレジスタおよびアレイは、レジスタファイル124およびリオーダバッファ126のエレメントにアクセスするために、オペランドポインタバス136、137上で駆動されるポインタと宛先ポインタ143との値を引出すために用いられる。非推論的な整数および浮動小数点データは共にレジスタファイル124にストアされる。浮動小数点スタックは、レジスタファイル124内のレジ

40

50

スタの形である。推論的な整数および浮動小数点データはリオーダバッファ 1 2 6 にストアされる。命令デコーダ 1 1 8 は、ルックaheadスタックポインタおよびアレイを用いて、浮動小数点オペランドの指定を、スタックのトップ位置に関するスタックエレメントの識別からレジスタファイル 1 2 4 内の物理レジスタの識別に変換する。この変換が行なわれると、リオーダバッファ 1 2 6 の推論的浮動小数点は整数オペランドと同様に処理される。データ処理のほとんどの局面に関して、プロセッサ 1 1 0 は浮動小数点データを整数データと同様に扱い、これにより専用論理が必要でなくなる。

【 0 0 3 4 】

命令デコーダ 1 1 8 は、命令処理パイプラインの最初にある。整数データと浮動小数点データとをパイプラインの各段で同じように一貫して処理することが有利である。スタックのルックahead状態は、R O P がデコードされるときに決定される。命令デコーダ 1 1 8 は、ルックaheadスタックポインタおよびリマップアレイの更新を制御し、浮動小数点オペランドの識別をスタック上の位置の指定から固定レジスタの指定に変換する。命令デコーダ 1 1 8 は、命令パイプラインの最初の位置にあるため、この命令デコーダ 1 1 8 により浮動小数点データおよび整数データをプロセッサパイプラインにおけるできるだけ早い段階で一貫した態様で処理することができるようになる。

10

【 0 0 3 5 】

レジスタファイル 1 2 4 は、スタックトップ位置ポインタおよびリマップレジスタを含む、浮動小数点スタック、浮動小数点スタック制御ポインタおよびアレイを保持するためのレジスタを有する。したがって、命令デコーダ 1 1 8 はスタック制御エレメントの推論的状态を保持し、リオーダバッファ 1 2 6 は推論的状态にあるいかなるスタックデータをも保持し、レジスタファイル 1 2 4 は非推論的浮動小数点スタックデータおよびスタック制御エレメントをストアする。

20

【 0 0 3 6 】

リオーダバッファ 1 2 6 はプロセッサの回復および再始動手順を制御する。浮動小数点スタック回復および再始動機能は、スタックをレジスタファイル 1 2 4 に物理的に組込むことによって、およびオペランドが回収されるとスタックレジスタおよびアレイの書込を制御するためにリオーダバッファ 1 2 6 を用いることによって達成される。リオーダバッファ 1 2 6 は、スタックの推論的状态を含むプロセッサ 1 1 0 の推論的状态を追跡するため、この更新のタイミングを制御する。

30

【 0 0 3 7 】

プロセッサ 1 1 0 の分岐予測能力およびそれが浮動小数点スタックに与える影響をよりよく理解するために、図 6 に詳細に示される命令キャッシュ 1 1 6 のアーキテクチャを考える。命令キャッシュ 1 1 6 は、命令デコーダ 1 1 8 のためのプリフェッチされた x 8 6 命令バイトをプリデコードする。命令キャッシュ 1 1 6 は、キャッシュコントロール 4 0 8 、フェッチプログラムカウンタ (P C) 4 1 0 、フェッチ p c バス 4 0 6 、プリデコード 4 1 2 、コードセグメント 4 1 6 、バイトキューシフト 4 1 8 、バイトキュー 1 3 5 、および 3 つのアレイ、すなわち命令ストアアレイ 4 5 0 とアドレスタグアレイ 4 5 2 とサクセッサアレイ 4 5 4 とに組織化されるキャッシュアレイ 4 0 0 を含む。

【 0 0 3 8 】

コードセグメントレジスタ 4 1 6 は、リクエストされたメモリアクセスの有効性をチェックするために用いられるコードセグメントディスクリプタのコピーを保持する。コードセグメント 4 1 6 は、アプリケーションのアドレス空間にあるアドレスである論理アドレスをプロセッサ 1 1 0 のアドレス空間にあるアドレスである線形アドレスに変換するために分岐ユニット 1 2 0 において用いられるコードセグメント (C S) ベース値を与える。C S ベースは、C S ベースライン 4 3 0 を介して分岐ユニット 1 2 0 に伝えられる。プリデコード 4 1 2 は、内部アドレス / データバス 1 1 1 を介して、プリフェッチされた x 8 6 命令バイトを受取り、各々の x 8 6 命令バイトにプリデコードされたビットを割当て、プリデコードされた x 8 6 命令バイトをバス 4 0 4 を介して命令ストアアレイ 4 5 0 に書込む。バイトキュー 1 3 5 は、キャッシュアレイ 4 0 0 からの予測実行された命令を保持し

40

50

、16以下の有効なプリデコードされた×86命令バイトを16本のバス348を介して命令デコーダ118に与える。バイトキューシフト418は、×86の境界において命令を循環させ、マスクし、かつシフトする。シフトは、×86命令のすべてのROPが命令デコーダ118によってディスパッチされるとシフト制御ライン474上の信号にตอบสนองして生じる。キャッシュコントロール408は、命令キャッシュ116の動作を管理するために制御信号を発生する。

【0039】

レジスタ410にストアされかつフェッチpcバス406を介してやり取りされるフェッチPCは、キャッシュアレイ400の3つのアレイのアクセス中にフェッチされるべき命令を識別する。中位フェッチPCビットは、検索のために各アレイからのエントリをアドレス指定するキャッシュインデックスである。高位ビットは、比較420によってアドレス指定されたタグと比較され、かつアドレスタグアレイ452から取出されるアドレスタグである。一致すると、それはキャッシュヒットを表わす。低位ビットは、命令ストアアレイ450からのアドレス指定されかつ取出されたエントリのアドレス指定されたバイトを識別するオフセットである。フェッチPC410、キャッシュコントロール408、およびキャッシュアレイ400は協働して、フェッチpcバス406を介して伝えられたアドレスを維持しかつ再送する。フェッチPCレジスタ410は、ポインタ値を保持するか、ポインタを増分するか、内部アドレス/データバス111を介してポインタを受取るか、またはターゲットpcバス322からのポインタをロードすることによって、1つのサイクルからその次のサイクルでポインタを更新する。ターゲットpcは、分岐命令が実行されかつそれが誤予測されたものであることがわかると分岐ユニット120から受取られる分岐フラグ310の分岐誤予測フラグ417にตอบสนองしてキャッシュコントロール408によってフェッチPCレジスタ410にロードされる。

【0040】

アドレスタグアレイ452のエントリは、キャッシュヒットを識別するためのアドレスタグと、アドレスタグの有効性を示すための有効ビットと、命令ストアアレイ450のバイトの各々に対応し、プリデコードされた×86命令バイトが有効な×86命令バイトおよび有効なプリデコードビットを含むかどうかを示すためのバイト有効ビットとを含む。

【0041】

分岐予測をサポートするサクセッサアレイ454は、サクセッサインデックス、サクセッサ有効ビット(NSEQ)、およびブロック分岐インデックス(BBI)を含むエントリを有する。サクセッサアレイが命令ストアアレイ450をアドレス指定するとNSEQがアサートされ、命令ブロックの分岐がいずれも「予測発生されていない」場合はNSEQはアサートされない。NSEQがアサートされかつ予測実行された最後の命令バイトの現在の命令ブロック内のバイト位置を指定するときのみBBIが規定される。サクセッサインデックスは、推論的分岐のターゲット位置から始まる、その次の予測実行された命令の最初のバイトのキャッシュ位置を示す。

【0042】

分岐命令は、命令キャッシュ116と分岐ユニット120との動作を調製することによって行なわれる。たとえば、命令キャッシュ120が分岐がまだ発生していないと予測すると、命令はシーケンシャルにフェッチされる。その後分岐が分岐ユニット120による実行の際に発生されると、予測は間違っていることになり、分岐ユニット120は分岐誤予測フラグ417および分岐発生フラグ418をアサートする。分岐ユニット120は、ターゲットpcバス322を介して正しいターゲットPCを命令キャッシュ116に戻し、これはフェッチPCレジスタ410にストアされる。命令ストアアレイ450は、フェッチPCレジスタ410の値に従って、ターゲットpcアドレスで始まる命令ストリームを与え、バイトキュー135を再び満たし始める。ROB126およびFPスタックの推論的状态は流される。

【0043】

命令キャッシュ120が分岐が発生したと予測すると、その次の命令はシーケンシャルで

10

20

30

40

50

はない。サクセッサアレイ 454 のエントリが予測発生された分岐命令に割当てられ、NSEQビットがアサートされると、分岐命令の最後のバイトを指すようにBBIが設定され、ターゲット命令の命令キャッシュ116内の位置を示すようにサクセッサインデックスが設定される。サクセッサインデックスは、完全なアドレスではなく、命令ストアアレイ450のターゲット命令のインデックス、カラムおよびオフセットをストアする。シーケンシャルではないその次の命令に関するフェッチPCは、サクセッサインデックスによって与えられたインデックスおよびカラムを用いてキャッシュブロックにアクセスすることによって、およびそのブロック内にストアされたアドレスタグの高位ビットとその前のサクセッサインデックスからのインデックスおよびオフセットビットとを連結することによって構成される。

10

【0044】

構成された分岐ターゲットは、命令キャッシュ116からフェッチpcバス406を介して命令デコーダ118に送られ、命令デコーダ118によって、命令がデコードされるときにデコードPCを維持するために用いられる。

【0045】

命令デコーダ118は、分岐ユニット120に分岐命令をディスパッチすると、デコードpcバス313を介してデコードPCを送り、オペランドバス130を介してターゲットの分岐オフセットを送る。この情報は、分岐命令を実行するために、および予測を確認するために分岐ユニット120によって用いられる。

【0046】

図8および図9に示される命令デコーダ118は、バイトキュー135からプリデコードされたx86命令バイトを受取り、それらをROPのそれぞれのシーケンスに翻訳し、複数のディスパッチ位置からROPをディスパッチする。単純な命令に関しては、翻訳は、ハードウェアに組込まれた高速変換経路を介して行なわれる。マイクロコードROMシーケンスは、使用頻度の少ない命令と、3つよりも多いROPに翻訳する複雑な命令とを扱う。命令デコーダ118は高速経路またはマイクロコードROMからのROP情報を選択し、かつ増加させ、機能ユニットによる実行のために完全なROPを与える。

20

【0047】

ROPマルチプレクサ500は、バイトキュー135の先頭にあるx86命令から始まる、バイトキュー135の1つ以上のプリデコードされたx86命令を1つ以上の利用可能なディスパッチ位置に同時に送る。ROPディスパッチ位置ROP0、1、2、3(510、520、530、540)はそれぞれ、高速変換器0、1、2、3(順に、512、522、532、542)と、共通段0、1、2、3(514、524、534、544)、マイクロコードROM0、1、2、3(516、526、536、546)とを含む。各ディスパッチ位置は、共通段、高速変換器、およびMROMを含む。MROM516、526、536、546は、マイクロコードROM(MROM)コントローラ560によって制御される。

30

【0048】

共通段は、アドレス指定モードの処理を含む、高速経路およびマイクロコードROM命令に共通するパイプライン処理およびx86命令変換動作を扱う。

40

【0049】

MROMコントローラ560は、命令タイプおよび操作コードを与える、ディスパッチウィンドウを満たすROPの数を予測する、命令キャッシュ116の分岐予測に従ってバイトキュー135のシフトをガイドする、ROPマルチプレクサ500にROPの数を知らせてバイトキュー135の先頭にあるx86命令に関してディスパッチする、マイクロコードおよび制御ROMにアクセスする、等の制御機能を果たす。MROMコントローラ560は、2つの方法、すなわち命令レベルのシーケンス制御、およびマイクロ分岐ROPを用いてROPの順序づけを制御する。命令レベルの分岐およびマイクロ分岐ROPはともに、誤った予測を確認し、かつ訂正するために分岐ユニット120にディスパッチされる。命令レベルシーケンス制御フィールドは、マイクロコードサブルーチン呼出/リターン

50

、ブロックに整列されたMROM位置に対する無条件分岐、プロセッサの状態に基づく条件つき分岐、およびシーケンスの終わりの識別のようないくつかの能力を与える。命令レベルのシーケンスROPがディスパッチされると、(命令アドレスではなく)MROMアドレスがターゲットの形成または分岐の訂正のために送られる。

【0050】

マイクロ分岐ROPが無条件分岐および状態フラグ125に基づく条件つき分岐を与える。マイクロ分岐ROPは、実行のために分岐ユニット120にディスパッチされる。MROMコントローラ560は、分岐ユニット120のマイクロ分岐誤予測論理によって開始されるマイクロコードROMエントリポイントを受入れる。分岐ユニット120によって発生されたマイクロコードエントリポイントは、ターゲットpcバス322を介して命令デコーダ118に送られる。マイクロ分岐が訂正されると、分岐ユニット120は、訂正アドレスがPCではなくMROMアドレスであることをターゲットpcバス322を介して命令デコーダ118に示す。

10

【0051】

ROPセレクト0、1、2、3(518、528、538、548)は、共通段の出力と組合せて高速変換器またはMROMの出力を選択し、この情報をレジスタファイル124、リオーダバッファ126、および種々の機能ユニットに送る。

【0052】

ROP共有590は、すべてのディスパッチ位置によって共有される資源によって用いられる情報をディスパッチする。ROP共有590は、機能ユニットにディスパッチするために、操作コード/タイプバス150にROP操作コードを符号化したものを与える。

20

【0053】

分岐ユニット120は、操作コードと、1ビット交換アンドフロー信号、2ビットキャッシュラム選択識別子、1ビット分岐予測発生選択信号、1ビットマイクロ分岐インジケータ、および分岐ユニット120がターゲットpcバス322上の予測発生されたアドレスを分岐予測発生FIFO(図10の906)に書込むべきであるかどうかを示す1ビット信号を含む他のROP共有590の出力とを受取る。さらに、整数フラグソースオペランドを識別する3ビット読取フラグポイントが、分岐ユニット120にマップされる最初のディスパッチされていないROPの位置に基づいて設定される。分岐ユニット120にROPがマップされていないならば、読取フラグポイントは0に設定される。2ビット利用インジケータは、分岐ユニット120にマップされる最初のディスパッチされていないROPのディスパッチ位置を設定するように符号化される。

30

【0054】

命令デコーダ118は、デコードPC582、デコーダコントロール584、およびデコーダスタック586を含む。デコーダコントロール584は、バイトキュー135のx86命令の数、(ライン570からの)機能ユニットの状態、および(ライン572からの)リオーダバッファの状態に基づいて発行されるべきROPの数を決定する。デコーダコントロール584は、バイトキュー135が完全に実行されたx86命令の数だけシフトしかつバイトキュー135の始まりが常にその次の完全なx86命令の開始となるように、バイトキュー135に発行されたROPの数をシフト制御ライン474を介して送る。例外または分岐が誤って予測されると、デコーダコントロール584は、例外マイクロコードルーチンのために、新しいフェッチPCが入力されるかまたはエントリポイントがMROMに送られるまで、付加的なROPの発行を妨げる。

40

【0055】

デコードPC582は、バイトキュー135からの各々のx86命令の論理PCを追跡する。シーケンシャルでないフェッチが検出されると、デコードPC582は新しいポイントを含む。シーケンシャルな命令が分岐の後に生じると、デコードPC582は、壊されていないシーケンスの最初と最後の位置の間のバイトキュー135のx86バイトの数をカウントし、この数を現在のPCに加えて、そのシーケンスに続くその次のPCを決定する。デコードPCは、DPCバス313を介して分岐ユニット120に伝えられる。

50

【 0 0 5 6 】

デコーダスタック 5 8 6 は、ルックアヘッドスタックトップ位置 (T O S) ポインタ 5 0 2、ルックアヘッドリマップアレイ 5 0 4、およびルックアヘッドフル/エンプティアレイ 5 0 6 を含む種々の浮動小数点スタックポインタアレイおよびレジスタのルックアヘッドコピーを保持する。これらのアレイおよびポインタは、スタックを分岐の誤予測または例外に従った適切な状態に戻すことを含む、スタックに影響を与える R O P の推論的な発行から生じる浮動小数点スタックの推論的変更を扱う。

【 0 0 5 7 】

ルックアヘッドリマップアレイ 5 0 4 は、各々がスタックアレイの 1 つのレジスタを指定するポインタのアレイである。スタックの例示的な実施例では、ルックアヘッドリマップアレイ 5 0 4 は、各々がレジスタファイル 1 2 4 内の浮動小数点スタックアレイ 7 0 0 のエレメントを識別する 8 つの 3 ビットポインタのアレイである。ルックアヘッド T O S 5 0 2 は、ルックアヘッドリマップアレイ 5 0 4 の 1 つのポインタを選択する 3 ビットポインタである。ルックアヘッドフル/エンプティアレイ 5 0 6 は、スタックエントリがフル (1) であるかエンプティ (0) であるかを指定する単一ビットのアレイである。

【 0 0 5 8 】

スーパースカラプロセッサでは、動作がディスパッチされても、その実行が適切であることの確認にはならない。分岐が予測されると、その予測のうちのいくつかは不正確である。ルックアヘッドリマップアレイ 5 0 4、ルックアヘッド T O S 5 0 2、およびルックアヘッドフル/エンプティアレイ 5 0 6 は、浮動小数点スタックの推論的状态のコピーをセーブするために用いられ、それにより誤予測された分岐からの回復が加速される。浮動小数点スタックを変更する動作に関しては、命令デコーダ 1 1 8 は、命令をデコードすると、浮動小数点スタックアレイ 7 0 0 の未来の状態を更新する。命令デコーダ 1 1 8 は、スタックポインタを増分または減分する命令をデコードすると、ルックアヘッド T O S 5 0 2 を更新する。同様に、命令デコーダ 1 1 8 は、浮動小数点交換命令 (F X C H) をデコードすると、その命令によって特定されるようなポインタを交換することによってルックアヘッドリマップアレイ 5 0 4 の未来の状態を調節する。スタックの状態がいかなる 2 つの分岐命令の間でも変化し得るため、スタック情報はすべての分岐動作のために保存される。

【 0 0 5 9 】

浮動小数点 R O P に関しては、ルックアヘッド T O S 5 0 2 およびルックアヘッドリマップアレイ 5 0 4 は、A オペランドポインタ 1 3 6、B オペランドポインタ 1 3 7、および宛先レジスタポインタ 1 4 3 の値を決定するために組合せて用いられる。したがって、浮動小数点 R O P がデコードされると、そのオペランドは、浮動小数点スタックの位置によって明確にまたは暗に指定される。スタックのトップ位置にあるオペランドに関しては、ルックアヘッド T O S 5 0 2 は、ルックアヘッドリマップアレイ 5 0 4 のエレメントを指し、このルックアヘッドリマップアレイ 5 0 4 のエレメントは浮動小数点スタックアレイ 7 0 0 上の位置を指定する。この位置は、レジスタファイル 1 2 4 における浮動小数点レジスタに対応する。この位置は、スタックのトップ位置にあるいかなるオペランドまたは宛先レジスタに関しても、A オペランドポインタ 1 3 6、B オペランドポインタ 1 3 7、および宛先レジスタポインタ 1 4 3 として適用される。同様に、スタックのトップ位置に関するいかなる位置に対するポインタも、ルックアヘッド T O S 5 0 2 から指定された量だけオフセットされたポインタを適用することによって決定される。このようにしてルックアヘッド T O S 5 0 2 およびリマップアレイ 5 0 4 からオペランドおよび宛先ポインタを引出すことにより、レジスタファイル 1 2 4 およびリオーダーバッファ 1 2 6 が浮動小数点 R O P および整数 R O P の両方に関して同じ態様で推論的にまたは非推論的にデータを処理することができるようになる。

【 0 0 6 0 】

図 1 0 を参照して、レジスタファイル 1 2 4 は、読取デコーダ 6 6 0、レジスタファイルアレイ 6 6 2、書込デコーダ 6 6 4、レジスタファイルコントロール 6 6 6、およびレジ

10

20

30

40

50

スタファイルオペランドバスドライバ668を含む。読取デコーダ660はAオペランドポインタ136およびBオペランドポインタ137を受取り、4対の64ビットのAオペランドアドレス信号およびBオペランドアドレス信号RA0、RA1、RA2、RA3、RB0、RB1、RB2、RB3によってレジスタファイルレイ662をアドレス指定する。レジスタファイルレイ662は、ライトバックバス134を介してリオーダバッファ126から結果データを受取る。リオーダバッファエントリが3個以下の他のリオーダバッファエントリと平行に回収されると、エントリに関する結果データがライトバックバス134のうちの1つに置かれ、そのエントリに関する宛先ポインタがそのライトバックバス134に対応する書込ポインタ133に置かれる。ライトバックバス134上のデータは、書込デコーダ664に与えられる書込ポインタ133上のアドレス信号に従ってレジスタファイルレイ662の指定されたレジスタに送られる。

10

【0061】

浮動小数点スタックの種々のレジスタおよびアレイに影響を及ぼす特定のROPを回収すると、リオーダバッファ126は、浮動小数点リマップアレイ674、浮動小数点トップ・オブ・スタック(TOS)レジスタ672、および浮動小数点フル/エンプティアレイ676を含むレジスタファイル124内の種々の浮動小数点スタックレジスタにデータを駆動する。レジスタファイル124内に配置される浮動小数点スタックアレイ700(図11)は、拡張された実データをストアするための8つの82ビット数値レジスタのアレイである。レジスタの各々は、1つの符号ビット、19ビット指数フィールド、および62ビット有効数字部フィールドを含む。浮動小数点リマップアレイ674は、各々が浮動小数点スタックアレイ700のレジスタに対するポインタである8つのポインタのアレイである。浮動小数点TOS672は、浮動小数点リマップアレイ674へのポイントを指定する3ビットのポインタである。浮動小数点フル/エンプティアレイ676は、スタックアレイの位置がフル(1)であるかエンプティ(0)であることを示し、各々が浮動小数点スタックアレイ700の元素に対応する単一ビットのアレイである。

20

レジスタファイルレイ662は、プロセッサ機能ユニットにおいて演算されかつ発生される結果をストアするための複数のアドレス指定可能なレジスタを含む。図11は、8つの32ビット整数レジスタ(EAX、EBX、ECX、EDX、ESP、EBP、ESI、EDI)、8つの82ビット浮動小数点レジスタFP0~FP7、16個の41ビット一時整数レジスタETMP0~ETMP15、および本実施例では一時整数レジスタETMP0~ETMP15と同じ物理レジスタ位置にマップされる8つの82ビット一時浮動小数点レジスタFTMP0~FTMP7を含む40個のレジスタを備える例示的なレジスタファイルレイ662を示している。浮動小数点レジスタFP0~FP7は、浮動小数点スタックアレイ700としてアドレス指定され、これらはルックアヘッドTOS502およびルックアヘッドリマップアレイ504を用いて得られるとAオペランドポインタ136、Bオペランドポインタ137および宛先レジスタポインタ143を用いてアクセスされる。

30

【0062】

図12を参照して、リオーダバッファ126は、リオーダバッファ(ROB)コントロールおよびステータス870、ROBアレイ874、およびROBオペランドバスドライバ876を含む。ROBコントロールおよびステータス870は、ROPのソースオペランドおよび宛先オペランドを識別する入力を受取るようにAオペランドポインタ136、Bオペランドポインタ137、および宛先ポインタ(DEST REG)バス143に接続される。ROBアレイ874は、ROBコントロールおよびステータス870によって制御されるメモリアレイである。ROBアレイ874は、機能ユニットから結果を受取るように、結果バス132に接続される。先頭、末尾、Aオペランド選択、Bオペランド選択、および結果選択信号を含む制御信号は、ROBコントロールおよびステータス870からROBアレイ874に伝えられる。これらの制御信号が、結果バス132から入力されかつライトバックバス134、ライトポインタ133、Aオペランドバス130、Bオペランドバス131、Aオペランドタグバス148、およびBオペランドタグバス149に

40

50

出力される R O B アレイエレメントを選択する。各リオーダバッファアレイエレメントに 1 つである 1 6 個の宛先ポインタが、従属性をチェックするために R O B アレイ 8 7 4 から R O B コントロールおよびステータス 8 7 0 に与えられる。適切な従属性検査回路は、1 9 9 4 年 4 月 2 6 日出願の米国特許出願 (スコット・エイ・ホワイト (Scott A. White) 「環状けた上げルックaheadを用いる範囲発見回路 (A Range-Finding Circuit using Circular Carry Lookahead) 」) に詳細に記載されており、これを引用によりここに援用する。

【 0 0 6 3 】

図 1 3 は、図 1 2 と関連して、各々が 4 1 ビットの結果フィールド、9 ビットの宛先ポインタフィールド、4 ビットの下位プログラムカウンタフィールド、1 1 ビットの浮動小数点操作コードフィールド、1 1 ビットの浮動小数点フラグレジスタフィールド、および 2 4 ビットのコントロールおよびステータスフィールドを有する 1 6 個のエントリを含むリオーダバッファアレイ 8 7 4 の一例である。4 1 ビットの結果フィールドは、機能ユニットから受取った結果をストアするために与えられる。2 つのリオーダバッファエントリは、浮動小数点結果をストアするために用いられる。整数の結果は 4 1 ビットのうちの 3 2 ビットにストアされ、残りの 9 ビットは状態フラグを保持するために用いられる。R O B アレイ 8 7 4 の各エントリの宛先ポインタフィールド (D E S T P T R 8 : 0) は、レジスタファイル 1 2 4 の宛先レジスタを指定する。浮動小数点操作コードフィールドは、リオーダバッファエントリに割り当てられる命令に対応する x 8 6 浮動小数点操作コードのビットのサブセットをストアする。浮動小数点フラグレジスタフィールドは、浮動小数点動作から得られる浮動小数点フラグの状態をストアする。浮動小数点フラグは、浮動小数点機能ユニット 1 2 2 によって検出される精度、アンダフロー、オーバフロー、ゼロ除算、非正規化オペランドおよび無効オペランドのエラーに関する情報をストアする。コントロールおよびステータスフィールドは、たとえば A L L O C A T E ビット、B R A N C H T A K E N ビット、M I S P R E D I C T ビット、V A L I D ビット、E X I T ビット、U P D A T E E I P ビット、および E X C E P T I O N ビット等の R O B エントリの状態を示すビットを含む。A L L O C A T E ビットは、リオーダバッファエントリが割り当てられるかどうかを指定する。B R A N C H T A K E N ビットは、分岐ユニット 1 2 0 が分岐が発生された分岐命令を実行したことを信号で示す。M I S P R E D I C T ビットは、分岐が不正確に予測されることを示す。V A L I D ビットは、結果が有効でありかつ命令が終了することを示す。E X I T ビットは、R O P が特定の x 8 6 命令の R O P のシーケンスにおいて最後の R O P であることを示し、拡張命令ポインタ (E I P) レジスタ (図示せず) の更新をトリガするために用いられる。U P D A T E E I P ビットはまた、E I P レジスタが更新されるべきであることを示す。E X C E P T I O N ビットは、命令の実行により例外またはエラー状態が生じたことを示す。

【 0 0 6 4 】

さらに、コントロールおよびステータスフィールドはまた、スタックポインタを更新するための S T A C K ビットを含む。命令デコーダ 1 1 8 は、浮動小数点 R O P をディスパッチすると、スタックを更新するための情報をリオーダバッファ 1 2 6 に送る。この情報は、動作が回収されるとスタックポインタで行なうための動作を指定するコードを含む。スタックはプッシュされるか、ポップされるか、2 度ポップされるか、または変えないままにすることが可能である。リオーダバッファ 1 2 6 は、動作の実行が終了しかつオペランドが回収されるまで、リオーダバッファアレイ 8 7 4 内のエントリの S T A C K ビットコントロールおよびステータスフィールドにこの情報を保持する。

【 0 0 6 5 】

機能ユニットがスタック変更命令の実行を終了しかつそれ以前のすべてのプログラムの順序の動作が終了されかつそれらのオペランドが回収されると、リオーダバッファ 1 2 6 は、もし分岐の誤予測または例外等のエラーが生じていなければ動作を回収する。スタックは、リオーダバッファアレイ 8 7 4 内のエントリのコントロールフィールドによって指定される動作に従って更新される。たとえば、浮動小数点 T O S 6 7 2 は、スタックをポッ

プする場合には増分され、スタックを2回ポップする場合には2だけ増分され、プッシュするかまたは変えないままにする場合には減分される。

【0066】

F X C H命令が実行されると、分岐ユニット120は4本の結果バス132のうちの1本を介してルックアヘッドリマップアレイのコピーをリオーダバッファ126に送る。回収の際に、リオーダバッファ126は、ライトバックバス134のうちの1つを介してこのルックアヘッドリマップアレイ504の値を浮動小数点リマップアレイ674に駆動し、この浮動小数点リマップアレイ674でその値がストアされる。リオーダバッファ126から浮動小数点T O S 672への付加的なライン(図示せず)は、スタックポインタを更新するために用いられる。レジスタファイルアレイ662は、浮動小数点スタックアレイ700のエントリが更新されると浮動小数点フル/エンプティアレイ676に0および1を書込む回路(図示せず)を含む。このようにして、推論的な浮動小数点スタック交換が非推論的となる。

10

【0067】

図14に示される分岐ユニット120は、ジャンプおよび呼出動作、復帰マイクロルーチンを含む、シーケンシャルなプログラム順序に従っていない命令のフェッチを制御する。分岐ユニット120は、加算器910およびインクリメンタ912に接続される分岐保存局902と、分岐予測比較論理908と、分岐リマップアレイ904とを含む。分岐リマップアレイ904は、浮動小数点スタックの一部である。分岐ユニット120はさらに、「予測発生される」分岐を追跡する分岐予測発生F I F O 906を含む。分岐予測発生F I F O 906のエントリは、対応する分岐のキャッシュ位置と、予測発生された分岐のP Cとを保持する。予測発生された分岐のP Cは、分岐が正しく予測されるかどうかを決定するために分岐予測比較論理906に与えられる。加算器910およびインクリメンタ912は、デコードP Cに関する分岐のアドレスを計算する。命令キャッシュ116によって分岐が予測発生されると、そのシーケンシャルでない予測されたターゲットP Cは、分岐ブロックのP C、カラム、およびB B Iから形成される分岐の位置とともに分岐予測発生F I F O 906に駆動され、かつ分岐予測発生F I F O 906にラッチされる。分岐ユニット120は、加算器910またはインクリメンタ912を用いて、プログラムカウンタを決定することによって対応する分岐R O Pを実行する。たとえば、分岐が発生されると、分岐命令のP Cからのターゲットプログラムカウンタと、オペランドバス130を介してオペランドとして供給されたオフセットパラメータとを計算するために加算器910が用いられる。分岐ユニット120によって更新されるプログラムカウンタと、D P Cバス313を介して命令デコーダ118から供給されるデコードP Cとが一致すると、分岐ユニット120は結果バス132を介して結果をリオーダバッファ126に駆動する。この結果は、ターゲットP Cと、一致を示す状態コードとを含む。分岐が誤予測されると、正しいターゲットは命令キャッシュ116に駆動され、フェッチR Cを再送する。

20

30

【0068】

分岐保存局902は、命令デコーダ118から操作コード/タイプバス150を介してR O P操作コードを受取り、かつ、Aオペランドバス130およびBオペランドバス131を介してレジスタファイル124およびリオーダバッファ126からオペランドを受取り、かつ、さらに結果バス132から結果データを受取るマルチエレメントF I F Oアレイである。保存局の各エレメントは、1つの分岐命令に関する操作コード情報をストアする。複数個の分岐命令は、そのキュー内に保持され得る。分岐保存局902によって受取られる情報は、デコードP C、分岐予測、および分岐オフセットを含む。デコードP Cは、デコードP Cバス313を介してやり取りされる。分岐予測は、分岐予測ラインを介して伝えられる。オフセットは、リオーダバッファ126を通過しAオペランドバス130およびBオペランドバス131を介して分岐ユニット120に送られる。オフセットは、リオーダバッファ126を通過しAオペランドバス130およびBオペランドバス131を介して分岐ユニット120に送られる。

40

【0069】

50

命令デコーダ 118 は、分岐ユニット 120 に分岐命令をディスパッチすると、分岐保存局 902 にストアされるルックアヘッド TOS 502 およびルックアヘッドフル/エンブティアレイ 506 とやり取りされる。好ましくは、ルックアヘッドリマップアレイ 504、ルックアヘッドフル/エンブティアレイ 506、およびルックアヘッド TOS 502 は、予測が正しいときにはプロセッサ 110 がある態様で機能しかつ予測が間違っている場合にはそれと異なった態様で機能するように、分岐ユニット 120 による処理に利用可能である。

【0070】

予測発生された分岐命令 ROP がデコードされ発行されると、デコード PC、オフセット、および予測がディスパッチされ、分岐ユニット 120 の保存局 902 に保持される。予測されたターゲットカウンタがデコード PC と一致すると、分岐は正しく予測されたことになり、正しい予測を反映する結果情報が正しくリオーダバッファ 126 に戻される。この情報は、ターゲット PC と、一致が達成されたことを示す状態コードとを含む。分岐が誤予測されると、分岐ユニット 120 は正しいターゲットを命令キャッシュ 116 およびリオーダバッファ 126 の両方に駆動し、命令ブロックインデックスを命令キャッシュ 116 に送る。このインデックスは、分岐予測発生 FIFO 906 を更新するために用いられる予測情報を表わす。リオーダバッファ 126 は、その後続く ROP の結果を取消すことにより、誤予測された分岐に回答する。

【0071】

分岐ユニット 120 はまた、誤予測が発生した場合、命令デコーダ 118 からの論理アドレスを線形アドレスに変換する。これを行なうために、コードセグメントベースポイントの局所コピーが、命令キャッシュ 116 のコードセグメント 416 によって分岐ユニット 120 に供給される。分岐ユニット 120 は、浮動小数点交換命令 (FXCH) を実現しかつ浮動小数点動作を加速するために、浮動小数点 TOS 672、浮動小数点リマップアレイ 674、および浮動小数点フル/エンブティアレイ 676 を含む浮動小数点スタック回路の推論的更新を管理する。分岐ユニット 120 は、推論的分岐が生じたときには常に現在のスタック状態のコピーを保存することによってこれらの目的を果たす。分岐リマップアレイ 904 は、各 FXCH 命令でディスパッチされるルックアヘッドリマップアレイ 504 からコピーされる。他の実施例では、分岐リマップアレイ 904 は、ルックアヘッドリマップアレイ 504 と同じ情報をストアするため、絶対に必要であるわけではない。しかしながら、例示的な実施例では、分岐命令ごとにはではなく必要な場合にのみルックアヘッドリマップアレイ 504 とやり取りする。ここに記載する実施例では、ルックアヘッドリマップアレイ 504 は FXCH 命令に回答してのみ変化するため、ルックアヘッドリマップアレイ 504 は FXCH がリクエストされたときにのみ分岐ユニット 120 に送られる。

【0072】

分岐ユニット 120 は、スタックポインタ、リマップアレイおよびフル/エンブティアレイの正しいコピーを、最後に成功した分岐後に存在した状態にストアすることによって、誤予測に回答する。分岐 ROP が終了すると、分岐ユニット 120 は、分岐予測結果を送るように結果バス 132 を駆動する。分岐が正しく予測されると、浮動小数点 TOS 672、浮動小数点リマップアレイ 674、および浮動小数点フル/エンブティアレイ 676 は変更されずにセーブされる。

【0073】

分岐の誤予測、例外、割込またはトラップなしで FXCH 命令が通常に実行されると、分岐ユニット 120 は命令デコーダ 118 によって送られるルックアヘッドリマップアレイ 504 の値をストアする。実行が終了すると、分岐ユニット 120 はルックアヘッドリマップアレイ 504 の値を結果バス 132 に書込む。命令が回収されると、リオーダバッファ 126 は、ルックアヘッドリマップアレイ 504 を浮動小数点リマップアレイ 674 に書込むことによってレジスタの交換にコミットする。しかしながら、分岐ユニット 120 は、スタックアンダフローエラー等の FXCH 命令に関する問題を検出すると、リオーダ

10

20

30

40

50

バッファ126によって、プロセッサをF X C H命令の際に再始動させる再同期化応答を開始するようにされる。この再同期化応答は、エス・エイ・ホワイト(S. A. White)およびエム・ディ・ゴダード(M. D. Goddard)による「スーパースカラプロセッサの再同期化(RESYNCHRONIZATION OF A SUPERSCALAR PROCESSOR)」と題された本願と同日出願の同時継続中の米国特許出願において議論されており、これを引用によりここに援用する。

【0074】

分岐ユニット120は、F X C H命令R O Pを実行する前にスタックエラーに関して検査する。スタックアンダフローエラーが検出されると、分岐ユニット120はリオーダバッファ126にエラー通知コードを戻し、これによりリオーダバッファ126に再同期化応答を開始させる。これにより、プロセッサをF X C H命令の際に再始動させる。しかしながら、スタックアンダフロー状態の後の再同期化の際に生じるF X C H命令は他のF X C Hとは異なる。特に、非再同期化F X C H命令は、1つのF X C H R O Pを含む。再同期化F X C H命令は、2対の浮動小数点加算(F A D D) R O Pと1つのF X C H R O Pとを含む5つのR O Pを含む。この2対のF A D D R O Pはそれぞれ、F X C H命令において交換される2つの浮動小数点レジスタに0を加える。スタックアンダフローエラーは、空のスタック位置からオペランドを読取るうとすることによって生じる。浮動小数点ユニット122は、ルックアヘッドフル/エンプティレジスタ506に従って、レジスタが空であるかといったかき決定する。交換された浮動小数点レジスタが有効データを含んでいれば、0を加えてもデータの値は変わらない。有効データを含んでいなければ、浮動小数点ユニット122がF A D D R O Pを実行しかつ交換された浮動小数点レジスタが空であれば、浮動小数点ユニット122はトラッピングがマスクされていなければトラップ応答を開始することによって、またはクワイエット非数字(Q N a N)コードをレジスタにロードすることによって応答する。

10

20

【0075】

スタックアンダフローの後に生じる再同期化により、プロセッサ110はF X C H命令に戻り、既知の状態のデータ、すなわち有効データまたはQ N a Nコードを配置し、無効データを用いて実行されるいかなる命令をも含むF X C Hの後に生じる命令を再試行する。

【0076】

なお、すべての浮動小数点命令は、82ビット浮動小数点データに適応するために、41ビットオペランドバス130、131および41ビット結果バス132に関して少なくとも1対のR O Pを含む。

30

【0077】

分岐が誤予測されると、この誤予測された分岐に関して、分岐リマップアレイ904と保存局902にストアされるスタックトップ位置ポインタおよびフル/エンプティアレイとは、誤予測された分岐の前のスタックの状態を示す。分岐ユニット120は、局所的にストアされたりマップおよびT O S値を命令デコーダ118内のルックアヘッドリマップアレイ504およびルックアヘッドT O Sに書込み、スタックの状態を誤予測された分岐の事実上前の状態に戻す。分岐ユニット120のみが誤予測を検出するため、別の機能ユニットではなく分岐ユニット120がスタックをテストしかつ回復する。

【0078】

プロセッサ110が例外状態を検出すると、リオーダバッファ126は、実行が既知の状態再開されるようにそのエントリを流すことによって回復を達成する。リオーダバッファコントロール870は、スタックに関して同様の回復動作を実行する。例外の場合には、リオーダバッファ126は浮動小数点リマップアレイ674をルックアヘッドリマップアレイ504に書込み、浮動小数点T O S 27をルックアヘッドT O S 502に書込み、浮動小数点フル/エンプティアレイ676をルックアヘッドフル/エンプティアレイ506に書込む。

40

【0079】

浮動小数点スタックがF P Uの外で実現されるため、プロセッサ110は浮動小数点演算命令と並行して浮動小数点交換を実行する。この理由のため、浮動小数点スタックコン

50

ポーネント回路は浮動小数点ユニット以外のユニットに組込まれる。したがって、ルックアヘッドリマップアレイ504およびルックアヘッドTOS502は、命令デコーダ118に組込まれる。浮動小数点TOS672、浮動小数点リマップアレイ674、および浮動小数点スタックアレイ700は、レジスタファイル124内に置かれる。分岐ユニット120は、分岐リマップアレイ904を与える。同様に、パラレルな命令処理を促進するために、FXCH命令は、浮動小数点ユニットではなく分岐ユニット120内で実行される。

【0080】

図15および図16はそれぞれ、リマップアレイMAP 23:0 924とスタックトップ位置ポインタTOS 2:0 926とに従ってスタックエントリを選択する、スタック選択信号STi 2:0 928を引出すためのスタック回路920、およびスタック選択信号STI 2:0 929を引出すためのスタック回路922を示している。4つのディスパッチ位置の各々に関してルックアヘッドリマップアレイ504およびルックアヘッドTOS502を与えるために、スタック回路920のマルチプレクサ930および加算器932は、命令デコーダ118において4回複製される。ルックアヘッドリマップアレイ504はMAP 23:0 924に対応する。ルックアヘッドTOS502はTOS 2:0 926に対応する。MAP 23:0 924に対応する浮動小数点リマップアレイ674、およびTOS 2:0 926に対応する浮動小数点TOS672を与えるために、レジスタファイル124には1つのスタック回路920も含まれている。

10

20

【0081】

同様に、ルックアヘッドスタック選択信号であるスタック選択信号STI 2:0 929を引出すために、命令デコーダ118にはスタック回路922のマルチプレクサ934および加算器936が含まれる。スタック回路922は、4つのデコーダディスパッチ位置によって共有される。浮動小数点スタック選択信号であるスタック選択信号STI 2:0 929を引出すために、レジスタファイル124にスタック回路922のマルチプレクサ934および加算器936が含まれる。

【0082】

STi 2:0 928またはSTI 2:0 929に対応する浮動小数点スタック選択信号は、図11のレジスタファイルアレイ662をアドレス指定するビット 5:3 をセットする。これにより、浮動小数点命令は、スタックのトップ位置に関する位置を指定することによってスタックのエントリを選択する。したがって、スタック回路920または922は、レジスタファイルアレイ662をアドレス指定するためにSTi 2:0 928またはSTI 2:0 929を引出す。下位41ビットにアクセスするためにレジスタファイルアドレスビット 8:6 を「100」にセットすることにより、および浮動小数点の数の上位41ビットにアクセスするためにレジスタファイルアドレスビット 8:6 を「110」にセットすることにより、浮動小数点オペランドはオペランドバス130、131上に駆動される。浮動小数点ROPに関して推論的执行およびフォーディングが達成されるように、浮動小数点データの従属性に関してテストするためにSTi 2:0 928またはSTI 2:0 929の信号はリオーダバッファ126に与えられる。

30

40

【0083】

1つの24ビットレジスタMAP 23:0 924内で、ルックアヘッドリマップアレイ504の8つのポインタは、一連の連結された3ビットレジスタMAP 2:0 ~ MAP 23:21 に構成される。同様に、浮動小数点リマップアレイ674の8つのポインタは、1つの24ビットレジスタMAP 23:0 924内に構成される。ルックアヘッドTOS502および浮動小数点TOS 2:0 はそれぞれ3ビットポインタTOS 2:0 926によって示される。図15および図16に示されるMAP 23:0 およびTOS 2:0 の内容は、スタックの初期状態を表わす。

【0084】

50

3ビットMAPレジスタ(MAP 2:0 ... MAP 23:21 924)におけるデータは、3ビットのリマップされたスタック信号STi 2:0 (ここで、iはスタックのトップ位置に関する8つのスタック位置0~7のうちの1つを選択する)を発生するために、8方向マルチプレクサ930に与えられる。ST0 2:0は、スタックのトップ位置におけるスタックのリマップされたエントリを識別し、TOS 2:0 926は0である。ST1 2:0は、スタックのトップ位置のエントリの後の位置のリマップされたスタックエントリを識別する。加算器932は、マルチプレクサ930においてST1 2:0を選択するために、TOS 2:0 ポインタに1を加える。ポインタiが増加すると、STi 2:0は、スタックの物理的な限界(7)を超えるポインタがより低いスタックアドレス(0)にラップするように、循環的にシーケンシャルに付加的なスタックエレメントをアドレス指定する。ST7 2:0は、TOS 2:0 926によってアドレス指定されたエレメントの前の位置にあるリマップアレイ924のエレメントである。

10

【0085】

いくつかのx86命令は、特定のスタックエレメントに作用する動作を指定する。たとえば、8つのスタックエレメントのうちのいずれも、ROPによって用いられるスタックエレメントを規定するために、命令のmodrm バイトから得られるREG2を用いて指定することができる。図16では、命令デコーダ118またはレジスタファイル124は、TOS 2:0 926とREG2との和によって指定されるリマップされたスタックエントリSTI 2:0を選択する。加算器936はポインタ値を加え、その和をマルチプレクサ934に与えて、STI 2:0 929が得られる。

20

【0086】

図17を参照して、8つのスタックエレメントi=0~7に関するSTiEMPTY944を発生するために、フル/エンプティアレイ942(EMPTY 7:0)に保持されるデータがマルチプレクサ938に与えられるエンプティ回路946が示されている。出力信号STiEMPTY944は、スタックのエレメントがいっぱいであるかまたは空であるかを指定する。STiEMPTY944は、ルックアヘッドスタックレジスタSTi 2:0 928の出力によってアドレス指定されるルックアヘッドフル/エンプティアレイEMPTY(EMPTY 7 ... EMPTY 0)のエレメントの値である。STiEMPTY944の値1は、指定された浮動小数点スタックアレイエレメントが規定されている(いっぱいである)ことを示し、値0は、スタックエレメントが規定されていない(空である)ことを示す。4つのディスパッチ位置の各々に関してルックアヘッドフル/エンプティアレイ506を与えるために、エンプティ回路946のマルチプレクサ938は命令デコーダ118において4回複製される。ルックアヘッドフル/エンプティアレイ506(EMPTY 7:0)は、ルックアヘッドスタックレジスタSTi 2:0の出力によってアドレス指定されるフル/エンプティアレイ942(EMPTY 7:0)に対応する。浮動小数点スタックレジスタSTi 2:0の出力によってアドレス指定されるフル/エンプティアレイ942(EMPTY 7:0)に対応する浮動小数点フル/エンプティアレイ806を与えるために、1つのエンプティ回路946もレジスタファイル124に含まれる。

30

40

【0087】

ポインタREG2を用いて8つのフル/エンプティアレイエレメントの各々をアドレス指定することができる。図18を参照して、フル/エンプティアレイ942(EMPTY 7:0)に保持されるデータがマルチプレクサ940に与えられてSTIEMPTY945信号を発生するエンプティ回路948が示されている。信号STI 2:0 929によって、スタックフル/エンプティアレイ942のエレメントが選択される。STIEMPTY945は、ポインタREG2によって決定されるスタック信号STI 2:0 929によってアドレス指定されるフル/エンプティアレイEMPTY(EMPTY 7 ... EMPTY 0)のエレメントの値である。ルックアヘッドフル/エンプティアレイ506を与えるために、エンプティ回路948のマルチプレクサ940は、命令デコー

50

【 0 0 9 2 】

図 2 0 (A) ~ 図 2 0 (C) は、 F A D D P および F X C H 命令がディスパッチされ、そのいずれの命令も実行される前のスタックレジスタおよびアレイの値を示している。 F A D D P は、命令デコーダ 1 1 8 によって、スタックのトップ位置の浮動小数点スタックアレイ 7 0 0 のエン트리 2 . 0 を T O S (位置 5) から 1 だけ除いた位置のスタック値 3 . 0 に加え、 T O S を (位置 5) に増分し、その和である 5 . 0 を T O S にストアする R O P シーケンスに変換される。したがって、図 2 0 (A) では、命令デコーダ 1 1 8 はルックアヘッド T O S 5 0 2 を 5 に更新してスタックポップを実現し、ルックアヘッドフル / エンプティアレイ 5 0 6 の位置 4 を 0 に設定する。

【 0 0 9 3 】

F X C H は、 T O S の記憶エレメントの内容と、 T O S から 2 つのエレメントだけ除いた指定されたスタック位置の内容との交換を命令する。プロセッサは、スタックレジスタにあるデータを交換することによってではなく、ルックアヘッドリマップアレイ 5 0 4 におけるポインタ 5 とポインタ 7 とを交換することによってこれを行なう。図 2 0 (A) では、命令デコーダ 1 1 8 は、 T O S の位置 5 のポインタと位置 7 のポインタとを交換し、 F A D D P および F X C H をディスパッチする。図 2 0 (C) は、浮動小数点 T O S 6 7 2 、リマップアレイ 6 7 4 、スタックアレイ 7 0 0 、およびフル / エンプティアレイ 6 7 6 が R O P がディスパッチされても図 1 6 の値から変わらないことを示している。

【 0 0 9 4 】

図 2 1 (A) ~ 図 2 1 (C) は、 F A D D P および F X C H R O P の実行後、および F M U L がディスパッチされた後のスタックレジスタおよびアレイを示している。図 2 1 (A) では、 F M U L はスタックを変更しないため、 F M U L がディスパッチされてもルックアヘッド T O S 5 0 2 またはフル / エンプティアレイ 5 0 6 は変化しない。同様に、交換命令 F X C H のみがリマップアレイの値を変えるため、 F M U L がディスパッチされても、ルックアヘッドリマップアレイ 5 0 4 は変化しない。図 2 1 (B) では、 F X C H の実行により、ルックアヘッドリマップアレイ 5 0 4 は分岐リマップアレイ 9 0 4 にコピーされる。図 2 1 (C) は、 F A D D P 、 F X C H または F M U L がいずれも回収されず、 R O P が回収されるまで浮動小数点 T O S 6 7 2 、リマップアレイ 6 7 4 、スタックアレイ 7 0 0 、およびフル / エンプティアレイ 6 7 6 が変化しないことを示している。

【 0 0 9 5 】

図 2 2 (A) ~ 図 2 2 (C) は、 F A D D P 、 F X C H 、 および F M U L の R O P の回収後のスタックレジスタおよびアレイを示している。 F A D D P に応答して、浮動小数点機能ユニット 1 2 2 は、スタックの前のトップ位置からの 2 . 0 と、スタックのその次の位置からの 3 . 0 とを加算し、そこにその和をストアする。浮動小数点 T O S 6 7 2 は、 5 に増分される。 F X C H の実行の際に、命令が回収されるとルックアヘッドリマップアレイ 5 0 4 は浮動小数点リマップアレイ 6 7 4 に書込まれる。 F A D D P が回収されると、ルックアヘッド T O S 5 0 2 が更新される。 F M U L は、スタックのトップ位置のエン트리 (位置 5 の 8 . 0) を T O S から 1 除いた位置のスタックエン트리 (位置 6 の 4 . 0) で乗算する。 F M U L は、 T O S の位置 5 においてその積をストアする。図 2 2 (C) では、 F M U L R O P に応答して、浮動小数点スタックアレイ 7 0 0 は、乗算の積を含む

【 0 0 9 6 】

分岐ユニット 1 2 0 がスタックアンダフローエラー等の F X C H 命令に関する問題を検出すると、分岐ユニット 1 2 0 は、再同期化状態の存在を示す状態フラグ (図示せず) をリオーダバッファ 1 2 6 に戻す。これらのフラグは、アサートされた例外状態通知を含む。リオーダバッファ 1 2 6 は、例外信号および再同期化信号 (図示せず) を分岐ユニット 1 2 0 に送ることによって、再同期化応答を開始する。分岐ユニット 1 2 0 は、フェッチ P C を F X C H 命令の位置に再送し、かつ、ルックアヘッド T O S 5 0 2 、ルックアヘッドリマップアレイ 5 0 4 、およびルックアヘッドフル / エンプティアレイ 5 0 6 を F X C H のデコードの前の状態に復元させることによって、これらの信号に応答する。この状態で

10

20

30

40

50

、ルックアヘッドTOS502およびルックアヘッドフル/エンプティアレイ506は、図20(A)に示されるようにFADDPをデコードした後の状態に対応するように更新され、ルックアヘッドリマップアレイ504は、図19(A)に示されるようにFXCHのデコードの前の状態に復元される。

【0097】

分岐ユニット120によってFXCH命令および分岐命令が誤予測されたことが発見された後に条件つき分岐命令がディスパッチされると、分岐ユニット120は命令キャッシュ116のフェッチPCを適切な命令ポイントに再送し、FXCH命令に対応する、図21(B)および図22(B)に示される分岐リマップアレイ904にストアされるアレイでルックアヘッドリマップアレイ504を書換える。

10

【0098】

プロセッサ110の機能エレメントによって例外状態が検出されると、例外が回収されたときの浮動小数点TOS672、リマップアレイ674、およびフル/エンプティアレイ676はそれぞれルックアヘッドTOS502、リマップアレイ504、およびフル/エンプティアレイ506に書込まれる。

【0099】

プロセッサ110は、複数段パイプラインとして動作する。図23は、シーケンシャル実行パイプラインに関するタイミング図である。段は、順に、フェッチ段、デコード1段、デコード2段、実行段、結果段、および回収段を含む。

【0100】

デコード1の間、推論的命令がフェッチされ、命令デコーダ118が命令をデコードし、命令が有効になる。命令デコーダ118は、STI、STIEMPTY、STi、およびSTiEMPTY(i=0~7)を含むスタック情報がデコード2の間に更新されるように、ルックアヘッドTOS502、ルックアヘッドフル/エンプティアレイ506、およびルックアヘッドリマップアレイ504を更新する。

20

【0101】

デコード2の間、命令デコーダ118の出力は有効になる。たとえば、オペランドバス130、131およびオペランドタグバス148、149はデコード2の初期段階で有効になり、レジスタファイル124およびリオーダバッファ126からのオペランドとリオーダバッファ126からのオペランドタグとがデコード2の後の方で利用可能になるように

30

【0102】

実行の間、オペランドバス130、131およびタグ148、149は有効になり、機能ユニットの保存局に与えられる。機能ユニットはROPを実行し、結果バスに関して調停する。FXCH ROPが実行されると、分岐ユニット120は現在のルックアヘッドリマップアレイ504をセーブする。分岐命令に関しては、分岐ユニット120は、ルックアヘッドTOS502およびルックアヘッドフル/エンプティアレイ506を保存する。誤予測された分岐に関しては、ルックアヘッドTOS502、ルックアヘッドフル/エンプティアレイ506、およびルックアヘッドリマップアレイ504は、分岐ユニット120によってセーブされた値から復元される。

40

結果の間、機能ユニットは結果をリオーダバッファ126および保存局に書込む。スタック交換命令結果が書込まれると、結果段の終わりのほうの段階で浮動小数点リマップアレイ674は分岐リマップアレイ904によって置き換えられる。プッシュまたはポップするROPの結果がリオーダバッファ126に書込まれた後、TOS672および浮動小数点フル/エンプティアレイ676は結果段の終わりのほうの段階で更新される。回収の間、オペランドは、リオーダバッファ126からレジスタファイル124に回収される。

【0103】

図24は、スーパースカラプロセッサにおいてスタックを制御するための方法の一部分として、命令デコーダ118によって行なわれる手順のフローチャートである。この手順は、ディスパッチウィンドウにおいてディスパッチされる4つ以下のROPの動作ごとに繰

50

返される。例示的なプロセッサ 110 では、2つ以下の浮動小数点命令または1つだけの浮動小数点命令と、2つの非浮動小数点命令とが1つのディスパッチウィンドウに置かれる。これにより、ディスパッチウィンドウにおいて、浮動小数点スタックに影響を及ぼす ROP の数は効果的に2つに制限される。命令デコーダ 118 はステップ 950 で命令をデコードし、ステップ 952 でデコードされた命令がスタックに影響を及ぼす命令であるかどうかを決定する。分岐命令等の、スタックを直接変えない命令も命令デコーダ 118 によって処理される。フローチャートを簡略化するために、図 24 にはスタックパラメータを更新する機能のみが示されている。スタック調節命令は、スタックエレメント交換 ROP と、スタックをプッシュおよびポップする ROP を含む。

【0104】

論理ステップ 954 の制御下で、ROP がスタックをプッシュまたはポップすると、命令デコーダ 118 は、ルックアヘッド TOS 502 を減分または増分することにより、およびルックアヘッドフル/エンブティアレイ 506 を更新することによりルックアヘッド TOS を更新する。ステップ 956 で、ルックアヘッド TOS 502 は、プッシュ機能の場合は減分され、ポップ機能の場合は増分される。なお、プッシュ動作またはポップ動作に関してスタックポインタを増分または減分することによって異なるスタック実現例を調節してもよい。プッシュ動作において減分され、ポップ動作において増分されるスタックが開示されたスタックの実施例と同等のものであり本発明の範囲内であることが理解される筈である。スタックのプッシュの際に特定されるルックアヘッドフル/エンブティアレイ 506 のエレメントは 1 に設定され、TOS ポインタは減分される。スタックのポップの 20
前に指定されるルックアヘッドフル/エンブティアレイ 506 のエレメントは 0 にクリアされ、TOS ポインタ 502 はポップ動作において増分される。

【0105】

論理ステップ 958 で識別されるスタックエレメント交換 ROP に関しては、命令デコーダ 118 は、ステップ 960 で、命令によって指定されるルックアヘッドリマップアレイ 504 のエレメントを交換する。

【0106】

ステップ 962 で、スタックに影響を及ぼさない ROP を含むすべての ROP が命令デコーダ 118 によって種々の機能ユニットにディスパッチされる。たとえば、分岐動作は分岐ユニット 120 にディスパッチされる。図 25 は、スーパースカラプロセッサにおいて 30
スタックを制御するための方法の第 2 の部分として、分岐ユニット 120 によって行なわれる手順のフローチャートである。分岐ユニット 120 にディスパッチされる ROP は、スタック交換命令および種々の分岐 ROP を含む。ROP は、動作識別ステップ 964 において識別される。

【0107】

論理ステップ 965 に従って命令がスタックエレメント交換命令であれば、分岐ユニット 120 はステップ 966 で STACK UNDER 識別をテストすることによってスタックアンダフローエラーが起こったかどうかを決定する。アンダフローが生じれば、ステップ 967 で再同期化の手順が管理される。スタックアンダフローが起こっていなければ、命令デコーダ 118 によって交換命令に関して更新されたルックアヘッドリマップアレイ 504 がステップ 968 でセーブされる。ルックアヘッドリマップアレイ 504 のすべての 40
エレメントは、分岐リマップアレイ 904 内のエントリに書込まれる。

【0108】

論理ステップ 970 で検出された分岐 ROP に関しては、分岐ユニット 120 はスタックパラメータをディスパッチされた分岐 ROP と関連させるために、ステップ 972 で保存局 902 にルックアヘッド TOS 502 およびルックアヘッドスタックフル/エンブティアレイ 506 をセーブする。保存局 902 は、分岐 ROP の実行を妨げる競合を解決し、ステップ 974 で ROP を発行する。ROP が発行されると、分岐ユニット 120 は分岐確認ステップ 976 を実行する。誤予測が検出されると、分岐ユニット 120 は、予測訂 50
正論理ステップ 978 に応じて、命令デコーダ 118 におけるルックアヘッド TOS 50

10

20

30

40

50

2 およびルックアヘッドスタックフル/エンブティアレイ506を、ステップ972で分岐ユニット保存局902にストアされた値に置き換えることによって、ステップ980でプロセッサ110のルックアヘッド状態を復元する。

【0109】

分岐が予測されても誤予測されても、分岐ユニット120は、ステップ982で結果情報を結果バス132のうちの一つに書込むことによって現在の分岐動作を終了する。図26は、スーパースカラプロセッサにおいてスタックを制御するための方法の第3の部分として組合された、リオーダバッファ126およびレジスタファイル124によって行なわれる手順の概略的なフローチャートである。分岐命令が終了すると結果バス132を介してリオーダバッファ126およびレジスタファイル124に戻される分岐情報は、ルックアヘッドリマップアレイ504を含む。浮動小数点機能ユニット122が実行を終了すると更新されるパラメータは、浮動小数点TOS672および浮動小数点フル/エンブティアレイ676である。リオーダバッファ126およびレジスタファイル124は、スタックをプッシュまたはポップする浮動小数点動作またはスタック交換動作が実行を終了し、かつそのオペランドが回収されると、スタックに関連するレジスタおよび位置を更新する。ROPの識別は、識別ステップ984で認識される。

10

【0110】

論理ステップ986での決定に従ってROPがスタック交換命令であれば、ステップ988でリオーダバッファ126内の浮動小数点リマップアレイ674は、分岐ユニット120における分岐リマップアレイ904からルックアヘッドリマップアレイ504に置き換えられる。同様に、論理ステップ990に従って動作がスタックプッシュまたはポップであれば、浮動小数点TOS672はそれぞれ減分または増分される。ステップ992で、スタックがプッシュされた後にTOS672によってアドレス指定される浮動小数点フル/エンブティアレイ676の要素は、プッシュが回収されると1に設定される。スタックをポップする前にTOS672によってアドレス指定される浮動小数点フル/エンブティアレイ676の要素は、スタックポップが回収されると0にクリアされる。

20

【0111】

以上の説明では、種々のブロック、回路、ポインタおよびアレイの位置を含む、スタックおよびプロセッサの多数の属性を特に特定している。スタックは、例示的に浮動小数点スタックとして実施されている。これらの属性は本発明の範囲を制限するものではなく、好ましい実施例を説明するためのものである。たとえば、種々のデータ構造の各々はプロセッサのいかなる位置に実現されてもよい。スタックは独立した汎用スタックであってもよく、または特定の機能ブロック内に配置されてもよい。スタックは、汎用スタックの呼出に回答して動作してもよく、または特定の動作が実行されているときにのみ機能してもよい。スタックは浮動小数点動作と関連していなくてもよい。スタックは、スーパースカラ以外のプロセッサに組み込まれてもよく、または、多くのパイプラインを有しかつクロックサイクルの間に種々の多くのROPを処理する能力を有するスーパースカラプロセッサに組み込まれてもよい。本発明の範囲は、前掲の特許請求の範囲およびそれと同等のものによってのみ決定される。

30

【図面の簡単な説明】

40

【図1】図2および図3の配置を示す図である。

【図2】データスタックが分布される種々の主なブロックを示すプロセッサのアーキテクチャレベルの概略ブロック図の上半分を示す図である。

【図3】データスタックが分布される種々の主なブロックを示すプロセッサのアーキテクチャレベルの概略ブロック図の下半分を示す図である。

【図4】図2および図3のプロセッサにおける浮動小数点機能ユニットの概略ブロック図である。

【図5】図2および図3のプロセッサにおいて浮動小数点スタックをサポートする機能ブロックを示すブロック図である。

【図6】スタックに関連する機能を果たす命令キャッシュのアーキテクチャレベルのブ

50

ック図である。

【図 7】図 8 および図 9 の配置を示す図である。

【図 8】スタックの機能ブロックを含む命令デコーダのアーキテクチャレベルのブロック図の左半分を示す図である。

【図 9】スタックの機能ブロックを含む命令デコーダのアーキテクチャレベルのブロック図の右半分を示す図である。

【図 10】図 2 および図 3 のプロセッサ内のレジスタファイルのアーキテクチャレベルのブロック図である。

【図 11】図 10 に示されるレジスタファイルのメモリフォーマットを示す図である。

【図 12】図 2 および図 3 のプロセッサ内のリオーダバッファのアーキテクチャレベルのブロック図である。

【図 13】図 12 のリオーダバッファ内のメモリフォーマットを表わす図である。

【図 14】スタック機能ブロックを含む分岐ユニットのアーキテクチャレベルのブロック図である。

【図 15】ルックアヘッドスタック機能ブロックの相互接続を示す命令デコーダの機能ブロックを示す図である。

【図 16】ルックアヘッドスタック機能ブロックの相互接続を示す命令デコーダの機能ブロックの図である。

【図 17】ルックアヘッドスタック機能ブロックの相互接続を示す命令デコーダの機能ブロックの図である。

【図 18】ルックアヘッドスタック機能ブロックの相互接続を示す命令デコーダの機能ブロックの図である。

【図 19】(A)、(B) および (C) は、図 2 および図 3 のプロセッサにおいてスタックを制御するためのレジスタ、アレイおよびポインタと、その 1 回目の内容を示す図である。

【図 20】(A)、(B) および (C) は、図 2 および図 3 のプロセッサにおいてスタックを制御するためのレジスタ、アレイおよびポインタと、その 2 回目の内容を示す図である。

【図 21】(A)、(B) および (C) は、図 2 および図 3 のプロセッサにおいてスタックを制御するためのレジスタ、アレイおよびポインタと、その 3 回目の内容を示す図である。

【図 22】(A)、(B) および (C) は、図 2 および図 3 のプロセッサにおいてスタックを制御するためのレジスタ、アレイおよびポインタと、その 4 回目の内容を示す図である。

【図 23】プロセッサ 110 における複数段シーケンシャル実行パイプラインに関するタイミング図である。

【図 24】組合せてスタックを制御する種々の機能ブロックにおいて行なわれる手順のフロー図である。

【図 25】組合せてスタックを制御する種々の機能ブロックにおいて行なわれる手順のフロー図である。

【図 26】組合せてスタックを制御する種々の機能ブロックにおいて行なわれる手順のフロー図である。

【符号の説明】

502 ルックアヘッドスタックポインタ
 504 リマップアレイ
 672 スタックポインタ
 674 リマップアレイ
 700 データエレメント

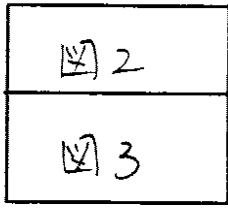
10

20

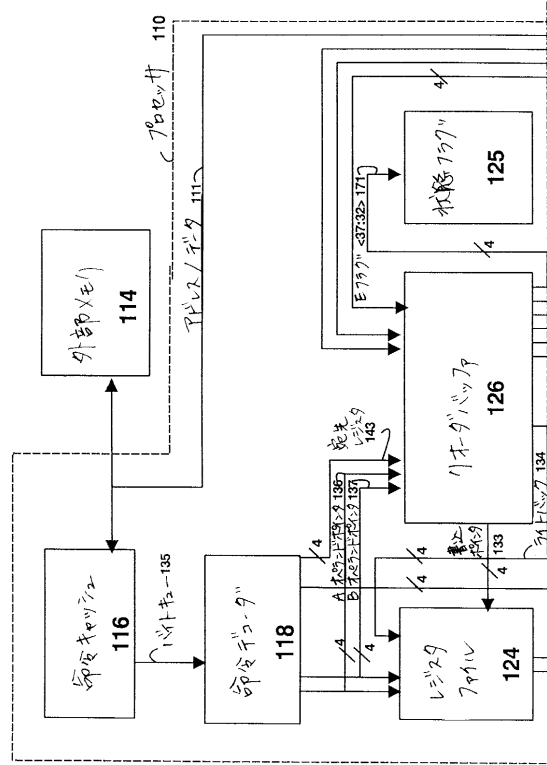
30

40

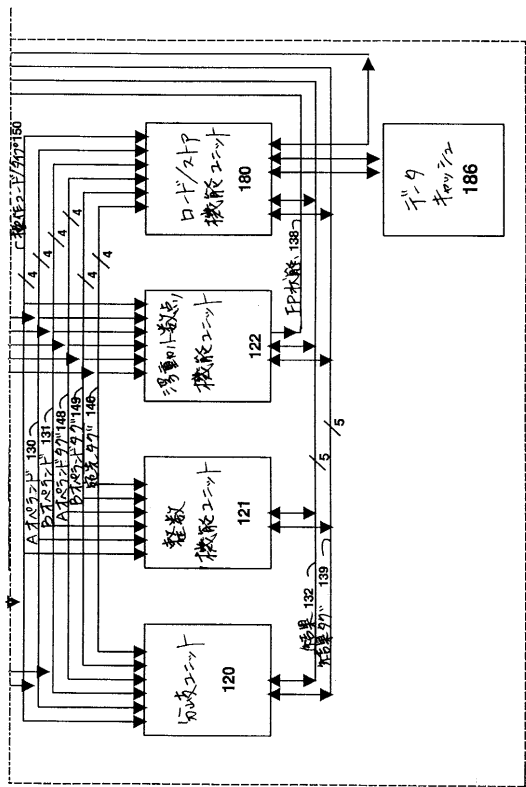
【 図 1 】



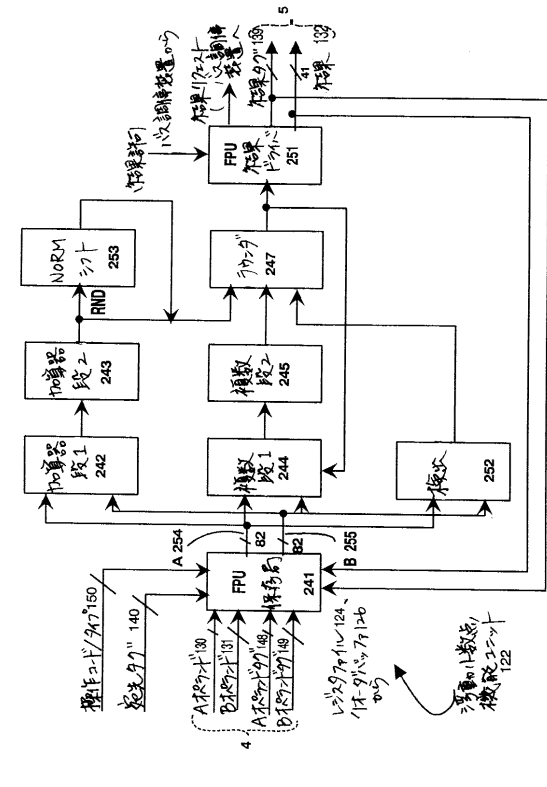
【 図 2 】



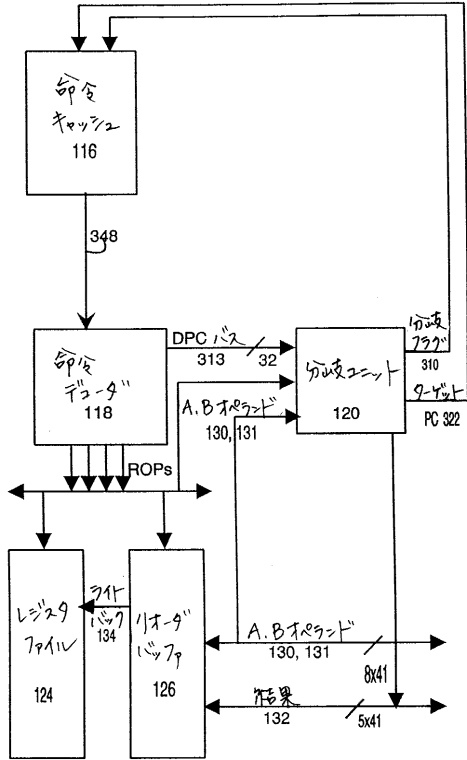
【 図 3 】



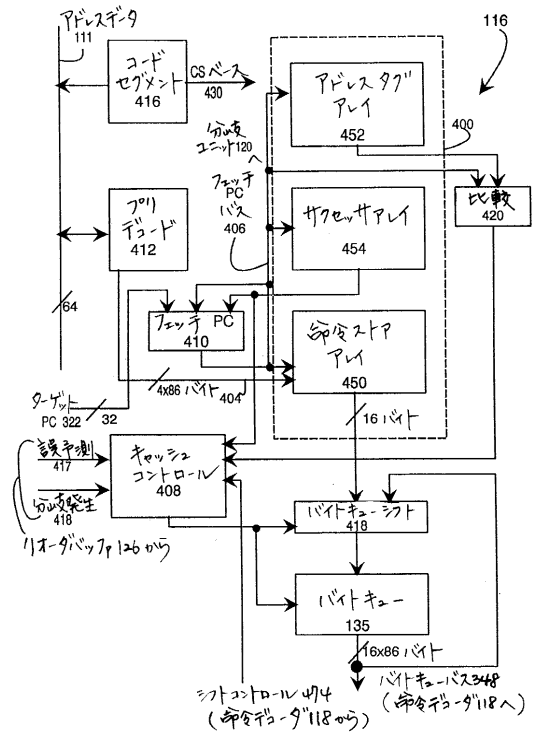
【 図 4 】



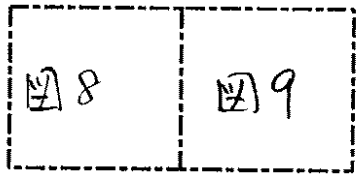
【 図 5 】



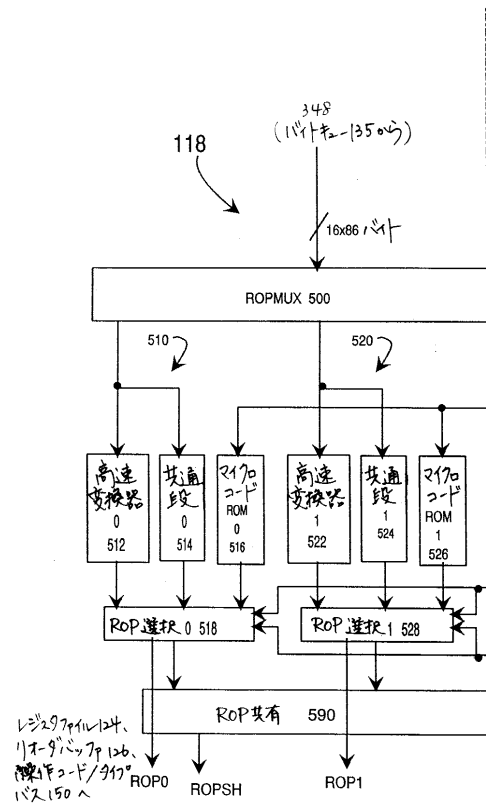
【 図 6 】



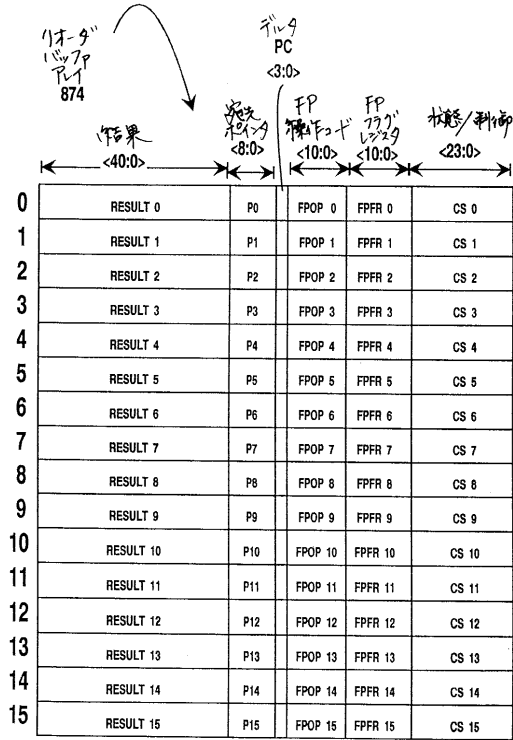
【 図 7 】



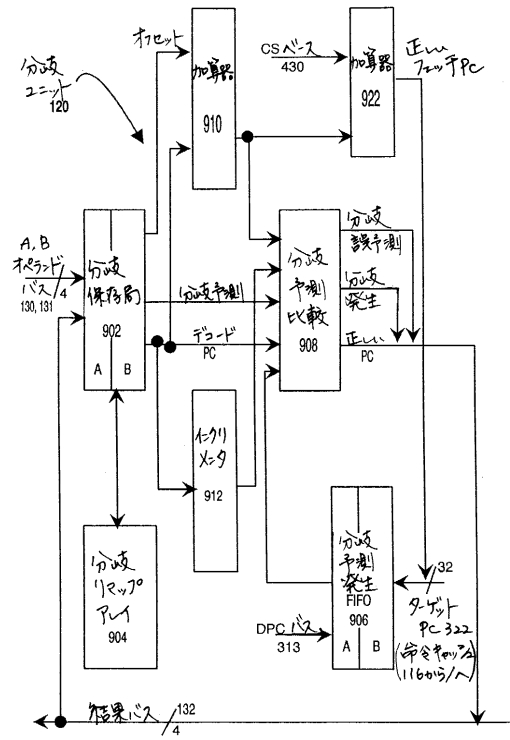
【 図 8 】



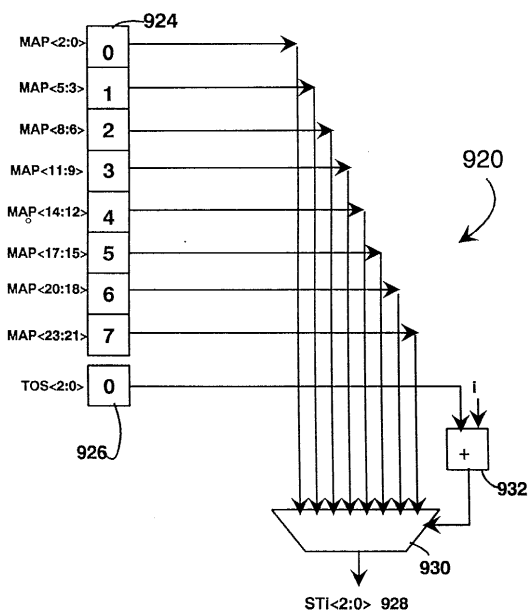
【 図 1 3 】



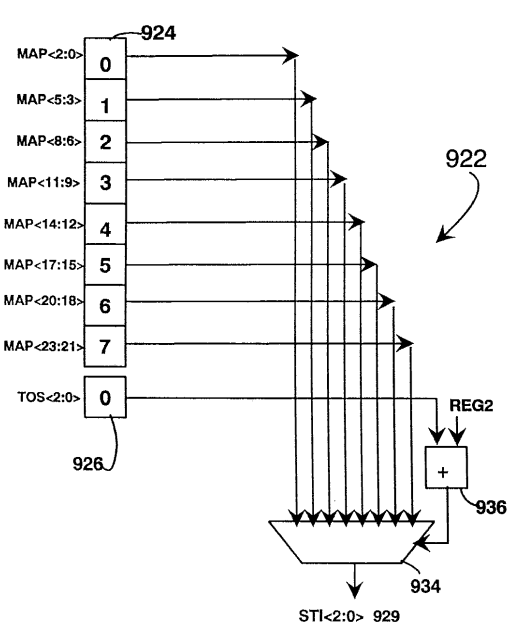
【 図 1 4 】



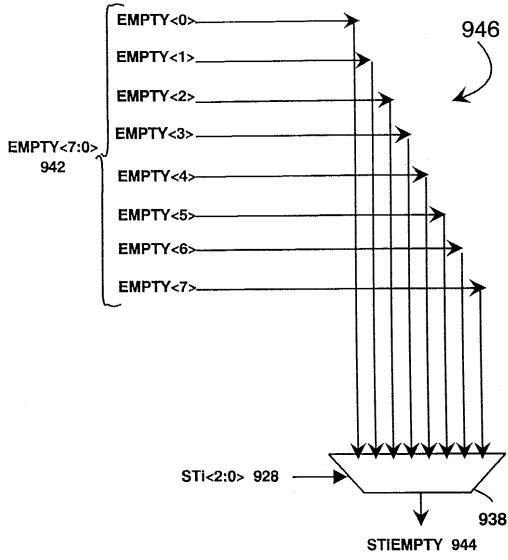
【 図 1 5 】



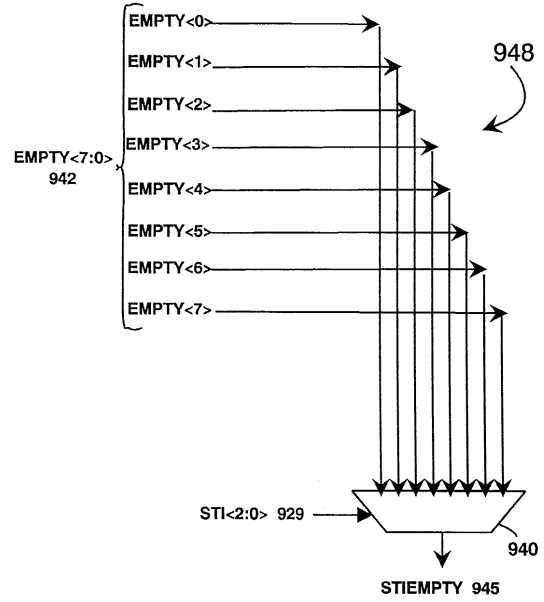
【 図 1 6 】



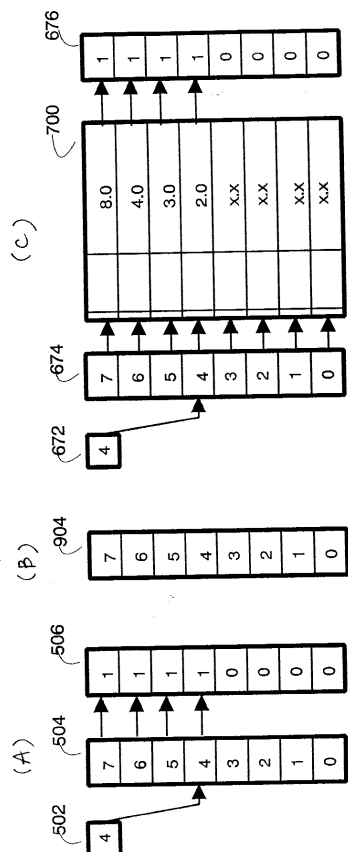
【 図 17 】



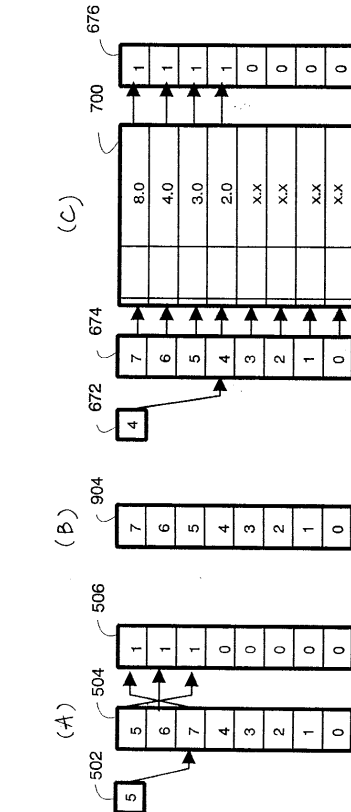
【 図 18 】



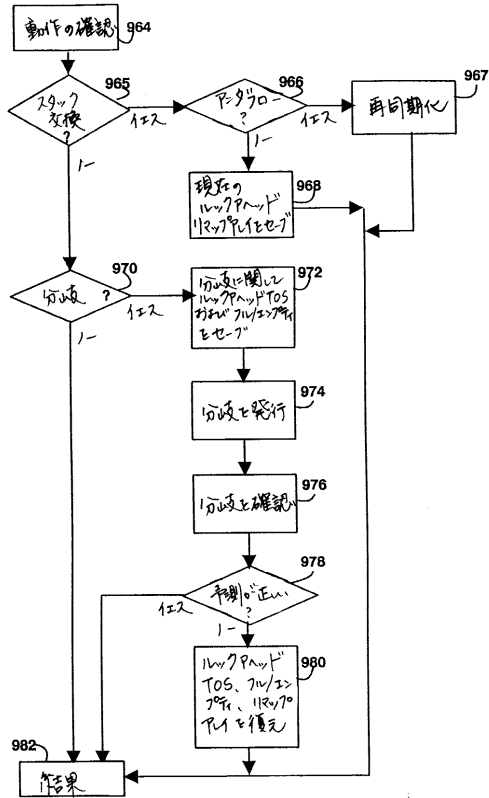
【 図 19 】



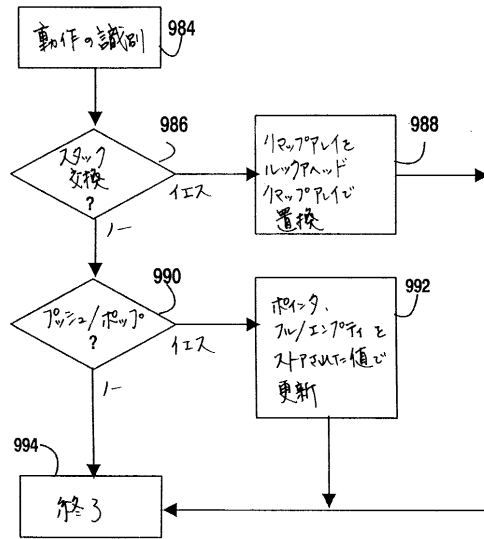
【 図 20 】



【図 25】



【図 26】



フロントページの続き

(74)代理人 100091409

弁理士 伊藤 英彦

(74)代理人 100096781

弁理士 堀井 豊

(72)発明者 マイケル・ディー・ゴッダード

アメリカ合衆国、78739 テキサス州、オースティン、オールド・ハーバー・レーン、643
4

(72)発明者 スコット・エイ・ホワイト

アメリカ合衆国、78748 テキサス州、オースティン、ペレンニアル・コート、11303

審査官 後藤 彰

(56)参考文献 特開平6 - 28185 (JP, A)

(58)調査した分野(Int.Cl.⁷, DB名)

G06F 9/38