

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3577052号
(P3577052)

(45) 発行日 平成16年10月13日(2004.10.13)

(24) 登録日 平成16年7月16日(2004.7.16)

(51) Int. Cl.⁷

G06F 9/38

F I

G06F 9/38 310X

請求項の数 10 (全 23 頁)

(21) 出願番号	特願2002-77091 (P2002-77091)	(73) 特許権者	000003078
(22) 出願日	平成14年3月19日(2002.3.19)		株式会社東芝
(65) 公開番号	特開2003-280896 (P2003-280896A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成15年10月2日(2003.10.2)	(74) 代理人	100058479
審査請求日	平成14年5月13日(2002.5.13)		弁理士 鈴江 武彦
		(74) 代理人	100084618
			弁理士 村松 貞男
		(74) 代理人	100068814
			弁理士 坪井 淳
		(74) 代理人	100092196
			弁理士 橋本 良郎
		(74) 代理人	100091351
			弁理士 河野 哲
		(74) 代理人	100088683
			弁理士 中村 誠

最終頁に続く

(54) 【発明の名称】 命令発行装置及び命令発行方法

(57) 【特許請求の範囲】

【請求項1】

命令をアウト・オブ・オーダーに投機的に発行する命令発行部と、
前記命令発行部により発行された命令と、パイプラインの各ステージにあるロード命令を含む複数の命令との直接的な依存関係を検出する第1の検出回路と、
前記第1の検出回路の出力信号と、前記ロード命令のキャッシュ・ミス信号とに基づき、
前記命令発行部により発行された命令と、前記パイプラインの各ステージにあるキャッシュ・ミスしたロード命令との間接的な依存関係を検出する第2の検出回路と
を具備することを特徴とする命令発行装置。

【請求項2】

前記第1の検出回路は、前記パイプラインステージの数と同数設けられ、命令の実行結果が順次書き込まれるデスティネーション・レジスタの番号を保持する直列接続された第1の記憶部と、
前記第1の記憶部に保持された前記デスティネーション・レジスタの番号と前記ロード命令に続く命令のソース・レジスタの番号とを比較し、前記他の命令が前記ロード命令に直接的に依存関係を有しているかどうかを判定する複数の比較器と
を具備することを特徴とする請求項1記載の命令発行装置。

【請求項3】

前記第2の検出回路は、複数のロード命令に対する依存関係を保持するため、パイプラインステージごとに依存元のロード命令に対する依存関係を保持する第1のラッチ回路と、

10

20

前記キャッシュ・ミス信号を前記パイプラインの動作に同期して保持する複数の第2のラッチ回路と、

前記第2のラッチ回路の出力信号と、前記各比較器の出力信号とに応じて、前記ロード命令に直接依存し、前記キャッシュ・ミス信号を含んだ信号を発生する複数の第1の論理回路と、

前記第1のラッチ回路の出力信号と前記比較器の出力信号、及び前記第1の論理回路の出力信号に応じて、前記ロード命令に間接的に依存する命令を検出する第2の論理回路とを具備することを特徴とする請求項2記載の命令発行装置。

【請求項4】

前記命令発行部は、前記第2の検出回路の出力信号に応じて、前記ロード命令に依存する命令を無効化することを特徴とする請求項3記載の命令発行装置。 10

【請求項5】

前記命令発行部は、キャッシュがリフィルされた後、前記無効化された命令を再発行することを特徴とする請求項4記載の命令発行装置。

【請求項6】

命令をアウト・オブ・オーダーに投機的に発行する命令発行部と、

前記命令発行部により発行された命令と、パイプラインの各ステージにあるロード命令を含む複数の命令との直接的な依存関係を検出する第1の検出回路と、

前記第1の検出回路の出力信号と、前記ロード命令のキャッシュ・ミス信号とに基づき、前記命令発行部により発行された命令と、前記パイプラインの各ステージにあるキャッシュ・ミスしたロード命令との間接的な依存関係を検出する第2の検出回路と、 20

パイプラインで実行中の命令の書き込みレジスタに有効なデータが有るか否かを示す情報を保持する第1の記憶部と、

前記第1の記憶部の出力信号に応じて、レジスタが使用可能かどうかを示す情報を記憶する第2の記憶部と、

前記第1、第2の記憶部の出力信号に応じて、前記命令発行部のソースオペランドの有効性を示す情報を更新する更新回路と

を具備することを特徴とする命令発行装置。

【請求項7】

前記第2の記憶部は、前記第2の検出回路の出力信号に応じて、キャッシュ・ミスした前記ロード命令に依存するレジスタに対応するフラグをクリアする第3の論理回路を有することを特徴とする請求項6記載の命令発行装置。 30

【請求項8】

第1の検出回路でロード命令と後続の命令との直接的な依存関係を検出し、

第2の検出回路で前記ロード命令と後続の命令との間接的な依存関係を検出し、この検出した間接的な依存関係を直接的な依存関係に変換し、

前記ロード命令にキャッシュ・ミスが発生したことを示す信号と変換された前記直接的な依存関係とにより、前記ロード命令と間接的な依存関係を有する命令を検出することを特徴とする命令発行方法。

【請求項9】 40

前記検出されたロード命令と直接的な依存関係を有する命令、及び前記検出されたロード命令と間接的な依存関係を有する命令を無効化することを特徴とする請求項8記載の命令発行方法。

【請求項10】

キャッシュがリフィルされた場合、前記無効化された命令を再発行することを特徴とする請求項9記載の命令発行方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばアウト・オブ・オーダー(Out-of-order)に命令を発行する 50

マイクロプロセッサに係わり、特に、命令スケジュールユニットに使用される命令発行装置及び命令発行方法に関する。

【0002】

【従来の技術】

マイクロプロセッサにおける命令の実行方式としてアウト・オブ・オーダー実行がある。このアウト・オブ・オーダー実行は、先行命令と依存関係の無い後続命令を順不同に実行する方式である。アウト・オブ・オーダー実行は、演算器を有効に利用することが可能であり、マイクロプロセッサを高速に動作させることができる。

【0003】

ところで、アウト・オブ・オーダーに命令を発行するマイクロプロセッサは、投機的に命令を発行し実行する。このため、ロード命令にキャッシュ・ミスが発生した場合、このロード命令にデータが依存した幾つかの命令を無効とする必要がある。この後、キャッシュ・メモリがリフィルされると、キャッシュ・ミスしたロード命令に依存関係のある命令群が再発行され、実行される。

【0004】

図14は、ロード命令に続いて発行された複数の命令とロード命令との依存関係を示している。ここで、I、R、E、Mは、パイプラインの各ステージを示し、Iは命令フェッチ、Rはレジスタ・リネーミング、Eは実行、Mはデータ・キャッシュ・アクセスである。ロード命令が発行されてから命令がオペランドを読み出すまでのレイテンシは3サイクルある。このため、ロード命令が発行された次のサイクルと、その次のサイクル(図14に示すスロット1、2)にロード命令とデータが依存のある命令を発行できないようにスケジュールされる。スロット3とスロット4では、キャッシュがヒットすると仮定してロード命令に依存関係のある命令が投機的に発行される。キャッシュ・ミスが判明するのはMステージである。このため、命令のスケジューリングによる遅延のため、スロット4の命令発行の時点において、スロット0のキャッシュ・ミスの有無を考慮できない。

【0005】

ところが、スロット0のロード命令がキャッシュ・ミスしているため、データを得ることができない。このため、スロット3とスロット4の命令は発行されたものの正しく実行できないこととなる。したがって、キャッシュ・ミスしたスロット0のロード命令と、スロット3、4にある命令を全て破棄した後、キャッシュのリフィルを行い、ロード命令を再発行する。さらに、スロット3、4にあった破棄された命令を再発行する。スロット3と4には、ロード命令とデータが依存関係のない命令が配置されることもある。この場合、命令を破棄する必要はない。しかし、依存関係の有無によって破棄するかどうかを決定することは困難である。このため、スロット3、4の命令をとにかく破棄し、後に再実行する。したがって、無駄に命令を破棄してしまい命令実行効率が低下する。

【0006】

各スロットは、複数の命令を実行できる。近時、1つのスロットにおいて、2つの整数演算命令を同時に実行できるマイクロプロセッサが開発されている。この場合、合計4命令を破棄することになる。4命令が全てロード命令に依存のない場合、全てが無駄に破棄されたことになる。

【0007】

例えば、文献「R. E. Kessler, "The Alpha 21264 Microprocessor Architecture", Proceedings International Conference on Computer Design: VLSI in Computers and Processors, 1998, ICCD '98, p 90-95」には、キャッシュ・ミスしたロード命令に依存関係のある命令群を再実行するための方法が開示されている。

【0008】

上記文献場合、ロード命令がヒットするか否かを予測し、ヒットと予測した場合のみ依存関係のある命令を発行することによって、命令を破棄する確率を下げている。しかし、口

10

20

30

40

50

ード命令がヒットと予測され、ロード命令と依存関係のない命令が発行された場合においても、実際にはロード命令がミスヒットすることがある。この場合、ロード命令と依存関係のない命令が無駄に破棄されることになる。

【 0 0 0 9 】

【 発明が解決しようとする課題 】

上記依存関係のない命令を無駄に破棄しないようにするため、スロット 3、4 の命令がロード命令に依存するかどうかを判定し、依存関係のある命令だけを破棄するようにしたとする。しかし、実際には、ロード命令以降の命令がロード命令の結果に依存するかどうかを判断するだけでは不十分である。すなわち、ロード命令に直接依存しなくとも、例えばロード命令に直接依存するスロット 3 の命令にスロット 4 の命令がさらに依存するかどうか

10

【 0 0 1 0 】

しかし、一般に、間接的に依存する命令は検出せず、投機的に発行した依存関係を有する全ての命令をキャンセルする。この場合、キャンセルしなくてもよい命令をキャンセルしてしまい、実行効率が低下する。また、多段の間接依存関係を全て検出するためには、データ・フロー・グラフをトレースする必要がある。これを実現しようとする場合、ハードウェアコストが大きくなるとともに、周波数低下の原因にもなる。

【 0 0 1 1 】

20

本発明は、上記課題を解決するためになされたものであり、その目的とするところは、ロード命令において、キャッシュ・ミスが発生した際、このロード命令と多段の依存関係を有する命令を高速に検出することが可能な命令発行装置及び命令発行方法を提供しようとするものである。

【 0 0 1 2 】

【 課題を解決するための手段 】

本発明の命令発行装置は、上記課題を解決するため、命令をアウト・オブ・オーダーに投機的に発行する命令発行部と、前記命令発行部により発行された命令と、パイプラインの各ステージにあるロード命令を含む複数の命令との直接的な依存関係を検出する第 1 の検出回路と、前記第 1 の検出回路の出力信号と、前記ロード命令のキャッシュ・ミス信号とに基づき、前記命令発行部により発行された命令と、前記パイプラインの各ステージにあるキャッシュ・ミスしたロード命令との間接的な依存関係を検出する第 2 の検出回路とを具備している。

30

【 0 0 1 3 】

本発明の命令発行方法は、第 1 の検出回路でロード命令と後続の命令との直接的な依存関係を検出し、第 2 の検出回路で前記ロード命令と後続の命令との間接的な依存関係を検出し、この検出した間接的な依存関係を直接的な依存関係に変換し、前記ロード命令にキャッシュ・ミスが発生したことを示す信号と変換された前記直接的な依存関係とにより、前記ロード命令と間接的な依存関係を有する命令を検出することを特徴とする。

【 0 0 1 4 】

40

【 発明の実施の形態 】

以下、本発明の実施の形態について図面を参照して説明する。

【 0 0 1 5 】

図 1 は、命令発行装置と実行ユニットの構成を示している。先ず、図 1 の構成について概略的に説明する。

【 0 0 1 6 】

命令発行装置は、例えば T ステージ、R ステージ、S ステージ、D ステージ、A ステージを有している。R ステージ以降の各ステージは、整数ユニット (IU) 及び浮動小数点ユニット (FPU) からなる 2 系統の回路を有している。

【 0 0 1 7 】

50

Tステージは、命令フェッチステージであり、命令をフェッチする命令フェッチユニット11を有している。命令フェッチユニット11は、例えば2命令を同時にフェッチする。

【0018】

Rステージはレジスタ・リネーミング・ステージである。Rステージは、前記命令フェッチユニット11に接続された命令デコーダ12、レジスタ・リネーミング・ユニット13a、13bを有している。レジスタ・リネーミング・ユニット13a、13bは、さらに命令デコーダ12に接続されている。命令デコーダ12は命令フェッチユニット11から供給される命令をデコードする。各レジスタ・リネーミング・ユニット13a、13bは、例えばデコードされた2命令の論理レジスタに未使用の物理レジスタをそれぞれ割り当てる。

10

【0019】

Sステージは、命令スケジューリングステージである。このSステージは、命令・ウィンドウ・バッファ(命令発行部)14a、14b、レジスタ・スコア・ボード・ユニット15a、15bを有している。命令・ウィンドウ・バッファ14aは、命令デコーダ12、レジスタ・リネーミング・ユニット13a、及びレジスタ・スコア・ボード・ユニット15aに接続されている。また、命令・ウィンドウ・バッファ14bは、命令デコーダ12、レジスタ・リネーミング・ユニット13b、及びレジスタ・スコア・ボード・ユニット15bに接続されている。

【0020】

前記レジスタ・スコア・ボード・ユニット15a、15bは、例えばフリップフロップ回路により構成され、パイプラインで実行中の命令の書き込みレジスタに有効なデータが有るか否かを示す情報(フラグ)を保持する。命令・ウィンドウ・バッファ14a、14bは、レジスタ・リネーミング後の物理レジスタ番号等を保持し、レジスタ・スコア・ボード・ユニット15a、15bからの命令ステータスに基づき、所定の条件を満足した場合、命令を発行する。命令・ウィンドウ・バッファ14aは、パイプラインI0、I1に命令を発行する。

20

【0021】

前記レジスタ・スコア・ボード・ユニット15aは、依存関係検索回路(DLC)16に接続されている。DLC16は、ロード命令に直接的、又は間接的に依存した命令を検索する。DLC16はレジスタ・スコア・ボード・ユニット15aに設けられている。これは、一般に、ロード命令が直接データを書き込むのは整数レジスタファイルだからである。しかし、命令セットによっては、浮動小数点レジスタファイルでデータを書き込むこともある。したがって、図1に破線で示すように、DLC16をレジスタ・スコア・ボード・ユニット15bに設けてもよい。

30

【0022】

命令・ウィンドウ・バッファ14a、レジスタ・スコア・ボード・ユニット15a及びDLC16の詳細については後述する。

【0023】

Dステージは、レジスタ・リード・ステージである。Dステージは、レジスタ・ファイル17a、17bを有している。レジスタ・ファイル17aは前記命令ウィンドウ・バッファ・14aに接続され、レジスタ・ファイル17bは前記命令ウィンドウ・バッファ・14bに接続されている。

40

【0024】

Aステージは、ALU演算ステージである。Aステージは、演算ユニット18、19、及び浮動小数点ユニット20を有している。演算ユニット18は、整数ユニット18a、ロード・ストア・ユニット18bを有している。演算ユニット19は、整数ユニット19a、乗算除算ユニット19を有している。整数ユニット18a、ロード・ストア・ユニット18b、整数ユニット19a、乗算除算ユニット19は、レジスタ・ファイル17aに接続されている。浮動小数点ユニット20はレジスタ・ファイル17bに接続されている。

【0025】

50

前記ロード・ストア・ユニット18bは、アウト・オブ・オーダー実行を行うプロセッサにおいて、アウト・オブ・オーダーに処理されるロード命令、ストア命令に対して、メモリを介したデータ依存関係を維持する。具体的には、ロード・ストア・ユニット18bは、メモリアクセス命令の順番を把握し、アウト・オブ・オーダーに発行されるメモリアクセス命令の順序を管理する。さらに、ロード・ストア・ユニット18bは、ロード命令の実行において、図示せぬデータ・キャッシュがミスヒットした場合、キャッシュ・ミス信号L0Miss1n(nはパイプラインのステージ)を出力する。このキャッシュ・ミス信号L0Miss1nは前記DLC16に供給される。

【0026】

図2は、本実施例のパイプラインの一例を示すダイアグラムである。各ステージの意味は、次の通りである。 10

【0027】

F：命令フェッチステージ1

I：命令フェッチステージ2

T：命令転送

R：レジスタ・リネーミング

S：命令スケジューリング

D：レジスタ・リード

A：ALU演算

W：ライト・バック 20

X：Wの次

Y：Xの次

Z：Yの次

C：完了

M：データ・キャッシュ・アクセス

図1に示す構成において、Tステージは、図2のF、I、Tステージに対応している。

【0028】

次に、図1に示す各部の動作について説明する。

【0029】

(命令フェッチ)

命令フェッチユニット11は、実行すべき2命令をフェッチする。この命令フェッチユニット11によりフェッチされた2命令は、Rステージに供給される。 30

【0030】

(レジスタ・リネーミング)

命令デコーダ12は、命令フェッチユニット11から供給される命令をデコードし、その命令がソースオペランドを必要とするか、及び演算結果はデスティネーション・レジスタに書き込まれるかを判断する。レジスタ・リネーミング・ユニット13a、13bは、命令とデコード情報に基づき、命令のソース・レジスタとデスティネーション・レジスタの論理レジスタ番号に物理レジスタ番号を割り当てる。図示せぬマッピングテーブルには、論理レジスタ番号に対してそれまでに割り当てられた物理レジスタ番号が対応して保存されている。このため、論理レジスタ番号をキーとして最後に割り当てられた物理レジスタ番号を検索することができる。ソース・レジスタを割り当てる場合、命令コードから取り出した論理レジスタ番号(Rs、Rt)をマッピングテーブルにインデックスとして入力し、物理レジスタ番号(PRs、PRt)を検索する。デスティネーション・レジスタ(Rd)を割り当てる場合、まず、未使用の物理レジスタ番号を保持しているフリーリストから未使用の物理レジスタ番号を取り出す。この物理レジスタ番号をデスティネーション・レジスタに割り当てる。さらに、割り当てた物理レジスタ番号(PRd)を論理レジスタ番号をキーとして参照できるように、マッピングテーブルに書き込む。同じ論理レジスタ番号にそれまで割り当てられていた物理レジスタ番号(マッピングテーブルにおいて上書きされた物理レジスタ番号)は、論理レジスタ番号とあわせてアクティブリストに書き 40 50

込まれる。アクティブリストは、最大で64個の命令をキューイングすることができる。アクティブリストには各エントリにインデックス番号が設けられている。このインデックス番号は、ITagとして他のユニットにおいて命令をアイデンティファイするために用いられる。

【0031】

(命令・ウィンドウ・バッファ)

図3は、前記命令・ウィンドウ・バッファ14a、14bの一例を示している。命令・ウィンドウ・バッファ14a、14bは、例えば16個のエントリを有している。各エントリは命令の古い順に並んでいる。新しい命令が命令フェッチユニット11から供給された場合、空いているエントリうち、最も古い命令が入っているエントリに近いエントリに書き込まれる。

10

【0032】

命令・ウィンドウ・バッファ14a、14bは、命令デコーダ12から供給される命令デコード情報と、レジスタ・リネーミング・ユニット13a、13bから供給される物理レジスタ番号と、命令フェッチユニット11から供給される命令コードと、命令バリッド(Valid)信号とを記憶する。すなわち、命令フェッチユニット11から出力された命令バリッド信号が“1”であるとき、命令・ウィンドウ・バッファ14a、14bは、命令コードと、物理レジスタ番号などを空いたエントリに書き込む。命令・ウィンドウ・バッファに空きエントリが無くなった場合、命令フェッチユニット11に対してフェッチ停止要求(Fetch Stall Request)信号をアサートする。

20

【0033】

命令・ウィンドウ・バッファ14a、14bは、コンプレッサ14cを有している。命令を実行ユニットに発行した後、このコンプレッサ14cは、発行された命令のエントリを無効化し、空きエントリを作成する。

【0034】

前述したように、Rステージ以降の各ステージは、整数ユニット(IU)及び浮動小数点ユニット(FPU)からなる2系統の回路を有している。しかし、以下の説明において、FPUの動作は省略し、IUの動作のみ説明する。

【0035】

図4は、命令・ウィンドウ・バッファを構成する各エントリのフォーマットを示している。図4に示す各フィールドについて簡単に説明する。

30

【0036】

ITag: 命令に対してユニークに振られた識別子であり、0~63のいずれかの値をとる。この値は、アクティブリストのエントリ番号と等しい。

【0037】

Instruction: 32bit長の命令コードそのもの。

【0038】

FU: 命令を発行すべきFunctional Unitを示すフィールド。Rステージで命令がデコードされ、命令の種類によってFU(ファンクション・ユニット)が決定される。このFUは、レジスタ・リネーミング情報とともに命令・ウィンドウ・バッファに書き込まれる。FUは4ビットにより構成されている。bit3は命令がALU命令であり、I0の整数ユニットに発行されるべきものを示す。bit2はロード・ストア・ユニットである。bit1はI1の整数ユニットに命令が発行されるべきことを示し、bit0は乗除算ユニットに命令が発行されるべきことを示している。

40

【0039】

PRs、PRt、PRf: ソースオペランドの物理レジスタ番号。

【0040】

PRd: デスティネーションの物理レジスタ番号。

【0041】

RsRdy、RtRdy、RfRdy: ソース・レジスタのPRs、PRt、PRfが使

50

用可能であることを示すフラグ。すなわち、R s、R t、R fと同一番号の物理レジスタに書き込む命令が実行を終了して演算結果が（内部パイパスや、レジスタ・ファイルを通して）使える状態になる3サイクル前にセットされる。この3サイクルは、このR d yビットを参照してから命令を発行し、さらに命令がオペランドを読み出すまでのレイテンシ分に当たる。

【0042】

Entry Rdy：グローバルなエントリReadyビットであり、何らかの理由、例えば命令がイン・オーダーに実行される場合にセットされる。また、ある時点で実行不可能な場合にはクリアされる。

【0043】

L1MissSM：ロード命令又はストア命令である場合、キャッシュ・ミス、非キャッシュ・アクセス等のステータスを保持するレジスタ。命令のキャッシュ・ミス後において、再発行（ロールバック）タイミングを決めるためのもの。

【0044】

InFlight：エントリの命令が現在実行中であることを示す。

【0045】

Rsv：エントリが次のサイクルでどのユニット（I0/I1）に発行予定であることを示す。

【0046】

Valid：有効なエントリであるかどうかを示す。

【0047】

（命令・ウィンドウ・バッファ・エントリのアップデート）

命令・ウィンドウ・バッファ14aは、各エントリをアップデートするためのアップデート回路を有している。

【0048】

図5は、命令・ウィンドウ・バッファ14aのアップデート回路21の一例を示している。図5において、図1と同一部分には同一符号を付している。

【0049】

命令・ウィンドウ・バッファ14aの各エントリには、アップデート回路21が接続されている。このアップデート回路21は、命令・ウィンドウ・バッファ14aに蓄えられた命令を、先行する命令の実行進度に応じて各種ステータスビットを更新する。すなわち、アップデート回路21には、RAT（Register Availability Table）22が接続されている。このRAT22には、レジスタ・スコア・ボード・ユニット15aが接続されている。レジスタ・スコア・ボード・ユニット15a及びRAT22は、物理レジスタ番号をキーとして参照する記憶部であり、物理レジスタが使用可能かどうかを示している。このRAT22は、データの演算が終了すると、レジスタ・スコア・ボード・ユニット15aから供給される信号に応じて、演算結果を格納する物理レジスタにフラグがセットされる。アップデート回路21は、RAT22から供給されるレジスタのステータス及びレジスタ・スコア・ボード・ユニット15aから供給される命令のステータスに基づき、エントリを毎サイクルアップデートする。

【0050】

さらに、命令・ウィンドウ・バッファ14aの各エントリには、前記DLC16が接続されている。このDLC16は、ロード・ストア・ユニット18bから出力されるキャッシュ・ミス信号に応じてロード命令に依存する命令を検索する。DLC16から出力される依存関係を示す信号Depend1Aは、レジスタ・スコア・ボード・ユニット15a、及びRAT22に供給される。DLC16から信号Depend1Aが出力された場合、レジスタ・スコア・ボード・ユニット15aの命令のステータスに基づき、依存関係のある物理レジスタに対するRAT22のエントリを無効化する。さらに、アップデート回路21は、命令・ウィンドウ・バッファ14a内において、依存関係のある物理レジスタを無効状態にリセットする。ロード命令の実行時において、キャッシュ・ミスが発生した時の

10

20

30

40

50

詳細な動作は後述する。

【0051】

(命令発行)

前述したように、本実施形態の命令発行装置は、同時に2つの命令を発行する。命令・ウィンドウ・バッファ14aの各エントリの命令は、次の条件を満たした時、発行可能な状態となる。

【0052】

(1) $R s R d y$ 、 $R t R d y$ 、 $R f R d y$ 、 $H s R d y$ 、 $E n t r y R d y$ が全てセットされている(発行許可の状態になっている)。

【0053】

(2) FUで指定される命令実行ユニット(IU0、IU1、LSU、MAC)が以前の演算を終了し、命令を受けられる状態であること。

【0054】

(3) レジスタ・ファイルのライトポートコンフリクトがないこと(レジスタ・ファイルに結果を書き込むべきタイミングの時、ライトポートが空いていること)。

【0055】

(4) $I n F l i g h t$ ビットがクリアされていること(同一命令が実行中でないこと)。

【0056】

(5) $L 1 M i s s S M$ が発行停止状態でないこと。

【0057】

図6は、上記条件を判定するためのディスパッチ決定回路31の一例を示している。このディスパッチ決定回路31は、命令・ウィンドウ・バッファ14aの各エントリに独立に設けられている。図6は、1つのエントリのディスパッチ・ロジックを示している。ディスパッチ決定回路31は、命令・ウィンドウ・バッファ14aの各エントリとレジスタ・スコアボード・ユニット15aに接続されている。ディスパッチ決定回路31は、命令・ウィンドウ・バッファ14aの各エントリとレジスタ・スコアボード・ユニット15aから供給される信号に応じて上記条件を判定する。この判定に応じてディスパッチ決定回路31は、各エントリがそれぞれどの実行ユニットに命令を発行することが可能であることを示す信号 $d i s p a t c h a b l e \ t o \ I 0$ 、 $I 1$ を出力する。

【0058】

図7は、発行可能なエントリから発行予定エントリを決定する回路の一例を示している。各エントリのディスパッチ決定回路から出力される信号 $d i s p a t c h a b l e \ t o \ I 0$ 、 $I 1$ は、プライオリティ・セクタ41の入力端に供給される。このプライオリティ・セクタ41の出力端は、アップデート回路42に供給される。

【0059】

プライオリティ・セクタ41は、複数のエントリが同時に同一実行ユニットに対して発行可能であった場合、そのうちの最も古いエントリから出力される信号 $d i s p a t c h a b l e \ t o \ I 0$ 、 $I 1$ を選択する。さらに、プライオリティ・セクタ41は、選択されたエントリに対して信号 $d i s p a t c h \ E n t X \ t o \ I Y$ ($X = 0, 1 \sim 15$)、($Y = 0, 1$) を出力する。この信号 $d i s p a t c h \ E n t X \ t o \ I Y$ ($X = 0, 1 \sim 15$)、($Y = 0, 1$) は、アップデート回路42に供給される。このアップデート回路42は、信号 $d i s p a t c h \ E n t X \ t o \ I Y$ ($X = 0, 1 \sim 15$)、($Y = 0, 1$) がアサートされたエントリに対応する $R s v$ ビットをセットする。

【0060】

(16 - 1 M u x 制御について)

図8は、命令・ウィンドウ・バッファ14aの一例を示す構成図である。図8は、16個のエントリからパイプラインI0とパイプラインI1に命令を発行する様子を示している。各エントリ0~15にはマルチプレクサ(MUX)51、52の入力端が接続されてい

10

20

30

40

50

る。マルチプレクサ51、52は、各エントリのRsvビットの内容に応じて制御される。マルチプレクサ51の出力端は、ラッチ回路53に接続され、マルチプレクサ52の出力端は、ラッチ回路54に接続されている。ラッチ回路53は、パイプラインI0に命令を発行し、ラッチ回路54は、パイプラインI1に命令を発行する。

【0061】

前述したように、命令・ウィンドウ・バッファ14aの各エントリに設けられた命令発行予定を示すRsvビットがセットされている場合、そのエントリは、次のサイクルでディスパッチされる命令である。このため、Rsv[1]がセットされている場合、マルチプレクサ52を介してパイプI0に進み、Rsv[0]がセットされている場合、マルチプレクサ51を介してパイプI1に進む。すなわち、Sステージの最後(Rsvビットが既にセットされているサイクル)において、Rsvビットの値に応じて、マルチプレクサ51、52によりパイプI0とI1それぞれに対して、16個のエントリのうち、1つのエントリが選択される。この選択されたエントリは、ラッチ回路53、54にラッチされる。ラッチ回路53、54の出力信号はレジスタ・ファイル17aを介して各演算ユニットに送られる。ラッチ回路53の出力信号は、パイプラインI0に設けられた整数ユニット18a、及びロード・ストア・ユニット18bに供給される。ラッチ回路54の出力信号は、パイプラインI1に設けられた整数ユニット19a、乗算・除算ユニット19bに供給される。各演算ユニットは、レジスタ・ファイル17aからデータを読み出し、決められた演算やメモリアクセスを行う。各演算ユニットの演算結果はレジスタ・ファイル17aに書き込まれる。

【0062】

(RATの参照と更新)

図5に示すRAT22は、前述したように、物理レジスタ番号をキーにして参照するテーブルであり、その物理レジスタが使用可能かどうかを示している。このRAT22はレジスタ・スコア・ボードロジックの一部でもある。RAT22のエントリに例えば“1”がセットされている場合、そのエントリに対応する物理レジスタのデータは既に求められており、参照可能であることを示している。また、RAT22のエントリに例えば“0”がセットされている場合、そのエントリに対応する物理レジスタのデータは参照できない。

【0063】

アップデート回路21は、命令・ウィンドウ・バッファ14aの各エントリのRs、Rt、Rfに対応するRAT22を参照する。この結果、RAT22のRs、Rt、Rfに対応するエントリに“1”がセットされている場合、RsRdy、RtRdy、RfRdyをセットする。また、アップデート回路21は、命令・ウィンドウ・バッファ14aの各エントリのRs、Rt、Rfに対応するRAT22を参照する。この結果、RAT22のRs、Rt、Rfに対応するエントリに“0”がセットされている場合、RsRdy、RtRdy、RfRdyをクリアする。

【0064】

データの依存関係をチェックするため、命令ディスパッチ時にRAT22を参照するタイミングと、実際にデータを参照する(レジスタ・ファイル17aを読み出す、あるいはデータをバイパスする)タイミングにはずれがある。このため、ある命令の実行が終了した時、その物理デスティネーション・レジスタにデータが書き込まれるよりも3サイクル早いタイミングで、その書き込みレジスタのRAT22をセットする。

【0065】

図9は、ALU命令の動作タイミングの一例を示している。図9において、RAT22はSステージでセットされる。一方、データが実際に得られるのは3サイクル後のWステージである。このため、RAT22のセットタイミングと書き込みタイミングとがずれている。

【0066】

図10は、ロード命令の動作タイミングの一例を示している。ロード命令の場合、Wステージの3サイクル前であるDステージでRAT22をセットする。

10

20

30

40

50

【 0 0 6 7 】

また、その物理レジスタを使用しなくなったとき、その物理レジスタに対応する R A T 2 2 をクリアする。すなわち、同一論理レジスタに別の物理レジスタが割り当てられ、その使用が終了した場合、以前に割り付けられていた物理レジスタを開放する。このとき、その物理レジスタに対応する R A T 2 2 をクリアする。

【 0 0 6 8 】

また、通常、R A T 2 2 は投機的に実行された命令のデスティネーション・レジスタに対しても即座に更新される。これは依存関係のある命令を最短のレイテンシで実行し、アウト・オブ・オーダーのメリットを生かすためである。しかし、分岐予測ミスや、例外が発生した場合、予測ミスした分岐命令や、例外が発生した命令が終了したイン・オーダーの時点に R A T 2 2 を戻す必要がある。例えば例外が発生した命令の次の命令は実行前で止まる必要がある。このため、その命令が書き込む物理レジスタは、R A T 内において無効にしなければならない。このような R A T を便宜上ワーキング R A T と呼ぶ。

10

【 0 0 6 9 】

ところが、実際は投機的に命令を実行している。このため、既にワーキング R A T はセットされている可能性がある。したがって、命令の実行が完了した時、例外発生や分岐予測ミスを判定し、実行が完了した時の状態を持つイン・オーダーで更新する R A T (便宜上、イン・オーダー R A T と呼ぶ) を別に 1 セット設けておく。例外発生、分岐予測ミス時、イン・オーダー R A T の内容をワーキング R A T に一括コピーする。これによって、ワーキング R A T を、分岐予測ミスや例外発生直後の状態にリストアすることができる。

20

【 0 0 7 0 】

(データ・キャッシュ・ミス時の動作)

図 1 0 に示すロード命令のタイミング図から分かるように、ロード命令のデスティネーション・レジスタ R d に対応する R A T のセットは、レイテンシを最短にするため、ロード命令の D ステージで行われる。これはロード命令のキャッシュ・ミスが判明する W ステージの 3 サイクル前である。すなわち、この 3 サイクル間はロード命令がミスするかもしれない状況にもかかわらず、ロード命令の実行結果にデータが依存した命令を発行してしまう。このような構成とすることにより、仮に、ロード命令がヒットした場合、最短のレイテンシで命令を実行することが可能となる。

30

【 0 0 7 1 】

本質的には、R A T の更新のためのサイクル、参照のためのサイクルと、ディスパッチのためのサイクルの 3 サイクルがこれに当たる。しかし、これを 0 サイクルとすることはできない。このため、そのサイクル分だけ、投機実行となる期間が確実に存在する。

【 0 0 7 2 】

キャッシュがヒットした場合、何も問題は生じない。したがって、命令の実行を続ければよい。しかし、キャッシュ・ミスが発生した場合、次のような処理を行う必要がある。すなわち、

(1) キャッシュ・ミスしたロード命令、ならびに、ロード命令に依存しスケジュールが完了し、又は実行中の命令を無効化する。

40

【 0 0 7 3 】

(2) R A T におけるロード命令のデスティネーション・レジスタ、及びロード命令に依存した命令のデスティネーション・レジスタをクリアする。

【 0 0 7 4 】

(3) 無効化された命令をキャッシュがリフィルされた後、再実行する。

【 0 0 7 5 】

上記処理を行うため、先ず、ロード命令に依存した実行中の命令と、ロード命令と無関係な命令とを区別しなければならない。また、上述したように、ロード命令は、3 サイクルの投機的実行期間がある。このため、ロード命令に直接的に依存した命令だけでなく、ロード命令に依存した第 1 の命令に、依存する第 2 の命令、さらに第 2 の命令に依存する第

50

3の命令といった3重の依存関係まで検出する必要がある。また、ある命令のソース・レジスタ R_s は第1のロード命令に依存し、ソース・レジスタ R_t は第2のロード命令に依存するというように、複数のロード命令に並列した依存関係も検出する必要がある。さらに、それらが複合した依存関係も検出する必要がある。

【0076】

図11は、上記ロード命令と他の命令との依存関係の例を示すパイプライン・ダイアグラム、及びデータ・フロー・グラフを示している。図11(a)(b)(c)に示す例は、全てキャッシュ・ミスが判明する前に命令発行しなければならないケースである。これらの例において、レジスタ番号は論理レジスタではなく物理レジスタを示すものとする。

【0077】

図11(c)に示す2並列2重間接依存の場合を例に説明する。データ・フロー・グラフにおいて、印で示すレジスタはキャッシュ・ミスが判別される前のロード命令の結果である。ロード命令に着目すると、 r_4 は r_1 に依存し、 r_7 は r_2 に依存する。さらに、 r_8 は r_4 、 r_7 に依存し、 r_{10} は r_4 に依存する。

【0078】

図11(c)において、(1)のlw(ロード)命令がキャッシュ・ミスし、(2)のlw(ロード)命令がキャッシュ・ヒットした場合、次のように処理する。

【0079】

まず、(1)のロード命令に対応する r_1 に依存するものは全て無効化する。しかし、(2)のロード命令に対応する r_2 に依存するものは有効である。このため、RATの r_4 、 r_{10} 、 r_8 を無効化する。さらに、これら r_4 、 r_{10} 、 r_8 を使用する(3)、(5)、(6)の命令を無効化し、再発行する。ただし、RATの r_7 と(4)のsub命令は無効化しない。

【0080】

上記一連の動作を実行するため、次の処理を行う。

【0081】

(1) 依存関係検索回路(DLC)16による間接依存の検出。

【0082】

(2) RATの更新。

【0083】

(3) 命令・ウィンドウ・バッファでのロールバック動作。

【0084】

(DLCによる間接依存の検出)

まず、DLC16によるロード命令と依存した命令の検出について説明する。

【0085】

図12は、DLC16の一実施例を示している。図12において、第1の検出回路16aは、ロード命令と直接的に依存したレジスタを検出する。また、第2の検出回路16bは、多段の間接的な依存関係を検出する。

【0086】

第1の検出回路16aは、パイプラインステージの数と同数のレジスタ $R_1 \sim R_6$ 、比較器 $C_1 \sim C_6$ 、 $C_{11} \sim C_{16}$ オア回路 $OR_1 \sim OR_6$ を有している。前記レジスタ $R_1 \sim R_6$ は、直列接続され、所謂シフトレジスタを構成している。これらレジスタ $R_1 \sim R_6$ は、Dステージの命令・ウィンドウ・バッファ14aから命令の実行に応じて順次出力されるデスティネーション・レジスタ(R_d)の番号を保持する。前記比較器 $C_1 \sim C_6$ の一方入力端には、命令・ウィンドウ・バッファ14aから順次出力されるソース・レジスタ(R_t)の番号が供給される。これら比較器 $C_1 \sim C_6$ の他方入力端には、前記レジスタ $R_1 \sim R_6$ の出力信号がそれぞれ供給される。また、前記比較器 $C_{11} \sim C_{16}$ の一方入力端には、命令・ウィンドウ・バッファ14aから順次出力されるソース・レジスタ(R_s)の番号が供給される。これら比較器 $C_{11} \sim C_{16}$ の他方入力端には、前記レジスタ $R_1 \sim R_6$ の出力信号がそれぞれ供給される。前記比較器 $C_1 \sim C_6$ の出力端はオア

10

20

30

40

50

回路OR1～OR6の一方入力端に供給される。前記比較器C11～C16の出力端は前記オア回路OR1～OR6の他方入力端に供給される。

【0087】

一方、第2の検出回路16bは、アンドオア回路AOR1～AOR6、アンド回路A1～A4、ラッチ回路XA、YA、ZA、ZZA、YM、ZM、ZW、L0Miss1X、L0Miss1Y、L0Miss1Z、及びオア回路OR7により構成されている。アンドオア回路AOR1～AOR6は、アンド回路とオア回路が直列接続されている。アンドオア回路AOR1～AOR6は、ロード命令に間接的に依存する命令を検出し、この検出した依存関係を直接依存にマッピングする。

【0088】

前記オア回路OR1の出力信号EqAは、アンドオア回路AOR1、AOR2、AOR3を構成するアンド回路の一方入力端に供給される。前記オア回路OR2の出力信号EqMは、アンドオア回路AOR4、AOR5を構成するアンド回路の一方入力端に供給される。前記オア回路OR3の出力信号EqWは、アンドオア回路AOR6を構成するアンド回路の一方入力端、及びアンド回路A1の一方入力端に供給される。前記オア回路OR4の出力信号EqXは、アンド回路A2の一方入力端に供給される。前記オア回路OR5の出力信号EqYは、アンド回路A3の一方入力端に供給される。前記オア回路OR6の出力信号EqZは、アンド回路A4の一方入力端に供給される。

【0089】

一方、ロード・ストア・ユニット18bから供給されるキャッシュ・ミス信号L0Miss1Wは、前記アンド回路A1の他方入力端に供給されるとともに、ラッチ回路L0Miss1Xに供給される。ラッチ回路L0Miss1Xの出力信号は、前記アンド回路A2の他方入力端に供給されるとともに、ラッチ回路L0Miss1Yに供給される。ラッチ回路L0Miss1Yの出力信号は、前記アンド回路A3の他方入力端に供給されるとともに、ラッチ回路L0Miss1Zに供給される。ラッチ回路L0Miss1Zの出力信号は、前記アンド回路A4の他方入力端に供給される。

【0090】

前記アンド回路A4、A3、A2の出力信号DDZ、DDY、DDXは、前記アンドオア回路AOR6、AOR5、AOR3を構成するオア回路の一方入力端にそれぞれ供給される。前記アンドオア回路AOR6を構成するオア回路の出力信号は、前記アンドオア回路AOR4を構成するオア回路の一方入力端に供給される。前記アンドオア回路AOR4を構成するオア回路の出力信号は、前記アンドオア回路AOR1を構成するオア回路の一方入力端に供給される。前記アンドオア回路AOR5を構成するオア回路の出力信号は、前記アンドオア回路AOR2を構成するオア回路の一方入力端に供給される。

【0091】

前記アンド回路A1の出力信号DDWは、ラッチ回路XAに供給される。前記アンドオア回路AOR1、AOR2、AOR3を構成するオア回路の出力信号は、前記ラッチ回路ZZA、ZA、YAの入力端に供給される。これらラッチ回路XA、YA、ZAの出力信号は、オア回路OR7の入力端に供給される。また、ラッチ回路XA、YA、ZAの出力信号は、前記アンドオア回路AOR3、AOR2、AOR1を構成するアンド回路の他方入力端にそれぞれ供給される。

【0092】

前記ラッチ回路XAの出力信号は、ラッチ回路YMに供給され、前記ラッチ回路YAの出力信号は、ラッチ回路ZMに供給される。前記ラッチ回路YMの出力信号はラッチ回路ZWに供給される。前記ラッチ回路ZM、YMの出力信号は、前記アンドオア回路AOR4、AOR5を構成するアンド回路の他方入力端にそれぞれ供給される。前記ラッチ回路ZWの出力信号は、前記アンドオア回路AOR6を構成するアンド回路の他方入力端に供給される。前記オア回路OR7の出力端から、後述する依存関係の有無を示す信号Depend1Aが出力される。

【0093】

10

20

30

40

50

上記構成のDL C 1 6は、次のステップで依存関係を検出する。

【0094】

(1) 物理レジスタ番号比較。

【0095】

(2) 直接依存の検出。

【0096】

(3) 間接依存を検出し、検出した間接依存を直接依存へ写像。

【0097】

(4) 依存信号の生成。

【0098】

(5) 直接依存のステージング。

【0099】

図11(c)を参照して、上記DL C 1 6の動作について説明する。図11(c)において、(1)のlw(ロード)命令がキャッシュ・ミスが発生するものとする。

【0100】

命令・ウィンドウ・バッファ14aからは、図11(c)に(1)~(6)で示す順序に従って、各命令のデスティネーション・レジスタ番号、ソース・レジスタRs、Rtの番号が出力される。デスティネーション・レジスタ番号はDL C 1 6のレジスタR1に供給される。レジスタR1に保持されたデスティネーション・レジスタ番号は、パイプラインの各ステージの実行に従ってレジスタR1~R6に順次シフトされる。また、各命令のソ

10

20

【0101】

時刻t4において、(3)のadd命令がDステージにある。このため、add命令の2つのソース・レジスタRs、Rtの番号が、実行状態(in-flight)のロード命令のデスティネーション・レジスタ番号と一致するかどうかを検索される。同時に、add命令の2つのソース・レジスタRs、Rtの番号が、実行状態のロード命令に依存する別の命令のデスティネーション・レジスタ番号と一致するかどうかを検索される。具体的には、比較器C1~C6、C11~C16により、ソース・レジスタRs、Rtの番号と、A、M、W、X、Y、Zの各ステージのデスティネーション・レジスタ番号Rdとを比

30

【0102】

すなわち、時刻t4において、Dステージのソース・レジスタRsの番号と、(1)のlw命令のWステージに対応するレジスタR3に保持されたデスティネーション・レジスタRdの番号が共にレジスタ番号“r1”である。このため、比較器C13から一致信号が出力され、オア回路OR3の出力信号EqWが“1”となる。比較器C13以外の比較器から一致信号が出力されないため、オア回路OR3以外のオア回路の出力信号は“0”となる。

【0103】

一方、キャッシュ・ミスは(1)のlw命令のWステージで判明する。このため、時刻t4において、キャッシュ・ミス信号L0Miss1Wは“1”となり、このキャッシュ・ミス信号L0Miss1Wと、オア回路OR3の出力EqWがアンド回路A1に供給される。このため、AND回路A1の出力信号DDWは“1”となる。この信号DDWは、Wステージのロード命令にDステージの命令が直接依存するかどうかを示す信号である。さらに、この信号DDWが“1”である場合、Wステージのロード命令にDステージの命令が直接依存し、且つキャッシュ・ミスが発生していることを示している。

40

【0104】

また、ラッチ回路L0Miss1Xは、前記キャッシュ・ミス信号L0Miss1Wを1サイクル遅らせた信号を保持する。このため、ラッチ回路L0Miss1Xは、Xステージのロード命令がキャッシュ・ミスしているとき“1”となる。同様に、ラッチ回路L0

50

M i s s Y、L O M i s s Zは、Yステージ、Zステージのロード命令がキャッシュ・ミスしているとき“1”となる。ラッチ回路L O M i s s 1 X、L O M i s s Y、L O M i s s Zの出力信号は、オア回路O R 4、O R 5、O R 6の出力信号E q X、E q Y、E q Zとともにアンド回路A 2、A 3、A 4にそれぞれ供給される。このため、アンド回路A 2、A 3、A 4の出力信号D D X、D D Y、D D Zが“1”である場合、Xステージ、Yステージ、及びZステージのロード命令にDステージの命令が直接依存し、且つキャッシュ・ミスが発生していることとなる。

【0105】

次に、時刻t₅において、信号D D Wが前のサイクルで“1”であったため、ラッチ回路X Aが“1”となる。ラッチ回路X Aの信号は、信号D D Wを1サイクル遅らせたこととなる。このため、ラッチ回路X A信号は、Aステージの命令がXステージのロード命令に依存することを意味している。ラッチ回路X Aの出力信号に応じてオア回路O R 7の出力信号D e p e n d 1 Aが“1”となる。信号D e p e n d 1 Aは、ラッチ回路X A、Y A、Z A、Z Z Aのオアである。このため、信号D e p e n d 1 Aは、パイプラインのXステージ、Yステージ、Zステージ、Z Zステージのいずれかのロード命令にAステージの命令が依存しており、且つそのロード命令がキャッシュ・ミスしていることを示している。ラッチ回路X A、Y A、Z A、Z Z Aは、キャッシュ・ミスの情報を含んだ信号を保持する。したがって、ラッチ回路X A、Y A、Z A、Z Z Aの出力信号は、キャッシュ・ミスの確認済みの信号となる。

【0106】

また、図11(c)に示す(2)のl w(ロード)命令と(4)のs u b命令が依存関係を有している。(2)のl w命令はキャッシュ・ヒットするとの仮定であるため、アンド回路A 1の出力信号D D Wは“0”となる。

【0107】

次に、時刻t₆において、図11(c)に示す(5)のx o r命令がDステージにある。このため、x o r命令が依存するロード命令の有無が検索される。すなわち、Dステージにあるx o r命令のソース・レジスタR s、R tの番号“r 4”“r 7”と、各ステージのレジスタR 1~R 6に保持されたデスティネーション・レジスタの番号とが比較される。この場合、Mステージのデスティネーション・レジスタの番号が(3)のa d d命令に使用されるレジスタ番号“r 4”となっている。さらに、Aステージのラッチ回路R 1に保持された(4)のs u b命令のデスティネーション・レジスタ番号は“r 7”である。このため、比較器C 1 2、C 1の出力信号が“1”となる。したがって、オア回路O R 2の出力信号E q Mが“1”となり、オア回路O R 1の出力信号E q Aが“1”となる。

【0108】

また、時刻t₆において、前記レジスタX Aの出力信号“1”が、レジスタY Mにセットされる。このため、レジスタY Mの出力信号が“1”となる。このレジスタY Mの出力信号は、オア回路O R 2の出力信号E q Mとともに、アンドオア回路A O R 5に供給される。このため、アンドオア回路A O R 5から信号“1”が出力される。この信号はアンドオア回路A O R 2を介してラッチ回路Z Aに信号Y Dとして供給される。

【0109】

さらに、前記O R回路O R 1の出力信号は、アンドオア回路A O R 1、A O R 2、A O R 3を構成するアンド回路の一方入力端に供給される。しかし、このとき、ラッチ回路X A、Y A、Z A、Z Z Aの出力信号は、いずれも“0”である。このため、アンドオア回路A O R 1、A O R 2、A O R 3を構成する各アンド回路の入力条件は成立しない。このため、Aステージにある(4)のs u b命令との間の依存関係は保持されない。この理由は、(4)のs u b命令が依存関係を有する(2)のl w命令がキャッシュ・ヒットしている。このため、時刻t₆において、ラッチ回路X Aの出力信号が“0”となっているためである。このようにして、キャッシュ・ミスが発生したロード命令に直接的、及び間接的に依存した命令を検出することができる。

【0110】

10

20

30

40

50

すなわち、第2の検出回路16bは、実行状態にある全ての命令の依存関係をA～Zステージにあるキャッシュ・ミスした全てのロード命令との間で依存関係を検出している。換言すると、第2の検出回路16bは、多段の間接的な依存関係を検出して直接依存関係に変化し、このうちのキャッシュ・ミスの場合だけ依存関係を検出する。第2の検出回路16bにより、キャッシュ・ミスしたロード命令に依存する全ての命令がどのステージに存在するかを、複雑なリストを用いることなく、直接検出することができる。

【0111】

上記説明において、ロード命令のキャッシュ・ミスは、Wステージで判明すると仮定した。しかし、ロード命令のキャッシュ・ミスがXステージ、又はYステージにおいて判明するような場合も考えられる。このような場合、投機的実行期間が長くなるため、投機命令数が増加し、間接依存の段数が増える。しかし、上記構成のDLC16を用いることにより、最小限のハードウェア構成により、直接的及び間接的な依存関係を検出することが可能である。

10

【0112】

上記のように、DLC16によりキャッシュ・ミスが発生したロード命令に依存した命令が検出されると、第2の検出回路16bを構成するOR回路OR7から依存関係の有無を示す信号Depend1Aが出力される。この信号Depend1Aは、図5に示すレジスタ・スコア・ボード・ユニット15a、及びRAT22に供給される。レジスタ・スコア・ボード・ユニット15a、及びRAT22は、信号Depend1Aに応じて内容が更新される。

20

【0113】

(キャッシュ・ミスによるRATの更新)

図13は、RAT22の更新回路22aの一例を示している。この更新回路22aは、例えば複数のアンド回路A21～A25、複数の比較器C21～C24、オア回路OR11、OR12、ノア回路NR1により構成されている。

【0114】

通常、ALU命令の最後のSステージ、又はロード命令のDステージにおいて、その命令が書き込むデスティネーション・レジスタRdに対応するRATのエントリがセットされる。これはその物理レジスタを参照する命令の発行遅延を考慮したものである。

【0115】

図13において、ALU命令の場合、比較器C21により最後のSステージにおけるデスティネーション・レジスタ(物理レジスタ)Rdの番号と、RAT22のエントリ番号(n)とを比較する。また、ロード命令の場合、比較器C22により、Dステージにおけるデスティネーション・レジスタRdの番号と、RAT22のエントリ番号とを比較する。これらデスティネーション・レジスタRdの番号とRAT22のエントリ番号が一致し、かつ有効な命令がそのステージに存在する場合はRAT22がセットされる。

30

【0116】

尚、図13は、ワーキングRATであり、分岐予測ミスを回復するためのイン・オーダーRATからのリストアと、物理レジスタが開放された場合にRATをクリアするパスは含まれていない。

40

【0117】

一方、ロード命令にキャッシュ・ミスが発生した場合において、ロード命令に依存した命令がAステージにある時、そのデスティネーション・レジスタRdの番号と、RAT22のエントリ番号とが比較器C23により比較される。この比較の結果、これらが一致し、且つ、DLC16から供給される信号Depend1Aが“1”のとき、ロード命令に依存する命令の結果を書き込むデスティネーション・レジスタに対するRAT22のフラグがクリアされる。前述したように、信号Depend1Aが“1”であることは、Aステージにある命令がロード命令に依存関係を有し、且つロード命令がキャッシュ・ミスしたことを意味する。すなわち、Aステージにある命令は、正しいソースオペランドを得ることができなくなる。したがって、この命令の実行結果も正しくないため、RAT22のそ

50

の命令のデスティネーション・レジスタのフラグをクリアする。

【0118】

また、キャッシュ・ミスしたロード命令の実行結果が供給されるデスティネーション・レジスタRdもクリアする。すなわち、ロード命令においてキャッシュ・ミスが発生した場合、Xステージにあるロード命令のデスティネーション・レジスタRdと、RAT22のエントリ番号が比較器C24により比較される。この比較の結果、両者が一致し、且つキャッシュ・ミス信号L0Miss1Xが“1”である場合、RAT22のキャッシュ・ミスしたロード命令の実行結果が供給されるデスティネーション・レジスタRdのフラグがクリアされる。

【0119】

このようにキャッシュ・ミスしたロード命令のデスティネーション・レジスタRdと、それに依存した命令のデスティネーション・レジスタRdであって、既にRAT22の対応するレジスタのエントリにセットされているフラグは全てクリアされる。

【0120】

また、RAT22のフラグをクリアすることにより、キャッシュ・ミスが判明したロード命令のXステージ以降のタイミングでは、多重間接依存を含めてRdを参照できなくなる。さらに、図5に示すアップデート回路21は、RAT22の内容に基づいて命令ウィンドウ・バッファ14aのRsRdy、RtRdy、RfRdyをクリアする。このため、キャッシュ・ミスが発生したロード命令と依存関係を有する命令は発行できなくなる。

【0121】

上記のような動作が、毎サイクル実行されることにより、キャッシュ・ミスを起こしたロード命令と直接的、及び間接的に依存するレジスタが無効化され、且つキャッシュ・ミスが発生したロード命令と依存関係を有する命令が無効される。

【0122】

(IWBでのロールバック動作)

ロード命令がキャッシュ・ミスが発生した場合、キャッシュ・ミスしたロード命令と、それに依存した全ての命令を再実行する。この動作をロールバックと呼ぶ。ここでは、そのロールバック方法について述べる。

【0123】

命令・ウィンドウ・バッファ14aから命令を発行したあと、キャッシュ・ミスの判明しない実行中のロード命令、又はストア命令と、それ以降の全ての命令を命令・ウィンドウ・バッファ14aに保持したままにする。その際、命令・ウィンドウ・バッファ14aのInFlightビットをセットしておく。ロード命令、又はストア命令は、キャッシュがヒットした場合、Xステージで命令・ウィンドウ・バッファ14aのValidビットをクリアして命令・ウィンドウ・バッファから消去する。キャッシュ・ミスが発生した場合、InFlightビットをクリアするとともにValidビットはセットしたままとする。同時にL1MissSMビットをキャッシュ・ミスの状態に変更する。キャッシュのリフィルが終了すると、L1MissSMビットが初期状態にリセットされる。この後、ロード命令、又はストア命令が再度スケジューリングされ発行される。

【0124】

一方、ロード命令に依存した命令と、ロード命令に対して間接的に依存した命令について、命令がAステージに到達した時に、信号Depend1Aが“1”である場合、間接的な依存関係も含めて依存元であるロード命令がキャッシュ・ミスしている。このため、命令・ウィンドウ・バッファからこの命令を消去せずに残しておく。また、信号Depend1Aが“0”である場合、依存するロード命令はヒットしているので命令・ウィンドウ・バッファからこの命令をクリアする。

【0125】

上記実施形態によれば、DLC16は、ロード命令に直接的に依存する命令を検出する第1の検出回路16aと、ロード命令に間接的に依存する命令を検出する第2の検出回路16bとを有し、第2の検出回路16bは、実行状態にある全ての命令とA～Zステージに

10

20

30

40

50

ある全てロード命令との間で多段の間接的な依存関係を検出し、このうち、キャッシュ・ミスが発生している場合だけ、間接的な依存関係を検出している。このため、このDLC16は、キャッシュ・ミスが発生したロード命令に対して直接的、又は間接的に依存する命令を高速に検出することができる。

【0126】

しかも、このDLC16は、キャッシュ・ミスしたロード命令に依存する全ての命令がどのステージに存在するかを、複雑なリストを用いたり、データ・フロー・グラフを全てトレースしたりすることなく、直接検出することができる。したがって、回路規模の増大を防止できる利点を有している。

【0127】

さらに、このDLC16は、キャッシュ・ミスしたロード命令に対して直接的、及び間接的に依存する命令のみ無効化している。このため、キャッシュ・ミスしたロード命令以降の全ての命令を無効化する場合に比べて、命令の無駄な無効化を防止できる。したがって、再発行する命令の数を削減できるため、命令発行効率を向上できる。

【0128】

また、DLC16の出力信号に基づき、レジスタ・スコア・ボード15a、及びRAT22の内容は毎サイクル変更されている。このため、DLC16により検出されたロード命令に依存するレジスタ、及び命令を効率よく破棄することができる。さらに、命令・ウィンドウ・バッファ14aの内容は、レジスタ・スコア・ボード15a、及びRAT22の内容に従って毎サイクル更新されている。このため、キャッシュがリフィルされた後、破

【0129】

本発明は、上記実施形態に限定されるものではなく、その他、本発明の要旨を変えない範囲において種々変形実施可能なことは勿論である。

【0130】

【発明の効果】

以上、詳述したように本発明によれば、ロード命令において、キャッシュ・ミスが発生した際、このロード命令と多段の依存関係を有する命令を高速に検出することが可能な命令発行装置を提供できる。

【図面の簡単な説明】

【図1】本発明の命令発行装置の実施形態を示す構成図。

【図2】本実施例のパイプラインの一例を示すダイアグラム。

【図3】命令・ウィンドウ・バッファの一例を示す構成図。

【図4】命令・ウィンドウ・バッファを構成する各エントリの一例を示す構成図。

【図5】命令・ウィンドウ・バッファのアップデート回路の一例を示す構成図。

【図6】ディスパッチ決定回路の一例を示す構成図。

【図7】発行予定エントリを決定する回路の一例を示す構成図。

【図8】命令・ウィンドウ・バッファの一例を示す構成図。

【図9】ALU命令の動作タイミングの一例を示す図。

【図10】ロード命令の動作タイミングの一例を示す図。

【図11】図11(a)(b)(c)は、ロード命令と他の命令との依存関係の例を示すパイプライン・ダイアグラム及びデータ・フロー・グラフ。

【図12】DLC(依存関係検索回路)の一実施例を示す回路図。

【図13】RATの更新回路の一例を示す回路図。

【図14】ロード命令に続いて発行された複数の命令との依存関係を示す図。

【符号の説明】

11...命令フェッチユニット、

12...命令デコーダ、

14a、14b...命令・ウィンドウ・バッファ、

15a、15b...レジスタ・スコア・ボード・ユニット、

10

20

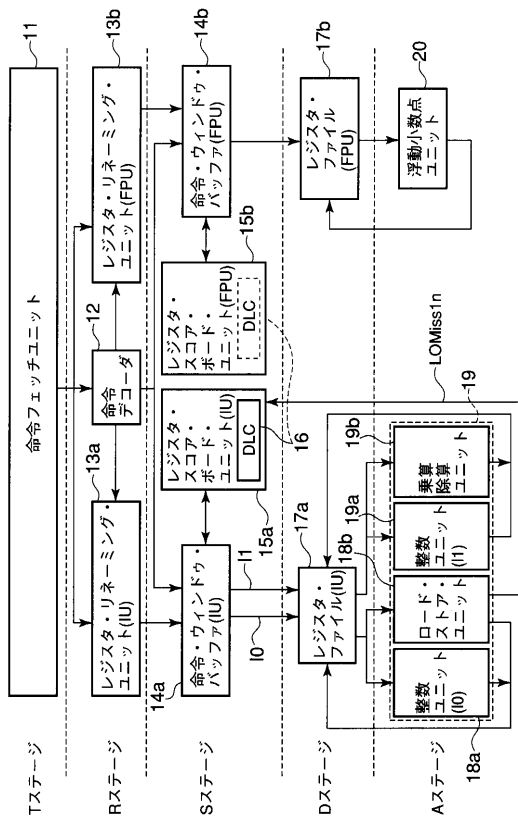
30

40

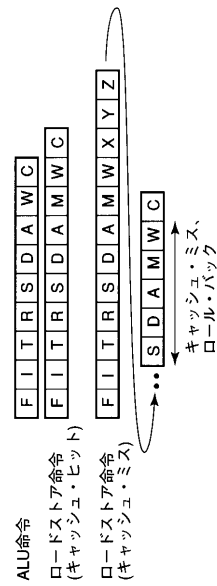
50

- 16 ... DLC、
- 16a ... 第1の検出回路、
- 16b ... 第2の検出回路、
- R1 ~ R6 ... レジスタ、
- C1 ~ C6、C11 ~ C16 ... 比較器、
- OR1 ~ OR6 ... オア回路、
- AOR1 ~ AOR6 ... アンドオア回路、
- A1 ~ A4 ... アンド回路、
- XA、YA、ZA、ZZA、YM、ZM、ZW ... ラッチ回路。

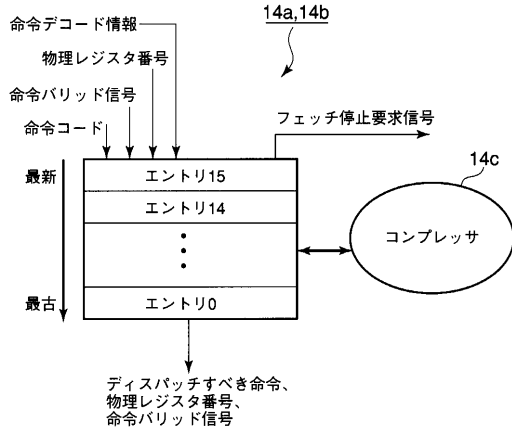
【 図 1 】



【 図 2 】



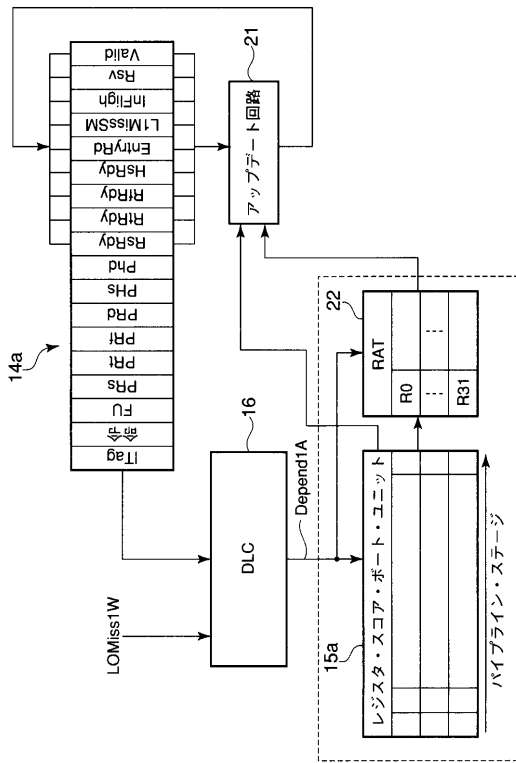
【 図 3 】



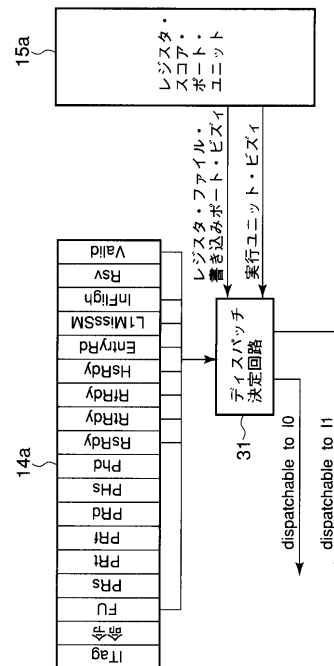
【 図 4 】

命令タグ	命令
32	Instruction
4	FU
2	フランクシヨン・ユニット(bit3:10, bit2:Ld, bit1:11, bit0:Mc)
2	命令タイプ(00:other, 01:Br, 10:St, 11:CPO)
6	PRi
6	コンディショナル・ムーブRt(flag) レジスタ番号
6	Rs物理レジスタ番号
6	Rt物理レジスタ番号
6	Rd物理レジスタ番号
3	PHs
3	PHd
1	H/L0デスティネーション・物理レジスタ番号
1	H/L0デスティネーション・物理レジスタ番号
1	物理 Rsは使用可能
1	物理 Rtは使用可能
1	物理 H/L0デスティネーションは使用可能
1	物理 H/L0デスティネーションは使用可能
1	EntRdy
2	L1MissSM
1	データ・キャッシュ・ミス状態
2	命令実行中
1	次サイクルで発行予定
1	有効エントリ

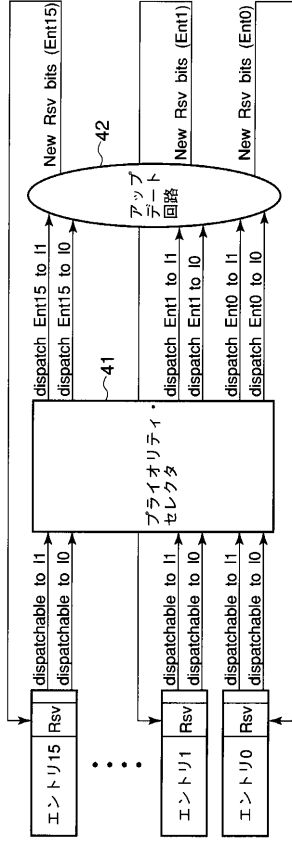
【 図 5 】



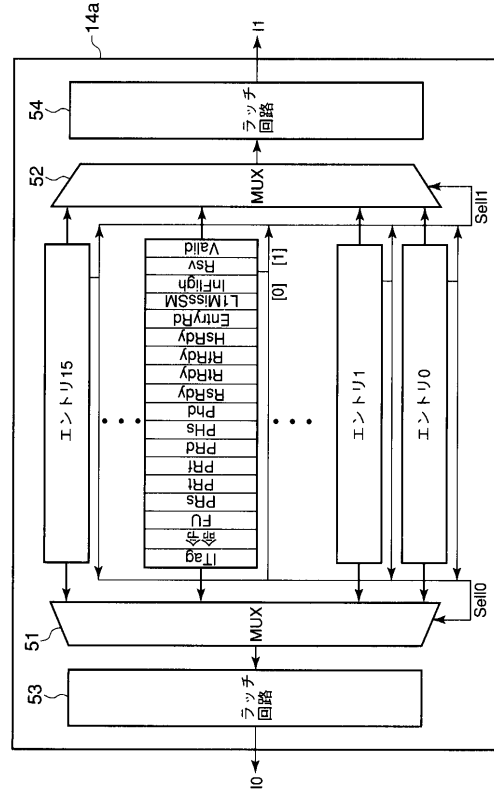
【 図 6 】



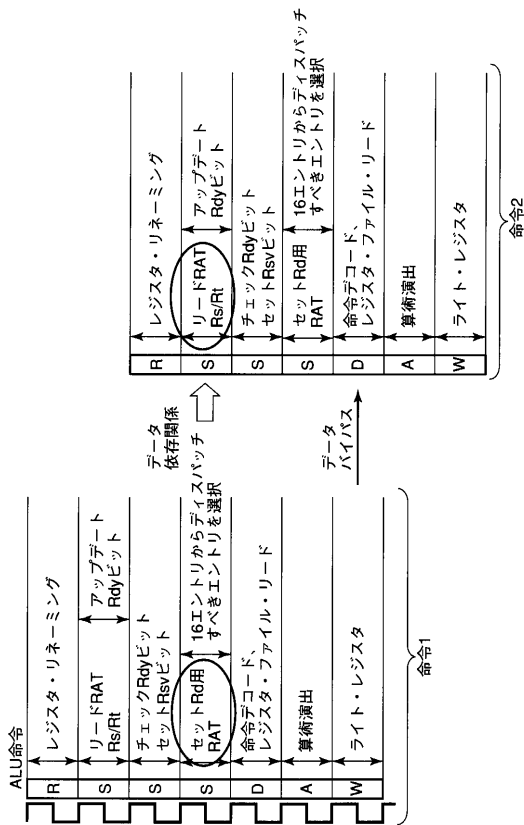
【 図 7 】



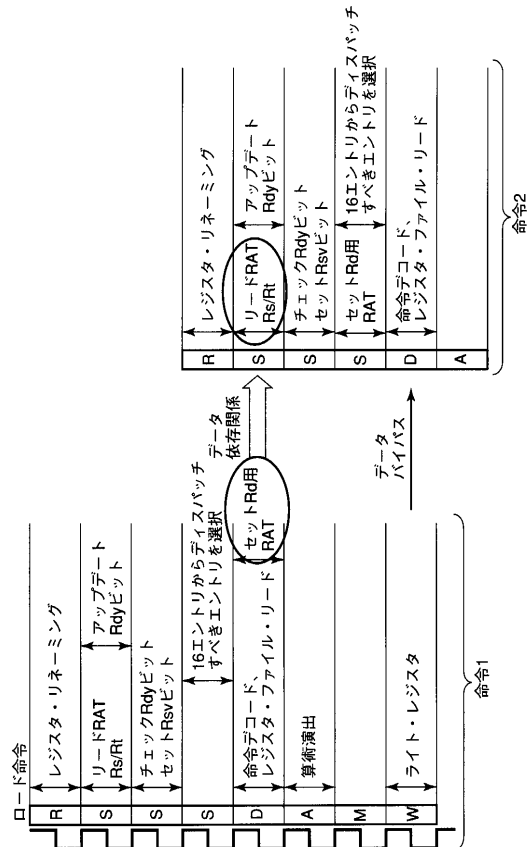
【 図 8 】



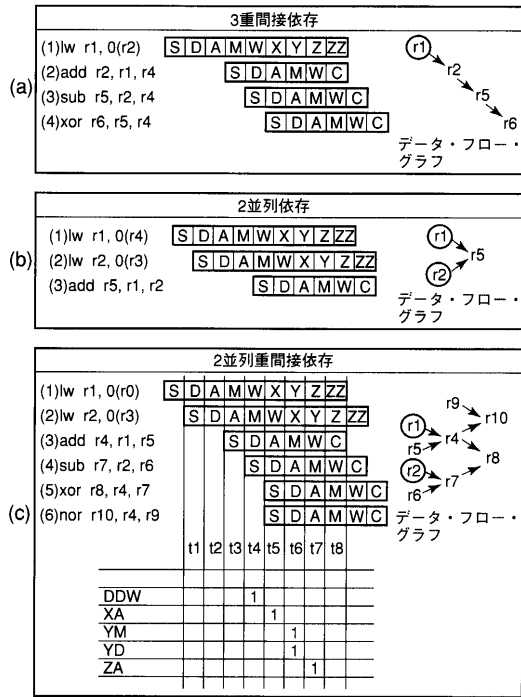
【 図 9 】



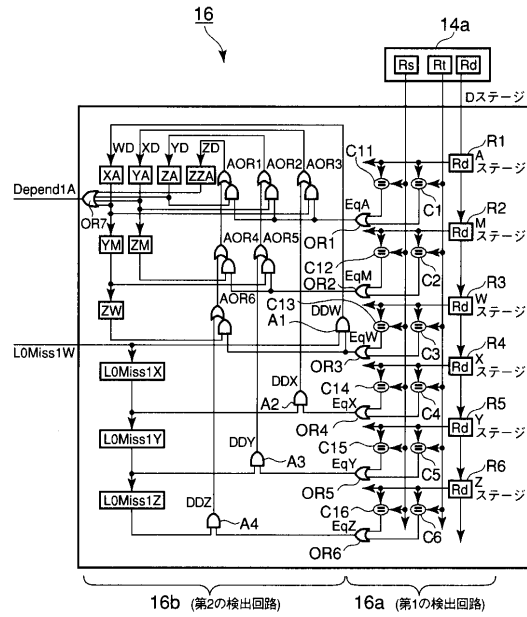
【 図 10 】



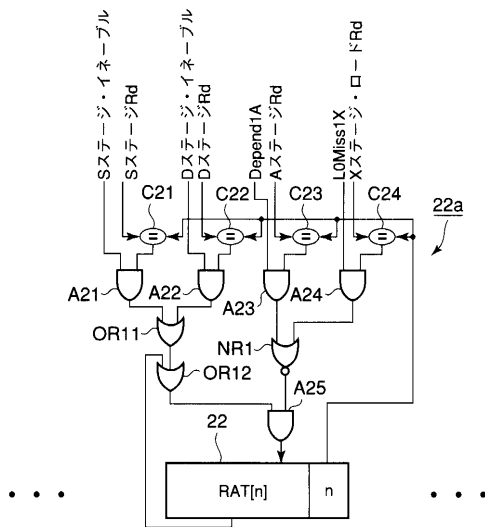
【 図 1 1 】



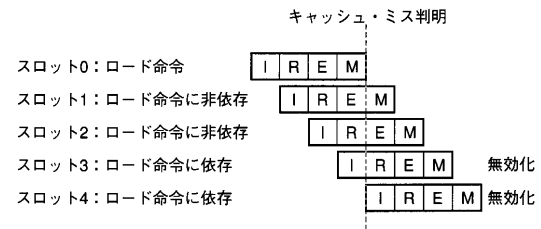
【 図 1 2 】



【 図 1 3 】



【 図 1 4 】



フロントページの続き

(74)代理人 100070437

弁理士 河井 将次

(72)発明者 照山 竜生

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝マイクロエレクトロニクスセンター内

審査官 後藤 彰

(56)参考文献 特開平6 - 67878 (JP, A)

特開平8 - 328854 (JP, A)

(58)調査した分野(Int.Cl.⁷, DB名)

G06F 9/38