

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3704046号  
(P3704046)

(45) 発行日 平成17年10月5日(2005.10.5)

(24) 登録日 平成17年7月29日(2005.7.29)

(51) Int.Cl.<sup>7</sup>

G06F 9/38

F I

G06F 9/38 350A

請求項の数 3 (全 33 頁)

(21) 出願番号	特願2001-19488 (P2001-19488)	(73) 特許権者	398038580
(22) 出願日	平成13年1月29日(2001.1.29)		ヒューレット・パカード・カンパニー
(65) 公開番号	特開2001-216159 (P2001-216159A)		HEWLETT-PACKARD COMPANY
(43) 公開日	平成13年8月10日(2001.8.10)		アメリカ合衆国カリフォルニア州パロアルト
審査請求日	平成16年4月13日(2004.4.13)		ハノーバー・ストリート 3000
(31) 優先権主張番号	09/493504	(74) 代理人	100087642
(32) 優先日	平成12年1月28日(2000.1.28)		弁理士 古谷 聡
(33) 優先権主張国	米国(US)	(74) 代理人	100063897
			弁理士 古谷 馨
		(74) 代理人	100076680
			弁理士 溝部 孝彦

最終頁に続く

(54) 【発明の名称】 データハザードを検出するために用いられるデータを融合するためのシステム及び方法

## (57) 【特許請求の範囲】

## 【請求項1】

コンピュータプログラムの命令を実行するコンピュータシステムであって、

コンピュータプログラムの前記命令を処理するように構成された複数のパイプラインと、

前記パイプラインに接続された融合回路であって、前記パイプラインから複数のレジスタ識別子を同時に受信して該複数のレジスタ識別子を単一のレジスタ識別子へと結合するよう構成されており、該単一のレジスタ識別子が、複数のビットを有すると共に、該融合回路により結合された前記複数のレジスタ識別子のうちの少なくとも1つにより識別される各レジスタを識別するものであり、該融合回路が、受信した前記複数のレジスタ識別子のうちの1つの少なくとも1つのビットを、受信した前記複数のレジスタ識別子のうちの別の1つの別のビットと結合して、前記単一のレジスタ識別子の前記複数のビットのうちの1つを生成する、融合回路と、

前記融合回路に接続されたハザード検出回路であって、前記融合回路から前記単一のレジスタ識別子を受信して該単一のレジスタ識別子を該ハザード検出回路が受信した他の情報と比較するよう構成され、更に、前記単一のレジスタ識別子と該ハザード検出回路が受信した前記他の情報との比較に基づき特定のタイプのデータハザードが存在するか否かを検出するよう構成されている、ハザード検出回路と

からなるコンピュータシステム。

## 【請求項2】

10

20

前記パイプラインに接続された複数のデコーダを更に含み、該複数のデコーダの各々が、前記複数のレジスタ識別子のうちの1つをエンコードされた形式でそれぞれ受信して該受信した前記1つのエンコードされたレジスタ識別子をデコードするように構成されており、前記融合回路が、前記デコーダを介して前記パイプライン(132)に接続されている、請求項1に記載のコンピュータシステム。

【請求項3】

コンピュータプログラムの命令を実行するコンピュータシステムであって、

前記命令を処理する手段と、

前記処理手段から第1のレジスタ識別子を受信する手段であって、前記第1のレジスタ識別子は前記命令のうちの1つに関連するとともに複数のレジスタのうちの1つを識別し、前記第1のレジスタ識別子は前記複数のレジスタのそれぞれに対応する複数のビットを有し、前記レジスタの各々は前記第1のレジスタの前記ビットの少なくとも1つにそれぞれ対応し、前記第1のレジスタ識別子の前記ビットのうちの前記第1のレジスタ識別子によって識別される前記1つのレジスタに対応するビットはアサートされ、前記第1のレジスタ識別子の前記ビットのうちの前記第1のレジスタ識別子によって識別される前記1つのレジスタ以外のいずれのレジスタに対応するビットもデアサートされる、第1のレジスタ識別子を受信する手段と、

前記処理手段から第2のレジスタ識別子を受信する手段であって、前記第2のレジスタ識別子は前記命令のうちの他の命令に関連するとともに前記複数のレジスタのうちの1つを識別し、前記第2のレジスタ識別子は前記複数のレジスタのそれぞれに対応する複数のビットを有し、前記レジスタの各々は前記第2のレジスタ識別子の前記ビットの少なくとも1つにそれぞれ対応し、前記第2のレジスタ識別子の前記ビットのうちの前記第2のレジスタ識別子によって識別される前記1つのレジスタに対応するビットはアサートされ、前記第2のレジスタ識別子によって識別される前記1つのレジスタ以外のいずれのレジスタに対応する各ビットもデアサートされる、第2のレジスタ識別子を受信する手段と、

前記第1のレジスタ識別子を前記第2のレジスタ識別子と結合させ、前記第1および第2のレジスタ識別子によって識別される前記レジスタを識別する第3のレジスタ識別子を生成する手段と、

前記第3の識別子を他のレジスタ識別子と比較して、特定種類のデータハザードが存在するか否かを検出する手段と、

からなるコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一般にデータ処理技術に関し、特に、コンピュータプログラムの複数の命令間のデータハザードを示すデータを生成し、該データを融合してデータハザードの検出に必要なとなる回路及び複雑さを最小限にするシステム及び方法に関する。

【0002】

【従来の技術】

多くのプロセッサの性能を向上させるためにパイプライン処理が開発された。パイプライン処理では、プロセッサは、同時に多数の命令を処理することができる少なくとも1つのパイプラインを備える。このため、1つのパイプラインでの1つの命令の実行が、同パイプラインでの先行する命令を実行した結果が利用可能となる前に開始される場合があり、その結果としてデータ従属性ハザードによる誤りが生じることになる。

【0003】

データ従属性は、パイプラインで実行されることになる1つの命令が他の命令の実行により生成されるデータを利用する場合に存在するものであり、該他の命令によって生成されたデータが上記1つの命令によってまだ利用可能でない場合にデータ従属性ハザードが生じることになる。例えば、後続の命令が、その実行時に、先行する命令を実行することによって生じたデータを利用する場合がある(例えば、後続の加算命令が先行するロード命

10

20

30

40

50

令によって読み出されたデータを利用する場合がある)。先行する命令の実行により生じるデータが利用可能になる前に後続の命令が実行される場合には、後続のデータは不正なデータを使用し、その結果としてデータ従属性誤りが生じる。従って、後続の命令によって用いられるデータが利用可能になるまで、又はデータ従属性誤りが生じるまで、2つの命令の間にデータ従属性ハザードが存在する。

#### 【0004】

言うまでもなく、データ従属性誤りを防ぐことができるように、データ従属性ハザードを検出することが重要である。しかし、データ従属性ハザードを検出するための回路は複雑な場合が多く、またしばしばプロセッサにおいて比較的大きな領域を利用する。これは、同時に命令を実行する複数のパイプラインを含むスーパースケラプロセッサの場合に特に当てはまる。ここで、1つのパイプライン内の命令は、同じパイプライン内の別の命令との従属性を有するのみならず、別のパイプライン内の別の命令との従属性も有する場合がある。このため、データ従属性ハザードを十分に検査するためには、1つのパイプライン内の第1の命令は、該第1の命令とデータ従属性ハザードを共有し得る各パイプライン内の各命令と比較される必要がある。その結果として、プロセッサ内のパイプラインの数が増え、データ従属性ハザードを規定するデータ従属性を検出するために必要とされる回路及び複雑さが急激に増大することになる。

#### 【0005】

##### 【発明が解決しようとする課題】

このため、これまでに取扱いわれてこなかったが、コンピュータプログラムの命令間のデータハザードを検出するための最小限の複雑さと回路を有する効率的な処理システムが当業界で必要とされている。

#### 【0006】

本発明は、これまで議論されたような従来技術の不十分な点及び不完全な点を克服するものである。本発明は一般に、命令データの融合を行ってコンピュータプログラムの命令間のデータハザードを効率的に検出する処理システム及び方法を提供する。

#### 【0007】

##### 【課題を解決するための手段】

アーキテクチャに関しては、本発明のシステムは、複数のパイプライン、融合回路、及びハザード検出回路を用いる。各パイプラインは、コンピュータプログラムの命令を受信して処理し、融合回路は、該パイプラインから複数のレジスタ識別子を受信する。各レジスタ識別子は、複数のレジスタのうちの1つを識別し、融合回路は、複数のレジスタ識別子を結合させて単一のレジスタ識別子とし、該単一のレジスタ識別子が、融合回路により受信された複数のレジスタ識別子により識別される各レジスタを識別できるようにする。次いでハザード検出回路が、前記単一のレジスタ識別子をハザード検出回路により受信された他の情報と比較し、特定のタイプのデータハザードが存在するか否かを検出する。融合回路の前記結合ステップにより、データハザードを検出するために必要とされるハザード検出回路による比較回数を低減させることができ、ハザード検出回路を実施するための回路及び複雑さも同様に低減させることができる。

#### 【0008】

本発明の別の態様によれば、パイプラインにデコーダが接続され、該デコーダを介して前記パイプラインに融合回路が接続される。各デコーダは、複数のパイプラインの1つにより送信された複数のレジスタ識別子の1つを受信しデコードする。

#### 【0009】

本発明の別の態様によれば、属性インタフェイスが、複数のパイプラインの1つに接続されて、属性データを受信する。次いで該属性インタフェイスは、前記属性データに基づき、複数のパイプラインの1つにより送信された複数のレジスタ識別子の1つにおける複数のビットのうちの少なくとも1つの値を制御する。

#### 【0010】

また、本発明は、コンピュータシステムにおいてデータハザードを検出するための方法を

10

20

30

40

50

提供するものとみなすことができる。該方法は、次のステップにより幅広く概念化することができる。即ち、コンピュータプログラムの複数の命令を処理し、該複数の命令の1つに関連する第1のレジスタ識別子であって、複数のレジスタにそれぞれ対応する複数のビットを有しており、前記複数のレジスタの各々が該第1のレジスタ識別子の前記複数のビットのうちの少なくとも1つに対応するようになっている、第1のレジスタ識別子を受信し、前記複数の命令のうちの別の命令に関連する第2のレジスタ識別子であって、前記複数のレジスタにそれぞれ対応する複数のビットを有しており、前記複数のレジスタの各々が前記第1のレジスタ識別子の前記複数のビットのうちの少なくとも1つに対応するようになっている、第2のレジスタ識別子を受信し、前記第1のレジスタ識別子において、該第1のレジスタ識別子により識別される前記複数のレジスタのうちの1つに対応する1つのビットをアサートし、前記第1のレジスタ識別子により識別される前記1つのレジスタ以外の何れかのレジスタに対応する該第1のレジスタ識別子の各ビットをデアサートし、前記第2のレジスタ識別子において、第2のレジスタ識別子により識別されるレジスタのうちの1つに対応する1つのビットをアサートし、該第2のレジスタ識別子により識別される1つのレジスタ以外の何れかのレジスタに対応する該第2のレジスタ識別子の各ビットをデアサートし、前記第1のレジスタ識別子と前記第2のレジスタ識別子とを結合させ、該結合ステップに応じて第3のレジスタ識別子を生成し、該第3のレジスタ識別子が第1及び第2のレジスタ識別子により識別されるレジスタを識別するものであり、該第3のレジスタ識別子を別のレジスタ識別子と比較し、該比較ステップに基づきデータハザードを検出する、という各ステップである。

10

20

#### 【0011】

本発明の他の特徴及び利点は、図面と併せて以下の詳細な説明を検討することにより当業者には明らかとなる。かかる特徴及び利点は全て、本書に記載する本発明の範囲内にあり、また請求の範囲により保護されるものである、ということが意図されている。

#### 【0012】

##### 【発明の実施の形態】

本発明は、図面を参照することにより更に理解を深めることができる。図面の構成要素は、互いに対して必ずしも一定の尺度となっておらず、本発明の原理を明確に説明することに重点が置かれている。更に、同様の符号は、複数の図面を通して対応する要素を示している。

30

#### 【0013】

本発明は一般に、データハザードに起因する誤りを防ぐことができるように、コンピュータプログラムの命令を実行する処理システム内のデータハザードを効率的に検出するシステム及び方法に関する。本発明の原理を示すために、図1を参照すると、従来のスーパースケラ処理システム15が示されている。処理システム15は、コンピュータプログラムの命令を受信し、各命令を複数のパイプライン21の1つに割り当てる命令分散ユニット18を備えている。各パイプライン21は、該パイプライン21が受信した各命令を実行するように構成される。

#### 【0014】

各パイプライン21は通常は、特定のタイプの命令（例えば、整数演算、浮動小数点演算、メモリ演算等）のみを処理するよう構成される。このため、命令分散ユニット18は、命令と互換性を有するパイプライン21にのみ各命令を割り当てるよう構成される。更に、単純化のため、図1では、比較ロジック22又は24が1つのパイプライン21に接続されるように示されているが、各パイプライン21は同様に比較ロジック22又は24に接続されることに留意されたい。

40

#### 【0015】

図1に示すように、パイプライン21は典型的には、複数ステージで命令を処理する。本明細書で用いる「ステージ」とは、パイプライン21の任意の部分であり、該パイプライン21は、命令を処理するものであり、また入力にラッチを備えており、これにより該ステージに入力されるデータのタイミングをクロック信号のエッジに応じて制御できるようになっ

50

ている。図1に示すパイプライン21は、4つのステージ、即ち、レジスタステージ25、実行ステージ28、例外検出ステージ32、及び書き込みステージ35で命令を処理する。別の実施形態では、処理システム15は、他のタイプ及び組み合わせのステージで命令を処理することが可能である。

#### 【0016】

図1に示すシステム15では、複数のパイプライン21のうちの1つにより受信される命令は、最初にレジスタステージ25で処理される。該レジスタステージ25では、命令を実行するために必要とされるオペランドが取得される。オペランドが取得されると、前記命令は、該命令を実行する実行ステージ28に入る。該実行ステージ28で命令が実行された後、該命令は例外検出ステージ32に入り、該例外検出ステージ32で、(例えばデータに信頼性がないことを示す)実行中のオーバーランといった状態がチェックされる。該例外検出ステージ32の終了後、命令は書き込みステージ35に入り、該書き込みステージ35で、実行ステージ28の結果が、レジスタファイル39内のレジスタ37又は特定のメモリ位置に書き込まれる。

10

#### 【0017】

典型的には、パイプライン21の各ステージ25,28,32,35は、特定の時点では1つの命令のみを処理するが、該ステージ25,28,32,35がそれぞれの命令を同時に処理して、各パイプライン21が多数の命令を処理できるようにすることも可能である。例えば、図1に示すシステム15では、複数のパイプライン21のうちの1つが4つの命令を同時に処理することが可能であり、この場合、該1つのパイプライン21の各ステージ25,28,32,35が前記4つの命令をそれぞれ1つずつ処理する。更に、各パイプライン21は、その他のパイプライン21が他の命令を処理している間に命令を処理することが可能である。よって、図1に示すシステム15を介して比較的多数の命令を同時に処理することが可能である。

20

#### 【0018】

タイミングを制御するために、命令は典型的には、クロック信号のエッジに応じてステージ25,28,32,35を段階的に進む。例えば、書き込みステージ35内の命令は、レジスタステージ25、実行ステージ28、及び例外検出ステージ32内の命令がそれぞれ同一パイプライン21の実行ステージ28、例外検出ステージ32、及び書き込みステージ35に入るのと同じクロックエッジで、パイプライン21から出ることが可能である。しかし、命令は典型的には、データ従属性誤りを生じさせることなく各ステージが該命令を完全に処理するまでステージ25,28,32,35から出ることが抑止される。

30

#### 【0019】

ここで、クロック信号の次のアクティブエッジの前にステージ25,28,32,35内の命令の処理がデータ従属性誤りを生じることなく終了することができない場合に、ステージ25,28,32,35内の命令を停止させる(stall)必要がある場合がある。例えば、例外検出ステージ32及び書き込みステージ35内の命令の処理が、データ従属性誤りを生じることなく終了し、これらのステージ32,35内の命令がそれぞれクロック信号の次のエッジで該ステージ32,35から出力される一方、それと同じパイプライン21の実行ステージ28内の命令が、次のクロックエッジが生じる前に利用することができない(例えば、先行するロード命令によって未だ読み込み中である)データを利用する場合がある。このため、実行ステージ28内の命令は次のクロックエッジで実行ステージ28から出ることができない。これは、該命令の処理が、次のクロックエッジの発生前に誤りを生じることなく終了することができないからである。換言すれば、実行ステージ28内の命令が停止される。実行ステージ28が、データ従属性誤りを生じることなく命令を完全に処理することが可能となるまで、該命令は停止されたままとなる。「Superscalar Processing System and Method for efficiently Performing In-Order Processing of Instructions」と題する1999年9月7日出願の米国特許出願第09/390,199号には、データ従属性ハザードを防ぐために命令を停止させる適切なプロセスがより詳細に記載されている。

40

#### 【0020】

各ステージ25,28,32,35は通常は、任意の特定の時点で1つの命令のみを処理するよう構

50

成されているため、先の例における同一パイプライン21のレジスタステージ25内の命令も、少なくとも実行ステージ28内の命令が実行ステージ28から出力可能となるまでレジスタステージ25内で停止されることになる。このため、ステージ25,28,32,35のうちの1つで命令が停止される場合には、該命令と同一のパイプライン21により処理されている後続の各命令もまた停止され、該停止は、該後続の命令が次のステージ25,28,32,35に入る準備が整っている場合であっても行われることになる。

#### 【0021】

命令はパイプライン21を通して段階的に進行するので、データ従属性ハザードを検出して、（通常は命令の少なくとも1つを停止させることにより）データ従属性誤りを防止できるようにすることが望ましい。データ従属性ハザードの検出は、通常は、データ従属性を検出し、該データ従属性に関連するデータがまだ利用できないことを判定することにより行われる。このため、2つの命令間にデータ従属性ハザードが存在するか否かを検出するために、通常は、1) 2つの命令間にデータ従属性が存在するか否か、2) 該データ従属性に関連するデータ（即ち、複数の命令のうちの1つにより生成されて他の命令により利用されるデータ）が利用可能であるか否かの判定が行われる。

#### 【0022】

2つの命令間のデータ従属性は、通常は、それらの命令に関連するレジスタ識別子を比較することにより検出される。ここで、各命令は通常は、該命令によりどのレジスタ37が利用されることになるかを示す少なくとも1つのレジスタ識別子を含む。例えば、データを生成する（例えば書き込む）命令（「プロデューサ」と呼ばれる）は、該プロデューサの実行時にデータが書き込まれるべきレジスタを識別するレジスタ識別子を含み、記憶されているデータ又はプロデューサにより生成されたデータを利用する（例えば取り出す）命令（「コンシューマ」と呼ばれる）は、該コンシューマの実行時にデータが取り出されるべきレジスタ37を識別するレジスタ識別子を含む。複数のパイプライン21のうちの1つにおける後続のステージ28,32,又は35が、先行するステージ25,28又は32により処理されるコンシューマと同一のレジスタ識別子を有するプロデューサを処理している場合には、それら2つの命令間にデータ従属性が存在する。更に、プロデューサにより生成されてコンシューマにより用いられるデータが、未だコンシューマに対して利用可能となっていない場合には、該2つの命令間のデータ従属性によりデータ従属性ハザードが生じることになる。

#### 【0023】

本書の目的上、1つの命令が1つのステージで処理された後に該命令が別のステージで処理される場合には、該別のステージは前記1つのステージに対して「後続の」ステージであるとする。例えば、図1では、実行ステージ28はレジスタステージ25の後続ステージであり、レジスタステージ25は実行ステージ28の先行ステージである。

#### 【0024】

ほとんどの命令は、その実行時にデータの取り出しとデータの格納との両方を行うという点で、コンシューマでもプロデューサでもあることに留意されたい。本書で用いるように、「プロデューサ」と呼ばれる命令のレジスタ識別子は、該命令が別のレジスタ37からデータを取り出す場合（したがって別のレジスタ識別子と関連する場合）であっても、該命令がデータを格納することになるレジスタ37を識別する。更に、「コンシューマ」と呼ばれる命令のレジスタ識別子は、該命令が別のレジスタ37にデータを格納する場合（したがって別のレジスタ識別子と関連する場合）であっても、該命令がデータを取り出すことになるレジスタ37を識別する。

#### 【0025】

コンシューマのためのオペランドがレジスタステージ25で取得されるため、通常は該レジスタステージ25内の各コンシューマのレジスタ識別子が後続のステージ28,32,35内の各プロデューサのレジスタ識別子と比較されて、データ従属性ハザードを生じさせるデータ従属性が該レジスタステージ25内のコンシューマとの間に存在するか否かが判定される。かかるレジスタ識別子の比較を可能にするために、各命令に関連するレジスタ識別子を、該

10

20

30

40

50

命令と共にパイプライン21を通過させる場合がある。

【0026】

ここで、レジスタ識別子（通常は $n$ ビットのエンコードされた値）が最初に、図2の従来のシステムにより示すように、デコーダ42により $m$ ビット値（ $m$ は通常は $2^n$ ）へとデコードされる。値 $m$ は、システム15に関連するレジスタ37の数に対応し、 $m$ ビットレジスタ識別子の各ビットは、特定のレジスタ識別子37に対応する。デコーダ42は、 $n$ ビットレジスタ識別子によりどのレジスタ37が識別されるかを判定し、その識別されたレジスタ37に対応する $m$ ビットレジスタ識別子内のビットをアサートするよう構成される。 $m$ ビットレジスタ識別子の残りのビットはデアサートされる。このため、複数のレジスタのうちの1つのレジスタ37に関連するビットがアサートされた場合、 $m$ ビットレジスタ識別子は該1つのレジスタ37を識別する。

10

【0027】

$m$ ビットレジスタ識別子は、該レジスタ識別子に関連する命令がラッチ56,58,61,63によりステージ25,28,32,35へラッチされるのと同じクロックエッジで、ラッチ44,46,48,52によりステージ25,28,32,35へそれぞれラッチされる。このため、レジスタ識別子及びそれに関連する命令は、同時に同ステージ25,28,32,35内に存在することになる。それらのレジスタ識別子に関連する命令がステージ25,28,32,35にそれぞれラッチされると、図2に示すように、それらの命令が処理回路66,69,72,75によりそれぞれ処理される。処理回路66,69,72,75は、ステージ25,28,32,35に関して上述した機能をそれぞれ実行する。ここで、処理回路66はオペランドを取得し、処理回路69は命令を実行し、処理回路72は例外を検査し、処理回路75は、命令の実行により生成されたデータをレジスタ37又はメモリ内の特定の位置に書き込む。

20

【0028】

ステージ25,28,32,35内の $m$ ビットレジスタ識別子は、それら $m$ ビットレジスタ識別子に関連する何れかの命令間にデータ従属性が存在するか否かを判定するために、 $m$ ビットレジスタ識別子を比較するよう構成された比較ロジック22へ伝送される。ここで、レジスタステージ25内のコンシューマが、後続のステージ28,32,35の何れかのプロデューサとのデータ従属性を有するか否かを判定するために、比較ロジック22は、レジスタステージ25内のコンシューマの $m$ ビットレジスタ識別子を、実行ステージ28、例外検出ステージ32、及び書き込みステージ35内のプロデューサの各 $m$ ビットレジスタ識別子と比較する。ステージ25,28又は32内のプロデューサの比較された $m$ ビットレジスタ識別子の何れかが、レジスタステージ25内のコンシューマの $m$ ビットレジスタ識別子と一致する場合には、比較ロジック22は、該2つの一致したレジスタ識別子に関連する命令間にデータ従属性が存在するものと判定する。更に、ステージ28,32,35内の命令のうちの1つにより生成されたデータであってレジスタステージ25内のコンシューマとのデータ従属性を有するデータが、未だレジスタステージ25内のコンシューマにより用いることができない場合には、比較ロジック22は、該2つの命令間にデータ従属性ハザードが存在するものと判定する。

30

【0029】

図3は、パイプライン21の1つにおいて命令を処理し、及びデータ従属性ハザードを生じさせるデータ従属性を検査するために、従来から用いられている回路の別の実施形態を示している。 $n$ ビットのエンコードされたレジスタ識別子に関連する命令が、ステージ25,28,32,35内にラッチされ、上述のように処理回路66,69,72,75により処理される。しかし、該 $n$ ビットレジスタ識別子がデコーダ42（図2）によりデコードされることはない。代わりに、該 $n$ ビットレジスタ識別子は、ステージ25,28,32,35に命令がそれぞれラッチされるのと同じクロックエッジで、ラッチ87,89,91,93によりステージ25,28,32,35にそれぞれラッチされる。このため、該 $n$ ビットレジスタ識別子は、該 $n$ ビットレジスタ識別子と関連する命令と共にステージ25,28,32,35を通過する。

40

【0030】

各ステージ25,28,32,35内の $n$ ビットレジスタ識別子は、比較ロジック24へと伝送される。該比較ロジック24は、該 $n$ ビットレジスタ識別子に関連する命令間にデータ従属性ハザ

50

ードを明示するデータ従属性が存在するか否かを判定するためにレジスタ識別子を比較するよう構成されている。ここで、レジスタステージ25内のコンシューマが、後続のステージ28,32及び/又は35内のプロデューサの何れかとのデータ従属性を有するか否かを判定するために、比較ロジック24は、レジスタステージ25内のコンシューマのnビットレジスタ識別子を、実行ステージ28、例外検出ステージ32、及び書き込みステージ35内のプロデューサの各nビットレジスタ識別子と比較する。nビットレジスタ識別子がエンコードされているため、比較ロジック24は、nビットコンパレータを用いて、レジスタステージ25内のレジスタ識別子がステージ28,32及び/又は35内のレジスタ識別子の何れかと一致するか否かを判定する。ステージ28,32又は35の何れかにおけるプロデューサの比較されたnビットレジスタ識別子の何れかが、レジスタステージ25内のコンシューマのnビットレジスタ識別子と一致する場合には、比較ロジック24は、該2つの一致するレジスタ識別子と関連する命令間にデータ従属性が存在するものと判定する。更に、ステージ28,32又は35内の命令のうちの1つにより生成されたデータであってレジスタステージ25内のコンシューマとデータ従属性を有するデータが、未だレジスタステージ25内のコンシューマにより用いることができない場合には、比較ロジック24は、該2つの命令間にデータ従属性ハザードが存在するものと判定する。

10

#### 【0031】

単純化のため、図2及び図3に示す回路は、ステージ25,28,32,35を通して1つのパイプライン21内の命令のレジスタ識別子のみを進行させるものである点に留意されたい。更に、上述の回路は、パイプライン21により処理される各命令毎に1つのレジスタ識別子しか処理しないものである。多くの命令は、2つ以上のレジスタ識別子を含むことが可能であるため、同一命令について多数のレジスタ識別子を取り扱うために必要に応じて更に多くの回路を実施すべきである。更に、レジスタステージ25内のコンシューマのレジスタ識別子は、同一パイプライン21内のステージ28,32,35のプロデューサのレジスタ識別子と比較されるだけでなく、他のパイプライン21の後続のステージ28,32及び/又は35の何れかの各プロデューサのレジスタ識別子とも比較されるべきである。このため、パイプライン21の数が増えると、データ従属性ハザードを検出するための配線及び他の回路が急激に増加することになる。

20

#### 【0032】

特に、プロデューサにより生成されたデータがコンシューマにとって利用可能となるのに要する待ち時間の長さが均一でも一定でもないとする、データ従属性ハザードを検出すると共にデータ従属性誤りを防止するのに必要となる回路は、比較的複雑なものとなる。更に図2において、より高性能の処理システム15の場合には、レジスタ37の数(m)は通常は比較的大きなものとなる。このため、mビットレジスタ識別子を各ステージ25,28,32,35にラッチするために(即ち、ステージ間でmビットレジスタ識別子を伝送するために)必要となる配線の数も比較的多くなる(例えば数百本)。この更に増える配線は、処理システム15内の貴重な空間を浪費し、システム15の全体的な配線の設計をより一層複雑化させるものとなる。

30

#### 【0033】

更に、図3におけるエンコードされたレジスタ識別子を比較するためのnビットコンパレータ(該nビットコンパレータ用の配線も含む)もまた、処理システム15内の貴重な空間を浪費し、システム15の全体的な配線の設計をより一層複雑化させるものとなる。更に、ステージ25,28,32,35を通してmビットレジスタ識別子及び該レジスタ識別子と関連する命令を個別にラッチすることにより、ラッチの数が増大し、このためシステム15を実施するために必要となる回路及び空間の量も増大する。複雑さ及び空間に関連する問題は、パイプライン21の数が増加すると共に悪化する。

40

#### 【0034】

一般に、本発明は、データ従属性ハザードを効率的に検出するシステム及び方法を提供する。図4は、本発明を実施するために用いることが可能な処理システム100を示している。同図に示すように、該処理システム100は、メモリ109に格納されているコンピュータブ

50



ログラム107からの命令を実行するためにコンピュータシステム105内で使用することが可能である。

【0035】

処理システム100は、ローカルインタフェイス112を介してシステム105内の他の要素との通信を行い、及び該要素を駆動する。ローカルインタフェイス112は、1つ又は2つ以上のバスを備えることができる。更に、入力装置114、例えばキーボード又はマウスを用いて、システム105のユーザからデータを入力することができ、画面ディスプレイ116又はプリンタ118を用いてユーザにデータを出力することができる。ディスク記憶機構112がローカルインタフェイス112に接続され、不揮発性ディスク（例えば磁気ディスクや光ディスク）との間でデータを送受信することができる。システム105は、該システム105がネットワーク125とデータを交換できるようにするネットワークインタフェイス123に接続することができる。

10

【0036】

データ従属性、ひいてはデータ従属性ハザードを検出するための回路を除き、処理システム100の構成は、従来の処理システム15の構成と同一であることが好ましい。このため、図5に示すように、処理システム100は、従来のシステム15に関して上述したように、レジスタステージ133、実行ステージ136、例外検出ステージ139、及び書き込みステージ142において、パイプライン132を介して命令を処理する。所望により、他のステージ及び他のステージの組み合わせを介して、パイプライン132により実行される処理を分割することが可能である点に留意されたい。更に図5では、比較ロジック144がデコーダ155,157,159,161を介して1つのパイプライン132に接続されたものとして単純化して示されているが、各パイプライン132を同様のデコーダを介して比較ロジック144に同様に接続するのが好ましいことに留意されたい。

20

【0037】

図6に示すように、処理システム100は、システム100のタイミングを制御するために用いられるラッチ145,147,149,151を備えている。ここで、当業界で既知の技術を用いて、ラッチ145,147,149,151はそれぞれ処理回路66,69,72,75に命令をラッチする。従来のシステム15と同様に、ラッチ145,147,149,151は、各ステージ133,136,139,142内の処理回路66,69,72,75がそれぞれ特定の時点で1つの命令のみを処理するよう制御されることが好ましい。更に、図6に示すパイプライン132は、最大で4つの命令、即ち、各処理回路66,69,72,75毎に1つの命令を、同時に処理することが可能である。しかし、他の実施形態では、任意の処理回路66,69,72及び/又は75は、ある時点で同時に2つ以上の命令を処理することも可能である。

30

【0038】

ラッチ145,147,149,151は、ラッチ145,147,149,151によりラッチされている命令によりそれぞれ規定されるエンコードされたnビットレジスタ識別子をデコーダ155,157,159,161にそれぞれ転送するよう構成される。このため、各ラッチ145,147,149,151は、処理回路66,69,72,75へ命令をそれぞれ転送する際に、各命令により規定されるエンコードされたnビットレジスタ識別子をデコーダ155,157,159,161にそれぞれ転送する。

【0039】

40

各デコーダ155,157,159,161は、受信したnビットレジスタ識別子を、該受信したnビットレジスタ識別子と同一レジスタ37を識別するmビットレジスタ識別子へとデコードするよう構成される。該値mは、システム100と関連するレジスタ37の数に対応することが好ましく、各mビットレジスタ識別子内の各ビットは、特定のレジスタ37に対応する。各デコーダ155,157,159,161は、受信したnビットレジスタ識別子によりどのレジスタ37が識別されるかを判定し、及び該識別されたレジスタ37に対応するmビットレジスタ識別子内のビットをアサートするよう構成されることが好ましい。該mビットレジスタ識別子内の残りのビットはデアサートされる。その結果として、各mビットレジスタ識別子は、アサートされた1つのビットとデアサートされた残りのビットとを有することになる。そのアサートされたビットは、受信したnビットレジスタ識別子を含む命令により用いられるレ

50

レジスタ37に対応する。しかし、他の技術を用いてnビットレジスタ識別子をデコードを行い、及び他の識別子を用いて命令に関連するレジスタ37を識別することも可能である点に留意されたい。

【0040】

各デコーダ155,157,159,161は、デコードされたmビットレジスタ識別子を比較ロジック144へ転送し、該比較ロジック144が、受信した各mビットレジスタ識別子を少なくとも1つの他の受信したmビットレジスタ識別子と比較し、データ従属性及び/又はデータ従属性ハザードが存在するか否かを判定する。2つの命令の間のデータ従属性ハザードは、該2つの命令間にデータ従属性がない場合には存在しないことに留意されたい。このため、データ従属性ハザードの検出はまたデータ従属性の検出であるとみなされるべきである。

10

【0041】

比較ロジック144の機能を例示するために、従来のシステム15と同様に、レジスタステージ133内のコンシューマが、他の何れかのステージ136,139,142内の何れかのプロデューサとのデータ従属性ハザードを有するか否かを判定することが好ましい。この例では、コンシューマがレジスタステージ133内にあり、プロデューサが他のステージ136,139,142内にあり、デコーダ155によりデコードされるmビットレジスタ識別子が、デコーダ157,159,161によりデコードされるmビットレジスタ識別子とそれぞれ比較されるものと仮定する。

【0042】

このため、図6に示すように、デコーダ155によりデコードされるmビットレジスタ識別子はANDロジック164,166,168に転送され、デコーダ157,159,161によりデコードされるmビットレジスタ識別子はANDロジック164,166,168にそれぞれ転送される。次いで、各ANDロジック164,166,168は、レジスタステージ133で処理されているコンシューマと他の任意のステージ136,139,142で処理されている何れかのプロデューサとの間にデータ従属性が存在するか否かを判定するために、受信したmビットレジスタ識別子を比較する。レジスタステージ133で処理されているコンシューマと、これからデータを生成する(即ちコンシューマが利用可能なデータを未だ生成していない)他のステージ136,139,142で処理されている何れかのプロデューサとの間にデータ従属性が存在する場合には、比較ロジック144は、該コンシューマと該1つのプロデューサとの間にデータ従属性ハザードが存在するものと判定する。次いで、制御回路(図示せず)が、かかるデータ従属性誤りを防ぐための判定に応じてレジスタステージ133内のコンシューマを停止させる(又は他のステップを実行する)ことが可能である。

20

30

【0043】

図7は、ANDロジック164,166,168の実施に適したロジックを示している。ここで、各ANDロジック164,166,168は、受信したmビットレジスタ識別子の1つのからの1ビットと受信した他のmビットレジスタ識別子からの1ビットとを各々が受信するm個のANDゲート172を備えている。各ANDゲート172により受信された両方のビットは、同一のレジスタ37に対応するものであることが好ましい。このため、受信した2つのmビットレジスタ識別子が一致する場合には、ANDゲート172の1つがアサート出力を生成することになる。したがって、ANDゲート172の出力を解析することが可能であり、ANDゲート172の何れかの出力がアサートされる場合には、一致したmビットレジスタ識別子と関連する命令間にデータ従属性が存在するものと判定することができる。ANDゲート172の出力を迅速に解析するために、各出力はORゲート175(図8)に転送することが可能であり、該ORゲート175は、2つの比較されたmビットレジスタ識別子が一致する場合にのみ、アサートされた出力を生成する。

40

【0044】

システム100の設計の結果として、データ従属性ひいてはデータ従属性ハザードを検出するために必要とされる配線を含む回路及び空間の量が従来のシステム15と比べて低減されることになる。特に、ステージ間でレジスタ識別子を転送するために図2において用いられる配線の数を著しく低減させることができ、また比較ロジック144の実施を、図3の比較ロジック24に関して説明したようなnビットコンパレータの実施よりも遙かに単純で小

50

さなものとすることができる。更に、デコーダ155,157,159,161並びに比較ロジック144の実施に用いられるデバイスは、空間及び配線の複雑さを最小限に抑えるよう容易に配置することができる。更に、デコーダ155,157,159,161がラッチ44,46,48及び52(図2)よりも小さな空間しか利用しないようにすることができる。その結果として、データ従属性及び/又はデータ従属性ハザードを検出するためのシステム100の回路を、従来のシステム15の回路よりも遙かに小型に及び効率的に構成することができる。

#### 【0045】

図6が、パイプライン132内の各命令からの1つのレジスタ識別子を他のレジスタ識別子と比較することができる回路を示していることに留意されたい。しかし、命令は2つ以上のレジスタ識別子を含むことも可能である。このため、図6の回路と同様に追加の回路を実施して、各命令の各レジスタ識別子が確実に検査できるようにすることも可能である。例えば、レジスタステージ133内のコンシューマは、実行時に2つの異なるレジスタ37からデータを取り出すことが可能であり、したがって、2つの異なるレジスタ識別子を含むことが可能である。かかる命令に適應させるために、レジスタステージ133は、別のデコーダ155を備えることが好ましい。該デコーダ155は、他のレジスタ識別子を、比較ロジック144により他のステージ136,139,142内の他のmビットレジスタ識別子と比較することが可能なmビットレジスタ識別子へとデコードする。したがって、図6に示す設計は、必要に応じて、2つ以上のレジスタ識別子と関連する命令に適應させるよう拡張することが可能なものである、ということは当業者には明らかであろう。

#### 【0046】

更に、図6は、単一のパイプライン132についての回路を示したものである。スーパースケラシステム100では、異なるパイプライン132内の命令間のデータ従属性ハザード、ひいてはデータ従属性を検査することが望ましい。このため、本書で説明する技術に従って、異なるパイプライン132の命令間のデータ従属性及び/又はデータ従属性ハザードを検査するために、1つのパイプライン132内の命令のmビットレジスタ識別子を、同一の又は異なるパイプライン132内の命令のmビットレジスタ識別子と比較することが可能である、ということは当業者には明らかであろう。例えば、1つのパイプライン132のデコーダ155によりデコードされるmビットデータ識別子を、ANDロジック164,166,168と類似したロジックを介して別のパイプライン132のデコーダ157,159,161によりデコードされるmビットレジスタ識別子と比較して、該1つのパイプライン132のレジスタステージ133内の命令と、他のパイプライン132のステージ136,139,142内の命令との間にデータ従属性及び/又はデータ従属性ハザードが存在するか否かを判定することが可能である。

#### 【0047】

更に、上記のように、レジスタステージ133内の各コンシューマと関連するレジスタ識別子を、ステージ136,139,142内の各プロデューサのレジスタ識別子と比較することが一般に望ましい。しかし、他の実施形態では、何れか1つのパイプライン132の何れか1つのステージ133,136,139又は142からのレジスタ識別子を、何れか1つのパイプライン132の何れか1つのステージ133,136,139又は142からのレジスタ識別子と比較して、該比較されたレジスタ識別子と関連する2つの命令間にデータ従属性が存在するか否かを判定することが可能である。

#### データの融合

必要に応じて、比較ロジック144に与えられるデータを融合させて、データハザードの効率的な検出を可能にすることが可能である。例えば、図9に示すように、融合回路177,179は、デコーダ155,157,159,161から受信したデータを融合させるように及び該データと比較ロジック144とのインタフェイスを形成するように構成することが可能である。

#### 【0048】

融合回路177は、図9に示すパイプライン132のレジスタステージ133からのmビットレジスタ識別子と、別のステージ及び/又は別のパイプライン132からの少なくとも1つのmビットレジスタ識別子とを受信し、該受信したmビットレジスタ識別子を規定するデータの少なくとも一部を結合するように構成される。例えば、図9に示す実施形態では、融合

回路177は、同図に示すパイプライン132のレジスタステージ133と関連するデコーダ155からのmビットレジスタ識別子と、別のパイプライン132からのmビットレジスタ識別子とを受信する。次いで融合回路177は、該受信したレジスタ識別子を結合させて、該受信したレジスタ識別子により識別される各レジスタ37(図5)を識別する単一のmビットレジスタ識別子を形成する。

【0049】

ここで、融合回路177により生成されるmビットレジスタ識別子内の各ビットは、システム100の異なるレジスタ37の異なる1つに対応する。更に融合回路177は、受信したmビットレジスタ識別子の1つにより識別されるレジスタ37に対応する先行するmビットレジスタ識別子内の各ビットをアサートし、及びmビットレジスタ識別子の残りのビットをデアサートするように設計される。

10

【0050】

しかし、融合回路177は、何れかのパイプライン132のレジスタステージ133内の各コンシューマと関連する各mビットレジスタ識別子を受信することが好ましいことに留意されたい。このため、2つのパイプライン132のみがシステム100内に存在し、及び実行時に各コンシューマが単一のレジスタ37からしか読み出しを行わないものと仮定すると、図9に示す回路は本発明の好適実施形態に従うものとなる。しかし、多くの応用形態では、パイプライン132の数は3つ以上であり、単一のコンシューマの実行を介して読み出しが行われるレジスタ37は2つ以上存在する。その結果として、融合回路177により受信されるレジスタ識別子の数が増加する場合がある。

20

【0051】

何れにしても、融合回路177により生成されるmビットレジスタ識別子は、該融合回路177により受信されるmビットレジスタ識別子により識別される各レジスタ37を識別し、何れかのパイプライン132のレジスタステージ133内の何れかのコンシューマの実行を介して読み出されることになる各レジスタ37を識別することが好ましい。ここで、受信されるmビットレジスタ識別子により識別されるレジスタ37の1つに対応する、融合回路177により生成されるmビットレジスタ識別子の各ビットがアサートされることになる。融合回路177により生成されるmビットレジスタ識別子の残りのビットはデアサートされることになる。

【0052】

30

融合回路179は、図9に示すパイプライン132のステージ136,139及び/又は142からmビットレジスタ識別子を受信し、及び/又は別のパイプライン132からmビットレジスタ識別子を受信するように構成される。次いで、融合回路179は、融合回路177と同様に、受信されたmビットレジスタ識別子を規定するデータの少なくとも一部を結合するよう設計される。例えば、図9に示す実施形態では、融合回路179は、同図に示すパイプライン132の各ステージ136,139,142からのmビットレジスタ識別子と、別のパイプライン132の各ステージ136,139,142からのmビットレジスタ識別子とを受信する。次いで融合回路179は、該受信したレジスタ識別子を結合させて、該受信したレジスタ識別子により識別される各レジスタ37(図5)を識別する1つのmビットレジスタ識別子を形成する。

【0053】

40

ここで、融合回路179により生成されるmビットレジスタ識別子内の各ビットは、システム100の異なるレジスタ37の1つに対応する。更に、融合回路179は、受信したmビットレジスタ識別子の1つにより識別されるレジスタ37に対応する先行するmビットレジスタ識別子内の各ビットをアサートし、該mビットレジスタ識別子の残りのビットをデアサートするよう設計される。

【0054】

しかし、融合回路179は、何れかのパイプライン132の何れかのステージ133,139,142内の何れかのプロデューサと関連する各mビットレジスタ識別子を受信することが好ましいことに留意されたい。このため、上述のように、2つのパイプライン132のみがシステム100内に存在し、及び実行時に各プロデューサが1つのレジスタ37に対してのみ書き込みを行

50

うものと仮定すると、図9に示す回路は好ましい実施形態に従うものとなる。しかし、多くの応用形態では、パイプライン132の数は3つ以上であり、1つのプロデューサの実行を通して書き込まれるレジスタ37は2つ以上存在する場合がある。その結果として、融合回路179により受信されるレジスタ識別子の数が増加することになる。更に、融合回路179は、必ずしも各ステージ136, 139, 142からレジスタ識別子を受信する必要はない。

【0055】

何れにしても、融合回路179により生成されるmビットレジスタ識別子は、該融合回路179により受信されるmビットレジスタ識別子により識別される各レジスタ37を識別し、この際に、何れかのパイプライン132のステージ136, 139, 142内の何れかのプロデューサにより書き込まれる各レジスタ37を識別することが好ましい。ここで、融合回路179により生成されるmビットレジスタ識別子の各ビットであって受信されるmビットレジスタ識別子により識別されるレジスタ37の1つに対応する各ビットがアサートされることになる。融合回路179により生成されるmビットレジスタ識別子の残りのビットはデアサートされることになる。

【0056】

図10は、融合回路177を実施するために用いることができる典型的な回路を示している。ここで、融合回路177は、m個のORゲート180を備えることが可能である。各ORゲート180は、該融合回路177により受信される各レジスタ識別子からの1ビットを受信する。更に、1つのORゲート180により受信される各ビットは、同一のレジスタ37(図5)に対応するものであることが好ましい。このため、融合回路177により受信されるmビットレジスタ識別子のうちの1つにより識別される各レジスタ37毎に、ORゲート180の1つがアサートされた出力を生成することになる。その結果として、融合回路177は、mビットレジスタ識別子を生成し、そのmビットレジスタ識別子の各ビットがレジスタ37の1つに対応する。受信した何れかのレジスタ識別子により識別されるレジスタ37に対応するmビットレジスタ識別子のビットはアサートされ、一方、残りのビットはデアサートされることになる。

【0057】

図11は、融合回路179を実施するために用いることができる典型的な回路を示す。ここで、融合回路179は、m個のORゲート181を備えることが可能である。各ORゲート181は、融合回路179により受信される各レジスタ識別子からの1ビットを受信する。更に、単一のORゲート181により受信される各ビットが同一レジスタ37に対応するものであることが好ましい。このため、融合回路179により受信されるmビットレジスタ識別子のうちの1つにより識別される各レジスタ37毎に、ORゲート181の1つがアサートされた出力を生成することになる。その結果として、融合回路179は、mビットレジスタ識別子を生成し、該mビットレジスタ識別子の各ビットがレジスタ37の1つに対応する。受信した何れかのレジスタ識別子により識別されるレジスタ37に対応するmビットレジスタ識別子のビットはアサートされ、一方残りのビットはデアサートされることになる。

【0058】

融合回路177, 179により現時点で生成されるmビットレジスタ識別子を比較することにより、比較ロジック144は、パイプライン132により処理されている命令間にデータハザードが存在するか否かを判定することができる。融合回路177, 179により生成されたレジスタ識別子を比較するために図7に示すANDロジック164を実施することが可能である。

【0059】

ここで、各ANDゲート172(図7)は、融合回路177, 179により生成される各レジスタ識別子からの1ビットを受信する。更に、1つのANDゲート172により受信される各ビットは、同一のレジスタ37に対応するものであることが好ましい。このため、融合回路177, 179から受信されるmビットレジスタ識別子の両方により識別される各レジスタ37毎に、ANDゲート172の1つがアサートされた出力を生成することになる。その結果として、図9のANDロジック164は、mビットレジスタ識別子を生成し、該mビットレジスタ識別子の各ビットがレジスタ37の1つに対応する。融合回路177, 179から受信した両方のレジスタ識別子

により識別されるレジスタ37に対応するmビットレジスタ識別子のビットはアサートされ、一方、残りのビットはデアサートされることになる。

#### 【0060】

このため、ANDロジック164により生成されるmビットレジスタ識別子のアサートされた各ビットは、対応するレジスタ37に関連するデータハザードが存在する可能性あることを示し、ANDロジック164により生成されるmビットレジスタ識別子のデアサートされた各ビットは、対応するレジスタ37に関連するデータハザードが存在しないことを示す。従って、比較ロジック144は、ANDロジック164により生成されるmビットレジスタ識別子内のアサートされた各ビット毎にデータハザードを検出するよう構成することが可能である。

#### 属性データ

システム100の効率を更に向上させるために、追加の回路を実施して不必要な停止(stall)を防止し、これによりパイプライン132の命令を処理する際に受ける遅延を低減させることが可能である。これに関連して、パイプラインにより処理されている命令をイネーブル及びディセーブルする述語(predication)技術が開発されている。イネーブルされた命令はパイプライン132により実行され、ディセーブルされた命令は実行されることなくパイプライン132内を通過する。本出願人の「System and Method for Providing Predicate Data」と題する米国特許出願(代理人整理番号第10971191号)には、プロセッサ性能を向上させるために述語データを用いるプロセスについて更に説明されている。

#### 【0061】

停止による悪影響を最小限にするために、述語データを解析して、命令がパイプライン132により実行されるべきか否かを判定することが可能である。述語データから、ある命令が実行されるべきではないものと判定できる場合には、その命令は、その命令と他の命令との間のデータハザードに起因してデータ誤りを生じさせる可能性はない。このため、その命令のレジスタ識別子が別の命令のレジスタ識別子と一致する場合であっても、それら命令の一方が記述データによりディセーブルされる場合には、実際にはそれら命令間にデータハザードが存在することはない。その結果として、不要な停止を防ぐために、2つの命令の一方が述語データによりディセーブルされている場合には、比較ロジック144は、それら命令間のデータハザードを検出するのをやめることが望ましい。

#### 【0062】

更に、以下により詳細に説明するように、2つの命令のレジスタ識別子が一致する場合であっても、パイプライン132により処理されている命令のタイプに基づいて、該2つの命令間にデータハザードが存在しないものと判定することが可能な場合もある。結果的に、追加の回路をシステム100内に配設し、該回路により、属性データ(パイプライン132により処理されている命令の述語状態及び/又はタイプを示すデータ等)を解析して、不要な停止の発生を防止できるようにすることが望ましい。

#### 【0063】

図12は、データハザードの検出時に属性データ(即ち命令のタイプ及び/又は述語状態を示すデータ)を解析するためにシステム100により用いることが可能な追加の回路183, 185, 187, 189を示している。同図に示すように、デコーダ155, 157, 159, 161により生成されるmビットレジスタ識別子はそれぞれ属性インタフェイス183, 185, 187, 189に入力され、属性インタフェイス183, 185, 187, 189はそれぞれラッチ145, 147, 149, 151から属性データを受信する。各属性インタフェイス183, 185, 187, 189は、受信した属性データと、受信したmビットレジスタ識別子とのインタフェイスを形成して、受信したmビットレジスタ識別子と関連する命令を実行するためにどのレジスタ37(図5)が用いられるかを示すだけでなく命令及び/又はその命令の述語状態を示すデータも生成する。このため、比較回路191は、命令を実行するために用いられるレジスタ37だけでなく命令のタイプ及び/又は命令の述語状態にも基づいて、上記命令と別の命令との間にデータハザードが存在するか否かを検出することが可能である。その結果、比較ロジック191は、比較ロジック144よりも正確にデータハザードを検出するよう構成される。

#### 【0064】

10

20

30

40

50

例示を目的として、例外検出ステージ139内のプロデューサがマルチメディア（mmu）命令である場合にのみ、レジスタステージ133内の非マルチメディア（非mmu）コンシューマが、該プロデューサとのデータハザードを規定するものと仮定する。更に、非mmuコンシューマが現時点でレジスタステージ133内に存在するものと仮定する。比較ロジック144（図6）が属性データを解析せず、このため、パイプライン132により処理されている命令のタイプ及び命令の述語状態を考慮しない場合には、比較ロジック144は、パイプライン132のうちの1つのデコーダ159により受信されるレジスタ識別子がパイプライン132のうちの1つのデコーダ155により同時に受信されるレジスタ識別子と一致する場合には必ず、データハザードを検出することになる。

【0065】

したがって、レジスタステージ133内のコンシューマと同一レジスタ識別子を有する、パイプライン132の例外検出ステージ139内の少なくとも全てのプロデューサが、少なくとも書き込みステージ142に進むまで、レジスタステージ133内の上記コンシューマは停止されることになる。換言すれば、少なくとも1つのタイプのプロデューサは、該プロデューサが少なくとも書き込みステージ142に到達するまで、レジスタステージ133内の少なくとも1つのタイプのコンシューマにとって利用不可能なデータを生成するため、該コンシューマは、該コンシューマに先行すると共に該コンシューマと同一のレジスタ識別子を有する全てのプロデューサが少なくとも書き込みステージ142に到達するまで、レジスタステージ133内で停止されることになる。さもないと、パイプライン132のうちの1つの例外検出ステージ139内のプロデューサがmmu命令であり及びレジスタステージ133内のコンシューマが非mmu命令である場合にデータ誤りが発生する可能性がある。

【0066】

しかし、比較回路191は、命令のレジスタ識別子のみならず命令のタイプ及び／又は述語状態を示す属性データを受信し解析するよう設計される。このため、比較ロジック191は、2つの命令が同一のレジスタ識別子を有する場合であっても、例外検出ステージ139内のプロデューサとレジスタステージ133内のコンシューマとの間にデータハザードが存在しないことを検出することができる。

【0067】

ここで、レジスタステージ133のラッチ145が、レジスタステージ133内の命令の述語状態を示す1ビットの属性データを伝送し、及びレジスタステージ133内の命令が特定のタイプの命令（例えば上記実施形態におけるmmu命令）であるか否かを示す1ビットの属性データを伝送する。更に、例外検出ステージ139のラッチ149が、例外検出ステージ139内の命令の述語状態を示す1ビットの属性データを伝送し、及び例外検出ステージ139内の命令が特定のタイプの命令（例えば上記実施形態におけるmmu命令）であるか否かを示す1ビットの属性データを伝送する。属性インタフェイス183,187はラッチ145,149から属性データをそれぞれ受信し、該受信した属性データ及びデコードされたレジスタ識別子を示すデータをデコーダ155,159からハザード検出回路196へ送信する。

【0068】

ハザード検出回路196は、1) レジスタステージ133内のコンシューマが、例外検出ステージ139内のプロデューサと同一のレジスタ識別子を有し、2) レジスタステージ133及び例外検出ステージ139内の両方の命令が、述語イネーブルされており（predicate enabled）、3) レジスタステージ133及び例外検出ステージ139内の命令のタイプが、2つの命令間にデータハザードが存在し得るものである場合にのみ、データハザードを検出するよう設計される。例示のため、（a）レジスタステージ133内のコンシューマが非mmu命令であり、かつ例外検出ステージ139内のプロデューサがmmu命令であるか、（b）レジスタステージ133内のコンシューマがmmu命令であり、かつ例外検出ステージ139内のプロデューサが非mmu命令である場合にのみ、上記条件3）が満たされるものと仮定する。しかし、他のタイプの命令及び／又は他のタイプの命令の組み合わせもまた条件3）を満たすことが可能であり、この場合には、本発明の上記実施形態を実施するために用いた回路を変更することが必要となる（これについては以下で詳述する）、ということが当業者には明らかであ

10

20

30

40

50

ろう。

【 0 0 6 9 】

3つの上記条件（即ち条件1）～3））の何れかの条件が満たされない場合に、ハザード検出回路196はデータハザードを検出ししない。その結果として、幾つかの状況で、レジスタステージ133内のコンシューマの停止を、該コンシューマと同一のレジスタ識別子を有するプロデューサが書き込みステージ142に達するまで防止し又は排除することができ、これにより該コンシューマを先行してパイプライン132により処理することが可能となる。

【 0 0 7 0 】

図13は、属性インタフェイス183に接続されるデコーダ155の典型的な実施形態を示している。ここで、該デコーダ155は、処理システム100のレジスタ37（図5）にそれぞれ対応するm行のデコードロジック202を有する一列のロジックである。該デコードロジック202の各行は、デコーダ155に送信されるnビットレジスタ識別を受信し、該nビットレジスタ識別子がデコードロジック202の当該行に対応するレジスタ37を識別する場合にアサートされた出力を送信し、該nビットレジスタ識別子が別のレジスタ37を識別する場合にはデアサートされた出力を送信するように設計される。換言すれば、デコードロジック202の各行は、デコーダ155により生成されるmビットレジスタ識別子のうちの1ビットを出力する。

10

【 0 0 7 1 】

属性インタフェイス183は、m行のインタフェイスロジック204を備えている。該m行のインタフェイスロジック204は、デコードロジック202の各行にそれぞれ対応し、したがってシステム100のレジスタ37に対応するものである。各行のインタフェイスロジック204は、デコードロジック202のうちの1つの出力を受信し、及びラッチ145から伝送される属性データを受信するよう設計される。他の属性インタフェイス185, 187, 189により処理される属性データはそれぞれラッチ147, 149, 151から受信される。

20

【 0 0 7 2 】

ラッチ145は、レジスタステージ133内の命令の述語状態を示す（即ちレジスタステージ内の命令がイネーブルされたか否かを示す）1ビット値と、レジスタステージ133内の命令が特定のタイプの命令であるか否かを示す少なくとも1ビット値とを属性インタフェイス183に伝送するよう構成することが可能である。例えば、ラッチ145は、レジスタステージ133内の命令が述語イネーブルされた場合にのみ複数のビット（即ち図14において接続部207を介して伝送されるビット）のうちの1つをアサートし、レジスタステージ133内の命令がmmu命令である場合にのみ他のビット（接続部208を介して伝送されるビット）をアサートすることが好ましい。

30

【 0 0 7 3 】

属性インタフェイス183内の各行のインタフェイスロジック204は、デコーダ155から受信されるビットがアサートされているか否かを示すだけでなくレジスタステージ133内の命令の述語状態及び/又はタイプも示す出力を生成するように構成される。例えば、図13及び図14により示される実施形態における各行のインタフェイスロジック204が2ビット出力を生成することが可能である。単一行のインタフェイスロジック204により出力されるビットの何れも、該行のインタフェイスロジック204がデコーダにより現時点で受信されたnビットレジスタ識別子により識別されるレジスタ37に対応し、かつレジスタステージ133内の命令が述語イネーブルされている場合にのみ、アサートすることが可能である。換言すれば、各行のインタフェイスロジック204は、デコーダ155から受信されるビット値がデアサートされる場合、又は接続部207を介して伝送されるビット値がデアサートされる場合に、両方の出力ビットをデアサートするよう構成される。

40

【 0 0 7 4 】

属性インタフェイス183の各行のインタフェイスロジック204は更に、ラッチ145から受信される属性データが、レジスタステージ133内の命令が特定のタイプの命令であることを示す場合にのみ、各出力ビットをそれぞれアサートするように構成される。例えば、属性インタフェイス183に関する図14のANDゲート209は、レジスタステージ133内の命令がmm

50



u命令である場合にのみアサートされた出力を生成し、属性インタフェイス183に関する図14のANDゲート210は、レジスタステージ133内の命令が非mmu命令である場合にのみアサートされた出力を生成することが可能である。

【0075】

ここで、ANDゲート209の出力がアサートされる場合には、レジスタステージ133内の命令が、1) イネーブルされ、2) mmu命令（即ち例外検出ステージ139内の命令が非mmu命令である場合にのみ該例外検出ステージ139内の命令とのデータハザードを規定することができるタイプのもの）であり、3) ANDゲート209を含む当該行のインタフェイスロジック204に対応するレジスタ37を利用する、ということがわかる。更に、ANDゲート210の出力がアサートされる場合には、レジスタステージ133内の命令が、1) イネーブルされ、2) 非mmu命令（即ち例外検出ステージ139内の命令がmmu命令である場合にのみ該例外検出ステージ139内の命令とのデータハザードを規定することができるタイプのもの）であり、3) ANDゲート210を含む当該行のインタフェイスロジック204に対応するレジスタ37を利用する、ということがわかる。

【0076】

デコーダ159及び属性インタフェイス187の回路は、上記デコーダ155及び属性インタフェイス183の回路と類似又は同一のものとすることが可能である。更に、ラッチ145と同様に、ラッチ149は、例外検出ステージ139内の命令がイネーブルされた場合にのみ、属性インタフェイス187へ伝送される属性ビット（即ち図14において接続部207を介して伝送されるビット）のうちの1つをアサートし、例外検出ステージ139内の命令が特定のタイプの命令（例えばmmu命令）である場合にのみ、属性インタフェイス187へ伝送される他の属性ビット（即ち図14において接続部208を介して伝送されるビット）をアサートする。

【0077】

属性インタフェイス183の出力と同様に、属性インタフェイス187に関するインタフェイスロジック204（図13）の各行の出力を2ビット出力とすることが可能である。ここで、属性インタフェイス187に関する図14のANDゲート209の出力は、例外検出ステージ139内の命令が、1) イネーブルされ、2) mmu命令（即ちレジスタステージ133内の命令が非mmu命令であるときにのみ、レジスタステージ133内の命令とのデータハザードを規定することができるタイプのもの）であり、及び、3) ANDゲート209を含むインタフェイスロジック204の行と対応するレジスタ37を用いる場合にのみ、アサートされることが好ましい。更に、上記の実施形態では、属性インタフェイス187に関する図14のANDゲート210の出力は、例外検出ステージ139内の命令が、1) イネーブルされ、2) 非mmu命令（即ちレジスタステージ133内の命令がmmu命令であるときにのみ、レジスタステージ133内の命令でデータハザードを規定することができるタイプのもの）であり、及び、3) ANDゲート209を含むインタフェイスロジック204の行と対応するレジスタ37を用いる場合にのみ、アサートされる。

【0078】

ハザード検出口ジック196は、属性インタフェイス183,187からの出力を受信して比較し、該ロジック196内に入力される情報に基づいてデータハザードが存在するか否かを検出するように構成される。ここで、レジスタステージ133内のコンシューマのレジスタ識別子と例外検出ステージ139内のプロデューサのレジスタ識別子とが一致し、該コンシューマ及び該プロデューサの両方がイネーブルされ、及び該コンシューマ及び該プロデューサがデータハザードを規定するタイプのものである（例えば、(a) レジスタステージ133内のコンシューマが非mmu命令であり、かつ例外検出ステージ139内のプロデューサがmmu命令であり、又は(b) レジスタステージ133内のコンシューマがmmu命令であり、例外検出ステージ139内のプロデューサが非mmu命令である）場合にのみ、ハザード検出口ジック196がデータハザードを検出する。

【0079】

図15は、上記の実施形態においてハザード検出口ジック196を実施するために用いることが可能な回路を示している。ここで、ハザード検出口ジック196はm行の比較ロジック2

10

20

30

40

50

11を備えている。該比較ロジック211の各行は、属性インタフェイス183からのインタフェイスロジック204の1つの行と、属性インタフェイス187からのインタフェイスロジック204の1つの行とからの出力を受信するよう構成される。ハザード検出回路196内の同一行の比較ロジック211に出力を伝送する属性インタフェイス183,187のインタフェイスロジック204の各行は、同一のレジスタ37に対応することが好ましい。換言すれば、ハザード検出回路196内の比較ロジック211の同一行に接続される、属性インタフェイス183,187のインタフェイスロジック204の両方の行は、デコーダ155,159により同時に受信されるnビットレジスタ識別子が一致する場合にのみ、アサートされた出力を生成することが可能である。このため、比較ロジック211の行の何れもが、少なくとも属性インタフェイス183からの1ビットのアサートされた出力と属性インタフェイス187からの1ビットのアサートされた出力とを同時に受信しない場合には、レジスタステージ133及び例外検出ステージ139内の命令に関連するnビットレジスタ識別子が一致せず、又はレジスタステージ133又は例外検出ステージ139内の命令の少なくとも1つがディセーブルされている。その結果として、一行の比較ロジック211が、属性インタフェイス183からの少なくとも1つのアサートされたビットと、属性インタフェイス187からの1つのアサートされたビットを受信しない場合には、ハザード検出ロジック196はデータハザードを検出せず、現時点でデータハザードが検出されないことを示す信号を送信する。

10

**【0080】**

しかし、複数行の比較ロジック211のうちの1つが、少なくとも属性インタフェイス183からの1ビットのアサートされた出力と属性インタフェイス187からの1ビットのアサートされた出力とを同時に受信する場合には、レジスタステージ133及び例外検出ステージ139内の命令に関連するnビットレジスタ識別子が一致する。このため、レジスタステージ133及び例外検出ステージ139内の両方の命令がイネーブルされ、更に、これら2つの命令間でデータハザードが存在するか否かを判定するための解析が実行される。ここで、レジスタステージ133及び例外検出ステージ139内の命令がデータハザードを生成しないタイプのものであることを上述の属性インタフェイス183,187からの出力が示さない限り、アサートされた出力を受信する比較ロジック211の行はデータハザードを検出する。

20

**【0081】**

例えば、上記の典型的な実施形態では、ハザード検出回路196は、(a)属性インタフェイス187からの出力が例外検出ステージ139内のプロデューサがmmu命令であることを示し、かつ属性インタフェイス183からの出力が、レジスタステージ133内のコンシューマが非mmu命令であることを示す場合にのみ、又は(b)属性インタフェイス187からの出力が、例外検出ステージ139内のプロデューサが非mmu命令であることを示し、かつ属性インタフェイス183からの出力が、レジスタステージ133内のコンシューマがmmu命令であることを示す場合にのみ、データハザードを検出することが可能となる。

30

**【0082】**

図16は、上記の実施形態に関する一行分の比較ロジック211を実施するために用いることができる回路を示している。同図の接続部221は、属性インタフェイス183(図12)内のインタフェイスロジック204の行に含まれるANDゲート209(図14)の出力に接続され、接続部223は、属性インタフェイス183(図12)内の上述のインタフェイスロジック204の行に含まれるANDゲート210(図14)の出力に接続される。更に、図16の接続部227は、属性インタフェイス187(図12)内のインタフェイスロジック204の行に含まれるANDゲート210(図14)の出力に接続され、接続部229は、上述の属性インタフェイス187(図12)内のインタフェイスロジック204の行に含まれるANDゲート209(図14)の出力に接続される。

40

**【0083】**

このため、接続部231上の値は、(1)レジスタステージ133内のコンシューマのレジスタ識別子が例外検出ステージ139内のプロデューサのレジスタ識別子と一致し、(2)レジスタステージ133及び例外検出ステージ139内の命令が何れも述語イネーブルされ、(3)レジスタステージ133内のコンシューマが非mmu命令であり、及び(4)例外検出ステージ139

50

内のプロデューサがmmu命令である場合にのみ、アサートされることになる。更に、接続部233上の値は、(1)レジスタステージ133内のコンシューマのレジスタ識別子が例外検出ステージ139内のプロデューサのレジスタ識別子と一致し、(2)レジスタステージ133及び例外検出ステージ139内の命令が何れも述語イネーブルされ、(3)レジスタステージ133内のコンシューマがmmu命令であり、及び(4)例外検出ステージ139内のプロデューサが非mmu命令である場合にのみ、アサートされることになる。

#### 【0084】

その結果として、ハザード検出回路196内の比較ロジック211の行は、接続部231又は233上の値がアサートされた際に、レジスタステージ133内のコンシューマと例外検出ステージ139内のプロデューサとの間のデータハザードを検出する。ここで、ORゲート235により出力される値は、比較ロジック211の行がデータハザードを検出するか否かを示すものとなる。ORゲート235の出力の値がアサートされると、比較ロジック211の行は、レジスタステージ133及び例外検出ステージ139内の2つの命令間にデータハザードが存在することを示す。逆に、ORゲート235の出力の値がデアサートされると、比較ロジック211の行は、レジスタステージ133及び例外検出ステージ139内の2つの命令間のデータハザードが該比較ロジック211の行によって検出されないことを示す。

#### 【0085】

図17に示すように、ハザード検出回路196内の比較ロジック211の各行に対するORゲート235の出力は、ORゲート252により結合され、該ORゲート252の出力が、ハザード検出回路196がデータハザードを検出するか否かを示すようになる。ORゲート252の出力の値がアサートされると、ハザード検出回路196は、レジスタステージ133及び例外検出ステージ139内の2つの命令間にデータハザードが存在することを示す。逆に、ORゲート252の出力の値がデアサートされると、ハザード検出回路196は、レジスタステージ133及び例外検出ステージ139内の2つの命令間にデータハザードが存在しないことを示す。

#### 【0086】

図6に示すシステム100の場合と同様に、ハザードを検出するために、レジスタステージ133及び例外検出ステージ139以外のステージ内の命令のレジスタ識別子及び属性データが比較されることに留意されたい。ここで、デコーダ157,161、属性インタフェイス185,189、及びハザード検出ロジック194,198の構成は、それぞれ、デコーダ155,159、属性インタフェイス183,187、及びハザード検出ロジック196の構成と同様にすることが可能である。更に、比較ロジック144と同様に、比較ロジック191は、本発明に従って、1つのパイプライン132内の命令のレジスタ識別子及び属性データを、別のパイプライン132内の別の命令のレジスタ識別子及び属性データと比較して、データハザードを検出することが可能である。

#### 【0087】

更に、本発明の原理から逸脱することなく図12に示す回路の変更形態を実施できることは当業者には明らかであろう。例えば、述語状態を示すビット又は命令タイプを示すビットといった1ビットの属性データのみを属性インタフェイス183,185,187,189へ伝送し、該ビットを用いてデータハザードを正確に検出することが可能である。ここで、比較回路191は、2つの命令のレジスタ識別子が一致し、かつ両方の命令がイネーブルされる場合にのみ、データハザードを検出するように構成することが可能であり、又は、2つの命令のレジスタ識別子及びタイプに基づきデータハザードを検出するように構成することが可能である。

#### 【0088】

更に、パイプライン132から属性インタフェイス183,185,187,189へ送信される属性データによって異なる命令タイプを示すことが可能であり、また異なる数のビット値を送信して、レジスタ識別子、述語状態、及び/又は属性情報を表すことが可能である。特に、上述の命令タイプに加えて又は上述以外の命令タイプによりデータハザードを規定することが可能であり、ステージ133,136,139及び/又は142のうちの何れかのステージの属性データが異なる命令タイプを示すことも可能である。属性インタフェイス183,185,187及び/又

は189のそれぞれに送信される複数組の属性データは、1ビット又は2ビット以上の情報とすることが可能である。更に、属性インタフェイス183,185,187及び/又は189並びにハザード検出回路194,196及び/又は198の回路の構成は、特に、他のタイプの命令間のデータハザードを示すよう変更することを必要とする可能性がある。

#### 【0089】

更に、データハザードを生成し得る命令タイプは、各ステージ間で変化し得るものである。このため、属性インタフェイス183,185,187及び/又は189にそれぞれ伝送される属性データにより示される命令タイプは異なる可能性がある。例えば、書き込みステージ142内にある2つのタイプの命令が、レジスタステージ133内の特定のタイプの命令との間でデータハザードを生成する可能性がある。その結果として、属性インタフェイス189は3ビット（書き込みステージ142内の命令の述語状態を示す1ビットと、該命令が、レジスタステージ133内の命令とのデータハザードを規定する可能性のある2つのタイプの命令か否かを示す2ビット）の属性データを受信する可能性がある。次いで、該データが本発明の原理に従って解析されて、データハザードを検出することが可能となる。別のステージ133,136,139及び/又は142内の別のタイプの命令とのデータハザードを規定し得る各ステージ133,136,139及び/又は142内の命令のタイプを示す各133,136,139及び/又は142からの属性データのみを送信することにより、本発明を実施するための回路の規模を最小限にすることができる。

10

#### 【0090】

更に、必要に応じて、図2及び/又は図3に示す従来のシステム15に属性インタフェイス183,185,187及び/又は189を組み込んで、比較回路22及び/又は24が属性情報及びレジスタ識別子に基づきデータハザードを検出できるようにすることも可能である。

20

#### 【0091】

更に、本発明の原理を用いて、書き込み - 書き込み（以下WAW(write-after-write)と称す）ハザードを検出することも可能である点に留意されたい。WAWハザードは、（1）先行する命令と後続の命令の何れも同一レジスタに書き込みを行う場合であって、（2）先行する命令が実際に該レジスタに書き込みを行う前に存在するものである。データ従属性ハザードを検出するのと同様に、WAWハザードは、2つの書き込み命令が同一のレジスタ識別子を含むか否かを判定することにより検出することができる。このため、上述の回路を用いて、2つの書き込み命令が同一のレジスタを用いる際に検出することができる。次いで、追加の回路を用いて、先行する命令からのデータが該レジスタに書き込まれたか否かを判定することが可能である。次いで上述の情報を用いてWAWハザードが存在するか否かを判定することが可能である。

30

#### 属性データの融合

システム100の効率を最大限にし、かつシステム100の実施に必要な配線数を低減させるために、属性インタフェイス183,185,187及び/又は189により生成されるデータを、図9のデコード155,157,159及び/又は161により生成されるデータと同様に、融合させることが可能である。これらの原理を例示するために、図18を参照する。同図は、属性データを含むデータであってデータハザードの検出に利用されるデータを融合するために用いることが可能な典型的な回路を示している。

40

#### 【0092】

例示のため、1) コンシューマがプロデューサと同一レジスタ識別子を有し、2) コンシューマ及びプロデューサの両方が述語イネーブルされ、及び 3) コンシューマ及びプロデューサのタイプが該2つの命令間にデータハザードが存在し得るものである場合にのみ、レジスタステージ133内のコンシューマとステージ136,139又は142の1つのステージ内のプロデューサとの間にデータハザードが存在し得るものと仮定する。また例示のため、(a) プロデューサが実行ステージ136、例外検出ステージ139、又は書き込みステージ142内にあるときに、コンシューマが非mmu命令であり、及びプロデューサがmmu命令であり、(b) プロデューサが実行ステージ136、例外検出ステージ139、又は書き込みステージ142内にあるときに、コンシューマがmmu命令であり、及びプロデューサが非mmu命令であり、又

50

は(c)プロデューサが実行ステージ136内にあるときに、コンシューマが非mmu命令であり、及びプロデューサが算術演算装置(ALU)命令である場合にのみ、上記条件3)が満たされるものと仮定する。しかし、以下で詳細に説明するように、他の命令タイプ及び/又は他の命令タイプの組み合わせもまた上記条件3)を満たすことが可能であり、この場合には本発明の上記形態を実施するために用いられる回路を変更する必要がある、ということが当業者には明らかであろう。

【0093】

上記仮定に鑑み、データハザードを検出するために、属性インタフェイス212は、図12の属性インタフェイス183と同一に構成され、このため、図13に示すようにm行のインタフェイスロジック204を備えたものとなる。属性インタフェイス212内のインタフェイスロジック204の各行は、レジスタ37の異なる1つに対応し、2ビット出力を生成する。レジスタステージ133内のコンシューマが述語イネーブルされない場合、又はコンシューマのレジスタ識別子がインタフェイスロジック204の行に対応するレジスタ37を識別しない場合には、インタフェイスロジック204の行により出力されるビットはアサートされない。更に、レジスタステージ133内のコンシューマがmmu命令である場合にのみ、インタフェイスロジック204のその行により出力されるビットの1つはアサートされ、そのコンシューマが非mmu命令である場合にのみ、他のビットがアサートされる。その結果として、属性インタフェイス212からハザード検出回路196に入力される値は、図12の属性インタフェイス183により生成されることになる値と同一値をとることになる。

【0094】

更に、属性インタフェイス216,218の構成は、図12の属性インタフェイス187と同一である。このため、属性インタフェイス212の出力と同様に、属性インタフェイス216,218はそれぞれ、図13に示すようにm行のインタフェイスロジック204を備える。属性インタフェイス216のインタフェイスロジック204の各行は、レジスタ37の異なる1つに対応し、2ビット出力を生成する。更に、属性インタフェイス218のインタフェイスロジック204の各行は、レジスタ37の異なる1つに対応し、2ビット出力を生成する。例外検出ステージ139内のプロデューサが述語イネーブルされない場合、又はプロデューサのレジスタ識別子がインタフェイスロジック204の行に対応するレジスタ37を識別しない場合には、属性インタフェイス216のインタフェイスロジック204の行により出力されるビットはアサートされない。同様に、書き込みステージ142内のプロデューサが述語イネーブルされない場合、又は該プロデューサのレジスタ識別子がインタフェイスロジック204の行に対応するレジスタ37を識別しない場合には、属性インタフェイス218内のインタフェイスロジック204の行により出力されるビットはアサートされない。

【0095】

更に、例外検出ステージ139内のプロデューサがmmu命令である場合にのみ、属性インタフェイス216内のインタフェイスロジック204の行により出力されるビットの1つがアサートされ、該プロデューサが非mmu命令である場合にのみ、他のビットがアサートされる。更に、書き込みステージ142内のプロデューサがmmu命令である場合にのみ、属性インタフェイス218内のインタフェイスロジック204の行により出力されるビットの1つがアサートされ、該プロデューサが非mmu命令である場合にのみ、他のビットがアサートされる。その結果として、属性インタフェイス216,218から融合ロジック223に入力される値は、ステージ139,142内の命令が、(1)特定のレジスタ37を識別するか否か、(2)述語イネーブルされるか否か、及び(3)mmu命令又は非mmu命令であるか、を示すものとなる。

【0096】

属性インタフェイス214は、属性インタフェイス212,216,218とは異なり、各ラッチ147から3ビットの属性データを受信する。これらのビットのうち2ビットは、属性インタフェイス212,216,218により受信される属性ビットと同様に、実行ステージ136内の命令が述語イネーブルされるか否か、及び実行ステージ136内の命令がmmu命令であるか否かを示すものとなる。ここで、実行ステージ136内の命令が述語イネーブルされると、そのビットの1つがアサートされ、実行ステージ136内の命令がmmu命令であると、他のビットがアサー

トされる。属性インタフェイス214により受信される第3のビットは、そのアサート時に、実行ステージ136内の命令がALU命令であることを示すものとなる。

【0097】

図19に示すように、属性インタフェイス214は、m行のインタフェイスロジック224を備える。該インタフェイスロジック224の各行は、レジスタ37の異なる1つに対応し、3ビット出力を生成する。実行ステージ136内のプロデューサが述語イネーブルされない場合、又は該プロデューサのレジスタ識別子がインタフェイスロジック224の行に対応するレジスタ37を識別しない場合には、インタフェイスロジック224の行により出力されるビットはアサートされない。更に、実行ステージ136内のプロデューサがmmu命令である場合にのみ、インタフェイスロジック224の行により出力されるビットの1つがアサートされ、該プロデューサが非mmu命令である場合にのみ、該ビットのうちの別のビットがアサートされる。該プロデューサがALU命令である場合にのみ、最後のビットがアサートされる。その結果として、属性インタフェイス214から融合ロジック223に入力される値は、実行ステージ136内の命令が、(1)特定のレジスタ37を識別するか否か、(2)述語でイネーブルされるか否か、(3)mmu命令、非mmu命令、又はALU命令であるか、を示すものとなる。

【0098】

図20は、一行のインタフェイスロジック224を実施するために用いることができる典型的な回路を示す。ここで、各ANDゲート231がデコーダ157から1ビットを受信し、接続部232を介してラッチ147から属性データのビットの1つを受信する。デコーダ157からのビットは、インタフェイスロジック224の行が、デコーダ157により受信されデコードされたレジスタ識別子により識別されるレジスタ37に対応する場合にアサートされる。更に、接続部232を介して、各ANDゲート231により受信された属性データの上述のビットは、実行ステージ136内の命令が述語イネーブルされる場合にアサートされる。またANDゲート231の1つは、接続部233を介して、実行ステージ136内の命令がmmu命令であるか否かを示す属性データのビットを受信する。また別のANDゲート231は、上述の属性データのビットを反転させたものを受信し、残りのANDゲート231は、接続部234を介して、実行ステージ136内の命令がALU命令であるか否かを示す属性データのビットを受信する。

【0099】

図21に示すように、属性インタフェイス214,216,218の出力を受信する融合ロジック223は、複数のレジスタ37にそれぞれ対応するm行の融合ロジック237を備えている。該融合ロジック237の各行は、属性インタフェイス214内の一行のインタフェイスロジック224、属性インタフェイス216内の一行のインタフェイスロジック204、及び属性インタフェイス218内の一行のインタフェイスロジック204に接続され、そこからの出力を受信する。更に、融合ロジック237の同一行に接続されるインタフェイスロジック204,224の行は、融合ロジック237の行と同一のレジスタに対応する。このため、2つの属性インタフェイス214,216又は218に関連して該属性インタフェイスに接続されるステージ136,139又は142内の命令が、一致するレジスタ識別子を有し、及び何れも述語イネーブルされる場合にのみ、一行のインタフェイスロジック237が、2つの属性インタフェイス214,216又は218からのアサートされた出力を受信することが可能となる。

【0100】

図22に示すように、融合ロジック237の各行は、融合ロジック223により受信されるビット値を結合するORゲート241,243といったORゲートを備えている。ORゲート241は、属性インタフェイス214,216,218からの各ビットを受信し、該ビットがアサートされるとき、レジスタステージ133内の命令が第1のタイプである場合にデータハザードが存在することを示す。ORゲート243は、属性インタフェイス214,216,218からの各ビットを受信し、該ビットがアサートされるとき、レジスタステージ133内の命令が第2のタイプである場合にデータハザードが存在することを示す。

【0101】

例えば、上述のように、ステージ136,139又は142のうちの何れかのステージ内のmmuプロデューサは、レジスタステージ133内の非mmuコンシューマとのデータハザードを規定する

10

20

30

40

50

ことが可能である。このため、図 2 2 に示す実施形態では、ORゲート241は、ステージ136, 139又は142内の命令がmmu命令であるか否かを示す、属性インタフェイス214, 216及び218内のインタフェイスロジック204, 224の対応する行（即ち、図 2 2 に示す融合ロジック237の行に接続される属性インタフェイス214, 216, 218内のインタフェイスロジック204, 224の行）から各ビット値を受信することが好ましい。その結果として、ORゲート241の出力は、ステージ136, 139又は142内のプロデューサの1つが、（1）融合ロジック237の行に対応するレジスタ37を識別するレジスタ識別子と関連し、（2）述語イネーブルされ、及び（3）mmu命令である場合に、アサートされる。

#### 【0102】

更に、上記のように、ステージ136, 139又は142のうちの何れかのステージ内の非mmuプロデューサ、又は実行ステージ136内のALUプロデューサは、レジスタステージ133内の非mmuコンシューマとのデータハザードを規定することが可能である。このため、図 2 2 に示す実施形態では、ORゲート243は、ステージ136, 139又は142内の命令がmmu命令であるか否かを示す、属性インタフェイス214, 216, 218内のインタフェイスロジック204, 224の対応する行（即ち、図22に示す融合ロジック237の行に接続される属性インタフェイス214, 216, 218内のインタフェイスロジック204, 224の行）から各ビット値を受信し、実行ステージ136内の命令がALU命令であるか否かを示す、属性インタフェイス214内のインタフェイスロジック224の対応する行からのビットを受信することが好ましい。その結果として、ORゲート243の出力は、（1）ステージ136, 139又は142内の命令の1つが、（a）融合ロジック237の行に対応するレジスタ37を識別するレジスタ識別子と関連し、（b）述語イネーブルされ、及び（c）非mmu命令である場合にアサートされ、又は、（2）実行ステージ136内の命令が、（a）融合ロジック237の行に対応するレジスタ37を識別するレジスタ識別子と関連し、（b）述語イネーブルされ、及び（c）ALU命令である場合にアサートされる。

#### 【0103】

このため、図 1 5 及び図 1 8 により示されるハザード検出回路196を用いて、属性インタフェイス212及び融合回路223の出力を受信し、及び該出力に基づいてデータハザードを検出することが可能である。ここで、ハザード検出回路196は、複数のレジスタ37にそれぞれ対応するm行の比較ロジック211を備えている。比較ロジック211の各行は、比較ロジック211の行と同一のレジスタ37に対応する属性インタフェイス212内のインタフェイスロジック204の行からの出力を受信し、比較ロジック211の行と同一のレジスタ37に対応する融合ロジック237の行からの出力を受信する。このため、単一行の比較ロジック211は、レジスタステージ133内の述語イネーブルされた命令のレジスタ識別子が、ステージ136, 139又は142のうちの1つのステージ内の述語イネーブルされた命令のレジスタ識別子と一致する場合にのみ、属性インタフェイス212及び融合回路223の両方からのアサートされた出力を受信することができる。

#### 【0104】

更に、接続部211（図 1 6 ）上の信号は、レジスタステージ133内のコンシューマが、（1）比較ロジック211の行に対応するレジスタ37を識別するレジスタ識別子を有し、（2）述語イネーブルされ、及び（3）mmu命令である場合にのみ、アサートされることになる。接続部227上の信号は、ORゲート243（図 2 2 ）に接続されており、ステージ136, 139又は142のうちの1つのステージ内のプロデューサが、（1）比較ロジック211の行に対応するレジスタ37を識別するレジスタ識別子を有し、（2）述語イネーブルされ、及び（3）非mmu命令又はALU命令である場合にのみ、アサートされることになる。

#### 【0105】

接続部223（図 1 6 ）上の信号は、レジスタステージ133内のコンシューマが、（1）比較ロジック211の行に対応するレジスタ37を識別するレジスタ識別子を有し、（2）述語イネーブルされ、及び（3）非mmu命令である場合にのみ、アサートされることになる。更に、接続部229上の信号は、ORゲート241（図 2 2 ）に接続されており、ステージ136, 139又は142のプロデューサが、（1）比較ロジック211の行に対応するレジスタ37を識別するレジスタ識別子を有し、（2）述語イネーブルされ、及び（3）mmu命令である場合にのみ、アサ

10

20

30

40

50

ートされることになる。したがって、接続部231,233上の信号の一方は、システム100により処理されている命令のうちの2つの命令間にデータハザードが存在する場合にのみ、アサートされることになる。これらの2つの命令の一方は、上述の比較ロジック211の行に対応するレジスタ37を識別するレジスタ識別子を有するレジスタステージ133内のコンシューマである。このコンシューマは、データ誤りを防ぐために、停止されることが必要な場合もある。

【0106】

比較ロジック144及び／又は191にデータを入力する前に、デコーダ155,157,159及び／又は161(図9)により、並びに／又は属性インタフェイス212,214,216及び／又は218(図18)により生成されるデータを融合することにより、データハザードを検出するための比較ロジック144及び／又は191による比較回数をより低減させることができる。このため、多数のパイプライン132及び／又はレジスタ識別子に対してデータハザードを検出するために必要とされる回路及びその複雑さを著しく低減させることができる。

【0107】

更に、データハザードを検出する回路及びその複雑さを低減させるために、融合回路177,179及び／又は223(図9及び図18)を、例えば、「ワイヤードOR」構造といった「ワイドOR(wide OR)」構造により実施することが可能である。「ワイドOR」構造は、最小限の回路及びその複雑性を介して多数の入力信号のOR演算を行って一層少数の出力信号を生成する周知の構造である。このため、「ワイドOR」構造を用いて、デコーダ155,157,159及び／又は161(図9)により、並びに／又は属性インタフェイス212,214,216及び／又は218(図18)により生成されるデータを融合することにより、処理システム100に関連するデータハザードを検出するために必要とされる回路の規模を最小限にすることができる。

【0108】

図18は、同一のパイプライン132の命令間でレジスタ識別子と属性データとを比較する回路のみを示したものであることに留意されたい。しかし、図9に示す実施形態と同様に、図18の融合ロジック223及びハザード検出回路196に送信されるデータは、本発明の原理から逸脱することなく、パイプライン132の何れかから送信することが可能である。実際に、各パイプライン132が融合ロジック223及び／又はハザード検出回路196に同様にデータを送信して、システム100に関連する各データハザードをハザード検出回路196により検出できるようにすることが好ましい。しかし、ハザード検出回路196及び／又は融合ロジック223は、本書で説明したそれらの実施形態に変更を加えて、異なるパイプライン132から該融合ロジック223及び／又はハザード検出回路196へデータを送信するために必要となる付加的な接続部に適応させる必要がある場合もある。更に、融合ロジックは、図9の融合ロジック177と同様に、異なるパイプライン132のレジスタステージ133から送信されるデータを融合するよう実施する必要がある場合もある。

【0109】

本発明の上記の実施形態、特に何れかの「好ましい」実施形態は、本発明の単なる例示にすぎず、本発明の原理を明確に理解するために説明したものにすぎない、ということを強調しておく必要がある。本発明の思想及び原理から実質的に逸脱することなく、本発明の上記実施形態に対し、多くの変形及び修正を加えることが可能である。かかる全ての変形及び修正は、本開示及び本発明の範囲内において特許請求の範囲によって保護されるものとなることが意図されている。

【0110】

以下においては、本発明の種々の構成要件の組み合わせからなる例示的な実施態様を示す。

1. コンピュータプログラム(107)の命令を実行するシステム(100)であって、前記コンピュータプログラム(107)の前記命令を処理するよう構成された複数のパイプライン(132)と、  
該パイプライン(132)に接続された融合回路(177,179,223)であって、前記パイプライン(132)から複数のレジスタ識別子を同時に受信して該複数のレジスタ識別子を単一のレジス

10

20

30

40

50



タ識別子へと結合するよう構成されており、該単一のレジスタ識別子が、複数のビットを有すると共に、該融合回路(177,179,223)により結合された前記複数のレジスタ識別子のうちの少なくとも1つにより識別される各レジスタを識別するものであり、該融合回路(177,179,223)が、受信した前記複数のレジスタ識別子のうちの1つの少なくとも1つのビットを、受信した前記複数のレジスタ識別子のうちの別の1つの別のビットと結合して、前記単一のレジスタ識別子の前記複数のビットのうちの1つを生成する、融合回路(177,179,223)と、

前記融合回路(177,179,223)に接続されたハザード検出回路(164,196)であって、前記融合回路(177,179,223)から前記単一のレジスタ識別子を受信して該単一のレジスタ識別子を該ハザード検出回路が受信した他の情報と比較するよう構成され、更に、前記単一のレジスタ識別子と該ハザード検出回路が受信した前記他の情報との比較に基づき特定のタイプのデータハザードが存在するか否かを検出するよう構成されている、ハザード検出回路(164,196)と

を備えている、コンピュータプログラム(107)の命令を実行するシステム(100)。

2. 前記パイプライン(132)に接続された複数のデコーダ(155,157,159,161)を更に備えており、該複数のデコーダ(155,157,159,161)の各々が、前記複数のレジスタ識別子のうちの1つをエンコードされた形式でそれぞれ受信して該受信した前記1つのエンコードされたレジスタ識別子をデコードするよう構成されており、前記融合回路(177,179,223)が、前記デコーダ(155,157,159,161)を介して前記パイプライン(132)に接続されている、前項1に記載のシステム。

3. 前記複数のパイプライン(132)のうちの1つに接続され、前記複数のレジスタ識別子のうちの1つを受信するよう構成された、属性インタフェース(212,214,216,218)であって、属性データを受信すると共に該属性データに基づいて前記少なくとも1つのビットの値を制御するよう構成され、更に前記1つのレジスタ識別子を前記融合回路(177,179,223)に送信するよう構成されている、属性インタフェース(212,214,216,218)を更に備えている、前項1に記載のシステム。

4. 前記属性データが、複数の前記命令のうちの1つの命令のタイプを示すものである、前項3に記載のシステム。

5. 前記属性データが、複数の前記命令のうちの1つの命令の述語状態を示すものである、前項3に記載のシステム。

6. コンピュータシステム(105)においてデータハザードを検出するための方法であって、

コンピュータプログラム(107)の複数の命令を処理し、

該複数の命令のうちの1つと関連する第1のレジスタ識別子を受信し、該第1のレジスタ識別子が、複数のレジスタ(37)にそれぞれ対応する複数のビットを有しており、前記レジスタ(37)の各々が、前記第1のレジスタ識別子の前記複数のビットのうちの少なくとも1つに対応するようになっており、

前記命令のうちの別の命令と関連する第2のレジスタ識別子を受信し、該第2のレジスタ識別子が、前記複数のレジスタ(37)にそれぞれ対応する複数のビットを有しており、前記レジスタ(37)の各々が、前記第1のレジスタ識別子の前記複数のビットのうちの少なくとも1つに対応するようになっており、

前記第1のレジスタ識別子において、該第1のレジスタ識別子により識別される前記複数のレジスタ(37)のうちの1つに対応する前記複数のビットのうちの1つをアサートし、前記第1のレジスタ識別子により識別される前記1つのレジスタ(37)以外の何れかの前記レジスタ(37)に対応する前記第1のレジスタ識別子の前記各ビットをデアサートし、前記第2のレジスタ識別子において、該第2のレジスタ識別子により識別される前記複数のレジスタ(37)のうちの1つに対応する前記複数のビットのうちの1つをアサートし、前記第2のレジスタ識別子により識別される前記1つのレジスタ(37)以外の何れかの前記レジスタ(37)に対応する前記第2のレジスタ識別子の前記各ビットをデアサートし、前記第1のレジスタ識別子と前記第2のレジスタ識別子とを結合し、

10

20

30

40

50

該結合ステップに応じて第3のレジスタ識別子を生成し、該第3のレジスタ識別子が、前記第1及び前記第2のレジスタ識別子により識別される前記レジスタ(37)を識別するものであり、

前記第3のレジスタ識別子を別のレジスタ識別子と比較し、  
該比較ステップに基づきデータハザードを検出する、  
という各ステップを有する、コンピュータシステム(105)においてデータハザードを検出するための方法。

7. 前記第3のレジスタ識別子の生成ステップが、  
前記第1及び前記第2のレジスタ識別子の一方におけるアサートされたビットに基づいて前記第3のレジスタ識別子における複数の前記ビットのうちの1つをアサートする、  
というステップを含む、前項6に記載の方法。

10

8. 前記1つの命令のタイプを示す属性データを受信し、  
該属性データに基づいて前記第1のレジスタ識別子内の前記1つのビットの値を制御する、  
という各ステップを更に有する、前項6に記載の方法。

9. 前記1つの命令の述語状態を示す属性データを受信し、  
該属性データに基づいて前記第1のレジスタ識別子における前記1つのビットの値を制御する、  
という各ステップを更に有する、前項6に記載の方法。

#### 【図面の簡単な説明】

20

【図1】従来技術による処理システムを示すブロック図である。

【図2】図1に示すパイプラインを詳細に示すブロック図である。

【図3】図2に示すパイプラインの別の実施形態を示すブロック図である。

【図4】本発明の原理に従う処理システムを用いたコンピュータシステムを示すブロック図である。

【図5】図4に示す処理システムを示すブロック図である。

【図6】図5に示すパイプライン及び比較ロジックを詳細に示すブロック図である。

【図7】図6に示すANDロジックを詳細に示すブロック図である。

【図8】図7に示すANDロジックの別の実施形態を示すブロック図である。

【図9】図6のパイプラインと比較ロジックとの間で通信されるデータを融合するために用いることができる融合回路を示すブロック図である。

30

【図10】図9に示す融合回路の一部を詳細に示すブロック図である。

【図11】図9に示す融合回路の別の部分を詳細に示すブロック図である。

【図12】属性データを用いてデータハザードを検出する、本発明によるパイプライン及び比較ロジックの別の実施形態を示すブロック図である。

【図13】図12に示すデコーダ及び属性インタフェースを詳細に示すブロック図である。

【図14】図13に示すインタフェースロジックのうちの一行分を詳細に示すブロック図である。

【図15】図12に示すハザード検出回路を詳細に示すブロック図である。

40

【図16】図15に示す比較ロジックのうちの一行分を詳細に示すブロック図である。

【図17】図15に示すハザード検出回路における比較ロジックの各行の出力を結合するために用いられるORゲートを示すブロック図である。

【図18】図12のパイプラインと比較ロジックとの間で通信されるデータを融合するために用いることができる融合回路を示すブロック図である。

【図19】図18に示す属性インタフェースを詳細に示すブロック図である。

【図20】図19に示すインタフェースロジックのうちの一行分を詳細に示すブロック図である。

【図21】図18に示す融合回路を詳細に示すブロック図である。

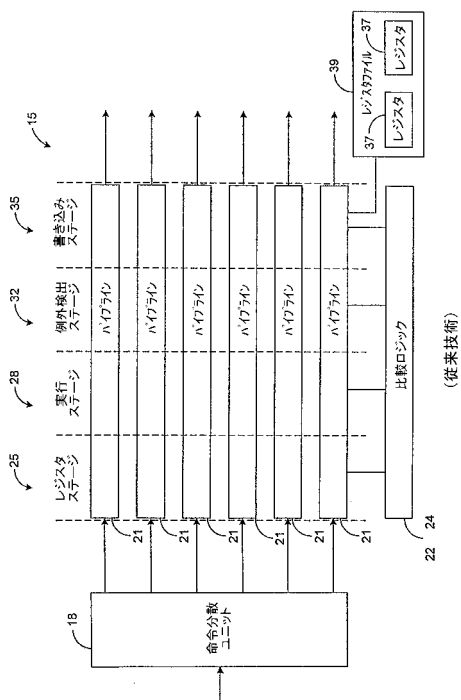
【図22】図21に示す融合回路のうちの一行分を詳細に示すブロック図である。

50

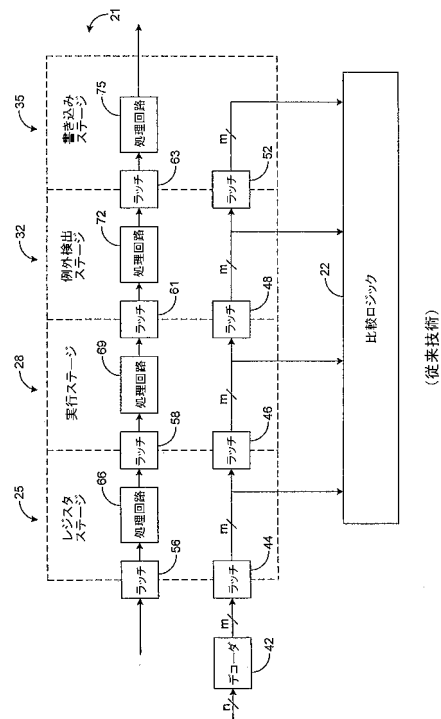
## 【符号の説明】

37	レジスタ
100	システム
107	コンピュータプログラム
132	パイプライン
155, 157, 159, 161	デコーダ
164, 196	ハザード検出回路
177, 179, 223	融合回路
212, 214, 216, 218	属性インタフェイス

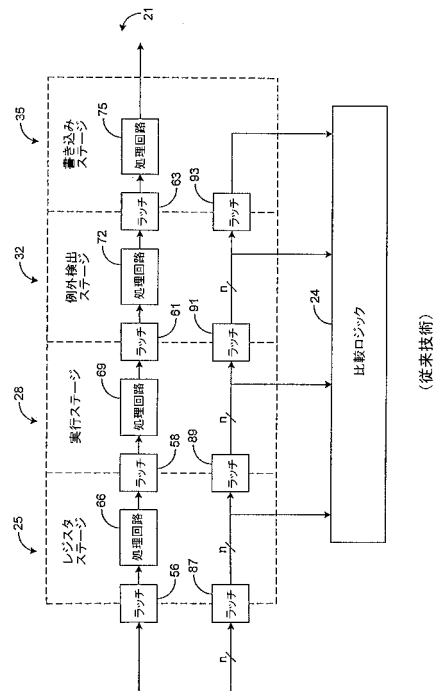
【図 1】



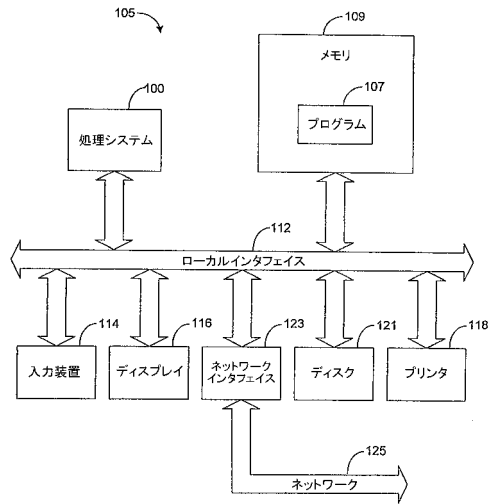
【図 2】



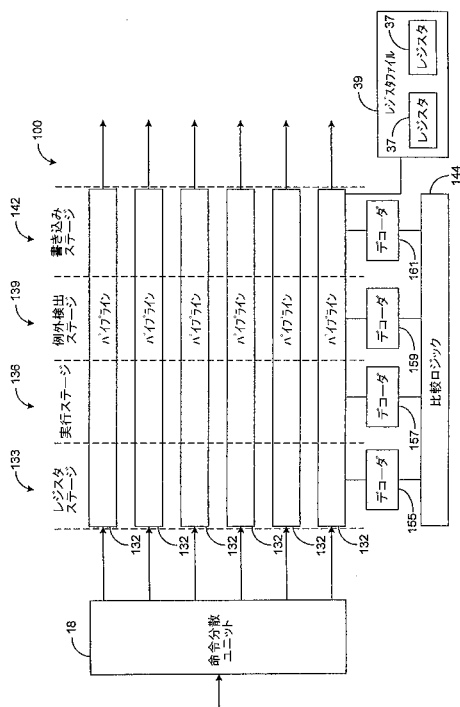
【図 3】



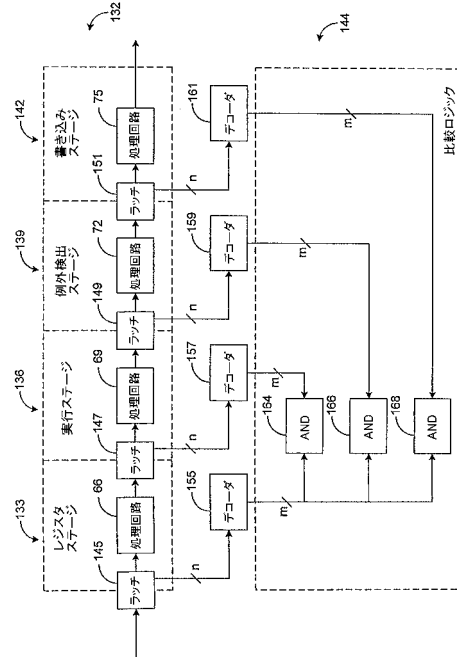
【図 4】



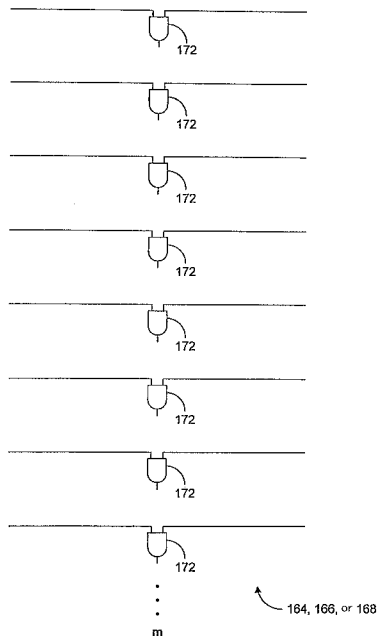
【図 5】



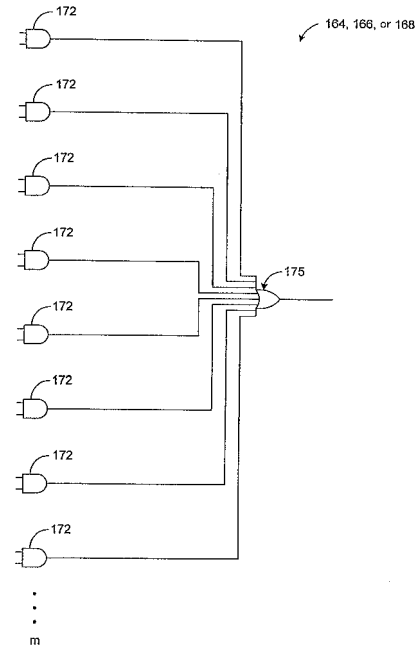
【図 6】



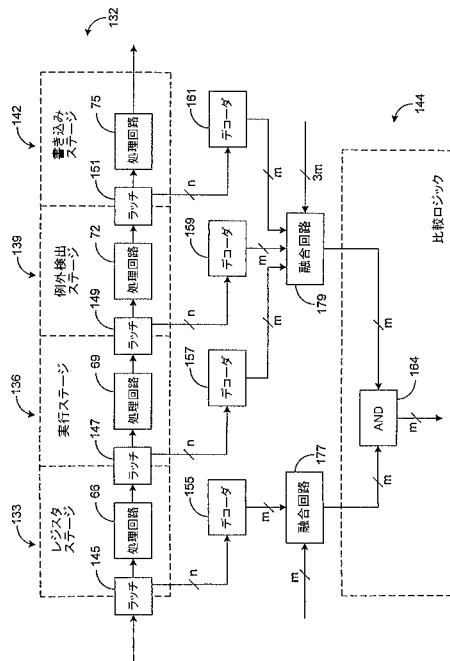
【図 7】



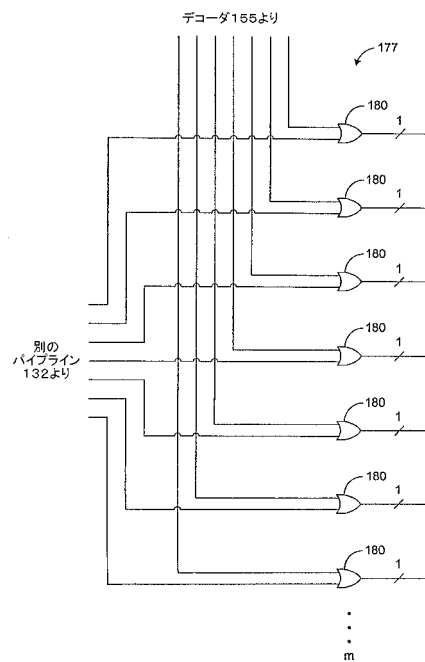
【図 8】



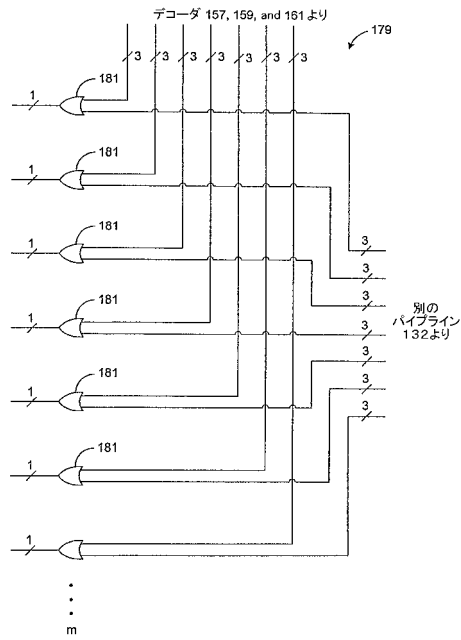
【図 9】



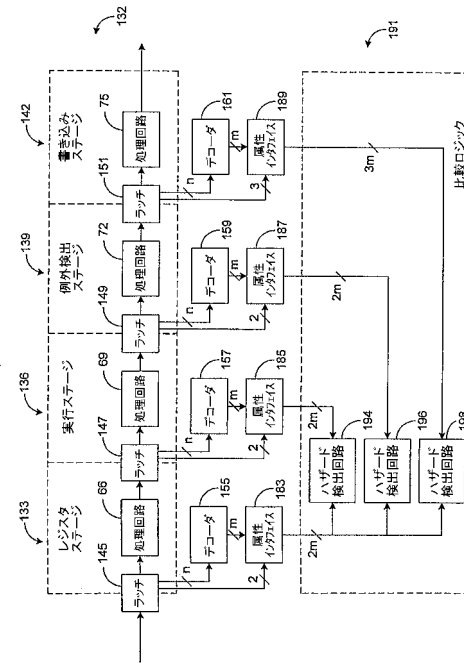
【図 10】



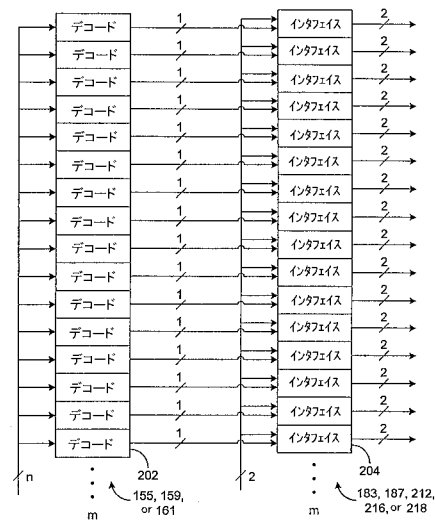
【図 1 1】



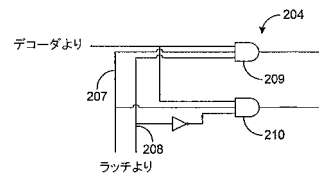
【図 1 2】



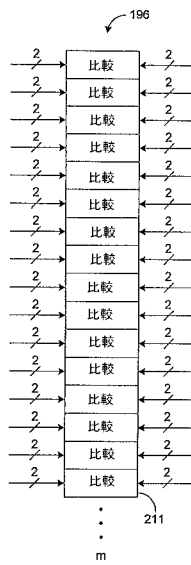
【図 1 3】



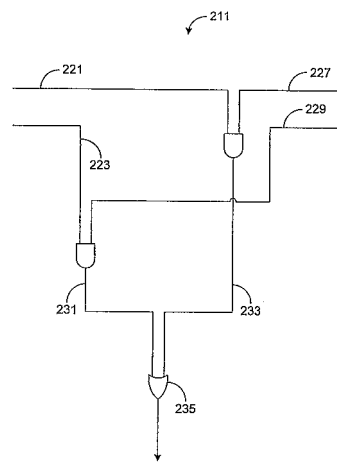
【図 1 4】



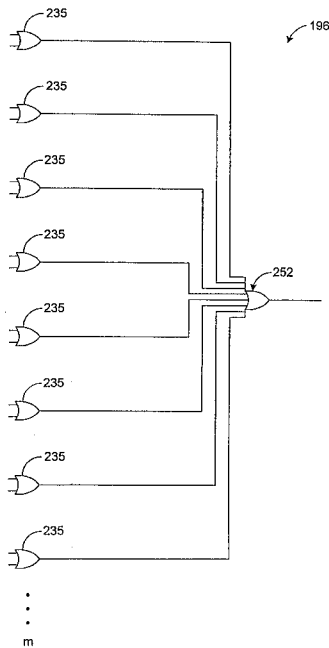
【図 15】



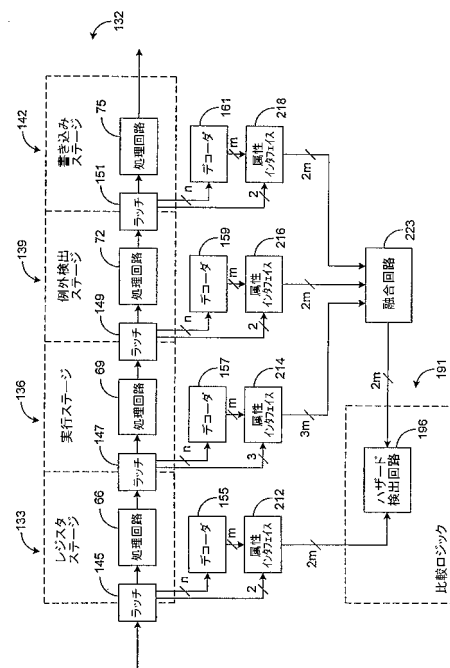
【図 16】



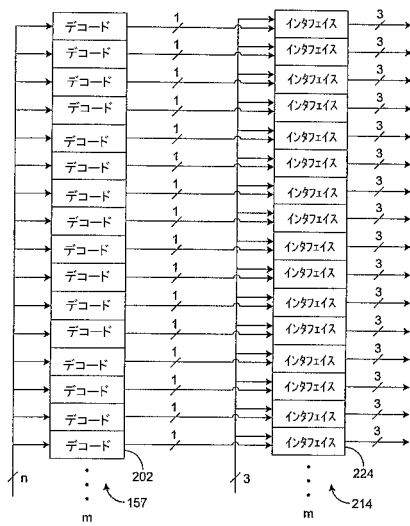
【図 17】



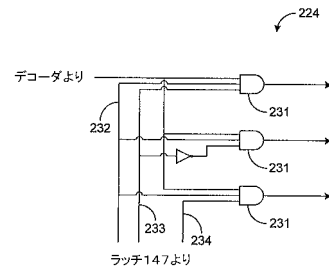
【図 18】



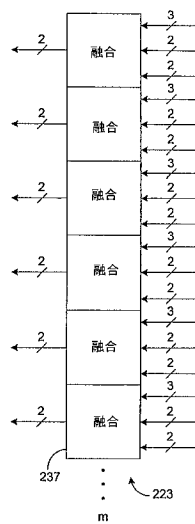
【図 19】



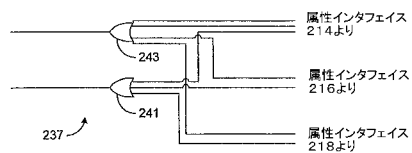
【図 20】



【図 21】



【図 22】





---

フロントページの続き

- (72)発明者 ロニー・リー・アーノルド  
アメリカ合衆国コロラド州 80528, フォートコリンズ, スティルウォーター・クリーク・ドライブ・2200
- (72)発明者 ドナルド・チャールズ・ソルティス・ジュニア  
アメリカ合衆国コロラド州 80526, フォートコリンズ, ローズゲート・コート・4414

審査官 後藤 彰

- (56)参考文献 特開平5-108348(JP, A)  
特開2001-209538(JP, A)  
特開2001-209536(JP, A)

- (58)調査した分野(Int.Cl.<sup>7</sup>, DB名)  
G06F 9/38