(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3631146号
(P3631146)
(45)発行日　平成17年3月23日(2005.3.23)　(24)登録日　平成16年12月24日(2004.12.24)

(51)Int.Cl.[7]　F I
G06F　9/38　G06F　9/38　350A

請求項の数 9　(全 20 頁)

(21)出願番号　特願2001-2037(P2001-2037)
(22)出願日　平成13年1月10日(2001.1.10)
(65)公開番号　特開2001-209536(P2001-209536A)
(43)公開日　平成13年8月3日(2001.8.3)
審査請求日　平成14年12月12日(2002.12.12)
(31)優先権主張番号　09/490390
(32)優先日　平成12年1月24日(2000.1.24)
(33)優先権主張国　米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT－PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100081721
弁理士　岡田　次生
(72)発明者　ロニー・リー・アーノルド
アメリカ合衆国80528コロラド州フォート・コリンズ、スティルウォーター・クリーク・ドライブ　2200

最終頁に続く

(54)【発明の名称】データハザードを検出するシステム

(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処理するコンピュータシステムであって、
各々が符号化レジスタ識別子を含んでいる複数の命令を同時に処理するように構成された少なくとも１つのパイプラインと、
前記パイプラインの第１ステージに結合された第１のデコーダであって、前記複数の命令のうち１つの命令の符号化レジスタ識別子を復号化して、コンピュータシステムに関連する複数のレジスタのうち１つに対応するアサートされた１ビットを含む第１の復号化レジスタ識別子にするように構成された第１のデコーダと、
前記パイプラインの第２ステージに結合された第２のデコーダであって、前記符号化レジスタ識別子を復号化して、コンピュータシステムに関連する前記複数のレジスタのうち１つに対応するアサートされた１ビットを含む第２の復号化レジスタ識別子にするように構成された第２のデコーダと、
前記第１および第２のデコーダとインタフェースされた比較ロジックであって、前記第１の復号化レジスタ識別子および前記第２の復号化レジスタ識別子を、前記命令のうち他の命令の符号化レジスタ識別子から復号化された他の復号化レジスタ識別子とそれぞれ比較するように構成された比較ロジックと、
を備えるコンピュータシステム。
【請求項２】
前記比較ロジックは、前記第１および第２の復号化レジスタ識別子の一方と前記他の復号

化レジスタ識別子のうちの１つとの該比較ロジックによる比較に基づいて、データ従属性ハザードを検出するようにさらに構成されている、請求項１に記載のコンピュータシステム。

【請求項３】

前記第１の復号化レジスタ識別子は複数のレジスタにそれぞれ対応する第１の複数ビットを含み、

前記複数のレジスタのうちの１つに対応する前記第１の複数ビットのうちの１ビットはアサートされ、該第１の複数ビットの残りのビットはデアサートされ、

前記第２の復号化レジスタ識別子は前記複数のレジスタにそれぞれ対応する第２の複数ビットを含み、

前記レジスタのうちの前記１つに対応する前記第２の複数ビットのうちの１ビットはアサートされ、該第２の複数ビットの残りのビットはデアサートされる、請求項１に記載のコンピュータシステム。

【請求項４】

前記パイプラインは、該パイプラインの第１ステージにおける前記複数の命令のうちの１つの命令と該パイプラインの第２ステージにおける前記複数の命令のうちの別の命令とを同時に処理するように構成されており、

前記第１のデコーダは前記第１ステージに結合され、前記第２のデコーダは前記第２ステージに結合されている、請求項１に記載のコンピュータシステム。

【請求項５】

前記パイプラインの第３ステージに結合され、前記複数の命令のうち別の命令の符号化レジスタ識別子を復号化して第３の復号化レジスタ識別子にするように構成された第３のデコーダをさらに含み、

前記比較ロジックは、前記第３のデコーダとインタフェースされるとともに、前記第３の復号化レジスタ識別子と前記第１の復号化レジスタ識別子とを比較するように構成されている、請求項１に記載のコンピュータシステム。

【請求項６】

第１ステージにおける第１の命令と第２ステージにおける第２の命令とを同時に処理するように構成されたパイプラインと、

前記第１ステージに結合され、前記第１の命令に関連する第１の符号化レジスタ識別子を受け取るとともに、該第１の符号化レジスタ識別子を復号化して第１の復号化レジスタ識別子にするように構成された第１のデコーダと、

前記第２ステージに結合され、前記第２の命令に関連する第２の符号化レジスタ識別子を受け取るとともに、該第２の符号化レジスタ識別子を復号化して第２の復号化レジスタ識別子にするように構成された第２のデコーダと、

前記第１および第２のデコーダとインタフェースされ、前記第１および第２の復号化レジスタ識別子を受け取るとともに、該第１および第２の復号化レジスタ識別子を、他の復号化された命令の符号化レジスタ識別子から復号化された他の復号化レジスタ識別子とそれぞれ比較するように構成された比較ロジックと、

を備えるコンピュータプログラムの命令を処理する処理システムであって、

前記第１の復号化レジスタ識別子と前記第２の復号化レジスタ識別子は、処理システムに関連する複数のレジスタうちの１つに対応するアサートされた１ビットをそれぞれ含む、処理システム。

【請求項７】

前記比較ロジックは、前記第１および第２の復号化レジスタ識別子の一方と前記他の復号化レジスタ識別子のうちの１つとの該比較ロジックによる比較に基づいて、データ従属性ハザードを検出するようにさらに構成されている、請求項６に記載の処理システム。

【請求項８】

前記第１の復号化レジスタ識別子は前記複数のレジスタにそれぞれ対応する第１の複数ビットを含み、

前記レジスタのうちの１つに対応する前記第１の複数ビットのうちの１ビットはアサートされ、該第１の複数ビットの残りのビットはデアサートされ、
前記第２の復号化レジスタ識別子は前記複数のレジスタにそれぞれ対応する第２の複数ビットを含み、
前記レジスタのうちの１つに対応する前記第２の複数ビットのうちの１ビットはアサートされ、該第２の複数ビットの残りのビットはデアサートされる、
請求項６に記載の処理システム。
【請求項９】
コンピュータプログラムの命令を処理するデータ従属性ハザードを検出する方法であって、
パイプラインを有する処理システムに複数の命令を送信するステップと、
前記複数の命令のうち１つの命令に関連する符号化レジスタ識別子を復号化することによって、第１の復号化レジスタ識別子を生成する一方、前記複数の命令のうち前記１つの命令は、前記パイプラインの第１部分によって処理される、ステップと、
前記符号化レジスタ識別子を復号化することによって第２の復号化レジスタ識別子を生成する一方、前記１つの命令は前記パイプラインの第２部分によって処理される、ステップと、
前記複数の命令のうち別の命令に関連する符号化レジスタ識別子を復号化することによって第３の復号化レジスタ識別子を生成するステップと、
前記複数の命令のうち別の命令に関連する符号化レジスタ識別子を復号化することによって第４の復号化レジスタ識別子を生成するステップと、
前記第１の復号化レジスタ識別子を前記第３の復号化レジスタ識別子と比較するステップと、
前記第２の復号化レジスタ識別子を前記第４の復号化レジスタ識別子と比較するステップと、
前記２つの比較ステップのうちの一方に基づいてデータ従属性ハザードを検出するステップと、を含み、
前記第１の復号化レジスタ識別子を生成するステップ及び前記第２の復号化レジスタ識別子を生成するステップが、それぞれ、前記処理システムに関連する複数のレジスタにそれぞれ対応する複数のビットを定義することと、前記実行ステップで利用されたレジスタに対応する前記ビットのうち１ビットをアサートすることを含む、
データ従属性ハザード検出方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、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命令を処理し、命令に関連するレジスタ識別子を比較することにより命令間のデータハザードを検出するシステムおよび方法に関する。
【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ている。パイプライン処理では、プロセッサは、複数命令を同時に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ている。従って、パイプラインにおける１つの命令の実行が、そのパイプラインにおける先行する命令の実行の結果が使用可能になる前に開始する場合があり、それによって、データ従属性ハザードからエラーが発生する可能性がある。
【０００３】
パイプラインによって実行されるある命令が他の命令の実行によって生成されたデータを利用する場合、データ従属性が存在し、他の命令によって生成されたデータをある命令によって使用可能でない場合に、そのデータ従属性によりデータ従属性ハザードが発生する。例えば、後の命令が、実行時に、前の命令の実行によって生成されるデータを利用する

場合（例えば、後の加算命令が、前のロード命令によって検索されたデータを利用する場合）がある。前の命令の実行からのデータが使用可能となる前に後の命令が実行されると、後の命令は誤ったデータを利用し、その結果データ従属性エラーとなる。従って、後の命令によって利用されるデータが使用可能となるまで、またはデータ従属性エラーが発生するまで、２つの命令間にデータ従属性ハザードが存在する。

【０００４】

【発明が解決しようとする課題】

言うまでもなく、データ従属性エラーを防止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。しかしながら、データ従属性ハザードを検出する回路は複雑なことが多く、プロセッサ内で相対的に大容量の領域を利用することが多々ある。これは、特に、命令を同時に実行する複数のパイプラインを含むスーパスカラプロセッサにおいて当てはまる。この場合、あるパイプラインにおける命令は、同じパイプラインにある他の命令との従属性を有するだけでなく、他のパイプラインにある他の命令との従属性をも有する可能性がある。従って、データ従属性ハザードを適切にチェックするために、あるパイプラインの第１の命令は、第１の命令とデータ従属性ハザードを共有する可能性のある各パイプラインにおける各命令と比較されなければならない。結果として、プロセッサ内のパイプラインの数が増大するにつれて、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な回路および複雑さが大幅に増大する。

【０００５】

このように、本業界において、コンピュータプログラムの命令間のデータハザードを検出するために必要な回路構成および複雑さを最小化するという未解決の要請が存在する。

【０００６】

【課題を解決するための手段】

本発明は、上述したような従来技術の欠点および欠陥を克服する。一般に、本発明は、コンピュータプログラムの命令を処理し、命令間のデータハザードを検出するシステムおよび方法を提供する。

【０００７】

構造上、本発明のシステムは、少なくとも１つのパイプラインと、第１のデコーダと、第２のデコーダと、比較ロジックと、を利用する。パイプラインは、コンピュータプログラムの命令を受取り、同時に処理する。第１のデコーダおよび第２のデコーダはパイプラインに接続され、パイプラインによって処理されている命令に関連するレジスタ識別子を復号化する。比較ロジックは、第１のデコーダおよび第２のデコーダとインタフェースされ、第１のデコーダおよび第２のデコーダによって生成される復号化レジスタ識別子を他の復号化レジスタ識別子とそれぞれ比較する。

【０００８】

本発明の別の特徴によると、各復号化レジスタ識別子は、複数のレジスタにそれぞれ対応する複数のビットを含む。特定のレジスタを識別するために、特定のレジスタに対応するビットがアサート（ａｓｓｅｒｔ）され、ビットの残りがデアサート（ｄｅａｓｓｅｒｔ）される。従って、比較ロジックは、２つの命令に関連する復号化レジスタ識別子が一致するか否かを判断することにより、データ従属性ハザードが２つの命令間に存在するか否かを判断することができる。

【０００９】

また、本発明は、コンピュータプログラムの命令を処理する方法を提供するものとして見ることもできる。本方法は、以下のステップにより広く概念化することができる。すなわち、処理システムのパイプラインに命令を送信するステップと、命令がパイプラインの第１の部分によって処理されている間に、命令に関連する符号化レジスタ識別子を復号化するステップと、命令がパイプラインの第２の部分によって処理されている間に、符号化レジスタ識別子を復号化するステップと、命令がパイプラインの第１の部分によって処理されている間に符号化レジスタ識別子を復号化する前記ステップに基づいて第１の復号化レジスタ識別子を生成するステップと、命令がパイプラインの第２の部分によって処理され

ている間に符号化レジスタ識別子を復号化する前記ステップに基づいて第２の復号化レジスタ識別子を生成するステップと、第１および第２の復号化レジスタ識別子を他の復号化レジスタ識別子と比較するステップと、該比較するステップに基づいてデータ従属性ハザードを検出するステップと、である。

【００１０】

本発明の他の特徴および利点は、当業者にとって以下の詳細な説明により明らかとなろう。かかるすべての特徴および利点は、本明細書において本発明の範囲内に含まれ、特許請求の範囲によって保護されるよう意図されている。

【００１１】

【発明の実施の形態】

本発明は、一般に、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検出することによって、データハザードからのエラーを防止することができるシステムおよび方法に関する。本発明の原理を説明するために、従来のスーパスカラ処理システム１５（以下、単に処理システム１５とする）を示す図１を参照する。処理システム１５は、コンピュータプログラムの命令を受取り各命令を複数のパイプライン２１のうちの１つに割当てる命令分配ユニット１８を含む。各パイプライン２１は、そのパイプライン２１が受取った各命令を実行するよう構成されている。

【００１２】

通常、各パイプライン２１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に構成されている。従って、命令分配ユニット１８は、各命令をその命令と適合したパイプライン２１にのみ割当てるよう構成されている。さらに図１では、比較ロジック２２または２４が、簡単のために１つのパイプライン２１に結合されているように示されているが、各パイプライン２１が比較ロジック２２または２４に同様に結合されているということに留意するべきである。

【００１３】

図１に示すように、パイプライン２１は典型的に命令を複数ステージで処理する。本明細書で使用されているように、「ステージ」は、命令を処理し、またその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答して制御できるように、その入力でラッチを含む、パイプライン２１のあらゆる部分である。図１に示すパイプライン２１は、命令を４つのステージで処理する。すなわち、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および書込みステージ３５である。他の実施の形態では、処理システム１５は、他のタイプのステージおよびそのステージの組合せで命令を処理することが可能である。

【００１４】

図１に示す処理システム１５において、パイプライン２１の１つによって受取られる命令は、まずレジスタステージ２５において処理される。レジスタステージ２５では、命令の実行に必要なあらゆるオペランドが取得される。オペランドが取得されると、命令は、命令を実行する実行ステージ２８に入る。実行ステージ２８で命令が実行された後、その命令は、例外検出ステージ３２に入る。例外検出ステージ３２は、例えばデータの低信頼性を示す実行中のオーバラン等の状態をチェックする。例外検出ステージ３２が完了した後、命令は書込みステージ３５に入る。書込みステージ３５は、レジスタまたはメモリのロケーションに、実行ステージ２８の結果を書込む。

【００１５】

典型的に、パイプライン２１の各ステージ２５、２８、３２、３５は、一度に１つの命令のみを処理し、ステージ２５、２８、３２、３５がそれぞれの命令を同時に処理することで各パイプライン２１は複数の命令を処理することができる。例えば、図１に示す処理システム１５において、パイプライン２１の１つは４つの命令を同時に処理することができ、そのときパイプライン２１の各ステージ２５、２８、３２、３５は４つの命令のうちの１つを処理している。さらに、各パイプライン２１は、他のパイプライン２１が他の命令を処理している時に命令を処理することができる。従って、図１に示す処理システム１５

により、相対的に大量の命令を同時に処理することが可能である。

【００１６】

タイミングを制御するために、命令は典型的にクロック信号のエッジに応答してステージ２５、２８、３２、３５を通してステップする。例えば、書込みステージ３５における命令は、レジスタステージ２５、実行ステージ２８および例外検出ステージ３２における命令が、それぞれ同じパイプライン２１の実行ステージ２８、例外検出ステージ３２および書込みステージ３５にステップインするのと同じクロックエッジで、パイプライン２１からステップアウトする。しかしながら、典型的に、ステージがデータ従属性エラー無しに命令を完全に処理することができるまで、その命令はステージ２５、２８、３２または３５からステップアウトしないようになっている。

【００１７】

この点で、ステージ２５、２８、３２または３５における命令の処理がクロック信号の次のアクティブなエッジに先立ちデータ従属性エラーなしに完了することができない場合は、ステージ２５、２８、３２または３５において命令をストールさせる必要のあるときがある。例えば、例外検出ステージ３２および書込みステージ３５における命令の処理がデータ従属性エラーなしに完了し、これらステージ３２、３５における命令が、クロック信号の次のエッジでそれぞれそのステージ３２、３５からステップして出る場合がある。しかしながら、同じパイプライン２１の実行ステージ２８における命令が、次のクロックエッジが発生する前に、使用できない（例えば、先のロード命令によってまだ検索されているような）データを利用する場合がある。従って、命令の処理が次のクロックエッジが発生する前にエラー無しに完了することができないので、実行ステージ２８における命令は、次のクロックエッジにおいて実行ステージ２８からステップアウトされない。言い換えれば、実行ステージ２８における命令はストールされる。この命令は、実行ステージ２８がデータ従属性エラーなしに命令を完全に処理することができるまで、ストールされたままでなければならない。１９９９年９月７日に出願された「Ｓｕｐｅｒｓｃａｌａｒ Ｐｒｏｃｅｓｓｉｎｇ Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｆｏｒ Ｅｆｆｉｃｉｅｎｔｌｙ Ｐｅｒｆｏｒｍｉｎｇ Ｉｎ－Ｏｒｄｅｒ Ｐｒｏｃｅｓｓｉｎｇ ｏｆ Ｉｎｓｔｒｕｃｔｉｏｎｓ」と題する、出願番号第０９／３９０，１９９号の米国特許出願は、データ従属性ハザードを防止するために命令をストールする適切な処理をより詳細に述べており、参照により本明細書に援用される。

【００１８】

各ステージ２５、２８、３２、３５は、通常、１度に１つの命令のみを処理するよう構成されているため、上述した例において同じパイプライン２１のレジスタステージ２５における命令もまた、少なくとも実行ステージ２８における命令が実行ステージ２８からステップアウトすることができるようになるまで、レジスタステージ２５においてストールされていなければならない。従って、ステージ２５、２８、３２または３５のうち一つにおいて命令がストールされている時、同じパイプライン２１によって処理されている後の命令もまた、後の命令がそれぞれ次のステージ２８、３２または３５にステップインする用意ができている場合であっても、ストールされていなければならない。

【００１９】

命令がパイプライン２１を通ってステップするにつれて、命令のうちの少なくとも１つをストールすることによって、データ従属性ハザードを検出してデータ従属性エラーを防止することが通常望ましい。通常、データ従属性ハザードは、データ従属性を検出することによって、およびデータ従属性に関連するデータがまだ使用可能でないと判断することによって検出される。従って、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、２つの判断が通常行われる。すなわち、１）２つの命令間にデータ従属性が存在するか、および２）データ従属性に関連するデータ（すなわち、命令のうちの１つによって生成され他の命令によって利用されるデータ）が使用可能であるか、である。

【００２０】

２つの命令間のデータ従属性は、通常、命令に関連するレジスタ識別子を比較することに

よって検出される。この点で、各命令は通常、その命令によりどのレジスタが使用され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、データを処理する（例えば、書込む）命令（「プロデューサ」と呼称する）は、プロデューサを実行する時にデータが書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、格納されたデータまたはプロデューサによって生成されるデータを利用する（例えば、検索する）命令（「コンシューマ」と呼称する）は、コンシューマを実行する時にデータが検索されるべきレジスタを識別するレジスタ識別子を含む。パイプライン２１のうちの１つにおける後のステージ２８、３２または３５が、前のステージ２５、２８または３２によって処理されているコンシューマと同じ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、２つの命令間にはデータ従属性が存在する。さらに、プロデューサが生成しコンシューマが使用するデータがコンシューマにはまだ使用不可能である場合、２つの命令間のデータ従属性はデータ従属性ハザードを生成する。

【００２１】

本明細書において、命令が他のステージによって処理された後に、あるステージで処理された場合、そのステージは他のステージより「後」である。例えば、図１において、実行ステージ２８はレジスタステージ２５より「後」であり、レジスタステージ２５は実行ステージ２８より「前」である。

【００２２】

なお、大部分の命令は、実行中にデータの検索とデータの格納を共に行うという点で、コンシューマでもプロデューサでもあることに留意するべきである。本明細書で使用されているように、「プロデューサ」と呼称する命令のレジスタ識別子は、その命令が別のレジスタからデータを検索し、そのため別のレジスタ識別子に関連付けられる場合があっても、その命令がデータを格納するレジスタを識別する。さらに、「コンシューマ」と呼称する命令のレジスタ識別子は、その命令が別のレジスタにデータを格納し、そのため別のレジスタ識別子に関連付けられる場合があっても、その命令がデータを検索するレジスタを識別する。

【００２３】

コンシューマのオペランドはレジスタステージ２５で取得されるので、レジスタステージ２５における各コンシューマのレジスタ識別子は、通常、後のステージ２８、３２、３５における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成するあらゆるデータ従属性がレジスタステージ２５におけるコンシューマに対して存在するか否かが判断される。このような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は、命令と共にパイプライン２１内を通ってステップすることもある。

【００２４】

ここで、レジスタ識別子は、通常ｎビットの符号化された値であるが、図２の従来のシステムによって示すように、まずデコーダ４２によってｍビット値（ｍは通常2^nである）に復号化される。値ｍは、処理システム１５に関連するレジスタの数に対応し、ｍビットレジスタ識別子の各ビットは、特定のレジスタに対応する。デコーダ４２は、ｎビットレジスタ識別子がいずれのレジスタを識別するかを判断し、識別したレジスタに対応するｍビットレジスタ識別子のビットをアサートするよう構成されている。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。従って、あるレジスタに関連するビットがアサートされると、ｍビットレジスタ識別子はレジスタのうちのそのレジスタを識別する。

【００２５】

ｍビットレジスタ識別子は、そのレジスタ識別子に関連する命令がラッチ５６、５８、６１、６３によってそれぞれステージ２５、２８、３２、３５にラッチされるのと同じエッジで、ラッチ４４、４６、４８、５２によってそれぞれステージ２５、２８、３２、３５にラッチされる。従って、レジスタ識別子とその関連する命令は、同時に同じステージ２５、２８、３２、または３５に存在しなければならない。図２に示すように、レジスタ識別子に関連する命令は、ステージ２５、２８、３２、３５にそれぞれラッチされると、処

理回路６６、６９、７２、７５によって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について以下に述べる機能をそれぞれ実行する。すなわち、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は例外に対してチェックを行い、処理回路７５は命令の実行によって生成されたデータをレジスタまたはメモリのロケーションに書込む。

【００２６】

ステージ２５、２８、３２、３５におけるｍビットレジスタ識別子は、比較ロジック２２に送信される。比較ロジック２２は、ｍビットレジスタ識別子を比較して、ｍビットレジスタ識別子に関連する命令のいずれかの間にデータ従属性が存在するか否かを判断するよう構成されている。この場合、レジスタステージ２５におけるコンシューマが後のステージ２８、３２、３５におけるプロデューサのいずれかとのデータ従属性を有しているか否かを判断するために、比較ロジック２２は、レジスタステージ２５におけるコンシューマの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および書込みステージ３５におけるプロデューサのｍビットレジスタ識別子の各々と比較する。ステージ２８、３２または３５のいずれかにおけるプロデューサの比較したｍビットレジスタ識別子のいずれかが、レジスタステージ２５におけるコンシューマのｍビットレジスタ識別子と一致する場合、比較ロジック２２は、２つの一致する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と判断する。さらに、レジスタステージ２５におけるコンシューマとのデータ従属性を有するステージ２８、３２または３５における命令のうちの１つによって生成されたデータが、レジスタステージ２５におけるコンシューマによってまだ使用可能でない場合、比較ロジック２２は、２つの命令間にデータ従属性ハザードが存在すると判断する。

【００２７】

図３は、パイプライン２１の１つにおいて命令を処理し、データ従属性ハザードを生成するデータ従属性をチェックするために従来から使用されている回路の別の実施の形態である。ｎビットの符号化レジスタ識別子に関連する命令は、上述したように、ステージ２５、２８、３２、３５にラッチされ、処理回路６６、６９、７２、７５によって処理される。しかしながら、ｎビットレジスタ識別子は、図２のようにデコーダ４２によって復号化されない。代わりに、ｎビットレジスタ識別子は、命令がそれぞれステージ２５、２８、３２、３５にラッチされるのと同じエッジで、それぞれラッチ８７、８９、９１、９３によって、ステージ２５、２８、３２、３５にラッチされる。従って、ｎビットレジスタ識別子は、ｎビットレジスタ識別子に関連する命令と共に、ステージ２５、２８、３２、３５を通ってステップする。

【００２８】

各ステージ２５、２８、３２、３５におけるｎビットレジスタ識別子は、比較ロジック２４に送信される。比較ロジック２４は、レジスタ識別子を比較して、ｎビットレジスタ識別子に関連する命令のいずれかの間にデータ従属性ハザードを規定するデータ従属性が存在するか否かを判断するよう構成されている。この場合、レジスタステージ２５におけるコンシューマが、後のステージ２８、３２および／または３５におけるプロデューサのいずれかとのデータ従属性を有するか否かを判断するために、比較ロジック２４は、レジスタステージ２５におけるコンシューマの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および書込みステージ３５におけるプロデューサのｎビットレジスタ識別子の各々と比較する。ｎビットレジスタ識別子が符号化されているため、比較ロジック２４は、ｎビットコンパレータを使用して、レジスタステージ２５におけるレジスタ識別子が、ステージ２８、３２および／または３５におけるレジスタ識別子のいずれかと一致するか否かを判断する。ステージ２８、３２または３５のいず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が、レジスタステージ２５におけるコンシューマのｎビットレジスタ識別子と一致する場合、比較ロジック２４は、２つの一致する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と判断する。さらに、レジスタステージ２５におけるコンシューマとデータ従属性を有する、ステージ２８、３２または３

５における命令のうちの１つによって生成されるデータが、レジスタステージ２５におけるコンシューマによって使用可能でない場合、比較ロジック２４は、２つの命令間にデータ従属性ハザードが存在すると判断する。

【００２９】

なお、図２および図３に示す回路は、簡単のために、単一のパイプライン２１における命令のレジスタ識別子のみをステージ２５、２８、３２、３５を通ってステップさせていることに留意されたい。さらに、上述した回路は、パイプライン２１によって処理される各命令に対する単一のレジスタ識別子のみを処理する。多くの命令は複数のレジスタ識別子を含むので、同じ命令に対する複数のレジスタ識別子を扱うために、必要に応じて追加の回路を実装しなければならない。さらに、レジスタステージ２５におけるコンシューマのレジスタ識別子は、同じパイプライン２１のステージ２８、３２および／または３５におけるプロデューサのレジスタ識別子と比較されなければならないだけでなく、いずれかの他のパイプライン２１における後のステージ２８、３２および／または３５のいずれかにおける各プロデューサのレジスタ識別子とも比較されなければならない。従って、パイプライン２１の数が増加するにつれて、データ従属性ハザードを検出するための配線および他の回路は大幅に増加する。

【００３０】

特に、プロデューサが生成したデータがコンシューマに使用可能となるために必要な待ち時間が一様でも一定でもない場合、データ従属性ハザードを検出しデータ従属性エラーを予防するために必要な回路は、相対的に複雑である。さらに、図２において、処理システム１５の性能が向上するに従い、通常、レジスタの数（ｍ）は相対的に大きくなる。従って、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（すなわち、ステージからステージにｍビットレジスタ識別子を送信する）ために必要な配線の数は相対的に大きい（例えば数百）。この追加の配線は、処理システム１５内の貴重な空間を利用し、さらに、処理システム１５の配線設計全体を複雑にする。

【００３１】

加えて、図３の符号化レジスタ識別子を比較するｎビットコンパレータ（ｎビットコンパレータ用の配線を含む）もまた、処理システム１５内の貴重な空間を利用し、処理システム１５の配線設計の複雑さを増大させる。さらに、ステージ２５、２８、３２、３５を通してレジスタ識別子に関連するｍビットレジスタ識別子および命令を別々にラッチすることにより、ラッチの数が増大し、その結果、処理システム１５を実現するために必要な回路および空間の量が増大する。複雑さおよび空間に関連する問題は、パイプライン２１の数が増加するに従って悪化する。

【００３２】

一般に、本発明は、レジスタ識別子を比較し、それによってデータ従属性ハザードを検出するために必要な配線および他の回路の量を低減するシステムおよび方法を提供する。図４は、本発明の好ましい実施の形態の原理に従う処理システム１００を示す。図４に示すように、処理システム１００は、メモリ１０９に格納されたコンピュータプログラム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できる。

【００３３】

処理システム１００は、１つまたは複数のバスを含むことができるローカルインタフェース１１２を介して、コンピュータシステム１０５内の他の要素と通信しかつそれらを駆動する。さらに、例えばキーボードまたはマウス等の入力装置１１４を用いて、コンピュータシステム１０５のユーザからデータを入力することができ、スクリーンディスプレイ１１６またはプリンタ１１８を用いて、ユーザにデータを出力することができる。ディスク格納機構１２１はローカルインタフェース１１２に接続されて、不揮発性ディスク（例えば、磁気、光等）に対しおよび不揮発性ディスクからデータを転送することができる。コンピュータシステム１０５は、コンピュータシステム１０５をネットワーク１２５とデータを交換することができるようにするネットワークインタフェース１２３に接続すること

が可能である。

【００３４】

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回路以外は、処理システム１００の構成は従来の処理システム１５の構成と同じであることが好ましい。従って図５において、処理システム１００は、従来からの処理システム１５について上述したように、パイプライン１３２を介してレジスタステージ１３３、実行ステージ１３６、例外検出ステージ１３９および書込みステージ１４２において命令を処理する。なお、望ましい場合、他のステージおよびステージの他の組合せを介して、パイプライン１３２によって実行される処理を分割することも可能であることに留意するべきである。さらに、図５において、比較ロジック１４４は、簡単のためにデコーダ１５５、１５７、１５９、１６１を介して１つのパイプライン１３２に結合されているように示されているが、好ましい実施の形態では、各パイプライン１３２が同様のデコーダを介して同様に比較ロジック１４４に結合されていることに留意するべきである。

【００３５】

図６に示すように、処理システム１００は、処理システム１００のタイミングを制御するために使用されるラッチ１４５、１４７、１４９、１５１を含む。この場合、本技術分野における周知の技術により、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。従来の処理システム１５と同様に、ラッチ１４５、１４７、１４９、１５１は好ましくはステージ１３３、１３６、１３９、１４２の各々の処理回路６６、６９、７２、７５の各々が１度に１つの命令のみを処理するよう制御される。さらに、図６に示すパイプライン１３２は、処理回路６６、６９、７２、７５の各々について１命令ずつ、４つまでの命令を同時に処理することができる。しかしながら、他の実施の形態では、処理回路６６、６９、７２および／または７５のいずれもが１度に複数の命令を同時に処理することができてもよい。

【００３６】

ラッチ１４５、１４７、１４９、１５１は、ラッチ１４５、１４７、１４９、１５１によってラッチされている命令によってそれぞれ定義される符号化ｎビットレジスタ識別子を、デコーダ１５５、１５７、１５９、１６１にそれぞれ送信するよう構成されている。従って、ラッチ１４５、１４７、１４９、１５１の各々が、処理回路６６、６９、７２、７５にそれぞれ命令を送信する時、ラッチ１４５、１４７、１４９、１５１の各々もまた、各命令によって定義される符号化ｎビットレジスタ識別子をデコーダ１５５、１５７、１５９、１６１に送信する。

【００３７】

デコーダ１５５、１５７、１５９、１６１の各々は、受取ったｎビットレジスタ識別子を、受取ったｎビットレジスタ識別子と同じレジスタを識別するｍビットレジスタ識別子に復号化するよう設計されている。値ｍは、好ましくは、処理システム１００に関連するレジスタの数に対応しており、各ｍビットレジスタ識別子の各ビットは、特定のレジスタに対応する。各デコーダ１５５、１５７、１５９、１６１は、好ましくは受取ったｎビットレジスタ識別子によっていずれのレジスタが識別されているかを判断し、識別されたレジスタに対応するｍビットレジスタ識別子のビットをアサートするよう構成されている。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。その結果、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ビットの残りとを有しているはずである。アサートされたビットは、受取ったｎビットレジスタ識別子を含む命令によって利用されるレジスタに対応する。しかしながら、ｎビットレジスタ識別子を復号化するために他の技術を採用してもよく、また、命令に関連するレジスタを識別するために、他の識別子を使用してもよい。「Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｆｏｒ Ｕｔｉｌｉｚｉｎｇ Ｉｎｓｔｒｕｃｔｉｏｎ Ａｔｔｒｉｂｕｔｅｓ ｔｏ Ｄｅｔｅｃｔ Ｄａｔａ Ｈａｚａｒｄｓ」と題され（代理人整理番号第１０９７１３３３号）、本発明の発明者によって出願された米国特許出願は、デコーダ１５５、１５７、１５９および／または１６１を実現するために使用できるデコーダを記述しており、参照により本明細

書に援用される。

【００３８】

各デコーダ１５５、１５７、１５９、１６１は、復号化ｍビットレジスタ識別子を比較ロジック１４４に送信する。比較ロジック１４４は、受取ったｍビットレジスタ識別子の各々を、少なくとも１つの他の受取ったｍビットレジスタ識別子と比較して、データ従属性および／またはデータ従属性ハザードが存在するか否かを判断する。２つの命令間のデータ従属性ハザードは、２つの命令間にデータ従属性が無ければ存在しないことに留意されたい。従って、データ従属性ハザードの検出は、データ従属性の検出としても考慮しなければならない。

【００３９】

比較ロジック１４４の機能を説明するために、従来の処理システム１５と同様に、レジスタステージ１３３におけるコンシューマが、他のステージ１３６、１３９または１４２のいずれかにおけるあらゆるプロデューサとデータ従属性ハザードを有するか否かを判断することが望ましいと想定する。この例では、コンシューマはレジスタステージ１３３にあり、プロデューサは他のステージ１３６、１３９、１４２にあるものとすると、デコーダ１５５によって復号化されるｍビットレジスタ識別子は、デコーダ１５７、１５９、１６１によって復号化されるｍビットレジスタ識別子の各々と比較されなければならない。

【００４０】

従って、図６に示すように、デコーダ１５５によって復号化されるｍビットレジスタ識別子は、ＡＮＤロジック１６４、１６６、１６８に送信され、デコーダ１５７、１５９、１６１によって復号化されるｍビットのレジスタ識別子は、各々ＡＮＤロジック１６４、１６６、１６８に送信される。そして、ＡＮＤロジック１６４、１６６、１６８の各々は、受取ったｍビットレジスタ識別子を比較して、レジスタステージ１３３において処理されているコンシューマと、他のステージ１３６、１３９、１４２のいずれかにおいて処理されているプロデューサのいずれかとの間に、データ従属性が存在するか否かを判断する。レジスタステージ１３３におけるコンシューマと、まだ自身のデータを生成していない（すなわち、コンシューマに対して使用可能なデータを作成していない）他のステージ１３６、１３９、１４２におけるプロデューサのいずれか１つとの間にデータ従属性がある場合、比較ロジック１４４は、コンシューマと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と判断する。そして、制御回路（図示せず）は、かかる判断に応答してレジスタステージ１３３においてコンシューマをストールする（または他の処置をとる）ことにより、データ従属性エラーを防止する。

【００４１】

図７は、ＡＮＤロジック１６４、１６６、１６８を実現するために適したロジックを示す。この場合、各ＡＮＤロジック１６４、１６６、１６８は、各々が受取ったｍビットレジスタ識別子のうちの１つからのビットと、他の受取ったｍビットレジスタ識別子からのビットとを受取る、ｍ個のＡＮＤゲート１７２を含む。各ＡＮＤゲート１７２によって受取られる両ビットは、好ましくは、同じレジスタに対応する。従って、２つの受取ったｍビットのレジスタ識別子が一致すると、ＡＮＤゲート１７２の１つはアサートされた出力を生成しなければならない。従って、ＡＮＤゲート１７２の出力を分析することができ、ＡＮＤゲート１７２の出力のいずれかがアサートされると、一致しているｍビットレジスタ識別子に関連する命令間に、データ従属性が存在すると判断することができる。ＡＮＤゲート１７２の出力を迅速に分析するために、出力の各々をＯＲゲート１７５（図８）に送信することができる。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。

【００４２】

処理システム１００の設計の結果として、データ従属性、さらにはデータ従属性ハザードを検出するために必要な、配線を含む回路および空間の量が、従来の処理システム１５と比較して低減される。特に、図２においてステージからステージへレジスタ識別子を送信するために使用される配線の数を、大幅に低減することができ、比較ロジック１４４の実

現を、図３の比較ロジック２４について述べたようなｎビットコンパレータによる実現より大幅に単純かつ小さくすることができる。加えて、デコーダ１５５、１５７、１５９、１６１および比較ロジック１４４の実現に使用される装置は、空間および配線の複雑さを最小化するように容易に配置することができる。さらに、デコーダ１５５、１５７、１５９、１６１が使用する空間は、ラッチ４４、４６、４８、５２（図２）より少なくすることができる。その結果、データ従属性および／またはデータ従属性ハザードを検出するシステム１００の回路を、従来の処理システム１５の回路より大幅にコンパクトかつ効率的にすることができる。

【００４３】

図６は、パイプライン１３２における各命令からの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するべきである。しかしながら、命令が複数のレジスタ識別子を含むことも可能である。従って、図６の回路と同様に、各命令の各レジスタ識別子を確実に検査できるようにするために、追加の回路を実装してもよい。例えば、レジスタステージ１３３におけるコンシューマは、実行時に２つの異なるレジスタからデータを検索してもよく、そのため、２つの異なるレジスタ識別子を含んでもよい。かかる命令に適応するために、レジスタステージ１３３は、好ましくは、比較ロジックによって他のステージ１３６、１３９、１４２における他のｍビットレジスタ識別子と比較することができるｍビットレジスタ識別子に他のレジスタ識別子を復号化する、別のデコーダ１５５を含む。従って、図６に示す設計を必要に応じて拡張することにより、複数のレジスタ識別子に関連する命令に適応させることができることは当業者には明らかであろう。

【００４４】

さらに、図６は、単一のパイプライン１３２の回路を示す。スーパスカラシステム１００では、異なるパイプライン１３２における命令間のデータ従属性ハザードを、ゆえにデータ従属性をチェックすることが望ましい。従って、本明細書で述べられている方法に従って、あるパイプライン１３２における命令のｍビットレジスタ識別子を、同じまたは他のパイプライン１３２における命令のｍビットレジスタ識別子と比較することによって、異なるパイプライン１３２の命令間のデータ従属性および／またはデータ従属性ハザードをチェックすることができることは当業者には明らかであろう。例えば、あるパイプライン１３２のデコーダ１５５によって復号化されるｍビットレジスタ識別子を、ＡＮＤロジック１６４、１６６、１６８と同様のロジックを介して、他のパイプライン１３２のデコーダ１５７、１５９、１６１によって復号化されるｍビットレジスタ識別と比較して、あるパイプライン１３２のレジスタステージ１３３における命令と他のパイプライン１３２のステージ１３６、１３９、１４２における命令との間に、データ従属性および／またはデータ従属性ハザードがあるか否かを判断することができる。

【００４５】

さらに、本明細書に述べられているように、レジスタステージ１３３における各コンシューマに関連するレジスタ識別子を、ステージ１３６、１３９、１５２における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的に望ましい。しかしながら、他の実施の形態において、いずれか１つのパイプライン１３２のいずれか１つのステージ１３３、１３６、１３９または１４２からのレジスタ識別子を、いずれか１つのパイプライン１３２のいずれか１つのステージ１３３、１３６、１３９または１４２からのレジスタ識別子と比較して、比較されたレジスタ識別子に関連する２つの命令の間にデータ従属性が存在するか否かを判断することができる。

【００４６】

実施の形態によっては、ある命令間では比較を行わないように、ＡＮＤロジック１６４、１６６および／または１６８あるいはデコーダ１５５、１５７、１５９および／または１６１を動作可能または動作不可能にすることが望ましい。例えば、比較ロジック１４４は、ステージ１３６、１３９および１４２における命令によって生成されるデータがレジスタステージ１３３における命令によって使用可能であるか否かを判断する追加の回路（図

示せず）を含んでもよい。ステージ１３６、１３９または１４２におけるプロデューサによって生成されるデータが、レジスタステージ１３３におけるコンシューマによって使用可能であるとき、プロデューサとコンシューマとの間にデータ従属性ハザードが無い場合がある。従って、プロデューサによって生成されるデータが使用可能である時に、プロデューサおよびコンシューマのレジスタ識別子を比較することが望ましくない場合がある。

【００４７】

かかる状況において、追加の回路は、プロデューサのレジスタ識別子を処理するデコーダ１５７、１５９または１６１に動作不能信号を送信するか、または、プロデューサのレジスタ識別子を処理するＡＮＤロジック１６４、１６６または１６８に動作不能信号を送信することができる。プロデューサのレジスタ識別子を処理するデコーダ１５７、１５９または１６１あるいはＡＮＤロジック１６４、１６６または１６８を動作不能にすることによって、コンシューマのレジスタ識別子とプロデューサのレジスタ識別子との比較が防止される。従って、図７におけるＡＮＤゲート１７２のうちのいずれかのアサートされた出力は、好ましい実施の形態について上述したようなデータ従属性を示すのみでなく、データ従属性ハザードをも示す。

【００４８】

動作不能信号に応じて、デコーダ１５７、１５９または１６１あるいはＡＮＤロジック１６４、１６６または１６８を動作不能にするには様々な方法があることに留意されたい。１つの実施の形態では、デコーダ１５７、１５９、１６１の各々は、動作不能にされると、その動作不能なデコーダ１５７、１５９または１６１によって送信されるｍビットレジスタ識別子のすべてのビットをクリアするよう構成されてもよい。その結果、動作不能なデコーダ１５７、１５９または１６１によって送信されるｍビットレジスタ識別子は、図７に示すＡＮＤゲート１７２のうちのいずれか１つにおいてアサートされた出力を生成しなくなる。

【００４９】

別の実施の形態では、各ＡＮＤロジック１６４、１６６、１６８は、それぞれ、図８におけるＯＲゲート１７５の出力について、各ＡＮＤロジック１６４、１６６、１６８が動作不能であるか否かを示す信号とＡＮＤをとるよう構成されてもよい。動作不能信号に応じて、各ＡＮＤロジック１６４、１６６、１６８はＯＲゲート１７５の出力とデアサートされた信号とのＡＮＤをとり、動作不能信号を受取っていない場合は、各ＡＮＤロジック１６４、１６６、１６８はＯＲゲート１７５の出力とアサートされた信号とのＡＮＤをとる。その結果、各ＡＮＤロジック１６４、１６６、１６８は、各ＡＮＤロジック１６４、１６６、１６８が動作不能信号を受取っていない場合にのみ、アサートされた出力を生成することができる。なお、他の実施の形態において、デコーダ１５７、１５９、１６１および／またはＡＮＤロジック１６４、１６６、１６８を動作不能にする他の方法および他の状況があってもよい。さらに、レジスタステージ１３３における命令のレジスタ識別子の比較を行わないことが望ましい状況では、上述した方法により、デコーダ１５５および／またはＡＮＤロジック１６４、１６６、１６８が動作不能にされてもよい。

【００５０】

本発明の原理を利用して、ライトアフタライト（ＷＡＷ）ハザードを検出することができることに留意するべきである。ＷＡＷハザードは、（１）前の命令および後の命令の両方が同じレジスタに書込む場合、および（２）前の命令が実際にレジスタに書込む前、に存在する。データ従属性ハザードの検出と同様、２つの書込み命令が同じレジスタ識別子を含むか否かを判断することにより、ＷＡＷハザードを検出することができる。従って、上述した回路を用いて、２つの書込み命令が同じレジスタを利用する場合を検出することができる。そして、追加の回路を用いて、前の命令からのデータがレジスタに書込まれているか否かを判断することができる。そして、上述した情報を使用して、ＷＡＷハザードが存在するか否かを判断することができる。

【００５１】

動作

処理システム１００の好ましい実施の形態の使用法および動作を以下に述べる。

【００５２】

例示の目的で、図６のステージ１３３、１３６、１３９、１４２の各々が命令を処理しており、レジスタステージ１３３における命令と他のステージ１３６、１３９、１４２における命令のいずれかとの間にデータ従属性ハザードが存在するか否かを判断することが望ましいと想定する。ラッチ１４５は、レジスタステージ１３３に入る命令をラッチしその命令を処理回路６６に供給するとき、その命令に含まれるｎビットレジスタ識別子をデコーダ１５５に供給する。デコーダ１５５は、好ましい実施の形態について上述した方法に従って、受取ったｎビットレジスタ識別子を復号化してｍビットレジスタ識別子にする。従って、ｍビットレジスタ識別子は、レジスタステージ１３３における命令が実行されるとき、その命令によって使用されるレジスタに対応する１つのアサートされたビットを有していなければならない。そして、デコーダ１５５は、ｍビットレジスタ識別子をＡＮＤロジック１６４、１６６、１６８に送信する。

【００５３】

ラッチ１４７は、実行ステージ１３６に入る命令をラッチし、その命令を処理回路６９に供給するとき、その命令に含まれるｎビットレジスタ識別子をデコーダ１５７に供給する。デコーダ１５７は、好ましい実施の形態について上述した方法に従って、受取ったｎビットレジスタ識別子を復号化してｍビットレジスタ識別子にする。従って、ｍビットレジスタ識別子は、実行ステージ１３６における命令が実行されるとき、その命令によって使用されるレジスタに対応する１つのアサートされたビットを有していなければならない。そして、デコーダ１５７は、ｍビットレジスタ識別子をＡＮＤロジック１６４に送信する。

【００５４】

ラッチ１４９は、例外検出ステージ１３９に入る命令をラッチし、その命令を処理回路７２に供給するとき、その命令に含まれるｎビットレジスタ識別子をデコーダ１５９に供給する。デコーダ１５９は、好ましい実施の形態について上述した方法に従って、受取ったｎビットレジスタ識別子を復号化してｍビットレジスタ識別子にする。従って、ｍビットレジスタ識別子は、例外検出ステージ１３９における命令が実行されるとき、その命令によって使用されるレジスタに対応する１つのアサートされたビットを有していなければならない。そして、デコーダ１５９は、ｍビットレジスタ識別子をＡＮＤロジック１６６に送信する。

【００５５】

ラッチ１５１は、書込みステージ１４２に入る命令をラッチし、その命令を処理回路７５に供給するとき、その命令に含まれるｎビットレジスタ識別子をデコーダ１６１に供給する。デコーダ１６１は、好ましい実施の形態について上述した方法に従って、受取ったｎビットレジスタ識別子を復号化してｍビットレジスタ識別子にする。従って、ｍビットレジスタ識別子は、書込みステージ１４２における命令が実行されるとき、その命令によって使用されるレジスタに対応する１つのアサートされたビットを有していなければならない。そして、デコーダ１６１は、ｍビットレジスタ識別子をＡＮＤロジック１６８に送信する。

【００５６】

そして、上述したｍビットレジスタ識別子は、ＡＮＤロジック１６４、１６６、１６８によって比較されて、レジスタステージ１３３における命令と他のステージ１３６、１３９、１４２における命令のいずれかとの間にデータ従属性が存在するか否かが判断される。この場合、ＡＮＤロジック１６４、１６６、１６８は、デコーダ１５５から送信されるｍビットレジスタ識別子におけるアサートされたビットと、デコーダ１５７、１５９、１６１から送信されるｍビットレジスタ識別子の少なくとも１つにおけるアサートされたビットとが同じレジスタに対応する場合に、かかるデータ従属性が存在することを示す。そして、上述した情報を用いて、レジスタステージ１３３における命令と後のステージ１３６、１３９および／または１４２における命令のいずれかとの間にデータハザードが存在す

るか否かを判断することができる。本発明の原理を使用することによって検出することができるデータハザードの例には、リードアフタライト（ＲＡＷ）ハザード、ライトアフタリード（ＷＡＲ）ハザードおよびライトアフタライト（ＷＡＷ）ハザードがある。

【００５７】

本発明の精神および原理から実質的に逸脱することなく、多くの変形および変更を本発明の上記実施の形態に対して行うことができる。

【００５８】

本発明は例として以下の実施形態を含む。

（１）　コンピュータプログラム（１０７）の命令を処理するシステム（１００）であって、

第１の命令と第２の命令とを同時に処理するよう構成され、該第１の命令を定義するデータに基づいて第１の符号化レジスタ識別子を送信するよう構成され、該データに基づいて第２の符号化レジスタ識別子を送信するようさらに構成されている少なくとも１つのパイプライン（１３２）と、

前記パイプライン（１３２）に結合され、前記第１の符号化レジスタ識別子を復号化して第１の復号化レジスタ識別子にすると共に、該第１の復号化レジスタ識別子を送信するよう構成された第１のデコーダ（１５５、１５７、１５９、１６１）と、

前記パイプライン（１３２）に結合され、前記第２の符号化レジスタ識別子を復号化して第２の復号化レジスタ識別子にすると共に、該第２の復号化レジスタ識別子を送信するよう構成されており、前記第１および第２の復号化レジスタ識別子が前記第１の命令を実行するために使用されるレジスタを識別する、第２のデコーダ（１５５、１５７、１５９、１６１）と、

前記第１および第２のデコーダ（１５５、１５７、１５９、１６１）とインタフェースされ、前記第１の復号化レジスタ識別子および前記第２の復号化レジスタ識別子を他の復号化レジスタ識別子とそれぞれ比較するよう構成された比較ロジック（１４４）と、

を含むシステム。

【００５９】

（２）　前記比較ロジック（１４４）は、前記第１および第２の復号化レジスタ識別子の一方と別の復号化レジスタ識別子との該比較ロジック（１４４）による比較に基づいて、データ従属性ハザードを検出するようにさらに構成されている、前記（１）に記載のシステム。

【００６０】

（３）　前記第１の復号化レジスタ識別子は複数のレジスタにそれぞれ対応する第１の複数のビットを含み、

前記レジスタのうちの１つに対応する前記第１の複数のビットのうちの１つはアサートされ、該第１の複数のビットの残りはデアサートされ、

前記第２の復号化レジスタ識別子は、前記複数のレジスタにそれぞれ対応する第２の複数のビットを含み、

前記レジスタのうちの前記１つに対応する前記第２の複数のビットのうちの１つはアサートされ、該第２の複数のビットの残りはデアサートされる、前記（１）に記載のシステム。

【００６１】

（４）　前記パイプライン（１３２）は、該パイプライン（１３２）の第１のステージ（１３３、１３６、１３９、１４２）における前記第１の命令と該パイプライン（１３２）の第２のステージ（１３３、１３６、１３９、１４２）における前記第２の命令とを同時に処理するよう構成され、前記第１のデコーダ（１５５、１５７、１５９、１６１）は該第１のステージ（１３３、１３６、１３９、１４２）に結合され、前記第２のデコーダ（１５５、１５７、１５９、１６１）は該第２のステージ（１３３、１３６、１３９、１４２）に結合されている、前記（１）に記載のシステム。

【００６２】

（５）　コンピュータプログラム（１０７）の命令を処理する処理システム（１００）であって、
第１のステージ（１３３、１３６、１３９、１４２）における第１の命令と第２のステージ（１３３、１３６、１３９、１４２）における第２の命令とを同時に処理するように構成されたパイプライン（１３２）と、
該第１のステージ（１３３、１３６、１３９、１４２）に結合され、前記第１の命令に関連する第１の符号化レジスタ識別子を受取るよう構成され、該第１の符号化レジスタ識別子を復号化して第１の復号化レジスタ識別子にし、該第１の復号化レジスタ識別子を送信するよう構成された第１のデコーダ（１５５、１５７、１５９、１６１）と、
前記第２のステージ（１３３、１３６、１３９、１４２）に結合され、前記第２の命令に関連する第２の符号化レジスタ識別子を受取るよう構成され、該第２の符号化レジスタ識別子を復号化して第２の復号化レジスタ識別子にし、該第２の復号化レジスタ識別子を送信するよう構成された第２のデコーダ（１５５、１５７、１５９、１６１）と、
前記第１および第２のデコーダ（１５５、１５７、１５９、１６１）とインタフェースされ、前記第１および第２の復号化レジスタ識別子を受取ると共に、該第１および第２の復号化レジスタ識別子を他のレジスタ識別子とそれぞれ比較するよう構成された比較ロジック（１４４）と、
を含む処理システム。

【００６３】

（６）　前記比較ロジック（１４４）は、前記第１および第２の復号化レジスタ識別子のうち一つと前記他の復号化レジスタ識別子のうち一つとの該比較ロジック（１４４）による比較に基づいて、データ従属性ハザードを検出するようさらに構成されている、前記（５）に記載のシステム。

【００６４】

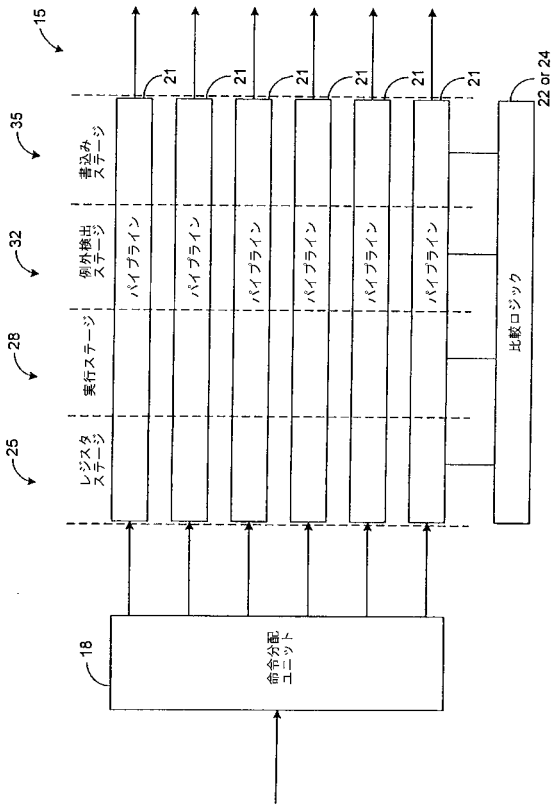
（７）　前記第１の復号化レジスタ識別子は、前記複数のレジスタにそれぞれ対応する第１の複数のビットを含み、
前記レジスタのうちの１つに対応する前記第１の複数のビットのうちの１つはアサートされ、前記第１の複数のビットの残りはデアサートされ、
前記第２の復号化レジスタ識別子は、前記複数のレジスタにそれぞれ対応する第２の複数のビットを含み、
前記レジスタのうちの１つに対応する前記第２の複数のビットのうちの１つはアサートされ、前記第２の複数のビットの残りはデアサートされる、
前記（５）に記載のシステム。

【００６５】

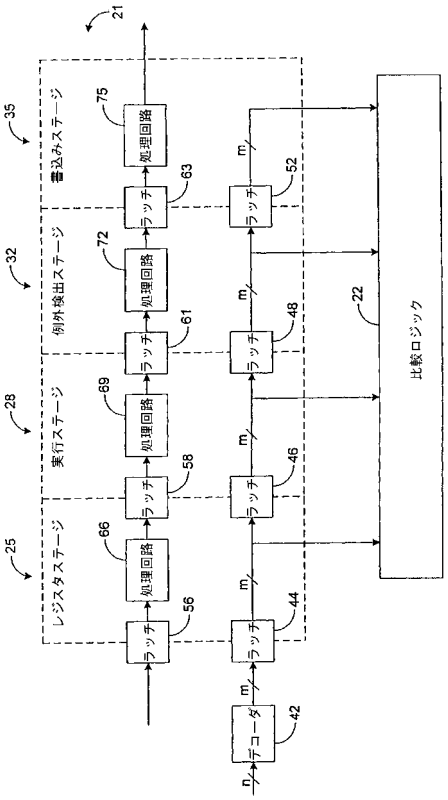
（８）　コンピュータプログラム（１０７）の命令を処理する方法であって、
命令を処理システム（１００）のパイプライン（１３２）に送信するステップと、
前記命令が前記パイプライン（１３２）の第１の部分によって処理されている間に、該命令に関連する符号化レジスタ識別子を復号化するステップと、
前記命令が前記パイプライン（１３２）の第２の部分によって処理されている間に、前記符号化レジスタ識別子を復号化するステップと、
前記命令が前記パイプライン（１３２）の前記第１の部分によって処理されている間に前記符号化レジスタ識別子を復号化する前記ステップに基づいて、第１の復号化レジスタ識別子を生成するステップと、
前記命令が前記パイプライン（１３２）の前記第２の部分によって処理されている間に前記符号化レジスタ識別子を復号化する前記ステップに基づいて、第２の復号化レジスタ識別子を生成するステップと、
前記第１および第２の復号化レジスタ識別子を他の復号化レジスタ識別子と比較するステップと、
該比較するステップに基づいてデータ従属性ハザードを検出するステップと、を含む方法。

【００６６】
（９）　前記命令を実行するステップをさらに含み、前記生成するステップの各々は、
複数のレジスタにそれぞれ対応する複数のビットを定義するステップと、
前記実行するステップにおいて使用されるレジスタに対応する前記ビットのうちの１つをアサートするステップと、
を含む、前記（８）に記載の方法。
【００６７】
【発明の効果】
本発明によれば、データ従属性、さらにはデータ従属性ハザードを検出するために必要となる配線を含む回路および空間の量を、従来の処理システムと比較して低減することができる。
【図面の簡単な説明】
【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のより詳細な図を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の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採用する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および比較ロジックのより詳細な図を示すブロック図である。
【図７】図６に示すＡＮＤロジックのより詳細な図を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の形態を示すブロック図である。
【符号の説明】
１００　　処理システム
１０７　　コンピュータプログラム
１３２　　パイプライン
１３３　　レジスタステージ
１３６　　実行ステージ
１３９　　例外検出ステージ
１４２　　書込みステージ
１４４　　比較ロジック
１５５、１５７、１５９、１６１　　デコーダ

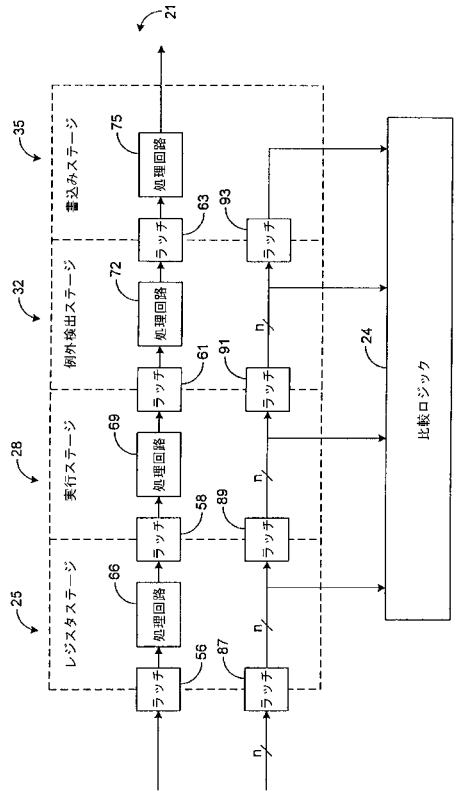
【図１】


15
35
書込みステージ
32
例外検出ステージ
28
実行ステージ
25
レジスタステージ
パイプライン
21
22 or 24
比較ロジック
18
命令分配ユニット

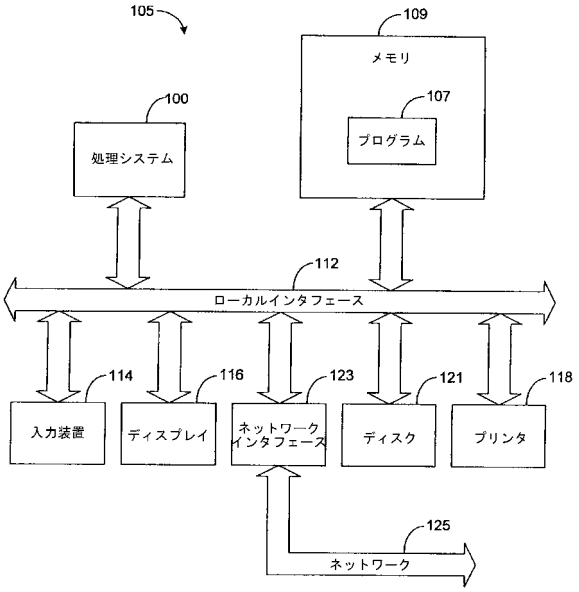
【図２】


21
35
書込みステージ
32
例外検出ステージ
28
実行ステージ
25
レジスタステージ
処理回路
ラッチ
デコーダ
比較ロジック
22
42
44
46
48
52
56
58
61
63
66
69
72
75

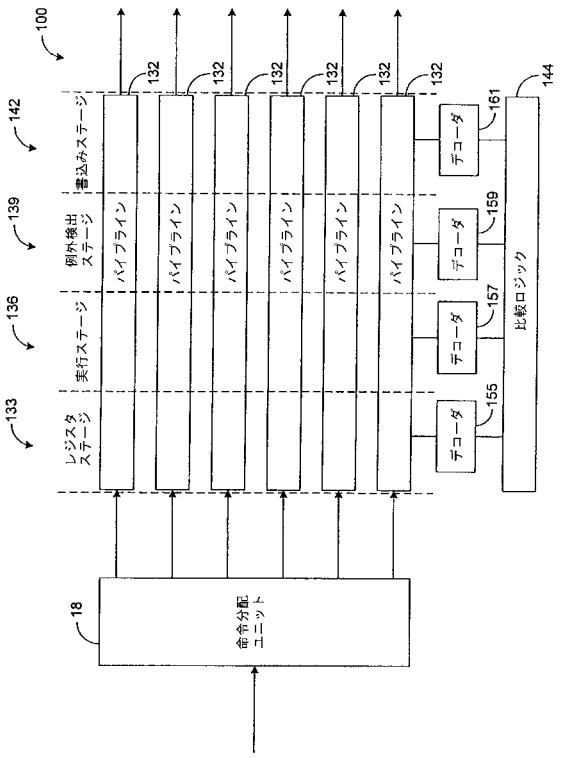
【図３】


21
35
書込みステージ
32
例外検出ステージ
28
実行ステージ
25
レジスタステージ
処理回路
ラッチ
比較ロジック
24
56
58
61
63
66
69
72
75
87
89
91
93

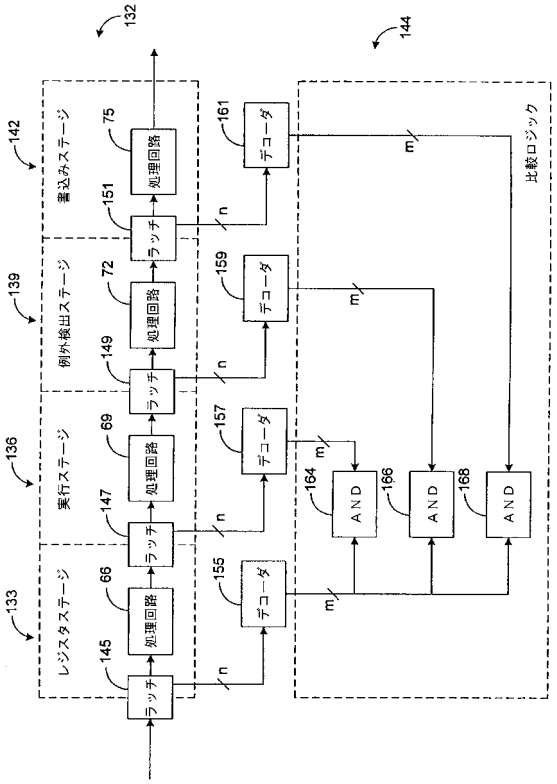
【図４】


105
109
メモリ
100
処理システム
107
プログラム
112
ローカルインタフェース
114
入力装置
116
ディスプレイ
123
ネットワークインタフェース
121
ディスク
118
プリンタ
125
ネットワーク

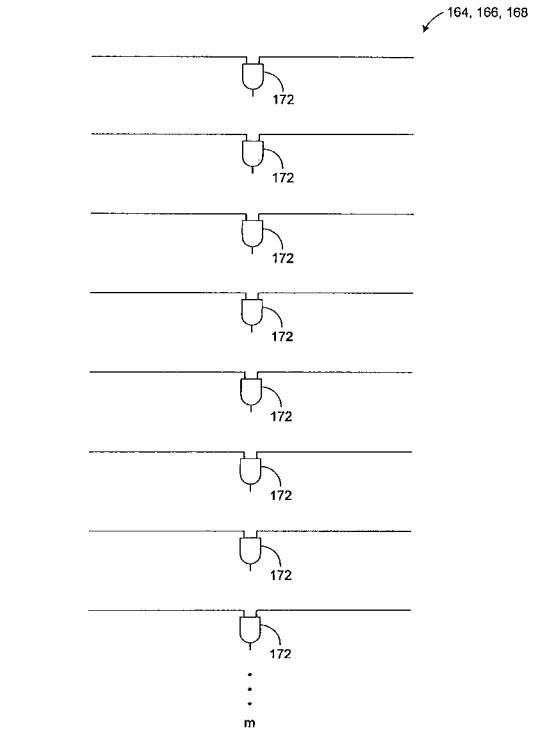
【図５】


100
142
書込みステージ
139
例外検出ステージ
136
実行ステージ
133
レジスタステージ
パイプライン
132
デコーダ
161
159
157
155
比較ロジック
144
命令分配ユニット
18

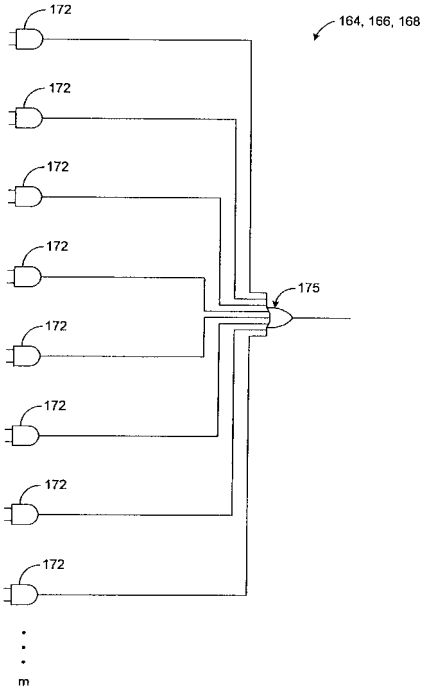
【図６】


132
144
142
書込みステージ
75
処理回路
151
ラッチ
139
例外検出ステージ
72
149
136
実行ステージ
69
147
133
レジスタステージ
66
145
デコーダ
161
159
157
155
n
m
AND
164
166
168
比較ロジック

【図７】


164, 166, 168
172
m

【図８】


172
164, 166, 168
175
m

フロントページの続き

(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８０５２６コロラド州フォート・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７－９３１５３（ＪＰ，Ａ）
特開平５－１０８３４８（ＪＰ，Ａ）
特開平４－３２８６３５（ＪＰ，Ａ）
特開平１－２３１１２６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 9/38