

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3906363号
(P3906363)

(45) 発行日 平成19年4月18日(2007.4.18)

(24) 登録日 平成19年1月26日(2007.1.26)

(51) Int. Cl.		F I		
G06F	9/38	(2006.01)	G06F	9/38 350B
G06F	9/34	(2006.01)	G06F	9/38 310X
			G06F	9/38 370A
			G06F	9/34 330

請求項の数 6 (全 19 頁)

<p>(21) 出願番号 特願2004-254408 (P2004-254408)</p> <p>(22) 出願日 平成16年9月1日(2004.9.1)</p> <p>(65) 公開番号 特開2006-72625 (P2006-72625A)</p> <p>(43) 公開日 平成18年3月16日(2006.3.16)</p> <p>審査請求日 平成18年8月9日(2006.8.9)</p> <p>特許法第30条第1項適用 平成16年3月5日 名古屋大学主催の「平成15年度 名古屋大学工学部電子情報学科卒業研究発表」において文書をもって発表</p> <p>特許法第30条第1項適用 2004年5月14日 社団法人情報処理学会発行の「情報処理学会研究報告 情報研報Vol. 2004 No. 48」に発表</p> <p>早期審査対象出願</p>	<p>(73) 特許権者 504139662 国立大学法人名古屋大学 愛知県名古屋市千種区不老町1番</p> <p>(74) 代理人 100068755 弁理士 恩田 博宣</p> <p>(74) 代理人 100105957 弁理士 恩田 誠</p> <p>(72) 発明者 安藤 秀樹 愛知県名古屋市千種区不老町1番 国立大学法人名古屋大学内</p> <p>(72) 発明者 嶋田 創 愛知県名古屋市千種区不老町1番 国立大学法人名古屋大学内</p>
--	--

最終頁に続く

(54) 【発明の名称】 クラスタ化スーパーパスカラプロセッサ及びクラスタ化スーパーパスカラプロセッサにおけるクラスタ間の通信制御方法

(57) 【特許請求の範囲】

【請求項1】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、

全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、

前記各クラスタ間の通信に用いられるバイパス経路と、

前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行う制御ユニットとを備え、

前記制御ユニットは、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させるものであって、

命令を前記命令ウィンドウに書き込む前に該命令ウィンドウ中の先行命令に依存関係がないかをチェックし、依存関係がある場合には前記先行命令が割り当てられたクラスタに選択的に広域通信を実施するように要求し、該クラスタは前記割り当てられた先行命令の実行後にその実行結果を選択的に広域通信するクラスタ化スーパーパスカラプロセッサ。

【請求項2】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、

全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、
前記各クラスタ間の通信に用いられるバイパス経路と、
前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行う制御ユニットとを備え、
前記制御ユニットは、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させるものであって、
命令が前記命令ウィンドウから発行されるときに、他の命令ウィンドウ中にその命令の実行結果を参照する後続命令がないかをチェックし、参照する後続命令がある場合には、その後続命令が割り当てられているクラスタに前記命令の実行結果を選択的に広域通信させるクラスタ化スーパースカラプロセッサ。

10

【請求項 3】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、
全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、
前記各クラスタ間の通信に用いられるバイパス経路と、
前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行う制御ユニットとを備え、
前記制御ユニットは、前記実行結果に対する他のクラスタからの過去の参照回数に基づいて該実行結果を広域通信させるクラスタ化スーパースカラプロセッサ。

20

【請求項 4】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、
前記各クラスタ間の通信に用いられるバイパス経路と
を備えたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法において

前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行うときに、
当該広域通信において、命令を前記命令ウィンドウに書き込む前に該命令ウィンドウ中の先行命令に依存関係がないかをチェックし、依存関係がある場合には前記先行命令が割り当てられたクラスタに選択的に広域通信を実施するように要求し、該クラスタは前記割り当てられた先行命令の実行後にその実行結果を選択的に広域通信することで、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させるようにしたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法。

30

【請求項 5】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、
前記各クラスタ間の通信に用いられるバイパス経路と
を備えたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法において

40

前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行うときに、
当該広域通信において、命令が前記命令ウィンドウから発行されるときに、他の命令ウィンドウ中にその命令の実行結果を参照する後続命令がないかをチェックし、参照する後続命令がある場合には、その後続命令が割り当てられているクラスタに前記命令の実行結

50

果を選択的に広域通信させることで、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させるようにしたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法。

【請求項 6】

命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、

前記各クラスタ間の通信に用いられるバイパス経路とを備えたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法において

10

前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行うときに、

当該広域通信において、前記実行結果に対する他のクラスタからの過去の参照回数に基づいて該実行結果を広域通信させるようにしたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、クラスタ化スーパースカラプロセッサ及びクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法に関する。

20

【背景技術】

【0002】

従来、プロセッサの性能向上を目的とした命令の並列実行とそれに伴う問題点を解決するために、例えば、非特許文献 1 に記載されたクラスタ化スーパースカラプロセッサが知られている。なお、このスーパースカラプロセッサのクラスタ化技術は、命令の実行を行う機能ユニットや命令を一時格納しておく命令ウィンドウをクラスタと呼ばれる塊に分割する技術である。各クラスタは、機能ユニットと、命令ウィンドウと、演算の実行結果を記憶するレジスタ・ファイルとを備えている。全てのクラスタに含まれるレジスタ・ファイルには、全てのクラスタに含まれる機能ユニットにおける実行結果が書き込まれる。即ち、各クラスタに含まれるレジスタ・ファイルは、同一の内容を保持する。図 1 2 に、クラスタ化スーパースカラプロセッサの構成を示す。

30

【0003】

プロセッサ 100 の命令処理に際しては、まず、メインメモリ (図示略) から命令キャッシュ 110 に命令が格納される。そして、この命令は、フェッチ・ユニット 120 によって読み出され、デコーダ 130 によってデコードされた後、ステアリング・ユニット 140 によって各命令におけるデータの依存関係が解析され、その依存関係に応じて命令が各クラスタ 150、160 の命令ウィンドウ 151、161 に割り当てられる。そして、該命令ウィンドウ 151、161 ではデータ依存関係を満たした命令が読み出され、命令の実行に使用されるデータはレジスタ・ファイル 152、162 から読み出され、機能ユニット 153、154、163、164 によって命令が実行される。そして、この命令の実行結果は、レジスタ・ファイル 152、162 に書き込まれる。命令の実行に際しては、必要に応じてメインメモリやデータ・キャッシュ 170 からデータが読み出される。

40

【0004】

また、機能ユニット 153、154、163、164 によって実行された実行結果を直後の命令で使用する場合には、実行結果はレジスタ・ファイル 152、162 に書き込まれる前に後続命令に転送される。この転送経路を、以下、バイパス経路と呼ぶ。なお、このバイパス経路は、クラスタ内のバイパス経路を形成するクラスタ内バイパス C I B とクラスタ間のバイパス経路を形成するクラスタ間バイパス C B B とから構成される。

【0005】

50

このように、スーパスカラプロセッサのクラスタ化を行うことで、クラスタ化していないスーパスカラプロセッサと比較して、各クラスタ内の機能ユニット数を減少させることができる。そして、機能ユニット数が減少することで、クラスタ内バイパス経路の配線長は短くなり、配線遅延は短縮される。

【0006】

しかし、クラスタ化したプロセッサにおいても各クラスタで必要なレジスタ・ファイルの大きさはクラスタ化していないプロセッサと実質的に同じであるため、レジスタ・ファイルの配線は短縮されず、配線遅延は短縮されない。

【0007】

このレジスタ・ファイルの遅延を削除する技術として、例えば、非特許文献2に記載された階層化レジスタ・ファイルがある。図13は、この階層化レジスタ・ファイルを組み込んだプロセッサの構成を示す。階層化レジスタ・ファイルは、レジスタ・キャッシュRC(上位レジスタ・ファイル)とメイン・レジスタ・ファイルMRF(下位レジスタ・ファイル)とから構成され、レジスタ・キャッシュRCはデータパス中に組み込まれている。レジスタ・キャッシュRCは容量がメイン・レジスタ・ファイルMRFに比べて小さく、高速なアクセスが可能である。メイン・レジスタ・ファイルMRFは機能ユニット251~254の全ての演算結果を保持し、レジスタ・キャッシュRCはメイン・レジスタ・ファイルMRFの一部の値を保持する。

10

【0008】

ここで、命令に必要とする値がレジスタ・キャッシュRCに存在すれば、機能ユニット251、252、253、254はメイン・レジスタ・ファイルMRFをアクセスする場合に比べて短いアクセス時間でレジスタ値を得ることができる。必要とする値がレジスタ・キャッシュRCに存在しなければ、レジスタ値は、メイン・レジスタ・ファイルMRFからレジスタ・キャッシュRCにレジスタ値を読み込んだ後でなければ得ることができず、長いアクセス時間を要する。

20

【0009】

なお、ここで前者のようにプロセッサから要求されたデータがレジスタ・キャッシュRCに存在する場合をヒットと呼び、レジスタ・キャッシュRCに存在しない場合をミスと呼ぶ。また、アクセスしたデータがレジスタ・キャッシュRCで見つかる割合をヒット率といい、アクセスしたデータがレジスタ・キャッシュRCで見つからない割合をミス率という。また、レジスタ・キャッシュRCのアクセスにかかる時間をヒット時間と呼ぶ。一方、メイン・レジスタ・ファイルMRFをアクセスするのにかかる参照時間をミス・ペナルティと呼ぶ。

30

【0010】

レジスタ・キャッシュRCは小さくて高速であるため、ヒット時間(例えば、1クロックサイクル)は階層化前のレジスタ・ファイルのアクセス時間よりも短くなる。

【非特許文献1】S. Palacharla, N. P. Jouppi, and J. E. Smith, "Complexity-Effective Superscalar Processors," Proceedings of 24th International Symposium on Computer Architecture, pp. 206-218, June 1997.

40

【非特許文献2】J. L. Cruz, A. Gonzalez, M. Valero, and N. P. Tophan, "Multiple-Banked Register File Architectures," Proceedings of 27th International Symposium on Computer Architecture, pp. 316-325, June 2000.

【発明の開示】

【発明が解決しようとする課題】

【0011】

ところで、レジスタ・ファイルのアクセス時間短縮のために、上述の階層化レジスタ・

50

ファイルとクラスタ化スーパーカプラプロセッサとを組み合わせることが考えられる。即ち、図12に示すプロセッサ100にメイン・レジスタ・ファイルMRFを備えるとともに、各クラスタ150, 160のレジスタ・ファイル152, 162をレジスタ・キャッシュRCに変更する。しかしながら、単に上述の階層化レジスタ・ファイルを当該クラスタ化スーパーカプラプロセッサに組み込むと、以下のような問題が生ずる。

【0012】

従来の方法では、命令の実行結果は、すべてのクラスタのレジスタ・キャッシュRCに書き込まれる。ここで、レジスタ・キャッシュRCはメイン・レジスタ・ファイルMRFの一部の値しか保持できないため、有効にレジスタ・キャッシュRCを利用するためには、この先に実行される命令が必要とするレジスタ値のみを保持することが望ましい。一般に、命令の実行結果は、ごく少数の演算にしか参照されないことが知られている。従って、クラスタ数分コピーされた実行結果の一部しか参照されないこととなり、他は無駄にレジスタ・キャッシュRCの記憶領域を消費することとなる。

10

【0013】

これにより、レジスタ・キャッシュRCに記憶され参照される可能性のあった有用なレジスタ値が消去される可能性が高まる。これは、レジスタ・キャッシュRCのミス率を上昇させ、ミス・ペナルティにより性能を低下させることとなっていた。

【0014】

本発明は、こうした実情に鑑みてなされたものであって、その目的は、レジスタ・キャッシュのミス率を低下させ、性能を向上させることができるクラスタ化スーパーカプラプロセッサ及びクラスタ化スーパーカプラプロセッサにおけるクラスタ間の通信制御方法を提供することにある。

20

【課題を解決するための手段】

【0015】

上記課題を解決するため、請求項1に記載の発明は、命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、前記各クラスタ間の通信に用いられるバイパス経路と、前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行う制御ユニットとを備え、前記制御ユニットは、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させるものであって、命令を前記命令ウィンドウに書き込む前に該命令ウィンドウ中の先行命令に依存関係がないかをチェックし、依存関係がある場合には前記先行命令が割り当てられたクラスタに選択的に広域通信を実施するように要求し、該クラスタは前記割り当てられた先行命令の実行後にその実行結果を選択的に広域通信させる。ここで、広域通信とは、あるクラスタで出力された実行結果を別のクラスタの上位レジスタ・ファイルにデータ通信することをいう。

30

【0016】

このような構成によれば、階層的（上位、下位）レジスタ・ファイルを備え、小さくて高速な上位レジスタ・ファイルを積極的に参照することで、レジスタ・ファイルのアクセス時間を短くすることができる。

40

【0017】

また、制御ユニットは、あるクラスタで生成される値が、別のクラスタで使用される場合を見つけて、前記あるクラスタで生成される値（実行結果）を他のクラスタへ通信させることで、クラスタに備えられた上位レジスタ・ファイル（レジスタ・キャッシュ）には有効な値が保持される確率が高くなる。このことで広域通信させることでクラスタ化スーパーカプラプロセッサにおけるレジスタ・キャッシュのミス率を低下させることにより、ミス・ペナルティを被る確率が低下し、性能が向上する。

【0018】

50

また、請求項 1 に記載の発明は、前記制御ユニットは、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させる。

【0019】

このような構成によれば、制御ユニットは、命令ウィンドウ中の命令のデータ依存を解析してクラスタ間の広域通信を判断する。このように、データ依存を解析して選択的に広域通信を行うため、命令ウィンドウ中の命令のデータ依存に関しては、命令の実行結果をその実行結果を必要とする他のクラスタに送信することができる。

【0020】

さらに、請求項 1 に記載の発明は、前記制御ユニットは、命令を前記命令ウィンドウに書き込む前に該命令ウィンドウ中の先行命令に依存関係がないかをチェックし、依存関係がある場合には前記先行命令が割り当てられたクラスタに選択的に広域通信を実施するように要求し、該クラスタは前記割り当てられた先行命令の実行後にその実行結果を選択的に広域通信する。

10

【0021】

このような構成によれば、制御ユニットは、今後、確実に結果を使用するクラスタの前記上位レジスタ・ファイルのみにデータ通信するため、不要な値が上位レジスタ・ファイルに格納されるといった事態を防ぎ、効率よく上位レジスタ・ファイルを使用することができる。

【0022】

請求項 2 に記載の発明は、前記制御ユニットは、命令が前記命令ウィンドウから発行されるときに、他の命令ウィンドウ中にその命令の実行結果を参照する後続命令がないかをチェックし、参照する後続命令がある場合には、その後続命令が割り当てられているクラスタに前記命令の実行結果を選択的に広域通信させることをその要旨とする。

20

【0023】

このような構成によっても、制御ユニットは、今後、確実に結果を使用するクラスタの前記上位レジスタ・ファイルのみにデータ通信するため、不要な値が上位レジスタ・ファイルに格納されるといった事態を防ぎ、効率よく上位レジスタ・ファイルを使用することができる。

【0024】

また、請求項 3 に記載の発明は、前記制御ユニットは、前記実行結果に対する他のクラスタからの過去の参照回数に基づいて該実行結果を広域通信させることをその要旨とする。

30

【0025】

このような構成によれば、制御ユニットは、過去の履歴に従って広域通信予測を行うため、時間的に（順序的に）離れた命令間のデータ依存に関してもクラスタ間の広域通信を行うことができる。

【0026】

また、請求項 4 に記載の発明は、命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、前記各クラスタ間の通信に用いられるバイパス経路とを備えたクラスタ化スーパースカラプロセッサにおけるクラスタ間の通信制御方法において、前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行うようにした。

40

【0027】

このような方法によれば、あるクラスタで生成される値が、別のクラスタで使用される場合を見つけて、前記あるクラスタで生成される値を他のクラスタへ広域通信させることで、クラスタに備えられた上位レジスタ・ファイル（レジスタ・キャッシュ）には有効な値が保持される確率が高くなる。このことでクラスタ化スーパースカラプロセッサにおける

50

レジスタ・キャッシュのミス率を低下させることにより、ミス・ペナルティを被る確率が低下し、性能が向上する。

【0028】

また、請求項4に記載の発明は、命令のデータ依存関係に基づいて前記実行結果を、当該実行結果を必要とするクラスタに対し選択的に広域通信させる。

このような方法によれば、命令ウィンドウ中の命令のデータ依存を解析してクラスタ間の広域通信を判断する。このように、データ依存を解析して選択的に広域通信を行うため、命令ウィンドウ中の命令のデータ依存に関しては、命令の実行結果をその実行結果を必要とする他のクラスタに送信することができる。

【0029】

また、請求項4に記載の発明は、命令を前記命令ウィンドウに書き込む前に該命令ウィンドウ中の先行命令に依存関係がないかをチェックし、依存関係がある場合には前記先行命令が割り当てられたクラスタに選択的に広域通信を実施するように要求し、該クラスタは前記割り当てられた先行命令の実行後にその実行結果を選択的に広域通信する。

【0030】

このような方法によれば、前記広域通信を好適に実現することができる。また、こうした通信制御方法を採用することで、今後、確実に結果を使用するクラスタの前記上位レジスタ・ファイルのみにデータ通信するため、不要な値が上位レジスタ・ファイルに格納されるといった事態を防ぎ、効率よく上位レジスタ・ファイルを使用することができる。

【0031】

また、請求項5に記載の発明は、命令が前記命令ウィンドウから発行されるときに、他の命令ウィンドウ中にその命令の実行結果を参照する後続命令がないかをチェックし、参照する後続命令がある場合には、その後続命令が割り当てられているクラスタに前記命令の実行結果を選択的に広域通信させることをその要旨とする。

【0032】

このような方法によっても、前記広域通信を好適に実現することができる。また、こうした通信制御方法を採用することで、今後、確実に結果を使用するクラスタの前記上位レジスタ・ファイルのみにデータ通信するため、不要な値が上位レジスタ・ファイルに格納されるといった事態を防ぎ、効率よく上位レジスタ・ファイルを使用することができる。

【0033】

また、請求項6に記載の発明は、前記実行結果に対する他のクラスタからの過去の参照回数に基づいて該実行結果を広域通信させることをその要旨とする。

このような方法によっても、前記広域通信を好適に実現することができる。また、こうした通信制御方法を採用することで、過去の履歴に従って広域通信予測を行うため、時間的に（順序的に）離れた命令間のデータ依存に関してもクラスタ間の広域通信を行うことができる。

【発明の効果】

【0034】

本発明によれば、クラスタ化スーパースカラプロセッサにおけるレジスタ・キャッシュのミス率を低下させることにより、ミス・ペナルティを被る確率が低下し、性能が向上する。

【発明を実施するための最良の形態】

【0035】

（第1の実施形態）

以下、本発明を具体化した第1の実施形態を図面に従って説明する。

はじめに、図1を参照して、このスーパースカラプロセッサの構成について説明する。図1は、本発明にかかるクラスタ化スーパースカラプロセッサの一例を示すブロック図である。

【0036】

同図1に示すように、階層化レジスタ・ファイルを組み込んだクラスタ化スーパースカラ

10

20

30

40

50

プロセッサ 1 は、非特許文献 1 (図 1 2) では各クラスタに設けられていたレジスタ・ファイルの代わりに、各クラスタにレジスタ・キャッシュ R C を設ける。そして、メイン・レジスタ・ファイル M R F は全体に 1 つだけ置き、すべてのレジスタ値を保持する。そして、命令結果が出力されたときは、必ずメイン・レジスタ・ファイル M R F に書き込むようにする。

【 0 0 3 7 】

なお、レジスタ・キャッシュ R C には後述するように、様々な書き込みポリシーのもとに書き込むことになる。このレジスタ・キャッシュ R C への書き込みは、結果を出した命令の属するクラスタのレジスタ・キャッシュ R C への書き込みと、それとは異なるクラスタのレジスタ・キャッシュ R C への書き込みに分けることができる。

10

【 0 0 3 8 】

図 1 に示すように、プロセッサ 1 は階層化レジスタ・ファイルを組み込んだクラスタ化スーパースカラプロセッサであり、命令キャッシュ 1 0、フェッチ・ユニット 2 0、デコーダ 3 0、ステアリング・ユニット 4 0、複数 (図 1 では 2 つ) のクラスタ 5 0、6 0、メイン・レジスタ・ファイル M R F、データ・キャッシュ 7 0 を備えている。クラスタ 5 0 は命令ウィンドウ 5 1 と、レジスタ・キャッシュ R C と、機能ユニット 5 2、5 3 を備えている。同様に、クラスタ 6 0 は命令ウィンドウ 6 1 と、レジスタ・キャッシュ R C と、機能ユニット 6 2、6 3 を備えている。

【 0 0 3 9 】

そして、プロセッサ 1 の命令処理に際しては、該プロセッサ 1 のメインメモリ (図示略) へのアクセスに基づいて、命令がメインメモリから命令キャッシュ 1 0 に書き込まれる。そして、この命令は、フェッチ・ユニット 2 0 によって読み出され (命令フェッチ)、デコーダ 3 0 によってデコード (復号化) される。そしてこのデコードされた命令は、ステアリング・ユニット 4 0 によって、命令間の依存関係 (データ依存関係 : ある実行結果を出力する先行命令と、その実行結果を用いる後続命令の関係) が解析され、その依存関係に応じてこの命令が各クラスタ 5 0、6 0 の命令ウィンドウ 5 1、6 1 に割り当てられる。

20

【 0 0 4 0 】

そして、該命令ウィンドウ 5 1、6 1 ではデータ依存関係を満たした命令が読み出され、この命令の実行に使用されるデータはレジスタ・キャッシュ R C から読み出され、機能ユニット 5 2、5 3、6 2、6 3 によって命令が実行される。そして、この実行結果は、レジスタ・キャッシュ R C とメイン・レジスタ・ファイル M R F に書き込まれることとなる。

30

【 0 0 4 1 】

命令の実行に際しては、必要に応じてデータ・キャッシュ 7 0 からデータが読み出され、演算が実行された後、その演算結果は、レジスタ・キャッシュ R C とメイン・レジスタ・ファイル M R F に書き込まれることとなる。また、機能ユニット 5 2、5 3、6 2、6 3 によって演算された実行結果を直後の演算で使用する場合には、実行結果はレジスタ・キャッシュ R C に書き込まれる前にバイパス経路によって、後続命令に転送される。このバイパス経路は、クラスタ内のバイパス経路を形成するクラスタ内バイパス C I B とクラスタ間のバイパス経路を形成するクラスタ間バイパス C B B とから構成される。

40

【 0 0 4 2 】

ステアリング・ユニット 4 0 は命令の依存関係を考慮して命令を各クラスタ 5 0、6 0 の命令ウィンドウ 5 1、6 1 に分配して登録する。この命令の依存関係を考慮した分配方法の例をあげる。

【 0 0 4 3 】

まず、ステアリング・ユニット 4 0 にデスティネーション物理レジスタ番号 (d s t) でインデクスされた表を持つ。この表の内容は、その d s t に実行結果を出力する命令が登録されたクラスタ番号である。ステアリング・ユニット 4 0 に送られた命令は、まず、命令の第 1 ソース・オペランド (s r c 1) で表を引き、表より s r c 1 の値が出力され

50

るクラスタの番号を得る。もし、そのクラスタの命令ウィンドウにその命令を登録できるならば、そのクラスタに登録を行う。そのクラスタへの登録が行えない場合は、第2ソース・オペランド (s r c 2) を用いて同様のことを行う。 s r c 2 を用いた場合でも登録するクラスタが決まらない場合は、命令ウィンドウの空いているクラスタに登録する。最後に、登録された命令のデスティネーション・オペランドに対応する表のエントリ (登録項目) に、その命令が登録されたクラスタ番号を書き込む。

【 0 0 4 4 】

なお、レジスタ・キャッシュ R C の容量は小さいため、有効な値を保持しなければレジスタ・キャッシュミスが増加し、性能低下を引き起こす。従って、どの値をレジスタ・キャッシュ R C に書き込むかが重要になる。

10

【 0 0 4 5 】

なお、ここで前記上位レジスタ・ファイルの最も単純な書き込みポリシーは、すべての結果をすべての上位レジスタ・ファイルに書き込むものである。ただし、この方法は不必要な値も書き込むために、必要な値を上位レジスタ・ファイルから失う確率が高くなり、レジスタ・キャッシュミスの増加によって性能が低下してしまう。そこで、結果の中から、あるクラスタで生成される値が、別のクラスタで使用される場合を見つけ、その値のみを広域通信する。

【 0 0 4 6 】

ここで、本実施形態の選択的広域通信では、命令ウィンドウ 5 1、6 1 は、現在の状況をみて、命令の結果が出る前に、その結果に依存している命令が別のクラスタに存在するかをチェックし、存在する場合は、そのクラスタのレジスタ・キャッシュ R C に結果を広域通信する。この方法は、今後、確実に結果を使用するクラスタのレジスタ・キャッシュ R C のみに広域通信するため、不要な値でレジスタ・キャッシュ R C が汚染される (クラスタ内の機能ユニットに使用されないデータ、つまり不要なデータがレジスタ・キャッシュ R C に書き込まれる) ことを防ぎ、効率よくレジスタ・キャッシュ R C を使用することができる。以後、この方法を「 o n - t h e - f l y (その場で) 」と呼ぶこととする。本実施形態では、「 o n - t h e - f l y 」を実現するために、「 r e q u e s t & w r i t e (要求と書き込み) 」という方法で選択的広域通信を行った。

20

【 0 0 4 7 】

次に、図 2、図 3 を併せ参照して、「 r e q u e s t & w r i t e 」について説明する。なお、図 2 (a) は、命令の登録方法を示すハードウェア構造図、図 2 (b) は、広域通信要求の登録方法を示すハードウェア構造図であり、図 3 は、「 r e q u e s t & w r i t e 」の処理手順を説明するフローチャートである。

30

【 0 0 4 8 】

はじめに、「 r e q u e s t & w r i t e 」の概要について説明する。この「 r e q u e s t & w r i t e 」では、制御ユニットは、命令を命令ウィンドウ 5 1、6 1 に書き込む前に、該命令ウィンドウ 5 1、6 1 中の先行命令に依存関係がないかを確認する (後述するステップ S 1 0 2)。そして、該命令が登録されるクラスタとは異なるクラスタに存在する先行命令に依存関係がある場合には、命令ウィンドウ中の先行命令を特定し (後述するステップ S 1 0 4)、その命令に対し、自身が割り当てられるクラスタに広域通信するように要求を出す (後述するステップ S 1 0 5)。この要求された命令の実行結果は、命令の実行終了後に、指示されたクラスタのレジスタ・キャッシュ R C に広域通信される (後述するステップ S 1 0 6)。

40

【 0 0 4 9 】

「 r e q u e s t & w r i t e 」では、「 o n - t h e - f l y 」を実現するため、まず、図 2 (a) の左図に示すように、デスティネーション物理レジスタ番号を索引とするクラスタ番号表を準備する。クラスタ番号表の各要素は、結果が出力されるクラスタ番号からなる。また、各クラスタの命令ウィンドウの各エントリ (登録項目) に広域通信先を示すフラグを追加する。これは、クラスタ番号表に対応するビットベクタ (n 個のビット列からなるデータ構造) である。さらに、広域通信先フラグの更新のため、命令ウィンド

50

ウ中のデスティネーション物理レジスタ番号のエントリをCAM（連想メモリ）にする。ここで、連想メモリとは、メモリに記憶されているデータの位置をアドレス（番地）で示すのではなく、内容で示す記憶方法である。この連想メモリは、入力されるデータと一致する値を持つエントリを探し出す。以下、これを連想検索と呼ぶ。

【0050】

ここで、クラスタ番号表と広域通信先フラグとを用いて「on-the-fly」により選択的広域通信する手順を図2、図3に示す。

まず、図2(a)に示すように、命令を命令ウィンドウに登録するとき、その命令のデスティネーション物理レジスタ番号(dst)に対応するクラスタ番号表のエントリに命令が登録されるクラスタ番号を書き、命令ウィンドウの広域通信先フラグをすべて0とする（前処理：ステップS101）。

10

【0051】

同時に、図2(b)に示すように、ソース物理レジスタ番号(src)でクラスタ番号表を参照し、読み出されたクラスタ番号（ソース側）と当該命令が登録されるクラスタ番号（デスティネーション側）とが等しいかどうかを比較器Cによって、比較する（ステップS102）。

【0052】

ここで、クラスタ番号が同じであれば（ステップS102が「YES」）、命令ウィンドウは、依存関係にある先行命令が同じクラスタの命令ウィンドウ中に存在すると判断して、広域通信は行わない（ステップS103）。

20

【0053】

一方、もしクラスタ番号が一致しなければ（ステップS102が「NO」）、命令ウィンドウ中の先行命令を探すために、ソース物理レジスタ番号で命令ウィンドウを連想検索する（ステップS104）。そして、命令ウィンドウは、依存関係にある先行命令が他のクラスタの命令ウィンドウ中に存在すると判断して、先行命令の広域通信フラグのうち、当該命令のクラスタに対応するフラグ(Flag)を立てる（ステップS105）。そして、命令ウィンドウから命令が発行される時に命令ウィンドウ中の広域通信先フラグを参照し、フラグの立っているクラスタへ結果を広域通信する（ステップS106）。

【0054】

以上説明した第1の実施形態によれば、以下のような効果を得ることができる。

30

(1) クラスタで生成される値が、別のクラスタで使用される場合を見つけて、あるクラスタで生成される値を他のクラスタへ選択的に広域通信するようにした。このように広域通信することでクラスタに備えられたレジスタ・キャッシュRCには有効な値が保持される確率が高くなる。このことでクラスタ化スーパースカラプロセッサにおけるレジスタ・キャッシュRCのミス率を低下させることにより、ミス・ペナルティを被る確率が低下し、性能が向上する。

【0055】

(2) 命令ウィンドウ51、61中の命令のデータ依存を解析して広域通信を判断した。このように、データ依存を解析して選択的に広域通信を行うため、命令ウィンドウ51、61中の命令のデータ依存に関しては、命令の実行結果を、その実行結果を必要とする他のクラスタに送信することができる。

40

【0056】

(3) 今後、確実に結果を使用するクラスタのレジスタ・キャッシュRCのみに広域通信するため、不要な値によってレジスタ・キャッシュRCが汚染されるといった事態を防ぎ、効率よくレジスタ・キャッシュRCを使用することができる。

【0057】

なお、本発明の実施形態は、以下のように変更してもよい。

・上記実施形態では、クラスタ数は2つであるとしたが、このクラスタ数は3つ以上であってもよい。このようにクラスタ数が複数であっても、依存関係のある先行命令が他のクラスタの命令ウィンドウ中にあると判断したときは、そのクラスタに対して選択的に広

50

域通信を行うことで、レジスタ・キャッシュRCのミス率を減らし、性能を向上させることができる。

【0058】

・上記実施の形態では、プロセッサの構成を例示したが、上述した構成に限定されるものではない。要は、

(a) 命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタ。

(b) 全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイル。

(c) 前記各クラスタ間の通信に用いられるバイパス経路。

(d) 前記バイパス経路を介して前記機能ユニットの実行結果を出力したクラスタから別のクラスタにデータ通信させる広域通信を行う制御ユニット。

を備えていればよい。

【0059】

(第2の実施形態)

次に、本発明を具体化した第2の実施形態を図面に従って説明する。

第2の実施形態では、上記第1の実施形態で説明した「on-the-fly」の別の実現方法「check&write(チェックと書き込み)」について説明する。

【0060】

ここで、図4、図5を参照して、「check&write」について第1の実施形態を参照しながら説明する。なお、図4は、「check&write」の動作を説明するハードウェア構造図である。図5は、「check&write」の処理手順を説明するフローチャートである。

【0061】

はじめに、「check&write」の概要について説明する。この「check&write」では、制御ユニットは、命令が命令ウィンドウ51、61から発行されるときに、他の命令ウィンドウ51、61中にその命令の実行結果を参照する命令がないかをチェックする(後述するステップS201)。そして、参照する命令がある場合には(後述するステップS202が「YES」)、その命令が割り当てられているクラスタのレジスタ・キャッシュRCに実行結果を広域通信させる。

【0062】

この「check&write」では、「on-the-fly」を実現するために、図4に示すように、各クラスタにレジスタ・キャッシュRC(図1)の書き込み許可表を設ける。これは、各物理レジスタに対して、そのクラスタのレジスタ・キャッシュRCへの書き込み許可を示す表である。なお、表のインデックスは物理レジスタ番号、エントリ内容は書き込み許可を示す1ビットのフラグである。このレジスタ・キャッシュRCの書き込み許可表を用いて、「on-the-fly」により選択的広域通信する手順を説明する。

【0063】

まず、命令ウィンドウには、他のクラスタから命令の実行結果が送られてくる1サイクル以上前に、その結果が使用可能となることを示すタグが送られている。このタグは実行結果が格納される物理レジスタ番号がよく用いられる。タグは命令ウィンドウに送られ、命令ウィンドウ内の例えば上述の第1ソース・オペランドsrc1および第2ソース・オペランドsrc2と比較される(ステップS201)。もし、その結果を使用する命令があるならば、タグは一致し(ステップS202が「YES」)、対応するreadyビットをセット(ステップS203)することになる。逆にいえば、命令ウィンドウ内でタグの一致がなければ、その命令ウィンドウ内には結果を必要とする命令が存在しないことになる。よって、命令ウィンドウ内でタグの一致があった場合はそのクラスタのレジスタ・キャッシュRCに結果を書き込み(ステップS204)、一致がなければ書き込まない(ステップS205)ようにすれば、「on-the-fly」を実現できる。

10

20

30

40

50

【0064】

レジスタ・キャッシュRC書き込み許可表のタグに対応するエントリには、命令ウィンドウのエントリのタグの全比較結果（例えば、`src1__match`、`src2__match`）のORを取った結果を書き込む。

【0065】

そして、他のクラスタから広域通信された結果は、このレジスタ・キャッシュRCの書き込み許可表のフラグ（`Flag`）を参照し、そのクラスタのレジスタ・キャッシュRCへの書き込みを決定する。

【0066】

以上説明した第2の実施形態によっても、先の第1の実施形態による上記（1）～（3）の効果と同様の効果もしくはそれに準じた効果を得ることができる。

（第3の実施形態）

次に、本発明を具体化した第3の実施形態を図面に従って説明する。

【0067】

第3の実施形態では、上記第1の実施形態および第2の実施形態で説明した「`on-the-fly`」とは異なる方法を用いてクラスタ間を選択的に広域通信する方法（「広域通信予測」）について説明する。

【0068】

広域通信予測とは、過去にある命令の実行結果が、他のクラスタから参照された回数を数え、その回数から、次に同一の命令が実行されるときに広域通信するかどうかを判断する方法である。

【0069】

ここで、図面を参照して、広域通信予測について説明する。なお、図6（a）は広域通信判定表を示すハードウェア構造図、図6（b）は広域通信要求表を示すハードウェア構造図である。

【0070】

まず、図6（a）に示すように、どの命令に広域通信するか示す広域通信判定表を用意する。この表は、命令アドレスの下位ビットを索引とし、命令識別のためのタグ、広域通信の可否を判断するための2ビット飽和カウンタ（`2bc`）、前回の命令の命令ウィンドウ登録後に`2bc`の更新があったかどうかを示す更新フラグ（`Flag`）を持つ。命令識別のためのタグは命令アドレスのうち、索引として用いなかった上位ビットを使用する。また、図6（b）に示すように、ある物理レジスタに対する広域通信要求を広域通信判定表に反映するために、広域通信要求表を用意する。この表は、物理レジスタ番号を索引とし、命令アドレス、クラスタ番号（`cls`）を登録する。

【0071】

以下、広域通信の判断方法および広域通信判定表と広域通信要求表の更新手順を説明する。

まず、図7～図9を参照して命令の命令ウィンドウ登録時における、広域通信の判断方法および広域通信判定表と広域通信要求表の更新手順について説明する。なお、図7は、命令の命令ウィンドウ登録時における処理手順を説明するハードウェア構造図、図8は、2ビット飽和カウンタ（`2bc`）の更新手順を説明するハードウェア構造図、図9は、命令の命令ウィンドウ登録時における、処理手順を説明するフローチャートである。ちなみに、この処理は命令が命令ウィンドウに入力される度に行われる。ここで、2ビット飽和カウンタ（`2bc`）の初期値を2とし、フラグ（`Flag`）の初期値を0とする。

【0072】

まず、命令アドレスの下位ビットで広域通信判定表を引き、命令アドレスの上位ビットとタグとを比較する（ステップS301）。図7に示すように、この比較に際しては、例えば比較器Cが用いられる。そして、命令アドレスとタグとが一致するか否かを判断する（ステップS302）。命令アドレスとタグとが一致すれば（ステップS302が「YES」）、次に前回の命令登録後に広域通信があったか否かを判断する（ステップS303

)。ここで、前回の命令登録後に広域通信があれば (F l a g が 1 : ステップ S 3 0 3 が「 Y E S 」)、広域通信判定表の 2 ビット飽和カウンタ (2 b c) の値は変化させない (ステップ S 3 0 4)。

【 0 0 7 3 】

一方、前回の命令登録後に広域通信がなければ (F l a g が 0 : ステップ S 3 0 3 が「 N O 」)、広域通信判定表の 2 b c をデクリメント (減少) させる。すなわち、図 8 に示すように、例えば減算器 S を用いて 2 b c の値を 1 減少させる (例えば、2 b c の値が初期値の 2 であれば、2 b c の値は 1 となる)。そして、この減算器 S によって減算した値と、減算する前の値とをマルチプレクサ (データセレクタ) m u x に入力する。そして、このマルチプレクサ m u x には、例えば F l a g の値に基づく制御信号が送られ、上記入力された値の中から 1 つを選択する。このような構成とすることで、2 b c の更新が可能となる。

10

【 0 0 7 4 】

次に、この 2 b c の値を用いて広域通信予測を行う (ステップ S 3 0 6)。この広域通信予測は、この 2 b c の値と、2 の値とをマルチプレクサ m u x に入力し、例えば比較器 C の比較結果に基づく制御信号がマルチプレクサ m u x に送られ、上記入力された値の中から 1 つを選択して予測値としている。そして、広域通信予測が行われた後、F l a g を初期値の 0 に戻して (ステップ S 3 0 7)、図 7 に示すように、広域通信要求表のデスティネーション物理レジスタ番号に対応するエントリに、命令アドレスと命令が登録されるクラスタ番号 c l s を書く (ステップ S 3 0 8)。

20

【 0 0 7 5 】

他方、命令アドレスとタグとが一致しなければ (ステップ S 3 0 2 が「 N O 」)、次に、広域通信判定表にタグ、2 b c の初期値 (広域通信すると弱く予測)、F l a g の初期値 0 を書き込む (ステップ S 3 0 9)。そして、2 b c の初期値 2 に準じ、広域通信を行うものとする (ステップ S 3 1 0)。広域通信を行った後は、上述のステップ S 3 0 8 の処理を行う。

【 0 0 7 6 】

次に、図 1 0、図 1 1 を参照して後続命令による広域通信要求の登録について説明する。なお、図 1 0 は広域通信要求の登録処理を説明するハードウェア構造図、図 1 1 は後続命令による広域通信要求の登録の処理手順を説明するフローチャートである。ちなみに、この処理は命令が命令ウィンドウに入力される度に行われる。

30

【 0 0 7 7 】

まず、ソース物理レジスタ番号で広域通信要求表を引き、表から得られた命令アドレスの下位ビットで広域通信判定表を引き、命令アドレスの上位ビットとを比較する (ステップ S 4 0 1)。図 1 0 に示すように、この比較に際しては、例えば比較器 C 1 が用いられる。そして、命令アドレスの上位ビットとタグとが一致するか否かが判断される (ステップ S 4 0 2)。ここで、命令アドレスの上位ビットとタグとが一致すれば (ステップ S 4 0 2 が「 Y E S 」)、次に、広域通信要求表から得られたクラスタ番号 (c l s) と命令が登録されるクラスタ番号とを比較する (ステップ S 4 0 3)。同じくこの比較に際しては、例えば比較器 C 2 が用いられる。そして、広域通信要求表から得られたクラスタ番号 (c l s) と命令が登録されるクラスタ番号とが異なるか否かが判断される (ステップ S 4 0 4)。ここで、広域通信要求表から得られたクラスタ番号 (c l s) と命令が登録されるクラスタ番号とが異なる (ステップ S 4 0 4 が「 Y E S 」) ことと、上記命令アドレスの上位ビットとタグとが一致する (ステップ S 4 0 2 が「 Y E S 」) こととの AND 条件に基づいて、広域通信判定表の 2 b c をインクリメント (増加) させる。すなわち、図 1 0 に示すように、例えば加算器 A を用いて 2 b c の値を 1 増加させる。また、F l a g を 1 にする (ステップ S 4 0 6)。

40

【 0 0 7 8 】

一方、上記 AND 条件が成立する以外場合 (ステップ S 4 0 2、ステップ S 4 0 4 のうち少なくとも一方が「 N O 」)、すなわち、データ依存元が表に登録されていないか、あ

50

るいはデータ依存があっても結果が同一クラスタで生成されるような場合には、広域通信判定表の2bcの更新を行わない(ステップS407)。

【0079】

このように、過去に他のクラスタから参照された情報を広域通信判定表のFlagとして記憶し(ステップS406)、このFlagを基に、広域通信の可否を判断するための2ビット飽和カウンタ(2bc)を制御する(ステップS304またはステップS305)。そして、この2bcの値を用いて広域通信を行う。この広域通信は、2bcが0あるいは1であれば、例えば広域通信しない。2bcが例えば2あるいは3であれば、広域通信する等々の情報を予め決めておくことで、実際の広域通信を実行することが可能となる。

10

【0080】

以上説明した第3の実施形態によっても、先の第1の実施形態による上記(1)の効果と同様の効果もしくはそれに準じた効果を得ることができるとともに、新たに次のような効果を得ることもできる。

【0081】

(4)過去の履歴に従って広域通信予測を行うこととしたため、時間的に(順序的に)離れた命令間のデータ依存に関してもクラスタ間の広域通信を行うことができる。

次に、以上の実施形態から把握することのできる請求項以外の技術的思想を記載する。

【0082】

(イ)命令が格納される命令ウィンドウと、該命令ウィンドウから読み出された命令に従ってレジスタ値を出力するとともに命令の実行結果が格納される上位レジスタ・ファイルと、命令を実行する機能ユニットとを備える複数のクラスタと、全ての前記機能ユニットの実行結果が格納される下位レジスタ・ファイルと、前記各クラスタ間の通信に用いられるバイパス経路とを備えた通信制御方式において、前記実行結果が、別のクラスタで使用される場合を見つけるステップと、前記バイパス経路を介して前記あるクラスタで生成される値を他のクラスタへ広域通信させるステップと、を実行することを特徴とする通信制御方式。

20

【0083】

(ロ)命令の実行結果が出る前に、その結果に依存している命令が別のクラスタに存在するかをチェックするステップと、存在する場合には、そのクラスタに実行結果を選択的に広域通信させるステップと、を実行する(イ)に記載の通信制御方式。

30

【0084】

(ハ)命令を前記命令ウィンドウに書き込む前に、該命令ウィンドウ中の先行命令に依存関係がないかチェックするステップと、依存関係がある場合には、その命令に対し、自身が割り当てられるクラスタにデータ通信するように要求を出すステップと、該要求された命令の実行結果は、命令の実行終了後に、指示されたクラスタの前記上位レジスタ・ファイルに選択的に広域通信されるステップと、を実行する(ロ)に記載の通信制御方式。

【0085】

(ニ)命令が前記命令ウィンドウから発行されるときに、他の命令ウィンドウ中にその命令の実行結果を参照する命令がないかチェックするステップと、参照する命令がある場合には、その命令が割り当てられているクラスタの前記上位レジスタ・ファイルに実行結果を選択的に広域通信させるステップと、を実行する(ロ)に記載の通信制御方式。

40

【0086】

(ホ)ある命令が実行結果のデータ通信が必要だったか否かという履歴をとるステップと、その履歴からその命令の実行時にデータ通信が必要か否かを決定するステップと、必要と判断した場合のみ、全てのクラスタの前記上位レジスタ・ファイルに広域通信させるステップと、を実行する(イ)に記載の通信制御方式。

【図面の簡単な説明】

【0087】

50

【図 1】本発明にかかるクラスタ化スーパースカラプロセッサの一例を示すブロック図。

【図 2】第 1 の実施形態の選択的広域通信を説明する、(a) は命令の登録方法を示すハードウェア構造図、(b) は広域通信要求の登録方法を示すハードウェア構造図。

【図 3】同第 1 の実施形態の選択的広域通信について、その処理手順を説明するフローチャート。

【図 4】第 2 の実施形態の選択的広域通信を説明するハードウェア構造図。

【図 5】同第 2 の実施形態の選択的広域通信について、その処理手順を説明するフローチャート。

【図 6】第 3 の実施形態の選択的広域通信について、(a) は広域通信判定表を示すハードウェア構造図、(b) は広域通信要求表を示すハードウェア構造図。

10

【図 7】同第 3 の実施形態について、命令の命令ウィンドウ登録時における処理手順を説明するハードウェア構造図。

【図 8】同第 3 の実施形態について、2 ビット飽和カウンタ (2 b c) の更更新手順を説明するハードウェア構造図。

【図 9】同第 3 の実施形態について、命令の命令ウィンドウ登録時における、処理手順を説明するフローチャート。

【図 10】同第 3 の実施形態について、広域通信要求の登録処理を説明するハードウェア構造図。

【図 11】同第 3 の実施形態について、後続命令による広域通信要求の登録の処理手順を説明するフローチャート。

20

【図 12】従来のクラスタ化スーパースカラプロセッサの一例を示すブロック図。

【図 13】従来の階層化レジスタ・ファイルを組み込んだプロセッサの一例を示すブロック図。

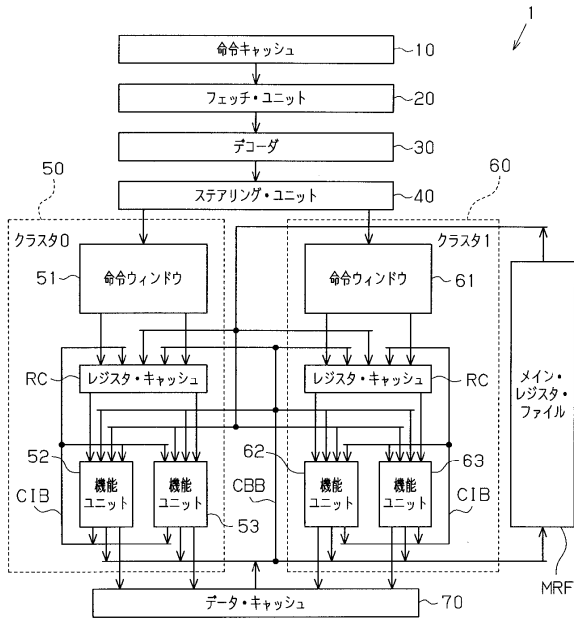
【符号の説明】

【 0 0 8 8 】

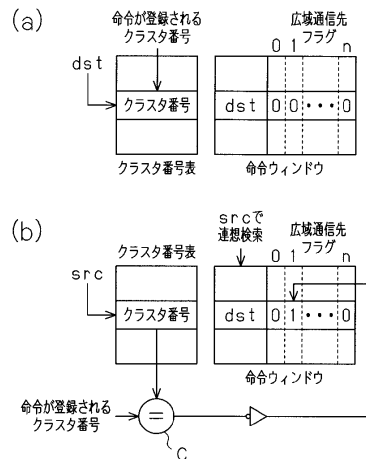
1 ... プロセッサ、1 0 ... 命令キャッシュ、2 0 ... フェッチ・ユニット、3 0 ... デコーダ、4 0 ... ステアリング・ユニット、5 0、6 0 ... クラスタ、5 1、6 1 ... 命令ウィンドウ、5 2、5 3、6 2、6 3 ... 機能ユニット、7 0 ... データ・キャッシュ、A ... 加算器、C ... 比較器、C I B ... クラスタ内バイパス、C B B ... クラスタ間バイパス、M R F ... メイン・レジスタ・ファイル (下位レジスタ・ファイル)、m u x ... マルチプレクサ (データセレクタ)、R C ... レジスタ・キャッシュ (上位レジスタ・ファイル)、S ... 減算器。

30

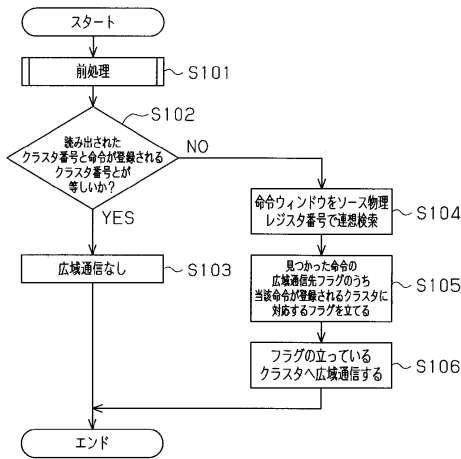
【 図 1 】



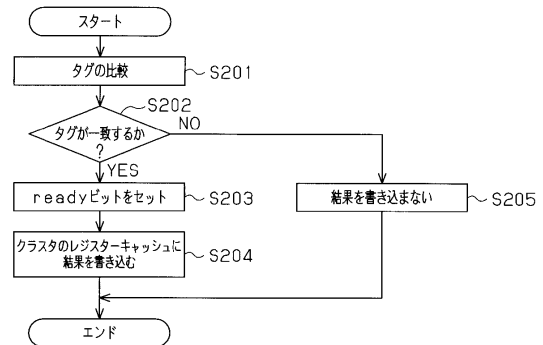
【 図 2 】



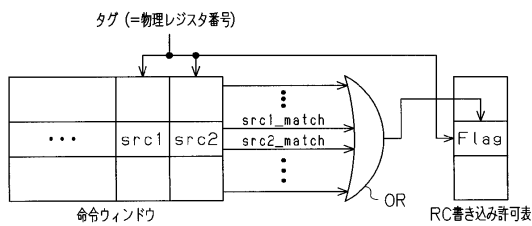
【 図 3 】



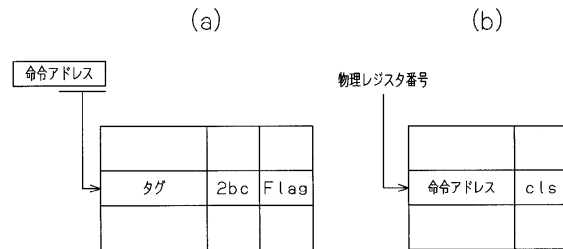
【 図 5 】



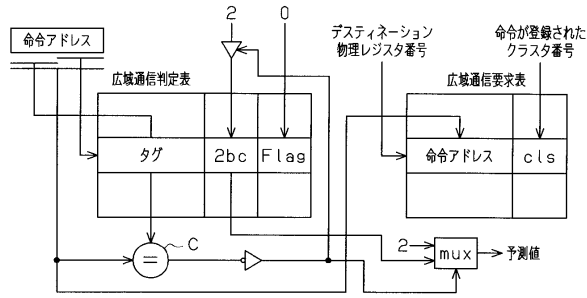
【 図 4 】



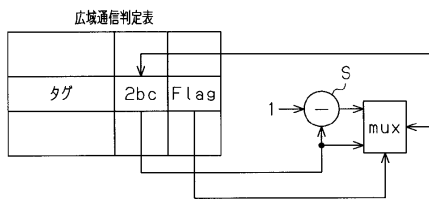
【 図 6 】



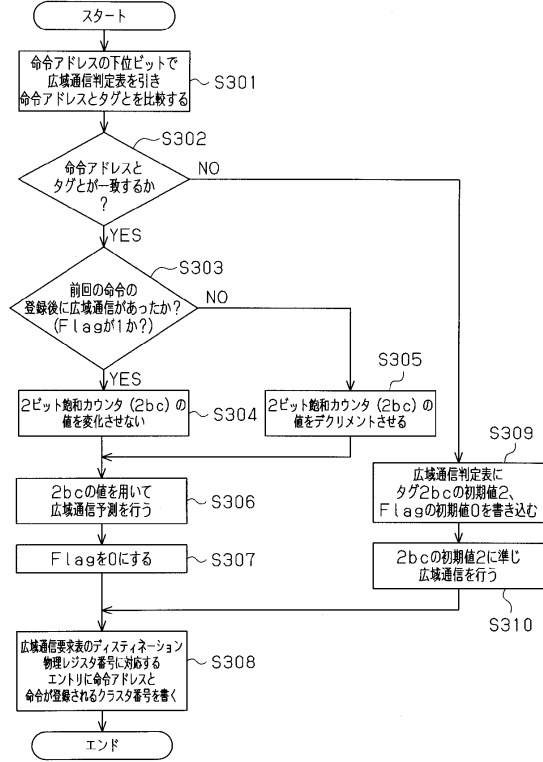
【図7】



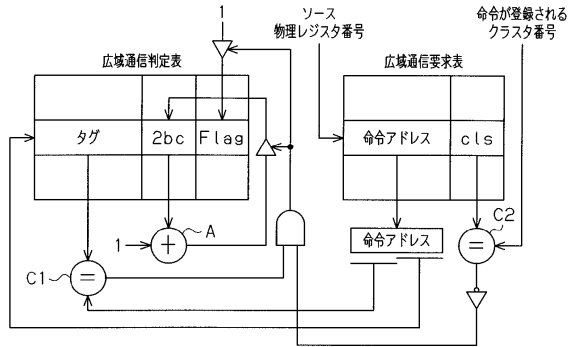
【図8】



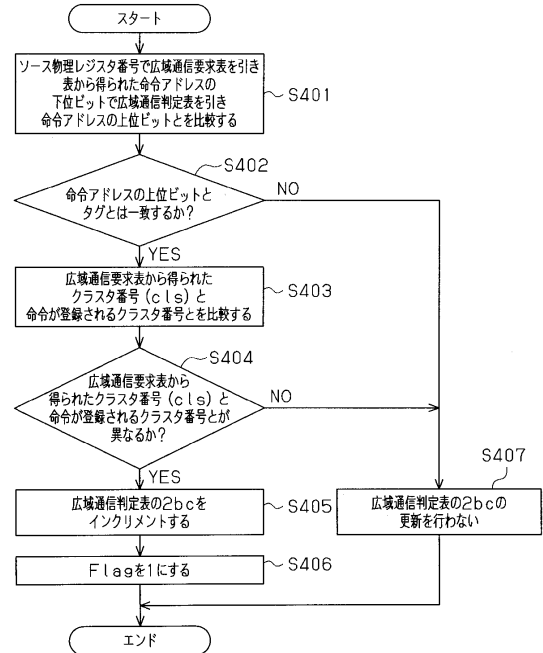
【図9】



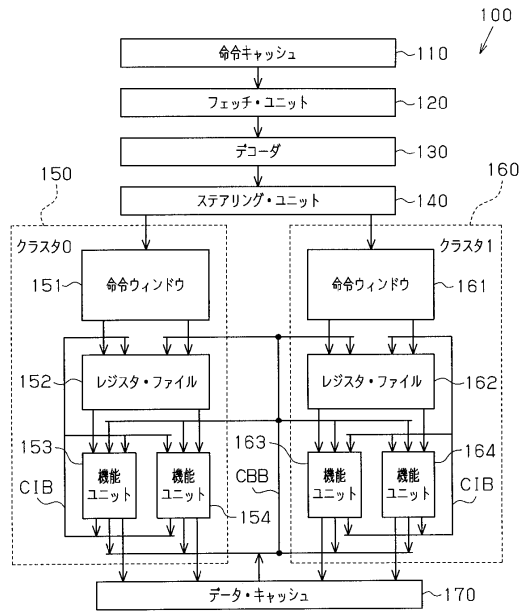
【図10】



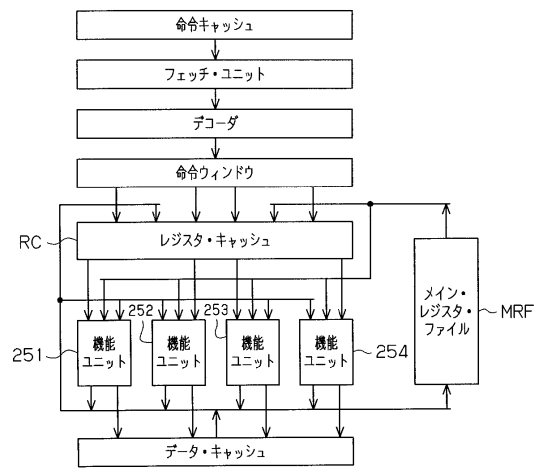
【図11】



【 図 1 2 】



【 図 1 3 】



フロントページの続き

(72)発明者 望月 厚志

愛知県名古屋市千種区不老町1番 国立大学法人名古屋大学内

審査官 石川 正二

(56)参考文献 特開平08-212070(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06F 9/34