

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6225554号
(P6225554)

(45) 発行日 平成29年11月8日(2017.11.8)

(24) 登録日 平成29年10月20日(2017.10.20)

(51) Int.Cl.

F I

G 0 6 F 9/38 (2006.01)

G 0 6 F 9/38 3 1 0 E

請求項の数 8 (全 18 頁)

| | | | |
|-----------|------------------------------|-----------|--------------------------------|
| (21) 出願番号 | 特願2013-168694 (P2013-168694) | (73) 特許権者 | 000005223 |
| (22) 出願日 | 平成25年8月14日(2013.8.14) | | 富士通株式会社 |
| (65) 公開番号 | 特開2015-36922 (P2015-36922A) | | 神奈川県川崎市中原区上小田中4丁目1番1号 |
| (43) 公開日 | 平成27年2月23日(2015.2.23) | (74) 代理人 | 100090273 |
| 審査請求日 | 平成28年5月10日(2016.5.10) | | 弁理士 國分 孝悦 |
| | | (72) 発明者 | 伊東 利郎 |
| | | | 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 |
| | | (72) 発明者 | 秋月 康伸 |
| | | | 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内 |
| | | 審査官 | 清木 泰 |

最終頁に続く

(54) 【発明の名称】 演算処理装置及び演算処理装置の制御方法

(57) 【特許請求の範囲】

【請求項 1】

複数のステージングラッチを有し、前記複数のステージングラッチのうち、最終段のステージングラッチを含む第1の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするパイプライン動作と、前記複数のステージングラッチのうち、前記第1の複数のステージングラッチよりも前段側に位置する第2の複数のステージングラッチ間におけるデータの遷移に複数のクロックを必要とするマルチサイクル動作とにより第1の命令を実行する第1の命令実行部と、

第2の命令を実行する第2の命令実行部と、

前記第1の命令と前記第2の命令とを入力し、前記第1の命令の実行と前記第2の命令の実行とが一部重複するように、前記第1の命令を前記第1の命令実行部に発行するとともに、前記第2の命令を前記第2の命令実行部に発行する命令制御部とを有し、

前記命令制御部は、前記第1の命令実行部が前記第1の命令の実行中に、前記第1の命令の不定の終了サイクルを確定し、前記確定した終了サイクルに応じて、前記第2の命令の前記第2の命令実行部への発行の抑止を解除することを特徴とする演算処理装置。

【請求項 2】

前記第2の命令実行部は、

複数の第2のステージングラッチを有し、前記複数の第2のステージングラッチのうち、最初段のステージングラッチを含む第3の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするパイプライン動作と、前記複数の第2のステー

10

20

ジングラッチのうち、前記第 3 の複数のステージングラッチよりも後段側に位置する第 4 の複数のステージングラッチ間におけるデータの遷移に複数のクロックを必要とするマルチサイクル動作とにより、前記第 2 の命令を実行することを特徴とする請求項 1 記載の演算処理装置。

【請求項 3】

前記第 2 の命令実行部は、複数の第 2 のステージングラッチを有し、非共有マルチサイクル動作により、前記第 2 の命令を実行するものであり、

前記非共有マルチサイクル動作は、前記複数の第 2 のステージングラッチ間におけるデータの遷移に複数のクロックを必要とするものであり、

前記ステージングラッチ間にそれぞれ位置する回路の資源に関して、前記第 2 の命令実行部における前記非共有マルチサイクル動作と、前記演算処理装置が備える他の命令実行部は、回路の資源を共有していないことを特徴とする請求項 1 記載の演算処理装置。

10

【請求項 4】

前記第 2 の命令実行部は、複数の第 2 のステージングラッチを有し、非共有パイプライン動作と共有パイプライン動作により、前記第 2 の命令を実行するものであり、

前記非共有パイプライン動作は、前記複数の第 2 のステージングラッチのうち、最初段のステージングラッチを含む第 3 の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするものであり、

前記共有パイプライン動作は、前記複数の第 2 のステージングラッチのうち、前記第 3 の複数のステージングラッチよりも後段側に位置する第 4 の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするものであり、

20

前記ステージングラッチ間にそれぞれ位置する回路の資源に関して、前記第 2 の命令実行部における前記非共有パイプライン動作と、前記演算処理装置が備える他の命令実行部は、回路の資源を共有していないものであり、

前記ステージングラッチ間にそれぞれ位置する回路の資源に関して、前記第 2 の命令実行部における前記共有パイプライン動作と、前記演算処理装置が備える他の命令実行部は、回路の資源を共有しているものであることを特徴とする請求項 1 記載の演算処理装置。

【請求項 5】

前記命令制御部は、

前記第 1 の命令実行部が前記第 1 の命令の実行中において、前記第 1 の命令のマルチサイクル動作の実行と、前記第 2 の命令のマルチサイクル動作、非共有マルチサイクル動作又は共有パイプライン動作の実行とが重複しないように、前記第 2 の命令の前記第 2 の命令実行部への発行を抑止することを特徴とする請求項 2 ~ 4 のいずれか 1 項に記載の演算処理装置。

30

【請求項 6】

前記命令制御部は、

前記第 1 の命令の実行における前記パイプライン動作と、前記第 2 の命令の実行とが一部重複するように、前記第 1 の命令を前記第 1 の命令実行部に発行するとともに、前記第 2 の命令を前記第 2 の命令実行部に発行することを特徴とする請求項 1 ~ 5 のいずれか 1 項に記載の演算処理装置。

40

【請求項 7】

前記命令制御部は、

前記第 1 の命令の実行における前記パイプライン動作又は前記マルチサイクル動作と、前記第 2 の命令の実行とが一部重複するように、前記第 1 の命令を前記第 1 の命令実行部に発行するとともに、前記第 2 の命令を前記第 2 の命令実行部に発行することを特徴とする請求項 1 ~ 6 のいずれか 1 項に記載の演算処理装置。

【請求項 8】

複数のステージングラッチを有し、前記複数のステージングラッチのうち、最終段のステージングラッチを含む第 1 の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするパイプライン動作と、前記複数のステージングラッチのうち

50

、前記第 1 の複数のステージングラッチよりも前段側に位置する第 2 の複数のステージングラッチ間におけるデータの遷移に複数のクロックを必要とするマルチサイクル動作とにより第 1 の命令を実行する第 1 の命令実行部と、第 2 の命令を実行する第 2 の命令実行部とを有する演算処理装置の制御方法において、

前記演算処理装置が有する命令制御部が、前記第 1 の命令と前記第 2 の命令とを入力し

、
前記命令制御部が、前記第 1 の命令の実行と前記第 2 の命令の実行とが一部重複するように、前記第 1 の命令を前記第 1 の命令実行部に発行するとともに、前記第 2 の命令を前記第 2 の命令実行部に発行し、

前記命令制御部が、前記第 1 の命令実行部が前記第 1 の命令の実行中に、前記第 1 の命令の不定の終了サイクルを確定し、前記確定した終了サイクルに応じて、前記第 2 の命令の前記第 2 の命令実行部への発行の抑止を解除することを特徴とする演算処理装置の制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、演算処理装置及び演算処理装置の制御方法に関する。

【背景技術】

【0002】

互いに依存関係にある 2 以上の命令を発行する命令発行制御部と、実行パイプラインとを有する情報処理装置が知られている（例えば、特許文献 1 参照）。命令発行制御部は、命令デコード部と、命令が使用するリソースの使用状態を管理するリソース管理部とを有する。発行タイミング決定&リソース割り当て部は、リソースの使用状態に基づきデコード済み命令が使用するリソースが現在から何サイクル後に使用可能になるのかを判断し、デコード済み命令の発行タイミングとして決定し、リソースの使用状態を更新しリソースの割り当てを行う。発行確定命令待機バッファは、発行タイミングの決定及びリソースの割り当てが行われた命令を、その発行タイミングとなるまでの間バッファリングして保持し、その発行タイミングで実行パイプラインに発行する。

【0003】

また、マルチスレッドプロセッサの、複数のスレッドにより共用使用されるパイプラインのディスパッチ時点において、1つのスレッドをブロックする方法が知られている（例えば、特許文献 2 参照）。1つのスレッドの命令に対する長い待ち時間の条件は、パイプラインを共用使用するスレッドのすべてを停止させることができる。ディスパッチブロック信号命令は、ディスパッチ時に長い待ち時間の条件を含むスレッドをブロックする。ブロックの長さは、待ち時間の長さとも一致するため、長い待ち時間の条件が解除された後に、パイプラインは、ブロックされたスレッドから命令をディスパッチできる。ディスパッチ時に 1つのスレッドをブロックすることにより、プロセッサは、ブロックの間、他のスレッドから命令をディスパッチできる。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】特開 2012 - 173755 号公報

【特許文献 2】特開 2006 - 351008 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

2 個の命令をオーバーラップさせて発行すれば、スループットを向上させることができる。しかし、オーバーラップ可能な命令とオーバーラップ困難な命令とがある。オーバーラップ困難な命令であっても、その一部をオーバーラップできれば、スループットを向上させることができる。

10

20

30

40

50

【 0 0 0 6 】

1つの側面では、本発明の目的は、2個の命令を発行する際にスループットを向上させることができる演算処理装置及び演算処理装置の制御方法を提供することである。

【課題を解決するための手段】

【 0 0 0 7 】

演算処理装置は、複数のステージングラッチを有し、前記複数のステージングラッチのうち、最終段のステージングラッチを含む第1の複数のステージングラッチ間におけるデータの遷移に単一のクロックのみを必要とするパイプライン動作と、前記複数のステージングラッチのうち、前記第1の複数のステージングラッチよりも前段側に位置する第2の複数のステージングラッチ間におけるデータの遷移に複数のクロックを必要とするマルチ
10 サイクル動作とにより第1の命令を実行する第1の命令実行部と、第2の命令を実行する第2の命令実行部と、前記第1の命令と前記第2の命令とを入力し、前記第1の命令の実行と前記第2の命令の実行とが一部重複するように、前記第1の命令を前記第1の命令実行部に発行するとともに、前記第2の命令を前記第2の命令実行部に発行する命令制御部とを有し、前記命令制御部は、前記第1の命令実行部が前記第1の命令の実行中に、前記第1の命令の不定の終了サイクルを確定し、前記確定した終了サイクルに応じて、前記第2の命令の前記第2の命令実行部への発行の抑止を解除する。

【発明の効果】

【 0 0 0 8 】

一部重複するように第1の命令及び第2の命令を発行することにより、スループットを
20 向上させることができる。

【図面の簡単な説明】

【 0 0 0 9 】

【図1】図1は、演算処理装置としてのプロセッサを含む情報処理システムの構成例を示す図である。

【図2】図2は、プロセッサの構成例を示す図である。

【図3】図3は、図2に示した命令発行制御部の構成例を示す図である。

【図4】図4(A)及び(B)は、図3の取出し可能命令検出部の一部の構成例を示す図である。

【図5】図5は、演算器のパイプライン動作を示す図である。
30

【図6】図6は、演算器のマルチサイクル動作を示す図である。

【図7】図7は、スループット1のパイプライン動作を示す図である。

【図8】図8は、命令発行制御部の命令発行例を示す図である。

【図9】図9は、2個の複合マルチサイクル動作の命令発行を示す図である。

【図10】図10は、複合マルチサイクル動作及び共有完全パイプライン動作の命令発行を示す図である。

【図11】図11は、2個の複合マルチサイクル動作の命令発行を示す図である。

【図12】図12は、複合マルチサイクル動作及び共有完全パイプライン動作の命令発行を示す図である。

【図13】図13は、発行抑止信号を用いて動作を部分的にオーバーラップさせる方法を
40 示す図である。

【図14】図14は、演算命令のサイクルステージを説明するための図である。

【図15】図15は、先行命令が複合マルチサイクル動作であり、後続命令が複合マルチサイクル動作である場合のタイミングチャートである。

【図16】図16は、先行命令が複合マルチサイクル動作であり、後続命令が純粋マルチサイクル動作である場合のタイミングチャートである。

【図17】図17は、先行命令が複合マルチサイクル動作であり、後続命令が共有完全パイプライン動作である場合のタイミングチャートである。

【発明を実施するための形態】

【 0 0 1 0 】

10

20

30

40

50

図 1 は、演算処理装置としてのプロセッサを含む情報処理システムの構成例を示す図である。図 1 に示す情報処理システムは、例えば複数のプロセッサ 1 1 A、1 1 B 及びメモリ 1 2 A、1 2 B と、外部装置との入出力制御を行うインターコネクト制御部 1 3 とを有する。

【 0 0 1 1 】

図 2 は、プロセッサ 1 1 の構成例を示す図である。プロセッサ 1 1 は、演算処理装置であり、図 1 のプロセッサ 1 1 A 及び 1 1 B に対応し、例えば命令のアウトオブオーダー実行やパイプライン処理の機能を有する。

【 0 0 1 2 】

命令フェッチステージでは、命令フェッチ部 2 1、命令バッファ 2 4、分岐予測回路 2 2、一次命令キャッシュメモリ 2 3、及び二次キャッシュメモリ 3 4 等が動作する。命令フェッチ部 2 1 は、分岐予測回路 2 2 からフェッチする命令の予測分岐先アドレス、分岐制御部 3 0 から分岐演算により確定した分岐先アドレス等を受け取る。命令フェッチ部 2 1 は、受け取った予測分岐先アドレス、分岐先アドレス、及び命令フェッチ部 2 1 内で作成した分岐しない場合にフェッチする命令の連続した次のアドレス等から、1つのアドレスを選択して次の命令フェッチアドレスを確定する。命令フェッチ部 2 1 は、確定した命令フェッチアドレスを一次命令キャッシュメモリ 2 3 に出力し、出力された確定後の命令フェッチアドレスに対応する命令コードをフェッチする。

【 0 0 1 3 】

一次命令キャッシュメモリ 2 3 は、二次キャッシュメモリ 3 4 の一部のデータを格納しているものであり、二次キャッシュメモリ 3 4 は、メモリコントローラ 3 5 を介してアクセス可能なメモリの一部のデータを格納しているものである。一次命令キャッシュメモリ 2 3 に該当するアドレスのデータが存在しない場合には二次キャッシュメモリ 3 4 からデータをフェッチし、二次キャッシュメモリ 3 4 に該当するデータが存在しない場合にはメモリからデータをフェッチする。本実施形態では、メモリはプロセッサ 1 1 の外部に配置しているため、外部にあるメモリとの入出力制御はメモリコントローラ 3 5 を介して行われる。一次命令キャッシュメモリ 2 3 や二次キャッシュメモリ 3 4、メモリの該当するアドレスからフェッチされた命令コードは、命令バッファ 2 4 に格納される。

【 0 0 1 4 】

分岐予測回路 2 2 は、命令フェッチ部 2 1 から出力された命令フェッチアドレスを受け取り、命令フェッチと並行して分岐予測を実行する。分岐予測回路 2 2 は、受け取った命令フェッチアドレスを基に分岐予測を行い、分岐の成立又は不成立を示す分岐方向と予測分岐先アドレスとを命令フェッチ部 2 1 へ返す。命令フェッチ部 2 1 は、予測された分岐方向が成立であった場合には次の命令フェッチアドレスとして予測された分岐先アドレスを選択する。

【 0 0 1 5 】

命令発行ステージでは、命令デコーダ 2 5 及び命令発行制御部 2 6 が動作する。命令デコーダ 2 5 は、命令バッファ 2 4 から命令コードを受け取って命令の種別や必要な実行資源等を解析し、解析結果を命令発行制御部 2 6 に出力する。命令発行制御部 2 6 は、リザベーションステーションの構造を持つ。命令発行制御部 2 6 は、命令で参照するレジスタ等の依存関係を見て、依存関係のあるレジスタの更新状況や同じ実行資源を用いる命令の実行状況等から実行資源が命令を実行可能かどうかを判断する。命令発行制御部 2 6 は、実行資源が命令を実行可能であると判断した場合には、レジスタ番号やオペランドアドレス等の命令の実行に必要な情報を実行資源に対して出力する。また、命令発行制御部 2 6 は、実行可能な状態になるまで命令を格納しておくバッファの機能も有する。演算器制御回路 2 7 は、命令発行制御部 2 6 から入力した情報に応じて演算器 2 8 を制御する。

【 0 0 1 6 】

命令実行ステージでは、演算器 2 8、一次オペランドキャッシュメモリ 2 9、及び分岐制御部 3 0 等の実行資源が動作する。演算器 2 8 は、レジスタ 3 1 や一次オペランドキャッシュメモリ 2 9 からデータを受け取り、四則演算、論理演算、三角関数演算、及びアド

10

20

30

40

50

レス計算等の命令に対応した演算を実行し、演算結果をレジスタ 3 1 や一次オペランドキャッシュメモリ 2 9 に出力する。一次オペランドキャッシュメモリ 2 9 は、命令キャッシュメモリ 2 3 と同様に、二次キャッシュメモリ 3 4 の一部のデータを格納しているものである。一次オペランドキャッシュメモリ 2 9 は、ロード命令によるメモリから演算器 2 8 やレジスタ 3 1 へのデータのロードや、ストア命令による演算器 2 8 やレジスタ 3 1 からメモリへのデータのストア等に用いられる。各実行資源は、命令実行の完了通知を命令完了制御部 3 2 へ出力する。

【 0 0 1 7 】

分岐制御部 3 0 は、命令デコーダ 2 5 から分岐命令の種別を受け取り、演算器 2 8 から分岐先アドレスや分岐条件となる演算の結果を受け取って、演算結果が分岐条件を満たしていれば分岐成立、満たしていなければ分岐不成立の判断を行い、分岐方向を確定する。また、分岐制御部 3 0 は、演算結果と分岐予測時の分岐先アドレスと分岐方向が一致するかどうかの判断や、分岐命令の順序関係の制御も行う。分岐制御部 3 0 は、演算結果と予測とが一致した場合には命令完了制御部 3 2 へ分岐命令の完了通知を出力する。一方、演算結果と予測とが一致しなかった場合には分岐予測失敗を意味するので、分岐制御部 3 0 は、命令完了制御部 3 2 へ分岐命令の完了通知とともに後続命令のキャンセル及び再命令フェッチ要求を出力する。

【 0 0 1 8 】

命令完了ステージでは、命令完了制御部 3 2、レジスタ 3 1、及び分岐履歴更新部 3 3 が動作する。命令完了制御部 3 2 は、命令の各実行資源から受け取った完了通知を基に、コミットスタックエントリに格納された命令コード順に命令完了処理を行い、レジスタ 3 1 の更新指示を出力する。レジスタ 3 1 は、命令完了制御部 3 2 からレジスタ更新指示を受け取ると、演算器 2 8 や一次オペランドキャッシュメモリ 2 9 から受け取る演算結果のデータを基にレジスタの更新を実行する。分岐履歴更新部 3 3 は、分岐制御部 3 0 から受け取る分岐演算の結果を基に分岐予測の履歴更新データを作成し、分岐予測回路 2 2 に出力する。

【 0 0 1 9 】

図 3 は、図 2 に示した命令発行制御部 2 6 の構成例を示す図である。図 3 には、リザベーションステーションの機能を実現する命令発行制御部 2 6 の構成例を示している。図 3 に示す命令発行制御部 2 6 は、複数の出力ポート P A 及び P B を有し、各出力ポート P A 及び P B からそれぞれ 1 つの命令を出力することで複数の命令を同時に出力可能なものである。図 3 には、2 つの出力ポート P A 及び P B を有する例を示している。

【 0 0 2 0 】

命令デコーダ 2 5 でデコードされた命令は、リザベーションステーションのエントリ本体 3 9 の空いているエントリに登録される。登録される内容は、エントリが有効であることを示すバリッドビット (V)、命令におけるディスティネーションレジスタ等の命令オペランドを識別するタグ、及びデコード済みオペコード等である。リザベーションステーションのエントリ本体 3 9 に登録された命令は、取出し可能命令検出部 3 6 により、実行済命令のタグ等に基づいて先行命令との間のレジスタ依存関係が解析され実行可能であると判定されると、エントリ本体 3 9 から取出し可能な命令として検出される。取出し可能な命令は、ポート調停部 3 7 により出力ポート P A 及び P B の調停を受け、調停の結果、出力されることが決定した命令は演算器 2 8 へ送出される。なお、命令デコーダ 2 5 から取出し可能命令検出部 3 6 に命令に係る情報をバイパスする経路を設けることで、命令が 1 クロックサイクルのレイテンシでリザベーションステーションを通過できるようにすることも可能である。発行抑止信号設定部 3 8 は、出力ポート P A 及び P B の命令がオーバーラップ不可である場合には、発行抑止信号を出力する。発行抑止信号が出力されると、ポート調停部 3 7 による調停は行われず、命令の発行が待機させられる。

【 0 0 2 1 】

図 4 (A) 及び (B) は、図 3 の取出し可能命令検出部 3 6 の一部の構成例を示す図であり、エントリ n にバッファされている命令が特定の出力ポート P A 又は P B から取り出

10

20

30

40

50

されることを許可又は禁止する論理回路の一例を示す。図4(A)は、出力ポートPAについてのエントリnに対応する回路を示しており、図4(B)は、出力ポートPBについてのエントリnに対応する回路を示している。

【0022】

図4(A)に示すように、取出し可能命令検出部36は、出力ポートPAについては、論理積(AND)回路41、42、及び否定論理和(NOR)回路43を有する。AND回路41には、信号En_MC_OP及び信号INH_PA_MC_OPが入力される。また、AND回路42には、信号En_FL_A_OP及び信号INH_PA_FL_A_OPが入力される。NOR回路43には、AND回路41及び42の出力信号が入力され、その演算結果が信号En_ENA_PAとして出力される。

10

【0023】

また、図4(B)に示すように、取出し可能命令検出部36は、出力ポートPBについては、AND回路44、45、及びNOR回路46を有する。AND回路44には、信号En_MC_OP及び信号INH_PB_MC_OPが入力される。また、AND回路45には、信号En_FL_A_OP及び信号INH_PB_FL_A_OPが入力される。NOR回路46には、AND回路44及び45の出力信号が入力され、その演算結果が信号En_ENA_PBとして出力される。

【0024】

図4(A)及び(B)において、入力信号En_MC_OPは、エントリnにバッファされている命令が、使用する演算器28を複数サイクル(マルチサイクル)占有し続ける命令であることを示す信号である。入力信号INH_PA_MC_OPは、出力ポートPAに接続されている演算器28が、この演算器28を複数サイクル占有し続ける命令によって既に使用されていることを示し、新たにこの演算器28を使用する命令が出力ポートPAから取り出されることを禁止する信号である。信号En_MC_OPと信号INH_PA_MC_OPとを論理積演算して得られる信号は、エントリnにバッファされている命令が演算器28を複数サイクル占有し続ける命令であって、かつ、出力ポートPAに接続されている演算器28が既に使用されていることから、エントリnの命令が出力ポートPAから取り出されることを禁止する信号である。

20

【0025】

入力信号En_FL_OPは、エントリnにバッファされている命令が、最大の出力遅延サイクル数が固定であるパイプライン化された演算器28を用いる命令であることを示す信号である。ここで、最大の出力遅延サイクル数が固定であるとは、例えば、演算器28の演算レイテンシが4サイクル又は6サイクルである場合に、レイテンシが高々6サイクルであると演算終了以前に予見できるといった意味である。入力信号INH_PA_FL_A_OPは、出力ポートPAに接続されている演算器28であり、最大の出力遅延サイクル数が固定であるパイプライン化された演算器28について、その演算結果出力のための伝送路が他の命令によって使用される見込みであることを示し、新たにこの演算器28を使用する命令が出力ポートPAから取り出されることを禁止する信号である。信号En_FL_A_OPと信号INH_PA_FL_A_OPとを論理積演算して得られる信号は、エントリnにバッファされている命令が最大の出力遅延サイクル数が固定であるパイプライン化された演算器28を用いる命令であり、かつ、その演算結果出力のための伝送路が他の命令によって使用される見込みであることから、エントリnの命令が出力ポートPAから取り出されることを禁止する信号である。出力信号En_ENA_PAは、エントリnにバッファされている命令が出力ポートPAから取り出されることを許可する信号である。なお、図4(B)に示す各信号は、前述した図4(A)に示す各信号を出力ポートPAと出力ポートPBとを入れ換えたものに相当する。

30

40

【0026】

ある演算器の結果出力のための伝送路が他の命令によって使用される状態が起こる場合としては、演算器が複数種類あって、それぞれのレイテンシが異なる場合があげられる。後続命令が用いる小さいレイテンシの演算器の結果出力の伝送路が、先行命令が用いる大きいレイテンシの演算器の結果出力のために使用されることが予め確定している場合には、その伝送路を用いる演算器が接続された出力ポートへの後続命令の出力を禁止する制御

50

を行う。前述した信号En_MC_OP、En_FLA_OPは、命令の種類によって異なる命令実行時の制御を指示する信号であり、命令デコーダ25から送られる。前段のパイプラインステージから命令がエントリに登録された後に、レイテンシ1サイクルで通過することができるリザベーションステーションを構成するために、これらの信号の直前にバイパス経路を設けてもよい。入力信号INH_PA_MC_OP及びINH_PB_MC_OPは、発行抑止信号設定部38の発行抑止信号に対応している。

【0027】

例えば、同時に1命令発行し、アウトオブオーダー実行を行うパイプラインを想定しているが、スーパースカラであっても、インオーダー実行であってもよい。

【0028】

図5は、演算器（命令実行部）28のパイプライン動作を示す図である。演算器28は、例えば、複数のステージングラッチ51及び組み合わせ回路52を有する。パイプライン動作では、クロックサイクル毎に組み合わせ回路52の演算結果が後段のステージングラッチ51に伝達され、スループット1（毎クロックサイクル結果を出力する）の動作が行われる。パイプライン動作は、複数のステージングラッチ51を有し、複数のステージングラッチ51間におけるデータの遷移に単一のクロックのみを必要とする動作である。

【0029】

図6は、演算器（命令実行部）28のマルチサイクル動作を示す図である。例えば、前段の組み合わせ回路52は、後段の組み合わせ回路52の演算結果61を入力して演算を行う。この部分では、複数クロックサイクルで結果が出力されるマルチサイクル動作が行われる。マルチサイクル動作は、複数のステージングラッチ51を有し、複数のステージングラッチ51間におけるデータの遷移に複数のクロックを必要とする動作である。

【0030】

図7は、図5に対応し、スループット1のパイプライン動作を示す図である。パイプライン動作では、単一クロックサイクル動作が行われ、各パイプラインステージ71がスループット1である。命令発行制御部26は、複数の命令を順に発行し、複数の命令をオーバーラップさせることにより、スループットを向上させることができる。

【0031】

図8は、命令発行制御部26の命令発行例を示す図である。純粋マルチサイクル動作81は、例えば除算・平方根の演算であり、複数のステージングラッチ51間におけるデータの遷移に複数のクロックを必要とするとともに、複数のステージングラッチ51間にそれぞれ位置する回路52を、他の命令の実行の演算器28の回路とは共有しない非共有マルチサイクル動作である。非共有完全パイプライン動作82は、例えば乗算・加算の演算であり、他の動作と資源を共有しないパイプライン動作のみの動作である。共有完全パイプライン動作83は、パイプライン動作84～86のみの動作であり、一部のパイプライン動作85が他の動作89と資源（回路）を共有する。複合マルチサイクル動作87は、パイプライン動作88、マルチサイクル動作89及びパイプライン動作90を有し、マルチサイクル動作89が他の動作85と資源（回路）を共有する。

【0032】

図9は、2個の複合マルチサイクル動作91及び95の命令発行を示す図である。横軸は時刻であり、縦軸は命令発行順である。複合マルチサイクル動作91は、図5及び図6の複数のステージングラッチ51を有し、パイプライン動作92、マルチサイクル動作93及びパイプライン動作94を順に行う。パイプライン動作94は、図5に示すように、複数のステージングラッチ51のうち、最終段のステージングラッチ51を含む第1の複数のステージングラッチ51間におけるデータの遷移に単一のクロックのみを必要とする動作である。マルチサイクル動作93は、図6に示すように、複数のステージングラッチ51のうち、上記の第1の複数のステージングラッチ51よりも前段側に位置する第2の複数のステージングラッチ51間におけるデータの遷移に複数のクロックを必要とする動作である。

【0033】

10

20

30

40

50

複合マルチサイクル動作 9 5 は、図 5 及び図 6 の複数の第 2 のステージングラッチ 5 1 を有し、パイプライン動作 9 6、マルチサイクル動作 9 7 及びパイプライン動作 9 8 を順に行う。パイプライン動作 9 6 は、図 5 に示すように、複数の第 2 のステージングラッチ 5 1 のうち、最初段のステージングラッチ 5 1 を含む第 3 の複数のステージングラッチ 5 1 間におけるデータの遷移に単一のクロックのみを必要とする動作である。マルチサイクル動作 9 7 は、図 6 に示すように、複数の第 2 のステージングラッチ 5 1 のうち、上記の第 3 の複数のステージングラッチ 5 1 よりも後段側に位置する第 4 の複数のステージングラッチ 5 1 間におけるデータの遷移に複数のクロックを必要とする動作である。ここで、マルチサイクル動作 9 3 及び 9 7 は、資源を共有しているため、複合マルチサイクル動作 9 1 及び 9 5 は相互にオーバーラップさせることが困難であり、スループット低下の原因になる。本実施形態では、一部オーバーラップさせることにより、スループットを向上させる。その詳細は、後に図 1 1 を参照しながら説明する。

10

【 0 0 3 4 】

図 1 0 は、複合マルチサイクル動作 1 0 1 及び共有完全パイプライン動作 1 0 5 の命令発行を示す図である。複合マルチサイクル動作 1 0 1 は、パイプライン動作 1 0 2、マルチサイクル動作 1 0 3 及びパイプライン動作 1 0 4 を順に行う。共有完全パイプライン動作 1 0 5 は、複数の第 2 のステージングラッチ 5 1 を有し、パイプライン動作 1 0 6、パイプライン動作 1 0 7 及びパイプライン動作 1 0 8 を順に行う。ここで、マルチサイクル動作 1 0 3 及びパイプライン動作 1 0 7 は、資源を共有しているため、複合マルチサイクル動作 1 0 1 及び共有完全パイプライン動作 1 0 5 は相互にオーバーラップさせることが困難であり、スループット低下の原因になる。パイプライン動作 1 0 6 は、複数の第 2 のステージングラッチ 5 1 のうち、最初段のステージングラッチ 5 1 を含む第 3 の複数のステージングラッチ 5 1 間におけるデータの遷移に単一のクロックのみを必要とし、上記の第 3 の複数のステージングラッチ 5 1 間にそれぞれ位置する回路 5 2 を、他の命令の実行で使用する演算器 2 8 の回路とは共有しない非共有パイプライン動作である。パイプライン動作 1 0 7 は、複数の第 2 のステージングラッチ 5 1 のうち、上記の第 3 の複数のステージングラッチ 5 1 よりも後段側に位置する第 4 の複数のステージングラッチ 5 1 間におけるデータの遷移に単一のクロックのみを必要とし、上記の第 4 の複数のステージングラッチ 5 1 間にそれぞれ位置する回路 5 2 を、他の命令の実行に使用する演算器 2 8 の回路と共有する共有パイプライン動作である。本実施形態では、一部オーバーラップさせることにより、スループットを向上させる。その詳細は、後に図 1 2 を参照しながら説明する。

20

30

【 0 0 3 5 】

図 1 1 は、図 9 に対応し、2 個の複合マルチサイクル動作 9 1 及び 9 5 の命令発行を示す図である。マルチサイクル動作 9 3 及び 9 7 は、資源を共有している。そのため、命令発行制御部 2 6 がマルチサイクル動作 9 3 を発行している期間 1 1 1 では、図 3 の発行抑止信号設定部 3 8 は発行抑止信号を取出し可能命令検出部 3 6 に出力する。これにより、期間 1 1 1 では、取出し可能命令検出部 3 6 は、マルチサイクル動作 9 7 の発行を禁止する。2 個の複合マルチサイクル動作 9 1 及び 9 5 は、相互に一部を時間的にオーバーラップさせることができる。具体的には、パイプライン動作 9 6 は、マルチサイクル動作 9 3 とオーバーラップする。マルチサイクル動作 9 7 は、パイプライン動作 9 4 とオーバーラップする。これにより、スループットを向上させることができる。特に、レイテンシが長い処理をオーバーラップさせることの効果は大きい。

40

【 0 0 3 6 】

なお、パイプライン動作 9 6 は、マルチサイクル動作 9 3 の他、パイプライン動作 9 2 の一部にもオーバーラップさせることができる。また、パイプライン動作 9 8 は、パイプライン動作 9 4 の一部にオーバーラップさせることができる。

【 0 0 3 7 】

図 1 2 は、図 1 0 に対応し、複合マルチサイクル動作 1 0 1 及び共有完全パイプライン動作 1 0 5 の命令発行を示す図である。マルチサイクル動作 1 0 3 及びパイプライン動作

50

107は、資源を共有している。そのため、命令発行制御部26がマルチサイクル動作103を発行している期間121では、図3の発行抑止信号設定部38は発行抑止信号を取出し可能命令検出部36に出力する。これにより、期間121では、取出し可能命令検出部36は、パイプライン動作107の発行を禁止する。複合マルチサイクル動作101及び共有完全パイプライン動作105は、相互に一部を時間的にオーバーラップさせることができる。具体的には、パイプライン動作106は、マルチサイクル動作103とオーバーラップする。パイプライン動作107は、パイプライン動作104とオーバーラップする。パイプライン動作108は、パイプライン動作104とオーバーラップする。これにより、スループットを向上させることができる。特に、レイテンシが長い処理をオーバーラップさせることの効果は大きい。なお、パイプライン動作106は、マルチサイクル動作103の他、パイプライン動作102の一部にもオーバーラップさせることができる。

10

【0038】

図13は、マルチサイクル演算命令の発行抑止信号135及び136を用いて動作を部分的にオーバーラップさせる方法を示す図である。本実施形態では、部分的なパイプライン制御を導入し、演算処理のオーバーラップを実現するため、オーバーラップ可能な命令の最大数だけ命令情報ラッチを用意する。言い換えると、1つのパイプラインステージが複数クロックサイクルにわたるパイプライン処理を行う。演算器28に対して2命令までオーバーラップさせようとした場合、演算器28全体が2つの仮想的なパイプラインステージに分割されているものとして制御する。命令の状態は、2つのパイプラインステージに対応して保持される。図13のタイミングチャートは、制御信号を示し、発行から数サイ

20

【0039】

先行命令は、パイプライン1段目信号131及びパイプライン2段目信号132を有する。後続命令は、パイプライン1段目信号133及びパイプライン2段目信号134を有する。命令発行制御部26は、先行命令に応じて、パイプライン1段目信号131を出力し、その後、パイプライン2段目信号132を出力する。パイプライン1段目信号131が出力されると、発行抑止信号設定部38は発行抑止信号135を出力する。命令発行制御部26は、発行抑止信号135の出力が終了するまで、後続命令のマルチサイクル演算命令の発行を抑止し、発行抑止信号135の出力が終了すると、後続命令のマルチサイ

30

【0040】

図14は、演算命令のサイクルステージを説明するための図である。サイクルステージでは、P、B1、B2、X1～Xnが順に行われる。Pは、実行可能な命令の調停及び取り出しを行うパイプライン処理のサイクルステージである。B1は、レジスタ読み出し1サイクル目のパイプライン処理のサイクルステージである。B2は、レジスタ読み出し2サイクル目のパイプライン処理のサイクルステージである。X1～Xnは、演算の実行サイ

40

【0041】

図15～図17は、命令発行制御部26の制御方法を示すタイミングチャートであり、時間の経過に伴う信号及び命令の状態変化を示す。時間は、左から右へ流れている。上段の両方向矢印の線分は命令情報1を保持するラッチの信号状態を示し、下段の両方向矢印の線分は命令情報2を保持するラッチの信号状態を示す。片方向の矢印は、信号及び状態

50

変化についての因果関係を表している。例えば、 $A \rightarrow B$ は、 A を契機（条件）として B が変化することを示している。ただし、 A が、 B の変化の必要条件に過ぎない場合がある。

【0042】

サイクルは、命令の処理段階（命令ステージ）のことであり、回路構成がパイプライン動作及びマルチサイクル動作のいずれであっても命令ステージは毎クロックサイクル遷移する（同一サイクルが続くような待ち状態はない）ものとして表記する。この例では、発行サイクル P から実行サイクル X_1 までのレイテンシが3クロックサイクルである例を示している。発行サイクル P から実行サイクル X_1 までのレイテンシはこれに限られない。発行サイクル P の前にレジスタ読み出しサイクル B_1 、 B_2 を行う構成であってもよい。

【0043】

図15は、図11に対応し、先行命令が複合マルチサイクル動作91であり、後続命令が複合マルチサイクル動作95である場合を示す図である。先行命令と後続命令とは、レジスタ依存関係がなく、演算順序の制約を持たないものとする。依存関係のある命令同士の場合は、演算処理 $X_1 \sim X_m$ をオーバーラップして実行することができない。

【0044】

複合マルチサイクル動作を行う先行命令と複合マルチサイクル動作を行う後続命令との演算処理がオーバーラップするクロックサイクル数を m とする。オーバーラップクロックサイクル数 m は、先行命令の複合マルチサイクル動作91の最後部パイプライン動作94のクロックサイクル数と、後続命令の複合マルチサイクル動作95の先頭部のパイプライン動作96のクロックサイクル数との和とすることが好適であるが、これより少なくても良い。

【0045】

複合マルチサイクル動作を行う先行命令が発行されたことにより、発行抑止信号設定部38は、先行命令のサイクル P で発行抑止信号に1をセットする。これにより、次のクロックサイクルで発行抑止信号が1となる。発行抑止信号が1となることにより、後続命令のマルチサイクル演算命令に対して発行抑止がかかる。すなわち、発行条件を満たさなくなり、命令発行制御部26は、命令を発行しなくなる。また、既に発行されている可能性のある次のクロックサイクルでサイクル P となるマルチサイクル演算命令に対してはキャンセル処理が行われる。キャンセルにより命令は無効となる。発行抑止信号を1にすることにより、同一の演算回路に対して複数の命令による演算処理が競合することが防がれる。

【0046】

複合マルチサイクル動作を行う先行命令が発行された後、演算器28は、サイクル B_1 及び B_2 でレジスタなどからオペランドデータを受け取り、サイクル X_1 からオペランドデータを用いて演算を開始する。先行命令のサイクル X_1 では、命令の情報（有効フラグ、命令種、命令タグ、結果を書き込むレジスタなどを含む）が命令情報1のラッチにセットされる。命令の情報は、演算処理が行われている間保持される。

【0047】

演算の終了時点をサイクル X_n と表記するが、演算開始時点では n の値は未定である。マルチサイクル演算命令は、演算開始から演算終了までのサイクル数（演算レイテンシ）が発行の時点では不定な命令である。演算命令の種類や、演算データのパターンにより演算レイテンシは変化する。演算レイテンシは、演算器制御回路27により決定される。マルチサイクル演算命令の場合、演算器制御回路27は、演算終了の $k + m$ サイクル前の実行サイクル X_{n-k-m} までに実行サイクル数 n を確定することができる。命令発行制御部26は、先行命令の演算終了の $k + m$ サイクル前の実行サイクル X_{n-k-m} に演算終了予告信号が演算器制御回路27から通知され、演算終了サイクル X_n の時刻が確定する。発行抑止信号設定部38は、命令情報1を保持しているラッチの有効フラグが、命令が有効であると示していて、命令種が複合マルチサイクル動作の命令であることを示しており、かつ、命令の状態が実行サイクル X_{n-p-m} となっている場合、発行抑止信号を0にリセットする。

10

20

30

40

50

【0048】

その後、例えば、複合マルチサイクル動作を行う先行命令がサイクル $X_{n-p-m+2}$ に、複合マルチサイクル動作を行う後続命令が発行される。命令情報1を保持しているラッチの有効フラグが、命令が有効であると示していて、かつ、命令の状態がサイクル X_{n-m} となっているとき、命令情報1を保持しているラッチの内容が、命令情報2を保持しているラッチへ移動する。これにより、命令情報1を保持しているラッチに新たに後続命令の情報を保持することが可能となる。この命令情報の移動のタイミングは、サイクル X_{n-m} であることが好適である。サイクル X_{n-m} ではない構成も可能であるが、 n のとれる値の範囲が狭まり、演算レイテンシ n の最小値の制約が大きくなる。または、オーバーラップ量 m が小さくなる。

10

【0049】

命令情報の移動タイミングをサイクル $X_{n-m'}$ とした場合、具体的なデメリットは、複合マルチサイクル動作を行う先行命令と複合マルチサイクル動作を行う後続命令とについて、命令情報2のラッチの情報が保持されている期間について注目すると、 $m' - n - m$ 、つまり、 $m + m' - n$ となることである。したがって、 n のとれる値の最小値が大きくなる、又は、オーバーラップ量 m が小さくなる。

【0050】

なお、命令情報1のラッチについて注目すると、 $n - m' - n - m$ 、つまり、 $m - m'$ である。以上より、 $m = m'$ となることが好適である。

【0051】

20

複合マルチサイクル動作を行う後続命令のサイクル X_1 では、複合マルチサイクル動作を行う先行命令と同様に、命令情報1がラッチにセットされる。命令情報1は、複合マルチサイクル演算を行っている間保持される。先行命令がサイクル X_n になると演算処理は終了し、命令情報2を保持しているラッチの内容は、図示されていない後続の命令処理段階に対応するラッチへ移動する。

【0052】

複合マルチサイクル動作を行う先行命令のサイクル X_{n-m+1} からサイクル X_n までの間の m クロックサイクルは、複合マルチサイクル動作を行う後続命令の演算処理（サイクル X_1 以降の m サイクル）とオーバーラップして実行されており、演算器28のスループットが向上している。例えば、複合マルチサイクル動作を用いる命令を連続して実行する場合のスループットは $n / (n - m)$ 倍となる。

30

【0053】

次に、後続命令が複合マルチサイクル動作を用いる命令の場合を説明する。後続命令がマルチサイクル演算命令の場合、演算終了の $k + m$ サイクル前までに演算レイテンシが確定し、命令発行制御部26は、サイクル X_{n-k-m} で演算器制御回路27より演算終了予告信号が通知される。発行抑止信号設定部38は、命令情報1を保持しているラッチの有効フラグが、命令が有効であると示していて、命令種が複合マルチサイクル動作を用いる命令であることを示しており、かつ、命令の状態がサイクル X_{n-p-m} となっているとき、発行抑止信号を0にリセットする。ここで、先行命令のサイクル X_n と後続命令のサイクル X_{n-p-m} との間の時刻の前後関係は不定である。

40

【0054】

命令情報1を保持しているラッチの有効フラグが、命令が有効であると示していて、命令の状態がサイクル X_{n-m} となっているとき、命令情報1を保持しているラッチの内容が、命令情報2を保持しているラッチへ移動する。先行命令の情報は、命令情報2を保持しているラッチから既に移動しており、衝突しない。ここでは、命令情報1及び2のラッチを持つ場合、 $m \leq n - m$ という制約を仮定している。

【0055】

図16は、先行命令が複合マルチサイクル動作であり、後続命令が純粋マルチサイクル動作である場合を示す図である。先行命令の複合マルチサイクル動作は、図15の先行命令と同じである。後続命令の純粋マルチサイクル動作は、図8の純粋マルチサイクル動作

50

8 1と同じであり、複数の第2のステージングラッチ5 1を有し、複数の第2のステージングラッチ5 1間におけるデータの遷移に複数のクロックを必要とするとともに、複数の第2のステージングラッチ5 1間にそれぞれ位置する回路5 2を、他の命令の実行で使用する演算器2 8の回路とは共有しない非共有マルチサイクル動作である。図1 6のタイミングチャートは、図1 5のタイミングチャートに対して、後続命令のサイクル $Xn - k - m$ までは同じである。以下、図1 6が図1 5と異なる点を説明する。

【0056】

後続命令（純粹マルチサイクル動作）は、複合マルチサイクル動作を行う先行命令のサイクル $Xn - p - m + 2$ のタイミングで発行される。図1 6では、図1 5に対して、後続命令の状態に起因する発行抑止信号のリセットのタイミングが変わっている。発行抑止信号設定部3 8は、命令情報2を保持しているラッチの有効フラグが、保持された命令が有効であると示している、命令種が純粹マルチサイクル動作の命令であることを示しており、かつ、命令の状態がサイクル $Xn - p$ となっているとき、発行抑止信号を0にリセットする。

10

【0057】

この場合も、複合マルチサイクル動作を行う先行命令のサイクル $Xn - m + 1$ からサイクル Xn までの間の m クロックサイクルは、後続命令の演算処理（サイクル $X1$ 以降の m サイクル）とオーバーラップして実行されており、演算器2 8のスループットが向上している。

【0058】

20

図1 7は、図1 2に対応し、先行命令が複合マルチサイクル動作1 0 1であり、後続命令が共有完全パイプライン動作1 0 5である場合を示す図である。図1 7のタイミングチャートは、図1 5のタイミングチャートに対して、先行命令のサイクル $Xn - p - m$ までは同じである。以下、図1 7が図1 5と異なる点を説明する。

【0059】

後続命令（共有完全パイプライン動作）は、複合マルチサイクル動作を行う先行命令のサイクル $Xn - p - m + 2$ のタイミングで発行される。先行命令のサイクル $Xn - p - m + 2$ サイクルのタイミング以降では、発行抑止信号が0であることにより、後続命令が発行抑止されなくなる。これは、先行命令と後続命令とで演算器2 8内の演算回路が競合しなくなったためである。これにより、後続命令は、抑止されずに、パイプライン動作を実行する。

30

【0060】

この場合も、複合マルチサイクル動作を行う先行命令のサイクル $Xn - m + 1$ からサイクル Xn までの間の m クロックサイクルは、共有完全パイプライン動作を行う後続命令の演算処理（サイクル $X1$ 以降の m サイクル）とオーバーラップして実行されており、演算器2 8のスループットが向上している。

【0061】

図1 5～図1 7では、命令発行制御部（命令制御部）2 6は、最終に行うパイプライン動作及びその前に行うマルチサイクル動作を含む複合マルチサイクル動作の先行命令（第1の命令）と、後続命令（第2の命令）とを入力する。そして、命令発行制御部2 6は、先行命令の実行と後続命令の実行とが一部重複するように、先行命令を演算器（命令実行部）2 8に発行するとともに、後続命令を演算器（命令実行部）2 8に発行する。

40

【0062】

図1 5では、後続命令は、最初に行うパイプライン動作及びその後に行うマルチサイクル動作を含む複合マルチサイクル動作の命令である。図1 6では、後続命令は、非共有のマルチサイクル動作の命令である。図1 7では、後続命令は、最初に行う非共有のパイプライン動作及びその後に行う共有のパイプライン動作を含む共有完全パイプライン動作の命令である。発行抑止信号設定部3 8は、発行抑止信号のリセットタイミングを命令種によって切り替える。

【0063】

50

命令発行制御部 26 は、先行命令のマルチサイクル動作が後続命令と資源を共有する期間では後続命令の発行を抑止する。先行命令の最終に行うパイプライン動作は、後続命令の動作とオーバーラップするように発行される。より好ましくは、先行命令の最終に行うパイプライン動作及びその前に行うマルチサイクル動作は、後続命令の動作とオーバーラップするように発行される。これにより、スループットを向上させることができる。

【0064】

命令発行制御部 26 は、先行命令を実行する場合、ステージングラッチ 51 間に位置するいずれかの回路 52 が、後続命令を実行することによりステージングラッチ 51 間に位置する回路と共有されるときは、後続命令の演算器 28 への発行を抑止する。

【0065】

また、命令発行制御部 26 は、先行命令の実行における最後のパイプライン動作と、後続命令の実行とが一部重複するように、先行命令及び後続命令を演算器 28 に発行する。また、命令発行制御部 26 は、先行命令の実行における最後のパイプライン動作又はその前のマルチサイクル動作と、後続命令の実行とが一部重複するように、先行命令及び後続命令を演算器 28 に発行する。

【0066】

なお、上記実施形態は、何れも本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。

【符号の説明】

【0067】

- 11 プロセッサ
- 21 命令フェッチ部
- 22 分岐予測回路
- 23 一次命令キャッシュメモリ
- 24 命令バッファ
- 25 命令デコーダ
- 26 命令発行制御部
- 27 演算器制御回路
- 28 演算器
- 29 一次オペランドキャッシュメモリ
- 30 分岐制御部
- 31 レジスタ
- 32 命令完了制御部
- 33 分岐履歴更新部
- 34 二次キャッシュメモリ
- 35 メモリコントローラ
- 36 取り出し可能命令検出部
- 37 ポート調停部
- 38 発行抑止信号設定部
- 39 エントリ本体

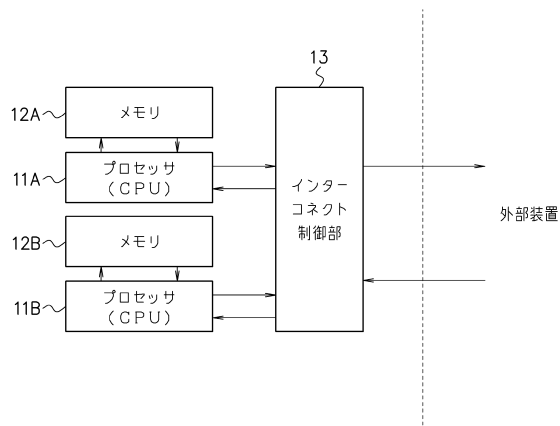
10

20

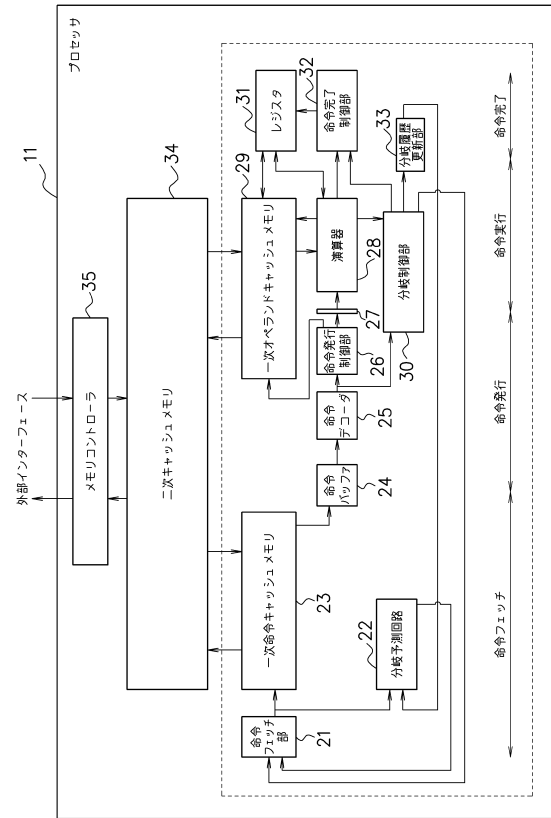
30

40

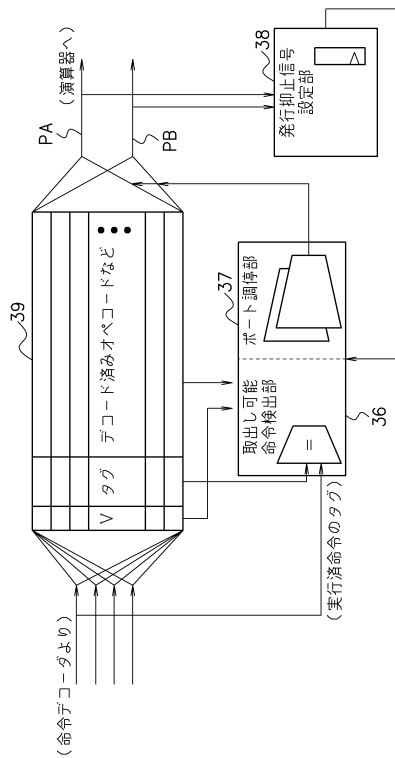
【図 1】



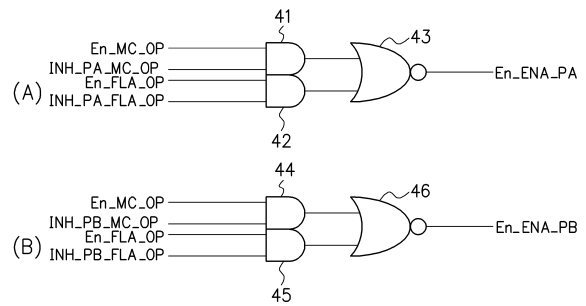
【図 2】



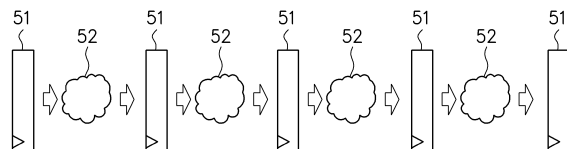
【図 3】



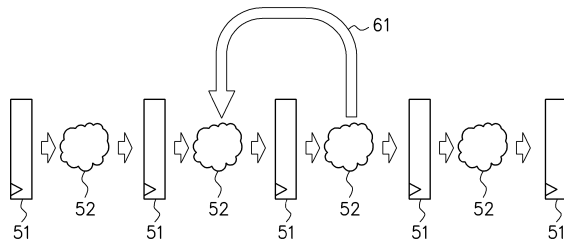
【図 4】



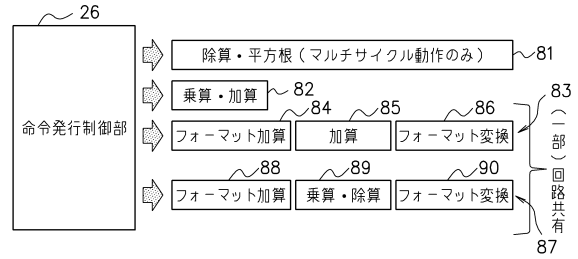
【図 5】



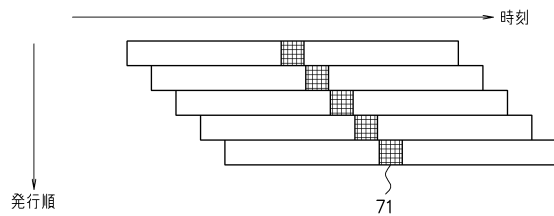
【図 6】



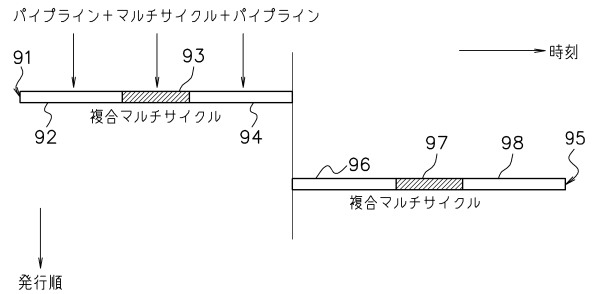
【図 8】



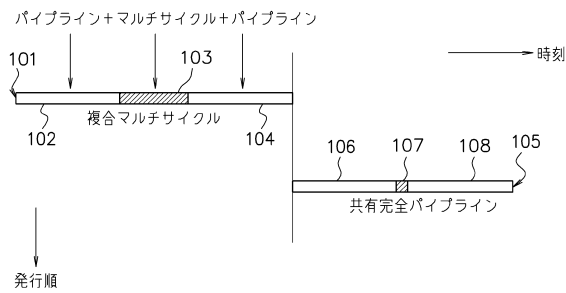
【図 7】



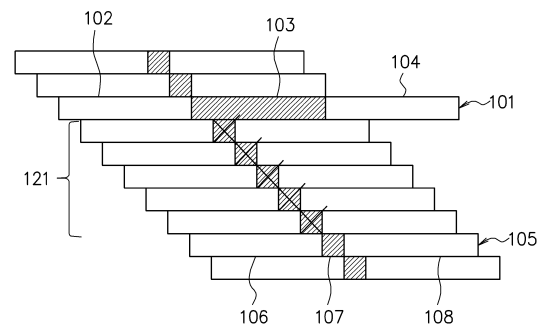
【図 9】



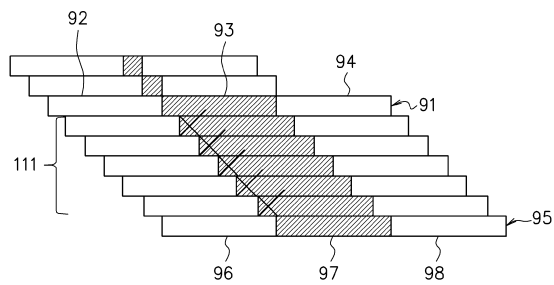
【図 10】



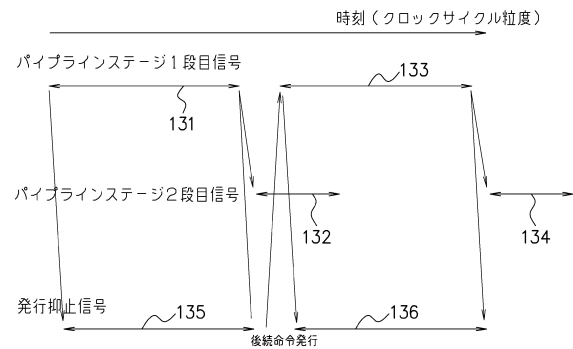
【図 12】



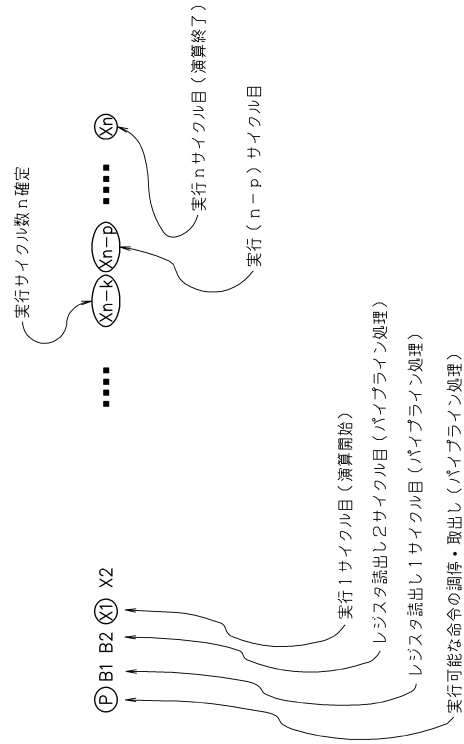
【図 11】



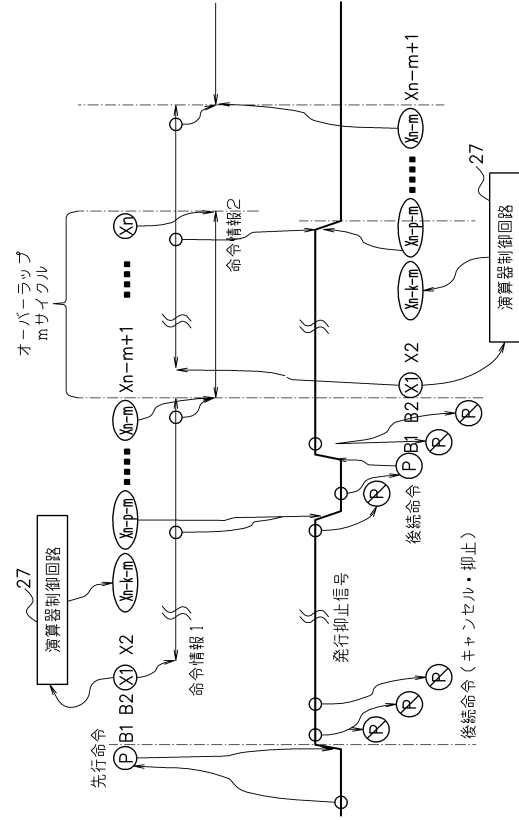
【図 13】



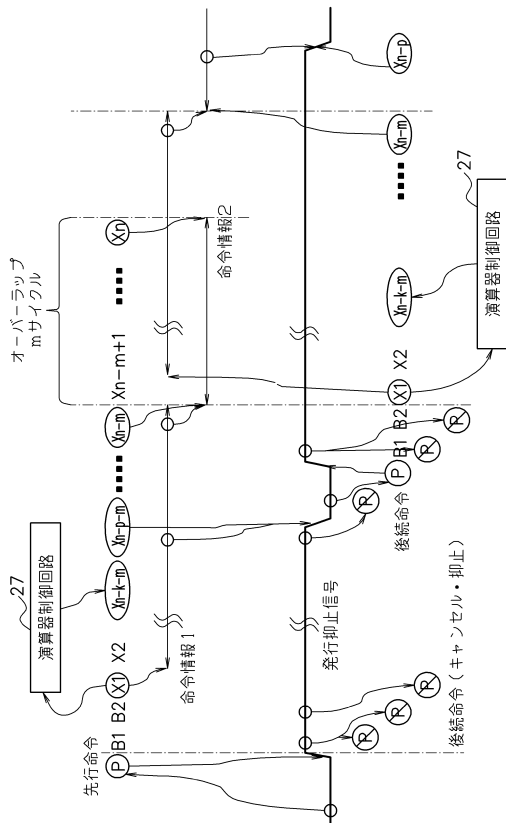
【図 14】



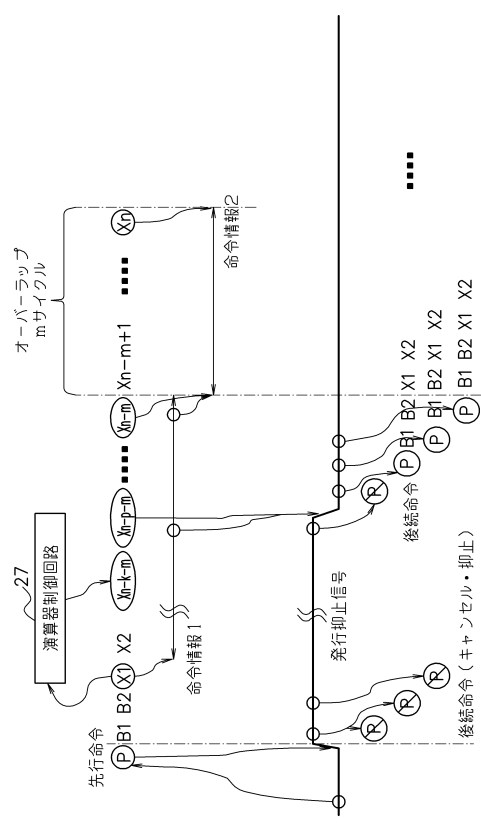
【図 15】



【図 16】



【図 17】



フロントページの続き

(56)参考文献 特開平 0 7 - 2 4 4 5 8 8 (J P , A)
特開平 0 8 - 3 0 5 5 6 7 (J P , A)
米国特許第 0 5 6 0 4 8 7 8 (U S , A)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 9 / 3 0 - 9 / 4 2
G 0 6 T 1 / 0 0 - 1 / 4 0
G 0 6 T 3 / 0 0 - 5 / 5 0
G 0 6 T 9 / 0 0 - 9 / 4 0