

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2018-45370

(P2018-45370A)

(43) 公開日 平成30年3月22日(2018.3.22)

(51) Int.Cl.		F I				テーマコード (参考)
G06F 9/38 (2006.01)		G06F 9/38		310F		5B013
G06F 9/30 (2018.01)		G06F 9/38		350B		5B033
G06F 9/318 (2006.01)		G06F 9/30		350F		
		G06F 9/30		320A		

審査請求 未請求 請求項の数 12 O L (全 35 頁)

(21) 出願番号	特願2016-178608 (P2016-178608)	(71) 出願人	000005223
(22) 出願日	平成28年9月13日 (2016.9.13)		富士通株式会社
			神奈川県川崎市中原区上小田中4丁目1番1号
		(74) 代理人	100072718
			弁理士 古谷 史旺
		(74) 代理人	100151002
			弁理士 大橋 剛之
		(74) 代理人	100201673
			弁理士 河田 良夫
		(72) 発明者	坂下 聡太
			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
		Fターム(参考)	5B013 AA12 CC10
			5B033 BB00 BE05

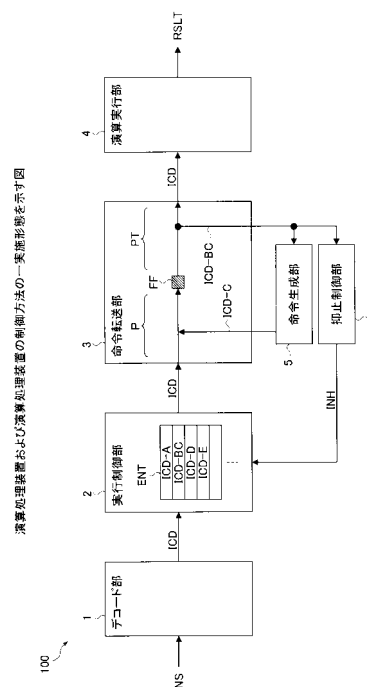
(54) 【発明の名称】 演算処理装置および演算処理装置の制御方法

(57) 【要約】

【課題】 複合命令を実行する演算処理装置の性能を向上する。

【解決手段】 演算処理装置は、命令をデコードするデコード部と、デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力する実行制御部と、実行制御部が出力する命令を順次転送する命令転送部と、命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、複合命令から生成した個別命令を命令転送部に出力する命令生成部と、命令転送部が転送した個別命令を実行する演算実行部を有する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

命令をデコードするデコード部と、
前記デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力する実行制御部と、
前記実行制御部が出力する命令を順次転送する命令転送部と、
前記命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、前記複合命令から生成した個別命令を前記命令転送部に出力する命令生成部と、
前記命令転送部が転送した個別命令を実行する演算実行部を有することを特徴とする演算処理装置。

10

【請求項 2】

前記命令生成部が前記複合命令に含まれる個別命令を前記命令転送部に出力する場合、前記実行制御部に命令の出力を抑止させる抑止情報を出力する抑止制御部を有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】

前記実行制御部が出力する前記複合命令は、前記複合命令であることを識別する識別情報を含み、

前記抑止制御部は、前記実行制御部が出力する前記識別情報に基づいて、前記抑止情報を出力することを特徴とする請求項 2 記載の演算処理装置。

【請求項 4】

前記命令転送部は、前記抑止制御部が前記抑止情報を出力しない場合、前記実行制御部が出力する前記識別情報を選択し、前記抑止制御部が前記抑止情報を出力する場合、命令が前記複合命令でないことを示す識別情報を選択する第 1 の選択部を有し、前記第 1 の選択部で選択した識別情報を転送することを特徴とする請求項 3 記載の演算処理装置。

20

【請求項 5】

前記命令転送部は、前記抑止制御部が前記抑止情報を出力しない場合、前記実行制御部が出力する命令を選択し、前記抑止制御部が前記抑止情報を出力する場合、前記命令生成部が出力する個別命令を選択する第 2 の選択部を有し、前記第 2 の選択部で選択した命令を転送することを特徴とする請求項 2 ないし請求項 4 のいずれか 1 項記載の演算処理装置。

30

【請求項 6】

前記実行制御部が出力する命令は、命令が有効であることを示す有効情報を含み、

前記命令転送部は、前記実行制御部が出力する有効情報または前記抑止情報を有効情報として前記演算実行部に転送する論理和部を有することを特徴とする請求項 2 ないし請求項 5 のいずれか 1 項記載の演算処理装置。

【請求項 7】

前記命令生成部は、前記実行制御部が前記複合命令を出力する場合、前記複数の個別命令のうち、前記演算実行部による実行順が 2 番目以降の個別命令を生成し、

前記命令転送部は、前記実行制御部が前記複合命令を出力する場合、前記複合命令の転送に続いて、前記命令生成部が生成する実行順が 2 番目以降の個別命令を転送し、

40

前記演算実行部は、前記複合命令を受信した場合、前記複数の命令のうち実行順が最初の命令を実行することを特徴とする請求項 1 ないし請求項 6 のいずれか 1 項記載の演算処理装置。

【請求項 8】

前記命令転送部は、

前記実行制御部が出力する命令を受信する第 1 のステージと、

前記第 1 のステージから転送される命令を前記演算実行部に出力する第 2 のステージを有し、

前記命令生成部は、前記第 2 のステージに転送される命令が前記複合命令の場合、前記実行順が 2 番目以降の個別命令を前記第 1 のステージに出力することを特徴とする請求項

50

7 記載の演算処理装置。

【請求項 9】

前記実行制御部が順次出力する先行命令および後続命令が、前記先行命令の演算結果を前記後続命令で使用する所定の組み合わせの特定命令であることを検出する依存検出部を有し、

前記命令転送部は、前記実行制御部が出力する命令が前記複合命令の場合、または、前記依存検出部が前記特定命令を検出した場合、前記実行制御部の出力の選択を停止し、前記命令生成部の出力を選択する選択部を有し、

前記命令生成部は、前記実行制御部が出力する命令が前記複合命令の場合、前記複合命令に含まれる個別命令を前記命令転送部に出力し、前記実行制御部が出力する命令が前記複合命令でない場合、前記命令転送部で転送された命令を前記命令転送部に出力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 10】

前記命令生成部が前記複合命令に含まれる個別命令を前記命令転送部に出力する場合、または前記依存検出部が前記特定命令を検出した場合、前記実行制御部に命令の出力を抑制させる抑制情報を出力する抑制制御部を有し、

前記選択部は、前記抑制制御部が前記抑制情報を出力しない場合、前記実行制御部が出力する命令を選択し、前記抑制制御部が前記抑制情報を出力する場合、前記命令生成部が出力する命令を選択することを特徴とする請求項 9 記載の演算処理装置。

【請求項 11】

前記実行制御部が出力する命令は、命令が有効であることを示す有効情報を含み、

前記命令転送部は、前記依存検出部が前記特定命令を検出した場合、前記実行制御部が出力した前記後続命令に含まれる有効情報を無効状態に設定する無効設定部を有することを特徴とする請求項 9 または請求項 10 記載の演算処理装置。

【請求項 12】

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デコード部が、命令をデコードし、

前記演算処理装置が有する実行制御部が、前記デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力し、

前記演算処理装置が有する命令転送部が、前記実行制御部が出力する命令を順次転送し

、前記命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、前記演算処理装置が有する命令生成部が、前記複合命令から生成した個別命令を前記命令転送部に出力し、

前記演算処理装置が有する演算実行部が、前記命令転送部が転送した個別命令を実行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、演算処理装置および演算処理装置の制御方法に関する。

【背景技術】

【0002】

プロセッサ等の演算処理装置は、デコード部がデコードした命令をリザベーションステーションに保持し、演算実行部で実行する命令の順序をデータの依存関係に応じて入れ換えるアウトオブオーダー処理を実行する。実行可能な順に命令を実行することで、デコードされた順に命令を実行する場合に比べて、命令の実行効率は向上する。ところで、乗算、減算に加えて加算、減算が実行可能な乗算器に命令を投入する第 1 のリザベーションステーションと、加算、減算のみが実行可能な加算器に命令を投入する第 2 のリザベーションステーションとを有する演算処理装置が提案されている。演算処理装置は、第 1 および第 2 のリザベーションステーションのそれぞれに対応して、デコードした命令をリザベーシ

ョンステーションに投入する前に保持する第 1 のレジスタと第 2 のレジスタとを有する。

【0003】

第 1 および第 2 のレジスタのいずれかから出力される乗算命令または除算命令は、専用のレジスタに格納する先行フローと、専用のレジスタに格納された乗算命令または除算命令を第 1 のリザベーションステーションに投入する後続フローとに展開される。これにより、第 1 および第 2 のレジスタに加算命令または減算命令のみが保持される場合、デコードサイクルでのリザベーションステーションへの命令の投入に掛かる時間は、命令を複数のフローに展開しない場合に比べて短縮される（例えば、特許文献 1 参照）。

【0004】

また、デコードした命令に対応するマイクロプログラムの制御情報を、フロー毎に保持する制御リザベーションステーションを設けることで、アウトオブオーダーで実行される命令は、マイクロプログラムにより制御可能になる（例えば、特許文献 2 参照）。

【先行技術文献】

【特許文献】

【0005】

【特許文献 1】特開 2004 - 38751 号公報

【特許文献 2】特開 2000 - 181702 号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

命令をデコードするデコード部が、複数の個別命令を組み合わせた複合命令を複数の個別命令に分解する機能を有する場合、分解された複数の個別命令は、リザベーションステーションに投入される。この場合、リザベーションステーションの使用率は、複合命令をリザベーションステーションに投入する場合に比べて上昇し、アウトオブオーダーの効率は低下する。

【0007】

1 つの側面では、本発明は、複合命令を実行する演算処理装置の性能を向上することを目的とする。

【課題を解決するための手段】

【0008】

一つの観点によれば、演算処理装置は、命令をデコードするデコード部と、デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力する実行制御部と、実行制御部が出力する命令を順次転送する命令転送部と、命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、複合命令から生成した個別命令を命令転送部に出力する命令生成部と、命令転送部が転送した個別命令を実行する演算実行部を有する。

【0009】

別の観点によれば、演算処理装置の制御方法は、演算処理装置が有するデコード部が、命令をデコードし、演算処理装置が有する実行制御部が、デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力し、演算処理装置が有する命令転送部が、実行制御部が出力する命令を順次転送し、命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、演算処理装置が有する命令生成部が、複合命令から生成した個別命令を命令転送部に出力し、演算処理装置が有する演算実行部が、命令転送部が転送した個別命令を実行する。

【発明の効果】

【0010】

複合命令を実行する演算処理装置の性能を向上することができる。

【図面の簡単な説明】

【0011】

【図 1】演算処理装置および演算処理装置の制御方法の一実施形態を示す図である。

10

20

30

40

50

【図 2】図 1 に示す演算処理装置の動作の一例を示す図である。

【図 3】演算処理装置および演算処理装置の制御方法の別の実施形態を示す図である。

【図 4】図 3 に示すレジスタ管理部およびレジスタ部の一例を示す図である。

【図 5】図 3 に示す演算処理装置における固定小数点演算器に対応する命令パイプラインの概要を示す図である。

【図 6】図 5 に示す命令パイプラインにおいて、ステージ P からステージ X までに関連する回路要素の一例を示す図である。

【図 7】図 5 および図 6 に示す命令パイプラインにおいて、オペランドデータをバイパスさせるタイミングの一例を示す図である。

【図 8】図 3 に示す演算処理装置の動作の一例を示す図である。

10

【図 9】図 3 に示す演算処理装置の動作の別の例を示す図である。

【図 10】図 6 に示す命令パイプラインにおける P サイクルおよび P T サイクルでの動作フローの一例を示す図である。

【図 11】演算処理装置および演算処理装置の制御方法の別の実施形態において、リザーベーションステーションから固定小数点演算器までの命令パイプラインの一例を示す図である。

【図 12】図 11 に示す命令パイプラインを有する演算処理装置の動作の一例を示す図である。

【図 13】図 11 に示す命令パイプラインを有する演算処理装置の動作の別の例を示す図である。

20

【図 14】図 11 に示す命令パイプラインを有する演算処理装置の動作の別の例である。

【図 15】図 11 に示す命令パイプラインにおける P サイクルおよび P T サイクルでの動作フローの一例を示す図である。

【発明を実施するための形態】

【0012】

以下、図面を用いて実施形態を説明する。

【0013】

図 1 は、演算処理装置および演算処理装置の制御方法の一実施形態を示す。図 1 に示す演算処理装置 100 は、CPU (Central Processing Unit) 等のプロセッサであり、デコード部 1、実行制御部 2、命令転送部 3、演算実行部 4、命令生成部 5 および抑止制御部 6 を有する。

30

【0014】

デコード部 1 は、命令キャッシュ等から出力される命令 I N S をデコードし、デコードにより得られた命令コード I C D およびレジスタ番号等を実行制御部 2 に出力する。以下の説明では、デコードにより得られた命令コード I C D およびレジスタ番号等の情報は、命令 I C D とも称される。実行制御部 2 は、デコード部 1 から出力される命令 I C D を保持する複数のエントリ E N T を有し、エントリ E N T に保持した命令 I C D を実行可能な順に出力するアウトオブオーダー処理を実行する。但し、実行制御部 2 は、抑止情報 I N H を受けている間、命令 I C D の出力を抑止する。

【0015】

40

命令転送部 3 は、実行制御部 2 が出力する命令 I C D を順次転送し、演算実行部 4 に出力する。例えば、命令転送部 3 は、クロックに同期して命令 I C D を保持するフリップフロップ F F を挟んで、ステージ P の処理部とステージ P T の処理部とを有し、命令パイプラインの一部として機能する。以下の説明では、ステージ P の処理部とステージ P T の処理部とは、ステージ P、ステージ P T とも称される。ステージ P は、実行制御部 2 から出力される命令 I C D を受信してフリップフロップ F F に出力する。ステージ P T は、フリップフロップ F F に命令を保持し、保持した命令 I C D を演算実行部 4 に出力する。

【0016】

命令生成部 5 は、命令転送部 3 で転送される命令 I C D が複数の個別命令 I C D - B、I C D - C を組み合わせた複合命令 I C D - B Cであることを検出した場合、複合命令 I

50

C D - B Cに含まれる個別命令 I C D - Cを生成する。個別命令 I C D - Cは、複合命令 I C D - B Cに含まれる個別命令 I C D - B、I C D - Cのうち、演算実行部 4 による実行順が 2 番目の命令である。命令生成部 5 は、生成した個別命令 I C D - Cを命令転送部 3 に出力する。例えば、命令生成部 5 は、ステージ P Tで転送される複合命令 I C D - B Cを検出した場合、複合命令 I C D - B Cに含まれる個別命令 I C D - Cを生成し、生成した個別命令 I C D - Cをステージ P に出力する。なお、デコード部 1 は、複合命令 I N Sを複数の個別命令 I C B - B、I C B - Cに分解する機能を持たないため、複合命令 I N Sを受信した場合、複合命令 I N Sを示す命令コード I C Dを生成して実行制御部 2 に出力する。

【 0 0 1 7 】

10

演算実行部 4 は、命令転送部 3 から転送される命令 I C Dが示す演算を実行し、演算結果 R S L Tを出力する。但し、演算実行部 4 は、複合命令 I C D - B Cを受信した場合、複合命令 I C D - B Cに含まれる複数の個別命令 I C D - B、I C D - Cのうち実行順が最初の個別命令 I C D - Bを実行する。なお、図 1 では、演算結果 R S L Tを格納するレジスタと、演算結果 R S L Tをレジスタに格納する前に演算実行部 4 内の演算器にパイパスする経路とは省略される。

【 0 0 1 8 】

抑止制御部 6 は、命令転送部 3 のステージ P Tで転送される命令 I C Dが複合命令 I C D - B Cであることを検出した場合、実行制御部 2 に命令の出力を抑止させる抑止情報 I N Hを、実行制御部 2 に出力する。すなわち、抑止制御部 6 は、命令生成部 5 が複合命令 I C D - B Cに含まれる個別命令 I C D - Cを命令転送部 3 に出力する場合、抑止情報 I N Hを出力する。

20

【 0 0 1 9 】

なお、命令生成部 5 は、ステージ P における命令 I C Dが複合命令 I C D - B Cであることを検出した場合、複合命令 I C D - B Cに含まれる個別命令 I C D - Bと個別命令 I C D - Cとをステージ P に順次出力してもよい。この場合、抑止制御部 6 は、命令転送部 3 のステージ Pで転送される命令 I C Dが複合命令 I C D - B Cであることを検出した場合、抑止情報 I N Hを 2 クロックサイクルにわたり出力する。また、命令転送部 3 のステージ P は、抑止情報 I N Hの出力中、実行制御部 2 からの命令 I C Dの受信を停止し、命令生成部 5 からの個別命令 I C D - B、I C D - Cを選択してフリップフロップ F Fに出力する選択部を有する。これにより、演算実行部 4 は、複合命令 I C D - B Cを命令 I C D - Bとして解釈することなく、受信した個別命令 I C D - B、I C D - Cの演算を順次実行することができる。

30

【 0 0 2 0 】

図 2 は、図 1 に示す演算処理装置 1 0 0 の動作の一例を示す。すなわち、図 2 は、演算処理装置 1 0 0 の制御方法の一例を示す。図 2 に示す例では、複合命令でない通常の単一の命令 I N S - A、複合命令 I N S - B Cおよび通常の単一の命令 I N S - D、I N S - E が、デコード部 1 に順次供給される（図 2 (a)）。デコード部 1 は、受信した命令 I N S - A、I N S - B C、I N S - D、I N S - Eを順次デコードする。そして、デコード部 1 は、デコードにより得られた命令 I C D - A、I C D - B C、I C D - D、I C D - E（命令コード I C Dおよびレジスタ番号等の情報）を実行制御部 2 に順次出力する（図 2 (b)）。

40

【 0 0 2 1 】

実行制御部 2 は、デコード部 1 から受信した命令 I C D - A、I C D - B C、I C D - D、I C D - Eを空いているエン트리 E N Tに保持し、エン트리 E N Tに保持した命令 I C Dを実行可能な順に出力する（図 2 (c)）。図 2 に示す例では、説明を分かりやすくするため、実行制御部 2 は、エン트리 E N Tに保持した順に命令 I C Dを出力する。実行制御部 2 は、複合命令 I C D - B Cを分解した個別命令 I C D - B、I C D - Cではなく、複合命令 I C D - B Cをエン트리 E N Tに保持する。このため、デコード部 1 で分解された個別命令 I C D - B、I C D - Cを実行制御部 2 に保持する場合に比べて、エントリ

50

ENTの使用効率を向上することができ、アウトオブオーダの効率を向上することができ、演算処理装置100Aの性能を向上することができる。また、エントリENTの使用効率を向上できるため、エントリENTの数を抑制することができ、実行制御部2の回路規模の増大を抑制することができる。

【0022】

命令転送部3は、実行制御部2から受信する命令ICD-A、ICD-BCを、ステージP、PTに順次転送し、ステージPTは、命令ICD-A、ICD-BCを演算実行部4に出力する(図2(d))。命令生成部5は、命令転送部3のステージPTで転送される複合命令ICD-BCを検出し、複合命令ICD-BCに含まれる実行順が2番目の個別命令ICD-Cを生成し、生成した個別命令ICD-Cを命令転送部3のステージPに出力する(図2(e))。

10

【0023】

抑止制御部6は、命令転送部3のステージPTで転送される命令ICDが複合命令ICD-BCであることを検出した場合、抑止情報INHを実行制御部2に出力する(図2(f))。実行制御部2は、抑止情報INHの受信に基づいて、命令ICDの出力を停止する(図2(g))。図2に示す例では、実行制御部2は、命令ICD-Dを出力中に抑止情報INHを受信し、命令ICD-Dの出力を停止する。抑止制御部6により抑止情報INHを出力することで、命令生成部5が出力する命令と実行制御部2が出力する命令とが衝突することを抑止することができ、演算処理装置100の誤動作を抑止することができる。

20

【0024】

命令転送部3のステージPは、実行制御部2からの命令ICD-Dの代わりに、命令転送部3からの個別命令ICD-Cを受信し、受信した個別命令ICD-CをステージPTに転送する(図2(h))。そして、個別命令ICD-Cは、演算実行部4に出力される。抑止制御部6は、ステージPTに個別命令ICD-Cが転送され、ステージPTから複合命令ICD-BCが追い出されたことに基づいて、抑止情報INHの出力を停止する。これにより、実行制御部2は、出力を停止していた命令ICD-D、ICD-Eを順次出力する(図2(i))。

【0025】

演算実行部4は、命令ICD-Aを受信し、演算を実行し、演算結果RSLT-Aを出力する(図2(j))。演算実行部4は、複合命令ICD-BCを受信した場合、複合命令ICD-BCに含まれる複数の個別命令ICD-B、ICD-Cのうち実行順が最初の個別命令ICD-Bを実行する(図2(k))。例えば、複合命令ICD-BCの命令コードと個別命令ICD-Bの命令コードとは、ビット列の一部が共通である。演算実行部4は、複合命令ICD-BCの命令コードに含まれる共通のビット列に基づいて、復号命令ICD-BCを個別命令ICD-Bと解釈して実行する。これにより、演算実行部4は、複合命令ICD-BCの受信に基づいて、複合命令ICD-BCに含まれる実行順が最初の個別命令ICD-Bを実行することができる。換言すれば、命令生成部5により個別命令ICD-Bを生成することなく、個別命令ICD-Bを実行することができる。なお、上述したように、命令生成部5により、複合命令ICD-BCに含まれる個別命令ICD-Bと個別命令ICD-CとをステージPに順次生成してもよい。

30

40

【0026】

この後、演算実行部4は、命令ICD-C、ICD-D、ICD-Eを順次に受信し、それぞれの演算を実行し、演算結果RSLT-C、RSLT-D、RSLT-Eを出力する(図2(l))。このように、複合命令ICD-BCは、2つの個別命令ICD-B、ICD-Cに分解されて実行される。

【0027】

以上、図1に示す実施形態では、デコード部1で分解された個別命令ICD-B、ICD-Cを実行制御部2に保持する場合に比べて、エントリENTの使用効率を向上することができ、アウトオブオーダの効率を向上することができる。この結果、デコード部1で

50

分解された個別命令 I C D - B、I C D - C を実行制御部 2 に保持する場合に比べて、複合命令 I C D - B C を実行する演算処理装置 1 0 0 の性能を向上することができる。

【 0 0 2 8 】

抑止制御部 6 により抑止情報 I N H を出力することで、命令生成部 5 が出力する個別命令と実行制御部 2 が出力する命令とが衝突することを抑止することができ、演算処理装置 1 0 0 の誤動作を抑止することができる。演算実行部 4 は、複合命令 I C D - B C の受信に基づいて個別命令 I C D - B を実行するため、命令生成部 5 により個別命令 I C D - B を生成することなく、個別命令 I C D - B を実行することができる。

【 0 0 2 9 】

ステージ P T で検出した複合命令 I C D - B C に基づいて、2 番目に実行される個別命令 I C D - C をステージ P に出力することで、複合命令 I C D - B C と複合命令 I C D - B C に含まれる個別命令 I C D - C とを演算実行部 4 に連続して転送することができる。複合命令 I C D - B C と個別命令 I C D - C とを不可分に演算実行部 4 に転送できるため、演算実行部 4 は、複合命令 I C D - B C に含まれる個別命令 I C D - B、I C D - C を、クロックサイクルを空けることなく連続して実行できる。例えば、個別命令 I C D - B の演算結果を個別命令 I C D - C による演算で使用する場合、演算結果を演算実行部 4 に転送する複数のバイパス経路のうち、最短のバイパス経路を使用することができる。この結果、個別命令 I C D - B、I C D - C が連続して実行されず、最短のバイパス経路が使用されない場合に比べて、演算処理装置 1 0 0 の性能を向上することができる。

【 0 0 3 0 】

複合命令 I C D - B C を個別命令 I C D - B、I C D - C に分解し、演算実行部 4 で順次実行することで、複合命令 I C D - B C を実行する専用の演算器を演算実行部 4 に設けることなく、複合命令 I C D - B C を実行することができる。これにより、複合命令 I C D - B C を実行する専用の演算器を演算実行部 4 に設ける場合に比べて、演算実行部 4 および演算処理装置 1 0 0 の回路規模を削減することができる。

【 0 0 3 1 】

これに対して、複合命令がデコード部 1 により分解され、分解された個別命令が実行制御部 2 に投入される場合、実行制御部 2 は、分解した複数の個別命令を発行する間に他の命令を発行する場合がある。分解した個別命令間に他の命令が挿入され、かつ、分解した個別命令間にデータの依存性がある場合、分解した個別命令間でのデータが受け渡しは、レジスタ等を介して実行される。このため、分解した個別命令を不可分に実行する場合に比べて、レジスタ等の使用率が高くなり、アウトオブオーダー用に割り当て可能なレジスタが減るため、アウトオブオーダーの効率は低下する。さらに、デコード部 1 により分解された個別命令を不可分に実行する場合、他の命令の発行を止めて、分解した複数の個別命令を連続して発行する機能が、実行制御部 2 に追加されるため、実行制御部 2 の回路規模が増大する。換言すれば、実行制御部 2 から発行される複合命令を分解する機能を命令転送部 3 に対応して設けることで、デコード部 1 により複合命令を分解する場合に比べて、実行制御部 2 の回路規模の増大を抑止することができる。

【 0 0 3 2 】

図 3 は、演算処理装置および演算処理装置の制御方法の別の実施形態を示す。図 1 に示す実施形態で説明した要素と同一または同様の要素については、同一の符号を付し、これ等については、詳細な説明は省略する。

【 0 0 3 3 】

図 3 に示す演算処理装置 1 0 0 A は、命令キャッシュ 1 0、命令バッファ 1 2、デコード部 1 4、レジスタ管理部 1 6、リザベーションステーション部 1 8、バイパス制御部 2 0、演算実行部 2 2 およびデータキャッシュ 2 4 を有する。演算処理装置 1 0 0 A は、C P U 等のプロセッサであり、図 2 では、プロセッサに搭載されるプロセッサコアの一部が示される。なお、演算処理装置 1 0 0 A は、命令 I N S を複数のステージに分けて実行する命令パイプラインを有する。命令パイプラインの例は、図 5 および図 6 に示される。

【 0 0 3 4 】

命令キャッシュ 10 は、二次キャッシュまたはメインメモリ等から転送される命令 I N S を格納する一次命令キャッシュである。命令バッファ 12 は、命令キャッシュ 10 から転送される命令 I N S を順次保持し、保持した命令 I N S をデコード部 14 に順次出力する。

【 0 0 3 5 】

デコード部 14 は、命令バッファ 12 から転送される命令 I N S を解読し、解読した命令 I N S に含まれる命令コード I C D、レジスタ番号 G P R N およびバリッドフラグ V をリザベーションステーション部 18 に投入する。レジスタ番号 G R P N は、オペランドデータの格納用のレジスタ G P R を示す。デコード部 14 は、固定小数点の演算命令をリザベーションステーション R S E (Reservation Station for Execution) に投入する。デコード部 14 は、メモリアクセス命令 (ロード命令またはストア命令) をリザベーションステーション R S A (Reservation Station for Address) に投入する。

【 0 0 3 6 】

さらに、デコード部 14 は、解読した命令 I N S が複数の演算命令を組み合わせた複合命令の場合、複合命令の命令コード I C D とともにセカンドフローフラグ 2 F L の値をリザベーションステーション R S E に投入する。複合命令が、加算命令 a d d と減算命令 s u b とを組み合わせた加減算命令 a d d s u b の場合、加減算命令 a d d s u b を示す命令コード I C D に基づいて、加算命令 a d d が固定小数点演算器 28 により実行される。そして、固定小数点演算器 28 は、図 6 に示す命令コード操作部 I C D O P が加減算命令 a d d s u b に基づいて生成する減算命令 s u b を、例えば、加算命令 a d d の演算結果を用いて実行する。このように、複合命令は、固定小数点演算器 28 を複数回使用する命令である。加減算命令 a d d s u b に含まれる加算命令 a d d および減算命令 s u b は、個別命令の一例である。

【 0 0 3 7 】

デコード部 14 は、複合命令を複数の演算命令に分解することなくリザベーションステーション R S E に投入する。これにより、デコード部 14 で分解した複数の演算命令をリザベーションステーション R S E に投入する場合に比べて、リザベーションステーション R S E の使用効率を向上することができる。すなわち、デコード部 14 で分解した複数の演算命令をリザベーションステーション R S E に投入する場合に比べて、アウトオブオーダーの効率を向上することができ、演算処理装置 100 A の性能を向上することができる。また、リザベーションステーション R S E の使用効率を向上するため、リザベーションステーション R S E の回路規模の増大を抑制することができる。

【 0 0 3 8 】

レジスタ管理部 16 は、命令 I N S の実行により得られる演算結果等を格納するレジスタ G P R と、演算結果等を一時的に格納するレジスタ G U B (GPR Update Buffer) との対応関係を保持するテーブル G P R T (GPR Pending Register Table) を有する。レジスタ管理部 16 は、デコード部 14 からのレジスタ番号 G P R N に基づいてテーブル G P R T を参照し、未使用のレジスタ G U B のいずれかをレジスタ G P R に対応して割り当てる。

【 0 0 3 9 】

そして、レジスタ管理部 16 は、割り当てたレジスタ G U B とレジスタ G P R との対応関係を示す情報をテーブル G P R T に記録し、割り当てたレジスタ G U B を示すレジスタアドレス U B A (Update Buffer Address) を出力する。テーブル G P R T に記録されるレジスタ G U B とレジスタ G P R との対応関係を示す情報は、命令 I N S の実行が完了したことに基いて破棄される。テーブル G P R T の例は、図 4 に示される。なお、演算実行部 22 が浮動小数点演算器と、浮動小数点演算に使用するレジスタを有する場合、レジスタ管理部 16 は、浮動小数点演算に使用するレジスタに対応してテーブル G P R T と同様のテーブルを有する。

【 0 0 4 0 】

リザベーションステーション部 18 は、固定小数点の演算命令を保持するリザベーションステーション R S E と、ロード命令およびストア命令を保持するリザベーションステー

10

20

30

40

50

ションRSAとを有する。なお、演算実行部22が浮動小数点演算器を有する場合、リザベーションステーション部18は、浮動小数点の演算命令が保持されるリザベーションステーションを有する。リザベーションステーションRSEは、デコード部14から出力される固定小数点データの演算命令用の情報を保持する複数のエントリ（例えば、64エントリ）を有する。リザベーションステーションRSEは、デコード部14がデコードした演算命令を保持し、保持した命令を実行可能な順に出力することで、アウトオブオーダー実行を制御する実行制御部の一例である。リザベーションステーションRSAは、デコード部14から出力されるメモリアクセス命令（ストア命令またはロード命令）用の情報を保持する複数のエントリ（例えば、64エントリ）を有する。

【0041】

例えば、リザベーションステーションRSE、RSAは、デコード部14から出力されるバリッドフラグVの値、命令コードICDおよびレジスタ番号GPRNと、レジスタ管理部16から出力されるレジスタアドレスUBAとをエントリに保持する。さらに、リザベーションステーションRSEは、セカンドフローフラグ2FLの値をエントリに保持する。

【0042】

リザベーションステーションRSEは、レジスタ管理部16から出力されるレジスタアドレスUBAおよびバイパス制御部20から出力されるバイパスイネーブル信号BPENに基づいて、エントリに保持した演算命令の依存関係を判定する。そして、リザベーションステーションRSEは、エントリに保持した演算命令の中から実行可能な演算命令を選択し、選択した演算命令の情報を演算実行部22に投入する。演算実行部22に投入される情報は、命令コードICD、バリッドフラグVの値、レジスタアドレスRADおよびセカンドフローフラグ2FLの値を含む。レジスタアドレスRADは、データを読み出すレジスタGPRを示すアドレスと、データを読み出すレジスタGUBを示すアドレスと、データを書き込むレジスタGUBを示すアドレスとを含む。以下の説明では、バリッドフラグVは、バリッド信号Vとも称され、セカンドフローフラグ2FLは、セカンドフロー信号2FLとも称される。

【0043】

同様に、リザベーションステーションRSAは、レジスタ管理部16から出力されるレジスタアドレスUBAおよびバイパス制御部20から出力されるバイパスイネーブル信号BPENに基づいて、エントリに保持したメモリアクセス命令の依存関係を判定する。そして、リザベーションステーションRSAは、エントリに保持したメモリアクセス命令の中から実行可能なロード命令またはストア命令を選択し、選択したストア命令の情報を演算実行部22に投入する。演算実行部22に投入される情報は、命令コードICD、バリッド信号VおよびレジスタアドレスRADを含む。

【0044】

バイパス制御部20は、バイパス管理テーブルBPTBLを有する。バイパス管理テーブルBPTBLは、各レジスタGUBに対応するバイパスフラグを有する。バイパス制御部20は、リザベーションステーションRSE（またはRSA）から出力されるレジスタ番号GUBNに対応するバイパスフラグをセットする。そして、バイパス制御部20は、セットしたバイパスフラグに対応するバイパスイネーブル信号BPENを、レジスタ番号GUBNを出力したリザベーションステーションRSE（またはRSA）に出力する。リザベーションステーションRSE（またはRSA）は、バイパスイネーブル信号BPENの受信に基づいて、先行の命令の実行により得られるデータを後続の命令にバイパスさせることが可能であると判断し、後続の命令を発行する。

【0045】

演算実行部22は、固定小数点演算器28、アドレス計算演算器30およびレジスタ部32を有する。なお、演算実行部22は、論理演算器を有してもよい。固定小数点演算器28は、加算または減算を実行する加算器ADDと、乗算または除算を実行する乗算器MULとを有する。加算器ADDおよび乗算器MULは、リザベーションステーションRS

10

20

30

40

50

E から投入される演算命令に基づいて、レジスタ G P R に保持されたデータ D A T を用いて演算を実行し、演算結果 R S L T を次のクロックサイクルでレジスタ G U B に格納する。レジスタ G U B に格納された演算結果 R S L T は、演算命令の完了時にレジスタ G P R に格納される。

【 0 0 4 6 】

アドレス計算演算器 3 0 は、リザベーションステーション R S A から投入されるメモリアクセス命令に基づいて、レジスタ G P R に保持されたデータ D A T を用いてアクセスアドレスを計算し、計算したアクセスアドレスをデータキャッシュ 2 4 に出力する。アドレス計算演算器 3 0 は、ロード命令では、データキャッシュ 2 4 から出力されるデータ（ロードデータ）をレジスタ G U B に格納する。アドレス計算演算器 3 0 は、ストア命令では、レジスタ G P R から出力されるデータ（ストアデータ）を、アクセスアドレスとともにデータキャッシュ 2 4 に出力する。データキャッシュ 2 4 は、二次キャッシュまたはメインメモリ等に保持されるデータを一時的に保持する一次データキャッシュである。

【 0 0 4 7 】

レジスタ部 3 2 は、命令 I N S で指定される複数のレジスタ G P R（General-Purpose Register）と、演算結果等を一時的に格納する複数のレジスタ G U B（GPR Update Buffer）とを有する。各レジスタ G P R、G U B には、固定小数点データが格納される。例えば、レジスタ G P R およびレジスタ G U B の数は、それぞれ 6 4 個であり、レジスタ G U B を識別するレジスタアドレス U B A は、6 ビットである。

【 0 0 4 8 】

図 4 は、図 3 に示すレジスタ管理部 1 6 およびレジスタ部 3 2 の一例を示す。レジスタ管理部 1 6 のテーブル G P R T は、6 4 個のレジスタ G P R にそれぞれ対応する 6 4 個のエントリ G P（G P 0 - G P 6 3）を有する。各エントリ G P は、フラグ G P R - V およびレジスタアドレス U B A を格納する領域を有する。各フラグ G P R - V は、レジスタ G P R へのレジスタ G U B の割り当て時に " 0 " に設定され、レジスタ G P R を指定した命令の実行の完了時に " 1 " に設定される。レジスタ G P R へのレジスタ G U B の割り当ては、テーブル G P R T にレジスタアドレス U B A を格納することにより実行される。

【 0 0 4 9 】

図 4 に示す例では、レジスタ G P R 0 に対応するエントリ G P 0 に 2 進数で " 0 0 1 1 " のレジスタアドレス U B A が格納され、レジスタアドレス U B A = " 0 0 1 1 " に対応するレジスタ G U B 3 がレジスタ G P R 0 に割り当てられる。レジスタ G P R 2 に対応するエントリ G P 2 に 2 進数で " 0 0 0 1 " のレジスタアドレス U B A が格納され、レジスタアドレス U B A = " 0 0 0 1 " に対応するレジスタ G U B 1 がレジスタ G P R 2 に割り当てられる。レジスタ G P R 3 に対応するエントリ G P 3 に 2 進数で " 0 0 1 0 " のレジスタアドレス U B A が格納され、レジスタアドレス U B A = " 0 0 1 0 " に対応するレジスタ G U B 2 がレジスタ G P R 3 に割り当てられる。

【 0 0 5 0 】

レジスタ管理部 1 6 は、レジスタアドレス U B A をテーブル G P R T に格納したクロックサイクルで、格納したレジスタアドレス U B A をリザベーションステーション R S E（または R S A）に出力する。各リザベーションステーション R S E、R S A は、デコード部 1 4 から出力されるレジスタ番号 G P R N と、テーブル G P R T から出力されるレジスタアドレス U B A とに基づいて、図 4 に一点鎖線で示すレジスタ G P R、G U B の対応付けを検出する。

【 0 0 5 1 】

図 5 は、図 3 に示す演算処理装置 1 0 0 A における固定小数点演算器 2 8 に対応する命令パイプラインの概要を示す。命令パイプラインは、命令が順次転送される複数のステージ D、D T、P、P T、B 1、B 2、X を有する。リザベーションステーション R S E は、ステージ D T とステージ P の間に設けられ、フリップフロップ F F の機能を有する。ステージ X は、固定小数点演算器 2 8 を含む。ステージ P、P T は、リザベーションステーション R S E が出力する命令を順次転送する命令転送部の一例である。

【 0 0 5 2 】

固定小数点の演算命令を実行する命令パイプラインのステージは以下の通りである。なお、以下では、命令パイプラインの各ステージは、サイクルとも称される。

(a) ステージ D (Decode) : デコード部 1 4 が命令をデコードし、デコードした命令コード I C D およびレジスタ番号 G P R N 等を入力する。また、デコード部 1 4 からレジスタ番号 G P R N を受信したレジスタ管理部 1 6 が、レジスタ G P R にレジスタ G U B を割り当て、割り当てたレジスタ G U B を示すレジスタアドレス U B A を出力する。

(b) ステージ D T (Decode Transfer) : D サイクルの実行により得られた命令コード I C D、レジスタ番号 G P R N およびレジスタアドレス U B A 等をリザベーションステーション R S E に格納する。リザベーションステーション R S E は、空いているエントリ E N T (E N T 0、E N T 1、E N T 2、...) をセクタ S E L 0 により選択し、選択したエントリ E N T に、受信した命令コード I C D、レジスタ番号 G P R N およびレジスタアドレス U B A 等を格納する。以下の説明では、命令コード I C D、レジスタ番号 G P R N およびレジスタアドレス U B A 等は、命令とも称される。

(c) ステージ P (Priority) : リザベーションステーション R S E から演算実行部 2 2 に投入する命令を決定する。リザベーションステーション R S E は、エントリ E N T に保持された命令の中から実行可能な命令をセクタ S E L 1 により選択し、選択した命令を出力する。

(d) ステージ P T (Priority Transfer) : P サイクルで投入を決定した命令を演算実行部 2 2 に投入する。

(e) ステージ B 1、B 2 (Buffer) : 演算の実行に必要なデータを決定し、レジスタ G P R からデータを読み出す。

(f) ステージ X (Execute) : 固定小数点演算器 2 8 で命令を実行し、演算結果を得る。

【 0 0 5 3 】

図 6 は、図 5 に示す命令パイプラインにおいて、ステージ P からステージ X までに関連する回路要素の一例を示す。図 5 に示すフリップフロップ F F は、図 6 においても斜線の矩形で示される。リザベーションステーション R S E のエントリ E N T およびレジスタ G P R、G U B は、ラッチの機能を有するため、フリップフロップ F F と同様に斜線の矩形で示される。なお、レジスタ G U B からレジスタ G P R にオペランドデータを転送する経路は、破線の矢印で示される。また、各ステージ P、P T、B 1、B 2、X 内で伝送される信号は、各信号の前にステージ P、P T、B 1、B 2、X を示す符号 " P - "、" P T - "、" B 1 - "、" B 2 - "、" X - " が付けられる。

【 0 0 5 4 】

命令パイプラインは、命令コード操作部 I C D O P と、ステージ P に設けられるリザベーションステーション R S E のセクタ S E L 1、オアゲート O R 1 およびセクタ S E L 2、S E L 3 とを有する。なお、命令コード操作部 I C D O P は、ステージ P 内に設けられてもよい。また、命令パイプラインは、ステージ P T に設けられるアンドゲート A N D 1、セクタ S E L 4 および比較器 C M P 1、C M P 2、C M P 3 を有する。さらに、命令パイプラインは、ステージ B 1 に設けられるセクタ S E L 5、S E L 6、S E L 7 と、ステージ B 2 に設けられるセクタ S E L 8 と、ステージ X に設けられる固定小数点演算器 2 8 とを有する。

【 0 0 5 5 】

ステージ P において、リザベーションステーション R S E のセクタ S E L 1 は、リザベーションステーション R S E のエントリ E N T に保持された実行可能な命令のいずれかを選択し、選択した命令を出力する。セクタ S E L 1 から出力される命令は、バリッド信号 P - V、セカンドフロー信号 P - 2 F L、命令コード P - I C D およびレジスタアドレス P - R A D を含む。

【 0 0 5 6 】

バリッド信号 P - V 等のバリッド信号 V は、論理 1 の場合、命令 (セカンドフロー信号 2 F L、命令コード I C D およびレジスタアドレス R A D) が有効であることを示す。バ

10

20

30

40

50

リッド信号 V は、命令が有効であることを示す有効情報の一例である。論理 1 のセカンドフロー信号 $PT - 2FL$ は、リザベーションステーション RSE から複合命令が出力されたことを示し、論理 0 のセカンドフロー信号 $PT - 2FL$ は、リザベーションステーション RSE から複合命令以外の命令が出力されたことを示す。図 3 に示すデコード部 14 によりセカンドフロー信号 $2FL$ を生成し、リザベーションステーション RSE から出力することで、 P サイクルまたは PT サイクルにおいて複合命令を検出する回路を設けることなく、複合命令を検出することができる。セカンドフロー信号 $2FL$ は、複数の命令を組み合わせた複合命令を識別する識別情報の一例である。レジスタアドレス $P - RAD$ は、レジスタ GPR を示すレジスタ番号 $GPRN$ 、オペランドデータを書き込むレジスタ GUB を示すレジスタアドレス $WUBA$ およびオペランドデータを読み出すレジスタ GUB を示すレジスタアドレス $RUBA$ を含む。

【0057】

アンドゲート $AND1$ は、バリッド信号 $PT - V$ とセカンドフロー信号 $PT - 2FL$ とがともに論理 1 の場合、抑止信号 INH を論理 1 (有効レベル) に設定する。すなわち、アンドゲート $AND1$ は、リザベーションステーション RSE のセクタ $SEL1$ から複合命令が出力された場合、複合命令の PT サイクルで論理 1 の抑止信号 INH を出力する。アンドゲート $AND1$ は、バリッド信号 $PT - V$ とセカンドフロー信号 $PT - 2FL$ とのいずれかが論理 0 の場合、抑止信号 INH を論理 0 (無効レベル) に設定する。アンドゲート $AND1$ は、命令コード操作部 $ICDOP$ が複合命令に含まれる命令を示す命令コード $P - ICD2$ を出力する場合、リザベーションステーション RSE に命令の出力を抑止させる抑止信号 INH を出力する抑止制御部の一例である。抑止信号 INH は、抑止情報の一例である。

【0058】

命令コード操作部 $ICDOP$ は、ステージ PT の命令コード $PT - ICD$ が複合命令を示す場合、2 番目に実行される命令を示す命令コード $P - ICD2$ と、2 番目に実行される命令に使用されるレジスタアドレス $P - RAD2$ とを生成する。命令コード操作部 $ICDOP$ は、複合命令に含まれる命令を生成し、生成した命令をステージ P に出力する命令生成部の一例である。

【0059】

複合命令が加算と減算とを順次実行する加減算命令 $addsub$ の場合、最初に行われる命令は加算命令 add であり、2 番目に実行される命令は減算命令 sub である。複合命令が 2 つの加算を順次実行する $addadd$ 命令の場合、最初に行われる命令と 2 番目に実行される命令は、加算命令 add である。

【0060】

なお、命令コード操作部 $ICDOP$ は、ステージ PT における命令コード $P - ICD$ が複合命令であることを検出した場合、複合命令に含まれる複数の命令の命令コード $PT - ICD$ をステージ P に順次出力してもよい。この場合、複合命令に含まれる命令の数に対応するクロックサイクルの期間に抑止信号 INH を生成する信号生成部が、アンドゲート $AND1$ の出力に設けられる。これにより、固定小数点演算器 28 は、複合命令を 1 番目に実行される命令として解釈することなく、受信した命令を実行することができる。

【0061】

例えば、式 (1) に示す加減算命令 $addsub$ を実行する場合、加減算命令 $addsub$ は、式 (2) に示す add 命令と式 (3) に示す sub 命令とに分解される。そして、式 (2) に示す add 命令が実行された後、式 (3) に示す sub 命令が実行される。式 (1) から式 (3) において、符号 g は、レジスタ GPR を示す。式 (1) は、 $(g1 + g2) - g3$ の演算結果を $g0$ に格納することを示す。式 (2) は、 $g1 + g2$ の演算結果を $g0$ に格納することを示す。式 (3) は、 $g0 - g3$ の演算結果を $g0$ に格納することを示す。

$addsub \quad g0, g1, g2, g3 \quad (1)$

$add \quad g0, g1, g2 \quad (2)$

10

20

30

40

50

`sub g0, g0, g3` (3)

複合命令を分解した複数の命令において、命令パイプライン内に転送される最初に行われる命令はファーストフローと称され、命令パイプライン内に転送される2番目に実行される命令は、セカンドフローと称される。

【0062】

式(3)に示すように、命令コード操作部ICDOPは、減算命令`sub`の演算結果を格納するレジスタ`g0`に対応するレジスタアドレス`WUBA`を、元の複合命令`addsub`の演算結果を格納するレジスタ`g0`に対応するレジスタアドレス`WUBA`に設定する。また、命令コード操作部ICDOPは、減算命令`sub`の被減数(`g0`)に対応するレジスタアドレス`RUBA`を、加算命令`add`のレジスタアドレス`WUBA`(`g0`)と同じ値に設定する。これにより、ファーストフローとして実行される加算命令`add`の演算結果を、セカンドフローとして実行される減算命令`sub`に最短のバイパス経路を使って引き継ぐことができる。

【0063】

オアゲートOR1は、バリッド信号`P-V`または抑止信号`INH`を、フリップフロップFFを介してバリッド信号`PT-V`として出力する。オアゲートOR1により、複合命令の`PT`サイクルの次のクロックサイクルにおけるバリッド信号`PT-V`は、バリッド信号`P-V`の論理に拘わりなく論理1(有効レベル)に設定される。これにより、リザベーションステーション`RSE`から出力されない減算命令`sub`(セカンドフロー)を命令パイプライン処理により実行することができる。オアゲートOR1は、リザベーションステーション`RSE`が出力する論理1のバリッド信号`PT-V`または論理1の抑止信号`INH`をバリッド信号`PT-V`として演算実行部22に転送する論理和部の一例である。

【0064】

セクタSEL2は、抑止信号`INH`が無効レベル(論理0)を示す場合、リザベーションステーション`RSE`からのセカンドフロー信号`P-2FL`を、フリップフロップFFを介してセカンドフロー信号`PT-2FL`として出力する。セクタSEL2は、抑止信号`INH`が有効レベル(論理1)を示す場合、論理0を、フリップフロップFFを介してセカンドフロー信号`PT-2FL`として出力する。これにより、複合命令に基づいてバリッド信号`PT-V`が2クロックサイクルの期間に論理1に設定される場合にも、アンドゲートAND1が抑止信号`INH`を論理1に設定し続けることを抑止することができる。この結果、論理1の抑止信号`INH`により誤った命令がセクタSEL2、SEL3から出力されることを抑止することができ、演算処理装置100Aの誤動作を抑止することができる。セクタSEL2は、抑止信号`INH`の論理に基づいて、リザベーションステーション`RSE`から出力されるセカンドフロー信号`PT-2FL`、または命令が複合命令でないことを示す論理0を選択する第1の選択部の一例である。

【0065】

セクタSEL3は、抑止信号`INH`が無効レベル(論理0)を示す場合、命令コード`P-ICD`およびレジスタアドレス`P-RAD`を、フリップフロップFFを介して命令コード`PT-ICD`およびレジスタアドレス`PT-RAD`として出力する。セクタSEL3は、抑止信号`INH`が有効レベル(論理1)を示す場合、命令コード`PT-ICD2`およびレジスタアドレス`PT-RAD2`を、フリップフロップFFを介して命令コード`PT-ICD`およびレジスタアドレス`PT-RAD`として出力する。すなわち、複合命令を示すセカンドフロー信号`PT-2FL`により論理1の抑止信号`INH`が生成された場合、命令コード操作部ICDOPが生成する命令がPサイクルに投入される。セクタSEL3により、リザベーションステーション`RSE`が出力する命令と、命令コード操作部ICDOPが出力する命令とが衝突することを抑止することができ、演算処理装置100Aの誤動作を抑止することができる。セクタSEL3は、抑止信号`INH`に基づいて、リザベーションステーション`RSE`から出力される命令または命令コード操作部ICDOPから出力される命令を選択する第2の選択部の一例である。

【0066】

10

20

30

40

50

なお、リザベーションステーション R S E のセクタ S E L 1 は、論理 1 の抑止信号 I N H を受けている間、命令の出力を停止し、バリッド信号 P - V を論理 0 に設定する。したがって、論理 1 の抑止信号 I N H によりセクタ S E L 3 が命令コード操作部 I C D O P の出力を選択している期間に、リザベーションステーション R S E から命令コード P - I C D およびレジスタアドレス P - R A D が出力されることを抑止することができる。これにより、命令コード P - I C D およびレジスタアドレス P - R A D がリザベーションステーション R S E から消失されることを抑止することができ、演算処理装置 1 0 0 A の誤動作を抑止することができる。

【 0 0 6 7 】

比較器 C M P 1、C M P 2、C M P 3 は、オペランドデータを選択するためのオペランド選択信号 P T - O P S E L を生成する。比較器 C M P 1 は、先行する命令の B 1 サイクルのレジスタアドレス B 1 - W U B A と、後続の命令の P T サイクルのレジスタアドレス P T - R U B A とを比較する。比較器 C M P 2 は、先行する命令の B 2 サイクルのレジスタアドレス B 2 - W U B A と、後続の命令の P T サイクルのレジスタアドレス P T - R U B A とを比較する。比較器 C M P 3 は、先行する命令の X サイクルのレジスタアドレス X - W U B A と、後続の命令の P T サイクルのレジスタアドレス P T - R U B A とを比較する。

【 0 0 6 8 】

そして、各比較器 C M P 1 - C M P 3 は、レジスタアドレス W U B A、R U B A の一致を検出した場合、先行の命令の演算結果（オペランドデータ）を、後続の命令を実行する固定小数点演算器 2 8 にバイパスさせるオペランド選択信号 P T - O P S E L を出力する。なお、各比較器 C M P 1 - C M P 3 は、レジスタアドレス P T - R U B A とともに、図 4 に示すフラグ G P R - V の値を受信し、フラグ G P R - V の値が論理 1 の場合に、上述したオペランド選択信号 P T - O P S E L を出力する。フラグ G P R - V の値が論理 1 の場合、命令の実行が完了し、レジスタ G U B 2 に保持されたデータはレジスタ G P R 3 に移動済みであり、レジスタ G U B 2 の割り当ては解除されている。この場合、比較器 C M P 1 - C M P 3 のいずれかは、セクタ S E L 6 の出力をセクタ S E L 7 に選択させるオペランド選択信号 P T - O P S E L と、セクタ S E L 7 の出力をセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L とを生成する。

【 0 0 6 9 】

X サイクルの次のクロックサイクルで演算結果はレジスタ G U B に書き込まれるため、バイパス処理は実行されない。このため、命令パイプラインは、先行の命令の X サイクル以降のレジスタアドレス W U B A と後続の命令の P T サイクルのレジスタアドレス P T - R U B A とを比較する比較器を持たない。比較器 C M P 1 - C M P 3 によるオペランドデータをバイパスさせる制御の例は、図 7 に示される。

【 0 0 7 0 】

セクタ S E L 4 は、レジスタアドレス X - W U B A を 1 クロックサイクル遅らせたレジスタアドレス X - W U B A 2 が示すレジスタ G U B のいずれかに、固定小数点演算器 2 8 による演算結果 R S L T を格納する。セクタ S E L 5 は、レジスタアドレス P T - U B A が示すレジスタ G U B のいずれかから演算に使用するオペランドデータを読み出す。なお、セクタ S E L 5 が読み出すオペランドデータの数は、命令のオペランドの数に依存する。

【 0 0 7 1 】

セクタ S E L 7 は、オペランド選択信号 B 1 - O P S E L に基づいて選択したオペランドデータを、フリップフロップ F F を介してオペランドデータ B 2 - O P としてセクタ S E L 8 に出力する。セクタ S E L 8 は、オペランド選択信号 B 2 - O P S E L に基づいて選択したオペランドデータを、フリップフロップ F F を介してオペランドデータ X - O P として固定小数点演算器 2 8 に出力する。

【 0 0 7 2 】

なお、図 6 では省略されるが、セクタ S E L 7、S E L 8 は、演算に使用する即値を

選択する機能と、固定小数点演算器 28 以外の演算器からバイパスされるデータを選択する機能を有する。また、セクタ S E L 7、S E L 8 は、ロード命令により読み出されるデータを選択する機能を有する。

【0073】

図 7 は、図 5 および図 6 に示す命令パイプラインにおいて、オペランドデータをバイパスさせるタイミングの一例を示す。クロックサイクルは、命令パイプラインを動作させるクロックのサイクルを示す。図 7 に示すタイミング (a)、(b)、(c)、(d) は、先行の命令が加算命令 "a d d g 3, 4, g 1" であり、後続の命令が加算命令 "a d d g 5, 4, g 3" である。先行の加算命令 a d d は、即値 (= 4) とレジスタ G P R 1 (g 1) が保持する値とを加算し、加算結果をレジスタ G P R 3 (g 3) に書き込む命令である。後続の加算命令 a d d は、即値 (= 4) とレジスタ G P R 3 (g 1) が保持する値とを加算し、加算結果をレジスタ G P R 5 (g 5) に書き込む命令である。

10

【0074】

先行の加算命令 a d d の加算結果が書き込まれるレジスタ G P R 3 と、後続の加算命令 a d d で加算に使用するレジスタ G P R 3 とは同じであり、データの依存性がある。先行の加算命令 a d d および後続の加算命令 a d d で使用するレジスタ G R R 3 には、共通のレジスタ G U B (例えば、G U B 2) が割り当てられる。なお、図 7 では、各加算命令 a d d において、D T サイクルと P サイクルとが連続して実行されるが、D T サイクルでリザベーションステーション R S E に投入された命令が、次のクロックサイクルでリザベーションステーション R S E から発行されるとは限らない。

20

【0075】

タイミング (a) では、図 6 に示す比較器 C M P 1 は、クロックサイクル 5 において、先行の加算命令 a d d のレジスタアドレス B 1 - W U B A (G U B 2) と後続の加算命令 a d d のレジスタアドレス P T - R U B A (G U B 2) との一致を検出する。そして、比較器 C M P 1 は、固定小数点演算器 28 の出力をセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。オペランド選択信号 P T - O P S E L は、オペランド選択信号 B 1 - O P S E L、B 2 - O P S E L として順次伝達される。セクタ S E L 8 は、後続の加算命令 a d d の B 2 サイクルにおいて、固定小数点演算器 28 の出力を選択することで、先行の加算命令 a d d の加算結果 (オペランドデータ) を後続の加算命令 a d d にバイパスさせる。これにより、先行の加算命令 a d d の実行が完了した次のクロックサイクルで、先行の加算命令 a d d の実行結果を用いて後続の加算命令 a d d を実行することができる。

30

【0076】

タイミング (b) では、図 6 に示す比較器 C M P 2 は、クロックサイクル 6 において、先行の加算命令 a d d のレジスタアドレス B 2 - W U B A (G U B 2) と、後続の加算命令 a d d のレジスタアドレス P T - R U B A (G U B 2) との一致を検出する。そして、比較器 C M P 2 は、固定小数点演算器 28 の出力をフリップフロップ F F で 1 クロックサイクル遅らせた加算結果をセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。セクタ S E L 8 は、後続の加算命令 a d d の B 2 サイクルにおいて、固定小数点演算器 28 の出力に接続されたフリップフロップ F F の出力を選択することで、先行の加算命令 a d d の加算結果を後続の加算命令 a d d にバイパスさせる。これにより、先行の加算命令 a d d の実行が完了した 2 クロックサイクル後に、先行の加算命令 a d d の実行結果を用いて後続の加算命令 a d d を実行することができる。

40

【0077】

タイミング (c) では、図 6 に示す比較器 C M P 3 は、クロックサイクル 7 において、先行の加算命令 a d d のレジスタアドレス X - W U B A (G U B 2) と、後続の加算命令 a d d のレジスタアドレス P T - R U B A (G U B 2) との一致を検出する。そして、比較器 C M P 3 は、固定小数点演算器 28 の出力をフリップフロップ F F で 1 クロックサイクル遅らせた加算結果を B 1 サイクルでセクタ S E L 7 に選択させるオペランド選択信号 P T - O P S E L を生成する。また、比較器 C M P 3 は、セクタ S E L 7 が選択した

50

加算結果を B 2 サイクルでセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。セクタ S E L 7、S E L 8 は、後続の加算命令 a d d の B 1、B 2 サイクルにおいて、固定小数点演算器 2 8 の出力に接続されたフリップフロップ F F の出力を順次選択することで、先行の加算命令 a d d の加算結果を後続の加算命令 a d d にバイパスさせる。これにより、先行の加算命令 a d d の実行が完了した 3 クロックサイクル後に、先行の加算命令 a d d の実行結果を用いて後続の加算命令 a d d を実行することができる。

【 0 0 7 8 】

タイミング (d) では、各比較器 C M P 1 - C M P 3 は、いずれのクロックサイクルにおいても先行の加算命令 a d d のレジスタアドレス P T - R U B A (G U B 2) と、後続の加算命令 a d d のレジスタアドレス W U B A との一致を検出しない。この場合、後続の加算命令 a d d の加算結果をレジスタ G U B に格納する前にバイパスさせる動作は実行されず、後続の加算命令 a d d に使用する先行の加算命令 a d d の加算結果は、レジスタ G U B から読み出される。タイミング (d) では、比較器 C M P 1 - C M P 3 のいずれかは、セクタ S E L 5 の出力 (すなわち、レジスタ G U B 2 に格納された加算結果) をセクタ S E L 7 に選択させるオペランド選択信号 P T - O P S E L を生成する。また、比較器 C M P 1 - C M P 3 のいずれかは、セクタ S E L 7 が選択したオペランドデータを B 2 サイクルでセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。そして、セクタ S E L 7、S E L 8 は、後続の加算命令 a d d の B 1、B 2 サイクルにおいて、レジスタ G U B 2 に格納された先行の加算命令 a d d の加算結果を順次選択する。

【 0 0 7 9 】

セクタ S E L 7 によるレジスタ G U B 2 の出力の選択は、テーブル G P R T (図 4) においてレジスタ G U B 2 を割り当てたレジスタ G P R 3 に対応するエントリ G P 3 のフラグ G P R - V が論理 0 であり、かつ、比較器 C M P 1 - C M P 3 のいずれの比較結果も一致しない場合に実行される。テーブル G P R T のレジスタ G P R 3 に対応するエントリ G P 3 のフラグ G P R - V が論理 1 の場合、命令の実行が完了し、レジスタ G U B 2 に保持されたデータはレジスタ G P R 3 に移動済みであり、レジスタ G U B 2 の割り当ては解除されている。この場合、比較器 C M P 1 - C M P 3 の比較結果に拘わらず、セクタ S E L 6、S E L 7、S E L 8 により、レジスタ G P R 3 に格納されたオペランドデータが選択される。

【 0 0 8 0 】

図 8 は、図 3 に示す演算処理装置 1 0 0 A の動作の一例を示す。すなわち、図 8 は、演算処理装置 1 0 0 A の制御方法の一例を示す。図 8 では、式 (1) に示す加減算命令 " a d d s u b g 0 , g 1 , g 2 , g 3 " が実行される。命令パイプラインは、加減算命令 a d d s u b を、加算命令 a d d を実行するファーストフロー 1 s t と、減算命令 s u b を実行するセカンドフロー 2 n d とに分解して処理する。図 8 では、説明を分かりやすくするために、レジスタアドレス R A D は、レジスタ G P R の番号が示され、レジスタアドレス U B A (レジスタ G U B の番号) の記載は省略される。なお、図 8 では D T サイクルと P サイクルとが連続して実行されるが、図 7 と同様に、D T サイクルでリザベーションステーション R S E に投入された命令が、次のクロックサイクルでリザベーションステーション R S E から発行されるとは限らない。

【 0 0 8 1 】

クロックサイクル 3 にリザベーションステーション R S E から加減算命令 a d d s u b が発行され、加減算命令 a d d s u b の P サイクルが実行される (図 8 (a))。バリッド信号 P - V はおよびセカンドフロー信号 P - 2 F L は、論理 1 に設定され、命令コード P - I C D は、" a d d s u b " に設定される。演算に使用するオペランドデータを読み出すレジスタ G P R を示すリード用のレジスタアドレス P - R A D (R) は、レジスタ G P R 1、G P R 2、G P R 3 を示す。演算結果を格納するレジスタ G P R を示すライト用のレジスタアドレス P - R A D (W) は、レジスタ G P R 0 を示す。

【 0 0 8 2 】

クロックサイクル 4 において、図示は省略されるが、命令コード操作部 I C D O P は、P T サイクルの加減算命令 a d d s u b に基づき、命令コード P - I C D 2 (s u b) と、レジスタアドレス P - R A D 2 (P - R A D (R)、P - R A D (W)) とを生成する。ここで、命令コード操作部 I C D O P は、リード用の P - R A D (R) を (1 , 2) から (0 , 3) に変更するが、ライト用のレジスタアドレス P - R A D (W) は " 0 " のままにする。

【 0 0 8 3 】

クロックサイクル 4 の P T サイクルでは、クロックサイクル 3 の P サイクルの情報がそのまま転送される。クロックサイクル 4 では、リザベーションステーション R S E から命令が発行されないため、バリッド信号 P - V は論理 0 に設定される (図 8 (b))。バリッド信号 P T - V およびセカンドフロー信号 P T - 2 F L は、ともに論理 1 のため、図 6 に示すアンドゲート A N D 1 は、抑止信号 I N H を論理 1 に設定する (図 8 (c))。これにより、図 6 に示すオアゲート O R 1 の出力は論理 1 に設定され、次のクロックサイクル 5 のバリッド信号 P T - V は論理 1 に設定される (図 8 (d))。リザベーションステーション R S E のセクタ S E L 1 は、論理 1 の抑止信号 I N H を受けている間、P サイクルの命令の出力を停止する。

【 0 0 8 4 】

論理 1 の抑止信号 I N H により、セクタ S E L 2 は論理 0 を選択するため、次のクロックサイクル 5 のセカンドフロー信号 P T - 2 F L および抑止信号 I N H は論理 0 に設定される (図 8 (e)、(f))。論理 0 の抑止信号 I N H により、リザベーションステーション R S E のセクタ S E L 1 は、停止していた P サイクルの命令の出力を再開する。但し、図 8 に示す例では、リザベーションステーション R S E から出力される後続の命令の記載は省略される。

【 0 0 8 5 】

図 6 に示すセクタ S E L 3 は、論理 1 の抑止信号 I N H に基づいて、命令コード操作部 I C D O P が生成した命令コード P - I C D 2 とレジスタアドレス P - R A D 2 とを選択する。これにより、セカンドフローとして実行される減算命令 s u b の情報が、クロックサイクル 5 の P T サイクルに転送され、減算命令 s u b は、P T サイクルから開始される (図 8 (g))。この後、ファーストフローとして実行される加減算命令 a d d s u b 中の加算命令 a d d およびセカンドフローとして実行される減算命令 s u b は、クロックサイクル毎に後段のステージに順次転送される。

【 0 0 8 6 】

クロックサイクル 5 において、図 6 に示す比較器 C M P 1 は、B 1 サイクルのレジスタアドレス B 1 - R A D (W) = " 0 " が、P T サイクルのレジスタアドレス P T - R A D (R) = " 0 " に一致することを確認する。そして、比較器 C M P 1 は、固定小数点演算器 2 8 の出力をセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。

【 0 0 8 7 】

クロックサイクル 7 において、固定小数点演算器 2 8 は、図示しない命令コード X - I C D (a d d s u b) と、レジスタアドレス X - R A D (R) = (1 , 2 , 3) とレジスタアドレス X - R A D (W) = " 0 " とを受ける。固定小数点演算器 2 8 の加算器 A D D は、図 1 に示す演算実行部 4 と同様に、複合命令 " a d d s u b g 0 , g 1 , g 2 , g 3 " を加算命令 " a d d g 0 , g 1 , g 2 " と解釈する機能を有する。すなわち、加算器 A D D は、レジスタアドレス X - R A D (R) = (1 , 2 , 3) の最初の 2 つ (1 , 2) を、加算に使用するレジスタ番号と解釈する。そして、加算器 A D D は、加算命令 " a d d g 0 , g 1 , g 2 " を実行し、加算結果をレジスタ G P R 0 に対応するレジスタ G U B に格納する (図 8 (h))。

【 0 0 8 8 】

クロックサイクル 5 に比較器 C M P 1 が生成したオペランド選択信号 P T - O P S E L

10

20

30

40

50

に基づいて、加算命令 `a d d` による加算結果は、クロックサイクル 8 で固定小数点演算器 28 にバイパスされる (図 8 (i))。そして、固定小数点演算器 28 は、加算命令 `a d d` による加算結果を使用して、セカンドフローとして実行される減算命令 `s u b` を実行し、リザベーションステーション `R S E` から投入された加減算命令 `a d d s u b` の実行が完了する。

【0089】

なお、複合命令は、3 以上の命令を含んでもよい。この場合、命令コード操作部 `I C D O P` は、2 番目以降に実行される命令を示す命令コード `P - I C D 2` およびレジスタアドレス `P - R A D 2` を順次に生成する。また、複合命令に含まれる 2 番目以降に実行される命令の数に対応するクロックサイクルの期間に抑止信号 `I N H` を生成する信号生成部が、

10

【0090】

また、複合命令に含まれる 1 番目に実行される命令が、演算サイクル (X サイクル) を X 1 サイクルと X 2 サイクルとにより実行する命令であるとする。この場合、命令コード操作部 `I C D O P` は、複合命令を検出した次のクロックサイクルで命令コード `P - I C D 2` とレジスタアドレス `P - R A D 2` とを P サイクルに投入する。また、命令パイプラインは、抑止信号 `I N H` を図 8 に対して 1 クロックサイクル遅れて生成する制御回路を有する。

【0091】

図 9 は、図 3 に示す演算処理装置 100A の動作の別の例を示す。すなわち、図 9 は、演算処理装置 100A の制御方法の一例を示す。図 8 と同一または同様の動作については、詳細な説明は省略する。図 9 は、図 7 に示すタイミング (a) に示す命令の実行時における命令パイプライン内の状態を示す。

20

【0092】

クロックサイクル 3 でリザベーションステーション `R S E` から加算命令 `a d d` が発行され、P サイクルが実行され、クロックサイクル 4 でリザベーションステーション `R S E` から減算命令 `s u b` が発行され、P サイクルが実行される (図 9 (a)、(b))。クロックサイクル 3 およびクロックサイクル 4 において、バリッド信号 `P - V` は論理 1 に設定される。加算命令 `a d d` および減算命令 `s u b` は、複合命令ではないため、クロックサイクル 3 およびクロックサイクル 4 において、セカンドフロー信号 `P - 2 F L` は論理 0 に設定

30

【0093】

クロックサイクル 4 の P T サイクルでは、クロックサイクル 3 の P サイクルの情報がそのまま転送され、クロックサイクル 5 の P T サイクルでは、クロックサイクル 4 の P サイクルの情報がそのまま転送される (図 9 (d)、(e))。クロックサイクル 4 およびクロックサイクル 5 において、命令コード操作部 `I C D O P` は、命令コード `P T - I C D` が複合命令でないため、命令コード `P - I C D 2` およびレジスタアドレス `P T - R A D 2` を生成しない。クロックサイクル 3 およびクロックサイクル 4 において、セカンドフロー信号 `P T - 2 F L` は論理 0 であるため、抑止信号 `I N H` は論理 0 に維持される (図 9 (f))。これにより、図 6 に示すセクタ `S E L 2` は、リザベーションステーション `R S E` からのセカンドフロー信号 `P - 2 F L` を選択し続ける。セクタ `S E L 3` は、リザベーションステーション `R S E` からの命令コード `P - I C D` およびレジスタアドレス `P - R A D` を選択し続ける。

40

【0094】

クロックサイクル 5 において、図 8 と同様に、比較器 `C M P 1` は、B 1 サイクルのレジスタアドレス `B 1 - R A D (W) = " 3 "` が、P T サイクルのレジスタアドレス `P T - R A D (R) = " 3 "` に一致すること検出する。そして、比較器 `C M P 1` は、固定小数点演算器 28 の出力をセクタ `S E L 8` に選択させるオペランド選択信号 `P T - O P S E L` を生成する。

【0095】

50

クロックサイクル7において、固定小数点演算器28は、加算命令"add g3, 4, g1"を実行し、加算結果をレジスタGPR3に対応するレジスタGUBに向けて出力する(図9(g))。クロックサイクル5に比較器CMP1が生成したオペランド選択信号PT-OPSELに基づいて、加算命令addによる加算結果は、クロックサイクル8で固定小数点演算器28にバイパスされる(図9(h))。減算命令subは、バイパスされた加算結果を用いて実行される。そして、リザベーションステーションRSEから投入された加算命令addと減算命令subの実行が完了する。

【0096】

図10は、図6に示す命令パイプラインにおけるPサイクルおよびPTサイクルでの動作フローの一例を示す。すなわち、図10は、演算処理装置100Aの制御方法の一例を示す。

10

【0097】

まず、ステップS100において、命令パイプラインは、バリッド信号PT-Vが論理1の場合、演算実行部22に対する命令が発行されたため、処理をステップS110に移行する。命令パイプラインは、バリッド信号PT-Vが論理0の場合、演算実行部22に対する命令が発行されていないため、処理を終了する。

【0098】

ステップS110において、命令パイプラインは、セカンドフロー信号PT-2FLが論理1の場合、加減算命令addsub等の複合命令が発行されたと判定し、複合命令を2つのフローに分解して実行するために、処理をステップS120に移行する。命令パイプラインは、セカンドフロー信号PT-2FLが論理0の場合、複合命令以外の通常の命令が発行されたと判定し、処理をステップS150に移行する。

20

【0099】

ステップS120において、命令パイプラインの命令コード操作部ICDOPは、複合命令を分解することで実行順が2番目の命令(図8では減算命令sub)を生成し、生成した命令を命令コードP-ICD2およびレジスタアドレスP-RAD2に設定する。次に、ステップS140において、命令パイプラインは、抑止信号INHを論理0に設定する。そして、命令パイプラインは、次のPTサイクルのバリッド信号PT-Vとセカンドフロー信号PT-2FLとを論理1、論理0にそれぞれ設定し、リザベーションステーションRSEに命令の発行を抑止させる。次に、ステップS150において、命令パイプラインは、リザベーションステーションRSEが発行した命令を実行し、処理を終了する。

30

【0100】

以上、図3から図10に示す実施形態においても、図1および図2に示す実施形態と同様の効果を得ることができる。例えば、複合命令を複数の個別命令に分解してリザベーションステーションRSEに投入する場合に比べて、エントリENTの使用効率を向上することができる。この結果、複合命令を分解した複数の個別命令をリザベーションステーションRSEに投入する場合に比べて、複合命令を実行する演算処理装置100Aの性能を向上することができる。加算器ADDは、複合命令addsubを加算命令addと解釈するため、最初に実行する加算命令addを命令コード操作部ICDOP等により生成することなく、加算命令addを実行することができる。

40

【0101】

複合命令を複数の個別命令に分解することで、複合命令を実行する専用の演算器を演算実行部22に設けることなく、複合命令を実行することができる。これにより、複合命令を実行する専用の演算器を演算実行部22に設ける場合に比べて、演算実行部22および演算処理装置100Aの回路規模を削減することができる。また、複合命令を分解した複数の個別命令をリザベーションステーションRSEに保持しないため、複合命令に含まれる複数の個別命令を不可分で実行するための制御機能は、リザベーションステーションRSEに搭載されない。これにより、リザベーションステーションRSEの回路規模の増大を抑止することができる。

50

【0102】

P Tサイクルで検出した複合命令に基づいて、2番目に行われる個別命令をステージPに出力することで、複合命令に含まれる複数の個別命令を連続して実行することができる。このため、分解した個別命令間にデータの依存性がある場合にも、バイパス経路を使用してデータを受け渡すことができる。したがって、分解した個別命令間に他の命令が挿入される場合に比べて、先行の個別命令から後続の個別命令の受け渡すデータがレジスタGUBに保持される期間を短縮することでレジスタGUBの使用率を下げることができ、アウトオブオーダーの効率を向上することができる。

【0103】

さらに、図3から図10に示す実施形態では、以下に示す効果を得ることができる。すなわち、図3に示すデコード部14によりセカンドフロー信号2FLを生成し、リザベーションステーションRSEから出力することで、PサイクルまたはPTサイクルにおいて複合命令を検出する回路を設けることなく、複合命令を検出することができる。複合命令に基づいてバリッド信号PT-Vが2クロックサイクルの期間に論理1に設定される場合にも、セクタSEL2により、アンドゲートAND1が抑止信号INHを論理1に設定し続けることを抑止することができる。この結果、誤った命令がセクタSEL2、SEL3から出力されることを抑止することができ、演算処理装置100Aの誤動作を抑止することができる。

【0104】

セクタSEL3により、リザベーションステーションRSEが出力する命令と、命令コード操作部ICDOPが出力する命令とが衝突することを抑止することができ、演算処理装置100Aの誤動作を抑止することができる。オアゲートOR1により、論理1の抑止信号INHをバリッド信号PT-Vとして転送することで、命令コード操作部ICDOPが出力する命令コードP-ICD2に対応するバリッド信号PT-Vを生成することができる。これにより、リザベーションステーションRSEから出力されない2番目に行われる個別命令をパイプライン処理により実行することができる。

【0105】

なお、図3から図11に示す実施形態は、固定小数点演算器28が実行する複合命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用する例について説明される。しかしながら、図3から図10に示す実施形態は、固定小数点演算器28と論理演算器とにより実行される複合命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用されてもよい。また、図3から図11に示す実施形態は、浮動小数点演算器が実行する複合命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用されてもよい。

【0106】

図11は、演算処理装置および演算処理装置の制御方法の別の実施形態において、リザベーションステーションRSEから固定小数点演算器28までの命令パイプラインの一例を示す。図3から図6と同一または同様の要素については、同じ符号を付し、詳細な説明は省略する。

【0107】

図11に示す演算処理装置100Bの命令パイプラインは、オアゲートOR2、インバータIV1、アンドゲートAND2および命令依存チェック部ICDCCKが追加されたことを除き、図6に示す構成と同様である。オアゲートOR2、インバータIV1およびアンドゲートAND2は、ステージPTに設けられる。命令依存チェック部ICDCCKは、ステージPTの外に設けられるが、ステージPT内に設けられてもよい。演算処理装置100Bのその他の構成は、図3から図5に示す構成と同様である。

【0108】

なお、命令コード操作部ICDOPは、図6で説明した機能に加えて、命令コードPT-ICDおよびレジスタアドレスPT-RADを、命令コードP-ICD2およびレジスタアドレスP-RAD2として出力する機能を有する。すなわち、命令コード操作部IC

10

20

30

40

50

D O P は、リザベーションステーション R S E が出力する命令が複合命令でない場合、ステージ P T で転送された命令をステージ P に出力する機能を有する。

【 0 1 0 9 】

ところで、図 3 に示す加算器 A D D から乗算器 M U L へのデータの転送時間は、加算器 A D D と乗算器 M U L との物理的な距離の増加に依存して長くなる。ここで、物理的な距離は、加算器 A D D の出力から乗算器 M U L の入力まで配線長に対応する。この実施形態では、加算器 A D D と乗算器 M U L との物理的な距離が、所定の制限値を超える。例えば、加算器 A D D での加算結果を、固定小数点演算器 2 8 内の加算器 A D D の出力からセレクト S E L およびフリップフロップ F F を介して固定小数点演算器 2 8 内の乗算器 M U L にバイパスするために掛かる時間は、1 クロックサイクルを超える。このため、加算命令 a d d と乗算命令 m u l とが連続して実行され、かつ、データの依存性がある場合、乗算器 M U L による乗算命令 m u l の演算を 1 クロックサイクル後ろにずらす制御が命令パイプラインで実行される。固定小数点演算器 2 8 の出力からセレクト S E L およびフリップフロップ F F を介して固定小数点演算器 2 8 にバイパス経路は、最短のバイパス経路である。最短のバイパス経路以外のバイパス経路では、乗算器 M U L による乗算命令 m u l の演算を 1 クロックサイクル後ろにずらす制御を実行することなく、加算結果は正常にバイパスされる。

10

【 0 1 1 0 】

命令依存チェック部 I C D C K は、例えば、P T サイクルの命令と B 1 サイクルの命令との依存性のチェックを実行し、所定の依存関係を検出した場合、依存検出信号 D E T を論理 1 に設定する。所定の依存関係は、先行の加算命令 a d d の B 1 サイクルと後続の乗算命令 m u l の P T サイクルとが同じクロックサイクルで実行され、かつ、加算命令 a d d の演算結果を乗算命令 m u l で使用する場合（データの依存性がある場合）に検出される。上記の所定の依存関係を有する一对の命令は、特定命令と称される。換言すれば、特定命令は、先行命令を実行する演算実行部 2 2 内の演算器が演算結果を出力してから、後続命令を実行する演算実行部 2 2 内の他の演算器に演算結果がバイパスされるまでの時間が、1 クロックサイクルを超える組み合わせの命令である。命令依存チェック部 I C D C K は、リザベーションステーション R S E が順次出力する先行命令および後続命令が所定の組み合わせ（a d d と m u l ）であり、かつ、先行命令の演算結果を後続命令で使用する特定命令であることを検出する依存検出部の一例である。

20

30

【 0 1 1 1 】

オアゲート O R 2 は、セカンドフロー信号 P T - 2 F L または依存検出信号 D E T が論理 1 の場合、論理 1 をアンドゲート A N D 1 に出力する。すなわち、アンドゲート A N D 1 は、バリッド信号 P T - V が論理 1 でリザベーションステーション R S E から複合命令が出力された場合、または命令依存チェック部 I C D C K が特定命令を検出した場合、論理 1 の抑止信号 I N H を出力する。オアゲート O R 2 およびアンドゲート A N D 1 は、命令コード操作部 I C D O P が複合命令に含まれる命令を出力する場合、または命令依存チェック部 I C D C K が特定命令を検出した場合、抑止信号 I N H を出力する抑止制御部の一例である。

【 0 1 1 2 】

40

セレクト S E L 2 は、図 6 と同様に、抑止信号 I N H が論理 0 の場合、リザベーションステーション R S E から出力されるセカンドフロー信号 P - 2 F L を選択し、抑止信号 I N H が論理 1 の場合、論理 0 を選択する。セレクト S E L 3 は、図 6 と同様に、抑止信号 I N H が論理 0 の場合、リザベーションステーション R S E から出力される命令を選択し、抑止信号 I N H が論理 1 の場合、命令コード操作部 I C D O P から出力される命令を選択する。但し、論理 1 の抑止信号 I N H は、命令コード操作部 I C D O P が複合命令に含まれる個別命令を出力する場合、または命令依存チェック部 I C D C K が特定命令を検出した場合に出力される。

【 0 1 1 3 】

インバータ I V 1 は、依存検出信号 D E T の論理を反転し、反転した論理をアンドゲート

50

トAND2に出力する。アンドゲートAND2は、依存検出信号DETが論理0の場合、バリッド信号PT-Vを、フリップフロップFFを介してステージB1に転送し、依存検出信号DETが論理1の場合、バリッド信号PT-VのステージB1への転送を遮断する。すなわち、命令依存チェック部ICDCにより特定命令が検出された次のクロックサイクルのバリッド信号B1-Vは、論理0に設定される。インバータIV1およびアンドゲートAND2は、命令依存チェック部ICDCが特定命令を検出した場合、リザベーションステーションRSEが出力した後続命令に含まれるバリッド信号B1-Vを無効状態に設定する無効設定部の一例である。

【0114】

図12は、図11に示す命令パイプラインを有する演算処理装置100Bの動作の一例を示す。すなわち、図12は、図11に示す命令パイプラインを有する演算処理装置100Bの制御方法の一例を示す。図8と同一または同様の動作については、詳細な説明は省略する。図12では、加算命令"add g0, g1, g2"と乗算命令"mul g2, g0, 3"とが、リザベーションステーションRSEから命令パイプラインのPサイクルに連続して投入される。なお、リザベーションステーションRSEから命令パイプラインに複合命令が投入される場合の動作は、図8と同じである。また、図7および図9に示す動作は、図11に示す演算処理装置100Bでも実行される。

【0115】

加算命令addは、レジスタGPR1(g1)が保持する値とレジスタGPR2(g2)が保持する値とを加算し、加算結果をレジスタGPR0(g0)に書き込む命令である。乗算命令mulは、レジスタGPR0(g0)が保持する値と即値(=3)とを乗算し、乗算結果をレジスタGPR2(g2)に書き込む命令である。乗算命令mulは、加算命令addの演算結果(g0)を用いて乗算を実行するため、加算命令addと乗算命令mulとの間にはデータの依存性がある。

【0116】

演算処理装置100Bの性能を向上するため、データの依存性がある場合、演算結果を、レジスタGUBに書きこむ前に固定小数点演算器28に転送するバイパス制御が実行される。しかしながら、加算命令addを実行する加算器ADDと乗算命令mulを実行する乗算器MULとが物理的に離れているため、加算命令addと乗算命令mulとの間でのデータのバイパスは、2クロックサイクルを用いて実行される。この実施形態では、演算結果のバイパスのクロックサイクル数を増加する制御は、リザベーションステーションRSEによる命令の発行制御ではなく、命令パイプライン内で実行される。

【0117】

クロックサイクル3でリザベーションステーションRSEから加算命令addが発行され、加算命令addのPサイクルが実行される(図12(a))。バリッド信号P-Vは論理1に設定される。加算命令addは、複合命令ではないため、セカンドフロー信号P-2FLは論理0に設定される。命令コードP-ICDは"add"に設定される。リード用のレジスタアドレスP-RAD(R)は、レジスタGPR1、GPR2を示し、ライト用のレジスタアドレスP-RAD(W)は、レジスタGPR0を示す。この後、図9に示す加算命令addと同様に、クロックサイクル3のPサイクルの情報が、クロックサイクル毎に次のサイクルに転送される。

【0118】

クロックサイクル4でリザベーションステーションRSEから乗算命令mulが発行され、乗算命令mulのPサイクルが実行される(図12(b))。バリッド信号P-Vは論理1に設定される。乗算命令mulは、複合命令ではないため、セカンドフロー信号P-2FLが論理0に設定される。命令コードP-ICDは"mul"に設定される。リード用のレジスタアドレスP-RAD(R)は、レジスタGPR0を示し、ライト用のレジスタアドレスP-RAD(W)は、レジスタGPR2を示す。なお、命令パイプラインは、即値(="3")も保持する。クロックサイクル4のPサイクルの情報は、次のクロックサイクル5でPTサイクルに転送される。

10

20

30

40

50

【 0 1 1 9 】

クロックサイクル5において、図11に示す命令依存チェック部ICDCKは、命令コードB1-ICDの"add"と、命令コードPT-ICDの"mul"とを検出する。また、命令依存チェック部ICDCKは、レジスタアドレスB1-RAD(W)の"0"と、レジスタアドレスPT-RAD(R)の"0"との一致を検出する。命令依存チェック部ICDCKは、検出結果に基づいて、加算命令addと乗算命令mulとが、所定の依存関係を有する特定命令であると判定し、依存検出信号DETを論理1に設定する(図12(c))。

【 0 1 2 0 】

クロックサイクル5において、比較器CMP1は、加算命令addのレジスタアドレスB1-RAD(W)の"0"と後続の加算命令addのレジスタアドレスPT-RAD(R)の"0"との一致を検出する。そして、比較器CMP1は、固定小数点演算器28の出力をセクタSEL8に選択させるオペランド選択信号PT-OPSELを生成する。

【 0 1 2 1 】

クロックサイクル5において、オアゲートOR2は、論理1の依存検出信号DETに基づいて論理1を出力する。アンドゲートAND1は、オアゲートOR2の出力とバリッド信号PT-Vとがともに論理1であるため、抑止信号INHを論理1に設定する(図12(d))。アンドゲートAND2は、インバータIV1の出力が論理0であるため、バリッド信号PT-Vの論理1に拘わらず、論理0を出力する。すなわち、アンドゲートAND2は、論理1の依存検出信号DETに基づいて、バリッド信号PT-Vの論理1の出力をマスクする。これにより、クロックサイクル6において、バリッド信号B1-Vは、論理0に設定される(図11(e))。論理0のバリッド信号B1-Vは、クロックサイクル毎に順次転送されるため、リザベーションステーションRSEから投入された乗算命令mulのファーストフロー(1st)は、B1、B2、Xサイクルを実行することなく終了する(図12(f))。

【 0 1 2 2 】

オアゲートOR1は、論理1の抑止信号INHに基づいて論理1を出力するため、クロックサイクル6のバリッド信号PT-Vは論理1に設定される(図12(g))。命令コード操作部ICDOPは、クロックサイクル5で受信した命令コードPT-ICDおよびレジスタアドレスPT-RADを、そのまま命令コードP-ICD2およびレジスタアドレスP-RAD2として出力する。このため、PTサイクルのクロックサイクル6では、クロックサイクル5と同じ命令コードPT-ICDおよびレジスタアドレスPT-RADが出力される(図12(h))。すなわち、乗算命令mulのファーストフロー1stを1クロックサイクル後ろにずらしたセカンドフロー2ndがPTサイクルから開始される(図12(i))。

【 0 1 2 3 】

クロックサイクル6において、命令依存チェック部ICDCKは、命令の依存関係を検出しないため、依存検出信号DETを論理0に設定する(図12(j))。アンドゲートAND1は、論理0の依存検出信号DETに基づいて、抑止信号INHを論理0に設定する(図12(k))。アンドゲートAND2は、論理1のバリッド信号PT-Vと、論理0の依存検出信号DETをインバータIV1で反転させた論理とに基づいて、論理1を出力する。これにより、クロックサイクル7において、バリッド信号B1-Vは、論理1に設定される(図12(l))。この後、加算命令addおよびセカンドフロー2ndとして実行される乗算命令mulは、クロックサイクル毎に次のステージに順次転送される。

【 0 1 2 4 】

クロックサイクル7において、固定小数点演算器28の加算器ADDは、図示しない命令コードX-ICD(add)と、レジスタアドレスX-RADとを受けて演算を実行し、演算結果を出力する(図12(m))。演算結果は、セクタSEL8により選択され、2クロックサイクル後のクロックサイクル9に乗算器MULにバイパスされる(図12(n))。そして、クロックサイクル9において、乗算器MULは、バイパスされた演算

10

20

30

40

50

結果を用いてセカンドフロー 2 n d の乗算命令 m u l を実行する (図 1 2 (o)) 。

【 0 1 2 5 】

図 1 3 は、図 1 1 に示す命令パイプラインを有する演算処理装置 1 0 0 B の動作の別の例を示す。すなわち、図 1 3 は、図 1 1 に示す命令パイプラインを有する演算処理装置 1 0 0 B の制御方法の一例を示す。図 8 および図 1 2 と同一または同様の動作については、詳細な説明は省略する。図 1 3 では、加算命令 " a d d g 0 , g 1 , g 2 " と乗算命令 " m u l g 2 , g 0 , 3 " とが、1クロックサイクルの間隔を空けて、リザベーションステーション R S E から命令パイプラインのステージ P に投入される。図 1 2 と同様に、乗算命令 m u l は、加算命令 a d d の演算結果 (g 0) を用いて乗算を実行するため、加算命令 a d d と乗算命令 m u l との間にはデータの依存性がある。

10

【 0 1 2 6 】

加算命令 a d d は、クロックサイクル 3 にリザベーションステーション R S E から発行され、乗算命令 m u l は、クロックサイクル 5 にリザベーションステーション R S E から発行される (図 1 3 (a) 、 (b)) 。命令依存チェック部 I C D C K は、乗算命令 m u l の P T サイクル (クロックサイクル 6) において、所定の依存関係を検出しないため、依存検出信号 D E T を論理 0 に維持する (図 1 3 (c)) 。オアゲート O R 2 は、セカンドフロー信号 P T - 2 F L と依存検出信号 D E T とがともに論理 0 であるため、論理 0 をアンドゲート A N D 1 に出力する。アンドゲート A N D 1 は、オアゲート O R 2 の出力が論理 0 であるため、抑止信号 I N H を論理 0 に維持する (図 1 3 (d)) 。

【 0 1 2 7 】

20

クロックサイクル 6 において、図 1 1 に示す比較器 C M P 2 は、図示しない B 2 サイクルのレジスタアドレス B 2 - R A D (W) = " 0 " が、P T サイクルのレジスタアドレス P T - R A D (R) = " 0 " に一致すること検出する。そして、比較器 C M P 2 は、固定小数点演算器 2 8 の出力をフリップフロップ F F で 1 クロックサイクル遅らせた加算結果をセクタ S E L 8 に選択させるオペランド選択信号 P T - O P S E L を生成する。この後、加算命令 a d d および乗算命令 m u l は、クロックサイクル毎に次のステージに順次転送される。

【 0 1 2 8 】

クロックサイクル 7 において、固定小数点演算器 2 8 の加算器 A D D は、図示しない命令コード X - I C D (a d d) と、レジスタアドレス X - R A D とを受けて演算を実行し、演算結果を出力する (図 1 3 (e)) 。セクタ S E L 8 は、比較器 C M P 2 が生成したオペランド選択信号 P T - O P S E L に基づいて、フリップフロップ F F により 1 クロックサイクル遅らせた演算結果を選択することで、演算結果をバイパスする (図 1 3 (f)) 。そして、クロックサイクル 9 において、乗算器 M U L は、バイパスされた加算命令 a d d の演算結果を用いて乗算命令 m u l を実行する (図 1 3 (g)) 。

30

【 0 1 2 9 】

図 1 4 は、図 1 1 に示す命令パイプラインを有する演算処理装置 1 0 0 B の動作の別の例を示す。すなわち、図 1 4 は、図 1 1 に示す命令パイプラインを有する演算処理装置 1 0 0 B の制御方法の一例を示す。図 8 、図 1 2 および図 1 3 と同一または同様の動作については、詳細な説明は省略する。図 1 4 では、加算命令 " a d d g 0 , g 1 , g 2 " と減算命令 " s u b g 2 , g 0 , 3 " とが、リザベーションステーション R S E から命令パイプラインのステージ P に連続して投入される。減算命令 s u b は、加算命令 a d d の演算結果 (g 0) を用いて減算を実行するため、加算命令 a d d と減算命令 s u b との間にはデータの依存性がある。

40

【 0 1 3 0 】

加算命令 a d d は、クロックサイクル 3 にリザベーションステーション R S E から発行され、減算命令 s u b は、クロックサイクル 4 にリザベーションステーション R S E から発行される (図 1 4 (a) 、 (b)) 。図 1 2 と同様に、命令依存チェック部 I C D C K は、P T サイクルにおいて、所定の依存関係を検出しないため、依存検出信号 D E T を論理 0 に維持する (図 1 4 (c)) 。これにより、抑止信号 I N H は、論理 0 に維持される

50

(図14(d))。

【0131】

クロックサイクル5において、図11に示す比較器CMP1は、B1サイクルのレジスタアドレスB1-RAD(W) = "0"が、PTサイクルのレジスタアドレスPT-RAD(R) = "0"に一致することを確認する。そして、比較器CMP1は、固定小数点演算器28の出力をセクタSEL8に選択させるオペランド選択信号PT-OPSELを生成する。この後、加算命令addおよび減算命令subは、クロックサイクル毎に次のステージに順次転送される。

【0132】

クロックサイクル7において、固定小数点演算器28の加算器ADDは、加算命令addを実行し、演算結果を出力する(図14(e))。セクタSEL8は、比較器CMP1が生成したオペランド選択信号PT-OPSELに基づいて、固定小数点演算器28から出力された演算結果を選択することで、演算結果をバイパスする(図14(f))。そして、クロックサイクル8において、加算器ADDは、バイパスされた演算結果を用いて減算命令subを実行する(図14(g))。

【0133】

図15は、図11に示す命令パイプラインにおけるPサイクルおよびPTサイクルでの動作フローの一例を示す。すなわち、図15は、図11に示す命令パイプラインを有する演算処理装置100Bの制御方法の一例を示す。図10と同じ処理については、詳細な説明は省略する。図15は、ステップS110で否定判定(N)された場合、ステップS122、S124、S126、S128、S130が実行されることを除き、図10の処理と同様である。すなわち、ステップS110において、セカンドフロー信号PT-2FLが論理1の場合の処理は、図10と同じである。

【0134】

ステップS110において、セカンドフロー信号PT-2FLが論理0の場合、処理はステップS122に移行される。ステップS122、S124、S126では、連続する2つの命令間に所定の依存関係があるか否か(すなわち、特定命令であるか否か)が検出される。例えば、加算命令addのB1サイクルと乗算命令mulのPTサイクルとが同じクロックサイクルで実行され、かつ、加算命令addと乗算命令mulとの間にデータの依存性があるか否かが検出される。

【0135】

ステップS122において、命令パイプラインは、乗算命令mulのPTサイクルの場合、処理をステップS124に移行し、乗算命令mulのPTサイクルでない場合、処理をステップS150に移行する。ステップS124において、命令パイプラインは、B1サイクルのレジスタアドレスB1-WUBAとPTサイクルのレジスタアドレスPT-RUBAとが一致する場合、処理をステップS126に移行する。一方、命令パイプラインは、B1サイクルのレジスタアドレスB1-WUBAとPTサイクルのレジスタアドレスPT-RUBAとが一致しない場合、処理をステップS150に移行する。

【0136】

ステップS126において、命令パイプラインは、バリッド信号B1-V1が論理1で、サイクルB1の命令コードB-ICDが加算命令addである場合、特定命令を検出したため、処理をステップS128に移行する。命令パイプラインは、バリッド信号B1-V1が論理0、または、サイクルB1の命令コードB-ICDが加算命令addでない場合、処理をステップS150に移行する。

【0137】

ステップS128において、命令パイプラインは、命令コード操作部ICDOPにより、命令コードPT-ICDおよびレジスタアドレスPT-RADを、命令コードP-ICD2およびレジスタアドレスP-RAD2に設定する。次に、ステップS130において、命令パイプラインは、次のクロックサイクルのバリッド信号B1-Vを論理0に設定し、処理をステップS140に移行する。この後、ステップS140、S150では、図1

10

20

30

40

50

0と同じ処理が実行される。

【0138】

以上、図11から図15に示す実施形態においても、図1から図10に示す実施形態と同様の効果を得ることができる。例えば、複合命令を命令パイプライン内で複数の個別命令に分解することで、分解された複数の個別命令をリザベーションステーションRSEに投入する場合に比べて、エン트리ENTの使用効率を向上することができ、アウトオブオーダーの効率を向上することができる。この結果、分解された複数の個別命令をリザベーションステーションRSEに投入する場合に比べて、複合命令を実行する演算処理装置100Bの性能を向上することができる。複合命令に含まれる複数の個別命令を不可分で実行するための制御機能をリザベーションステーションRSEから省くことができるため、リザベーションステーションRSEの回路規模の増大を抑止することができる。アンドゲートAND1、オアゲートOR1およびセレクトSEL2、SEL3により、命令パイプライン内に誤った命令が転送されることを抑止することができ、演算処理装置100Bの誤動作を抑止することができる。

10

【0139】

さらに、図11から図15に示す実施形態では、以下に示す効果を得ることができる。すなわち、命令パイプライン内に命令依存チェック部ICDCKを設けることで、特定命令を検出する機能をリザベーションステーションRSEに設けることなく、特定命令を検出することができる。例えば、命令依存チェック部ICDCKは、B1、PTサイクルで転送される命令コードB1-ICD、PT-ICDを比較し、B1、PTサイクルでそれぞれ転送されるレジスタアドレスUBAを比較することで、特定命令を検出することができる。

20

【0140】

また、命令パイプライン内に命令依存チェック部ICDCKを設けることで、特定命令を検出する機能をリザベーションステーションRSEに設ける場合に比べて、演算処理装置100Bの回路規模の増大を抑止することができる。例えば、リザベーションステーションRSEにより特定命令を検出する場合、全てのエントリENTに保持された命令コードICDを相互にチェックする機能が、リザベーションステーションRSEに設けられる。また、特定命令を検出した場合、特定命令のうち後続の命令を先行の命令に対して所定のクロックサイクルを空けて命令パイプライン内に投入する機能が、リザベーションステーションRSEに設けられる。このため、特定命令を検出する機能を有するリザベーションステーションRSEの回路規模は、図11に示すリザベーションステーションRSEの回路規模に比べて増大する。

30

【0141】

複合命令において2番目に実行される個別命令を生成する命令コード操作部ICDOPを利用して、特定命令における2番目に実行される命令を1クロックサイクルずらす制御を実行することができる。このため、特定命令における2番目に実行される命令を1クロックサイクルずらす制御を実行する回路を新たに設ける場合に比べて、命令パイプラインの回路規模の増大を抑制することができる。

【0142】

オアゲートOR2により、複合命令の投入時と特定命令の検出時に抑止信号INHを論理1に設定することで、セカンドフロー2ndの命令と、リザベーションステーションRSEから投入される命令とが衝突することを抑止することができる。これにより、命令パイプライン内に誤った命令が転送されることを抑止することができ、演算処理装置100Bの誤動作を抑止することができる。

40

【0143】

インバータIV1とアンドゲートAND2とにより、特定命令が検出された次のクロックサイクルのバリッド信号B1-Vを無効にすることで、乗算命令mulのファーストフロー1stのB1サイクル以降を無効にすることができる。これにより、命令コード操作部ICDOPが生成する1クロックサイクル遅らせた乗算命令mulのセカンドフロー2

50

ndを乗算命令mulのファーストフロー1stと衝突させることなく実行することができる。したがって、加算命令addの演算結果を、乗算命令mulのセカンドフロー2ndを実行する乗算器MULにバイパスすることができる。

【0144】

なお、図11から図15に示す実施形態は、固定小数点演算器28が実行する複合命令および特定命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用する例について説明される。しかしながら、図11から図15に示す実施形態は、固定小数点演算器28と論理演算器とにより実行される複合命令および特定命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用されてもよい。また、図11から図15に示す実施形態は、浮動小数点演算器が実行する複合命令および特定命令をファーストフロー1stとセカンドフロー2ndとに分解する命令パイプラインに適用されてもよい。

10

【0145】

以上の実施形態において説明した発明を整理して、付記として以下の通り開示する。

(付記1)

命令をデコードするデコード部と、

前記デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力する実行制御部と、

前記実行制御部が出力する命令を順次転送する命令転送部と、

前記命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、前記複合命令から生成した個別命令を前記命令転送部に出力する命令生成部と、

20

前記命令転送部が転送した個別命令を実行する演算実行部を有することを特徴とする演算処理装置。

(付記2)

前記命令生成部が前記複合命令に含まれる個別命令を前記命令転送部に出力する場合、前記実行制御部に命令の出力を抑止させる抑止情報を出力する抑止制御部を有することを特徴とする付記1記載の演算処理装置。

(付記3)

前記実行制御部が出力する前記複合命令は、前記複合命令であることを識別する識別情報を含み、

30

前記抑止制御部は、前記実行制御部が出力する前記識別情報に基づいて、前記抑止情報を出力することを特徴とする付記2記載の演算処理装置。

(付記4)

前記命令転送部は、前記抑止制御部が前記抑止情報を出力しない場合、前記実行制御部が出力する前記識別情報を選択し、前記抑止制御部が前記抑止情報を出力する場合、命令が前記複合命令でないことを示す識別情報を選択する第1の選択部を有し、前記第1の選択部で選択した識別情報を転送することを特徴とする付記3記載の演算処理装置。

(付記5)

前記命令転送部は、前記抑止制御部が前記抑止情報を出力しない場合、前記実行制御部が出力する命令を選択し、前記抑止制御部が前記抑止情報を出力する場合、前記命令生成部が出力する個別命令を選択する第2の選択部を有し、前記第2の選択部で選択した命令を転送することを特徴とする付記2ないし付記4のいずれか1項記載の演算処理装置。

40

(付記6)

前記実行制御部が出力する命令は、命令が有効であることを示す有効情報を含み、

前記命令転送部は、前記実行制御部が出力する有効情報または前記抑止情報を有効情報として前記演算実行部に転送する論理和部を有することを特徴とする付記2ないし付記5のいずれか1項記載の演算処理装置。

(付記7)

前記命令生成部は、前記実行制御部が前記複合命令を出力する場合、前記複数の個別命令のうち、前記演算実行部による実行順が2番目以降の個別命令を生成し、

50

前記命令転送部は、前記実行制御部が前記複合命令を出力する場合、前記複合命令の転送に続いて、前記命令生成部が生成する実行順が 2 番目以降の個別命令を転送し、

前記演算実行部は、前記複合命令を受信した場合、前記複数の命令のうち実行順が最初の命令を実行することを特徴とする付記 1 ないし付記 6 のいずれか 1 項記載の演算処理装置。

(付記 8)

前記命令転送部は、

前記実行制御部が出力する命令を受信する第 1 のステージと、

前記第 1 のステージから転送される命令を前記演算実行部に出力する第 2 のステージを有し、

前記命令生成部は、前記第 2 のステージに転送される命令が前記複合命令の場合、前記実行順が 2 番目以降の個別命令を前記第 1 のステージに出力することを特徴とする付記 7 記載の演算処理装置。

(付記 9)

前記実行制御部が順次出力する先行命令および後続命令が、前記先行命令の演算結果を前記後続命令で使用する所定の組み合わせの特定命令であることを検出する依存検出部を有し、

前記命令転送部は、前記実行制御部が出力する命令が前記複合命令の場合、または、前記依存検出部が前記特定命令を検出した場合、前記実行制御部の出力の選択を停止し、前記命令生成部の出力を選択する選択部を有し、

前記命令生成部は、前記実行制御部が出力する命令が前記複合命令の場合、前記複合命令に含まれる個別命令を前記命令転送部に出力し、前記実行制御部が出力する命令が前記複合命令でない場合、前記命令転送部で転送された命令を前記命令転送部に出力することを特徴とする付記 1 記載の演算処理装置。

(付記 10)

前記命令生成部が前記複合命令に含まれる個別命令を前記命令転送部に出力する場合、または前記依存検出部が前記特定命令を検出した場合、前記実行制御部に命令の出力を抑制させる抑止情報を出力する抑止制御部を有し、

前記選択部は、前記抑止制御部が前記抑止情報を出力しない場合、前記実行制御部が出力する命令を選択し、前記抑止制御部が前記抑止情報を出力する場合、前記命令生成部が出力する命令を選択することを特徴とする付記 9 記載の演算処理装置。

(付記 11)

前記実行制御部が出力する命令は、命令が有効であることを示す有効情報を含み、

前記命令転送部は、前記依存検出部が前記特定命令を検出した場合、前記実行制御部が出力した前記後続命令に含まれる有効情報を無効状態に設定する無効設定部を有することを特徴とする付記 9 または付記 10 記載の演算処理装置。

(付記 12)

前記特定命令は、前記先行命令を演算する前記演算実行部内の演算器が演算結果を出力してから、前記後続命令を演算する前記演算実行部内の他の演算器に前記演算結果がバイパスされるまでの時間が、1 クロックサイクルを超える組み合わせの命令であることを特徴とする付記 9 ないし付記 11 のいずれか 1 項記載の演算処理装置。

(付記 13)

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デコード部が、命令をデコードし、

前記演算処理装置が有する実行制御部が、前記デコード部がデコードした命令を保持し、保持した命令を実行可能な順に出力し、

前記演算処理装置が有する命令転送部が、前記実行制御部が出力する命令を順次転送し、

前記命令転送部が転送する命令が複数の個別命令を組み合わせた複合命令である場合、前記演算処理装置が有する命令生成部が、前記複合命令から生成した個別命令を前記命令

10

20

30

40

50

転送部に出し、

前記演算処理装置が有する演算実行部が、前記命令転送部が転送した個別命令を実行することを特徴とする演算処理装置の制御方法。

(付記 14)

前記演算処理装置が有する抑止制御部が、前記命令生成部が前記複合命令に含まれる命令を前記命令転送部に出し、前記実行制御部が命令の出力を抑止させる抑止情報を出し、

(付記 15)

前記実行制御部が出力する前記複合命令は、前記複合命令であることを識別する識別情報を含み、

前記抑止制御部は、前記実行制御部が出力する前記識別情報に基づいて、前記抑止情報を出し、

(付記 16)

前記演算処理装置が有する依存検出部が、前記実行制御部が順次出力する先行命令および後続命令が、前記先行命令の演算結果を前記後続命令で使用する所定の組み合わせの特定命令であることを検出し、

前記命令転送部は、前記実行制御部が出力する命令が前記複合命令の場合、または、前記依存検出部が前記特定命令を検出した場合、前記実行制御部の出力の選択を停止し、前記命令生成部の出力を選択する選択部を有し、

前記命令生成部は、前記実行制御部が出力する命令が前記複合命令の場合、前記複合命令に含まれる命令を前記命令転送部に出し、前記実行制御部が出力する命令が前記複合命令でない場合、前記命令転送部で転送された命令を前記命令転送部に出し、

(付記 17)

前記演算処理装置が有する抑止制御部が、前記命令生成部が前記複合命令に含まれる命令を前記命令転送部に出し、または前記依存検出部が前記特定命令を検出した場合、前記実行制御部が命令の出力を抑止させる抑止情報を出し、

前記選択部は、前記抑止制御部が前記抑止情報を出し、前記実行制御部が出力する命令を選択し、前記抑止制御部が前記抑止情報を出し、前記命令生成部が出力する命令を選択することを特徴とする付記 16 記載の演算処理装置の制御方法。

(付記 18)

前記実行制御部が出力する命令は、命令が有効であることを示す有効情報を含み、

前記命令転送部の無効設定部が、前記依存検出部が前記特定命令を検出した場合、前記実行制御部が出力した前記後続命令に含まれる有効情報を無効状態に設定することを特徴とする付記 16 または付記 17 記載の演算処理装置の制御方法。

【0146】

以上の詳細な説明により、実施形態の特徴点および利点は明らかになるであろう。これは、特許請求の範囲がその精神および権利範囲を逸脱しない範囲で前述のような実施形態の特徴点および利点にまで及ぶことを意図するものである。また、当該技術分野において通常の知識を有する者であれば、あらゆる改良および変更に容易に想到できるはずである。したがって、発明性を有する実施形態の範囲を前述したものに限定する意図はなく、実施形態に開示された範囲に含まれる適当な改良物および均等物に拠ることも可能である。

【符号の説明】

【0147】

1 ... デコード部； 2 ... 実行制御部； 2FL ... セカンドフローフラグ、セカンドフロー信号； 3 ... 命令転送部； 4 ... 演算実行部； 5 ... 命令生成部； 6 ... 抑止制御部； 10 ... 命令キャッシュ； 12 ... 命令バッファ； 14 ... デコード部； 16 ... レジスタ管理部； 18 ... リザベーションステーション部； 20 ... バイパス制御部； 22 ... 演算実行部； 24 ... データキャッシュ； 28 ... 固定小数点演算器； 30 ... アドレス計算演算器； 32 ... レジスタ部； 100、100A、100B ... 演算処理装置； AND1、AND2 ... アンドゲート； B1、

10

20

30

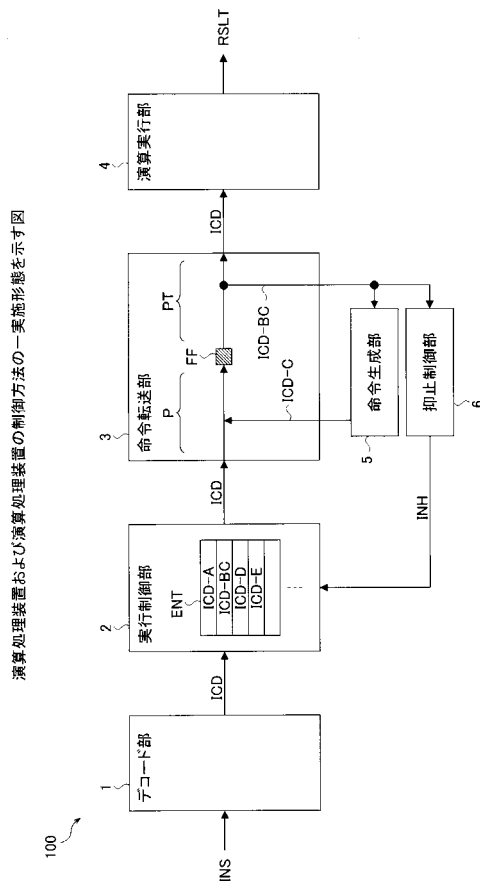
40

50

B 2 ... ステージ ; B P T B L ... バイパス管理テーブル ; C M P 1、C M P 2、C M P 3 ... 比較器 ; D ... ステージ ; D A T ... データ ; D T ... ステージ ; E N T ... エントリ ; F F ... フリップフロップ ; G P ... エントリ ; G P R ... レジスタ ; G P R N ... レジスタ番号 ; G P R T ... テーブル ; G P R - V ... フラグ ; G U B ... レジスタ ; G U B N ... レジスタ番号 ; I C D ... 命令コード ; I C D - B C ... 複合命令 ; I C D C K ... 命令依存チェック部 ; I C D O P ... 命令コード操作部 ; I N H ... 抑止情報、抑止信号 ; I N S ... 命令 ; I V 1 ... インバータ ; O P S E L ... オペランド選択信号 ; O R 1、O R 2 ... オアゲート ; P、P T ... ステージ ; R A D ... レジスタアドレス ; R S A、R S E ... リザベーションステーション ; R S L T ... 演算結果 ; R U B A ... レジスタアドレス ; S E L 1、S E L 2、S E L 3、S E L 4、S E L 5、S E L 6、S E L 7、S E L 8 ... セクタ ; U B A ... レジスタアドレス ; V ... バリッドフラグ、バリッド信号 ; W U B A ... レジスタアドレス ; X ... ステージ

10

【図 1】



【図 2】

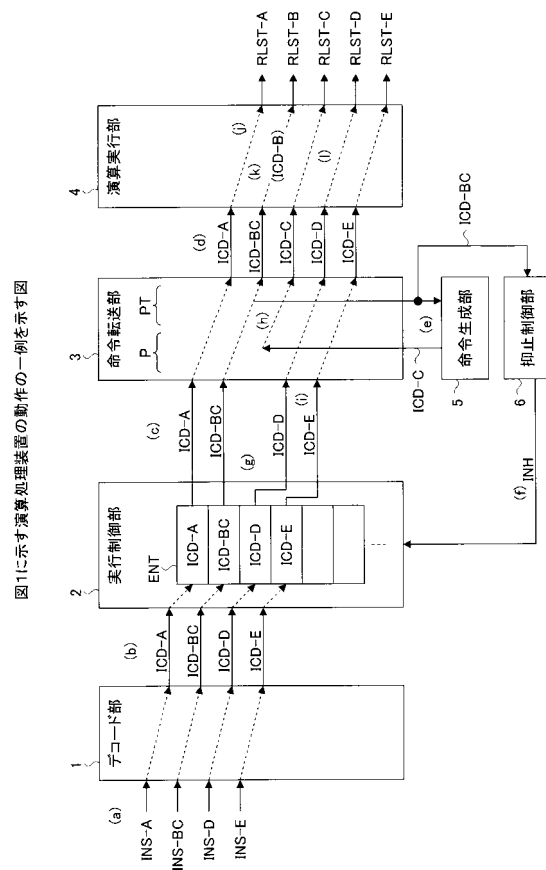
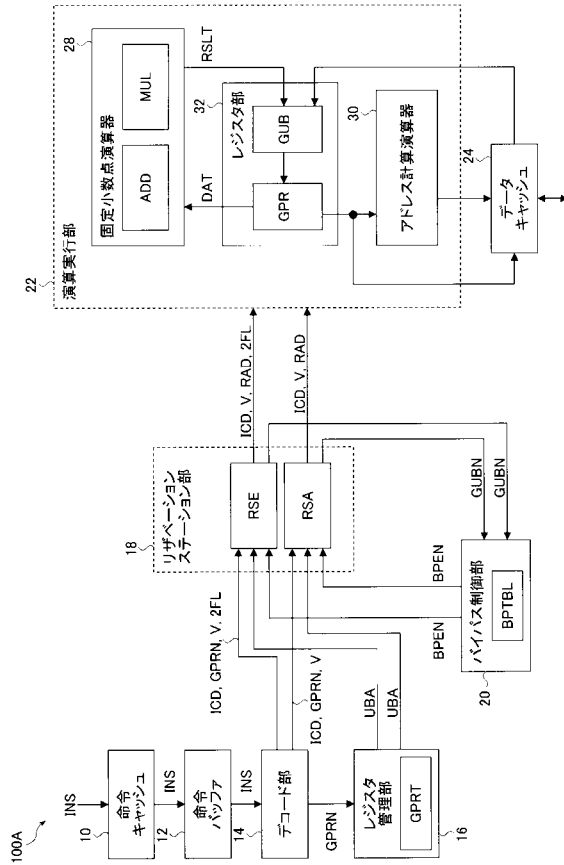


図1に示す演算処理装置の動作の一例を示す図

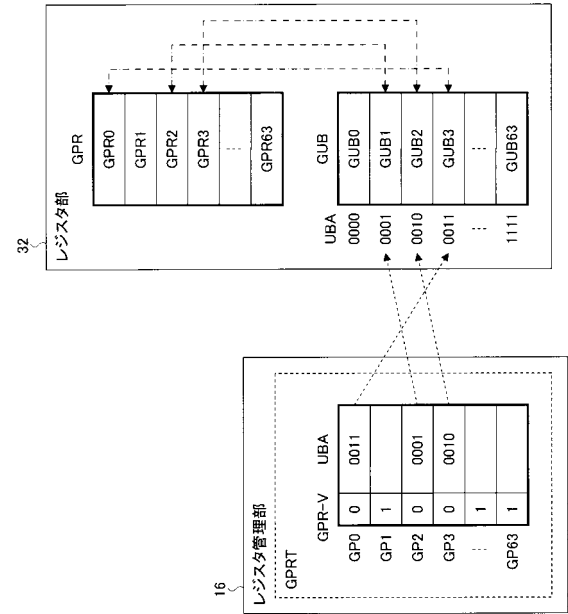
【図 3】

演算処理装置および演算処理装置の制御方法の別の実施形態を示す図



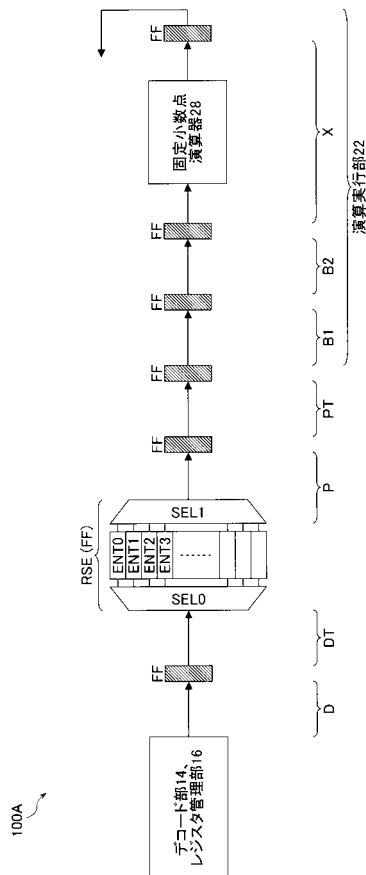
【図 4】

図3に示すレジスタ管理部およびレジスタ部の一例を示す図



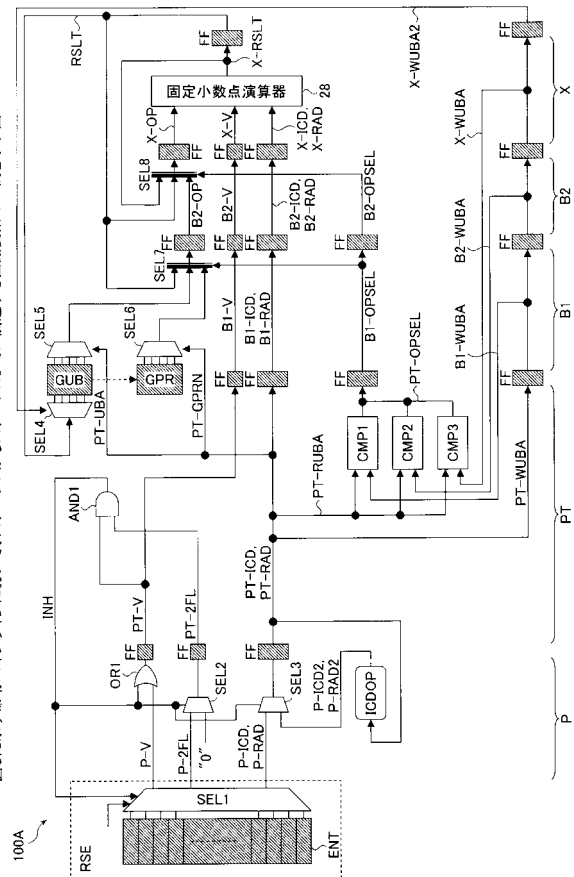
【図 5】

図3に示す演算処理装置における固定小数点演算器に対する命令パイプラインの概要を示す図



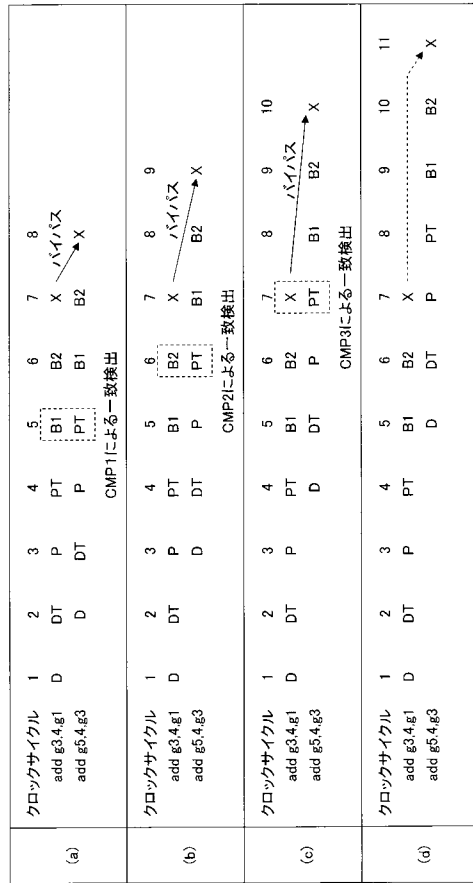
【図 6】

図5に示す命令パイプラインにおいて、ステージPからステージXまでに関連する回路要素の一例を示す図



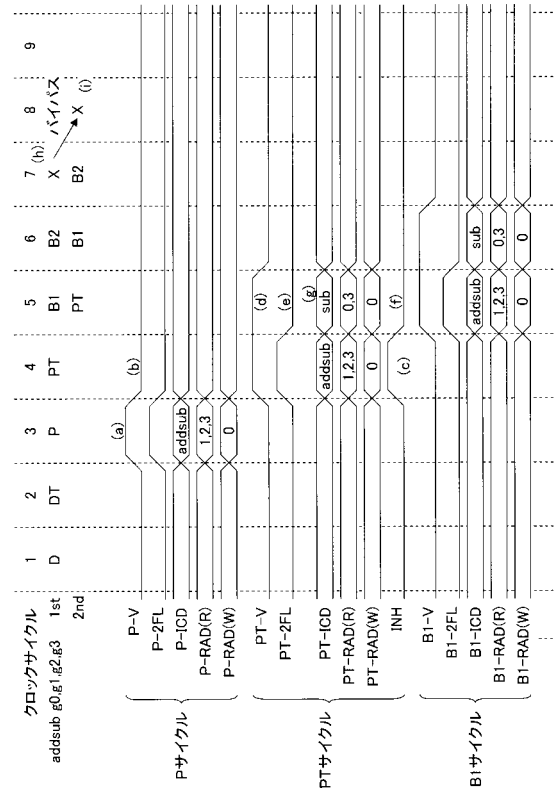
【図 7】

図5および図6に示す命令パイプラインにおいて、オペランドデータをバイパスさせるタイミングの一例を示す図



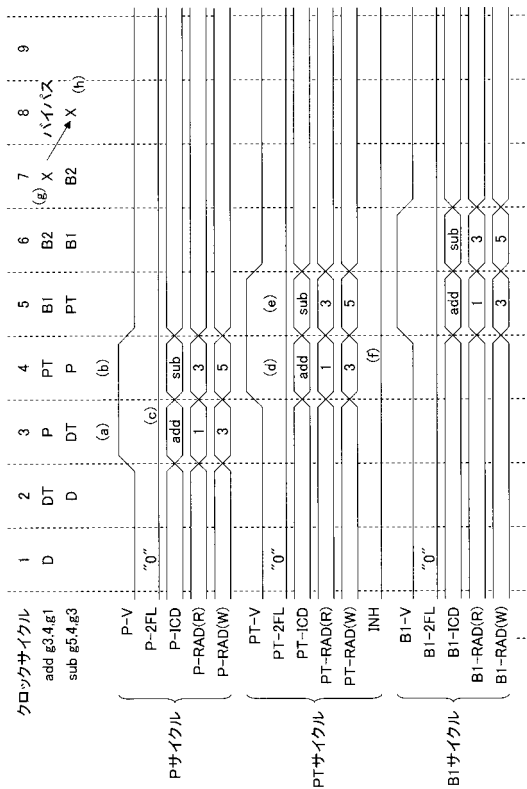
【図 8】

図3に示す演算処理装置の動作の一例を示す図



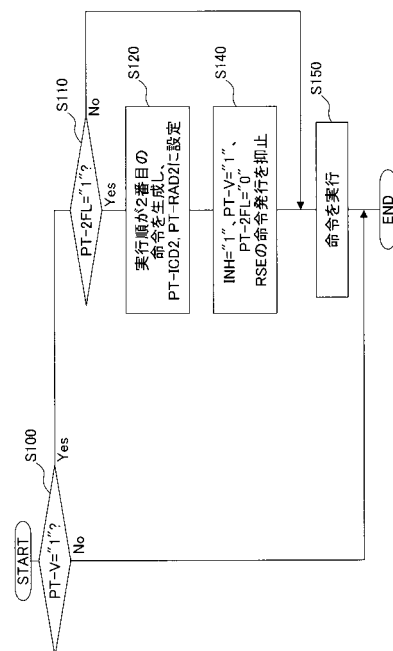
【図 9】

図3に示す演算処理装置の動作の別の例を示す図



【図 10】

図6に示す命令パイプラインにおけるPサイクルおよびPTサイクルでの動作フローの一例を示す図



【図 15】

図 11 に示す命令パイプラインにおける P サイクルおよび PT サイクルでの動作フローの一例を示す図

