

(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2014-2555
(P2014-2555A)

(43) 公開日 平成26年1月9日(2014.1.9)

(51) Int.Cl.
G06F 9/38 (2006.01)

F I
G06F 9/38 310G

テーマコード(参考)
5B013

審査請求 未請求 請求項の数 4 O L (全 15 頁)

(21) 出願番号 特願2012-137331 (P2012-137331)
(22) 出願日 平成24年6月18日 (2012.6.18)

(71) 出願人 000005223
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 100090273
弁理士 園分 孝悦
(72) 発明者 堀尾 一生
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 AA14 AA18

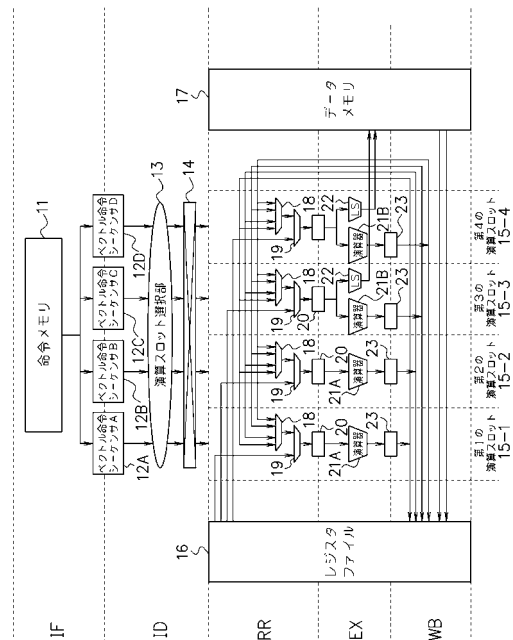
(54) 【発明の名称】 プロセッサ

(57) 【要約】

【課題】 プロセッサが有する演算リソースを有効に活用し、高い性能を得られるようにする。

【解決手段】 1つのベクトル命令に係る処理を複数の部分命令処理に分割して部分命令処理に係る実行指示をベクトル命令シーケンサより発行し、演算スロット選択部が所定の優先順位に従ってベクトル命令シーケンサからの実行指示が供給される演算スロットを選択して実行指示に応じた処理を実行させるとともに、演算スロット選択部がベクトル命令に基づく複数の実行指示を、その処理の内容や演算スロットへの実行指示の供給状況に応じて演算スロットに供給するようにして、1つのベクトル命令から分割された複数の実行指示を異なる演算スロットに発行可能にし、演算スロットを有効に活用して使用率を高められるようにする。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

複数の演算命令を有するベクトル命令処理を複数の部分命令処理に分割して当該部分命令処理に係る実行指示を発行する発行部と、

前記実行指示を受けて当該実行指示に応じた処理を行う複数の処理部と、

前記複数の処理部の内から 1 つの前記処理部を選択して、前記発行部から発行される前記実行指示を供給する選択部とを備え、

前記選択部は、前記ベクトル命令処理に基づく複数の前記実行指示を、前記実行指示に応じた処理の内容及び前記処理部への実行指示の供給状況に応じて前記処理部に供給することを特徴とするプロセッサ。

10

【請求項 2】

前記選択部は、一の前記ベクトル命令処理に基づく前記実行指示であって、前記発行部から同時に発行された複数の前記実行指示を、異なる前記処理部に同時に供給し並列実行させることを特徴とする請求項 1 記載のプロセッサ。

【請求項 3】

前記複数の処理部は、第 1 の処理を実行可能な第 1 の処理部と、前記第 1 の処理を実行可能であるとともに前記第 1 の処理とは異なる第 2 の処理を実行可能な第 2 の処理部とを含み、

前記選択部は、前記第 2 の処理を実行させる命令の前記実行指示を、前記第 1 の処理を実行させる命令の前記実行指示よりも優先して前記第 2 の処理部に供給することを特徴とする請求項 1 又は 2 記載のプロセッサ。

20

【請求項 4】

前記処理部は、前記第 1 の処理を実行可能であるとともに前記第 1 の処理及び前記第 2 の処理とは異なる第 3 の処理を実行可能な第 3 の処理部を含み、

前記第 2 の処理部には、前記第 1 の処理を実行させる命令の前記実行指示及び前記第 2 の処理を実行させる命令の前記実行指示が供給され、

前記第 3 の処理部には、前記第 1 の処理を実行させる命令の前記実行指示及び前記第 3 の処理を実行させる命令の前記実行指示が供給され、

前記選択部は、前記第 3 の処理を実行させる命令の前記実行指示を、前記第 1 の処理を実行させる命令の前記実行指示よりも優先して前記第 3 の処理部に供給することを特徴とする請求項 3 記載のプロセッサ。

30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プロセッサに関する。

【背景技術】

【0002】

ベクトル命令は、1 つの命令で多数のデータについて演算を行うような命令である。そのベクトル命令を実行可能なプロセッサが、ベクトル・プロセッサである。例えばベクトル命令では、プロセッサ内の連続したレジスタ領域に対して操作を行う。1 つのベクトル命令で実行する演算処理の数は、ベクトル・プロセッサが実際に有する演算器の数に制限されず、むしろベクトル・プロセッサが実際に有する演算器の数よりも多く設定されることが普通である。ベクトル・プロセッサが有する演算器の数よりもベクトル命令で実行する演算処理の数が多い場合、ベクトル・プロセッサは、そのベクトル命令に係るすべての演算処理を 1 サイクルで処理することができない。

40

【0003】

例えば、図 6 (A) に示した命令 “Add r0-5 + r6-11 r12-17” は、レジスタ r_i とレジスタ $r(i+6)$ とのデータを加算した結果をレジスタ $r(i+12)$ に格納するベクトル命令である (i は添え字であり、 $i = 0, 1, 2, 3, 4, 5$)。すなわち、命令 “Add r0-5 + r6-11 r12-17” は、この 1 つのベクトル命令で 6 つの加算演算を指

50

示する。ベクトル・プロセッサが、6つの加算器を備えていれば命令“Add r0-5 + r6-11 r12-17”による処理を1サイクルで実行することが可能であるが、それだけの加算器を備えているとは限らない。命令“Add r0-5 + r6-11 r12-17”を実行するために、ベクトル・プロセッサ内部では、この1つのベクトル命令を、図6(B)に示すように複数の部分命令Add 0-0~Add 0-5の集まりのようにみなしてベクトル命令を分割して実行する。すなわち、ベクトル・プロセッサは、その内部で、ベクトル命令による処理を、処理自体の種別が同じであるが、演算に使用するデータ(レジスタ)が異なる複数の処理に分割して実行する。

【0004】

図7は、ベクトル命令を実行可能なベクトル・プロセッサの構成例を示す図である。図7において、IFは命令フェッチステージ、IDは命令デコードステージ、RRはレジスタ読み出しステージ、EXは演算実行ステージ、WBはレジスタ書き込みステージである。図7に示したベクトル・プロセッサは、5ステージのパイプライン構成のプロセッサであり、4つの演算スロットを有している。

【0005】

命令フェッチステージIFでは、ベクトル命令が命令メモリ101から読み出される。ベクトル命令には、レジスタファイル105の複数のエントリ(データ)に対して一括して演算を指示するベクトル命令を含む。命令デコードステージIDでは、演算スロット選択部102が、4つの演算スロット104-1~104-4の内から、命令フェッチステージIFにおいて読み出された命令に係る処理を実行する演算スロットを選択する。命令フェッチステージIFにおいて命令メモリ101から読み出された命令は、演算スロット選択部102によって選択されたスロットに対応するベクトル命令シーケンサ103-1~103-4に格納される。また、命令デコードステージIDでは、ベクトル命令シーケンサ103-1~103-4が、格納されているベクトル命令を複数の部分命令に分割して、対応する演算スロット104-1~104-4に順次発行する。

【0006】

レジスタ読み出しステージRRでは、レジスタファイル105から読み出したデータ及びレジスタファイル105へ書き込む前の演算結果であるバイパスデータの内から、演算処理に使用するデータがマルチプレクサ107、108により選択される。マルチプレクサ107、108によって選択されたデータは、ソースデータとしてパイプラインラッチ109に格納される。演算実行ステージEXでは、命令が演算命令であれば、命令によって指示された演算処理が演算器110A、110Bにより実行され、演算結果がパイプラインラッチ112に格納される。また、演算実行ステージEXでは、命令がロード・ストア命令であれば、命令によって指示されたデータメモリ106へのアクセスがロード・ストアユニット111により実行され、データメモリ106に対してデータの読み出しや書き込みが行われる。レジスタ書き込みステージWBでは、演算実行ステージEXでの演算結果がレジスタファイル105の指定されたレジスタに書き込まれる。

【0007】

ここで、図7に示した例では、第1の演算スロット104-1及び第2の演算スロット104-2が有する演算器110Aは、乗算器を有し乗算命令が実行可能である。一方、第3の演算スロット104-3及び第4の演算スロット104-4が有する演算器110Bは、乗算器を有していない。また、第3の演算スロット104-3及び第4の演算スロット104-4が、ロード・ストアユニット111を有する。したがって、演算スロット選択部102は、命令メモリ101から読み出されたベクトル命令が乗算命令である場合には命令に係る処理を実行する演算スロットとして、第1の演算スロット104-1又は第2の演算スロット104-2を選択する。また、演算スロット選択部102は、命令メモリ101から読み出されたベクトル命令がロード・ストア命令である場合には命令に係る処理を実行する演算スロットとして、第3の演算スロット104-3又は第4の演算スロット104-4を選択する。

【先行技術文献】

10

20

30

40

50

【特許文献】

【0008】

【特許文献1】特開2002-55788号公報

【発明の概要】

【発明が解決しようとする課題】

【0009】

図7に構成例を示したベクトル・プロセッサにおいて、1サイクルに1ベクトル命令を命令メモリ101から読み出して発行可能であるとする。また、演算スロット選択部102は、空いている演算スロットが複数存在する場合には、第1の演算スロット104-1、第2の演算スロット104-2、第3の演算スロット104-3、第4の演算スロット104-4の順で、第1の演算スロット104-1から優先的に命令を発行するものとする。

10

【0010】

このような場合、命令メモリ101から読み出されたベクトル命令は、演算スロット選択部102により選択された、ある1つのベクトル命令シーケンサ103-1~103-4に格納される。そして、ベクトル命令シーケンサ103-1~103-4によって、格納された1つのベクトル命令から分割された複数の部分命令は、すべて同じ演算スロット104-1~104-4に対して発行される。例えば、ベクトル命令Add 0がベクトル命令シーケンサ103-1に格納された場合には、図7に示すようにベクトル命令Add 0から分割されたすべての部分命令Add 0-1、Add 0-2、Add 0-3、...が第1の演算スロット104-1に対して発行される。そのため、例えば関数呼び出しの先頭などでは、すべての演算スロット104-1~104-4にて命令に係る処理を実行する状態となるまでに、少なくとも4サイクルを要することになり、その間、幾つかの演算スロット104-1~104-4では処理を行っていない。また、例えば第1の演算スロット104-1及び第2の演算スロット104-2が、先行のベクトル命令に係る処理を実行しているときに、命令メモリ101から読み出された後続のベクトル命令が乗算命令である場合、演算スロット104-1又は104-2での処理が完了して空くまで待機しなければならない。

20

【0011】

本発明は、プロセッサが有する演算リソースを有効に活用し、高い性能を得ることができるようになることを目的とする。

30

【課題を解決するための手段】

【0012】

プロセッサの一態様は、複数の演算命令を有するベクトル命令処理を複数の部分命令処理に分割して当該部分命令処理に係る実行指示を発行する発行部と、受けた実行指示に応じた処理を行う複数の処理部と、複数の処理部の内から1つの処理部を選択し、発行部から発行される実行指示を供給する選択部とを備える。選択部は、ベクトル命令処理に基づく複数の実行指示を、実行指示に応じた処理の内容及び処理部への実行指示の供給状況に応じて処理部に供給する。

【発明の効果】

【0013】

開示のプロセッサは、ベクトル命令処理に基づく複数の実行指示を処理部に適切に供給して処理を実行することができ、演算リソースを有効に活用して使用率を高め性能向上を図ることができる。

40

【図面の簡単な説明】

【0014】

【図1】本発明の実施形態におけるプロセッサの構成例を示す図である。

【図2】本実施形態におけるベクトル命令シーケンサの構成例を示す図である。

【図3】本実施形態における演算スロット選択部及びクロスバーの構成例を示す図である。

【図4】本実施形態及び一般的な技術での処理動作の例を示す図である。

50

【図5】本実施形態及び一般的な技術での処理動作の他の例を示す図である。

【図6】ベクトル命令を説明するための図である。

【図7】ベクトル命令を実行可能なプロセッサの構成例を示す図である。

【発明を実施するための形態】

【0015】

以下、本発明の実施形態を図面に基づいて説明する。

【0016】

図1は、本発明の一実施形態によるプロセッサの構成例を示す図である。本実施形態におけるプロセッサは、ベクトル命令を実行可能なベクトル・プロセッサであり、その内部で、ベクトル命令による処理を、処理自体の種別が同じであるが、演算に使用するデータ（レジスタ）が異なる複数の処理に分割して実行する。なお、以下の説明では、ベクトル命令による処理を分割した複数の処理の各処理に係る実行指示を、ベクトル命令の部分命令とも称する。言い換えれば、本実施形態におけるプロセッサは、その内部で、1つのベクトル命令を複数の部分命令の集まりのようにみなして、ベクトル命令を部分命令に分割して処理を実行する。

10

【0017】

図1において、IFは命令フェッチステージ、IDは命令デコードステージ、RRはレジスタ読み出しステージ、EXは演算実行ステージ、WBはレジスタ書き込みステージである。図1に示すプロセッサは、5ステージのパイプライン構成のベクトル・プロセッサであり、4つの演算スロットを有している。

20

【0018】

命令メモリ11には、プロセッサにてプログラムに応じた処理を実行するための命令列が格納されている。命令メモリ11に格納されるベクトル命令には、レジスタファイル16の複数のエントリ（データ）に対して一括して演算を指示するベクトル命令を含む。

【0019】

ベクトル命令シーケンサ12A～12Dは、命令メモリ11から読み出されたベクトル命令が格納される。ベクトル命令シーケンサ12A～12Dは、格納されたベクトル命令を、ベクトル命令におけるベクトル長（Vector Length）個の部分命令に分割して発行する。ベクトル命令シーケンサ12A～12Dのそれぞれは、1つのベクトル命令が格納されると、そのベクトル命令が分割された部分命令のすべてを発行し終わるまで、次のベクトル命令が格納されない。

30

【0020】

演算スロット選択部13は、4つの演算スロット15-1～15-4の内から、ベクトル命令シーケンサ12A～12Dから発行される部分命令に係る処理を実行する演算スロットを選択する。演算スロット選択部13は、各ベクトル命令シーケンサ12A～12Dより演算スロット15-1～15-4に対する部分命令の発行に係るリクエストを受け取る。演算スロット選択部13は、受け取ったリクエストに対して、所定の優先順位に従ってベクトル命令シーケンサ12A～12Dに演算スロット15-1～15-4を割り当てる。クロスバー14は、演算スロット選択部13による割り当てに応じて、ベクトル命令シーケンサ12A～12Dから発行される部分命令を演算スロット15-1～15-4に供給する。

40

【0021】

演算スロット15-1～15-4は、ベクトル命令シーケンサ12A～12Dから発行される部分命令に係る処理を行う。演算スロット15-1～15-4は、マルチプレクサ18、19、パイプラインラッチ20、23、及び演算器21A（又は21B）を有する。また、第3の演算スロット15-3及び第4の演算スロット15-4は、さらにロード・ストアユニット22を有する。

【0022】

マルチプレクサ18は、レジスタファイル16へ書き込む前の演算結果であるバイパスデータの内から命令実行ステージにおいて使用されるデータを選択する。また、マルチプ

50

レクサ 19 は、レジスタファイル 16 から読み出したデータ及びマルチプレクサ 18 により選択されたデータの内から命令実行ステージにおいて使用されるデータを選択する。マルチプレクサ 19 により選択されたデータは、パイプラインラッチ 20 に格納される。

【0023】

演算器 21 A、21 B は、供給される命令が演算命令である場合に、命令によって指示された演算処理を実行する。演算器 21 A、21 B は、例えば加算命令に応じた加算演算などを実行可能である。演算器 21 A、21 B による演算結果は、パイプラインラッチ 23 に格納される。ここで、本実施形態においては、第 1 の演算スロット 15 - 1 及び第 2 の演算スロット 15 - 2 が有する演算器 21 A は、乗算器を有し乗算命令が実行可能である。一方、第 3 の演算スロット 15 - 3 及び第 4 の演算スロット 15 - 4 が有する演算器 21 B は、乗算器を有していない。ロード・ストアユニット 22 は、供給される命令がロード・ストア命令である場合に、命令によって指示されたデータメモリ 17 へのアクセスを実行し、データメモリ 17 に対するデータの読み出しや書き込みを行う。

10

【0024】

図 2 は、本実施形態におけるベクトル命令シーケンサ 12 の構成例を示す図である。バッファ群 31 は、命令メモリ 11 から読み出されベクトル命令シーケンサ 12 に格納されるベクトル命令の命令情報を保持する複数のバッファを有する。バッファ 32 は、ベクトル命令のオペコードを保持する。バッファ 33、34 は、ベクトル命令に係る処理で使用されるデータ（ソースデータ）が格納されているソースレジスタのレジスタ番号をそれぞれ保持する。バッファ 35 は、ベクトル命令に係る処理により得られるデータ（演算結果）を格納するディスティネーションレジスタのレジスタ番号を保持する。

20

【0025】

バッファ 32 ~ 35 に保持される情報は、命令メモリ 11 からのベクトル命令の発行時（ベクトル命令シーケンサ 12 へのベクトル命令の格納時）にバッファ 32 ~ 35 へ書き込まれる。また、バッファ 33、34、35 に書き込まれるレジスタ番号は、ベクトル命令で処理の対象とするレジスタ番号の初期値である。例えば前述した命令 “Add r0-5 + r6-11 r12-17” であれば、バッファ 33、34、35 には “0”、“6”、“12” がレジスタ番号として書き込まれて保持される。

【0026】

レジスタ 36 は、ベクトル命令におけるベクトル長が格納される。バリッド制御部 37 は、バッファ群 31（バッファ 32 ~ 35）に保持されているエントリが有効であるか否かを示すバリッド情報を制御する。カウンタ 38 は、ベクトル命令シーケンサ 12 からの部分命令の発行回数をカウントする。

30

【0027】

命令メモリ 11 からのベクトル命令をベクトル命令シーケンサ 12 に格納するとき、バッファ群 31（バッファ 32 ~ 35）にベクトル命令の命令情報が書き込まれるとともに、そのベクトル命令におけるベクトル長がレジスタ 36 に設定される。また、バリッド制御部 37 に信号 `set` が入力されてエントリが有効であることを示すバリッド情報がセットされるとともに、カウンタ 38 のカウント値が信号 `reset` により初期値 “0” にリセットされる。

40

【0028】

部分命令（ベクトル命令による処理を分割した処理の実行指示）の発行が許可されると、ベクトル命令シーケンサ 12 は、すでに発行した部分命令の発行回数に応じた部分命令を発行する。すなわち、ベクトル命令シーケンサ 12 は、バッファ 32 に保持されているオペコードを有し、バッファ 33、34、35 に格納されているレジスタ番号にカウンタ 38 のカウント値を加算した値を演算対象のレジスタ番号とする部分命令を発行する。また、ベクトル命令シーケンサ 12 は、部分命令を発行する度に、カウンタ 38 のカウント値を 1 インクリメントする。ベクトル命令シーケンサ 12 は、この動作を繰り返して、レジスタ 36 に設定されたベクトル長回分の部分命令を発行する。

【0029】

50

そして、ベクトル長回分の部分命令を発行することで、カウンタ38のカウント値がレジスタ36に設定されたベクトル長の値に達すると、ベクトル命令シーケンサ12では、バリッド制御部37に信号resetが入力されてバリッド情報がリセットされる。これにより、バッファ32～35に保持されているエントリが無効にされ（インバリデイトされ）、このベクトル命令シーケンサ12への次のベクトル命令の供給が可能になる。

【0030】

図3に、本実施形態における演算スロット選択部13及びクロスバー14の構成例を示す。実行スロット選択部13は、部分命令の処理の内容や演算スロットへの部分命令の投入状況に応じて、部分命令について演算スロット15-1～15-4の割り当てを行う。本実施形態において、実行スロット選択部13は、次の(1)～(3)に基づく優先順位に従って、発行が要求される部分命令について演算スロット15-1～15-4を割り当てる。なお、この割り当ては1サイクル毎に行う。

(1) 乗算器を含む演算器21Aを有する第1の演算スロット15-1及び第2の演算スロット15-2には、乗算命令を優先的に投入する。

(2) ロード・ストアユニット22を有する第3の演算スロット15-3及び第4の演算スロット15-4には、ロード・ストア命令を優先的に投入する。

(3) (1)及び(2)による割り当ての結果、空いている演算スロットがあれば、先行するベクトル命令の部分命令から優先的に投入する。

【0031】

図3は、本実施形態における演算スロット選択部13及びクロスバー14の構成例を示す図である。図1において、CSP回路41、42、43は、入力される複数のリクエスト信号の中から最も優先順位が高い信号を選択するロジックをリングバッファ上に実現した回路である。CSP回路41、42、43は、リングバッファに保持された複数のリクエスト信号から、ヘッド（先頭）に一番近い有効なリクエストに対してアクノリッジを返すように動作する回路である。CSP回路の各々は、例えばDana S. Henry, Bradley C. Kuszmaul, Gabriel H. Loh, and Rahul Sami, "Circuits for Wide-Window Superscalar Processors," ISCA'00 Proceedings of the 27th annual international symposium on Computer architectureに記載されたcyclic segmented prefix回路と同等の回路である。

【0032】

CSP回路41-1、41-2は、乗算命令の発行に係るリクエスト信号ReqMについて処理を行う。CSP回路41-1、41-2には、ベクトル命令シーケンサ12A～12Dの各々から2スロット分に対応するリクエスト信号ReqMが入力可能である。乗算器を有する第1の演算スロットに対応するCSP回路41-1は、あるサイクルにて1つ以上のリクエスト信号ReqMがアサートされている場合に、その中で最も優先順位が高いリクエスト信号を選択する。また、乗算器を有する第2の演算スロットに対応するCSP回路41-2は、CSP回路41-1での選択結果を反映させてリクエスト信号ReqMについて処理を行う。すなわち、第2の演算スロットに対応するCSP回路41-2は、あるサイクルにて2つ以上のリクエスト信号ReqMがアサートされている場合に、その中で2番目に優先順位が高いリクエスト信号を選択する。

【0033】

CSP回路42-3、42-4は、ロード・ストア命令の発行に係るリクエスト信号ReqLSについて処理を行う。CSP回路42-3、42-4には、ベクトル命令シーケンサ12A～12Dの各々から2スロット分に対応するリクエスト信号ReqLSが入力可能である。ロード・ストアユニットを有する第3の演算スロットに対応するCSP回路42-3は、あるサイクルにて1つ以上のリクエスト信号ReqLSがアサートされている場合に、その中で最も優先順位が高いリクエスト信号を選択する。また、ロード・ストアユニットを有する第4の演算スロットに対応するCSP回路42-4は、CSP回路42-3での選択結果を反映させてリクエスト信号ReqLSについて処理を行う。すなわち、第4の演算スロットに対応するCSP回路42-4は、あるサイクルにて2つ以上の

リクエスト信号 R e q L S がアサートされている場合に、その中で 2 番目に優先順位が高いリクエスト信号を選択する。

【 0 0 3 4 】

C S P 回路 4 3 - 1、4 3 - 2、4 3 - 3、4 3 - 4 は、乗算命令とロード・ストア命令とを除いたその他の部分命令の発行に係るリクエスト信号 R e q O について処理を行う。C S P 回路 4 3 - 1、4 3 - 2、4 3 - 3、4 3 - 4 には、ベクトル命令シーケンサ 1 2 A ~ 1 2 D の各々から 4 スロット分に対応するリクエスト信号 R e q O が入力可能である。各演算スロットにそれぞれ対応する C S P 回路 4 3 - 1 ~ 4 3 - 4 は、リクエスト信号 R e q O がアサートされている場合に、すでに選択されているものを除いたアサートされているリクエスト信号 R e q O の中で最も優先順位が高いリクエスト信号を選択する。

10

【 0 0 3 5 】

セレクタ 4 4 - 1、4 4 - 2、4 4 - 3、4 4 - 4 は、優先順位つきのセレクタである。第 1 の演算スロットに対応するセレクタ 4 4 - 1 は、C S P 回路 4 1 - 1 の出力がアサートされていればそれを選択し、そうでない場合には C S P 回路 4 3 - 1 の出力を選択する。同様に、第 2 の演算スロットに対応するセレクタ 4 4 - 2 は、C S P 回路 4 1 - 2 の出力がアサートされていればそれを選択し、そうでない場合には C S P 回路 4 3 - 2 の出力を選択する。また、第 3 の演算スロットに対応するセレクタ 4 4 - 3 は、C S P 回路 4 2 - 3 の出力がアサートされていればそれを選択し、そうでない場合には C S P 回路 4 3 - 3 の出力を選択する。第 4 の演算スロットに対応するセレクタ 4 4 - 4 は、C S P 回路 4 2 - 4 の出力がアサートされていればそれを選択し、そうでない場合には C S P 回路 4 3 - 4 の出力を選択する。

20

【 0 0 3 6 】

セレクタ 4 4 - 1、4 4 - 2、4 4 - 3、4 4 - 4 は、選択結果に従って、どの部分命令を選択するかをインデックスをセレクタ 4 5 - 1、4 5 - 2、4 5 - 3、4 5 - 4 に供給する。また、セレクタ 4 4 - 1、4 4 - 2、4 4 - 3、4 4 - 4 での選択結果は、各ベクトル命令シーケンサ 1 2 A ~ 1 2 D にも供給され、カウンタのカウント値が更新される。このようにして、発行が要求される部分命令についての演算スロットの割り当てが行われる。図 3 に示した例では、C S P 回路 4 1、4 2、4 3 及びセレクタ 4 4 を含む回路が、演算スロット選択部 1 3 に対応する。

30

【 0 0 3 7 】

セレクタ 4 5 - 1、4 5 - 2、4 5 - 3、4 5 - 4 は、ベクトル命令シーケンサ 1 2 A ~ 1 2 D の各々から 4 スロット分に対応する部分命令が入力可能となっている。セレクタ 4 5 - 1 ~ 4 5 - 4 の各々は、対応するセレクタ 4 4 - 1 ~ 4 4 - 4 より供給されるインデックスに従って、部分命令を選択して対応する演算スロットに供給する。図 3 に示した例では、セレクタ 4 5 - 1 ~ 4 5 - 4 を含む回路が、クロスバー 1 4 に対応する。

【 0 0 3 8 】

次に、本実施形態によるプロセッサの動作について説明する。

命令フェッチステージ I F では、ベクトル命令が命令メモリ 1 1 から読み出されてベクトル命令シーケンサ 1 2 A ~ 1 2 D に格納される。命令デコードステージ I D では、ベクトル命令シーケンサ 1 2 A ~ 1 2 D が、格納されているベクトル命令を複数の部分命令に分割して、これら部分命令の発行を要求するリクエスト信号を演算スロット選択部 1 3 に出力する。そして、演算スロット選択部 1 3 が、ベクトル命令シーケンサ 1 2 A ~ 1 2 D からのリクエスト信号に基づいて、所定の優先順位に従い部分命令について演算スロット 1 5 - 1 ~ 1 5 - 4 の割り当てを行う。発行する部分命令は、演算スロット選択部 1 3 による割り当てに応じて、クロスバー 1 4 を介し各演算スロット 1 5 - 1 ~ 1 5 - 4 に供給される。

40

【 0 0 3 9 】

レジスタ読み出しステージ R R では、レジスタファイル 1 6 から読み出したデータ及びバイパスデータの内から、命令に係る処理に使用するデータがマルチプレクサ 1 8、1 9 により選択され、ソースデータ(ソースオペランド)としてパイプラインラッチ 2 0 に格

50

納される。演算実行ステージEXでは、命令が演算命令であれば、命令によって指示された演算処理が演算器21A、21Bにより実行され、演算結果がパイプラインラッチ23に格納される。また、演算実行ステージEXでは、命令がロード・ストア命令であれば、命令によって指示されたデータメモリ17へのアクセスがロード・ストアユニット22により実行され、データメモリ17に対してデータの読み出しや書き込みが行われる。レジスタ書き込みステージWBでは、演算実行ステージEXでの演算結果がレジスタファイル16の指定されたレジスタに書き込まれる。

【0040】

次に、本実施形態におけるプロセッサ及び一般的な技術での処理動作の例について説明する。なお、以下では、1サイクルに1ベクトル命令を命令メモリ11から読み出して発行可能であるとする。また、4本の演算スロットを有し、第1の演算スロットと第2の演算スロットとだけが乗算命令を実行可能であり、第3の演算スロットと第4の演算スロットとだけがロード・ストア命令を実行可能であるとする。

10

【0041】

また、本実施形態における演算スロット選択部は、前述のように以下の(1)~(3)の規定に従って、部分命令について演算スロットの割り当てを行うものとする。

(1) 第1の演算スロット及び第2の演算スロットには、乗算命令を優先的に投入する。このとき、第1の演算スロットから優先的に乗算命令を投入する。

(2) 第3の演算スロット及び第4の演算スロットには、ロード・ストア命令を優先的に投入する。このとき、第3の演算スロットから優先的にロード・ストア命令を投入する。

20

(3) (1)及び(2)による割り当ての結果、空いている演算スロットがあれば、先行するベクトル命令(乗算命令及びロード・ストア命令を除く)の部分命令から優先的に投入する。このとき、第1の演算スロット、第2の演算スロット、第3の演算スロット、第4の演算スロットの順で、第1の演算スロットから優先的に部分命令を投入する。

【0042】

図4は、関数呼び出しの先頭などにおいて、ベクトル命令(加算命令)Add 0 ベクトル命令(加算命令)Add 1 ベクトル命令(加算命令)Add 2 ベクトル命令(加算命令)Add 3の順にベクトル命令が発行される際の処理動作の例を説明するための図である。図4(A)に一般的な技術での処理動作を示し、図4(B)に本実施形態におけるプロセッサでの処理動作を示す。なお、ベクトル命令間に依存関係はなく、各ベクトル命令のベクトル長は6、すなわち各ベクトル命令は6分割されて実行されるものとする。

30

【0043】

図4(A)に示す一般的な技術での処理動作では、まずサイクル1にて、命令メモリ11から読み出されたベクトル命令Add 0が、第1の演算スロットに割り当てられる。そして、サイクル1~サイクル6の期間において、ベクトル命令Add 0の部分命令Add 0-0~Add 0-5に係る処理が、1サイクル毎に順次実行される。次のサイクル2にて、命令メモリ11から読み出されたベクトル命令Add 1が、第2の演算スロットに割り当てられる。そして、サイクル2~サイクル7の期間において、ベクトル命令Add 1の部分命令Add 1-0~Add 1-5に係る処理が、1サイクル毎に順次実行される。同様に、サイクル3にて命令メモリ11から読み出されたベクトル命令Add 2、及びサイクル4にて命令メモリ11から読み出されたベクトル命令Add 3が、第3の演算スロット及び第3の演算スロットにそれぞれ割り当てられる。そして、サイクル3~サイクル8の期間において、ベクトル命令Add 2の部分命令Add 2-0~Add 2-5に係る処理が順次実行され、サイクル4~サイクル9の期間において、ベクトル命令Add 3の部分命令Add 3-0~Add 3-5に係る処理が順次実行される。

40

【0044】

一方、本実施形態におけるプロセッサでは、図4(B)に示すように、サイクル1にて、命令メモリ11からベクトル命令Add 0が読み出されてベクトル命令シーケンサに格納され部分命令Add 0-0~Add 0-5に分割される。そして、サイクル1において、この部分

50

命令Add 0-0 ~ Add 0-5に係る演算スロットの割り当てを行い、サイクル1ではベクトル命令Add 0の部分命令Add 0-0 ~ Add 0-3に係る処理が並列して実行される。

【0045】

次のサイクル2にて、命令メモリ11からベクトル命令Add 1が読み出されてベクトル命令シーケンサに格納され部分命令Add 1-0 ~ Add 1-5に分割される。そして、サイクル2において、ベクトル命令Add 0の部分命令Add 0-4 ~ Add 0-5及びベクトル命令Add 1の部分命令Add 1-0 ~ Add 1-5に係る演算スロットの割り当てを行う。その結果、サイクル2では先行するベクトル命令Add 0の部分命令Add 0-4 ~ Add 0-5に係る処理が第1の演算スロット及び第2の演算スロットで、後続のベクトル命令Add 1の部分命令Add 1-0 ~ Add 1-1に係る処理が第3の演算スロット及び第4の演算スロットで、並列して実行される。なお、サイクル2において、ベクトル命令Add 0についての処理が終了するので、ベクトル命令Add 0が格納されていたベクトル命令シーケンサへの新たなベクトル命令の格納が可能になる。

10

【0046】

続く、サイクル3にて、命令メモリ11からベクトル命令Add 2が読み出されてベクトル命令シーケンサに格納され部分命令Add 2-0 ~ Add 2-5に分割される。そして、サイクル3において、ベクトル命令Add 1の部分命令Add 1-2 ~ Add 1-5及びベクトル命令Add 2の部分命令Add 2-0 ~ Add 2-5に係る演算スロットの割り当てを行う。その結果、サイクル3では先行するベクトル命令Add 1の部分命令Add 1-2 ~ Add 1-5に係る処理が第1 ~ 第4の演算スロットで並列して実行される。このサイクル3において、ベクトル命令Add 1についての処理が終了するので、ベクトル命令Add 1が格納されていたベクトル命令シーケンサへの新たなベクトル命令の格納が可能になる。

20

【0047】

次のサイクル4にて、命令メモリ11からベクトル命令Add 3が読み出されてベクトル命令シーケンサに格納され部分命令Add 3-0 ~ Add 3-5に分割される。そして、サイクル4において、ベクトル命令Add 2の部分命令Add 2-0 ~ Add 2-5及びベクトル命令Add 3の部分命令Add 3-0 ~ Add 3-5に係る演算スロットの割り当てを行う。その結果、サイクル4では先行するベクトル命令Add 2の部分命令Add 2-0 ~ Add 2-3に係る処理が第1 ~ 第4の演算スロットで並列して実行される。

30

【0048】

以下、同様にして、サイクル5においてベクトル命令Add 2の部分命令Add 2-4 ~ Add 2-5及びベクトル命令Add 3の部分命令Add 3-0 ~ Add 3-1に係る処理が第1 ~ 第4の演算スロットで並列して実行される。また、サイクル6においてベクトル命令Add 3の部分命令Add 3-2 ~ Add 3-5に係る処理が第1 ~ 第4の演算スロットで並列して実行される。

【0049】

図4(A)に示した一般的な技術での処理動作では、1つのベクトル命令に係る処理が固定された1つの演算スロットで実行されるので、関数呼び出しの先頭や末尾などにおいて、何ら処理をしていない演算スロットが存在することになる。一方、本実施形態におけるプロセッサでの処理動作では、1つのベクトル命令の部分命令を異なる演算スロットに発行可能にし、1つのベクトル命令に係る処理を異なる演算スロットで実行可能にする。これにより、図4(B)に示したように、例えば1つのベクトル命令の部分命令に係る処理を複数の演算スロットで並列に実行することが可能になる。また、例えば関数呼び出しの先頭や末尾などにおいても、何ら処理をしていない演算スロットが発生することを抑制することができる。したがって、プロセッサが有する演算スロット(演算リソース)を有効に活用して性能向上を図り、演算リソースの使用率を高め、高い性能を得ることができる。

40

【0050】

図5は、ベクトル命令(加算命令)Add 0 ベクトル命令(加算命令)Add 1 ベクトル命令(乗算命令)Mul 0 ベクトル命令(乗算命令)Mul 1の順にベクトル命令が発行されるとき処理動作の例を説明するための図である。図5(A)に一般的な技術での処

50

理動作を示し、図5(B)に本実施形態におけるプロセッサでの処理動作を示す。なお、ベクトル命令間に依存関係はなく、各ベクトル命令のベクトル長は6、すなわち各ベクトル命令は6分割されて実行されるものとする。

【0051】

図5(A)に示す一般的な技術での処理動作では、まずサイクル1にて、命令メモリ11から読み出されたベクトル命令Add 0が、第1の演算スロットに割り当てられる。そして、サイクル1～サイクル6の期間において、ベクトル命令Add 0の部分命令Add 0-0～Add 0-5に係る処理が、1サイクル毎に順次実行される。次のサイクル2にて、命令メモリ11から読み出されたベクトル命令Add 1が、第2の演算スロットに割り当てられる。そして、サイクル2～サイクル7の期間において、ベクトル命令Add 1の部分命令Add 1-0～Add 1-5に係る処理が、1サイクル毎に順次実行される。

10

【0052】

次に命令メモリ11からベクトル命令Mul 0が読み出されることとなるが、乗算命令は第1の演算スロット及び第2の演算スロットだけで実行可能であり、それらの内の少なくとも1つの演算スロットが使用可能となるのはサイクル7以降である。したがって、図5(A)に示す一般的な技術での処理動作では、サイクル7において、命令メモリ11から読み出されたベクトル命令Mul 0が、第1の演算スロットに割り当てられる。そして、サイクル7～サイクル12の期間において、ベクトル命令Mul 0の部分命令Mul 0-0～Mul 0-5に係る処理が、1サイクル毎に順次実行される。同様に、サイクル8にて命令メモリ11から読み出されたベクトル命令Mul 1が、第2の演算スロットに割り当てられ、サイクル8～サイクル13の期間において、ベクトル命令Mul 1の部分命令Mul 1-0～Mul 1-5に係る処理が順次実行される。

20

【0053】

一方、本実施形態におけるプロセッサでは、図5(B)に示すように、サイクル1にて、命令メモリ11からベクトル命令Add 0が読み出されてベクトル命令シーケンサに格納され部分命令Add 0-0～Add 0-5に分割される。そして、サイクル1において、この部分命令Add 0-0～Add 0-5に係る演算スロットの割り当てを行い、サイクル1ではベクトル命令Add 0の部分命令Add 0-0～Add 0-3に係る処理が並列して実行される。

【0054】

次のサイクル2にて、命令メモリ11からベクトル命令Add 1が読み出されてベクトル命令シーケンサに格納され部分命令Add 1-0～Add 1-5に分割される。そして、サイクル2において、ベクトル命令Add 0の部分命令Add 0-4～Add 0-5及びベクトル命令Add 1の部分命令Add 1-0～Add 1-5に係る演算スロットの割り当てを行う。その結果、サイクル2では先行するベクトル命令Add 0の部分命令Add 0-4～Add 0-5に係る処理が第1の演算スロット及び第2の演算スロットで、後続のベクトル命令Add 1の部分命令Add 1-0～Add 1-1に係る処理が第3の演算スロット及び第4の演算スロットで、並列して実行される。なお、サイクル2において、ベクトル命令Add 0についての処理が終了するので、ベクトル命令Add 0が格納されていたベクトル命令シーケンサへの新たなベクトル命令の格納が可能になる。

30

【0055】

続く、サイクル3にて、命令メモリ11からベクトル命令Mul 0が読み出されてベクトル命令シーケンサに格納され部分命令Mul 0-0～Mul 0-5に分割される。そして、サイクル3において、ベクトル命令Add 1の部分命令Add 1-2～Add 1-5及びベクトル命令Mul 0の部分命令Mul 0-0～Mul 0-5に係る演算スロットの割り当てを行う。ここで、本実施形態において、乗算命令は第1の演算スロット及び第2の演算スロットだけが実行可能であり、第1の演算スロット及び第2の演算スロットには、乗算命令を優先的に投入することとなっている。したがって、サイクル3ではベクトル命令Mul 0の部分命令Mul 0-0～Mul 0-1に係る処理が第1の演算スロット及び第2の演算スロットで実行され、ベクトル命令Add 1の部分命令Add 1-2～Add 1-3に係る処理が第3の演算スロット及び第4の演算スロットで実行される。

40

50

【0056】

次のサイクル4にて、命令メモリ11からベクトル命令Mul1が読み出されてベクトル命令シーケンサに格納され部分命令Mul1-0~Mul1-5に分割される。そして、サイクル4において、ベクトル命令Add1の部分命令Add1-4~Add1-5、ベクトル命令Mul0の部分命令Mul1-2~Mul1-5及びベクトル命令Mul1の部分命令Mul1-0~Mul1-5に係る演算スロットの割り当てを行う。その結果、サイクル3ではベクトル命令Mul0の部分命令Mul0-2~Mul0-3に係る処理が第1の演算スロット及び第2の演算スロットで実行され、ベクトル命令Add1の部分命令Add1-4~Add1-5に係る処理が第3の演算スロット及び第4の演算スロットで実行される。このサイクル4において、ベクトル命令Add1についての処理が終了するので、ベクトル命令Add1が格納されていたベクトル命令シーケンサへの新たなベクトル命令の格納が可能になる。

10

【0057】

次のサイクル5では、ベクトル命令Mul0の部分命令Mul0-4~Mul0-5及びベクトル命令Mul1の部分命令Mul1-0~Mul1-5に係る演算スロットの割り当てが行われ、サイクル5においてベクトル命令Mul0の部分命令Mul0-4~Mul0-5に係る処理が第1の演算スロット及び第2の演算スロットで実行される。以下、同様にして、サイクル6においてベクトル命令Mul1の部分命令Mul1-0~Mul1-1に係る処理が第1の演算スロット及び第2の演算スロットで実行される。また、サイクル7、8において、ベクトル命令Mul1の部分命令Mul1-2~Mul1-3に係る処理及び部分命令Mul1-4~Mul1-5に係る処理のそれぞれが第1の演算スロット及び第2の演算スロットで実行される。

20

【0058】

本実施形態におけるプロセッサでの処理動作では、1つのベクトル命令の部分命令を異なる演算スロットに発行可能にし、1つのベクトル命令に係る処理を異なる演算スロットで実行可能にする。これにより、図5(B)に示したように、例えば1つのベクトル命令の部分命令に係る処理を複数の演算スロットで並列に実行することが可能になり、プロセッサが有する演算スロット(演算リソース)を有効に活用して性能向上を図り、演算リソースの使用率を高め、高い性能を得ることができる。

【0059】

また、乗算命令やロード・ストア命令など特定の演算リソースを有する演算スロットだけで実行可能なベクトル命令である場合、そのベクトル命令の部分命令が特定の演算リソースを有する演算スロットに優先的に投入される。これにより、特定の演算リソースを必要とするベクトル命令を優先して実行することが可能になり、プロセッサが有する演算リソースを有効に活用して性能向上を図り、演算リソースの使用率を高め、高い性能を得ることができる。

30

【0060】

なお、本実施形態では、4つのベクトル命令シーケンサ及び4つの演算スロットを有する例を一例として示したが、これに限定されるものではない。本実施形態におけるプロセッサが有する、ベクトル命令シーケンサの数及び演算スロットの数は、任意である。また、ベクトル命令シーケンサと演算スロットとの対応関係もないので、ベクトル命令シーケンサの数と演算スロットの数は、同数でなくとも良い。

40

【0061】

なお、前記実施形態は、何れも本発明を実施するにあたっての具体化のほんの一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、また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】

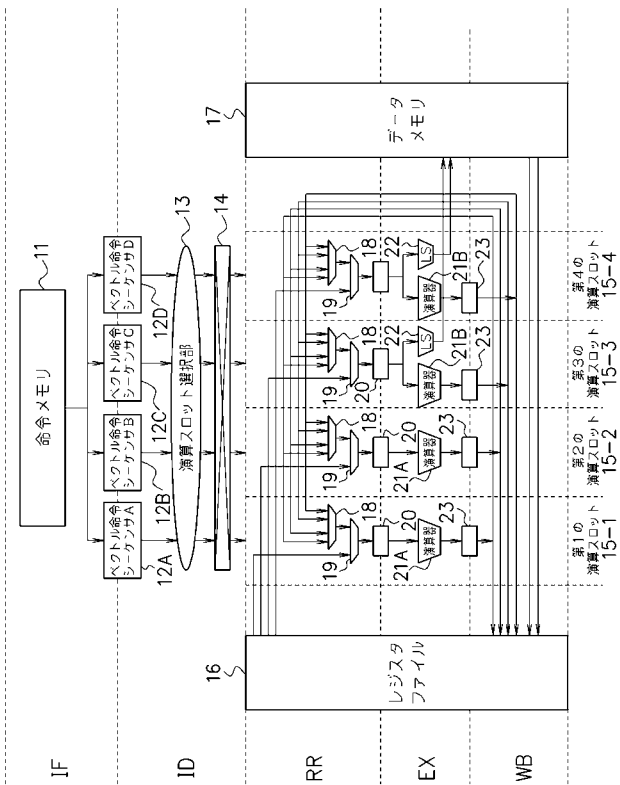
【0062】

- 11 命令メモリ
- 12 ベクトル命令シーケンサ
- 13 演算スロット選択部

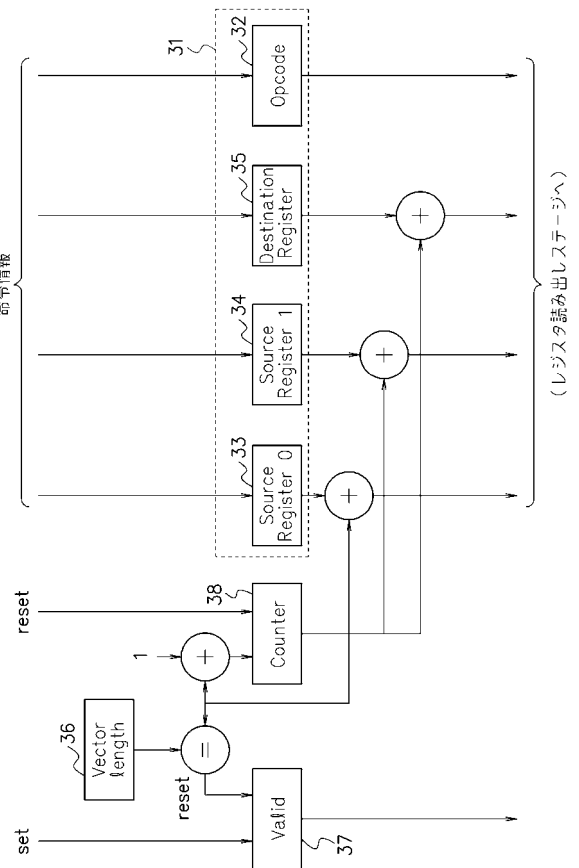
50

- 14 クロスバー
- 15 演算スロット
- 16 レジスタファイル
- 17 データメモリ
- 18、19 マルチプレクサ
- 20、23 パイプラインラッチ
- 21A、21B 演算器
- 22 ロード・ストアユニット

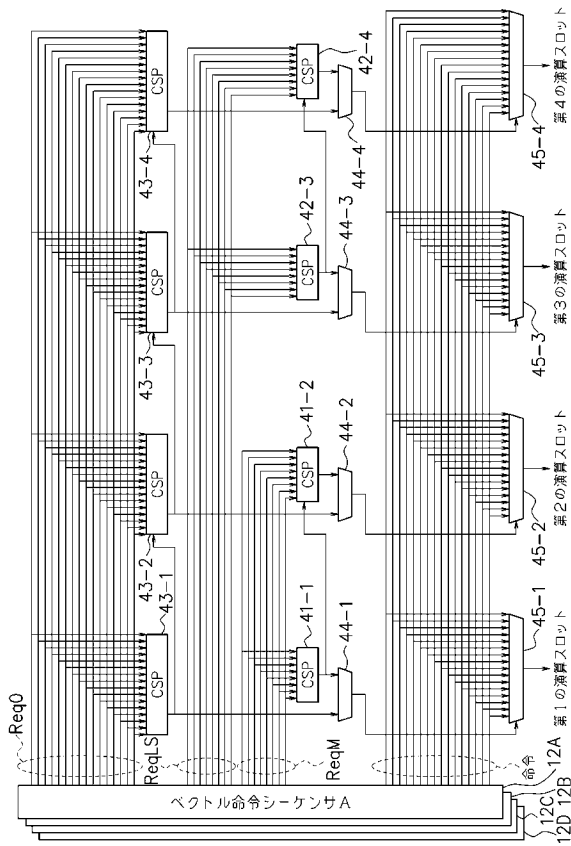
【 図 1 】



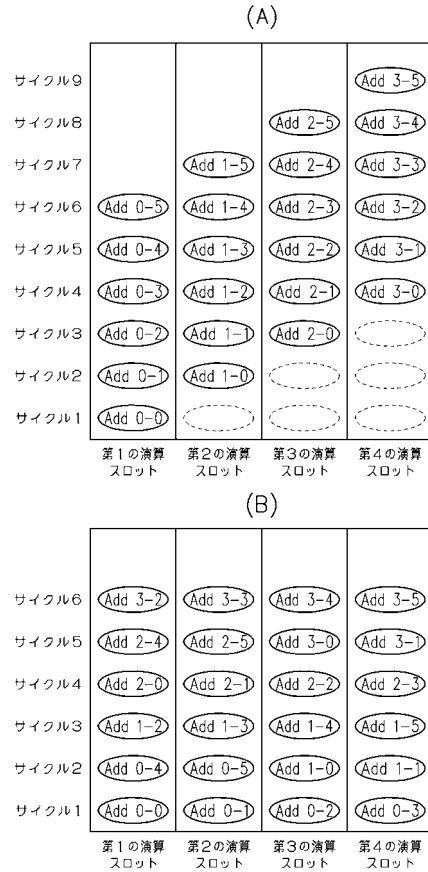
【 図 2 】



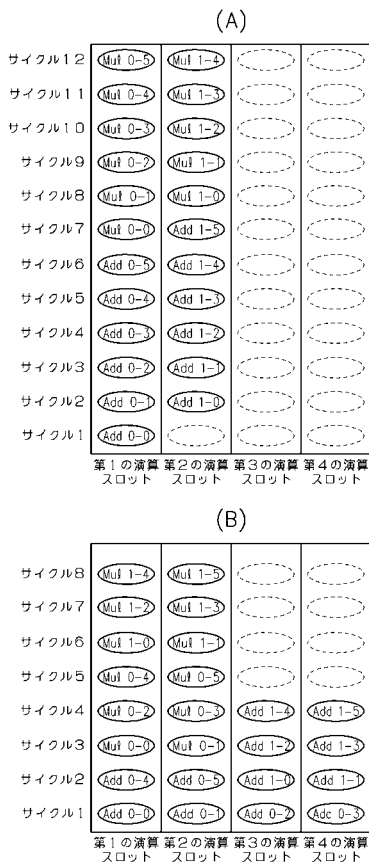
【図3】



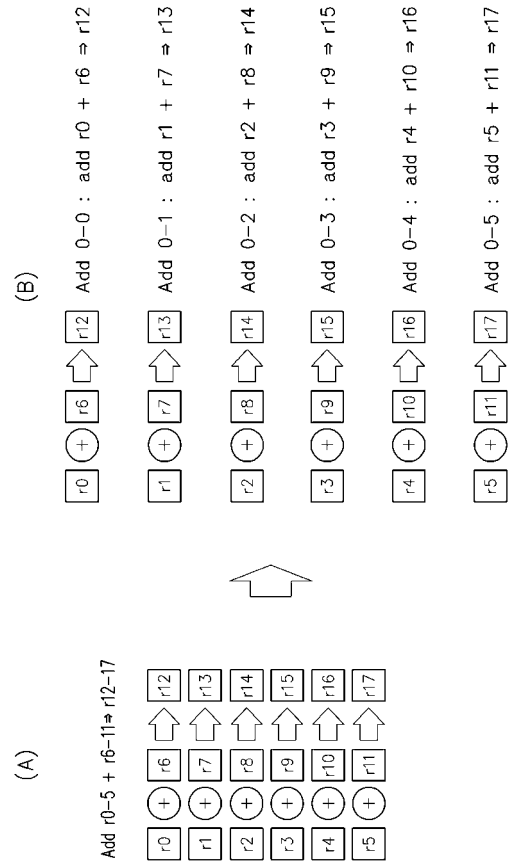
【図4】



【図5】



【図6】



【 図 7 】

