

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6094356号
(P6094356)

(45) 発行日 平成29年3月15日(2017.3.15)

(24) 登録日 平成29年2月24日(2017.2.24)

(51) Int.Cl.

F I

G O 6 F 9/38 (2006.01)
G O 6 F 9/30 (2006.01)G O 6 F 9/38 3 7 O C
G O 6 F 9/38 3 1 O X
G O 6 F 9/38 3 7 O B
G O 6 F 9/30 3 5 O F

請求項の数 5 (全 22 頁)

(21) 出願番号 特願2013-89479 (P2013-89479)
(22) 出願日 平成25年4月22日(2013.4.22)
(65) 公開番号 特開2014-215624 (P2014-215624A)
(43) 公開日 平成26年11月17日(2014.11.17)
審査請求日 平成28年1月13日(2016.1.13)(73) 特許権者 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 100099759
弁理士 青木 篤
(74) 代理人 100119987
弁理士 伊坪 公一
(74) 代理人 100081330
弁理士 樋口 外治
(74) 代理人 100114177
弁理士 小林 龍
(72) 発明者 吉村 和浩
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

最終頁に続く

(54) 【発明の名称】 演算処理装置

(57) 【特許請求の範囲】

【請求項1】

演算処理装置であって、

演算を実行する演算器と、

ストリーム処理を実行するストリームエンジンと、

前記演算器および前記ストリームエンジンに命令を発行する命令発行部と、を有し、

前記命令発行部が前記ストリームエンジンに発行する命令は、ステップ命令により実現される複数の操作のうちのいずれかの操作を実現するためのマイクロ命令であり、

前記マイクロ命令は、前記ストリームエンジンの各パイプラインステージの操作を制御するための短ビット長のマイクロ命令であり、

前記各パイプラインステージは、それぞれ対応する前記マイクロ命令に従って独立して処理を実行し、

前記演算処理装置は、前記ストリームエンジンの各パイプラインステージの操作を制御するためのマイクロ命令が埋め込まれたV L I W命令を処理する、

ことを特徴とする演算処理装置。

【請求項2】

前記ストリームエンジンは、

メモリからデータを読み出す読出回路と、

前記読み出したデータに対してストリーム処理を実行する実行回路と、

前記ストリーム処理された演算結果を前記メモリに書き込む書込回路と、を含む、

10

20

ことを特徴とする請求項 1 に記載の演算処理装置。

【請求項 3】

前記読出回路は、POP 部および第 1 レジスタを含み、

前記 POP 部は、前記メモリにおける、先頭アドレスおよびストリーム長で示された第 1 メモリ部からデータを読み出して前記第 1 レジスタに格納する、

ことを特徴とする請求項 2 に記載の演算処理装置。

【請求項 4】

前記実行回路は、EXEC 部および第 2 レジスタを含み、

前記 EXEC 部は、前記第 1 レジスタに格納されたデータに対してストリーム処理を実行し、前記ストリーム処理された演算結果を前記第 2 レジスタに格納する、

ことを特徴とする請求項 3 に記載の演算処理装置。

【請求項 5】

前記書込回路は、PUSH 部を含み、

前記 PUSH 部は、前記第 2 レジスタに格納された演算結果を、前記メモリにおける、先頭アドレスおよびストリーム長で示された第 2 メモリ部に書き込む、

ことを特徴とする請求項 4 に記載の演算処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本明細書で言及する実施例は、演算処理装置に関する。

【背景技術】

【0002】

近年、スマートフォンやタブレットコンピュータといった携帯端末の通信量の増大に伴って、より高速な無線通信方式が注目されている。このような高速無線通信方式として、例えば、LTE (Long Term Evolution) が普及されてきており、さらに高性能な次世代移動通信システムの LTE アドバンスド (LTE-Advanced) も標準化が完了し、実用化に向けて様々な提案がなされている。

【0003】

ところで、例えば、LTE アドバンスドを適用する場合、無線通信ベースバンド処理として、膨大な行列演算処理を行うことになる。これは、LTE アドバンスドに限ったことではなく、WiMAX 2 (Worldwide Interoperability for Microwave Access 2) や現在使用されている方式を含めて様々な無線通信方式 (規格) でも同様である。

【0004】

一般的に、無線通信ベースバンド処理において、通信速度の向上に比例して膨大な行列演算を行うことになるが、例えば、上述した LTE アドバンスドでは、行列演算が全体の演算量の多くを占めている。

【0005】

行列演算処理 (ストリーム処理のひとつ) を高速に実行するためには、行列データが格納されたメモリと演算器を直列に接続し、メモリから読み出したデータに対して、行列演算を行ない、演算結果をメモリに書き出すストリームエンジンが適している。

【0006】

そこで、例えば、LTE アドバンスドの無線通信ベースバンド処理を行う演算処理装置 (演算処理システム) として、汎用プロセッサであるベースプロセッサと、ストリームエンジンを持つコプロセッサを組み合わせたものが提案されている。

【0007】

ところで、従来、ベースプロセッサと、ストリームエンジンを持つコプロセッサを組み合わせた演算処理システムとしては、様々なものが提案されている。

【先行技術文献】

【特許文献】

【0008】

10

20

30

40

50

【特許文献1】特開2011-197774号公報

【特許文献2】特開平08-069377号公報

【発明の概要】

【発明が解決しようとする課題】

【0009】

前述したように、無線通信ベースバンド処理を行う演算処理システムとして、ベースプロセッサと、ストリームエンジンを持つコプロセッサを組み合わせたものが提案されている。

【0010】

このような演算処理システムでは、例えば、コプロセッサ命令であるストリーム命令を実行するとき、ベースプロセッサは、ハンドシェイクにより、コプロセッサの状態監視、データ転送および実行制御等を行うため、オーバーヘッドが生じる。このオーバーヘッドは、例えば、通信サイクルオーバーヘッドと呼ばれている。

【0011】

さらに、例えば、コプロセッサのストリームエンジンがストリーム処理を実行中に割り込みが発生した場合、割り込み処理は、ストリーム処理の実行が完了するまで待機して行うことになる。

【0012】

すなわち、割り込み発生時にコプロセッサがビジー状態の場合、ベースプロセッサはコプロセッサがアイドル状態になるまで待機することになり、通信サイクルオーバーヘッドはさらに増加してしまう。

【課題を解決するための手段】

【0013】

一実施形態によれば、演算処理装置であって、演算を実行する演算器と、ストリーム処理を実行するストリームエンジンと、前記演算器および前記ストリームエンジンに命令を発行する命令発行部と、を有する演算処理装置が提供される。前記命令発行部が前記ストリームエンジンに発行する命令は、ステップ命令により実現される複数の操作のうちのいずれかの操作を実現するためのマイクロ命令であり、前記マイクロ命令は、前記ストリームエンジンの各パイプラインステージの操作を制御するための短ビット長のマイクロ命令である。前記各パイプラインステージは、それぞれ対応する前記マイクロ命令に従って独立して処理を実行し、前記演算処理装置は、前記ストリームエンジンの各パイプラインステージの操作を制御するためのマイクロ命令が埋め込まれたVLIW命令を処理する。

【発明の効果】

【0014】

開示の演算処理装置は、サイクルオーバーヘッドを低減して処理を高速化することができるという効果を奏する。

【図面の簡単な説明】

【0015】

【図1】図1は、演算処理装置の一例を示すブロック図である。

【図2】図2は、本実施例に係る演算処理装置の一例を示すブロック図である。

【図3】図3は、本実施例の演算処理装置における動作を説明するための図である。

【図4】図4は、本実施例の演算処理装置におけるストリームエンジンの停止動作を説明するための図である。

【図5】図5は、図4を参照して説明したストリームエンジンの停止動作による効果の一例を説明するための図である。

【図6】図6は、本実施例の演算処理装置における読出回路の動作の一例を説明するための図である。

【図7】図7は、本実施例の演算処理装置における読出回路の動作の他の例を説明するための図である。

【図8】図8は、本実施例の演算処理装置における実行回路の動作の一例を説明するため

10

20

30

40

50

の図である。

【図 9】図 9 は、本実施例の演算処理装置における実行回路の動作の他の例を説明するための図である。

【図 10】図 10 は、本実施例の演算処理装置における書込回路の動作の一例を説明するための図である。

【図 11】図 11 は、本実施例の演算処理装置における書込回路の動作の他の例を説明するための図である。

【図 12】図 12 は、本実施例の演算処理装置におけるパラメータ情報の一例を説明するための図である。

【図 13】図 13 は、本実施例の演算処理装置におけるステップ命令を説明するための図 (その 1) である。

10

【図 14】図 14 は、本実施例の演算処理装置におけるステップ命令を説明するための図 (その 2) である。

【図 15】図 15 は、本実施例の演算処理装置におけるステップ命令の変形を説明するための図である。

【図 16】図 16 は、本実施例の演算処理装置におけるマイクロ命令を説明するための図 (その 1) である。

【図 17】図 17 は、本実施例の演算処理装置におけるマイクロ命令を説明するための図 (その 2) である。

【図 18】図 18 は、本実施例の演算処理装置におけるマイクロ命令によるアクセス制御を説明するための図である。

20

【図 19】図 19 は、本実施例の演算処理装置におけるマイクロ命令を V L I W 命令に埋め込む様子を示す図である。

【図 20】図 20 は、図 19 に示す V L I W 命令のプロローグ処理を説明するための図である。

【図 21】図 21 は、図 19 に示す V L I W 命令のエピローグ処理を説明するための図である。

【発明を実施するための形態】

【0016】

まず、演算処理装置の実施例を詳述する前に、図 1 を参照して、演算処理装置の一例およびその問題点を説明する。

30

【0017】

図 1 は、演算処理装置の一例を示すブロック図であり、汎用プロセッサであるベースプロセッサと、ストリームエンジンを持つコプロセッサを組み合わせた演算処理装置 (演算処理システム) を示すものである。

【0018】

図 1 において、参照符号 I F は命令読出 (Instruction Fetch) ステージ、I D は命令解釈 (Instruction Decode) ステージ、そして、R R / I I はレジスタ読出 (Register Read) および命令発行 (Instruction Issue) ステージを示す。

【0019】

40

また、参照符号 E X は実行 (EXecution) ステージ、M A はメモリアクセス (Memory Access) ステージ、そして、R W はレジスタ書込 (Register Write) ステージを示す。図 1 に示す演算処理システムは、例えば、汎用プロセッサであるベースプロセッサ 100 と、ストリームエンジン 200 を含むコプロセッサ 300 を有する。

【0020】

ベースプロセッサ 100 において、I F ステージでは、命令読出部 101 が命令メモリ 108 から命令をフェッチ (読み出し) し、また、I D ステージでは、命令解釈部 102 が命令読出部 101 で読み出された命令を受け取ってデコード (解釈) する。

【0021】

R R / I I ステージでは、レジスタ読出部 103 がレジスタ 110 のリード (読み出し)

50

を行うと共に、命令発行部 104 が命令解釈部 102 で解釈された命令を演算器 105 へ発行する。

【0022】

EXステージでは、演算器 105 が命令発行部 104 から発行された命令に従った演算を実行し、また、MAステージでは、メモリアクセス部 106 がメモリ(データメモリ) 109 に対するロード(読み出し)またはストア(書き込み)のアクセスを行う。

【0023】

RWステージでは、レジスタ書込部 107 が演算器 105 による演算結果、または、データメモリ 109 からロードされたデータをレジスタ 110 に書き込む。

【0024】

ここで、図 1 の参照符号 P 100 で示されるように、ベースプロセッサ 100 では、レジスタ - メモリ間またはレジスタ - 演算器間の処理を 1 命令としてパイプライン実行するようになっている。

【0025】

コプロセッサ 300 において、IFステージでは、命令読出部 301 が命令メモリ 108 から命令を読み出し、また、IDステージでは、命令解釈部 302 が命令読出部 301 で読み出された命令を受け取って解釈する。

【0026】

RR/IIステージでは、レジスタ読出部 303 がレジスタ 310 のリードを行うと共に、命令発行部 304 が命令解釈部 302 で解釈された命令をストリームエンジン 200 へ発行する。ここで、ストリームエンジン 200 は、演算器 205、および、データメモリ 400 に対するロードまたはストアのアクセスを行うメモリアクセス部 206 を含む。

【0027】

図 1 の参照符号 P 200 で示されるように、命令発行部 304 からストリームエンジン 200 への命令はストリーム命令であり、1 ストリーム命令が発行されると、メモリ - 演算器間の 1 ストリーム処理が完了するまでパイプライン実行するようになっている。

【0028】

すなわち、EXおよびMAステージでは、ストリームエンジン 200 における演算器 205 およびメモリアクセス部 206 が命令発行部 304 から発行されたストリーム命令に従ってストリーム処理が完了するまで処理を行う。なお、RWステージでは、レジスタ書込部 307 がストリームエンジン 200 によりストリーム処理されたデータ(演算結果)をレジスタ 310 に書き込む。

【0029】

ここで、図 1 において、参照符号 P 150 は、ベースプロセッサ 100 によるコプロセッサ 300 の処理を示し、例えば、コプロセッサ 300 のストリーム命令発行によりコプロセッサ 300 とハンドシェイクする処理を示す。すなわち、ベースプロセッサ 100 は、例えば、コプロセッサ 300 の状態を監視し、コプロセッサ 300 の実行制御を行い、そして、コプロセッサ 300 へのデータ転送を制御する。

【0030】

図 1 を参照して説明したベースプロセッサ 100 と、ストリームエンジン 200 を持つコプロセッサ 300 を組み合わせた演算処理システムは、ストリームエンジン 200 によりストリーム処理を実行するとき、サイクルオーバーヘッドの問題がある。

【0031】

すなわち、コプロセッサ命令であるストリーム命令を実行するとき、ベースプロセッサ 100 は、ハンドシェイクにより、コプロセッサ 300 の状態を監視し、コプロセッサ 300 との間のデータ転送を行ない、そして、コプロセッサ 300 の実行を制御する。

【0032】

そのため、ベースプロセッサ 100 とコプロセッサ 300 の間には、オーバーヘッド(通信サイクルオーバーヘッド)が生じる。また、例えば、コプロセッサ 300 のストリームエンジン 200 がストリーム処理を実行中に割り込みが発生した場合、そのストリーム処

10

20

30

40

50

理の実行が完了するまで待機することになり、通信サイクルオーバーヘッドはさらに増加する。

【0033】

以下、本実施例の演算処理装置を、添付図面を参照して詳述する。図2は、本実施例に係る演算処理装置の一例を示すブロック図である。図2と上述した図1の比較から明らかなように、図2に示す演算処理装置(プロセッサ)1は、図1におけるベースプロセッサ100に対応する構成を含み、さらに、ストリームエンジン2を内蔵している。

【0034】

すなわち、図2に示されるように、プロセッサ1は、レジスタ10、命令読出部11、命令解釈部12、レジスタ読出部13、命令発行部14、演算器15、メモリアクセス部16、レジスタ書込部17、命令メモリ18およびデータメモリ19を含む。ここで、命令発行部14は、演算器15に命令を発行するだけでなく、ストリームエンジン2に対しても命令(例えば、ステップ命令)を発行するようになっている。

【0035】

ストリームエンジン2は、データメモリ4からデータを読み出してレジスタ221、222に書き込むPOP部21、レジスタ221、222に書き込まれたデータに対してストリーム処理を実行してレジスタ24に書き込むEXEC部23を含む。さらに、ストリームエンジン2は、レジスタ24に書き込まれたデータをデータメモリ4に書き込むPUSH部25も含む。

【0036】

図2において、参照符号IF、ID、RR/II、EX、MAおよびRWは、それぞれ図1を参照して説明したのと同様のステージを示す。

【0037】

すなわち、IFステージでは、命令読出部11が命令メモリ18から命令をフェッチ(読み出)し、また、IDステージでは、命令解釈部102が命令読出部101でフェッチされた命令を受け取ってデコード(解釈)する。

【0038】

RR/IIステージでは、レジスタ読出部13がレジスタ10のリード(読み出し)を行うと共に、命令発行部14が命令解釈部12で解釈された命令を演算器15およびストリームエンジン2へ発行する。

【0039】

EXステージでは、演算器15が命令発行部14から発行された命令に従った演算を実行すると共に、ストリームエンジン2が命令発行部14から発行された命令に従ったストリーム処理を実行する。ここで、命令発行部14からストリームエンジン2への命令は、上述したように、ステップ命令とされている。

【0040】

MAステージでは、メモリアクセス部16がメモリ(データメモリ)19に対するロードまたはストアのアクセスを行う。さらに、MAステージでは、ストリームエンジン2(POP部21またはPUSH部25)がメモリ(データメモリ)4に対するロード(読み出し)またはストア(書き込み)のアクセスを行う。

【0041】

RWステージでは、レジスタ書込部17が演算器15による演算結果またはデータメモリ19からロードされたデータをレジスタ10に書き込むと共に、レジスタ書込部17がストリームエンジン2によりストリーム処理されたデータをレジスタ10に書き込む。

【0042】

図3は、本実施例の演算処理装置における動作を説明するための図である。図3の参照符号P1と、前述した図1の参照符号P100の比較から明らかなように、図1におけるベースプロセッサ100の対応個所では、レジスタ-メモリ間またはレジスタ-演算器間の処理を1命令としてパイプライン実行するようになっている。

【0043】

10

20

30

40

50

また、図 3 の参照符号 P 2 1 ~ P 2 3 に示されるように、プロセッサ 1 に内蔵されたストリームエンジン 2 では、命令発行部 1 4 から発行されたステップ命令に従って、ステップ毎の処理を実行する。

【 0 0 4 4 】

ここで、処理 P 2 1 は、ストリームエンジン 2 の P O P 部 2 1 がデータメモリ 4 からデータを読み出してレジスタ 2 2 1 , 2 2 2 に書き込む処理である。また、処理 P 2 2 は、E X E C 部 2 3 がレジスタ 2 2 1 , 2 2 2 に書き込まれたデータに対してストリーム処理を実行してレジスタ 2 4 に書き込む処理である。

【 0 0 4 5 】

さらに、処理 P 2 3 は、P U S H 部 2 5 がレジスタ 2 4 に書き込まれたデータをデータメモリ 4 に書き込む処理である。これらの処理 P 2 1 ~ P 2 3 は、命令発行部 1 4 から発行されたステップ命令に従ってパイプライン実行される。

【 0 0 4 6 】

なお、本明細書では、ストリームエンジン 2 は、3 つの処理 P 2 1 ~ P 2 3 を 3 つのステップ命令で処理 (3 ステップ命令で 1 回転) する場合を例として示している。しかしながら、これは単なる例であり、4 つ以上の処理で 1 回転の処理とし、この 1 回転の処理を多数回繰り返すことでストリーム処理を実行してもよいのはいうまでもない。

【 0 0 4 7 】

図 4 は、本実施例の演算処理装置におけるストリームエンジンの停止動作を説明するための図である。例えば、プロセッサ 1 に内蔵されたストリームエンジン 2 がストリーム処理を実行中に割り込みが発生した場合、命令発行部 1 4 がストリームエンジン 2 に対するステップ命令の発行を停止する。

【 0 0 4 8 】

このように、命令発行部 1 4 がストリームエンジン 2 に対するステップ命令の発行を停止すると、ストリームエンジン 2 における全ての処理 P 2 1 ~ P 2 3 が停止する。すなわち、P O P 部 2 1 は、データメモリ 4 からデータを読み出してレジスタ 2 2 1 , 2 2 2 に書き込む処理 P 2 1 を停止する。

【 0 0 4 9 】

また、E X E C 部 2 3 は、レジスタ 2 2 1 , 2 2 2 に書き込まれたデータに対してストリーム処理を実行してレジスタ 2 4 に書き込む処理 P 2 2 を停止する。そして、P U S H 部 2 5 は、レジスタ 2 4 に書き込まれたデータをデータメモリ 1 9 に書き込む処理 P 2 3 を停止する。

【 0 0 5 0 】

このように、本実施例の演算処理装置は、ストリームエンジン 2 の動作をステップ命令により細粒度で制御しているため、ストリーム処理中に割り込みが発生した場合には、ストリーム処理を直ちに停止して割り込み処理を行うことができる。

【 0 0 5 1 】

すなわち、本実施例の演算処理装置によれば、例えば、割り込み発生時にステップ命令の発行を止めることにより、ストリームエンジン 2 を直ちに停止することができる。換言すると、本実施例の演算処理装置によれば、命令発行を止めた後、ストリームエンジン 2 の各パイプラインステージ (処理 P 2 1 ~ P 2 3) は自律的に停止することができ、サイクルオーバーヘッドを低減して処理を高速化することが可能になる。

【 0 0 5 2 】

図 5 は、図 4 を参照して説明したストリームエンジンの停止動作による効果の一例を説明するための図であり、図 5 (a) は、前述した図 1 に示す演算処理システムによる動作を示し、図 5 (b) は、図 4 を参照して説明した演算処理装置による動作を示す。

【 0 0 5 3 】

ここで、前提として、1 ストリーム処理のサイクル数 (クロックサイクル数) を 2 0 0 サイクル、演算データパスのレイテンシを 1 0 サイクル、そして、1 ストリーム処理に使用するパラメータ情報のビット幅を 3 2 0 ビットとする。

10

20

30

40

50

【 0 0 5 4 】

また、外部 - メモリ間のデータ転送はストリーム処理とオーバーラップ動作し、データ転送サイクルは隠蔽されているものとする。さらに、図 5 (a)において、ベースプロセッサ 1 0 0 - コプロセッサ 3 0 0 間のデータパスを 3 2 ビットとし、パラメータ情報は、ベースプロセッサ 1 0 0 からコプロセッサ 3 0 0 へ 1 0 サイクルで転送されるものとする。

【 0 0 5 5 】

従って、図 5 (a)では、通信サイクルオーバーヘッドは、例えば、1 0 [サイクル](データ転送) + 1 0 [サイクル](演算データパス) = 2 0 [サイクル]となる。

【 0 0 5 6 】

また、図 5 (b)において、データパスは密結合されているため、パラメータ情報は 1 サイクルで転送されるとする。なお、本明細書において、密結合とは、バスレベルで結合された複数のプロセッサが共通のメモリにアクセスするというのではなく、共通の命令発行部 1 4 が演算器 1 5 およびストリームエンジン 2 に対して命令を発行して制御することを意味する。

【 0 0 5 7 】

従って、図 5 (b)では、通信サイクルオーバーヘッドは、例えば、1 [サイクル](データ転送) + 1 0 [サイクル](演算データパス) = 1 1 [サイクル]となる。

【 0 0 5 8 】

まず、図 5 (a)に示されるように、図 1 に示す演算処理システムにおいて、例えば、3 回目のストリーム処理(A 2)における 5 0 サイクル目で割り込みが発生した場合、その 3 回目のストリーム処理を全て完了した後に、別のストリーム処理(B 0)を実行する。

【 0 0 5 9 】

そのため、演算処理システムでは、別のストリーム処理(B 0)を完了するまでに、 $2\ 0\ 0 + 2\ 0 + 2\ 0\ 0 + 2\ 0 + 5\ 0 + 1\ 5\ 0 + 2\ 0 + 2\ 0\ 0 = 8\ 6\ 0$ [サイクル]だけ要することになる。

【 0 0 6 0 】

一方、図 4 を参照して説明した本実施例の演算処理装置(プロセッサ) 1 において、例えば、3 回目のストリーム処理(A 2)における 5 0 サイクル目で割り込みが発生した場合、直ちにその 3 回目のストリーム処理を止めて別のストリーム処理(B 0)を実行する。

【 0 0 6 1 】

そのため、本実施例のプロセッサ 1 では、別のストリーム処理(B 0)を完了するまでに、 $2\ 0\ 0 + 1\ 1 + 2\ 0\ 0 + 1\ 1 + 5\ 0 + 1\ 1 + 2\ 0\ 0 = 6\ 8\ 3$ [サイクル]だけ要することになる。

【 0 0 6 2 】

すなわち、本実施例のプロセッサ 1 によれば、同じ処理を行うのに、8 6 0 [サイクル]から 6 8 3 [サイクル]へ 1 7 7 [サイクル]だけ処理を高速化することが可能なことが分かる。

【 0 0 6 3 】

なお、図 5 は、単なるストリーム処理の一例を説明するためのもので、例えば、1 ストリーム命令による処理サイクル数が多いほど、或いは、ストリーム処理を実行中の割り込み発生頻度が高いほど、より一層高速化の効果が大きくなるのはいうまでもない。

【 0 0 6 4 】

図 6 は、本実施例の演算処理装置における読出回路の動作の一例を説明するための図であり、図 7 は、本実施例の演算処理装置における読出回路の動作の他の例を説明するための図である。

【 0 0 6 5 】

図 6 および図 7 に示されるように、読出回路 2 1 0 は、POP 部 2 1 およびレジスタ 2 2 1 , 2 2 2 を含み、データメモリ 4 は、メモリ部 4 1 , 4 2 を含む。なお、メモリ部 4 1 および 4 2 は、例えば、データメモリ 4 における異なる番地(先頭アドレス)のバンク化されたメモリ領域を示すもので、2 つのメモリを持たなくてもよいのはもちろんである。

10

20

30

40

50

【 0 0 6 6 】

図 6 に示されるように、読出回路 2 1 0 において、P O P 部 2 1 は、データメモリ 4 のメモリ部(第 1 バンク) 4 1 から先頭アドレスおよびストリーム長を指定して第 1 データを読み出し、レジスタ 2 2 1 に格納する。

【 0 0 6 7 】

さらに、読出回路 2 1 0 において、P O P 部 2 1 は、データメモリ 4 のメモリ部(第 2 バンク) 4 2 から先頭アドレスおよびストリーム長を指定して第 2 データを読み出し、レジスタ 2 2 2 に格納する。この読出回路 2 1 0 の処理は、例えば、前述した図 3 の演算処理装置における処理 P 2 1 に対応する。

【 0 0 6 8 】

すなわち、P O P 部 2 1 は、データメモリ 4 からストリームデータを読み出し、ストリーム処理の読出ステージ(P O P 部 2 1)と実行ステージ(E X E C 部 2 3)の間のレジスタ(パイプラインレジスタ) 2 2 1, 2 2 2 に投入(格納)してパイプライン処理を実行する。

【 0 0 6 9 】

このように、例えば、第 1 バンク 4 1 および第 2 バンク 4 2 にバンク化されたデータメモリ 4 から、先頭アドレスおよびストリーム長を指定してストリームデータを読み出すことにより、メモリのポート数およびサイクルオーバーヘッドを解消することができる。

【 0 0 7 0 】

また、図 7 に示されるように、例えば、D M A (Direct Memory Access) 5 により、メモリ部(第 1 および第 2 バンク) 4 1, 4 2 から読み出されたデータを F I F O (First In First Out) バッファ 6 1, 6 2 を介して読出回路 2 1 0 に供給することもできる。すなわち、データメモリ 4 からのデータ転送を D M A 5 に任せ、F I F O バッファ 6 1, 6 2 から読み出しデータを取り出すこともできる。

【 0 0 7 1 】

図 8 は、本実施例の演算処理装置における実行回路の動作の一例を説明するための図である。図 8 に示されるように、実行回路 2 3 0 は、E X E C 部 2 3 およびレジスタ 2 4 を含む。

【 0 0 7 2 】

実行回路 2 3 0 において、E X E C 部 2 3 は、レジスタ 2 2 1 および 2 2 2 に書き込まれたデータに対してストリーム処理を実行し、その演算結果をレジスタ 2 4 に書き込む。この実行回路 2 3 0 の処理は、例えば、前述した図 3 の演算処理装置における処理 P 2 2 に対応する。

【 0 0 7 3 】

すなわち、E X E C 部 2 3 は、レジスタ 2 2 1 および 2 2 2 に投入されたデータに対してストリーム処理を実行し、その演算結果を E X E C 部 2 3 と P U S H 部 2 5 の間のレジスタ(パイプラインレジスタ) 2 4 に投入してパイプライン処理を実行する。

【 0 0 7 4 】

図 9 は、本実施例の演算処理装置における実行回路の動作の他の例を説明するための図であり、実行回路 2 3 0 を多段の E X E C 部 2 3 1 ~ 2 3 3 およびレジスタ 2 4 1 ~ 2 4 3 としたものである。

【 0 0 7 5 】

このとき、読出回路 2 1 0 のレジスタは、初段の 2 つの E X E C 部 2 3 1 および 2 3 2 に対応させて、4 つのレジスタ 2 2 1 a, 2 2 1 b および 2 2 2 a, 2 2 2 b とされている。

【 0 0 7 6 】

また、実行回路 2 3 0 のレジスタも、3 つの E X E C 部 2 3 1 ~ 2 3 3 による演算結果を格納するために 3 つのレジスタ 2 4 1 ~ 2 4 3 とされている。なお、図 9 に示す実行回路は単なる例であり、様々な構成を適用することができるのはもちろんである。

【 0 0 7 7 】

このように、実行回路 2 3 0 (演算器データパス)は多段構成でもよく、演算結果を毎サ

10

20

30

40

50

イクル E X E C 部 2 3 3 と P U S H 部 2 5 の間のレジスタ(パイプラインレジスタ) 2 4 3 に投入してパイプライン処理を実行することができる。

【 0 0 7 8 】

図 1 0 は、本実施例の演算処理装置における書込回路の動作の一例を説明するための図であり、図 1 1 は、本実施例の演算処理装置における書込回路の動作の他の例を説明するための図である。

【 0 0 7 9 】

図 1 0 に示されるように、書込回路 2 5 0 は、P U S H 部 2 5 を含み、レジスタ 2 4 に格納された演算結果をデータメモリ 4 のメモリ部 4 3 に書き込む。すなわち、E X E C 部 2 3 と P U S H 部 2 5 間のパイプラインレジスタ 2 4 から出力データを取り出し、例えば、先頭アドレスとストリーム長で示されたメモリ領域へ書き込む。

10

【 0 0 8 0 】

この書込回路 2 5 0 の処理は、例えば、前述した図 3 の演算処理装置における処理 P 2 3 に対応する。ここで、メモリ部 4 3 は、例えば、データメモリ 4 において、メモリ部 4 1 , 4 2 とは異なるメモリ領域とすることができる。

【 0 0 8 1 】

図 1 0 に示す書込回路 2 5 0 は、レジスタ 2 4 に格納された演算結果を直接データメモリ部 4 3 に書き込む。これに対して、図 1 1 に示す書込回路 2 5 0 は、レジスタ 2 4 に格納された演算結果を F I F O バッファ 7 に書き込み、その F I F O バッファ 7 に書き込まれたデータを D M A 8 がメモリ部 4 3 に転送する。

20

【 0 0 8 2 】

すなわち、図 1 1 に示す書込回路 2 5 0 は、レジスタ 2 4 に格納された演算結果を順に F I F O バッファ 7 に書き込み、F I F O バッファ 7 からメモリ部 4 3 (データメモリ 4) へのデータ転送は、D M A 8 に任せるようになっている。

【 0 0 8 3 】

図 1 2 は、本実施例の演算処理装置におけるパラメータ情報の一例を説明するための図である。ストリーム処理に使用するパラメータ情報は、例えば、各ストリーム (i) の先頭アドレス (a i)、ストリーム長 (l i)、演算オペコード (o) および演算モード (m) は単一かつ長ビット長の命令(セット命令: s e t)で表現することができる。

【 0 0 8 4 】

30

このセット命令(パラメータ情報)は、参照符号 P 1 0 で示されるように、命令メモリ 1 8 から読み出されて、パラメータレジスタ 1 4 0 へ一括して代入(セット)される。そして、各パイプラインステージ (P O P 部 2 1 , E X E C 部 2 3 および P U S H 部 2 5) は、参照符号 P 1 1 で示されるように、パラメータレジスタ 1 4 0 からパラメータ情報を参照してパイプライン実行する。

【 0 0 8 5 】

図 1 3 および図 1 4 は、本実施例の演算処理装置におけるステップ命令を説明するための図である。図 1 3 および図 1 4 に示されるように、本実施例の演算処理装置(ストリームエンジン 2) は、セット命令により制御することができる。

【 0 0 8 6 】

40

すなわち、参照符号 P 2 0 で示されるように、命令メモリ 1 8 からステップ命令を読み出し、そのステップ命令の実行により、ストリームエンジン 2 の各パイプラインステージの処理 P 2 1 ~ P 2 3 を制御することができる。なお、ステップ命令は、例えば、予めプログラマーにより作成されたものが使用される。

【 0 0 8 7 】

ここで、ステップ命令 step 1 ~ step N は、命令メモリ 1 8 から順番に読み出されて命令発行部 1 4 からストリームエンジン 2 へ発行され、各パイプライン処理 P 2 1 ~ P 2 3 が実行される。

【 0 0 8 8 】

図 1 3 に示されるように、ステップ命令は、命令発行部 1 4 からストリームエンジン 2

50

へ発行され、1つのステップ命令により、POP部21, EXEC部23およびPUSH部25が1つの処理(P21, P22, P23)を実行する。

【0089】

すなわち、図14(a)に示されるように、処理P21は、POP部21がデータメモリ4からデータを読み出してレジスタ221, 222に書き込む処理である。また、図14(b)に示されるように、処理P22は、EXEC部23がレジスタ221, 222に書き込まれたデータに対してストリーム処理を実行してレジスタ24に書き込む処理である。

【0090】

さらに、図14(c)に示されるように、処理P23は、PUSH部25がレジスタ24に書き込まれたデータをデータメモリ19に書き込む処理である。これらの処理P21~P23は、命令発行部14から発行されたステップ命令に従ってパイプライン実行される。

10

【0091】

図15は、本実施例の演算処理装置におけるステップ命令の変形を説明するための図である。前述した図13では、N個のステップ命令 step 1~step N は、そのまま命令メモリ18から読み出されて命令発行部14からストリームエンジン2へ発行されている。

【0092】

これに対して、図15に示す変形例では、セット命令を、連続する繰り返し処理(ループ処理)を効率よく実行するためのループ処理専用の命令(ゼロオーバーヘッドループ命令)と組み合わせるようになっている。

20

【0093】

すなわち、N個のステップ命令 step 1~step N は、ゼロオーバーヘッドループ命令(loop N step)とすることで、命令列を増加させないようにすることができる。なお、ゼロオーバーヘッドループ命令においても、例えば、割り込み発生時、ストリームは、直ちに実行中のステップで処理を停止するようになっている。

【0094】

図16および図17は、本実施例の演算処理装置におけるマイクロ命令を説明するための図である。図16に示されるように、命令発行部14からストリームエンジン2へ発行される命令は、マイクロ命令とされている。

【0095】

30

すなわち、図16の参照符号P30で示されるように、命令メモリ18からマイクロ命令を読み出し、そのマイクロ命令の実行により、ストリームエンジン2の各パイプラインステージの処理P21~P23を制御するようになっている。

【0096】

例えば、図17(a)に示す処理P21に対してpop命令を割り当て、図17(b)に示す処理P22に対してexec命令を割り当て、そして、図17(c)に示す処理P23に対してpush命令を割り当て、各マイクロ命令により実行する。これにより、各パイプラインステージの処理P21~P23をマイクロ命令により個別に制御することができる。

【0097】

図18は、本実施例の演算処理装置におけるマイクロ命令によるアクセス制御を説明するための図である。

40

【0098】

ここで、図18(a)は、pop, execおよびpush命令を全て発行した場合を示し、図18(b)は、pop命令を停止した場合を示し、そして、図18(c)は、push命令を停止した場合を示す。なお、演算処理装置には、前述した図7および図11のように、DMA5, 8およびFIFOバッファ61, 62, 7が設けられている。

【0099】

まず、図18(a)に示されるように、pop命令, exec命令およびpush命令の全てが発行されると、各パイプラインステージの処理P21~P23が毎サイクル実行される。

50

【0100】

次に、図18(b)に示されるように、pop命令を停止すると、すなわち、exec命令およびpush命令のみ実行すると、POP部21は、FIFOバッファ61, 62からのデータ読み出しを停止する。

【0101】

これにより、FIFOバッファ61, 62は、DMA(入力DMA)5によるデータ転送で満状態となり、DMA5がFIFOバッファ61, 62の満状態を検出して自動停止する。すなわち、マイクロ命令であるpop命令を停止することにより、ストリームエンジン2のパイプライン処理を停止することができる。

【0102】

10

さらに、図18(c)に示されるように、push命令を停止すると、すなわち、pop命令およびexec命令のみ実行すると、PUSH部25は、レジスタ24からデータを読み出してFIFOバッファ7に格納する動作を停止する。

【0103】

これにより、FIFOバッファ7は空状態となり、DMA(出力DMA)8がFIFOバッファ7の空状態を検出して自動停止する。すなわち、マイクロ命令であるpush命令を停止することにより、ストリームエンジン2のパイプライン処理を停止することができる。

【0104】

このように、pop命令, exec命令およびpush命令のマイクロ命令を使用することで、例えば、割り込み発生時でも、DMA5, 8がメモリアクセスを自律的に制御することができる。すなわち、メモリ-演算器間のデータ転送の制御を簡略化することができ、メモリアクセス制御のハードウェア量を削減することが可能になる。

20

【0105】

図19は、本実施例の演算処理装置におけるマイクロ命令をVLIW命令に埋め込む(パックする)様子を示す図である。図16~図18を参照して説明したように、マイクロ命令を使用する場合、例えば、VLIW(Very Long Instruction Word: 超長命令語)命令に埋め込むことで各処理を同時に実行することができ、実行サイクル数を削減することが可能となる。

【0106】

30

すなわち、複数のマイクロ命令をVLIW命令に埋め込むことで、ループ処理の命令数を削減することができ、さらに、ループの実行サイクル数を削減することもできる。また、ベースプロセッサ(VLIWプロセッサを想定: 演算処理装置1)の命令セットアーキテクチャを有効に活用することも可能になる。

【0107】

図19は、M個のマイクロ命令をN個のVLIW命令にパックする様子を示しているが、ここで、VLIW 1命令~VLIW 3命令によるプロローグ処理、および、VLIW N-2命令~VLIW N命令によるエピローグ処理を、図20および図21を参照して説明する。

【0108】

図20は、図19に示すVLIW命令のプロローグ処理を説明するための図であり、図20(a)はVLIW 1命令の処理を示し、図20(b)はVLIW 2命令の処理を示し、そして、図20(c)はVLIW 3命令の処理を示す。

40

【0109】

ここで、図19に示されるように、プロローグ処理は、停止しているストリームエンジン2起動させる処理で、VLIW 1[pop], VLWI 2[pop, exec]およびVLWI 3[pop, exec, push]の3つの命令を実行することで達成される。

【0110】

まず、図20(a)に示されるように、VLIW 1命令によるpop命令のみ実行する。すなわち、pop命令により、POP部21がデータメモリ4からデータを読み出してレジスタ221, 222に書き込む処理P21を実行する。これにより、レジスタ221, 22

50

2 には、E X E C 部 2 3 が演算処理を行うデータが投入されたことになる。

【 0 1 1 1 】

次に、図 2 0 (b) に示されるように、VLIW 2 命令による p o p 命令および e x e c 命令を実行する。すなわち、p o p 命令により上述した処理 P 2 1 を実行すると共に、e x e c 命令により、E X E C 部 2 3 がレジスタ 2 2 1 , 2 2 2 に書き込まれたデータに対してストリーム処理を実行してレジスタ 2 4 に書き込む処理 P 2 2 を実行する。

【 0 1 1 2 】

これにより、レジスタ 2 2 1 , 2 2 2 には、E X E C 部 2 3 が演算処理を行うデータが投入され、また、レジスタ 2 4 には、P U S H 部 2 5 がデータメモリ 4 に書き込む演算結果のデータが投入されたことになる。

10

【 0 1 1 3 】

そして、図 2 0 (c) に示されるように、VLIW 3 命令による p o p 命令、e x e c 命令および P U S H 命令を実行する。すなわち、p o p 命令により処理 P 2 1 を実行すると共に、e x e c 命令により処理 P 2 2 を実行し、さらに、P U S H 命令により、P U S H 部 2 5 がレジスタ 2 4 に書き込まれた演算結果データをデータメモリ 4 に書き込む処理 P 2 3 を実行する。

【 0 1 1 4 】

なお、このエピローグ処理以降、図 2 1 を参照して説明するエピローグ処理まで、VLIW 3 命令と同じ命令 (VLIW 4 命令、VLIW 5 命令、...) により処理 P 2 1 ~ P 2 3 によるパイプライン処理が継続して実行される。

20

【 0 1 1 5 】

図 2 1 は、図 1 9 に示す V L I W 命令のエピローグ処理を説明するための図であり、図 2 1 (a) は VLIW N-2 命令の処理を示し、図 2 1 (b) は VLIW N-1 命令の処理を示し、そして、図 2 1 (c) は VLIW N 命令の処理を示す。

【 0 1 1 6 】

ここで、図 1 9 に示されるように、エピローグ処理は、図 2 0 を参照して説明したプロローグ処理と逆に動作中のストリームエンジン 2 を停止させる処理である。このエピローグ処理は、VLIW N-2 [pop, exec, push]、VLIW N-1 [exec, push] および VLIW N [push]、の 3 つの命令を実行することで達成される。

【 0 1 1 7 】

まず、図 2 1 (a) に示されるように、VLIW N-2 命令による p o p 命令、e x e c 命令および p u s h 命令を行う。この VLIW N-2 命令は、図 2 0 (c) を参照して説明した VLIW 3 命令、すなわち、処理 P 2 1 ~ P 2 3 により継続して実行されるパイプライン処理と同じものである。

30

【 0 1 1 8 】

次に、図 2 1 (b) に示されるように、VLIW N-1 命令による e x e c 命令および p u s h 命令を実行する。すなわち、p o p 命令を除くことにより、P O P 部 2 1 がデータメモリ 4 からデータを読み出してレジスタ 2 2 1 , 2 2 2 に書き込む処理 P 2 1 を停止する。これにより、レジスタ 2 2 1 , 2 2 2 は空状態となる。

【 0 1 1 9 】

そして、図 2 1 (c) に示されるように、VLIW N 命令による P U S H 命令のみ実行する。すなわち、p o p 命令および e x e c 命令を除くことで、レジスタ 2 2 1 , 2 2 2 だけでなく、レジスタ 2 4 も、空状態となる。

40

【 0 1 2 0 】

なお、p o p 命令、e x e c 命令および p u s h 命令の 3 つのマイクロ命令によりストリームエンジン 2 を制御するのは、単なる例であり、さらなるマイクロ命令を追加し、或いは、異なるマイクロ命令を適用するといった様々な変更が可能なものではない。

【 0 1 2 1 】

なお、上述した実施例では、L T E アドバンス等における行列演算処理を行う演算処理装置を例として説明したが、本実施例は、このような無線通信デバイスに適用する演算

50

処理装置に限定されず、様々な演算処理装置に幅広く適用することが可能である。

【0122】

以上、実施形態を説明したが、ここに記載したすべての例や条件は、発明および技術に適用する発明の概念の理解を助ける目的で記載されたものであり、特に記載された例や条件は発明の範囲を制限することを意図するものではない。また、明細書のそのような記載は、発明の利点および欠点を示すものでもない。発明の実施形態を詳細に記載したが、各種の変更、置き換え、変形が発明の精神および範囲を逸脱することなく行えることが理解されるべきである。

【0123】

以上の実施例を含む実施形態に関し、さらに、以下の付記を開示する。

10

(付記1)

演算を実行する演算器、および、ストリーム処理を実行するストリームエンジンを含み、前記演算器のデータパスと前記ストリームエンジンのデータパスを密結合した、ことを特徴とする演算処理装置。

【0124】

(付記2)

さらに、

命令を発行する命令発行部を有し、

前記命令発行部は、前記演算器に対する命令を発行すると共に、前記ストリームエンジンに対する命令も発行する、

20

ことを特徴とする付記1に記載の演算処理装置。

【0125】

(付記3)

前記ストリームエンジンは、

メモリからデータを読み出す読出回路と、

前記読み出したデータに対してストリーム処理を実行する実行回路と、

前記ストリーム処理された演算結果を前記メモリに書き込む書込回路と、を含む、

ことを特徴とする付記2に記載の演算処理装置。

【0126】

(付記4)

30

前記読出回路は、POP部および第1レジスタを含み、

前記POP部は、前記メモリにおける、先頭アドレスおよびストリーム長で示された第1メモリ部からデータを読み出して前記第1レジスタに格納する、

ことを特徴とする付記3に記載の演算処理装置。

【0127】

(付記5)

前記実行回路は、EXEC部および第2レジスタを含み、

前記EXEC部は、前記第1レジスタに格納されたデータに対してストリーム処理を実行し、前記ストリーム処理された演算結果を前記第2レジスタに格納する、

ことを特徴とする付記4に記載の演算処理装置。

40

【0128】

(付記6)

前記実行回路は、階層化された複数のEXEC部および各階層の前記EXEC部間に設けられた複数の第3レジスタを含む、

ことを特徴とする付記5に記載の演算処理装置。

【0129】

(付記7)

前記書込回路は、PUSH部を含み、

前記PUSH部は、前記第2レジスタに格納された演算結果を、前記メモリにおける、先頭アドレスおよびストリーム長で示された第2メモリ部に書き込む、

50

ことを特徴とする付記 5 または付記 6 に記載の演算処理装置。

【 0 1 3 0 】

(付記 8)

前記命令発行部が前記ストリームエンジンに発行する命令は、ステップ命令であり、
前記ストリームエンジンの各パイプラインステージは、1つの前記ステップ命令に従ってそれぞれ1つの処理を実行する、

ことを特徴とする付記 2 乃至付記 7 のいずれか 1 項に記載の演算処理装置。

【 0 1 3 1 】

(付記 9)

前記ストリーム処理に使用するパラメータ情報は、単一かつ長ビット長のセット命令で表現される、

ことを特徴とする付記 8 に記載の演算処理装置。

【 0 1 3 2 】

(付記 1 0)

前記ストリーム処理に使用するパラメータ情報は、各ストリームの先頭アドレス、ストリーム長および演算モードを含む、

ことを特徴とする付記 9 に記載の演算処理装置。

【 0 1 3 3 】

(付記 1 1)

さらに、

前記ストリーム処理に使用するパラメータ情報を一括してセットするパラメータレジスタを含み、

前記ストリームエンジンの各パイプラインステージは、前記パラメータレジスタからパラメータ情報を参照してパイプライン実行する、

ことを特徴とする付記 8 に記載の演算処理装置。

【 0 1 3 4 】

(付記 1 2)

前記命令発行部が前記ストリームエンジンに発行する命令は、ステップ命令を分解して、前記ストリームエンジンの各パイプラインステージの操作を制御する短ビット長のマイクロ命令であり、

前記各パイプラインステージは、それぞれ対応する前記マイクロ命令に従って独立して処理を実行する、

ことを特徴とする付記 2 乃至付記 7 のいずれか 1 項に記載の演算処理装置。

【 0 1 3 5 】

(付記 1 3)

さらに、

前記メモリと前記読出回路の間に設けられた第 1 F I F O バッファを有し、

前記メモリは D M A 制御され、前記メモリからデータを読み出す読出回路の処理を制御する第 1 マイクロ命令を停止することで、前記第 1 F I F O バッファを満状態として、前記ストリームエンジンのパイプライン処理を停止する、

ことを特徴とする付記 1 2 に記載の演算処理装置。

【 0 1 3 6 】

(付記 1 4)

さらに、

前記書込回路と前記メモリの間に設けられた第 2 F I F O バッファを有し、

前記メモリは D M A 制御され、前記メモリへデータを書き込む書込回路の処理を制御する第 2 マイクロ命令を停止することで、前記第 2 F I F O バッファを空状態として、前記ストリームエンジンのパイプライン処理を停止する、

ことを特徴とする付記 1 2 に記載の演算処理装置。

【 0 1 3 7 】

10

20

30

40

50

(付記 1 5)

前記演算器が V L I W 命令により制御されるとき、

前記ストリームエンジンの各パイプラインステージの操作を制御するマイクロ命令を V L I W 命令に埋め込む、

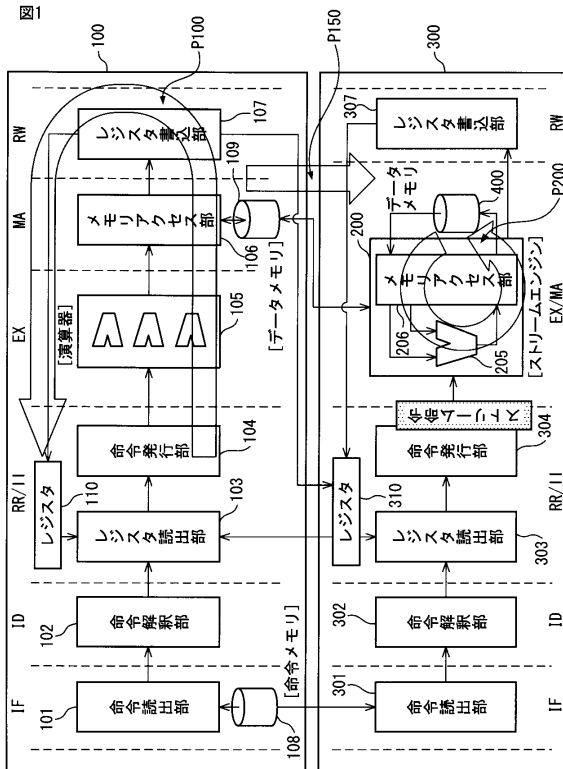
ことを特徴とする付記 1 2 乃至付記 1 4 のいずれか 1 項に記載の演算処理装置。

【符号の説明】

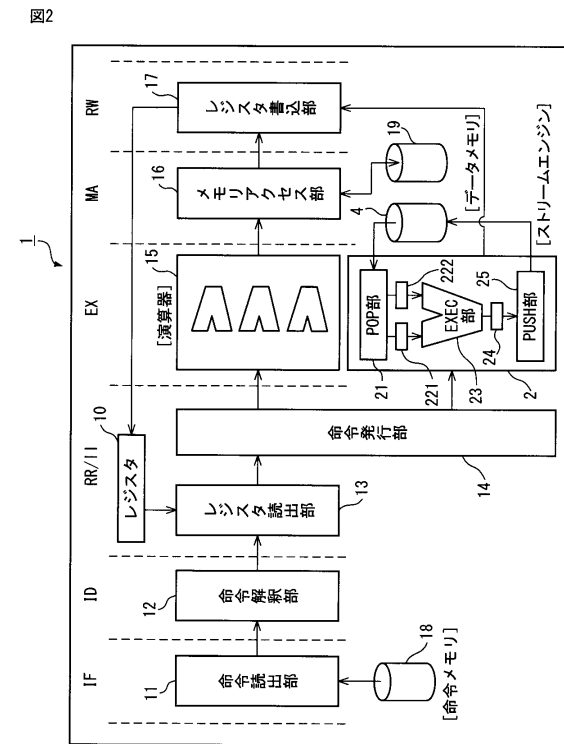
【 0 1 3 8 】

1	プロセッサ	
2, 2 0 0	ストリームエンジン	
4, 4 0 0	データメモリ	10
5, 8	D M A	
7, 6 1, 6 2	F I F O バッファ	
1 0, 1 1 0, 3 1 0	レジスタ	
1 1, 1 0 1, 3 0 1	命令読出部	
1 2, 1 0 2, 3 0 2	命令解釈部	
1 3, 1 0 3, 3 0 3	レジスタ読出部	
1 4, 1 0 4, 3 0 4	命令発行部	
1 5, 1 0 5	演算器	
1 6, 1 0 6	メモリアクセス部	
1 7, 1 0 7	レジスタ書込部	20
1 8, 1 0 8	命令メモリ	
1 9, 1 0 9	データメモリ	
2 1	P O P 部	
2 3	E X E C 部	
2 4, 2 2 1, 2 2 2, 2 2 1 a, 2 2 1 b, 2 2 2 a, 2 2 2 b, 2 4 1 ~ 2 4 3	レジスタ	
2 5	P U S H 部	
4 1 ~ 4 3	メモリ部	
1 0 0	ベースプロセッサ	
1 4 0	パラメータレジスタ	30
2 1 0	読出回路	
2 3 0	実行回路	
2 5 0	書込回路	
3 0 0	コプロセッサ	

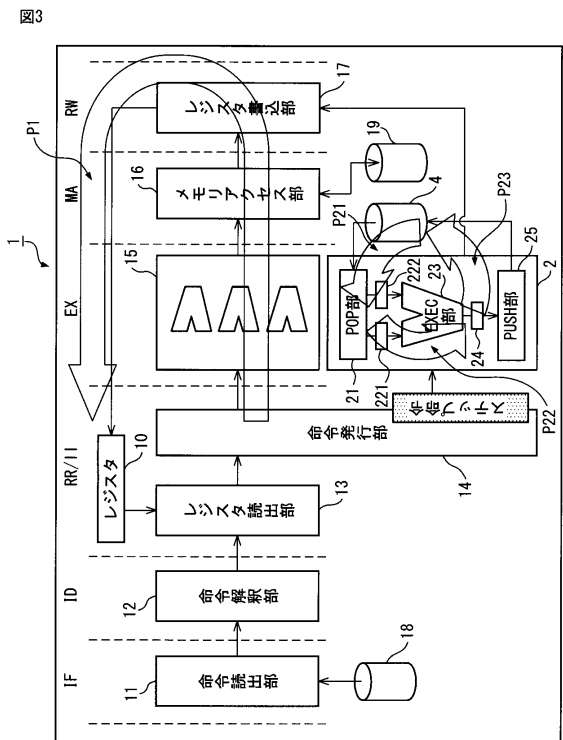
【 図 1 】



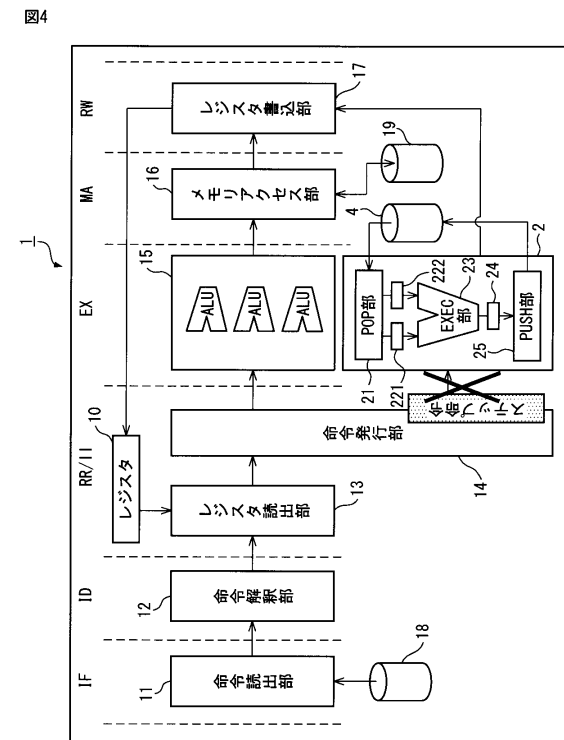
【 図 2 】



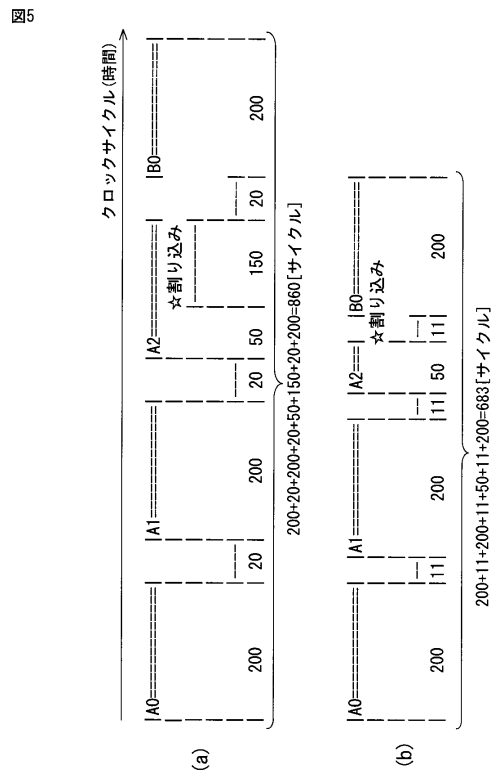
【 図 3 】



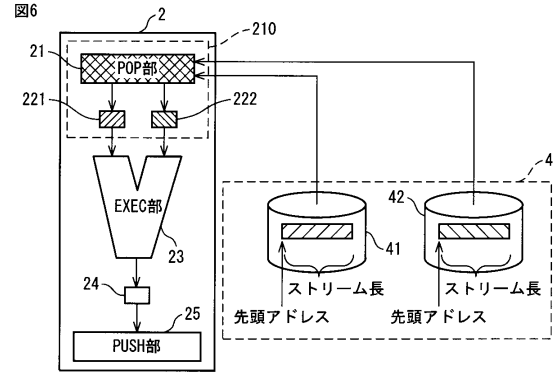
【 図 4 】



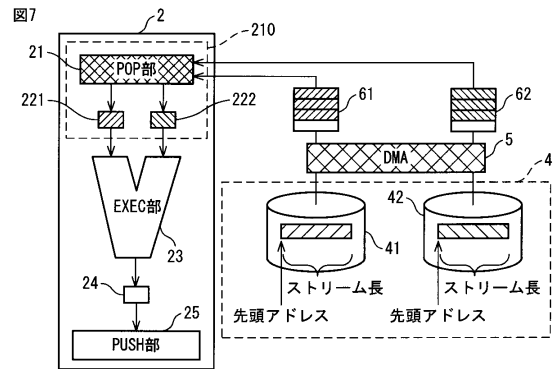
【 図 5 】



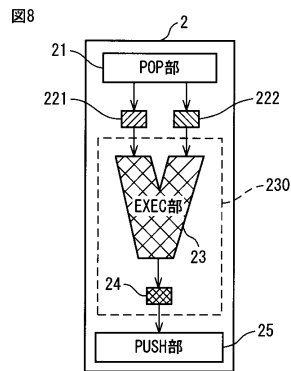
【 図 6 】



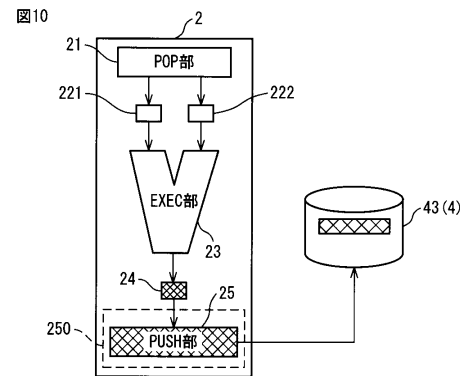
【圖 7】



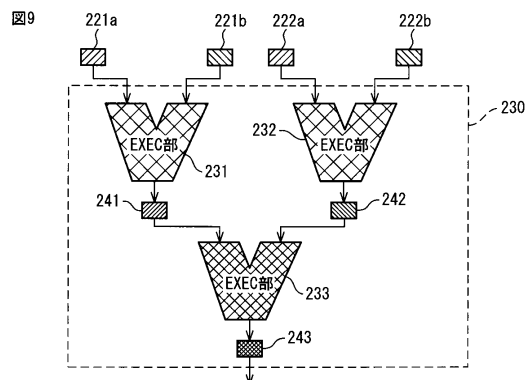
【圖 8】



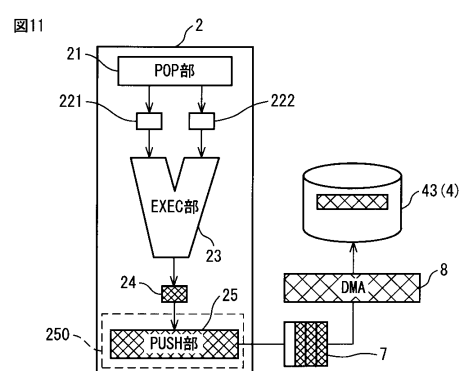
【 図 1 0 】



【圖 9】

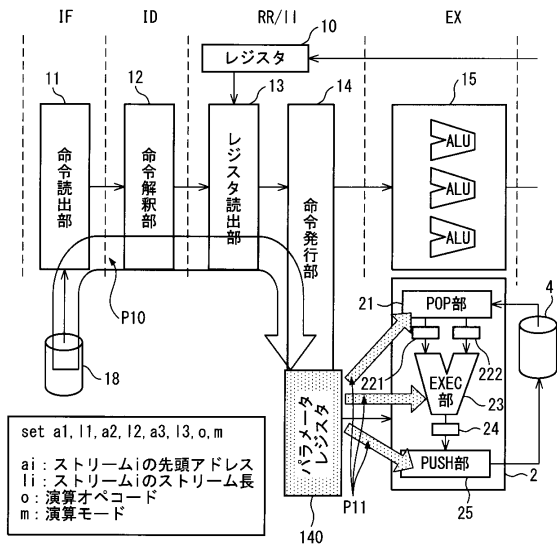


【 図 1 1 】



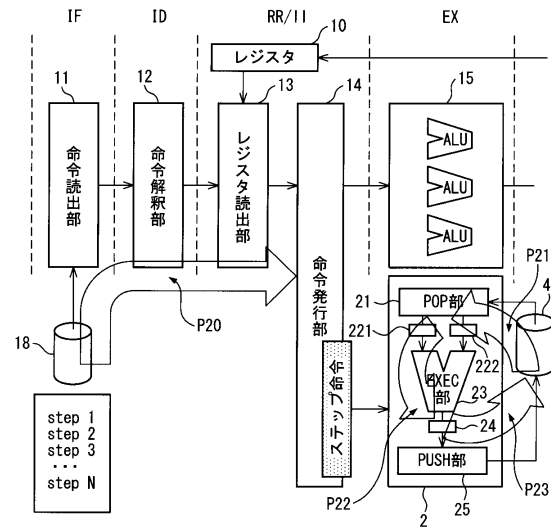
【図 12】

図12



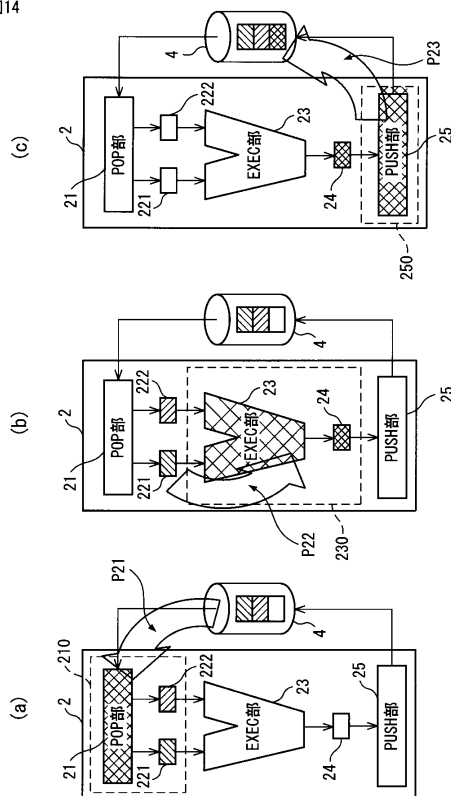
【図 13】

図13



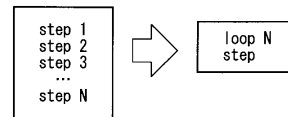
【図 14】

図14



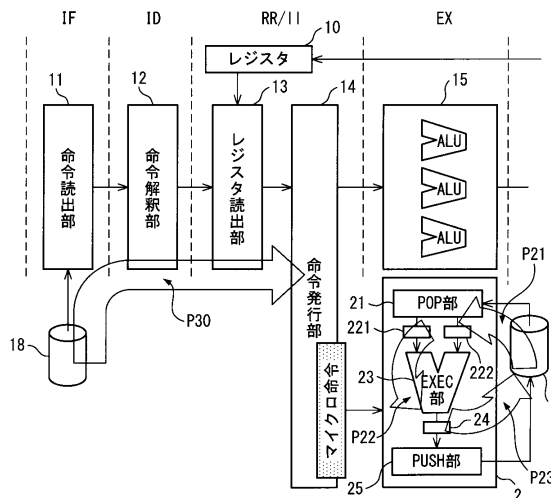
【図 15】

図15



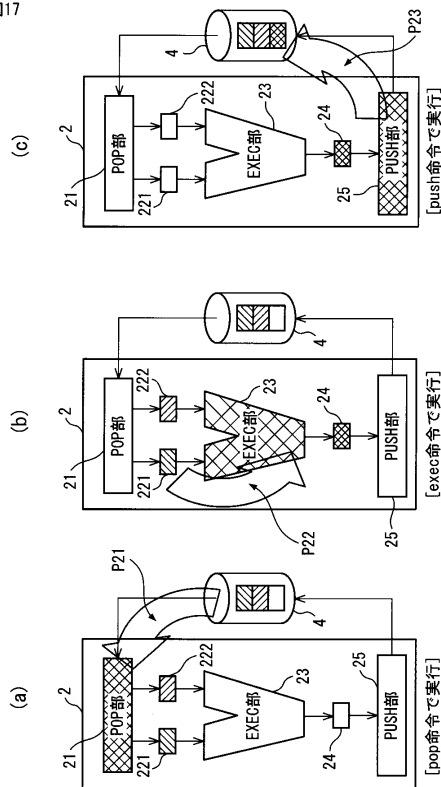
【図 16】

図16



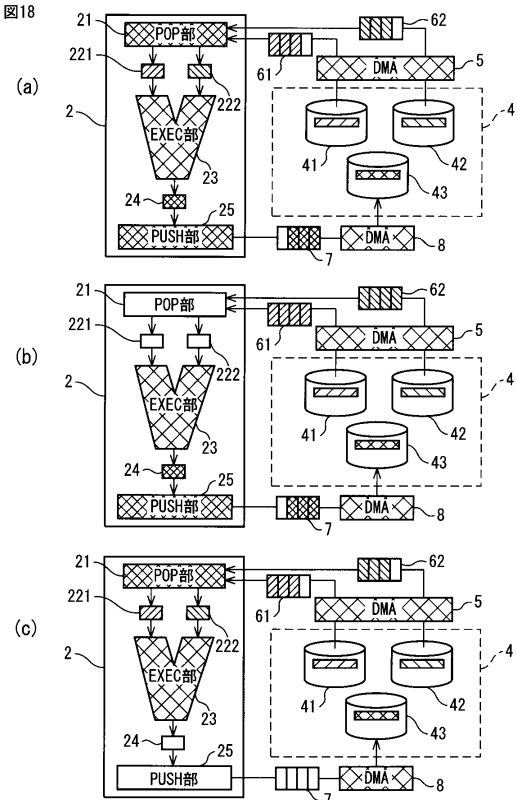
【図17】

図17



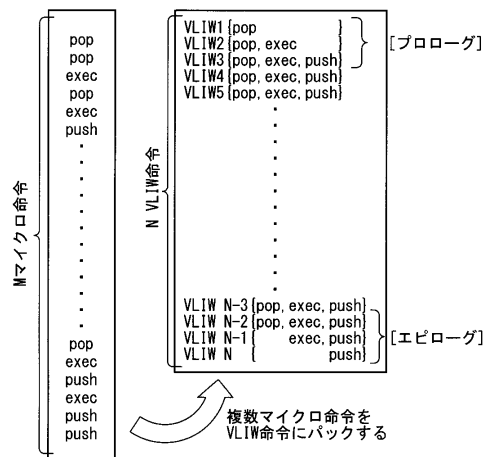
【図18】

図18



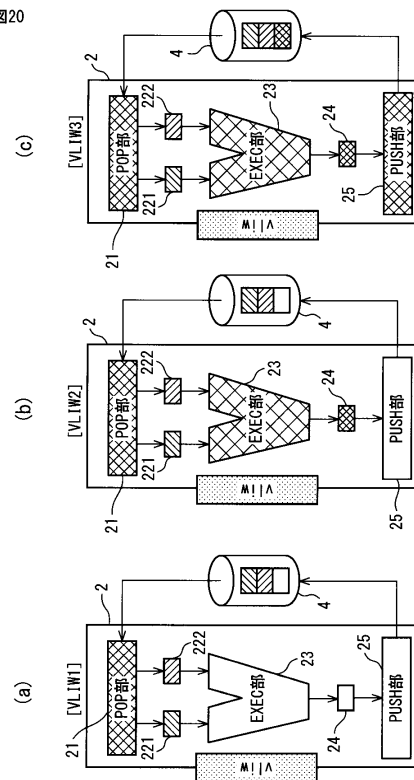
【図19】

図19



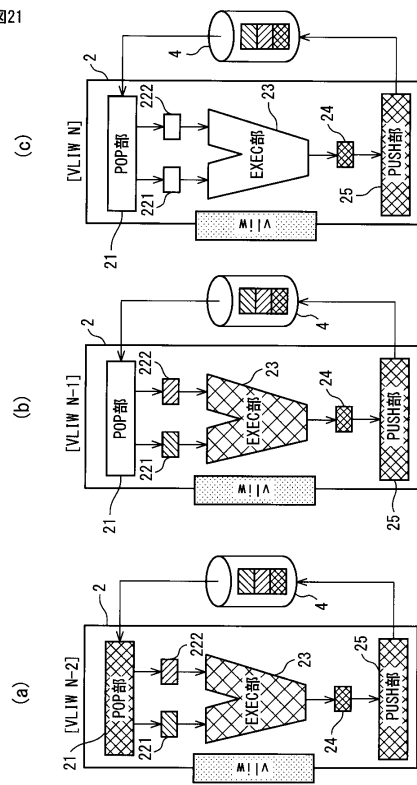
【図20】

図20



【図 21】

図21



フロントページの続き

(72)発明者 葛 毅

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 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

審査官 清木 泰

(56)参考文献 米国特許出願公開第2010/0325188 (US, A1)

特開2011-034189 (JP, A)

特表平11-510291 (JP, A)

特開2000-207202 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/30 - 9/42

G06T 1/00 - 1/40

G06T 3/00 - 5/50

G06T 9/00 - 9/40

H04N 7/12

H04N19/00 - 19/98

G06F17/10 - 17/18

H04B 1/69 - 1/719

H04J 1/00 - 1/20

H04J 4/00 - 13/22

H04J99/00

H04L 5/00 - 5/12