

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6340887号
(P6340887)

(45) 発行日 平成30年6月13日(2018.6.13)

(24) 登録日 平成30年5月25日(2018.5.25)

(51) Int.Cl.

G06F 9/38 (2006.01)
G06F 9/35 (2006.01)

F 1

G06F 9/38
G06F 9/34370A
310

請求項の数 10 (全 35 頁)

(21) 出願番号 特願2014-82660 (P2014-82660)
 (22) 出願日 平成26年4月14日 (2014.4.14)
 (65) 公開番号 特開2015-203950 (P2015-203950A)
 (43) 公開日 平成27年11月16日 (2015.11.16)
 審査請求日 平成29年1月10日 (2017.1.10)

(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 100094525
 弁理士 土井 健二
 (74) 代理人 100094514
 弁理士 林 恒徳
 (72) 発明者 田端 猛一
 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 吉田 利雄
 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】演算処理装置及び演算処理装置の制御方法

(57) 【特許請求の範囲】

【請求項 1】

命令をデコードする命令デコーダと、
 前記命令デコーダによりメモリアクセス命令のエントリを生成されるメモリアクセスエントリ部と、
 前記メモリアクセスエントリ部から出力された前記メモリアクセス命令のエントリをメモリに対して実行するメモリアクセスパイプラインと、
 前記命令デコーダにより複数のデータを1つの命令で処理するマルチデータ命令のエントリを生成されるマルチデータ命令エントリ部と、

複数の演算器と複数のマルチデータ命令用レジスタとを有し、前記マルチデータ命令エントリ部から出力された前記マルチデータ命令のエントリの処理を前記複数の演算器により並列に処理し、前記複数のマルチデータ命令用レジスタに演算結果を格納する演算パイプラインとを有し、

前記演算パイプラインは、前記複数のマルチデータ命令用レジスタに格納されている複数のメモリアドレスについて前記メモリにメモリアクセスするインダイレクトメモリアクセスを実行する前記マルチデータ命令のエントリの前記マルチデータ命令エントリ部からの出力に応答して、前記メモリアクセスパイプラインに前記インダイレクトメモリアクセスを実行する前記マルチデータ命令に対応する複数のメモリアクセス命令のメモリアクセス要求を生成し、前記複数の演算器が前記複数のマルチデータ命令用レジスタから取得した前記複数のメモリアドレスを前記メモリアクセスパイプラインに供給する演算処理装置

10

20

。

【請求項 2】

前記演算パイプラインは、前記メモリアクセスパイプラインの第1サイクルのステージに前記複数のメモリアクセス命令のメモリアクセス要求を生成し、前記メモリアクセスパイプラインの前記第1サイクルより後の第2サイクルのステージに前記複数のメモリアドレスを供給する

請求項1に記載された演算処理装置。

【請求項 3】

前記演算パイプラインは、前記メモリアクセスパイプラインに生成した複数のメモリアクセス命令のメモリアクセス要求のパイプライン転送タイミングにあわせて前記複数のメモリアドレスを供給する

10

請求項2に記載された演算処理装置。

【請求項 4】

さらに、前記メモリアクセスパイプラインに接続されたキャッシュユニットを有し、

前記演算パイプラインは、前記メモリアクセスパイプラインに生成する前記複数のメモリアクセス命令のメモリアクセス要求に前記キャッシュユニット内のアクセス先メモリアドレスを格納する複数のフェッチポートの識別情報を含める

請求項1に記載された演算処理装置。

【請求項 5】

前記演算パイプラインは、前記メモリアクセスパイプラインに、前記複数のメモリアクセス命令のメモリアクセス要求をシリアルに生成し、前記複数のメモリアドレスをシリアルに供給する

20

請求項1、2、3のいずれかに記載された演算処理装置。

【請求項 6】

前記メモリアクセスパイプラインが複数設けられ、

前記演算パイプラインは、前記複数のメモリアクセスパイプラインに、前記複数のメモリアクセス命令のメモリアクセス要求の少なくとも一部のメモリアクセス要求を並列に生成し、前記複数のメモリアドレスの少なくとも一部のアドレスを並列に供給する

請求項1、2、3のいずれかに記載された演算処理装置。

【請求項 7】

さらに、前記メモリアクセスパイプラインに接続されたキャッシュユニットを有し、

30

前記キャッシュユニットは、前記複数のメモリアクセス命令のメモリアクセス要求に応答して、前記複数のマルチデータ命令用レジスタとの間でデータ転送を行う

請求項1に記載された演算処理装置。

【請求項 8】

前記演算パイプラインは、前記メモリアクセスエントリ部に抑止信号を出力して、前記メモリアクセスエントリ部に、前記メモリアクセスパイプラインに生成する前記複数のメモリアクセス命令のメモリアクセス要求と衝突するメモリアクセス命令のエントリの出力を抑止させる

請求項1に記載された演算処理装置。

40

【請求項 9】

前記演算パイプラインは、前記マルチデータ命令エントリ部に抑止信号を出力して、前記マルチデータ命令エントリ部に、前記メモリアクセスパイプラインにシリアルに生成する前記複数のメモリアクセス命令のメモリアクセス要求と衝突する前記インダイレクトメモリアクセスを実行する前記マルチデータ命令のエントリの出力を抑止させる

請求項5に記載された演算処理装置。

【請求項 10】

命令をデコードする命令デコーダと、

前記命令デコーダによりメモリアクセス命令のエントリを生成されるメモリアクセスエントリ部と、

50

前記メモリアクセスエントリ部から出力された前記メモリアクセス命令のエントリをメモリに対して実行するメモリアクセスパイプラインと，

前記命令デコーダにより複数のデータを1つの命令で処理するマルチデータ命令のエントリを生成されるマルチデータ命令エントリ部と，

複数の演算器と複数のマルチデータ命令用レジスタとを有し，前記マルチデータ命令エントリ部から出力された前記マルチデータ命令のエントリの処理を前記複数の演算器により並列に処理し，前記複数のマルチデータ命令用レジスタに演算結果を格納する演算パイプラインとを有する演算処理装置の制御方法において，

前記演算パイプラインが，前記複数のマルチデータ命令用レジスタに格納されている複数のメモリアドレスについて前記メモリにメモリアクセスするインダイレクトメモリアクセスを実行する前記マルチデータ命令のエントリの前記マルチデータ命令エントリ部からの出力に応答して，前記メモリアクセスパイプラインに前記インダイレクトメモリアクセスを実行する前記マルチデータ命令に対応する複数のメモリアクセス命令のメモリアクセス要求を生成し，

前記演算パイプラインが，前記複数の演算器が前記複数のマルチデータ命令用レジスタから取得した前記複数のメモリアドレスを前記メモリアクセスパイプラインに供給する演算処理装置の制御方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は，演算処理装置及び演算処理装置の制御方法に関する。

【背景技術】

【0002】

演算処理装置であるC P Uまたは演算処理部であるC P Uコアの高速化手法として，スーパースカラやアウト・オブ・オーダ，S I M D (Single Instruction Multiple Data)などの技術が知られている。例えば，スーパースカラでは複数の命令を同時に処理し，アウト・オブ・オーダではC P Uコア内部の資源について処理が可能になったものを順不同で処理しプログラムの順序通り完了させる。

【0003】

一方，S I M Dは，1つの命令で複数のデータを並列に処理する。S I M D処理ではS I M Dレジスタと呼ばれるレジスタを用いる。S I M Dレジスタには，S I M D命令で並列に処理可能な個数と同数もしくはそれ以上の複数個のデータが1つの固まりとして格納される。S I M D命令で指定されるオペランドによりこの複数個のデータを1つの固まりとして格納するS I M Dレジスタが選択され，S I M Dレジスタ内の複数個のデータごとに命令の処理が実行される。この複数個のデータの各々を要素と呼び並列に処理するデータの要素数をS I M D幅と呼ぶ。S I M D命令においては，例えば命令で指定される第一ソースオペランド，第二ソースオペランド，第三ソースオペランドをS I M Dレジスタから読み出し，S I M D演算を行いデスティネーションオペランドにデータを書き込む。このS I M D処理では，複数個のデータについて同じ命令の処理が並列に実行される。

【0004】

また，S I M D命令以外の命令の処理においては，汎用レジスタと呼ばれるレジスタを用いる。汎用レジスタには，例えばメモリアクセスやS I M Dで並列に処理を行わないデータが格納される。

【0005】

従来は，メモリとS I M Dレジスタ間のデータ転送は，汎用レジスタに格納されたアドレスを用いて行われる。メモリアクセス命令においては，オペランドアドレス生成器が，命令で指定される第一ソースオペランド，第二ソースオペランドを汎用レジスタから読み出し，メモリアクセスのためのアドレスの生成を行う。この読み出したアドレスを先頭アドレスとして用いて，メモリの連続するアドレス領域に存在するS I M D幅のデータを読み出し，デスティネーションオペランドに対するS I M Dレジスタに読み出したデータを

10

20

30

40

50

ロードする，もしくはメモリの連続するアドレス領域に SIMD レジスタから読み出した SIMD 幅のデータをストアする。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2009-163442号公報

【特許文献2】特公平4-79026号公報

【特許文献3】特開2004-38750号公報

【特許文献4】特開2011-34450号公報

【発明の概要】

10

【発明が解決しようとする課題】

【0007】

しかしながら，並列に処理が可能なデータがメモリの連続するアドレス領域に存在しない場合は，上記の SIMD ロード命令， SIMD ストア命令を適用することはできない。

【0008】

さらに，従来の CPU コアには，独立した複数のアドレスが格納された SIMD レジスタをソースオペランドに指定してメモリアクセスするインダイレクトメモリアクセス命令を SIMD 命令で実行するための構成が設けられていない。したがって，従来の CPU コアは，例えば， SIMD レジスタの各々の要素を個別にアクセス可能な構成を利用して， SIMD 幅に対する複数のメモリアクセス命令によりデータ転送を行う。

20

【0009】

または，ブロックロード命令のようにインダイレクトアクセス命令を複数の命令に分解して， SIMD 幅に対する回数の命令を順次実行することも考えられる。しかし，上記のように複数の命令を実行する方法では，命令デコーダを SIMD 幅と同じ回数使用しなければならず，また命令実行のためのリザベーションステーションやコミットスタックエントリを SIMD 幅と同数使用しなければならず， CPU コアの多くの内部資源を使用する。さらに，命令デコーダを複数サイクル占有するため，依存関係のない後続命令をアウト・オブ・オーダで実行することができず，アウト・オブ・オーダで処理できる CPU コアの構成を生かすことができない。

【0010】

30

そこで，本実施の形態の一つの目的は，レジスタに格納された複数個の独立したデータをアドレスとして使用し，メモリ領域の複数箇所にアクセスする命令を1つの命令で実行する演算処理装置及び演算処理装置の制御方法を提供することにある。

【課題を解決するための手段】

【0011】

本実施の形態の第1の側面は，

命令をデコードする命令デコーダと，

前記命令デコーダによりメモリアクセス命令のエントリを生成されるメモリアクセスエントリ部 (RSA) と，

前記メモリアクセスエントリ部から出力された前記メモリアクセス命令のエントリをメモリに対して実行するメモリアクセスパイプライン (EAGA) と，

40

前記命令デコーダにより複数のデータを1つの命令で処理するマルチデータ命令のエントリを生成されるマルチデータ命令エントリ部 (RSF) と，

複数の演算器と複数のマルチデータ命令用レジスタとを有し，前記マルチデータ命令エントリ部から出力された前記マルチデータ命令のエントリの処理を前記複数の演算器により並列に処理し，前記複数のマルチデータ命令用レジスタに演算結果を格納する演算パイプライン (FLA) とを有し，

前記演算パイプラインは，前記複数のマルチデータ命令用レジスタに格納されている複数のメモリアドレスについて前記メモリにメモリアクセスするマルチデータインダイレクトメモリアクセス命令のエントリの出力に応答して，前記メモリアクセスパイプラインに

50

前記マルチデータインダイレクトメモリアクセス命令に対応する複数のメモリアクセス要求を生成し、前記複数の演算器が前記複数のマルチデータ命令用レジスタから取得した前記複数のメモリアドレスを前記メモリアクセスパイプラインに供給する演算処理装置である。

【発明の効果】

【0012】

第1の側面によれば、マルチデータインダイレクトメモリアクセス命令を少ない資源で効率的に実行する。

【図面の簡単な説明】

【0013】

【図1】本実施の形態における演算処理装置が実現可能なインダイレクトメモリアクセス方式を説明する図である。

【図2】ブロックロード命令によるパイプライン処理の例を示す図である。

【図3】本実施の形態における SIMD インダイレクトメモリアクセス命令によるパイプライン処理を示す図である。

【図4】本実施の形態における演算処理装置を搭載した情報処理装置を示す図である。

【図5】CPUコア30の全体構成を示す図である。

【図6】本実施の形態の SIMD インダイレクトメモリアクセス(ロードまたはストア)命令を実行する CPUコアの構成を示す図である。

【図7】浮動小数点演算リザベーションステーション R S F にエントリとして格納されるフラグ構成を示す図である。

【図8】SIMD インダイレクトロード命令と通常のロード命令の処理を示すフローチャートである。

【図9】SIMD インダイレクトロード命令と通常のロード命令の処理を示すフローチャートである。

【図10】SIMD インダイレクトメモリアクセスの一つである SIMD インダイレクトロード命令のパイプライン及びタイムチャートを示す図である。

【図11】本実施の形態の SIMD インダイレクトメモリアクセス(ロードまたはストア)命令を実行する CPUコアの構成を示す図である。

【図12】演算用インターフェース331とアドレス用インターフェース310の構成を示す図である。

【図13】SIMD 幅が2の SIMD インダイレクトメモリアクセス命令の場合のパイプラインとアドレス用インターフェース310の入出力信号変化を示す図である。

【図14】SIMD 幅が4の SIMD インダイレクトメモリアクセス命令の場合のパイプラインとアドレス用インターフェース回路310の入出力信号変化を示す図である。

【図15】図6の1つのメモリアクセスパイプライン E A G A を有する場合の演算用インターフェース331とアドレス用インターフェース310の構成を示す図である。

【図16】SIMD 幅2の場合での後続する R S A から投入されるメモリアクセスとの衝突を示す図である。

【図17】SIMD 幅4の場合での後続する R S A から投入されるメモリアクセスとの衝突を示す図である。

【図18】SIMD 幅4の場合での後続する SIMD インダイレクトメモリアクセス命令のエントリの投入により生成されるメモリアクセス要求との衝突を示す図である。

【図19】インダイレクトメモリアクセス要求の衝突を回避する抑止信号を生成する演算用インターフェース333の構成を示す図である。

【図20】R S F とその SIMD インダイレクトメモリアクセス命令のエントリの出力抑止回路を示す図である。

【図21】R S A とその通常のメモリアクセス命令のエントリの出力抑止回路を示す図である。

【図22】C S E 内の完了待ち合わせ回路を示す図である。

【発明を実施するための形態】

【0014】

本実施の形態において1つの命令で複数のデータについて処理を行う命令をSIMD命令（またはマルチデータ命令）と称する。SIMD命令は、例えばSIMD幅の数のデータについてSIMD幅の数の演算器が並列に処理を行い、SIMD幅の数のレジスタを1つのレジスタ単位とするSIMDレジスタに処理結果を格納する。

【0015】

図1は、本実施の形態における演算処理装置が実現可能なインダイレクトメモリアクセス方式を説明する図である。図1は、SIMDレジスタ332_1に格納された複数の独立したデータをアドレスとして使用し、1つの命令でメモリ領域14の複数箇所にアクセスするインダイレクトメモリアクセス方式である。図1の例では、SIMD幅が4の例であり、このようなインダイレクトメモリアクセスを行う命令を、SIMDインダイレクトメモリアクセス命令と称する。

【0016】

図1(A)は、SIMDインダイレクトロード命令（またはSIMD間接ロード命令）の例であり、SIMDレジスタ332_1に格納された4つの独立したデータをアドレスとして利用し、メモリ14の4つのアドレスADD_0-ADD_3のデータDATA_0-DATA_3を読み出し、別のSIMDレジスタ332_2に書込む。このSIMDインダイレクトロード命令は、例えば次のように記述される。

load %f100 %f200

ここで、%f100はアドレスが格納されたSIMDレジスタ332_1のレジスタ番号であり、%f200はデータを書込むSIMDレジスタ332_2のレジスタ番号である。

【0017】

図1(B)は、SIMDインダイレクトストア命令（またはSIMD間接ストア命令）の例であり、SIMDレジスタ332_1に格納された4つの独立したデータをアドレスとして利用し、別のSIMDレジスタ332_3内のデータDATA_0-DATA_3をメモリ14の4つのアドレスADD_0-ADD_3の領域に書き込む。このSIMDインダイレクトストア命令は、例えば次のように記述される。

store %f100 %f300

ここで、%f100はアドレスが格納されたSIMDレジスタ332_1のレジスタ番号であり、%f300は書き込みデータが格納されたSIMDレジスタ332_3のレジスタ番号である。

【0018】

上記の場合、SIMDレジスタ332_1に4つの独立したアドレスを書き込む処理は、例えば4回のロード命令を実行することで行われる。または、メモリの連続するアドレスに4つの独立したアドレスを書き込んでおり、メモリの先頭アドレスをソースアドレスとするSIMDロード命令を実行することで行われる。

【0019】

図2は、ブロックロード命令によるパイプライン処理の例を示す図である。ここでのブロックロード命令は、例えばメモリの連続領域のデータを複数の汎用レジスタに書き込む命令である。ブロックロード命令は、命令デコーダでデコードされると、命令デコーダが複数のメモリアクセス命令を生成し、その複数のメモリアクセス命令が、順次命令デコーダでデコードされ、メモリアクセス用リザベーションステーションにエントリされ、メモリアクセスされる。つまり、マルチフロー方式である。

【0020】

したがって、4つのデータをメモリからロードする場合は、ブロックロード命令は4つのメモリアクセス命令に分割され、それぞれ命令デコードとリザベーションステーションへのエントリとメモリアクセスとが4回繰り返される。そのため、後続の演算命令は、4サイクルにわたりデコード待ち状態となる。

【0021】

10

20

30

40

50

このようなブロックロード命令の手法を利用して上記の SIMD インダイレクトロード命令を実現しようとすると、同様に、命令デコードとリザベーションステーションへのエントリとインダイレクトロードの処理とを 4 回繰り返す必要があり、CPU コア内の資源を 4 サイクルにわたり占有し、後続の演算命令の命令デコードが可能になるのはマルチフローの最後の命令のデコードが完了した後になる。これでは、後続命令が依存関係のない命令の場合に利用可能なアウト・オブ・オーダの利点を生かすことができない。

【 0 0 2 2 】

[本実施の形態]

図 3 は、本実施の形態における SIMD インダイレクトメモリアクセス命令によるパイプライン処理を示す図である。本実施の形態の SIMD インダイレクトメモリアクセス命令では、1 つの SIMD インダイレクトメモリアクセス命令を命令デコーダがデコードし、命令デコーダが 1 つの命令を SIMD リザベーションステーションにエントリし、4 回のメモリアクセスを繰り返し実行する。したがって、命令デコーダは 1 サイクルで開放されるので、SIMD インダイレクトメモリアクセス命令と依存関係のない後続の演算命令を、次のサイクルで命令デコードすることができる。したがって、アウト・オブ・オーダのメリットを生かすことができる。さらに、図 3 には示されていないが、1 つの SIMD インダイレクトメモリアクセス命令を SIMD リザベーションステーションにエントリするので、リザベーションステーションに複数のエントリを使用する必要はなく、コミットスタックエントリのエントリも 1 つしか使用しないので、CPU コア内の資源を効率的に使用する。

10

20

【 0 0 2 3 】

図 4 は、本実施の形態における演算処理装置を搭載した情報処理装置を示す図である。コンピュータなどの情報処理装置 10 は、CPU / メモリボード 12 と、大容量の記憶装置であるハードディスク 11 とを有する。CPU / メモリボード 12 は、CPU チップである演算処理装置 20 と、演算処理装置 20 と外部のハードディスク 11 などを接続するインタコネクト 13 と、DRAM 等のメモリ 14 とを有する。

【 0 0 2 4 】

演算処理装置 20 は、例えば、4 つの CPU コア（演算処理部）30A - 30D と、4 つの CPU コアで共有される二次キャッシュ 24 と、入出力インターフェース 26 と、メインメモリ 14 へのアクセスを制御するメモリアクセスコントローラ 28 とを有する。

30

【 0 0 2 5 】

図 5 は、CPU コア 30 の全体構成を示す図である。CPU コア 30 は、分岐命令の予測を行う分岐予測部 302 と、プログラムカウンタ PC と分岐予測部 302 の予測に基づいて命令フェッチアドレスを生成する命令フェッチアドレス生成器 301 と、一次命令キャッシュ 303 と、フェッチされた命令をデコードする命令デコーダ 305 と、レジスタリネーミング部 306 と、メモリアクセス用リザベーションステーション RSA (Reservation Station for Address generate) と、整数演算用リザベーションステーション RSE (Reservation Station for Execute) と、浮動小数点 SIMD リザベーションステーション RSF (Reservation Station for Floating) と、分岐用リザベーションステーション RSB (Reservation Station for Branch) と、コミットスタックエントリ CSE (Commit Stack Entry) とを有する。

40

【 0 0 2 6 】

メモリアクセス用リザベーションステーション RSA のメモリアクセスパイプライン EA GA は、アドレス用インターフェース 310 と、オペランドアドレス生成器 311 と、アドレス選択回路 313 と、一次データキャッシュ 312 とを有する。整数演算用リザベーションステーション RSE の整数演算パイプライン EXA は、演算用インターフェース 333 と、固定小数点演算器 320 と、固定小数点リネーミングレジスタ 321 と、固定小数点レジスタ 322 とを有する。

【 0 0 2 7 】

また、浮動小数点 SIMD リザベーションステーション RSF の浮動小数点 SIMD 演

50

算パイプラインF L Aは、演算用インタフェース3 3 3と、最大S I M D幅の数のS I M D演算器3 3 0と、浮動小数点S I M Dリネーミングレジスタ3 3 1と、浮動小数点S I M Dレジスタ3 3 2とを有する。さらに、C P Uコア3 0は、2つのプログラムカウンタP C、N E X T P Cを有する。また、C P Uコア3 0は、演算器3 2 0、3 3 0が生成したデータを一時的に格納するストアバッファS T Bを有する。

【0 0 2 8】

S I M D演算器3 3 0のS I M D幅は、例えば2もしくは4を命令で指定可能である。浮動小数点S I M Dレジスタは最大S I M D幅の4つ要素で構成されている。これらのレジスタの要素をそれぞれ、要素0、要素1、要素2、要素3と呼ぶ。浮動小数点S I M D幅2の演算を行う場合、S I M Dレジスタの要素0と要素1を使用する。浮動小数点S I M D幅4の演算を行う場合、S I M Dレジスタのすべての要素を使用する。10

【0 0 2 9】

メモリアクセスパイプラインE A G A、整数演算パイプラインE X A、浮動小数点S I M D演算パイプラインF L Aは、それぞれ1つのパイプラインまたは2つ以上のパイプラインを有してもよく、それぞれ独立に命令を実行可能である。また、一次データキャッシュ3 1 2は、メモリアクセスパイプラインE A G Aのパイプライン数が2の場合、それに合わせて、2つのポートを設け同時に最大2つのアドレスによりアクセスを行うことができるようにもよい。さらに、メモリアクセスパイプラインE A G Aのパイプライン数を、最大S I M D幅と同じ4組にしてもよい。その場合は、一次データキャッシュ3 1 2も4つのポートを有して同時の最大4つのアドレスによりアクセスを行うことできるようになるのが望ましい。20

【0 0 3 0】

命令フェッチアドレス生成器3 0 1は、分岐予測部3 0 2またはプログラムカウンタP Cからの命令アドレスを選択し、一次命令キャッシュ3 0 3に対して命令フェッチリクエストを発行する。一次命令キャッシュ3 0 3は、命令フェッチリクエストに応じた命令を命令バッファ3 0 4に格納する。命令バッファ3 0 4から命令デコーダ3 0 5に対しては、プログラムにより指定された順番通りに、すなわちインオーダで命令が供給され、命令デコーダ3 0 5は、命令バッファから供給された命令をインオーダでデコードする。

【0 0 3 1】

命令デコーダ3 0 5は、デコードした命令の種類に応じて、各リザベーションステーションR S A、R S E、R S F及びR S B Rのいずれかに、命令に対応する必要なエントリを作成する。これとともに命令デコーダ3 0 5はデコードされたすべての命令に対応するエントリをC S Eに作成する。30

【0 0 3 2】

レジスタリネーミング部3 0 6は、リザベーションステーションR S A、R S E、R S Fのいずれかにエントリが作成された場合に、命令に応じた処理で使用されるレジスタのアドレスに、リネーミングレジスタ3 2 1、3 3 1のアドレスを割り当てる。

【0 0 3 3】

リザベーションステーションR S A、R S E、R S Fは、保持されたエントリのうち、処理に必要な資源（データ、演算器、レジスタ等）が準備されたものから順次パイプラインに出力し、後段のパイプラインE A G A、E X A、F L Aに出力したエントリに対応する処理を実行させる。これにより、命令がアウト・オブ・オーダで実行される。40

【0 0 3 4】

浮動小数点演算用リザベーションステーションR S Fには、例えば、S I M D演算命令に対応するエントリが格納される。1つのパイプラインF L AはS I M D幅の数のS I M D演算器3 3 0を有する。S I M D演算器3 3 0はR S Fからのエントリに基づいて演算対象とするデータを選択し、S I M D幅の数のS I M D演算器で演算を並列に実行する。演算結果は浮動小数点・S I M Dリネーミングレジスタ3 3 1に一時的に格納される。

【0 0 3 5】

メモリアクセス用リザベーションステーションR S Aには、命令デコーダ3 0 5により50

SIMDインダイレクトメモリアクセス命令以外のメモリアクセス命令に対応するエントリが生成され、格納される。そして、RSAは格納されている複数のエントリのいずれかを選択してパイプラインに出力する。メモリアクセス命令のエントリがパイプラインに出力されると、そのエントリに対応するメモリアクセス要求がパイプラインの各ステージを順番に転送する。オペランドアドレス生成回路311は、RSAのエントリのメモリアクセス要求に基づいて演算対象とするデータを選択し、アドレスを生成し、生成されたアドレスを用いてメモリアクセス要求を一次データキャッシュ312に入力する。一次データキャッシュ312は、メモリアクセス要求に対するメモリアクセスを実行する。

【0036】

コミットスタックエントリCSEは、命令デコーダ305によりデコードされたすべての命令に対応するエントリを保持し、各エントリに対応する処理の実行状況を管理し、これらの命令をインオーダで完了させる。例えば、CSEは、次に完了させるべきエントリに対応する処理の結果が、固定小数点リネーミングレジスタ321および浮動小数点SIMDリネーミングレジスタ331に格納されたと判定すると、格納されたデータを固定小数点レジスタ322または浮動小数点SIMDレジスタ332に出力させる。これにより、各リザベーションステーションでアウト・オブ・オーダに実行された命令が、インオーダで完了する。

【0037】

図5のCPUコア30では、固定小数点演算パイプラインEXAはSIMD構成になっていない。一方、浮動小数点演算パイプラインFLAはSIMD構成になっていて、最大SIMD幅の数のSIMD演算器330を有する。しかし、固定小数点演算パイプラインEXAもSIMD構成になっていてもよい。

【0038】

本実施の形態のCPUコア30は、浮動小数点SIMD演算パイプラインFLAの演算用インターフェース333の出力信号をメモリアクセスパイプラインEAGAのアドレス用インターフェース310に供給してメモリアクセス命令を生成させるためのバス334と、SIMD演算器330が取得したアドレスをアドレス選択回路313に供給するためのバス335とを有する。アドレス用インターフェース310は、演算用インターフェース333の出力信号に基づいて生成したSIMD幅の数のメモリアクセス命令をメモリアクセスパイプラインEAGAに出力する。また、アドレス選択回路313は、オペランドアドレス生成器331からのバスに代えて浮動小数点SIMD演算器330からのバス335を選択し、SIMD演算器330が浮動小数点SIMDレジスタ332や浮動小数点SIMDリネーミングレジスタ331から取得したアドレスを、前述のSIMD幅のメモリアクセス命令と共に一次データキャッシュ312へ供給する。

【0039】

[実施の形態のSIMDインダイレクトメモリアクセス命令を処理する構成と処理の概略]

図6は、本実施の形態のSIMDインダイレクトメモリアクセス(ロードまたはストア)命令を実行するCPUコアの構成を示す図である。図6には、後述するパイプラインの各サイクルが括弧付きで示されている。

【0040】

図6のCPUコア30は、メモリアクセス用リザベーションステーションRSA(またはメモリアクセスエントリ部)がメモリアクセス命令のエントリを出力する1つのメモリアクセスパイプラインEAGAを有する。また、浮動小数点SIMDリザベーションステーションRSF(またはマルチデータ命令エントリ部)がSIMD命令のエントリを出力するSIMD演算パイプラインFLAも1つ有する。そして、SIMD演算パイプラインFLAは、最大SIMD幅4と同じ数の浮動小数点SIMD演算器330を有する。

【0041】

本実施の形態のSIMDインダイレクトメモリアクセス命令の処理の概略は次のとおりである。命令デコーダ305は、SIMDインダイレクトメモリアクセス命令をデコード

10

20

30

40

50

して，そのエントリを浮動小数点 SIMD リザベーションステーション R S F に生成する。R S F は，SIMD インダイレクトメモリアクセス命令のエントリを SIMD 演算パイプライン F L A に出力すると，それに応答して SIMD 演算パイプライン F L A がバス 334 を介して SIMD 幅に対応した数のメモリアクセス要求をメモリアクセスパイプライン E A G A に生成する。具体的には，演算用インタフェース 333 が投入されたエントリのフラグ信号群を，バス 334 を介してアドレス用インタフェース 310 に供給し，アドレス用インタフェース 310 がそのフラグ信号群に基づいてメモリアクセスパイプライン E A G A に複数のメモリアクセス命令のアクセス要求を順次生成する。または，演算用インタフェース 333 がそのフラグ信号に基づいて複数のメモリアクセス命令のアクセス要求を順次生成し，バス 334 を介してアドレス用インタフェース 310 に供給してパイプライン E A G A に生成してもよい。10

【 0 0 4 2 】

また，SIMD インダイレクトメモリアクセス命令のエントリの投入または出力に応答して，SIMD 幅の数の SIMD 演算器 330 は，SIMD 幅の数のアドレスを浮動小数点 SIMD レジスタ 332 から並列に取得し，バス 335 を介してメモリアクセスパイプライン E A G A に供給する。具体的には，SIMD 演算器 330 は，取得した複数のアドレスをバス 335 を経由して順次アドレス選択回路 313 に供給する。アドレス選択回路 313 は，バス 335 から供給される複数のアドレスを，先に生成された複数のメモリアクセス命令のアクセス要求のタイミングに合わせて選択し，一次データキャッシュ 312 に出力する。20

【 0 0 4 3 】

具体的には，図 6 中の右上に示したとおり，アドレス選択回路 313 は，オペランドアドレス生成器 311 のアドレスフラグ A_E A G A _ ADD とバス 335 のいずれかを選択するセレクタ L5 を有する。そして，後述するように SIMD インダイレクトメモリ命令のエントリが SIMD 演算用パイプライン F L A に出力されたことに応答してアドレス用インタフェース 310 が生成するフラグ信号 B1_E A G A _ I N D I R E C T の「1」により，セレクタ L5 はバス 335 側を選択し，バス 335 を経由して供給されるアドレスを選択し，アドレスフラグ A_E A G A _ ADD として一次データキャッシュ 312 に転送する。これにより，アドレス用インタフェース 310 が B1 サイクルのステージで生成したメモリアクセス要求の転送タイミングに整合して，B1 サイクルのステージより後の A サイクルのステージでバス 335 を介してアドレスが供給され，SIMD インダイレクトメモリアクセス命令のアドレスを加えたメモリアクセス要求が一次データキャッシュ 312 に転送される。30

【 0 0 4 4 】

一次データキャッシュ 312 は，ロード命令の場合は一次キャッシュ 312 からまたはメモリ 14 から読み出した複数のデータを，浮動小数点 SIMD リネーミングレジスタ 331 に格納する。そして，コミットスタックエントリからの指令に応じて，読み出した複数のデータを浮動小数点 SIMD リネーミングレジスタ 331 から浮動小数点 SIMD レジスタ 332 に転送する。これらのレジスタ 331, 332 は，SIMD 幅の数のレジスタが一括してレジスタ番号で特定される。また，ストア命令の場合は，浮動小数点 SIMD レジスタ 332 に格納されている複数のデータを一次キャッシュ 312 またはメモリ 14 に順次書き込む。40

【 0 0 4 5 】

SIMD インダイレクトメモリアクセス命令の処理の概略をより具体的に説明すると次の通りである。

【 0 0 4 6 】

まず，命令デコーダ 305 は SIMD インダイレクトメモリアクセス命令をデコードし，R S F 及び C S E にエントリを作成する。C S E のエントリ番号（エントリされた命令識別情報）を I I D と呼ぶ。演算やメモリアクセスの完了の際に C S E にこの I I D と完了信号を通知することにより，C S E は命令完了の判定を行う。エントリの作成と同時に50

，命令デコーダ305は，一次データキャッシュ312が管理する資源であるフェッチポートFPをSIMD幅と同数の連続した個数確保する。フェッチポートFPは，一次データキャッシュがメモリアクセスを行う際に必要なメモリアドレスを記憶しておく資源であり，通常のメモリアクセス命令では1つのFPが確保される。SIMDインダイレクトメモリアクセス命令ではSIMD幅と同数のアドレスによりアクセスを行うため，複数のプリフェッチポートFPを使用する。

【0047】

図7は，浮動小数点演算リザベーションステーションRSFにエントリとして格納されるフラグ構成を示す図である。SIMDインダイレクトメモリアクセス命令を実行するため，インダイレクトフラグINDIRECTとフェッチポートフラグFPが追加されている。INDIRECTフラグはデコードした命令がSIMDインダイレクトメモリアクセス命令の場合に「1」となる。FPフラグはデコード時に確保した先頭のFP番号を示す。また，RSFには，これら以外にもSIMD演算器330に演算種の指示を行うOPCODE，命令のSIMD幅を識別する4SIMDフラグ(幅が2なら「0」，4なら「1」)，演算に使用するオペランドを示すR1_ADDRESS，CSEのエントリ番号を示すIIDなどを格納する。

【0048】

RSFは，SIMDインダイレクトメモリアクセス命令のエントリに必要な資源が準備され実行可能となると，浮動小数点SIMDパイプラインFLAの演算用インタフェース333にその命令のエントリを出力または投入する。

【0049】

演算用インタフェース333は，RSFから出力された命令がSIMDインダイレクトメモリアクセス命令のエントリである場合，そのエントリのインダイレクト命令，SIMD幅，IID，FP，FLAの命令が有効か否かを示すバリッドのフラグ信号群を，バス334を介して，アドレス用インタフェース310に転送する。アドレス用インタフェース310はこのフラグ信号群に基づいて，メモリアクセスパイプラインEAGAに複数のメモリアクセス命令のメモリアクセス要求をシリアルに生成する。

【0050】

上記の複数のメモリアクセス命令の生成と同時に，SIMD演算器330は，演算用インタフェース333からのフラグ信号に基づいて，SIMD幅の数のアドレスをSIMDレジスタ332から並列に読み出す。SIMDレジスタ332のレジスタ番号は，SIMDインダイレクトメモリアクセス命令のソースオペランドに示されている。SIMD演算器330は，SIMDレジスタ332からのアドレスの読み出しが完了すると，その複数のアドレスをアドレス選択回路313にバス335を介して転送する。そして，アドレス用インタフェース310が，アドレス用インタフェース310がパイプラインEAGAに順次生成した複数のメモリアクセス要求と，バス335を介して転送されてきた複数のアドレスとを，タイミングを整合させて，アドレス選択回路313に転送する。すなわち，アドレス選択回路313は，オペランドアドレス生成器331からのアドレスに代えて，SIMD演算器330から供給されてきたアドレスを選択し，複数のアドレスを一次データキャッシュ312にシリアルに転送する。一次データキャッシュ312は，複数のアドレスそれぞれについてデータの読み出しありはSIMDレジスタ内のデータの書き込みを行う。

【0051】

一次データキャッシュ312は，データの読み出しを完了すると読み出したデータを浮動小数点SIMDリネーミングレジスタに格納するとともに，CSEに読み出しが完了したエントリ識別情報IIDと完了通知を送る。データの書き込みの場合は，一次データキャッシュ312は，単にCSEに書き込みが完了したエントリ識別情報IIDと完了通知を送る。CSEはエントリ識別情報IIDと完了通知によりSIMD幅すべての要素の読み出しありは書き込みが完了するのを待ち，SIMDインダイレクトメモリアクセス命令を完了させる。

10

20

30

40

50

【 0 0 5 2 】

次に，演算用インターフェース333は，SIMDインダイレクト命令のエントリに基づいて生成されるメモリアクセス要求が，後続のメモリアクセス命令のアクセス要求と衝突することを防止するために，RSAとRSFに命令のエントリの出力を抑止する抑止信号を生成する。すなわち，第1に，演算用インターフェース333は，SIMDインダイレクトメモリアクセス命令のエントリに応答して，抑止信号336をRSAに出力し，RSAに，そのSIMDインダイレクトメモリアクセス命令に基づいてアドレス用インターフェース310で生成されるメモリアクセス要求と衝突する後続のメモリアクセス命令のエントリの出力を抑止させる。第2に，演算用インターフェース333は，SIMDインダイレクトメモリアクセス命令のエントリに応答して，抑止信号337をRSFに出力し，RSFに，後続のSIMDインダイレクトメモリアクセス命令のエントリの出力を抑止させる。これにより，先行するSIMDインダイレクトメモリアクセス命令によりアドレス用インターフェース310に生成された複数サイクルにわたるメモリアクセス要求と，後続のSIMDインダイレクトメモリアクセス命令に基づいて生成されるメモリアクセス要求とが衝突することを防止する。10

【 0 0 5 3 】

図8，図9は，SIMDインダイレクトロード命令と通常のロード命令の処理を示すフローチャートである。まず，命令フェッチ(S1)，命令バッファに格納(S2)，命令デコード(S3)が行われ，命令デコーダの結果，通常のロード命令の場合(S4のNO)，工程S5以下の処理が行われ，SIMDインダイレクトメモリアクセス命令の場合(S4のYES)，工程S21以下の処理が行われる。20

【 0 0 5 4 】

通常のロード命令の場合(S4のNO)，命令デコーダ305は，フェッチポートFPを1つ確保し，RSAヘロード命令のエントリを作成する(S5)。RSAは，ロード命令のエントリを投入する準備が完了したことを確認し(S6のYES)，先行するSIMDインダイレクトメモリアクセス命令に基づいて生成されるメモリアクセス命令と衝突していない場合(S7のNO)，ロード命令のエントリをメモリアクセスパイプラインEAGAに出力または投入する。

【 0 0 5 5 】

メモリアクセスパイプラインEAGAでは，オペランドアドレス生成器311が固定小数点レジスタ322などからデータを読み出し(S8)，オペランドアドレス生成器311がアドレスを生成し(S9)，アドレス選択回路313がオペランドアドレス生成器からのアドレスを選択する(S10)。そして，一次データキャッシュ312がそのアドレスを使用してデータを読み出す処理を実行する(S11)。一次データキャッシュ312が読み出したデータを浮動小数点SIMDリネーミングレジスタに格納してその処理を完了すると(S12)，通常のロード命令の場合は，CSEがフェッチポートFPを1個開放し，SIMDリネーミングレジスタ331からSIMDレジスタ332に読み出しデータを転送する(S19)。30

【 0 0 5 6 】

以上のように，通常のロード命令は，RSAにロード命令のエントリが生成され，メモリアクセスパイプラインEAGAのオペランドアドレス生成器331がアドレスの取得と生成を行い，一次データキャッシュへ312にロード要求を行う。40

【 0 0 5 7 】

なお，連続するメモリアドレスに対するSIMDロード命令のエントリがRSAに生成された場合は，オペランドアドレス生成器311がその先頭のアドレスを固定小数点レジスタ322などから読み出し，一次データキャッシュ312が連続する例えば2つのアドレスのデータを2つのSIMDリネーミングレジスタに格納する。ただし，この連続するメモリアドレスに対するSIMDロード命令は，本実施の形態における複数のSIMDレジスタ内の独立した複数のアドレスに対するSIMDインダイレクトメモリアクセス命令とは異なる命令である。50

【 0 0 5 8 】

次に，S I M D インダイレクトメモリアクセス命令の場合（S4のYES），命令デコーダはフェッチポートF PをS I M D 幅の個数確保し，R S F にS I M D 命令のエントリを生成する（S21）。R S F は，エントリを投入する準備が完了したことを確認し（S22のYES），先行するS I M D インダイレクトメモリアクセス命令に基づいて生成されるメモリアクセス命令と衝突していない場合（S23のNO），S I M D インダイレクトメモリアクセス命令のエントリをS I M D 演算パイプラインF L A に出力または投入する。

【 0 0 5 9 】

そして，S I M D 演算パイプラインF L A のS I M D 演算器3 3 0 が，S I M D 幅の数のアドレスをS I M D レジスタ3 3 2 から並列に読み出す（S23）。この読み出しには後述するとおり2サイクルを要する。そして，この読み出しとともに，S I M D 演算パイプラインF L A の演算用インタフェース3 3 3 がバス3 3 4 を介してアドレス用インタフェース3 1 0 にフラグ信号群を転送し，メモリアクセスパイプラインE A G A にメモリアクセスのリクエスト0を生成させる（S24）。生成されたリクエストはメモリアクセスパイプラインE A G A を転送する。さらに，S I M D 演算器3 3 0 はS I M D レジスタ3 3 2 から取得したS I M D 幅の数のアドレスを，バス3 3 5 を介してアドレス選択回路3 1 3 に供給し（S35），アドレス選択回路3 1 3 は，リクエスト0に基づきS I M D 演算器からのアドレスを選択し（S26），リクエスト0にフェッチポートF P を割り当てる（S27）。

【 0 0 6 0 】

上記の工程S23-S27を2回繰り返す。2回目はメモリアクセスのリクエスト1が生成される。さらに，S I M D 幅が4の場合（S28のYES），工程S23-S27と同じ処理工程S29-S32を2回繰り返す。これによりメモリアクセスのリクエスト2，3が生成される。

【 0 0 6 1 】

そして，一次データキャッシュ3 1 2 は，フェッチポートF P のアドレスを利用してデータを読み出す処理を実行開始する（S11）。C S E は，S I M D 幅が2の場合は2回の一次データキャッシュの処理完了通知（S12,S14）に応答して，S I M D 幅が4の場合は4回の一次データキャッシュの処理完了通知（S12,S14,S16,S17）に応答して，S I M D 処理の完了を検出し，フェッチポートF P をS I M D 幅の個数開放し，S I M D リネーミングレジスタ3 3 1 からS I M D レジスタ3 3 2 に読み出しデータを転送する（S20）。

【 0 0 6 2 】

以上のように，S I M D インダイレクトロード命令の場合は，命令デコーダがS I M D インダイレクトロード命令を1回デコードし，命令デコーダがR S F にS I M D インダイレクトロード命令のエントリを1つ生成し，S I M D 演算パイプラインF L A がメモリアクセスパイプラインE A G A にS I M D 幅の数のメモリアクセスのリクエストを生成し，複数のS I M D 演算器に複数のアドレスをS I M D レジスタから並列に取得させ，S I M D 演算器が取得した複数のアドレスをメモリアクセスパイプラインE A G A に転送して複数のメモリアクセスのリクエストに合体させ，メモリアクセスパイプラインE A G A が一次データキャッシュへのロードリクエストを行う。S I M D インダイレクトストア命令の場合も，一次データキャッシュがメモリにストアすることを除いて上記のロード命令と同じ動作である。

【 0 0 6 3 】

次に，本実施の形態のS I M D インダイレクトメモリアクセス命令のパイプライン処理を説明する。まず，S I M D インダイレクトロード命令のパイプラインステージを以下に示す。図6に括弧付きで示したステージを参照することで以下のパイプラインステージが明らかになる。

D (D e c o d e) : 命令デコーダが命令をデコードする。

D T (D e c o d e T r a n s f e r) : D サイクルの命令を転送し，R S F に格納する。

P (P r i o r i t y) : R S F がS I M D 演算器へ投入する命令のエントリを決定し出

10

20

30

40

50

力(投入)する。

P T (Priority Transfer) : Pサイクルのエントリのフラグ信号群を, 演算用インタフェースを介して転送し, SIMD演算器330に投入する。

B 1 , B 2 (Buffer) : SIMD演算器が演算に必要なデータをレジスタから入力する。例えば, データは浮動小数点 SIMDレジスタ332やリネーミングレジスタ331から取得される。この例では取得に2サイクルを要する。

X (Execution) : SIMD演算器がメモリアクセスに必要なデータを読み出す。

A (Address) : SIMD演算器がメモリにアクセスするアドレスをアドレス選択回路313に転送する。

10

T (Tag) : 一次データキャッシュがアドレスに基づいてタグにアクセスする。

M (Match) : 一次データキャッシュが読み出したキャッシュタグを比較する。

B (Buffer) : 一次データキャッシュから読み出したデータをバッファする。

R (Result) : 一次データキャッシュアクセスを完了する。

R T (Result) : Rサイクルのデータを転送し, リネーミングレジスタへの書き込みを行い, CSEへ完了通知を行う。

C (Commit) : すべての要素が完了したかどうかの命令完了の判定を行う。

W (Write) : 完了した命令による各種レジスタの更新やリソースの解放を行う。このとき, 浮動小数点 SIMDリネーミングレジスタ331から SIMDレジスタ332に読み出したデータを転送する。

20

【0064】

SIMDインダイレクトロード命令以外の通常ロード命令のパイプラインステージを以下に示す。

D (Decode) : 命令をデコードする。

D T (Decode Transfer) : Dサイクルの命令を転送し, RSAに命令のエントリを格納する。

P (Priority) : リザベーションステーションRSAから実行ユニットへ投入する命令のエントリを決定し出力(投入)する。

B 1 , B 2 (Buffer) : オペランドアドレス生成器がロードアドレス生成に必要なデータを決定しレジスタから入力する。

30

A (Address) : オペランドアドレス生成器がメモリにアクセスするアドレスを計算する。

T (Tag) : 一次データキャッシュが計算したアドレスに基づいてタグにアクセスする。

M (Match) : 一次データキャッシュが読み出したキャッシュタグを比較する。

B (Buffer) : 一次データキャッシュから読み出したデータをバッファする。

R (Result) : 一次データキャッシュアクセスを完了する。

R T (Result) : Rサイクルのデータを転送し, リネーミングレジスタへの書き込みを行い, CSEへ完了通知を行う。

C (Commit) : 命令完了の判定を行う。

40

W (Write) : 完了した命令による, 各種レジスタの更新やリソースの解放を行う。このとき, リネーミングレジスタからレジスタに転送する。

【0065】

図10は, SIMDインダイレクトメモリアクセスの一つである SIMDインダイレクトロード命令のパイプライン及びタイムチャートを示す図である。

【0066】

R S Fは, タイミング3のPサイクルで SIMDインダイレクトロード命令のエントリを SIMD演算パイプラインFLAに投入する。そして, タイミング5のB1サイクルで, 演算用インタフェース333が SIMDインダイレクトロード命令のフラグ信号を出力する。

50

【0067】

SIMD幅2のSIMDインダイレクトロード命令の場合は、SIMD演算パイプラインFLAが、タイミング6において、アドレス用インタフェース310内のパイプラインEAGAにメモリアクセス用のリクエスト0を生成し、次のタイミング7において、リクエスト1を生成する。生成されたメモリアクセスのリクエストは、メモリアクセスパイプラインEAGAにて、SIMDインダイレクトロード命令以外のロード命令におけるB1サイクルとなる。生成したメモリアクセスのエントリ識別情報IIDはSIMD演算パイプラインFLAから送られたものを使用し、フェッチポートFPはSIMD演算パイプラインFLAのFPの値とそれに1を加算した値を使用する。SIMD演算パイプラインFLAのSIMD演算器330が、タイミング7のX1サイクルでSIMDレジスタ（またはSIMDリネーミングレジスタ）から読み出した複数のデータのうち要素0を、タイミング8のX2サイクルでSIMDレジスタ（またはSIMDリネーミングレジスタ）から読み出した要素1をそれぞれシリアルにメモリアクセスパイプラインEAGAのアドレス選択回路313に転送する。タイミング8, 9（Aサイクル）で、アドレス選択回路313は、SIMD演算パイプラインFLAから転送されてきた要素0と要素1のアドレスを選択し、一次データキャッシュ312に転送する。一次データキャッシュにアクセスしたデータすべてが存在した場合、一次データキャッシュ312は、タイミング13, 14で読み出したデータをSIMDリネーミングレジスタ331に転送し、タイミング14ですべてのメモリアクセスの完了報告を行う。その結果、CSEは、命令完了の判定を行い、SIMDリネーミングレジスタ331のデータをSIMDレジスタ332に転送する。

【0068】

また、SIMD幅4のSIMDインダイレクトロード命令の場合は、SIMD演算パイプラインFLAが、タイミング6, 7, 8, 9において、アドレス用インタフェース内のパイプラインEAGAにメモリアクセス用のリクエスト0, 1, 2, 3をシリアルに生成する。生成された4つのメモリアクセスのリクエストは、メモリアクセスパイプラインEAGAにて、SIMDインダイレクトロード命令以外のロード命令におけるB1サイクルとなる。生成したメモリアクセスのリクエストのエントリ識別情報IIDはSIMD演算パイプラインFLAから送られたものを使用し、フェッチポートFPはSIMD演算パイプラインFLAのFP値とそれに1, 2, 3を加算した値を使用する。SIMD演算パイプラインFLAのSIMD演算器330が、タイミング7, 8, 9, 10のX1, X2, X3, X4サイクルでSIMDレジスタ（またはSIMDリネーミングレジスタ）から読み出したSIMDのデータのうち要素0と要素1と要素2と要素3をそれぞれ、シリアルにメモリアクセスパイプラインEAGAのアドレス選択回路313に転送する。タイミング8, 9, 10, 11（Aサイクル）で、アドレス選択回路313は、SIMD演算パイプラインFLAから転送されてきたアドレスをそれぞれ選択し、一次データキャッシュ312に転送する。一次データキャッシュにアクセスしたデータすべてが存在した場合、一次データキャッシュ312は、タイミング13, 14, 15, 16で読み出したデータをSIMDリネーミングレジスタ331に転送し、タイミング16ですべてのメモリアクセスの完了報告を行う。その結果、CSEは、命令完了の判定を行い、SIMDリネーミングレジスタ331のデータをSIMDレジスタ332に転送する。

【0069】

以上、SIMDインダイレクトロード命令について説明したが、SIMDインダイレクトストア命令でも、SIMD演算パイプラインFLAがメモリアクセスパイプラインEAGAにSIMD幅の数のメモリアクセスのリクエストを生成することと、SIMD演算器がSIMD幅の数のアドレスをSIMDレジスタから並列に取得してメモリアクセスパイプラインEAGAにシリアルに転送することと、一次データキャッシュにSIMD幅の数のメモリストアのリクエストを投入することは同じである。SIMDインダイレクトストア命令の場合は、一次データキャッシュはSIMDレジスタに格納されているSIMD幅の数のデータを一次キャッシュメモリまたはメモリに書き込む。

【0070】

10

20

30

40

50

[本実施の形態における SIMD インダイレクトメモリアクセスの詳細説明]

図 11 は、本実施の形態の SIMD インダイレクトメモリアクセス（ロードまたはストア）命令を実行する CPU コアの構成を示す図である。図 11 の CPU コアの構成における SIMD ダイレクトメモリアクセスの詳細な説明を行う。

【0071】

図 11 の CPU コア 30 は、図 6 と異なり、メモリアクセス用リザベーションステーション RSA（またはメモリアクセスエントリ部）が、メモリアクセス命令のエントリを出力するメモリアクセスパイプラインとして、2つのパイプライン EAGA, EAGB を有する。それに対応して、一次データキャッシュ 312 は、2つのメモリアクセス要求を並列に処理する構成を有する。また、浮動小数点 SIMD リザベーションステーション RSF（またはマルチデータ命令エントリ部）が、SIMD 命令のエントリを出力する SIMD 演算パイプラインとして、2つのパイプライン FLA, FLB を有する。そして、SIMD 演算パイプライン FLA, FLB は、最大 SIMD 幅 4 と同じ数の浮動小数点 SIMD 演算器 330 を、それぞれ有する。浮動小数点 SIMD レジスタ 332 と浮動小数点 SIMD リネーミングレジスタ 331 は、最大 SIMD 幅 4 と同じ数のレジスタが一括してレジスタ番号で指定可能である。それ以外の構成は、図 6 と同じである。

【0072】

したがって、SIMD 演算パイプライン FLA は、SIMD インダイレクトアクセスマリ命令のエントリに応答して、2つのメモリアクセス要求を2つのメモリアクセスパイプライン EAGA, EAGB に同時に生成することができ、SIMD 演算器 330 は、SIMD レジスタ 332 から取得した2つのアドレスを2つのメモリアクセスパイプライン EAGA, EAGB のアドレス選択回路 313 に並列に転送することができる。後述する図 13 に示す通りである。

【0073】

SIMD 幅が 2 の場合は、SIMD 演算パイプライン FLA は、バス 334 を介して、1 サイクルで 2 つのメモリアクセス要求を 2 つのパイプライン EAGA, EAGB に生成する。すなわち、演算用インタフェース 333 がフラグ信号群をバス 334 を介してアドレス用インタフェース 310 に転送し、アドレス用インタフェース 310 は、その転送されたフラグ信号群に基づいて、1 サイクルで 2 つのメモリアクセス要求を 2 つのパイプライン EAGA, EAGB に生成する。そして、SIMD 演算器 330 は、バス 335 を介して、1 サイクルで 2 つのアドレスを 2 つのパイプライン EAGA, EAGB のアドレス選択回路 313 に転送する。アドレス選択回路 313 内のセレクタ L5（図 6 参照）は、前述のとおり、インダイレクトフラグ信号 B1_EAGA_INDIRECT, B1_WAGB_INDIRECT の「1」により、バス 335 側を選択し、SIMD 演算器 330 から供給される 2 つのアドレスを 2 つのパイプライン EAGA, EAGB に出力する。これにより、アドレス用インタフェース 310 が生成した 2 つのメモリアクセス要求に、アドレス選択回路 313 のサイクル A のステージで、バス 335 から供給された 2 つのアドレスが加えられる。

【0074】

また、SIMD 幅が 4 の場合は、SIMD 演算パイプライン FLA は、2 サイクルで 4 つのメモリアクセス要求を 2 つのパイプライン EAGA, EAGB に生成し、2 サイクルで 4 つのアドレスを転送する。図 14 に示すとおりである。

【0075】

図 8, 図 9 のフローチャート図は、図 11 の CPU コアの構成にも適用できる。ただし、図 1 の CPU コアは、2 つのメモリアクセスパイプライン EAGA, EAGB を有するので、図 8 の工程 S24-S27, 工程 S29-S32 をそれぞれ 1 回行えば良い。

【0076】

[演算用インタフェース 331 とアドレス用インタフェース 310 によるインダイレクトメモリアクセス要求の生成]

図 12 は、演算用インタフェース 331 とアドレス用インタフェース 310 の構成を示

10

20

30

40

50

す図である。演算用インターフェース 331 は、RSF から投入される演算命令のエントリから後段の SIMD 演算器 330 などに対して制御信号を適切なタイミングで出力する。同様に、アドレス用インターフェース 310 は、RSA から投入されるメモリアクセス命令のエントリから後段のオペランドアドレス生成器 311 などに対して制御信号を適切なタイミングで出力する。

【0077】

アドレス用インターフェース 310 は、RSA から 2 つのパイプライン EAGA, EAGB に投入された、SIMD インダイレクトメモリアクセス命令以外の通常メモリアクセス命令のエントリのフラグ信号を、ラッチ回路群 F1_A, F1_B でラッチし、後段のオペランドアドレス生成器 311 に転送する。一方、演算用インターフェース 331 は、RSF から SIMD 演算パイプライン FLA に投入された SIMD インダイレクトメモリアクセス命令のエントリのフラグ信号を、アドレス用インターフェース 310 にバス 334 を介して転送する。そして、アドレス用インターフェース 310 内のアンドゲート A1, A2, ラッチ回路群 F2, F3, セレクタ L1, L2, L3, L4, オアゲート AR1, R2, 加算器 ADD1, ADD2 らの回路が、その転送されてきたフラグ信号に基づいて、2 つのメモリアクセスパイプライン EAGA, EAGB に、それぞれメモリアクセスのリクエストを生成する。

【0078】

図 12 では、アドレス用インターフェース 310 が破線で囲まれた回路を有するように示されている。しかし、破線で囲まれた回路の一部を演算用インターフェース 333 が有するようにしてもよい。したがって、演算用インターフェース 333 とアドレス用インターフェース 310 とバス 334 の構成により、SIMD 演算パイプライン FLA が、2 つのメモリアクセスパイプライン EAGA, EAGB に、それぞれメモリアクセスのリクエストを生成する。

【0079】

図 12 の各信号について説明する。

【0080】

パイプライン FLA 側のエントリのフラグ信号については、次の通りである。入力信号（バリッド信号）B1_FLA_VALID_EAITF は、浮動小数点・SIMD パイプラインの B1 サイクルでパイプライン FLA の SIMD 演算器 330 に演算要求が出たときに 1 となる。

【0081】

入力信号（インダイレクト信号）B1_FLA_INDIRECT_EAITF は、浮動小数点・SIMD パイプラインの B1 サイクルで演算要求が SIMD インダイレクトメモリアクセス命令であった場合 1 となる。

【0082】

入力信号（4 SIMD 信号）B1_FLA_4SIMD_EAITF は、浮動小数点・SIMD パイプラインの B1 サイクルで SIMD 幅が 4 であるときに 1 となる。

【0083】

入力信号（IID 信号）B1_FLA_IID_EAITF には、パイプライン FLA で実行される命令のエントリの識別情報 IID が転送される。

【0084】

入力信号（FP 信号）B1_FLA_FP_EAITF は、SIMD インダイレクトメモリアクセス命令において命令デコーダ 305 で確保したフェッチポート FP の先頭の番号を転送する。

【0085】

パイプライン EAGA, EAGB 側のエントリのフラグ信号については、次の通りである。入力信号（バリッド信号）P_EAGA_VALID, P_EAGB_VALID は、RSA からオペランドアドレス生成器 331 及び一次データキャッシュ 312 へメモリアクセス要求が出力された時に 1 となる。

【0086】

10

20

30

40

50

入力信号(FP 信号) P_EAGA_F , P_EAGB_F には , RSA からオペラントアドレス生成器 331 にメモリアクセス要求が出たとき , 一次キャッシュメモリ 312 で使用するフェッチポート番号 FP 番号が転送される。

【 0087 】

入力信号(IID 信号) P_EAGA_IID , P_EAGB_IID には , RSA からオペラントアドレス生成器 331 にメモリアクセス要求が出たとき , それぞれの要求に対応するエントリ識別情報 IID が転送される。

【 0088 】

アドレス用インターフェース回路 310 は , SIMD インダイレクトメモリアクセス命令のエントリが SIMD 演算パイプライン FLA に投入された場合 , 演算用インターフェース 333 が output するフラグ信号を用いて , メモリアクセスパイプライン EAGA , EAGB に 2 つもしくは 4 つのメモリアクセス要求を生成する。このメモリアクセス要求は , 以下に説明する 4 つの出力信号 B1_EAGA_*** と , 4 つの出力信号 B1_EAGB_*** に対応する。また , アドレス用インターフェース回路 310 は , 通常のメモリアクセス命令のエントリがメモリアクセスパイプライン EAGA , EAGB に投入された場合は , そのエントリのフラグ信号をそのままオペラントアドレス生成器 311 に転送する。

10

【 0089 】

出力信号(バリッド信号) B1_EAGA_VALID_OR は , オアゲート R1 により出力され , RSF が投入した SIMD インダイレクトメモリアクセス命令により生成されたメモリアクセス要求と , RSA からの通常のメモリアクセス命令に対するメモリアクセス要求の論理和である。このバリッド信号が 1 であるとき , メモリアクセスパイプライン EAGA のオペラントアドレス生成器 311 及び一次データキャッシュ 312 へのメモリアクセス要求が有効となる。

20

【 0090 】

出力信号(バリッド信号) B1_EAGB_VALID_ID_OR は , メモリアクセスパイプライン EAGB 側のバリッド信号であり , 上記と同様である。

【 0091 】

出力信号(インダイレクト信号) B1_EAGA_INDIRECT , 出力信号 B1_EAGB_INDIRECT は , 対応するバリッド信号 B1_EAGA_VALID_ID_OR , B1_EAGB_VALID_ID_OR 信号が 1 であるときに有効になる信号であり , メモリアクセス要求が SIMD インダイレクトメモリアクセス命令により生成されたことを示す。オアゲート R2 が output する。この信号は , 後続のオペラントアドレス生成器 311 を経由してアドレス選択回路 313 に転送され , アドレス選択回路 313 において SIMD 演算器 330 からバス 335 を介して転送されるアドレスを選択するために使用される。

30

【 0092 】

出力信号(IID 信号) B1_EAGA_IID , 出力信号(IID 信号) B1_EAGB_IID は , 対応するバリッド信号 B1_EAGA_VALID_ID_OR , B1_EAGB_VALID_ID_OR 信号が 1 であるときに有効になる信号である。 SIMD インダイレクトメモリアクセス命令である場合 , セレクタ L4 が演算用インターフェース 333 から転送されてきた入力信号 B1_FLA_IID_EAITF のエントリ識別情報 IID を選択する。もしそうでない場合 , セレクタ L4 は , RSA からの IID 信号 P_EAGA_IID , P_EAGB_IID を選択する。

40

【 0093 】

出力信号(FP 信号) B1_EAGA_F , 出力信号(FP 信号) B1_EAGB_F は , 対応するバリッド信号 B1_EAGA_VALID_ID_OR , B1_EAGB_VALID_ID_OR 信号が 1 であるときに有効になる信号である。 SIMD インダイレクトメモリアクセス命令の場合の場合で , SIMD 幅が 2 である場合は , 入力 FP 信号 B1_FLA_F_P_EAITF で転送されてきた FP 値と , 加算器 ADD2 で +1 加算した FP 値とが , セレクタ L3 で選択され出力される。一方 , SIMD 幅が 4 である場合は , 次の

50

クロックサイクルで、入力 F P 信号 B 1 _ F L A _ F P _ E A I T F で転送されてきた F P 値に加算器 ADD 1 で + 2 された F P 値と、加算器 ADD 2 で + 1 加算した F P 値とが、セレクタ L 3 で選択され出力される。例えば、S I M D 幅 4 であり、S I M D インダイレクトメモリアクセス命令でありかつ F P 信号 B 1 _ F L A _ F P _ E A I T F で転送された値が 5 であった場合、図 14 のタイミング 6 でパイプライン E A G A に生成されたリクエストの F P 信号 B 1 _ E A G A _ F P は 5、パイプライン E A G B に生成されたリクエストの F P 信号 B 1 _ E A G B _ F P は 6、タイミング 7 でパイプライン E A G A に生成されたリクエストの F P 信号 B 1 _ E A G A _ F P は 7、パイプライン E A G B に生成されたリクエストの F P 信号 B 1 _ E A G B _ F P は 8 になる。S I M D インダイレクトメモリアクセス命令でない場合は、R S A からの F P 信号 P _ E A G A _ F P , P _ E A G B _ F P がそれぞれセレクタ L 3 で選択される。
10

【 0 0 9 4 】

図 13 は、S I M D 幅が 2 の S I M D インダイレクトメモリアクセス命令の場合のパイプラインとアドレス用インタフェース 310 の入出力信号変化を示す図である。S I M D 演算パイプライン F L A の演算用インタフェース 333 が、タイミング 5 のサイクル B 1 で図 12 の入力信号 (B 1 _ F L A _ * * *) を出力し、アドレス用インタフェース 310 が、それらの入力信号に基づいてタイミング 6 で図 12 の出力信号 (B 1 _ E A G A _ * * * , B 1 _ E A G B _ * * *) によるメモリアクセス要求を生成する。

【 0 0 9 5 】

タイミング 5 の入力 I I D 信号 B 1 _ F L A _ I D D _ E A I T F (= 2) がセレクタ L 1 を介してラッチ F 2 でラッチされ、タイミング 6 の出力 I I D 信号 B 1 _ E A G A _ I I D , B 1 _ E A G B _ I I D が共に 2 になる。
20

【 0 0 9 6 】

タイミング 5 の入力バリッド信号 B 1 _ F L A _ V A L I D _ E A I T F (= 1) と入力カインダイレクト信号 B 1 _ F L A _ I N D I R E C T _ E A I T F (= 1) の論理積がアンドゲート A 1 を介してラッチ F 2 でラッチされ、オアゲート R 1 , R 2 を介して、タイミング 6 の出力バリッド信号 B 1 _ E A G A _ V A L I D _ O R , B 1 _ E A G B _ V A L I D _ O R が共に 1 になり、出力インダイレクト信号 B 1 _ E A G A _ I N D I R E C T , B 1 _ E A G B _ I N D I R E C T が共に 1 になる。
30

【 0 0 9 7 】

そして、タイミング 5 の入力 F P 信号 B 1 _ F L A _ E A I T F (= 4) がセレクタ L 2 を介してラッチ F 2 _ F P でラッチされ、セレクタ L 3 を介して、タイミング 6 の出力 F P 信号 B 1 _ E A G A _ F P (= 4) , B 1 _ E A G B _ F P (= 5) になる。

【 0 0 9 8 】

上記の動作により、S I M D 演算用パイプライン F L A は、演算用インタフェース 333 が出力するフラグ信号により、タイミング 6 で、アドレス用インタフェース 310 内の 2 つのメモリアクセスパイプライン E A G A , E A G B に、2 つのメモリアクセス要求を生成する。

【 0 0 9 9 】

図 14 は、S I M D 幅が 4 の S I M D インダイレクトメモリアクセス命令の場合のパイプラインとアドレス用インタフェース回路 310 の入出力信号変化を示す図である。S I M D 演算パイプライン F L A の演算用インタフェース 333 が、タイミング 5 のサイクル B 1 で図 12 の入力信号 (B 1 _ F L A _ * * *) を出力し、アドレス用インタフェース 310 が、それらの入力信号に基づいて、タイミング 6 , 7 で図 12 の出力信号 (B 1 _ E A G A _ * * * , B 1 _ E A G B _ * * *) によるメモリアクセス要求を生成する。
40

【 0 1 0 0 】

タイミング 5 の演算用インタフェース 333 が出力する入力信号と、タイミング 6 でアドレス用インタフェース 310 内のメモリアクセスパイプライン E A G A , E A G B に生成される出力信号は、図 13 の S I M D 幅 2 の場合と同じである。

【 0 1 0 1 】

ただし，S I M D 幅 4 の場合は，タイミング 6 のラッチ F 2 の入力 I I D 信号をセレクタ L 1 を介してラッチ F 2 が再度ラッチし，タイミング 6 のアンドゲート A 1 の出力と入力 4 S I M D 信号 B 1 _ F L A _ 4 S I M D _ E A I T F のラッチ信号の論理積を，アンドゲート A 2 を介してラッチ F 3 がラッチする。また，タイミング 6 の入力 F P 信号 B 1 _ F L A _ F P _ E A I T F の値に加算器 ADD 1 で + 2 した値を，セレクタ L 2 を介してラッチ F 2 _ F P がラッチする。それに対応して，タイミング 7 では，タイミング 6 と同様にして，メモリアクセスパイプライン E A G A , E A G B の出力バリッド信号，出力インダイレクト信号が 1 を維持し，出力 I I D 信号が 2 を維持し，出力 F P 信号が 6 , 7 になる。

【 0 1 0 2 】

10

上記の動作により，S I M D 演算用パイプライン F L A は，演算用インタフェース 3 3 3 の出力する信号により，タイミング 6 で，アドレス用インタフェース 3 1 0 内の 2 つのメモリアクセスパイプライン E A G A , E A G B に，2 つのメモリアクセス要求を生成し，さらに，タイミング 7 でメモリアクセスパイプライン E A G A , E A G B にさらに 2 つのメモリアクセス要求を生成する。

【 0 1 0 3 】

図 1 5 は，図 6 の 1 つのメモリアクセスパイプライン E A G A を有する場合の演算用インタフェース 3 3 1 とアドレス用インタフェース 3 1 0 の構成を示す図である。S I M D インダイレクトメモリアクセス命令の場合，図 1 2 と異なり次のような動作になる。図 1 0 も参照して説明する。

20

【 0 1 0 4 】

まず，タイミング 5 の入力バリッド信号 B 1 _ F L A _ V A L I D _ E A I T F と入力インダイレクト信号 B 1 _ F L A _ I N D I R E C T _ E A I T F の論理積が，アンドゲート A 1 を介して 2 つのラッチ F 2 _ 1 でラッチされ，そのラッチ F 2 _ 1 の出力がさらに次のタイミングでラッチ F 2 _ 2 でラッチされ，タイミング 6 , 7 で，出力バリッド信号 B 1 _ E A G A _ V A L I D _ O R と出力インダイレクト信号 B 1 _ E A G A _ I N D I R E C T が 2 サイクルにわたり 1 を出力する。

【 0 1 0 5 】

S I M D 幅が 4 の場合は，さらに，タイミング 7 のラッチ F 2 _ 2 の出力と入力 4 S I M D 信号のラッチ F 2 _ 2 の出力の論理積が，アンドゲート A 2 を介してラッチ F 3 _ 1 でラッチされ，そのラッチ F 3 _ 1 の出力がさらに次のタイミングでラッチされ，タイミング 8 , 9 で，出力バリッド信号 B 1 _ E A G A _ V A L I D _ O R と出力インダイレクト信号 B 1 _ E A G A _ I N D I R E C T が 2 サイクルにわたり 1 を出力する。

30

【 0 1 0 6 】

タイミング 5 の入力 I I D 信号 B 1 _ F L A _ I I D _ E A I T F は，セレクタ L 1 を介してラッチ F 2 で 4 回ラッチされ，タイミング 6 , 7 , 8 , 9 でセレクタ L 4 を介して出力 I I D 信号 B 1 _ E A G A _ I I D として出力される。

【 0 1 0 7 】

タイミング 5 の入力 F P 信号 B 1 _ F L A _ F P _ E A I T F は，セレクタ L 2 を介してラッチ F 2 _ F P がラッチし，その後，3 サイクルで加算器 ADD 1 でそれぞれ + 1 したフェッチポート F P の値をラッチ F 2 _ F P がラッチする。そして，タイミング 6 , 7 , 8 , 9 で，出力 F P 信号 B 1 _ E A G A _ F P が，入力 F P 値，それに + 1 , + 2 , + 3 された F P 値になる。

40

【 0 1 0 8 】

[衝突を回避するための R S A と R A F による新たなエントリ投入の抑止]

図 1 6 , 図 1 7 は，S I M D 幅 2 の場合と 4 の場合での後続する R S A から投入されるメモリアクセスとの衝突を示す図である。いずれも，図 1 1 の例で示している。

【 0 1 0 9 】

本実施の形態では，R A F が S I M D インダイレクトメモリアクセス命令のエントリを S I M D 演算パイプライン F L A に投入すると，S I M D 演算パイプライン F L A が，演

50

算用インターフェース 333 が出力する信号を利用して、アドレス用インターフェース 310 内のメモリアクセスパイプライン EAGA, EAGB に複数のメモリアクセス要求を生成する。そのため、その生成されたメモリアクセス要求と後続の RSA から投入されるメモリアクセス要求とが衝突する場合がある。図 11 の例では、SIMD 幅が 2 の場合は 1 回メモリアクセス要求が生成されるので 1 回衝突する場合があり、SIMD 幅が 4 の場合は 2 回メモリアクセス要求が生成されるので 2 回衝突する場合がある。図 6 のメモリアクセスパイプライン EAGA が 1 つの例では、SIMD 幅 2 では 2 回衝突し、SIMD 幅 4 では 4 回衝突する場合がある。

【0110】

図 11 の例で説明すると以下のとおりである。図 16, 17 には衝突が B1 への取消線で示されている。10

【0111】

(1) 図 16 の SIMD 幅 2 の場合は、タイミング 3 で RSF がパイプライン FLA に SIMD 幅 2 の SIMD インダイレクトメモリアクセス命令のエントリを出力し、タイミング 5 で RSA がパイプライン EAGA もしくは EAGB にメモリアクセス命令のエントリを出力した場合、タイミング 6 で、SIMD インダイレクトメモリアクセス命令により生成されるメモリアクセス要求のサイクル B1 の信号と、RSA から転送されるメモリアクセス要求のサイクル B1 の信号が衝突する。

【0112】

(2) 図 17 の SIMD 幅 4 の場合は、タイミング 3 で RSF がパイプライン FLA に SIMD 幅 4 のインダイレクト命令のエントリを出力し、タイミング 5 もしくは 6 において RSA がパイプライン EAGA もしくは EAGB にメモリアクセス命令のエントリを出力した場合、次の衝突が発生する。20

【0113】

すなわち、タイミング 5 で RSA がパイプライン EAGA もしくは EAGB にメモリアクセス命令のエントリを出力した場合、タイミング 6 で SIMD インダイレクトメモリアクセス命令により生成されるメモリアクセス要求のサイクル B1 の信号と、RSA から転送されるメモリアクセス要求のサイクル B1 の信号が衝突する。

【0114】

また、タイミング 6 で RSA がパイプライン EAGA もしくは EAGB にメモリアクセス命令のエントリを出力した場合、タイミング 7 で SIMD インダイレクトメモリアクセス命令により生成されるメモリアクセス要求のサイクル B1 の信号と、RSA から転送されるメモリアクセス要求のサイクル B1 の信号とが衝突する。30

【0115】

図 18 は、SIMD 幅 4 の場合での後続する SIMD インダイレクトメモリアクセス命令のエントリの投入により生成されるメモリアクセス要求との衝突を示す図である。いずれも、図 11 の 2 つのメモリアクセスパイプライン EAGA, EAGB を有する例で示している。

【0116】

本実施の形態では、SIMD インダイレクトメモリアクセス命令のエントリの投入に応答して、SIMD 演算パイプライン FLA が、演算用インターフェース 333 が出力する信号を利用して、メモリアクセスパイプライン EAGA, EAGB にメモリアクセス要求を生成する。そのため、その生成されたメモリアクセス要求が、後続の SIMD インダイレクトメモリアクセス命令のエントリの投入に応答してメモリアクセスパイプライン EAGA, EAGB に生成されるメモリアクセス要求と衝突する場合がある。図 11 の例では、SIMD 幅が 4 の場合に 2 回メモリアクセス要求が生成されるので、後続の SIMD インダイレクトメモリアクセス命令に対応するメモリアクセス要求と、1 回衝突する場合がある。図 6 の例では、SIMD 幅 2 では 1 回衝突し、SIMD 幅 4 では 3 回衝突する場合がある。40

【0117】

図11の例で説明すると図18に示されるように以下のとおりである。図18には衝突がB1への取消線で示されている。

【0118】

(3) タイミング3でRSFがSIMD幅4のSIMDインダイレクトメモリアクセス命令のエントリを出力し、タイミング4でRSFがSIMD幅2もしくは4のSIMDインダイレクトメモリアクセス命令のエントリを出力した場合、次のとおり衝突が発生する。すなわち、タイミング3でRSFから出力された4 SIMDインダイレクトメモリアクセス命令により生成されたメモリアクセス要求のサイクルB1の信号と、次のタイミング4でRSFから出力された2または4 SIMDインダイレクトメモリアクセス命令により生成されたメモリアクセス要求のサイクルB1の信号とが、タイミング7で衝突する。

10

【0119】

図19は、インダイレクトメモリアクセス要求の衝突を回避する抑止信号を生成する演算用インターフェース333の構成を示す図である。演算用インターフェース333は、RSFが投入するSIMDインダイレクトメモリアクセス命令のエントリのPサイクルのフラグ信号を入力し、ラッチ群F10でラッチし、さらにラッチ群F11でラッチする。それにより、演算用インターフェース333は、Pサイクルから2サイクル後のB1サイクルの出力信号を、SIMD演算パイプラインFLAのSIMD演算器330と、メモリアクセスパイプラインEAGA、EAGBのアドレス用インターフェース310に転送する。演算用インターフェース333が2つのラッチ群F10、F11を有するのは、例えば、タイミングを調整するためである。

20

【0120】

そして、演算用インターフェース333は、Pサイクルの3つのフラグ信号からRSFへの後続のSIMDインダイレクトメモリアクセス命令のエントリの投入を抑止する抑止信号INH_FLA_INDIRECT_OPと、PTサイクルの2つのフラグ信号からとともに、B1サイクルの3つのフラグ信号からも、RSAへの後続のメモリアクセス命令の投入を抑止する抑止信号INH_RSA_PRIORITYを生成する。

【0121】

演算用インターフェース333の動作は次のとおりである。

【0122】

入力信号(バリッド信号)P_FLA_VALIDは、浮動小数点・SIMDパイプラインのPサイクルでパイプラインFLAへのSIMD演算器に演算要求が出力されたときに1となる。

30

【0123】

入力信号(インダイレクト信号)P_FLA_INDIRECTは、入力バリッド信号P_FLA_VALIDが1のときに有効となる信号であり、演算要求がSIMDインダイレクトメモリアクセス命令の場合に、浮動小数点・SIMDパイプラインのPサイクルで1となる。

【0124】

入力信号(4 SIMD信号)P_FLA_4SIMDは、入力バリッド信号P_FLA_VALIDが1のときに有効となる信号であり、SIMD演算器の演算幅が4であるときに浮動小数点・SIMDパイプラインのPサイクルで1となる。

40

【0125】

入力信号(IDD信号)P_FLA_IIDは、入力バリッド信号P_FLA_VALIDが1のときに有効となる信号であり、パイプラインFLAで実行される演算のCSEのエントリ番号を示す。

【0126】

入力信号(FP信号)P_FLA_FPは、入力バリッド信号P_FLA_VALIDが1かつ入力インダイレクト信号P_FLA_INDIRECTが1のときに有効となる信号であり、SIMDインダイレクトメモリアクセス命令において命令デコーダで確保された一次データキャッシュ内のフェッчポートFPの先頭番号を示す。

50

【0127】

演算用インターフェース333は、5つの入力信号をラッチF10, F11でラッチして中継し、5つの出力信号B1_FLA_VALID_EAITF, B1_FLA_INDIRECT_EAITF, B1_FLA_4 SIMD_EAITF, B1_FLA_IID_EAITF, B1_FLA_FP_EAITFを、アドレス用インターフェース310に転送し、メモリアクセスのリクエストを生成させる。

【0128】

同様に、演算用インターフェース333は、4つの入力信号をラッチF10, F11でラッヂして中継し、4つの出力信号B1_FLA_VALID, B1_FLA_INDIRECT, B1_FLA_4 SIMD, B1_FLA_IIDを、SIMD演算器に転送する。
10

【0129】

演算用インターフェース333では、アンドゲートA4がPサイクルの2つの入力信号P_FLA_VALID, P_FLA_INDIRECTの論理積を後続の通常メモリアクセス命令の抑止信号INH_RSA_PRIORITYとして生成し、RSAに転送する。これにより、RSAは、後続のメモリアクセス命令のエントリのメモリアクセスパイプラインEAGA, EAGBへの投入を抑止する。

【0130】

図16に示されるとおり、タイミング3のPサイクルの2つの信号が全て1の場合に、タイミング4で抑止信号INH_RSA_PRIORITYが1になり、タイミング5においてRSAがメモリアクセス命令のエントリのパイプラインEAGA, EAGBへの投入を抑止する。これにより、タイミング6でB1サイクルの信号が発生せず、衝突が回避される。
20

【0131】

さらに、演算用インターフェース333では、アンドゲートA5がPサイクルの3つの入力信号P_FLA_VALID, P_FLA_INDIRECT, P_FLA_4 SIMDの論理積を後続の通常メモリアクセス命令の抑止信号INH_RSA_PRIORITYとして生成し、RSAに転送する。これにより、RSAは、後続のメモリアクセス命令のメモリアクセスパイプラインEAGA, EAGBへの投入を抑止する。

【0132】

図17に示されるとおり、タイミング3のPサイクルの3つの信号が全て1の場合に、タイミング5で抑止信号INH_RSA_PRIORITYが1になり、タイミング6においてRSAがメモリアクセス命令のパイプラインEAGA, EAGBへの投入を抑止する。これにより、タイミング7でB1サイクルの信号が発生せず、衝突が回避される。図17では、図16と同様にして、タイミング4で抑止信号INH_RSA_PRIORITYが1になり、タイミング5におけるRSAでのメモリアクセス命令の投入が抑止されている。
30

【0133】

そして、演算用インターフェース333では、アンドゲートA3がPサイクルの3つの入力信号P_FLA_VALID, P_FLA_INDIRECT, P_FLA_4 SIMDの論理積を後続のSIMDインダイレクトメモリアクセス命令の抑止信号INH_FLA_INDIRECT_OPとして生成し、RSFに転送する。これにより、RSFは、後続のSIMDインダイレクトメモリアクセス命令のエントリのSIMD演算パイプラインFLAへの投入を抑止する。
40

【0134】

図18に示されるとおり、タイミング3のPサイクルの3つの信号が全て1の場合に、抑止信号INH_FLA_INDIRECT_OPが1になり、次のタイミング4においてRSFがSIMDインダイレクトメモリアクセス命令のエントリのパイプラインFLAへの投入を抑止する。これにより、タイミング7で生成されたB1サイクルの信号が発生せず、衝突が回避される。
50

【0135】

図20は、RSFとそのSIMDインダイレクトメモリアクセス命令のエントリの出力抑止回路を示す図である。RSFは、例えば20個のエントリ保持部337を有し、リザベーションステーションRSFに生成された命令のエントリに対応するフラグが格納されている。フラグの例は図7に示されている。

【0136】

各エントリ保持部337に対応するRSFエントリ出力条件検出回路338は、これらのフラグを用い、RSF内のエントリそれぞれについてパイプラインへの出力可能条件が成立したことを検出する。このRSFエントリ出力条件検出回路338は、RSFそれぞれに格納された命令のエントリが処理可能となった場合に1を出力し、出力が可能でない場合は0を出力する。
10

【0137】

抑止回路339は、演算用インターフェース333で生成された抑止信号INH_FLA_INDIRECT_OPと、RSFのエントリ保持部337に格納されているINDIRECTフラグが共に1の場合に、出力条件検出回路338の出力を強制的に0にする。これにより、対応するRSFエントリが出力可能であるか否かを示すREADY信号がラッチRSFxx_READYにラッチされる。xxは00-19である。

【0138】

FLA出力選択回路340は、このREADY信号が1のRSFエントリから、次に出力するRSFエントリを選択し、演算用インターフェース333へ出力する。ただし、SIMDインダイレクトメモリアクセス命令の場合は、INDIRECTフラグが1になるので、抑止信号INH_FLA_INDIRECT_OPが1となったとき、そのエントリのREADY信号が0となるため、FLA出力選択回路340はそのSIMDインダイレクトメモリアクセス命令のエントリを選択することはない。SIMDインダイレクトメモリアクセス命令以外の命令の場合は、INDIRECTフラグが0になるので、エントリ出力条件検出回路338の出力がREADY信号として使用される。したがって、SIMDインダイレクトメモリアクセス命令以外の命令については、必要な資源が準備されたエントリがあればその命令のエントリが出力される。これによりRSFが、抑止信号INH_FLA_INDIRECT_OPに応じて、SIMDインダイレクトメモリアクセス命令のエントリの出力を抑止する。
20
30

【0139】

図21は、RSAとその通常のメモリアクセス命令のエントリの出力抑止回路を示す図である。RSAは、例えば20個のエントリ保持部314を有する。各エントリ保持部314に対応するRSAエントリ出力条件検出回路315は、RSAエントリそれぞれについてパイプラインへの出力可能条件が成立したことを検出する。このRSAエントリ出力条件検出回路315は、RSAそれぞれに格納された命令が処理可能となった場合に1を出力し、出力が可能でない場合は0を出力する。

【0140】

抑止回路316は、演算用インターフェース333で生成された抑止信号INH_RSA_PRIORITYが1の場合に、RSAエントリ出力条件検出回路315の出力を強制的に0にする。これにより、対応するRSAエントリが出力可能であるか否かを示すREADY信号がラッチRSAxx_READYにラッチされる。
40

【0141】

EAGA/EAGB出力選択回路317は、READY信号が1のRSAエントリから出力するRSAエントリを選択し、メモリアクセスパイプラインEAGAまたはEAGBに出力し、アドレス用インターフェースへ転送する。抑止信号INH_RSA_PRIORITYが1のとき、RSAエントリ出力条件検出回路315から出力された値に関わらず、すべてのRSAのREADY信号が0になる。これによりEAGA/EAGB出力選択回路315は出力可能なエントリがないため、メモリリクエストをメモリアクセスパイプラインEAGA/EAGBにエントリを出力しない。これによりRSAが、抑止信号IN
50

H_RSA_PRIORITYに応じて、メモリアクセス命令のエントリの出力を抑止する。

【0142】

図22は、CSE内の完了待ち合わせ回路を示す図である。図22には、CSEの1つのエントリに対する完了待ち合わせ回路が示されている。

【0143】

まず、CSEのエントリにインダイレクトフラグCSE_INDIRECTが含まれている。CSEのエントリがSIMDインダイレクトメモリアクセス命令の場合、そのエントリのインダイレクトフラグCSE_INDIRECTが1になる。また、その命令が4 SIMDの場合に4 SIMD信号CSE_4 SIMDが1になる。CSEにエントリされた命令がSIMDインダイレクトメモリアクセス命令であった場合、一次データキャッシュ312が同じCSEのエントリ番号IIDに対して2 SIMDなら2回、4 SIMDなら4回の完了報告をCSEに行う。10

【0144】

入力信号(インダイレクト信号)CSE_INDIRECT、入力信号(4 SIMD信号)CSE_4 SIMDは、命令デコーダ305によりCSEに登録されたエントリのフラグである。入力信号CSE_INDIRECTが1のときCSEのエントリがSIMDインダイレクトメモリアクセス命令であることを示す。入力信号CSE_4 SIMDが1のとき、CSEのエントリのSIMD幅が4であることを示し、0のときSIMD幅が2であることを示す。20

【0145】

本実施の形態の一次データキャッシュ312は、2つの独立したメモリアクセスを同時に処理する。そのため、一次データキャッシュ312は、メモリアクセス完了信号を2つ独立して通知する。

【0146】

入力信号RT_STV_0、RT_STV_1は一次データキャッシュから転送されるメモリアクセスの完了信号である。

【0147】

入力信号RT_STV_0_CSE_SEL、RT_STV_1_CSE_SELは、一次データキャッシュにおいて処理中のエントリ番号IIDが、CSEのエントリ番号と一致したとき1となる。30

【0148】

RT_STV_0とRT_STV_0_CSE_SELが1となったとき、もしくはRT_STV_1とRT_STV_1_CSE_SELが1となったとき、アンドゲートA8またはA9の出力により、CSEへのメモリアクセス完了報告が有効となる。メモリアクセス完了報告が有効になると、加算器351が3ビットの入力信号に+1加算してメモリアクセス完了回数記憶素子353に出力する。

【0149】

命令デコードがCSEにエントリを作成したときに、メモリアクセス完了回数記憶素子353を0にリセットする。その後、一次データキャッシュ312からの完了報告により、RT_STV_0とRT_STV_0_CSE_SEL両方が1になった場合、もしくはRT_STV_1とRT_STV_1_CSE_SEL両方が1になった場合、加算器351がメモリアクセス完了回数を+1加算する。40

【0150】

メモリアクセス命令の種類により、メモリアクセス完了回数が規定の値(1, 2, 4回)となったとき、出力信号(完了信号)CSE_MEM_COMPが1となる。アンドゲートA6により、SIMDインダイレクトメモリアクセス命令かつSIMD幅が4の場合、4回のメモリアクセス完了が通知されたとき、加算器351のビット2の出力が1になり、完了信号CSE_MEM_COMPが1となる。SIMDインダイレクトメモリアクセス命令かつSIMD幅が2の場合、2回のメモリアクセス完了が通知されたとき、加算50

器 351 のビット 1 の出力が 1 になり，完了信号が 1 となる。そして， SIMD インダイレクト命令でないメモリアクセス命令の場合は，1 回メモリアクセス完了が通知されたとき，加算器 351 のビット 0 の出力が 1 になり，完了信号 CSE_MEM_COMP が 1 となる。

【 0151 】

完了判定回路 354 は，この完了信号 CSE_MEM_COMP を入力し，命令が完了可能となったことを示す信号を生成する。完了判定回路 354 は，処理が完了した命令をプログラムの順番に完了したと判定し，例えばリネーミングレジスタからレジスタに処理結果を転送し，エントリを開放する。

【 0152 】

以上の通り，本実施の形態によれば， SIMD インダイレクトメモリアクセス命令のエントリを RSF に生成し，そのエントリが SIMD 演算用パイプライン FLA に出力されると，メモリアクセスパイプライン EAGA, EGB に SIMD 幅に応じた数のメモリアクセスを生成し， SIMD 演算器 330 が複数の SIMD レジスタ 332 に格納されている独立した複数のアドレスを取得してメモリアクセスパイプライン EAGA, EGB に転送し，一次データキャッシュ 312 がその複数のアドレスを使用して複数の SIMD レジスタ 332 に格納されている複数のデータについてメモリアクセスを行う。したがって，命令デコーダや CSE, RSA, RSF のエントリなどの資源を効率的に使用して SIMD インダイレクトメモリアクセス命令を実行する。

【 0153 】

以上の実施の形態をまとめると，次の付記のとおりである。

【 0154 】

(付記 1)

命令をデコードする命令デコーダと，

前記命令デコーダによりメモリアクセス命令のエントリを生成されるメモリアクセスエントリ部と，

前記メモリアクセスエントリ部から出力された前記メモリアクセス命令のエントリをメモリに対して実行するメモリアクセスパイプラインと，

前記命令デコーダにより複数のデータを 1 つの命令で処理するマルチデータ命令のエントリを生成されるマルチデータ命令エントリ部と，

複数の演算器と複数のマルチデータ命令用レジスタとを有し，前記マルチデータ命令エントリ部から出力された前記マルチデータ命令のエントリの処理を前記複数の演算器により並列に処理し，前記複数のマルチデータ命令用レジスタに演算結果を格納する演算パイプラインとを有し，

前記演算パイプラインは，前記複数のマルチデータ命令用レジスタに格納されている複数のメモリアドレսについて前記メモリにメモリアクセスするマルチデータインダイレクトメモリアクセス命令のエントリの出力に応答して，前記メモリアクセスパイプラインに前記マルチデータインダイレクトメモリアクセス命令に対応する複数のメモリアクセス要求を生成し，前記複数の演算器が前記複数のマルチデータ命令用レジスタから取得した前記複数のメモリアドレսを前記メモリアクセスパイプラインに供給する演算処理装置。

【 0155 】

(付記 2)

前記演算パイプラインは，前記メモリアクセスパイプラインの第 1 サイクルのステージに前記複数のメモリアクセス要求を生成し，前記メモリアクセスパイプラインの前記第 1 サイクルより後の第 2 サイクルのステージに前記複数のメモリアドレսを供給する付記 1 に記載された演算処理装置。

【 0156 】

(付記 3)

前記演算パイプラインは，前記メモリアクセスパイプラインに生成した複数のメモリアクセス要求のパイプライン転送タイミングにあわせて前記複数のメモリアドレսを供給す

10

20

30

40

50

る

付記 2 に記載された演算処理装置。

【 0 1 5 7 】

(付記 4)

さらに，前記メモリアクセスパイプラインに接続されたキャッシュユニットを有し，

前記演算パイプラインは，前記メモリアクセスパイプラインに生成する前記複数のメモリアクセス要求に前記キャッシュユニット内のアクセス先メモアドレスを格納する複数のフェッチポートの識別情報を含める

付記 1 に記載された演算処理装置。

【 0 1 5 8 】

10

(付記 5)

前記演算パイプラインは，前記メモリアクセスパイプラインに，前記複数のメモリアクセス要求をシリアルに生成し，前記複数のメモアドレスをシリアルに供給する

付記 1 , 2 , 3 のいずれかに記載された演算処理装置。

【 0 1 5 9 】

(付記 6)

前記メモリアクセスパイプラインが複数設けられ，

前記演算パイプラインは，前記複数のメモリアクセスパイプラインに，前記複数のメモリアクセス要求の少なくとも一部のメモリアクセス要求を並列に生成し，前記複数のメモアドレスの少なくとも一部のアドレスを並列に供給する

20

付記 1 , 2 , 3 のいずれかに記載された演算処理装置。

【 0 1 6 0 】

(付記 7)

さらに，前記メモリアクセスパイプラインに接続されたキャッシュユニットを有し，

前記キャッシュユニットは，前記複数のメモリアクセス要求に応答して，前記複数のマルチデータ命令用レジスタとの間でデータ転送を行う

付記 1 に記載された演算処理装置。

【 0 1 6 1 】

30

(付記 8)

前記演算パイプラインは，前記メモリアクセスエントリ部に抑止信号を出力して，前記メモリアクセスエントリ部に，前記メモリアクセスパイプラインに生成する前記複数のメモリアクセス要求と衝突するメモリアクセス命令のエントリの出力を抑止させる

付記 1 に記載された演算処理装置。

【 0 1 6 2 】

(付記 9)

前記演算パイプラインは，前記マルチデータ命令エントリ部に抑止信号を出力して，前記マルチデータ命令エントリ部に，前記メモリアクセスパイプラインにシリアルに生成する前記メモリアクセス要求と衝突するマルチデータインダイレクトメモリアクセス命令のエントリの出力を抑止させる

付記 5 に記載された演算処理装置。

40

【 0 1 6 3 】

(付記 10)

前記演算パイプラインに出力される前記マルチデータインダイレクトメモリアクセス命令のエントリは，マルチデータインダイレクトメモリアクセスを示すインダイレクトメモリアクセス信号と，前記複数のデータの数を示すマルチデータ幅情報信号とを有し，

前記演算パイプラインは，前記メモリアクセスパイプラインに，前記マルチデータ幅情報信号が示す数の前記メモリアクセス要求を生成し，前記マルチデータ幅情報信号が示す数の前記複数のメモアドレスを供給する

付記 1 に記載された演算処理装置。

【 0 1 6 4 】

50

(付記 11)

命令をデコードする命令デコーダと，
前記命令デコーダによりメモリアクセス命令のエントリを生成されるメモリアクセスエントリ部と，

前記メモリアクセスエントリ部から出力された前記メモリアクセス命令のエントリをメモリに対して実行するメモリアクセスパイプラインと，

前記命令デコーダにより複数のデータを1つの命令で処理するマルチデータ命令のエントリを生成されるマルチデータ命令エントリ部と，

複数の演算器と複数のマルチデータ命令用レジスタとを有し，前記マルチデータ命令エントリ部から出力された前記マルチデータ命令のエントリの処理を前記複数の演算器により並列に処理し，前記複数のマルチデータ命令用レジスタに演算結果を格納する演算パイプラインとを有する演算処理装置の制御方法において，
10

前記演算パイプラインが，前記複数のマルチデータ命令用レジスタに格納されている複数のメモリアドレスについて前記メモリにメモリアクセスするマルチデータインダイレクトメモリアクセス命令のエントリの投入に応答して，前記メモリアクセスパイプラインに前記マルチデータインダイレクトメモリアクセス命令に対応する複数のメモリアクセス要求を生成し，

前記演算パイプラインが，前記複数の演算器が前記複数のマルチデータ命令用レジスタから取得した前記複数のメモリアドレスを前記メモリアクセスパイプラインに供給する演算処理装置の制御方法。
20

【0165】

(付記 12)

前記演算パイプラインが，前記メモリアクセスパイプラインの第1サイクルのステージに前記複数のメモリアクセス要求を生成し，前記メモリアクセスパイプラインの前記第1サイクルより後の第2サイクルのステージに前記複数のメモリアドレスを供給する
付記 11 に記載された演算処理装置の制御方法。

【符号の説明】

【0166】

301：命令フェッチャドレス生成器

302：分岐予測機構

30

303：一次命令キャッシュ

304：命令バッファ

305：命令デコーダ

306：レジスタリネーミング部

R S A：メモリアクセス用リザベーションステーション（アドレス生成リザベーションステーション），メモリアクセスエントリ部

310：アドレス用インターフェース

311：オペランドアドレス生成器

312：一次データキャッシュ

313：アドレス選択回路

40

E A G A，E A G B：オペランドアドレス生成器，メモリアクセスパイプライン

S T B：ストアバッファ

R S E：固定小数点演算用リザベーションステーション

320：固定小数点演算器

322：固定小数点レジスタ

321：固定小数点リネーミングレジスタ

R S F：浮動小数点演算用リザベーションステーション，マルチデータ命令エントリ部

330：浮動小数点S I M D演算器，マルチデータ命令用演算器

332：浮動小数点S I M Dレジスタ，マルチデータ命令用レジスタ

331：浮動小数点S I M Dリネーミングレジスタ

50

3 3 3 : 演算用インタフェース

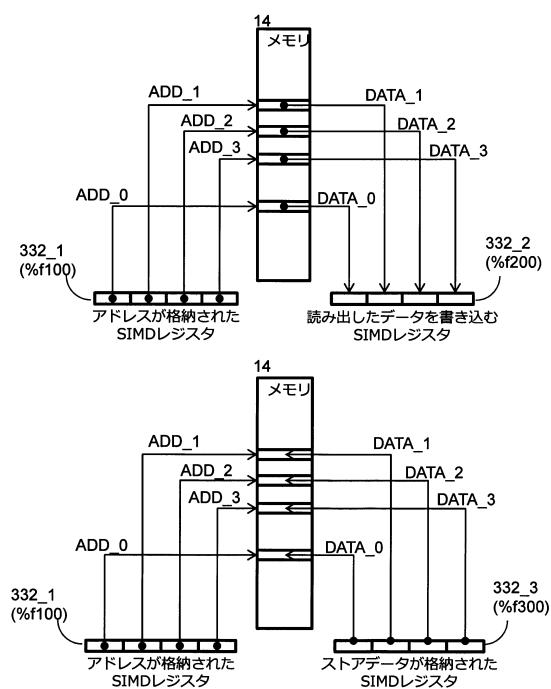
F L A , F L B : 浮動小数点 SIMD 演算パイプライン , SIMD 演算パイプライン

C S E : コミットスタックエントリ

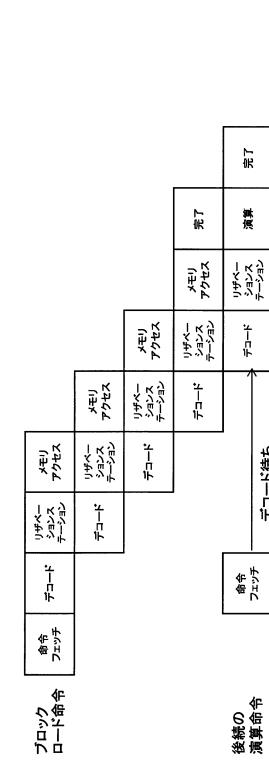
R S B R : 分岐用リザベーションステーション

P C : プログラムカウンタ

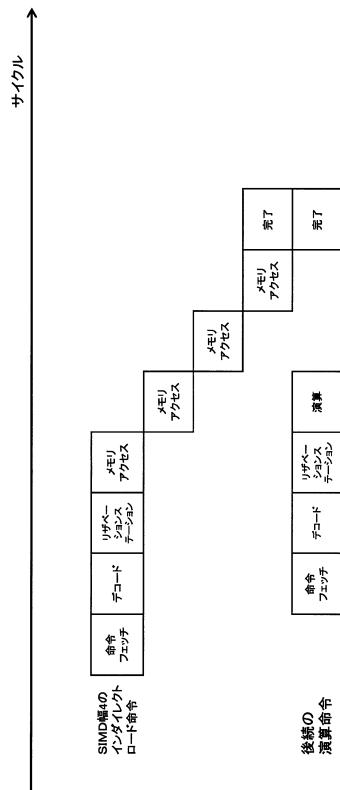
【図 1】



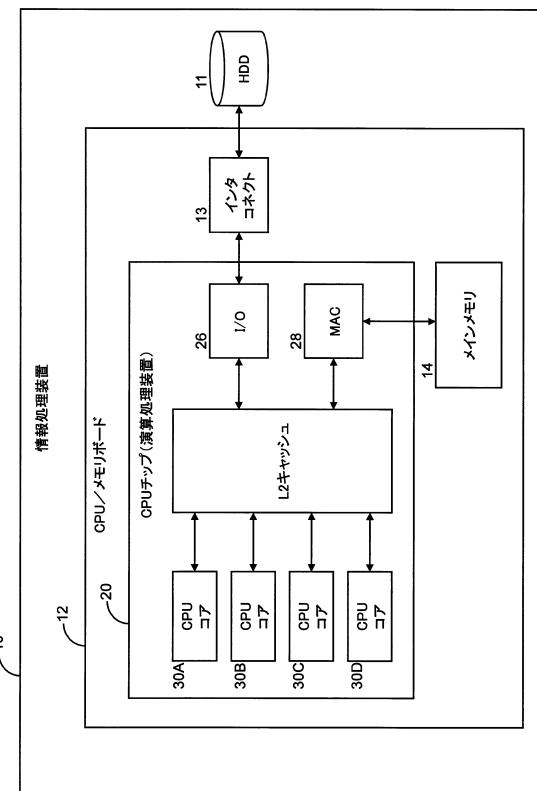
【図 2】



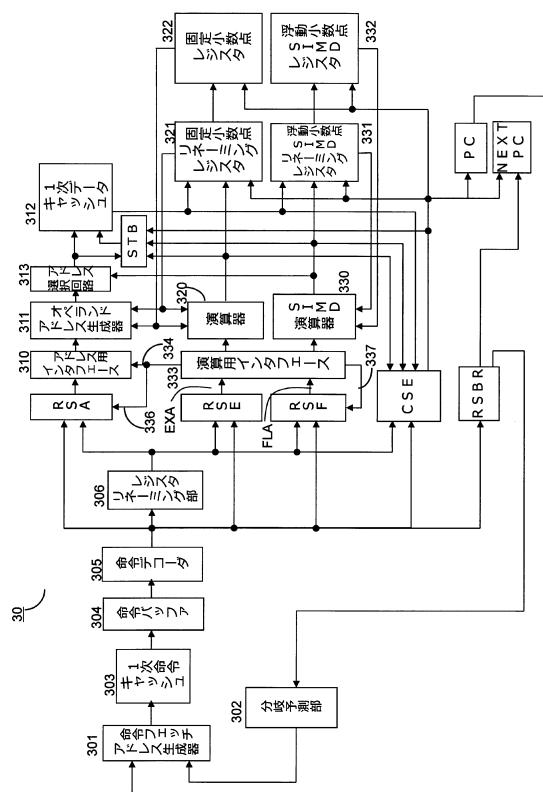
【図3】



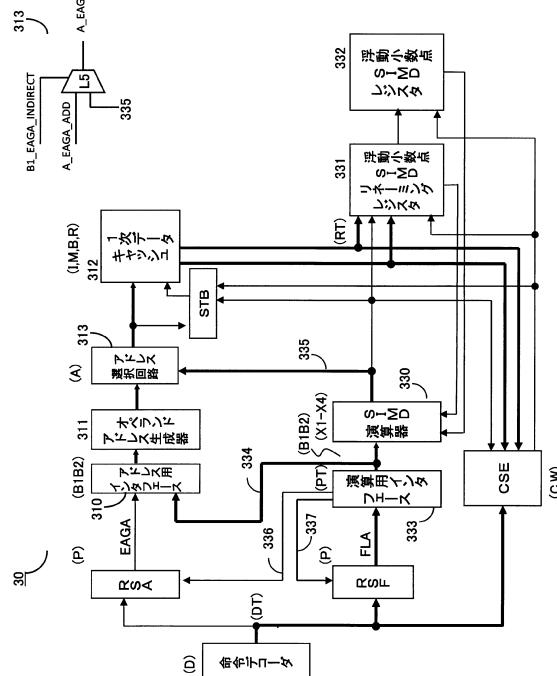
【図4】



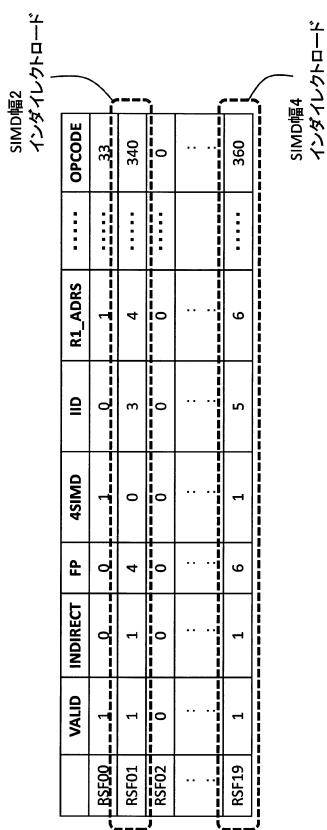
【 四 5 】



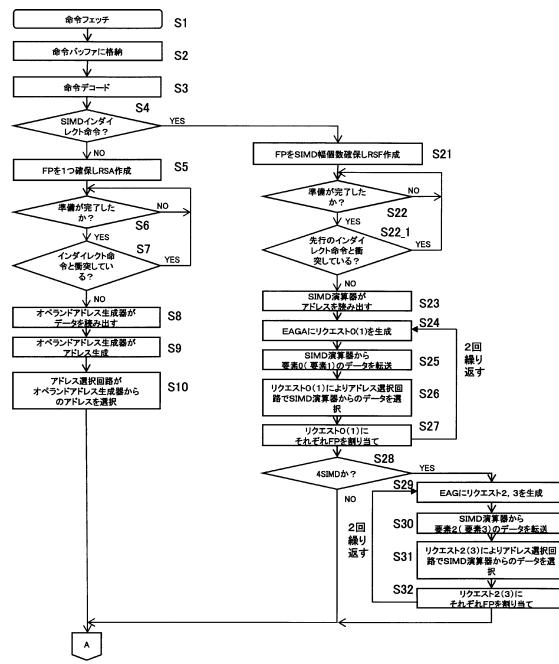
【図6】



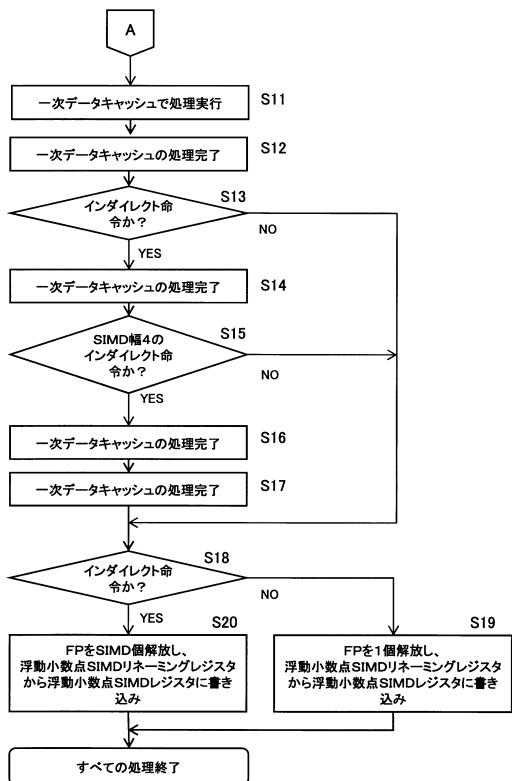
【図7】



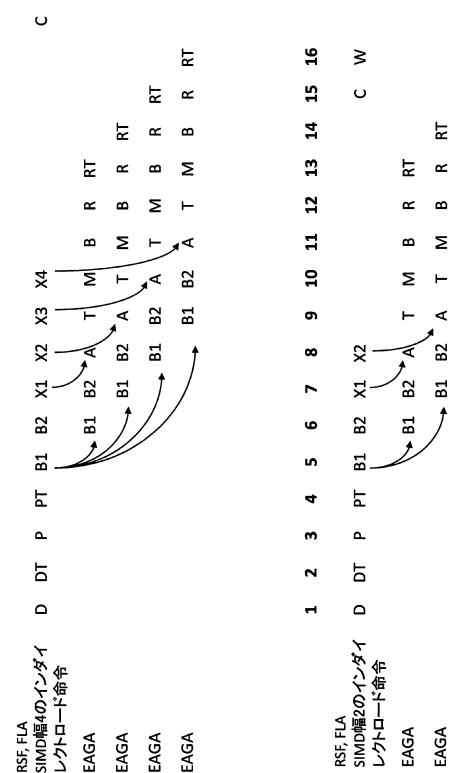
【図8】



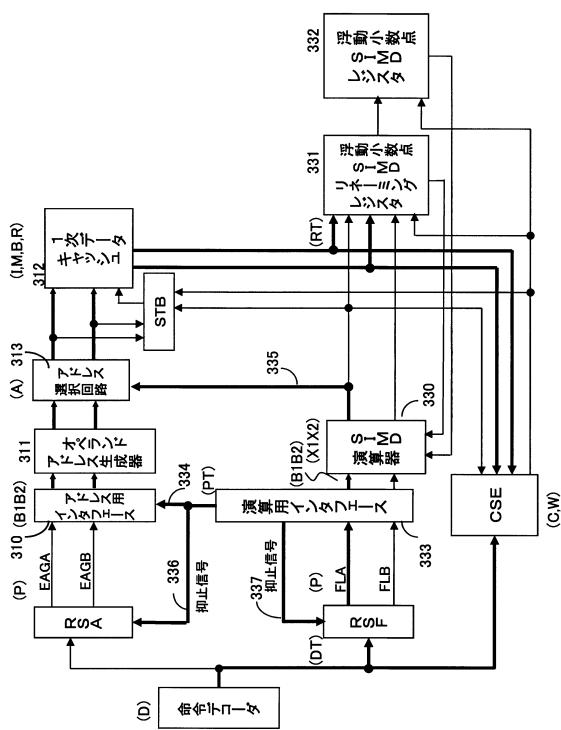
【図9】



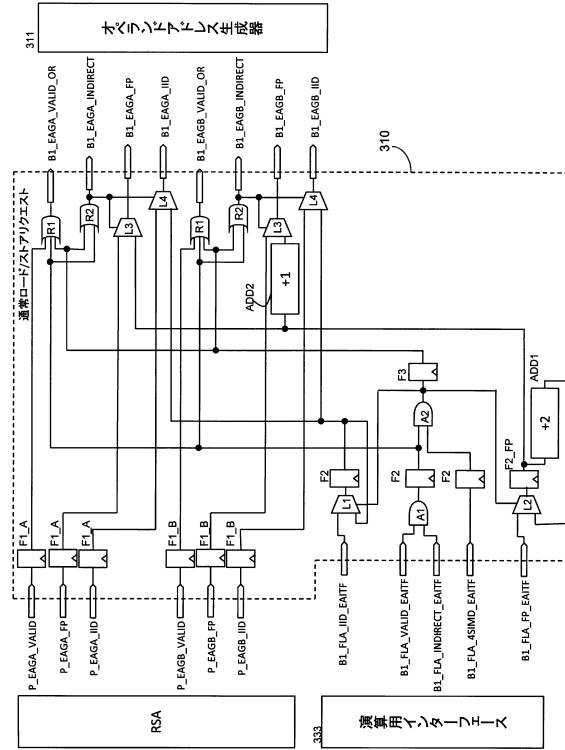
【図10】



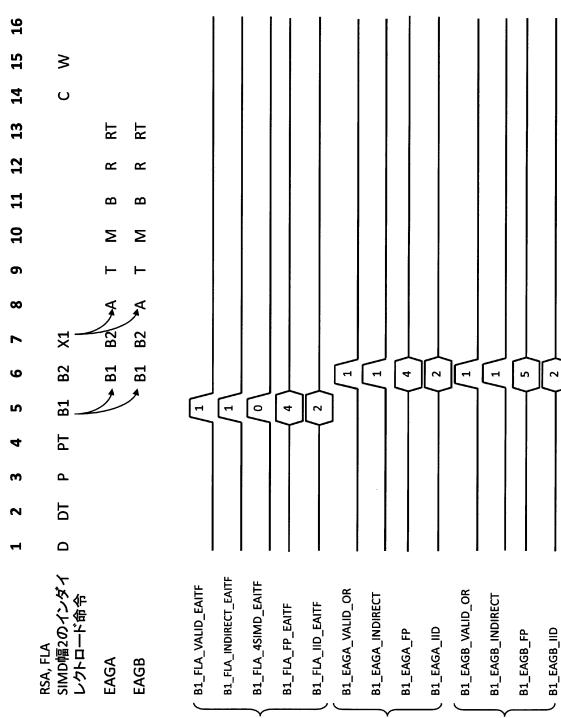
【図11】



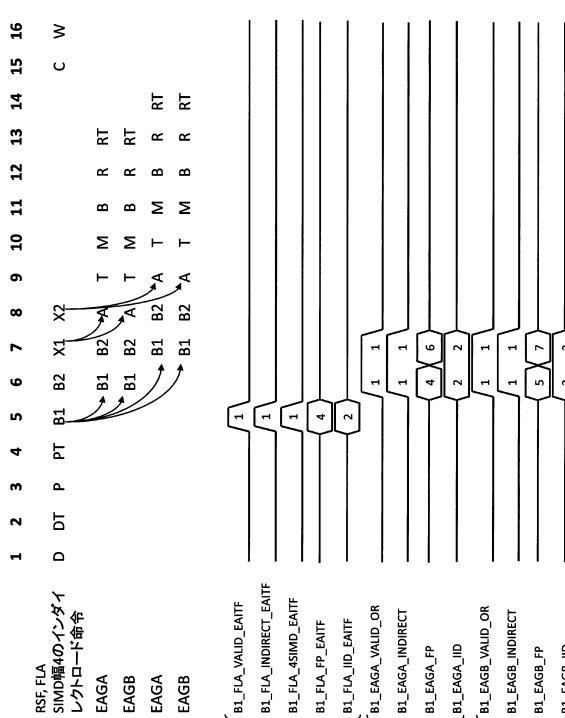
【図12】



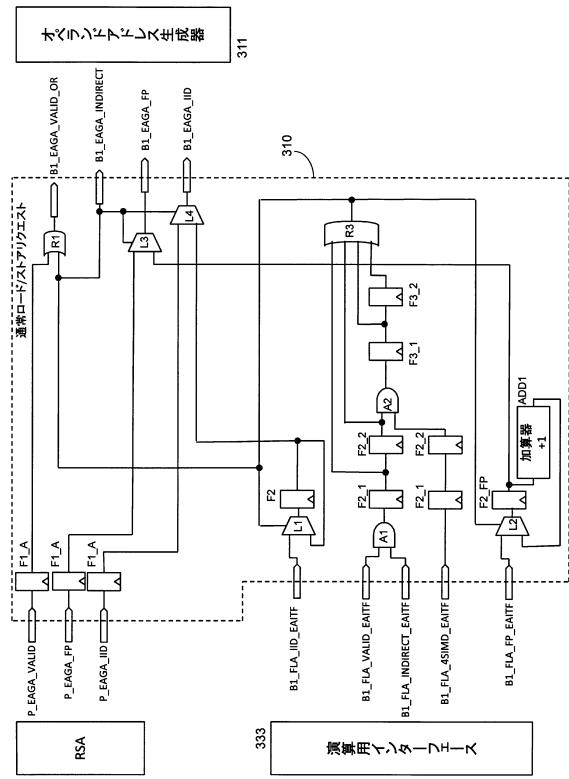
【 义 1 3 】



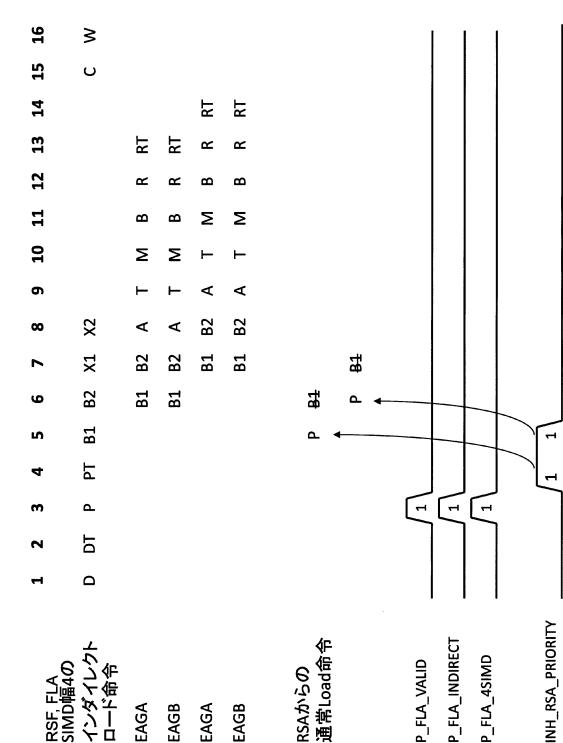
【図 1-4】



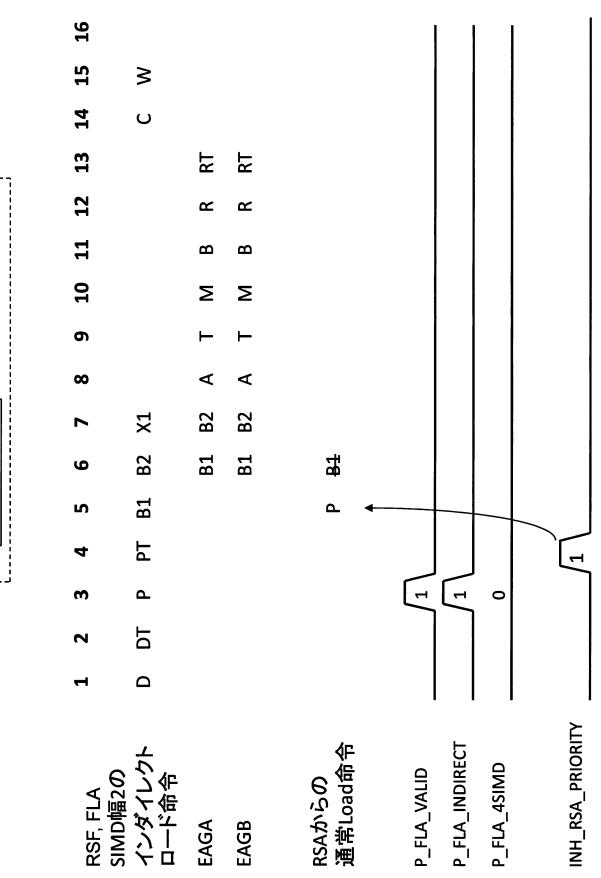
【図 15】



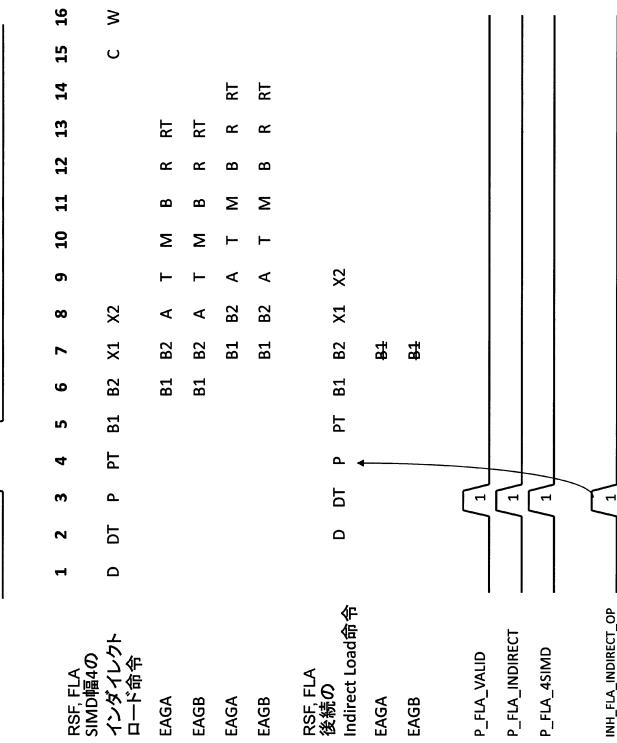
【図 17】



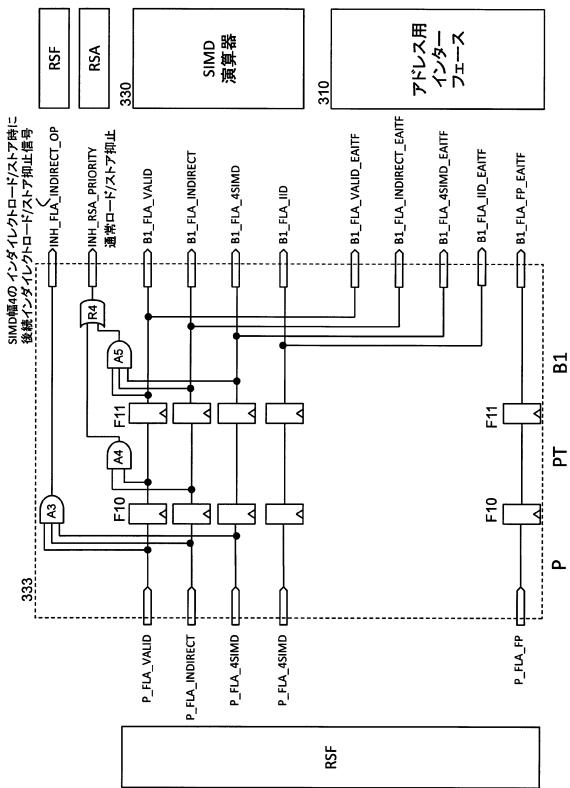
【図 16】



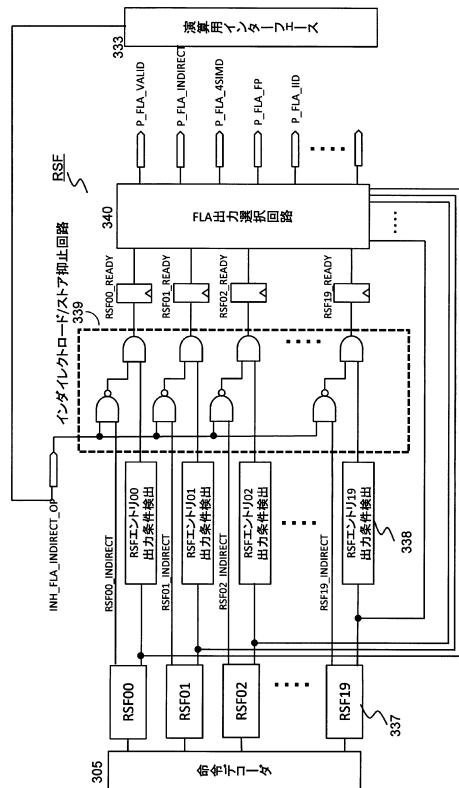
【図 18】



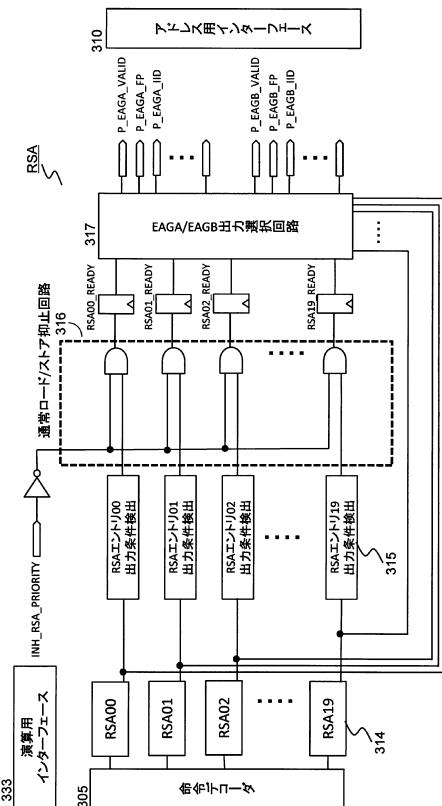
【図19】



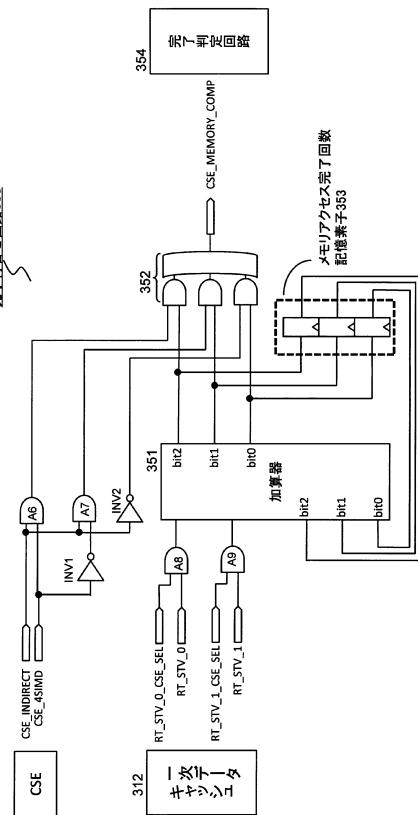
【図20】



【図21】



【図22】



フロントページの続き

(72)発明者 秋月 康伸
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 井上 宏一

(56)参考文献 米国特許出願公開第2012/0060016(US, A1)
特開2012-150634(JP, A)
吉田 利雄, 外3名, 「スーパーコンピュータ「京」のCPU SPARC64 VIIIfx」, FUJITSU, 富士通株式会社, 2012年 5月10日, 第63巻, 第3号, pp.254-259

(58)調査した分野(Int.Cl., DB名)
G 06 F 9 / 30 - 9 / 42