

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4916151号
(P4916151)

(45) 発行日 平成24年4月11日(2012.4.11)

(24) 登録日 平成24年2月3日(2012.2.3)

(51) Int. Cl. F I
G O 6 F 15/80 (2006.01) G O 6 F 15/80
G O 6 F 12/06 (2006.01) G O 6 F 12/06 5 2 5 A

請求項の数 2 (全 13 頁)

(21) 出願番号	特願2005-284144 (P2005-284144)	(73) 特許権者	302062931
(22) 出願日	平成17年9月29日 (2005.9.29)		ルネサスエレクトロニクス株式会社
(65) 公開番号	特開2007-94812 (P2007-94812A)		神奈川県川崎市中原区下沼部1753番地
(43) 公開日	平成19年4月12日 (2007.4.12)	(74) 代理人	100064746
審査請求日	平成20年6月3日 (2008.6.3)		弁理士 深見 久郎
		(74) 代理人	100085132
			弁理士 森田 俊雄
		(74) 代理人	100083703
			弁理士 仲村 義平
		(74) 代理人	100096781
			弁理士 堀井 豊
		(74) 代理人	100109162
			弁理士 酒井 将行
		(74) 代理人	100111246
			弁理士 荒川 伸夫

最終頁に続く

(54) 【発明の名称】 並列演算装置

(57) 【特許請求の範囲】

【請求項1】

複数のエントリに対応するデータを並列に演算する並列演算装置であって、
 前記複数のエントリに対応するデータの演算を行なう主演算手段と、
 マイクロ命令が格納されるマイクロ命令格納手段と、
 前記マイクロ命令格納手段に格納されたマイクロ命令を解釈し、前記主演算手段に演算を行なわせる制御手段とを含み、

前記主演算手段は、前記複数のエントリに対応するデータを格納するメモリマットと、
 前記複数のエントリに対応して設けられ、前記メモリマットに格納されるデータに対して演算を行なう複数の演算ユニットとを含み、

前記複数の演算ユニットのそれぞれは、前記制御手段によって設定される少なくとも1ビットのレジスタを含み、

前記制御手段は、前記複数の演算ユニットに含まれる前記レジスタに同じ値を設定し、
 該レジスタに設定された値を前記メモリマットに書込ませ、

前記並列演算装置はさらに、定数値を格納する定数レジスタを含み、

前記制御手段は、前記定数レジスタに設定される定数値を所定のビット単位で前記複数の演算ユニットに含まれる前記レジスタに順次設定し、

前記制御手段は、前記定数レジスタに設定される定数値を読み込んで解析し、解析結果に応じて前記複数の演算ユニットに含まれる前記レジスタに前記所定ビット単位で値を順次設定し、

前記並列演算装置はさらに、前記定数レジスタのビット位置を指定する第1のレジスタと、

前記メモリマットの複数のエントリのビット位置を指定する第2のレジスタと、
前記定数レジスタから前記メモリマットの複数のエントリに転送する定数のビット数を
指定する第3のレジスタとを含み、

前記制御手段は、前記マイクロ命令格納手段から所定命令を読み込んだときに、前記第1
のレジスタによって指定された前記定数レジスタのビット位置から、前記第3のレジスタ
によって指定されたビット数分の定数を、前記第2のレジスタによって指定された前記メ
モリマットのビット位置を先頭に前記所定ビット単位で転送する、並列演算装置。

【請求項2】

複数のエントリに対応するデータを並列に演算する並列演算装置であって、
前記複数のエントリに対応するデータの演算を行なう主演算手段と、
マイクロ命令が格納されるマイクロ命令格納手段と、
前記マイクロ命令格納手段に格納されたマイクロ命令を解釈し、前記主演算手段に演算
を行なわせる制御手段とを含み、

前記主演算手段は、前記複数のエントリに対応するデータを格納するメモリマットと、
前記複数のエントリに対応して設けられ、前記メモリマットに格納されるデータに対し
て演算を行なう複数の演算ユニットとを含み、

前記複数の演算ユニットのそれぞれは、前記制御手段によって設定される少なくとも1
ビットのレジスタを含み、

前記制御手段は、前記複数の演算ユニットに含まれる前記レジスタに同じ値を設定し、
該レジスタに設定された値と前記メモリマットからロードした値との演算を行なわせ、

前記並列演算装置はさらに、定数値を格納する定数レジスタを含み、
前記制御手段は、前記定数レジスタに設定される定数値を所定のビット単位で前記複数
の演算ユニットに含まれる前記レジスタに順次設定し、

前記制御手段は、前記定数レジスタに設定される定数値を読み込んで解析し、解析結果に
応じて前記複数の演算ユニットに含まれる前記レジスタに前記所定ビット単位で値を順次
設定し、

前記並列演算装置はさらに、前記定数レジスタのビット位置を指定する第1のレジスタ
と、

前記メモリマットの複数のエントリのビット位置を指定する第2のレジスタと、
前記定数レジスタに格納される定数値のうち演算対象の定数のビット数を指定する第3
のレジスタとを含み、

前記制御手段は、前記マイクロ命令格納手段から所定命令を読み込んだときに、前記第1
のレジスタによって指定された前記定数レジスタのビット位置から前記第3のレジスタに
よって指定されたビット数分の定数と、前記第2のレジスタによって指定された前記メモ
リマットのビット位置から前記第3のレジスタによって指定されたビット数分のデータと
を前記所定ビット単位で演算する、並列演算装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、SIMD (Single Instruction stream-Multiple Data stream) 型、MIMD (Multiple Instruction stream-Multiple Data stream) 型などの並列演算装置に関し、特に、高速に定数を設定することが可能な並列演算装置に関する。

【背景技術】

【0002】

近年、携帯端末機器などの普及に伴い、音声、画像などの大量のデータを高速に処理するデジタル信号処理の重要性が高くなってきている。このようなデジタル信号処理においては、SIMD型、MIMD型などの並列演算装置を搭載した処理システムが用いられることが多くなっている。

10

20

30

40

50

【 0 0 0 3 】

このような処理システムに搭載される並列演算装置は、一般に、メモリマットと演算処理ユニット（ALU）とを含む。並列演算装置を制御するホストCPU（Central Processing Unit）と並列演算装置とを接続するシステムバスのバス幅が32ビット、並列演算装置のメモリマットに設定する定数のビット幅が32ビット、定数のエントリ数が1024個の場合、定数をエントリの全てに設定するために必要となる転送サイクル数はベストケースで1024サイクルとなる。

【 0 0 0 4 】

また、並列演算に関する技術として、特開平5 - 197550号公報に開示された発明がある。この発明は、ALUの構成に簡単な機能を追加して処理速度を向上することのできる超並列計算機のALU構成方式に関するものであり、各種の演算を実行するための演算器とデータを記憶するメモリとをそれぞれに有する複数のプロセッサを備え、この各プロセッサを1つの制御プロセッサで並列に動作させる。各演算器において、アドレス格納手段は、メモリをアクセスする際のアドレスを格納するものであって、アドレス制御手段は、アドレス格納手段に格納されるべきアドレスを生成するとともに、このアドレス値に従ってメモリをアクセスすることで、メモリの中の異なるメモリ領域をアクセスするよう構成するものである。

【特許文献1】特開平5 - 197550号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 5 】

上述したように、メモリマットとALUとを含んだ並列演算装置において、システムバス幅が32ビット、定数のビット幅が32ビット、定数のエントリ数が1024個の場合、全エントリに共通の定数値（係数）を設定するときでも同じデータを1024回設定する必要がある。したがって、並列演算装置の処理性能が高くても定数を設定するのに必要な転送サイクルが多くなってしまい、システム全体のボトルネックになり得るといった問題点があった。

【 0 0 0 6 】

また、特許文献1に開示された発明を用いたとしても、このような問題点を解決することはできない。

【 0 0 0 7 】

本発明は、上記問題点を解決するためになされたものであり、その目的は、メモリマットの全エントリに共通の定数値を設定する場合に、高速に定数値の設定が行なえる並列演算装置を提供することである。

【 0 0 0 8 】

他の目的は、メモリマットの全エントリのデータに対して共通の定数値を演算する場合に、高速に演算が行なえる並列演算装置を提供することである。

【課題を解決するための手段】

【 0 0 0 9 】

本発明のある局面に従えば、複数のエントリに対応するデータを並列に演算する並列演算装置であって、複数のエントリに対応するデータの演算を行なう主演算手段と、マイクロ命令が格納されるマイクロ命令格納手段と、マイクロ命令格納手段に格納されたマイクロ命令を解釈し、主演算手段に演算を行なわせる制御手段とを含み、主演算手段は、複数のエントリに対応するデータを格納するメモリマットと、複数のエントリに対応して設けられ、メモリマットに格納されるデータに対して演算を行なう複数の演算ユニットとを含み、複数の演算ユニットのそれぞれは、制御手段によって設定される少なくとも1ビットのレジスタを含み、制御手段は、複数の演算ユニットに含まれるレジスタに同じ値を設定し、レジスタに設定された値をメモリマットに書込ませる。

【 0 0 1 0 】

本発明の別の局面に従えば、複数のエントリに対応するデータを並列に演算する並列演

10

20

30

40

50

算装置であって、複数のエントリに対応するデータの演算を行なう主演算手段と、マイクロ命令が格納されるマイクロ命令格納手段と、マイクロ命令格納手段に格納されたマイクロ命令を解釈し、主演算手段に演算を行なわせる制御手段とを含み、主演算手段は、複数のエントリに対応するデータを格納するメモリマットと、複数のエントリに対応して設けられ、メモリマットに格納されるデータに対して演算を行なう複数の演算ユニットとを含み、複数の演算ユニットのそれぞれは、制御手段によって設定される少なくとも1ビットのレジスタを含み、制御手段は、複数の演算ユニットに含まれるレジスタに同じ値を設定し、レジスタに設定された値とメモリマットからロードした値との演算を行なわせる。

【発明の効果】

【0011】

10

本発明のある局面によれば、制御手段が、複数の演算ユニットに含まれるレジスタに同じ値を設定し、レジスタに設定された値をメモリマットに書込ませるので、メモリマットの全エントリに共通の定数値を設定する場合に、高速に定数値の設定を行なうことが可能となる。

【0012】

本発明の別の局面によれば、制御手段が、複数の演算ユニットに含まれるレジスタに同じ値を設定し、レジスタに設定された値とメモリマットからロードした値との演算を行なわせるので、メモリマットの全エントリのデータに対して共通の定数値を演算する場合に、高速に演算を行なうことが可能となる。

【発明を実施するための最良の形態】

20

【0013】

(第1の実施の形態)

図1は、本発明の第1の実施の形態における並列演算装置が搭載される処理システムの構成例を示すブロック図である。この処理システムは、システムLSI (Large Scale Integrated circuit) 1と、外部システムバス5を介してシステムLSI 1に接続されるSDRAM (Synchronous Dynamic Random Access Memory) 2と、SRAM (Static Random Access Memory) 3と、ROM (Read Only Memory) 4とを含む。

【0014】

ROM 4には、リセット直後に実行されるIPL (Initial Program loader) などが格納される。システムLSI 1は、外部システムバス5を介してSDRAM 2、SRAM 3およびROM 4にアクセスしながら、演算処理を実行する。

30

【0015】

システムLSI 1は、複数の基本演算ブロック11-1~11-kと、出力IP (Intellectual Property) 12と、入力IP 13と、システムLSI 1の全体的な制御を行なうホストCPU 14と、割込みコントローラ15と、CPU周辺回路16と、DMA (Direct Memory Access) コントローラ17と、外部システムバス5を介して外部メモリ等との間のデータの入出力を制御する外部バスコントローラ18と、専用ロジックIP 19とを含む。これらのブロックは、ホストCPUシステムバス20に接続される。なお、基本演算ブロック11-1~11-kのそれぞれが並列演算装置に相当する。

【0016】

40

出力IP 12は、ホストCPUシステムバス20を介して受けた出力データを出力信号に変換して外部に出力する。入力IP 13は、外部からの入力信号を内部処理用データに変換してホストCPUシステムバス20に出力する。

【0017】

割込みコントローラ15は、基本演算ブロック11-1~11-kからの割込信号を受け、ホストCPU 14に対して割込要求を出力する。

【0018】

ホストCPU 14は、外部のROM 4などに格納されたプログラムを実行して、ホストCPUシステムバス20を介して基本演算ブロック11-1~11-kにデータを設定したり、基本演算ブロック11-1~11-kによる演算結果を受けて所定の処理を行なっ

50

たりする。また、ホストCPU 14は、割込みコントローラ 15から割込要求を受けると、割込要求に対応した割込処理を行なう。

【0019】

CPU周辺回路 16は、ホストCPU 14がプログラムやデバッグを実行する際に使用するタイマ、シリアルI/Oなどの機能を含む。

【0020】

DMAコントローラ 17は、基本演算ブロック 11-1~11-kからのDMA要求信号を受け、基本演算ブロック 11-1~11-kと外部システムバス5に接続される外部メモリとの間のDMA転送を制御する。

【0021】

専用ロジックIP 19は、ホストCPU 14のデータ処理を補助するための論理回路によって構成され、各機能ブロックを用いて必要な処理機能を実現する。

【0022】

基本演算ブロック 11-1~11-kのそれぞれは、実際のデータの演算処理を行なう主演算回路 21と、コントローラ 22と、マイクロ命令メモリ 23と、ワークデータメモリ・レジスタ群 24と、ホストCPUシステムバス 20を介して他のブロックとの間のデータの入出力を行なうシステムバスI/F (Interface) 25と、直交変換回路 26と、スイッチ 27とを含む。

【0023】

マイクロ命令メモリ 23は、主演算回路 21による演算処理を指定するマイクロ命令を格納する。コントローラ 22は、マイクロ命令メモリ 23から読込んだマイクロ命令に従って主演算回路 21の演算処理を制御する。

【0024】

ワークデータメモリ・レジスタ群 24は、コントローラ 22による中間処理データなどを格納するワークエリアとして用いられるメモリと、後述するレジスタ群とを含む。スイッチ 27は、システムバスI/F 25および直交変換回路 26からのデータを選択的に主演算回路 21に出力し、主演算回路 21からのデータをシステムバスI/F 25または直交変換回路 26に選択的に出力する。

【0025】

主演算回路 21は、複数のメモリセルが行列状に配列されかつ複数のエントリに分割されるメモリマット A, B (32)と、メモリマット A, B (32)の各エントリに対応して配置され、指定された演算処理を行なうALU (Arithmetic and Logic Unit) 31-1~31-mと、ALU 31-1~31-m間のデータ転送経路を設定するALU間相互接続用スイッチ回路 33とを含む。

【0026】

基本的には、メモリマット A, B (32)の各列が1エントリを構成し、1エントリに多ビットデータの各ビットが格納される。ALU 31-1~31-mは、対応のエントリからのデータをビットシリアルに受けて演算処理を行ない、処理結果をメモリマット A, B (32)の指定されたエントリに格納する。

【0027】

また、ALU間相互接続用スイッチ回路 33は、ALU 31-1~31-mの接続経路を切換え、異なるエントリのデータの演算をALU 31-1~31-mに行なわせることも可能である。各エントリに異なるデータを格納し、ALU 31-1~31-mに並列演算処理を行なわせることにより、高速にデータ処理を行なうことも可能である。

【0028】

基本演算ブロック 11-1~11-kには、異なるCPUアドレス領域が割付けられる。また、基本演算ブロック 11-1~11-k内のメモリマット A, B (32)、コントローラ 22内の制御レジスタ、マイクロ命令メモリ 23およびワークデータメモリ・レジスタ群 24にも、それぞれ異なるCPUアドレスが割付けられる。

【0029】

10

20

30

40

50

したがって、基本演算ブロック 11 - 1 ~ 11 - k のそれぞれのマイクロ命令メモリ 23 に異なる内容のマイクロ命令を格納することにより、異なる演算処理を並行して行なわせることができる。また、基本演算ブロック 11 - 1 ~ 11 - k において、異なるアドレス領域のデータについて同一の演算処理が行なわれるように、マイクロ命令メモリ 23 に同一の演算内容のマイクロ命令が格納されてもよい。

【0030】

ホストCPU 14 およびDMAコントローラ 17 は、割付けられたアドレスに応じてアクセス対象の基本演算ブロック 11 - 1 ~ 11 - k を識別し、アクセス対象の基本演算ブロックに対するアクセスを実行する。

【0031】

図2は、主演算回路21の機能を模式的に示す図である。メモリマットA, B (32) は、メモリセルが行列状に配列され、m個のエントリに分割される。エントリのそれぞれは、メモリマットAおよびメモリマットBのそれぞれにおいて、nビットのビット幅を有する。

【0032】

ALU31 - 1 ~ 31 - m は、m個のエントリのそれぞれに対して設けられ、加算、論理積、一致検出 (EXOR)、反転 (NOT) などの演算を行なう。ALU31 - 1 ~ 31 - m は、メモリマットAおよびメモリマットBに同時にアクセス可能である。

【0033】

ALU31 - 1 ~ 31 - m は、対応するエントリのメモリマットAおよびメモリマットBからのデータのロードと、メモリマットAおよびメモリマットBへのデータのストアとを行ないながら、ビットシリアルで演算処理を実行する。図2においては、メモリマットBに定数が設定されているところを示している。

【0034】

図3は、本発明の第1の実施の形態における並列演算装置の主演算装置21の機能的構成をさらに詳細に説明するための図である。各エントリ間のデータ転送は、ALU間相互接続用スイッチ回路33によって、ALU31 - 1 ~ 31 - m を介して行なわれる。なお、エントリ間のデータ転送は、ALUを介して行なわれなくともよい。

【0035】

ALU31 - 1 ~ 31 - m のそれぞれは、1ビットのXレジスタ34およびXHレジスタ35を有し、ALUによる演算結果を一時的に記憶する。各エントリのXレジスタおよびXHレジスタは、コントローラ22からの制御信号が接続されており、コントローラ22からの信号によって、1にセット、0にクリアすることが可能である。なお、このセット/クリアは、全エントリに対して同時に行なうことも可能である。

【0036】

コントローラ22は、マイクロ命令メモリ23から読込んだマイクロ命令を解釈し、メモリマットA, B (32) への定数の設定命令であれば、定数レジスタ28の内容を取得してXレジスタ34およびXHレジスタ35に設定し、指定されたメモリマットA, B (32) のビット位置にXレジスタ34およびXHレジスタ35の内容を順次書込む。以下、定数の設定手順について説明する。

【0037】

まず、ホストCPU 14 は、外部システムバス5および外部バスコントローラ18を介して外部のメモリから定数値を読み込み、ホストCPUシステムバス20およびシステムバスI/F25を介して定数レジスタ28に定数値を設定する。そして、コントローラ22は定数の設定命令を読み込む。

【0038】

(1) コントローラ22は、定数レジスタ28に設定されている定数値の最下位の2ビットを読み込み、全エントリのXレジスタ34およびXHレジスタ35に設定する。ここで、定数値の設定方法として、コントローラ22が定数レジスタ28に設定されている2ビットの値を解釈し、全エントリのXレジスタ34およびXHレジスタ35のセット/クリ

10

20

30

40

50

ア制御を行なうようにしてもよいし、定数レジスタ28に設定されている2ビットの値を直接全エントリのXレジスタ34およびXHレジスタ35に設定するようにしてもよい。

【0039】

なお、Xレジスタ34およびXHレジスタ35を用いて2ビット単位で定数を設定する方法について説明しているが、実装条件によっては1ビット単位で定数値を設定してもよいし、3ビット以上の単位で定数値を設定するようにしてもよい。

【0040】

(2)次に、Xレジスタ34およびXHレジスタ35に設定された値が、コントローラ22によって指定されたメモリマットB(32)のビット位置に書込まれる。

【0041】

(3)そして、コントローラ22は、定数レジスタ制御信号を出力して、定数レジスタ28から読み込む定数値のビット位置を2ビット上位側にシフトし、その2ビットを全エントリのXレジスタ34およびXHレジスタ35に設定する。このシフト量も定数を何ビット単位で設定するかによって違ってくる。なお、定数値が書込まれるメモリマットB(32)のビット位置も順次シフトされる。

【0042】

以上の(1)~(3)の処理(図3の(1)~(3)に対応)を繰り返し、定数レジスタ28に設定された定数値が全てメモリマットB(32)に書込まれたときに定数値の設定処理が終了する。なお、上記(1)~(3)の1回の処理は、1サイクルで実行されるものとする。

【0043】

以上説明したように、本実施の形態における並列演算装置によれば、コントローラ22が定数レジスタ28に設定された定数値を所定ビット単位でALU31-1~31-m内のレジスタに設定し、その値を各エントリの所定ビット位置に同時に書込むようにしたので、定数値の設定処理を高速で行なうことが可能となった。

【0044】

たとえば、上述のようにXレジスタ34およびXHレジスタ35を用いてメモリマットB(32)に32ビットの定数値を設定する場合、定数レジスタ28への定数値の設定サイクル数“1”を含めたとしても、エントリ数に関係なく17サイクルで行なうことができる。一方、従来のようにホストCPUが各エントリ毎に定数値を外部メモリから読み込み、メモリマットの各エントリに定数値を設定する場合には、エントリ数が1024であれば、1024サイクルを要することになる。したがって、本実施の形態においては、従来よりも60倍高速に定数値を設定できることになる。エントリ数が増えれば増えるほど、その差が大きくなる。

【0045】

(第2の実施の形態)

図4は、本発明の第2の実施の形態における並列演算装置の構成例を示す図である。この並列演算装置は、ALU31-1~31-mと、メモリマットA, B(32)と、定数レジスタK0~K7と、プログラムカウンタPC0~PC3と、ループカウンタLC0~LC3と、アレイポインタP0~P3と、汎用レジスタR0~R7と、ステータス・レジスタSTと、エントリコミュニケータ移動量指定レジスタECRと、haltレジスタHLTとを含む。なお、このレジスタ群は、図1に示すワークデータメモリ・レジスタ群24に設けられる。

【0046】

定数レジスタは、K0~K7で128ビットのレジスタを構成する。アレイポインタP0(MP)およびP2(MP2)は、定数レジスタK0~K7に格納される128ビットの定数のビット位置を示し、転送元アドレスとして参照される。アレイポインタP1(AP)およびP3(AP2)は、メモリマットA, B(32)の各エントリのビット位置を示し、転送先アドレスとして参照される。ループカウンタLC0~LC3は、メモリアレイA, B(32)の各エントリに書込む定数のビット数を示す。

10

20

30

40

50

【0047】

汎用レジスタR0～R7は、演算対象のデータや演算結果などを格納するのに使用される。ステータス・レジスタSTは、基本演算ブロックの状態を示すフラグなどが格納されるレジスタである。また、haltレジスタHLTは、基本演算ブロックの処理が停止中であるか否かを示す。なお、レジスタの個数は、これらに限られるものではない。

【0048】

図5(a)は、2ビット単位で定数をメモリマットに設定する命令の一例を示す図である。m×2__c p__k(a p, k p, n)命令は、Kレジスタのビット位置k pからnビット分の定数を、メモリマットA, B(32)の全エントリのビット位置a pからnビットのメモリ領域へ2ビット単位で順次コピーする命令である。たとえば、転送先アドレスa pはアレイポインタP1(AP)に設定され、転送元アドレスk pはアレイポインタP0(MP)に設定され、ビット長nはループカウンタLC0に設定される。

10

【0049】

図5(b)は、m×2__c p__k命令を用いたプログラムの一例を示す図である。まず、set reg命令を実行することによって、定数レジスタK0に“12345”が設定される。そして、m×__s t V命令を実行することによって、定数レジスタK0に設定した定数を各エントリのメモリマットに設定するか否かを制御する。このVフラグは、各エントリのALUにレジスタとして割付けられている。

【0050】

最後に、m×2__c p__k命令を実行することによって、メモリマットA, B(32)の全エントリのビット位置-16～-1の領域に、定数レジスタK0のビット位置0から2ビット単位で16ビット分の定数が順次コピーされる。

20

【0051】

図6は、1ビット単位で定数をメモリマットに設定する命令の一例を示す図である。m×__c p__k(a p, k p, n)命令は、Kレジスタのビット位置k pからnビット分の定数を、メモリマットA, B(32)の全エントリのビット位置a pからnビットのメモリ領域へ1ビット単位で順次コピーする命令である。この場合には、図3に示すXレジスタ34またはXHレジスタ35のいずれか一方のみが使用される。

【0052】

以上説明したように、本実施の形態における並列演算装置によれば、m×2__c p__k命令やm×__c p__k命令をサポートすることにより、これらの命令を1回発行するだけで各エントリへの定数の設定を行なうことが可能となった。

30

【0053】

また、設定したい定数の複数パターンをKレジスタに予め設定しておき、必要な定数のビット位置とビット数とを指定することにより、複数の定数を容易にメモリマットの各エントリに設定することが可能となる。

【0054】

(第3の実施の形態)

本発明の第2の実施の形態においては、2ビット単位または1ビット単位で定数をメモリマットの各エントリに設定する命令について説明した。本実施の形態においては、定数レジスタKに設定された定数をメモリマットに設定せずに、定数レジスタKに設定された定数を参照して直接演算を行なうものである。

40

【0055】

本発明の第3の実施の形態における並列演算装置の構成例は、図4に示す第2の実施の形態における並列演算装置の構成例と同様である。したがって、重複する構成および機能の詳細な説明は繰返さない。

【0056】

図7(a)は、定数レジスタの定数を2ビット単位で参照して加算を行なう命令の一例を示す図である。m×2__a d d__k(a p, k p, n)命令は、メモリマットA, B(32)のビット位置a pからnビットの領域の内容と、Kレジスタのビット位置k pから

50

nビットの領域の内容とを2ビット単位で加算し、その加算結果をメモリマツトA, B (32)のビット位置apからnビットの領域に格納する命令である。

【0057】

図7(b)は、Kレジスタのビット位置kpを説明するための図である。8個のKレジスタは、連続した128ビットの領域として見え、たとえば、ビット位置kpが定数レジスタK0の所定位置を指し示す場合、定数レジスタK0の内容が2ビット単位で順次参照され、定数レジスタK0のビット15まで参照されると、次は定数レジスタK1のビット0から順次参照される。

【0058】

並列演算装置が $m \times 2 _ \text{add} _ k$ 命令を実行する場合、ALU31-1~31-mのそれぞれは、図3に示すXレジスタ34およびXHレジスタ35に設定された値と、メモリマツトからロードした2ビットのデータとを加算して、加算結果をメモリマツトにストアする。

【0059】

図8は、定数レジスタの定数を1ビット単位で参照して加算を行なう命令の一例を示す図である。 $m \times _ \text{add} _ k (ap, kp, n)$ 命令は、メモリマツトA, B (32)のビット位置apからnビットの領域の内容と、Kレジスタのビット位置kpからnビットの領域の内容とを1ビット単位で加算し、その加算結果をメモリマツトA, B (32)のビット位置apからnビットの領域に格納する命令である。

【0060】

図9(a)は、定数レジスタの定数を2ビット単位で参照して減算を行なう命令の一例を示す図である。 $m \times 2 _ \text{sub} _ k (ap, kp, n)$ 命令は、メモリマツトA, B (32)のビット位置apからのnビットの領域の内容から、Kレジスタのビット位置kpからのnビットの領域の内容を2ビット単位で減算し、その減算結果をメモリマツトA, B (32)のビット位置apからnビットの領域に格納する命令である。

【0061】

並列演算装置が $m \times 2 _ \text{sub} _ k$ 命令を実行する場合、ALU31-1~31-mのそれぞれは、メモリマツトからロードした2ビットのデータから、図3に示すXレジスタ34およびXHレジスタ35に設定された値を減算して、減算結果をメモリマツトにストアする。

【0062】

図9(b)は、定数レジスタの定数を1ビット単位で参照して減算を行なう命令の一例を示す図である。 $m \times _ \text{sub} _ k (ap, kp, n)$ 命令は、メモリマツトA, B (32)のビット位置apからのnビットの領域の内容から、Kレジスタのビット位置kpからのnビットの領域の内容を1ビット単位で減算し、その減算結果をメモリマツトA, B (32)のビット位置apからnビットの領域に格納する命令である。

【0063】

以上説明したように、本実施の形態における並列演算装置によれば、定数レジスタKに設定された定数を参照して直接演算を行なうようにしたので、定数レジスタKに設定された定数をメモリマツトに設定する処理を省くことができ、処理性能を向上させることが可能となった。

【0064】

今回開示された実施の形態は、すべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【図面の簡単な説明】

【0065】

【図1】本発明の第1の実施の形態における並列演算装置が搭載される処理システムの構成例を示すブロック図である。

10

20

30

40

50

【図2】主演算回路21の機能を模式的に示す図である。

【図3】本発明の第1の実施の形態における並列演算装置の主演算装置21の機能的構成をさらに詳細に説明するための図である。

【図4】本発明の第2の実施の形態における並列演算装置の構成例を示す図である。

【図5】(a)は、2ビット単位で定数をメモリマットに設定する命令の一例を示す図である。(b)は、 $m \times 2_cp_k$ 命令を用いたプログラムの一例を示す図である。

【図6】1ビット単位で定数をメモリマットに設定する命令の一例を示す図である。

【図7】(a)は、定数レジスタの定数を2ビット単位で参照して加算を行なう命令の一例を示す図である。(b)は、Kレジスタのビット位置kpを説明するための図である。

【図8】定数レジスタの定数を1ビット単位で参照して加算を行なう命令の一例を示す図である。

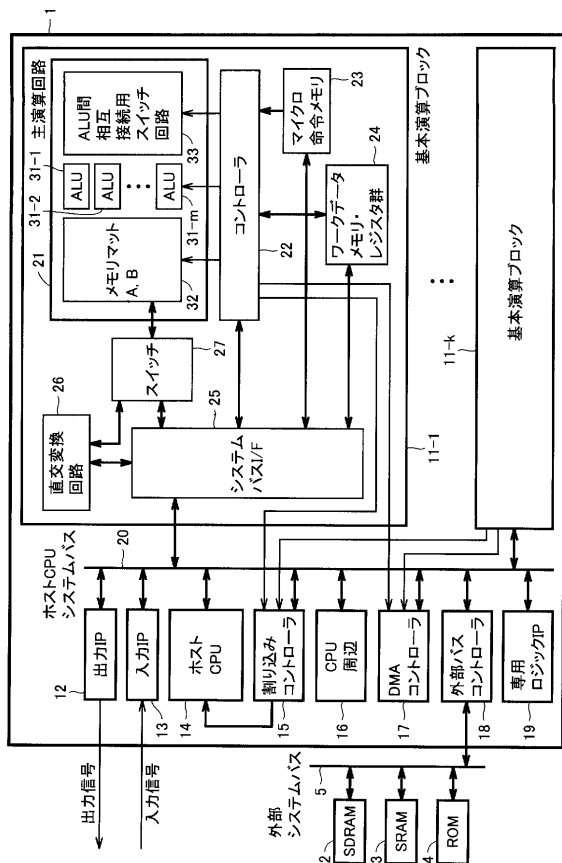
【図9】(a)は、定数レジスタの定数を2ビット単位で参照して減算を行なう命令の一例を示す図である。(b)は、定数レジスタの定数を1ビット単位で参照して減算を行なう命令の一例を示す図である。

【符号の説明】

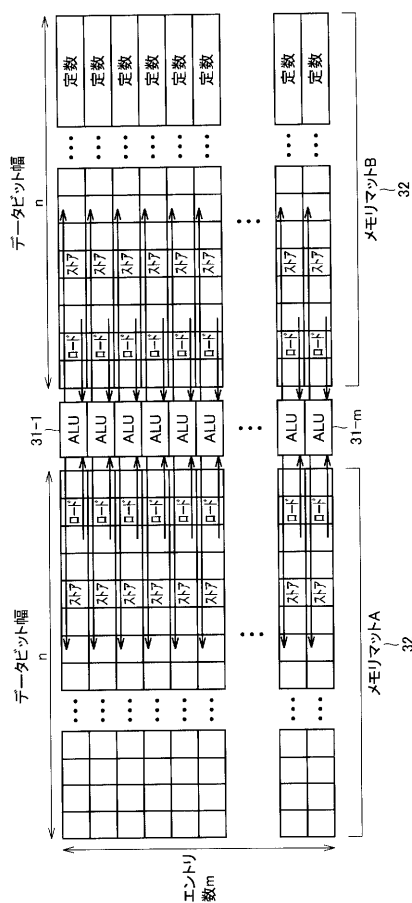
【0066】

1 システムLSI、2 SDRAM、3 SRAM、4 ROM、5 外部システムバス、11-1~11-k 基本演算ブロック、12 出力IP、13 入力IP、14 ホストCPU、15 割り込みコントローラ、16 CPU周辺回路、17 DMAコントローラ、18 外部バスコントローラ、19 専用ロジックIP、20 ホストCPUシステムバス、21 主演算回路、22 コントローラ、23 マイクロ命令メモリ、24 ワークデータメモリ・レジスタ群、25 システムバスI/F、26 直交変換回路、27 スイッチ、28 定数レジスタ、31-1~31-m ALU、32 メモリマットA、B、33 ALU間相互接続用スイッチ回路、34 Xレジスタ、35 XHレジスタ。

【図1】



【図2】



10

20

【 9 】

(a) Code : mx2_sub_k
Name : constant subtraction (2bit-mode)
Prototype: void mx2_sub_k (int ap, int kp, int n);
Function : $A[ap:n]=A[ap:n]-K[kp:n]$
Argument: ap: destination address [AP]
 kp: source address [MP]
 n : bit length [LC0]

(b) Code : mx_sub_k
Name : constant subtraction (1bit-mode)
Prototype: void mx_sub_k (int ap, int kp, int n);
Function : $A[ap:n]=A[ap:n]-K[kp:n]$
Argument: ap: destination address [AP]
 kp: source address [MP]
 n : bit length [LC0]

フロントページの続き

(74)代理人 100124523

弁理士 佐々木 真人

(74)代理人 100098316

弁理士 野田 久登

(72)発明者 水本 勝也

東京都千代田区丸の内二丁目4番1号 株式会社ルネサステクノロジ内

(72)発明者 野田 英行

東京都千代田区丸の内二丁目4番1号 株式会社ルネサステクノロジ内

審査官 高橋 正徳

(56)参考文献 特許第3212634(JP, B2)

特開平11-007388(JP, A)

特開平11-053186(JP, A)

特開平10-326258(JP, A)

特開2000-182037(JP, A)

特表2005-515525(JP, A)

特開平11-007387(JP, A)

特開平11-053187(JP, A)

藤田 善弘 Yoshihiro FUJITA, メモリ集積型SIMDプロセッサIMAP A 3.84 GIPS Integrated Memory Array Processor, 電子情報通信学会論文誌 (J78-D-I) 第2号 THE TRANSACTIONS OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, 日本, 社団法人情報通信学会 THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, 1995年 2月25日, 第J-78-D-I巻 第2号, p.82-90

(58)調査した分野(Int.Cl., DB名)

G06F 15/80,

G06F 12/06,

G06F 9/30,

G06F 9/38