

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4015411号
(P4015411)

(45) 発行日 平成19年11月28日(2007.11.28)

(24) 登録日 平成19年9月21日(2007.9.21)

(51) Int.C1.

F 1

G06F 7/00 (2006.01)
G06F 9/30 (2006.01)G06F 7/00 201A
G06F 9/30 370

請求項の数 4 (全 25 頁)

(21) 出願番号 特願2001-379549 (P2001-379549)
 (22) 出願日 平成13年12月13日 (2001.12.13)
 (65) 公開番号 特開2003-177911 (P2003-177911A)
 (43) 公開日 平成15年6月27日 (2003.6.27)
 審査請求日 平成16年12月10日 (2004.12.10)

(73) 特許権者 503121103
 株式会社ルネサステクノロジ
 東京都千代田区大手町二丁目6番2号
 (74) 代理人 110000350
 ポレール特許業務法人
 (74) 代理人 100068504
 弁理士 小川 勝男
 (72) 発明者 山田 哲也
 東京都国分寺市東恋ヶ窪一丁目280番地
 株式会社日立製作所 中央研究所内
 (72) 発明者 荒川 文男
 東京都国分寺市東恋ヶ窪一丁目280番地
 株式会社日立製作所 中央研究所内

審査官 石井 茂和

最終頁に続く

(54) 【発明の名称】演算装置及びその演算装置を用いた情報処理装置

(57) 【特許請求の範囲】

【請求項 1】

入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、
 前記第1及び第2のデータに上位側と下位側の有効の情報を有するサイズ情報を有し、
 前記第1及び第2のデータの上位側と、下位側と、サイズ情報を保持する記憶手段と
 、
 前記第2のデータを反転する反転手段と、
 前記第2の上位側のデータの反転を抑止する反転抑止手段と、
 演算の制御を行う制御手段と、
 前記第1のデータの下位側と、前記第2のデータの下位側の反転手段出力と、前記制御手段の制御信号とを入力とする第1の演算手段と、
 前記第1及び第2の下位側のデータの最上位ビットを上位側のデータとして使用するための論理手段と、
 前記論理手段出力に対し、前記第1の上位側のデータと、反転抑止手段とを選択する上位側入力選択手段と、
 前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報と、前記制御手段の制御信号とを入力とする桁上げ生成手段と、
 前記桁上げ生成手段出力の桁上げを記憶する桁上げ記憶手段と、前記上位側入力選択手段の出力と前記桁上げ生成手段出力とを入力とする第2の演算手段と、
 前記第1及び第2のデータのサイズ情報から結果のサイズ情報を生成するサイズ情報生

10

20

成手段と、
から構成され、

前記第1の演算手段から演算結果の下位側と、前記第2の演算手段から演算結果の上位側とを生成することを特徴とする演算装置。

【請求項2】

入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、

前記第1及び第2のデータに上位側と下位側の有効の情報を有するサイズ情報を有し、

前記第1及び第2のデータの上位側と、下位側と、サイズ情報を保持する記憶手段と

、
前記第2のデータを反転する反転手段と、 10

前記第2の上位側のデータの反転を抑止する反転抑止手段と、

演算の制御を行う制御手段と、

前記第1のデータの下位側と、前記第2のデータの下位側の反転手段出力と、前記制御手段の制御信号とを入力とする第1の演算手段と、

前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報を、前記制御手段の制御信号とを入力とし、桁上げを生成する桁上げ生成手段と、

前記第1及び第2の下位側のデータの最上位ビットと前記第1及び第2のデータのサイズ情報を前記桁上げ生成手段出力の桁上げと、前記第1の上位側のデータと、
反転抑止手段の出力とを入力とする第2の演算手段と、

前記第1の上位側のデータと、反転抑止手段と、前記桁上げ生成手段出力の桁上げとを入力とする第3の演算手段と、 20

前記桁上げ生成手段出力の桁上げを記憶する桁上げ記憶手段と、

前記第2及び第3の演算手段から結果を選択する上位演算結果選択手段と、

前記第1及び第2のデータのサイズ情報をから結果のサイズ情報を生成するサイズ情報生成手段と、から構成され、

前記第1の演算手段から演算結果の下位側と、前記上位演算結果選択手段から演算結果の上位側とを生成することを特徴とする演算装置。

【請求項3】

入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、

前記第1及び第2のデータに上位側及び下位側の有効の情報を有するサイズ情報を有し 30

、
前記第1及び第2のデータの上位側及び下位側と、前記サイズ情報を保持する記憶手段と、

前記第2のデータを反転する反転手段と、

前記第2の下位側のデータの反転を抑止する反転抑止手段と、

演算の制御を行う制御手段と、

前記制御手段の制御信号を保持する制御信号記憶手段と、

前記制御手段の制御信号と前記制御信号記憶手段出力とを入力とする第1の選択手段と

、
前記第1のデータの下位側と、前記反転抑止手段の出力と、 40

前記第1の選択手段出力とを入力とする第1の演算手段と、

前記第1及び第2の下位側のデータと、前記第1のデータのサイズ情報を入力とする第2の選択手段と、

前記第1及び第2のデータのサイズ情報を入力として、前記第1の演算手段と前記第2の選択手段を選択する第3の選択手段と、

前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報を、前記制御手段の制御信号とを入力として、桁上げを生成する桁上げ生成手段と、

前記第1の上位側のデータと、前記第2のデータの上位側の反転手段出力と、前記桁上げ生成手段出力の桁上げとを入力とする第2の演算手段と、

前記第1及び第2のデータのサイズ情報をから結果のサイズ情報を生成するサイズ情報生 50

成手段と、
から構成され、

前記第3の選択手段から演算結果の下位側と、前記第2の演算手段から演算結果の上位側とを生成することを特徴とする演算装置。

【請求項4】

請求項1～3のいずれか1項に記載の演算装置と、
バス上のデータの入力を行う際に、サイズ情報を付加するサイズ情報付加手段と、
前記バス上のデータの出力を行う際に、前記サイズ情報をデータに反映するためにデータを変更する論理手段を持つデータ変更論理手段と、
前記演算装置と、前記サイズ情報付加手段と、前記データ変更論理手段とを接続して、
データの記憶とサイズ情報の更新を行うサイズ情報更新手段と、
から構成されることを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マイクロプロセッサやデジタルシグナルプロセッサ等に使用される演算装置およびその演算装置を用いた情報処理装置に関する。

【0002】

【従来の技術】

近年のマイクロプロセッサでは、レジスタファイルや演算器群、それらを結ぶ内部バスを同ビット幅とし、制御論理を簡単化することで、ハードウェアの複雑化を避け、大幅な周波数向上を遂げてきた。一方、機器組み込み向けアプリケーションプログラムでは、コストに直接影響するメモリ容量を削減するために、必要最低限のビット幅のデータを使用することが多い。

【0003】

そこで、メモリからのロードデータや即値の有効ビット幅が演算器やレジスタファイルのビット幅より小さい場合、演算器やレジスタファイルのビット幅に合うよう符号拡張される。例えば、画像情報で有効ビット幅が8ビットのとき、演算器ビット幅が32ビットであれば、上位24ビットを符号拡張し、32ビットデータとして演算される。有効ビット8ビットのみ動作させる場合に対し、上位24ビット分の電力が多く消費される。

【0004】

このため、消費電力を抑えるために演算器を有効ビット幅で動作させるいくつかの方法が知られている。

【0005】

例えば、特開平7-248895号公報（従来例1）に示されるように、Nビットの演算器を上位側の（N-K）ビットと下位側Kビットに分割し、入力データがKビットのみ有効である場合は、演算器の入力データの（N-K）ビットを“0”（論理レベル）とし、キャリー伝搬を抑止することで、上位側の演算器を停止させる。

【0006】

また、特開平7-271554号公報（従来例2）に示されるように、入力データの上位側のビットを“0”とする代わりに、上位側のラッチ更新を抑止することによって同様の低消費電力化が行われる。この場合、データの有効ビット幅の制御は、低消費電力のためのモードビットを参照して行っている。

【0007】

別の方法として、特開2000-112712号公報（従来例3）に示されるように、入力データから有効ビット幅を検出する方法がある。例えば8ビット加算器で、入力データから有効桁をチェックし、下位4ビットのみが有効とすると、上位側のビットを“0”に固定する。そして、加算結果は下位4ビットと下位の最上位ビットの符号拡張から得られる。但し、下位桁から上位桁へのキャリー伝搬は抑止されていない。

【0008】

10

20

30

40

50

このように、低消費電力のために演算器の動作ビット数を減少する方法がいくつか知られている。すなわち、従来の演算器の制御方法として、キャリー伝搬抑止に加え、

制御方法 A： 上位側加算器の入力を“0”固定する方法（従来例1や従来例3に相当）

、

制御方法 B： 上位側ラッチ更新抑止による上位側データを固定する方法（従来例2に相当）、が知られている。

ここで、キャリー伝搬抑止とは、下位側のみ有効の場合に、上位側にキャリー“0”を出力することを指す。

【0009】

また、従来の演算器の制御単位としては、

10

制御単位 A： 特殊レジスタに設定するモードビットを参照する方法、

制御単位 B： 命令をデコードし、命令単位で制御する方法、

制御単位 C： データによって制御する方法、が知られている。

【0010】

【発明が解決しようとする課題】

しかしながら、前述した上記の従来の演算器の制御方法 A と B によれば、次のような問題点がある。

上位側加算器の入力を“0”固定する制御方法 A の場合、上位側を使用しない下位桁のみ有効命令であっても、最初のサイクルの電力消費は抑止できないという問題点がある。なお、2サイクル目以降、下位桁のみ有効な命令が連続する場合には電力が消費されない。上位側ラッチ更新抑止による上位側データを固定する制御方法 B の場合、上位側を使用しない下位桁のみ有効命令であっても、直前の命令で下位加算器から上位加算器へのキャリー信号が“1”（論理レベル）であったとき、最初のサイクルの電力消費を抑止することができないという問題点がある。下位桁のみ有効な命令では、常にキャリー信号として“0”が供給されるためである。なお、2サイクル目以降、下位桁のみ有効な命令が連続する場合には電力が消費されない。

20

【0011】

また、前述した従来の演算器の制御単位 A, B, C によれば、次のような難点がある。

モードビットを参照する制御単位 A の制御方法の場合、実現は最も容易であるが、モードビット設定のためのオーバヘッドがあるという点である。モードビットを設定するためには、モードビットを設定するための命令を実行する必要がある。

30

命令単位で制御する制御単位 B の場合、各命令を制御部にてデコードし、必要ビット幅の制御を出力する。モードビット参照の制御単位 A の方法に比べれば、命令単位で制御が行えるので電力削減の効果が大きいが、演算器を最大幅使用すると定義した命令では、データの有効桁によらず常に演算器は最大幅で動作するという問題がある。

データ単位で制御する制御単位 C の制御方法が、最も電力削減効果が高い方法である。データの有効桁が命令の有効桁より小さい場合は、命令で定義した演算器の使用幅より小さい幅でしか動作しないという利点がある。実現方法は入力データをチェックし、符号拡張部分を調べ、最小のビット幅で演算させればよい。

【0012】

40

ところで、CPU (Central Processing Unit) あるいはDSP (Digital Signal Processor) の電力としては、演算器のみではなく、レジスタファイルやレジスタファイルと演算器が接続される内部バスで消費される電力も無視できない。そして、レジスタファイルの上位ビットが固定された場合、レジスタファイルの上位ビットの電力が消費しない利点がある。また、内部バスにおいても、上位ビットが固定された場合、内部バスの上位ビットの電力が消費しない利点がある。

【0013】

そこで、本発明の目的は、下位桁のみ有効な命令が実行されるとき、最初のサイクルの電力消費をも抑止することができる演算装置を提供することである。

【0014】

50

また、この本発明に係る演算装置を用いて、データ単位で演算器のみでなく、CPUあるいはDSP全体の消費電力を削減することができる情報装置を提供することも本発明の目的の一つである。

【0015】

【課題を解決するための手段】

本発明に係る演算装置は、入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、

前記第1及び第2のデータに上位側と下位側の有効の情報を有するサイズ情報を有し、前記第1及び第2のデータの上位側と、下位側と、サイズ情報を保持する記憶手段と、前記第2のデータを反転する反転手段と、

前記第2の上位側のデータの反転を抑止する反転抑止手段と、

演算の制御を行う制御手段と、

前記第1のデータの下位側と、前記第2のデータの下位側の反転手段出力と、前記制御手段の制御信号とを入力とする第1の演算手段と、

前記第1及び第2の下位側のデータの最上位ビットを上位側のデータとして使用するための論理手段と、

前記論理手段出力に対し、前記第1の上位側のデータと、反転抑止手段とを選択する上位側入力選択手段と、

前記桁上げ記憶手段と、前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報を、前記制御手段の制御信号とを入力とする前記第2の演算手段の入力の桁上げを生成する桁上げ生成手段と、

前記桁上げ生成手段出力の桁上げを記憶する桁上げ記憶手段と、

前記上位側入力選択手段の出力と前記桁上げ生成手段出力とを入力とする第2の演算手段と、

前記第1及び第2のデータのサイズ情報をから結果のサイズ情報を生成するサイズ情報生成手段と、

から構成され、

前記第1の演算手段から演算結果の下位側と、前記第2の演算手段から演算結果の上位側とを生成することを特徴とするものである。

【0016】

また、本発明に係る演算装置は、入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、

前記第1及び第2のデータに上位側と下位側の有効の情報を有するサイズ情報を有し、前記第1及び第2のデータの上位側と、下位側と、サイズ情報を保持する記憶手段と、前記第2のデータを反転する反転手段と、

前記第2の上位側のデータの反転を抑止する反転抑止手段と、

演算の制御を行う制御手段と、

前記第1のデータの下位側と、前記第2のデータの下位側の反転手段出力と、前記制御手段の制御信号とを入力とする第1の演算手段と、

前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報を、前記制御手段の制御信号とを入力とし、桁上げを生成する桁上げ生成手段と、

前記第1及び第2の下位側のデータの最上位ビットと前記第1及び第2のデータのサイズ情報を前記桁上げ生成手段出力の桁上げを入力とする第2の演算手段と、

前記第1の上位側のデータと、反転抑止手段と、前記桁上げ生成手段出力の桁上げとを入力とする第3の演算手段と、

前記桁上げ生成手段出力の桁上げを記憶する桁上げ記憶手段と、

前記第2及び第3の演算手段から結果を選択する上位演算結果選択手段と、

前記第1及び第2のデータのサイズ情報をから結果のサイズ情報を生成するサイズ情報生成手段と、

から構成され、

10

20

30

40

50

前記第1の演算手段から演算結果の下位側と、前記上位演算結果選択手段から演算結果の上位側とを生成する構成としてもよい。

【0017】

また、本発明に係る演算装置は、入力される第1及び第2のデータをそれぞれ上位側と下位側に分割し、

前記第1及び第2のデータに上位側及び下位側の有効の情報を有するサイズ情報を有し、前記第1及び第2のデータの上位側及び下位側と、前記サイズ情報を保持する記憶手段と、

前記第2のデータを反転する反転手段と、

前記第2の上位側のデータの反転を抑止する反転抑止手段と、

演算の制御を行う制御手段と、

前記制御手段の制御信号を保持する制御信号記憶手段と、

前記制御手段の制御信号と前記制御信号記憶手段出力を入力とする第1の選択手段と、

前記第1のデータの下位側と、前記第2のデータの下位側の反転手段出力と、

前記第1の選択手段出力を入力とする第1の演算手段と、

前記第1及び第2の下位側のデータと、前記第1のデータのサイズ情報を入力とする第2の選択手段と、

前記第1及び第2のデータのサイズ情報を入力として、前記第1の演算手段と前記第2の選択手段を選択する第3の選択手段と、

前記第1の演算手段の桁上げと、前記第1及び第2のデータのサイズ情報をと、

前記制御手段の制御信号とを入力として、桁上げを生成する桁上げ生成手段と、

前記第1の上位側のデータと、前記反転抑止手段の出力と、前記桁上げ生成手段出力の桁上げとを入力とする第2の演算手段と、

前記第1及び第2のデータのサイズ情報から結果のサイズ情報を生成するサイズ情報生成手段と、

から構成され、

前記第3の選択手段から演算結果の下位側と、前記第2の演算手段から演算結果の上位側とを生成する構成であってもよい。

【0018】

また、本発明に係る情報処理装置すなわちマイクロプロセッサやデジタルシグナルプロセッサは、上記のいずれかの演算装置と、

バス上のデータの入力を行う際に、サイズ情報を付加するサイズ情報付加手段と、

前記バス上のデータの出力を行う際に、前記サイズ情報をデータに反映するためにデータを変更する論理手段を持つデータ変更論理手段と、

前記演算装置と、前記サイズ情報付加手段と、前記データ変更論理手段とを接続して、データの記憶とサイズ情報の更新を行うサイズ情報更新手段と、

から構成されることを特徴とするものである。

【0019】

【発明の実施の形態】

以下、本発明の好適な実施の形態について、添付図面を参照しながら詳細に説明する。

【0020】

<実施の形態1>

図1は、本発明に係る演算装置の第1の実施の形態を示す加減算回路のブロック図である。加減算回路の2つの入力データを、データ1及びデータ2とする。

図1において、参照符号101及び102はデータ1を上位側と下位側に2分する際の上位レジスタと下位レジスタである。例としてデータ1のビット幅を32ビットとすると、レジスタ101は16ビットの上位側、レジスタ102は下位側に相当する。データ1には、サイズ情報S1を示すサイズビットレジスタ103がある。サイズ情報S1はデータと常に付随する情報として扱われる。同様に、もう一つの入力データであるデータ2を上位側と下位側に2分する際の上位レジスタ104及び下位レジスタ105と、データ2の

サイズ情報 S 2 を示すビットレジスタ 106 がある。上記レジスタ 101 ~ 106 は全てフリップフロップで構成され、クロック(clock) 1 の立ち上がりで動作する。

【0021】

107 及び 108 はそれぞれデータ 1 、データ 2 の上位側、下位側の加算器である。109 は制御部であり、演算の減算を示す SUB 信号と並列演算を示す SIMD 信号を出力し、演算の制御を行う。また、図 1 では省略してあるが、制御部 109 は上記レジスタ 101 ~ 106 のイネーブル信号も命令単位で出力する。

【0022】

110 及び 111 は、減算を行う際に使用するデータ 2 と SUB 信号を入力とする EOR (排他的論理和) ゲートであり、それぞれデータ 2 の上位側、下位側に対応する。112 はデータ 2 のサイズビット S 2 に従って SUB 信号をマスクするための AND ゲートである。

【0023】

113 はキャリー制御部であり、データ 1 とデータ 2 のサイズビットレジスタ 103 , 106 に格納されたサイズ情報 S 1 , S 2 と SUB 信号及び SIMD 信号を入力とし、下位側の加算器 108 から上位側の加算器 107 への桁上げ信号の制御を行う。キャリー制御用のために、キャリー保持レジスタ 114 により以前のキャリー C を保持する。サイズ情報が異なるデータの演算を行うときは、サイズを合わせるためにビット長の小さいデータを符号拡張する必要があり、符号拡張制御部 115 にてサイズビットレジスタ 103 , 106 を入力として符号拡張制御を行う。加算器 107 , 108 の出力ライン 116 , 117 に、データ上位側及び下位側の演算結果データ D3H 及び D3L をそれぞれ出力する。演算結果に付随する情報として、信号ライン 118 に結果サイズ情報 S 3 を出力する。

【0024】

ここで、図 1 に示す本実施の形態における加減算回路の動作を説明する前に、表 1 と図 2 を参照して、加減算の方法を説明する。データ 1 とデータ 2 は、32 ビット符号付整数とし、上位 16 ビットをそれぞれ D1H , D2H と表し、下位 16 ビットをそれぞれ D1L , D2L と表す。サイズビットは 1 ビットであり、“1”的とき 32 ビット、“0”的とき 16 ビットを示す。

【0025】

【表 1】

30

表 1

命令	内容	CIU	CIL
ADD32	32+32, 32+16, 16+32	CO	0
ADD16	16+16	C	0
SUB32	32-32, 32-16, 16-32	CO	1
SUB16	16-16	C	1
ADDS	16+16(SIMD)	0	0
SUBS	16-16(SIMD)	1	1

40

表 1 には、本実施の形態における加減算回路で実行される命令と、命令の内容と、上位側加算器 107 へのキャリーイン CIU と、下位側加算器 108 へのキャリーイン CIL とを示している。減算命令は“データ 1 ” - “データ 2 ”のみ用意されているが、“データ 2 ” - “データ 1 ” の拡張は容易であり、説明を簡単にするために、“データ 1 ” - “データ 2 ” を実現する回路となっている。

50

【0026】

表1において、COとは下位側加算器108のキャリーアウトを示し、Cとはキャリー保持レジスタ114に保持されたキャリーを示す。ADD32では少なくとも一方が32ビットのデータでの加算命令を示し、サイズ情報の組合せにより、32+32、16+32、32+16の3通りの演算が起こり得る。

SUB32は、ADD32の加算を減算に置換えた命令である。ADD16とSUB16とはそれぞれ、2つの下位16ビットでの加算と減算を行う命令である。ADDSとSUBSとはそれぞれ、32ビットデータで上位16ビット、下位16ビットを並列に加算と減算を行う命令である。

【0027】

10

図2は、表1の命令のうち4通りの加減算の例を示している。

(1) データ1が32ビット、データ2が16ビットの加算(ADD32)の場合：

データ1は32ビットなので、サイズビットが“1”、データ2は16ビットなので、サイズビットは“0”である。データ幅が異なる場合、データのビット長が大きいデータとビット長を合わせるために、ビット長が小さいデータは符号拡張を行う。従って、データ2は符号拡張される。32ビット加算なので下位側加算のキャリーインCILは“0”であり、上位側加算のキャリーインCIUは下位側からの桁上げ(キャリーアウト)COである。結果は32ビットであり、サイズは“1”となる。

【0028】

(2) データ1が32ビット、データ2が16ビットの減算(SUB32)の場合：

20

データ1は32ビットなので、サイズビットが“1”、データ2は16ビットなので、サイズビットは“0”である。32ビット減算なので、データ2は符号拡張と反転を行い、下位側加算のキャリーインCILは“1”であり、上位側加算のキャリーインCIUは下位側からの桁上げCOである。結果は、32ビットであり、サイズは“1”となる。

【0029】

(3) データ1、データ2が16ビットの加算(ADD16)の場合：

データ1、データ2は16ビットなのでサイズビットは“0”である。上位側は演算に無関係なため、ドントケアである。そこで、上位側は前サイクルと同じ結果となるようにする。入力データを前サイクルと同じデータにし、上位側加算のキャリーインCIUを前サイクルと同じキャリーCとすればよい。下位側加算のキャリーインCILを“0”とすれば、結果の下位16ビットは正しい演算結果が得られる。結果のサイズは“0”となる。

30

【0030】

(4) データ1、データ2が16ビットの並列減算(SUBS16)の場合：

データ1、データ2は32ビットなのでサイズビットは“1”である。上位側と下位側を並列に演算させるため、データ2の入力を反転し、上位側加算のキャリーインCIU、下位側加算のキャリーインCILをどちらも“1”とする。結果は32ビットであり、サイズは“1”となる。

【0031】

次に、図1の加減算器の動作を図2の(2)のSUB32(32ビット-16ビット)を例にとり説明する。データ1入力は32ビットなので、データ1のサイズビットレジスタ103に“1”、データ1の上位レジスタ101と下位レジスタ102にそれぞれD1HとD1Lが格納される。

40

【0032】

データ2入力は16ビットなので、データ2のサイズビットレジスタ106に“0”、データ2の下位レジスタ105にD2Lが格納され、上位レジスタ104は変化しない。制御部109から減算用の制御信号であるSUB信号には“1”が供給される。SUB信号により、データ2の下位レジスタ105に格納されたデータの出力D2LがEORゲート111により反転されて下位側の加算器108に与えられると共に、下位側の加算器108にキャリーインCILとしてSUB信号の“1”が与えられる。下位側の加算器108において、データ1の下位レジスタ102に格納されたD1Lと、データ2の下位レジス

50

タ105の反転出力 D2L (は、反転記号の代用である) 、キャリーイン CIL = “1”が入力として与えられ、加算した後、下位側加算器108の結果D3Lが出力ライン117に出力される。下位側加算器108のキャリーアウトCOは、キャリー制御部113に入力されるが、スルー出力で上位側加算器107のキャリーインCIUとして入力される。データ1とデータ2のビット幅を合わせるために、データ2は下位側の最上位ビット(MSB: Most Significant Bit)が符号拡張され、符号拡張制御部115に入力される。EORゲート111の出力のMSBを使用するため、データ2の下位レジスタ105のMSBに対し、反転された符号が拡張されている。符号拡張制御部115では、サイズ情報としてデータ1、データ2のサイズビットレジスタ103、106と上位側データD1H, D2Hと符号拡張された下位側データD1L, D2Lとを入力し、データ幅が同じ場合は上位側データを、データ幅が異なる場合は、データ幅が小さい方のセレクタは符号拡張下位側データを選択する。

【0033】

図2の(2)の例では、データ1側はデータ1の上位レジスタ101の出力D1Hを選択し、データ2側は符号拡張下位側データを選択して、それぞれ上位側加算器107に入力される。上位側加算器で加算し、結果を上位側データD3Hとして出力ライン116に出力する。信号ライン118には結果のサイズS3は、入力サイズのどちらかが32ビットのときは“1”、どちらも16ビットデータのときは“0”となる。図2の(2)の例では結果サイズS3として“1”を出力する。

【0034】

図2の(1)のADD32(32ビット+16ビット)は上記SUB32に対し、制御部109のSUB信号が“0”となることにより実行される。下位側加算器108の変更点は、データ2の入力とキャリーインであるが、SUB信号が“0”となることにより下位側加算器108のデータ2の入力はスルーとなり、キャリーインCILは“0”となる。上位側加算器107のデータ2側のデータとしては、データ2の下位レジスタ105のMSBと同じ値が符号拡張され、供給される。

【0035】

ADD32命令とSUB32命令の上記と異なるデータ幅においては、2つの入力データが32ビットのときは、符号拡張の代わりに上位側のデータを使用することにより実現できる。データ1のビット幅が16ビット、データ2のビット幅が32ビットのときは、データ1の下位側データD1LのMSBを符号拡張することにより実現できる。

【0036】

図2の(3)のADD16(16ビット+16ビット)は、本実施の形態において、上位側の加算器107のトランジスタのスイッチングを抑止する点で、従来例に対し低消費電力を達成する好適な例である。2つの入力はどちらも16ビットであるため、データ1、データ2のサイズビットレジスタ103, 106は両方とも“0”である。したがって、データ1、データ2の上位側レジスタ101, 104は変更されず、前の値を保持する。データ1、データ2の下位側レジスタ102, 105に入力データが格納される。加算のため、制御部109からSUB信号は“0”が出力される。

【0037】

SUB信号の“0”に従って、下位側加算器108の入力として、データ1の下位レジスタ102の出力D1Lと、データ2のEORゲート111のスルー出力と、キャリーインCILの“0”とが与えられ、加算が実行される。加算結果は16ビットであり、結果サイズビットS3は“0”となる。

【0038】

ここで、ADD16命令においては、演算結果の実行に、上位側の加算器107を使用しない。上位側の加算器のスイッチングを停止することで、消費電力が削減できる。上位側加算器の消費電力を削減する方法を説明する。上位側の加算器107のスイッチングを停止するためには、加算器の2つの入力とキャリーインを固定すればよい。固定の方法は次のように行う。

10

20

30

40

50

【0039】

まず、データ1、データ2のサイズビットレジスタ103, 106がどちらも“0”であることから、符号拡張制御部115のセレクタは、H側すなわちデータ1、データ2の上位側D1H, D2Hが選択される。データ1の上位側レジスタ101は前の値が保持されているため固定、データ2側は、データ2のサイズビットレジスタ106が“0”であることからSUB信号がマスクされ、EORゲート111は、前の値を保持するデータ2の上位側レジスタ104のデータD2Hをスルーで出力するため固定であり、上位側加算器107の2つの入力データは固定となる。キャリーイン信号に関しては、並列実行を示すSMD信号が“0”、2つのサイズビットレジスタ103, 106が“0”的ため、前のキャリーを保持するキャリー保持レジスタ114の出力Cが上位加算器107のキャリーイン信号CIUとして選択される。キャリー保持レジスタ114の出力が固定値のため、上位加算器107のキャリーイン信号も固定値である。従って、上位加算器107の入力が全て固定され、トランジスタのスイッチングが抑止され、電力を消費しない。10

【0040】

SUB16命令では、SUB信号が“1”となるため、下位側のデータ2とキャリーインCILとが反転される点が異なるが、他はADD16命令と同様の処理で実現される。上位側の加算器107に関してはADD16命令と同様に電力を消費しない。

【0041】

更に、ADD16命令かSUB16命令が連続するとき、制御部109からのSUB信号は命令に従って変化するが、データ2のサイズビットレジスタ106の出力S2によってANDゲート112でマスクされることにより、常に固定値の入力が上位加算器107に供給され、電力は消費されない。20

【0042】

図2の(4)のSUBS16(“16ビット”-“16ビット”的並列実行)命令は、上位側データと下位側データを次の方法で並列に実行する。下位側の加算器108に、データ1の下位レジスタ102の出力D1Lと、データ2の下位レジスタ105の反転出力D2Lと、SUB信号“1”によるキャリーインCILの入力とが与えられ、16ビット結果を出力する。2入力のデータサイズはどちらも“1”であることから、符号拡張制御部115では、データ1の上位レジスタ101の出力D1Hとデータ2の上位レジスタ104の反転出力D2Hが選択され、上位側加算器107の入力データとなる。上位側加算器のキャリーイン信号として、下位側の加算器108からの桁上げ信号伝搬を抑止する必要があるが、制御部109からの並列実行信号SMD信号により下位側からの桁上げ信号伝搬の抑止を行い、SUB信号により強制的に“1”を上位側加算器107のキャリーイン信号CIUとして与えている。30

このことにより、下位桁の影響を受けずに並列に16ビット減算を実行することができる。

【0043】

ADDSS16命令に関しては、SUB信号が“0”となることにより、上位側および下位側の加算器にそれぞれデータ2の上位レジスタ104のスルー出力D2H、下位レジスタ105のスルー出力をデータ2側の入力として与え、キャリーインCIUとして強制的に“0”とすることにより実現できる。40

【0044】

上記のように本実施の形態によれば、データ1、データ2の下位16ビット演算命令が連続するとき、上位加算器107の入力データとキャリーが固定されるため、電力を消費しない。特に、下位16ビット加算命令と下位16ビット減算命令が1命令ずつ交互に実行される場合においても、SUB信号によるデータ2の上位側データのスルーまたは反転がANDゲート112により抑止されるため、上位加算器107の入力データとキャリーが固定され、電力を消費しない。

【0045】

また、下位16ビット演算命令と32ビット演算命令が1命令ずつ交互に実行されるとき50

は、下位 16 ビット演算命令が実行するときに上位加算器 107 の電力消費が行われない。従って、下位 16 ビットのみ有効な命令が実行されるとき、最初のサイクルから電力消費を抑止することができるといえる。

【0046】

従来例の制御方法 A の上位加算器の入力を “0” 固定する方法と、制御方法 B の上位側データを固定する方法すなわち上位レジスタを固定する方法とでは、以下のような場合に電力消費を必要とするのに対し、本実施の形態の加減算回路では上述したように電力が消費されない利点がある。

【0047】

制御方法 A の上位加算器の入力を “0” 固定する方法では、下位 16 ビット演算命令が連続する場合、最初の下位 16 ビット演算命令を実行するときは、上位加算器 107 の電力を消費するが、その後の下位 16 ビット演算命令では上位加算器 107 の電力消費は抑止される。下位 16 ビット演算命令と 32 ビット演算命令が 1 命令ずつ交互に実行される場合、下位 16 ビット命令を実行するとき、必ず上位加算器 107 の入力を “0” にするため、前命令の入力が “0” 以外のときは常に上位加算器 107 の電力を消費する。

【0048】

制御方法 B の上位レジスタを固定する方法では、下位 16 ビット演算命令を実行するとき、上位レジスタを固定し、下位加算器から上位加算器 107 へのキャリーを強制的に “0” にする。下位 16 ビット演算命令が連続する場合、最初の下位 16 ビット演算命令を実行するときは、下位加算器 108 から上位加算器 107 へのキャリーを強制的に “0” にするため、前サイクルのキャリーが “1” であると、上位加算器 107 は電力を消費する。

【0049】

下位 16 ビット加算命令と下位 16 ビット減算命令が 1 命令ずつ交互に実行する場合、データ 1 とデータ 2 の上位レジスタを固定しても、データ 2 が減算命令では反転されたため、上位加算器 107 のデータ 2 側の入力が常に反転することになり、消費電力を抑止できない。

【0050】

また、下位 16 ビット演算命令と 32 ビット演算命令が 1 命令ずつ交互に実行される場合、32 ビット演算命令実行時に、常に下位加算器 108 から上位加算器 107 へのキャリーが “1” であると、下位 16 ビット演算命令実行時にキャリーを強制的に “0” にするため、下位 16 ビット演算命令においても常に上位加算器 107 の電力を消費する。

【0051】

以上のように、下位 16 ビット演算命令が連続する場合、下位 16 ビット演算命令と 32 ビット命令が交互に実行される場合においても、従来に対し消費電力を低く抑えられることを示した。

【0052】

図 1 では、簡単のために加減算器を示したが、論理ゲートを追加することにより、容易に算術論理演算装置 (A L U : Arithmetic and Logic Unit) を実現することができる。

【0053】

図 3 に、図 1 の加減算回路を用いたマイクロプロセッサの例として構成図を示す。

【0054】

D R A M や S D R A M 等の外部メモリ 301 は、外部データバス 302 および外部アドレスバス 303 によりマイクロプロセッサ 304 に接続されている。マイクロプロセッサ (M P U) 304 は、C P U 305、および外部とのデータの授受を行うバスインタフェースユニット 306 から構成され、C P U とバスインタフェースユニット間はロードストアバス 307 にて接続される。尚、上記 D R A M や S D R A M 等のメインメモリは外部に設けるのではなく、マイクロプロセッサ 304 と一緒に 1 枚の半導体基板内に内蔵されてもよい。その場合は、バス 302 や 303 は同一チップ内の内部バスで構成される。C P U 305 は、制御部 308 とデータバス部 309 から構成される。

10

20

30

40

50

【0055】

データバス部は、データのロードおよびストアを行うロードストアユニット310と、図1で示される構成のALU311と、レジスタファイル312とから構成される。図1は加減算器であるが、ALUに容易に機能拡張可能なため、以後ALUと呼ぶことにする。

【0056】

図4にレジスタファイル312の構成例を示す。このレジスタファイル312は、ロードデータとALU演算結果のどちらかを選択する入力セレクタ402と、この入力セレクタ部402で選択されたデータのサイズチェックを行うデータチェック部403と、データを保持するデータ保持部404と、ALUの入力データD1およびデータD2とストアデータをデータ保持部404から選択する出力セレクタ408とから構成される。

10

【0057】

データ保持部404はレジスタ数は8本で、サイズビットレジスタファイル405と、上位側データレジスタファイル406と、下位側データレジスタファイル407とから構成される。データ保持部404のデータ幅は32ビットで、上位16ビット、下位16ビットである。1ビットのサイズビットは、“0”的き下位16ビットのみ有効であり、“1”的き32ビットが有効であることを示す。出力セレクタ408で、ALUデータD1はレジスタ[3:0]、ALUデータD2はレジスタ[7:4]、ストアデータはレジスタ[7:0]のいずれかから任意に選択できることとする。

【0058】

図4のレジスタファイル312の動作を説明する。ロードデータをALUのデータD1として読み出すことを考える。図3に示したCPU305では、ロードストア部310とALU311にフォワーディングセレクタを備えていないことを前提としているので、図4ではロードストア部310とALU311はレジスタファイル312と接続される。

20

【0059】

まず、入力セレクタ402において、制御部308からロードデータを選択するセレクタ信号が供給され、ロードデータがデータチェック部403の入力に与えられる。

【0060】

データチェック部403では、上位側データが符号拡張であるかどうかをチェックし、入力のサイズが“0”か上位側が符号拡張であれば、サイズを強制的に0にし、それ以外は“1”とする。

30

【0061】

データ保存部404では、レジスタ[7:0]のいずれかにデータがライト(書き込み)される。例として、レジスタ[0]にライトされるとすると、サイズビットレジスタ[0]にサイズビットがライトされ、サイズビットが“0”的き下位側データレジスタ407のみライトされ、サイズビットが“1”的き上位側と下位側データレジスタ406, 407にライトされる。出力セレクタ部408において、ALUデータD1側のレジスタが[3:0]のいずれかから選択される。

ここでは、[0]レジスタが選択され、ALUデータD1側に[0]レジスタが出力される。

【0062】

図5(a)に、図4のデータチェック部403の構成図を示す。入力および出力は、どちらもサイズビットSと、データ上位側と、データ下位側であり、ビット幅はそれぞれ1ビット、16ビット、16ビットである。サイズビットはデータ下位側のMSBとデータ上位側のオール(AL1)“1”検出501、データ下位側のMSBと上位側のAL1“0”検出502、或いはサイズビットの“0”検出を行い、いずれかの条件が成立するとき、サイズビットは“0”となる。すなわち、入力が16ビットデータか、入力が32ビットデータで、かつ、データ上位側が符号拡張のときにサイズビットが“0”となる。

40

AL1“1”検出501とAL1“0”検出502は、図5(b), (c)に示されるように、それぞれANDツリー、NORツリーを構成することで実現できる。同図(b)がAL1“1”検出、(c)がAL1“0”検出である。

50

【0063】

図6に、図4のデータ保持部404のうちレジスタ[i](0 i 7)の構成図を示す。

サイズビット601、データ上位側602、データ下位側603は、1(1の反転)のクロックが供給されるラッチで構成される。ビット幅はそれぞれ1ビット、16ビット、16ビットである。サイズビット601、データ下位側のデータ603のラッチは、レジスタ[i]のライトイネーブル信号を表す制御部からのライトイネーブル[i]と、1のゲーテッドクロックである。データ上位側は、ライトイネーブル[i]とゲーテッドクロック1に加え、サイズビットとのゲーテッドクロックである。すなわち、データ上位側は、サイズビットが“1”で、かつ、上位側が符号拡張でなく、イネーブルなデータのときのみライトされる。10

【0064】

レジスタファイルはビット幅が大きいので、サイズビットが“0”的ときのデータ上位側のライト抑止による電力低減の効果は大きい。データが全て16ビットのプログラムを実行した場合、サイズビットが全て“0”となることから、レジスタファイル部において約50%の電力削減が可能である。

【0065】

図7に、図3のロードストア部310の構成例を示す。ロードストア部701はロード部702とストア部705から成り、ロードデータ、ストアデータはレジスタファイル312と接続される。ロードストアバス709はデータの上位、下位ビットの32ビットのビット幅で、CPU内の内部データ信号はサイズビットと、データ上位、下位ビットとの33ビットのビット幅で接続される。ロード出力とストア入力はどちらも33ビットである。20

【0066】

ロード部702に関して、ロードサイズの情報は、制御部308で命令をデコードすることにより得られる。16ビットロード命令の実行時はサイズビットが“0”、32ビットロード命令の実行時はサイズビットが“1”となる。そして、制御部308によりロードの適切なタイミングに応じて、32ビットロード命令では上位側ロードイネーブル信号703と下位側ロードイネーブル信号704が“1”、16ビットロード命令では下位側ロードイネーブル信号704のみが“1”となり、データが正しくロードされる。30

ストア部705に関して、ストア命令の種類とサイズ情報から適正なデータを生成しストアする。

【0067】

32ビットストア命令でデータのサイズ情報が“0”的ときは、下位データのMSBを符号拡張して生成した32ビットデータをストアする必要がある。そこで、ストア用の上位データはサイズビットが“0”的ときは下位データMSBの符号拡張、サイズビットが“1”的ときは上位データを選択するセレクタ706を用意する。そして、制御部308によりストアの適切なタイミングに応じて、32ビットストア命令では上位側ストアイネーブル信号707と下位側ストアイネーブル信号708が“1”、16ビットストア命令では下位側ストアイネーブル信号708のみが“1”となり、ストア用に生成されたデータが正しくストアされる。40

【0068】

以上により、本発明の第1の実施の形態として、図1の加減算回路(ALU)と、図4、5、6のレジスタファイルと、図7のロードストア部及びそれらを用いた図3に示すマイクロプロセッサとを示した。

【0069】

<実施の形態2>

図8に、本発明に係る演算装置の第2の実施の形態である加減算回路のブロック図を示す。前述した第1の実施の形態の図1の加減算回路(ALU)では、下位レジスタを起点とし、MSBを符号拡張し、上位側の加算器で加算するパスがタイミングが最も厳しいクリ

ティカルパスである。そこで、そのクリティカルパスを解消するのが、図8の回路構成である。図1以外は、第1の実施の形態と共通である。

【0070】

図8の加減算回路のブロック図を説明する。加減算回路の2つの入力データをデータ1とデータ2とする。図8において、参照符号801及び802はデータ1を上位側と下位側に2分する際の上位レジスタと下位レジスタである。例としてデータ1のビット幅を32ビットとすると、レジスタ801は16ビットの上位側、レジスタ802は下位側に相当する。データ1には、サイズ情報S1を示すサイズビットレジスタ803がある。同様に、804及び805はデータ2を上位側と下位側に2分する際の上位レジスタと下位レジスタであり、806はデータ2のサイズビットレジスタである。

10

【0071】

上記レジスタは全てフリップフロップで構成され、クロックの1の立ち上がりで動作する。807及び808はそれぞれデータの上位側、下位側の加算器である。809は制御部であり、演算の減算を示すSUBと並列演算を示すSMD信号を出力し、演算の制御を行う。また、図8では省略してあるが、制御部809は上記レジスタのイネーブル信号も命令単位で出力する。810及び811は減算を行う際に使用するデータ2とSUB信号を入力とするEORゲートであり、それぞれデータ2の上位側、下位側に対応する。812はデータ2のサイズビットS2に従ってSUB信号をマスクするためのANDゲートである。

【0072】

また、813はキャリー制御部であり、データ1とデータ2のサイズビットレジスタ803、806のサイズ情報S1、S2と、SUB信号、SMD信号を入力とし、下位側の加算器808から上位側の加算器807への桁上げ信号の制御を行う。キャリー制御用のため、キャリー保持レジスタ814により以前のキャリーを保持する。サイズ情報が異なるデータの演算を行うときは、サイズを合わせるためにビット長の小さいデータを符号拡張して加算する必要がある。

20

【0073】

本実施の形態では、高速化を図るため、インクリメンタ、デクリメンタを用いて、上位側を予め計算しておき、下位側から上位側加算器へのキャリーインC1Uにより上位側の加算結果を決定するインクリメンタ/デクリメンタ制御部815を使用する。上位結果セレクタ816において、データ1とデータ2のビット幅が異なるときはインクリメンタ/デクリメンタ制御部815の出力を選択し、データ1とデータ2のビット幅が同じときは上位側加算器807の結果データD3Hをライン817に出力する。ライン818にデータ下位側加算器808の演算結果データD3Lを出力する。演算結果に付随する情報として、ライン819に結果サイズS3を出力する。

30

【0074】

前述した第1の実施の形態との相違点は、第1の実施の形態が符号拡張制御により符号拡張データを上位側加算器に入力していたのに対し、本実施の形態では、インクリメンタ、デクリメンタを用いてインクリメンタ/デクリメンタ制御部815にて、上位側加算器807を用いずに高速に計算している点である。インクリメンタ/デクリメンタ制御部815内の“+1”はインクリメンタであり、入力をDとすると“D+1”を出力し、“-1”はデクリメンタであり、入力をDとすると“D-1”を出力する。インクリメンタ/デクリメンタ制御部815の動作を説明するために、表2を用いて説明する。尚、インクリメンタ/デクリメンタ制御部815以外の動作は図1と同じであるため、説明を省略する。

40

【0075】

【表2】

表 2

下位MSB	キャリー-CIU	上位桁演算結果
0	0	D+0
0	1	D+1
1	0	D-1
1	1	D+0

10

インクリメンタ／デクリメンタ制御部 815 は、入力データ 1 とデータ 2 のビット幅が異なるときに使用するので、表 2 の第 1 項の下位 MSB は、ビット幅が “0” のデータの下位 MSB である。

【0076】

第 2 項のキャリー-CIU は下位加算器 808 から上位加算器 807 へのキャリーであり、第 3 項はビット幅が “1” の上位側データを D としたときの上位側演算結果である。例としてデータ 1 のサイズを “1”、データ 2 のサイズを “0” とすると、下位 MSB はデータ 2 の下位桁の MSB を指し、データ 1 の上位側データが D となる。

20

【0077】

下位 MSB が “0” で、キャリー-CIU が “0” のときは、上位側の演算結果が D であることからデータ 1 の上位側データ D をスルー出力するように、インクリメンタ／デクリメンタ制御部 815 の出力で選択される。

【0078】

下位 MSB が “0” で、キャリー-CIU が “1” のときは、上位側の演算結果が D+1 であるので、データ 1 の上位側データ D のインクリメンタ結果を出力するように、インクリメンタ／デクリメンタ制御部 815 で選択される。

30

【0079】

下位 MSB が “1” で、キャリー-CIU が 0 のときは、上位側の演算結果が D-1 であるので、データ 1 の上位側データ D のデクリメンタ結果を出力するように、インクリメンタ／デクリメンタ制御部 815 で選択される。

【0080】

下位 MSB が “1” で、キャリー-CIU が “1” のときは、上位側の演算結果が D であることからデータ 1 の上位側データ D をスルー出力するように、インクリメンタ／デクリメンタ制御部 815 で選択される。

以上のように、インクリメンタ／デクリメンタ制御部 815 を用いて、上位側加算を行う例を示した。

【0081】

図 1 の上位側加算器を用いる場合に対し、下位桁からのキャリー-イン CIU から上位側加算結果に至る論理段数が小さいため、高速化が可能である。図 8 では、両データが 16 ビットの演算が連続するとき、インクリメンタ／デクリメンタ制御部 815 の論理が動作してしまうため、図 1 に対し、電力を消費するが、両データのサイズビットが “0” のときに、セレクタの固定、データの固定を行うことにより、図 1 と同等の消費電力化も可能である。

40

【0082】

＜実施の形態 3＞

図 9 は本発明の第 3 の実施の形態を示す固定小数点の加減算回路のブロック図である。これまでの第 1 及び第 2 の実施の形態は、データの形式が整数に限って説明してきた。DS

50

Pなどのデジタル信号処理では、データの形式として固定小数点を使用する。本実施の形態では、データの形式が固定小数点の演算においても回路に変更を加えることで対応できることを示す。

【0083】

加減算回路の2つの入力データをデータ1とデータ2とする。図9において、参照符号901及び902はデータ1を上位側と下位側に2分する際の上位レジスタと下位レジスタである。例としてデータ1のビット幅を32ビットとすると、レジスタ901は16ビットの上位側、レジスタ902は下位側に相当する。データ1には、サイズ情報を示すサイズビットレジスタ903がある。同様に、904及び905は、データ2を上位側と下位側に2分する際の上位レジスタと下位レジスタであり、906はデータ2のサイズビットレジスタである。上記レジスタは全てフリップフロップで構成され、クロック1の立ち上がりで動作する。907及び908はそれぞれデータの上位側、下位側の加算器である。909は制御部であり、演算の減算を示すSUB信号と並列演算を示すSMD信号を出力し、演算の制御を行う。また、図9では省略してあるが、制御部909は上記レジスタのイネーブル信号も命令単位で出力する。

【0084】

910及び911は、減算を行う際に使用するデータ2とSUB信号を入力とするEORゲートであり、それぞれデータ2の上位側、下位側に対応する。912は、データ2のサイズビットに従ってSUB信号をマスクするためのANDゲートである。

【0085】

913は下位桁加算器のキャリーイン制御部であり、データ1とデータ2のサイズビットレジスタ903、906を入力とし、下位側の加算器908へのキャリーイン制御を行う。キャリーイン制御部913用のためにキャリー保持レジスタ914により以前のキャリーを保持する。915は上位桁加算器907へのキャリー制御部であり、データ1とデータ2のサイズビットレジスタ903、906と、SUB信号及びSMD信号を入力とし、下位側の加算器908から上位側の加算器907への桁上げ信号の制御を行う。サイズ情報が異なるデータの演算を行うときに、下位桁をスルー出力する制御をスルー制御部916にて行う。出力ライン917及び918に、それぞれデータ上位側及び下位側の演算結果データD3H、D3Lを出力する。演算結果に付随する情報として、ライン919に結果サイズS3を出力する。

【0086】

図9の加減算回路の動作を説明する前に、図10の固定小数点における演算について説明する。図10は表1の命令のうち4通りの加減算の例を示している。なお、図10において、.は小数点位置を表す。

【0087】

(1)データ1、データ2共に32ビットの加算(ADD32)の場合:

データ1とデータ2共に32ビットなので、どちらもサイズビットは“1”である。32ビット加算なので、下位加算器908のキャリーは“0”、上位加算器907のキャリーインC1Uは下位加算器からの桁上げ(キャリーアウト)C0である。結果は32ビットであり、サイズは“1”となる。

【0088】

(2)データ1が32ビット、データ2が16ビットの加算(ADD32)の場合:

データ1は32ビットなので、サイズビットが“1”、データ2は16ビットなので、サイズビットは“0”である。サイズビットが“0”的とき、小数点位置に合わせてデータを配置すると、下位桁のデータはゼロとなる。下位桁に注目すると、データ1の下位桁D1Lと、ゼロとの加算なので、下位桁の結果はデータ1の下位桁D1Lをスルー出力すればよい。そこで、下位桁加算器908の入力として、D1側にD1Lを、D2側に前サイクルデータを、キャリーインC1Lに前サイクルキャリーCをそれぞれ与える。そしてD1Lのスルー出力は、加算器908とは別の下位結果スルー制御部916を使用する。上位側加算のキャリーインC1Uは強制的に“0”を与え、上位側加算器917にて加算す

10

20

30

40

50

る。結果は 32 ビットであり、サイズは “1” となる。

【0089】

(3) データ 1、データ 2 が 16 ビットの加算 (ADD16) の場合 :

データ 1 とデータ 2 は、16 ビットなのでサイズビットは “0” である。下位側は演算に無関係なため、ドントケアである。そこで、下位側は前サイクルと同じ結果となるようする。すなわち、入力データを前サイクルと同じデータにして、下位側加算のキャリーイン CIL を前サイクルと同じキャリー C とすればよい。上位側加算のキャリーイン CIU を “0” とすれば、結果の上位 16 ビットは正しい演算結果が得られる。結果のサイズは “0” となる。

【0090】

(4) データ 1、データ 2 が 16 ビットの並列減算 (SUBS16) の場合 :

データ 1 とデータ 2 は、32 ビットなのでサイズビットは “1” である。上位側と下位側を並列に演算させるため、データ 2 の入力を反転し、上位側加算のキャリーイン CIU、下位側加算のキャリーイン CIL をどちらも “1” とする。結果は 32 ビットであり、サイズは “1” となる。

【0091】

次に、図 9 の加減算回路の動作、特にキャリー上位制御部 915、キャリーアンダーフィード 913 について説明する。

【0092】

(1) データ 1、データ 2 が 32 ビットの ADD32 命令、または SUB32 命令の場合 :
上位桁加算器 907 と下位桁加算器 908 を用いる。入力として、データ 1 のサイズビットレジスタ 903 に “1” 、データ 1 の上位レジスタ 901 と下位レジスタ 902 にそれぞれ D1H、D1L が格納される。データ 2 も同様にサイズビットレジスタ 906 に “1” 、データ 2 の上位レジスタ 904 と下位レジスタ 905 にそれぞれ D2H、D2L が格納される。

【0093】

制御部 909 から SUB 信号が供給され、SUB32 命令のとき “1” 、ADD32 命令のとき “0” となる。キャリーアンダーフィード 913 では、SUB 信号が選択されるため、下位加算器 908 へのキャリーイン CIL は SUB32 命令のとき “1” 、ADD32 命令のとき “0” となる。上位加算器 907 へのキャリーイン CIU は、下位加算器 908 からの桁上げ CO である。上位加算器 907、下位 908 へのデータ 2 の入力は、SUB 信号との排他的論理和 EOR であるため、ADD32 命令のときはそれぞれ D2H、D2L であり、SUB32 命令のときはそれぞれ D2H、D2L である。下位桁、上位桁とともに加算結果を結果とする。

【0094】

(2) データ 1、データ 2 の一方が 32 ビットで、もう一方が 16 ビットの ADD32 命令、または SUB32 命令の場合 : 上位桁加算器 907 と下位桁スルーフィード 916 を用いる。例としてデータ 2 側を 16 ビットとすると、入力として、データ 1 のサイズビットレジスタ 903 に “1” 、データ 1 の上位レジスタ 901 と下位レジスタ 902 にそれぞれ D1H、D1L が格納される。データ 2 はサイズビットレジスタ 906 に “0” 、データ 2 の上位レジスタ 904 に D2H が格納される。データ 2 の下位レジスタ 905 は変化しない。

【0095】

制御部 909 から SUB 信号が供給され、SUB32 命令のとき “1” 、ADD32 命令のとき “0” となる。キャリーアンダーフィード 913 では、キャリー保持信号が選択されるため、下位加算器 908 へのキャリーイン CIL は C となる。上位加算器へのキャリーイン CIU は、SUB 信号が選択されるため、上位加算器 907 へのキャリーイン CIU は SUB32 命令のとき “1” 、ADD32 命令のとき “0” となる。下位桁では ADD32 命令、SUB32 命令のどちらの場合も、データ 1 の下位桁スルーフィード 916 が正しい結果となるため、下位結果スルーフィード 916 にて、データ 1 の下位桁 D1L を出力するよ

10

20

30

40

50

う制御される。

【0096】

データ2入力は、SUB信号とのEORであるため、ADD32命令のときはD2HとD2Lであり、SUB32命令のときはD2HとD2Lである。データ2側が16ビットのときは、SUB信号による反転がマスクされる。上位桁は加算結果を結果とする。

【0097】

(3)データ1、データ2のどちらも16ビットのADD16命令、またはSUB16命令の場合：

上位桁加算器907と下位桁スルー制御916を用いる。入力として、データ1のサイズビットレジスタ903に“0”、データ1の上位レジスタ901にD1Hが格納される。
データ1の下位レジスタ902は変化しない。同様に、データ2はサイズビットレジスタ906に“0”、データ2の上位レジスタ904にD2Hが格納され、データ2の下位レジスタ905は変化しない。

【0098】

制御部909からSUB信号が供給され、SUB16命令のときは“1”となり、ADD16命令のときは“0”となる。キャリーワーク制御部913では、キャリー保持信号が選択されるため、下位加算器908へのキャリーアインCILはCとなる。上位加算器907へのキャリーアインCIUは、SUB16命令が選択されるため、上位加算器へのキャリーアインCIUはSUB16命令のとき“1”となり、ADD16命令のとき“0”となる。下位桁の加算器908は、データ1、データ2の入力、キャリーアイン信号CILはいずれも前サイクルの値と同じであるため、16ビット命令が連続するときは、下位桁加算器908の電力を消費しない。

【0099】

データ2入力は、SUB信号との排他的論理和EORであるため、ADD16命令のときはD2H、D2Lであり、SUB32命令のときはD2H、D2Lである。データ2側は16ビットなので、SUB信号による反転がマスクされる。下位桁、上位桁は加算結果を結果とする。

【0100】

(4)データ1、データ2のどちらも32ビットの並行命令ADDSS16、命令またはSUBS16命令の場合：

上位桁加算器907と下位桁加算器908を用いる。入力として、データ1のサイズビットレジスタ903に“1”、データ1の上位レジスタ901と下位レジスタ902にそれぞれD1H、D1Lが格納される。データ2も同様にサイズビットレジスタ906に“1”、データ2の上位レジスタ904と下位レジスタ905にそれぞれD2H、D2Lが格納される。制御部909からSUB信号が供給され、SUBS16命令のときは“1”、ADDSS16命令のときは“0”となる。また、SMD信号が供給され“1”である。キャリーワーク制御部913では、SUB信号が選択されるため、SUBS16命令のときは“1”となり、ADDSS16命令のときは“0”となる。上位加算器907へのキャリーアインCIUも同様に、SUB信号が選択されるため、上位加算器へのキャリーアインCIUはSUBS16命令のときは“1”となり、ADDSS16命令のときは“0”となる。データ2入力は、SUB信号との排他的論理和EORであるため、ADD16命令のときはD2H、D2Lであり、SUB32命令のときはD2H、D2Lである。下位桁、上位桁は加算結果を結果とする。

【0101】

以上のように、図9に示す加減算器において、固定小数点データにおける表2の命令の動作を示した。16ビット命令では、下位側の加算器908の2つの入力データD1, D2とキャリーアイン信号CILが固定されるため、16ビット命令が連続するときは、下位側加算器908の電力が消費されないという利点を持つ。

【0102】

本実施の形態では、図3のCPU305をDSPに置きかえた場合に、データを固定小数

10

20

30

40

50

点の形式とした A L U 3 1 1 の構成例を図 9 に示した。固定小数点データ形式の D S P とするためには、レジスタファイル 3 1 2 やロードストアユニット 3 1 0 も変更する必要がある。

【 0 1 0 3 】

本実施の形態におけるレジスタファイルのデータチェック部 4 0 3 の構成例を図 1 1 に示す。図 5 ではデータの上位側が符号拡張であるかどうかを調べたが、図 1 1 では下位 1 6 ビットがゼロであるかどうかを調べ、ゼロであるときにサイズビットを“ 0 ”とする。入力、出力はどちらもサイズビットと、データ上位側と、データ下位側であり、ビット幅はそれぞれ 1 ビット、 1 6 ビット、 1 6 ビットである。

【 0 1 0 4 】

サイズビットは下位側の A 1 1 “ 0 ” 検出 1 1 0 1 あるいはサイズビットの“ 0 ” 検出を行い、いずれかの条件が成立するときにサイズビットは“ 0 ”となる。すなわち、入力が 1 6 ビットデータか、入力が 3 2 ビットデータで、かつ、下位側がゼロのときにサイズビットが“ 0 ”となる。A 1 1 “ 0 ” 検出 1 1 0 1 は図 5 (c) に示したように、それぞれ N O R ツリーにより構成される。

【 0 1 0 5 】

図 1 2 に、本実施の形態におけるレジスタファイルのデータ保持部 4 0 4 のうち、レジスタ [i] (0 i 7) の構成図を示す。図 5 ではデータのサイズが“ 0 ”のとき、上位側のラッチの更新を抑止したが、図 1 2 では、データのサイズが“ 0 ”のとき、下位側のラッチの更新を抑止する。サイズビット 1 2 0 1 、データ上位側 1 2 0 2 、データ下位側 1 2 0 3 は 1 (1 の反転) のクロックが供給されるラッチで構成される。ビット幅はそれぞれ 1 ビット、 1 6 ビット、 1 6 ビットである。サイズビット 1 2 0 1 、データ上位側のデータ 1 2 0 2 のラッチは、レジスタ [i] のライトイネーブル信号を表す制御部 9 0 9 からのライトイネーブル [i] と 1 のゲーテッドクロックである。データ下位側 1 2 0 3 はライトイネーブル [i] と 1 に加え、サイズビット 1 2 0 1 とのゲーテッドクロックである。

【 0 1 0 6 】

すなわち、データ下位側は、サイズビットが“ 1 ”で下位側がゼロでなく、イネーブルなデータのときのみライトされる。レジスタファイルはビット幅が大きいので、サイズビットが“ 0 ”のときのデータ下位側のライト抑止による電力低減の効果は大きい。データが全て 1 6 ビットのプログラムを実行した場合、サイズビットが全て“ 0 ”となることから、レジスタファイル部において約 5 0 % の電力削減が可能である。

【 0 1 0 7 】

図 1 3 に、本実施の形態におけるロードストア部の構成を示す。

ロードストア部 1 3 0 1 はロード部 1 3 0 2 とストア部 1 3 0 6 から成り、ロードデータとストアデータはレジスタファイルと接続される。ロードストアバス 1 3 1 0 はデータの上位ビットと下位ビットとの 3 2 ビットであり、 C P U 内の内部データ信号はサイズビットと、データ上位ビットと、下位ビットとの 3 3 ビットのビット幅で接続される。ロードデータ出力とストア入力は、どちらも 3 3 ビットである。

【 0 1 0 8 】

ロード部 1 3 0 2 では、ロードサイズの情報は制御部 9 0 9 において命令をデコードすることにより得られ、 1 6 ビットロード命令の実行時はサイズビットが“ 0 ”となり、 3 2 ビットロード命令の実行時はサイズビットが“ 1 ”となる。そして、 3 2 ビットロード時 (ワードロード = “ 0 ”) は、バス 1 3 1 0 上のデータがそのままロードデータとなるが、 1 6 ビットロード時 (ワードロード = “ 1 ”) には、バス上の下位データがロードデータの上位側のデータとなる。

【 0 1 0 9 】

そこで、制御部 9 0 9 によりロードの適切なタイミングに応じて、 3 2 ビットロード命令では上位側ロードイネーブル信号 1 3 0 3 と下位側ロードイネーブル信号 1 3 0 4 が“ 1 ”、かつ、ワードロード信号 1 3 0 5 が“ 0 ”となり、バス上の 3 2 ビットデータがその

10

20

30

40

50

ままロードデータとなる。16ビットロード命令では、下位側ロードイネーブル信号1304とワードロード信号1305が“1”なので、バス上の下位データが、上位側のロードデータとなり、正しくロードされる。

【0110】

ストア部1306では、ストア命令の種類とサイズ情報から適正なデータを生成しストアする。32ビットストア命令でデータのサイズ情報が“1”的ときは、上位側のストアデータをバス上の上位側に、下位側のストアデータをバス上の下位側にストアする。16ビットストア命令は、上位側のストアデータをバス上の下位側にストアし、32ビットストア命令でデータのサイズ情報が“0”的ときは、下位データをゼロにマスクした32ビットデータをストアする必要がある。そこで、ストア用の下位データは、32ビットストア時(ワードストア=“0”)にサイズビットが“0”的ときは下位データをゼロにマスクし、16ビットストア時(ワードストア=“1”)に上位側データとなるようにワードストア信号1307により動作するセレクタを用意する。そして、制御部909によりストアの適切なタイミングに応じて、32ビットストア命令では上位側ストアイネーブル信号1308と下位側ストアイネーブル信号1309が“1”、16ビットストア命令では下位側ストアイネーブル信号1309のみが“1”となり、ストア用に生成されたデータが正しくストアされる。

【0111】

図13のロード部1302においては、図7に対しバス上の下位側のデータを上位側のロードデータとするセレクタがある点、ストア部1306においては、図7の構成では符号拡張用のセレクタがあるのに対し、図13の構成では、上位側のストアデータをバス上の下位側のデータとするセレクタがあるという相違点がある。

【0112】

以上に述べた図10のALUと、図4、図11、図12のレジスタファイルと、図13のロードストアユニットを、図3の構成で使用することにより、固定小数点データによるDSPが構成できる。

【0113】

以上、本発明の好適な実施の形態についてを述べたが、本発明は前記実施の形態に限定するわけではなく、本発明の精神を逸脱しない範囲において種々の設計変更をなし得ることは勿論である。例えば、実施の形態ではビット幅に関して、全長32ビットに対し16ビットずつに2分しているが、全長や分割単位の変更の拡張も容易である。全長128ビット、8ビットずつ16分割してもよい。

また、図6と図9では、1クロックのラッチを用いて構成したが、1クロックのフリップフロップを用いたレジスタファイルへの拡張も容易である。

【0114】

【発明の効果】

前述した実施の形態より明らかなように、本発明によれば、演算データにサイズ情報を設け、データを上位と下位に分割し、データ単位にサイズビットを設け、サイズビットにより有効/無効を判定して、演算を最小の有効データ幅で動作させる。すなわち、サイズ情報に基づいて常に最小の有効データ幅での演算が可能となるため、常に最大データ幅を使用する従来の演算装置に比べて、電力削減ができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態を示す加減算回路の構成例を示すブロック図。

【図2】表1に示した命令のうちの4通りの加減算の例を示す説明図。

【図3】第1～3の実施の形態の加減算回路を適用するマイクロプロセッサのブロック図。

【図4】第1～3の実施の形態のレジスタファイルの構成例を示すブロック図。

【図5】第1及び第2の実施の形態のレジスタファイルのデータ保持部の構成例を示すブロック図。

【図6】第1及び第2の実施の形態のレジスタファイルのデータチェック部の構成例を示すブロック図。

すブロック図。

【図 7】第 1 及び第 2 の実施の形態のロードストア部の構成例を示すブロック図。

【図 8】第 2 の実施の形態の加減算回路の構成例を示すブロック図。

【図 9】第 3 の実施の形態の加減算回路の構成例を示すブロック図。

【図 10】表 1 に示した命令のうちの 4 通りの固定小数点における演算例を示す説明図。

【図 11】第 3 の実施の形態のレジスタファイルのデータチェック部の構成例を示すブロック図。

【図 12】第 3 の実施の形態のレジスタファイルのデータ保持部の構成例を示すブロック図。

【図 13】第 3 の実施の形態のロードストア部の構成例を示すブロック図。

10

【符号の説明】

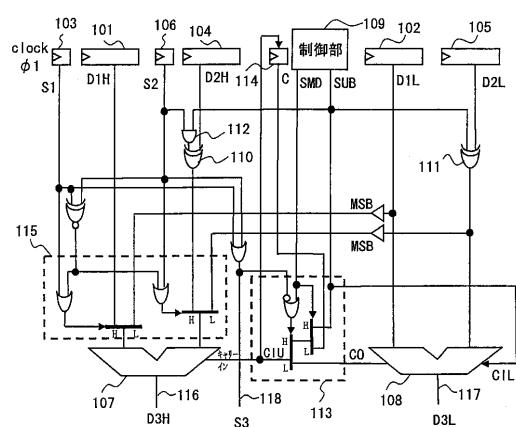
1 0 1 , 8 0 1 , 9 0 1 ... データ 1 の上位桁レジスタ、 1 0 2 , 8 0 2 , 9 0 2 ... データ 1 の下位桁レジスタ、 1 0 3 , 8 0 3 , 9 0 3 ... データ 1 のサイズビットレジスタ、 1 0 4 , 8 0 4 , 9 0 4 ... データ 2 の上位桁レジスタ、 1 0 5 , 8 0 5 , 9 0 5 ... データ 2 の下位桁レジスタ、 1 0 6 , 8 0 6 , 9 0 6 ... データ 2 のサイズビットレジスタ、 1 0 7 , 8 0 7 , 9 0 7 ... 上位桁加算器、 1 0 8 , 8 0 8 , 9 0 8 ... 下位桁加算器、 1 0 9 , 8 0 9 , 9 0 9 ... 制御部、 1 1 0 , 1 1 1 , 8 1 0 , 8 1 1 , 9 1 0 , 9 1 1 ... E OR ゲート、 1 1 2 , 8 1 2 , 9 1 2 ... AND ゲート、 1 1 3 , 8 1 3 ... キャリー制御部、 1 1 4 , 8 1 4 , 9 1 4 ... キャリー保持レジスタ、 1 1 5 ... 符号拡張制御部、 1 1 6 , 8 1 7 , 9 1 7 ... 結果上位桁の出力ライン、 1 1 7 , 8 1 8 , 9 1 8 ... 結果下位桁の出力ライン、 1 1 8 , 8 1 9 , 9 1 9 ... 結果サイズの出力ライン、 3 0 1 ... 外部メモリ、 3 0 2 ... 外部データバス、 3 0 3 ... 外部アドレスバス、 3 0 4 ... マイクロプロセッサ (MPU) 、 3 0 5 ... CPU 、 3 0 6 ... バスインターフェースユニット、 3 0 7 ... ロードストアバス、 3 0 8 ... 制御部、 3 0 9 ... データバス部、 3 1 0 , 7 0 1 , 1 3 0 1 ... ロードストア部、 3 1 1 ... ALU 、 3 1 2 ... レジスタファイル、 4 0 2 ... 入力セレクタ、 4 0 3 ... データチェック部、 4 0 4 ... データ保持部、 4 0 5 , 6 0 1 , 1 2 0 1 ... サイズレジスタ、 4 0 6 , 6 0 2 , 1 2 0 2 ... 上位桁レジスタ、 4 0 7 , 6 0 3 , 1 2 0 3 ... 下位桁レジスタ、 4 0 8 ... 出力セレクタ、 5 0 1 ... A 1 1 “ 1 ” 検出、 5 0 2 , 1 1 0 1 ... A 1 1 “ 0 ” 検出、 7 0 2 , 1 3 0 2 ... ロード部、 7 0 3 , 1 3 0 3 ... ロードイネーブル上位、 7 0 4 , 1 3 0 4 ... ロードイネーブル下位、 7 0 5 , 1 3 0 6 ... ストア部、 7 0 6 ... ストア上位桁セレクタ、 7 0 7 , 1 3 0 8 ... ストアイネーブル上位、 7 0 8 , 1 3 0 9 ... ストアイネーブル下位、 8 1 5 ... インクリメンタ / デクリメンタ制御部、 8 1 6 ... 上位結果セレクタ、 9 1 3 ... キャリー下位制御部、 9 1 5 ... キャリー上位制御部、 9 1 6 ... 下位結果スルーブル制御部、 1 3 0 5 ... ワードロード信号、 1 3 0 7 ... ワードストア信号、 1 3 1 0 ... ロードストアバス、 D 1 H ... データ 1 の上位ビット、 D 2 H ... データ 2 の上位ビット、 D 3 H ... データ上位側の演算結果、 D 1 L ... データ 1 の下位ビット、 D 2 L ... データ 2 の下位ビット、 D 3 L ... データ下位側の演算結果、 SMD 信号 ... 並列演算を示す信号、 SUB 信号 ... 減算を示す信号、 CIU ... 上位側加算器へのキャリーイン、 CIL ... 下位側加算器へのキャリーイン、 S , S 1 , S 2 , S 3 ... サイズビット。

20

30

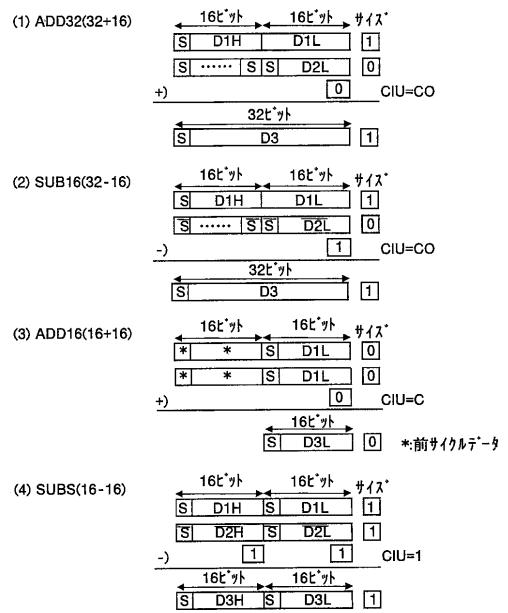
【図1】

図1



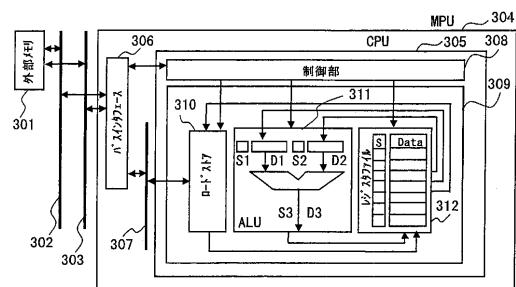
【図2】

図2



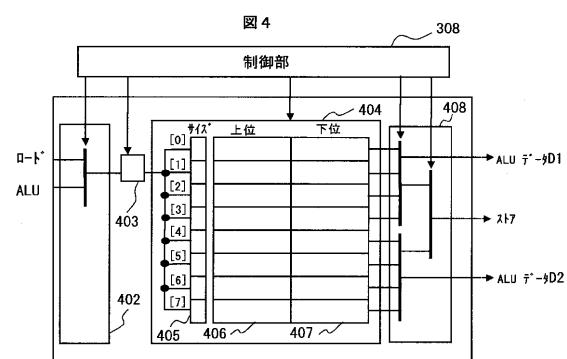
【図3】

図3



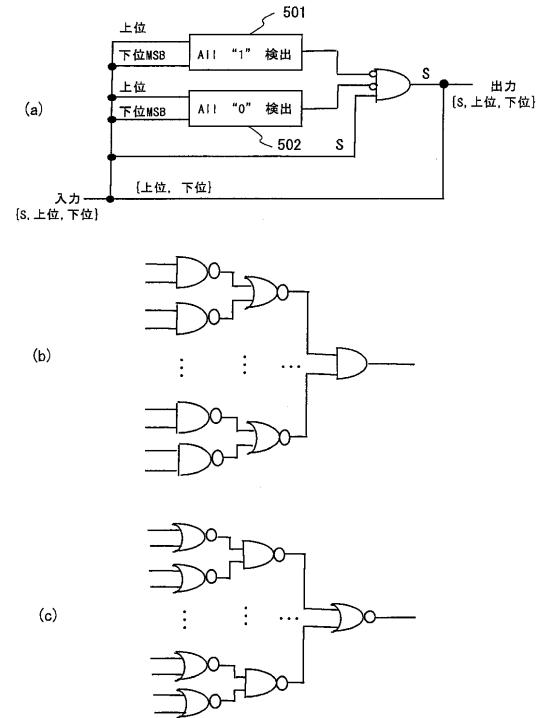
【図4】

図4

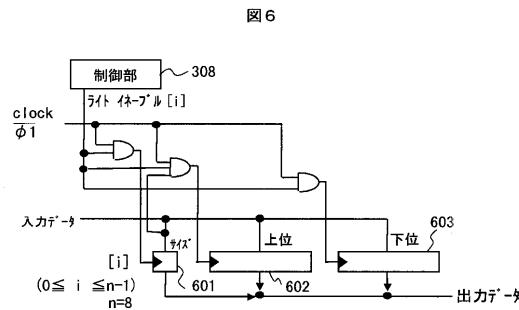


【図5】

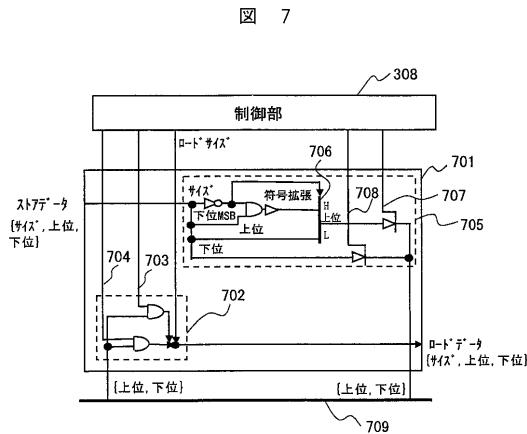
図5



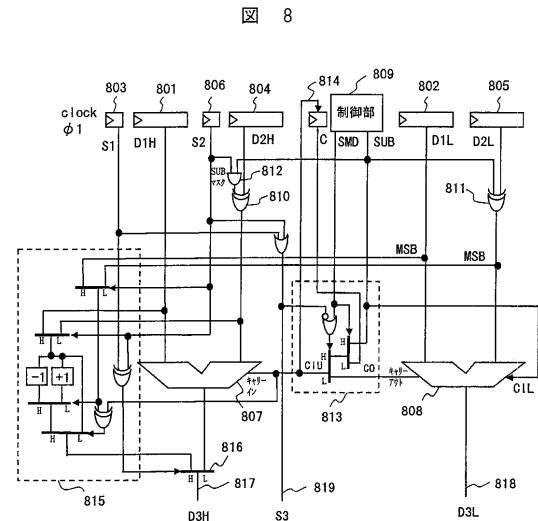
【図6】



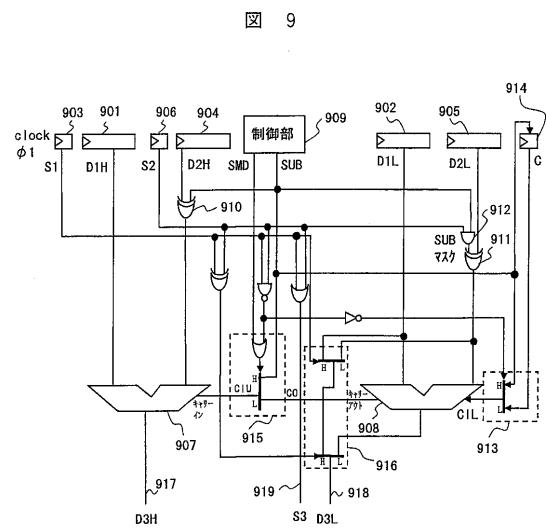
【図7】



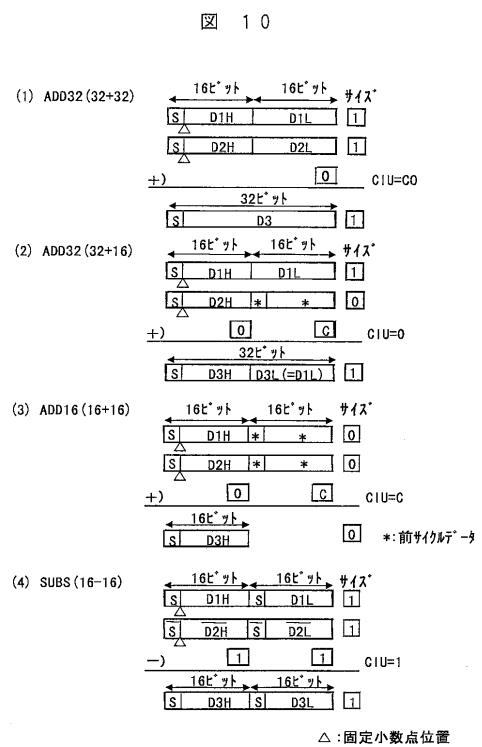
【図8】



【図9】

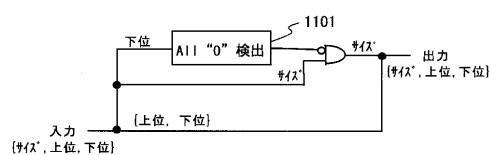


【図10】



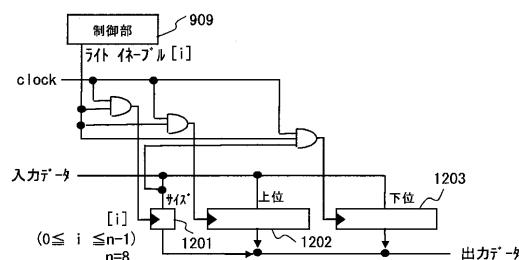
【図 1 1】

図 1 1



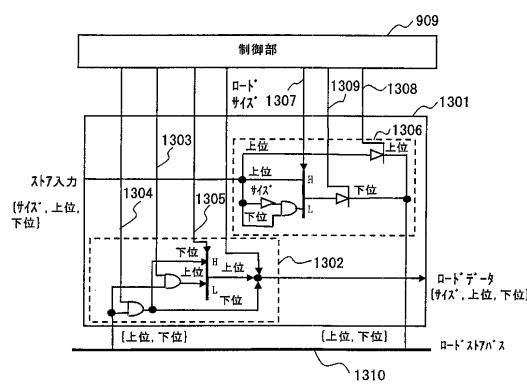
【図 1 2】

図 1 2



【図 1 3】

図 1 3



フロントページの続き

(56)参考文献 特開平07-248895(JP,A)
特開2001-216136(JP,A)
特開平07-271554(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 7/00

G06F 9/30

WPI(DIALOG)