

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3711422号
(P3711422)

(45) 発行日 平成17年11月2日(2005.11.2)

(24) 登録日 平成17年8月26日(2005.8.26)

(51) Int. Cl.⁷

F I

G06F 9/30
G06F 9/305

G06F 9/30 310E
G06F 9/30 350A
G06F 9/30 340F

請求項の数 1 (全 28 頁)

<p>(21) 出願番号 特願平8-353167 (22) 出願日 平成8年12月16日(1996.12.16) (65) 公開番号 特開平9-231070 (43) 公開日 平成9年9月5日(1997.9.5) 審査請求日 平成14年11月14日(2002.11.14) (31) 優先権主張番号 特願平7-332207 (32) 優先日 平成7年12月20日(1995.12.20) (33) 優先権主張国 日本国(JP)</p> <p>前置審査</p>	<p>(73) 特許権者 000002369 セイコーエプソン株式会社 東京都新宿区西新宿2丁目4番1号 (74) 代理人 100090479 弁理士 井上 一 (74) 代理人 100090387 弁理士 布施 行夫 (74) 代理人 100090398 弁理士 大淵 美千栄 (72) 発明者 久保田 哲 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内 (72) 発明者 工藤 真 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内</p> <p style="text-align: right;">最終頁に続く</p>
---	---

(54) 【発明の名称】 情報処理回路

(57) 【特許請求の範囲】

【請求項1】

ターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、

前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、

前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、

前記命令コード解析手段は、

前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、

前記命令実行手段は、

前記即値拡張手段により拡張された即値に基づきターゲット命令を実行し、

前記命令コード解析手段は、

プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、

前記命令実行手段は、

前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行し、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする情報処理回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理回路及び半導体集積回路装置、前記情報処理回路を内蔵するマイクロコンピュータ、該マイクロコンピュータを用いて構成された電子機器に関する。

【0002】

【背景の技術】

従来、32ビットのデータを処理できるRISC方式のマイクロコンピュータでは、32ビット幅に固定された命令コードを用いられていた。その理由は、固定長ビット幅の命令コードを用いると、可変長ビット幅の命令コードを用いる場合に比べ、命令のデコードに要する時間を短縮でき、また、マイクロコンピュータの回路規模を小さくすることが出来るからである。

【0003】

ところが、32ビットのマイクロコンピュータにおいても、特に32ビットも必要としない命令コードも多い。従って全ての命令の命令コードを32ビットで記述すると、命令コードに冗長な部分が生じる命令が多くなり、メモリの使用効率が悪くなる。

【0004】

このような場合、冗長な命令コードを論理圧縮しておき、マイクロコンピュータの内部で元の命令にデコードしながら、命令を実行することも可能である。しかし、このような方式では、制御回路が複雑になるという問題があった。

【0005】

【発明が解決しようとする課題】

そこで、本願の発明者は、制御回路を複雑にすることなくメモリの使用効率を向上させるために、処理出来るデータのビット幅より短いビット幅の固定長命令コードを処理するマイクロコンピュータについての検討を行っていた。

【0006】

しかし、例えば32ビット固定長の命令コードを単に16ビットの固定長にすると以下のような問題点が生じる。

【0007】

通常命令コードにはオペコードや即値以外のオペランドも記述するため、16ビットの命令コードを使用しても、使用できる即値のビット数は16ビットよりもさらに小さいものとなる。すなわち、32ビットのデータを処理できるにもかかわらず、16ビットより小さい即値しか命令コードで指定できないという問題が生じる。

【0008】

また、例えば、16ビットの命令コードでは、3オペランド命令のオペランドのフィールドを確保することは難しい。従って、この様に短い命令コードでは記述が難しいオペレーションをどの様に処理するのかという問題が生じる。

【0009】

上記問題を解決するためには、短い命令コードに含まれる即値を、必要に応じて拡張する機能が必要となる。また、短い命令コードで記述出来ないオペレーションを実行するために、オペレーション内容を拡張する機能が必要となる。

【0010】

本発明の目的は、制御回路を複雑にすることなく、必要に応じて命令コードに含まれる即値を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器を提供することである。

【0011】

10

20

30

40

50

また、本発明の他の目的は、制御回路を複雑にすることなく、オペレーション内容を拡張する機能を有する情報処理回路、半導体集積回路装置、マイクロコンピュータ、電子機器を提供することである。

【0012】

【課題を解決するための手段】

本願請求項1にかかる発明は、ターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、

前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、

前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、

前記命令コード解析手段は、

前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、

前記命令実行手段は、

前記即値拡張手段により拡張された即値に基づきターゲット命令を実行し、

前記命令コード解析手段は、

プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、

前記命令実行手段は、

前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行し、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする。

【0013】

【発明の実施の形態】

本実施の形態の特徴について説明する。

第1の実施の形態は、

所与のターゲット命令と、該ターゲット命令の機能を拡張するためのプリフィックス命令が入力される情報処理回路であって、

前記ターゲット命令及びプリフィックス命令を入力し、該命令コードのオペレーション内容を解析する命令コード解析手段と、

前記命令コード解析手段が解析したオペレーション内容に基づき命令を実行する命令実行手段とを含み、

前記命令コード解析手段は、

前記プリフィックス命令に基づき、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張する即値拡張手段を含むよう構成され、

前記命令実行手段は、

前記即値拡張手段により拡張された即値に基づきターゲット命令を実行することを特徴とする。

命令コード解析手段は、入力した命令コードを解読し、命令の対象となるデータの記憶されている記憶手段の番地を計算する等、命令実行手段が命令を実行するために必要な処理を行う。

【0014】

第1の実施の形態の情報処理回路には、ターゲット命令及び該ターゲット命令の機能を拡張するプリフィックス命令が入力される。プリフィックス命令は、それ単独では命令実行手段において実行されず、後続のターゲット命令が実行される際に、そのターゲット命

10

20

30

40

50

令の機能を拡張する機能を有する。

【0015】

即値拡張手段は、プリフィックス命令に基づき、ターゲット命令の実行に必要な即値を拡張する。ターゲット命令の実行に必要な即値とは、例えばターゲット命令の命令コードに含まれている即値や、プリフィックス命令の命令コードに含まれている即値であって、実行する際に必要となる即値をいう。

【0016】

ここでいう即値の拡張とは、例えば通常のゼロ拡張やサイン拡張のほか、命令コード中に記載されている即値のビット幅を伸張して、伸張された部分に、所与のビットを補う場合も含む。すなわち、第1の実施の形態によれば、プリフィックス命令に基づき、前記所与のビットを補って、即値を拡張することが出来る。

10

【0017】

従って、命令コードのビット数を大きくする原因となる即値を含む命令コードのビット数を短く設定することが可能となる。このため、可変長命令コードをデコードするための複雑な制御回路を有しなくても、短い固定長命令コードの採用するという簡単な構成で、メモリーの使用効率を改善することができる。

【0018】

第2の実施の形態は、

第1の実施の形態において、

固定長の命令コードを入力して実行するよう構成されていることを特徴とする。

20

【0019】

第2の実施の形態によれば、固定長命令コードを使用することにより、可変長ビット幅の命令コードを用いる場合に比べ、命令のデコードに要する時間を短縮でき、また、情報処理回路の回路規模を小さくすることが出来る。

【0020】

また、プリフィックス命令により、即値を拡張することが出来るため、命令コードを長くする原因となる即値を短く設定することができる。従って、固定長の命令コードの長さを短く設定することができるため、命令コードの冗長な部分を効率よく削減し、メモリー、特に命令コードを記憶するメモリーの使用効率を改善することができる。

【0021】

第3の実施の形態は、

第2の実施の形態において、

前記固定長命令コードのビット幅が、前記情報処理回路が処理出来るデータ又はアドレスのビット幅以下であることを特徴とする。

30

【0022】

第3の実施の形態によれば、命令コードのビット幅を情報処理回路が処理できるデータ又はアドレスのビット幅以下の小さなビット幅にしても、前記情報処理回路が処理できるビット幅に即値を拡張をすることができる。この様に、命令コードのビット幅を情報処理回路が処理できるデータ又はアドレスのビット幅以下にできるため、メモリーの使用効率を改善することができる。

40

【0023】

第4の実施の形態は、

第3の実施の形態において

前記情報処理回路は、16のビット幅の固定長の命令コードを入力するよう構成され、

前記即値拡張手段は、

前記プリフィックス命令に基づき、前記ターゲット命令の実行に必要な即値を32ビットの幅に拡張し、

前記命令実行手段は、

前記即値拡張手段で拡張された32ビットの即値を用いてターゲット命令を実行する算術論理演算手段を含むことを特徴とする。

50

【0024】

従来、32ビットのデータを処理できるRISC方式のマイクロコンピュータでは、固定長32ビットの命令コードを用いて、処理が行なわれていた。しかし、前記命令コードの中には、特に32ビットも必要としない命令コードも多く、これがメモリーの使用効率悪化の原因となっていた。

【0025】

すなわち、通常、命令コードは該命令の基本的なオペレーションを定めるクラスコード、オペコードや、即値以外のオペランドで構成される。ここにおいて即値以外のオペランドとは、例えばソースレジスタ(rs)やデスティネーションレジスタ(rd)に対応するコード等をさす。このため、16ビットの命令コードを使用する場合、使用できる即値のビット数はさらに小さいものとなり、通常6ビット程度となる。一般に、命令コードが短い程、即値のビット幅の確保が困難になる。しかし、本発明によれば、即値の拡張が行えるため、32ビットのデータを処理出来る情報処理回路においても、16ビットという短い命令コードが使用できる。このため、メモリーの使用効率大幅にを改善することができる。

10

【0026】

第5の実施の形態は、

第1の実施の形態～第4の実施の形態のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードの即値及び前記ターゲット命令の命令コードの即値に基づき、前記ターゲット命令の命令コードに含まれる即値を拡張することを特徴とする。

20

【0027】

このようにすると、ターゲット命令の命令コードに含まれている即値を拡張して実行したい場合、ターゲット命令の実行に必要な即値の一部をターゲット命令の命令コードの即値として指定し、ターゲット命令の実行に必要な即値の残りの部分をプリフィックス命令の命令コードの即値として指定するという簡単な構成で、即値を拡張することが出来る。

【0028】

従って、命令コード中の即値用のビット数を短かく設定することが可能となり、メモリの使用効率を改善することができる。

【0029】

第6の実施の形態は、

第1の実施の形態～第5の実施の形態のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するデータを拡張することを特徴とする。

30

【0030】

ここにおいて、データとはアドレスに対比される概念であり、その値の内容がアドレス以外を表すものをさす。ターゲット命令の実行に使用するデータとは、ターゲット命令の命令コードに含まれている即値のデータや、ターゲット命令の命令コードには含まれていないが、実行する際に使用する即値のデータをいう。このようにすると、簡単な構成で拡張されたデータの即値を用いてターゲット命令を実行することが出来る。

40

【0031】

第7の実施の形態は、

第1の実施の形態～第6の実施の形態のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の命令コードに含まれた即値を用いてターゲット命令の実行に使用するアドレスを拡張することを特徴とする。

【0032】

ここにおいて、アドレスとはデータに対比される概念であり、その値の内容が記憶装置内のある特定の位置や場所を示すための情報をいう。ターゲット命令の実行に使用するアド

50

レスとは、ターゲット命令の命令コードに含まれている即値のアドレスや、ターゲット命令の命令コードには含まれていないが、実行する際に使用する即値のアドレスをいう。このようにすると、簡単な構成で拡張されたアドレスの即値を用いてターゲット命令を実行することが出来る。

【0033】

第8の実施の形態は、

第1の実施の形態～第7の実施の形態のいずれかにおいて、

前記即値拡張手段は、

複数のプリフィックス命令の命令コードに含まれた各即値を用いて、該プリフィックス命令の機能拡張の対象となるターゲット命令の実行に必要な即値を拡張することを特徴とする。

10

【0034】

このようにすると、複数のプリフィックス命令の即値を組あわせることによって、ターゲット命令の実行に必要な即値を任意に拡張することが出来る。従って、短い命令コードを用いても、大きな即値を用いた実行が扱えるようになるため、命令コードの短縮による弊害を防ぐことが出来る。

【0035】

第9の実施の形態は、

第8の実施の形態において、

前記複数のプリフィックス命令の各命令コードは、プリフィックス命令であることを判別するための共通のコードをそれぞれ含むことを特徴とする。

20

【0036】

第9の実施の形態によれば、異なるプリフィックス命令を複数設ける必要が無い。このため、ビット数の少ない命令コードを処理する情報処理回路においても、他の種類の命令を割愛することなく、即値の拡張が可能となる。

【0037】

第10の実施の形態は、

第8の実施の形態又は第9の実施の形態のいずれかにおいて、

前記即値拡張手段は、

プリフィックス命令の連続入力回数に基づき決定される処理状態を記憶する処理状態記憶手段と、

30

前記処理状態記憶手段に記憶された処理状態に基づき、入力されたプリフィックス命令の即値を第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持する前記データ保持手段と、

前記データ保持手段の第1のレジスタ～第mのレジスタ（mは1以上の整数）に保持された即値に基づき、前記ターゲット命令の命令の実行に必要な即値を拡張して生成する手段とを含むことを特徴とする。

【0038】

ここにおいてプリフィックス命令の連続入力回数に基づき決定される処理状態とは、連続入力回数そのものによって決定される処理状態でもよいし、連続入力回数からある特定のルールに従って一意的に導き出される処理状態でもよい。例えば、プリフィックス命令を3回以上くりかえして入力した場合、3回目以降は常に同じ処理状態となるよう構成してもよい。そして、この様にして決定された処理状態を参照してプリフィックス命令の即値を即値保持手段に保持し、該保持された即値を組み合わせることによって即値を拡張することが出来る。このようにすると、簡単な構成で、プリフィックス命令の連続した入力回数に基づいて、ターゲット命令の実行に必要な即値を拡張することが出来る。

40

【0039】

第11の実施の形態は、

第1の実施の形態～第10の実施の形態のいずれかにおいて、

前記命令コード解析手段は、

50

プリフィックス命令入力後にターゲット命令を入力した場合、該ターゲット命令のオペレーション内容を拡張して解釈するターゲット命令機能拡張手段を含み、

前記命令実行手段は、

前記ターゲット命令機能拡張手段で拡張されたオペレーション内容で該ターゲット命令を実行することを特徴とする。

【0040】

このようにすると、ターゲット命令をプリフィックス命令と組み合わせて実行させることにより、ターゲット命令に拡張された機能を実現させることができる。このため、命令数を減らすことができ、さらには命令コードに使用するビット数を減らすことが可能となる。

【0041】

また、ビット数の少ない命令コードでは、実現が難しい機能を、プリフィックス命令とターゲット命令を組み合わせるにより実現することも可能となる。このため、命令コードの短縮によるメモリの使用効率を改善することができる。

【0042】

第12の実施の形態は、

第11の実施の形態において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後に2オペランド命令である所定のターゲット命令を入力した場合、該ターゲット命令の2つのオペランドとプリフィックス命令に含まれた即値を用いて、該ターゲット命令のオペレーション内容を3オペランド命令に拡張して解釈することを特徴とする。

【0043】

このようにすると、2オペランド命令のターゲット命令をプリフィックス命令と組み合わせて実行させることにより、該ターゲット命令の操作機能を3オペランド命令に拡張して実行させることができる。従って、2オペランド分記述領域しかない短い命令コードを使用しても3オペランド命令を実現することが出来る。このため、命令数及び命令コードに使用するビット数を減らすことが可能となる。

【0044】

また、さらに、プリフィックス命令で即値データを拡張して、3オペランド命令を実現することが出来るため、通常1命令で3オペランド命令を実行する場合に比べて大きな即値データを扱うことができる。

【0045】

第13の実施の形態は、

第11の実施の形態において、

前記ターゲット命令機能拡張手段は、

プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたデータ又はアドレスを修正して実行するようにオペレーション内容を拡張することを特徴とする。

【0046】

ここにおいて、データのアドレスを修正して実行するとは、データ又はアドレスに四則演算又は論理演算、シフト演算等を施して実行することをいう。このようにすると、ターゲット命令をプリフィックス命令を組み合わせて実行させることにより、ターゲット命令の命令コードで指定されたレジスタに格納されたデータ又はアドレスをプリフィックス命令の命令コードに含まれた即値で修正して実行することが出来るため、命令数又は命令コードに使用するビット数を減らすことが可能となる。

【0047】

第14の実施の形態は、

第13の実施の形態において、

前記ターゲット命令機能拡張手段は、

10

20

30

40

50

プリフィックス命令入力後にレジスタ指定値を有する所定のターゲット命令を入力した場合、プリフィックス命令の命令コードに含まれた即値に基づき、ターゲット命令の命令コードに指定されたレジスタに格納されたアドレスのディスプレイメントを作成し、該ディスプレイメントを用いて実行するようにオペレーション内容を拡張することを特徴とする。

【0048】

このようにすると、プリフィックス命令を用いることにより、実行時のアドレスのディスプレイメントを作成することが可能となる。

【0049】

さらに、プリフィックス命令を用いることにより、オペランドのビット数に限定されずにディスプレイメントの付加が可能になる。 10

【0050】

第15の実施の形態は、

第1の実施の形態～第14の実施の形態のいずれかにおいて、

前記命令コード解析手段は、

プリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張するプリフィックス命令機能拡張手段を含むことを特徴とする。

【0051】

ここにおいて、複数の種類のプリフィックス命令がある場合は、同種類のプリフィックス命令の連続入力回数に基づき、プリフィックス命令のオペレーション内容を拡張する。ここでの同種類のプリフィックス命令とは、該プリフィックス命令のオペレーション内容が同一である場合をいう。したがって、複数の種類のプリフィックス命令がある場合であって、それらの機能がオペコード等によってことなるような場合は、同種類のプリフィックス命令とは、同じオペコード等を有するものをいう。 20

【0052】

この様すると、プリフィックス命令のオペレーション内容をその入力回数に基づき変更させることが出来るため、少ない種類のプリフィックス命令で多くの機能を実現することが出来る。

【0053】

第16の実施の形態は、

第1の実施の形態～第15の実施の形態のいずれかにおいて、

前記命令コード解析手段に入力する命令コードを記憶する命令コード記憶手段さらに含むことを特徴とする。 30

【0054】

この様にすると、短くて冗長部分の少ない命令コードが格納された、メモリーの使用効率のよい情報処理装置を実現することができる。従って、同じ大きさであればより多くの命令を格納出来る情報処理装置を提供することが出来、また、同じ機能であれば、少ないメモリ容量で実現することができる。

【0055】

第17の実施の形態は、

第1の実施の形態～第16の実施の形態の情報処理回路を含むことを特徴とする半導体集積回路装置である。 40

【0056】

第17の実施の形態によれば、小さな回路構成で高機能な半導体集積回路装置を提供することが出来る。

【0057】

第18の実施の形態は、

第1の実施の形態～第17の実施の形態の情報処理回路を含むことを特徴とするマイクロコンピュータである。

【0058】

第18の実施の形態によれば、簡単な構成でメモリの使用効率のよい、マイクロコンピュータを提供することが出来る。

【0059】

第19の実施の形態は、

第18の実施の形態において、RISC方式であることを特徴とするマイクロコンピュータである。

【0060】

RISC方式のマイクロコンピュータは、固定長の短い命令コードを処理することにより、命令のデコードに要する時間を短縮し、マイクロコンピュータの回路規模を小さくすることを目的としている。従って、第19の実施の形態によれば、これらの目的を容易に実現出来るRISC方式のマイクロコンピュータを提供することが出来る。

10

【0061】

第20の実施の形態は、

第18の実施の形態又は第19の実施の形態のマイクロコンピュータを用いて制御することを特徴とする電子機器である。

【0062】

第20の実施の形態によれば、簡単な構成でメモリの使用効率のよい情報処理回路を内蔵しているため、安価で高機能な電子機器を提供することが出来る。

【0063】

以下、本発明の実施の形態を図面に基づき説明する。

20

【0064】

(実施の形態1)

1. CPUの機能説明

図1は、本発明の実施の形態のCPUの機能ブロック図である。該CPU30は32ビット幅のデータを扱うが16ビットの命令コードを処理するよう構成されている。

【0065】

本CPU30は、命令デコード部40と、命令実行部60とレジスタファイル90を含む。そしてこのCPUは、16ビットの命令データバス130と、命令データアクセスのための命令アドレスバス120と、32ビットのデータバス140と、データアクセスのためのデータアドレスバス310と、コントロール信号のためのコントロールバス390を介して外部と信号のやり取りを行う。

30

【0066】

前記命令デコード部40は、入力した命令コードを解読し、該命令を実行するために必要な処理を行うのもで、本発明の特徴的な機能であるext命令処理部50を含む。レジスタファイル90は、汎用レジスタR0~R15の16本の汎用レジスタ、プログラムカウンタ(PC)、プロセッサステータスレジスタ(PSR)、スタックポインタ(SP)、算術ローレジスタ(ALR)、算術ハイレジスタ(AHR)等のCPUで使用するレジスタを有している。命令実行部60は、前記命令デコード部40が解析した命令のオペレーション内容に基づき該命令の実行をおこなうもので、データの演算を行うデータ演算部70と、アドレスの演算を行うアドレス演算部80を含む。

40

【0067】

該CPU内部のハードウェアのデータサイズは32ビットであるため、前記命令実行部60のデータ演算部70やアドレス演算部80で行われる算術論理演算や、前記レジスタファイルのレジスタのビット数および、データの転送等はすべて32ビットになる。ところが、前記CPU30が処理する命令コードは16ビットの固定長である。すなわち、命令データバス及び命令アドレスバスを介して接続されている図示しない命令コード記憶部には、固定長16ビットの命令コードが格納されている。従って、1命令中の有効な即値のビット数は、命令コードのうちクラスコードやオペコードに占められるビット数を除いたビット数となる。このため、1命令中の有効な即値のビット数は、例えば6ビットから8ビットという比較的短いビット幅に制限されている。

50

【0068】

そこで、本実施の形態では、e x t命令という固定長16ビットの命令を新たに設け、該命令を用いることにより即値のビット幅を拡張させることが出来るよう構成している。また、e x t命令を用いることにより固定長16ビットという短いビット幅の命令コードでは記述出来ないオペレーションを可能にするよう構成している。

【0069】

e x t命令は、それ単独ではCPUにおける実行を何ら行わないが、ターゲット命令の直前におくことにより、そのターゲット命令の機能を拡張する命令であるプリフィックス命令である。なお、該プリフィックス命令による機能拡張の対象となる命令を、ターゲット命令という。

10

【0070】

このようなe x t命令の処理を行うのがe x t命令処理部50であり、詳細は後述する。

【0071】

2. 半導体集積回路の構成

マイクロコンピュータに内蔵されている半導体集積回路を例にとり、本発明の特徴的な機能を実現するための回路構成の一例と動作内容について説明する。

【0072】

図2は、マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。該半導体集積回路100は、CPU30と、ROM110、図示しないRAMを含む。前記CPU30は、前記ROM110、図示しないRAM等と、16ビットの命令データバス130、該命令データアクセスのための命令アドレスバス120、32ビットのデータバス140とを介して、外部と信号のやり取りを行う。

20

【0073】

前記ROM110には、CPU30に実行させる処理を記述した16ビットの命令コードが記憶されており、命令コード記憶手段として機能する。そして、該ROM110に通常の命令の命令コードと同じように、前記プリフィックス命令やターゲットとなる命令の命令コードが記憶される。

【0074】

前記CPU30は、命令レジスタ150と、命令デコード回路160と、即値生成回路170と、EXT1レジスタ172、EXT2レジスタ174、レジスタファイル90と、ALU190とを含む。

30

【0075】

命令レジスタ150は、前記ROM110から、命令データバス130を介して入力された命令コードを格納する。命令デコード回路160は、命令レジスタ150に格納された命令コードを解釈し、実行に必要な制御信号を出力する。また、命令コード中の即値を切り出し、必要に応じてEXT1レジスタ172、EXT2レジスタ174に保持したり、即値生成回路170に出力する。

【0076】

即値生成回路170は、前記命令デコード回路160から出力された即値及びEXT1レジスタ172、EXT2レジスタ174に保持された即値を用いて、32ビットに拡張された即値を生成する。すなわち、命令デコード回路160と即値生成回路170とEXT1レジスタ172とEXT2レジスタ174は、図1に示された命令デコード部40及びe x t命令処理部50として機能する。

40

【0077】

ALU190は、前記拡張された即値やレジスタファイル90のレジスタに格納された値や図示しない主記憶のデータに対して、算術演算や論理演算やシフト演算を行う。すなわち、ALU190は図1に示された命令実行部40のデータ演算部70として機能する。

【0078】

次に本発明の特徴的な機能である、拡張された即値の生成を行う構成について詳細に説明する。本発明の即値の拡張は後述するよう様々なパターンがあるが、本実施の形態では、

50

プリフィックス命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす拡張を行う場合を例に取り、その詳細な構成を説明する。

【0079】

本マイクロコンピュータで処理される命令は、通常の命令と、プリフィックス命令にわたることができる。通常の命令のうち所定の命令については、プリフィックス命令のターゲットとなることができる（プリフィックス命令のターゲットになった命令をターゲット命令という）。

【0080】

本マイクロコンピュータでは、16ビット固定長命令コードを採用しているため、後述するように通常の命令では、即値を指定するフィールドとして、例えば6ビットしか確保できない。しかし、ALU190等により実行される際のデータ長は32ビットであるため、前記即値生成回路170で、32ビットに拡張された即値が生成され、ALU190に入力される。この様な場合、通常のビットの拡張としては、以下の2種類がある。すなわち、前記6ビットの上位26ビットをすべてゼロとするゼロ拡張と、前記6ビットの最上位ビットの値を上位26ビットに拡張するサイン拡張である。しかしこの様な拡張では、CPUは32ビットの演算能力があるにもかかわらず、意味のある内容を有する大きなビット数の即値を扱うことが出来ない。従って、本実施の形態では、プリフィックス命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす拡張を行うことができるような構成を採用している。本実施の形態ではこの様な拡張を、プリフィックス命令の命令コードに含まれる即値を保持する手段と、保持された即値に基づき即値のビット数を増やす手段とで実現している。

【0081】

まずプリフィックス命令の命令コードに含まれる即値を保持する手段について説明する。該手段として、命令デコード回路160は図示しないステートマシンを有している。該ステートマシンの状態は、命令デコード回路160が処理した命令が、プリフィックス命令であるか、通常の命令であるかによって遷移する。命令デコード回路160は、ステートマシンの状態に基づき、処理したプリフィックス命令の即値をEXT1レジスタ172やEXT2レジスタ174に保持させる。即値生成回路170は、ステートマシンの状態に基づいて前記EXT1レジスタ172やEXT2レジスタ174に保持された即値を用いて、拡張された即値の生成を行う。なお、保持手段は、レジスタのほかラッチ回路でもよい。

【0082】

なお、後述するように本実施の形態のマイクロコンピュータで使用するプリフィックス命令はext命令一種類のみであり、ext命令の命令コードは13ビットの即値を含んでいる。

【0083】

図3は、命令デコード回路160が有するステートマシンの状態の遷移を表した図である。前記ステートマシンは、S0、S1、S2の3種類の状態を保持することができる。S0は、ext命令（プリフィックス命令）でない通常の命令を入力した状態を表しており、S1は、1回目のext命令（プリフィックス命令）を入力した状態をあらわしており、S2は、2回目のext命令（プリフィックス命令）を入力した状態を表している。

【0084】

同図に示すように、ステートマシンの状態がS0の時、通常の命令（ext=0）を入力すると状態はS0のままであり（400）、ext命令（ext=1）を入力すると状態はS1に遷移する（410）。また、ステートマシンの状態がS1の時、通常の命令（ext=0）を入力すると状態はS0に遷移し（420）、ext命令（ext=1）を入力すると状態はS2に遷移する（430）。さらに、ステートマシンの状態がS2の時、通常の命令（ext=0）を入力すると状態はS0に遷移し（440）、ext命令（ext=1）を入力すると状態はS2のままである（450）。

【0085】

10

20

30

40

50

すなわち、前記ステートマシンは、通常の命令が入力されるとS0の状態になり、ext命令を連続して2回以上入力した場合は、再び通常の命令を入力するまでS2の状態を保持するよう構成されている。

【0086】

図4は、即値の保持に関して命令デコード回路160が行う動作のアルゴリズムを示したフローチャート図である。

【0087】

まず命令デコード回路160は、命令レジスタに入力された命令コードを解読し、該命令コードがext命令であるか否か及びステートマシンの現在の状態に基づき、ステートマシンの状態設定を行う(ステップ10)。

10

【0088】

ステートマシンの状態がS1であるときは(ステップ20)、1回目ext命令が入力された状態であり、命令デコード回路160は、1回目のext命令の13ビットの即値をext1レジスタ172に保持する(ステップ30)。

【0089】

ステートマシンの状態がS2であるときは(ステップ40)、2回以上連続してext命令が入力された状態であり、命令デコード回路160は、入力されたext命令の13ビットの即値をext2レジスタ174に保持する(ステップ50)。このため、ext命令が3回以上連続した場合は、最新のext命令の即値でext2レジスタが上書きされることになる。従って、ext命令が連続した場合は、最初のext命令と最後のext命令の即値が有効となり、その中間のext命令は無視されることになる。

20

【0090】

そして、状態がS0である場合は、通常の命令が入力された状態であるため、該命令の命令コードが即値を含むか否か判断し(ステップ60)、即値を含む場合は、該命令コードの即値を即値生成回路に出力する(ステップ70)。

【0091】

次に、保持された即値に基づき即値のビット数を増やす手段について説明する。該手段として、即値生成回路170は図示しない即値生成用信号生成手段を有しており、該即値生成用信号生成手段で生成された即値生成信号に基づき拡張された即値を生成する。

【0092】

図5は、入力される命令コード及び前記ステートマシンの状態と前記即値生成用信号生成回路が生成する即値生成用信号との関係を示したタイミングチャート図である。

30

【0093】

前記即値生成用信号生成回路が生成する信号はext信号530と、ext_low信号550とext_high信号560である。図5の510は入力される命令コードを表しており、540は前記ステートマシンの状態を表している。またクロック信号520は、図2において図示しないクロック信号発生器で生成されるか又はクロック入力端子から入力される信号である。該クロック信号520は、CPUにおける各種動作の同期をとるために用いられる。例えば、該クロック信号520の立ち上がり同期して命令アドレスバス120に命令アドレスが出力される。また、該クロック信号520の1周期(1マシンサイクル)毎に前記命令アドレスに基づいてROM110から命令コードが読み出され、命令レジスタ150に保持される。そして、読み出された命令コードに応じたオペレーションは、1マシンサイクル内で完了する。

40

【0094】

ext信号530は、ext命令が入力されたときに'1'となる信号であって、前記ステートマシンの状態遷移を生じさせる入力信号である。すなわち、前記即値生成用信号生成回路は、入力された命令がext命令である場合ext信号530を'1'にし、入力された命令が通常の命令である場合ext信号530を'0'にするよう動作する。

【0095】

ext_low信号550は、ステートマシンの状態に応じて出力される信号であり、状

50

態が 'S1' の時 '1' となる 1 クロック分遅延した遅延信号である。すなわち、前記即値生成用信号生成回路は、状態が 'S1' である時、次のクロック信号の立ち上がり同期をとって、`ext_low` 信号 550 を '1' にし、その次のクロック信号の立ち上がり同期をとって、`ext_low` 信号 550 を '0' とする。

【0096】

`ext_high` 信号 560 は、ステートマシンの状態に応じて出力される信号であり、状態が 'S2' の時 '1' となる 1 クロック分遅延した遅延信号である。すなわち、前記即値生成用信号生成回路は、状態が 'S2' である時、次のクロック信号 520 の立ち上がり同期をとって、`ext_high` 信号 560 を '1' にし、その次のクロック信号 520 の立ち上がり同期をとって、`ext_high` 信号 560 を '0' とする。

10

【0097】

図 6 は、命令コードに即値を含んでいる場合に即値生成回路が、ステートマシンの状態及び即値生成用信号に基づき即値を拡張するアルゴリズムを示したフローチャート図である。

【0098】

ステートマシンの状態が S0 で無い場合は (ステップ 110)、CPU がプリフィックス命令である `ext` 命令を処理している状態であり、通常の命令を処理している状態ではない。この場合には、即値生成回路 170 は、即値の生成を行わない (ステップ 120)。プリフィックス命令自体は、命令実行部 60 における ALU での演算等の実行は行われないため、プリフィックス命令の処理時 (ステートマシンの状態が S1 又は S2 の場合) に即値を生成する必要はないからである。

20

【0099】

ステートマシンの状態が S0 で `ext_low` 信号 550 = '1' の場合は (ステップ 110)、一回の `ext` 命令の後にターゲット命令である通常の命令を処理している状態である。この場合には、EXT1 レジスタ 172 に保持されている先の `ext` 命令の命令コードに含まれていた 13 ビットの即値と、即値生成用信号線 176 を介して入力された該ターゲット命令の命令コードに含まれていた 6 ビットの即値をつなげて出来る 19 ビットの即値を、32 ビットにゼロ拡張又はサイン拡張する (ステップ 140)。

【0100】

ステートマシンの状態が S0 で `ext_high` 信号 560 = '1' の場合は (ステップ 150)、複数回連続して入力した `ext` 命令の後にターゲット命令である通常の命令を処理している状態である。この場合には、EXT1 レジスタ 172 に保持されている一番先の `ext` 命令の命令コードに含まれていた 13 ビットの即値と、EXT2 レジスタ 174 に保持されている一番最後の `ext` 命令の命令コードに含まれていた 13 ビットの即値と、即値生成用信号線 176 を介して入力された該ターゲット命令の命令コードに含まれていた 6 ビットの即値とをつなげて 32 ビットの即値を生成する (ステップ 160)。

30

【0101】

ステートマシンの状態が S0 で `ext_low` 信号 550 = '0' かつ `ext_high` 信号 560 = '0' の場合は、ターゲット命令でない通常の命令 (直前の命令が `ext` 命令でない通常の命令) を処理している状態である。この場合には、即値生成用信号線 176 を介して入力された該通常の命令の命令コードに含まれていた 6 ビットの即値を 32 ビットにゼロ拡張又はサイン拡張する (ステップ 170)。

40

【0102】

この様に、命令デコード回路 160 および即値生成回路 170 は `ext` 命令の即値を用いて、後続のターゲット命令の即値のビット数を増やす即値拡張手段として機能する。

【0103】

なお、前記処理は、命令コードに即値を含んでいる場合である。しかし、命令コード中に即値を含んでいない場合でも、ステップ 140 の処理とステップ 160 の処理で命令コード中の即値を用いないようにし、ステップ 170 の処理を省くことにより前記処理と同様に適用出来る。

50

【0104】

e x t 命令を連続して複数回実行することにより、2回目以降のe x t 命令のオペレーション内容を拡張することが出来る。すなわち、前述したような構成をとることにより、命令デコード回路160と即値生成回路170とE X T 1レジスタ172とE X T 2レジスタ174は、プリフィックス命令の連続入力回数に基づき、該プリフィックス命令のオペレーション内容を拡張して解釈するプリフィックス命令機能拡張手段として機能する。

【0105】

3. e x t 命令(プリフィックス命令)を用いたターゲット命令の即値拡張の具体例
本発明のe x t 命令による即値拡張の代表的な実施の形態を、命令コード中に即値と汎用レジスタを指定し、該即値と汎用レジスタに格納されている値の演算を行うタイプの命令(便宜上、以下タイプ1命令という)の場合を例に取り説明する。

10

【0106】

まず、タイプ1命令及びe x t 命令の命令コードの構成について説明する。図7(A)は、該タイプ1命令の命令コード210のビットフィールドを示した図であり、図7(B)は、e x t 命令の命令コード220のビットフィールドを示している。ビットフィールドの上の数字はビットの位置を示しており、同図(A)(B)に示すように、命令コードはビット15からビット0まで16ビット幅のフィールドを有している。

【0107】

同図(A)に示すように、タイプ1命令の命令コード210は、ビット15からビット13に3ビットのクラス指定領域212と、ビット12からビット10に3ビットのオペコードを指定する領域214と、ビット9からビット4に6ビットの即値を指定する領域216と、ビット3からビット0に4ビットのレジスタ指定領域218を有している。

20

【0108】

クラス指定領域212とは、命令のグループを定義するフィールドである。本マイクロコンピュータで使用する命令は、所定のルールに従って、グループ分けされているため、どのグループに属する命令であるのかがクラス指定領域に指定されている。

【0109】

オペコードを指定する領域214には、命令の操作機能を規定するオペコードが格納されている。なお、クラス毎にそのクラスに属するオペコードが決まっているので、該オペコードを指定する領域214には、前記クラス指定領域212で指定されたクラスに属するオペコードが格納されている。

30

【0110】

また、前記即値を指定する領域216には、6ビットの即値(i m m 6)の値が格納されており、レジスタ指定領域218には、いずれかの汎用レジスタ(r d)を示すコードが格納されている。

【0111】

タイプ1命令は、前記即値(i m m 6)と汎用レジスタ(r d)に対して前記オペコードで示される演算を行い、結果を汎用レジスタ(r d)に対して書き込む動作を行う命令である。

【0112】

また、同図(B)に示すように、e x t 命令の命令コード220は、ビット15からビット13に3ビットのクラス指定領域222と、ビット12からビット0に13ビットの即値を指定する領域224とを有している。前記即値を指定する領域224には、13ビットの即値(i m m 1 3)の値が格納されている。

40

【0113】

該e x t 命令はオペコードを有していないが、本実施の形態のマイクロコンピュータが処理するプリフィックス命令は、e x t 命令1種類のみであり、所定のクラスを該プリフィックス命令に割り当てているため、クラス指定領域222に指定されたクラスによって、e x t 命令であると判別することが出来る。

【0114】

50

e x t 命令はプリフィックス命令であるため、単独ではCPUにおけるALUでの演算等の実行を何ら行わないが、後続のターゲット命令が実行される際に、そのターゲット命令の実行に使用する即値を拡張する機能を有する。例えばタイプ1命令のように命令コードに即値を含んでいる命令がe x t 命令のターゲット命令となった場合、タイプ1命令実行時に、タイプ1命令の命令コードに含まれる即値をe x t 命令の6ビットの即値(i m m 6)を用いて拡張する機能を有する。

【0115】

次に前記タイプ1命令が実行される際の即値の拡張について詳しく説明する。

【0116】

まず、タイプ1命令が単独で実行された場合の実行内容について説明する。図2において、まず、タイプ1命令はROM110から命令データバス120を介して命令レジスタ150に入力される。そして、命令でコード回路160で解読され、内容に応じた演算がおこなわれる。タイプ1命令においては、命令コードで指定された汎用レジスタ(r d)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、命令コードで指定された即値(i m m 6)は、命令デコード回路により切り出され、即値生成用信号線176を介して即値生成回路170に入力される。そして、即値生成回路170で、前記即値(i m m 6)は32ビットにゼロ拡張あるいはサイン拡張され、前記拡張された即値はALU190に入力される。そしてALU190はタイプ1命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(r d)に格納する。

【0117】

前記タイプ1命令は単独で実行することも出来るし、直前の1又は複数のe x t 命令と組み合わせることも可能である。e x t 命令と組み合わせる場合の実行内容についても、基本的には前記単独実行の場合と同様であるが、命令デコード回路により切り出された即値(i m m 6)から拡張された即値が生成される過程が異なってくる。すなわち、e x t 命令と組み合わせる場合は、E X T 1レジスタ172やE X T 2レジスタ174に保持されている先のe x t 命令の即値を用いてi m m 6の拡張が行われる。この過程および、この過程に先だってe x t 命令の即値がE X T 1レジスタ172やE X T 2レジスタ174に保持される過程は、図3～図6で説明した通りである。従って、タイプ1命令が単独で実行されたか、直前の1又は複数のe x t 命令と組み合わせる場合により、即値生成回路170で、生成される拡張された即値が異なってくる。

【0118】

図8(A)～(C)は、前記各場合のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

【0119】

図8(A)は、タイプ1命令が単独で実行された場合のオペレーションを表した式と、実行に使用される拡張された即値230のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(i m m 6)がゼロ拡張又はサイン拡張のいずれかの方法で拡張されて32ビットの即値230となる。ゼロ拡張された場合は、ビット6からビット31の領域232はすべてゼロとなり、サイン拡張された場合は、ビット6からビット31の領域232は、すべてi m m 6の最上位ビットすなわちビット5と同じビットとなる。

【0120】

図8(B)は、直前の1のe x t 命令と組み合わせる場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値240のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(i m m 6)がビット5からビット0のフィールド246にセットされ、直前の1のe x t 命令の13ビットの即値(i m m 13)がビット18からビット6にセットされ、19ビットの即値(i m m 19)が生成される。そして、前記19ビットの即値(i m m 19)はゼロ拡張又はサイ

10

20

30

40

50

ン拡張のいずれかの拡張方法で32ビットの即値240となる。ゼロ拡張された場合は、19ビットから31ビットの領域242はすべてゼロとなり、サイン拡張された場合は、19ビットから31ビットの領域242は、すべてimm19の最上位ビットすなわち18ビット目と同じビットとなる。

【0121】

図8(C)は、2回のext命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値250のビットフィールド図である。同図に示すように、タイプ1命令の6ビットの即値(imm6)がビット5からビット0のフィールド256にセットされ、1回目のext命令の13ビットの即値(imm13)がビット31からビット19にセットされ、2回目のext命令の13ビットの即値(imm13)がビット18からビット6にセットされ、32ビットの即値(imm32)250が生成される。

10

【0122】

なお、2回以上のext命令と組み合わせて実行場合は、最初と、最後のext命令が有効となるよう構成されている。すなわち、最初のext命令が前記1回目のext命令と同様に機能し、最後のext命令が前記2回目のext命令と同様に機能する。

【0123】

4.プリフィックス命令を用いたターゲット命令のオペレーション内容拡張の具体例
本発明のプリフィックス命令によるターゲット命令のオペレーション内容拡張の代表的な実施の形態を、2つ汎用レジスタを用いて演算を行うタイプの命令(便宜上、以下タイプ2命令という)の場合を例に取り説明する。

20

【0124】

まず、タイプ2命令の命令コードの構成について説明する。図9は、該タイプ2命令の命令コード260のビットフィールドを示した図である。

【0125】

同図に示すように、タイプ2命令の命令コード260は、ビット15からビット13に3ビットのクラス指定領域262と、ビット12からビット8に5ビットのオペコードを指定する領域264と、ビット7からビット4に4ビットのレジスタ指定領域266と、ビット3からビット0に4ビットのレジスタ指定領域268とを有している。

【0126】

また、前記レジスタ指定領域266、268には、それぞれいずれかの汎用レジスタ(rs)(rd)を示すコードが格納されている。rsとはレジスタソース、rdとはレジスタディスティネーションを意味するものある。タイプ2命令は、汎用レジスタ(rs)に格納されているデータと汎用レジスタ(rd)に格納されているデータにたいしてオペコードで示される演算を行い、結果を汎用レジスタ(rd)に書き込む動作を行う2オペランド命令である。

30

【0127】

次に、タイプ2命令が単独で実行された場合のCPUの動作について説明する。図2において、まず、タイプ2命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読され、内容に応じた演算がおこなわれる。タイプ2命令においては、命令コードで指定された汎用レジスタ(rd)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、命令コードで指定された汎用レジスタ(rs)に格納されたデータがレジスタファイル90よりデータバス184を介して、ALU190に入力される。そしてALU190はタイプ2命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(rd)に格納する。

40

【0128】

前記タイプ2命令は単独で実行することも出来るし、直前の1又は複数のext命令と組み合わせて実行することも可能である。

【0129】

50

図10(A)~(C)は、タイプ2命令を単独で実行する場合及び直前の1又は2回のe x t命令と組み合わせて実行する場合のオペレーションを表した式と、実行に使用する即値のビットフィールドを示した図である。

【0130】

図10(A)は、タイプ2命令が単独で実行された場合のオペレーションを表した式である。単独で実行される場合は、同式に示すように、タイプ2命令の本来のオペレーションが実行されるため、即値は使用されない。なお、タイプ2命令が単独で実行された場合の実行内容については前述した通りである。

【0131】

図10(B)は、タイプ2命令が直前の1のe x t命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値270のビットフィールド図である。同式に示すように、タイプ2命令が直前の1のe x t命令と組み合わせて実行された場合は、e x t命令に基づき生成される即値(i m m 1 3)と、汎用レジスタ(r s)の内容との演算結果を、汎用レジスタ(r d)に書き込むようにオペレーション内容が拡張する。即値270は、e x t命令に含まれる即値(i m m 1 3)が32ビットにゼロ拡張して生成される。

10

【0132】

タイプ2命令が1のe x t命令と組み合わせて実行された場合の実行内容について説明する。図2において、まず、タイプ2命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合(前記ステートマシンの状態がS0以外の場合)は、以下のオペレーションを実行する。すなわち、命令コードで指定された汎用レジスタ(r s)に格納されたデータがレジスタファイル90よりデータバス182を介して、ALU190に入力される。また、E X T 1レジスタ172に保持されていた直前のe x t命令の即値(i m m 1 3)は、即値生成回路170によりゼロ拡張されて前記即値270が生成され、ALU190に入力される。そして、ALU190はタイプ2命令のオペコードで示される演算を行い、演算結果をデータバス192を介して、レジスタファイル90の汎用レジスタ(r d)に格納する。

20

【0133】

なお、E X T 1レジスタ172やE X T 2レジスタ174に保持されている先のe x t命令の即値を用いて、タイプ2命令の実行に使用する即値を生成す過程および、この過程に先だつてe x t命令の即値がE X T 1レジスタ172やE X T 2レジスタ174に保持される過程は、図3~図6で説明した通りである。

30

【0134】

従つて、タイプ2命令をe x t命令と組み合わせて実行した場合、オペランドに即値と2つの汎用レジスタとを有するような3オペランド命令を実行したのと同様の効果を有する。

【0135】

図10(C)は、タイプ2命令が2回のe x t命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値280のビットフィールド図である。同式に示すように、タイプ2命令が2回のe x t命令と組み合わせて実行された場合は、2回のe x t命令に基づき生成される即値(i m m 2 6)と、汎用レジスタ(r s)の内容との演算結果を、汎用レジスタ(r d)に書き込むようにオペレーション内容が拡張する。即値280は、1回目のe x t命令に含まれている即値i m m 1 3をビット25からビット13にセットし、2回目のe x t命令に含まれている即値をビット12からビット0にセットしたi m m 2 6を32ビットにゼロ拡張又はサイン拡張して生成される。

40

【0136】

なお、2回以上のe x t命令と組み合わせて実行された場合は、最初と、最後のe x t命令が有効となるよう構成されている。すなわち、最初のe x t命令が前記1回目のe x t

50

命令と同様に機能し、最後の e x t 命令が前記 2 回目の e x t 命令と同様に機能する。

【 0 1 3 7 】

タイプ 2 命令が 2 回または複数回の e x t 命令と組み合わせて実行された場合の実行内容は、基本的には、1 回の e x t 命令と組み合わせて実行された場合と同様であるが、即値生成回路 1 7 0 が前記即値 2 8 0 を生成する過程がことになって来る。

【 0 1 3 8 】

すなわち、即値生成回路 1 7 0 は、E X T 1 レジスタ 1 7 2 に保持されていた 1 回目の e x t 命令の即値 (i m m 1 3) と、E X T 2 レジスタ 1 7 4 に保持されていた最後の e x t 命令の即値 (i m m 1 3) を用いて前記即値 2 8 0 を生成する。

【 0 1 3 9 】

なお、E X T 1 レジスタ 1 7 2 や E X T 2 レジスタ 1 7 4 に保持されている先の e x t 命令の即値を用いて、タイプ 2 命令の実行に使用する即値を生成す過程および、この過程に先だつて e x t 命令の即値が E X T 1 レジスタ 1 7 2 や E X T 2 レジスタ 1 7 4 に保持される過程は、図 3 ~ 図 6 で説明した通りである。

【 0 1 4 0 】

この様に、e x t 命令を用いることにより、ターゲット命令のオペレーション内容を拡張することができるため、命令デコード回路及び即値生成回路は、ターゲット命令機能拡張手段として機能する。

【 0 1 4 1 】

また、タイプ 2 命令を 2 回又は複数回の e x t 命令と組み合わせて実行した場合、2 つの e x t 命令の 1 3 ビットの即値 (i m m 1 3) つなげたものをゼロ拡張又はサイン拡張した即値を使用できるため、1 つの e x t 命令と組み合わせた場合よりも大きな有効ビット数を有する即値を処理することが出来る。

【 0 1 4 2 】

この様に、タイプ 2 命令を 2 回又は複数回の e x t 命令と組み合わせて実行した場合、1 回の e x t 命令と組み合わせて実行した場合にくらべて機能が拡張するため、命令デコード回路及び即値生成回路は、プリフィックス命令機能拡張手段として機能する。

【 0 1 4 3 】

5 . プリフィックス命令を用いたターゲット命令のアドレスの即値拡張の具体例次に本発明のプリフィックス命令によるターゲット命令実行時のアドレスの拡張について説明する。図 1 1 は、前記マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。基本的には、図 2 に示す構成と同様であり、図 2 と同様の機能を有する部分については、同符号を付している。なお、簡単のため、図 2 の E X T 1 レジスタ 1 7 2 等は省略して、命令デコード回路 1 6 0 と即値生成回路 1 7 0 のみ図示している。また、図 2 の構成に加えて、R A M 3 2 0 、プログラムカウンタ P C 3 3 0 、アドレス演算器 3 4 0 、Xバス 3 5 0 を図示している。なお、説明の都合上、プログラムカウンタ P C 3 3 0 をレジスタファイル 9 0 とは別に図示している。また、アドレス演算器 3 4 0 は加減乗除等の演算を行うが、本具体例では、便宜上加算演算をする機能のみを取り上げ説明する。

【 0 1 4 4 】

前記 R A M 3 2 0 は、実行に使用するデータ等を記憶している。アドレス演算器 3 4 0 は、前記 R A M 3 2 0 に記憶されているデータのアドレスを演算するものであり、即値生成回路 1 7 0 で生成された即値及びプログラムカウンタ P C 3 3 0 の値もしくはレジスタファイル 9 0 に格納された値を入力するよう構成されている。プログラムカウンタ P C 3 3 0 には、現在実行中の命令のアドレスが格納されている。そして、前記アドレス演算器 3 4 0 で演算されたアドレスによりデータアドレスバス 3 1 0 を介して、R A M 3 2 0 のアドレスを指定し、又は、命令アドレスバス 1 2 0 を介して、R O M 1 1 0 の命令のアドレスを指定する。そして、前記各アドレスバス 3 1 0 、1 2 0 で指定されたアドレスのデータ又は命令コードは、データバス 1 4 0 又は命令データバス 1 3 0 を介して C P U 3 0 に入力されよう構成されている。

10

20

30

40

50

【 0 1 4 5 】

次に、オペランドにアドレスの即値を有し、プログラムカウンタ P C 3 3 0 に格納されている値に該即値を加算した相対アドレスを実行に使用するタイプの命令（便宜上、以下タイプ 3 命令という）におけるプリフィックス命令によるアドレスの即値の拡張を説明する。

【 0 1 4 6 】

図 1 2 は、タイプ 3 命令の一例である P C 相対サブルーチンコール命令（相対アドレスでサブルーチンをコールする命令）の命令コード 2 9 0 のビットフィールドを示した図である。

【 0 1 4 7 】

同図に示すように、タイプ 3 命令の命令コード 2 9 0 は、ビット 1 5 からビット 1 3 に 3 ビットのクラス指定領域 2 9 2 と、ビット 1 2 からビット 9 に 4 ビットのおペコードを指定する領域 2 9 4 と、ビット 8 の所定のオペコードに付帯したビットを指定する d ビット領域 2 9 6 と、ビット 7 からビット 0 に 8 ビットの即値を指定する領域 2 9 8 を有している。

【 0 1 4 8 】

P C 相対サブルーチンコール命令は、前記即値を指定する領域 2 9 8 に格納された 8 ビットの即値 (i m m 8) を 1 ビット左にシフトすることで 9 ビットの値 (s i n 9) とし、該 s i n 9 の値をサイン拡張して得られる 3 2 ビットの即値と P C (プログラムカウンタ) の値とを加算した値をアドレスとして該アドレスに分岐する命令である。

【 0 1 4 9 】

図 1 1 をもちいて、該 P C 相対サブルーチンコール命令が単独で実行される場合の C P U の動作について説明する。即値生成回路 1 7 0 によって、該命令コード含まれる 8 ビットの即値は、1 ビット左にシフトされ、サイン拡張され 3 2 ビットの即値 (s i g n 9) となる。そして、該 3 2 ビットの即値 (s i g n 9) はアドレス演算器 3 4 0 に入力される。また、プログラムカウンタ P C 3 3 0 に格納されているアドレスの値もアドレス演算器 3 4 0 に入力される。アドレス演算器 3 4 0 は、前記即値 (s i g n 9) とプログラムカウンタ P C 3 3 0 に格納されているアドレスの値を加算して分岐アドレスを生成する。そして、分岐アドレスに格納されたサブルーチンが実行されることになる。

【 0 1 5 0 】

P C 相対サブルーチンコール命令のようなタイプ 3 命令は、単独で実行することも出来るし、直前の 1 又は複数の e x t 命令と組み合わせて実行することも可能である。e x t 命令と組み合わせて実行された場合の実行内容についても、基本的には前記単独実行の場合と同様であるが、命令デコード回路 1 6 0 の即値生成回路 1 7 0 で、命令コードより切り出された即値 (i m m 8) から拡張された即値が生成される過程が異なってくる。すなわち、e x t 命令と組み合わせて実行された場合は、E X T 1 レジスタ 1 7 2 や E X T 2 レジスタ 1 7 4 に保持されている先の e x t 命令の即値を用いて i m m 8 の拡張が行われる。この過程および、この過程に先だって e x t 命令の即値が E X T 1 レジスタ 1 7 2 や E X T 2 レジスタ 1 7 4 に保持される過程は、図 3 ~ 図 6 で説明した通りである。従って、タイプ 3 命令が単独で実行されたか、直前の 1 又は複数の e x t 命令と組み合わせて実行されたかにより、分岐先のアドレスをかえることが出来る。

【 0 1 5 1 】

図 1 3 (A) ~ (C) は、前記各場合のタイプ 3 命令のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

【 0 1 5 2 】

図 1 3 (A) は、タイプ 3 命令が単独で実行された場合のオペレーションを表した式と、実行に使用される拡張された即値 4 0 0 のビットフィールド図である。C P U 3 0 は、1 6 ビット固定長の命令コードを用いているため、命令アドレスの値は常に偶数バイトアドレスとなり、その結果、命令アドレスの最下位ビットは常に " 0 " となる。そのため、拡張の際には、同図に示すように、タイプ 3 命令の 8 ビットの即値 (i m m 8) は 1 ビット

10

20

30

40

50

左にシフトた9ビットの即値となる。そして該9ビットの即値はサイン拡張されて32ビットの即値400 (s i g n 9) となる。

【 0 1 5 3 】

図13 (B) は、タイプ3命令が直前の1の e x t 命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値410のビットフィールド図である。同図に示すように、タイプ3命令の8ビットの即値 (i m m 8) と e x t 命令の13ビットの即値 (i m m 1 3) を結合した21ビットの即値は、1ビット左にシフトされて、22ビットの即値となる。該22ビットの即値は、サイン拡張されて32ビットの即値410 (s i g n 2 2) となる。

【 0 1 5 4 】

図13 (C) は、タイプ3命令が2回の e x t 命令と組み合わせて実行された場合のオペレーションを表した式と、該オペレーションの実行に使用される拡張された即値420のビットフィールド図である。同図に示すように、タイプ3命令の8ビットの即値 (i m m 8) と1回目の e x t 命令の10ビットの即値 (i m m 1 0) と、2回目の e x t 命令の13ビットの即値 (i m m 1 3) とを結合した31ビットの即値は、1ビット左にシフトされて、32ビットの即値 (s i g n 3 2) となる。

【 0 1 5 5 】

なお、2回以上の e x t 命令と組み合わせて実行場合は、最初と、最後の e x t 命令が有効となるよう構成されている。すなわち、最初の e x t 命令が前記1回目の e x t 命令と同様に機能し、最後の e x t 命令が前記2回目の e x t 命令と同様に機能する。

【 0 1 5 6 】

6 . プリフィックス命令を用いてターゲット命令の命令中にディスプレースメントを付加する具体例

次に、オペランドには即値を有さず、オペランドに指定したレジスタに格納された値をアドレスとして用いて実行を行うタイプの命令 (便宜上、以下タイプ4命令という) において、プリフィックス命令を用いてアドレスのディスプレースメントを付加する例を説明する。

【 0 1 5 7 】

命令コードの構成は図9に示すタイプ2命令と、2個のレジスタの指定領域をオペランドに有する点で共通する。オペコードの内容が異なってくるが、本実施の形態の説明において特に影響は無いので説明を省略する。タイプ4命令の一例であるロード命令を例にとり、説明する。ロード命令は、 r s で指定されるアドレスに記憶されているデータを r d に示すレジスタにロードするための命令である。

【 0 1 5 8 】

まず、該ロード命令が単独で実行される場合の C P U の動作について説明する。 C P U は、命令コードで指定された汎用レジスタ (r s) に格納された値をアドレスとするデータを R A M 3 2 0 から読み出し、命令コードで指定された汎用レジスタ (r d) に格納する。

【 0 1 5 9 】

ロード命令のようなタイプ4命令は、単独で実行することも出来るし、直前の1又は複数の e x t 命令と組み合わせて実行することも可能である。 e x t 命令と組み合わせて実行された場合は、 e x t 命令の命令コードに含まれる即値によって、ディスプレースメントが付加されて、タイプ4命令が実行される。

【 0 1 6 0 】

図14 (A) (B) は、タイプ4命令を直前の1又は2回の e x t 命令と組み合わせて実行する場合に生成されるディスプレースメントを示した図である。

【 0 1 6 1 】

図14 (A) は、タイプ4命令が1回の e x t 命令とくみあわされて実行された場合のディスプレースメントを表したている。同図に示すように、直前の1の e x t 命令の即値 (i m m 1 3) をゼロ拡張して32ビット即値 (i m m 1 3) 4 3 0 が生成される。

10

20

30

40

50

【0162】

タイプ4命令が1のe x t命令と組み合わせて実行された場合の実行内容について図11を用いて説明する。まず、タイプ4命令はROM110から命令データバス130を介して命令レジスタ150に入力される。そして、命令デコード回路160で解読されるが、このとき該命令がターゲット命令になっている場合（前記ステートマシンの状態がS0以外の場合）は、命令コードで指定された汎用レジスタ（r s）に格納されたアドレスをレジスタファイル90から取り出してXバス350を介して、アドレス演算器340に入力する。

【0163】

また、E X T 1レジスタ172に保持されていた直前のe x t命令の即値（i m m 1 3）は、即値生成回路170によりゼロ拡張されて32ビットの即値430が生成され、アドレス演算器340に入力される。アドレス演算器340はこれらの2つの入力を加算してアドレスを生成する。生成されたアドレスによりデータアドレスバス310を介して、RAM320のアドレスを指定し、該アドレスに格納されたデータをデータバス140を介してレジスタファイル90に入力し、命令コードで指定された汎用レジスタ（r d）に格納する。

10

【0164】

図14（B）は、タイプ4命令が2回のe x t命令と組あわされて実行された場合のディスプレイメントを表している。同図に示すように、1回目のe x t命令の即値（i m m 1 3）をビット25からビット13の13ビットのフィールド444にセットし、2回めのe x t命令の即値（i m m 1 3）をビット12からビット0の13ビットのフィールド446にセットし、ゼロ拡張して32ビットの即値440（i m m 2 6）が生成される。

20

【0165】

タイプ4命令が2回のe x t命令と組み合わせて実行された場合の実行内容について図11を用いて説明する。この場合も1回のe x t命令と組み合わせた場合と基本的には同様であるが、ディスプレイメントとして生成される即値が異なってくる。2回のe x t命令と組み合わせた場合は、E X T 1レジスタ172及びE X T 2レジスタ174に保持されていた2回のe x t命令の即値を用いて、図14（B）に示すような即値440が生成され、該即値440がディスプレイメントとしてアドレス演算器340に入力される。

【0166】

なお、E X T 1レジスタ172やE X T 2レジスタ174に保持されている先のe x t命令の即値を用いて、タイプ4命令の実行に使用する即値を生成す過程および、この過程に先だってe x t命令の即値がE X T 1レジスタ172やE X T 2レジスタ174に保持される過程は、図3～図6で説明した通りである。

30

【0167】

従って、タイプ4命令をe x t命令と組み合わせて実行した場合、オペランドにディスプレイメントを付加して実行するようにオペレーション内容が拡張する。

【0168】

（実施の形態2）

図15は、本実施の形態のマイクロコンピュータのハードウェアブロック図である。

40

【0169】

該マイクロコンピュータ10は、32ビットマイクロコントローラであり、CPU30とROM110とRAM320、高周波発信回路710、低周波発信回路720、リセット回路730、プリスケラ740、16ビットプログラマブルタイマ750、8ビットプログラマブルタイマ760、クロックタイマ770、インテリジェントDMA780、高速DMA790、割り込みコントローラ600、シリアルインターフェース610、バスコントロールユニット（BCU）620、A/D変換器630、D/A変換器640、入力ポート660、出力ポート660、I/Oポート670、及びそれらを接続する各種バス680等、各種ピン690等を含む。

【0170】

50

前記CPU30は、16ビット固定長命令コードを処理し、実行時のデータサイズは32ビットのCPUである。該CPU30は、前述した、実施の形態1の構成を有しており、命令コード解析手段、即値拡張手段、ターゲット命令機能拡張手段、プリフィックス命令機能拡張手段、命令実行手段として機能する。

【0171】

従って、16ビットの命令コードを使用しても、大きな即値の取り扱いや、短い命令コードで記述出来ない3オペランド命令の実行を行うことが出来るため、メモリの使用効率のよいマイクロコンピュータを提供することが出来る。また、該マイクロコンピュータを半導体集積回路として構成する場合、記憶させる命令数が同一であれば、32ビット幅の固定長命令コードを使用する場合に比べ、容量は半分でよい。従って、チップのサイズを小さくすることができ歩留まりのよい半導体集積回路装置を製造することが出来る。

10

【0172】

本発明のマイクロコンピュータは例えばプリンター等のパソコン周辺機器や、携帯機器等の各種の電子機器に適用可能である。この様にすると、簡単な構成でメモリの使用効率のよい情報処理回路を内蔵することができるため、安価で高機能な電子機器を提供することが出来る。

【0173】

なお本発明は、上記実施例で説明したものに限らず、種々の変形実施が可能である。

【0174】

例えば上記実施例では、メモリの使用効率を高めるうえで特に有効な例、すなわち32ビットのデータが処理可能なCPU又はマイクロコンピュータにおいて、16ビットの固定長命令コードを使用する場合を例に取り説明したがこれに限られない。CPU又はマイクロコンピュータで処理可能なビット数及び命令コードのビット数によらず本発明を適用することが出来る。また、命令コードが固定長であるか否かによらず本発明を適用することが出来る。

20

【0175】

そして、本発明を適用することにより、命令コードの即値の拡張及び、オペレーション内容の変更を簡単な構成で容易に行えるCPU又はマイクロコンピュータを提供することが出来る。

【0176】

また、上記本実施の形態では、ext命令を2回以上連続実行した場合は1回目と最後のext命令のみ有効に機能する場合を説明したが、3回以上のext命令を有効にするよう構成してもよい。また、有効にするext命令の選択は1回目と最後に限られず所定のルールに従ってさだめればよい。

30

【0177】

また、複数のext命令を組み合わせるによりext命令の機能を変化させる例も上記実施例に限られない。

【0178】

また、上記本実施の形態では、プリフィックス命令がext命令1種類のみである場合を説明したが、複数の異なる種類のプリフィックス命令を設定してもよい。

40

【0179】

【図面の簡単な説明】

【図1】本発明の実施の形態のCPUの機能ブロック図である。

【図2】マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。

【図3】命令デコード回路が有するステートマシンの状態の遷移を表した図である。

【図4】即値の保持に関して命令デコード回路が行う動作のアルゴリズムを示したフローチャート図である。

【図5】入力される命令コード及び前記ステートマシンの状態と前記即値生成用信号生成回路が生成する即値生成用信号との関係を示したタイミングチャート図である。

50

【図6】即値生成回路が、ステートマシンの状態及び即値生成用信号に基づき即値を拡張するアルゴリズムを示したフローチャート図である。

【図7】同図(A)(B)は、タイプ1命令及びext命令の命令コードのビットフィールドを示した図である。

【図8】同図(A)~(C)は、オペレーションを表した式及びオペレーションの実行に使用される拡張された即値のフィールド図である。

【図9】タイプ2命令の命令コードのビットフィールドを示した図である。

【図10】同図(A)~(C)は、タイプ2命令のオペレーションを表した式と、実行に使用する即値のビットフィールドを示した図である。

【図11】マイクロコンピュータに内蔵されている半導体集積回路の構成の説明に必要な部分を図示したものである。 10

【図12】タイプ3命令の命令コードのビットフィールドを示した図である。

【図13】同図(A)~(C)は、タイプ3命令のオペレーションを表した式と該オペレーションの実行に使用される拡張された即値のフィールド図である。

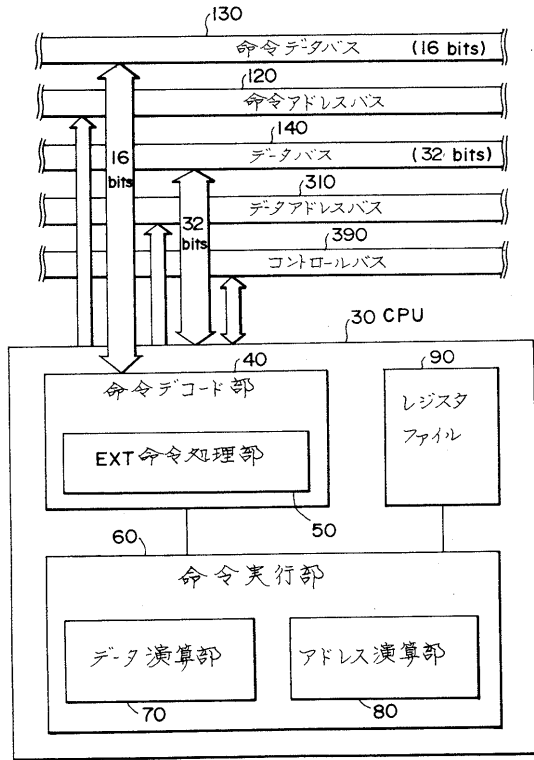
【図14】同図(A)(B)は、タイプ4命令を実行する場合に生成されるディスプレイメントを示した図である。

【図15】実施の形態2のマイクロコンピュータのハードウェアブロック図である。

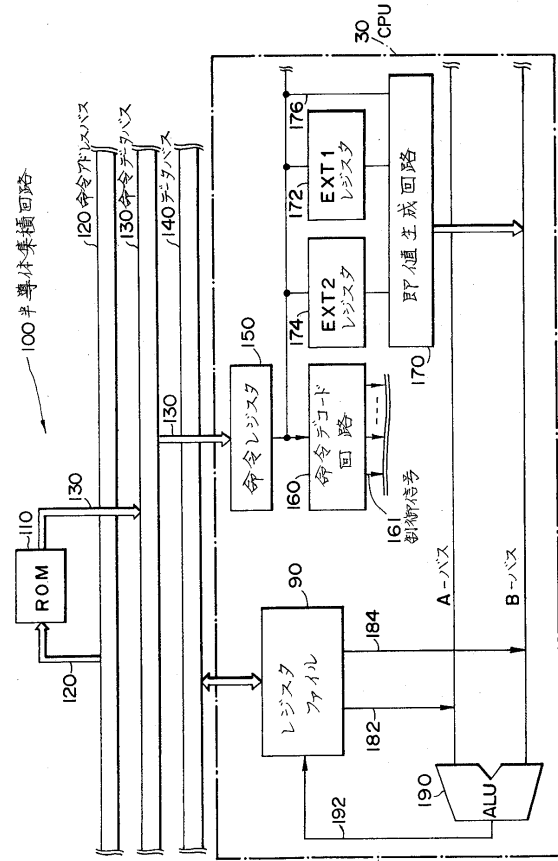
【符号の説明】

10	マイクロコンピュータ	
30	CPU	20
40	命令デコード部	
50	ext命令処理部	
60	命令実行部	
70	データ演算部	
80	アドレス演算部	
90	レジスタファイル	
110	ROM	
120	命令アドレスバス	
130	命令データバス	
140	データバス	30
150	命令レジスタ	
160	命令デコード回路	
161	制御信号	
170	即値生成回路	
172	EXT1レジスタ	
174	EXT2レジスタ	
190	ALU	
310	データアドレスバス	
320	RAM	
330	プログラムカウンタPC	40
340	アドレス演算器	

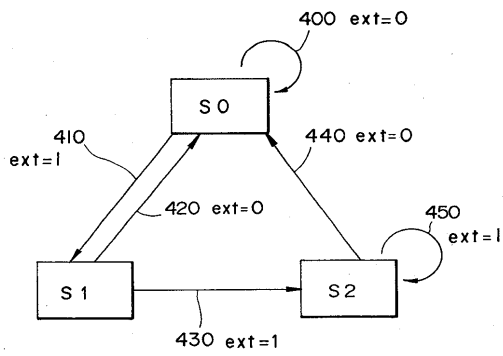
【図1】



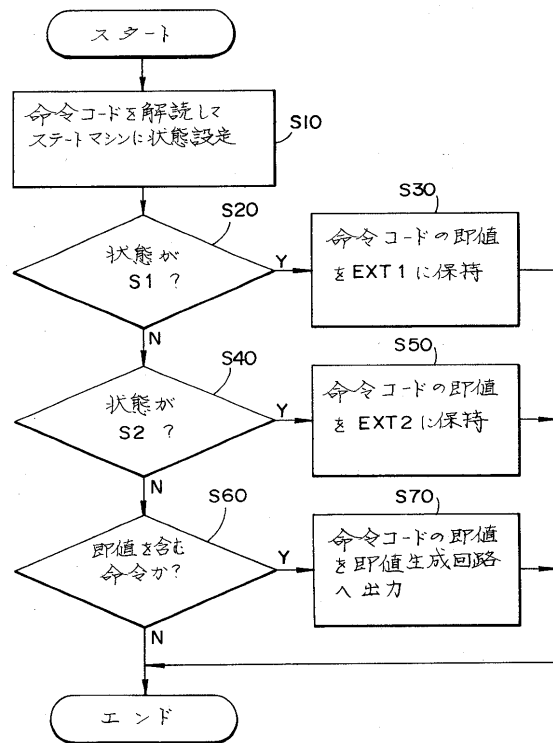
【図2】



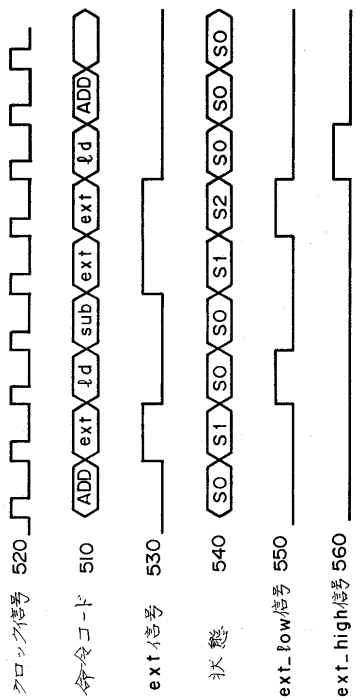
【図3】



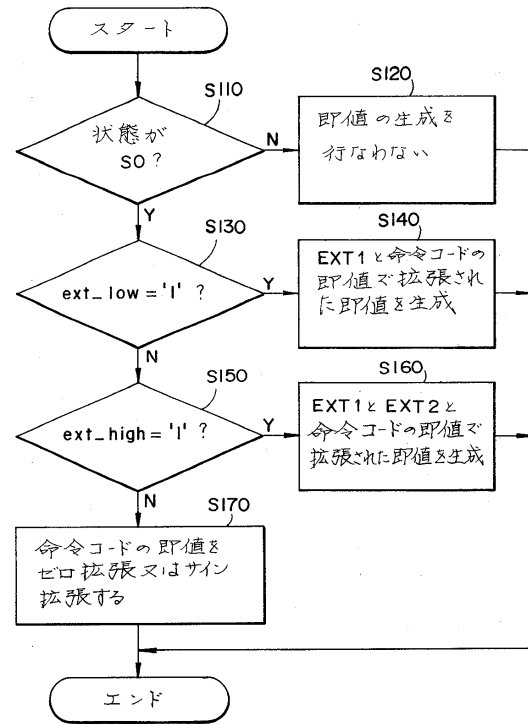
【図4】



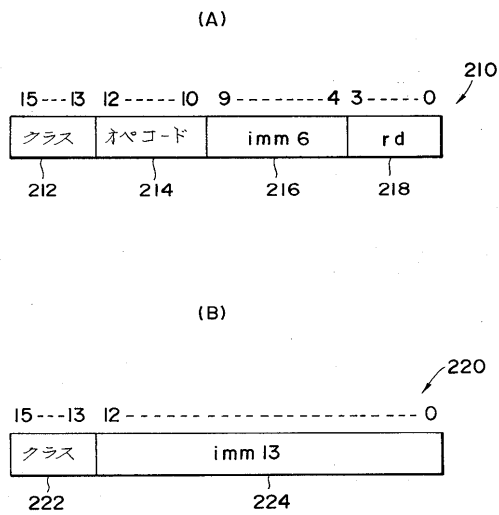
【 図 5 】



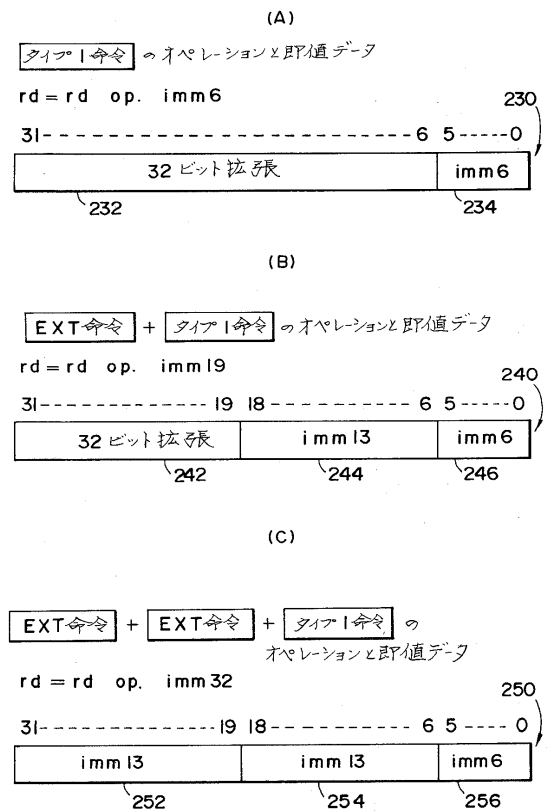
【 図 6 】



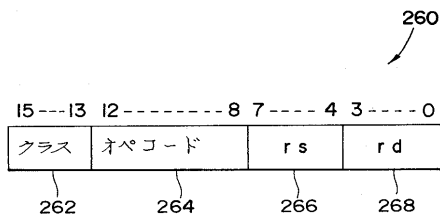
【 図 7 】



【 図 8 】



【 図 9 】



【 図 10 】

(A)

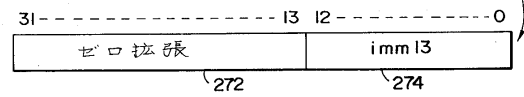
タイプ 2 命令 の オペレーション

rd = rs op. rd

(B)

EXT 命令 + タイプ 2 命令 の オペレーション と 即値 データ

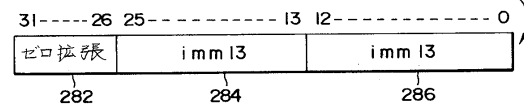
rd = rs op. imm13



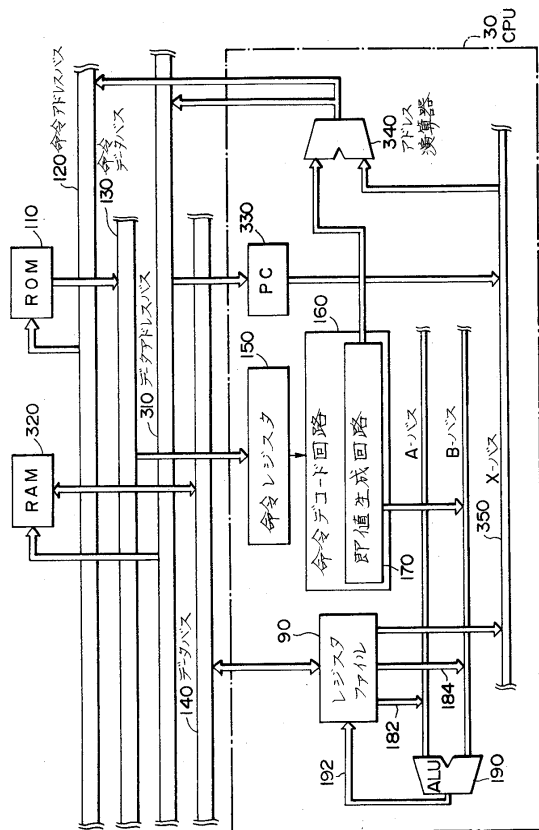
(C)

EXT 命令 + EXT 命令 + タイプ 2 命令 の
オペレーション と 即値 データ

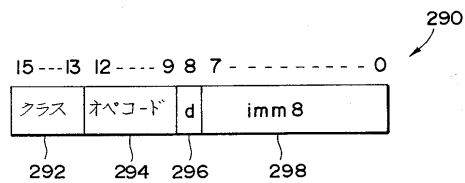
rd = rs op. imm26



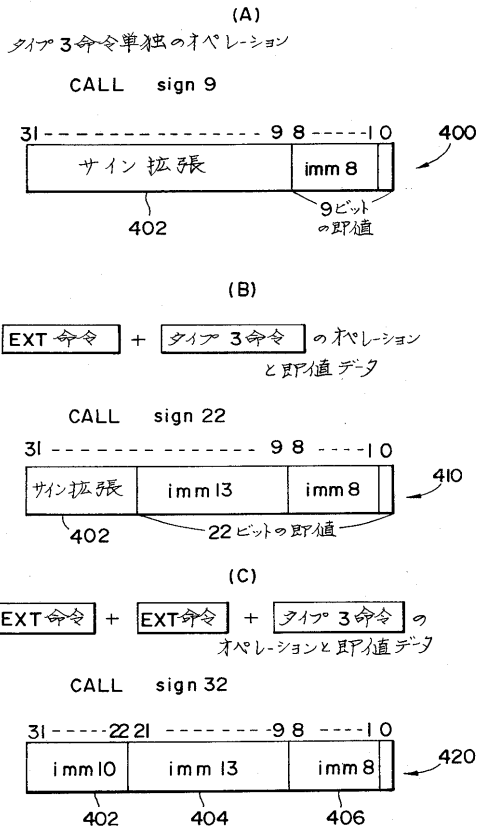
【 図 11 】



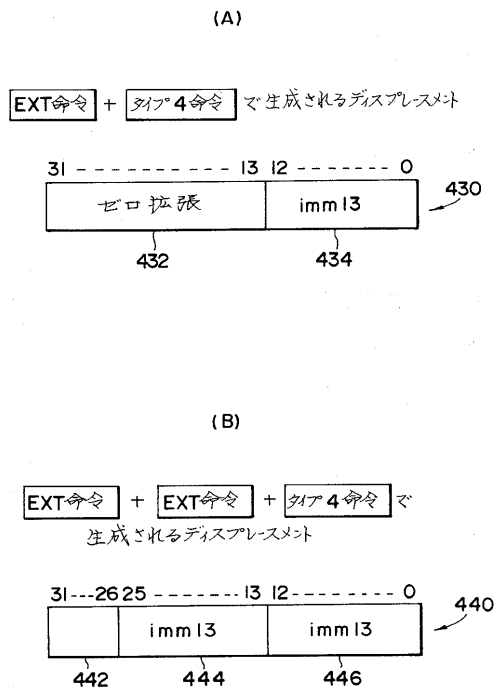
【 図 12 】



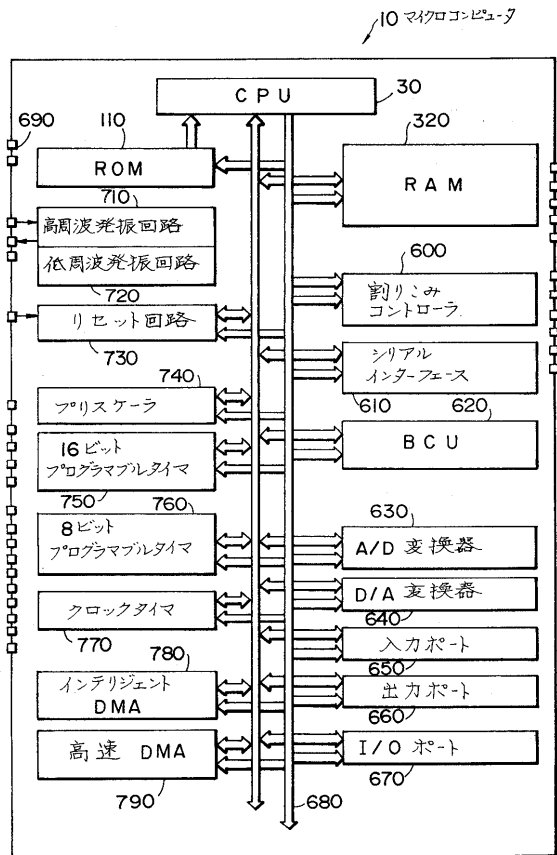
【 図 1 3 】



【 図 1 4 】



【 図 1 5 】



フロントページの続き

(72)発明者 宮山 芳幸

長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

審査官 酒井 恭信

(56)参考文献 特開平03-188526(JP,A)

特開平03-201133(JP,A)

特開平04-172533(JP,A)

特開昭62-003338(JP,A)

特開平10-021071(JP,A)

(58)調査した分野(Int.Cl.⁷, DB名)

G06F 9/30