(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3727395号

(P3727395)

(45)発行日　平成17年12月14日(2005.12.14)　(24)登録日　平成17年10月7日(2005.10.7)

(51)Int.Cl.[7]	F I		
G06F 13/16	G06F	13/16	510D
G06F 9/34	G06F	9/34	350B
G06F 9/38	G06F	9/38	370X
G06F 12/06	G06F	12/06	525A
G06F 15/78	G06F	15/78	510G

請求項の数 17　(全 38 頁)

(21)出願番号　特願平7-347441
(22)出願日　平成7年12月14日(1995.12.14)
(65)公開番号　特開平9-22379
(43)公開日　平成9年1月21日(1997.1.21)
審査請求日　平成14年9月5日(2002.9.5)
(31)優先権主張番号　特願平7-132906
(32)優先日　平成7年5月2日(1995.5.2)
(33)優先権主張国　日本国(JP)

(73)特許権者　503121103
株式会社ルネサステクノロジ
東京都千代田区丸の内二丁目４番１号
(74)代理人　100089071
弁理士　玉村　静世
(72)発明者　大須賀　宏
東京都小平市上水本町５丁目２０番１号
株式会社日立製作所　半導体事業部内
(72)発明者　木内　淳
東京都小平市上水本町５丁目２０番１号
株式会社日立製作所　半導体事業部内
(72)発明者　長谷川　博宣
東京都小平市上水本町５丁目２０番１号
株式会社日立製作所　半導体事業部内

最終頁に続く

(54)【発明の名称】マイクロコンピュータ

(57)【特許請求の範囲】

【請求項１】

ＣＰＵと、

上記ＣＰＵに従って動作し、乗算器を含むＤＳＰと、

上記ＣＰＵから選択的にアドレスが供給される第１乃至第３アドレスバスと、

上記ＣＰＵ及び上記ＤＳＰに接続された第１データバスと、

上記ＤＳＰに接続された第２及び第３データバスと、

上記第１及び第２アドレスバスと上記第１及び第２データバスとに接続され、上記ＣＰＵから供給されるアドレスによってアクセスされる第１メモリと、

上記第１及び第３アドレスバスと上記第１及び第３データバスとに接続され、上記ＣＰＵから供給されるアドレスによってアクセスされる第２メモリと、

上記ＣＰＵから上記ＤＳＰに対し、ＤＳＰ動作を制御するための第１制御信号を供給するための制御信号線と、

上記第１アドレスバス及び上記第１データバスに接続されたインターフェース回路とを有し、

上記ＣＰＵは第１アドレスバスを介して第１アドレスを出力し、上記第１データバスを介して命令を含む第１データを取り込み、

上記ＣＰＵは、複数の汎用レジスタと、算術論理演算回路と、上記第１アドレスバスに接続され、上記第１アドレスを格納するためのプログラムカウンタと、上記第１データバスに接続され、上記第１データバスから供給された上記命令を格納するための命令レジス

タと、上記命令レジスタに格納された命令をデコードし、デコード結果に従う制御信号を発生するための命令デコーダとを含み、上記制御信号に基づき、上記第１制御信号を上記制御信号線を介して出力し、
上記ＤＳＰは、上記第１制御信号に基づいて上記第１メモリまたは第２メモリの少なくとも一方から取得したデータを用いて演算処理を実行することが可能であることを特徴とするマイクロコンピュータ。
【請求項２】
請求項１において、
上記第１及び第２メモリのそれぞれは、ＲＡＭ及びＲＯＭを含むことを特徴とするマイクロコンピュータ。
【請求項３】
請求項１又は２において、
上記ＤＳＰは、
上記第１乃至第３データバスのそれぞれに対応して接続された第１乃至第３データバッファ手段と、
上記乗算器及び上記第１乃至第３データバッファ手段に接続可能な内部バスと、
上記内部バスを介して上記第１乃至第３データバッファ手段の各々に接続可能な複数のレジスタ手段と、
上記内部バスに接続可能な論理演算手段、
上記ＣＰＵから供給された上記第１制御信号をデコードし、上記データバッファ手段、乗算器、論理演算手段及びレジスタ手段を制御するための第１デコード回路とを含むことを特徴とするマイクロコンピュータ。
【請求項４】
請求項１において、
上記ＤＳＰは、上記命令デコーダから選択的に出力される上記第１制御信号を受け、上記乗算器の動作を制御するための第２制御信号を発生するデコード回路を有することを特徴とするマイクロコンピュータ。
【請求項５】
請求項１において、
上記ＤＳＰは、上記制御信号線から供給された上記第１制御信号をデコード可能なデコード部を含むことを特徴とするマイクロコンピュータ。
【請求項６】
ＣＰＵと、
上記ＣＰＵに接続された第１乃至第３アドレスバスと、
上記第１及び第２アドレスバスに接続された第１メモリと、
上記第１及び第３アドレスバスに接続された第２メモリと、
上記第１及び第２メモリとＣＰＵに接続された第１データバスと、
上記第１メモリに接続された第２データバスと、
上記第２メモリに接続された第３データバスと、
上記第１データバス及び上記第１アドレスバスに接続されたインターフェース回路と、
上記第１乃至第３データバスに接続されたＤＳＰとを有し、
上記ＣＰＵは、上記第１アドレスバスを介して第１アドレスを出力し、上記第１アドレスに応じた第１データを上記第１データバスを介して入力し、
上記第１データは上記ＣＰＵで実行可能な命令を含み、
上記ＣＰＵは、命令を解読して制御信号を生成する命令デコード回路を有し、
上記命令デコード回路は、上記第１データに含まれる命令を解読し、解読結果に応じて、上記ＤＳＰの動作を制御するための制御信号を出力することが可能であり、
上記ＤＳＰは、上記制御信号に基づいて上記第２データバスを介して上記第１メモリから第２データを読み出し、上記第３データバスを介して上記第２メモリから第３データを読み出し、上記第２データ及び第３データを用いて演算処理を実行することが可能であり

、
上記ＣＰＵから上記ＤＳＰに対して、上記制御信号を転送するための制御信号線を有することを特徴とするマイクロコンピュータ。
【請求項７】
請求項６において、
上記ＤＳＰで第２データと第３データを用いて実行される演算処理は、積和演算であることを特徴とするマイクロコンピュータ。
【請求項８】
請求項６において、
上記命令デコード回路は、上記ＣＰＵの制御を行うための制御信号を生成することが可能であることを特徴とするマイクロコンピュータ。
【請求項９】
請求項６において、
上記ＤＳＰは、命令デコーダを有し、
上記命令デコーダは、上記ＣＰＵより制御信号線を介して出力される上記制御信号を解読することが可能であり、解読結果に応じて、上記第２データバスまたは第３データバスを介して入力されるデータを用いて演算処理を実行することが可能であることを特徴とするマイクロコンピュータ。
【請求項１０】
第１乃至第３アドレスバスと、
データを伝達することが可能な第１乃至第３データバスと、
上記第１及び第２アドレスバスと上記第１及び第２データバスに接続された第１メモリと、
上記第１及び第３アドレスバスと上記第１及び第３データバスに接続された第２メモリと、
上記第１乃至第３アドレスバスへ第１乃至第３アドレス信号をそれぞれ供給する中央処理部と、
上記第２及び第３アドレス信号によるアクセスに応答して上記第１及び上記第２メモリから出力された第１及び第２データを上記第２及び第３データバスを介して、１つのバスサイクル内で取り込むディジタル信号処理部とを有し、
上記中央処理部は、上記第１アドレス信号に応じて、上記第１データバスを介して命令を取り込むことが可能であり、
上記中央処理部は、上記命令を解読して制御信号を発生する命令デコーダを含み、
上記ディジタル信号処理部は、上記中央処理部内の上記命令デコーダから供給される上記制御信号を受けて、上記ディジタル信号処理部の内部回路の動作を制御する内部制御信号を発生するデコード回路を有することを特徴とするマイクロコンピュータ。
【請求項１１】
請求項１０において、
上記ディジタル信号処理部は、内部バスと、上記内部バスに接続された乗算器及び算術論理演算回路と、データレジスタと、上記第２データバスと上記内部バスとの間に接続された第１データバッファ回路と、上記第３データバスと上記内部バスとの間に接続された第２データバッファ回路とを含むことを特徴とするマイクロコンピュータ。
【請求項１２】
請求項１１において、
上記乗算器及び算術論理演算回路、上記データレジスタ、上記第１及び第２データバッファ回路は、上記内部制御信号によってその動作が制御されることを特徴とするマイクロコンピュータ。
【請求項１３】
請求項１０において、
上記中央処理部は、算術論理演算回路を含み、

上記算術論理演算回路の動作は、上記制御信号によって制御されることを特徴とするマイクロコンピュータ。

【請求項１４】

セントラルプロセッシングユニットと、

上記セントラルプロセッシングユニットにより制御可能なディジタル信号処理ユニットと、

上記セントラルプロセッシングユニットに接続された第１乃至第３アドレスバスと、

上記ディジタル信号処理ユニットに接続された第１乃至第３データバスとを有し、

上記第１データバスは、セントラルプロセッシングユニットに接続可能であり、

上記第１及び第２アドレスバスと第１及び第２データバスに接続される第１メモリと、

上記第１及び第３アドレスバスと第１及び第３データバスに接続される第２メモリとを有し、

上記セントラルプロセッシングユニットは、複数の汎用レジスタと、算術論理演算回路と、上記第１アドレスバスに接続され命令アドレスを格納するためのプログラムカウンタと、上記第１データバスに接続され上記第１データバスから供給された命令を格納する命令レジスタと、上記命令レジスタに格納された命令をデコードしデコード結果に従う制御信号を発生する命令デコーダとを含み、

上記複数の汎用レジスタ及び上記算術論理演算回路の動作は、上記制御信号によって制御され、

上記ディジタル信号ユニットは、上記セントラルプロセッシングユニット内の上記命令デコーダから供給される上記制御信号を受けて制御されることを特徴とするマイクロコンピュータ。

【請求項１５】

請求項１４において、

上記ディジタル信号処理ユニットは、上記制御信号を受けて内部制御信号を発生するデコード回路と、内部バスと、上記内部バスに接続された乗算器及び算術論理演算回路と、上記内部バスに接続されたデータレジスタと、上記第２データバスと上記内部バスとの間に接続された第１データバッファ回路と、上記第３データバスと上記内部バスとの間に接続された第２データバッファ回路とを含み、

上記セントラルプロセッシングユニットは、上記第２アドレスバスを介して第１アドレスを出力可能で、上記第３アドレスバスを介して第２アドレスを出力可能で、

上記ディジタル信号処理ユニットは、上記制御信号に応じて上記第２データバスから第１データを読み出して上記第１データバッファ回路に格納し、上記第３データバスから第２データを読み出して上記第２データバッファに格納し、

上記乗算器及び算術論理演算回路、上記データレジスタ、上記第１及び第２データバッファ回路は、上記内部制御信号によってその動作が制御されることを特徴とするマイクロコンピュータ。

【請求項１６】

第１プロセッシングユニットと、

上記第１プロセッシングユニットに同期して動作し、乗算器を含む第２プロセッシングユニットと、

上記第１プロセッシングユニットに接続された第１、第２及び第３アドレスバスと、

上記第２プロセッシングユニットに接続された第１、第２及び第３データバスとを有し、

上記第１データバスは上記第１プロセッシングユニットに接続され、

上記第１アドレスバス及び第１データバスに接続されるインタフェース回路と、

上記第１及び第２アドレスバスと上記第１及び第２データバスに接続された第１記憶装置と、

上記第１及び第３アドレスバスと上記第１及び第３データバスに接続された第２記憶装置とを有し、

上記第１プロセッシングユニットは、上記第１アドレスバスを介して出力した第１アドレス信号に応じて上記インタフェース回路を介して外部から上記第１データバス経由でフェッチした命令を解読し、解読結果にしたがって上記第２プロセッシングユニットを制御するための制御信号を出力するためのデコード回路を有し、

上記第２プロセッシングユニットは、上記制御信号を解読するためのデコード部を有することを特徴とするマイクロコンピュータ。

【請求項１７】

請求項１６において、

上記第１プロセッシングユニットは、中央処理装置を含み、

上記第２プロセッシングユニットは、ディジタル信号処理装置を含むことを特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はセントラルプロセッシングユニットとディジタル信号処理ユニットとを有する半導体集積回路化された論理ＬＳＩに係り、高速演算処理を要するマイクロコンピュータに適用して有効な技術に関する。

【０００２】

【従来の技術】

算術論理演算器と共に乗算器を同一チップに搭載したマイクロコンピュータについて記載されたものの例としては特願平４－２９６７７８号又は米国特許出願第１４５１５７号がある。これによればマイクロコンピュータのようなロジックＬＳＩチップは、セントラルプロセッシングユニット、バス、メモリ、乗算器を供え、特にメモリからデータを読み出す間に、該読出しデータに関する乗算命令のコマンドをセントラルプロセッシングユニットから乗算器へ転送するコマンド信号線を有する。その結果、セントラルプロセッシングユニットがメモリからデータを読み出す間に、読み出しデータに関する乗算命令のコマンドをセントラルプロセッシングユニットから乗算器へ転送するので、メモリと乗算器との間で直接データを転送することが可能になる。

【０００３】

【発明が解決しようとする課題】

本発明者らはセントラルプロセッシングユニットと共にディジタル信号処理ユニットを一つのＬＳＩに搭載してディジタル信号処理を高速化することについて検討した。このとき、前記従来技術はメモリから乗算器へ直接データを転送可能にしている点において乗算処理の高速化を実現しているが、セントラルプロセッシングユニットによる命令実行のパイプライン処理を想定したとき、セントラルプロセッシングユニットが実行すべき命令のフェッチサイクルと乗算処理のためのメモリアクセスサイクルとが競合するような事態に対しては考慮されていなかった。また、加算や乗算のための複数のオペランドを並列的にメモリから読み出して演算処理の高速化を図る点についても考慮されていない。更にその場合には、セントラルプロセッシングユニットによる外部アクセスとの関係も考慮しなければ、マイクロコンピュータの使い勝手が悪くなることが見出された。また、セントラルプロセッシングユニットと共にディジタル信号処理ユニットを一つのＬＳＩに搭載する場合、ＣＰＵ命令とＤＳＰ命令とのコード割り当てやＤＳＰ命令のフォーマットを工夫することも、命令デコード回路などの論理規模の増大を極力抑える上においては必要であることが見出された。

【０００４】

本発明の目的は、セントラルプロセッシングユニットと共にディジタル信号処理ユニットを一つのＬＳＩに搭載してディジタル信号処理を高速化することにある。本発明の別の目的は、セントラルプロセッシングユニットと共にディジタル信号処理ユニットを一つのＬＳＩに搭載したとき、その物理的な規模の増大を極力抑えることである。

【０００５】

本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【０００６】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【０００７】

すなわち、マイクロコンピュータは、セントラルプロセッシングユニット（２）と、該セントラルプロセッシングユニットから選択的にアドレスが伝達される第１乃至第３のアドレスバス（ＩＡＢ，ＹＡＢ，ＸＡＢ）と、前記第１のアドレスバス（ＩＡＢ）及び第２のアドレスバス（ＹＡＢ）に接続され、セントラルプロセッシングユニットからのアドレスによってアクセスされる第１のメモリ（５，７）と、前記第１のアドレスバス（ＩＡＢ）及び第３のアドレスバス（ＸＡＢ）に接続され、セントラルプロセッシングユニットからのアドレスによってアクセスされる第２のメモリ（４，６）と、前記第１及び第２のメモリと前記セントラルプロセッシングユニットとに接続されてデータが伝達される第１のデータバス（ＩＤＢ）と、前記第１のメモリに接続されてデータが伝達される第２のデータバス（ＹＤＢ）と、前記第２のメモリに接続されてデータが伝達される第３のデータバス（ＸＤＢ）と、前記第１のアドレスバスと第１のデータバスに接続された外部インタフェース回路（１２）と、第１乃至第３のデータバスに接続されセントラルプロセッシングユニットに同期動作されるディジタル信号処理ユニット（３）と、前記ディジタル信号処理ユニットの動作を制御するＤＳＰ制御信号（２０）をセントラルプロセッシングユニットからディジタル信号処理ユニットに伝達する制御信号線とを１チップに含んで半導体集積回路化されて成る。

【０００８】

上記した手段によれば、内蔵メモリは、ディジタル信号処理プロセッサ（３）による積和演算を考慮して第１のメモリ（５，７）と第２のメモリ（４，６）に２面化され、セントラルプロセッシングユニット（２）が第１のメモリと第２のメモリを第３のバス（ＸＡＢ，ＸＤＢ）と第２のバス（ＹＡＢ，ＹＤＢ）によってそれぞれ並列的にアクセス可能にされている。これにより、内蔵メモリから２個のデータを同時にディジタル信号処理ユニットに転送可能にされる。さらに、第３のバス（ＸＡＢ，ＸＤＢ）と第２のバス（ＹＡＢ，ＹＤＢ）は、外部にインタフェースされる第１のバス（ＩＡＢ，ＩＤＢ）とも個別化されているので、セントラルプロセッシングユニットは第２のメモリ（４，６）と第１のメモリ（５，７）のアクセスに並行して外部メモリアクセスも可能にされる。このように、それぞれセントラルプロセッシングユニット（２）に接続された第１乃至第３の３種類のアドレスバス（ＩＡＢ，ＸＡＢ，ＹＡＢ）及びデータバス（ＩＤＢ，ＸＤＢ，ＹＤＢ）があるために、当該３種類の内部バスを使用して同一クロックサイクルで異なるメモリアクセス動作を実行することが可能である。したがって、プログラムやデータが外部メモリに存在する場合にも容易に対応して演算処理の高速化を実現できる。

【０００９】

マイクロコンピュータの使い勝手を向上させるには、前記第１のメモリと第２のメモリの夫々をＲＡＭとＲＯＭから構成するとよい。

【００１０】

前記セントラルプロセッシングユニットにおける積和演算などの繰返し演算のためのアドレス生成の高速化のためには、セントラルプロセッシングユニットはモジュロアドレス出力部（２００）を備えるとよい。このとき、モジュロアドレス出力部で生成されたアドレスは前記第２又は第３のアドレスバスに選択的に出力可能にすることが望ましい。

【００１１】

前記ディジタル信号処理プロセッサは、第１乃至第３のデータバス（ＩＤＢ，ＹＤＢ，ＸＤＢ）と個別的にインタフェースされる第１乃至第３のデータバッファ手段（ＭＤＢＩ，ＭＤＢＹ，ＭＤＢＸ）と、夫々のデータバッファ手段に内部バスを介して接続可能にされ

た複数のレジスタ手段（３０５～３０８）と、前記内部バスに接続された乗算器（３０４）及び算術論理演算器（３０２）と、前記ＤＳＰ制御信号をデコードして前記データバッファ手段、乗算器、算術論理演算器、及びレジスタ手段の動作を制御するデコーダ（３４）とを含んで構成することができる。

【００１２】

命令デコードという点に着目したとき、マイクロコンピュータは、セントラルプロセッシングユニット（２）と、前記セントラルプロセッシングユニットによってアクセス制御されるメモリ（４～７）と、前記メモリ及び前記セントラルプロセッシングユニットとの間でデータが伝達されセントラルプロセッシングユニットに同期動作されるディジタル信号処理ユニット（３）とを１チップに含んで半導体集積回路化される。このマイクロコンピュータによって実行可能な命令セットは、セントラルプロセッシングユニット（２）が実行すべきＣＰＵ命令と、データフェッチのためのアドレス演算等の一部の処理をセントラルプロセッシングユニットに負担させてディジタル信号処理ユニット（３）が実行すべきＤＳＰ命令とを含む。前記セントラルプロセッシングユニットは、前記データバスを介して１６ビット固定長のＣＰＵ命令と、１６ビット又は３２ビット長のＤＳＰ命令とをフェッチする命令レジスタ（２５）と、前記命令レジスタにフェッチされた命令の一部の複数ビットに基づいて、ＣＰＵ命令とＤＳＰ命令とを識別し、識別結果に応じて、前記ディジタル信号処理ユニットの動作制御のためのＤＳＰ制御信号（２０）とセントラルプロセッシングユニットの動作制御のためのＣＰＵ制御信号とを生成するデコーダ（２４）とを含んで構成することができる。

【００１３】

例えば、ＣＰＵ命令は命令コードの最上位４ビットが”００００”～”１１１０”とされる範囲に割り当てられている。ＤＳＰ命令は、命令コードの最上位４ビットが”１１１１”とされる範囲に割り当てられている。さらに命令コードの最上位６ビットが”１１１１００”及び”１１１１０１”の範囲にに割り当てられた命令は、ＤＳＰ命令でも１６ビット長の命令コードとされる。命令コードの最上位６ビットが”１１１１１０”の命令は、３２ビット長の命令コードとされる。命令コードの最上位６ビットが”１１１１１１”の範囲には命令を割り当てておらず、その範囲を未使用領域とする。このように、最大３２ビットの命令に対するコード割り当てに上記のような規則を設けることにより、各命令コードの一部例えば最上位側６ビットをデコードすれば、当該命令がＣＰＵ命令であるか、１６ビット長のＤＳＰ命令であるか、３２ビット長のＤＳＰ命令であるかを、小さな論理規模のデコーダで判定することができ、常に３２ビット全部を一度にデコードすることを要しない。

【００１４】

前記デコーダは、命令レジスタの上位１６ビットをデコードして前記ＣＰＵデコード信号（２４３）及びＤＳＰデコード信号（２４４）を生成する第１のデコード回路（２４０）と、第１のデコード回路にて３２ビット長のＤＳＰ命令を識別したときには命令レジスタの下位１６ビットをコード化した信号を、それ以外の命令を識別したときには出力が無効であることを意味するコードを出力するコード変換回路（２４２）とを含み、前記ＤＳＰデコード信号及びコード変換回路の出力をＤＳＰ制御信号（２０）とする。

【００１５】

ＤＳＰ命令の命令フォーマットの点に着目したとき、マイクロコンピュータは、セントラルプロセッシングユニット（２）と、前記セントラルプロセッシングユニットに同期動作されるディジタル信号処理ユニット（３）と、前記セントラルプロセッシングユニット及び前記ディジタル信号処理ユニットが共通接続された内部バス（ＩＤＢ）とを含んで半導体集積回路化され、前記セントラルプロセッシングユニットは、ディジタル信号処理ユニットとの間でのデータ転送を当該セントラルプロセッシングユニットに対して規定する第１のコード領域（図１８に例示される１６ビットＤＳＰ命令のビット９～ビット０）を有する第１フォーマットの命令と、前記第１のコード領域と同一フォーマットの第２のコード領域（図２０、図２１に例示される３２ビットのＤＳＰ命令のＡフィールド）を有する

と共に、当該第２のコード領域で規定された転送データを用いた演算処理をディジタル信号処理ユニットに対して規定する第３のコード領域（図２０、図２１に例示される３２ビットのＤＳＰ命令のＢフィールド）を有する第２フォーマットの命令とを実行するための実行制御手段を備えて成る。

【００１６】

これにより、実行制御手段は、第１及び第２フォーマットの夫々の命令を実行するとき、第１のコード領域と第２のコード領域に対して共通のデコード論理を持つデコード手段を採用でき、マイクロコンピュータの論理規模の縮小に寄与する。

【００１７】

前記第１フォーマットの命令及び第２フォーマットの命令は、それが第１フォーマットか第２フォーマットかを示すための第４コード領域（例えば１６ビットＤＳＰ命令におけるビット１５～ビット１０、３２ビットＤＳＰ命令におけるビット３２～ビット２６）を有する。

【００１８】

前記実行制御手段は、前記第１フォーマットの命令と第２フォーマットの命令に共通に用いられる命令レジスタ（２５）と、前記命令レジスタにフェッチされた命令に含まれる前記第１のコード領域と第４のコード領域又は第２のコード領域と第４のコード領域をデコードするデコード手段（２４０）と、そのデコード結果に従ってアドレス演算を行い、前記データ転送制御を行う実行手段とを含んで構成することができる。

【００１９】

前記命令レジスタは、前記第１のコード領域と第４のコード領域又は第２のコード領域と第４のコード領域の保持に共用される上位領域（ＵＩＲ）と、前記第３のコード領域の保持に利用される下位領域（ＬＩＲ）とを有し、前記デコード手段は、前記第４のコード領域のデコード結果に基づいて、前記命令レジスタが第２フォーマットの命令を保持したことを示す制御信号（２４８）を出力し、その制御信号に基づいて、前記下位領域から第３のコード領域のコードデータを前記ディジタル信号処理ユニットに向けて供給する手段（２４２，２４２Ａ，２４２Ｂ）を含むことができる。

【００２０】

【発明の実施の形態】

図１には本発明の一実施例に係るマイクロコンピュータ１の全体的なブロック図が示される。同図に示されるマイクロコンピュータは半導体集積回路製造技術によって単結晶シリコンのような１個の半導体基板に形成されている。マイクロコンピュータ１は、セントラルプロセッシングユニットとしてのＣＰＵコア（CPU Core）２、ディジタル信号処理ユニットとしてのＤＳＰエンジン（DSP Engine）３、X-ＲＯＭ４、Y-ＲＯＭ５、X-ＲＡＭ６、Y-ＲＡＭ７、割り込みコントローラ（Interrupt Controller）８、バスステートコントローラ（Bus State Conttroller）９、内蔵周辺回路（Peripheral Circuit）１０、１１、外部メモリインターフェース（External Memory Interface）１２、クロックパルスジェネレータ（ＣＰＧ）１３より構成されている。前記X-ＲＯＭ４，Y-ＲＯＭ５は命令若しくは定数データ等を格納するための読み出し専用又は電気的に書き換え可能なリード・オンリ・メモリであり、X-ＲＡＭ６，Y-ＲＡＭ７はデータの一時格納若しくはＣＰＵコア２とＤＳＰエンジン３の作業領域などとして利用されるランダム・アクセス・メモリである。前記X-ＲＯＭ４とX-ＲＡＭ６を内部命令／データ用のＸメモリ（Internal Instrucution/Data X Mem.）と総称し、Y-ＲＯＭ５とY-ＲＡＭ７を内部命令／データ用のＹメモリ（Internal Instrucution/Data Y Mem.）と総称する。

【００２１】

本実施例のマイクロコンピュータ１はそのバス構成として、外部メモリインタフェース１２に結合される内部アドレスバスＩＡＢ及び内部データバスＩＤＢ、外部メモリインタフェース１２に結合されない内部アドレスバスＸＡＢ及び内部データバスＸＤＢ、外部メモリインタフェース１２に結合されない内部アドレスバスＹＡＢ及び内部データバスＹＤＢ、そして内蔵周辺回路１０，１１のための周辺アドレスバスＰＡＢ及び周辺データバスＰ

ＤＢを備える。尚、コントロールバスについては図示を省略してあるが、アドレスバス及びデータバスの対に対応してそれぞれ設けられている。

【００２２】

ＣＰＵコア２には、外部メモリインターフェース１２を通してチップ外部に接続可能なデータバスＩＤＢが接続され、割り込みコントローラ８からの割り込み信号８０が与えられる。ＣＰＵコア２はＤＳＰエンジン３を制御するための制御信号２０をＤＳＰエンジン３に供給する。さらにＣＰＵコア２は、外部メモリインターフェース１２を通してチップ外部に接続可能なアドレスバスＩＡＢと外部メモリインターフェース１２には接続されていないアドレスバスＸＡＢ，ＹＡＢにアドレス信号を出力する。ＣＰＵコア２は、クロックパルスジェネレータ（ＣＰＧ）１３から出力されるノンオーバーラップ２相のクロック信号Ｃｌｏｃｋ１，Ｃｌｏｃｋ２を動作基準クロック信号として動作される。ＣＰＵコア２についてはその詳細を後述するが、図１のＣＰＵコア２には、レジスタファイル２１、算術論理演算器（ＡＬＵ）２２、アドレス加算器（Add-ALU）２３、デコーダ２４、命令レジスタ（ＩＲ）２５が代表的に図示されている。レジスタファイル２１はアドレスレジスタやデータレジスタとして任意に利用され、また、プログラムカウンタ、そしてコントロールレジスタなどを含む。デコーダ２４は命令レジスタ２５にフェッチされた命令をデコードして内部制御信号（図１には図示を省略）及び制御信号２０を生成する。命令レジスタ（ＩＲ）２５は、それぞれ１６ビットの上位側領域（ＵＩＲ）と下位側領域（ＬＩＲ）から成る。詳細については後述するが、下位側領域（ＬＩＲ）の値は選択的に上位側領域（ＵＩＲ）にシフト可能にされている。尚、割込み等の例外発生時の命令実行手順を制御したり、例外発生に対する内部状態の退避復帰をハードウェア的に制御したりするためのシーケンス制御回路は図示を省略してある。

【００２３】

ＤＳＰエンジン３は前記データバスＩＤＢ，ＸＤＢ，ＹＤＢに接続され、クロック信号Ｃｌｏｃｋ１，Ｃｌｏｃｋ２を動作基準クロック信号として動作される。ＤＳＰエンジン３についてはその詳細を後述するが、図１のＤＳＰエンジン３には、データレジスタファイル３１、算術論理演算器及びシフタ（ALU/Shifter）３２、乗算器（ＭＡＣ）３３、及びデコーダ３４が代表的に図示されている。データレジスタファイル３１は積和演算等に利用される。デコーダ３４はＣＰＵコア２から与えられる制御信号２０をデコードして、ＤＳＰエンジン３の内部制御信号（図１には図示を省略）を生成する。

【００２４】

Ｘ-ＲＯＭ４及びＸ-ＲＡＭ６はアドレスバスＩＡＢ，ＸＡＢとデータバスＩＤＢ，ＸＤＢに接続されている。Ｙ-ＲＯＭ５及びＹ-ＲＡＭ７はアドレスバスＩＡＢ，ＹＡＢとデータバスＩＤＢ，ＹＤＢに接続されている。内蔵メモリは、ＤＳＰエンジン３による積和演算を考慮してＸメモリ４，６とＹメモリ５，７に２面化され、内部バスＸＡＢ，ＸＤＢとＹＡＢ，ＹＤＢによってそれぞれ並列的にアクセス可能にされている。さらに、内部バスＸＡＢ，ＸＤＢとＹＡＢ，ＹＤＢは外部にインタフェースされるバスＩＡＢ，ＩＤＢとも個別化されているので、Ｘメモリ４，６とＹメモリ５，７のアクセスに並行して外部メモリアクセスも可能にされる。Ｘメモリ４，６及びＹメモリ５，７はＤＳＰエンジン３による積和演算のためのデータ一時記憶領域、定数データの記憶領域などとして利用される。尚、Ｘ－ＲＡＭ，Ｙ－ＲＡＭはＣＰＵコア２のデータ一時記憶領域若しくはワーク領域としても利用可能であることは言うまでもない。

【００２５】

前記割り込みコントローラ８は、内蔵周辺回路１０，１１などからの割り込み要求信号（Interrupts）８１を入力し、各種割込み要求に対する優先順位付けや割込み要求に対するマスキングのための情報に従って割込み要求を調停して受け付け、受け付けた割込み要求に応ずる割り込みベクタ（Interrupt Vector）８２をアドレスバスＩＡＢに出力し、さらに割り込み信号８０をＣＰＵコア２に出力する。

【００２６】

バスステートコントローラ９はアドレスバスＩＡＢ，ＰＡＢとデータバスＩＤＢ，ＰＤＢ

に接続され、アドレスバスＰＡＢ及びデータバスＰＤＢに接続されている内蔵周辺回路１０，１１とＣＰＵコア２とのインタフェース制御を行う。

【００２７】

外部メモリインターフェース１２は、アドレスバスＩＡＢとデータバスＩＤＢに接続され、マイクロコンピュータ１のチップ外部の図示を省略したアドレスバスとデータバスに接続され、外部とのインタフェース制御を行う。

【００２８】

図２にマイクロコンピュータ１のアドレスマップの一例が示される。本実施例のマイクロコンピュータ１は３２ビットで規定されるアドレス空間を管理する。前記アドレスバスＩＡＢはビット幅が３２ビットとされる。そのアドレス空間の中には、例外処理ベクタ領域、X-ROM空間（Ｘ－ＲＯＭ４に割り当てられたアドレス空間）、X-RAM空間（Ｘ－ＲＡＭ７に割り当てられたアドレス空間）、Y-ROM空間（Ｙ－ＲＯＭ５に割り当てられたアドレス空間）、Y-RAM空間（Ｙ－ＲＡＭ７に割り当てられたアドレス空間）、内蔵周辺回路割付け空間（内蔵周辺回路１０，１１が割り当てられたアドレス空間）などが存在する。図２の例はX-ROM４は２４ＫＢ、X-RAM６は４ＫＢ、Y-ROM５は２４ＫＢ、Y-RAM７は４ＫＢが割り当てられている。

【００２９】

図２に従えば、１６進数表記でH'００００００００～H'０００００３ＦＦの空間の２５６Ｂ領域には例外処理ベクタ領域が割り付けられている。H'０００００４００～H'０１ＦＦＦＦＦＦにはユーザによって使用可能な通常空間が割り付けられている。通常空間はマイクロコンピュータ１の外部に接続可能なメモリ領域とされる。H'０２０００００～H'０２００５ＦＦＦには、X-ROM空間が割り付けられている。H'０２００６０００～H'０２００６ＦＦＦには、X-RAM空間が割り付けられている。H'０２００７０００～H'０２００７ＦＦＦはX-RAM_Mirror空間となっており、ここをアクセスすると実際にはH'０２００６０００～H'０２００６ＦＦＦのX-RAM空間をアクセスすることになる。H'０２００８０００～H'０２００ＦＦＦＦは、X-RAM，RAM_Mirror空間となっており、ここをアクセスすると実際にはH'０２０００００～H'０２００７ＦＦＦのX-ROM空間およびX-RAM空間をアクセスすることになる。H'０２０１００００～H'０２０１５ＦＦＦには、Y-ROM空間が割り付けられている。H'０２０１６０００～H'０２０１６ＦＦＦには、Y-RAM空間が割り付けられている。H'０２０１７０００～H'０２０１７ＦＦＦはY-RAM_Mirror空間となっており、ここをアクセスすると実際にはH'０２０１６０００～H'０２０１６ＦＦＦのY-RAM空間をアクセスすることになる。H'０２０１８０００～H'０２０１ＦＦＦＦは、Y-ROM，RAM_Mirror空間となっており、ここをアクセスすると実際にはH'０２０１００００～H'０２０１７ＦＦＦのY-ROM空間およびY-RAM空間をアクセスすることになる。H'０２０２００００～H'０７ＦＦＦＦＦＦＦＦには、通常空間が割り付けられている。H'０８０００００～H'１ＦＦＦＦＦＦＦＦには、予約領域が割り付けられている。この予約領域は、ユーザチップ（実チップ）の場合にはアクセス不可能になっており、エバチップ（エミュレーションなどに利用される評価用チップ）の場合にはＡＳＥ空間（エミュレーション用の制御空間）領域として割り当てられる。H'２０００００００～H'２７ＦＦＦＦＦＦＦＦには、通常空間が割り付けられている。H'２８０００００～H'ＦＦＦＦＦＤＦＦには、予約領域が割り付けられている。H'ＦＦＦＦＦＥ００～H'ＦＦＦＦＦＦＦＦには　内蔵周辺回路のレジスタアドレス値を割り付ける内蔵周辺回路割り付け領域が割り付けられている。

【００３０】

図３にはモジュロアドレス出力部を詳細に示したＣＰＵコア２のブロック図が示される。図３において破線で囲った部分がモジュロアドレス出力部２００である。モジュロアドレス出力部２００は、モジュロアドレスレジスタ（例えばＡ０Ｘ）より出力した値をバッファ（例えばＭＡＢＸ）を通してアドレスバス（例えばＸＡＢ）に出力すると同時に、モジュロアドレスレジスタ（Ａ０Ｘ）より出力した値を加算手段（例えばＡＬＵ）で加算して

再びモジュロアドレスレジスタ（Ａ０Ｘ）に格納するアドレス更新出力動作などを行う回路ブロックであり、積和演算のような繰返し演算のためのデータアクセスアドレスを順次更新して生成する。ランダムロジック回路（Random Logic Circuit）２０１と記載された回路ブロックは、図１のデコーダ２４や前記シーケンス制御回路、そしてコントロールレジスタやステータスレジスタなどを含む回路ブロックである。

【００３１】

図３においてＣ１，Ｃ２，ＤＲ，Ａ１，Ｂ１，Ａ２，Ｂ２，ＤＷはそれぞれＣＰＵコア２内部の代表的に示されたバスである。ＣＰＵコア２とデータバスＩＤＢとのインタフェースは前記命令レジスタ（ＩＲ）２５及びデータバッファ（Data Buffer）２０３にて行われる。命令レジスタ（ＩＲ）２５にフェッチされた命令はランダムロジック回路（Random Logic Circuit）２０１に含まれる前記デコーダ２４等に供給される。ＣＰＵコア２とアドレスバスＩＡＢとのインタフェースはプログラムカウンタ（ＰＣ）２０４及びアドレスバッファ（Address Buffer）２０５で行われる。ＣＰＵコア２とアドレスバスＸＡＢとのインタフェースはメモリアドレスバッファ（ＭＡＢＸ）２０６で行われ、ＣＰＵコア２とアドレスバスＹＡＢとのインタフェースはメモリアドレスバッファ（ＭＡＢＹ）２０７で行われる。アドレスバッファ２０５へのアドレス情報の入力経路は、バスＣ１，Ａ１，Ａ２の中から選択可能にされ、メモリアドレスバッファ２０６，２０７へのアドレス情報の入力経路は、バスＣ１，Ｃ２，Ａ１，Ａ２の中から選択可能にされる。算術演算器（ＡＵ）２０８はプログラムカウンタ２０４の値のインクリメントに利用される。２０９は汎用レジスタ（Reg.）、２１０はアドレスのインデックス修飾に利用されるインデックスレジスタ（Ｉｘ）、２１１は同じくインデックス修飾に利用されるインデックスレジスタ（Ｉｙ）、２１２はアドレス演算専用の加算器（ＰＡＵ）、２１３は算術論理演算器（ＡＬＵ）である。

【００３２】

制御ビットＭＸＹはアドレスバスＸＡＢ又はアドレスバスＹＡＢのどちらのアドレスに対しモジュロ演算を行うかを指定し、論理値”１”によってアドレスバスＸＡＢを、論理値”０”によってアドレスバスＹＡＢを指定する。制御ビットＤＭはモジュロ演算を行うか否かを指示し、論理値”１”によってモジュロ演算を行うことを指示し、論理値”０”によってモジュロ演算を行わないことを指示する。モジュロスタートアドレスレジスタ（ＭＳ）２１４はモジュロ演算開始アドレスを格納し、モジュロエンドアドレスレジスタ（ＭＥ）２１５はモジュロ演算終了アドレスを格納する。

【００３３】

モジュロアドレスレジスタ（Ａ０ｘ,Ａ１ｘ）２１６は現在のモジュロアドレスを格納するカレントアドレスレジスタ、２１７はモジュロエンドアドレスレジスタ（ＭＥ）２１５の値とモジュロアドレスレジスタ（Ａ０ｘ,Ａ１ｘ）２１６の値とを比較するコンパレータ（ＣＭＰ）、２１８はコンパレータ２１７の出力と制御ビットＭＸＹ，ＤＭの３入力に対して論理積を採るアンドゲート、２１９はバスＣ１の値とモジュロスタートアドレスレジスタ（ＭＳ）２１４の値とを選択するセレクタであり、それらはアドレスバスＸＡＢに関するモジュロ演算に利用される。セレクタ２１９はアンドゲート２１８の論理値”１”出力によってレジスタ（ＭＳ）２１４の値を選択し、選択した値をモジュロアドレスレジスタ（Ａ０ｘ,Ａ１ｘ）２１６に与える。モジュロアドレスレジスタ２１６はＡ０ｘ又はＡ１ｘの何れかが選択されて利用される。

【００３４】

モジュロアドレスレジスタ（Ａ０ｙ,Ａ１ｙ）２２６は現在のモジュロアドレスを格納するカレントアドレスレジスタ、２２７はモジュロエンドアドレスレジスタ（ＭＥ）２１５の値とモジュロアドレスレジスタ（Ａ０ｙ,Ａ１ｙ）２１６の値とを比較するコンパレータ（ＣＭＰ）、２２８はコンパレータ２２７の出力と制御ビットＭＸＹの反転ビットと制御ビットＤＭとの３入力に対して論理積を採るアンドゲート、２２９はバスＣ２の値とモジュロスタートアドレスレジスタ（ＭＳ）２１４の値とを選択するセレクタであり、それらはアドレスバスＹＡＢに関するモジュロ演算に利用される。セレクタ２２９はアンドゲ

ート２２８の論理値”１”出力によってレジスタ（ＭＳ）２１４の値を選択し、選択した値をモジュロアドレスレジスタ（Ａ０ｙ,Ａ１ｙ）２２６に与える。モジュロアドレスレジスタ２２６はＡ０ｙ又はＡ１ｙの何れかが選択されて利用される。

【００３５】

尚、ランダムロジック回路２０１に記載されたOP Codeは命令レジスタ２５から供給される命令コードを意味し、CONSTは定数値を意味する。

【００３６】

ここで、ＣＰＵコア２におけるモジュロ演算動作として、例えば、モジュロアドレスレジスタ（Ａ０ｘ）２１６を用いて、アドレスバスＸＡＢへ供給すべきアドレス情報をモジュロ演算にて生成する動作を説明する。

【００３７】

先ず、モジュロ演算開始アドレスがモジュロスタートアドレスレジスタ（ＭＳ）２１４に、モジュロ演算終了アドレスがモジュロエンドアドレスレジスタ（ＭＥ）２１５にそれぞれ書き込まれる。モジュロアドレスレジスタ（Ａ０ｘ）にはモジュロ演算を開始するアドレス値が書き込まれる。次にアドレスバスＸＡＢのアドレスに対しモジュロ演算を行うので、ＸＡＢ、ＹＡＢのどちらのアドレスに対しモジュロ演算を行うかを決定する制御ビットＭＸＹに対し論理値”１”が書き込まれる（アドレスバスＹＡＢに対しモジュロ演算を行う場合は、制御ビットＭＸＹに論理値”０”が書き込まれる）。最後にモジュロ演算を行うか否かを判定する制御ビットＤＭに論理値”１”が書き込まれる。

【００３８】

モジュロ演算命令は例えば、ＭＯＶＳ.W @Ａｘ，Ｄｘとされる。この命令記述において、Ａｘはモジュロアドレスレジスタ（Ａ０ｘ）２１６又はモジュロアドレスレジスタ（Ａ１ｘ）２１６とされ、ＤｘはＤＳＰエンジン３内のレジスタに対応する。図３にはＤｘは図示されていない。上記モジュロ演算命令が実行されると、モジュロアドレスレジスタ（Ａ０ｘ）２１６より値が読み出され、メモリアドレスバッファ（ＭＡＢＸ）２０６及び算術論理演算器（ＡＬＵ）２１３に入力される。メモリアドレスバッファ（ＭＡＢＸ）２０６に入力された値はそのままアドレスバスＸＡＢに出力されて、ＸＲＯＭ４またはＸＲＡＭ６のアドレスを指定する。一方、算術論理演算器（ＡＬＵ）２１３に入力されたモジュロアドレスレジスタ（Ａ０ｘ）２１６の値は、インデックスレジスタ（Ｉｘ）２１０の値又は定数（Const）が加算される。インデックスレジスタ（Ｉｘ）２１０との加算を行なう場合は、命令ＭＯＶＳ.W @(Ａｘ，Ｉｘ)，Ｄｘ等を実行したときであり、定数加算される場合は命令ＭＯＶＳ.W @Ａｘ，Ｄｘ等を実行したときである。その加算結果は算術論理演算器（ＡＬＵ）２１３より出力される。算術論理演算器（ＡＬＵ）２１３より出力された値は、セレクタ２１９に入る。このセレクタ２１９のもう一方の入力は、モジュロスタートアドレスレジスタ（ＭＳ）２１４に格納されているモジュロ演算開始アドレスである。

【００３９】

セレクタ２１９の出力が算術論理演算器（ＡＬＵ）２１３の出力になるか、モジュロスタートアドレスレジスタ（ＭＳ）２１４の値になるかは、次のようにして決定される。モジュロアドレスレジスタ（Ａ０ｘ）２１６の値とモジュロエンドアドレスレジスタ（ＭＥ）２１５の値は、コンパレータ（ＣＭＰ）２１７で常に比較されており、一致すればコンパレータ（ＣＭＰ）２１７より論理値”１”が出力され、不一致ならば論理値”０”が出力される。コンパレータ（ＣＭＰ）２１７より出力された値は、制御ビットＤＭ，ＭＸＹと共にアンドゲート２１８で論理積が採られ（この例の場合、ＤＭ、ＭＸＹ共に論理値”１”なので、コンパレータ２１７の値がそのままアンドゲート２１８から出力される。）、セレクタ２１９に入力される。セレクタ２１９は、アンドゲート２１８より入力される値が論理値”１”の場合にモジュロスタートアドレスレジスタ（ＭＳ）２１４の値を選択し、論理値”０”の場合には算術論理演算器（ＡＬＵ）２１３からの出力値を選択する。

【００４０】

アンドゲート２１８より入力される値が論理値”０”の間は、算術論理演算器（ＡＬＵ）

２１３からの出力値を選択し続けるため、アドレスバスＸＡＢに出力される値は、順次更新されていく。モジュロエンドアドレスレジスタ（ＭＥ）２１５の値とモジュロアドレスレジスタ（Ａ０ｘ）２１６の値とが一致すると、アンドゲート２１８からセレクタ２１９に入力される値が論理値”１”になり、モジュロスタートアドレスレジスタ（ＭＳ）２１４の値を選択する。それによって、モジュロアドレスレジスタ（Ａ０ｘ）２１６はモジュロスタートアドレスレジスタ（ＭＳ）２１４の値によって初期化される。

【００４１】

上記モジュロ演算の説明では、モジュロアドレスレジスタ（Ａ０ｘ）２１６を利用したときの動作を説明をしたが、モジュロ演算命令ＭＯＶＳ.W ＠Ａｘ，Ｄｘにおけるアドレスをモジュロアドレスレジスタ（Ａ１ｘ）２１６に指定することも可能である。また制御ビットＭＸＹに論理値”０”を指定すれば、アドレスバスＹＡＢに対してモジュロ演算が可能になる。この場合、モジュロ演算命令ＭＯＶＳ.W ＠Ａｘ，Ｄｘにおけるアドレスを、モジュロアドレスレジスタ（Ａ０ｙ）２２６又は（Ａ１ｙ）２２６を指定するための値Ａｙに変更しなければならない。また制御ビットＤＭに０を指定すれば、モジュロ演算の実行を禁止することもできる。

【００４２】

図４にはＤＳＰエンジン３の一例ブロック図が示される。ランダムロジック回路（Random Logic Circuit）３０１と記載された回路ブロックは、図１のデコーダ３４や制御回路、そしてコントロールレジスタやステータスレジスタなどを含む回路ブロックである。その他にＤＳＰエンジン３は、算術論理演算器（ＡＬＵ）３０２、シフタ（ＳＦＴ）３０３、乗算器（ＭＡＣ）３０４、レジスタ（Reg.）３０５、レジスタ（Ａ０,Ａ１）３０６、レジスタ（Ｙ０,Ｙ１）３０７、レジスタ（Ｘ０,Ｘ１）３０８、メモリデータバッファ（ＭＤＢＩ）３０９、メモリデータバッファ（ＭＤＢＸ）３１０、メモリデータバッファ（ＭＤＢＹ）３１１を備える。メモリデータバッファ（ＭＤＢＹ）３１１はデータバスＹＤＢとバスＤ２を接続する。メモリデータバッファ（ＭＤＢＸ）３１０はデータバスＸＤＢとバスＤ１を接続する。メモリデータバッファ（ＭＤＢＩ）３０９はデータバスＩＤＢとバスＣ１，Ｄ１，Ａ１，Ｂ１に接続している。乗算器（ＭＡＣ）３０４はバスＡ１及びＢ１よりデータを入力し、それに対する乗算結果をバスＣ１及びＤ１に出力する。シフタ（ＳＦＴ）３０３はバスＡ２よりデータを入力し、シフト演算結果をバスＣ２に出力する。算術論理演算器（ＡＬＵ）３０２はバスＡ２及びＢ２よりデータを入力し、演算結果をバスＣ２に出力する。

【００４３】

図５にはマイクロコンピュータ１の命令セットに含まれる命令のフォーマット及び命令コードの一例が示される。マイクロコンピュータ１は、ＣＰＵ命令とＤＳＰ命令の２種類の命令をサポートしている。ＣＰＵ命令の全てとＤＳＰ命令の一部は、１６ビット長の命令コードであり、残りのＤＳＰ命令は３２ビット長の命令コードになっている。ＣＰＵ命令とは、ＤＳＰエンジン３を動作させることなく専らＣＰＵコア２によって実行される命令である。ＤＳＰ命令とは、アドレス演算若しくはオペランドアクセスなどの一部の処理をＣＰＵコア２に負担させてＤＳＰエンジン３が実行する命令である。

【００４４】

ＣＰＵ命令は命令コードの最上位側の４ビットが”００００”～”１１１０”までの空間に命令が割り当てられている。ＤＳＰ命令は、命令コードの最上位側の４ビットが”１１１１”に全て割り当てられている。さらに命令コードの最上位側の６ビットが”１１１１００”及び”１１１１０１”に割り当てられた命令は、ＤＳＰ命令でも１６ビット長の命令コードになっている。命令コードの最上位側の６ビットが”１１１１１０”の命令は、３２ビット長の命令コードになっている。命令コードの最上位側の６ビットが”１１１１１１”の空間には命令を割り当てておらず、未使用領域（未定義命令領域）となっている。将来この領域を利用して命令コードを更に拡張することができる。この命令フォーマットより明らかなように、各命令コードの最上位側の６ビットをデコードすれば、当該命令がＣＰＵ命令であるか、１６ビット長のＤＳＰ命令であるか、３２ビット長のＤＳＰ命令

であるか、未定義命令であるかの判定を、小さな論理規模のデコーダで行うことができる。図５のＣＰＵ命令フォーマットにおいて、ｎｎｎｎはディスティネーションオペランドの指定領域、ｓｓｓｓはソースオペランドの指定領域、ｄｄｄｄはディスプレースメントの指定領域、ｉｉｉｉｉｉｉｉはイミディエイト値の指定領域である。尚、ＡＤＤ命令などの場合は、ｎｎｎｎもソースオペランドの指定領域とされ、演算結果はｎｎｎｎに格納される。また、図３に基づいて説明した前記モジュロ演算命令は、図５の命令ＭＯＶＳ．Ｗ ＠Ｒ２，Ａ０に対応されるが、図５における命令記述はオペランド指定の記述形態が図３で説明した内容と相違されている。これは単なる形式の相違であり、実質は同じである。

【００４５】

図６にはＣＰＵコア２のデコーダ２４とＤＳＰエンジン３のデコーダ３４との接続構成例が示される。マイクロコンピュータ１による命令フェッチは３２ビット単位で命令レジスタ（ＩＲ）２５に行われる。デコーダ２４は第１のデコード回路２４０、第２のデコード回路２４１、及びコード変換回路２４２を備える。第１のデコード回路２４０は命令レジスタ（ＩＲ）２５の上位側１６ビットの領域（ＵＩＲ）の値をデコードして、当該命令がＣＰＵ命令か、１６ビットのＤＳＰ命令か、３２ビットのＤＳＰ命令かに応じて、ＣＰＵデコード信号２４３、ＤＳＰデコード信号２４４、コード変換制御信号２４５、及びシフト制御信号２４６を生成する。第２のデコード回路２４１はＣＰＵデコード信号２４３をデコードして、ＣＰＵコア２内部の演算器やレジスタ選択などを行う各種内部制御信号（ＣＰＵ制御信号）２４７を生成する。コード変換回路２４２は、コード変換制御信号２４５にて活性化されると、命令レジスタ（ＩＲ）２５の下位側１６ビットの領域（ＬＩＲ）が保持する情報のビット数を圧縮し若しくはそのまま出力し、コード変換制御信号２４５にて非活性化されると、その出力の無効を意味する情報（ノンオペレーションコード）を出力する。コード変換回路２４２は、信号２４５が非活性状態のとき下位側１６ビットの領域（ＬＩＲ）の値に代えてノンオペレーションコードを出力するという意味では、セレクタによって実現することも可能である。ＤＳＰデコード信号２４４とコード変換回路２４２の出力は、前記ＤＳＰ制御信号２０としてＤＳＰエンジン３のデコーダ３４に供給される。前記第１のデコード回路２４０は、命令レジスタ（ＩＲ）２５の上位側１６ビットの領域（ＵＩＲ）に格納された最上位側の６ビットをデコードすることにより、当該命令コードがＣＰＵ命令か、１６ビットのＤＳＰ命令か、３２ビットのＤＳＰ命令かを判定することができる。

【００４６】

デコードされた命令が１６ビット命令である場合、コード変換制御信号２４５は非活性状態とされ、それによってコード変換回路２４２は出力の無効を意味するノンオペレーションコードを出力する。また、デコードされた命令が１６ビット命令である場合にはシフト制御信号２４６が活性化され、それを受ける命令レジスタ（ＩＲ）２５はその下位側１６ビットの領域（ＬＩＲ）の値を上位側１６ビットの領域（ＬＩＲ）にシフトさせ、シフトされた命令を次に実行すべき命令の全部若しくは一部として利用する。例えば命令レジスタＩＲの上位側１６ビット領域ＵＩＲに１６ビットＣＰＵ命令が格納され、下位側ビット領域ＬＩＲに３２ビットＤＳＰ命令の上位１６ビットの命令コードが格納された場合について説明する。まず、上位側１６ビット領域ＵＩＲに格納された１６ビットＣＰＵ命令が第１デコード回路２４０にてデコードされ、その結果に従ってＣＰＵコア２はその命令を実行し、下位側１６ビット領域ＬＩＲに格納された３２ビットＤＳＰ命令の上位１６ビットの命令コードデータは、上位側１６ビット領域ＵＩＲに転送される。このときランダムロジック回路２０１は、算術演算器ＡＵ２０８に対し、プログラムカウンタＰＣに格納されるべきアドレスのアドレス演算を実行させる。プログラムカウンタＰＣは、算術演算器ＡＵ２０８によって演算されたアドレス演算結果に従うアドレスを格納する。プログラムカウンタＰＣに格納されたアドレスに従って、上記３２ビットＤＳＰ命令の下位１６ビットの命令コードデータが、それを格納する命令メモリから命令レジスタＩＲの下位側１６ビット領域ＬＩＲに転送される。これにより、３２ビットＤＳＰ命令が命令レジスタＩＲ

に格納される。そして、この命令レジスタＩＲに格納された３２ビットＤＳＰ命令は、デコーダ２４を介してＤＳＰエンジン３のデコーダ３４に供給される。また、他の方法として、図示していないが、複数の命令プリフェッチバッファがＣＰＵコア２内に設けられている。複数の命令プリフェッチバッファは、現在実行されている命令から数サイクル先に実行されるべき命令をプリフェッチする。このようなプリフェッチバッファが設けられている場合において、上述のように３２ビットＤＳＰ命令の上位１６ビットの命令コードデータが下位側領域ＬＩＲから上位側１６ビット領域ＵＩＲに転送されるとき、ランダムロジック回路２０１は、上記３２ビットＤＳＰ命令の下位１６ビットの命令コードデータがプリフェッチされている命令プリフェッチバッファを選択する。その選択された命令プリフェッチバッファから３２ビットＤＳＰ命令の下位１６ビットの命令コードデータが読み出され、命令レジスタＩＲの下位側１６ビット領域ＬＩＲに格納される。

【００４７】

デコードされた命令が１６ビットのＣＰＵ命令である場合、ＤＳＰデコード信号２４４はノンオペレーションを意味するコードとされる。デコードされた命令が１６ビットのＤＳＰ命令である場合には、ＣＰＵ制御信号２４７はＣＰＵデコード信号２４３に基づいて第２のデコード回路２４１が生成し、ＤＳＰエンジン３内部の制御信号は実質的にＤＳＰデコード信号２４４をデコーダ３４が解読して生成する。デコードされた命令が３２ビットのＤＳＰ命令である場合、ＣＰＵ制御信号２４７はＣＰＵデコード信号２４３に基づいて第２のデコード回路２４１が生成し、ＤＳＰエンジン３内部の制御信号はＤＳＰデコード信号２４４とコード変換回路２４２の出力をデコーダ３４が解読して生成する。

【００４８】

マイクロコンピュータ１の命令セットには命令コード長が、１６ビットのものと３２ビットのものがあり、上述のように１６ビット長命令と３２ビット長命令では処理が異なるので、それぞれの場合を分けてその動作を詳述する。

【００４９】

始めに１６ビット長命令の場合について説明する。第１のデコード回路２４０は命令レジスタ（ＩＲ）２５にフェッチされた３２ビットの命令コードの内、上位１６ビットをデコードする。第１のデコード回路２４０では、命令コードの最上位６ビットのコードが”１１１１１０”、”１１１１１”以外のときは１６ビット長命令であることがわかるので、このときはＣＰＵデコード信号２４３とＤＳＰデコード信号２４４の出力と共に、命令レジスタ（ＩＲ）２５の下位１６ビット領域ＬＩＲの命令コードデータを上位１６ビット領域ＵＩＲにシフトさせるシフト制御信号２４６を活性化する。活性化されたシフト制御信号２４６を受けた命令レジスタ（ＩＲ）２５は、下位１６ビット領域ＬＩＲに格納されている命令コードを上位１６ビット領域ＵＩＲにシフトする。シフトされた命令コードは、その次に第１のデコード回路２４０でデコードされることになる。デコーダ２４より出力されるＣＰＵデコード信号２４３は、第２デコード回路２４１に出力され、ＤＳＰデコード信号２４４は、ＤＳＰエンジン３に供給される。また、第１のデコード回路２４０は１６ビット長命令であることがわかると、コード変換制御信号２４５を非活性とし、これによってコード変換回路２４２は、下位１６ビットの命令コードが無効であることを示すコードをＤＳＰ制御信号２０の一部として生成する。ＤＳＰエンジン３側では第１のデコード回路２４０より出力されたＤＳＰデコード信号２４４とコード変換回路２４２より出力されたコード信号とをＤＳＰ制御信号２０として入力すると、デコーダ３４が当該ＤＳＰ制御信号２０のデコードを行なう。１６ビットのＤＳＰ命令の場合、コード変換回路２４２より出力されたＤＳＰ制御信号は無効を表わす信号になっているので、デコーダ３４はＤＳＰデコード信号２４４に着目して、ＤＳＰエンジン３内にある乗算器（ＭＡＣ）３０４、算術論理演算器（ＡＬＵ）３０２、及びシフタ（ＳＦＴ）３０３等の制御信号を出力する。ＤＳＰエンジン３はそれら制御信号に従って演算処理を行なう。

【００５０】

次に３２ビット長命令の場合について説明する。ＣＰＵコア２内部にある第１のデコード回路２４０では、命令レジスタ（ＩＲ）２５に３２ビットの命令コードを格納する。そし

て上位１６ビットを第１のデコード回路２４０でデコードし、デコード信号２４３，２４４を出力する。第１のデコード回路２４０では、命令コードの最上位６ビットのコードが”１１１１１０”のときは３２ビット長命令であることがわかるので、コード変換制御信号２４５を活性化し、これによってコード変換回路２４２は、命令レジスタ（ＩＲ）２５の下位１６ビットの命令コードをコード変換する。コード変換された情報はＤＳＰデコード信号２４４と共にＤＳＰエンジン３にＤＳＰ制御信号２０として供給される。デコーダ３４はＤＳＰ制御信号２０をデコードしてＤＳＰエンジン３内部の制御信号を生成する。尚、デコーダ２４，３４は例えばランダムロジック回路で実現することができる。

【００５１】

図１７には図６に対応される別の実施例が示される。図６の実施例では、命令レジスタ２５の下位領域ＬＩＲの命令データが上位領域ＵＩＲにシフトされるものとして説明した。図１７の実施例は、前記命令レジスタ２５と内部データバスＩＤＢとの間に、命令プリフェッチキューを構成する直列２段の命令プリフェッチバッファ２５０，２５１を供え、命令プリフェッチバッファ２５０，２５１の保持データをセレクタ２５２で選択して命令レジスタ２５に与えるよいうになっている。命令プリフェッチバッファ２５０，２５１及び命令レジスタ２５の夫々は、３２ビット単位でデータを保持し、その保持動作は、制御信号φ１～φ３（ＣＬＫ１に同期）によって制御される。特に図示されないが、命令プリフェッチバッファ２５０，２５１及び命令レジスタ２５の夫々は、マスタ・スレーブの構成を有し、マスタ段は対応される制御信号の立ち上がりに同期して入力のラッチ動作を行い、スレーブ段は対応される制御信号の立ち下がりに同期して入力のラッチ動作を行う。これによって、直列２段の命令プリフェッチバッファ２５０，２５１には、プリフェッチされた前後の命令データが格納される。

【００５２】

前記セレクタ２５２は、選択制御信号φ４に従って、ポートＰａに供給される３２ビットの命令データ又はポートＰｂ供給される３２ビットの命令データを選択して命令レジスタ２５に与える。前記ポートＰａには、命令プリフェッチバッファ２５０の上位１６ビット領域ＵＰＢ１を下位側とし、命令プリフェッチバッファ２５１の下位１６ビット領域ＬＰＢ２を上位側とする３２ビットの命令データが供給される。ポートＰｂには命令プリフェッチバッファ２５１に格納されている３２ビットの命令データがそのまま供給される。

【００５３】

これにより、命令プリフェッチバッファ２５１が３２ビットのＤＳＰ命令を保持しているとき、セレクタ２５２は、ポートＰｂの出力を選択することによって当該３２ビットのＤＳＰ命令を命令レジスタ２５にセットすることができる。命令プリフェッチバッファ２５１が１６ビットのＤＳＰ命令又は１６ビットのＣＰＵ命令を上位領域ＵＰＢ２に保持しているとき、セレクタ２５２は、ポートＰｂの出力を選択することによって当該１６ビットの命令を命令レジスタ２５の上位領域ＵＩＲにセットすることができる。命令プリフェッチバッファ２５１が１６ビットのＤＳＰ命令又は１６ビットのＣＰＵ命令を下位領域ＬＰＢ２に保持しているときは、セレクタ２５２が、ポートＰａの出力を選択することによって当該１６ビットの命令を命令レジスタ２５の上位領域ＵＩＲにセットすることができる。命令プリフェッチバッファ２５１が３２ビットＤＳＰ命令の上位側１６ビット命令コードを下位領域ＬＰＢ２に保持し、命令プリフェッチバッファ２５０がその上位領域ＵＰＢ１に当該３２ビットＤＳＰ命令の下位側１６ビット命令コードを保持しているときは、セレクタ２５２が、ポートＰａの出力を選択することによって当該３２ビットＤＳＰ命令を命令レジスタ２５にセットすることができる。

【００５４】

図１７において２５３は、前記命令プリフェッチバッファのラッチ制御信号φ１，φ２、命令レジスタ２５のラッチ制御信号φ３、及び前記選択制御信号φ４を生成する制御ロジックである。この制御ロジック２５３は、１６ビット命令か３２ビット命令かを示す制御信号２４８と命令プリフェッチバッファ２５０，２５１の各領域に実行されないまま残っている命令コードの状態に従って、前記制御信号φ１～φ４を生成する。この制御ロジッ

ク２５３は命令フェッチのための制御論理の一部を構成する。尚、前記制御信号２４８は、第１のデコード回路２４０が命令レジスタ２５の上位領域ＵＩＲから供給される命令コードデータの上位側６ビットをデコードして生成されるものであり、その詳細については後述する。

【００５５】

前記制御論理２５３による命令レジスタ２５への命令コードデータのセットは以下のようにされる。外部からの命令フェッチは、ＣＰＵコア２の命令フェッチタイミング（例えば後述する複数段のパイプラインステージにおける命令フェッチステージＩＦ）において、命令プリフェッチバッファ２５０に３２ビットの命令コードデータを新たに格納する余地がある場合に行われる。そのタイミングで命令フェッチが行われるときは、命令プリフェッチバッファ２５１にはまだ実行されていない命令が残っている。命令プリフェッチバッファ２５１の領域ＵＰＢ２，ＬＰＢ２に格納されている命令コードの双方がまだ実行されていない第１の状態の場合には、命令プリフェッチバッファ２５１の３２ビットの出力がポートＰｂを介してセレクタ２５２で選択されて命令レジスタ２５にセットされる。一方、命令プリフェッチバッファ２５１の下位領域ＬＰＢ２に格納されている命令コードだけがまだ実行されていない第２の状態の場合には、命令プリフェッチバッファ２５０にプリフェッチした上位領域ＵＰＢ１と命令プリフェッチバッファ２５１の下位領域ＬＰＢ２の命令コードデータがポートＰａを介して命令レジスタ２５にセットされる。

【００５６】

前記第１の状態において、命令レジスタ２５の上位領域ＵＩＲにセットされた命令コードデータをデコード回路２４０がデコードした結果、それが３２ビット命令を構成するものである場合には、そのとき、命令プリフェッチバッファ２５０にプリフェッチされた３２ビットの命令コードデータがそのまま命令プリフェッチバッファ２５１に転送される。一方、デコード結果によって１６ビット命令であることが検出されたときは、命令プリフェッチバッファ２５０から次段のバッファ２５１へのデータシフトは行われない。

【００５７】

前記第２の状態では、ポートＰａを介する命令レジスタ２５へのデータセットの後、命令プリフェッチバッファ２５０にプリフェッチされている３２ビットの命令コードデータは、そのまま命令プリフェッチバッファ２５１にシフトされてセットされる。このシフトセット後、命令プリフェッチバッファ２５０に未だ実行されていない命令コードデータが存在しないならば、命令プリフェッチバッファ２５０には、次の命令フェッチタイミングで命令コードデータがプリフェッチされる。

【００５８】

このような制御により、命令フェッチタイミングの後には、まだ処理されていない命令コードデータが命令レジスタ２５にセットされる。このとき、実行されるべき命令が、１６ビットＣＰＵ命令、１６ビットＤＳＰ命令又は３２ビットＤＳＰ命令の何れであっても、その上位側１６ビットは必ず第１のデコード回路２４０に供給されることになる。

【００５９】

図６で説明したコード変換回路２４２は、図１７ではセレクタ２４２Ａとコード変換ロジック２４２Ｂによって構成される。また、第１のデコード回路２４０は、図６の説明ではそれがデコードした命令コードが１６ビット命令であるか否かによってそのレベルが制御される制御信号２４５，２４６を生成したが、図１７の例では、それがデコードした命令コードが１６ビット命令であるのか３２ビット命令（本実施例において３２ビット命令はＤＳＰ命令である）であるのかを識別するための制御信号２４８を出力する。セレクタ２４２Ａは、制御信号２４８が１６ビット命令を意味するときは、ノーオペレーションコードＮＯＰを選択してコード変換ロジック２４２Ｂに供給し、制御信号２４８が３２ビットＤＳＰ命令であることを意味するときは、命令レジスタ２５の下位領域ＬＩＲの命令コードをコード変換ロジック２４２Ｂに供給する。コード変換ロジック２４２Ｂは、特に制限されないが、命令レジスタ２５の下位領域ＬＩＲの命令コードデータの一部例えばレジスタ選択のためのコード情報をＤＳＰエンジン３のデコーダ３４に適する形態に修正して出

力する。
【００６０】
図１７の実施例において第１のデコード回路２４０は命令レジスタ２５の上位領域ＵＩＲが保持する１６ビットの命令コードデータを解読し、これによって得られたＣＰＵデコード信号２４３を第２のデコード回路２４３に与え、また、ＤＳＰデコード信号２４４をデコーダ３４に与える。ＣＰＵデコード信号２４３は、ＣＰＵ命令及びＤＳＰ命令の何れにおいても有意とされ、第２のデコード回路２４１に供給される。第２のデコード回路２４１は、ＣＰＵデコード信号２４３をデコードして、ＣＰＵコア２が行うべきアドレス演算やデータ演算のための制御情報、及び内部メモリＸ－ＲＯＭ４，Ｙ－ＲＯＭ５，Ｘ－ＲＡＭ，Ｙ－ＲＡＭそして外部メモリをアクセスしたりするためのアドレスバスやデータバスの選択制御情報等を出力する。前述の通り、ＤＳＰ命令に対しても、それに必要なアドレス演算やデータパスの選択はＣＰＵコア２が行う。
【００６１】
前記ＤＳＰデコード信号２４４は、前述の通り、第１のデコード回路２４０に供給される命令コードがＤＳＰ命令のためのコードデータである場合に有意とされるデコード信号である。有意ＤＳＰデコード信号２４４は、例えば、ＣＰＵコア２で行われるアドレス演算に従ってアクセスされるメモリとの間でデータの受け渡しを行うＤＳＰエンジン３内のレジスタ等の指定情報を含んでいる。第１のデコード回路２４０に供給される命令コードがＣＰＵ命令である場合には、ＤＳＰデコード信号２４４は無効を意味するコードにされる。
【００６２】
ここで、マイクロコンピュータ１の命令セットに含まれる前記ＤＳＰ命令のコードを更に詳述する。図１８及び図１９は夫々１６ビットのＤＳＰ命令の命令コードが示され、図２０及び図２１には３２ビットのＤＳＰ命令の命令コードが示される。前述のように、ＤＳＰ命令は、命令コードの最上位側の４ビットが”１１１１”に割り当てられ、命令コードの最上位側の６ビットが”１１１１００”及び”１１１１０１”は１６ビットのＤＳＰ命令、命令コードの最上位側の６ビットが”１１１１１０”の命令は３２ビットのＤＳＰ命令とされる。
【００６３】
図１８の第１欄（X Side of Data Transfer）に示される１６ビットＤＳＰ命令の命令フォーマットはＸメモリ（Ｘ－ＲＯＭ４，Ｘ－ＲＡＭ６）とＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送命令であり、第２欄（Y Side of Data Transfer）に示される命令フォーマットはＹメモリ（Ｙ－ＲＯＭ５，Ｙ－ＲＡＭ７）とＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送命令である。上記命令フォーマットにおいて、Ａｘ，ＡｙはＣＰＵコア２に含まれるレジスタアレイ２０９（図３参照）に含まれるレジスタを指定し、Ａｘ＝”０”はレジスタＲ４を指定し、Ａｘ＝”１”はレジスタＲ５を指定し、Ａｙ＝”０”はレジスタＲ６を指定し、Ａｙ＝”１”はレジスタＲ７を指定する。Ｄｘ，Ｄｙ，ＤａはＤＳＰエンジンに含まれるレジスタを指定し、Ｄｘ＝”０”はレジスタＸ０、Ｄｘ＝”１”はレジスタＸ１、Ｄｙ＝”０”はレジスタＹ０、Ｄｙ＝”１”はレジスタＹ１、Ｄａ＝”０”はレジスタＡ０、Ｄａ＝”１”はレジスタＡ１を夫々指定する。Ｉｘ，Ｉｙはイミディエイト値を意味する。
【００６４】
図１９に示される１６ビットＤＳＰ命令の命令フォーマットは、マイクロコンピュータ１の外部に接続された図示しないメモリとＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送命令である。ＡｓはＣＰＵコア２に内蔵されたレジスタアレイ２０９（図３参照）に含まれるレジスタを指定し、ＤｓはＤＳＰエンジンに内蔵されるレジスタＸ１，Ｘ０，Ｙ１，Ｙ０，Ａ１，Ａ０やレジスタアレイ３０５（図４参照）に含まれるレジスタを指定する。
【００６５】
３２ビットＤＳＰ命令のフォーマットは、３２ビットＤＳＰ命令であることを示すコード

”１１１１１０”の領域（ビット３１～ビット２６）、Ａフィールド（ビット２５～ビット１６）及びＢフィールド（ビット１５～ビット０）に大別される。図２０はＡフィールドに着目した場合の当該フィールドのコードとそれに対応されるにニーモニックを示し、図２１はＢフィールドに着目した場合の当該フィールドのコードとそれに対応されるにニーモニックを示す。

【００６６】

図２０に示されるＡフィールドのコードは、図１８に示される１６ビットＤＳＰ命令のビット９～ビット０のコードと同一であり、第２０図の第１欄（X Side of Data Transfer）に示されるＡフィールドのコードはＸメモリ（Ｘ－ＲＯＭ４，Ｘ－ＲＡＭ６）とＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送を規定し、第２欄（Y Side of Data Transfer）に示されるＡフィールドのコードはＹメモリ（Ｙ－ＲＯＭ５，Ｙ－ＲＡＭ７）とＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送を規定する。当該Ａフィールドに含まれるビットＡｘ，Ａｙ，Ｄｘ，Ｄｙ，Ｄａが指定する内容は図１８と全く同じである。

【００６７】

図２１に示されるＢフィールドのコードは、ＤＳＰエンジン３の内部で行われる算術演算、論理演算、シフト演算、レジスタ間のロード／ストアなどの処理を規定する。例えば、ＤＳＰエンジン３の内部で行われる乗算（ＰＭＵＬＳ）、減算（ＰＳＵＢ）、加算（ＰＡＤＤ）、丸め（ＰＲＮＤ）、シフト（ＰＳＨＬ）、論理積（ＰＡＮＤ）、排他的論理和（ＸＯＲ）、論理和（ＯＲ）、インクリメント（ＰＩＮＣ）、ディクリメント（ＰＤＥＣ）、クリア（ＣＬＲ）等の演算や、ＤＳＰエンジン３の内部で行われるロード（ＰＬＤＳ）及びストア（ＰＳＴＳ）等を規定する。図２１の第３欄（3 Operand Operation with Condition）は、条件付きのコードであり、その条件（if cc）としては、ＤＣ（データコンプリート）ビット（データの処理完了を示すビット）の論理値又は無視を選択することができる。

【００６８】

実際の３２ビットＤＳＰ命令は、Ｂフィールドのコードとフィールドのコードとが任意に組み合わされて記述される。即ち、３２ビットのＤＳＰ命令は、マイクロコンピュータ１の内部又は外部から演算対象とされるオペランドをフェッチし、それをＤＳＰエンジン３の内部で演算する処理を規定する。上述の説明から明らかなように、オペランドフェッチのためのアドレス演算やデータパスの選択はＣＰＵ２によって行われる。３２ビットＤＳＰ命令においてオペランドフェッチを規定するＡフィールドのコードは１６ビットのＤＳＰ命令と同じである。１６ビットＤＳＰ命令は、ＤＳＰエンジン３内部のレジスタに対する初期設定などに利用される。

【００６９】

図１７等に示される構成を参照しても明らかなように、３２ビットＤＳＰ命令のＡフィールドのコードデータは命令レジスタ２５における上位領域ＵＩＲにセットされる。また、Ａフィールドと同一のフォーマットを有する１６ビットＤＳＰ命令も上位領域ＵＩＲにセットされる。したがって、その何れにおいても、ＣＰＵコア２は、必要なアドレス演算及びデータフェッチ（若しくはオペランドフェッチ）に必要なデータパスの選択を同様に行えばよい。換言すれば、３２ビットＤＳＰ命令を実行するためのデータフェッチ（若しくはオペランドフェッチ）と１６ビットＤＳＰ命令を実行するためのデータフェッチ（若しくはオペランドフェッチ）とに必要とされるデコード回路２４０、２４１が共通化され、この点においても、マイクロコンピュータ１の論理規模の縮小に寄与する。３２ビットＤＳＰ命令のＡフィールドが指定するＤＳＰエンジン３の内部レジスタの指定情報や１６ビットＤＳＰ命令が指定するＤＳＰエンジン３の内部レジスタの指定情報は、前記ＤＳＰデコード信号２４４としてＤＳＰエンジン３に与えられる。ＤＳＰデコード信号２４４を有意とするか否かは、前記第１のデコード回路２４０が上位領域ＵＩＲの最上位側の４ビットをデコードして決定する。

【００７０】

次に、本実施例のマイクロコンピュータにおける演算制御の内容を図７乃至図１６の命令実行タイミングチャートを参照しながら説明する。本実施例のマイクロコンピュータ１は、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰステージの５段パイプライン動作を行なっている。ＩＦは命令フェッチステージ、ＩＤは命令デコードステージ、ＥＸは演算実行ステージ、ＭＡはメモリアクセスステージ、ＷＢ/ＤＳＰはメモリから取得したデータをＣＰＵコア２のレジスタに取り込むステージまたはＤＳＰエンジン３がＤＳＰ命令を実行するステージである。各図においてInstruction/Data Accessは内部バスＩＡＢ，ＩＤＢを介するメモリアクセスを意味し、アクセス対象は内蔵メモリ４～７の他にマイクロコンピュータ１の外部メモリも可能にされる。X,Y Mem. Accessは内部バスＸＡＢ，ＸＤＢやＹＡＢ，ＹＤＢを介するメモリアクセスを意味し、アクセス対象は内蔵メモリ４～７に限られる。Isnt.Fetchは命令レジスタ（ＩＲ）２５への命令フェッチタイミング、Fetch.Regは命令レジスタ（ＩＲ）２５、Source Data Outはソースデータ出力、Destination Inはディスティネーションデータの入力タイミング、Destination Registerはディスティネーションレジスタ、をそれぞれ意味する。Pointer Reg.はポインターレジスタ、Address Calc.はアドレス演算、Data Fetchはデータフェッチ、DSP Control signal Decord Timingはデコーダ３４によるＤＳＰ制御信号２０のデコードタイミングを意味する。

【００７１】

図７はＣＰＵコア２内部のＡＬＵ演算命令の実行タイムチャートを示す。ここではＡＬＵ演算命令として、ＡＤＤ　Ｒｍ，Ｒｎ を一例とする。

【００７２】

ＩＦステージ直前におけるクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期して、実行すべき命令（ＡＤＤ　Ｒｍ，Ｒｎ ）が格納されているアドレスがアドレスバスＩＡＢに出力される。Instruction/Data Mem. Accessでは、ＩＦステージでメモリアクセス動作が行われる。具体的にはクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間でアドレスバスＩＡＢで指定されたアドレスのデコードが行われ、ＩＦステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間で命令アクセスが行われる。そのためＩＦステージのクロック信号Ｃｌｏｃｋ２の立ち上がりからデータバスＩＤＢに命令が出力される。データバスＩＤＢに出力された命令は、ＩＤステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して命令レジスタ（ＩＲ）２５に取り込まれる。ＩＤステージでは命令レジスタ（ＩＲ）２５に取り込まれたデータのデコードが行なわれる。ＥＸステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、ソースデータが格納されているレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１，Ｂ１にレジスタの値が出力される。命令ＡＤＤ　Ｒｍ，Ｒｎでは、ＲｍとＲｎに指定したレジスタがソースレジスタとされる。ＲｍとＲｎはＣＰＵコア２の内部の任意のレジスタ（図３では、レジスタ２０９内の任意のレジスタ、Ａ０ｘ，Ａ１ｘ，Ｉｘ，Ａ０ｙ，Ａ１ｙ，Ｉｙ、ＲｍおよびＲｎとして指定可能）を指定できる。ＣＰＵコア２の内部バスＡ１，Ｂ１に出力されたデータは算術論理演算器（ＡＬＵ）２１３で加算演算が行われ、その結果はＣＰＵコア２の内部バスＣ１に出力される。ＣＰＵコア２の内部バスＣ１に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してディスティネーションレジスタ（ディスティネーションレジスタは、ＡＤＤ　Ｒｍ，Ｒｎ　命令でＲｎに指定したレジスタとされる）に格納される。このように、ＣＰＵコア２の内部のＡＬＵ演算命令では、ＩＦ，ＩＤ，ＥＸの３段のパイプラインステージで命令実行動作が完了する。

【００７３】

図８はメモリからＣＰＵコア２へのデータ読み込み動作のタイムチャートを示す。メモリからＣＰＵコア２へのデータ読み込み動作命令の一例として、ＭＯＶ.Ｌ　@Ｒｍ，Ｒｎを例にとって動作説明をする。命令フェッチ（ＩＦ）、命令デコード（ＩＤ）までの動作は図７と同じなのでその部分の詳細な説明は省略する。

【００７４】

ＥＸステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アドレ

スポインタとなるレジスタのデータはＣＰＵコア２の内部バスＡ１に出力される。この例では、アドレスポインタとなるレジスタは、Ｒｍで指定したレジスタになる。Ｒｍに指定できるレジスタは、ＣＰＵコア２に含まれる任意のレジスタ（図３では、Ｒｅｇ.に含まれる任意のレジスタ、Ａ０ｘ，Ａ１ｘ，Ｉｘ，Ａ０ｙ，Ａ１ｙ，ＩｙがＲｍとして指定可能）である。ＣＰＵコア２の内部バスＡ１に出力されたデータは、アドレスバッファ２０５に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＩＡＢに出力される。一方ＣＰＵコア２の内部バスＡ１に出力されたデータは算術論理演算器（ＡＬＵ）２１３で演算が行なわれる。この場合、算術論理演算器（ＡＬＵ）２１３は０加算演算を行なう。その結果はＣＰＵコア２の内部バスＣ１に出力される。ＣＰＵコア２の内部バスＣ１に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、Ｒｍで指定されレジスタ）に格納される。Instruction/Data Mem. Accessでは、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期して、アドレスバスＩＡＢに出力したアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスを行なう。そのためＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりからデータバスＩＤＢにデータが出力される。データバスＩＤＢに出力されたデータは、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期してＣＰＵコア２に取り込まれ、ＣＰＵコア２の内部バスＤＷにデータが出力される。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してＣＰＵコア２の内部バスＤＷ上のデータがディステネーションレジスタに格納されて、動作を終了する。この例では、ディステネーションレジスタはＲｎに指定したレジスタになる。Ｒｎに指定できるレジスタは、ＣＰＵコア２に含まれる任意のレジスタ（図３では、Ｒｅｇ.内の任意のレジスタ、Ａ０ｘ，Ａ１ｘ，Ｉｘ，Ａ０ｙ，Ａ１ｙ，ＩｙがＲｎとして指定可能）である。以上のようにメモリからＣＰＵコア２へのデータ読み込み動作命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰの５段のパイプラインステージで命令実行動作が完了する。

【００７５】

図９はＣＰＵコア２からメモリへのデータ書込み動作命令のタイムチャートを示す。ＣＰＵコア２からメモリへのデータ書込み動作命令の一例として、ＭＯＶ.Ｌ　Ｒｍ，@Ｒｎを例にとって動作を説明する。命令フェッチ（ＩＦ）、命令デコード（ＩＤ）の動作は図８と同じなので、その部分の詳細な説明は省略する。

【００７６】

ＥＸステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アドレスポインタとなるレジスタのデータがＣＰＵコア２の内部バスＡ１に出力される。この例では、アドレスポインタとなるレジスタは、Ｒｎで指定したレジスタになる。Ｒｎに指定できるレジスタは、ＣＰＵコア２に含まれる任意のレジスタ（図３では、Ｒｅｇ.内の任意のレジスタ，Ａ０ｘ，Ａ１ｘ，Ｉｘ，Ａ０ｙ，Ａ１ｙ，ＩｙがＲｎとして指定可能）である。ＣＰＵコア２の内部バスＡ１に出力されたデータは、アドレスバッファ２０５に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＩＡＢに出力される。一方ＣＰＵコア２の内部バスＡ１に出力されたデータは算術論理演算器（ＡＬＵ）２１３で演算が行われる。この場合、算術論理演算器（ＡＬＵ）２１３は０加算演算を行なう。その演算結果はＣＰＵコア２の内部バスＣ１に出力される。ＣＰＵコア２の内部バスＣ１に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、Ｒｎで指定したレジスタ）に格納される。命令 ＭＯＶ.Ｌ　Ｒｍ，@Ｒｎの場合、ＥＸステージでアドレス演算を行なうと同時に、メモリへ書き込むべきデータをデータバスＩＤＢに出力する準備が行われる。ＥＸステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、メモリへ書き込むべきデータが格納されているレジスタよりＣＰＵコ

ア２の内部バスＤＲへ値が出力される。この例の場合、メモリへ書き込むべきデータが格納されているレジスタは、Ｒｍで指定したレジスタになる。Ｒｍに指定できるレジスタは、ＣＰＵコア２に含まれる任意のレジスタ（図３では、Ｒｅｇ.内の任意のレジスタ，Ａ０ｘ，Ａ１ｘ，Ｉｘ，Ａ０ｙ，Ａ１ｙ，ＩｙがＲｍとして指定可能）である。ＣＰＵコア２の内部バスＤＲへ出力された値は、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してデータバスＩＤＢに出力される。Instruction/Data Mem. Accessでは、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＩＡＢに出力されたアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してデータバスＩＤＢに出力されたデータが書込まれて、動作を終了する。メモリからＣＰＵコア２へのデータ書込み動作命令では、ＣＰＵコア２としてはデータバスＩＤＢにデータを出力した時点で動作が終了するので、ＩＦ，ＩＤ，ＥＸ，ＭＡの４段のパイプラインステージで動作が完了する。

【００７７】

図１０はＤＳＰ命令を実行するときのタイムチャートを示す。ＤＳＰ命令の一例として、ＰＡＤＤＣ Ｓｘ，Ｓｙ，Ｄｚ　ＮＯＰＸ　ＮＯＰＹ を例にとって動作説明を行う。この命令は、ＤＳＰエンジン３内のレジスタに格納されているデータの加算を行ない、ＤＳＰエンジン３とX-ＲＯＭ４やX-ＲＡＭ６、及びＹ-ＲＯＭ５やＹ-ＲＡＭ７との間でのデータ転送は行なわないという命令である。

【００７８】

命令フェッチの動作は図７と同じなのでその部分の詳細な説明は省略する。ＩＤステージでは、クロック信号Ｃｌｏｃｋ１からクロック信号Ｃｌｏｃｋ２の期間でＣＰＵコア２で取り込んだ命令コードのデコードが行なわれ、ＩＤステージのクロック信号Ｃｌｏｃｋ２のタイミングで命令コードをデコードした結果がＤＳＰ制御信号２０としてＤＳＰエンジン３に出力される。ＤＳＰエンジン３では、ＣＰＵコア２よりＤＳＰ制御信号２０を入力すると、ＭＡステージまでの期間で入力したＤＳＰ制御信号２０をデコードする。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、ソースデータが格納されているレジスタがアクセスされ、ＤＳＰエンジン３の内部バスＡ２，Ｂ２にレジスタの値が出力される。この例では、ソースデータが格納されているレジスタは、ＳｘおよびＳｙで指定したレジスタになる。ＳｘおよびＳｙに指定できるレジスタは、ＤＳＰエンジン３内部の任意のレジスタ（図４では、Ｒｅｇ.内の任意のレジスタがＳｘおよびＳｙとして指定可能）である。ＤＳＰエンジン３の内部バスＡ２，Ｂ２に出力されたデータは算術論理演算器（ＡＬＵ）３０２で演算が行なわれ、その結果はＤＳＰエンジン３の内部バスＣ２に出力される。ＤＳＰエンジン３の内部バスＣ２に出力された演算結果は、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してディステネーションレジスタに格納される。この例では、ディステネーションレジスタは、Ｄｚで指定されたレジスタになる。Ｄｚに指定できるレジスタは、ＤＳＰエンジン３内部の任意のレジスタ（図４では、Ｒｅｇ.内の任意のレジスタ）である。以上のようなＤＳＰ命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰの５段のパイプラインステージで動作が完了する。

【００７９】

図１１はＸ，Ｙメモリ４～７からＤＳＰエンジン３へのデータ読み込み動作命令のタイムチャートを示す。Ｘ，Ｙメモリ４～７からＤＳＰエンジン３へのデータ読み込み動作命令の一例として、ＭＯＶＸ.Ｗ　@Ａｘ，Ｄｘ　ＭＯＶＹ.Ｗ　@Ａｙ，Ｄｙ を例にとってその動作を説明する。この命令は、ＡｘおよびＡｙで指定したアドレスに格納されているデータをＤｘおよびＤｙで指定したレジスタに転送するという命令である。命令フェッチ、命令デコードの動作は図１０と同じなのでその部分の詳細な説明は省略する。

【００８０】

Ｘ，Ｙメモリ４～７からＤＳＰエンジン３へのデータ読み込み動作命令を実行する場合、

アクセスするメモリのアドレスはＣＰＵコア２が生成する。そのためＥＸステージにおけるクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アクセスすべきアドレスが格納されているレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１～Ａ２にレジスタの値が出力される。この例では、アクセスすべきアドレスが格納されているレジスタは、Ａｘ，Ａｙで指定したレジスタになる。Ａｘに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｘ，Ａ１ｘであり、Ａｙに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｙ，Ａ１ｙである。ＣＰＵコア２の内部バスＡ１～Ａ２に出力されたデータは、メモリアドレスバッファ（ＭＡＢＸ，ＭＡＢＹ）に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＸＡＢ，ＹＡＢに出力される。一方ＣＰＵコア２の内部バスＡ１～Ａ２に出力されたデータはＡＬＵ２１３，ＰＡＵ２１２でアドレス演算が行なわれる。この場合、ＡＬＵ２１３およびＰＡＵ２１２は０加算演算を行なう。その演算結果はＣＰＵコア２の内部バスＣ１及びＣ２に出力される。ＣＰＵコア２の内部バスＣ１及びＣ２に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、ＡｘおよびＡｙで指定したレジスタ）に格納される。Ｘ，Ｙメモリ４～７では、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングでアドレスバスＸＡＢ，ＹＡＢに出力されたアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスが行なわれる。そのためＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりからデータバスＸＤＢ，ＹＤＢにデータが出力される。データバスＸＤＢ，ＹＤＢに出力されたデータは、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期してＤＳＰエンジン３に取り込まれ、ＤＳＰエンジン３の内部バスＤ１，Ｄ２にデータが供給される。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してＤＳＰエンジン３の内部バスＤ１，Ｄ２上のデータがディステネーションレジスタに格納されて、動作を終了する。この例では、ディステネーションレジスタはＤｘおよびＤｙに指定したレジスタになる。Ｄｘに指定できるレジスタは、ＤＳＰエンジン３に含まれるレジスタＸ０，Ｘ１であり、Ｄｙに指定できるレジスタは、ＤＳＰエンジン３に含まれるレジスタＹ０，Ｙ１である。以上のようにメモリからＤＳＰエンジン３へのデータ読み込み動作命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰの５段のパイプラインステージで動作が完了する。斯る並列的なデータ読込み動作は、相互に独立したバスＸＡＢ，ＸＤＢとＹＡＢ，ＹＤＢとを介してＣＰＵコア２がＸ，Ｙメモリ４～７をアクセスできるようになっているからである。

【００８１】

図１２はＤＳＰエンジン３からＸ，Ｙメモリ６，７へのデータ書込み動作のタイムチャートを示す。ＤＳＰエンジン３からＸ，Ｙメモリ６，７へのデータ書込み動作命令の一例として、ＭＯＶＸ.Ｗ　Ｄａ，　@Ａｘ　ＭＯＶＹ.Ｗ　Ｄａ，@Ａｙ　を例にとってその動作を説明をする。この命令は、Ｄａで指定したレジスタに格納されているデータをＡｘおよびＡｙで指定したレジスタに格納されているアドレスに転送するという命令である。

【００８２】

命令フェッチ、命令デコードの動作は図１１と同じなのでその部分の詳細な説明は省略する。ＤＳＰエンジン３からＸ，Ｙメモリ６，７へのデータ書込み動作命令を実行する場合、アクセスされるべきメモリアドレスはＣＰＵコア２が生成する。そのためＥＸステージにおけるクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アクセスすべきアドレスが格納されているレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１～Ａ２にレジスタの値が出力される。この例では、アクセスすべきアドレスが格納されているレジスタは、Ａｘ，Ａｙで指定したレジスタになる。Ａｘに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｘ，Ａ１ｘであり、Ａｙに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｙ，Ａ１ｙである。ＣＰＵコア２の内部バスＡ１，Ａ２に出

力されたデータは、メモリアドレスバッファ（ＭＡＢＸ，ＭＡＢＹ）に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＸＡＢ，ＹＡＢに出力される。

【００８３】

ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、転送されるべきデータが格納されているＤＳＰエンジン３の内部レジスタがアクセスされ、ＤＳＰエンジン３の内部バスＤ１，Ｄ２に当該レジスタの値が出力され、それらがメモリデータバッファ（ＭＤＢＸ，ＭＤＢＹ）に格納される。この例の場合、転送されるべきデータが格納されているＤＳＰエンジン３の内部レジスタはＤａで指定されたレジスタになる。Ｄａで指定できるレジスタは、ＤＳＰエンジン３に含まれるレジスタＡ０及びＡ１である。ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期して、メモリデータバッファ（ＭＤＢＸ，ＭＤＢＹ）に格納されたデータはデータバスＸＤＢ，ＹＤＢに出力される。Ｘ，Ｙメモリ６，７では、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングでアドレスバスＸＡＢ，ＹＡＢに出力されたアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスが行なわれる。そのため、データバスＸＤＢ，ＹＤＢに出力されたデータはＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから書込まれる。以上のようにＤＳＰエンジン３からＸ，Ｙメモリ６，７へのデータ書込み動作命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡの４段のパイプラインステージで動作が完了する。斯る並列的なデータ書込み動作は、相互に独立したバスＸＡＢ，ＸＤＢとＴＡＢ，ＹＤＢとを介してＣＰＵコア２がＸ，Ｙメモリ４，６をアクセスできるようになっているからである。

【００８４】

図１３はメモリからＤＳＰエンジン３へのデータ読み込み動作のタイムチャートを示す。メモリからＤＳＰエンジン３へのデータ読み込み動作命令の一例として、ＭＯＶＳ.Ｌ　@Ａｓ，　Ｄｓ　を例にとってその動作を説明をする。この命令は、Ａｓで指定したアドレスに格納されているデータをＤｓで指定したレジスタに転送するという命令である。

【００８５】

基本動作は、図１１に示したＸ，Ｙメモリ４～７からＤＳＰエンジン３へのデータ読み込み動作と同じである。図１１と図１３の違いは、図１１では対象となるメモリがＸ，Ｙメモリ４～７なのでＸバス，Ｙバスを使用するのに対し、図１３では対象となるメモリはマイクロコンピュータ１がサポートする空間に接続されているメモリなので、バスＩＡＢ，ＩＤＢを使用するということである。ＥＸステージクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アクセスすべきアドレスを保有しているレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１にレジスタの値が出力される。この例では、アクセスすべきアドレスが格納されているレジスタは、Ａｓで指定したレジスタになる。Ａｓで指定可能なレジスタはＣＰＵコア２に含まれるＲｅｇ.内の任意のレジスタである。ＣＰＵコア２の内部バスＡ１に出力されたデータは、アドレスバッファ２０５に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＩＡＢに出力される。一方ＣＰＵコア２の内部バスＡ１に出力されたデータは算術論理演算器（ＡＬＵ）２１３でアドレス演算が行なわれる。この場合、算術論理演算器（ＡＬＵ）２１３は０加算演算を行なう。その演算結果はＣＰＵコア２の内部バスＣ１に出力される。ＣＰＵコア２の内部バスＣ１に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、Ａｓで指定したレジスタ）に格納される。アクセス対象となるメモリでは、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングでアドレスバスＩＡＢに出力されたアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスが

行なわれる。そのためＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりからデータバスＩＤＢにデータが出力される。データバスＩＤＢに出力されたデータは、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期してＤＳＰエンジン３に取り込まれ、当該データがＤＳＰエンジン３の内部バスＤ１に供給される。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してＤＳＰエンジン３の内部バスＤ１上のデータがディステネーションレジスタに格納されて、動作を終了する。この例では、ディステネーションレジスタはＤｓに指定したレジスタになる。Ｄｓに指定できるレジスタは、ＤＳＰエンジン３内の任意のレジスタである。以上のようにメモリからＤＳＰエンジン３へのデータ読み込み動作命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰの５段のパイプラインステージで動作が完了する。

【００８６】

図１４はＤＳＰエンジン３からメモリへのデータ書込み動作のタイムチャートを示す。ＤＳＰエンジン３からメモリへのデータ書込み動作命令の一例として、ＭＯＶＳ.Ｌ　Ｄｓ，　@Ａｓを例にとってその動作を説明する。この命令は、Ｄｓで指定したレジスタに格納されているデータをＡｓで指定したアドレスに転送するという命令である。

【００８７】

基本動作は図１２に示したＤＳＰエンジン３からＸ，Ｙメモリへのデータ書込み動作と同じである。図１２と図１４の違いは、図１２では対象となるメモリがＸ，ＹメモリであるのでバスＸＡＢ，ＸＤＢ、バスＹＡＢ，ＹＤＢを使用するのに対し、図１４では対象となるメモリがマイクロコンピュータ１がサポートする空間に接続されているメモリなので、バスＩＡＢ，ＩＤＢを使用するということである。

【００８８】

ＥＸステージクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、転送先のアドレスを保有しているレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１にレジスタの値が出力される。この例では、アクセスすべきアドレスが格納されているレジスタは、Ａｓで指定したレジスタになる。Ａｓで指定可能なレジスタはＣＰＵコア２に含まれるレジスタＲｅｇ.内の任意のレジスタである。ＣＰＵコア２の内部バスＡ１に出力されたデータは、アドレスバッファ２０５に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＩＡＢに出力される。一方ＣＰＵコア２の内部バスＡ１に出力されたデータは算術論理演算器（ＡＬＵ）２１３でアドレス演算が行なわれる。この場合、算術論理演算器（ＡＬＵ）２１３は０加算演算を行なう。その演算結果はＣＰＵコア２の内部バスＣ１に出力される。ＣＰＵコア２のバスＣ１に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、Ａｓで指定したレジスタ）に格納される。

【００８９】

ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、転送すべきデータを格納しているＤＳＰエンジン３内部のレジスタの値がＤＳＰエンジン３の内部バスＤ１に出力され、メモリデータバッファ（ＭＤＢＩ）に格納される。ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期して、メモリデータバッファ（ＭＤＢＩ）に格納されたデータがデータバスＩＤＢに出力される。この例では、転送すべきデータを格納しているＤＳＰエンジン３内部のレジスタはＤｓに指定したレジスタになる。Ｄｓに指定できるレジスタは、ＤＳＰエンジン３内の任意のレジスタである。アクセス対象となるメモリでは、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングでアドレスバスＩＡＢに出力したアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスが行なわれる。そのためＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりタイミングで、ＤＳＰエンジン３より出力されたデータがメモリに書込まれる。以上のようにＤＳＰエンジン３から外部メモリへのデータ書込み動作命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡの４段のパイプラインステージで動作が完了

する。
【００９０】
次に、ＤＳＰ演算命令の一例として、ＰＡＤＤ　Ｓｘ，Ｓｙ，Ｄｕ　ＰＭＵＬ　Ｓｅ，Ｓｆ，Ｄｇ　ＭＯＶＸ.Ｗ　@Ａｘ，Ｄｘ　ＭＯＶＹ.Ｗ　@Ａｙ，Ｄｙ を例にとり、図１５を用いてその動作説明をする。この命令は、ＤＳＰエンジン３内のレジスタに格納されているデータの加算、乗算を行ない、Ｘ-ＲＯＭ４やＸ-ＲＡＭ６及びＹ-ＲＯＭ５やＹ-ＲＡＭ７からＤＳＰエンジン３へのデータ転送を行なうという命令であり、図１０と図１１の動作を合わせた動作である。命令フェッチ、命令デコードの動作は図１０と同じなのでその部分の詳細な説明は省略する。
【００９１】
Ｘ，ＹメモリからＤＳＰエンジン３へのデータ読み込み動作命令を実行する場合、アクセスすべきメモリのアドレスはＣＰＵコア２が生成する。そのためＥＸステージにおけるクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、アクセスすべきアドレスを保有するレジスタがアクセスされ、ＣＰＵコア２の内部バスＡ１，Ａ２にレジスタの値が出力される。この例では、アクセスすべきアドレスが格納されているレジスタは、Ａｘ，Ａｙで指定したレジスタになる。Ａｘに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｘ，Ａ１ｘであり，Ａｙに指定できるレジスタはＣＰＵコア２に含まれるレジスタＡ０ｙ，Ａ１ｙである。ＣＰＵコア２の内部バスＡ１，Ａ２に出力されたデータは、メモリアドレスバッファ（ＭＡＢＸ，ＭＡＢＹ）に格納され、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してアドレスバスＸＡＢ，ＹＡＢに出力される。一方ＣＰＵ内部バスＡ１，Ａ２に出力されたデータはＡＬＵ２１３，ＰＡＵ２１２でアドレス演算が行なわれ（この場合、ＡＬＵ２１３およびＰＡＵ２１２は０加算演算を行なう）、その結果はＣＰＵコア２の内部バスＣ１及びＣ２に出力される。ＣＰＵコア２の内部バスＣ１及びＣ２に出力された演算結果は、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してポインタレジスタ（この場合、ＡｘおよびＡｙで指定したレジスタ）に格納される。Ｘ，Ｙメモリでは、ＭＡステージのクロック信号Ｃｌｏｃｋ１の立ち上がりからクロック信号Ｃｌｏｃｋ２の立ち上りの期間で、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングでアドレスバスＸＡＢ，ＹＡＢに出力されたアドレスのデコードが行なわれ、ＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりから次のクロック信号Ｃｌｏｃｋ１の立ち上がりの期間でデータアクセスが行なわれる。そのためＭＡステージのクロック信号Ｃｌｏｃｋ２の立ち上がりからデータバスＸＤＢ，ＹＤＢにデータが出力される。データバスＸＤＢ，ＹＤＢに出力されたデータは、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期してＤＳＰエンジン３に取り込まれ、ＤＳＰエンジン３の内部バスＤ１，Ｄ２にデータが出力される。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してＤＳＰエンジン３の内部バスＤ１，Ｄ２上のデータがディステネーションレジスタ（Distination Reg.）に格納されて、動作を終了する。この例では、ディステネーションレジスタはＤｘおよびＤｙに指定したレジスタになる。Ｄｘに指定できるレジスタは、ＤＳＰエンジン３内のＸ０，Ｘ１、Ｄｙに指定できるレジスタは、ＤＳＰエンジン３内のＹ０，Ｙ１である。
【００９２】
上記データ転送に並行して、ＤＳＰ演算動作も同時に行なわれる。ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ１の立ち上がりのタイミングに同期して、ソースデータが格納されているレジスタがアクセスされ、ＤＳＰエンジン３の内部バスＡ１，Ａ２、Ｂ１，Ｂ２にレジスタの値が出力される。この例では、ソースデータが格納されているレジスタは、ＡＤＤ（加算）動作の場合はＳｘおよびＳｙで指定したレジスタになり、ＭＵＬ（乗算）動作の場合はＳｅおよびＳｆで指定したレジスタになる。Ｓｘ，Ｓｙ，Ｓｅ及びＳｆに指定できるレジスタは、ＤＳＰエンジン３内部の任意のレジスタである。ＤＳＰエンジン３の内部バスＡ１，Ｂ１に出力されたデータはＭＡＣ３０４で乗算演算が行なわれ、その結果はＤＳＰエンジン３内部バスＣ１に出力される。ＤＳＰエンジン３の内部バスＡ２，Ｂ２

に出力されたデータはＡＬＵ３０２で加算演算が行なわれ、その結果はＤＳＰエンジン３内部バスＣ２に出力される。ＤＳＰエンジン３の内部バスＣ１およびＣ２に出力された演算結果は、ＷＢ/ＤＳＰステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングに同期してディステネーションレジスタに格納される。この例のディステネーションレジスタは、ＡＤＤ動作の場合はDu，ＭＵＬ動作の場合はDgで指定したレジスタになる。DuおよびDgに指定できるレジスタは、ＤＳＰエンジン３内部の任意のレジスタである。

【００９３】

以上のように、ＤＳＰエンジン３内のレジスタに格納されているデータの加算、乗算を行ない、Ｘ-ＲＯＭ４やＸ-ＲＡＭ６及びＹ-ＲＯＭ５やＹ-ＲＡＭ７からＤＳＰエンジン３へのデータ転送を行なう命令では、ＩＦ，ＩＤ，ＥＸ，ＭＡ，ＷＢ/ＤＳＰの５段のパイプラインステージで動作が完了する。

【００９４】

ＤＳＰ演算命令の第２の例として、

Ｉｎｓｔ１：ＰＡＤＤ　Ａ０，Ｍ０，Ａ０　ＰＭＵＬ　Ａ１，Ｘ０，Ａ１　ＭＯＶＸ.Ｗ　@Ｒ４，Ｘ１　ＭＯＶＹ.Ｗ　@Ｒ６，Ｙ０

Ｉｎｓｔ２：ＡＤＤ　　Ｒ８，Ｒ９

Ｉｎｓｔ３：ＡＤＤ　　Ｒ１０，Ｒ１１

Ｉｎｓｔ４：ＡＤＤ　　Ｒ１２，Ｒ１３

の４つの連続する命令を例にとり、図１６を用いてその動作説明をする。この４つの命令は、アドレスバスＩＡＢ，ＸＡＢ、及びＹＡＢを同時に使用することで、同一クロックサイクルに異なる動作を実現する例である。Ｉｎｓｔ１からＩｎｓｔ４までの命令動作は、図７及び図１５と同じなのでその部分の詳細な説明は省略する。

【００９５】

始めにＩｎｓｔ１のＩＦステージで、Ｉｎｓｔ１の命令フェッチが行われる。Ｉｎｓｔ１のＩＤステージ時に、Ｉｎｓｔ２ではＩＦステージになるため、命令フェッチが行われる。

【００９６】

Ｉｎｓｔ１のＥＸステージでは、Ｘ，Ｙメモリへのアクセスを行うためのアドレス演算を行っているときに、Ｉｎｓｔ２ではＩＤステージのため命令デコードを行い、Ｉｎｓｔ３ではＩＦステージのため命令フェッチを行う。

【００９７】

Ｉｎｓｔ１のＭＡステージでは、ＥＸステージで演算されたアドレスがアドレスバスＸＡＢ、およびＹＡＢに出力され（実際にアドレスを出力するタイミングは、ＥＸステージのクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングからである）、データバスＸＤＢ及びＹＤＢよりデータが取り込まれる。このときＩｎｓｔ２ではＥＸステージのためＲ８とＲ９のＡＤＤ演算を行って動作が完了され、Ｉｎｓｔ３はＩＤステージのため命令デコードを行う。そしてＩｎｓｔ４はＩＦステージのため、Ｉｎｓｔ４が格納されているアドレスをアドレスバスＩＡＢへ出力する。実際にアドレスバスＩＡＢへ出力するタイミングは、Ｉｎｓｔ４のＩＦステージの半サイクル前のクロック信号Ｃｌｏｃｋ２の立ち上がりのタイミングからである。このタイミングは、Ｉｎｓｔ１においてアドレスバスＸＡＢ，ＹＡＢにアドレスを出力するタイミング（ＥＸステージの後半及びＭＡステージの前半）と同じタイミングである。すなわちアドレスバスＸＡＢ及びＹＡＢはデータ転送のために使用され、アドレスバスＩＡＢでは命令フェッチのために使用される。マイクロコンピュータ１では、それぞれＣＰＵコア２に接続された内部アドレスバスＩＡＢ，ＸＡＢ，ＹＡＢと内部データバスＩＤＢ，ＸＤＢ，ＹＤＢがあるために、当該３種類の内部バスを使用して同一クロックサイクルで異なるメモリアクセス動作を実行することが可能である。

【００９８】

この後Ｉｎｓｔ１は、ＷＢ/ＤＳＰステージにおいてＤＳＰ演算を行って動作を完了し、Ｉｎｓｔ２はすでに動作完了、Ｉｎｓｔ３はＥＸステージのため Ｒ１０とＲ１１のＡＤＤ演算を行って動作完了し、Ｉｎｓｔ４ではＩＤステージのため命令デコードを行う。

【００９９】
次のサイクルでは、Ｉｎｓｔ４のＥＸステージのみが行われ、Ｒ１２とＲ１３のＡＤＤ演算を行い動作を完了する。
【０１００】
本実施例によれば以下の作用効果を得る。内蔵メモリは、ＤＳＰエンジン３による積和演算を考慮してＹメモリ５，７とＸメモリ４，６に２面化され、ＣＰＵコア２がＹメモリ５，７とＸメモリ４，６を内部バスＸＡＢ，ＸＤＢと内部バスＹＡＢ，ＹＤＢによってそれぞれ並列的にアクセス可能にされている。これにより、内蔵メモリ４～７から２個のデータを同時にＤＳＰエンジン３に転送可能にされる。さらに、内部バスＸＡＢ，ＸＤＢと内部バスバスＹＡＢ，ＹＤＢは、外部にインタフェースされる内部バスＩＡＢ，ＩＤＢとも個別化されているので、ＣＰＵコア２はＸメモリ４，６とＹメモリ５，７のアクセスに並行して外部メモリアクセスも可能にされる。このように、それぞれＣＰＵコア２に接続された３種類のアドレスバスＩＡＢ，ＸＡＢ，ＹＡＢ及びデータバスＩＤＢ，ＸＤＢ，ＹＤＢがあるために、当該３種類の内部バスを使用して同一クロックサイクルで異なるメモリアクセス動作を実行することが可能である。したがって、プログラムやデータが外部メモリに存在する場合にも容易に対応して演算処理の高速化を実現できる。
【０１０１】
前記Ｘメモリ４，６とＹメモリ５、７の夫々をＲＡＭとＲＯＭから構成することにより、マイクロコンピュータの使い勝手を更に向上させることができる。
【０１０２】
上述のように、内蔵メモリはＸメモリ４、６とＹメモリ５，７に２面化され、２面化された各メモリはＲＯＭとＲＡＭを供え、ＲＡＭをデータメモリ、ＲＯＭをプログラムメモリとすることにより、データメモリとプログラムメモリの分離も可能になり、ＤＳＰエンジン３に２個のデータを並列的に転送し、また、命令フェッチ、データ転送、及び演算を並列パイプライン処理にて能率的に行うことができる。
【０１０３】
ＣＰＵコア２がモジュロアドレス出力部２００を備えることにより、ＣＰＵコア２における積和演算などの繰返し演算のためのアドレス生成を高速化することができる。
【０１０４】
ＣＰＵ命令は命令コードの最上位４ビットが”００００”～”１１１０”までの空間に命令が割り当てられている。ＤＳＰ命令は、命令コードの最上位４ビットが”１１１１”に全て割り当てられている。さらに命令コードの最上位６ビットが”１１１１００”及び”１１１１０１”の空間に割り当てられた命令は、ＤＳＰ命令でも１６ビット長の命令コードになっている。命令コードの最上位６ビットが”１１１１１０”の命令は、３２ビット長の命令コードになっている。命令コードの最上位６ビットが”１１１１１１”の空間には命令を割り当てておらず、未使用領域となっている。このように、最大３２ビットの命令に対するコード割り当てに上記のような規則を設けることにより、命令コードの最上位側６ビットをデコードすれば、当該命令がＣＰＵ命令であるか、１６ビット長のＤＳＰ命令であるか、３２ビット長のＤＳＰ命令であるかを、小さな論理規模のデコーダで判定することができ、常に３２ビット全部を一度にデコードすることを要しない。
【０１０５】
図１７に基づいて説明したように、命令フェッチタイミングの後には、まだ処理されていない命令コードデータが命令レジスタ２５にセットされ、このとき、実行されるべき命令が、１６ビットＣＰＵ命令、１６ビットＤＳＰ命令又は３２ビットＤＳＰ命令の何れであっても、その上位側１６ビットを必ず第１のデコード回路２４０に供給することができる。
【０１０６】
３２ビットＤＳＰ命令のＡフィールドのコードは命令レジスタ２５における上位領域ＵＩＲにセットされ、Ａフィールドと同一のフォーマットを有する１６ビットＤＳＰ命令も上位領域ＵＩＲにセットされる。したがって、その何れにおいても、ＣＰＵコア２は、必要

なアドレス演算及びデータフェッチに必要なデータパスの選択を同様に行うことができる。すなわち、３２ビットＤＳＰ命令を実行するためのデータフェッチと１６ビットＤＳＰ命令を実行するためのデータフェッチとのためにデコード回路２４０、２４１を共通化でき、この点においても、マイクロコンピュータ１の論理規模を縮小することができる。

【０１０７】

以上本発明者によってなされた発明を実施例に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。例えばＣＰＵ命令、１６ビットＤＳＰ命令、３２ビットＤＳＰ命令の識別は命令の最上位６ビットを利用することに限定されず、命令コードの数に応じて増減できる。また、命令レジスタに対する下位１６ビットを上位へシフトさせる機能は別の機能に置き換え可能である。また、ＣＰＵコアやＤＳＰエンジンに含まれるレジスタ本数や演算器の種類は上記実施例に限定されず適宜変更可能である。また、メモリの数を２個に限定せずに増加させることが可能である。そしてメモリの数に合わせてメモリに接続されるアドレスバス、データバスの本数を増加させることは可能である。例えば、Ｘ，Ｙメモリの他に新たにＺメモリを設ける。それに合わせてＣＰＵとＺメモリの間にアドレスバスＺＡＢ，ＤＳＰエンジンとＺメモリの間にデータバスＺＤＢを接続する。このような構成にすれば、積和演算時にＸ，ＹメモリからデータをＤＳＰエンジンに取り込むだけでなく、現在実行中の命令以前に演算終了しているデータをＺバスを介してＺメモリ回路に同時に書き込むことが可能となる。１つの命令で演算データの取り込み、メモリへの書き込みが可能となるのでマイクロコンピュータ全体のスループットがさらに向上する。本発明は、移動体通信機器における情報の圧縮伸張処理やフィルタリング処理、サーボ制御、プリンタにおける画像処理等に適用される機器組み込み制御用マイクロコンピュータとしての利用に最適である。

【０１０８】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【０１０９】

すなわち、内蔵メモリは、ディジタル信号処理プロセッサによる積和演算を考慮して第１のメモリと第２のメモリに２面化され、第３のバスと第２のバスによってそれぞれ並列的にアクセス可能にされているから、セントラルプロセッシングユニットは内蔵メモリから２個のデータを同時にディジタル信号処理ユニットに転送することができる。

【０１１０】

さらに、第３のバスと第２のバスは、外部にインタフェースされる第１のバスとも個別化されているので、セントラルプロセッシングユニットは第２のメモリと第１のメモリのアクセスに並行して外部メモリアクセスすることができる。

【０１１１】

このように、それぞれセントラルプロセッシングユニットに接続された第１乃至第３の３種類のアドレスバス及びデータバスがあるために、当該３種類の内部バスを使用して同一クロックサイクルで異なるメモリアクセス動作を実行することができるので、プログラムやデータが外部メモリに存在する場合にも容易に対応して演算処理の高速化を実現できる。

【０１１２】

さらに、内蔵メモリは第１のメモリと第２のメモリに２面化され、２面化された各メモリはＲＯＭとＲＡＭを供え、ＲＡＭをデータメモリ、ＲＯＭをプログラムメモリとすることにより、データメモリとプログラムメモリの分離も可能になり、ディジタル信号処理ユニットに２個のデータを並列的に転送し、また、命令フェッチ、データ転送、及び演算を並列パイプライン処理にて能率的に行うことができる。

【０１１３】

したがって、セントラルプロセッシングユニットと共にディジタル信号処理ユニットを一

つのＬＳＩに搭載したときにディジタル信号処理の高速化を実現できる。

【０１１４】

ＣＰＵ命令とＤＳＰ命令が混在された命令に対し、命令コードの一部をデコードすることによって当該命令がＣＰＵ命令であるか、１６ビット長のＤＳＰ命令であるか、３２ビット長のＤＳＰ命令であるかを識別可能に命令コードを割り当てることにより、小さな論理規模のデコーダで命令の種別を判定することができ、常に３２ビット全部を一度にデコードすることを要しない。したがって、セントラルプロセッシングユニットと共にディジタル信号処理ユニットを一つのＬＳＩに搭載したとき、その物理的な規模の増大を極力抑えることができる。

【０１１５】

ＤＳＰ命令の命令フォーマットとして、ディジタル信号処理ユニットとの間でのデータ転送を当該セントラルプロセッシングユニットに対して規定する第１のコード領域（図１８に例示される１６ビットＤＳＰ命令のビット９～ビット０）を有する第１フォーマットの命令と、前記第１のコード領域と同一フォーマットの第２のコード領域（図２０、図２１に例示される３２ビットのＤＳＰ命令のＡフィールド）を有すると共に、当該第２のコード領域で規定された転送データを用いた演算処理をディジタル信号処理ユニットに対して規定する第３のコード領域（図２０、図２１に例示される３２ビットのＤＳＰ命令のＢフィールド）を有する第２フォーマットの命令とを採用することにより、第１及び第２フォーマットの夫々の命令を実行する手段は、第１のコード領域と第２のコード領域に対して共通のデコード論理を持つデコード手段を採用でき、この点においても、マイクロコンピュータの論理規模を縮小することができる。

【図面の簡単な説明】

【図１】本発明の一実施例に係るマイクロコンピュータの全体ブロック図である。

【図２】マイクロコンピュータの一例アドレスマップである。

【図３】モジュロアドレス出力部を詳細に示したＣＰＵコアのブロック図である。

【図４】ＤＳＰエンジンの一例ブロック図である。

【図５】マイクロコンピュータの命令フォーマット及び命令コードに関する一例説明図である。

【図６】ＣＰＵコアのデコーダとＤＳＰエンジンのデコーダとの接続構成を示すブロック図である。

【図７】ＣＰＵコア内部でのＡＬＵ演算命令の実行タイムチャートである。

【図８】メモリからＣＰＵコアへデータを読込む命令の実行タイムチャートである。

【図９】ＣＰＵコアからメモリへデータを書込み命令の実行タイムチャートである。

【図１０】ＤＳＰ命令を実行するときの一例タイムチャートである。

【図１１】Ｘ，Ｙメモリから ＤＳＰエンジンへデータを読込む命令の実行タイムチャートである。

【図１２】ＤＳＰエンジンからＸ，Ｙメモリへデータを書込む命令の実行タイムチャートである。

【図１３】メモリからＤＳＰエンジンへデータを読込む命令の実行タイムチャートである。

【図１４】ＤＳＰエンジンからメモリへデータを書込む命令の実行タイムチャートである。

【図１５】ＤＳＰ演算命令の一例実行タイムチャートである。

【図１６】ＤＳＰ演算命令を連続して実行するときの一例タイムチャートである。

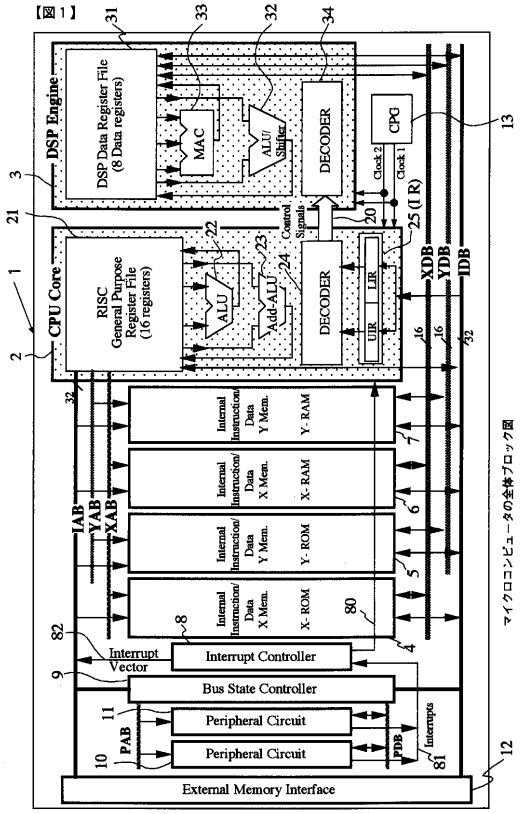
【図１７】図６に対応される別の実施例を示すブロック図である。

【図１８】マイクロコンピュータの内蔵メモリとＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送を規定する１６ビットＤＳＰ命令のコードを示す命令フォーマット図である。

【図１９】マイクロコンピュータの外部メモリとＤＳＰエンジン３の内蔵レジスタとの間におけるデータ転送を規定する１６ビットＤＳＰ命令のコードを示す命令フォーマット図

である。
【図２０】３２ビットＤＳＰ命令のＡフィールドに着目した場合における当該フィールドのコードとそれに対応されるにニーモニックなどを示す命令フォーマット図である。
【図２１】３２ビットＤＳＰ命令のＢフィールドに着目した場合における当該フィールドのコードとそれに対応されるにニーモニックなどを示す命令フォーマット図である。
【符号の説明】
１　マイクロコンピュータ
２　ＣＰＵコア（セントラルプロセッシングユニット）
２０　ＤＳＰ制御信号
２４　デコーダ
２４０　第１のデコード回路
２４１　第２のデコード回路
２４２　コード変換回路
２４３　ＣＰＵデコード信号
２４４　ＤＳＰデコード信号
２４５　コード変換制御信号
２４７　ＣＰＵ制御信号
２５　命令レジスタ
２５０，２５１　命令プリフェッチバッファ
２００　モジュロアドレス出力部
２０６，２０７　メモリアドレスバッファ
２１２　アドレス演算器
２１３　算術論理演算器
２１４　モジュロスタートアドレスレジスタ
２１５　モジュロエンドアドレスレジスタ
２１６、２２６　モジュロアドレスレジスタ
３　ＤＳＰエンジン（ディジタル信号処理ユニット）
３４　デコーダ
３０２　算術論理演算器
３０４　乗算器
３０９，３１０，３１１　メモリデータバッファ
４　Ｘ－ＲＯＭ（第２のメモリ）
５　Ｙ－ＲＯＭ（第１のメモリ）
６　Ｘ－ＲＡＭ（第２のメモリ）
７　Ｙ－ＲＡＭ（第１のメモリ）
１２　外部メモリインタフェース

【図１】

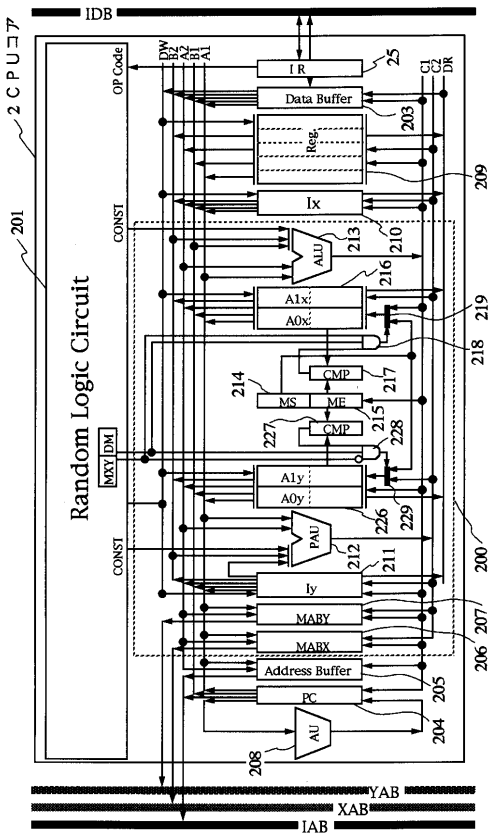


【図２】

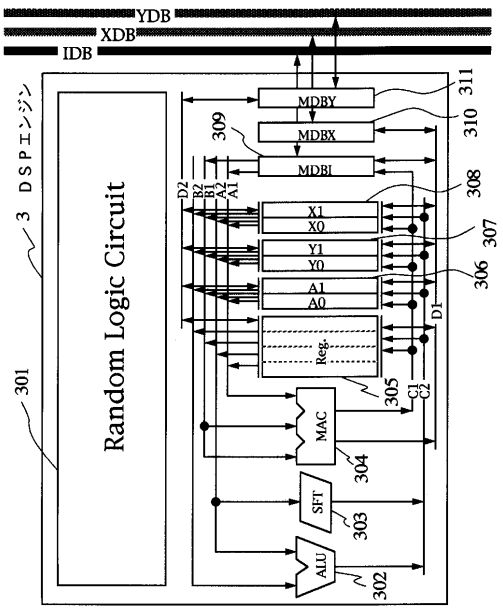


マイクロコンピュータのアドレスマップの一例

【図３】

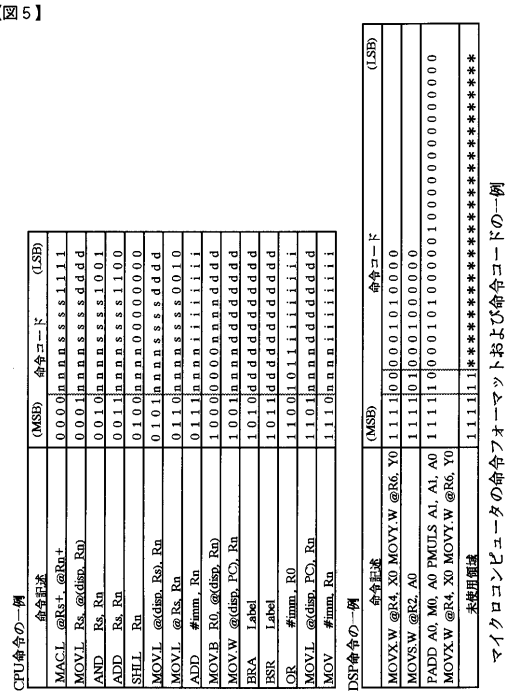


【図４】



【図５】

【図5】



CPU命令の一例

命令記述	(MSB) 命令コード (LSB)
MAC.L @Rs+, @Rn+	0 0 0 0 n n n n s s s s 1 1 1 1
MOV.L Rs, @(disp, Rn)	0 0 0 1 n n n n s s s s d d d d
AND Rs, Rn	0 0 1 0 n n n n s s s s 1 0 0 1
ADD Rs, Rn	0 0 1 1 n n n n s s s s 1 1 0 0
SHLL Rn	0 1 0 0 n n n n 0 0 0 0 0 0 0 0
MOV.L @(disp, Rs), Rn	0 1 0 1 n n n n s s s s d d d d
MOV.L @Rs, Rn	0 1 1 0 n n n n s s s s 0 0 1 0
ADD #imm, Rn	0 1 1 1 n n n n i i i i i i i i
MOV.B R0, @(disp, Rn)	1 0 0 0 0 0 0 0 n n n n d d d d
MOV.W @(disp, PC), Rn	1 0 0 1 n n n n d d d d d d d d
BRA Label	1 0 1 0 d d d d d d d d d d d d
BSR Label	1 0 1 1 d d d d d d d d d d d d
OR #imm, R0	1 1 0 0 1 0 1 1 i i i i i i i i
MOV.L @(disp, PC), Rn	1 1 0 1 n n n n d d d d d d d d
MOV #imm, Rn	1 1 1 0 n n n n i i i i i i i i

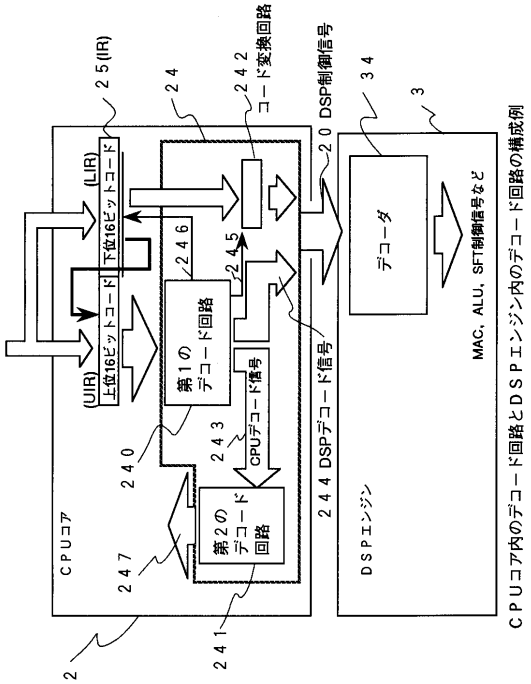
DSP命令の一例

命令記述	(MSB) 命令コード (LSB)
MOVX.W @R4, X0 MOVY.W @R6, Y0	1 1 1 1 0 0 0 0 0 0 1 0 1 0 0 0
MOVS.W @R2, A0	1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 0
PADD A0, M0, A0 PMULS A1, A1, A0	1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0
MOVX.W @R4, X0 MOVY.W @R6, Y0	1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0
未使用命令	1 1 1 1 1 * * * * * * * * * * * *

マイクロコンピュータの命令フォーマットおよび命令コードの一例

【図６】

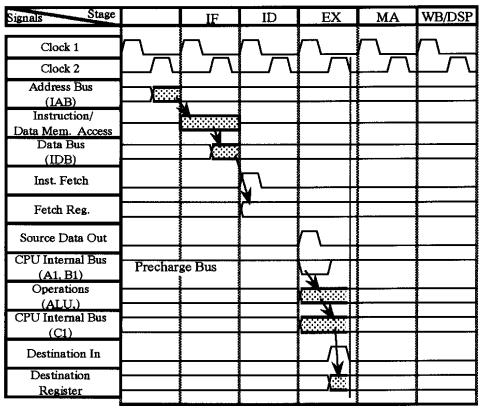
【図6】



CPUコア内のデコード回路とDSPエンジン内のデコード回路の構成例

【図７】

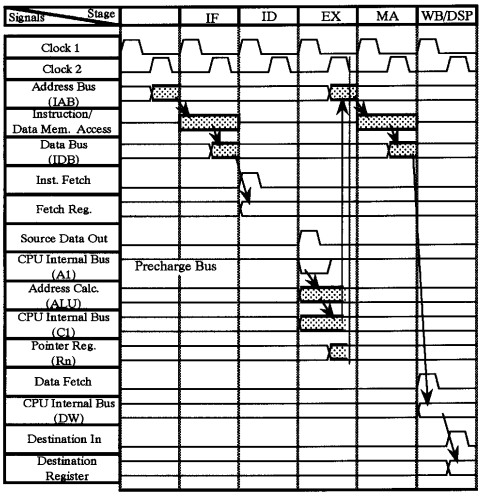
【図7】



ALU Operation in CPU

【図８】

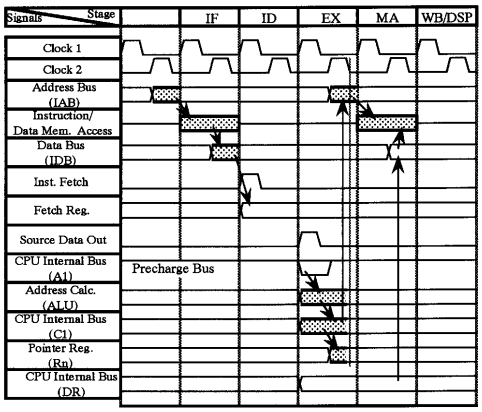
【図8】



Data Move from Mem. to CPU

【図９】

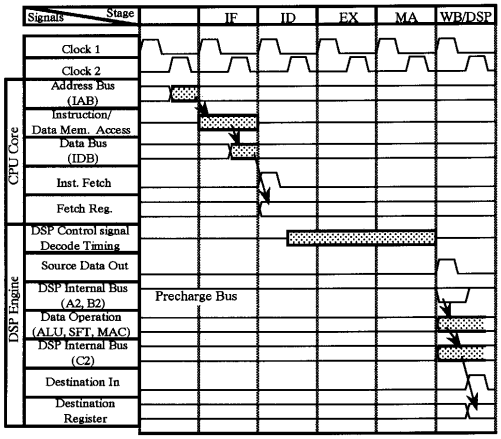
【図9】



Data Move from CPU to Mem.

【図１０】

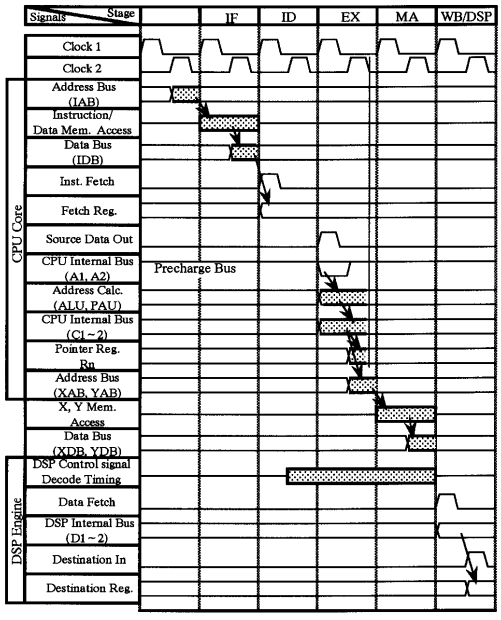
【図10】



DSP Data Operation in CPU & DSP

【図１１】

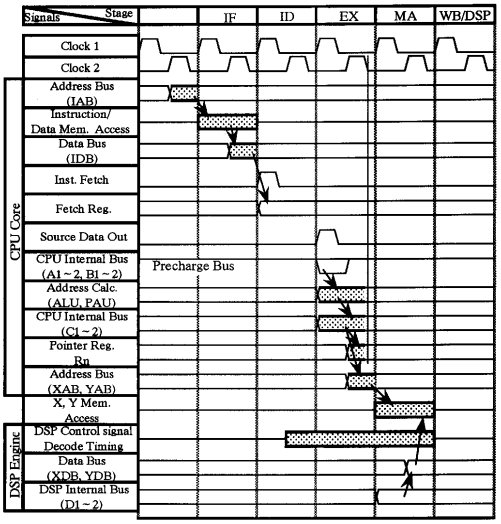
【図11】



Data Move (MOVX, MOVY) from Mem. to DSP

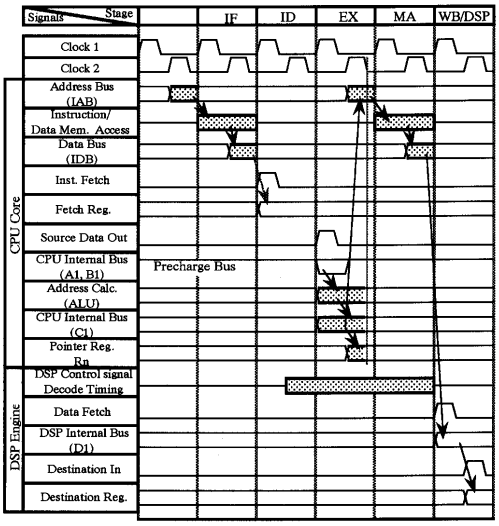
【図１２】

【図12】



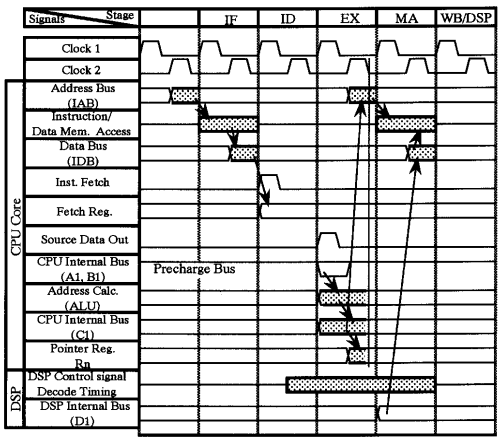
Data Move (MOVX, MOVY) from DSP to Mem.

【図１３】



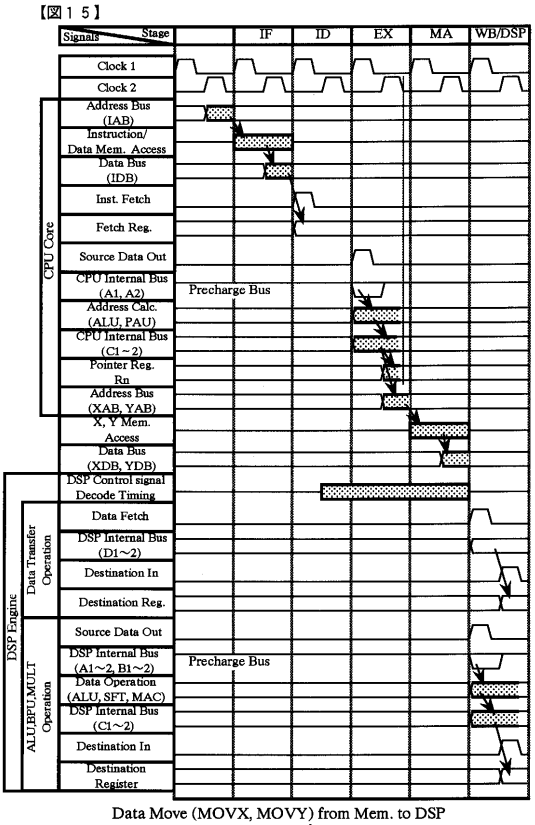
Data Move (MOVSY) from Mem. to DSP

【図１４】



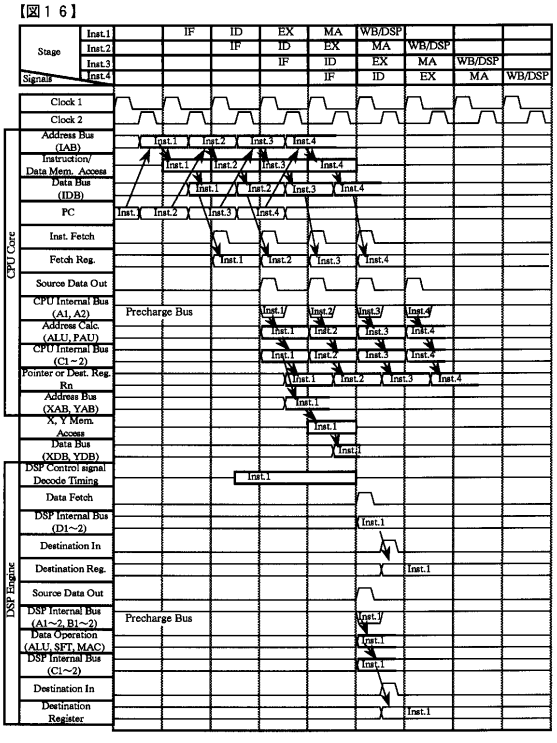
Data Move (MOVSY) from DSP to Mem.

【図１５】



Data Move (MOVX, MOVY) from Mem. to DSP
and
DSP Data Operation in CPU & DSP

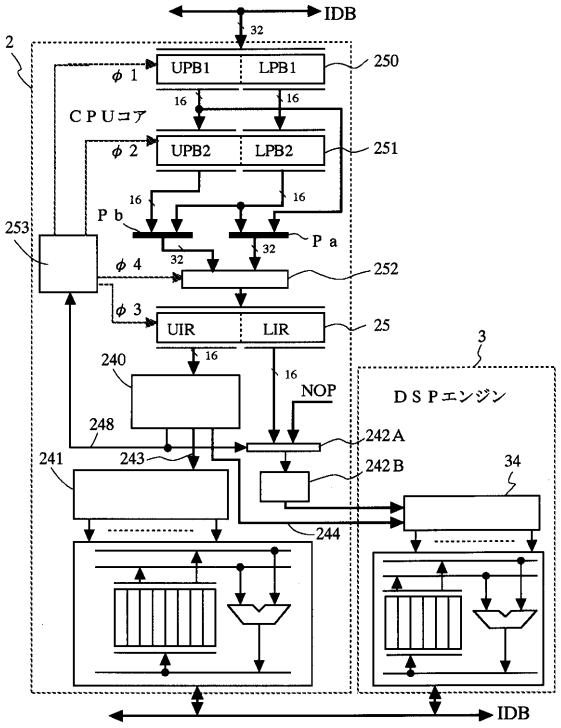
【図１６】



An Example of Access to X,YAB and IAB at the same cycle

【図１７】

【図１７】



【図１８】

【図１８】

DSP16b命令コード（1）

Class	Mnemonics	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X Side of Date Transfer	NOPX	1	1	1	1	0	0	*	*	*	*	*	*	0	0		
	MOVX.W @Ax, Dx							Ax		Dx		0		0	1		
	MOVX.W @Ax+, Dx													1	0		
	MOVX.W @Ax+Ix, Dx											1		1	1		
	MOVX.W Da, @Ax									Da				0	1		
	MOVX.W Da, @Ax+													1	0		
	MOVX.W Da, @Ax+Ix													1	1		
Y Side of Date Transfer	NOPY	1	1	1	1	0	0	*	*	*	*	*	*	0	0		
	MOVY.W @Ay, Dy								Ay		Dy		0			0	1
	MOVY.W @Ay+, Dy															1	0
	MOVY.W @Ay+Iy, Dy															1	1
	MOVY.W Da, @Ay										Da		1			0	1
	MOVY.W Da, @Ay+															1	0
	MOVY.W Da, @Ay+Iy															1	1

Ax : 0=R4, 1=R5 Ay : 0=R6, 1=R7 Dx : 0=X0, 1=X1 Dy : 0=Y0, 1=Y1
Da : 0=A0, 1=A1

【図１９】

【図１９】

DSP16b命令コード（2）

Class	Mnemonics	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Single Date Transfer	MOVS.W @As, Ds	1	1	1	1	1	0	1	As		Ds	0 : (#1)		0	0	0	0
	MOVS.W @As, Ds											1 : (#1)		0	0	1	
	MOVS.W @As+, Ds								0 : R4			2 : (#1)		1	0		
	MOVS.W @As+Ix, Ds								1 : R5			3 : (#1)		1	1		
	MOVS.W Ds, @-As								2 : R2			4 : (#1)		0	0		1
	MOVS.W Ds, @As								3 : R3			5 : A1		0	1		
	MOVS.W Ds, @As+											6 : (#1)		1	0		
	MOVS.W Ds, @As+Ix											7 : A0		1	1		
	MOVS.L @-As, Ds											8 : X0		0	0	1	0
	MOVS.L @As, Ds											9 : X1		0	1		
	MOVS.L @As+, Ds											A : Y0		1	0		
	MOVS.L @As+Ix, Ds											B : Y1		1	1		
	MOVS.L Ds, @-As											C : M0		0	0		1
	MOVS.L Ds, @As											D : A1G		0	1		
	MOVS.L Ds, @As+											E : M1		1	0		
	MOVS.L Ds, @As+Ix											F : A0G		1	1		

【図２０】

【図２０】

DSP32b命令コード（1）

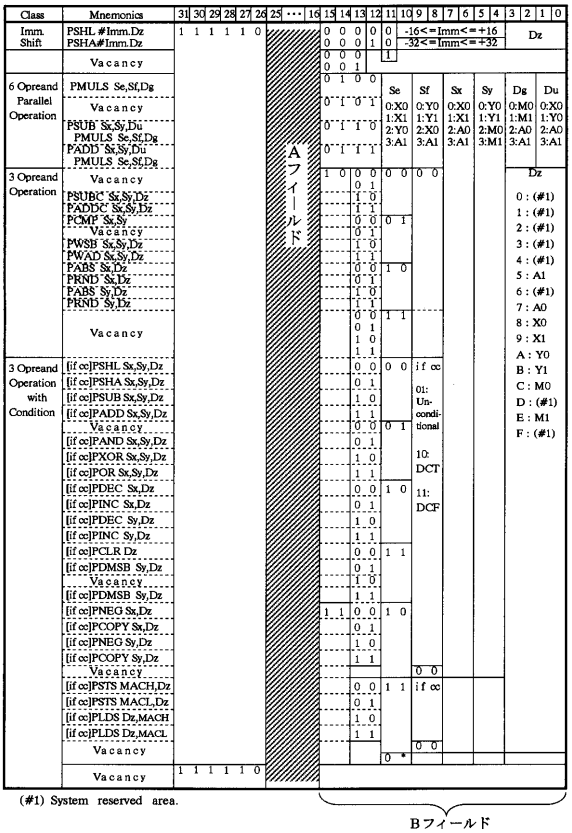
Class	Mnemonics	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	・・・・・・・・・・	0
X Side of Date Transfer	NOPX	1	1	1	1	1	0	*		*		*		0	0				Bフィールド	
	MOVX.W @Ax, Dx							Ax		Dx		0		0	1					
	MOVX.W @Ax+, Dx													1	0					
	MOVX.W @Ax+Ix, Dx											1		1	1					
	MOVX.W Da, @Ax									Da				0	1					
	MOVX.W Da, @Ax+													1	0					
	MOVX.W Da, @Ax+Ix													1	1					
Y Side of Date Transfer	NOPY							*		*		*		0	0				Bフィールド	
	MOVY.W @Ay, Dy								Ay		Dy		0			0	1			
	MOVY.W @Ay+, Dy															1	0			
	MOVY.W @Ay+Iy, Dy											1				1	1			
	MOVY.W Da, @Ay										Da		1			0	1			
	MOVY.W Da, @Ay+															1	0			
	MOVY.W Da, @Ay+Iy															1	1			

Ax : 0=R4, 1=R5 Ay : 0=R6, 1=R7 Dx : 0=X0, 1=X1 Dy : 0=Y0, 1=Y1 Da : 0=A0, 1=A1

Aフィールド

【図２１】

【図２１】　　ＤＳＰ３２ｂ命令コード（２）



(#1) System reserved area.

フロントページの続き

(72)発明者　馬路　徹
東京都小平市上水本町５丁目２０番１号　株式会社日立製作所　半導体事業部内
(72)発明者　野口　孝樹
東京都小平市上水本町５丁目２０番１号　株式会社日立製作所　半導体事業部内
(72)発明者　赤尾　泰
東京都小平市上水本町５丁目２０番１号　株式会社日立製作所　半導体事業部内
(72)発明者　馬場　志朗
東京都小平市上水本町５丁目２０番１号　株式会社日立製作所　半導体事業部内

審査官　多賀　実

(56)参考文献　特開平０６－０２８３１０（ＪＰ，Ａ）
特開昭６３－０７９１６２（ＪＰ，Ａ）
特開平０４－０２３０６３（ＪＰ，Ａ）
特開昭６３－１８０１５１（ＪＰ，Ａ）
特開平０８－１２３６８２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.^{7}，ＤＢ名)
G06F 9/30-38,
12/00-06,
13/16-18,
15/16-177,
15/78,
17/10-18