(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3839068号
(P3839068)
(45)発行日　平成18年11月1日(2006.11.1)　(24)登録日　平成18年8月11日(2006.8.11)

(51)Int.Cl.　F I
G06F 15/78 (2006.01)　G06F 15/78 510G
G06F 15/78 510E
G06F 15/78 510P
G06F 15/78 510K
G06F 15/78 510C

請求項の数 8　(全 37 頁)

(21)出願番号　特願平6-96039
(22)出願日　平成6年5月10日(1994.5.10)
(65)公開番号　特開平7-302255
(43)公開日　平成7年11月14日(1995.11.14)
審査請求日　平成13年5月2日(2001.5.2)

(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　100080001
弁理士　筒井　大和
(72)発明者　三ッ石　直幹
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　酒井　恭信

最終頁に続く

(54)【発明の名称】半導体集積回路装置

(57)【特許請求の範囲】
【請求項１】
第１、第２のデータ処理装置と、バス制御手段と、第１、第２の読み出し可能な記憶手段と、データ入出力手段とを有する半導体集積回路装置であって、
前記第１のデータ処理装置と、前記バス制御手段と、前記第１の読み出し可能な記憶手段とが第１のバスによって相互に接続され、
前記第２のデータ処理装置と、前記バス制御手段と、前記第２の読み出し可能な記憶手段とが第２のバスによって相互に接続され、
かつ前記バス制御手段と、前記データ入出力手段とが第３のバスによって相互に接続され、
前記バス制御手段は前記第１のバスと前記第２のバスを前記第３のバスに接続するかしないかを選択し、前記第１のデータ処理装置の前記第１のバスを使用した読み出しと、前記第２のデータ処理装置の前記第２のバスを使用した読み出しとを並行して行うことが可能であって、
前記半導体集積回路装置は外部バスに接続され、前記第１のバスを介して前記第１のデータ処理装置から外部へデータを出力し、または前記外部バスを介して前記第１のデータ処理装置にデータを入力する制御を可能とし、
前記データ入出力手段はレジスタ手段を含み、前記データ入出力手段に含まれるレジスタ手段の少なくとも１つのレジスタ手段は複数のビットを持ち、別のレジスタ手段の状態によって前記第１、第２のデータ処理装置のいずれから書込み可能とするかを選択可能と

することを特徴とする半導体集積回路装置。
【請求項２】
請求項１記載の半導体集積回路装置であって、
前記データ入出力手段は割込要求可能とされ、
更に、割込制御手段を有し、前記データ入出力手段が出力する割込信号を前記割込制御手段に入力し、前記割込制御手段は前記第１、第２のデータ処理装置にそれぞれ第１、第２の割込要求信号を出力し、
前記割込制御手段は、データ処理装置選択手段、第１、第２の割込優先順位判定手段、第１、第２のマスク制御手段を有し、前記割込信号が活性状態となることに呼応して、前記第１、第２の割込要求信号のいずれを活性状態とするかを選択可能とされ、
前記データ処理装置選択手段は、レジスタの設定に従って、所定の割込信号を前記第１、第２の割込優先順位判定手段のいずれに要求するかを選択し、
前記第１、第２の割込優先順位判定手段は、入力された割込信号の優先順位を判定し、判定結果を、それぞれ、前記第１、第２のマスク制御手段に出力し、
前記第１、第２のマスク制御手段は、前記第１、第２のデータ処理装置の割込マスク状態にて、それぞれ、前記第１、第２のデータ処理装置に割込を要求するとともに、少なくとも１つの割込は、前記データ処理装置選択手段を介さず、前記第１、第２の割込優先順位判定手段に供給されることを特徴とする半導体集積回路装置。
【請求項３】
請求項１または２記載の半導体集積回路装置であって、
前記第２のデータ処理装置が書込み可能なレジスタ手段を有し、前記レジスタ手段の状態によって前記第１のデータ処理装置に割込を要求することを可能とすることを特徴とする半導体集積回路装置。
【請求項４】
請求項１、２または３記載の半導体集積回路装置であって、
割込を要求する端子を有し、前記端子に所定の信号変化が発生することに呼応して、前記第１、第２のデータ処理装置の両方に割込を要求することを可能とすることを特徴とする半導体集積回路装置。
【請求項５】
請求項４記載の半導体集積回路装置であって、
前記第１、第２のデータ処理装置のいずれから書込み可能とするかを選択可能とされるレジスタ手段は端子の状態を制御するものであることを特徴とする半導体集積回路装置。
【請求項６】
請求項１、２、３、４または５記載の半導体集積回路装置であって、
前記半導体集積回路装置の全体を停止する全体停止状態、前記第１のデータ処理装置を停止する第１の待機状態、前記第２のデータ処理装置を停止する第２の待機状態をそれぞれ設定する全体停止手段、第１の待機手段、第２の待機手段を有し、前記全体停止状態、前記第１の待機状態、前記第２の待機状態は前記第１、第２のデータ処理装置が所定の遷移命令を実行したことに呼応して遷移するものであって、前記第１のデータ処理装置が前記全体停止状態を設定する前記遷移命令を実行したとき、前記第２のデータ処理装置が前記第２の待機状態であれば前記全体停止状態に遷移し、前記第２のデータ処理装置が前記第２の待機状態でなければ前記第１の待機状態に遷移することを特徴とする半導体集積回路装置。
【請求項７】
請求項１、２、３、４、５または６記載の半導体集積回路装置であって、
テストモード設定手段と、バッファ手段とを有し、前記バッファ手段は、前記テストモード設定手段がテストモードを設定したときに、アドレス、リード信号またはライト信号、アドレス空間選択信号を入力し、前記リード信号またはライト信号に従ってデータを入出力するものであって、前記アドレス空間選択信号は、前記入力されたアドレスが前記第１、第２のデータ処理装置のいずれのアドレス空間に対応するかを指定するものであって

、前記アドレス空間選択信号と前記アドレスに従って前記読み出し可能な記憶手段または前記データ入出力手段が選択されて、前記リード信号またはライト信号に従って前記読み出し可能な記憶手段または前記データ入出力手段はリードまたはライトされることが可能とされることを特徴とする半導体集積回路装置。

【請求項８】

請求項１、２、３、４、５、６または７記載の半導体集積回路装置であって、

前記第１、第２の読み出し可能な記憶手段は第１、第２の電気的に書込み可能な記憶手段であって、書込みモード設定手段と、バッファ手段とを有し、前記バッファ手段は、前記書込みモード設定手段が書込みモードを設定したときに、アドレス、データおよび制御信号を入力し、前記制御信号に従って前記第１、第２の電気的に書込み可能な記憶手段に書込みを行うものであって、前記第１、第２の電気的に書込み可能な記憶手段が連続的に書込み可能なアドレスに配置されることを特徴とする半導体集積回路装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】

本発明は、半導体集積回路装置に関し、たとえばシングルチップマイクロコンピュータにおける多数の割込によるデータ転送および使用者毎の固有の処理を可能とし、かつ処理効率を向上させることができ、さらにこのシングルチップマイクロコンピュータのデバック効率の向上が可能な半導体集積回路装置およびそのエミュレーション用プロセッサならびにエミュレータに適用して有効な技術に関する。

【０００２】

【従来の技術】

たとえば、シングルチップマイクロコンピュータは、昭和５９年１１月３０日、オーム社発行の『ＬＳＩハンドブック』Ｐ５４０およびＰ５４１に記載されるように、中央処理装置（ＣＰＵ）を中心にしてプログラム保持用のＲＯＭ（リードオンリメモリ）、データ保持用のＲＡＭ（ランダムアクセスメモリ）、およびデータの入出力を行うための入出力回路などの機能ブロックが一つの半導体基板上に形成されてなる。

【０００３】

このようなシングルチップマイクロコンピュータに、ダイレクトメモリアクセスコントローラ（ＤＭＡＣ）を内蔵し、ＣＰＵとは独立にデータ転送を内蔵したものに、平成５年３月、（株）日立製作所発行『Ｈ８／３００３　ハードウェアマニュアル』または特願平４－１３７９５４号がある。このＤＭＡＣは、割込要求によって起動でき、リピートモード、ブロック転送モードなどを行うことができ、ステッピングモータの制御やプリンタの印字データ制御に好適である。たとえば、最大８チャネルの転送を行うことができる。

【０００４】

また、ＤＭＡＣの転送はＣＰＵと独立しているために、データ転送に必要なサイクル数のみＣＰＵを停止するのみでよく、ＣＰＵは実行中の処理を継続できる。たとえば、バイトデータの転送、転送元・転送先アドレスをインクリメントする場合、前記の例では、デッドサイクル１ステートを含めてデータ転送６ステートである。

【０００５】

【発明が解決しようとする課題】

ところが、前記のようなＤＭＡＣにおいては、転送元アドレス、転送先アドレス、転送カウンタ、制御レジスタを各チャネルに持つために、多数のチャネルのデータ転送を行おうとすれば、前記レジスタを多数組み設けねばならず、ＤＭＡＣの、ひいては半導体集積回路装置全体の論理的・物理的規模を増大させてしまう。さらに、規模の増大は製造費用を増大させてしまう。すなわち、割込要因の全て、あるいは大多数のデータ転送を行うことは困難である。

【０００６】

一方、このようなレジスタを、記憶密度が高い汎用のＲＡＭ上に配置し、論理的・物理的規模の増大を防いだデータ転送装置、いわゆるデータトランスファコントローラ（ＤＴＣ

）を内蔵したものに、昭和６３年１２月、（株）日立製作所発行『Ｈ８／５３２　ハードウェアマニュアル』などがある。このＤＴＣでは、実質的に全ての割込要因によってデータ転送を行うことができる。

【０００７】

しかしながら、転送情報を保持するレジスタを汎用のＲＡＭ上に配置したために、データ転送に先だって、このレジスタをＤＴＣに読み込み、この読み込んだ内容に従ってデータ転送を行い、さらにデータ転送によって更新されたレジスタ情報をＲＡＭに退避する必要がある。この期間はＣＰＵを停止しなければならず、データ転送に比較して、大きな時間をレジスタの読み込み・退避に費やしてしまう。

【０００８】

このため、シングルチップマイクロコンピュータまたは半導体集積回路装置全体の処理効率を低下させてしまう。たとえば、バイトデータの転送、転送元・転送先アドレスをインクリメントする場合、前記の例では、データ転送５ステートに対してレジスタの読み込み・退避に３０ステートを要し、合計３５ステートの期間ＣＰＵは停止することになる。

【０００９】

また、ＤＭＡＣあるいはＤＴＣの動作の制御は、制御レジスタの制御ビットの内容によって指定されるものであり、転送モードや起動要因を選択できるのみである。この制御ビットのビット数を増やせば、転送データの加算（チェックサム）や、パリティ計算などを行うことも可能であるが、使用者毎に固有の処理を行わせることは困難である。また、多様な処理を実現しようとすれば、これを制御する回路の論理的・物理的規模が増加してしまう。

【００１０】

当然、ＣＰＵを利用して、データ転送に付随する使用者毎の固有の処理を行うことができる。たとえば、割込が発生したときに、主たる処理を中断して所望の処理を行うものである。しかしながら、ＣＰＵが行うべき主たる処理を中断しなければならなくなってしまい、またＣＰＵの内部状態の退避・復帰を行わなければならないために、オーバヘッドが発生し、シングルチップマイクロコンピュータ全体の処理性能を向上することができなくなってしまう。

【００１１】

そこで、本発明の目的は、多数の割込によるデータ転送および使用者毎の固有の処理を可能とし、かつ処理効率を向上させることができるシングルチップマイクロコンピュータまたは半導体集積回路装置を提供することにある。

【００１２】

さらに、このシングルチップマイクロコンピュータのデバック効率を向上させることができる半導体集積回路装置のエミュレーション用プロセッサならびにエミュレータを提供するものである。

【００１３】

本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【００１４】

【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、以下のとおりである。

【００１５】

すなわち、本発明の半導体集積回路装置は、第２のデータ処理装置を設け、第１のデータ処理装置の使用する第１のバスと第２のデータ処理装置の使用する第２のバスを分離可能とし、第１のデータ処理装置のバス使用と第２のデータ処理装置のバス使用を同時に行うことを可能とし、さらに第１のデータ処理装置と第２のデータ処理装置のそれぞれの動作に必要な機能ブロックまたはモジュールをそれぞれの第１または第２のバスによって接続し、特に第１のデータ処理装置と第２のデータ処理装置が実行すべきプログラムを格納可

能な第１の読み出し可能な記憶手段と第２の読み出し可能な記憶手段を設け、第１のデータ処理装置が第１の読み出し可能な記憶手段から、第２のデータ処理装置が第２の読み出し可能な記憶手段からプログラムを読み出して動作するようにしたものである。

【００１６】

また、第１、第２のデータ処理装置と、これらに共通の読み出し可能な記憶手段をバスによって相互に接続し、第１のデータ処理装置は、単位時間に実行可能な命令より大きい量の命令をバスを使用して記憶手段から読み出し、連続的にバスを使用して記憶手段を読み出すことなく、第１のデータ処理装置の命令実行速度の低下を最小限として第２のデータ処理装置が命令を記憶手段から読み出すようにしたものである。

【００１７】

この場合に、割込要因のうち、第１の要因は第１のデータ処理装置に、第２の要因は第２のデータ処理装置に要求可能にするものであり、第１のデータ処理装置のアドレス空間は第２のデータ処理装置のアドレス空間を包含するようにするとよく、また入出力手段は第１のデータ処理装置と第２のデータ処理装置からリード／ライト可能とし、２つのデータ処理装置の動作を調停する機能を設けるようにしたものである。

【００１８】

また、この半導体集積回路装置を評価するためのエミュレーション用プロセッサは、第１、第２のデータ処理装置が排他的に動作するバスをいずれのデータ処理装置が使用しているかを示す信号を出力し、第１、第２のデータ処理装置を停止するための第１、第２のブレーク割込入力端子を有するものである。

【００１９】

さらに、このプロセッサを搭載したエミュレータは、第１、第２のデータ処理装置の動作情報を同時に記憶手段に蓄積するか、一方のデータ処理装置の動作情報を選択的に記憶手段に蓄積するかをシステム開発装置からの指定に基づいて選択する手段を有したり、または第１、第２のデータ処理装置にブレーク割込を要求する条件をシステム開発装置からの指定に基づいて独立に設定する手段を有するものである。

【００２０】

【作用】

前記した半導体集積回路装置によれば、２つのデータ処理装置に並立したバスによって独立した２つの読み出し可能な記憶手段に格納した命令のリード可能にしたり、バスの幅を拡張してデータ処理装置が単位時間に実行可能な命令数以上の命令をリード可能にしたり、２つのデータ処理装置を並立的に動作可能な命令を供給する機能を内蔵することにより、複数の記憶手段を同時に選択可能とし、複数の記憶手段の同時書込みまたは同時読み出しを行うことによって割込によるデータ処理のチャネル数の制約をなくし、さらにデータ処理内容を任意とすることができる。

【００２１】

また、第１、第２のデータ処理装置が同時に動作することにより、シングルチップマイクロコンピュータまたは半導体集積回路装置の処理速度を向上することができる。

【００２２】

さらに、エミュレーション用プロセッサ、およびこれを搭載したエミュレータによれば、第１のブレーク割込によって第１のデータ処理装置を停止し、また第２のブレーク割込によって第２のデータ処理装置を停止でき、このように２つのデータ処理装置を独立に停止する割込を有することにより、デバック効率を向上することができる。

【００２３】

【実施例】

以下、本発明の実施例を図面に基づいて詳細に説明する。

【００２４】

（実施例１）

図１は本発明の一実施例であるシングルチップマイクロコンピュータを示すブロック図、図２は本実施例のシングルチップマイクロコンピュータにおける全体のアドレスマップの

説明図、図３はシングルチップマイクロコンピュータのバスの接続状態のブロック図、図４は割込コントローラの概略ブロック図、図５は割込要求ビットおよび割込許可回路の概略ブロック図、図６はバスの動作タイミング図、図７は低消費電力状態制御回路の概略ブロック図、図８は入出力ポートを制御するレジスタ構成の説明図、図９は入出力ポートの概略ブロック図、図１０はＰＲＯＭのアドレスマップ、図１１はＰＲＯＭモードの主要部のブロック図、図１２はテストモードの主要部のブロック図、図１３はエミュレーション用プロセッサの概略ブロック図、図１４はエミュレータの概略ブロック図、図１５は本実施例におけるシングルチップマイクロコンピュータの変形例を示すブロック図である。

【００２５】

まず、図１により本実施例のシングルチップマイクロコンピュータの構成を説明する。

【００２６】

本実施例のシングルチップマイクロコンピュータは、中央処理装置（ＣＰＵ１：第１のデータ処理装置、ＣＰＵ２：第２のデータ処理装置）、リードオンリメモリ（ＲＯＭ１：第１の読み出し可能な記憶手段、ＲＯＭ２：第２の読み出し可能な記憶手段）、ランダムアクセスメモリ（ＲＡＭ１、ＲＡＭ２、ＲＡＭＰ）、タイマ、パルス出力回路、シリアルコミュニケーションインタフェース（ＳＣＩ）、Ａ／Ｄ変換器（Ａ／Ｄ）、入出力ポート（ＩＯＰ０～ＩＯＰ１１：データ入出力手段）、割込コントローラ（割込制御手段）、バスコントローラ（バス制御手段）、クロック発振器の機能ブロック（またはモジュール）から構成され、一つの半導体基板上に形成されている。

【００２７】

これらの機能ブロックは、内部バスによって相互に接続される。内部バスはアドレスバス、データバスの他、リード信号、ライト信号を含み、さらにバスサイズ信号あるいはシステムクロックを含んでもよい。この内部アドレスバスには、ＩＡＢ１（第１のバス）、ＩＡＢ２（第２のバス）、ＰＡＢ（第３のバス）が存在し、また内部データバスにはＩＤＢ１（第１のバス）、ＩＤＢ２（第２のバス）、ＰＤＢ（第３のバス）が存在する。この内部データバスＩＤＢ１、ＩＤＢ２、ＰＤＢは１６ビット構成である。また、ＣＰＵ１、ＣＰＵ２の命令は１６ビット単位としている。

【００２８】

なお、ＣＰＵ１、ＲＯＭ１、ＲＡＭ１はＩＡＢ１、ＩＤＢ１に接続され、ＣＰＵ２、ＲＯＭ２、ＲＡＭ２はＩＡＢ２、ＩＤＢ２に接続されている点で相違される。また、ＲＡＭＰはＰＡＢ、ＰＤＢに接続される。このＩＡＢ１、ＩＡＢ２、ＰＡＢはバスコントローラでインタフェースされ、同様にＩＤＢ１、ＩＤＢ２、ＰＤＢはバスコントローラでインタフェースされる。

【００２９】

また、ＰＡＢ、ＰＤＢはバスコントローラ、ＲＡＭＰ、タイマ、パルス出力回路、ＳＣＩ、Ａ／Ｄ変換器、割込コントローラ、ＩＯＰ０～１１に接続される。ＣＰＵ１、ＣＰＵ２のいずれか一方がＰＡＢ、ＰＤＢを使用して、これらをリード／ライトできる。割込コントローラは、タイマ、ＳＣＩ、入出力ポートの出力する割込信号を入力し、ＣＰＵ１またはＣＰＵ２に割込要求信号とベクタ番号を出力する。

【００３０】

入出力ポートは、外部バス信号、入出力回路の入出力信号と兼用とされている。ＩＯＰ０～３はアドレスバス出力、ＩＯＰ４、５はデータバス入出力、ＩＯＰ６はバス制御信号入出力信号と兼用されている。外部アドレス、外部データは、それぞれこれらの入出力ポートに含まれるバッファ回路を介してＩＡＢ１、ＩＤＢ１と接続されている。ＰＡＢ、ＰＤＢは入出力ポートのレジスタをリード／ライトするために使用し、外部バスとは直接の関係はない。

【００３１】

このバス制御信号出力は、アドレスストローブ、ハイ／ロウデータストローブ、リードストローブ、ライトストローブ、バスアクノリッジ信号などがあり、またバス制御入力信号にはウェイト信号、バスリクエスト信号などがある。これらの入出力信号は図示はされな

い。外部バス拡張を行うことは、動作モードなどで選択され、これらの入出力ポートの機能も選択される。

【００３２】

また、ＩＯＰ７はタイマ入出力、ＩＯＰ８はパルス出力、ＩＯＰ９はＳＣＩ入出力、ＩＯＰ１０はアナログ入力、ＩＯＰ１１は外部割込要求（ＩＲＱ）入力と兼用されている。なお、タイマ、ＳＣＩ、Ａ／Ｄ変換器とＩＯＰ７、ＩＯＰ９、ＩＯＰ１０との入出力信号は図示はされない。

【００３３】

そのほか、電源端子Ｖｃｃ、Ｖｓｓ、アナログ電源端子ＡＶｃｃ、ＡＶｓｓ、リセット入力ＲＥＳ、スタンバイ入力ＳＴＢＹ、割込入力ＮＭＩ、クロック入力ＥＸＴＡＬ、ＸＴＡＬ、動作モード入力ＭＤ０、ＭＤ１、ＭＤ２などの入力端子がある。

【００３４】

パルス出力回路は、タイマから与えられるトリガ信号に従って、図示はされないレジスタ（ＮＤＲ）に保持した内容をＩＯＰ８と兼用の端子から出力する。

【００３５】

次に、図２によりシングルチップマイクロコンピュータのアドレスマップを説明する。

【００３６】

たとえば、ＣＰＵ１は４Ｇバイトのアドレス空間、ＣＰＵ２は６４ｋバイトのアドレス空間を持つ。また、ＲＯＭ１、ＲＡＭ１はＣＰＵ１のアドレス空間上にのみ、ＲＯＭ２、ＲＡＭ２はＣＰＵ２のアドレス空間上にのみ存在する。斜線部で示されるＲＡＭＰ、およびタイマ、パルス出力回路、ＳＣＩ、Ａ／Ｄ変換器、ＩＯＰ０～１１、割込コントローラの各機能ブロックのアドレス、すなわち内部レジスタは、１つの物理的アドレスに対し、それぞれのＣＰＵのアドレス空間上で固有の論理的アドレスを有する。

【００３７】

このアドレスは、それぞれアドレスで対応付けられており、具体的にはＣＰＵ２のアドレスの上位にＨ’ＦＦＦＦを付加したアドレスがＣＰＵ１のアドレスになる。たとえば、ＲＡＭＰの先頭アドレスはＣＰＵ１ではＨ’ＦＦＦＦＦＢ８０、ＣＰＵ２ではＨ’ＦＢ８０である。

【００３８】

すなわち、ＣＰＵ１はＲＯＭ１、ＲＡＭ１、ＲＡＭＰ、内部レジスタをリード／ライトでき、ＣＰＵ２はＲＯＭ２、ＲＡＭ２、ＲＡＭＰ、内部レジスタをリード／ライトできる。

【００３９】

次に、図３により、このシングルチップマイクロコンピュータのアドレスバス、データバスの接続状態を説明する。

【００４０】

図３において、Ｉ／Ｏには図１のタイマ、パルス出力回路、ＳＣＩ、Ａ／Ｄ変換器、ＩＯＰ０～１１、割込コントローラの各機能ブロックを含む。クロック発振器はバスと接続されていないので省略する。

【００４１】

アドレスバス、データバスとして、ＣＰＵ１、ＲＯＭ１、ＲＡＭ１、外部を接続するＩＡＢ１、またＣＰＵ２、ＲＯＭ２、ＲＡＭ２を接続するＩＡＢ２、さらにＲＡＭＰ、Ｉ／Ｏを接続するＰＡＢがある。このようなアドレスバス、データバスはバスコントローラ（ＢＳＣ）でインタフェースされている。

【００４２】

たとえば、ＣＰＵ１がＲＯＭ１からプログラムをリードする場合、またはＲＡＭ１とのデータのリード／ライトする場合、ＩＡＢ１、ＩＤＢ１を使用して動作が行われる。このＣＰＵ１はＰＡＢ、ＰＤＢを使用しない。

【００４３】

一方、ＣＰＵ２がＲＯＭ２からプログラムをリードする場合、またはＲＡＭ２とのデータのリード／ライトする場合、ＩＡＢ２、ＩＤＢ２を使用して動作が行われる。このＣＰＵ

２はＰＡＢ、ＰＤＢを使用しない。このとき、ＣＰＵ１とＣＰＵ２は互いに独立したバスを用いて動作し、並列に動作することができる。

【００４４】

また、ＣＰＵ１とＣＰＵ２のいずれか一方が、ＲＡＭＰまたはＩ／Ｏとのデータのリード／ライトを行う場合、バス権調停が行われる。この場合に、ＣＰＵ２によるバス権要求がＣＰＵ１によるバス権要求より優先される。このバス権調停は、ＩＡＢ１／ＩＡＢ２に示されるアドレスとＣＰＵ１／２が出力するコマンド信号によって、バスコントローラが行う。コマンド信号にはリード要求、ライト要求、サイズ信号等を含む。

【００４５】

一方がＲＡＭＰまたはＩ／Ｏとのデータのリード／ライト中に、他方がＲＡＭＰまたはＩ／Ｏとのデータのリード／ライトを行おうとすると、後者は前者のリード／ライトが終了するまで、バスコントローラから待機信号を与えられ、待機状態とされる。たとえば、ＣＰＵ２がＲＡＭＰまたはＩ／Ｏとのデータのリード／ライトを行っている間、ＣＰＵ１はＲＯＭ１、ＲＡＭ１のリード／ライトを行うことができる。

【００４６】

さらに、ＣＰＵ１とＣＰＵ２が、同時にＲＡＭＰまたはＩ／Ｏとのデータのリード／ライトを行おうとした場合、一方のＣＰＵは停止状態にすることになるが、プログラムをＲＡＭＰまたはＩ／Ｏには配置しないことによって、ＲＡＭＰまたはＩ／Ｏとのデータのリード／ライトをデータのみに限定して頻度を低下することができる。

【００４７】

この場合に、リードした命令に基づいてデータのリード／ライトを行うのであるから、命令のリードとデータのリード／ライトの回数の比は１／２以上にはならない。また、データのリード／ライト命令で全てのプログラムが構成されることはないから、前記の比はさらに小さくできる。一般的には１／４～１／５程度であると考えられる。この実施例中に記載されるプログラムによっても同様であることが示される。従って、ＣＰＵ１とＣＰＵ２の並列動作性を高めることができる。

【００４８】

このような並列動作を有効に利用するためには、ＣＰＵ１が全体的な設定と、ＲＯＭ１、ＲＡＭ１あるいは外部のデバイスを用いた主たるプログラム処理を行い、ＣＰＵ２がＩ／Ｏの逐次的な制御を行うようにするとよい。また、ＣＰＵの命令をＲＯＭ１またはＲＯＭ２に配置し、作業領域をＲＡＭ１またはＲＡＭ２に配置するとよい。

【００４９】

次に、図４により割込コントローラを詳細に説明する。

【００５０】

たとえば、割込要因には内部割込、外部割込の２種類があり、それぞれ割込要因フラグを有する。この内部割込の要因フラグは、タイマ、ＳＣＩ、Ａ／Ｄ変換器の入出力回路が所定の状態になったときに“１”にセットされ、また外部割込の要因フラグは、外部割込入力端子が所定のレベルになったとき、または所定の信号変化が発生したときに“１”にセットされる。この割込要因フラグは、ＣＰＵ１／２のライト動作によって“０”にクリアされる。

【００５１】

割込要因フラグの各ビットの出力は、割込許可回路に入力され、この割込許可回路には、さらに割込許可レジスタの内容、すなわち割込許可ビットが入力される。この割込許可レジスタは、ＣＰＵ１／２からリード／ライト可能なレジスタで、対応する割込を許可するか、禁止するかを選択する。

【００５２】

たとえば、割込要因フラグが“１”にセットされ、割込許可ビットが“１”にセットされていると、割込が要求される。すなわち、割込許可回路は、対応する割込要因フラグと割込許可ビットとを入力とした論理積回路で構成される。

【００５３】

割込許可回路の出力は、ＣＰＵ選択回路に入力され、このＣＰＵ選択回路には、さらに割込選択レジスタＩＲＱＳＥＬの内容が入力される。この割込選択レジスタは、割込が要求されたとき、ＣＰＵ１またはＣＰＵ２のいずれに割込を許可するかを選択する。

【００５４】

たとえば、割込選択レジスタのビットが“１”にセットされていると、ＣＰＵ２に割込が要求される。割込選択レジスタのビットが“０”にクリアされていると、ＣＰＵ１の割込が要求される。すなわち、ＣＰＵ選択回路は、対応する割込信号と割込選択ビットの論理積回路、および割込信号と割込選択ビットの反転信号の論理積回路で構成される。前者の論理積回路の出力がＣＰＵ２割込要求信号とされ、後者の論理積回路の出力がＣＰＵ１割込要求信号とされる。

【００５５】

ＣＰＵ選択回路の出力は、ＣＰＵ１割込要求とＣＰＵ２割込要求が独立に、優先順位判定回路に入力され、この優先順位判定回路には、さらにプライオリティレジスタの出力が入力される。このプライオリティレジスタは割込要因のグループ毎に、たとえば２レベルの優先順位を設定する。

【００５６】

この場合に、ＣＰＵ１／２割込要求のそれぞれについて優先順位を判定する。この判定の結果、優先順位の最高のものが選択され、ベクタ番号が生成され、ＣＰＵ１／２割込要求のそれぞれのベクタ番号が出力される。

【００５７】

このＣＰＵ１／２割込要求信号、ベクタ番号はマスクレベル判定回路に入力され、このマスクレベル判定回路には、さらに対応するＣＰＵ１／２の割込マスクビットが入力される。ここで、要求された割込がＣＰＵの割込マスクレベル以下であれば保留される。そして、ＣＰＵ１／２のそれぞれに対し、割込要求信号とベクタ番号が出力される。なお、ＣＰＵ１は２ビットのマスクビットを持つが、ＣＰＵ２は１ビットのマスクビットを持つ。

【００５８】

さらに、それぞれのＣＰＵに対する割込要求信号が活性状態になると、このＣＰＵは実行中の命令の終了時点で割込例外処理を開始し、ベクタ番号に対応するベクタアドレスから分岐先アドレスを取り出し、割込処理ルーチンへ分岐する。

【００５９】

このような優先順位判定や割込マスクレベルについては、前記平成５年３月、（株）日立製作所発行『Ｈ８／３００３　ハードウェアマニュアル』または特願平４－１３７９５５号などによって公知であるので、詳細な説明は省略する。

【００６０】

また、ＣＰＵ１またはＣＰＵ２のいずれに要求されるかによらず、ベクタ番号は一定である。しかし、アドレス空間の大きさに対応して、ベクタアドレスは、ＣＰＵ１では４バイト単位、ＣＰＵ２では２バイト単位とされる。

【００６１】

ノンマスカブル割込ＮＭＩは、１本の端子ＮＭＩから入力されるが、２つのＣＰＵに同時に割込を要求できる。これは対応する割込許可ビットを持たない。このＮＭＩ割込は、マスク不可能な特性上、シングルチップマイクロコンピュータまたは半導体集積回路の暴走時などに、ＣＰＵの処理を強制的に停止させるために用いるため、１本のＮＭＩ端子によって、ＣＰＵ１およびＣＰＵ２の両方にＮＭＩ割込を要求することは都合がよい。少なくとも２本の端子にするのに比較して、端子１本を節約して、入出力ポートなどとして有効に使用することができる。

【００６２】

また、割込要因フラグには、ＣＰＵ２がライト可能な割込要求ビットを持つ。この割込許可ビットを“１”にセットすると、ＣＰＵ１に割込を要求できる。また、割込要求ビットはＣＰＵ２から“１”にセットすることが可能であり、ＣＰＵ１からは“０”にクリアすることができる。ＣＰＵ１が“０”にクリアする場合は、事前に“１”の状態をリードし

た後、“０”をライトする必要がある。
【００６３】
次に、割込コントローラにおいて、図５により割込要求ビットと割込許可回路を詳細に説明する。
【００６４】
たとえば、２つのフリップフロップＦＦ１、ＦＦ２を含む。このＦＦ１は、割込要求ビットを構成し、クリア信号ＣＬＲとしてリセット信号ＲＥＳＥＴが入力される。また、セット信号Ｓには論理積回路ＡＮＤ１の出力が入力され、リセット信号Ｒには論理積回路ＡＮＤ３Ａの出力が入力され、出力Ｑは、論理積回路ＡＮＤ４を介して割込要求として出力されるほか、論理積回路ＡＮＤ２、出力バッファに入力される。
【００６５】
論理積回路ＡＮＤ１には、ＣＰＵ２のバス権信号、レジスタライト信号、および内部データバスの所定のビットが入力される。なお、レジスタリード／レジスタライト信号は、リード信号／ライト信号と、アドレスをデコードしてこのレジスタが選択されたことを示す信号との論理積で生成される信号とする。たとえば、レジスタライト信号は、ＣＰＵ２またはＣＰＵ１がこの割込要求ビットのアドレスにライトしたとき、活性状態“１”となる。
【００６６】
ＦＦ２は、割込要求ビットの制御を行い、クリア信号ＣＬＲとしてリセット信号ＲＥＳＥＴが入力される。また、セット信号Ｓには論理積回路ＡＮＤ２の出力が入力され、リセット信号Ｒには論理積回路ＡＮＤ３Ａの出力が入力され、出力Ｑは論理積回路ＡＮＤ３Ａに入力される。
【００６７】
論理積回路ＡＮＤ２には、ＦＦ１の出力、ＣＰＵ１のバス権信号、およびレジスタリード信号が入力される。
【００６８】
論理積回路ＡＮＤ３Ａには、ＦＦ２の出力、および内部データバスの所定のビットの反転、およびＡＮＤ３Ｂの出力が入力される。このＡＮＤ３Ｂには、ＣＰＵ１のバス権信号、レジスタライト信号が入力される。
【００６９】
論理積回路ＡＮＤ４には、ＦＦ１の出力、および、図示はされない割込許可レジスタの所定のビットの出力である割込許可信号を入力し、出力は割込要求信号とされる。なお、選択ビットは設けられず、必ずＣＰＵ１に割込を要求するとよい。
【００７０】
出力バッファは、ＦＦ１の出力をデータとして入力し、レジスタリード信号をクロック信号として入力する。この出力は内部データバスの所定のビットに結合される。
【００７１】
従って、ＣＰＵ２がこの割込要求ビットのアドレスに“１”をライトしたとき、ＡＮＤ１の出力が“１”となって、ＦＦ１が“１”にセットされる。このように、割込許可レジスタの所定のビットが“１”にセットされていると、ＣＰＵ１に割込が要求される。
【００７２】
さらに、ＣＰＵ１がこの状態で割込要求ビットのアドレスをリードすると、出力バッファを介してデータバスに割込要求ビットの内容を出力するとともに、ＡＮＤ２の出力が“１”となって、ＦＦ２が“１”にセットされる。
【００７３】
続いて、ＣＰＵ１が割込要求ビットに“０”をライトすると、ＡＮＤ３Ａの出力が“１”となって、ＦＦ１およびＦＦ２が“０”にクリアされる。同様に、ＣＰＵ１がライト可能な割込要求ビットを持ち、この割込許可ビットはＣＰＵ２からのみライトでき、ＣＰＵ２に割込を要求できる。
【００７４】

いわゆるリードモディファイライト命令では、リードとライトの間にバスを解放しないようにするとよい。たとえば、ビットセット命令は、指定されたアドレスのデータをバイトサイズでＣＰＵに取込み、ＣＰＵ内部で指定された１ビットを“１”にセットし、バイトサイズでデータをライトするものである。

【００７５】

このリードとライトの間はＰＡＢ、ＰＤＢを他方のＣＰＵに解放しないようにする。このバスの解放を禁止する信号は、ＣＰＵ１／２がバスコントローラに出力する前記コマンド信号に含めることができる。

【００７６】

次に、図６により内部バスの動作タイミングの一例を説明する。

【００７７】

本実施例のシングルチップマイクロコンピュータは、たとえばシステムクロックφに同期して動作し、ＩＡＢ１／２はφの反転信号であるφ＃に同期して出力され、またＣＰＵ１／２のＲＯＭ１／２およびＲＡＭ１／２に対するリードは１ステートで行われる。

【００７８】

まず、ＩＡＢ１／２はφ＃に同期して１ステート出力され、特に制限はされないものの、ＲＯＭ１／２およびＲＡＭ１／２の中でφに同期してラッチされる。これに対応するリードデータはφ＃に同期して出力され、φ＃が活性状態の期間にＣＰＵに取り込まれる。たとえば、Ｔ１のφ＃に同期してＩＡＢ１／２に出力されたアドレスに対するデータはＴ２のφ＃が活性状態の期間に、それぞれＣＰＵ１／２に取り込まれる。

【００７９】

一方、ＲＡＭＰに対するリード／ライトは２ステート、Ｉ／Ｏに対するリード／ライトは３ステートで行われる。このφ＃に同期したＩＡＢ１／２は、バスコントローラでφに同期化される。

【００８０】

この場合に、ＣＰＵ１のプログラムは以下の通りとする。なお、命令の表記方法は、平成元年７月、（株）日立製作所発行『Ｈ８／３００シリーズ　プログラミングマニュアル』に記載のＣＰＵと同様とする。

【００８１】

```
ＣＭＰ．Ｂ　＃ＣＯＤＥ，Ｒ０Ｌ
ＢＥＱ　　Ｎ１
ＢＴＳＴ　　＃０，＠ＰＯＲＴ
ＢＮＥ　　Ｎ２
```

たとえば、ＣＭＰ、ＢＥＱ、ＢＮＥ命令は２バイト長、ＴＳＴ命令は４バイト長とする。

【００８２】

始めに、アドレスｎからＣＭＰ命令をリードする。このＣＭＰ命令の実行を開始すると、まずアドレスｎ＋２からＢＥＱ命令をリードするとともに、内部でイミディエイトデータＣＯＤＥと汎用レジスタＲ０Ｌの内容を比較して、コンディションコードに反映する。

【００８３】

さらに、ＢＥＱ命令の実行を開始すると、まず、アドレスｎ＋４からＢＴＳＴ命令をリードする。次に、分岐先のアドレスＮ１から命令をリードするとともに、前記比較結果が一致しているかを判定する。一致していなければ、アドレスＮ１の命令は無視して、すでにリード済みのＢＴＳＴ命令の実行を行う。

【００８４】

続いて、ＢＴＳＴ命令の第２ワードをリードする。このＢＴＳＴ命令を実行すると、絶対アドレスＰＯＲＴで指定されるＩ／Ｏから、ＰＡＢ／ＰＤＢを使用して、バイトサイズでデータをリードする。次にアドレスｎ＋６からＢＮＥ命令をプリフェッチする。

【００８５】

また、ＣＰＵ２のプログラムは以下の通りとする。

【００８６】

ＢＣＬＲ　＃ＲＤＲＦ，@ＳＳＲ
たとえば、ＢＣＬＲ命令は４バイト長とする。
【００８７】
始めに、アドレスｍ、アドレスｍ＋２からＢＣＬＲ命令をリードする。このＢＣＬＲ命令の実行を開始すると、まず絶対アドレスＳＳＲで指定されるＩ／Ｏから、ＰＡＢ／ＰＤＢを使用してバイトサイズでデータをリードする。
【００８８】
続いて、アドレスｍ＋４から次の命令をプリフェッチするとともに、内部でリードデータのイミディエイトデータＲＤＲＦで指定されるビットを“０”にクリアする。このデータを、絶対アドレスＳＳＲで指定されるＩ／Ｏへ、ＰＡＢ／ＰＤＢを使用してライトする。次の命令の実行を開始し、さらにアドレスｍ＋６から次の命令をプリフェッチする。
【００８９】
Ｔ１では、ＣＰＵ１／２のＲＯＭ１／２からの命令フェッチが行われ、ＩＡＢ１／２にアドレスが出力される。
【００９０】
Ｔ２では、ＲＯＭ１／２からＩＤＢ１／２に読み出された命令コード（ＣＭＰ命令、ＢＣＬＲ命令の第１ワード）をＣＰＵ１／２がそれぞれ取り込むとともに、次の命令フェッチのアドレスがＩＡＢ１／２に出力される。
【００９１】
Ｔ３では、ＲＯＭ１／２からＩＤＢ１／２に読み出された命令コード（ＢＥＱ命令、ＢＣＬＲ命令の第２ワード）をＣＰＵ１／２がそれぞれ取り込むとともに、ＣＰＵ１は次の命令フェッチのアドレスを、ＣＰＵ２はアドレスＳＳＲを、それぞれＩＡＢ１／２に出力する。
【００９２】
この場合に、ＰＡＢ／ＰＤＢを使用するのがＣＰＵ２だけなので、ＣＰＵ２のＰＡＢ／ＰＤＢの使用が許可される。ここで、ＳＣＩのリードが３ステート必要であるため、ＣＰＵ２待機信号がハイレベルになる。
【００９３】
Ｔ４では、バス権信号がロウレベルになって、ＣＰＵ２のＰＡＢ／ＰＤＢの使用を表示する。ＩＡＢ２の内容がＰＡＢに出力されて、ＳＣＩのレジスタＳＳＲの内容がリードされる。このリードデータはＴ６で得られる。
【００９４】
この場合に、ＲＯＭ１からＩＤＢ１に読み出された命令コード（ＢＴＳＴ命令の第１ワード）をＣＰＵ１が取り込むとともに、ＣＰＵ１は分岐先アドレスＮ１をＩＡＢ１に出力する（ＢＴＳＴ命令の第２ワード）。
【００９５】
Ｔ５では、ＲＯＭ１からＩＤＢ１に読み出された命令コードをＣＰＵ１が取り込むが、これは分岐条件が不成立で実行されない。先にフェッチ済みのＢＴＳＴ命令の第１ワードが有効になる。この場合に、ＣＰＵ１は次の命令フェッチのアドレスをそれぞれＩＡＢ１に出力する。
【００９６】
Ｔ６では、ＳＣＩのレジスタＳＳＲから読み出されたデータをＰＤＢ、ＩＤＢ２を介してＣＰＵ２が取り込む。また、ＣＰＵ２待機信号がロウレベルになり、次の命令フェッチのアドレスをＩＡＢ２に出力する。
【００９７】
この場合に、ＣＰＵ１はアドレスＰＯＲＴをＩＡＢ１に出力し、またＣＰＵ２のＰＡＢ／ＰＤＢの使用が終了しているので、ＣＰＵ１のＰＡＢ／ＰＤＢの使用が許可される。ここで、ＰＯＲＴのリードが３ステート必要であるため、ＣＰＵ１待機信号がハイレベルになる。
【００９８】

Ｔ７では、バス権信号がハイレベルになって、ＣＰＵ１のＰＡＢ／ＰＤＢの使用を表示する。ＩＡＢ１の内容がＰＡＢに出力されて、入出力ポートのレジスタＰＯＲＴの内容がリードされる。このリードデータはＴ９で得られる。

【００９９】

この場合に、ＣＰＵ１待機信号がロウレベルになり、次の命令フェッチのアドレスをＩＡＢ１に出力し、またＣＰＵ２はアドレスＳＳＲをＩＡＢ２に出力する。ここで、ＰＡＢ／ＰＤＢをＣＰＵ１が使用しているので、ＣＰＵ２の動作は保留され、ＣＰＵ２待機信号がハイレベルになる。

【０１００】

Ｔ８では、ＣＰＵ２はアドレスＳＳＲにライトすべきデータをＩＤＢ２に出力する。

【０１０１】

Ｔ９では、入出力ポートのレジスタＰＯＲＴから読み出されたデータをＰＤＢ、ＩＤＢ２を介してＣＰＵ１が取り込む。このＣＰＵ１は次の命令フェッチのアドレスをＩＡＢ１に出力する。この場合に、ＣＰＵ１待機信号がロウレベルになり、次の命令フェッチのアドレスをＩＡＢ１に出力する。

【０１０２】

Ｔ１０では、ＲＯＭ１からＩＤＢ１に読み出された命令コード（ＢＮＥ命令）をＣＰＵ１が取り込む。次の命令フェッチのアドレスをＩＡＢ１に出力する。この場合に、バス権信号がロウレベルになって、ＣＰＵ２のＰＡＢ／ＰＤＢの使用を表示する。ＩＡＢ２の内容がＰＡＢに出力されて、ＩＤＢ２の内容がＳＣＩのレジスタＳＳＲにライトされる。

【０１０３】

Ｔ１１、Ｔ１２でライト動作が行われる。この場合に、Ｔ１２でＣＰＵ２待機信号がロウレベルになり、次の命令フェッチのアドレスをＩＡＢ２に出力する。また、ＣＰＵ１は上記同様の動作を繰り返す。

【０１０４】

なお、ＣＰＵ１は、外部ＣＰＵとすることもでき、共通部分を外部のＣＰＵが周辺機能としてリード／ライトすることができる。また、ＣＰＵ２が共通部分をリード／ライトする場合には、外部ＣＰＵにウェイト信号を活性状態にして、待機させる。このとき、シングルチップマイクロコンピュータが外部ＣＰＵに対する割込信号を出力する。

【０１０５】

続いて、データ転送のプログラム例を以下に示す。ここでは、ＳＣＩの受信データのチェックサムを行い、結果は内蔵ＲＡＭ２上に配置した作業用アドレスＳＵＭに格納するものとする。また、受信データ数は内蔵ＲＡＭ２上に配置した作業用アドレスＳＵＭに格納されているものとする。

【０１０６】

その他、転送回数を記憶する作業用アドレスＣＮＴ、およびＳＣＩの受信データレジスタＲＤＲ、ステータスレジスタＳＳＲ、割込コントローラの割込選択レジスタＩＲＱＳＥＬを使用する。

【０１０７】

```
        PUSH   R0                    (1)
        MOV.B  @RDR, R0L             (2)
        MOV.B  @SUM, R0H             (3)
        ADD.B  R0L, R0H              (4)
        MOV.B  R0H, @SUM             (5)
        MOV.B  @CNT, R0H             (6)
        DEC.B  R0H                   (7)
        MOV.B  R0H, @CNT             (8)
        BEQ    L1                    (9)
        BCLR   #RDRF, @SSR           (10)
        BRA    L2                    (11)
L1:     BNOT   #RXI, @IRQSEL         (12)
L2:     RTE                          (13)
```

始めに、ＣＰＵ２に受信完了割込ＲＸＩが要求され、割込ルーチンに分岐する。まず、汎用レジスタＲ０をスタックに退避する（１）。そして、ＳＣＩの受信データレジスタＲＤＲの内容を汎用レジスタＲ０の下位８ビットＲ０Ｌにリードする（２）。

【０１０８】

また、内蔵ＲＡＭ２上に配置した作業用アドレスＳＵＭの内容を汎用レジスタＲ０の上位８ビットＲ０Ｈにリードする（３）。そして、Ｒ０ＨとＲ０Ｌの内容を加算する（４）。この加算結果をアドレスＳＵＭにライトする（５）。

【０１０９】

さらに、内蔵ＲＡＭ２上に配置した作業用アドレスＣＮＴの内容を汎用レジスタＲ０Ｈにリードする（６）。そして、Ｒ０Ｈの内容をデクリメント（－１）する（７）。このデクリメント結果をアドレスＣＮＴにライトする（８）。

【０１１０】

この内容が“０”であれば、Ｌ１に分岐し（９）、割込選択レジスタＩＲＱＳＥＬの受信完了割込ＲＸＩに対応するビットを反転する（１２）。このとき、割込要求フラグＲＤＲＦが“１”にセットされたままなので、ＣＰＵ１に受信完了割込ＲＸＩが要求される。そして、割込ルーチンから復帰する（１３）。

【０１１１】

また、前記アドレスＣＮＴにライトされた内容が“０”でなければ（９）、割込要求フラグＲＤＲＦを“０”にクリアする（１０）。そして、分岐命令を実行した（１１）後、割込ルーチンから復帰する（１３）。

【０１１２】

また、以下のプログラム例により、マルチプロセッサ受信のアドレス判定を行うことができる。なお、この場合にはアドレス判定の結果、受信したアドレスが自身のアドレスに一致していれば、受信完了割込の要求先をＣＰＵ１に変更するものとする。

【０１１３】

```
      PUSH   R0                    (1)
      MOV.B  @RDR,R0L              (2)
      CMP.B  #ID,R0L               (3)
      BEQ    L1                    (4)
      BCLR   #RDRF,@SSR            (5)
      BRA    L2                    (6)
L1:   BNOT   #RXI,@IRQSEL          (7)
L2:   RTE                          (8)
```

始めに、ＣＰＵ２に受信完了割込ＲＸＩが要求され、割込ルーチンに分岐する。まず、汎用レジスタＲ０をスタックに退避する（１）。そして、ＳＣＩの受信データレジスタＲＤＲの内容を汎用レジスタＲ０の下位８ビットＲ０Ｌにリードする（２）。

【０１１４】

この内容を、マルチプロセッサ受信の自分のＩＤと比較する（３）。この結果、一致していれば、Ｌ１に分岐し（４）、割込選択レジスタＩＲＱＳＥＬの受信完了割込ＲＸＩに対応するビットを反転する（７）。このとき、割込要求フラグＲＤＲＦが“１”にセットされたままなので、ＣＰＵ１に受信完了割込ＲＸＩが要求される。そして、割込ルーチンから復帰する（８）。

【０１１５】

また、自分のＩＤと比較した結果が一致していなければ（４）、割込要求フラグを“０”にクリアする（５）。そして、分岐命令を実行した（６）後、割込ルーチンから復帰する（８）。

【０１１６】

なお、この場合に１本の割込で複数の転送を行うことができる。たとえば、ステッピングモータを駆動して、かつこの加速／減速を行うことができる。また、タイマのコンペアマッチ割込ＯＣＭＩ毎に、ポートから出力するデータを更新してモータを駆動する。さらに、コンペアレジスタの内容を更新して、データ出力の周期を変更して加速／減速を行うことができる。

【０１１７】

この状態の表示に、内蔵ＲＡＭ２上に配置した作業用アドレスＵＰＤＷの内容を用いる。このビット６が“０”にクリアされているときに定速動作であり、“１”にセットされているときは加速または減速、すなわちビット７が“０”にクリアされているときは減速、“１”にセットされているときには加速動作である。なお、初期値はビット６、７ともに“１”にセットしておく。

【０１１８】

また、パルス出力回路のネクストデータレジスタＮＤＲに次の出力パルスを格納するものとし、タイマの定数レジスタＯＣＲにパルス出力周期を設定するものとする。タイマカウンタが定数レジスタの値に一致すると、図示はされないコンペアマッチ信号が発生し、ＮＤＲの内容がＩＯＰ８と兼用の端子から出力される。同時にタイマカウンタの値は０にクリアされる。このＮＤＲは４ビットとする。

【０１１９】

このようなパルス出力については、特願平４－１１７９６９号と同様とする。

【０１２０】

```
ＰＵＳＨ　　Ｒ０                    （１）
ＭＯＶ．Ｂ　＠ＮＤＲ，Ｒ０Ｌ          （２）
ＲＯＴＬ．Ｂ　Ｒ０Ｌ                  （３）
ＢＬＤ　　＃４、Ｒ０Ｌ                （４）
```

ＢＳＴ　　＃０，Ｒ０Ｌ　　（５）
ＭＯＶ．Ｂ　Ｒ０Ｌ，＠ＮＤＲ　　（６）
始めに、ＣＰＵ２にコンペアマッチ割込ＯＣＭＩが要求され、割込ルーチンに分岐する。まず、汎用レジスタＲ０をスタックに退避する（１）。そして、ネクストデータレジスタＮＤＲの内容を汎用レジスタＲ０の下位８ビットＲ０Ｌにリードする（２）。

【０１２１】
この内容を回転する（３）。ここでは、４ビット単位の回転とするために、回転後のビット４をキャリフラグに格納し（４）、このキャリフラグをビット０に格納する（５）。この４ビット単位の回転が終了した結果をＮＤＲに書き込む（６）。この内容が次のコンペアマッチで端子から出力される。

【０１２２】

```
        BTST   #6, @UPDW          (7)
        BEQ    L3                 (8)
        MOV. W @OCR, R0           (9)
        BTST   #7, @UPDW          (10)
        BNE    L1                 (11)
        SUB. W #CNST, R0          (12)
        BRA    L2                 (13)
    L1:ADD. W  #CNST, R0          (14)
    L2:MOV. W  R0, @OCR           (15)
```

続いて、アドレスＵＰＤＷのビット６の内容を検査し（７）、この内容が“０”（定速動作）であれば、Ｌ３に分岐し（８）、定数レジスタＯＣＲの内容は保持される。また、“０”でなければ、定数データレジスタＯＣＲの内容を汎用レジスタＲ０にリードする（９）。

【０１２３】
さらに、アドレスＵＰＤＷのビット７の内容を検査し（１０）、この内容が“０”（減速動作）であれば、Ｌ１に分岐し（１１）、“０”でなければ（加速動作）、定数値ＣＮＳＴを汎用レジスタＲ０から減算し（１２）、Ｌ２に分岐する（１３）。このＬ１では定数値ＣＮＳＴを汎用レジスタＲ０に加算し（１４）、Ｌ３では汎用レジスタＲ０の内容を定数レジスタＯＣＲに書き込む（１５）。

【０１２４】

```
L3:MOV. B  @CNT, R0H          (16)
   DEC. B  R0H                (17)
   BNE     L5                 (18)
   BTST    #6, @UPDW          (19)
   BEQ     L6                 (20)
   BTST    #7, @UPDW          (21)
   BEQ     L4                 (22)
   BCLR    #6, @UPDW          (23)
   MOV     #CNT1, R0H         (24)
   BRA     L5                 (25)
L4:BSET    #6, @UPDW          (26)
   BCLR    #7, @UPDW          (27)
   MOV     #CNT2, R0H         (28)
L5:MOV. B  R0H, @CNT          (29)
   BRA     L7                 (30)
L6:BNOT    #OCMI, @IRQSEL     (31)
L7:RTE                        (32)
```

続いて、アドレスＣＮＴの内容を汎用レジスタＲ０Ｈにリードする（１６）。この内容Ｒ０Ｈの内容をデクリメント（－１）する（１７）。この内容が“０”でなければ、Ｌ５に分岐し（１８）、“０”であれば、アドレスＵＰＤＷのビット６の内容を検査する（１９）。

【０１２５】

さらに、この内容が“０”（定速動作）であれば、Ｌ６に分岐し（２０）、“０”でなければ、ビット７の内容を検査し（２１）、さらにこの内容が“０”（減速動作）であれば、Ｌ６に分岐し（２２）、“０”でなければ（加速動作）、アドレスＵＰＤＷのビット６を“０”にクリアして、定速動作に遷移し（２３）、低速動作の回転数ＣＮＴ１をＲ０Ｈに設定する（２４）。そして、Ｌ５に分岐する（２５）。

【０１２６】

Ｌ４では、減速動作に遷移するため、アドレスＵＰＤＷのビット６を“１”にセット（２６）し、ビット７を“０”にクリアする（２７）。また、減速動作の回転数ＣＮＴ２をＲ０Ｈに設定する（２８）。

【０１２７】

さらに、Ｌ５では、汎用レジスタＲ０Ｈの内容をアドレスＣＮＴにライトする（２９）。そして、分岐命令を実行（３０）した後、割込ルーチンから復帰する（３２）。

【０１２８】

そして、減速動作の終了後、Ｌ６では割込選択レジスタＩＲＱＳＥＬのにコンペアマッチ割込ＯＣＭＩに対応するビットを反転する（３１）。このとき、割込要求フラグＯＣＭＦが“１”にセットされたままなので、ＣＰＵ１に受信完了割込ＲＸＩが要求される。そして、割込ルーチンから復帰する（３２）。

【０１２９】

また、起動は、前記割込要求レジスタを用いて、ＣＰＵ１からＣＰＵ２に割込を要求して、たとえばＲＡＭＰ上の所定のアドレスにコマンドを配置して指示をすればよい。

【０１３０】
以上により、たとえば特願平４－１３７９５４号または特願平４－１１７９６９号に記載の方法に比べて、ＲＡＭにデータを展開しておく必要がなく、ＲＡＭの使用量を節約できる。
【０１３１】
また、ＣＰＵ２を、割込に対応したデータ処理に主として使用すれば、ＣＰＵ２は常に処理すべきプログラムがあるとは限らない。そこで、１つの割込によるデータ処理が終了した時点で、処理すべきプログラムがなければ、ＣＰＵ２は低消費電力状態、いわゆるスリープ状態にするのがよい。
【０１３２】
このスリープ状態では、専用のＳＬＥＥＰ命令を実行することにより遷移する。そして、ＣＰＵ２に供給されるクロックが停止される。また、割込が要求されると、スリープ状態は解除される。
【０１３３】
次に、図７により低消費電力状態制御回路を詳細に説明する。
【０１３４】
なお、この低消費電力状態制御回路は、たとえばクロック発振器内に構成され、この低消費電力状態は３個のフリップフロップＳＬＰＦ１、ＳＬＰＦ２、ＳＳＢＹＦで制御される。
【０１３５】
フリップフロップＳＬＰＦ１は、ＣＰＵ１がＳＬＥＥＰ命令を実行すると、セット状態となる。このとき、ＣＰＵ１はスリープ状態となる。また、ＣＰＵ１に対する割込要求が発生するとクリア状態となる。
【０１３６】
フリップフロップＳＬＰＦ２は、ＣＰＵ２がＳＬＥＥＰ命令を実行すると、セット状態となる。このとき、ＣＰＵ２はスリープ状態となる。また、ＣＰＵ２に対する割込要求が発生するとクリア状態となる。
【０１３７】
フリップフロップＳＳＢＹＦは、図示はされない制御ビットＳＳＢＹを“１”にセットした状態で、一方のＣＰＵがスリープ状態で、他方のＣＰＵがＳＬＥＥＰ命令を実行するとセット状態となる。このソフトウェアスタンバイ状態では、クロック発振器を始め、全ての機能の動作が停止する。内部はリセット状態となる。
【０１３８】
ただし、規定の電圧が与えられている限り、ＲＡＭ１、ＲＡＭ２、ＲＡＭＰの内容は保持される。また、入出力ポートの状態も保持される。そして、ＮＭＩ割込要求が発生するとクリア状態となる。なお、内部はリセット状態となるためにＮＭＩ以外の割込要求は発生しない。
【０１３９】
このソフトウェアスタンバイ状態に遷移するためには、一方のＣＰＵがＳＳＢＹビットを“１”にセットし、ＳＬＥＥＰ命令を実行すればよい。他方のＣＰＵの状態に応じて、直ちにソフトウェアスタンバイ状態に遷移するか、一旦スリープ状態に遷移した後、他方のＣＰＵがＳＬＥＥＰ命令を実行した後にソフトウェアスタンバイ状態に遷移するかが自動的に選択される。このため、プログラムを作成する場合に、相互の動作状態を確認する必要がなく、プログラムの作成効率を向上することができる。
【０１４０】
また、割込処理を開始したとき、直前が動作中であったか、スリープ状態であったかを示すビットを設けるとよい。この動作中に割込を受け付けた場合には、ＣＰＵの内部のレジスタの内容を退避しなければならないが、スリープ中に割込を受け付ければ、ＣＰＵの内部のレジスタには作業中のデータはないので退避する必要がない。この退避動作および処理終了時の復帰動作を行わなければ、高速化を行うことができる。

【０１４１】
さらに、入出力ポートは、いずれのＣＰＵからライト可能にするか選択可能にする。
【０１４２】
次に、図８により入出力ポートのレジスタを詳細に説明する。
【０１４３】
この入出力ポートのレジスタは、データディレクションレジスタ（ＤＤＲ）、データレジスタ（ＤＲ）、ポート制御レジスタ（ＰＳＥＬ）からなる。
【０１４４】
ＤＤＲは、ライト専用レジスタであり、“０”にクリアされているときに入力、“１”にセットされているときには出力となる。
【０１４５】
ＤＲは、出力データを格納する。このＤＤＲを“１”にセットすると、ＤＲの内容が出力される。リード時にはＤＲの内容または端子の状態が読み出される。すなわち、ＤＤＲが“０”にクリアされているときは端子の状態、“１”にセットされているときにはＤＲの内容が読み出される。
【０１４６】
ＰＳＥＬが、各入出力端子の制御をいずれのＣＰＵで行うかを選択する。このポート制御レジスタ（ＰＳＥＬ）の制御ビットは、データレジスタ（ＤＲ）とデータディレクションレジスタ（ＤＤＲ）のビットを共通に制御する。ＰＳＥＬが“０”にクリアされているときはＣＰＵ１によって、“１”にセットされているときにはＣＰＵ２によってライトが行われる。
【０１４７】
なお、ポートにプルアップＭＯＳを内蔵したり、２重出力バッファとしたりする場合、これらの制御を行うレジスタも、ＰＳＥＬで共通に制御すればよい。このようなレジスタの例は、前記平成５年３月、（株）日立製作所発行『Ｈ８／３００３　ハードウェアマニュアル』Ｐ２６３～Ｐ３０６、Ｐ４１９～Ｐ４４５に記載されている。
【０１４８】
次に、図９により入出力ポートを詳細に説明する。
【０１４９】
本実施例の入出力ポートにおいては、データディレクションレジスタ（ＤＤＲ）、データレジスタ（ＤＲ）、ポート制御レジスタ（ＰＳＥＬ）はフリップフロップで構成される。いずれも、内部データバスの所定のビットをデータ入力（Ｄ）とし、リセット信号（ＲＥＳＥＴ）をクリア入力（ＣＬＲ）としている。
【０１５０】
ＤＤＲの入力クロック（Ｃ）は、論理積回路ＡＮＤ１の出力である。この出力（Ｑ）は、出力バッファＯＢＵＦの制御信号およびセレクタＳＥＬ１の選択信号とされる。
【０１５１】
ＤＲの入力クロック（Ｃ）は、論理積回路ＡＮＤ２の出力である。この出力（Ｑ）は出力バッファＯＢＵＦのデータ入力およびセレクタ１のデータ入力とされる。
【０１５２】
ＰＳＥＬの入力クロック（Ｃ）は、ＰＳＥＬリード信号ＷＲＰＳＥＬである。この出力（Ｑ）はセレクタＳＥＬ２の選択信号とされる。
【０１５３】
出力バッファＯＢＵＦは、制御信号としてＤＤＲ出力、データ入力はＤＲ出力とされ、出力信号は端子Ｐに接続される。
【０１５４】
入力バッファＩＢＵＦは、入力信号が端子Ｐに接続され、出力信号がセレクタＳＥＬ１のデータ入力とされる。
【０１５５】
セレクタＳＥＬ１は、ＤＤＲ出力を選択信号、ＤＲ出力および入力バッファＩＢＵＦ出力

をデータ入力とする。このＤＤＲ出力が“０”のときは入力バッファＩＢＵＦ出力が選択され、ＤＤＲ出力が“１”のときにはＤＲ出力が選択される。このセレクタＳＥＬ１の出力はＤＲリードバッファの入力とされる。

【０１５６】

セレクタＳＥＬ２は、ＰＳＥＬ出力を選択信号、ＣＰＵ１／２バス権信号をデータ入力とする。この出力はライト許可信号となる。ＰＳＥＬが“０”のときはＣＰＵ１バス権で、ＣＰＵ１がライト許可状態となる。ＰＳＥＬが“１”のときにはＣＰＵ２バス権で、ＣＰＵ２がライト許可状態となる。

【０１５７】

ＤＲリードバッファＤＲＢＵＦは、セレクタＳＥＬ１の出力を入力し、ＤＲリード信号ＲＤＤＲをクロックとし、出力信号は内部データバスの所定のビットに接続される。このＲＤＤＲが活性状態のとき、セレクタＳＥＬ１の出力を内部データバスに出力する。

【０１５８】

ＰＳＥＬリードバッファＰＳＢＵＦは、ＰＳＥＬフリップフロップの出力を入力し、ＰＳＥＬリード信号ＲＤＰＳＥＬをクロックとし、出力信号は内部データバスの所定のビットに接続される。このＲＤＰＳＥＬが活性状態のとき、ＰＳＥＬフリップフロップの出力を内部データバスに出力する。

【０１５９】

論理積回路ＡＮＤ１は、ライト許可信号とＤＤＲライト信号ＷＲＤＤＲを入力とし、出力信号はＤＤＲの入力クロックとなる。

【０１６０】

論理積回路ＡＮＤ２は、ライト許可信号とＤＲライト信号ＷＲＤＲを入力とし、出力信号はＤＲの入力クロックとなる。

【０１６１】

ＲＯＭ１およびＲＯＭ２をＰＲＯＭとすることができ、このＰＲＯＭを外部からプログラムするとき、２つのＲＯＭを外部から連続したアドレスとするようにするとよい。

【０１６２】

次に、図１０によりＰＲＯＭのアドレスマップを説明する。

【０１６３】

たとえば、ＰＲＯＭ１は６４ｋバイト、ＰＲＯＭ２は３２ｋバイトとする。ＭＣＵモードでは、ＰＲＯＭ１はＨ’０００００００００～Ｈ’００００ＦＦＦＦに、ＰＲＯＭ２はＨ’００００～Ｈ’７ＦＦＦに配置される。

【０１６４】

また、ＰＲＯＭモードでは、ＰＲＯＭ１はＨ’０００００～Ｈ’０ＦＦＦＦに、ＰＲＯＭ２はＨ’１００００～Ｈ’１７ＦＦＦに配置される。

【０１６５】

次に、図１１によりＰＲＯＭモードを詳細に説明する。

【０１６６】

このＰＲＯＭモードは、モード端子ＭＤ１、ＭＤ０、およびＳＴＢＹ端子をいずれもロウレベルとすることによって設定される。このとき、図示はされないモード設定回路によって、内部信号ＥＰＭが活性状態（ハイレベル）とされる。このＰＲＯＭモードのときには、ＣＰＵ１／２はＢＵＦＣ１／２によって、アドレスバス、データバスから切り離される。

【０１６７】

図１１において、ＣＳ制御回路、バッファＢＵＦＣ１／２などはバスコントローラに含まれてなる。

【０１６８】

ＰＲＯＭは、図１におけるＲＯＭ１およびＲＯＭ２とし、メモリアレイ、アドレスデコーダ、入出力回路、制御回路からなる。このアドレスデコーダにはアドレスが入力され、入出力回路はデータの入出力を行い、また制御回路は制御信号を入力し、入出力回路を制御

してデータの書込み／読み出しなどを行う。また、メモリアレイは不揮発記憶素子が配列されている。

【０１６９】

制御回路には、ＥＰＭ、ＣＳ－１／２＃、ＯＥ＃、ＰＧＭ＃、およびＭＳ－１／２＃、リード信号が入力される。このＥＰＭが活性状態のとき、ＰＲＯＭとしての書込み／読み出し動作が、ＣＳ－１／２＃、ＯＥ＃、ＰＧＭ＃に基づいて行われる。また、ＥＰＭが非活性状態のときには、ＣＰＵのアドレス空間上でのＲＯＭとしての読み出し動作が、ＭＳ－１／２＃、リード信号に基づいて行われる。

【０１７０】

ＰＲＯＭモードのとき、ＶＰＰ、ＣＳ＃、ＯＥ＃、ＰＧＭ＃で制御される。このような制御は、たとえば１９９３年３月、（株）日立製作所発行『ＨＩＴＡＣＨＩ　ＩＣ　Ｍｅｍｏｒｙ　Ｄａｔａ　Ｂｏｏｋ　Ｎｏ．２（１０ｔｈ　Ｅｄｉｔｉｏｎ）』Ｐ５１０～Ｐ５２４に記載されるＰＲＯＭと同様である。

【０１７１】

ＶＰＰ端子はＲＥＳ端子と、ＣＳ＃、ＯＥ＃、ＰＧＭ＃はＰ６１、Ｐ６２、Ｐ６３端子と、アドレスＡ０－１５はＰ００－０７、Ｐ１０－１７端子と、Ａ１６はＰ６０端子と、データＤ７－０はＰ４０－Ｐ４７端子と兼用にされている。なお、ＶＰＰは図示はされないが、ＰＲＯＭ１／２に与えられる。

【０１７２】

内部信号としては、ＯＥ＃、ＰＧＭ＃がバッファ回路を介して入力され、ＰＲＯＭ１、ＰＲＯＭ２に供給される。ＣＳ－１＃は、アドレス上位信号Ａ１６の反転信号との論理積信号、ＣＳ－２＃は、アドレス上位信号Ａ１５の反転信号、およびＡ１６との論理積信号で生成される。このＣＳ－１＃、ＣＳ－２＃が、それぞれＰＲＯＭ１、ＰＲＯＭ２に与えられる。

【０１７３】

ＰＲＯＭモードのとき、ＩＤＢ１とＩＤＢ２がバッファを介して接続される。ＰＲＯＭ２のデータはＩＤＢ２、ＩＤＢ１およびＩＯＰ３を介して入出力される。すなわち、ＣＳ－２＃が活性状態で読み出し動作のとき、ＰＲＯＭ２から読み出したＩＤＢ２の内容がＩＤＢ１に出力される。さらに、読み出し動作のとき、ＩＤＢ１の内容がＩＯＰ３を介して外部に出力される。

【０１７４】

また、書込み動作のとき、外部の内容がＩＯＰ３を介してＩＤＢ１に入力される。ＣＳ－２＃が活性状態で書込み動作のとき、ＩＤＢ１の内容がＩＤＢ２に出力され、ＰＲＯＭ２に与えられる。さらに、ＩＤＢ１の内容がＩＯＰ３を介して外部に出力される。

【０１７５】

ＰＲＯＭモードのとき、ＩＡＢ１とＩＡＢ２がバッファを介して接続される。すなわち、ＩＯＰ１、２を介して入力されたＩＡＢ１のアドレスが、ＩＡＢ２に入力され、ＰＲＯＭ２に与えられる。なお、ＰＲＯＭ１／２の最大容量が６４ｋバイトであるので、それぞれのＰＲＯＭに与えられるアドレスは１６ビットである。

【０１７６】

たとえば、読み出し動作はＣＳ－１／２＃、ＯＥ＃が活性状態、ＰＧＭ＃が非活性状態のときに行われ、書込み動作はＣＳ－１／２＃、ＰＧＭ＃が活性状態、ＯＥ＃が非活性状態のときに行われる。

【０１７７】

その他、ベリファイやページ書込みなどが可能とされるが、本発明には直接の関係はないので、詳細な説明は省略する。

【０１７８】

１つのＰＲＯＭモードで、ＰＲＯＭ１、ＰＲＯＭ２を、外面的には１つのＰＲＯＭとして一括して書込むことができる。これによって、書込みを効率的に行うことができる。

【０１７９】

次に、図１２によりテストモードを詳細に説明する。

【０１８０】

このテストモードは、モード端子ＭＤ１、ＭＤ０をいずれもロウレベル、およびＳＴＢＹ端子をハイレベルとすることによって設定される。このとき、図示はされないモード設定回路によって、内部信号ＴＭが活性状態（ハイレベル）とされる。

【０１８１】

図１２において、制御回路１／２、バス権調停回路、選択回路、バッファＢＳＢＵＦ、バッファＢＵＦＣ１／２はバスコントローラに含まれてなる。

【０１８２】

バス権調停回路は、ＣＰＵ１／２のいずれが、ＰＡＢ／ＰＤＢを使用するかを選択して制御信号ＰＡＫを出力する。このＰＡＫがハイレベルのときはＣＰＵ１、ＰＡＫがロウレベルのときにはＣＰＵ２がＰＡＢ／ＰＤＢを使用する。テストモードの場合は、Ｐ６２端子から与えられる選択信号によってＰＡＫ信号は制御される。

【０１８３】

テストモードのとき、ＣＰＵ１／２はＢＵＦＣ１／２によって、アドレスバス、データバス、リード信号／ライト信号から切り離される。

【０１８４】

アドレスはＩＯＰ３～ＩＯＰ０から、バッファＢＵＦＰを介してＩＡＢ１に入力され、さらにＢＳＢＵＦを介してＩＡＢ２およびＰＡＢに入力される。なお、ＩＡＢ１は４Ｇバイトのアドレス空間に対応して３２本のアドレスを有するが、ＩＡＢ２は６４ｋバイトのアドレス空間に対応して１６本、ＰＡＢはＲＡＭＰ、Ｉ／Ｏのアドレスに対応して１０本のアドレスを有する。従って、外部からアドレスを与えるためにはＩＡＢ１を利用するのがよい。

【０１８５】

データバスは、ＩＯＰ４によって、バッファＢＵＦＰを介してＩＤＢ１と入出力される。リード／ライト動作およびアドレスに対応して、ＩＤＢ２またはＰＤＢの内容がＢＳＢＵＦを介してＩＤＢ１と入出力される。

【０１８６】

リード信号、ライト信号は、Ｐ６０端子、Ｐ６１端子からＢＵＦＰを介して入力される。ＲＤ１／２、ＷＲ１／２として、ＲＯＭ１、ＲＡＭ１／ＲＯＭ２、ＲＡＭ２に供給される。また、バス権信号ＰＡＫによって選択されてＲＤＰ、ＷＲＰとなって、ＲＡＭＰ、Ｉ／Ｏに供給される。テストモードにおいては、ＰＡＫ信号によらず、ＲＤ１／２／Ｐは常に同様の動作となり、ＷＲ１／２／Ｐも常に同様の動作となる。

【０１８７】

制御回路１／２は、ＩＡＢ１／２のアドレスに基づいて、選択信号ＭＳＲＯＭ１／２、ＭＳＲＡＭ１／２、ＭＳＲＡＭＰ、ＭＳＩＯを出力する。ＭＳＲＡＭＰ、ＭＳＩＯは共通信号であるので、ＰＡＫ信号に基づいて、一方が出力、他方はハイインピーダンスとなる。たとえば、ＰＡＫ信号がハイレベルのときは制御回路１が出力し、制御回路２はハイインピーダンスとなる。

【０１８８】

また、テストモード時には、ＭＳＲＯＭ１／２、ＭＳＲＡＭ１／２はＰＡＫ信号に基づいて一方が禁止される。これは外部から与えるアドレスに基づいて、１つの機能ブロックが選択されるようにするためである。たとえば、ＰＡＫ信号がハイレベルのときはＭＳＲＯＭ１、ＭＳＲＡＭ１はＩＡＢ１に基づいた出力、ＭＳＲＯＭ２、ＭＳＲＡＭ２は非活性状態とされる。

【０１８９】

ＲＯＭ１／ＲＡＭ１をテストする場合には、Ｐ６２端子をハイレベルにして、ＰＡＫ信号をハイレベルにした状態で、アドレス、リード信号／ライト信号を与えることによってリード／ライトを行うことができる。

【０１９０】

また、ＲＯＭ２／ＲＡＭ２をテストする場合には、Ｐ６２端子をロウレベルにして、ＰＡＫ信号をロウレベルにした状態で、アドレス、リード信号／ライト信号を与えることによってリード／ライトを行うことができる。このアドレスは下位１６ビットのみが有効になる。

【０１９１】

さらに、ＲＡＭＰ／Ｉ／Ｏをテストする場合には、Ｐ６２端子をハイレベルにして、ＰＡＫ信号をハイレベルにした状態で、アドレス、リード信号／ライト信号を与えることによってリード／ライトを行うことができる。

【０１９２】

上記によって、ＣＰＵによってアドレスが与えられ、かつデータの入出力が行われるべき内部バスに、内蔵の機能ブロックを外部からリード／ライトするためのアドレスデータなどを外部から供給可能にして、この機能ブロックに必要なデータのリード／ライトを外部から直接行うことができるようにし、このシングルチップマイクロコンピュータに内蔵される機能ブロックのテストの容易化、ならびにテスティング効率の向上を実現できる。

【０１９３】

また、内蔵ＲＡＭをテストする場合、データのテストは独立して行い、アドレスデコーダのテストは並行して行うようにするとよい。このアドレスデコーダのテストを行うには、ＲＡＭ１／２／Ｐを同時に選択可能にすることによって同時にテストを行うことができる。

【０１９４】

この場合に、図１２に対してＲＡＭテストモードを設ければよい。テストモードにした状態で、Ｐ６３端子をロウレベルとすると前記同様のテストモード、Ｐ６３端子をハイレベルにするとＲＡＭテストモードとなる。

【０１９５】

このＲＡＭテストモードでは、ＲＡＭ１／２／Ｐの選択信号ＭＳＲＡＭ１、ＭＳＲＡＭ２、ＭＳＲＡＭＰを同時に活性状態にすることができる。ただし、リードするデータはそれぞれ、ビット０～１、ビット２～３、ビット４～７とする。

【０１９６】

ＣＰＵ１とＣＰＵ２は、実質的に同一のＣＰＵであるか、またはＣＰＵ１がＣＰＵ２を包含するようにし、ソフトウェアの開発環境を共有できるようにするのがよい。このようなＣＰＵには、たとえば特願平４ー２２６４４７号がある。

【０１９７】

以上のようなシングルチップマイクロコンピュータを評価するエミュレータは、使用者が外部からブレーク条件を設定するとき、いずれのＣＰＵにブレークを設定するかを指定可能にするとよい。また、２つのＣＰＵの条件が両方成立したときにブレークするように指定可能にするとよい。

【０１９８】

次に、図１３によりエミュレーション用プロセッサの構成を説明する。

【０１９９】

エミュレーション用プロセッサは、ＣＰＵ１、ＣＰＵ２を含むマイクロコンピュータ部分、および図示はされないバッファ回路などを含むエミュレーション用プロセッサ専用ブロックから構成され、公知の半導体製造技術によって１つの半導体基板上に形成されている。

【０２００】

このエミュレーション用プロセッサは、応用システムと信号の送受信を行うユーザインタフェース、およびマイクロコンピュータ開発装置と信号の送受信を行うエミュレーションインタフェースを有している。

【０２０１】

エミュレーション用バスとしては、ＩＡＢ１およびＩＤＢ１、ＩＡＢ２およびＩＤＢ２をエミュレーションインタフェースを介して入出力する。また、ＰＡＢおよびＰＤＢのいず

れのＣＰＵが使用しているかを示すバス権表示信号を出力している。これは図１３のＰＡＫ信号に相当するものとされる。また、バス権調停によって、ＣＰＵが待機状態とされていることを示す信号を出力している。

【０２０２】

ＰＡＢ、ＰＤＢのアクセス状態は明示的には表示されないが、バス権表示信号とＩＡＢ１およびＩＤＢ１、ＩＡＢ２およびＩＤＢ２によるエミュレーション用バスの内容によって判断することができる。このＰＡＢ、ＰＤＢによるエミュレーション用バスを出力しないことによって、エミュレーション用プロセッサの端子数を削減でき、これによってパッケージサイズを小さくして、ひいてはエミュレータのサイズを縮小できる。

【０２０３】

ブレーク要求端子は、ＢＲＫ１＃、ＢＲＫ２＃の２本がエミュレーションインタフェースに含まれてなる。これに基づいてＢＲＫ１、ＢＲＫ２信号によって、それぞれＣＰＵ１、ＣＰＵ２にブレーク割込を要求する。これらは対応するＣＰＵ選択ビットを持たず、優先順位判定回路を経て、割込要求信号と並列したＣＰＵへのブレーク要求信号を活性状態にする。また、ベクタ番号信号は一般の割込要求と兼用される。

【０２０４】

ＣＰＵ１、ＣＰＵ２が、ブレーク割込を受け付けると、それぞれブレークアクノリッジ信号ＢＲＫＡＫ１、ＢＲＫＡＫ２が活性状態になる。たとえば、ＣＰＵ２が、前記のようなＳＣＩの受信処理、パルス出力処理を行っている場合、ＣＰＵ１をブレーク割込によって停止しても、ＣＰＵ２はユーザの処理を実行するためにＳＣＩの受信を中断してしまい、いわゆるオーバランエラーが発生したり、パルス出力を中断してしまい、たとえばユーザシステムのモータ駆動ができなくなってしまうようなことを回避できる。

【０２０５】

次に、図１４により、エミュレーション用プロセッサを搭載したエミュレータの構成を説明する。

【０２０６】

コネクタ部がシングルチップマイクロコンピュータの代わりに応用システム（ユーザシステム）に装着される。エミュレーション用プロセッサは、このコネクタ部とインタフェースケーブルを介し、ターゲットシステムインタフェースを用いて応用システムと信号の入出力を行う。

【０２０７】

また、エミュレーション用プロセッサはエミュレーションインタフェースを用いてエミュレーションバス１／２に接続される。このエミュレーションバス１がＩＡＢ１／ＩＤＢ１に対応し、エミュレーションバス２がＩＡＢ２／ＩＤＢ２に対応する。これらのエミュレーションバスには、図示はされない状態信号、制御信号などを含む。

【０２０８】

このエミュレーションバスを用いて、エミュレーション用プロセッサから、応用システムとエミュレーション用プロセッサの内部状態に応じた情報などが出力され、またエミュレーション用プロセッサに対し、エミュレーションのための各種制御信号が入力される。そして、エミュレーション用プロセッサの、図示はされないエミュレートモード端子が電源レベルに固定され、エミュレーション用プロセッサ内部ではエミュレートモードが設定される。

【０２０９】

また、エミュレーションバスには、特に制限はされないものの、応用システムまたはターゲットマイクロコンピュータ内蔵のメモリを代行するためのＲＡＭでなるようなエミュレーションメモリと、エミュレーション用プロセッサの制御状態やエミュレーションバスの状態を監視して、その状態が予め設定された状態に達した時に、エミュレータ専用割込を入力して、ＣＰＵによるユーザプログラムの実行を停止させ、エミュレーション用プログラム実行状態に遷移させる（ブレーク）ためのブレーク制御回路が接続される。

【０２１０】

さらに、このエミュレーションバスには、ＣＰＵのリード動作またはライト動作を示す信号、命令リード動作を示す信号などに基づき、エミュレーションバスに与えられるアドレスやデータさらには制御情報を逐次蓄えるリアルタイムトレース回路などが接続される。本実施例においては、エミュレーションバス１／２がエミュレーションメモリ、ブレーク制御回路、リアルタイムトレース回路などにそれぞれ接続される。

【０２１１】

従って、エミュレーションメモリは、ＩＡＢ１／ＩＤＢ１およびＩＡＢ２／ＩＤＢ２のデータを独立して並立的に入出力する。また、ブレーク制御回路は、ＩＡＢ１／ＩＤＢ１およびＩＡＢ２／ＩＤＢ２のアドレス／データ、その他の情報を独立に判定する。このＩＡＢ１／ＩＤＢ１側の条件が成立した後、ＩＡＢ２／ＩＤＢ２側の条件が成立したときにブレークを要求することが可能にされる。前記の通り、ブレーク割込はＣＰＵ１およびＣＰＵ２の両方に同時に要求することもできるし、一方のＣＰＵのみに要求することもできる。

【０２１２】

また、リアルタイムトレース回路は、ＩＡＢ１／ＩＤＢ１およびＩＡＢ２／ＩＤＢ２のアドレス／データ、その他の情報を独立に蓄積し、一方のＣＰＵがブレークしたときには、他方のＣＰＵのアドレス／データ、その他の情報のみを蓄積する。また、一方のＣＰＵのみのアドレス／データ、その他の情報を選択的に蓄積することもできる。

【０２１３】

このエミュレーションメモリ、ブレーク制御回路、リアルタイムトレース回路はコントロールバスに接続され、コントロールバスを介してコントロールプロセッサの制御を受けるようになっている。このコントロールバスは、エミュレーション用プロセッサ制御回路に接続されるとともに、インタフェース回路を介して、特に制限はされないものの、パーソナルコンピュータなどのシステム開発装置に接続される。

【０２１４】

たとえば、システム開発装置から入力されたプログラムをエミュレーションメモリに転送し、内蔵ＲＯＭ上に配置されるべきこのプログラムをＣＰＵがリードすると、エミュレーションメモリ上のプログラムがリードされる。また、ブレーク条件や、リアルタイムトレース条件などもシステム開発装置から与えることができる。

【０２１５】

このブレーク条件としては、少なくとも、ＣＰＵ１とＣＰＵ２の条件が独立して設定できるようにされる。また、リアルタイムトレース条件としては、少なくとも、ＣＰＵ１とＣＰＵ２のアドレス／データ、その他の情報を同時に蓄積するか、選択的に一方のＣＰＵのアドレス／データ、その他の情報のみを蓄積するかを選択できるようにする。

【０２１６】

従って、本実施例のシングルチップマイクロコンピュータによれば、データ処理装置としてのＣＰＵ１，ＣＰＵ２、バス制御手段としてのバスコントローラ、読み出し可能な記憶手段としてのＲＯＭ１，ＲＯＭ２、データ入出力手段としてのＩＯＰ０～ＩＯＰ１１を主要な構成とすることにより、以下の作用効果を得ることができる。

【０２１７】

(1).ＣＰＵ１とＲＯＭ１を接続するＩＡＢ１，ＩＤＢ１と、ＣＰＵ２とＲＯＭ２を接続するＩＡＢ２，ＩＤＢ２とを独立に設け、ＲＯＭ１にＣＰＵ１のプログラムを、ＲＯＭ２にＣＰＵ２のプログラムを格納することにより、ＣＰＵ１とＣＰＵ２が並列動作することを可能にし、またＲＡＭＰとＩＯＰ０～ＩＯＰ１１などによる入出力回路を共通のＰＡＢ，ＰＤＢに接続し、ＣＰＵ１およびＣＰＵ２が利用可能とすることにより、ＣＰＵ１とＣＰＵ２の制御すべきデータを任意に配分して利用することができる。これによって、シングルチップマイクロコンピュータの処理速度を向上し、資源利用効率を向上できる。

【０２１８】

(2).１つのシングルチップマイクロコンピュータで２つの処理を同時に行うことにより、従来の複数の半導体集積回路装置を利用していたシステムを代替して、小型化を実現する

ことができる。
【０２１９】
(3).ＣＰＵ１およびＣＰＵ２が、相互に割込を要求することを可能にすることにより、容易にＣＰＵ１，２の相互の同期を実現することができる。
【０２２０】
(4).ＣＰＵ１およびＣＰＵ２の動作によって、ソフトウェアスタンバイのようなシングルチップマイクロコンピュータの全体を停止状態とする場合、ＳＳＢＹビットを“１”にセットした状態で、一方のＣＰＵがＳＬＥＥＰ命令を実行したときに、他方のＣＰＵの状態に応じて、直ちにソフトウェアスタンバイ状態に遷移するか、一旦スリープ状態に遷移した後、他方のＣＰＵがＳＬＥＥＰ命令を実行した後にソフトウェアスタンバイ状態に遷移するかが自動的に選択することによって、プログラムを作成する場合に、相互の動作状態を確認する必要がなく、プログラムの作成効率を向上することができる。
【０２２１】
(5).ＩＯＰ０～ＩＯＰ１１をビット単位で、いずれのＣＰＵ１，２が利用するかを指定するレジスタＤＲを設けることにより、１本の入出力ポートを２つのＣＰＵ１，２で利用することができ、資源の利用効率をより向上することができる。
【０２２２】
(6).割込を、いずれのＣＰＵ１，２に要求するかを指定するレジスタを割込許可回路に設けることにより、入出力回路を２つのＣＰＵ１，２で任意に配分して利用することができ、資源の利用効率をより向上することができる。
【０２２３】
(7).一方のＣＰＵを、主に割込によるデータ転送を行うことに使用することにより、割込時の種々の処理を容易に行うことができる。
【０２２４】
(8).ＲＯＭ１，２をＰＲＯＭで構成した場合、ＰＲＯＭモード時にはＲＯＭ１とＲＯＭ２を連続したアドレスに配置することにより、ＲＯＭ１とＲＯＭ２を１つのＰＲＯＭと互換性を持たせ、ＰＲＯＭ書込みを効率化することができる。
【０２２５】
(9).１つの物理的アドレスに対応して、ＣＰＵ１，２毎に異なる論理的アドレスを有する場合、または異なるＣＰＵ１，２の１つの論理的アドレスに対応して、異なる物理的アドレスを有する場合に、いずれの論理アドレスを使用するかを選択する制御信号を与えることによって、ＣＰＵ１，２によってアドレスが与えられ、かつデータの入出力が行われるべき複数の内部バスに、外部から内蔵の機能ブロックをリード／ライトするためのアドレスデータなどを前記内部バスより少ないバッファ回路を介して供給可能にして、この機能ブロックに必要なデータのリード／ライトを外部から直接行うことができるようにし、このシングルチップマイクロコンピュータに内蔵される機能ブロックのテストの容易化、ならびにテスティング効率の向上を実現できる。
【０２２６】
(10).ＣＰＵ１とＣＰＵ２の互換性を持たせることにより、開発効率を向上することができ、またサポートツールを共通化することを可能にし、開発装置の開発効率を向上することができる。これにより、使用者が必要とする開発装置の数を低減し、開発装置に必要な費用を低減することができる。
【０２２７】
(11).このシングルチップマイクロコンピュータのエミュレータにおいては、２つのＣＰＵ１，２を独立に停止する割込を有することにより、デバック効率を向上することができる。
【０２２８】
また、本実施例においては、図１５に示すように、図１のシングルチップマイクロコンピュータに対して、ダイレクトメモリアクセスコントローラ（ＤＭＡＣ）がＣＰＵ１と同様にＩＡＢ１，ＩＤＢ１に接続されて追加されることにより、ＣＰＵ１とＤＭＡＣを、バス

コントローラのバス権調停に基づいて互いに排他的に動作させることができる。
【０２２９】
すなわち、ＤＭＡＣは、起動要因として、ＣＰＵ１が所定の制御ビットを“１”にセットすることによって、データ転送を行うオートリクエストや、ＤＭＡ要求端子に所定の信号が与えられたときに、データ転送を行う外部リクエスト機能を持ち、また転送モードとして、オートリクエストのとき、ＣＰＵ１を停止してバスを専有してデータ転送を行うバーストモードや、１回の転送毎にバス権を解放してＣＰＵ１と交互に動作しつつ、データ転送を行うサイクルスチールモードを持つことにより可能となる。
【０２３０】
また、割込要求信号を起動要求信号として、データ転送を行うことができ、ＣＰＵ２が割込でデータ転送を行うより高速のデータ転送が可能である。
【０２３１】
さらに、バス調停は、リード／ライトの競合時に行うほか、時分割で動作するようにしてもよい。たとえば、８ステート毎にＣＰＵ１およびＣＰＵ２にバス権を与える。また、８ステート間にバスが終了しなかった場合には、終了するまで動作する。たとえば、ＣＰＵ１には４ステート、ＣＰＵ２には１２ステートずつバス権を与えてもよい。
【０２３２】
（実施例２）
図１６は本発明の他の実施例であるシングルチップマイクロコンピュータを示すブロック図、図１７は本実施例のシングルチップマイクロコンピュータにおけるバスの動作タイミング図である。
【０２３３】
本実施例のシングルチップマイクロコンピュータは、実施例１に対して、ＩＡＢ１とＩＡＢ２、ＩＤＢ１とＩＤＢ２が共通化されてＩＡＢ／ＩＤＢとされ、データバス幅を３２ビットとし、同様にＲＯＭ１とＲＯＭ２、ＲＡＭ１とＲＡＭ２が共通化されてリードオンリメモリ（ＲＯＭ：読み出し可能な記憶手段）／ランダムアクセスメモリ（ＲＡＭ）とされ、これらは３２ビット同時にリード／ライト可能にしている。
【０２３４】
また、ＣＰＵ１（第１のデータ処理装置）およびＣＰＵ２（第２のデータ処理装置）は時分割で、内部アドレスバス（ＩＡＢ）／内部データバス（ＩＤＢ）を利用して、ＲＯＭ／ＲＡＭをリード／ライトする。前記の通り、これらのＣＰＵ１／２の命令は１６ビット単位であり、最小命令を１ステートで実行する。
【０２３５】
すなわち、１つのＣＰＵ１／２に対して、命令については１６ビットを１ステートでリードして、命令リード量と命令実行量が同等となる。３２ビットで命令をリードすれば、命令リードの回数を半分にして、ＩＡＢ／ＩＤＢおよびＲＯＭ／ＲＡＭの使用頻度を１／２とすることができる。従って、ＣＰＵ１およびＣＰＵ２が交互にＩＡＢ／ＩＤＢを利用してＲＯＭから命令を読み出すことができ、並列に実行可能とすることができる。
【０２３６】
実際には、分岐命令などのように３２ビットで命令をリードしても、１６ビットが無駄になってしまったり、データアクセスのときには使用頻度を減らせなっかたりして完全な並列動作はできないが、前記の通り命令のリード回数が多く、本実施例によっても、実施例１に対して処理速度をそれほど低下させることがない。また、ＲＯＭ／ＲＡＭを共通化することによって、物理的規模を低下させたり、ＣＰＵ１／２の使用するメモリ容量を任意に設定したりすることができる。
【０２３７】
次に、図１７により内部バスの動作タイミングの一例を説明する。
【０２３８】
なお、ＣＰＵ１／２のプログラムは図６と同様であり、バスの使用権は、前回バスを使用していたＣＰＵ１／２が非優先とされ、そのほかの場合にはＣＰＵ２を優先するものとす

る。
【０２３９】
たとえば、ＩＡＢ１／２は、φ＃に同期して出力され、またＣＰＵ１／２のＲＯＭおよびＲＡＭに対するリードは、３２ビット一括して１ステートで行われる。ただし、これは４の倍数番地から始まる３２ビットデータに限定される。
【０２４０】
まず、ＩＡＢ１／２はφ＃に同期して、１ステート出力され、特に制限はされないものの、ＲＯＭおよびＲＡＭの中でφに同期してラッチされる。これに対応するリードデータはφ＃に同期して出力され、φ＃が活性状態の期間にＣＰＵ１／２に取り込まれる。
【０２４１】
また、前記同様に、ＲＡＭＰに対するリード／ライトは２ステート、Ｉ／Ｏに対するリード／ライトは３ステートで行われる。φ＃に同期したＩＡＢ１／２は、バスコントローラでφに同期化される。この場合に、先頭命令の存在するアドレスｍ、ｎは４の倍数番地とされる。
【０２４２】
Ｔ１では、ＣＰＵ１／２のＲＯＭからの命令フェッチが要求されるが、ＣＰＵ２の命令フェッチが優先され、ＩＡＢにアドレスｍが出力される。
【０２４３】
Ｔ２では、ＲＯＭからＩＤＢに読み出された命令コード（ＢＣＬＲ命令の第１、第２ワード）をＣＰＵ２が取り込むとともに、ＣＰＵ１の命令フェッチのアドレスｎがＩＡＢに出力される。
【０２４４】
Ｔ３では、ＲＯＭからＩＤＢに読み出された命令コード（ＣＭＰ、ＢＥＱ命令）をＣＰＵ１が取り込むとともに、ＣＰＵ２はアドレスＳＳＲをＩＡＢに出力する。自動的に、ＣＰＵ２のＰＡＢ／ＰＤＢの使用が許可される。この場合に、ＳＣＩのリードが３ステート必要であるため、ＣＰＵ２待機信号がハイレベルになる。
【０２４５】
Ｔ４では、ＩＡＢの内容がＰＡＢに出力されて、ＳＣＩのレジスタＳＳＲの内容がリードされる。このリードデータはＴ６で得られる。一方、ＣＰＵ１が命令フェッチを要求するが、ＣＰＵ２がバスを使用中であるので、保留とされ、ＣＰＵ１待機信号がハイレベルになる。
【０２４６】
Ｔ５では、ＣＰＵ１／２ともに待機状態とされる。
【０２４７】
Ｔ６では、ＳＣＩのレジスタＳＳＲから読み出されたデータをＰＤＢ、ＩＤＢを介してＣＰＵ２が取り込む。また、ＣＰＵ２は次の命令フェッチを要求するが、ＣＰＵ１の命令フェッチが優先され、ＣＰＵ１待機信号がロウレベルとなり、ＣＰＵ２待機信号ハイレベルのままである。この場合に、ＣＰＵ１の命令フェッチのアドレスｎ＋４をＩＡＢに出力する。
【０２４８】
Ｔ７では、ＲＯＭからＩＤＢに読み出された命令コード（ＢＴＳＴ命令の第１ワード）をＣＰＵ１が取り込む。この場合に、ＣＰＵ１は分岐先命令のフェッチを要求するが、ＣＰＵ２の命令フェッチが優先され、ＣＰＵ２待機信号がロウレベルとなり、ＣＰＵ１待機信号ハイレベルになる。
【０２４９】
Ｔ８では、ＲＯＭからＩＤＢに読み出された次の命令コードをＣＰＵ２が取り込む。この場合に、ＣＰＵ１待機信号がロウレベルとなり、ＣＰＵ１の命令フェッチのアドレスＮ１をＩＡＢに出力する。
【０２５０】
Ｔ９では、ＲＯＭからＩＤＢに読み出された命令コードをＣＰＵ１が取り込むが、これは

分岐条件が不成立で実行されない。先にフェッチ済みのＢＴＳＴ命令の第１ワードが有効になる。この場合に、ＣＰＵ１は次の命令フェッチを要求するが、ＣＰＵ２が優先され、ＣＰＵ１待機信号ハイレベルになる。

【０２５１】

一方、ＣＰＵ２はアドレスＳＳＲをＩＡＢに出力する。自動的に、ＣＰＵ２のＰＡＢ／ＰＤＢの使用が許可される。ここで、ＳＣＩのリードが３ステート必要であるため、ＣＰＵ２待機信号がハイレベルになる。

【０２５２】

Ｔ１０では、ＩＡＢの内容がＰＡＢに出力され、ＣＰＵ２はアドレスＳＳＲにライトすべきデータをＩＤＢ経由ＰＤＢに出力する。ここで、ＣＰＵ１は待機状態とされる。

【０２５３】

Ｔ１１では、ＣＰＵ２待機信号がロウレベルになる。

【０２５４】

Ｔ１２では、ＣＰＵ１待機信号がロウレベルとなり、ＣＰＵ１の命令フェッチのアドレスｎ＋６をＩＡＢに出力する。

【０２５５】

Ｔ１３では、ＲＯＭからＩＤＢに読み出された命令コード（ＢＴＳＴ命令の第２ワード）をＣＰＵ１が取り込む。この場合に、ＣＰＵ１はアドレスＰＯＲＴのリードを要求し、ＣＰＵ２はＲＯＭからの命令フェッチを要求するが、ＣＰＵ２が優先され、ＣＰＵ１待機信号ハイレベルになる。ここで、ＩＡＢにＣＰＵ２の命令フェッチのアドレスｍ＋８が出力される。

【０２５６】

Ｔ１４では、ＲＯＭからＩＤＢに読み出された次の命令コードをＣＰＵ２が取り込む。この場合に、ＣＰＵ１はアドレスＰＯＲＴをＩＡＢに出力する。ここで、入出力ポートのリードが３ステート必要であるため、ＣＰＵ１待機信号が再度ハイレベルとなる。

【０２５７】

Ｔ１５では、ＩＡＢの内容がＰＡＢに出力されて、入出力ポートのレジスタＰＯＲＴの内容がリードされる。このリードデータはＴ１７で得られる。一方、ＣＰＵ２が命令フェッチを要求するが、ＣＰＵ１がバスを使用中であるので、保留とされ、ＣＰＵ２待機信号がハイレベルになる。

【０２５８】

以上のように、本実施例においては、図６の実施例１に比較して、実行時間が長くなっているが、少なくとも１つのＣＰＵ１／２で実現する場合よりも短くできる。

【０２５９】

まず、割込やサブルーチン分岐などでＣＰＵ１／２の内部状態を退避したり、復帰したりする作業を除くことができる。また、２つのＣＰＵ１／２を単純に交互に動作するよりも短くできる。これは、ＲＯＭからＣＰＵ１／２が単位時間に実行できるより大きい量の命令を単位時間にリード可能にし、バスを使用しないでよい時間を作り、これを他方のＣＰＵ１／２が利用可能としているためである。

【０２６０】

これは、内蔵のＲＯＭにプログラムを配置する場合に特に有効であり、ＣＰＵがリード／ライトする大部分が命令のリードであることと、内部のメモリは外部のメモリに対して読み出し速度を向上しやすいこと、バスの幅を容易に広げられるためである。

【０２６１】

このとき、ＣＰＵ２は、前記ＤＭＡＣやＤＴＣのようなデータ処理装置またはデータ転送装置であっても、同様にＣＰＵ１の処理速度を低下することを最小限として、データ転送を行うことができる。特に、データ転送に先立って転送情報をＲＡＭから読み出す必要がある前記ＤＴＣに特に有効である。

【０２６２】

従って、本実施例のシングルチップマイクロコンピュータによれば、データ処理装置とし

てのＣＰＵ１，ＣＰＵ２、読み出し可能な記憶手段としてのＲＯＭを主要な構成とし、ＩＡＢ／ＩＤＢ、ＲＯＭ／ＲＡＭを共通化することにより、実施例１に加えて以下の作用効果を得ることができる。

【０２６３】

すなわち、ＣＰＵ１／２が単位時間に実行できる命令のバイト数より、大きなバイト数を単位時間でリード可能なＩＡＢ／ＩＤＢに、複数のＣＰＵ１／２を接続することにより、複数のＣＰＵ１／２が時分割的にＩＡＢ／ＩＤＢを利用して実質的に並立動作を可能にし、また共通のＲＯＭ／ＲＡＭを任意に配分して利用し、シングルチップマイクロコンピュータの処理速度を向上し、資源利用効率を向上することができる。

【０２６４】

以上、本発明者によってなされた発明を実施例１および２に基づき具体的に説明したが、本発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【０２６５】

たとえば、バスの具体的な回路構成についても種々変更可能であり、ＩＡＢ１、ＩＤＢ１とＰＡＢ、ＰＤＢを直結してもよい。この場合には、ＣＰＵ１とＣＰＵ２が並立的にプログラムを読み出して実行できるようにすればよく、これによって相互に動作を制約する頻度が高くなるが、物理的規模を縮小できる場合がある。

【０２６６】

また、ＲＡＭ１、ＲＡＭ２、ＲＡＭＰを１つの機能ブロックとしてもよい。

【０２６７】

さらに、ＣＰＵの数も３個以上にすることができ、この場合には複数のＣＰＵを第１のバスに接続して、バスを時分割で利用し、さらに別の単数または複数のＣＰＵを第２のバスに接続してもよい。

【０２６８】

また、内蔵のＲＯＭはマスクＲＯＭ、ＰＲＯＭの他、ＥＥＰＲＯＭ、フラッシュメモリなどのＣＰＵが実行すべきプログラムを格納するメモリであればよい。このＲＯＭ１とＲＯＭ２が異なっていてもよい。

【０２６９】

さらに、タイマ、ＳＣＩなどの割込を要求する機能ブロックあるいはパルス出力回路などの割込処理の対象となる機能ブロックの種類、機能、数などには何等制約されない。

【０２７０】

以上の説明では、主として本発明者によってなされた発明をその利用分野であるシングルチップマイクロコンピュータに適用した場合について説明したが、これに限定されるものではなく、その他の半導体集積回路装置にも適用可能であり、本発明は、少なくとも複数のデータ処理装置を内蔵した半導体集積回路装置に適用することができる。

【０２７１】

【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、以下のとおりである。

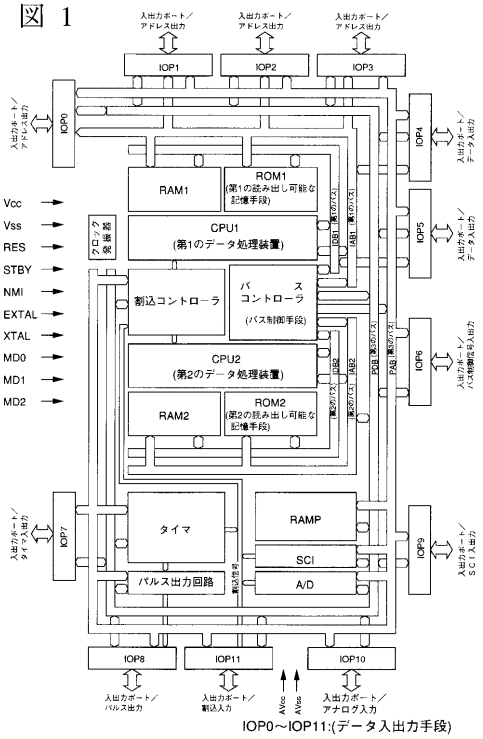
【０２７２】

(1).２つのデータ処理装置に並立したバスによって独立した２つの読み出し可能な記憶手段に格納した命令をリード可能にしたり、バスの幅を拡張してデータ処理装置が単位時間に実行可能な命令数以上の命令をリード可能にして、２つのデータ処理装置を並立的に動作可能な命令を供給する機能を内蔵するシングルチップマイクロコンピュータまたは半導体集積回路装置において、複数の記憶手段を同時に選択可能とし、複数の記憶手段の同時書込みまたは同時読み出しを行うことによって割込によるデータ処理のチャネル数の制約をなくし、さらにデータ処理内容を任意とすることができるので、任意の転送処理の実現、物理的規模の縮小、転送チャネル数の制約解除、あるいはシングルチップマイクロコンピュータまたは半導体集積回路装置の製造費用の削減が可能となる。

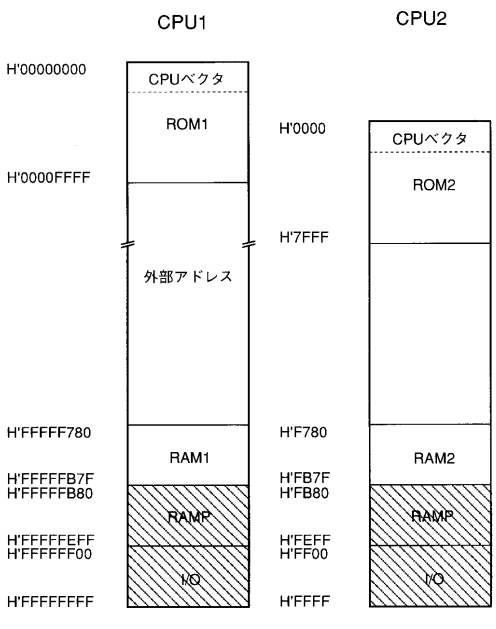
【０２７３】
(2).前記(1) により、第１、第２のデータ処理装置を同時に動作させることができるので、シングルチップマイクロコンピュータまたは半導体集積回路装置の処理速度の向上が可能となる。
【０２７４】
(3).前記シングルチップマイクロコンピュータまたは半導体集積回路装置のエミュレーション用プロセッサ、さらにエミュレータによれば、２つのデータ処理装置を独立に停止させることができるので、デバック効率の向上を図ることが可能となる。
【図面の簡単な説明】
【図１】本発明の実施例１であるシングルチップマイクロコンピュータを示すブロック図である。
【図２】実施例１のシングルチップマイクロコンピュータにおける全体のアドレスマップの説明図である。
【図３】実施例１におけるシングルチップマイクロコンピュータのバスの接続状態のブロック図である。
【図４】実施例１における割込コントローラの概略ブロック図である。
【図５】実施例１における割込要求ビットおよび割込許可回路の概略ブロック図である。
【図６】実施例１におけるバスの動作タイミング図である。
【図７】実施例１における低消費電力状態制御回路の概略ブロック図である。
【図８】実施例１における入出力ポートを制御するレジスタ構成の説明図である。
【図９】実施例１における入出力ポートの概略ブロック図である。
【図１０】実施例１におけるＰＲＯＭのアドレスマップである。
【図１１】実施例１におけるＰＲＯＭモードの主要部のブロック図である。
【図１２】実施例１におけるテストモードの主要部のブロック図である。
【図１３】実施例１におけるエミュレーション用プロセッサの概略ブロック図である。
【図１４】実施例１におけるエミュレータの概略ブロック図である。
【図１５】実施例１におけるシングルチップマイクロコンピュータの変形例を示すブロック図である。
【図１６】本発明の実施例２であるシングルチップマイクロコンピュータを示すブロック図である。
【図１７】実施例２のシングルチップマイクロコンピュータにおけるバスの動作タイミング図である。
【符号の説明】
ＣＰＵ１　中央処理装置（第１のデータ処理装置）
ＣＰＵ２　中央処理装置（第２のデータ処理装置）
ＲＯＭ１　リードオンリメモリ（第１の読み出し可能な記憶手段）
ＲＯＭ２　リードオンリメモリ（第２の読み出し可能な記憶手段）
ＲＡＭ１，ＲＡＭ２，ＲＡＭＰ　ランダムアクセスメモリ
ＳＣＩ　シリアルコミュニケーションインタフェース
Ａ／Ｄ　Ａ／Ｄ変換器
ＩＯＰ０～ＩＯＰ１１　入出力ポート（データ入出力手段）
ＩＡＢ１　内部アドレスバス（第１のバス）
ＩＡＢ２　内部アドレスバス（第２のバス）
ＰＡＢ　内部アドレスバス（第３のバス）
ＩＤＢ１　内部データバス（第１のバス）
ＩＤＢ２　内部データバス（第２のバス）
ＰＤＢ　内部データバス（第３のバス）
ＤＭＡＣ　ダイレクトメモリアクセスコントローラ
ＲＯＭ　リードオンリメモリ（読み出し可能な記憶手段）
ＲＡＭ　ランダムアクセスメモリ

ＩＡＢ　内部アドレスバス
ＩＤＢ　内部データバス

【図１】
図 1

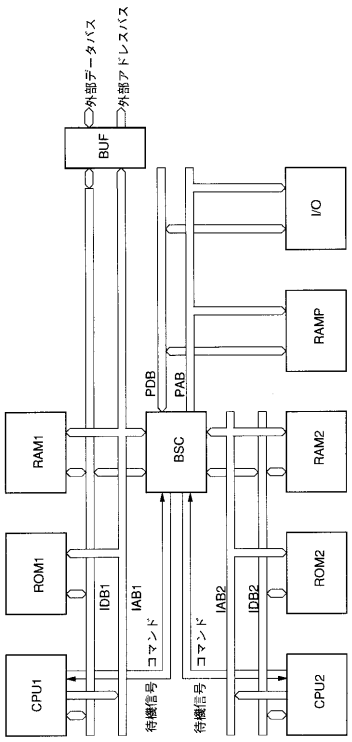


【図２】
図 2



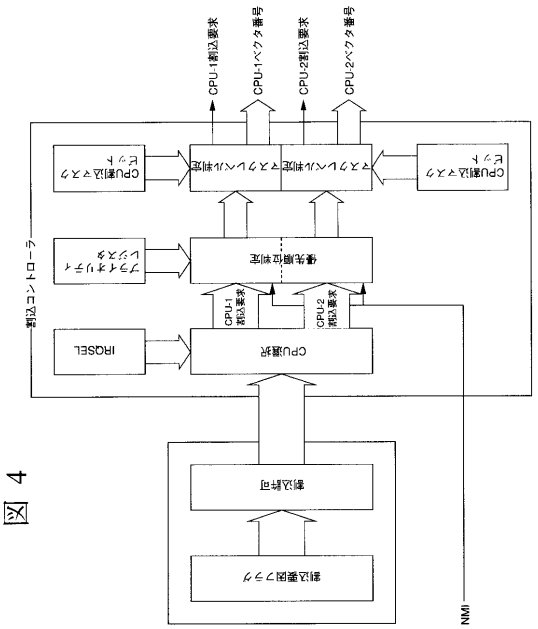
【図３】

図３



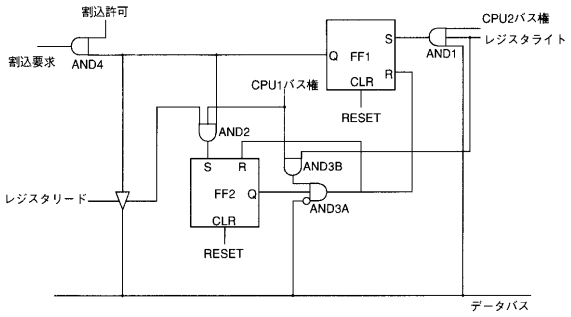
【図４】

図４



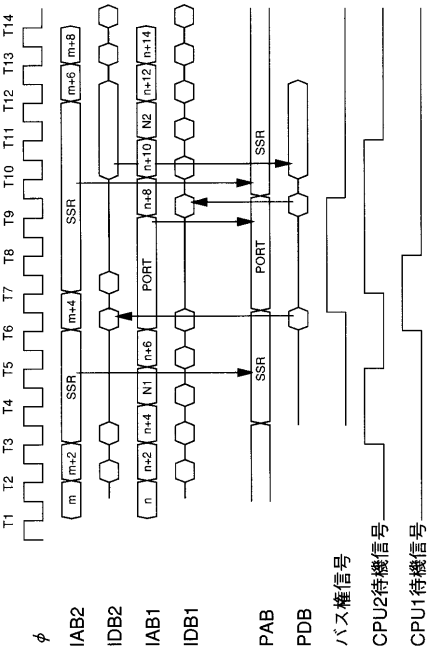
【図５】

図５



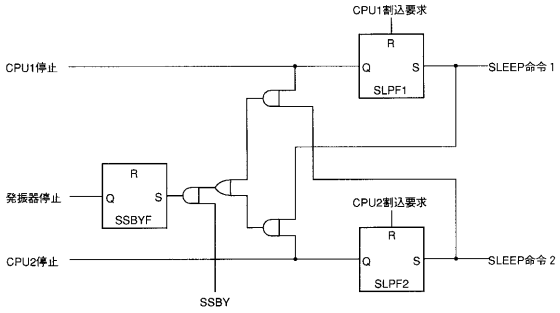
【図６】

図６



【図７】

図 7



【図８】

図 8

DDR

ビット：	7	6	5	4	3	2	1	0
	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
初期値：	0	0	0	0	0	0	0	0
R／W：	W	W	W	W	W	W	W	W

DR

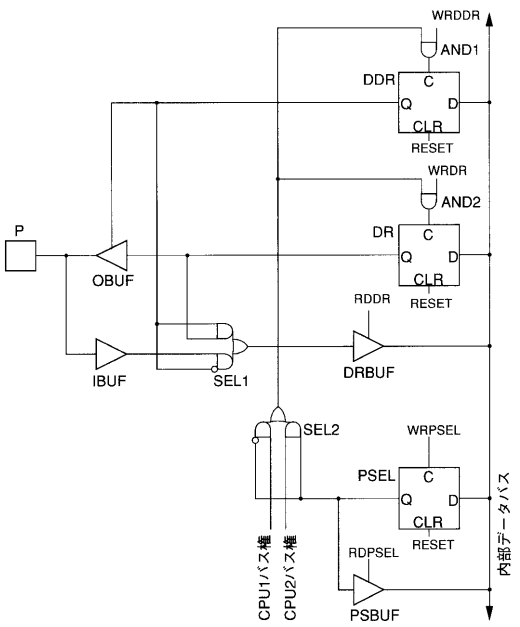
ビット：	7	6	5	4	3	2	1	0
	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
初期値：	0	0	0	0	0	0	0	0
R／W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PSEL

ビット：	7	6	5	4	3	2	1	0
	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0
初期値：	0	0	0	0	0	0	0	0
R／W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

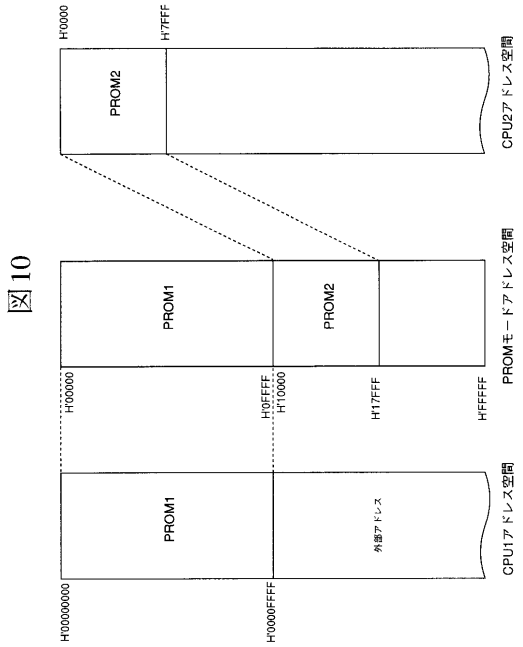
【図９】

図 9



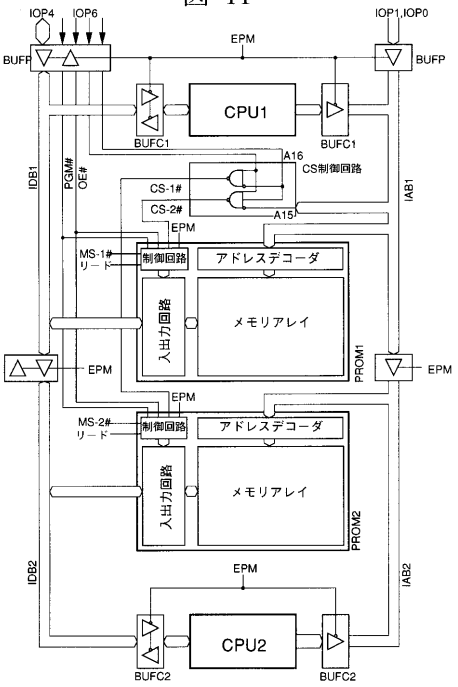
【図１０】

図 10



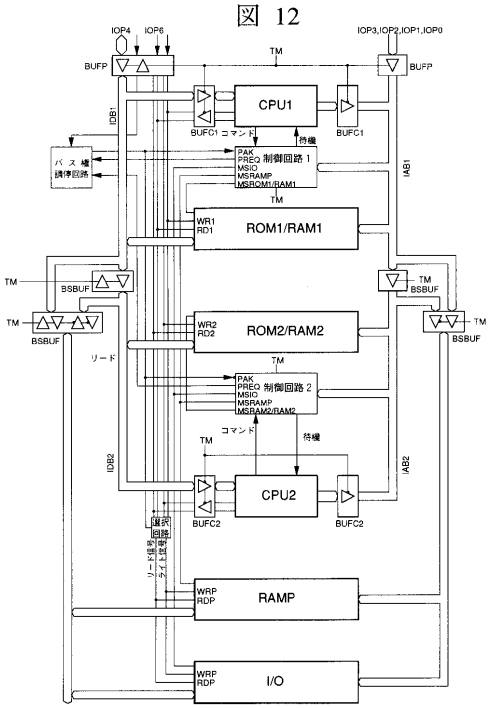
【図１１】

図 11



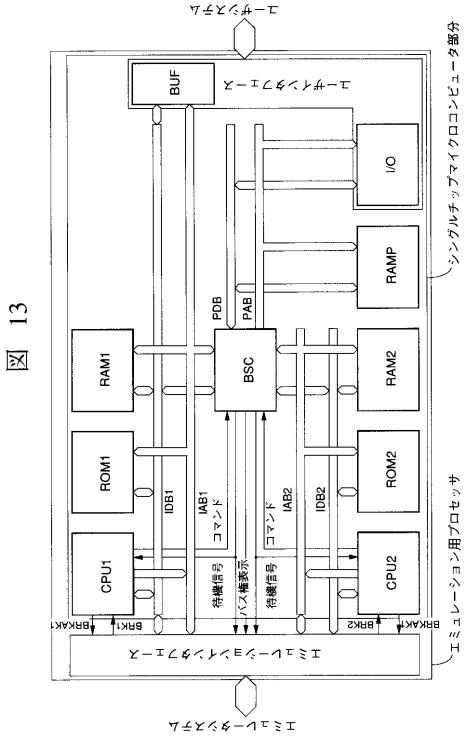
【図１２】

図 12



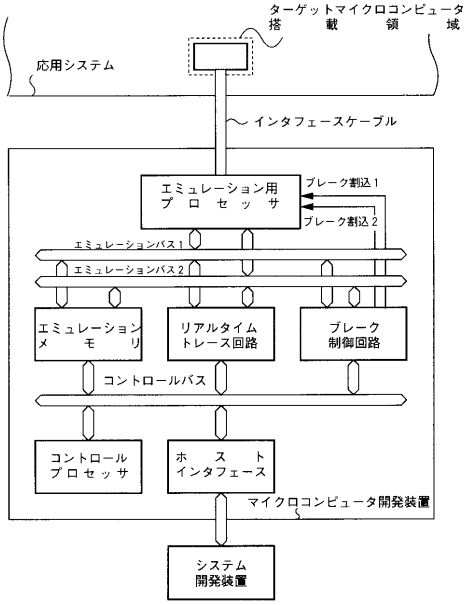
【図１３】

図 13



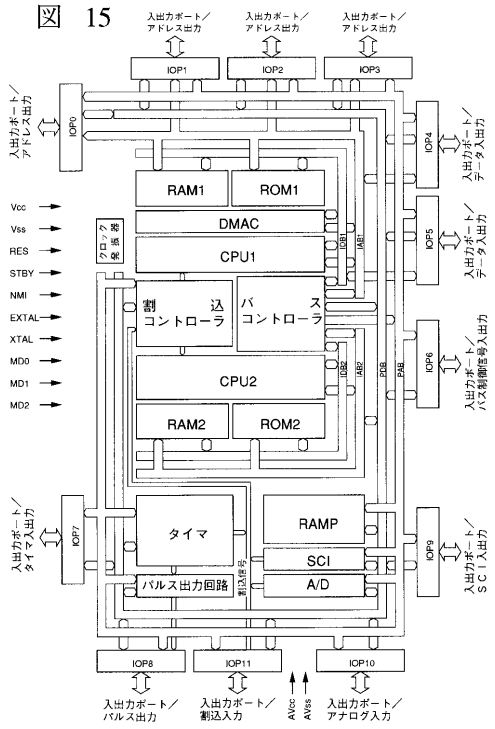
【図１４】

図 14



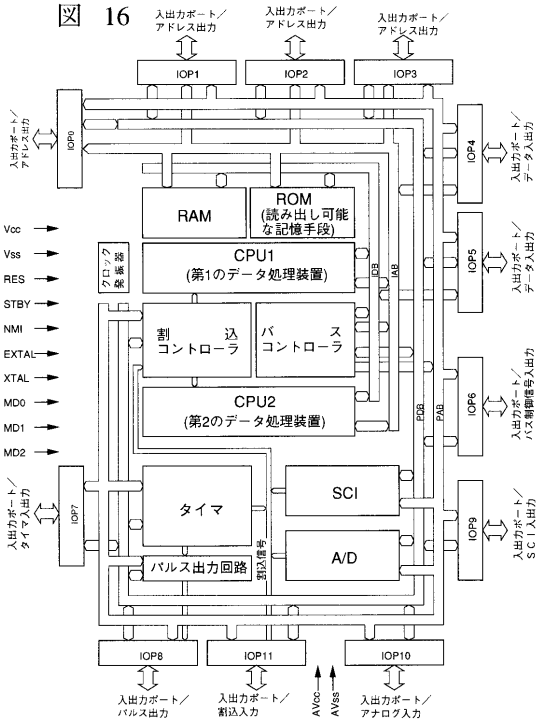
【図１５】

図 15



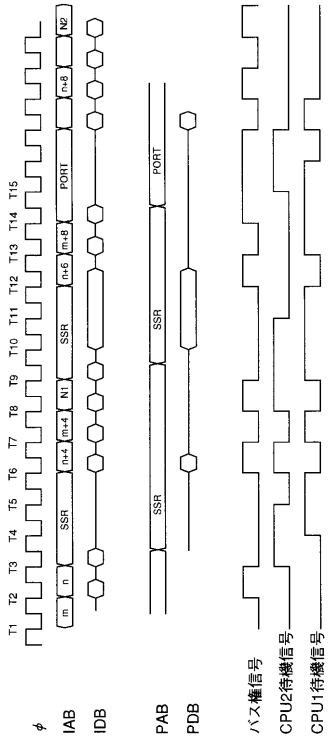
【図１６】

図 16



【図１７】

図 17



フロントページの続き

(56)参考文献 特開昭６２－２２１０６２（ＪＰ，Ａ）
特開昭６２－１５２０６４（ＪＰ，Ａ）
特開平０４－２６０９６１（ＪＰ，Ａ）
特開昭６１－１０１８６８（ＪＰ，Ａ）
特開平０６－１１９２３９（ＪＰ，Ａ）
特開昭６２－２１０５６４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 15/78