(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3608804号
(P3608804)
(45)発行日　平成17年1月12日(2005.1.12)　(24)登録日　平成16年10月22日(2004.10.22)

(51)Int.Cl.[7]　F I
G06F 13/42　G06F 13/42　350C
G06F 13/12　G06F 13/12　340G

請求項の数 8　(全 27 頁)

(21)出願番号　特願平5-112841
(22)出願日　平成5年5月14日(1993.5.14)
(65)公開番号　特開平6-324990
(43)公開日　平成6年11月25日(1994.11.25)
審査請求日　平成12年5月15日(2000.5.15)

前置審査

(73)特許権者　395015319
株式会社ソニー・コンピュータエンタテインメント
東京都港区南青山二丁目６番２１号
(74)代理人　100099324
弁理士　鈴木　正剛
(72)発明者　山本　靖之
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　石橋　俊哉
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

審査官　石井　茂和

最終頁に続く

(54)【発明の名称】バス制御装置

(57)【特許請求の範囲】
【請求項１】
中央演算処理装置が接続されるメインバスと、複数の周辺装置が接続されるサブバスとを接続するためのバス制御装置であって、
前記複数の周辺装置それぞれのタイミング制御条件を規定したタイミングテーブルを保持する記憶手段と、
前記中央演算処理装置と前記複数の周辺装置それぞれとの間のアクセスを制御する制御手段とを有し、
前記制御手段は前記中央演算処理装置から前記複数の周辺装置の一つに対するアクセス要求を受け付けると、前記タイミングテーブルに規定されたそのアクセス要求にかかる周辺装置のタイミング制御条件に従って前記アクセス要求を実行する、
バス制御装置。
【請求項２】
前記タイミング制御条件はバス占有時間を含み、前記制御手段は前記中央演算処理装置から前記複数の周辺装置の一つに対するアクセス要求を実行した後、少なくとも前記バス占有時間においてその周辺装置による前記サブバスの占有を継続させる、請求項１記載のバス制御装置。
【請求項３】
前記タイミング制御条件はリカバリタイムを含み、前記制御手段は前記中央演算処理装置からのアクセス要求が同一の周辺装置への連続したアクセス要求である場合には、前記リ

カバリタイムに従ってそのアクセス要求を実行する、請求項１または２記載のバス制御装置。
【請求項４】
中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、
前記周辺装置の動作タイミング情報を保持する記憶手段と、
前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、
前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、
前記記憶手段には、前記動作タイミング情報として前記周辺装置のアクセスタイム情報が保持され、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、その周辺装置のデータ書き込み要求に対するアクセスタイム経過後にデータ書き込み処理を実行し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、その周辺装置のアクセスタイム経過後にデータ読み出し処理を実行するバス制御装置。
【請求項５】
前記アクセスタイム情報は、リードディレイタイムおよびライトディレイタイムの情報を含み、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、前記ライトディレイタイム経過後にデータ書き込み処理を実行し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、前記リードディレイタイム経過後に前記周辺装置に対するデータ読み出し処理を実行する請求項４記載のバス制御装置。
【請求項６】
中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、
前記周辺装置の動作タイミング情報を保持する記憶手段と、
前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、
前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、
前記記憶手段には、前記動作タイミング情報として前記周辺装置のバス占有時間情報が保持され、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、データ書き込み処理を実行後、前記バス占有時間情報に従って前記サブバス上に書き込みデータを保持し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、データ読み込み処理実行後、前記バス占有時間情報に従って前記周辺装置へのアクセスを待機させるバス制御装置。
【請求項７】
前記バス占有時間情報は、ホールドタイムおよびフローティングタイムの情報を含み、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、データ書き込み処理を実行後、少なくとも前記ホールドタイムにおいて前記サブバス上に書き込みデータを保持し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合に

は、データ読み込み処理実行後、少なくとも前記フローティングタイムにおいて前記周辺装置へのアクセスを待機させる請求項６記載のバス制御装置。
【請求項８】
中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、
前記周辺装置の動作タイミング情報を保持する記憶手段と、
前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、
前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、
前記記憶手段には、前記動作タイミング情報として前記周辺装置のリカバリタイムの情報が保持され、前記制御手段は、前記中央演算処理装置から前記周辺装置に対するアクセス要求を受けた場合に、そのアクセス要求が同一の周辺装置の連続したアクセス要求である場合には、前記リカバリタイムの経過を待って、そのアクセス要求を実行するバス制御装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、例えばコンピュータなどに用いて好適なバス制御装置に関する。
【０００２】
【従来の技術】
従来のコンピュータにおいては、そのＣＰＵのバスアクセス速度より低速の周辺装置（例えば、インターフェイスボードなど）を使用する場合、その周辺装置のバスアクセスタイミングにあわせて、ＣＰＵのバスアクセスタイミングを遅くすると、装置全体の処理速度が低下するため、いわゆるウエイトコントローラなどを設け、ＣＰＵが出力したリード信号やライト信号などを周辺装置に供給するタイミングを調整するようになされている。
【０００３】
また、従来のコンピュータでは、そのＣＰＵのバス幅と異なるバス幅の周辺装置を使用する場合、即ち、例えば３２ビットのバス幅のＣＰＵに対し、８ビットのバス幅の周辺装置を使用する場合、データのパック／アンパックを行う専用のハード機構を設け、ＣＰＵが出力した３２ビットのデータを８ビットのデータに４分割して（アンパック）、周辺装置に供給し、また周辺装置が出力した８ビットのデータを４回読み込み、３２ビットのデータにして（パック）、ＣＰＵに供給するようになされている。
【０００４】
【発明が解決しようとする課題】
ところで、上述のウエイトコントローラは、コンピュータに接続する周辺装置にあわせて、個別に設計しなければならず、装置が高コスト化する課題があった。
【０００５】
さらに、周辺装置によっては、ＣＰＵが、周辺装置からデータを読み込む場合、ＣＰＵのデータのリードが終了しても、しばらくの間、データを出力し続けている、即ちフローティングタイムの長いものがある。また、周辺装置によっては、ＣＰＵが、周辺装置にデータを書き込む場合、ＣＰＵのデータのライト信号の出力が終了しても、しばらくの間、データバス上にデータを保持する必要のある、即ちホールドタイムの長いものがある。
【０００６】
このようなフローティングタイムやホールドタイムの長い周辺装置に対しては、上述のウエイトコントローラだけでは、動作タイミングを調整しきれず、その周辺装置にハード的な工夫を施さなければならなかった。
【０００７】
従って、装置が大型化、高コスト化する課題があった。
【０００８】
また、上述のパック／アンパックを行うハード機構も、ウエイトコントローラと同様に、

コンピュータに接続する周辺装置にあわせて、例えばパック／アンパックのトリガ発生論理などを個別に設計しなければならず、装置が高コスト化する課題があった。

【０００９】

さらに、例えば上述のように、３２ビットのバス幅のＣＰＵに対し、８ビットのバス幅の周辺装置を使用する場合、ＣＰＵが出力した３２ビットのデータを、アンパックし、４つの８ビットのデータにして、これらを周辺装置に書き込むとき、その周辺装置が、例えばメモリであれば、データを書き込むアドレスを１ずつ、４回インクリメントする必要があるが、周辺装置が、例えばいわゆるＦＩＦＯ型のバッファなどを有する、データの読み書きが固定アドレスで行われるものであれば、同一（固定）アドレスに対し、４つの８ビットのデータを書き込むようにする必要がある。

【００１０】

ここで、本明細書中においては、ＣＰＵとバス幅の異なる周辺装置に対してデータを読み書きする場合に、アドレスが、上述のようにインクリメントされたり、固定であったりすることをアドレス形式という。

【００１１】

しかしながら、上述のハード機構は、周辺装置のアドレス形式を考慮して、周辺装置に対してデータを読み書きするアドレスを指定するようにはなされていなかった。

【００１２】

従って、例えばアプリケーションプログラムなどは、周辺装置のアドレス形式を考慮して設計、製作しなければならないので、面倒であった。

【００１３】従って、本発明の課題は、装置を大型化、高コスト化することなく、ＣＰＵと周辺装置との間のデータのやりとりを制御するバス制御装置を提供することにある。

【００１４】

【課題を解決するための手段】

（１）上記課題に対応した本発明の第１のバス制御装置は、中央演算処理装置が接続されるメインバスと、複数の周辺装置が接続されるサブバスとを接続するためのバス制御装置であって、前記複数の周辺装置それぞれのタイミング制御条件を規定したタイミングテーブルを保持する記憶手段と、前記中央演算処理装置と前記複数の周辺装置それぞれとの間のアクセスを制御する制御手段とを有し、前記制御手段は前記中央演算処理装置から前記複数の周辺装置の一つに対するアクセス要求を受け付けると、前記タイミングテーブルに規定されたそのアクセス要求にかかる周辺装置のタイミング制御条件に従って前記アクセス要求を実行する、ことを特徴とする。

【００１５】

（２）この第１のバス制御装置において、前記タイミング制御条件がバス占有時間を含み、前記制御手段は前記中央演算処理装置から前記複数の周辺装置の一つに対するアクセス要求を実行した後、少なくとも前記バス占有時間においてその周辺装置による前記サブバスの占有を継続させる、ようにしてもよい。

（３）また、この第１のバス制御装置において、前記タイミング制御条件がリカバリタイムを含み、前記制御手段は前記中央演算処理装置からのアクセス要求が同一の周辺装置への連続したアクセス要求である場合には、前記リカバリタイムに従ってそのアクセス要求を実行する、ようにすることもできる。

【００１６】

（４）さらに、上記課題に対応した本発明の第２のバス制御装置は、中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、前記周辺装置の動作タイミング情報を保持する記憶手段と、前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、前記記憶手段には、前記動作タイミング情報として前記周辺装置のアクセスタイム情報が保持され、

前記制御手段は、

（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、その周辺装置のデータ書き込み要求に対するアクセスタイム経過後にデータ書き込み処理を実行し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、その周辺装置のアクセスタイム経過後にデータ読み出し処理を実行する、ことを特徴とする。
（５）この第２のバス制御装置において、前記アクセスタイム情報は、リードディレイタイムおよびライトディレイタイムの情報を含み、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、前記ライトディレイタイム経過後にデータ書き込み処理を実行し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、前記リードディレイタイム経過後に前記周辺装置に対するデータ読み出し処理を実行するようにしてもよい。
【００１７】
（６）さらに、上記課題に対応した本発明の第３のバス制御装置は、中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、前記周辺装置の動作タイミング情報を保持する記憶手段と、前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、前記記憶手段には、前記動作タイミング情報として前記周辺装置のバス占有時間情報が保持され、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、データ書き込み処理を実行後、前記バス占有時間情報に従って前記サブバス上に書き込みデータを保持し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、データ読み込み処理実行後、前記バス占有時間情報に従って前記周辺装置へのアクセスを待機させる、ことを特徴とする。
（７）この第３のバス制御装置において、前記バス占有時間情報は、ホールドタイムおよびフローティングタイムの情報を含み、
前記制御手段は、
（a）前記中央演算処理装置から前記周辺装置に対するデータ書き込み要求を受けた場合には、データ書き込み処理を実行後、少なくとも前記ホールドタイムにおいて前記サブバス上に書き込みデータを保持し、
（b）前記中央演算処理装置から前記周辺装置からのデータ読み出し要求を受けた場合には、データ読み込み処理実行後、少なくとも前記フローティングタイムにおいて前記周辺装置へのアクセスを待機させるようにしてもよい。
（８）さらに、上記課題に対応した本発明の第４のバス制御装置は、中央演算処理装置が接続されるメインバスと、周辺装置が少なくとも一つ接続されるサブバスとを接続するためのバス制御装置であって、前記周辺装置の動作タイミング情報を保持する記憶手段と、前記中央演算処理装置または前記周辺装置が出力するデータをラッチするデータバッファ手段と、前記中央演算処理装置と前記周辺装置との間のアクセスを制御する制御手段とを有し、前記記憶手段には、前記動作タイミング情報として前記周辺装置のリカバリタイムの情報が保持され、前記制御手段は、前記中央演算処理装置から前記周辺装置に対するアクセス要求を受けた場合に、そのアクセス要求が同一の周辺装置の連続したアクセス要求である場合には、前記リカバリタイムの経過を待って、そのアクセス要求を実行する、ことを特徴とする。
【００１８】
これら第１乃至第４のバス制御装置によれば、装置全体を大型化、高コスト化することな

く、中央演算処理装置と周辺装置との間のデータのやり取りのタイミングを制御することができる。また、動作タイミングに関する情報として実際に用いる情報の種類により、周辺装置の様々なタイミング特性に応じたアクセス制御が可能になる。
さらに、上記（１）から（８）に記載されたバス制御装置の記憶手段に記憶された周辺装置の動作タイミング情報を書き換え可能に構成することにより、例えば周辺装置の仕様変更や、新たな周辺装置の追加にも対応できる。
【００２４】
【実施例】
図１は、本発明のバス制御装置を適用したゲートウェイ４が設けられたコンピュータの一実施例の構成を示すブロック図である。ＣＰＵ１は、例えば３２ビットのＣＰＵで、ＲＯＭ５に記憶されているシステムプログラムや、ＲＡＭ２などに記憶されたアプリケーションプログラムにしたがって所定の処理を行う。ＲＡＭ２は、例えば３２ビット幅の高速ＤＲＡＭで、アプリケーションプログラムや装置の動作上必要なデータなどを記憶する。デバイス３は、例えば３２ビット幅の高速デバイスで、音声処理や画像処理などの所定の処理を行う。
【００２５】
ＣＰＵ１、ＲＡＭ２、およびデバイス３は、３２ビットの高速バスとしてのメインバス９に接続され、ＣＰＵ１と、ＲＡＭ２またはデバイス３との間では、メインバス９を介して、高速で３２ビットのデータがやりとりされるようになされている。
【００２６】
ＲＯＭ５は、例えば８ビット幅のＲＯＭで、システムプログラムを記憶している。バックアップメモリ（メモリ）６は、例えば１６ビット幅のメモリで、データなどのバックアップを記憶する。デバイス７は、例えば８ビット幅のデバイスで、音声処理や画像処理などの所定の処理を行う。デバイス８は、例えば１６ビット幅の入出力インターフェイスである。
【００２７】
ＲＯＭ５、メモリ６、デバイス７、および８は、いずれもその動作速度が、ＣＰＵ１より低速で、例えば１６ビットの汎用バスとしてのサブバス１０に接続されている。但し、８ビット幅のＲＯＭ５とデバイス７は、１６ビットのサブバス１０の下位８ビットに接続されている。
【００２８】
なお、図示していないがメインバス９およびサブバス１０は、データバス、アドレスバス、および制御バスから構成されている。
【００２９】
ゲートウェイ４は、メインバス９とサブバス１０との間に接続され、ＣＰＵ１と、ＲＯＭ５、メモリ６、デバイス７、または８との間のデータのやりとりを制御する。
【００３０】
即ち、ゲートウェイ４は、例えば図２に示すように、データバッファ１１、多重化器（ＭＵＸ）１２ａ乃至１２ｄ、タイミングテーブル記憶部１３、バス幅テーブル記憶部１４、カウンタ１５、コントローラ１６、メインバスアドレスデコーダ１７、アドレステーブル記憶部１８、サブバスアドレスデコーダ１９、ステイタスバッファ２０、アドレスバッファ２１から構成される。
【００３１】
なお、図２において、点線は、制御信号線を意味する。
【００３２】
データバッファ１１は、コントローラ１６に制御され、ＣＰＵ１からメインバス９を介して供給される３２ビットのデータをラッチし、その上位から８ビットごとの４つのデータを、ＭＵＸ１２ａ乃至１２ｄにそれぞれ出力する。また、データバッファ１１は、８ビットの多重化器１２ａ乃至１２ｄからのデータをラッチし、３２ビットのデータとしてメインバス９に出力する。

【００３３】
それぞれ８ビット幅で構成されているＭＵＸ１２ａ乃至１２ｄは、コントローラ１６に制御され、データバッファ１１からの４つの８ビットのデータをそれぞれラッチし、２つの１６ビットのデータ、または４つの８ビットのデータとしてサブバス１０に出力する。また、ＭＵＸ１２ａ乃至１２ｄは、サブバス１０からの２つの１６ビットのデータ、または４つの８ビットのデータをラッチし、３２ビットのデータとしてデータバッファ１１に出力する。
【００３４】
タイミングテーブル記憶部１３には、サブバス１０に接続されたＲＯＭ５、メモリ６、デバイス７、または８それぞれの、例えばリードディレイタイム、ライトディレイタイム、ホールドタイム、フローティングタイム、およびリカバリタイムが記述された、図３に示すようなタイミングテーブルが記憶されている。
【００３５】
なお、リードディレイタイムまたはライトディレイタイムとは、それぞれデータがリードまたはライトされるときのアクセスタイムを意味し、ホールドタイムとは、ライト信号の出力が終了しても、バス上にデータを保持する必要のある時間を意味する。また、フローティングタイムとは、データのリードが終了しても、バス上にデータを保持する必要のある時間を意味し、リカバリタイムとは、同一の周辺装置に連続してアクセスするときに必要なインターバル（１度目のアクセスが終了してから、次のアクセスまでに設ける必要のある時間間隔）を意味する。
【００３６】
バス幅テーブル記憶部１４には、サブバス１０に接続されたＲＯＭ５、メモリ６、デバイス７、または８が扱うデータ長（以下、バス幅という）と、そのアドレス形式が記述された、図４に示すようなバス幅テーブルが記憶されている。
【００３７】
なお、図４のバス幅テーブルにおいて、アドレス形式が固定となっている周辺装置（本実施例においては、デバイス７および８）は、例えばＦＩＦＯ型のデータバッファを内蔵しており、アンパックされて複数に分割されたデータの読み書きの際に、所定の同一アドレス（固定アドレス）を指定する必要のあるものである。また、アドレス形式が可変となっている周辺装置（本実施例においては、ＲＯＭ５およびメモリ６）は、アンパックされて複数に分割されたデータの読み書きの際に、分割されたデータの１つの読み書きの後、次のデータを読み書きするのに、アドレスをインクリメント（またはデクリメント）して指定する必要のあるものである。
【００３８】
カウンタ１５は、図示せぬクロック発生器からメインバス９およびコントローラ１６を介して供給されるシステムクロック（Ｓｙｓｔｅｍ　Ｃｌｏｃｋ）をカウントする。
【００３９】
コントローラ１６は、タイミングテーブル記憶部１３に記憶されたタイミングテーブル（図３）に基づいて、ＣＰＵ１からメインバス９を介して供給されるリード（Ｒｅａｄ）信号またはライト（Ｗｒｉｔｅ）信号のタイミングを変え、ＳＲｅａｄ信号またはＳＷｒｉｔｅ信号としてサブバス１０に出力する。また、コントローラ１６は、タイミングテーブル記憶部１３に記憶されたタイミングテーブル（図３）に基づき、所定のタイミングで、後述するＤａｔａＯｋ信号を出力する。
【００４０】
さらに、コントローラ１６は、バス幅テーブル記憶部１４に記憶されたバス幅テーブル（図４）に記述されたバス幅およびＣＰＵ１からメインバス９を介して供給されるＡｃｃｅｓｓＴｙｐｅ（ＣＰＵ１が扱うデータ長）に基づいて、データバッファ１１を制御し、データをパック／アンパックさせる。また、コントローラ１６は、バス幅テーブル記憶部１４に記憶されたバス幅テーブル（図４）に記述されたアドレス形式に基づいて、アドレスバッファ２１を制御する。

【００４１】
メインバスアドレスデコーダ１７は、ＣＰＵ１よりメインバス９を介して出力されたアドレス（Ａｄｄｒｅｓｓ）をデコードし、そのデコード結果と、アドレステーブル記憶部１８に記憶されているアドレスデコードテーブルとを比較する。そして、その比較結果から、ＣＰＵ１がアクセスしようとする周辺装置が、メインバス９に接続されたＲＡＭ２またはデバイス３であるか否かを判定し、それがＲＡＭ２またはデバイス３であると判定した場合には、メインバス９を介してＲＡＭ２またはデバイス３にチップセレクト信号を出力する。
【００４２】
アドレステーブル記憶部１８は、ＲＡＭ２、デバイス３、ＲＯＭ５、メモリ６、デバイス７、または８にそれぞれ割り当てられたアドレスのデコード結果が記述されたアドレスデコードテーブルを記憶している。
【００４３】
サブバスアドレスデコーダ１９は、ＣＰＵ１よりメインバス９およびアドレスバッファ２１を介して出力されたアドレスをデコードし、そのデコード結果と、アドレステーブル記憶部１８のアドレスデコードテーブルに記述されている、ＲＡＭ２、デバイス３、ＲＯＭ５、メモリ６、デバイス７、または８にそれぞれ割り当てられたアドレスのデコード結果とを比較する。そして、その比較結果から、ＣＰＵ１がアクセスしようとする周辺装置が、サブバス１０に接続されたＲＯＭ５、メモリ６、デバイス７、または８であるか否かを判定し、それがサブバス１０に接続された周辺装置のうちのいずれか（ＲＯＭ５、メモリ６、デバイス７、および８のうちのいずれか）であると判定した場合には、その周辺装置に、サブバス１０を介してチップセレクト信号を出力する。
【００４４】
ステイタスバッファ２０は、メインバスアドレスデコーダ１７およびサブバスアドレスデコーダ１９のうちのいずれかが、最後にチップセレクト信号を出力した周辺装置、即ちＣＰＵ１が最後にアクセスした周辺装置（周辺装置に割り当てられたアドレスのデコード結果）を記憶する。さらに、ステイタスバッファ２０は、その記憶値と、次にサブバスアドレスデコーダ１９がチップセレクト信号を出力した周辺装置とが同一である場合、即ちサブバス１０に接続された任意の周辺装置へのアクセスが連続して行われる場合、コントローラ１６に、後述するリカバリ処理をさせるための制御信号を出力する。
【００４５】
アドレスバッファ２１は、ＣＰＵ１からメインバス９を介して供給されたアドレスをラッチする。そして、アドレスバッファ２１は、コントローラ１６の制御に基づいて、ラッチしたアドレスをそのまま保持したり、あるいは所定のタイミングで、例えば１ずつインクリメント（またはデクリメント）する。
【００４６】
次に、その動作について説明するが、説明をわかりやすくするために、実際には同時に行われるタイミングの制御と、バス幅およびアドレス形式の制御とを分けて説明する。
【００４７】
まず、タイミングの制御について、図５および図６のタイミングチャートを参照して説明する。なお、図５および図６においては、信号Ｒｅａｄ，ＤａｔａＯｋ，ＣｈｉｐＳｅｌｅｃｔ，ＳＲｅａｄ，ＳＷｒｉｔｅは、アクティブローの論理で示してある。また、装置の各ブロックは、システムクロック（図５（ａ）および図６（ａ））の立ち上がりエッジのタイミングに同期して動作するものとする。
【００４８】
例えばＣＰＵ１がデバイス７から１バイト（８ビット）のデータを読み出す場合、図５に示すように、ＣＰＵ１よりデバイス７に割り当てられたアドレス（Ａｄｄｒｅｓｓ）（図５（ｃ））がメインバス９を介して出力され、このアドレスは、アドレスバッファ２１にラッチされるとともに、メインバスアドレスデコーダ１７に入力される。
【００４９】

アドレスバッファ２１にラッチされたアドレスは、信号ＳＡｄｄｒｅｓｓ（図５（ｈ））としてサブバス１０に出力されるとともに、サブバスアドレスデコーダ１９に出力される。

【００５０】

メインバスアドレスデコーダ１７およびサブバスアドレスデコーダ１９では、ＣＰＵ１からのアドレスがデコードされ、そのデコード結果と、アドレステーブル記憶部１８に記憶されたアドレスデコードテーブルとが比較される。

【００５１】

上述したようにいまＣＰＵ１からは、デバイス７に割り当てられたアドレスが出力されているので、サブバスアドレスデコーダ１９によって、ＣＰＵ１がアクセスしようとする周辺装置がデバイス７であることが検出され、サブバス１０を介してデバイス７に、チップセレクト信号（ＣｈｉｐＳｅｌｅｃｔ）（図５（ｆ））が出力される。

【００５２】

サブバスアドレスデコーダ１９のデコード結果は、ステイタスバッファ２０に記憶されるとともに、ステイタスバッファ２０を介してコントローラ１６に出力される。これにより、コントローラ１６によって、ＣＰＵ１のアクセスの対象がデバイス７であることが認識される。

【００５３】

さらに、ＣＰＵ１から、メインバス９を介してリード信号（Ｒｅａｄ）（図５（ｂ））およびバイトアクセスを示すアクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）が、コントローラ１６に供給される。

【００５４】

コントローラ１６は、ＣＰＵ１からのリード信号（Ｒｅａｄ）を受信すると、サブバス１０にリード信号（ＳＲｅａｄ）（図５（ｇ））を出力する。

【００５５】

ここで、図３のタイミングテーブルに示すように、デバイス７は、２クロックのリードディレイタイムを有する。即ち、デバイス７においては、チップセレクト信号およびリード信号（ＳＲｅａｄ）が供給されてから、データをサブバス１０に出力するまでに２クロックだけかかる。

【００５６】

そこで、コントローラ１６は、リード信号（ＳＲｅａｄ）をサブバス１０に出力すると、まずカウンタ１５をリセットし、カウンタ１５に、システムクロック（図５（ａ））のタイミングで、０からのカウントを開始させる。

【００５７】

さらに、コントローラ１６において、タイミングテーブル記憶部１３のタイミングテーブル（図３）に記述されたデバイス７のリードディレイタイムが参照され、カウンタ１５のカウント値がデバイス７のリードディレイタイムに等しくなるまで、ウエイト時間がおかれる。

【００５８】

従って、この場合、リード信号（ＳＲｅａｄ）が出力されてから、２クロックのウエイト時間がおかれることになる。

【００５９】

サブバス１０にリード信号（ＳＲｅａｄ）が出力されてから、２クロック分の時間が経過すると、デバイス７において、１バイトのデータ（ＳＤａｔａ）（図５（ｉ））が、１６ビットのサブバス１０の下位８ビットに出力されるとともに、コントローラ１６において、ＭＵＸ１２ｄおよびデータバッファ１１に、サブバス１０の下位８ビットのデータをラッチするように、ＣＰＵ１からの、８ビット（１バイト）のアクセスタイプを示す信号ＡｃｃｅｓｓＴｙｐｅに基づく制御信号が出力される。

【００６０】

これにより、デバイス７からの１バイトのデータ（ＳＤａｔａ）は、ＭＵＸ１２ｄを介し

てデータバッファ１１の下位８ビットにラッチされる。

【００６１】

デバイス７からのデータ（ＳＤａｔａ）が、データバッファ１１にラッチされると、メインバス９にデバイス７からのデータを出力することができることを示すＤａｔａＯｋ信号（図５（ｅ））が、コントローラ１６からメインバス９を介してＣＰＵ１に供給されるとともに、コントローラ１６からサブバス１０に出力されていたリード信号（ＳＲｅａｄ）（図５（ｇ））が停止される。

【００６２】

さらに、このとき、コントローラ１６は、カウンタ１５をリセットし、これによりカウンタ１５において、システムクロック（図５（ａ））のタイミングで、０からのカウントが開始される。

【００６３】

そして、データバッファ１１にラッチされたデータが、メインバス９上に出力され（図５（ｄ））、これがＣＰＵ１によって読み込まれる。

【００６４】

すると、ＣＰＵ１は、メインバス９へのリード信号（Ｒｅａｄ）（図５（ｂ））およびアドレス（Ａｄｄｒｅｓｓ）（図５（ｃ））の出力を停止する。ＣＰＵ１が、メインバス９へのリード信号（Ｒｅａｄ）（図５（ｂ））の出力を停止すると、コントローラ１６は、サブバスアドレスデコーダ１９からデバイス７へのチップセレクト信号（ＣｈｉｐＳｅｌｅｃｔ）（図５（ｆ））の出力を停止させる。

【００６５】

そして、コントローラ１６は、デバイス７からのデータがデータバッファ１１にラッチされた（データのリードが終了した）時刻から、デバイス７のフローティングタイムだけ経過し、デバイス７からサブバス１０へのデータ（ＳＤａｔａ）（図５（ｉ））の出力が停止すると、即ちタイミングテーブル（図３）に記述されたデバイス７のフローティングタイムに、カウンタ１５のカウント値が等しくなると、アドレスバッファ２１からサブバス１０へのアドレス（ＳＡｄｄｒｅｓｓ）（図５（ｈ））の出力を停止させる。

【００６６】

ところで、ＣＰＵ１は、上述したようにメインバス９へのアドレス（Ａｄｄｒｅｓｓ）（図５（ｃ））の出力を停止した時点で、デバイス７に対するアクセスが終了したものと認識する（実際にＣＰＵ１にとっては終了している）。

【００６７】

従って、ＣＰＵ１が、メインバス９へのアドレス（Ａｄｄｒｅｓｓ）（図５（ｃ））の出力を停止した直後、例えばサブバス１０に接続されたメモリ６にデータを書き込む場合を考えると、デバイス７からのデータがデータバッファ１１にラッチされた（データのリードが終了した）時刻から、デバイス７のフローティングタイム（２クロック）だけ経過するまでは、即ち図５において、P_1 で示す時刻を経過するまでは、デバイス７がサブバス１０上にデータ（ＳＤａｔａ）を出力しているので、このデータと、ＣＰＵ１がメモリ６へ出力するデータが、サブバス１０上でバッティングすることになる。

【００６８】

そこで、この場合、ゲートウェイ４においては、ＣＰＵ１が出力するデータを、データバッファ１１でラッチしておき、メモリ６へのデータの書き込みが終了したことを示す信号ＤａｔａＯｋを、コントローラ１６からＣＰＵ１に出力するようになされている。

【００６９】

これにより、ＣＰＵ１は、メモリ６へのデータの書き込みが終了した（実際には、終了していないが）と認識することになる。従って、メインバス９は、即座に開放され、ＣＰＵ１は、サブバス１０に接続された周辺装置にアクセスする処理を除く処理を実行することができるようになる。

【００７０】

一方、ゲートウェイ４においては、デバイス７からのデータがデータバッファ１１にラッ

チされた（データのリードが終了した）時刻から、デバイス７のフローティングタイムだけ経過するのを待って（時刻P_1（図５）が経過するのを待って）、コントローラ１６、サブバスアドレスデコーダ１９、アドレスバッファ２１からサブバス１０を介してメモリ６に、ライト信号（ＳＷｒｉｔｅ）、チップセレクト信号、またはアドレス（ＳＡｄｄｒｅｓｓ）がそれぞれ出力される。そして、データバッファ１１にラッチされたＣＰＵ１からのデータが、サブバス１０を介してメモリ６に出力されて書き込まれる。

【００７１】

従って、この場合、ＣＰＵ１は、デバイス７からのデータがデータバッファ１１にラッチされた（データのリードが終了した）時刻から、デバイス７のフローティングタイムだけ経過するのを待たずに、メモリ６へのデータの書き込み処理を行うことができるので、装置全体の処理速度を向上させることができる。

【００７２】

また、この場合、メモリ６へのデータの書き込み処理は、いわばＣＰＵ１からゲートウェイ４に渡され、ゲートウェイ４によって行われる。従って、ＣＰＵ１は、即座に他の処理を実行することができるようになるので、ＣＰＵ１と周辺装置との間のアクセスタイミングを考える必要がなくなり、これによりアプリケーションプログラムの開発を容易に行うことができるようになる。

【００７３】

次に、図６は、ＣＰＵ１からデバイス８へのデータの書き込み処理を説明するタイミングチャートである。まず、ＣＰＵ１よりデバイス８に割り当てられたアドレス（Ａｄｄｒｅｓｓ）（図６（ｃ））がメインバス９を介して出力され、このアドレスは、アドレスバッファ２１にラッチされるとともに、メインバスアドレスデコーダ１７に入力される。

【００７４】

アドレスバッファ２１にラッチされたアドレスは、信号ＳＡｄｄｒｅｓｓ（図５（ｈ））としてサブバス１０に出力されるとともに、サブバスアドレスデコーダ１９に出力される。

【００７５】

メインバスアドレスデコーダ１７およびサブバスアドレスデコーダ１９では、ＣＰＵ１からのアドレスがデコードされ、そのデコード結果と、アドレステーブル記憶部１８に記憶されたアドレスデコードテーブルとが比較される。

【００７６】

上述したようにいまＣＰＵ１からは、デバイス８に割り当てられたアドレスが出力されているので、サブバスアドレスデコーダ１９によって、ＣＰＵ１がアクセスしようとする周辺装置がデバイス８であることが検出され、サブバス１０を介してデバイス８に、チップセレクト信号（ＣｈｉｐＳｅｌｅｃｔ）（図６（ｆ））が出力される。

【００７７】

サブバスアドレスデコーダ１９のデコード結果は、ステイタスバッファ２０に記憶されるとともに、ステイタスバッファ２０を介してコントローラ１６に出力される。これにより、コントローラ１６によって、ＣＰＵ１のアクセスの対象がデバイス８であることが認識される。

【００７８】

さらに、ＣＰＵ１から、メインバス９を介してライト信号（Ｗｒｉｔｅ）（図６（ｂ））が、コントローラ１６に供給されるとともに、データ（Ｄａｔａ）がデータバッファ１１に供給されてラッチされる。

【００７９】

ＣＰＵ１からのデータ（Ｄａｔａ）が、データバッファ１１にラッチされると、コントローラ１６は、ＣＰＵ１に対し、データ（Ｄａｔａ）のデバイス８への書き込みが終了した（実際には終了していないが）ことを示すＤａｔａＯｋ信号（図６（ｅ））を、コントローラ１６からメインバス９を介してＣＰＵ１に供給する。

【００８０】

ＣＰＵ１は、ＤａｔａＯｋ信号を受信すると、データ（Ｄａｔａ）の書き込みが終了したと認識して、ライト信号（Ｗｒｉｔｅ）、アドレス（Ａｄｄｒｅｓｓ）、およびデータ（Ｄａｔａ）の出力を停止し、次の処理を実行する。

【００８１】

一方、コントローラ１６は、ＣＰＵ１からのライト信号（Ｗｒｉｔｅ）を受信すると、サブバス１０にライト信号（ＳＷｒｉｔｅ）（図６（ｇ））を出力するとともに、データバッファ１１に、ラッチしたデータを、信号ＳＤａｔａ（図６（ｉ））として、サブバス１０に出力するように、制御信号を出力する。

【００８２】

これにより、データバッファ１１からは、データ（ＳＤａｔａ）がサブバス１０上に出力される。

【００８３】

ここで、図３のタイミングテーブルに示すように、デバイス８は、３クロックのライトディレイタイムを有する。即ち、デバイス８においては、チップセレクト信号およびライト信号（ＳＷｒｉｔｅ）が供給されてから、データをサブバス１０から取り込むまでに３クロックだけかかる。

【００８４】

さらに、デバイス８は、１クロックのホールドタイムを有する。即ち、デバイス８に対してデータの書き込みを行う場合においては（デバイス８に対するデータの書き込みを補償するには）、ライト信号（ＳＷｒｉｔｅ）の供給を停止してから、少なくとも１クロック分の時間が経過するまでは、データ（ＳＤａｔａ）およびアドレス（ＳＡｄｄｒｅｓｓ）をサブバス１０上に供給し続ける必要がある。

【００８５】

そこで、コントローラ１６は、ライト信号（ＳＷｒｉｔｅ）（図６（ｇ））をサブバス１０に出力すると、まずカウンタ１５をリセットし、カウンタ１５に、システムクロック（図６（ａ））のタイミングで、０からのカウントを開始させる。

【００８６】

さらに、コントローラ１６において、タイミングテーブル記憶部１３のタイミングテーブル（図３）に記述されたデバイス８のライトディレイタイムが参照され、カウンタ１５のカウント値がデバイス８のライトディレイタイムに等しくなるまで、ウエイト時間がおかれる。

【００８７】

従って、この場合、ライト信号（ＳＷｒｉｔｅ）が出力されてから、３クロックのウエイト時間がおかれることになる。

【００８８】

サブバス１０へライト信号（ＳＷｒｉｔｅ）が出力されてから、３クロック分の時間が経過すると、コントローラ１６からのライト信号（ＳＷｒｉｔｅ）およびチップセレクト信号の出力が停止される。

【００８９】

コントローラ１６は、ライト信号（ＳＷｒｉｔｅ）（図６（ｇ））の出力を停止すると、カウンタ１５をリセットし、カウンタ１５に、システムクロック（図６（ａ））のタイミングで、０からのカウントを開始させる。

【００９０】

そして、コントローラ１６において、タイミングテーブル記憶部１３のタイミングテーブル（図３）に記述されたデバイス８のホールドタイムが参照され、カウンタ１５のカウント値がデバイス８のホールドタイムに等しくなるまで、ウエイト時間がおかれる。

【００９１】

従って、この場合、ライト信号（ＳＷｒｉｔｅ）が出力が停止されてから、１クロックのウエイト時間がおかれることになる。

【００９２】

ライト信号（ＳＷｒｉｔｅ）に出力が停止されてから、１クロック分の時間が経過すると（１クロック分以上の時間が経過した後の、システムクロックの最初の立ち上がりエッジのタイミング（図６において、P_2で示す時刻）で）、データバッファ１１またはアドレスバッファ２１それぞれからのデータ（ＳＤａｔａ）またはアドレス（ＳＡｄｄｒｅｓｓ）の出力が停止され、これによりデバイス８へのデータの書き込み処理が終了する。

【００９３】

以上のようにして、ホールドタイムを有するデバイス８に対するアクセスタイミングが制御され、デバイス８に対するデータの書き込みが保証される（デバイス８に、データが正確に書き込まれる）。

【００９４】

さらに、デバイス８へのデータの書き込み処理は、いわばＣＰＵ１からゲートウェイ４に渡され、ゲートウェイ４によって行われる。従って、上述したように、ＣＰＵ１は、即座に他の処理を実行することができるようになるので、装置全体の処理速度を向上させることができる。

【００９５】

次に、例えばメモリ６に２回連続してデータの書き込みが行われる場合（２回連続してメモリ６へのアクセスが行われる場合）、まず１回目は、上述のデバイス８へのデータの書き込み処理における場合と同様の処理が行われる。

【００９６】

即ち、ゲートウェイ４において、メモリ６のライトディレイタイム（図３）を考慮したタイミング制御がなされる（メモリ６のホールドタイム（図３）は０であるから、無視される）。

【００９７】

ここで、図３のタイミングテーブルに示したように、メモリ６は１０クロックのリカバリタイムを有する。即ち、メモリ６に２回連続してアクセスする場合においては、１回目のアクセスが終了してから、１０クロック分の時間を経過するまでは、２回目のアクセスが保証されない。

【００９８】

そこで、ゲートウェイ４においては、メモリ６への１回目のアクセスの終了後、メモリ６への２回目のアクセスが開始され、ステイタスバッファ２０に、サブバスアドレスデコーダ１９がチップセレクト信号を出力した周辺装置に割り当てられたアドレスのデコード結果が供給されると、そのデコード結果と、いまステイタスバッファ２０に記憶されている記憶値とが比較される。

【００９９】

ここで、１回目の、メモリ６へのデータの書き込み処理が終了したときには、ステイタスバッファ２０に、サブバスアドレスデコーダ１９の最新のデコード結果、即ち直前にアクセスされた周辺装置がメモリ６であることが記憶されている。

【０１００】

従って、この場合、サブバスアドレスデコーダ１９からいま出力されたデコード結果と、ステイタスバッファ２０にいま記憶されている記憶値とが同一であると判定される。

【０１０１】

すると、コントローラ１６は、リカバリ処理を行う。

【０１０２】

即ち、コントローラ１６は、まずカウンタ１５をリセットし、カウンタ１５に、システムクロックのタイミングで、０からのカウントを開始させる。

【０１０３】

同時に、コントローラ１６において、タイミングテーブル記憶部１３のタイミングテーブル（図３）に記述されたメモリ６のリカバリタイムが参照され、それが０より大きい値に設定されていれば、カウンタ１５のカウント値がメモリ６のリカバリタイムに等しくなるまで、ウエイト時間がおかれる。

【０１０４】
従って、この場合、メモリ６への１回目のアクセスが終了してから、１０クロック分のウエイト時間（インターバル）がおかれ、その後に、メモリ６に対する２回目のデータの書き込みが行われる。
【０１０５】
以上のようにして、リカバリタイムを有するメモリ６に対するアクセスタイミングが制御され、メモリ６に対して、連続してアクセスが行われても、その動作が保証される。
【０１０６】
また、ゲートウェイ４によるメモリ６への１回目のアクセスの終了後、ＣＰＵ１がメモリ６に対して２回目の書き込みデータとして出力するデータは、ゲートウェイ４のデータバッファ４で、即座にラッチされる。
【０１０７】
すると、コントローラ１６からは、信号ＤａｔａＯｋが、ＣＰＵ１に出力される。
【０１０８】
これにより、ＣＰＵ１は、メモリ６へのデータの書き込みが終了した（実際には、データバッファ１１にラッチされただけで、メモリ６へのデータの書き込みは終了していない）と認識し、メインバス９は開放される。従って、ＣＰＵ１は、ゲートウェイ４によるメモリ６への１回目のアクセスの終了後、即座に他の処理を実行することができるので、装置全体の処理速度を向上させることができる。
【０１０９】
なお、タイミングテーブル記憶部１３に記憶されたタイミングテーブル（図３）は、所定のプログラムをＣＰＵ１に実行させることによって、書き換えることができるようになされている。従って、例えばＲＯＭ５、バックアップメモリ６、デバイス７，８の仕様が変更されたり、新たな周辺装置が追加されても、タイミングテーブルを書き換えるだけで済み、例えば今まで使用していたアプリケーションソフトの互換性がなくなり、使用不可能となることが防止される。
【０１１０】
次に、バス幅およびアドレス形式の制御について、図７および図８のタイミングチャートを参照して説明する。なお、図７および図８においては、信号Ｒｅａｄ，ＤａｔａＯｋ，ＣｈｉｐＳｅｌｅｃｔ，ＳＲｅａｄ，ＳＷｒｉｔｅは、アクティブローの論理で示してある。また、装置の各ブロックは、システムクロック（図７（ａ）および図８（ａ））の立ち上がりエッジのタイミングに同期して動作するものとする。
【０１１１】
例えばＣＰＵ１が、８ビットのＲＯＭ５から３２ビットのデータを読み出す場合、図７に示すように、ＣＰＵ１より、ＲＯＭ５から読み出そうとするデータのアドレス（Ａｄｄｒｅｓｓ）（図７（ｃ））がメインバス９を介して出力され、このアドレスは、アドレスバッファ２１にラッチされるとともに、メインバスアドレスデコーダ１７に入力される。
【０１１２】
アドレスバッファ２１にラッチされたアドレスは、信号ＳＡｄｄｒｅｓｓ（図７（ｈ））としてサブバス１０に出力されるとともに、サブバスアドレスデコーダ１９に出力される。
【０１１３】
メインバスアドレスデコーダ１７およびサブバスアドレスデコーダ１９では、ＣＰＵ１からのアドレスがデコードされ、そのデコード結果と、アドレステーブル記憶部１８に記憶されたアドレスデコードテーブルとが比較される。
【０１１４】
上述したようにＣＰＵ１からは、ＲＯＭ５のアドレスが出力されているので、サブバスアドレスデコーダ１９によって、ＣＰＵ１がアクセスしようとする周辺装置がＲＯＭ５であることが検出され、サブバス１０を介してＲＯＭ５に、チップセレクト信号（ＣｈｉｐＳｅｌｅｃｔ）（図７（ｆ））が出力される。

【０１１５】
サブバスアドレスデコーダ１９のデコード結果は、ステイタスバッファ２０に記憶されるとともに、ステイタスバッファ２０を介してコントローラ１６に出力される。これにより、コントローラ１６によって、ＣＰＵ１のアクセスの対象がＲＯＭ５であることが認識される。
【０１１６】
さらに、ＣＰＵ１から、メインバス９を介してリード信号（Ｒｅａｄ）（図７（ｂ））および読み出すデータ長が、３２ビットであることを示すアクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）が、コントローラ１６に供給される。
【０１１７】
コントローラ１６は、ＣＰＵ１からのリード信号（Ｒｅａｄ）を受信すると、サブバス１０にリード信号（ＳＲｅａｄ）（図７（ｇ））を出力する。
【０１１８】
これにより、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ）から、８ビットのデータ（ＳＤａｔａ）（図７（ｉ））が、１６ビットのサブバス１０の下位８ビットに出力される。
【０１１９】
ここで、図４のバス幅テーブルに示すように、ＲＯＭ５は、そのバス幅が８ビットであり、アドレス形式が可変とされている。即ち、ＲＯＭ５から、アクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）で示される３２ビットのデータを読み出すには、バス幅単位としての８ビット単位で、アドレスをインクリメント（またはデクリメント）しながら、４回読み出す処理を行う必要がある。
【０１２０】
そこで、コントローラ１６は、ＣＰＵ１からのアクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）およびバス幅テーブル記憶部１４のバス幅テーブル（図４）に記述されたＲＯＭ５のバス幅およびアドレス形式に基づいて、次のような制御を行う。
【０１２１】
即ち、３２ビットのデータが４つの８ビットに分割されたデータを、上位の８ビットからd_4，d_3，d_2，d_1と表した場合、この４つの８ビットのデータd_4，d_3，d_2，d_1が、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ＋３），（ＳＡｄｄｒｅｓｓ＋２），（ＳＡｄｄｒｅｓｓ＋１），（ＳＡｄｄｒｅｓｓ）にそれぞれ書き込まれているとすると、コントローラ１６は、まず上述のようにしてサブバス１０に出力された、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ）からの８ビットのデータを、ＭＵＸ１２ｄを介して３２ビットのデータバッファ１１の下位８ビットにラッチさせる。
【０１２２】
そして、コントローラ１６は、アドレスバッファ２１にラッチされているアドレス（ＳＡｄｄｒｅｓｓ（＝Ａｄｄｒｅｓｓ））を1だけインクリメントさせ、アドレス（ＳＡｄｄｒｅｓｓ＋１）としてサブバス１０を介してＲＯＭ５に出力させる。
【０１２３】
これにより、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ＋１）から、８ビットのデータがサブバス１０に出力される。この８ビットのデータは、ＭＵＸ１２ｃを介してデータバッファ１１に、その下位８ビットの次の８ビットのデータとしてラッチされる。
【０１２４】
以下、同様にして、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ＋２），（ＳＡｄｄｒｅｓｓ＋３）から、８ビットのデータが読み出され、データバッファ１１に順次ラッチされる。なお、このとき、コントローラ１６は、アドレスバッファ２１に記憶されたアドレスをインクリメントするタイミングに基づいて、図７（ｇ）に示すように、ＲＯＭ５に対するリード信号（ＳＲｅａｄ）をＯＮ／ＯＦＦする。
【０１２５】
以上のようにして、ＲＯＭ５のアドレス（ＳＡｄｄｒｅｓｓ），（ＳＡｄｄｒｅｓｓ＋１），（ＳＡｄｄｒｅｓｓ＋２），（ＳＡｄｄｒｅｓｓ＋３）からそれぞれ読み出された８

ビットのデータが、データバッファ１１にラッチされ、３２ビットのデータとされると、即ち４つの８ビットのデータが３２ビットのデータにパックされると、コントローラ１６は、メインバス９を介してＣＰＵ１に信号ＤａｔａＯｋ（図７（ｅ））を出力し、さらにデータバッファ１１にラッチされている３２ビットのデータ（図７（ｄ））をメインバス９に出力させる。

【０１２６】

ＣＰＵ１は、信号ＤａｔａＯｋを受信すると、メインバス９上の３２ビットのデータを取り込み、これによりＲＯＭ５からの３２ビットのデータの読み込みが終了する。

【０１２７】

なお、バス幅テーブル記憶部１４に記憶されたバス幅テーブル（図４）は、所定のプログラムをＣＰＵ１に実行させることによって、書き換えることができるようになされている。従って、例えばＲＯＭ５が８ビットのＲＯＭから、１６ビットのものへ変更された場合には、バス幅テーブル（図４）のＲＯＭ５のバス幅を８ビットから１６ビットに変えるだけでよい。

【０１２８】

この場合、ＲＯＭ５から２つの１６ビットのデータが読み出され、そのうちの一方（例えば、アドレスが下位の１６ビットのデータ）は、ＭＵＸ１２ｃおよび１２ｄを介して、データバッファ１１の下位１６ビットにラッチされ、その他方（例えば、アドレスが上位の１６ビットのデータ）は、ＭＵＸ１２ａおよび１２ｂを介して、データバッファ１１の上位１６ビットにラッチされて、３２ビットにパックされる。

【０１２９】

次に、ＣＰＵ１によって、デバイス７への３２ビットのデータの書き込みが行われる場合、図８に示すように、ＣＰＵ１より、デバイス７へデータを書き込もうとするアドレス（Ａｄｄｒｅｓｓ）（図８（ｃ））がメインバス９を介して出力され、このアドレスは、アドレスバッファ２１にラッチされるとともに、メインバスアドレスデコーダ１７に入力される。

【０１３０】

アドレスバッファ２１にラッチされたアドレスは、信号ＳＡｄｄｒｅｓｓ（図８（ｈ））としてサブバス１０に出力されるとともに、サブバスアドレスデコーダ１９に出力される。

【０１３１】

メインバスアドレスデコーダ１７およびサブバスアドレスデコーダ１９では、ＣＰＵ１からのアドレスがデコードされ、そのデコード結果と、アドレステーブル記憶部１８に記憶されたアドレスデコードテーブルとが比較される。

【０１３２】

上述したようにＣＰＵ１からは、デバイス７のアドレスが出力されているので、サブバスアドレスデコーダ１９によって、ＣＰＵ１がアクセスしようとする周辺装置がデバイス７であることが検出され、サブバス１０を介してデバイス７に、チップセレクト信号（ＣｈｉｐＳｅｌｅｃｔ）（図８（ｆ））が出力される。

【０１３３】

サブバスアドレスデコーダ１９のデコード結果は、ステイタスバッファ２０に記憶されるとともに、ステイタスバッファ２０を介してコントローラ１６に出力される。これにより、コントローラ１６によって、ＣＰＵ１のアクセスの対象がデバイス７であることが認識される。

【０１３４】

さらに、ＣＰＵ１から、メインバス９を介してライト信号（Ｗｒｉｔｅ）（図８（ｂ））および書き込むデータ長が、３２ビットであることを示すアクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）が、コントローラ１６に供給されるとともに、デバイス７へ書き込む３２ビットのデータ（Ｄａｔａ）（図８（ｄ））がメインバス９に出力される。

【０１３５】

コントローラ１６は、ＣＰＵ１からのライト信号（Ｗｒｉｔｅ）を受信すると、メインバス９上の３２ビットのデータをデータバッファ１１にラッチさせるとともに、サブバス１０にライト信号（ＳＷｒｉｔｅ）（図８（ｇ））を出力する。

【０１３６】

さらに、コントローラ１６は、ＣＰＵ１に信号ＤａｔａＯｋを出力する。これにより、ＣＰＵ１は、デバイス７へのデータの書き込みが終了した（実際には、終了していないが）と認識し、メインバス９は、開放される。従って、その後ＣＰＵ１は、その他の処理を、即座に実行することができるので、装置全体の処理速度を向上させることができる。

【０１３７】

ここで、図４のバス幅テーブルに示すように、デバイス７は、そのバス幅が８ビットであり、アドレス形式が固定とされている。即ち、デバイス７へ、アクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）で示される３２ビットのデータを書き込むには、これをデバイス７のバス幅単位としての８ビットのデータに４分割し、この４つの８ビットのデータを、アドレスを固定して書き込むようにする必要がある。

【０１３８】

そこで、コントローラ１６は、ＣＰＵ１からのアクセスタイプ信号（ＡｃｃｅｓｓＴｙｐｅ）と、バス幅テーブル記憶部１４のバス幅テーブル（図４）に記述されたデバイス７のバス幅およびアドレス形式に基づいて、次のような制御を行う。

【０１３９】

即ち、コントローラ１６は、データバッファ１１にラッチされている３２ビットのデータを、８ビット幅のＭＵＸ１２ａ乃至１２ｄにそれぞれ読み出させることにより、４つの８ビットのデータにアンパックする。

【０１４０】

そして、コントローラ１６は、まずＭＵＸ１２ｄに読み出させた、３２ビットのデータのうちのＬＳＢ側の８ビットのデータを、サブバス１０の下位８ビットに出力させる。これにより、デバイス７のアドレス（ＳＡｄｄｒｅｓｓ）に、８ビットのデータが書き込まれる。

【０１４１】

さらに、コントローラ１６は、アドレスバッファ２１に、記憶しているアドレス（ＳＡｄｄｒｅｓｓ）を保持させたまま、ＭＵＸ１２ｃに読み出させた、３２ビットのデータのうちのＬＳＢ側の８ビットの次の上位８ビットのデータを、サブバス１０の下位８ビットに出力させる。これにより、デバイス７のアドレス（ＳＡｄｄｒｅｓｓ）に、８ビットのデータが書き込まれる。

【０１４２】

以下、同様にして、デバイス７のアドレス（ＳＡｄｄｒｅｓｓ）に、ＭＵＸ１２ｂおよび１２ａにデータバッファ１１から読み出させた８ビットのデータが順次書き込まれる。なお、このとき、コントローラ１６は、ＭＵＸ１２ａ乃至１２ｄに、データバッファ１１からデータを読み出させるタイミングに基づいて、図８（ｇ）に示すように、デバイス７に対するライト信号（ＳＷｒｉｔｅ）をＯＮ／ＯＦＦする。

【０１４３】

デバイス７は、例えばＦＩＦＯ型のバッファを内蔵しており、このバッファに、アドレス（ＳＡｄｄｒｅｓｓ）に対して書き込まれた４つの８ビットのデータが、順次スタックされ（積み込まれ）、その後、デバイス７において、所定の処理が施される。

【０１４４】

なお、例えばデバイス７が、バス幅８ビットのものから、１６ビットのものへ変更された場合には、上述した場合と同様にして、バス幅テーブル（図４）のデバイス７のバス幅を８ビットから１６ビットに変えるだけでよい。

【０１４５】

この場合、まずデータバッファ１１にラッチされた３２ビットのデータのうち、下位１６ビットのデータが、ＭＵＸ１２ｃおよび１２ｄによって読み出され、さらにその上位１６

ビットのデータが、ＭＵＸ１２ａおよび１２ｂによって読み出されることによって、３２ビットのデータが、２つの１６ビットのデータにアンパックされる。

【０１４６】

従って、周辺装置のバス幅またはアドレス方式が変更されたり、新たな周辺装置が追加されても、バス幅テーブルを書き換えるだけで済み、例えば今まで使用していたアプリケーションソフトの互換性がなくなり、使用不可能となることが防止される。

【０１４７】

以上説明を分かりやすくするために、タイミングの制御と、バス幅およびアドレス形式の制御とを分けて説明したが、ゲートウェイ４では、タイミングテーブル（図３）およびバス幅テーブル（図４）に基づいて、図９のフローチャートに示すように、これらの制御が、並列して（同時に）行われるようになされている。

【０１４８】

即ち、ゲートウェイ４においては、まずステップＳ１において、ＣＰＵ１からメインバス９を介してライト信号またはリード信号とともにアドレスが出力されるのが待たれ、ＣＰＵ１からライト信号またはリード信号とともにアドレスが出力されると、ステップＳ２に進み、周辺装置に対し、ＣＰＵ１からのアドレスが出力されるとともに、そのアドレスのデコードが行われ、そのデコード結果に対応する周辺装置にチップセレクト信号が出力される。

【０１４９】

そして、ステップＳ３に進み、ＣＰＵ１からのアドレスをデコードした結果から、ＣＰＵ１がアクセスしようとしている周辺装置が、サブバス１０に接続されたものか否かが判定される。

【０１５０】

ステップＳ３において、ＣＰＵ１がアクセスしようとしている周辺装置が、サブバス１０に接続されたものでないと判定された場合、即ちＣＰＵ１がアクセスしようとしている周辺装置が、メインバス９に接続されたものである場合、上述のリカバリ処理を行う必要がないため、ステップＳ４に進み、ステイタスバッファ２０がクリアされ、ステップＳ５に進む。

【０１５１】

ステップＳ５において、メインバス９に接続された周辺装置に対するＣＰＵ１のアクセスが行われ、それが終了すると、ステップＳ１に戻り、再びステップＳ１からの処理を繰り返す。

【０１５２】

また、ステップＳ３において、ＣＰＵ１がアクセスしようとしている周辺装置が、サブバス１０に接続されたものであると判定された場合、ステップＳ６，Ｓ７に順次進み、タイミングテーブル記憶部１３、バス幅テーブル記憶部１４から、タイミングテーブル、バス幅テーブルがそれぞれ読み出される。

【０１５３】

そして、ステップＳ８に進み、ＣＰＵ１のアクセスタイプと、ＣＰＵ１がアクセスしようとしている周辺装置のバス幅が等しいか否かが判定される。ステップＳ８において、ＣＰＵ１のアクセスタイプと、ＣＰＵ１がアクセスしようとしている周辺装置のバス幅が等しいと判定された場合、ステップＳ１０に進み、変数Ｎに１がセットされ、ステップＳ１１に進む。

【０１５４】

また、ステップＳ８において、ＣＰＵ１のアクセスタイプと、ＣＰＵ１がアクセスしようとしている周辺装置のバス幅が等しくないと判定された場合、ステップＳ９に進み、変数Ｎにパック／アンパックを行う回数がセットされ、ステップＳ１１に進む。

【０１５５】

ステップＳ１１においては、ステップ１でＣＰＵ１より出力された信号がライト信号であるか、またはリード信号であるかが判定される。

【０１５６】
ステップＳ１１において、ステップ１でＣＰＵ１より出力された信号がリード信号であると判定された場合、ステップＳ１３に進み、周辺装置からのデータのリード処理が行われる。
【０１５７】
即ち、ステップＳ１３では、まず図１０に示すステップＳ２１において、リカバリ処理が行われる。
【０１５８】
ここで、ステップＳ１３のリカバリ処理は、図１１のフローチャートに示すように行われる。
【０１５９】
即ち、まずステップＳ３１において、ステイタスバッファ２１が参照され、いまＣＰＵ１がアクセスしようとしている周辺装置が、その直前にＣＰＵ１がアクセスした周辺装置であるか否かが判定される。
【０１６０】
ステップＳ３１において、いまＣＰＵ１がアクセスしようとしている周辺装置が、その直前にＣＰＵ１がアクセスした周辺装置でないと判定された場合、ステップＳ３２に進み、ステイタスバッファ２０に、いまＣＰＵ１がアクセスしようとしている周辺装置が記憶され、処理を終了する。
【０１６１】
また、ステップＳ３１において、いまＣＰＵ１がアクセスしようとしている周辺装置が、その直前にＣＰＵ１がアクセスした周辺装置であると判定された場合、ステップＳ３３に進み、タイミングテーブル（図３）が参照され、いまＣＰＵ１がアクセスしようとしている周辺装置がリカバリタイムを有するか否かが判定される。
【０１６２】
ステップＳ３３において、いまＣＰＵ１がアクセスしようとしている周辺装置がリカバリタイムを有さないと判定された場合、処理を終了する。
【０１６３】
ステップＳ３３において、いまＣＰＵ１がアクセスしようとしている周辺装置がリカバリタイムを有すると判定された場合、ステップＳ３４に進み、そのリカバリタイム分のウエイト時間がおかれ、処理を終了する。
【０１６４】
以上のようにして、ステップＳ３１乃至Ｓ３４のリカバリ処理が終了すると、即ち図１０のステップＳ２１の処理が終了すると、ステップＳ２２に進み、ＣＰＵ１が、データを読み出そうとしている周辺装置にサブバス１０を介してリード信号（ＳＲｅａｄ）が出力され、ステップＳ２３に進む。
【０１６５】
ステップＳ２３において、タイミングテーブル記憶部１３のタイミングテーブルに記述された、ＣＰＵ１がデータを読み出そうとしている周辺装置のリードディレイタイム分のウエイト時間がおかれ、ステップＳ２４に進み、周辺装置からデータが読み出され、それがデータバッファ１１に、その下位ビットの方からラッチされるとともに、リード信号（ＳＲｅａｄ）の出力が停止される。
【０１６６】
そして、ステップＳ２５に進み、いまデータの読み出しを行った周辺装置が、フローティングタイムを有するか否かが判定される。ステップＳ２５において、いまデータの読み出しを行った周辺装置が、フローティングタイムを有すると判定された場合、ステップＳ２６に進み、そのフローティングタイム分のウエイト時間がおかれ、ステップＳ２７に進む。
【０１６７】
ステップＳ２５において、いまデータの読み出しを行った周辺装置が、フローティングタ

イムを有さないと判定された場合、ステップＳ２６をスキップして、ステップＳ２７に進み、変数Ｎが１だけデクリメントされる。

【０１６８】

そして、ステップＳ２８に進み、変数Ｎが０に等しいか否かが判定される。ステップＳ２８において、変数Ｎが０に等しくないと判定された場合、ステップＳ２９に進み、周辺装置からデータをさらに読み出すための次のアドレスの出力処理が行われる。

【０１６９】

即ち、ステップＳ２９では、まず図１２に示すステップＳ４１において、図９のステップＳ７でバス幅テーブル記憶部１４から読み出したバス幅テーブル（図４）が参照され、いまデータの読み出しを行った周辺装置のアドレス形式が固定であるか否かが判定される。

【０１７０】

ステップＳ４１において、いまデータの読み出しを行った周辺装置のアドレス形式が固定であると判定された場合、即ちその周辺装置からデータをさらに読み出すのに、アドレスバッファ２１に記憶されたアドレスを増減させる必要がない場合、ステップＳ４２をスキップして、処理を終了する。

【０１７１】

また、ステップＳ４１において、いまデータの読み出しを行った周辺装置のアドレス形式が固定でなく、可変であると判定された場合、ステップＳ４２に進み、アドレスバッファ２１に記憶されたアドレスが１だけインクリメントされ、処理を終了する。

【０１７２】

以上のようにして、次のアドレスの出力処理が終了した後、即ち図１０のステップＳ２９の処理の終了した後、ステップＳ２１に戻り、ステップＳ２８において、変数Ｎが０に等しいと判定されるまで、ステップＳ２１乃至Ｓ２９の処理を繰り返す。

【０１７３】

即ち、ステップＳ２１乃至Ｓ２９において、周辺装置から、そのバス幅のデータが順次読み出され、そのデータが、データバッファ１１に、その下位ビットの方から順次ラッチされることにより、パックされる。そして、この処理が、

（ＣＰＵ１のアクセスタイプ）／（周辺装置のバス幅）回

だけ繰り返される。

【０１７４】

一方、ステップＳ２８において、変数Ｎが０に等しいと判定された場合、即ちステップＳ２１乃至Ｓ２９の処理により、データバッファ１１に、その下位ビットから順次ラッチされたデータのデータ長が、ＣＰＵ１のアクセスタイプに等しくなった場合、ステップＳ３０に進み、データバッファ１１にラッチされているデータがメインバス９を介してＣＰＵ１に供給され、処理を終了する。

【０１７５】

以上のようにして、周辺装置からのデータのリード処理が終了すると、即ち図９のステップＳ１３の処理が終了すると、ステップＳ１４に進む。

【０１７６】

一方、図９のステップＳ１１において、ステップ１でＣＰＵ１より出力された信号がライト信号であると判定された場合、ステップＳ１２に進み、周辺装置へのデータのライト処理が行われる。

【０１７７】

即ち、ステップＳ１２では、まずＣＰＵ１からメインバス９を介して出力されたデータが、データバッファ１１にラッチされる。そして、図１３に示すステップＳ５１において、ＣＰＵ１に対し、信号ＤａｔａＯｋが出力され、ステップＳ５２に進む。

【０１７８】

これにより、上述したようにＣＰＵ１は、周辺装置へのデータの書き込みが終了した（実際には、終了していないが）と認識する。

【０１７９】

ステップＳ５２において、上述した図１０のステップＳ２１（図１１のステップＳ３１乃至Ｓ３４）と同様のリカバリ処理が行われ、ステップＳ５３に進み、データバッファ１１にラッチされたＣＰＵ１からのデータが、いまそのデータの書き込みを行う周辺装置のバス幅（本実施例では、８ビットまたは１６ビット）にあわせて、その下位ビットの方からＭＵＸ１２ａ乃至１２ｄを介して読み出され、信号ＳＤａｔａとしてサブバス１０に出力されるとともに、ライト信号（ＳＷｒｉｔｅ）がサブバス１０に出力される。

【０１８０】

そして、ステップＳ５４に進み、図９のステップＳ６でタイミングテーブル記憶部１３から読み出されたタイミングテーブルに記述された、いまデータの書き込みを行おうとしている周辺装置のライトディレイタイムが参照され、その分だけのウエイト時間がおかれる。

【０１８１】

これにより、周辺装置のバス幅にあわせて、データバッファ１１から読み出されたデータが、その周辺装置へ書き込まれる。

【０１８２】

その後、ステップＳ５５において、ライト信号（ＳＷｒｉｔｅ）の出力が停止され、ステップＳ５６に進み、いまデータの書き込みを行った周辺装置が、ホールドタイムを有するか否かが判定される。ステップＳ５６において、いまデータの書き込みを行った周辺装置が、ホールドタイムを有すると判定された場合、ステップＳ５７に進み、そのホールドタイム分のウエイト時間がおかれ、ステップＳ５８に進む。

【０１８３】

一方、ステップＳ５６において、いまデータの書き込みを行った周辺装置が、ホールドタイムを有さないと判定された場合、ステップＳ５７をスキップして、ステップＳ５８に進み、変数Ｎが１だけデクリメントされる。

【０１８４】

そして、ステップＳ５９に進み、変数Ｎが０に等しいか否かが判定される。ステップＳ５９において、変数Ｎが０に等しくないと判定された場合、ステップＳ６０に進み、周辺装置にデータをさらに書き込むための次のアドレス、データの出力処理が行われる。

【０１８５】

即ち、ステップＳ６０では、まず図１４に示すステップＳ７１において、図９のステップＳ７でバス幅テーブル記憶部１４から読み出されたバス幅テーブル（図４）が参照され、いまデータの書き込みを行った周辺装置のバス幅が、８ビットであるか否かが判定される。

【０１８６】

ステップＳ７１において、いまデータの書き込みを行った周辺装置のバス幅が、８ビットでないと判定された場合、即ちそのバス幅が１６ビットである場合（本実施例においては、サブバス１０に接続された周辺装置のバス幅が８ビットまたは１６ビットのいずれかとしている）、ステップＳ７２に進み、データバッファ１１にラッチされた３２ビットのデータの上位１６ビットがＭＵＸ１２ａ，１２ｂに出力され、ステップＳ７４に進む。

【０１８７】

また、ステップＳ７１において、いまデータの書き込みを行った周辺装置のバス幅が、８ビットであると判定された場合、ステップＳ７３に進み、データバッファ１１にラッチされた３２ビットのデータのうちの、直前に出力された８ビットの次の上位８ビットがＭＵＸ１２ｃ（１２ｂまたは１２ａ）に出力され、ステップＳ７４に進む。

【０１８８】

ステップＳ７４においては、再びバス幅テーブル（図４）が参照され、いまデータの書き込みを行った周辺装置のアドレス形式が固定であるか否かが判定される。

【０１８９】

ステップＳ７４において、いまデータの書き込みを行った周辺装置のアドレス形式が固定であると判定された場合、即ちその周辺装置にデータをさらに書き込むのに、アドレスバ

ッファ２１に記憶されたアドレスを増減させる必要がない場合、ステップＳ７５をスキップして、処理を終了する。

【０１９０】

また、ステップＳ７４において、いまデータの書き込みを行った周辺装置のアドレス形式が固定でなく、可変であると判定された場合、ステップＳ７５に進み、アドレスバッファ２１に記憶されたアドレスが１だけインクリメントされ、処理を終了する。

【０１９１】

以上のようにして、次のアドレス、データの出力処理が終了した後、即ち図１３のステップＳ６０の処理の終了した後、ステップＳ５１に戻り、ステップＳ５９において、変数Ｎが０に等しいと判定されるまで、ステップＳ５１乃至Ｓ６０の処理を繰り返す。

【０１９２】

即ち、ステップＳ５１乃至Ｓ６０において、ＣＰＵ１からのデータを記憶しているデータバッファ１１から、周辺装置のバス幅にあわせて、その下位ビットの方からデータが順次読み出されることにより、データがアンパックされ、周辺装置に書き込まれる。そして、この処理が、

（ＣＰＵ１のアクセスタイプ）／（周辺装置のバス幅）回

だけ繰り返される。

【０１９３】

一方、ステップＳ５９において、変数Ｎが０に等しいと判定された場合、即ちステップＳ５１乃至Ｓ６０の処理により、データバッファ１１の下位ビットから、データが順次読み出され、そのデータが周辺装置にすべて書き込まれた場合、ステップＳ６１に進み、周辺装置へのデータ（ＳＤａｔａ）の出力が停止され、処理を終了する。

【０１９４】

以上のようにして、周辺装置へのデータのライト処理が終了すると、即ち図９のステップＳ１２の処理が終了すると、ステップＳ１４，Ｓ１５に順次進み、周辺装置へ出力されていたチップセレクト信号、アドレス（ＳＡｄｄｒｅｓｓ）の出力が停止される。

【０１９５】

そして、ステップＳ１に戻り、再びステップＳ１からの処理を繰り返す。

【０１９６】

以上のようにして、ゲートウェイ４では、タイミングの制御と、バス幅およびアドレス形式の制御と、並列して（同時に）行われるので、装置全体の処理速度を向上させることができる他、ハードウェアに対する依存性の少ないアプリケーションプログラムを提供することが可能となる。

【０１９７】

【発明の効果】

よって、本発明のバス制御装置によれば、装置全体を大型化、高コスト化することなく、中央演算処理装置と周辺装置との間のデータのやり取りを制御することができる。

【図面の簡単な説明】

【図１】本発明のバス制御装置を適用したコンピュータの一実施例の構成を示すブロック図である。

【図２】図１の実施例におけるゲートウェイ４のより詳細なブロック図である。

【図３】タイミングテーブルを示す図である。

【図４】バス幅テーブルを示す図である。

【図５】デバイス７からＣＰＵ１へデータの読み出しが行われるときの動作を説明するタイミングチャートである。

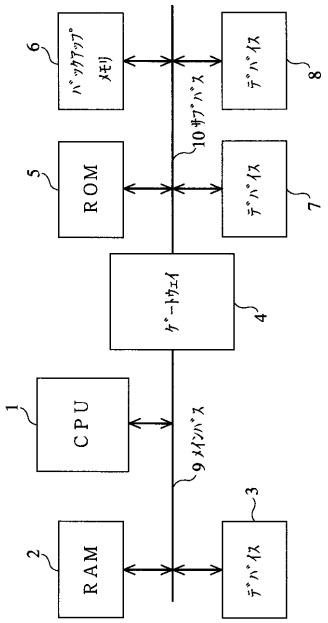
【図６】ＣＰＵ１からデバイス８へデータの書き込みが行われるときの動作を説明するタイミングチャートである。

【図７】ＲＯＭ５からＣＰＵ１へデータの読み出しが行われるときの動作を説明するタイミングチャートである。

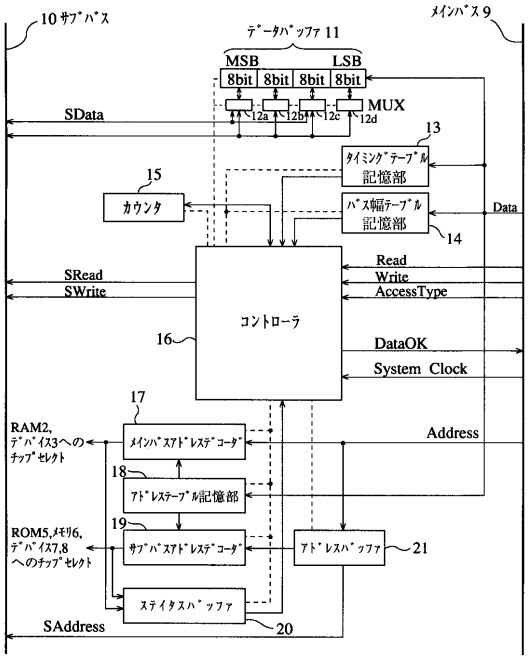
【図８】ＣＰＵ１からデバイス７へデータの書き込みが行われるときの動作を説明するタ

イミングチャートである。
【図９】ゲートウェイ４の動作を説明するフローチャートである。
【図１０】図９のフローチャートのステップＳ１３のデータのリード処理のより詳細を説明するフローチャートである。
【図１１】図１０のフローチャートのステップＳ２１のリカバリ処理のより詳細を説明するフローチャートである。
【図１２】図１０のフローチャートのステップＳ２９の次のアドレスの出力処理のより詳細を説明するフローチャートである。
【図１３】図９のフローチャートのステップＳ１２のデータのライト処理のより詳細を説明するフローチャートである。
【図１４】図１３のフローチャートのステップＳ６０の次のアドレス、データの出力処理のより詳細を説明するフローチャートである。
【符号の説明】
４　ゲートウェイ
１１　データバッファ
１２ａ乃至１２ｄ　多重器
１３　タイミングテーブル記憶部
１４　バス幅テーブル記憶部
１５　カウンタ
１６　コントローラ
１７　メインバスアドレスデコーダ
１８　アドレステーブル記憶部
１９　サブバスアドレスデコーダ
２０　ステイタスバッファ
２１　アドレスバッファ

【図１】



【図２】



ゲートウェイ 4

【図３】

（単位：ｸﾛｯｸ）

	ﾃﾞﾊﾞｲｽ7	ﾃﾞﾊﾞｲｽ8	ROM5	ﾒﾓﾘ6
ﾘｰﾄﾞﾃﾞｨﾚｲﾀｲﾑ	2	3	2	4
ﾗｲﾄﾃﾞｨﾚｲﾀｲﾑ	1	3	—	4
ﾎｰﾙﾄﾞﾀｲﾑ	1	1	0	0
ﾌﾛｰﾃｨﾝｸﾞﾀｲﾑ	2	0	0	0
ﾘｶﾊﾞﾘﾀｲﾑ	0	0	0	10

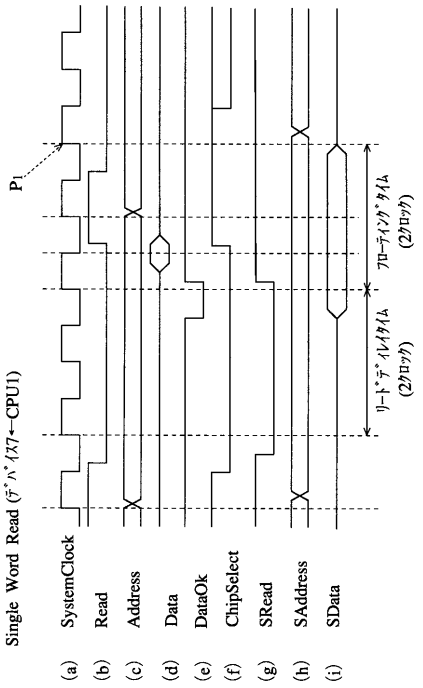
ﾀｲﾐﾝｸﾞﾃｰﾌﾞﾙ

【図４】

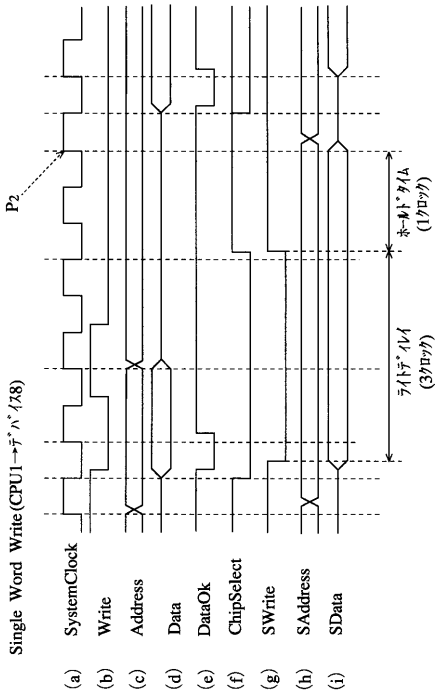
	ﾃﾞﾊﾞｲｽ7	ﾃﾞﾊﾞｲｽ8	ROM5	ﾒﾓﾘ6
ﾊﾞｽ幅	8bit	16bit	8bit	16bit
ｱﾄﾞﾚｽ形式	固定	固定	可変	可変

ﾊﾞｽ幅ﾃｰﾌﾞﾙ

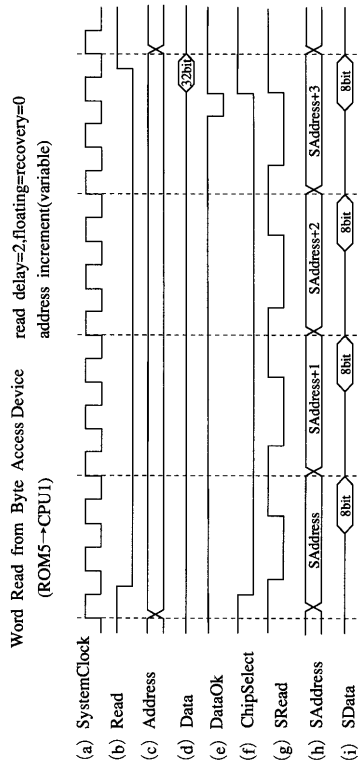
【図５】



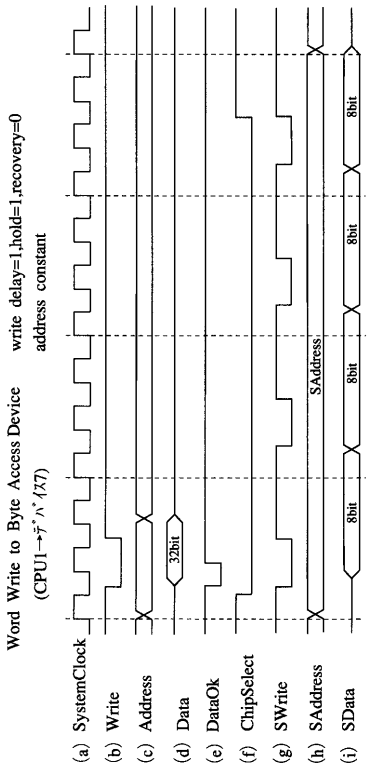
【図６】



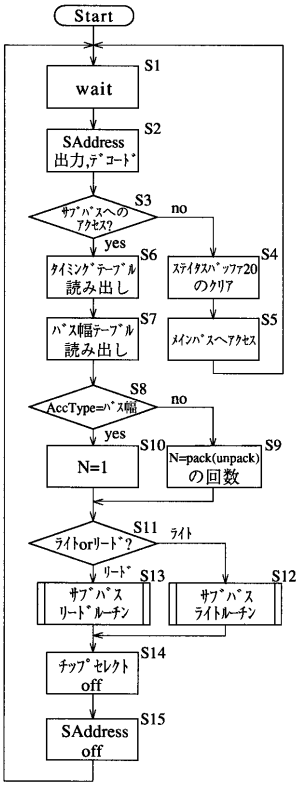
【図７】



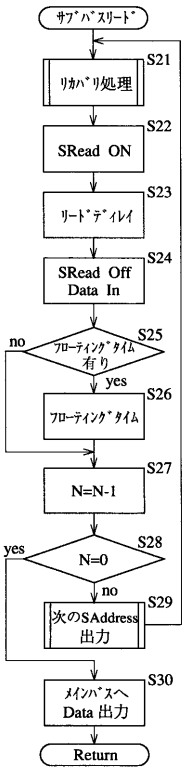
【図８】



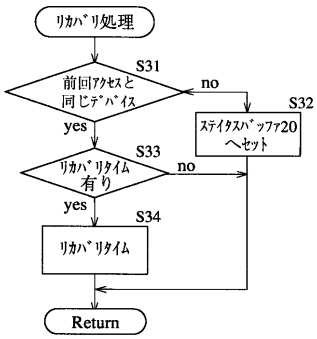
【図９】



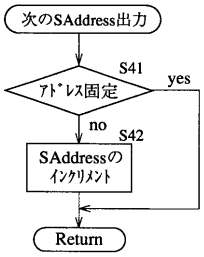
【図１０】



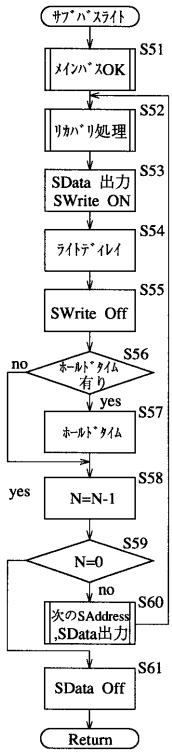
【図１１】



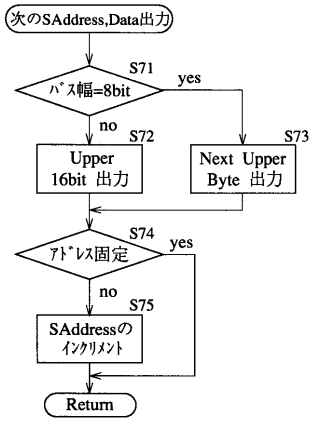
【図１２】



【図１３】


ｻﾌﾞﾊﾞｽﾗｲﾄ
S51
ﾒｲﾝﾊﾞｽOK
S52
ﾘｶﾊﾞﾘ処理
S53
SData 出力
SWrite ON
S54
ﾗｲﾄﾃﾞｨﾚｲ
S55
SWrite Off
S56
no
ﾎｰﾙﾄﾞﾀｲﾑ
有り
yes
S57
ﾎｰﾙﾄﾞﾀｲﾑ
S58
yes
N=N-1
S59
N=0
no
S60
次のSAddress
,SData出力
S61
SData Off
Return

【図１４】


次のSAddress,Data出力
S71
ﾊﾞｽ幅=8bit
yes
no
S72
Upper
16bit 出力
S73
Next Upper
Byte 出力
S74
ｱﾄﾞﾚｽ固定
yes
no
S75
SAddressの
ｲﾝｸﾘﾒﾝﾄ
Return

フロントページの続き

(56)参考文献 特開平０４－３１５２４４（ＪＰ，Ａ）
特開平０３－０８８０５９（ＪＰ，Ａ）
特開平０１－１９１９６６（ＪＰ，Ａ）
特開昭６４－０４４５６６（ＪＰ，Ａ）
特開昭６１－２９６４５６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 13/42 350
G06F 13/12 340
WPI(DIALOG)