

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4259751号  
(P4259751)

(45) 発行日 平成21年4月30日(2009.4.30)

(24) 登録日 平成21年2月20日(2009.2.20)

(51) Int.Cl.	F I
<b>G06F 13/36 (2006.01)</b>	G06F 13/36 510
<b>G06F 13/362 (2006.01)</b>	G06F 13/362 510G
<b>G06F 13/368 (2006.01)</b>	G06F 13/368 A
<b>G06F 13/42 (2006.01)</b>	G06F 13/42 310

請求項の数 19 (全 24 頁)

(21) 出願番号	特願2000-511115 (P2000-511115)	(73) 特許権者	500107083
(86) (22) 出願日	平成10年7月22日 (1998.7.22)		ソニックス・インコーポレーテッド
(65) 公表番号	特表2001-516100 (P2001-516100A)		アメリカ合衆国・94040・カリフォル
(43) 公表日	平成13年9月25日 (2001.9.25)		ニア州・マウンテン ビュー・ウエスト
(86) 国際出願番号	PCT/US1998/015006		エル カミノ リアル・2440・スイ
(87) 国際公開番号	W01999/013405		ト 620
(87) 国際公開日	平成11年3月18日 (1999.3.18)	(74) 代理人	100064621
審査請求日	平成17年7月4日 (2005.7.4)		弁理士 山川 政樹
(31) 優先権主張番号	08/924, 368	(72) 発明者	ウィンガード, ドリユー・エリック
(32) 優先日	平成9年9月5日 (1997.9.5)		アメリカ合衆国・94070・カリフォル
(33) 優先権主張国	米国 (US)		ニア州・サン カルロス・ヒューイット
			ドライブ・1059

最終頁に続く

(54) 【発明の名称】 リアルタイムで動的に帯域幅を割り当てる完全パイプライン式固定待ち時間通信システム

(57) 【特許請求の範囲】

【請求項1】

複数のバス・サイクルの間に動作する同期バスであって、前記複数のバス・サイクルは複数の繰り返しフレームに分割され、各フレームはさらに複数のパケット・バス・サイクルに分割され、それぞれのパケット・バス・サイクルは少なくとも1つのクロック・サイクルを含む同期バスと、

前記同期バスに結合された少なくとも2つの第1と第2のイニシエータ・サブシステムと、

前記同期バスに結合された少なくとも1つのターゲット・サブシステムとから構成され、

前記第1のイニシエータ・サブシステムは前記バスを使用するため事前に割り当てられた少なくとも1つのパケットを含むように構成されるとともに、事前に割り当てられたパケット・バス・サイクル内で、コマンドとターゲット・サブシステムのアドレスを有する第1のトランザクション要求をクロック・サイクル実行中に送出するように構成され、

前記第2のイニシエータ・サブシステムは、前記第1のトランザクションが完了する前に、事前に割り当てられた第2のパケット・バス・サイクル内で、第2のトランザクションの要求を第2のクロック・サイクル実行中に前記同期バスを介して送出するように構成され、

前記ターゲット・サブシステムは、前記第1の要求を受け取り、前記アドレスが前記ターゲット・バスシステムに対応しているかどうかを判断するように構成され、前記要求の

アドレスが前記ターゲット・サブシステムのアドレスに対応している場合、前記ターゲット・サブシステムは前記第2のクロック・サイクルの後に発生する第3のクロック・サイクルにおいて前記第1の要求に応答することを特徴とするコンピュータ・バス・システム

【請求項2】

前記少なくとも2つのイニシエータ・サブシステムのそれぞれは、そのイニシエータ・サブシステムに事前に割り当てられた少なくとも2つ以上のパケット・バス・サイクルを有することを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項3】

前記事前に割り当てられたどのパケット・バス・サイクルも最大1つのイニシエータ・サブシステムに割り当てられていることを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項4】

前記第1のイニシエータ・サブシステムは、前記少なくとも1つの事前に割り当てられたパケット・バス・サイクルの前のクロック・サイクルが実行されている間に、前記少なくとも1つの事前に割り当てられたパケット・バス・サイクルを使用する意向を信号で送信することを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項5】

パケット・バス・サイクルが前記第1のイニシエータ・サブシステムに事前に割り当てられているが、前記第1のイニシエータ・サブシステムによって使用されていない場合、またはパケット・バス・サイクルがいかなる特定のイニシエータ・サブシステムにも事前に割り当てられていない場合に、どのイニシエータ・サブシステムも公平なラウンド・ロビン・スキームに従い、そのパケット・バス・サイクルに対するアクセス権を調停することが許されていることを特徴とする請求項4に記載のコンピュータ・バス・システム。

【請求項6】

前記第1のイニシエータ・サブシステムは、構成データを格納するように構成されたアドレス可能なメモリを備え、前記構成データは前記パケット・バス・サイクルの事前割当てを制御し、前記バスを介して前記アドレス可能なメモリはアクセスされることを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項7】

前記イニシエータ・サブシステムへの前記パケット・バス・サイクルの事前割当ては、前記バスを介して通常のトランザクションに匹敵する速度で、通常のバス動作と並行してプログラム可能であることを特徴とする請求項1に記載のコンピュータ・バス・システム

【請求項8】

前記同期バスは一組のフラグ線を有し、各フラグ線は1つの送信側サブシステムに割り当てられ、前記フラグ線は前記送信側サブシステムと少なくとも1つの受信側サブシステム間の制御情報を送信するために用いられ、前記制御情報は1つのサブシステムのアドレス空間またはパケット・バス・サイクルの事前割当てには無関係であることを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項9】

前記一組のフラグ線の各フラグ線は1つのフラグ番号に関連づけられ、前記バスはフラグ番号を搬送する一組のフラグ番号線をさらに有し、そのフラグ番号は前記一組のフラグ線のうち少なくとも1つの関連づけられたフラグ線が情報監視され得ることを示すことを特徴とする請求項8に記載のコンピュータ・バス・システム。

【請求項10】

フラグ番号は、ターゲット・サブシステムが所定の待ち時間を満たすよう適時に要求に応答できない旨をイニシエータ・サブシステムに知らせるために用いられることを特徴とする請求項9に記載のコンピュータ・バス・システム。

【請求項11】

10

20

30

40

50

ターゲット・サブシステムが応答できない場合、前記ターゲット・サブシステムは前記フラグ線上に信号をアサートし、

前記第1のイニシエータ・サブシステムは前記要求を再発行することにより前記シグナルに応答し、前記ターゲット・サブシステムは前記再発行された要求に応答し、前記フラグ線上の信号のアサートを解除することを特徴とする請求項10に記載のコンピュータ・バス・システム。

【請求項12】

前記イニシエータ・サブシステムおよび前記ターゲット・サブシステムはそれぞれ、前記バスのバス・クロックに対し同期するように構成されたクライアント・クロックを有するクライアントと、

前記クライアントと前記バスの間に接続されたインターフェース・モジュールとを備え、前記クライアント・クロックは前記バス・クロックから導出され、前記バス・クロックの比率はプログラム可能であることを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項13】

前記イニシエータ・サブシステムおよび前記ターゲット・サブシステムはそれぞれ、クライアント・クロックを有するクライアントと、前記クライアントと、前記バス・クロックに前記クライアント・クロックを同期させる

同期論理を有する前記バスとの間に接続されたインターフェース・モジュールとを備え、前記クライアント・クロックのクライアント周波数は前記バスのバス・クロックの周波数の動作と無関係であることを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項14】

前記ターゲット・サブシステムは、前記第1の要求に対し、前記第1のクロック・サイクルの後に所定回数のクロック・サイクルだけ応答することを特徴とする請求項1に記載のコンピュータ・バス・システム。

【請求項15】

前記所定回数のクロック・サイクルはプログラム可能であることを特徴とする請求項14に記載のコンピュータ・バス・システム。

【請求項16】

前記所定回数のクロック・サイクルは前記バスを介した通常のトランザクションに匹敵する速度で、通常のバス動作と並行してプログラム可能であることを特徴とする請求項15に記載のコンピュータ・バス・システム。

【請求項17】

サブシステム間でバスを介してデータを送信する方法であって、第1のイニシエータから前記バス上への第1のクロック・サイクルにおける第1のトランザクションからの要求を送信するステップと、

第2のイニシエータから前記バス上への第2のクロック・サイクルにおける第2のトランザクションからの要求を送信するステップと、

少なくとも1つのターゲットが前記バスから前記第1の要求を受信するステップと、前記少なくとも1つのターゲットが前記ターゲットアドレスに対応するアドレスを有している場合、前記ターゲットにより第3のクロック・サイクルにおいて第1の要求に応答するステップと、

前記要求に記述されたコマンドを実行し、前記バス上に前記第1の要求に対する応答を送信するステップと

から構成され、

前記第1の要求は少なくとも1つのコマンドとターゲット・アドレスを有し、前記第1のクロック・サイクルは前記第1のイニシエータに事前に割り当てられた第1の packets ・バス・サイクルの一部であり、各 packets ・バス・サイクルは少なくとも1つのクロック・サイクルを有し、複数の packets ・バス・サイクルは前記バス上に繰り返しフレーム

10

20

30

40

50

を形成し、

前記第2の要求も少なくとも1つのコマンドとターゲット・アドレスを有し、前記第2のクロック・サイクルは前記第2のイニシエータに事前に割り当てられた第2のポケット・バス・サイクルの一部であり、前記第2のクロック・サイクルは前記第1のクロック・サイクル完了後および前記第1のトランザクションが完了する前に発生し、

前記第3のクロック・サイクルは前記第2のクロック・サイクルの後で発生することを特徴とする方法。

【請求項18】

ポケット・バス・サイクルは前記第1のイニシエータに事前に割り当てられるが前記第1のイニシエータに使用されない場合、第2のレベル調停を実行するステップをさらに有し、そのステップにおいて前記ポケット・バス・サイクルは別のイニシエータに割り当てられることを特徴とする請求項17に記載の方法。

10

【請求項19】

各サブシステムがイニシエータとターゲットのうち少なくとも1つを有する少なくとも2つのサブシステムを結合するコンピュータ・バス・システムにおける、パラメータを変更するための方法であって、この方法は、

各サブシステム内で、第1のアドレス可能なメモリを決めるステップと、

各サブシステム内で、前記サブシステムによって使用される構成データを含んでいる第2のアドレス可能なメモリを決めるステップと、

少なくとも1つのポケット・バス・サイクルをイニシエータに事前に割り当て、アドレス指定された前記バスを介して、第1のポケット・バス・サイクルのクロック・サイクルで前記サブシステムの前記第1のメモリに書き込むためのコマンドを発行するステップと

20

、  
全てのサブシステムに前記書き込みコマンドを発行するように前記イニシエータを構成するステップと、

前記書き込みコマンドに回答して、前記第1のメモリの内容に基づいて、新しい構成データをそれぞれのサブシステムの第2メモリに書き込むステップと、

各サブシステムが再構成のために前記新しい構成データを使用するステップと

から構成され、

前記バス上の各ポケット・バス・サイクルは少なくとも1つのクロック・サイクルを有し、複数のポケット・バス・サイクルは前記バス上に繰り返しフレームを形成し、

30

前記新しい構成データは、前記バスを介してトランザクションが発生するためのプログラム可能なバスの待ち時間の長さ、前記イニシエータに事前割り当てられたポケット・バス・サイクルの変更可能な帯域幅の範囲を決定することを特徴とする方法。

【発明の詳細な説明】

【0001】

(発明の背景)

(1. 発明の分野)

本発明は、コンピューティング・デバイス同士を接続するためのバス・システムに関する。

40

【0002】

(2. 背景技術)

電子コンピューティング/通信システムでは、特性が多様化して複雑さが増す一方で、1つの機能あたりの物理的なサイズとコストが削減され続けているため、システム設計者が受ける制約は次第に多くなっている。市場に並ぶまでの期間が短いことがビジネス上不可欠な要素の1つである一般消費者向け電子機器市場などの大規模でコストの影響を受けやすい市場では、このようなシステムに対する需要量が注目されている。より複雑なシステムを一層短期間かつ低価格で持ち込むことができる唯一の手法は、最新の集積回路技術を有効利用すること、ソフトウェアに機能性を持たせること、サブシステム・コンポーネントを1つのデザインから後継のデザインに移行することである。

50

## 【 0 0 0 3 】

サブシステム・コンポーネントを後継デザインで再利用することは、極めて一般的になっており、時間の節約以外にも多くの利益が得られている。第1に、新しいシステムデザインの要件と性能を解析する際に、正確な結果を出すことのできる実績のあるサブシステムのモデルを書くことができる。実績のない新しいサブシステムのモデルは、実績のあるサブシステムほどの正確ではなく、また適切にデザインに影響を与えられない。第2に、システム設計者がさらに高いレベルの抽象性に注目しつつ最終的に得られるシステム実装の予測可能性を高めることで、実績のあるサブシステムが全体の設計プロセスを簡略化するビルディング・ブロックとして利用可能になる。第3に、ハードウェアサブシステムの再利用によって、ソフトウェアへの投下資本を保護してこれらのサブシステムを管理し、ハードウェア的なビルディング・ブロックの選択後すぐにシステム・ソフトウェアの実装を行うことができる。最後に、サブシステムの再利用によって、検証および試験に関する投下資本が保護される。所望のシステムが高度に集積化されているため、必要とされるサブシステムは最終的には集積回路内に埋め込まれることになる。埋め込まれたデザインでは、デザインの機能性の確認に多くの手間と労力が必要になり、個々のシステムを試験して正しく構築されていることを証明しようとする、費用負担を伴う遅滞が生じるかコストをかけてシステムを手直ししなければならない場合がある。したがって、デザインを再利用しても、最大の成果はサブシステムの検証および試験の整合性を維持できること以外にないことが多い。

10

## 【 0 0 0 4 】

再利用に対する従来の手法には様々な長所と短所がある。このような手法の重要な特徴に、様々なサブシステムが互いに提供しあう通信インターフェースがある。1つの手法として、サブシステムが通信を行わなければならない各ピアとそのサブシステムとの間に、カスタマイズされたポイントツーポイントインターフェースを定義することがあげられる。このインターフェース・スタイルは、第1のサブシステムから設計対象となるサブシステムにデータが送られ、これによってデータが処理され、データが第2のサブシステムに出力されるアプリケーションで特に普及している。このカスタマイズされた手法では、プロトコルが簡略化され、保証性能が得られ、関係のないサブシステムへの依存性が排除される。しかしながら、カスタマイズされたインターフェースは当然ながら柔軟性のないものとなる。新しいアプリケーションでインターフェースを共用していないサブシステムと既存のサブシステムとをインターフェースしなければならない場合、デザインの手直しが必要になる。同じアプリケーションでこのサブシステムを別のサブシステムと通信させなければならない場合、サブシステムまたはインターフェースのコピーが複数必要になり、システムの非効率化を招くことがある。さらに、アップデートがなされたアルゴリズムをこのサブシステムに実装しようとする、カスタマイズされた古いインターフェースに新しいアルゴリズムを強引に組み込んで既存のピア・サブシステムと通信を行えるようにしなければならない。

20

30

## 【 0 0 0 5 】

2つ目の手法として、標準化されたインターフェースを利用してシステムを決めることがあげられる。標準化されたインターフェースを様々な形態の共用相互接続とペアにして通信サブシステムを構成したものはよく見られる。標準化されたインターフェースの多くはすでに確立されているコンピュータ・バス・プロトコルに準拠している。この手法の大きな利点の1つが、各エージェントがシステム内の他のすべてのエージェントと通信できるという通信パターンの柔軟さである。コンピュータ・バスを使用すると、バスが十分な性能を維持できる範囲で、異なるエージェントをシステムに必要な数だけいくつでも一緒に接続できるため、請求システム設計を行うことができる。決定的な利点は、ソフトウェアに実装された複雑なアルゴリズムでハードウェアのサブシステムと通信を行う必要のあるシステムに、プロセッサ・メモリ通信の基本であるアドレス/データ転送を自然な形でマッピングできることである。共用相互接続の確立に伴って、様々なイニシエータ装置およびターゲット・サブシステムの間で通信資源を割り当てる必要が生じる。コンピュータ・バスの

40

50

場合、資源の割当ては一般に調停と呼ばれる。

【 0 0 0 6 】

標準化されたバス・インタフェースの主な欠点は、調停が原因で発生する転送遅延（通常、待ち時間と呼ばれる）がどの程度になるか分からないことである。待ち時間が分からないことで、リアルタイム性に関する制約を満たさなければならないサブシステムでは問題が生じる。期限内に到着しなかったデータが原因でシステムの動作がおかしくなる場合があるからである。2つ目の欠点は、コンピュータ・システムの継承が原因で利用可能な転送機会（通常、帯域幅と呼ばれる）が不足することである。過去においてのみならず現在にあって、バスは一般に自己に接続されたプロセッサのピーク転送速度をサポートするように設計されているため、プロセッサからの要求のない転送機会は他のエージェントが利用できるようにされる。したがって、帯域幅の合計がプロセッサのピーク帯域幅を上回るシステムでは、マルチレベル・バス・スキームを用いるか、バスと専用接続とを混在させてその帯域幅を分割しなければならない。最後の欠点は、アドレス/データ転送へのマッピングが容易ではない通信（例えば、2つのエージェント間でのハンドシェイク信号など）またはイニシエータではないエージェントから開始された通信（例えば、データが利用できることをイニシエータ装置に通知するようターゲット・サブシステムによってドライブされる割り込み信号）を効果的にサポートできないことである。

10

【 0 0 0 7 】

要するに、既存の通信手法は、デザインの効果的な再利用のための要件を満たしていない。必要なのは、性能特性が極めて多様であるシステムにおいてサブシステムを最大限に利用できる新しい構造である。コンピュータ・バスを拡張して性能面および通信スタイル面の制約を排除することができれば、それがすなわち高度に集積化された多くのシステムの基盤となり得る。

20

【 0 0 0 8 】

さらに、サブシステムの通信要件にはかなりの違いがある。キーボードや赤外線リモート・コントローラ、LEDディスプレイのような入出力装置などのいくつかのサブシステムは、帯域幅要件が非常に低いため待ち時間の变化に極めて強い。RISC CPUまたはデジタル信号プロセッサなどの他のサブシステムでは、利用可能な帯域幅が高くメモリへの待ち時間が低いが、速度の遅いプログラムの実行を犠牲にして帯域幅に関する制約と可変の待ち時間を許容できることが要求される。T1電話回線リンクまたはSVGADディスプレイなどのさらに他のサブシステムでは、高帯域幅要件に対応できるが、帯域幅と待ち時間における不明確さを許容できない。このようなリアルタイム・サブシステムは、帯域幅および待ち時間が保証されている必要があり、付加的な性能を与えることに何ら利点はない。

30

【 0 0 0 9 】

したがって、かかる通信機構は、性能特性が様々に異なる複数のサブシステムの相互作用を可能にし、標準化されたインタフェースを提供することでサブシステムの再利用性を大幅に高めるものであることが望ましい。しかしながら、これは通信機構の動作周波数を可変にして設計中のシステムの所望の帯域幅をサポートしなければならないという問題に対処するものではない。標準化されたインタフェースが通信インタフェースである場合、通信インタフェースの周波数を変更するためにはサブシステムの動作周波数も変更する必要がある。サブシステムの周波数をシステム要件より増やすと電力が浪費され、アルゴリズム上の問題が発生する可能性があり（サブシステムの中には、定周波数で動作させなければならないものがある）、新しく設定する周波数がそれまで設計されていたサブシステムの動作可能周波数よりも高くなると、デザインを全く再利用できない可能性もある。必要なのは、通信インタフェースの周波数と様々なクライアント・サブシステムの動作周波数とを切り離し、各クライアント・サブシステムが自己の要件に沿って動作できるようにする方法である。

40

【 0 0 1 0 】

（発明の概要）

50

本発明は、高い度合いでのデザインの再利用、性能分析の単純化、フレキシブルな通信能力、設計時間の短縮を可能にする予測可能性の高い通信能力を提供することによって、極めて複雑なシステムの設計を合理化する。

【0011】

一実施態様において、シングル・チップ・コンピュータ・バス・システムは、少なくとも1つのイニシエータ装置と、少なくとも1つのターゲット・サブシステムとを含む。バス・サイクルは、複数の繰り返しフレームに合わせて分割される。少なくとも1つのイニシエータ装置に、要求を送信するための特定の要求フレームと、要求フレームより所定の数のクロック・サイクルだけ後で送信する特定の応答フレームとが割り当てられている。要求を発行する際、イニシエータ装置は所定のフレームの間に要求を発行する。この要求には、ターゲット・サブシステムを識別するアドレスと、ターゲットがとるべきアクションを示すコマンドとが含まれている。バスに接続された各ターゲット・サブシステムは、発行される要求パケットを監視し、ターゲット・サブシステムのアドレスに対応するアドレスを含むこれらの要求パケットに対して応答する。ターゲットは、パケットに記述されたコマンドを実行し、要求パケットの発行から所定の数のクロック・サイクルだけ後に応答パケットを送信することによって応答を行う。

10

【0012】

好ましくは、多様な制約を有するサブシステム間の通信を一様化するシステムに上記のものを実装しておく。例えば、一実施態様において、高帯域幅と待ち時間が低いことを必要とするイニシエータにサイクルを事前に割り当てる。残りのイニシエータでは、例えばラウンドロビン形式などで利用可能なフレームを求めて調停がなされる。

20

【0013】

本発明の目的、特徴および利点は、以下の詳細な説明から明らかになる。

【0014】

( 詳細な説明 )

本発明によれば、タイミング制約の異なる様々なサブシステムをサポートする革新的な通信プロトコルおよびバス構造が単一の半導体装置上に得られる。特に、後述するように、本発明のシステムおよび方法によれば、2つ以上のサブシステム間で高帯域幅かつ待ち時間の低い通信路を提供するための効率的な機構が得られると共に、リアルタイム性能を必要とするサブシステムに最小限の保証帯域幅を与える効率的な機構が得られる。さらに、本発明のシステムおよび方法によれば、サブシステムの要件とシステム・レベルの要件とを切り離すための効率的な機構が得られる。これによって、システムの待ち時間と帯域幅の要件が様々な異なる被接続サブシステムを、それらのサブシステムに手を加えることなく再利用することができる。また、本発明のシステムおよび方法によれば、アドレス/データスタイルのメモリ・マッピング通信、専用ポイントツーポイント通信および一対多のブロードキャストスタイルの通信を実現するための効率的な機構が得られる。ここで提案する通信方法は、さらに大きなシステムの一部であり、かつ共用の通信バスを介して互いに通信を行う異なるサブシステム間での通信を管理する。このシステムは、1つ以上の集積回路またはチップからなるものであることができ、サブシステムは多くの論理ブロックを含む集積回路上または単一の論理機能を実装した集積回路上の論理ブロックであることができる。

30

40

【0015】

図1は、本発明の教示内容に従って動作するシステムの一実施形態を示している。このシステムは、通信バス35を介して互いに通信を行う、3つのターゲット・サブシステム10、15、20と、2つのイニシエータ・サブシステム25、30とに接続されている。本願明細書では、ターゲット・サブシステム10、15、20を要求の受信と要求に対する応答のみしか行うことのできないサブシステムであると定義する。代表的なターゲット・サブシステムには、メモリサブシステムおよび入出力(I/O)装置がある。イニシエータ・サブシステム25、30は、要求の受信と要求に対する応答の他、要求を発行することもできるサブシステムである。代表的なイニシエータ・サブシステムには、中央処理

50

装置（CPU）コア、デジタル信号プロセッサ（DSP）コア、ダイレクト・メモリ・アクセス（DMA）エンジン、コプロセッサなどがある。一実施形態では、各サブシステム10、15、20、25、30とバス35との間にインタフェース・ユニット40、45、50、55、60が接続されている。これらのインタフェース・モジュール40、45、50、55、60は、バス・プロトコルに従ってパケットを発行および/またはパケットに回答する論理を含む。また、後述するように、効率的なデータ転送を行うためには、インタフェースがサブシステムおよびバス・クロックと同期していることが好ましい。最後に、本発明では、インタフェース・ユニット40、45、50、55、60に機能が分散していることが好ましいため、通信バス上のトランザクションを管理するための専用のコントローラユニットは不要である。さらに、インタフェース・ユニットをサブシステムの論理とマージしたり、あるいはバスに接続された別のブロック（40、45、50、55、60）として実装することも可能である。

10

#### 【0016】

一実施形態において、図1の通信バス35は、以下の線すなわち、コマンド/アドレス・ライン65、アサート・ライン70、クロック・ライン75、データ/応答ライン80およびフラグ・ライン85から構成される。このシステムにはまた、後述するような第2レベルの調停を実現するためのライン90、95などのトークン・リング・ラインも含まれている。

#### 【0017】

アドレス/コマンド・ライン65は、ターゲット・サブシステムを識別するアドレスの送信に用いられる。好ましくは、アドレス・ラインはさらにターゲット内の資源を識別する。好ましくは、特定のターゲット・サブシステムのアドレスと、サブシステム内の資源との間の特定の割当て関係を変え、ターゲットごとに異なるものとすることができる。さらに、この割当てがプログラム可能なものであると好ましい。

20

#### 【0018】

アサート・ライン70は、要求を送信するのに自己に事前に割り当てられたパケットを使おうとしていることを示すためにイニシエータ・サブシステムが使用する単一のラインであると好ましい。アサート・ラインがアクティブでなければ、トークンを保持しているイニシエータがそのパケットに対するアクセス権を得る。クロック・ライン75にはクロック信号が供給されている。このクロック信号によって複数のサブシステムがバス上で動作を同期させる。データ/応答ライン80は、データの送信と、要求に対する指定のターゲット応答の転送とに用いられる。フラグ・ライン85は、特殊な通信で利用可能な汎用ラインであるため、このラインによってバス・システムはさらにフレキシブルかつ丈夫なものとなっている。これは、アドレス/データ転送へのマッピングが容易ではない通信をサポートする上で特に有利である。例えば、後述するように、これらの汎用ラインを使用して、2つのサブシステム間でのハンドシェイク信号や、データが利用できることをイニシエータ装置に通知するようターゲット・サブシステムによってドライブされる割込み信号などのイニシエータではないサブシステムから開始された通信をサポートすることができる。

30

#### 【0019】

図2aおよび図2bは、イニシエータ・インタフェース・モジュールの一実施形態への接続の簡略ブロック図を示している。このモジュールは、要求の発行（論理250など）、クライアントとのインタフェース（クロック/シンクロナイザ245）、構成（構成レジスタ240）の維持、受信したコマンドに従っての構成の更新（論理250など）をはじめとする、本願明細書に記載の機能を実行するのに必要な論理とメモリとを含む。モジュールの動作に関する構成を更新するためのコマンドを受信するために、イニシエータは、ターゲットとしても機能し、そのためアドレス/デコード論理242を含む。説明を簡単にするために、イニシエータについては、埋め込み型のターゲットを有するものとして説明する（その主な機能がイニシエータの機能であるため）が、本願明細書にて説明するように、ターゲットの機能性も維持されている。しかしながら、サブシステムがその時々で

40

50



ターゲットまたはイニシエータとして機能し得ることまで考慮すれば柔軟性は維持される。図2bの表に信号をまとめておく。特に、「サイクル」は信号がアクティブであるバス・クロック(s b c l o c k)の位相を示している。バスはパイプライン化されているため、複数のトランザクションを同時進行させることができる。しかしながら、すべてのトランザクションで待ち時間が等しくなることが好ましいため、各トランザクションは同一のテンプレートに厳密に沿っている。サイクルS0を本願明細書では調停サイクルと呼ぶ。サイクルS1を本願明細書ではコマンド/アドレス・サイクルと呼ぶ。S1から所定の数のサイクル後に生じるSMは、データ/応答サイクルである。サイクルSNはエラー・サイクルである。

#### 【0020】

図2aを参照すると、クロック信号ラインs b c l o c k 2 0 5がバス・クロックを受信する。クライアントがバスと同期して動作するか何らかの誘導周波数で動作する場合、クロック回路242がインタフェースをクライアントまでドライブするのに必要なクロック分割を行う。トークン・リングにて実現される第2レベルの調停では、信号s b T o k e n I n (ラウンドロビン・トークン・grant・in)およびs b T o k e n O u t (ラウンドロビン・トークン・grant・out) 2 0 7、2 0 9を用いて、それぞれ入力トークンおよび出力トークンを示す。s b A s s e r t 2 1 1は、事前に割り当てられたパケットの所有者によって、パケット使用の意志を示すためにアサートされる。信号s b C m d 2 1 3は、図2cに示すようなトランザクションのタイプを識別する信号である。信号s b A d d r 2 1 5はターゲットのアドレスを搬送する。アイドル・コマンドおよびブロードキャスト・コマンド以外のコマンドはいずれもこのアドレス機構を利用している。s b W i d t h 2 1 7信号はデータの幅を示し、s b D a t a 2 1 9はデータを搬送し、s b R e s p 2 2 1は図2dに示すような応答情報を搬送する。s b F l a g N u m 2 2 3は、後述するように、監視対象となる信号のレシーバについてs b F l a g 2 2 5の特定のフラグを識別する。信号s b F l a g 2 2 5は、実装に応じて様々な方法で利用できる構成可能なフラグ・ラインである。

#### 【0021】

ターゲット・クライアントとの間をインタフェースするインタフェース・モジュールはさらに、ターゲット・クライアントとの間をインタフェースするためのターゲット・サブシステム信号線230を有する。この信号については図2eで説明する。これはトランザクション・タイプを含むが、その一例を図2fに示す。また、イニシエータ・クライアントとの間をインタフェースするインタフェース・モジュールはさらに、図2hに示すようなイニシエータ・クライアント・トランザクション・タイプを含む、図2gに示すようなイニシエータ・サブシステム信号線235を有する。

#### 【0022】

通信は、フレーム動作、時分割多重アクセス、同期動作、パイプライン動作、固定待ち時間動作およびスプリットトランザクションを組み合わせることで達成される。時分割多重アクセス(TDMA)システムでは、バス・サイクルが複数のフレームに分割される。所定のソースと宛先装置との間での装置による転送は、事前に与えられたフレームである。同期システムは、同一のクロックに沿って装置が転送を行う装置の1つである。本実施形態では、パイプライン式のバスは、トランザクションに関連したアドレス・フィールドとコマンド・フィールドとが、データよりも前にクロック・サイクルでバスを介して送信され、最初のトランザクションのデータ/応答と後から発生したトランザクションのアドレス/コマンド部分との重畳が可能な状態で応答が送信されるバスである。完全パイプライン式のバスでは、クロックの各サイクルで新しいトランザクションを開始することができる。したがって、前のトランザクションが完了する前に別のトランザクションを開始することができる。さらに、本発明では、トランザクションの開始とトランザクションの完了との間の待ち時間が固定されているため、性能面で一定の利益が得られることが分かっている。このため、完全パイプライン式の固定待ち時間スプリット・トランザクション・バスでは、コマンドと関連のデータ転送との間の待ち時間は、既知の保証値となる。この特

10

20

30

40

50

性は、固定待ち時間の完全パイプライン式動作を行うプロセッサまたはデジタル信号処理コアと通信を行うバスでは極めて重要である。

【 0 0 2 3 】

T D M Aである従来技術の通信システムでは、フレームは一般に、ポイントツーポイントボイス通信などの事前に指定されたポイントツーポイント通信に対して割り当てられる。ポイントツーポイント通信の変化は、一般にポイントツーポイント通信リンクのいずれかの終端での変化が原因で、ゆっくりと起こる。本発明は、転送 / 動作コマンドを開始する権利のみを割り当てることによって、これらの欠点を解決する。これによって、イニシエータ・サブシステムはリアルタイムで複数のターゲット・サブシステムと容易に通信を行うことができる。イニシエータ・サブシステムは、ターゲット・サブシステムのアドレスを識別するアドレス情報を含む要求を発行する。このため、イニシエータは、使用するコマンドとアドレスとに基づいて、1フレームの間に1つまたはそれ以上のターゲット・サブシステムと通信を行うことができる。

10

【 0 0 2 4 】

本発明の通信システムによれば、サブシステム間に共用相互接続を介して極めて高性能で待ち時間の低い通信路が得られる。必要なシステム帯域幅をサポートするために、通信バスを個々のサブシステムの動作周波数よりも高い動作周波数で動作させなければならないことがある。バス周波数とサブシステム周波数との間に差があることで、各サブシステムが要求する通信のパターン次第では、サブシステムの設計が対応の手間と労力を要するものとなり得る。特に、シングルサブシステムでは、自己に入出力される長いシーケンスのトラフィックの間に自システムよりも高速な通信バスに追いつかなくなることが多く、問題はサブシステムが自己のピーク帯域幅よりも高速にデータを処理しなければならないとは限らない点である。

20

【 0 0 2 5 】

本発明のシステムおよび方法ではさらに、バースト転送の利用を最小限に抑えることを試みている。バーストは連続したバス・サイクルで起こる一連のデータ転送であり、このとき一般に暗示アドレスがインクリメントされる。従来のコンピュータ・バスでは、このようなバースト・トラフィックを、バスの調停とアドレス転送が原因で生じる時間的な非効率を減らすことによって性能を高めるための方法の1つとして重要視している。送信側サブシステムおよび受信側サブシステムよりも高い周波数で動作する通信バスでバースト転送を重要視することによるシステムコストへの影響は、送信元と宛先で高価な格納場所を確保することによる影響である。送信元は、バースト長分のデータがバッファされるまで転送を開始することができず、宛先では転送されてくるデータを格納するためのバッファを用意しなければならない。さらに、最初のデータの送信前に送信元で作成される最後のデータを待つまさにそのプロセスによって、転送全体の待ち時間が増えることになる。

30

【 0 0 2 6 】

したがって、送信元または宛先での自然な動作速度でデータを転送できるようにする方法が、より良い解決法であることは理解できよう。このため、個々の転送の時間インターリーブされたセットが一連の連続したバースト転送にとって代わる。この手法を用いると、格納量が減り、転送の両終端部における効率が増す。また、適宜構成した時間インターリーブ・スキームを用いることで、実時間要求の厳しいことが特徴である要求側のマルチメディアおよび通信トラフィックと従来のコンピューティング・トラフィックとをシームレスに統合することができる。これらの特徴すべてを持たせるには、通信バスの基本的な調停スキームを改善する必要がある。

40

【 0 0 2 7 】

本発明のシステムの一実施形態では、2レベルの調停を使用する。これについて図3 aおよび図3 bを参照して概説する。図3 aを参照すると、帯域幅割当てが実行される。割当ては、システムの初期化時に実行可能である。さらに、本願明細書において説明するように、利用可能なライト・バス・トランザクションを用いてサブシステムの構成レジスタを更新することで、動作中に動的に再割当てを行うことができる。帯域幅割当てを行うため

50

に、ステップ302では、パケットを事前に異なるイニシエータ・サブシステムに割り当てられる。バス・システムに接続されたイニシエータ・サブシステムおよびイニシエータ・サブシステムでの需要に応じて、様々な数のパケットを異なるイニシエータ・サブシステムに割り当ててもよい。例えば、1つのイニシエータ・サブシステムの帯域幅要件を高くし、このサブシステムに利用可能なパケットを高い比率で割り当てることができる。別のイニシエータ・サブシステムには利用可能なパケットを低い比率で事前に割り当て、さらに別のイニシエータ・サブシステムにはどのパケットも割り当てないようにすることができる。さらに、ラウンドロビン調停用に利用できるパケットを常に残しておけるように、いくつかのパケットには何ら事前割当てを行わないようにしてもよい。

**【0028】**

ステップ304では、トークン・リング・ネットワークを確立し、例えばラウンドロビン調停などの第2レベルの調停を実行する。一般に、トークン・リングに対するメンバは事前に割り当てられたパケット以外のイニシエータ・サブシステムでよいが、事前に割り当てられたパケットであるイニシエータ・サブシステムをトークン・リング・ネットワークのメンバにしてもよい。

**【0029】**

2レベルの調停がなされると、バス・システムは動作を開始することができる。図3bを参照すると、ステップ306では、送信されてくる特定のパケットを事前に割り当てられたイニシエータがそのパケットを使う予定であるかどうかを判断する。これは例えば本実施形態のアサート・ラインなどの特定のバス・ラインを監視することによって判断できる。アサートされると、パケットを事前に割り当てられたイニシエータはトランザクションを実行するためのアクセス権を持ち（ステップ308）、要求が発行される（ステップ310）。イニシエータがアクセス権を必要としない場合、このパケットは、トークン・リング・ネットワークを介してなされる第2レベルの調停で利用可能である。好ましくは、図3bに示すように、事前に割り当てられたイニシエータがアクセス権を解放した直後にトークンを持っているイニシエータが要求を発行するように、第2レベルの調停（ステップ312、314）と第1レベルの調停（すなわちステップ306、308）とを同時に行う。しかしながら、調停プロセスを順次行うようにしてもよい。

**【0030】**

再度図3bを参照すると、ステップ312では、トークンの現在の所有者がパケットを必要としている場合、トークンを取り込んだイニシエータがこのトークンを持ち続け、事前に割り当てられたイニシエータでのアクセス権の解放後にアクセス権を与えられる（ステップ314）。それ以外の場合、次のイニシエータがアクセス権を取得することができるように、トークンはリング内の次のイニシエータ・サブシステムに向けてパスされる。アクセス権を要求しているイニシエータにトークンが到達するまでこのプロセスが続けられる。本実施形態では、トークンの宛先がトークンの最初の保持者であり、トークンがイニシエータ・サブシステムからアクセス権を要求されずにリングを一周することが分かっている場合、最初の所有者がトークンを持ち続け、ステップ314でアクセス権を与えられると、調停を完了できるようにするためのダミー要求を発行する。

**【0031】**

イニシエータ・サブシステムが発行する要求には、意図されたターゲットのアドレスが含まれている。好ましくは、この要求にはさらに、ターゲットにおける特定の資源のアドレスも含まれる。ターゲット・サブシステムは発行される各要求を監視し、このアドレスがターゲットのアドレスに対応しているかどうかを判断する（ステップ322）。対応している（例えば、一致）場合、ターゲット・サブシステムは、所定の数のクロック・サイクル後にこの要求に応答する（ステップ324）。例えば、要求がライト・オペレーションである場合、ターゲット・サブシステムはデータを受信し、要求の発行から所定の数のクロック・サイクル後に完了応答を発行する。この要求がリード・オペレーションである場合、ターゲット・サブシステムは、要求の発行から所定の数のクロック・サイクル後にリード・データを提供する。一実施形態では、発行されるアドレスとターゲット・アドレス

10

20

30

40

50

との間には1:1の対応関係が考慮されている。しかしながら、1つまたはそれ以上の装置にマッチするマスクに関してなされる比較など、様々なレベルの対応関係を考慮してもよい。

#### 【0032】

上述したように、パケットのバス・サイクルがサブシステムに割り当てられると、このサブシステムは、そのバス・サイクルに関連したスロットの間要求を発行する権利を有する。これによって、異なるサブシステムに帯域幅を事前に割り当てることが可能になる。バス・サイクルが異なるサブシステムに前もって割り当てられるため、TDM Aおよび予測可能な帯域幅の利点は保証される。この特徴は、リアルタイム帯域幅要件が固定のクライアントをサポートする際には非常に重要である。これら2つの概念すなわち、時分割多重アクセスと固定待ち時間のパイプライン式トランザクションを組み合わせることで、待ち時間と帯域幅の両方を保証する極めて高性能の通信プロトコルに必要なものが得られる。

10

#### 【0033】

上記にて簡単に説明したように、第1レベルの調停は以下のようにして行われる。一実施形態では、利用可能なすべてのバス・サイクルを複数の繰り返しフレームに区切る。説明の都合上、フレームを複数のパケットに区切り、これを複数のサイクルに対応させる。例えば、256サイクルのフレームを2サイクルのパケット128個に区切る。このように、フレームを2サイクルのパケット128個から構成することができ、このようなデザインではフレームが256回のバス・サイクル各々で繰り返される。第1レベルの調停スキームは、異なるサブシステムへのパケットの割当てを中心としたものである。イニシエータ・サブシステムのみがコマンドを発行できるため、パケットが割り当てられるのはイニシエータ・サブシステムだけになる。

20

#### 【0034】

本実施形態では、各イニシエータ・サブシステムが同期カウンタを有する。この同期カウンタは、ロックステップにおいて他のすべてのカウンタと共に動作する。このカウンタの出力を利用して、現在のカウンタ値の出力に関連したパケットが特定のインタフェース・モジュールに割り当てられているかどうかを判断する基準となる出力を発するプログラム可能な論理回路をインデックス化する。カウンタを各イニシエータ・インタフェース・モジュールに実装し、これらのカウンタをロックステップにおいて維持することで、システムは、好ましくは2つのイニシエータ・インタフェース・モジュールが同一のパケットに割り当てられない範囲で完全に分散されたTDM Aタイム・ホイールを実装する。インタフェース・モジュールにおけるプログラム可能な論理回路は、システムの初期化時に割当てに関する情報を用いて事前にプログラムされ、および/またはシステムの実行時に通信バスを異なるイニシエータ・サブシステムに介して発行されるコマンドによって再プログラムしてもよい。これによって、システム内で帯域幅を動的に再割当てすることができる。

30

#### 【0035】

パケットの割当ては、例えばパケットが存在する直前のバス・サイクルなど、パケットが存在する前のバス・サイクル中に行われるのが好ましい。様々な実施例を用いることができるが、プログラム可能な論理回路の最も用途が広い実施例は、小型のランダム・アクセス・メモリ(RAM)の形としてのルックアップ・テーブルである。このようなRAMの深度は、フレーム内のパケットの数に等しい。したがって、単にフレーム・カウンタでRAMにインデックス化するだけでTDM Aタイム・ホイールが実装される。好ましくは、テーブルが格納されているRAMはユーザから可視で、標準的なデータ・トランザクションで用いられているものと同じのリード/書き込みコマンドを用いたソフトウェアによって、読み取りおよび書き込みのために通信バスを介してアクセス可能なものである。これによって、通常のリード/ライト・トランザクションに匹敵するトランザクション速度でバスを介してシステムの帯域幅割当てを変更し、割当てを変更して必要に応じて帯域幅を不均等に分散させ、特定のアプリケーションをサポートするようにすることができる。

40

#### 【0036】

特定の状況下では、サブシステムが1つまたはそれ以上の事前に割り当てられたサイクル

50

を使用しない場合がある。これは、いくつかのシステム・トラフィックが予測可能であるが、予測可能なトラフィックですら一律ではないような場合に起こり得る。例えば、ディスプレイに転送されるフレーム・バッファのトラフィックは非常に予測しやすいが、ディスプレイが水平方向および垂直方向の再トレースを行うとトラフィックのフローは中断される。さらに、イニシエータ・サブシステムの中には、性能要件が極めて低いものもあり、このようなサブシステムに帯域幅を事前に割り当てることで未使用のサイクルが多く発生する場合がある。したがって、時分割多重されたアクセス・プロトコルを第2レベルの調停で改善するのが望ましい。

**【 0 0 3 7 】**

第2レベルの調停では、事前割当てがなされないパケットまたは事前割当てはなされたが使用されなかったパケットについては、バス上の他のイニシエータ・サブシステムがトークン・パッシング機構を用いた公平なラウンド・ロビン・スキームでその権利を争う。システム内のいずれかのサブシステムがバスを使用できれば、このような第2レベルの調停が使用されないパケットが残らないようにすることができる。この第2レベルの調停の機構が利用可能なシステム帯域幅に対する全体の利用効率を高くする。それにもかかわらず、上述した通信方法の利点がすべて維持されている。

**【 0 0 3 8 】**

本実施形態では、第2レベルの調停を、トークン・パッシング機構を用いた公平なラウンド・ロビン・スキームとして行う。第2レベルの調停に参加するイニシエータはすべて、トークンを巡回させるようリングに接続されている。一般に、予測不能な待ち時間を許容できるイニシエータが第2レベルの調停に参加する。また、第1レベルの調停に参加するサブシステムを第2レベルの調停に参加させることも考慮されている。初期化時に1つのイニシエータ・サブシステムにトークンを割り当てる。このトークンは、現在のパケットに関連するバス・サイクル中、バス・コマンドを発行する条件付きの権利を示す。この権利は、パケットに事前に割り当てられ、パケットの使用権をアサートしていない所有者または割当てがなされていないパケットに依存するため、条件付きである。上述した第1レベルの調停の後に割当てがなされないパケットが残った場合、第2レベルの調停を用いてこのパケットの調停を行う。このように、T D M Aスキームで最初の拒否権を割り当て、事前に割り当てられた所有者が自己のパケットの使用権をアサートしなければならないが、このパケットを最終的にトークンを持つイニシエータが利用できるようにする。

**【 0 0 3 9 】**

各調停サイクル（すなわち、各パケット）では、トークンがこれを前のサイクルで保持していたイニシエータに残る場合や、このトークンを要求している次のイニシエータにリングを巡回して送られる場合があるが、既存のトークン・リングシステムとは異なり、本発明の一実施形態では、トークンをリング内で完全に1周させて供給元であるイニシエータに戻してもよい。トークンの動きは、システム条件とシステムのルールに定められた公平性の制約に左右される。例えば、2つ以上のパケットに対してトークンを使うことは1つのイニシエータにとっては公平ではないが、別のイニシエータ・サブシステムは待つことができることもある。使用する機会を得ないままにトークンを解放することはどのイニシエータ・サブシステムにとっても不公平である。このように、一実施形態では、トークンを要求し、これを受信するイニシエータ・サブシステムが、通信バス上で正確に1パケット分のコマンドを開始する機会が得られるまでこのトークンを保持できるようにトークン・パッシングのルールを打ち立てる。その機会が、パケットが所有者に事前に割り当てられた装置によってなされる使用の判断次第で、到達する調停サイクルでトークンが発生する場合もあれば、多数のサイクルにわたって発生しない場合もある。最後に、トークンを持っているイニシエータでこれを使う機会が得られると、このイニシエータはそのT o k e n O u t信号を論理1にドライブしてリング内でのトークンの巡回を開始する。トークンを要求するイニシエータが他に存在しない場合、このトークンはリングを完全に1周して送信元に戻る。

**【 0 0 4 0 】**

トークンがリングを無限に巡回するのを防止するために、トークンの送信を開始するイニシエータ・サブシステムがリングを一時的に崩すようにすると好ましい。このように、いずれかの調停サイクルでトークンの送信を開始するイニシエータ・サブシステムを、その調停サイクルの終了時まで再度自己のところで終わるようにすることができる。線が不確定な状態になり、不適切なバス動作と電力の散逸量の増大を招くことがないよう、通信バス信号は常にアクティブにドライブされているのが望ましいことに注意されたい。現実問題として、事前に割り当てられたパケット所有者が最初の拒否権をアサートしない場合は常に、トークンを持つことになるイニシエータがバス・コマンドを開始するようにする。このような場合、発行すべき有用なコマンドが存在しないのであれば、トークンを持っているイニシエータに不要なコマンド（無害なリードまたはライト、あるいはNOP/Idleコマンド）を開始させる。

10

#### 【0041】

調停サイクルの開始時にトークンを所有しているイニシエータがトークンを使用する機会を得られず、発行すべきコマンドがある場合、このイニシエータはトークンを保持し続けることができる。それ以外の場合、イニシエータは信号線をハイにし（例えば、TokenOut）、他のイニシエータ・サブシステムに対して、第2レベルの調停でトークンの権利を争えることを示す。トークンを持っているイニシエータ・サブシステム（すなわち、最後にトークンを使用したイニシエータ・サブシステム）は、このトークンをリング内の次のイニシエータ・サブシステムにパスする。次のイニシエータ・サブシステムは、サイクルを使用できるのであればこのトークンを専有する。それ以外の場合、トークンは割

当てがなされるまで巡回される。いずれの場合も、その特定のイニシエータ・サブシステムが次のトークン・リング調停の開始点としてトークンを保持する。トークンを使用するイニシエータ・サブシステムが存在しない場合、このトークンは最初にトークンを持っていたイニシエータに戻る。第1レベルと第2レベルのすべてのパケット調停は同一のサイクルで起こる。この第2レベルの調停では保証帯域幅は得られないが、アクセス権の保証に必要なものが得られる。第2レベルの調停を使用すると、他のイニシエータがバスへのアクセスしようとしている場合、どのイニシエータ・システムも2つ以上のパケットに対してバスを支配することはできない。

20

#### 【0042】

図4aは、第2レベルの調停の一実施形態の状態を示し、図4bの表はこの状態を簡単に説明したものである。本実施形態では、2つの実装態様によってリングでの伝搬遅延を最小限に抑える。第1に、トークンを生成しているインタフェース・モジュールは、純粹に現在の状態の機能としてそれを行う。さらに、そのインタフェース・モジュールはトークンを決められたとおりにパスすることを決定するか、または純粹には現在の状態の機能として行わない。これら2つの特徴は、トークンをできる限り迅速にリングに送り出し、トークンが中間のインタフェース・モジュールを伝搬するのにどれだけの時間が必要か不確かであることによる遅延を避けることができる。

30

#### 【0043】

状態への入力は、ReqToken、TokenIn、Assert、Reset\_n、Firstであり、出力はTokenOut、GrantToken、nextstateである。ReqTokenは、インタフェースがトークンを要求していることを示す。TokenInは、このイニシエータに対してトークンを利用できることを示す。Assertは、事前に割り当てられたイニシエータがそのパケットを使用することを示すのに使われる。reset\_nは、システムを強制的にリセット状態にするのに使われる。TokenOutは、トークンをリング内の次のイニシエータにパスするのに使われる。GrantTokenは、要求の発行準備をすべきであることをイニシエータに通知するのに使われる。NextStateは、イニシエータ・サブシステムの特定のインタフェース・モジュールの次の状態を示す。

40

#### 【0044】

状態NOTOKENでは、サブシステムのインタフェース・モジュールはトークンを必要

50

とせず、所有もしていない。このため、TokenInでトークンが到着すると、TokenOutによってすみやかにパスされる。トークンを要求しているインタフェース・モジュールを一度サンプリングすると、状態マシンはNOTOKENを離れる。状態WANTTOKENでは、インタフェース・モジュールはトークンを持っておらず、トークンを必要としている。したがってこのモジュールはリングを崩し、TokenInでトークンが届いたらこれを取得できるようにする。トークンが届くと、インタフェース・モジュールは、事前に割り当てられたスロット所有者がアサートを出していなければ(GrantTokenの通知によって)このトークンを使用する。アサートがアクティブである場合、トークンを使用できる機会が得られるまでこのモジュールがトークンを保持するため、状態マシンはHAVETOKENに移行する。

10

**【0045】**

状態HAVETOKENでは、モジュールにはトークンがあり、モジュールはトークンを使用できる機会を待っている。どのモジュールにもアサートがなされない状態になると、その時点でスロットが利用可能になり、上記のモジュールが(GrantTokenを通知することによって)これを使用する。この状態が発生するか、あるいはインタフェース・モジュールが(おそらく、事前に割り当てられたスロットによって要求が満足されたことが理由で)ReqTokenのアサートを中止した場合、状態マシンはGENTOKENに移行する。このモジュールにはトークンがあるため、TokenInラインに注意を払う必要はない(例えばゼロでなければならない)。状態GENTOKENでは、モジュールは調停サイクルの最初にトークンをリングで巡回させ始める。このモジュールは、WANTTOKENの(すなわちトークンをアクティブに要求している)モジュールがない場合はリングを崩さなければならない。トークンがリングを1周してこのモジュールに戻ると、このインタフェース・モジュールはGrantTokenを受信し、装置がトークンを要求したかどうかとは無関係にコマンドの開始を要求される。

20

**【0046】**

状態マシンがリセット状態にある(Reset\_nがローにドライブされている)間、開始状態は、最初のラベルが付されたモジュール以外のすべてのインタフェース・モジュールに対してNOTOKENに設定される。最初の装置はリセット時にトークンを持って動作が開始されるという点でユニークであるため、リングにいつでも正確に1つのトークンが存在するようにされる。さらに、本実施形態では、ReqToken列のDon't Careエントリ(リセット時以外)によって、正確に1つの装置が常にコマンドを開始するよう保証されていることに注意されたい。事前に割り当てられたスロット所有者がアサート・ラインをハイにした場合は、調停の終了時にトークンを持つことになるインタフェース・モジュールがコマンドを開始しなければならない。

30

**【0047】**

上述した割当法を利用して、要件の異なる様々なクライアントをサポートすることができる。特定のクライアントの需要を満たすよう同一のシステムをプログラムすることができるため、事前に割り当てられたクライアントおよびパケットの適切な部分をシステムで必要としていることに応じてプログラムすることができる。

**【0048】**

バス上に発行される最小限のコマンドの組み合わせは、リード・コマンドと書込みコマンドである。この通信方法を自然に拡張し、バスに接続されたサブシステムを制御するように特別に設計された制御コマンドをはじめとする他のコマンドをサポートすることができる。図5は、本発明の教示内容に従って動作する完全パイプライン式の固定待ち時間バスを介して行われるトランザクションの一例のタイミング図を示している。各トランザクションは、調停部分とコマンド/アドレス部分とデータ/応答部分とから構成される。バスは固定待ち時間バスであるため、2つのトランザクションすなわち要求と応答との間には(クロック・サイクル数の)固定の遅延がある。図5では、調停とコマンド/アドレスとの間の待ち時間は1サイクルであり、コマンド/アドレスとデータとの間の待ち時間は2サイクルが2回であると仮定する。パケット(調停ユニット)は1バス・サイクルである

40

50

。サイクル1では、イニシエータ・サブシステムI-Dが自己に事前に割り当てられたパケットを使用する権利をアサートするため、サイクルタイムスロット2では、バスを介してI-Dがコマンドおよびアドレスをターゲットに送信する。説明の都合上、このコマンドをコマンド1とする。すべてのターゲット・インタフェース・モジュールには、トランザクションがそれらのクライアント・サブシステムに対して指定されたものであるかどうかを判断するためのアドレス・デコード用論理が含まれている。タイムスロット4では、指定されたターゲット・インタフェース・モジュールが要求コマンドに应答する。リード要求コマンドの場合、リード・データがターゲットからイニシエータに送られる。ライト要求コマンドの場合、ライト対象となるデータがサイクル4でイニシエータからターゲットに送られる。

10

## 【0049】

サイクル4においてターゲットからデータ应答ラインを介してターゲットによってドライブされた应答信号は、トランザクションが成功したかどうかを示すものである。一実施形態では、「Valid」、「Busy」、「Retry」、「No Response」の4つの应答が可能である。「Valid」应答はトランザクションが成功したことを示す。「Busy」应答はターゲット・モジュールがコマンドに应答可能であったことを示す。「Retry」应答はリード・コマンドについてのみ許容されるものであり、ターゲットがコマンドを満足しようとしていたにもかかわらず、一定のバス待ち時間の間にこれを達成できなかったことを示す。イニシエータはその要求を後で再トライする。「No Response」の場合、コマンドと一緒に供給されたアドレスがどのターゲットとも一致せず、どのターゲットも应答しなかったことを示す。

20

## 【0050】

続けて図5を参照すると、クロック・サイクル2で第2の調停サイクルが開始され、サイクル3で別のトランザクションが発行される。この状況では、事前に割り当てられたイニシエータはいずれもバス・サイクルの使用権をアサートしていないため、サイクル2でトークンを持つことになるイニシエータ・サブシステムI-Eがサイクル3でコマンドおよびアドレスを発行する。このコマンドに関連したデータと应答信号は、バス・サイクル待ち時間後にサイクル5で発行、転送される。

## 【0051】

上述したように、パケットの動的な再割当てに必要なものが得られるよう、インタフェース・モジュールはプログラム可能であると好ましい。一実施形態では、各サブシステムのインタフェース・モジュールが、モジュールおよびバスの構成データを格納するための一組の構成レジスタ(240、図2)を有する。

30

## 【0052】

好ましくは、構成レジスタは、対応するサブシステムに事前に割り当てられたパケットを識別するデータを格納する。構成レジスタはシステム・アドレス空間にアドレスを有し、よって利用可能なバス・トランザクションを用いて読み出しおよび書き込みが可能である。構成レジスタは、バッファされた構成レジスタとバッファされない構成レジスタの二組あるのが好ましい。バッファされないレジスタに書き込まれるデータは、レジスタへの書き込み動作がなされるとインタフェース・モジュールおよびシステムの他の部分から可視となる。バッファされたレジスタに書き込まれたデータは、保持レジスタ位置に格納され、そのデータが保持レジスタから構成レジスタに転送された後でのみ可視となる。この転送は、バス上のブロードキャスト・コマンドによって開始される。ブロードキャスト・コマンドは、すべてのインタフェース・モジュールに同時に影響を与えるコマンドである。このため、バッファされたレジスタは一般に、新しいデータをシステムの他の部分から見えるようにできる前にいくつかのインタフェース・モジュールを同時に更新することが望ましい場合に使用される。

40

## 【0053】

バッファされたレジスタの使用例は、システム内のアドレス・マップへの割り当てである。各ターゲット・インタフェースまたはモジュールは、ローカルに格納されたマスクと、

50



ターゲットがどのアドレスに応答するかを指定するマッチ・データ値を有する。バス上のアドレスとマッチ・フィールドとを比較し、結果をマスクする。マスクされた結果のすべてのビットが0である場合、この特定のインタフェース・モジュールはアドレス指定されている。マスク/マッチ・データは、構成レジスタに格納され、通常のバス・コマンドによって書き込み可能なものである。システムを正しく動作させるには、ターゲット・アドレスの範囲がシステム内で一意に割り当てられている必要がある。例えばアドレスのスイッチングなどアドレスの変更がなされる場合、影響されるすべてのインタフェース・モジュールで同時に新しい割り当てを行うようにする。このようにしないと、一方のアドレス・マップから他方のアドレス・マップへの遷移時にアドレス空間の重なりができてしまう場合がある。

10

**【0054】**

上述したように、本発明は、バッファされた構成レジスタとバッファされない構成レジスタとの使用に限定されるものではない。さらに、構成レジスタ内のデータのうちのいくつかまたはすべてがシステムの耐用期間の間変更を必要としない場合、これらのすべてまたはいくつかのデータを読み専用メモリ（ROM）に格納することができる。ROMは一般に必要なダイ空間が少ないため、構成位置への不要な書き込みが発生して不適切なシステム構成になるのを防止できる。一般的なシステムには、バッファされた構成レジスタ、バッファされない構成レジスタ、ROMの組み合わせを有するインタフェース・モジュールが含まれていると好ましい。構成レジスタに含まれる情報は、イニシエータ・サブシステムに事前に割り当てられたパケットに関するデータ、待ち時間、要求アドレス・マッチングの目的でインタフェース・モジュールとターゲットの接続されたアドレス装置との間の通信周波数から構成される。

20

**【0055】**

上記の重要な特徴の1つは、レジスタがシステム・アドレス空間の一部であるため、利用可能な通常のリード・コマンドと書き込みコマンドとを用いて、システムのリアルタイム動作時に構成データを書き込むことができるという事実である。これによって、システム設計者は通信システムの多くの重要な特徴を簡単なソフトウェアまたは場合によっては事前に計算してROMに格納したコマンドシーケンスによってすら変更することができる。

**【0056】**

従来技術にまさる本発明の重要なユニークな特徴の1つは、バス待ち時間がシステムの初期化時またはシステムの実行時にプログラム可能であるという事実である。好ましい実施形態では、待ち時間は通信バスを介してプログラム可能である。完全にパイプライン化された通信バスであるため、このような動作にはブロードキャスト・コマンドが必要になり、パイプライン深度がすべてのインタフェース・モジュールで同時に変更される。低待ち時間の応答がシステムにとって重要である場合（メモリへのアクセスなど）は、インタフェース・モジュールの適当な構成レジスタに新しい待ち時間を書き込むことによって、低待ち時間を通信システムにプログラムすることが可能である。一方、バスを介してなされる重要な通信の大半が長待ち時間のサブシステム（非常に迅速に応答することのできないシステム）間でのものである場合、システム待ち時間を上記より高く設定することができる。通常のバス・トランザクションに匹敵する速度で待ち時間をプログラムできるという

30

40

**【0057】**

待ち時間を変更できるだけでなく、適当な構成レジスタを更新する同一の方法を使用して帯域幅割り当ても変更することが可能である。バスを介しての通常のリード・トランザクションまたはライト・トランザクションに匹敵する速度で、システムの初期化時または実行時のいずれかに、帯域幅割り当てをプログラムできるという事実、従来技術からの大きな改善である。

**【0058】**

プログラム可能な帯域幅割り当てとプログラム可能なバス待ち時間とを組み合わせることで、他にはない利点が得られる。特に、イニシエータおよびターゲットの個々の構成ごとに、

50

システム設計者がシステムの性能を最適化することが可能である。さらに、帯域幅割当および待ち時間を動作時に更新可能であるため、条件および構成を変更して最適な状態を維持することができる。さらに、後述するように、ターゲットおよびそのインタフェース・モジュールが通信を行う周波数をイニシエータ装置がプログラムできるようにすることで、柔軟性と構成可能性がもたらされる。これは、イニシエータおよび/またはターゲット・クライアントの様々なサブシステム・タイミング要件に適合する場合に特に望ましい。

#### 【0059】

帯域外の通信を管理する一組のシステム・レベル制御線（図1においてフラグと呼んでいる）を加えることによって、バス機能性をさらに高めることが望ましい。帯域外の信号は、アドレス/データ指向のパイプライン型バス・モデルには沿わない信号である。一例として、装置のステータス信号、割り込み、ハンドシェイク信号があげられる。好ましい実施形態では、これらの線は、以下のようにして基本的なリード動作の性能を高めるのにも使用される。

#### 【0060】

例えば、図5に示すようなシステムの動作では、ターゲット・サブシステムがリード要求コマンドに適時に応答してバス待ち時間要件を満たし得る場合、「Valid」応答が応答バス経由で送信される。ターゲット・サブシステムがバス待ち時間要件に適時に応答できない場合、「Retry」応答が送信される。所望のデータがいつ利用可能であるのかイニシエータで推測し、バス・サイクルの浪費や待ち時間の増大を引き起こすのではなく、ターゲット・インタフェース・モジュールで帯域外信号（本願明細書では、フラグとも呼ぶ）のうちの1つを使用する。ターゲット・インタフェース・モジュールは、ターゲット・インタフェース・モジュールでデータを取得した後、フラグを用いてイニシエータに正確な指示を返す。

#### 【0061】

図6は、帯域外信号線を用いた再トライ機構を示している。クライアントが通信バスの待ち時間を満たすことができない場合、ターゲット・インタフェース・モジュールはイニシエータに応答パケットを発行し、「Retry」を示す。同一のタイムスロットで、フラグ線信号のビット数をFlagNumライン（分かりやすくするために、図1のフラグ・ラインの一部である）で送信する。これは、いつ再トライするかを示す識別されたフラグを監視すべきであることをイニシエータ・サブシステムに対して示す。

#### 【0062】

図6の例では、Read要求コマンドをサイクル1で送信する。バス待ち時間は2サイクルであるため、サイクル3でターゲット・インタフェース・モジュールによってデータを返さなければならない。しかしながら、データはまだ利用できないため、ターゲットは再トライ応答を応答線で返し、フラグ線の数（本例では7）をFlagNumバスで返す。ターゲット・インタフェース・モジュールがそのクライアント（すなわち、接続されている装置）からデータを受信すると、このデータをローカルメモリに格納し、示されたフラグ線をハイにドライブし、イニシエータに対してデータが利用できるようになったことを示す。図6では、サイクル4の後に、ターゲットがフラグ7でデータが利用できることを示している。この時点で、イニシエータ・サブシステムは要求コマンドを再発行することができる。これは、図3ではサイクル6で行われている。ターゲット・インタフェース・モジュールが要求コマンドを満たす（図3ではサイクル8）と、この要求コマンドと関連付けられていたフラグをアサート解除する。

#### 【0063】

通信システムは、異なるターゲット・インタフェース・モジュールに割り当てられる複数のフラグを含むと好ましい。フラグ線をモジュール同士で共用することができ、また特定のモジュールで専有することもできる。好ましい実施形態では、フラグの割当てが構成レジスタに格納されているため、この割当てを通常のバス・トランザクションに匹敵する速度で通信バスを介してプログラム可能である。

#### 【0064】

フラグ線を様々な目的に使用することができる。例えば、フラグ線の異なった使い方として、ハンドシェイク信号および割込み信号の管理があげられる。これらの信号は、インタフェース・モジュールまたはサブシステムで特別なイベントを示す信号である。このような信号の例として、エラー信号またはビジー状態（ストップ送信要求）を示す信号、待機状態（サブシステムがいつでも要求を受けられる状態にある）を示す信号があげられる。これらの信号は本来、帯域外信号であり、フラグ線を用いて転送される。これらの目的のため、割込み信号またはハンドシェイク信号向けの目的で、フラグ線はターゲット・インタフェース・モジュールおよびそのサブシステムクライアントに一意に割り当てられている。好ましい実施形態では、このデータ割当てが構成レジスタに格納されているため、システムで変更が必要になった際に動的に変更することができる。イニシエータ・サブシステムで同一の情報をプログラムし、ハンドシェイクまたは割り込みの目的で監視すべきフラグ線がどれであるのかイニシエータが分かるようにする。

10

**【0065】**

上述したように、バスとサブシステムとの間のインタフェース・モジュールは、クライアントとの間をインタフェースしながらバス・プロトコルを実装するための論理を有する。さらに、サブシステムのクライアント・クロックをシステムバス・クロックから導出する別の論理（245、図2）がインタフェースに含まれていると好ましい。構成レジスタを介したインタフェース・モジュールによって、システムの特徴および性能要件が極めて異なる様々なシステムでサブシステムを再利用できるレベルのプログラム性が得られる。システムバス・クロックの周波数は、特にシステムの所望の性能を満たすためにバスで必要な全帯域幅であるシステム要件によって定まる。クライアントのクライアント・クロック周波数をシステムバス・クロックから切り離すことが可能であり、これをクライアントで必要な機能性と関連させるようにする。したがって、インタフェース・モジュールによって、システムとサブシステム要件とを切り離すための強力な手段が得られる。

20

**【0066】**

従来の固定待ち時間バス・システムでは、利用可能な帯域幅を増やすためにはバスとすべてのクライアントの動作周波数を増やす必要がある。クライアントが待ち時間を満たすために短時間で応答する必要があるためである。本発明のシステムを用いると、バス周波数の増加に応じてバス待ち時間（サイクルで測定）を増すことができる。これによって、クライアントがほぼ一貫した周波数で動作しやすくなる。全体の待ち時間（時間）が比較的一定になるためである。デカップリングの形状をインタフェース・モジュールとクライアントとの間の非同期的な接続まで拡張することも考慮されている。

30

**【0067】**

周波数デカップリングに対する要件は、主に本手法を用いることで利益のある多種多様なシステムに基づくものである。1つのシステム例として、ワイヤレスのパーソナル通信機があげられる。この通信機では、マイクロコントローラ・コア、組み込み式メモリ、キーパッド入力、液晶ディスプレイ（LCD）出力などの低性能でコンピュータ由来のサブシステムと共に、（ワイヤレス通信を行うために）性能が一定のリアルタイム・サブシステムが必要である。通信用のキーデザインの目標は、信頼できる通信とできるだけ多くの機能とを（マイクロコントローラ上のソフトウェアに実装して）低電力で提供することである。2つ目のシステム例としては、デジタル衛星テレビ放送を受信するためのセットトップ・ボックスがあげられる。セットトップ・ボックスのデザインでは、さらに高いビットレートでのデータ通信とビデオ・データおよびオーディオ・データを解凍するためのさらに高い計算性能とが必要とされ、システム全体での性能を保証してドロップ・フレームを回避しなければならない。3つ目の例は、非同期転送モード（ATM）スイッチである。このスイッチのデザインでは、性能の監視と診断を行うだけの目的で中央処理装置（CPU）が必要になることが多い。通常の動作では、共用相互接続を介した同一のサブシステム間でパケットを切り換える必要がある。多くの回路すなわちスイッチング・アプリケーションの場合と同様に、性能保証はATMスイッチを正しく動作させる上で重要なものである。

40

50

## 【0068】

以下、様々なシステムのデザイン再利用によって得られる機会を示す。ビデオオンデマンド用のワイヤレス・システムと、パーソナル通信機からの双方向ワイヤレス・サブシステム、ATMスイッチからのATMパケット・フレーム・エンジン、セットトップ・ボックスからのビデオ解凍サブシステムとを組み合わせることができる。相互に動作させることが意図されていなかったサブシステム同士を組み合わせ、高速に集積化されコスト効率のよいシステムにすることが可能である。

## 【0069】

以上、好ましい実施形態について本発明を説明した。当業者には、上記の説明に鑑みて、多数の変更、修正、バリエーションおよび用途が理解できようことは明らかである。

10

## 【図面の簡単な説明】

【図1】 本発明の教示内容に従って動作するシングル・チップ通信システムの一実施形態を示す図である。

【図2a】 イニシエータ・インタフェース・モジュールの一実施形態の簡略ブロック図を示す図である。

【図2b】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図2c】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図2d】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

20

【図2e】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図2f】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図2g】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図2h】 図2aのインタフェース・モジュールに対して入出力される信号を説明するための図である。

【図3a】 バスへのアクセスについて調停をセットアップするためのプロセスの一実施形態の簡略流れ図である。

30

【図3b】 要求を発行し、要求に応答するためのプロセスの一実施形態を説明するための図である。

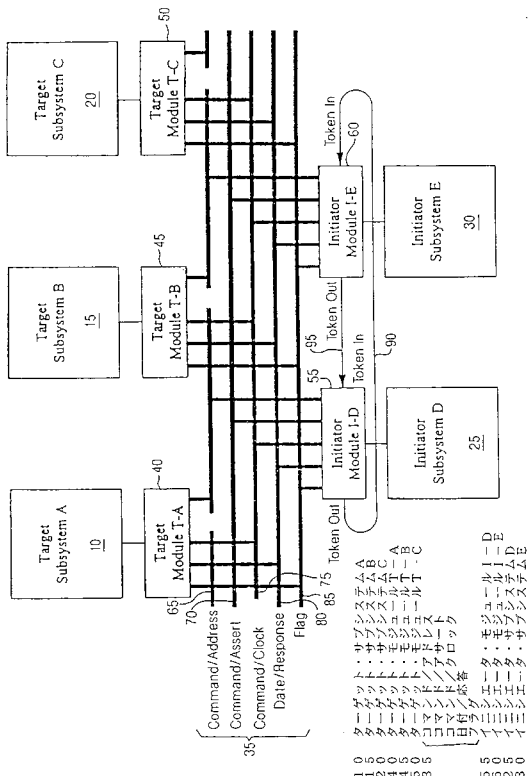
【図4a】 第2レベルの調停での調停状態の一実施形態を示す図である。

【図4b】 第2レベルの調停での調停状態の一実施形態を示す図である。

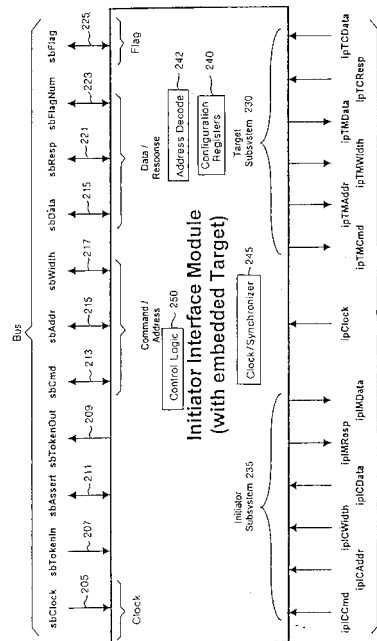
【図5】 本発明の教示内容によるバス・トランザクションを示すタイミング図である。

【図6】 本発明の教示内容による再トライ・プロセスを示す図である。

【図 1】



【図 2 a】



Client Interface  
 Clock クロック  
 Command/Address コマンド/アドレス  
 Command/Assert コマンド/アサート  
 Data/Response データ/レスポンス  
 Flag フラグ  
 Initiator Interface Module Initiator インターフェイス・モジュール  
 (with embedded Target) (埋込みターゲット)  
 sbClock sbClock  
 sbError sbError  
 sbFlag sbFlag  
 sbFlagNum sbFlagNum  
 sbReset sbReset  
 sbTokenIn sbTokenIn  
 sbTokenOut sbTokenOut  
 sbWidth sbWidth  
 sbResp sbResp  
 sbAddr sbAddr  
 sbCmd sbCmd  
 sbAssert sbAssert  
 tpClk tpClk  
 tpTMCmd tpTMCmd  
 tpTMCmdValid tpTMCmdValid  
 tpTMBurst tpTMBurst  
 tpTMWidth tpTMWidth  
 tpTCDataIn tpTCDataIn  
 tpTMDataOut tpTMDataOut  
 tpTMDataValid tpTMDataValid  
 tpTCReady tpTCReady  
 tpTCResp tpTCResp  
 tpTCInterrupt tpTCInterrupt  
 tpTMReset tpTMReset

【図 2 b】

信号名	周期	幅	機能
Signal Name	Cycle	Width	Function
sbAssert	S0	1	Slice Owner Asserts it will use the Slot (Initiator Only)
sbTokenIn	S0	1	Round-Robin Token Grant In (Initiator Only)
sbTokenOut	S0	1	Round-Robin Token Grant Out (Initiator Only)
sbAddr	S1	32	Transaction Address
sbCmd	S1	3	Transaction Type
sbWidth	S1	2	Transaction Width
sbData	SM	64	Read/Write Data
sbResp	SM	3	Transaction Response
sbFlagNum	SM	6	Retry Flag Number
sbError	SN	1	Backplane Error
sbFlag	NA	≤32	Device Event Flags
sbReset	NA	1	Backplane Reset
sbClock	NA	1	Silicon Backplane Master Clock

【図 2 e】

信号名	幅	機能
Signal Name	Width	Function
tpTMCmd	≤32	Transaction Address
tpTMCmdValid	3	Transaction Type
tpTMBurst	1	Command Valid
tpTMWidth	3	Burst type
tpTCDataIn	2	Data Width
tpTMDataOut	≤64	Read Data
tpTMDataValid	≤64	Write Data
tpTCReady	1	Write Data Valid
tpTCResp	1	Client Available
tpTCInterrupt	2	Target Client Response
tpTMReset	1	Target Client Interrupt
tpClk	1	Target Client Reset
	1	Target Client Clock

【図 2 c】

トランザクションタイプ

sbCmd [2]	sbCmd [1]	sbCmd [0]	Transaction Type
0	0	0	Idle
0	0	1	Write
0	1	0	Read
1	1	1	Broadcast

【図 2 f】

トランザクションタイプ

tpTMCmd [2]	tpTMCmd [1]	tpTMCmd [0]	Transaction Type
0	0	0	Idle
0	0	1	Write
0	1	0	Read

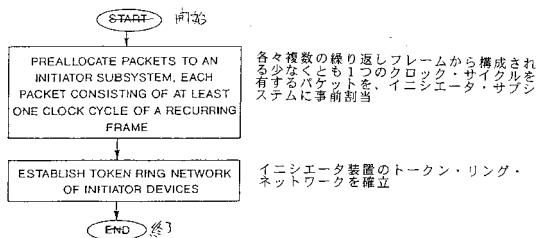
【図 2 d】

sbResp [1]	sbResp [0]	
0	0	No Response (NULL)
0	1	Data Valid/Accepted (DVA)
1	0	Retry (RTRY or RTRYFLG)
1	1	Busy (BUSY)

【図 2 g】

信号名	幅	機能
Signal Name	Width	Function
ipIMValid	1	IM Ready to Accept Data
ipICValid	1	IC Request Valid
ipICAddr	32	IC Transaction Address
ipICCmd	3	IC Transaction Type
ipICBurst	3	IC Burst Mode
ipICWidth	2	IC Data Transaction Width
ipIMResp	2	IM Transaction Response
ipICData	≤64	Write Data from Client to Module
ipIMData	≤64	Read Data from Module to Client
ipTMCmd	3	TM Transaction Type
ipTMAddr	≤32	TM Transaction Address
ipTMBurst	3	TM Burst Mode
ipTMWidth	2	TM Data Transaction Width
ipTCReady	1	TC Ready Signal
ipTCInterrupt	1	TC Interrupt signal
ipTCResp	2	TC Transaction Response
ipTMData	≤64	Write Data from Module to Client
ipTCData	≤64	Read Data from Client to Module
ipReset	1	Initiator Client Reset
ipClk	1	Initiator Client Clock

【図 3 a】

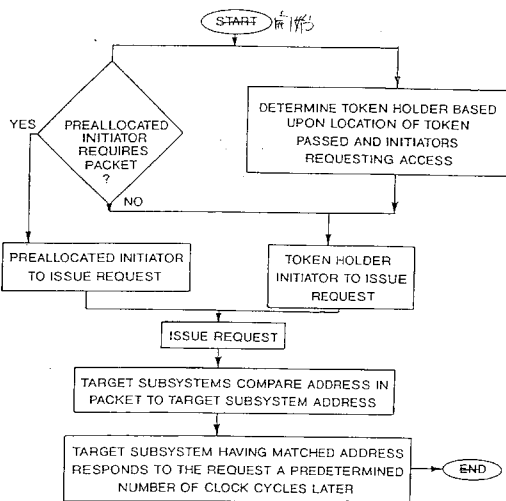


【図 2 h】

トランザクションタイプ

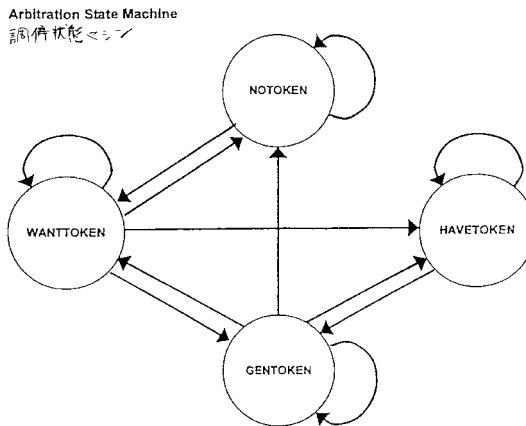
ipICCmd[2]	ipICCmd[1]	ipICCmd[0]	Transaction Type
0	0	0	Idle
0	0	1	WriteUnord
0	1	0	ReadUnord
0	1	1	ReadEx
1	0	0	ReadOrd
1	0	1	WriteOrd
1	1	0	ReadPre
1	1	1	Broadcast

【図 3 b】



- ① 事前に割り当てられたイニシエータがパケットを要求しているか
- ② バスに渡されたトークンの位置とアクセスを要求しているイニシエータに基づいてトークン・ホルダを判断
- ③ 事前に割り当てられたイニシエータで要求を發行
- ④ トークン・ホルダ・イニシエータで要求を發行
- ⑤ 要求を發行
- ⑥ ターゲット・サブシステムがパケット内のアドレスとターゲット・サブシステムのアドレスとを比較
- ⑦ マッチしたアドレスを有するターゲット・サブシステムが所定の数のクロック・サイクルだけ後に要求に回答

【図 4 a】



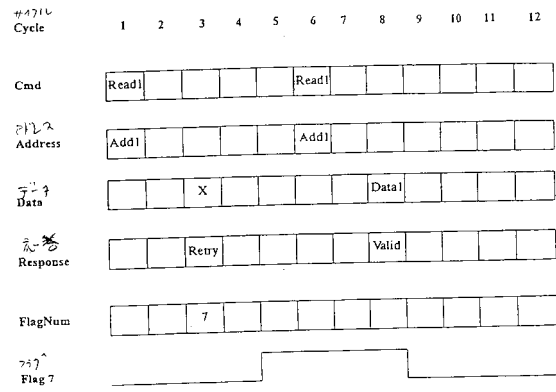
【図 4 b】

Current State	Inputs					Outputs		Next State
	ReqToken	TokenIn	Assert	Reset_n	First	TokenOut	Grant Token	
NOTOKEN	0	X	X	0	X	TokenIn	0	NOTOKEN
NOTOKEN	1	X	X	0	X	TokenIn	0	WANTTOKEN
WANTTOKEN	0	0	X	0	X	0	0	NOTOKEN
WANTTOKEN	1	0	X	0	X	0	0	WANTTOKEN
WANTTOKEN	X	1	0	0	X	0	1	GENTOKEN
WANTTOKEN	0	1	1	0	X	0	0	GENTOKEN
WANTTOKEN	1	1	1	0	X	0	0	HAVETOKEN
HAVETOKEN	X	X(0')	0	0	X	0	1	GENTOKEN
HAVETOKEN	0	X(0')	1	0	X	0	0	GENTOKEN
HAVETOKEN	1	X(0')	1	0	X	0	0	HAVETOKEN
GENTOKEN	0	0	X	0	X	1	0	NOTOKEN
GENTOKEN	1	0	X	0	X	1	0	WANTTOKEN
GENTOKEN	X	1	0	0	X	1	1	GENTOKEN
GENTOKEN	0	1	1	0	X	1	0	GENTOKEN
GENTOKEN	1	1	1	0	X	1	0	NOTOKEN
X	X	X	X	1	0	0	0	NOTOKEN
X	X	X	X	1	1	0	0	GENTOKEN

【 図 5 】

1	2	3	4	5
制御 1 Arbitration 1 Pre-allocated to I-D: 最初の 制御	制御 2 Arbitration 2 Token used by I-E: 次の 制御			
コマンド 1 Command 1 I-D	コマンド 2 Command 2 I-E			
アドレス 1 Address 1 I-D	アドレス 2 Address 2 I-E			
		データ 1 Data 1 I-D	データ 2 Data 2 I-E	
		応答 1 Response 1 I-D	応答 2 Response 2 I-E	

【 図 6 】



Retry Mechanism  
リトライ機構

---

フロントページの続き

(72)発明者 ロスイー、 ジョー、 ジョー、 ジョー、 ジョー、 ジョー、 ジョー、 ジョー、 ジョー、 ジョー、  
アメリカ合衆国・94025・カリフォルニア州・メンロパーク・アヴィアベニュー・2181

審査官 西出 隆二

(56)参考文献 特開平07-084945(JP, A)  
特開平06-028306(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/36

G06F 13/362

G06F 13/368

G06F 13/42