

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-178590
(P2004-178590A)

(43) 公開日 平成16年6月24日(2004.6.24)

(51) Int. Cl.⁷

G06F 11/28
G06F 11/22

F I

G06F 11/28 315B
G06F 11/28 L
G06F 11/22 340A
G06F 11/22 340C

テーマコード(参考)

5B042
5B048

審査請求 未請求 請求項の数 1 O L (全 25 頁)

(21) 出願番号 特願2003-392615 (P2003-392615)
(22) 出願日 平成15年11月21日(2003.11.21)
(31) 優先権主張番号 301887
(32) 優先日 平成14年11月22日(2002.11.22)
(33) 優先権主張国 米国(US)

(71) 出願人 501229528
テキサス インストルメンツ インコーポ
レイテッド
アメリカ合衆国、テキサス、ダラス、チャ
ーチル ウエイ 7839
(74) 代理人 100066692
弁理士 浅村 皓
(74) 代理人 100072040
弁理士 浅村 肇
(74) 代理人 100094673
弁理士 林 拓三
(74) 代理人 100118821
弁理士 祖父江 栄一

最終頁に続く

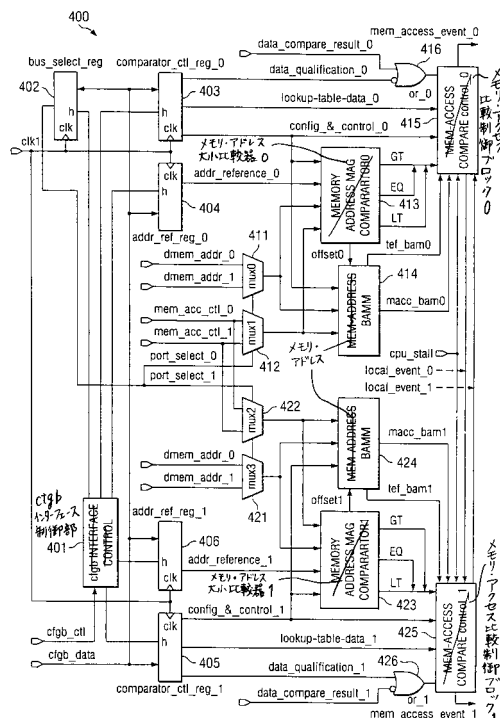
(54) 【発明の名称】 データ一致認定および完全または部分的重複によって、多数のサイズのメモリ・アクセスを検出するアドレス範囲比較器

(57) 【要約】

【課題】 データ一致および完全/部分的重複によって、多数のサイズのメモリ・アクセスを検出するメモリ・アクセス・アドレス比較器を提供する。

【解決手段】 メモリ・アクセス・アドレス比較器は、入力メモリ・アクセス・アドレスをそれぞれの基準アドレスと比較する2つの比較器を含む。比較器は、アドレス・サイズ、完全な重複または部分的な重複、超過、未満、同等、不等、以下、および以上というような、選択可能な判断基準によって一致の指示を出し、選択的に連鎖することができる。入力マルチプレクサは、メモリ・アクセス・アドレス・バスの選択を可能にする。比較器出力コントローラは、対応するデータ一致に選択的に応ずることができる。コンフィギュレーション・インターフェース・コントローラにより、中央演算装置は、メモリ・マップ・レジスタを通じて、基準アドレス、比較データおよび制御機能を指定することができる。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

メモリ・アクセス・アドレス比較器であって、

第 1 基準アドレスを格納する第 1 プログラマブル基準アドレス・レジスタと、

前記第 1 基準アドレスおよび第 1 アドレス入力を受け取り、前記第 1 基準アドレスと前記アドレス入力との間の整合を示す少なくとも 1 つの第 1 整合信号を発生する第 1 アドレス整合マップ作成部と、

前記第 1 プログラマブル基準アドレス・レジスタからの前記第 1 基準アドレスと前記第 1 アドレス入力とを受け取り、前記第 1 基準アドレスと前記アドレス入力上のアドレスとの関係に応じて、超過出力、未満出力または同等出力を発生する第 1 比較器と、

10

第 2 基準アドレスを格納する第 2 プログラマブル基準アドレス・レジスタと、

前記第 2 基準アドレスおよび第 2 アドレス入力を受け取り、前記第 2 基準アドレスと前記アドレス入力との間のオフセットを示す少なくとも 1 つの第 2 オフセット信号を発生する第 2 アドレス整合マップ作成部と、

前記第 2 プログラマブル基準アドレス・レジスタからの前記第 2 基準アドレスと前記アドレス入力とを受け取り、前記第 2 基準アドレスと前記アドレス入力上のアドレスとの間の関係に応じて、超過出力、未満出力または同等出力を発生する第 2 比較器と、

第 1 および第 2 アドレス整合マップ作成部のそれぞれならびに第 1 および第 2 比較器のそれぞれに接続された第 1 および第 2 メモリ・アクセス制御ユニットであって、各々、前記少なくとも 1 つの整合信号、前記対応する比較器の前記超過出力、前記未満出力および前記同等出力の少なくとも 1 つに応じて、対応する第 1 および第 2 ローカル・イベント信号を選択的に発生し、更に、前記他方のメモリ・アクセス制御ユニットの前記ローカル信号に応じて対応する第 1 および第 2 メモリ・アクセス・イベント信号を選択的に発生する、第 1 および第 2 メモリ・アクセス制御ユニットとを備え、

20

前記第 1 メモリ・アクセス制御ユニットが、前記少なくとも 1 つの第 1 整合信号を受け取った場合、前記第 1 ローカル・イベント信号および前記第 1 メモリ・アクセス・イベント信号が、前記第 1 基準アドレスおよび前記アドレス入力間における完全な重複または部分的重複に選択的に応じ、

前記第 2 メモリ・アクセス制御ユニットが、前記少なくとも 1 つの第 2 整合信号を受け取った場合、前記第 2 ローカル・イベント信号および前記第 2 メモリ・アクセス・イベント信号が、前記第 2 基準アドレスおよび前記アドレス入力間における完全な重複または部分的重複に選択的に応ずることを特徴とするメモリ・アクセス・アドレス比較器。

30

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、特に高度集積デジタル信号処理システムのためのエミュレーション・ハードウェアに関し、特にデータ一致認定(クオリフィケーション)および完全または部分的重複(オーバーラップ)によって、多数のサイズのメモリ・アクセスを検出するアドレス範囲比較器に関する。

【背景技術】

40

【0002】

高度なウエハ・リソグラフィおよび表面実装パッケージング技術は、電子的設計のシリコン・レベルおよびプリント回路基板レベル双方において、増々複雑な機能を統合しつつある。しかしながら、設計密度が高まり、相互接続ピッチが狭くなったことの結果、検査やエミュレーションのために回路に物理的にアクセスし難くなった。完成した製品を制御および観察しつつ検査およびデバッグすることができるようにするためには、設計中にも検査可能である(designed-in testability)必要がある。あらゆる製造上の欠陥は、製品を出荷する前の最終検査の間に検出できることが好ましい。この基本的な要望に応ずることは、論理設計フェーズにおいて検査を可能にし、自動検査機器が製品を検査できるように考慮しなければ、複雑な設計では達成が困難である。

50

【0003】

機能性および製造上の欠陥についての検査に加えて、アプリケーション・ソフトウェアの開発では、システムまたはサブシステムの設計フェーズにおいても同様のレベルのシミュレーション、観察可能性および制御可能性が必要となる。設計のエミュレーション・フェーズは、最終機器またはアプリケーションがシステム・ソフトウェアとリンクされたときに、1つ以上のIC（集積回路）のシステムがこれらの中で正しく機能することを保証しなければならない。自動車産業、電気通信、防衛システム、生活支援システムにおいてICの使用が増大するに連れて、総合的な検査や広範なリアル・タイム・デバッグは、欠くことができない必要事項となっている。

【0004】

機能的検査では、設計者が検査ベクトルを発生して仕様に一致することを確認するが、この検査方法は今でも広く用いられ続けている。超大型システムでは、この方法は、高いレベルでの障害検出に用いるには不相当であることがわかっている。完全な検査可能性、制御可能性および観察可能性を得るには、自動的に発生する検査パターンが望ましい。これらは、システム・レベルからトランジスタ・レベルまでの検査階層全体に及ぶ主要な目標である。

【0005】

大規模な設計における別の問題は、検査のために長時間およびかなりの出費が設計にかかることである。再利用可能設計（design-for-reusability）の概念と調和する検査性回路、システムおよび方法を有することができれば望ましいであろう。このようにすれば、初期デバイスに実装される検査性、シミュレーションおよびエミュレーション回路、システムならびに方法を再利用することによって、その後のデバイスおよびシステムの検査性、シミュレーションおよびエミュレーションのための限界設計コストを低く抑えることができる。検査性、シミュレーションおよびエミュレーションに先取的な計画性がなければ、後に検査パターンの作成およびアップグレードを行う際に、設計時間の著しい延長を招くことになる。

【0006】

再利用可能なモジュールを設計し、その検査パターンを完全に作成し評価するために多大な投資を行っても、後にモジュールを使用する際に、これを特定用途ロジック内に埋め込んでしまうことがある。こうなると、モジュールにアクセスするのは困難または不可能となる。したがって、この落とし穴を回避することが望ましい。

【0007】

IC設計の進展に伴って、内部可視性および制御性の低下、障害対応可能範囲（fault coverage）の減少、状態を変化させる機能の減少、検査の展開および検証に関する問題の増大、設計シミュレーションの複雑化、CAD（コンピュータ支援設計）ツールの絶え間ないコスト上昇を招いている。基板の設計では、レジスタの可視性および制御性が低下し、設計検証におけるデバッグおよびシミュレーションが複雑化し、多くの回路を1つのパッケージに封入したために物理的なアクセスが不可能になったことによって従来のエミュレーションができなくなり、基板上の配線の複雑さが高まり、設計ツール、異種素子同士の封入（mixed-mode packaging）、および生産性のための設計コストが上昇するという副作用がある。アプリケーションの設計では、状態の可視性が低下し、高速エミュレーションが難しくなり、時間シミュレーションの調整（scaled time simulation）が必要となり、デバッグ処理の複雑さが高まり、エミュレータのコストが上昇するといった副作用があげられる。生産における副作用には、可視性および制御性の低下、検査ベクトルおよびモデルの複雑化、検査の複雑化、異種素子同士の封入増加、自動検査機器のコスト上昇し、許容度の厳格化等があげられる。

【0008】

走査に基づくエミュレーションおよびマルチプロセッシング・デバッグを利用したエミュレーション技術が導入されたのは、10年以上も前のことである。1988年、設計サイクル時間に対する圧力、およびオンチップ・エミュレーションに新たに利用可能な空間

10

20

30

40

50

が動機となって、従来の回路内エミュレーションから走査に基づくエミュレーションに推移した。設計サイクル時間に対する圧力は、3つの要因によって生じた。オンチップ・メモリの使用増大等による集積度の上昇によって、より多くの設計時間が必要となった。クロック・レートの上昇とは、エミュレーション支援ロジックのために電氣的な進入が増大したことを意味する。封入処理の一層の精巧化のために、エミュレータの接続性の問題が生じた。今日、これらと同じ要因が、新たな展開を伴って、今日の複雑で、クロック・レートが一層高く、集積度が高い設計に必要なシステム・デバッグ機構を、走査に基づくエミュレータが達成できるための課題となっている。結果的に、システムは、小型化、高速化、および低コスト化する。これらの性能は向上し、フットプリントは増々高密度化している。これら実践的なシステムの傾向の各々が、システム・アクティビティ (system activity) の観察、即ち、迅速なシステム開発を可能にするための秘訣に悪影響を及ぼす。この影響を「可視性の消失 (vanishing visibility)」と呼ぶ。

【0009】

図1は、経時的な可視性および制御、ならびにシステムの高集積化の傾向を示す。アプリケーション開発者は、図1に示す最適可視性レベルを好む。この最適可視性レベルでは、関係するシステム・アクティビティ全ての可視性および制御が得られる。集積度の着実な進歩およびクロック・レートの上昇のために、時の経過と共に実際の可視性および制御の可用性が徐々に低下しつつある。これらの威力によって、可視性および制御のギャップ、最適な可視性および制御レベルと実際に使用可能なレベルとの間の差が生じる。時が経つにつれて、このギャップは広がっていく。アプリケーション開発メーカはギャップ生長レートを最小にすべく努力している。開発ツール・ソフトウェアおよび共に用いるハードウェア構成要素は、より少ないリソースで、しかも多種多様な方法で、より多くのことを行わなければならない。この使いやすさという課題に対する取り組みは、これらの影響力によって一層強化されている。

【0010】

今日の高集積度システム・オン・チップ (SOC: System-On-a-Chip) 技術では、可視性および制御のギャップは、時間の経過と共に劇的に拡大しつつある。ロジック・アナライザや組み込み若しくは区分 (partitioned) プロトタイプ・システムのような従来からのデバッグ選択肢は、今日のシステムの集積レベルや、増々高まりつつあるクロック・レートに歩調を合わせることはできない。集積度の上昇に伴って、多数のサブシステム構成要素を接続するシステム・バスがチップ上に移動するため、従来のロジック・アナライザはこれらのバスにアクセスできなくなる。バスの可視性の制約または事実上の喪失により、ロジック・アナライザのようなツールを用いても、開発中のシステムを制御するために必要なシステムのアクティビティを視認したり、トリガ機構を設けることができなくなった。この可視性の喪失に伴って、制御が不能となる。何故なら、アクセスできないものを制御することは困難であるからである。

【0011】

この傾向に対処するために、システム設計者はこれらのバスを表出させるように努めた。このため、システム構成要素は、表出したバスを用いて試作システムの構成が可能となるように組み立てられた。この手法も、システム・クロック・レートの絶え間ない上昇歩調によって阻害される。中央演算装置 (CPU) のクロック・レートが上昇すると、チップ間のインターフェースの速度はそれについて行けなくなる。チップ間通信レートの遅れを補償するためにインターフェース待機状態が追加されることにより、区分システムの性能は、その集積されている相手と歩調を合わすことができないことを、開発者は突き止めた。ある時点では、この性能低下は許容できないレベルにまで達し、区分プロトタイプ・システムはもはや実用的なデバッグ選択肢ではなくなった。現在では、生産機器は、アプリケーションの開発のためのプラットフォームとして機能しなければならない。

【0012】

また、CPUクロック・レートの上昇により、他の単純な可視性機構の可用性も制限される。CPUクロック・レートは最大I/O状態レートを上回ることができるので、ネー

10

20

30

40

50

ティブな形式で (native form) で情報をエクスポートする可視性ポートはもはやCPUについて行くことはできない。また、オンチップ・サブシステムも、CPUクロック・レートよりは遅いクロック・レートで動作する。この手法は、システムの設計を簡略化し、電力消費を削減するためには用いることができる。これらの開発は、単純な可視性ポートは、もはや、CPUアクティビティの正しい状態若しくは明確な表示 (clear view) を送り出すためには、あてにできないことを意味する。可視性および制御性が低下するに連れて、アプリケーションを開発するために用いられる開発ツールの生産性が低下する。また、これらのツールは、可視性および制御性を維持するために必要なツールの複雑化のために、増々使用が困難になると思われる。システム・オン・チップによって生じた可視性、制御、および使いやすさの問題のために、製品開発サイクルが長引くことが多くなっている。 10

【0013】

集積化の傾向が開発者には厳しいデバッグ環境を強いるとしても、これらは、問題をデバッグする新たな手法が出現するという希望も与える。高密度化やクロック・レートの上昇は、開発サイクル時間に対する重圧となるが、これらを解決する機会も生み出す。オンチップ・デバッグ機能は、これまで以上に手頃な価格となった。超大型メモリ構造が増々高速高性能チップ上で拡大するに連れて、CPUおよびメモリ・サブシステムに伴うランダム・ロジックに係るシステム・コストは、システム・コスト全体の割合としては低下しつつある。数千個のゲートによるコスト上昇分は、これまでになく低くなっている。このサイズの回路は、場合によっては、今日のチップ設計の曲がり角に追いやられる場合もある。今日の高密度パッケージにおけるピン当たりのコスト上昇分も低下している。このため、より多くのピンをデバッグのために割り当てることが容易となった。手頃な価格のゲートおよびピンを共に用いることによって、システム・オン・チップによってもたらされた課題に取り組むために必要な、新たなオンチップ・エミュレーション機構 (ファシリティ) の配備が可能となる。 20

【0014】

また、生産機器がアプリケーション・デバッグ・プラットフォームとしても機能する場合、これらは市場の目標に合わせて時間を捻出するのに十分なデバッグ機能 (ケイパビリティ) を備えていなければならない。デバッグの必要性は個々のアプリケーションで異なるので、オンチップ・デバッグ機構を調節し、市場およびコストの要望に対して時間のバランスが取れるようにすることが望ましい。これらオンチップ機能はチップの循環コストに影響を及ぼすので、いずれの解決策にとっても規模変更可能性若しくはケスラビリティ (scalability) が最も重要である。「必要なものにだけ支払う」ことは、オンチップ・ツールの配備にとっても処理原則であるはずである。この新たな枠組みにおいて、システム構築者は、チップ・コストの制約や製品開発チームのデバッグについての要望のバランスを取りながら、オンチップ・デバッグ機構を、それ以外の機能性と共に指定することができる。 30

【0015】

図2は、4つのエミュレータ構成要素を含むエミュレータ・システム100を示す。これら4つの構成要素は、デバッガ・アプリケーション・プログラム110、ホスト・コンピュータ120、エミュレーション・コントローラ130、およびオンチップ・デバッグ機構 (ファシリティ) 140である。図2は、これらの構成要素の接続を示す。ホスト・コンピュータ120は、ホスト120外部のエミュレーション・コントローラ130に接続されている。また、エミュレーション・コントローラ130はターゲット・システム140にも接続されている。ユーザは、デバッガ・アプリケーション・プログラム110を通じて、ターゲット・システム140上でターゲット・アプリケーションを制御することが好ましい。 40

【0016】

ホスト・コンピュータ120は通常パーソナル・コンピュータである。ホスト・コンピュータ120は、エミュレータ・コントローラ130を通じてデバッグ機能にアクセスす 50

る。デバッガ・アプリケーション・プログラム 110 は、ホスト・コンピュータ 120 を通じて、ユーザに易しい形態でデバッグ機能を提供する。デバッグ・リソースは、必要に応じて、デバッグ・アプリケーション・プログラム 110 によって割り当てられ、このためのユーザの負担を軽減する。ソース・レベルのデバッグは、デバッグ・リソースを利用し、それらの複雑性をユーザには隠している。デバッガ・アプリケーション・プログラム 110 は、オンチップ・トレースおよびトリガ機構と共に、対象のチップ・アクティビティを選択し、記録し、表示する手段を設ける。トレース表示は、トレース・ログを生成したソース・コードと自動的に相関付けられる。エミュレータは、デバッグ制御およびトレース記録機能双方を備えている。

【0017】

デバッグ機構は、JTAG または同様のシリアル・デバッグ・インターフェースを介した標準的なエミュレータ・デバッグ・アクセスを用いてプログラムすることが好ましい。ピンは貴重なので、本発明の好適な実施形態では、デバッグ・ピンの集合を、トレース、トリガ、およびその他のデバッグ機能で共有し、シリコン・コストにおける増加分を少なく済ませるようにした。固定したピン・フォーマットにも対応可能である。ピン共有選択肢を採用する (deploy) 場合、デバッグ・ピンの利用は、各デバッグ・セッションの開始時に決定され、その後ターゲット・システム 140 にアプリケーション・プログラムを実行するように指令する。これによって、トレース・エクスポート帯域幅 (trace export bandwidth) を最大化する。トレース帯域幅を最大化するには、最大数のピンをトレースに割り当てる。

【0018】

システム内部のデバッグ機能および構築 (ビルディング) ブロックは、可変とすることもできる。したがって、デバッガ・アプリケーション・プログラム 100 は、実行時 (ランタイム) にコンフィギュレーションを確定する。この手法では、ハードウェア・ブロックが、コンフィギュレーションおよびレジスタ編成に関する一連の制約を満たす必要がある。他の構成要素は、システム・メモリ・マップにおいてブロックやその他のペリフェラル (peripheral) を突き止め位置づけるように設計されたハードウェア検索機能を設ける。デバッガ・アプリケーション・プログラム 110 は、検索機構を用いてリソースを突き止める。モジュールが位置するアドレスおよびタイプ ID が、発見された各ブロックを一意に特定する。一旦 ID が発見されたなら、設計データベースを用いれば、正確なコンフィギュレーションならびに全てのシステム入力および出力を確認することができる。

【0019】

ホスト・コンピュータ 120 は、一般に、少なくとも 64 M バイトのメモリを含み、Windows (登録商標) 95、SR-2、Windows (登録商標) NT、またはそれ以降のバージョンの Windows (登録商標) を走らせることができる。ホスト・コンピュータ 120 は、エミュレータが必要とする通信インターフェースの 1 つに対応していなければならない。これらは、イーサネット (登録商標) 10T および 100T、TCP/IP プロトコル、ユニバーサル・シリアル・バス (USB)、ファイアワイヤ IEEE 1394、ならびに SPP、EPP および ECP のようなパラレル・ポートを含むことができる。

【0020】

ホスト・コンピュータ 120 は、リアル・タイムのデータ交換帯域幅を決定するにあたって主要な役割を果たす。まず、ホスト・エミュレータ間通信は、最大維持リアル・タイム・データ交換帯域幅を規定する際に主要な役割を果たす。何故なら、エミュレータ・コントローラ 130 はその受信リアル・タイム・データ交換バッファが満杯になったら直ちにこれらを空にしなければならないからである。第 2 に、リアル・タイム・データの発信または受信を行うホスト・コンピュータ 120 は、受信したリアル・タイム・データ交換データの準備および送信または処理および格納を維持するための十分な処理能力またはディスク帯域幅を有していなければならない。技術的現状では、ファイアワイヤ (Firewire) 通信チャネル (IEEE 1394) を備えたパーソナル・コンピュータが、最大のリア

10

20

30

40

50

ル・タイム・データ交換帯域幅を得るには好ましい。この帯域幅は、他の通信選択肢よりも性能を10倍高めることができる。

【0021】

エミュレーション・コントローラ130は、ホスト・コンピュータ120とターゲット・システム140との間にブリッジを設ける。エミュレーション・コントローラ130は、ホスト・コンピュータ120上で実行するデバッグ・アプリケーション・プログラム110とターゲット・システム140上で実行するターゲット・アプリケーションとの間で受け渡されるあらゆるデバッグ情報を処理する。現時点において好適な最小のエミュレータ・コンフィギュレーションは、リアル・タイム・エミュレーション、リアル・タイム・データ交換、トレース、および高度分析といった機能の全てに対応する。

10

【0022】

エミュレーション・コントローラ130は、好ましくは、実行制御、メモリ、レジスタ・アクセスのようなリアル・タイム・エミュレーション機能に、3、4、または5ビットの走査型インターフェース(scan based interface)を通じてアクセスする。リアル・タイム・データ交換機能は、走査によって、または走査以外の直接的なターゲット・エミュレータ間接続を用いる3つの上位帯域幅リアル・タイム・データ交換フォーマットを用いることによってアクセスすることができる。入力および出力トリガによって、他のシステム構成要素がチップにデバッグ・イベントを知らせることや、その逆が可能となる。ビットI/Oによって、エミュレータはシステム入力および出力を刺激すること、または監視することが可能となる。ビットI/Oは、工場検査およびその他の時間的に厳しくない(non-time-critical)低帯域幅のエミュレータ/ターゲット動作に対応するために用いることができる。拡張動作モードは、デバイス検査およびエミュレーション動作モードを指定するために用いられる。エミュレータ・コントローラ130は、通信部およびエミュレーション部に区分される。通信部は、ホスト通信リンクを支援し、一方エミュレーション部はターゲットにインターフェースして、ターゲット・デバッグ機能およびデバイス・デバッグ・ポートを管理する。エミュレーション・コントローラ130は、先にこの中で概説した業界標準の通信リンクの1つを用いて、ホスト・コンピュータ120と交信する。ホスト・エミュレータ間接続は、汎用の配線技術を用いて行われる。ホスト・エミュレータ間の分離は、用いるインターフェースに適用される規格によって管理される。

20

【0023】

エミュレーション・コントローラ130は、1本または複数のターゲット・ケーブルを通じてターゲット・システム140と交信する。デバッグ、トレース、トリガ、およびリアル・タイム・データ交換機能は、ターゲット・ケーブルを共有し、場合によっては、同じデバイス・ピンを共有する。ターゲット・システム140が、1本のケーブルには収容しきれないトレース幅を配備する場合、1本よりも多いターゲット・ケーブルが必要となる場合もある。トレース、リアル・タイム・データ交換、およびデバッグ通信は全て、このリンクを通じて行われる。エミュレータ・コントローラ130は、少なくとも2フィート(約60cm)のターゲット・エミュレータ間分離を設けることが好ましい。このエミュレーション技術は、50MHzまでの検査クロック・レート、および200から300MHzまたはこれ以上のトレース・クロック・レートが可能である。エミュレータの設計がターゲット・システム140の制約を緩和する技法を用いても、これらのレートでエミュレータ・コントローラ130およびターゲット・システム140間で通知を行うには、設計に努力が必要である。このエミュレーション技術は、チップのデバッグ・ピンの配置、基板のレイアウトに制約を賦課する場合があります。正確なピンのタイミングを必要とする。タイミングの制約を満たすのを補助するために、オンチップ・ピン・マクロを設ける。

30

40

【0024】

オンチップ・デバッグ機能は、開発者に、豊富な開発機能を2段階の拡張可能な手法で提供する。第1段階では、CPUのメガ・モジュールに組み込んだリアル・タイム・エミュレーション機能を利用した機能性を備える。このリアル・タイム・エミュレーション機能は、一定の機能性を有し、永続的にCPUの一部であるが、殆どの場合、コアの外側に

50

、高性能リアル・タイム・データ交換、高度分析およびトレース機能が追加される。これらの機能を個々に選択し、チップに加入する。エミュレーション・ペリフェラルをシステム設計に加入することにより、第2段階の機能性を形成する。コスト効率的なエミュレーション・ペリフェラルのライブラリは、システムを作成するための構築ブロックを含み、高度分析、高性能リアル・タイム・データ交換、およびトレース機能の構造化を可能にする。好適な実施形態では、5つの標準デバッグ・コンフィギュレーションが提供されるが、カスタム・コンフィギュレーションにも対応する。具体的なコンフィギュレーションについては、この中で後に扱う。

【発明の開示】

【発明が解決しようとする課題】

10

【0025】

本発明は、2つの個別の比較器として動作可能なエミュレーション比較器であり、各々、プロセッサが発行するプロセッサ・メモリ・アクセス・アドレス(PMAA)およびプロセッサ・メモリ・アクセス・アドレス(PMAD)値をレジスタに格納されている基準値と比較する。あるいは、これは単一のデバイスとして動作することができ、メモリ・アクセス・アドレスを、二重境界範囲(double bounded range)を形成する2組の基準値と比較する。メモリ・アクセスのデータ内容は、アドレス比較に対する追加のクオリファイア(qualifier)として任意に用いることもできる。

【課題を解決するための手段】

【0026】

20

比較器の実施に必要な具体的なリソースは、所与の実施態様の要件および制約に依存して大きく変化する可能性がある。本願において記載する例では、範囲比較器は、2つの単一基準アドレス比較器と、対応するデータ比較器とを含み、2つのアドレスおよび2つのデータ比較論理機能に加えて、13のユーザ可視レジスタ(user-visible register)を必要とする。

【0027】

2つの比較器の各々は、制御レジスタによって行われる選択に基づいて、独立した結果または依存した結果を得ることができる。2つのデータ比較器の結果は、関連するアドレス比較器に対するクオリファイアとして用いることができ、あるいは制御レジスタのコンフィギュレーションによっては無視することもできる。

30

【0028】

プロセッサ・メモリ・アクセス・アドレスは、バイト、半ワード、ワード、二重(ダブル)ワードのように、プロセッサのアーキテクチャに応じて、多数のサイズとすることができる。ここに記載する実施態様における比較器は、基準を多数のサイズのメモリ・アクセスと比較することができる。

【0029】

データ比較認定(qualification)を別として、比較器は、6つのアドレス比較判断基準の1つに基づいて一致を示すことができる。プロセッサ・メモリ・アクセス・アドレスは、基準と等しい場合、基準と等しくない場合、基準よりも大きい場合、基準以上の場合、基準未満の場合、または基準以下の場合があり得る。

40

【0030】

本発明は、完全一致または部分的一致の判断基準が可能である。これらは、均等一致(equality match)を求める場合には、2つのモードで具体化される。第1モードでは、プロセッサ・メモリ・アクセスが基準における全バイトと一致(オーバーラップ)した場合にのみ、一致が生ずる。第2モードでは、プロセッサ・メモリ・アクセスの少なくとも1バイトが基準と一致した場合に、一致が生ずる。

【0031】

本願において説明する例で用いられるデータ比較器は、同一性均等比較器(identity equality comparator)である。データ比較器が一致を示すのは、重複判断基準によって認定された基準における対応するビットと一致するデータにおけるビットが全て等しい場合

50

のみである。データ比較器は、基準マスク値を含む。この例の64ビット・メモリ・データ・バスの場合、2つの32ビット・レジスタが、2つのデータ比較器の各々に必要となる。マスクによって、一致に必要なデータ・ビットのカスタム化が可能になる。したがって、必ずしもビットの全てにおいて一致が必要な訳ではない。

【0032】

比較器は、プロセッサのメモリ・マップへのインターフェースを通じてプログラムすることができる。本願では、このインターフェースをコンフィギュレーション・バス(cfb)と呼ぶ。

【0033】

比較器の機能は、ハードウェア・デバッグ・システムが用いるツールとして利用することができ、比較器の結果を入力として取り込み、デバッグ・システムのコンフィギュレーションに応じて、広い範囲の応答を与える。

10

本発明のこれらおよびその他の態様を図面に示す。

【発明を実施するための最良の形態】

【0034】

メモリ内の特定の位置に対するプロセッサのメモリ・アクセスを検出することができる、プロセッサ・コードの開発者がソフトウェアの問題を解決する(troubleshooting)する際に役立つので、これは必要である。本発明の比較器は、多数の特徴と高い柔軟性を有し、ソフトウェア・デバッグの作業を支援する強力なツールである。デバッグ処理には無関係な他の潜在的な用途には、プロセッサ動作の関連では、システム・イベントの相互作用または変化を開始させるために行う、プロセッサのメモリ・アクセスの検出を含むことができる。

20

【0035】

本発明の比較器は、特定のメモリ・アドレスまたはアドレス範囲に対するプロセッサのメモリ・アクセスの有無を検出することができ、更にプロセッサのメモリ・アクセスに伴うデータによって、特定のメモリ・アドレスまたはアドレス範囲も認定することができる。ハードウェア・デバッグ・システムに比較器を一体化すると、プロセッサが実行するコードにおける問題を検出および解決するシステムの機能を高めることができる。

【0036】

本発明の比較器は、デバッグ・プロセスがプロセッサのメモリ・アクセスの検出に基づく場合のソフトウェア問題の検出において、機能を強化し、柔軟性を高める。比較器において完全な重複または部分的重複の検出判断基準を使用可能であることは、あらゆるバイト境界におけるメモリ位置にアクセスを要求する可能性があるプロセッサ・アーキテクチャのために実施する場合には、特に重要である。この機能がないと、かかるアーキテクチャにおけるプロセッサのメモリ・アクセス・イベントの適正な検出には重大な制約が生じ、多くの場合不可能になることさえある。

30

【0037】

図3は、ターゲット・システム140を用いたオンチップ・デバッグ・アーキテクチャの一例を示す。このアーキテクチャは、数個のモジュール・クラスを用いてデバッグ機能を構成する。これらのクラスの1つに、バス・イベント検出器210、補助イベント検出器211、およびカウンタ/ステートマシン213を含むイベント検出器がある。第2のクラスのモジュールは、トリガ・ビルダ(trigger builder)220を含むトリガ発生器である。第3のクラスのモジュールは、トレース収集230およびフォーマット化を含むデータ取得である。第4のクラスのモジュールは、トレース・エクスポート240、リアル・タイム・データ交換エクスポート241を含むデータ・エクスポートである。トレース・エクスポート240は、局部発振器245からのクロック信号によって制御される。局部発振器245については、後に詳しく説明する。最後のクラスのモジュールは、走査入力/出力をCPUコア201にインターフェースする走査アダプタ250である。最終的なデータのフォーマット化およびピンの選択は、ピン・マネージャおよびピン・マクロ260において行われる。

40

50

【0038】

システム・オン・チップ (system-on-chip) の個々の実施形態のいずれに対しても、デバッグ機能のサイズおよびそれに付随する処理能力は、完全な機能を削除するか、または配備するイベント検出器およびトリガ・ビルダの数を制限することによって調節することができる。加えて、トレース機能は、プログラム・カウンタ・トレースのみからプログラム・カウンタおよびデータならびにASICおよびCPU発生データのトレースまで徐々に増大させることができる。また、リアル・タイム・データ交換機能を任意に配備することもできる。オンチップ・ツールをカスタム化できるので、アプリケーション開発の枠組みを変化させることができる。過去においては、所与のCPUコアを用いたチップ設計は全て、固定した1組のデバッグ機能に限定されていた。現在では、チップの設計毎に最適化されたデバッグ機能が得られる。この枠組みの変化によって、システム・アーキテクトには、製品開発リスクを管理するために必要なツールが手頃なコストで与えられることになった。尚、同じCPUコアであっても、異なるピン出力を有する異なるペリフェラルと共に用いることによって、異なるシステム・オン・チップ製品を具体化できることを注記しておく。これらの異なる実施形態は、異なるデバッグおよびエミュレーション・リソースを必要とする場合もあり得る。本発明のモジュール性により、かかる実施形態の各々は、個々のシステム・オン・チップに応用するために必要なデバッグおよびエミュレーション・リソースのみを含むだけで済む。

10

【0039】

リアル・タイム・エミュレーション・デバッグの基礎的な構成要素は、アプリケーションの開発に関係する基本的なデバッグおよび計測管理 (instrumentation) に取り組むために用いられる。これは、あらゆる実行制御およびレジスタ可視性機能、ならびにブレークポイントやウォッチポイント機能のような最小のリアル・タイム・データ交換および分析機能を内蔵している。これらのデバッグ動作は、オンチップ・ハードウェア機構を用いて、アプリケーションの実行を制御し、レジスタやメモリへのアクセスを得る。リアル・タイム・エミュレーションが対応可能なデバッグ動作の一部は、ソフトウェア・ブレークポイントの設定、そのポイントにおける機械状態の観察、命令毎に正確な決定を観察するための1ステップ・コード前進、既知のメモリ位置への疑似書き込みの検出、およびメモリおよび周辺レジスタの視認および変更である。

20

【0040】

リアル・タイム・エミュレーション機構は、CPUメガ・モジュール (CPU mega-module) 内に組み込まれ、CPUコア201のファブリック (fabric) に編入される (weave)。これによって、CPUコア201を用いた設計が、デバッガ・アプリケーション・プログラム110のベースライン・デバッグ、測定管理、およびデータ転送機能に対応するだけの十分なデバッグ機構を有することに確証を与える。各CPUコア201は、基礎的な1組のエミュレーション機能を組み込んでいる。これらの機能には、実行 (run)、1命令ステップ、休止および自由実行 (halt and free run) のような実行制御、レジスタやメモリの表示および修正、ソフトウェアおよび最小限のハードウェア・プログラム・ブレークポイントを含むブレークポイント、ならびに最小限のハードウェア・データ・ブレークポイントを含むウォッチポイントを含むが、これらに限定されるのではない。

30

40

【0041】

図4は、範囲比較器のアドレス構成要素を一体化した、2つの個別のアドレス比較器を含む比較器400を示す。比較器400は、バス・イベント検出器210の一部であることが好ましい。比較器400の柔軟な組み合わせにより、多種類のバス・イベント検出が可能となる。第1比較器は図4の上側にあり、第2比較器は下側にある。各アドレス比較器は、4つの主要構成要素、即ち、プロセッサ・メモリ・アクセス・サンプリング機構 (マルチプレクサ411、412、421および422)、コンフィギュレーション・リソース (バス選択レジスタ402、比較器制御レジスタ403および405、アドレス基準レジスタ404および406)、プログラム・アクセス・アドレス大小比較器 (メモリ・アドレス大小比較器413および423)、およびメモリ・アクセス比較制御ブロック (

50

4 1 5 および 4 2 5) を有する。

【 0 0 4 2 】

図 4 は、プロセッサ・メモリ・アクセス・サンプリング機構として機能する、4 つのマルチプレクサ 4 1 1、4 1 2、4 2 1 および 4 2 2 を示す。図 4 は、2 系統のプログラム・メモリ・アクセス・アドレス・バスを示す。マルチプレクサ 4 1 1 および 4 2 1 は、使用可能なデータ・アドレス・バス `dmem_addr_0` および `dmem_addr_1` の中から所望のアドレス・バス信号を選択する。マルチプレクサ 4 1 2 および 4 2 2 は、選択したデータ・アドレス・バスに付随する、対応の制御バス信号 `mem_acc_ctl_0` および `mem_acc_ctl_1` を選択する。ユーザは、バス選択レジスタ 4 0 2 に書き込むことによって、2 つの比較器に対するバス選択を制御する。

10

【 0 0 4 3 】

図 4 は、比較器の構成の一部である 5 つのレジスタを示す。バス選択レジスタ 4 0 2、比較器制御レジスタ 4 0 3 および 4 0 5、ならびにアドレス基準レジスタ 4 0 4 および 4 0 6 は、コンフィギュレーション・バス・インターフェース (`cfgb`) と呼ぶメモリ・インターフェースを通じて、他のコンフィギュレーション変更可能なハードウェア・リソースと同様に、アクセスすることができる。このメモリ・インターフェースは、コンフィギュレーション制御信号 (`cfgb_ctl`) を受け取るコンフィギュレーション・インターフェース制御部 4 0 1 を含む。コンフィギュレーション・データ (`cfgb_data`) およびコンフィギュレーション・クロック (`cfgb_clk`) が、バス選択レジスタ 4 0 2、比較器制御レジスタ 4 0 3、4 0 5、およびアドレス基準レジスタ 4 0 4、4 0 6 に供給される。レジスタ書き込み要求の受信および `cfbg_ctl` を通じたレジスタの識別時に、コンフィギュレーション・インターフェース制御部 4 0 1 は、コンフィギュレーション・データ・バス上のデータを格納するレジスタを選択する。選択されたレジスタは、次のコンフィギュレーション・クロック信号の時点で、コンフィギュレーション・データ・バス上のデータを格納する。

20

【 0 0 4 4 】

バス選択レジスタ 4 0 2 に格納されたデータは、マルチプレクサ 4 1 1、4 1 2、4 2 1 および 4 2 2 を通じて、それぞれの基準値と比較される入力バスを規定する。マルチプレクサ 4 1 1 は、メモリ・アドレス大小比較器 4 1 3 およびメモリ・アドレス・バイト整合 (`alignment`) マップ作成部 (`maker`) (`BAMM`) 4 1 4 に供給するために、第 1 バス `dmem_addr_0` または第 2 バス `dmem_addr_1` のいずれかを選択する。マルチプレクサ 4 1 2 は、メモリ・アドレス大小比較器 4 1 3 およびメモリ・アドレス・バイト整合マップ作成部 4 1 4 に接続するために、第 1 制御バス `mem_acc_ctl_0` または第 2 制御バス `mem_acc_ctl_1` のいずれかを選択する。以下で説明するが、最終的な一致の判定は、それぞれの制御バス信号において示されるメモリ・アクセスのデータ・サイズに左右される可能性もある。マルチプレクサ 4 1 1 および 4 1 2 の選択は、`port_select_0` 信号を通じてバス選択レジスタ 4 0 2 によって制御される。アドレス基準レジスタ 4 0 4 は、比較のために、基準アドレスをメモリ・アドレス大小比較器 4 1 3 およびメモリ・アドレス・バイト整合マップ作成部 4 1 4 に供給する。同様に、マルチプレクサ 4 2 1 および 4 2 2 は、`port_select_1` 信号を通じたバス選択レジスタ 4 0 2 の制御の下で、アドレスおよび対応する制御バスを、メモリ・アドレス大小比較器 4 2 3 およびメモリ・アドレス・バイト整合マップ作成部 4 2 4 のために選択する。アドレス基準レジスタ 4 0 6 は、メモリ・アドレス大小比較器 4 2 3 およびメモリ・アドレス・バイト整合マップ作成部 4 2 4 のために基準アドレスを格納する。

30

40

【 0 0 4 5 】

メモリ・アドレス大小比較器 4 1 3 および 4 2 3 は、基準値と選択したプロセッサ・メモリ・アクセス・アドレスとの間で実際の比較を行う。メモリ・アドレス大小比較器 4 1 3 および 4 2 3 は、各々、プロセッサ・メモリ・アクセス・アドレスが基準アドレスと等しい場合には同等信号 (`EQ`) を発生し、プロセッサ・アクセス・アドレスが基準アドレスよりも大きい場合には超過信号 (`GT`) を発生し、プロセッサ・アクセス・アドレスが

50

基準アドレス未満の場合には未満信号 (LT) を発生する。また、メモリ・アドレス大小比較器 4 1 3 および 4 2 3 は、それぞれのオフセット信号 offset_0 および offset_1 も発生する。これらのオフセット信号は、対応するメモリ・アドレス・バイト整合マップ作成部 4 1 4 または 4 2 4 に供給される。メモリ・アドレス大小比較器 4 1 3 およびメモリ・アドレス・バイト整合マップ作成部 4 1 4 は、コンフィギュレーションおよび制御信号 config_&_control_0 を比較器制御レジスタ 4 0 3 から受け取る。メモリ・アドレス大小比較器 4 2 3 およびメモリ・アドレス・バイト整合マップ作成部 4 2 4 は、同様のコンフィギュレーションおよび制御信号 config_&_control_1 を比較器制御レジスタ 4 0 5 から受け取る。

【 0 0 4 6 】

10

オフセット信号 offset_0 および offset_1 は、プロセッサ・メモリ・アクセス・アドレスと基準アドレスとの間のオフセットをバイト単位で記述する。それぞれのメモリ・アドレス・バイト整合マップ作成部 4 1 4 および 4 2 4 は、2つの信号を発生する。第1の信号を基準バイト整合マップ (ref_bam) と呼ぶ。第2の信号をメモリ・アクセス・バイト整合マップ (macc_bam) と呼ぶ。これらの信号は、基準アドレスとプログラム・メモリ・アクセス・アドレスとの間の完全な重複または部分的な重複を検出する際の基礎となる。

【 0 0 4 7 】

メモリ・アドレス大小比較器 4 1 3 および 4 2 3 からの出力 GT、LT、EQ、ならびにメモリ・アドレス・バイト整合マップ作成部 4 1 4 および 4 2 4 からの基準バイト整合マップ (ref_bam) およびメモリ・アクセス・バイト整合マップ (macc_bam) は、各メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 に供給される。メモリ・アドレス・バイト整合マップ作成部 4 1 4 および 4 2 4 からのメモリ・アドレス・バイト整合マップ信号およびメモリ・アクセス・バイト整合マップは、各メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 に供給される。メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 は、これらの入力信号、各ORゲート 4 1 6 および 4 2 6 からの比較器認定 (クオリフィケーション)、ならびに比較器制御レジスタ 4 0 3 および 4 0 5 において定義されるコンフィギュレーション情報に応じて、対応する mem_access_event 出力を発生する。メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 は、対応するメモリ・アドレス大小比較器 4 1 3 および 4 2 3 からの3つの結果信号 (GT, LT, EQ) を取り込み、これらを6つの比較モード選択に拡張する。これらは、超過、未満、同等、不等、以下、および以上である。同時に、対応する比較器制御レジスタ 4 0 3 および 4 0 5 からのコンフィギュレーション信号に基づいてバイト整合マップ信号 ref_bam および macc_bam を比較し、重複一致結果を生成する。2つの比較器の各々は、6つの比較モデルの内の1つにプログラムすることができ、当該モデルの検出判断基準および重複一致結果を用いて、部分的比較器検出イベント (partial comparator detection event) を生成する。メモリ・アクセス比較制御ブロック 4 1 5 は、local_event_0 信号をメモリ・アクセス比較制御ブロック 4 2 5 に供給する。メモリ・アクセス比較制御ブロック 4 2 5 は、local_event_1 信号をメモリ・アクセス比較制御ブロック 4 1 5 に供給する。

20

30

【 0 0 4 8 】

各メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 は、それぞれの信号 look_up_table_0 および look_up_table_1 を通じて、対応する比較器制御レジスタ 4 0 3 および 4 0 5 から4エントリの参照テーブルを受け取る。テーブル参照信号によって、最終出力 mem_access_event_0 および mem_access_event_1 が、他のメモリ・アクセス比較制御ブロックからの local_event 信号に依存することが可能となる。この依存性により、各比較器が一方の限度値を検査することによって、アドレス範囲の比較が可能となる。中央演算装置のストール信号 (cpu_stall) がアクティブのとき、それぞれの比較器制御レジスタ 4 0 3 および 4 0 5 に格納されているコンフィギュレーション・データに応じて、メモリ・アクセス比較制御ブロック 4 1 5 および 4 2 5 のいずれかまたは双方を非動作、ディスエーブルすることができる。

40

【 0 0 4 9 】

50

メモリ・アクセスが正確に基準アドレスに等しくないことにユーザが関心を示す可能性はいくつかある。基準アドレスおよびバス・アドレスのそれぞれのデータ・サイズによっては、異なるアドレスを有するバス・アクセスは基準アドレスに触れる場合がある。例えば、データ・バイトの正確なアドレスを有していないと、ワードまたは半ワードの書き込みが、1バイトのデータを上書きしてしまう場合がある。通常、ユーザは、追跡しているメモリ・アクセスが基準に触れるか否かについてのみ関心があり、正確なデータ・サイズやバス・アドレスの組み合わせには関心がない。比較動作が正しく行われたか否か判定する規則は、単に、「対象とするバイトのいずれにも触れた場合、比較は正しく行われた」というように、単に述べることができる。このような一致は、アドレス・バス値およびデータ・サイズ双方に依存する。

10

【0050】

基準アドレスは、32ビット・アドレス境界に整合された一連のバイトと見なされ、基準アドレスのバイト・アドレスおよび基準アドレスに關与するサイズに基づいて配置される。基準アドレスに關与するサイズは、1バイト(バイト)、2バイト(半ワード)、または4バイト(全ワード)とすることができる。バス・アドレスは、同じように表される。比較器の一致が宣告されるのは、以下の条件のいずれもが真と評価されたときである。同等比較をイネーブルしたときに、バス・アドレスが基準アドレスのいずれのバイトを参照することによっても、同等一致が生じる。超過比較を指定したときに、基準アドレスが指定する最上位バイトよりも高いアドレスを有するいずれかのバイトを、バス・アドレスが参照することによって、超過一致が生じる。未満比較を指定したときに、基準アドレスが指定する最下位バイトよりも低いアドレスを有するいずれかのバイトを、バス・アドレスが参照することによって、未満一致が生じる。尚、これら3つの条件は同時に一致を生じ得ることを注記しておく。例えば、完全な1ワードのメモリ・アクセスが、1バイトとして指定された基準アドレスと重複する場合、この条件が生じる。これによって、いずれのサイズのアクセスであっても、指定した1組のバイトの上位、下位、またはその上に触れれば、一致を宣告することができる。

20

【0051】

表1は、いずれのバイト・アドレス可能な境界にて開始する全(フル)ワード、半ワードおよびバイト・アクセスに対応するメモリ・システムについて、非整合バス比較の事例を5つ示すものである。表1の中央にある3つの事例は、基準アドレスおよびバス・アドレスのサイズを条件とする。他の2つの事例は、アドレス・サイズの情報がなくとも、完全な比較を可能とする。

30

【表1】

32ビット・アドレス				
基準アドレス+1	基準アドレス+1	基準アドレス	基準アドレス-1	基準アドレス-1
バスアドレス >				
	バスアドレス =			
		バスアドレス =		
			バスアドレス =	
				バスアドレス <

40

表 1

【0052】

50

アクセスは、バス・アドレス[1:0]およびサイズ[1:0]データから作成した、1連の7つのバイト・イネーブルに分割される。これら7つのバイト・イネーブルによって、32ビット・ワードに対応するバイト・アドレスのいずれから開始する全ワード・アクセスでも指定し、このアクセスによる作用を受けるバイトの位置を規定することができる。メモリ・アクセス比較制御ブロック415および425は、それぞれのメモリ・アドレス大小比較器413および423からのGT、LT、EQ出力と、それぞれのメモリ・アドレス・バイト整合マップ作成部414および424からの基準バイト整合マップ(ref_bam)およびメモリ・アクセス・バイト整合マップ(macc_bam)とを受け取り、それぞれのバス及び基準データ・サイズに基づいて、要求されたアドレス比較を形成する。

【0053】

表2は、比較器制御レジスタ403に格納されている認定(クオリフィケーション)信号config_&_control_0およびlook_up_table_0のコード化の一例を示す。比較器制御レジスタ413に格納されているconfig_&_control_1およびlook_up_table_1も同様にコード化する。

10

【表2】

ビット	名称
11	EXACT
10:9	REF
8	EXE
7	STALL
6	ETYPE
5	確保
4	R/W
3:0	ACCESS

20

表 2

30

【0054】

EXACTビット(ビット11)は、アドレス比較の正確度を規定する。EXACT=0の場合、いずれのバイトが基準アドレスに触れても、一致が生ずる。EXACT=1の場合、正確なアドレス一致の場合にのみ、一致が生ずる。

【0055】

REFビット(ビット10および9)は、アドレスおよびデータ基準値のサイズを定義する。これを表3に示す。

40

【表 3】

REF[10:9]	基準サイズ
00	バイト
01	半ワード
10	ワード
11	ダブルワード

10

表 3

【0056】

EXEビット（ビット8）は、認定プログラム・カウンタ・イベントを定義する。EXE = 1の場合、ディスパッチ・ユニット（dispatch unit）からの1つの命令の実行を伴うプログラム・カウンタ・バス・イベント全ての認定が許可される。EXE = 0の場合、1つの命令発行を伴うプログラム・カウンタ・バス・イベントの認定はマスクされる。このマスクングによって、アドレスのみに基づいた単純な関数範囲（function range）を許可する（STALLをセットしている限り）。これは、プログラム・カウンタ・アドレス範囲にのみ有用である。

20

【0057】

STALLビット（ビット7）は、ストールがバス・イベントとして認定可能か否か制御する。STALL = 1の場合、アクティブなパイプライン・サイクルを伴うバス・イベントの認定はマスクされる。これによって、EXEビットが0である限り、ブロック範囲イベントをレベルとして許可する。これは、プログラム・カウンタ・アドレス範囲にのみ有用である。STALL = 0の場合、データ・アクセスおよびプログラム・カウンタ・アクセスにおけるストールを許可して、バス・イベントを認定する。

【0058】

ETYPEビット（ビット6）は、エミュレーション・メモリ・アドレスの検出を制御する。ETYPE = 0の場合、エミュレーション・アクセス・サイクルは検出されない。

30

R/Wビット（ビット4）は、読み出し動作または書き込み動作について、一致が生じたか否か判定を行う。R/W = 1の場合、比較器は読み出し動作について検出を行う。R/W = 0の場合、比較器は書き込み動作について検出を行う。

QLUフィールド（ビット3：0）は、データ・サイズに必要なバス・イベントの認定を定義する。表4は、これらのビットのコード化を示す。

【表 4】

ビット [3:0]	検出したアクセス・サイズ
0000	アクセス無し
0001	バイト
0010	半ワード
0011	半ワードおよびバイト
0100	ワード
0101	ワードおよびバイト
0110	ワードおよび半ワード
0111	ワード、半ワードおよびバイト
1000	ダブルワード
1001	ダブルワードおよびバイト
1010	ダブルワードおよび半ワード
1011	ダブルワード、半ワードおよびバイト
1100	ダブルワードおよびワード
1101	ダブルワード、ワードおよびバイト
1110	ダブルワード、ワードおよび半ワード
1111	ダブルワード、ワード、半ワードおよびバイト

表 4

【0059】

これらのビットは、バス検出ハードウェアに対して対象となるアクセス・サイズまたは複数のアクセス・サイズを判定する。参照テーブルは、1組のビットを、検出したアクセス・サイズにマップする。

【0060】

図5は、随伴するコンパニオン・データ比較器 (companion data comparator) を示す。好適な実施形態では、2つのデータ比較器500を用い、それぞれのORゲート416および426を介して、各メモリ・アクセス比較制御ブロック415および425に1つずつ接続されている。実施態様の多くでは、1つのみのデータ比較器500を用いて、両アドレス比較器で共有することも容認できる。こうすると、いくらか柔軟性が低下するという欠点がある。

【0061】

データ比較器500の好適な実施形態は、64ビット・データ・バス上で動作し、したがって32ビット・データ・バスの実施態様に要するよりも、2倍のリソースがデータ経路において必要となる。データ比較器500は、6つの論理セクションを含む。即ち、コンフィギュレーション・リソース (401、402、403、404、505、506および507)、サンプリング・ロジック (510、511、512、513、514)、メモリ・データ同一性比較器 (521および522)、メモリ・アクセス・データ・パイ

ト・アドレス範囲重複比較器(530)、およびメモリ・アクセス・データ比較制御部(540)である。

【0062】

コンフィギュレーション・リソースは、コンフィギュレーション・インターフェース制御部410および7つのレジスタを含む。バス選択レジスタ402およびインターフェース制御部401は、図4に示したアドレス比較器400と共有される。比較器制御レジスタ403およびアドレス基準レジスタ404は、対応するアドレス比較機能と共有される。64ビットの比較データは、2つの32ビット・ワードとして格納される。下位データ基準レジスタ(data_reference_reg_l)505は、最下位(least significant)32ビットを格納し、上位データ基準レジスタ(data_reference_reg_h)507は、最上位(most significant)32ビットを格納する。64ビットのマスク・データは、2つの32ビット・ワードとして格納される。下位マスク・レジスタ(mask_reg_l)506は、最下位32ビットを格納し、上位マスク・レジスタ(mask_reg_h)508は、最上位32ビットを格納する。マスク・レジスタ506および508におけるビット位置が0にセットされていると、一致と見なされ、1にセットされていると、比較が行われる。これらのレジスタには、プロセッサ内にある他のコンフィギュレーション変更可能なハードウェア・リソースと同様に、コンフィギュレーション・バスおよびコンフィギュレーション・インターフェース制御部401を通じてアクセスすることができる。

【0063】

サンプリング・ロジックは、メモリ・アクセスに関するデータ・バスおよび関連する制御およびアドレスの情報選択を定義する。マルチプレクサ511は、下位メモリ・データ・バイト同一性比較器521への供給のために、最下位32ビット・データ・バスmem_data_low_0またはmem_data_low_0のいずれかを選択する。マルチプレクサ512は、メモリ・データ・バイト同一性比較器522への供給のために、最上位32ビット・データ・バスmem_data_high_0またはmem_data_high_0のいずれかを選択する。選択は、バス選択レジスタ402からのport_select信号、ならびにそれぞれのデータ・バスに対応するメモリ・アクセス制御信号mem_acc_ctl_0およびmem_acc_ctl_1にตอบสนองして、データ・マルチプレクサおよび経路制御ロジック510によって制御される。ユーザは、バス選択レジスタ402に書き込むことによって、比較に用いるためのバス選択を制御する。

【0064】

下位メモリ・データ・バイト同一性比較器521は、下位データ基準レジスタ505からの基準データの最下位32ビット、下位マスク・レジスタ506からのマスク・データの最下位32ビット、およびマルチプレクサ511からのメモリ・データの選択された最下位32ビットを受け取る。同様に、上位メモリ・データ・バイト同一性比較器522は、上位基準レジスタ507からの基準データの最上位32ビット、上位マスク・レジスタ508からの最上位マスク・データの32ビット、およびマルチプレクサ512からのメモリ・データの選択された最上位32ビットを受け取る。メモリ・データ・バイト同一性比較器521および522は、対応する一致信号byte_data_match_0およびbyte_data_match_1を発生する。メモリ・データ・バイト同一性比較器521および522は、各々、対応するマスクを用いて、データ・バスの個々のバイト各々に対して同一性比較を行う。各バス一致信号は、対応する4バイトについて別個に一致を示す。これによって、多数のサイズに対する同時動作が可能となる。最小データ・サイズが1バイトであり、他のサイズが全てバイトの倍数であると仮定すると、データ比較器は、バイト(8ビット)、半ワード(16ビット)、ワード(32ビット)、およびダブルワード(64ビット)という4つのデータ・サイズのいずれについても、一致の指示を与えることができる。

【0065】

メモリ・アクセス・データ・バイト・アドレス範囲重複検出器530は、メモリ・アクセスのバイトと基準のバイトとの間の相対的な重複に関する情報address_byte_overlapを発生する。マルチプレクサ513は、データ・マルチプレクサおよび経路制御ロジック510からのport_select信号の制御の下で、一方のメモリ・アクセス制御信号mem_acc_ctl

_0またはmem_acc_ctl_1を選択する。同様に、マルチプレクサ514は、ポート選択信号の制御の下で、アドレス・バスdmem_addr_0およびdmem_addr_1の一方を選択する。メモリ・アクセス・データ・バイト・アドレス範囲重複検出器530は、マルチプレクサ513からの選択されたメモリ・バス制御信号、マルチプレクサ514からの選択されたアドレス・バス信号、アドレス基準レジスタ404からのアドレス、および比較器制御レジスタ403からのコンフィギュレーションおよび制御信号を受け取る。メモリ・アクセス・データ・バイト・アドレス範囲重複検出器530は、アドレス比較器400において用いられる重複計算と同様に、8つの重複信号(address_byte_overlap)を生成し、バイト整合マップを作成する。

【0066】

メモリ・アクセス・データ比較制御部540は、data_compare_result_0を発生する。メモリ・アクセス・データ比較制御部540は、8つの部分的データ比較結果、4つがメモリ・データ・バイト同一性比較器521からのbyte_data_match_1、そして4つがメモリ・データ・バイト同一性比較器522からのbyte_data_match_hと、メモリ・アクセス・データ・バイト・アドレス範囲重複検出器530から8つのaddress_byte_overlap信号と、メモリ・アクセスからの制御バス情報(mem_acc_ctl_0およびmem_acc_ctl_1)と、比較器制御レジスタ403からの参照テーブル(look up table)・データとを受け取る。メモリ・アクセス・データ比較制御部540は、これらの信号を用いて、データ一致信号(data_compare_result_0)を発生する。前述のように、このデータ一致信号は、アドレス比較を認定するために用いることができる。

【0067】

表5は、ダブルワード未満の基準データ・サイズについて、値の複製方法を示す。

【表5】

基準 サイズ	B7	B6	B5	B4	B3	B2	B1	B0
バイト	値	値	値	値	値	値	値	値
半 ワード	値		値		値		値	
ワード	値				値			
ダブル ワード	値							

表 5

【0068】

列B0からB7は、下位データ基準レジスタ505および上位データ基準レジスタ507における8バイトに対応する。下位バイトB0・B3は、下位データ基準レジスタ505に格納され、上位バイトB4・B7は上位データ基準レジスタ507に格納される。このようなデータの複製を、適切なデータ・ストロープと共に行うことによって、メモリ・アクセス・サイズおよび基準サイズのいずれの組み合わせについても、いかなる所望の一致でも可能となる。データ比較には、アクセスによるデータ・バイト基準のみが含まれる。

【0069】

本発明の好適な実施形態は、2つのデータ比較器500を含む。第1のデータ比較器を図5に示す。第2のアドレス比較器は、比較器制御レジスタ403が比較器制御レジスタ

10

20

30

40

50

405と置き換えられ、アドレス基準レジスタ404がアドレス基準レジスタ406と置き換えられ、比較出力data_compare_result_1が、ORゲート416ではなく、ORゲート426に供給されることを除いて同様である。更に別の代替物として、図5に示すような2つのデータ比較器を共に用いて、データ範囲の比較を行うこともできる。一方のデータ比較器はデータ範囲の下側の境界を検出し、第2のデータ比較器はデータ範囲の上側の境界を検出する。このデータ範囲比較の結果は、データ比較器認定信号を通じて、データ比較認定ORゲート416および426の選択された一方または双方に供給することができる。

【0070】

以上の説明に関して更に以下の項目を開示する。

10

(1) メモリ・アクセス・アドレス比較器であって、

第1基準アドレスを格納する第1プログラマブル基準アドレス・レジスタと、

前記第1基準アドレスおよび第1アドレス入力を受け取り、前記第1基準アドレスと前記アドレス入力との間の整合を示す少なくとも1つの第1整合信号を発生する第1アドレス整合マップ作成部と、

前記第1プログラマブル基準アドレス・レジスタからの前記第1基準アドレスと前記第1アドレス入力とを受け取り、前記第1基準アドレスと前記アドレス入力上のアドレスとの関係に応じて、超過出力、未満出力または同等出力を発生する第1比較器と、

第2基準アドレスを格納する第2プログラマブル基準アドレス・レジスタと、

前記第2基準アドレスおよび第2アドレス入力を受け取り、前記第2基準アドレスと前記アドレス入力との間のオフセットを示す少なくとも1つの第2オフセット信号を発生する第2アドレス整合マップ作成部と、

20

前記第2プログラマブル基準アドレス・レジスタからの前記第2基準アドレスと前記アドレス入力とを受け取り、前記第2基準アドレスと前記アドレス入力上のアドレスとの関係に応じて、超過出力、未満出力または同等出力を発生する第2比較器と、

第1および第2アドレス整合マップ作成部のそれぞれならびに第1および第2比較器のそれぞれに接続された第1および第2メモリ・アクセス制御ユニットであって、各々、前記少なくとも1つの整合信号、前記対応する比較器の前記超過出力、前記未満出力および前記同等出力の少なくとも1つに応じて、対応する第1および第2ローカル・イベント信号を選択的に発生し、更に、前記他方のメモリ・アクセス制御ユニットの前記ローカル信号に応じて対応する第1および第2メモリ・アクセス・イベント信号を選択的に発生する、第1および第2メモリ・アクセス制御ユニットと、

30

を備え、

前記第1メモリ・アクセス制御ユニットが、前記少なくとも1つの第1整合信号を受け取った場合、前記第1ローカル・イベント信号および前記第1メモリ・アクセス・イベント信号が、前記第1基準アドレスおよび前記アドレス入力間における完全な重複または部分的重複に選択的に応じ、

前記第2メモリ・アクセス制御ユニットが、前記少なくとも1つの第2整合信号を受け取った場合、前記第2ローカル・イベント信号および前記第2メモリ・アクセス・イベント信号が、前記第2基準アドレスおよび前記アドレス入力間における完全な重複または部分的重複に選択的に応ずる、メモリ・アクセス・アドレス比較器。

40

(2) 更に、

第1データ・アクセス・アドレス・バスと、

第2データ・アクセス・アドレス・バスと、

前記第1データ・アクセス・アドレス・バスを受ける第1入力と、前記第2データ・アクセス・アドレス・バスを受ける第2入力と、制御入力と、前記第1アドレス整合マップ作成部および前記第1比較器のアドレス入力に接続された出力とを有する第1マルチプレクサと、

前記第1データ・アクセス・アドレス・バスを受ける第1入力と、前記第2データ・アクセス・アドレス・バスを受ける第2入力と、制御入力と、前記第2アドレス整合マップ

50

作成部および前記第 2 比較器のアドレス入力に接続された出力とを有する第 2 マルチプレクサと、

前記第 1 および第 2 マルチプレクサの前記制御入力に接続されたプログラマブル・バス選択レジスタであって、前記第 1 および第 2 マルチプレクサを独立して制御し、前記第 1 および第 2 比較器の前記アドレス入力への供給のために、前記第 1 データ・アクセス・アドレス・バスまたは前記第 2 データ・アクセス・アドレス・バスのいずれかを選択する、プログラマブル・バス選択レジスタとを備えている第 1 項記載のメモリ・アクセス・アドレス比較器。

(3) 更に、

前記第 1 メモリ・アクセス制御ユニットに接続され、前記第 1 メモリ・アクセス制御ユニットを制御して、前記アドレス入力、前記第 1 基準アドレスに対して(1)超過、(2)未満、(3)同等、(4)不等、(5)以下、および(6)以上から選択された1つである場合、前記第 1 ローカル信号を発生し、前記第 1 メモリ・アクセス・イベント信号が前記第 2 ローカル信号に依存するか、またはこれとは無関係とするかを選択する、第 1 プログラマブル比較制御レジスタと、

前記第 2 メモリ・アクセス制御ユニットに接続され、前記第 2 メモリ・アクセス制御ユニットを制御して、前記アドレス信号が、前記第 2 基準アドレスに対して(1)超過、(2)未満、(3)同等、(4)不等、(5)以下、および(6)以上から選択された1つである場合、前記第 2 ローカル信号を発生し、前記第 2 メモリ・アクセス・イベント信号が前記第 1 ローカル信号に依存するか、またはこれとは無関係とするかを選択する、第 2 プログラマブル比較制御レジスタとを備えている第 1 項記載のメモリ・アクセス・アドレス比較器。

(4) 更に、

前記第 1 メモリ・アクセス制御ユニットに接続され、前記第 1 メモリ・アクセス制御ユニットを制御して、前記アドレス入力、前記第 1 基準アドレスおよび前記第 1 アドレス入力間の完全な重複または部分的な重複から選択した一方である場合、前記第 1 ローカル信号を発生する、第 1 プログラマブル比較制御レジスタと、

前記第 2 メモリ・アクセス制御ユニットに接続され、前記第 2 メモリ・アクセス制御ユニットを制御して、前記アドレス入力、前記第 2 基準アドレスおよび前記第 2 アドレス入力間の完全な重複または部分的な重複から選択した一方である場合、前記第 2 ローカル信号を発生する、第 2 プログラマブル比較制御レジスタとを備えている第 1 項記載のメモリ・アクセス・アドレス比較器。

(5) 前記第 1 アドレス入力、前記第 1 アドレスにおけるメモリ・アクセスに対応するデータ・サイズの指示を含み、

更に、前記第 1 メモリ・アクセス制御ユニットに接続され、前記第 1 基準アドレスに対応するデータ・サイズを指定する第 1 プログラマブル比較制御レジスタを備え、前記第 1 基準アドレスおよび前記第 1 アドレス入力のそれぞれのデータ・サイズに応じて、前記アドレス入力、前記第 1 基準アドレスおよび前記第 1 アドレス入力間の完全な重複または部分的な重複から選択した1つである場合に、前記第 1 メモリ・アクセス制御ユニットが前記第 1 ローカル信号を発生し、

前記第 2 アドレス入力、前記第 2 アドレスにおけるメモリ・アクセスに対応するデータ・サイズの指示を含み、

更に、前記第 2 メモリ・アクセス制御ユニットに接続され、前記第 1 基準アドレスに対応するデータ・サイズを指定する第 2 プログラマブル比較制御レジスタを備え、前記第 2 基準アドレスおよび前記第 2 アドレス入力のそれぞれのデータ・サイズに応じて、前記アドレス入力、前記第 2 基準アドレスおよび前記第 2 アドレス入力間の完全な重複または部分的な重複から選択した1つである場合に、前記第 2 メモリ・アクセス制御ユニットが前記第 2 ローカル信号を発生する、第 1 項記載のメモリ・アクセス・アドレス比較器。

(6) 更に、

少なくとも1つのデータ比較器であって、各々、メモリ・アクセスのデータに対応する

データ基準と一致するか否かを示す、対応するデータ一致信号を発生する、データ比較器を備えており、

前記第1メモリ・アクセス制御ユニットが、前記少なくとも1つのデータ比較器から前記一致信号を受け取った場合、前記第1ローカル・イベント信号および前記第1メモリ・アクセス・イベント信号が、前記対応するデータ一致信号に選択的に依存し、

前記第2メモリ・アクセス制御ユニットが、前記少なくとも1つのデータ比較器から前記一致信号を受け取った場合、前記第2ローカル・イベント信号および前記第2メモリ・アクセス・イベント信号が、対応するデータ一致信号に選択的に依存する、第1項記載のメモリ・アクセス・アドレス比較器。

(7) 更に、

前記第1メモリ・アクセス制御ユニットに接続され、前記第1メモリ・アクセス制御ユニットを制御して、前記第1アドレス入力が、前記第1基準アドレスおよび前記第1アドレス入力間の完全な重複または部分的な重複から選択した一方である場合、前記第1ローカル信号を発生する、第1プログラマブル比較制御レジスタと、

前記第2メモリ・アクセス制御ユニットに接続され、前記第2メモリ・アクセス制御ユニットを制御して、前記アドレス入力が、前記第2基準アドレスおよび前記第2アドレス入力間の完全な重複または部分的な重複から選択した一方である場合、前記第2ローカル信号を発生する、第2プログラマブル比較制御レジスタとを備えている、第6項記載のメモリ・アクセス・アドレス比較器。

(8) 前記少なくとも1つのデータ比較器は、単一のデータ一致信号を発生する単一のデータ比較器から成り、前記第1および第2メモリ・アクセス制御ユニットが前記単一のデータ比較器から前記単一のデータ一致信号を受け取る、第6項記載のメモリ・アクセス・アドレス比較器。

(9) 前記少なくとも1つのデータ比較器は、

第1データ一致信号を発生する第1データ比較器であって、前記第1メモリ・アクセス制御ユニットが前記第1データ一致信号を受け取る、第1データ比較器と、

第2データ一致信号を発生する第2データ比較器であって、前記第2メモリ・アクセス制御ユニットが前記第2データ一致信号を受け取る、第2比較器と、
から成る、第6項記載のメモリ・アクセス・アドレス比較器。

(10) 前記少なくとも1つのデータ比較器は、

前記データ基準を格納するプログラマブル・データ基準レジスタと、

前記プログラマブル・データ基準レジスタ内に格納されている前記データ基準のそれぞれの区間と、データ・バス上のデータとを比較し、それぞれの区間の一致を示す区間一致信号を発生する、データ比較器と、

前記区間一致信号を受け取り、該区間一致信号に応じて選択的にデータ比較信号を発生するメモリ・アクセス・データ比較制御ユニットと、
を含む、第9項記載のメモリ・アクセス・アドレス比較器。

(11) 前記少なくとも1つのデータ比較器は、更に、

前記メモリ・アクセス・データ比較制御ユニットに接続され、前記メモリ・アクセス・データ比較制御ユニットを制御して、前記アドレス入力が、全区間の一致または選択した区間の一致から選択した一方である場合、前記データ比較信号を発生するプログラマブル比較制御レジスタを含む、第10項記載のメモリ・アクセス比較器。

(12) 更に、

第1データ・バスと、

第2データ・バスと、

前記第1データ・バスを受ける第1入力と、前記第2データ・バスを受ける第2入力と、制御入力と、前記データ比較器に接続された出力とを有するマルチプレクサと、

前記第1マルチプレクサの制御入力に接続されたプログラマブル・バス選択レジスタであって、前記マルチプレクサを制御して、前記データ比較器の前記アドレス入力に供給するために、前記第1データ・バスまたは前記第2データ・バスのいずれかを選択する、プ

10

20

30

40

50

ログラマブル・バス選択レジスタと、
を備えている第9項記載のメモリ・アクセス・アドレス比較器。

【0071】

(13) メモリ・アクセス・アドレス比較器は、入力メモリ・アクセス・アドレス(402)をそれぞれの基準アドレス(404、406)と比較する2つの比較器(413、423)を含む。比較器(413、423)は、アドレス・サイズ、完全なまたは部分的な重複、超過、未満、同等、不等、以下、および以上というような、選択可能な判断基準によって一致の指示を出し、選択的に連鎖することができる。入力マルチプレクサ(411、412、421、422)は、メモリ・アクセス・アドレス・バスの選択を可能にする。比較器出力コントローラ(415、425)は、対応するデータ一致(data_compare_result_0、data_compare_result_1)に選択的に応ずることができる。コンフィギュレーション・インターフェース・コントローラ(401)により、中央演算装置は、基準アドレス、比較データおよび制御機能(cfgb_data)を、メモリ・マップ・レジスタ(420、403、404、405、406)を通じて指定することができる。

10

【図面の簡単な説明】

【0072】

【図1】典型的な集積回路の可視性および制御を、システム集積度の増大による時間の関数として示す図。

【図2】本発明を適用可能なエミュレーション・システムを示す図。

【図3】構成変更可能なエミュレーション機能を用いた典型的な集積回路を示すブロック図。

20

【図4】2つの結合されたメモリ・アクセス・アドレス比較器を示すブロック図。

【図5】データ比較器の一例を示すブロック図。

【符号の説明】

【0073】

- 100 エミュレータ・システム
- 110 デバッガ・アプリケーション・プログラム
- 120 ホスト・コンピュータ
- 130 エミュレーション・コントローラ
- 140 オンチップ・デバッグ機構
- 201 CPUコア
- 210 バス・イベント検出器
- 211 補助イベント検出器
- 213 カウンタ/ステートマシン
- 220 トリガ・ビルダ
- 230 トレース収集
- 240 トレース・エクスポート
- 241 リアル・タイム・データ交換エクスポート
- 245 局部発振器
- 250 走査アダプタ
- 260 ピン・マネージャおよびピン・マクロ
- 400 比較器
- 401 コンフィギュレーション・インターフェース制御部
- 402 バス選択レジスタ
- 403、405 比較器制御レジスタ
- 404、406 アドレス基準レジスタ
- 411、412、421、422 マルチプレクサ
- 413、423 メモリ・アドレス大小比較器
- 414、424 メモリ・アドレス・バイト整合マップ作成部
- 415、425 メモリ・アクセス比較制御ブロック

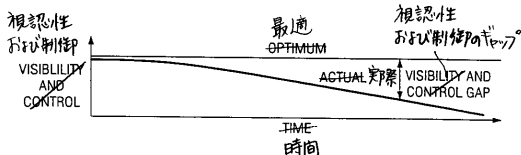
30

40

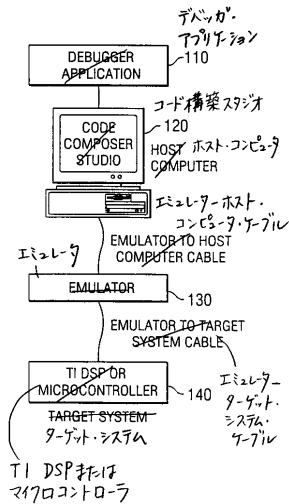
50

- 4 1 6、 4 2 6 O R ゲート
- 5 0 0 データ比較器
- 5 0 5 下位データ基準レジスタ
- 5 0 7 上位データ基準レジスタ
- 5 0 6 下位マスク・レジスタ
- 5 0 8 上位マスク・レジスタ
- 5 1 0 データ・マルチプレクサおよび経路制御ロジック
- 5 1 1、 5 1 2、 5 1 3、 5 1 4 マルチプレクサ
- 5 2 1、 5 2 2 メモリ・データ同一性比較器
- 5 3 0 メモリ・アクセス・データ・バイト・アドレス範囲重複比較器
- 5 4 0 メモリ・アクセス・データ比較制御部

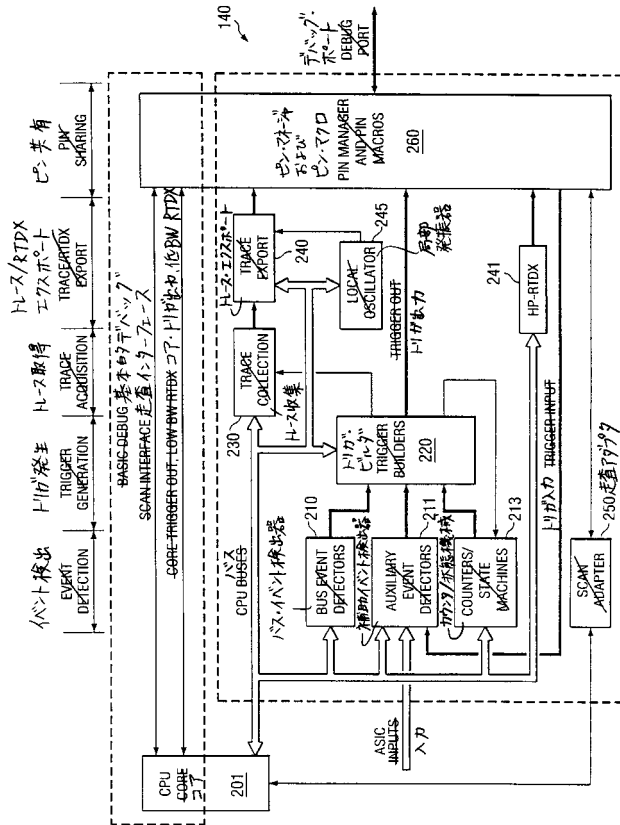
【 図 1 】



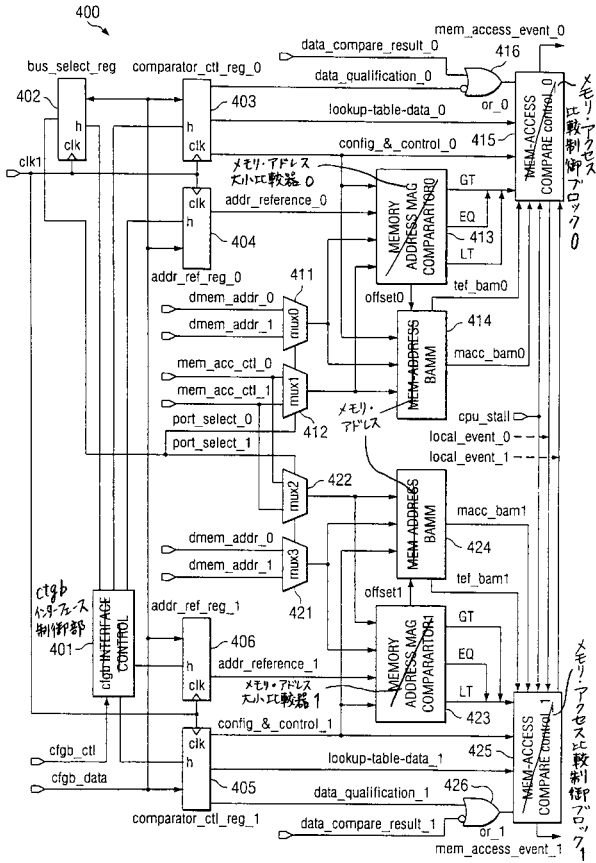
【 図 2 】



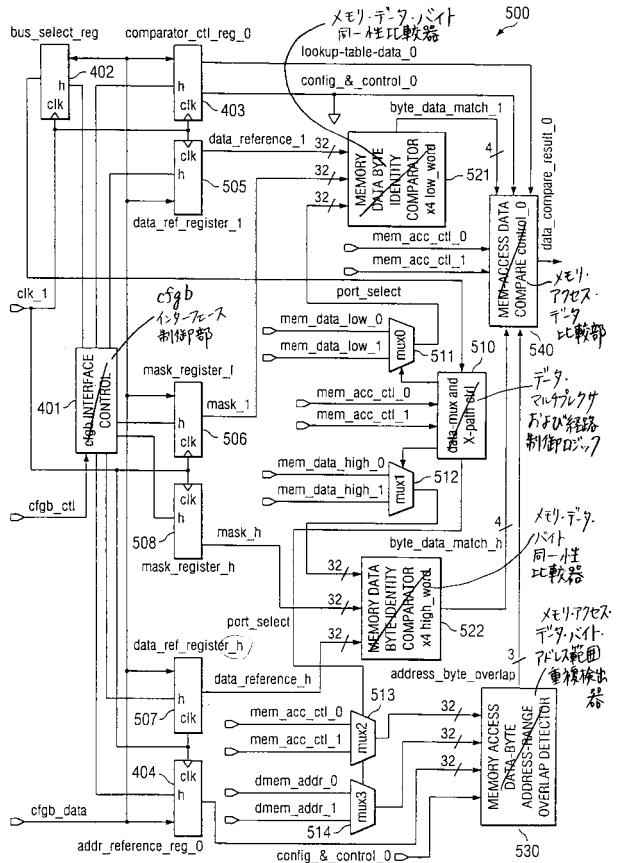
【 図 3 】



【図4】



【図5】



フロントページの続き

(72)発明者 ホセ エル フロレス

アメリカ合衆国 テキサス、リチャードソン、 ボルティモア ドライブ 1 4 1 2

(72)発明者 ルイス ナルディーニ

アメリカ合衆国 テキサス、ダラス、 フリント フォールズ ドライブ 8 8 2 5

(72)発明者 マリア ビー、エイチ、ギル

アメリカ合衆国 テキサス、プレイノー、エンカント コート 3 8 0 4

Fターム(参考) 5B042 GA13 GC02 HH03 HH04 HH25 LA09 LA13 LA16

5B048 AA12 BB02 EE06 FF06