

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3935847号
(P3935847)

(45) 発行日 平成19年6月27日(2007.6.27)

(24) 登録日 平成19年3月30日(2007.3.30)

(51) Int. Cl.

H03K 19/173 (2006.01)

F I

H03K 19/173 101

請求項の数 13 (全 60 頁)

(21) 出願番号	特願2002-591993 (P2002-591993)	(73) 特許権者	591025439
(86) (22) 出願日	平成14年4月17日(2002.4.17)		ザイリンクス インコーポレイテッド
(65) 公表番号	特表2004-529581 (P2004-529581A)		X I L I N X I N C O R P O R A T E D
(43) 公表日	平成16年9月24日(2004.9.24)		アメリカ合衆国 カリフォルニア州 95
(86) 国際出願番号	PCT/US2002/012234		124-3400 サン ホセ ロジック
(87) 国際公開番号	W02002/095598		ドライブ 2100
(87) 国際公開日	平成14年11月28日(2002.11.28)	(74) 代理人	100064746
審査請求日	平成17年3月10日(2005.3.10)		弁理士 深見 久郎
(31) 優先権主張番号	09/861,112	(74) 代理人	100085132
(32) 優先日	平成13年5月18日(2001.5.18)		弁理士 森田 俊雄
(33) 優先権主張国	米国 (US)	(74) 代理人	100083703
			弁理士 仲村 義平
		(74) 代理人	100096781
			弁理士 堀井 豊

最終頁に続く

(54) 【発明の名称】 プログラマブルインターフェイスコアおよび中央処理装置を含むプログラマブルロジックデバイス

(57) 【特許請求の範囲】

【請求項 1】

プログラマブルロジックデバイスであって、

中央処理装置と、

前記中央処理装置に結合されるプログラマブルインターフェイスとを含み、前記プログラマブルインターフェイスは、コアを含み、前記プログラマブルロジックデバイスはさらに、

前記プログラマブルインターフェイスに結合される周辺バスを含み、ブリッジモジュールが、前記周辺バスを前記プログラマブルインターフェイスに結合し、前記プログラマブルロジックデバイスはさらに、

前記プログラマブルインターフェイスを介して情報を前記中央処理装置へ提供すること、および前記プログラマブルインターフェイスを介して情報を前記中央処理装置から受取ることのうちの一方のための第1のデバイスを含む、プログラマブルロジックデバイス。

【請求項 2】

複数のデバイスをさらに含み、各デバイスは、前記プログラマブルインターフェイスを介して情報を前記中央処理装置に提供すること、前記プログラマブルインターフェイスを介して情報を前記中央処理装置から受取ること、および前記プログラマブルインターフェイスを介して別のデバイスと通信することのうちの1つのためにある、請求項1に記載のプログラマブルロジックデバイス。

【請求項 3】

前記プログラマブルインターフェイスは、前記複数のデバイスを結合するためのクロスポイントスイッチを含み、

前記クロスポイントスイッチは、アドレス／制御経路およびデータ経路を含む、請求項2に記載のプログラマブルロジックデバイス。

【請求項4】

前記アドレス／制御経路は、第1のプログラマブルリソースセットから形成され、前記データ経路は、第2のプログラマブルリソースセットから形成される、請求項3に記載のプログラマブルロジックデバイス。

【請求項5】

前記アドレス／制御経路およびデータ経路は、トリミング可能である、請求項3に記載のプログラマブルロジックデバイス。

10

【請求項6】

前記クロスポイントスイッチは、パラメータ表示可能である、請求項3に記載のプログラマブルロジックデバイス。

【請求項7】

前記複数のデバイスからの第1のデバイスグループが、前記周辺バスに結合され、前記ブリッジモジュールを介して前記プログラマブルインターフェイスに結合される、請求項1に記載のプログラマブルロジックデバイス。

【請求項8】

前記複数のデバイスからの第2のデバイスグループは、前記ブリッジモジュールを介して前記プログラマブルインターフェイスに接続されない、請求項7に記載のプログラマブルロジックデバイス。

20

【請求項9】

前記第1のデバイスグループは、ユーザによって指定される複数のコアを含む、請求項7に記載のプログラマブルロジックデバイス。

【請求項10】

少なくとも1つのコアが、マスタ／スレーブ機能、または、マスタ機能またはスレーブ機能を提供する、請求項9に記載のプログラマブルロジックデバイス。

【請求項11】

少なくとも1つのメモリデバイスをさらに含み、前記少なくとも1つのメモリデバイスは、前記第2のデバイスグループの一部である、請求項8に記載のプログラマブルロジックデバイス。

30

【請求項12】

オフチップデバイスをさらに含み、前記オフチップデバイスは、前記第2のデバイスグループの一部であり、

前記オフチップデバイスは、ダブルデータレートデバイスまたはメモリデバイスを含む、請求項8に記載のプログラマブルロジックデバイス。

【請求項13】

前記プログラマブルインターフェイスおよび前記周辺バスに結合される高速バスインターフェイスをさらに含み、前記高速バスインターフェイスは、オフチップデバイスに対して通信を提供する、請求項1に記載のプログラマブルロジックデバイス。

40

【発明の詳細な説明】

【技術分野】

【0001】

発明の分野

この発明は、プログラマブルロジックデバイスに関し、具体的には、インターフェイスコアおよび中央処理装置を含むプログラマブルロジックデバイスに関する。

【背景技術】

【0002】

関連技術の説明

50

マイクロプロセッサは、コンピュータを制御する周知の集積回路である。マイクロプロセッサがコンピュータを制御する様態によって、コンピュータの速度およびパワーが決定される。コンピュータを制御するために、マイクロプロセッサは、無数の命令を処理し、その各々が、マイクロプロセッサに対してある特定の動作を実行するように伝える。典型的なマイクロプロセッサは、所望の演算および論理機能を実行する中央処理装置（CPU）およびCPUが入力を受取ってこれらの機能を実行し、および/または、その結果をコンピュータの構成要素の残りの部分と通信することを可能にする入力/出力ユニットを含む。

【0003】

図1は、CPU101および2つの拡張バス102および103を含む公知のマイクロプロセッサ100を例示する。拡張バスによって、CPU101が他のデバイスと通信することが可能となり、マイクロプロセッサの動作が拡張される。プロセッサローカルバス（PLB）とも称される拡張バス102は、CPU101を高速デバイス104に接続する。これらの高速デバイス104は、メモリおよび高性能の周辺装置を含み得る。PLB102を制御してそれ自体の転送を扱うデバイスが、「マスタ」と呼ばれ、マスタからコマンドを受取ってデータを送るデバイスが、「スレーブ」と呼ばれる。

【0004】

オンチップ周辺バス（OPB）とも称される拡張バス103が、低速デバイス105へのアクセスをCPU101に提供する。これらの低速デバイス105は、UARTおよびイーサネット（R）接続を含み得る。なお、低速デバイス105は、高速デバイス104のように、マスタとスレーブとの両方を含み得る。しかしながら、これらの低速デバイス105がCPU101の性能に影響を及ぼすことを防ぐために、OPB103は、CPU101に直接接続されない。代わりに、OPB103は、OPブリッジ106を介してPLB102に結合される。低速デバイスにメインバス上の貴重な帯域幅を消費させないように、大部分のシステムで、バスのセグメンテーションが行なわれることが検討されることが注目されるべきである。しかしながら、メインバスにほとんど影響を及ぼさずに、より高速の周辺装置の有利な動作を可能にする、メインバスと並列なバスとして、セグメンテーション部を扱うことが、多くの場合に有利である。OPブリッジ106は、データを1つのバスから別のものへと移動させるように設計されるシステムビルディングブロックである。OPブリッジ106は、自動的にデータフォーマットおよびプロトコルを変換し得、OPB103とPLB102との間の情報転送を容易にする。

【0005】

特定用途向け集積回路（ASIC）も、集積回路技術において周知である。ASICは、予め定められたアプリケーション用に設計されるチップである。ASICを構築するために、複数の基本的な回路セルが相互接続される。通例、これらの回路セルは、ASICのメーカによって提供されるライブラリ内で見つけられる。現在、いくつかのメーカは、ASIC上での集積のためのマイクロプロセッサセルを提供している。これらの種類のマイクロプロセッサは、「組込用（embedded）」マイクロプロセッサと呼ばれる。

【0006】

プログラマブルロジックデバイス（PLD）も、集積回路技術において周知である。PLDは、論理設計を実現するフィールドでユーザによってプログラムされ得る。1つの種類のPLDは、書替え可能ゲートアレイ（FPGA）である。通例のアーキテクチャでは、FPGAは、プログラマブル入力/出力ブロック（IOB）によって囲まれるコンフィギュラブルロジックブロック（CLB）のアレイを含む。IOBは、パッケージピンとCLBとの間でインターフェイスを提供し、CLBは、FPGA上でロジックを構成するための機能的要素を提供する。CLBおよびIOBは、プログラマブルルーティングリソースの階層によって相互接続される。これらのCLB、IOB、およびプログラマブルルーティングリソースは、コンフィギュレーションビットストリームをFPGAにロードすることによって、カスタマイズされる。このコンフィギュレーションビットストリームは、ソフトウェアツールを用いて生成される。

10

20

30

40

50

【 0 0 0 7 】

図 2 は、ザイリンクス社 (Xilinx, Inc.) が販売するバーテックス (Virtex) (商標) F P G A 内の C L B 2 0 0 を示す簡素化されたブロック図を例示する。C L B 2 0 0 は、2 つの「スライス」2 0 1 A および 2 0 1 B を含む。各スライス 2 0 1 は、プログラマブルインターコネクトから第 1 の複数の入力信号 G 1 - G 4 を受けるための第 1 の L U T 2 0 2 と、同様にプログラマブルインターコネクトから第 2 の複数の入力信号 F 1 - F 4 を受けるための第 2 の L U T 2 0 3 とを含む。各スライス 2 0 1 は、L U T 2 0 2 からの出力信号 Y を (その D 端子上で) 受け、さらには信号 Y Q を (その Q 端子上で) 出力するためのフリップフロップ 2 0 4 をさらに含む。同様の様態で、各スライス 2 0 1 は、L U T 2 0 3 からの出力信号 X を (その D 端子上で) 受け、さらには信号 X Q を (その Q 端子上で) 出力するためのフリップフロップ 2 0 5 をさらに含む。バーテックス (商標) F P G A についてのこれ以外の詳細は、ザイリンクス社によって出版される「プログラマブルロジックデータブック 2 0 0 0」(“The Programmable Logic Data Book 2000”) の第 3 - 7 頁から第 3 - 1 7 頁、および第 3 - 7 6 頁から第 3 - 8 7 頁上に開示される。

10

【 0 0 0 8 】

バーテックス F P G A のようないくつかの F P G A は、「コア」と呼ばれる予め設計された機能を備えたブロックを組み込むようにプログラムされ得る。一実施例では、コアは、1 つ以上の機能を実行するように F P G A をプログラムする予め定められたコンフィギュレーションビットセットを含み得る。別の実施例では、コアは、設計の接続性および論理を示す回路図 (schematics) またはソースコードを含み得る。典型的なコアは、デジタル信号処理 (D S P) 機能、メモリ、記憶素子、および数学的機能を提供し得るが、これらに限定されているわけではない。コアには、特定の F P G A のための、最適に構成されたレイアウトが提供され得る。コアはさらに、パラメータ表示可能であり得、つまり、ユーザがパラメータを入力して特定のコア機能を活性化または変更することを可能にする。

20

【 0 0 0 9 】

図 3 は、F P G A 内で 1 つ以上のコアを用いるための 1 つの I P プロセスレイアウト 3 0 0 を例示する。ここで、I P とは、F P G A のための論理設計等の一般的な知的財産を指す。この発明の譲受人または他によって開発された、ユーザおよびコアによって提供される両方の論理設計が、I P とみなされる。レイアウト 3 0 0 では、システムレベルツール 3 0 1 によって活性化されるコアジェネレータ 3 0 4 が、サードパーティが提供する 1 つ以上のプラグインコア 3 0 2 を受取り得る。代替的には、コアジェネレータ 3 0 4 は、ライブラリセット 3 0 3 内で提供されるコアを用い得、これらのライブラリ 3 0 3 は、コアジェネレータ 3 0 4 の一部を形成する。一旦コアが選択されると、それは、F P G A ソフトウェアツール 3 0 6 に提供され得る。例示的な F P G A ソフトウェアは、アライアンス (Alliance) (商標)、ファンデーション (Foundation) (商標)、またはファンデーション I S E (商標) ソフトウェアを含み得、すべてが、この発明の譲受人によってライセンス供与されている。このソフトウェアは、ユーザが提供するトップレベル設計 3 0 5 を受取ることにもできる。このトップレベル設計は、コアに加えて F P G A 上で実現される論理設計を示す。一実施例では、F P G A ソフトウェアツール 3 0 6 は、V H D L、V e r i l o g、または標準の概略的な形で、トップレベル設計 3 0 5 を受取り得る。F P G A ソフトウェアツール 3 0 6 は、F P G A 3 0 7 をプログラムするコンフィギュレーションビットストリームを生成して、コアおよびトップレベル設計によって指定される機能を提供する。

30

40

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 1 0 】

技術の向上が実現されるにつれて、F P G A は、以前は A S I C 専用であったアプリケーションへとますます移行しつつある。しかしながら、物理的なリソースは、A S I C 上よりも F P G A 上においてより制約される。したがって、F P G A 上でマイクロプロセッサ構成要素を提供することは、特別な配慮を必要とする重要な作業となる。

50

【課題を解決するための手段】

【0011】

発明の概要

この発明に従うと、プログラマブルロジックデバイス（ＰＬＤ）は、中央処理装置（ＣＰＵ）およびＣＰＵに結合されるプログラマブルインターフェイスを含み、プログラマブルインターフェイスは、ユーザによって指定されるコアにより作成される。通例、ＰＬＤは、複数のデバイスをさらに含み、各デバイスは、プログラマブルインターフェイスを介してＣＰＵに情報を提供するか、プログラマブルインターフェイスを介してＣＰＵから情報を受取るか、またはプログラマブルインターフェイスを介して別のデバイスと通信するかのいずれかである。

10

【0012】

一実施例では、プログラマブルインターフェイスは、複数のデバイスを結合するためのクロスポイントスイッチを含む。クロスポイントスイッチは、アドレス／制御経路とデータ経路との両方を含む。アドレス／制御経路は、ＰＬＤ上の第１のプログラマブルリソースセットから形成され得、データ経路は、ＰＬＤ上の第２のプログラマブルリソースセットから形成され得る。第１および第２のプログラマブルリソースセットは、別々であり得るか、またはＰＬＤ上でオーバーラップし得る。

【0013】

一実施例では、ＰＬＤは、プログラマブルインターフェイスに結合されるオンチップ周辺バス（ＯＰＢ）をさらに含む。ブリッジモジュールが、ＯＰＢをプログラマブルインターフェイスに結合する。マスタとスレーブとの両方を含む、ＰＬＤ上の第１のデバイスグループが、ＯＰＢに結合され得る。したがって、この第１のマスタおよびスレーブグループは、ブリッジモジュールを用いてプログラマブルインターフェイスと通信し得る。第１のマスタおよびスレーブグループは、低速デバイスとして通例特徴づけられる周辺デバイスを含み得る。一実施例では、これらのマスタおよびスレーブのうちの少なくとも１つが、コアを用いるＰＬＤ上で作成される。別の実施例では、これらのマスタおよびスレーブのうちの少なくとも１つが、チップ外に置かれる。さらに別の実施例では、コアを用いて、マスタ機能とスレーブ機能との両方を有するデバイス（第１のグループのうちの１つ）が作成され得る。第１のデバイスグループは、ＰＬＤ上またはチップ外に存在し得る。

20

【0014】

同様にマスタおよびスレーブを含む、ＰＬＤ上の第２のデバイスグループは、プログラマブルインターフェイスに結合され得る。第２のデバイスグループは、メモリデバイスまたはダブルデータレートデバイス等の高速デバイスを含み得る。第２のデバイスグループは、ＰＬＤ上またはチップ外に存在し得る。一実施例では、第２のデバイスグループ内のマスタのうちの１つは、プログラマブルインターフェイスおよびＯＰＢに結合される高速バスインターフェイスを含み、高速バスインターフェイスは、チップ外のデバイスに通信を提供する。この発明のＰＬＤは、他の構成要素を含むシステムに容易に組込まれ得、プログラマブルインターフェイスおよび高速バスインターフェイスは、有利なことには、これらの構成要素に対して、ＣＰＵ関連の機能への効率的なアクセスを提供する。

30

【0015】

この発明の一実施例では、プログラマブルインターフェイスは、クロスポイントスイッチを含む。アドレス／制御論理では、クロスポイントスイッチは、アドレスをマスタから受取るための複数のアドレスデコーダと、デコードされたアドレスをアドレスデコーダから受取り、さらにはマスタとスレーブとの間でトランザクションをアービトレートするための複数のアクセスアービタとを含み得る。クロスポイントスイッチは、アドレスデコーダとアクセスアービタとの間で、十分な、またはそれ未満の接続性を提供し得る。

40

【0016】

データ論理では、クロスポイントスイッチは、マスタから書込データを受取るための複数の書込マルチプレクサを含み得、各書込マルチプレクサは、書込データを１つのスレーブに対して選択的に提供し、クロスポイントスイッチは、スレーブから読出データを受取

50

るための複数の読出マルチプレクサを含み得、各読出マルチプレクサは、読出データを1つのマスタに選択的に提供する。書込マルチプレクサおよび読出マルチプレクサは、複数のアクセスアービタによって制御され得る。クロスポイントスイッチは、各書込マルチプレクサと複数のマスタデバイスとの間で、十分な、またはそれ未満の接続性を提供し得る。クロスポイントスイッチはまた、各読出マルチプレクサと複数のスレーブデバイスとの間で、十分な、またはそれ未満の接続性を提供し得る。一実施例では、アドレス/制御および/またはデータ論理は、マスタとスレーブとの間のトランザクションをパイプライン化するためのレジスタをさらに含む。

【0017】

この発明の別の実施例に従うと、組込用中央処理装置(CPU)を有する書替え可能ゲートアレイ(FPGA)は、コアによって作られるハイブリッドスイッチを含み、ハイブリッドスイッチは、FPGA上で実現されるユーザ論理とCPUとの間で通信を提供する。この実施例では、CPUは、マスタデバイスを含み、論理は、複数のスレーブデバイスおよび複数のマスタデバイスを含む。このハイブリッドスイッチは、クロスポイントおよび共有バス構成の両方を含んで、マスタデバイスとスレーブデバイスとの間に経路を提供する。

【0018】

具体的には、ハイブリッドスイッチは、マスタからアドレスを受取るための複数のアドレスデコーダと、デコードされたアドレスをアドレスデコーダから受取り、さらにはマスタとスレーブとの間でトランザクションをアービトレートするための複数のアクセスアービタとを含み得る。少なくとも1つのアクセスアービタが、1つよりも多くのスレーブに対してアービトレーションを提供し、少なくとも1つのアクセスアービタが、1つのスレーブに対してのみアービトレーションを提供する。クロスポイントスイッチはまた、書込データをマスタから受取り、さらには書込データをスレーブに提供するための複数の書込マルチプレクサを含み得る。一実施例では、少なくとも1つの書込マルチプレクサが、その書込データを1つよりも多くのスレーブに提供し、少なくとも1つの書込マルチプレクサが、その書込データを1つのスレーブに対してのみ提供する。クロスポイントスイッチは、読出データをスレーブから受取り、さらには読出データをマスタに選択的に提供するための複数の読出マルチプレクサも含み得る。一実施例では、少なくとも1つの読出マルチプレクサが、その読出データを1つよりも多くのマスタに提供し、少なくとも1つの読出マルチプレクサが、その読出データを1つのマスタに対してのみ提供する。書込マルチプレクサおよび読出マルチプレクサは、複数のアクセスアービタによって制御され得る。ハイブリッドスイッチは、この発明のクロスポイントスイッチのように、マスタとスレーブとの間のトランザクションのためのパイプライン方式を含み得る。

【0019】

この発明の一実施例に従うと、書替え可能ゲートアレイ(FPGA)は、複数のマスタデバイスを含み、組込用マイクロプロセッサは、少なくとも1つのこのようなマスタデバイスであり、書替え可能ゲートアレイ(FPGA)はさらに、複数のスレーブデバイス、および複数のマスタデバイスと複数のスレーブデバイスとの間で接続性を選択的に提供するための手段を含む。接続性を選択的に提供するための手段は、コアによって生成され得る。

【0020】

アドレス/制御論理では、十分な接続性を選択的に提供するための手段は、マスタデバイスからのアドレスをデコードするための手段と、デコードされたアドレスに基づいて、複数のマスタデバイスと複数のスレーブデバイスとの間でトランザクションをアービトレートするための手段とを含み得る。一実施例では、接続性を選択的に提供するための手段は、デコードするための手段とアービトレートするための手段との間で、十分な接続性を提供し、他の実施例では、接続性を選択的に提供するための手段は、デコードするための手段とアービトレートするための手段との間で、十分には満たない接続性を提供する。

【0021】

10

20

30

40

50

データ論理では、接続性を選択的に提供するための手段は、書込データを複数のマスタデバイスから受取り、さらには指定されるスレーブデバイスに書込データを選択的に提供するための手段と、読出データを複数のスレーブデバイスから受取り、さらには指定されるマスタデバイスに読出データを選択的に提供するための手段とを含み得る。一実施例では、書込データを受取るための手段と読出データを受取るための手段とは、アービトレーションするための手段によって制御される。

【0022】

接続性を選択的に提供するための手段は、書込データを受取るための手段と複数のマスタデバイスとの間で、十分な、またはそれ未満の接続性を提供し得る。加えて、接続性を選択的に提供するための手段は、読出データを受取るための手段と複数のスレーブデバイスとの間で、十分な、またはそれ未満の接続性を提供し得る。

10

【0023】

有利なことには、この発明は、公知のPLDソフトウェアツールに含まれ得る。一実施例では、システムレベルツールによって活性化されるコアジェネレータは、サードパーティによって提供される、この発明のプログラマブルインターフェイスコアを受取り得る。代替的には、コアジェネレータは、コアジェネレータの一部を形成するライブラリセット内で提供されるプログラマブルインターフェイスコアを用い得る。一旦プログラマブルインターフェイスコアが選択されると、それは、PLDソフトウェアツールに提供され得る。このソフトウェアは、ユーザが提供するトップレベル設計も受取り得る。このPLDソフトウェアツールは、PLDをプログラムするコンフィギュレーションビットストリームを生成して、プログラマブルインターフェイスコアおよびトップレベル設計が指定する機能を提供する。

20

【0024】

重要なことには、この発明のプログラマブルインターフェイスコアは、複数のコンプライアンスレベルを有し得る。たとえば、コアは、特定の機能だけがPLD上で提供されることを許容し得る。たとえば、一実施例では、プロセッサローカルバス(PLB)機能のサブセットが許容される。既存のIPまたはパラメータ表示されないIPを含む実施例等の他の実施例では、この発明は、サブセットを、そのIPによって必要とされる機能に(または、そのIPに基づいた、予め定められたコンプライアンスレベルに)調整し得る。

【0025】

30

この発明に従うと、プログラマブルロジックデバイス(PLD)上の中央処理装置(CPU)とユーザによって実現されるPLD上の論理との間でインターフェイスを提供するための方法が提供される。この方法は、論理が提供する複数のマスタデバイスと論理が提供する複数のスレーブデバイスとの間で接続性を選択的に提供するステップを含み、CPUは、少なくとも1つのマスタデバイスを含む。アドレス/制御論理中に接続性を選択的に提供するステップは、アドレスデコードを用いてマスタデバイスからのアドレスをデコードし、さらにはアクセスアービタおよびデコードされたアドレスを用いて複数のスレーブデバイスと複数のマスタデバイスとの間でトランザクションをアービトレーションするステップを含む。一実施例では、接続性を選択的に提供するステップは、アドレスデコードとアクセスアービタとの間の十分な接続性を提供する。別の実施例では、接続性を選択的に提供するステップは、アドレスデコードとアクセスアービタとの間の十分には満たない接続性を提供する。

40

【0026】

この発明の方法では、データ論理内で接続性を選択的に提供するステップは、複数のマスタデバイスから書込データを受取り、さらには第1のマルチプレクサセットを用いて書込データを指定されるスレーブデバイスに選択的に提供するステップと、複数のスレーブデバイスから読出データを受取り、さらには第2のマルチプレクサセットを用いて読出データを指定されるマスタデバイスに選択的に提供するステップとを含む。接続性を選択的に提供するステップは、第1のマルチプレクサセットと複数のマスタデバイスとの間で、十分な、またはそれ未満の接続性を提供し得る。接続性を選択的に提供するステップは、

50

第2のマルチプレクサセットと複数のスレーブデバイスとの間で、十分な、またはそれ未満の接続性も提供し得る。

【0027】

この発明の方法では、接続性を選択的に提供するステップは、複数のマスタデバイスと複数のスレーブデバイスとの間でトランザクションをパイプライン化するステップを含む。有利なことには、このパイプライン化は、アドレス/制御論理およびデータ論理内で提供され得、したがって、両方において性能が向上する。

【0028】

この発明に従うと、プログラマブルロジックデバイス（PLD）上の中央処理装置（CPU）とユーザによって実現されるPLD上の論理との間のインターフェイスを提供するための方法が与えられる。この方法は、ライブラリからコアを選択するステップを含み、コアは、論理が提供する複数のマスタデバイスと論理が提供する複数のスレーブデバイスとの間の接続性を選択的に提供し、CPUは、少なくとも1つのマスタデバイスを含む。この方法は、コア内で少なくとも1つのパラメータをセットするステップをさらに含み、したがって、接続性に影響を及ぼす。

【0029】

別の実施例では、システムを構築するために必要とされる各コアからの情報内容が、データ分析のために、集中型機構に集められる。このデータは、コアをインスタンス生成する上で、またはパラメータ表示において、ユーザが代替的な選択を行なうことができるようにするために、ユーザにとって利用可能である。たとえば、コアのパラメータ表示における特定の選択によって、結果として、システム全体に関して異なる性能が得られ得る。

【発明の効果】

【0030】

この発明は、ASIC環境での標準的な組込CPUと比較して、いくつかの重要な利点を提供する。第1に、この発明は、ユーザのニーズおよび/または関連の設計に基づいてパラメータ表示され得る機能の提供において、かなりの柔軟性を提供する。これらのパラメータ表示される機能は、PLD上のプログラマブルリソース内で実現され得、これらの機能がいつでも修正されることが可能になる。第2に、これらのプログラマブルリソースのため、プログラマブルインターフェイス用に実際に必要とされるリソースのみが実現され得、ユーザがPLDの残りの部分の使用を最適化することが可能となる。たとえば、アドレス/制御論理とデータ論理との両方が、「トリミング」され得る。言い換えると、ユーザの設計によって必要とされていない経路（および、それらの関連論理）が除去され得、信号速度を上昇させ、シリコン面積が縮小され得る。別の例では、追加的な領域が利用可能となるため、マスタまたはスレーブデバイスの数が増加し得る。第3に、この発明は、プロセッサローカルバスの機能を効率的に制限することができ、したがって、PLDがASICの性能レベルに近づくか、それどころか、それを越えることを可能にする。

【発明を実施するための最良の形態】

【0031】

図の詳細な説明

以下の目次は、主題構成の概要を読者に伝えるために提供される。

【0032】

目次

I．用語および規定

A．定義

B．信号の名称および番号付け規定

C．タイミング規定

II．リソース区別の認識

III．制約的バス機能

A．PLBサブセット機能

1．アービトレーション制御機能

10

20

30

40

50

2. データ転送機能	
B. PLBスーパーセット機能	
C. 信号: PLBからPICへのマッピング	
IV. FPG A実現例: 概要	
A. 中央処理装置	
B. プログラマブルインターフェイスコア (PIC)	
C. マスタおよびスレーブデバイス	
D. オンチップ周辺バス (OPB)	
E. 高速バスインターフェイス	
V. FPG A実現例: 詳細	10
A. マスタ/スレーブデバイス: ポートおよびタイミング	
1. マスタデバイスとPICとの間の通信	
a. マスタのためのアドレスポート	
b. マスタのための書込ポート	
c. マスタのための読出ポート	
2. マスタデバイスのタイミング	
3. スレーブデバイスとPICとの間の通信	
a. スレーブのためのアドレスポート	
b. スレーブのための書込ポート	
c. スレーブのための読出ポート	20
4. スレーブデバイスのタイミング	
B. プログラマブルインターフェイスコア	
C. DCU/ICU: PLBに対するインターフェイス	
D. BRAM: PLBに対するインターフェイス	
E. OPBブリッジモジュール	
1. ブリッジアウトモジュール	
2. ブリッジインモジュール	
F. ハイブリッドプログラマブルインターフェイスコア	
G. ソフトウェアツール。	
【0033】	30
I. 用語および規定	
この発明は、当業者にとって馴染みのあり得る、または馴染みのない場合もある、特定の用語および規定を用いて説明される。したがって、以下の定義、信号および番号付け規定、およびタイミング規定は、読者がこの発明を理解するときの助けとなるように、さらに詳細に説明される。	
【0034】	
A. 定義	
この説明で用いられる頭文字、略記、および技術用語は、この段落の後で、アルファベット順に定義される。なお、いくつかの頭文字、略記、および技術用語は、詳細な説明で定義され得る。	40
【0035】	
BEAR: バスエラーアドレスレジスタ	
BESR: バスエラーシンドロームレジスタ	
BRAM: プログラマブルロジックデバイス上で提供されるブロックランダムアクセスメモリ (RAM)	
バースト: 1つよりも多くのデータ単位が転送されるトランザクションであり、通例、マスタとスレーブとの間で迅速にデータブロックを移動させるために用いられる	
キャッシュ: 命令および/またはデータを含む、最近アクセスされたデータ内容を記憶するためのCPU内部の小さく、かつ速いメモリ	
コア: ある特定の機能を提供するための、予め設計されたプログラマブルロジックデバ	50

イス部分であって、この部分は、通例、プログラマブルロジックデバイス上の予め定められた量の物理的リソースを用いて実現される

CPU：通例、制御ユニットおよび算術演算装置（ALU）を含む中央処理装置

DCU：データキャッシュユニット

DDR：ダブルデータレート

HSBI：PIC上の高速マスタポートを有するデバイスとして、ここで包括的に言及される高速バスインターフェイス

ICU：命令キャッシュユニット

IP：集積回路、特に、FPGAに組込むための設計

リトルエンディアン：16ビットまたは32ビットワードを仮定した場合、より低いアドレスでのバイトがより低い有効性を有するアーキテクチャ 10

ISA：命令セットアーキテクチャ

OPB：オンチップ周辺バス

PIC：この発明に従って、CPUとプログラマブルロジックデバイスリソースとの間に提供されるプログラマブルインターフェイスコア

パイプライン方式：第1のバスサイクルが完了する前に、マスタデバイスが第2のバスサイクルを実行し始め得るプロセス、つまり、いくつかのバスサイクルを、各々、異なるステージで同時に処理すること

PLB：プロセッサローカルバス

パワーPC：IBM、モトローラ、およびアップルコンピュータによって共同で設計された規格を満たすRISCマイクロプロセッサであって、規格は、共通の命令セットアーキテクチャ（ISA）を特定する 20

レジスタ：FPGA上で提供される高速記憶領域

SDRAM：従来のメモリよりも高いクロック速度で動作することができ、さらにはそれ自体をいくつかのCPUのバスと同期させることのできる同期ダイナミックランダムアクセスメモリ（DRAM）

シングルデータビート：非バースト転送、つまり、バスにおけるシングルクロックサイクル内のシングルデータ

ワード：32ビットのデータ量を示す。

【0036】 30

B．信号の名称および番号付け規定

この発明の特徴を説明するために、さまざまな信号が参照される。ここで用いられる規定に従うと、図6および図8の信号名は、信号の流れの方向、どのデバイスがその信号を送っているのか、どのデバイスがその信号を受取るのかを表す接頭部で始まる。他の図の信号名は、送信を行なうデバイスのみを表す接頭部で開始し得る。デバイス名は、参照しやすくするために、略されている。接頭部の後に下線（ ）が続き、次に、信号自体の名前が続く。接尾部「 n」は、アクティブロー信号を表す。たとえば、信号名PIC2M f o oは、プログラマブルインターフェイスコア（PIC）によって駆動され、かつマスタデバイスによって受取られる信号“f o o”を示す。

【0037】 40

Nビットバスビットは、右側の0から左側のN-1にまで番号が付けられる。なお、この番号付け規定は、さまざまなプロプラエタリCPUとの互換性の問題にかかわらず、PIC ISAを変更せず、またはCPUソフトウェア互換性に影響を及ぼさない。

【0038】

C．タイミング規定

高性能を達成するために、タイミングが考慮されるべきである。当業者に知られているように、タイミングは、多くの場合、ある特定の設計実現例に特有である。したがって、例示的なタイミング参照を含むことは、この発明のパイプライン方式をここでサポートするが、すべての設計のためのタイミングをカバーすることは意図されない。

【0039】

タイミング情報に関して、用語「早い」、「中位」、および「遅い」が用いられ得る。ここで用いられる規定では、早いタイミングが最善であり、遅いタイミングが最も望ましくない。

【 0 0 4 0 】

入力信号が早いタイミングを有する場合、ユーザは、データを、それをラッチする前に処理するためにフルクロックサイクルを有する。早い入力信号タイミングは、信号がレジスタ出力部から駆動されることを暗示する。出力信号が早いタイミングを有する場合、ユーザは、データを送り出す前に、フルクロックサイクルを用い得る。早い出力信号タイミングは、信号がレジスタの入力を駆動することを暗示する。

【 0 0 4 1 】

入力信号が中位のタイミングを有する場合、ユーザは、データを、それをラッチする前に処理するために、約半分のクロックサイクルを有する。中位の入力信号タイミングは、信号が、少量の論理またはユーザへのルーティングを通して、レジスタ出力部から駆動されることを暗示する。出力信号が中位のタイミングを有する場合、ユーザは、データを送り出す前に約半分のクロックサイクルを用い得る。中位の出力信号タイミングは、信号が、レジスタの入力部に到達する前に、少量の論理またはルーティングを駆動することを暗示する。

【 0 0 4 2 】

入力信号が遅いタイミングを有する場合、ユーザは、データを、それをラッチする前に処理するために、クロックサイクルのごく一部を有する。遅い入力信号タイミングは、信号が、多量の論理およびルーティングを通してレジスタ出力部から駆動されることを暗示する。したがって、ユーザは、データを、それを用いる前にラッチするための小さなウィンドウを有する。出力信号が遅いタイミングを有する場合、ユーザは、データを送り出す前に、クロックサイクルのごく一部を用い得る。遅い出力信号タイミングは、信号が、多量の論理またはルーティングを通してレジスタ入力部を駆動することを暗示する。したがって、ユーザは、レジスタから直接この出力を駆動すべきである。

【 0 0 4 3 】

上記の点を考慮すると、特定の実現例が、各信号が提供する手段を説明し得る。いくつかの信号は、より多くの時間を必要とし（たとえば、遅い信号）、これは、システムの動作時の周波数に悪影響を及ぼし得る。これらの問題点に対処するために、パイプライン方式が実現されて、信号のタイミングを緩和し得る。これらのパイプラインレジスタは、たとえレイテンシがあったとしても、クロック周期内のより早い部分に信号を再同期させる機会を提供する。

【 0 0 4 4 】

II. リソース区別の認識

A S I C 内ではなく、プログラマブルロジックデバイス（たとえば、F P G A）内に中央処理装置（C P U）を組込むためには、これらの2種類のデバイスの区別が理解され、対処されなければならない。

【 0 0 4 5 】

性能に影響を及ぼし得る、F P G AとA S I Cとの間の1つの重要な違いは、論理リソースである。具体的には、上述のように、F P G A内の論理ジェネレータは、通例、C L B内のルックアップテーブルを用いて構築される。したがって、高いファンインまたは複雑な組合せの論理機能が、一般に、いくつかのルックアップテーブルレベルを必要とし、したがって、かなりの遅延が作られる。対照的に、A S I Cは、特定の論理機能を実行するために設計されるカスタム構造を提供し得、したがって、多数の論理レベルを除去する。

【 0 0 4 6 】

性能に影響を及ぼし得る、F P G AとA S I Cとの間の別の重要な違いは、ルーティングリソースである。具体的には、F P G Aは、プログラマブルおよび専用の（たとえば、キャリアチェーンロジック）ルーティングチャンネル両方の上で信号をルーティングする。

10

20

30

40

50

対照的に、ASICは、カスタマイズされた信号経路を提供し、したがって、プログラマビリティのオーバーヘッドを解消する。したがって、ASIC相応の速度に近づくために、FPGAは、通例FPGA上で提供されるよりも厚いトレースおよびより大きなバッファを含む必要があり得る。これらのルーティングリソースのアップサイジングは、FPGAのシリコン領域の増大を必要とし得、したがって、その製造コストを望ましくないほど上昇させる。

【0047】

上述の区別に関して、CPUとインターフェイスをとるFPGAリソースは、高速設計で性能と帯域幅との両方を達成することを著しく妨害し得る。しかしながら、この発明に従うと、さらには以下で詳細に説明されるように、FPGAファブリックを有利に活用して、CPUを含む、向上されたトランザクションと標準のトランザクションとの両方が提供され得る。結果として得られるシステムは、現在のASICシステムに近づくことができ、さらにはそれを越えることさえもできる。なぜならば、システムは、この問題に対処するために、IC技術ではなくアーキテクチャを用いるためである。

【0048】

III. 制約的PLB機能

通例のCPUは、PLBプロトコルセットとともに動作して、他のデバイスと通信する。FPGA上で完全なPLBプロトコルセットを提供することは、かなりの論理を必要とし得、したがって、性能と面積との両方に悪影響を及ぼす。したがって、この発明の1つの特徴に従うと、FPGA性能に大きな影響を及ぼすことなく、標準のPLBプロトコルのうちのいくつかは、有利に除去され得る。

【0049】

総括では、この発明は、複雑なデータ経路操作を必要とするPLBプロトコルを解消する。複雑なデータ経路操作は、広いバスおよび長距離に広がる多量の論理を用いる。したがって、FPGA内での複雑なデータ経路操作は、IPユーザに対して制約のある機能だけを追加しながら、かなりのリソースを独占し得る。

【0050】

この発明は、非同期ハンドシェーキングを必要とするPLBプロトコルも解消する。非同期ハンドシェーキングでは、信号は、通例、長い距離（つまり、マスタからスレーブまで、さらにはマスタに戻るまでの経路）を伝わり、有効な論理を通り、これらのすべては、単一のクロックサイクル内にある。したがって、非同期ハンドシェーキングは、複雑なデータ経路操作のように、このタイミング目標を達成するために、かなりのリソースを独占しなければならない。

【0051】

さまざまな標準PLBプロトコルをサポートしないことによって、いくつかの利点を実現され得る。第1に、この発明は、データ論理内のクリティカルパスをかなり簡素化することができる。データ経路論理を簡素化することは、重要である。なぜならば、これらの広いバスは、FPGAにおいて長距離に及ぶためである。言換えると、データ経路内の論理を減じることによって、有利なことには、信号速度が上がる。第2に、この発明は、制御論理内のクリティカルパスをかなり簡素化することができ、したがって、信号速度をさらに上げる。第3に、これらのPLBプロトコルを除去することによって、トランザクションがより確定的となり得る。たとえば、マスタが所与のトランザクションに対してスレーブからより小さなセットの応答を予測できる場合、論理は、通例、簡素化され得る。明らかに、マスタまたはスレーブが有し得る応答またはアクションの種類が多ければ多いほど、論理は、これらの応答またはアクションに対応するために、より複雑にならなければならない。

【0052】

なお、PLBプロトコルの除去は、CPUとともに実行され得るPLB機能の数に直接的な影響を及ぼす。しかしながら、以下で詳細に説明されるように、この発明は、この問題に対処する。

10

20

30

40

50

【 0 0 5 3 】

A . P L B サブセット機能

この発明に従うと、F P G A 内で実現される P L B 機能のサブセットは、C P U が通例実行するトランザクションの種類に焦点を置き、このクリティカルコンポーネントの性能を最適化する。具体的には、P L B 上の大部分のデータ転送は、デバイス メモリ間、またはメモリーデバイス間であり、デバイス間では行われない。したがって、メモリ性能は、P L B において貴重である。重要なことには、大部分の F P G A ユーザメモリインターフェイスが、C P U のそれらと同様のトランザクションを実行する。したがって、P L B 機能のサブセットは、F P G A 上で起こる可能性が最も高いさまざまな種類のトランザクションを有利にサポートし得る。さまざまな P L B 機能を F P G A に組込むことに関する分析が、以下に提供される。

10

【 0 0 5 4 】

1 . アービトレーション制御機能

アドレスデコード機能：標準的な P L B 内のスレーブ内でアドレスデコードを実現することは、現在、共有バスを用いて行われている。共有バスでは、マスタが要求するアドレスが、バス上のすべてのスレーブに提供される。スレーブは、アドレスをデコードし、選択されたスレーブは、次に、アドレス肯定応答をルーティングして、要求をしているマスタに戻す。残念ながら、このプロセスによって、かなりの遅延が起こる。したがって、この発明に従うと、アドレスデコーディングは、(図 5 A - 図 5 E で説明される) プログラマブルインターフェイスコア (P I C) 内で分散されて実行される。このコアのトランザクションフォワードの性質のために、スレーブには、トランザクションがそれに対して呈示される場合、アドレスヒットが保証される。この状態で、この発明は、現在スレーブ内で提供されるアドレスデコード論理を大きく減じ、より容易なスレーブの開発および構築が促進される。なお、ハイブリッドシステムでは、多数のスレーブがバスを共有する場合、追加的なアドレスデコードがスレーブ内に提供され得る。このハイブリッドシステムでは、バスを共有するスレーブの数は、比較的小さく、したがって、アドレスデコード論理が簡素化される。

20

【 0 0 5 5 】

バスエラー (アドレスまたはデータエラー) 機能：バスエラー機能は、この発明の P I C コアと標準の P L B との両方によってサポートされる。

30

【 0 0 5 6 】

アボート機能：アボート機能をサポートすることは、トランザクションフォワードシステム内では困難である。なぜならば、トランザクションは、アボート信号がアサートされる機会を持つ前に、スレーブに渡され得るためである。なお、標準の P L B では、トランザクションは、スレーブが要求を肯定応答した後は、アボートされ得ない。さらに、アボートを P I C コアへと渡すためには、追加的な論理が必要とされ得、システムのタイミングにマイナスの影響を与える。したがって、この発明では、一般的なルールとして、マスタは、アボート信号を生成しない。

【 0 0 5 7 】

マスタがアボート信号を生成する場合、マスタは、インターフェイスモジュールを用い得、これは、要求を迅速に認識し、要求を肯定応答し、さらには要求を直接関連のスレーブに送る。このインターフェイスモジュールは、P I C に対するポートとマスタとの間の経路内に置かれる。したがって、マスタはアボート信号を生成しないという一般的なルールは、維持される。

40

【 0 0 5 8 】

なお、パイプライン化されたシステム内でアボート信号をサポートすることによって、結果として、ユーザのコードが分岐する、または割込む場合に性能の向上がいくらか得られ得る。具体的には、アボートがサポートされ、マスタが、アボートをアサートする時間を許容されている場合、不必要なトランザクションは、早くに中止され得、データ論理の使用が有利に減じられる。分岐する、または割込むコードは、現在マスタが要求する情報

50

の必要性をなくす。データは必要とされていないことを発見した後に、マスタがトランザクションをアポートできる場合、性能が向上する。しかしながら、この利点は、システムの残りの部分のより低いクロック周波数および複雑性によって相殺され得る。

【 0 0 5 9 】

優先されるアービトレーション機能：動的プライオリティに基づいてマスタ間でアービトレートするための組合せ論理を構築することによって、FPGA内の性能を著しく落とすおそれのある多くの論理レベルが必要とされ得る。したがって、この発明の一実施例では、ラウンドロビンまたは固定プライオリティアービトレーションが提供される。他の実施例では、ユーザが選択するアービトレーションが提供され得る。

【 0 0 6 0 】

再試行機能：（時として、再アービトレート機能とも呼ばれる）再試行機能が、通例、複数のマスタ間のデッドロックを防止するために、共有バスアーキテクチャ内で必要とされる。しかしながら、この発明のPICコアに従うと、多数のトランザクションが並列に進行し得、再試行の必要性を著しく減じるか、または単に減じる。なお、トランザクションフォワードシステム内で再試行機能を実現することによって、スレーブからマスタへと信号を渡すことが必要とされ得、アービタ内の論理の複雑性が望ましくないほど増す。

【 0 0 6 1 】

なお、再試行機能は、共有バスを用いるハイブリッドシステム内で提供され得る。ハイブリッドシステムでは、アクセスアービタが修正されて再試行機能が活性化され得、再試行信号がマスタに戻される必要がなくなる。

【 0 0 6 2 】

バスロック機能：バスロック機能によって、ある特定のスレーブが1つのマスタによってのみアクセス可能であることが保証される。しかしながら、バスロック機能は、クリティカルタイミングパス内にある、PICコア内のアービトレーション論理に影響を及ぼし得る。したがって、バスロック機能がイネーブルされ得るが、おそらくは、いくつかの性能損失を伴う。

【 0 0 6 3 】

アドレスパイプライン機能：アドレスパイプライン機能が、標準のPLBと、この発明のPICコアとの両方においてサポートされる。パイプライン方式は、多数のトランザクションが並列に進行することを可能にすることによって、システム性能を著しく向上させ得る。なお、コード分岐は、パイプライン方式の利点を衰えさせ得る。

【 0 0 6 4 】

転送タイプ機能：標準のPLB内の転送タイプ機能は、たとえば、ダイレクトメモリアクセス(DMA)およびフライバイ(fly-by)を含み得る。一般に、各転送タイプは、異なるプロトコルの組合せを必要とする。したがって、複数の転送タイプを提供することによって、かなりの複雑性がFPGAシステムに追加され得る。この発明に従うと、転送は、基本的なメモリ転送に限定されており、したがって、FPGAの複雑性および論理が簡素化される。

【 0 0 6 5 】

2. データ転送機能

シングルデータビート転送機能：シングルデータビート転送機能、つまり、単一データ単位転送は、CPUの基本的な動作であり、したがって、この発明によってサポートされる。なお、この転送機能は、書込動作または読出動作のいずれかであり得、いずれかのバイトイネーブルパターンを含み得る。各トランザクションは、データにおいて最大バス幅のサイズを転送するのみである。なお、この機能は、バーストとは異なる。

【 0 0 6 6 】

不確定な長さのバースト機能：この発明によって提供されるように、パイプライン化されたアドレスおよびデータ経路内で不確定な長さのバースト機能を提供することによって、重大な性能問題がもたらされる。たとえば、マスタは、バーストの終りを、要求されるスレーブに伝達するために、1つ以上のサイクルを必要とし得る。しかしながら、その間

10

20

30

40

50

、スレーブは、あまりに多すぎるデータを送るおそれがある。さらに、アドレスデコーディング論理は、不確定な長さのバーストが、割当てられた時間内にアドレス境界をオーバーランするかを判断することを計算的に困難と認識し得る。したがって、この発明では、不確定な長さのバーストは、サポートされない。

【0067】

バースト終了機能：標準のPLBによって、スレーブがいつでも固定長のバーストを終了させることが可能となる。しかしながら、再度、マスタは、終了信号をスレーブに伝達するために、1つ以上のサイクルを必要とし得る。この時間期間、スレーブは、あまりに多すぎるデータを送るおそれがある。したがって、一実施例では、この発明は、バースト終了機能をサポートしない。代わりに、この発明は、確定的な挙動を、つまり、スレーブが読出肯定応答(RdAck)信号および書込肯定応答(WrAck)信号の正確な数を戻すことを必要とする。別の実施例では、バースト終了信号を用いてデータの流れが制御される場合、スレーブは、RdAck信号とWrAck信号との間に待ち状態を挿入して、より多くの時間を提供し得る。代替的には、問題が生じた場合、スレーブは、読出エラーを示すか、または割込を生成し得る。

10

【0068】

確定した長さのバースト機能：確定した長さのバーストを、つまり、2から16の転送を提供することによって、マスタの設計が簡素化されるが、代わりに、アドレス境界またはページ境界交差を検出するために論理をスレーブに追加することが必要とされる。したがって、一実施例では、この機能は、この発明においてサポートされず、代わりに、キャッシュラインアライメントが提供される。この実施例では、PICコアは、そのサイズにアラインされる2のべき乗の固定長バーストを実行し得る(すなわち、キャッシュライン転送)。別の実施例では、確定した長さのバースト機能がサポートされ、ユーザは、アドレス修飾子を定義し、これらの修飾子をサポートするようにマスタ/スレーブIPを設計する。別の実施例では、確定した長さのバースト機能が、バースト長サイズに対するアラインされた転送として強制的に実現され、アドレス境界チェックを有する必要性をなくす。

20

【0069】

バーストサイズ機能：バーストサイズを変化させることによって、スレーブ内の設計経路の複雑性が増す。具体的には、スレーブは、いずれかの数のデータ幅を扱うために、データ経路を実現する必要がある。一実施例では、バーストサイズは、データ経路幅にセットされ得る。

30

【0070】

キャッシュライン転送機能：キャッシュライン転送、つまり、特定の長さおよびアライメントのアトミックデータバーストが、アドレスまたはページ境界アライメントに関してスレーブの設計を簡素化する。この発明に従うと、キャッシュライン転送がサポートされる。この発明の実施例では、2、4、8、16、および32ワードのキャッシュライン転送が提供される。なお、これらのサイズは、DDR SDRAMを有利にサポートする。

【0071】

ダイナミックバスサイジング機能：ダイナミックバスサイジングは、FPGAの複雑性を著しく増すおそれのあるデータミラーリングおよびステアリングロジックを必要とする。さらに、マスタとスレーブとの両方は、それらの内部論理をデータ転送サイクルの正確な数に調整できるように、転送幅を把握しなければならない。したがって、この発明に従うと、PICコアは、異なる幅のマスタおよびスレーブをサポートしない。代わりに、転送幅は、システムを通して固定されており、したがって、FPGA内の複雑性が減じられる。一実施例では、パラメータ表示されるIPは、32ビットまたは64ビット転送にセットされ得る。ダイナミックバスサイジングの効果は、バイトイネーブル信号およびソフトウェア制御の使用を通して適応され得る。

40

【0072】

B. PLBスーパーセット機能

50

この発明に従うと、いくつかの実現される機能は、標準の P L B 機能のスーパーセットである。たとえば、この発明は、ダブルデータレート (D D R) データ経路のためのサポートを提供し、データ経路は、アドレス経路のクロック周波数の 2 倍で動作し得る。加えて、この発明は、ダイナミック書込バイトイネーブルメカニズムを含んで、書込トランザクションが接続ファブリックにおいて最適化されることが可能になる。具体的には、転送される書込データの各バイトに対して、この発明は、追加的なビットを提供し得、データの各バイトが選択的にスキップされることが可能となる。この状態で、ダイナミック書込バイトイネーブルメカニズムは、効率的にバイトパターンを書込むことができ、非連続的なメモリ書込動作中のシステム性能が強化される。これは、 D M A を用いるシステムにおいて特に有利である。具体的には、 D M A エンジン、は、ミスアラインされたデータを順に 10
実行する (sequence through) ための複雑な論理をもはや必要としない。代わりに、 D M A エンジン、は、転送内容に適応するために適切なバイトイネーブルパターンを示すのみでよい。さらに、 D M A エンジン、は、そのデータを転送するために多くのバスサイクルをもはや消費せず、追加的な利用可能な性能がシステムに加えられる。最後に、この発明の P I C コアは、トランザクションフォワードアーキテクチャを用いてパイプライン方式をサポートし、より高い全システム性能を可能にする。これらのスーパーセット機能は、 F P G A 内の I P 性能を飛躍的に高める。

【 0 0 7 3 】

C . 信号 : P L B から P I C へのマッピング

上述の P L B 機能を提供するために、さまざまな信号が、システムのマスタ / スレーブ 20
と P L B との間で転送される。これらの P L B 信号は、以下に列挙されており、この発明に対するマッピングを示す。たとえば、表 1 は、マスタの観点からさまざまな P L B 信号を、これらの信号が基本的な C P U 内で必要とされるか、および、これらの信号が妥当な性能をもって F P G A 内で実現され得るかを列挙する。

【 0 0 7 4 】

【表 1】

表 1

信号名	方向	CPUが必要とされるか	FPGA内
PLB 要求修飾子			
SYS_plbClk	入力	Yes	Yes
SYS_plbReset	入力	Yes	Yes
PLB_MnAddrAck	入力	Yes	Yes
PLB_MnRearbitrate	入力	No	Yes (共有バス)
PLB_MnSSize	入力	No	No
PLB_MnBusy	入力	Yes	Yes
PLB_MnErr	入力	Yes	Yes
PLB_pendReq	入力	No	Yes (スタティック値)
PLB_pendPri	入力	No	Yes
PLB_reqPri	入力	No	Yes (スタティック値)
Mn_request	出力	Yes	Yes
Mn_priority	出力	No	Yes
Mn_busLock	出力	No	Yes
Mn_RNW	出力	Yes	Yes
Mn_BE	出力	Yes (非バースト)	Yes (非バースト または 固定長バースト)
Mn_size	出力	Yes (非バースト またはキャプチャ ライン)	Yes (非バースト、 固定長バースト、または キャプチャライン)
Mn_type	出力	No	No
Mn_Msize	出力	No	No
Mn_compress	出力	Yes	Yes
Mn_guarded	出力	Yes	Yes
Mn_ordered	出力	No	Yes
Mn_locker	出力	No	Yes
Mn_abort	出力	Yes	No
Mn_Abus	出力	Yes	Yes
PLB 書き込みデータバス			
PLB_MnWrDAck	入力	Yes	Yes
PLB_MnWrBTerm	入力	No	Yes
Mn_wrBurst	出力	No	No
Mn_wDBus	出力	Yes	Yes
PLB 読み出しデータバス			
PLB_MnRdAck	入力	Yes	Yes
PLB_MnRdBTerm	入力	No	Yes
PLB_MnRdWdAddr	入力	Yes	Yes
PLB_MnRdDBus	入力	Yes	Yes
Mn_rdBurst	出力	No	No
スレープセット信号			
SYS_plbClk2x	入力	No	Yes
PLB_MnWrAck	入力	No	Yes

【0075】

表 2 は、スレープの観点からさまざまな PLB 信号を、これらの信号が基本的な CPU 内で必要とされるか、および、これらの信号が妥当な性能をもって FPGA 内で実現され

10

20

30

40

50

得るかを列挙する。

【 0 0 7 6 】

【表 2】

表 2-1

信号名	方向	CPU内で必要とされる	FPGA内
PLB 転送修飾子			
SYS_plbClk	入力	Yes	Yes
SYS_plbReset	入力	Yes	Yes
PLB_PValid	入力	Yes	Yes
PLB_busLock	入力	No	Yes
PLB_pendReq	入力	No	Yes
PLB_pendPri	入力	No	Yes (スタティック値)
PLB_reqPri	入力	No	Yes (スタティック値)
PLB_masterID	入力	No	Yes
PLB_RNW	入力	Yes	Yes
PLB_BE	入力	Yes (非バースト)	Yes (非バースト または 固定長バースト)
PLB_size	入力	Yes (非バースト または キャッシュライン)	Yes (非バースト、 固定長バースト、 または キャッシュライン)
PLB_type	入力	No	No
PLB_Msize	入力	No	Yes (スタティック値)
PLB_compress	入力	Yes	Yes
PLB_guarded	入力	Yes	Yes
PLB_ordered	入力	No	Yes
PLB_lockErr	入力	No	Yes
PLB_abort	入力	No	Yes (スタティック値)
PLB_Abus	入力	Yes	Yes
Sl_addrAck	出力	Yes	Yes
Sl_wait	出力	No	No
Sl_Ssize	出力	No	No
Sl_rearbitrate	出力	No	Yes (共有バス)
Sl_Mbusy	出力	Yes	Yes
Sl_Merr	出力	Yes	Yes
PLB プラ パイプ			
PLB_SAVValid	入力	Yes (性能)	Yes
PLB_rdPrim	入力	Yes (性能)	Yes

【表 3】

表 2-2

PLB_wrPrim	入力	Yes (性能)	Yes
PLB 書き込みデータバス			
PLB_wrDBus	入力	Yes	Yes
PLB_wrBurst	入力	No	No
Sl_wDack	出力	Yes	Yes
Sl_wrComp	出力	Yes	Yes
Sl_wrBTerm	出力	No	Yes
PLB 読出しデータバス			
PLB_rdBurst	入力	No	No
PLB_Sl_rdDBus	出力	Yes	Yes
PLB_rdWdAddr	出力	Yes	Yes

10

【0078】

IV. FPG A 実現例

この発明の一実施例に従うと、プログラマブルインターフェイスコア (PIC) は、FPG A 上および FPG A 外の両方にある他の装置と中央処理装置 (CPU) との間で通信を提供する。この発明によって指定される上述の PLB 機能サブセットのために、PIC コアは、標準の FPG A リソース、つまり、ルックアップテーブル、マルチプレクサ、およびフリップフロップを有利に用いて、これらの機能を効率的に実現し得る。この発明は、これらのリソースにうまくマッピングされる論理機能およびパイプライン方式を活用する。

20

【0079】

A. 中央処理装置 (CPU)

図 4 は、FPG A 400 上の非プログラマブルリソースから形成される中央処理装置 (CPU) 401 を含む FPG A 400 を例示する。一実施例では、CPU 401 は、IBM のパワー PC 405 CPU を含む。しかしながら、この発明は、いずれの CPU にも同様に適用可能である。さらに、この発明は、いくつの数の CPU とともに用いられてもよい。具体的には、図 4 に例示されるアーキテクチャが、FPG A 上の各 CPU に対して繰返され得る。

30

【0080】

この実施例では、CPU 401 は、ブロックランダムアクセスメモリ (BRAM) 408 A および 408 B 等の複数の「オンチップ」メモリと直接通信し得る。各 CPU は、このようなオンチップメモリに対するプロプラエタリ接続およびそれ自体のポートを規定するため、これらの詳細は、ここで提供されない。

【0081】

CPU 401 は、任意の数の周辺モジュール、つまり、マスタモジュール 406 またはスレーブモジュール 404 によって CPU 401 に送られる「フラグ」を妨害および優先する割込コントローラ (INTC) 409 にも結合される。したがって、いずれかのスレーブモジュール 404 またはマスタモジュール 406 が、割込コントローラ 409 に結合され得る (線は図示せず)。これらのフラグは、CPU 401 からの即時アテンションを要求する周辺モジュールに関連した「イベント」を表す。この状態で、CPU 401 は、周辺モジュールの連続的なチェックを行なうことから解放され、CPU 401 が可能な限り効率的に動作し得ることが確実となる。たとえば、周辺モジュールからの 1 つのフラグは、CPU 401 に対して、ユーザがキーボード上の「入力」キーを押したことを知らせ得る。これらのフラグの生成は、当業者には周知であるため、ここで詳細に説明されない

40

50

。周辺モジュールによって生成される割込に应答するために、CPU 401は、PIC 402およびOPB 403を用いる（両方が以下で詳細に説明される）。

【0082】

なお、CPU 401は、FPGAファブリックに対するCPU 401の専用ピンアウトの結合を容易にする構造を含み得る。この発明は、異なる構造を有するさまざまなCPUとともに機能し得る。

【0083】

FPGA 400は、複数の高速デバイス410およびマスタ/スレーブモジュール404/406からCPU 401までのデータ転送を可能にする複数のデータ経路（図示せず）をさらに含む。したがって、図4に示される経路は、この発明に従ったトランザクション経路を示すことが意図される。

10

【0084】

B. プログラマブルインターフェイスコア（PIC）

この発明に従うと、（FPGA 400上およびチップ外の両方にある）他のデバイスとの高性能通信を提供するために、CPU 401は、プログラマブルインターフェイスコア（PIC）402を用いる。プログラマブルインターフェイスコア402は、名前が示すように、FPGAソフトウェアツール（図3のFPGAソフトウェアツール306参照）によって生成されるコンフィギュレーションビットを用いてFPGA 400上で実現される。1つの最適な実施例では、プログラマブルインターフェイスコア402は、クロスポイントスイッチを含み、クロスポイントスイッチは、（CPU 401または高速バスインターフェイス（HSBI）407等の）いずれかのマスタデバイスがいずれかのスレーブデバイス410と通信することを選択的に可能にする。さらに、クロスポイントスイッチでは、第1のトランザクションが進行している間に、別のマスタデバイスが、同時に別のスレーブデバイスと通信することができる。この状態で、この発明は、FPGA 400上およびチップ外の両方における、スレーブデバイスとマスタデバイスとの間の高性能インターフェイスを提供し得る。

20

【0085】

図4に示される実施例では、プログラマブルインターフェイスコア402は、4入力（マスタ）×4出力（スレーブ）（4×4）クロスポイントスイッチを用いて、マスタデバイスをスレーブデバイスに接続し得る。具体的には、マスタデバイスは、CPU 401の命令キャッシュユニット（ICU）と、CPU 401のデータキャッシュユニット（DCU）と、高速バスインターフェイス（HSBI）407と、OPBブリッジインモジュール（OPB/B I）とを含み得、スレーブデバイスは、DDR SDRAMコントローラデバイス410Aと、オフチップメモリ410B - 410Dのうちの1つと、オンチップメモリ410E（一実施例では、BRAM）と、OPBブリッジアウトモジュール（OPB/B O）とを含み得る。NのマスタデバイスとMのスレーブデバイスとを有する他の実施例では、プログラマブルインターフェイスコア402は、N×Mクロスポイントスイッチとして構成され得る。しかしながら、重要なことには、他の種類のスイッチと比較して増大するクロスポイントスイッチ接続性のために、クロスポイントスイッチの実現は、FPGA 400上の利用可能なリソースを最適化するように慎重に評価されなければならない。この評価は、以下で詳細に説明される。

30

40

【0086】

図5Aは、プログラマブルインターフェイスコアの一部として提供される、簡素化された4×4クロスポイントスイッチ500を例示し、クロスポイントスイッチ500は、4つのマスタデバイス501A - 501Dおよび4つのスレーブデバイス502A - 502Dを選択的に結合する。一実施例では、クロスポイントスイッチ400のサイズは、16×16スイッチにまで拡大され得る。なお、論理リソースによって、通例、クロスポイントスイッチのサイズが制限されるが、ルーティングリソースも制限的であり得る。したがって、クロスポイントスイッチの最大サイズは、FPGAに左右される。一実施例では、最大サイズは、PICを生成するために用いられるパラメータおよびN×Mサイズを考慮

50

に入れて計算され得る。クロスポイントスイッチ500は、トランザクションのアドレスおよびデータ経路のための2つの別個の構造を含む。実際のFPGAでは、図5B(500A)および図5C(500B)を参照して説明されるこれらの構造は、オーバーラップするプログラマブルリソース上に置かれ得る。他の実施例では、トランザクションのアドレスおよびデータ論理内で用いられる構造は、オーバーラップしないプログラマブルリソースから形成される。

【0087】

図5Bは、アドレス論理内のクロスポイントスイッチ(以下では、クロスポイントスイッチ500A)の一実現例のより詳細な図を示す。この実現例では、クロスポイントスイッチ500Aは、Nのアドレスデコーダ511(Nは、マスタデバイス501の数に等しい)と、Mのアクセスアービタ512(Mは、スレーブデバイス502の数に等しい)とを含む。マスタデバイス501A-501Dは、データキャッシュユニット(DCU)、命令キャッシュユニット(ICU)、高速バスインターフェイス(HSBI)、およびオンチップ周辺バスブリッジインモジュールOPB/BIを含む。スレーブデバイス502A-502Dは、DDRデバイス、スタティックランダムアクセスメモリ(SRAM)、ブロックRAM(BRAM)、およびOPBブリッジアウトマスタOPB/BOを含む。

【0088】

OPB/BIおよびOPB/BOモジュールに関して、CPU401(図4)がスレーブモジュール404またはマスタ/スレーブモジュール405と通信する必要がある場合(以下で説明)、プログラマブルインターフェイスコア(PIC)402内のOPBブリッジアウトモジュールOPB/BOが、CPUのコマンドをOPB403に対して駆動することが注目される。しかしながら、PIC402の観点からは、このOPB/BOモジュールは、スレーブデバイスとして機能する。対照的に、マスタモジュール406またはマスタ/スレーブモジュール405がデバイス410に対してコマンドを発行する場合、その要求は、プログラマブルインターフェイスコア402内のブリッジインモジュールOPB/BIによって受取られる。したがって、PIC402の観点からは、OPB/BIは、マスタとして機能するように見える。したがって、OPB/BOおよびOPB/BIは、マスタ機能とスレーブ機能との両方を有するプログラマブルインターフェイスコア502内のブリッジモジュールとして特徴付けられ得る。OPB/BOおよびOPB/BIは、図16および図17を参照しながらより詳細に説明される。

【0089】

クロスポイントスイッチ500A内では、アクセスデコーダ511は、そのそれぞれのマスタデバイス501からアドレスを受取り、次に、アドレスをデコードして、マスタデバイスがどのスレーブデバイス502にアクセスすることを望むかを判断する。デコーディングの後、アドレスデコーダ511は、要求されたスレーブデバイス502に関連したアクセスアービタ512にアドレスを送る。多数のマスタデバイス501が1つのスレーブデバイス502にアクセスすることを望み得ることが思い出されたい。したがって、アクセスアービタ512を用いて、要求を行なうマスタデバイス間でプライオリティを決定する。アクセスアービタ512によって用いられ得るいくつかの公知のアルゴリズムは、シングルレベルラウンドロビン、マルチレベルラウンドロビン、最長時間未使用法、およびプライオリティベースのアプローチ(OPBバスアービタ503Aを参照しながら以下でより詳細に説明)を含む。なお、トランザクションは読出動作であるのか、または書込動作であるかということ、およびデータ転送サイズを含む、マスタデバイス501からの追加的な情報が、デコードされたアドレスとともにスレーブデバイス502に転送される。

【0090】

クロスポイントスイッチ500A内でアドレスデコーダ511およびアクセスアービタ512を用いることによって、前のトランザクションが完了する前に、マスタデバイス501が別のトランザクションを開始することが可能になる。言い換えると、デコードされたアドレスがアクセスアービタ512に一旦提供されると、アドレスデコーダ511は、

10

20

30

40

50

次のアドレスをデコードし始め得る。この種のプロセスは、「トランザクションフォワード」と呼ばれる。この発明は、トランザクションフォワードを有利に用いて、クロスポイントスイッチの性能を高める。

【0091】

この発明の一実施例では、このトランザクションフォワード特徴は、（簡素化のために1つしか示されていない）多数のレジスタ513をアドレス論理内に含むことによって、さらに強化され得る。図5Bでは、これらのレジスタは、アドレスデコーダ511の前に、アドレスデコーダ511の後およびアクセスアービタ512の前に、および/または、アクセスアービタ512の後およびスレーブデバイス502の前に、置かれ得る。なお、追加的なレジスタが、アドレス論理内のいずれかのポイントに置かれ得る（しかしながら、各アドレス経路は、アドレス/制御経路において一貫したタイミングを保証するために、同じ場所にレジスタを有するべきである）。この発明では、これらのレジスタ513は、アドレスおよび制御情報のためにパイプライン方式を提供し得、したがって、他のトランザクションが処理されている間に、要求を行なうマスタが別のトランザクションを開始することが可能になる。したがって、この状態で、この発明のパイプライン方式は、システムの全アドレス/制御帯域幅を増大させることができ、多数のトランザクションのためのアドレス論理の速度を上げる。

【0092】

ある特定のスレーブデバイス502が、トランザクションに対して解放されているとアクセスアービタ512によって判断された後、そのアクセスアービタ512は、データ論理内で提供される構造と通信して、図5Cを参照しながら説明されるような適切な接続を生成する。図5Cは、データ論理のためのクロスポイントスイッチ（以下では、クロスポイントスイッチ500B）の一実現例のより詳細な図を示す。この実現例では、クロスポイントスイッチ500Bは、Nの書込マルチプレクサ514（Nは、書込動作において書込され得るスレーブデバイス502の数に等しい）と、Mの読出マルチプレクサ515（Mは、読出動作においてデータを受取り得るマスタデバイス501の数に等しい）とを含む。マルチプレクサ514A - 514Dおよび515A - 515Dは、アクセスアービタ512A - 512D（図5B）によって制御される。

【0093】

アドレス論理と同様に、クロスポイントスイッチ500Bは、レジスタ513によって提供されるパイプライン方式を含み得る。一実施例では、これらのレジスタ513は、マスタデバイス501と書込マルチプレクサ514との間に、および、スレーブデバイス502と読出マルチプレクサ515との間に置かれる。なお、追加的なレジスタがデータ論理内のいずれかのポイントに置かれ得る。しかしながら、各データ経路は、データ経路において一貫したタイミングを保証するために、同じ場所にレジスタを有するべきである。この発明では、これらのレジスタ513は、転送されるデータに対してパイプライン方式を提供し得、別の読出または書込トランザクションが処理されている間に、要求を行なうマスタまたはアクセスされるスレーブのいずれかが、別の読出または書込トランザクションを開始することが可能になる。したがって、この状態で、この発明のパイプライン方式は、システムの全データ帯域幅も増大させることができ、多数のトランザクションのデータ論理のスループットを上げる。

【0094】

上述のように、この発明は、アドレスおよびデータ論理の両方においてトランザクションフォワードを有利に用い得る。パイプライン方式によって、トランザクションフォワードの最適化が可能になる。具体的には、マスタデバイスは、スレーブデバイスと直接通信しないため、パイプラインレジスタが、アドレス/データ論理内の多数の場所で提供され得る。したがって、一旦マスタデバイスがアドレスを送り出し、スレーブから肯定応答を受取ると、マスタデバイスは、そのトランザクションに関連した情報の残り（制御とデータとの両方）をパイプライン処理し得る。一旦スレーブデータ肯定応答信号がマスタに到達すると、マスタは、スレーブが実際にデータを受取るのを待たずにデータを供給できる

10

20

30

40

50

点において、パイプライン方式は役立つ。この時点で、マスタデバイスは、別のトランザクションを開始できる。パイプライン化の量によって、同時にアクティブになり得るトランザクションの数が決定される。

【 0 0 9 5 】

なお、全システム性能に関して、レイテンシ（つまり、このパイプライン方式に関連した遅延）も考慮されなければならない。たとえば、一実施例では、レジスタは、クロック信号を必要とするフリップフロップを用いて実現され得る。このクロック信号は、アドレスデコードおよびアドレスアービタがそれらのそれぞれの機能を実行するために必要な時間に基づいてセットされ得る。なお、パイプライン方式は、周波数と直接相関関係がある。したがって、パイプライン化すればするほど、より高い周波数が提供される。しかしながら、周波数を増大させることは、また、レイテンシを望ましくないほど増大させる。したがって、以下にさらに詳細に説明されるように、この発明は、（パイプライン方式によって提供されるような）帯域幅とレイテンシとの間でバランスを確立する。

10

【 0 0 9 6 】

別の利点として、この発明は、各スレーブデバイス内で提供される論理を大きく減じることができる。具体的には、この発明に関連するマスタデバイスと通信するスレーブデバイスは、アドレスデコードするために必要とされる複雑な回路をなくすことができる。したがって、この発明は、スレーブデバイスの設計、実現、および製造の作業を簡素化する。

【 0 0 9 7 】

20

対照的に、共有バスシステムでは、マスタデバイスは、アドレスをすべてのスレーブデバイスに一斉送信し得る。したがって、このシステムでは、各スレーブデバイスは、アドレスをデコードし、アドレスがそのスレーブデバイスに関係するかを判断するための論理を含み得る。アドレスがデコードされた後、適切なスレーブデバイスが信号をマスタデバイスに送り返し、これが、次に、トランザクションに関する情報の残りを、識別されたスレーブデバイスに送り得る。

【 0 0 9 8 】

上の説明から理解され得るように、共有バスシステムは、システム内に含まれ得るパイプラインの量を大きく制限し得る。たとえば、一旦マスタデバイスがアドレスを送り出すと、すべてのスレーブデバイスがアドレスをデコードしなければならず、他の指定されていないスレーブデバイスが、他のマスタデバイスと他のトランザクションを進めることが妨げられる。したがって、先行技術の共有バスシステムは、本質的に、この発明のクロスポイントスイッチよりも低い性能を提供する。なお、（以下で説明される）ハイブリッドシステムでは、共有バス上のスレーブの数が制限され得る。この状態で、アドレスを一斉送信し、さらにはスレーブからの応答を待つ時間が、最小限にされ得る。言い換えると、高い帯域幅を必要とするデバイスは、共有バス構成に内在する余分なレイテンシを支え得るものと釣り合わされ得、ある特定のシステムが最適化される。

30

【 0 0 9 9 】

この発明のスイッチファブリックとして F P G A 内のクロスポイントスイッチを構成することによって、いくつかの利点が提供される。第 1 に、クロスポイントスイッチは、「非閉塞的」であり、入力は、出力に対する非輻輳経路が保証されることが確実である。この局面は、この発明のパイプライン化されたアーキテクチャにおいて特に有利である。第 2 に、F P G A リソースを用いて実現されるクロスポイントスイッチによって、接続が迅速に、選択的に変更されることが可能となる。なお、このような接続の変更は、したがって、経路の変更も、他のいずれの経路の接続にも影響を及ぼさない。第 3 に、クロスポイントスイッチは、プログラマブルに「トリミング」されて、F P G A 上で用いられるリソースを最小にし得る。

40

【 0 1 0 0 】

たとえば、図 5 D は、アドレス / 制御論理がトリミングされるクロスポイントスイッチ 5 0 0 A の一実施例を示す。この実施例では、ユーザは、あるアドレス / 制御論理が用い

50

られないことを決定した。なお、ソフトウェアは、どの論理がトリミングされ得るかをユーザが特定する助けとはなり得るが、このような決定をするためには、ユーザが通例、ソフトウェアに基本的なパラメータを提供しなければならない。この発明は、用いられない論理を除去し、その論理に関連するリソースがユーザの設計の他のコンポーネントのために用いられることを可能にする。代替的には、たとえユーザの設計の他のコンポーネントが、トリミングされた論理を用いない場合でも、ユーザの設計は縮小され得、おそらくは、より小さなFPGAが用いられるか、またはより多くのマスタまたはスレーブが同じサイズのFPGA内で実現されることさえも可能となる。図5Dでは、以下のアドレス/制御論理が除去される。すなわち、ICUからOPB/BO、HSBIからBRAM、HSBIからOPB/BO、OPB/BIからBRAM、およびOPB/BIからOPB/BOである。

10

【0101】

別の例として、図5Eは、データ論理がトリミングされるクロスポイントスイッチ500Bの一実施例を示す。この実施例では、ユーザは、あるデータ論理が用いられないことを決定した。この発明は、用いられない論理を有利に除去し得る。図5Eでは、以下の書込データ論理が除去される。すなわち、HSBIからBRAMおよびOPB/BO、ICUからいずれかのスレーブデバイス（つまり、ICUはリードオンリデバイス）、OPB/BIからBRAMおよびOPB/BOである。加えて、この実施例では、以下の読出論理も除去される。すなわち、BRAMからHSBIおよびOPB/BI、およびOPB/BOからHSBIおよびOPB/BIである。

20

【0102】

この発明に従うと、いずれかの経路（および、関連の論理）のトリミングによって、残りの回路が簡素化され得る。たとえば、図5Eでは、書込マルチプレクサ514Cおよび514Dの各々は、1つの入力のみを有する。したがって、これらのマルチプレクサは、たとえば、簡素なパストランジスタとして実現され得る。読出マルチプレクサ515Cおよび515D等の他のマルチプレクサも簡素化され得る。論理を簡素化することによって、システムの速度および他の性能パラメータが改善され得る。したがって、この発明に従ったプログラマブルリソースをトリミングする能力によって、FPGAの性能および面積において、著しい改善が提供され得る。

【0103】

30

この発明の一実施例では、クロスポイントスイッチ500A/500Bは、アドレス/制御論理がCLKで動作し、さらにはデータ論理がCLK2Xで動作するように、設計される。具体的には、スレーブデバイス502がトランザクションに利用可能であることが、アクセスアービタ512によって一旦判断されると、そのアクセスアービタ512は、適切な書込マルチプレクサ514または読出マルチプレクサ515を制御することによって、そのトランザクションのための必要な論理を生成し得る。したがって、この時点では、追加的な論理または経路変更が必要とされないため、書込または読出動作は、さらなる遅延を伴わずに進行することができる。したがって、一実施例では、アドレス/制御論理は、133MHzで動作し得、データ論理は、266MHzで動作し得る。

【0104】

40

重要なことには、デュアルクロックシステムでは、制御経路は、公称データクロックレートの1/2で動作して、そのタスクを達成するための優先的な時間をこの経路に提供する。さらに、1/2対1比が存在するため、データレイテンシは、実質的に減じられ、データ転送周波数が上昇し得る。

【0105】

なお、クロスポイントスイッチ500A/500Bは、1つ以上の特定の信号をマスタデバイス501に転送するための専用の追加的なラインを含み得る。これらの追加的なラインは、肯定応答信号を転送し得るが、転送に限定されているわけではない。たとえば、一実施例では、トランザクションフォワードは、要求を行なうマスタデバイス501に対してアクセスアービタ512により提供される肯定応答信号によって、トリガされ得る。

50

具体的には、制御信号およびデータのパイプライン処理は、要求を行なうマスタデバイス 501 が肯定応答信号を受け取った後に、開始される。

【0106】

C. マスタおよびスレーブモジュール

この発明に従うと、通例、先行技術のシステム内では周辺デバイスとして提供される、マスタモジュール 406 とスレーブモジュール 404 との両方が、ユーザのトップレベル設計の一部として、またはコアとして、FPGA 上に置かれ得る。この発明の一実施例では、FPGA 400 は、マスタとスレーブとの両方として働くハイブリッドモジュール 405 を含み得る。具体的には、ハイブリッドマスタ/スレーブモジュール 405 は、特定の時間期間中にトランザクションを開始し、オンチップ周辺バス (OPB) 403 を制御することができ、他の期間中は、トランザクションを受入れ、データを OPB バス 403 に提供し得る。たとえば、ネットワーキングアプリケーションでは、データパケットが、スレーブとしてのハイブリッドモジュール 405 によって受取られると、ハイブリッドモジュール 405 は、DMA を用いるマスタとしてパケットをメモリに送り得る。別の実施例では、マスタ機能は、スレーブモジュール内に組込まれる。

10

【0107】

有利なことには、この発明によって、パラメータ表示が容易にされるため、マスタおよびスレーブモジュールの数は、動的にさえも、変更され得る。したがって、この発明は、ユーザの設計を実現するために必要とされる FPGA 上のリソースのみを用いる。

【0108】

20

D. OPB バス

この発明に従うと、オンチップ周辺バス (OPB) 403 は、プログラマブルインターフェイスコア 402 とモジュール 404 / 406 との間の通信を容易にする接続インフラストラクチャを含む。OPB 403 は、複数のより低い性能の周辺デバイス、つまり、モジュール 404 / 406 をサポートする共有バスアーキテクチャを用いる。これらのモジュール 404 / 406 は、たとえば、シリアルポートおよびイーサネット (R) 接続を用い得る。一実施例では、モジュール 404 / 406 は、IBM のコアコネク (CoreConnect) (商標) OPB 仕様に従い得る。

【0109】

重要なことには、OPB 403 は、FPGA 400 上に提供されるプログラマブルルーティングリソースから形成され得る。具体的には、バスのラインがプログラマブルルーティングリソースから形成され得、スイッチング論理は、CLB 内のプログラマブルリソースから形成され得る。この状態で、OPB 403 は、FPGA 400 上のどこに位置付けられてもよい。さらに、OPB 403 は、いかなる数のモジュール 404 / 406 にも対応するように構成され得る。言い換えると、3つのマスタモジュール 406 および 1つのスレーブモジュール 404 のみが、ユーザによって FPGA 400 上に置かれる場合、この発明は、これらの 4つのモジュールのみのためのポートが作成されることを保証する。したがって、この発明に従うと、OPB 403 は、完全にパラメータ表示可能である。

30

【0110】

OPB 403 は、単一のマスタモジュール 406 がスレーブモジュール 404 またはマスタ/スレーブ 405 と通信することを可能にするスイッチング論理を含む。具体的には、共有バスアーキテクチャのために、他のマスタモジュール 406 またはマスタ/スレーブ 405 は、各アクティブトランザクション中に、それらのそれぞれのスレーブモジュール 404 とのそれらのトランザクションを開始することを待たなければならない。これらのトランザクションにプライオリティを割当ててするために、OPB アービタ 403A が提供される。

40

【0111】

この発明の一実施例に従うと、OPB アービタ 403A は、いずれかの種類のアービトレーション方式を提供するように構成され得、さらには FPGA 400 が動作しているときに動的に変化するようにさえも構成され得るコアである。なお、OPB 403 のように

50

、OPBアービタ403Aは、完全にパラメータ表示可能である。したがって、OPBアービタ403Aは、ユーザによって指定された数のマスタモジュール406をアービトレートするために必要なFPGA400上のプログラマブルリソースのみを用いる。一般に、OPBアービタ403Aは、予め構築されたアルゴリズムを用いて、どのマスタモジュール406がOPB403を制御し得るかということと、その制御の時間期間とを決定する。

【0112】

アービトレーションは、公平性とプライオリティという相反する目標を解決しなければならない。公平性は、1つのマスタモジュール406がOPB403を独占することを許可されるべきではないことを要求する。しかしながら、プライオリティは、特定の状況下では、予め定められたマスタモジュール406がより頻繁にOPB403を用いてタイムクリティカルな目標を達成すべきであることを要求する。OPBアービタ403Aによって用いられ得るいくつかの典型的なアルゴリズムは、シングルレベルラウンドロビン、マルチレベルラウンドロビン、最長時間未使用法、およびプライオリティベースのアプローチである。

10

【0113】

シングルレベルラウンドロビンアプローチでは、小さな時間単位、つまり、量子が定義される。(特定のマスタ306に関連する)すべてのランザクションが、巡回待ち行列内に置かれる。OPBアービタ403Aは、待ち行列に従い、マスタモジュールによるOPB403の使用を割当てて、1量子の時間間隔のためのランザクションを達成し得る。待ち行列内の最後のランザクションの後に、いずれかの新しいランザクションが加えられ得る。

20

【0114】

量子の終わりの前にランザクションが終了すると、マスタモジュール406は、自発的にOPB403を解除し得る。しかしながら、ランザクションが量子の終わりにおいても依然として動作している場合、マスタモジュール406は、プリエンプトされ得、ランザクションは、待ち行列の終わりの部分に加えられ得る。なお、別の実施例では、マスタモジュール406は、そのランザクションを完了することができる。このとき、OPBアービタ403Aは、OPB403を待ち行列内の次のランザクションに割当て得る。

30

【0115】

マルチレベルラウンドロビンアプローチでは、少なくとも2つの巡回待ち行列が形成される。たとえば、第1の待ち行列と第2の待ち行列とが形成されると仮定すると、OPB403を頻繁に用いるランザクションが、第1の待ち行列内に置かれ得、OPB403をそれほど頻繁に用いないランザクションが、第2の待ち行列内に置かれ得る。第2の待ち行列内のランザクションは、同じ待ち行列内に存在するならば、OPB403に対して等しいアクセスを有し得る。しかしながら、第2の待ち行列内のランザクションは、グループとして、第1の待ち行列内の各ランザクションと同等の、OPB403に対するアクセスを有し得る。言い換えると、第2の待ち行列のランザクションは、「スーパートランザクション」を効率的に形成し、スーパートランザクションは、第1の待ち行列内のランザクションのうちの1つであると考えられる。したがって、第1の待ち行列ランザクションのラウンドごとに、第2の待ち行列の1つのランザクションが実行され得る。このアプローチでは、量子の終わりの前にランザクションが終了すると、マスタモジュール406は、自発的にOPB403を解除し得る。しかしながら、ランザクションが量子の終わりにおいても依然として動作している場合は、マスタモジュール406は、プリエンプトされ得、ランザクションは、適切な待ち行列の終わりの部分に加えられ得る。

40

【0116】

最長時間未使用法では、任意の待ち行列が形成される。OPBアービタ403Aは、最初は待ち行列に従い、待ち行列内の次のマスタモジュール406がOPB403の制御権

50

を得ることを可能にする前に、各トランザクションを終了させ得る。しかしながら、O P Bアービタ4 0 3 Aが、待ち行列内において次ではないマスタモジュール4 0 6からバス制御の要求を受取った場合、O P Bアービタ4 0 3 Aは、最長時間未使用のO P B 4 0 3を有したマスタモジュール4 0 6に対して（実行されているトランザクションの完了後に）O P B 4 0 3の制御権を与え得る。

【0 1 1 7】

最後に、プライオリティベースのアプローチでは、O P Bアービタ4 0 3 Aは、マスタモジュール4 0 6によって実行される関連のトランザクションのプライオリティにのみ基づいて、バス制御を決定し得る。このアプローチでは、より高いプライオリティの各トランザクションが、より低いプライオリティのトランザクションが開始許可され得る前に、完了し得る。一実施例では、プライオリティベースのアプローチは、F P G A内の性能に最も大きな影響を及ぼすマスタに対してプライオリティを許可することを含む。

10

【0 1 1 8】

E．高速バスインターフェイス

この発明の一実施例に従うと、F P G A 4 0 0は、バス4 0 7 Aおよび4 0 7 Bを介して、プログラマブルインターフェイスコア4 0 2とオフチップデバイス（図示せず）との間で多くの量のデータを転送することのできる高速バスインターフェイス（H S B I）4 0 7を含む。例示的なオフチップデバイスは、ギガビットI / O、高速I / O、P C I - X、または他の公知の高性能プロトコルを用いて、インターフェイス4 0 7と通信し得る。このインターフェイス4 0 7は、コアの形で提供され得る。なお、転送されるデータは、（フラッシュメモリ、S R A Mメモリ、および／または別の種類のメモリ等の）オフチップメモリまたは（B R A M等の）オンチップメモリから、またはそこへと、転送され得る。

20

【0 1 1 9】

一実施例では、H S B I 4 0 7は、バス4 0 7 Cを通して非データ移動のためにアドレス指定され得る。これらのバスは、C P U 4 0 1または他のバスマスタが、H S B I 4 0 7の内部レジスタセットに対する局所的なアクセスを有することを可能にする。たとえば、H S B I 4 0 7は、I N T C 4 0 9に対する割込を生成して、それがデータパケットを受取り、さらにはそれを外部のD D Rメモリデバイスに転送することを完了したことを示し得た。C P U 4 0 1は、割込に応答するとき、バス4 0 7 Aを用いてH S B I 4 0 7と対話（talk）して、サービスに対するその要求が何についてであるかを把握し得る。しかしながら、バス4 0 7 Aを用いる行為は、いずれの進行中のデータ移動動作も割込まれる必要があり得ることを意味する。したがって、一実施例では、テリアリ（teritary）バス4 0 7 Cセットが、O P B 4 0 3に対するマスタおよびスレーブ動作の両方のために存在する。バス4 0 7 Cを用いることにより、C P U 4 0 1は、バス4 0 7 A上のデータ移動に影響を与えずにH S B I 4 0 7と対話できる。さらに、H S B I 4 0 7がデータをデバイス4 0 4 Cに移動させたい場合、それは、まずP I C 4 0 2にアクセスする必要なしに、および／または、それをメモリに記憶することなく、それを行ない得る。むしろ、H S B I 4 0 7は、O P B 4 0 3上でトランザクションを開始し、データを直接転送することができ、データをダブルバッファする必要性をなくし、システム全体の著しい性能向上に貢献する。

30

40

【0 1 2 0】

V．F P G A実現例：詳細

A．P I Cおよびデバイス：ポート、タイミング、および実施例

マスタデバイスおよびスレーブデバイスは、よりモジュラな設計を可能にする包括的インターフェイスを用いる。具体的には、これらのモジュールは、プログラマブルインターフェイスコアに「プラグイン」され得る包括的なポート（以下で詳細に説明）を用いる。ここでの説明では、用語「ポート」は、マスタ／スレーブデバイスとプログラマブルインターフェイスコアとの間に提供される信号セットを含み得る。これらの包括的ポートのため、いかなる数のマスタデバイスおよびスレーブデバイスが、いずれの組合せで、この発

50

明に含まれてもよい。一実施例では、NのマスタデバイスとMのスレーブデバイスとが、プログラマブルインターフェイスコアに結合され、NおよびMは、1以上の整数である（いくつかの事例では、同じ整数である）。

【0121】

なお、この発明では、より多くの機能を提供する複雑なプロトコルとより速く実行され得る簡潔なプロトコルとの間で、正しいバランスを選ぶことの重要性が認識される。ここで詳細に説明される実施例では、比較的簡素なプロトコルが用いられ、依然として大部分のタスクを達成しながらも、非常に速く実行される。

【0122】

1. マスタデバイスとPICとの間の通信

図6は、いずれかのマスタデバイスとプログラマブルインターフェイスコアとの間で転送され得る典型的な信号を例示する。この転送を容易にするために、マスタデバイスは、アドレスポート601A、書込データポート602A、および読出データポート603Aを有するインターフェイス600Aを含む。同様の構成で、プログラマブルインターフェイスコアは、アドレスポート601B、書込データポート602B、および読出データポート603Bを有するPICインターフェイス600Bを含む。一実施例では、ポート間で転送される各信号（または、信号セット）は、関連のライン/バスを有し、その信号の効率的な転送を容易にする。なお、ここでの説明では、信号の名称は、関連のライン/バスも指し得る。以下に説明される実施例に従うと、標準のPLBプロトコルが制限され、より少ない機能が提供されるが、FPGAがずっと速く動作することが可能になる。

【0123】

a. マスタのためのアドレスポート

アドレスポート601Aは、以下の信号を出力し得る。

【0124】

M2PIC__AddrReq

このステータス信号は、マスタがデータ転送を要求することを望むことを表す。なお、M2PIC__AddrReqがアクティブである場合、他の信号、具体的には、M2PIC__Addr[31:0]、M2PIC__AddrQual、およびM2PIC__RdWr_nも有効でなければならない。この発明の一実施例では、論理1信号（1）が、要求はペンディングであることを表し、論理0（0）信号が、1つの要求もペンディングでないことを表す。この信号は、中位のタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

【0125】

M2PIC__Addr[0:31]

この32ビット信号は、アドレス、つまり、要求される転送の、メモリスペース内の特定のバイトを表す。上述のように、この信号は、M2PIC__AddrReqが有効である同じサイクル上で有効でなければならない。このバスは、クリティカルバス上にある。なぜならば、それは、プログラマブルインターフェイスコアのためのアドレスデコード（たとえば、図5Bのアドレスデコード511参照）に信号を与えるためである。したがって、一実施例では、上位ビットのみが実際には比較され、下位のビットは、中位のタイミングを有するとして扱われ得る。具体的には、上位ビットは、スレーブのアドレスを識別し、下位ビットは、スレーブ内の目標メモリを識別する。したがって、スレーブの識別が本発明のアービトレーションに先行し（たとえば、図5Bのアクセスアービタ512を参照）、いずれかの1つの時間期間中に、1つのトランザクションのみがスレーブ内でアクティブになり得るため、上位ビットの読出しは、遅いタイミングとして特徴づけられるべきであり、スレーブ内の実際の記憶場所は、中位のタイミングとして特徴づけられ得る。

【0126】

なお、要求されるデータ転送は、好ましくは、転送サイズに対してアドレスアラインされる。たとえば、8ワード転送では、アドレスは8ワード境界上にあるべきである。言い換えると、2ワード転送は、論理アドレス0、2、4等で始まり得、4ワード転送は、論

10

20

30

40

50

理アドレス 0、4、8 等で始まり得、8 ワード転送は、論理アドレス 0、8、16 等で始まり得る（例外として、目標ワード第 1 転送(target word first transfer)を特定する場合、アラインされていないアドレスが与えられ得る）。アライメントは、DDR SDRAM のようなデバイスのために必要とされる。なぜならば、これらのデバイスは、アラインされたブロック転送を実行するように設計されているためである。この信号は、CLK のクロックドメインを有する。

【0127】

M2PIC__RdWr__n

この信号は、アドレス要求信号(M2PIC__AddrReq)に付随するための読出または書込フラグを提供する。上述のように、この信号は、M2PIC__AddrReq が有効である同じサイクル上で有効でなければならない。一実施例では、論理 1 信号(1)が、読出動作が要求されることを表し、論理 0 (0) 信号が、書込動作が要求されることを表す。この信号は、中位のタイミングの信号として特徴づけられ、CLK のクロックドメインを有する。

【0128】

M2PIC__AddrQual[16:0]

このアドレス転送修飾子信号は、プロプラエタリ CPU、一実施例では、IBM のパワー PC 405 に特有である。上述のように、この信号は、M2PIC__AddrReq が有効である同じサイクル上で有効でなければならない。これらの信号は、プログラマブルインターフェイスコアによってデコードまたは処理されず、適切なスレーブデバイスへとルーティングされるだけである。この信号は、中位のタイミングの信号として特徴づけられ、CLK のクロックドメインを有する。この信号のためのビット定義は、以下のとおりである。

【0129】

【表 4】

ビット	名称	記述
[16]	キャッシュ可能な属性	データのキャッシュ可能性 (未使用の場合、0 に設定)
[15]	ガード付き属性	ガードデータ (未使用の場合、0 に設定)
[14]	ユーザによって定義される属性	ユーザによって定義される属性 (未使用の場合、0 に設定)
[13]	ライトスルー	ライトスルー属性 (未使用の場合、0 に設定)
[12:11]	プライオリティ	転送要求プライオリティ番号が大きいほど、要求は重要である。この情報は、現在、スレーブアービトレーション論理内で用いられない。 (00 への設定を推奨)
[10:8]	転送サイズ	000=1 ワード(1-4 バイト) - 32 ビット 000=2 ワード(1-8 バイト) - 64 ビット 001=4 ワードバースト 010=8 ワードバースト 011=16 ワードバースト 100=32 ワードバースト 101-111=リザーブされる
[7:0]	シングルビート転送 バイトイネーブル	これらは、シングルデータビート転送中のバイトイネーブルである(転送サイズ=000)。ビット [7:4] は、32 ビット PIC システム内で定義されない

10

20

30

40

50

【0130】

なお、32ビットシステムでは、ユーザは、32ワードバーストモードをサポートすることを望まない場合もある。なぜならば、これは、FPGAプリミティブに効率的にマッピングしない構造を必要とし得るためである。

【0131】

バイトイネーブルがシングルビート（つまり、1または2ワード）転送に提供されて、CPUのようなデバイスが、バイトおよびハーフワード読出のようなより小さなサイズの読出を実行することを可能にする。この機能は、たとえば、（図16および図17を参照しながら詳細に説明される）OPBブリッジ内で特に重要である。なぜならば、いくつかのOPBデバイス（特に、スレーブ）は、バイト長レジスタのみを有し得るためである。DRAMまたはSRAM等のメモリデバイスにとっては、たとえばバイトのみが必要とされている場合でも、ワードまたはダブルワードで読出すことに問題はなく、これらのデバイスは、シングルビート転送バイトイネーブルを無視し得る。これらのバイトイネーブルビットは、情報がデータ経路バイトイネーブルに対して冗長ではあるが、書込に対しても有効である。

10

【0132】

アドレスポート601Aは、以下の信号を受取ることもできる。

【0133】

PIC2M__AddrAck

（プログラマブルインターフェイスコアからの）このストロブ信号は、現在のアドレスを肯定応答し、もしあれば、次のアドレス要求をフェッチする。PIC2M__AddrAckがアサートされた後のサイクル上で、M2PIC__AddrReq、M2PIC__Addr、M2PIC__RdWr_n、およびM2PIC__AddrQualが更新される。この信号は、アドレス要求当たり、1CLKサイクルだけアサートされる。この信号は、遅いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

20

【0134】

PIC2M__Busy

この信号は、要求しているマスタに対して、そのトランザクションのうちの1つがスレーブ内で進行していることを伝える。この信号は、プログラマブルインターフェイスコアがそのPIC2M__AddrAckを発行した後に、ハイとなり、スレーブが（以下で詳細に説明される）そのS2PIC__RdXferDoneまたはS2PIC__WrXferDone信号をアサートしたときにローに戻って、データ転送がスレーブによって完了したことを表す。一実施例では、PIC2M__Busy信号およびマスタのM2PIC__AddrReq信号が、ORゲートに提供される。この状態で、マスタは、それがプログラマブルインターフェイスコア上のすべてのそのトランザクションをいつ完了したかを判断し得る。この信号は、“Sync”または“EIEIO”（エンフォース命令実行インオーダ（Enforce Instruction Execution In Order））命令をCPU内でサポートする。一実施例では、論理1は、プログラマブルインターフェイスコアがデータ転送でビジーであることを示し、論理0は、進行中のトランザクションが存在しないことを示す。なお、スレーブは、それがトランザクションを内部で完了していない場合、それがS2PIC__RdXferDone/S2PIC__WrXferDoneをアサートした後、信号PIC2M__Busyをアクティブに保持することが許可されている。この信号は、中位のタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

30

40

【0135】

PIC2M__Err

この信号は、マスタに対して、その進行中のトランザクションのうちの1つがエラーを生じさせたことを伝える。読出エラーに対しては、この信号は、（以下で説明される）PIC2M__RdDataAckとともにアクティブになり得る。書込エラーに対しては、この信号は、プログラマブルインターフェイスコア上のトランザクション中またはその後に、アサートされ得る。より具体的には、PIC2M__Busy信号がアクティブなまま

50

である限り、書込トランザクションがプログラマブルインターフェイスコア上で完了した後に、エラー信号がアサートされ得るのみである。この時点で、要求を行なっているマスタは、エラーレジスタ、つまり、スレーブのBEARおよびBESRを調べて、エラーは何であったのかを判断し得る。この信号は、遅いタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0136】

b. マスタのための書込ポート

この発明に従うと、マスタは、それが転送要求している全書込データを、それが供給できることを確信していなければならない。データフロー制御信号は存在しないため、マスタは、転送されているデータの流れを遅くする、または中断することはできない。しかしながら、スレーブデバイスは、データ肯定応答信号（つまり、PIC2M_WrDataAck）を遅らせる(hold off)ことによって、データ転送間で待機状態を強制することができる。これらの制約の観点から、マスタ書込ポート602Aは、以下の信号を出力し得る。

【0137】

M2PIC_Data

この信号は、書込転送データを提供する。一般に、転送されるべき書込データの第1の部分は、対応のアドレス要求がスレーブに到達する機会を持つときまでに、利用可能である。一実施例では、アドレスデータがプログラマブルインターフェイスコアをパイプライン方式で通る(pipeline through)ために2つのCLK（または、4つのCLKX2）サイクルのレイテンシが存在する。したがって、アドレス要求が発行された後、書込データは、2つのCLKサイクルに対して準備ができていべきである。なお、スレーブが書込に対して所与の最小応答時間を有するカスタムシステムでは、書込データのアベイラビリティは、この量だけ遅延され得る。

【0138】

一実施例では、書込転送データは、64ビットを含む。しかしながら、他の実施例では、より少ない、またはより多くのビットが用いられ得る。いずれの場合も、書込転送データを運ぶバスは、最適な幅にまでスケールダウン/アップされ得る。なお、このバスは、プログラマブルインターフェイスコアにおいて長い距離に及び得るため、そのタイミングが重要である。したがって、この信号は、遅いタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0139】

M2PIC_DataBE[7:0]

この書込転送データバイトイネーブル信号は、M2PIC_Data信号と同じタイミングを有し、バイトイネーブル修飾子として書込データに付随する。一実施例では、インターリーブドビットまたはオールビットオフを含むいずれかのバイトイネーブルパターンが許容される。たとえば、高速マスタは、16ワードバースト転送を開始し、第1のデータ転送上ですべてのバイトイネーブルをオフにして、3動作ではなく1動作で14ワードバーストを効率的に実行し得る（8ワードバースト+4ワードバースト+1ダブルワード）。この信号は、32ビットPICシステム内では4ビット幅であり得る。この信号は、遅いタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0140】

M2PIC_DataQual[N:0]

この信号は、追加的な情報が書込データおよび/または書込バイトイネーブルとともに転送されることを可能にする。たとえば、一実施例では、パリティまたはエラー訂正コードビットが、書込データとともに転送され得る。なお、各M2PIC_DataQual信号は、各データとともに変化し得る。この信号は、遅いタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0141】

PIC2M_WrDataAck[7:0]

10

20

30

40

50

このストローブは、書込データを肯定応答し、次のデータ部分が転送されることを可能にする。P I C 2 M _ W r D a t a A c k 信号がアサートされた後のサイクル上では、もしあれば、次のデータが駆動されなければならない。各 P I C 2 M _ W r D a t a A c k ビット（この実施例では、合計 8 ビット）は、その対応のバイトレーンのデータを肯定応答する。この様態で、より小さな幅またはより遅いスレーブが、一度にいくつかのバイトレーンを肯定応答できる。しかしながら、たとえバイトイネーブルビットがそのバイトレーンのためにセットされていなくても、トランザクションは、すべてのバイトレーンに対して肯定応答される。この様態で、マスタは、データが適切に書込まれたことを知る。P I C 2 M _ W r D a t a A c k 信号の動的な性質は有利である。なぜならば、それによって、データが転送において連続しないことが可能となるためである。一実施例では、P I C 2 M _ W r D a t a A c k ビットは、増加するバイトアドレスの順序で（つまり、システムに応じて、ビット [7] から [0]、または [0] から [7] で）サイクルする。32 ビットシステムでは、この信号は、4 ビット幅であり得る。この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

【 0 1 4 2 】

c . マスタのための読出ポート

読出動作では、マスタデバイスは、それが転送要求している全データを、それが受取ることができることを確信していなければならない。データフロー制御信号は存在しないため、マスタデバイスは、転送されているデータの流れを遅くする、または中断することはできない。しかしながら、スレーブデバイスは、データ肯定応答信号（つまり、P I C 2 M _ R d D a t a A c k ）を遅らせる (hold off) ことによって、データ転送間で待機状態を強制することができる。

【 0 1 4 3 】

読出ポート 6 0 3 A は、以下の信号を受取り得る。

【 0 1 4 4 】

P I C 2 M _ D a t a

この信号は、転送されるべき読出データを含む。一実施例では、これは、64 ビットのデータを含む。別の実施例では、より少ない、またはより多くのビットが転送され得る。いずれの場合にも、この信号バスを転送しているバスは、適切なサイズにスケールダウン / アップされ得る。このバスは、プログラマブルインターフェイスコアにおいて長い距離に及び得るため、そのタイミングは、重要である。したがって、この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

【 0 1 4 5 】

P I C 2 M _ D a t a Q u a l

この信号は、読出データ転送修飾子を指す。一実施例では、ビット定義は、以下のとおりである。

【 0 1 4 6 】

【表 5】

ビット	名称	記述
[4:0]	読出ワードアドレス	キャッシュライン転送内のどのワードまたはダブルワードが送られているのかを特定する。ビット [0] は、64 ビットデータ経路を備えたシステム内では常に 0 である。

【 0 1 4 7 】

なお、一実施例では、C P U は、サイズにおいて最大 32 ワードの転送を要求し得る。したがって、5 ビットの「読出ワードアドレス」が、C P U をサポートするために必要とされる（論理的に、この信号は、32 ワードよりも大きな転送を定義できない）。この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

【 0 1 4 8 】

P I C 2 M _ R d D a t a A c k [7 : 0]

この読出ストローブ信号は、読出データが受取られることを許可する。各 P I C 2 M _ R d D a t a A c k ビットは、その対応のバイトレーンのためのデータを個別に肯定応答し、より小さな幅またはより遅いスレーブが一度に数個のバイトレーンを肯定応答することを可能にする。スレーブは、たとえバイトイネーブルビットがそのバイトレーンに対してセットされていなくても、すべてのバイトレーンが肯定応答されることを保証する。加えて、増加するバイトアドレスの順序で（これは、システムがビッグエンディアンまたはリトルエンディアンであるかに応じて、ビット [7] から [0]、または [0] から [7] になり得る）、スレーブは、P I C 2 M _ R d D a t a A c k ビットをサイクルし得る。結果として、32ビットPICシステムでは、この信号は、4ビット幅になり得る。この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

10

【 0 1 4 9 】

2 . マスタデバイスのタイミング

図7は、マスタデバイスが、8ワード書込動作を生成し、次に8ワード読出動作を生成するタイミング図を示す。書込動作では、マスタは、時間 t 1 においてアドレス要求 (m _ a d d r _ r e q) をアサートする。同時に、マスタは、トランザクションが書込動作であることを示し (m _ a d d r _ r _ w n がローに遷移する)、書込アドレス (m _ a d d r [3 1 : 0]) および転送サイズならびに他のいずれかの修飾子 (m _ a d d r _ q u a l []) を送る。

20

【 0 1 5 0 】

1クロック (c l k) サイクルのレイテンシ後（レイテンシは、他の実施例では異なり得る）、マスタは、時間 t 3 においてプログラマブルインターフェイスコアからアドレス肯定応答 (m _ a d d r _ a c k) を受取り、マスタが時間 t 5 において別のトランザクションを開始することを可能にする。なお、時間 t 1 において、マスタは、書込データ (D 1 w) を送るが、時間 t 3 においてのアドレス肯定応答 (m _ a d d r _ a c k) の受取り後まで待機してから、時間 t 3、t 4、および t 5 において書込データの残り (D 2 w、D 3 w、および D 4 w) を引き続き送る (1 D D R クロック (c l k 2 x) サイクルのレイテンシ)。 (この実施例では、ワードは、32ビットに等しい。したがって、64ビットデータ経路では、4つの64ビット転送のみが必要とされる。)

30

図7では、読出動作が書込動作の直後に続く。したがって、アドレス要求 (m _ a d d r _ r e q) は、論理1のままである。しかしながら、マスタは、トランザクションが読出動作であることを示さなければならない。したがって、m _ a d d r _ r _ w n は、時間 t 5 においてハイに遷移する。この発明の一実施例のパイプライン方式に従うと、時間 t 5 において、つまり、書込動作が時間 t 7 で完了する前に、マスタは、読出アドレス (m _ a d d r [3 1 : 0]) およびアドレス修飾子 (m _ a d d r _ q u a l []) を送り得る。マスタデバイスは、時間 t 8 で開始し、かつ時間 t 12 で終了する読出修飾子 (Q 1 r、Q 2 r、Q 3 r、および Q 4 r) および読出データ (D 1 r、D 2 r、D 3 r、および D 4 r) を受取る。

40

【 0 1 5 1 】

3 . スレーブデバイスと P I C との間の通信

スレーブデバイスは、データを転送するためのコマンドを受取り、マスタデバイスが要求する読出動作または書込動作を実行する。図8は、スレーブデバイスとプログラマブルインターフェイスコアとの間で転送され得る典型的な信号を示す。この転送を容易にするために、スレーブデバイスは、アドレスポート 801A、書込データポート 802A、および読出データポート 803A を有するインターフェイス 800A を含む。同様の構成で、プログラマブルインターフェイスコアは、アドレスポート 801B、書込データポート 802B、および読出データポート 803B を有する P I C インターフェイス 800B を含む。なお、いくつかの信号は、マスタポートのためのそれら (図6) と類似しており、

50

それらの信号の方向が逆になっている。

【0152】

a. スレーブのためのアドレスポート

アドレスポート801Aは、以下の信号を受取り得る。

【0153】

PIC2S__AddrReq

このステータス信号は、マスタからのペンディングアドレス要求が存在するかを示す。

PIC2S__AddrReqがアクティブであるとき、アドレス(PIC2S__Addr)およびアドレス修飾子(PIC2S__AddrQual、PIC2S__RdWr_n)も有効でなければならない。一実施例では、論理1は、要求がペンディングであることを示し、論理0は、要求がペンディングでないことを示す。この信号は、早いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

10

【0154】

PIC2S__RdWr_n

アドレス要求信号に付随するこの信号は、要求が読出動作であるか、または書込動作であるかを識別する。この信号は、PIC2S__AddrReqが有効である同じサイクル上で有効でなければならない。一実施例では、論理1は、READ動作を示し、論理0は、WRITE動作を示す。この信号は、早いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

【0155】

PIC2S__Addr[31:0]

この信号は、要求される転送のアドレスを提供する。具体的には、アドレス値は、メモリスペース内の特定のバイトを示し、PIC2S__AddrReqが有効である同じサイクル上で有効でなければならない。なお、マスタは、データ転送のサイズに関してアラインされたアドレス値を提供する。この信号は、早いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

20

【0156】

PIC2S__AddrQual[20:0]

このアドレス転送修飾子信号は、プロプラエタリCPU、一実施例では、IBMのパワーPC405に特有である。上述のように、このアドレス修飾子情報は、PIC2S__AddrReq信号が有効である同じサイクル上で有効でなければならない。この信号は、早いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。ビット定義は、以下のとおりである。

30

【0157】

【表 6】

ビット	名称	記述
[20:17]	マスタ ID	この信号は、現在スレーブにアクセスしているマスタの ID 番号を特定する。この情報は、たとえば、性能の最適化、または特定のマスタのための挙動の修正に有用であり得る。マスタの場所はシステム固有であるため、この信号が用いられる場合には、ユーザは、互換性の問題を考慮すべきである。
[16]	キャッシュ可能な属性	データのキャッシュ可能性 (未使用の場合、0 に設定)
[15]	ガード付き属性	ガードデータ (未使用の場合、0 に設定)
[14]	ユーザによって定義される属性	ユーザによって定義される属性 (未使用の場合、0 に設定)
[13]	ライトスルー	ライトスルー属性 (未使用の場合、0 に設定)
[12:11]	プライオリティ	この信号は、転送要求プライオリティを提供する。一実施例では、番号が大きいほど、要求は重要である。
[10:8]	転送サイズ	000=1 ワード(1-4 バイト、32 ビットシステム) 000=1 ダブルワード(1-8 バイト、64 ビットシステム) 001=4 ワードバースト 010=8 ワードバースト 011=16 ワードバースト 100=32 ワードバースト 101-111=リザーブされる
[7:0]	シングルビット転送 バイトイネーブル	これらは、シングルデータビット転送中に提供されるバイトイネーブルである(転送サイズ=000)。ビット[7:4]は、32 ビットシステム内では定義されない

10

20

30

【0158】

バイトイネーブルがシングルビット転送のために提供されて、CPU のようなデバイスが、バイトおよびハーフワード読出のようなより小さなサイズの読出を実行することを可能にする。これは、たとえば、OPBブリッジにおいて特に重要である。なぜならば、いくつかの OPB デバイスは、バイト長レジスタのみを有し得るためである。DRAM または SRAM 等のメモリデバイスにとっては、たとえばバイトのみが必要とされている場合でも、ワードまたはダブルワードで読出すことに問題はなく、これらのデバイスは、シングルビット転送バイトイネーブルを無視し得る。これらのバイトイネーブルビットは、たとえば情報がデータ経路バイトイネーブルに対して冗長であっても、書込に対して有効である。

40

【0159】

アドレスポート 801A は、以下の信号を出力することもできる。

【0160】

S2PIC__AddrAck

このストロブ信号は、アドレスを肯定応答し、もしあれば、次のアドレス要求を取出す。S2PIC__AddrAck 信号がアサートされた後のサイクル上では、PIC2S__AddrReq 信号、PIC2S__Addr 信号、PIC2S__RdWr__n 信号、お

50

よび `P I C 2 S _ A d d r Q u a l` 信号が、もしあれば、次のアドレス要求に更新される。このストローブ信号は、アドレス要求当り 1 C L K サイクルに対してアサートされるのみである。この信号は、早いタイミングの信号として特徴づけられ、C L K のクロックドメインを有する。

【 0 1 6 1 】

アドレスパイプライン方式をサポートすることを望むスレーブは、この発明のコアが次のアドレスを取出すことができるように、アドレス情報を迅速に肯定応答し得る。第 1 の要求に関連するデータ転送をスレーブが完了するまでは、スレーブは、第 2 のアドレス要求を肯定応答することはできない。言い換えると、スレーブは、パイプラインアドレス要求を用いて次の転送をセットアップすることはできるが、スレーブは、時期尚早に第 2 のアドレスを肯定応答してはならない。

10

【 0 1 6 2 】

この一般的なルールに対する 1 つの例外は、第 2 の要求が異なるデータポートを用いる場合、スレーブは、迅速に第 2 の要求を肯定応答してもよいことである。たとえば、スレーブは、読出転送を肯定応答し、次に、書込転送を肯定応答し、次に、第 3 のアドレスを見ることができる。

【 0 1 6 3 】

`S 2 P I C _ B u s y [1 5 : 0]`

この信号は、マスタに対して、そのトランザクションのうちの 1 つがスレーブ内で進行中であることを伝える。具体的には、スレーブは、トランザクションを開始した対応のマスタまでビジー信号を駆動して戻すことを担う。この発明の相互接続コアは、スレーブが（以下で説明される）`S 2 P I C _ R d X f e r D o n e` または `S 2 P I C _ W r X f e r D o n e` 信号を発行する時まで、マスタに対してこのビジー信号を自動的にアサートし得る。代替的には、スレーブは、`S 2 P I C _ W r / R d X f e r D o n e` 信号の後にビジー信号を駆動し続けて、トランザクションがスレーブ内で完了していないこと（つまり、トランザクションが依然として別のバス上でペンディングであるか、または依然としてスレーブ内で待ち行列を作っていること）をマスタに対して伝え得る。この信号は、“`S y n c`” または “`E I E I O`”（エンフォース命令実行インオーダ）命令を C P U 内でサポートする。一実施例では、論理 1 は、スレーブが依然としてデータ転送にビジーであることを示し、論理 0 は、スレーブが進行中のトランザクションを有さないことを示す。この信号は、中位のタイミングの信号として特徴づけられ、C L K のクロックドメインを有する。

20

30

【 0 1 6 4 】

`S 2 P I C _ E r r [1 5 : 0]`

この信号は、マスタに対して、その進行中のトランザクションのうちの 1 つがエラーを引起こしたことを伝える。具体的には、スレーブは、トランザクションを開始した対応のマスタに対して `E r r` 信号を駆動して戻すことを担う。読出エラーでは、この信号は、`S 2 P I C _ R d D a t a A c k` 信号とともにアクティブになり得る。書込エラーでは、この信号は、（`S 2 P I C _ B u s y` 信号がアクティブのままであるならば）コア上でのトランザクション中または後の任意のときにアサートされ得る。エラーを生成し得るスレーブは、（`B E A R` および `B E S R` 等の）エラーレジスタを含み得、マスタがエラーの原因を見つけ出すことを可能にする。この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

40

【 0 1 6 5 】

b. スレーブのための書込ポート

書込データポート 8 0 2 A は、以下の信号を受取り得る。

【 0 1 6 6 】

`P I C 2 S _ W r D a t a R e q`

スレーブは、書込データがマスタから転送されることを要求し始める前に、この信号がアクティブになるのを待つ。マスタは、マスタが書込むことを望む全データをそれが供給

50

できることを保証しなければならないため、この信号は、データ転送の途中で変更されることはない。この信号がアクティブになった後、それは、スレーブが `S2PIC_WrXferDone` 信号をアサートするまで、ハイのままである。スレーブは、`S2PIC_WrDataAck` 信号をアサートし、`PIC2S_WrDataReq` 信号がアクティブになる同じサイクル上でデータを転送し始め得る。この信号は、早いタイミングの信号として特徴づけられ、`CLK` のクロックドメインを有する。

【0167】

`PIC2S_Data[63:0]`

この信号は、書込転送データを含む。なお、64ビットが示されているが、このバスは、32ビット幅にまでスケールダウンされて、32ビットシステム内で用いられ得る。このバスは、コアにおいて長い距離に及び得るため、そのタイミングが重要である。この信号は、中位のタイミングの信号として特徴づけられ、`CLK2X` のクロックドメインを有する。

10

【0168】

`PIC2S_DataBE[7:0]`

この信号は、`PIC2S_Data` 信号と同じタイミングを有し、バイトイネーブル修飾子として書込データに付随する。「インターリーブド」ビットまたは「オールビットオフ」を含むすべてのバイトイネーブルパターンが許容される。なお、32ビットシステムでは、このバスは、4ビット幅にまでスケールダウンされ得る。書込データのように、このバスも、コアにおいて長い距離に及び得るため、そのタイミングも重要である。この信号は、中位のタイミングの信号として特徴づけられ、`CLK2X` のクロックドメインを有する。

20

【0169】

`PIC2S_DataQual[N:0]`

この信号は、通例はCPUによって必要とされないが、PICのために提供され得る。たとえば、別のCPUまたはシステムが、書込データおよび/または書込バイトイネーブル信号とともに転送される追加的な情報から利益を得ることができ得る。一実施例では、この追加的な情報は、パリティビットまたはエラー訂正コードを含み得る。この信号は、中位のタイミングの信号として特徴づけられ、`CLK2X` のクロックドメインを有する。

30

【0170】

書込データポート `802A` は、以下の信号を出力することもできる。

【0171】

出力：`S2PIC_WrDataAck[7:0]`

このストロブ信号は、書込データを肯定応答し、次のデータ部分が転送されることを可能にする。`S2PIC_WrDataAck` 信号と `PIC2S_Data` 信号上で利用可能な次のデータワードとの間のレイテンシは、システムに左右される。一実施例では、デフォルトは、`S2PIC_WrDataAck` 信号とアサートされている次のデータとの間の `3CLK2X` サイクル遅延であり得る。`S2PIC_WrDataAck` 信号の各ビットは、その対応のバイトレーンのためのデータを個別に肯定応答し、より小さな幅またはより遅いスレーブが一度にいくつかのバイトレーンを肯定応答することを可能にする。なお、スレーブは、たとえバイトイネーブルビットがそのバイトレーンのためにセットされていなくても、すべてのバイトレーンを肯定応答しなければならない。加えて、スレーブは、増加するバイトアドレスの順序で（これは、システムがビッグエンディアンまたはリトルエンディアンであるかに応じて、ビット[7]から[0]、または[0]から[7]になり得る）`S2PIC_WrDataAck` ビットをアサートしなければならない。遅いスレーブデバイスはまた、`S2PIC_WrDataAck` 信号のアサート間にギャップを挿入して、スレーブがデータを転送するレートを制御することが許可される。結果として、32ビットシステムでは、この信号は、4ビット幅であり得る。この信号は、遅いタイミングの信号として特徴づけられ、`CLK2X` のクロックドメインを有する。

40

【0172】

50

S2PIC_WrXferDone

スレーブは、この信号をアサートして、スレーブが書込データバスの使用を終え、さらには全データを転送したことを示す。この信号は、スレーブが送る必要のあるS2PIC_WrDataAck信号の最後のサイクル上でアサートされ得る。なお、S2PIC_WrXferDone信号は、対応の書込データがPIC2S_Data信号およびPIC2S_DataBE信号を介して到着する前に、アサートされ得る。この信号は、中位のタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

【0173】

c. スレーブのための読出ポート

読出ポート803Aは、以下の信号を受取り得る。

10

【0174】

PIC2S_RdDataReq

スレーブは、マスタが要求する読出データを転送し始め得る前に、この信号がアクティブになることを待たなければならない。マスタは、それが読出することを望むすべてのデータをそれが受取り得ることを保証しなければならないため、この信号は、データ転送の途中で変化することはない。この信号がアクティブになった後、それは、スレーブがS2PIC_RdXferDone信号をアサートするまで、ハイのままである。スレーブは、S2PIC_RdDataAck信号をアサートし、PIC2S_RdDataReq信号がアクティブになる同じサイクル上でデータを転送し始めることができる。この信号は、早いタイミングの信号として特徴づけられ、CLKのクロックドメインを有する。

20

【0175】

読出データポート803Aは、以下の信号を出力することもできる。

【0176】

S2PIC_Data[63:0](CLK2X、中位)

この信号は、転送される読出データを含む。この信号は、32ビットシステム内の32ビットにまでスケールダウンされ得る。この信号は、中位のタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0177】

S2PIC_DataQual[2:0]

この信号は、読出データ転送修飾子を含む。一実施例では、ビット定義は、以下のとおりである。

30

【0178】

【表7】

ビット	名称	記述
[4:0]	読出ワードアドレス	キャッシュライン転送内のどのワードまたはダブルワードが送られているのかを特定する。ビット[0]は、64ビットデータ経路を備えるシステム内では常に0である。

40

【0179】

一実施例では、CPUは、サイズにおいて最大32ワードまでの転送を要求し得る。したがって、5ビットの「読出ワードアドレス」が、このCPUをサポートするために必要とされる。論理的には、したがって、「読出ワードアドレス」フィールドは、32ワードよりも大きな転送に対して定義されないことになる。目標ワード第1転送（つまり、任意の順序でワードを転送する能力）をサポートしないスレーブは、通常のシーケンシャルな順序でデータを転送し、応じて「読出ワードアドレス」ビットをセットするだけでよい。この信号は、中位のタイミングの信号として特徴づけられ、CLK2Xのクロックドメインを有する。

【0180】

50

S 2 P I C _ R d D a t a A c k [7 : 0]

この読出ストローブ信号は、読出データが受取られることを許可する。各 S 2 P I C _ R d D a t a A c k ビットは、その対応のバイトレーンのためのデータを肯定応答し、より小さな幅またはより遅いスレーブが一度にいくつかのバイトレーンを肯定応答することを可能にする。スレーブは、たとえばバイトイネーブルビットがそのバイトレーンに対してセットされていなくても、すべてのバイトレーンを肯定応答しなければならない。加えて、スレーブは、増加するバイトアドレスの順序で（これは、システムがビッグエンディアンまたはリトルエンディアンであるかに応じて、ビット [7] から [0]、または [0] から [7] にまでなり得る）、S 2 P I C _ R d D a t a A c k ビットをアサートしなければならない。結果として、32ビットシステムでは、この信号は、4ビット幅であり得る。この信号は、遅いタイミングの信号として特徴づけられ、C L K 2 X のクロックドメインを有する。

10

【 0 1 8 1 】

S 2 P I C _ R d X f e r D o n e (C L K 、 中 位)

この信号は、スレーブによってアサートされて、それが、読出データバスの使用を終え、すべての要求されるデータを転送したことを示す。この信号は、早いタイミングの信号として特徴づけられ、C L K のクロックドメインを有する。

【 0 1 8 2 】

4 . スレーブデバイスのタイミング

図 9 は、8ワード書込動作中のスレーブのタイミング図を示す。時間 t 1 において、スレーブは、アドレス要求 (s _ a d d r _ r e q)、アドレス (s _ a d d r [3 1 : 0])、およびアドレス修飾子 (s _ a d d r _ q u a l []) を受取る。時間 t 3 では、スレーブは、アドレス肯定応答 (s _ a d d r _ a c k) を生成し、マスタが別のトランザクションを進めることを可能にし、時間 t 3 では、データ D 1 w の書込動作を始める。なお、この実施例では、時間 t 7 においてスレーブがデータ D 2 w を書込む前に、2クロック (c l k) サイクルのレイテンシが提供される。このレイテンシは、パイプライン方式に基づき、他の実施例では異なり得る。

20

【 0 1 8 3 】

一旦スレーブが時間 t 3 において書込動作を開始すると、スレーブは、データを書込むために必要とされる時間（つまり、2クロック (c l k) サイクル）の間、書込肯定応答 (s _ w d a t a _ a c k [0]) をアサートする。言い換えると、スレーブは、すべてのデータを求め、それが実際にデータを受取る前に、書込トランザクションのサイズを肯定応答する。この状態で、一旦データの残り（つまり、データ D 2 w - D 4 w ）が受取られると、スレーブは、3 D D R サイクルで、つまり、時間 t 7 - t 9 で、このデータを書込み得る。スレーブが、時間 t 9 (1クロック (c l k) サイクルのレイテンシ) でトランザクションのためのデータを受取ったことを確認した後、スレーブは、書込転送終了信号 (s _ w x f e r _ d o n e) をマスタに戻す。なお、P I C は、書込データ要求を時間 t 3 - t 11 の間、アサートし、これは、スレーブに対するアクティブ書込期間を示す。

30

【 0 1 8 4 】

図 10 は、8ワード読出動作中のスレーブのタイミング図を示す。時間 t 1 において、スレーブは、アドレス要求 (s _ a d d r _ r e q)、アドレス (s _ a d d r [3 1 : 0])、およびアドレス修飾子 (s _ a d d r _ q u a l []) を受取る。時間 t 3 において、スレーブは、アドレス肯定応答 (s _ a d d r _ a c k) を生成し、マスタが別のトランザクションを進めることを可能にする。なお、スレーブは、マスタからの書込データを待っていないため、スレーブは、1クロック (c l k) サイクルのレイテンシの後、時間 t 5 において、その読出動作をすぐに開始できる。

40

【 0 1 8 5 】

一旦、時間 t 5 においてスレーブが読出動作を開始すると、スレーブは、データを読出すために必要とされる時間（つまり、2クロック (c l k) サイクル）の間、読出肯定応

50

答 (s _ r d a t a _ a c k [0]) をアサートする。1クロック (c l k) サイクルのレイテンシ後、スレーブは、読出転送終了信号 (s _ r x f e r _ d o n e) をマスタに戻す。なお、スレーブは、読出データ要求を時間 t 3 - t 9 の間、アサートし、これは、スレーブのためのアクティブ読出期間を示す。

【 0 1 8 6 】

B . プログラマブルインターフェイスコア

図 1 1 A および図 1 1 B は、P I C に関連するさまざまなアドレス / 制御信号を含むプログラマブルインターフェイスコア (P I C) 内のアドレス / 制御経路の一実施例を示す。図 1 1 A および図 1 1 B では、4つのマスタデバイスが示される。すなわち、D C U 1 1 0 1 A (C P U のデータキャッシュユニット)、I C U 1 1 0 1 B (C P U の命令キャッシュユニット)、O P B スレーブ (ブリッジイン) 1 1 0 1 C、および高速バスインターフェイス (H S B I) 1 1 0 1 D を示す。この実施例では、各マスタ 1 1 0 1 は、その要求をファーストインファーストアウト (F I F O) レジスタ 1 1 0 4 に提供する。したがって、マスタ 1 1 0 1 からの要求は、F I F O 1 1 0 4 内で待ち行列に入れられる。

10

【 0 1 8 7 】

マスタデバイス 1 1 0 1 A 内に示されるレジスタへの書込および活性化のための回路 1 1 0 2 が、マスタデバイス 1 1 0 1 B - 1 1 0 1 D において繰返され得る。具体的には、各マスタデバイス 1 1 0 1 は、論理ゲート 1 1 0 2、たとえば、(簡素化のために、1つのフリップフロップとして図示される) 複数のフリップフロップ 1 1 0 3 を活性化するための A N D ゲートを含み得る。論理ゲート 1 1 0 2 が受取る制御信号に基づいて、適切な制御信号が、レジスタ 1 1 0 4 の書込イネーブル (W E N) 端子に提供される。レジスタ 1 1 0 4 は、一杯のときは、その F U L L 端子上で適切な論理信号をアサートし、これが次に論理ゲート 1 1 0 2 に提供されて、レジスタ 1 1 0 4 の W E N を不活性化する。

20

【 0 1 8 8 】

フリップフロップ 1 1 0 3 も、トランザクションのためのアドレス修飾子およびアドレスを受取る。なお、フリップフロップ 1 1 0 3 は、D D R、つまり、c l k 2 x で、この情報を受取り得、F I F O 1 1 0 4 は、標準の F P G A クロック、つまり、c l k で、この情報を受取る。F I F O 1 1 0 4 A - 1 1 0 4 D は、それらのそれぞれのアドレスを P I C 1 1 0 0 内の複数のアドレスコンパレータ 1 1 0 5 A - 1 1 0 5 D に提供する。各アドレスコンパレータ 1 1 0 5 は、その入来アドレスをデコードし、デコードされたアドレスを F P G A 内のスレーブデバイスの、その記憶されたアドレスと比較する。

30

【 0 1 8 9 】

上述のように、一実施例では、アドレスの上位ビットのみが、スレーブデバイスを識別し、下位ビットは、スレーブデバイス内のメモリアドレスを識別する。この状態で、アドレスコンパレータ 1 1 0 5 は、どのスレーブデバイスが要求されているのかを迅速に判断することができる。一旦「ヒット」が判断されると、つまり、要求されるスレーブが、トランザクションに対して識別されると、スレーブ識別情報が、複数のレジスタ 1 1 0 6 に提供され、これらが、次に、情報をアービトレーション論理 1 1 0 8 に転送する。なお、アドレスに関連するアドレス修飾子は、エンコードされず、したがって、アドレスコンパレータ 1 1 0 5 を通過しない。一実施例では、アドレス修飾子は、F I F O 1 1 0 4 からレジスタ 1 1 0 6 へと直接転送され、次に、マルチプレクサ 1 1 0 7 に転送される。

40

【 0 1 9 0 】

マルチプレクサ 1 1 0 7 は、どのアドレス修飾子が別のレジスタ 1 1 0 9 セットに出力されるのかを選択的に選ぶ。なお、レジスタ 1 1 0 9 は、レジスタ 1 1 0 6 のように、この発明のパイプライン方式を提供し得る。この発明の他の実施例は、より少ない、またはより多くのレジスタを含み得る。マルチプレクサ 1 1 0 7 は、ステートマシン 1 1 1 0 によって制御され、これは、循環アービトレーション論理 1 1 0 8 およびマルチプレクサ 1 1 0 7 から入力を受取る (なお、一旦ステートマシン 1 1 1 0 が適切なマスタを選択すると、ステートマシン 1 1 1 0 は、選択されたマスタが読出または書込トランザクションを

50

要求しているのかを知る必要がある)。最初、ステートマシン 1110 は、アイドル状態にあり、それは、(レジスタ 1109 を介して)スレーブデバイスに対し、1つの要求もペンディングでないことを伝える。アービトレーション論理 1108 によって提供される、選択されるアドレス要求を受けた後、ステートマシン 1110 は、「チャンネル」がセットされるアドレス送信状態に入る。言換えると、ステートマシン 1110 は、トランザクションのための選択されたマスタデバイスを肯定応答し、アドレス要求およびアドレス修飾子を適切なスレーブデバイスに送り始める。一旦、スレーブデバイスがアドレス要求を受取ると、スレーブは、受取りを肯定応答し、ステートマシン 1110 をアイドル状態に再セットし得る。この状態で、ステートマシン 1110 は、次のトランザクションを見ることができる。なお、次のトランザクションが、現在のトランザクションに現在関与しているスレーブに対するものであるならば、現在のトランザクションが完了するまで、スレーブは、その肯定応答を送らない。

10

【0191】

図 12A および図 12B は、この発明に従った書込データ経路に関連する回路の一実施例を示す。図 12A は、図 11A に示される同じ 4 つのマスタデバイスを示す。すなわち、DCU 1101A、ICU 1101B、OPB スレーブ 1101C、および高速バスインターフェイス(HSBI)1101Dを示す。この実施例では、各マスタ 1101 は、そのデータをファーストインファーストアウト(FIFO)レジスタ 1204 に提供する。したがって、マスタ 1101 からの要求は、FIFO 1204 内で待ち行列に入れられる。

20

【0192】

マスタデバイス 1101A 内で示される FIFO 1204 にデータを転送するための回路が、マスタデバイス 1101B - 1101D において繰返され得る。具体的には、各マスタデバイス 1101 は、論理ゲート 1202、たとえば、(簡素化のために、1つのフリップフロップとして図示される)複数のフリップフロップ 1203 を活性化するための AND ゲートを含み得る。論理ゲート 1202 が受取る制御信号に基づいて、適切な制御信号が、FIFO 1204 の書込イネーブル(WEN)端子に提供される。

【0193】

フリップフロップ 1203 も、トランザクションのためのデータを受取る。なお、フリップフロップ 1203 は、この情報を DDR、つまり、clk2x で受取り得、FIFO 1204 は、この情報を標準の FPGA クロック、つまり、clk で受取る。FIFO 1204A - 1204D は、それらのそれぞれのデータをパイプライン方式のために複数のレジスタ 1205 に提供し、次に、マルチプレクサ 1206 に提供する。なお、フリップフロップ対 FIFO の実現は、アプリケーションに依存し得る。たとえば、マスタが書込データの全 DDR レートについていくことができる場合、フリップフロップが用いられ得る。しかしながら、マスタが全 DDR レートについていくことができない場合、書込データは FIFO に送られ、マスタがそれ自体のペースで書込データを処理することが可能となり得る。

30

【0194】

マルチプレクサ 1206 は、どのデータがデータ経路に沿って別のパイプラインレジスタ 1210 に出力されるのかを選択的に選ぶ。なお、PIC 1200 のフルクロスポイント実現例では、ステートマシン 1208、マルチプレクサ 1206、およびパイプラインレジスタ 1210 が、各スレーブデバイスごとに繰返され得る。マルチプレクサ 1206 は、(フリップフロップ 1209 を介して)ステートマシン 1208 によって制御され、これは、このデータ経路が現在用いられているかを示すスレーブから入力を受取る。経路が用いられている場合(この場合には、論理 1 信号がアサートされる)、データ経路は、「ビジー」であり、ステートマシン 1208 は、別のマスタがその同じデータ経路上で書込動作を開始することを可能にしない。

40

【0195】

最初は、ステートマシン 1208 は、アイドル状態にあり、それは、スレーブデバイス

50

に対して、１つの書込要求もペンディングでないことを伝える。なお、別のスレーブがそれを用いているために、書込データリソースがビジーである場合には、ステートマシン１２０８は、ＩＤＬＥでとどまらなければならない。書込要求を受けた後、ステートマシン１２０８は、転送状態に入り、（１つのスレーブデバイスに対するデータ経路に関連した）マルチプレクサ１２０６を活性化してデータを送る。一旦スレーブデバイスがデータを受取ると、スレーブは、適切な論理信号を論理（ＡＮＤ）ゲート１２１１に提供することによって、受取りを肯定応答し得る。なお、論理ゲート１２１１およびパイプラインレジスタ１２１２は、スレーブデバイスの数だけ繰返される。論理ゲート１２１１もマルチプレクサ選択信号を受ける。言換えると、論理ゲート１２１１のうちの１つのみが、マルチプレクサ選択信号と書込肯定応答との両方のためのアクティブ信号を受取る。次に、レジスタ１２１２の出力が、ＯＲゲート１２１３に提供され、これは、次に、書込肯定応答信号をマスタ１１０１のうちの１つ（この場合、マスタデバイス１１０１Ａ）に提供する。書込トランザクションが完了した後、スレーブは、書込転送終了信号をステートマシン１２０８に送り、ステートマシン１２０８がそのアイドル状態に再び入ることを可能にする。

10

【０１９６】

図１３Ａおよび図１３Ｂは、この発明に従った読出データ経路に関連する回路の一実施例を示す。図１３Ａでは、図１１Ａと同じ４つのマスタデバイスが示される。すなわち、ＤＣＵ １１０１Ａ、ＩＣＵ １１０１Ｂ、ＯＰＢスレーブ１１０１Ｃ、および高速バスインターフェイス（ＨＳＢＩ）１１０１Ｄが示される。この実施例では、マスタデバイス１１０１Ａおよび１１０１Ｂは、それぞれ、それらの読出データを受取るためのフリップフロップ１３０１Ａおよび１３０１Ｂを含み、マスタデバイス１１０１Ｃおよび１１０１Ｄは、それぞれ、それらの読出データを受取るためのファーストインファーストアウト（ＦＩＦＯ）レジスタ１３０２Ｃおよび１３０２Ｄを含む。なお、フリップフロップ対ＦＩＦＯの実現は、アプリケーションに依存し得る。たとえば、マスタが、読出データの全ＤＤＲレートについていくことができる場合には、フリップフロップが用いられ得る。しかしながら、マスタが全ＤＤＲレートについていくことができない場合には、データはＦＩＦＯに送られ得、マスタがそれ自体のペースで読出データを処理することが可能になる。

20

【０１９７】

最初は、ステートマシン１３０３は、アイドル状態にあり、それは、スレーブデバイスに対して、１つの読出要求もペンディングでないことを伝える。この状態は、ＯＲゲート１３０４に提供される複数の読出マルチプレクサ選択信号によって決定される。これらのマルチプレクサ選択信号のうちのいずれか１つがアクティブハイであって、読出データ経路がアクティブであることを示す場合、ステートマシン１３０３は、アイドル状態にとどまる。読出要求を受取り、かつ読出データ経路がインアクティブであることを確認した後、ステートマシン１３０３は、転送状態に入り、（１つのマスタデバイスに対するデータ経路に関連した）マルチプレクサ１３０７を活性化してデータを送る。なお、フリップフロップ１３０６は、読出データ経路内でパイプライン方式を提供する。したがって、スレーブデバイス（図示せず）とフリップフロップ１３０６との間、およびステートマシン１３０３とフリップフロップ１３０６との間で実質的に等しい遅延を保証するために、フリップフロップ１３０５が、フリップフロップ１３０６への制御経路内に加えられる。

30

40

【０１９８】

一旦スレーブデバイスが要求を受取ると、スレーブは、適切な論理信号を論理（ＡＮＤ）ゲート１３０８に提供することによって、要求を肯定応答し得る。なお、論理ゲート１３０８およびパイプラインレジスタ１３０９は、スレーブデバイスの数だけ繰返される。論理ゲート１３０８も、マルチプレクサ選択信号を受ける。言換えると、論理ゲート１３０８のうちの１つのみが、マルチプレクサ選択信号と読出肯定応答との両方のためのアクティブ信号を受ける。次に、レジスタ１３０９の出力がＯＲゲート１３１０に提供され、これは、次に、読出肯定応答信号をマスタ１１０１のうちの１つ（この場合、マスタデバイス１１０１Ａ）に提供する。書込トランザクションが完了した後、スレーブは、書込転

50

送終了信号をステートマシン 1208 に送り、ステートマシン 1208 がそのアイドル状態に再び入ることを可能にする。

【0199】

重要なことには、図 11A / 11B、図 12A / 12B、および図 13A / 13B に示される回路によって例示されるように、この発明は、パイプラインレジスタ間の論理の量を最小にする。この状態で、アドレス / 制御経路とデータ経路との両方が、「バランスをとる」。言換えると、両方の経路に沿った論理およびパイプラインレジスタは、均一に分配され、レジスタ（または、論理）間の遅延が実質的に同じになることが保証される。次に、このセットされたタイミングによって、経路の論理が簡素化され得る。

【0200】

C. DCU / ICU : PLB に対するインターフェイス

図 14 は、PLB に対するデータキャッシュユニット (DCU) (または、命令キャッシュユニット (ICU)) インターフェイスの一実施例を示す。この種のインターフェイスは、当該技術分野では公知であり、設計内で用いられている CPU に対してプロプラエタリである。なお、この発明は、いずれの CPU にも適用可能であるため、インターフェイス 1400 は、使用される CPU に基づいて変わり得る。図 14 では、インターフェイス 1400 は、IBM のパワー PC 405 CPU 用として示される。

【0201】

D. ブロック RAM : PIC に対するインターフェイス

図 15 は、2つのポート A および B を有するブロック RAM (BRAM) 1501 を示し、各ポートは、それ自体のクロッキング、制御、アドレス、読出 / 書込機能、およびデータ幅を有し、独立した読出 / 書込能力を可能にする。この実施例では、ポート B は、FPGA 上のユーザ論理に結合され、ポート A は、インターフェイス 1502 に結合される。インターフェイス 1502 は、有限ステートマシン (FSM) 1503 を介して、PIC から信号を受け、PIC にさまざまな信号を提供する。次に、FSM 1503 は、BRAM 1501 内のポート A の書込イネーブルを制御する。インターフェイス 1502 は、カウンタ 1504 をさらに含み、PIC から複数のアドレスを受け取り、さらにはそのカウンタに基づいて、ある特定のアドレスを提供する。なお、このカウンタのロードおよびイネーブル機能も、FSM 1503 によって制御される。この状態で、さまざまなアドレスがロードされ得るが、FSM 1503 が、いつそれらが BRAM 1501 に提供されるのかを決定する。ポート A の書込および読出データ端子は、PIC に結合される。

【0202】

BRAM 1501 は、ポート A とポート B との両方からの同じメモリセルの同時アクセスを可能にする。したがって、一方のポートが、所与のメモリセルに書込む場合、他方のポートは、その書込動作中は、同じメモリセルをアドレス指定できない。なお、競合の解消に取り組むために、ユーザは、データコヒーレンスおよび同期の問題に対処するようにユーザ論理およびソフトウェアを設計し得る。

【0203】

E. OPBブリッジモジュール

1. OPBブリッジアウトモジュール

OPB / BO は、プログラマブルインターフェイスコア (PIC) に対するスレーブデバイスおよび OPB に対するマスタデバイスとして働くブリッジアウトモジュールである。図 16 は、ブリッジアウトモジュール 1600 の一実施例のハイレベル図を示し、これは、パイプラインアーキテクチャを用いて、それが高クロック周波数で OPB マスタインターフェイス上で動作することを可能にする。一実施例では、OPB / BO は、PIC 側で 64 ビット DDR データ経路をサポートし、OPB 側で 32 ビットバイトイネーブル転送をサポートする。ブリッジアウトモジュール 1600 は、論理 1601 を含んで PIC 転送要求をデコードし、FIFO 1602 および 1603 を含んでトランザクションを待ち行列に入れる。具体的には、ダイナミックバイトイネーブルを備えた書込トランザクションは、FIFO 1602 および 1603 内で待ち行列に入れられて、OPB 転送が完了

10

20

30

40

50

することを待たずに、P I C マスタ（図示せず）からデータをオフロードすることができる。ブリッジアウトモジュール 1 6 0 0 は、O P B マスタ 1 6 0 5 の制御およびデータ経路論理の一部として含まれる複数のステータスレジスタも含む。これらのステータスレジスタは、P I C マスタによって開始されるトランザクションから結果として得られるいずれかの O P B エラーを記録し得る。O P B マスタ 1 6 0 5 は、エラーおよびビジー信号を適切に生成して各 P I C マスタに戻す論理をさらに含む。

【 0 2 0 4 】

ブリッジアウトモジュール 1 6 0 0 を含む書込動作では、プログラマブルインターフェイスコア（P I C）は、論理 1 6 0 1 によってデコードされる書込コマンド（スレーブ制御信号 1 6 0 6 のうちの 1 つ）を発行する。ブリッジアウトモジュール 1 6 0 0 が別のトランザクションの実行にビジーではないと、デコード論理 1 6 0 1 が判断すると、デコード論理 1 6 0 1 は、データを書込データ F I F O 1 6 0 3 に転送し始める。デコード論理 1 6 0 1 は、アドレスおよび転送修飾子をアドレス F I F O 1 6 0 2 にもロードする。ブリッジアウトモジュール 1 6 0 0 は、F I F O 1 6 0 2 / 1 6 0 3 が一杯になったとき、さらなるいずれのデータ転送も停止する。一実施例では、書込データ F I F O 1 6 0 3 は、6 4 ビット幅であるが、転送サイズに応じて、3 2 ビットまたは 6 4 ビット幅の量のデータで書込まれ得る。

【 0 2 0 5 】

O P B マスタ 1 6 0 5 のポートに接続される別個の論理グループが、ペンディングトランザクションを求めてアドレス F I F O 1 6 0 2 をポーリングする。アドレス F I F O 1 6 0 2 が空でない場合、トランザクション要求がそこから読出される。アドレス、転送サイズ、および読出 / 書込情報が、O P B データ転送プロセスを開始する O P B マスタ 1 6 0 5 内のステートマシンに送られる。O P B マスタ 1 6 0 5 内の O P B 転送論理は、O P B に対するバスアクセスを要求し、O P B マスタ 1 6 0 5 に O P B バスが許可された場合に書込転送を開始する。転送がバースト（つまり、1 つよりも多いフルワード）である場合、（バスロックおよびシーケンシャルアドレス信号等の）適切な信号がアサートされて O P B 書込バーストが実行される。書込データが、書込データ F I F O 1 6 0 3 から転送されてデータを O P B に供給する。転送が完了すると、アドレス F I F O 1 6 0 2 は、追加的な転送要求を求めて再びポーリングされる。いずれかの肯定応答またはタイムアウト状態が、B E A R / B E S R（図示せず）が更新された状態で、対応の P I C マスタにシグナルバックされる。

【 0 2 0 6 】

ブリッジアウトモジュール 1 6 0 0 を含む読出動作では、読出転送要求は、論理 1 6 0 1 によってデコードされ、アドレス F I F O 1 6 0 2 内で、それが一杯でないと仮定して、待ち行列に入れられる。アドレス F I F O 1 6 0 2 をポーリングする O P B マスタ 1 6 0 5 内の論理は、ペンディング読出要求を認識し、O P B 上において読出転送を実行する。1 つよりも多くのフルワードが読出されるべき場合、バースト読出動作が実行される。読出データは、フリップフロップ 1 6 0 4 を介して P I C マスタに戻される。再び、いずれかの肯定応答またはタイムアウト状態が、B E A R / B E S R が更新された状態で、対応の P I C マスタにシグナルバックされる。

【 0 2 0 7 】

O P B とインターフェイスをとる O P B マスタ 1 6 0 5 内の論理は、好ましくは、設計の O P B 性能目標を満たすように設計される。具体的には、O P B は、シングルサイクルハンドシェーキングプロトコルを用いるため、O P B マスタ 1 6 0 5 の出力がフリップフロップから直接駆動されることが重要である。言換えると、最悪の事態の分析では、すべての信号は、遅いタイミングを有すると仮定されるべきであり、したがって、レジスタ駆動されて O P B における最大伝搬時間を提供すべきである。加えて、一実施例では、O P B マスタ 1 6 0 5 によって受取られている信号は、再びレジスタされる前に、1 つ以下の論理レベルをトラバースし、信号が O P B を通ってスレーブモジュールへと伝搬し、ブリッジアウトモジュール 1 6 0 0 に戻る時間量を最大にする。

【0208】

ブリッジアウトモジュール1600では、アドレスおよび書込データFIFO1602 / 1603は、高いクロックレートで実行するように設計され得る。一実施例では、FIFO設計は、シフトレジスタルックアップテーブル(SRL)FPGAプリミティブに基づいており、読出ポートと書込ポートとの間で同期的関係を必要とする。なお、SRLベースのFIFOの実現は、どのシフトレジスタタップがアクセスされるかを制御するアップ/ダウンカウンタを通例含む。カウンタは、FIFOが書込まれるが、読出されない場合にカウントアップし、FIFOが読出されるが、書込まれない場合に、カウントダウンする。他の様態では、カウンタは、変更されない。それらの簡素性のため、SRLベースのFIFOは、通例、いずれの保護メカニズムもサポートしない。したがって、空のときにFIFOが読出される場合、または一杯の状態ではFIFOが書込まれる場合、回復不能なエラーが生じ得る。これを防ぐために、論理1601は、FIFO1602 / 1603が適切に用いられることを保証するように設計されるべきである。

10

【0209】

上述のように、ブリッジアウトモジュール1600は、フルワードバイトイネーブル転送を用い得る。より高い性能およびより少ない面積を提供するために、ブリッジアウトモジュール1600は、従来のOPBデバイスによって用いられる「ダイナミックバスサイジング」機能を取除き得る。したがって、この実施例では、ブリッジアウトモジュール1600に接続されるOPBモジュールは、フルワードバイトイネーブル転送プロトコルを用い得る。

20

【0210】

しかしながら、FPGAファブリックの観点では、ブリッジアウトモジュール1600は、性能と論理使用との間で異なるトレードオフを提供するように、容易にカスタマイズされ得る。たとえば、ポストされる書込動作がサポートされない場合、アドレスFIFO1602は除去され、書込データFIFO1603は簡素化され得る。エラーおよびビジー信号を扱うOPBマスタ1605内の論理を簡素化することによって、さらなる論理減少がなされ得る。なお、トランザクションをブリッジアウトモジュール1600内で待ち行列に入れることはできないため、これらの変形例は、性能を低下させ得るが、面積を著しく減少させ得る。

【0211】

30

この発明の別の実施例に従うと、アドレスFIFO1602および書込データFIFO1603は、SRL FIFOとともに設計される代わりに、ブロックRAM(BRAM)ベースのFIFOに変えられて、論理使用を減じる。さらに、BRAM FIFOは、より高いクロック対出力遅延を有し、達成され得る最大OPBクロックレートを潜在的に減じる。BRAM FIFOは、非同期FIFOを実現するためにも用いられ得、OPBクロックがブリッジアウトモジュール1600のクロックからデカップルされることを有利なことに可能にする。

【0212】

デカップリングが望まれる場合、最高の性能が必要とされないことを仮定して、OPBマスタ1605内の論理は、簡素で、容易な態様で設計され得る。具体的には、(ブリッジアウトモジュール1600のクロックレートの半分等の)より遅いOPBクロックレートでは、論理のうちの多くは、マルチサイクル経路を含むかまたはより多くの論理レベルが用いられることを可能にするように、設計され得、設計の論理使用を著しく減じる。

40

【0213】

2. OPBブリッジインモジュール

OPB / Bモジュールは、スレーブとしてOPBトランザクション要求を受取り、マスタとして、それらの要求をPIC転送に変換する。図17は、この発明に従ったブリッジインモジュール1700の一実施例のハイレベル図を示す。ブリッジインモジュール1700内では、OPBからトランザクションをオフロードするためにデータをバッファリングする書込データFIFO1704にOPB書込データが記憶される。可変長バースト

50

が P I C スレーブモジュールによってサポートされない一実施例では、O P B バースト書込は、P I C 上で一連のシングルビートまたはキャッシュライン転送に変換され得る。O P B 読出の結果として、P I C スレーブモジュールに対して発行されるデータフェッチが得られる。バースト読出要求は、ブリッジインモジュール 1 7 0 0 に最大 8 ワードのデータプリフェッチを実行させる。ブリッジインモジュール 1 7 0 0 は、必要なクロックドメインおよびバス幅変換も実行して、P I C 6 4 ビット D D R 読出 / 書込データ経路と O P B 3 2 ビット非 D D R 共有データ経路との間でデータを移動させ得る。

【 0 2 1 4 】

書込動作では、O P B ブリッジインモジュール 1 7 0 0 によって受取られる O P B 上のトランザクションは、書込データ F I F O 1 7 0 4 へとバッファリングされる。書込データ F I F O 1 7 0 4 は、書込ポスティングをサポートして、トランザクションが P I C 上で実行されている間に O P B が停滞する (tied up) ことを防ぐ。書込トランザクションの行先アドレスおよび書込バーストの長さが、別個の書込要求 F I F O 1 7 0 5 に記録される。書込要求がブリッジインモジュール 1 7 0 0 内で完全に待ち行列に入れられた後、O P B トランザクションは完了し、O P B は、他のトランザクションに利用可能となる。書込データ F I F O 1 7 0 3 または書込要求 F I F O 1 7 0 5 が一杯である場合、いずれかの追加の書込要求が、再試行信号が論理 1 7 0 1 によりアサートされるようにする。

【 0 2 1 5 】

アドレスポート制御論理 1 7 0 6 内のステートマシンは、ペンディング書込トランザクションを求めて書込要求 F I F O 1 7 0 5 を連続的にポーリングする。書込要求 F I F O 1 7 0 5 が空ではないことを、制御論理 1 7 0 6 が認識すると、待ち行列に入れられた書込トランザクションが取出され、処理される。O P B からの書込トランザクションは、いずれかのワード境界で開始し、最大 3 2 ワード長になり得る。したがって、書込トランザクションは、P I C によってサポートされる一連のワードまたはキャッシュラインアラインド転送に変換され得る。この変換は、開始アドレスおよびバースト長を見て、用いられ得る最大の P I C 転送サイズを計算する、制御論理 1 7 0 6 内でも提供される要求アライメント論理の助けを用いて実行される。P I C 転送は、アドレス F I F O 1 7 1 0 へと入れられ、ここで、トランザクションは、P I C (マスタ) ポート上で呈示され得る。完全なトランザクションセットがアドレス F I F O 1 7 1 0 にロードされるまで、元の書込バーストの任意の残りの部分は、反復的に制御論理 1 7 0 6 を通して送られる。なお、この時点では、書込データは、待ち行列に入れられ、P I C にとって利用可能である。

【 0 2 1 6 】

読出トランザクションは、P I C スレーブからの読出フェッチを制御し、さらにはデータを O P B を通して戻す制御論理 1 7 0 6 内の別個のステートマシンを通して扱われる。O P B 読出要求が非バーストである場合、シングルワード読出要求が、アドレス F I F O 1 7 1 0 にロードされる。O P B 読出要求がバースト動作である場合、ブリッジインモジュール 1 7 0 0 は、必要な最小のサイズの読出要求をアドレス F I F O 1 7 1 0 にロードすることによって、次の 8 ワードキャッシュライン境界までのデータをフェッチする。

【 0 2 1 7 】

読出要求から戻されるデータは、P I C 6 4 ビット D D R 読出データ経路をサポートするシフトレジスタ 1 7 0 9 に記憶される。シフトレジスタ 1 7 0 9 内のデータは、次に、(上で詳細に説明される) 適切な 3 2 ビット O P B データ経路へと多重伝送され、対応の転送肯定応答信号がアサートされる。8 ワード境界を超えて続く読出バーストについては、新しい 8 ワードキャッシュライン要求が、P I C を介して次の記憶場所に対して作成されてバーストを続ける。ブリッジインモジュール 1 7 0 0 がデータをキャッシュしない一実施例では、たとえデータが前のトランザクションから利用可能であったとしても、各読出は、新しい P I C トランザクション要求を必要とする。なお、P I C からのいずれかの読出エラーにより、O P B エラー肯定応答信号がアサートされ得る (図示されないが、シフトレジスタ 1 7 0 9 の論理によってキャプチャされ得る) 。一実施例では、論理 1 7 0 1 は、それが読出データが戻されるのを待つ間、タイムアウト抑制をアサートし得る。こ

10

20

30

40

50

の機能は、P I C トランザクションが完了するために不確定な量の時間がかかり得るため、有益である。

【0218】

書込制御論理 1703 および 1707 の一部を形成するアドレスデコード論理は、P I C から複製され、ブリッジインモジュール 1700 は、P I C 側に存在することが知られているアドレスのためのトランザクションのみを受入れる。この状態で、ブリッジインモジュール 1700 は、アドレスエラーを生成することを妨げられ、さらにはブリッジアウトモジュール (図 16 参照) を備えたフィードバックループを引起こすことを妨げられる。さらに、この複製によって、ブリッジインモジュールとブリッジアウトモジュールとの間の P I C 接続経路が取除かれることも可能となり、貴重な論理リソースが節約される。

10

【0219】

アドレス F I F O 1710、書込要求 F I F O 1705、および書込データ F I F O 1704 は、高いクロックレートで実行されるように設計され得る。具体的には、一実施例では、これらの F I F O は、読出ポートと書込ポートとの間で同期的な関係を必要とするシフトレジスタルックアップテーブル (S R L) F P G A プリミティブに基づき得る。上述のように、それらの簡素化のために、S R L ベースの F I F O は、通例、いずれの保護メカニズムもサポートしない。したがって、空のときに F I F O が読出される場合、または一杯であるときに F I F O が書込まれる場合、回復不能なエラーが生じ得る。これを防ぐために、論理 1706 / 1703 / 1707 は、F I F O 1710 / 1705 / 1704 が適切に用いられることを保証するように設計されるべきである。

20

【0220】

ブリッジインモジュール 1700 内の P I C マスタポートの高クロック周波数は、設計のタイミング要件を満たすことを困難にし得る。したがって、2 X クロックドメインで動作するか、または 1 X クロックの経路に敏感な (sensitive) 論理は、フリップフロップ間に最大 1 論理レベルが存在するように、慎重に設計されなければならない。この設計制約によって、タイミングを改善する助けとなり得る特定の F P G A プリミティブおよび / または効率的な論理構造の使用が促進される。

【0221】

この発明の一実施例では、読出プリフェッチサイズは、8 ワードから 16 (または、32) ワードへと増大し、長い O P B 読出バーストが一般的である場合に、ブリッジインモジュール 1700 の効率を改善する。なお、短い O P B バースト読出は、より高いレイテンシを経験し得る。したがって、設計者は、何が最適なプリフェッチサイズであるかを決定しなければならない。有利なことには、この発明は、追加的な論理をほとんど用いることなく 16 または 32 ワードプリフェッチを用いるように変更され得る。具体的には、上述のように、P I C からの読出データは、64 ビット幅 S R L アレイに記憶され得る。各 S R L は、本質的には 16 のレジスタの深さ (deep) であるため、32 ワードの読出データのための十分な記憶部分が既に存在する。したがって、この場合、ユーザは、読出データ多重論理を変更し、読出ステートマシン論理を修正して、より長い読出要求をアドレス F I F O 1710 にロードし得る。

30

【0222】

この発明の別の実施例では、書込要求 F I F O 1705 が除去されて、設計内で用いられる論理の量が減じられ得る。しかしながら、この F I F O の除去は、書込ポスティング機能をディセーブルにし得る。書込ポスティング機能をディセーブルにすることは、P I C 上のペンディングトランザクションにより停止される書込トランザクションのために、O P B 性能を減じるおそれがある。

40

【0223】

書込データ F I F O 1704 内の制御論理は、多量の冗長論理を用いる。この冗長性は、ファンアウトを減じ、F I F O 1704 が所望の D D R クロックレートまでランすることを可能にする。より低いクロック速度では、この冗長制御論理のうちの多くが除去され得る。加えて、タイミング制約が緩和されている場合には、図 17 のレジスタ 1711、

50

１７０２、および１７０８等のいくつかの読出データ経路パイプラインレジスタも取出されて、レイテンシおよびリソース使用が減じられ得る。

【０２２４】

F．ハイブリッドプログラマブルインターフェイスコア

この発明の別の実施例では、プログラマブルインターフェイスコア（PIC）は、上で詳細に説明されたクロスポイントスイッチに加えて、共有バスの要素を含む。読者の便宜を図って、図１８Ａおよび図１８Ｂは、それぞれ、共有バスアーキテクチャのアドレス／制御経路およびデータ経路を示す。図１８Ａに示されるように、アドレス／制御経路のための共有バス１８００Ａは、マスタデバイス１８０１のアドレス／制御信号を受取るアービタ１８０３を提供することによって、実現され得る。一旦、アクセスアービタ１８０３が、どのマスタデバイス１８０１が選択されてそのトランザクションを進めるべきかを決定すると、要求は、すべてのスレーブデバイス１８０２に送られる。各スレーブデバイス１８０２は、アドレス／制御情報をデコードし、それがそのトランザクションのための要求されるスレーブであるかを判断する。図１８Ｂを参照して、データ経路用の共有バス１８００Ｂは、マスタデバイス１８０１から書込データを受取る単一の書込マルチプレクサ１８０４を提供することによって、実現され得る。共有バス内では、すべてのスレーブデバイス１８０２が、選択される書込データを受取るが、１つのスレーブデバイス１８０２しか、実際にはデータを書込まない。読出動作では、スレーブデバイス１８０２は、そのスレーブデバイスが読出トランザクションにおいてアクティブでない限りは、論理０信号を出力し得る。この状態で、ＯＲゲート１８０５は、読出情報を読出マルチプレクサ１８０６に転送する。制御回路は、図示されていないが、書込動作を完了するために、どの書込データが書込マルチプレクサ１８０４によって選択されるのかを決定し、さらには、読出動作を完了するために、どの読出マルチプレクサ１８０６が選択されるのかを決定する。

【０２２５】

図１９Ａおよび図１９Ｂは、それぞれ、この発明の別の実施例に従ったハイブリッドクロスポイント／共有バスアーキテクチャのアドレス／制御経路およびデータ経路を示す。図１９Ａに示されるように、アドレス／制御経路用のハイブリッドスイッチ１９００Ａは、それぞれ、マスタデバイス１９０１Ａ－１９０１Ｄからアドレス／制御情報を受取るための複数のアドレスデコーダ１９１１Ａ－１９１１Ｄを含む。ハイブリッドスイッチ１９００Ａ内では、３つのアクセスアービタ１９１２Ａ－１９１２Ｃが、アドレスデコーダ１９１１Ａ－１９１１Ｄから、デコードされたアドレスを受取り得る。アクセスアービタ１９１２Ｃは、２つのスレーブデバイス１９２２Ａおよび１９２２Ｂのためのアービトレーションを決定する。この実施例では、アクセスアービタ１９１２Ｃに結合される共有バスのために、スレーブデバイス１９２２Ａおよび１９２２Ｂのうちの１つのみが、トランザクション内でアクティブになり得る。スレーブデバイス１９０２Ａおよび１９０２Ｂとは異なり、スレーブデバイス１９２２Ａおよび１９２２Ｂは、デコーディング論理を含んで、それらのうちのどれがトランザクションのために選択されたのかを判断しなければならない。なお、ハイブリッドスイッチ１９００Ａの共有バス部分では、パイプラインレジスタが除去された。

【０２２６】

図１９Ｂに示されるように、データ経路用のハイブリッドスイッチ１９００Ｂは、複数の読出マルチプレクサ１９１９を含み、その１つ１つは、各マスタデバイス１９０１用である。なお、スレーブデバイス１９０２Ｃおよび１９０２Ｄは、クロスポイントスイッチ（図５Ｃ参照）内でのように読出マルチプレクサ１９０１に直接出力を提供する代わりに、論理（ＯＲ）ゲート１９３０に出力を提供する。したがって、ハイブリッドスイッチ１９００Ｂ内では、任意の時点において、１つのスレーブデバイス１９０２Ｃまたは１９０２Ｄしか、読出情報をマスタデバイス１９０１に提供し得ない。この実施例では、スレーブデバイス１９０２Ｃおよび１９０２Ｄのうちのインアクティブスレーブデバイスが、論理０を出力し得る。ハイブリッドスイッチ１９００Ｂは、スレーブデバイス１９０２の数

10

20

30

40

50

に満たない、複数の書込マルチプレクサ 1914A - 1914C も含む。したがって、スレーブデバイス 1902C と 1902D との両方が、書込マルチプレクサ 1914C によって提供される書込データを受取る。(アクセスアービタ 1912C によって提供されるアドレスにより決定されるような(図 19A 参照)) 選択されるスレーブデバイス 1902 は、データを書込み、選択されていないスレーブデバイス 1902 は、データを無視する。

【0227】

重要なことには、ハイブリッドスイッチ 1900A / 1900B 内のマスタプロトコルおよびスレーブプロトコルの両方が、クロスポイントスイッチ 500A / 500B に関して上で詳細に説明されたように同じままであり得る。しかしながら、一実施例では、再アービトラートポートが追加されて、ハイブリッドスイッチの共有バス部分上のデッドロックが防がれ得る。なお、ハイブリッドスイッチ 1900A / 1900B は、クロスポイントスイッチのように、不必要な経路をなくすためにトリミングされ得る。したがって、たとえば、マスタデバイス 1901B (つまり、ICU) に関連する書込経路が、トリミングされ得る。

10

【0228】

ハイブリッドスイッチ 1900A / 1900B は、フルクロスポイントスイッチと比較して、用いられるリソースの数を減じ得る。具体的には、ハイブリッドスイッチ内で用いられる論理ゲートは、クロスポイントスイッチのマルチプレクサよりも、より少ないリソースを用いる。しかしながら、この利点は、潜在的な性能低下とバランスをとらなければならない。一実施例では、この発明は、第 2 のスレーブデバイスセットよりも遅い第 1 のスレーブデバイスセットが、ハイブリッドスイッチの共有バス部分を用いるように構成され得、第 2 のスレーブデバイスセットがハイブリッドスイッチのクロスポイント部分を用いるように構成され得ることを決定し得る。FPGA 環境では、システム設計者は、共有バス/クロスポイントスイッチリソースの適切な混合を容易に決定して性能を最大にし、面積を最小にすることができる。

20

【0229】

G. ソフトウェアツール

この発明に従うと、プログラマブルロジックデバイス (PLD) のためのコンフィギュレーションビットストリームを生成するための方法が提供される。この方法は、図 3 を参照しながら説明されるシステムに容易に組込まれ得る。具体的には、この発明の方法は、コアジェネレータを活性化することと、コアジェネレータを用いてプログラマブルインターフェイスコアを選択することと、コンフィギュレーションビットストリームを生成する PLD ソフトウェアツールにプログラマブルインターフェイスコアを提供することとを含む。PLD ソフトウェアは、この発明のプログラマブルインターフェイスを受取ることに加えて、ユーザが提供するトップレベル設計を受取ることができる。重要なことには、プログラマブルインターフェイスコアは、複数のコンプライアンスレベルを選択的に提供することができる。たとえば、プログラマブルインターフェイスコアは、上で詳細に説明されたように、プロセッサローカルバス (PLB) 機能のサブセットを提供し得る。一実施例では、プログラマブルインターフェイスコアは、トップレベル設計が必要とする機能にサブセットを調整することができる。

30

40

【0230】

ここで提供されるこの発明の説明は、例示にすぎず、限定ではない。具体的には、この発明のさまざまな実施例が、上で詳細に説明された。これらの実施例の変形例は、当業者には明らかであろう。したがって、この発明の範囲は、添付の請求項によってのみ規定され得る。

【図面の簡単な説明】

【0231】

【図 1】中央処理装置、プロセッサローカルバス、およびオンチップ周辺バスを含むマイクロプロセッサを示す、簡素化された図である。

50

【図 2】この発明の譲受人が販売するバーテックス F P G A 内のコンフィギャラブルロジックブロックを示す、簡素化されたブロック図である。

【図 3】F P G A 内の 1 つ以上のコアを用いるための 1 つの I P プロセスレイアウトを示す図である。

【図 4】この発明に従った基本的な組込用プロセッサ F P G A システムを示す図である。

【図 5 A】この発明に従った 4 × 4 クロスポイントスイッチを示す簡素化された図である。

【図 5 B】アドレス論理内での図 5 A のクロスポイントスイッチの一実現例を示すより詳細な図である。

【図 5 C】データ論理内での図 5 A のクロスポイントスイッチの一実現例を示すより詳細な図である。

10

【図 5 D】アドレス / 制御論理がトリミングされる図 5 B のクロスポイントスイッチの一実施例を示す図である。

【図 5 E】データ論理がトリミングされる図 5 C のクロスポイントスイッチの一実施例を示す図である。

【図 6】本発明のプログラマブルインターフェイスコアおよびマスタデバイスのために提供されるポートと、このようなポート間で転送され得る信号との全体像を示す図である。

【図 7】8 ワード書込動作およびその後続く 8 ワード読出動作中のマスタのタイミング図である。

【図 8】本発明のプログラマブルインターフェイスコアおよびスレーブデバイスのために提供されるポートと、このようなポート間で転送され得る信号との全体像を示す図である。

20

【図 9】8 ワード書込動作中のスレーブのタイミング図である。

【図 10】8 ワード読出動作中のスレーブのタイミング図である。

【図 11 A】この発明に従ったプログラマブルインターフェイスコア内のアドレス / 制御論理の一実施例であって、経路が、コアに関連するさまざまなアドレス / 制御信号を含む、一実施例を示す図である。

【図 11 B】この発明に従ったプログラマブルインターフェイスコア内のアドレス / 制御論理の一実施例であって、経路が、コアに関連するさまざまなアドレス / 制御信号を含む、一実施例を示す図である。

30

【図 12 A】この発明に従った書込データ論理に関連する回路の一実施例であって、論理は、コアに関連するさまざまな書込信号を含む、一実施例を示す図である。

【図 12 B】この発明に従った書込データ論理に関連する回路の一実施例であって、論理は、コアに関連するさまざまな書込信号を含む、一実施例を示す図である。

【図 13 A】この発明に従った読出データ論理に関連する回路の一実施例であって、論理は、コアに関連するさまざまな読出信号を含む、一実施例を示す図である。

【図 13 B】この発明に従った読出データ論理に関連する回路の一実施例であって、論理は、コアに関連するさまざまな読出信号を含む、一実施例を示す図である。

【図 14】プロセッサローカルバスに対する命令キャッシュユニットまたはデータキャッシュユニットのインターフェイスの一実施例を示す図である。

40

【図 15】この発明のプログラマブルインターフェイスコアに対するデュアルポート機能を有するブロック R A M (B R A M) のインターフェイスを示す図である。

【図 16】この発明に従ったプログラマブルインターフェイスコア内のブリッジアウトモジュールの一実施例を示すハイレベル図である。

【図 17】この発明に従ったプログラマブルインターフェイスコア内のブリッジインモジュールの一実施例を示すハイレベル図である。

【図 18 A】共有バスアーキテクチャのアドレス / 制御論理を示す図である。

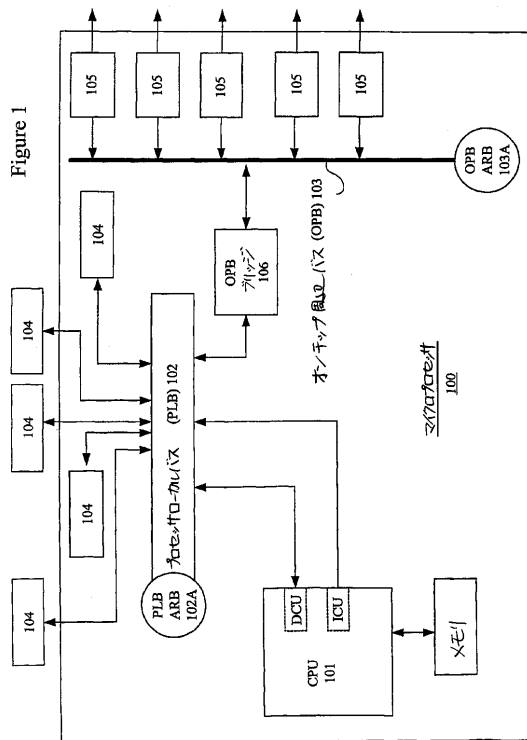
【図 18 B】共有バスアーキテクチャのデータ論理を示す図である。

【図 19 A】この発明の一実施例に従ったハイブリッドクロスポイント / 共有バスアーキテクチャのアドレス / 制御論理を示す図である。

50

【図 19 B】この発明の一実施例に従ったハイブリッドクロスポイント / 共有バスアーキテクチャのデータ論理を示す図である。

【図 1】



【図 2】

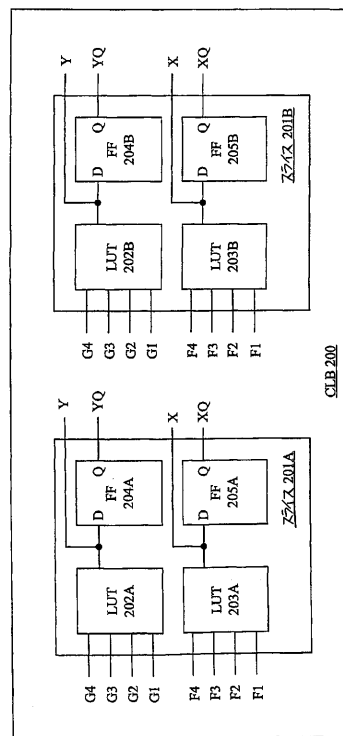


Figure 2

【図 3】

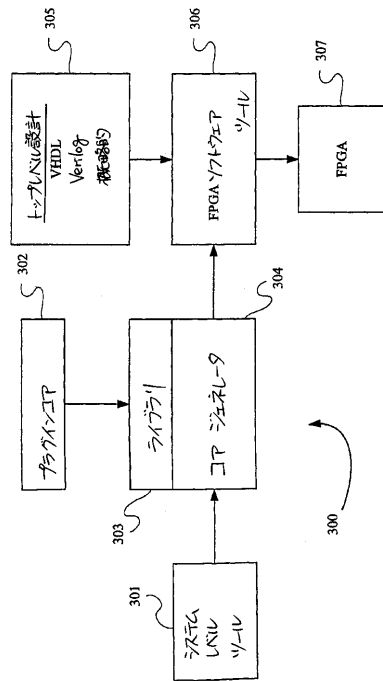


Figure 3

【図 4】

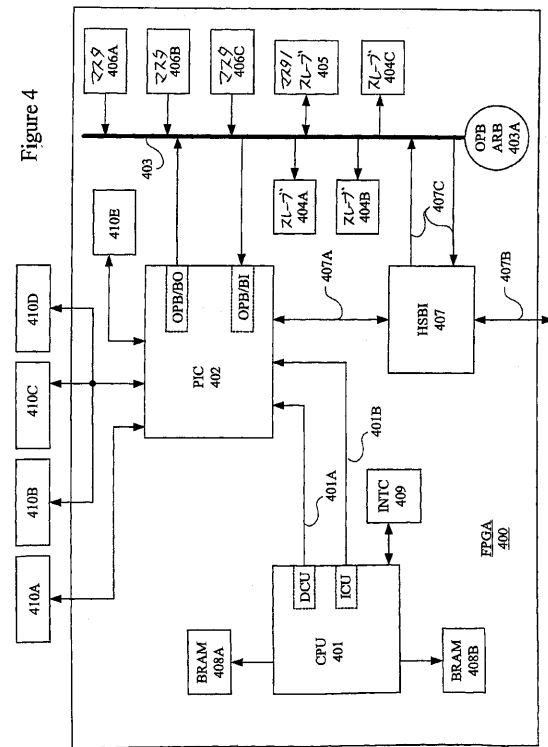


Figure 4

【図 5 A】

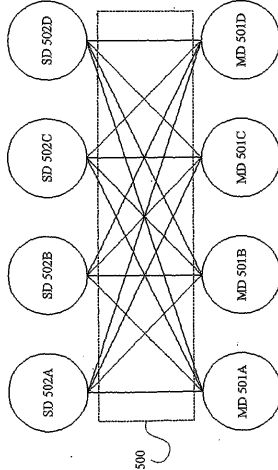


Figure 5A

【図 5 B】

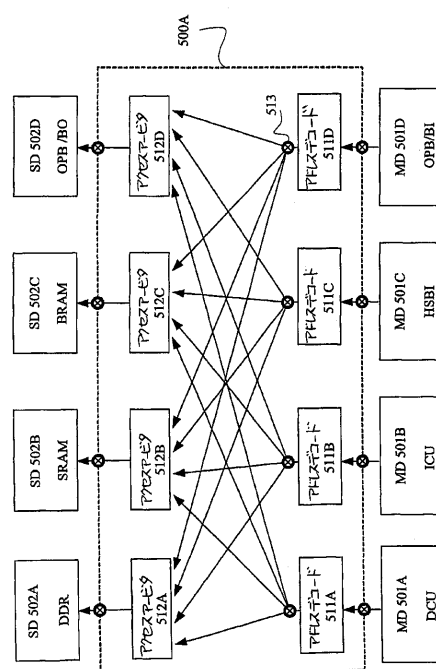


Figure 5B

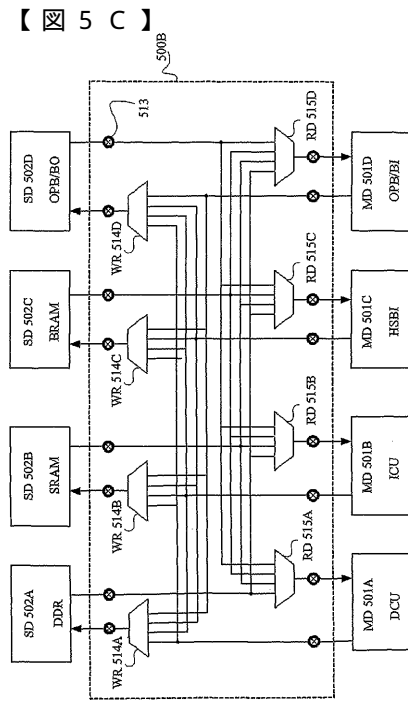


Figure 5C

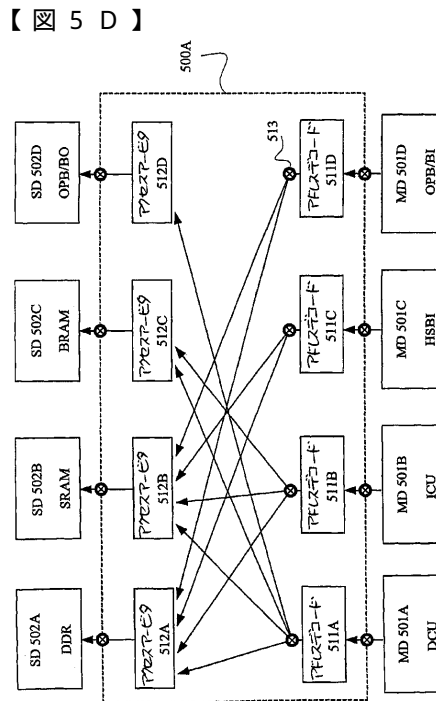


Figure 5D

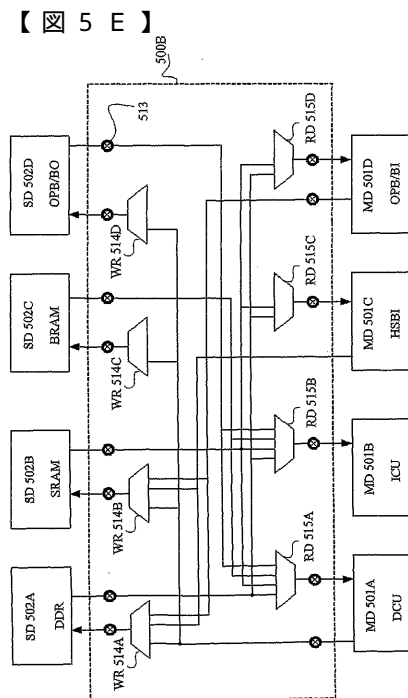


Figure 5E

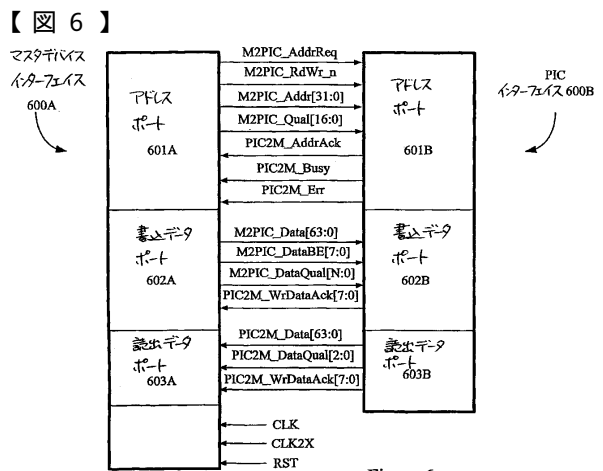


Figure 6

【図 7】

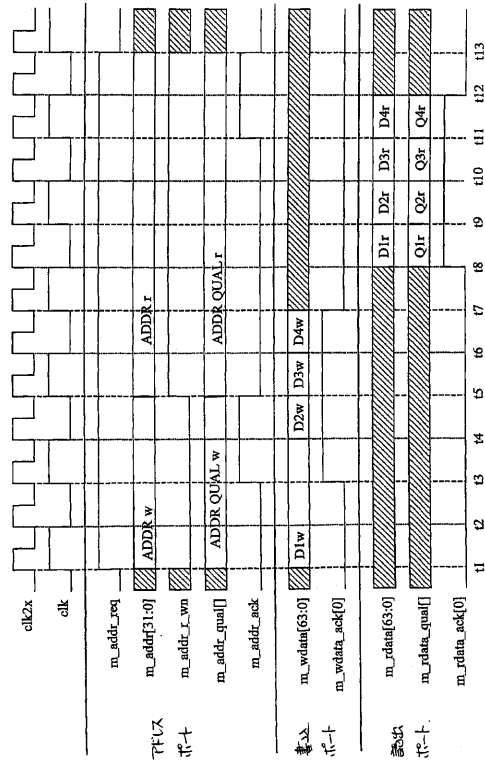


Figure 7

【図 8】

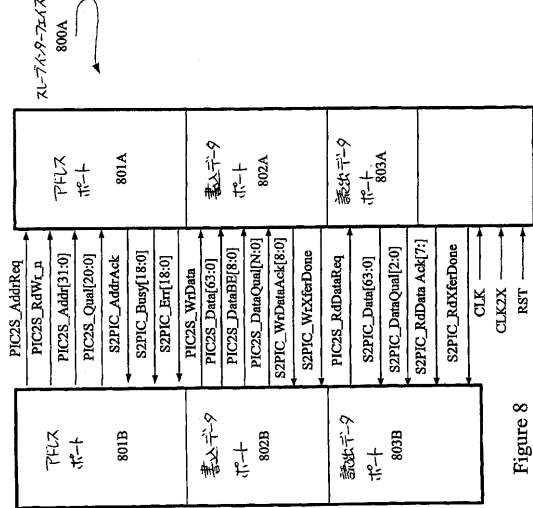


Figure 8

【図 9】

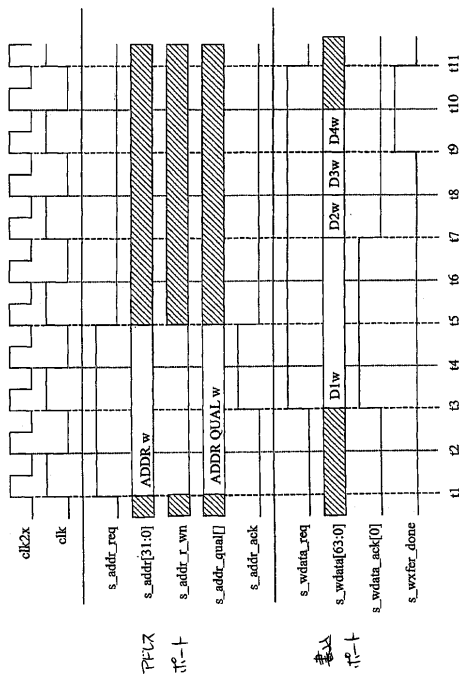


Figure 9

【図 10】

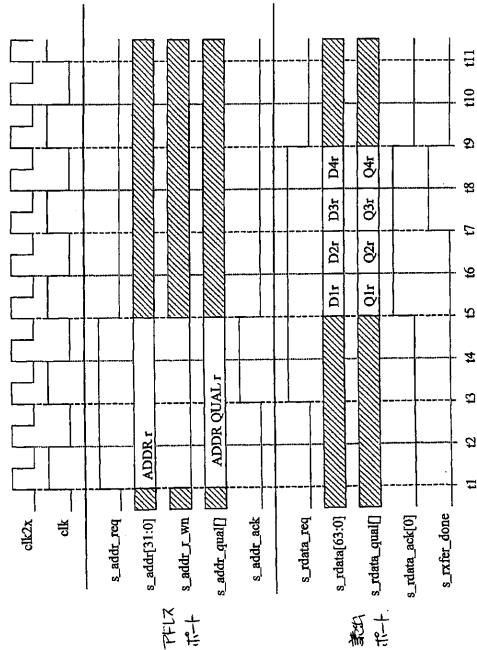


Figure 10

【図 11 A】

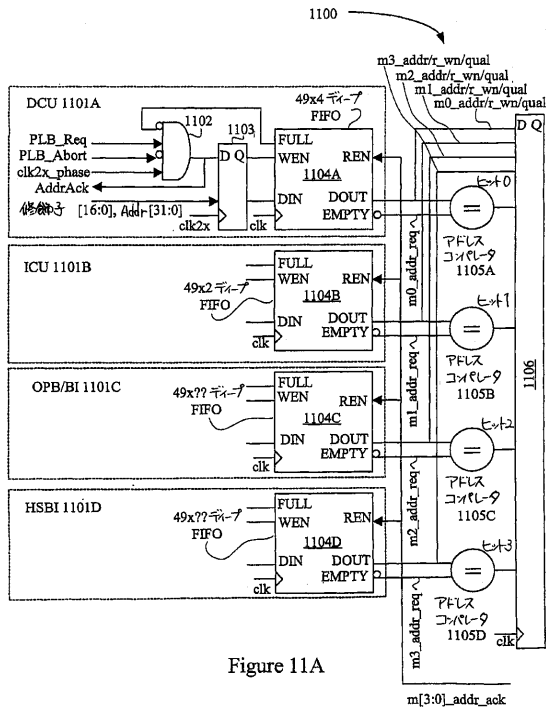


Figure 11A

【図 11 B】

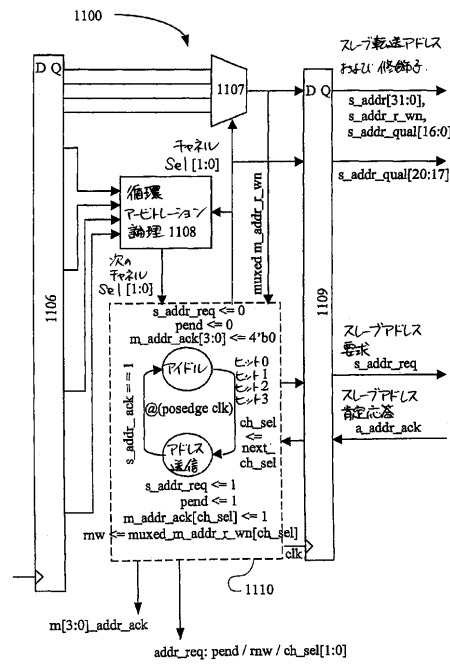


Figure 11B

Figure 11A Figure 11B
図 11A 図 11B

【図 12 A】

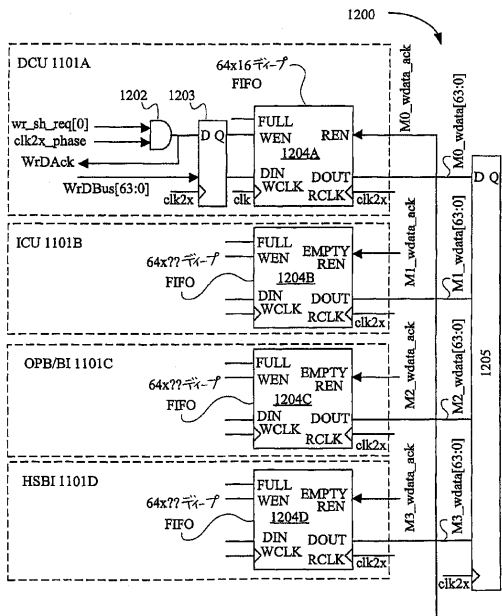


Figure 12A

【図 12 B】

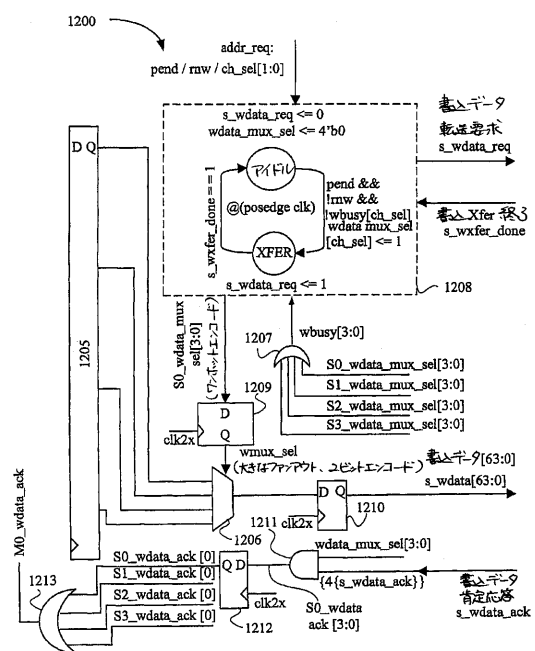


Figure 12B

Figure 12A Figure 12B
図 12A 図 12B

【図 13 A】

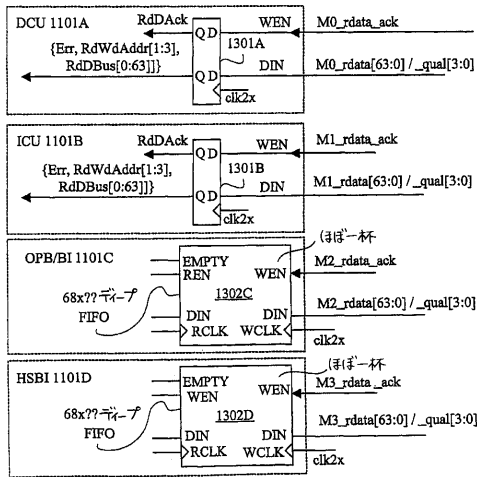


Figure 13A

Figure 13A Figure 13B
RdAck

【図 13 B】

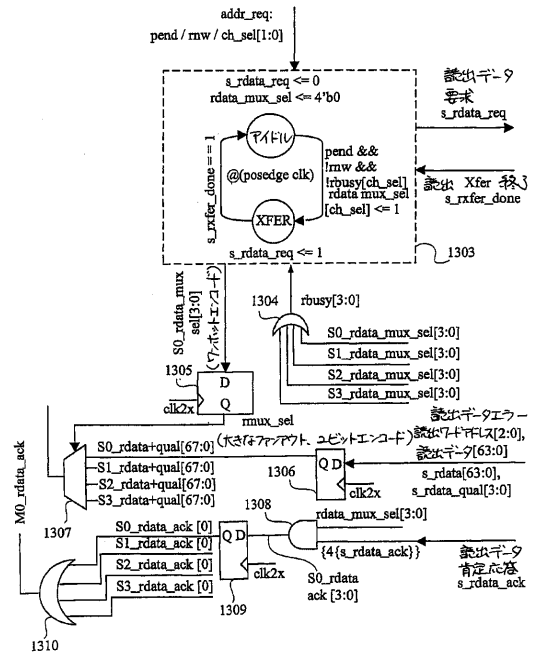


Figure 13B

【図 14】

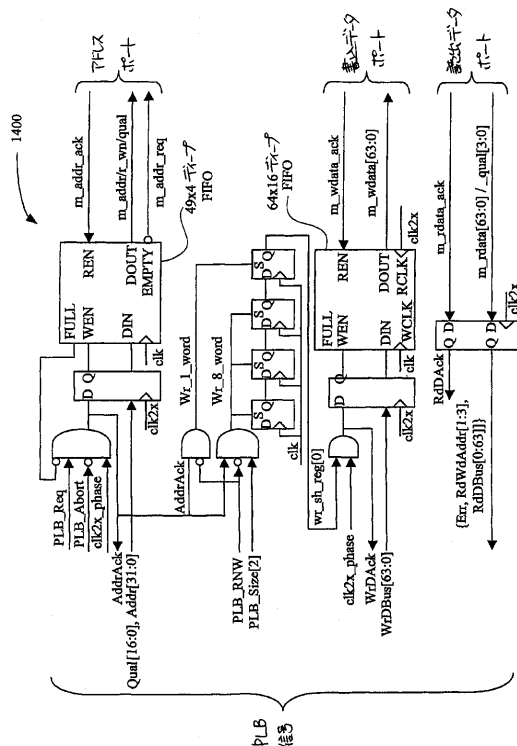


Figure 14

【図 15】

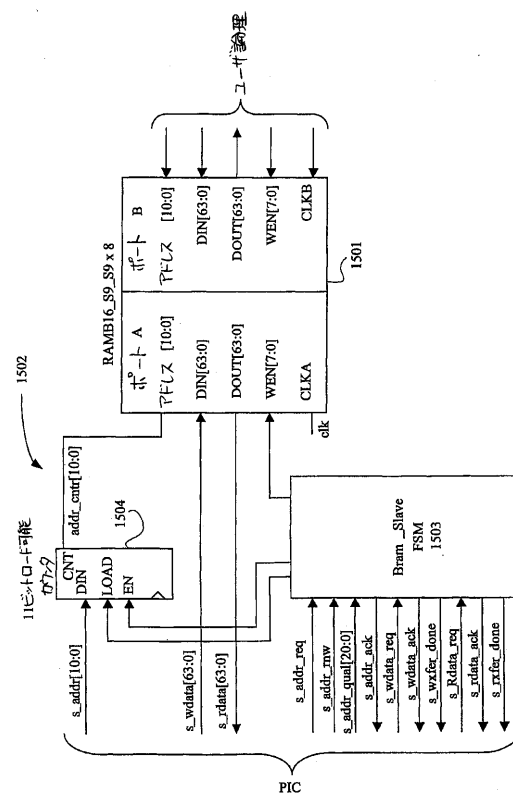


Figure 15

【図 16】

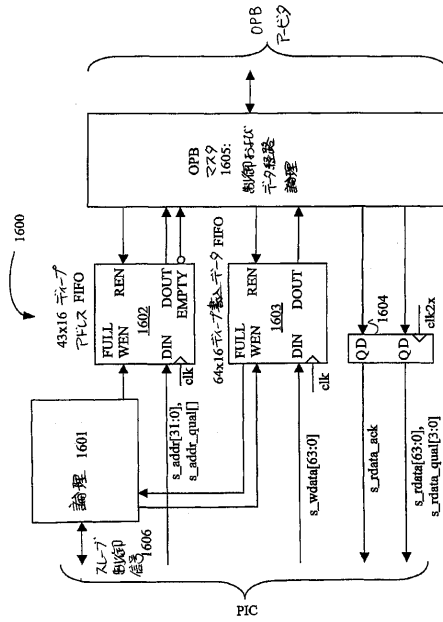


Figure 16

【図 17】

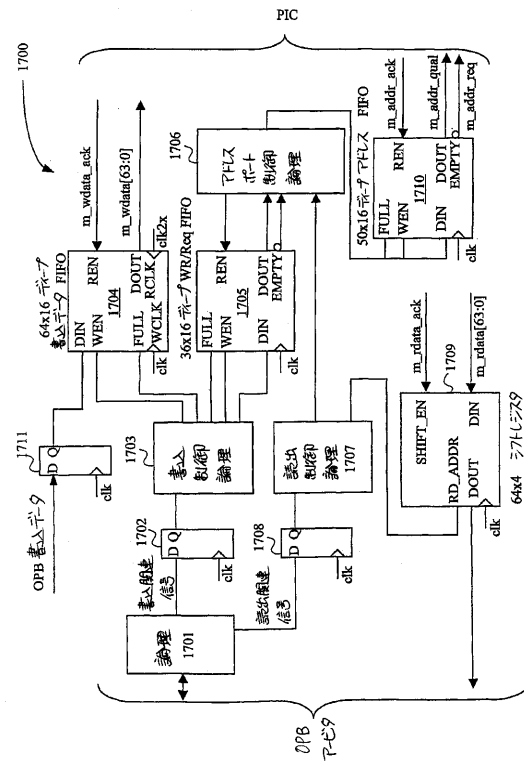


Figure 17

【図 18 A】

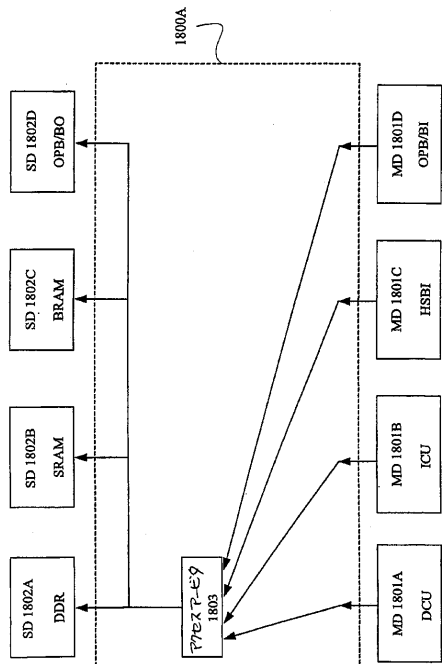


Figure 18A

【図 18 B】

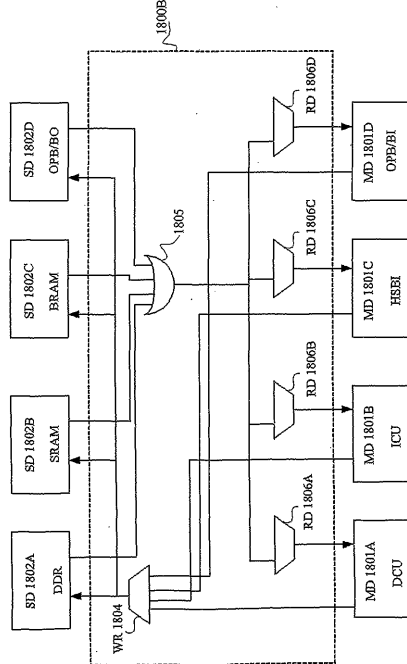


Figure 18B

【図 19 A】

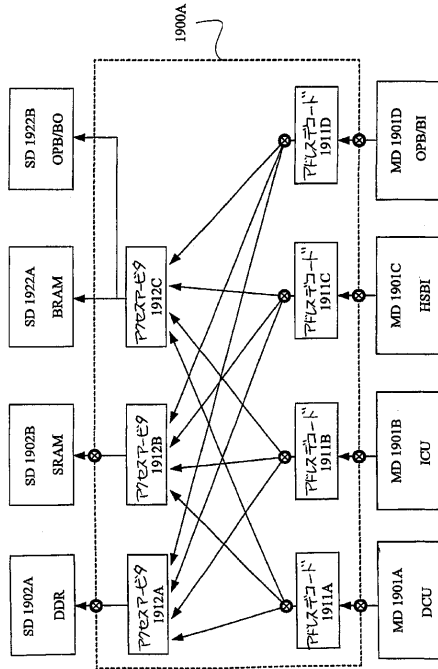


Figure 19A

【図 19 B】

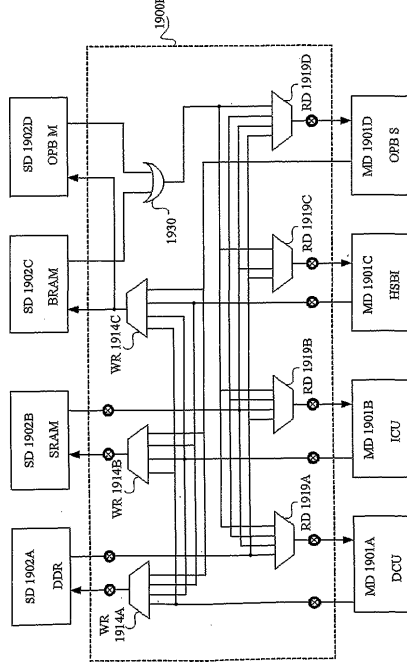


Figure 19B

フロントページの続き

(74)代理人 100098316

弁理士 野田 久登

(74)代理人 100109162

弁理士 酒井 将行

(72)発明者 ダオ, カン・キム

アメリカ合衆国、 9 5 1 3 1 カリフォルニア州、 サン・ノゼ、 モーニング・スター・ドライブ、
6 0 9

(72)発明者 バクスター, グレン・エイ

アメリカ合衆国、 9 5 0 3 3 カリフォルニア州、 ロス・ガトス、 ステージ・ストップ・ウェイ、
2 4 0 5 5

審査官 清水 稔

(56)参考文献 特開2000-105759(JP, A)

特表2000-509948(JP, A)

特表平07-503804(JP, A)

特開平08-328824(JP, A)

特開平08-069447(JP, A)

特開2001-092648(JP, A)

(58)調査した分野(Int.Cl., DB名)

H03K 19/173