

FPGA による PCI バス・インターフェース回路の実装

小野 雅晃¹

筑波大学システム情報工学等支援室 (装置開発班)

〒305-8573 茨城県つくば市天王台 1-1-1

概要

クラスター型コンピュータ向けネットワーク・インターフェース MaestroNI 用に PCI バス・インターフェース回路を開発した。PCI (Peripheral Component Interconnect) バス・インターフェース回路は FPGA (Field Programmable Gate Array) 内に実装され、PCI ターゲット転送、PCI マスタ転送を処理することができる。PCI バス・インターフェース回路の PCI ターゲット転送は単一転送に制限されるが、PCI マスタ転送はバースト転送に対応している。PCI バスの動作周波数は 66 MHz、データ幅は 64 ビットであり、理論上のスループットの最大値は 528 Mbytes/sec である。PCI バス・インターフェース回路は、市販の IP (Intellectual Property) を使用せず、PCI の規格を基に全て独自に設計・開発した。現在、Maestro2 および Maestro3 ネットワーク・インターフェース基板の FPGA に実装されて、NI 基板とホスト・パーソナル・コンピュータ (ホスト PC) を結ぶインターフェースとして使用されている。

キーワード：FPGA、PCI バス、Maestro

1. はじめに

PCI バスは、パーソナル・コンピュータ(PC)の拡張スロットとして親しまれているバス規格である。2 枚目のグラフィックカードや、イーサネットカード、A/D コンバータカード、キャプチャカード、いろいろな用途の拡張基板用のバスとして使用されている。PCI バスは PCI SIG (Peripheral Component Interconnect Special Interest Group) により 1992 年に PCI 仕様書第 1 版が出版され、スタートを切った。その後、1999 年に第 2 世代として、データ転送速度を向上させた PCI-X バスが発表された。PCI-X バスは主にサーバー機器用の拡張ボード規格として使用された。2002 年には第 3 世代として、高速シリアル通信を採用した PCI-Express が発表された。PCI-Express が主な拡張ボード・インターフェース規格として採用された現在でも、PCI バス・コネクタは、PC に必ず搭載されている。

第 4 回筑波大学技術職員技術発表会で発表した Maestro2 ネットワーク・インターフェース (Maestro2NI)^[1]と第 6 回筑波大学技術職員技術発表会で発表した Maestro3 ネットワーク・インターフェース (Maestro3NI)^[2]は、PCI バスに接続される拡張基板である。2 つのネットワーク・インターフェース(NI)は、ホスト PC のマザーボード上の 64 ビット幅、3.3 V 電圧の PCI バス・コネクタに挿入されて使用される。PCI バス・インターフェース信号は、NI

の PCI カード・エッジコネクタから FPGA にプリント配線され、FPGA 中に実装された PCI バス・インターフェース回路に接続されている。PCI バス・インターフェース回路は、NI とホスト PC の間で PCI バスを通してデータ転送を行う。データ転送は 2 種類あり、1 つが PCI ターゲット転送、もう 1 つが PCI マスタ転送である。PCI ターゲット転送は、ホスト PC が要求元となるデータ転送である。PCI マスタ転送は、NI が要求元となるデータ転送である。

PCI バス・インターフェース回路の PCI ターゲット転送は単一転送に制限されるが、PCI マスタ転送はバースト転送に対応している。PCI バスの動作周波数は 66 MHz、データ幅は 64 ビットである。

2. PCI バスの概要

2.1 PCI バスの特徴

PCI バスには次の特徴がある。

1. 64 ビットまたは 32 ビット・データ幅の同期バスである。
2. 割り込み信号とエラー信号以外のほとんどの制御信号にサステインド・トライステート方式を使用している。
3. メモリ空間、IO 空間、コンフィギュレーション空間を持っている。
4. プラグ・アンド・プレイに対応できる。
5. アドレスバスとデータバスが共用である。
6. データ転送を始める際に、マスタ・デバイスが C/BE# にコマンドを出力するので、ターゲット・デバイスは転送の種類をあらかじめ知ることができる。
7. バイト・イネーブルを持っていて、バイトごとに書き込みができる。
8. 制御信号のほとんどがアクティブ・ロー信号である。

2. のサステインド・トライステート方式は、トライステート方式の一種であり、アクティブ・ローの制御信号に使用されている。サステインド・トライステート信号は、マザーボード側で抵抗によってプルアップされている。アクティブ・ローの信号は、デバイスによって 0 レベル (アサート) にドライブされてから、1 レベル (ディアサート) に戻される動作を繰り返して、データ転送のやり取りを行う。その際に、ドライバのドライブを 0 レベルの間に OFF し、プルアップ抵抗で浮遊容量を充電して 1 レベルにすると、1 レベルに達するまでの時間が長くなる。その点を改良するために、サステインド・トライステート方式は、1 レベルにドライブしてから

¹ E-mail: ono@sie.tsukuba.ac.jp; Tel: 029-853-5195

ドライブを OFF する。図 1 に通常のトライステートバッファの波形とサステインド・トライステートバッファの波形を示す。

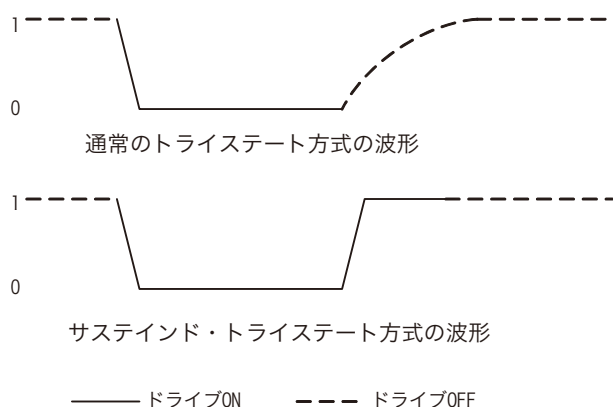


図 1 通常のトライステート方式の波形とサステインド・トライステート方式の波形の比較

2.2 PCI バスの空間

PCI バスの空間にはメモリ空間、IO 空間、コンフィギュレーション空間の 3 つがある。

メモリ空間は、ホスト PC のプロセッサのメモリ空間にマップされる。通常は BIOS(Basic Input/Output System)によって、PCI ボードで使用するメモリ領域をマップする。PCI バス・インターフェース回路では、Maestro2NI および Maestro3NI の SDRAM 領域、ネットワーク・バッファ領域、FLASH ROM (Flash Read Only Memory)、設定用レジスタの 4 つをマップしている。

IO 領域は、x86 プロセッサの IO と同様に、64 K バイトの領域がある。IO 領域は PCI バス・インターフェース回路ではサポートされていない。

コンフィギュレーション空間³⁾は、PCI ボードのステータスの監視や設定を行う領域で、8 ビット長のアドレス空間を持っている。図 2 にコンフィギュレーション空間のアドレス・マップを示す。コンフィギュレーション空間のレジスタはコンフィギュレーション・レジスタと呼ばれる。

アドレス 00h (h は 16 進数を表す) のベンダ ID は PCI ボード (PCI バスに接続されている基板を PCI ボードと呼ぶことにする) を作成したメーカーの固有コードである。ベンダ ID は PCI SIG によって、各メーカーに割り当てられる。

アドレス 02h のデバイス ID はメーカーのいわば型番となる。

アドレス 09h ~ 0Bh のクラスコードは PCI ボードの種類を表す。たとえば、PCI バス・インターフェース回路では基本クラスがネットワーク・コントローラを示す 02h、サブクラスがその他のネットワーク・コントローラを示す 80h、プログラミング・インターフェースが 00h となる。

ホスト PC のデバイス・ドライバは、ベンダ ID、デバイス ID、クラスコードなどから自分の制御すべきボードがどれなのかを判定することができる。

アドレス 04h のコマンドは、一般的にコマンド・レジスタと呼ばれている。コマンド・レジスタは、

ビット				アドレス
31	16	15	0	
デバイス ID		ベンダ ID		00h ^{a)}
ステータス		コマンド		04h
クラス・コード			リビジョン ID	08h
BIST	ヘッダ・タイプ	レイテンシ・タイム	キャッシュ・ライン・サイズ	0Ch
ベース・アドレス・レジスタ (SDRAM)				10h
ベース・アドレス・レジスタ (ネットワーク・バッファ)				14h
ベース・アドレス・レジスタ (FLASH ROM)				18h
ベース・アドレス・レジスタ (設定用レジスタ)				1Ch
ベース・アドレス・レジスタ (未使用)				20h
ベース・アドレス・レジスタ (未使用)				24h
カードバス CIS ポインタ				28h
サブシステム ID		サブシステムベンダ ID		2Ch
拡張 ROM ベースアドレス				30h
予 約			新機能ポインタ	34h
予 約				38h
最大レイテンシ	最小グラント	PCI インタラプトピン	インタラプトライン	3Ch
デバイス固有レジスタ領域 (設定用レジスタ、Block RAM)				4Ch ~ FCh

a) 00h の h は hexadecimal (16 進数) を表す

図 2 コンフィギュレーション空間のアドレス・マップ

PCI ボードのさまざまな機能を制御するためのレジスタである。アドレス 06h のステータスは、一般的にステータス・レジスタと呼ばれ、読み出しのみのレジスタである。ステータス・レジスタは、PCI ボードの状態を表す。

アドレス 10h ~ 24h の領域は、ベース・アドレス・レジスタに割り当てられている。ベース・アドレス・レジスタは、32 ビット長で、PCI ボード上のメモリ、IO (Input/Output) の領域の大きさとホスト PC でのアドレスを示す。このレジスタの 1 つ 1 つがメモリ空間、IO 空間に対応する。ベース・アドレス・レジスタの示す空間が、どちらの空間かを表すためにビット 0 が使用される。ビット 0 が 0 のときはメモリ空間となり、1 のときは IO 空間となる。ホスト PC のコンフィギュレーション・アクセスが終了した時点でのベース・アドレス・レジスタは、ホスト PC のアドレス・マップに対応するアドレスが入力されている。Maestro3NI では、10h に SDRAM、14h にネットワーク・バッファ、18h に FLASH ROM、1Ch に設定用レジスタが割り当てられている。その他のベース・アドレス・レジスタ領域は未使用のため 0 に固定されている。ベース・アドレス・レジスタの設定方法は第 5 章で述べる。

3. PCI バスの信号

PCI バスは 184 本の信号線を持っている。信号線は、システム信号、アドレス・データ・属性信号、制御信号、アービトレーション信号、エラー関連信

号、割り込み信号、キャッシュ・サポート信号、JTAG/バウンダリ・スキャン信号に分けられる。

システム信号は CLK、RST#で構成される。アドレス・データ・属性信号は AD[63:00]、C/BE[7:0]、PAR、PAR64で構成される。制御信号は、FRAME#、IRDY#、TRDY#、STOP#、LOCK#、IDSEL、DEVSEL#で構成される。アービトラージ信号は、REQ#、GNT#、REQ64#、ACK64#で構成される。エラー関連信号は、PERR#、SERR#で構成される。インタラプト信号は、INTA#、INTB#、INTC#、INTD#で構成される。キャッシュ・サポート信号は、SBO#、SDONE#で構成される。最後の JTAG/バウンダリ・スキャン・ピンは TCK、TDI、TDO、TMS、TRST#で構成される^[3]。

4. PCI バスの信号プロトコル

Maestro3NI の PCI ターゲット転送と PCI マスタ転送、コンフィギュレーション・サイクルを例にとって、PCI バスの信号プロトコルを説明する。

4.1 PCI ターゲット書き込み転送

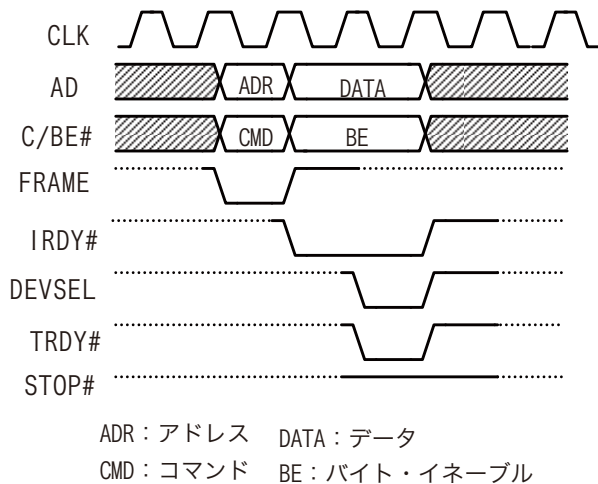


図 3 PCI ターゲット・ライト

図 3 はホスト PC から Maestro3NI への PCI ターゲット書き込み転送 (PCI ターゲット・ライト) を表している。図 3 で、実線はドライブされている信号を示している。破線は、バスの信号線がプルアップ抵抗でプルアップされているだけで、どこのドライバからもドライブされていない状態を表している。バスのハッチングは、バスがハイ・インピーダンス状態にあることを示している。なお、インタラプトなどの信号を除く大部分の信号は CLK の立ち上がり同期して変化する。

C/BE#、FRAME#、IRDY#はマスタ・デバイス (この場合はホスト PC) が常時ドライブする信号である。PCI ターゲット・ライトの場合は、AD もマスタ・デバイスが常時ドライブする。

ターゲット・デバイス (この場合は Maestro3NI) は、FRAME#が 0 になってアサートされたときに動作を開始する。ターゲット・デバイスは FRAME#が 0 になり、アサートされたときに、AD からアドレスを取得する。それと同じタイミングで、C/BE#にはコマンドが出力されている。コマンドはメモリや IO

やコンフィギュレーションの書き込み、読み出しをコードでターゲット・デバイスに知らせる。取得したアドレスが自分のベース・アドレス・レジスタで示されるアドレスだったら、ターゲット・デバイスは DEVSEL#をアサートして応答を開始する。Maestro3NI の PCI バス・インターフェース回路では DEVSEL#信号は、FRAME#がアサートされたクロックの 1 クロック後にアサートを開始している (デコード・タイミングは中速)。

マスタ・デバイスは FRAME#の後のクロックで、AD に書き込むデータ、C/BE#にバイト・イネーブルを出力し、IRDY#をアサートする。

ターゲット・デバイスは DEVSEL#と同じタイミングで、書き込みデータを受け入れ可能なことを示すために TRDY#をアサートする。IRDY#と TRDY#が同時にアサートされたタイミングで、PCI データ転送が成立する。図 3 (Maestro3NI) の場合は、単一転送であるため、1 つデータ転送が終了した時点で PCI ターゲット転送を完了する。FRAME#、IRDY#、DEVSEL#、TRDY#、STOP#はサステインド・トライステート信号なので、0 から 1 にドライブしてからハイ・インピーダンス状態となる。AD と C/BE#は通常のトライステート信号であるので、図 3 の斜線の部分でハイ・インピーダンス状態となる。

4.2 PCI マスタ転送の読み出し

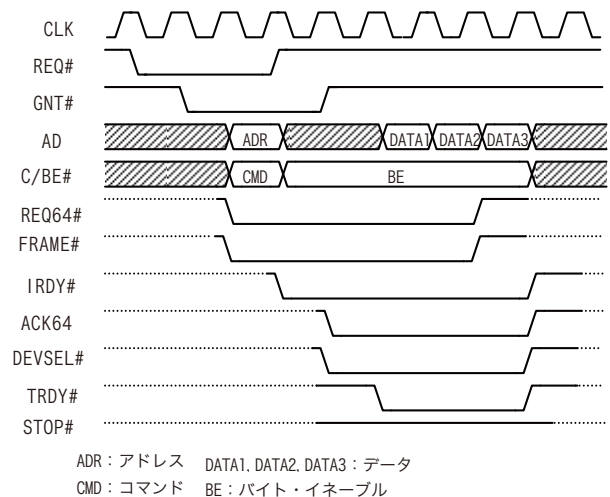


図 4 PCI マスタ・リード

図 4 は Maestro3NI からホスト PC への PCI マスタ転送の読み出し (PCI マスタ・リード) を表している。図 3 から追加された信号のうち、REQ#、GNT#はホスト PC の PCI バス・アービトラージ回路がドライブする。REQ64#はマスタ・デバイス (この場合は Maestro3NI) がドライブし、ACK64#はターゲット・デバイス (この場合はホスト PC) がドライブする。

図 4 の PCI マスタ転送は、Maestro3NI のプロセッサがホスト PC のメモリを読み出すために、PCI バス・インターフェース回路に 1 ワード (32 bit) のリードを要求した場合である。PCI バス・インターフェース回路は、初めに、バスを要求するために REQ#を 0 にアサートする。一般的にホスト PC のチップセットに内蔵されている PCI バス・アービトラージ

オン回路は、PCI バス・インターフェース回路から出されたバス要求を許可するために GNT#を 0 にアサートする。これでバスの使用が許可されたが、他のデバイスがまだ PCI バスを使用している可能性があるため、マスタ・デバイスは FRAME#、IRDY#を監視して、どちらも 1 にディアサートされたときにバスが使用されていないと認識する。バスが使用されていないときに、マスタ・デバイスは FRAME#と、もし 64 ビットアクセスをしたいときは REQ64#をアサートする。PCI バス・インターフェース回路のマスタ・アクセスはすべて 64 ビットで行われるので、図 4 では REQ64#がアサートされている。

図 3 の PCI ターゲット・ライトとの違いの 1 つは、FRAME#が連続してアサートされていることである。図 4 のように、連続してアサートする場合は、連続的にデータを転送（バースト転送）できる。3 つの 64 ビットデータをリードしているが、そのうちの 32 ビット分のデータのみ使用して、その他のデータは捨てている。これは、データバスのパイプライン化によって、バースト転送時のスループットを確保するために必要な措置である。さらに、ライトの場合は、マスタ・デバイスがトランザクションの間すべて、AD バスをドライブしていたが、リードの場合はアドレスを出力した後でドライブを OFF し、ターゲット・デバイスが AD バス上にリード・データを出力する。

PCI マスタ・リードの場合も PCI ターゲット・ライトの時と同様に、双方の制御信号のやり取りで、データ転送を行う。IRDY#と TRDY#が同時にアサートされた時が、データ転送が成功した場合である。マスタ・デバイスはデータ転送終了の 1 つ前に FRAME#を 1 にディアサートする。最後の 1 つの転送が終了したら、マスタ・デバイスは IRDY#をディアサートした次のクロックで、ドライバをハイ・インピーダンス状態にする。

ターゲット・デバイスの制御信号線は図 3 に比べて ACK64#が追加されている。ACK64#は REQ64#に対する応答としてアサートされる。つまり、ターゲット・デバイスが 64 ビット幅のデータ転送に対応している場合は、ACK64#をアサートする。ACK64#がアサートされない場合、マスタ・デバイスは 64 ビット幅のデータ転送を中断して、32 ビット幅のデータ転送に変更する。ただし、Maestro3NI の PCI バス・インターフェース回路は、マスタ・アクセス時の 32 ビット幅データ転送に未対応である。

その他の制御信号は、バースト転送として連続的にアサートされる以外は、特に変更はない。

4.3 コンフィギュレーション・サイクル

図 5 はコンフィギュレーション・書き込みサイクル（コンフィギュレーション・ライト）を表している。

コンフィギュレーション・サイクルは、コンフィギュレーション・レジスタにアクセスするために使用される。ホスト PC が電源 ON で、ブートしたときには、コンフィギュレーション・レジスタのベース・アドレス・レジスタは値が入力されていない。よって、この状態では、PCI ボードはアクセスに回答できない。このような場合でも、コンフィギュレーション・サイクルには、応答することができる。

それは、PCI スロットごとにユニークな IDSEL という信号が PCI データ転送の始まりの時にアサート(1)されるかどうかを監視すれば、データ転送の始まりを認識できるからである。コンフィギュレーション・ライトは、AD にアドレスが 8 ビットのみ出力されているなどの違いはあるが、その他はほとんど PCI ターゲット・ライトと同様である。

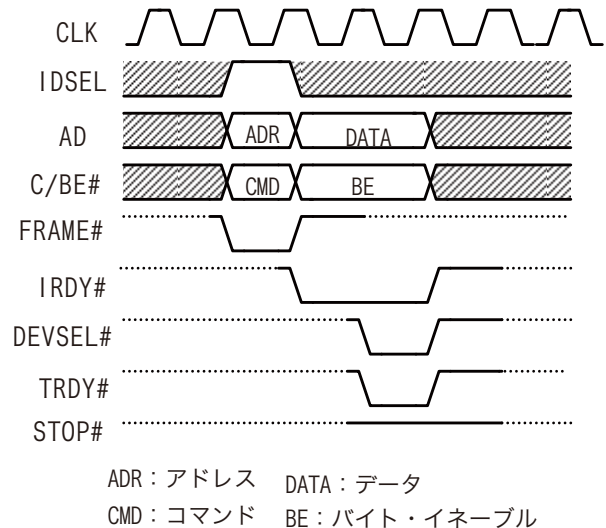


図 5 コンフィギュレーション・ライト

5. ホスト PC 起動時のコンフィギュレーション

ホスト PC を電源 ON すると起動する BIOS (Basic Input/Output System)が、PCI ボードにコンフィギュレーション・サイクルを発生させて、PCI ボード上の各コンフィギュレーション・レジスタを読み、ベース・アドレス・レジスタを操作して、PCI ボードのコンフィギュレーションを行う。

ベース・アドレス・レジスタは、メモリ領域または IO 領域をホスト PC のメモリ空間または IO 空間に確保するためのレジスタである。ベース・アドレス・レジスタには、メモリ空間では 4 ビット、IO 空間では 2 ビットの予約領域がある。この予約領域は最下位ビットから割り当てられており、割り当てられるメモリ領域の最小値は 16 バイト、IO 領域は 4 バイトとなる。予約領域の上のビットは、PCI ボードの要求するメモリ領域、IO 領域の大きさによって 0 に固定される。固定された以外の上位ビットには、アドレスを記憶する DFF (Delay Flip Flop) が実装される。例えば、Maestro3NI のベース・アドレス・レジスタ 0 (コンフィギュレーション・アドレス 10h ~13h) は SDR SDRAM のアドレスが入力される。Maestro3NI の SDR SDRAM の容量は 128 M バイトなので、必要とするアドレスビットは 27 ビットである。よって、ベース・アドレス・レジスタ 0 の 0 に固定されるビットはビット 4 からビット 26 までとなる。後のビット 27 からビット 31 までは、ホスト PC から割り当てられるアドレスを記憶するために、DFF を実装した。

BIOS によるベース・アドレス・レジスタの設定手順は以下のとおりである。

1. ホスト PC は、コンフィギュレーション・ライトを発行して、PCI ボードのベース・アドレス・レジスタにオール 1 を書き込む。
2. ホスト PC は、コンフィギュレーション・リードを発行して、PCI ボードの書き込んだベース・アドレス・レジスタを読み出す。
3. 読み出したデータのビット列のどこまでが 0 か (1 を書いたはずなのに) を調べて、そのベース・アドレス・レジスタで要求された領域の大きさを確認する。
4. 要求された領域をホスト PC のアドレス・マップ上に確保して、そのアドレスをコンフィギュレーション・ライトで、もう一度 PCI ボードのベース・アドレス・レジスタに書き込む。

このように、ベース・アドレス・レジスタの設定を通して、PCI ボードのリソースがホスト PC のアドレスにマップされ、ホスト PC のプロセッサからアクセスが可能となる。

6. PCI バス・インターフェース回路

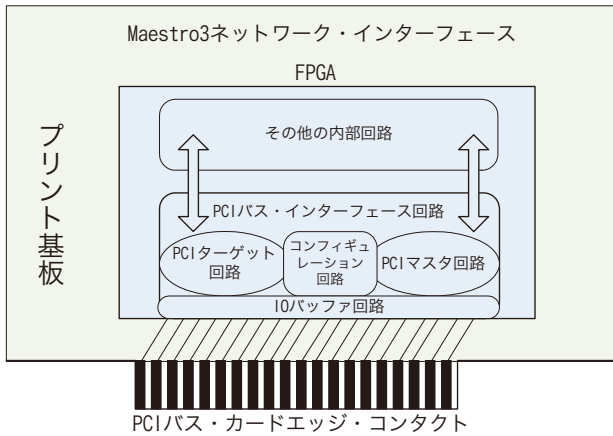


図 6 PCI バス・インターフェース回路のブロック図



図 7 Maestro3NI の PCI バス、FPGA 部分

PCI バス・インターフェース回路は Maestro3 ネットワークの一部として、主にホスト PC のメモリと Maestro3 ネットワークのデータのやり取りに使用されている。

PCI バス・インターフェース回路のブロック図を図 6 に示す。Maestro3NI の PCI バス、FPGA 部分の写真を図 7 に示す。PCI バス・インターフェース回

路はコンフィギュレーション回路、PCI ターゲット回路、PCI マスタ回路と IO バッファ回路で構成されている。

6.1 PCI ターゲット回路

PCI ターゲット回路は、単一転送のみをサポートし、バースト転送はサポートしていない。ホスト PC のプロセッサからのアクセスは 32 ビット・アクセスなので、FPGA 内の 64 ビット・バスとのインターフェースを図るため、ワードレーン変換機能を備えている。図 8 に PCI ターゲット回路のデータパス・ブロック図を示す。

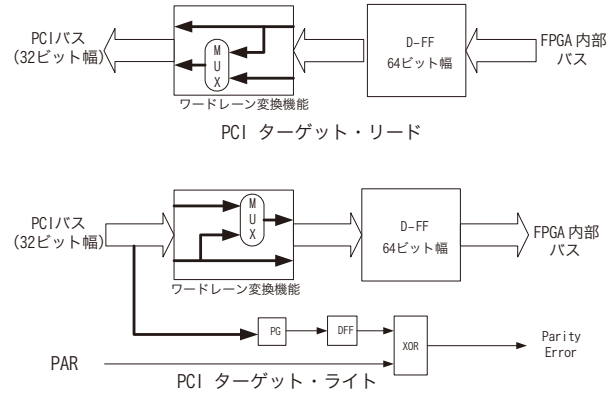


図 8 PCI ターゲット回路のデータパス・ブロック図

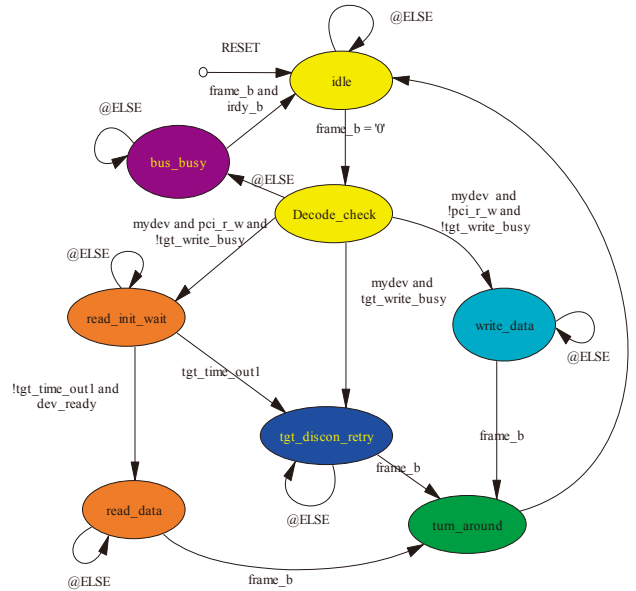


図 9 PCI ターゲット回路のメイン・ステート・マシン

PCI ターゲット・リードの場合は、FPGA 内部バスの上位 32 ビットと下位 32 ビットをマルチプレクサによって切り替え、32 ビット分を下位の PCI バスに送る。PCI ターゲット・ライトの場合は、PCI バスからアドレスを取得して、FPGA 内部バスの上位 32 ビットに出力するか、下位 32 ビットに出力するかを決定する (図 8 参照)。

PCI ターゲット回路のメイン・ステート・マシンを図9に示す。

リセット後は、idle ステートからスタートする。FRAME# (frame_b) が0 にアサートされたときに、Decode check ステートに遷移し、自分のアドレス(ベース・アドレス・レジスタに設定してあるアドレス)にヒットしたかを調べる。自分のアドレスにヒットしていない場合は、bus_busy ステートに遷移し、そのPCIバス・トランザクションが終了するまで、そのステートで待機する。自分のアドレスにヒットした場合は、ライトの場合とリードの場合に分かれる。

ライトの場合に、以前のトランザクションが終了している場合は write_data ステートに遷移し、FPGA 内部バスに書き込みデータを送る。以前のトランザクションが終了していない場合は、tgt_discon_retry ステートに遷移して、リトライ動作を行う。

リードの場合には、read_init_wait ステートに遷移して、FPGA 内部バスからリード・データが到着するのを待つ。リード・データが到着したら、reda_data ステートに遷移して、PCI バス上にリード・データを出力する。

ライトの場合も、リードの場合もデータ転送が終了したら、turn_around ステートに遷移する。ここでは、サステインド・トライステート状態のために制御信号をデアサート (1) する。

tgt_discon_retry ステートは、リトライで終了する場合に遷移するステートである。リトライで終了する場合には、ライト時に以前のトランザクションが終了していない場合や、read_init_wait ステートで処理を待っているときにタイムアウトでPCIバス・トランザクションを終了する場合がある。

図9のステート・マシンを使用して、図3のPICターゲット・ライトを実現している。

6.2 PCI マスタ回路

PCI マスタ回路は、64 ビット幅のデータ転送のみをサポートしている。IO バッファ回路、PCI ターゲット回路の一部を含めた入出力部分のブロック図を図10に示す。

図10中のIOB (Input Output Block) は、FPGA のパッド近くに実装された入出力ブロックである。IOB内には入力用、出力用、トライステート・イネーブル用の DFF が内蔵されている。PG (Parity Generator) は、偶数パリティを生成する回路である。

図10の上部が主に PCI マスタ・ライトおよび、PCI ターゲット・リードが使用する回路である。図10の下部が主に PCI マスタ・リードおよび、PCI ターゲット・ライトが使用する回路である。PCI マスタ・リード、ライトともに 512 深度の FIFO (first-in first-out) に転送データを蓄積している。PCI マスタ・ライトの場合は、ライト用 FIFO から IOB を通じて AD に出力し、PCI マスタ・リードの場合は IOB を通じてリード用 FIFO に入力している。どちらの AD も同じ FPGA の端子であり、図では便宜上、分けて書かれている。

PCI マスタ・ライトのリード、ライトともに、PCI 側と FPGA 側にそれぞれ転送数カウンタを持っている。そのため、FIFO の入力、出力両側の転送数がわかるようになっている。

PAR は AD[31:0] と C/BE#[3:0] の偶数パリティを出力する。PCI マスタ・アクセスの場合と、PCI ターゲット・リードの場合に PCI バス・インターフェース回路からパリティを出力する。

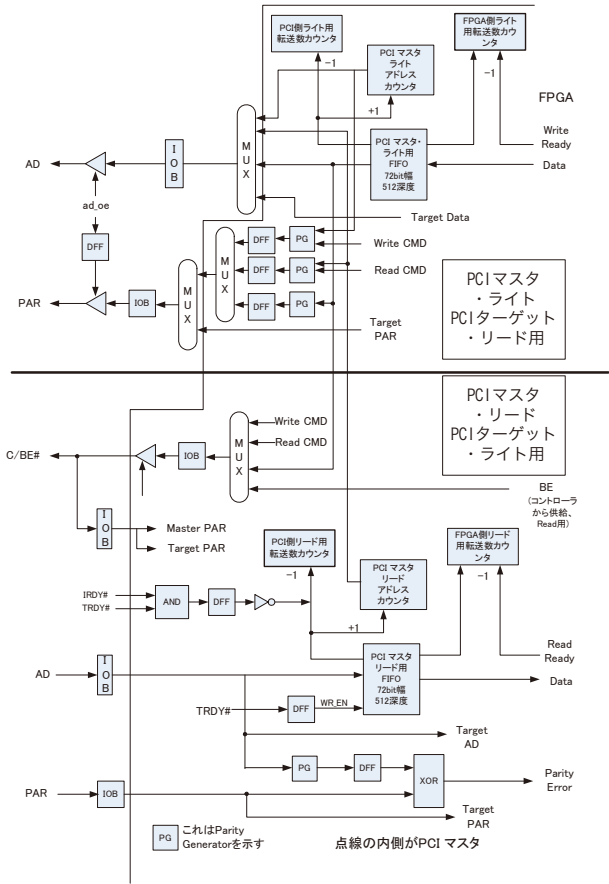


図10 PCI マスタ回路入出力部分ブロック図

C/BE# は PCI バス・アクセスの最初でコマンドを出力し、その後はバイト・イネーブルを出力する必要がある。そのため、マルチプレクサ(MUX)で入力を切り替えている。

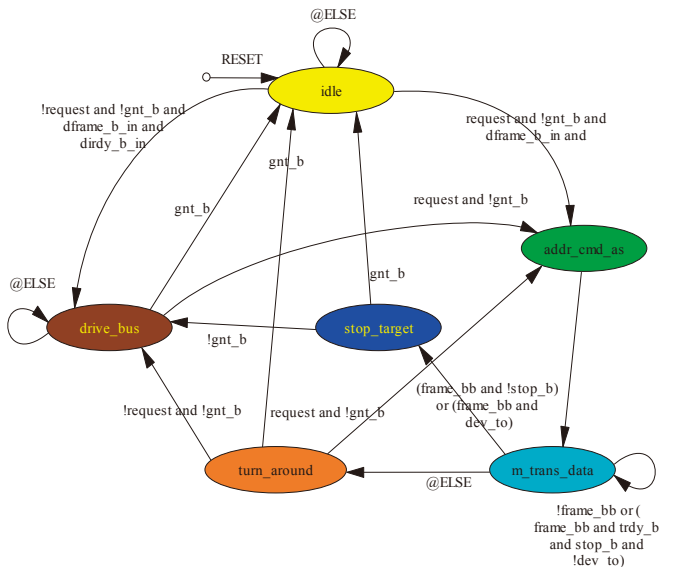


図11 PCI マスタ回路のメイン・ステート・マシン

図 11 に PCI マスタ回路のメイン・ステート・マシン^[4]を示す。

リセット後は、idle ステートからスタートする。REQ#(request) をアサートして、GNT#(gnt_b) がアサートされて、FRAME#(frame_b) と IRDY#(irdy_b) がディアサートされたときに、addr_cmd_as ステートに遷移する。addr_cmd_as ステートで、AD にアドレス、C/BE#にコマンドを出力してデータ転送が始まる。addr_cmd_as ステートから無条件に遷移する m_trans_data ステートでは、データ転送を行う。正常に終了するときには、turn_around ステートに遷移して、ドライブしている制御信号(FRAME#, IRDY#) を 1 にディアサートする。ターゲット・デバイスから STOP#をアサートされたとき、または、ターゲット・デバイスから応答がなかった場合は、stop_target ステートに遷移して、ディスクネクスト処理を行う。turn_around ステート、または、stop_target ステートで GNT#がアサートされていたら、バス・パーキングを行う必要があるので drive_bus ステートに遷移する。バス・パーキングとは、PCI バス・トランザクションを行わずに AD と C/BE#をドライブすることである。turn_around ステート、または、stop_target ステートで GNT#がディアサートされていたら、idle ステートに戻る。

以上のような動作を行い、PCI マスタ転送を行っている。

6.3 コンフィギュレーション回路

コンフィギュレーション回路は、図 2 に示したコンフィギュレーション空間に割り当てられたコンフィギュレーション・レジスタを実装している。

6.4 IO バッファ回路

IO バッファ回路は、PCI ターゲット回路および PCI マスタ回路と FPGA チップのパッドの間にあるバッファ回路である。IOB 内の DFF を積極的に使用して、タイミング・バジェットを改善することもできる。

7. 工夫および苦勞した箇所について

PCI バス・インターフェース回路を実装する上で、工夫および苦勞した箇所は以下の通りである。

1. PCI バス 66 MHz の仕様によると、入力信号のセットアップ時間は 3 nsec、出力信号のクロックからの出力時間は 6 nsec である。これは大変厳しい数値である。PCI バスの規格によると、入力信号を同じクロック内に判定する必要がある。この規格からすると、入力信号は必ず組み合わせ回路を通過して DFF に入力する必要があるため、セットアップ時間を確保することが難しい。PCI バス・インターフェース回路では、DCM (Digital Clock Manager) を使用して、クロックの位相をシフトし、クロックからの出力時間の余裕をセットアップ時間に割り振っている。それでもまだ、3 nsec の規格は満足しないが、動作には問題はない。

2. PCI バス規格書には、64 ビット幅の PCI バスも 32 ビット幅の PCI バスのトランザクションにも対応しなければならないとある。PCI バス・インターフェース回路の PCI マスタ・アクセスは 64 ビットのみ対応する。余計な回路に対応しないことによって、回路構成を簡単にし、FPGA のクリティカルパスの遅延を小さくすることができる。
3. PCI マスタ回路の出力用 FIFO は、データのクロックからの出力時間を確保するために、出力信号は DFF を使用してラッチされている。そのため、PCI データ転送の終了時にデータ転送なしのディスクネクストをされると IOB の DFF に残っているデータが消失する危険がある。この問題を回避するために PCI マスタ・ライト用 FIFO は、リワインド端子つきの FIFO を自作して使用した。
4. PCI バス・インターフェース回路の作成中に、ホスト PC のチップセットが変更になると、PCI バスの反応も異なり、PCI バス・インターフェース回路のバグが顕在化することが多かった。バグの発生する条件が限定的であるため、バグの原因を突き止めるのが困難であった。そのような場合は、シミュレーションで再現を試みた。それで再現しない場合は、検出回路を FPGA に内蔵し、FPGA 内蔵ロジック・アナライザ (ChipScope) を使用してバグを特定した。このような場合は、バグの原因の特定が非常に難しい。

8. まとめ

本報告では、PCI バスの概要について述べた後、自作の PCI バス・インターフェース回路について述べた。現在、PCI バス・インターフェース回路は、Maestro2NI と Maestro3NI の PCI バス・インターフェースとして使用されている。PCI バス・インターフェース回路のビット幅は 64 ビット、動作周波数は 66 MHz である。理論上のスループットの最大値は 528 Mbytes/sec である。

9. 謝辞

Maestro2 および Maestro3 システムの開発に参加の機会を与えて頂いた、筑波大学システム情報工学研究科コンピュータサイエンス専攻の和田耕一教授に深く感謝いたします。

参考文献

- [1] 小野雅晃, Maestro2 ネットワークインターフェースの開発, 筑波大学技術報告 25 (2005) 42-49.
- [2] 小野雅晃, Maestro3 クラスタネットワーク用基板の設計, 筑波大学技術報告 27 (2007) 37-44.
- [3] 滝誠一 他, PCI バスの詳細と応用へのステップ, CQ 出版社 OpenDesign No.7 (1995) 4-140.
- [4] PCI SIG, PCI Local Bus Specification Revision 2.2, PCI SIG (1998) 1-322.

Implementation of PCI bus interface circuit by FPGA

Masaaki Ono

Academic Service Office for Systems and Information Engineering, University of Tsukuba,
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan

A PCI (Peripheral Component Interconnect) bus interface circuit was developed for a network interface of a dedicated cluster network called Maestro. The PCI bus interface circuit is implemented on FPGA (Field Programmable Gate Array) and can handle PCI target transfer as well as PCI master transfer. Although the PCI target transfer of the PCI bus interface circuit is limited for single transfer, the PCI master transfer supports burst transfer.

The operating frequency of the PCI bus is 66 MHz, the data widths are 64 bits, and the maximum theoretical throughput is 528 Mbytes/sec. The PCI bus interface circuit was designed and implemented without using commercial IP (Intellectual Property). Currently, the PCI bus interface circuit is implemented on FPGA of the Maestro2 and Maestro3 network interface boards to connect the NI boards with the host personal computers.

Keywords: FPGA; PCI bus; Maestro