



Design Gateway Co.,Ltd

本社: 〒184-0012
 東京都小金井市中町 3-23-17
 電話/FAX: 050-3588-7915
 E-mail: sales@dgway.com
 URL: www.dgway.com

特長

- CPU なしで NVMe 規格の PCIe SSD をアクセスするためのアプリケーション・レイヤを実装
- 接続容易な制御ユーザ I/F とデータ FIFO I/F
- Xilinx 製 AXI PCIe ブリッジと 128 ビット・バスで直結
- 内蔵 BRAM で構築する 2 種類のデータ・バッファ容量モードを用意、外付け DDR メモリ不要
- IDENTIFY, WRITE, READ の 3 コマンドをサポート
- 以下の NVMe デバイスをサポート
 - ベース・クラス・コード:01h (マス・ストレージ), サブ・クラス・コード:08h (不揮発性メモリ), プログラミング・インターフェイス:02h (NVMHCI)
 - MPSMIN (最少メモリ・ページ・サイズ): 0 (4Kbyte)
 - MDTs (最大データ転送サイズ): 0 (制限なし) または、少なくとも 5 (128K バイト)
- AB16-PCIeXOVR アダプタと Xilinx 評価ボードで実機動作するリファレンス・デザイン添付
- 安心の日本語サポート

Core Facts	
コアの提供情報	
提供ドキュメント	リファレンスデザインマニュアル 実機デモ手順書
提供形態	暗号化したネットリスト
制約ファイル	リファレンスデザインで Constrain file を提供
検証方法	リファレンス・デザインの実機検証
参照デザイン言語	VHDL で記述
リファレンス・デザイン	Vivado プロジェクトによる 実機動作デザイン
対応検証ボード	KCU105, VC707, VC709, ZC706, KC705, Zynq Mini-ITX
技術サポート	
デザインゲートウェイ・ジャパンによる国内サポート	

表 1: 7 シリーズ (PCIe GEN2/PCIe GEN3) コンパイル結果

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices ¹	BRAMTile ²	Design Tools
Virtex-7	XC7VX690TFFG1761-2	250	1039	1140	571	57/114	Vivado2015.4
Virtex-7	XC7VX485TFFG1761-2	125	1039	1138	576	57/114	Vivado2015.4
Zynq-7000	XC7Z045FFG900-2	125	1039	1140	561	57/114	Vivado2015.4
Kintex-7	XC7K325TFFG900-2	125	1039	1138	574	57/114	Vivado2015.4

表 2: UltraScale シリーズ (PCIe GEN3) コンパイル結果

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	BRAMTile ²	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	250	1039	1157	343	57/114	Vivado2015.4

注:

- 1) 本コンパイル結果は Xilinx 製 AXI PCIe ブリッジコアやメモリコントローラ等を含まないコア単体でのリソース情報です
- 2) データ・バッファ 256KByte の Mode1 (メモリ節約モード) の場合 57BRAMTile、データ・バッファ 512KByte、の Mode2 (パフォーマンス・モード) の場合 114BRAMTile の BRAM が必要となります
- 3) 実際のスライス消費カウントはユーザロジックやフィット条件等に依存します

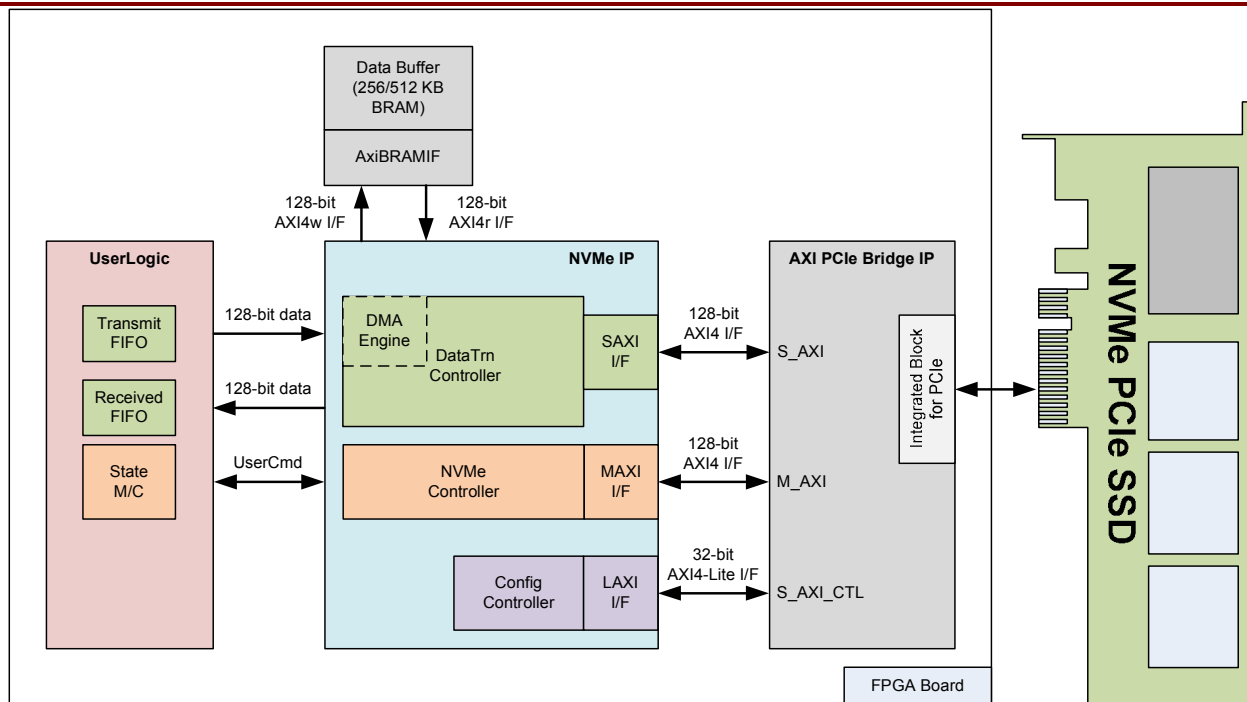


図 1: NVMe-IP コアのブロック図

アプリケーション情報

NVMe-IP コアは Xilinx 製 AXI PCIe ブリッジ IP コアと組み合わせて動作し、MicroBlaze 等の CPU の介在なしで NVMe 規格 PCIe SSD へのリード/ライト・アクセスを可能とするソリューションを提供します。FPGA 内蔵 BRAM メモリをデータ・バッファとして使い、ユーザ回路と NVMe SSD 間のデータを転送します。大容量かつ高速でのデータ書き込み/読み出しアクセスが必要な組み込みストレージの最適解です。特に M.2 フォームファクタの最新 NVMe SSD と組み合わせることで、超小型のストレージ製品が短期間で開発可能となります。

概略

NVMe-IP コアは NVM Express 規格に準拠した PCIe SSD をアクセスするためのホスト・コントローラ機能を提供します。NVMe SSD の物理インターフェイスは PCI Express のため、Xilinx 標準の AXI PCIe Bridge IP コアを使うことで PCIe プロトコルを適用します。NVMe-IP コアは PCIe バスにアクセスし Identify, ライト, リードの3種類の NVMe コマンドをサポートするため NVMe 制御レジスタのロジックを内蔵します。一般的に NVMe SSD は複数のコマンドを並列して実行するため、SSD からのデータ要求はシーケンシャルな並びにはなりません。このため SSD からのランダムなデータ要求に対応するため BRAM で構築した内蔵メモリによるデータ・バッファを必要とします。NVMe-IP コア製品には 2 種類のデータ・バッファ容量に対応したコアが同梱されます。1 つは MODE1 のエコノミック(メモリ節約)モードでデータ・バッファとして 256K バイトの BRAM メモリを使います。もうひとつは MODE2 のパフォーマンス・モードでデータ・バッファの容量は 512K バイトです。ユーザ・インターフェイスはコマンド・開始アドレス・転送長のパラメータをセットするだけの使いやすいデザイン(弊社 dglF 規格)であり、データ・インターフェイスは一般的な FIFO による接続となります。IP コア内部には一切の非同期回路部分を排除しているため、IP コアのクロック・ドメインは AXI PCIe ブリッジ IP からの出力クロックをそのまま使う必要があります。IP コアがパケット転送時に異常を検出した場合、エラー情報とあわせてエラー信号をアサートします。

リファレンス・デザインは Xilinx 評価ボードで動作し、IP コアの購入前に Web サイトから評価用 bit ファイルをダウンロードできます。

データ・バッファとして FPGA 内部 BRAM ではなく外部 DDR メモリを使う NVMe-IP コアもオプション製品として提供可能です。

コア機能の説明

図 2 にリセット解除後の NVMe-IP コアの動作シーケンスを示します。

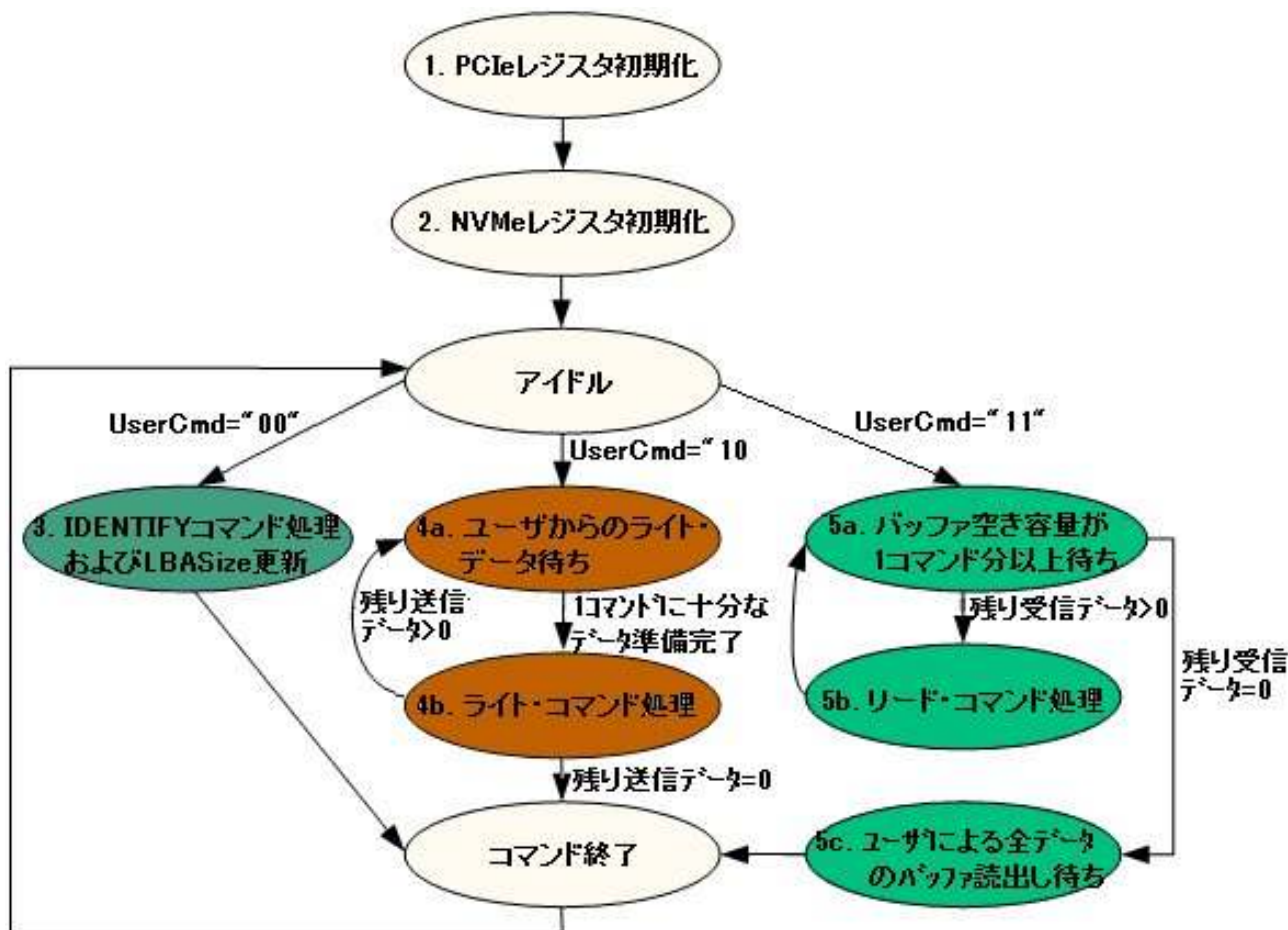


図 2: NVMe-IP コアの動作フロー

- 1) IP コアは AXI PCI ブリッジ・コアのレジスタと PCIe コンフィグレーション空間をセットし PCIe 環境を NVMe で動作するよう初期化します。
- 2) IP コアは SSD 内 NVMe コントローラのパラメータや動作環境をセットします。この一連の初期化が全て完了すると IP コアはアイドル状態となり、ユーザ回路からの指示を待ちます。
- 3) ユーザ回路からの最初の指示は Identify コマンドとする必要があり、このコマンドによって LBASize が更新され有効な SSD 容量情報がユーザ回路から認識できるようになります。
- 4) ライト・コマンドの場合、IP コアは 1 コマンドでの転送サイズに十分な量のライト・データがユーザ回路からバッファに書き込まれるのを待ちます。(NVMe-IP コアの 1 コマンドにおける最大転送サイズは 128K バイトです) その後 IP コアは NVMe SSD に対してライト・コマンドを発行します。そして IP コアは SSD から全てのライト・コマンド処理が完了したステータスを確認した後にアイドル状態に復帰します。
- 5) リードコマンドの場合、IP コアはデータ・バッファの空き容量が 1 コマンドでの転送サイズに対して十分であることを確認し、NVMe SSD に対してリード命令を発行します。その後 IP コアは全てのコマンド処理が完了しユーザ回路がバッファ内の全リード・データを読み出したことを確認した後にアイドル状態に復帰します。

上記のシーケンスにおいて、NVMe-IP コアは AXI PCIe ブリッジ・コアのインターフェイスからコンフィグレーション、NVMe、データ・インターフェイスの 3 グループに分類して実装されます。各グループでの詳細について以下に説明します。

コンフィグレーション (Configuration)

システムの電源投入後、PCIe ルート・コンプレックス・システムは PCIe 規格に従い PCIe デバイスに対してコンフィグレーション・データをリード・ライトする必要があります。デバイスに対するコンフィグレーション・データの書き込みは本ブロック内で実行します。また、PCIe 割込み信号やステータスも読み出されシステムが通常状態でない場合にユーザ回路からモニタできるよう出力されます。

- **コンフィグレーション・コントローラ (Config Controller)**

このモジュールは2つの動作シーケンスを内蔵します。1つは PCIe クラスをチェックする初期化シーケンスであり、BAR アドレスのセット、MSI 割込み許可、マスタ・モードの設定を行います。もう一つは PCIe エラーと割込みステータスをモニタします。どちらのシーケンスともレジスタ・アクセスを介して制御されます。

- **LAXI I/F**

コンフィグレーション・レジスタは 32 ビット AXI4-Lite 規格で制御されます。本 IP は AXI PCIe ブリッジ IP のコンフィグレーション・データや内部レジスタおよび PCIe SSD のコンフィグレーション・データへのリード・ライトをマスタ・モードで実行するようデザインされています。

NVMe

この回路ブロックは IP コアのメイン・コントローラです。PCIe 初期化プロセスが完了すると、本ブロックは SSD の NVMe レジスタ初期化を開始します。ライト・リード動作を制御するため、本ブロックでライト・リード・コマンド・パケットが生成され、デバイスに対して新しいコマンド要求を送信するためドアベル・レジスタをセットします。さらに、完了パケットをモニタしデバイスからの完了パケットをフラッシュするためドアベル・レジスタをセットします。コマンド/ステータス/データを格納するメモリ・アドレスは本回路ブロックによりコマンド・パケットとレジスタ用に設定されます。

- **NVMe コントローラ**

NVMe レジスタはシステム起動後に一度だけ本モジュールが初期化します。ユーザからのコマンドを処理するため、本モジュールはユーザ入力をデコードしコマンド ID、オペコード、データ・ポインタなどのコマンド FIS パラメータを準備します。総転送長が 128K バイト以上の場合、コントローラは SSD に対して複数のコマンドを生成します。新たなコマンド要求を開始するためドアベル・レジスタをセットする前にデータ・バッファの状態をモニタし、128K バイト・ライト・コマンドの場合有効なデータが少なくとも 128K バイト準備できているか、128K バイト・リード・コマンドの場合空き容量が 128K バイト以上であることを確認します。デバイスからの完了パケットはエラーが発生していないかどうかモニタされます。ライト・コマンドの場合 SSD から戻される最後のステータス・パケットを受信するか、リード・コマンドの場合データ・バッファの最終データがユーザ回路から読み出された時点でビジー・ステータスはクリアされます。

- **MAXI I/F**

NVMe レジスタは PCIe デバイスの BAR0/BAR1 にマッピングされ、128 ビット AXI4 バスを介してアクセスできます。ただし NVMe レジスタへのアクセスは AXI PCIe ブリッジ IP の S_AXI ポートを通して 32 ビットのシングル・アクセスのみが使われます。LAXI I/F と同様、このモジュールもマスタ・モードで動作します。

Data

この回路ブロックは AXI PCIe ブリッジ IP からのアドレス要求をデコードし各要求のデータ転送元や転送先を選択します。ユーザ回路内送信 FIFO からのライト・データは AXI PCIe ブリッジ IP へ直接転送はされず BRAM で構築したデータ・バッファを介して転送されます。また AXI PCIe ブリッジ IP からのリード・データも同様に一旦 BRAM でのデータ・バッファを介してユーザ回路の受信 FIFO へと転送されます。データ・バッファのデータ転送は 128 ビット AXI バスを通し DMA エンジンを使って行われます。

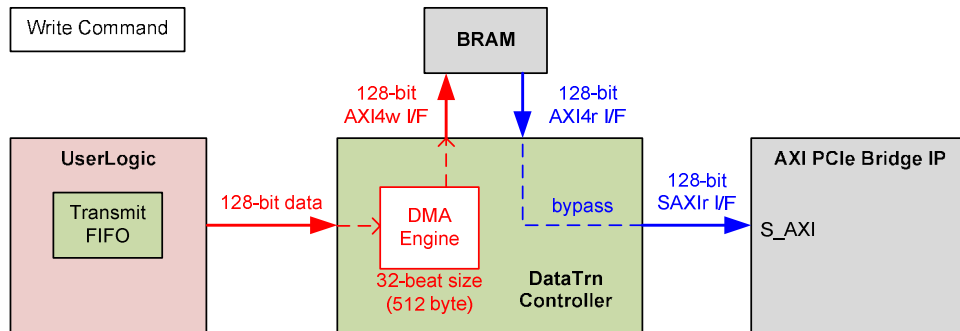


図 3: ライト・コマンドでのデータの流れ

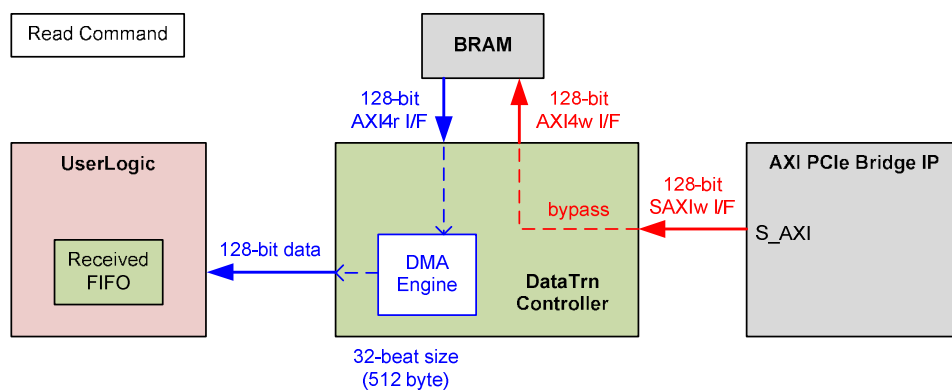


図 4: リード・コマンドでのデータの流れ

- **データ転送コントローラ (DataTrn Controller)**

図 3 および図 4 に示すようにデータ・バッファの AXI4 バスは 2 つのマスターと接続しますが、ひとつは内部 DMA エンジンで、もう一つは AXI PCIe ブリッジ・コアの S_AXI ポートでデータ転送方向別に接続します。DMA エンジンは常にバースト・サイズを 32 ビートまたは 512 バイトでセットします。制御/ステータス・データにおいては転送要求のアドレスからデータ種別がデコードされパケットが処理されます。

S_AXI ライト要求でデータ・タイプは Identify データ、Admin 完了データ、IO 完了データの何れかとなります。各完了データ内のステータス値は本コントローラでモニタされ、各転送でエラーが発生していないことを確認します。IO 完了データ内のコマンド ID 値はどのコマンドが完了したのかを確認するためチェックされます。

S_AXI リード要求でデータ・タイプは PRP リスト、Admin サブミッション、IO サブミッションの何れかとなり、NVMe コントローラから送信されます。

- **SAXI I/F**

これは 128 ビット AXI4 バスのスレーブ側です。データ・エリアにおいて S_AXI インターフェイスとデータ・バッファのインターフェイスはバイパスしております。非データ・エリアにおいて AXI4 インターフェイスは処理を簡単にするため RAM インターフェイスに変換しております。

ユーザ回路

ユーザ回路はコマンド、アドレス、転送長を送信する非常に簡単な回路で実装できます。データは FIFO インターフェイスで転送できます。本システムでは NVMe-IP コアと接続する大容量バッファ(256M バイト)を接続するため、ユーザ・ロジック内の FIFO は少容量のもので構いません。

AXI PCIe ブリッジ IP および PCIe 統合ブロック

NVMe-IP コアと PCIe バスは、Xilinx の AXI PCIe ブリッジ IP および PCIe 統合ブロックを介して接続します。このため一つの FPGA で接続できる NVMe SSD の最大数はその FPGA が内蔵する PCIe 統合ブロックの数に制限されますのでご注意ください。AXI PCIe ブリッジ IP は NVMe-IP コアとは AXI4 バスで接続し、PCIe 統合ブロックと直結します。AXI PCIe ブリッジ IP のより詳細については PCIe GEN2 については“PG055 LogiCore IP AXI Bridge for PCI Express”を、PCIe GEN3 については“PG194 AXI Bridge for PCI Express Gen3”のドキュメントを参照してください。

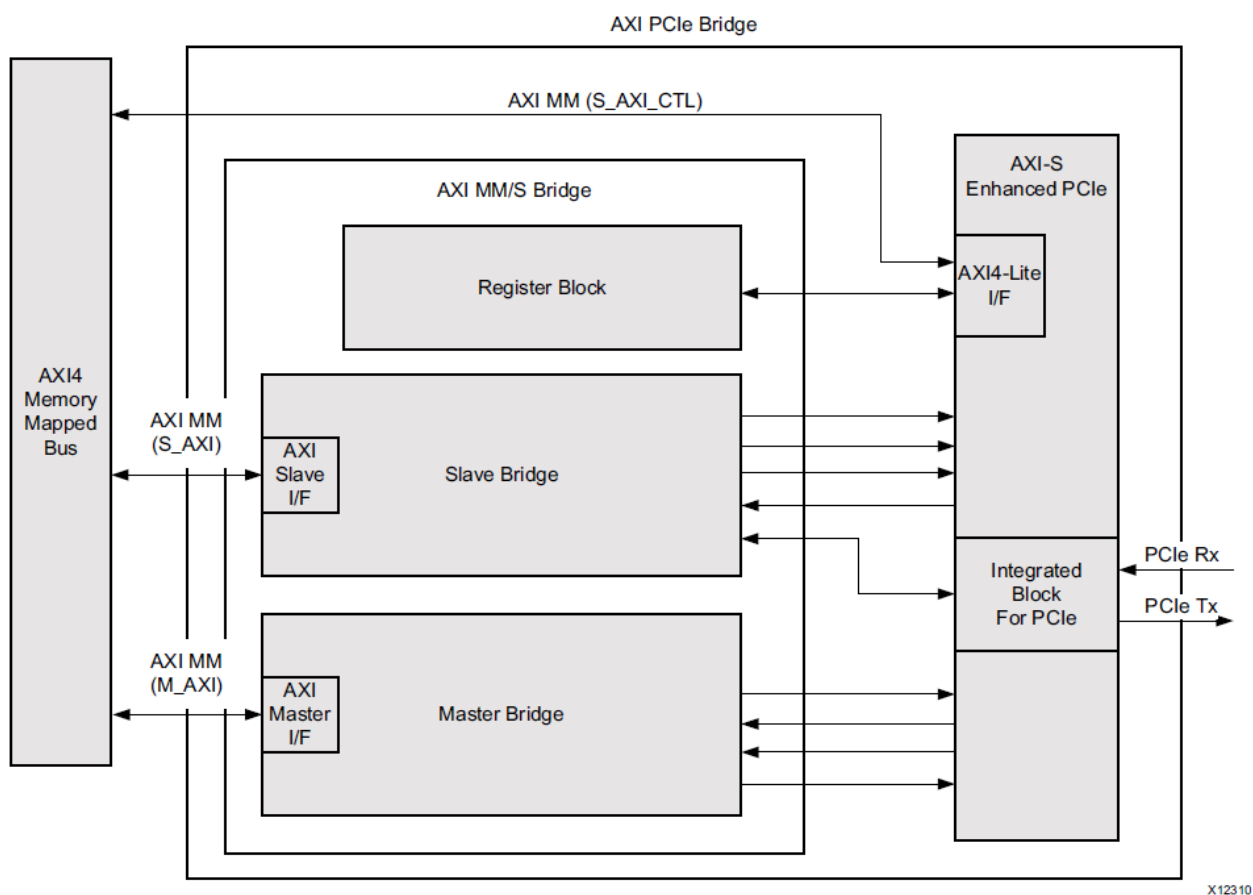


図 5: AXI PCIe ブリッジの接続

コアの I/O 信号

コアの I/O 信号について下表 3 に説明します。

表 3: コア I/O 信号

信号名	方向	説明
システム信号		
RstB	In	アクティブ Low のリセット信号、Clk 信号が安定になったら解除すること
Clk	In	AXI4-Lite および AXI4 バス・インターフェイスと同期する AXI PCIe ブリッジ IP からのクロック信号、PCIe Gen2 の場合 125MHz で Gen3 の場合 250MHz
ユーザ・インターフェイス (dgIF TypeS)		
UserCmd[1:0]	In	ユーザ・コマンド種別、'00': IDENTIFY、'10':ライト、'11':リード
UserAddr[47:0]	In	ライト/リード先の開始セクタ・アドレス(1 セクタ=512 バイト)をセット
UserLen[47:0]	In	総転送セクタ・サイズ、1 以上かつ(LBASize-UserAddr)以下の値をセットする必要がある
UserReq	In	コマンド指示の要求、本 IP コアがアイドル状態(UserBusy='0')でのみ発行可能。 UserCmd/UserAddr/UserLen に有効な値をセットし本信号を High アサートすることでコマンドを指示する
UserBusy	Out	IP コアのビジー・ステータス、この信号が'1'の場合次のコマンド要求を行うことができない。
LBASize[47:0]	Out	SSD の全容量をセクタ・サイズで表示、初期値は 0。ユーザ回路から IDENTIFY DEVICE コマンドを指定し SSD の容量データを取得した後で本信号に有効な値が出力される。
UserError	Out	エラー・フラグ、UserErrorType がゼロでなく何らかのエラーが発生した場合にアサートされる、本信号は RstB をアサートするとゼロ・クリアされる。
UserErrorType[31:0]	Out	エラー・ステータス情報、各ビットの意味は以下の通り [0] - PCIe クラス・コードが正しくない [1] - 接続 SSD の CAP(Controller Capabilities)レジスタからのエラーが以下の原因で発生 * MPSMIN (メモリ・ページ・サイズ最小値)がゼロでない * NVM コマンド・セット・フラグ(CAP レジスタの bit37)が'1'ではない * DSTRD (ドアベル・ストライド)が'0'でない [2] - Admin 完了エントリがタイムアウト時間内に返送されなかった [3] - Admin 完了エントリのステータス・レジスタが'0'でないか、フェーズ・タグ/コマンド ID が無効である。より詳細は AdmCompStatus 信号を参照すること [4] - IO 完了エントリがタイムアウト時間内に返送されなかった [5] - IO 完了エントリのステータス・レジスタが'0'でないか、フェーズ・タグ無効である。 より詳細は IOCompStatus 信号を参照すること [31:6] - 未使用 注意: bit[2]/[4]は TimeOutSet で設定したタイムアウト経過によりセットされる
UserFifoWrCnt[15:0]	In	受信(リード・データ)用 FIFO のライト・データ・カウンタ値、コアが FIFO のフル状態をチェックするために使う。FIFO サイズが 16 ビット以下の場合上位ビットには'1'をセットする必要がある
UserFifoWrEn	Out	SSD からリードしたデータと同期してイネーブルすることでリード用 FIFO へ書き込む
UserFifoWrData[127:0]	Out	受信用 FIFO に出力する 128 ビット・リード・データ、UserFifoWrEn と同期して出力
UserFifoRdCnt[15:0]	In	送信(ライト・データ)FIFO のリード・データ・カウンタ値、コアが FIFO 内の有効データ数をチェックするために使う。FIFO サイズが 16 ビット以下の場合上位ビットには'0'をセットする必要がある
UserFifoEmpty	In	送信 FIFO のエンプティ・フラグ、FIFO の有効データ数をチェックするために使う。
UserFifoRdEn	Out	送信 FIFO からコアへのデータ出力要求
UserFifoRdData[127:0]	In	送信 FIFO からコアへ転送される 128 ビット・ライト・データ、UserFifoRdEn の次クロックに有効な値をコアへ出力する必要がある

信号名	方向	説明
NVMe IP インターフェイス		
TestPin[31:0]	Out	本コアのテスト出力端子(通常は非使用)
TimeOutSet[31:0]	In	接続 SSD のコマンド完了待ちタイムアウト設定値 設定時間単位は Clk クロック周期単位 (Gen2 の場合 8ns で Gen3 の場合 4ns)
LinkSpeed[1:0]	Out	PCIe リンク速度 "00": 未リンク状態, "01": Gen1 (2.5 Gbps), "10": Gen2 (5.0 Gbps), "11": Gen3 (8.0 Gbps).
PCleInt	In	AXI PCIe ブリッジ IP からの割り込み出力.
PCleIntStatus[31:0]	Out	AXI PCIe ブリッジ IP 内の割り込みデコード・レジスタ(0x138)の最新の読み出し値を示す。この値はエラー発生時に PCleInt がアサートされた時に更新される。より詳細については Xilinx 社発行ドキュメントの "PG055 AXI Memory Mapped to PCI Express Gen2"または"PG194 AXI Bridge for PCI Express Gen3 Subsystem"を参照のこと
AdmCompStatus[15:0]	Out	[0] - Admin 完了エントリ内のフェーズ・タグまたはコマンド ID が無効である [15:1] - Admin 完了エントリのステータス・フィールド値
IOCompStatus[15:0]	Out	[0] - IO 完了エントリ内のフェーズ・タグが無効である [15:1] - IO 完了エントリのステータス・フィールド値
NVMeCAPReg[31:0]	Out	SSD からの NVMe キャパビリティ・レジスタ出力の抜粋情報 [15:0] - MQES (Maximum Queue Entries Supported) [19:16] - DSTRD (Doorbell Stride) [20] - NVM command set flag [24:21] - MPSPMIN (Memory Page Size Minimum) [31:25] - 未使用
IdenCtrlWrEn	Out	IdenCtrlWrData および IdenCtrlWrAddr と同期して出力するライト・イネーブル信号
IdenCtrlWrAddr[7:0]	Out	IdenCtrlWrData のインデックスを 128 ビット単位で示す、IdenCtrlWrEn に同期
IdCtrlWrData[127:0]	Out	IDENTIFY コマンドの 4K バイトの Identify Controller データ、IdenCtrlWrWrEn に同期.
IdenNameWrEn	Out	IdenNameWrData および IdenNameWrAddr と同期して出力するライト・イネーブル信号
IdenNameWrAddr[7:0]	Out	IdenNameWrData のインデックス、IdenNameWrEn に同期
IdenNameWrData[127:0]	Out	IDENTIFY コマンドの 4K バイトの Identify Namespece データ、IdenNameWrWrEn に同期.
マスタ AXI4-Lite インターフェイス		
LXiwAwAddr[31:0]	Out	ライト・アドレス、ライト・バースト・トランザクションで最初の転送のアドレスを出力
LXiwAwValid	Out	ライト・アドレス有効、ライト・アドレスと制御信号が有効であることを示す
LXiwAwReady	In	ライト・アドレス・レディ、スレーブ側はライト・アドレスと制御信号の受信が可能な状態を示す
LXiwData[31:0]	Out	ライト・データ
LXiwStrb[3:0]	Out	ライト・ストロブ、有効データが格納されたバイト・レーンを示す
LXiwValid	Out	ライト有効、有効なライト・データとストロブが出力されていることを示す
LXiwReady	In	ライト・レディ、スレーブ側はライト・データを受信可能なことを示す
LXiwBValid	In	ライト応答有効、ライトの有効応答を示す
LXiwBReady	Out	応答レディ、マスターはライト応答を受信可能なことを示す
LXiwArAddr[31:0]	Out	リード・アドレス、リード・バースト・トランザクションで最初の転送のアドレスを出力
LXiwArValid	Out	リード・アドレス有効、リード・アドレスと制御信号が有効であることを示す
LXiwArReady	In	リード・アドレス・レディ、スレーブ側はリード・アドレスと制御信号の受信が可能な状態を示す
LXiwArData[31:0]	In	リード・データ
LXiwArValid	In	リード有効、有効なリード・データが LXiwArData バス上に現れていることを示す
LXiwArReady	Out	リード・レディ、マスター側はリードデータと応答を受信できることを示す

信号名	方向	説明
マスタ AXI4 バス・インターフェイス		
MAxiAwId[3:0]	Out	ライト・アドレス ID、常に'0000b'が出力される
MAxiAwAddr[31:0]	Out	ライト・アドレス、ライト・バースト・トランザクションで最初の転送のアドレスを出力
MAxiAwRegion[3:0]	Out	リージョン・アイデンチファイア、常に'0000b'が出力される
MAxiAwLen[7:0]	Out	バースト長、1バーストでの転送長を示す。シングル・アクセスの場合常に'00'hとなる
MAxiAwSize[2:0]	Out	バースト・サイズ、各バーストの転送におけるサイズを示す、32ビット・アクセスでは常に'010'bが出力される
MAxiAwBurst[1:0]	Out	バースト・タイプ、バースト内の各転送でのアドレス計算方法を決定付ける。常に'01'b すなわちインクリメンタルが指定される。
MAxiAwValid	Out	ライト・アドレス有効、ライト・アドレスと制御信号が有効であることを示す
MAxiAwReady	In	ライト・アドレス・レディ、スレーブ側はライト・アドレスと制御信号を受信できることを示す
MAxiwData[127:0]	Out	ライト・データ
MAxiwStrb[15:0]	Out	ライト・ストロブ、有効なデータを持つバイト・レーン情報を示す
MAxiwLast	Out	最終ライト、ライト・バーストでの最終転送を示す
MAxiwValid	Out	ライト有効、有効なライト・データとストロブが出力されていることを示す
MAxiwReady	In	ライト・レディ、スレーブ側はライト・データを受信できることを示す
MAxiBValid	In	ライト応答有効、ライト応答が有効であることを示す
MAxiBReady	Out	応答レディ、マスターはライト応答を受信できることを示す
MAxiArId[3:0]	Out	リード・アドレス ID、常に'0000'b が出力される
MAxiArAddr[31:0]	Out	リード・アドレス、リード・バースト・トランザクションで最初の転送のアドレスを出力
MAxiArRegion[3:0]	Out	リージョン・アイデンチファイア、常に'0000b'が出力される
MAxiArLen[7:0]	Out	バースト長、1バーストでの転送長を示す。シングル・アクセスの場合常に'00'hとなる
MAxiArSize[2:0]	Out	バースト・サイズ、各バーストの転送におけるサイズを示す、32ビット・アクセスでは常に'010'bが出力される
MAxiArBurst[1:0]	Out	バースト・タイプ、バースト内の各転送でのアドレス計算方法を決定付ける。常に'01'b すなわちインクリメンタルが指定される。
MAxiArValid	Out	リード・アドレス有効、リード・アドレスと制御信号が有効であることを示す
MAxiArReady	In	リード・アドレス・レディ、スレーブ側はリード・アドレスと制御信号を受信できることを示す
MAxirData[127:0]	In	リード・データ
MAxirValid	In	リード有効、有効なリード・データが受信がバス上に用意されていることを示す
MAxirReady	Out	リード・レディ、マスター側はリード・データとリード応答を受信できることを示す

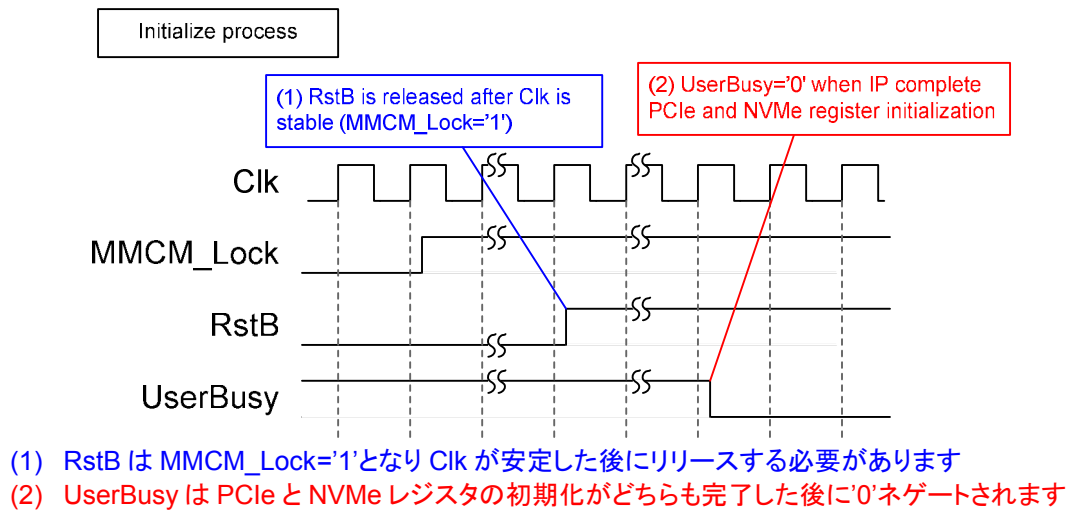
信号名	方向	説明
スレーブ AXI4 バス・インターフェイス		
SXAwAddr[31:0]	In	ライト・アドレス、ライト・バースト・トランザクションで最初の転送のアドレスが入力される
SXAwLen[7:0]	In	バースト長、1バーストでの転送長が入力される。
SXAwValid	In	ライト・アドレス有効、入力されるライト・アドレスと制御信号が有効であることを示す
SXAwReady	Out	ライト・アドレス・レディ、スレーブ側はライト・アドレスと制御信号を受信できることを示す
SXiwData[127:0]	In	ライト・データ
SXiwStrb[15:0]	In	ライト・ストロブ、有効なデータを持つバイト・レーン情報を示す
SXiwLast	In	最終ライト、ライト・バーストでの最終転送を示す
SXiwValid	In	ライト有効、有効なライト・データとストロブが入力されていることを示す
SXiwReady	Out	ライト・レディ、スレーブ側はライト・データを受信できることを示す
SXIBResp[1:0]	Out	ライト応答、ライト・トランザクションのステータスを示す、常に'00'b(OK)を出力する
SXIBValid	Out	ライト応答有効、ライト応用が有効であることを示す
SXIBReady	In	応答レディ、マスターはライト応答を受信できることを示す
SXArAddr[31:0]	In	リード・アドレス、リード・バースト・トランザクションで最初の転送のアドレスが入力
SXArLen[7:0]	In	バースト長、1バーストでの転送長を示す。
SXArValid	In	リード・アドレス有効、リード・アドレスと制御信号が有効であることを示す
SXArReady	Out	リード・アドレス・レディ、スレーブ側はリード・アドレスと制御信号を受信できることを示す
SXirData[127:0]	Out	リード・データ
SXirResp[1:0]	Out	リード応答、リード・トランザクションのステータスを示す、常に'00'b(OK)を出力する
SXirLast	Out	最終リード、リード・バーストでの最終転送を示す
SXirValid	Out	リード有効、有効なリード・データが受信がバス上に用意されていることを示す
SXirReady	In	リード・レディ、マスター側はリード・データとリード応答を受信できることを示す

信号名	方向	説明
データ・バッファ接続用マスタ AXI4 インターフェイス		
BufAxiAwAddr[31:0]	Out	ライト・アドレス、ライト・バースト・トランザクションの先頭の転送アドレスを示す
BufAxiAwLen[7:0]	Out	バースト長、1バーストでの転送長を示す、ライト・コマンドでは 32 ビットでリード・コマンドでは SAXIAwLen と同じ値となる。
BufAxiAwSize[2:0]	Out	バースト・サイズ、各バーストの転送におけるサイズを示す、128 ビット・アクセスのため常に 100b である。
BufAxiAwValid	Out	ライト・アドレス有効、ライト・アドレスと制御信号が有効であることを示す。
BufAxiAwReady	In	ライト・アドレス・レディ、スレーブ側はライト・アドレスと制御信号を受信できることを示す
BufAxiwData[127:0]	Out	128 ビット・ライト・データ
BufAxiwStrb[15:0]	Out	ライト・ストロブ、どのバイト・レーンに有効なデータが出力されているのかを示す。ライト・コマンドでは FFFFh でリード・コマンドでは SAXiwStrb と同じ値となる。
BufAxiwLast	Out	最終ライト、ライト・バーストでの最終転送を示す
BufAxiwValid	Out	ライト・アドレス有効、ライト・アドレスとストロブが有効であることを示す
BufAxiwReady	In	ライト・レディ、スレーブ側はライト・データを受信できることを示す
BufAxiBValid	In	ライト応答有効、ライト応答が有効であることを示す
BufAxiBReady	Out	応答レディ、マスターはライト応答を受信できることを示す
BufAxiArAddr[31:0]	Out	リード・アドレス、リード・バースト・トランザクションで最初の転送のアドレスを出力
BufAxiArLen[7:0]	Out	バースト長、1バーストでの転送長を示す。リード・コマンドでは 32 ビットでライト・コマンドでは SAXIArLen と同じ値となる。
BufAxiArSize[2:0]	Out	バースト・サイズ、各バーストの転送におけるサイズを示す、128 ビット・アクセスのため常に 100'b が出力される
BufAxiArValid	Out	リード・アドレス有効、リード・アドレスと制御信号が有効であることを示す
BufAxiArReady	In	リード・アドレス・レディ、スレーブ側はリード・アドレスと制御信号を受信できることを示す
BufAxiArData[127:0]	In	128 ビット・リード・データ
BufAxiArValid	In	リード有効、有効なリード・データが受信がバス上に用意されていることを示す
BufAxiArReady	Out	リード・レディ、マスター側はリード・データとリード応答を受信できることを示す

タイミング波形

初期化

IP コアの同期クロック信号(Clk 入力信号)は AXI PCIe ブリッジ IP 内の MMCM から作られます、このため下図 6 に示すように、AXI PCIe ブリッジ IP からの MMCM_Lock 出力信号をモニタし、クロックが安定したことを確認した後にコアのリセット信号となる RstB をリリースする必要があります。その後 IP コアは PCIe コンフィグレーション・レジスタと PCIeSSD 内の NVMe レジスタを初期化します。両方の初期化シーケンスが完了した後に UserBusy は'0'にネゲートされます。



- (1) RstB は MMCM_Lock='1'となり Clk が安定した後にリリースする必要があります
- (2) UserBusy は PCIe と NVMe レジスタの初期化がどちらも完了した後に'0'ネゲートされます

図 6: システム起動時の RstB と UserBusy のネゲート・タイミング

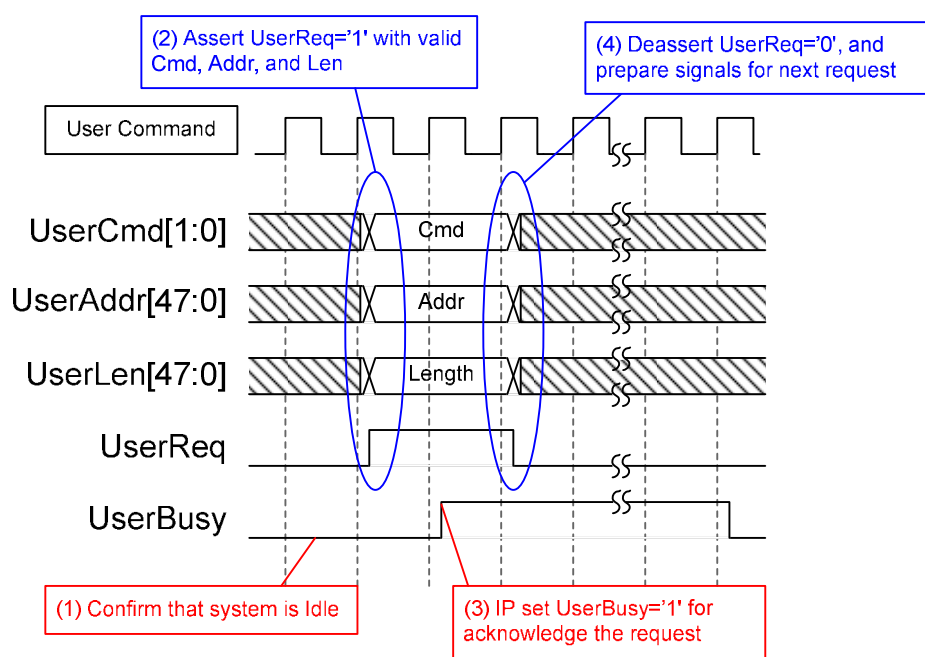
ユーザ・インターフェイス(dgIF typeS)

ユーザ・インターフェイスは2種類のインターフェイスに分類されます、すなわち一つはコマンド・インターフェイスでもう一つはデータ・インターフェイスです。これらユーザ・インターフェイスを非常にシンプルで使いやすい仕様で定義したものが dgIF で、dgIF typeS は DesignGateway 社のストレージ系 IP コア間で共通のユーザ・インターフェイスです。

図7にコマンド・インターフェイスのタイミング波形を示します。IP コアに新たなライト/リード命令を送る前に必ず UserBusy をモニタし IP コアがアイドル状態であることを確認する必要があります。UserCmd, UserAddr,そして UserLen は UserReq='1'でコマンドを出力する期間は図7に示すように必ず有効な値を保持しなくてはなりません。IP コアは UserBusy='1'としてコマンドを受信したことを示しコマンドの実行を開始します。UserBusy='1'となった以降は UserReq はクリアでき、ユーザ回路からは次の新たなコマンド用のパラメータを用意することができます。

(注意) UserAddr や UserLen は Identify コマンドでは使われないため無視されます

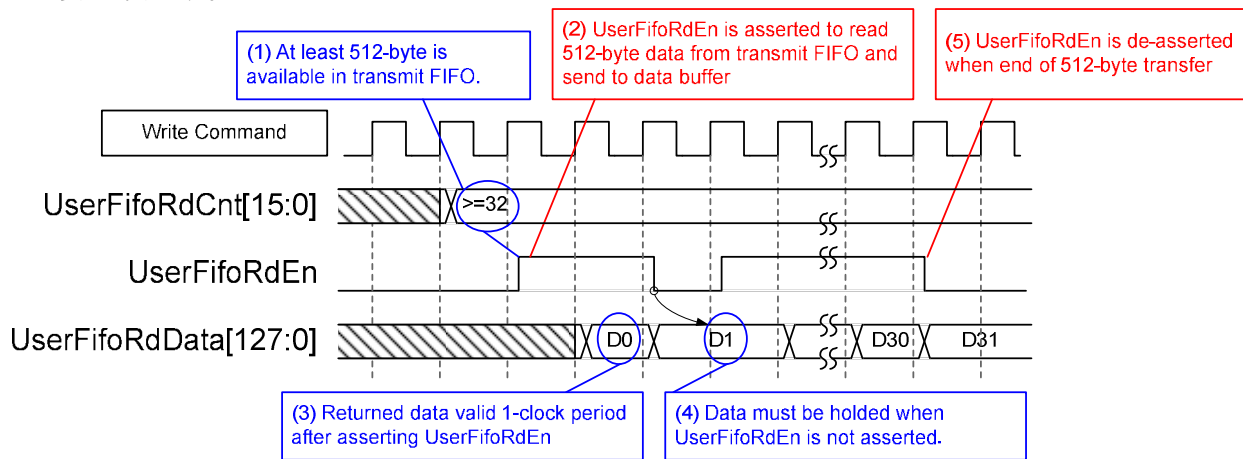
一方データ・インターフェイスにおいてはライト・コマンドにおけるデータ送信 FIFO はコアから読み出され、リード・コマンドにおけるデータ受信 FIFO はコアからライトされます。データ・インターフェイスのタイミング波形は図8および図9で示します。



- (1) システムがアイドル状態であることを確認します
- (2) 有効なコマンド、アドレス、転送長をセットし UserReq='1'をアサートします
- (3) IP コアは要求を受付けたことを UserBusy='1'で示します
- (4) ユーザ回路は UserReq='0'とし、次のコマンドのためのパラメータを準備できます

図7: ユーザ・コマンドのタイミング波形

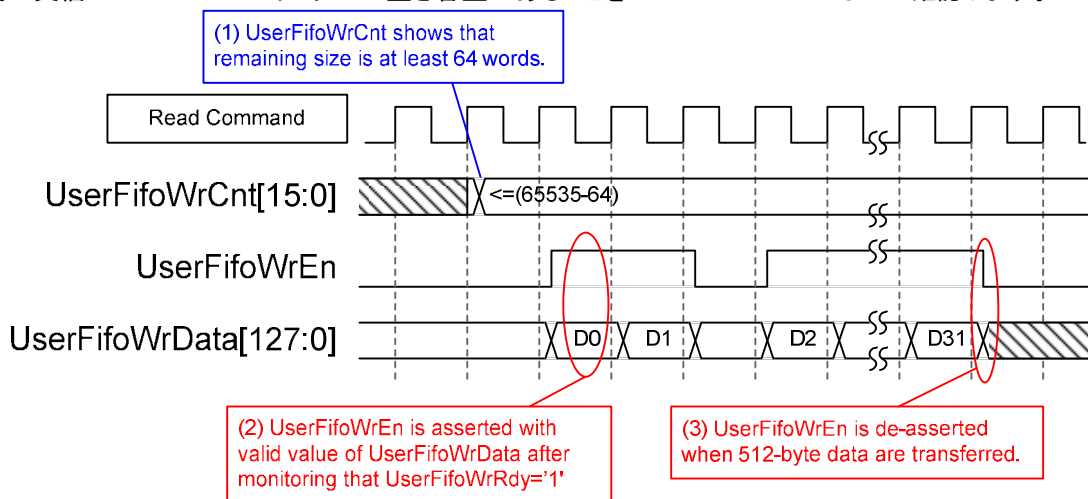
ライト・コマンドにおいてユーザ回路内の送信 FIFO からのライト・データは 128 ビット AXI4 バス・インターフェイスを介してデータ・バッファへと転送されます。データ・バッファへのバースト・サイズは 32 ビットまたは 512 バイト固定です。IP コアはバースト転送を開始する前に UserFifoRdCnt をモニタし、少なくとも 512 バイトのライト・データが送信 FIFO にあることを確認します。一般的な FIFO と同様、UserFifoRdData は図 8 に示すように UserFifoRdEn がアサートされた次のクロック・ピリオドで有効な値となり、またネゲートされた場合次にアサートされるまで有効なデータ値を保持する必要があります。



- (1) コアは送信 FIFO に 512 バイト以上のライト・データが用意されていることを確認します
- (2) UserFifoRdEn がアサートされ送信 FIFO からデータ・バッファに向け 512 バイトのデータが出力されます
- (3) 有効なライト・データは UserFifoRdEn='1' の次クロックで出力されます
- (4) UserFifoRdEn がアサートされない場合データは保持する必要があります
- (5) 512 バイトの転送後 UserFifoRdEn は'0'ネゲートします

図 8: ライト・コマンドにおける送信 FIFO からコアへのデータ転送タイミング波形

リード・コマンドにおいて UserFifoWrEn は UserFifoWrData 上の有効なリード・データと合わせてアサートされ、データ・バッファ内の受信データが受信 FIFO へと転送されます。ライト・コマンドと同様、512 バイト・データを FIFO へ転送する前に受信 FIFO に 512 バイト以上の空き容量があることを UserFifoWrCnt によって確認します。

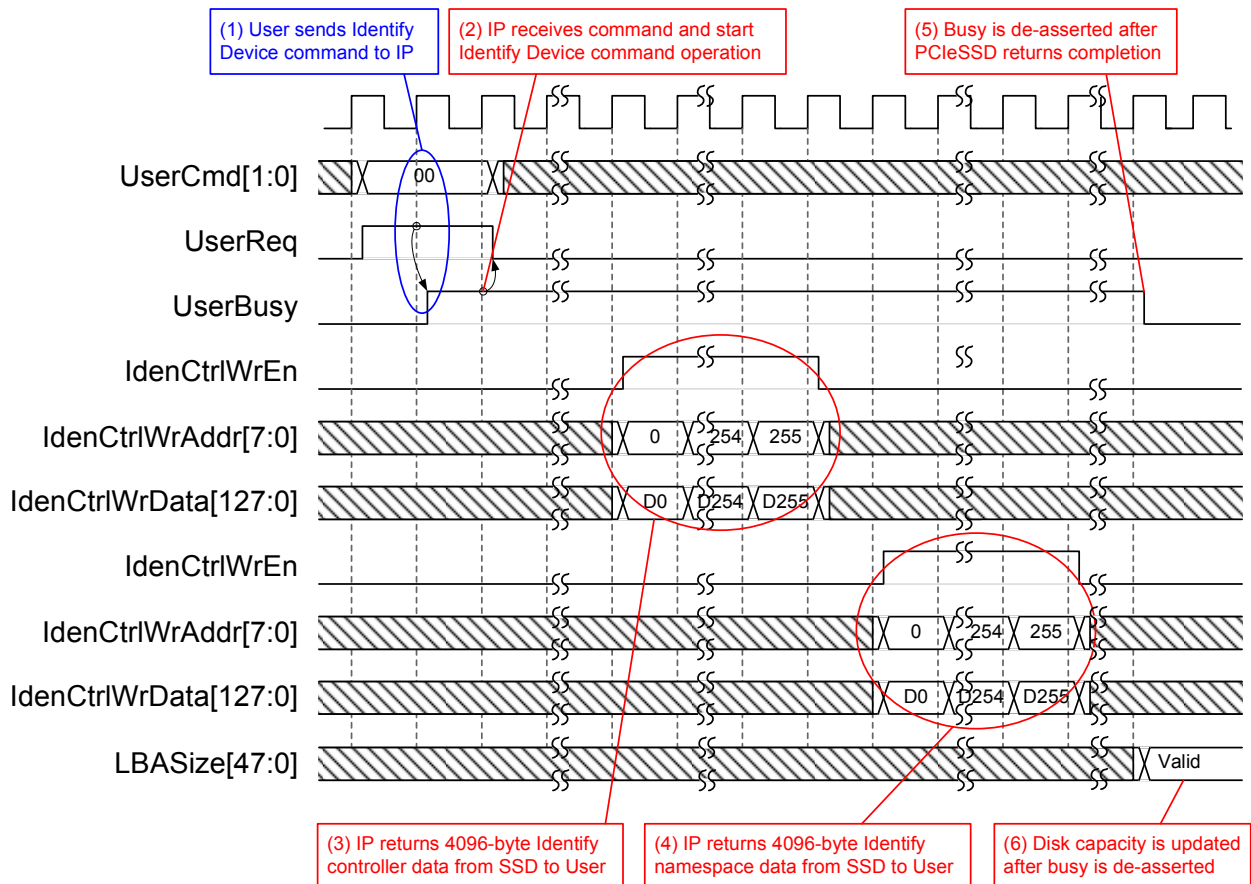


- (1) 受信 FIFO に 64 ワード(512 バイト)以上の空き領域があることを UserFifoWrCnt で確認します
- (2) 512 バイト以上の空き領域を確認した次クロックから UserFifoWrEn と UserFifoWrData が出力されます
- (3) UserFifoWrEn は 512 バイトを転送すると'0'ネゲートします

図 9: リード・コマンドにおけるコアから受信 FIFO へのデータ転送タイミング波形

IdenCtrl/IdenName

IP コアにライトあるいはリード・コマンドを送信する前に、ユーザ回路はまず IDENTIFY コマンドを発行し LBASize 出力を更新する必要があります。この LBASize 値はユーザ回路にてリード・ライト命令のアドレス+転送長の合計がそれぞれこの値を超えないよう制御するために使います。



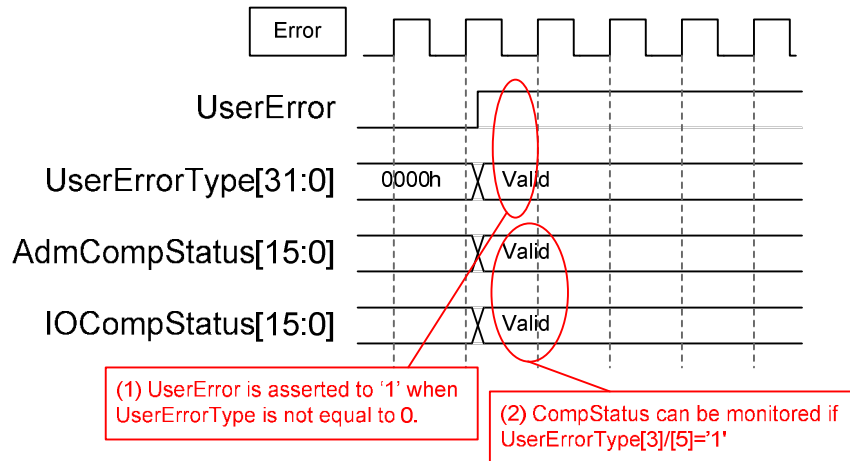
- (1) ユーザ回路は IDENTIFY コマンドの発行を IP コアに要求します
- (2) IP コアはコマンド要求を受け、IDENTIFY コマンド実行を開始します
- (3) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY コントローラ・データをユーザ回路に出力します
- (4) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY ネームスペース・データをユーザ回路に出力します
- (5) NVMeSSD が完了を返送すると UserBusy がネゲートされます
- (6) UserBusy がネゲートされるとディスク容量情報となる LBASize も更新されます

図 10: IDENTIFY コマンド後に LBASize が更新される

IDENTIFY コマンド発行においては、図 10 に示すように UserBusy がネゲートされていることを確認した上で UserCmd と UserReq をセットします。UserAddr と UserLen 入力については IDENTIFY コマンドでは必要ありません。コアによるコマンド実行後、4096 バイトの IDENTIFY コントローラ・データと 4096 バイトの IDENTIFY ネームスペース・データが出力されます。この双方の IDENTIFY データは連続しては出力されません。データは SSD の性質に依って複数のバースト転送に分割されることがあります。そして UserBusy がネゲートされると有効な LBASize データが出力されます。

エラー

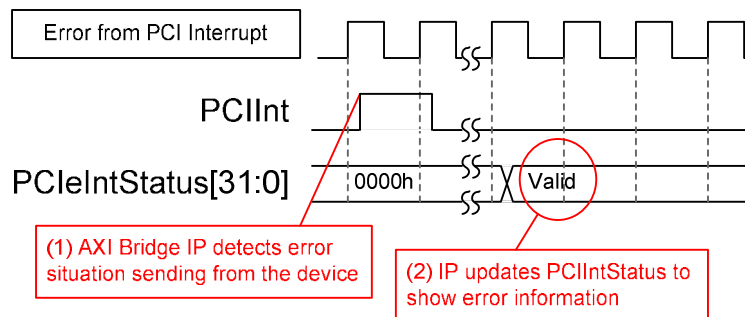
通常の動作時には UserError および UserErrorType 信号の全ビットはゼロです。UserError 信号は UserErrorType の各ビットを OR 条件して生成されます。UserErrorType のいずれかのビットが '1' にセットされた場合、図 11 に示すように UserError もアサートされ RstB が '0' アサートされるまで保持します。AdmCompStatus または IOCompStatus の値がエラー状態であった場合、UserErrorType のビット[3]/[5]がセットされます。このときユーザ回路側では AdmCompStatus または IOCompStatus を読み込むことでより詳細なエラー情報を確認することができます。



- (1) UserErrorType 値がゼロでない場合 UserError 信号は '1' アサートされます
- (2) UserErrorType[3]/[5]='1' の場合、CompStatus で詳細情報が確認できます

図 11: エラー・フラグのタイミング波形

これとは別に PCIe レベルでエラーが発生するケースがあります。NVMeSSD が PCIe レベルでエラーを検出した場合、ホスト(NVMe-IP コア)側に対して割り込みメッセージを送信します。この場合 AXI ブリッジ IP コアから PCIInt 信号がアサートされます。NVMe-IP コアは AXI ブリッジ IP コアから割り込みステータスをリードし PCIIntStatus 出力信号の値を更新します。ユーザ回路は PCIIntStatus 信号をモニタすることで PCIe レベルで何かエラーが発生したことを検出できます。通常の正常動作時は PCIeInt はアサートされず、PCIIntStatus 出力は図 12 に示すように全ビットが常に '0' となります。



- (1) AXI ブリッジ IP コアはデバイスからのエラー情報を検出します
- (2) NVMe-IP コアは PCIIntStatus を更新しエラー情報を出力します

図 11: PCIe のエラー・ステータスのタイミング波形

コアの検証方法

NVMe-IP コアは KCU105/VC709/VC707/ZC706/KC705/Zynq Mini-ITX 等の Xilinx 標準評価ボードと AB16-PCIeXOVR アダプタ(Zynq Mini-ITX の場合は不要)にて実機動作を検証できます。

推奨される設計スキルに関して

本 IP をユーザ回路上に迅速・確実に実装するために、Xilinx の Vivado ツールについての技術スキルを推奨します。

注文情報

本データシートに記載された NVMe-IP は以下の Xilinx 各ファミリーが対象となります。

製品型番	対象ファミリー	実機評価ボード	発売状況
NVMe-IP-KU	Kintex Ultrascale	KCU-105	発売中
NVMe-IP-KT7	Kintex-7	KC-705	発売中
NVMe-IP-AT7	Artix-7	AC-701	(開発中)
NVMe-IP-VT7	Virtex-7	VC-707	発売中
NVMe-IP-ZQ7	Zynq-7000	ZC-706	発売中

IP コアの価格やライセンス条件等についてはデザイン・ゲートウェイ (sales@dgway.com) または国内 Xilinx 各代理店までお問い合わせください。

また、コアご購入後のサポートは製品添付の実機動作リファレンス・デザインと同一の環境が前提となるため、対象ファミリーの Xilinx 評価ボードおよび AB16-PCIeXOVR アダプタの手配が必須となります。

履歴

リビジョン	日付	更新内容
1.0	Jun-2-2016	Initial Release
1.0J	2016/6/3	日本語版の初期版リリース
1.1J	2016/06/21	Kintex-Ultrascale を正式サポート
1.2J	2016/09/06	Zynq-7000 を正式サポート
1.3J	2016/09/09	Kintex-7 (KC705 の実機動作環境)を正式サポート
1.4J	2016/10/28	VC709 および Zynq Mini-ITX の実機デモ環境を追加サポート
1.5J	2016/12/17	データ・バッファを外付け DDR から内蔵 BRAM へ改良
1.51J	2017/02/21	表 3 の PCIeIntStatus 説明誤記を修正