

NVMe-IP コア・データシート

2016/12/21

Product Specification

Rev1.1J



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
- Altera 製 AvalonMM PICE ブリッジと 128 ビット・バスで直結
- 内蔵ブロック・メモリで構築する 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)
 - MDTSS (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 ArriaV GX スタータ・キットまたは Arria10SoC 開発キット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	QuartusII プロジェクトによる 実機動作デザイン
対応検証ボード	Arria10SoC Development Kit ArriaV GX Starter Kit
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: コンパイル結果

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers	Block Memory bit ¹	Design Tools
ArriaV GX	5AGXFB3H4F35C4	125	686	1192	2,907,152 / 4,194,304	QuartusII 15.1
Arria10 SX	10AS066N3F40E2SGE2	250	737	1444	2,907,152 / 4,194,304	QuartusII 16.0

注:

- 1) 本コンパイル結果は Altera 製 AvalonMM 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 2,907,152bit、データ・バッファ 512KByte の Mode2 (パフォーマンス・モード) の場合 4,194,304bit のブロック・メモリが必要となります
- 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

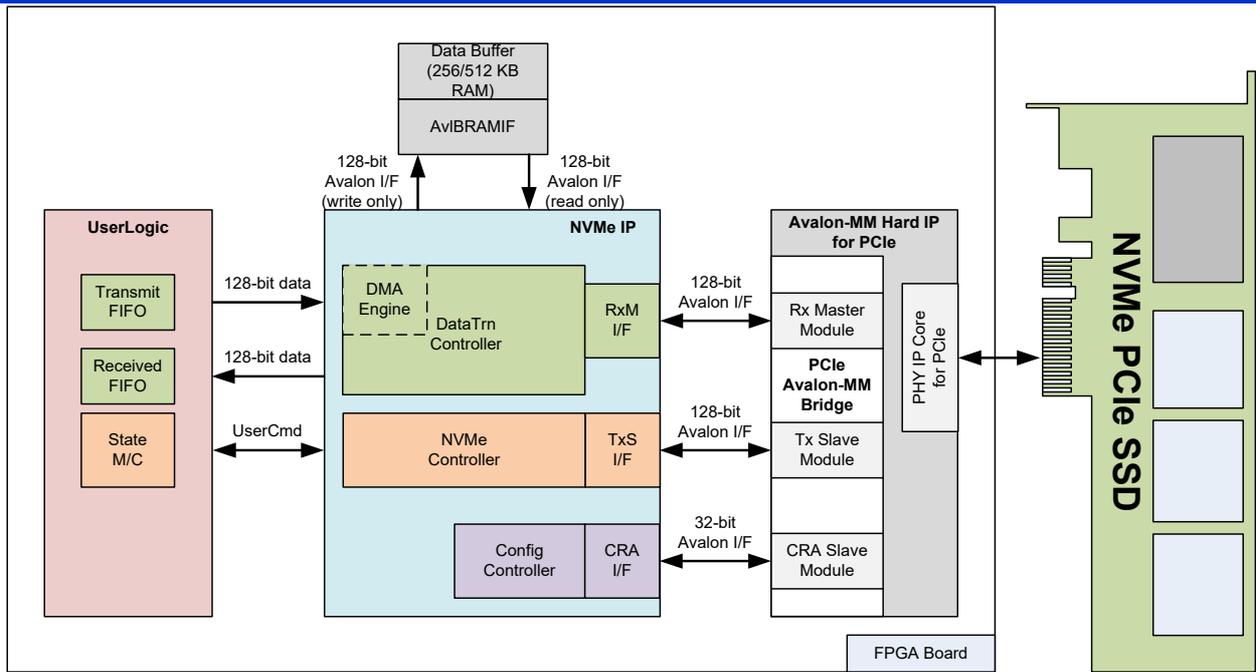


図 1: NVMe-IP コアのブロック図

アプリケーション情報

NVMe-IP コアは Altera 製 AvalonMM PCIe ブリッジ IP コアと組み合わせで動作し、NiosII 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 内蔵ブロック・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、Altera 標準の AvalonMM 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 FPGA 内部ブロック・メモリで構築したデータ・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 dgIF 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 AvalonMM PCIe ブリッジ IP からの出カ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 Altera 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 sof 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 内部ブロック・メモリではなく外部 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。

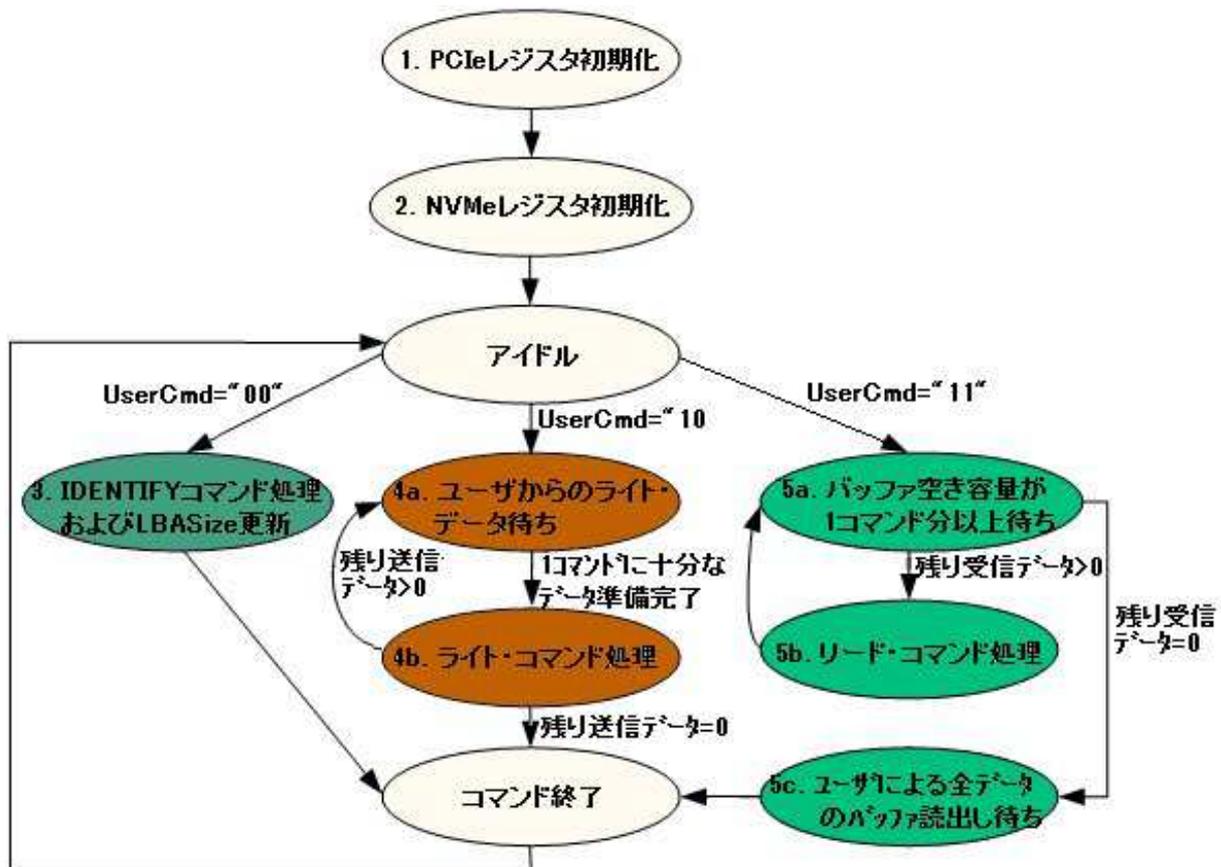


図 2: NVMe-IP コアの動作フロー

- 1) IP コアは AvalonMM 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 で動作するように初期化します。
- 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 AvalonMM 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 PCIe デバイス・クラスのチェック、BAR アドレスのセット、MSI 割込み許可、マスタ・モードの設定を行う初期化シーケンスを制御します。シーケンスの大半は Avalon MM PCIe ブリッジに対してコンフィグレーション・ライト/リード・サイクルを必要とします。1 パケットを送信するためには、AvalonMM PCIe ブリッジに対して多数のレジスタ・アクセスを必要とします。このコンフィグレーション・パケットを発生するシーケンスは CRA I/F 内で実装されます。

- **CRA I/F**

このモジュールは AvalonMM PCIe ブリッジ・コアを通してコンフィグレーションのライトやリードを行う TLP パケットを生成するために使われます。ライト・パケットの場合コンフィグレーション・コントローラからのアドレスとデータ入力からパケットを生成し、リード・パケットからリード・データをデコードしコンフィグレーション・コントローラに送信します。

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**

この回路ブロックは2つの機能を持ちます、1 つは初期化機能でもう一つはコマンド処理機能です。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。その後はユーザからのコマンド入力を待ちます。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・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。

- TxS 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 Avalon 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 AvalonMM PCIe ブリッジ IP の TxS 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。CRA 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 AvalonMM 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。FIFO からのライト・データは AvalonMM PCIe ブリッジ IP へ直接転送はされず FPGA 内部ブロック・メモリで構築したデータ・バッファを介して転送されます。また AvalonMM PCIe ブリッジ IP からのリード・データも一旦ブロック・メモリによるデータ・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 Avalon バスを通し DMA エンジンを使って行われます。

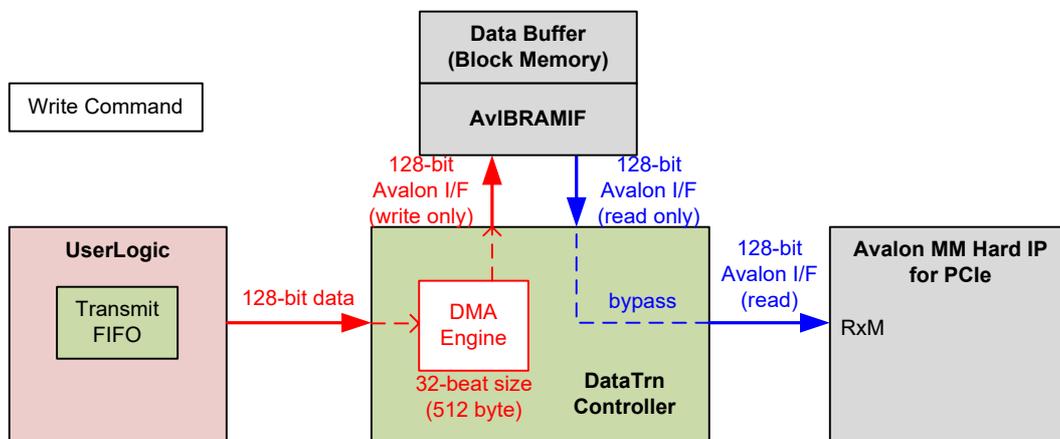


図 3: ライト・コマンドでのデータの流れ

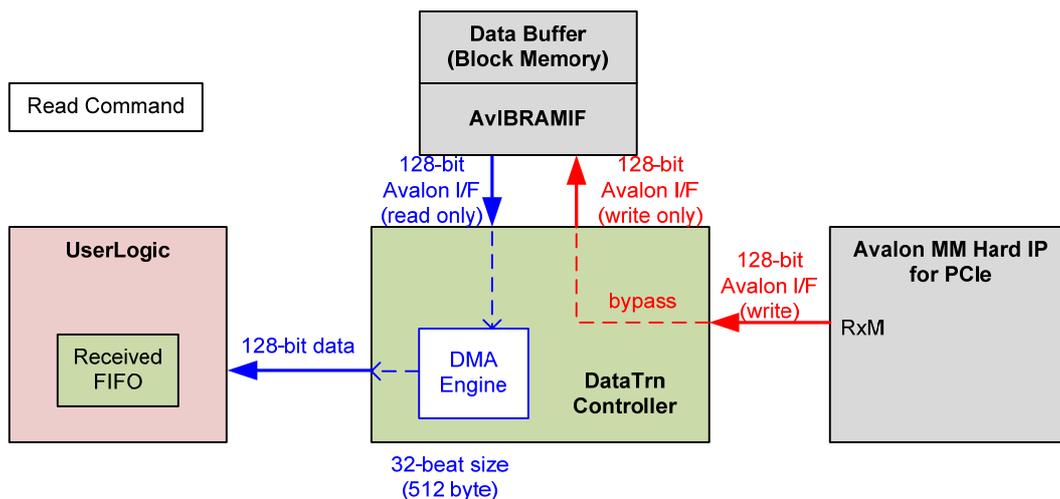


図 4: リード・コマンドでのデータの流れ

- **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・バッファは 2 つの 128bit AvalonMM マスターをサポートする必要がありますが、ひとつはライト・オンリー・ポートでもう一つはリード・オンリー・ポートです。どちらの接続もコマンドのデータ転送方向によって、内部 DMA エンジンまたは AvalonMM PCIe ブリッジの RxM と接続します。DMA エンジンは常にバースト・サイズを 32 ビットまたは 512 バイトでセットします。パフォーマンスを向上するため、リード要求はパイプライン転送をサポートし、データ・バッファからのリード・データ転送を待たずに多数のリード・コマンドを発行することができます。したがって内部ブロック・メモリと NVMe-IP コア間のインターコネクトは高いパフォーマンスを実現するためパイプライン転送をサポートする設定が必要となります。

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・タイプがデコードされ、パケットが処理されます。RxM ライト要求からのデータ・タイプは IDENTIFY データ、Admin 完了データ、または IO 完了データのいずれかとなります。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。RxM リード要求においてはデータ・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。

- **RxM I/F**

これは 128 ビット Avalon バスのスレーブ側です。データ・エリアにおいて RxM 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 Avalon 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。データはユーザが使い慣れた FIFO インターフェイスで転送できます。

Avalon-MM PCIe ブリッジ IP

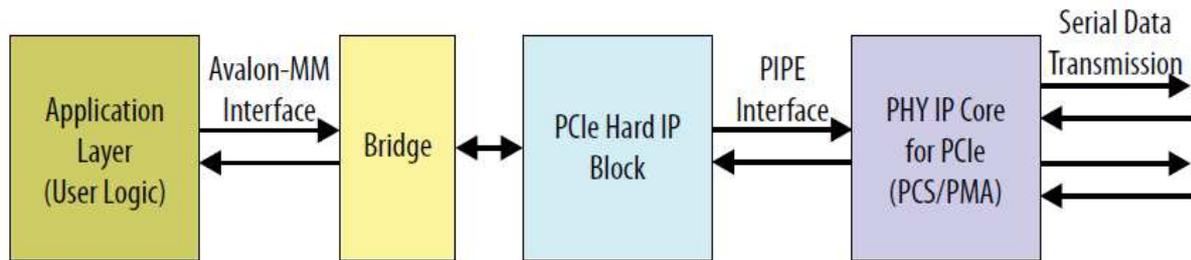


図 5: Avalon-MM PCIe ハード IP コアの内部構造

Avalon-MM インターフェイスを使う PCIe ハード IP コアは複雑な PCIe プロトコルを簡略化します。例えばトランザクション層プロトコル(TLP)エンコード/デコードを全て処理します。この IP コアは PCIe ハード IP ブロックを含むため、一つの FPGA で接続できる NVMe SSD の最大数は PCIe ハード IP ブロック数に制限されますのでご注意ください。より詳細については Altera 社発行の“ArriaV Avalon-MM Interface for PCIe Solutions”または“Arria10 Avalon-MM Interface for PCIe Solutions”ドキュメントを参照してください。

https://www.altera.com/en_US/pdfs/literature/ug/ug_a5_pcie_avmm.pdf

https://www.altera.com/en_US/pdfs/literature/ug/ug_a10_pcie_avmm.pdf

コアの I/O 信号

コアの I/O 信号について下表 2 に説明します。

表 2: 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	Avalon バス・インターフェイスと同期する AvalonMM 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)用 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

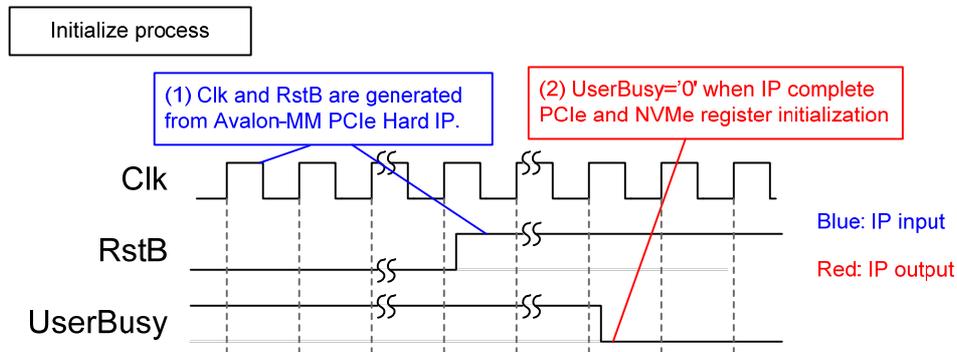
信号名	方向	説明
NVMe IP インターフェイス		
TestPin[31:0]	Out	本コアのテスト出力端子(通常は非使用)
TimeOutSet[31:0]	In	接続 SSD のコマンド完了待ちタイムアウト設定値 設定時間単位は Clk クロック周期単位(Gen2 の場合 8ns で Gen3 の場合 4ns)
LinkSpeed[1:0]	Out	PCIe リンク速度 "00": 未リンク状態, "01": Gen1 (2.5 Gbps), "10": Gen2 (5.0 Gbps), "11": Gen3 (8.0 Gbps).
PCIeInt	In	Avalon-MM PCIe ハード IP からの割込み出力.
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] - MPSMIN (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.
CRA インターフェイス		
CraChipSel	Out	アサートし CRA ポートを選択
CraAddress[13:0]	Out	ライト/リード先のバイト・アドレス、32 ビット・アクセスのため Bit[1:0]は常に 00b となる
CraByteEnable[3:0]	Out	バイト・イネーブル
CraRead	Out	リード要求、有効なリード・アドレスが出力されていることを示す
CraReadData[31:0]	In	リード・データ
CraWrite	Out	ライト要求、有効なライト・アドレス/ライト・データ/バイト・イネーブルが出力されていることを示す
CraWriteData[31:0]	Out	ライト・データ
CraWtRequest	In	ウェイト要求
TxS インターフェイス		
TxSChipselect	Out	アサートし送信スレーブ・ポートを選択
TxSByteEnable[15:0]	Out	ライト・データのバイト・イネーブル信号
TxSReadData[127:0]	In	PCIe ブリッジ IP コアからのリード・データ
TxSWriteData[127:0]	Out	送信スレーブ・ポートへのライト・データ
TxSRead	Out	リード要求信号
TxSWrite	Out	ライト要求信号
TxSBurstCount[5:0]	Out	要求されたデータ転送量を示す、シングル・アクセスの場合常に 000001b
TxSReadDataValid	In	有効なリード・データが出力されていることを示す
TxSWaitRequest	In	リード・データまたはライト・データのウェイト要求
TxSAddress[28:0]	Out	ライト/リード・アドレス

信号名	方向	説明
RxM インターフェイス		
RxMAddress[31:0]	In	アクセスのライト/リード・アドレス
RxMBurstCount[5:0]	In	ライト/リード要求での正確なバースト・カウント
RxMByteEnable[15:0]	In	ライト・データ用のバイト・イネーブル
RxMWrite	In	PCIe IP コアからのライト要求
RxMRead	In	PCIe IP コアからのリード要求
RxMWaitRequest	Out	データ転送のウェイト要求
RxMWriteData[127:0]	In	ライト・データ
RxMReadData[127:0]	Out	リード・データ
RxMReadDataValid	Out	リード・データが有効であることを示す
データ・バッファ用の Avalon MM マスター・インターフェイス		
BufAvwAddress[31:0]	Out	アクセスのライト・アドレス
BufAvwBurstCount[5:0]	Out	ライト要求での正確なバースト・カウント
BufAvwByteEnable[15:0]	Out	ライト・データ用のバイト・イネーブル
BufAvwWrite	Out	データ・バッファへ送られるライト要求
BufAvwWaitRequest	In	ライト転送でのデータ保持(ウェイト)要求
BufAvwWriteData[127:0]	Out	ライト・データ
BufAvrAddress[31:0]	Out	アクセスのリード・アドレス
BufAvrBurstCount[5:0]	Out	リード要求での正確なバースト・カウント
BufAvrByteEnable[15:0]	Out	リード・データ用のバイト・イネーブル、常に 0xFFFF がセットされる
BufAvrRead	Out	データ・バッファへ送られるリード要求
BufAvrWaitRequest	In	リード転送でのデータ保持(ウェイト)要求
BufAvrReadData[127:0]	In	リード・データ
BufAvrReadDataValid	In	リード・データが有効であることを示す

タイミング波形

初期化

NVMe-IP コアの同期クロック信号(Clk)とリセット信号(RstB)は AvalonMM PCIe ブリッジ IP 内で作られます。RstB が解除されると NVMe-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) Clk と RstB は AvalonMM PCIe ブリッジ IP で生成されます
- (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

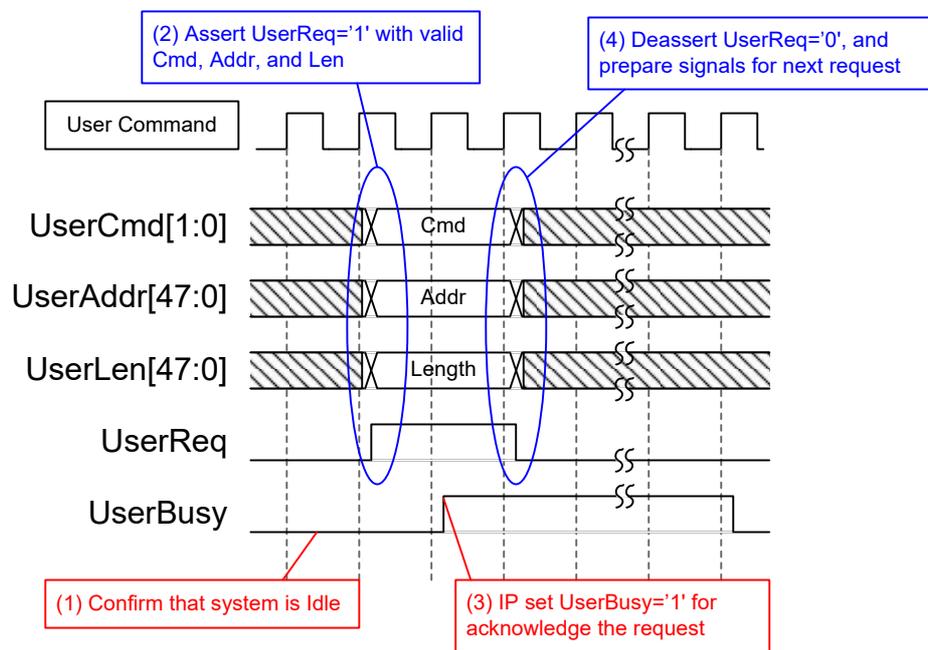
図 6: システム起動時の RstB と UserBusy のタイミング

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。

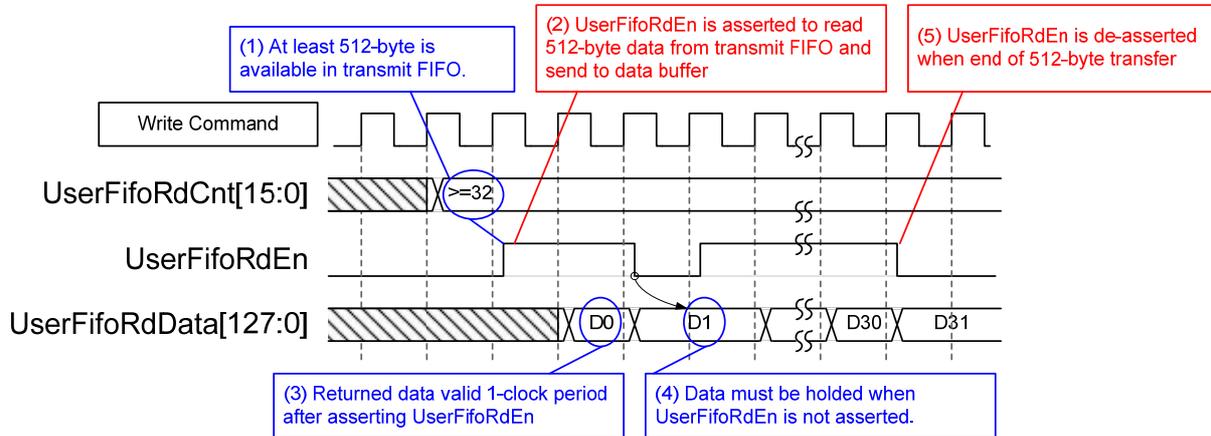
(注意) UserAddr や UserLen は Identify コマンドでは使われなため無視されます
 一方データ・インターフェイスにおいてはライト・コマンドにおけるデータ送信 FIFO はコアから読み出され、リード・コマンドにおけるデータ受信 FIFO はコアからライトされます。データ・インターフェイスのタイミング波形は図8および図9で示します。



- (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

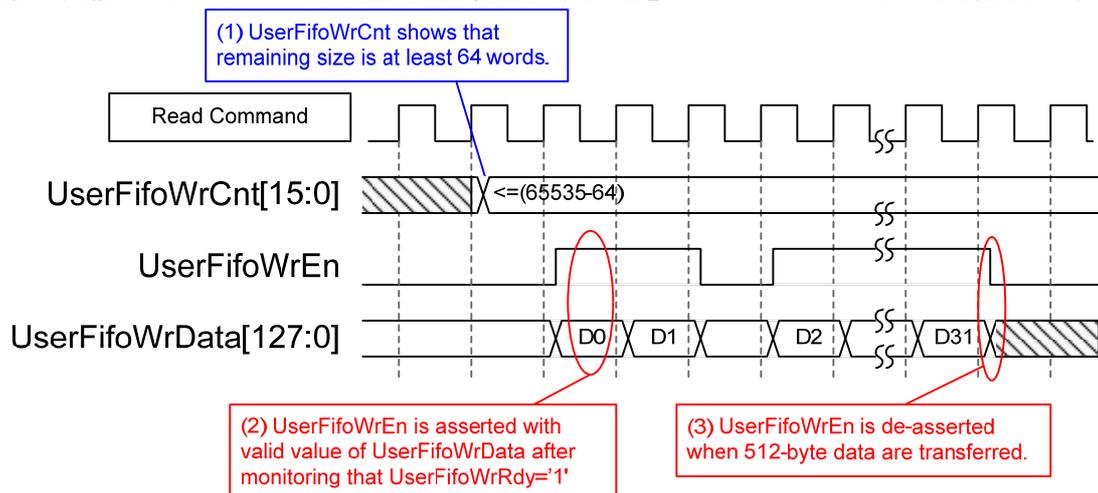
ライト・コマンドにおいてユーザ回路内の送信 FIFO からのライト・データは 128 ビット Avalon-MM バス・インターフェイスを介してデータ・バッファへと転送されます。データ・バッファへのバースト・サイズは 32 ビットまたは 512 バイト固定です。IP コアはバースト転送を開始する前に UserFifoRdCnt をモニタし、少なくとも 512 バイトのライト・データが送信 FIFO にあることを確認します。一般的な FIFO と同様、UserFifoRdData は図 8 に示すように UserFifoRdEn がアサートされた次のクロック・ペリオドで有効な値となり、またネゲー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

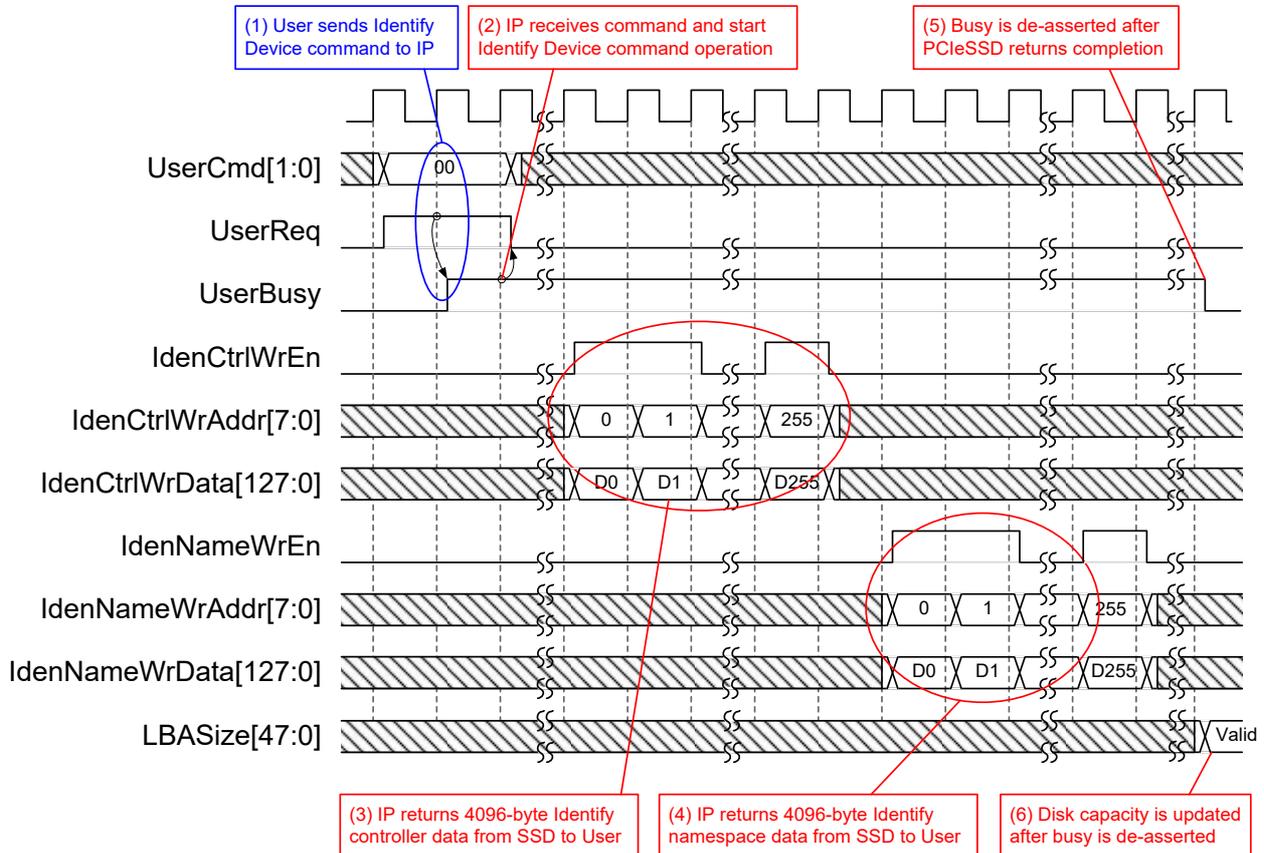


- (1) 受信 FIFO に 64 ワード(512 バイト)以上の空き領域があることを UserFifoWrCnt で確認します
- (2) 512 バイト以上の空き領域を確認した次クロックから UserFifoWrEn と UserFifoWrData が出力されます
- (3) UserFifoWrEn は 512 バイトを転送すると'0'ネゲートします

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

NVMe-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。



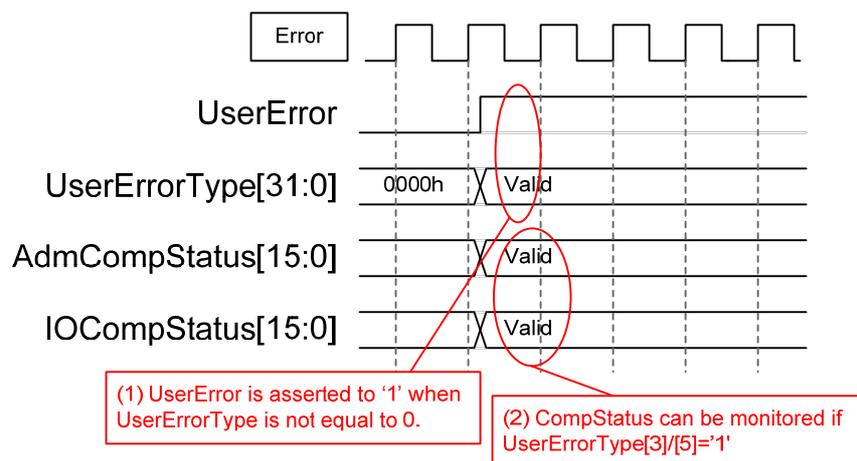
- (1) ユーザ回路は IDENTIFY コマンドの発行を IP コアに要求します
- (2) IP コアはコマンド要求を受け、IDENTIFY コマンド実行を開始します
- (3) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY コントローラ・データをユーザ回路に出力します
- (4) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY ネームスペース・データをユーザ回路に出力します
- (5) NVMeSSD が完了を返送すると UserBusy がネゲートされます
- (6) UserBusy がネゲートされるとディスク容量情報となる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

コアの検証方法

NVMe-IP コアは ArriaV GX スタータ・キットや Arria10SoC 開発キット等の Altera 標準評価ボードと AB16-PCIeXOVR アダプタ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、Altera の QuartusII 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 Altera 各ファミリが対象となります。

製品型番	対象ファミリ	実機評価ボード	発売状況
NVMe-IP-A5GX	ArriaV GX	DK-START-5AGXB3N	発売中
NVMe-IP-A10SX	Arria10 SX	DK-SOC-10AS066S-A	発売中

IP コアの価格やライセンス条件あるいは上記以外のファミリ対応状況につきましては、デザイン・ゲートウェイ (sales@dgway.com) または国内 Altera 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、対象ファミリの Altera 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
1.0J	2016/08/10	日本語版の初期版リリース
1.1J	2016/12/20	データ・バッファを外付け DDR から内蔵ブロック・メモリへ改良