



**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](http://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 (dgIF typeS)
- PCIe ハード IP と Avalon-ST にて 128 ビット・バスで直結
- ユーザ回路側クロックは PCIe クロック(Gen2 の場合 125MHz, Gen3 の場合 250MHz)と同じかそれ以上で動作
- 内蔵ブロック・メモリで 256KByte のデータ・バッファを構築、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)
  - MPSSMIN (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,Arria10SoC 開発キット,Arria10GX 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 Arria10GX 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 (MODE1)

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers	Block Memory bit <sup>1</sup>	Design Tools
ArriaV GX	5AGXFB3H4F35C4	212	1175	2133	2,162,688	QuartusII 16.0
Arria10 SX	10AS066N3F40E2SGE2	280	1144	2120	2,162,688	QuartusII 16.0

注:

1) 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

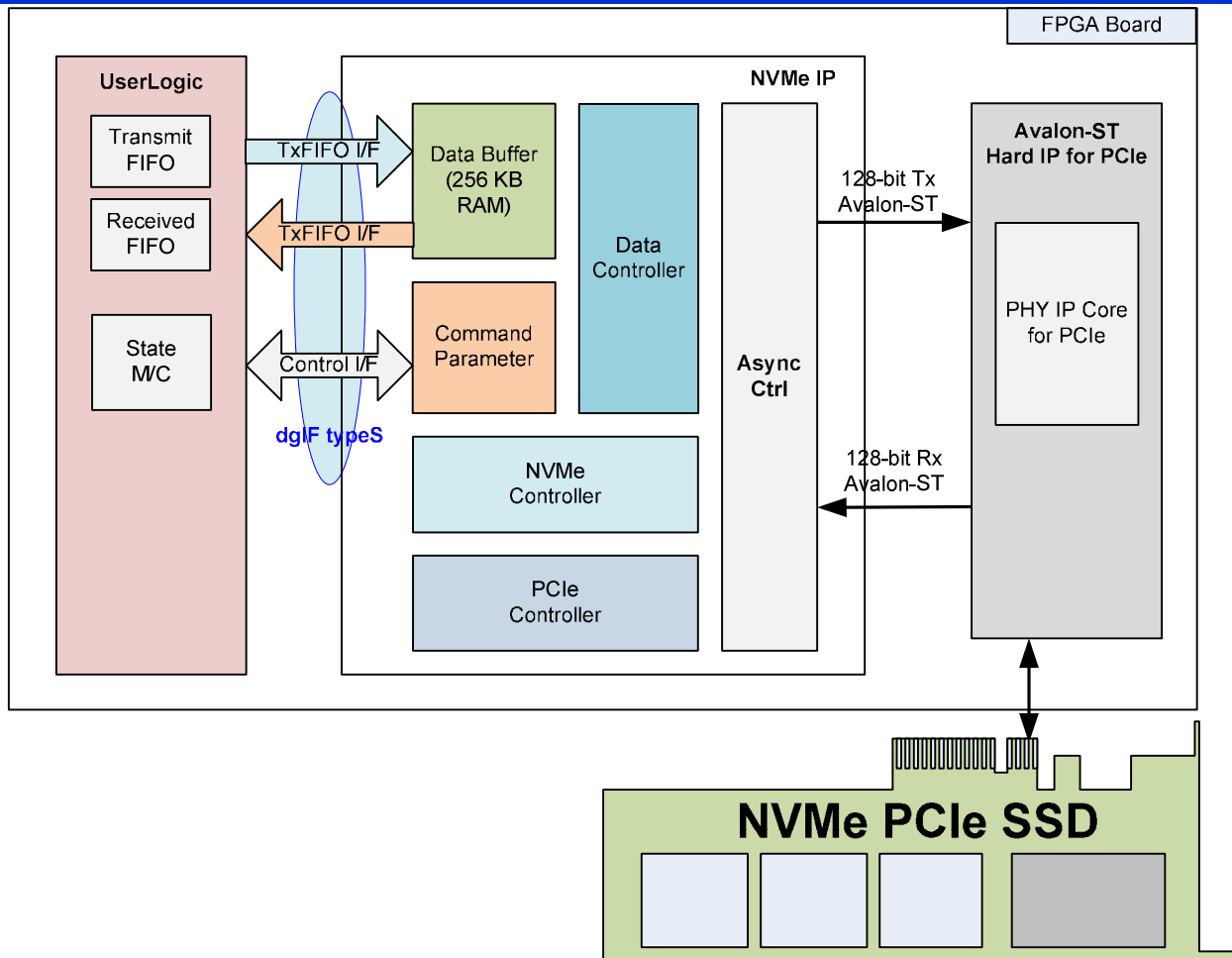


図 1: NVMe-IP コアブロック図

## アプリケーション情報

NVMe-IP コアは Intel 製 PCIe ハード IP の Avalon-ST インターフェイスと直結して動作し、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理想的なソリューションを提供します。256K バイトの 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 間のデータを転送するため外付けの DDR メモリを必要としません。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(以下本 IP コアとします)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、Avalon-ST バスで接続した FPGA トランシーバ内蔵の PCIe ハード IP コアを使うことで PCIe プロトコルを適用します。本 IP コアは Identify, ライト, リードの 3 種類の NVMe コマンドをサポートします。NVMe SSD は複数のコマンドを並列して実行するため、本 IP コアは同時に発行される多数のライト/リード・コマンドでのデータを送信/受信するためデータ・バッファを内蔵します。複数コマンドを同時処理することにより、NVMe SSD の持つ最大限のパフォーマンスを引き出すことができます。

本 IP コアのユーザ・インターフェイスはシンプルなインターフェイス(弊社 dglF 規格)であり、制御(コマンド)系とデータ系に分かれます。制御系インターフェイスはコマンド種類・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です。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。Avalon-ST PCIe ハード IP コアの制限により、ユーザ・ロジック側のクロック周波数は PCIe クロック周波数 (PCIe Gen3 の場合 250MHz、PCIe Gen2 の場合 125MHz) と同じかそれ以上とする必要があります。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 Intel 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。

## 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。

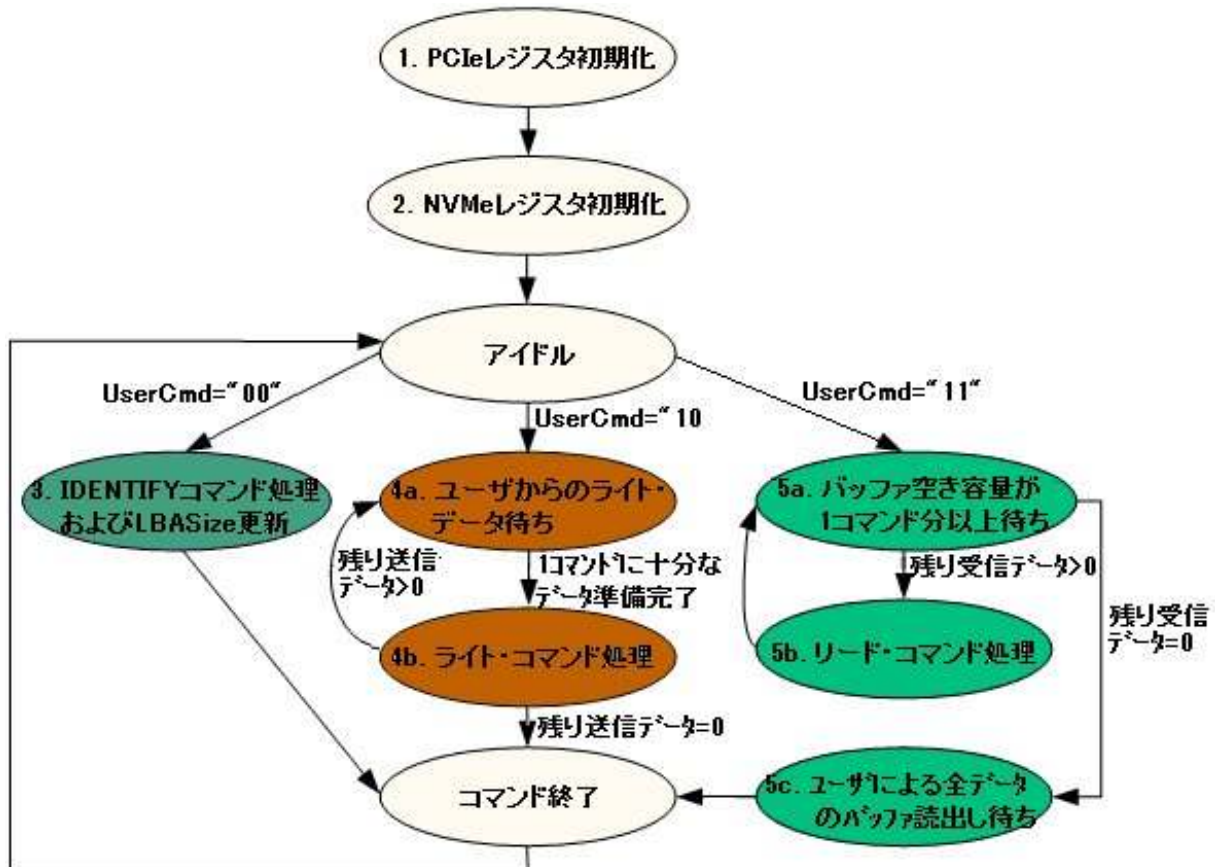


図 2: NVMe-IP コアの動作フロー

- 1) 本 IP コアは 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) 本 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メモリ空間をアクセスしてセットします。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、以下の処理を行います
  - ✓ 本 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。(本 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)
  - ✓ 本 IP コアは NVMe SSD に対してライト・コマンドを発行します。
  - ✓ 本 IP コアはコマンドの全データが完全に転送されたことを示す SSD からのステータスを待ちます。
  - ✓ 残り転送セクタ数がゼロでない場合、本 IP コアは次に続くコマンド用としてデータ・バッファ内のライト・データ数をチェックし続けます。
  - ✓ 残り転送セクタ数がゼロとなると本 IP コアはアイドル状態に復帰します。
- 5) リードコマンドの場合、以下の処理を行います
  - ✓ 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ます。
  - ✓ 本 IP コアは NVMe SSD に対してリード・コマンドを発行します。
  - ✓ 本 IP コアはコマンドの全データが完全に転送されたことを示す SSD からのステータスを待ちます。
  - ✓ 残り転送セクタ数がゼロでない場合、本 IP コアは次に続くコマンド用として空き容量をチェックし続けます。
  - ✓ 残り転送セクタ数がゼロとなると本 IP コアはアイドル状態に復帰します。

図 2 に示した動作シーケンスを実装するため、本 IP コアは 3 つのサブ・モジュールに分かれます、すなわち PCIe コントローラ、NVMe コントローラ、データ・コントローラです。PCIe コントローラはシステムの電源が投入されると SSD と通信するため PCIe 環境をセットアップします。次に NVMe コントローラが NVMe 規格に準じて SSD 内 NVMe レジスタを初期化し、全初期化フェーズを完了します。

コマンド実行フェーズにおいては、ユーザ回路が本コアのユーザ・インターフェイス(dglF typeS)を通してコマンドを発行することで開始します。NVMe コントローラはユーザ回路からのコマンドをデコードし入力されたコマンド・パラメータを本 IP コアにセットします。各 NVMe コマンドのシーケンスは NVMe コントローラにより制御されます。コマンド・パケット、ステータス・パケット、データ・パケットはデータ・コントローラで処理されます。

## PCIe

NVMe プロトコルは物理インターフェイスや下位層プロトコルに PCIe 規格を使います、このため初期化シーケンスと下位層の通信は PCIe コントローラ内で実装されます。

- **PCIe コントローラ**

本モジュールはステート・マシンを内蔵し PCIe クラスのチェック、BAR アドレスのセット、マスター・モードの指定を実行します。PCIe 環境をセットする基本パラメータは SSD 内のコンフィグレーション空間内にマッピングされます。コンフィグレーション空間へのライトやリードを行うため、パケットが 128 ビット送信/受信 Avalon-ST バスを通して転送されます。

Avalon-ST PCIe ハード IP コアはまず最初に 128 ビット送信/受信 Avalon-ST バスを介してコンフィグレーション空間をセットアップする必要があります。PCIe コントローラ内蔵のステート・マシンはこの Avalon-ST PCIe ハード IP 内のコンフィグレーション空間への初期化シーケンスを実装しています。

## NVMe

NVMe 規格ではインターフェイス用として 4 つのキュー・タイプが使われます、すなわち NVMe ホストが Admin コマンドを発行するための Admin サブミッション、SSD が ACK を返送するための Admin 完了、ホストが I/O コマンドを発行するための I/O サブミッション、SSD が ACK を返送するための I/O 完了の 4 つです。SSD へ新たなコマンドを発行する場合、NVMe ホストはコマンドをサブミッション・キューにセットしドアベル・レジスタのサブミッション・キュー末尾ポインタを更新します。SSD がコマンド処理を完了すると SSD は完了キューに完了ステータスをセットします。NVMe ホストはプロセス完了後、ドアベル・レジスタの完了キュー先頭ポインタを更新します。各コマンド動作のシーケンスは NVMe コントローラ内で実装されますが、データ・パケットはデータ・コントローラ内で処理されます。データ・パケットは 2 種類あり、すなわち一種類はデータ・バッファに保持する生データでもう一種類はコマンド・パラメータ内に保持される制御/ステータス・データです。

- **NVMe コントローラ (NVMe Controller)**

ユーザ回路が本 IP コアへ新たなコマンドを送ると NVMe コントローラはユーザ回路から指定されたコマンド種類・アドレス・転送長を処理します。その後コマンド・パラメータを保持するためサブミッション・キューを作成しドアベル・レジスタにサブミッション・キューの末尾ポインタを更新します。ライト/リード・コマンドにおいて総転送数が 128K バイトより大きい場合、NVMe コントローラは複数のコマンドに分割します。

ライト・コマンドの場合、ユーザ回路へのビジー信号は I/O 完了キューにエラーなしの状態が SSD から報告された後でクリアされます。リード・コマンドの場合はユーザ回路がデータ・バッファ内の全データを読み出した後でビジー信号がクリアされます。

- **データ・バッファ (Data Buffer)**

256K バイトのブロック・メモリで実装されます。ライト・コマンドの場合ユーザ回路からの生データを一時的に保持するために使われ、リード・コマンドの場合 SSD からユーザ回路へ転送する生データを保持します。

- **コマンド・パラメータ (Command Parameter)**

このブロックはコマンド・パケット(Admin および I/O サブミッション・キュー)とステータス・パケット(Admin および I/O 完了キュー)を保持します。コマンド・パケットは NVMe コントローラにより生成されデータ・コントローラにより読み出されます。ステータス・パケットはデータ・コントローラにより送信され NVMe コントローラにより読み出されます。

- **データ・コントローラ (Data Controller)**

本ブロックでは 2 種類のデータ・タイプが処理されます。データ・バッファ内の生データとコマンド・パラメータ内の制御データはメモリ空間の異なるエリアにマッピングされます。データ・コントローラは SSD から送られてくる要求に含まれるアドレス情報をデコードすることでデータ送信元またはデータ受信先を選択します。PCIe ハード IP へ送信するパケットを生成するために、データ・バッファからのデータやコマンド・パラメータは PCIe ハード IP から受信した要求パケット内から抽出した TLP ヘッダと結合します。このため本ブロック内には TLP パケットのヘッダを保持する少量のメモリ空間を含みます。データ・コントローラのデータ・バス幅は PCIe ハード IP コアのバス・サイズと同じ 128 ビットです。

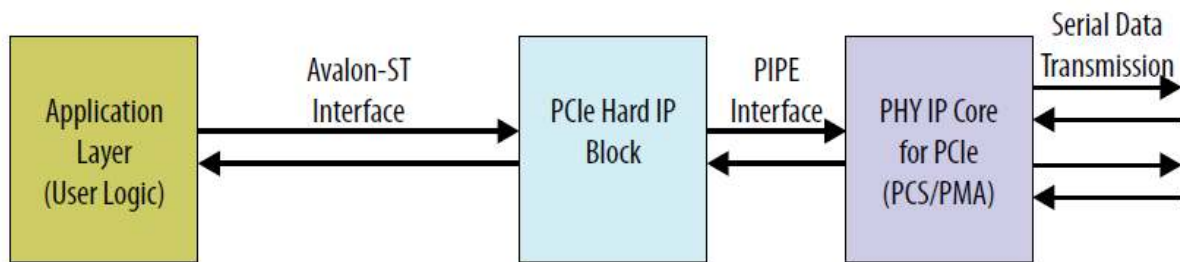
- **非同期コントロール(AsyncCtrl)**

本 IP コアは PCIe クロックと異なるユーザ・クロックのクロック・ドメインをサポートするためのブロックです。ただし**ユーザ・クロックは PCI クロックと同じかそれ以上の周波数でなくてはなりません**。この非同期コントロールは異なるクロック・ドメイン間の通信を行うため分散 RAM で実装する少量の非同期 FIFO を内蔵します。

## ユーザ回路

ユーザ回路はコマンド種類・アドレス・転送長パラメータをコアへ送信するだけのシンプルなインターフェイス(dgIF typeS)のため、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。

## Avalon-ST PCIe ハード IP



**図 3: Avalon-ST PCIe ハード IP コアのアーキテクチャ**

PCIe ハード IP ブロックは Intel 製 FPGA にて PCIe プロトコル・スタックを内蔵します。このハード IP ブロックはトランシーバ・モジュール、物理層、データ・リンク層、トランザクション層を含みます。1 個の FPGA に接続できる NVMe SSD の最大数はその FPGA が内蔵する PCIe ハード IP ブロックの個数と一致します。Avalon-ST PCIe ハード IP のより詳細については Intel 社発行の”ArriaV Avalon-ST Interface for PCIe Solutions User Guide”や、”Intel Arria10 Avalon-ST Interface for PCIe Solutions User Guide”ドキュメントをを参照してください。

[https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_a5\\_pcie\\_avst.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_a5_pcie_avst.pdf)

[https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_a10\\_pcie\\_avst.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_a10_pcie_avst.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 の同期リセット信号、Clk 信号が安定となったら解除すること
Clk	In	NVMe-IP コアの動作システム・クロック。Clk の周波数は Avalon-ST PCIe ハード IP から出力される PCIeCLK と同じかそれ以上とする必要がある。(PCIe Gen2 の場合 125MHz、Gen3 の場合 250MHz)
ユーザ・インターフェイス (dgIF TypeS、Clk に同期)		
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。SSD の特性から本アドレスの下位 3bit[2:0]='000'として SSD ページ・サイズとなる 4K バイト・サイズのアライメントに合わせることが推奨される。大多数の SSD では開始セクタ・アドレスが 4K バイト単位のアライメントに合っていない場合ライト/リードのパフォーマンスが低下する。
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	<p>エラー・ステータス情報、各ビットの意味は以下の通り</p> <ul style="list-style-type: none"> <li>[0] - PCIe クラス・コードが正しくない</li> <li>[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 <ul style="list-style-type: none"> <li>* MPSMIN (メモリ・ページ・サイズ最小値)がゼロでない</li> <li>* NVM コマンド・セット・フラグ(CAP レジスタの bit37)が'1'ではない</li> <li>* DSTRD (ダブル・ストライド)が'0'でない</li> <li>* MQES (サポートする最大キュー・エン트리)が'7'あるいはそれ以上</li> </ul>                     各レジスタのより詳細は NVMeCAPReg 信号を参照することで確認できる                 </li> <li>[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</li> <li>[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</li> <li>[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</li> <li>[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</li> <li>より詳細は IOCompStatus 信号を参照すること</li> <li>[6] - 完了 TLP パケット・サイズが正しくない</li> <li>[7] - Avalon-ST PCIe ハード IP が内部バッファから ECC エラーを検出した</li> <li>[8] - サポートしていない要求(UR)フラグが完了 TLP パケットで報告された</li> <li>[9] - コンプリータ・アポート(CA)フラグが完了 TLP パケットで報告された</li> <li>[10] - メモリ・ライト要求 TLP パケットの Length[1:0]値が 0 でないか、128bit 単位のアライメントに合っていない</li> <li>[11] - メモリ・ライト要求 TLP パケットの Address[3:2]が 0 でないか、128bit 単位のアライメントに合っていない</li> <li>[31:12] - 未使用</li> </ul> <p>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</p>



信号名	方向	説明
ユーザ・インターフェイス [続き] (dglF TypeS、Clk に同期)		
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

(※) 注意

- 上記信号名で"UserFifo"がつく dglF typeS データ系信号は本 IP コアと FIFO 間での接続信号です、コアと FIFO 間の具体的な接続方法はリファレンス・デザインを参照してください。
- ユーザ回路は FIFO を挟んでコアと FIFO 反対側で接続するので一般的な FIFO 制御方法となります、FIFO は Intel 標準 FIFO ライブラリをそのまま使用します。

信号名	方向	説明
NVMe IP インターフェイス (Clk に同期)		
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 (1/Clk 周波数)
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).
PCIeLinkup	In	PCIe ハード IP の LTSSM ステートが L0 ステートの場合'1'アサート
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.

(※) 注意

- 各信号の具体的な接続方法についてはリファレンス・デザインを参照してください。

信号名	方向	説明
Avalon-ST PCIe ハード IP インターフェイス		
PCleRstB	In	アクティブ Low の同期リセット信号、PCIe ハード IP コアがリセット状態でない場合解除する
PCleClk	In	Avalon-ST PCIe ハード IP コアからのクロック出力を接続する (PCIe Gen2 の場合 125 MHz、PCIe Gen3 の場合 250 MHz)
PCleRxValid	In	PCleRxData が有効であることを示す。PCleRxReady がネゲートされてから 2 クロック以内にネゲートする
PCleRxEOP	In	PCleRxValid がアサート中に TLP の最終サイクルであることを示す
PCleRxReady	Out	NVMe-IP コアはデータを受信できる状態であることを示す
PCleRxError	In	Avalon-ST PCIe ハード IP 内の受信バッファで修正できない ECC エラーが発生したことを示す
PCleRxData[127:0]	In	受信データ・バス
PCleTxValid	Out	PCleTxReady がアサート中に PCleTxData が有効であることを示す
PCleTxSOP	Out	PCleTxValid とともに TLP の先頭データであるkとを示す
PCleTxEOP	Out	PCleTxValid とともに TLP の最終データであるkとを示す
PCleTxEmpty	Out	パケット最終(PCleTxEOP アサート時)の上位 QWord(上位 64bit)が有効データを含むかどうかを示す。'0': PCleTxData[127:0]すべてが有効, '1': PCleTxData[63:0](下位)のみ有効
PCleTxReady	In	Avalon-ST PCIe ハード IP がデータを受信できることを示す
PCleTxError	Out	送信した TLP にエラーがあったことを示す、この信号は常に'0'を出力する
PCleTxData[127:0]	Out	送信データ・バス

## タイミング波形

### 初期化

本 NVMe-IP コアの初期化シーケンスは以下となります。

- 1) 同期クロック信号(Clk 入力信号)が安定すると RstB がユーザ回路によりネゲートします。
- 2) 本 IP コアは PCIeRstB と PCIeLinkup の両方が '1' となるまで待機し、PCIe ハード IP コアの動作準備が整ったことを確認します。
- 3) リセット動作が完了すると PCIeRstB が '1' ネゲートされます、PCIeRstB は PCIeClk ドメインで生成されます。
- 4) PCIe ハード IP コアの LTSSM ステートが L0 に遷移し PCIeLinkup がアサートされます。
- 5) 本 IP コアは初期化プロセスを開始します。
- 6) コアによる初期化プロセスが完了すると UserBusy がネゲートされます。

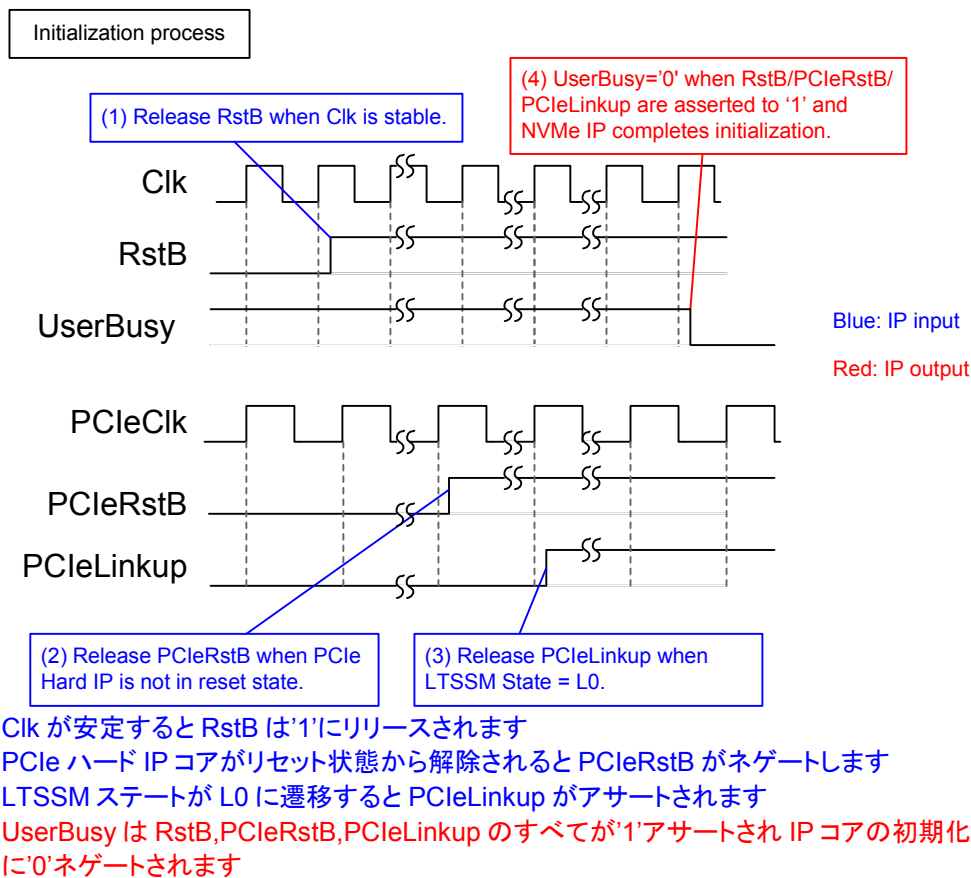


図 4: システム起動時の UserBusy のタイミング

ユーザ・インターフェイス(dglF typeS)

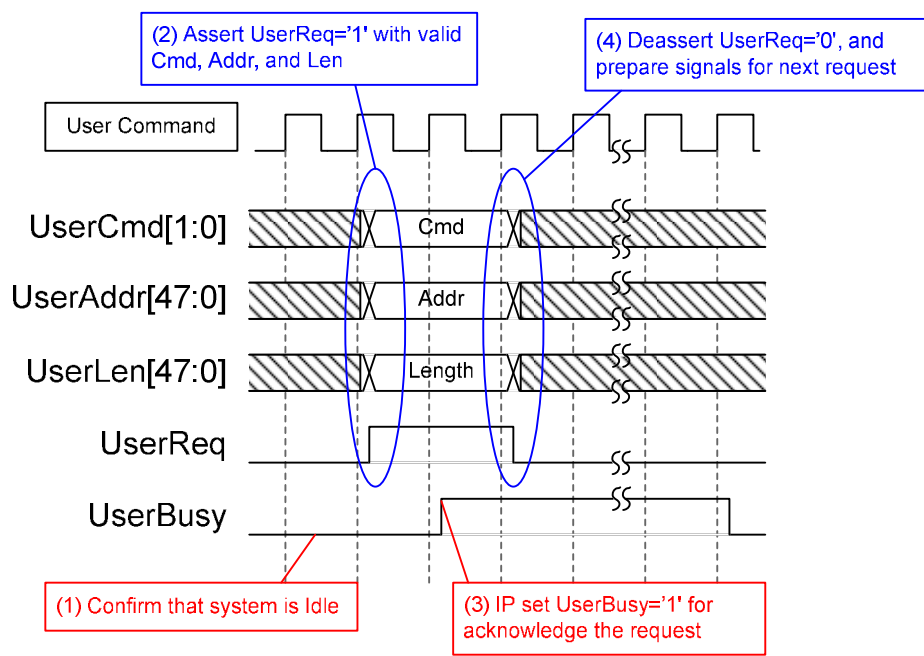
制御インターフェイス

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、すなわち一つは制御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 dglF で、dglF 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 5 に制御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 6 および図 7 で示します。

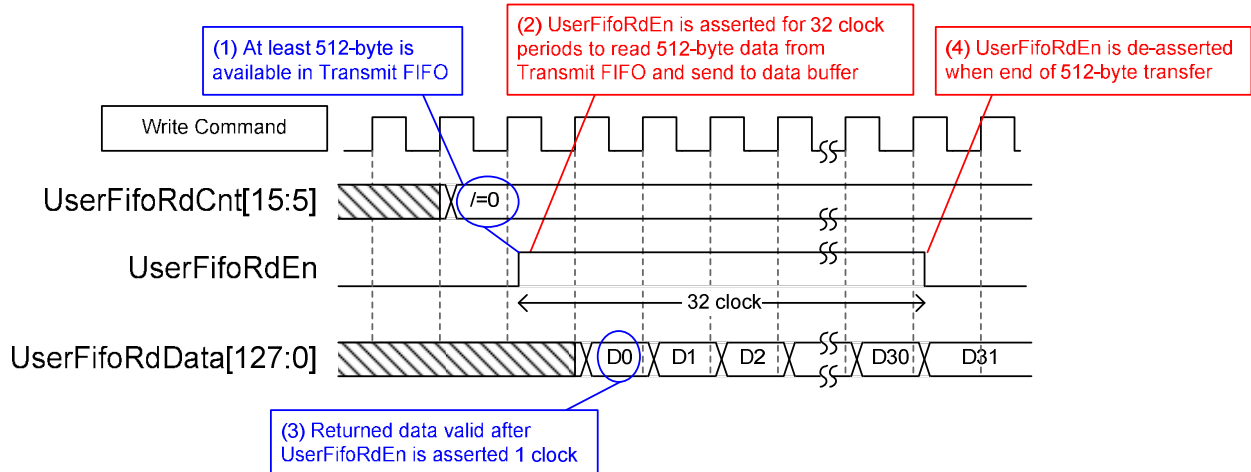


- (1) システムがアイドル状態であることを確認します
- (2) 有効なコマンド,アドレス,転送長をセットし UserReq='1'をアサートします
- (3) IP コアは要求を受付けたことを UserBusy='1'で示します
- (4) ユーザ回路は UserReq='0'とし、次のコマンドのためのパラメータを準備できます

図 5: ユーザ・コマンドのタイミング波形

## データ・インターフェイス

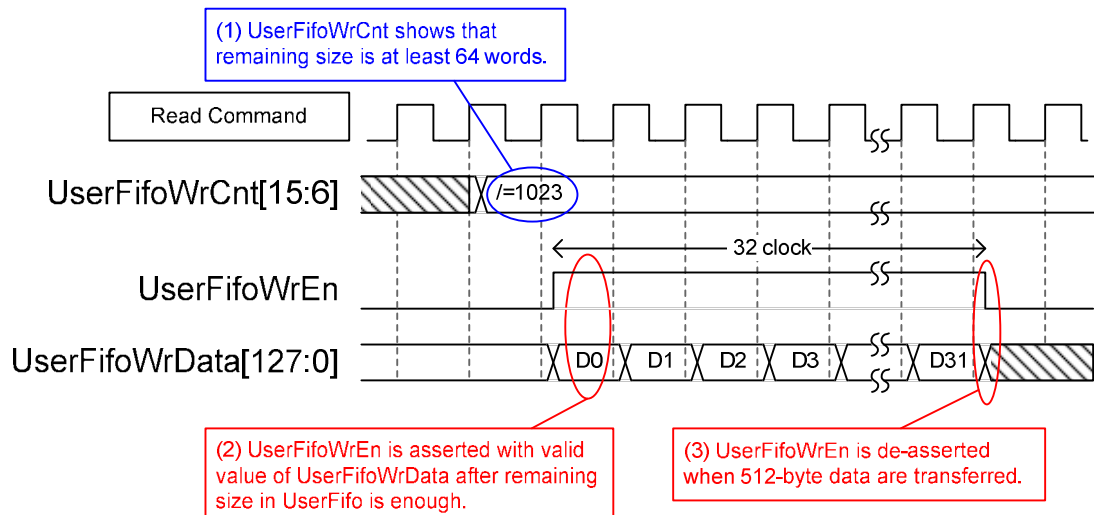
ライト・コマンドにおいてユーザ回路内の送信 FIFO からの 128 ビットのライト・データは本 IP コア内のデータ・バッファへと転送されます。IP コア内部 DMA エンジンは UserFifoRdCnt 信号を監視し送信 FIFO 内のデータが 512 バイトまたはそれ以上となるまで待機します。その後図 6 に示すように UserFifoRdEn を 32 クロック期間アサートし 512 バイト・データを転送します。一般的な FIFO タイミングと同様、UserFifoRdData は UserFifoRdEn がアサートされた次のクロック期間で出力されます。



- (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 は 32 クロック期間連続してアサートされ送信 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) 512 バイトの転送後 UserFifoRdEn は'0'ネゲートします

図 6: 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。リード・コマンドにおいても、512 バイト・データを FIFO へ転送する前に受信 FIFO に 1024 バイトまたはそれ以上の空き容量があることを UserFifoWrCnt によって確認します。



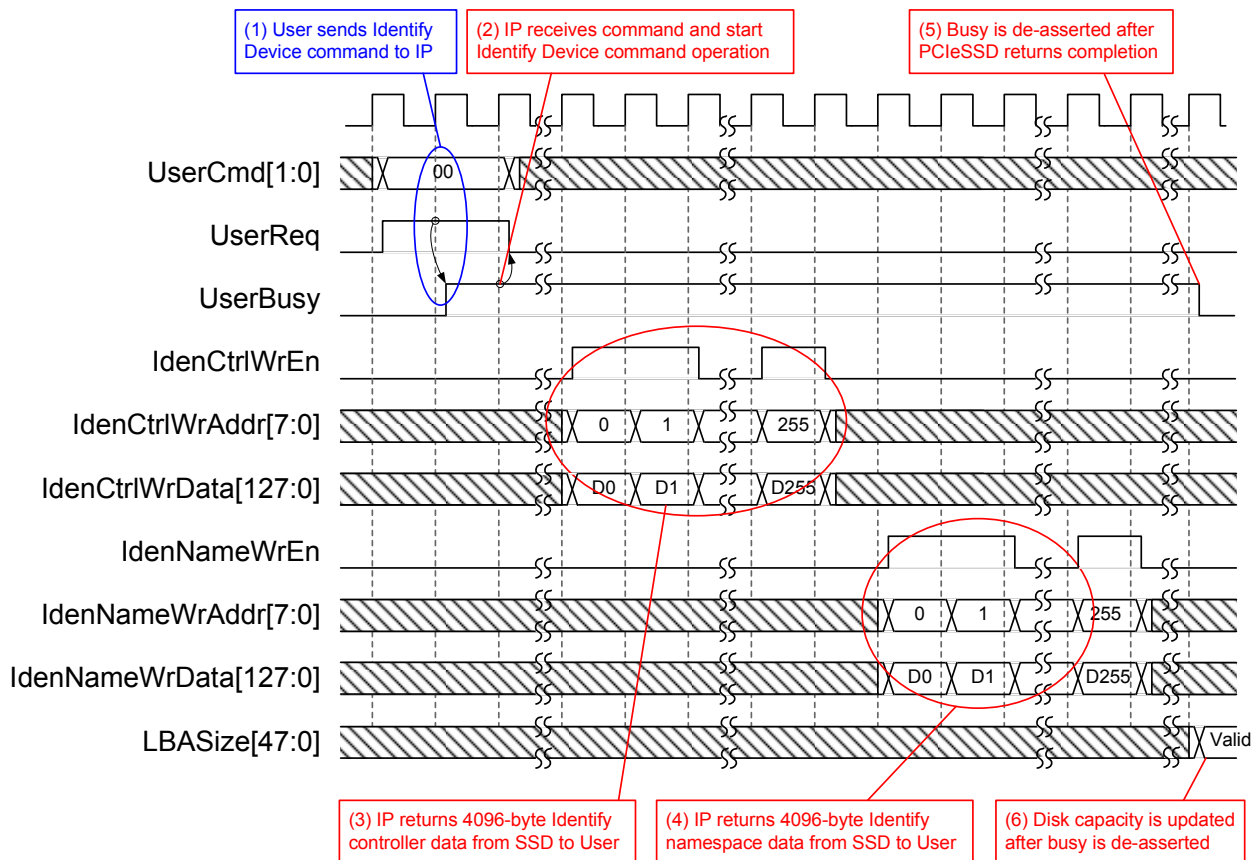
- (1) 受信 FIFO に 64 ワード以上の空き領域があることを UserFifoWrCnt で確認します
- (2) 十分な空き領域を確認すると UserFifoWrEn と UserFifoWrData が出力されます
- (3) UserFifoWrEn は 512 バイトを転送すると'0'ネゲートします

図 7: 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

図 6 および図 7 で示したデータ・インターフェイスは本 IP コアと送信 FIFO のリード側/受信 FIFO のライト側間の信号です。ユーザ・ロジックと送信 FIFO のライト側/受信 FIFO のリード側は(Intel 標準 FIFO ライブラリ)の一般的な FIFO インターフェイスで接続できます。

### IdenCtrl/IdenName

本 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。



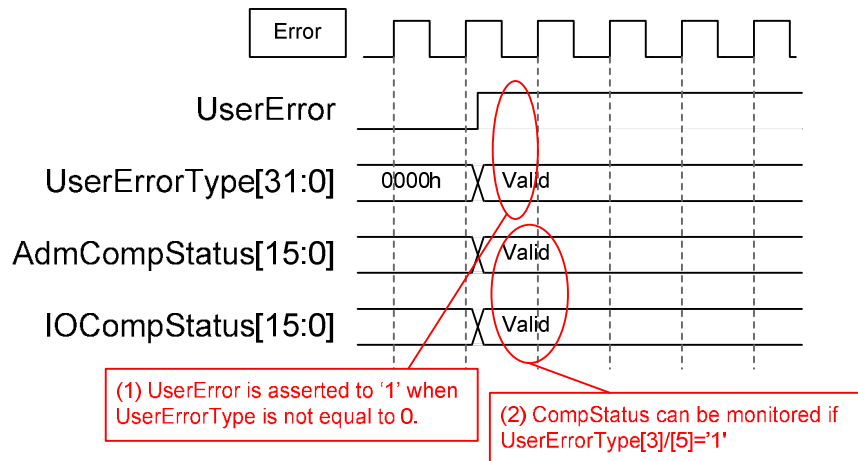
- (1) ユーザ回路は IDENTIFY コマンドの発行を IP コアに要求します
- (2) IP コアはコマンド要求を受け、IDENTIFY コマンド実行を開始します
- (3) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY コントローラ・データをユーザ回路に出力します
- (4) IP コアは NVMeSSD からの 4096 バイトの IDENTIFY ネームスペース・データをユーザ回路に出力します
- (5) NVMeSSD が完了を返送すると UserBusy がネゲートされます
- (6) UserBusy がネゲートされるとディスク容量情報となる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

**図 8: 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 8 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 の性質によ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 9 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 9: 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, Arria10SoC 開発キット, Arria10GX 開発キット等の Intel 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、Intel 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 Intel 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
NVMe-IP-A10GX	Arria10 GX	DK-DEV-10AX115S-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](mailto:sales@dgway.com) ) または国内 Intel 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 Intel 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
2.0J	2017/05/19	PCIe ハード・マクロの Avalon-ST 直結にコアを改良
2.1J	2017/06/09	Mode の廃止、誤字の修正