

# AHCIPCIeSSD IP Core (APS-IP)

データシート

2016/07/14

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](http://www.dgway.com)

## 特長

- AHCI 対応 PCIe SSD へ NCQ コマンドを介してアクセスするアプリケーション・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 PCIe Avalon-MM ハード IP コアと 128 ビット・バスで直結
- 最少のリソース消費量、CPU や外部 DDR メモリ不要
- IDENTIFY DEVICE, WRITE FPDMA QUEUED, READ FPDMA QUEUED の3コマンドをサポート
- Arria V GX スタータ・ボード+ AB16-PCIeXOVR アダプタ、ArriaV SoC 開発ボード、または、Arria10 SoC 開発ボードで実機動作するリファレンス・デザイン添付

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制約ファイルを提供
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	ArriaV GX Starter board/ ArriaV SoC Development board Arria10 SoC Development board
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: V シリーズ(PCle GEN2) QuartusII コンパイル結果

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers <sup>1</sup>	Pin <sup>2</sup>	Block Memory bit	Design Tools
ArriaV GX	5AGXFB3H4F35C4	125	593	790	-	-	QuartusII 15.1
ArriaV ST	5ASTFD5K3F40I3	125	593	793	-	-	QuartusII 15.1

表 2: 10 シリーズ(PCle GEN3) QuartusII コンパイル結果

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers <sup>1</sup>	Pin <sup>2</sup>	Block Memory bit	Design Tools
Arria10 SX	10AS066N3F40E2SGE2	250	628	923	-	-	QuartusII 16.0

注:

1) 実際のリソース消費化ウンとはユーザロジックやフィット条件などに依存します、また本リソース結果は APS-IP コア単体のみで PCIe Avalon-MM ブリッジ IP コアは含みません

2016/07/14

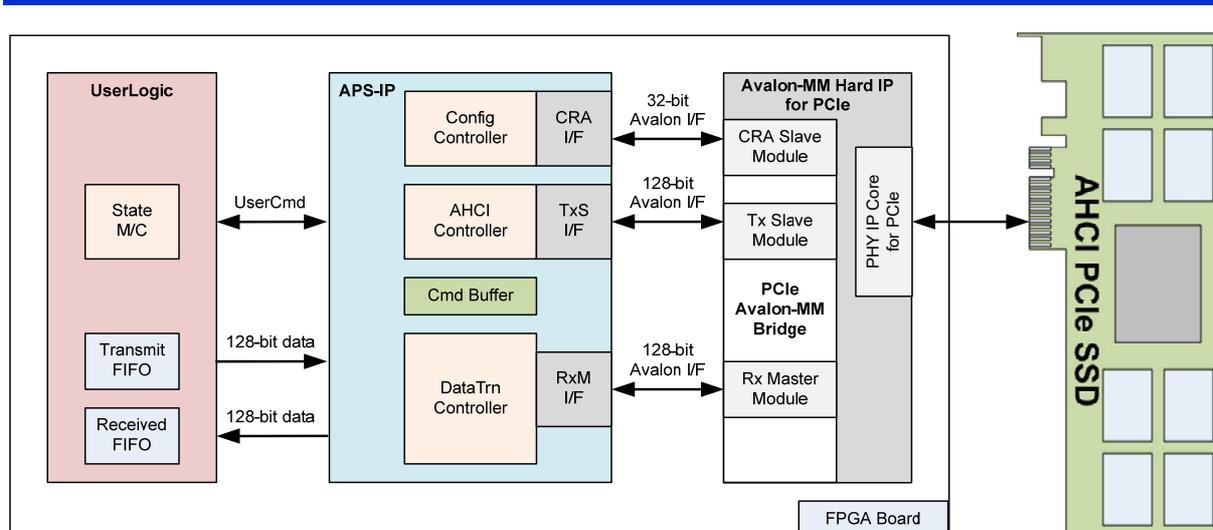


図1：APS IP ブロック図

## アプリケーション情報

APS IP は Altera 製 PCIe Avalon-MM Hard IP コアと組み合わせて動作し、NiosII 等の CPU や DDR など外部メモリなしでも 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。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 PCIe プロトコルを標準で使う M.2 フォームファクタの最新 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

APS IP コアは AHCI 規格を通して PCIe 接続 SSD をアクセスするホスト・コントローラを実装します。本 IP コアは Altera 製 PCIe Avalon-MM Hard IP コアとダイレクトに接続し PCIe SSD へパケットを送受信します。消費リソースを最小限に抑えるため、コアでは3種類の ATA コマンドのみサポートします。すなわち IDENTIFY COMMAND はドライブ容量を取得するために、WRITE FPDMA QUEUED は SSD へデータをライトするために、そして READ FPDMA QUEUED はデータをリードするためにサポートします。本 IP コアはコマンドの制御に CPU ファームウェアと異なり純ハードワイヤード・ロジックでデザインされており、さらにコマンド間切替のオーバーヘッドを抑制するキューイングに対応しているため、理論上最高のパフォーマンスが得られます。

ユーザ・インターフェイスは非常にシンプルにデザインされており、コマンド(R/W 種別)、アクセス開始アドレス、転送長をセットするだけで動作します。またリード・ライト・データもごく一般的な FIFO と直結するため、データも汎用 FIFO へのアクセスだけで実装できます。コア内部には一切の非同期ロジックがないため、コアは Avalon-MM PCIe Hard IP コアと同一のクロック・ドメインで実装する必要があります。またパケット転送中に何らかの異常をコアが検出するとエラー信号を出力します。

Arria10 SoC 開発ボードや ArriaV SoC 開発ボード、または、弊社製 AB16-PCIeXOVR アダプタを接続した ArriaV GX スタータ開発キットに評価対象の PCIe SSD を接続し、弊社 Web サイトから評価用の sof ファイルをダウンロードすることでコアの購入前に実機動作を確認できます。

## コア機能の説明

APS IP コアは AHCI 対応 PCIe SSD に対するデータのリード・ライトに必要なパケットを生成し、パケット・シーケンスを自動制御します。Avalon バスおよび Altera 製 Avalon-MM PCIe Hard IP コアと接続するため3種類のインターフェイス・モジュールをコア内部に実装します。

## コンフィグレーション

システムの電源が投入されると PCIe ルート・コンプレックスは PCIe 規格に準拠した PCIe SSD に対してコンフィグレーション・データの読み出しと設定を行う必要があります。コア内部の本回路ブロックにてコンフィグレーションを実行します。また、PCIe 割込み信号やステータスもモニタされ接続 SSD が正常な状態であることをチェックします。

- **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サであり、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 エラーと割り込みステータスをモニタします。どちらのシーケンサともレジスタ・アクセスを介して制御されます。

- **CRA I/F**

このモジュールは Avalon-MM PCIe Hard IP コアのコンフィグレーション・ライト/リード・データや PCIe SSD のコンフィグレーション・データを含む TLP パケットを生成するために使われ、また Avalon-MM PCIe Hard IP コアの内部レジスタを実装します。本 IP コアはマスタ・モードで動作し Avalon-MM PCIe Hard IP のコンフィグレーションを完了すると TLP パケットはクリアされます。

## AHCI

この回路ブロックは AHCI 規格に準拠して SSD の初期化を実行します。また、AHCI レジスタを介して SATA リンクアップ、SATA エラー、SATA 割込みをモニタします。また、コマンド発行のイネーブル・フラグやコマンド/ステータス/データを保管するメモリ・アドレスは本回路ブロックで設定されます。

- **AHCI コントローラ**

このモジュールではシステムのブートアップ時に AHCI レジスタの初期化を行います。ユーザ回路から新たなコマンドとパラメータが本コアに指示されたとき、本モジュールはユーザ回路からの指示をデコードしコマンド・バッファに FIS をセットした後に AHCI レジスタに対して動作開始セットします。その後本モジュールはコマンド完了を待つためステータス・フラグをモニタしたエラー発生の有無をチェックします。

- **TxS I/F**

AHCI レジスタは 128 ビット Avalon バスを通してアクセスされる PCIe デバイスの BAR5 空間にマップされます。ただし AHCI レジスタのアクセスは Avalon-MM PCIe Hard IP コアの TxS ポートを通して 32 ビットのシングル・アクセスのみとなります。CRA I/F と同様、本モジュールもマスタ・モードで動作します。

- **コマンド・バッファ (Cmd Buffer)**

このバッファは AHCI 規格のコマンド・リストとコマンド・テーブルを格納します。

## Data

この回路ブロックはユーザ・ロジック内の送信/受信 FIFO と Avalon-MM PCIe Hard IP コア間のデータ転送およびコマンド・バッファと Avalon-MM PCIe Hard IP コア間のコマンド/ステータス転送を制御します。

- **データ転送コントローラ(DataTrn Controller)**

このモジュールは要求されたアドレス情報をデコードしメモリの送受信先を選択します。このモジュールの接続先は3種類のメモリ・インターフェイスがあり、それはデータ転送用のユーザ FIFO、コマンド・テーブルとコマンド・リスト用のコマンド・バッファ、そして IDENTIFY データです。

- **RxM I/F**

このモジュールは 128 ビット Avalon バスのスレーブ側です。128 ビットのバウンダリをまたいだアクセスができないため本モジュールは 128 ビット以外のバス・サイズではシングル・アクセスのみサポートします。各トランザクションのバースト長はマスタ・モードの Avalon-MM PCIe Hard IP コア側で決められます。

## 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。データ送受信用の FIFO サイズはユーザ・システムのパフォーマンスとリソース消費量の要件に応じてフレキシブルに設定することができます。

## Avalon-MM PCIe Hard IP コア

Avalon-MM バスを介した PCIe Hard IP コアのインターフェイス・モジュールです。より詳細については“Arria V Avalon-MM Interface for PCIe Solutions”または“Arria10 Avalon-MM Interface for PCIe Solutions”のドキュメントを参照してください。

## 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
<b>ユーザ・インターフェイス</b>		
RstB	In	アクティブ Low のリセット信号、Clk 信号が安定となったら解除すること
Clk	In	Avalon-MM PCIe Hard IP コアからのクロック出力を接続し本コア全体を Avalon バスと同期するために使う、PCIe Gen2 の場合 125MHz で PCIe Gen3 の場合 250MHz
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	SSD のコマンド完了までのタイムアウト設定時間、クロック周期単位での設定 (PCIe Gen2 の場合 8ns で PCIe Gen3 の場合 4ns 単位)
UserCmd[1:0]	In	ユーザ・コマンド種別、'00': IDENTIFY DEVICE、'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] - SSD のリセット処理が設定したタイムアウト時間を経過したが完了しない [1] - 接続 SSD が AHCI プロトコルに対応していない [2] - AHCI 準拠の全 32 チャンネルの PHY LINKUP が検出されない [3] - シグネチャ FIS が ATA ドライブではないエラー [4] - 512 バイトの IDENTIFY DEVICE データがタイムアウトの 1 秒以内に受信できない [5] - Portx 割込みステータス・レジスタエラーを検出した (Portx 割込み詳細については PortIntStatus 信号を参照のこと) [6] - SSD からの MSI 割込みが設定したタイムアウト時間を経過しても受信されない [7] - PCIe SSD の CI レジスタが'0'クリアされず設定したタイムアウト時間以内にコマンドが完了しない [31:8] - 未使用 注意: bit[0]/[4]/[6]/[7]のタイムアウト期間は TimeOutSet[31:0]で設定する
PortIntStatus[31:0]	Out	Port 割込みステータス・レジスタ(AHCI レジスタの 0x10)の最新の読み出し値を示す UserErrorType[5]='1'の場合本レジスタを参照する より詳細については“Serial ATA AHCI Specification”を参照のこと

信号名	方向	説明
<b>FIFO インターフェイス</b>		
UserFifoWrRdy	In	リード用 FIFO の空き容量が少なくとも 512 バイト以上ある場合に '1' アサートシコアに通知
UserFifoWrEn	Out	リード用 FIFO にリードしたデータと同期してイネーブルすることで FIFO へ書き込む
UserFifoWrData[127:0]	Out	リード用 FIFO に出力する 128 ビット・リード・データ、UserFifoWrEn と同期して出力
UserFifoEmpty	In	ライト用 FIFO が空である場合に '1' アサートシコアに通知
UserFifoRdEn	Out	ライト用 FIFO からコアへのデータ出力要求
UserFifoRdData[127:0]	In	ライト用 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
<b>その他のインターフェイス</b>		
IdRamWrEn	Out	IdRamWrData と同期して出力するライト・イネーブル信号
IdRamWrData[127:0]	Out	IDENTIFY DEVICE コマンドで受信する 512 バイトの IDENTIFY データ、IdRamWrEn に同期。
PCIELinkup	In	Avalon-MM PCIe Hard IP コアからの割込み出力。
<b>CRA インターフェイス</b>		
CraChipSel	Out	CRA ポートを選択するためアサートする
CraAddress[13:0]	Out	ライト・リードのバイト・アドレス、32 ビット・アクセスのため Bit[1:0]は常に '00' となる
CraByteEnable[3:0]	Out	バイト・イネーブル
CraRead	Out	リード・イネーブル、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	ウェイトの延長要求
<b>TxS インターフェイス</b>		
TxSChipselect	Out	送信スレーブ・ポートを選択するためアサートする
TxSByteEnable[15:0]	Out	ライト・データ用のバイト・イネーブル
TxSReadData[127:0]	In	PCIe Hard 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	データ要求数を示す、シングル・アクセスでは常に '000001' b となる
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、あるいはライトの延長要求
TxSAddress[28:0]	Out	ライト/リード・アドレス

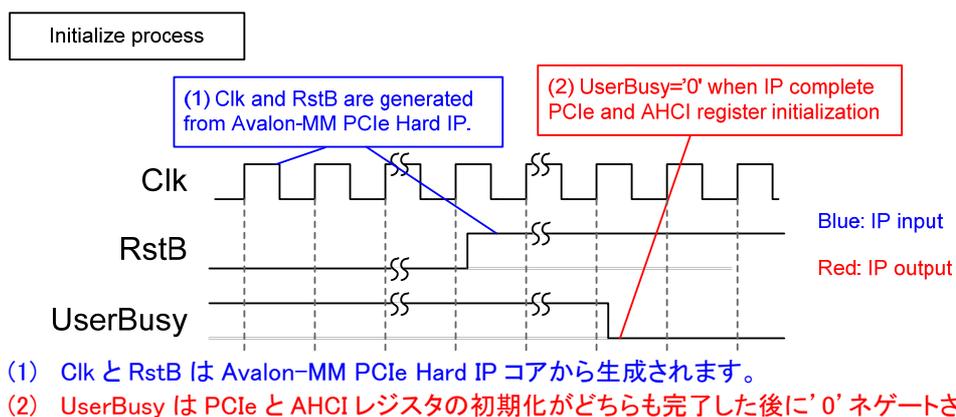
信号名	方向	説明
<b>RxM インターフェイス</b>		
RxMAddress[31:0]	In	アクセス先のライト/リード・アドレス
RxMBurstCount[7:0]	In	ライト/リード要求のバースト・カウント値をセット
RxMByteEnable[15:0]	In	ライト・データにおけるバイト・イネーブル
RxMWrite	In	PCIe Hard IP コアからのライト要求
RxMRead	In	PCIe Hard IP コアからのリード要求
RxMWaitRequest	Out	データ転送のウェイト要求
RxMWriteData[127:0]	In	ライト・データ
RxMReadData[127:0]	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
RxMAddress[31:0]	In	ライト/リード・アドレス、ライト/リード・バースト・トランザクションの転送先頭におけるアドレス

注意:

- 1) スレーブ RxM インターフェイスにおいて、この IP コアは 128 ビットにアライメントが揃っていないバースト転送 (AvBurstCount が 1 でない転送) はサポートしていません。
- 2) スレーブ RxM インターフェイスにおいて、この IP コアは 128 ビットのバウンダリをまたぐ転送はサポートしていません、従ってバースト転送の場合 AvAddress[3:0]は常に '0000' b とする必要があります。

### タイミング波形

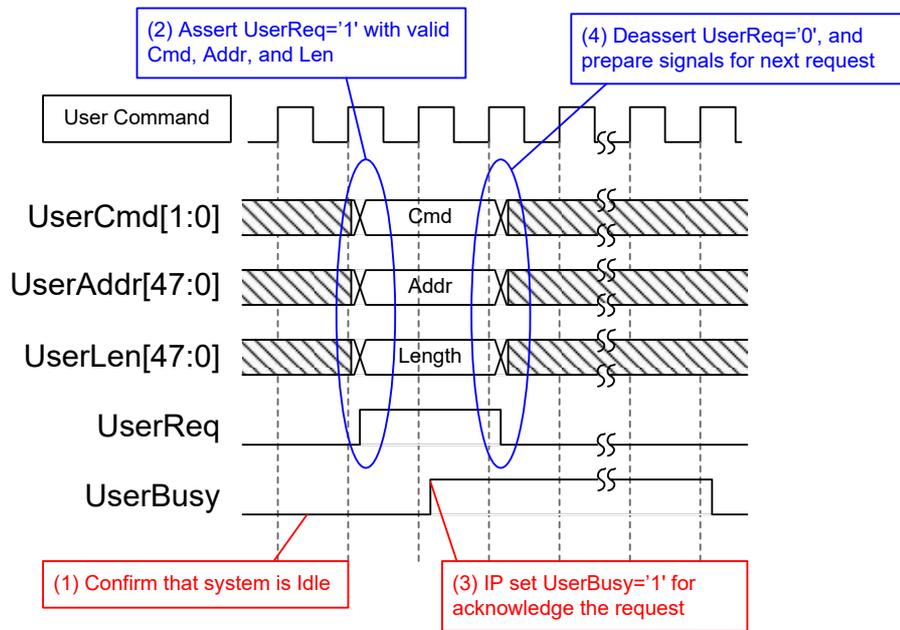
IP コアの同期クロック信号(Clk 信号)およびリセット信号(RstB)は Avalon-MM PCIe Hard IP コアから作られます。RstB がリリースされると IP コアは PCIe コンフィグレーション・レジスタと PCIe SSD の AHCI レジスタを初期化します。両方の初期化シーケンスが完了すると図 2 に示すように UserBusy が '0' にネゲートされます。



- (1) Clk と RstB は Avalon-MM PCIe Hard IP コアから生成されます。
- (2) UserBusy は PCIe と AHCI レジスタの初期化がどちらも完了した後に '0' にネゲートされます

図 2: システム起動時の RstB と UserBusy のネゲート・タイミング

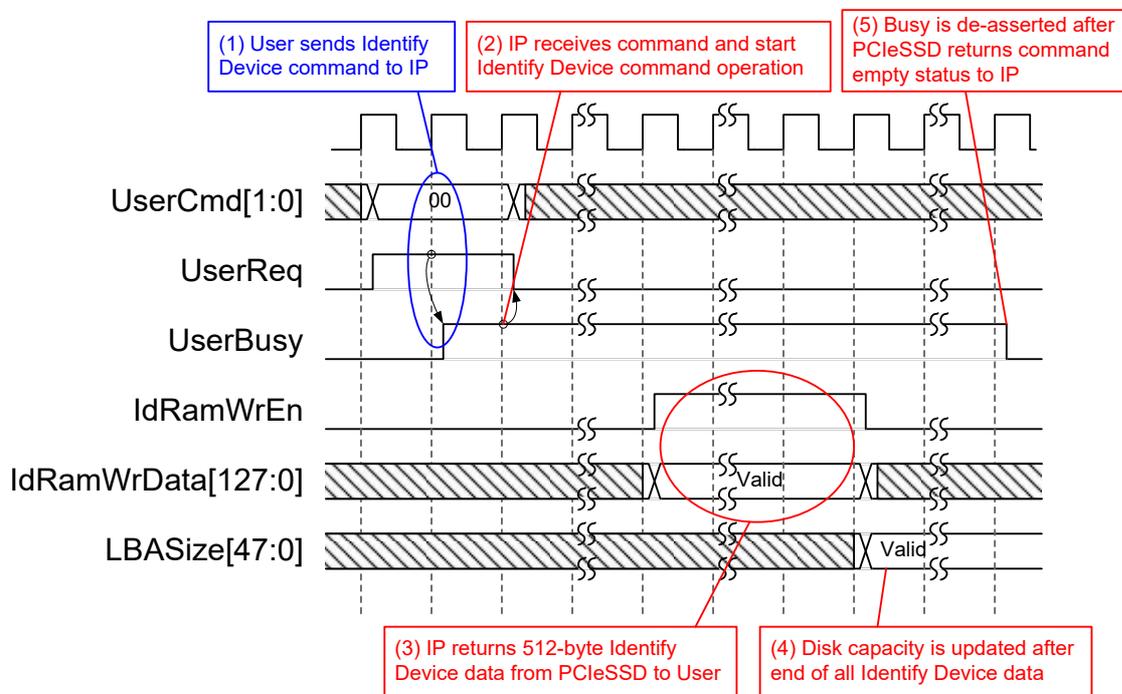
IP コアに新たなライト/リード命令を送る前に必ず UserBusy をモニタし IP コアがアイドル状態であることを確認する必要があります。UserCmd, UserAddr,そして UserLen は UserReq='1' でコマンドを出力する期間は図 3 に示すように必ず有効な値を保持しなくてはなりません。IP コアは UserBusy='1' としてコマンドを受信したことを示しコマンドの実行を開始します。UserBusy='1' となった以降は UserReq はクリアでき、ユーザ回路からは次の新たなコマンド用のパラメータを用意することができます。



- (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 3: ユーザ・コマンドのタイミング波形

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 DEVICE 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。

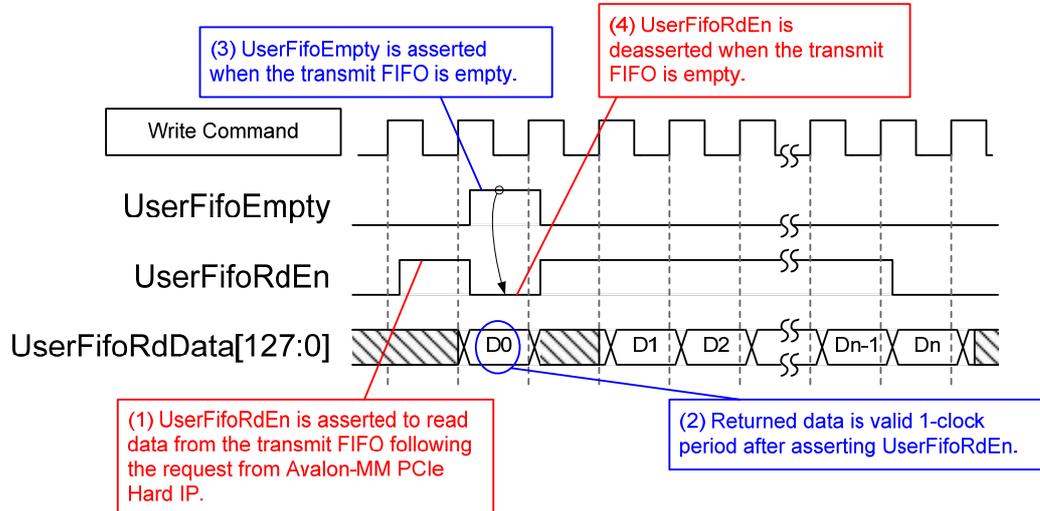


- (1) ユーザ回路は IDENTIFY DEVICE コマンドの発行を IP コアに要求します
- (2) IP コアはコマンド要求を受け、IDENTIFY DEVICE コマンド実行を開始します
- (3) IP コアは PCIeSSD からの 512 バイトの IDENTIFY DEVICE データをユーザ回路に出力します
- (4) 全 IDENTIFY DEVICE データ受信が完了するとディスク容量情報となる LBASize が更新されます
- (5) PCIeSSD がコマンド完了ステータスを IP コアに返すと UserBusy が '0' にネゲートされます

図 4: IDENTIFY DEVICE コマンド後に LBASize が更新される

図 4 のように UserBusy='0' となっている時に UserCmd と UserReq がセットできます。ただこの IDENTIFY DEVICE コマンドでは UserAddr と UserLen 入力は不要です。その後 512 バイトの IDENTIFY DEVICE データが IdRam を通じて出力され、さらに LBASize 出力が有効になります。PCIe SSD がコマンド動作の最後に有効なステータス情報を戻すと IP コアからの UserBusy が '0' にネゲートされます。

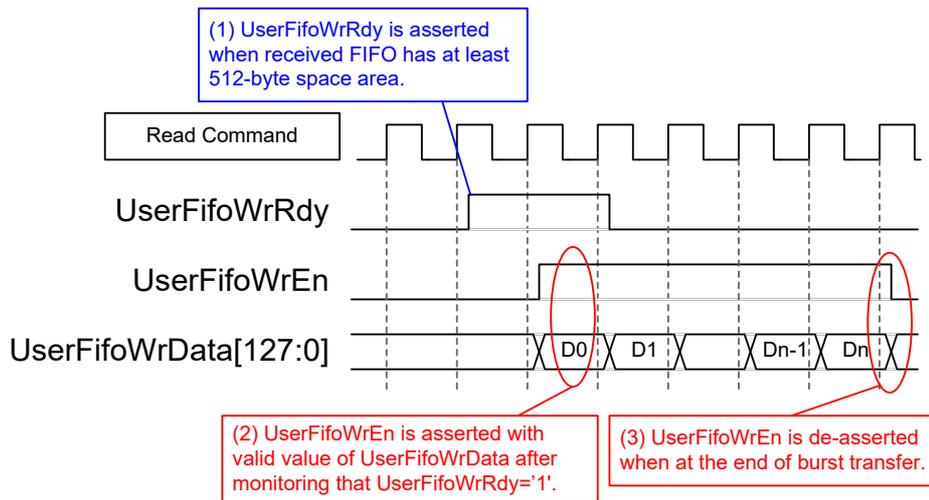
ライト・コマンドにおいて送信 FIFO からのライト・データは RxM インターフェイスを介して Avalon-MM PCIe Hard IP コアへ送信されます。送信 FIFO が空でないことが UserFifoEmpty 信号により確認されると UserFifoRdEn がアサートされます。一般的な FIFO と同様、UserFifoRdData は図 5 に示すように UserFifoRdEn がアサートされた次のクロック・ピリオドで有効な値となります。



- (1) Avalon-MM PCIe Hard IP コアからの要求により UserFifoRdEn がアサートされ送信 FIFO からデータが出力されます
- (2) FIFO からのデータは UserFifoRdEn の次クロック・ピリオドで出力されます
- (3) 送信 FIFO が空の場合 UserFifoEmpty がアサートされます
- (4) 送信 FIFO が空になると UserFifoRdEn はネゲートします

図 5: ライト・コマンドにおける送信 FIFO のタイミング波形

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、Avalon-MM PCIe Hard IP コアから要求されたサイズのバースト転送数と一致するまでリード・データが受信 FIFO へと転送されます。UserFifoWrRdy は受信 FIFO に少なくとも 512 バイトの空き容量があることを確認するために使われます。従ってバースト・サイズが 512 バイト以上であった場合、IP コアは 512 バイト単位に分割して転送を複数回実行します。

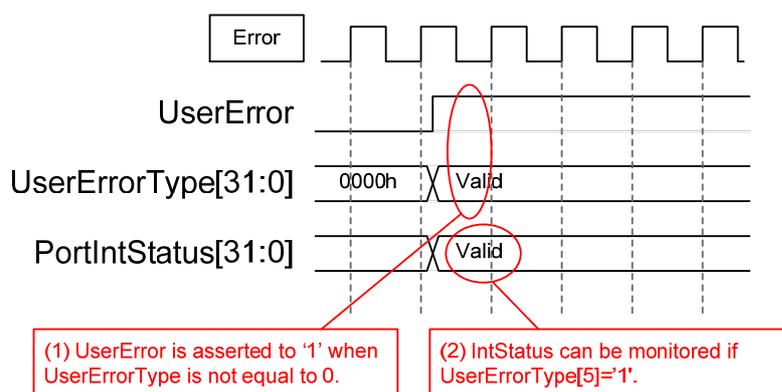


- (1) 受信 FIFO に 512 バイト以上の空き領域があると UserFifoWrRdy が '1' アサートされます
- (2) UserFifoWrRdy='1' を確認した次クロックから UserFifoWrEn と UserFifoWrData が出力されます
- (3) UserFifoWrEn はバースト転送の最後に到達するとネゲートします

図 6: リード・コマンドにおける受信 FIFO のタイミング波形

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 7 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。

PortIntStatus の値がエラー状況である場合、UserErrorType のビット[5]がセットされます。このときユーザ回路側では PortIntStatus 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[5]='1' の場合、PortIntStatus で詳細情報が確認できます

**図 7: エラー・フラグのタイミング波形**

## コアの検証方法

APS-IP コアは Arria10 SoC 開発ボードや ArriaV SoC 開発ボードや AB16-PCIeXOVR アダプタを装着した ArriaV GX スタータ開発キットで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 Recommended Design Experience

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 APS-IP は以下の Xilinx 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
APS-IP-A5GX	ArriaV GX	ArriaV GX スタータ開発キット (DK-START-5AGXB3N)
APS-IP-A5ST	ArriaV ST (SoC)	ArriaV SoC 開発キット (DK-DEV-5ASTD5N)
APS-IP-A10SX	Arria10 SX (SoC)	Arria10 SoC 開発キット (DK-SOC-10AS066S-A)

## 履歴

リビジョン	日付	更新内容
1.0J	2016/03/03	日本語版の初期版リリース
1.1J	2016/07/14	Arria10 SX のサポート開始