

## AHCIPCIeSSD IP Core (APS-IP) データシート

2016/03/03

Product Specification

Rev1.2J



### 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
- Xilinx 製 AXI PCIe ブリッジと 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
- PCIe GEN3 対応 (Kintex Ultrascale 搭載の KCU105 で PCIe GEN3 動作検証済み)
- AB16-PCIeXOVR アダプタと KC705/VC707/ZC706/KCU105 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 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	KC705, VC707, ZC706, KCU105
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) コンパイル結果

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices <sup>1</sup>	IOB	BRAMTile	PLL	GTX	Design Tools
Kintex-7	XC7K325TFFG900-2	125	568	1328	478	-	-	-	-	Vivado2014.4
Virtex-7	XC7VX485TFFG1761-2	125	568	1329	503	-	-	-	-	Vivado2014.4
Zynq-7000	XC7Z045FFG900-2	125	568	1359	498	-	-	-	-	Vivado2014.4

表 2: Ultrascale (PCIe GEN3) コンパイル結果

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile	PLL	GTH	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	250	568	1459	272	-	-	-	-	Vivado2015.4

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

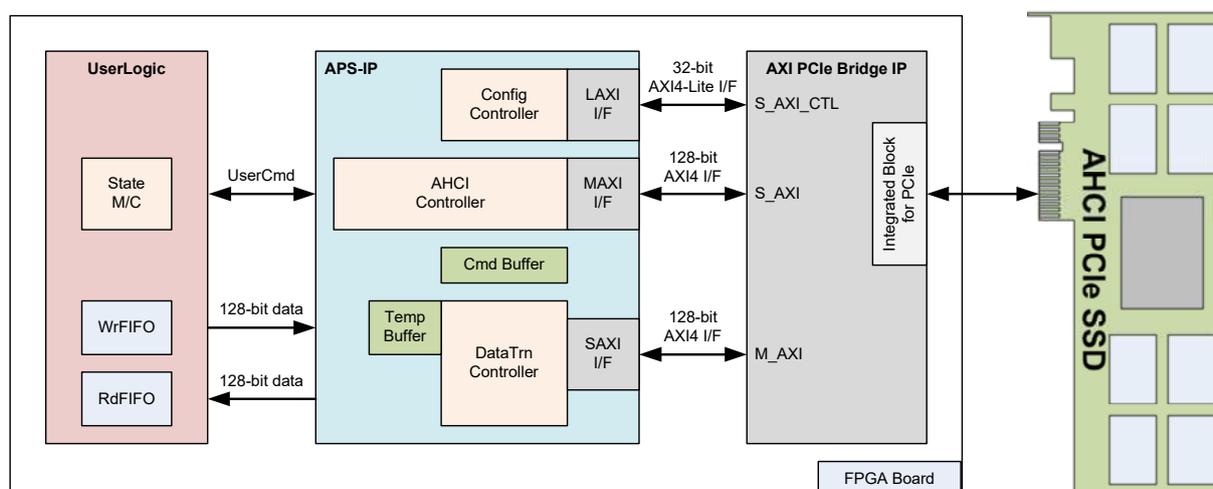


図1：APS IP ブロック図

## アプリケーション情報

APS IP 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 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 Xilinx 製 AXI PCIe ブリッジ・コアと接続し 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 ヘ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 AXI PCIe ブリッジ・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。

Xilinx 標準の KC705/ZC706/VC707/KCU105 ボードに弊社製 AB16-PCIeXOVR アダプタおよび評価対象の PCIe SSD を接続し、弊社 Web サイトから評価用の bit 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。AXI バスおよび Xilinx 製 AXI PCIe ブリッジ・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。

- **LAXI I/F**

コンフィグレーション・レジスタは 32 ビット AXI-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。

## 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。

- **MAXI I/F**

AHCI レジスタは 128 ビット AXI 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 AXI PCIe ブリッジ IP の S\_AXI を通して 32 ビットのシングル・アクセスのみとなります。LAXI 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 FIFO と AXI PCIe ブリッジ IP 間のデータ転送およびコマンド・バッファと AXI PCIe ブリッジ間のコマンド/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。

- **SAXI I/F**

このモジュールは 128 ビット AXI4 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 AXI PCIe ブリッジ IP 側で決められます。

- **一時バッファ(Temp Buffer)**

小規模な RAM で構成されていて SAXI4 I/F のデータ・フローを制御するためユーザからのライト・データを一時的に格納します。

## 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。

## AXI PCIe ブリッジ IP

AXI4 バスと PCIe 間は Xilinx の IP コアで接続します。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。

## 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
ユーザ・インターフェイス		
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	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
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)
CmdQEn	In	コマンド・キューのイネーブル・ビット、'0': キュー深さ=1 固定、'1': キュー深さは 1 以上。通常は '1' をセットする。接続 SSD で UserErrorType[4] が 1 となった場合は本ビットを 0 にセットしコマンド・キューを禁止する必要がある。
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。
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).
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時間内に受信できない [5] - Portx 割り込みステータス・レジスタエラーを検出した (Portx 割り込み詳細については PortIntStatus 信号を参照のこと) [6] - SSD からの MSI 割り込みがタイムアウト時間を経過しても受信されない [7] - コマンド完了を示す PCIe SSD の CI レジスタの '0' クリアが、タイムアウト時間を経過しても検出されない [8] - PCIe 割り込みステータスがエラー状態であることを示す (割り込みデコード・レジスタを参照し PCIeIntStatus 信号を確認) [31:10] - 未使用 注意: bit[0]/[4]/[6]/[7]は TimeOutSet で設定したタイムアウト経過によりセットされる

信号名	方向	説明
<b>ユーザ・インターフェイス (続き)</b>		
PCIEIntStatus[31:0]	Out	AXI PCIe ブリッジ IP 内の割込デコード・レジスタ(0x138)の最新の読み出し値を示す。 UserErrorType[8]='1' の場合本レジスタを参照する 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
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
UserFifoRdRdy	In	ライト用 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
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.
PCIEInt	In	AXI PCIe ブリッジ IP からの割込み出力.
<b>マスタ AXI4-Lite インターフェイス</b>		
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
LXiwData[31:0]	In	リード・データ
LXiwValid	In	リード有効、有効なリード・データが LXiwData バス上に現れていることを示す
LXiwReady	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

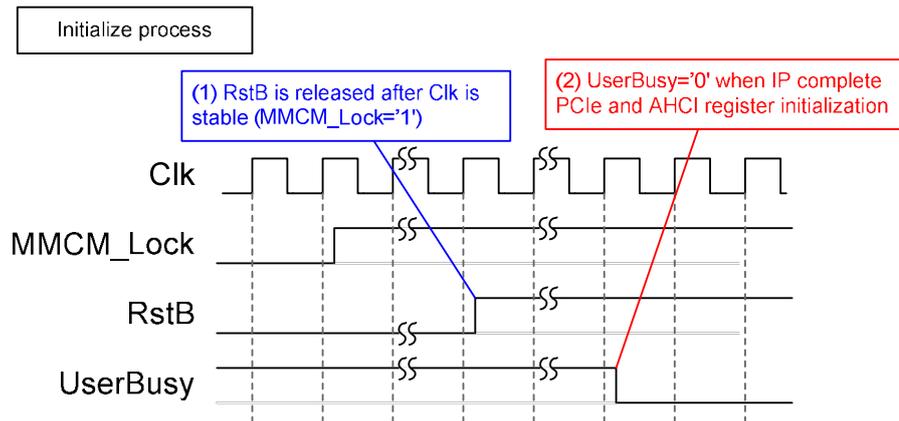
信号名	方向	説明
スレーブ AXI4 バス・インターフェイス		
SXAwAddr[31:0]	In	ライト・アドレス、ライト・バースト・トランザクションで最初の転送のアドレスが入力される
SXAwLen[7:0]	In	バースト長、1バーストでの転送長が入力される。
SXAwValid	In	ライト・アドレス有効、入力されるライト・アドレスと制御信号が有効であることを示す
SXAwReady	Out	ライト・アドレス・レディ、スレーブ側はライト・アドレスと制御信号を受信できることを示す
SXAwData[127:0]	In	ライト・データ
SXAwStrb[15:0]	In	ライト・ストロブ、有効なデータを持つバイト・レーン情報を示す
SXAwLast	In	最終ライト、ライト・バーストでの最終転送を示す
SXAwValid	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
SXAwReady	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
SXAwResp[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
SXAwValid	Out	ライト応答有効、ライト応用が有効であることを示す
SXAwReady	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
SXArData[127:0]	Out	リード・データ
SXArResp[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
SXArLast	Out	最終リード、リード・バーストでの最終転送を示す
SXArValid	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
SXArReady	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

注意:

- 1) スレーブ AXI4 インターフェイスにおいて、IP コアは(AwLen/ArLen がゼロではない)ナロー・トランザクションのバースト転送はサポートしていません
- 2) スレーブ AXI4 インターフェイスにおいて、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ないため、AwAddr/ArAddr[3:0]はバースト転送において必ず'0000'とする必要があります。

## タイミング波形

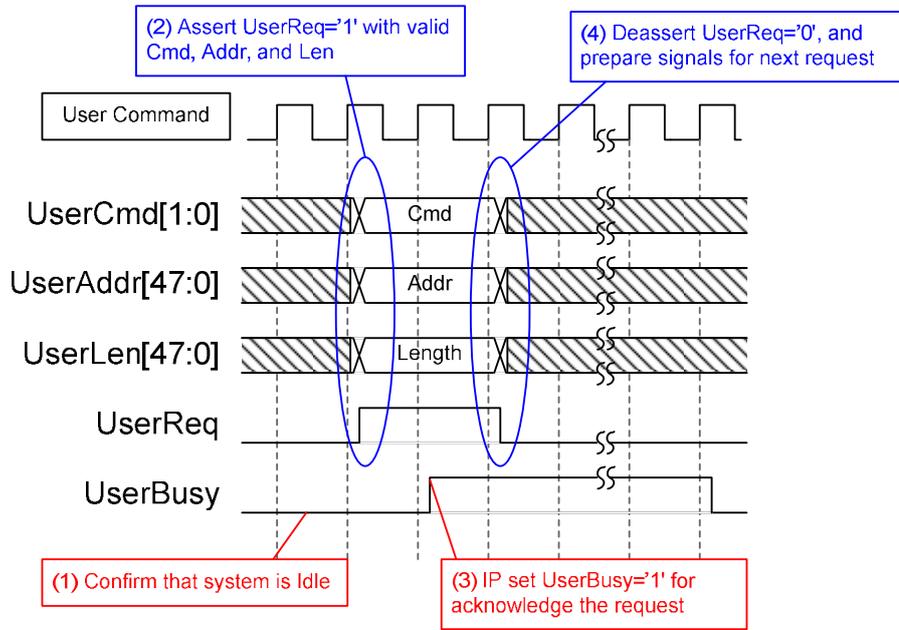
IP コアの同期クロック信号(Clk 入力信号)は AXI PCIe ブリッジ IP 内の MMCM から作られます、このため下図 2 に示すように、AXI PCIe ブリッジ IP からの MMCM\_Lock 出力信号をモニタし、クロックが安定したことを確認した後にコアのリセット信号となる RstB をリリースする必要があります。その後 IP コアは PCIe コンフィグレーション・レジスタと PCIeSSD 内の AHCI レジスタを初期化します。両方の初期化シーケンスが完了した後に UserBusy は '0' にネゲートされます。



- (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 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**

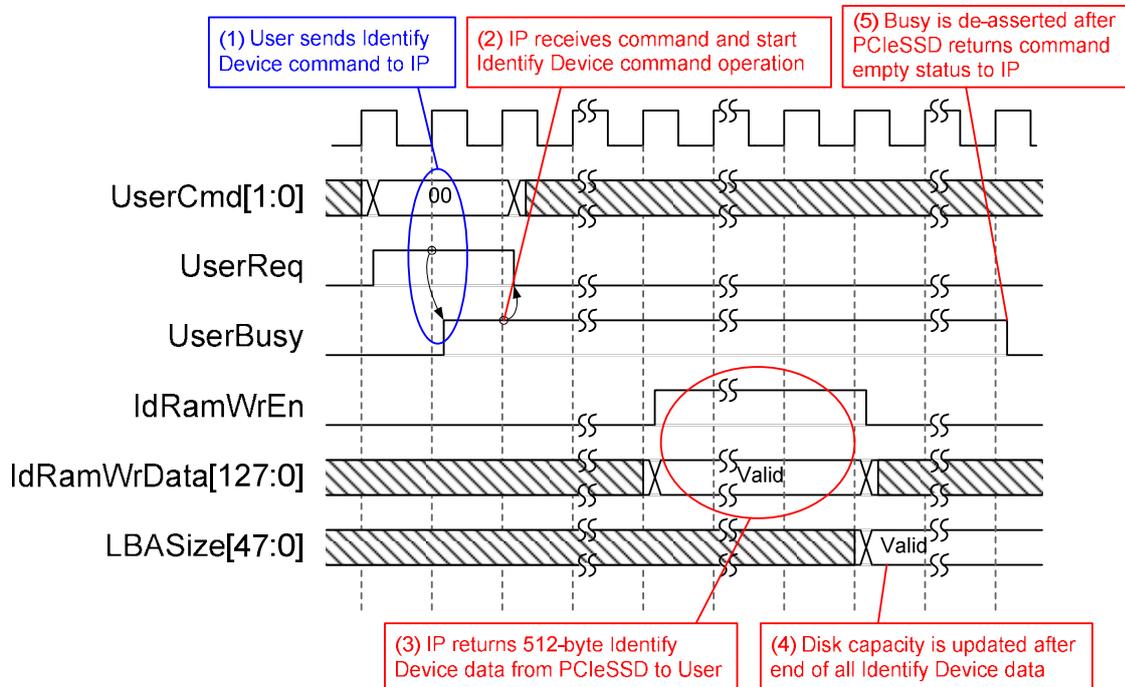
IP コアに新たなライト/リード命令を送る前に必ず UserBusy をモニタし IP コアがアイドル状態であることを確認する必要があります。UserCmd, UserAddr,そして UserLen は UserReq='1' でコマンドを出力する期間は図 3 に示すように必ず有効な値を保持しなくてはなりません。IP コアは UserBusy='1' としてコマンドを受信したことを示しコマンドの実行を開始します。UserBusy='1' となった以降は UserReq はクリアでき、ユーザ回路からは次の新たなコマンド用のパラメータを用意することができます。



- (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。

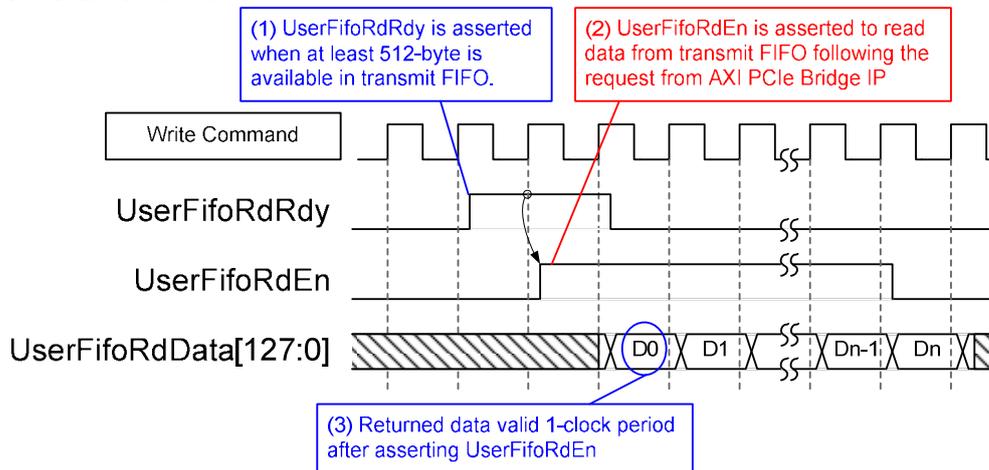


- (1) ユーザ回路は IDENTIFY DEVICE コマンドの発行を IP コアに要求します
- (2) IP コアはコマンド要求を受け、IDENTIFY DEVICE コマンド実行を開始します
- (3) IP コアは PCIeSSD からの 512 バイトの IDENTIFY DEVICE データをユーザ回路に出力します
- (4) 全 IDENTIFY DEVICE データ受信が完了するとディスク容量情報となる LBASize が更新されます
- (5) PCIeSSD がコマンド完了ステータスを IP コアに戻すと UserBusy が '0' にネゲートされます

**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。

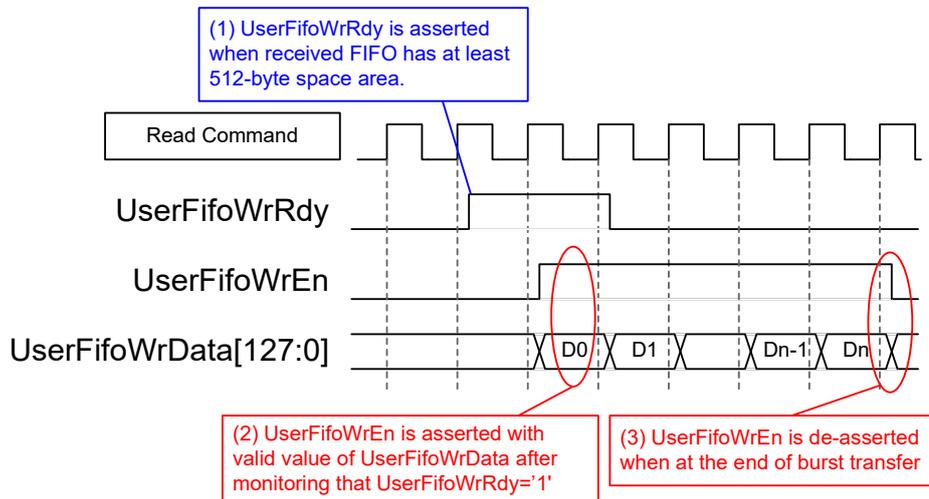
ライト・コマンドにおいて送信 FIFO からのライト・データはスレーブ AXI4 バスを介して AXI PCIe ブリッジ IP へ送信されます。ここで AXI PCIe ブリッジ IP からのバースト・サイズが 512 バイト以上であった場合、IP コアはトランザクションを 512 バイト単位の転送に自動的に分割します。このときバースト転送を開始する前に UserFifoRdRdy をモニタし、少なくとも 512 バイトの空き容量が送信 FIFO にあることを確認します。一般的な FIFO と同様、UserFifoRdData は図 5 に示すように UserFifoRdEn がアサートされた次のクロック・ピリオドで有効な値となります。



- (1) 送信 FIFO に 512 バイト以上の空き領域があると UserFifoRdRdy が '1' アサートされます
- (2) AXI PCIe ブリッジ IP 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
- (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

**図 5: ライト・コマンドにおける送信 FIFO のタイミング波形**

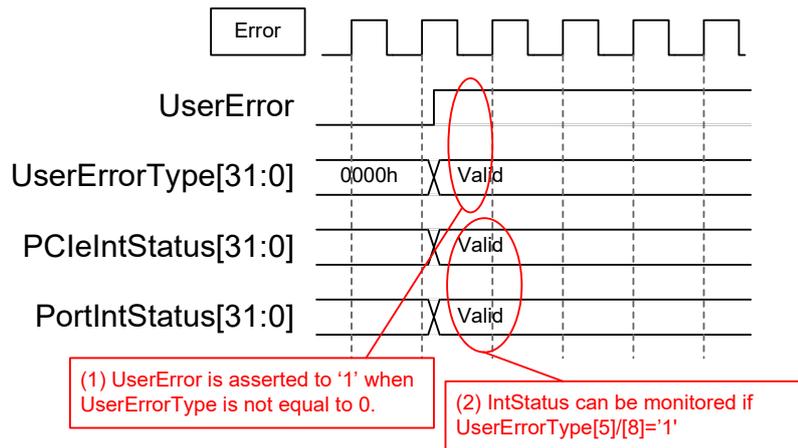
リード・コマンドにおいて UserFifoWrEn は UserFifoWrData 上の有効なリード・データと合わせてアサートされ、AXI PCIe ブリッジ IP から要求されたサイズのバースト転送数と一致するまでリード・データが受信 FIFO へと転送されます。UserFifoWrRdy は受信 FIFO に少なくとも 512 バイトの空き容量があることを確認するために使われます。ライト・コマンドと同様、バースト・サイズが 512 バイト以上であった場合、IP コアは 512 バイト単位に分割して転送を実行します。



- (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。PCIeIntStatus または PortIntStatus の値がエラー状況である場合、UserErrorType のビット[5]/[8]がセットされます。このときユーザ回路側では PCIeIntStatus または 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]/[8]='1' の場合、PortIntStatus/PCIeIntStatus 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 KC705/ZC706/VC707/KCU105 等の Xilinx 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、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 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-KT7	Kintex-7	KC-705
APS-IP-ZQ7	Zynq-7000	ZC-706
APS-IP-VT7	Virtex-7	VC-707
APS-IP-KU	Kintex Ultrascale	KCU-105

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](mailto: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
1.0J	2015/10/20	日本語版の初期版リリース
1.1J	2016/2/23	Kintex Ultrascale 向けの PCIe GEN3 新規対応、それに伴う TimeOutSet ポート追加
1.11J	2016/02/24	UserErrorType[31:0]のビット位置の誤りを修正
1.2J	2016/03/03	LinkSpeed 信号を追加