



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
- ユーザ・クロック周波数は PCIe クロック(Gen2=125MHz, Gen3=250MHz)と同じかそれ以上で動作
- FPGA 内蔵の PCIe 統合ブロックと 128 ビット・バスで直結
- BRAM で構築する 256K バイトのデータ・バッファをコアに内蔵
- IDENTIFY, WRITE, READ, SMART, Flush, Shutdown, の 6 コマンドをサポート
- さらにオプションでカスタム Admin コマンドやカスタム NVMe コマンドの実装が可能
- 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)
 - LBA ユニットの他には 512 バイトの他に 4096 バイト(4K セクタ)をサポート
- 実機動作するリファレンス・デザイン
 - KC705, VC707, VC709, ZC706, AC701, KCU105, ZCU106 は AB16-PCIeXOVR アダプタを使用
 - VCU118 は AB17-M2FMC アダプタを使用
 - Zynq Mini-ITX はアダプタ不要で動作
- 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	KC705, VC707, VC709, ZC706, AC701, Zynq Mini-ITX, KCU105, ZCU106, VCU118
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	300	4549	3204	1591	66	Vivado2017.4
Virtex-7	XC7VX485TFFG1761-2	300	4549	3203	1638	66	Vivado2017.4
Zynq-7000	XC7Z045FFG900-2	300	4549	3201	1532	66	Vivado2017.4
Kintex-7	XC7K325TFFG900-2	300	4549	3204	1574	66	Vivado2017.4
Artix-7	XC7A200TFBG672-2	225	4549	3089	1500	66	Vivado2017.4

表 2: UltraScale/UltraScale+シリーズ(PCle 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	400	4495	2951	842	66	Vivado2017.4
Zynq-Ultrascale+	XCZU7EV-FFVC1156-2E	400	4495	2955	934	66	Vivado2017.4
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L	400	4495	2954	859	66	Vivado2017.4

注:

- 1) 本コンパイル結果はコア単体でのリソース情報です
- 2) 実際のリソー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

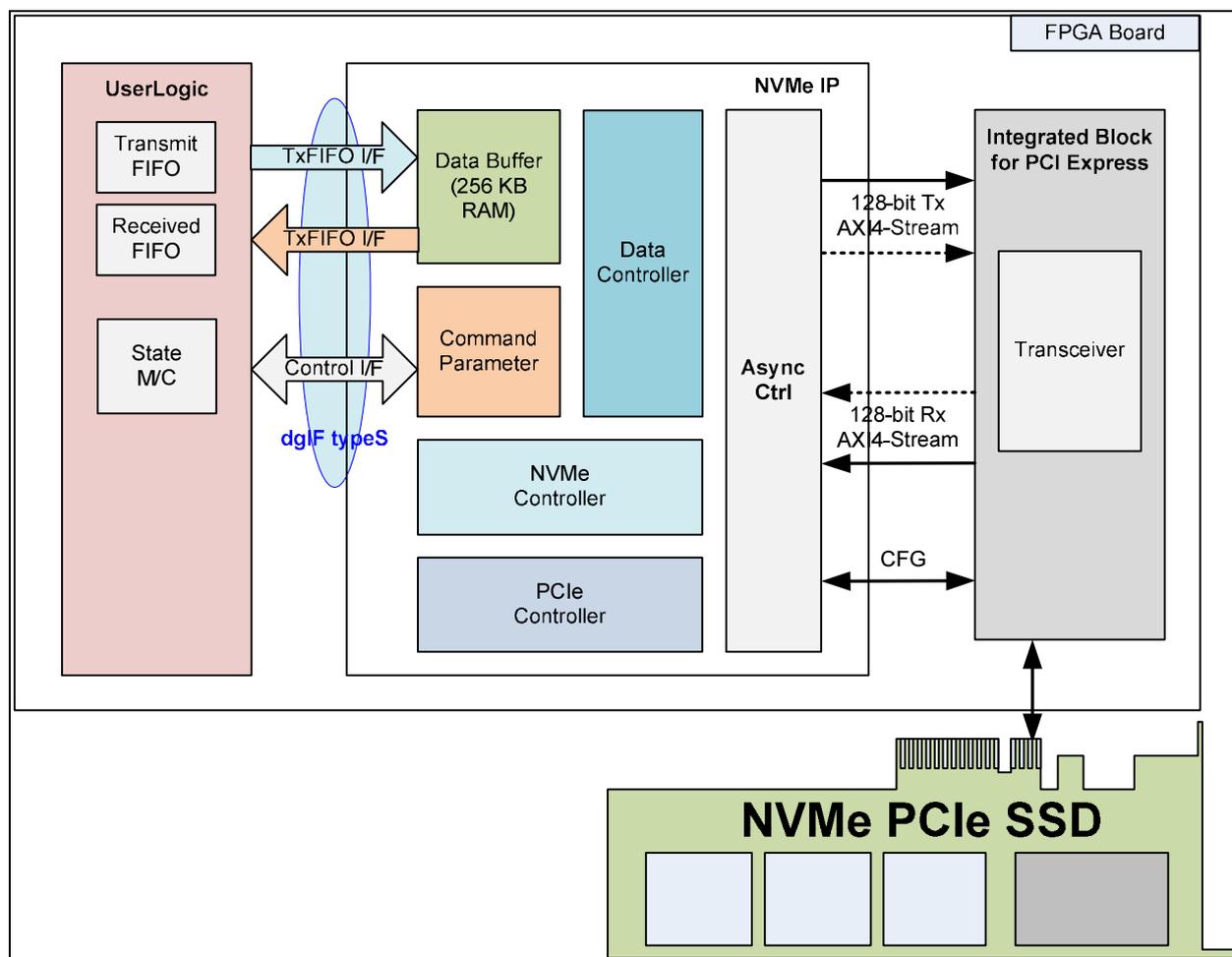


図 1: NVMe-IP コアのブロック図

アプリケーション情報

NVMe-IP コアは Xilinx 製 FPGA 内蔵 PCIe 統合ブロックと組み合わせて動作し、NVMe 規格 PCIe SSD へのリード/ライト・アクセスを MicroBlaze 等の CPU の介在なしかつ外部 DDR メモリ不要で可能とするソリューションを提供します。256K バイトの BRAM メモリで構築したデータ・バッファを IP コアに内蔵し、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時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 NVMe 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、本 IP コアより下位層のハードウェアは Xilinx 製 FPGA 内蔵の PCIe 統合部ブロックを使います。本 IP コアは Identify, ライト, リード, SMART, Flush, Shutdown の 6 種類の NVMe コマンドをサポートします。

本 IP コアはライト/リード・コマンド実行時に NVMe SSD とユーザ回路間のデータ・バッファとして使う 256K バイト RAM を内蔵します。このバッファによりシステムは SSD へのライトやリードを高いパフォーマンスが維持できます。

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。

SMART および Flush コマンドをサポートするためコマンド・パラメータの受信にカスタム・コマンドのインターフェイスが追加されています。また、SMART データはカスタム・コマンド RAM を通して転送されます。

PCIe 統合ブロックの制約からユーザ回路のクロック周波数は PCIe クロック周波数(Gen3 の場合 250MHz で Gen2 の場合 125MHz)と同じかそれ以上とする必要があります。本 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、本 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することで実機評価が可能です。

コア機能の説明

図 2 にリセット解除後の本 IP コアの動作シーケンスを示します。

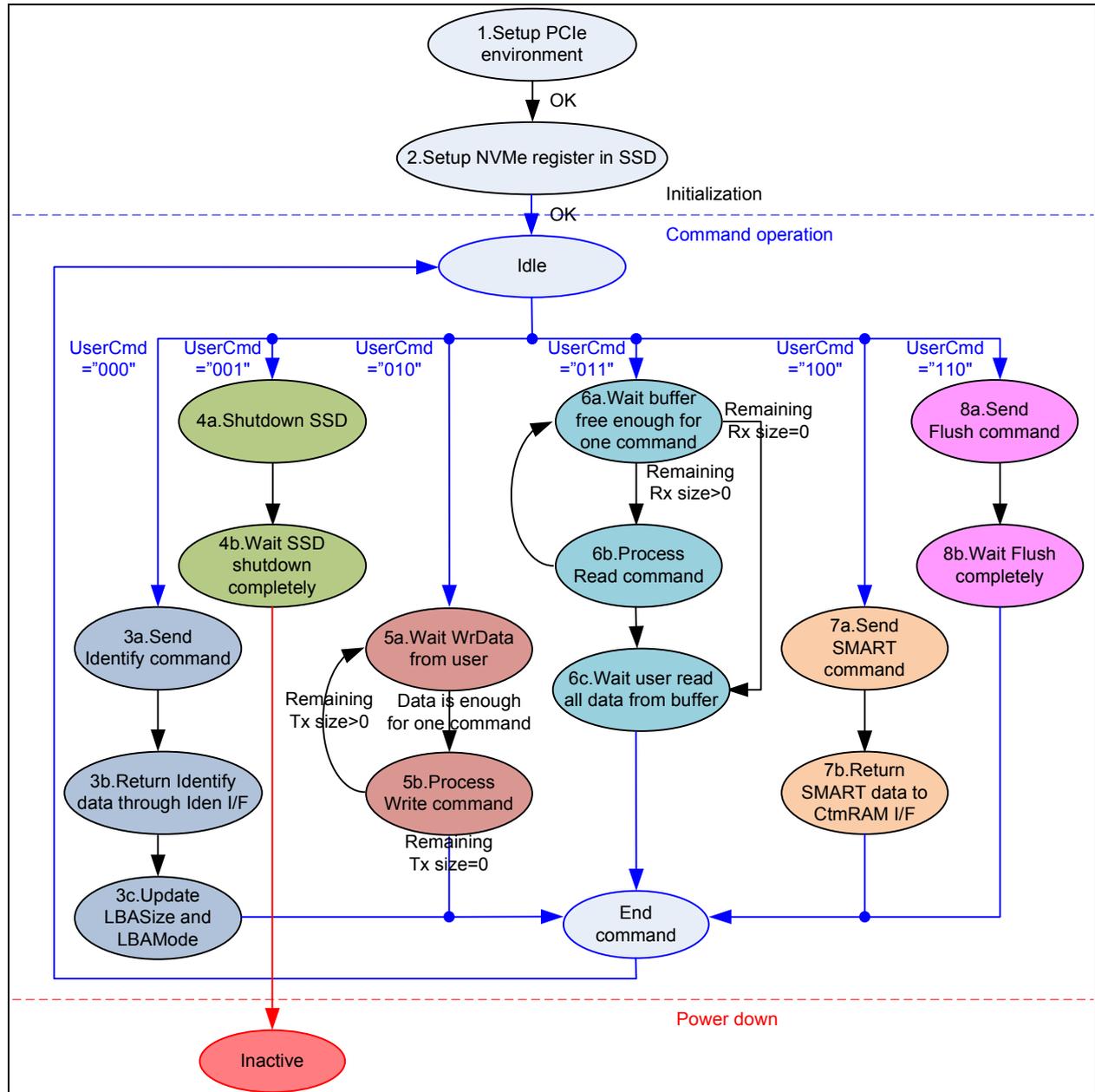


図 2: NVMe-IP コアの動作フロー

- 1) 本 IP コアは NVMe 動作に向けて PCIe 環境を設定します。
- 2) 本 IP コアは SSD 内 NVMe コントローラのパラメータを設定します。こ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(UserCmd="000")とする必要があり、このコマンドによって LBASize(SSD 容量)および LBA モード(LBA 単位が 512 バイトあるいは 4K バイト)が更新されます。
- 4) 電源切断前の最終コマンドは Shutdown コマンド(UserCmd="001")とする必要があります。このコマンドは SSD を正しいシーケンスで終了処理するために使います。電源切断前に Shutdown コマンドを発行しない場合、SSD 内のライト・データが保証できないことがあります。Shutdown コマンド実行後 SSD は非活動モードに入り電源を再投入しない限り一切のコマンドを受けなくなります。

- 5) ライト・コマンド(UserCmd="010")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。
- 6) リード・コマンド(UserCmd="011")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。
- 7) SMART コマンド(UserCmd="100")の場合、以下の処理を行います。
 - 本 IP コアは NVMe SSD に対して SMART/Health 情報を取得するため Get Log Page コマンドを発行します。
 - SSD から 512 バイト・データが返送されるのでカスタム・コマンド RAM インターフェイス (CtmRamAddr=0x000-0x01F)を通してユーザ回路に転送します。
- 8) Flush コマンド(UserCmd="110")の場合、以下の処理を行います。
 - 本 IP コアは NVMe SSD に対して Flush コマンドを発行します。
 - SSD が動作完了のステータスを報告するまで待機します。

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 ホスト・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 ホスト・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。

NVMe ホスト・コントローラは各ユーザ・コマンドの packets・シーケンスを NVMe 規格に沿って制御します。ユーザからの入力パラメータが受信されるとコマンド・パケットを生成します。データ・コントローラは SSD への全パケットを生成/デコードする役目を持ちます。

より詳細については以下に説明します。

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 AXI4 ストリームを通して転送されます。

PCIe 統合ブロックはまず最初に CFG インターフェイ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 PCIe 統合ブロック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 2 種類のコマンド・タイプが定義されています、一つは Admin コマンドでもう一つは NVM コマンドです。Admin コマンドは Admin サブミッション・キューと Admin 完了キューを通して制御されます。一方 NVM コマンドは I/O サブミッション・キューと I/O 完了キューで制御されます。

Identify/Shutdown/ライト/リードの各コマンドの場合、サブミッション・キュー・エントリーが用意され完了キュー・エントリー内のステータス値が NVMe ブロックによりモニタされエラー状態がチェックされます。ライト/リード・コマンド実行時には 256K バイトのデータ・バッファがユーザ回路と SSD 間の転送データをバッファリングするために使われます。Identify コマンドの場合 SSD からの Identify データは Identify インターフェイスを通して転送されます。

SMART/Flush コマンドにおいてはカスタム・コマンド・インターフェイスを通してサブミッション・キューがユーザにより用意されます。NVMe ブロックは他のコマンドと同様完了キュー・エントリー内のエラー状態をモニタします。より詳細には、SMART/Flush コマンドの完了キュー・エントリーはカスタム・コマンド・インターフェイスを通じた出力信号にマッピングされます。SMART コマンドでの SSD からのデータ出力はカスタム・コマンド RAM インターフェイスを通してユーザ回路へ転送されます。

NVMe ブロック内の各モジュール詳細については以下の NVMe-IP コア・英文データシートを参照してください。

https://www.dgway.com/products/IP/NVMe-IP/dg_nvme_ip_data_sheet_en.pdf

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。また、SMART および Identify コマンドで SSD から返送されるデータの出力はシンプルなデュアルポート RAM に接続されます。Identify コマンドで返送されるデータ・サイズは 8K バイトで SMART コマンドで返送されるデータ・サイズは 512 バイトです。全てのデータ・インターフェイスのバス幅は 128 ビット幅です。

PCIe 統合ブロック

Xilinx 製 FPGA はデバイス内部に PCIe 接続用として統合されたハードウェア・ブロックを実装します。ひとつの FPGA デバイスで接続可能な NVMe SSD の最大数はそのデバイス内の PCIe 統合ブロック数に制限されます。PCIe 統合ブロックの詳細については以下の Xilinx 発行ドキュメントを参照してください。

- PG054: 7 Series FPGAs Integrated Block for PCI Express™
- PG023: Virtex-7 FPGA Gen3 Integrated Block for PCI Express
- PG156: UltraScale Devices Gen3 Integrated Block for PCI Express
- PG213: UltraScale+ Devices Integrated Block for PCI Express

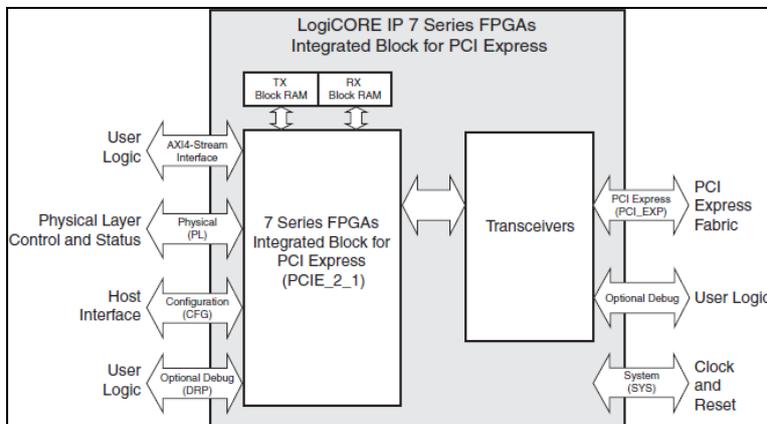


図 3: PCIe 統合ブロック(7 シリーズ FPGA)

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。(特に注意が必要な箇所は赤フォントで示しています。)

表 3: コア I/O 信号

信号名	方向	説明
システム信号 (dglF TypeS)		
RstB	In	アクティブ Low のコア同期リセット信号、Clk 信号が安定となったら Clk に同期して '1' に解除すること
Clk	In	本 IP コアの動作システム・クロック、Clk の周波数は(PCle 統合ブロックからの出力クロックである)PCleClk と同一かそれ以上とする必要がある。(PCle Gen2 の場合 125MHz かそれ以上、Gen3 の場合 250MHz かそれ以上)
コントロール・インターフェイス (dglF TypeS 制御系、Clk に同期)		
UserCmd[2:0]	In	ユーザ・コマンド種別、("000": Identify, "001": Shutdown, "010": ライト, "011": リード, "100": SMART, "110": Flush, "101": 未使用, "111": 未使用)
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、4K セクタ・フォーマット(LBAMode='1')の場合、UserAddr[2:0]は常に"000"とし 4K バイトのアライメントに合わせなくてはならない。512 バイト・フォーマット(LBAMode='0')の場合も、SSD の特性として UserAddr[2:0]は"000"をセットし SSD ページ・サイズの 4K バイト・サイズのアライメントと合わせることが推奨される。殆どの SSD は開始セクタ・アドレスが 4K バイトのアライメントに揃っていない場合、ライト/リードのパフォーマンスは低下する。
UserLen[47:0]	In	総転送セクタ・サイズを 512 バイト単位でセットする、有効な範囲は 1 以上かつ(LBASize-UserAddr)以下。LBAMode='1' の場合、UserLen[2:0]は必ず"000"として 4K バイト・サイズのアライメントと合わせる必要がある。
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。コマンド・パラメータ(UserCmd, UserAddr, UserLen, CtmSubmDW0-DW15)は UserReq='1'の期間中は有効な値を維持する必要がある。
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 512 バイト単位で表示、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。
LBAMode	Out	SSD のセクタ・フォーマット。('0': 512 バイト・セクタ, '1': 4K バイト・セクタ)、初期値は 0。ユーザ回路から IDENTIFY DEVICE コマンドを指定し SSD のセクタ・フォーマット情報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。

(※) 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
コントロール・インターフェイス [続き] (dglF TypeS 制御系、Clk に同期)		
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(Contoller 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 * MQES (サポートする最大キュー・エン트리値)が 7 またはそれ以上 各レジスタのより詳細は NVMeCAPReg 信号を参照すること [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 [6] – 完了 TLP パケット・サイズが正しくない [7] – PCIe 統合ブロックが内部バッファで ECC エラーを検出した [8] – 完了 TLP パケットで未サポートの要求(UR)フラグを検出した [9] – 完了 TLP パケットでコンプリータ・アポート(CA)フラグを検出した [15:10] – 未使用 [16] – サポートされない LBA ユニット(LBA ユニットが 512 バイトか 4K バイトでない) [31:17] – 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
データ・インターフェイス [続き] (dglF TypeS データ系、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。(この信号は使われていない)
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 Xilinx 標準 FIFO ライブラリをそのまま使用します。
- 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
NVMe-IP インターフェイス信号 Clk に同期		
IPVersion[31:0]	Out	IP コアのバージョン番号
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)、0 をセットするとタイムアウト機能は無効となる。
PCleLinkup	In	PCIe 統合ブロックの 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] - MPSPMIN (Memory Page Size Minimum) [31:25] - 未使用
IdenWrEn	Out	IdenWrData および IdenWrAddr と同期して出力するライト・イネーブル信号
IdenWrDWEn[3:0]	Out	IdenWrData における 32 ビット毎の DWord データ・イネーブル、IdenWrEn に同期 Bit[0-3] はそれぞれ 128 ビット幅の IdenWrData における [31:0], [63:32], [95:64], [127:96] の DWord 毎のイネーブルに対応する
IdenWrAddr[7:0]	Out	IdenWrData のインデックスを 128 ビット単位で示す、IdenWrEn に同期 0x000-0x0FF の 4k バイトは Identify Controller データ 0x100-0x1FF の 4K バイトは Identify Namespace データ
IdenWrData[127:0]	Out	4K バイトの Identify Controller データまたは Identify Namespace データ、IdenWrWrEn 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
NVMe-IP カスタム・インターフェイス信号 Clk に同期		
CtmSubmDW0[31:0]- CtmSubmDW15[31:0]	Out	SMART または Flush コマンド用のサブミッション・キュー・エントリ出力 16D ワード DW0-DW15 が Command DWord0-15 に対応する。 全信号は UserReq='1'かつ UsrCmd='100'(SMART)または'110'(Flush)のときに有効な値を維持する必要がある。
CtmCompDW0[31:0] - CtmCompDW3[31:0]	Out	SMART/Flush コマンドからの完了キュー・エントリ出力 4D ワード DW0-3: 完了 DWord0-3
CtmRamWrEn	Out	SMART コマンドにおいて CtmRamAddr および CtmRamWrData のバス上に有効な値が出力されているときに '1' アサートする。
CtmRamWrDWEn[3:0]	Out	CtmRamWrData における 32 ビット毎の DWord データ・イネーブル、CtmRamWrEn に同期 Bit[0-3] はそれぞれ 128 ビット幅の CtmRamWrData における [31:0], [63:32], [95:64], [127:96] の DWord 毎のイネーブルに対応する
CtmRamAddr[8:0]	Out	SMART データ返送時に CtmRamWrData のインデックスを示す。 (オプションで)その他(SMART 以外)のカスタム・コマンドでも CtmRamRdData を通してデータを入力するときのインデックスとなる。
CtmRamWrData[127:0]	Out	SMART コマンドでの 512 バイト・データ出力
CtmRamRdData[127:0]	In	(オプションで)その他のカスタム・コマンドにおけるデータ入力

信号名	方向	説明
Gen3 用 PCIe 統合ブロック・インターフェイス (PCIeClk に同期)		
PCleRstB	In	アクティブ Low の PCIe 同期リセット信号、PCIe 統合ブロックがリセット状態でない場合に '1' ネゲートする
PCleClk	In	PCIe 統合ブロックの同期クロック、PCIe Gen3 の場合 250MHz
Gen3 用コンフィグレーション管理インターフェイス (PCIeClk に同期)		
PCleCfgDone	In	リード/ライト動作完了、動作完了時 1 クロック分アサートする
PCleCfgRdEn	Out	リード・イネーブル、リード動作時にアサート
PCleCfgWrEn	Out	ライト・イネーブル、ライト動作時にアサート
PCleCfgWrData[31:0]	Out	コンフィグレーションおよび管理レジスタへのコンフィグレーションを行うためのライト・データ
PCleCfgByteEn[3:0]	Out	ライト・データのバイト・イネーブル、例えば bit[0] で PCleCfgWrData[7:0] がイネーブル
PCleCfgAddr[18:0]	Out	リード/ライト・アドレス
Gen3 用リクエスト要求インターフェイス (PCIeClk に同期)		
PCleMtTxData[127:0]	Out	リクエスト要求データ・バス
PCleMtTxKeep[3:0]	Out	ビット '1' が PCleMtTxData の (32 ビット幅の) D ワード '1' に有効なデータが含まれることを示す
PCleMtTxLast	Out	TLP の最終サイクルでアサートしパケットの末尾を示す
PCleMtTxReady[3:0]	In	データ受け取り信号、データは PCleMtTxValid と PCleMtTxReady がともにアサートしたクロック・サイクルで転送される
PCleMtTxUser[59:0]	Out	リクエスト要求ユーザ・データ、PCleMtTxValid が '1' アサートの時に有効
PCleMtTxValid	Out	PCleMtTxData バスに有効データ出力中を示す、本 IP コアはパケット転送中有効信号をアサートし続ける
Gen3 用コンプリータ要求インターフェイス (PCIeClk に同期)		
PCleMtRxData[127:0]	In	PCIe 統合ブロックからの受信データ
PCleMtRxKeep[3:0]	In	ビット '1' が PCleMtRxData の (32 ビット幅の) D ワード '1' に有効なデータが含まれることを示す
PCleMtRxLast	In	パケットの最終ビットでアサートしパケットの末尾を示す
PCleMtRxReady	Out	本 IP コアはデータを受信できる時にアサートする
PCleMtRxUser[74:0]	In	転送された TLP のサイドバンド情報、PCleMtRxValid が '1' アサート時に有効
PCleMtRxValid	In	PCIe 統合ブロックが PCleMtRxData バスに有効データ出力中を示す、PCIe 統合ブロックはパケット転送中有効信号をアサートし続ける
Gen3 用コンプリータ完了インターフェイス (PCIeClk に同期)		
PCleSITxData[127:0]	Out	本 IP コアからの完了データ
PCleSITxKeep[3:0]	Out	ビット '1' が PCleSITxData の (32 ビット幅の) D ワード '1' に有効なデータが含まれることを示す
PCleSITxLast	Out	パケットの最終サイクルでアサートしパケットの末尾を示す
PCleSITxReady[3:0]	In	PCIe 統合ブロックはデータを受信できる時にアサートする
PCleSITxUser[32:0]	Out	転送された TLP のサイドバンド情報、PCleSITxValid が '1' アサート時に有効
PCleSITxValid	Out	PCleSITxData バスに有効データ出力中を示す、本 IP コアはパケット転送中有効信号をアサートし続ける
Gen3 用リクエスト完了インターフェイス (PCIeClk に同期)		
PCleSIRxData[127:0]	In	PCIe 統合ブロックからの受信データ
PCleSIRxKeep[3:0]	In	ビット '1' が PCleSIRxData の (32 ビット幅の) D ワード '1' に有効なデータが含まれることを示す
PCleSIRxLast	In	パケットの最終ビットでアサートしパケットの末尾を示す
PCleSIRxReady	Out	本 IP コアはデータを受信できる時にアサートする
PCleSIRxUser[84:0]	In	転送された TLP のサイドバンド情報、PCleSIRxValid が '1' アサート時に有効
PCleSIRxValid	In	PCIe 統合ブロックが PCleSIRxData バスに有効データ出力中を示す、PCIe 統合ブロックはパケット転送中有効信号をアサートし続ける

(※) 注意

- 上記の各信号は Gen3 対応 PCIe 統合ブロック向けの NVMe-IP コア信号です、Gen2 対応版については次頁を参照してください。

信号名	方向	説明
Gen2 用 PCIe 統合ブロック・インターフェイス (PCIeClk に同期)		
PCleRstB	In	アクティブ Low の PCIe 同期リセット信号、PCIe 統合ブロックがリセット状態でない場合に '1' ネゲートする
PCleClk	In	PCIe 統合ブロックの同期クロック、PCIe Gen2 の場合 125MHz
Gen2 用コンフィグレーション管理インターフェイス (PCIeClk に同期)		
PCleCfgDone	In	リード/ライト動作完了、動作完了時 1 クロック分アサートする
PCleCfgRdEn	Out	リード・イネーブル、リード動作時にアサート
PCleCfgWrEn	Out	ライト・イネーブル、ライト動作時にアサート
PCleCfgWrData[31:0]	Out	コンフィグレーションおよび管理レジスタへのコンフィグレーションを行うためのライト・データ
PCleCfgByteEn[3:0]	Out	ライト・データのバイト・イネーブル、例えば bit[0] で PCleCfgWrData[7:0] がイネーブル
PCleCfgAddr[18:0]	Out	リード/ライト・アドレス
Gen2 用送信インターフェイス (PCIeClk に同期)		
PCleTxData[127:0]	Out	PCIe 統合ブロックへの送信データ
PCleTxKeep[15:0]	Out	送信データ・ストロー部、PCleTxData のどのデータ・バイトが有効かを示す
PCleTxLast	Out	フレーム末尾の送信、PCleTxValid がアサート時のみ有効
PCleTxReady[3:0]	In	PCIe 統合ブロックがデータ受信できる時にアサート、PCITxValid と PCleTxReady が同時にアサートされたときに PCleTxData の 1 データ・バイト転送が成功したことを示す。
PCleTxUser[3:0]	Out	Bit[3]: 転送元が廃止された Bit[2]: 転送動作が(正しく停止せず)超えてしまった Bit[1]: 転送エラーが発生した Bit[0]: 転送時の ECRC が発生した 本 IP コアでは常に '0000b' を出力する
PCleTxValid	Out	本 IP コアが PCleTxData 上に有効なデータを出力していることを示す
Gen2 用受信インターフェイス (PCIeClk に同期)		
PCleRxData[127:0]	In	PCIe 統合ブロックからの受信データ、PCleTxValid がアサート中有効
PCleRxKeep[15:0]	In	受信データ・ストロー部、PCleRxData のどのデータ・バイトが有効かを示す
PCleRxLast	In	フレーム末尾の受信、PCleRxValid がアサート時のみ有効
PCleRxReady	Out	本 IP コアが PCleRxData のデータ受信できる時にアサート、PCleRxValid と PCleRxReady が同時にアサートされたときに PCleRxData の 1 データ・バイト転送が成功したことを示す
PCleRxUser[21:0]	In	Bit[0]: 受信 ECRC エラー Bit[14:13]: PCleRxData の新しいパケット・ヘッダ開始を示す Bit[21]: PCleRxData のパケット末尾を示す 本 IP コアではその他のビットは無視する
PCleRxValid	In	本 IP コアが PCleRxData 上に有効なデータを出力していることを示す

(※) 注意

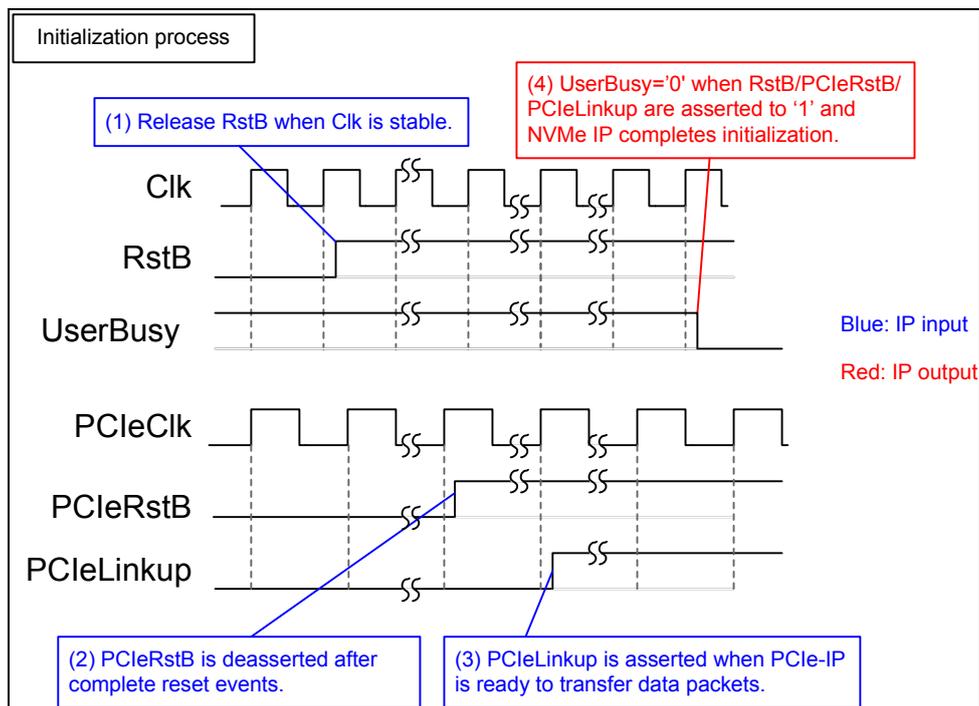
- 上記の各信号は Gen2 対応 PCIe 統合ブロック向けの NVMe-IP コア信号です、Gen3 対応版については前頁を参照してください。

タイミング波形

初期化

本 IP コアの初期化プロセスは以下の通りです

- 1) 同期クロック信号(Clk 入力信号)が安定した後で 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 統合ブロック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 統合ブロックおよび接続したアップストリーム・リンク先ポートの準備が整いデータ・パケットの交換ができるようになると 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。
- 上記初期化シーケンスが完了した後に、本 IP コアはユーザ回路からのコマンドを受信できるようになります。



- (1) Clk が安定すると RstB は '1' にリリースされます
- (2) リセット処理が完了すると PCIeRstB がネゲートします
- (3) PCIe 統合ブロックのデータ・パケット転送準備が整うと PCIeLinkup がアサートされます
- (4) UserBusy は RstB, PCIeRstB, PCIeLinkup のすべてが '1' アサートされ IP コアの初期化が完了した後に '0' ネゲートされます

図 4: システム起動時の UserBusy ネゲート・タイミング

ユーザ・インターフェイス(dgIF typeS)

制御インターフェイス

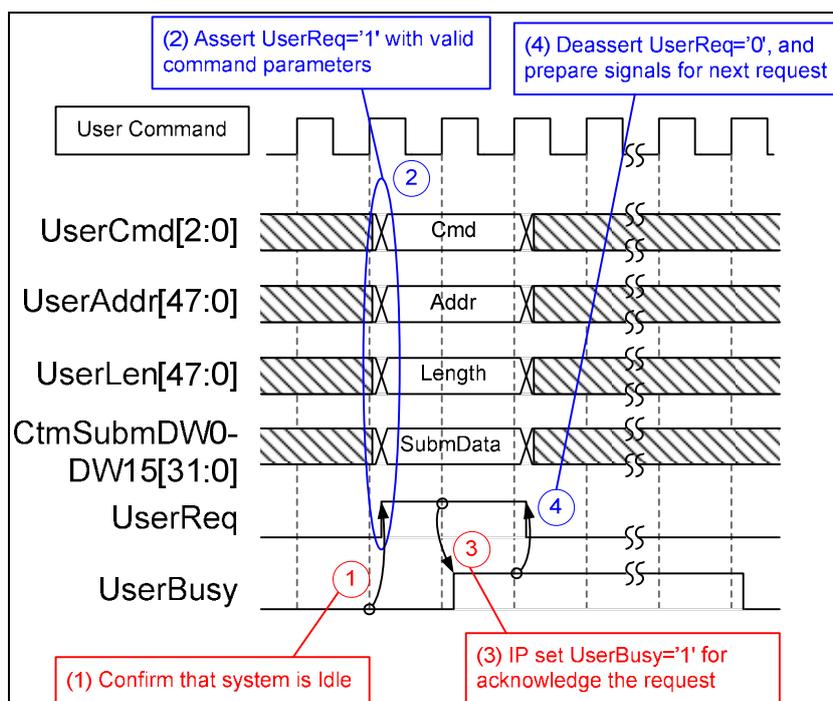
ユーザ・インターフェイスは2種類のインターフェイスに分類されます、すなわち一つは制御インターフェイスでもう一つはデータ・インターフェイスです。これらユーザ・インターフェイスを非常にシンプルで使いやすい仕様で定義したものが dgIF で、dgIF typeS は DesignGateway 社のストレージ系 IP コア間で共通のユーザ・インターフェイスです。

図 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, UserLen, そして CtmSubmDW0-DW15 は UserReq='1' でコマンドを出力する期間は図 5 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

- 1) UserAddr や UserLen は Identify, Shutdown, SMART, Flush コマンドでは使われないため無視されます。
- 2) CtmSubmDW0-DW15 は Identify, Shutsown, Write, Read 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。

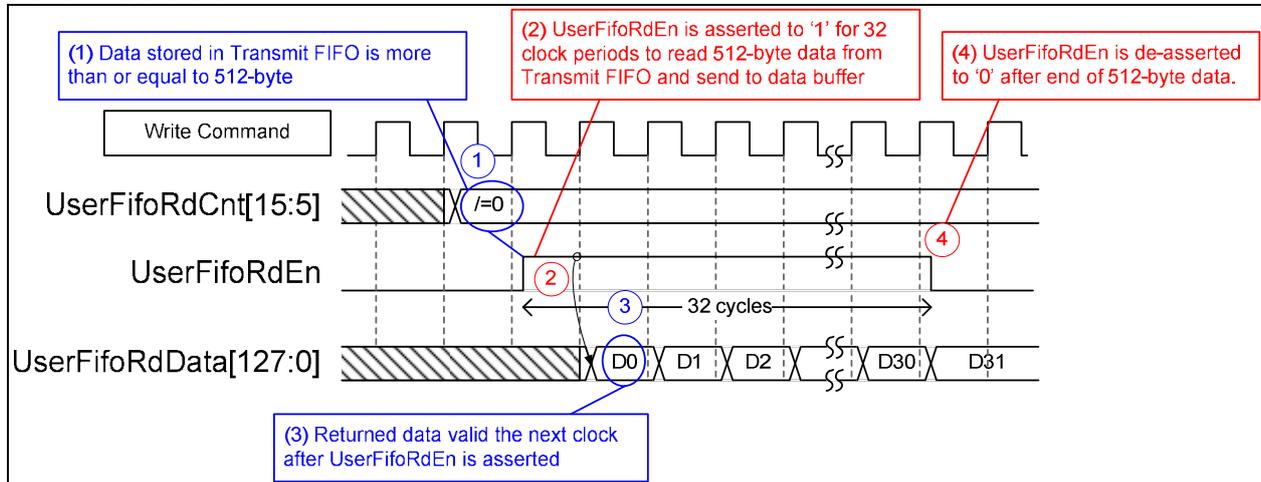


- (1) システムがアイドル状態であることを確認します
- (2) 有効なコマンド・パラメータをセットし UserReq='1' をアサートします
- (3) IP コアは要求を受付けたことを UserBusy='1' で示します
- (4) ユーザ回路は UserReq='0' とし、次のコマンドのためのパラメータを準備できます

図 5: 制御インターフェイスのタイミング波形

データ・インターフェイス

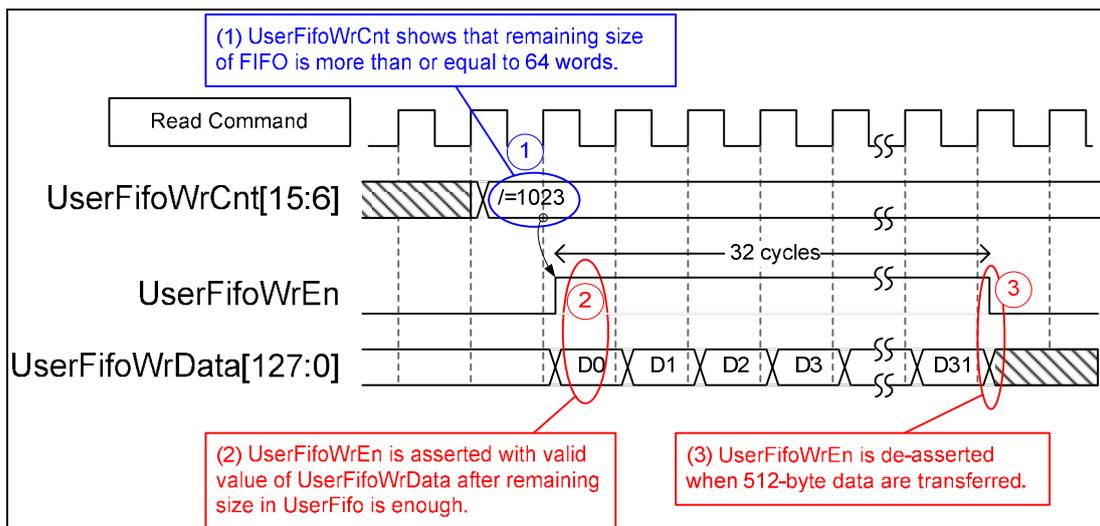
ライト・コマンドにおいてユーザ回路内の送信 FIFO からのライト・データは本 IP コア内のデータ・バッファへと転送されます。IP コア内部 DMA エンジンが UserFifoRdCnt 信号を監視し送信 FIFO 内のデータが 512 バイトまたはそれ以上となるまで待機します。その後図 6 に示すように UserFifoRdEn を 32 クロック期間アサートし 512 バイト・データを転送します。一般的な FIFO タイミングと同様、UserFifoRdData は UserFifoRdEn がアサートされた次のクロック期間で出力されます。



- (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 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) 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。



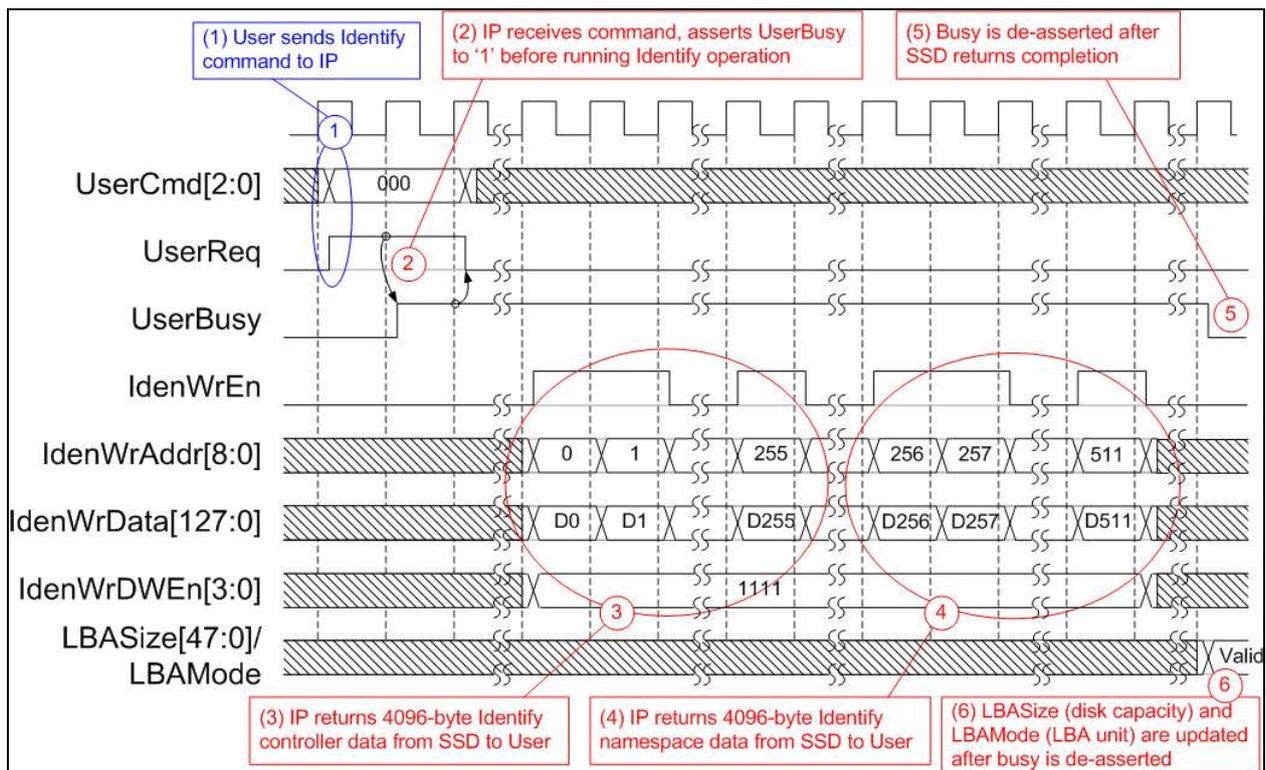
- (1) 受信 FIFO に 64 ワードかそれ以上の空き領域があることを UserFifoWrCnt で確認します
- (2) 十分な空き領域を確認した次クロックから UserFifoWrEn と UserFifoWrData が出力されます
- (3) UserFifoWrEn は 512 バイトを転送すると'0'ネゲートします

図 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(Xilinx 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

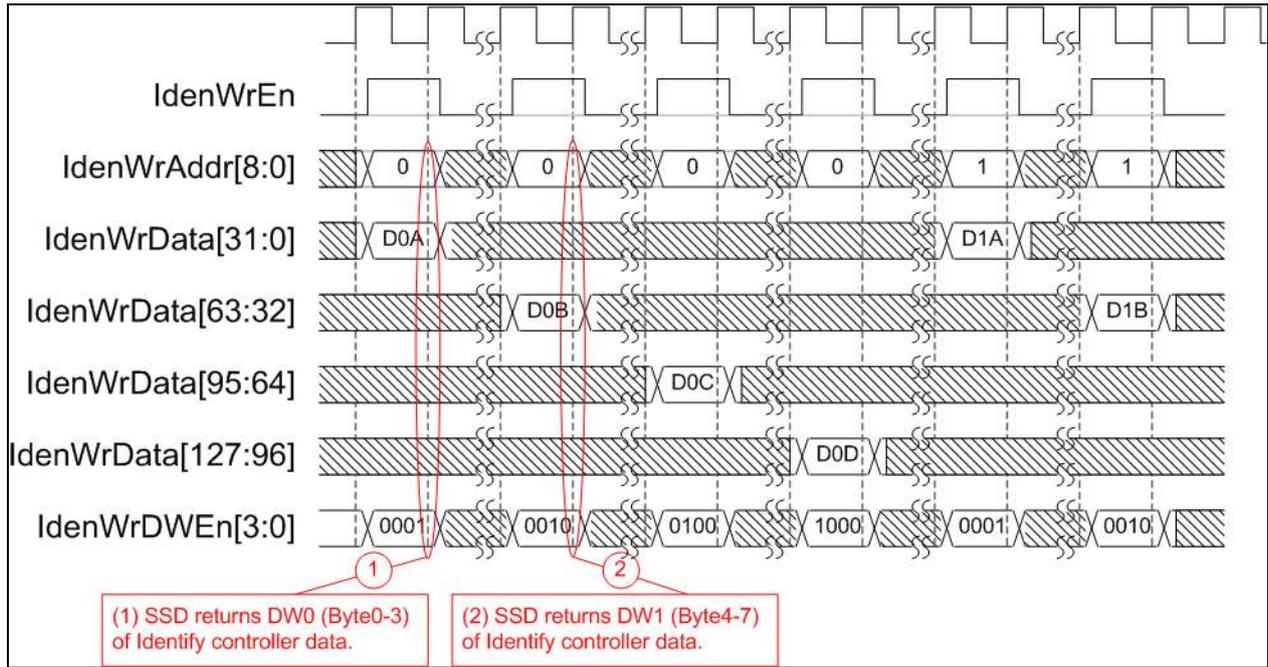
本 IP コアにライトあるいはリード・コマンドを送信する前に、ユーザ回路はまず IDENTIFY コマンドを発行し LBASize と LBAMode 出力を更新する必要があります。LBASize 値はユーザ回路にてリード・ライト命令のアドレス + 転送長の合計がそれぞれこの値を超えないよう制御するために使います。また、LBAMode='1' (LBA ユニットが 4K バイト) の場合、ライト・リードコマンドにおける UserAddr と UserLen はいずれも 4K バイトのアライメントに合わせなくてはなりません。



- (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 が完了を返すと 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 と LBAMode(LBA ユニット)も更新されます

図 8: IDENTIFY コマンド後に LBASize と LBAMode 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。IDENTIFY コントローラ・データ転送中 IdenWrAddr は 0-255 であり IDENTIFY ネームスペース・データ転送中 IdenWrAddr は 256-511 となります。8 バイト(128 ビット)幅データは各クロック期間で有効な値を出力し IdenWrAddr および IdenWrEn 信号に同期します。一般的に IdenWrDWEEn は '1111' となり全 128 ビットの IdenWrData が有効となります。最後に UserBusy がネゲートされるとともに、LBASize データと LBAMode 出力が更新されます。



- (1) SSD は IDENTIFY コントローラ・データのうち DW0(Byte0-3)を返送する
- (2) SSD は IDENTIFY コントローラ・データのうち DW1(Byte4-7)を返送する

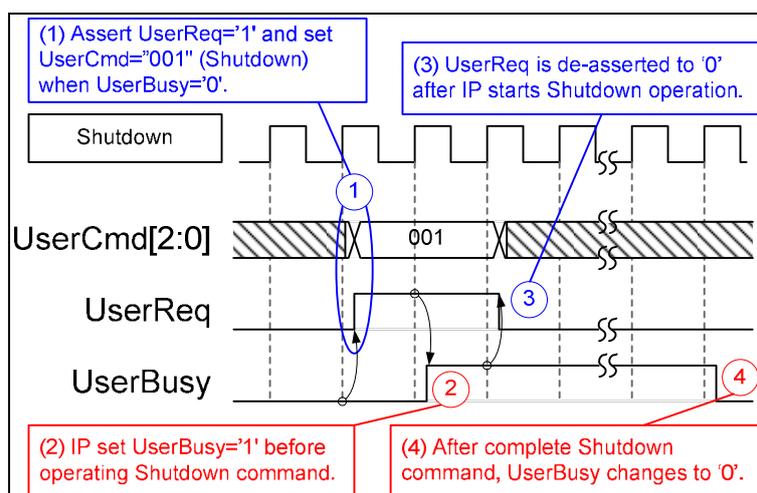
図 9: IdenWrDWEEn のタイミング波形

接続した SSD によっては IDENTIFY コントローラ・データや IDENTIFY ネームスペース・データが 128 ビット幅フォーマットではなく 32 ビット幅フォーマットで返送されることがあります。このため IdenWrData 信号のうち有効な Dword を示すため IdenWrDWEEn を使う必要があります。図 9 に示すように IdenWrDWEEn のビット[0]は IdenWrData のビット[31:0]が有効であることを示します。(その他のビット[1]/[2]/[3]も同様にそれぞれの DWord すなわち IdenWrData のビット[63:32]/[95:64]/[127:96]が有効であることを示します。) この IdenWrDWEEn も IdenWrData と同様 IdenWrEn='1' で有効となります。

Shutdown

システムの電源を切断する前には本コマンドにより NVMe SSD を Shutdown することが推奨されます。このコマンドを発行することで SSD は正しい Shutdown シーケンスを実行します。Shutdown の実行により SSD 内のキャッシュデータは不揮発メモリ領域に退避されるので、SSD への全ライト・データが保証されます。本コマンド発行後は本 IP コアおよび接続先 SSD はシステムの電源が再投入されるまで一切アクセスできなくなります。

Shutdown コマンドは UserCmd を"001"にセットして送信します。UserAddr, UserLen, CtmSubmDW0-DW15 は本コマンドでは使われません。



- (1) ユーザ回路は UserBusy='0' ネゲートを確認後、UserCmd="001"として Shutdown コマンドの発行を IP コアに要求します
- (2) IP コアは Shutdown シーケンス実行前に UserBusy='1' アサートします。
- (3) ユーザ回路が UserReq を'0'ネゲートした後で IP コアは Shutdown シーケンスを開始します。
- (4) IP コアは NVMe SSD に対する Shutdown シーケンスを完了すると UserBusy を'0'ネゲートします。

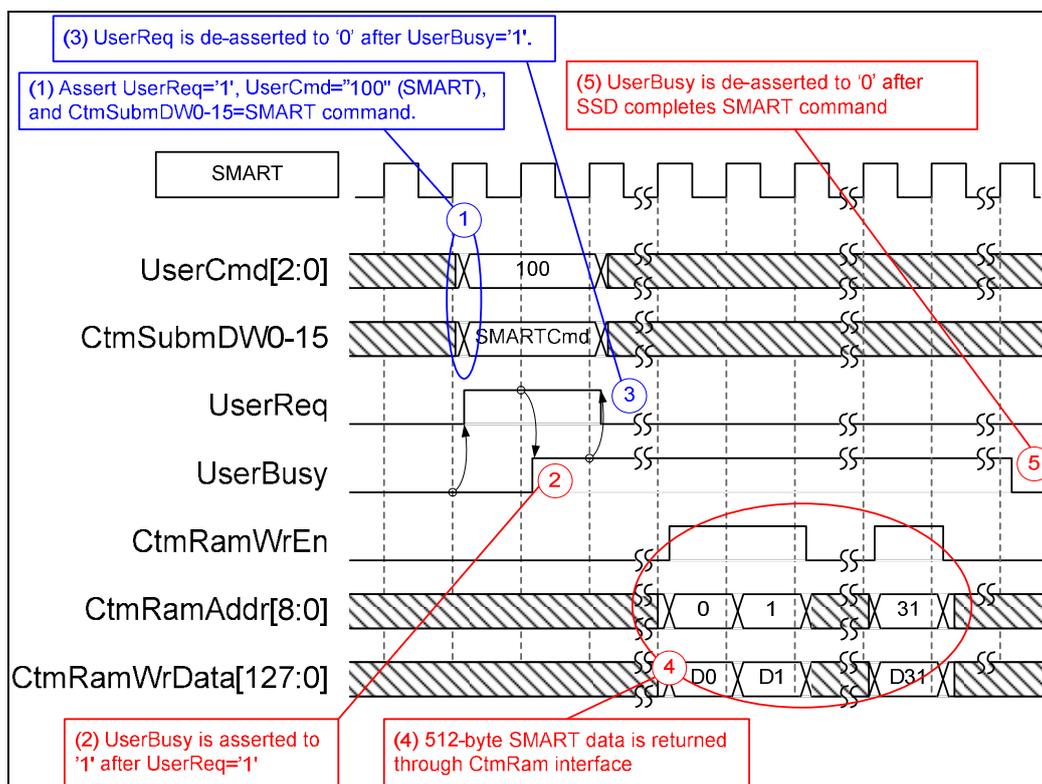
図 10: SHUTDOWN コマンドの送信

SMART

SSD の健康状態を確認するため SMART コマンドを実行し、カスタム・コマンド・インターフェイスを通して 512 バイトの SMART データを取得することができます。SMART コマンドを実行する場合 UserCmd="100" をセットしますが合わせて 16 個の D ワード・データを CtmSubmDW0-DW15 に設定して UserReq を '1' アサートします。UserAddr と UserLen 入力は本 SMART コマンドでは使われません。その後 SSD は CtmRam(カスタム・コマンド RAM)インターフェイスを通して 512 バイトの SMART データを返送します。CtmRamWrEn='1' アサートされた同じクロック期間で 128 ビットの SMART データが CtmRamWrData バス上に出力されます。CtmRamAddr は 32 サイクル x 128 ビット(8 バイト)で出力される SMART データのデータ・インデックスとして使います。

SMART コマンドにおける CtmSubmDW0-DW15 は以下の値にセットする必要があります。

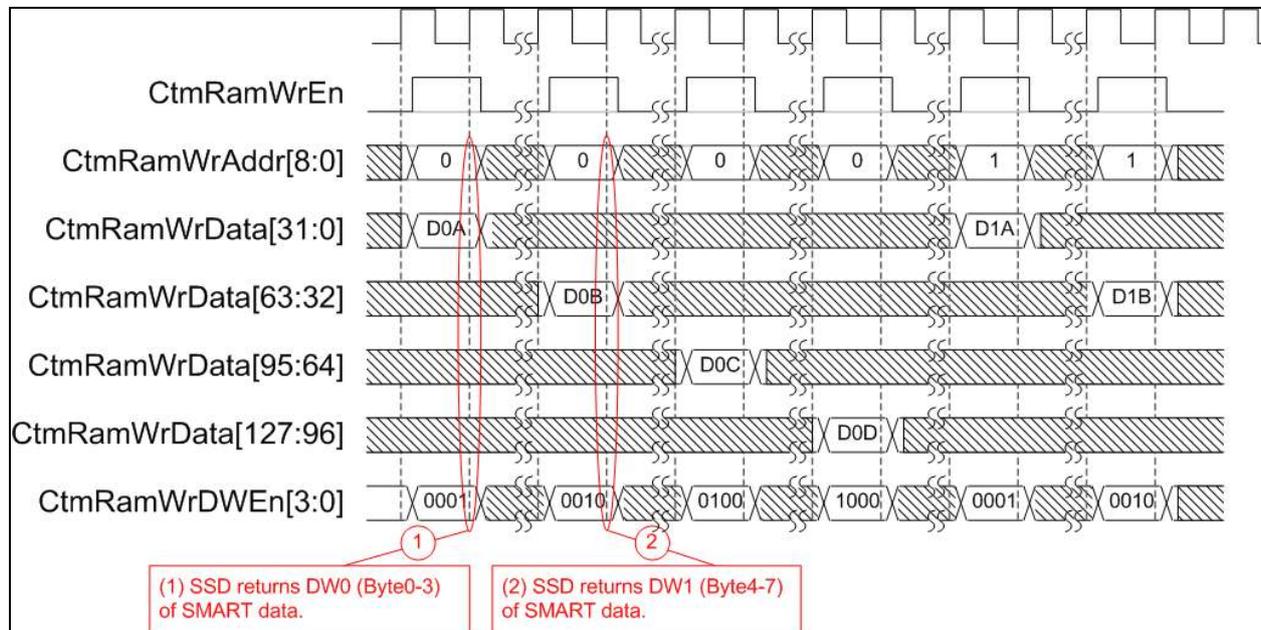
CtmSubmDW0	= 0x0000_0002
CtmSubmDW1	= 0xFFFF_FFFF
CtmSubmDW2 – CtmSubmDW5	= 0x0000_0000
CtmSubmDW6	= 0x2000_0000
CtmSubmDW7 – CtmSubmDW9	= 0x0000_0000
CtmSubmDW10	= 0x007F_0002
CtmSubmDW11 – CtmSubmDW15	= 0x0000_0000



- (1) ユーザ回路は UserCmd="100" および CtmSubDW0-DW15 を SMART コマンドに設定し UserReq= '1' アサートします。
- (2) IP コアは UserReq= '1' で UserBusy='1' アサートします。
- (3) UserBusy='1'アサート後ユーザ回路は UserReq を'0'ネゲートします。
- (4) 512 バイトの SMART データが CtmRam インターフェイスを通して転送されます。
- (5) SMART コマンドが完了すると IP コアは UserBusy を'0'ネゲートします。

図 11: SMART コマンドの実行例

図 11 に示すように SMART データは CtmRam インターフェイスを通して送信されます。CtmRamWrEn は連続して出力されないことがあるので注意してください。CtmRamAddr は CtmRamWrData のインデックスを示します。CtmRamAddr=0 で CtmRamWrEn='1' の場合、SMART データのうちバイト 0～バイト 15 の 16 バイト(128 ビット)分が CtmRamWrData 信号上に出力されます。同様に最終 SMART データ(バイト 496～バイト 511)は CtmAddr=31 で CtmRamWrEn='1' の場合に出力されます。一般的に CtmRamWrDWEEn は '1111' となり CtmRamWrData の全 128 ビットが同時に有効となります。



- (1) SSD は SMART データのうち DW0(Byte0-3)を返送する
 (2) SSD は SMART データのうち DW1(Byte4-7)を返送する

図 12: CtmRamWrDWEEn のタイミング波形

IDENTIFY コマンドと同様に接続した SSD によっては SMART データが 128 ビット幅フォーマットではなく 32 ビット幅フォーマットで返送されることがあります。このため CtmRamWrData 信号のうち有効な Dword を示すため CtmRamWrDWEEn を使う必要があります。図 12 に示すように CtmRamWrDWEEn のビット[0]は CtmRamWrData のビット[31:0]が有効であることを示します。(その他のビット[1]/[2]/[3]も同様にそれぞれの DWord すなわち CtmRamWrData のビット[63:32]/[95:64]/[127:96]が有効であることを示します。) この CtmRamWrDWEEn も CtmRamWrData と同様 CtmRamWrEn='1' で有効となります。

Flush

一般的にユーザが SSD へデータをライトする場合ライト・データはまず SSD の内部キャッシュに格納されます。SSD がキャッシュ内のデータを不揮発領域に転送する前に電源が突然切断された場合、キャッシュ内のライト・データは損失します。データの損失を防ぐには以下 2 つの方法があります。

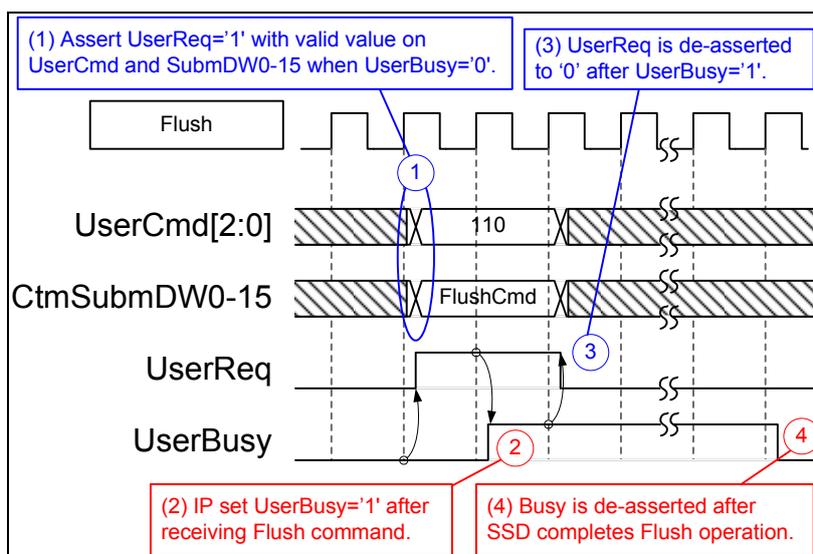
ひとつはシステムの電源切断前に Shutdown コマンドを発行することです。このコマンドはシステムの電源切断前に必要です。そしてもう一つの方法としては SSD へ Flush コマンドを発行することです。本 Flush コマンドを使うことでキャッシュ内のデータは SSD の不揮発メモリ領域へと転送されます。この Flush コマンドはシステムがアイドル中で実行でき、その時点での全ライト・データを不揮発領域に退避します。Flush コマンドを発行した後での本 IP コアや SSD はアクティブ状態を維持します。

SMART コマンドと同様に Flush コマンドも CtmSubmDW0-DW15 を以下の値でセットする必要があります。

CtmSubmDW0	= 0x0000_0000
CtmSubmDW1	= 0x0000_0001
CtmSubmDW2 – CtmSubmSW15	= 0x0000_0000

UserAddr と UserLen 入力は Flush コマンドでは使われません。

図 13 に示すように Flush コマンドを発行した後、ユーザ回路は UserBusy が '1' から '0' に遷移するのを確認し、キャッシュ内ライト・データの不揮発領域への退避が完了するまで待機します。



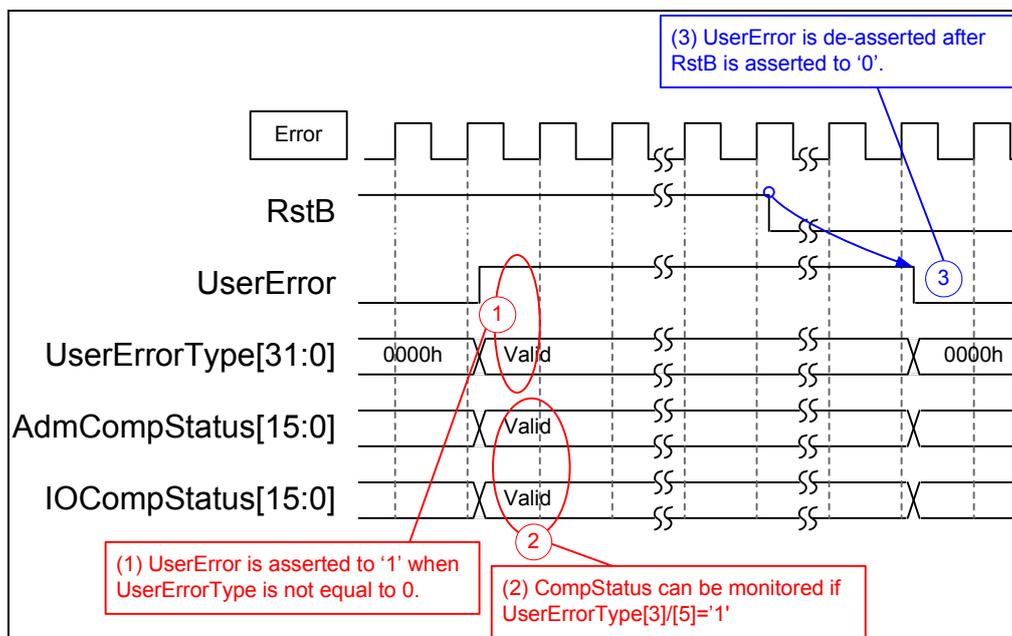
- (1) ユーザ回路は UserBusy='0' を確認し UserCmd="110" および CtmSubDW0-DW15 を Flush コマンドに設定し UserReq='1' アサートします。
- (2) IP コアは UserReq='1' で Flush コマンドを受信すると UserBusy='1' アサートします。
- (3) UserBusy='1' アサート後ユーザ回路は UserReq を '0' ネゲートします。
- (4) SSD で Flush 動作が完了すると UserBusy が '0' ネゲートします。

図 13: Flush コマンドの実行例

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 14 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
- (3) UserError は RstB が '0' アサートされるとクリアされます。

図 14: 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 KC705/VC707/VC709/ZC706/AC701/Zynq Mini-ITX/KCU105/ZCU106/VCU118 等の Xilinx 標準評価ボードと AB16-PCIeXOVR/AB17-M2FMC 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-VUP	Virtex Ultrascale+	VCU118	発売中
NVMe-IP-ZUP	Zynq Ultrascale+	ZCU106	発売中
NVMe-IP-KU	Kintex Ultrascale	KCU105	発売中
NVMe-IP-KT7	Kintex-7	KC705	発売中
NVMe-IP-AT7	Artix-7	AC701	発売中
NVMe-IP-VT7	Virtex-7	VC707 or VC709	発売中
NVMe-IP-ZQ7	Zynq-7000	ZC706	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、対象ファミリの Xilinx 評価ボードおよび AB16-PCIeXOVR/AB17-M2FMC 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
2.0J	2017/06/08	AXI PCIe ブリッジをコア内部に取り込みコアと PCIe 統合ブロックを直結する改良を実施
2.01J	2017/06/09	誤字の修正等
2.1J	2017/12/02	Zynq UltraScale+ (ZCU106) を正式サポート
3.0J	2018/07/23	4K セクタ・フォーマット, SMART/Flush/Shutdown 各コマンドをサポートした Version4 コアに更新
3.1J	2018/11/27	128 ビットのアライメントに合致しないメモリ・ライト要求をサポート
3.2J	2018/11/28	Artix-7(AC701) を正式サポート, 表 1/表 2 の各リソース消費情報を更新