



XAPP1263 (v1.0) 2015 年 8 月 17 日

# AXI USB 2.0 デバイス： バルク転送およびアイソクロナス転送における パフォーマンスの検証

著者：Prasad Gutti

## 概要

このアプリケーション ノートでは、バルクおよびアイソクロナス トランザクションにおいて、AMBA® (Advanced Microcontroller Bus Architecture) AXI (Advanced eXtensible Interface) を用いるザイリンクス USB (Universal Serial Bus) 2.0 [参照 1] 高速デバイスが発揮するパフォーマンスを検証します。作成されるテスト システムは、Kintex® UltraScale All Programmable SoC に基づいています。バルク トランザクションとアイソクロナス トランザクションのパフォーマンスは、2 つの別々のホスト ドライバーで測定されます。ここでは、バルクおよびアイソクロナス トランザクション用の USB システムと対応する ELF ファイルの開発方法について説明します。この資料は、ザイリンクスの Vivado® Design Suite および Vivado IP インテグレーター [参照 2] に関する基本的な知識を前提としています。

## はじめに

AXI USB 2.0 デバイスは、最小限のリソースでデザインの USB 接続を可能にします。このインターフェイスは、USB を中心とした高性能デザイン、ブリッジ、および従来ポートの置き換えに最適です。USB 2.0 プロトコルは、単一の半二重シリアルバス上で多数のデバイスを接続します。AXI USB 2.0 デバイスは、480Mb/s (高速) または 12Mb/s (最速) で動作し、プラグアンドプレイに対応しています。ホストはバスを制御し、要求された動作を指定するトークンをデバイスに送信します。AXI USB 2.0 デバイスは、最大 8 つのエンドポイントをサポートします。これには、データのソースまたはシンクとなるデフォルトの制御エンドポイントが含まれます。USB 2.0 デバイスのエンドポイント 0 は、制御トランザクションでデバイスを列挙するのに使用されます。

AXI USB 2.0 デバイスの 7 つのユーザー エンドポイントは、バルク、割り込み、またはアイソクロナスとして構成できます。また、各エンドポイントは、(ホストへの) 入力または (ホストからの) 出力として構成できます。各エンドポイントは、同じサイズのエンドポイント データ用 ping/pong バッファを 2 つ備えています。ユーザー エンドポイントのデータ バッファは単方向であり、各エンドポイントのエンドポイント コンフィギュレーション/ステータス レジスタで設定されます。バッファのサイズは、バルク エンドポイントは 0 ~ 512 バイト、割り込みエンドポイントは 64 バイト、アイソクロナス エンドポイントは最大 1,024 バイトに設定できます。

このアプリケーション ノートは、Kintex UltraScale デバイスのバルクおよびアイソクロナス トランザクションで USB が発揮するパフォーマンスの検証を目的としています。ここでは、次の要素を使用します。

- AXI USB 2.0 デバイスおよび DDR コントローラーを備えた MicroBlaze™ プロセッサ リファレンス システム
- デバイスをバルク エンドポイントまたはアイソクロナス エンドポイントとして構成する、ELF (Executable and Linkable Format) ファイル
- USB デバイスのパフォーマンス測定に使用されるホスト アプリケーション

AXI USB 2.0 デバイスは、エンドポイント バッファとメモリ間でデータを転送する内蔵 DMA (Direct Memory Access) コントローラーで構成されます。USB 2.0 デバイス コアは、専用の AXI4 スレーブ インターフェイスとオプションの AXI4 マスター インターフェイスで構成されます。オプションのマスター インターフェイスは、デバイス パラメーター Enable DMA Support によって設定されます。AXI USB 2.0 Controller が DMA をサポートするように設定すると、デザインに DMA ロジックが組み込まれます。DMA ロジックが組み込まれている場合は、AXI スレーブ インターフェイスからユーザー エンドポイント バッファにアクセスできません。DMA が無効の場合は、AXI スレーブ インターフェイスからエンドポイント バッファにアクセスできます。

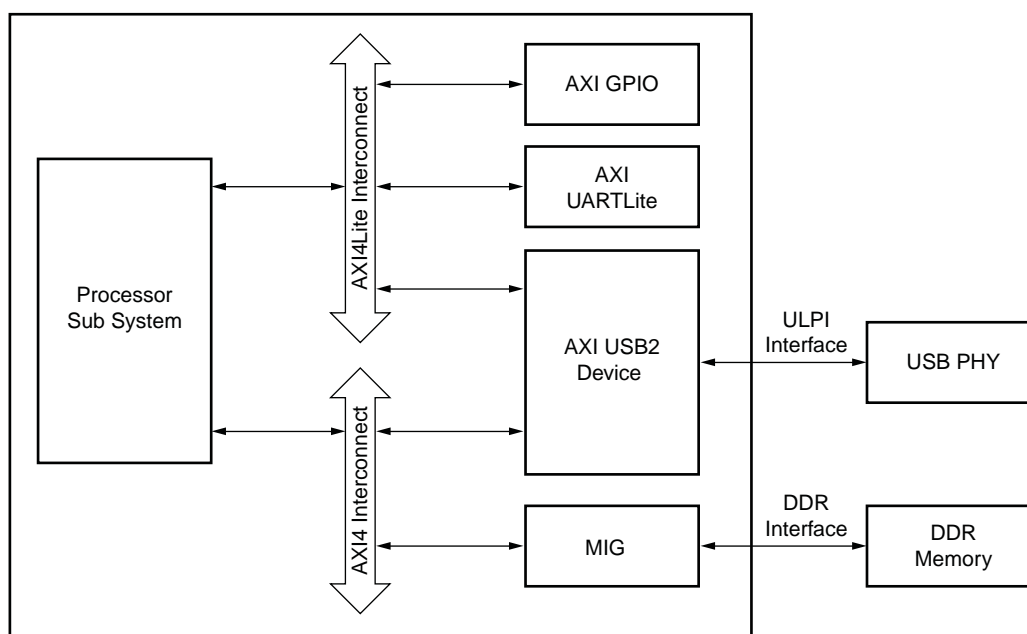
# リファレンス デザイン

## ハードウェア

このシステムは AXI に基づいています。AXI は、AMBA 仕様に従って標準化された IP インターフェイスプロトコルです。リファレンス デザインに使用される AXI インターフェイスは、AXI4 メモリ マップ方式インターフェイスと AXI4-Lite インターフェイスで構成されます。クロック ジェネレーターとプロセッサ システムのリセット ブロックは、システム全体にクロックおよびリセットを供給します。I/O ペリフェラルおよびプロセッサ サポート IP を含むエンベデッド MicroBlaze プロセッサ サブシステムにより、システムの高度な制御が行われます。

パフォーマンスとエリアのバランスをとるようにシステムを最適化するには、複数の AXI インターコネクトブロックを使用し、AXI インターコネクトブロックを個別に調整および最適化して、セグメント型または階層型の AXI インターコネクト ネットワークをインプリメントします。

AXI USB 2.0 デバイス アプリケーションの標準的なシステムを 図 1 に示します。



X891\_01\_100912

図 1 : AXI USB 2.0 デバイス アプリケーションの標準的なシステム

## AXI USB 2.0デバイスの使用

AXI USB 2.0 デバイス IP は、データ幅 32 ビットの AXI ベース システム上に接続できます。この IP は、USB PHY 側の ULPI インターフェイスをサポートします。さらに、パラメーター Enable DMA Support によって DMA モードまたは非 DMA モードに設定できます。DMA を有効にすると、AXI マスター インターフェイスが有効になります。AXI4 マスター インターフェイスは、DDR コントローラーがスレーブとして接続される AXI4 インターコネクトに接続する必要があります。IP の割り込みピンは、プロセッサ システムの割り込みコントローラーブロックに接続します。IP の設定に使用するパラメーターは次のとおりです。

- Enable DMA Support : 1 に設定すると、内蔵 DMA ブロックが AXI4 マスター インターフェイスと共にデザインに組み込まれます。この DMA ブロックにより、ユーザー エンドポイント バッファーは AXI スレーブ インターフェイスから切断されます。
- Enable Un-aligned Transactions : 1 に設定すると、データ リアライメントのサポートがデザインに追加されます。
- ULPI PHY Reset Polarity : このパラメーターは、PHY の要件に従って設定する必要があります。ACTIVE\_LOW に設定すると、AXI USB 2.0 デバイス IP によって生成される ULPI リセットはアクティブ Low 信号になります。

ACTIVE\_HIGH に設定すると、AXI USB 2.0 デバイス IP によって生成される ULPI リセットはアクティブ High 信号になります。

- Enable USB Error Counters : 1 に設定すると、USB リンク エラー ログ カウンターがデザインに追加されます。
- ENABLE HSIC PHY : HSIC PHY モジュールを有効にし、I/O レベルに HSIC インターフェイスを追加します。このパラメーターが有効に設定されている場合、ULPI インターフェイスは I/O レベルで利用できません。
- Connect Signaling Width : このパラメーターは、ENABLE HSIC PHY が 1 に設定されている場合のみ有効です。HSIC PHY が送信する接続信号のクロック サイクル数を示します。

## ソフトウェア

### マス ストレージ アプリケーション

マス ストレージ アプリケーションはバルク トランザクションを使用します。バルク トランザクションを利用するには、2つのエンドポイント バッファをバルク In およびバルク Out エンドポイントとして構成し、最大パケット サイズを 512 バイトに設定する必要があります。バルク In エンドポイントを使用して、読み出し操作のパフォーマンスを測定します。バルク Out エンドポイントを使用して、書き込み操作のパフォーマンスを測定します。

バルク アプリケーションは、システムを初期化して USB および UART デバイスを有効にします。その後、ホストからのデバイス列挙コマンド (SET/GET 記述子) に対応します。列挙が完了した後、デバイスがホストから読み出し要求を受信した場合は、次の手順が実行されます。

1. DMA を介して、USB デバイス内に存在する ping/pong バッファにデータが書き込まれます。
2. USB デバイスは、ホスト デバイスから In トークンを受信すると、ping/pong バッファ内に存在するデータで応答します。
3. 対応するエンドポイントのバッファ解放状態を示す割り込みが、プロセッサ リファレンス システムに送信されます。
4. プロセッサ リファレンス システムはバッファ解放割り込みを検出し、次に送信するデータで ping/pong バッファを充填します。

ホストがデータを送信する限り、この手順が繰り返されます。

デバイスが書き込みコマンドを受信した場合は、次の手順が実行されます。

1. ホスト デバイスは、Out トークンに続いてデータ パケットを送信します。
2. データを受信すると、受信データは ping/pong バッファに書き込まれます。
3. 対応するエンドポイントのバッファ フル状態を示す割り込みが、プロセッサ リファレンス システムに送信されます。
4. プロセッサ リファレンス システムはデータを読み出し、DMA を介して DDR にデータを保存します。
5. ホストがデータを送信する限り、この手順が繰り返されます。

### アイソクロナス アプリケーション

アイソクロナス アプリケーションの読み出しと書き込みは、別々のアプリケーションでテストされます。isoc\_in アプリケーションは、アイソクロナス トランザクションを使用する読み出し操作をテストします。isoc\_out アプリケーションは、書き込み操作のパフォーマンスを測定します。isoc\_in アプリケーションはデバイスをマイクロフォンとしてプログラムし、1つのエンドポイントをアイソクロナス In エンドポイントとして使用します。isoc\_out アプリケーションはデバイスをスピーカーとしてプログラムし、1つのエンドポイントをアイソクロナス Out エンドポイントとして使用します。

アイソクロナス アプリケーションは、システムを初期化して USB および UART デバイスを有効にした後、ホストからのデバイス列挙コマンド (SET/GET 記述子) に対応します。列挙が完了すると、デバイスがマイクロフォンとして構成されている場合は、次の手順が実行されます。

1. DMA を介して、USB デバイス内に存在する ping/pong バッファにデータが書き込まれます。
2. USB デバイスは、ホスト デバイスから In トークンを受信すると、ping/pong バッファ内に存在するデータで応答します。

3. 対応するエンドポイントのバッファ解放状態を示す割り込みが、プロセッサ リファレンス システムに送信されます。
4. プロセッサ リファレンス システムはバッファ解放割り込みを検出し、次に送信するデータで ping/pong バッファを充填します。
5. ホストがデータを送信する限り、この手順が繰り返されます。

デバイスがスピーカーとして構成されている場合は、次の手順が実行されます。

1. ホスト デバイスは、Out トークンに続いてデータ パケットを送信します。
2. データを受信すると、受信データは ping/pong バッファに書き込まれます。
3. 対応するエンドポイントのバッファ フル状態を示す割り込みが、プロセッサ リファレンス システムに送信されます。
4. プロセッサ リファレンス システムはデータを読み出し、DMA を介して DDR にデータを保存します。
5. ホストがデータを送信する限り、この手順が繰り返されます。

---

## 必要な環境

### ハードウェア

このリファレンス システムに必要なハードウェア ボードと追加の機器には、次のものがあります。

- ザイリンクス KCU105 プロダクション ボード
- LPC コネクタ付きのザイリンクス USB PHY FMC カード
- JTAG プラットフォーム ケーブル USB
- USB mini-B ケーブル (1 本)
- マイクロ USB ケーブル (1 本)

### ソフトウェア

このリファレンス システムの構築およびダウンロードには、次の設計ツールをインストールしておく必要があります。

- ザイリンクス Vivado Design Suite 2015.1
- ザイリンクス ソフトウェア開発キット 2015.1
- COM ポートを介す UART シリアル通信向け Tera Term/putty ターミナル エミュレーター

### ホストレベルのソフトウェア要件

ホスト側の USB バルク パフォーマンス モニターのソフトウェア要件は次のとおりです。

- Windows XP (SP3) 以上 (32 ビットまたは 64 ビット)
- NET Framework 4

---

## リファレンス デザイン ファイル

リファレンス デザイン ファイルは、次のリンクからダウンロードできます。

<https://secure.xilinx.com/webreg/clickthrough.do?cid=393319>

注記：リファレンス デザイン ファイルへのアクセスには、登録が必要です。

図 2 に、このアプリケーション ノートに付属するデザイン ファイルのディレクトリ構造を示します。

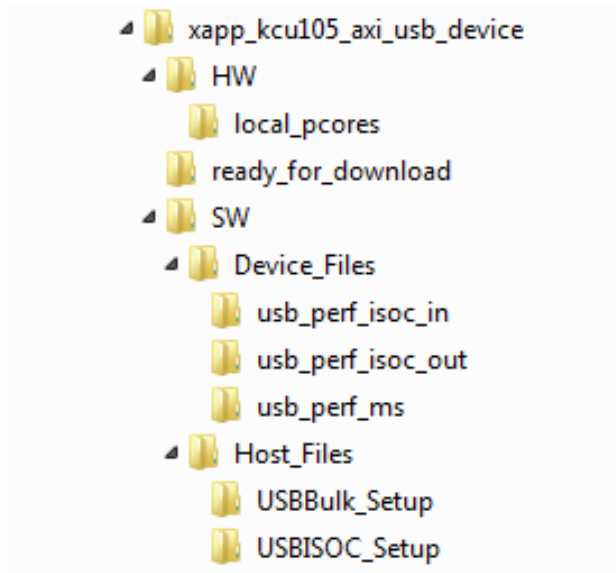


図 2：ディレクトリ構造

xapp1263\_kcu105\_axi\_usb\_device：最上位フォルダーです。

HW：このフォルダーには、local\_pcores フォルダー、all.tcl、および kcu105.xdc の各ファイルが含まれます。

- local\_pcores：このフォルダーは空であり、システムでローカル pcore が使用されている場合はその pcore をこのフォルダーに格納できます。
- all.tcl：このファイルには、Tcl コマンドを使用する、axi USB デバイス用の MicroBlaze ベースの完全なハードウェアシステムが含まれます。
- kcu105.xdc：このファイルには、AXI USB、GPIO、および UART Lite IP に対してシステムで必要な LOC 制約が含まれます。これらはピンを介して外部パリティフェラルへ接続する際に使用します。

Ready\_for\_download：このフォルダーには、design\_1\_wrapper.bit、isoc\_in\_perf.elf、isoc\_out\_perf.elf、および bulk\_perf.elf ファイルが含まれます。design\_1\_wrapper.bit は、FPGA XMD コマンドを使用してプログラムする必要があります。isoc\_in\_perf.elf、soc\_out\_perf.elf、および bulk\_perf.elf ファイルのダウンロードには、XMD コマンドの dow を使用します。

SW：このフォルダーには、Device\_Files および HOST\_Files フォルダーが含まれます。

- Device\_files：このフォルダーには、システムが USB のパフォーマンスをチェックするために使用する bulk\_perf、isoc\_in\_perf、および isoc\_out\_perf アプリケーションが含まれます。
- HOST\_files：このファイルには、USB のパフォーマンス グラフを監視するために HOST PC にインストールされるアプリケーションが含まれます。



# リファレンス デザインの実行手順

## ハードウェアのセットアップ

図 3 に、スループットの測定を目的としたハードウェア セットアップを示します。

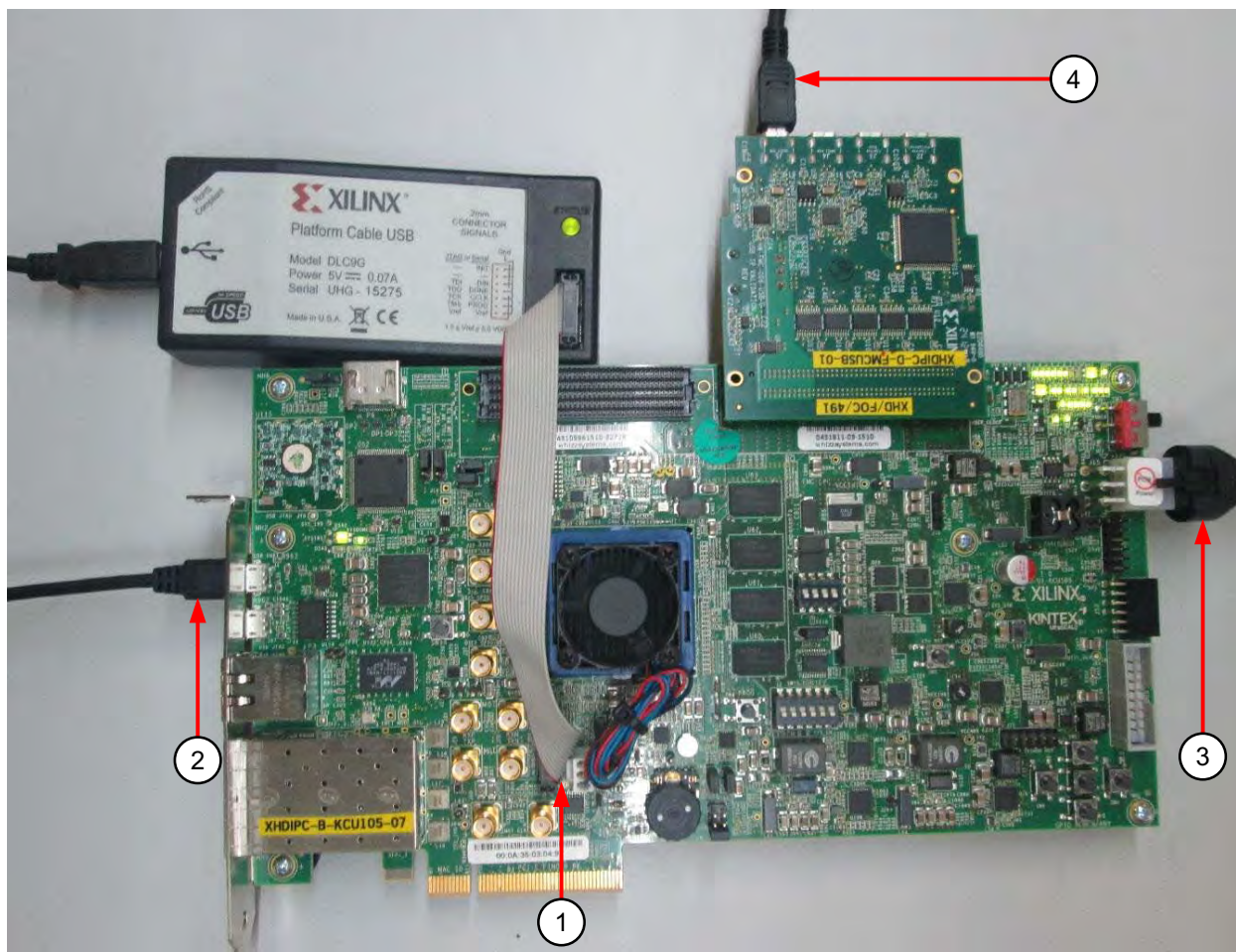


図 3 : KCU105 ボードを使用するハードウェアのセットアップ

1. BIT ファイルと ELF ファイルをプログラムするために、JTAG USB プラットフォーム ケーブルまたは USB Type A/Micro B ケーブルをホスト PC から KCU105 ボードへ接続します。(このセットアップでは JTAG プラットフォーム ケーブルを使用)
2. ホスト PC と KCU105 ボードのシリアル通信用 USB UART ポートを Micro USB ケーブルで接続します。
3. ボードに電源を投入し、KCU105 ボードの LPC ポート (J65) に USB PHY FMC カードを接続します。

**注記** : スロットにすでに FMC カードが挿入されている Ultrascale ボードに電源を投入した場合、FMC スロットは電力を受信しません。この問題の回避策として、先にボードに電源を投入してから、スロットに FMC カードを挿入してください。詳細は、UG917 [参照 3] の「FMC VADJ\_1V8 電源レール」を参照してください。

4. ホスト アプリケーションを実行するシステムに、FMC カードの J65 USB ポートを接続します。図 3 にボードのセットアップを示します。
5. ホスト PC 上で PuTTY プログラムを、次の設定で起動します。
  - ボーレート : 9600
  - データビット : 8

- パリティ:なし
- ストップ ビット:1
- フロー制御:なし

---

## リファレンス デザインの実行

このセクションでは、ハードウェアおよびソフトウェアの両方についてリファレンス デザインを構築する方法を説明します。

最初にリファレンス デザイン ファイルをローカル ディレクトリに解凍します。リファレンス デザイン (BIT および ELF ファイルの生成を含む) を構築し、そのデザインをハードウェアでテストすることも可能ですが、このリファレンス デザイン ファイルに含まれている生成済みの BIT および ELF ファイルを使用してテストすることも可能です。

次の手順を実行して、リファレンス デザインの構築およびハードウェア上でのテストを行います。

- リファレンス デザインを構築する
- ハードウェア上でデザインを実行する

リファレンス デザイン ファイル内にある生成済みの BIT および ELF ファイルを使用してデザインを実行し、ハードウェア上でテストする場合には、手順 2 の「ハードウェア上でデザインを実行する」の「重要」コメントに従う必要があります。

### リファレンス デザインを構築する

以降のセクションでは、ハードウェア デザインを生成し (BIT および ELF ファイルの生成を含む)、SDK ワークスペースを構築する際の手順を説明します。

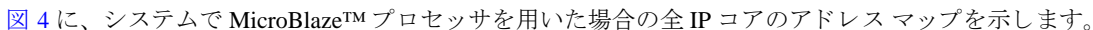
### Vivado デザイン ツール プロジェクトを作成し、ビットストリームを生成する

ここでは、Vivado Design Suite 2015.1 で新規プロジェクトを開始する手順について説明します。

1. Vivado Design Suite を起動します。
2. Vivado 統合設計環境 (IDE) で [Tcl Console] 開きます (または [Window] → [Tcl Console] をクリック)。
3. Vivado IDE の [Tcl Console] で、HW ディレクトリへ変更します。

```
cd xapp1263_axi_usb_device/HW
```
4. source コマンドで、all.tcl ファイルを実行します  

```
source all.tcl
```
5. プロジェクトが作成されて、出力ファイルが生成されます。デザインは合成およびインプリメントされて、ビットストリームが生成されます。

 **図 4** に、システムで MicroBlaze™ プロセッサを用いた場合の全 IP コアのアドレス マップを示します。

**注記:** リファレンス デザインで提供されている、生成済みの BIT および ELF ファイルを使用する場合は、このセクションを飛ばして「ハードウェア上でデザインを実行する」へ進んでください。

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
microblaze_0					
Data (32 address bits : 4G)					
lmb_bram_if_cntlr_0	SLMB	Mem	0x0000_0000	128K	0x0001_FFFF
axi_bram_ctrl_0	S_AXI	Mem0	0xC000_0000	128K	0xC001_FFFF
axi_bram_ctrl_1	S_AXI	Mem0	0xC200_0000	128K	0xC201_FFFF
mig_0	CO_DDR4_S_AXI	CO_DDR4_AD...	0x8000_0000	1G	0xBFFF_FFFF
axi_intc_0	S_AXI	Reg	0x4120_0000	64K	0x4120_FFFF
axi_uartlite_0	S_AXI	Reg	0x4060_0000	64K	0x4060_FFFF
axi_timer_0	S_AXI	Reg	0x41C0_0000	64K	0x41C0_FFFF
axi_usb2_device_0	S_AXI	Reg	0x4480_0000	64K	0x4480_FFFF
Instruction (32 address bits : 4G)					
lmb_bram_if_cntlr_1	SLMB	Mem	0x0000_0000	128K	0x0001_FFFF
axi_bram_ctrl_0	S_AXI	Mem0	0xC000_0000	128K	0xC001_FFFF
axi_bram_ctrl_1	S_AXI	Mem0	0xC200_0000	128K	0xC201_FFFF
mig_0	CO_DDR4_S_AXI	CO_DDR4_AD...	0x8000_0000	1G	0xBFFF_FFFF
axi_usb2_device_0					
DMA_DATA (32 address bits : 4G)					
axi_bram_ctrl_0	S_AXI	Mem0	0xC000_0000	128K	0xC001_FFFF
axi_bram_ctrl_1	S_AXI	Mem0	0xC200_0000	128K	0xC201_FFFF
mig_0	CO_DDR4_S_AXI	CO_DDR4_AD...	0x8000_0000	1G	0xBFFF_FFFF

図 4 : IP インテグレーターにおける IP コアのベース アドレスと高位アドレス

## SDK でハードウェアワークスペースをエクスポートする

次の手順に従って、バルク トランザクションの ELF ファイルを作成します。アイソクロナス In および Out トランザクションの ELF ファイルも同じ手順で作成できます。ソース ファイルは、SW/Device\_Files ディレクトリの bulk\_perf、isoc\_in\_perf、および isoc\_out\_perf フォルダーにあります。

1. ビットストリーム生成完了後、SDK ですでにエクスポートされているハードウェアワークスペースを開きます。
2. xsdk GUI を開きます。
3. ワークスペースパスを「project\_1/project\_1.sdk」とし、[OK] をクリックします。
4. SDK に BSP を追加します。[File] → [New] → [Board Support Package] をクリックします。
5. 新規プロジェクトを作成します。
  - a. [File] → [New] → [Application project] をクリックします。
  - b. 新しいプロジェクト名を入力します。例 : bulk\_perf
  - c. BSP 用に [Use Existing] を選択します。[Next] をクリックします。
  - d. 空のプロジェクトを選択して [OK] をクリックします。
  - e. [Project Explorer] で、bulk\_perf プロジェクトの下の [src] フォルダーを選択します。
  - f. SRC ディレクトリを右クリックし、[Import] をクリックします。
  - g. [General] → [File System] をクリックし、[Next] をクリックします。
  - h. SW/Device\_Files/bulk\_perf ディレクトリを指定し、[OK] をクリックします。
  - i. 次の SW ファイルを選択してインポートし、[Finish] をクリックします。
    - xusb\_cp9.c
    - xusb\_cp9.h
    - xusb\_storage.c
    - xusb\_storage.h
    - xusb\_types.h
  - j. elf ファイルが生成されます。

上記の手順は、isoc\_in\_perf および isoc\_out\_perf アプリケーション用 ELF の生成に応用できます。



## ホスト側ドライバーをインストールする

このセクションでは、ホスト側ドライバーのインストール方法を説明します。

バルク トランザクションとアイソクロナス トランザクションのパフォーマンスは、別々のアプリケーションで測定されません。アイソクロナス トランザクションのパフォーマンスを測定するには、`SW/Host_Files/USBISOC_Setup` ディレクトリに移動します。バルク トランザクションのパフォーマンスを測定するには、`SW/Host_Files/USBulk_Setup` ディレクトリに移動します。パフォーマンス測定には、両方のアプリケーションを別々にインストールする必要があります。

32 ビット マシンでは `Xilinx USB Perfmon Setup x86.msi`、64 ビット マシンでは `Xilinx USB Perfmon Setup x64.msi` をダブルクリックします。手順に従ってインストールを完了します。

**注記** : 64 ビット版 Windows 7 では、PC を起動し、Windows が起動を開始したら、F8 キーを押しながら [Advanced Boot Options] を開きます。[Advanced Boot Options] プロンプトで、[Disable Driver Signature Enforcement] を [No] に設定して無効にします。

**注記** : ボードが \*.bit ファイルでコンフィギュレーションされ、\*.elf ファイルが読み込まれていることを確認してから、ドライバーをインストールします。

アプリケーションが動作しない場合は、もう一度 MicroBlaze プロセッサ アプリケーションを読み込みます。アプリケーションが問題なく動作するまで、数回 (最大で 2 ~ 3 回) 読み込まなければならないことがあります。

## ハードウェア上でデザインを実行する

次の手順に従って、ハードウェア セットアップでビットストリームおよび ELF ファイルを実行します。

1. JTAG ケーブルと USB-UART ケーブルをボードに接続します。
2. test フォルダを作成します。



**重要** : リファレンス デザインで提供されている生成済みの BIT および ELF ファイルへアクセスする場合は、test フォルダを作成する代わりに、`ready_for_download` フォルダを使用します。手順 3、4、5 を飛ばして、手順 6 から開始してください。

3. 次の場所から test フォルダに BIT ファイルをコピーします。  
`project_1/project_1.runs/impl_1/design_1_wrapper.bit`
4. 次の場所から test フォルダに ELF ファイルをコピーします。  
`project_1/project_1.sdk/bulk_perf/Debug/bulk_perf.elf`  
`project_1/project_1.sdk/isoc_in/Debug/isoc_in.elf`  
`project_1/project_1.sdk/isoc_out/Debug/isoc_out.elf`
5. コマンド プロンプトに次のコマンドを入力して test フォルダを開きます。  
`cd test`
6. コマンド プロンプトに `xmd` と入力して、Xilinx Microprocessor Debugger (XMD) を起動します。
7. XMD プロンプトに次のコマンドを入力して、JTAG ケーブルを介して `design_1_wrapper.bit` で FPGA をコンフィギュレーションします。  
`fpga -f design_1_wrapper.bit`
8. XMD プロンプトに次のコマンドを入力して、FPGA 上で実行されるプロセッサへ接続します。  
`connect mb mdm`
9. XMD プロンプトに次のコマンドを入力して、FPGA をリセットおよび停止します。  
`rst`  
`stop`
10. 結果を確認するためにハイパーターミナルを起動し、デフォルト コンフィギュレーションでボー レートを 9600 に設定します。ボードと PC に UART ケーブルが接続されていることを確認してください。

## バルク USB としてボードを構成する手順

11. ELF ファイルをメモリ (ブロック RAM または DDR) にダウンロードし、ソフトウェアをボードで実行します。

```
stop; rst; dow bulk_perf.elf; run
```

12. 「USB enumerated」と表示されない場合は、前の手順のコマンドをもう一度実行します。

## アイソクロナス USB としてボードを構成する手順

13. Isoc-in 用の XMD ターミナルで次のコマンドを入力します。

```
rst;stop;dow isoc_in.elf;run
```

ハードウェアは In モードのアイソクロナス デバイスとして使用できる状態です。

14. Isoc-out 用の XMD ターミナルで次のコマンドを入力します。

```
rst;stop;dow isoc_out.elf;run
```

15. デバイスの準備が完了したら、UsbPerform.exe を使用してアプリケーションを実行できます。構成全体が完了したら、FMC カードに接続されているポート以外のすべての USB ポートを注意して削除します。UsbPerform.exe を実行します。

注記：開始レートは 1024K 以上にしてください。USB ハブ上のトランザクションによってレート/グラフは変化します。通常は変化の幅を 10% 以内にします。ただし、新しい USB デバイスをシステムに接続した場合、この値は大きく変化することがあります。

## 結果

ドライバーがインストールされ、システムが検出された状態で UsbPerfmon.exe を実行すると、[図 5](#) のような USB パフォーマンス モニターが表示されます

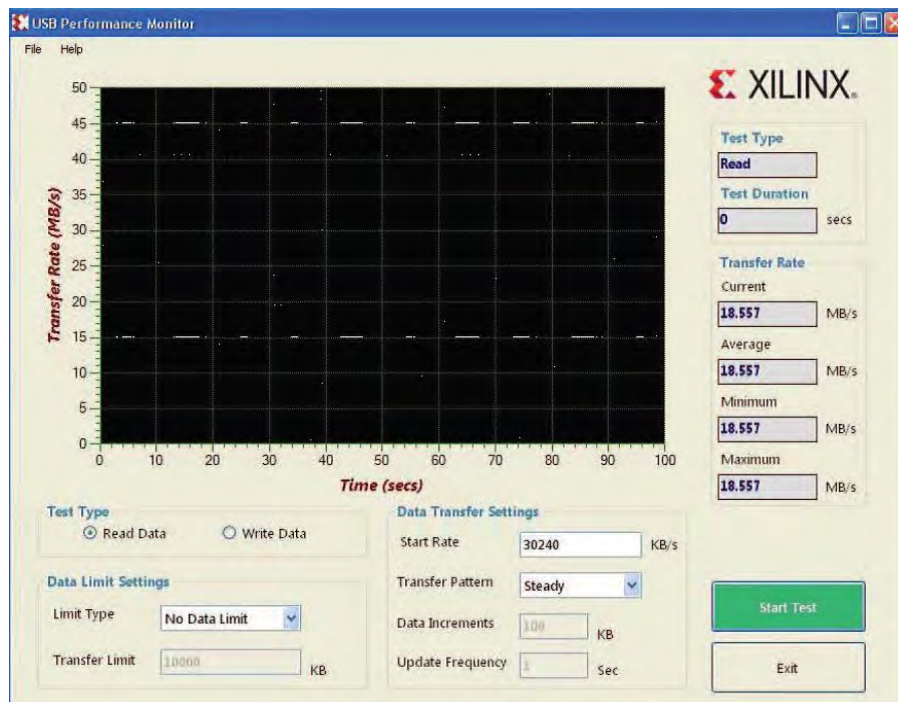


図 5: USB パフォーマンス モニター

パフォーマンス測定ソフトウェアでは、GUI を使用して転送レートとテスト タイプを変更できます。変更可能なオプションは、ディスプレイの下部にあります。出力はディスプレイの右側に表示されます。設計者は次の項目を変更できます。

- [Test Type]: テストは [Read] または [Write] モードで実行できます。読み出しトランザクションと書き込みトランザクションのスループットは別々に測定できます。
- [Data Limit Settings]: 1 トランザクションあたりの最大伝送データは、このタブで制限できます。
- [Data Transfer Settings]: 転送レートの設定は、このタブで変更できます。[Start Rate] は、ホストからのデータ転送のスタート レートを指定します。転送レートの変化パターンは、[Transfer Pattern] で設定できます。

## バルク トランザクションのパフォーマンス測定

読み出しと書き込みのパフォーマンスは別々に測定できます。書き込み操作については、次の GUI パラメーターを指定してパフォーマンス測定を実行します。

- Start Rate = 40,024 KB/s
- Transfer Pattern = Steady

この設定で、AXI USB 2.0 デバイスを使用して 18.005 の最大パフォーマンスが得られました (図 6 参照)。



図 6: バルク トランザクションのパフォーマンス測定 (スタート レート 40024KB/s)

図 6 の [Transfer Rate] に示すように、現在の測定値 ([Current]) は、そのインスタンスで (1 秒間) 測定したパフォーマンスを示します。平均測定値 ([Average]) は、その時点までに計算されたパフォーマンスの平均値を示します。最小測定値 ([Minimum]) は、時間内に測定されたパフォーマンスの最小値を示します。最大測定値 ([Maximum]) は、その秒までに測定されたパフォーマンスの最大値を示します。

図 7 に、次の GUI パラメーター設定での結果を示します。

- Start Rate = 1,024KB/s
- Transfer Pattern = Fixed Increment
- Data increments = 1,000KB
- Update frequency = 1sec (データ レートを引き上げる間隔)

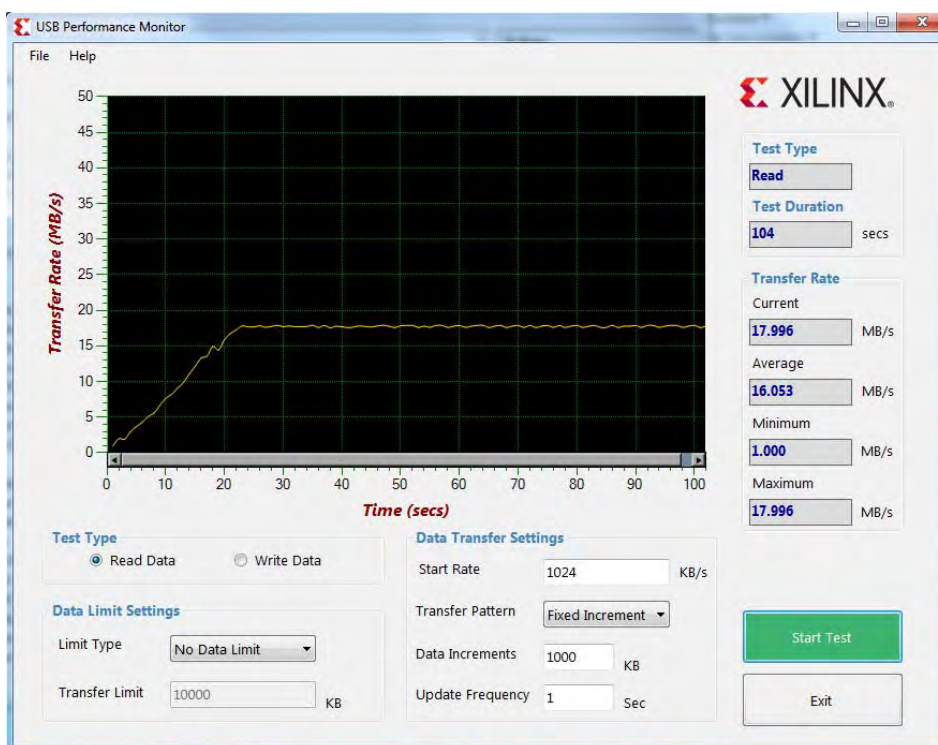


図 7: バルクトランザクションのパフォーマンス測定 (1024 KB/s)

注記: パフォーマンスの数値は、システムに接続されるその他の USB デバイスの数と、ホストが 1 マイクロフレームで送信できるパケット数によって異なります。



## アイソクロナス トランザクションのパフォーマンス測定

図 8 に、開始データレートが 1,024KB/s で、固定増分パターンを使用したアイソクロナス Out トランザクションのパフォーマンス測定を示します。達成された最大スループットは 23.067MB/s で、理論上の最大スループットは 24MB/s です。



図 8: アイソクロナス Out トランザクションのパフォーマンス測定

図 9 に、スタートデータレートが 1,024KB/s で、固定増分パターンを使用したアイソクロナス In トランザクションのパフォーマンス測定を示します。達成された最大スループットは 22.906MB/s で、理論上の最大スループットは 24MB/s です。



図 9: アイソクロナス In トランザクションのパフォーマンス測定



## リファレンス デザインの詳細

リファレンス デザイン ファイルは、次のリンクからダウンロードできます。

<https://secure.xilinx.com/webreg/clickthrough.do?cid=393319>

リファレンス デザイン ファイルへのアクセスには、登録が必要です。

### AXI-USB リファレンス デザイン

表 1 に、このアプリケーション ノートのリファレンス デザインの詳細を示します。

表 1: リファレンス デザインの詳細

パラメーター	説明
一般	
開発者	Prasad Gutti
ターゲット デバイス	Kintex UltraScale FPGA XCKU040-FFVA1156-2-E
ソース コードの提供	あり
ソース コードの形式	VHDL、Verilog
既存のザイリンクス アプリケーション ノート/ リファレンス デザイン、またはサードパーティから デザインへのコード/IP の使用	SDK 用に提供されたリファレンス デザインおよび Vivado IP カタログから生成されるコア
シミュレーション	
論理シミュレーションの実施	N/A
タイミングシミュレーションの実施	N/A
論理シミュレーションおよびタイミング シミュ レーションでのテストベンチの利用	N/A
テスト ベンチの形式	N/A
使用したシミュレータ/バージョン	N/A
SPICE/IBIS シミュレーションの実施	N/A
インプリメンテーション	
使用した合成ツール/バージョン	Vivado Design Suite : System Edition 2015.1
使用したインプリメンテーション ツール/バー ジョン	Vivado Design Suite : System Edition 2015.1
スタティック タイミング解析の実施	あり (PAR/TRCE のタイミングにパス)
ハードウェア検証	
ハードウェア検証の実施	あり
使用したハードウェア プラットフォーム	Kintex UltraScale FPGA KCU105 評価キット

## 参考資料

このアプリケーション ノートの参考資料は次のとおりです。

注記：日本語版のバージョンは、英語版より古い場合があります。

- 『LogiCORE IP AXI Universal Serial Bus (USB) 2.0 デバイス v5.0 製品ガイド』(PG137)
- 『Vivado Design Suite ユーザー ガイド：IP インテグレーターを使用した IP サブシステムの設計』(UG994：[英語版](#)、[日本語版](#))
- 『KCU105 ボード ユーザー ガイド』(UG917)

## 改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2015 年 8 月 17 日	1.0	初版

## 法的通知

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、および全て受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、全ての保証および条件を負わない (否認する) ものとし、また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、<http://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照して下さい。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うこととなります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。<http://japan.xilinx.com/legal.htm#tos> で見られるザイリンクスの販売条件を参照して下さい。

© Copyright 2015 Xilinx, Inc. Xilinx, Xilinx のロゴ、Artix、ISE、Kintex、Spartan、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他の各国のザイリンクス社の商標です。全てのその他の商標は、それぞれの保有者に帰属します。

この資料に関するフィードバックおよびリンクなどの問題につきましては、[jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com) まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。