



XAPP1234 (v1.0) 2015 年 2 月 11 日

AXI Quad SPI IP コアのスループット性能測定

著者 : Prasad Gutti

概要

このアプリケーション ノートでは、AXI Quad SPI IP コアをデュアルおよびエンハンスド クワッド モードで使用し、SPI フラッシュ メモリから 1MB のデータを読み書きする場合の SPI 帯域幅測定について説明します。また、Kintex® UltraScale™ FPGA 用ザイリンクス KCU105 評価ボードと Numonyx SPI メモリを使用して得られた測定結果を示します。サンプルソフトウェア ファイルを数か所変更すれば、ほかのボードでも測定可能です。

リファレンス デザイン システムは、Vivado Design Suite : System Edition 2014.4 に付属する Vivado® IP インテグレーターによって構築されます。IP インテグレーターでは、プロセッサ、インターコネクト、割り込みコントローラー、ペリフェラル IP コア、メモリ コントローラー、UART をインスタンス化することでシステムを構築できます。デザインには、ザイリンクスのソフトウェア開発キット (SDK) を使用して構築されたソフトウェアも含まれます。このソフトウェアは、MicroBlaze™ プロセッサ サブシステム上で動作し、制御、ステータス、モニターの各機能を実行します。リファレンス デザインの ZIP ファイルには IP インテグレーターおよび SDK の完全なプロジェクト ファイルが含まれており、これらをデザインの検討や再構築に活用したり、新規デザインのテンプレートとして使用することが可能です。

このアプリケーション ノートは、クワッド SPI システムのインプリメント方法、および `axi_timer` を使用してすべてのモードにおける QSPI のスループットと帯域幅を測定する方法を説明します。ここでは、コアはエンハンスド クワッド、デュアル、およびスタンダード SPI モードに設定され、SPI クロック レートは 100MHz です。また、クワッド モードで得られるスループット値がデュアル モードで得られる値のほぼ 2 倍になることも示します。

はじめに

AXI Quad SPI コアは、レガシ、エンハンスド、XIP の各モードをサポートします。これら 3 つのモードはさらに、スタンダード、デュアル、クワッドという 3 つの SPI モードに分類されます。スタンダード モードのコマンドは 1 本のライン (IO1)、デュアル モードは 2 本のライン (IO0、IO1)、クワッド モードは 4 本のライン (IO0、IO1、IO2、IO3) を使用してデータの読み書きを実行します。レガシ モードは、以前のバージョン (v1_00a) のコアをベースとするアプリケーションをサポートします。エンハンスド モードは AXI4 メモリ マップド インターフェイスをサポートし、送信および受信 FIFO の固定長バースト機能をサポートします。エンハンスド モードは、DTR または DRR FIFO のフィルまたは読み出しに必要な AXI インターフェイス時間を削減します。これらの FIFO はコンパイル時に設定を変更でき、深さは 16 または 256 エレメントのいずれかを選択可能です。表 1 に、各モードで使用する AXI4 インターフェイスを示します。

表 1 : AXI Quad SPI コアの各コンフィギュレーション モードで使用する AXI4 インターフェイス

| モード | AXI4-Lite インターフェイス | AXI4 インターフェイス |
|------------|--------------------|---------------|
| レガシ モード | あり | - |
| エンハンスド モード | - | あり |
| XIP モード | あり | あり |

使用する SPI スレーブの種類に応じて、3 つの SPI モードのうちいずれか 1 つが選択されます。表 2 に、SPI モードとサポートされる SPI クロック周波数および対応する I/O インターフェイスを示します。

表 2: SPI モードと SCK 分周比および I/O インターフェイス

| SPI モード | SPI クロック分周比 wrt.EXT_SPI_CLK | I/O インターフェイス (CS と SCK は常に存在) |
|---------|--------------------------------|----------------------------------|
| スタンダード | 2、4、8、16xn (ただし n = 1 ... 128) | IO0、IO1 |
| デュアル | 2 | IO0、IO1 |
| クワッド | 2 | IO0、IO1、IO2、IO3 |

SPI 側の帯域幅を最大にするには、4 本のラインのすべてでデータ トランザクションが行われるクワッド モードでコアを使用してください。このモードは、Fast Read Quad Output (0x6B h)、Fast Read Quad I/O (0xEB h)、Quad Input Page Program (0x32 h) コマンドをサポートするため、SPI 帯域幅が最大限に活用できます。これらのコマンドはいずれも 4 本の I/O ラインによる SPI フラッシュ メモリへの読み書きに対応しています。

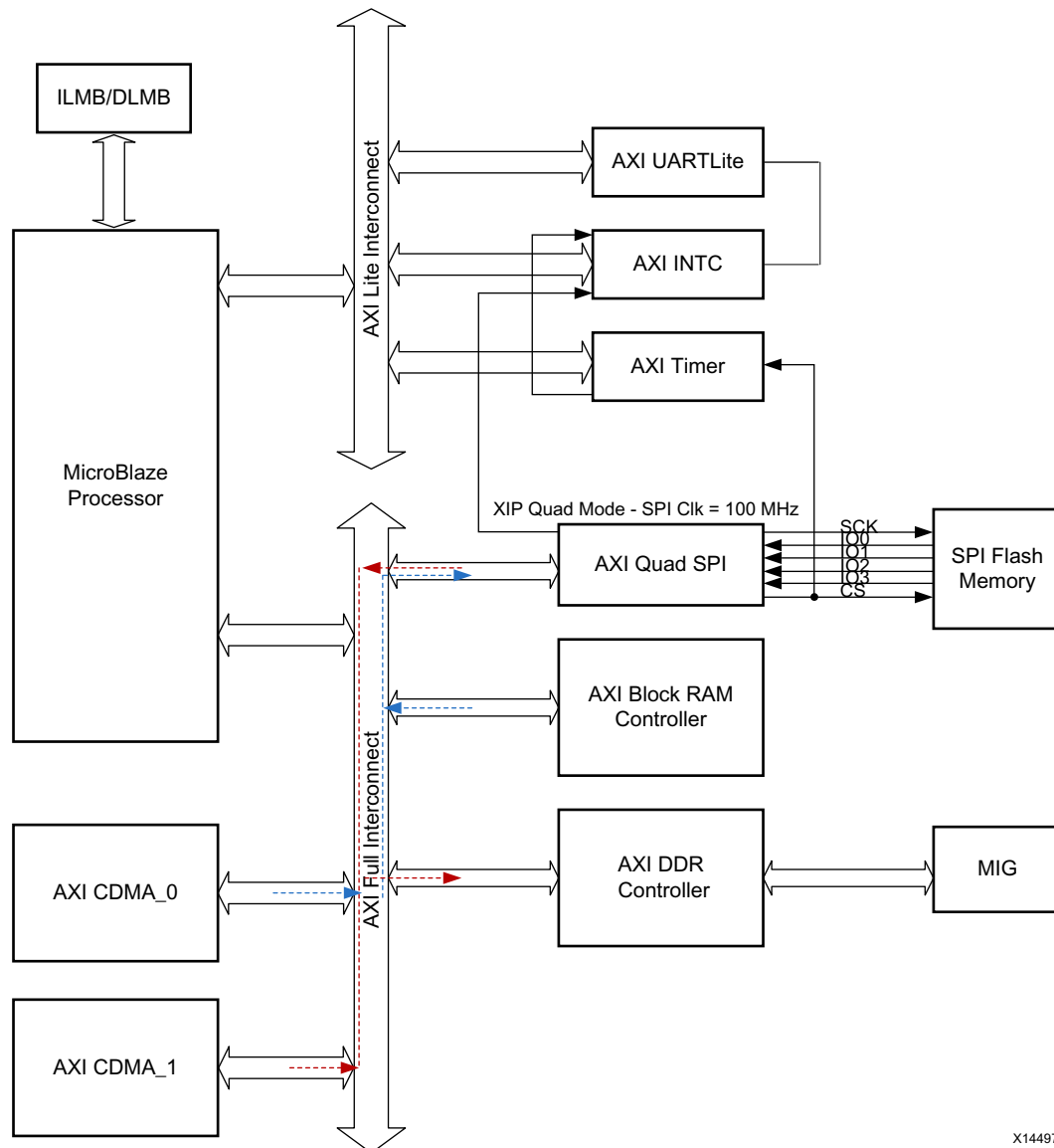
リファレンス デザイン

SPI フラッシュ メモリから 1MB を読み出す場合のシステム スループットを調べるには、性能が最適となるようコアをクワッド モードに設定する必要があります。Winbond および Numonyx の SPI メモリは、現在ではスタンダード モードよりもデュアルおよびクワッド モードの使用を推奨しています。スタンダード モードに比べて、デュアル モードのコマンドは SPI 帯域幅を約 2 倍、クワッド モードのコマンドは約 4 倍改善します。

AXI Quad SPI v3.2 コアでは SPI 帯域幅が損失なく使用されます。これは、送信データ FIFO にデータが存在する限り、SPI クロックが動作を継続することを意味します。DTR FIFO が半分空いた時点でデータを再度フィルしたり、空きスペースがないか DTR Occupancy レジスタをポーリングしたりするなど、効率的なソフトウェア コーディング手法を採用すれば SPI トランザクション間にアイドル サイクルは発生しません。このような手法は、エンハンスド クワッド モードに加えてレガシ モードにも適用できます。バースト機能の追加により、1MB のデータの読み書きに要する時間は、レガシ クワッド モードに比べてエンハンスド クワッド モードの方が短くなります。今回の測定では、すべて KCU105 ボード上の Numonyx メモリをアクセス先とし、100MHz の SPI クロックを使用しています。

ハードウェア

図 1 に、AXI Quad SPI スループット測定システムのデザインを示します。



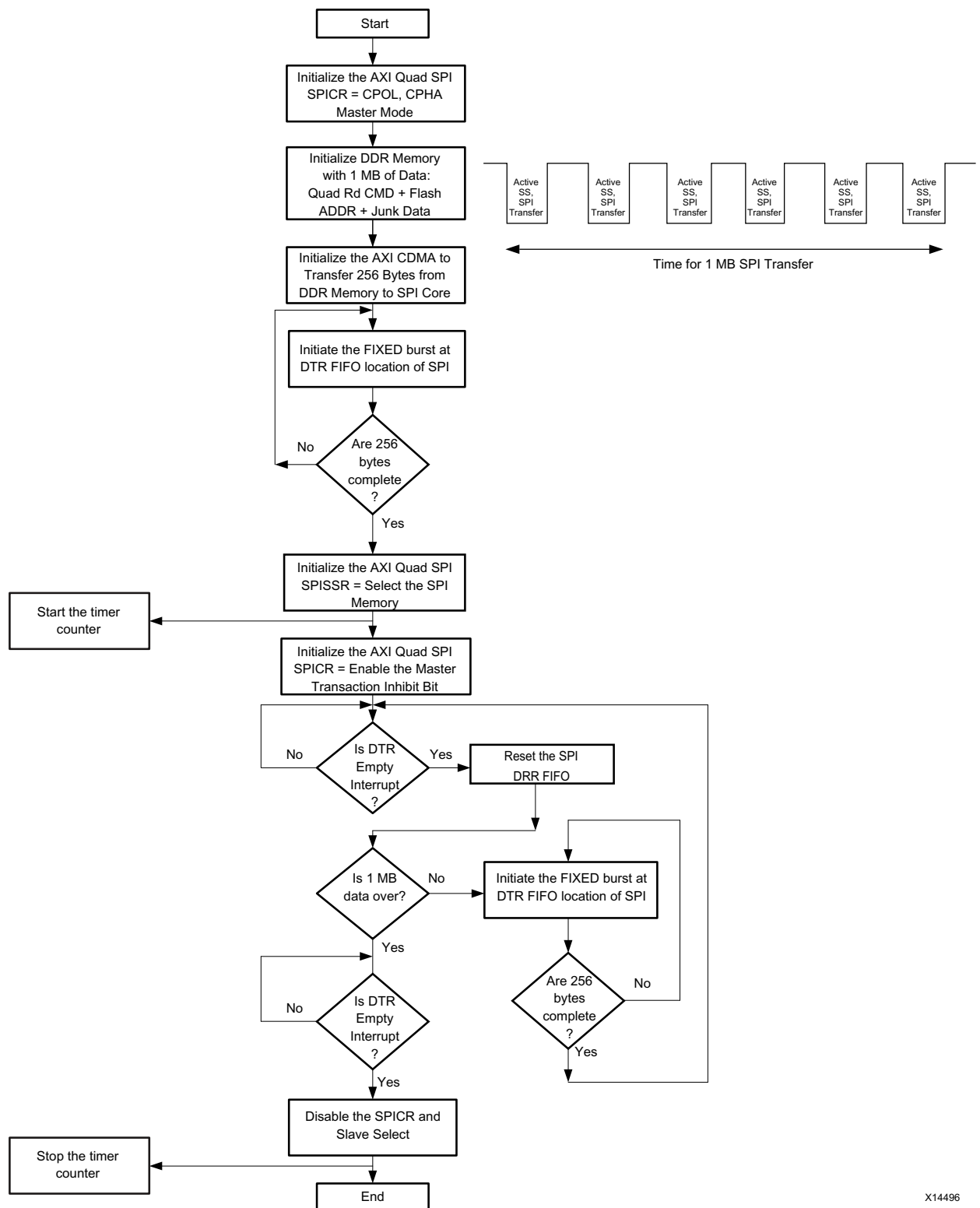
X14497

図 1: スループット測定システム デザイン - エンハンスド クワッド モード

ソフトウェア システム

アプリケーション ノートのソフトウェア デザインには、書き込みスループットと読み出しスループットの両測定が含まれます。

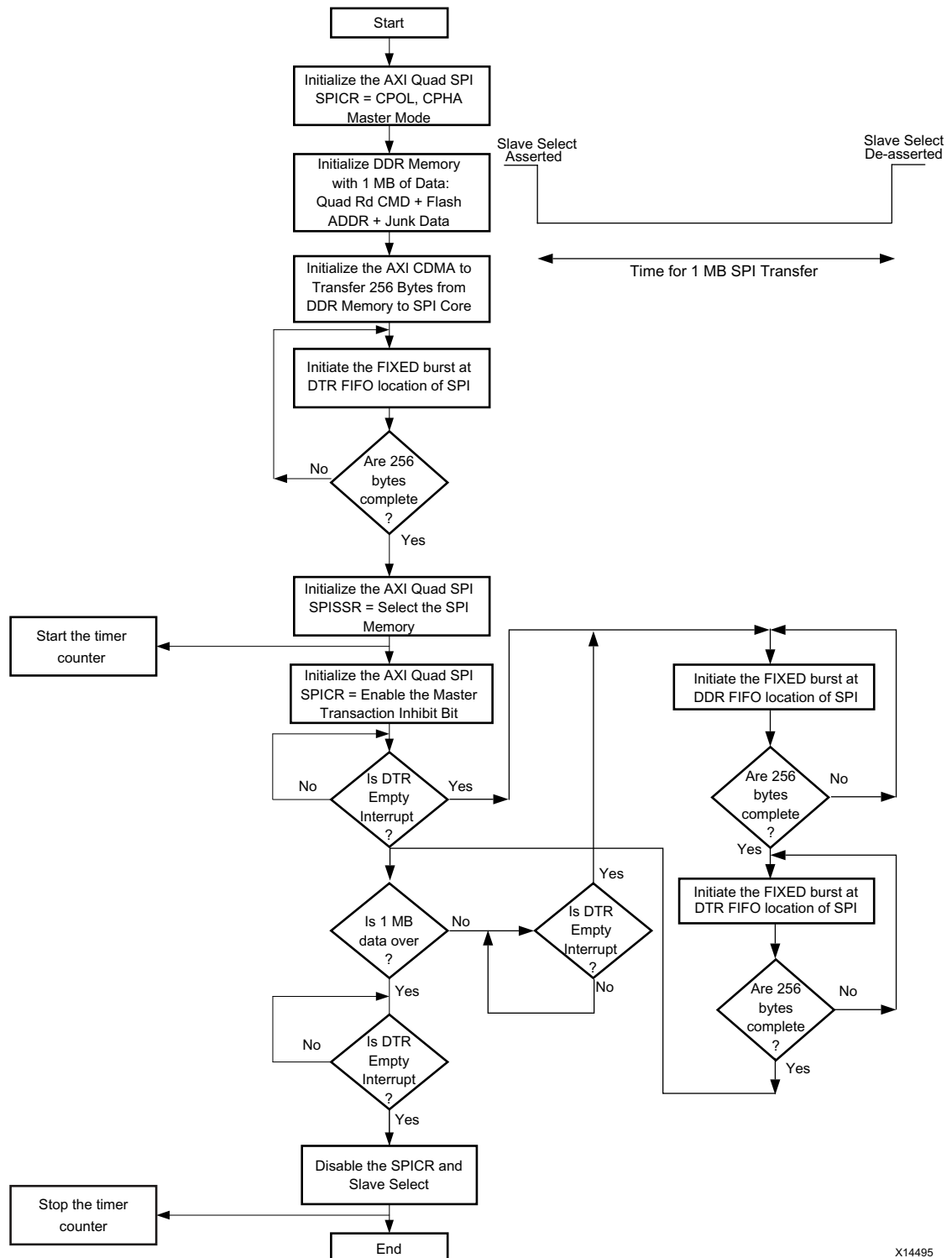
図 2 は、AXI4 メモリ マップド インターフェイスを含むシステムでスループットを測定するために、SPI フラッシュ メモリに 1MB のデータを書き込む場合を示しています。



X14496

図 2: ソフトウェア フローチャート - エンハンスド クワッド モード書き込みトランザクション

図 3 に、SPI フラッシュ メモリから 1MB のデータを読み出す場合のソフトウェア フローチャートを示します。



X14495

図 3 : ソフトウェア フローチャート - エンハンスド クワッド モード読み出しトランザクション

書き込みおよび読み出しのプロセスの相違

すべての SPI フラッシュ デバイスは 8 ビット インターフェイス モードを備えています。実際のデータ伝送が開始されるのは、コマンド、アドレスバイト、ダミーバイトが送信された後です。このデータは 8 ビット パケット モードで伝送されます。フラッシュ書き込みはシングルページに限られる一方、読み出しにはそのような制約がありません。チップ セレクトがアサートされ、SPI クロックがアクティブである限り、SPI フラッシュはホストへのデータ送信を継続します。これは、読み出しとは異なり、SPI デバイスの書き込みプロセスが比較的低速であることを意味します。

ツール フローおよび検証

表 3 に、付属のリファレンス デザインで使用されるツール フローおよび検証手順を示します。

表 3: リファレンス デザインの詳細

| パラメーター | 説明 |
|---|---|
| 全般 | |
| 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 (一部コアは暗号化済み) |
| 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 |
| 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 合成 |
| 使用したインプリメンテーション ツール/バージョン | Vivado インプリメンテーション |
| 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 | KCU105 評価キット |

必要な環境

ハードウェア

このシステムに必要なハードウェア ボード、その他の機器は次のとおりです。

- ザイリンクス KCU105 評価ボード (Rev B)
- JTAG USB プラットフォーム ケーブルまたは Type-A/Micro-B USB ケーブル
- uart データ伝送用マイクロ USB ケーブル

ソフトウェア

- Vivado IP インテグレーター 2014.4
- Vivado Design Suite : System Edition 2014.4
- SDK 2014.4

リファレンス デザイン ファイル

リファレンス デザインのデザイン ファイルは、次のリンク先からダウンロードできます。

- デュアル モード リファレンス デザイン - <https://secure.xilinx.com/webreg/clickthrough.do?cid=378726>
- クワッド モード リファレンス デザイン - <https://secure.xilinx.com/webreg/clickthrough.do?cid=378727>

ダウンロードには登録が必要です。

図 4 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。このアプリケーション ノートにはエンハンスド クワッドおよびデュアル QSPI の各モードに対応する 2 つのシステムが含まれます。

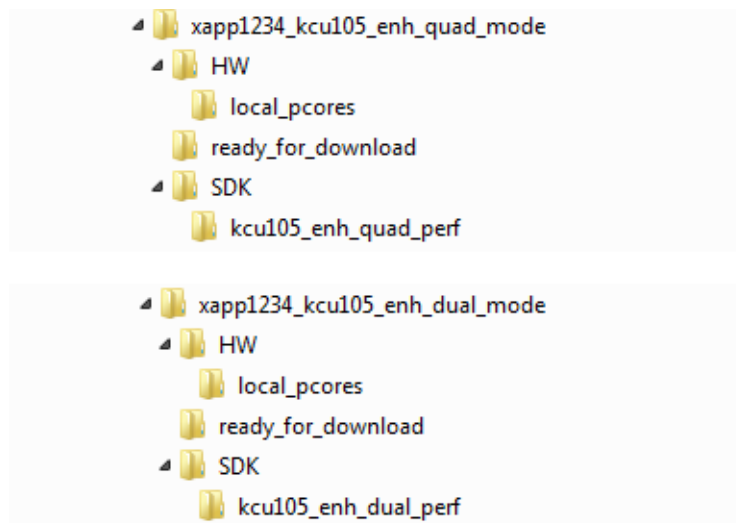


図 4: ディレクトリ構造

xapp1234_kcu105_enh_quad_mode.zip ファイルには次のファイルおよびフォルダーが含まれます。

xapp1234_kcu105_enh_quad_mode: 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、design_1_wrapper.vhd、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、クワッド SPI エンハンスド クワッド モード用の 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、ピンを介して外部ペリフェラルへ接続する際に使用する、クワッド SPI 用システムと UART Lite IP に必要な LOC 制約が含まれます。

design_1_wrapper.v: 最上位ラッパー ファイルです。このファイルには、システムの最上位ポートが含まれ、このファイルに対して STARTUPE3 がインスタンス化されています。STARTUPE3 は、クロックを SPI フラッシュメモリに転送するために使用します。

ready_for_download: このフォルダーには、design_1_wrapper_enh_quad_mode.bit および kcu105_enh_quad_perf.elf ファイルが含まれます。design_1_wrapper_enh_quad_mode.bit ファイルのダウンロードには、XMD コマンドの fpga を使用します。kcu105_enh_quad_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。

SDK: このフォルダーには、エンハンスド クワッド モードのクワッド SPI システムのスループット性能を測定するためにシステムが使用するソフトウェア ファイルが含まれます。

xapp1234_kcu105_enh_dual_mode.zip ファイルには次のファイルおよびフォルダーが含まれます。

xapp1234_kcu105_enh_dual_mode : 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、design_1_wrapper.vhd、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、クラウド SPI エンハンスド デュアル モード用の MicroBlaze ベースの完全なシステムが含まれます。

kcu105.xdc : このファイルには、ピンを介して外部ペリフェラルへ接続する際に使用する、クラウド SPI 用システムと UART Lite IP に必要な LOC 制約が含まれます。

design_1_wrapper.vhd : 最上位ラッパー ファイルです。このファイルには、システムの最上位ポートが含まれ、このファイルに対して STARTUPE3 がインスタンシエートされています。STARTUPE3 は、クロックを SPI フラッシュ メモリに転送するために使用します。クラウド SPI デュアル モード デザインでは、Quad SPI IP が SPI フラッシュ データを送受信する際、2 本の SPI_IO ラインしか使用されません。

注記 : Kintex UltraScale FPGA には制限があるため、システムでは 4 本の I/O ピンがすべて使用されています。それらのうち 2 本は IO0 と IO1 に接続され、残りの 2 本はシステムで、問題の回避を目的としたダミーとして使用されます。

ready_for_download : このフォルダーには、design_1_wrapper_enh_dual_mode.bit および kcu105_enh_dual_perf.elf ファイルが含まれます。design_1_wrapper_enh_dual_mode.bit ファイルのダウンロードには、XMD コマンドの fpga を使用します。kcu105_enh_dual_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。

SDK : このフォルダーには、エンハンスド デュアル モードのクラウド SPI システムのスループット性能を測定するためにシステムが使用するソフトウェア ファイルが含まれます。

リファレンス デザインの手順

セットアップ

図 5 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 5: 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. 電源ケーブルを接続し、KCU105 ボードの電源を投入します。
4. 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。

1. 構築を開始する前に、リファレンス デザインをローカル ディレクトリに解凍します。この手順では、リファレンス デザインを「XAPP1234」と表記します。
2. 次に示す手順のいずれかを実行します。

- 。 リファレンス デザインを構築し (BIT および ELF ファイルの生成を含む)、ハードウェアでテストします。これを実行するには、次のセクションに記載の手順に従います。
 - 「リファレンス デザインを構築する」
 - 「ハードウェアでデザインを実行する」

または

- 。 XAPP1234 リファレンス デザインに含まれる、生成済みの BIT および ELF ファイルを使用し、ハードウェアでテストします。これを実行するには、次のセクションに記載の手順に従います。
 - 「ハードウェアでデザインを実行する」、特に手順 2 にある「重要」の記載事項に注意が必要です。

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。

注記 : XAPP1234 リファレンス デザインに含まれる 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無視し、「ハードウェアでデザインを実行する」に進んでください。

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 2014.4 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] ビューを開きます。[Tcl Console] ビューが表示されない場合は、[Window] → [Tcl Console] をクリックします。
3. [Tcl Console] に次のコマンドを入力し、HW ディレクトリに移動します。

```
cd xapp1234_kcu105_enh_quad_mode/HW
```

または

```
cd xapp1234_kcu105_enh_dual_mode/HW
```

4. 提供されている all.tcl ファイルを source コマンドで実行します。

```
source all.tcl
```

プロジェクトが作成されると、Vivado Design Suite は出力ファイルの生成、デザインの合成とインプリメンテーション、ビットストリームの生成を実行します。

図 6 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。

| Cell | Slave Interface | Base Name | Offset Address | Range | High Address |
|------------------------------------|-----------------|---------------|----------------|-------|--------------|
| microblaze_0 | | | | | |
| Instruction (32 address bits : 4G) | | | | | |
| mig_0 | C0_DDR4_S_AXI | C0_DDR4_AD... | 0x8000_0000 | 1G | 0xBFFF_FFFF |
| lmb_bram_if_cntlr_1 | SLMB | Mem | 0x0000_0000 | 64K | 0x0000_FFFF |
| axi_quad_spi_0 | AXI_FULLL | MEM0 | 0xC400_0000 | 64K | 0xC400_FFFF |
| axi_bram_ctrl_1 | S_AXI | Mem0 | 0xC200_0000 | 128K | 0xC201_FFFF |
| axi_bram_ctrl_0 | S_AXI | Mem0 | 0xC000_0000 | 128K | 0xC001_FFFF |
| Data (32 address bits : 4G) | | | | | |
| mig_0 | C0_DDR4_S_AXI | C0_DDR4_AD... | 0x8000_0000 | 1G | 0xBFFF_FFFF |
| lmb_bram_if_cntlr_0 | SLMB | Mem | 0x0000_0000 | 64K | 0x0000_FFFF |
| axi_uartlite_0 | S_AXI | Reg | 0x4060_0000 | 64K | 0x4060_FFFF |
| axi_timer_0 | S_AXI | Reg | 0x41C0_0000 | 64K | 0x41C0_FFFF |
| axi_quad_spi_0 | AXI_FULLL | MEM0 | 0xC400_0000 | 64K | 0xC400_FFFF |
| axi_intc_0 | s_axi | Reg | 0x4120_0000 | 64K | 0x4120_FFFF |
| axi_cdma_1 | S_AXI_LITE | Reg | 0x44A2_0000 | 64K | 0x44A2_FFFF |
| axi_cdma_0 | S_AXI_LITE | Reg | 0x44A1_0000 | 64K | 0x44A1_FFFF |
| axi_bram_ctrl_1 | S_AXI | Mem0 | 0xC200_0000 | 128K | 0xC201_FFFF |
| axi_bram_ctrl_0 | S_AXI | Mem0 | 0xC000_0000 | 128K | 0xC001_FFFF |
| axi_cdma_1 | | | | | |
| Data (32 address bits : 4G) | | | | | |
| mig_0 | C0_DDR4_S_AXI | C0_DDR4_AD... | 0x8000_0000 | 1G | 0xBFFF_FFFF |
| axi_quad_spi_0 | AXI_FULLL | MEM0 | 0xC400_0000 | 64K | 0xC400_FFFF |
| axi_bram_ctrl_1 | S_AXI | Mem0 | 0xC200_0000 | 128K | 0xC201_FFFF |
| axi_bram_ctrl_0 | S_AXI | Mem0 | 0xC000_0000 | 128K | 0xC001_FFFF |
| axi_cdma_0 | | | | | |
| Data (32 address bits : 4G) | | | | | |
| mig_0 | C0_DDR4_S_AXI | C0_DDR4_AD... | 0x8000_0000 | 1G | 0xBFFF_FFFF |
| axi_quad_spi_0 | AXI_FULLL | MEM0 | 0xC400_0000 | 64K | 0xC400_FFFF |
| axi_bram_ctrl_1 | S_AXI | Mem0 | 0xC200_0000 | 128K | 0xC201_FFFF |
| axi_bram_ctrl_0 | S_AXI | Mem0 | 0xC000_0000 | 128K | 0xC001_FFFF |

図 6 : 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

ビットストリームの生成が完了したら、次を実行します。

1. 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 で次を実行します。
 - a. [File] → [New] → [Board Support Package] をクリックし、BSP を生成します。
 - b. [File] → [New] → [Applicaton project] をクリックし、新しい空のプロジェクトを作成します。プロジェクト名を「kcu105_enh_quad_perf」とし、このプロジェクトについては [Use Existing BSP] を選択します。
5. パッケージに含まれるソフトウェア ソース コードを SDK から src フォルダにインポートします。

次に、ELF ファイルが生成されます。

ハードウェアでデザインを実行する

次の手順に従って、ハードウェア セットアップにビットストリームおよび ELF ファイルを使用します。

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 : XAPP1223 リファレンス デザインに含まれる、生成済みの BIT および ELF ファイルへアクセスするには次を実行します。

- ready_for_download フォルダへ移動する (テスト フォルダを作成しない場合)
- 手順 3、手順 4 および手順 5 をスキップし、手順 6 から実行する

3. design_1_wrapper.bit ファイルを project_1/project_1.runs/impl_1/ からコピーし、test フォルダにペーストします。
4. kcu105_enh_quad_perf.elf ファイルを project_1/project_1.sdk/kcu105_enh_quad_perf/Debug/ フォルダからコピーし、test フォルダにペーストします。
5. 次のコマンドを入力して test フォルダに移動します。

```
cd test
```
6. コマンド プロンプトで xmd と入力し、ザイリンクス マイクロプロセッサ デバッガ (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 ケーブルから FPGA を design_1_wrapper.bit でコンフィギュレーションします。

```
fpga -f design_1_wrapper.bit
```
8. 次のコマンドを入力して FPGA 上で動作するプロセッサに接続します。

```
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は 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。
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 kcu105_enh_quad_perf.elf  
run
```
12. テストが完了するとハイパーターミナルに SUCCESS、PASSED、または FAILED と表示されます。

```
stop
```

結果

図 7 と図 8 に、出力画面の例を示します。

```

systest-client (on truffle18)
*****
System Performance for AXI Quad SPI Configured in QUAD mode
Entire Chip erase takes more time ... Please wait
Chip Erased
Write started for 4096 pages, Test Bytes : 0x35 & 0x85
### Time Taken for 1 MB Data Write by QSPI : 73912 us    ###
Throughput in MB/sec : 13
Read started for 4096 pages
### Time Taken for 1 MB Data Read by QSPI : 53939 us    ###
Throughput in MB/sec : 18
Data Comparison ... OK
Enhanced QUAD Mode 1MB Data Transfer Test PASSED

```

図 7: エンハンスド クワッド モードの出力例

```

systest-client (on truffle18)
*****
System Performance for AXI Quad SPI Configured in DUAL mode
Entire Chip erase takes more time ... Please wait
Chip Erased
Write started for 4096 pages, Test Bytes : 0x60 & 0x20
### Time Taken for 1 MB Data Write by QSPI : 116171 us    ###
Throughput in MB/sec : 8
Read started for 4096 pages
### Time Taken for 1 MB Data Read by QSPI : 95901 us    ###
Throughput in MB/sec : 10
Data Comparison ... OK
Enhanced DUAL Mode 1MB Data Transfer Test PASSED

```

図 8: エンハンスド デュアル モードの出力例

参考資料

1. 『AXI Quad SPI IP (Kintex-7) コア用スループット性能の測定』(XAPP797: [英語版](#)、[日本語版](#))
2. 『LogiCORE IP AXI Quad SPI 製品ガイド』([PG153](#))
3. 『Spartan-3A DSP 1800A スタータープラットフォームで SPI を利用したフラッシュ メモリからのブートロード』(XAPP1053: [英語版](#))
4. 『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](#))

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2月11日 | 1.0 | 初版 |

法的通知

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

© Copyright 2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

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 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。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。