



XAPP1183 (v1.0) 2013 年 11 月 18 日

Zynq-7000 AP SoC プロセッシング システムと XADC AXI インターフェイスを使用したアナログ データの取得

著者 : Mrinal J. Sarmah, Radhey S. Pandey

概要

このアプリケーション ノートは、ザイリンクスのアナログ/デジタルコンバーター (XADC) で専用の Vp/Vn アナログ入力を使用し、アナログ データを取得する方法について説明します。このデザインでは、ザイリンクスの DMA (ダイレクト メモリ アクセス) IP を使用して XADC の出力データを直接システム メモリへ転送する使用事例を採用しています。Zynq®-7000 All Programmable (AP) SoC プロセッシング システム (PS) で動作する Linux ベースのアプリケーションがメモリにバッファリングされているデータを読み出し、LabView ベース アプリケーションの GUI でデータを収集して、高速フーリエ変換 (FFT) 処理を実行して XADC 出力データの信号対雑音比 (SNR) を計測します。

はじめに

Zynq-7000 ファミリーは、ザイリンクスの AP SoC アーキテクチャで構成されています。この製品は、豊富な機能を備えたデュアルコア ARM® Cortex™-A9 ベースのプロセッシング システム (PS) とザイリンクスの 28nm プログラマブル ロジック (PL) を 1 つのデバイスに組み合わせたものです。PS は ARM Cortex-A9 CPU を中核として、オンチップ メモリ、外部メモリ インターフェイス、幅広い周辺接続インターフェイスを備えています。

XADC は、統合された 12 ビット 17 チャンネル 1MSPS の ADC です。XADC が PL にインスタンス化されている場合、Zynq-7000 AP SoC PS と XADC を AXI インターフェイスを使用して接続できます。XADC はすべての Zynq-7000 AP SoC にエンベデッド ブロックとして用意されています。

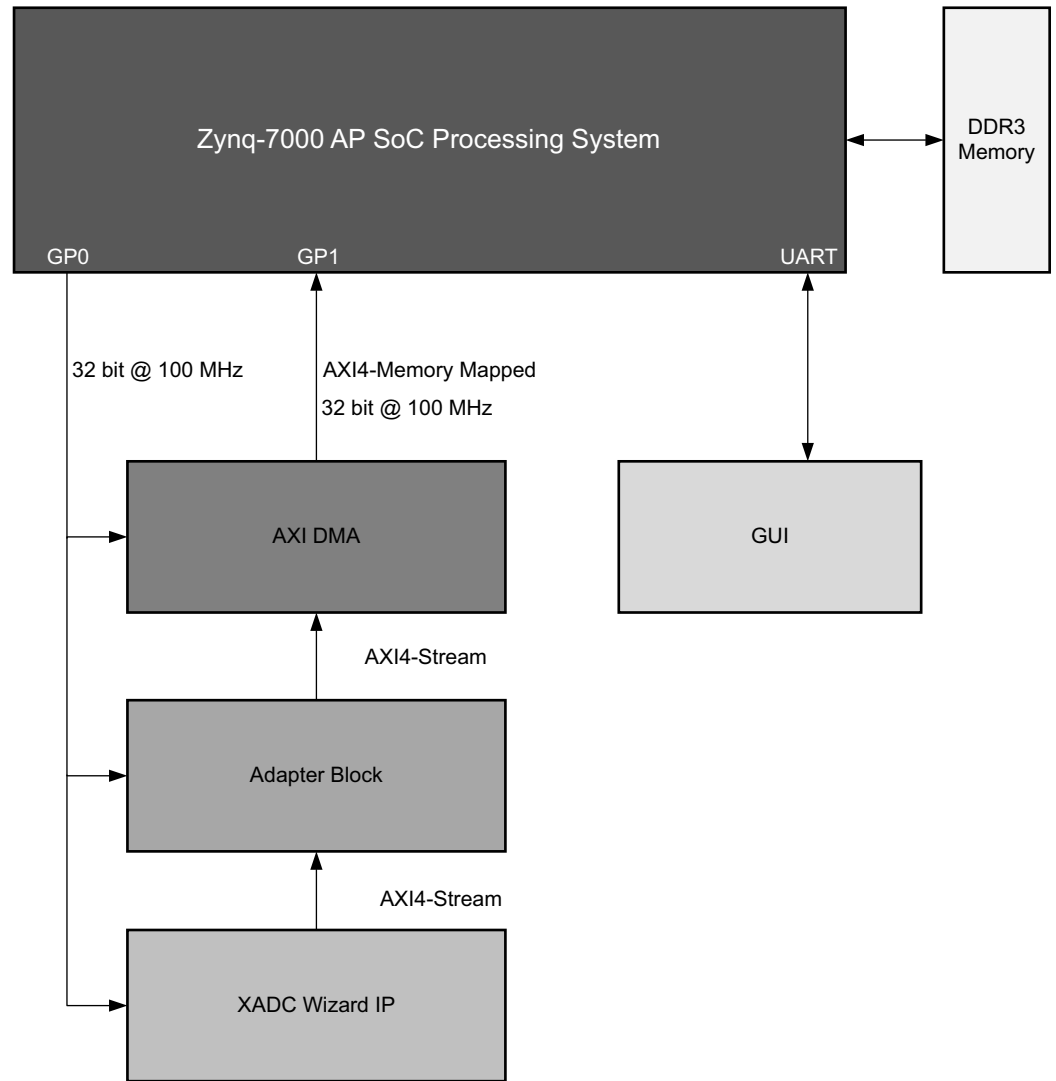
LogiCORE™ XADC Wizard IP は、AXI4-Lite 準拠のインターフェイスとオプションの AXI4-Stream インターフェイスを提供します。AXI4-Lite インターフェイスは XADC のコンフィギュレーションに使用され、AXI4-Stream インターフェイスはデータ通信に使用されます。AXI4-Stream インターフェイスでは、XADC のデータ インターフェイスを別の信号処理 IP と接続することが可能です。このアプリケーション ノートでは、AXI4-Lite インターフェイスを使用して XADC のコンフィギュレーション パラメーターを制御する方法と AXI4-Stream インターフェイスを使用して入力アナログ データ サンプルを取得する方法について説明します。

このアプリケーション ノートでは、汎用 (GP) ポート インターフェイスを使用して XADC と PS 間にデータパスを構築する PL のハードウェア デザインを提供します。XADC AXI4-Stream インターフェイスとの接続には AXI DMA を使用し、DMA が PS DDR3 に XADC サンプルを格納します。Cortex-A9 プロセッサを用いて、ユーザー指定のコンフィギュレーション パラメーター用に XADC を設定します。

XADC の設定と取得したサンプル値の表示には、LabView ベースの GUI インターフェイスが提供されています。この GUI ではサンプル値が解析され、サンプル値や XADC のさまざまな性能評価 (信号対雑音比 (SNR)、高調波の総ひずみ (THD)、有効ビット数 (ENOB)) のグラフが表示されます。LabView GUI は、UART インターフェイスを介して XADC と接続します。

ハードウェア デザインの概要

XADC IP は AXI4-Stream インターフェイスを使用し、XADC サンプルをストリーミング アプリケーションへ渡します。インターフェイスからの TVALID 信号がアサートされると、XADC サンプルが有効です。ハードウェア デザインは 4096 の XADC サンプルを組み込み、AXI DMA IP が AXI4-Stream 転送のフレーム境界を認識するために必要な TLAST 信号を生成します。AXI DMA は 4096 サンプルを PS DDR3 メモリへ書き戻し、プロセッサへ割り込み信号をアサートします。図 1 に、ハードウェア デザインのブロック図を示します。



X1183_01_110613

図 1：ハードウェア ブロック図

データ フローのシーケンスは次のとおりです。

1. XADC の Linux ドライバーが XADC を初期化し、サンプリング周波数を 961.54 KS/S (キロ サンプル/秒) に設定して、XADC が連続サンプリング モードで動作するように設定します。
2. Linux ドライバーが XADC DMA レジスタ (S2MM データおよびリングス レジスタを含む) を設定します。
3. XADC ブロックが外部アナログ ソースからデータを取得し、FIFO に格納します。
4. FIFO アダプターブロックが、DMA ソフトウェアでプログラムされた Coalesce Count 値に基づいて EOP (パケット終了) を生成します。デフォルトの Coalesce Count 値は、8192 バイト サンプルです。

5. FIFO アダプター ブロックの出力は、DMA AXI4-Stream インターフェイスで接続されます。
6. データ転送が完了すると、XADC Wizard IP がファブリックと PS 間の割り込みピンを介して割り込みをアサートします。

XADC Wizard の割り込みポートは、Zynq-7000 PS IRQ_F2P 割り込みポート 0 (ID 91) へ接続されます。

このデザインは、Vivado® Design Suite IP Integrator (IPI) フローを使用してブロック デザインを作成しています。ブロック デザイン作成後、IP 固有のラッパー ファイルをインスタンスエートするハードウェア ラッパーを生成します。

表 1 に、ハードウェア デザインで使用したザイリンクス IP を示します。

表 1：ハードウェア デザインで使用した IP

IP 名	説明	設定
Processing System-7	Processing System-7 エンベデッド ブロックをインスタンスエートするラッパー ファイルを生成する	Processing System-7 IP ウィザードであらかじめ設定された ZC702 のデフォルト値
AXI Interconnect IP	Processing System-7 IP からの AXI3 トランザクションを XADC Wizard IP 用の AXI4 トランザクションに変換する	マスター インターフェイスとスレーブ インターフェイスをそれぞれ 1 つに設定
XADC Wizard IP	XADC エンベデッド ブロックをインスタンスエートし、AXI4-Lite トランザクションをエンベデッド ブロックで必要となる DRP トランザクションに変換する	連続サンプリング モードをサポートするように設定
Proc Sys Reset	ペリフェラルおよび AXI Interconnect IP へリセットを適用する	AXI Interconnect および XADC Wizard IP へリセットを適用するように設定
AXI DMA IP	<ul style="list-style-type: none"> • XADC Wizard IP の AXI4-Stream インターフェイスからサンプルを取得して、DDR3 メモリに格納する • DMA IP は GP スレーブ ポートへ接続される 	スキッター/ギャザー動作を無効にして、シンプルな DMA モードに設定

表 2 に、PL に含まれるメモリ マップされたペリフェラルとアドレス マップを示します。

表 2：ペリフェラルとアドレス マップ

PL のペリフェラル	アドレス マップ (16 進数)
XADC Wizard IP	43C00000-43C0FFFF
Adapter block	43C10000-43C1FFFF
AXI DMA IP	40400000-4040FFFF

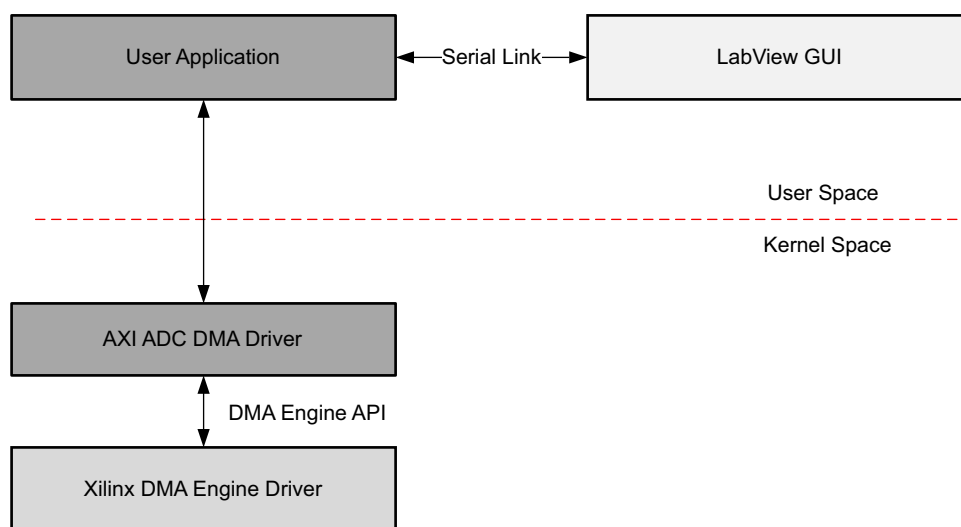
ソフトウェア アーキテクチャ

このアプリケーション ノートで使用するソフトウェア アプリケーションは、Linux DMA フレームワーク ドライバーと一般的なドライバーの実装をベースにしています。このアプリケーション ノートのドライバーは、DMA エンジン API を使用して、ザイリンクスの ADC DMA ドライバーでエクスポートされたサービスを呼び出します。

このアプリケーション ノートのソフトウェア コンポーネントは次のとおりです。

- ザイリンクスの AXI DMA エンジン ドライバー
- XADC DMA ドライバー
- ユーザー アプリケーション
- LabView GUI

図 2 に、ソフトウェアのブロック図を示します。



X1183_02_100913

図 2：ソフトウェアのブロック図

ザイリンクスの AXI DMA エンジン ドライバー

AXI DMA エンジン ドライバーは、DMA トランザクション用にザイリンクスの AXI DMA IP を構成します。これは、標準のザイリンクス git ツリーの一部です。

パス：`$linux-xlnx/drivers/dma/xilinx/xilinx_axidma.c`

XADC DMA ドライバー

このドライバーは、AXI XADC IP および DMA IP を構成します。ADC サンプルの読み出し用にユーザー アプリケーションにキャラクター ドライバー インターフェイスを提供します。このアプリケーション ノートでは、このドライバーのソース コードを提供しています。XADC DMA ドライバーは、次の手順で実行します。

1. プラットフォーム固有のパラメーターのデバイス ツリーを解析し、デバイス ドライバーの構造を初期化します。
2. ユーザー アプリケーションにキャラクター ドライバー インターフェイスを登録し、コールバック関数 (オープン、リード、ioctl、クローズ) をサポートするインターフェイスを提供します。
3. チャンネルに DMA コヒーレント メモリを割り当てます (8KB)。
4. DMA コンフィギュレーションをセットアップします (DMA コンフィギュレーション レジスタをアップデート)。

5. 各チャンネルのイベント キューおよびコンフィギュレーションを初期化します。
6. XADC パラメーターを設定します。
7. ADC サンプル数に応じてアダプター ブロックを設定します。

AXI DMA スレーブ ドライバー

AXI DMA スレーブ ドライバーは、Linux DMA Engine API フレームワークを使用して、ザイリンクスの AXI DMA ドライバーと通信します。次にスレーブ DMA の設定手順を示し、その後の各セクションでそれぞれの手順について詳しく説明します。

1. DMA スレーブ チャンネルを割り当てます。
2. スレーブおよびコントローラー固有のパラメーター値を設定します。
3. トランザクション用の記述子を取得します。
4. トランザクションを発行します。
5. ペンディング要求を発行して、コールバック通知を待機します。

DMA スレーブ チャンネルの割り当て

チャンネルの割り当ては、スレーブ DMA の内容によって多少異なります。通常、クライアント ドライバーには特定 DMA コントローラーからのチャンネルのみが必要です。場合によっては、特定のチャンネルが必要になります。

チャンネルを要求するには、`dma_request_channel()` API を使用します。

```
/* request DMA channel */
axi_xadc_dev->rx_chan = dma_request_channel(mask, axi_xadc_filter,
&match);
```

このインターフェイスを介して割り当てられたチャンネルは、`dma_release_channel()` が呼び出されるまで独占されます。

スレーブおよびコントローラー固有のパラメーター値の設定

次に、DMA ドライバーへ具体的な情報を渡します。スレーブ DMA が使用可能な一般的な情報のほとんどは、`struct dma_slave_config` にあります。これにより、クライアントは DMA の方向、DMA アドレス、バス幅、DMA バースト長などをペリフェラル用に指定できます。

スレーブおよびコントローラー固有のパラメーター値を設定するには、次を実行します。

```
/* configures DMA controller parameters*/
rx_dev->device_control(axi_xadc_dev->rx_chan, DMA_SLAVE_CONFIG, (unsigned
long)&config);
```

トランザクション用の記述子の取得

スレーブとして使用する場合、DMA エンジンには次のようなモードのスレーブ転送をサポートします。

`slave_sg` - ペリフェラルから/ペリフェラルへスキャッター ギャザー バッファーのリストをフェッチする

トランザクション用の記述子を取得するには、次を実行します。

```
/* get handle for transaction */
rx_d = rx_dev->device_prep_slave_sg(axi_xadc_dev->rx_chan,
axi_xadc_dev->rx_sg, AXI_XADC_BD_COUNT, DMA_DEV_TO_MEM,
axi_xadc_dev->flags, NULL);
```

トランザクションの発行

記述子の準備が整い、コールバック情報が追加されると、記述子を DMA エンジン ドライバーのペンディング キューに入れる必要があります。

トランザクションを発行するには、次を実行します。

```
/* submit transaction */  
axi_xadc_dev->rx_cookie = rxd->tx_submit(rxd);
```

DMA のペンドイング要求の発行とコールバック通知の待機

`issue_pending` API を呼び出すことで、ペンドイング キューのトランザクションがアクティブになります。チャンネルがアイドル状態の場合は、キュー内の最初のトランザクションが開始され、後続のトランザクションは保留状態となります。各 DMA 動作が完了すると、キュー内の次のトランザクションが開始されて、タスクレットがトリガーされます。タスクレットはセットされると、通知用としてクライアントドライバの完了コールバックルーチンを呼び出します。

保留中のトランザクションをフラッシュするには、次を実行します。

```
/*flush pending transactions */  
dma_async_issue_pending(axi_xadc_dev->rx_chan);
```

AXI XADC のコンフィギュレーション

コンフィギュレーション手順は次のとおりです。

1. XADC コンフィギュレーション レジスタ 1 にシーケンサ モードをプログラムします。
2. ADC チャンネル セレクション シーケンサ レジスタの V_p/V_n チャンネルを有効にします。
3. ADC コンフィギュレーション レジスタ 2 に約数を書き込みます。

AXI XADC DMA IOCTL の実装

実行手順は次のとおりです。

1. AXI_XADC_DMA_CONFIG

DMA トランザクション用に DMA バッファをマップします。

- scatterlist 配列を初期化する
- DMA エンジンを設定する
- DMA コールバック関数を登録する
- DMA トランザクションを発行する

注記：AXI_XADC_DMA_CONFIG では、DMA 動作が開始されませんが、ペンドイング キューに追加されます。

2. AXI_XADC_DMA_START

DMA トランザクションが開始します。

3. AXI_XADC_DMA_STOP

DMA トランザクションが終了します。

AXI XADC DMA 読み出しの実装

AXI XADC DMA は、ADC サンプルをユーザー空間のバッファへコピーするためのインターフェイスを提供します。DMA トランザクションを発行して開始した後、このドライバーはイベントキューでのユーザーアプリケーションの呼び出しをブロックします。イベントキューでタイムアウトが発生すると、DMA トランザクションが正常に完了しているかを確認します。その後、取得された ADC サンプルは、ユーザー空間のバッファへ格納されます。

図 3 に、アプリケーションフロー図を示します。

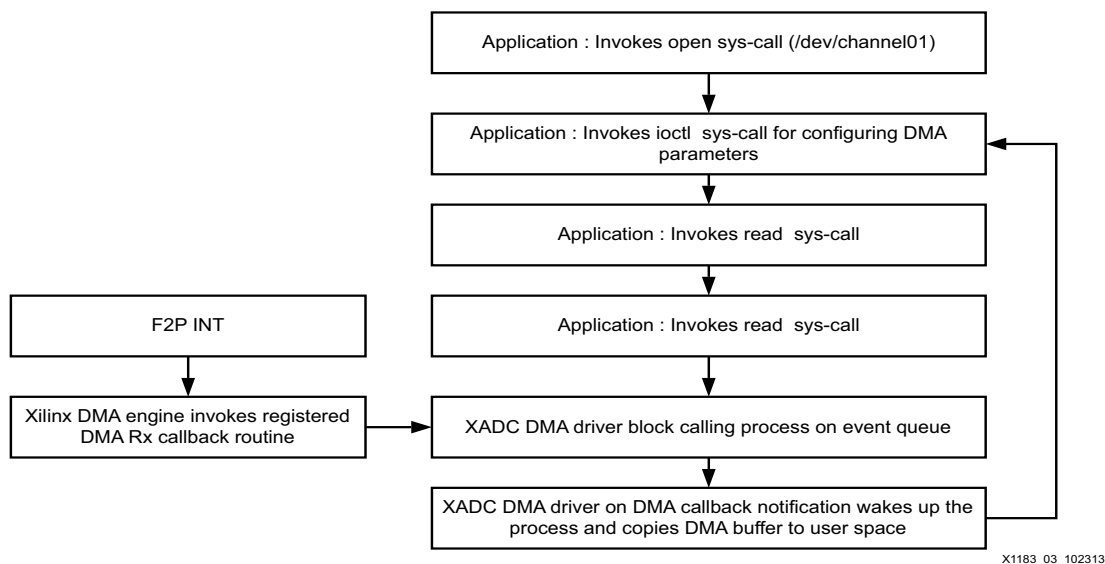


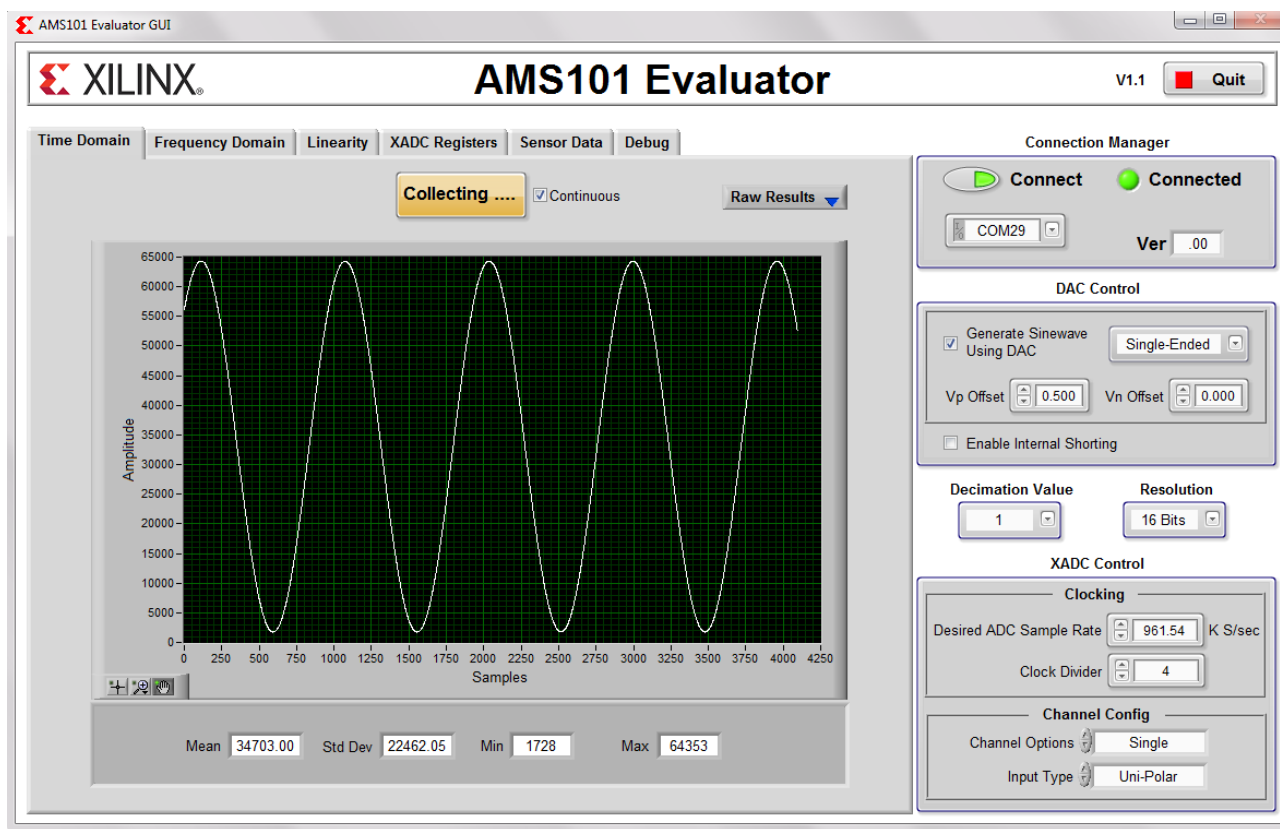
図 3 : アプリケーション フロー図

ユーザー アプリケーション

ユーザー アプリケーションは、AXI XADC DMA ドライバーと AMS101 Evaluator GUI 間のインターフェイスを提供します。このアプリケーション ノートでは、既存の AMS101 Evaluator GUI を再利用しています。評価カードは、ザイリンクスのウェブサイト ([ザイリンクス AMS 101 評価カード](#)) からダウンロードできます。データ取得モジュールとコマンド処理モジュールが含むこの GUI は、ビルトインの LabView 機能を使用して FFT を計算し、その範囲を表示します。AMS101 Evaluator GUI の詳細は、『7 シリーズ FPGA AMS ターゲット リファレンス デザイン ユーザー ガイド』(UG960) [参照 1] を参照してください。

注記：AMS101 Evaluator GUI のインストールの詳細は、『AMS101 評価カード ユーザー ガイド』(UG886) [参照 2] の第 2 章を参照してください。

図 4 に、LabView GUI の [Time Domain] ビューを示します。

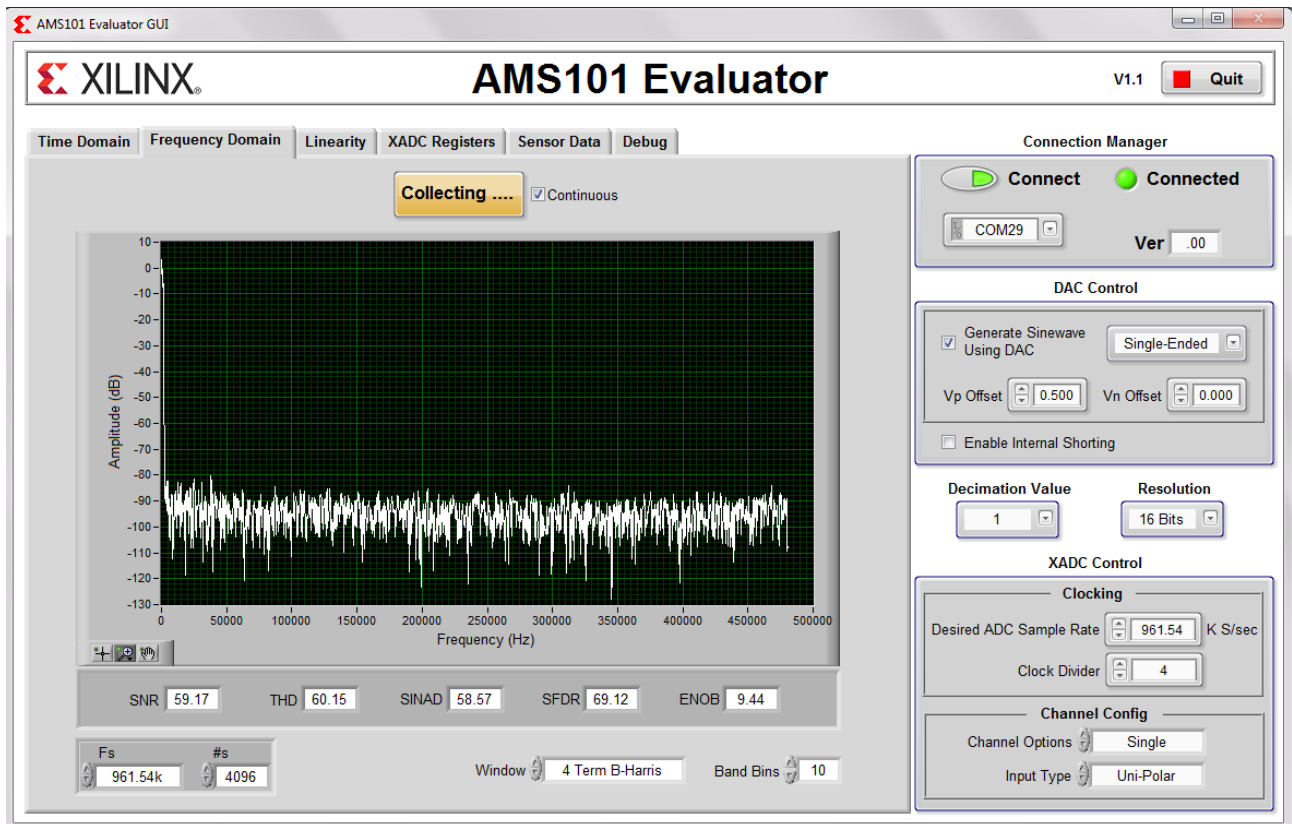


X1183_04_102313

図 4 : LabView GUI の [Time Domain] ビュー

注記：このアプリケーション ノートでは単極モードのみインプリメントしているため、単極モードの XADC 動作を選択してください。選択されているサンプリング周波数は 961.54KS/S です。外部の正弦波ソースをアナログ入力ソースとして接続する必要があります。

図 5 に、LabView GUI の [Frequency Domain] ビューを示します。



X1183_05_102313

図 5 : LabView GUI の [Frequency Domain] ビュー

データ取得モジュールは、チャンネル デバイス ノード (/dev/channel101) を開き、DMA トランザクションを構成して開始した後、それらを循環バッファへ送ります。コマンド処理モジュールは、通信用のシリアルポートを構成します。LabView GUI からシリアルポートを介してあらかじめ定義されたフォーマットでコマンドを受信し、取得したサンプルを送信して表示します。このアプリケーションノートでは、ユーザーアプリケーションのソースコードを提供しています。

ハードウェア要件

このデザインは、ZC702 評価キットでテストできます。AMS101 評価カードを用いて外部データを取得するには、『AMS101 評価カード ユーザーガイド』(UG886) [参照 2] の説明に従って AMS101 カードを ZC702 評価ボードに接続する必要があります。外部信号は XADC エンベデッドブロックの専用チャンネル (Vp/Vn) に接続でき、取得したサンプルは LabView GUI を使用して読み出すことができます。

実験結果

XADC は、サンプリング周波数が 961.54kSPS に設定されています。つまり、可能な最大信号帯域幅は 480kHz であることを意味します。実験用セットアップでは、外部の正弦波ジェネレーターを使用して複数の周波数レベルが生成されており、結果となる信号品質の測定基準が計算されています。信号ジェネレーターの出力にローパス フィルターがない場合、SNR の平均値は 0 ~ 1V の幅で 58dB に近い値となります。

図 3 に、異なる入力周波数に対する信号対雑音比 (SNR)、高調波の総ひずみ (THD)、および有効ビット数 (ENOB) を示します。

表 3 : SNR、THD、および ENOB の値

周波数 (kHz)	SNR	THD	ENOB
1	59.17	60.15	9.44
2	59.25	59.04	9.50
5	59.24	57.64	8.54
10	59.55	57.20	8.51

まとめ

このリファレンス デザインは、アナログ データ取得アプリケーションで XADC Wizard IP の AXI4-Stream インターフェイスを使用するためのプラットフォームを提供します。AXI DMA を使用して、プロセッサの介入なしで XADC サンプルをプロセッサ メモリへ転送する方法を示しています。

このデザインでは、異なる周波数レベルでの XADC の性能評価を定量的に解析します。XADC の信号品質は、整合性に欠けるなサンプルを生成する非理想的な外部ソースを使用して測定されています。

リファレンス デザイン

リファレンス デザインは、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=351581>

readme ファイルの指示に従って、ハードウェアとソフトウェア コードを作成します。表 4 に、リファレンス デザインの詳細を示します。

デザインを再構築する場合は、次の Wiki ページを参照してください。

[「AnalogDataAcquisition_AXI_XADC」](#)

表 4 : リファレンス デザインの詳細

パラメーター	説明
全般	
開発元	ザイリンクス
ターゲット デバイス	Zynq-7000 AP SoC
ソース コードの提供	あり
ソース コードの形式	C
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、CORE Generator™ ツール、サードパーティからデザインへのコード/IP の使用	あり
シミュレーション	
論理シミュレーションの実施	なし
タイミングシミュレーションの実施	なし
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

表 4: リファレンス デザインの詳細 (続き)

パラメーター	説明
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	なし
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 2013.2
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 2013.2
スタティック タイミング解析の実施	Vivado Design Suite 2013.2
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	ZC702

付録 I: デバイス ツリー

ここでは、ザイリンクスの Linux デバイス ツリーの DMA XADC エントリの一部を示します。

```

axi_dma_0: axidma@0x40400000 {
    #address-cells = <1>;
    #size-cells = <1>;
    compatible = "xlnx,axi-dma";
    ranges = <0x40400000 0x40400000 0x10000>;
    reg = <0x40400000 0x10000>;
    xlnx,sg-include-stscntrl-strm=<0x0>;

    dma-channel@0x40400000 {
        compatible = "xlnx,axi-dma-s2mm-channel";
        interrupt-parent = <&gic>;
        interrupts = <0 59 4>;
        xlnx,datawidth = <0x08>;
        xlnx,genlock-mode = <0x1>;
        xlnx,include-dre = <0x1>;
        xlnx,device-id = <0x0>;
    };
};

axi_xadc_dma: axi_xadc@0x43c00000 {
    #address-cells = <1>;
    #size-cells = <1>;
    compatible = "xlnx,axi-xadc-dma";
    reg = <0x43c00000 0x10000 0x43c10000 0x10000 >;
};

```

付録 II : LabView シリアル インターフェイス

表 5 では、さまざまな GUI 操作のレジスタ マップについて説明します。UART を介す送信は、アドレスの後ろにデータバイトが続きます。アドレス制限は 16 ビットで、データ値も 16 ビットです。書き込み (16 ビットのアドレス、16 ビットの書き込みデータ) の場合は、OK 肯定応答信号がアサートされます。読み出しの場合、R AAAAA の後ろに読み出しデータが続きます。

表 5 : GUI 操作のレジスタ マップ

GUI コマンド/クリック	UART 送信		UART 受信
	アドレス	データ	
接続の確立 : デザインバージョン レジスタ (読み出し専用)	0x0000		読み出し専用のデザインバージョンレジスタです。16 ビットのデータを返します。デザインバージョン (D11-8),(D7-0) となります。 D15-12 は、デバイスファミリーを示します (異なるファミリーのデザインに対して同じ GUI を利用できるため)。 0001 - Artix®-7 0010 - Kintex®-7 0011 - Virtex®-7 KC705 デザインの場合、0x2100 を返します。 UART コマンド : R 0000
タイムドメイン/周波数ドメイン タブ : 取得データ (読み出し専用)	0x0001		FPGA は、LV-GUI でさらなる処理を行うために 4096 サンプルのローデータを送信します。 同時サンプリングモードの場合、デュアルディスプレイが必要となり、FFT は VAUX[0] チャンネルに対応する偶数サンプルと VAUX[8] チャンネルに対応する奇数サンプルを含む 8192 サンプルを返します。 UART コマンド : R 0001

参考資料

この文書の参考資料は次のとおりです。

- 『7 シリーズ FPGA AMS ターゲット リファレンス デザイン ユーザー ガイド』([UG960](#))
- 『AMS101 評価カード ユーザー ガイド』([UG886](#))
- 『Zynq-7000 All Programmable SoC テクニカル リファレンス マニュアル』([UG585](#))
- 『7 シリーズ FPGA および Zynq-7000 All Programmable SoC の XADC 12 ビット 1MSPS デュアルアナログ - デジタル コンバーター ユーザー ガイド』([UG480](#))

5. 『DMA Engine API Guide』(Documentation/dmaengine.txt)
6. [ザイリンクス ウェブサイト「ザイリンクス AMS 101 評価カード」](#)
7. [ザイリンクス Wiki ページ「AnalogDataAcquisition_AXI_XADC」](#)

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
2013 年 11 月 18 日	1.0	初版

Notice of Disclaimer

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 the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; 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 Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

本資料は英語版 (v1.0) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

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。