

WinDriver を使った DMA 転送テスト プログラム開発例

本特集では、Jungo 社の WinDriver を使用した PCI/PCI-Express ボードの DMA 転送テスト プログラムを作成する手順を紹介します。ターゲットは、東京エレクトロン デバイス社製の、Virtex-5 FPGA 搭載 PCI-Express 評価ボードです。サンプルの FPGA デザインにより、DDR2-RAM へ DMA 転送が可能です。下記は、そのレジスタ仕様です。

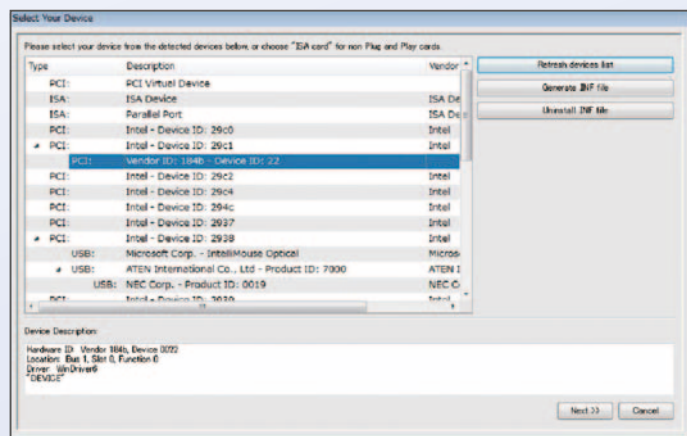
WinDriver ツールキットをインストール後、WinDriver のカーネル モジュールがホスト PC の PCI-Express バスに接続されたデバイスへのアクセスを提供します。WinDriver の GUI ベースの DriverWizard を使用して、ターゲットのデバイスの動作確認を行い、ドライバの雛形となるコードを生成します。以下の 7 つの手順でターゲット デバイスの DMA 転送テスト プログラムを作成します。

評価ボードのレジスタ仕様

オフセット	レジスタ内容	レジスタ名
0x00	DMA Write パケット数	DMA_WSIZE
0x04	DMA Read パケット数	DMA_RSIZ
0x08	DMA Write スタートアドレス	DMA_WCPUADR
0x0c	DMA Read スタートアドレス	DMA_RCPUADR
0x18	DMA Write スタート	DMA_WSTART
0x1c	DMA Read スタート	DMA_RSTART
0x20	DMA Write パケットサイズ	DMA_WPKTSIZE
0x24	DMA Read パケットサイズ	DMA_RPKTSIZE
0x28	DMA ステータス [0]: Write 完了 [1]: Read 完了	DMA_STATUS
0x2c	DMA 割り込みマスク	DMA_INTMSK

手順 1. ターゲット デバイスの選択

DriverWizard を起動し、Select Your Device 画面にホスト PC に接続されたすべての Plug-and-Play デバイスが表示されるので、その一覧からターゲット デバイスを選択します。

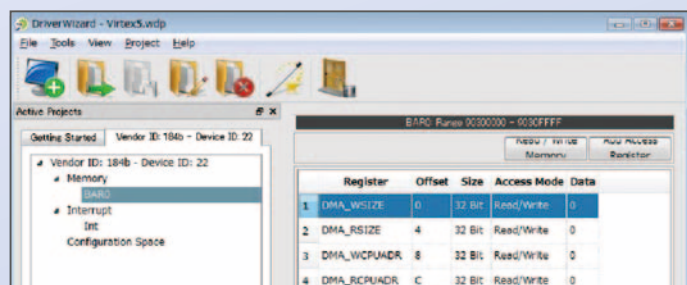


手順 2. INF ファイルの生成とインストール

WinDriver を使用してターゲット デバイスと正しく通信するには、INF ファイルをインストールする必要があります。DriverWizard を使用して INF ファイルを自動的に生成し、インストールします。

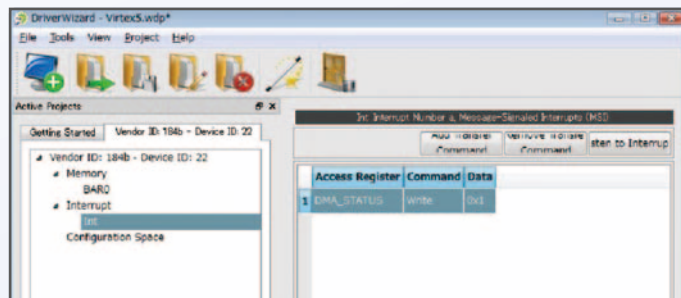
手順 3. レジスタの定義

DriverWizard はデバイスのリソースを自動的に検出します。開発者は、ボードの仕様に応じてレジスタ名やオフセット値などレジスタへのアクセス定義を行います。この定義は、後で自動生成されるコードに反映されます。



手順 4. 割り込み処理の定義

割り込み処理を DriverWizard を使用して定義します。一般的な処理として、ボードからの割り込みを検出した場合に、ステータスをクリアするなどの処理があります。このボードの場合には、ステータス・レジスタのあるビットに書き込むことで、ステータスをクリアします。この定義も、後で自動生成されるコードに反映されます。



手順 5. ドライバ アプリケーションの自動生成

Visual Studio 向けに、ドライバ アプリケーションを自動的に生成します。WinDriver は、一般的な開発環境、開発言語に対応しています。

手順 6. コードの改造

DriverWizard で自動生成されたコードの一部を DMA 転送サンプル プログラムに置き換え、また、手順 3 で定義したレジスタ アクセス関数を利用して、転送データサイズ、DMA バッファ物理アドレス、割り込み許可マスク、DMA 転送開始フラグなどへのアクセスを、仕様に合わせて実装します。

サンプル・プログラムのコード例

```
WDC_DriverOpen(WDC_DRV_OPEN_DEFAULT, NULL);
...
hDev = WDC_PciDeviceOpen(...);
dwStatus = WDC_IntEnable(hDev, NULL, 0, 0,
    interrupt_handler, (PVOID)hDev, FALSE);
...
MyDeviceStartDMA(hDev); /* 仕様依存の実装 */
printf("Press Enter\n");
fgets(line, sizeof(line), stdin);
...
WDC_IntDisable(hDev);
WDC_PciDeviceClose(hDev);
...
WDC_DriverClose();
```

手順 7. アプリケーションのビルドとデバッグ

最後に、DriverWizard で生成されたプロジェクトを使用して、ドライバ アプリケーションをビルドすれば、ターゲット デバイスが期待通りに動作するかデバッグ作業に移れます。作成したドライバ アプリケーションを起動し、DMA 転送を実行すると、DMA バッファにデータが読み込まれていることが確認できれば成功です。ここでも、ハードウェアの性能を十分に引き出すことができますが、更にチューニングを行えば、カーネルモード レベルの最高のパフォーマンスを引き出すことが可能です。

このように WinDriver を使用すると、WDK や OS の内部構造を学習しなくても、高性能なドライバ アプリケーションを短時間で簡単に開発することができます。また、ユーザーモードでドライバ開発を行うので、デバッグが容易であることも大きな特徴です。WinDriver で開発したドライバ コードはサポートするすべての OS 間で、ソース互換性があります。

WinDriver はフル機能を備えた体験版を提供しています。是非お試しください。弊社 Web サイトから無償でダウンロードできます。
<http://www.xlsoft.com/jp/>

また、東京エレクトロン デバイス株式会社主催の無償セミナー「動かす!!初めての PCI/PCIe ドライバ開発セミナー」にて、この WinDriver と評価ボード実機を使ってご体験いただけます。是非ご参加下さい。