

# 新MicroBlazeシステムと BLANCAシステムの接続方法

前章で MicroBlaze コアを組み込んだ基本システムを設計した。本章では、ユーザが設計した IP コアを接続するために、PLB バスと BLANCA システム・バスのブリッジ回路を設計した事例を紹介する。新 SDRAM コントローラのバースト・ポートにグラフィックス・コントローラのフレーム・バッファ読み出し回路を接続する事例についても解説する。

(編集部)

大牧 正知

## 1. XPS\_exif による バス・ブリッジの作成

次は PLB バスと BLANCA システム・バスをつなぐブリッジ xps\_exif を作成します。

### xps\_exif の作成

作り方は opb\_exif の作成時とほぼ同じで、EDK に用意されているウィザードを使用します。このウィザードを使用すると、PLB からユーザがアクセスしやすい簡単なインターフェースへのバス・ブリッジ回路を挿入した形の、サンプルのユーザ IP コアを作成してくれます。

まず、メニューの Hardware から「Create or Import Peripheral」を選択します[図 1(a)]。するとウィザードが起動します。[Next] ボタンをクリックすると、新しい周辺 IP コアを作成するのか、既にある IP コアをインポートするのかを聞かれるので、上のラジオ・ボタンを選択して [Next] をクリックします[図 1(b)]。

次に、EDK のユーザ・リポジトリ・フォルダに保存するのか、XPS プロジェクトの下に保存するのかを聞かれます。ユーザ・リポジトリ・フォルダに保存すると、XPS 起動時にそのフォルダを選択すればその IP コアが使えるようになります。XPS プロジェクトの下に保存すると、その XPS プロジェクトでのみ使えるようになります。どちらでもよいのですが、今回は下のラジオ・ボタンを選択して [Next] をクリックします[図 1(c)]。

そして IP コアの名前を決めます。今回は「xps\_exif」という名前にしておきます[図 1(d)]。[Next] をクリックす

ると、PLB v4.6 のコアなのか FSL のコアなのかを聞かれます。今回は「PLB v4.6」を選択して [Next] をクリックします[図 1(e)]。

今度は PLB-ユーザ・インターフェース・ブリッジの機能を決めます。今回は単純にブリッジできればいいので余分な機能は必要ありません。ただ、ブリッジ・アドレス空間は opb\_exif のときと同じように二つ用意します。従って、「User logic memory space」にのみチェックを入れて、[Next] をクリックします[図 1(f)]。次にバースト機能を使うかどうかを聞かれます。現在の仕様の BLANCA システム・バスではバースト転送をサポートしていないので、OFF のままにしておきます。

[Next] をクリックするとアドレス空間の数を聞かれます。今回は以前の EDK デザインに合わせて「2」に設定します[図 1(g)]。再度 [Next] をクリックすると、ユーザ側に出す信号の選択画面になります。今回は機能が少ないので、信号も少ないようです。とりあえずすべての信号を出しておきたいので、すべての信号にチェックが入ったまま [Next] をクリックします[図 1(h)]。

次は BFM の設定です。BFM というのは MicroBlaze を使わずにハードウェア・シミュレーションで疑似的なマスタ回路がバスを動かして、ユーザ IP コアへのアクセスを簡単にシミュレーションできるようにするものです。ただ現在は MicroBlaze が含まれているシミュレーションでもそれほど複雑ではなくなってきたので、最近ではあまり使われないようです。今回は OFF のまま [Next] をクリックします。

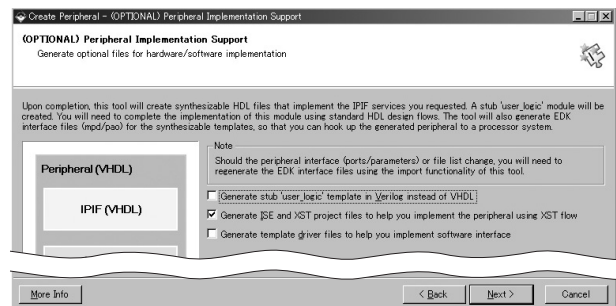
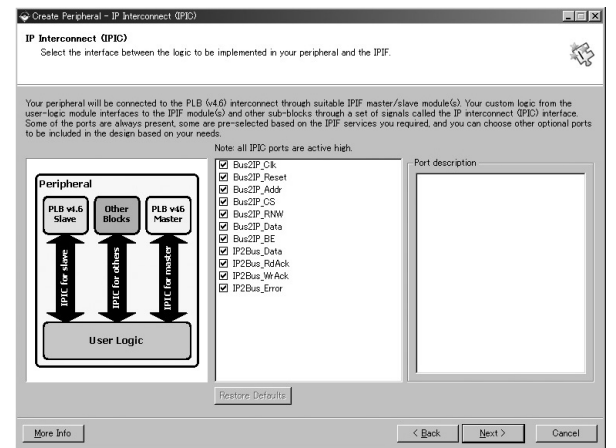
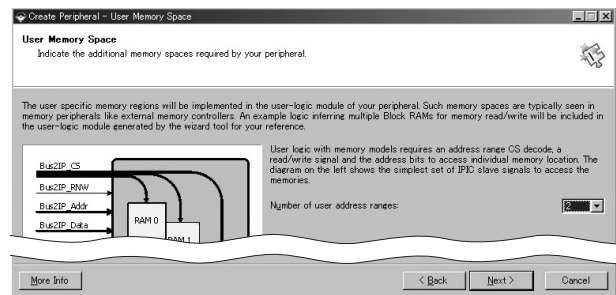
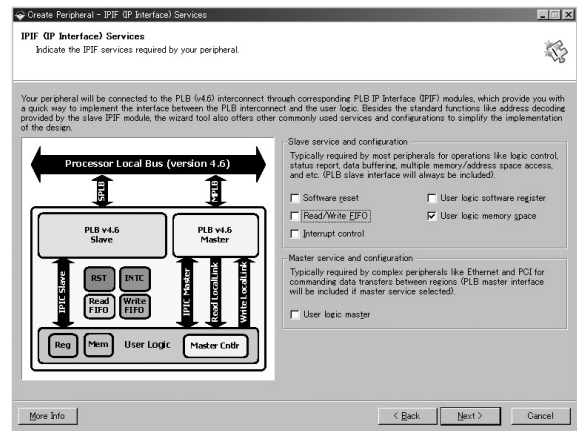
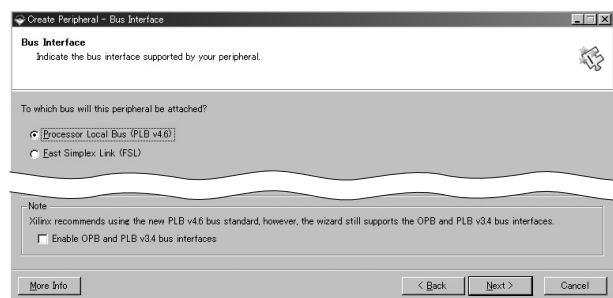
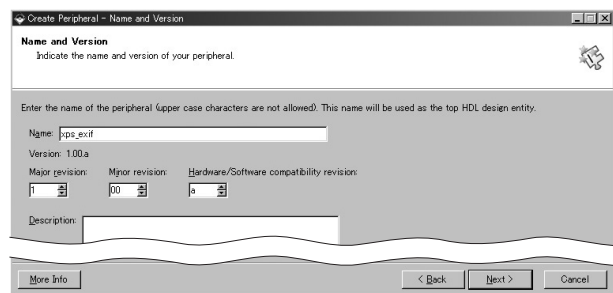
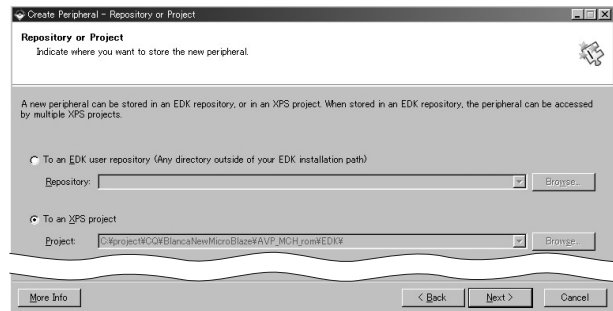
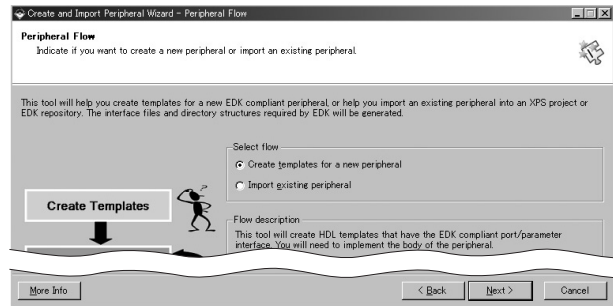


図1 IPIF ウィザードの操作

Pro  
1  
App1  
2  
3  
4  
App2  
5