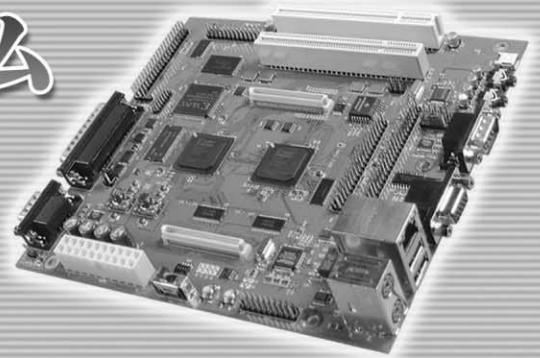


コンピュータ・システム 技術学習キット 活用通信



大牧 正知

第3回 EDKを使ったMicroBlazeシステム開発

今回は、実際に統合開発環境キット EDK を使って MicroBlaze システムを設計するフローについて説明します。

1 EDK を使ったデザイン・フロー

EDK における設計は、ハードウェア・フローとソフトウェア・フローがあります。ハードウェア・フローでは、EDK で用意されている IP コア、あるいは、ユーザが作成した IP コアのうち、どれをどのように接続し、アドレス・マッピングするかを設定していきます。MicroBlaze と EDK で用意されている周辺 IP コアのみを使用してシステムを構成するのであれば、HDL は必要なく、GUI 上によるパラメータ設定だけでハードウェア設計は完了します。したがって、ハードウェア・エンジニアでなくとも、設計は可能だと思います。

ソフトウェア・フローでは、ドライバやライブラリの生成の設定を行い、それで生成されたドライバやライブラリを使って、実際のアプリケーションを C 言語などで記述していきます。ソフトウェア・アプリケーションができあがったら、それを EDK にてコンパイルすれば、オブジェクト・ファイルが生成されます。あとは必要に応じて、デバッグなどを行っていきます。

なお、MicroBlaze で実際にシステムを設計する記事として、参考文献(1)なども参照してください。

2 Base System Builder ウィザードの使い方

クロックやリセットの接続をまちがうと MicroBlaze は動かない

まっさらの状態から設計を始める場合、まず MicroBlaze やメモリ・コントローラを追加して、バスを接続し、クロックやリセットを接続して...という感じで設計していくわけですが、ここでクロックの接続を一つまちがえただけでも、MicroBlaze はウンともスンとも動きません。そうなるといった何が原因で動かないのか追求するのにかなりの時間がかかります。

そこで、とにかく MicroBlaze が動くシステムを作ることが、まずは重要になります。MicroBlaze が動いていれば、たとえ周

辺回路で動かないものがあった場合でも、MicroBlaze からアクセスしてみたときのようすなどを見ることで原因を調べることができるので、かなりデバッグしやすくなります。

この、とにかく MicroBlaze が動くシステムを簡単に作るために用意されているものが、Base System Builder ウィザード(以下 BSB)というものになります。

Base System Builder ウィザードとは

この BSB は、Xilinx 社製の FPGA 評価ボードなどで、そのボード上に乗っている周辺デバイスを使う/使わないを指定するチェック・ボックスをクリックすることで、その機能をもった MicroBlaze のシステムができあがるというものです。その評価ボード用にピンが固定されるので、非常に簡単にほぼ必ず動作する MicroBlaze のシステムができあがります。

この BSB を使うには、動かしたい評価ボード用のボード情報ファイルが必要になります。そのファイルは、通常、評価ボード・メーカーが作成しますが、もちろん、すべての評価ボード・メーカーが EDK の BSB 用情報ファイルを作成しているわけではありません。Xilinx のパートナーの会社が出している評価ボードであれば、この情報ファイルも作成されているようです。

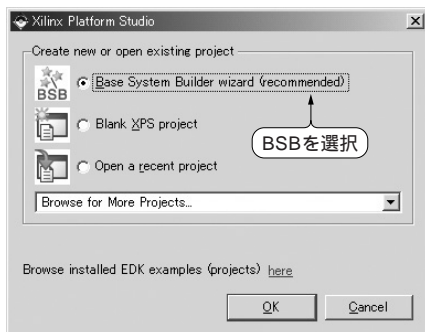
今回は、コンピュータ・システム技術学習キット(2006年3月発売予定、CQ 出版)で動かすため、この BSB 用情報ファイルがありません。また、ユーザがオリジナルのボードを作った場合も、もちろん BSB 用情報ファイルがありません。このような場合のために、とくにボードを特定しないフローも用意されています。

プロジェクトと実装する評価ボードの設定

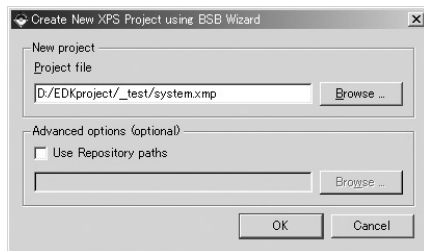
それでは、実際に MicroBlaze が動く最小のシステムを作ってみたいと思います。今回は、MicroBlaze に FPGA 内部のブロック RAM と UART を接続しただけのシンプルな構成にしてみます。

まずはスタート・メニューから EDK を起動します。Xilinx Platform Studio が、MicroBlaze の設計をするための GUI ツールです。起動すると、BSB を使うかどうかを聞いてくるので、今回は使う設定にして OK をクリックします[図 1(a)]。

次に、プロジェクトを保存する場所を指定します。下の Advanced options にて、BSB 用情報ファイルを指定できるの



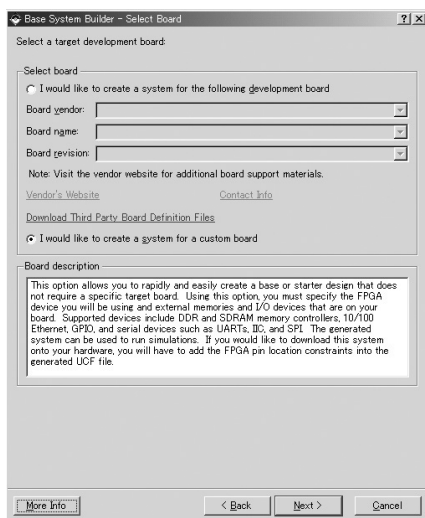
(a) EDK オープン画面



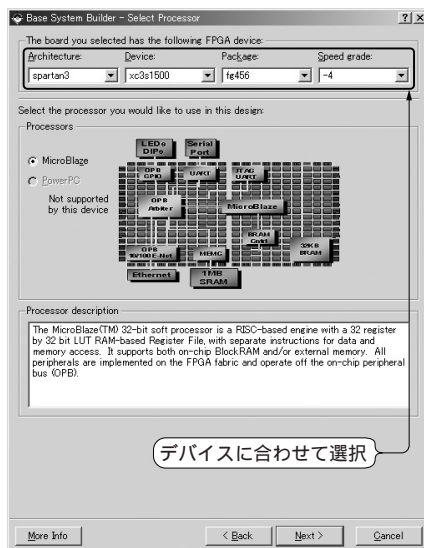
(b) プロジェクトの場所を指定



(c) 新規デザイン作成



(d) ユーザ・カスタム・ボードを選択



(e) デバイスを指定

図1
プロジェクト作成から評価ボードの設定

ですが、今回はないので何も入力しないでおきます【図1(b)】。OKを押すと、次に新規デザインなのかどうかを聞かれるので、新規デザインを選択してNextをクリックします【図1(c)】。

すると次に、評価ボード用フローを使うのか、ユーザ・カスタム・ボード用フローを使うのか聞かれます。今回はコンピュータ・システム技術学習キットのボードを使うので、ユーザ・カスタム・ボード用フローを使う設定にして、Nextをクリックします【図1(d)】。次に、デバイスの指定を行います。本ボードはSpartan3が二つ搭載されていますが、将来的にSDRAMなどを使うことを考えるので、A/Vプロセッサという名称が付いている1500のほうにMicroBlazeを実装してみます【図1(e)】。

MicroBlazeと周辺機能の設定

Nextをクリックすると、MicroBlazeの設定画面になります。まず動作周波数ですが、Reference clock frequencyの欄には、FPGAに接続されているクロックの周波数を入力します。すると、Processor-Bus clock frequencyの欄はプルダウンでいくつか周波数が選べるようになります。このクロックは、FPGAのクロック・マネジメントの機能を使って、^{てい}通倍/分周が行われるように生成されます。その下にはリセットの極性があります。ポー

ド上のリセット・ボタンの極性に従ってこの部分を設定します。

Debug I/Fは通常On-Chip H/W Debug moduleを選択します。ここを選択しておけば、ソフトウェア・デバッグが行えます。Local Memoryは、LMBに接続されるブロックRAMの容量です。ブロックRAMに格納したいプログラムの大きさにあわせて設定します。後からでも簡単に変更できるので、とりあえず最小の8Kバイトを設定しておきます。Cache setupの項目では、キャッシュを使わないか、普通のOPBキャッシュを使うのか、キャッシュ・リンクを使うのかを設定します。FPU命令を使いたい場合は、Enable floating point unitにチェックを入れます【図2(a)】。

Nextをクリックすると、コア追加の画面になります。Add Deviceをクリックすると、追加するコアを選ぶ画面が出るので、ここで使いたいコアを選んでOKを押します【図2(b)】。すると、コアが追加され、コア・パラメータが設定できるようになるので、UARTの場合は、ボーレートやパリティなどの設定を行います【図2(c)】。この作業を必要なコアだけ繰り返すのですが、今回はこのUARTのみの追加にしておきます。

Nextをクリックすると、サンプル・ソフトウェア・プロジェ