

特集1 動かないLSIを動かす方法

第5章

FPGA内部信号観測ツールを活用する

—— ChipScopeを使ったPCIインターフェースの検証

林 幹人

LSIの内部信号を観測したいとき、一般には未使用ピンへ観測したい信号を引き出さなければならない。この方法は、I/Oピンに余裕がなければ採用できず、また、プリント基板上にテスト用の配線パターンを用意する必要がある。しかしFPGAの場合、JTAGポートを利用すれば内部信号を観測することができる。ここでは、米国Xilinx社のFPGA内部信号観測ツール「ChipScope」の活用法を解説する。

(編集部)

最近のプリント基板をみると、FPGAと大容量メモリ、I/Oバッファ、チップ抵抗、コンデンサだけのようものがが増えてきています。このようにFPGAがプリント基板上の機能の大半を占めるようになると、FPGAのシミュレーションがまるでプリント基板のシミュレーションのようになります。シミュレーションに必要な入力パターン・データの量が膨大になり、これを作るだけでもたいへんな作業になっています。かといってシミュレーションによる検証を軽く見ていると後で大きなしっぺ返しをくらう可能性があります。

ところで、FPGAでは設計した回路をすぐに実機上で動作させられることが大きな特徴の一つです。そこで筆者は、FPGAの中身の動作を比較的簡単に見ることができれば、シミュレーション不足を補えるのではないかと考えました。そのためのツールとして、筆者は米国Xilinx社の「ChipScope」というツールを使用し、実機による検証の効率を上げています。

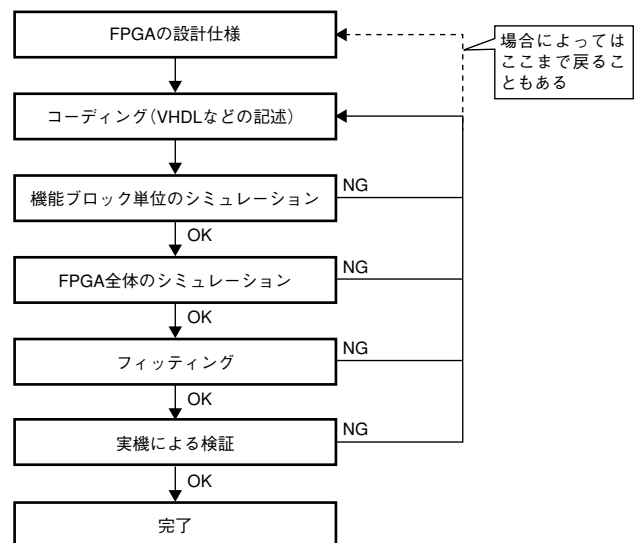
LSIの開発と検証

LSIの一般的な開発フローを図1に示します。まず第一の検証は、機能ブロック単位のシミュレーションです。

機能ブロックは設計によって異なるのでケース・バイ・ケースではありますが、一般には入力信号の数や内部状態の数が比較的少なくなります。したがってシミュレーションはそれほど苦となるものではないでしょう。上流設計に

おいて後工程のことも考慮したうえで機能ブロック分けがなされており、この段階のシミュレーションをしっかりと行っていれば、後々の小さなミスはあったにしても、設計の大きな手戻りが発生するような事態は避けられます。

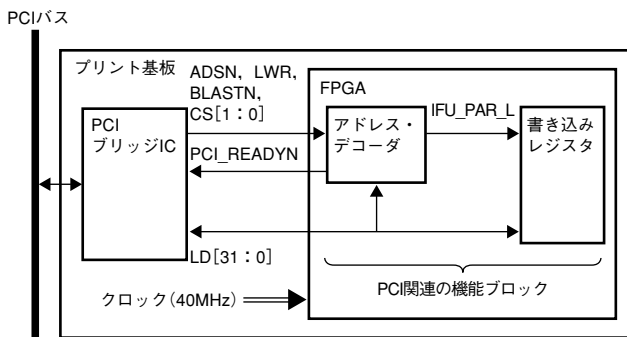
次の段階が、いくつかの機能ブロックを組み合わせた(最終的には、FPGA全体の)シミュレーションとなります。さすがにこの段階になると模擬しなければならない入力信号数や状態の場合分けが膨大な数となり、シミュレーションにかなりの時間を要することになります。ASIC設計では、この時点で各ノードの活性化率を100%近くにしな



【図1】LSIの一般的な開発フロー

5

【図3】
シミュレーションを行うために作成した入力信号波形
PCIバスから78h番地に対する書き込み動作をシミュレーションしたものを。



【図2】FPGAに実装されるPCIインターフェース部の機能ブロック図

ればならず、また開発費用が高額なためあとで作り直せないというかなりのプレッシャーがあります。FPGAの場合はやり直しがきくので少しは気が楽ですが、必要な作業であることに違いはありません。

シミュレーションが終了すると、プリント基板にFPGAを実装し、実機による検証へと移っていくわけです。

シミュレーションによる検証

図2に、筆者らが設計したFPGAに実装されるPCIインターフェース部の機能ブロック図を記載します。

PCIブリッジから信号を受けて、それに対するアドレス・デコードおよびデータの読み出し/書き込み動作のシミュレーションを例にとって説明していきます。

このシミュレーションを行うために作成した入力信号波形を図3に示します。シミュレーションでは、PCIバスからの書き込み動作に対して、PCIブリッジから出力されるアドレス・ストロブ(ADSN)、読み出し/書き込み制御信号(LWR)、アドレスおよびデータ(LD[31:0])を受け、アドレス・デコードにより自分へのアクセスであることを認識します。そして、その動作の完了を知らせるための信号(PCI_READYN)を出力することと、取り込むべき

データを受け取ることができるかどうかを確認します。筆者のシステムでは、PCIのバースト・モードは使用していません。BLASTN信号を受けてはいるものを使用していません。

図3の入力信号波形は、PCIバスから78h番地に対する書き込み動作をシミュレーションしたものです。

いちばん上に示されている波形は、このシステムで使用する40MHzのマスタ・クロックです。使用するツールによって操作に違いはありますが、クロックのような繰り返し波形は、波形エディタの波形リPEAT・コピー機能を使うことで、必要な時間分だけ簡単に入力できます。周期性のない信号は、実際に画面上で必要な時間分だけ時間軸を選択し、所望の論理('0'、'1'の論理値やデータ値)を記述していきます。

この作業がなかなかたいへんで、入力をまちがえれば回路が正しいのにうまく動かないといった結果が得られることになります。最悪の結果をもたらすのが、入力波形がうまいぐあいにまちがってしまい、回路が誤っているにもかかわらず、正しく動作してしまった場合です。この場合、実機による検証の段階になってから、「シミュレーションではうまく動いたのに、なぜ?」という状況になってしまいます。このような場合に、今回紹介するChipScopeが威力を発揮します。

図4にシミュレーション結果を示します。見るべき信号は、自分へのアクセスであることを認識したうえで書き込み動作の完了をPCIブリッジに知らせるPCI_READYNと、受け取るデータを内部でラッチするイネーブル信号であるIFU_PAR_Lです。このシミュレーションによって、この部分のブロックの設計が思惑どおりになっていることを確認できました。

今回のシミュレーションは、結果を得るために10,000ステップも実行すれば十分で、その結果がモニタに表示され