

特集1 動かないLSIを動かす方法

第2章

シミュレーションの基礎

— Quartus IIとModelSimを使ったチュートリアル
宮部秀行

実機を動作させながらデバッグができるFPGAでは、シミュレーションが軽視されがちである。簡単な回路の場合、シミュレーションすることなしに実機動作させることもあるようだ。しかし、ここでは動作保証が必要なLSIを開発できない。幅広い条件で設計したとおり回路が動作するかどうかを検証するためには、シミュレーションの実施が必須である。ここではまず、シミュレーションを行わなかったがために発生した不具合の事例を紹介する。後半はFPGAユーザー向けのシミュレーションのチュートリアルである。(編集部)

FPGAとASICの両方の設計に携わったことのある方に、どちらのほうが設計しやすいかと尋ねると、多くの場合、FPGAのほうが設計しやすいという答えが返ってきます。FPGAはやり直しができるので気楽だから、というのがその理由です。

ASICはインシヤル・コスト(マスク代などの初期投資)だけで数百万~数千万円にもなるので、そう何度もやり直すことができません。ですから、回路の検証には相当の時間を割き、隅々まで入念にチェックを行います。検証を甘く見る人はいません。

FPGAでは、このようなやり直しによるコストはほとんど発生しません。これはFPGAを使ううえでの大きな利点の一つでしょう。反面、検証が甘くなりがちのようです。

設計という観点からすれば、FPGAだからといって検証をおろそかにしてよいというわけはありません。甘い検証は、そのままデバッグの長期化に直結します。

シミュレーションなしでは動かない

シミュレーションは、設計した機能(LSI全体または機能ブロック)の入力ピンから信号を与え、その信号に対する回路の振る舞いが正常かどうかを確認する方法です。入力として与える信号をいろいろと変えてみながら、出力がどのように変化するか、その変化は意図したとおりなのかを調べていきます。

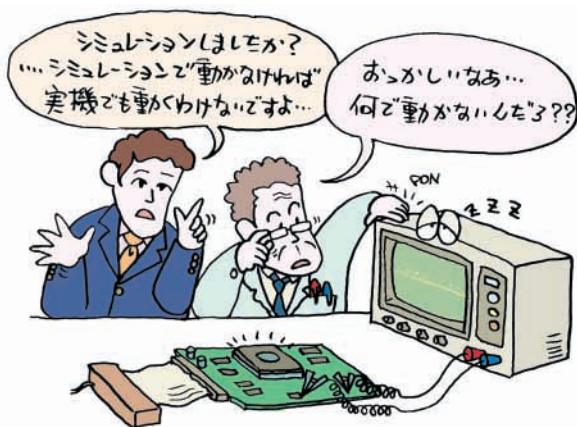
シミュレーションを行うときには、いきなりLSI全体をシミュレーションするのではなく、各階層ごとに行っていきます。シミュレーションにより、設計した回路が論理的に正しいかどうかを調べます。

それでは、このシミュレーションを軽視したばかりに、実機で動作せずトラブった事例をいくつか紹介しましょう。

●シミュレーションしなければ...

デバッグ中なのだがFPGAが思ったように動作せず、デバイスの不良なのかどうかの見極めに苦慮していると相談を受けたことがあります。このような場合、デバイスを疑う人が多いようですが、筆者は設計そのものが正しいかどうかを真っ先に疑ってかかります。

このとき筆者は、まず、



〔図1〕シミュレーションしなければ...

「シミュレーションは行いましたか？」
と尋ねてみました。すると、
「非常にシンプルな設計だったのでだいたいぶだろうと思って、シミュレーションは行わなかった」と言うのです。
「いくら自信があっても、シミュレーションくらいはやったほうがいいですよ」

と促して、シミュレーションを行っていただいたところ、みごとに動きません。「おかし～なあ～」といくら首をひねってみても、シミュレーションで動かない設計が実機で動作するわけがありません。今回の不具合は起こるべくして起きたということです(図1)。

シミュレーション結果から原因を推測し、RTL記述(VHDL)を変更したところ、シミュレーションで正常な動作を確認できました。これをデバイスに書き込んだところ、問題なく動くようになりました。

●非同期回路があると…

デバイスが動いたり動かなかったりする現象が出ていると連絡を受け、手助けのために先方の実験室へと乗り込みました。シミュレーションは行っているとのことで、結果の波形ファイルを見せてもらうと、確かに動いているもようです。ただ一つ、気になる点を見つけました。波形ファイルの入力信号がまったく同じタイミングで変化していたのです。

「これって全部同期信号？」

と尋ねたところ、非同期信号も混じっているとのことです。原因はここにありました。

非同期信号を検証する場合、そのとおりのタイミングで波形を与えなくてはなりません。ところが現実には、どのタイミングで信号が入ってくるかわかりません。あらゆる組み合わせについてシミュレーションを行おうとすると、シミュレーション回数が膨大になってしまいます。それはちょっと無理なので、ある程度の回数で妥協する方法が一般的です。

まず、非同期信号に留意しながらシミュレーションを行いました。すると驚いたことに、デバイスが動いたり動かなかったりという現象をシミュレータ上で再現することができました。特定の非同期信号が入ってくるタイミングによって、デバイスが動かない場合があったようです。きちんとした手順でシミュレーションを行ってれば、事前に

発見できていた問題だったわけです(図2)。

原因がわかればあとは簡単です。非同期部分を同期回路に変更したところ、実機でも正常に動作するようになりました。

いずれも笑えない話です。どちらの例も原因を突き止めるまでに数日を要しています。これは注意していれば防ぐことのできた、まさに人災だったと言えるでしょう。

シミュレーションについて

シミュレーションには、二つの方法があります。

一つは機能シミュレーションです。RTLシミュレーションと呼ばれることもあります。このシミュレーションでは、作成したRTL記述が論理機能の面で正しいかどうかを確認することができます。配置配線後に生じるセル遅延や配線遅延はいっさい考慮されていないので、シミュレーションに要する時間は比較的短くて済みます。

もう一つはタイミング・シミュレーションです。ゲート・レベル・シミュレーションや実負荷シミュレーションと呼ばれることもあります。配置配線後のセル遅延や配線遅延をシミュレーション結果に反映させることが多く、より実機動作に近いシミュレーション結果を得ることが可能です。ただし、機能シミュレーションと比べるとシミュレーションに要する時間は長く、大規模なFPGAの場合、数十秒の動作を模擬するために数日を要するという例すらあります。そのため、FPGA向けの設計では完全同期設計を行うことを前提に、タイミング・シミュレーションを省略するケースも増えています。



〔図2〕非同期回路があると…

2