



ピン情報の流用とボード・レベル・シミュレーション

FPGAの設計フローを ボード・レベルの視点で見直す

関口英明

ここでは、複数のFPGA（またはComplex PLD）を搭載するボードを効率良く開発するための設計フローを紹介する。FPGAでは、信号のピン割り当てを後から自由に変更できる。そこで、プリント基板の部品配置が終わるまで、ピン割り当てを確定させない。また、回路図の中のピン情報を論理合成ツールへ渡せるようにする。さらに、ボード・レベルのシミュレーションを実施することで、検証の品質と効率を高める。（編集部）

年々、設計者には製品の開発期間を短縮することが求められています。一方、EDAツールの進化により、設計効率は飛躍的にアップしてきました。しかし、設計における各段階で使用するツールの開発元が異なると、ツール間の

データの受け渡しなどに人手を要し、設計効率がダウンすることがあります。

ここでは、プリント基板設計ツールとFPGA（またはComplex PLD）設計ツールの間をスムーズにつないで、FPGAとそれを搭載するボードを効率良く開発する手法を紹介します。前半ではFPGAの外部、すなわちプリント基板設計との連携に関するフローを示します。後半ではFPGA内部の設計に関するフローを示します。

なお、筆者は、以下のような環境でしごとを行っています。

- プリント基板からFPGAまで、すべて社内で行っている（プリント基板のパターン設計を担当する部署が社内にあるので、情報交換しやすい）。
- プリント基板からFPGAまで、1人の設計者が担当する機会が多い（設計フローについて、自分で方針を立てやすい）。

以上のことから、FPGAの内部だけでなく、プリント基板のレベルでさまざまな設計改善を行いやすい環境にあります。

●プリント基板設計とFPGA設計の協調型フロー

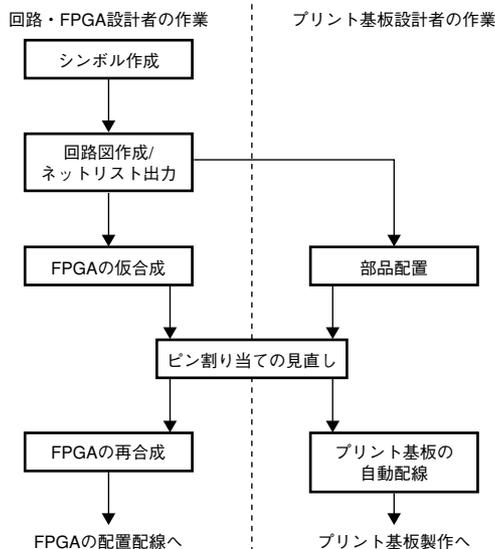
ここで紹介する設計フローのねらいは以下のとおりです。

- 作業を並行処理できるようにする
- 基板上に部品を配置した後、ピン割り当てを見直す
- ピン割り当てのミスによるデバッグの時間を短くする
- 人手による入力時間のむだを排除する

設計フローを図1に示します。この図のフローに従って、各工程の作業を説明していきます。

1) 回路図シンボルを作成する

最近では、FPGAベンダがテキスト・ファイルのピン配



〔図1〕プリント基板設計とFPGA設計のフロー

部品配置中にFPGAのピン割り当てをチェックするため、FPGAの仮合成を行うところがこのフローの特徴。ただし、ピン情報の入力が手作業だと、やる気がでない。

列情報をホームページで公開しています。公開していない場合は、PDFフォーマットのデータシートからテキストの形でピン配列情報を抜き取ります。このテキストから回路図シンボルを自動作成します。この方法により、ミスのないシンボルを作成できます

2) プリント基板用の回路図を作成する

この時点では、FPGAの信号のピン割り当てについて深く考えず、適当に割り当てておきます。プリント基板の配置結果が優先されるため、FPGAの信号のピン割り当ては後で変わります。ここではネットリストを作成するための回路図であると考えます。ただし、電源(特に複数電源)用ピンとコンフィグレーション用ピンはまちがえないように注意します。

3) 回路図エディタでネットリストを出力する

回路図からネットリストを抽出し、プリント基板設計者に回路図とともに渡します。

4) 基板上の部品を配置する

プリント基板設計者は部品配置を行います。その間に回路設計者はFPGA内部の設計を行います。

5) FPGAの仮合成を行う

ネットリストから各FPGAのピン情報を抽出し、FPGA用論理合成ツールへ入力します。この段階で、いったん配置配線まで行います。そして、信号の振り分けミスがないかどうかをチェックします。ミスが見つかったら、プリント基板設計者に伝えます。この具体例は後ほど説明します。

6) ピン割り当ての見直しを行う

プリント基板設計者が部品配置を終了した時点で、回路設計者が部品配置をチェックします。このとき、部品配置の結果をもとに、FPGAの信号のピン割り当てを見直します。配線がクロスしていないかをチェックし、プリント基板設計者と相談してピン割り当てを改善すれば、最終的なネットリストの完成です。

7) プリント基板の自動配線を行う

ピン割り当てを見直した後、プリント基板設計者は自動配線を行います。

8) FPGAの再合成を行う

このピン割り当てがFPGAの最終的なピン割り当てになります。ここで再度、ピン情報をFPGA用論理合成ツールへ読み込みます。回路(FPGA)設計者は基板が完成するまでにFPGAの内部回路の設計を完成させます。

●ピン情報を論理合成ツールに読み込ませる

「たかがピン設定」と思われるかもしれませんが、最近ではFPGAの多ピン化により、人手による入力が難しくなっています。人手で行うと必ずミスが発生する危険が付きまといまいます。特にBGA(ball grid array)などのパッケージを使っている場合、ジャンパなどで後から修正することはほとんど不可能です。また、ピン入力のミスによってデバッグ時間が取られるのは、バカげています。赤鉛筆で色塗りをすることが好きな方は別ですが…。

ここでは、回路図エディタからピン情報を抽出し、論理合成ツール(米国Mentor Graphics社のLeonardo Spectrum)へ引き渡す方法を紹介し、論理合成ツールのTcl(Tool Command Language)機能を利用します。ピン情報をテキスト・ファイルにして、Tclのスクリプトで読み込みます。Tclを利用する利点としては、コマンド一つで論理合成から配置配線までの実行をコントロールできることが挙げられます。GUI(graphical user interface)のようにあちらこちらをクリックする必要がないので、筆者は好んで利用しています。

ここで紹介する作業のねらいは、以下のとおりです。

- FPGAに依存しない論理合成の段階で、可能なかぎりの設定を行う

各FPGAベンダの配置配線ツールを利用する段階でピン割り当てを行うこともできますが、その方法だとFPGAベンダごとに手順やツールを考える必要があります。論理合成の段階で行えば、デバイスやFPGAベンダに左右されない作業環境を実現できます。コストの関係で、設計途中で使用するFPGAが変更になることはよくあることです。

1) 回路図のネットリストからピン情報を抽出する

図2の回路図を例に、ピン情報をFPGA用論理合成ツールへ読み込む方法について説明します。この回路図から出力されたネットリストがリスト1です。リスト2のPerlスクリプトを利用して、リスト1からピン情報を抽出します。使いかたは、Perlの環境をインストールし、WindowsではMS-DOSプロンプト(いわゆるDOS窓)のコマンド・ラインで、

```
perl cnvpin.pl [ネットリスト・ファイル] [対象部品番号] > [ピン情報]
```

と入力します。例えば、

```
perl cnv.pl test.net IC3 > pin.dat
```

となります。