

設計再利用を考慮した HDL記述のガイドライン

——LSI設計を“組み立てパソコン”の世界にしないために…

鳥海 佳孝

後々のプロジェクトで再利用することを前提に機能マクロ(IP)を開発する場合、記述の可読性やドキュメントに対する配慮のほかに、「パラメタライズ」という手法の利用が不可欠になる。これは、ビット幅や特定機能の要・不要に対応したパラメータを導入して、ユーザが容易に機能マクロをカスタマイズできるようにするための仕組みである。ここでは、再利用性の高い機能マクロを作成するにあたっての注意点と、パラメタライズの手法の詳細について解説する。 (編集部)

ここではタイトルにあるように、設計再利用を考慮したHDL記述のガイドラインについて、お話しさせていただきま。この手の話をするとき、キーワードとして欠かせないのがIP (intellectual property)という言葉です。この「IP」という言葉は、いろいろな意味が込められて使用されていると思います。今回の特集では、とくに断らない限り、「論理合成可能なモデル、およびライブラリ」を指して使うことにします。本誌付属のCD-ROMには、次章で紹介するさまざまな機能マクロのサンプル記述が収録されていますが、一部を除いて、これは論理合成可能な記述(すなわちIP)を集めたものです。

さて、現在、LSIの回路規模が大きくなるにつれて、IPを使用しないとLSIを短期間に作ることができないと言われるようになってきました。このIPを提供する、いわゆるIPプロバイダが注目を集めており、ここ数年、いろいろな新しい会社が登場しています。LSIの回路規模がさらに大きくなると、その面積の7~8割を(外部調達や自社開発の)IPが占めるようになるとも言われています。おそらくこのような時代はもう目の前に来ており、遅かれ早かれこうしたIPを使った設計が

主流になっていくのだと思います。

その一方で、技術的な側面から見た場合、筆者自身は、「外部調達のIPを使った設計で本当によいのだろうか?」という疑問をもっています。たしかにビジネス的な観点から考えると、以下のような理由で外部調達のIPを使わざるを得ないというのも理解できます。

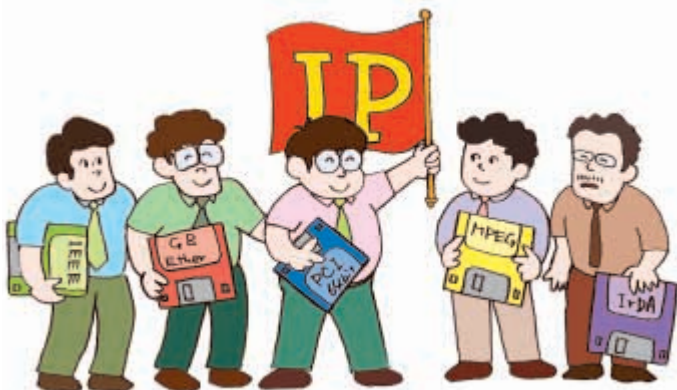
- 短期間にモノ(LSI)を出したい。
- バグのために何度もリワーク(再設計)することは許されない。
- 一つのプロジェクトに何人もの設計者を割くことができない。

しかし、このエンジニアリングの世界で本当にそれでよいのでしょうか? 最近の韓国や台湾(大きな地震があったので、先行き不透明なところはありますが…)の景気よさは、数年前の不況のときに技術的な投資を怠らなかつたからではないか、と考えています。

LSIが組み立てパソコンのようになる

さて、外部調達の(つまり市販の)IPを使用してみんながLSIを設計し始めると、どうなるのでしょうか。たとえば適当なかどうかわかりませんが、それは秋葉原で売られている出来合いのパーツを買ってきて作る組み立てパソコンのようになるのではないかと思います。つまりだれが作っても同じようなものができあがってしまいます。このような状態になると、いったいどのようにして製品の差別化を図るのでしょうか。おそらく、組み立てパソコンの場合と同じようにソフトウェア(OSやアプリケーション)の勝負、あるいはモノの値段だけの勝負になっていくのだと思います。

ところで組み立てパソコンの世界において、技術的な蓄積はどこに集まるかという、組み立てパソコンに使用されているボードやチップなどを提供しているメーカーではないかと思



IPを提供するベンチャ企業が續々登場

います。こうしたメーカの技術競争があるからこそ、次のようなボードやチップを提供したらよいか、どのようなものを作るべきかなどが見えてくるのだと思います。

LSIを作る世界でも、やがて組み立てパソコンの世界と同じようなことが起こるのかもしれませんが(すでにそうなっているのかもしれないが…)。つまり技術的な蓄積はIPプロバイダのなかにしか存在せず、LSIを作る側はたんにそのパーツ(IP)を利用するだけで、価格競争や、どうしてもいいオマケを付ける競争に向かっていくのかもしれませんが。ただ欧米の場合は、IPプロバイダのなかに技術が蓄積されていくだけまだよいのですが、IPプロバイダがほとんど存在しない日本では、このような方向にすら行っていないような気がします。

設計者一人一人がIPプロバイダに

上記のような状況に陥らないためにはどうしたらよいのでしょうか。それはやはり、LSI設計に携わっている設計者自身がIPプロバイダを目指すくらいの気概をもって回路を設計していくことではないかと考えます。IPプロバイダに対して、使用するIPについていろいろと率直に意見を言ったり、的確な指示が出せるように、やはり一度はIPを作ってみるべきではないかと思えます。

技術的な進歩は、それまでの失敗や苦勞の上に成り立つものです。たんに言われたまま、提供されるものを黙って使っているだけでは、しかも使えるか使えないかの判断を下しているだけでは、でき上がったLSIに技術的な差が現れてこないのではないかと思います。

前述したように、ビジネス的な観点から見ると、自分でIPを作っている余裕などないというのが現実でしょう。しかし、そこを乗り越えていかないと、日本の半導体業界が将来にわたって好ましい方向へ進んでいかないのではないかと筆者は思うのです。

前置きがたいへん長くなりました。今回の特集では、IP化や設計再利用を意識したHDL記述のポイントについて解説します。次章のサンプル記述のなかにも、そういった配慮をして記述したものが含まれています。詳しくは、次章の解説および付属CD-ROMに収録されているサンプル記述を参照してください。



LSI設計は、だれが作っても差のない“組み立てパソコン”の世界になる!?

じつは筆者は、6、7年ほど前にも、こうしたIPを設計したことがあります。当時はまだIPという言葉が存在しておらず(あったのかもしれないが、少なくとも筆者は知らなかった)、論理合成可能なモデルということで「シンセサイザブル・モデル」と呼んでいました。そのとき設計したものは8ビットのCPUモデルと、その周辺回路(ペリフェラル)である割り込みコントローラやパラレルI/Oなどでした。限られた誌面ではありますが、これらを作ったときの経験を交えながら、使いやすいIPを開発するコツをお話しさせていただきたいと思えます。

柔軟なモデルを実現する“パラメタライズ”

IP化や設計再利用を意識してモデルを作る場合、HDLを記述するうえで避けて通れないテクニックが「パラメタライズ」です。これは、インスタンス化のとき(Verilog-HDLであればmoduleの、VHDLであればentityの“箱”を置くとき)に、たとえばビット幅をパラメータとして与え、そのIPのビット幅をユーザが自由にカスタマイズできるようにするテクニックです(図1)。

パラメタライズの手法を用いていないと、そのIPをあるときは4ビットで、あるときは8ビットで使いたいという場合に、4ビットと8ビットの両方の下位モジュール(moduleまたはentity)を用意しなければなりません。このような方法をとっていたのでは、同じ種類の“箱”に対して、さまざまなモジュールを用意する必要があり、管理しきれなくなります。しかも、どこまでのビット幅のものを用意すればよいのか見当がつかえません。同じ種類の“箱”に対して、パラメータを与えることによって、ビット幅を柔軟に変更できる仕組みを作り

【図1】インスタンス化してパラメータを与える例

ある機能の“箱”をインスタンス化するとき、ビット幅などのパラメータを与えることによって所望の仕様のモジュールが得られる。この方法を用いれば、すべてのビット幅に対して個別にモジュールを作成する必要がなくなる。管理や保守の面でも扱いやすい。

Verilog-HDL

```
DFE #(8) IO(.C(CLK),.D(DATA_IN),.Q(DATA_OUT));
//
// Verilog-HDLのパラメータの受け渡し
```

Verilog-HDLのパラメータの受け渡しには“#”を使う

VHDL

```
IO:DFE generic map (WIDTH => 8)
--
-- VHDLのパラメータの受け渡し
port map (C => CLK, D => DATA_IN, Q => DATA_OUT);
```

VHDLのパラメータの受け渡しはgeneric mapで

いずれも下位階層で、それぞれparameter、genericを使用していることが条件