

HDLによる 設計法 実践講座

Verilog_HDL編 6

HDL設計の現在と明日

小林 優

本講座の締めくくりとして、HDL設計の今後についてお話ししていきたいと思います。

設計手法の過去と未来

筆者自身、EDAツールには人一倍興味があり、手書き図面から始まって、回路図入力CAD、いわゆる「シリコン・コンパイラ」もいち早く体験してきました。

HDL設計との出会いは、今から5～6年前になります。「これは使える」と直感したものの、しばらく使ってみると、「なんだ、こんなものか」と思いました。過度の期待をしていたので、設計に関して20～30%の省力化でも期待はずれだと思ったものでした。

しかし、使いこなしていくうちに、手放せない設計手法であるとわかりました。「もう回路図入力にはもどれない」というのが筆者の実感です。

HDL設計の今後

「これからあと何年このHDL設計手法が、論理設計の主流でいられるか」という質問を受けることがあります。ビヘイピア合成や、ESDA(Electronic System Design Automation)ツールなど、目新しいツールが次々に登場し、従来のHDLツールにとって変わるかのような印象を受けます。しかし筆者は次のように予想します。

- ・あと10年は、HDL設計が主流であろう
 - ・ビヘイピア・レベルからの真の上位論理合成や、C++、JAVAからの論理合成も登場するかもしれないが、ごく一部に利用されるのみである
 - ・ESDAツールなどは、補助ツールとして、いわばユーティリティ・ソフトのような形で日常的に使われる
- 現在HDL設計が主流だといっても、すべての分野で主流になっているわけでは

ありません。ASICやFPGAの内部の設計に関して、主流になったというだけで、回路図中心で設計されている人も数多くいます。また、用途によっては回路図設計のほうが、回路規模、動作スピード、消費電力の面で、特性を出しやすいということで、HDLをあえて使用しないというところもあります。

したがって、回路図設計が完全にならないうちに、主流になったHDL設計手法も、今後ほかの手法に完全にとって変わってしまうことはないでしょう。

システム・オン・シリコン

システム・オン・シリコンとは

システム・オン・シリコンとは、従来基板上で組んでいたシステムをシリコンのチップ上で実現してしまうというものです。

図6.1のように、従来ではCPUを中心に、ROM、RAM、ユーザ設計のASIC、各種I/Oチップなどを、基板上で配線していました。10年ほど前のシステムを思い出してください。ところが現在では、これらすべてを一つのチップに内蔵することが可能になりました。まさにシリコン上にシステムが構築されたこととなります。

これが可能になったのは、大規模ASICが実現されたほかに、CPUやROM、RAMなどのコアが充実してきたためです。

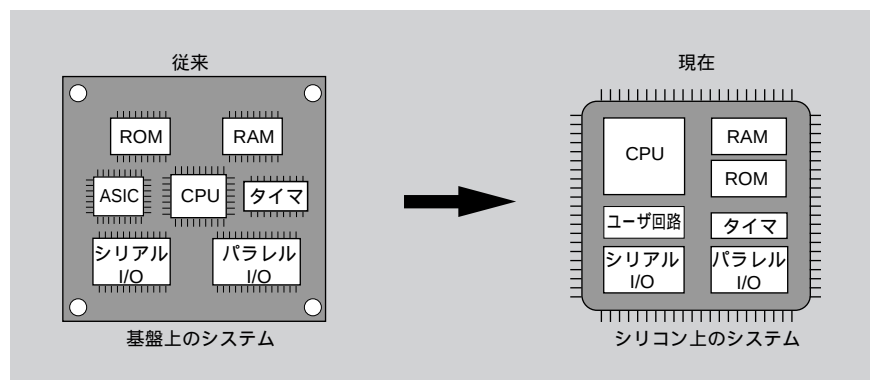


図6.1 システム・オン・シリコン

2種類に分類されるコア

コアを大別すると、

・ソフト・マクロ

... タイマ、シリアルI/O、
パラレルI/Oなど

・ハード・マクロ

... CPU、ROM、RAMなど

となります。ハード・マクロのROM、RAMは、ツールを用いてコアを作成するため、

・コンパイルド・セル

と呼ばれることもあります。

ソフト・マクロは、ASICのライブラリ・セルによって作られた回路です。ネットリストで供給され、シミュレーション時もレイアウト時も、顧客が設計した回路と区別なく扱います。したがって、レイアウトにより配置・配線が変わります。

一方ハード・マクロは、すでにマス

ク・パターンまでできたブロックです。レイアウト時には、一つの大きなブロックとして配置されます。レイアウト時に内部の配置・配線が変化することはありません。

これらのコアは、顧客の回路(ユーザ回路)と接続して一つのチップを形成します。HDLでの設計時には、構造記述を用いて、最上位の階層でコアとユーザ回路を接続します。

このネットリストを用いて図6.2に示すようにシステムの検証を行います。コアの部分は半導体ベンダにより提供され、HDL設計用にすべてHDLで記述されています。ハード・マクロは、シミュレーション専用のモデルです。ソフト・マクロは、ネットリストそのものです。ユーザの回路は当初RTLで行い、論理合成後ゲート・レベルでの遅延のついた完全な検証を行います。

検証時の留意点

RTLでの検証時には、注意すべき点があります。一般にRTLでの記述には遅延をつけません。一方シミュレーション用のコア・モデルには、実チップを想定した遅延がついています。このため、ユーザ回路とコアとの接続部で動作不良を起こすことがあります。たとえば、レジスタ間でのデータの受け渡しがあった場合、データが突き抜けてしまうことがあります。

対策としては、接続するデータ信号に、
wire [7:0] #DELAY dbus;
のように、適度な遅延を付加することができます。

シミュレーション用コアの構造

半導体ベンダ提供のシミュレーション用コア・モデルは、ハード・マクロ、ソフト・マクロで構造が異なり、その内容には興味深いものがあります。

CPUやROM、RAMなどのハード・マクロは、ビヘイビア・レベルでのHDL記述です。たとえばCPUなどでは、図6.3(a)に示すように、if文やtaskを多用してVerilog-HDLの記述力を最大限利用しています。図の例は、インストラクション・デコーダの部分を想定したものです。

一方、タイマやシリアルI/Oなどのソフト・マクロは、ゲート・レベルのHDL記述です。図6.3(b)のように、セルを接続したネットリストになっています。レイアウト時には、論理合成した後のユーザ回路と区別なく扱われます。したがって、ソフト・マクロ内でタイミング・エラーなどを起こしていないかどうかのチェックが必要です。ソフト・マクロ内でのタイミング・エラーの原因は、接続回路のタイミング設計ミス、半導体ベンダ側でのソフト・マクロの作成ミス、たとえば旧ライブラリからの変換ミスなどが考えられます。

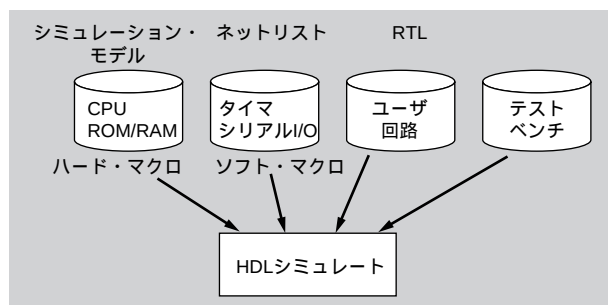


図6.2 システムの検証

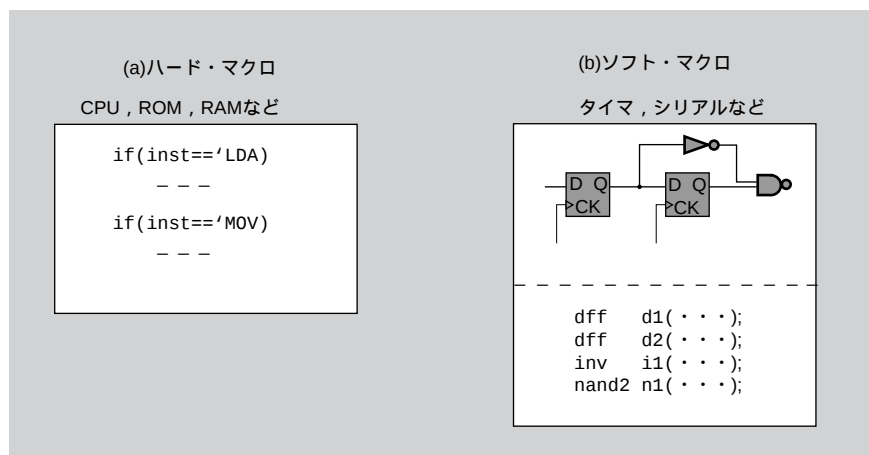


図6.3 シミュレーション・モデルの構造