

LSI 設計で 身を立てる!



第3章

LSIはこうして作る

— LSIの設計工程とASIC/FPGA比較

古川寛

本稿では、LSIがどのようにして設計されるのかについて解説します。前半では、ASIC (application specific integrated circuit ; 特定用途向けIC) を設計する場合を想定し、設計フローに沿って説明していきます。近年のデジタルLSI設計においては、EDA (electronic design automation ; 設計自動化) ツールを使わないことは考えられず、設

計の半分はこれをいかに使いこなすかがかぎになってきています。最近のトレンドなどもスパイスとして盛り込みました。後半では、最近広く使われてきたFPGAの設計手法を、ASIC設計の場合と比較しながら解説します。再設計が容易である特徴を生かすことで、検証の考えかたが変わってきます。(筆者)

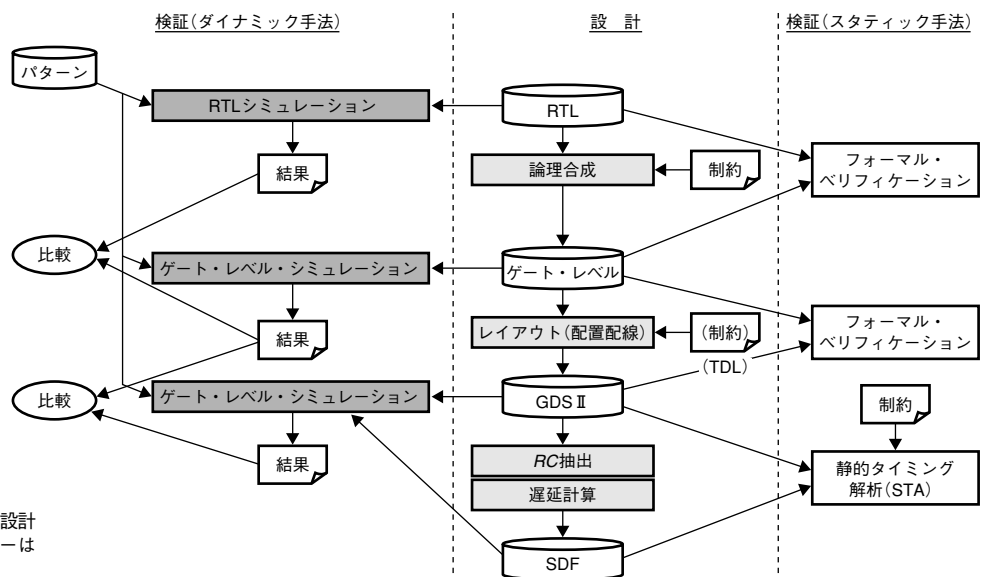
LSIの設計フロー

一般的な設計フローを図1に示します。設計工程のフローなのですが、ツール・チェーンとも呼ばれます。

各設計フェーズをしっかりと設計することはもちろん重要ですが、それらがうまくつながらないようなフローだと、けっきょく設計できません。特にツール・チェーンで使用するツールの入出力データのフォーマットが正しく合致していなければなりません。製品設計がうまくいくかどうかは、最初に適切な設計フローを構築できるかどうかにかか

っていると言えます。

また設計フローは、大きく「設計(実装)」と「検証」の2種類に分けられます。こう動く「はず」というふうに作り込みますが、あくまでも「はず」であり、その後確実に動くことを検証する必要があります。これはツールにも言えることです。残念ながらソフトウェア(ハードウェアも?)にバグは付き物ですから、TAT(turn around time)を短くするだけでなく、最終的にバグを作り込まないために、漏れの



〔図1〕
LSIの設計フロー

一般的な設計工程を示している。今やLSI設計においてEDAツールは不可欠。設計フローはツール・チェーンとも呼ばれる。

ない検証を行うことが必要です。

多くの場合、論理合成よりも前の工程を「フロントエンド」、レイアウト以降を「バックエンド」と呼んでいます。これはゲート・レベルのネットリストで設計データを受け渡していたころのなごりです。最近ではRTLの設計データで受け渡すような場合、論理合成以降をバックエンドと称することもあり、作業の切り分けはあいまいになっています。

HDL 設計

HDL (hardware description language ; ハードウェア記述言語) としては、2種類の代表的な言語があります。すなわち、VHDLとVerilog HDLです。名称が非常に似ていますが、まったくの別物です。いずれもIEEEで規格化されています。VHDLには、IEEE 1076で規格化された1987と1993というバージョンがあります。Verilog HDLには、IEEE 1364で規格化された1995と、最近できた2001があります。さらにはSystemVerilogという規格も出てきています。

各言語の特徴として、VHDLは規格が厳密、Verilog HDLはC言語ライクで記述があいまい、という点が挙げられます。現在、ASIC設計においてもっとも広く使用されているのはVerilog HDLです。そのため、ASIC向けのEDAツールのほとんどが、VHDLよりVerilog HDLのサポートを数ヶ月～数年早く始めたり、あるいはVHDLをサポートしなかったりしています。本稿では、HDLの例題を示す場合、Verilog HDLで示すようにします。

●HDL記述の抽象度

HDLの記述レベルには、設計抽象度が異なる、ビヘイビ

ア・レベル (behavioral level), RTL (register transfer level), ゲート・レベル (gate level) などがあります。

ビヘイビア・レベルは、回路構成などを考慮せず、動作のみを表現するものです。RTLではフリップフロップを明示して、その間を組み合わせ回路でつなぎます。LSIの製造技術には依存しません。ゲート・レベルは、論理セルを一つずつつないでいきます。

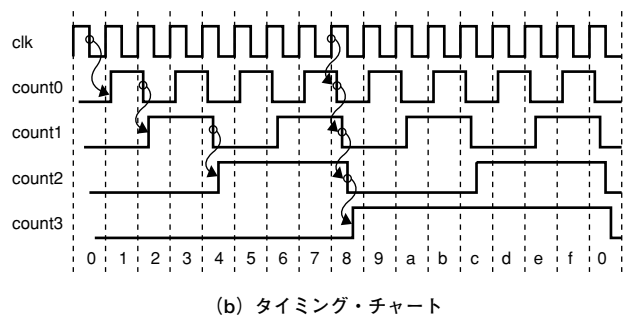
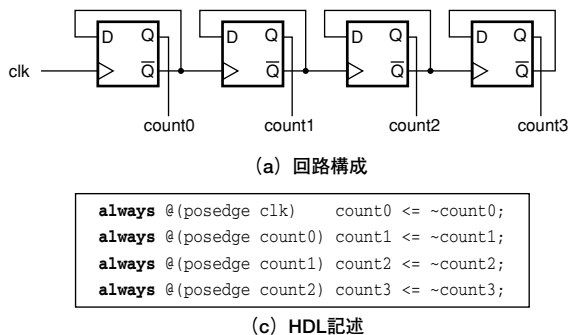
最近のLSI設計では、ほとんどの場合、RTLでHDLコードを記述し、シミュレーションで動作を確認し、論理合成ツールを使ってゲート・レベルのネットリストに変換します。ASIC設計の場合、RTLではVHDLとVerilog HDLの二つを使うことがありますが、ゲート・レベルではVerilog HDLを使うことがほとんどです。またゲート・レベルではEDIF (electronic design interchange format) という形式のデータが使われることもあります。

●なぜHDL設計か —— 回路設計時の理由

最近のLSIは、なぜHDLを使って設計するのでしょうか。ひと言で言ってしまうと、設計効率化のためです。大規模回路の設計で効率を上げるには、設計抽象度を上げてしまうことがいちばんです。

それでは、設計抽象度を上げるということは、どのようなことなのでしょうか。

かつての回路設計では、バイナリ・カウンタを実現する場合、図2のような非同期回路で実現していました。これをそのままRTLで記述すると図2(c)のようになり、特にうまみがありません。なぜなら、回路図をそのままテキストで記述しているにすぎないからです。しかし、同じ機能を図3(c)のように記述できればメリットが出てきます。設計抽象度を上げることで、記述量がぐっと減り、設計効率



〔図2〕非同期回路

バイナリ・カウンタの例。HDLで記述しても、設計効率の点ではあまりメリットがない。