

① HDL設計導入時の意志決定

● 小林 優 ●

はじめに

HDL設計は、論理回路設計を簡易化するための手法です。しかし、ASIC設計においてはさまざまな問題があります。

前回までのサイモン・ゲームの連載では、FPGA化を前提としていたため、HDL設計の効率的な側面だけを強調して解説することができました。

しかしASICの設計では、規模、スピード、コスト、設計スケジュールなど厳しい制約があります。HDL設計だからといって、すべての面で容易化されるわけではありません。

本連載では、これからASIC設計にHDLを本格的に導入される方や、さらなるスキル・アップを図りたい方のために、HDL設計のつまづきやすい点と対処法を一話完結で解説します。

筆者はHDLに手を染めて以来、HDL

でしか設計していません。また、HDL設計のコンサルタントとして独立して以来、HDL設計関連でしかお客様と接していませんので、論理回路設計の大半がHDL設計に移行したかのような錯覚を覚えます。

しかし世の中では、回路図設計は多くの場面で利用されており、その価値がゼロになることはありません。ただ、着実に移行していることは確かです。

そこで今回は、従来回路図設計中心で設計してこられた方が、あるチップの新規開発を機にHDL設計に移行することを想定して、導入時の意志決定について考えてみたいと思います。

HDL vs VHDLの論争に巻き込まれたことがあります。

今考えると、実に不毛な議論だったように思います。なぜなら、表1(a)に示すように、言語の表面的な優劣を根拠に、「敵対」する言語の不利な点を示しては、自分に有利な論理を展開していたからです。

標準でなかるうが(当時Verilog-HDLはIEEE標準でないことを突かれた)、記述の量が3倍だろうが(苦しまぎれにVHDLの弱点をこう主張した)、あまり価値のある議論ではありません。所詮道具なので、表1(b)のように、ライブラリの対応状況と検証の分担(レイアウト後の実配線検証まで同一シミュレータで行うか否か)によって決めるべきだったのです。

ちなみにサインオフ・シミュレータとは、半導体メーカーが実配線検証で使用可とみなしたシミュレータです。いっぽう

Verilog-HDL vs VHDL

言語の優劣の議論は不毛
今から数年前、筆者は社内のVerilog-

表1 Verilog-HDL vs VHDL
(a)かつては言語の表面的な優劣で議論していた

	Verilog-HDL	VHDL
習得のしやすさ		×
文法的な曖昧さ	×	
記述が簡潔		×
符号が扱える	×	

(b)現実的にはライブラリの対応で決めるべき
X社Yシリーズ ゲートアレイのライブラリ対応状況

	各社シミュレータ	ライブラリ	サインオフ	ゴールデン
Verilog-HDL	A社シミュレータ			
	B社シミュレータ		×	×
VHDL	C社シミュレータ			×
	D社シミュレータ		×	×

ゴールデン・シミュレータとは、半導体メーカーの社内でもおもに使用しているシミュレータです。

現実的な観点で言語を決定すべきもし現在同様な議論が生じたなら、次のように考えるべきでしょう。

(1) 言語の優劣は、それほどない

ASIC開発のための論理合成可能な記述に関しては、多少の向き不向きがあっても優劣の差はほとんどありません。それでも「××のほうがよい」というのは、ほとんど趣味、嗜好の範囲です。ビールと日本酒とどちらがよいのかの議論と同じでしょう(筆者は両方好きです)。

(2) 設計資産に合わせる

社内やグループ会社、協力会社などの設計資産があるのなら、それを利用しない手はありません。HDLを用いる場合も、一から設計するより、既存の設計資産を流用するほうがはるかに早いはずです。また、社内で実績がある言語のほうがトラブルの解決時間が短くてすみます。

(3) ライブラリに合わせる

開発するチップのスペック(規模、スピード)に応じたライブラリが、

- ・各シミュレータ用があるか否か
- ・そのリリース時期はいつか

などを考慮して判断します。

また、ライブラリ依存性を極力小さくして、半導体ベンダに対する優位性をもちたいのであれば、最大公約数的なシミュレータを選びます。言語の選択はシミュレータに合わせることになります。

(4) 従来使用していたEDAツールとの「しがらみ」で決める

EDAツールは一般に高額です。新しいものが出たからといってすぐ導入できるものではありません。また、いくら効果が期待できるからといって、過去のツールをすべて捨ててしまうような導入形態は、決裁者がもっとも嫌うやり方です。

したがって、従来使用しているツールベンダがどのようなHDLツールを提供しているか、どの言語をメインに据えているかを調査します。もし自社の考えと一致するのであれば、候補の一つとなる

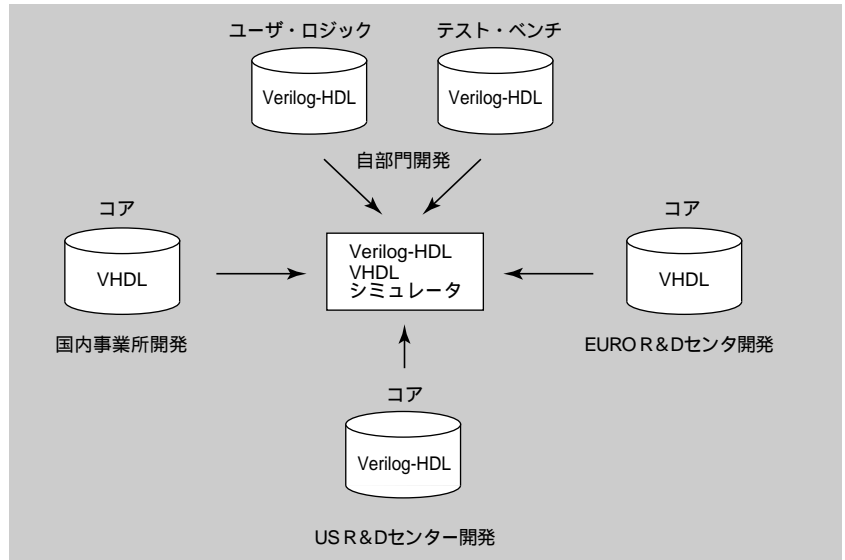


図1 両言語を使わざるを得ない場合もある

でしょう。

ただし、新しい手法に一気に切り替えようとするのであれば、旧来EDAベンダとの決別の覚悟が必要かもしれません。(5)上記のいずれでも決まらないときはVerilog-HDLにする...

筆者の個人的の主観では、Verilog-HDLのほうが「気楽」です。めんどろな型変換もなく、文法エラーのわかりにくさありません。

さらに参考書の一つに拙書を加えていただければ、鬼に金棒です。

HDLバイリンガルをめざそう

Verilog-HDLがVHDLかで悩まなくてもよくなった例があります。つまり両方使わざるを得なくなったというものです。

大規模チップの設計では全社的な設計資産を活用するために、ほかの事業所ばかりでなく、海外のR&Dセンターで開発したコアを用いる場合があります。すべて同一の言語で記述されていけばよいのですが、各々の部門の考え方により異なる場合もあります(図1)。

Verilog-HDLとVHDL間でのコンバータも販売されていますが、合成可能な記述の範囲だけで、テスト・ベンチまで変換してくれるわけではありません。したがって、個別に確認するか、Verilog-

HDLとVHDLのコ・シミュレータを用いることになり、おのずと両言語を使うこととなります。

プログラマの世界では、アセンブラ、Fortran, BASIC, Pascal, C, C++言語と渡り歩き、最近ではJAVAに手を出す方もいます。それに比べれば、Verilog-HDLかVHDLか、などというのは大した問題ではありません。「両方もやる」「できるようになる」というのが、正しいハードウェア・エンジニアの姿かもしれません。

ツールの選択

～PC版かWS版か～

現在のPCは2～3年前のWSの能力PCの高性能化には目を見張るものがあります。筆者は元来Macintosh信奉者でしたが、独立の決意を機にPCを自作しました。1996年5月の時点で20万円そこそこで、Pentium 166MHz, 32Mバイト・メモリ, 1.2Gバイト HDDのマシンができあがりました。今なら(11月現在)もっと安く手にはいるでしょう。

このマシンで、PC版のVerilog-HDLシミュレータを走らせると、ワークステーションのSPARC 20と遜色ないと感じました。SPARC 20で設計していたのは