

特集

パソコン用EDAツール 総解説と総覧

第2章 従来の電子回路設計の概念を越えたEDAツール

HDLによる設計，論理合成，大規模PLDの設計

従来の回路図入力を基本とするゲート・レベルの設計手法から，ますます高度化するシステムに対応するため，デジタル回路の設計手法は抽象度の高いHDL設計へと移ろうとしています。

従来は大企業の一部に限られていたのが，パソコン

で動く低価格ツールの出現により大きく加速され，だれもが普通に使えるものになりつつあります。

このような状況をふまえ，HDLでの回路開発に必要なツールに対して知っておくべき基本的な知識を解説します。

第1節 HDL設計の入力ツール

倉重克己

設計入力ツールは，コンピュータを使った設計の開始点です。従来のゲート・レベルの設計手法では回路図エディタのみでしたが，HDL設計では回路図エディタの他にテキスト・エディタ，場合によっては高い抽象度の入力をサポートするビジュアル入力ツールが使われます。

回路図エディタ

HDL設計でもモジュール化と階層化の概念は不可欠で，ブロック間を配線して階層化する回路図エディタは必要です。

ただし，使用するネットリストは，

HDLのネットリストで，VHDLを使用するならVDHLのネットが抽出できなければなりません。また，論理合成ツールによっては，論理合成の結果を見るためのビューワとしての回路図エディタを使用することもあります。

テキスト・エディタ

HDLはハードウェアを定義するための専用のコンピュータ言語です。C言語などが逐次(シーケンシャル)処理を基本とするのに対し，HDLはハードウェアの本質である同時(コンカレント)処理の

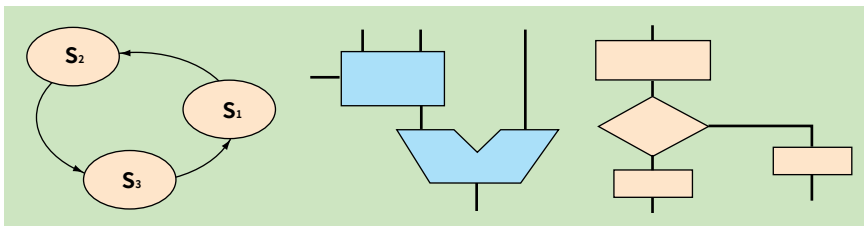
モデリングを考慮にしていることを除いて，コンピュータ言語という点で両者に基本的な違いはありません。設計の手順がたいへんよく似ています。

そして設計の基本であるモジュールの設計入力は汎用言語と同様の普通のテキスト・エディタでの入力で十分ですが，専用のものも入手可能です。言語シンタックスを理解し，入力をサポートする機能などが汎用言語のものと同様に用意されています。

ビジュアル設計入力ツール(ESDA)

昔から対象の設計モジュールの性格により，いろいろな表記がされてきました。

たとえば，ステートマシンには状態遷移図，信号やデータの処理・演算のフローにはデータ・パスを示す図などです。このようなHDLの発達は従来なら紙の上で表記されていたものが，そのままコンピュータへ入力することが可能になり



〔図1〕 抽象度の高い入力手法

ました(図1)。この方法で入力された結果は設計入力ファイルであるのと同時に設計ドキュメントでもあります。

入力された設計は、まずその表記のレベルで徹底的に検証します。別の表記、またはHDLで記述された他のモジュールと合わせてのシミュレーションを行うため、あるいは論理合成のためHDLに

自動的に変換されます。

また、データ・パス要素には論理合成ではなく、実装までを考慮して準備されたハードマクロ・モジュールに直接マッピングされることがあります。

このビジュアル設計入力ツールは仕様書、すなわちシステム・レベルの設計の段階から設計者の思考を助け確実な仕様

の決定に関与するので、ESDA(Electronic System Design Automation)ツールと称されます。

この分野のツールは便利ですが、まだ発展途上ということもあり、使いこなすためには、少なくとも出力されるHDLを読むことができる程度の知識が最低限必要でしょう。

第2節 HDLシミュレータ

倉重克己

VHDLやVerilog-HDLなどの汎用HDL言語は、システム・レベルから配置配線後のモデルまで一つの言語での表現が可能で、設計が進むにしたがって実装の制約下で抽象度の低いレベルに変換され、より具体的なものになっていきます(図2)。

テスト・ベンチ

HDLシミュレータは工程のすべての局面で使用され、各レベルでのシミュレーション結果に基本的な差がないことを確認します。

VHDLやVerilog-HDLのシミュレータでは、検証の対象とするモデルに対して、それをテストするためのテスト・ベンチと呼ぶ新たなHDL記述を追加します。

テスト・ベンチは、テスト・スティミュラス(テスト・ベクター)を含み、これ

をテストの対象とするモデルに与え、シミュレーションを自動実行させるような手順をHDLで記述します。結局、テスト・ベンチ・モジュールを最上位とする新しい一つの閉じたモデルが完成するわけです(図3)。

これをコンパイル、リンクしてコンピュータ上の実行モジュールを作成し、シミュレータ・コントロール・パネル上で実行をコントロールしながら、波形を観察することができるようになります。しかし、効率よく確実なテストをするためには、必要十分なスティミュラスのセットとそれに対するシステムの応答を正確に把握する必要があります。スティミュラスが十分にシステムの動作をカバーしていないとテストのものができませんし、不必要に多すぎても意味がありません。

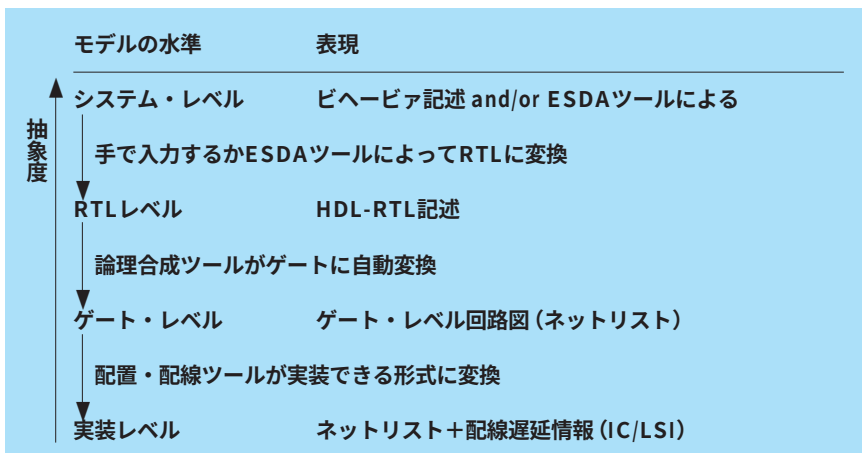
このスティミュラスのセットとそれに

対する応答(期待値)を正確に把握する作業はテストではなく、設計の一部です。

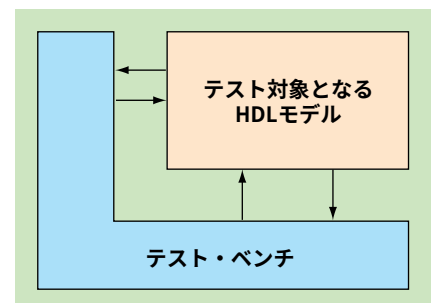
なぜなら、外部からのスティミュラスの応答というシステムの捉え方は、システムを外からみた場合の機能の把握という点でビヘービア・レベルの設計と同じです。したがって、適切なものを用意できるのは設計者だけです。

結局、テスト・スティミュラスの作成は内部構造の設計と同様に重要な作業です。なお、テスト・ベンチの記述はビヘービア・レベルで記述してシミュレーションを行いますが、論理合成には使用しません。

スティミュラスは各レベルで同一のものが使用されます。シミュレーションにあたっては、現在のレベルでのテストを完全にしてから次のレベルに行くことが原則です。次のレベルでは新たに発生する問題のみを解決すべきなのです。前のレベルでの問題を次のレベルで解決しようとするのは、問題をより複雑な場に持ち込んでいるだけにすぎません。



(図2) 抽象度のレベル



(図3) テスト・ベンチを含む最終的なデザイン