

ハフマン符号 デコーダの設計

——サンプル記述付きCベース設計事例(1)

赤星博輝

第3章



ここでは、Cベース設計によって設計の効率化を図る際に重要な役割を果たす「ビヘイビア合成」について解説する。C言語入力のビヘイビア合成ツールを利用する場合、一般にC言語記述を修正しながら対話的に合成作業を進める必要がある。後半では、ビヘイビア合成ツールを利用したハフマン符号デコーダの設計事例を紹介する。なお、ここで紹介するサンプル記述は、本誌のホームページ(<http://www.cqpub.co.jp/dwm/>)からダウンロードできる。(編集部)

デジタルLSI開発における設計自動化の流れは、自動レイアウトなどの下流から論理合成などの上流へと移ってきました。さらに現在は、「Cベース設計」という流れが注目を集めています。C言語で設計することによって何が変わるのでしょうか? 図1のように、一般的に言われていることについて、少し考えてみます。

●ビヘイビア合成ツールを使うと早期に設計が終わるの?

C言語などを用いたアルゴリズムやビヘイビアの検証は、昔から行われていました。これまではこうした検証を行った後、HDLなどを使って一からRTL(register transfer level)設計を行っていました。図2のようにアルゴリズムやビヘイビア(つまり、機能)の検証用に作成したC言語記述からビヘイビア合成ツールを使ってRTL記述を自動生成

できれば、これまで行っていた作業工程を一つ省略できます。つまり、設計期間を短縮できます。

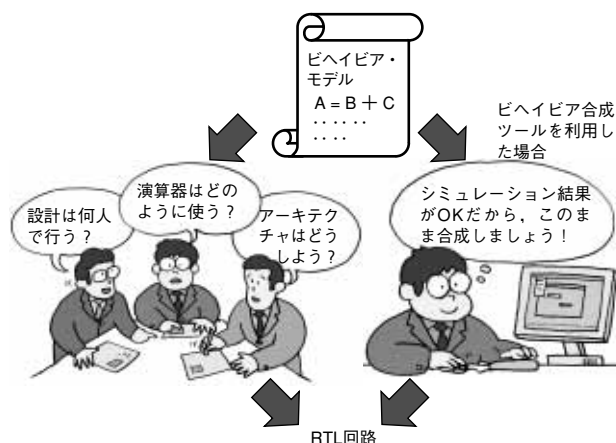
こう説明すると、「じゃあ、検証用モデルからビヘイビア合成を行って、さまざまな制約を満たすだけの性能を持つRTL回路を生成できるのか?」という疑問がわいてきます。これに対する現状の回答は、「性能的に厳しくないときには、検証モデルから合成したものを使えることもある」といった程度だと思います。

ビヘイビア合成ツールを利用する場合、性能の高いRTL回路を生成するためには、C言語記述を修正しながら対話的に合成作業を進める必要があります。



〔図1〕C言語で何が変わる?

Cベース設計でよく出てくる疑問としては、図のようなものがある。その真相を考えてみよう。



〔図2〕検証用モデルから合成できる?

アルゴリズム/ビヘイビア・モデルを作成し、シミュレーションなどによって動作を確認した後、これまでは設計者がRTL記述を一から作成していた。ビヘイビア合成ツールを使用することで、アルゴリズム/ビヘイビア・モデルからRTLを生成することが可能となってきた。

●抽象度が高いから設計が早く終わるの？

ちまたのうわさでは、Cベース設計はHDL設計より「抽象度」が高いとのこと。 「抽象度」ということばはあいまいなので、ここで整理してみます。

C言語とHDLの違いは「逐次動作と並列動作の違い」と言えます。 もともと、Verilog HDLの構文はC言語を意識しているわけですし、VHDLもプログラミング言語であるAdaの影響を受けた言語です。つまり、演算的な部分の差はほとんどないと言えます（もちろん、ビット操作や不定値、ハイ・インピーダンスなど、ハードウェア向けに拡張されている部分はあるが…）。例えば、C言語でもVerilog HDLでも、加算は $A + B$ と記述すれば済むわけですから、演算子を用いた設計では差がないと言えます。

C言語は逐次動作の記述です。一方、HDLを使うと、並列に動作する回路を直接記述することができます。逐次的な記述では（並列性を考慮する必要がないため）動作を書けばそれで作業は終わりです。一方、並列動作の記述では、複数の演算器や記憶素子をどれだけ用意し、どのように使用するかなどについてよく考える必要があります。

Cベース設計は抽象度が高いから設計が早く終わるというのは、これまでのLSI設計で重要だった並列動作に関する検討を行うことなく、逐次動作を記述するからだと言えます。人間にとって、並列動作よりも逐次動作を考えるほうが楽なのかもしれません。

●ツールがなんでも自動でやってくれるの？

「C言語で記述すると、ツールが自動的にベストな回路を作成してくれる」というのはEDAツール開発者の夢ではありますが、現実にはなっていません。

EDAツールの内部ではいろいろな技術を使って回路を最適化しますが、その最適化の目標は設計者が思い描いているものと必ずしも一致していません。また、計算機パワーの限界もあり、最適な回路を合成できていない場合がほとんどです。設計者としては、夢を見る前に技術を磨きたいものです。

●ほんとうに必要なビット幅を指定する

EDAツールは「適材適所」の考えかたに基づいて使わなければなりません。向かない回路構成やアプリケーションに対して無理に適用しても、設計を効率化できないということを頭に入れておきましょう。そのためには、ツールが

採用している基礎的な技術（処理アルゴリズムや最適化の技法）についてある程度理解していないと、どのような局面でどのツール（のどの機能）を使うべきかを判断できないことになります。

ここではビヘイビア合成について簡単に復習しておきます。一般に、Cベース設計対応のビヘイビア合成ツールは、C/C++を拡張した言語、もしくはC++クラス・ライブラリを使用してアルゴリズムやビヘイビアを記述します。何が拡張されているかというと、純粋なC言語では記述できないビット幅やタイミングについての指定などです。

C言語ではint型、short型、char型などのデータ型が用意されていますが、あらかじめ用意されたビット幅以外を使用することは想定していません。しかし、ハードウェアの設計では、このビット幅がLSIの面積や動作速度に大きな影響を与えるので、ほんとうに必要なビット幅を指定してやる必要があります。ビヘイビア合成ツールでは、アトリビュートやライブラリを使うことによって、詳細な設定を行います。

記述を作成したら、それが正しいかどうかを検証し、その後、ビヘイビア合成ツールを使ってRTL回路を生成します。ビヘイビア合成を行う前に、いくつか決定しておくことがあります。それは、動作クロックと使用するテクノロジー（プロセス技術やデバイス・アーキテクチャ）の決定です。動作クロックによって、設計する回路が1クロック何nsで動作するかが決まります。また、使用するテクノロジーによって、演算器（例えば加算器や乗算器など）の動作速度（遅延時間）が決まります。これらの情報をもとに、1クロックでどの程度の処理を行えるかをツールが計算します。

●スケジューリングによってアーキテクチャを決定

ビヘイビア合成の主要な技術の一つに、「スケジューリング」があります。これについて、なぜ動作クロックとテクノロジーに関係あるのかを含めて説明します。例として、以下のような簡単な記述を考えてみます。

$$O = (A+B+C) * (D+E+F);$$

とりあえず、ここでは T_1 、 T_2 という二つのテクノロジーを使用することにします。また、テクノロジー T_1 の加算器の遅延時間が4ns、乗算器の遅延時間が9ns、テクノロジー T_2 の加算器の遅延時間が11ns、乗算器の遅延時間が14nsとします。

ビヘイビア合成ツールは、まず入力された記述を解析し