

# 第2章

## 民生機器向けSOC開発と 先端的セキュリティ処理のIPコア化

### CyberWorkBenchの活用事例

森岡澄夫

Cベース設計手法や動作合成ツールは、黎明期から普及期に移り行った。ツール・ユーザによる設計事例の発表や製品開発への利用も、もはや珍しい話ではない。現時点では、従来RTL (Register Transfer Level) で作成していた回路を少ない工数で開発した、という量的改善の報告が多い。しかしCベース設計や動作合成の価値はそこに留まらない。本稿では、NECのCベース設計環境「CyberWorkBench」を取り上げ、市販製品への適用事例、動的再構成デバイスの開発環境としての利用、RTL設計では不可能だった先端的IPコアの開発を成功させた事例など、質的改善を中心に紹介する。 (筆者)

### 1. CyberWorkBenchの概要と特徴

#### ● 設計対象を限定しない本格的な動作合成

C言語による回路記述からRTL (Register Transfer Level) コードを生成するツールには複数の製品があります。実際には、それらには相当の違いがあります。例えば、回路生成の仕方一つとっても、C言語の各文をほぼ1対1でVerilog HDLやVHDLの一文に置換するツール、クロック・サイクルを全部人手で切るツール、スケジューリングや最適化を全自動で行ってくれるツールなどがあります(当然、生成回路がどれほど実利用に耐えうるかも異なる)。また、簡単なCPUアクセラレータやシーケンス制御回路などの作成に特化したツールもあれば、特にそのような制限がないツールもあります。

そうした中でのCyber Work Bench (CWB)は、正統

的な動作合成ツールという位置づけになります。すなわち、C記述を解析した上でクロック・サイクルの設定やさまざまな最適化を施し、要求された性能の回路を生成します。後述しますが、設計対象についても、「パイプライン回路だけ」とか「シーケンサだけ」といった限定はありません。ターゲット・デバイスは主にスタンダード・セルASICですが、ゲート・アレイやFPGAにも使えます。

#### ● 合成のみならず検証まで含んだ統合環境

CWBは単独の動作合成ツールからなるわけではありません。図1に示すさまざまなツール群から構成されています。C言語を利用したLSI設計に必要な作業工程を、ひとつおきカバーしています。

これらのツール群は、大きく分けて動作合成系(図1の中央)、検証系(図1の左右)、支援ツールやIP (Intellectual Property) コアからなります。各種ツールは、図2に示すGUI上で連携して動作するようになっています(シェル上でコマンド・ライン操作も可能)。

#### ● 動作合成系とその特徴

入力言語として、ANSI-C(ただし変数などに任意のビット幅を指定できるよう拡張したもの)とSystemCの両方が使えます。ハードウェアなのでメモリ・アロケーションや再帰は使えませんが、それ以外の文法上の制限はほぼありません。任意ビット幅の変数や演算が使えるので、演算精度の確保やオーバフロー発生などに神経を尖らせずに済み、普通のC/C++よりも素直にアルゴリズムを書けるケース

#### Keyword

Cベース設計、動作合成、CyberWorkBench、携帯電話、動的再構成デバイス、セキュリティ処理、グループ署名

# 特集 Cベース設計の時代がやってきた！

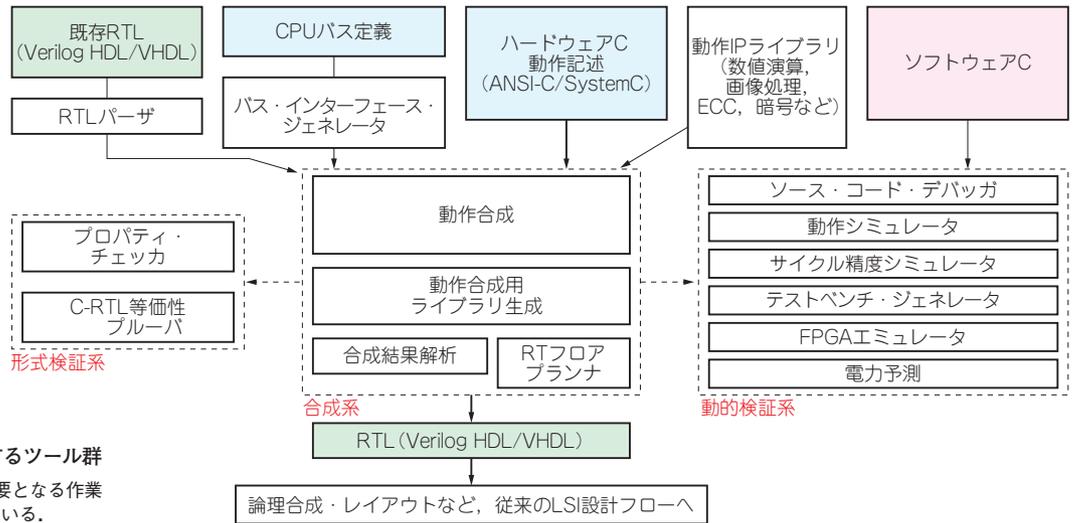


図1 CyberWorkBenchを構成するツール群  
C言語を利用したLSI設計に必要な作業工程を、ひととおりカバーしている。

もあります。

動作合成は基本的に自動で、演算器粒度での処理並列化や回路共有は特に念入りに行われます。C言語で書かれた逐次処理が、その通りに1ステップずつ実行されるわけではありません(それでは非常に遅い回路になってしまう)。ただし、最適化の仕方を自分で制御したければ、ソース・コード中にコメントの形でディレクティブを置くことができます。

また、ソース・コード中にクロック・サイクルを明示的に指定できます。サイクル指定部分と未指定部分の混在も可能です。I/Oシーケンス制御では明示的にサイクルを指定し、演算処理は自動でサイクルを切る、といった使い分けができます。「サイクルを指定するくらいならHDL (Hardware Description Language)でRTLコードを書けばよいではないか」と思うかもしれませんが、状態を明示したステート・マシンを書くよりもずっと楽です(HDLでテストベンチを作るときシーケンスを直列に書くが、それと同じ)。

## ● 検証系とその特徴

検証系は、動的検証(シミュレーション)や形式検証などを行うためのツール群です。

### (1) 動的検証系

動的検証系は、各抽象レベルでC++シミュレーションをするものです。動作レベルだけでなく、サイクル精度(クロック単位)シミュレーション機構もあります。合成された回路のインターフェース・プロトコル確認や、既存RTL資産との接続確認などに用います。

### (2) 形式検証系

形式検証系は、通常のRTL設計で用いられているRTL-

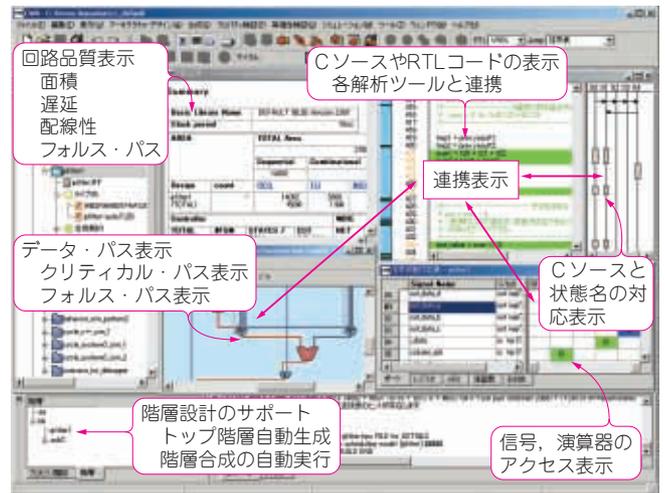


図2 CyberWorkBenchの画面例

各種ツールは、GUI上で連携して動作する。コマンド・ライン操作も可能。

ゲート等価性証明やモデル・チェックに相当します。

### (3) その他

FPGAによる論理エミュレーションの支援機能などがあります。

## ● 各種支援ツール群や動作IP

これら以外にも、各種の設計支援ツールがあります。

- HDLで書かれた既存のRTL設計資産を取り込み、図1中のシミュレーションに組み込みます。場合によっては動作合成することも可能です。
- AMBAバス (AXI, AHB, APB) やそのマスタ/スレーブ・インターフェース回路を自動生成できます。
- 数値演算, 画像処理, ECC (Error Correcting Code), 暗号などのIPコア群があります(動作IPと呼ばれる)。