

# 第1章

# システム LSI 構築のかなめ、 トップダウン設計・検証

——大規模 FPGA と CPU コアを活用して効率化

大庭信之

大規模 LSI の開発では、はじめからすべてを RTL (register transfer level) で設計することは容易ではない。抽象度の高いレベルからトップダウンに設計を進めるのが一般的である。ここでは、設計をいくつかの階層に分けて考え、トップダウンでシステム LSI を設計する手法を解説する。また、抽象度の高い段階や、複数の抽象度が混在する設計レベルで検証を行うために、FPGA と CPU コアを活用する手法について説明する。

(編集部)

近年、半導体技術の進歩とともに、より多くの機能が1個の LSI へ集積できるようになってきました。さまざまな回路ブロックとマイクロプロセッサを混載したシステム LSI (SOC : system on a chip と呼ばれることもある) が注目されています。

LSI の規模が大きくなり、いろいろな機能が集積されるに従って、その設計と検証には多くの時間と労力が必要になります。そこで、大規模 LSI の設計では、最初からいきなり RTL (register transfer level) コードを記述するのではなく、高位の C++ モデルなどを作ることで、トップダウンの設計手順を踏むことが多くなっています<sup>(1)</sup>。また、検証作業も RTL コードのみを使うのではなく、より高位のモデルを使って高速に行うことが多くなりました。しかし、依然として LSI の開発期間の7割が検証に割かれているという報告もあり<sup>(2)</sup>、多くの時間と労力が必要です。

検証作業を効率化・高速化する一つの方法として、FPGA などを用いたハードウェア・プロトタイプングがあります。LSI を製造する前に、設計を FPGA 上に実装して検証する方法で、今では広く用いられています<sup>(2)(3)</sup>。

FPGA の進歩も著しく、大規模化、高速化だけでなく、いろいろなハード IP (intellectual property) を搭載したシ

リーズも登場してきています。

話が少々それますが、時代ごとに LSI 技術の進歩をけん引した電子部品を「テクノロジー・ドライバ」と呼ぶことがあります。1980年代は DRAM、1990年代はマイクロプロセッサ、そして2000年から現在に至るまでは FPGA がそのテクノロジー・ドライバであると言われています。大容量かつ高速な FPGA が入手できるようになり、通信機器、ネットワーク機器、デジタル家電など、いろいろな分野で活用されています。

FPGA ベンダの米国 Xilinx 社からは、PowerPC のハード・マクロを搭載する FPGA 「Virtex-II Pro」、「Virtex-4」シリーズが発売されています。32ビット RISC プロセッサ PowerPC 405 コアが1個ないし2個搭載され、最大 450 MHz で動作し、700Drystone MIPS を超える性能を発揮します。

本稿では、トップダウン型の LSI 設計について、実例を交えながら述べます。例えば、FPGA に内蔵されたマイクロプロセッサを用いて、複数の設計抽象度で記述された LSI を検証する方法を紹介します。

## 1. トップダウン設計・検証

LSI の大規模化、高機能化に伴い、全体を効率良く設計・検証する方法が求められています。そこで、近年、システム・レベルからネットリスト・レベルまで一貫性のあるトップダウン設計手法が用いられるようになってきました。

### ● トップダウン設計とは

最初に LSI の仕様を仕様レベルで記述します。端的に言えば、LSI が何をするのかを明確にします。続いて、設計

表2  
LSI設計のポイントの例

ポイント	説明
ソフトウェアとハードウェアのすみわけ	一つの処理を実行するため、1)すべてをハードウェアで行う、2)汎用CPUもしくはDSP上で走るソフトウェアで行う、3)あるいはハードウェアとソフトウェア両方で協調しながら行う、という3通りの場合が考えられる。
ハードウェア化の効果、ソフトウェアで実現するメリット	一般的に、ハードウェア化することで処理速度は上がり、また消費電力は下がる(そうならないとハードウェア化する意味がない)。しかし、ハードウェアでぎちぎちに組んでしまうと、アルゴリズムやデータ構造を少し変えようと思っても対応しにくくなる。一方、ソフトウェアで実現すれば、あとからアルゴリズムやデータ構造の変更要求に柔軟に対応できる。
レジスタ、メモリ	シングルポートかマルチポートか。同時に二つ以上のアクセスが発生する場合、マルチポートのレジスタやメモリを用意するのが簡単。しかし、デュアルポート・メモリは、シングルポート・メモリに比べて、ハードウェア量が50%~100%増しになることがある。また、デュアルポート・メモリであっても、同時に同じアドレスにライトとリードが発生した場合、どのデータがリードされるかはメモリによるので、注意が必要。資源節約のため、一つのレジスタやメモリを複数の用途で使わずともある。
タイミング(動作遅延)の見積もり	言うまでもないことだが、所定の速度(周波数)で回路が動作しなければならない。例えば加算器でも、ハードウェア量が少ないが遅いリプル・キャリ型を用いるのか、ハードウェア量が多いが速いキャリ・ルックアヘッド型を用いるのか、というように、同じ処理でも異なった回路構成にできないかを考える。
マルチサイクル動作	通常の同期システムでは、グローバル・クロックに同期した入力ラッチと出力ラッチにはさまれた組み合わせ回路は、1クロック・サイクル中に演算を終了している必要がある。しかし、ものによっては2クロック・サイクルかかってよいケースもある。あえてメイン・クロックの半分の周波数をもつクロックを別に用意し、回路の最適化を図ることもある。
レイテンシとスループット	レイテンシは入力を与えてから出力結果が得られるまでの時間のことであり、スループットは単位時間当たりには得られる結果量である。パイプラインを深くすることでスループットを上げるという手段はよく用いられるが、そうするとレイテンシが長くなる。概してレイテンシが長い回路は設計がむずかしくなる。とくに、データによってレイテンシが変わってくるような回路は、動作をきちんと把握しないと設計が破たんしてしまう。

をより詳細なレベル、すなわちトランザクション・レベル、サイクル精度レベルへブレイクダウンしながら、論理回路を合成可能なRTLまで変換していきます(表1)。もちろん、かならずしもこのステップを1段1段たどるとはかぎりません。例えば、簡単なハードウェア・モジュールであれば、仕様レベルから直接RTL設計へ進むこともあります。

求められている処理、機能をハードウェア化するためには、アルゴリズムを正確にハードウェアで実現するだけでなく、いろいろな条件や制約をクリアしなければなりません。そのためには、表2のような観点からハードウェアを煮詰める必要があります。

トップダウン設計を行うことにより、このような条件、制約を的確に設計へ取り込みやすくなります。そして、複雑な大規模LSIでも見通し良く、バグも少なく、開発することができます。また、より高位のレベルで記述されたモデルを使うことにより、すべてのモデルをRTLで記述した場合と比べて、高速にシミュレーションすることができます。

トップダウン設計・検証は、具体的にどのように進んでいくのでしょうか。

図1はシステムLSIの一例です。このシステムLSIにはPowerPC 405がCPUコアとして搭載され、システム・バス(PLB: Processor Local Bus)を通して、メモリ・インターフェース、MPEGエンコーダ、MPEGデコーダ、DES(data encryption standard)/AES(advanced encryption

表1 設計抽象度

下位に行くほど、より最終回路に近くなっていく。動作タイミングも正確に記述される。

設計抽象度	検証レベル	タイミング	実装
仕様レベル	コンセプト	untimed	ソフトウェア
トランザクション・レベル	アーキテクチャ	timed/ untimed	ソフトウェア
サイクル精度レベル	マイクロ アーキテクチャ	timed	ソフトウェア
RTL(レジスタ・トランスファ・レベル)	実装	timed	ハードウェア

standard)暗号エンジンと接続されています。また、PLBはPLB-OPBブリッジを介してI/Oバス(OPB: On-chip Peripheral Bus)とつながります。OPB上にはRSA(Rivest, Shamir and Adleman)/ECC(elliptic-curve cryptography)暗号エンジン、乱数発生回路、汎用I/Oインターフェースが接続されています。では、どのように設計していくのか、レベルごとに見ていきましょう。

### ●仕様レベル(specification level)

仕様レベルでは、LSIのコンセプトおよび何を行うものであるのかを記述します。

「本LSIは暗号機能を搭載した画像処理プロセッサです。画像処理はMPEG-4規格をベースに行います。画像データの暗号化/復号化を含んだセキュアな伝送はSSL(secure socket layer)プロトコルを使います。」