



システムLSIのアーキテクチャ設計 自動化技術とその応用に関する研究

アルゴリズム抽象記述からシステムLSIを自動設計する

戸川 望

早稲田大学理工学術院 基幹理工学部情報理工学科
教授

研究の背景

半導体加工技術の進歩は、1つのLSI（大規模集積回路）の中に、プロセッサ、メモリ、周辺回路ならびにこれらの間のインタフェース回路を集積することを可能とし、現在では、LSIそのものが1つの「システム」を形成するに至っている。その意味で今日のLSIは「システムLSI」やシステムオンチップ（System-on-a-Chip; SoC）とも呼ばれている。

システムLSIには数億個のトランジスタが集積され、もはや人手だけで設計することは非現実的である。「計算機による設計自動化」が不可欠であると言える。

さて設計自動化の歴史を見ると、配置設計や配線設計といった設計レベルの下位工程から設計自動化が進んでいったが、LSIの集積規模と1人の設計者が設計できる回路規模の差は年々広がって来ており（これはデザインライシストと呼ばれる）、より上流工程の設計自動化、換言すれば「システムLSIのアーキテクチャ設計を自動化」する新たな設計パラダイムの構築が急

務となっている。アーキテクチャ設計の自動化は、いわば設計者がこれまで経験と勘によって実現してきたLSI設計を計算機によって自動化するもので、問題設定そのものならびに解法、最適化が極めて難しいものとなっている。

研究の成果

本研究では、LSI設計技術の中でも上記のような背景を持つ、これまで最も困難とされてきたアーキテクチャ設計自動化技術に着眼して来た。

1. FPGAを対象とした論理設計・物理設計の同時最適化技術の構築

本研究の成果の第一として、FPGA（電氣的書き換え可能なLSI）と呼ばれるLSIに着目し、その上位設計工程（論理設計）と下位設計工程（物理設計）を完全に同一化する新たな設計技術を考案した。また実際のFPGAに対しこの設計技術を適用し大きな有効性を確認した[1, 2]。本研究は、FPGAが持つ構造の規則性を利用す

ることで物理設計を論理設計の一部として捉えることに成功したものであり、FPGA設計技術の新たな方向性を見出し、さらにその方向性を決定づける貴重な研究成果となった。

また、上記の考えを発展・応用することで、複数のFPGAデバイスを用いたアーキテクチャエミュレーション環境において、システムを複数のFPGAに分割する手法を考案・構築した[3]。

2. 仮想プロセッサアプローチによる 特定用途向けプロセッサ合成技術の構築

本研究の成果の第二として、特定用途のマイクロプロセッサに注力し、特定のアプリケーションプログラムの記述から、そのアプリケーションプログラムに最適な「命令列」を自動的に合成し、また同時にプロセッサハードウェアそのものも自動合成する新たなフレームワークを考案・構築した[4, 5, 6, 7]。これらの技術の鍵となるのは、計算機内部において特定のプロセッサハードウェア・方式に依存しない「仮想プロセッサ」を考案・構築することで、アプリケーションプログラムが本質的に持つ構造を数理モデル化することを可能とし、その上で、面積/遅延/電力などの制約条件のもと、最適化アルゴリズムを構築することを可能とした点である。アプリケーションプログラムを仮想プロセッサ上で実装し、次に仮想プロセッサを現実のプロセッサに近付けることで、最終的に「最適化」されたプロセッサ構成を得ることを可能とした(図1)。

上記のプロセッサ合成フレームワークの実例として、[4, 7]では、アプリケーションプログラムとして画像処理に着目し、

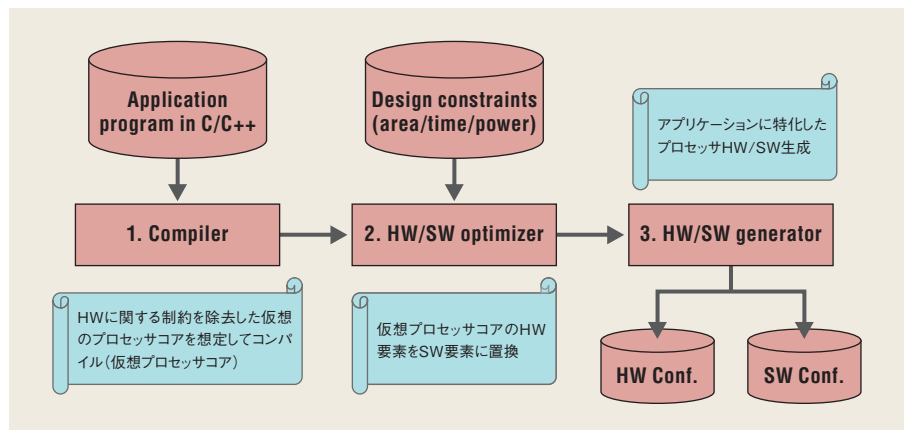


図1 仮想プロセッサアプローチによるアプリケーションプログラムに特化したプロセッサの自動合成フレームワーク。仮想プロセッサを導入することで、まずアプリケーションプログラムが持つ本質的な性質を引き出し(フェーズ1)、その後、プロセッサ設計制約を満足するよう最適化する(フェーズ2)ことを可能とした。

SIMD (Single Instruction Multiple Datastream) 命令と呼ばれる命令セットを念頭に、アプリケーションプログラムに最適な命令列の自動合成を可能としたアルゴリズムを考案し、実際の画像処理アプリケーションを用いてこれを実証している(図2)。

上記のプロセッサ合成フレームワークのもう1つの実例として、【5, 6】では、連想メモリ(CAM; Content Addressable Memory)と呼ばれる、ハードウェア演算機能を備えたメモリを最適構成する設計技術を考案・構築している。人手で設計したCAMプロセッサと同等以上の性能を有するCAM命令列を自動合成することに成功した。

また特に、【8】では、これまでのアーキテクチャ設計自動化技術を用いて、LDPCと呼ばれる次世代の符号化方式に基づく低消費電力化システムLSIを設計し、また実チップとして試作した(図3)。

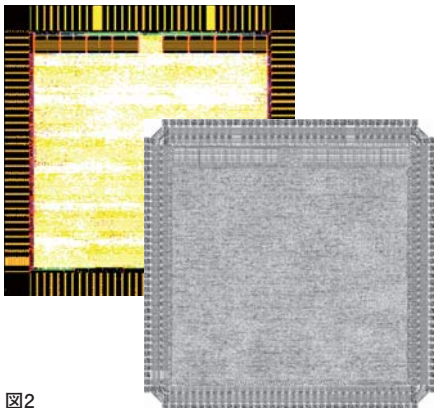


図2 仮想プロセッサアプローチによって生成された実際の画像処理チップ。このチップは、MPEG-4コアプロファイルのリアルタイム符号化を可能とした信号処理プロセッサである。

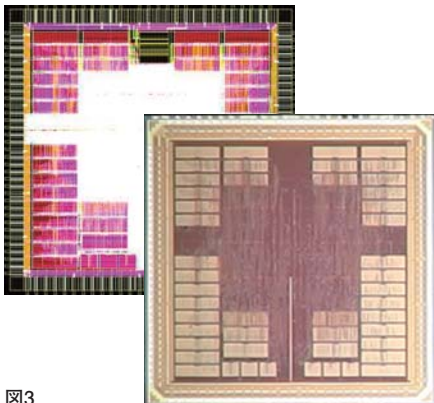


図3 アーキテクチャ自動合成技術を利用して合成したLDPC符号化チップ。IEEE/ACM ISSCC/DAC Student Design Contest で一位を受賞。(http://www.dac.com/43rd/studcon.html)

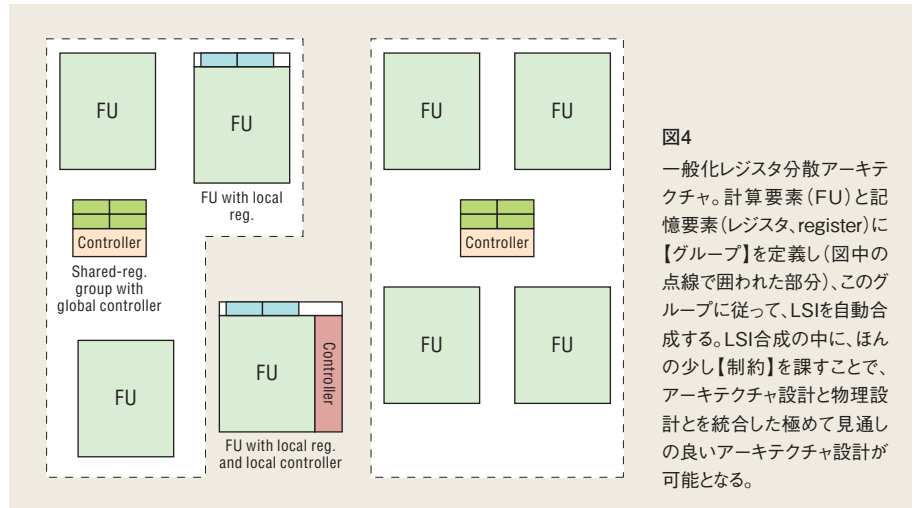


図4 一般レジスタ分散アーキテクチャ。計算要素(FU)と記憶要素(レジスタ, register)に【グループ】を定義し(図中の点線で囲われた部分)、このグループに従って、LSIを自動合成する。LSI合成の中に、ほんの少し【制約】を課すことで、アーキテクチャ設計と物理設計とを統合した極めて見通しの良いアーキテクチャ設計が可能となる。

3. 一般レジスタ分散アーキテクチャモデルによるアーキテクチャ設計・物理設計の統一化技術の構築

本研究の成果の第三として、素子遅延に比較して配線遅延が極めて支配的になる、次世代・次々世代の半導体加工技術を想定し、アプリケーションプログラムの動作記述の段階から、計算素子と記憶素子とをグループ化し近接配置させることを念頭に置いた「一般レジスタ分散アーキテクチャモデル」を考案し(図4)、これに基づくアーキテクチャ設計・物理設計の統一化技術を提案・構築した^[9]。

将来展望

今や1つのLSIが単体の機能を有することはなく、複数の機能が有機的に結合した極めて複雑なシステムを形成するに至っ

ている。それゆえ、その「設計」は職人技が必要とされるが、現在のTime-to-Marketの短縮、Time-in-Marketの短縮は、長時間にわたるLSI設計期間は許されるものではない。できるだけLSI実装の詳細に関与することなく、いかにアプリケーションプログラム開発と同じレベルでLSI開発を行うかが鍵となっている。

本研究の成果はいずれも、従来より抽象度が高いレベル、すなわちアプリケーションプログラムの動作レベルから「直接的」にシステムLSIを設計できるかに焦点を当てたものであり、上記で見たように様々な観点から、これを可能とした。とりわけ、上記の研究成果の3で見た技術は、従来の階層化LSI設計から一歩進んだパラダイムシフトを促すものであり、今後このような技術が不可欠になることは間違いないと考えている。

References(参考文献)

- [1] N. Togawa, M. Sato, and T. Ohtsuki: A performance-oriented simultaneous placement and global routing algorithm for transport-processing FPGAs, *IEICE Trans. on Fundamentals*, vol. E80-A, no. 10, pp. 1795-1806, 1997.
- [2] N. Togawa, M. Yanagisawa, and T. Ohtsuki: Maple-opt: A performance-oriented simultaneous technology mapping, placement, and global routing algorithm for FPGAs, *IEEE Trans. on CAD*, vol. 17, no. 9, pp. 803-818, 1998.
- [3] N. Togawa, M. Sato, and T. Ohtsuki: A circuit partitioning algorithm with replication capability for multi-FPGA systems, *IEICE Trans. on Fundamentals*, vol. E78-A, no. 12, pp. 1765-1776, 1995.
- [4] N. Togawa, M. Yanagisawa, and T. Ohtsuki: A hardware/software cosynthesis system for digital signal processor cores, *IEICE Trans. on Fundamentals*, vol. E82-A, no. 11, pp. 2325-2337, 1999.
- [5] N. Togawa, T. Wakui, T. Yoden, M. Terajima, M. Yanagisawa, and T. Ohtsuki: CAM processor synthesis based on behavioral descriptions, *IEICE Trans. on Fundamentals*, vol. E83-A, no. 12, pp. 2464-2473, 2000.
- [6] N. Togawa, T. Totsuka, T. Wakui, M. Yanagisawa, and T. Ohtsuki: A hardware/software cosynthesis system for processor cores with content addressable memories, *IEICE Trans. on Fundamentals*, vol. E86-A, no. 5, pp. 1082-1092, 2003.
- [7] N. Togawa, K. Tachikake, Y. Miyaoka, M. Yanagisawa, and T. Ohtsuki: A SIMD instruction set and functional unit synthesis algorithm with simd operation decomposition, *IEICE Trans. on Information and Systems*, vol. E88-D, no. 7, pp. 1340-1349, 2005.
- [8] K. Shimizu, N. Togawa, T. Ikenaga, and S. Goto: Low power LDPC code decoder architecture based on intermediate message compression technique, *IEICE Trans. on Fundamentals*, vol. E91-A, no. 4, pp. 1054-1061, 2008.
- [9] A. Ohchi, S. Kohara, N. Togawa, M. Yanagisawa and T. Ohtsuki: Floorplan-driven high-level synthesis for distributed/shared-register architectures, *IPSS Trans. on System LSI Design Methodology*, vol. 1, pp. 78-90, 2008.