

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年 6 月 12 日現在

機関番号：12601

研究種目：若手研究(B)

研究期間：2010～2011

課題番号：22760245

研究課題名（和文）高性能・省面積とプログラマビリティを両立可能なハードウェア自動合成手法

研究課題名（英文）Automatic Synthesis Method of High-Performance, Area-Efficient and Programmable Hardware

研究代表者

吉田 浩章 (YOSHIDA HIROAKI)

東京大学・大規模集積システム設計教育研究センター・特任助教

研究者番号：10456163

研究成果の概要（和文）：システムLSIの開発コスト増大と開発期間短縮に伴い、仕様変更や設計誤りによる製造後修正を可能とする技術の重要性が増している。本研究ではバグ修正など多くの製造後修正が小規模かつ局所的であることに着目し、制御回路の大部分を結線論理で実現した上で部分的にパッチを当てることで制御を修正するパッチ可能アクセラレータを提案し、また修正前後の設計記述からパッチをコンパイルする手法を提案した。例題を用いた評価結果を通じて提案方式が電力効率の面で従来方式よりも優位であることを示した。

研究成果の概要（英文）：With the shorter time-to-market and the rising cost in SoC development, the demand for post-silicon programmability has been increasing. This research proposed a highly energy-efficient accelerator which enables post-silicon engineering change by a control patching mechanism. Then, this research proposed a patch compilation method from a given pair of an original design and a modified design. Experimental results demonstrated that the proposed accelerators offered high energy efficiency competitive to fixed-function accelerators and can achieve about 5X higher efficiency than the existing programmable accelerators.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2010年度	1,500,000	450,000	1,950,000
2011年度	1,200,000	360,000	1,560,000
年度			
年度			
年度			
総計	2,700,000	810,000	3,510,000

研究分野：

科研費の分科・細目：

キーワード：電力効率、製造後機能修正、Engineering Change Order (ECO)、高位合成

1. 研究開始当初の背景

現在の高度情報化社会において、システム LSI は必要不可欠な構成要素の一つとなっている。近年の半導体技術の向上の一方で、製造技術は非常に複雑となっており、製造時の故障やばらつきによる歩留まりの低下が問題となってきている。またシステム LSI の大規模化により設計プロセスは非常に複雑となっており、設計誤りによる手戻りが多数発生し、結果として開発コストが肥大化している。

近年では設計生産性の向上を目指し、C 言語などを用いて高い抽象度で設計を記述し自動的にハードウェアを生成する高位合成技術が広く利用されている。しかしながら、生成されるハードウェアは一般的に機能が固定されているため、製造後の修正は不可能である。このような背景から、ハードウェアに柔軟性を持たせることで製造後の機能修正を可能とする実装方式としてプログラマブルアクセラレータ方式が提案されている。しかしながら、これらの方式では再構成可能にするために大規模なメモリ素子を用いており、性能や面積のオーバーヘッドが大きく実用的ではない。

2. 研究の目的

一般的にシステム LSI の設計は慎重に行われるため、製造後に機能全体の変更が必要となることは稀であり、多くは設計誤り・バグの修正や仕様変更に伴う小規模かつ局所的な修正である。従来方式であるプログラマブルアクセラレータは機能全体を変更可能なプログラマビリティを持っているが、一方で性能や面積のオーバーヘッドが非常に大きくなっている。本研究では、プログラマビリティが可変な方式を提案し、ユーザの所望のプログラマビリティを実現した上で性能や面積効率が最適なハードウェアを自動合成する手法を提案する。

所望のプログラマビリティをどのように与えるかという課題に対しては、本研究ではプログラマビリティの用途に応じた指標で指定する方法を提案する。ハードウェア高位合成の最新技術を基礎として、プログラマビリティの用途に応じた設計の多様性を制約として与えることを可能にすることで、必要最小限のオーバーヘッドを持つプログラマブルなハードウェアを合成する技術を確立し、またその設計環境を構築することを目的とする。

3. 研究の方法

本研究では、目的達成に向けて解決する必要のある下記の 4 課題を設定し、各項目の研究・開発を行う。

(1) プログラマビリティの用途に応じた設計多様性の指定方法の提案

プログラマビリティが必要とされる状況として、製造時故障の修復、設計誤りの修正、仕様変更に応じた機能変更、単一ハードウェアでの複数機能の実行、などを想定している。各状況に対して、現実的な指標で設計多様性を指定できるような方法を提案していく。例えば、プログラマビリティを故障修復に使用したい場合には「各機能ユニットの故障率が X% の時、性能の期待値が Y」という指定方法が考えられる。またハードウェアに仕様変更に対する耐性を持たせたいのであれば、「設計記述の X% が変更されても、性能の最悪値が Y 以上」といった制約も考えられる。

(2) 設計の多様性を制約として考慮可能な高位合成手法の提案

(1) で提案した方法で指定された設計多様性を制約として、性能や面積が最適なハードウェアを自動合成する手法を提案する。設計多様性は確率的・統計的な事象を扱っているため、性能や面積についても分布を持った統計的な値として扱わなくてはならない。このような統計的な値を制約・評価関数として最適化するアルゴリズムは一般的にモデル依存になってしまい、適用可能範囲が限定されてしまう。本研究では、現実的なアプローチとして、与えられた設計多様性に基づいて多数の異なる設計記述を生成し、これらの設計集合内の各設計に対して高位合成を適用する。このアプローチを用いることで、高位合成手法は分布を持たない性能・面積値のみを考慮すればよく、従来手法を活用できる。

具体的には、まず対象とするアプリケーションの C 言語記述と設計多様性として故障耐性や仕様変更耐性の仕様が与えられる。これらに基づいて、元のアプリケーション記述に対して設計ばらつきが挿入され、多数の異なる設計記述が生成される。次に最小限の構成要素からなる初期ハードウェアを生成する。インクリメンタル高位合成手法は、各設計記述に対して、制約を満たすように初期ハードウェアに機能ユニットや配線を追加していく。この手法は、単一アプリケーションに対してゼロからハードウェアを合成する従来手法とは異なる手法である。

(3) プロトタイプツールの実装

研究代表者はすでに高位合成の研究を進めてきており、その中で独自の高位合成ツールを実装している。このツールの入力としては、一般的な高位合成ツールと同じく C 言語を用いており、C 言語の構文解析や依存解析にはコンパイラ・インフラストラクチャ LLVM を用いている。そのツールを基にして、設計ばらつき挿入ツールおよびインクリメンタル高位合成ツールを実装していく。特にインクリメンタル高位合成部分は多数の設計記述に対して合成を繰り返し行うため、効率良く合成を行う必要がある。本研究では商用コンパイラにも使用されている Swing Modulo Scheduling と呼ばれる手法を基に実装を進めていく予定である。

(4) FPGA・ASIC を用いた有効性の実証

評価用例題としては JPEG や H. 264 などの画像処理プログラムを使用する。これらのプログラムは過去のバージョンも入手可能なため、提案手法が合成したハードウェアが複数の異なるバージョンを効率的に実行することを示すことで本研究の有効性を実証する予定である。

まずは JPEG や H. 264 の重要な構成要素である逆コサイン変換 (IDCT) の部分のみを対象にしてハードウェア合成を行い、FPGA ボード上に実装する。次にアプリケーション全体に対して実際にチップ試作を行い、実際に動作させることで実用的にも有用であることを示す。

4. 研究成果

当初計画では、複数アプリケーションの実行といった比較的大規模なプログラマビリティが必要な用途も想定していたが、実用的に最も重要であると考えられるバグ修正・仕様変更を主な対象に絞り、課題設定の見直しを行った。その上で研究を実施し、下記に示す 4 つの成果を得た。

(1) 専用ハードウェアに匹敵する電力効率で機能修正を可能とするパッチ可能アクセラレータ方式の提案

バグ修正や仕様変更など多くの製造後修正が小規模かつ局所的であることに着目し、制御回路の大部分を結線論理で実現した上で部分的にパッチを当てることで制御を修正するパッチ可能アクセラレータ方式を提案した。制御回路全体をメモリで実現する従来

方式とは異なり、制御の大部分は効率の良い結線論理回路で実行し、修正が必要な場合にものみメモリ素子を用いることで専用ハードウェアに匹敵する電力効率を実現可能である。

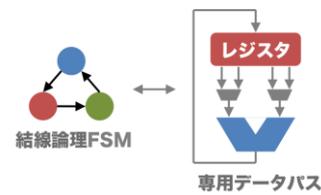


図1 専用ハードウェア方式(製造後修正不可)



図2 プログラマブルアクセラレータ方式(従来方式)

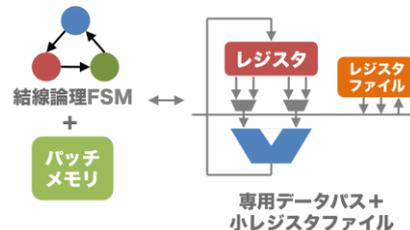


図3 パッチ可能アクセラレータ方式(提案方式)

画像処理などで広く用いられている逆離散コサイン変換を例題とした評価実験では、提案方式は専用ハードウェアに比べて数%の電力効率低下で機能修正が可能であることを示し、また従来方式に比べて 5 倍以上の高い電力効率を達成可能であることを示した。

(2) パッチ可能アクセラレータ向け自動合成技術

最近のシステム LSI の設計では、C 言語などの高位記述言語で設計を記述し、高位合成ツールによってハードウェアを自動合成する手法が広く利用されている。本研究で提案したパッチ可能アクセラレータの広い利用を目指し、C 言語を入力とした自動合成技術を提案した。

本技術は①初期設計段階向けパッチ可能アクセラレータの自動合成手法と②設計修正が行われた場合のパッチ生成手法の 2 手法からなる。パッチ可能アクセラレータ自動合成手法は既存の高位合成手法に基づいており、初期設計記述に対して想定される変更の割合を入力として、必要量のメモリ素子が自動

的に挿入される。

また、パッチ生成手法では修正前後のCプログラムを入力として、2プログラム間の差分を特定し、その差分からパッチを生成する。高い電力効率を達成するためには、パッチを最小化することが重要である。そのため、2プログラム間の最小差分を求める手法を提案した。この手法では、変数名の変更など機能に関係のない修正に影響されず、機能の変更部分のみを求めることが可能であるため、パッチ適用後においても高い電力効率を達成することが可能である。

(3)パッチ可能アクセラレータ向け設計ツールプロトタイプの開発

(2)の提案技術に基づいたパッチ可能アクセラレータ向けの設計ツールのプロトタイプを開発した。このプロトタイプはコンパイラ・インフラストラクチャ LLVM に基づいて実装されており、C 言語を入力設計記述として、Verilog HDL 言語で記述されたパッチ可能アクセラレータを自動生成する。また、C 言語記述が変更された場合には、最小差分を自動的に特定し、パッチを生成する。高位合成分野において広く利用されているベンチマーク CHStone を例題に用いて実際にパッチ可能アクセラレータの生成・パッチ適用を行い、実用的な例題に対しても自動合成技術が実現可能であることを示した。

(4) H. 263 動画再生システムを用いた提案方式の有効性実証

パッチ可能アクセラレータを用いて実際に H. 263 動画再生システムを構築した。このシステムは Xilinx 社 ML605 評価用ボードを用いており、ボード上の Virtex-6 FPGA 内に Microblaze プロセッサとパッチ可能アクセラレータを実装し、外部液晶モニタに動画を表示する。H. 263 再生機能の大部分は Microblaze 上のソフトウェアで実現されているが、性能ボトルネックとなる逆離散コサイン変換の計算部分をパッチ可能アクセラレータで実現している。初期設計では逆離散コサイン変換にバグがあり再生画像に乱れがあるが、パッチを適用することで正しい再生画像を得た。面積・性能・消費電力についても専用ハードウェアに匹敵していることを確認した。この実験を通じて、提案技術の実現可能性および有用性を実証した。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 0 件)

[学会発表] (計 6 件)

① Hiroaki Yoshida and Masahiro Fujita, “An Energy-Efficient Patchable Accelerator For Post-Silicon Engineering Changes,” IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Oct. 2011.

② Hiroaki Yoshida and Masahiro Fujita, “A Highly Energy-Efficient Accelerator Enabling Post-Silicon Engineering Changes and Its Patch Compilation Method,” Work-In-Progress Session, ACM/IEEE Design Automation Conference (DAC), June 2011.

③ 吉田浩章, 藤田昌宏, “インクリメンタル高位合成に向けた設計記述間差分の計算手法,” 情報処理学会研究報告, vol. 2011-SLDM-149, no. 21, pp. 1-6, 2011年3月.

④ 吉田浩章, 藤田昌宏, “仮想マルチプロセッサモデルに基づく高速 SoC プロトタイプピンギ手法,” 電子情報通信学会技術研究報告, vol. 110, no. 316, pp. 7-12, 2010年11月.

⑤ 吉田浩章, 藤田昌宏, “動的パッチ読み出し機構を備えた製造後機能修正可能アクセラレータ,” 情報処理学会研究報告, vol. 2010-SLDM-146, no. 6, pp. 31-36, 2010年10月.

⑥ 吉田浩章, 藤田昌宏, “製造後機能修正可能な高電力効率アクセラレータの高位設計手法,” 情報処理学会 DA シンポジウム 2010 論文集, pp. 45-50, 2010年9月.

[産業財産権]

○出願状況 (計 2 件)

名称: Accelerator and data processing method

発明者: 吉田浩章, 藤田昌宏

権利者: 東京大学

種類: 特許

番号: U.S. Patent Application 61/446208

出願年月日: 2012年2月23日

国内外の別: 国外

名称：アクセラレータ及びデータ処理方法
発明者：吉田浩章, 藤田昌宏
権利者：東京大学
種類：特許
番号：特願 2010-193136
出願年月日：2010 年 8 月 31 日
国内外の別：国内

6. 研究組織

(1) 研究代表者

吉田 浩章 (YOSHIDA HIROAKI)
東京大学・大規模集積システム設計教育研
究センター・特任助教
研究者番号：10456163