平成 19 年度
実績報告

「ディペンダブル VLSI システムの基盤技術」
平成 19 年度採択研究代表者

坂井　修一

東京大学大学院情報理工学系研究科　教授

アーキテクチャと形式的検証の協調による超ディペンダブル VLSI

1. 研究実施の概要

本研究課題は、検証技術とディペンダブルアーキテクチャ技術の２つを核としてこれにテスト技術・回路技術を加え、それぞれを新規に研究開発するとともに、これら諸技術の協調・融合によって、個々の技術では達成できないディペンダビリティを VLSI 上に実現する技術を研究開発する。このように、個々の技術を磨きながら各技術の協調融合によってかつてなかったディペンダビリティを実現することは、JST 研究開発戦略センターの戦略イニシャティブ「情報化社会の安全と信頼を担保する情報技術体系の構築 ―ニュー・ディペンダビリティを求めて―」にある「従来のようにフォールトを予防する努力だけでは不十分であり、フォールトの存在を前提とする情報化社会のデザインあるいは情報システムの設計方法論が必要である。たとえ一部の要素にフォールトが発生したとしてもシステム全体としては期待どおりの良質で確かなサービスを提供し続けるディペンダブルな情報システムを実現する新しい情報技術の体系が求められる」という記述の実現をめざしたものである。

初年度である平成 19 年度は、形式的検証方式・テスト方式・製造後回路修正機能の検討、回路要素技術・アーキテクチャ要素技術の検討、プロセッサシミュレータ整備、などを行った。形式的検証に関しては、特に、等価性検証ツール開発のフレームワーク構築、および、演算系回路に対する等価性検証手法の研究を行い、有効性を示した。回路に関しては、タイミング故障耐性をもつフリップフロップを新規提案し、評価して有効性の検証を行った。アーキテクチャに関しては、故障検知・回復を行う超ディペンダブルアーキテクチャを提案するとともに、ソフトウェアによるプロセッサシミュレータ、FPGA によるテストベッドなどを開発・整備した。さらに、このようなアーキテクチャを形式的検証するための初期検討を行った。

2. 研究実施内容

（文中にある参照番号は 4.(1)に対応する）

2.1　ディペンダブルアーキテクチャグループ

ディペンダブルアーキテクチャグループでは、回路技術とアーキテクチャ技術によってVLSIの信頼性を飛躍的に向上させる技術の研究開発を行っている。

平成19年度は、第一に、ディペンダブル回路として、信号遷移タイミング監視に基づく遅延補償フリップフロップの提案と予備評価を行った[1]。

タイミングエラーを実行時に検出する方式として、すでにRazorとCanaryが提案されている。Razorは遅延クロックによって動作するシャドーラッチを用いてタイミングエラーを検出する技術だが、ショートパスが発生する可能性があり、また、メタステーブル防止のための負荷回路が必要になる問題があった。Canaryはシャドーラッチの入力に遅延回路を入れて表のラッチの値と比較するもので、ショートパス問題は解決するが、通常のラッチより早く信号が到着する必要があり、はじめからマージンを大きくとらなければならない欠点がある。

本研究で提案した遅延保証フリップフロップは、信号の遷移を表すパルスを用い、フリップフロップに入力されるクロックとデータの信号遷移タイミングを評価するもの（図１）であり、これによってRazor、Canaryの問題点を解決する。また、最終的な結果に影響を及ぼさないようなエラーを許容することにより、タイミング制約を緩和することができる。これらの効果について評価を行い、有効性を確認した。

遅延保証フリップフロップでは、データ入力信号の遷移タイミングを動作時に監視し、タイミングエラーの発生を検出あるいは予測する。そして、タイミングエラーの発生が検出されると、フリップフロップが誤ったデータを取り込まないようにデータの取り込みタイミングを調整する。こうして得られた回路動作時のプロファイルをもとに、電源電圧や動作周波数を調整することによって、電源揺らぎや、製造時のプロセスばらつきによって発生する予測しがたいタイミングエラーを回路動作時に自動的に回避することが可能となる。また、動作時にエラーの発生状況をモニタリングするので、これらの外乱に対する設計段階でのマージンを削減することが可能であり、動作環境に合わせて電源電圧を調整できることは省電力化効果ももつことになる。さらに、動作時信号監視システムを実現するRazor、Canaryなどの既存手法と比べて、遅延補償フリップフロップは回路のタイミング設計自由度を高く保つことができる。

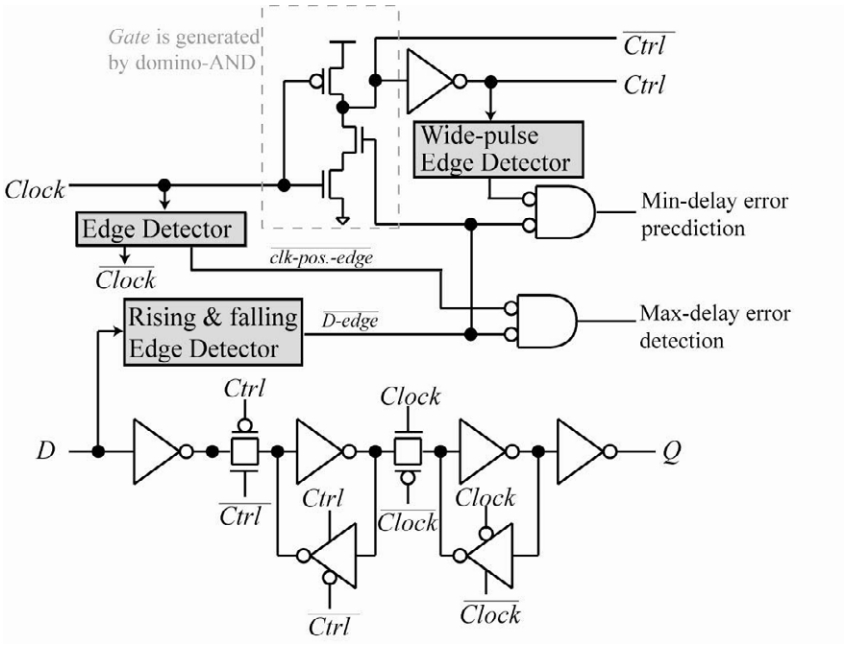

図１．タイミング故障を動的に検知・修正するフリップフロップ

本研究ではこの遅延保証フリップフロップのタイミングエラー耐性に関する回路動作レベルでの評価検討を行い、既存手法との対比も行いながらその特徴を明らかにした。その結果として、電源電圧の揺らぎに対するタイミングエラー耐性の高さや、クロックゲーティング・ＤＶＦＳなどの既存省電力化技術と併用した場合の高い省電力化効果を確認した。また実際に集積回路システム中に遅延補償フリップフロップを適用する場合に必要となる遅延補償の影響やメモリセルにおけるエラー監視などの要素技術についても考察を行った。

この遅延保証フリップフロップは、本研究の最終成果である超ディペンダブルVLSIにおいて、設計時に遅延時間の保証が（テストに時間とコストがかかりすぎるなどの理由で）困難である箇所に用いられる。

第二に、坂井・五島研究室で開発したプロセッサシミュレータ「鬼斬」をディペンダビリティ実験・検証用に改良した。「鬼斬」は、アウトオブオーダ処理を行うスーパスカラプロセッサの cycle accurate な汎用ソフトウェアシミュレータであり、新アーキテクチャに対する動作確認・性能測定用に用いられる。改良された「鬼斬」は、本研究では、1) VLSIのディペンダビリティを高めるためのアーキテクチャ技術が正しく動作するか、2) それによってプログラム動作の正当性が維持されるか、3) それによって性能にどのような影響が及ぶか、を検証するためのものである。従来のプロセッサシミュレータは、基本パイプラインの性能評価を行うことはできるが、この３点に適したものは存在しなかった。本 CREST では、タイミングエラー検知・回復用アーキテクチャ、耐永久故障アーキテクチャなどの

研究を行うため、これらの要件を満足するシミュレータが必要であり、今回の整備が必須であった。

また、FPGA によるテストベッドを用いた実験環境の整備を行い、設計ミスや製造エラーに対する耐性をアーキテクチャで受け持つ方式について実験を行った（図２）。本テストベッドは前 CREST「ディペンダブル情報処理基盤」において設計・試作されたものであり、前 CREST では、これの上で電源電圧、クロック周波数を操作してフォールト、エラーが起きること、これを検出することができることを確認した。

本 CREST では、初年度の研究として、エラーによって生じた間違った実行結果によってプロセッサステートを更新しないためのフレームワークを設計した。これは、検出されたエラーを含む実行結果に、エラー情報を付加して伝搬するネットワークと、その情報の付いた実行結果によっては絶対に更新されないようなレジスタファイルの構成からなる。この機構を、前 CREST で設計・試作した超ディペンダブルテストベッドの上で FPGA 回路を新たに設計することで実現し、所期の動作ができることを確認した。

図２．超ディペンダブルプロセッサ・テストベッド
（ハードウェアは前 CREST において開発、FPGA 回路を新規開発）

第三に、耐永久故障アーキテクチャに関して新規提案を行い、これを検証するための FPGA を用いたテストベッドを設計・試作した。本提案は、VLSI の永久故障時に、制御部・実行部の区別なくアーキテクチャを再構成可能としたもの（図３）であり、本テストベッドでは、複数の FPGA からなるネットワークによって、このアーキテクチャをエミュレートする。ユーザ・ロジック用と耐故障ネットワーク用にそれぞれ 16 個の FPGA を用いたボードを本 CREST で新規に設計・試作した（図４、これは図２の超ディペンダブルプロセッサテストベッドとは別のものである）。平成 20 年度は本ボードを用いた詳細提案と評価が課題となる。

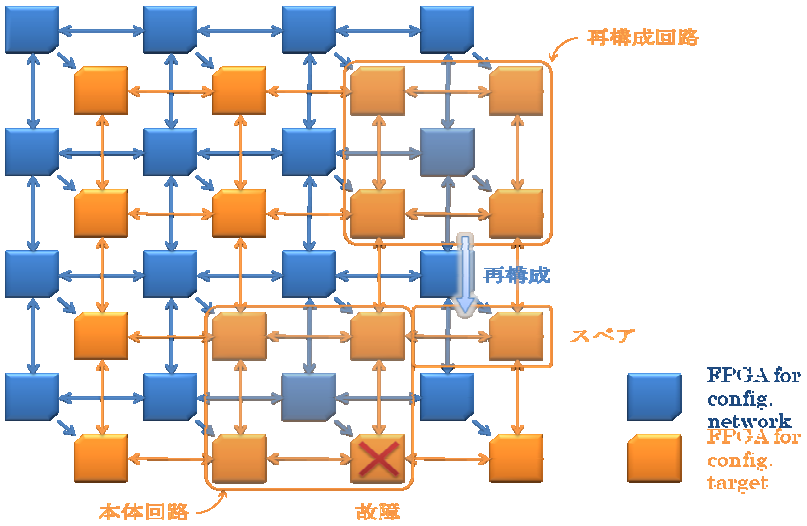

図３．耐永久故障アーキテクチャ

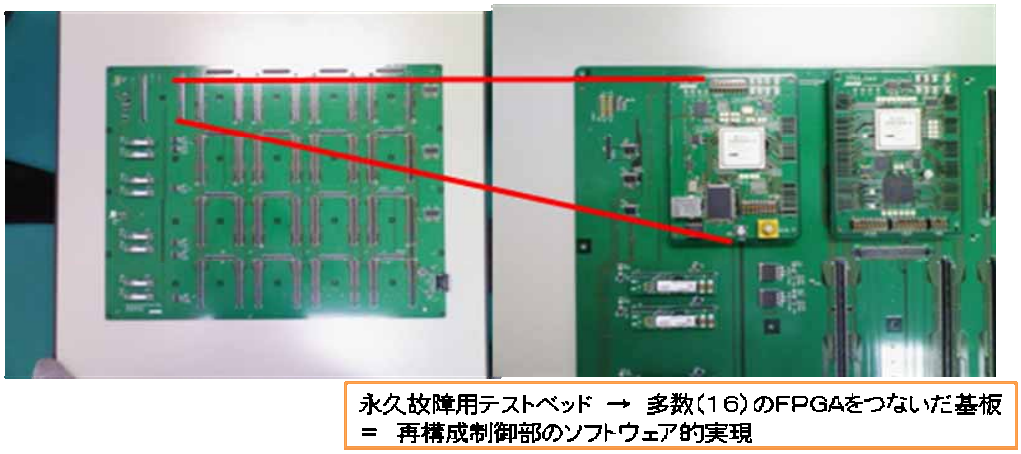

図４．耐永久故障用テストベッドボード（ハードウェア新規開発）

2.2　形式的検証グループ

形式的検証グループでは、形式的検証の適用による VLSI システム設計の信頼性向上を目指して研究を進めており、特に、上位設計において適用可能な形式的検証技術の研究開発を対象としている。その中でも、設計記述を詳細化・最適化していく際に、等価性を検証することは、設計誤りの検出に有効である。本年度は、等価性検証ツール開発のフレームワーク構築、および、演算系回路に対する等価性検証手法の研究を行った。等価性検証ツールの内部構成図を図５に示す。設計記述は、初めに拡張システム依存グラフ(ExSDG: Extended System Dependence Graph)に変換され、指定された等価性を設計記述が満たすかどうかを形式的に検証する。加えて、検証ツールは、設計解析・典型的な設計誤りの発見、

デバッグ支援、などの検証・デバッグに有用な機能を追加することが可能な構成となっている。これは、本研究で用いる ExSDG に、設計解析・検証に必要な抽象構文木や依存関係の情報が全て含まれているためである。今年度の等価性検証ツール開発では、ツール内で共通に検証・解析に利用する拡張システム依存グラフ(ExSDG: Extended System Dependence Graph)の構築と関連インターフェースの整備、基本となる記号シミュレーションに基づく検証手法の実装、ExSDG の静的解析によって設計記述中の設計誤りを検出する手法の考案と実装[2]を行った。現在、逆離散コサイン変換や楕円フィルタにおける最適化前後の等価性を検証することができる。設計中に含まれる典型的な誤りの静的検出手法では、ExSDG を解析することにより、変数の未初期化やデッドロックの検出を数千行規模の設計記述に対して実施可能であることが確認できた。

演算系回路に対する等価性検証手法の研究では、線形テイラー展開グラフ(LTED: Linear Tailor Expansion Diagram)を利用した手法の提案を行った[3]。LTED では、二分決定グラフ(BDD: Binary Decision Diagram)に比べて、ワードレベルの演算式をより少ないメモリ量で表現することができる。この性質を利用し、2 つの演算系回路で実行される算術式を LTED で表現し、等価性検証や内部等価点検出を効率的に行う手法を提案した。評価実験では、Phase-Shift Keying や Digital Image Rejection Unit などの検証が、BDD や論理式の充足可能性判定手法を利用した従来の手法に比べて高速に実行できることが確認できた。

形式的検証においては、大規模な設計記述を扱う際には、設計の抽象化が必要となる。しかし、抽象化された設計において発見された反例が実際の設計では実現できない場合、抽象化を修正する必要がある。このとき、従来手法に比べて、より効率的に修正を行う手法を提案した[4]。提案手法では、実用的な検証問題に対して、従来手法に比べてより少ない predicate による抽象化の修正が可能であることが確認できた。

詳細化が進むと、設計は最終的に基本セルの回路へと変換される。このとき、使用されているセルの種類が少ない方が検証を適用する際に有効であると考えられる。我々は、与えられた性能制約下において回路中で使用するセルの種類を最小化する問題の定式化と効率的解法を世界で初めて提案した[5]。評価実験では、性能最適回路と同等性能で平均 74%のセル種類削減が可能であることを確認した。

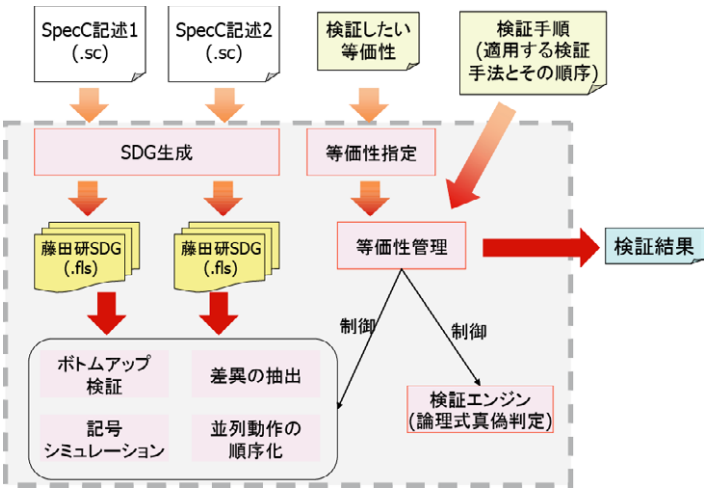

図5. 等価性検証ツールの構成

なお、ここでターゲットとしている設計は、MPEG エンコーダハードウェアから本研究提案の超ディペンダブルプロセッサ本体までの大規模なものであり、現在、これらについて具体的に進めている。

2.3　グループ間共同の研究開発

2.2 で述べたアーキテクチャについて、これを形式的検証するための初期検討を行った。

3. 研究実施体制

(1)「ディペンダブルアーキテクチャ」グループ

①研究分担グループ長:坂井　修一(東京大学、教授)

②研究項目

- タイミング故障耐性をもつフリップフロップの提案・予備評価
- ディペンダブルプロセッサシミュレータの研究開発
- FPGA を用いたテストベッドによる実験環境の整備
- 設計ミスや製造エラーに対する耐性をアーキテクチャで受け持つ方式の検討・提案
- 耐永久故障アーキテクチャ用テストベッドの研究開発

(2)「形式的検証」グループ

①研究分担グループ長:藤田　昌宏(東京大学、教授)

②研究項目

- 上位設計記述に対する形式的等価性検証ツール開発

- 演算系回路に対する効率的な等価性検証手法
- 大規模設計記述の等価性検証のためのボトムアップな検証手法
- C ベース言語による上位設計記述の理解・マニュアル作成のための解析技術
- プログラマブル素子の挿入による In-field 回路デバッグ技術

4. 研究成果の発表等

(1) 論文発表 (原著論文)

[1] Kenichiro Hirose (The University of Tokyo), Yasuo Manzawa (The University of Tokyo), Masahiro Goshima (The University of Tokyo), and Shuichi Sakai (The University of Tokyo): Delay-Compensation Flip-Flops for Timing-Error Tolerant Circuit Design, Int'l Conf. on Solid State Devices and Materials (SSDM), pp. 480—481, 2007.

[2] Shunsuke Sasaki (The University of Tokyo), Tasuku Nishihara (The University of Tokyo), Daisuke Ando (The University of Tokyo), Masahiro Fujita (The University of Tokyo): Hardware/Software Co-design and Verification Methodology from System Level Based on System Dependence Graph, Journal of Universal Computer Science, Vol. 13, No. 13, pp. 1972-2001, 2007.

[3] Bijan Alizadeh (The University of Tokyo), Masahiro Fujita (The University of Tokyo): Automatic Merge-point Detection for Sequential Equivalence Checking of System-level and RTL Descriptions, 5th International Symposium on Automated Technology for Verification and Analysis, pp.129-144, Tokyo, Japan, October 2007.

[4] Thanyapat Sakunkonchak (The University of Tokyo), Satoshi Komatsu (The University of Tokyo), Masahiro Fujita (The University of Tokyo): Using Counterexample Analysis to Minimize the Number of Predicates for Predicate Abstraction, 5th International Symposium on Automated Technology for Verification and Analysis, pp.553-563, Tokyo, Japan, October 2007.

[5] Hiroaki Yoshida (The University of Tokyo), Masahiro Fujita (The University of Tokyo): Performance- Constrained Different Cell Count Minimization for Continuously-Sized Circuits, Design, Automation & Test in Europe, pp.1099-1102, Munich, Germany, March 2008.

(2) 特許出願

平成 19 年度　国内特許出願件数 1 件(CREST 研究期間累積件数:1 件)