

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. 研究の概要

1. チーム全体の研究の概要

①本研究の背景、社会や産業に存在する問題と本研究の課題

社会インフラとしての情報システムの重要性は加速度的に増しており、その基盤となる VLSI のディペンダビリティは社会の安全・安心に欠かせないものとなっている。VLSI のディペンダビリティは、従来、シミュレーションによる動作検証や冗長回路による多重実行などの手段によって確保されてきたが、VLSI の微細化・複雑化と用途の拡大によって、従来にはなかった高度なディペンダビリティが求められるようになった。

本研究課題は、VLSI システムのディペンダビリティを飛躍的に向上させる技術を新規提案し、その有効性を検証することを目標とする。特に従来は得ることのできなかつた VLSI の信頼性の獲得を飛躍的に高める技術を提案し、シミュレーションと試作機の製作・評価を通してこれを検証する。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。

②本研究チームの達成目標。

◆形式的検証技術

- (1) C/C++言語ベース VLSI 高位設計において、形式的検証手法を高度に進めることで、近未来のマイクロプロセッサを含む VLSI 設計の信頼性を飛躍的に高める。
- (2) 上位設計からテストを陽に意識しテスト容易化・検証容易化を実現する設計手法の確立を通してテスト・検証コストの大幅な削減を図るとともに、必要に応じてフィールドプログラマブル性を部分的に導入可能な合成手法も併用することで、テスト段階において設計の部分的修正を可能とし、形式的検証でとりきれない設計ミスや製造上の故障についても、テスト段階でとることができるようにする。

◆回路・アーキテクチャ技術

- (3) 形式的検証、テスト段階の修正でも取り切れない製造ミスや故障について、これを回路技術で緩和する。具体的には、タイミング故障検出・回復のための回路についての研究開発および評価を行う。
- (4) 以上の(1)(2)(3)の手段においてとりきれない設計ミスや製造故障をアーキテクチャのレベルで解決するディペンダブルアーキテクチャを開発する。具体的には、タイミング故障防止・検出・回復のためのアーキテクチャの研究、永久故障検出・回復のためのアーキテクチャの研究開発を行う。(3)(4)の回路技術・アーキテクチャによる解決において最も特徴的な点は、従来最悪値の積算でなされていた VLSI 設計を、「典型値+回路・アーキテクチャによる補正」によって実現可能とすることである。

◆統合技術

- (5) ディペンダビリティ向上のために、(4)で提案したディペンダブルアーキテクチャ技術を形式的に検証する。
- (6) 既存のアーキテクチャに加えて最新のアーキテクチャを形式的検証する。具体的には、マルチコア

- やマルチスレッドなどに含まれる新しい機構・アーキテクチャについて、その形式的検証を行う。
- (7) 各設計階層間のディペンダビリティ役割分担を最適化し（どの設計階層で実現するのが最適かを定めるなど）、最終的にこれまでの手段では得られなかった VLSI のディペンダビリティを達成する。以上本課題の遂行によって、多階層の機構によって、総合的に VLSI のディペンダビリティを実現する（図1）基盤が築かれる。

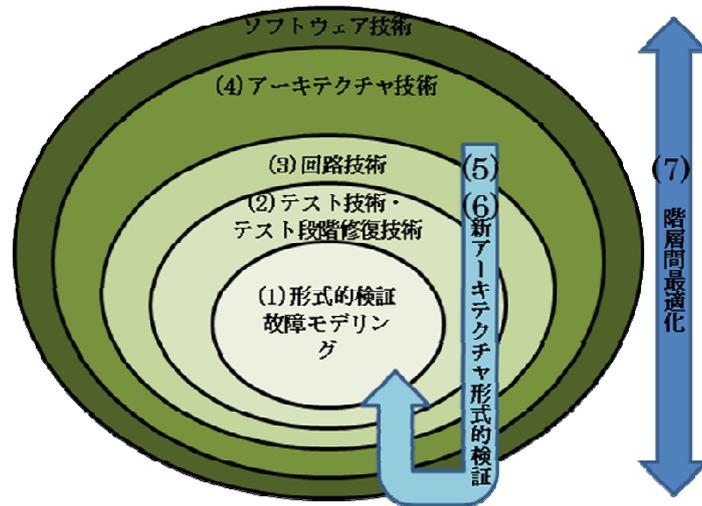


図1. ディペンダビリティの階層

③本研究のアプローチ

等価性検証を高度に進めることで、適用規模を従来の一桁以上拡大してプロセッサや SoC など大規模システムまで拡大し、かつ、現実的な時間内に検証できるようにすることで、真に実用的な形式検証を実現する。

デザインばらつきに対する耐性をもつ回路・アーキテクチャ技術を新規開発し、VLSI のタイミング制約を緩和させるとともに、動作速度を数十%以上向上させる。この技術は、サーバ用・組み込み用を問わず、あらゆるデジタル VLSI で用いることが可能であるが、現在はサーバ・PC 用途の汎用プロセッサ LSI に搭載することを考え、知財化などを進めている。

宇宙・深海などの人間が修理することの困難な場所で起こる故障に対して、動的再構成技術を柔軟に用いることでこれを回避・回復し、低コスト・高信頼の VLSI システムを構築できるようにする。再構成可能アーキテクチャを有効利用するだけでなく、「再構成制御そのものを仮想化する」という斬新なアイデアを導入し、故障箇所によらず修復可能なアーキテクチャを実現する。

故障を自動で回避したり、自動多重実行したりする機構をルータ内にもつメニーコア VLSI を提案・試作し、真にディペンダブルなプロセッサ VLSI の将来像を明らかにする。これによって、複雑なソフトウェアを構築することなく、プログラム実行時に高いディペンダビリティが確保されるようになる。

④研究実施方法

1) 本研究チーム運営の方針、研究グループ間の分担・協力関係

形式的検証、アーキテクチャ、ディペンダビリティ支援ルータの3つのグループを置き、それぞれ検証、回路・アーキテクチャ、メニーコアの研究を受け持つとともに、後2者がディペンダブルアーキテクチャやメニーコアプロセッサの設計データを形式的検証グループに提供し、形式的検証グループがこれらの検証を行う。また、メニーコアプロセッサの中でアーキテクチャグループの耐タイミング故障技術の利用を考える。

それぞれの技術について数ヶ月に一度綿密な情報交換を行い、進捗や研究協力に関する確認や、必要に応じて新規研究課題の追加などを行う。こうした場では、さらに、VLSI のディペンダビリティをどの階層でど

のように実現するべきか討論を行い、コストを含む最適化について検討する。

2) 領域外部の企業等との連携

- ・等価性検証ツールを国内外の5大学、5企業にNDA締結の上で提供し、評価を行っている(詳細は2で記述、以下同様)
- ・ポストシリコンデバッグ・検証について、スタンフォード大学、トロント大学と共同研究を進めている。
- ・算術演算回路の解析について、ドイツ・Kaiserslautern 大学に対してプロトタイプツールを提供し、従来手法より大幅に性能が改善することを確認した。
- ・永久/過渡故障耐性をもつFPGAアーキテクチャについて、JAXA 斎藤教授との間で、宇宙用高信頼デバイスに関する情報交換を行っており、この分野への展開をはかっている。

3) 領域内他研究チームとの連携関係

- ・安浦チームとの間で、ディペンダブルデザインと耐タイミング故障技術に関する共同の検討を行っている
- ・米田チームとの間で、NoC、メニーコア技術に関する共同の検討を行っている。上位層を本チーム吉瀬グループが、階層を米田チームが分担する方向で検討中である。

2. 研究グループの研究の概要

(1)「ディペンダブルアーキテクチャ」グループ(坂井グループ)

①本研究グループの研究課題、ならびに所属する研究チームの課題との関係

本研究グループは、回路技術・アーキテクチャ技術によって VLSI のディペンダビリティを向上させることを目標に、研究を進める。さらに、ディペンダブルプロセッサの設計データを形式的検証グループに提供し協力してその動作検証を行うとともに、形式的検証グループおよびディペンダビリティ支援ルータグループと協力して、VLSI システム上でのこれら技術の組合せと最適な役割分担の検討・評価を行う。

回路・アーキテクチャ分野で本グループが受け持つ研究課題は以下の3点である。

- a. タイミング故障耐性を持つプロセッサの構成方式
- b. タイミング故障耐性をもつクロッキング方式
- c. 永久/過渡故障耐性をもつFPGAアーキテクチャ

研究課題 a 及び b については、回路に発生するタイミング故障を検出し、回復するための要素技術である。半導体プロセスの微細化に伴い、チップ内の回路遅延のばらつきが増大している。従来の設計手法では、最も遅延の大きいパスの最悪のケースに合わせて電圧や、クロック周波数を決定してきた。このようなワースト・ケース設計は、チップ内ばらつきが増大するに従ってマージンの量が大きくなり、悲観的になりすぎる傾向がある。その結果、このままワースト・ケース設計を続けると、近い将来、微細化による性能の向上が見込めなくなる可能性がある。

研究課題 c は、高信頼の LSI を安価に実現することを目指したものである。高信頼性が要求される分野(航空宇宙、原子力、医療など)においては、そのシステムで用いられる LSI は当然、高信頼のものでなければならない。高信頼部品の代表的なものに MIL 規格品(米軍仕様の高信頼部品)が存在するが、これらは非常に高価で、入手性が悪い。また、新規に専用の LSI を開発する選択肢も、コスト的に現実的でない。このような理由から、高信頼システムの開発は、部品代だけでも多額の費用がかかるという問題を抱えている。

②本研究グループの達成目標。

本グループの研究課題 a, b における目標は、DVFS (Dynamic Voltage and Frequency Scaling) と動的タイミング故障検出・回復を組み合わせることで、実際的な電圧、周波数での動作を可能とすることである。

そのために必要となるのは、タイミング故障を検出した際に、正しくプロセッサを回復させることである。研究課題 a では、プロセッサ内でタイミング故障によって影響の受けた情報を確定させずに、正しく再実行する仕組みを提案、実装することを目指す。

また、タイミング故障自体を起こりにくくするアプローチも重要な課題となる。研究課題 b では、製造時、動作時の各種のばらつきに柔軟に対応できるクロッキング方式を提案、実装する。従来の FF によるクロッキング方式と比べ、タイミングに関する制約を大きく緩和させ、タイミング故障の発生確率を減少させるとともに、より

積極的なクロック周波数の向上を狙う。

上記の二つの技術を組み合わせることで、DVFS + 動的タイミング故障回復を実現し、その有効性を確認することを目指す。この手法によって、実際の遅延に基づく電圧、周波数設定が可能になると、ワースト・ケース設計によるマージンの見積もりから解放される。結果として、半導体の微細化による性能向上が、今後も見込めることとなる。

研究課題 c では、FPGA をベースにして安価に高信頼 LSI を実現することが目標である。ただし、耐故障性を実現するために追加するハードウェアをできるだけ小さくする必要がある。耐故障性のための追加ハードウェアを十分に小さく抑えることができれば、通常の FPGA からのコストアップを最小限に留めることができる。結果として、通常用途向けに大量に製造、販売しながら、そのまま安価な高信頼 LSI として使用可能な FPGA が実現できると考えられる。

③研究のアプローチ

研究課題 a の故障回復技術については、関連する既存手法に Razor、Razor II がある。この手法は、データパス上のタイミング故障にのみ対応するもので、対象とするプロセッサも In-Order の単純なものに限定されている。実用的なプロセッサへの応用を考えると、Out-of-Order の実行系を備える複雑なプロセッサに対しても適用できる故障回復手法が必要不可欠である。

本グループで提案する手法は、リセットをベースにすることでデータパスのみならず、制御パスに生じたタイミング故障も包括的に検出、回復可能とする。さらに、In-Order の実行系では生じなかった Out-of-Order 特有の問題に対しても解決策を提案する。これはロード・ストアキューに生じたタイミング故障を、従来の方法では正確に回復できなかった問題に対応するものである。これらのアプローチは、プロセッサへの故障回復技術の応用を考える上で大きな意義があるといえる。

研究課題 b のクロッキング方式は、遅延に対して許容できる範囲を大きく広げることでタイミング故障耐性を持たせる手法である。関連研究の Razor の手法では、信号の遅延がサイクル・タイムを超えた瞬間にエラーとして検出し、回復動作が実行されるのに対し、本手法では遅延の累積が一定以上を超えるまで検出、回復を行わない。言い換えると、あるステージで動的に遅延量が増加し、サイクル・タイムを超えていても、次のステージで取り戻すことができれば、何の支障もなく、動作を継続することができる。この性質により、動的な遅延の変動に対しても、そのばらつきを吸収することができる。

本手法ではこの考え方に加え、入力ばらつきに着目した。入力ばらつきとは、入力値によって出力値が確定するまでの遅延時間が、変動することを指した新しい概念である。入力ばらつきに関する考察の結果、出力値にクリティカルパスが関係する確率は非常に低いことが判明した。入力ばらつきを考慮して、より実際的な遅延にもとづき、動作点を定めることで、タイミングに関する制約を緩和した。この手法は、回路レベルのアプローチであるため、その適用範囲はプロセッサにとどまらず、同期回路一般に応用可能な技術である。

課題 c では、FPGA に耐故障性を付加する際に、追加ハードウェアをいかに最小限に抑え、通常用途と両立させるかが重要な点となる。FPGA に構成する回路を三重冗長化することで故障検出を行う。さらに、検出箇所を動的に再構成することで故障から回復を行う。この際、回路を再構成するために必要な計算を行う回復用ロジックが必要となる。このロジックを従来のようにハードワイアードの固定論理として追加するのではなく、ユーザロジックとしてソフトコアの形で実装する。この結果、ハードウェアを追加せずに高信頼機能を付加することが可能である。同時に回復用ロジックに生じた故障も自身で検出、回復可能であるため単一故障点が生じないという利点も生まれる。

④研究実施方法(研究チーム内外の連携関係など)

課題 a, b, c について、出口指向のプロトタイプ作成・評価を行い、結果をベンダーなどに提供して評価する流れとなっている。

課題 a, b については、アーキテクチャグループで提案したディペンダブルアーキテクチャを形式的検証グループに提供し、形式的な検証を行うことを予定している。また、プロセッサ VLSI のベンダーに対して知財として提供するための準備を進めている。

課題 c については、JAXA 宇宙科学研究本部の齋藤教授と連携し、宇宙用高信頼デバイスに関する情報交換を行っている。今後も協力体制を築きつつ、さらに関連ベンダーとも具体的な検討を行い、宇宙開発の現場への実用化をはかっていく予定である。

(2)「形式的検証」グループ (藤田グループ)

①本研究グループの研究課題、ならびに所属する研究チームの課題との関係

本グループでは、設計の正しさの検証技術、各種論理的・電氣的誤設計のデバッグ支援技術、ならびにディペンダブルアーキテクチャ設計の形式的な解析手法に基づく検証・デバッグ・最適化支援技術を研究開発し、他グループの研究成果に適用し融合することによって、マイクロプロセッサやネットワークプロセッサなどから構成されるディペンダブルアーキテクチャ全般に対するディペンダブルな設計技術の構築を目指す。また、形式的検証技術に関しては、そのツール化を推進し、利用が期待できる企業と協調し、実設計や企業で実際に利用されている設計フローの中での評価を通じた実用性向上に関しても研究開発を進めていく。

<課題1> 等価性検証ツールの研究開発

VLSIシステムのディペンダビリティを保証するための1つの手段として、設計段階での検証を十分に行うことは必要不可欠である。一般的に、形式的検証では、設計記述の規模に対して指数的に必要な計算量が増大するため、大規模な設計が扱えないのが現状であり、この問題を解決することが主要な技術的課題である。

<課題2> ポストシリコンデバッグ・検証

本課題では、製造後のVLSIチップ中の論理的なバグを中心に、バグ(エラー)の検出方法やデバッグ方法について研究を行う。本課題で主に対象としているのは、ソフトウェアのような頻度の極めて少ないエラーに起因する不具合ではなく、機能設計(論理設計)段階で検出可能なバグである。また、デバッグ結果に基づいて、チップの再製造を回避しつつ機能を修復する技術の開発も課題の1つである。

<課題3> ディペンダブルアーキテクチャの形式検証

マイクロプロセッサのディペンダビリティ向上の有効な手段として、タイミングエラーを検出し、エラーから自動的に回復するアーキテクチャが提案されている。設計の正しさの保証には形式的検証手法が有効であるが、計算量の増大のため現実的な規模を扱うことは難しい。よって、タイミングエラーのリカバリ機能などディペンダブル機能を取り込んだ最先端プロセッサの効率的な形式的検証手法の研究開発は、最も重要な技術的課題の1つである。また検証結果を踏まえた設計デバッグ技術も重要な課題である。

②本研究グループの達成目標。

個々の課題に対する目標は以下の通りである。

<課題1> 等価性検証ツールの研究開発

等価性検証ツールにおける目標は、既存の手法・商用ツールに比べて、検証可能な規模を1桁以上高めることである。なお、研究成果をツール化することで、企業などで実際の設計を用いた評価も進めていく。

<課題2> ポストシリコンデバッグ・検証

ポストシリコン(チップ製造後)の検証・デバッグにおける最も重要な点は、チップの実動作速度で生成される大量の信号データを処理できることである。信号をトランザクションレベルで抽象化することでデバッグの効率化を実現する。また、設計誤り修正などの機能修復をチップ再製造することなく実現する回路機構を開発する。

<課題3> ディペンダブルアーキテクチャの形式検証

ディペンダブルアーキテクチャを正しく検証するためには、エラー回復アーキテクチャ単体の検証だけで不十分であり、マイクロプロセッサアーキテクチャ全体を含めて検証する必要がある。本課題では、実用的なアーキテクチャに対しても数時間程度の時間で検証することが可能な手法の確立およびその検証環境(ツール化)の開発を目標とする。また設計デバッグに関しては、従来技術を発展させ、マイクロプロセッサへも適用可能とする。

③研究のアプローチ

形式的検証技術に関しては、新規技術の研究開発のみではなく、等価性検証ツールとして企業などへ提供可能な形にし、実設計も利用して評価改良を進めていく。ポストシリコン検証に関しては、主に SoC (System on Chip)を対象とし、C 言語ベースのトランザクションレベルの設計記述を有効利用することで、検証・デバッグ時における解析する必要がある設計や動作の「量」の大幅な減少を実現し、効率化を目指す。ディペンダブルアーキテクチャ検証では、抽象化による検証の効率化とともに、デバッグ過程についても支援していく。

<課題1> 等価性検証ツールの研究開発

これまで研究開発を行ってきた等価性検証技術の実用化を目指し、「検証技術実証」グループ（若林グループ）と協力して、実設計環境における技術評価を進める。具体的には、実用的な規模の例題に対して技術を適用することで、現実的に検証可能な規模の評価を行う。

<課題2> ポストシリコンデバッグ・検証

大量に生成されるチップ信号データから検証・デバッグに必要な情報を抽出する手法として、通信トランザクションの単位で実動作過程を解析する手法を提案した。これにより、非常に効率的に設計誤り箇所を特定することが可能となる。また、ポストシリコンにおける機能修復技術としてパッチ可能ハードウェアの回路機構を提案している。これは FPGA などの再構成可能素子とは異なり、専用ハードウェアの性能・効率を実現しつつ機能修復することを可能としている。

<課題3> ディペンダブルアーキテクチャの形式的検証

本課題においても、対象とするアーキテクチャを抽象化することで簡単なモデルを構築し、このモデルに対して形式的検証手法を適用することで大規模な設計に対しても現実的な時間内で検証をすることが可能となる。さらに、デバッグ過程も支援するため、従回路レベルで提案されていたマルチプレクサを内部信号線に挿入することによる設計誤り箇所特定手法を拡張して導入し、特定された設計誤りを自動的に修正する技術の提案も行った。

④研究実施方法(研究チーム内外の連携関係など)

新規技術の学会発表(招待講演を含む)や世界的に著名な大学や先端研究機関でのセミナー開催を通じた交流も進めている。具体的には、各課題に対して、以下の通り進めてきている。特に、等価性検証ツールは企業などへ提供し、共同で評価を行っている。

<課題1> 等価性検証ツールの研究開発

これまで研究開発を行ってきた等価性検証技術の実用化を目指し、同チーム内「検証技術実証」グループ（若林グループ）と協力して、実設計環境における技術評価を進めている。

<課題2> ポストシリコンデバッグ・検証

ポストシリコン機能修復技術の1つとして、回路中に部分的にプログラマブル素子を挿入する技術の研究開発をトロント大学の Veneris 教授と立命館大学の山下教授と協力して進めている。

<課題3> ディペンダブルアーキテクチャの形式的検証

当チーム内坂井・五島グループで開発しているアウト・オブ・オーダー実行スーパースカラプロセッサをモデルとして、エラー回復機構の検証を行い、その設計の正しさを証明した。手法自体は一般的なプロセッサに適用可能である。

(3)「ディペンダビリティ支援ルータ」グループ(吉瀬グループ)

①本研究グループの研究課題、ならびに所属する研究チームの課題との関係

プロセッサの性能向上と消費電力の削減を目的に、広い分野にわたり、複数のコアを搭載するマルチコアプロセッサが主流となりつつある。今後は、半導体集積度の向上により、さらに多くのコアを集積するメニーコアプロセッサへと向かうと考えられる。一方、LSI の一層の微細化により、ソフトウェア率およびばらつきが増加し、信頼性の低下が深刻化している。微細化が進むことで臨界電荷量が低減すると、中性子などの影響に

よりビットが反転する。マルチコアやメニーコアプロセッサにおいてはプロセスの微細化とともに集積するトランジスタ数が増加するため、ソフトウェア率の増加が懸念される。同時に、電源、温度の不均一などに起因するチップ内のランダムなばらつきが信頼性の低下を招く。

このような背景において、これから VLSI プロセッサの主流となるマルチコアプロセッサ、メニーコアプロセッサにおけるディペンダビリティを飛躍的に向上させるアーキテクチャとして、超ディペンダビリティ支援の高機能ルータを開発することが本研究グループの課題である。成果は、ディペンダブルアーキテクチャグループ、形式的検証グループと統合して全体として VLSI システムのディペンダビリティを飛躍的に高める技術とする。具体的には、形式的検証グループにメニーコアの設計データを提供し、協力して形式的検証を行うこととする。

②本研究グループの達成目標。(目標が達成されると①に述べた課題がどのように解決されるかを記載して下さい。)

(1) 低オーバーヘッドで柔軟にディペンダビリティのレベルを向上させられる高機能ルータアーキテクチャの研究開発および評価を行う。

ここで検討する方式は、メニーコアプロセッサ内のネットワーク(NoC)における送受信パケットのレベルで、高機能化されたオンチップルータを核として冗長実行(多重実行)を実現するアーキテクチャである。高機能ルータにパケットの複製などの冗長実行の仕組みを組み込むことで、コアの構成を変更することなく冗長実行が可能となる。また、割り当てるコアの数を柔軟に変更できるため、実行時に、要求に応じてディペンダビリティを向上できるという利点がある。この仕組みを活用して、メニーコアプロセッサにおけるディペンダビリティの飛躍的な向上を達成する。

(2) ディペンダビリティ向上のために(1)で提案したアーキテクチャを実装した場合のハードウェア量および実現可能性を FPGA システムの活用および中規模高機能ルータのチップ試作を通じて明らかにする。

(3) ディペンダビリティ向上のために(1)で提案したアーキテクチャ技術の有用性を、形式的検証によって証明する。

(4) ディペンダビリティを考慮したタスク割り当て方式を含むメニーコアプロセッサのためのスケジューリング技術を研究開発する。

③研究のアプローチ(目標達成のために本研究グループで提案している方法・アイデアの適用効果、適用対象の広さ、優位性、独創性)

低オーバーヘッドで柔軟にディペンダビリティのレベルを向上させられる高機能ルータアーキテクチャとして、SmartCore(Smart many-core system with redundant cores and multifunction routers)を提案している。SmartCore システムは、オペレーティングシステムなどのシステムソフトウェアと協調して動作し、高機能ルータの支援による複数コアの多重実行によって、メニーコアプロセッサにおける信頼性の向上、チップ内ネットワークのバンド幅およびレイテンシの改善を目指す方式である。

SmartCore システムでは、アプリケーション実行のために本来必要とされるノード(1つのコアや1つのルータを含むメニーコアプロセッサの1つの要素)をマスターノード、多重実行のために SmartCore システムが導入するノードをミラーノードと定義する。高機能ルータにおいて、(a) マスターノードが受信するパケットを複製しミラーノードに送信する、(b) ミラーノードから送信されるパケットの送信先をマスターノードに変更する、(c)ミラーノードから届くパケットを待ち合わせ、マスターノードにおいてパケットの比較およびエラー検出をおこなう、という3つの機能により多重実行を実現し、信頼性の向上を達成する。

高機能ルータの詳細検討はクロックレベルのソフトウェアシミュレータを用いる評価と、FPGA システムを用いる大規模な評価の2つの方法を適切に利用して進めていく。ソフトウェアシミュレータには、現実の制約にとらわれずに機能を実装できるという利点があるが、一方で、シミュレーションに要する時間が膨大になるという欠点がある。一方、FPGA システムを用いる場合には、構築のために時間を含むコストを要するが、高速に動作するという利点がある。このため、両システムを適切に使い分けながら、高機能ルータアーキテクチャを評価する。加えて、FPGA システムの活用および中規模高機能ルータのチップ試作を通じて、提案方式のハードウェア規模および実現可能性を明らかにする。

汎用プロセッサにおけるディペンダビリティの向上技術としては、冗長性を利用して故障を検出する技術が提案されている。DIVA(Dynamic Implementation Verification Architecture)や IRI(Instruction Re-Issue)などの初期の技術は命令を2回実行して得られた結果を比較することで故障を検出する。素子特性のばらつきや

最悪ケースに配慮した設計の限界から典型的なケースを指向して Razor やその改良型のカナリア・フロップフロップといった設計手法も提案されている。これらの技術は汎用プロセッサ内部に多重実行やタイミング調整の仕組みを構築する。

一方、我々は、従来の発想を大幅に転換させ、来るべきメニーコアプロセッサの環境を前提として、高性能ルータを積極的に利用することで動的な要求変更に対応する技術、すなわち超ディペンダビリティ支援の高機能ルータの実現を目指す。このような研究はこれまでに例がない。

④ 研究実施方法(研究チーム内外の連携関係など)

研究チーム外の連携として、ネットワークオンチップ(NoC)技術について米田チームとミーティングをおこなひ、研究を加速させるための体制を構築しつつある。特にトヨタなど、応用分野の方々の意見聴取を行った。

領域アドバイザーの方々のアドバイスを受けながら、NoC 技術の出口戦略に関して様々な観点で検討している。

また、坂井グループの成果をコア内に適用し、メニーコアレベルでは当該グループの成果を入れる成果の統合についても検討を進めている。

(4) 「検証技術実証」グループ (若林グループ)

① 本研究グループの研究課題、ならびに所属する研究チームの課題との関係

本研究グループは、設計の正しさの検証技術を実設計現場に導入することによって VLSI のディペンダビリティを向上させることを目標に、研究を進める。そのために、「形式検証」グループ(藤田グループ)と連携して、等価性検証ツールの実用化に対する評価検討を行う。

まず、等価性検証ツールの評価環境の構築を行う必要がある。形式的検証グループが研究開発した等価性検証ツールの実用化のためには、産業界で用いられる設計環境においてより多くの多面的なデータを用いてその性能の評価を行う必要がある。しかしながら、C言語設計は次世代の設計技術であり、実設計回路を評価データとして入手することは容易ではない。これに対して、CyberWorkBench は、動作レベルIPとして多くの設計資産を有している。そこで、CyberWorkBench の有する豊富な設計資産を等価性検証ツールの評価に使用するための環境を構築する必要がある。

次に、等価性検証ツールの実設計現場での運用可能性の評価を行う必要がある。一般に、形式的検証は設計記述の規模に対して指数的に必要な計算量が増大するため大規模な回路が扱えないと考えられている。そこで、等価性検証ツールで扱える回路の規模感を実回路設計における実感として正しく認識する必要がある。さらに、等価性検証ツールの実用化のためには、産業界の実設計現場でのどのようにツールを運用するのが良いかを定め、実運用した場合にはどのような課題が起こりうるかを明らかにし、ここで出てきた課題を等価性検証ツールのさらなる改良・機能拡張につなげなければならない。そのため、等価性検証ツールの限界を勘案しつつ実用に供するに堪えるツール運用のシナリオを定め、そのシナリオに沿ってツールを運用し、そこで出てくる課題を洗い出す必要がある。

② 本研究グループの達成目標。

まず、等価性検証ツールの評価環境の構築として、NEC が提供する CyberWorkBench と形式検証ツール FLEC とのインターフェイスのプロトタイプを作成する。特に、CyberWorkBench が有する設計資産である動作 IP を等価性検証ツールに入力できることを目標とする。

次に、等価性検証ツールの実設計現場での運用可能性の評価を行う。等価性検証ツールの性能限界を把握するとともに、ツールの特性に応じた適用シナリオを策定し、シナリオに沿ってツールを運用する。

以上により、CyberWorkBench が有する実回路データを用いて等価性検証ツール評価を行い、実運用上の課題を洗い出し、実用化につなげることができる。

③ 研究のアプローチ

CyberWorkBench は、NEC から販売されている高位統合設計環境であり、C 言語および SystemC 言語を入力とする動作合成ツールを中心として各種検証ツールも内包し、SoC および FPGA の設計及び検証を C

言語で行うことを可能としている。社内において広く実製品の設計に用いられた実績を有している。また 2007 年から本格的に外販を開始しており、現在では数十社の顧客を有している。このように、次世代の設計上流工程において標準になりうる設計環境である。さらに、CyberWorkBench は浮動小数点演算や三角関数などの数値演算回路などの C 言語や SystemC で記述された実設計回路の設計資産を動作 IP としてツールに併せて提供している。DES, AES, CRC などの誤り訂正・暗号回路などの設計資産も有している。

CyberWorkBench と等価性検証ツールのインターフェイスの作成により、これら豊富な実設計回路の設計資産を用いて等価性検証ツールの評価を行うことができる。

等価性検証ツールの実設計現場での運用可能性の評価では、評価のモチーフとして、等価性検証ツールの挙動を調査するための基本的なデータをはじめ、CyberWorkBench に含まれる動作 IP から浮動小数点演算や三角関数などの数値演算回路、設計資産から DES, AES, CRC などの誤り訂正・暗号回路、色領域変換やフィルター、JPEG などの画像処理回路、および、PCI や AMBA などのバス制御回路を予定しており、評価に適したデータを適宜選択するものとする。

④研究実施方法(研究チーム内外の連携関係など)

「形式的検証」グループと密接に連携する。まず、「形式的検証」グループより形式検証ツールの提供を受け CyberWorkBench とのインターフェイスを作成する。さらに、実設計データを用いた評価をもとに実用化に向けた課題を洗い出し、「形式的検証」グループへフィードバックを行う。

§ 2. 研究実施体制

(1) 東京大学 坂井グループ

① 研究分担グループ長: 坂井 修一 (東京大学大学院情報理工学系研究科、教授) (研究代表者)

② 研究項目

- ・ 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
- ・ 永久/過渡故障耐性を持つ**FPGA**アーキテクチャ用テストベッドの開発
- ・ タイミング故障耐性を持つクロッキング方式
- ・ タイミング故障耐性を持つプロセッサアーキテクチャ
- ・ 永久/過渡故障耐性をもつ **FPGA** アーキテクチャ

(2) 東京大学 藤田グループ

① 研究分担グループ長: 藤田 昌宏 (東京大学大規模集積システム設計教育研究センター、教授) (主たる共同研究者)

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ならびに回路最適化手法
- ・ タイミングエラーリカバリー機能を持つパイプライン演算回路
- ・ マイクロプロセッサアーキテクチャアルゴリズムの形式的検証手法
- ・ 大規模設計記述の等価性検証のためのトップダウン並びにボトムアップ検証手法
- ・ シミュレーション(エミュレーション)ベース検証における検証カバレッジ向上手法
- ・ **Post-silicon** デバッグ手法並びに、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

(3) 東京工業大学 吉瀬グループ

① 研究分担グループ長: 吉瀬 謙二 (東京工業大学大学院情報理工学研究科、准教授) (主たる共同研究者)

② 研究項目

- ・ 高機能ルータアーキテクチャの研究開発および評価
- ・ 形式的検証を用いた高機能ルータアーキテクチャ技術の有用性の検討
- ・ 中規模高機能ルータのチップ試作による実現可能性の検討

(4) NEC 若林グループ

① 研究分担グループ長: 若林 一敏 (日本電気株式会社組込みシステムソリューション事業部システム IP コア研究所、主管研究員) (主たる共同研究者)

② 研究項目

- ・ 形式的等価性検証技術の実用化に向けた実証実験とその評価

§ 3. 研究実施内容

(文中に番号がある場合は(4-1)に対応する)

(1) 研究の成果と自己評価

(1) 成果1. 「タイミング故障耐性を持つ Out-of-Order プロセッサの構成方式」(坂井グループ)

① 内容

前年度までに、タイミング故障検出機構と DVFS を組み合わせた 2-way スーパースカラプロセッサを FPGA 上に実装していたが、これを改良して、Out-of-Order プロセッサとし、さらに制御系を含むあらゆる部位の故障に耐えられるものとした。

具体的には、Out-of-Order プロセッサ内に存在する全てのフリップフロップ(制御系を含む)を故障検出機能の付いた改良型フリップフロップに置き換え、さらに故障検出信号を伝達するためのフォールト通知ネット

ワークを追加した。故障検出信号は、パイプラインステージに沿って伝達される。故障によって影響を受けた命令がレジスタファイルやプログラムカウンタなどのプロセッサ状態を更新する前にリセットをかけ、命令を再実行する。

本設計を、外部回路によって、電源電圧、動作周波数を任意に設定可能なテストベッドシステムのFPGA上に実装した。電圧、周波数が、動作可能範囲を超えた場合、タイミング故障を検出し、故障発生後も正しく再実行を行う様子が確認できた。

② 有用性

従来の手法では、ただ一度の故障も許されないため、十分なマージンを見込んだ設計をする必要があった。しかし、半導体の微細化に伴い、チップ内の信号の伝達遅延のばらつきが増大しているため、十分なマージンを確保すると性能が向上しないという問題があった。また、昨年度までのわれわれの手法は、もともと効率的な Out-of-Order プロセッサ上には実現できないものであった。

本手法では、Out-of-Order プロセッサに改良型のタイミング故障検出・回復機構を組み込んだことにより、仮に故障が発生しても、回復し実行を継続することができる。そのため DVFS を利用した積極的な電圧・周波数設定が可能になった。その結果、Out-of-Order プロセッサに従来のマージンを見込んだ悲観的な動作点の設定を行う必要がなくなる。

③ 優位比較

本手法と同様にプロセッサに対しアーキテクチャレベルでの回復を行う手法に、SPARC 64V、Razor 及び Razor II が存在する。

SPARC 64V は高信頼設計のなされた商用のサーバ向けプロセッサである。これにはデータパス上の全てのステージにパリティが付加されており、パリティによってデータパス上の故障を検出する。しかし、制御パス上にはパリティが付加されておらず、制御パス上のタイミング故障は検出することができない。そもそも、制御パス上に故障が発生した場合にプロセッサは設計者の意図しない状態に遷移する可能性があり、このような状態から正常な状態へと回復することは、データパス上の故障の回復に比べて格段に困難である。

Razor 及び Razor II では、タイミング故障を検出するフリップフロップ、及びそれを用いた故障耐性を持つプロセッサを提案している。しかし、ここで提案されているプロセッサも上記と同様に理由で制御パス上に発生したタイミング故障から回復することができない。

一方、本手法では故障からの回復にリセットを用いることで、データパスのみならず、制御パス上のタイミング故障も包括的に検出・回復可能である。さらに、Out-of-Order プロセッサにこのような仕組みを入れた成功例はこれまでになかった。

(2) 成果 2. 「タイミング故障耐性を持つクロッキング方式」(坂井グループ)

①内容

Razor と二相ラッチとを組み合わせ、従来よりもタイミング故障耐性が高い、柔軟なクロッキング方式を提案・初期評価した。

具体的には、実効的な遅延をステージ間で貸し借りすることにより、製造ばらつきに対処したうえで、最大 2 倍のクロック周波数を達成することに成功した。

②有用性

デザインばらつきの拡大に伴い、静的なワーストケース設計を行うことがたいへん困難になっている。ノーマルケース設計に近い設計をしながら、動作時にタイミング故障を抑止する機構が必須であり、本提案は、これを実現するものである。

③優位比較

すでに提案されている方式の中で、Razor は、実効遅延を貸し借りするという概念はない。チップ内のいずれかのクリティカルパスが活性化される可能性が 100%に近いいため、DVFS のマージンを削る効果しか持たないことになる。また、別方式である TIMBER は、ステージ間で遅延を貸し借りするという概念を持つものの、二相ラッチではなく単層 FF をベースにしているため、回路構成が非常に複雑になる欠点がある。

これらに比べて提案方式は、より大きなデザインばらつきに対応できると同時に、回路構成が単純で設計・製作しやすい利点を持つ。

(3) 成果 3. 「ポストシリコン機能修復回路機構およびその設計支援環境」(藤田グループ)

①内容

設計の複雑化に従い、チップ製造前にすべての設計誤りを取り除くことは年々難しくなっており、このためポストシリコンデバッグ技術が非常に重要となっている。しかしながら、たとえデバッグを用いて設計誤りを発見・修正することができたとしても、ほとんどの場合チップの再設計・再製造が必要となってしまう、設計期間・コストの増大は回避できない。この問題を解決するため、チップ製造後であっても小規模な機能修正が可能な回路機構であるパッチ可能ハードウェアを提案した。また、修正前後の設計記述から自動的にパッチを生成する設計ツールの開発を行った。

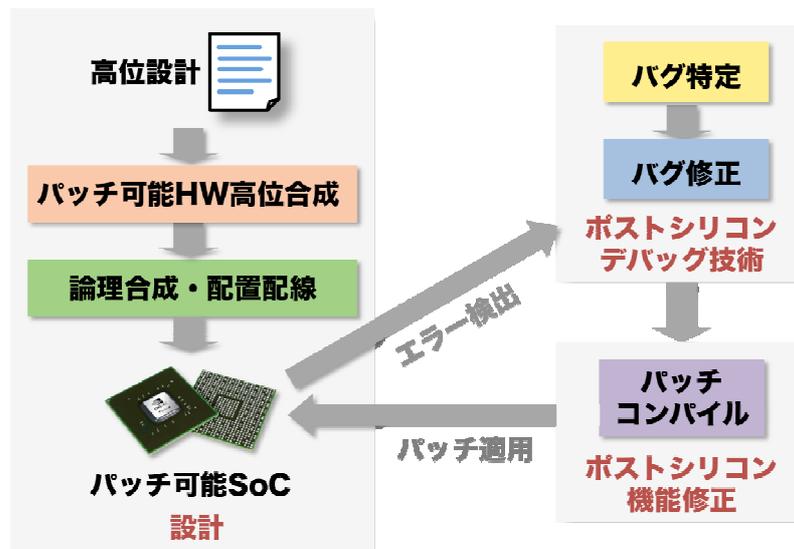


図1 パッチ可能ハードウェアを用いたポストシリコン機能修復のフロー

②有用性

製造後機能修正は FPGA などのプログラマブル素子や汎用プロセッサを用いることによっても実現可能ではあるが、専用ハードウェアに比べて性能や効率は大幅に低下してしまう。具体的には、専用ハードウェアに比べて FPGA は性能比で数倍、電力効率比で数十倍劣る。提案機構は専用ハードウェアに比べて数%程度の性能・効率低下で機能修正が可能であることを実設計を用いて示した。提案機構では機能全体の変更は不可能であるが、全体の 10~20%程度の修正が可能であり、設計誤り修正に対しては十分であると考えられる。

③優位比較

設計誤りといった比較的小規模の機能修正に特化したプログラマブルハードウェアは過去に例がなく、先駆的な研究といえる。代替方式としては機能全体を変更可能な FPGA や汎用プロセッサといった方式が考えられるが、上で述べた通り性能・効率の面で提案方式が圧倒的に優位である。この研究についてはすでに国内外で論文賞を受賞しており、その正当性はすでに評価されていると考えられる。

(4) 成果 4. 「ディペンダブルアーキテクチャのデバッグ・自動修正技術」(藤田グループ)

①内容

ディペンダブルアーキテクチャの検証の結果、何らかの設計誤りがあることが判明した場合には、設計誤り箇所を特定し修正するデバッグ作業が必要となる。しかしながら、ディペンダブルアーキテクチャは非常に複雑であり、そのデバッグは容易ではない。本研究では、回路レベルで提案されていたマルチプレクサを内部信号線に挿入して設計誤り箇所を特定する従来手法を拡張し、これを抽象化されたアーキテクチャモデルに適用することで設計誤り箇所を特定する手法を提案した。また、特定された設計誤りを自動的に修正する手法についても提案を行った。

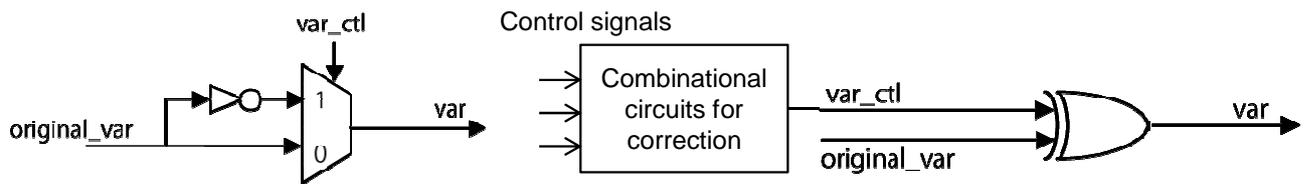


図2 アーキテクチャモデルへのマルチプレクサ挿入(左)と修正後の回路(右)

②有用性

タイミングエラー回復機構を備えたスーパスカラ・アウトオブオーダ・プロセッサに対し、意図的にバグを混入させた上で提案手法を適用したところ、数分程度(対象プロセッサの複雑度による)で設計箇所を特定し正しく修正可能であることを確認した。タイミングエラーだけでなく、例えばソフトウェアからの自動回復機能のあるプロセッサのデバッグも同様に行うことができる。

③優位比較

従来のプロセッサのバグ箇所特定手法は回路レベル(レジスタ転送レベル・ゲートレベル)で行うものが主であり、そのため複雑なものを扱うことが困難である。提案手法はより抽象度の高いマイクロアーキテクチャレベルでバグ箇所特定を行うことで複雑なプロセッサに対しても効率的にデバッグを行うことが可能である。また設計誤り修正に関しても、既存手法である回路レベルの手法をマイクロアーキテクチャレベルに適用したのは提案手法が初めての試みである。

(5) 成果 5. 「高機能ルータアーキテクチャ」(吉瀬グループ)

①内容

ディペンダビリティを支援する高機能ルータアーキテクチャを有するメニーコアプロセッサの枠組みとして SmartCore システムを提案するとともに、高機能ルータの基本アーキテクチャを開発(IPSJ SIG-ARC 2008 年 10 月)した。また、ディペンダビリティのレベルを変更できる高機能ルータアーキテクチャを開発(IPSJ SIG-ARC 2010 年 1 月)を開発した。ここでは、レジスタ転送レベル(RTL)で、高機能ルータの内部構成を定義するとともに、その構成で正しく動作することを確認した。

②有用性

64 コア構成のメニーコアプロセッサを想定し、NAS Parallel Benchmark を用いたソフトウェアシミュレーションの評価から、高機能ルータによるミラーノードから届くパケットの待ち合わせによるアプリケーションの性能低下を評価した。その結果、SmartCore システムによる多重実行の性能低下は最大で 4.1%と、実用的なレベルであることを明らかにした(IPSJ SIG-ARC 2010 年 1 月, ICNC' 10 2010 年 11 月)。

開発中の FPGA システム(ScalableCore システム)に SmartCore システムの一部を実装し、正しく動作することを確認した(SACSYS2010 ポスター 2010 年 5 月)。この成果をシンポジウムのポスターとして発表し、優秀ポスター賞を受賞している。

③ 優位比較

汎用プロセッサにおけるディペンダビリティの向上技術としては、冗長性を利用して故障を検出する技術が提案されている。DIVA(Dynamic Implementation Verification Architecture)や IRI(Instruction Re-Issue)などの初期の技術は命令を 2 回実行して得られた結果を比較することで故障を検出する。素子特性のばらつきや最悪ケースに配慮した設計の限界から典型的ケースを指向して Razor やその改良型のカナリア・フロップフロップといった設計手法も提案されている。これらの技術は汎用プロセッサ内部に多重実行やタイミング調整の仕組みを構築する。

一方、我々は、従来の発想を大幅に転換させ、来るべきメニーコアプロセッサの環境を前提として、高機能ルータを積極的に利用することで動的な要求変更に対応する技術、すなわち超ディペンダビリティ支援の高機能ルータの実現を目指す。このような研究はこれまでに例がない。

(6) 成果 6. 「メニーコアプロセッサの評価環境の開発」(吉瀬グループ)

①内容

高機能ルータアーキテクチャを評価するためのベースとして、マルチコアおよびメニーコアプロセッサを対象としたソフトウェアおよびプロセッサアーキテクチャ研究のための基盤環境を構築した(電子情報通信学会論文誌 2010年10月)。この基盤環境には、プロセッサアーキテクチャ M-Core、クロックレベルのシミュレータ SimMc、ライブラリなどが含まれる。

高機能ルータの詳細評価に利用する FPGA システムとして ScalableCore システムを開発した。多数の FPGA を搭載する従来の評価用システムでは、評価対象のアーキテクチャに応じてスケラブルに構成を変更することが困難であった。この欠点を解決し、高機能ルータの評価を容易にするために、必要に応じてメッシュ接続された FPGA ボードの枚数を増加することができるスケラブルな構成を可能とする ScalableCore システムを提案し、ScalableCore System 1.0 を実装した(SACISIS2010 2010年5月)。この改良版として、通信速度の向上および設置面積削減し、100 個を超える FPGA を接続する ScalableCore System 3.3 を実装した(ARC 2012年3月)。

提案システムのチップ試作の準備として、32ビットの RISC プロセッサを搭載するチップ、4 個のルータを搭載するチップを試作(CMOS 0.18um)した(IPSJ SIG-ARC 2009年4月)。この成果に関連するポスター発表はシンポジウムにおける Most Interesting Poster 賞を獲得している。

②有用性

FPGA ボードを 100 枚用いた ScalableCore システムを構築し、ソフトウェアシミュレータと比較して約 129 倍の高速化を達成した。

③優位比較

ScalableCore システムと同様に FPGA を用いたシミュレーション環境には、RAMP をはじめとして、FAST、ATLAS、ProtoFlex などがある。RAMP は、ソフトウェアとハードウェアの両面からマルチコア/メニーコアの研究を加速するためのプラットフォームの研究を推進している。RAMP Gold では、単一の FPGA 上に、機能ユニットをエミュレートする Functional Model と、そのタイミングのみを調停する Timing Model という 2 つの部分に分けてターゲットプロセッサを実装している。そして 1 つの機能ユニットを時間軸方向で多重に利用する host-multithreading という手法を用いて、高スループットを達成している。

FAST は、4 個の Processor Tile を持つシミュレーションプラットフォームであり、チップマルチプロセッサにおけるメモリシステムの研究に適している。

ATLASでは、トランザクショナルメモリをサポートした CMP についての実用的なシミュレーション環境を提案している。また、ProtoFlex では、OS の動作をも模倣するフルシステムシミュレーションにおいて 38 倍の高速化を実現している。

これらシステムの多くは大規模 FPGA を用いている。一方、ScalableCore システムは、メッシュ接続のメニーコアプロセッサにターゲットを絞り、小規模な FPGA を用いて拡張可能(スケラブル)な形でシステムを構成する点に特徴がある。

同様に多数の小規模な FPGA を用いる試みとして、Cube(Imperial College London)がある。これは、64 個の小規模な FPGA をボード上にタイル状に配置する構成のシステムであるが、十分に安定動作していないという報告がある。一方、我々が構築している ScalableCore システムは、非常に安定動作しており実用的なシステムである。

(7) 成果 7. 「等価性検証ツールの運用可能性の評価」(若林グループ)

①内容

まず第1に、CyberWorkBench の実設計回路を等価性検証ツール FLEC の入力形式に変換するツールのプロトタイプを作成した。CyberWorkBench の設計記述である C 言語記述を、FLEC に入力可能な SpecC 記述に変換する。第2に、基礎的なデータを用いて等価性検証ツール FLEC を実設計現場で活用するための課題を抽出した。特に、規模と分岐数という、実用化における 2 つの大きな課題を認識した。第3に、等価性検証ツール FLEC を実設計現場で活用するシナリオを策定した。検証済みの動作 IP から異なる製品向けの RTL を自動合成した際の合成前後の等価性を検証するシナリオと、検証済みの動作 IP に(機能を保持したまま)性能チューニングのための設計変更を加えた際の設計変更前後の等価性を検証するシナリオの、2 つ

のシナリオを候補として検討した結果、後者のシナリオのうち、特に実用的な要望の強いC言語記述への設計変更前後の等価性検証に絞ったシナリオを策定した。第4に、CyberWorkBenchの動作IPから浮動小数点加減算器を選択し、前記シナリオに沿って実際の設計現場で利用した設計変更例を作成した。最後に、規模と分岐数という実用化における2つの課題を解決するために、比較対象を設計変更部分に限定する手法の基本アイデアを策定し基礎実験を行った。単純に設計変更部分を抜き出して比較するだけだと、規模が大きすぎたり検証に失敗したりするが、提案手法ではこれらの問題を解決している。

②有用性

CyberWorkBenchと等価性検証ツールFLECとのインターフェイスは、単純に入力形式を変換するだけでなく、FLECの制限となっているポインタや関数も併せて解消するため、浮動小数点加減算器などの動作IPの設計記述をFLECに入力することが可能である。さらに、C言語中の任意の部分を階層化して抽出する機構を備えるため、設計変更部分のみを抽出し規模の問題を回避することができる。しかも、ただ単に指定された部分を抽出するだけでなく、データフローを解析して必要な入出力やインターフェイスの定義を追加するなど、抽出後の記述の整合性の保持を達成しており、検証だけでなく大規模回路の合成にも有効に応用可能な技術となっている。

一般に形式的手法は大規模回路の検証が難しいことが知られているが、分岐数の課題は認識されていたもののそれほど問題になることは広く知られていなかった。これは、分岐の問題がC言語記述を対象とする場合特有であったためであろう。このように、今まで知られていなかった課題を明らかにしたことは、今後の実用化に向けた前進である。

また、浮動小数点加減算器のシナリオ例は、ASIC向けに設計された回路をFPGA向けに設計変更するという状況を想定したものであり、設計変更せずそのまま利用するよりも動作周波数が30%向上させることができた。動作IPは設計再利用性が高いとはいえ、実際の設計現場では製品品質を維持するために同様のチューニングを頻繁に行うことが予想され、現実性の高いシナリオ例といえる。

さらに、比較対象を設計変更部分に限定する手法は、前記シナリオで施される設計変更の局所性が高い性質を利用し、比較範囲を限定して規模と分岐数という課題を解決するものであり、一般的に大規模な回路への適用が難しいといわれる形式的検証手法を実用化するうえで有効と考えている。

③優位比較

従来、設計変更時の等価性の保証にはシミュレーションと呼ばれる手法が広く用いられている。しかしながら、シミュレーションを用いて網羅的な検証を行うには多大な時間を要する。一般的に高速であるといわれる動作レベルシミュレーションを用いても 10^{11} パターン(=1,000億パターン)を検査するのに数か月程度の期間がかかる。しかも、このパターン数は全入力パターンのごく一部であり、網羅度としては $5.4 \times 10^{-7}\%$ に過ぎず、コーナーケースバグを見落とす可能性は否定できない。インターデザイン社のFastVeriなど動作レベルシミュレーションを高速化する手法もいくつか提案されてはいるものの、今回のシナリオで想定するような設計変更には適用できず根本的な解決には至っていない。しかも、性能チューニングのための設計変更は、設計記述変更から動作合成、論理合成、配置配線を何度も繰り返すことにより行われることが多く、設計変更のたびに多大な時間をかけてシミュレーションを行うことは非現実的である。

これに対して、形式的手法を用いた等価性検証は、入力パターンなし、かつ、比較的高速に機能等価性を保証することができ、特にディペンダビリティが要求される回路の正当性保証には適当である。

しかしながら、先に述べたように、一般的に形式的手法は大規模回路への適用が難しい。比較対象を設計変更部分に限定する手法は設計変更の局所性を利用したものであり、すべての設計変更に適用できるものではない。

(2)上記(1)のうち、特筆すべき成果

(1)特に顕著な成果(科学や技術の新しい分野の展望など)

- ・複雑なOut-of-Orderタイミング故障の検知・修復について、単純で低コストな機構で、大きな効果が得られる方式を提案・検証できた。
- ・大規模実用回路に適用できる形式的検証ツールの開発に成功し、産学官の多くのユーザを獲得した。
- ・高機能ルータに関して、想定以上に早く大規模テストベッドの開発に成功し、実用化への道筋を得ることができた。大規模システムにおける自動的な冗長実行などの機能も動作確認された。

(2) 当初計画で想定外であった重要・新規な展開

- ・耐故障 VLSI について、宇宙開発分野に加えて、組み込みシステムなどでの応用を具体的に検討できるようになった。これは、本研究が進んだことが第一の理由だが、組み込み系などで複雑なプロセッサが使われるようになり、研究の進捗と世の中の進捗が同期するようになったことが第二の理由である。
- ・形式的検証に関して、NECが新しいグループとして参加し、実用化への具体的な道筋をつけることができた。
- ・ディペンダブルアーキテクチャの検証について、坂井 G と藤田 G の協調による成果が得られた。

§ 4. 成果発表等

(4-1)原著論文発表

●論文詳細情報

1. V. Izosimov, M. Lora, G. Pravadelli, F. Fummi, Peng Zebo, G. Di Guglielmo, and M. Fujita, “Optimization of Assertion Placement in Time-Constrained Embedded Systems,” 6th IEEE European Test Symposium, pp.171-176, May 2011, doi:10.1109/ETS.2011.35.
2. Y. Lee, T. Matsumoto, and M. Fujita, “An Automatic Method of Mapping I/O Sequences of Chip Execution onto High-level Design for Post-Silicon Debugging,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E94-A, No.7, pp.1519-1529, Jul. 2011, doi: 10.1587/transfun.E94.A.1519.
3. G. Di Guglielmo, M. Fujita, F. Fummi, G. Pravadelli, and S. Soffia, “EFSM-based model-driven approach to concolic testing of system-level design,” 9th IEEE/ACM International Conference on Formal Methods and Models for Codesign, pp.201-209, Cambridge, UK, Jul. 2011, doi:10.1109/MEMCOD.2011.5970527.
4. R. Krishnamoorthy, S. Das, K. Varadarajan, M. Alle, M. Fujita, S K Nandy, and R. Narayan, “Data Flow Graph Partitioning Algorithms and Their Evaluations for Optimal Spatio-temporal Computation on a Coarse Grain Reconfigurable Architecture,” IPSJ Transactions on System LSI, Vol.4, pp.193-209, Aug. 2011, doi:10.2197/ipsjtsldm.4.193.
5. Shinya Takamaeda (Tokyo Tech), Ryosuke Sasakawa (Tokyo Tech), Yoshito Sakaguchi (Tokyo Tech), and Kenji Kise (Tokyo Tech): An FPGA-based Scalable Simulation Accelerator for Tile Architectures, COMPUTER ARCHITECTURE NEWS, Vol. 39, No. 4, pp. 38-43 (International Workshop on Highly-Efficient Accelerators and Reconfigurable Technologies HEART2011), Sep. 2011, DOI: 10.1145/2082156.2082166.
6. H. Yoshida and M. Fujita, “An Energy-Efficient Patchable Accelerator For Post-Silicon Engineering Changes,” IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Taipei, Taiwan, Oct. 2011, doi:10.1145/2039370.2039376.
7. 佐野伸太郎(東工大), 吉瀬謙二(東工大): メニーコアプロセッサのための通信衝突に着目したタスク配置手法, 情報処理学会論文誌コンピューティングシステム, Vol.4, No.4, pp. 96-109, Oct. 2011.
8. B. Alizadeh and M. Fujita, “Modular equivalence verification of polynomial datapaths with multiple word-length operands,” 16th IEEE International High Level Design Validation and Test Workshop, pp.9-16, Napa Valley, CA, USA, Nov. 2011, doi:10.1109/HLDVT.2011.6114160.
9. A.M. Gharehbaghi and M. Fujita, “Formal Verification Guided Automatic Design Error Diagnosis and Correction of Complex Processors”, High-Level Design Validation and Test Workshop, pp. 121-127, Napa Valley, CA, USA, Nov. 2011, doi:10.1109/HLDVT.2011.6113987.
10. H. Mangassarian, H. Yoshida, A. Veneris, S. Yamashita, and Masahiro Fujita, “On Error Tolerance and Engineering Change with Partially Programmable Circuits,” 17th Asia and South Pacific Design Automation Conference, pp. 695 - 700, Sydney, Australia, Jan. 2012. doi:10.1109/ASPDAC.2012.6165045.
11. 佐藤真平(東工大), 吉瀬謙二(東工大): Distributed Shared-Buffer NoC ルータのためのパイプラインバイパス手法 情報処理学会論文誌コンピューティングシステム, Vol.5, No.1, pp.88-102, 2012
12. A.M. Gharehbaghi, M. Fujita, “Transaction-Based Post-Silicon Debug of Many-Core System-on-Chips”, International Symposium on Quality Electronic Design, pp. 703-709, Santa Clara, CA, USA, Mar. 2012.

(doi 情報は未発表のため、発表され次第追加予定)

13. Shinya Takamaeda (Tokyo Tech), Shintaro Sano (Tokyo Tech), Yoshito Sakaguchi (Tokyo Tech), Naoki Fujieda (Tokyo Tech), and Kenji Kise (Tokyo Tech): ScalableCore System: A Scalable Many-core Simulator by Employing Over 100 FPGA, The 8th International Symposium on Applied Reconfigurable Computing (ARC2011), Lecture Notes in Computer Science, Vol.7199/2012, pp.138-150, March 2012, DOI: 10.1007/978-3-642-28365-9_12.

(4-2)知財出願

- ①平成23年度特許出願件数(国内 0 件)
- ②CREST 研究期間累積件数(国内 3 件)