

機関番号：14301

研究種目：基盤研究 (B)

研究期間：2008～2010

課題番号：20300016

研究課題名 (和文) テスト容易な演算回路の自動合成に関する研究

研究課題名 (英文) Research on synthesis of easily-testable arithmetic circuits

研究代表者

高木 直史 (TAKAGI NAOFUMI)

京都大学・大学院情報学研究科・教授

研究者番号：10171422

研究成果の概要 (和文)：

テストパターン数が演算数のビット数によらない桁上げ選択加算器、および、テストパターン数が回路の段数に比例する並列プレフィクス加算器の構成法を開発した。また、乗算器に関して、テストパターン数が演算数のビット数によらない4-2加算木および種々の加算木の構成法を開発するとともに、桁上げ保存加算器で構成される部分積加算部がすべて回路の段数に比例するパターン数でテストできることを示した。また、テスト容易な並列プレフィクス加算器および4-2加算木を自動合成するツールのプロトタイプを作成した。

研究成果の概要 (英文)：

We have developed a carry select adder which can be tested by a test set whose cardinality is independent of the operand size, and a parallel prefix adder which can be tested by a test set whose cardinality is proportional to the depth of the circuit. For multiplier design, we have developed a 4-2 adder tree, as well as other adder trees, which can be tested by a test set whose cardinality is independent of the operand size, and also shown that any partial product compressor consisting of carry save adders can be tested by a test set whose cardinality is proportional to the depth of the circuit. We have also developed a prototype tool for synthesizing easily testable parallel prefix adders and 4-2 adder trees.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2008年度	3,200,000	960,000	4,160,000
2009年度	4,100,000	1,230,000	5,330,000
2010年度	1,600,000	480,000	2,080,000
年度			
年度			
総計	8,900,000	2,670,000	11,570,000

研究分野：情報科学

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：VLSI のテスト、算術演算回路、乗算器、加算器、テスト容易化設計

1. 研究開始当初の背景

VLSI（大規模集積回路）は、コンピュータはもちろん、AV家電をはじめとするさまざまな家電製品、携帯電話端末等の情報通信機器、自動車のさまざまな制御装置等々に組み込まれ、情報化社会の基盤となっている。VLSIの誤動作が社会に甚大な損失をもたらす恐れがあるため、VLSIの信頼性の向上が大きな課題となっている。近年、情報通信分野の高信頼・高安全性を求める研究の重要性が世界的に認識されており、ヨーロッパでは、「SecureIST(Information Society Technology)」プロジェクトが推進されており、我が国でも、平成19年度に、JST CRESTに「ディペンダブルVLSIの基盤技術」研究領域が新設された。

VLSIの信頼性の向上には、設計の正しさの保証と製造されたVLSIのテスト（故障検査）が不可欠である。VLSIの大規模化、複雑化に伴い、設計の正しさを保証するための設計検証、および、テストのための入力パターンを求めるテスト設計に多大な時間を要するようになってきており、設計期間の50～70%を占めるとも言われている。また、テストに要する時間およびコストも増大している。

VLSIの信頼性の向上には、VLSIの重要な構成要素である、加算器や乗算器等の演算回路について、テスト容易な構成法を開発し、設計の正しさが保証されたテスト容易な演算回路を自動合成するシステム（IPライブラリ）の開発が重要である。

演算回路については、古くから研究がなされ、さまざまな構成法が知られている。既に、IPライブラリが市販されており、演算数（オペランド）のビット長等を指定すれば、設計の正しさが保証された演算回路が自動合成されるようになってきている。しかし、これらのIPライブラリの演算回路では、テストについては考慮されていない。一方、演算回路のテストについても研究がなされているが、これらの多くは、個々の構成法に対してテストパターンを求めるものである。また、テストを容易にするために、演算回路を改造することもあり、その場合には設計の正しさを保証するための検証が必要となる。

このように、テスト容易な演算回路の自動合成システムの開発が望まれている。

2. 研究の目的

本研究では、VLSIの重要な構成要素である、加算器や乗算器等の演算回路について、テスト容易な構成法を開発し、設計の正しさが保証されたテスト容易な回路を自動合成するシステムの実現に必要な基盤技術を確認することを目的とする。

3. 研究の方法

まず、テスト容易な加算器および乗算器の構成法を開発し、これらの演算回路が正しく動作することを形式的手法及びシミュレーションにより確認する。さらに、これらに基づくテスト容易な演算回路を自動合成するツールのプロトタイプを作成する。

テスト容易な加算器および乗算器の構成法の開発に当たっては、故障モデルとして単一セル機能故障を用いる。このモデルでは、2進数一桁の加算を行う全加算器等の基本的な回路ブロックを「セル」として扱い、回路中の一つのセルが故障し、少なくとも一つの入力パターンに対して、誤った出力を生じるものと仮定する。セルの具体的なゲートレベルでの設計に依らない普遍的なモデルであり、VLSIのテスト設計で広く用いられる単一縮退故障を包含する、より、強力なモデルである。たとえば、一つのセル内の多重縮退故障も対象となる。また、演算回路が正しく動作することの検証も、論理値のレベルより上位の数のレベルで行う。

4. 研究成果

(1) テスト容易な桁上げ選択加算器

演算数のビット数に依らず、16個のパターンでテスト可能（すべての単一セル機能故障を検出可能）な桁上げ選択加算器の構成法を開発した。

桁上げ選択加算器は、比較的高速でハードウェア量が少ない加算器であり、マルチブロック構成にし、各ブロックのサイズを演算数のビット長の平方根に比例するようにすることにより、演算数のビット長の平方根に比例する計算時間で加算を行える。桁上げ選択加算器については、これまで、テスト容易な構成法は知られていなかった。

提案する構成法では、テスト用に外部入力線を1本追加するとともに、各ブロックにインバータ（NOTゲート）を一つ付加する。16パターンからなるテスト集合は、単一セル機能故障モデルの下で、最小である。

(2) テスト容易な並列プレフィクス加算器

回路の段数に比例する個数のパターンでテスト可能（すべての単一セル機能故障を検出可能）な並列プレフィクス加算器の構成法を開発した。

並列プレフィクス加算器は、桁上げ先見加算器を一般化したもので、桁上げ生成条件と桁上げ伝搬条件を木状に計算することにより、加算を高速に行う。さまざまな構造の木を構成可能で、回路面積や遅延時間などの要求に応じた加算器を構成できる。二分木状の木を構成すれば、回路の段数（計算時間）は演算数のビット長の対数に比例する。並列プ

レフィクス加算器については、これまで、Brent-Kung 加算器についてのみ、回路の段数に比例する個数のパターン数でテスト可能な構成が知られていた。

提案する構成法では、テスト用に外部入力線を1本追加するとともに、加算器を構成する4つの基本セルの機能を変更する。セルの接続関係は変更しない。したがって、従来の構成法と同様に、さまざまな構造の木を構成できる。Sklansky 加算器と Kogge-Stone 加算器では、木の高さ L に対し、 $30L+22$ 個のパターンでテスト可能である。

本研究の構成法により、代表的な並列プレフィクス加算器である Sklansky 加算器、Kogge-Stone 加算器、Brent-Kung 加算器、Han-Carlson 加算器を、演算数が16ビット、32ビット、64ビットの場合について設計した場合は、回路面積は約13%、遅延時間は約4%増加した。

本研究では、さらに、この構成法を改造し、木の葉から k 段分を直列構造に置き換え、 $6(2^k-1)+30(L-k)+46$ パターンでテスト可能な構成法を開発した。これにより、計算時間の制約が緩い場合は、一部を直列構造に置き換えることにより、より少ないテストパターンでテスト可能な並列プレフィクス加算器を構成できる。

(3) テスト容易な4-2加算木

並列乗算器は、一般に、部分積生成部、部分積加算部、最終加算器の三つの部分で構成される。部分積生成部は、乗数の一桁と被乗数から、部分積を生成する。部分積加算部は、部分積を桁上げ保存加算で加え合わせ、桁上げ保存形(二つの2進数)で表させた積を得る。最終加算器は、桁上げ保存形で表させた積を桁上げ伝搬を伴う加算により、2進数で表させた積に変換する。部分積加算部の構成により、配列型乗算器、Wallace 木乗算器、4-2加算木乗算器等が知られている。

配列型乗算器は、部分積を桁上げ保存加算で順次加え合わせていく。VLSI 実現に適した規則正しい回路構造であるが、計算時間(回路段数)は演算数のビット数に比例する。Wallace 木乗算器は、部分積を桁上げ保存加算で木状に加え合わせていく。計算時間は演算数のビット数の対数に比例し高速であるが、各桁上げ保存加算では3数を加え合わせて和を保存して2数を得るので、単純な木構造にはならず、回路構造が非常に複雑になる。4-2加算木は、桁上げ保存加算を2段直列に接続した4-2加算により、部分積を二分木状に加え合わせている。計算時間は演算数のビット数の対数に比例し高速である。Wallace 木乗算器よりは若干低速であるが、回路構造の規則性で優っている。

これまで、配列型乗算器については、演算

数のビット数に依らない個数のパターンでテストできる設計が知られていた。また、4-2加算木乗算器については、回路段数(演算数のビット長の対数)に比例する個数のパターンでテストできる設計が知られていた。

本研究では、演算数のビット長に依らず14個のパターンでテスト可能(すべての単一セル機能故障を検出可能)な4-2加算木乗算器の部分積生成部と部分積加算部の構成法、および、テストパターンを求める再帰的手法を開発した。最終加算器には、テスト容易な加算器を用い、部分積生成部および部分積加算部とは別個にテストすればよい。4-2加算木に若干の回路を付加し、加算木のテストに交互反転パターンとよぶ特別なパターンを用いる。部分積生成部と部分積加算部のハードウェアオーバーヘッドは64ビット乗算器で約15%である。

(4) テスト容易な種々の加算木

これまでに提案されているテスト容易な乗算器の構成手法は、対象とする乗算器の構成に強く依存するため、同じ手法を他のタイプの乗算器に適用することはできない。このため、要求性能が変更され、部分積加算部の構成を他のタイプに変更する場合、同じ手法では、テスト容易化できなかった。

本研究では、(3)で提案した加算木の構成法およびテスト設計法を拡張し、テスト容易な種々の部分積加算部および部分積生成部の構成法、および、そのテスト設計法を開発した。提案手法では、部分積加算部を3種類のブロックを組み合わせることで構成する。ブロックの組み合わせ方により、さまざまな構造の部分積加算部を構成できる。(3)で示したテスト容易な4-2加算木はもちろん、テスト容易な配列型乗算器も構成できる。部分積生成部と部分積加算部は、演算数のビット長に依らず14個のパターンでテスト可能(すべての単一セル機能故障を検出可能)である。

提案手法を用いることで、計算時間や面積等の対する性能要求に合わせたテスト容易な乗算器を設計できる。

(5) 桁上げ保存加算器で構成された乗算器の部分積加算部のテスト

(4)で提案した手法により、演算数のビット数に依らず14個のパターンでテスト可能な種々の加算木を構成できるが、Wallace 木は構成できない。(4)の手法で構成できる最も高速、すなわち、最も回路段数の小さな加算木は4-2加算木である。

本研究では、Wallace 木をはじめ、桁上げ保存加算器で構成される部分積加算部がすべて、若干の回路の付加により、回路の段数に比例する個数のパターンでテスト可能(す

すべての単一セル機能故障を検出可能) であることを示す。Wallace 木の段数は演算数のビット数の対数に比例するので、Wallace 木が演算数のビット数の対数に比例する個数のパターンでテスト可能であることが判明した。これまで、Wallace 木など、4-2加算木より回路段数が小さな部分積加算部については、テストに必要なパターン数と演算数のビット数との間の関係は知られていなかった。

本研究では、任意の構成の部分積加算部が部分積生成部を含めて高々6L+5 個のパターンでテストできることを示し、さらに、実用的な構成の部分積加算部が高々2L+9 個のパターンでテストできることを示す。ここに、L は加算木における桁上げ保存加算器の段数である。

(6) テスト容易な演算回路の自動合成ツールのプロトタイプ

(2)で提案したテスト容易な並列プレフィクス加算器の構成法に基づき、演算数のビット数を指定すれば、テスト容易な Sklansky 加算器を自動合成するとともにテストパターン集合を自動生成するツールのプロトタイプを作成した。

また、(3)で提案したテスト容易な4-2加算木の構成法に基づき、演算数のビット数を指定すれば、テスト容易な4-2加算木乗算器(最終加算器を除く)を自動合成するとともにテストパターン集合を自動生成するツールのプロトタイプを作成した。

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3件)

1. Nobutaka Kito, Kensuke Hanai, Naofumi Takagi : 「A C-Testable 4-2 Adder Tree for an Easily Testable High-Speed Multiplier」、IEICE Transactions on Information and Systems, vol. E93-D, no. 10, 2783-2791 頁、2010 年、査読有
2. 鬼頭信貴, 高木直史 : 「けた上げ保存加算器で構成された部分積加算部をもつ乗算器のテスト」、電子情報通信学会論文誌D、J92-D 巻、7号、994-1002 頁、2009 年、査読有
3. 鬼頭信貴, 高木直史 : 「種々の部分積加算構造をもつテスト容易な乗算器の設計手法」、電子情報通信学会論文誌D、J91-D 巻、10号、2478-2486 頁、2008 年、査読有

[学会発表] (計9件)

1. 藤井真一 : 「遅延制約下におけるテスト容

易な並列加算器の設計手法」、電子情報通信学会ディペンダブルコンピューティング研究会、2011年3月18日、宮古島マリンターミナル(宮古島市)

2. 藤井真一 : 「テスト容易な並列プレフィクス加算器の自動合成手法の検討」、電子情報通信学会ディペンダブルコンピューティング研究会、2010年3月28日、八丈シーパークリゾート(東京都八丈島町)
3. 鈴木秀俊 : 「テスト容易な並列プレフィクス加算器の設計」、2009 年電子情報通信学会基礎・境界ソサイエティ大会、2009 年9月17日、新潟大学(新潟市)
4. 鈴木秀俊 : 「テスト容易な並列プレフィクス加算器」、第61回F T C研究会、2009 年7月17日、フォレストピア奥伊勢(三重県大台町)
5. 鈴木秀俊 : 「テスト容易な並列プレフィクス加算器の設計手法」、電子情報通信学会ディペンダブルコンピューティング研究会、2009 年6月19日、機械振興会館(東京都港区)
6. 鬼頭信貴 : 「乗算器の種々の部分積加算部の順序故障テスト」、電子情報通信学会2009年総合大会、2009 年3月17日、愛媛大学(松山市)
7. Nobutaka Kito : 「Level-testability of multi-operand adders」、17th IEEE Asia Test Symposium、2008 年11月26日、京王プラザホテル札幌(札幌市)
8. 鬼頭信貴 : 「Test generation for multi-operand adders consisting of full adders」、電子情報通信学会ディペンダブルコンピューティング研究会、2008 年6月20日、機械振興会館(東京都港区)

[その他]

ホームページ等

<http://www.lab3.kuis.kyoto-u.ac.jp/research/easytest.html>

6. 研究組織

(1) 研究代表者

高木 直史 (TAKAGI NAOFUMI)
京都大学・大学院情報学研究科・教授
研究者番号 : 10171422

(2) 研究分担者

高木 一義 (TAKAGI KAZUYOSHI)
名古屋大学・大学院情報科学研究科
・准教授
研究者番号 : 70273844

(3) 連携研究者

中村 一博 (NAKAMURA KAZUHIRO)
名古屋大学・大学院情報科学研究科・助教
研究者番号 : 90335076