

| | |
|--------------|---|
| Title | 冗長化アルゴリズムからの耐故障データパス自動合成 |
| Author(s) | 坪石, 優 |
| Citation | |
| Issue Date | 2009-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/8116 |
| Rights | |
| Description | Supervisor:金子峰雄, 情報科学研究科, 修士 |

冗長化アルゴリズムからの耐故障データパス自動合成

坪石 優 (0710051)

北陸先端科学技術大学院大学 情報科学研究科

2009 年 2 月 5 日

キーワード: 耐故障, 冗長化アルゴリズム, 高位合成, 整数線形計画法.

今日の集積回路製造技術の進歩により、集積回路は我々の身近な製品に活用され生活の中核を担うようになった。しかしながら、高いリアルタイム性を要求する用途において集積回路の動作中に発生する故障は重大な障害に発展する危険性を孕む。そのため、耐故障技術は集積回路の設計する上で重要なものとなる。耐故障技術は、一時的な故障である過渡故障のみを対象とする誤り検出・再試行や永久故障を含めて対象とするモジュール三重化などが代表的なものとされる。また、モジュール三重化に伴うハードウェアオーバーヘッドを低減するために、通常動作では故障検出のみを行い、故障時に故障箇所をシステムから切り離して正常部分のみでシステムを再構成する手法なども提案されている。しかし、個々の手法は再試行や再構成をする際にシステムを一時的に停止しなければならないか、または多くのハードウェアオーバーヘッドを要するなど、高いリアルタイム性と低いハードウェアオーバーヘッドを兼ね備えた耐故障技術は少ない。

ABFT(Algorithm Based Fault Tolerance) は計算アルゴリズム自身に誤り検出訂正を行う冗長計算を導入することで高いリアルタイム性と低いハードウェアオーバーヘッドを両立する代表的な手法ではあるものの、その適用は線形的な計算アルゴリズムに限定されそれ以外には原理的に適用できない。これに対して、適用可能な計算アルゴリズムが制限されることの無いよう単純なアルゴリズム三重化を基本とし、高位合成の枠組みを用いてハードウェア量と実行時間のトレードオフをはかりながら最適な耐故障データパス回路を合成しようとする基礎概念が提案された。合成されたデータパスが耐故障性を有するためには演算の演算器への割り当てや演算結果のレジスタへの割り当てといった資源割り当てに制約が発生する。この手法では、ポート演算を計算途中に挿入することで、こうした制約が変化したり、緩和されることになり、ハードウェアオーバーヘッドの低減に寄与する反面、設計問題は非常に複雑なものとなり最適化手法は確立していない。

本研究では、このアルゴリズム三重化と高位合成を組み合わせる耐故障データパス合成問題に対して、ポート演算の挿入位置が入力情報として与えられる部分問題について検討を行った。具体的には、与えられる計算アルゴリズムとあらかじめ決定されたポート演算の挿入位置から上記の基礎概念による耐故障性を持ったスケジュールと資源割り当てを行

う問題に対して、整数線形計画法による厳密解法と資源割り当てを優先して行う発見的手法の二つの手法を提案した。はじめに、整数線形計画法では演算間の依存関係と演算と演算器間の関係に加え、ポート演算と演算の依存関係の制約や冗長化された演算間の依存関係の制約を変数の線形代数式にして定式化できることを示し、これを解くことによって問題の解が得られることを示した。しかしながら、入力となる計算アルゴリズムの規模に対する制約式数の膨大さから、実用的な求解はごく小規模の問題にとどまり、規模の大きな問題を実用時間で解くことは難しいことがわかった。次に、発見的手法による求解により大規模な問題を解決可能なアルゴリズムを提案した。この手法では、まず、与えられる計算アルゴリズムとポート演算の挿入位置から、計算アルゴリズム全体をコーンと呼ばれる部分計算に分割する。次いで各コーンの間での資源共有の可能性を考慮し、耐故障性の維持を優先としたコーンへの演算器割り当てを行う。その後、与えられたコーンへの資源を活用してコーンに含まれる演算をリストスケジューリングすることで資源共有を実現したハードウェアと時間へのマッピングを行う。これにより、きわめて短時間で解を導出することができ、かつ整数線形計画法による厳密解と遜色ない解を得ることができることを明らかにした。

提案手法により、ポート演算挿入後の耐故障データパス合成と最適化が可能となった。実際に合成されるデータパスの構造や実行ステップ数はポート演算挿入位置によって異なることになり、またシステムの信頼性もポート演算挿入位置によって異なることになる。提案手法はこうしたポート演算挿入位置の最適化に対しても重要な部分ルーチンとなるものと考えられる。

データパス合成を完成させるためには、演算結果に対してそれを格納するレジスタを決定するレジスタ割り当てが必要である。計算アルゴリズムのコーン分解が定まった時点で耐故障性からみた演算結果間のレジスタ共有の可否が決まり、スケジュールの結果から生存期間からみた演算結果間のレジスタ共有の可否が決まることになる。このことから、レジスタ割り当て問題は個々の演算結果を頂点とし、レジスタ共有可能な頂点間に辺を配した共有グラフにおいてクリーク分割を求める問題として定式化できる。これらの詳細な検討は今後の課題となっている。