



論理関数の自由度の新しい表現方法とその論理合成への応用

新しいタイプの論理素子を使いこなすための新しい論理合成手法の確立

山下 茂

立命館大学 情報理工学部
教授

研究の背景

1990年代までにLSI設計のためのCADの技術は大幅に発展し、実用的なLSIをCADツールにより設計できるようになった。CADツールの中に使われる設計技術の中でも特に論理合成の技術はかなり成熟したと考えられ、純粋な論理合成に関する新しい研究成果が学会などで発表される件数は少なくなってきた。

しかし、その一方で1990年代からよく使用されるようになってきたFPGA(Field Programmable Gate Array)などのようにその論理的なプリミティブが従来のAND/OR/NOTのような単純な論理でない場合、従来の論理合成技術を単純に利用するよりも、新たな論理合成技術の確立がより望ましいと考えられる。

また、近年LSIの信頼性の向上のために物理設計のレベルからアーキテクチャ設計のレベルで多くの研究が行われているが、論理合成のレベルでは、3重化を単純に利用した手法など限られたものしか研究が進んでいない。今後のLSI設計技術の更なる発展のためにも、論理合成のレベルで信頼性を向上させるための新しい設計技術の出現が望まれている。

研究の成果

本研究では、LSI設計技術の中でも純粋に論理合成の技術に関して新しい理論的なブレークスルーを目指して、新しい概念の創出とその論理合成への応用に関して研究を進めてきた。

1. 論理関数の自由度の新しい表現方法(SFPD)の考案

一般に論理回路には多くの潜在的な冗長性があり、各結線で実現すべき論理関数にはある程度の自由度(回路の出力が変わらないという条件でその関数が変化可能な範囲)が存在する。そのような自由度は論理回路の設計や回路の最適化の際に利用できる。従来のAND/OR/NOTで構成される論理回路では、回路内の各部分で実現される関数の値がある入力パターンの時に0か1にならなければならないのか、どちらでも良い(ドントケア)のかで論理関数の自由度を表現していた。

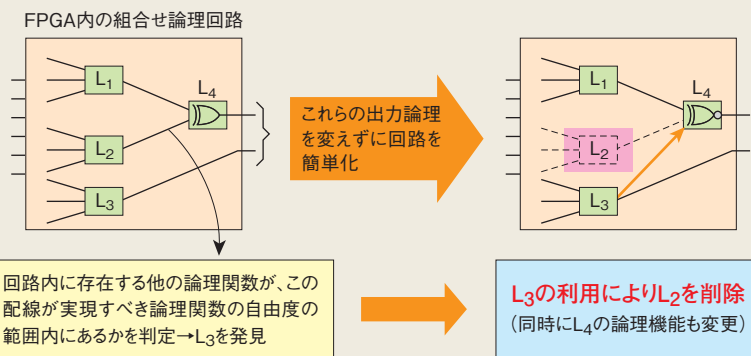
上述した従来のドントケアの概念と全く異なり、本研究ではSFPD(Sets of Pairs of Functions to be Distinguished)と呼ばれる関数の対の集合で関数の自由度を表現する概念と手法を考案した[1, 3]。SFPDの考え方では、「関数の対(f_0, f_1)に対して、 $f_0=1$ となる入力の時と $f_1=1$ となる入力の時とは論理関数 f の値(0または1)が異ならなければならない」という条件を

複数持つことにより関数 f の自由度を表現する。

また、本研究ではSFPDによる自由度の効率的な計算手法や回路変換への利用方法なども考案した[1, 3]。さらに、SFPDを利用すれば、どんな場合でも従来からの表現方法に比べて必ず大きな自由度が表現できることを数学的に証明した[3]。

SFPDを利用すれば、従来からの自由度の表現方法を利用する場合よりも、回路の変形の自由度が増大する。それを利用して、当初はLUT型のFPGAの回路の最適化に利用されたが、通常の回路の最適化にも応用されるようになり、現在では配置・配線ツール、テスト生成ツール、仕様変更への対応手法など、LSI設計の幅広い分野で応用されている。また、本研究のSFPDに関する2つの論文[1, 3]が数多くの学術論文で参照され、大学でのLSI設計技術の講義でも紹介されるなどSFPDは論理合成の分野ではメジャーな概念になりつつある。

図1 SFPDを用いたFPGA向けの論理回路の最適化手法



2. SPFDの概念を利用したLUT型FPGA向けの論理合成システムの開発

本研究では、SPFDをLUT型FPGAの回路の最適化[2, 5, 6]に利用し、その手法を様々な他の論理合成の技術と組み合わせることにより、LUT型FPGA向けのテクノロジーマッピングを行う設計ツール[4]を開発した。

例えば、図1に示す回路において、 L_2 の出力論理関数の自由度をSPFDで表現して、 L_3 の出力論理関数がある自由度に含まれると分かれば、図1のように回路を変形して L_2 を削除することが可能である。

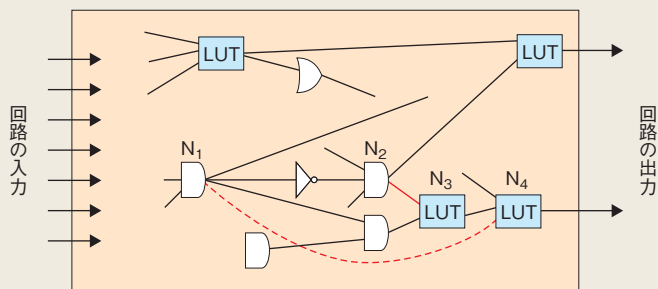
開発した手法[4]は、当時の商用の設計ツールよりも回路規模が小さく、配置・配線後の最大遅延時間も少ない回路を合成することができた。

3. SPFDの概念を利用した少ないオーバーヘッドで故障に対応可能な論理回路モデルの考案

本研究では、SPFDの応用として、従来からのようにモジュールを多重化することなく製造後の故障を回避して回路を使用可能とすることが可能なPPC (Partially-Programmable Circuit) という回路モデルを考案した[7]。

PPCの回路モデルを図2に示す。点線は設計後に追加する冗長結線であり、最初に回路を合成した際にはない結線である。もしSPFDにより、 N_1 から N_4 へ冗長な結線(図の点線)をつなげば N_2 から N_3 への結線が冗長になると判断可能であるとする。それは、 N_1 から N_4 へ冗長な結線をつなぐだけで、もし N_2 から N_3 の結線に故障が起こったとしても、 N_3 や N_4 などのLUTの内部論理を変更することにより N_2 から N_3 の結線を論理的に冗長とすることが可能であることを意味する。つまり、ある程度の数の冗長な結線をあらかじめ追加しておくだけで、回路の多くの部分の結線の故障にLUTの論理機能の変更をするだけで対応することが可能となることが期待できる。

図2 PPCの利用例



PPCは、従来手法の冗長なモジュールの追加による面積の増大や、FPGAのように配線を自由に変更できる機能による速度低下などのオーバーヘッドなしで耐故障性を向上させることができる画期的な回路モデルであると考えられる。

本研究では、SPFDを利用してPPCの回路を設計する手法および故障の修正方法も考案しベンチマーク回路でその有効性を確認した[7]。

将来の展望

SPFDは、当初はLUT型FPGAの最適化に利用されたが、その後配置・配線、テス

ト生成、仕様変更への対応などLSI設計技術の様々な分野に応用できることが分かり、さらに様々な応用が広がることを期待できる。

また、SPFDの概念を応用して考案したPPCの回路モデルは、今後ますます信頼性の確保が難しくなると考えられる次世代素子によるLSIの設計技術に、新たな設計手法を提供すると考えられる。これにより、今後の半導体産業において特に製造歩留まりを向上させる論理回路設計技術を確立し、将来的には廃棄するLSIの削減や一部が故障しても動作し続けるLSIの設計手法の確立などに寄与することが期待できる。

References(参考文献)

- [1] S. Yamashita, H. Sawada, A. Nagoya, "A New Method to Express Functional Permissibilities for LUT based FPGAs and Its Applications," Proc. Int'l Conf. Computer-Aided Design (ICCAD '96), pp. 254-261, Nov. 1996.
- [2] T. Kouda, S. Yamashita, Y. Kambayashi, "Reduction of the Number of FPGA Blocks by Maximizing Flexibility of Internal Functions," IEICE Trans. Fundamentals, Vol. E81-A, No. 12, pp. 2554-2562, Dec. 1998.
- [3] S. Yamashita, H. Sawada, A. Nagoya, "SPFD: A New Method to Express Functional Flexibility," IEEE Trans. CAD, Vol. 19, No. 8, pp. 840-849, Aug. 2000.
- [4] S. Yamashita, H. Sawada, A. Nagoya, "A General Framework to Use Various Decomposition Methods for LUT Network Synthesis," IEICE Trans. Fundamentals, Vol. E84-A, No. 11, pp. 2915-2922, Nov. 2001.
- [5] K. Tanaka, S. Yamashita, Y. Kambayashi, "SPFD-based One-to-Many Rewiring (OMR) for Delay Reduction of LUT-based FPGA Circuits," Proc. of GLS-VLSI 2004, pp. 348-353, Apr. 2004.
- [6] K. Tanaka, S. Yamashita, Y. Kambayashi, "SPFD-Based Flexible Transformation of LUT-Based FPGA Circuits," IEICE Trans. Fundamentals, Vol. E88-A, No. 4, pp. 1039-1046, Apr. 2005.
- [7] S. Yamashita, H. Yoshida and M. Fujita, "Increasing Yield Using Partially-Programmable Circuits," Proc. of The 16th Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI2010), pp. 237-242, Oct. 2010.