

論理合成システム EXLOG におけるテクノロジマッピング

4N-6

藤田 悟 岩本 雅彦 渡辺 正信

日本電気(株) C&C システム研究所

1 はじめに

論理合成システムのテクノロジマッピング部は、各種テクノロジへの対応が容易で、かつ面積最小などの要求条件を満たす必要がある。論理合成システム EXLOG[7] はもともとルールベースのシステムであるが、テクノロジ依存ルールの最適性問題・記述困難性などから、アルゴリズム化を検討している。この結果、テクノロジ依存のデータベース(面積、遅延など)を用いて、容易なテクノロジ対応を実現し、また、出力回路の面積に関しても従来より優れたものが実現できた。本稿は、その中心となる制約を利用したテクノロジマッピングの手法について報告する。

2 テクノロジマッピングシステムの概要

本テクノロジマッピングの対象とするデータは、論理多段化の終わった組合せ論理回路のデータである。

木構造の(出力分岐のない)論理式は、従来の研究[1][3][5]で、最適解を求めるアルゴリズムが提案されている。すなわち、論理式の出力側から、マッチする部品を順に適合させることをバックトラックしながら行い、最適な解を得れば良い。

これに対し、木構造をとらない論理式、すなわち、出力が多分岐となる回路を含む論理式の最適化に関しては、上記のような単純なバックトラックによる最適解の発見ができない。なぜなら、多分岐点において、最適解の判断基準がネットワーク中に広がってしまうからであり、部分最適解を全体の最適解へ展開させることができなくなるからである。

そこで、局所解を全体の最適解へ展開する手法として、本システムでは部分木間の関係を制約として解き、全体の最適解を求める方法を用いた。この方法の確立により、テクノロジマッピングは、部分木構造内の最適解の探索、部分木間最適化という、大きく2段階で実現されるようになる。

3 部分木間の最適化

3.1 部分木間の最適化の基本構想

部分木内の最適化を行った結果に対して部分木間の最適化を行う。部分木内の最適化では、入出力に付加されるインバータの面積評価を行わない解を求め、単に、インバータが必要であったかどうかの付加情報だけを残す。通常、解は複数獲得され、この中の最適な解の組合せを部分木間の問題として解く。部分木の接点において、

- 全てのピンにインバータが存在しなければ、0
- 全てのピンにインバータが存在すれば、0
- それ以外は、1

という評価を行い、この合計が入力論理式全体で最小になる組合せを制約問題として解く。この制約問題は、全数探索では組合せ的爆発を起こす。そこで、

Technology Mapping in EXLOG

Satoru FUJITA, Masahiko IWAMOTO, Masanobu WATANABE
C&C Systems Research Laboratories, NEC Corporation

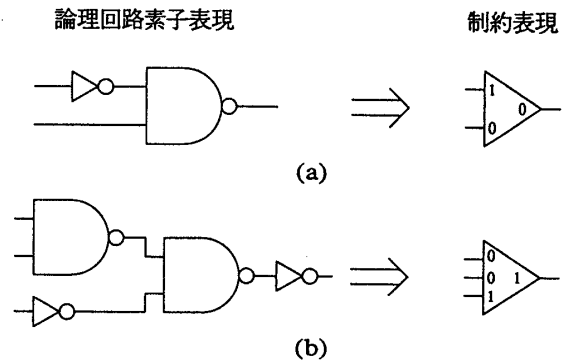


図1: 制約表現

1. 全体に関して、詰めやすいところから問題を限定し、制約問題を小さくする。
2. 制約条件によって一意に解が決定されなかった残った部分については、任意順に決定的に解を定める。

という方法で解くことを検討した。実験からは、1. の部分で問題の大半は解けており、2. で解かなければならない制約問題は簡単なものになる。

3.2 部分木間最適化の概要

部分木ごとに複数の最適解が得られている状態から始める。まず、制約を表現しやすい形にデータをモデル化する。すなわち、インバータのあるピンに対応する制約表現部を1とし、それ以外を0とする。図1で、(a)は、2-入力1-出力の回路で、上段の入力にインバータが付加されているので、そのピンに対応する制約表現を1にしている。(b)は、3-入力1出力の回路で、最下段の入力と出力にインバータが付加されているので、その対応する制約表現に1が記される。その他のピンは0で表現される。

次に、この制約表現をもとに制約解消部が、制約問題として解く。これの詳しい動作説明は、次節にて行う。ここでの動作は、0、1で表現された制約表現を用いて、部分木間の最適な組合せ対を決定することである。部分木の結合点において、以下の評価を行う。

- all 0 または、all 1 ならば、制約が成立する。(部分木の中にインバータが不要である。)
- 上記の条件が満たされない場合、制約は不成立となる。(部分木の中にインバータが必要である。)

図2に部分木間に課せられた制約の典型例を記す。破線の長方形が論理回路の部分木を表し、三角形で示されたものは、図1の制約表現である。複数の三角形は、部分木の最適解が複数存在することを表す。(a)で、後段の2つの解候補のうち、上の候補を選ぶことにより、all 1の制約が成立する。このようにして、論理回路全体で、できるだけ多くの制約が成立するように問題を解く。

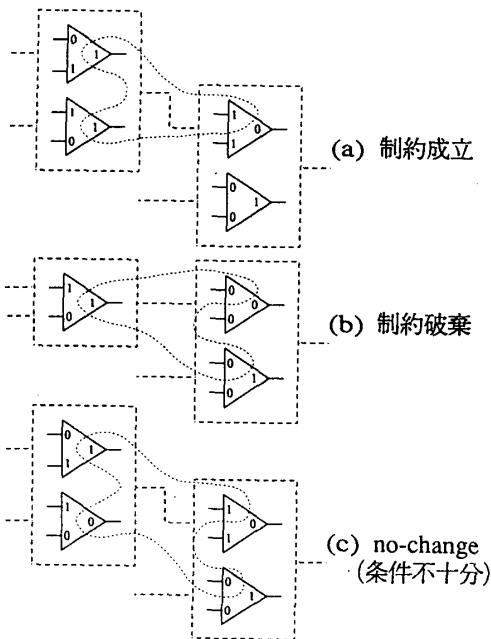


図 2: 制約問題

3.3 制約処理系

制約充足問題の研究 [2][4] は、全ての制約を満たす解が存在する問題を対象にするのに対し、この問題は、制約がいくつ解けるかの問題であり、制約充足問題の手法は直接利用できない。そこで、この制約問題を適当時間内に解くための効率の良い制約処理系を検討した。以下にその概略を示す。

1. 制約容易性評価 制約実現容易性を、制約に関係する部分木回路の最適解の数の積と定義し、それを計算する。
2. 制約選択 制約実現容易性評価の評価値の小さいもの、すなわち、制約の実現が容易なものを選択する。
3. 制約評価 制約に解があるかどうかを判断する。制約が不成立の場合、その制約を破棄する。制約が成立する場合、制約の伝搬を行う。制約変数の具体化が不十分の場合は、その制約は後で解くこととし、一時保留する。
4. 保留した制約の再評価 残りの制約がある場合は、1. の制約実現容易性評価の実行へ戻り、そうでない場合は、保留していた制約を再度評価対象にした後、1. に戻る。
5. 保留した制約の後処理 再評価後も保留された制約に対しては、後処理で解を絞り込む。この残った問題は、制約充足問題とほぼ同じアプローチで解くことができる。制約変数に対し制約条件の厳しいものから順に制約を満たすように、解を割り当てていくことで、全ての制約変数の解を決定する。

以上により、制約解消を行う。

3.4 部分木間最適化の効果

部分木間の最適化手法により、全体の回路を部分木に分割しても全体としての最適解に近い解を選択することが可能となった。これは、従来の木構造を持つ回路を対象にした最適解の探索に比べて、その扱える論理式の対象の制限を緩和した。また、大きな論理合成問題に対して、それを小さな問題に分割して解いても、その最適解として充分なものを保証することができるようになった。

また、制約解消の効率の良い解法を提供した。制約解消を全数探索で行うと、組合せ的爆発が生じるが、ここに述べた

表 1: 性能評価 (セル数)

	部分木内 最適化のみ	部分木間 最適化	比率
5xp1	101	68	0.67
9sym	191	129	0.67
bw	146	113	0.77
duke2	314	234	0.74
f2	16	12	0.75
rd53	40	30	0.75
rd73	99	77	0.77
sao1	150	116	0.77
sao2	138	95	0.68
vg2	67	56	0.83

方法により、妥当な時間内に制約問題を解くことができる。得られた解の品質も最適解に対して充分であることが実験より確認された。

4 性能評価

DAC'86 のベンチマーク [6] に対し、評価を行ったものを表 1 にまとめる。ゲートアレイの面積をセル数を用いて評価したもので、部分木内の最適化だけのものと部分木間の最適化を加えたものを比較している。

5 おわりに

以上、テクノロジマッピングの最適化について述べた。遅延評価などに問題を残しているが、初期の面積最適化の目標は達成された。今後、順序回路との融合における問題などを含め、検討を行っていく。

参考文献

- [1] Cohen, W. W., Bartlett, K. and de Geus, A. J., Impact of Metarules in a Rule Based Expert System for Gate Level Optimization, *Proceedings of ISCAS*, pp. 873-876, 1985.
- [2] Dechter, R. and Pearl, J., Network-Based Heuristics for Constraint-Satisfaction Problems, *Artificial Intelligence*, Vol. 34, pp.1-38, 1988.
- [3] Detjeus, E., Gannot, G., Rudell, B., Sangiovaum-Vincentelli, A. and Wang, A., Technology Mapping in MIS, *Proceedings of ICCAD*, pp. 116-119, 1987.
- [4] Freuder, E. C., A Sufficient Condition for Backtrack-Free Search, *Journal of the ACM*, Vol. 29, No. 1, pp. 24-32, 1982.
- [5] de Geus, A. J., Cohen, W. W., A Rule-Based System for Optimizing Combinational Logic, *IEEE Design & Test*, pp. 22-32, 1985.
- [6] de Geus, A. J., Logic Synthesis and Optimization Benchmarks for the 1986 Design Automation Conference, *Proceedings of 23rd DAC*, p.78, 1986.
- [7] 岩本雅彦, 渡辺正信, 河原林政道, 前田直孝, フルカスタム LSI 論理合成エキスパートシステム EXLOG における設計過程モデル, 信学会論文誌, Vol. J72-A, No. 8, pp. 1163-1171, 1989.