

代数的アプローチに基づく最小トランジスタ回路合成手法*

吉田 浩章 池田 誠 浅田 邦博[†]

東京大学 大学院 工学系研究科 電子工学専攻[‡]
東京大学 大規模集積システム設計教育研究センター (VDEC)[§]

〒 113-8656 東京都文京区本郷 7-3-1

E-mail: hiroaki@silicon.u-tokyo.ac.jp

概要 本稿では、最小トランジスタ数からなるスタティック CMOS トランジスタ回路を代数的アプローチによって合成する手法を提案する。提案アプローチは複数の回路構造を一つのグラフとして表現する手法に基づいている。このグラフ上において代数的変形を繰り返し適用することで可能な回路構造を暗黙的に網羅し、その中から最小解を効率よく見つけ出す。セルによる実現を前提としたベンチマーク回路を用いた計算機実験では、提案手法が短時間で最小回路を合成可能であることを確認し、その有効性を示す。

Abstract This paper presents an algebraic approach for synthesizing static CMOS circuits with the minimum number of transistors. Our approach relies on a graph structure which encodes multiple circuit structural configurations. The proposed procedure implicitly enumerates possible circuit configurations via algebraic transformations in the graph structure, and efficiently finds a minimum solution among them. Experimental results on a benchmark suite targeting standard cell implementations show that the proposed procedure successfully generated the minimum circuits in a reasonable runtime.

1 はじめに

トランジスタレベル最適化はゲートレベル最適化に比べ非常に高い性能を得ることが可能であることが知られている。セルレイアウト生成技術の近年の著しい進歩 [2][3] により、トランジスタレベル最適化はより現実的で実用的なものとなっている。最近では、そのような最適化手法が、実際の大規模設計事例に適用され、大幅な性能改善を達成したとの報告もある [4]-[6]。トランジスタレベル最適化は特にタイミングや消費電力の最適化に応用される場合が多いものの、面積の削減に対しても大きな可能性をもっている。面積最小化は論理合成において最も古典的な問題の一つであり、一般的に多段論理ネットワークのリテラル数削減によって達成される。初期の研究 [7][8] は厳密な多段論理最小化アルゴリズムを提案しているが、しかしながらそのアルゴリズムは効率が悪く、非常に小さな関数にのみ適用可能であった。最近になり [8] の効率を改善したアルゴリズムが提案されたものの、それでも 12 トランジスタからなる回路の合成に数時間を要する [9]。その計

算複雑性に加え、このアルゴリズムは 2 入力 NAND ゲートの数を最小化の目的関数としており、トランジスタ数を最小にすることを目的としていない。また、豊富なセルライブラリを用意しテクノロジマッピング技術を応用することで、任意のスタティック CMOS 回路を合成する手法も提案されている [10]。この手法ではあらかじめ数万個以上のセルを準備しなくてはならず、現実的ではない。

スタンダードセルに基づく設計フローにおいては、トランジスタレベル最適化はセル単位で適用される。この理由から、スタンダードセルとして実現可能な程度の大きさの回路を現実的な時間内で合成可能であれば、多少効率の悪いアルゴリズムでも有用であると考えられる。そのような目的のもと、本稿では最小スタティック CMOS トランジスタ回路を合成する手法を提案する。解空間を制限するため、可能な回路構造は代数的な変形のみによって探索される。提案手法は複数の回路構造をエンコード可能なグラフ構造 [1] を利用する。このグラフを利用することによって、可能な回路構造を暗黙的に列挙した後、最小解を効率的に発見する。最後に、スタン

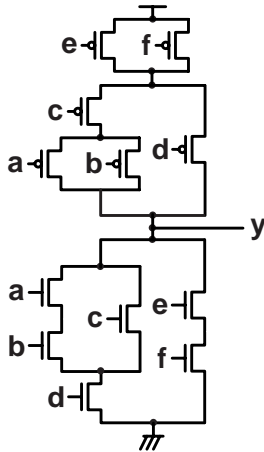


図 1: スタティック CMOS ゲート

ダードセルとしての実現可能な規模の例題を用いた
 計算機実験により、提案手法の妥当性を示す。

2 定義

2.1 スタティック CMOS 回路

スタティック CMOS ゲートはドレインとソース
 によって互いに接続されたトランジスタの集合であ
 る。また、互いに相補的な構造を持つ P 型トランジ
 スタ部分と N 型トランジスタ部分の 2 つの部分か
 らなる。これらのトランジスタを接点として見なす
 と、入れ子構造になった直並列(または並直列)接
 点回路網であり、ある論理関数を実現している。ス
 タティック CMOS ゲートは常に負ユネイト関数(単
 調減少関数)を実現している。例えば、図 1 に示す
 スタティック CMOS ゲートは $y = \overline{((ab + c) * d) + ef}$
 の論理関数を実現している。スタティック CMOS 回
 路はスタティック CMOS ゲートからなる回路と定
 義される。

2.2 等価 AND2/INV ネットワーク

ブーリアンネットワークはそれぞれのノードが論
 理関数を持つ閉路のない有向グラフとして定義さ
 れる。AND2/INV ネットワークはノードの種類が
 2 入力 AND ゲートもしくはインバータに限定され
 たブーリアンネットワークである。負ユネイト木は
 AND2/INV ネットワークの部分木であり、次のよう
 な性質を持つ: (1) 根は常にインバータである、(2)
 根から葉に至るどのような経路も奇数個のインバー

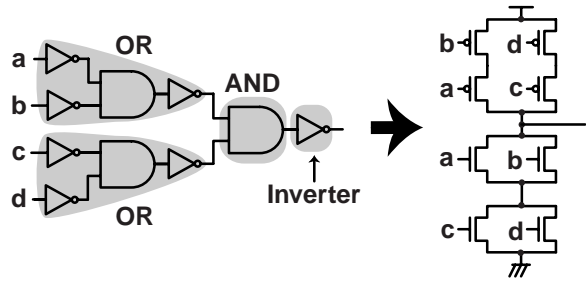
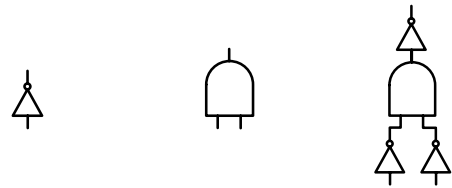


図 2: 負ユネイト木 (コスト=8)



(a) インバータ
 パターン
 (コスト=2)
 (b) AND パ
 ターン
 (コスト=2)
 (c) OR パ
 ターン
 (コスト=2)

図 3: 負ユネイト木を構成する基本パターン

タを持つ。言い換えれば、負ユネイト木は根にイン
 バータを持つ AND-OR 木と見なすことができる。
 負ユネイト木はその木構造を入れ子になった直並列
 ネットワークに変形することで一意にスタティック
 CMOS ゲートにマッピングすることができる。ある
 負ユネイト木からスタティック CMOS ゲートへの
 マッピングの例を図 2 に示す。等価 AND2/INV ネット
 ワークは互いに独立な負ユネイト木からなるネット
 ワークである。

負ユネイト木もしくは等価 AND2/INV ネットワー
 クのコストは、対応するスタティック CMOS 回路
 のトランジスタ数として定義される。負ユネイト木
 を構成する 3 つの基本パターンを図 3 に示す。どの
 パターンのコストも 2 であるため、負ユネイト木の
 コストの計算は、パターンの数の倍として求まる。
 これは以下のような考察に基づいている: (1) スタ
 ティック CMOS ゲートのトランジスタ数は入力数
 の倍である、(2) 負ユネイト木に新たなパターン(パ
 ターン b もしくは c) を加えることは入力数を一つ
 増やすことになる。同様に等価 AND2/INV ネット
 ワークのコストもパターンの数の倍として求めるこ
 とができる。

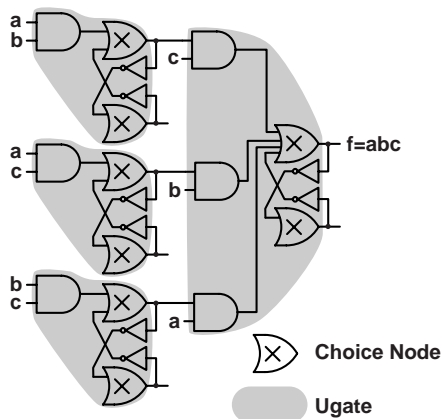


図 4: Mapping graph

2.3 マッピンググラフ

Lehman らによって提案されたマッピンググラフ [1] は一つのグラフ構造に複数の AND2/INV ネットワークを効率的にエンコードすることができる。マッピンググラフは AND2/INV ネットワークに選択ノードという新しい種類のノードを加えたものである。マッピンググラフ中において、選択ノードの出力はそのグラフ中で唯一のある論理関数を表現している。言い換えれば、どのような 2 つの選択ノードも同じ論理関数を持たない。等価な論理関数をすべてのインパータと 2 入力 AND ノードは対応する選択ノードの入力に接続される。図 4 は $f = abc$ を表現する複数の AND2/INV ネットワークをエンコードしたマッピンググラフの例である。図において、マッピンググラフは互いに共通部分を持たない ugate と呼ばれる部分グラフに分割できる。ugate 内においてインパータによって構成される閉路が存在するが、これは任意の数のインパータチェーンをエンコードするための機構である。それぞれの選択ノードについて入力を適当に選びだすことによって、マッピンググラフにエンコードされたある AND2/INV ネットワークを取り出すことができる。

3 合成手法

3.1 手法の概要

提案手法では、与えられた論理関数の集合について、これを実現する最小数のトランジスタによって

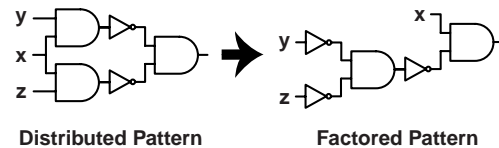


図 5: 分配律変形

構成されるスタティック CMOS 回路を発見することをその目的とする。

提案手法は大きく分けて 2 つのステップからなる。最初のステップは代数的変形を利用して、可能な AND2/INV ネットワークを列挙するマッピンググラフを構築する。次のステップは、マッピンググラフ中から最小のコストを持つ等価 AND2/INV ネットワークを発見することにより、最小スタティック CMOS 回路を生成することである。

3.2 マッピンググラフ構築手法

マッピンググラフの構造とともに、Lehman らはあるブーリアンネットワークのすべての可能な代数的分解をマッピンググラフにエンコードする手続きを提案した。まず、ブーリアンネットワークは適当な AND2/INV ネットワークに分解され、次に隣接した AND ゲートを統合してより大きな AND ゲートにする手続きを可能な限り行う。このネットワークのそれぞれの AND ゲートのすべての分解をエンコードすることにより、初期的なマッピンググラフを構築する。そのマッピンググラフに対して、網羅的に図 5 に示す分配律変形を適用することによって、最終的なマッピンググラフを得る。この手続きを便宜上標準グラフ構築手続きと呼ぶ。最終的なマッピンググラフは次のような性質を持つ [1]。

定理 3.1 標準グラフ構築手続きによって生成されたマッピンググラフは、あるブーリアンネットワークの任意の代数的変形のすべての AND2/INV ネットワークを含む。

標準グラフ構築手続きにおいて、マッピンググラフにエンコードされる AND2/INV ネットワークの集合は初期のブーリアンネットワークの構造に依存する。提案手法では、二段論理ブーリアンネットワークをその初期ネットワークとする。初期ネットワークにおいて、各々の初期出力は対応する論理関数のすべての主項の和を表現している。まず、標準

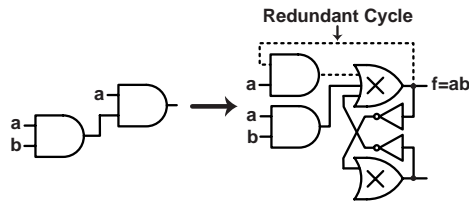


図 6: マッピンググラフ中における冗長な閉路の例

グラフ構築手続きを用いてマッピンググラフを生成する。すべての主項を含んでいるため、冗長な部分が存在する可能性があるが、そのような部分は選択ノードの機構によってグラフ中に閉路となって現れる。図 6 にマッピンググラフ中における冗長な閉路の例を示す。最後に ugate 中のインバータによる閉路以外のすべての閉路を除去する。この手続きを便宜上拡張グラフ構築手続きと呼ぶ。最終的なマッピンググラフは以下のような性質を持つ。

定理 3.2 拡張グラフ構築手続きによって生成されたマッピンググラフは、任意の非冗長な主項からなる二段論理ブーリアンネットワークの任意の代数的変形のすべての AND2/INV ネットワークを含む。

ここで、拡張グラフ構築手続きによって生成されたマッピンググラフは、非代数的分解によって生成されたネットワークや、主項でない項からなる二段論理や冗長な二段論理から得られるネットワークを必ずしも含むわけではないことに注意したい。

3.3 最小回路発見手法

最小回路発見は基本的にマッピンググラフにエンコードされたすべての等価 AND2/INV ネットワークを網羅することによって行う。トポロジカル逆順序に従い初期出力から順に選択ノードを訪れる。それぞれの選択ノードにおいて、図 3 に示したパターンに基づいてすべての可能なマッチングを行う。2 つ以上のファンアウトが存在する場合にはその選択ノードのすべての可能な複製も考慮に入れる。複製の後、選択ノードが初期出力のファンインであるが、複数のファンアウトを持つ場合には、その選択ノードにおいてはインバータパターンのみが許可される。これによって、探索される AND2/INV ネットワークがすべて等価 AND2/INV ネットワークであることが保証される。定義より、AND もしくは

OR パターンはインバータを根に持つ木の内部にのみあらわれるはずである。

上記の手続きは明示的にすべての回路構造を網羅しており、効率的ではない。そこでフロンティアという概念を導入する。フロンティアは、マッピンググラフをすべての初期出力を含む領域とすべての初期入力を含む領域に二分割する、選択ノードの集合である。ここで、すべての初期出力を含む領域を内部領域、すべての初期入力を含む領域を外部領域と呼ぶ。内部領域は以下のような性質を持つ。

補題 3.1 あるフロンティアで分割可能であるどのような最小コスト等価 AND2/INV ネットワークもそのフロンティアによって決定される内部領域の等価 AND2/INV ネットワークを含む。

本稿では正式な証明は行わないが、以下のような性質を利用して背理法で証明が可能である: (1) あるフロンティアにおけるパターンマッチングは内部領域でのマッチング結果に依存しない、(2) 全体のコストは内部領域のコストと外部領域のコストの和である。この補題によって、あるフロンティアにおいてすべての解を記録する必要はなく、最小解のみを記録すればよいということがわかる。この考察より、提案手法はダイナミックプログラミングに基づいたものとなっている。

複数の AND2/INV ネットワークがエンコードされたマッピンググラフが与えられると、まずトポロジカル逆順序に従い選択ノード n にラベル $L(n)$ を与える。もし m から n に経路が存在すれば $L(m) > L(n)$ となり、同時に n から n に経路が存在すれば $L(m) = L(n)$ となる。もし m と n の間に経路が存在しない場合には $L(m) \neq L(n)$ となる。3.2 節の手法で生成されたマッピンググラフは ugate 内を除いて閉路を持たないため、異なる ugate に属する選択ノードは常に異なるラベルを持つ。フロンティアのラベルはそのフロンティアに属する選択ノード中で最も小さいラベルとする。2 つの異なるフロンティア f と g について、 f のラベルが g のラベルよりも小さいとき、 f は g の内部にあるという。

最小回路発見手法の基本的な手続きを図 7 に示す。手続きはまず初期出力だけからなるフロンティアの状態から出発する。フロンティアは対応する内部領域の最小解を保持しており、最初は空の回路が保持されている。ここで内部領域の最小解を持ったあるフロンティアを考える。まず、フロンティアが

Given: Mapping graph
Procedure Minimum Circuit
 Assign labels to choice nodes
 Record an initial frontier
for L=1 **to** maximum label
 for each frontier F with label L
 Get a choice node $C \in F$ whose label is L
 for each match M at C
 Expand the frontier by including M
 Record the expanded frontier with the minimum solution
 end for
 end for
end for
 Obtain the minimum solution associated with the final frontier
 Transform the minimum network to a static CMOS circuit
end Procedure

図 7: 最小回路発見手法の基本的な手続き

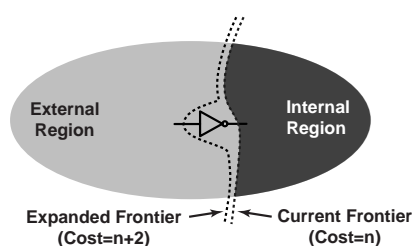


図 8: フロントアの拡張

ら最小ラベルを持つ選択ノードを選び、最初に述べたパターンマッチングを行い、フロントアを拡張する。ここで、元のフロントアのコストを n とすると、拡張されたフロントアのコストは $n+2$ となる。この新しいフロントアは対応する最小解とともに記録される。もしすでに同じフロントアが記録されている場合には新しいコストがより小さい場合にのみその記録を更新する。あるフロントアはすべての内部フロントアが処理された後においてのみ拡張可能である。フロントアの拡張の例を図 8 に示す。

初期入力のみからなるフロントアに対応する最小解が全体の最小解となる。最後に、この最小コスト等価 AND2/INV ネットワークのそれぞれの負ユネイト木をスタティック CMOS ゲートに変換することによって最小スタティック CMOS 回路を得る。

4 実験結果

我々は提案手法を SIS[11] 上に実装し、MCNC91 ベンチマーク回路からスタンダードセルとして実装可能な規模の回路を例題として選び、計算機実験を行った。例題回路はスタンダードセルとしては比較

的大きなものとなっている。スタンダードセルはその高さが固定であるため、大きな幅を持つトランジスタはレイアウトの段階で複数のトランジスタに分割される。トランジスタ数が非常に大きくなるとその形状は極端な横長となり、その後のセル配置およびセル間の配線に障害をきたしてしまう。

比較を行うために、同じ例題回路を SIS のテクノロジマッパーを利用してスタティック CMOS 回路を合成した。セルライブラリはインバータや AOI33 など 23 種類のスタティック CMOS ゲートからなる。それぞれのセルの面積は実際の実面積ではなく、そのセルのトランジスタ数となっている。これによって、総セル面積が回路の総トランジスタ数と等しくなる。理論的には、このライブラリを使用して面積最小化を行うことで最小トランジスタ回路が生成される。今回は、まず初期論理最適化として `script.rugged` を行い、面積最小テクノロジマッピング (`map -m 0.0`) を行った。

表 1 に SIS と提案手法の計算機実験結果の比較を示す。表において、右端の列はトランジスタ数の削減率を示す。本実験では SIS に比べて最大で 37.5% 小さいトランジスタ回路を生成しており、また実行時間も現実的である。しかしながら、提案手法は 40 個を越える回路の合成は現実的な時間で終了することができなかった。これは、中間的なフロントアの数が爆発的に増大してしまったためである。

5 結論

本稿では、最小トランジスタ数からなるスタティック CMOS トランジスタ回路を代数的アプローチによって合成する手法を提案した。提案手法は複数の回路構造を効率的に表現するグラフ構造を利用し、暗黙的に可能な回路構造を網羅する。また、ダイナミックプログラミングに基づいた手続きによって効率的に最小解を発見する。スタンダードセルによる実現を想定した例題回路に対する計算機実験では、実用的な時間で最小回路を生成できることを示した。またこの実験において、既存のゲートレベルのアプローチに比べ、提案手法が非常に小さい回路を生成することがあることを示した。これは、トランジスタレベル最適化が面積最小化に対しても大きな可能性を秘めていることを再確認するものである。

よく知られているように、パストラジスタ等の他のスタティック CMOS 回路方式が、スタティック

表 1: SIS と提案手法の比較

回路	入力数	出力数	SIS		提案手法		削減率 [%]
			トランジスタ数	CPU 時間 [秒]	トランジスタ数	CPU 時間 [秒]	
b1	3	4	26	0.1	26	0.0	0.0
C17	5	2	22	0.1	22	3.3	0.0
cm42a,e,f	4	2	22	0.1	20	0.1	9.1
cm42a,g,h	4	2	24	0.1	18	0.1	25.0
cm42a,i,j	4	2	22	0.1	18	0.1	18.2
cm42a,k,l	4	2	20	0.1	18	0.1	10.0
cm42a,m,n	4	2	22	0.1	18	0.1	18.2
decod,f,g	5	2	28	0.1	18	5.9	35.7
decod,h,i	5	2	28	0.1	20	6.0	28.6
decod,j,k	5	2	32	0.1	20	6.0	37.5
decod,l,m	5	2	28	0.1	20	5.9	28.6
decod,n,o	5	2	32	0.1	20	5.9	37.5
decod,p,q	5	2	28	0.1	20	5.9	28.6
decod,r,s	5	2	28	0.1	20	5.9	28.6
majority	5	1	32	0.1	20	1.1	37.5
t	5	2	22	0.1	22	0.0	0.0
x2,k,l	3	2	22	0.1	20	0.0	9.1
x2,m,o	6	2	22	0.1	20	1.4	9.1
z4m1,27	3	1	20	0.1	20	0.1	0.0

CMOS 回路に比べ少ないトランジスタ数で論理関数を実現することがある。今後の課題としては、本提案手法の他回路方式への拡張が挙げられる。

参考文献

- [1] E. Lehman *et al.*, “Logic Decomposition During Technology Mapping,” *IEEE Trans. on Computer-Aided Design*, vol. 16, no. 8, pp. 813–834, Aug. 1997.
- [2] M. Guruswamy *et al.*, “CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 327–332, Jun. 1997.
- [3] *abraCAD Documentation*, Synopsys, Inc., 2003.
- [4] R. Panda *et al.*, “Migration: A new technique to improve synthesized designs through incremental customization,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 388–391, Nov. 1998.
- [5] M. Cote and P. Hurat, “Faster and Lower Power Cell-Based Designs with Transistor-Level Cell Sizing,” in *Closing the Gap Between ASIC & Custom*, pp. 225–240, Kluwer Academic Publishers, 2002.
- [6] D. Bhattacharya and V. Boppana, “Design Optimization with Automated Flex-Cell Creation,” in *Closing the Gap Between ASIC & Custom*, pp. 241–268, Kluwer Academic Publishers, 2002.
- [7] E. L. Lawler, “An Approach to Multilevel Boolean Minimization,” *Journal of ACM*, pp. 283–295, 1964.
- [8] E. Davidson, “An Algorithm for NAND Decomposition under Network Constraints,” *IEEE Trans. on Computers*, vol. 18, pp. 1098–1109, 1969.
- [9] R. Drechsler and W. Gunther, “Exact Circuit Synthesis,” *International Workshop on Logic Synthesis*, 1998.
- [10] E. Detjens, G. Gannot, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, “Technology mapping in MIS,” in *Proceedings of IEEE Int. Conf. Computer-Aided Design*, pp. 116–119, Nov. 1987.
- [11] R. K. Brayton *et al.*, “MIS: A Multiple-level Logic Optimization System,” *IEEE Trans. on Computer-Aided Design*, vol. 6, no. 6, pp. 1062–1081, Nov. 1987.