

充足可能性判定を用いた CMOS 論理セルレイアウトの階層的生成手法

飯塚 哲也[†] 池田 誠^{†,††} 浅田 邦博^{†,††}

[†] 東京大学 大学院 工学系研究科

^{††} 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: †(iizuka,iked,asada)@silicon.u-tokyo.ac.jp

あらまし 本稿では、CMOS 論理セルのレイアウトを高速に生成するためのセルレイアウトの階層的生成手法を提案する。本手法では、まず与えられたトランジスタレベルネットリストを分割し、それぞれの論理ブロックに対して我々が以前に提案した充足可能性判定を用いた CMOS 論理セルレイアウト生成手法を適用する。この時に次のプロセスである論理ブロック配置の際にブロック間でなるべく多くの拡散を共有することができるように新たなコスト関数を設ける。次に、得られた回路ブロックを同様の手法を用いて最小幅に配置・配線し回路全体のレイアウトを生成する。本手法は階層化を用いているため、フラットに生成した最小幅のレイアウトよりも大きい幅のレイアウトを生成する可能性があるが、ほぼ全ての回路に対して最小幅の配置を生成できることを示す。さらに、本レイアウト生成手法の処理時間を従来手法と比較することで、特にトランジスタ数の多い回路に対して処理時間を大幅に削減できることを示し、従来手法より規模の大きい回路にも適用可能であることを示す。また、本手法と商用ツールとの比較から、商用ツールよりもセル幅はわずかに増加してしまうが、約 3% の処理時間で 32 種の CMOS 論理回路のレイアウトを生成できることを示す。

キーワード CMOS 論理セル, レイアウト高速生成, 回路分割, 充足可能性判定

Hierarchical Layout Synthesis for CMOS Logic Cells via Boolean Satisfiability

Tetsuya IIZUKA[†], Makoto IKEDA^{†,††}, and Kunihiro ASADA^{†,††}

[†] Dept. of Electronic Engineering, University of Tokyo

^{††} VLSI Design and Education Center (VDEC), University of Tokyo

[†] 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

E-mail: †(iizuka,iked,asada)@silicon.u-tokyo.ac.jp

Abstract This paper proposes a hierarchical layout synthesis method for high-speed layout synthesis of CMOS logic cells. The proposed method partitions a given transistor-level netlist into blocks and place all transistors hierarchically. Intra-block placement uses an exact transistor placement method which is based on Boolean Satisfiability. In this step, a new cost function is introduced to maximize the number of the connections by diffusion sharing between blocks. All blocks are placed in the minimum area and the routability of a generated placement is checked using Boolean Satisfiability. The proposed method reduces the runtime for cell synthesis drastically. Although this method has possibility to generate wider placements than the exact minimum width placement generated flatly, the experimental results show that the width becomes larger for only 1 out of 32 cells. The comparison results between a commercial tool show that although the width of the layouts generated by our method is a little larger, the proposed method generates 32 CMOS logic cells in only 3% runtime.

Key words CMOS logic cell, high-speed layout synthesis, circuit partitioning, Boolean Satisfiability

1. はじめに

セルベースの設計法は大規模集積回路の設計において最もよ

く採用される設計方法の一つである。セルベース設計に用いられるスタンダードセルライブラリの設計自動化の研究は盛んに行われており、スタンダードセルの自動設計によって高品質な

レイアウトを生成することができ、またその設計期間は大幅に短縮される。スタンダードセル自動設計の分野ではこれまで多くの論文が発表されている。その中でも、プロセスに依存しない形式的なレイアウトをあらかじめ持っている手法が良く使われる [1]。この手法では短時間で人手設計とほぼ同等のレイアウトを生成することができるが、事前に登録されているレイアウトしか生成することができず、近年提案されているトランジスタレベルの回路最適化などにおいて、セルの再合成を行って必要となった新たな回路を生成し配置配線に用いると言ったようなアプリケーションへの適用は不可能である。また、シミュレーテッドアニーリング (SA; Simulated Annealing) を用いたセルレイアウト生成手法が Guruswamy らによって提案されている [2]。この手法では人手で設計されたセルよりも品質の良いセルレイアウトを生成できる場合があるが、非常に多くの処理時間を要する。また、これらのヒューリスティックでは最適解を保証しない。

最適解を求めるセルレイアウト生成手法も過去に多く提案されている。Gupta らは定数線形計画法 (ILP; Integer Linear Programming) を用いた幅最小 CMOS セルレイアウト生成手法を提案している [3][4]。この手法では CMOS セル内のトランジスタを二次元的に配置し、P トランジスタ、N トランジスタが配置された列を複数持つレイアウトの幅最小化を行っている。この手法では相補的な関係を持つ P トランジスタと N トランジスタをペアとしてこれらを常に上下に配置し、最小幅のレイアウトを生成する。Maziasz らは CMOS セルの高さと同様に最小化する手法を提案している [5]。この手法ではセル内配線を考慮しながら CMOS セルの高さと同様に幅を最小化する事ができる。しかし、この手法も先ほどと同様に相補的な MOS をペアとして扱っており、さらにこの手法で仮定しているレイアウトスタイルには、横方向のポリシリコン配線が使用できないなどの実用的なものと合致しない部分がある。

我々は [6] において充足可能性判定 (SAT; Boolean Satisfiability) を用いた CMOS 論理セルの幅最小レイアウト生成手法を提案した。ここでは、セル内トランジスタ配置およびセル内配線において実用的なレイアウトスタイルを定義し、そのスタイルに従ってセル生成問題の SAT 問題への定式化を行った。この手法では相補的な関係を持つトランジスタを独立に扱い、同一のゲート端子を持つ P/N トランジスタが上下に並ぶというスタイルを採用することで、[3][4][5] などの手法よりも幅の小さい配置を生成することを可能とした。しかし、この手法ではトランジスタ数の小さい回路に対しては非常に高速にセルレイアウトを生成することが可能であるが、トランジスタ数の大きな回路に対して処理時間が大きくなってしまいうという欠点があった。

本稿では、与えられたネットリストをトランジスタの接続関係によって分割しトランジスタ配置を階層的に生成することで、特にトランジスタ数の大きいセルのレイアウト生成時間を大幅に短縮する手法を提案する。分割されたブロック内のトランジスタ配置では従来手法 [6] で提案されている手法を用いて最小幅の配置を生成するが、この際に次のブロック配置において最小幅の配置を得るため、ブロック間の拡散の共有のためのコスト関数を新たに設けている。このコスト関数を設けることで、

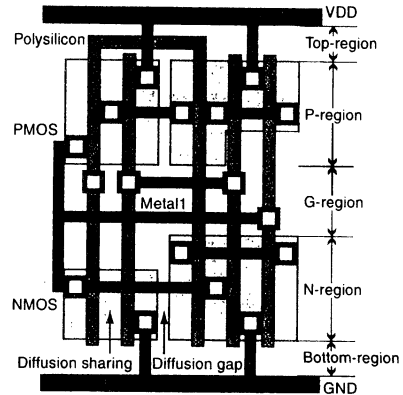


図1 本手法におけるセルレイアウトスタイル
Fig. 1 Cell layout styles of the proposed method.

最終的に生成される配置の幅をほとんど増加させることなく、階層化を用いて処理時間を大幅に削減することができる。生成された配置に対して従来手法と同様に充足可能性判定を用いてセル内配線可能性を判定することで、配線可能なセルレイアウトを高速に生成することを可能とする。

本稿では、まず第2節において今回採用したレイアウトスタイルについて述べ、第3節で提案するセルレイアウトの階層的生成手法について説明する。続いて第4節において実験結果について述べ、最後に第5節でまとめを述べる。

2. セルレイアウトスタイル

本手法で採用するレイアウトスタイルを表1に列挙し図1に図示する。これらのレイアウトスタイルは基本的に従来手法 [6] と同様のものである。P/N 列を一つずつ持つ CMOS セルの幅最小配置問題における設計スタイルの最初の指針は Uehara ら [7] によって与えられている。表1内の1から3は Uehara らのスタイルに基づいているが、Uehara らのスタイルでは相補的な関係を持つ P/N トランジスタを必ず上下に配置するのに対して、本手法ではゲート入力信号が共通な P/N トランジスタを上下に配置するというスタイルを採用している。このスタイルを用いることで全加算器などの一部の回路に対して、Uehara らのレイアウトスタイルよりも幅の小さい配置を生成することができる。表1内の4から14はセル内配線のスタイルを定義している。本手法では、従来手法と同様にセル内を5つの領域に分割している。図1に示すように、N/P 拡散上の領域をそれぞれ N-region, P-region とし、その間の部分を G-region とする。また、P-region の上の部分を Top-region, N-region の下の部分を Bottom-region と呼ぶ。レイアウトスタイルの6から13に示されているように、配線の種類によって、通過できる領域が制限されている。表1中の3および9について、異なるサイズのトランジスタの配置位置と G-region 内のゲート端子の接続にポリシリコン層を用いる点が従来手法と異なっているが、これらはセル内配線の可能性を向上させることが目的であり、大きな変更点ではない。本手法では以上のレイアウトスタイルの下で CMOS 論理回路のセルレイアウトを階層的に生成する。

表1 本手法におけるセルレイアウトスタイル
Table 1 Cell layout styles of the proposed method.

1. PMOSが上段、NMOSが下段の二列に配置される。
2. 同じゲート入力信号を持つPMOS/NMOSのみが縦に並んで配置される。
3. PMOSはその拡散領域の上端をそろえて、NMOSは下端をそろえて配置される。
4. セル内配線には第一メタル層とポリシリコン層のみを用いる。
5. 各端子は水平および垂直の直線状の配線で接続される。
6. PMOSの拡散同士を接続する配線はP-region内に配置される。
7. NMOSの拡散同士を接続する配線はN-region内に配置される。
8. ゲート端子同士を接続する配線はG-region, Top-region, Bottom-region内に配置される。
9. ゲート端子同士はTop-region, Bottom-region内ではポリシリコン層で、G-region内では第一メタル層またはポリシリコン層で接続される。
10. P/N拡散間はG-region内を縦に通過する第一メタル層によりP拡散の下端、N拡散の上端において接続される。
11. G-region内を縦に通過するメタル配線と他のゲート端子はG-region内で第一メタル層によって接続される。
12. 電源端子はP拡散の上端からTop-regionを通過して最上部の電源線に接続される。
13. 接地端子はN拡散の下端からBottom-regionを通過して最下部の接地線に接続される。
14. メタル配線と拡散およびゲートは一つのコンタクトによって接続される。

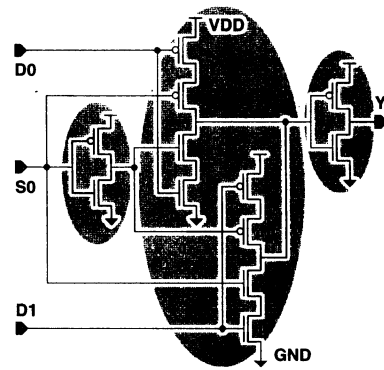


図2 Mux2セルの回路図とその論理ブロック分割結果
Fig. 2 Mux2 schematic and its logic block partitioning.

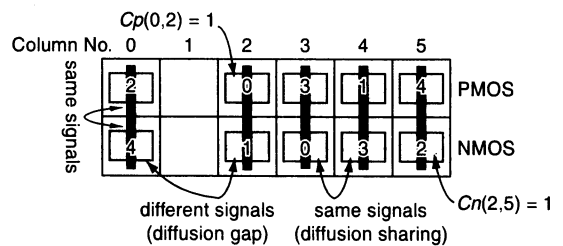


図3 トランジスタ配置問題の定義
Fig. 3 Problem definition of the transistor placement.

3. 階層的レイアウト生成手法

3.1 トランジスタ配置

本節では、提案するセルレイアウトの階層的生成手法におけるトランジスタ配置法について説明する。本手法ではまず与えられたトランジスタレベルネットリストをブロックに分割し、それぞれのブロックに対して充足可能性判定を用いて最小幅のトランジスタ配置を生成し、最後にそれらのブロックを同様に充足可能性判定を用いて最小幅に配置する。以後それぞれについて説明を行う。

3.1.1 回路分割

与えられたネットリストを分割し、階層的にレイアウトを生成する手法は過去にいくつか提案されている [8] [9]。これらの手法においてもレイアウト生成時間の短縮のため階層化を用いている。本手法でも同様に階層化を用いることで処理時間の短縮を図る。本手法では与えられたネットリストを論理ブロックと呼ぶ単位に分割する。論理ブロックは電源・接地端子を切断したときにソースおよびドレインで接続されているトランジスタの集合である。Mux2回路に対して本手法により回路分割を行った結果の例を図2に示す。本手法では図中にあるような出力がつかっているクロックトインバータは一つの論理ブロックとして分割される。トランジスタ配置において幅を最小化するためには拡散の共有を効率よく行うことが必要であり、拡散で接続されているトランジスタをひとつのブロックとする回路分割法により最終的なトランジスタ配置の幅が大きく増加しないことが予想される。

3.1.2 ブロック内配置の充足可能性判定への定式化

次に各ブロック内のトランジスタ配置の方法について説明する。本手法では充足可能性判定を用いて最小幅のブロック内トランジスタ配置を生成する。トランジスタ配置問題は N 個の N トランジスタと N 個の P トランジスタが与えられたとき、これら $2N$ 個のトランジスタを最小幅の領域内に配置する問題として定義することができ、図3に示すように配置する領域をグリッドに分け、各グリッドにトランジスタを配置して最小のコラム数を求める問題と言い換えることができる。第2節で説明したように、 P トランジスタは上段、 N トランジスタは下段に配置され、同じコラム内に配置されている P/N トランジスタは共通のゲート入力信号を持つものとする。隣接するコラムに配置されているトランジスタは最終的に生成される配置では互いに向かい合っている拡散を共有により接続するため同一信号の拡散を向かい合わせていなければならない。トランジスタの配置されていないコラムは拡散のギャップとなる。このような制約条件の中で与えられたトランジスタを配置することができる最小のコラム数を求める。これらの制約条件は従来手法 [6] で定式化されている制約条件と同様のものである。このとき、回路全体のトランジスタ配置の次のステップであるブロック配置の際に他のブロックとの間の接続をなるべく多く拡散の共有で行うことで全体の配置の幅を小さくすることができる。本手法による回路分割法で分割された各ブロック間で共有できる拡散は電源または接地線に接続されているもののみであるため、これらの信号が入力されている拡散をブロックの両端に配置し

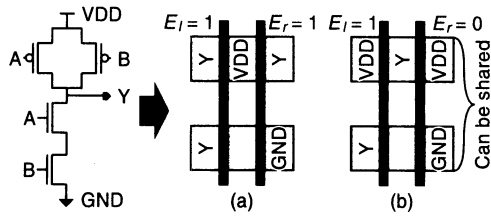


図4 ブロック間の拡散共有を最適化するために導入した変数

Fig. 4 Additional variables introduced to maximizing the number of the connections by diffusion sharing between logic blocks.

ておくことが望ましい。従って本手法では図4に示すように、電源・接地線に接続される拡散がブロック配置の端に配置されている場合に0となるような変数 (E_L , E_R) を新たに設け、この値を最小化する目的関数を導入することでブロック内の拡散の配置の最適化を行う。以下では本トランジスタ配置の定式化の詳細について説明を行う。

本定式化では各トランジスタに対して $W+1$ 個の0-1変数を必要とする。ここで、 W はトランジスタを配置する領域のコラム数である。 W 個の変数は各トランジスタがどのコラムに配置されているかを定めるため、1個の変数はそのトランジスタが反転して配置されているかどうかを判定するために必要となる。また、先ほど説明を行ったように両端の拡散が電源・接地端子に接続されているかどうかを判定するために2個の0-1変数を導入している。本定式化に用いられる変数を表2にまとめる。表は変数名とその変数の個数およびその変数の値を1とする状態をそれぞれ示している。本定式化ではこれらの変数を用いて上で説明した制約条件を和積型論理式および不等式の形の制約条件式として表現する。以後では和積形論理式による制約条件式をCNF (Conjunctive Normal Form) と呼び、不等式による制約条件式をPB (Pseudo Boolean) 制約と呼ぶことにする。これらの条件式について以下に示す。

目的関数: 左右両端に配置されている拡散は電源・接地線に接続されていることが望ましい。従って、本定式化における目的関数は以下のように表される。

$$\text{Minimize: } E_r + E_l \quad (1)$$

同一コラム上に2つのNMOSおよびPMOSが配置されてはならない: 本制約条件はPB制約として以下のように表現される。

$$\sum_{i=0}^{N-1} C_n(i, k) \leq 1, \quad 0 \leq k < W \quad (2)$$

$$\sum_{i=0}^{N-1} C_p(i, k) \leq 1, \quad 0 \leq k < W \quad (3)$$

全てのNMOSおよびPMOSは必ず1回現れる: 本制約条件はPB制約として以下のように表現される。

$$\sum_{k=0}^{W-1} C_n(i, k) = 1, \quad 0 \leq i < N \quad (4)$$

$$\sum_{k=0}^{W-1} C_p(i, k) = 1, \quad 0 \leq i < N \quad (5)$$

表2 本定式化に用いられる変数

Table 2 The variables used for the formulation.

変数名	個数	変数の値を1とする状態
$C_n(i, k)$	$N \times W$	NMOS i がコラム k に配置されている。
$C_p(i, k)$	$N \times W$	PMOS i がコラム k に配置されている。
$F_n(i)$	N	NMOS i が反転して配置されている。
$F_p(i)$	N	PMOS i が反転して配置されている。
E_r, E_l	2	ブロック内配置の右/左端の拡散が電源・接地端子に接続されていない。

異なるゲート入力信号を持つP/NMOSが同一コラム上に配置されてはならない: 本制約条件はPB制約として以下のように表現される。

$$C_n(i, k) + C_p(j, k) \leq 1, \quad 0 \leq k < W \quad (6)$$

式中の i, j の関係については以下の式が成り立つ。

$$0 \leq i < N, \quad 0 \leq j < N, \quad GATE_n(i) \neq GATE_p(j) \quad (7)$$

ここで、 $GATE_n(i)$, $GATE_p(j)$ はそれぞれNMOS i のゲート入力信号、PMOS j のゲート入力信号を表している。

異なる信号の拡散を向かい合わせているトランジスタは隣接するコラムに配置されない: N トランジスタに対する本制約条件は論理式として以下のように表現できる。

$$GAP_n(i, j) \wedge \bigvee_{k=0}^{W-2} (C_n(i, k) \wedge C_n(j, k+1)) = 0 \quad (8)$$

$$i \neq j, \quad 0 \leq i < N, \quad 0 \leq j < N$$

ここで、 $GAP_n(i, j)$ は $F_n(i)$ と $F_n(j)$ の関数であり、NMOS i がその右隣に配置されているNMOS j と拡散の共有ができないときに1となり、それ以外の時は0となる論理関数である。この制約条件式はCNFとして表現することが可能である。PMOSに対しても同様の制約条件をCNFとして表現する。

以上の制約条件を用いることでコラム数 W の可能なトランジスタ配置をすべて表現することが可能である。もしこれらの制約条件を満たす解が存在しない場合、与えられたトランジスタはコラム数 W で配置することができないことが保証される。従って、以下に示す手順によって最小幅のトランジスタ配置を生成することができる。

(1) ブロック内のトランジスタのネットリストを入力し、トランジスタ数を得る。初期幅 W をNMOSまたはPMOSの個数 N と等しくする。

(2) 幅 W での制約条件式を生成し、制約条件が充足可能であるかどうかを判定する。充足可能である場合は配置を生成し処理を終了する。充足可能でない場合はステップ3に進む。

(3) $W = W + 1$ として、ステップ2に戻る。

3.1.3 ブロック配置の充足可能性判定への定式化

すべてのブロック内のトランジスタ配置の終了後、それらのブロックを最小領域内に配置する。ブロック配置問題は上述のブロック内トランジスタ配置とほぼ同様の定式化によって充足可能性判定に定式化することができる。ブロック配置においては図5に示す様に、各ブロックは各コラムに最大1つ配置され、すべてのブロックは必ず一度ずつ現れる。拡散を共有すること

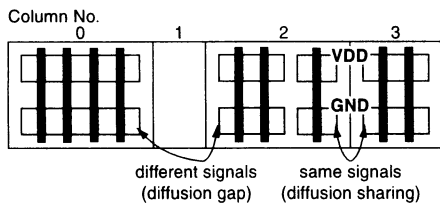


図5 ブロック配置の定式化

Fig. 5 Problem definition of the inter-block placement.

ができるブロックのみが隣接するコラムに配置され、ブロックの配置されていないコラムは拡散のギャップとなる。以上の制約を前節と同様に CNF および PB 制約として表現し、同様の手順によりコラム数を一つずつ増加させながら配置が可能かどうかを判定していくことで最小幅のブロック配置を得ることができる。

3.2 セル内配線

セル内配線については従来手法 [6] と同様の方法で充足可能性判定への定式化が可能である。セル内配線可能性の判定はトランジスタ配置の生成に比べて非常に短時間で処理が可能であるため階層化を行っていない。第2節で説明したように、本提案手法ではセル内配線可能性を向上させるため、トランジスタの配置位置およびポリシリコン層の使用に関して従来手法と異なる点がある。また、従来手法ではこれらの制約条件は CNF 形式の制約条件式のみとして表現されていたが、本手法では PB 制約も条件式として扱うため、条件式の表現法も異なっている。しかしながら、制約条件の設定の方法についてはほぼ同じ手順であり、本稿では詳細な説明を省略する。

3.3 セル生成フロー

最終的なセルレイアウト生成フローについて図6に示す。本セルレイアウト生成手法では回路分割後の各ブロックに対して、最小幅トランジスタ配置の生成後にブロック内の配線可能性の判定を行う。従って、ブロック配置の際にはすべて単体では配線可能なブロックが生成されていることになる。それらのブロックの最小幅配置を求めたあと再び全体の配線可能性の判定を行い、最終的に配線済みのセルレイアウトを生成する。配線可能性の判定箇所においては、配線不可能と判断されたトランジスタ配置を再び生成しないために、そのトランジスタ配置を制限する CNF による制約条件式を配置問題の制約に追加する。図中の Constraint Clause という項目はこの新たな制約条件式を追加すること示している。

4. 実験結果

以上で説明を行った階層的トランジスタ配置およびセル内配線手法を実装し、トランジスタレベルのネットリストからそれぞれの制約条件式を自動的に生成することを可能とした。今回の実験では、充足可能性判定を解くために PBS [10] という CNF と PB 制約を同時に扱うことのできる Boolean Solver を用いた。

4.1 トランジスタ配置

まず本手法を用いてトランジスタ配置を階層的に生成した場合と従来手法 [6] の比較について示す。トランジスタ配置のみ

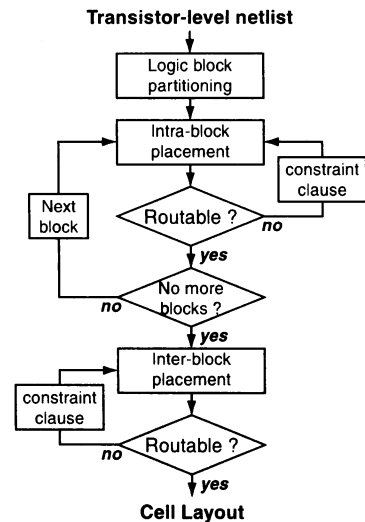


図6 本階層的セルレイアウト生成手法のフロー図

Fig. 6 Our hierarchical cell layout synthesis flow.

を求める場合は図6に示すフローの中での配線可能性の判定はすべて行わない。従ってブロック毎の配線可能性の判定も行わない場合の比較となる。表3に比較結果を示す。表は、実験に用いた回路名とそのトランジスタ数、生成されたトランジスタ配置のコラム数、処理時間を示している。表中の Conv. は従来手法 [6] を、Proposed は提案手法を示している。表には5種の回路のみの結果を示しているが実際には32種の回路について実験を行った。Total に表示されている値は32種の回路の総計である。今回例題として用いた32種の論理回路全てにおいて提案手法によるトランジスタ配置は階層化を用いない従来手法と比較して同じ幅、つまり最小幅の配置を生成した。処理時間については、トランジスタ数の少ない回路に対しては階層化の処理によりわずかに増加する場合もあるが、特に fad1(全加算器) などのある程度規模の大きな回路では処理時間を大幅に削減している。32種の回路の合計では処理時間は従来手法の約1%程度となり、高速に最小幅の配置を生成することができていることが示されている。

4.2 セルレイアウト生成

次にセル内配線まで行い配線済みのセルレイアウトを生成した結果について、商用のセルレイアウト生成ツール Progenesis 4.4 [11] および従来手法との比較を示す。商用ツールではコンパクションを行わず、シンボリックなレイアウトのみを生成した場合の結果を示している。表4に比較結果を示す。表は回路名、商用ツール・従来手法・提案手法それぞれのセル幅および処理時間について示し、表中の Comm. は商用ツール、Conv. は従来手法 [6]、Prop. は提案手法をそれぞれ示している。この表においても同様に5つの回路の結果について示しているが、実際には32種の回路について実験を行っており、Total はそれら全ての総計を表している。セル幅の値は配置の比較の際と同様にトランジスタ配置のコラム数を示している。Mux2 回路においては、配線まで行った際に従来手法よりも幅が1増加してい

表3 本手法と従来手法によるトランジスタ配置結果の比較

Table 3 Comparison results between our hierarchical transistor placement and the conventional method.

Cell name	#trans.	Width (#column)		Runtime (sec.)	
		Conv.	Proposed	Conv.	Proposed
aoi21	6	3	3	0.02	0.03
mux2	12	8	8	0.23	0.13
ao33	16	9	9	0.25	0.06
oa44	20	11	11	0.69	0.04
fad1	28	15	15	201.05	0.28
Total(32 circ.)	—	184	184	207.20	2.56
Ratio	—	1.000	1.000	1.000	0.012

る。これは階層化を用いた結果セル内配線を完了するためにブロック配置の際にコラム数が1つ増加したためである。しかし、32種の回路で従来手法よりも幅が増加した回路はmux2のみであった。これにより提案手法の階層化によってセル幅の最適性がほとんど損なわれないということが示された。本手法をトランジスタ数40の全加算器に適用し、生成されたレイアウトの例を図7に示す。このレイアウト生成にはおよそ1分の処理時間を必要とした。本階層化手法により従来手法では非常に多くの時間がかかってしまうような規模の大きな回路のレイアウトも短時間で生成可能であることが分かる。商用ツールとの比較から本手法によるセル幅はコラム数では約4%増加していることが分かる。これは、同じコラム上に上下に並んで配置されるP/Nトランジスタのゲート入力信号が共通でなければならぬという制約条件から、表中のmux2, ao33, oa44のように数個の回路において本手法の生成するセルレイアウトの幅が大きくなってしまったためである。しかしながら、本手法がセル生成に要する時間は商用ツールの約3%程度であり、わずかな面積の増加があるものの非常に高速なセルレイアウト生成を実現できていることが分かる。

5. まとめ

本稿ではCMOS論理セルレイアウトを高速に生成するためのセルレイアウトの階層的生成手法を提案した。今回の実験では、提案手法で用いている階層化手法および階層的トランジスタ配置手法により、32種のCMOS論理回路に対して従来手法によって生成された最小幅のトランジスタ配置と等しい幅の配置が生成された。トランジスタ配置に要する処理時間は従来手法と比較しておよそ1%程度となり大幅な高速化を実現できることを示した。また、提案するセルレイアウト生成手法と従来手法との比較により、1つの回路以外では従来手法による最小幅のレイアウトと等しい幅のレイアウトを高速に生成できることを示し、本階層化手法の有効性を示した。商用ツールとの比較により、本手法では生成されるセル幅が32種の回路の総計で4%程度大きくなってしまいが、セル生成に要する時間はおよそ3%となり高速なセルレイアウト生成を実現した。本セルレイアウト生成手法は新しいプロセスの開発に際していち早くセルライブラリを作成したい場合や、レイアウトレベルでの回路最適化において必要に際して新たなセルレイアウトを合成する場合などに特に有効な手法となると考えられる。

表4 本手法と従来手法および商用ツールによるレイアウト生成結果

Table 4 Comparison results with the commercial tool and the conventional method.

Cell name	Width (#column)			Runtime (sec.)		
	Comm.	Conv.	Prop.	Comm.	Conv.	Prop.
aoi21	3	3	3	18.78	0.04	0.06
mux2	6	8	9	17.16	6.42	1.45
ao33	9	10	10	28.12	392.26	2.57
oa44	11	12	12	18.80	116.41	4.60
fad1	15	15	15	73.06	305.76	1.24
Total(32)	182	188	189	833.31	1926.18	23.69
Ratio	1.000	1.033	1.038	1.000	2.311	0.028

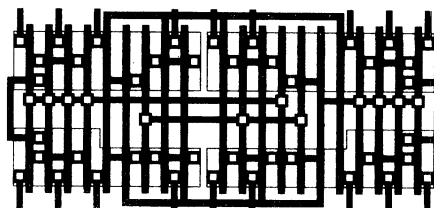


図7 本手法により生成された全加算器のレイアウト

Fig. 7 Layout of full adder generated by the proposed method.

謝 辞

本研究で御協力いただいた東京大学大規模集積システム設計教育研究センター (VDEC) 及び関係者の皆様に感謝致します。

文 献

- [1] H. Onodera, M. Hashimoto, and T. Hashimoto, "ASIC Design Methodology with On-Demand Library Generation," *IEEE Symp. VLSI Circuits Dig. of Tech. Papers*, pp. 57–60, 2001.
- [2] M. Guruswamy, R. L. Maziasz, D. Dulitz, S. Raman, V. Chiluvuri, A. Fernandez, and L. G. Jones, "CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries," in *Proc. ACM/IEEE 34th Design Automation Conference*, pp. 327–332, 1997.
- [3] A. Gupta and J. P. Hayes, "Width Minimization of Two-Dimensional CMOS Cells Using Integer Programming," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 660–667, 1996.
- [4] A. Gupta and J. P. Hayes, "CLIP: An Optimizing Layout Generator for Two-Dimensional CMOS Cells," in *Proc. ACM/IEEE 34th Design Automation Conference*, pp. 452–455, 1997.
- [5] R. L. Maziasz and J. P. Hayes, "Exact Width and Height Minimization of CMOS Cells," in *Proc. ACM/IEEE 28th Design Automation Conference*, pp. 487–493, 1991.
- [6] T. Iizuka, M. Ikeda, and K. Asada, "High Speed Layout Synthesis for Minimum-Width CMOS Logic Cells via Boolean Satisfiability," in *Proc. IEEE Asia South Pacific Design Automation Conf.*, pp. 149–154, 2004.
- [7] T. Uehara and W. M. vanCleemput, "Optimal Layout of CMOS Functional Arrays," *IEEE Trans. on Computers*, vol. C-30, no. 5, pp. 305–312, May 1981.
- [8] T. Sadakane, H. Nakao, and M. Terai, "A New Hierarchical Algorithm for Transistor Placement in CMOS Macro Cell Design," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 23.2.1–23.2.4, 1995.
- [9] A. Gupta, S. C. The, and J. P. Hayes, "XPRESS: A Cell Layout Generator with Integrated Transistor Folding," in *Proc. European Design & Test Conf.*, pp. 393–400, 1996.
- [10] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "Generic ILP versus Specialized 0-1 ILP: An Update," in *Proc. IEEE/ACM Int. Conf. on Computer Aided Design*, pp. 450–457, 2002.
- [11] *ProGenesis Guide*, Prolific, Inc., 2004.