

ゲートマトリックス方式に対するヒューリスティックアルゴリズムの性能評価

5X-3

中谷 公一 藤井 隆志 菊野 亨 吉田 典可  
 広島 大学

1. まえがき

CMOS論理回路をVLSIチップ上に実現するレイアウト設計方式の1つにゲートマトリックス方式<sup>[2]</sup>がある。この方式に対する研究として、p-ch, n-chのいずれか一方の面積の最小化を行うヒューリスティックアルゴリズムが提案されている。

筆者らも既に、p-ch, n-chの面積の和の最小化を行うヒューリスティックアルゴリズムを提案している<sup>[1]</sup>。本稿では、アルゴリズムの性能評価を行うため、Wingらのアルゴリズム<sup>[3]</sup>、及び、最適解との比較をする。

2. ゲートマトリックス方式

図1(a)に示すNORゲートの例を用いてゲートマトリックス方式<sup>[2]</sup>によるCMOS論理回路の実現方法を説明する。ゲートマトリックス方式による実現例を図1(b)に示す。マトリックスの列(ゲート線)はポリシリコンを用いて配線する。ゲート線の両側に拡散層を設けてMOSFETを形成する。このとき、同じ入力を持つMOSFETは同じゲート線上に、しかもp-MOSFETはp側(マトリックスの上半分)に、n-MOSFETはn側(下半分)に配置する。最後に、MOSFETのソース、ドレイン間をメタルを用いてマトリックスの行(トラック)上に配線する。

NORゲートの別の実現例を図1(c)に示す。この場合、MOSFET  $M_2$ と出力に対応するゲート線Zとの配線は、同図の点線で表された垂直拡散線を介して行う必要があり、図1(b)に比べて、p側でレイアウト面積が増加している。

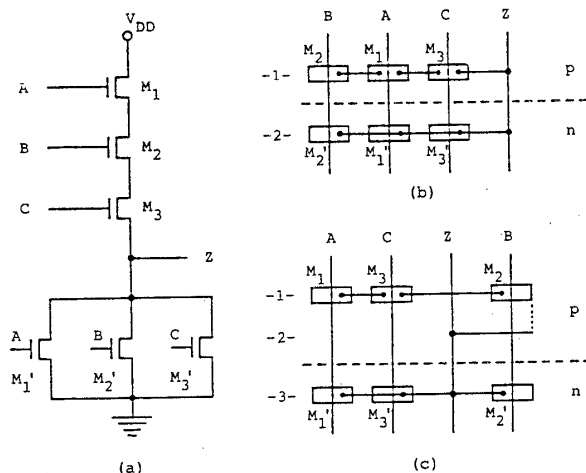


図1 ゲートマトリックス方式におけるNORゲート

3. レイアウト問題

与えられた論理回路はゲート線の集合  $S = \{g_i | 1 \leq i \leq |S|\}$  とゲートネットの集合  $N = \{n_i | 1 \leq i \leq |N|\}$  によって表される。ここで  $n_i$  は各論理ゲートごとに定義されるものとし、 $n_i = (X_i, y_i, z_i)$  と仮定する。但し、 $X_i$  は入力に対応するゲート線の集合、 $y_i$  は出力に対応するゲート線、 $z_i \in \{\text{NAND, NOR, NOT}\}$  は論理ゲートの種類をそれぞれ表す。 $S$  に対するゲート線の順序を  $P = (g_i(1), g_i(2), \dots, g_i(|S|))$  として定義する。

$S, N,$  及び、順序  $P$  が与えられたとき、MOSFETの直列接続に垂直拡散線が必要なゲートネットの集合を  $D(S, N | P) \subseteq N$  とする。このとき、関数  $V_P: D(S, N | P) \rightarrow \{\text{LEFT, RIGHT}\}$  を垂直拡散線の割り当てと呼ぶ。これは、 $n_i = (X_i, y_i, z_i) \in D(S, N | P)$ 、且つ、 $V_P(n_i) = \text{LEFT}(\text{RIGHT})$  ならば直列接続された  $X_i$  に属するMOSFETの中で最左端(最右端)に位置するMOSFETから垂直拡

散線を配線することを意味する。

ゲートマトリックス方式では、順序Pと割り当て $V_p$ が与えられるとレイアウトが一意に定まる。以上より、レイアウト問題GMLは次のように定義される。

[問題GML] ゲート線の集合Sとゲートネットの集合Nが与えられた時、トラック数が最小になるゲート線の順序Pと垂直拡散線の割り当て $V_p$ を求めよ。

問題GMLに対して筆者らは、p側、n側の両方の面積の和の最小化を行うヒューリスティックアルゴリズム(アルゴリズムGML)を提案している[1]。このアルゴリズムでは、SとNに基づいてグラフを構成し、その節点の順序を求めることによってゲート線の順序を決定する。

4. 適用例

論理回路 ( $S = \{g_i | 1 \leq i \leq 8\}$ ,  $N = \{n_i | 1 \leq i \leq 5\}$ ,  $n_1 = (\{g_2, g_4\}, g_3, \text{NAND})$ ,  $n_2 = (\{g_1, g_3, g_4\}, g_5, \text{NOR})$ ,  $n_3 = (\{g_2, g_4\}, g_8, \text{NAND})$ ,  $n_4 = (\{g_5\}, g_7, \text{NOT})$ ,  $n_5 = (\{g_5, g_8\}, g_6, \text{NOR})$ ) について考えてみる。対応する2つのレイアウト例を図2に示す。

図2(a)はWingらのアルゴリズム[3](アルゴリズムW)をn側に適用して求めたレ

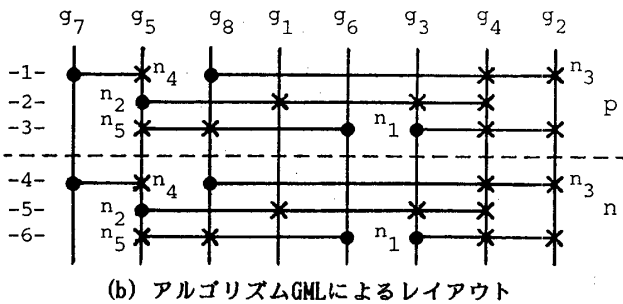
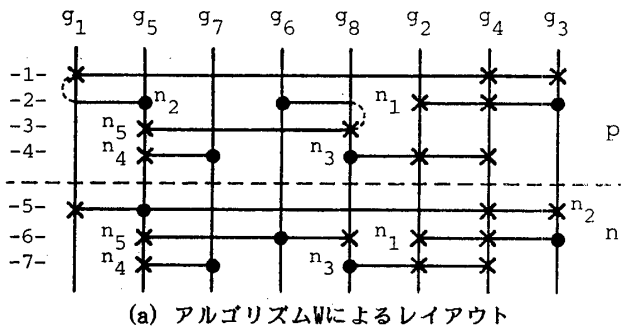


図2 レイアウトの比較

イアウト  $P1 = (g_1, g_5, g_7, g_6, g_8, g_2, g_4, g_3)$ ,  $V_{p1}(n_2) = \text{LEFT}$ ,  $V_{p1}(n_5) = \text{RIGHT}$  である。図2(b)はアルゴリズムGMLによるレイアウト  $P2 = (g_7, g_5, g_8, g_1, g_6, g_3, g_4, g_2)$ ,  $D(S, N | P2) = \phi$  である。図中の'X'と'●'はそれぞれMOSFETとコンタクトカットを表す。図2(b)では必要な水平トラック数が6となり、図2(a)に比べて少ない面積で回路を実現している。

5. 性能評価

アルゴリズムGMLの性能評価をシミュレーション実験によって行った。アルゴリズムは日本・データゼネラル社のECLIPSE MV/4000上でC言語を用いて実現した。実験には、実際に用いられている論理回路を含む50個のデータを用いた。

実験結果の一部を表1に示す。表中のアルゴリズムWによる解は、アルゴリズムWをp側、及び、n側に適用してみて、小さい場合の必要トラック数を選んだ。また、最適解は分枝限定法を用いて求めた。

表1より、アルゴリズムGMLは問題GMLに対する有効なヒューリスティック解法であると考えられる。

表1 実験結果

番号	データ		アルゴリズムGML		比較	
	S	N	CPU時間(sec.)	解(トラック数)	アルゴリズムW	最適解
1	8	5	0.34	6	7	6
2	12	10	1.08	10	10	10
3	13	7	1.51	10	12	10
4	15	11	3.28	20	21	18
5	20	17	5.29	10	12	10
6	29	25	18.58	20	21	18
7	37	28	40.88	22	22	-
8	42	36	54.39	24	27	-
9	48	40	88.72	29	30	-

文献

[1] 藤井他, "ゲートマトリックス方式におけるゲート線の順序決定アルゴリズム", 信学技報, CAS85-151, pp.25-32(1986).  
 [2] A.D.Lopez and H.-F.S.Law, "A dense gate matrix layout method for MOS VLSI", IEEE J. Solid-State Circuits, SC-15, 4, pp.736-740(1980).  
 [3] O.Wing, et al., "Gate matrix layout", IEEE Trans. on Computer-Aided Design, CAD-4,3, pp.220-231(1985).