論文 84-21-4-3

1차원 MOS-LSI 게이트 배열 알고리즘
(An Algorithm for One-Dimensional MOS-LSI Gate Array)

趙 仲 彙*, 鄭 正 和*
(Joung Hwee Cho and Jung Hwa Jung)

要 約

본 논문에서는 NAND 또는 NOR 게이트와 같은 기본 셀로 구성되는 1차원 MOS LSI의 칩 면적을 최소화하기 위한 레이아웃 알고리즘을 제안하고 있다. 배열하고자 하는 MOS 게이트들의 최좌측단과 최우측단에 입·출력 신호선을 표시하는 가상 게이트를 각각 설정하여 각 게이트 통과선 수를 최소화함으로써 수평 트랙 수를 최소로 하는 휴리스틱 알고리즘을 제안하고 실제의 논리회로를 택하여 프로그램 실험을 행함으로써 본 논문에서 제안한 알고리즘이 유용함을 보였다.

Abstract

This paper proposes a new layout algorithm in order to minimize chip area in one dimensional MOS - LSI composed of basic cells, such as NAND or NOR gates.

The virtual gates are constructed, which represent I/O of signal lines at the left-most and at the right-most side of the MCS gate array.

With this, a heuristic algorithm is realized that can minimize the number of straight connectors passing through each gate, and as the result, minimize the horizontal tracks necessary to route.

The usefulness of the algorithm proposed is shown by the execution of the experimental program on practical logic circuits.

I. 序 論

MOS-LSI 칩은 그림 1에 표시한 것과 같이 여러 종류의 기능함수(function)블록과 임의 논리 영역으로 구성된다. 기능함수 블록에는 RAM, ROM, 레지스터 및 입·출력 완충기 등이 있으며, 임의 논리 영역은 기능함수 블록과 입출력 관계를 갖는 변수로 구성되는 부울함수를 실현하기 위한 구분이다.

임의 논리 영역은 MOS 게이트의 1차원 배열로 구성되는데 일반적으로 전체 칩 면적의 1/4~1/3을 차지하게 되어 이 영역의 레이아웃 복잡도는 기능함수 블록의 복잡도에 비하여 매우 크게 된다. 따라서 이 영역의 레이아웃 자동화는 전체 칩을 레이아웃하는데 드는 비용과 시간을 크게 절감할 수 있다. 임의 논리 게이트의 레이아웃 디자인 방식은 [1]에서 1차원 MOS 게이트 배열 문제로 제시된 이후 많은 연구[2-4]가 행하여져 있다.

최근에 이 문제는 NP-완전문제로 밝혀졌으며[5] 그 이후 $O(n^5 \cdot m^2)$ (단, n은 게이트의 수, m은 신호선의 수이다.)[6]이 제안되어 있으나 칩면적을 최소화하는 게이트의 배열을 얻기 위한 알고리즘은 아직 제안되어 있지 않다. 따라서 본 논문에서는 배열하고자 하는 MOS 게이트의 최좌측단과 최우측단에 신호선의 입출력을 표시하는 가상적인 게이트(이하 가상 게이트라 부른다.)

*正會員, 漢陽大學校 工科大學 電子工學科
(Dept. Electron. Eng., Han Yang Univ.)
接受日字 : 1984年 1月 9日

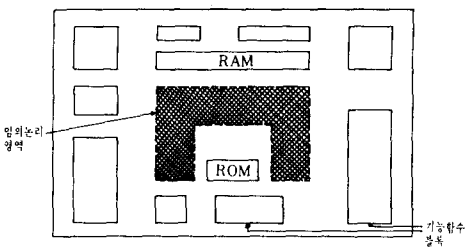

그림 1. MOS-LSI의 칩의 구성
Fig. 1. A chip schemetic of MOS-LSI.

g_l과 g_r을 각각 설정하여 이 게이트들을 관통하는 신호선수를 최소로 함으로써 수평 트랙수를 최소로 하는 능률 좋은 $O(n^2 \cdot m)$의 휴리스틱 알고리즘을 제안하고 실제의 예를 들어 비교 설명함으로써 본 논문에서 제안한 알고리즘의 유효성을 보인다.

Ⅱ. 예비 이론

그림 2에서는 NAND 게이트와 이를 MOS 트랜지스터로 구성한 회로도 및 레이아웃 구성도가 표시되어 있으며 복잡한 논리 함수는 내부적으로 연결된 게이트의 1차원 배열로 실현할 수 있다.

실제의 레이아웃에서 게이트 내부의 연결은 수직의 열을 따라 행하여지며 두 게이트사이의 연결은 수평의 행으로 만들어진다. 수평의 행으로 나타낸 신호선은 수직의 게이트 위에 단자를 지니는 것(그림2(c)의 A. C. D. P.)과 통과선을 지니는 것(그림2(c)의 B)으로 구분된다. 그림3과 같이 각 게이트의 폭(w_1)과 이웃하는 두 게이트사이의 간격(h)은 물리적인 제약에 따라 모두 일정하게 주어지므로 배열의 전체적인 폭(w_2)는 일정하다.

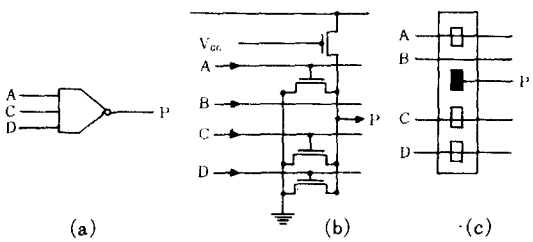

그림 2. (a) NAND 게이트 (b) 회로도 (c) 레이아웃 구성도
Fig. 2. (a) NAND gate. (b) Circuit schemetic. (c) Layout schematic.

따라서 칩면적을 최소화하는 문제는 각 신호선이 할당되어지는 수평 트랙 수를 최소로 하는 문제와 같게 된다. 그런데 그림3에 나타난 것과 같이 게이트 단자수는 게이트의 배열에 무관한 값을 가지며, 통과선 수는 게이트의 배열에 따라 변화하는 값을 가진다. 예로서 그림3 (a)의 게이트 배열에서는 총 통과선수가 14개이며 그림3 (b)에서는 0개이다. 따라서 수평 트랙수를 최소로 하는 문제는 게이트의 통과선 수를 최소로 함으로써 해결할 수 있다. 결국 칩 면적의 최소화하는 문제는 게이트의 통과선 수를 최소로 하는 문제로 귀착하게 된다.

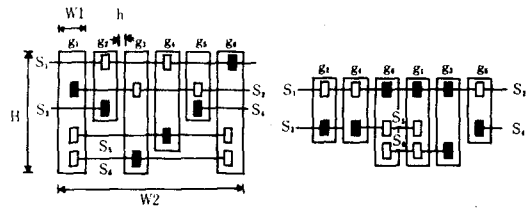

그림 3. 1차원 배열에서 게이트 배열의 예
Fig. 3. An example of gate order in a one-dimensional array.

본 논문에서 사용할 기호를 다음과 같이 정의한다.

1) $G=|g_l, g_1, g_2, \cdots\cdots g_m, g_r|$; 배열해야 할 게이트의 집합(단, g_l과 g_r은 가상 게이트이다.)

2) $S(g_i)$ $(1 \leq i \leq n)$; 게이트 g_i에 단자를 갖는 신호선들의 집합

3) $S=|S_1, S_2, \cdots\cdots, S_m|$; 수평 트랙에 할당되어야 할 신호선들의 집합

4) (S_j) $(1 \leq j \leq m)$; 신호선 S_j를 단자로 갖는 게이트의 집합

5) $g(L)$, $[g(R)]$; 배열의 좌측단(우측단)에 배열된 게이트의 집합

6) $S(L)$, $[S(R)]$; $g(L)$, $[g(R)]$에 속한 게이트에 단자를 갖는 신호선들의 집합

7) J_K; 신호선 S_K를 단자로 갖는 게이트의 갯수

8) $P_K(L)$, $[P_K(R)]$; 배열의 좌측단(우측단)에 있는 신호선 S_K의 단자수

문제의 본질을 보다 명확히 하기 위하여 좌측단(우측단)에 들어오는 입출력 신호선의 $g_l(g_r)$에의 할당은 회로 구성상 가장 인접한 곳에 행하는 것으로 한다.

Ⅲ. 알고리즘의 제안

초기 조건은 다음과 같다.

$S(L)=S(g_l)$, $S(R)=S(g_r)$, $g(L)=|g_l|$, $g(R)=|g_r|$

$$J_K=|G(S_K)|,\quad P_K(L)=\begin{cases}1; \text{ if } S_K \in S(L)\\ 0; \text{ otherwise}\end{cases}$$

$$P_K(R)=\begin{cases}1; \text{ if } S_K\in S(R)\\ 0; \text{ otherwise}\end{cases}$$

[단계 1.] G에 속하는 게이트가 1개 이하이면 단계 6으로 가고 그렇지 않을 경우 다음을 수행한다. 임의의 게이트를 g_i라 할 때, 모든 게이트에 대하여 $|S(g_i)\cap S(L)|$과 $|S(g_i)\cap S(R)|$을 각각 구한다.

[단계 2.] g_i에 대하여 좌측단에 할당 배열된 게이트와의 연결도 $C_i^{(1)}(L)$ 및 우측단에 이미 할당 배열된 게이트와의 연결도 $C_i^{(1)}(R)$을 각각 다음과 같이 정의하여 구한다.

$$C_i^{(1)}(L)=\frac{|S(g_i)\cap S(L)|}{|S(g_i)|}\qquad C_i^{(1)}(R)=\frac{|S(g_i)\cap S(R)|}{|S(g_i)|}$$

[단계 3.] g_i가 우측단에 할당 배열된 게이트에 무관하게 좌측단에 할당될 연결도 $C_i^{(2)}(L)$ 및 좌측단에 할당 배열된 게이트에 무관하게 우측단에 할당될 연결도 $C_i^{(2)}(R)$을 각각 다음과 같이 정의하여 구한다.

$C_i^{(2)}(L)=C_i^{(1)}(L)-C_i^{(1)}(R)$, $C_i^{(2)}(R)=-C_i^{(2)}(L)$

[단계 4.] $C_i^{(2)}(L)$이 가장 큰 값을 지니는 게이트를 g_i라 할 때, 이는 다른 게이트를 할당하는 것보다 통과선수가 적게 발생하게 되므로 g_i를 좌측단 배열의 최우측에 할당 배열할 게이트로 택한다. 만일 g_i가 2개 이상인 경우는 다음과 같은 경우1)과 경우2)로 구분된다.

경우1) $C_j^{(2)}(L)=\frac{ky}{kx}$ $C_i^{(2)}(L)=\frac{y}{x}$ (단, $k=2,3,\cdots$)

$|S(g_i)|>|S(g_j)|$이므로 할당할 게이트의 통과선 수가 적은 게이트 g_i를 할당 배열할 게이트로 택한다.

경우2) $C_j^{(2)}(L)=C_i^{(2)}(L)=b/a$일 때는 게이트 g_i, g_j에 대하여 통과선 수가 같게 발생하나 각 게이트의 신호선에 대하여 아직 할당되지 않은 게이트가 많은 것이 영역내에서 할당되는 경우가 통과선 수가 적게 발생할 것이다.

경우1)과 경우2)를 만족하는 선택함수는

$$Pg_i(L)=\sum_{\substack{K\in S(g_i)\\ P_K(R)=0}}\frac{J_K-P_K(L)+1}{J_K-P_K(L)}-\sum_{\substack{K\in S(g_j)\\ P_K(R)\neq 0}}\frac{J_K-P_K(R)+1}{J_K-P_K(R)}$$

이 되므로

$Pg_i(L)$이 가장 큰 g_i를 택하며, 이값 또한 같으면 임의로 택하여 좌측단 배열 $g(L)$의 최우측에 할당한 뒤 다음을 수행하고 [단계 5]로 간다.

$G\leftarrow G-\{g_i\}$, $S(L)\leftarrow S(L)\cup S(g_i)$

$P_K(L)\leftarrow P_K(L)+1$ (단, $K\in S(g_i)$)

[단계 5.] 단계 3에서 구한 $C_i^{(2)}(R)$이 가장 큰 값을 지니는 게이트를 g_j라 할 때 이는 다른 게이트를 할당하는 것보다 통과선 수가 적게 발생하게 되므로 g_j를 우측단 배열의 최좌측에 할당 배열할 게이트로 택한다.

만일, g_j가 2개 이상인 경우는 단계 4에서 서술한 경우1)과 경우2)로 구분된다. 이를 만족하는 선택함수는

$$Pg_j(R)=\sum_{\substack{K\in S(g_j)\\ P_K(L)=0}}\frac{J_K-P_K(R)+1}{J_K-P_K(R)}-\sum_{\substack{K\in S(g_j)\\ P_K(L)\neq 0}}\frac{J_K-P_K(L)+1}{J_K-P_K(L)}$$

이 되므로

$Pg_j(R)$이 가장 큰 g_j를 택하며, 이 값 또한 같으면 임의로 택하여 우측단 배열 $g(R)$의 최좌측에 할당 배열한 뒤 다음을 수행하고 단계 1로 간다.

$G\leftarrow G-\{g_j\}$, $S(R)\leftarrow S(R)\cup S(g_j)$

$P_K(R)\leftarrow P_K(R)+1$ (단, $K\in S(g_j)$)

[단계 6.] G에 속하는 게이트가 1개이면 이를 g_i라 하고 g_i를 좌측단 배열 $g(L)$의 최우측에 할당한다. 그렇지 않은 경우 단계 7로 간다.

[단계 7.] 우리가 원하는 게이트 열을 배열함으로써 G(S)를 얻고 프로그램 수행을 마친다.

1차원 MOS 게이트의 배열 G(S)를 얻은 후 각 신호선을 수평 트랙에 할당하여야 한다. 그런데, 본 논문의 목적은 수평 트랙 수를 최소화하여 칩 면적을 줄이는데 있으므로 각 신호선을 수평 트랙에 할당하는 문제는 [7]에서 제안한 '좌측 우선 할당 알고리즘'을 사용함으로써 쉽게 해결할 수 있다. 다음에 본 논문에서 제안한 알고리즘의 최대 시간 복잡도를 계산한다.

n과 m을 각각 게이트 수 및 신호선수라 할 때 단계 1의 시간 복잡도는 $O(n\cdot m)$이며 단계 2는 $O(n)$ 이고 단계 3 또한 $O(n)$이다. 단계 4에서 $C_i^{(2)}(L)$ 및 $Pg_i(L)$을 계산하기 위해서는 $O(n\cdot m)$이 필요하므로 단계 4의 시간 복잡도는 $O(n\cdot m)$이다. 단계 5는 단계 1로의 $O(n)$의 순환을 제외하고는 단계 4와 같이 $O(n\cdot m)$이다. 단계 6과 단계 7은 단순한 작동이므로 O(1)이다.

[예제][2]

그림 4는 NOR 게이트를 사용한 1비트 가산기-누적기의 논리 구성도이다. 이 논리 회로를 MOS-LSI

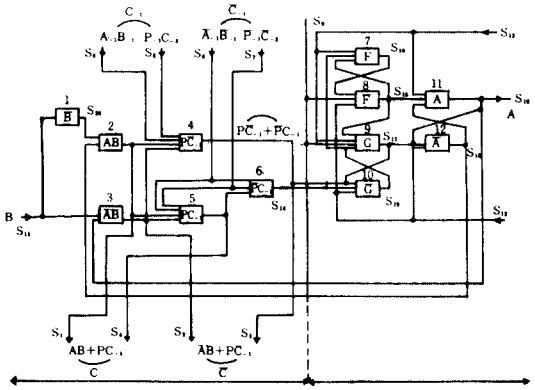

그림 4. 1비트 가산기-누적기의 논리도

Fig. 4. Logic diagram of one bit adder-accumulator.

로 구성하는 경우 좌측단에 할당되는 입출력선은 S_1~S_8이며 우측단에 할당되는 입출력선은 S_9~S_{13}이다.

이 논리 회로를 알고리즘에서 제안한 용어로 표시하면 다음과 같다.

$S(g_1)=\{S_{11}, S_{20}\}$ $S(g_2)=\{S_1, S_{18}, S_{20}\}$

$S(g_3)=\{S_2, S_{16}, S_{11}\}$ $S(g_4)=\{S_1, S_2, S_3, S_5, S_8\}$

$S(g_5)=\{S_1, S_2, S_4, S_8, S_7\}$ $S(g_6)=\{S_4, S_8, S_7, S_{14}\}$

$S(g_7)=\{S_{13}, S_{15}, S_{16}, S_{19}\}$ $S(g_8)=\{S_9, S_{12}, S_{15}, S_{16}\}$

$S(g_9)=\{S_9, S_{13}, S_{15}, S_{17}, S_{19}\}$ $S(g_{10})=\{S_3, S_{12}, S_{14}, S_{17}, S_{19}\}$

$S(g_{11})=\{S_{10}, S_{13}, S_{15}, S_{18}\}$ $S(g_{12})=\{S_{10}, S_{12}, S_{17}, S_{18}\}$

$G(S_1)=\{g_2, g_3, g_4, g_5\}$ $G(S_2)=\{g_4, g_5, g_6, g_5\}$

$G(S_3)=\{g_4, g_6, g_{10}\}$ $G(S_4)=\{g_5, g_6, g_6\}$

$G(S_5)=\{g_4, g_6\}$ $G(S_6)=\{g_2, g_5, g_6\}$

$G(S_7)=\{g_5, g_5, g_6\}$ $G(S_8)=\{g_5, g_6\}$

$G(S_9)=\{g_8, g_9, g_r\}$ $G(S_{10})=\{g_3, g_{11}, g_{12}, g_r\}$

$G(S_{11})=\{g_1, g_3, g_r\}$ $G(S_{12})=\{g_8, g_{10}, g_{12}, g_r\}$

$G(S_{13})=\{g_7, g_9, g_{11}, g_r\}$ $G(S_{14})=\{g_6, g_{10}\}$

$G(S_{15})=\{g_7, g_8, g_9, g_{11}\}$ $G(S_{16})=\{g_7, g_8\}$

$G(S_{17})=\{g_9, g_{10}, g_{12}\}$ $G(S_{18})=\{g_2, g_{11}, g_{12}\}$

$G(S_{19})=\{g_7, g_9, g_{10}\}$ $G(S_{20})=\{g_1, g_2\}$

본 논문에서 제안한 알고리즘을 사용하여 G(S)를 얻으면 $G(S)=\{g_4, g_3, g_5, g_6, g_2, g_1, g_3, g_{10}, g_{12}, g_{11}, g_9, g_7, g_8, g_r\}$이며 G(S)에 대하여 각 신호선을 수평 트랙에 할당하는 경우에 종래의 알고리즘을 사용하는 경우는 그림 6과 같으며 본 논문의 알고리즘을 사용하면 그림 5와 같이 되어서 칩 면적이 줄어듬을 알 수 있다.

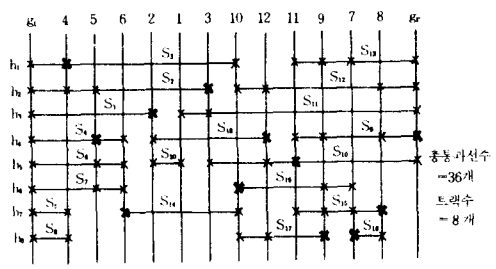

그림 5. 제안된 알고리즘에 의한 레이아웃
Fig. 5. Layout by proposed algorithm.

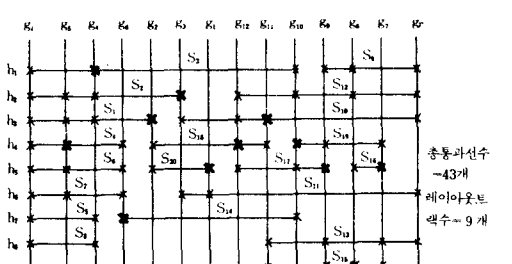

그림 6. 종래의 방법에 의한 레이아웃
Fig. 6. Layout by conventional method.

Ⅳ. 結 論

본 논문에서는 MOS-LSI 레이아웃 방식중의 하나인 1차원 MOS 배열 레이아웃 방식의 면적을 최소화하는 문제의 해를 구하는 방법을 제안했다.

배열하고자 하는 MOS 게이트의 좌측단과 우측단에 입·출력 신호선을 표시하는 가상 게이트 g_l과 g_r를 각각 설정하여 각 게이트의 통과선 수를 최소로 함으로써 수평트랙수를 최소로 하는 능률좋은 $O(n^2 \cdot m)$의 휴리스틱 알고리즘을 제안하였고 실제의 논리회로를 택하여 프로그램 실험을 행함으로써 본 논문에서 제안한 알고리즘의 유효성을 확인하였다.

따라서, 본 논문에서 제안한 알고리즘을 사용하여 MOS-LSI를 자동 설계하는 경우 칩 면적을 줄일 수 있을 뿐 아니라 설계시간을 줄임으로써 설계 비용을 줄일 수 있을 것으로 기대된다.

參 考 文 獻

[1] A. Weinberger, "Large scale integration of MOS complex logic: A layout method," *IEEE J. Solid-State Circuits,* vol. SC-2, no.4, pp. 182-190, 1967.

[2] R.P. Larsen, "Computer-aided preliminary Layout Design of customized MOS array", *IEEE Trans. Computers,* vol. C-20, no.5 pp. 512-523, 1971.

[3] H. Yoshizawa, H. Kawanishi and K. Kani, *A Heuristic Procedure for Ordering MOS Arrays.* Proc. 12th Design Automation Workshop, pp. 384-389, 1975.

[4] T. Ohtsuki, H. Mori and E.S. Kuh, "One-dimensional logic gate assignment and interval graphs", *IEEE Trans. Circuits and Systems,* vol. CAS-26, no. 9, pp. 675-684, 1979.

[5] T. Kashiwara and T. Fujisawa, *Np-Completeness of the Problem of Finding a Minimum-Clique Number Interval Graph Containing a Given Graph as a Subgraph.* Proc. IEEE Inter. Symp. Circuits and Systems, pp. 657-660, 1979.

[6] I. Shirakawa, N. Okuda and T. Harada, "A layout system for the random logic portion of MOS LSI", *IEEE Trans. Circuits and Systems,* vol. CAS-30, no. 8, pp. 572-581, 1981.

[7] A. Hashimoto and J. Stevens, *Wire Routing by Optimizing Channel Assignment within Large Apetures.* Proc. 8th Design Automation Workshop, pp. 155-169, 1971. *