

LSI의 레이아웃 CAD에 있어서 2-獨立 經路 問題 (Two-Disjoint Path Problem in LSI Layout CAD)

鄭 正 和*
(Jung Hwa Jung)

要 約

LSI의 레이아웃 CAD 시스템에 있어서 未結線 信號線에 대한 配線可能性을 判定하고, 그것을 修正·解決하는 것은 중요하다.

本 論文에서는 無向 그래프 $G = (V, E)$ 를 導入하여 2개의 절점 독립(vertex-disjoint) 경로의 存在 여부를 判定하는 알고리즘을 제안하고, 경로가 존재하는 경우는 미결선 신호선을 해결하고, 존재하지 않는 경우는 해결할 수 없음을 보이는 방법을 제시하였다. 제안한 알고리즘의 시간 복잡도는 $O(|V| \times |E|)$ 이다.

Abstract

A method finding out routability for unrouted signal lines and rerouting those which are turned out to be able to route in layout design of LSI, is described.

In this paper the problems of finding two-disjoint paths represented by an undirected graph $G = (V, E)$, where V, E are sets of vertices and edges respectively, are studied.

The existence of two-disjoint paths from s_1 to t_1 (called P_1) and from s_2 to t_2 (called P_2) indicated by the four vertices on the graph $s_1, t_1, s_2, t_2 \in V$ means that two distinct signal lines exist in layout design.

It turns out that the proposed time complexity in the algorithm is $O(|V| \times |E|)$.

I. 序 論

현재 LSI의 레이아웃 CAD 시스템에 있어서 100% 配線 가능한 自動 배선기의 개발이 요구되고 있다. 自動 배선기에 내장된 配線 알고리즘은 LSI의 종류에 따라 다양하나, 일반적으로 線分探索法, 채널 割当法, 미로法 등으로 大別할 수 있다. 이들은 단독 또는 둘 이상이 組合되어 사용되고 있으며, 초기 단계에서는

配線처리가 빠른 線分探索法 및 채널 割当法이 사용되며, 마지막 단계에 미로法이 사용되고 있다.^[6]

미로法은 경로가 있으면 반드시 찾아 내는 뛰어난 탐색능력을 가지고 있으며, 따라서 알고리즘의 적용에도 불구하고 配線이 불가능한 신호선은 경로가 없음을 의미한다. 이 배선 불가능한 신호선을 未結線 信號線이라 부른다.

현재 未結線 信號線은 인터랙티브 시스템을 도입한, 즉 인간이 介入하여 해결하고 있으나 많은 시간과 노력을 필요로 하고 있다.^[6]

本 論文에서는, 人間の 介入없이 未結線 信號線을 해결하는 方法을 제안한다.

無向 그래프 $G = (V, E)$ (V ; 절점들의 집합, E ;

* 正會員, 漢陽大學校 工科大學 電子工學科
(Dept. of Elec. Eng., Han-Yang Univ.)
接受日字: 1982年 9月 1日

가지들의 집합), $s_1, t_1, s_2, t_2 \in V$ 에 있어서 s_1 과 t_1 을 연결하는 경로를 P_1 , s_2 와 t_2 를 연결하는 경로를 P_2 라 한다. 2개의 절점 독립(vertex-disjoint) 경로 P_1, P_2 를 2개의 신호선이라 가정하여 G 위에 P_1, P_2 의 존재 여부를 판정하면, 이것이 2개의 신호선의 배선 가능성을 판정하는 것과 같다.

$(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ 인 K 쌍의 ($K \geq 3$) 각 절점 사이에 존재하는 K 개의 절점 독립 경로를 찾아내는 문제는 NP완전문제^[1]이다. 평면성 그래프 또는 코오달(chordal) 그래프에 있어 2-독립 경로 문제(이하 TPP라 부르며, 本文에서의 독립은 절점 독립을 의미함)은 $O(|E|)$ 로 解를 구하는 방법이 제안되어 있으며,^[2] 최근에는 3-연결이며 비평면성인 無向 그래프에 있어 $O(|V| \times |E|)$ 인 解法이 제안되어 있다.^[3]

本文에서는 2개의 경로를 가정하여 그 사이에 존재하는 교차 경로를 찾기 위해 브리지의 개념을 도입한다. 다음에, 제안한 알고리즘의 복잡도를 산출하여 $O(|V| \times |E|)$ 임을 보인다. 마지막으로, LSI의 레이아웃 설계에 있어서 未結線 信號線이 存在하는 實際의 경우를 例로 들어, 제안한 알고리즘을 적용하여 판정 해결하는 方法을 제시한다.

2. 알고리즘

TPP가 “진”인 경우는 2개의 독립 경로가 존재하는 것을 나타내며 “위”인 경우는 2개의 독립 경로가 존재하지 않는 것을 나타내는 것으로 하며 P 를 이루게 되는 절점들의 집합을 P 로, 가지들의 집합을 $E(P)$ 로 나타낸다. 또한, $P(s, t)$ 는 소오스 s 와 싱크 t 로 구성되는 경로를 나타내며 P 에 속하는 2개의 절점 u 와 v 사이의 부분경로는 $P[u;v]$ 로 나타낸다.

$G_0 = (V_0, E_0)$ 가 2-연결인 무향 그래프로 주어질 때 다음과 같이 알고리즘을 구성한다.

단계 1 ; 먼저 2개의 독립경로 $P(s_1, t_1)$ 와 $Q(s_2, t_2)$ 을 찾는다. $P(s_1, t_1)$ 와 $Q(s_2, t_2)$ 을 찾을 수 없는 경우는 TPP가 “진” 또는 “위”임을 밝힌다.

단계 2 ; PUQ 에 대한 모든 브리지의 집합 $B = \{B_1, B_2, \dots\}$ 를 찾는다. 각각의 브리지 B_i 는 다음과 같이 분류할 수 있다. ① $X, Y \in PUQ$ 에 대하여 $e = (X, Y) \in E_0 - E(P) - E(Q)$ 인 1개의 가지이다.

② $X \in V_0 - P - Q$ 인 절점 X 와, $P(s_1, t_1)$ 또는 $Q(s_2, t_2)$ 위에 존재하는 절점 Y 로 구성되는 경로 $R(X, Y)$ 를 갖는 최대 부분 그래프 $G = (V, E_0 - E(P) - E(Q))$ 이다. (여기서, $X, Y \in V$ 이다.)

브리지의 예는 그림 1에 나타냈다.

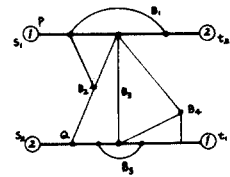


그림 1. 브리지의 예
Fig. 1. Example of bridges.

브리지 B 의 점점을 PUQ 에 속하는 B 의 점점으로 정의하며 B 의 모든 점점이 $P(s_1, t_2)$ 위에 있는 브리지를 B^P , $Q(s_2, t_1)$ 위에 있는 것을 B^Q , $P(s_1, t_2)$ 와 $Q(s_2, t_1)$ 위에 동시에 존재하는 것을 B^{PQ} 로 나타낸다.

단계 3 ; B^P 와 B^Q 에 속하는 모든 브리지를 제거한다. $P(s_1, t_2), Q(s_2, t_1)$ 위에 있는 두개의 다른 임의의 점점중 s_1 과 t_2 (s_2 와 t_1)에 가까운 점점을 각각 a 와 b (c 와 d)라 할때 B_1 이 $P_1(a, b)$ 를 포함하고, B_2 가 $P_2(a, c)$ 를 포함하는 브리지의 쌍(B_1, B_2)를 교차쌍이라 정의한다.

이 예는 그림 2와 같다.

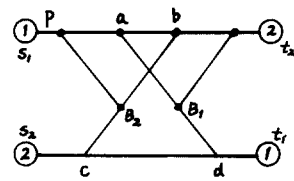


그림 2. 교차쌍의 예
Fig. 2. Example of alternating pair.

단계 4 ; 브리지의 교차쌍이 존재하면 TPP는 “진”이고 존재하지 않으면 다음 단계로 넘어간다.

단계 5 ; $P(s_1, t_2)$ 와 $Q(s_2, t_1)$ 의 각각에 2개 또는 그 이상의 점점을 갖는 브리지의 집합 B 를 찾아내며, 앞에 서술한 점점에 대한 조건을 만족하지 않는 브리지는 모두 버린다. 만일, B 에 속하는 브리지가 하나도 없다면 TPP는 “위”이다.

$B_i \in B$ 인 각각의 브리지에서 s_1 과 t_2 (s_2 와 t_1)에 가장 가까운 $P(s_1, t_2)$ ($Q(s_2, t_1)$) 위의 점점을 각각 a 와 b (c 와 d)라 할때 $BUP[a; b]UQ[c; d]$ 의 가지 및 절점들로 나타내지는 $G_0 = (V_0, E_0)$ 의 영역 그래프를 $G_0(B)$ 로 정의한다.

이 예는 그림 3과 같다.

2진 재귀함수 $f = (G_0(B), P(s_1, t_2), Q(s_2, t_1))$ 을 다음과 같이 정의한다. $G_0(B)$ 는 2개의 독립

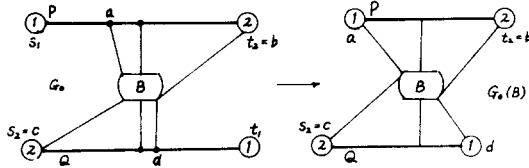


그림 3. 영역 그래프의 예
Fig. 3. Example of section graph.

경로 $P(s_1, t_2)$, $Q(s_2, t_1)$ 과 점점 s_1, t_1, s_2, t_2 를 포함하는 PUQ에 대한 영역 그래프이다. 2개의 독립한 경로 $P_1(s_1, t_1)$ 과 $P_2(s_2, t_2)$ 가 존재하면 f 는 “정”이며 존재하지 않으면 “오”이다.

위에서 정의의 한대로 브리지 B_i 에 대하여 2진 재귀 함수 $f(G_0(B_i), P[a; b], Q[c; d])$ 를 찾아내고 B 를 $B - |B_i|$ 로 한다. 적어도 하나의 $G_0(B)$ 에 대하여 $f = \text{“정”}$ 이면 TPP는 “진”이고 모든 $G_0(B)$ 에 대하여 $f = \text{“오”}$ 이면 TPP는 “위”이다.

다음 단계(단계 6~12)를 f 에 대하여 수행한다.

단계 6: $P(s_1, t_2)$ 와 $Q(s_2, t_1)$ 에 독립한 경로 $P'(s_1, t_2)$ 를 찾는다. $P'(s_1, t_2)$ 는 단계 5에서의 브리지 성질로부터 분명히 존재한다.

단계 7: 단계 2에서와 같은 과정을 $PUQU P'$ 에 적용하여 모든 브리지 $B = \{B_1, B_2, \dots\}$ 를 찾는다.

단계 8: 하나의 점점 a 가 $P - |s_1, t_2|$ 에 또 다른 점점 b 가 Q 에 속하는 브리지 B 를 가지면 f 는 “정”이나, 그렇지 않으면 다음 단계로 넘어간다. 이와 같은 브리지 B 가 존재하면 $P' - |s_1, t_2|$ 에 속하는 하나의 점점을 갖고 G 가 2-연결성이라는 가정에 의해 TPP는 “진”임이 증명된다.

이 예는 그림 4 와 같다.

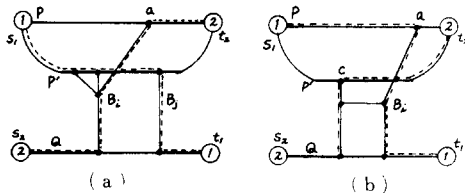


그림 4. 경로의 유도
Fig. 4. Extraction of two paths.

단계 9: 각각의 브리지에 대하여 모든 점점이 $P'(Q)$ 에 속하면 B^1 으로, PUQ 에 속하면 B^0 로 나타낸다.

B^1 에서 교차쌍이 존재하면 f 는 “정”이나 존재하지 않으면 다음 단계로 넘어간다.

단계 10: $P'(s_1, t_2)$ 와 $Q(s_2, t_1)$ 의 각각에 2개 또

는 그 이상의 점점을 갖는 각 브리지에 대하여 s_1 과 t_2 (s_2 와 t_1)에 가장 가까운 $P'(s_1, t_2)$ ($Q(s_2, t_1)$) 위에 점점을 a 와 b (c 와 d)라 할때 $f(G(B), P'[a; b], Q[c; d])$ 가 “정”이면 TPP는 “진”이고 “오”이거나 위의 성질을 만족하는 브리지가 존재하지 않으면 다음 단계로 넘어간다.

단계 11: $P'(s_1, t_2)$ 위에서 s_1 과 t_2 에 가까우며 s_1, t_2 와는 다른 B^1 의 점점을 x 와 y 라 할때 B_i 가 $P'[x; t_2] - |x, t_2|$ 위에 점점 u 를 B_i 가 $P'[s_1; y] - |s_1, y|$ 위에 점점 v 를 각각 갖고 $B_i, B_j \in B^1$ 인 교차쌍이 존재하면 f 는 “정”이나 존재하지 않으면 다음 단계로 넘어간다.

이 예는 그림 5 와 같다.

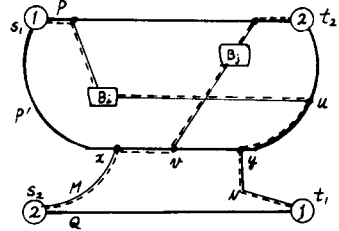


그림 5. 단계 11에서 2개 경로의 유도
Fig. 5. Extraction of two paths in step 11.

단계 12: 단계 11과 같은 방법으로 x 와 y 를 $P'[x; t_2] - |x, t_2|$ 와 $P'[s_1; y] - |s_1, y|$ 에 점점을 갖는 B^1 의 각 브리지에 대하여 다음을 수행하나 위의 성질을 만족하는 브리지가 없으면 f 는 “오”이다.

브리지 B 가 $P'(s_1, x)$ 에 점점을 가지면 s_1 을 s_1^* 로, x 를 x^* 로 하며 갖지 않으면 $s_1(x)$ 에 가장 가까운 $P(s_1, t_2)$ ($P'(s_1, t_2)$ 위의 점점을 $s_1^*(x^*)$ 라 한다.

s_1^*, x^* 를 제외한 부분경로 $P[s_1; s_1^*]$ 와 $P'[s_1; x^*]$ 을 제거함으로써 브리지 B 를 변경시킨다. $P'[y; t_2]$ 에 대해서도 위의 과정을 반복하여 s_1^* 와 x^* 에 각각 해당하는 t_2^* 와 y^* 를 정한다. $P[s_1^*; t_2^*]$ 와 $P'[x^*; y^*]$ 그리고 브리지 B 의 절점들에 의하여 결정되는 영역 그래프 G' 에 대하여 $f(G'(B) [s_1^*; t_2^*], P'[x^*; y^*])$ 이

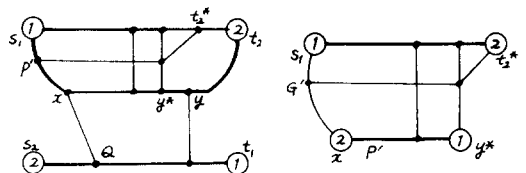


그림 6. 영역 그래프 G'
Fig. 6. Section graph G' .

“정” 또는 “오”를 정하여 TPP가 “진” 또는 “위”임을 정한다.

이 예는 그림 6과 같다.

3. 알고리즘의 복잡도

알고리즘은 적어도 한번은 진행하게 되는 단계 1~5의 메인루틴과 2진 재귀 함수 f 에 대한 진행으로 크게 나눌 수 있다.

복잡도가 선형임을 $O(|V| + |E|)$ 라 할때 단계 1은 플로워 방법^[7]을 사용하는데 단 2개의 경로만이 필요하므로 선형이다. 단계 2는 그래프 탐색방법을 사용하므로 선형이며, 단계 3은 단지 $O(1)$ 의 진행이다.

본 논문에서 사용한 브리지의 의미는 사이클대신 몇개의 경로에 대하여 정의된 점을 제외하고는 평면성 조사 알고리즘^[9, 10]에서 사용한 의미와 같다. 따라서 단계 4에서 교차쌍을 찾는것은 선형 평면성 조사 알고리즘^[11]의 일부이므로 선형이다. 단계 6은 f 의 재귀적 부름을 제외하고는 선형 2-연결도 조사 알고리즘^[8]을 사용하므로 선형이며 단계 6에서 경로 $P'(s_1, t_2)$ 를 찾는 것도 그래프 탐색 방법^[8]을 사용하므로 선형이다. 단계 7~12는 단계 1~5의 단순한 변형이므로 선형이다.

f 의 재귀적 부름은 $G = (V, E)$ 의 가지들 또는 절점들만을 생각하는데 이유는 다른 가지들에 관계되는 동작은 상수적으로 제한되기 때문이며 단계 10 또는 단계 12에서 유도되는 영역 그래프 $G' = (V', E')$ 에, $G = (V, E)$ 의 가지들 또는 절점들이 포함된다면 $|V'| < |V|$ 이므로 특별한 가지 또는 절점을 포함하는 영역 그래프의 수는 $O(|V|)$ 이다.

따라서, 본 논문에서 제안한 알고리즘의 복잡도는 $O(|V| \cdot |E|)$ 임을 알 수 있다.

4. 레이아웃에의 응용

LSI 레이아웃 설계를 컴퓨터에 의해 자동적으로 배선을 하는 경우 사용되는 자동 배선기는 100%의 완전 배선이 거의 불가능하게 된다. 이는 미로법 배선 알고리즘을 이용한 것으로 이 알고리즘은 신호선들을 차례로 이미 배선한것과 내부적인 접속이 없을때까지 계속 연결시킨 다음 남게 되는 미결선 신호선은 설계자가 오랜 경험을 통하여 얻은 지식으로 배선하게 된다. 따라서, 이는 많은 시간이 필요하며 잘못이 생길 우려가 많다.

본 논문에서는 미로법 배선 알고리즘을 사용하여 K 개의 신호선 N_1, N_2, \dots, N_K 가 배선되고 $(K + 1)$ 번째 신호선 N_{K+1} 이 배선될 수 없는 경우가 발생하면이

는 신호선 N_{K+1} 의 핀 s 와 t 를 분리시키는 “장벽”이 존재함을 의미한다. 그래프 이론적으로 “장벽”이라 함은 s 와 t 의 분리요소라 불리우는 최소의 집합 B 로 s 와 t 사이의 모든 경로는 B 에 있는 절점을 최소한 하나는 포함해야 한다.

본 알고리즘은 이미 배선된 신호선 $\{N_1, N_2, \dots, N_K\}$ 중 단 하나만을 다시 배선시킬 때 N_{K+1} 의 배선 가능성을 결정하는 것으로 $N_i \in \{N_1, N_2, \dots, N_K\}$ 인 쌍 (N_i, N_{K+1}) 에 대하여 적용시키며 이때 $(K-1)$ 개의 신호선 $\{N_1, \dots, N_{i-1}, N_{i+1}, \dots, N_K\}$ 에 해당하는 모든 절점들은 사용하지 않는다. 이때 TPP가 “진”이면 N_i 의 배선 경로를 수정한뒤 N_{K+1} 를 배선하며 모든 N_i 에 대하여 “위”이면 N_{K+1} 를 $N_1 \sim N_K$ 의 신호선과 내부적 접속이 없도록 배선할 수는 없게 된다.

(예) 그림 7에 보이는 바와 같이 동그라미 안에 있는 숫자 i 는 신호선 N_i 의 핀을 의미하는 것으로 신호선 $N_1 \sim N_5$ 는 미로법 배선 알고리즘을 적용하여 배선할 수 있다. 그러나 미로법 배선 알고리즘으로 N_6 은 $N_1 \sim N_5$ 와 접속이 없도록 배선할 수 없으므로 $N_1 \sim N_5$ 중 일부의 배선 경로가 수정되어야 한다. N_6 을 배선하기 위한 그래프 탐색 알고리즘이 그림 7(b)에 있는 것과 같이 X로 표시된 6개의 절점들로 구성되는 장벽을 결정하게 된다. 처음에 (N_4, N_6) 에 대하여 본 알고리즘을 적용하여 “위”임을 얻는다. 다음에 (N_5, N_6) 에 본 알고리즘을 적용하여 TPP가 “진”임을 얻는다. 따라서 N_5 의 배선 경로를 수정한뒤 N_6 을 배선하면 그림 7(c)에 보이는 바와 같게 배선됨을 알 수 있다.

$N_1 \sim N_6$ 의 완전한 배선은 그림 7(d)와 같다.

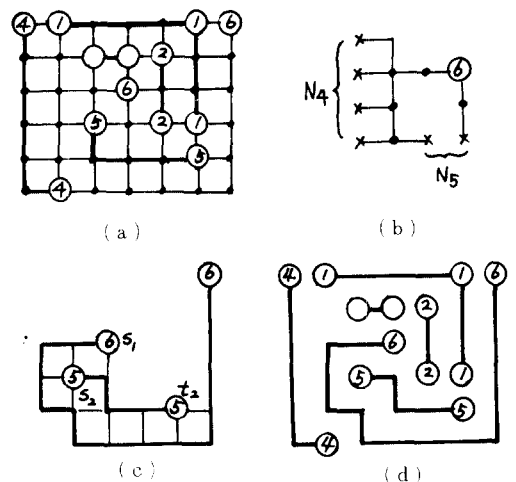


그림 7. 未結線 信號線의 修正

Fig. 7. Example of applying TPP algorithm.

V. 結 論

本 論文에서는 TPP를 해결하는 $O(|V| \times |E|)$ 의 알고리즘을 제안했다.

현재 LSI, PCB의 레이아웃CAD 시스템에 내장된 자동 배선기의 프로그램은 100% 結線率을 達成치 못하고 있으며, 未結線 신호선에 대하여는 인간의 경험과 판단에 의해 수정을 加하여 行하여 지고 있다. 이 경우 配線의 턴 어라운드(turn around) 時間中, 인간에 의한(인터 액티브 시스템과의 對話) 修正作業 時間이 거의 대부분을 차지하는 것으로 알려져 있다.^[6]

극심한 경우에는 처음부터 다시 시작하는 事例도 드물지 않다.

따라서, 人間の 介入을 最小로 하는것이 레이아웃 시스템의 성능 향상의 관건이라 볼 수 있다. 바꾸어 말하면 컴퓨터에 의한 자동修正이 필수 불가결하다. 그러나, 3개의 독립 경로를 求하는 문제는 NP완전 문제로 되어 現實적으로 最適解를 求하는 것은 불가능하다. 그러므로 TPP에 있어 복잡도를 감소시키는 알고리즘에 대한 연구가 필요하며, 3개의 독립 경로 以上에 對한 휴리스틱 알고리즘의 개발이 필요하다고 생각되어진다.

參 考 文 獻

- [1] Karp, R.M., *On the Complexity of Combinatorial Problem*, Networks 5, pp. 45-68, 1975.
- [2] Perl, Y and Shiloach, Y., "Finding two disjoint paths between two pairs of vertices in a graph" *J. Assoc. Comput. Math.* 25, pp. 1-9, 1978.
- [3] Shiloach, Y., *A Polynomial Solution to the Unidirected Two Paths Problem*, Tbid. 27, pp. 445-56, 1980.
- [4] Lee, C.Y., "An algorithm for path connections and its applications", *IRE Trans. EC-10*, pp. 345-54, 1961.
- [5] Aker, S.B., "A modification of Lee's path connection algorithm", *IEEE Trans. EC-16*, 1, pp. 97, 1967.
- [6] 鄭正和, "LSI의 Layout CAD," Chapt. 1, 早稻田大學 博士 請求論文, 1980.
- [7] Even S. and Tarjan, R.E., "Network flow and testing graph connectivity", *SIAM J. Comput.* 4, pp. 507-18, 1975.
- [8] Auslander, L and Parter, S.V., "On in bedding graphs in the sphere". *J. Math. Mech* 10, pp. 517-23, 1961.
- [9] Fisher, G.J. and Wing, O., "Computer recognition and extraction of planar graphs from the incidence matrix", *IEEE Trans. CT-13*, pp. 154-63, 1966.
- [10] Hopcroft, J.E. and Tarjan, R.E., "Efficient planarity testing". *J. ACM* 21, pp. 549-68, 1974.
- [11] Tarjan, R.E., "Depth-first search and linear graph algorithms", *SIAM J. Comput.* 1, pp. 146-60, 1972.