論文 87-24-4-9

간노드과잉을 고려한 단층 열 라우팅 알고리즘

(Single-row Routing Algorithm with Between Node Congestion)

李 南 日*, 李 相 祚**

(Nam Il Lee and Sang Jo Lee)

要 約

本 論文에서는 PCB기판 設計에서 매우 중요한 도로과잉과 간노드과잉을 最小化하기 위한 單層 列 라우팅 알고리즘을 제시하였으며 간노드과잉을 최소화하기 위하여 基準線通過라는 개념을 도입, 여러가지 경우에 적용한 결과 아주 좋은 成果를 거두었다.

基準線通過 횟수를 줄인다는 것은 간노드과잉을 줄이는데 기여할 뿐 아니라 복잡한 집적회로에서 네트의 길이를 줄인다는 것과 같게되므로 결과적으로 電力의 소비를 줄일 수 있게 된다.

Abstract

In this paper, the single-row routing algorithm for minimizing the street congestion and reducing the between node congestion is deviced.

To reduce the between node congestion, reference line crossing is defined and used. Reducing the number of total reference line crossing means reducing the total length which help reduce the power consumption of the integrated system and reducing the between node congestion which help reduce the wiring area.

This algorithm has been implemented and tested with various example, then produced good results.

I. 序 論

最近에 마이크로 電子工學의 發展으로 인하여 回路設計에 많은 진전이 이루어지고 있는데 여기서 가장 큰 관심사는 수 만개의 서로 連結된 셀(cell)을 포함하고 있는 칩(chip)이나, 回路의 배선을 構成하는 것이다. PCB(printed circuit board) 기판 설계에서 매우 중요한 單層 列 라우팅 問題에는 핀들의 列과 列 사이를 分離하는 문제와 한개의 열에서 인접한 핀 사이를 분리하는 문제가 있다. 첫번째 문제에 대해서는 이미 여러 논문[1~7]에서 언급된 바 있으며 이를 위한 必要充分 條件과 최적의 알고리즘까지 발표되었지만[10] 두번째 문제에 대해서는 아직 많은 硏究가 進行되지 않고 있다. 그런데 이들 두 문제는 서로 밀접한 관계를 가지고 있기 때문에 어느 하나만 따로 분리해서 생각하기는 어렵다.

本 論文에서는 이들 두 문제를 解決하기 위해서 基準線通過를 이용하여 도로밀집을 최소화하면서 네트의 길이를 줄이기 위한 효과적인 알고리즘을 提示하였으며 여러가지 경우의 資料를 가지고 遂行한 結果 좋은 成果를 거두었다.

*正會員, 東義工業專門大學 電子計算科

(Dept. of Comp. Science, Dongeui Junior College)

**正會員, 慶北大學校 電子工學科

(Dept. of Elec. Eng., Kyungpook National Univ.)

接受日字 : 1986年 8月 18日

Ⅱ. 本 論

1. 用語 說明

한개의 直線위에 均等한 간격으로 위치한 n개의 노드(node)의 集合을 V라고 하면 V={1, 2, 3,…, n}이 되고 전기적으로 同等한 값을 유지하는 것끼리 연결되는 노드로 구성되는 네트의 집합을 L이라고 하면 L={N_1, N_2, N_3,…,N_m}이 되며 이 네트는 다음 條件을 만족해야 한다.

$$N_i \cap N_j = \phi \ (i \neq j) \quad (1)$$

$$\bigcup_{i=1}^{m} N_i = \{1, 2, 3, \cdots, n\} \quad (2)$$

주어진 조건에 따라 노드들을 연결하는 수직 또는 수평선을 이용한 實在化(realization)의 예는 그림 1 과 같다.

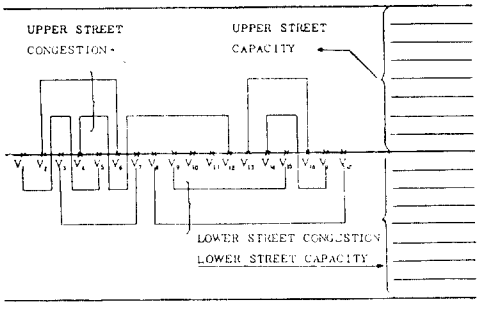


그림 1. 단층열 실세화와 기본용어의 예
Fig. 1. An Example of a Single-row Realization and Basic Terminology.

그림 1 에서 直線 R이 기준선(reference line) 이고 기준선 윗부분이 위쪽도로, 아랫부분이 아래쪽 도로이고, 인접한 노드사이를 통과하는 네트 수가 간노드밀집(between node congestion)이다. 위쪽 도로에서 허용되는 수평 트랙(track) 수는 위쪽도로 用量(upper street capacity)이고, 위쪽도로에서 實在化하는데 要求되는 수평트랙 수는 위쪽도로밀집(upper street congestion)이다.

각 用語에 대해서 아제 다음과 같은 표기법을 사용하기로 한다.

c_{us} : 위쪽도로밀집

c_{ls} : 아래쪽도로밀집

c_{ui} : 實在化에서 i번째 노드 위로 통과하는 수평 트랙 수

c_{li} : 實在化에서 i번째 노드아래로 통과하는 수평 트랙 수

위 表記法을 이용하면 다음과 같은 관계식을 얻을 수 있다.

$$c_{ls} = \max_i c_{li} \ (i = 1, 2, 3, \cdots, n)$$

$$c_{us} = \max_i c_{ui} \ (i = 1, 2, 3, \cdots, n)$$

여기서 c_{us}와 c_{ls}중 큰 값이 도로밀집 Q_d이다. 한 네트 N_i가 한 노드 V_i와 만나지 않고 위(아래)로 지나갈 때 N_i는 V_i를 위(아래)로 커버(cover)한다고 하고 노드 V_k에서 네트 N_i 위(아래)로 다른 네트 N_j가 지나갈 때 V_k에서 N_j는 N_i를 위(아래)로 커버한다고 한다. i번째 노드 V_i에서 수직선을 그었을 때 이 수직선에 의해서 잘리는 네트 수를 노드 V_i에서의 커트 수(cut number: c_i)라고 하면 노드커트 수 c_i는

$$c_i = c_{ui} + c_{li}$$

가 된다.

네트 N_j의 커트 수 q_j는 네트 N_j의 노드들의 커트수 중에서 가장 큰 것으로 한다. 즉 네트 N_i에 V_i, V_j, V_k의 노드가 있다면 네트 커트수 q_i은

$$q_i = \max\{c_i, c_j, c_k\}$$

가 된다. 例를 들면 그림 2 에서 V_{10}과 V_{11}로 연결되는 네트 N_5의 경우 C_{10}은 c_{u10}와 c_{l10}의 合 3 이 되고, c_{11}도 같은 방법으로 계산하면 3 이 된다. 그래서 네트 N_5의 커트수 q_5는 3 이 됨을 알 수 있다.

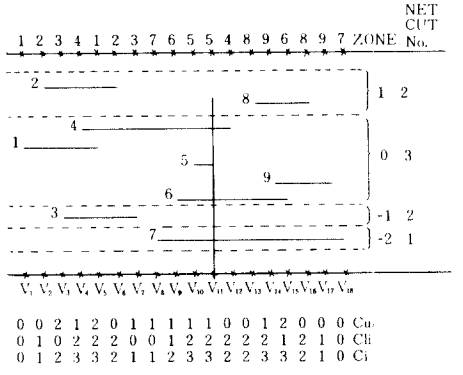


그림 2. 네트 실세화와 노드 커트수를 위한 간격 도식적 표현
Fig. 2. Interval Graphical Representation for the Net List Realization and Node Cut Numbers.

그림 2 에서 가장 위에 있는 네트의 커트 수가 q라면 c_{ls}는 최소한 q가 되고 가장 밑에 있는 네트커트 수가 q라면 c_{us}가 최소한 q가 되므로 가장 작은 커트 수를 가진 네트를 가장 바깥 쪽에 두어야만 좋은 實在化를

구하는 方法이 된다.[2] q_M 및 q_m을

$$q_M=\max_j q_j$$

$$q_m=\min_j q_j$$

와 같이 두면 $Q_0\geq\max\{q_m, q_t\}$를 만족하는 Q_0의 범위 내에서 도로밀집을 最小化하는 實在化를 구할 수 있다. 여기서 $q_t=\lceil q_M/2\rceil$인데, $\lceil x\rceil$는 x보다 작지않은 가장 작은 定數를 나타낸다. 또한 도로밀집을 최소화하기 위한 필요충분 조건으로 각 노드 V_i에 대해서

$$c_i=q_t+k\ (k=1, 2, 3,\cdots, q_M-q_t)$$

이고, V_i를 포함한 네트가 적어도 k개의 네트에 의해서 위로 그리고 아래로 同時에 커버되고 있을 때

$$Q_0=q_t$$

가 된다.[2]

2. 理 論

네트를 커트수에 따라 分類하고 존(Zone : Z)이라는 槪念을 사용하여 분류된 네트를 각 존별로 구별한다. 가장 큰 네트 수 q_M을 가진 네트를 존Z_0에, 커트 수가 q_M-i인 네트는 존Z_i에 분배한다.

그림 2에 대해서 위 사실을 適用하면 $q_M=3$이 되고 N_1, N_4, N_5, N_6 및 N_9가 Z_0에, N_2, N_3, N_8가 Z_1에, 그리고 N_7가 Z_2에 割當된다.

네트 리스트 $L=\{N_1, N_2,\cdots, N_m\}$와 도로밀집 Q_0를 가지고 최적의 實在化를 구하는 비트 順序 O_1이 있다고 할때, 네트커트 수가 Q_0보다 큰 네트 N_j와 N_k 사이에 네트커트 수가 Q_0보다 작은 네트 N_i가 있다면, N_i가 N_j와 N_k보다 앞에 오거나 뒤에 오는 다른 최적의 네트 순서 O_2가 存在한다.[10]

이 事實을 이용하여 네트커트 수가 Q_0보다 큰 네트 사이에 커트 수가 Q_0보다 작은 네트 사이에 커트 수가 Q_0보다 작은 네트가 존재하면 交換을 反復하여 커트수가 Q_0보다 큰 네트 사이에는 Q_0보다 작은 커트 수를 가진 네트가 존재하지 않도록 하면 다음 식(1)을 만족하는 네트순서를 구할 수 있다.

$$Z^L_{q_M}<Z^L_{q_M-1}<\cdots<Z^L_{q_M-Q_0+1}$$
$$<(\text{Nets with cutnumbers}>Q_0)$$
$$<Z^u_{q_M-Q_0+1}<\cdots<Z^u_{q_M} \qquad (1)$$

여기서

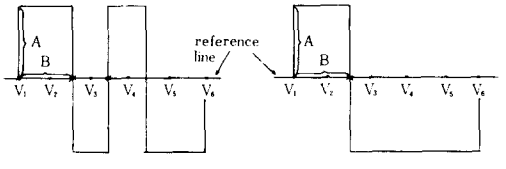
$$Z^u_i\cup Z^L_i=Z_i$$
$$Z^u_i\cap Z^L_i=\phi$$

이고 u와 L은 위쪽과 아래쪽도로를 의미한다. 식(1)을 좀 더 細分하면 다음과 같은 式(2)가 된다.

$$Z^L_{q_M}<Z^L_{q_M-1}<\cdots<Z^L_1<Z_0$$
$$<Z^u_1<Z^u_2\cdots<Z^u_{q_M} \qquad (2)$$

알고리즘을 通해서 식(2)를 만족하는 네트의 순서를 구하면 최적의 네트순서가 된다.

그런데, 각 존내에서의 Z^u와 Z^L의 구분은 도로밀집에서는 영향을 미치지 않으나 기준통과(reference line crossing) 횟수 RC에 영향을 미치게 된다. 여기서 기준선 통과란 한 네트가 기준선과 교차하는 것을 의미하며 각 네트에 대해서 기준선통과 횟수가 많으면 네트의 길이가 길어질 뿐 아니라 간노드밀집에도 영향을 미치게 되어 배선 설계에 큰 어려움을 더하게 된다.



(a) RC=3 (b) RC=1

그림 3. 기준선 통과의 예
Fig. 3. Illustration of Reference Line Crossing.

그림 3에서 (a)의 경우 전체 네트의 길이를 계산하면 8A+5B가 되며 (b)의 경우에는 4A+5B가 되어 길이면에서 기준선통과 횟수에 따라 顯著하게 차이가 남을 알 수 있다.

3. 알고리즘

네트 리스트 $L=\{N_1, N_2,\cdots, N_m\}$에서 L을 두개의 부리스트 L_1와 L_2를

$$L_1\cap L_2=\phi$$
$$L_1\cup L_2=L$$

이 되도록 분리했을 때 N_i가 L_1에 속한다고 하자. L에 있어서 L_1에 대한 N_i의 내부커트 수(internal cut number : $_{A}i$)는 L_1에서의 N_i의 커트수가 된다. 그리고 L_1에 대한 N_i의 外部커트 수(residual cut number : $_{R}i$)는 $\{L_2\cup N_i\}$에서의 N_i의 커트수가 된다.

그림 2에서 보인 네트리스트를 부리스트로 分離했을 때 그중 하나가 그림 4 (a)이다. 그림 4 (a)에서 N_3에 대한 내부커트 수와 외부커트 수를 구한 것이 그림 4 (b)가 되는데 여기서 N_3에 대한 내부커트 수는 (1, 1)이 되고 외부커트 수는 (1, 0)가 된다. 그래서 $_{A}3=1$, $_{R}3=1$이 된다. 그림 2를 각 노드의 순서대로 연결하면 그림 5 (a)처럼 된다. 實在化를 하는 경우 그림5 (a)에서 점선을 곧게 폈을 때 실선이 수직 또는 수평선만으로 이루어지도록 하면 얻고자하는 그림 5 (b)와 같이

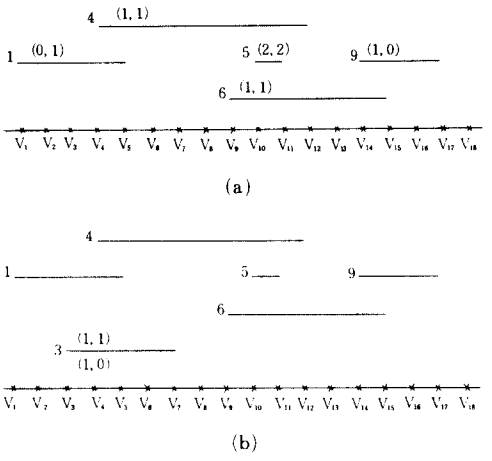


그림 4. (a) 내부 커트수를 고려한 그림 2 의 부분 집합
(b) 네트 N_3에 대한 내부 커트수와 외부 커트 수

Fig. 4. (a) Subset of Nets from Fig.2 with Internal Cut Numbers.
(b) Internal and Residual Cut Numbers for the Net N_3.

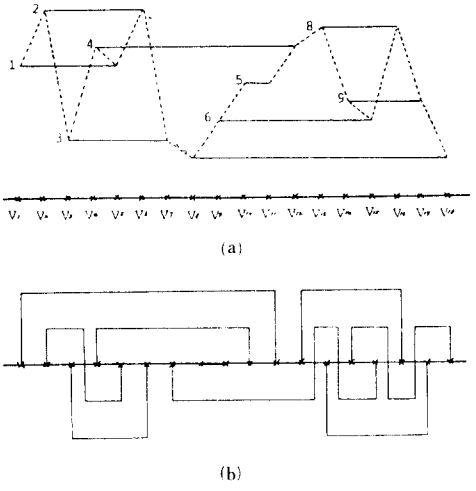


그림 5. (a) 그림 2 에 대한 간격 도식적 표현
(b) 그림 2 에 대한 네트 리스트의 실제화

Fig. 5. (a) Interval Graphical Representation with Respect to Fig.2.
(b) Net List Realization of Fig. 2.

된다. 그림 5 (a)에서의 점선이 實在化에서 기준선이 된다.

그림 4 (a)에서 노드 V_{13}과 V_{18}을 연결하는 네트 N_8을 그릴 경우 그림 6 에서와 같이 A 즉 위쪽도로에 넣는

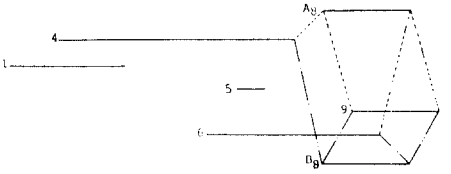


그림 6. 알고리즘에 적용한 기준선 통과
Fig. 6. Reference Line Crossing in the Algorithm.

경우와 B 즉 아래쪽도로에 넣을 경우의 差異點을 살펴보면, 먼저 A에서는 현 상태에서 實在化를 할 경우 기준선 통과가 1 번 (네트 N_9) 일어나지만 B의 경우에는 2 번(네트 N_6)의 기준선 통과가 發生함을 알 수 있다.

이 사실로 미루어 보아 같은 존에서 위쪽과 아래쪽 도로의 구분을 임의로 하지 않고 기준선 통과의 횟수를 보고 작은 쪽에 배당하는 것이 좋은 결과를 가져오게 됨을 알 수 있다.

기준선 통과를 顧慮한 實在化를 위한 네트 리스트의 순서를 구하는 알고리즘은 그림 7 과 같다. 네트를 커트 수에 따라 존으로 분류하고 존Z_0부터 시작해서 트랙에 配當한다. 같은 존에서의 순서는 重疊이 생길 때는 내부커트 수와 외부커트 수를 구해서 먼저 내부커트 수가 큰 순서대로 배당한다. 둘다 같은 경우에는 任意로 하고 중첩이 생기지 않을 때는 같은 트랙에 배당한다. 그리고 위쪽트랙과 아래쪽트랙의 區別은 기준선 통과 횟수를 구한 뒤 작은쪽 트랙에 配定한다.

이 알고리즘에서는 假想的인 트랙 T_i를 사용하였는데 i의 구간은 $(-Q_0, Q_0)$이며 트랙의 순서는 |i|가 증가하는 순서가 된다.

그림 2 에 대해서 그림 7 의 알고리즘을 적용하면 네트커트 수 3 이 가장 크므로 여기에 속한 네트를 즉, N_1, N_4, N_5, N_6 및 N_9가 존Z_0에 속하게 되고 네트커트 수가 2 인 N_2, N_3 및 N_8이 존Z_1에 그리고 네트커트수가 1 인 N_7이 존Z_2에 속함을 알 수 있다. 그림 4 (a)에서 내부커트 수를 보면 N_5는 2, N_1, N_4, N_6 및 N_9은 1 이 되고 외부커트 수를 N_1, N_4, N_6 및 N_9은 2 가 되어 N_1, N_4, N_6 및 N_9은 내부커트 수와 외부커트 수가 모두 같게 되므로 이들의 순서는 任意로 하면 된다.

그런데 N_1, N_5 및 N_9은 重疊이 되지 않으므로 같은 트랙 T_0에, N_4 및 N_6를 위쪽도로와 아래쪽도로로 분류하는데 둘다 기준선 통과 횟수가 0 이므로 임의로 배당하면 된다. 그래서 N_4를 트랙 T_{+1}에, 그리고 N_6는 트랙 T_{-1}에 排當하기로 한다. 지금까지의 순서는 N_6 (N_1, N_5, N_9), N_6가 된다.

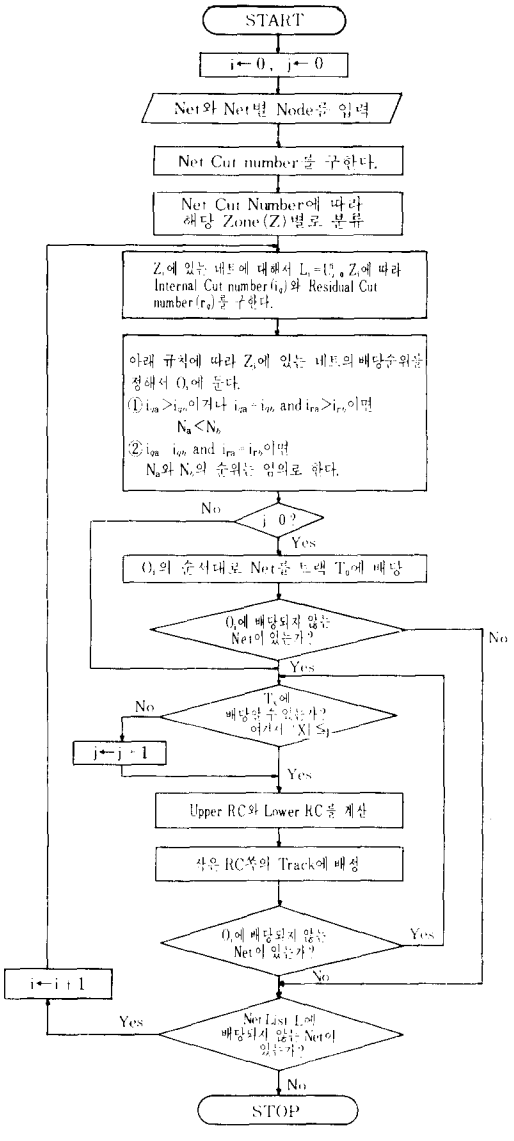


그림 7. 알고리즘 순서도
Fig. 7. Flow Chart of Algorithm.

다음은 존 Z_1에 속해 있는 N_2, N_3 및 N_8의 차례이다. 그런데 N_2 및 N_3는 내부커트 수가 2 이고, 외부커트 수가 0 으로 둘다 같게 되어 任意로 배정할 수 있다. N_8은 내부커트 수가 1 이다. 그래서 이들의 순서는 N_2, N_3, N_8이 된다. N_2와 N_3를 위쪽, 아래쪽으로 分類함에 있어서 이들의 기준선 통과 횟수를 구하면 N_2의 위쪽 기준선통과 횟수와 아래쪽 기준선통과 횟수가 다같이 1 이고 N_3에 대해서도 둘다 1 이 되므로 任意로 N_2를 트랙 T_2에, 그리고 N_3를 트랙 T_{-2}에 排定한다. 그런데 N_8의 경우 N_2와 N_3에 대해서 重疊이 일어나지 않으므로 T_2와 T_{-2}에 임의로 배정할 수 있으나 아래쪽 기준선통과 횟수는 2 이고 위쪽 기준선통과 횟수는 1 이므로 N_8은 기준선통과 횟수가 작은 트랙T_2에 배정한다. 지금까지의 部分的인 순서를 보면

(N_2, N_3), N_4, (N_1, N_5, N_9), N_6, N_3

이 된다.

이제 N_7의 경우인데 위쪽 기준선통과 횟수는 1 이고 아래쪽 기준선통과 횟수는 0 이므로 아래쪽 트랙T_{-2}에 배정한다. 그래서 이들의 全體순서는

(N_2, N_8), N_4, (N_1, N_5, N_9), N_6, N_3, N_7

이 된다.

이 最終 結果에 대한 트랙 排定은 그림 8 과 같다.

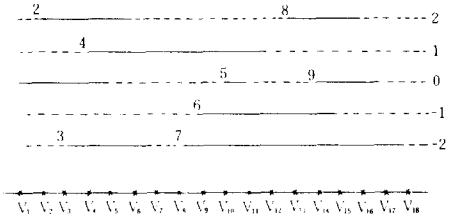


그림 8. 그림 1 에 대한 트랙 배정
Fig. 8. Track Assignment for the Example of Fig. 1.

4. 계산성 복잡성(computational complexity)

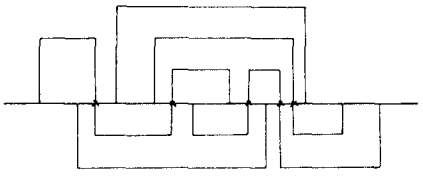
노드수를 n개, 네트수를 m개라면 모든 노드에 대해서 커트수를 구하는데 n*m의 수행이 필요하게 되며, 네트들을 적절한 존에 배당하기 위해서는 분류가 필요하게 되는데 이를 위해서 n+mlgm의 수행이 필요하고, 내부커트수와 외부커트 수를 구하기 위해서 n*m의 수행이 필요하며, 기준선통과 수를 구하는데 n*m의 수행이 요구된다. 따라서 이 알고리즘의 계산성 복잡성(computational complexity)는 O(n*m+mlgm)이 된다.

5. 結果 및 考察

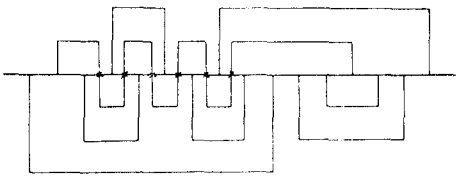
몇가지 경우의 네트리스트에 대해서 기준선 통과를 고려하기 전과 이를 고려한 알고리즘에 의한 結果를 比較해 보면 다음과 같다.

노드와 노드 사이의 間隔은 일정하므로 무시하고 도로와 도로사이의 간격을 1 로 했을 때 이들의 길이를 비교해 보면 (a)의 경우 종전 알고리즘에 적용한 결과 기준선통과 횟수가 5 가 되어 길이가 32인 반면 본 알고리즘에 의한 결과는 기준선통과 횟수가 3 으로 줄어

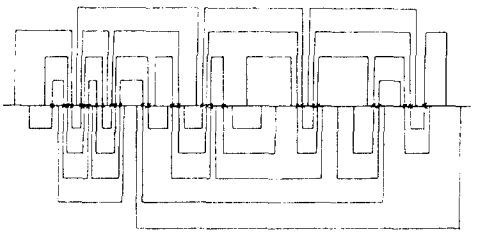
• 기준선 통과를 고려하기전 알고리즘의 결과



(a) RC=5, Q_0= 3

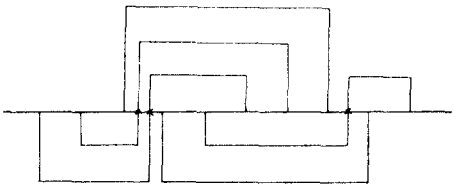


(b) RC=6, Q_0= 3

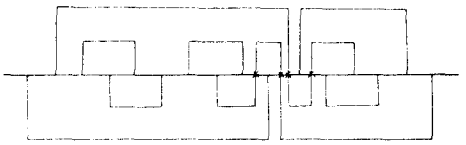


(c) RC=29, Q_0= 5

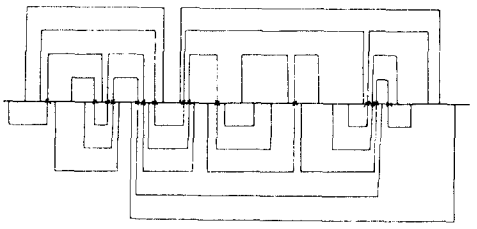
• 기준선 통과를 고려한 알고리즘의 결과



(a) RC= 3, Q_0= 3



(b) RC= 4, Q_0= 2



(c) RC=15, Q_0= 5

서 길이가 26이 되었다. (b)의 경우에도 마찬가지로 기준선통과 횟수가 6에서 4로 줄어서 길이는 42에서 35로 줄어 들었으며, (c)의 경우에는 기준선통과 횟수가 29에서 15로, 길이는 192에서 134로 줄었음을 알 수 있다.

위 결과로써 기준선통과를 고려한 본 알고리즘은 도로밀집의 最適條件을 만족하면서 길이면에서는 종전 알고리즘에 비해서 많이 減少됨을 알 수 있다.

Ⅲ. 結 論

本 論文에서는 PCB기판 설계에서 매우 중요한 單層列 라우팅 알고리즘을 다루었는데 一般的으로 다층 열 라우팅을 硏究하는데 있어서는 단순히 도로밀집을 最少化하기 위한 方向으로만 진행되는데 여기서는 도로밀집 뿐 아니라 기준선통과를 道入하여 네트의 길이를 줄이는 알고리즘을 提示하였다.

기준선통과를 알고리즘에 適用함으로써 이 槪念을 도입하기전 알고리즘의 問題點이었던 같은 층에서의 위쪽, 아래쪽 도로의 구분을 明確히 할 수 있었으며, 기준선통과의 횟수를 줄임으로서 회과 회을 연결하는 네트의 길이를 감소시킬 수 있었다.

그런데 기준선통과 횟수를 줄인다는 것이 반드시 라노드 밀집을 最少化하는 방법은 아니므로 라노드밀집의 최소화를 위한 연구가 진행되어 기판설계에 큰 發展을 기대해 본다.

參 考 文 獻

[1] B.S. Ting, E.S. Kuh and I. Shirakawa "The multilayer routing problem: algorithms and necessary and sufficient conditions for the single-row sngle lager case", *IEEE Trans. Circuits syst.*, CAS-23, 768-778, 1976.

[2] E.S. Kuh, T. Kashiwabara, T. Fujisawa, "On optimum singlerow routing", *IEEE, Trans. Circuits Syst.*, CAS-26, 361-368, 1979.

[3] S. Tsukiyama, E.S. Kuh and I. shirkawa, "An algorithm for single-row routing with prescribed street congestions", *IEEE Trans. Circuits, Syst.*, CAS-7, 1980.

[4] T. Yoshimura and E.S. Kuk, "Efficient algorithms for channel routing", *IEEE Trans, Circuit Syst.*, CAD-1, 1982.

[5] S. Tsukiyama, E.S. Kuh and I. Shiratawa, "On the layering problem of multilayer PWB wiring ", *IEEE Trans. Computers*, CAP-2, 1983.

[6] R. Raghavan and S. Sahni, "Single-row routing", *IEEE Trans. Computers*, CAD-32, 209-220, 1983.

[7] L. Anderson, "On street congestion in single-row routing", *IEEE Circuit theory*, 303-319, 1983.

[8] R. Raghawan and S.K. Sahni "The compltxity of single-row routing", *IEEE Trans. Circuits syst*, CAD-31, 1984.

[9] S.Y. Han and S.K. Sahni, "Single-row routing in narrow streets", *IEEE Trans, Computers*, CAD-3, 1984.

[10] T. Tang, M. Sadowsla, "An efficient single-row routing algorithm", *IEEE Trans. Computers*, CAD-3, 1984.

[11] H.W. Carter, and M.A. Breuer, "Efficiont singlelayer routing along a line of points", *IEEE Trans, Computers*, CAD-3, 1983.