

Divide-and-Conquer 기법을 사용한 스위치박스 배선기

(A Switchbox Router using Divide-and-Conquer Technique)

李昇昊* 鄭正和*

(Seung Ho Lee and Jong Wha Chong)

要 約

본 논문에서는 divide-and-conquer 기법을 사용하여 대규모 배선문제를 소규모 배선문제로 분할하여 해결하는 새로운 스위치박스 배선기인 CONQUEROR를 제안한다. CONQUEROR는 배선영역 분할 및 핀 할당 과정, 분할된 각 배선영역에서의 상세배선(detailed routing)과정, 그리고 배선 실패한 배선영역의 핀 재할당 과정으로 구성된다. 먼저 주어진 대규모 배선영역을 여러개의 소규모 부(sub) 배선영역으로 분할한후, 분할된 각 부 배선영역의 변(boundary)상에 웨이트식에 의한 핀 할당을 한다. 분할한 모든 부 배선영역에 핀 할당을 한후 부 배선영역 별로 상세배선을 수행한다. 상세배선 단계는 단일층에서의 미로배선(layerless maze routing), coloring에 의한 층 할당, 그리고 재배선(rip-up and reroute)순으로 수행된다. 상세배선이 실패한 부 배선영역이 발생하면 실패한 부 배선영역의 모든 핀들을 제거한후 재할당하여 다시 상세배선을 수행한다. Benchmark 데이터를 사용한 실험결과 CONQUEROR는 기존의 다른 스위치박스 배선기들에 비해 뛰어난 배선결과를 보였다.

Abstract

A new switchbox router, called CONQUEROR, is proposed in this paper. The proposed CONQUEROR efficiently routes large switchbox routing area using divide-and-conquer technique. The CONQUEROR consists of three phases: namely, partition of large routing area and assignment of optimal pins of sub-area, detailed routing of each sub-area, reassignment of pins after rip-up. First, large switchbox routing area is partitioned into several sub-areas and each sub-area contains 4-6 detailed grids. Then pins are assigned on boundary of sub-area by the estimated weight. Secondly, when global pin assignment is completed on all sub-areas, each sub-area is routed using detailed router. Also, detailed routing consists of three phases: layerless maze routing, assignment of layer using coloring, and rip-up and reroute. Lastly, if detailed routing of any sub-area fails, reassignment of pins after rip-up is invoked. Detailed routing is performed for the failed sub-area again. Benchmark test cases have been run, and on all the benchmark data known in the literature CONQUEROR has performed as well as or better than existing switchbox routers.

1. 서론

* 正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字: 1992年 11月 5日

상세배선(detailed routing)은 집적회로의 물리적
공정에서 컴팩션(compaction) 이전의 단계로서 배선

면적의 최소화, via의 최소화, 배선장의 최소화 등을 목적함수로 하여 배선영역 내에서 신호선들의 등전위 핀들을 서로 단락(short)이 생기지 않도록 연결한다. 한편 상세 배선기(detailed router)는, 전통적으로 채널 배선영역을 배선하는 채널 배선기와 스위치박스 배선영역을 배선하는 스위치박스 배선기로 분류되며, 전원이나 접지선을 배선하기 위한 river router¹와 패드(pad) 배선을 위한 moat router² 등과 같은 특수 배선기도 존재한다. 최근에는 칩면적을 줄이기 위하여 배선영역 이외의 셀 영역에서도 배선을 행하는 over-the-cell router³도 개발되고 있다. 이중 채널 배선기⁴는 그 효율성으로 인하여 널리 사용되어 왔으나, 최근에는 칩의 면적과 성능을 향상시킬 수 있는 빌딩블록 레이아웃 설계방식⁵이 널리 사용됨에 따라서, 빌딩블록 레이아웃 설계시 발생하는 스위치박스 배선영역을 효과적으로 자동 배선할 수 있는 스위치박스 배선기의 필요성이 크게 대두되고 있다.

스위치박스 배선기는 1980년대 초에 처음으로 발표되기 시작했다. 그러나 초기의 스위치박스 배선기^{6,7}들은 채널 배선법을 확장한 알고리즘을 적용하였기 때문에 배선결과가 만족스럽지 못했다. 그후 1985년에 발표된 WEAVER⁸는 knowledge-based 기법을 사용하여 배선결과를 향상시켰다. 그러나 수백개의 규칙들을 사용하였기 때문에 컴퓨터 수행시간이 많이 소모되어 그효율성이 상당히 떨어졌다. 1986년에 발표된 MIGHTY⁹는 incremental routing modification 기법을 사용하여 빠른 시간 내에 좋은 배선결과를 얻을수 있었으나 커다란 배선영역에 대해서는 효과적인 배선결과를 얻을수 없었다. 1988년 발표된 BEAVER¹⁰는 corner router, line sweep router, thread router 등 3개의 router를 단계별로 적용하여 배선을 한후 layerer를 사용하여 최종적으로 층을 할당함으로써 배선을 완료한다. BEAVER는 기존에 발표되었던 다른 스위치박스 배선기에 비해서 빠른 시간내에 비교적 좋은 배선결과를 얻었으나, 변(boundary)상의 핀이 어느 층에나 자유롭게 할당될수 있도록 하였기 때문에 이 자유도를 제거하면 좋지 않은 배선결과를 나타내었다. 1989년에 발표된 SILK¹¹는 greedy 알고리즘에 의해 초기배선을 한후 simulated evolution 기법을 사용하여 rip-up할 신호선들을 선택하고 변형된 미로 배선 알고리즘을 사용하여 재배선한다. SILK는 simulated evolution 기법을 사용하였기 때문에 최적의 해에 근접한 배선결과를 얻을수 있었으나 WEAVER와 마찬가지로 컴퓨터 수행시간이 많이 소모되는 불합리한 면이 있어 커다란 배선영역에 대해

서는 배선의 효율성이 떨어졌다. 그후로도 몇개의 스위치박스 배선기¹²가 발표되었으나 배선결과나 배선수행시간 면에서 여전히 불합리한 면을 나타내었다.

본 논문에서 제안하는 스위치박스 배선기인 CONQUEROR는 배선의 효율성을 높이기 위하여 divide-and-conquer 기법을 이용한다. 대규모 배선영역을 소규모의 작은 부 배선영역으로 분할한후 분할된 부 배선영역의 변상에 신호선들을 할당하여 각 부 배선영역별로 상세배선을 수행한다. 상세배선은 단일층에서의 미로배선(layerless maze routing), coloring에 의한 층 할당 그리고 재배선(rip-up and reroute)순으로 수행된다. 상세배선이 실패한 부 배선영역이 발생하면 실패한 부 배선영역의 모든 핀들을 제거한후 재할당하여 다시 상세배선을 수행한다. 모든 부 배선영역의 상세배선이 성공하면 각각의 부 배선영역을 결합하여 원래의 배선영역으로 복원시킴으로써 배선을 완료한다.

II. CONQUEROR의 전체 알고리즘 흐름도

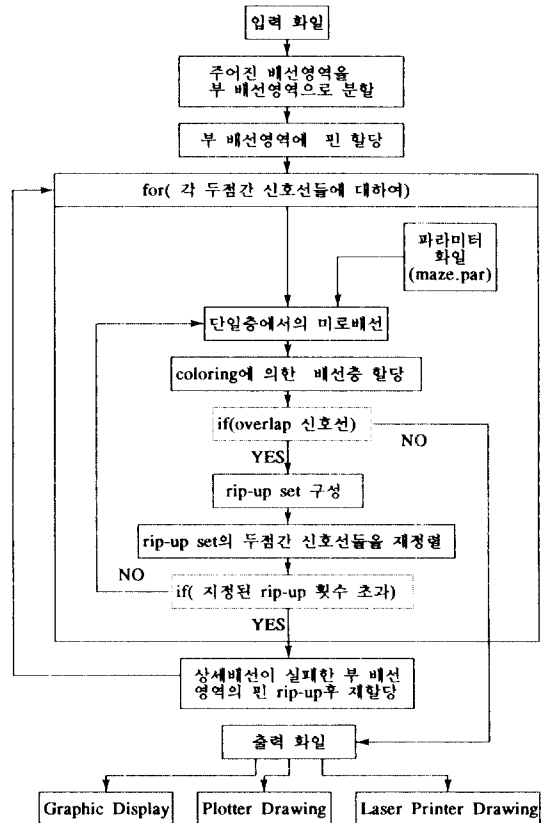


그림 1. CONQUEROR의 전체 알고리즘의 흐름도
Fig. 1. Overall flow chart of CONQUEROR.

그림 1은 CONQUEROR의 전체 알고리즘의 흐름도를 나타내고 있다. 본 논문에서 제안한 CONQUEROR는 2개의 메탈층을 이용하여 배선하는 2층 스위치박스 배선기이며 흐름도에 나타나있는 파라미터(parameter) 화일에는 단일층에서의 미로배선시 사용되는 여러가지 비용 값이 담겨있다. 따라서 파라미터 화일의 비용 값을 변경함에 따라 배선경로를 사용자가 원하는 형태로 바꿀수 있다.

III. 배선영역 분할 및 핀 할당

입력으로 주어진 커다란 스위치박스 배선영역을 그림 2와 같이 여러개의 작은 부(sub) 배선영역으로 분할한다. 이때 상세 그리드(detailed grid)와 가상 그리드(global grid)라 명칭되는 2개의 그리드가 생성된다. 상세 그리드들은 스위치박스 배선영역을 심볼릭하게 나타낸 것으로 실제 상세배선이 이루어지는 그리드들이다. 가상 그리드들은 분할된 부 배선영역들의 경계로 구성되는 새로운 그리드들로써 핀들이 할당되어지는 그리드들이다. 분할되는 하나의 부 배선영역의 크기는 4-6개의 상세 그리드가 포함되도록 한다.

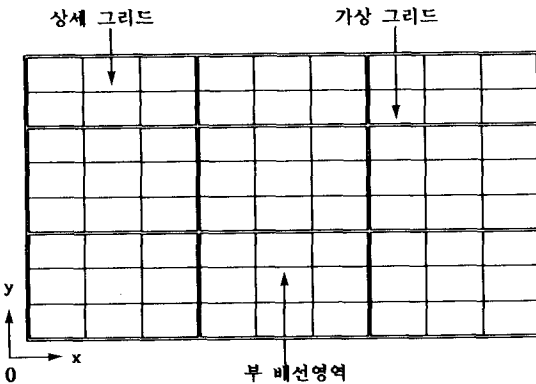


그림 2. 여러개의 작은 부 배선영역으로 분할된 스위치박스 배선영역

Fig. 2. Switchbox area partitioned into several sub-areas.

배선영역을 분할한후 분할된 부 배선영역에 핀 할당을 하기 위하여 먼저 모든 다점간 신호선(multipin net)들을 minimum cost spanning tree를 사용하여 두점간 신호선(two-pin net)들로 분할한다. 분할된 모든 두점간 신호선들의 두 핀을 대각선으로 연결하여 생기는 가장 작은 사각형을 zone으로 설정

하여 zone의 크기가 증가하는 순으로 정렬한다. 정렬된 순서에 의해서 두점간 신호선들을 웨이트식에 의하여 분할된 각 부 배선영역의 가상 그리드에 할당한다. 이때 가상 수직 그리드에는 1층, 가상 수평 그리드에는 2층이 할당되도록 한다. 웨이트 식은 배선장의 최소화, 배선장의 직선화, via의 최소화를 기할수 있도록 다음과 같이 정의 되어있다.

$$\text{웨이트 식} = (D + \alpha * S + \beta * V) * M$$

단 α, β 는 $\alpha < \beta$ 인 상수

D : 목적점(target point) 까지의 manhattan distance

S : 0 <- 배선형태가 일직선인 경우
1 <- otherwise

V : 1 <- 불필요한 via를 발생케 하는 경우
0 <- otherwise

M : if(difficult column OR difficult row) {
1 <- difficult column or row
2 <- neighbor difficult column or row
2 * ($\alpha + \beta$) <- otherwise
}

else
1

전술한 웨이트 식에서 difficult column이나 difficult row는 배선영역의 첫번째나 마지막번째 row나 column이 빈 경우가 있을때를 나타낸다. 그림 3에서와 같이 2번 신호선을 가상 그리드에 할당할 때 먼저 현재의 부 배선영역내의 모든 가상 그리드들에 대한 웨이트 값을 웨이트식에 의하여 계산한다. 계산된 가상 그리드들 중에서 가장 작은 웨이트값을 갖는 가상 그리드를 선택하여 신호선을 할당한다. 이때 이미 할당된 신호선에 의하여 할당이 불가능할 경우 다음으로 작은 웨이트값을 갖는 가상 그리드를 선택하여 핀들을 할당한다. 예를들어 그림 3의 신호선 5를 첫번째 부 배선영역의 가상 그리드에 할당할 경우, 첫번째 선택된 가상 그리드가 κ 위치일때 이 위치에 이미 배선 방향이 수직성분인 6번 신호선이 할당되어 있더라도 κ 위치에 할당할 신호선 5의 배선방향은 수평성분 방향이므로 κ 위치에 수평방향의 5번 신호선을 할당한다. 그러나 다음번 부 배선영역에서 선택된 가상 그리드 λ 의 위치에서는 이미 수평성분 방향으로 할당된 7번 신호선에 의하여 할당이 불가능하게 된다. 따라서 다음으로 선택된 가상 그리드 ω 의 위치에 수평성분 방향의 5번 신호선을 할당하게 된다.

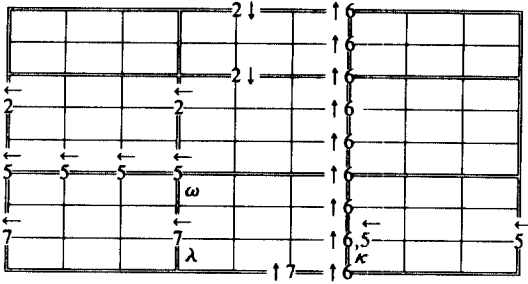


그림 3. 부 배선영역의 가상 그리드에 핀 할당예
Fig. 3. Example of pin assignment on global grid.

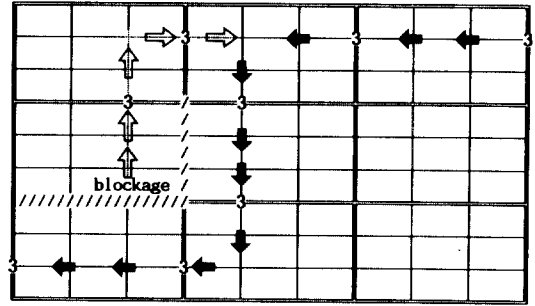


그림 4. Backtracking의 예
Fig. 4. Example of backtracking.

현재의 부 배선영역내의 모든 가상 그리드가 모두 할당 불가능할 경우 그림 4에서와 같이 그 이전의 부 배선영역으로 backtracking 하여 다른 경로를 찾아 목적점에 도달할때까지 핀들을 할당해 나간다.

그림 5는 분할된 부 배선영역에 핀들을 할당한 결과를 나타내고 있다.

CONQUEROR는 2층 배선기로서 개발되었기 때문에 2개의 층에 대하여 각각 할당된 결과를 나타내고 있다.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
14	14	14	14	14	14	14	14	14	14	14	14	14	0	0
13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
11	11	11	11	0	14	11	11	11	11	11	11	11	11	11
0	5	5	5	5	5	5	5	5	5	5	5	5	5	5
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	9	9	9	9	9	9	0	0	0	0	11	11	11	11
19	19	9	9	9	9	9	9	9	9	9	9	9	9	9
2	2	7	7	7	7	7	7	7	7	7	7	7	7	7
17	0	2	2	2	2	2	2	2	2	2	2	2	2	2
12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
16	16	16	0	0	0	19	19	19	19	19	19	19	19	19
4	0	6	6	6	6	6	6	6	6	6	6	6	6	6
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)

15	5	2	4	12	7	19	9	6	0	8	13	15	1	3	18
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	18	5	2	4	12	7	19	9	6	0	8	13	0	1	3
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	4	9	13	0	0	0	0	0	0	0	0	0	0	0	0
0	18	0	2	4	12	7	19	9	6	5	8	0	0	0	3
0	4	0	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
0	18	17	16	4	7	7	19	2	6	5	8	9	0	12	3
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
0	4	2	9	0	0	0	0	0	0	0	0	0	0	0	0
18	17	16	4	7	6	19	2	5	0	0	9	10	12	0	14

(b)

그림 5. 분할된 부 배선영역에 핀 할당된 결과

- (a) 수평성분의 핀 할당 결과
- (b) 수직성분의 핀 할당 결과

Fig. 5. Result of pin assignment on sub-areas.

- (a) result of pin assignment on horizontal global grid.
- (b) result of pin assignment on vertical global grid.

Ⅳ. 각 부 배선영역 내에서의 상세배선

모든 신호선들을 분할된 부 배선영역에 할당후, 각 부 배선영역 내에서 독립적으로 상세배선을 수행한다. 상세배선은 단일층에서의 미로배선(layerless maze routing), coloring에 의한 층 할당, 그리고 재배선(rip-up and reroute)순으로 수행된다.

1. 단일층에서의 미로배선(Layerless maze routing)

단일층에서의 미로배선은 배선층을 할당하지 않고 경로만을 결정하는 과정으로 Lee 알고리즘¹¹⁾을 기본으로 하여 여러 비용값(cost)을 도입한 변형된 형태이다. 사용되는 여러가지 비용값은 파라미터(parameter) 화일에 담겨있다. 변형된 특징은 경로 탐색(propagation)시 backtracking의 용이함을 위해 새로운 기법을 사용한다. 또한 zone 내에서 경로를 찾고 그 안에서 경로를 찾을 수 없는 경우에만 우회하도록 함으로써 특히 큰 배선영역의 경우 배선경로 탐색 시간을 단축시킨다. 먼저 출발점(start point)에서 갈 수 있는 모든 방향에 대한 그리드의 비용값을 계산하여 가장 작은 비용값을 갖는 그리드의 좌표를 리스트 구조에 첨가시킨다. 이미 비용값을 갖고 있는 그리드에 대해서는 현재 계산된 비용값과 비교하여 작은 비용값으로 대체한다. 즉 현재 그리드까지 오는 경로가 여러가지가 있을때 가장 적은 비용값을 갖는 경로만을 리스트에 첨가시킨다. 만일 비용값이 같을 경우는 bend수가 적은 경로를 우선적으로 선택하게 된다. 주어진 zone내에서 더이상 탐색할 그리드가 없을 경우 목적점(target point)에서 출발점까지 리스트에 매달린 그리드를 역방향으로 탐색하면 출발점에서 목적점까지 가장 비용값이 적은 경로를 발견할수 있게 된다. 경로탐색시 사용되는 비용값은 다음과 같다.

$$\begin{aligned} \text{현재 그리드의 비용값} &= \text{전 그리드까지의 비용값} \\ &+ \text{GridCost} + \text{OverlapCost} \\ &+ \text{CongestCost} \end{aligned}$$

GridCost는 한 그리드 전진하였을때 갖는 비용값이고, OverlapCost는 그림 6(a)에서와 같이 1번과 2번 신호선이 이미 그림과 같이 경로가 결정된 상태에서 3번의 경로를 찾을때 2번에 겹쳐 가는 대신 그림과 같은 경로를 찾을 수 있게 하는 비용값이다. CongestCost는 CONQUEROR가 2개의 층을 이용한 배선기 이므로 2개 이상의 신호선이 한 그리드에 할당이 되면 차후 그 그리드에서는 층 할당이 불가능해진다. 그럼에도 불구하고 그 그리드로의 탐색이 행해질 때에 감수해야할 비용값이 CongestCost가 된다. 그외에 파라미터 화일에 존재하는 비용값들 중에

서 BetterGridCost와WorseGridCost는 그림 6(b)와 같이 4번 신호선에 대하여 경로를 찾을때 이미 앞서 경로가 결정된 두점간 신호선의 어느 부분과 연결할지를 결정하는 비용값이다. 즉 그림 6(b)에서와 같이 bend로 표시된 ①에는 BetterGridCost가 적용되고 bend가 아닌 ②에는 WorseGridCost가 적용된다. 따라서 사용되는 비용값에 따라서 그림 6(b)에서와 같이 bend로 표시된 ①에 연결될수도 있고, bend가 아닌 ②연결될수도 있다. ParallelCost는 어느 신호선이 경로탐색시 현재 탐색하는 경로에 탐색방향과 같은 방향으로 다른 신호선이 존재할때 적용되는 비용값으로써 그림 6(c)에서 8번 신호선의 경우 β 에서 ParallelCost가 적용된다. BendCrossCost는 어느 신호선이 경로탐색시 갈수있는 경로의 수가 두가지일 경우 via가 생성될 가능성이 적은 쪽으로 가도록 하는 비용값으로써 그림 6(c)에서 8번 신호선의 경우 α 에서 BendCrossCost가 적용되어 via를 생성할 가능성이 많은 13번 신호선쪽으로 가지않고 다른 경로로 가고있다.

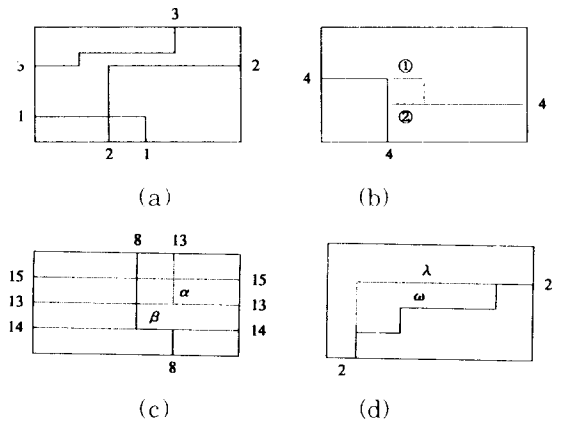


그림 6. 비용값에 의한 경로선택의 예
 (a) OverlapCost에 의한 경로선택의 예
 (b) BetterGridCost와 WorseGridCost에 의한 경로선택의 예
 (c) ParallelCost와 BendCrossCost에 의한 경로선택의 예
 (d) BlockingCost에 의한 경로선택의 예
 Fig. 6. Example of path finding by cost.
 (a) example of path finding by OverlapCost.
 (b) example of path finding by Better-GridCost and WorseGridCost.
 (c) example of path finding by Parallel-Cost and BendCrossCost.
 (d) example of path finding by BlockingCost.

BlockingCost는 재배선시 이미 찾았던 경로를 가 능한 다시 찾지 않도록 하는 비용값으로써 그림 6(d) 에서 2번 신호선의 경우 λ 부분이 재배선 전에 이미 찾았던 경로인데 재배선시에는 λ 부분에 경로탐색이 행해질때 BlockingCost가 적용되어 ω 의 경로가 찾 아진다.

NumberOfRipup은 재배선을 반복하는 횟수를 표시한다.

2. Coloring에 의한 층 할당

단일층에서의 미로배선에 의하여 모든 신호선에 대 한 경로를 찾은후, 신호선간의 단락을 해소하기 위하 여 coloring에 의하여 배선층을 할당한다. CON- QUEROR는 2개의 배탈층을 이용해 배선하므로 이 과정은 two coloring 문제이다.

그림 7은 앞서 단일층에서의 미로배선에 의하여 찾 아진 경로에 대한 예이다. 찾아진 각 경로의 wire 들 은 그리드 선분 단위로 분할되어 WireList에 저장되 어 있으므로 이를 사용하여 교차 그래프(intersec- tion graph)를 구성하게 된다. 교차 그래프에서 노 드는 그리드 단위의 선분을 나타내고 실선 형태의 예 지는 노드간의 교차 관계가 존재하지만 다른 신호선 노드인 경우에 형성되며, 점선 형태의 예지는 노드간의 교차 관계가 존재하지만 같은 신호선 노드인 경우 에 형성된다.

한편 배선영역의 변상에 위치한 선분들은 이미 배 선층이 결정되어 있으므로 교차 그래프를 구성한후 해당되는 노드에 고정된 색깔을 할당하고 색깔이 할 당된 노드들을 FixedNodeList에 저장한다.

그림 8은 그림 7에 대한 교차 그래프를 나타내고 있다. 이제 배선층을 결정하기 위하여 먼저 다른 신 호선간에 교차 관계가 있는 노드들에 대하여 coloring을 수행한다. 서로 다른 신호선의 노드와 노드간에 교차 관계가 있는 경우 그 두 노드는 서로 다른 색깔로 coloring 되어야 한다. 이러한 사실에 기초해 FixedNodeList에 저장된 노드, 즉 이미 색 깔이 결정된 노드에서 출발해 실선 형태의 예지를 따라 가면서 실선 예지간의 노드에는 서로 다른 색 을 할당한다.

이때 어떤 노드에서 서로 다른 색깔을 할당하는것 이 불가능할 경우가 발생하면 그노드에 overlap 이 라고 표시한후 coloring을 중지하고 FixedNodeList 의 다른 노드로 부터 coloring을 다시 수행한다. 더 이상 색깔을 할당할 노드가 없거나 FixedNodeList 에 노드가 없을때 다른 신호선간의 coloring을 끝낸 다.

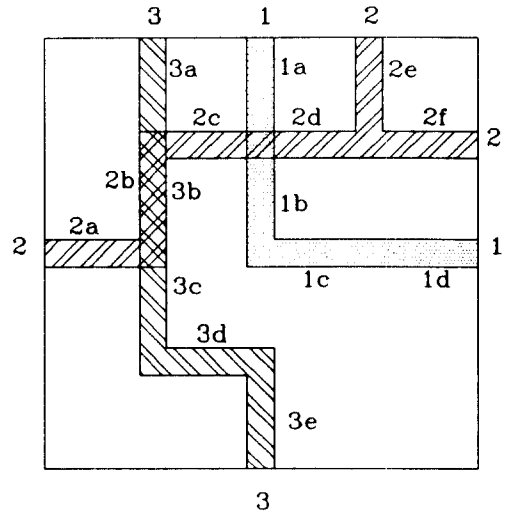


그림 7. 단일층에서의 미로배선 예
Fig. 7. Example of layerless maze routing.

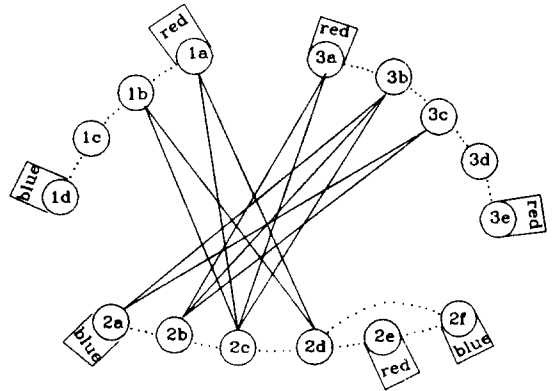


그림 8. 그림 7에 대한 교차 그래프
Fig. 8. Intersection graph for Fig. 7.

앞에서 설명한 coloring 과정에서는 다른 신호선과 서로 교차 관계가 없는 노드들, 즉 같은 신호선간에 교차 관계가 있는 노드들은 coloring이 되지 않는다. 따라서 같은 신호선간에 교차 관계가 있는 노드들에 대하여 via 수의 최소화를 목적함수로 하여 coloring을 수행한다. Coloring이 되지 않은 어떤 노드가 있을 때 그 노드의 색깔은 양쪽에 인접한 노드들의 색 깔이 결정되어 있으면 via 수를 최소화 하는 색깔을 결정할 수 있다. 즉 coloring 되지 않은 노드의 색깔을 결정할때 주변상황을 recursive 하게 탐색하여 탐 색한 결과를 바탕으로 coloring을 하는 방법을 사용

한다. 그림 9는 그림 8의 교차 그래프에 대하여 coloring을 수행한 결과를 나타내며 그림 10은 coloring한 결과를 배선영역상에 레이아웃으로 표현한 그림이다.

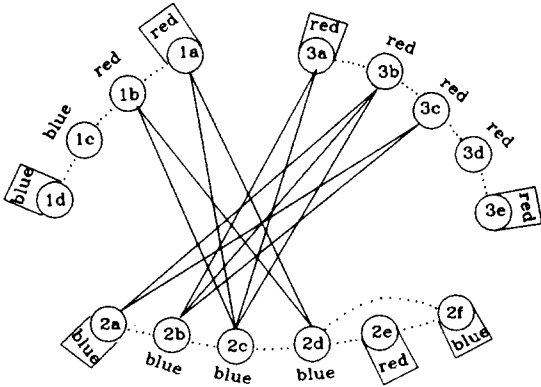


그림 9. 그림 8의 coloring 결과
Fig. 9. Coloring result for Fig. 8.

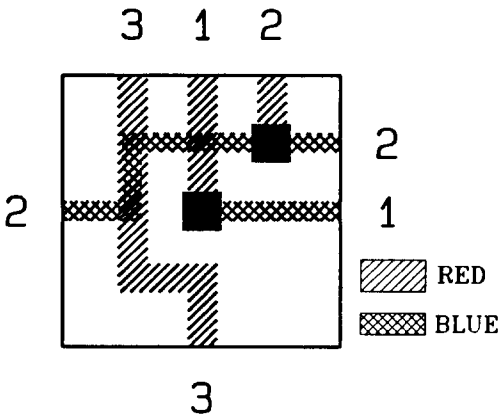


그림 10. 그림 9의 coloring 결과에 대한 레이아웃 표현
Fig. 10. Layout representation for coloring result of Fig. 9.

3. 재배선(rip-up and reroute)

단일층에서의 미로배선시 특정 그리드에 신호선이 밀집한경우 coloring 과정에서 overlap이 발생하는데 이러한 overlap이 일어난 신호선에 대해서 재배선을 하기 위하여 먼저 rip-up set을 구성한다. 즉 coloring 과정에서 overlap이 발생하여 색깔이 할당되지 않은 신호선들을 WireList에서 제거한후 rip-

up set에 저장하고, overlap된 신호선들과 교차 관계에 있는 신호선들도 WireList에서 제거한후 rip-up set에 저장한다. 구성된 rip-up set의 신호선들을 아래의 비용값이 큰순으로 정렬한다.

$$\text{비용값} = \alpha * \text{overlap된 부분의 갯수} + \beta * \text{via의 갯수} + \gamma * \text{wire segment의 총길이}$$

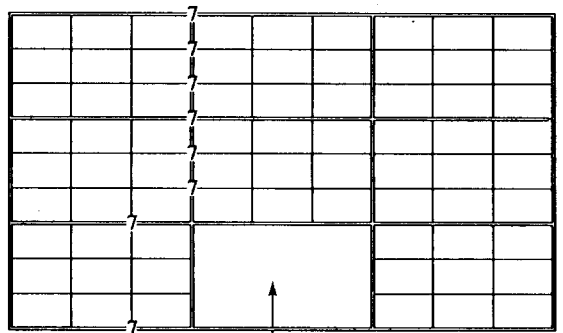
단 α, β, γ 는 $\alpha > \beta > \gamma$ 인 상수

재정렬된 순서에 의하여 단일층에서의 미로배선과 coloring에 의한 층 할당을 거쳐 재배선을 수행한다. 만일 이때에도 overlap이 발생하면 같은 방법으로 다시 재배선을 수행한다. 재배선을 반복하는 횟수는 파라미터 화일에 저장되어있다.

V. 상세배선이 실패한 부 배선영역의 핀 재할당후 재상세배선

1. 핀 우회 할당

상세배선이 실패한 부 배선영역 내의 핀들을 모두 제거한후 그림 11에서와 같이 상세배선이 실패한 부 배선영역을 우회하도록 핀들을 재할당한다.



상세배선이 실패한 부 배선영역

그림 11. 핀 우회 할당

Fig. 11. Reassignment of pin after detour.

2. 핀 이동

재할당된 핀들중에서 수직제한을 만족시키지 못하는 핀들을 수직제한을 만족시키기 위하여 이동한다. 그림 12(a)의 경우에서 처럼 3번 핀이 15번 핀과 수직제한이 발생하여 3번 핀이 상세배선 되지 않는 경우에는 그림 12(b)에서 처럼 3번 핀을 이동할수 있는지의 여부를 조사한후, 핀 이동을 하여 수직 제한을 해소시킨다.

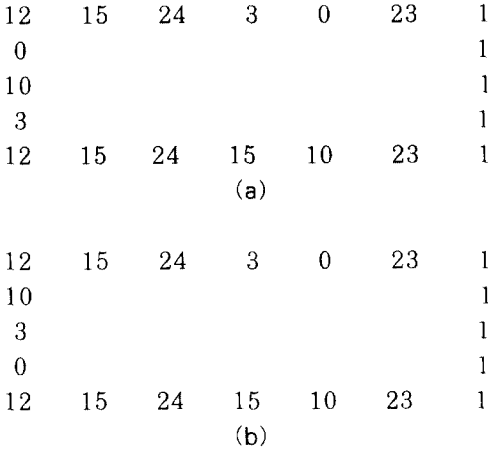


그림 12. 핀 이동 과정
(a) 수직제한 발생
(b) 핀 이동

Fig. 12. Process of pin movement.

- (a) vertical constraint.
- (b) pin movement.

3. 재상세배선 (redetailed routing)

신호선 우회 할당과 핀 이동을 완료한후 상세배선이 실패했던 부 배선영역에 대해서 상세배선을 다시 수행한다. 각 부 배선영역에 대해 상세배선이 모두 성공하면, 각각의 부 배선영역을 결합하여 원래의 스위치박스 배선영역으로 복원시킴으로써 전체 배선을 완료한다.

VI. 실험 및 고찰

본 논문에서 제안한 알고리즘의 유용성을 보이기 위해 SUN4/330 UNIX 상에서 C 언어로 실현하였으며, 배선결과를 X11 release4 그래픽스 라이브러리를 사용하여 MOTIF상에서 그래픽으로 출력하였다. 또한 배선결과를 HP7596B 플로터와 HP LaserJet III 프린터로 출력할수 있도록 하였다. 본 논문에서 제안한 알고리즘을 C 언어로 실현한 분량은 약 9000 라인 정도 이고, 배선결과를 MOTIF상에서 그래픽으로 출력하는 부분과 HP7596B 플로터와 HP LaserJet III 프린터로 출력하는 부분은 약 5000 라인 정도이다. 표 1은 Burstein과 Pelavin이 [5] 에서 제시한 difficult switchbox benchmark 예제에 대해서 본 논문에서 제안한 CONQUEROR로 자동 배선한 결과를 다른 스위치박스 배선기들과 비교하였다. 여기서 BEAVER의 배선결과는 변상의 핀들이 고정된 층에 있을때의 결과로 나타내었다. 표 1 에서 나타난 바와 같이 기존에 발

표된 다른 스위치박스 배선기들보다 비교적 빠른 시간 내에 우수한 배선 결과를 얻을수 있었다. 표 2 에서는 Luk이 [7] 에서 제시한 세계의 switchbox benchmark 예제에 대해서도 역시 기존에 발표된 다른 스위치박스 배선기들과 배선결과를 비교해 나타내었다. 표 2 에서 나타난 바와 같이 다른 switchbox benchmark 예제들에 대해서도, 기존에 발표된 다른 스위치박스 배선기들보다 우수한 배선결과를 얻을수 있었다. 표 2의 terminal intensive 예제에 대한 WEAVER의 배선결과가 틀리게 발표 되었기 때문에 정확한 배선결과를 괄호밖에, 틀리게 발표된 배선결과는 괄호안에 표시하였다. 그림 13은 Burstein의 difficult switchbox 예제, 그림 14는 Luk의 terminal intensive switchbox 예제, 그림 15는 Luk의 dense switchbox 예제, 그림 16은 modified dense switchbox 예제에 대하여 본 논문에서 제안한 CONQUEROR로 수행한 배선결과를 나타내었다.

표 1. Burstein 의 difficult switchbox에 대한 배선결과와 비교

Table 1. Routing comparison of burstein's difficult switchbox.

Router	#row	#col	#vias	#length	cpu time(seconds)
LUK'S GREEDY	16	23	58	577	?
WEAVER	15	23	41	531	1390s on VAX 11/850
MIGHTY	15	22	39	541	4s on VAX 11/850
BEAVER	15	22	42	536	1s on SUN3
SILK	15	22	36	528	69s on VAX 8550
PACKER	15	22	43	541	1400s on Apollo DN4000
CARIOCA	15	23	43	535	54s on SUN3/160
CONQUEROR	15	22	36	527	8s on SUN4/330

표 2. 다른 switchbox benchmark 예제에 대한 배선결과와 비교

Table2. Routing comparison of other switchbox benchmarks.

Example	Router	#row	#col	#vias	#length
terminal intensive	Luk	16	23	68	632
	WEAVER	16	23	51(49)	615
	MIGHTY	16	23	50	629
	BEAVER	16	23	53	632
	SILK	16	23	49	616
	CONQUEROR	16	23	48	622
dense	Luk	18	16	36	527
	WEAVER	18	16	31	517
	MIGHTY	18	16	32	530
	BEAVER	18	16	32	529
	SILK	17	16	29	516
	CONQUEROR	17	16	29	516
modified dense	Luk	17	16	29	510
	WEAVER	17	16	29	510
	MIGHTY	17	16	29	510
	BEAVER	17	16	29	510
	SILK	17	16	29	510
	CONQUEROR	17	16	29	510

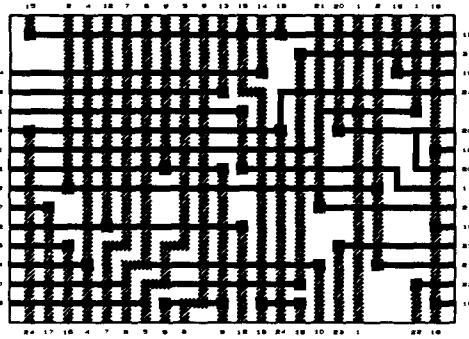


그림 13. Burstein의 difficult switchbox에 대한 CONQUEROR의 배선결과
 Fig. 13. Routing result of CONQUEROR for a burstein's difficult switchbox.

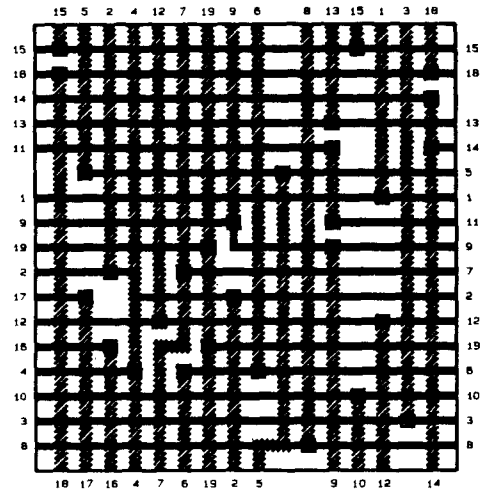


그림 16. Modified dense switchbox에 대한 CONQUEROR의 배선결과
 Fig. 16. Routing result of CONQUEROR for a modified dense switchbox.

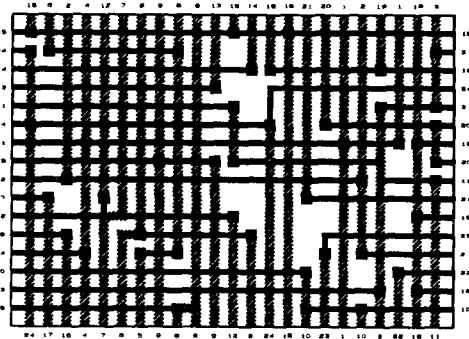


그림 14. Luk의 terminal intensive switchbox에 대한 CONQUEROR의 배선결과
 Fig. 14. Routing result of CONQUEROR for luk's terminal intensive switchbox.

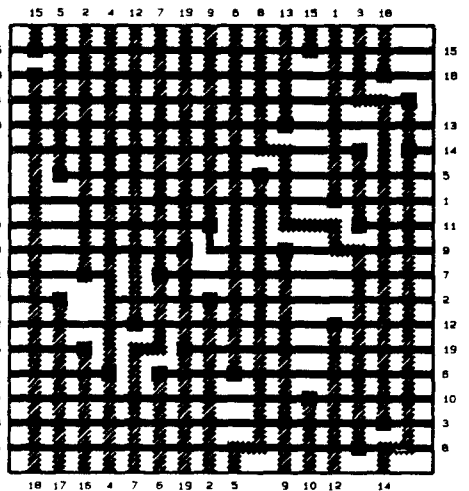


그림 15. Luk의 dense switchbox에 대한 CONQUEROR의 배선결과
 Fig. 15. Routing result of CONQUEROR for a luk's dense switchbox.

Ⅶ. 결론

본 논문에서는 대규모 스위치박스 배선영역을 divide-and-conquer 기법에 의해 여러개의 소규모 부 배선영역으로 분할하여 처리하는 스위치박스 배선 알고리즘을 제안하였다. 부 배선영역을 상세배선하는 알고리즘은 배선의 효율을 높이기 위하여 단일층에서의 미로배선, coloring에 의한 층 할당, 재배선의 3 단계로 나누어 수행하였다. 제안된 알고리즘을 프로그래밍 하여 여러 예제들에 적용한 결과 매우 좋은 결과를 얻을수 있었다. 따라서 칩 면적이나 성능을 향상시킬수 있는 많은 장점을 가진 빌딩블록 레이아웃 설계방식으로 칩을 설계할 경우, 필연적으로 발생하는 스위치박스 배선영역에 대하여 본 논문에서 제안한 CONQUEROR를 사용할 경우, 스위치박스 배선영역을 효과적으로 자동 배선 할수 있을 것으로 기대된다.

현재 본 논문에서 제안한 알고리즘을 빌딩블록 레이아웃 설계뿐만 아니라 앞으로는 PCB CAD 시스템에도 사용될수 있도록 하기 위하여 여러 형태의 배선 영역과 다층 배선에도 적용될수 있도록 하는 연구가 계속 되고있다.

參考文獻

[1] C. P. Hsu, "General river routing

algorithm.” in Proc. 20th Design Automation Conf., pp. 578-583, 1983.

[2] R. K. McGehee, “A practical moat router,” in Proc. 24th Design Automation Conf., pp. 216-222, 1987

[3] 이승호, 정정화, “새로운 Over-the-Cell 배선 시스템.” 대한전자공학회 논문지 제27권 제11호 pp.135-143, 11월 1990.

[4] A. Sangiovanni-Vincentelli, M. Santomauro, J. Reed, “A New Symbolic Channel Router : YACR2.” *IEEE Trans. on Computer-Aided Design*, Vol.CAD-4 No.3 pp.208-219, July, 1985.

[5] M. Burstein and R. Pelavin, “Hierarchical Wire Routing.” *IEEE Trans. on Computer-Aided Design*, Vol. CAD-2 No.4 pp.223-234, 1983.

[6] G. Hamachi and J. Ousterhout, “A Switchbox Router with Obstacle Avoidance”, in Proc. 21th Design Automation Conf. pp.173-179, 1984.

[7] W. K. Luk, “A Greedy Switchbox Router,” *Integration, the VLSI journal*3, pp.129-149, 1985.

[8] R. Joobbani and D.P. Siewiorek, “WEAVER: A Knowledge Based Routing Expert,” *IEEE Design & Test* VOL.3 pp.12-23, Feb. 1986.

[9] H. Shin and A. Sangiovanni-Vincentelli, “MIGHTY : A Rip-Up and Reroute Detailed Router.” in Proc. *Int. Conf. CAD*, pp.2-5, 1986.

[10] J. P. Cohoon and P. L. Heck, “BEAVER : A Computational-Geometry Based Tool for Switchbox Routing.” *IEEE Trans. on Computer Aided Design* Vol.7 pp.684-697, JUNE, 1988.

[11] Young-Long Lin, Yu-Chin Hsu, Fur-Shing Tsai “SILK : A Simulated Evolution Router.” *IEEE Trans. on Computer-Aided Design*, VOL.8 NO.10 pp.1108-1114, Oct. 1989.

[12] S. H. Gerez and O. E. Herrmann, “Switchbox Routing by Stepwise Reshaping.” *IEEE Trans. on Computer-Aided Design*, VOL.8, NO 12, pp.1250-1361, Dec. 1989.

[13] P. F. Dubios, A. P. Puissochet, A. M. Tagant, “A General and Flexible Switchbox Router : CARIOCA ,” *IEEE Trans. on Computer-Aided Design*, Vol.9 No.12, pp.1307-1317, DEC. 1990.

[14] C.Y. Lee, “An Algorithm for Path Connections and its Application,” *IRE Trans. on Electronic Computers*, Vol. EC-10, pp.346-365, Sept. 1961.

[15] 이승호, 김종훈, 강병익, 정정화, “BUFFALO : 계층적 배치설계와 Divide-and-Conquer 배선설계에 의한 빌딩블록 레이아웃 시스템”, 대한전자공학회 추계종합학술대회 논문집 제 14권 제2호 pp.573-576, 11월 1991.

著 者 紹 介

李昇昊(正會員) 第 27 卷 第 11 號 參照
 현재 한양대학교 대학원 전자공학과 박사과정.

鄭正和(正會員) 第 27 卷 第 11 號 參照
 현재 한양대학교 전자공학과 교수.