

논-그리드와 그리드 혼합 방식을 이용한 PCB 자동 배선 시스템 개발

최 영 규[†] 이 천 희^{††}

요 약

자동 배선 시스템의 배선영역 모델링 방법은 논-그리드(non-grid)와 그리드 방식을 사용하고 있다. 본 논문에서는 라우팅 속도를 개선하기 위해 논-그리드와 그리드 방식을 혼합하여 자동 배선 시스템을 개발 하였다. 그리드방식은 PCB 상에 전기적, 물리적 요소들이 적다 할지라도 보드와 그리드의 크기에 제약을 받기 때문에 메모리가 많아지게 되어 자동 배선 속도를 감소시키는 단점을 가지고 있고, 논-그리드 방식은 shape-based 방식의 영역 처리방식을 사용하므로 메모리를 상당히 적게 소요된다. 그러므로 본 논문에서는 논-그리드 방식과 그리드 방식을 적용하여 초기 배선시에는 전자를 적용해 속도를 개선시켰고, 초기 배선시 연결되지않은 배선은 후자를 적용해서 완전한 배선이 이루어 질 수 있도록 하였다. 따라서 본 시스템을 이와 같은 방식을 이용하여 IBM 486DX2-66 컴퓨터의 Windows NT 환경 하에서 Borland C++로 개발하였다.

A study on the development of PCB automatic routing system using the mixing method of non-grid and grid

Young-gyu Choi[†] and Cheon-hee Yi^{††}

ABSTRACT

Non-grid and grid method are used for modeling the routing region of the automatic routing system. In this study , we develop the automatic routing system by mixing the methods of non-grid and grid to improve the speed of routing. Grid method has a demerit which decreases the automatic routing speed because it is required a lot of memory by the limitation of the size of board and grid, although the electrical and physical elements are insufficient on the PCB, But non-grid is spent gurite less memory than grid method by using a shape-based patterns. Therefore, we used the methods of non-grid and grid altogether in this paper. In the former method, it is attributed to the improvement of speed, the latter one is applied only the connection of the failed routing in the former one, and it performs the complete automatic routing. This system was developed with C++ language under the Windows NT environment of IBM 486DX2-66 Computer.

1. 서 론

최근 전자기술의 급속한 발전으로 소형 및 다기능화가 이루어짐에 따라 PCB(Printed Circuit Board) 설계의 복잡도가 증가하므로 설계 기간의 단축이 상당한 요인으로 등장하게 되었다. 현

존하는 대부분의 배선 시스템은 한가지 방법에 대한 결점을 보완하기 위하여 미로 운행법 [6], Lee 알고리즘[3], 선분 탐색법 [4], 채널법 [5], 그리고 신호-열(signal=row)법[9,10]등과 같은 몇가지 배선법으로 구성되어 있다.

PCB 설계 과정에서 가장 중요한 자동 배선 알고리즘의 가장 핵심적인 요소는 신호의 상호 전달 매체로서 부품 핀간의 배선규칙인 PPD (pad pad distance)와 PTD(pad pattern dis-

† 정 회 원 : 충주산업대학교 컴퓨터공학과 전임강사

†† 정 회 원 : 청주대학교 전자공학과 교수

논문접수 : 1995년 3월 13일, 심사완료 : 1995년 7월 5일.

tance), TTD(pattern pattern distance)의 3가지의 기본적인 파라미터(parameter) 값에 의해 보드상에 제시되는 네트들을 전기적 오류없이 물리적으로 각 요소들 사이에 최소 인접 거리를 유지하도록 동박을 사용하여 연결시키는 배선 방법이다. PCB 자동 배선은 배선영역(routing region), 금지영역(forbidden region), 배선층(routing layer)에 대한 모델링을 수행하여 배선을 하게 된다.

배선영역은 실제로 볼 수 있는 가시적인 파라미터로 가시적인 객체(object)들을 프로그래밍화하고 실제 가시적인 형태에서 얻을 수 있는 정보 형태들은 프로그래밍되어 있는 데이터로부터 바로 얻을 수 있어야 한다. 따라서 주어진 영역을 프로그래밍화 한다는 것은 이 영역을 모델링하는 것과 같다. 여기서 영역 모델링이란 영역을 어떠한 형태로 분석해서 이를 데이터화하여 필요한 정보를 도출해 낼 수 있도록 하는 것이다. 따라서 본 논문에서는 이와 같은 배선 영역 모델링을 위하여 논-그리드(non-grid)와 그리드 방식을 사용한다. 그리드 방식은 메모리 맵을 사용하는 방식으로 Maze 배선기와 선분의 전파를 이용한 선분 탐색 방식, 하이타워 배선기(Hightower router)[1, 2]등으로 구분할 수 있다. 이와같은 배선 영역 모델링 방법은 배선 영역을 바둑판 모양으로 $m \times n$ 배열로 분리하여 배열 변수에 사각형 영역의 정보를 저장하기 때문에 m 과 n 을 크게 할 수록 더욱 정밀한 정보를 저장할 수 있다. 그러나 이것은 소요 메모리의 증가와 경로 탐색시 필요한 메모리 참조에 많은 제약이 따르므로 본 논문에서는 논-그리드 알고리즘으로 배선되지 않는 네트에 대해 개선된 그리드 알고리즘을 적용하여 자동 배선이 수행될 수 있도록 하였다.

논-그리드 배선 방식은 위에서 설명한 그리드 배선 방식의 여러가지 문제점을 해결하기 위해 사용되는 방식이다. 이 배선 방식은 배선시 그리드를 발생시키지 않기 때문에 단위 셀을 생성시키지 않는다. 이와같은 이유로 단위셀에 의한 영역 처리를 수행하지 않고, Object based (shape based)의 영역 처리 방식을 적용한 패턴 배선기를 사용하므로써 그리드 방식과 같은 메모리 증가 문제를 해소할 수 있게된다. 따라서 본 논문

에서는 이와 같은 설계규칙을 근간으로 하여 논-그리드와 그리드 배선 영역 분할 방법을 혼합 사용하여 PCB 자동 배선 시스템을 개발하였다.

2. 알고리즘 및 데이터포맷 고찰

기본적인 설계 규칙을 근간으로하여 PCB 자동 배선 설계 시스템을 구축하기 위해 기존에 발표된 논-그리드 알고리즘과 본 논문에서 제시한 개선된 그리드 알고리즘을 사용하여 혼합배선기를 개발하였다. 본 논문에서 사용된 논-그리드 배선 알고리즘은 메모리 배선알고리즘 Shape-Based알고리즘[9], Push 알고리즘과 4 배선알고리즘을 사용하였고, 그리드 배선알고리즘은 개선된 그리드 알고리즘을 적용하였다.

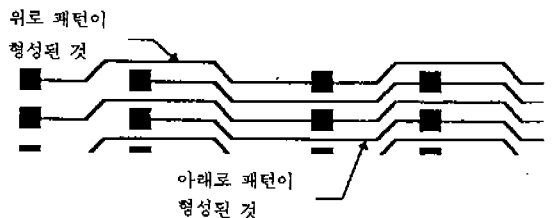
사용된 알고리즘과 데이터포맷의 내용을 기술하면 다음과 같다.

2.1 혼합 배선기에 사용된 알고리즘

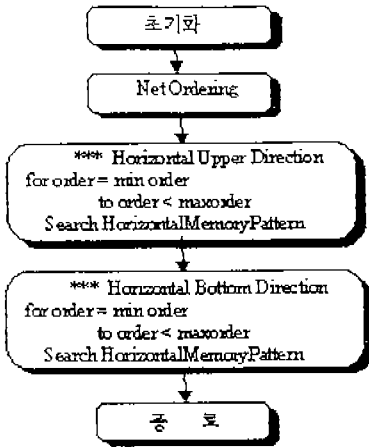
2.1.1 논그리드 배선알고리즘

2.1.1.1 메모리 배선기 (Memory Router)

메모리 배선기는 논-그리드 방식으로 네트로 연결된 패드의 위치가 수직 또는 수평의 위치에 있을 경우 패드간 연결되는 규칙은 패드의 일정한 거리 및 방향이 유지되어야 하며, 이때 패드 사이에 장애물이 존재하는 특징을 가지고 있다. 이와같은 특징을 갖는 네트들을 배선하려면 네트의 순서가 특정한 방향성을 갖도록 패드간의 연결 우선 순위를 좌측 상단에서 아래쪽 방향으로 네트 순서가 이루어 지도록 한다. 이때 배선되는 네트들의 수평 (수직) 방향의 패턴 형태는 두가지 형태로서 하나는 위(좌)로 패턴이 형성되는 것과 다른 하나는 아래(우)로 패턴이 형성되는



(그림 1) 메모리 배선 패턴 형태
(Fig. 1) Memory routing pattern type



(그림 2) 수평 메모리 라우터의 흐름도
(Fig. 2) Flowchart of horizontal memory router

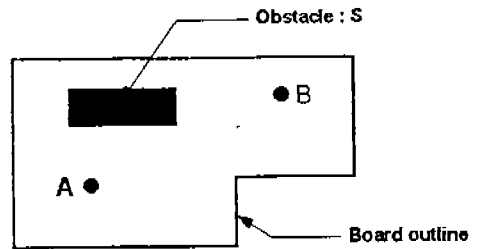
것으로써 수직(수평) 형태의 메모리 패턴을 생성하게 된다. 따라서 수평 형태의 메모리 패턴 생성 방법은 (그림 1)과 같고, 이에 대한 메모리 패턴 배선기의 처리 흐름도는 (그림 2)와 같다.

2.1.1.2 Shape-based 패턴 배선기

이 배선기법은 기존의 그리드 매트릭스 메모리 맵을 이용한 배선방식에서 탈피한 논-그리드 shape-based 방식을 사용하는 기법이다. 논-그리드 방식은 그리드 매트릭스 메모리 맵 방식에서 메모리 맵 형식을 사용하지 않고 shape-based를 사용한다. Shape-based 패턴 배선기는 배선기의 기본 구조를 유지하면서 빠른 패턴의 생성과 shape based 형태의 경로 검색[7,11]을 사용한 논-그리드 형식으로 배선을 수행한다. 패턴 배선기는 shape-based, 패턴생성, 경로검색인 3가지 부분으로 나누어 진다. Shape-based는 기존의 배선에 사용한 문제중 여러가지의 장애물들을 표현하는데 사용 기준에 따라 분류하게 된다. 메모리 맵은 매트릭스화된 메모리 맵에서 각 셀의 요소들에 위치하는 상황을 참조하여 배선을 수행하는 반면에, Shape-based는 메모리 맵을 참조하지 않고 직접 각각의 장애물들을 특정한 shape로 분석하여 이 shape를 참조하게 된다. 패턴생성은 패턴 배선기에 사용되는 패턴들을 생성해주는 기능을 가진다. 이것은 단순한 패턴으로 부터 복잡한 패턴으로 순차적으로 생성해 주

며, Shape의 기본 형태는 I, L, U형태[8]를 사용한다. I 형태 패턴은 수직 또는 수평 배선 네트에 사용되는 패턴을 만들기 위해 이용되는 형태로서 핀간 장애물이 없을 경우 배선이 가능하다, 만약 핀사이에 장애물이 있을 경우에는 메모리 패턴으로 처리 되도록 한다. 그러나 패턴이 형성되지 않을 경우에는 U 형태 패턴으로 처리되는 경우도 있다.

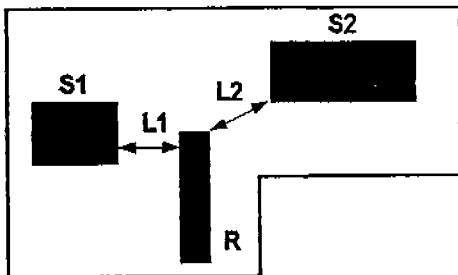
L 형태 패턴은 네트 각각의 핀이 수직 및 수평이 아닌 대각선상에 위치할 경우에 적용되는 형태이다. 이 형태는 2개의 패턴을 생성하며 45도를 적용할 경우에는 3개의 패턴을 생성하게 된다. U 형태 패턴은 수직, 수평 또는 사선형태의 네트를 연결하는데 적용된다. 이것은 3개의 패턴을 이용하여 패턴을 생성시키며, 중간 패턴이 수직 또는 수평으로 평행 이동하면서 패턴을 생성하게 된다. 그리고 경로검색은 생성된 패턴들을 기존의 패드와 패턴들간의 배선 위치 검색을 수행하여 생성된 패턴에 애러가 있는지를 검사하는 과정이다. 이것은 크게 패드와 패턴간의 위치 검색과 패턴과 패턴 간 위치 검색 함수로 구성된다. 패드와 패턴간의 위치 검색 함수, ROUTERPAD *ROUTER TYPE:: DistCalPad(float x1,float y1,float x2,float y2,int layer, ROUTER NET *n)는 주어진 영역(x1, y1, x2, y2)에서 기존의 배치된 패드와 비아의 위치 검색을 수행하여 만일 배선 위치 검색 애러가 발생할 경우 해당 패드의 포인터를 반환하였다. 여기서 층(layer)은 주어진 영역의 배선 층을 나타내며, n은 주어진 영역의 네트 포인터 정보를 처리한다. 또한 패턴과 패턴 간의 위치 검색 함수, ROUTERPATTERN *ROUTERTYPE::DistCal-



(그림 3) 논- 그리드의 배선 영역
(Fig. 3) Routing region of non-grid

Pattern(float x1,float y1,float x2,float y2,int layer,ROUTERNET *n)는 주어진 영역(x1, y1, x2, y2)와 기존에 생성된 패턴과의 배선 위치 검색을 수행하는 함수로 배선 층과 n이라는 부가적인 인수를 가지고 패턴간의 위치 검색을 수행하게 된다. 예를 들어, Shape based(object based) 영역 처리 방식을 아래에 주어진 영역을 가지고 설명해 보면 다음과 같다.

이것은 그리드 방식과 같이 사각형의 배선 영역 셀을 만들지않고 배선 영역을 board outline 내부로 한정한다. 위 (그림 3)에서 두점 A 와 B사이에서 연결 요구가 발생한다면 이 두점을 연결하는 경로 탐색은 영역 내부에 S라는 장애물 (obstacle)이 있을 경우 이 장애물의 존재 여부를 발견하고, 이를 배선에 적용시킬 수 있도록 shape based 알고리즘을 사용한다. Shape based 는 어떤 모델링된 객체들을 shape라는 구조로 처리하기 위해 배선에 필요한 shape를 장애물 shape와 routing shape 2 가지로 크게 나누어 처리한다. 이것은 (그림 4)에서와 같이 S1과 S2 의 장애물 shape와 R의 routing shape로 나타낼 수 있다.



(그림 4) Shape 모델링 방법
(Fig. 4) Shape modeling method

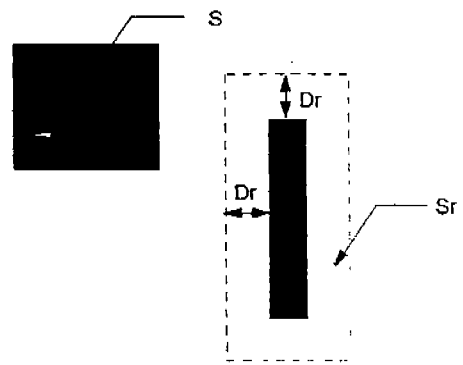
L1이 S1과 R간의 최소거리, L2는 S2와 R간의 최단거리라 할 경우 R shape가 배선에 수용될 수 있는지는 L1과 L2의 거리가 배선 규칙에 위배되는지를 검색하면 된다. 이것은 두 shape 간의 배선 규칙을 Dr이라고 할때 배선을 수용하기 위해서는 다음 조건을 만족 해야한다.

$$L1 \geq Dr \text{ AND } L2 \geq Dr$$

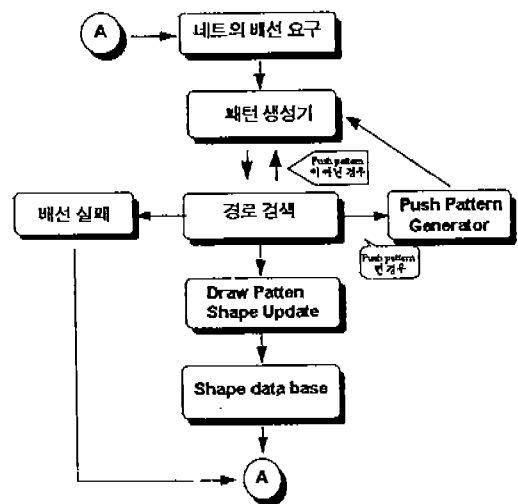
사각형 shape에 대한 효율적인 배선규칙은 두 개의 shape가 배선 규칙에 위배되는지를 검사하

기 위해 두사각형의 최단거리를 구하지 않고 (그림 5)에서와 같이 R 영역을 Dr 만큼 확장하여 Sr 영역을 생성하므로써 이 영역이 S영역과 overlapping 되지 않으면 S와 R 영역간의 최소 거리가 Dr보다 큰 값을 갖게 됨을 알 수 있다.

따라서 shape based의 경로 탐색은 shape based 영역 처리 방식을 사용하여 이루어지게 된다. Shape based 의 가장 기본이되는 경로 탐색은 패턴 생성 방식을 이용해 주어진 배선 영역의 장애물 shape 상태를 확인한 후 패턴을 생성하여 shape based 영역을 처리하고, 이에 대한 배선 규칙을 검사하게 된다. 패턴 생성은 가장 기본적인 패턴인 I, U, L 패턴 이외에 패턴 생성



(그림 5) Shape의 배선규칙 검사 방법
(Fig. 5) Routing rule check method of the shape

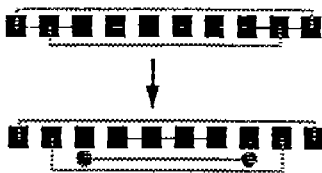


(그림 6) Shape based 패턴 라우터의 흐름도
(Fig. 6) Flowchart of the shape based pattern router

기로 부터 L+I, L+L, L+L+L 응용 shape 들을 생성하도록 처리하였다. Shape-based 배선기는 패턴 생성기의 능력과 shape 배선 검사기의 속도에 따라 그 성능이 좌우되므로 본 논문에서는 패턴 생성기의 성능개선에 중점을 두었다. 이와 같은 shape based 배선기의 전체적인 처리 과정을 흐름도로 기술해 보면 (그림 6)과 같다. 그림에서 A는 배선요구 네트의 끝인지 아니면 다음네트를 구하는지를 판단하는데 만일 네트의 끝이라면 다음 단계로 넘어가고, 그렇지 않다면 상부에 표시된 A부분으로 돌아와 다시 네트의 배선요구를 하게 된다.

2.1.1.3 푸쉬 패턴 생성(Push pattern generation)

배선시 특정 패턴이 장애가 될 경우 장애 패턴들을 일정한 간격으로 이동시키므로써 생기는 여분 공간을 이용하여 높은 배선율을 얻을 수 있도록 하는 방식이다. 이 배선 방식은 패턴들을 상하 좌우로 이동시켜 배선공간을 확보하여 배선되지 않은 네트를 (그림 7)과 같이 배선이 이루어질 수 있도록 하는 방법이다.



(그림 7) 푸쉬 패턴 생성 (Fig. 7) Push pattern generation

이와 같은 푸쉬 배선은 다음과 같은 함수를 이용하여 처리하게 된다.

```

c = PatternRouterPush( net - parameters );
if ( c ) { /* 배선 성공시 생성된 패턴을 패턴 큐에 푸쉬 */
    for(i=0; i<=c; i++)
        PushPattern( net - parameters );
}
    
```

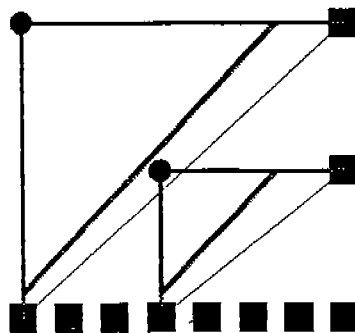
2.1.1.4 비아 최소화(Via Minimization)

비아 최소화는 비아 수를 최적화하는 기능으로서 단층 재-라우팅(rerouting)과 비아 최적화 기법이 이용된다. 단층 재-배선 기법은 단층 배선에서 비아가 필요없기 때문에 2 층으로 배선이 끝난 PCB를 단층으로 재배선하여 비아 최

소화를 얻는 방법이다. 즉, 현재 배선된 경로에 비아가 존재할 경우 비아를 사용하지 않는 다른 경로를 찾아 배선을 수행한다. 또한 이 기법은 비아 형태를 보고 최적화를 수행하는 기법으로서 패턴에 형성된 비아는 다음과 같은 2가지 형태로 구성된다. 첫번째 형태는 패턴 당 하나의 비아가 존재하는 경우로서 패턴 세그먼트 전후의 층이 서로 다른 경우로 하나의 비아가 발생하는 1 패턴 1비아 형태이다. 두번째 패턴 형태는 2개의 비아가 존재하는 경우로서 패턴 세그먼트 전후의 층이 다른경우로 2개의 비아가 발생하는 1 패턴 2 비아 형태이다. 1 패턴 2 비아는 비아 우선 처리 방법이 있어 1 패턴 1 비아 보다 우선순위가 높다. 즉, 1패턴 1비아는 패턴을 다른 층으로 변경할 경우 다른 하나의 비아가 발생하는 경우도 존재한다. 그러나 1패턴 2비아는 패턴 세그먼트 전후의 층에 연결된 패턴과 같은 층으로 변경할 경우 2개의 비아가 줄어들게 됨으로써 1패턴 2비아를 우선 검색한 후 1패턴 1비아를 검색하므로써 비아 최적화를 수행하게 된다.

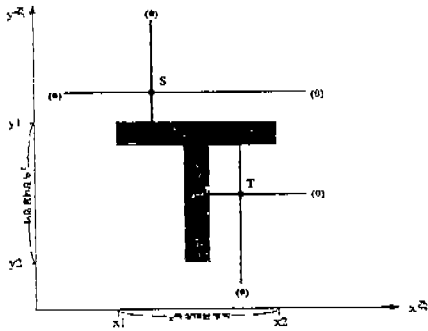
2.1.1.5 45도 배선 배선기

L Shape 패턴은 각 핀간의 네트가 수직 또는 수평이 아닌 대각선 상에 위치할 경우 적용되는 형태이다. L shape는 2개의 패턴으로 형성되나 45도를 적용할 경우는 (그림 8) 과 같이 3개의 패턴으로 형성된다.

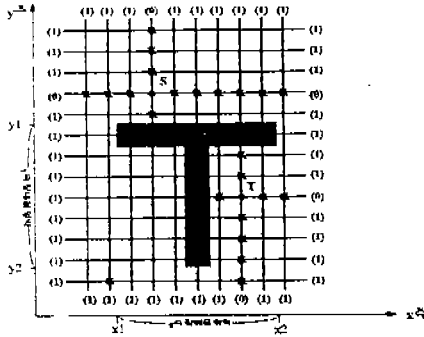


(그림 8) 45도 배선 예 (Fig. 8) 45 degree routing example

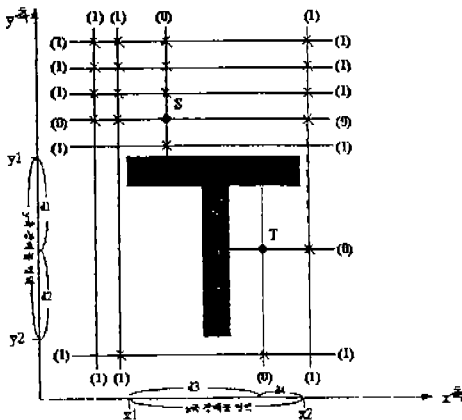
이와 같은 45도 배선은 다음과 같은 함수를 이용하여 처리하게 된다.



(a) 초기 출발점과 목적지에 라인생성
(a) Line generation from starting point to destination



(b) 그리드를 적용한 초기 알고리즘
(b) Initial algorithm using grid



(c) 장애물 영역과 인접한 같은 형태의 그리드를 제거한 경우.
(c) Deleted grid lines around a obstacle

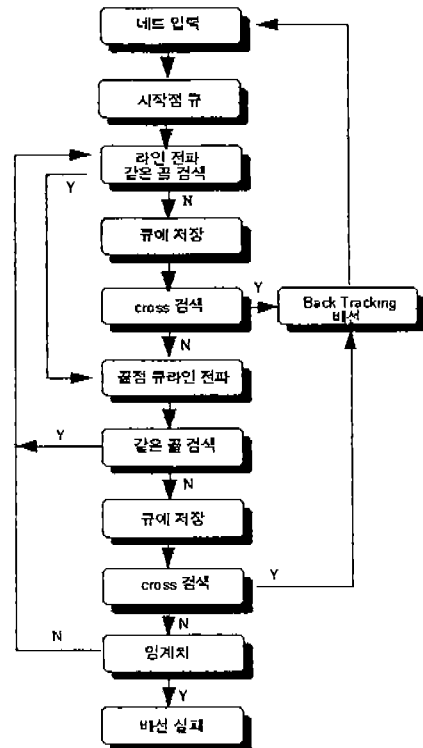
(그림 9) 개선된 그리드방식 알고리즘의 단순 응용
(Fig. 9) Simple application of improved grid method algorithm

```
void ROUTERTYPE::Route45(HWND hwnd)
{
    ROUTERNET *net;
    n=net;
    while(n) {
        Route45( hwnd, n );
        n=n->next;
    }
}
```

2.1.2 그리드 알고리즘

2.1.2.1 개선된 그리드 배선기

개선된 본 알고리즘은 Lee 알고리즘과 같이 깊이 우선탐색이다. 이 방법의 일반적인 개념 보드에 그리드를 생성하고 생성된 그리드와 장애물이 직각으로 만나는 그리드를 제거한 후에 나머지 그리드를 이용하여 backtracking 수행하여 비아 수와 배선 길이가 최소로 되는 경로를 찾아내는 방법이다. 탐색 절차는 출발점(S)와 목적지(T)에서 (그림 9(a))와 같이 선분을 생성하고, 0을 삽입한 후 (그림 9(b))와 같이 그리드 값을



(그림 10) 개선된 그리드방식의 흐름도
(Fig. 10) Flowchart of improved grid method

이용하여 생성되는 선분에는 1을 삽입하고 장애물 영역과 수직으로 만나는 x축과 y축 범위내의 1로 생성된 선분을 제거하면 (그림 9(c))와 같이 배선경로가 발생한다. 생성되는 경로에 대해 back tracking을 수행하여 비 수가 최소인 경로를 탐색 하여 최종배선을 수행하도록 하였다. 이 경우 N * N 크기의 격자에 대해 복잡도가 O(N²)만큼 커지게 되는 단점을 가지고 있으나 개선된 그리드 알고리즘은 O(N²-(제거된 선분))만큼 메모리가 감소하는 장점을 가지고 있고, 배선경로를 찾기위한 반복루프를 수행하지 않으므로 배선 속도를 향상시킬 수 있는 장점을 가지고 있다. (그림 10)은 개선된 알고리즘의 전체적인 흐름도를 나타내었다. 본 알고리즘은 shape-based 패턴 배선기에 의해 패턴이 생성되지 않은 네트 배선시에만 적용하도록 하였다.

2.2 배선기를 위한 데이터 포맷

앞 절에서 기술된 알고리즘들을 혼합 사용하는 자동 배선기를 프로그램화하기 위해 사용된 데이터 구조는 다음과 같다.

2.2.1 패드 구조 (Pad Structure)

패드는 PCB상의 부품에 대한 전기적 핀을 연결하는 요소로 각각의 부품은 다수의 패드를 보유할 수 있다. 따라서 패드정보 구조는 동적 메모리를 할당하여 (그림 11)과 같이 구성 하였다.

```
struct ROUTERPADS{
char value[8];
int pinid;
int lay;
float cx, cy;
float lx, ly, rx, ry;
char netname[8];
};
```

(그림 11) 패드 구조
(Fig. 11) Pad structure

여기서 value는 해당 핀이 위치하는 부품의 이름, netname 은 해당 핀에 연결된 네트의 이름, pinid는 해당 핀 번호, lay는 해당 핀의 층에 대한 정보를 설정한다. 만약 층의 정보가 "0"일 경

우는 through hole pad, "1"일 경우는 minimum layer SMD pad, "2"일 경우는 maximum layer SMD를 의미 한다. 그리고 cx, cy 는 해당 핀의 중심점 좌표, lx, ly, rx, ry는 해당 핀의 사각형 형태를 갖는 좌표 정보를 나타낸다.

2.2.2 부품 구조 (Part Structure)

PCB에 사용되는 파트 내부 구조는 (그림12)와 같은 정보로 구성되어 있다.

```
struct ROUTERPART{
char value[8];
int padcount;
int lay;
int draw;
float lx, ly,rx,ry;
ROUTERPAD *pad;
ROUTERPART *next;
};
```

(그림 12) 부품 구조
(Fig. 12) Part structure

여기서 value는 해당 부품 이름으로써 U1, U2, RES등과 같은 정보를 가진다. Padcount는 해당 부품이 보유하는 패드 의 수를 나타내고, lay는 해당 부품 층의 형태를 나타낸다. lay 정보에서 "0"은 through hole type을 의미하고, 1, 2 는 SMD type 을 나타낸다. Draw는 예약된 파라미터(reserved parameter)이고, lx, ly,rx,ry 는 해당 부품의 경계(boundary)영역이고, pad는 해당 부품의 패드 메모리 영역 이다. Next는 다

```
int PCBPARTLIST :: Add(char *s , char *n)
{
    PCBPART * temp;
    if(head==NULL) { // List에 파트가 존재하지않을 경우
head=new PCBPART;
cp=head;
cp->next=NULL;
}else{// List 에 파트가 1개 이상 존재할 경우
temp=new PCBPART;
temp->next = cp ->next;
cp->next=temp;
cp=temp;
}
    if(cp==NULL) return -1;
return LoadPackage( cp, s, n);
}
```

(그림 13) 부품삽입을 위한 알고리즘
(Fig. 13) Algorithms for the part insertion

음 부품의 메모리 포인터를 나타낸다. 부품 구조에 대한 메모리 상의 데이터 구조는 단일 링크드 리스트(linked list)로 구성되고 부품 삽입을 위한 알고리즘은 (그림 13)에 제시된 흐름에 의해 처리된다.

2.2.3 네트 구조(Net Structure)

PCB 배선기는 네트를 배선하는 주된 기능으로 전체 배선은 (그림 14)와 같은 네트 구조를 이용하여 처리하게 된다.

```

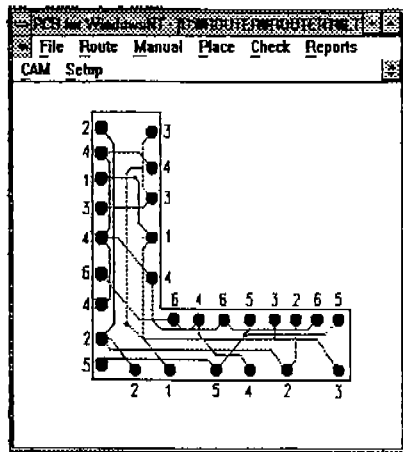
struct ROUTERNET{
char name[8];
char pt1[8];
char pt2[8];
float width;
int pin1;
int pin2;
float lx, ly, rx, ry;
int routed;
int selected;
ROUTERPATTERN *route;
ROUTERPATTERN *rcp;
ROUTERNET *next;
ROUTERPAD *p1;
ROUTERPAD *p2;
};
    
```

(그림 14) 라우팅을 위한 네트 구조
(Fig. 14) Net structure for the routing

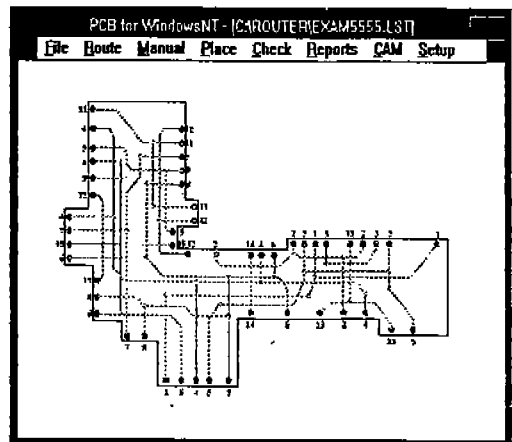
Name은 네트 명에 대한 정보로서 여러 네트중 동일 정보를 가지면 상호 연결된 네트로 간주한다. pt1, pt2는 해당 네트에 대한 연결된 패드가 존재하는 부품명이고, width는 해당 네트의 배선 두께, pin1, pin2는 해당 네트가 연결된 패드의 핀 번호, lx, ly, rx, ry는 네트의 시작과 끝점에 대한 정보이며, routed는 해당 네트의 배선 상태, selected는 순서화에 사용되는 해당 네트의 선택 파라미터 플래그에 대한 정보이다. 그리고 route는 배선된 패턴들이 존재하는 패턴 리스트의 헤더이며, rcp는 패턴들이 존재하는 패턴 리스트의 현재 정보, next는 리스트 구조의 다음 네트정보를 나타낸다. 마지막으로 p1, p2는 해당 네트가 연결된 패드의 포인터 값에 대한 정보를 제공한다. 이상과 같은 데이터 포맷들은 자동 배선 시스템 구축시 공용 데이터 구조로 활용할 수 있도록 정형화된 화일로 마련되어 진다.

3. 실험 및 고찰

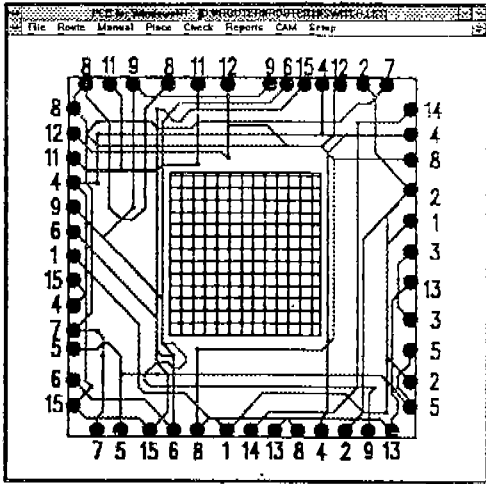
PCB 자동 배선 시스템은 IBM-PC 486DX2 66MHz 기종에서 MS-WindowNT 환경하의 Borland C++ 3.1Ver. 을 사용하여 개발 하였고 본시스템의 전체적인 구성도는 (그림 19)와 같다. 개발된 시스템 상에서 실행된 결과들을 <표 1>과 <표 2>에 나타내었고, 이러한 결과들은 수작업없이 자동 배선에 의해 얻어진 결과들이다. 본 논문에서는 Chen[9]이 제시한 (그림 15)와 같은 L-형 채널에 대한 수행 결과와 개발



(그림 15) L-형 채널 실행 결과
(Fig. 15) Executed result on L-shaped channel



(그림 16) 불규칙한 변을 갖는 L-형 채널
(Fig. 16) L-shaped channel with nonrectangular subchannels



(그림 17) 장애물이 있는 switch 박스
(Fig. 17) Switch box with a obstacle

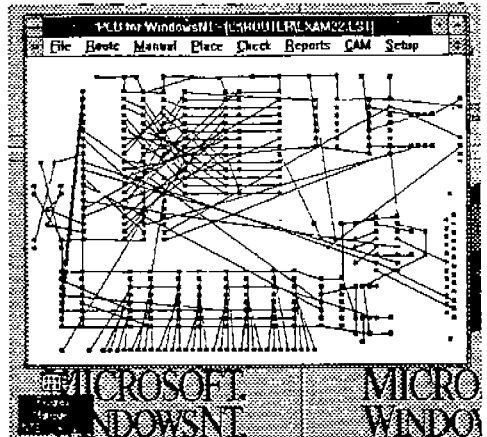
〈표 1〉 불규칙한 변을 갖는 L-형 채널 배선 결과
(Table 1) Routing results for L-shaped channel with nonrectangular subchannels

측정구분	측정형태	핀 수	Via 수	배선속도
	Chen의 L형	28	32	?
	Chen의 불규칙한 L형	50	67	?
	L형	28	5	82초
	불규칙한 L형	50	32	124초
	장애물이 있는 Switch Box	50	21	132초

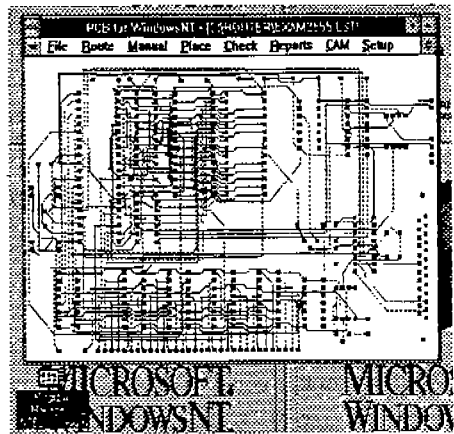
〈표 2〉 자동 배선 프로그램의 실행 결과
(Table 2) Executed results from automatic routing programs

측정형태	PCB종류	EX1	EX11	WPCB22	WPCB1	EXAM22
그리드 배선 영역	메모리	5.7Kbyte	8.2Kbyte	8.4Kbyte	8.5Kbyte	96.7Kbyte
Shapebased 메모리 배선 영역		28byte	40byte	36byte	96byte	232byte
배선 층		2	2	2	2	2
네트 수		14	20	18	48	116
핀 수		28	40	36	96	232
메모리 패턴 (%)		없음	없음	1%	78.1%	33.1%
shape-based 패턴 (%)		90%	73.6%	50%	15.6%	50.9%
개선된 그리드 배선 (%)		10%	26.4%	49%	6.3%	16.0%
총 계 (%)		100%	100%	100%	100%	100%
총 배선 시간		4초	15초	25초	41초	542초

된 혼합 자동 배선기로 수행한 결과를 비교 했을 때 비아 수가 32개에서 5개로 줄어들었고, 불규칙한 면을 갖는 채널인 (그림 16)을 수행했을 경우는 비아의 수가 67개에서 32개로 줄어들고 동시에 전체적인 배선 길이가 짧아짐을 알 수 있었다. 또한 장애물이 있는 50개의 핀수를 갖는 스위치 박스 (switch box)를 수행했을 경우 (그림 17) 과같이 효율적인 배선 결과를 얻을 수 있었다. 또한 혼합 배선기를 이용하여 〈표 2〉에서 제시된 5개의 불규칙한 부품 배치 PCB를 대상으로 자동 배선을 실행시켜 얻어진 특성을 비교 분



(a) 네트 연결도
(a) The result net connection



(b) 배선 결과
(b) The final executed the result

(그림 18) 네트 수 232개인 PCB의 라우팅 최종결과
(Fig. 18) Result of routing for the PCB with net NO. 232

석한 결과 보드상의 부품 배치에 따라 논-그리드와 그리드 방식간의 배선 비율이 다르겠지만 전반적으로 70% 정도는 논-그리드 방식을 적용한 메모리 패턴 배선기와 shape based 패턴 배선기로 수행되었고, 나머지 30% 정도는 개선된 그리드 알고리즘으로 수행하여 100%의 배선 결과를 얻을 수 있었고, 네트 수가 232개인 PCB의 배선 네트 연결 관계 및 배선실행 결과는 (그림 18)에 나타내었다.

4. 결 론

본 논문에서는 PCB 설계시 종전 방법들에서 제기되는 배선 영역의 메모리 문제로 인한 배선 속도 감소를 개선하기 위하여 그 문제점을 밝혀 개선을 행하였다. 배선시 문제는 보드의 배선영역을 그리드방식만을 적용할 경우 M*N 배열의 형태로 메모리 증가로 인해 탐색 시간이 증가하는 문제점을 갖고 있어 본 논문에서 제기된 알고리즘을 (그림 19)의 function 컬럼에 있는 자동 배선 순서대로 적용한 결과 경우 초기 배선시에 패드당 하나의 shape으로 간주하기 때문에 2byte의 메모리 영역을 소요하지 않으므로 배선 속도를 향상시킬수 있었다.

따라서 개발된 자동 배선 시스템도 그리드와 논-그리드 형식을 함께 사용하므로써 PCB 배선

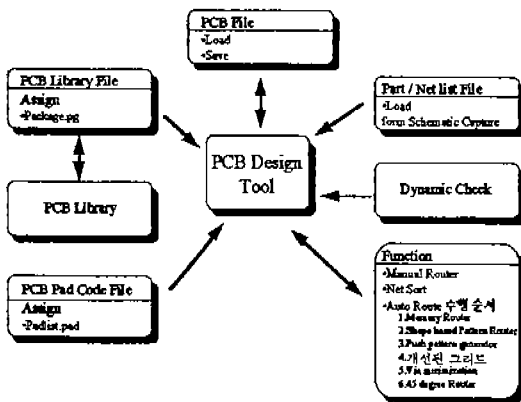
경로 탐색을 위한 배선 메모리 영역이 (표 2)에서 보는 바와 같이 감소됨을 알 수 있었고, 그에 따른 배선 속도와 설계 비용이 동시에 감소됨을 알 수 있었다.

그리고 긴배선의 경우 패턴생성을 하는데 30% 정도의 실패율을 보였으나, 이 경우에는 개선된 그리드 알고리즘을 적용함으로써 100% 배선 결과를 얻을 수 있었고, 이 결과를 다시 비아 최소화를 수행한 후, 45도 배선기를 실행하여 최종적인 배선 결과를 (그림 18)과 같이 얻을 수 있었다.

따라서 앞으로의 연구 과제는 shape 체크 속도를 빠르게 할 수 있는 알고리즘의 개발과 긴 배선의 경우에 대처할 수 있는 패턴 DB를 구축하여 super 자동 배선기 시스템을 구현하는데 있다.

참 고 문 헌

- [1] Hightower, David, "A Solution to line-Routing Problems on the Continuous Plane", Proceedings of the Design Automation Workshop, SHARE, ACM, and IEEE, 1969, pp. 1-24.
- [2] Hightower, David, "The interconnect Problem: A Tutorial", Proceedings of the Design Automation Workshop, IEEE, 1973, pp. 1-21.
- [3] F.Rubin, "The Lee path connection algorithm", IEEE Trans. Computers, vol. C-23, pp. 907-914, 1974.
- [4] M. F. Kelley and R. J. Smith, II, "An optimizing path printed circuit-board router", Proc. 10th Asilomar Conf. Circuits, Systems, and Computers, pp. 425-430, 1976.
- [5] M. T. Doreau and L. C. Abel, "A topologically based non-minimum distance routing algorithms", Proc. 15th Design Automation Conf., pp. 92-99, 1978.
- [6] Sokup, Jiri, Fast Maze Router, Proceed-



(그림 19) PCB 배선기의 전체 시스템 구성도
(Fig. 19) System configuration for PCB router

ings of the 15th. Design Automation Conference, IEEE, 1978, pp. 100-102

[7] Eustance, Robert Alan, " INTRA Region Routing ", The University of Central Florida, Ph. D. 1984.

[8] Watanabe, Hiroyuki, " IC Layout Generation and Compaction using Mathematical Optimization", The University of Rochester, Ph. D. 1984,

[9] Howard H. Chen, "Routing L-Shaped Channels in Nonslicing-Structure Placement,"in proc. 24th Design Automation Conf. pp. 152-158, 1987

[10] S. H. Gerez and O. E. Herrmann, "CRACKER : A General Area Router Based on Stepwise Reshaping ," in proc. Int. Conf. on CAD, pp.44-47, 1989.

[11] Arvind Srinivasan, Ernest S. Kuh," MOLE - A Sea-of-gates Detailed Router", Proceedings of the European Design Automation Conference, March 1990, pp. 446-450



최 영 규

1983년 청주대학교 전자공학과 졸업(학사)
 1986년 중앙대학교 전자공학과 졸업(석사)
 1995년 청주대학교 전자공학과 졸업(박사)
 1991년~92년 충주공업전문대 전자계산기과 전임강사

1993년~현재 충주산업대학교 컴퓨터공학과 전임강사
 관심분야 : CAD시스템 및 자동화



이 천 회

1971년 한양대학교 전자공학과 졸업(학사)
 1975년 성균관대학교 전자자료처리과 졸업(석사)
 1982년 한양대학교 전자공학과 졸업(석사)
 1987년 성균관대학교 전자공학과 졸업(박사)

1972년~72년 한국마벨
 1972년~77년 수송전기공업 고등학교 교사
 1977년~79년 동양공업 전문대학 전자과 전임강사
 1979년~현재 청주대학교 전자공학과 교수
 1983년~85년 미국 산호세캘리포니아 주립대학교 전자과 객원교수
 관심분야 : VLSI & CAD시스템 및 자동화 설계