

# 인쇄회로기판 자동배치/배선 시스템 개발에 관한 연구

김 현 기\* · 우 경 환\*\*

## 요 약

자동배치/배선 시스템의 배선영역 모델링 방법은 그리드와 논 그리드 방식을 사용하고 있다. 그리드 방식은 PCB상에 전기적, 물리적 요소들이 적다할지라도 보드와 그리드의 크기에 제약을 받기 때문에 메모리가 많아지게 되어 자동배치/배선 속도를 감소시키는 단점을 가지고 있다. 논 그리드 방식인 형상기반 방식은 영역처리 방식을 사용하므로 형상들을 메모리에 각각의 객체로서 존재시키며, 이들 객체는 고유한 데이터 크기를 갖기 때문에 메모리가 상당히 적게 소요된다. 그러므로 본 논문에서는 단일 원점에서 여러 목적지에 가장 빠르게 도달 할 수 있는 최단 경로 문제를 해결하는 경매 알고리즘을 적용하여 형상기반 방식에 의하여 메모리 낭비 없이 빠른 속도로 자동배치/배선할 수 있는 PCB 자동배치/배선 시스템을 개발하였다. 또한 본 시스템은 이와 같은 방식을 이용하여 PC에서 사용할 수 있도록 IBM Pentium 컴퓨터의 Windows 환경에서 Visual C++언어로 개발하였다.

## A Study on the Development Automatic Placement/Routing System in the PCB.

Hyun-Gi Kim<sup>†</sup> · Kyong-Hwan Woo<sup>\*\*</sup>

### ABSTRACT

The modeling methods of routing region used in the automatic placement/routing system are a grid and non-grid. Because the grid method is curbed by its size and a board if the electrical and physical elements on PCB are of small quantity, it has many memories. Therefore, it has demerit which decreases the speed of automatic placement/routing. The Shape-based type, non-grid method, makes the shapes exist as an individual element in a memory by using a region-processing method. Each individual element needs very small memory since it has its unique data size. Therefore, this paper aimed to develop the automatic placement/routing system which can automatically place and route the PCB without dissipation of memory at a high speed. To this aim, the auction algorithm method was applied which can make the memories be most rapidly reached from the original point to various destinations. Also, this system was developed by the Visual C++ in the Windows environment of IBM Pentium computer in order to use it in an individual PC system.

**키워드** : 형상기반방식(Shape-based Type), PCB, Non-grid 방식

### 1. 서 론

VLSI 기술의 급속한 발전으로 집적회로 설계 기술은 더욱더 복잡해지고 빠르게 되었고, 소형화, 다기능화가 이루어짐에 따라 PCB 설계의 복잡도가 증가하므로 설계 시간의 단축이 상당한 요인으로 등장하게 되었다.

PCB 설계 과정의 가장 중요한 자동배치/배선 알고리즘은 원시적인 방법에서부터 최신 CAD(Computer Aided Design) 설계 기술을 활용하는 방법에 이르기까지 고속화되어 가는 추세이므로 현존하는 대부분의 배선 시스템은 한 가지 방법에 대한 결점을 보완하기 위해 혼합-교환 그래프 알고리즘

[1], Building Block 알고리즘[2], 유전자 알고리즘[3], 경매 알고리즘[4] 등과 같이 몇 가지 알고리즘에 의해 배선 시스템은 구성된다.

PCB 설계에 있어서 가장 핵심적인 필요조건은, 부품 배치 후 신호의 상호 전달 매체로서 부품 핀간의 배선규칙인 PPD(Pad Pad Distance)와 PTD(Pad paTtern Distance), TTD(paTtern paTtern Distance)의 3가지 기본적인 파라미터 값에 의해 보드에 제시되는 네트들을 전기적 오류 없이 물리적으로 각 요소들 사이에 최소 인접거리를 유지하도록 동박으로 연결시키는 자동 배선으로 PCB 배선의 문제는 크게 3가지로 볼 수 있다. 첫째, 배선영역(Routing Region)은 패턴이 통과할 수 있는 영역으로 설계규칙과 장애물들에 의해 이미 준비되어있는 영역을 사용하게 된다. 이것은 매트릭스 형식의 메모리 맵 방식을 이용

<sup>†</sup> 정 회 원 : 극동정보대학 정보통신과 교수

<sup>\*\*</sup> 정 회 원 : 우송공업대학 의료복지과 교수

논문접수 : 2004년 4월 11일, 심사완료 : 2004년 7월 29일

함으로서 어떤 특정한 수치로 나타낼 수 있다. 둘째, 금지영역(Forbidden Region)으로 배선시 장애물을 나타내는 것으로 패턴의 진행을 막는 요인이 된다. 금지영역의 증가는 배선영역이 그만큼 감소되므로 금지영역의 최적화는 배선효율을 높일 수 있다. 셋째, 배선요구(Routing Request)는 PCB 상의 배선을 수행하게 될 네트성분으로 이 네트들은 특정한 방식으로 순서화(Ordering)하여 배선을 수행한다.

배선영역은 실제로 볼 수 있는 가시적인 파라미터로 가시적인 객체(object)들을 프로그램화하여 얻을 수 있는 정보형태들로 프로그램 되어있는 데이터로부터 즉시 획득할 수 있어야 한다. 주어진 영역을 프로그램화한다는 것은 이 영역을 모델링 하는 것으로 주어진 영역을 어떠한 형태로 분석해서 이를 데이터화하여 필요한 정보를 도출해내는 것이다.

배선영역을 모델링 하는 방식으로는 그리드(grid) 방식과 형상기반 방식을 사용하고 있다. 그리드 방식은 특정 영역에 대한 2차원적인 메모리 맵을 할당하여 배선에 필요한 경로를 탐색하는 방식으로 Maze 배선기와 선분의 전파를 이용한 선분 탐색 방식, 하이타워 배선기(Hightower router)[5] 등으로 구분할 수 있다. 이와 같은 배선영역 모델링 방법은 배선영역을 바둑판 모양으로 m×n 배열로 분리하여 배열 번수에 사각형 영역의 정보를 저장하기 때문에 m과 n을 크게 할수록 더욱 정밀한 정보를 저장할 수 있다. 그러나 소모 메모리의 증가와 경로를 탐색할 때 필요한 메모리 참조에 대한 많은 제약과 시간이 소요되므로 본 논문에서는 형상기반 방식을 적용하여 빠른 시간 내에 자동배치/배선이 수행될 수 있도록 하였다.

형상기반 배선 방식은 위에서 설명한 그리드 배선 방식의 여러 가지 문제점을 해결하기 위해 최근에 많이 사용되는 방식이다. 이 배선 방식은 배선시 그리드를 발생시키지 않기 때문에 단위 셀을 생성시키지 않는다. 이와 같은 이유로 단위셀에 의한 영역 처리를 수행하지 않고 object based의 영역처리 방식을 적용한 패턴 배선기를 적용하므로 그리드 방식과 같은 메모리 증가 문제를 해결할 수 있다. 따라서 본 논문에서는 위와 같은 설계규칙을 근간으로 하여 단일 원점에서 여러 목적지에 도달할 때 최단경로를 산출하는 경매 알고리즘을 네트들의 배선요구를 받아들여 자동으로 그 경로를 찾아 자동 배선하는 형상기반 방식에 적용하여 PCB 자동 배선 시스템을 개발하였다.

**2. 경매 알고리즘**

경매 알고리즘에서 각 개체를 할당하기 위해서는 반복적으로 실행하여야 하며, 각 반복 시 상보성(CS)조건을 충족

하여야 하고, 개체 *i*에 대한 대상 *j*의 가치는  $a_{ij} - b_j$ 이며, 상보성 조건은 다음 식 (1)과 같다.

$$a_{ij} - b_j = \max_{j \in A(i)} \{a_{ij} - b_j\} \tag{1}$$

만일 식 (1)에서 모든 개체가 할당된다면, 알고리즘은 종료하고, 그렇지 않으면 할당되지 않은 개체 *i*의 비어있지 않은 부분집합 *i*가 선정되고, 경매 알고리즘의 전형적인 반복이 실행된다.

경매 알고리즘은 의사 다항식과 같은 복잡성을 가지며, 한 개의 원점과 여러 개 목적지를 가진 문제에 대해 다른 알고리즘보다 우수한 수행 결과를 나타내었다. 경매 알고리즘의 의사 다항식 항목에 확장 단계와 축소 단계는 Pallottino와 Scutella[6]가 정의하였다.

경매 알고리즘의 수행 시간은  $O(m^2)$ 이며, 여기서 *m*은 원호의 수이다. 감소되지 않는 길이 순서로 각 노드의 나가는 원호에서 배열 개념을 이용하면 수행 시간은  $O(mn)$ 만큼 더 감소하게 되며, 여기서 *n*은 노드의 수이다.

경매 알고리즘은 단일 원점/모든 목적지의 최단경로 문제에 대해 우수한 수행 결과를 가지고 있기 때문에, 최단 경로 문제에 경매 알고리즘을 이용하는 필요성은 PCB 회로 설계시 실행 가능한 범위 내에서 최단 경로를 계산하므로 전체 선 경로 최소화에 의한 시스템 성능 향상과 비용을 최소화할 수 있는 장점이 있다.

**2-1. 경매 알고리즘의 전형적 반복 / 최단 경로 알고리즘**

최단 경로에 대한 경매 알고리즘은 항상 단순한 경로  $P = (1, i_1, i_2, \dots, i_k)$ 를 유지한다. 만일  $i_{k+1}$ 은 경로  $P = (1, i_1, i_2, \dots, i_k)$ 에 속하지 않는 노드이고,  $(i_k, i_{k+1})$ 은 원호이다.  $i_{k+1}$ 에 의하여 *P*의 확장은 경로  $(1, i_1, i_2, \dots, i_k, i_{k+1})$ 에 의한 *P*를 대체하는 것을 의미하며,  $i_{k+1}$ 에 의한 *P*의 확장이라고 부른다. 만일 *P*가 원점 노드만을 구성하지 않는다면, *P*를 축소하는 것은 경로  $(1, i_1, i_2, \dots, i_{k-1})$ 에 의하여 *P*를 대체하는 것을 의미한다.

알고리즘은 *P*와 함께 아래와 같은 성질을 충족시키는 가격 벡터 *p*를 포함한다.

$$p_i \leq a_{ij} + b_j, \quad \forall (i, j) \in A \tag{2a}$$

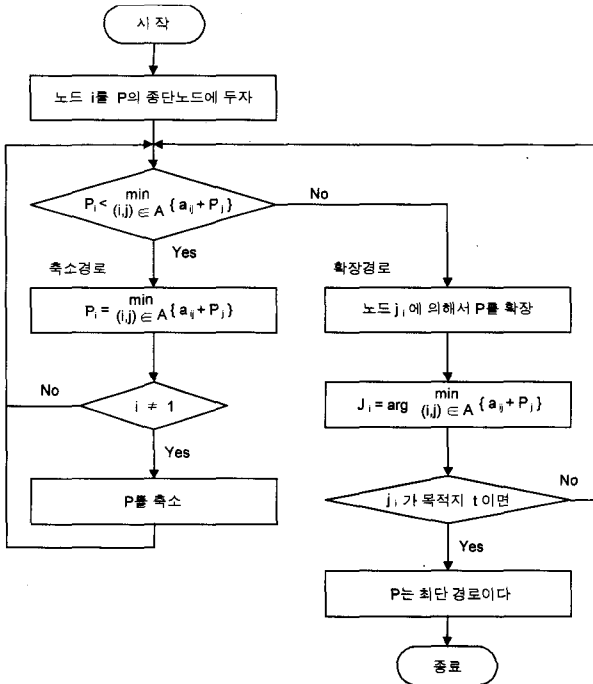
$$p_i = a_{ij} + b_j, \quad P \text{의 연속하는 노드 } i \text{와 } j \text{는 모든 쌍이다} \tag{2b}$$

이 조건은 최단 경로 문제의 공식화에 의한 상보성 조건은 물론이고, 등가 할당 문제에 대한 상보성 조건과 관계가 있을 수 있다[7].

만약 쌍  $(P, p)$ 이 상보성 조건을 충족한다면,  $p_1 - b_1$ 가 최

단 경로 거리인 반면 노드 1과 노드  $i \in P$  사이의  $P$ 의 일부는 1에서  $i$ 까지 최단 경로라고 볼 수 있다.

$i$ 를  $P$ 의 종단 노드에 둘 때 경매 알고리즘은 (그림 1)과 같다.



(그림 1) 경매 알고리즘의 전형적 반복

2.2 개선된 최단경로 경매 알고리즘

경매 알고리즘에서  $n$ 개체와  $n$ 대상의 결합 가능성을 고려해볼 때 대상  $j$ 가 가격  $b_j$ 를 취하고 대상을 받는 개체는  $b_i$ 를 지불한다고 가정할 때 개체  $i$ 에 대한 대상  $j$ 의 가치는  $a_{ij} - b_i$ 이며, 각 개체는 논리적 최대 값으로 대상  $j_i$ 에 다음 식과 같이 할당되어진다.

$$a_{ij} - b_i = \max_{j \in A(i)} \{a_{ij} - b_j\} \quad (3)$$

개선된 최단경로 경매 알고리즘에서의 할당 문제는 각 노드가  $i \neq t$ 일 때 개체노드  $i'$ 를 삽입하고  $i \neq t$ 와  $j \neq 1$ 를 가지는 최단경로 문제의 모든 원호  $(i, j)$ 에 대하여 원호  $(i', j)$ 를 삽입한다.

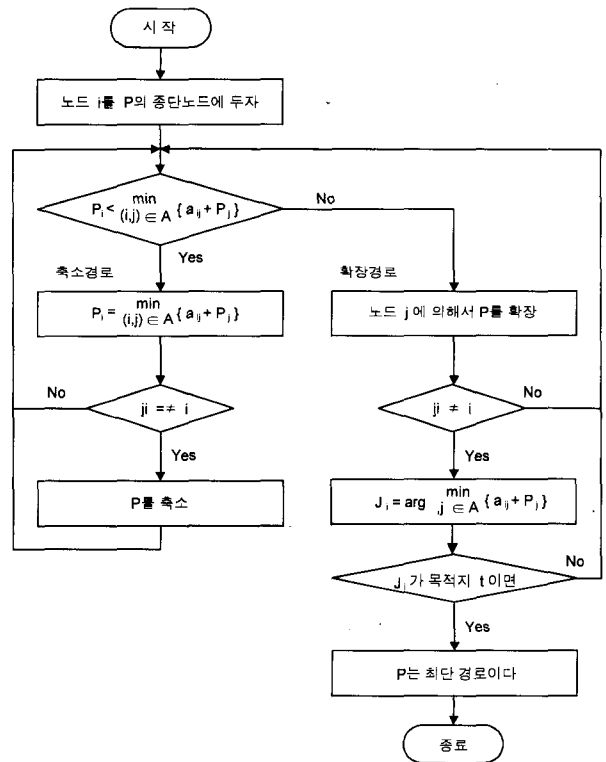
최단경로 문제에서 알고리즘은 항상 단순경로  $p = (1, i_1, i_2, \dots, i_k)$ 와  $p_1 = \infty$ 를 가지고 있는 가격벡터  $p$ 를 유지하면서 할당된다. 그러나 쌍  $(P, p)$ 는 상보성 조건을 만족시키지 않으나 다음 조건을 만족시키면서 최단경로를 산출한다.

$$\pi_i = a_{ij} + b_j, \quad \forall (i, j) \in p \quad (4)$$

$$\pi_i = b_i, \quad \forall i \in p \quad (5)$$

$$\therefore \pi_i = \min \left\{ b_i, \min_{(j|(i,j) \in A} (a_{ij} + b_j) \right\} \quad (6)$$

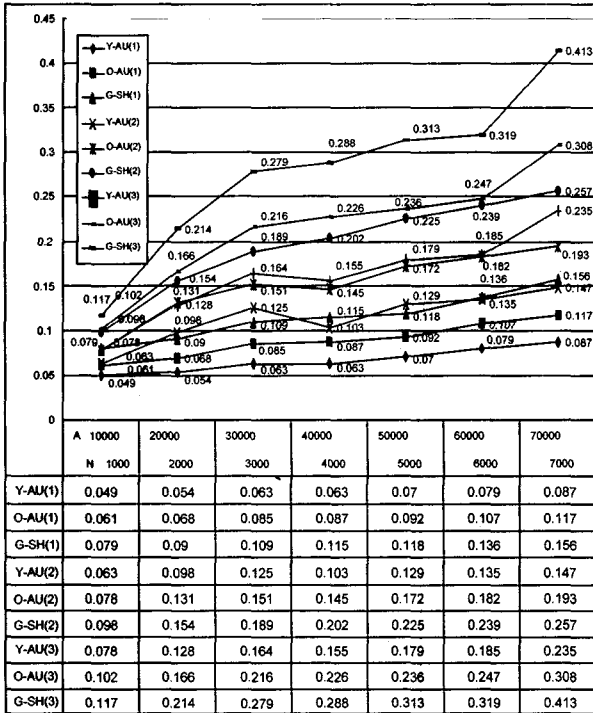
여기에서  $\pi_i$ 는 개체  $j$ 의 이득변수이고,  $a_{ij}$ 는 대상  $j$ 와 개체  $i$ 를 결합하기 위한 이득이고,  $b_i$ 는 개체  $i$ 의 가격,  $b_j$ 는 대상  $j$ 의 가격을 말한다. 위와 같은 조건을 만족하는 개선된 최단경로 경매 알고리즘에서 최단경로를 산출하기 위한 반복과정은 축소경로에서  $i \neq 1$ 일 때  $p$ 를 축소하고 그렇지 않으면 반복과정으로 진행하였으나, 개선된 알고리즘에서는  $j_i = i$ 일 때  $p$ 를 축소하고 그렇지 않으면 반복과정을 수행하였다. 확장경로에서는 추가적인 조건인  $j_i \neq i$ 를 추가하여  $j_i \neq i$ 일 때 다음과정인  $j_i = \arg \min_{j \in A(i)} \{a_{ij} + b_j\}$ 를 수행하게 하여 최단경로를 산출하였으며, 개선된 알고리즘은 (그림 2)와 같다.



(그림 2) 개선된 최단경로 경매 알고리즘

알고리즘의 수행방법은 단일원점과 다중 목적지의 선택된 세트에 대한 문제점을 해결하고 벤치마크 테스트 기법을 적용하여 노드의 수  $N$ 과 호의 수  $A$ 를 변화하면서 수행하였다. 실험에서는 노드의 수를 1,000에서 시작하여 1,000씩 증가시키고 7,000개로 제한하였으며, 호의 수는 동일한 노드의 수에 10배씩 증가시켜 사용하였고 목적지는 1개, 2개, 3개로 제한하여 실험하였다.

실험에 사용한 알고리즘은 개선된 최단경로 경매알고리즘 (Y-AU)과 기존 경매 알고리즘(O-AU), 일반적인 최단경로 알고리즘인 SHEAP (G-SH)알고리즘을 비교 분석하였으며, 최단경로 알고리즘의 수행결과는 (그림 3)에 나타내었다.



(그림 3) 1개, 2개, 3개 목적지를 가진 경우의 수행결과

실험결과 목적지가 1개일 경우에는 Y-AU는 O-AU보다 32.7%의 성능향상과 O-AU는 G-SH보다 40.0%의 성능향상을 보였다. 목적지가 2개일 경우에는 Y-AU는 O-AU보다 31.5%의 성능향상과 O-AU는 G-SH보다 39.0%의 성능향상을 보였으며, 목적지가 3개일 경우에는 Y-AU는 O-AU보다 33.5%의 성능향상과 O-AU는 G-SH보다 39.4%의 성능향상을 보였다. 또한 노드수가 4,000개일 경우에 두드러지게 다른 경우보다 성능이 좋은 것으로 나타났다.

### 3. 자동 배치/배선 알고리즘 및 데이터 포맷의 고찰

기본적인 설계 규칙을 근간으로 PCB 자동 배치/배선 설계 시스템을 구축하기 위해 단일 원점에서 다중 목적지의 최단경로를 산출하는 경매 알고리즘 및 형상기반 라인 탐색 배선 알고리즘과 형상기반 재배선 기법을 적용하여 자동 배치/배선 시스템을 개발하였다.

#### 3.1 형상기반 배선 알고리즘

PCB 배선 방법은 그리드 방식과 논-그리드 방식인 형상기반 배선 방식이 있다. Lee 알고리즘 또는 Maze 알고리즘

을 적용하여 배선하는 그리드 방식은 다음과 같은 문제점이 있으므로 본 논문에서는 형상기반 배선 방식을 적용하였다.

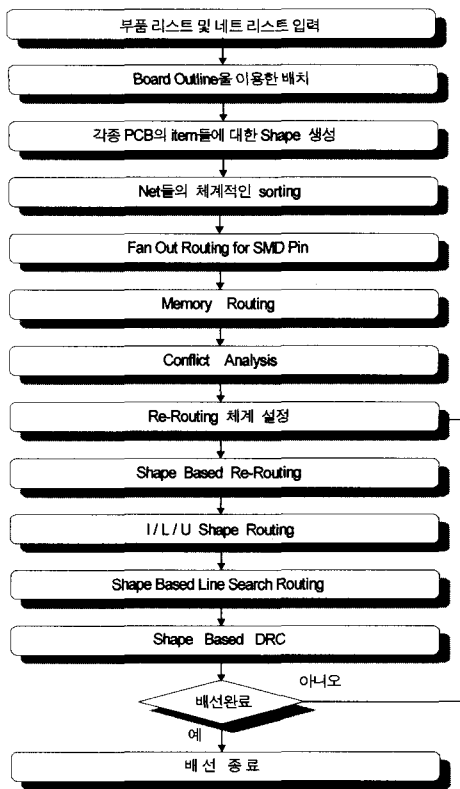
그리드 방식은 특정한 객체(PAD, Copper, Via등)들이 2차원적인 평면 또는 레이어가 다른 3차원적인 공간에 놓이게 되는데 이를 그리드라는 평면 위에 올려놓아 거리를 계산하는 방식으로 그리드에 대한 2차원적인 메모리를 할당하여 주어야하므로 많은 메모리가 요구되어지고, 사용하지 않는 메모리도 많이 발생된다. 이러한 그리드 매핑방식의 배선 알고리즘에서 메모리 극복에 이어 해당 객체간의 정확한 거리 계산에 대한 오차의 범위를 줄이는 문제가 제기된다. 즉, 입자화 되어있는 그리드안에 객체를 올려놓다 보면 해당 객체의 실제 크기보다 그리드의 셀 단위의 계산이 이루어지기 때문에 거리 계산의 오차가 발생할 수 있다. 그러므로 배선에 필요한 영역을 낭비하는 결과가 발생된다.

그리드 방식의 계산 오차 및 메모리 낭비를 줄이기 위해 형상기반 방식은 어떠한 객체를 그 형태 자체로 인식하여 계산하는 방식으로 PCB에서는 다음과 같은 shape들이 놓이게 된다. ① Round : 부품의 PAD와 Via의 모양을 정의한다. ② Rectangle : 부품의 사각 PAD 모양을 정의한다. ③ Line : 배선 하나의 세그먼트를 정의한다. ④ Polyline : 배선이 연속된 세그먼트 및 동판과 같은 영역을 지정한다. ⑤ Board Outline : 기본적인 라인 형태로 정의한다. 이러한 객체들은 각각의 위치정보, 두께 및 전기적인 특성 정보, 레이어 정보를 포함한다. 또한 이러한 shape들은 메모리에 각각의 객체로서 존재하며 이들 객체는 고유의 데이터 크기를 갖는다. 즉, 라인이 길다고 더 많은 메모리를 필요로 하지 않고 라인이 작다고 더 작은 메모리를 요구하지 않는다. 형상기반 방식에 의한 PCB 자동 배선 과정은 (그림 4)와 같다.

형상기반 자동 배선과정은 여러 단계를 거쳐서 배선이 완료되는데 본 논문에서 적용한 각 과정에서의 실행 방법은 다음과 같다.

- ① 부품 및 배선(netlist)의 입력 : 이 부분에서는 PCB에 적용할 부품들과 배선의 요구 사항인 네트들을 입력하는 단계이다. 부품은 해당 부품의 실크 부분과 실제 배선이 연결되고 부품이 실장될 Land 부분으로 나뉘어지며 Land는 Pad 또는 pin이라는 형태로 불리어진다.
- ② 각종 item들의 shape 생성 : 형상기반의 알고리즘을 적용할 수 있도록 각 PCB요소에 대하여 shape(Line, Circle, Arc, Polygon 등)들의 데이터베이스를 작성하여 추출 생성 해 주며 또한 보다 빠른 검색을 위하여 sorting을 한다. Sorting은 일반적으로 2차원 좌표에 대한 sorting을 수행하게 되는데 사용한 방법은 quick sorting기법을 적용하였다.

③ 네트들의 체계적인 sorting : 네트들은 거미줄과 같은 매우 복잡한 연결성을 보여 준다. 네트들을 배선하는데 있어서는 배선 순서, 즉 배선에 투입되는 순서를 적당하게 바꿈으로서 배선율을 높일 수 있다. 네트들을 위에서 아래 순서로 또는 좌에서 우의 순서로 sorting하거나, 네트의 길이가 짧은 쪽에서 긴 쪽으로 또는 그 반대 방향으로 sorting을 한다. 이중 가장 효율적인 방법은 짧은 쪽에서 긴 쪽으로 sorting하는 방법이다.



(그림 4) 자동 배선 과정

- ④ Fan Out Routing : SMD 부품들은 실제로 그 피치가 매우 작다. 작은 피치 안에서 via같은 것을 뚫을 수 없다. 이러한 이유로 배선의 시작 단계에서 fan out을 수행하면, SMD부품은 더 이상 SMD가 아니고 PTH pin이 된다. 즉 배선 알고리즘에 가장 제약을 주는 레이어 제한에 대하여 자유로울 수 있다.
- ⑤ Memory Routing : 메모리와 같은 소자는 그 특성상 matrix 구조를 갖고 또한 Data, Address Bus와 같은 병렬의 네트들이 많이 존재한다. 이러한 네트들을 via 없이 횡 또는 종으로 빠르게 배선을 수행한다.
- ⑥ Shape Based Line Search : autorouting의 주요 부분으로 Line Search를 이용한 배선을 수행한다.

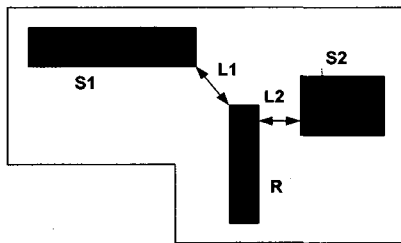
- ⑦ Shape Based DRC : routing한 네트들에 대하여 Design Rule Check을 수행하여 배선의 conflict를 찾아 data base 화하는 부분이며, 데이터베이스화 한 conflict 들은 다음 부분에서 해석되어 re-routing의 정보로 사용된다.
- ⑧ Rerouting 체계 설정 : 어떠한 배선이 배선을 수행하지 못하는 경우 가장 큰 요인은 해당 목적지까지 도달할 길 즉, 공간이 없다는 것이다. 이 경우 공간을 마련하여 특정한 배선을 조금 밀고 또는 특정 배선을 풀어 영역을 만든 후, 배선되지 않은 배선을 수행한다. 만일 배선이 수행되면 앞에서 푼 배선을 다시 배선하면 처음과 다른 배선 경로가 만들어지며 배선이 수행될 것이다.
- ⑨ I, L, U Shape Routing : Shape routing에서 패턴생성은 패턴 배선기에 사용되는 패턴들을 생성해주는 기능을 가진다. 이것은 단순한 패턴으로부터 복잡한 패턴으로 순차적으로 생성해 주며, Shape의 기본 형태는 네트들을 lookup 하여 I Type, L type, U type[8]을 이용하여 배선을 수행한다. I형 패턴은 수직 또는 수평 배선 네트에 사용되는 패턴을 만들기 위해 이용되는 형태로서 핀 간 장애물이 없을 경우 배선이 가능하나, 핀 사이에 장애물이 있을 경우에는 메모리 패턴으로 처리 되도록 구현한다.

L형 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. U형 패턴은 수직, 수평 또는 사선형태의 네트를 연결하는데 사용한다. 이것은 3개의 패턴을 이용하여 패턴을 생성시키며, 중간 패턴이 수직 또는 수평으로 평행 이동하면서 패턴을 생성한다. 패턴 생성시 경로 검색은 생성된 패턴들을 기존의 패드와 패턴들간의 배선 위치 검색을 하여 생성된 패턴에 에러가 발생했는지를 검사하는 과정이다. 이것은 크게 패드와 패턴간의 위치 검색과 패턴과 패턴간의 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 이미 배치된 패드와 비아 위치 검색을 수행하여 에러가 발생하면 해당 패드의 포인터를 반환하여 다시 위치 검색을 수행한다. 또한 패턴과 패턴간의 위치검색 함수 ROUTER PATTERN ROUTER TYPE :: DistCalPattern(float x1, float y1, float x2, float y2, int layer, ROUTER NET n)는 주어진 영역(x1, y1, x2, y2)와 이미 생성된 패턴과의 배선 위치 검색을 수행하는 함수로 배선 층과 n이라는 부가적인 인수를 가지고 패턴간의

위치 검색을 수행한다.

Shape based는 어떤 모델링 객체들을 shape라는 구조로 처리하기 위해 배선에 필요한 shape를 장애물 shape와 routing shape 2가지로 크게 나누어 실행하며, (그림 5)와 같이 S1과 S2의 장애물 shape와 R의 routing shape로 나타낼 수 있다.

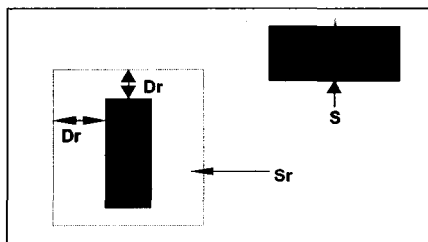
L1이 S1과 R간의 최단거리, L2는 S2와 R간의 최단거리라 할 경우 R shape가 배선에 수용될 수 있는지는 L1과 L2의 거리가 배선 규칙에 맞는지를 검사하면 된다. 두 shape간의 배선 규칙을 Dr이라 할 때 배선을 수용하기 위해서는 다음 조건을 만족해야 한다.



(그림 5) 형상기반 모델링 방법

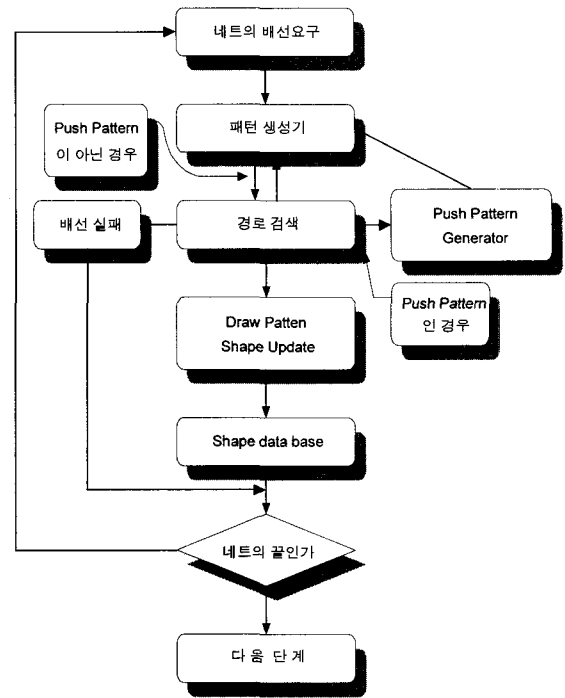
$$L1 \geq Dr \text{ AND } L2 \geq Dr$$

사각형 shape에 대한 효율적인 배선 규칙은 두 개의 shape가 배선 규칙에 위반되는지를 검사하기 위해 두 사각형의 최단거리를 구하지 않고 (그림 6)에서와 같이 R영역을 Dr만큼 확장하여 Sr 영역을 생성함으로써 이 영역이 S영역과 overlapping 되지 않으면 S와 R 영역간의 최단거리가 Dr보다 큰 값을 갖는다. 따라서 형상기반의 경로 탐색은 형상기반 영역처리 방식을 이용하여 이루어지게 된다.



(그림 6) 형상기반 배선규칙 검사 방법

패턴 생성은 가장 기본적인 패턴인 I, L, U 패턴 이외에 이들을 조합하여 I+L+L+I+I, I+L+L+U+L+I 등 극히 제한적인 조합 및 높은 배선율을 얻을 수 있기에 배선의 앞부분에서 빠르게 배선을 수행하는 기능으로 사용하기 위해 응용 shape들을 생성하도록 구현하였다. 이와 같은 형상기반 배선기의 전체적인 처리과정은 (그림 7)과 같다.



(그림 7) 형상기반 패턴 라우터 흐름도

### 3.2 Via의 최소화(Via Minimization)

비아 최소화는 비아 수를 최적화하는 기능으로 단층 재-배선(Re-routing)과 최적화 기법이 이용된다. 단층 재-배선 기법은 단층 배선에서 비아가 필요 없기 때문에 2층으로 배선이 끝난 PCB를 단층으로 재 배선하여 비아 최소화를 구하는 방법이다. 즉, 현재 배선된 경로에 비아가 존재할 경우 비아를 사용하지 않는 다른 경로를 찾아 배선을 수행한다. 이 기법은 비아 형태를 보고 최적화를 수행하는 기법으로 패턴에 형성된 비아는 다음과 같은 2가지 형태로 구현된다. 첫째, 패턴 당 하나의 비아가 존재하는 경우로 패턴 세그먼트 전후의 층이 서로 다른 경우로 하나의 비아가 발생하는 1 패턴 1 비아 형태이다. 둘째, 2개의 비아가 존재하는 경우로 패턴 세그먼트 전후의 층이 다른 경우로 2개의 비아가 발생하는 1 패턴 2 비아 형태이다. 1패턴 2 비아는 비아 우선 처리 방법에 있어서 1 패턴 1 비아 보다 우선 순위가 높다. 즉, 1 패턴 1 비아는 패턴을 다른 층으로 변경할 경우 다른 하나의 비아가 발생하는 경우도 존재한다. 그러나 1 패턴 2 비아는 패턴 세그먼트 전후의 층에 연결된 패턴과 같은 층으로 변경할 경우 2개의 비아가 줄어들게 됨으로 1 패턴 2 비아를 우선 검색한 후 1 패턴 1 비아를 검색하므로 비아 최적화를 수행할 수 있다.

### 3.3 형상기반 배치 구현

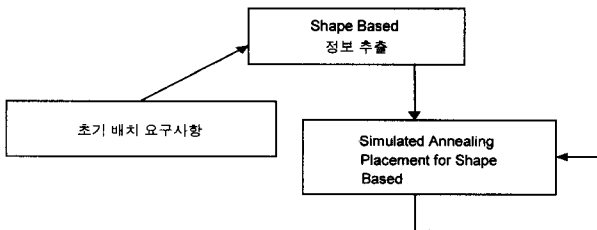
개념적으로 배치란 서로 다른 부품들을 특정한 영역에 올려놓은 다음 배선을 수행하도록 하는 과정을 말한다. 실제로

PCB는 조립이라는 공정의 라인과 연계가 되어 있어 배치 및 배선은 중요한 문제가 된다. 배치는 실제로 공정의 요구 즉, 부품의 공정상의 난이도 및 제품의 외형과 연계되어 있기 때문에 자동 배치라는 개념은 무의미하며 수동 배치와 자동 배치를 혼합하여 수행한다.

형상기반 배치란 거의 모든 부품은 그 형태가 정형화되어 있으나 실제로는 각각 다른 모습을 갖고 있다. 즉 배치에 사용되는 방법은 부품의 모양과는 상관없이 직/정 사각형의 모양이라 생각하여 배치를 수행한다. 이러한 이형의 부품을 배치하기 위해서는 부품들을 위치가 낮은 에너지 쪽으로 이동시키는 과정을 반복하여 부품들이 판에 효율적으로 배치시키는 simulated annealing 기법과 각 부품들의 위치 계산은 형상기반 기법을 적용하여 배치과정을 수행한다. (그림 8)은 이와 같은 기법을 적용한 형상기반 배치과정을 보여준다.

3.4 형상기반 배선기 데이터 포맷

앞 절에서 기술한 알고리즘들을 혼합하여 구현한 자동 배선 시스템을 프로그래밍하는데 사용한 데이터 구조는 다음과 같다.



(그림 8) 형상기반 배치과정

3.4.1 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 9)와 같이 구성하였다.

```

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

(그림 9) 패드 구조

여기에서 value는 해당 핀이 위치하는 부품의 이름, net-name은 해당 핀에 연결된 네트 이름, pinid는 해당 핀 번호, lay는 해당 핀의 층에 대한 정보를 나타낸다. 만약 층의 정

보가 "0"일 때는 through hole pad, "1"일 때는 minimum layer SMD pad, "2"일 때는 maximum layer SMD pad를 나타낸다. cx, cy는 해당 핀의 중심점 좌표이고, lx, ly, rx, ry는 해당 핀의 사각형 형태를 갖는 좌표 정보를 나타낸다.

3.4.2 부품 구조(Part Structure)

PCB에 사용되는 부품 내부 구조는 (그림 10)과 같이 구성하였다.

여기서 value는 해당 부품 이름으로 U1, U2, RES등과 같은 정조를 갖고 있다. padcount는 해당 부품이 보유하는 패드의 수를 나타내고, lay는 해당 부품 층의 형태를 나타내며, lay 정보가 "0"이면 through hole type을 나타내고, 1과 2는 SMD type을 나타낸다. draw는 예약된 파라미터이고, lx, ly, rx, ry는 해당 부품의 경계 영역이고, pad는 해당 부품의 패드 메모리 영역이다. next는 다음 부품의 메모리 포인터를 나타낸다. 부품 구조에 대한 메모리상의 데이터 구조는 단일 linked list로 구성되었으며, 부품 삽입을 위한 알고리즘은 (그림 11)에 나타내었다.

```

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

(그림 10) 부품 구조

```

int PCBPARTLIST :: Add(char *s, char *n)
{
    PCBPART *temp;
    if(head == NUL) {
        head = new PCBPART;
        cp = head;
        cp -> next = NULL;
    }
    else {
        temp = new PCBPART;
        temp -> next = cp -> next;
        cp -> next = temp;
        cp = temp;
    }
    if(cp == NULL) return -1;
    return LoadPackage(cp, s, n);
}
    
```

(그림 11) 부품 삽입 알고리즘

3.4.3 네트 구조(Net Structure)

PCB 자동 배선기는 네트를 배선하는 기능이 주된 기능으로 전체 배선은 (그림 12)와 같은 네트 구조를 이용하여 처리하게 구현하였다.

여기에서 name은 네트 이름에 대한 정보로서 여러 개의 네트 중에서 똑같은 정보를 가지면 상호 연결된 네트로 간주한다. pt 1, pt 2는 해당 네트에 대한 연결된 패드가 존재하

는 부품 이름이고, width는 해당 네트의 배선 두께를 나타내고, pin 1, pin 2는 해당 네트가 연결된 패드의 핀 번호이다. lx, ly, rx, ry는 네트의 시작과 끝점에 대한 정보이며, routed는 해당 네트의 배선 상태를 나타내며, selected는 순서화에 사용되는 해당 네트의 선택 파라미터 플래그에 대한 정보이다. route는 배선된 패턴들이 존재하는 패턴 리스트의 헤더이고, rcp는 패턴들이 존재하는 패턴 리스트의 현재 정보를 나타내고, next는 리스트 구조의 다음 네트 정보를 나타낸다. p1, p2는 해당 네트가 연결된 패드의 포인터 값에 대한 정보를 제공한다. 이와 같은 데이터 포맷들은 자동 배치/배선 시스템 구축시 공용데이터 구조를 활용할 수 있도록 정형화된 파일로 구현하였다.

```

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

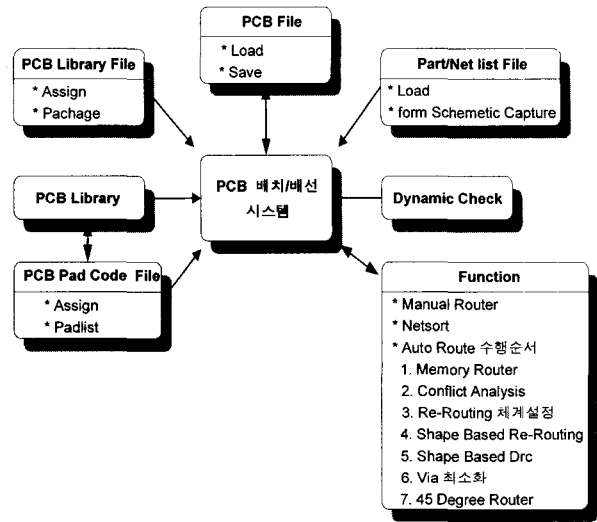
(그림 12) 배선을 위한 네트 구조

4. 실험 및 고찰

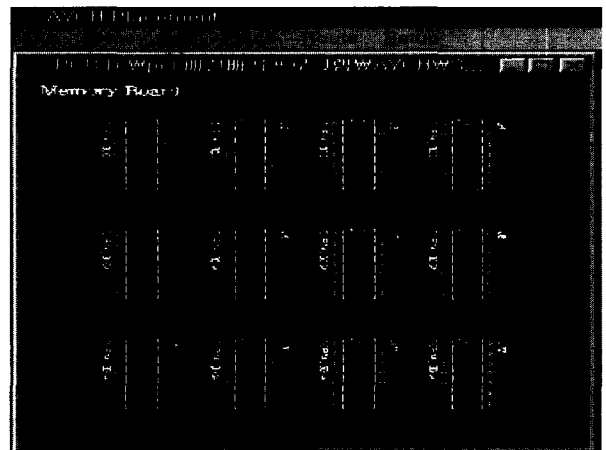
본 논문에서 구현한 PCB 자동 배치/배선 시스템은 IBM-PC pentium 컴퓨터에서 Windows 환경하의 Visual C++언어를 이용하여 개발하였으며, 본 시스템의 Placement Tool과 Autorouter Tool의 상호 동작 구조는 (그림 13)과 같으며, 배치 시스템은 수동과 자동 배치를 혼용하여 수행하도록 구현하였고, 배선 시스템은 배치가 완료된 다음 자동으로 배선하도록 구현하였다. 전체적인 시스템 구성도는 (그림 13)과 같다.

개발된 시스템 상에서 (그림 14)와 (그림 15)에서 보여준 메모리 보드 배치 배선의 실행된 결과를 보여준다.

본 시스템은 네트수가 200개인 PCB 배선 네트 연결 관계 및 배선 실행 결과는 (그림 16)에 나타내었으며, 형상기반 배선 방법으로도만 100%의 배선율은 <표 1>의 비교 분석을 통해 알 수 있다. <표 1>에서 보는 바와 같이 비아의 수가 현저히 감소하였고, 최단경로를 산출하는 경매 알고리즘을 형상기반 기법에 적용하여 시스템을 구축하였기 때문에, PCB 배선 경로 탐색을 위한 배선 메모리 영역이 감소됨을 알 수 있었으며, 그에 따른 배선 속도의 향상과 PCB 설계비용이 동시에 감소됨을 알 수 있었다.

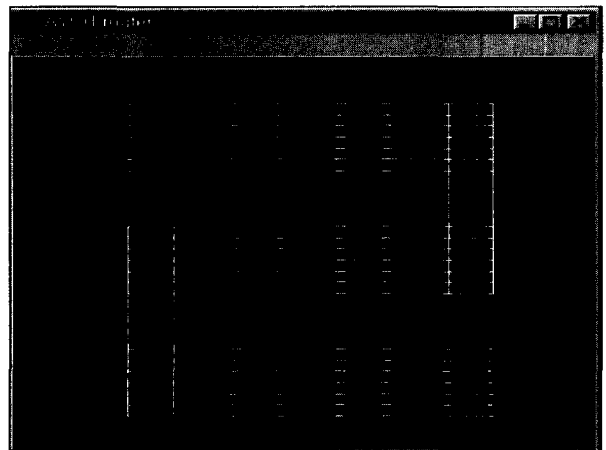


(그림 13) PCB 배선기 시스템 구성도



(그림 14) 메모리 보드 배치 결과

또한 비교 분석한 대부분의 시스템들은 그리드 배선과 형상기반 배선을 혼용하여 100%의 배선율은 나타내었다.

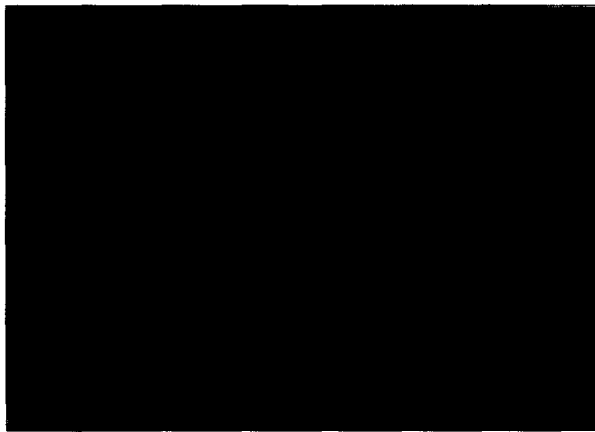


(그림 15) 메모리 보드 배선 결과





(a) 넷 연결도



(b) 배선 결과

(그림 16) 넷 수 200개인 PCB 라우팅 최종결과

〈표 1〉 자동 배선/배치 시스템의 실행결과

측정형태 \ PCB 종류	제안 PCB	PCB1	PCB2	PCB3
그리드 배선 메모리 영역	4.7 K/b	8.2 K/b	8.4 K/b	8.7 K/b
형상기반 메모리 영역	21 byte	40 byte	36 byte	96 byte
배선층	4	4	4	4
Via 수	12	20	36	30
그리드배선(%)	0	30	38.4	36.2
형상기반 배선(%)	100	70	61.6	63.8
총 계 (%)	100	100	100	100
총 배선시간(초)	3	12	20	36

### 5. 결 론

본 논문에서는 PCB 설계시 종전 알고리즘을 적용하여 구현한 방법들에서 제기되는 문제점인 배선 영역에서 많은 메모리 사용으로 인한 메모리 낭비와 배선 속도를 개선하기

위하여 단일 원점에서 여러 목적지에 도달할 때 최단경로를 산출하는 경매 알고리즘을 넷들의 배선요구를 받아들여 자동으로 그 경로를 찾아 자동 배선하는 형상기반 방식에 적용하여 PCB 자동 배선 시스템을 개발하여 PCB성능을 향상시켰다.

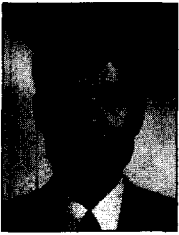
배선시 문제는 보드의 배선 영역을 그리드 방식만을 적용할 경우 M×N 배열의 형태로 메모리 증가로 인해 탐색 시간이 증가하는 문제점을 갖고 있어 본 논문에서 제기한 알고리즘을 (그림 7)에 있는 Function 칼럼에 있는 자동 배선 순서와 (그림 13)과 같이 자동 배선/배치 시스템의 상호동작 구조와 같이 적용한 결과 초기 배선시에 패드당 하나의 shape로 간주하기 때문에 2byte 메모리 영역을 소요하지 않으므로 배선 속도를 향상시킬 수 있었다.

따라서 본 논문에서 구현한 자동 배선/배치 시스템은 최단경로를 산출하는 경매 알고리즘을 형상기반 방식에 적용하여 구현하였기 때문에 PCB 배선 경로 탐색을 위한 배선 메모리 영역이 <표 1>에서와 같이 감소되는 것을 알 수 있었고, 그에 따른 배선 속도와 설계 비용이 동시에 감소됨을 알 수 있었다.

앞으로의 연구과제는 Shape 검색 속도를 빠르게 할 수 있는 알고리즘 개발과 긴 배선에 대처할 수 있는 DB를 구축하여 PC 상에서도 강력하게 사용할 수 있는 super 자동 배선/배치기 시스템을 구현하는데 있다.

### 참 고 문 헌

- [1] D. Hoey, C. E. Leiserson, "A layout for the shuffle-exchange network," Proc. of the 13th Annual ACM Sym. on the Theory of Computing, pp.334-341, 1981.
- [2] E. Berkean and E. Kinnen, "IC layout planning and placement by dimensional relaxation," IEEE Int Conference on Computer Design, pp.449-451, 1985.
- [3] John H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan, 1992.
- [4] Bertsekas, D. P., "A The Auction Algorithm for Shortest Paths," SIAM H. on Optimization, Vol.1, pp.425-447, 1991.
- [5] Hightower, David, "The interconnect Problem : A Tutorial," Proceedings of the Design Automation Workshop, IEEE, pp.1-21, 1973.
- [6] Pallottino, S. and Scutella', M. G., "Strongly polynomial Algorithms for Shortest Paths," Dipartimento di Informatica Report TR-19/91, University of Pisa, Italy, 1991.
- [7] Bertsekas, D. P., Linear Network Optimization: Algorithms and Codes, M. I. T. Press, Cambridge, Mass., 1991.
- [8] Watanabe, Hiroyuki, "IC Layout Generation and Compaction using Mathematical Optimization," The University of Rochester, Ph. D. 1984.



### 김 현 기

e-mail : kim@hyungi.com

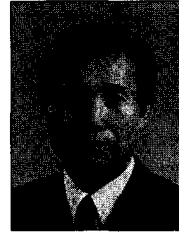
1986년 호서대학교 정보통신과(공학사)

1992년 호서대학교 대학원 정보통신과  
(공학석사)

2002년 청주대학교 대학원 전자공학과  
(공학박사)

1996년~현재 극동정보대학 정보통신과 부교수

관심분야 : VLSI & CAD, 실시간 정보처리, 컴퓨터 네트워크



### 우 경 환

e-mail : khwoo@joongkyung.ac.kr

1977년 청주대학교 졸업

1986년 숭실대학교 대학원 전자계산과  
(공학석사)

2001년 청주대학교 대학원 전자공학과  
(공학박사)

1990년~현재 우송공업대학 부교수(의료복지계열)

관심분야 : VLSI & CAD, PCB설계, 실시간 처리