

PCB 자동 배치 시스템의 설계

(Design of an Automatic Placement System for PCBs)

張銘洙*, 李樟淳*, 黃善泳**

(Myung Soo Jang, Jang Soon Lee and Sun Young Hwang)

要約

본 논문에서는 PCB 설계 시스템 구축의 한 부분으로 PCB에 장착되는 부품과 그들의 연결 관계인 네트 리스트를 받아들여 자동으로 최적의 부품 위치를 출력시켜주는 배치 시스템의 구현에 대하여 기술한다. 수정된 cluster 생성 알고리즘을 사용한 초기 배치 과정을 통하여 배치된 부품들은 GFDR (Generalized Force Directed Relaxation) 알고리즘을 사용한 배치 개선 과정을 거쳐 교환되며 배치 후처리 과정에서 부품 회전 또는 핀과 게이트의 교환을 통하여 최적화 된다. 실험 결과 구현된 배치 시스템이 총 배선 길이를 줄이면서 제조 가능한 결과를 출력함을 보였다.

Abstract

This paper presents the design of a placement system integrated in PCB design system, to get an optimal component positioning from part and net list. Unplaced components are placed in initial process using modified cluster development algorithm and are swapped in improvement process using the GFDR (Generalized Force Directed Relaxation) algorithm. The result is optimized in post process by component rotating or pin/gate swapping. Experimental results show that the placement system produces manufacturable layouts which are optimal in terms of total routing length.

1. 서론

최근 반도체와 관련된 전자 기술이 급속도로 발전하고 전자 기기들의 소형화, 다기능화, 고성능화가 이루어짐에 따라 전자 부품의 지지대 및 신호의 연결 매체로서 중요한 역할을 하는 PCB (Printed Circuit

Board) 설계의 복잡도 또한 증가하고 있으며, 전자 제품의 라이프 사이클이 단축됨에 따라 설계 기간의 단축이 중요한 요인으로 등장하게 되었다. 표면 실장 기술의 일반화와 다층 기판의 사용 등으로 인하여 과거의 숙련된 기술자에 의해 테이핑 방식의 아트웍으로 이루어지던 PCB 설계 방식은 더 이상 사용이 불가능하게 되었고 설계 기판의 재사용과 설계 기간을 줄이기 위해 CAD의 도입과 설계과정의 자동화가 빠른 속도로 이루어져 왔다. PCB 설계 과정의 중요한 요소인 부품의 배치를 자동으로 수행할 수 있는 알고

* 準會員, ** 正會員. 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)

接受日字: 1993年 1月 21日

리듬의 연구가 이루어져 많은 배치 알고리즘들^{1), 2), 3)}이 제안되었다. 그러나 이들 알고리즘들을 응용한 상용 PCB 레이아웃 시스템들을 살펴보면 여전히 설계자의 경험에 의존하는 수동 배치 또는 대화식 배치 방법이 선호되는데⁴⁾, 그 이유는 단순히 총 배선 길이를 최소화하거나 배치 밀도를 고려하는 기존의 알고리즘들이 PCB 제조 공정의 특성을 충분히 고려하지 않아 출력하는 결과가 설계자를 만족시키지 못하기 때문이다. PCB 설계에서의 부품 배치 문제는 다양한 크기의 부품을 배치한다는 측면에서는 VLSI 설계시 마크로 셀 배치 문제와 동일하지만 미리 부품의 위치가 기판 사양과 함께 정해지는 커넥터 같은 부품이 존재하고 자동 삽입기나 손에 의하여 부품이 장착되기 때문에 제조와 검사가 용이하도록 일정한 간격을 두고 일렬로 배치되는 특성 때문에 VLSI 배치 알고리즘을 PCB 배치 문제에 그대로 적용시킬 수 없다. 본 연구실에서는 PCB 레이아웃 시스템인 SPlayS (Sogang Pcb LAYout System)을 구현하였다. 구현된 시스템은 대화식 레이아웃 편집기와 자동 도구들과 이들을 지원하는 부품 라이브러리로 구성되어 있다. 본 논문에서 기술되는 자동 배치 시스템은 SPlayS의 자동 도구의 한 부분으로 설계자에 의한 수동 배치를 허용하고, 표면 실장 부품의 양면 배치를 지원한다. 구현된 자동 배치 시스템에 쓰인 알고리즘들은 PCB 제조 공정을 고려하여 총 배선 길이가 다소 늘어나더라도 제조와 검사가 용이한 배치 결과를 생성한다.

본 논문에서는 SPlayS에 사용된 자동 배치 시스템의 배치 전략을 설명하고 시스템을 구성하고 있는 초기 배치, 배치 개선 그리고 배치 후처리 과정과 사용된 알고리즘에 대하여 기술한다. 실험 결과로 각 과정의 실행 예를 통하여 총 배선 길이가 감소함을 보이며, 결론에서 제조 가능한 레이아웃 설계를 위하여 필요한 추후 연구에 대하여 기술한다.

II. 배치 전략

1. 속성에 따른 부품 배치

PCB 설계 시에 배치란 여러 가지 배치 제약 조건들을 지키면서 기판에 정의된 배치 장소에 부품을 할당하는 과정이다. PCB의 제작 과정에서 부품은 DIP, SIP, PLCC 등으로 대부분 규격화되어 있고 자동 삽입기나 손에 의해서 기판에 장착된다.¹⁾ PCB 설계시 부품 배치는 이를 고려하여 비슷한 크기의 부품은 일정한 행 또는 열로 배치하는 것이 기계나 손이 효율적으로 작업할 수 있어 마크로 셀 배치같이 최적의

위치에 셀을 배치하는 방식은 적합하지 않으며 미리 정의된 격자의 교차점에 부품을 배치하는 방식이 사용된다.¹⁴⁾ 격자의 교차점은 최소한 배치될 부품과 같거나 많아야 한다. 이때 다양한 크기의 부품이 존재하므로 단일 격자를 사용하는 것은 효율적이지 않으며, 격자의 셀 크기를 크게 하면 모든 부품이 배치될 수 없는 경우가 생기고 셀 크기를 작게 하면 부품이 겹치거나 실행 시간이 증가하게 된다. 기존의 시스템은 다중 격자를 사용하여 이를 해결하며 모든 부품에 대하여 적절한 격자들을 설정하는 작업은 설계자의 경험에 의존한다. 일반적으로 숙련된 설계자가 수동으로 부품 배치를 할 경우 비교적 크기가 큰 IC류는 넓은 간격을 두고 정렬하여 배치하고 그 사이에 크기가 작은 수동 부품들이 들어가는 방법을 쓴다.

SPlayS에서는 배치될 부품의 속성("FIX", "MATRIX", "FREE")에 따라 다른 배치 방법을 사용함으로써 제조 가능한 배치 결과를 얻는다. 기판 모양이 완성되면 함께 위치가 결정되는 커넥터 및 설계자에 의해 수동 배치된 부품들의 속성이 "FIX"로 설정되며, "FIX"의 속성을 갖는 부품은 자동 배치를 수행할 때 고정되며 점진적(constructive) 방법을 쓰는 초기 배치 시에 seed 부품으로써 작용한다. "MATRIX"의 속성을 갖는 부품은 설계자가 설정한 격자의 교점에 배치된다. 주로 IC류가 이 속성을 가지는데 기판에 실장되는 부품 중 IC류는 개별 소자에 비하여 크기가 크고 갯수는 적은 경우가 많다. 이러한 특성 때문에 배치 격자의 교점 숫자도 전체 부품을 고려했을 때보다 적어도 되고 빠르게 실행할 수 있으며 설계자가 원하는 일정한 행과 열을 따라 부품이 배치되는 결과를 얻는다. "FREE"의 속성을 갖는 부품은 배치될 부품의 전체 숫자와 설계자가 설정하는 배치 인수의 하나인 "Grid Density" 값에 따라 적절하게 생성된 내부 격자에 배치된다. "Grid Density" 값을 증가시키면 격자의 크기가 작아지므로 세밀한 배치가 가능한데 반하여 실행 속도는 떨어진다. "MATRIX" 부품 배치를 "FREE" 부품 배치보다 먼저 실행하기 때문에 "FREE" 부품 배치 시 seed component은 "FIX" 또는 "MATRIX" 속성을 가진 모든 부품이 된다. 따라서 "FREE"의 속성을 가지는 개별 소자들이 커넥터나 IC류 주위에 배치되는 결과를 얻고 배치 금지 영역과 수동 배치된 부품 영역에 설정되는 격자에 의해 생길 수 있는 dead space에는 "FREE"의 속성을 가지는 개별 소자들이 들어가므로 기판 활용도를 높일 수 있다. 그림 1은 부품의 속성에 따라 배치가 완료되는 예를 보이고 있다.

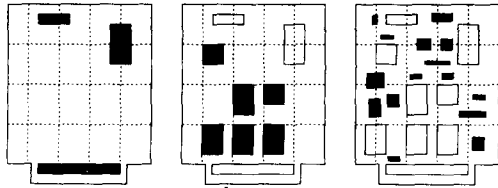


그림 1. 부품 속성에 따른 배치의 예
 (a) FIX 속성 부품 배치
 (b) MATRIX 부품 속성 배치
 (c) FREE 부품 속성 배치

Fig. 1. Placement example with different component attributes.

- (a) Placement of components with FIX attribute.
- (b) Placement of components with MATRIX attribute.
- (c) Placement of components with FREE attribute.

2. 자동 배치의 목표

배치에 있어서 여러 가지 제약 조건 중에서 구현된 배치기는 배치 허용 영역, 배치 불허 영역, 부품 간의 최소 허용 간격등의 설계 제약 조건과 단면 또는 양면 배치, 부품 간의 중첩 허용 등의 기술 제약 조건을 고려하여 배치를 수행한다. 또한 제약 조건이 만족되는 상황에서 전체 네트 길이를 최소화하면서 전체 기판에 네트가 고르게 분포될 수 있도록 배치한다. 이것은 네트 길이가 짧을 수록 배치 후에 배선기에 의해 실제 배선이 짧아지며, 네트가 고르게 분포되어 있을 수록 배선 실패의 확률이 작아지기 때문이다.

배치에는 부품의 위치를 지정해주는 작업과 함께 부품의 배치면, 부품 방향을 정해주는 작업을 해야 한다. SPLayS는 기본적으로 부품의 배치면과 방향을 편집기에서 수동 배치시에 얻지만 자동 배치시에 최적의 결과를 얻기위해 동적으로 수정된다. 최근에는 관통 핀 부품의 대부분이 표면 실장 부품 형식으로도 제조되기 때문에 양면 배치의 지원이 필수적이다. 구현된 배치기는 단면 배치 또는 양면 배치가 모두 가능하며 양면 배치 시에는 표면 실장 부품에 한하여 최적의 결과를 얻을 수 있도록 배치될 면을 자동 결정한다. 부품을 배치시에 회전시켜가면서 시도하는 경우 콤팩트한 배치가 가능하지만 부품의 방향이 일정하지 않으면 파워 배선과 자동 삽입에 어려움이 있으므로 "MATRIX"의 속성을 갖는 부품을 제외한 부품들이 자동 배치시에 회전을 시도한다.

3. 네트 배선 길이의 예측

실제 배선 길이를 줄이는 것이 배치에 있어서 중요한 요소이나 배치 단계에서는 배선 길이를 알 수 없으므로 네트의 각 점들을 가지고 근사 계산법을 통해 총 네트 배선 길이를 예측하게 된다. 네트 배선길이의 근사 계산법으로 MST (minimal spanning tree), Steiner tree, half perimeter, Euclidian distance, Manhattan distance 등이 있다.¹² MST와 Steiner tree는 다른 방법에 비하여 실제 배선 길이와 근사한 결과를 보이지만, 실행 시간이 상대적으로 늦어 배치 과정에서 반복적으로 쓰이는 총 배선 길이의 계산에는 적합하지 않다. 또한 half perimeter는 standard cell이나 gate array같이 배치 영역과 배선 영역이 구분되어져 있는 방식에는 적합하지만 PCB같이 배선이 부품 내로 지나가는 경우는 적당하지 않다. SPLayS에서는 전처리 과정에서 네트를 MST 방식으로 재정리하여 tree의 각 버텍스를 커넥션으로 할당하는 작업을 수행한다. 이 결과 네트는 2점 커넥션으로 분해된다. 그림 2에서 이 과정을 표시하였다. 배치 과정에서는 커넥션의 Manhattan distance를 계산하여 배선 길이로 삼는다.

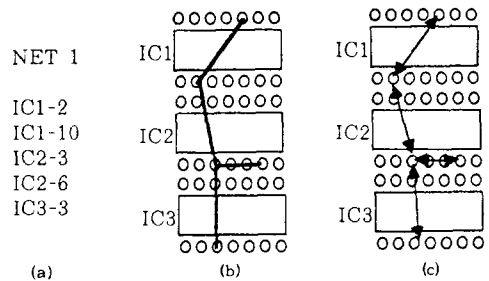


그림 2. 네트를 커넥션으로 분리하는 과정
 (a) 네트리스트 (b) MST (c) 커넥션

Fig. 2. Making connections from nets.
 (a) Net list, (b) MST, (c) Connections.

SPLayS에서 사용한 방법은 계산 시간이 오래 걸리는 MST 방식을 한번만 사용해 근사적인 네트 길이를 얻고, 계산이 간단한 Manhattan distance 방식은 배선 길이를 수시로 재계산해야 하는 배치의 반복부에 사용함으로써 두 방식의 장점만을 취할 수 있다. 배치 개선은 정해진 반복 횟수에 따라 여러번 시행할 수 있는데 한번의 배치 개선이 끝나면 다음 반복에 들어가기 전에 MST 방식을 사용하여 네트를 재정리함으로써 배선 길이의 부정확한 예측을 막고

새로운 배치 개선을 시도할 수 있다. 구현된 배치 개선부의 알고리즘인 GFDR (Generalized Force Directed Relaxation)이 부품의 핀에 연결된 커넥션들을 이용하여 force를 계산하므로 네트 전체의 길이보다는 커넥션의 길이를 계산하는 것을 빠르게 해야하며, 이 경우 계산에 가산만을 사용하는 Manhattan distance를 사용하는 것이 적합하다. 또, 구현된 배선기도 커넥션을 따라가며 orthogonal하게 배선하므로 Euclidian distance 보다 Manhattan distance가 실제 배선 길이에 가깝다.

4. 커넥션의 교차도

기존의 PCB 부품 배치는 부품이 기판에 고르게 배치되도록 하는 것이 중요한 목표였으나 다층 기판과 표면 실장 부품 이용에 따른 설계 기술의 발달로 기판의 면적이 부품의 총 면적과 비슷한 최근에는 이의 중요도가 떨어지게 되었다. 따라서 구현된 배치기에서는 기판 밀도 대신에 부품에 달려 있는 커넥션의 교차도를 고려한다. 커넥션의 교차도는 한 부품에 달려있는 각 커넥션에 대하여 다른 커넥션과 교차되는 횟수를 모두 더한 값으로 정의된다. 커넥션 교차도가 중요한 요인이 되는 이유는 핀이 일정 간격으로 행 또는 열로 늘어서 있고 이 사이를 일정 굵기의 도선이 지나가는 PCB의 특성 때문이다.

그림 3 (a)와 그림 3 (b)의 배선 전의 커넥션 길이의 합은 서로 같으나, 배선 후를 비교하면 그림 3 (a)은 그림 3 (b)보다 배선 길이도 길고 비아 삽입과 다층이 없으면 배선이 불가능하다. 구현된 배치기는 이 점을 고려하여 부품을 배치할 때 커넥션이 가장 겹치지 않는 곳에 배치가 이루어진다.

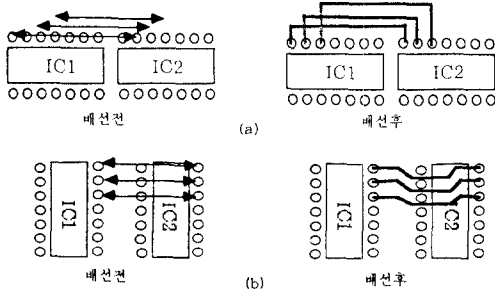


그림 3. 커넥션의 교차도에 따른 배선의 예
(a) 교차도 3 (b) 교차도 0

Fig. 3. Routing example by degree of intersections of connections.

- (a) Degree of intersections 3.
- (b) Degree of intersections 0.

Ⅲ. 초기 배치

1. 기존 배치 알고리즘 분석

최근 들어 배치 문제는 전처리 과정인 초기 배치와 후처리 과정인 배치 개선의 두 단계로 나누어 해결하는 방법을 일반적으로 사용하고 있다.^[3] 배치 개선은 모든 부품이 기판 위에 배치되어 있다고 가정하고 이루어지므로 초기 배치는 모든 부품이 기판 위에 배치가 가능하도록 보장하는 동시에 배치 개선부의 부담을 덜어 주기 위해 지역적인 최적화가 이루어지는 방향으로 진행되어야 한다.

일반적으로 일부분의 부품이 배치되고 나머지는 이동 가능한 부품으로 구성된 부품 위치 리스트를 입력으로 받아들여 모든 부품의 배치를 결정하여 출력하는 점진적 배치 방법이 초기 배치에 쓰이며 이 배치 방법들은 cluster growth 배치 방식, 분할 배치 방식, 전역적 배치 방식, branch-and-bound 배치 방식 등으로 크게 분류된다.^[3] 이 중 분할 배치 방식과 전역적 배치 방식은 좋은 결과를 얻는 반면 수행 시간이 길고 위치에 대한 제약 조건이 많을 경우 최적의 결과를 얻지 못할 수도 있으므로 마크로 셀과같이 배치가 하나의 rectangle 안에서 이루어지고 compact한 결과가 필수적인 응용 분야에 쓰인다. Branch-and-bound 배치 방식은 부품의 수가 증가하면 수행 시간이 현실적이지 못하기 때문에 실제로 넓게 쓰이지 않는다. Cluster growth 배치 방식은 seed component의 선택에 민감한 결과를 얻고 전체 배치면에 고르게 배치되지 못하고 seed를 중심으로 배치되므로 배선율이 떨어지는 단점이 있으나 빠른 수행 시간과 자동 배치 전에 일부 부품의 위치가 정해지고 기판의 모양이 일정하지 않으며 부품이 격자의 교차점 위에 배치되는 PCB설계의 특성이 위의 단점을 보상해 주기 때문에 초기 배치 알고리즘으로 적당하다.

SPLayS에서는 cluster growth 알고리즘의 하나인 cluster 생성 방법을 기반으로 PCB설계에 적합하도록 개선된 알고리즘을 이용하여 초기 배치를 수행한다.

2. Modified Cluster Development 알고리즘

Cluster 생성 알고리즘은 배치 안된 부품 중 하나를 배치된 부품과의 연결도를 고려하여 선택한 뒤 연결 길이를 줄일 수 있는 적당한 빈자리에 배치하는 작업을 모든 부품이 배치될 때까지 되풀이한다. Odawara의 CILP^[5]에 제안된 알고리즘은 다중 seed를 지원하며 각 seed 부품을 중심으로 cluster 생성이 독립적으로 동시에 진행된다. 따라서 설계자

가 자동 배치 전에 수동으로 여러 개의 부품을 배치하는 것이 가능하며, 이들 부품들은 자동 배치 시에 seed로 이용되므로 자동 배치의 결과에 설계자의 의도가 반영되는 이점을 가진다. 일정 숫자의 부품을 선택하여 이들의 최적 위치를 고려하여 한번에 배치하기 때문에 한번에 하나씩 배치할 경우보다 지역 최적화를 이룰 수 있다. 완전 배치를 위해 크기가 큰 부품이 우선권을 가지므로 두 부품이 겹쳐서 배치되면 크기가 작은 부품은 배치 장소에서 제거되어 다음에 다시 배치된다.

CILP에서 구현된 알고리즘은 배치 격자 크기보다 작은 부품의 배치에 Kuhn의 linear assignment 알고리즘⁶⁾을 사용하며 이 방법은 비교적 좋은 결과를 얻을 수 있는 장점이 있지만 구현이 어렵고 수행 시간이 길며 적용되는 부품 간에 연결이 존재하면 최적 해를 얻을 수 없는 등 단점이 많아 빠른 수행 속도를 얻어야 하는 초기 배치에 쓰기에는 적합하지 않다.

SPLayS에서 구현된 modified cluster 생성 알고리즘은 미리 정해진 숫자만큼 candidate 부품들을 선택하는 과정과 선택된 부품들을 선장된 격자 크기보다 큰 부품인 over size-cell 부품과 작은 부품인 single-cell 부품으로 나누어 배치하는 과정으로 크게 나누어진다. Candidate 부품의 선정에는 미 배치 부품 중 배치가 완료된 부품과의 연결도를 조사하여 그 값이 큰 순으로 정한다. 이 때 쓰이는 연결도 함수는 식 (3.1) 같이 정의된다. 여기서 seed 부품의 레벨은 0이며 clustering이 진행될 수록 레벨 값(L)은 증가한다. 따라서 레벨이 낮은 부품은 높은 부품보다 먼저 배치 되었음을 알 수 있다.

$$connectivity(m_i) = \sum_{k=0}^L \frac{1}{2^k} * C(m_i, k) \quad (3.1)$$

- C(m_i, k) : 부품 m_i에 달린 커넥션 중 레벨 k 부품에 연결되는 커넥션의 수
- L : 현재 레벨 값

이 함수에서 계수 1/2^k 때문에 하위 레벨의 커넥션이 더 중요하게 중요하게 작용된다. 다시 말해서 같은 커넥션 수를 가진 두 부품이 있다면 먼저 배치된 부품과의 관계가 밀접한 부품이 우선권을 가지며 이는 다중 seed에 의해 clustering이 동시에 되기 위해 필요한 계수이다. 부품의 배치는 각 교차점마다 부품을 입시로 배치한 뒤 비용 함수를 적용하여 가장 비용이 적게 드는 교차점부터 배치 제약 조건을 검사하여 만족하는 교차점에 부품을 위치시키게 된다.

이때 사용되는 비용 함수 C(x, y)는 구현된 배치기의 모든 알고리즘에 공통적으로 쓰이며 식 (3.2)와 같은데 비용이 적을 수록 짧고 쉬운 배선이 가능하다. 부품이 좌표 (x,y)에 놓이면 그 부품에 달려 있는 커넥션들의 좌표 값도 이에 맞게 수정한 뒤 비용 함수를 사용해야 한다.

$$C(x,y) = \alpha * conn_length + \beta * num_of_intersect \quad (3.2)$$

α, β : 계수

conn_length : 한쪽은 부품 m_i에 달려 있고 다른 한쪽은 부품 m_i(i≠j)에 달려 있는 커넥션들의 거리 * 가중치 의 합

num_of_intersect : 한쪽은 부품 m_i에 달려 있고 다른 한쪽은 부품 m_i(i≠j)에 달려 있는 커넥션들이 교차된 숫자

Candidate 부품 중 over size-cell 부품을 모두 배치한 뒤 나머지 single-cell 부품과 over size-cell 부품 배치 시 제거된 single-cell 부품을 합쳐 배치한다. 배치하는 방법은 over size-cell 부품 배치와 single-cell 부품 배치 때 같은 방법을 사용한다. 그림 4에 제안된 수정된 cluster 생성 알고리즘을 개략적으로 나타내었다. 배치할 부품 리스트와 배치 가능 장소를 표시한 그리드 리스트를 받아들여 각 레벨마다 배치할 부품 갯수를 정한다. 먼저 레벨 0 부품인 seed 부품과의 연결도를 계산하여 가장 높은 값을 가지는 부품을 위에서 미리 정한 갯수만큼 선택한다. 선택된 부품들은 크기에 따라 over size-cell 부품과 single-cell 부품으로 나뉘고 over size-cell 부품을 배치 비용이 가장 적은 곳에 배치한 뒤 single-cell 부품을 같은 방법으로 배치한다. 선택된 부품이 배치 성공한 경우 레벨 1 부품으로 설정되고 레벨은 2가 된다. 다시 레벨 0 부품과 레벨 1 부품과의 연결도를 계산하여 미 배치 부품들이 선택되고 위의 배치 과정을 반복한다. 최소 비용 장소에 미리 배치된 부품 때문에 배치가 불가능한 경우 양면 배치가 허용되면 다른 면에 배치를 시도한다. 만일 모든 시도에서 배치 장소가 정해지지 않으면 그 부품은 제외시키고 계속 진행되며 알고리즘은 모든 부품이 배치 또는 제외되는 경우 배치 성공한 부품의 위치와 제외된 부품 리스트를 출력하고 종료된다. 이때 제외된 부품들은 배치 제약 조건이 너무 강하기 때문에 배치에 실패한 부품들이므로 배치 허용 영역을 넓히거나 격자 수를 늘리는 등, 배치 제약 조건을 완화하여 다시 초기 배치를 수행시키거나, 편집기를 사용하여 설계자가 수

등으로 배치시켜 모든 부품의 배치를 완료한다. 크기와 연결도가 작은 부품이 뒤늦게 배치되는 알고리즘의 특성 때문에 만일 초기 배치에 실패한 부품이 있다고 해도 쉽게 수동 배치가 가능한 부품만이 존재하게 된다.

```

procedure Modified_Cluster_Development(Component_List, Grid_List)
begin
  Determine the number of candidate components :
  nlevel := 0 ; ( level 0 components are seed components )
  Set Placed Comp_List as empty
  repeat
    Calculate connectivity with placed components :
    Select candidate components having higher connectivity :
    if any candidate component exists then begin
      ( place over-cell components )
      Inmark all candidate components :
      repeat
        Select largest component among unmarked candidates :
        if selected component is over cell component then begin
          Place the component at minimal cost grid :
          if no grid is possible then
            the component is ignored :
          Mark the component :
          if smaller components is overlapped then
            Remove smaller components :
        end
        Mark the component :
      until all candidate components are marked :
      ( place single-cell components and removed smaller components )
      Inmark all candidate components except for over cell components :
      repeat
        Select the component having highest number of connections :
        Place the component at minimal cost
        and non-overlapped grid :
        if no grid is possible then
          the component is ignored :
        Mark the component :
      until all candidate components are marked :
      ( prepare next step )
      nlevel := nlevel + 1 ;
      Set level of candidate components as nlevel :
      Put candidate components to placed comp_list :
    end
  until all components are placed or ignored :
end

```

그림 4. 수정된 cluster 생성 알고리즘

Fig. 4. Modified cluster development Algorithm.

IV. 배치 개선

초기 배치에서는 배치가 끝난 부품과의 관계만이 고려할 수 있는 유일한 배치 정보이므로 최적 해를 구하기는 불가능하다. 배치 개선부에서는 완료된 부품 배치를 입력으로 받아 좀 더 짧은 총 배선 길이를 가지는 부품 배치를 출력하고, 양면 배치가 가능한 기판이라면 배치 개선을 위하여 부품의 배치 면을 바꾸는 작업 또한 같이 진행한다.

배치 개선에 쓰이는 알고리즘은 일정한 정지 조건에 이를 때까지 부품의 선택과 배치를 반복적으로 수행하는 특징을 가지고 있으며 크게 나누어 interchange, force directed method, unconnected set, simulated annealing 등의 방법이 있다.³⁾ Interchange는 전역적인 고려가 어렵고 unconnected set 방식은 동작 속도면에서 효율적이지 못

하여 최근에 거의 쓰이지 않는 방법이며 simulated annealing은 일반적으로 가장 좋은 해를 찾지만 수행 시간이 매우 오래 걸리고 평형 조건(equilibrium condition)을 설정하는 것도 어렵다. SPLayS에서 구현된 배치는 배치 개선에 빠르면서 비교적 좋은 결과를 얻을 수 있는 force directed 방법⁴⁾의 하나인 Goto의 GFDR 알고리즘⁵⁾을 PCB 배치에 적합하도록 보완하여 사용한다.

1. Force의 계산

한 부품의 force는 그 부품과 연결되어 있는 다른 부품과의 위상적인 관계로 계산된다. 구현된 배치는 2점 직선인 커넥션으로 부품들의 관계가 정해져 있으므로 이를 이용하여 직접 force를 계산할 수 있으며 그 부품의 중심점과 커넥션의 두 점 중 다른 부품에 연결된 점의 쌍이 벡터가 된다. Force는 이들 벡터의 합이므로 x, y에 대하여 나누어 구한다. 식(4.1)은 X 방향 force 벡터를 구하는 식이다. Y 방향도 같은 방식으로 구한다. 이 계산 식을 보면 설계자가 설정한 커넥션의 가중치가 고려되며 가중치는 1 이상의 값을 가진다.

$$force_x(mi) = \frac{\sum_{j=0}^K (X_{ij} - X_{mi}) W_j}{\sum_{j=0}^K W_j} \quad (4.1)$$

- K : 부품 mi에 달려 있는 커넥션 중 내부 커넥션을 제외한 나머지 커넥션들의 수
- X_{ij} : 커넥션 j의 두 점 중 부품 mb($i \neq b$)에 달려 있는 점의 X 좌표
- X_{mi} : 부품 mi의 중심점의 X 좌표
- W_j : 커넥션 j의 가중치 ($W_j > 0$)

2. 교환 집합을 구하는 알고리즘

GFDR 알고리즘은 기존의 force directed method들이 반복 과정에서 한번에 하나 씩의 부품만을 고려하는 이유로 지역적 최적화에 국한될 수밖에 없는 단점을 보완하여 탐색 트리 형태의 부품 선택을 통하여 전역적인 최적화에 접근할 수 있고, 탐색 트리의 깊이와 tree branch에 달리는 leaf의 숫자를 설계자가 결정함으로써 수행 시간과 결과의 질 사이의 trade-off를 조정할 수 있는 장점이 있다. GFDR 알고리즘은 모든 부품의 크기가 동일하다는 가정을 가지고 전체 부품 중 일부를 취하여 이들의 위치를 교환하는 방법으로 진행되므로 여러 가지 크기의 부품들이 배치 대상이 되는 PCB 부품 배치에

직접 적용할 수 없다. 구현된 알고리즘은 다양한 크기의 부품과 양면 배치를 지원하도록 보완되었다.

제안된 알고리즘은 크게 세 부분으로 나누어진다. 먼저 부품의 force를 개별적으로 구하여 force가 가장 높은 것을 선택하고, 선택된 부품부터 시작되는 교환 집합을 구한 뒤, 이들의 교환이 전체 비용을 줄이지 못하면 교환하지 않고 실패했음을 기록한다. 이 과정들은 모든 부품에 대하여 시도한 결과가 모두 실패할 때까지 계속 반복된다.

부품의 선택은 식 (4.1)에 기술한 force 벡터 계산식을 이용하여 나온 force 벡터의 크기를 구한 뒤 가장 값이 큰 부품을 선택하여 이루어진다. 교환 집합을 구하는 알고리즘은 GFDR의 핵심으로 미리 설정된 탐색 트리의 폭(ϵ)과 깊이(λ)를 만족하는 범위에서 서로 교환 가능한 부품의 집합을 구한다.⁴

PCB 부품은 임의의 크기를 가지고 있고 양면 배치가 가능하므로 이의 지원을 위해 GFDR 알고리즘의 보완이 필요하다. 제안된 알고리즘은 임의 크기의 부품을 지원하기 위해 교환 가능한 부품의 집합을 구할 때 새로 이동될 자리에서 부품의 중첩이 일어나지 않도록 미리 검사하고 만일 중첩이 일어나면 중첩이 발생하지 않는 다음 최소 비용 장소로 이동시킨다. 만일 중첩이 일어났을 때 그 집합을 포기하는 방법을 쓴다면 교환할 수 있는 경우가 매우 적어 지므로 일찍 반복이 중단되고 지역 최적해를 구할 가능성이 높다. 따라서 이 경우 다음 최소 비용 장소로 이동시키는 것이 다음 반복 때 새로운 교환 집합을 만들 가능성을 높여준다.

배치 개선 단계에서 부품의 배치면을 동적으로 변화시키기 위하여 부품의 비용을 구할 때 현재 면에서 최소 비용 장소와 다른 면으로 이동시의 최소 비용 장소를 같이 구한 뒤, 두 값을 비교하여 만일 다른 면으로 이동시의 비용이 더 작다면 그 부품의 배치면을 바꾼다. DIP 타입 부품은 관통 핀을 쓰므로 같은 장소에서 DIP 타입 부품끼리 양면 배치는 허용되지 않지만 큰 DIP 타입 부품의 뒷면에 들어가는 작은 SMD 타입 부품은 허용된다.

교환 집합을 구하는 알고리즘은 활성화된 부품의 비용을 가장 줄일 수 있는 장소를 ϵ 개 결정하여 그 장소의 부품들 중 하나를 선택하는 부분과 선택된 부품을 활성화시킨 뒤 λ 횟수까지 위의 선택 작업을 반복하는 부분으로 구성되어 있고 단계별로 설명하면 다음과 같다.

단계 1. Force계산에 의해 seed 부품을 선정한다.

단계 2. Seed 부품을 현재 부품(current component)으로 삼는다.

단계 3. 현재 부품의 비용이 최소가 되는 장소를 찾은 뒤 그 장소가 빈곳이면 단계 9로 간다. 만일 배치 면을 바꿀 경우 빈곳이 되면 현재부품의 배치 면을 바꾸고 단계 9로 간다.

단계 4. 만일 다른 부품이 있다면 이 곳에 현재 부품을 옮기고 다른 부품은seed 부품 장소로 옮긴 뒤 교환 비용을 계산한다. 이때 크기 문제로 중첩 배치가 일어나면 취소한 뒤 단계 3으로 가서 다른 장소를 찾는다.

단계 5. ϵ 횟수가 될 때까지 단계 3과 단계 4를 되풀이 한다. 이때 한번 계산된 최소 비용 장소는 다시 찾지 않는다.

단계 6. 최소 교환 비용을 가지는 장소의 부품을 목표 부품(targetcomponent)으로 삼는다.

단계 7. 목표 부품의 최소 비용 장소를 찾은 뒤 이 장소가 seed 부품의 장소와 일치하면 단계 9로 간다.

단계 8. 장소가 일치하지 않으면 목표 부품을 현재 부품으로 삼고 횟수가될 때까지 단계 3에서 단계 4를 반복한다.

단계 9. 지금까지 선택된 모든 부품들이 구하는 교환 집합이다.

V. 배치 후처리 과정

1. 부품 회전

PCB 제조 과정에서 자동 삽입기나 손에 의해 부품의 삽입이 일어나므로 부품을 일정한 방향으로 배치하는 것이 더 능률적이고 오류 발생을 막아주며 검사하기도 쉽다. 자동 배치에 들어가기 전에 설계자가 부품의 방향을 적절하게 설정했다는 가정을 가지고 초기 배치와 배치 개선 과정에서는 부품의 위치만이 변경될 뿐 부품의 방향은 입력된 방향을 유지한다. 부품 회전은 위와 같은 이유로 공정 상의 어려움을 넘지만 배치 결과를 개선시킬 수 있다. 최근에 진보된 시스템은 임의의 각도로 부품을 회전하여 배치하는 것이 가능하지만 특수한 경우를 제외하면 수평축 또는 수직축과 평행하게 부품이 배치되므로 SPLayS에서는 90도 회전만을 허용한다. 배치 완료된 부품을 하나씩 선택한 뒤, 부품 중심을 기준으로 90도씩 회전하여 계산한 비용을 비교하여 가장 적은 비용을 가지는 방향으로 부품의 방향을 설정한다. 부품의 회전은 power 배선과 같은 뒤의 배선 과정에 부담을 주므로 보통 개별 소자로 구성되는 "FREE"의 속성을

갖는 부품에 한하여 90도씩 회전시키며 비용을 계산하고 "MATRIX"의 속성을 갖는 부품들은 180도의 회전만 허용한다.

2. 핀과 게이트의 교환

스키매틱 캡처로 만든 논리 다이어그램은 ASIC화 하든지 기존 부품을 써서 PCB로 제조되는데, 후자의 경우 논리 수준의 게이트와 핀을 기판 수준의 물리적인 부품에 매핑 하는 과정을 거친다. 규격화와 경제적인 이유로 보통 여러 개의 논리 함수 또는 논리 게이트들이 하나의 IC에 들어 있고, 논리 게이트의 특성 상 입력 핀은 바뀌서 할당해도 기능이 동일한 경우가 있으므로 이들을 동등 게이트, 동등 핀이라 한다. 네트 리스트에서 게이트와 핀의 할당은 임의로 되므로 배치 후에 위치 정보를 가지고 동등 게이트의 교환과 동등 핀의 교환을 통하여 게이트와 핀의 할당을 최적화 시킬 수 있으나 네트 리스트 자체가 수정되므로 스키매틱 캡처로의 back annotation이 필요하다.

구현된 배치기에서는 컴포넌트 라이브러리가 제공하는 매핑 자료를 참고하여 게이트 교환을 행한 후에 핀 교환을 수행한다. 교환 방법은 교환 가능한 쌍에 대하여 교환을 수행하여 배치 비용의 증감을 기록한 뒤 비용 감소가 가장 현저한 교환을 결과로 받아들인다. 교환 방법이 각 게이트 또는 핀별로 순차적으로 시행되므로 교환 순서에 민감한 결과를 얻지만 교환이 IC 내부에서의 자리 바꿈이어서 결과의 차이가 크지 않으므로 간단하게 구현할 수 있는 방법을 사용하였다.

VI. 실험 결과

SPLayS 레이아웃 시스템은 MS-Windows 환경하의 IBM-PC 호환 기종에서 C언어와 Windows SDK를 사용하여 구현하였다. 학술지에 발표된 PCB 레이아웃 시스템이 거의 없고 PCB 분야에서 일반적이고 신뢰할 수 있는 표준 테스트 사양이 알려지지 않은 관계로 타 시스템과의 성능 비교가 어렵다. 본 논문에서는 개발된 시스템의 효용성을 입증하기 위해 다양한 기판을 정의하여 SPLayS로 설계함으로써 사용자 제약 조건을 만족하면서 제조 가능하고 최적화된 배치 결과를 얻어짐을 보였다.

배치 결과를 실험하기 위해서 표 1과 같은 특성을 갖는 기판을 임의로 정의하였다. 기판 A는 소규모 설계의 예로서 부품과 커넥션이 적어 결과를 쉽게 확인할 수 있다. 기판 B는 양면 배치 지원을 실험하기 위한 것으로 커넥션 수가 비교적 많고 표면 실장 부품들을 포함하고 있다.

표 1. 실험에 사용된 기판의 특성

Table 1. Properties of example boards.

	층 수	부품 수	커넥션 수	핀 수
기판 A	2	15	55	224
기판 B	2	62	273	764

기판 A를 입력으로 받아들여 커넥터들은 미리 수동 배치를 사용하여 위치를 결정한 뒤 속성을 "FIX"로 바꾸었다. 표 2에 총 네트 길이의 변화를 제시하였으며 자동 배치가 진행됨에 따라 총 네트 길이가 감소함을 확인 할 수 있다. 랜덤 배치는 가장 가까운 격자로 부품을 순차적으로 이동시키는 배치이다.

표 2. 자동 배치 결과

Table 2. Automatic placement results.

(단위 Inch)

	총 네트 길이
랜덤 배치	81.176
초기 배치	56.775
배치 개선	54.342
배치 후처리	52.694

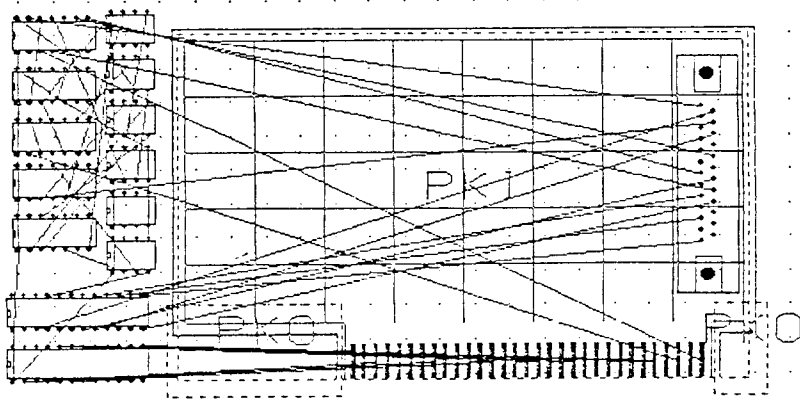
표 3에 배치 개선 시에 GFDR 알고리즘의 중요 인수인 ϵ 와 λ 의 변화에 따른 배치 개선도의 증가를 보였고, ϵ 과 λ 가 커짐에 따라 총 배선 길이는 감소하지만 $\epsilon = 3, \lambda = 3$ 이상 부터는 수행 시간 증가보다 뚜렷한 배치 개선 증가가 일어나지 않아 $\epsilon = 3, \lambda = 3$ 를 배치 인수로서 채택하였다.

표 3. 배치 개선 시에 ϵ 와 λ 의 변화에 따른 배치 개선도의 증가

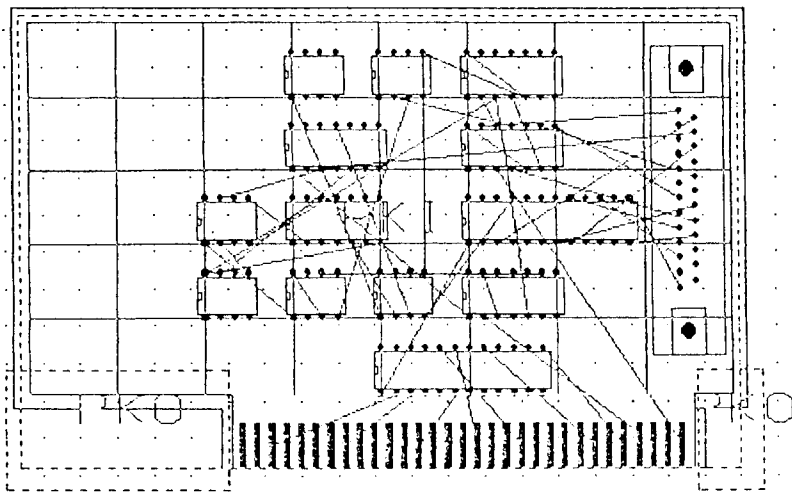
Table 3. Total net length for different values of ϵ and λ in placement improvement.

(단위 Inch)

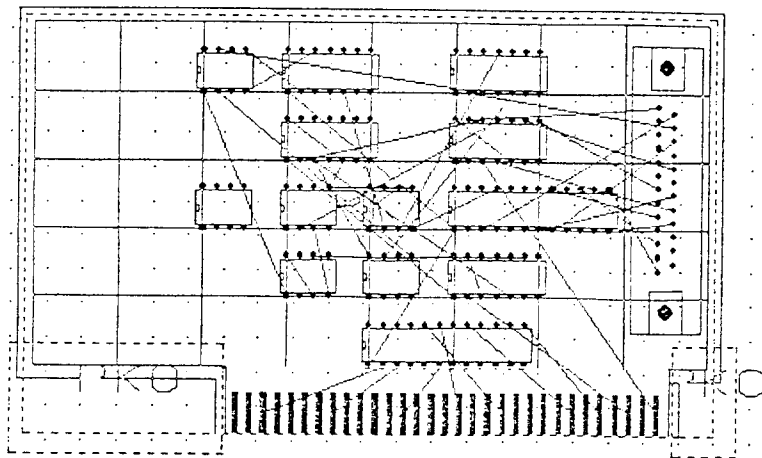
	총 네트 길이
$\epsilon = 0 / \lambda = 0$	64.567
$\epsilon = 3 / \lambda = 3$	60.368
$\epsilon = 6 / \lambda = 6$	60.368
$\epsilon = 9 / \lambda = 9$	59.747



(a)



(b)



(c)

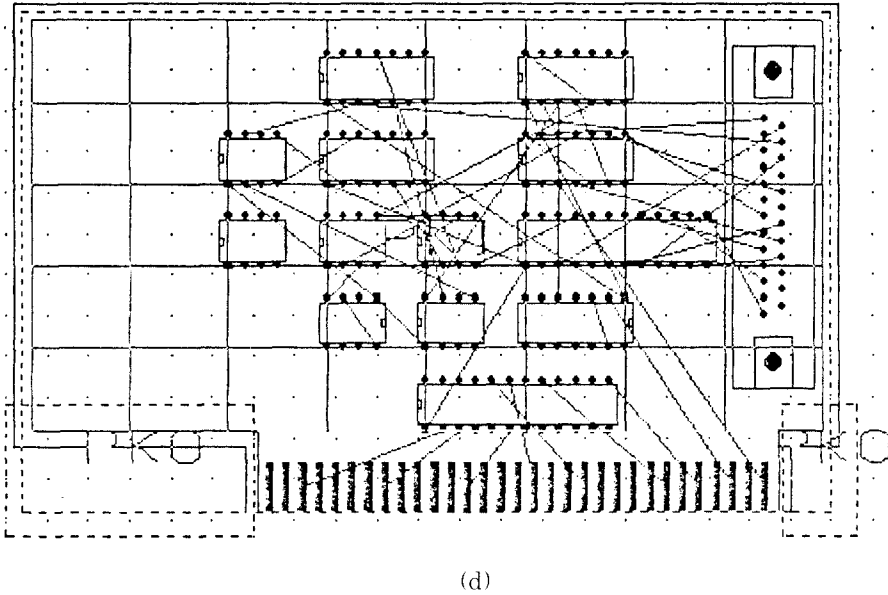


그림 5. 기판 A의 단면 배치 결과
 (a) 수동 배치 (b) 초기 배치 (c) 배치 개선 (d) 배치 후처리
 Fig. 5. Placement result of example board A.
 (a) Manual placement, (b) Initial placement,
 (c) Placement improvement, (d) Placement post processing.

그림 5 (a)는 A기판의 외곽선, 배치 제약 조건과 부품의 배치 속성을 나타내었다. 그림 5 (b)는 초기 배치 후, 그림 5 (c)는 배치 개선 후 그리고 그림 5 (d)는 배치 후처리 과정을 거친 최종 배치 결과이다. 그림 5 (a)와 그림 5 (d)를 비교하면 비교적 크기가 큰 부품들이 격자 상에 정렬되어 있고 그 사이에 작은 부품들이 들어가는 형태로 배치되었고 총 배선 길이를 줄일 수 있도록 최적화 되었음을 알 수 있다. 배치 영역을 최대한 이용하도록 고려된 기존의 알고리즘과는 달리 제안된 알고리즘은 네트의 교차도를 고려하므로 교차도가 심각하지 않은 경우 총 배선 길이를 줄일 수 있도록 밀집 배치가 되지만 만일 교차도가 심각할 경우 배선 길이가 다소 늘어나더라도 전체 영역을 사용한 배치 결과를 얻을 수 있다.

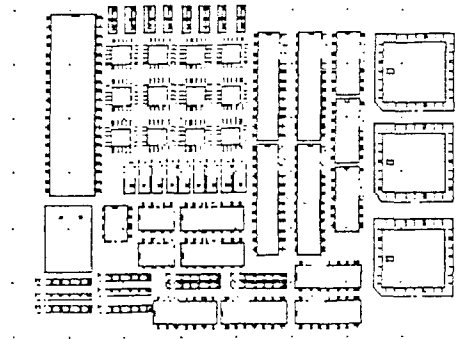
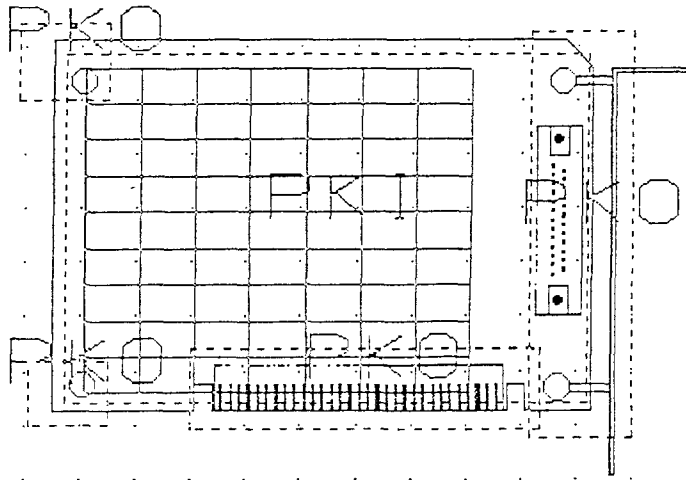
그림 6 (a)의 기판 B를 입력으로 받아 들어 양면 배치를 수행한 결과를 그림 6 (b)에 보였으며, 양면 배치 시에 기판 크기의 감소가 현저히 이루어지는 것을 확인 할 수 있다.

VII. 결론

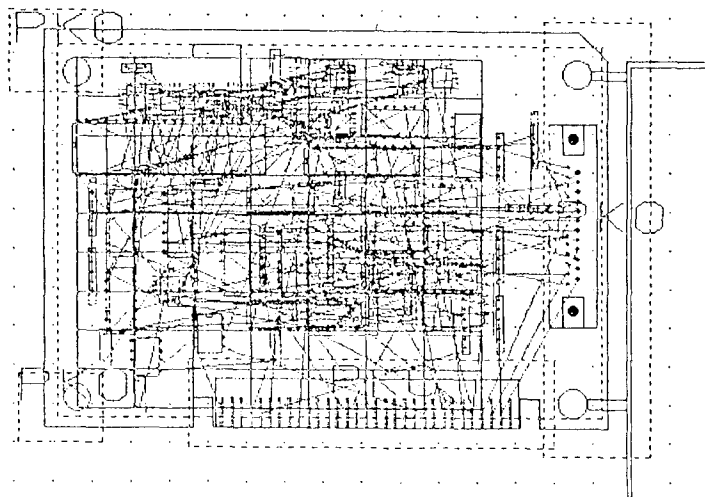
본 논문에서는 PCB 레이아웃 시스템인 SPLaySe에

서 부품의 자동 배치 시스템의 설계 및 구현에 이용된 알고리즘에 대해 기술하였다. 구현된 배치 시스템은 배치에 쓰이는 여러 가지 알고리즘들을 분석한 뒤 효율적인 PCB 레이아웃을 얻기 위해 개선된 알고리즘들을 적용하여 부품의 특성에 따라 다른 배치 전략을 사용하였으며, 최적화된 배치 결과를 얻기 위해 초기 배치, 배치 개선, 배치 후처리의 단계적 처리 과정을 사용하였다. 초기 배치를 위하여 cluster 생성 알고리즘을 양면 배치를 지원하도록 수정하여 사용하였고, 배치 개선을 위하여 GFDR 알고리즘을 바탕으로 PCB 배치에 적합하도록 보완된 절차를 사용하였다.

앞으로 더욱 동작 속도가 빠르고 집적도가 높은 PCB를 설계하기 위해서는 실제 제조된 PCB에서 가장 큰 동작 오류 요인인 기판의 발열과 진동을 설계 단계에서 제거하는 것이 필수적이며 열 또는 진동 해석 프로그램과의 연계로 배치시에 이들 요인들을 고려할 수 있는 배치 알고리즘의 개선이 필요하다. 또한 2차원적인 부품 배치에서 부품의 높이 까지 고려하는 입체적 배치로 확장이 가능한데 이 경우 표면 실장 부품 위로 관통 부품이 배치 되므로 기판의 이용도를 더욱 높일 수 있으므로 이에 대한 연구가 계속되어야 할 것이다.



(a)



(b)

그림 6. 기관 B의 양면 배치 결과 (a) 수동 배치 (b) 배치 완료

Fig. 6. Placement result of example board B. (a) Manual placement, (b) Completed placement.

參考文獻

[1] M. A. Breuer, *Design Automation of Digital Systems*, vol. 1, Prentice-Hall Inc., 1972.

[2] T. Ohtsuki, *Layout Design and Verification*, North-Holland, 1986. pp 1-198.

[3] B. T. Preas and M. J. Lorenzetti, *Physical Design Automation of VLSI Systems*, The Benjamin/Cummings Publishing Co., 1988.

[4] T. J. Byers, *Printed Circuit Board Design with Microcomputers*, Intertext Publications McGraw-Hill Book Company, 1991.

[5] G. Odawara, K. Iijima and T. Kiyomatsu, "Arbitrarily-Sized Module Location Technique in the LOP System," in Proc. of 19th Design Automation Conf., June 1982, pp. 718-726.

[6] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, 1955.

[7] M. Hanan, P. K. Wolf and B. J. Agule, "Some Experimental Results on Placement Techniques," in Proc. of 13th Design Automation Conf., June 1973, pp. 214-224.

[8] S. Goto, "An Efficient Algorithm for the Two-Dimensional Placement Problem in Circuit Layout," *IEEE Trans. on Circuits & Syst., CAS-28*, No. 1, Jan. 1981, pp. 12-18.

著者紹介



張銘洙(準會員)

1968年 7月生. 1991年 서강대학교 전자공학과 졸업. 1993년 서강대학교 전자공학과 공학석사 학위 취득. 1993년 2월 ~ 현재 삼성전자 반도체부문 연구원. 주관심 분야는 CAD 시스템, Layout

Synthesis 및 Timing Driven Layout 등임.



李樟淳(準會員)

1965년. 1月生. 1991년 서강대학교 전자공학과 졸업. 1993년 서강대학교 전자공학과 공학석사 학위 취득. 1993년 11월 ~ 현재 한국통신 전임 연구원. 주관심 분야는 CAD 시스템, Computer Architecture

및 광대역 통신망 구조 등임.

黃善泳(正會員)

1976년 2월 서울대학교 전자공학과 졸업. 1978년 2월 한국 과학원 전기 및 전자공학과 석사 취득. 1986년 10월 미국 Stanford 대학 공학 박사 학위 취득. 1976~1981년 삼성반도체 주식회사 연구원. 1986년 ~ 1989년 Stanford 대학 Center for Intergrated Systems 연구소 연구원. Fairchild Semiconductor Palo Alto Research Center 기술 자문. 1989년 3월 ~ 현재 서강대학교 전자공학과 부교수. 주관심분야는 CAD 시스템, Computer Architecture 및 Systems Design, VLSI 설계 등임.