論文 89-26-5-16

새로운 게이트 어레이 배치 알고리듬

(A New Placement Algorithm for Gate Array)

姜 秉 益*, 鄭 正 和*

(Byung Ik Kang and Jong Wha Chong)

要 約

본 논문에서는 게이트 어레이 방식의 레이아웃 설계를 위한 새로운 배치 알고리듬을 제안한다.

제안된 배치 알고리듬은 서로 크기가 다른 마크로셀을 처리할 수 있으며, I/O pad의 위치를 고려함으로써 칩의 내부 영역과 I/O pad간의 배선을 효율적으로 자동화한다.

알고리듬은 초기 분할, 초기 배치 및 배치 개선의 3단계로 구성된다. 초기 분할 단계에서는 각 I/O pad의 위치를 고려하여 clustering에 의해 전체 회로를 5그룹으로 분할한다. 초기 배치 단계에서는 I/O pad 및 주변 그룹과의 연결도를 고려한 clustering/min-cut 분할에 의해 각 셀의 위치를 할당한다. 또한, 배치 개선에서는 확률적 배선 밀도 함수를 도입하여 칩내의 배선 밀도를 균일화하기 위한 셀 이동 알고리듬을 제안한다.

Abstract

In this paper, a new placement algorithm for gate array layout design is proposed.

The proposed algorithm can treat the variable-sized macrocells and by considering the I/O pad locations, the routing between I/O pads and the internal region of a chip can be automated effectively.

The algorithm is composed of 3 parts, which are initial partitioning, initial placement and placement improvement. In the initial placement phase, a given circuit is partitioned into 5 sub-circuits, by clustering method with I/O pad seeds. In the initial placement phase, a combined clustering/min-cut algorithm which considers connectivities of cells not only with I/O pads but also with related partitioned groups is used repeatedly to assign a unique position to each cell. In the placement improvement phase, the concept of probabilistic wiring density is introduced, and cell moving algorithm is proposed to make the density in a chip even.

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)

接受日字 : 1988年 10月 15日

I. 서 론

최근 집적회로의 급속한 발전으로 인하여 시스템의 life-cycle이 짧아짐에 따라 회로 설계 시간을 단축하기 위하여 반 주문형(semi-custom) 설계 방식이

널리 사용되고 있다.

반 주문형 설계 방식중의 하나인 게이트 어레이는 확산 패턴까지 완료된 칩에 배선 패턴 즉 메탈 층에 의한 배선을 행함으로써 설계하고자 하는 회로를 구성하는 방법이다. 이 방법은 인간에 의해 설계된 회로(custom IC)에 비해 면적이 크고 전기적 특성(지연시간 및 power 소비)이 다소 나쁘다는 단점은 있지만 설계 시간을 단축할 수 있고 설계의 신뢰성이 높다는 장점으로 인해 소량 다품종의 LSI/VLSI 회로 설계에 많이 이용되고 있다.[1]

게이트 어레이 방식의 레이아웃은 설계의 복잡성으로 인하여 배치 설계와 배선 설계의 2단계로 나누어 행해지는데, 일반적으로 배치 설계는 다시 초기 배치와 배치 개선으로, 배선 설계는 global 배선과 detailed 배선으로 나뉘어 진다. 현재 많이 사용되고 있는 배치 설계 방법으로는 cluster 성장법[2,3], force directed relaxation[4,5], min-cut분할[6~8] 등이 있다. 그러나, 배치 문제는 NP 완전 문제로 판명되어[9] 최적해를 구하는 것이 불가능하므로 heuristic 원리에 의해 그 해를 구하는 것이 불가피하다.

게이트 어레이 칩은 셀들이 위치하는 배치 영역과 이들간의 연결을 위한 배선 영역, 즉 채널의 트랙 수가 고정되어 있으므로 신호선이 특정 채널에 집중되는 경우에는 채널에서의 트랙 수가 부족하게 되어 신호선이 미결선되는 경우가 발생한다. 따라서, 총 배선장이 다소 증가하더라도 배선 밀도를 균일화 함으로써 100% 배선율을 달성하는 것이 중요하다.

한편, 칩 설계시에는 칩 외부와의 연결을 고려하여 칩의 외부 연결 핀, 즉 I/O pad의 상대적인 위치를 미리 결정할 필요가 있다[10]. 배치 설계시 I/O pad의 위치를 고려하지 않으면 I/O pad와 칩의 내부 영역간의 배선이 매우 복잡해지고, 총 배선장이 증가하며, I/O pad와의 배선을 수작업으로 완성해야 하므로 완전한 레이아웃 자동화가 이루어지지 않는다.

또한, 셀 라이브러리에는 NAND, NOR, NOT 등과 같이 하나의 기본셀 영역만을 사용하는 기본 게이트 외에 플립 플럽 등의 마크로셀(macrocell)도 정의되어 있다. 마크로셀을 사용함으로써, 기본 게이트로만 회로를 실현하는 경우보다 칩 면적과 배선 혼잡도를 감소시키고 회로 성능을 향상시킬 수 있다.

본 논문에서는 I/O pad와 마크로셀을 고려한 배치 알고리듬을 제안한다.

배치 설계 과정에서, 설계 사양으로 미리 정의된 I/O pad의 위치를 고려함으로써 I/O pad와 내부 영역간의 배선을 자동화하고, 2개 이상의 기본 셀 영역을 사용하는 마크로셀의 처리도 가능하게 함으로써 칩의 면적과 총 배선장을 줄이고 설계상의 제약 조건을 해소한다.

제안된 알고리듬은 초기 분할, 초기 배치, 배치 개선의 3단계로 구성된다.

초기 분할 단계에서는 설계 사양을 만족하기 위해 I/O pad와 직접 연결된 셀들을 위치에 따라 seed로 선택한 후, clustering을 이용하여 5그룹으로 분할한다. 초기 배치 단계에서는 분할된 각 그룹에 대하여 clustering과 min-cut 알고리듬에 의한 분할을 반복적으로 적용하여 각 셀의 위치 좌표를 결정한다. 배치 개선 단계에서는 확률적인 배선 밀도 함수를 도입하고 각 셀간의 신호선 연결 요구에 대해 확률적 배선 밀도 행렬을 구한 후, 칩내의 최대 배선 밀도를 최소화하는 min-max 개념[3] 의 셀 이동 알고리듬을 제안한다.

제안된 배치 알고리듬을 IBM/PC-AT 상에서 C언어로 프로그램하고 실제의 레이아웃 예에 적용함으로써 그 효용성을 보인다.

Ⅱ. CMOS 게이트 어레이 칩

CMOS 게이트 어레이 칩은 그림 1과 같이 내부 영역과 외부 영역으로 구성된다. 내부 영역은 기본 셀들이 일렬로 배열되어 있는 셀 row와 신호선 연결을 위한 채널 영역의 반복적인 구조로 이루어지며 외부 영역은 bonding pad와 I/O 버퍼 셀들로 이루어진다.

확산 패턴까지의 공정이 끝나있는 상태의 칩을 베

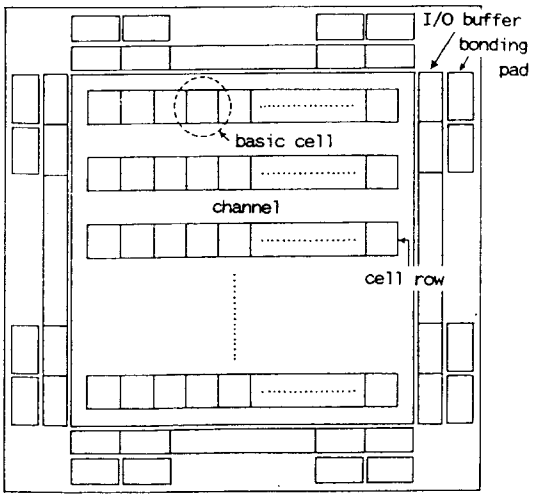


그림 1. CMOS 게이트 어레이 칩의 구조
Fig. 1. The structure of a CMOS gate array chip.

이스 어레이(base array) 또는 매스터 칩(master chip)이라고 한다. 각 기본셀은 논리 게이트를 실현하기 위한 트랜지스터를 구성할 수 있도록 p형과 n형의 확산 영역과 폴리실리콘 라인 등으로 구성되어 있다. 그 위에 형성될 배선 패턴, 즉 메탈층을 결정함으로써 NAND2, NOR3, NOT 등의 기본 게이트와 플립 플롭 등의 마크로셀을 형성할 수 있다.

본 논문에서는 논리를 형성하기 위해 하나의 기본셀 영역만을 사용하는 셀을 기본 게이트, 2개 이상의 기본셀 영역을 사용하는 셀을 마크로셀이라 정의한다. 따라서 본 논문에서 사용하는 셀들은 poly-cell 방식[1] 에서와 같이 높이는 같고 폭은 기본셀의 폭의 정수배가 된다. 셀의 크기는 그 셀이 차지하는 기본셀의 갯수로 정의하며, 한 셀 그룹의 크기는 그 그룹내의 각 셀의 크기의 합으로 정의한다. 또한, bonding pad와 I/O 버퍼 셀을 합쳐서 I/O pad라 한다.

Ⅲ. 배치 설계

본 논문에서 제안하는 배치 설계 과정은 그림 2 와 같이 초기 분할, 초기 배치, 배치 개선의 3단계로 구성된다.

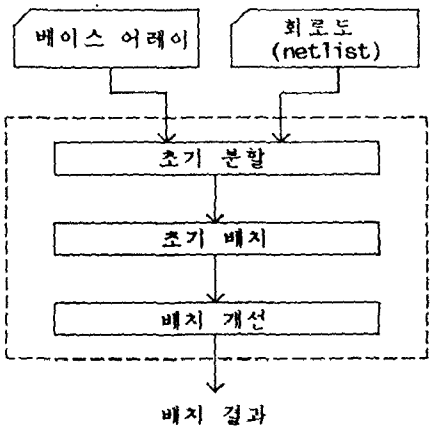


그림 2. 배치 설계 과정의 흐름도
Fig. 2. Flow of placement procedure.

1. I/O pad의 위치를 고려한 초기 분할

반 주문형 IC의 설계 사양에는 칩이 위치하게 될 보드상의 다른 모듈과의 연결을 고려하여 칩의 외부 연결 핀, 즉 I/O pad의 위치를 미리 정의할 필요가 있다. 배치 설계시 I/O pad의 위치에 대한 고려가 없으면 설계 사양에서 주어지는 I/O pad의 위치를 만족하기 위한 칩의 내부 영역과 I/O pad간의 배선이 매우 복잡하게 되고 이에 대한 자동화가 어렵기 때문에, 보통 내부 영역에 대한 레이아웃이 끝난 후 이들의 배선은 설계자에 의해 수작업으로 완성해야 하므로 완전한 레이아웃 자동화가 이루어지기 어렵다.

총 배선장을 줄이고 I/O pad와 내부 셀간의 배선을 용이하게 자동화하기 위해서는 I/O pad와 연결도가 많은 셀들을 그 I/O pad 부근에 배치하는 것이 효율적이다. 이를 위하여 본 논문에서 제안하는 배치 알고리듬의 첫 단계에서는 사용자의 입력에 의해 정의된 I/O pad의 위치 정보와 연결도를 고려하여 전체 회로를 5개의 부회로로 분할한다. 초기 분할된 5개의 부회로들을 방향에 따라 각각 N(north), S(south), E(east), W(west), C(center) 그룹이라고 한다.

그림 3(a)에서 동, 서 남, 북 방향의 I/O pad와 직접 연결된 셀들을 대응되는 각 그룹의 seed로 선정하고 seed와 연결도가 높은 셀들을 선택하여 clustering함으로써 전체 회로를 5개의 그룹으로 분할한다. 즉, 북쪽에 위치한 I/O pad와 연결도가 큰 셀들이 북쪽의 그룹(N그룹)에 포함되게 한다. 중앙의 그룹은 동, 서, 남, 북 4방향의 부회로에 포함되지 않는 셀들로 구성한다. 분할된 각 그룹은 그림 3(b)와 같이 트리로 표현되며 트리내의 각 노드는 해당 그룹에 대한 정보를 포함하는 데이타 구조의 포인터가 된다.

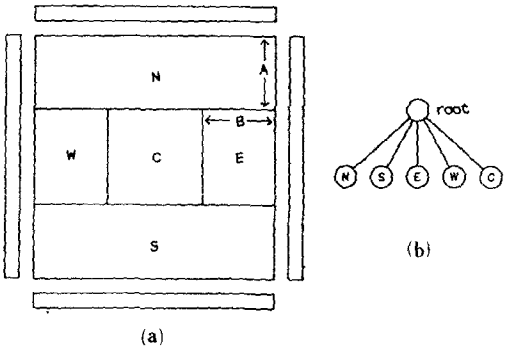


그림 3. (a) Clustering에 의한 초기 분할
(b) 초기 분할된 각 그룹의 트리 표현
Fig. 3. (a) Initial partitioning by clustering.
(b) Tree representation for initial-partitioned groups.

각 그룹에 대응되는 cluster의 크기는 게이트 어레이 칩의 dimension을 고려하여 결정된다.

그림 3(a)에서 각 그룹의 크기를 결정하기 위해 factor A와 B를 정의한다. 각 셀들은 최종적으로 미리 공정이 끝난 각 셀 row상에 위치하게 되므로 factor A는 분할된 각 그룹이 정수개의 셀 row수를 포함할 수 있게 결정된다. 실험적으로 A의 값은 0.25~0.4 사이의 범위에서 자동적으로 계산되게 했다. B의 값은 동, 서, 중앙의 cluster의 크기를 결정해 주며 동, 서 방향에 위치한 I/O pad의 수와 이들의 연결도에 의해 결정된다. 즉 동, 서 방향의 I/O pad수가 많거나 연결도가 상대적으로 높으면 B값은 상대적으로 크게 결정된다. 각 그룹의 크기는 이들의 값과 전체 회로의 크기에 따라 결정된다.

전체 회로의 크기를 SIZE, 그룹 N, S, E, W, C의 크기를 각각 N_SIZE, S_SIZE, E_SIZE, W_SIZE, C_SIZE라 하면 각각은 다음과 같이 결정된다.

$$N\_SIZE = S\_SIZE = A * SIZE$$
$$E\_SIZE = W\_SIZE = (1-2A) * B * SIZE$$
$$C\_SIZE = SIZE - (N\_SIZE + S\_SIZE + E\_SIZE + W\_SIZE)$$

위의 초기 분할 과정은 그림 4와 같다. 각 방향의 그룹에 대해서 그룹의 크기를 계산하고 I/O pad와 직접 연결된 셀들을 seed로 선택한다. 그룹의 크기 S와 seed의 집합 SEED를 이용하여 clustering( ) 과정을 수행하여 N, S, E, W 그룹을 결정하고 이 4그룹에 포함되지 않은 셀들을 C그룹에 포함시킨다.

Clustering 과정은 그림 5와 같다. Clustering하기 위한 셀을 선택하기 위해 모든 셀의 이득함수를 정의한다. 셀 c와 연결된 셀중에서 CLUST 내에 포함된 셀의 갯수를 I(c), CLUST에 포함되지 않은 셀의 갯수를 E(c)라 할 때, 셀 c의 이득함수 g(c)=I(c)−E(c)로 정의한다.

초기에 CLUST에 포함된 셀은 seed로 선택된 셀들이 된다. Clustering은 CLUST에 포함되지 않은 셀들 중에서 이득함수의 값이 가장 큰 셀을 선택하여 주어진 그룹의 크기 S에 도달할 때까지 CLUST에 포함시키는 과정을 반복한다. 하나의 셀이 CLUST에 포함되면 gain_update에 의해 이 셀과 연결된 셀들의 이득함수 값을 조정한다.

2. Clustering/min-cut에 의한 초기 배치

초기 분할이 종료되어 각 그룹이 결정되면 초기 배치 과정에서는 각 그룹에 대하여 수직과 수평방향의 분할을 반복적으로 적용하여 각 셀의 위치를 결정한다.

```
initial_partition()
{
  set factor A and B;
  for(each group of N, S, E, W) {
    set group size S;
    SEED← {cells which are directly connected to I/O pads}
    clustering(group, S, SEED);
  }
  group C←cells not included in N, S, E, or W;
}
```

그림 4. 초기 분할 과정
Fig. 4. Initial partitioning procedure.

```
clustering(group, S, CLUST)
{
  maxgv=large_negative_value;
  gain_update(each element in CLUST);
  do {
    for(each element i in CLUST)
      for(each net n connected to cell i)
        for(each cell c connected to net n)
          if(c not SELECTED && g(c)>maxgv) {
            maxgv=g(c);
            maxgc=c;
          }
    CLUST←CLUST U {maxgc};
    gain_update(maxgc);
    maxgv=large_negative_value;
    size of CLUST+=size of maxgc;
  } while(size of CLUST<S)
}
```

그림 5. Clustering 알고리듬
Fig. 5. Clustering algorithm.

하나의 그룹은 하나의 slice line에 의해 2개의 서브 그룹으로 분할된다. 그룹내의 셀들 중에서 I/O pad나 인접한 그룹내에 있는 셀과 직접 연결된 셀들을 분할될 2그룹의 seed로 선택하고, seed로 선택한 셀들로부터 연결도에 의한 clustering에 의해 2서브 그룹이 결정된다. 다음에는 이 slice line을 cut-line으로 하여 min-cut 알고리듬[6,7]에 의해 cut-line을 지나는 신호선의 수가 최소가 되도록 셀들을 이동시켜 최종 분할 결과를 구한다.

1) 분할 과정의 데이타 구조

분할이 진행되는 과정의 데이타 구조는 하나의 그룹을 노드로 하고 분할된 2서브 그룹을 그 노드의 children로 하는 2진 트리(binary tree)의 형태로 나타난다. 하나의 노드는 그 노드를 분할하는 slice line의 방향, parent 노드 및 children 노드의 포인터,

그룹내에 포함되어 있는 셀, 그룹의 크기, 칩 내에서 그룹을 나타내는 사각형의 위치 좌표 등의 정보를 갖는다.

2) 분할 순서

분할 순서는 인접한 셀 그룹과의 연결도를 고려하기 위해 트리내에서 같은 깊이(depth)를 유지하면서 BFS (breadth first search)[11]에 의해 결정된다. N, S, E, W, C의 각 그룹이 동시에 처리될 수 있도록 노드 포인터들을 갖고 있는 queue 구조[11] 를 만들고 이를 이용하여 각 그룹의 위치가 결정되도록 분할을 진행한다.

3) Slice line의 방향과 위치

분할이 진행될 때, 분할의 기준이 되는 slice line의 방향과 위치는 베이스 어레이의 dimension에 의해 결정된다. 하나의 노드가 수직 방향의 slice line에 의해 분할되면 그의 child 노드들은 수평 방향의 slice line에 의해 분할된다. 수직 방향의 분할은 노드가 하나의 셀을 포함할 때까지, 수평 방향의 분할은 노드가 하나의 셀 row를 포함할 때까지 진행한다. Slice line의 위치는 가능한한 분할되는 서브 그룹의 크기가 같도록 설정되는데, 수평 방향의 분할인 경우에는 분할한 결과 2서브 그룹이 정수개의 셀 row를 포함하도록 결정된다. 예를 들어, 3개의 셀 row를 포함하는 그룹을 수평 방향으로 분할할 때는 하나의 서브 그룹은 2개, 다른 하나의 그룹은 1개의 셀 row를 포함하도록 slice line의 위치를 결정한다.

4) Clustering/min-cut 알고리듬

하나의 그룹을 분할하여 2개의 서브 그룹을 형성하는 clustering/min-cut 과정은 그림 6 의 CL_MC() 와 같다.

CL_MC()는 clustering을 위해 I/O pad와 직접 연결된 셀들을 연결된 I/O pad에 가까운 서브 그룹의 seed(iopad seed)로 선택하고, 현재 분할 과정에 영향을 주는 그룹(related_g)에 포함되어 있는 셀과 직접 연결된 셀을 역시 seed(group seed)로 하여 2서브 그룹의 크기에 이를 때까지 clustering 한다. 분할 과정에 영향을 주는 그룹, 즉related_g는 트리에서 현재 고려중인 노드의ancestor노드들을 탐색하여 현재의 분할 방향과 같은 방향으로 분할된 노드가 나타내는 그룹이다. VH는 그 노드의 분할 방향이 수직, 또는 수평임을 나타내는 플래그이다. Clustering은 그림 5에서와 같은 방법을 사용한다.

Clustering에 의해 하나의 그룹이 2개의 서브 그룹 A, B로 분할되면 그림 7 의 min-cut 알고리듬을 적용

```
CL_MC(nodep)
NODE_t *nodep;
{
  NODE_t * ptr;
  /* iopad and related_group seed selection */
  select_iopad_seed() ;
  for(ptr=nodep→parent; nodep→VH ! =ptr→VH; ptr=ptr→parent)
        ;
  related_g=ptr;
  select_group_seed(related_g, nodep);
  clustering(nodep);
  mincut(nodep);
}
```

그림 6. Clustering/min-cut 알고리듬
Fig. 6. Clustering/min-cut algorithm.

```
min-cut()
{
  do {
    for(i=1; i<MIN(cells in A, cells in B); i++) {
      compute gain values of all a∈A, b∈B;
      find ai, bi that result maximum Hi = H(ai, bi)
      =h(ai)+h(bi)-d(ai, bi);
      save ai, bi and mark them LOCKED;
    }
    find k that maximize Hmax=Σ_{i=1}^{k} Hi
    if (Hmax>0)
      exchange a1, ···, ak with b1, ···, bk;
  } while (Hmax>0)
}
```

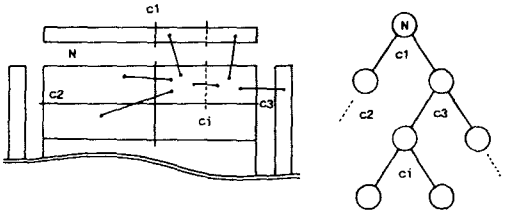
그림 7. Min-cut 알고리듬
Fig. 7. Min-cut algorithm.

하여 2그룹 사이를 지나는 신호선의 수가 최소가 되도록 분할 결과를 개선한다.

Min-cut 알고리듬은 널리 알려져 있는 Kernighan-Lin의 알고리듬[6] 을 이용하되, 서로 크기가 다른 셀들이 존재하므로 가능한한 2그룹의 크기가 균형을 이룰 수 있도록 2셀의 크기의 차이를 나타내는 d(c1, c2)의 항을 이득함수에 추가한다. 셀 a∈A의 이득값 h(a)는 a의 그룹 A, B와의 연결도를 나타내며 a와 연결된 신호선중 B내의 셀과 연결된 신호선 갯수에서 A내부의 셀과 연결된 신호선의 수를 뺀 값이다. h(a)의 값이 클 수록 셀 a가 그룹 A보다 그룹 B에 있어야 함을 나타낸다. 따라서, 이득함수 H(a, b)의 값은 a∈A, b∈B의 이득 값을 각각 h(a), h(b)라 할 때,

$$H(a, b) = h(a) + h(b) - d(a, b)$$

와 같다.



(a) (b)

그림 8. (a) 수직 방향의 분할 예
(b) 분할의 트리 표현

Fig. 8. (a) An example of horizontal partition.
(b) Tree representation of partition.

그림 8(a)는 N그룹내의 하나의 노드를 slice line에 의해 수직 방향으로 분할할 경우의 예이며 그림 8(b)는 이 경우의 트리 표현이다.

분할 과정은 트리내의 모든 leaf 노드들이 하나의 셀만을 포함할 때까지 반복된다. 각 셀의 위치는 그 셀을 포함하는 노드가 갖는 dimension의 중심점으로 계산된 후 실제 위치를 칩의 기본 셀에 할당하는 셀 할당 단계를 거쳐 그 위치가 결정된다.

5) 셀 할당

Clustering/min-cut 분할이 반복적으로 수행되어 모든 leaf 노드가 하나의 셀만을 포함하게 되면 각 셀을 칩상의 기본셀 영역에 위치하기 위해 셀 할당을 수행한다.

트리내의 모든 노드는 사각형의 dimension을 갖고 있으므로, 모든 셀들은 자신이 포함되어 있는 사각형의 중심점에 초기 위치를 갖는다. 초기 위치가 계산되면 기본셀을 기준으로 하여 칩의 dimension과 셀의 크기를 고려하여 각 셀이 위치할 기본셀의 위치로써 셀의 위치를 할당한다. 이 위치를 기본셀 단위로 normalized된 위치라고 한다. 셀에 할당된 위치는 중첩될 수 있으므로 중첩을 검색한 후, 이를 제거하여 최종 위치 좌표를 계산한다. 셀 할당 과정은 그림 9와 같다.

```
cell_assign()
{
  for(each cell in leaf node of tree) {
    assign center point as its initial position;
    assign normalized cell position;
  }
  overlap_check;
  remove_overlap;
}
```

그림 9. 셀 할당 과정
Fig. 9. Cell assign procedure.

3. 배치 개선

초기 배치가 종료되어 각 셀들의 위치가 결정되면 배치 개선을 수행한다.

게이트 어레이는 셀 간의 신호선 연결을 위한 각 채널내의 배선 트랙수가 결정되어 있으므로 채널내의 한 부분에서라도 배선 밀도가 배선 트랙수를 초과하면 100% 배선이 불가능하게 된다. 따라서 배치 개선시에는 칩내의 각 부분의 배선 밀도가 균일하게 되도록 최대 배선 밀도를 최소화하는 min-max 알고리듬[11] 을 적용한다.

1) 확률적인 배선 밀도 행렬

배선이 진행되기 전인 배치 단계에서 배선 밀도를 평가하기 위하여 각 신호선 연결 요구에 의한 확률적 배선 밀도 개념을 도입한다.[12]

회로내의 각 신호선에 대해 신호선에 연결된 셀들을 노드로 하는 완전 그래프를 형성하고, 각 셀간의 거리에 해당하는 웨이트를 각 에지(edge)에 부여한다. (그림10)

두 셀 C1과 C2의 좌표를 각각 (x1, y1), (x2, y2)라 할때, 이 2노드를 연결하는 에지의 웨이트 w12는,

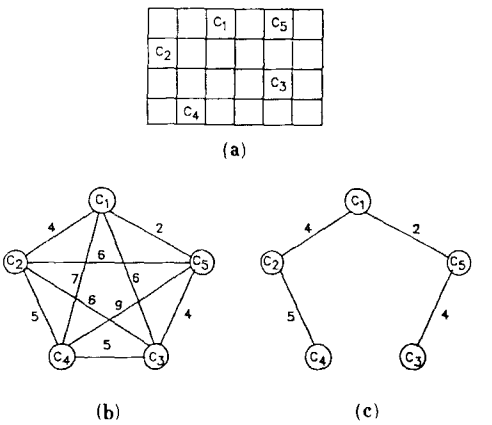


그림10. 다점간 신호선 연결 요구를 2점간 연결 요구로 분할
(a) 다점간 신호선 연결 요구
(b) (a)의 연결 요구에 대한 완전 그래프
(c) 신호선 그래프

Fig. 10. Division of a multi-point net into 2-point nets.
(a) A multi-point connection requirement.
(b) A complete graph for the connection requirement of (a).
(c) A signal graph.

$$w12 = Cx * |x1 - x2| + Cy * |y1 - y2|$$

와 같다. 이 때 Cx, Cy는 x축 및 y축 방향의 웨이트를 다르게 주기 위한 상수이다. Cx=1, Cy=2일 때 그림10(a)의 연결 요구에 대한 완전 그래프의 웨이트는 그림10(b)와 같다.

하나의 신호선에 의해 형성된 완전 그래프의 각 에지에 웨이트가 할당되면 minimum spanning tree 알고리듬[11] 을 적용하여 다점간 신호선을 2점간 신호선으로 분할한다. 모든 2점간 신호선에 대해 신호선에 연결된 2셀의 상대적인 위치에 따라 다음과 같이 확률적 배선 밀도를 구한다.

1) 한 셀 row상의 배선 요구

셀 row의 상하 채널을 사용하여 배선이 가능하므로 상하 채널에 각각 1/2의 확률적 배선 밀도가 발생한다.(그림11(a))

2) 인접한 2셀 row상의 배선 요구

2셀 row사이의 채널을 사용하여 배선하므로 이 채널에 1의 확률적 배선 밀도가 발생한다.(그림11(b))

3) 인접하지 않은 2셀 row상의 배선 요구

2셀 row사이의 각 채널에 대하여 같은 정도의 배선 확률이 존재하므로 이들 각 채널에 대해 1/(2셀 row사이의 채널 수)의 확률적 배선 밀도가 발생한다.(그림11(c))

모든 2점간 신호선에 대해 위의 방법으로 칩내의 각 좌표의 확률적 배선 밀도를 계산하여, (한 셀 row상의 기본셀 수) * (채널 수)의 밀도 행렬(density matrix)을 구성한다.

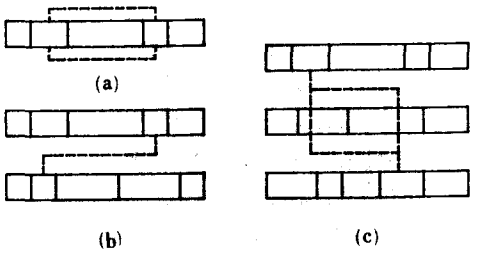


그림11. 확률적 배선 밀도의 계산
(a) 한 셀 row상의 배선
(b) 인접한 2셀 row상의 배선
(c) 인접하지 않은 2셀 row상의 배선

Fig. 11. Calculation of probabilistic wiring density.
(a) Wiring of two cells in a cell row.
(b) Wiring of two cells in two adjacent cell rows.
(c) Wiring of two cells in two non-adjacent cell rows.

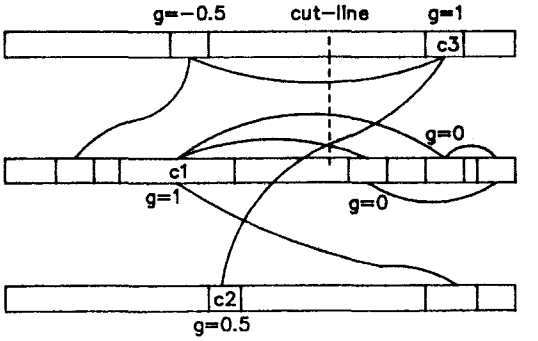


그림12. 배치 개선의 예
Fig. 12. An example of placement improvement.

2) 셀 이동에 의한 배치 개선

배선 밀도 행렬의 각 원소중에서 최대값이 채널당 주어진 배선 트랙수보다 큰 경우에는 이 부분의 밀도를 줄이기 위해 배치 개선을 수행한다.

최대 배선 밀도가 발생한 부분에 cut-line을 설정하고 cut-line을 통과하는 신호선에 연결된 셀중에서, cut-line의 반대쪽으로 이동하였을 때의 배선 밀도의 감소량을 이득함수로하여 이 값이 0이상인 셀들을 후보셀로 선택한다. 선택된 후보 셀들 중에서 cut-line의 한쪽에 있는 셀들의 집합을 A, 그 반대쪽에 있는 셀들의 집합을 B라고 할 때, 집합 A에서 이득함수의 값이 최대인 셀 Ca와 집합 B에서 이득함수가 최대인 셀 Cb를 선택한다.

Ca와Cb가 같은 셀 row상에 존재하거나, 서로 다른 셀 row상에 있더라도 서로 크기가 같으면 2셀의 위치를 교환한다. 또 2셀의 크기가 서로 다른 경우에도 셀 row내에 사용되지 않은 기본셀 영역이 있어서 2셀이 셀 row상에 위치할 수 있으면 2셀의 위치를 교환한다. 만일 서로 교환이 불가능하면 2셀중 이동할 수 있는 하나의 셀만을 이동한다.

교환이나 이동에 의해 셀의 위치가 변화하면 밀도 행렬에서 이에 따라 변화하는 밀도의 값을 다시 계산하고 위의 과정을 반복한다. 하나의 cut-line에 의해 선택되는 모든 셀에 대해서 이득함수의 값이 0 이상인 셀이 없으면 이 부분의 밀도는 더 이상 감소될 수 없다. 밀도 행렬의 모든 원소의 값이 주어진 배선 트랙수 보다 작을 때까지 이상의 과정을 반복하여 개선된 배치 결과를 구한다.

배치 개선의 예는 그림12와 같다. Cut-line을 지나는 신호선에 연결된 셀들 중에서 이득함수의 값이 0 이상이며 cut-line 왼쪽에 있는 후보 셀의 집합은 A

={C1, C2}이고, 오른쪽의 후보 셀의 집합은 B={C3}이다. 집합 A에서 최대 이득함수의 값을 갖는 C1과 집합 B의 C3는 서로 다른 셀 row에 있고 크기가 다르므로, C1과 C3는 C3가 있는 셀 row에 최소한 C1과 C3의 크기 차 만큼의 빈 기본셀 영역이 있을 때는 교환되고(이득=1+0.5=1.5), 그렇지 않으면 C3가 이동하여 cut-line의 왼쪽의 빈 자리에 위치하게 된다. (이득=0.5).

배치 개선 과정을 그림13의 place_improve( )에 나타냈다. S(C)는 셀 C의 크기이고, ROW(C)는 셀 C가 위치하는 셀 row를 나타낸다.

Ⅳ. 실험 및 고찰

본 논문에서 제안한 알고리듬을 IBM/PC AT상에서 C언어로 프로그램하고 min-cut법을 사용한 기존의 시스템[13]과 비교하였다. 본 논문의 알고리듬은 참고문헌[13]에서 사용한 배치 알고리듬에 비해 I/O pad의 위치를 고려하였고, 마크로셀을 처리할 수 있는 장점이 있다.

표 1에서 MC는 min-cut만을 사용한 참고문헌[13]의 결과이고, CL/MC는 본 논문에서 제안한 clustering/min-cut에 의한 결과이다.

표 1에서 예제 회로 A와 B는 마크로셀이 포함되지 않은 회로이고 예제 회로 C, D, E는 크기가 2-4인 마크로셀을 포함하고 있는 회로이다. 최대 배

```
place_improve( )
{
  make minimum spanning tree;
  make density matrix;
  while ((max_density=find_cutline( ))>given_track_capacity) {
    switch (select_cell_pair (cutline) ) {
    case MOVE:
      move a selected cell to the other side of cutline;
      break;
    case EXCHANGE:
      exchange the positions of two selected cells;
      break;
    default:
      exit;
    }
    modify density matrix;
  }
}
select_cell_pair (cutline)
{
  make candidate cell lists of A and B for the cutline;
  calculate gain values of candidate cells;
  select Ca and Cb with maximum gain values in A and B;
  if (gain of both Ca and Cb>0) {
    if (S (Ca) == S (Cb) | | ROW (Ca) == ROW (Cb) )
      return (EXCHANGE);
    else if (ROW (Ca) and ROW (Cb) have room for Cb and Ca, respectively)
      return (EXCHANGE);
    else
      return (MOVE);
  } else if (gain of 1 of 2 cells>0 && gain of the other<0)
    return (MOVE);
  else
    return (NULL);
}
```

그림13. 배치 개선 과정
Fig. 13. Placement improvement procedure.

표 1. 실험 결과
Table 1. Experimental results.

예제 회로	배치 방법	셀수	신호선 수	최대 배선밀도	총 배선장	수행시간 (초)
A	MC	91	85	10	4327	7.1
	CL/MC	91	100	9	4061	8.3
B	MC	141	133	12	6030	11.3
	CL/MC	141	159	10	5972	15.4
C	MC	190	179	15	11099	15.0
	CL/MC	131	140	12	8090	17.2
D	MC	–	–	–	–	–
	CL/MC	112	124	11	7096	12.5
E	MC	–	–	–	–	–
	CL/MC	261	267	20	37465	56.6

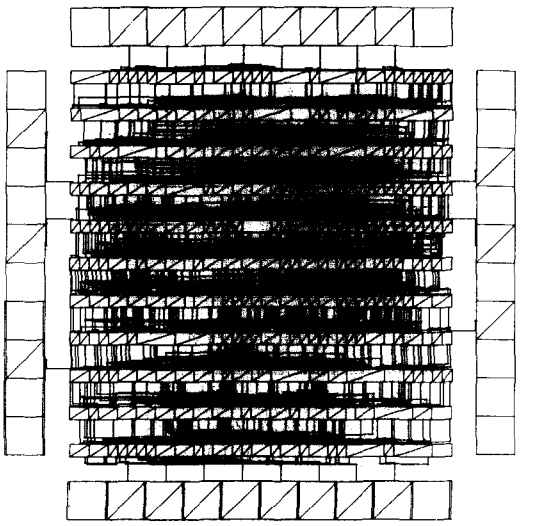


그림14. 예제 회로 E의 레이아웃 결과
Fig. 14. Layout of example circuit E.

선 밀도는 칩내에서 신호선의 혼잡도가 가장 높은 부분에서의 사용 트랙수로 나타냈으며 수행 시간은 IBM/PC-AT상에서 배치 결과를 얻기 위해 소요된 시간이다. 각 예제 회로의 신호선 수는 MC의 경우 I/O 신호선을 처리하지 못하므로 제외하였기 때문에 CL/MC의 경우보다 작다. 총 배선장 중에서 본 알고리듬에 의한 실험결과는 I/O pad로의 배선까지를 포함한 배선장이다.

예제 회로 A와 B에 대한 실험 결과, 본 알고리듬을 사용했을 때 총 배선장과 최대 사용 트랙수가 감소함을 알 수 있었다.

예제 회로 C의 경우에, MC는 마크로셀을 사용할 수 없으므로 마크로셀을 기본 게이트로 변환하여 실험하였다. 따라서 MC와 CL/MC의 셀수와 신호선 수가 서로 다르며 회로를 실현하기 위해 사용한 베이스 어레이도 MC(9×23)와 CL/MC(8×20)가 서로 다른 크기의 베이스 어레이를 사용했다. 예제 회로 D와 E의 경우는 MC로 실현할 수 없었다. 각 경우 모두 100% 배선이 가능했다.

그림14에 예제 회로 E의 배치 결과에 대해 배선을 수행하여 얻은 레이아웃 결과를 보였다.

V. 결 론

본 논문에서는 게이트 어레이 방식의 레이아웃 설계중 새로운 배치 알고리듬을 제안했다.

제안된 배치 알고리듬은 I/O pad의 위치를 고려하여 I/O pad와 셀의 내부 영역과의 배선을 쉽게 자동화할 수 있었으며, 기본 게이트 뿐 아니라 셀의 폭이 서로 다른 마크로셀도 처리할 수 있다.

제안된 알고리듬은 게이트 어레이의 특징인 주어진 채널 용량내에서 100%배선율을 달성하기 위해서 배선장의 최소화와 배선 밀도의 균일화를 목적함수로 하여 초기 분할, 초기 배치, 배치 개선의 3단계로 구성되었다. 초기 분할 단계에서는 I/O pad의 위치에 따라 선택된 seed로 부터 칩의 dimension을 고려하여 전체 회로를 5그룹으로 분할한다. 초기 배치 단계에서는 I/O pad 및 미리 분할된 다른 그룹과의 연결도를 고려하여 seed를 선택하고 clustering에 의해 하나의 그룹을 2그룹으로 분할한 후, 셀의 크기를 고려한 min-cut 알고리듬에 의해 분할 결과를 개선한다. 배치 개선에서는 초기 배치된 결과로 부터 확률적인 배선 밀도 행렬을 구하고 최대 밀도를 최소화하기 위해 셀 이동 알고리듬을 제안하였다.

參 考 文 獻

[1] W.F. DeCamp, R.A. Bechade and M.P. Concannon, "Gate array and standard cell approach," in Chap. 2, in **Design Methodologies**, S. Goto Ed. North-Holland, 1986.

[2] C.A. Palesko and L.A. Akers, "Logic partitioning *for minimizing gate arrays*," *IEEE Trans. on CAD*, vol. CAD-2, no. 2, Apr. 1983.

[3] M.A. Breuer, Design Automation of Digital Systems: Theory and techniques, Chap. 4-5, Prentice Hall, 1972.

[4] S. Goto, "An efficient algorithm for the two-dimensional placement in electrical circuit layout," *IEEE Trans, on CAS*, vol., CAS-28, no. 1, Jan. 1981.

[5] J.H. Jung, S. Goto, H. Hirayama, "A new approach to the two dimensional placement with wire-congestion in master slice layout design," 일본전자통신학회 논문집 vol.J. 64-A, no. 1, 1981.

[6] B.W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell system technical jounal, Feb. 1970.

[7] M.A. Breuer, "A class of min-cut placement algorithm," *Proc. 17th Design Automation Conf.*, pp. 284-290, 1977.

[8] H. Shiraishi and F. Hirose, "Efficient placement and routing techniques for master slice LSI", *Proc. 14th Design Automation Conf.*, pp. 458-464, 1980.

[9] M.T. Shing, T.C. Hu, "Computational complexity of layout problems," in Chap. 8, *in Layout Design and Verification*, T. Ohtsuki Ed., North-Holland, 1986.

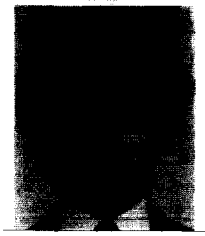
[10] LSI LOGIC, Databook and Design Manual, Oct. 1986.

[11] A. Aho, J.E. Hopcroft, and J.D. Ullman, Data Structures and Algorithms, Addison Wesley, 1983.

[12] 배영환, "VLSI Layout 설계 자동화를 위한 Gate Array Global Router에 관한 연구", 한양대학교 석사학위논문, 12. 1986.

[13] 이건배, 정정화, "게이트 어레이의 자동 배치, 배선 시스템", 대한 전자공학회 논문집, 제25권, 제5호, pp. 100-107, 5. 1988. *

著 者 紹 介



姜 秉 益(正會員)

1961年生. 1984年 한양대학교 전자공학과 졸업. 1986年 한양대학교 대학원 전자공학과 졸업, 공학석사학위 취득. 1986年 3月～현재 한양대학교 대학원 전자공학과 박사과정 재학중. 주관심 분야는 VLSI CAD, Layout 및 Simulation 등임.

鄭 正 和 (正會員) 第26卷 第1號 參照

현재 한양대학교 전자공학과 부교수