論文93-30A-3-14

클러스터링을 이용한 계층적 분할 방법

(A Hierarchical Partitioning Method Using Clustering)

金 衷 希*, 申 鉉 哲**

(Chung Hee Kim and Hyun Chul Shin)

要 約

회로의 분할은 집적회로의 계층적 설계를 위한 중요한 과정이다. 본 연구에서는 규모가 큰 회로의 효율적인 분할을 위한 새로운 알고리듬을 제안한다. 새로운 분할 알고리듬은 cluster를 이용한 2단계의 계층적 방법이다. 상위 단계에서는 주어진 threshold 값 이상으로 서로 밀접하게 연결된 셀들을 cluster로 형성하여 이들을 분할함으로써 복잡도를 줄인다. 반복적인 개선을 이용한 분할 결과가 초기 분할에 의존적이라는 단점을 극복하고 일관성 있게 좋은 결과를 얻기 위하여, cluster 형성의 threshold 값과 cluster의 초기 분할을 변화시키며 여러번 분할하여 가장 좋은 cluster 분할을 구한 후, 이를 초기 분할로 이용하여 하위 단계의 분할을 수행한다. 각 단계의 분할은 크기의 제약 조건을 동적으로 조정한 방법[5]을 사용하였다. 벤취마크 예제에 대하여 본 알고리듬을 적용한 결과 우수한 결과를 얻었으며 이는 본 방법이 효율적이고 효과적임을 보여준다.

Abstract

Partitioning is an important step in the hierarchical design of very large scale integrated circuits. In this research, a new effective partitioning algorithm based on 2-level hierarchy is presented. At the beginning, clusters are formed to reduce the problem size. To overcome the weakness of the iterative improvement techniques that the partitioning result is dependent on the initial partitioning and to consistently produce good results, the cluster-level partitioning is performed several times using several sets of parameters. Then the best result of cluster-partitioning is used as the initial solution for lower level partitioning. For each partitioning, the gradual constraint enforcing partitioning method [5] has been used. The clustering-based partitioning algorithm has been applied to several benchmark examples and produced promising results which show that this algorithm is efficient and effective.

I. 서 론

집적 회로의 집적도의 증가로 계층적 설계 방법이 설계의 기간을 줄이기 위하여 필수적이다. 따라서 계층적 구조를 얻기 위한 분할 방법이 중요한 비중을 차지하게 되었다. 좋은 분할은 설계의 복잡도를 현저히 줄일 수 있고 성능을 개선시킬 수 있다. 회로의

正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
(※이 논문은 1992년도 교육부 학술 연구 조성비에 의하여 연구되었음.)
接受日字: 1992年 9月 28日

분할 문제는 셀과 그 셀들을 연결하는 네트들로 이루어진 회로를 몇개의 부분집합으로 분할하는 것이다. 이때 각 부분집합들은 주어진 크기의 제약 조건을 만족해야 하며, 서로 다른 부분집합에 속한 셀 간의 연결정도를 나타내는 cost를 최소화 해야 한다. 이와같이 크기의 제약 조건을 고려한 분할은 NP-hard에 속하는 문제 [1]로 최적의 해를 구하기 어렵다. 그래서 여러가지 heuristic한 방법들이 사용되어 진다.

Kerninghan and Lin (K&L) [2]은 전체 그래프를 두 부분으로 분할하는 알고리듬을 제안하였다. 이 방법은 임의로 분할한 후 두 그룹에서 서로 교환하였을 때 연결도가 감소되는 부분 집합을 선택하여 교환하는것을 반복하여 연결도를 감소시키는 방법이다. Fiduccia and Mattheyses (F&M) [3]는 K&L의 방법을 개선한 2-way 분할 알고리듬을 제안하였다. 이 방법은 임의의 초기 분할후, 각 셀들을 다른 그룹으로 이동시킬때의 cost의 감소를 각 셀의 gain으로 계산하여, gain이 큰 순서대로 셀을 이동시켜 cost를 줄이는 방법이다. K&L의 방법이 $O(n^2 logn)$의 복잡도를 가지는데 비하여 이 방법은 전체 핀의 수에 비례하는 복잡도를 갖는것이 특징이다. 한편 F&M 방법을 응용하여 한번에 4개의 부분집합으로 분할하는 quadrisection [4] 방법도 제안 되었다.

F&M 방법은 두 그룹으로 분할 후 한 셀이 다른 그룹으로 이동하였을 때 cost가 가장 많이 감소하는 셀을 이동시킨다. 그러나 Level gain을 이용한 방법 [6]은 그 셀의 이동에 따른 다른 셀의 gain의 변화를 예측하여 이동시키는 방법이다. 이 방법을 k-way 분할 방법으로 확장시킨 방법도 Sanchis [7]에 의하여 제안 되었다. 최근에 부분 집합의 크기의 차이를 크게 허용하면서 연결도를 줄이는 ratio-cut 방법 [8]이 제안되었다. 그러나 이 방법은 두 부분집합의 크기의 차이가 작아야 되는 경우에는 사용할 수 없게 된다

본 방법은 cluster를 이용한 계층적 분할 알고리듬이다. 기존의 방법들은 대개 임의의 초기 분할 후, 개선 단계를 통하여 분할 결과를 개선 시킨다. 그러나 이러한 방법은 초기 분할에 의하여 최종 결과가 많은 영향을 받는다. 즉 초기 분할이 좋지 않으면 결과를 개선하여도 좋은 결과를 기대하기 어렵다. 그러므로 본 방법은 초기 분할에 중점을 두어 좋은 초기 분할에서 시작하여 결과를 개선 시켰으며 효율적인 수행을 위하여 2단계의 계층적 분할을 이용하였다. 먼저 서로 밀접하게 연결되어진 셀들을 cluster로 만든다. 이 cluster의 형성 방법에 따라 결과가 영향을 받으므로 변수를 변화시켜서 몇가지의 cluster를 구성하여 이들을 각각 몇가지의 서로 다른 random 초기 분할로부터 분할하여 가장 우수한 결과를 선택한다. 따라서 분할이 여러번 반복되지만 이는 cluster들의 분할이므로 복잡도가 작아서 효율적으로 다양한 해의 공간 (solution space)을 탐색할 수 있다. 가장 우수한 cluster 분할 결과를 셀 단계에서 초기 분할로 이용하여 최적에 가까운 결과를 얻도록 하였다. 이때 cluster의 분할과 셀 단계의 최종 분할은 F&M 방법을 개선한 분할 방법 GCEP [5]를 사용하였다. GCEP는 F&M 방법이 크기의 제약 조건을 미리 결정하여 주고 그 범위내에서 분할하는데 비하여, 크기의 제약 조건을 동적으로 조정하여 결과를 개선한 방법이다. 본 계층적 분할 방법은 실험 결과 F&M [3] 방법에 비하여 평균 60% 이상 우수한 결과를 보여주었고, 임의의 초기 분할로 부터 F&M 방법을 500회 수행하여 이중 가장 좋은 결과를 선택했을 때에 비하여도 17% 우수한 결과를 보여주었다. 또 primal-dual [11] 방법에 비하여 평균 3% 좋은 결과를 보여주었다

2장에서는 cluster 생성 방법에 대하여 기술하고, 3장에서는 새로운 분할 방법에 대하여 기술하며, 전체 알고리듬과 세부 사항을 4장에서 기술한다. 본 알고리듬의 수행 결과와 다른 방법과의 결과 비교를 5장에서 기술하고, 6장에서 결론을 맺는다.

Ⅱ. Cluster 생성 방법

cluster를 만드는 목적은 서로 밀접하게 연결된 셀들을 하나의 cluster로 만들어 처리함으로써 전체 수행 시간을 단축시키고 성능을 개선하는데 있다. 두 셀 또는 cluster 간의 친밀도는 그들에 연결된 모든 네트의 weight의 합, 공통된 네트의 weight의 합, 두 셀의 크기등의 함수로 정의하며, 이러한 친밀도를 이용하여 clustering 여부를 결정할 수 있다. Clustering을 위하여 반복적인 top-down ratio-cut 분할을 이용하는 방법 [11]도 제안 되었으나 이는 여러번의 분할을 수행하므로 복잡하다. 본 논문에서는 간단하면서도 효과적인 bottom-up clustering 방법을 개발하였다. 처음에는 각각의 셀 하나가 한개의 cluster가 된다. 그리고 모든 cluster 쌍에 대하여 cluster 간의 친밀도를 계산하고 친밀도가 가장 큰 cluster 쌍을 통합하여 하나의 cluster를 형성한다. 통합된 cluster에 연관된 친밀도는 통합 후 다시 계산하고, 친밀도가 가장 큰 cluster 쌍을 찾아서 이들의 친밀도가 threshold 값 이상이면 이들을 통합하여 하나의 cluster를 형성하는 과정을 모든 친밀도 값이 threshold 보다 작아질 때까지 반복한다.

두 cluster C와 D의 친밀도는 (1)식과 같이 계산하였다.

$$\text{closeness}(C,D) = \alpha * (\text{num\_cnet}(C,D) / \text{MIN}(\text{num\_net}(C), \text{num\_net}(D))) - \beta * (\text{cl\_size}(C,D) / \text{cell\_large}) \quad (1)$$

여기서 MIN(a,b)는 a와 b중에 작은 값을 나타내며, num_cnet(C,D) 는 C와 D에 공통으로 연결된 네트의 수이고 num_net(C)는 cluster C에서 다른 cluster와 연결된 네트의 수를 나타낸다. cl_size(C,D) 는 C와 D를 결합하여 하나의 새로운 cluster로 만들었을 때의 cluster 크기이다. cell_large는 가장 큰 셀의 크기이다. 실험결과 벤취마크 예제인 표준셀의 분할시 α =200, β =2이 적당하였다.

위 식의 첫번째 항은 C와 D간에 얼마나 서로 밀접하게 연결되어 있는지를 나타내고, 두번째 항은 cluster의 크기를 조절하기 위한 항으로 하나의 cluster의 크기가 너무 커지는 것을 막기 위하여 사용하였다. 첫번째 항에서 (3)식과 같이 공통으로 연결된 네트의 수를 두 cluster의 전체 네트의 수로 나누어 줄 수도 있겠으나, 본 방법은 (2)식과 같이 두 cluster중 네트의 수가 적은 값으로 공통으로 연결된 네트의 수를 나누어 주었다. 여기서 f_1이 f_2 보다 합리적이라는 것은 다음 예에서 알 수 있다.

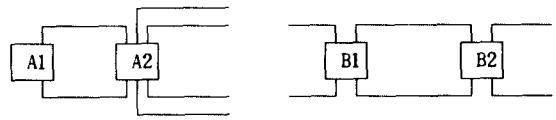
$$f_1(C,D) = \text{num\_cnet}(C,D) / \text{MIN}(\text{num\_net}(C), \text{num\_net}(D)) \quad (2)$$

$$f_2(C,D) = \text{num\_cnet}(C,D) / (\text{num\_net}(C) + \text{num\_net}(D)) \quad (3)$$

예를 들어 그림 1의 두 경우에 대한 f_1과 f_2의 값을 계산하면 다음과 같다.

$f_1(A1,A2) = 2 / 2$
$f_1(B1,B2) = 2 / 4$
$f_2(A1,A2) = 2 / 8$
$f_2(B1,B2) = 2 / 8$

식 (3)에 의한 그림 1의 (a), (b)의 f_2의 값은 서로 같으나 식 (2)에 의한 f_1값은 (a) 경우의 값이 크므로 A1과 A2가 먼저 하나의 cluster로 형성된다. 그림 1에서 (a)가 (b)보다 더 친밀도가 큰것을 알수 있다. 왜냐하면, A1은 통합이 된다면 그 대상이 A2 밖에 없으나, B1과 B2는 아직 분명하지 않기 때문이다. 실제 회로의 예를 분할하여 f_1과 f_2를 비교한 결과도 역시 f_1을 이용하는 것이 f_2또는 f_1+f_2를 이용할 때에 비하여 우수하게 나타났다.



(a) (b)

그림 1. 두 cluster의 친밀도
Fig. 1. Closeness of clusters.

다음은 cluster를 생성하는 알고리듬이며 변수 thres는 clustering을 위한 친밀도의 threshold 값이다. 알고리듬에서 find candidate(c)는 cluster c와 친밀도가 가장 큰 cluster를 찾는 procedure이고 친밀도가 threshold보다 큰 cluster 쌍이 있는한 통합이 계속된다. 여기서 thres 변수를 주는 대신에 원하는 cluster 수 (예를들면, 셀의 수/10 등)를 변수로 주어도 결과는 큰 차가 없다.

```
Algorithm : make_cluster( thres )

/* Each cell is a cluster at the beginning */
for( each cluster C) {
  find_candidate(C);
}
while( EVER ) {
  find the pair of clusters(C1,C2)
       which have the largest closeness;
  if( closeness(C1,C2) < thres )
     break;
  merge_cluster(C1,C2);
  /* Update closeness for the new cluster
       and clusters connected to C1 and C2 */
  for( each changed cluster C' )
     find candidate( C' );
}

find_candidate( C ) {
  /* Find the closest cluster for C */
  ov = -∞;
  for(each cluster D having a connection to C){
    v = closeness(C,D);
    if( v > ov ) {
```

```
            ov = v;
            C.closely_connected = D;
            C.closeness = v;
        }
    }
}
```

Ⅲ. 새로운 분할 방법

분할은 초기 분할과 이 초기 분할을 개선 시키는 반복적인 개선 단계로 나눌 수 있다. 임의의 초기 분할을 이용한 방법들은 초기 분할에 의해 최종 결과가 많은 차이를 보이게 된다. 예를들어 가장 많이 사용되어지는 방법의 하나인 F&M 방법[3]은 수행 시간이 짧고 비교적 좋은 결과를 보여준다. 그러나 여러 예제에서 일관성 있게 좋은 결과를 얻기 위해서는 임의로 초기 분할을 바꾸면서 여러번 수행하여야 한다. 한 실험 결과에 의하면 F&M 방법을 500번 수행하여 얻은 가장 좋은 결과가 다른 방법과 비교할만하게 나타났다.[12] 이러한 초기분할의 영향을 줄이고 다양한 분할 문제에 대하여 일관성 있게 우수한 결과를 얻기 위하여 본 연구에서는 cluster를 만들어 문제의 복잡도를 줄이고, 분할 알고리듬[5]을 이용하여 여러가지 분할과 초기 상태에 대하여 평가하여 좋은 cluster의 분할을 찾도록 하였다. 이때 어느 정도의 cluster를 만드는지에 따라서 결과가 달라지므로 cluster의 threshold 값을 변화시키며 반복적으로 상위 (cluster) 단계에서의 분할을 수행한다. 이와같이 cluster의 분할을 수행한 후 이를 이용하여 셀의 분할을 실시하며 셀의 분할은 한번의 분할로 완료된다. 이 방법은 매우 효율적이다. 왜냐하면 처음부터 셀 단계에서 여러가지 임의의 초기 해로부터 분할을 수행하여 좋은 해를 얻으려면 많은 CPU 시간이 필요할 것이나 cluster를 이용하면 분할의 복잡도가 크게 줄기 때문이다. 셀을 cluster로 만들어 처리하면 셀의 수에 비하여 적은수의 cluster를 분할하게 되고, 네트의 수도 cluster간을 연결하는 네트만 고려하므로 복잡도가 크게 줄어들게 된다. 이때 전체 cluster의 수는 초기의 셀의 수에 비하여 보통 10% 정도 이므로 몇번의 분할을 하여도 시간이 크게 증가하지 않는다.

이때 각 단계의 분할 알고리듬은 F&M 방법[3]을 개선한 방법인 점진적으로 크기 제약 조건을 만족시키는 방법[5]을 사용하였다. 이 방법은 F&M의 linear한 복잡도를 가지는 특성을 이용하면서 단계적으로 크기의 균형을 맞추어 주는 알고리듬 이다. 즉 F&M방법은 크기의 조건에 벗어나는 셀의 이동을 허용하지 않으나 이 알고리듬은 크기의 제약 조건을 변화시켜 동적으로 조정한다. 즉 초기에는 크기의 차이가 크게 벗어나는 것도 허용하며 iteration이 반복됨에 따라 크기 조건에서 벗어나는 폭을 감소 시켜서 마지막 단계에서는 나누어진 두 그룹의 크기의 제약 조건이 모두 만족되도록 하였다. 이렇게 초기 단계에서 셀의 이동을 크게 허용하면 부분적 최적화 (local optimum)에서 벗어나 전체적인 최적화 (global optimum)를 얻을수 있는 가능성이 커져서 최적에 가까운 결과를 얻을 수 있다. 또 K&L의 방법은 두 그룹에서의 교환하는 부분집합의 크기가 같아야 하고 이 부분집합을 교환 하였을 경우 cost의 변화를 계산하여 cost가 줄어드는 경우에만 허용하는데 비하여, 본 논문에서 사용한 알고리듬은 각 단계에서 그 단계에서 허용하는 그룹의 최소 크기 제한에 벗어나지 않는 범위에서 셀을 한 그룹에서 다른 그룹으로 이동시키며, 이때 좀더 최적에 가까운 해를 구하기 위하여 cost가 증가하는 경우도 부분적으로 허용하였다.

Ⅳ. 전체 알고리듬

계층적 구조를 사용한 본 방법은 다른 방법과 마찬가지로 초기 분할후 이를 개선하는 방법이다. 그러나 초기 분할이 잘 이루어 지지 않으면 이를 개선하여도 좋은 결과를 기대할 수 없으므로, 본 방법은 한번의 수행으로 좋은 결과를 얻기 위하여 여러번 서로 다른 cluster를 생성하고 몇번의 cluster 분할을 하여 이중에서 가장 좋은 결과를 셀 단계의 초기 분할로 사용하였다.

본 방법의 전체 알고리듬은 다음과 같다.

```
Algorithm : Overall

input_data;
/* Various cluster-level partitioning */
cut = infinity;
for( i=1; i<=K; i++ ) {
  /* thres [i]  contains the predefined threshold
     value for clustering at i-th iteration */
  make_cluster( thres [i]  );
  for( j=1; j<=M; j++ ) {
      random_initial_partition(j);
      mincut = GCEP();
      if( mincut < cut ) {
      cut = mincut;
```

```
        save result;
      }
    }
  }
  /* The save_result is used as the initial
     partitioning for cell level partitioning */
  flatten_clusters;
  /* Cell-level partitioning */
  GCEP();
  output;
```

위의 알고리듬에서 초기 분할은 주어진 cluster를 임의로 분할하여 사용 하였다. 그리고 cluster를 K번에 걸쳐서 수행하게 되는데 이때 두 cluster의 친밀도가 주어진 thres [i] 이상인 경우 가장 친밀도가 큰 cluster 쌍 부터 하나의 cluster로 만들게 된다. 그리고 각각의 clustering 결과에 대하여 GCEP [5] 알고리듬을 M번 수행하게 된다. 이와같이 K · M번 수행한 분할 결과중 가장 좋은 분할을 다음의 셀 분할 단계의 초기 분할로 이용하고, 셀 단계에서 GCEP를 이용한 분할 단계를 거쳐 최종 결과를 얻게 된다.

본 알고리듬에서는 40≤thres [i] ≤48의 값을 사용하였고 K=M=5를 사용하였다. 그러므로 다섯 단계의 cluster를 하면서 각각 5번의 GCEP 단계를 거쳐 우수한 cluster의 분할을 얻게 된다. 여기서 K와 M의 값을 더 크게하여 더 넓은 범위에서 cluster의 분할을 찾을 수 있다. 그러나 실험결과 시간의 증가에 비하여 결과의 개선이 적음을 알 수 있었다.

Ⅴ. 실험 결과

새로 제안한 방법을 몇개의 2-way 분할 방법과 비교 하였다. 본 알고리듬은 C언어를 사용하였고 UNIX 운영체계에 의하여 SUN4/40 sparc station에서 실행 하였다.

표 1은 실험에 사용한 MCNC 벤취마크 예제를 나타내었다. 5개의 테스트 예제와 2개의 gate array 셀 예제, 그리고 2개의 표준셀 예제이다. 표 2는 본 분할 알고리듬을 2-way 분할로 수행하여 다른 방법들과의 실험 결과를 비교한 것이다. 결과 비교에 사용된 다른 방법은 Simulated annealing [10] 과 F&M 방법 [3], 그리고 primal-dual 방법 [1] 이다. 이들 결과는 SUN4 sparc station에서 실행된 것으로 [12] 에서 인용하였다. Simulated annealing 방법 [10] 은 임의의 서로 다른 초기 분할로부터 10번 수행하여 가장 좋은 해와 해의 평균값을 나타내었고, F&M 방법도 임의의 서로 다른 초기 분할로부터 500번 분할을 수행하여 가장 좋은 해와 해의 평균을 나타내었다.

표 1. 사용 예제
Table 1. Examples.

Example	#IO	#nodes	#nets
Test02	70	1663	1866
Test03	65	1607	1699
Test04	36	1515	1738
Test05	63	2595	2910
Test06	69	1752	1745
PrimGA1	81	752	904
PrimGA2	107	2907	3029
PrimSC1	81	752	904
PrimSC2	107	2907	3029

각각의 수행 시간은 1회 수행한 시간으로 단위는 초이다. 결과 비교를 위하여 사용한 표 2의 실험 결과의 크기 조건은 다음과 같이 평균 크기에서 20% 이상 벗어나지 않도록 하였다.

average subset size * 0.8 <= subset size <= average subset size * 1.2

표 2에서 알수 있듯이 새로운 방법 (NEW)이 큰 회로를 분할하는데 효율적인 것으로 나타났다. 특히 가장 큰 예제인 PrimGA2및 PrimSC2에서 가장 우수한 결과를 나타내었으며, 수행 시간도 비슷한 결과를 보여준 primal-dual 방법보다 상당히 빠르게 나타났다. 실질적으로 CPU 시간의 증가는 셀의 수가 증가함에 따라 선형에 가깝게 증가함을 알 수 있다. 본 방법은 primal-dual 방법에 비하여 평균적으로 약 3% 좋은 결과를 얻었고, F&M 방법을 500번 수행한 결과중 가장 좋은 결과에 비하여 약 17% 좋은 결과를 얻었다. F&M 방법의 평균적인 cut의 수와 NEW의 cut의 수를 비교하면 1518과 600으로, NEW의 결과가 60% 이상 우수함을 알수있다.

Ⅵ. 결 론

집적회로의 효율적인 분할을 위한 새로운 계층적 분할 방법을 제안하였다. 먼저 좋은 cluster를 만들기 위한 새로운 cluster 생성 방법을 제안하였다. 새로운 cluster의 생성은 친밀도를 나타내는 효과적인

표 2. 분할 실험결과
Table 2. Partitioning Results.

(time: sec)

Example	SA (10 runs)			F-M (500 runs)			PD (1 run)		NEW (1 run)	
	mincut	avg.cut	sec/run	mincut	avg.cut	sec/run	mincut	sec/run	mincut	sec/run
Test02	90	109.0	4081	107	126.7	24.4	92	217	81	351
Test03	59	86.4	1284	81	145.8	20.6	58	530	62	255
Test04	53	69.5	2784	44	46.0	10.4	43	431	44	275
Test05	53	107.3	3176	42	44.9	20.1	42	1563	45	791
Test06	74	85.5	4710	74	180.7	55.3	62	1216	53	317
PrimGA1	36	38.9	2424	37	86.5	17.2	39	207	38	41
PrimGA2	131	168.5	1903	146	397.0	193.4	123	1612	119	504
PrimSC1	41	63.8	4490	39	88.6	17.7	39	195	39	41
PrimSC2	128	156.4	4920	157	402.3	191.3	120	2512	119	518
Total	665	885.3		727	1518.5		618		600	

함수를 사용하여 수행하였다. 본 계층적 분할 방법은 cluster를 만들어 회로의 규모를 축소시키고, parameter의 변화를 이용하여 좋은 cluster의 분할을 선택한다. 이 cluster의 분할을 셀의 초기 분할로 하여 분할 알고리듬을 수행하여 일관성 있게 좋은 분할이 이루어지도록 하였다. 기존의 임의의 초기 분할에서 시작하여 결과를 개선시키는 방법에서 초기 분할이 최종의 결과에 많은 영향을 주는 것에 비하여, 본 방법은 cluster를 이용하여 효율적으로 좋은 초기 분할을 구한 후, 셀 단계의 분할을 시작하므로 좋은 결과를 얻을 수 있다. 실험 결과 본 방법이 기존의 여러 방법들에 비하여 비교적 짧은 시간 내에 우수한 결과를 얻는데 효과적인 것으로 나타났다. 특히 F&M 방법의 평균 결과에 비해서는 60% 이상 개선된 결과를 얻을 수 있었다.

參 考 文 獻

[1] M. R. Garey and D. S. Johnson, "Computers and Intractability : A Guide to the Theory of NP-Completeness", Freeman, 1979.

[2] B. M. Kerninghan and S. Lin, "An Efficient Heuristic Procedure for Partitioning graph", Bell system technical Journal, Vol 49, No 2, pp 297-307, Feb. 1970.

[3] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions", *Proc.* 19th Design Automation Conference, pp 175-181, 1982.

[4] P. R. Suaris and G. Kedem, "Quadrisection : A New Approach to Standard Cell Layout", *Int. conf. on Computer Aided Design*, pp 474-477, 1987.

[5] 김 충희, 신 현철, "집적회로의 계층적 배치를 위한 새로운 분할 알고 리듬", 대한 전자 공학회 논문지, 29권 5호, 1992.

[6] B. Krishnamurthy, "An Improved Min-cut Algorithm for Partitioning VLSI Networks", *IEEE Trans. Comput.*, vol. C-33, pp.438-446, May 1984.

[7] L. A. Sanchis, "Multiple-Way Network Partitioning", *IEEE Trans. on Computers*. VOL. 38, NO 1, JAN, 1989.

[8] Y.C. Wei and C.K. Cheng, "Towards Efficient Hierarchical Designs by Ratio Cut Partitioning", *Int. Conf. on Computer Aided Design*, pp 298-301, 1989.

[9] S. Mayrhofer and U. Lauther,

"Congestion-Driven Placement Using a New Multi-Partitioning Heuristic", *Int. Conf. on Computer Aided Design*, pp 332-335, 1990.

[10] S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by Simulated Annealing", Science, May 13, pp 671-680, 1983.

[11] C. W. Yeh and C. K. Cheng, "A General Purpose Multiple way Partitioning Algorithm", *Proc.* 28th Design Automation Conference, pp 421-426, 1991.

[12] C. W. Yei, C. K. Cheng and T. Y. Lin, "An Experimental Evaluation of Partitioning Algorithms", *Int. Conf. of ASIC*, p14-1, 1991.

著 者 紹 介

金 表 希(正會員) 第 29 券 A編 第 5 號 參照
현재 한양대학교 전자공학과 박사과정 재학중.

申 鉉 哲(正會員) 第 29 券 A編 第 5 號 參照
현재 한양대학교 전자공학과 부교수.