論文98-35C-3-2

하이브리드 버켓을 이용한 대규모 집적회로에서의 효율적인 분할 개선 방법

(An Efficient Iterative Improvement Technique for VLSI Circuit Partitioning Using Hybrid Bucket Structures)

任 昌 慶 *, 鄭 正 和 *

(C.K.Eem and J.Chong)

요 약

본 논문에서는 대규모 집적회로 분할을 위한 빠르고 효율적인 분할 개선 방법을 제안하고 이를 구현하기 위한 하이브리드 버켓 구조를 제시한다. 빠른 수행 속도로 인해 대규모 집적회로 분할 방법으로 널리 이용되어 온 반복적 분할 개선 방법은 이동하고자 하는 셀의 선택 방식에 의해 그 알고리듬의 성능이 결정되었다. 이에 따라 분할 개선 방법의 성능 향상을 위해 다양한 셀 선택 방식이 제안되어 왔는데 특히 S. Dutt[13][14]의 클러스터링을 고려한 셀 선택 방식은 분할 개선 방법에 의한 분할 성능의 향상에 크게 기여하였다. 본 논문에서는 기존에 제안된 분할 개선 방법의 획일적인 셀 선택 방식이 내포하는 문제점을 제기하고 이의 보완을 위해 분할 개선 진행 상황에 따라 셀 선택 방식이 가변하는 새로운 분할 개선 방법을 제안한다. 또한 이의 효율적인 구현을 위해 서로 다른 특성을 갖는 하이브리드 버켓 구조를 제시한다. FM[3] 알고리듬과 동일한 time complexity를 갖는 본 분할 개선 방법은 ACM/SIGDA에서 제공한 벤치마크 회로를 대상으로 실험한 결과 FM[3], LA-3[4], CLIP[14]에 비해 평균적으로 각각 33-44%, 45-50%, 10-12%의 cutsize 감소가 있었고 constructive method의 대표적인 분할 방법으로 알려진 Paraboli[10]와 MELO[11]에 비해 적은 수행 시간으로 각각 12%, 24%의 cutsize 감소 효과가 있었다.

Abstract

In this paper, we present a fast and efficient Iterative Improvement Partitioning(IIP) technique for VLSI circuits and hybrid bucket structures on its implementation. The IIP algorithms are very widely used in VLSI circuit partition due to their time efficiency. As the performance of these algorithms depends on choices of moving cell, various methods have been proposed. Specially, Cluster-Removal algorithm by S. Dutt[13][14] significantly improved partition quality. We indicate the weakness of previous algorithms where they used a uniform method for choice of cells during the improvement. To solve the problem, we propose a new IIP technique that selects the method for choice of cells according to the improvement status and present hybrid bucket structures for easy implementation. The time complexity of proposed algorithm is the same with FM[3] method and the experimental results on ACM/SIGDA benchmark circuits show improvement up to 33-44%, 45-50% and 10-12% in cutsize over FM[3], LA-3[4] and CLIP[14] respectively. Also with less CPU time, it outperforms Paraboli[10] and MELO[11] represented constructive-partition methods by about 12% and 24%, respectively.

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronic Engineering, Hanyang University)

接受日字:1998年1月22日, 수정완료일:1998年2月27日

Ⅰ. 서 론

대규모 집적회로 설계에 있어서 중요한 역할을 수행해 온 분할에 관한 연구는 크게 Iterative Improvement Partition(IIP)와 Constructive Partition방법[7][8][9][10][11]으로 나뉘어 발전되어 왔고 이들 방법을 조합한 Multilevel Partition방법[15]도 다양하게 제안되고 있다.

그러나 이러한 지속적인 연구에도 불구하고 대부분의 상용 자동 설계 시스템에서는 수행 속도와 그 결과의 안정성에 의거하여 IIP방법에 기반을 둔 알고리듬을 채택하여 이용하고 있는 실정이다.

IIP방법의 역사를 살펴보면 Kernighan과 Lin[1]은 그래프 분할 방법을 제안하였고 이를 필두로 하여 Schweikert와 Kernighan[2]은 이를 하이퍼그래프로 모델링함으로써 회로 설계 분야에 응용할 수 있는 초석을 만들었다. 그 뒤를 이어서 지금까지도 널리 사용되고 있는 Fiduccia와 Mattheyses[3]의 FM 방법이 출연하게 되었다.

FM 방법의 가장 큰 의의는 이전의 분할 개선 방법이 pair-wise exchange에 의한 과도한 수행 시간을 갖는 반면에 FM 방법에서는 single-cell movement 방법과 버켓(bucket) 구조를 제안하여 수행 시간을 대폭 줄였다는 점이다. 이러한 FM 방법에 기초를 두고 Krishnamurthy[4]는 look -ahead gain에 의한 확장된 FM 분할 개선 방식을 제안하였는데 이는 비교적 적은 규모의 회로 분할에 있어 보다 나은 결과를 얻을 수 있었다. 또한 FM에 기반을 둔 분할 방법이 가지고 있는 tie-breaking 발생 문제를 보완하기 위해 Hagen등[12]은 LIFO scheme을 제안하여 다소 성능 향상을 가져왔다. 이 이외에도 다양한 방법이 제안되었으나 IIP 방식의 분할 방법에 의한 성능은 FM 방법에 비해 크게 향상되지 못했다. 그러나 근래 S. Dutt[14]는 셀의 gain을 initial gain과 updated gain으로 나누어 고려함으로써 비약적인 IIP 방법의 성능 향상을 가져왔다. 이는 기존의 셀 선택 방식과는 다르게 updated gain에 의한 셀 선택 방식을 제안함으로써 클러스터를 cutset에서 초기에 제거하는 효과를 얻을 수 있었고 이 방식에 의한 분할 개선은 다른 IIP 방법에 의한 분할 개선 성능에 비해 탁월한 결과를 얻을 수 있었다.

최근 J. Cong[16]의 loose/stable net을 초기에 제거하고자 하는 방법이 제안되었으나 gain의 가중치에 의한 버켓 크기의 과도한 증가와 이를 보완하기 위한 임계치 설정이 알고리듬의 안정성에 영향을 주어 임계치의 설정에 따라 실험 결과가 다르게 나오는 문제점을 안고 있다.

S. Dutt의 CLIP(CLuster-oriented Iterative-improvement Partitioner)에서는 분할 개선 시에 즉각적인 cutsize 감소량(total gain)에 따른 셀 선택 방식이 local minimum에 머무를 가능성이 높다는 사실을 지적하고 즉각적인 cutsize 감소량보다는 이전 이동 셀에 의한 cutsize 변화량(updated gain)을 셀 선택의 판단 근거로 이용하는 방식을 제안하였고 FM 방법과 거의 대등한 수행 속도로 놀라운 분할 성능 향상을 가져왔다.

그러나 이러한 CLIP 방법도 해결되어야 할 문제점을 여전히 내포하고 있는데 이는 다음과 같다. 우선 updated gain에 의한 셀 선택 방식은 분할 개선 초기에 클러스터를 cutset에서 제거하는 동안에는 효과적으로 수행되지만 이 후에는 오히려 total gain에 의한 셀 선택이 상황에 따라서는 더 나은 결과를 얻을 수 있다는 점이다. 또한 단순히 이전 이동 셀에 의한 gain의 변화량만을 가지고 클러스터 제거 효과를 판단하는 것은 무리가 있다. 이는 이동된 셀과 그에 따라 gain의 변화가 생긴 셀과의 연결 상태에 따라 클러스터 제거 효과가 결정되기 때문이다. 이는 제 2장에서 다시 언급하도록 한다. 그 외에 버켓 크기의 증가에 따라 tie-breaking 문제는 기존에 방식에 비해 감소할 수 있으나 여전히 tie-breaking 문제를 내포하고 있다는 점이다.

본 논문에서는 서로 다른 두 종류의 버켓 구조을 제시하고 이를 효율적으로 운영하여 분할 개선 상황에 따라 셀 선택 방법이 결정되는 알고리듬을 제안함으로써 이상의 문제점을 보완하고 이를 실험하여 알고리듬의 효율성을 입증한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 분할 개선 방법을 FM과 CLIP을 중심으로 살펴보고 그 문제점을 제기한다. 제 3장에서는 본 논문에서 제안하는 하이브리드 버켓을 이용한 분할 개선 방법과 그 효과에 관해 논하고 제 4장에서는 ACM/SIGDA의 벤치마크 회로를 대상으로 실험한 결과에 대해 언급한다. 끝으로 제 5장에서는 제안된 방법에 대한 결론을 논한다.

Ⅱ. 기존의 분할 개선 방법 고찰

1. 기존 분할 개선 방법

집적회로에서의 분할에 대해 언급하기 위해 회로의 하이퍼그래프 모델링에 대하여 우선 간략히 소개한다. 회로는 그 구성 요소를 셀과 네트로 분해하여 하이퍼그래프 G = (V, E)로 표현할 수 있다. 여기서 V는 회로상의 셀의 집합이고 E는 네트의 집합이다. 일반적으로 회로에 있어서의 양 분할(two-way partition)은 이 하이퍼 그래프 G를 두 부분집합 V_1과 V_2로 분할하는 것을 의미하고 이때 V의 모든 셀은 V_1 혹은 V_2중에 어느 한 부분집합에만 반드시 속하도록 제한한다. 또한 cut이란 V_1과 V_2에 걸쳐 있는 네트을 의미하고 어느 한쪽 부분 집합에만 소속되어 있는 네트를 uncut이라 한다. Cutset이란 cut 상태를 갖는 네트의 집합을 의미하고 이 집합의 크기를 cutsize라 한다. 분할의 목적 함수는 부분집합 V_1과 V_2의 크기 조건(balance criterion)을 만족시키는 상태에서 cutsize를 최소화하는 부분집합을 결정하는 것이다.

분할 개선에 의한 분할 방법의 전형적인 알고리듬은 두 단계의 과정으로 이루어지는데 이는 random 함수 혹은 일정한 방법에 의해 주어진 크기 조건을 만족시키는 부분집합 V_1과 V_2를 만드는 초기 분할 단계와 이 초기 분할을 가지고 분할 개선 방법을 통해 cutsize를 감소시키는 분할 개선 단계이다. 분할 개선 단계는 pass라는 일련의 과정이 반복적으로 수행되어 얻어진다. 분할 개선 단계에서 pass의 반복은 한 pass과정이 수행된 후 cutsize의 감소가 발생하지 않을 때까지 수행된다. 이때 일반적으로 한 pass의 구성은 그림 1과 같다.

```
while(there exists free cells)
        c = pick the best cell;
        move and lock cell c;
        for(each net n incident to cell c)
                update gain of cells in net n;
endwhile
```

그림 1. 일반적인 분할 개선 알고리듬에서의 한 pass의 구조

Fig. 1. One pass of universal Iterative Improvement Partition algorithm.

분할 개선에 의한 분할 방식의 성능은 그림 1에서의 best cell 선택 방식에 의해 결정되어 진다. FM 방법의 경우 best cell은 하나의 셀 이동에 의해 얻어지는 즉각적인 cutset 감소량을 그 셀의 gain이라고 정의하고 현재 gain값의 순서에 입각하여 best cell을 선정한다. FM에 의거한 대부분의 분할 개선 방법들은 이러한 즉각적인 gain에 근거한 셀 선택 방식을 채택하였다. 임의의 셀 **u**의 gain은 다음과 같이 정의 된다.

$$\text{Gain } g(\mathbf{u}) = \sum_{n_i \in E(\mathbf{u})} c(n_i) - \sum_{n_j \in I(\mathbf{u})} c(n_j)$$

여기서 $E(\mathbf{u})$는 현재 u와 연결되어 있는 네트 중에서 cutset에 속해 있는 네트를 의미하고 $I(\mathbf{u})$는 이동 후에 cutset에 추가되는 네트를 의미한다. 또한 $c(n_i)$는 네트 n_i의 부여되는 가중치(weight, cost)이다.

반면에 CLIP에서 제안된 방식은 하나의 셀 gain을 initial gain과 updated gain으로 분류하여 initial gain은 초기 분할 직후에 그 셀의 이동에 따른 즉각적인 cutsize 감소량을 의미하고 updated gain은 다른 셀의 이동에 의해 영향 받은 그 셀의 gain 변화량을 의미한다. 따라서 updated gain이 높은 셀은 이전에 이동한 셀에 의해 끌리는 힘이 크다고 해석할 수 있고 CLIP 방법에서는 이 끌리는 힘 즉, updated gain이 높은 셀을 best cell로 선택하여 우선 이동 한다. 이러한 선택 방식은 하나의 셀의 이동이 이와 연결된 또 다른 셀의 이동을 불러 일으킨다는 사실을 직관적으로 알 수 있다. 이러한 사실은 분할 개선 시에 한 cluster내의 임의의 셀이 **u**가 이동하면 다른 cluster내의 셀에 우선하여 **u**가 속해 있는 cluster내의 다른 셀들의 이동을 연쇄적으로 촉진시키는 효과를 얻을 수 있고 이는 한 cluster가 분할된 양 부분에 의해 나뉘어 지는 것을 방지(Cluster-Removal)할 수 있다. 이러한 새로운 셀 선택 방법은 분할 개선 방법의 성능 개선에 크게 기여했다.

2. 클러스터 제거에 의한 분할 개선 방법의 문제점

S. Dutt에 의한 updated gain에 의한 셀 선택 방식은 cutset상의 클러스터를 제거함에 있어 매우 효과적이다. 그러나 이러한 방식을 적용하면 대부분의 cluster는 분할 개선 초기에 분할 부분집합 V_1 혹은 V_2로 우선적으로 할당되어 그 이후 분할 개선의 셀 선택 방법으로 updated gain을 이용하는 것은 문제점으로 지적할 수 있다. 즉 오히려 즉각적으로 얻어질 수 있는 total gain에 의한 셀 선택이 상황에 따라서는 더 나은 결과를 얻을 수 있다는 점이다. 따라서 분할 개선의 진행 상황에 따라 셀 선택 방법이 다르게 적용되어야 한다.

또한 예를 들어 그림 2와 같은 상황에서 동일한 updated gain의 부여는 Cluster-Removal 효과를 얻을 수 없다. 이는 이미 이동한 셀에 의해 클러스터 제거 효과를 상실한 셀에 대해서는 best cell로 우선 선택될 필요가 없다는 것을 의미한다.

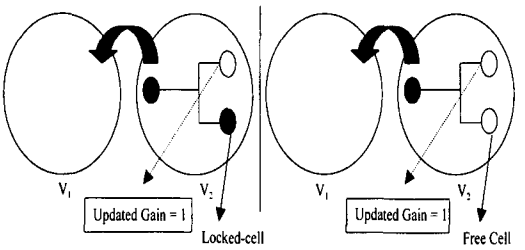


그림 2. Updated gain의 불합리한 할당 예
Fig. 2. An example of the absurd assignment of updated gain.

그림 2에서와 같이 이동하고자 하는 셀을 선택하여 이동한 후 이와 연결된 셀에 대하여 updated gain의 할당 시에 그 연결 네트상에 이미 양쪽 분할 부분(V_1과 V_2)내에 이미 이동된 셀(locked cell)이 최소한 각각 하나 이상이 존재하면 이미 clustering을 위한 updated gain의 효과는 상실되었다고 볼 수 있다. 따라서 이러한 경우의 updated gain 할당 시에 차별화 시킬 필요가 있다.

FM 이나 CLIP을 비롯한 분할 개선 방식은 tie-breaking이라는 공통된 문제점을 내포하고 있는데 이는 같은 best cell 조건을 만족하는 다수의 셀 중에 어느 셀을 우선하여 이동시키는가에 따라 결과가 매우 달라짐으로써 발생하는 문제이다. 이의 완전한 해결책은 지금으로서는 없지만 이러한 tie-breaking 문제를 최소화 시키는 방법은 선택 조건의 다변화에 있다. 본 논문에서는 이러한 선택 조건의 다변화를 통해 위에서 언급한 CLIP 방법의 몇 가지 문제점을 해소하고 tie-breaking 문제를 감소시키는 효과와 아울러 이의 간단한 구현을 위한 버켓 구조를 제 3장에서 논한다.

Ⅲ. 하이브리드 버켓을 이용한 분할 개선 방법

1. Locked-net의 정의

본 논문에서 제안하는 분할 방법을 언급하기 위해 Locked-net이라는 용어를 정의 한다. Locked-net은 cutset에 속한 네트 중에서 각 분할된 부분집합 V_1와 V_2에 각각 최소한 하나 이상의 Locked-cell를 가지고 있는 네트를 의미한다. 참고적으로 Locked-cell은 다른 분할 파트로의 이동이 금지된 셀 즉, 한 pass내에서 이미 이동된 셀과 설계자에 의해 강제로 이동이 금지된 셀을 의미한다.

2. 가변적인 셀 선택 방식

CLIP 방법에서 제안한 updated gain의 활용은 Cluster-Removal의 효과를 위해 반드시 필요한 요소지만 이를 전반적인 분할 개선 방법으로서 적용하는 방법은 위에서 언급한 바와 같이 몇 가지 문제점을 내포한다. 따라서 본 논문에서는 이러한 best cell 선택 방식을 보완하기 위해 updated gain에 의한 셀 선택 방식과 total gain에 의한 셀 선택 방식을 동시에 채택한다.

이러한 가변적인 셀 선택 방식을 이용함으로써 분할 개선 진행 중에 클러스터 제거 효과를 얻을 수 있는 셀을 발견하면 updated gain에 따라 우선적으로 이동하고 클러스터 제거 효과가 없거나 이미 상실한 셀에 대해서는 total gain에 의해 셀을 선택하도록 하여 셀의 상황에 따라 적절한 선택 방식을 적용할 수 있다.

이는 다음과 같은 방식으로 적용된다. 각 셀의 이동 시에 CLIP 방법과 마찬가지로 initial gain과 updated gain을 유지한다. Updated gain을 계산하는 시점에서 이전 이동 셀에 의해 updated gain이 증가하는 셀 중에서 Locked-net에 연결되지 않은 셀은 updated gain에 의한 셀 선택 방법을 이용하고 나머지 셀의 경우에는 total gain에 의한 셀 선택 방식을 이용한다.

이를 구현 관점으로 보면 updated gain에 의한 버켓(Major bucket)과 total gain에 의한 버켓(Minor bucket)을 동시에 유지하고 임의의 순간에 임의의 셀은 두 버켓 중에 하나의 버켓에 소속하도록 한다. 초기 분할 직후에는 모든 셀이 Minor bucket에 소속되도록 하고 total gain에 의해 best cell을 선택한다. 하나의 셀이 이동을 하면 이에 의해 영향을 받는 셀의 updated gain 계산 시에 그 셀을 버켓을 결정한다. 이때 Major bucket에 소속될 수 있는 셀은 클러스터 제거 효과를 갖는 셀로 국한하였으므로 updated gain이 증가하는 셀 중에 Locked-net에 연결되지 않은 셀로 제한한다. 이는 한 셀의 updated gain이 증가하

였다는 것은 이전 이동 셀이 이동한 파트로 그 셀을 이동시키려는 힘이 작용했다는 것을 의미하고 이것이 Locked-net에 연결되지 않았다는 것은 실제 이동 시키면 클러스터 제거 효과를 얻을 수 있다는 사실에 근거한다. 클러스터 제거 효과를 얻기 위해 두 버켓 중에 Major bucket상의 셀이 존재하면 우선적으로 best cell로 선택한다. 위에서 언급한 방법을 기술하면 그림 3과 같다.

```
while(there exists free cells)
    c = pick cell with maxmum upated gain
        from Major bucket;
    if(c == NULL)
        c = pick cell with maxmum total gain
            from Minor bucket;
    move and lock cell c;
    for(each net n incident to cell c)
        update gain of cells in net n
        and move the cell to major bucket if
        (the cell is updated gain > 0 and not
        in Locked-net);
endwhile
```

그림 3. 가변적인 셀 선택 알고리듬의 한 pass의 구조
Fig. 3. One pass of dynamic cell selection algorithm.

이러한 가변적인 셀 선택 방법을 이용함으로써 다음과 같은 효과를 얻을 수 있다. 우선 updated gain에 의한 버켓을 선택의 우선권이 있는 Major bucket으로 유지하고 updated gain이 증가한 셀을 수용함으로써 기존의 클러스터 제거 방법에 비해 확실한 효과를 얻을 수 있다. 기존의 방식은 updated gain이 감소하는 셀, 즉 클러스터 내의 셀 중에 임의의 셀이 이동되어 간 파트의 연결된 셀에 대하여 updated gain을 감소시키고 이를 셀 선택에 반영하였으나 이는 클러스터 제거 효과를 위한 우선 선택의 개념과는 상충된다. 즉 클러스터 제거 효과는 total gain에 관계없이 클러스터내의 셀을 우선적으로 이동 시킴으로써 얻어질 수 있는데[14] 이때 updated gain의 감소는 이동 제한의 역할을 할 뿐이며 우선 선택의 역할은 updated gain의 증가량이 반영한다. 또한 updated gain의 증가에도 불구하고 Locked-net에 의한 updated gain의 증가는 클러스터 제거 효과를 얻을 수 없으므로 이는 우선 선택 대상에서 제외된다. 그리고 이러한 방법을 이용함으로써 클러스터 제거 효과가 있는 상황 즉, 분할 개선 초기에만 updated gain에 의한 셀 선택 방법을 이용하고 그 외의 경우에 있어서는 대부분의 셀이 Minor bucket에 존재하여 일반적인 total gain에 의한 셀 선택 방법을 적용할 수 있다.

이와 같은 셀 선택 방식을 적용하기 위한 Major/Minor bucket 구조는 그림 4와 같다. 그림 4에서와 같이 Major bucket은 $4*p_{max}$의 크기를 갖고 Minor bucket은 $2*p_{max}$의 크기를 갖으며 이때 p_{max}는 주어진 회로에서 특정 셀에 연결되어 있는 네트 가중치의 총합 중 최대값을 의미한다. 이러한 버켓의 유지는 tie-breaking의 감소에도 많은 효과가 있는데 이는 우선 셀이 존재할 수 있는 버켓 영역이 확장된 것에 따라 확률적으로 tie-breaking의 발생 빈도를 낮출 수 있고 Major bucket의 경우 tie-breaking이 발생하면 total gain의 크기에 따라 선택할 수 있고 Minor bucket의 경우에는 updated gain에 의해 선택될 수 있다. 이 경우에 있어 tie-breaking이 발생하면 2차 선택 조건은 initial gain에 비례(Major)/반비례(Minor)하므로 추가적인 계산 없이 바로 선택할 수 있다.

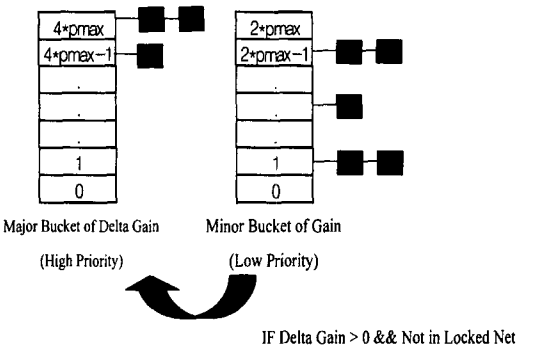


그림 4. 제안된 하이브리드 버켓 구조
Fig. 4. Proposed structures of hybrid buckets.

Ⅳ. 실험 및 고찰

제안된 알고리듬은 그림 3과 같이 기존의 FM 방법이나 CLIP 방법과 동일한 time complextity를 갖는다. 이는 gain의 변화량 계산 시에 즉각적으로 판단할 수 있는 조건에 따라 소속 버켓을 바꾸는 단계만이 추가적으로 필요하다. 물론 두 종류의 버켓을 유지함에 따른 약간의 계산이 필요하나 이는 수행 속도에는 거의 영향이 없다.

본 알고리듬은 ACM/SIGDA에서 제공한 벤치마크

회로 21개에 의해 실험되었다. 벤치마크 회로의 주요 사양은 표 1과 같다.

표 1. 실험 대상 회로의 주요 사양
Table 1. Benchmark circuit characteristics.

Circuits	# of Cells	# of Nets	# of Pins
p1	833	902	2908
bm1	882	903	2910
t4	1515	1658	5975
t3	1607	1618	5807
t2	1663	1720	6134
t6	1752	1541	6638
struct	1952	1920	5471
t5	2595	2750	10076
19ks	2884	3282	10547
p2	3014	3039	11219
s9324	5866	5844	14065
biomed	6514	5742	21040
s13207	8772	8651	20606
s15850	10470	10383	24712
industry2	12637	13419	48404
industry3	15406	21924	68290
s35932	18148	17828	48145
s38584	20995	20717	55203
avq.small	21918	22124	76231
s38417	23949	23843	57613
avq.large	25178	25384	82751

표 2에서는 FM, LA-3(Look Ahead Level 3)에 의한 실험 결과와 CLIP을 FM 방식으로 적용한 실험 결과를 가지고 본 논문에서 제안한 분할 개선 방법(HYIP)을 FM 방식으로 적용한 결과를 비교하였다. 이 실험은 random 함수를 이용한 동일한 초기 분할을 가지고 각 방법을 수행하여 분할 개선 효과를 측정하였다. 실험 결과 FM에 비해서는 33-44%, LA-3에 비해서는 44-50%가량의 cutsize 감소를 가져왔고 동일 time complexity를 갖는 기존 방법 중에 가장 뛰어난 결과를 가진 CLIP에 비해 10-12%의 성능 향상을 얻었다. 특히 대상 회로의 크기가 커질수록 평균 cutsize의 감소가 많아짐은 주목 할 만 하다.

표 3에서는 constructive method 중에 대표적인 분할기인 Paraboli와 MELO와의 수행 결과를 비교하였는데 본 알고리듬 및 FM, CLIP은 100회 수행한 결과이다. 100회를 수행 했음에도 불구하고 적은 수행 시간 안에 Paraboli에 비해 약 12%, MELO에 비해 약 24%의 Cutsize 감소가 있었다. 표 2와 3에서와 같이 대부분의 실험 대상 회로에 대해서 기존 방법에 비해 향상된 결과를 얻을 수 있었으며 이는 일반 회로

표 2. 기존 IIP 방법과의 Cutsize 비교 실험 결과
Table 2. Comparison of HYIP and other IIP based algorithms.

Circuit	Minimum Cut Size of 20 Runs							Average Cut Size of 20 Runs						
	Cut Size				Improvement(%)			Cut Size				Improvement(%)		
	FM	LA-3	CLIP FM	HYIP FM	HYIP (FM)	HYIP (LA)	HYIP (CLIP)	FM	LA-3	CLIP FM	HYIP FM	HYIP (FM)	HYIP (LA)	HYIP (CLIP)
P1	47	52	52	47	0.0	9.6	9.6	74.9	68.5	65.8	60.6	19.1	11.5	7.9
Bm1	54	53	49	47	13.0	11.3	4.1	79.5	67.5	65.0	58.3	26.7	13.6	10.3
t4	87	82	56	54	37.9	34.2	3.6	129.3	117.2	77.0	66.6	48.5	43.2	13.5
t3	75	80	57	58	22.7	27.5	- 1.7	106.8	106.2	72.3	65.4	38.8	38.4	9.5
t2	149	126	89	90	40.0	28.6	- 1.1	182.1	148.1	105.2	102.8	43.6	30.6	2.3
t6	67	70	60	63	6.0	10.0	- 4.8	94.2	84.4	70.1	73.3	22.2	13.2	-4.4
struct	46	44	37	33	28.3	25.0	10.8	58.0	49.6	45.6	41.6	28.3	16.1	8.8
t5	127	99	75	75	41.0	24.2	0.0	183.6	165.0	89.0	85.6	53.4	48.1	3.8
19ks	140	130	119	107	23.6	17.7	10.1	171.7	169.0	150.3	128.1	25.4	24.2	14.8
p2	212	149	149	143	32.5	4.0	4.0	273.9	233.4	233.2	211.6	22.8	9.3	9.3
s9324	59	43	49	45	23.7	- 4.4	8.2	84.7	81.1	89.5	66.1	22.0	18.5	26.2
biomed	83	90	84	84	- 1.2	6.7	0.0	117.4	170.7	108.4	98.9	15.8	42.1	8.8
s13207	98	85	98	70	28.6	17.7	28.6	122.6	118.9	123.5	101.6	17.1	14.6	17.7
s15850	109	87	80	85	22.0	2.3	- 5.9	176.9	140.4	140.9	121.0	31.6	13.8	14.1
ind2	264	422	260	185	29.9	56.2	28.9	627.5	732.4	369.6	298.9	52.4	59.2	19.1
ind3	272	504	261	241	11.4	52.2	7.7	506.0	758.0	376.6	338.7	33.1	55.3	10.1
s35932	85	168	102	89	- 4.5	47.0	12.8	210.3	231.4	127.1	123.6	41.2	46.6	2.8
s38584	100	85	49	52	48.0	38.8	- 5.8	299.8	271.2	83.2	88.5	70.5	67.4	-6.0
avq.sml	347	608	223	187	46.1	69.2	16.1	578.6	815.5	335.1	274.2	52.6	66.4	18.2
s38417	240	284	78	75	68.8	73.6	3.9	384.1	408.2	136.7	112.1	70.8	72.5	18.0
avq.lar	350	398	216	184	47.4	53.8	14.8	755.0	693.4	305.2	272.3	63.9	60.7	10.8
Total	3011	3659	2243	2014	33.1	45.0	10.2	5217	5630	3169	2789	46.5	50.4	12.0
Average of % Improvement					26.4	28.8	6.9					38.6	36.4	10.3

에 본 논문에서 제안된 방법을 적용할 경우 기존 방법에 의한 결과에 비해 일반적으로 더 많은 cutsize 감소를 얻을 것으로 예측할 수 있다.

표 3. 기존 Constructive 방법과의 Cutsize 비교 실험 결과
Table 3. Comparison of HYIP and other constructive algorithms.

Circuit	Cut Size				
	Paraboli	MELO	FM	CLIP FM	HYIP FM
P1	53		47	47	47
bm1			49	47	47
t4			80	53	53
t3		64	62	56	56
t2		48	124	87	88
t6		61	60	60	60
struct	40	60	41	33	32
t5		109	104	74	72
19ks		90	130	109	105
p2	146	38	182	148	143
s9324	74	102	51	44	45
biomed	135	119	83	83	83
s13207	91	169	78	76	70
s15850	91	79	104	75	70
ind2	193	115	264	174	185
ind3	267	104	263	241	241
s35932	62	52	85	83	58
s38584	55	319	63	47	47
avq.sml	224		297	200	177
s38417	49		147	66	63
avq.lar	139		350	185	169
Total	1619	1529	2664	1988	1430 1156 1911
Improv-e ment	*	*			11.7(%) 24.4(%)

끝으로 표 4에서는 위에서 언급한 각 방법의 수행 시간을 비교하였다. Paraboli는 DEC3000 Model 500 AXP에서 수행된 결과이고 MELO와 본 알고리듬은 SUN SPARC 10에서 수행된 결과이며 나머지는 SUN SPARC5 Model 85에서 수행된 결과이다. 비록 서로 다른 종류의 workstation상에서 수행되었지만 거의 비슷한 성능으로 가정할 수 있다[14]. 본 알고리듬은 표 4에서와 같이 기존의 constructive 방법에 비해서는 적은 수행 시간이 소요되었고 기존 IIP 방식의 FM과 CLIP 방법에 대해 대등한 수준의 수행 속도를 갖는다.

V. 결 론

본 논문에서는 대규모 집적회로 분할을 위한 빠르고 효율적인 분할 개선 방법을 제안하고 이를 구현하기 위한 하이브리드 버켓 구조를 제시하였다. Cutset상의 cluster를 제거하는데 효과적인 updated gain에 의한 셀 선택 방식은 분할 개선 초기에는 효과적으로 수행되지만 이 이후에는 오히려 total gain에 의한 셀 선택이 더 나은 결과를 얻을 수 있다는 점에 착안하여 서로 다른 두 종류의 버켓 구조을 제안하고 이를 효율적으로 운영하여 분할 개선 상황에 따라 셀 선택 방법이 결정되는 알고리듬을 제안하였고 또한 단순히 이전에 이동한 셀에 의한 gain의 증가량(updated gain)이 많은 셀이라고 해서 직접적으로 Cluster-Removal 효과를 얻을 수 있는 셀은 아닐 수 있다는 사실에 근거한 Locked-net의 개념을 알고리듬에 적용하였다. 그리고 제안된 버켓 구조을 이용함으로써 tie-breaking의 발생 빈도를 낮출 수 있다는 점을 제시하였다.

표 4. 분할 방법별 수행 시간
Table 4. Comparison of CPU times of various algorithms.

Circuit	CPU Times				
	Paraboli	MELO	FM (x100)	CLIP FM (x100)	HYIP FM (x100)
p1	18.3		0.27	0.44	0.18
bm1			0.58	0.43	0.20
t4		8	0.76	0.90	0.50
t3		9	0.58	0.96	0.50
t2		24	0.57	0.99	0.58
t6		27	0.54	0.88	0.52
struct	35.2	29	1.13	0.69	0.35
t5		31	1.59	1.62	1.09
19ks		38	1.81	2.23	1.03
p2	137.4	67	2.78	2.37	1.38
s9324	490.3	79	3.89	3.14	2.49
biomed	710.9	89	4.23	3.34	3.97
s13207	2060.4	516	4.24	5.01	5.11
s15850	2730.9	496	9.10	6.87	6.96
ind2	1367.3	710	11.07	13.17	17.02
ind3	760.7	1197	11.82	16.35	19.40
s35932	2626.7	1855	13.70	13.21	15.37
s38584	6517.5	0.33	18.44	15.73	19.37
avq.sml	4098.9		15.36	20.50	23.40
s38417	2041.5		19.49	17.70	21.71
avq.lar	4135.0			24.07	28.00
Total	27731	5175	12228	15060	16471 4188 16913

참 고 문 헌

[1] B. W. Kernighan and S. Lin, An Efficient

Heuristic Procedure for Partitioning Graphs, Bell System Tech. Journal, vol. 49, Feb. 1970, pp. 291-307.

[2] D. G. Schweikert and B. W. Kernighan, A Proper Model for the Partitioning of Electrical Circuits, Proc. 9th Design automation workshop, 1972, pp. 57-62.

[3] C. M. Fiduccia and R. M. Mattheyses, A linear-time heuristic for improving network patitions, Proc. ACM/IEEE Design Automation Conf., 1982, pp. 175-181.

[4] B. Krishnamurthy, An improved min-cut algorithm for partitioning VLSI networks, IEEE Trans. on Comput., vol. C-33, May 1984, pp. 438-446.

[5] Y. C. Wei and C. K. Cheng, Towards efficient hierarchical designs by ratio cut partitioning, Proc. Intl. Conf. Computer-Aided Design, 1989, pp. 298-301.

[6] Y. C. Wei and C. K. Cheng, An Improved Two-way Partitioning Algorithm with Stable Performance, IEEE Trans. on Computer-Aided Design, 1990, pp. 1502-1511.

[7] L. Hagen and A. B. Kahng, Fast Spectral Methods for Ratio Cut Partitioning and Clustering, Proc. Intl. Conf. Computer-Aided Design, 1991, pp. 10-13.

[8] J. Cong and M. Smith, A bottom-up clustering algorithm with applications to circuit partitioning in VLSI designs, Proc. ACM/IEEE Design Automation Conf., 1993, pp. 755-760.

[9] C. J. Alpert and A. B. Kahng, A General Framework for Vertex Orderings, With Applications to Netlist Clustering, Proc. Intl. Conf. Computer-Aided Design, 1994, pp. 63-67.

[10] B. M. Riess, K. Doll and F. M. Johannes, Partitioning Very Large Circuits Using Analytical Placement Techniques, Proc. ACM/IEEE Design Automation Conf., 1994, pp. 646-651.

[11] C. J. Alpert and S-Z Yao, Spectral Partitioning: The More Eigenvectors, the Better, Proc. ACM/IEEE Design Automation Conf., 1995.

[12] L. W. Hagen, D. J. Hwang and A. B. Kahang, On implementation choices for iterative improvement partitioning methods, Proc. European Design Automation Conf., Sept. 1995, pp. 144-149

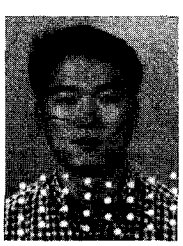
[13] S. Dutt and W. Deng, A Probaility-Based Approach to VLSI Circuit Partitioning, Proc. ACM/IEEE Design Automation Conf., 1996, pp. 100-105.

[14] S. Dutt and W. Deng, VLSI Circuit Partitioning by Cluster-Removal Using Iterative Improvement Techinigues, Proc. Intl Conf. Computer-Aided Design, 1996, pp. 194-200.

[15] C. J. Alpert, J-H Huang and A. B. Kahng, Multilevel Circuit Partitioning, Proc. ACM/IEEE Design Automation Conf., 1997, pp. 530-533.

[16] J. Cong, H. P. Li, S. K. Lim, T. Shibuya and D. Xu, Large Scale Circuit Partitioning With Loose/Stable Net Removal And Signal Flow Based Clustering, Proc. Intl Conf. Computer-Aided Design, 1997, pp. 441-446.

저 자 소 개



任 昌 慶(正會員)

1990년 2월 한양대학교 전자공학과 졸업(학사). 1992년 2월 한양대학교 대학원 전자공학과 졸업(공학석사). 1992년 3월 ~ 현재 한양대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 CAD 및 집적회로 설계, Network Management System 등

鄭 正 和(正會員) 第 33卷 A編 第 1號 參照

현재 한양대학교 전자공학과 교수 재직