

# Circuit Partitioning Algorithm Using Wire Redundancy Removal Method

김진국, 권기덕, 신봉식, 정정하  
한양대학교 전자통신전과 공학부  
전화 : 02-2290-0558

Jin-kuk kim, Ki-duk Kwon, Bong-sik Sihm, Jung-wha Chong  
Department of Electronic Engineering Hanyang University  
E-mail : jinfly@ihanyang.ac.kr

## Abstract

This paper presents a new circuit partitioning algorithm using wire redundancy removal. This algorithm consist of the two steps. In the first step, We propose a new IIP(Iterative Improvement Partitioning) technique that selects the method to choice cells according to improvement status using two kinds of bucket structures, the one kept by total gain, and the other by updated gain. In the second step, we select the target wire in the cut-set. We add a alternative wire in the circuit to remove the target wire. For this we use wire redundancy removal and addition method

The experimental results on MCNC benchmark circuits show improvement up to 41-50% in cut-size over previous algorithms

## I. 서론

회로 분할은 하나의 칩에 구현하기에는 주어진 회로의 크기가 너무 크거나, 회로의 입출력 핀의 개수가 하나의 칩에 연결된 입출력 핀의 개수를 초과할 경우에 주어진 회로를 몇 개의 작은 회로로 나누는 것을 말한다. 이는 회로의 복잡도를 줄일 수 있을 뿐만 아니라

전체 시스템의 신뢰성과 성능을 크게 개선 할 수 있다.

회로 분할은 크게 그래프 정보를 이용하는 방법과, 회로 정보를 이용하는 방법으로 구분된다. 그래프 정보를 이용하는 분할 알고리즘은 단순히 노드의 접합 정보만을 사용하여 분할을 수행하기 때문에 구현이 간단하지만, 실제 회로가 가지고 있는 정보를 고려하지 않기 때문에 최적의 분할 결과를 얻지 못한다. 반면 회로 정보를 이용하는 분할은 최적의 분할 결과를 얻을 수 있으나, 오버헤드가 크고, 분할 수행의 시간이 오래 걸린다는 단점이 있다.

본 논문에서는 이러한 분할 알고리즘의 문제점을 개선하기 위한 새로운 분할 알고리즘을 제시한다. 1차적으로 그래프 정보를 이용한 분할 알고리즘을 통해서 초기 분할을 수행한 후, 분할된 절단집합에 있는 신호선을 대상으로 회로 정보를 이용하여 2차 분할을 수행한다. 이를 위해서 분할 개선 진행 상황에 따라 노드 선택 방식이 가변하는 분할 방법과 절단집합에 있는 신호선을 제거하기 위한 방법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 분할 방법에 대해서 알아보고 그 문제점을 제기한다. 제 3장에서는 본 논문에서 제안하는 분할 알고리즘에 대해서 알아보고, 제 4장에는 MCNC 벤치마크를 회로를 대상으로 실험한 결과에 대해서 다룬다. 마지막으로

제 5장에서는 결론을 논한다.

## II. 기존의 회로 분할 방법

회로 분할 방법은 크게 그래프 정보를 이용하는 분할 방법과 회로 정보를 이용하는 방법이 있다. 그래프 정보를 이용한 대표적인 분할 방법은 FM 분할 알고리즘 [1]이다. FM 방법의 특징은 single-cell movement 방법과 버킷(bucket) 구조를 사용하여 수행 시간을 대폭 줄였다는 점이다. 그러나 FM 알고리즘은 동순위 경합(Tie-break) 문제를 여전히 내포하고 있다. 이를 개선하기 위해서 CLIP 알고리즘[2]이 제안되었다. 이는 이득 변화량(Updated gain)에 의한 노드 선택 방식을 사용함으로써 클러스터를 절단집합(*cut set*)상에서 초기에 제거하는 효과를 얻는다. 그러나 분할 개선 초기에 클러스터를 제거하는 동안에는 효과적으로 수행되지만 클러스터가 절단집합상에서 이미 제거되었을 경우에는 더 나쁜 결과를 가져 올 수 있다는 단점이 있다. 그래프를 이용한 분할 방법은 회로를 간단한 그래프의 형태로 표현하고, 각 노드 간의 집합 정보를 사용하지만, 회로 정보를 이용한 분할 알고리즘은 각각의 노드들에 대해서 회로 변형, 불필요한 게이트의 제거, 회로 정보를 이용한 게이트의 복사 등의 방법을 통해 절단집합의 수를 줄이는 방법을 사용한다. 실제 회로의 논리 정보를 이용하기 때문에 분할 결과에 있어서는 그래프를 이용한 분할 방법보다 좋은 성능을 보이지만, 전체 회로에 대해서 회로 정보를 이용하여 분할을 수행 할 경우, 구현이 복잡하고, 분할 수행 시간에 있어서도 그래프를 이용한 분할 방법보다 오래 걸린다는 단점이 있다.

## III. 중복 신호선 제거 방법을 이용한 회로 분할 방법

본 논문에서는 이러한 문제점을 해결하기 위하여 2 단계로 구성된 새로운 분할 알고리즘을 제안한다. 1차적으로 가변적인 셀 선택 방식의 CLIP 알고리즘을 통해서 분할을 수행한 후, 2차적으로는 그래프 정보를 이

용한 분할 결과를 통해서 절단집합상의 신호선을 대상으로 회로 정보를 이용하여 대상 신호선(Target Wire)을 결정하고, 이 것을 제거하기 위한 대상 신호선(Alternative wire)를 회로에 추가해서 대상 신호선을 제거하여 기능적으로 동일한 회로로 변형하는 방법을 사용한다.

### 3.1 가변적인 노드 선택방식의 CLIP 알고리즘

본 절에서는 이득 변화량에 의한 노드 선택 방식과 총 이득에 의한 노드 선택 방식을 가변적으로 사용하는 CLIP 알고리즘에 대해 다루도록 하겠다. 이를 위해서 Locked-edge와 Locked-node라는 새로운 용어를 정의 하도록 하겠다. Locked-edge는 절단 집합에 속한 에지 중세어 분할된 부분 집합 V1과 V2에 각각 최소한 하나 이상의 Locked-node를 가지고 있는 에지를 의미한다. Locked-node는 다른 부분 집합으로의 이동이 금지된 노드 즉, 이미 이전에 한 번 이동된 노드를 말한다. 이는 그림 3-1에서 보여주고 있다.

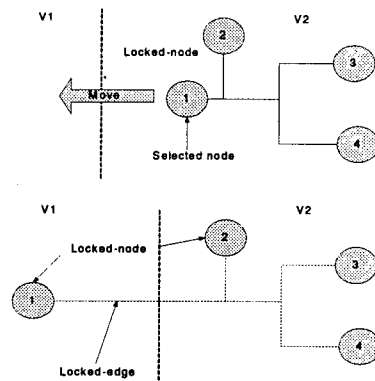


그림 3 - 1 Locked-node 와 Locked-edge 의 예

Locked-edge 상에 존재하는 노드는 이미 클러스터 제거 효과가 없는 노드로 생각 할 수 있다. 이러한 경우에는 총 이득에 따라 노드를 선택한다. 반면 Locked-edge 상에 존재하지 않는 노드는 이미 이전에 이동한 노드에 의해서 클러스터 제거 효과가 있는 노드를 의미한다. 그러므로 이러한 노드는 이득

변화량에 따라 노드를 선택해야 한다. 단 이러한 방법을 적용하기 위해서는 이득 변화량에 의한 선택이 총 이득에 의한 선택보다 우선 고려되어야 한다. 이는 이득 변화량에 의한 선택이 가능한 노드가 존재한다는 것은 클러스터가 아직 절단집합상에 존재한다는 것을 의미하므로 이는 우선적으로 제거되어야 하기 때문이다.

위와 같이 가변적인 노드 선택 방식의 사용하기 위해서는 이득 변화량에 의한 버킷(Major bucket)과 총 이득에 의한 버킷(Minor bucket)을 같이 구현해야 한다. 이는 CLIP 방법에서나 FM 방법에서 사용되는 버킷과 동일한 구조를 갖으며 Major bucket 은 이득 변화량에 의해 운용되고 Minor bucket 은 총 이득에 의해 운용된다. 두 버킷 상의 노드의 이동 조건은 이득 변화량이 0 보다 크거나 같고 Locked-edge 와 연결되어 있지 않으면 그 노드는 Major bucket 에 소속 될 수 있다. 전체적인 분할 알고리즘을 기술하면 그림 3-2 와 같다.

```

Step 1. Calculate the total gain of nodes.
Step 2. Insert th nodes into bucket array.
        Select the maximum total gain node n
        as the first base node to move
Step 3. Move node n and lock node n;
Step 4. while (there exists free cells)
        n = pick node with maximum updated gain;
        If (n == NULL)
            n = pick node with maximum total gain;
        Move node n and lock node n;
        For (each edge e incident to node n)
            Update updated gain of nodes in edge e
            If (updated gain ≥ 0
                and not in Locked-edge);
                move the node to major bucket;
            Else Move the node to minor bucket;
        Endfor;
    Endwhile;
End.
    
```

그림 3-2 가변적인 셀 선택 방식의 CLIP 방법

### 3.2 Wire Redundancy Removal 을 이용한 회로 분할 알고리즘

1차적으로 가변적인 노드 선택 방식의 CLIP 방법을 통해 분할을 통해 얻어진 절단집합의 신호선을 대상으로 타겟 신호선(Target Wire)을 결정하고, 기능적으로

동일한 대상 신호선(Alternative wire)를 회로에 추가해서 타겟 신호선을 제거하여 절단집합의 신호선의 개수를 줄이는 방법을 사용한다. 그림 3- 3은 이에 대한 예를 보여주고 있다. 이와 같이 신호선의 중복을 판별하고 제거하기 위해서는 그 신호선의 중복 결합(Redundancy fault)의 유무를 판단해야 한다. 이를 위해 먼저 그 신호선에서 주출력(Primary output)까지의 경로를 설정하여 지배자들(dominators)를 결정하고, 이들의 side input의 값을 제어 불가능값(non-controlling value)로 고정한다. AND(NAND) 게이트는 ' 1' 값을 가지고, OR(NOR) 게이트는 ' 0' 값을 가진다. 그 다음 SMA(Set of Mandatory Assignment)[3][4][5]를 계산하고, 그 값들을 비교하여 만약 서로 일치하지 않는 값이 나온다면, 그 신호선은 중복 결합을 가지는 것이고, 이는 이 신호선이 회로에서 중복이 된다는 것을 말한다.

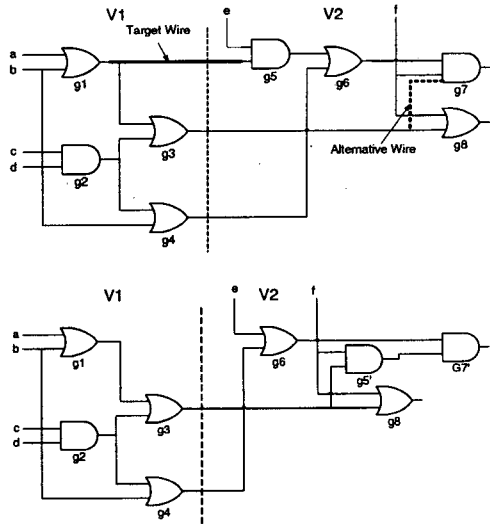


그림 3- 3 중복 신호선 제거를 이용한 회로 분할 예

예를 들면 그림 3-3에서 신호선(g3->g7)의 중복 고장을 판별하기 위해서  $SMA = \{g=0, g6=1, f=1, g1=0, g2=0, g5=0, a=0, b=0, g4=0, g6=0\}$ 가진다. 그러나 g6의 값이 서로 일치하지 않으므로 신호선(g3->g7)은 중복 결합이고, 회로에서 중복이 된다.

전체적인 알고리즘은 그림 3- 4와 같다.

```

Step 1. Perform CLIP using Adaptive cell choice;
Step 2. select a cut wire wt and Compute SMA of wt;
Step 3. For (each gate gi in the circuit)
    If(gi has mandatory assignment)
        Insert gi into source_array;
    If(gi is a dominator)
        Insert gi into the destination array;
Step 4. For (each gs in the source array)
    Add wire gs to gd;
    If(redundant) then Add wire(gs , gd);
    and remove wt;
Step 5. If (a cut wire is exist) then return Step 2;
End.
    
```

그림 3-4 중복 신호선 제거를 이용한 회로 분할

#### IV. 실험 결과

본 논문의 분할 알고리즘은 C++로 구현되었으며, 펜티엄 3.0Gh, 512Mb 메모리의 컴퓨터에서 실행되었으며, MCNC 벤치마크 13 개의 회로를 대상으로 실험을 하였다.

표 1 에서 FM 알고리즘, CLIP 알고리즘과 수행 결과와 비교하였다. 실험을 통해 본 논문이 제안하는 분할 알고리즘은 FM 알고리즘에 비해서 최소 33%에서 최대 65%의 성능 향상을 보였고, CLIP 방법에 비해서는 최소 31%에서 최대 51% 성능 개선을 보였다.

표 4 - 1 기존 분할 알고리즘과 성능 비교

Circuit	Cut Size			Improvement	
	FM	CLIP		FM	CLIP
alu2	98	86	54	44.5	36.7
alu4	154	138	79	48.9	43.0
misex3	131	127	72	45.0	43.3
duke4	114	99	50	55.9	49.2
C1355	46	41	22	52.5	46.7
C1908	173	140	80	54.0	43.1
C2670	102	86	47	54.0	45.5
C3540	188	133	65	65.5	51.3
C5315	142	109	55	61.6	50.0
C6288	170	155	106	37.9	31.9
C7552	146	131	97	33.6	26.0
rot	64	50	33	48.4	34.0
x3	25	21	13	47.2	37.1
Average Improvement				50.0	41.0

#### V. 결론

본 논문은 1차적으로 그래프 정보를 이용한 분할 알고리즘을 통해서 초기 분할을 수행한 후, 분할된 절단 집합에 있는 신호선을 대상으로 회로 정보를 이용하여 2차 분할을 수행하는 방법을 제안하였다.

제안한 분할 알고리즘은 1차적으로 클러스터 제거 효과가 있는 노드는 이득 변화량에 의해서 노드를 선택하고, 클러스터 제거 효과를 상실한 노드에 관해서는 총 이득을 통해 노드를 선택하는 가변적인 방법을 적용하였으며, 2차적으로는 절단집합상의 신호선을 대상으로 타겟 신호선을 결정하고, 이것을 제거하기 위한 대상 신호선(Alternative wire)를 회로에 추가해서 대상 신호선을 제거하여 기능적으로 동일한 회로로 변형하는 방법을 사용하였다. 실험을 통해서 제안한 분할 알고리즘은 기존의 알고리즘의 비해 41 - 50%의 성능 향상을 가져온 것을 볼 수 있었다.

#### References

- [1] C. M. Fiduccia and R. M. Mattheyses. "A linear-time heuristic for improving network partitions," in Proc. ACM/IEEE Design Automation Conf., 1982. pp.175-181.
- [2] S. Dutt and W. Deng, "VLSI circuit partitioning by cluster-removal using iterative improvement techniques," in Proc. IEEE Int. Conf. CAD, 1996, pp. 441-446.
- [3] S. C. Chang, K. T. Cheng, N. S. Woo, and M. Marek Sadowska, "Layout Driven Logic Synthesis for FPGAs," DAC-94.
- [4] K. T. Cheng and L. A. Entrena, "Multi-level Logic Optimization by Redundancy addition and Removal," ECDA-93
- [5] Luis A. Entrena and K. T. Cheng, "Sequential Logic Optimization by Redundancy Addition and Removal," Proc. International Conference on CAD 1993