

최대컷값을 최소화하는 k-way 분할 연구

김 경식*, 이 철동*, 유 영욱*, 진 주식**, 황 회음**
 *한국전자통신 연구소, **서울대학교

A Study on the K-way Partition Minimizing Maxcut

Kyung Sik Kim*, Chul Dong Lee*, Young Uk Yu*, Chu Shik Jhon**, and Hee Yung Hwang**

*ETRI, **Seoul National Univ.

<Abstract>

In this paper, we present a new k -way partitioning algorithm for a graph of an electrical circuit wherein nodes and edges are regarded as cells (modules) and nets, respectively. In contrast to the previous work, our method is based upon a linearly ordered partition paradigm. We also claim that the maximum number of netcuts mostly governs the performance of k -way partitioning, thus having influence on the construction of a new cost function. In addition, our approach elaborates upon balancing the partition size. Our experiments show excellent results in comparison with previous k -way partitioning algorithms.

1. 서론

표준 셀(standard cell)등을 이용하여 집적회로(IC)의 레이아웃(layout)을 설계할 때, 전체 레이아웃의 면적을 작게하고 배선시 각 네트(net)의 배선장(wire length)을 작게 하기 위해서는 셀들의 배치를 적절히 해야 한다. 셀들의 적절한 배치를 위해, 분할(partition)에 근거한 방식이 많이 사용되고 있는데, 이는 회로의 상위 레벨 정보인 셀의 연결 정보를 배치에 이용하고 있어 효율적인 결과를 기대할 수 있기 때문이다[1, 2].

본 논문에서 말하는 k -way 분할은, 셀들을 그 연결정보로부터 k 개의 집합으로 분할할 때 k 개의 집합을 일렬로 위치시키며, 최대컷값(maxcut)이 최소가 되도록 한다. 이 개념은 Breuer의 Min-max 컷값 목적함수[4]에 기초한 것으로서, 표준 셀의 레이아웃 구조에 적용시키기에 적합하다. 예로서, 그림 1의 표준 셀 레이아웃 구조를 보면, 이층(two layer)배선이 사용될 때, 사용된 수평배선 트랙(track)의 갯수가 채널(channel)의 폭(width)을 결정하며, 전체 칩(chip)의 면적을 좌우한다. 이때 채널폭의 하한치(lower bound)는 최대컷값이다.[4]

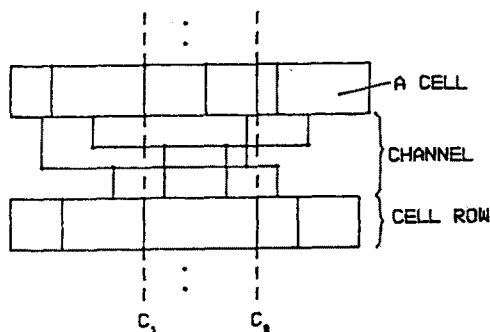


그림 1. 표준 셀 레이아웃 구조 예

따라서 각 셀열(cell row)내에서 셀들을 위치시킬 때 k -way 분할을 이용하여 최대 컷 값을 줄이는 것이 채널의 폭을 줄이게 된다. 이러한 k -way 분할을 위한 기존의 방법들은 대개 2-way 분할을 이용하여 k 개까지의 집합을 만들어 가고 있으나 [3, 4, 5], 원칙적으로 2-way 분할이란 만들어지는 집합의 내부 연결 네트수를 최소화 하고 있어, 1차 2-way 분할로 만들어진 집합을 2차로 2-way 분할할 때 최대컷값이 오히려 나쁘게 나올 수 있다. 그림 2에 이러한 예를 보인다.

본 논문에서 제시하는 k -way 분할 알고리즘은 그림2(b)에서 예를 보듯이 최대 컷 값을 최소화하는 데 목적이 있다. 그 방식은 각 집합에 속한 셀들이 다른 ($k-1$)개의 집합으로 이동될 때의 컷값의 줄어드는 크기를 계산하여 그 값을 비용함수 (cost function)로 잡는다.

k 값의 결정은 k -way 분할을 이용하는 배치 과정에서 결정하기로 하며, 본 논문에서는 k 값을 입력으로 받아 수행한다. 2절에서 k -way 분할에 대한 역사적 고찰로서 기존 알고리즘들을 소개하고 문제점을 지적하며 3절에서 본 논문에서 제시하는 방법의 알고리즘을 기술한다. 4절에서는 제시한 알고리즘에 의한 분할 수행 결과를 보이고 기존 방법에 의한 결과와 비교한다.

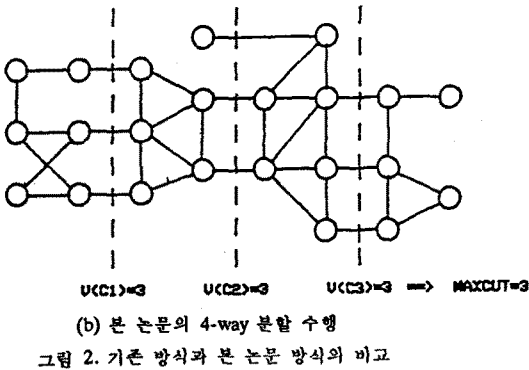
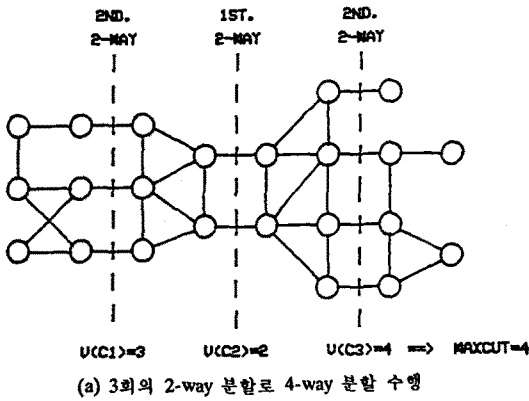


그림 2. 기존 방식과 본 논문 방식의 비교

2. 역사적 고찰

2-way 분할에 대해서는 현재 최적에 가까운 분할 결과를 내는 알고리즘들이 이미 개발되어 Min cut 배치 알고리즘 등에 활용되고 있으나 [3, 4, 6, 7], k-way 분할 (k>2)에 대해서는 연구가 별로 발전된 바가 없었다. 본 장에서는 기존 k-way 분할 알고리즘들을 설명하고 그 문제점들을 기술한다.

(1) 2-way 분할 반복 이용 방식 [3]

기본적인 개념은 주어진 집합을 크기가 n 인 k 개의 부집합으로 분할하는 것으로서, 부집합의 쌍들에 대해 2-way 분할방식을 적용하여, 두 부집합간의 최적화를 이루어 갑으로써 전체적인 최적화를 기하는 방식이다. 문제점은 첫째, 매 2-way 최적화가 이루어질 때 마다 다른 부집합들에 대한 최적 상황이 변하여 kC₂ 보다 경우의 수가 많아진다. 둘째, 세 개 이상의 집합에서 세 개 이상의 셀들이 옮겨지는 것이 전체적인 최적화를 이룰 때 이의 처리가 곤란하다.

(2) Cluster 성장 방식 [8]

K개의 연결도가 서로 제일 낮은 셀들을 뽑아 k 집합의 씨(seed)로 선정한 후 각 집합에, 기 배치된 셀들에 연결도가 가장 높은 셀을 포함시켜 집합을 성장시켜 나가며 k개 집합을 완성한 후, 인접하지 않은 집합간의 연결선을 최소화하도록 k개 집합을 일렬로 위치시킨 다음, 인접한 두

집합을 합쳤다가 2-way 분할을 수행(shuffling이라 함)하여 최적화를 기하는 방식이다. 문제점은 최대컷값의 최소화를 보장하지는 못한다는 점이다.

(3) Quadratic 프로그래밍 방식 [9]

그래프 분할을 quadratic 프로그래밍 문제로 변환하여 연결 행렬(connectivity matrix)과 집합수 k 를 입력으로 받아, 내부 연결선 수의 최대화를 기한다. 문제점으로는 여러 셀에 연결된 신호선(multi-terminal net)의 처리가 연결 행렬에서 적절히 표현되기 어렵다는 점이다.

3. K-way 분할 알고리즘

(1) K-way 분할 모델

그림 3에서 k-way 분할 모델을 보인다. 셀(모듈)과 네트가 각각 노드와 선분으로 표시된 전자회로의 그래프에 있어서, 셀들 k개의 집합 S_i, 1 ≤ i ≤ k 로 분할할 때, 컷선 C_j는 인접한 두 집합 S_j와 S_{j+1} 사이에 위치하며, 컷값 V(C_j)는 컷선 C_j에 의해 잘리는 네트의 갯수를 의미한다.

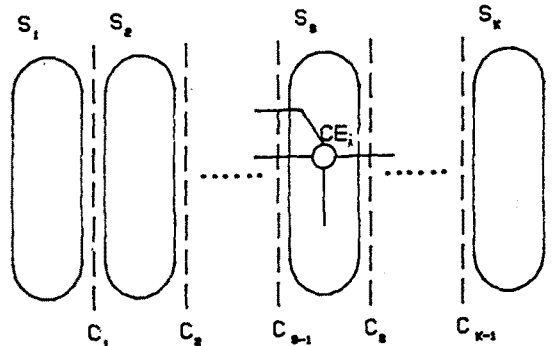


그림 3. k-way 분할 모델

(2) 비용 함수

최대컷값은 본 k-way 분할 알고리즘의 성능을 결정하는 주요한 요인으로서, 다음과 같은 비용함수를 정의하여 최대컷값을 줄인다.

임의의 한 셀을 CE_i라 할 때, cost(CE_i, S_j, C_m)를 셀 CE_i가 집합 S_j로 이동될 때 컷선 C_m에서 줄어드는 컷값을 표시하는 함수라 정의하면,

$$\text{cost}(CE_i, S_j, C_m) = \text{이동후 } V(C_m) - \text{이동전 } V(C_m)$$

따라서 셀 CE_i를 집합 S_j로 옮길 때의 총 줄어드는 컷값과 새로운 최대컷값을 cutsum(CE_i, S_j), maxcut(CE_i, S_j)로 각각 정의할 때,

$$\text{cutsum}(CE_i, S_j) = \sum_{m=1}^{k-1} \text{cost}(CE_i, S_j, C_m)$$

$$\text{maxcut}(CE_i, S_j) = \max\{V(C_m) + \text{cost}(CE_i, S_j, C_m), 1 \leq m \leq k-1\}$$

가 된다. 한편 셀 CE_i 의 이동할 대상 집합 S_j 는 집합의 크기 제한을 벗어나지 않는 범위내에서 $\maxcut(CE_i, S_j)$ 가 최소인 곳이 되며, 만일 같을 때는 $\text{cutsum}(CE_i, S_j)$ 이 최소인 곳이 된다.

(3) 집합의 크기 균형

집합의 크기는 포함된 셀들의 쪽의 합으로 정의한다. 전체 셀들의 쪽의 합을 W 라 볼 때, 각 집합의 크기가 W/k 가 되도록 하는 것이 균형의 목적이거나, 셀 쪽의 크기차로 인해 실제로 이것을 정확히 만족하기는 어렵다. 그러나 셀 쪽의 변화가 크지 않다는 가정하에서 본 논문에서는 다음과 같은 조건이면, "집합 S_j 가 균형 상태에 있다" 라고 간주하였다.

$$| \text{size}(S_j) - W/k | \leq \text{최대크기 셀 쪽}$$

(4) 입출력

본 논문에서 제안하는 k-way 분할 알고리즘의 입력으로, (가)논리회로로부터 추출되는 사용된 셀의 연결도에 대한 정보와 (나)각 셀의 성질(property) 즉, 셀의 쪽, 핀의 위치등의 정보를 보유하고 있는 셀 라이브러리(cell library)와 (다)분할 갯수 k 를 받는다. 이 때 k 값은 k-way 분할을 이용하는 배치와 같은 상위 과정에서 결정하는 것으로 가정한다. 출력으로, (가)분할 결과로서, k 집합 속의 셀 번호와 (나)각 컷선의 컷값과 (다)최대컷값을 인쇄한다.

(5) 알고리즘

- 단계 1: 입력 - 셀연결정보, 셀라이브러리, k
- 단계 2: 초기 분할 - k 집합 크기 균형 유지
- 단계 3: 각 셀별 $\text{cutsum}()$, $\text{maxcut}()$ 계산 후, 집합별로 $\text{maxcut}()$, $\text{cutsum}()$ 을 최소로 하는 순서로 (셀, 대상집합)의 배열 구성
- 단계 4: 크기제한 이내이고, $\text{maxcut}()$, $\text{cutsum}()$ 을 최소로 하는 (CE_i, S_j) 의 선택. 없으면 단계 8로 이동
- 단계 5: 셀 CE_i 를 S_j 로 옮김. 큐(queue)에 (CE_i, S_j) 보관
- 단계 6: 셀 CE_i 에 연결된 들의 $\text{maxcut}()$, $\text{cutsum}()$ 수정 및 (셀, 대상집합) 배열 수정
- 단계 7: 단계 4로 이동
- 단계 8: 큐를 이용하여 $\text{maxcut}()$ 을 최소로 했던 셀 이동 상태로 복원
- 단계 9: $\text{maxcut}()$ 이 줄어든 바 있으면 단계 3으로 이동
- 단계 10: 정한 횟수 이내의 perturbation 수행 (임의로 몇개의 셀을 강제로 이동시킴) 단계 3으로 이동
- 단계 11: perturbation 결과들 중에서 최소 최대컷값의 결과 인쇄
- 단계 12: 종료

4. 실험결과

표 1에 기존 k-way 분할 알고리즘들에 의한 최대컷값과 본 알고리즘에 의한 최대컷값을 비교하여 보인다.

표 1. 네가지 k-way 분할 알고리즘의 실험 결과

회로	셀갯수	네트수	k-way	최대컷값				참고문헌
				M1	M2	M3	M4	
1	20	55	4	15	15	15	14	[10]
2	23	36	4	4	3	4	3	그림 2
3	15	46	3	17	18	18	17	[9]

M1 : 2-way 분할 반복 이용 방식

M2 : Cluster 성장 방식

M3 : Quadratic 프로그래밍 방식

M4 : 본 알고리즘

표 1 에서 보이듯이 본 알고리즘이 최소의 최대컷값을 내는 분할을 수행하고 있다. 그림 4에 Chen[9]의 논문에서 3-way 분할을 실행한 결과를 보이며 (표 1의 회로 3 참조), 그림 5에 같은 회로를 본 알고리즘에 의해 분할한 결과를 보인다.

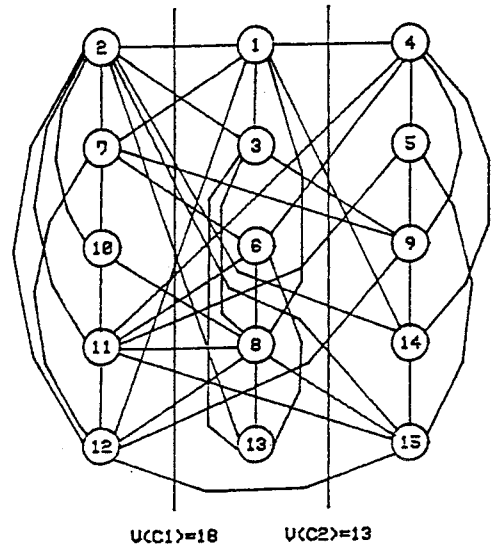


그림 4. Chen의 3-way 분할 결과

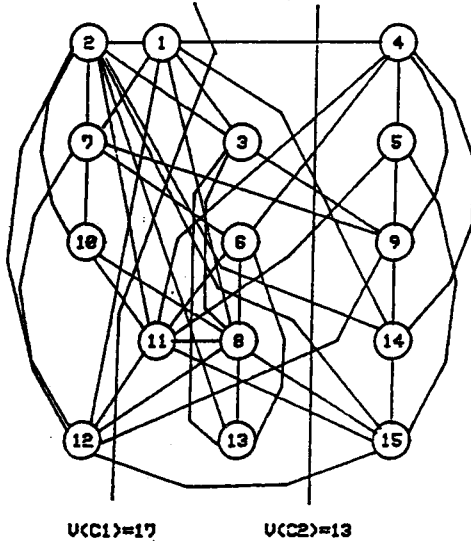


그림 5. 본 알고리즘의 3-way 분할 결과

5. 결론

본 논문에서는 Breuer 의 Min-cut 배치 논문 [4] 에서 open problem 으로 제시되었던 $\min[\text{Max}\{V(C)\}]$ 의 최소화 문제를, k-way 분할 모델을 제안하여 해결하는 한 알고리즘을 제시한다. 제시한 알고리즘의 실험결과는 기존 k-way 분할 알고리즘들 보다 작은 최대컷값을 냈다. 최대컷값의 최소화는 표준 셀의 레이아웃 구조에 유용할 것이다. 제시한 k-way 분할 알고리즘을 도입한 표준 셀 배치 시스템이 계속 연구 개발될 것이다.

참고 문헌

- [1] U.Lauther, "An Overview of Placement and Routing Techniques," Proc. VLSI & Computer, CompEuro 87, pp.615-620, Hamburg, May 1987.
- [2] B.T.Preas and P.G.Karger, "Automatic Placement: A Review of Current Techniques," Proc. 23rd. DAC., pp.622-629, 1986.
- [3] B.W.Kernighan and S.Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," BSTJ., pp.291-307, Feb. 1970.
- [4] M.A.Breuer, "Min-Cut Placement," J. Design Automation and Fault Tolerant Computing, vol.1, no.4, pp.343-362, Oct. 1977.
- [5] 홍성수, 선형 계산 시간 분할 알고리즘에 의한 표준 셀 배치에 대한 연구, 서울대학교 전산기공학과, 석사학위논문, 1988. 2.
- [6] C.M.Fiduccia and R.M.Matteyses, "A Linear-time Heuristic for Improving Network Partitions," Proc. 19th. DAC., pp.175-181, 1982.
- [7] A.E.Dunlop and D.W.Kernighan, "A Procedure for Placement of Standard Cell VLSI Circuits," IEEE Trans. CAD, vol.CAD-4, no.1, pp.92-98, Jan. 1985.
- [8] T.Kambe, et al., "A Placement Algorithm for Polycell LSI and Its Evaluation," Proc. 19th. DAC., pp.655-662, 1982.
- [9] C.C.Chen, "Placement and Partitioning Methods for Integrated Circuits Layout," Ph.D. Dissertation, UC Berkeley, EECS Dept., 1985.
- [10] E.R.Barnes, "An Algorithm for Partitioning the Nodes of a Graph," SIAM j. Alg. Disc. Meth., pp.541-550, Dec. 1982.