

# 효율적인 초기 배치를 이용한 개선된 Mongrel

성영태<sup>o</sup> 허성우

동아대학교 컴퓨터 공학과

saint<sup>o</sup>@donga.ac.kr, swhur@daunet.donga.ac.kr

## Improved Mongrel with Efficient Initial Placements

YoungTae Sung<sup>o</sup> SungWoo Hur

Dept. of Computer Engineering, Dong-A University

### 요약

본 논문에서는 표준 셀 배치기 Mongrel을 소개하고 두 가지 분할 기법(FM, hMETIS)을 통해 효율적인 초기 배치를 얻음으로써 Mongrel의 성능을 개선한다. Mongrel은 middle-down 방법론을 채택한 전역 배치와 상세 배치의 2단계 표준 셀 배치기이며 전역 배치 단계에서는 RBL(Relaxation-Based Local Search)를, 그리고 상세 단계에서는 최적 인터리빙(optimal interleaving) 기법을 통해 최적화 알고리즘을 수행한다. MCNC 벤치마크 회로를 이용한 실험을 통해 Mongrel의 성능을 평가하고 효율적인 초기 배치가 최종 결과에 미치는 영향을 분석한다.

### 1. 서론

셀 배치 문제는 VLSI 물리적 설계 단계에서 해결해야 할 중요한 문제 중 하나로서 칩 면적, 라우팅 그리고 칩 성능에 결정적인 영향을 미친다. 표준 셀의 경우 셀 배치 문제는 배선 길이를 줄이는 것을 주로 목적으로 하며 다음과 같이 기술된다. 고정 높이를 가진 셀들의 넷 리스트와 행(row)에 대하여 각 셀을 행에 배치하고 각 행 내에서 셀의 중첩 없이 배선 길이가 최소가 되는 셀의  $x$  좌표를 찾는 문제이다. 또한 배선 길이의 측정은 널리 사용되는 HP(Half-Perimeter) 측정법을 통해 계산한다.

Hur와 Lillis는 표준 셀 배치 문제를 해결하기 위하여 다양한 기법을 혼합적으로 도입한 Mongrel[1]을 제안하였다. Mongrel은 middle-down 방법론을 채택하고 있으며 크게 전역 배치와 상세 배치 두 단계로 배치 알고리즘을 수행한다. 전역 배치 단계는 거시적인 관점에서 레이아웃 내에 가상의  $n \times m$  격자를 생성하고 각 셀을 격자의 bin(bin)에 배치하는 것을 목적으로 한다. 상세 배치 단계에서 최적 인터리빙 기법을 통해 각 셀의 정확한 위치가 결정된다.

[1]에서는 초기배치가 최종배치의 질에 어떤 영향을 미치는지 언급하고 있지 않으나 실험에 의하면 Mongrel에 의해 생성된 최종 배치는 초기 배치의 질에 영향을 받는 것을 알 수 있었으며, 결국 좋은 초기 배치를 효율적으로 얻을 수 있다면 전역 배치의 질이 개선되고 나아가 최종배치의 결과가 좋아진다고 볼 수 있다.

본 논문에서는 초기 배치가 최종 배치에 미치는 영향에 대해서 연구하고, 초기 배치를 효율적으로 얻음으로써 Mongrel을 개선시킬 수 있음을 보였다.

초기 배치를 얻는 과정은 주로 분할 기법을 이용하여 셀들을  $n \times m$  격자에 배치하는 과정으로서 FM(Fiducia-Mattheyses)류의 반복 개선 분할 기법을 사용하거나 클러스터링 기법을 혼합한 형태의 분할 기법, 혹은 순서화 알고리즘[2] 등을 통해 수행될 수 있다. 최근 다단계 패러다임에 기반을 둔 클러스터링을 이용한

하이퍼 그래프 분할 기법(hMETIS)[3]이 소개 되었는데 이는 전통적인 반복 개선 알고리즘의 단점을 보완 하였다. 본 논문에서는 FM 기법과 hMetis 기법을 이용한 top-down 분할방식으로 초기 배치를 얻도록 Mongrel을 개선하였다.

본 논문의 구성은 다음과 같다. 2장에서는 Mongrel에 사용된 여러 가지 배치 기법을 살펴보고 3장에서는 초기 배치 기법을 설명한다. 4장에서는 표준 벤치마크 회로를 이용한 실험을 통하여 초기 배치가 칩 설계에 미치는 영향을 알아보고 Mongrel의 성능을 평가한다.

### 2. Mongrel의 개요

Mongrel의 전체적인 최적화 과정을 요약하면 다음과 같다. 먼저 효율적인 전역 배치를 위해 효과적으로 초기 배치를 얻는다. 이 과정은 주로 하향식 분할 기법을 통해 이루어진다. 초기 배치가 구성 되고나면 RBL(Relaxation-Based Local Search) 기법을 통해 전역 배치를 개선 시킨다. 전역 배치는 상세 배치로 변환되고 최적 인터리빙 알고리즘을 통해 상세 배치를 개선시킨다.

#### 2.1 전역 배치

RBL는 [4]에 처음 소개 되었으며, 선형 배치 문제를 해결하는 데 처음 적용되었다. Mongrel은 RBL를 각 행 단위의 배치 문제에 적용하였으며 전통적인 국부 탐색(local search) 구조를 가진다. 즉 현재 해 공간에서 다음 해를 찾아내고 목적 함수가 개선되었다면 탐색된 해 공간으로 이동한다. RBL의 핵심은 이웃 해를 어떻게 추출해 내는가이다. 이것은 독특한 분석적 엔진을 통해 수행되는데 단순한 셀 교환 방식의 이동이 아닌 거시적 관점에서 셀의 위치를 결정할 수 있는 기법이다. 이 과정을 요약하면 다음과 같다.

##### a. 부분 회로 추출

주어진 인수  $m$ 에 대하여,  $|M|=m$  이 되도록 부분 회로의 집

합 M을 추출한다. 부분 회로의 집합 M은 모바일 노드라 부르며 이것으로부터 고정 노드의 집합 F를 결정한다. 집합 F의 요소는 일의의 넷을 통해 모바일 노드와 직접 연결된 노드들이다  $F = \{v \in (V - M) \mid \exists e \in E, s.t. v \in e\}$ . 따라서 부분 회로는 모바일 노드의 집합 M과 고정 노드의 집합 F, 그리고 모바일 노드에 영향을 받는 넷들의 집합  $E' = \{e \mid e \cap M \neq \emptyset\}$ 으로 구성되며, 이 넷들을 활성넷이라 부른다.

b. 완화된 제약조건 하에서의 최적 배치

모바일 노드의 집합 M과 고정 노드의 집합 F, 그리고 활성 넷 E'이 결정되었다면 모바일 노드의 x/y 좌표 결정을 위해 각각 x와 y에 대한 하부 문제로 최적 배치문제를 분리한다. 완화된 조건 하에서 각 모바일 노드에 대한 최적의 x/y 좌표를 결정하기 위해서 x/y축으로 부분 회로를 사상시킨다. 사상된 부분 회로에서 모바일 노드에 대한 최적의 위치를 결정하기 위해서 x/y축에 대한 두 개의 서로 독립적인 선형 프로그램으로 분리시킨다. 선형 프로그램은 [4]에 보인 바와 같이 네트워크 플로우 기법을 통해 효율적으로 해결한다.

c. 적법화

RBLs를 통해 개선된 전역 배치는 셀의 중첩이나 빈과 행에 대한 크기 제약 조건을 만족하지 않을 수 있는데 이것을 해결하기 위하여 적법화 과정을 거친다. 이 과정에서 ripple-move를 통해 셀 밀집도가 높은 빈(bin)으로부터 셀들을 이동시키는데 이 때 목적함수인 배선길이를 항상 고려하기 때문에 적법화 시인 후에도 최종 배선길이는 크게 증가되지 않는 특징이 있다.

2.2 상세 배치

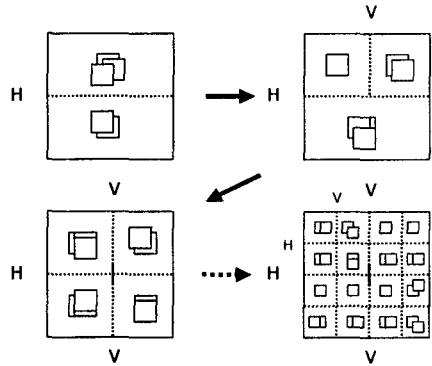
상세 배치 단계에서 전역 배치는 상세 배치로 변환되는데 이때 각 빈 내의 셀에 대한 상대적 위치가 결정된다. 각 행에 대하여 셀의 상대적 위치는 탐색 알고리즘에 기초해서 각 셀에 영향을 미치는 force-value를 사용하여 결정한다. 일단 초기 상세 배치가 구성되면 최적 인터리빙 기법을 통해 각 행(row) 단위로 최적화를 수행한다. 최적 인터리빙의 세부 단계는 다음과 같다.

- 주어진 윈도우 크기 W에 대하여, 윈도우 W에 있는 부분 열 A를 추출하고 A 내에 있는 셀들의 상대적 위치를 유지한다.
- 부분열 A에 속하지 않은 나머지 부분열을 부분열 B라 하고 마찬가지로 셀들의 상대적 위치를 유지한다.
- 부분열 A와 B의 최적 정렬을 찾도록 인터리브 시킨다.
- 가장 위쪽 행부터 마지막 행까지 윈도우를 왼쪽에서 오른쪽으로 이동시키면서 위 과정을 반복 적용한다.

두 부분열의 최적 조합 과정은 동적 프로그래밍 기법을 통해 효율적으로 해결될 수 있다. 주어진 윈도우 크기를  $n+m$ 이라 할 때,  $A = a_1, a_2, \dots, a_n$  그리고  $B = b_1, b_2, \dots, b_m$ 라고 하고,  $S_{ij}$ 를  $a_1, a_2, \dots, a_i (i \leq n)$ 와  $b_1, b_2, \dots, b_j (j \leq m)$ 로부터 얻은 최적 순서라 하고,  $C(S_{ij})$ 는  $S_{ij}$ 의 비용이라 한다면 부분열 A와 B를 인터리빙하는 것은 최소 비용의  $S_{n,m}$ 을 찾는 것과 같다. 동적 프로그래밍 기법은 테이블을 이용함으로써 이전까지의 국부 최적  $S_{ij}$ 를 항상 저장하고 있으며 이러한 값들은 각 단계마다 독립적인 최소값이다. 따라서 최적의  $S_{n,m}$ 은 각 표의 최소  $S_{ij}$ 의 조합으로 얻을 수 있다.

3. 초기 배치

초기 배치를 얻는 방법은 top-down 방식의 분할 기법을 사용하여 하향식으로 분할해 가면서  $n \times m$  크기의 격자를 형성하고 배치를 얻는 방법이다. 배치 과정은 각각의 분할 단계마다 핵심 영역의 상대적 비율에 따라 가로 혹은 세로로 분할하고  $n \times m$  격자가 생성 될때 까지 반복한다. 또한 분할 단계마다 생성된 정보는 슬라이싱 트리에 저장되며 최종 분할 후 각 셀의 위치는 해당 빈의 중심 좌표에 할당된다. 분할 기법으로는 크게 FM 분할 기법과 hMETIS 기법을 사용하였으며 초기 배치 과정을 [그림 1]에 나타내었다.



[그림 1] 초기 배치 과정

FM과 hMetis 분할 기법에 대해 간략히 설명하면 다음과 같다.

3.1 FM 분할 기법

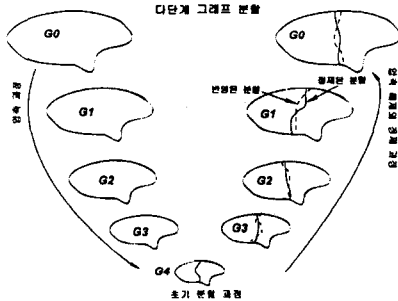
FM 알고리즘 [5]의 기본 전략은 이득값이 가장 높은 셀을 한번에 하나씩 옮기는 것이다. 먼저 모든 셀의 이득값을 계산하고 초기에 모든 셀을 이동 가능한 "free cell"로 설정한다. 그 다음 이득값이 가장 큰 셀을 선택해서 옮기고 다음 단계에서 다시 선택되는 일이 없도록 "locked cell"로 설정한다. 일단 옮겨진 셀에 대하여 이웃한 모든 셀의 이득값을 재조정한다. 이때 두 부분(partition) 사이의 컷 수를 저장하고 다시 위 과정을 반복한다. 이 과정은 모든 셀이 "locked cell"이 될 때 까지 반복한다. 여기서 각 셀이 옮겨질 때 마다 저장된 컷 수의 부분합 중 최대가 되는 셀 까지 분할에서 실제로 이동시키고 이것을 다음 단계의 입력으로 한다.

FM 알고리즘은 효율적인 이득버킷 자료구조와 이웃 셀 이득값 재조정 메커니즘을 통해 크기가 작은 회로의 경우 좋은 결과물 예상 할 수 있다. 하지만 전통적으로 지적되어 온 몇 가지 문제들 때문에 다양한 방법으로 확장 개선되었다 [6].

3.2 다단계 하이퍼 그래프 분할 기법(hMETIS)

클러스터링을 이용한 다단계 하이퍼 그래프 분할 기법(hMETIS)는 다단계 패러다임에 기반한 분할 기법으로서 다음과 같은 단계로 수행된다. 먼저 기존 그래프에서 연속적으로 압축 그래프를 생성한다. 이 과정은 그래프를 더 이상 압축 할 수 없을 때 까지 진행되며 가장 작은 압축 그래프가 생성되었을 때 압축 그래프에 대한 이분할(bisection)이 수행된다. 이분할 과정은 기본적으로 FM(Fiduccia-Mattheyses) 알고리즘이 사용되며 다음

단계, 즉 압축 해제와 정제 과정의 입력으로 주어진다. 압축 해제와 정제 과정은 기존의 그래프로 복원될 때 까지 진행된다. 이 과정을 [그림 2]에 나타내었다.



[그림 2] 다단계 하이퍼 그래프 분할 과정

기본적으로 압축 단계에 사용되는 압축 알고리즘으로는 한 쌍의 노드 단위로 압축을 수행하는 edge-clustering과 하이퍼 넷 단위로 압축하는 hyperedge-clustering이 있으며 정제 과정에는 FM 알고리즘과 하이퍼 넷 정제 알고리즘이 사용된다.

4. 실험 및 고찰

초기 배치가 최종 배치에 미치는 영향을 알아보기 위해 표준 벤치마크 회로를 사용하였고 FM 분할 기법과 hMETIS 분할 기법 두 가지에 대하여 비교하였다. 실험에 사용된 회로는 GSRC Bookshelf 형식을 따르며 펜티엄3-700Mhz(prim1, struct, biomed), 펜티엄4-2Ghz(ind2, ind3, avqs, avql) 리눅스 머신 상에서 각각 실험하였다. [표 1]에 회로 사양을 나타내었고 [표 2]와 [표 3]에 배선 길이와 성능을 각각 나타내었다. 배선 길이는 각각의 초기 배치를 이용하여 Mongrel이 최적화시킨 것으로서 각 회로에 대해 10회에 걸쳐 구한 결과 중 가장 좋은 것을 보였다. 참고로, 10개의 결과 값은 크게 차이가 나지 않았다.

[표 1] 회로사양

	prim1	struct	biomed	ind2	ind3	avqs	avql
#node	833	1952	6514	12637	15433	21918	25178
#cell	752	1888	6417	12142	15059	21854	25114
#pad	81	64	97	495	374	64	64
#row	17	21	44	69	52	79	83

[표 2] 배선 길이 비교 (단위 : micron)

	FM 기법	hMETIS 기법	개선율(%)
prim1	970077	986742	-1.6
struct	736894	712874	3.2
biomed	3484765	3348642	3.9
ind2	16280861	15094551	7.2
ind3	44828842	44107518	1.6
avqs	5832138	5398225	7.4
avql	6458739	5776154	10.5

시간 측정을 통한 성능 비교에서 FM을 이용한 초기 배치보다 hMETIS를 이용한 초기 배치가 월등히 우수함을 알 수 있는데 이것은 초기 배치를 얻는데 걸리는 시간을 줄이고 좋은 초기 배치를 얻음으로써 전역 배치와 상세 배치를 개선한 결과로

볼 수 있다.

[표 3] 성능 비교 (단위 : sec)

	FM 기법	hMETIS 기법	개선율(%)
prim1	8.94	4.09	54
struct	23.65	18.56	21
biomed	379.78	57.59	84
ind2	722.63	225.08	68
ind3	1045.05	473.08	54
avqs	1407.78	431.60	69
avql	4197.15	2739.35	35

5. 결론 및 향후 과제

본 논문에서는 하향식 분할 기법을 이용하여 효율적으로 초기 배치를 얻고 전역 배치의 성능을 높임으로써 Mongrel의 전체적인 성능을 개선하였다. 표에서 보이는 바와 같이 클러스터링을 이용한 다단계 하이퍼 그래프 분할 기법(hMETIS)이 FM을 이용한 방법 보다 우수한 최종 배치를 가져왔다.

[표 2]에서 보이는 바와 같이 배선 길이에서 클러스터링을 이용한 다단계 분할 기법을 통해 평균 4.7%의 개선율을 얻었으며 [표 3]에 나타난 결과를 통해 Mongrel의 성능이 평균 55% 정도 향상되었음을 알 수 있다. 이것은 효율적인 초기 배치가 최종 결과에 결정적인 영향을 미치는 중요한 과정이며 초기 배치를 개선함으로써 Mongrel의 전체 성능을 개선할 수 있음을 나타낸다.

클러스터링을 이용한 분할 기법은 초기 배치를 얻는 과정 뿐만 아니라 배치 과정의 전반적인 단계에서 사용될 수 있는 기법이다. 따라서 전역 배치 단계에서 수행되는 RBLs와 클러스터링 기법을 적절히 조합할 경우 전역 배치의 성능을 개선할 수 있을 것으로 예상된다.

5. 참고문헌

- [1] Sung-Woo Hur and John Lillis, "Mongrel: Hybrid Techniques for Standard Cell Placement," *Proc. of ICCAD*, pp.165-170, 2000.
- [2] C. J. Alpert and A. B. Kahng, "A General Framework for Vertex Orderings, With Applications to Circuit Clustering," in *Proc. of IEEE Trans. on CAD*, pp. 63-67, 1994.
- [3] G. Karypis, R. Attarwal, V. Kumar and S. Shekhar, "Multilevel Hypergraph Partitioning: Application in VLSI Domain," in *Proc. of Design Automation Conference*, pp. 526-529, 1997.
- [4] Sung-Woo Hur and John Lillis, "Relaxation and Clustering in a Local Search Framework: Application to Linear Placement," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 360-366, 1999.
- [5] C. M. Fiducia and R. M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions," in *Proc. of IEEE Design Automation Conference*, pp. 175-181, 1982.
- [6] C. J. Alpert and A. B. Kahng, "Recent Directions in Netlist Partitioning," *Integration, the VLSI Journal*, 19(1-2):1-81, 1995.