

論文98-35C-10-8

가변적인 연결도 임계치 설정에 의한 대규모 집적회로 설계에서의 안정적인 다단 분할 방법

(A Stable Multilevel Partitioning Algorithm for VLSI Circuit Designs Using Adaptive Connectivity Threshold)

任昌慶*, 鄭正和*

(C.K. Eem and J. Chong)

요 약

본 논문에서는 대규모 집적회로 설계에 있어 효율적이고 안정된 분할을 위한 새로운 다단 분할 방법을 제안한다. 대규모 회로의 설계에 반복적인 분할 개선 방법을 적용함에 있어 성능의 한계를 극복하기 위해 제안된 다단 분할 방법은 분할 계층구조의 형성 방식에 의해 그 성능이 결정되었다. 기존에 제안된 대부분의 다단 분할 방법은 계층구조를 형성하는 과정에서 실험에 의한 인위적인 제한 조건을 설정하여 분할 결과의 안정성이 저하되는 문제가 있었다. 이러한 안정성의 결여는 반복 수행시의 분할 결과 편차가 매우 커지는 상황을 초래한다. 본 논문에서는 이러한 인위적인 제한 조건의 설정을 최소화하고 계층구조 형성 과정에서 현재 회로 연결 상태를 고려하여 자율적인 제한조건에 의해 클러스터링을 수행하는 새로운 계층구조 형성 방식을 제안한다. 제안된 방법에 의해 형성된 분할 계층구조는 HYIP^[11]의 하이브리드 버킷을 이용한 분할 개선 방법을 반복적으로 적용하여 분할 결과를 얻는다. 본 다단 분할 방법은 ACM/SIGDA에서 제공한 벤치마크 회로를 대상으로 실험한 결과 기존 분할 방식^{[3] [4] [5] [8] [9]}에 비해 약 10-40% 가량의 최소 cutsize 감소 효과가 있었고 기존의 다단 분할 방법 중에 가장 효율적인 방법으로 평가되는 ML^[10]에 비해 제안된 방법이 최소 cutsize에 있어서는 약 5%, 평균 cutsize에 있어서는 평균 20%이상의 성능 향상을 가져 왔다. 더욱이 제안된 방법을 10회 수행한 결과가 ML 방법을 100회 수행한 결과 보다 앞서는 성능을 보였다.

Abstract

This paper presents a new efficient and stable multilevel partitioning algorithm for VLSI circuit design. The performance of multilevel partitioning algorithms that are proposed to enhance the performance of previous iterative-improvement partitioning algorithms for large scale circuits, depend on choice of construction methods for partition hierarchy. As the most of previous multilevel partitioning algorithms forces experimental constraints on the process of hierarchy construction, the stability of their performances goes down. The lack of stability causes the large variation of partition results during multiple runs. In this paper, we minimize the use of experimental constraints and propose a new method for constructing partition hierarchy. The proposed method clusters the cells with the connection status of the circuit. After constructing the partition hierarchy, a partition improvement algorithm, HYIP^[11] using hybrid bucket structure, unclusters the hierarchy to get partition results. The experimental results on ACM/SIGDA benchmark circuits show improvement up to 10-40% in minimum cutsize over the previous algorithm^{[3] [4] [5] [8] [9]}. Also our technique outperforms ML^[10] represented multilevel partition method by about 5% and 20% for minimum and average cutsize, respectively. In addition, the results of our algorithm with 10 runs are better than ML algorithm with 100 runs.

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronic Engineering, Hanyang Uni-

versity)

接受日字:1998年7月22日, 수정완료일:1998年9月21日

I. 서론

회로의 집적도 및 복잡성이 크게 증가함에 따라 대규모의 집적회로 설계에 적합한 분할 방법이 다양하게 제안되고 있다. 일반적으로 분할에 있어 빠른 수행 속도와 안정된 결과에 의거하여 분할 개선에 의한 분할 방법이 각광을 받고 있는데 분할 개선에 의한 방법 역시 대규모 회로 적용 시에 결과가 급격히 나빠지는 문제점이 발견되었다. 이러한 성능 저하를 보완하기 위하여 다단 분할 방법이 제안되었고 이에 대한 연구가 활발하게 이루어져 왔다^{[2] [4] [5] [6] [7] [10]}. 다단 분할 방법은 크게 두 단계로 나눌 수 있는데 우선 분할 대상 셀의 수를 줄이기 위해 클러스터링을 이용하여 분할 계층 구조를 형성하는 단계이다. 이 단계에 있어서는 계층 구조의 형성 방법이 분할 성능 및 속도를 결정하는 중요한 요소로써 작용하므로 형성 방법에 관한 많은 연구가 이루어지고 있다. 이 과정에 의해 형성된 계층구조를 가지고 실제 분할 개선을 수행하는 단계가 분할 개선 단계이다. 분할 개선은 분할 계층구조의 최상위 레벨부터 반복적으로 레벨을 낮추어 가면서 분할을 수행하는 단계로써 분할 개선의 초기분할은 직상위 레벨의 분할 결과를 이용한다. 따라서 다단 분할의 성능을 결정하는 주요 요인은 계층구조의 형성 방식과 분할 개선 방법으로 요약할 수 있다.

현재까지 가장 효율적인 다단 분할 방식으로 평가되는 ML^[10] 방법은 계층 구조의 형성 방식을 매칭에 의한 클러스터링 기법을 이용하였고 분할 개선 방법으로는 CLIP^[3] 방법을 채택하였다. ML방법에 있어서 특정 레벨에 대하여 상위 레벨을 형성하는 대략적인 방법은 다음과 같다. 주어진 레벨에 포함되어 있는 각각의 셀에 대하여 순차적으로 그 셀과 가장 연결도가 높은 셀을 선택한다. 이러한 방식으로 결정되어진 셀 쌍을 하나의 셀로 클러스터링하여 상위 레벨을 구성해 간다. 이와 같이 연결도가 높은 두 셀을 선택하여 하나의 상위 레벨 셀로 형성하는 방식을 매칭에 의한 클러스터링 방법이라한다. 물론 계층구조를 형성함에 있어 한 셀에 매칭되는 셀이 없는 경우에는 그 셀이 단독으로 상위 레벨 셀을 형성한다. 이러한 계층구조 형성 방식을 이용하기 위해서는 약간의 제한 조건이 요구되는데 그 첫번째가 매칭을 제한이다. 매칭을이란 레벨의 전체 셀 중에 매칭에 의해 상위 레벨 셀로 형성되는 비율을 의미하는데 이를 제한을 하지 않으면

주어진 레벨에서의 거의 100% 매칭 되기 때문이다. 그 이유는 특정 셀과 연결도가 가장 높은 셀은 그 연결도의 절대치에는 상관없이 대부분 상대적으로 존재하기 때문이다. 또 하나의 제한 조건은 레벨의 한계치의 설정이다. 이러한 한계치를 설정하지 않으면 최종적으로는 모든 셀이 아주 적은 수의 셀로써 클러스터링되기 때문이다. 그러나 이러한 제한 조건을 설정하는 것은 간단한 작업이 아니다. 그 이유는 전체 회로의 연결관계에 대한 고려 없이 설정된 확실적인 제한 조건은 분할 성능을 크게 저하시키기 때문이다. 이러한 이유로 ML에서는 이러한 제한 조건을 실험에 의해 결정하였다.

그러나 이러한 실험에 의한 인위적인 제한 조건의 설정은 회로에 따라서는 분할 성능이 크게 저하되는 문제점을 내포하고 있다. 일부 회로를 대상으로 실험한 결과에 따르면 제한 조건의 변경에 의해 결과가 대폭 개선될 수 있다는 사실을 발견하였다.

따라서 본 논문에서는 매칭 제한 조건을 회로 연결 상태에 따라 자율적으로 결정할 수 있는 방법을 제안하였다. 이러한 자율적인 매칭을 조절 방법은 레벨 한계치 역시, 회로의 연결 상태 및 분할 개선의 성능을 고려하여 결정하는 기능을 갖는다. 제안된 방법을 이용하여 회로의 연결 상태에 의해 자율적으로 계층구조를 형성하고 형성된 계층구조에 HYIP 분할 개선 방법을 반복적으로 적용함으로써 안정적이며 효율적인 분할 성능을 얻었으며 이를 실험을 통해 입증한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 다단 분할 방법을 ML을 중심으로 고찰하고 계층구조 형성시의 주요 분할 성능 결정 요인을 제시한다. 제 3장에서는 본 논문에서 제안하는 가변적인 연결도 임계치 설정에 의한 자율적인 계층구조 형성 방법에 대해 논하고 이러한 방법을 적용한 경우에 일부 실험 대상 회로의 매칭을 변화 예를 살펴본다. 제 4장에서는 ACM/SIGDA의 벤치마크 회로를 대상으로 실험한 결과 및 제안된 방법의 효과에 대해 언급한다. 끝으로 제 5장에서는 제안된 방법에 대한 결론을 논한다.

II. 기존의 다단 분할 방법 고찰

1. 기존의 다단 분할 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 V 는 회로상의 셀의 집합이고 E 는 네트의 집합이다. 일반적으로 회로에 있어서의 양 분할(bipartition)은 이 하이퍼그래프 G 를 두 부분집합 V_1 과 V_2 로 분할하는 것을 의미하고 이때 V 의 모든 셀은 V_1 혹은 V_2 중에 어느 한 부분집합에만 반드시 속하도록 제한한다. 또한 cut이란 V_1 과 V_2 에 걸쳐 있는 네트를 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_1 과 V_2 의 크기 조건(balance tolerance)을 만족시키는 상태에서 cutsize를 최소화 하는 부분집합을 결정하는 것이다.

다단 분할 방법은 이러한 분할 대상 셀을 계층 구조화하여 상위 계층으로부터 분할 개선을 반복 수행하여 분할 결과를 산출해 내는 방법으로써, 특히, 대규모 집적회로의 분할에 있어 효율적으로 이용될 수 있다. 일시에 분할 개선을 수행하는 것에 비해 다단 분할 방법은 계층 구조의 형성에 의해 초기 분할에 대한 의존도를 낮출 수 있다는 장점을 가진다. 이는 계층 구조의 형성에 의해 최상위 레벨에서만 초기 분할을 수행하고 그 이하 레벨에서는 직상위 레벨의 분할 개선 결과를 초기 분할로 이용할 수 있기 때문이다. 이때 최상위 레벨의 대상 셀 수는 분할 대상 셀 수에 비해 극히 적으므로 초기 분할의 결과에 따른 분할 결과 편차를 대폭 줄일 수 있다. 다단 분할 방법의 또 하나의 장점은 분할 개선의 성능이 대규모 셀에 대해서는 안정적이지 못하다는 단점을 보완할 수 있다는 점이다. 이는 상위 레벨로부터 반복적으로 분할 개선된 결과를 가지고 어느 특정 레벨에서 분할 수행함으로써 일시에 분할 개선을 수행한 결과보다 비교적 안정적인 결과를 얻을 수 있다.

다단 분할 방법은 위에서 언급한 바와 같이 연결도가 높은 셀들을 하나의 상위 레벨 셀로 클러스터링하여 계층구조를 형성하는 단계와 이를 최상위 레벨부터 최하위 레벨까지 반복적인 분할 개선을 통해 실제 분할 결과를 얻어 내는 분할 개선 단계로 이루어진다.

계층 구조의 형성을 위해 ML방법에 있어서는 매칭을 이용한 클러스터링 방법을 채택하였는데 이를 좀더 자세히 살펴보면 아래와 같다. 임의의 레벨에서 상위

레벨 형성 방법은 다음과 같은 일련의 절차를 가지고 수행된다. 우선 그 레벨의 셀을 random한 순서의 리스트로 만든 후 각각의 셀에 대해 순차적으로 그 셀과 연결된 셀 중에 가장 연결도가 높은 셀을 찾는다. 특정 셀에 대해 연결도가 가장 높은 셀을 발견하면 두 셀을 클러스터링하여 하나의 상위 레벨 셀로 합성한다. 이러한 방법은 매칭을 시도하는 셀의 순서에 따라 계층구조의 성능이 크게 좌우되는 문제점을 가지고 있다. 즉 순서에 따라서는 연결도가 회로 전체의 전반적인 연결도에 비해 아주 적은 셀간에도 매칭될 수 있다는 것이다. 더욱이 이러한 매칭은 실제 연결도가 높은 셀간의 매칭을 방해할 수 있다. 이는 일단 한 셀과 매칭되어 상위레벨로 올라간 셀은 다른 셀과의 매칭이 금지되기 때문이다. 따라서 매칭에 의한 클러스터링 방법은 순서 의존적이다. 또한 이 방법은 일정 제약 조건이 없다면 상대적으로 높은 연결도를 가진 셀은 대부분 존재하므로 매칭율은 거의 100%에 가깝다. 이러한 경우 거의 연결도가 없는 셀들간의 매칭이 상위 레벨 측면에서는 연결도가 높은 셀간의 매칭과 동일시되기 때문에 분할 성능을 저하시키는 요인으로 작용한다.

ML방법에서는 단순히 한 레벨에서의 매칭율을 획일적으로 제한하는 방법을 사용하였다. 매칭율을 100%, 50%, 33%등 세가지 경우를 가지고 실험한 결과 50%를 선택하였다. 그러나 순서 의존적인 계층구조 형성방법에 이러한 방식의 매칭율을 제한하는 것은 실제 반드시 매칭되어야 할 셀들간의 매칭을 촉진하는 것이 아니고 단순히 매칭 속도의 둔화, 즉 계층구조 레벨 확대라는 의미에 불과하다. 실제 50% 매칭율 적용 결과 보다는 33%의 매칭율 적용 결과가 우수하지만 수행 시간이 과도하게 증가한다는 단점이 있었다. 다시 말해서 이러한 방식의 제한 조건을 이용하면 매칭율이 높을수록 수행 속도는 향상되는 반면 수행 결과는 저하되고 반대의 경우에는 결과는 향상되나 지나친 수행 시간이 요구 된다는 것이다. 따라서 이러한 제한 조건은 성능 향상에 한계가 있다.

또한 위에서 언급한 바와같이 계층 구조의 형성 과정을 진행하면서 어느 레벨에서 계층구조 형성을 중지할 것인가를 결정해야 하는데 이 중지 조건을 ML에서는 한 레벨의 셀 수로 설정하였다. 즉 생성된 상위 레벨의 셀 수가 32개 이내이면 계층 구조의 형성을 중지하고 이를 최상위 레벨로 분할 개선을 수행하였다.

이는 분할 개선을 고려한 셀의 크기나 현재 레벨의 회로 연결 관계에는 무관한 단순 실험치에 불과하여 다양한 회로에 적용함에 있어 문제가 있다. 현재까지 가장 효율적인 성능을 가진 것으로 평가되는 ML의 경우를 예를 들었지만 대부분의 다단 분할 방법이 이러한 부류의 문제점을 내포하고 있다.

2. 계층구조 형성 방식과 분할 성능

다단 분할 방법에 있어 매칭에 의한 계층 구조 형성은 ML방법의 실험 결과에서 보여지듯이 다른 클러스터링 방법에 비해 뛰어난 성능을 가진 것으로 평가된다. 그러나 위에서 언급한 바와 같이 계층구조의 형성 방식에 있어 개선되어야 할 사항이 존재하는데 그 중에 분할 성능에 가장 큰 영향을 주는 것이 매칭을 제한조건이며 이를 제거하는 것이 가장 중요하다. 따라서 이러한 인위적인 매칭을 제한 조건을 제거하고 회로 연결 상태를 고려하여 자율적으로 매칭율이 제한되는 방식이 요구된다. 물론 이러한 자율적인 제한 방식이 전반적인 매칭율을 크게 낮춘다면 수행 속도가 과도하게 증가하는 문제가 발생하므로 이를 고려하여야 한다. 또한 계층 구조의 레벨을 결정하는 방법 역시, 최상위 레벨의 초기 분할과 직결되므로 적절한 방법이 요구된다.

본 논문에서는 위에 언급한 문제점을 해결하기 위한 새로운 자율적인 제한 방식을 제안하고 대규모 집적회로의 효율적인 분할을 위한 새로운 다단 분할방법에 대하여 제 3장에서 언급한다.

Ⅲ. 가변적인 연결도 임계치 설정에 의한 다단 분할 방법

1. 연결도 임계치의 정의

임의의 두 셀간의 연결도는 그들 간의 상호 연결 관계를 수치화 한 것으로써 두 셀에 연결된 네트의 가중치의 합에 의해 결정된다. 이와 같은 경우에 있어 multi-point connected 네트 대한 고려가 필요한데 이는 two-point connected 네트와 연결도를 차별화 하기 위함이다. 이러한 차별화 방법은 앞서서도 언급한 바와 같이 많은 방법이 제안되었는데 본 논문에서는 multi-connected 네트의 가중치를 $1/|e|$ 로 할당하는 방법을 이용하였다. 여기서 $|e|$ 는 그 네트와 연결된 셀 수를 의미한다.

이를 이용하여 임의의 두 셀 v, w 간의 연결도는 다음과 같이 정의된다.

$$\text{연결도 Conn}(v, w) = \sum_{e \in (e|v \in e, w \in e)} \frac{1}{|e|}$$

이와 같은 각 셀간의 연결도를 회로 전체에 적용하면 임의의 한 셀과 연결되어 있는 다른 셀과의 평균 셀당 총 연결도는 다음과 같이 계산되어진다.

$$\text{평균 셀당 총 연결도 AvgConn}_c = \frac{|N|}{|V|}$$

여기서 $|N|$ 은 그 하이퍼그래프 상의 네트 수의 총합이고 $|V|$ 는 셀 수의 총합이다. 이때 self loop 네트나 isolated 셀은 제외한다. 분할에 있어서 이러한 셀과 네트는 분할 결과에 영향을 주지 않으므로 무시할 수 있다. 평균 셀당 총 연결도는 각 셀에 연결된 네트 연결도의 총합을 셀 수로 나눈 값이다. 한 네트의 연결도의 합은 1이므로 각 셀에 연결된 네트의 연결도의 총합은 $|N|$ 이 됨을 쉽게 알 수 있다. 이러한 평균 셀 연결도는 한 셀이 다른 셀과 연결된 힘을 표시하므로 회로의 평균적인 복잡성을 판단하는데 좋은 정보로 이용될 수 있다.

또한 평균 네트에 의한 연결도 역시 주어진 회로에 대해 즉시 계산 될 수 있는데 이는 다음과 같이 정의될 수 있다.

$$\text{평균 네트당 연결도 AvgConn}_n = \frac{\sum_{e \in (N)} |e|}{|N|}$$

이러한 평균 네트당 연결도는 평균적으로 한 네트가 임의의 셀에 연결되었을 때 기대되어지는 연결도이다. 이외에 매칭을 결정하는 단계에서 각각의 셀에 대하여 다음과 같은 정보를 알 수 있다. 그 중에 하나는 그 셀의 총 연결도로서 그 셀에 연결된 네트의 연결도의 합이다.

$$\text{특정 셀에 대한 총 연결도 TotConn}(v) = \sum_{e \in (N|v \in e)} \frac{1}{|e|}$$

이 특정 셀에 대한 총 연결도는 그 셀과 연결된 다른 셀에 의해 받는 힘을 표시한다. 따라서 총 연결도가 높은 셀은 다른 셀과 강하게 연결(stringly connected)되어 있고 연결도가 낮은 셀은 비교적 약하게 연결(weakly connected) 됨을 알 수 있다. 또한 그 셀과 연결된 다른 셀간의 셀 당 평균 공유 네

트 수 AvgNet(v)는 매칭 수행을 위한 연결도를 계산 하면서 쉽게 구할 수 있다. 이는 다음과 같다.

$$\text{한 셀의 평균 공유 네트수 AvgNet}(v) = \frac{\sum_{e \in (v, e)} (|e| - 1)}{\text{연결 셀 수}}$$

연결도 임계치를 설정하는 것은 한 셀의 매칭 셀을 선택하는데 있어 매칭 연결도의 최소 임계치를 결정함으로써 이 임계치 이상의 연결 관계간의 매칭만을 허용하고자 함에 있다. 이는 무분별한 매칭을 방지하는데 그 목적이 있다. 이러한 매칭을 위한 임계치를 연결도 임계치라 정의하였고 이는 임의의 셀이 그 셀에 연결된 셀들과 가질 수 있는 평균 연결도로서 설정되며 그 임의의 셀과 다른 셀과의 연결 관계에 따라 가변하는 값을 갖는다. 이때 이 연결도 임계치는 회로 전체의 평균적인 네트당 연결도와 그 셀의 연결 상태에 의한 평균 공유 네트수의 곱으로 이루어진다. 이는 그 셀이 평균적인 네트에 의해 연결되었다고 가정할 때 그 셀과 다른 셀과의 연결 상태를 감안하여 계산된 연결된 셀과의 평균 연결도를 의미한다. 따라서 이 값 이하의 연결도는 그 셀에 있어서는 평균 이하의 연결 관계를 의미하므로 임계치로 이용할 수 있다. 이러한 연결도 임계치는 다음과 정의된다.

$$\text{연결도 임계치 ThresConn}(v) = \text{AvgConn}_n * \text{AvgNet}(v)$$

2. 연결도 임계치의 다단 분할 방법 적용

위에서 언급한 연결도 임계치는 특정 셀을 중심으로 회로 전체의 평균적 네트를 가지고 그 셀의 현재 평균 연결 관계를 갖는 경우의 연결도를 의미하므로 이 임계치 이하의 연결도를 갖는 셀은 그 특정 셀에 대해 낮은 연결도를 갖는 것으로 간주할 수 있다. 그러나 이러한 방식에 의해 매칭을 결정하면 셀의 총 연결도가 평균적인 셀 당 총 연결도보다 큰 경우에는 문제가 없지만, 그렇지 않은 경우에 있어서는 대부분 그 임계치를 초과하는 연결도를 갖는 매칭 셀을 찾는 것이 어렵다. 그 이유는 다음과 같다. 셀의 총 연결도가 평균 셀당 총 연결도보다 작다는 것은 일반적으로 그 셀에 평균 이하의 연결도를 갖는 네트가 비교적 적은 수가 연결된 경우가 확률적으로 높다는 것을 의미한다. 따라서 연결 관계의 편차가 매우 큰 경우를 제외하고는 임계치 이상의 연결도를 갖는 셀이 존재하지 않는다. 그러나 이러한 총 연결도가 낮은 셀도 그 셀의 연결

상황에 따라 상대적으로 높은 연결도를 갖는 셀과의 매칭이 필요한 경우가 있다. 이러한 평균 이하의 총 연결도를 갖는 셀의 매칭을 유도하기 위해 연결도 임계치를 다음과 같이 정한다.

```

If(AvgConnc < TotConn(v))
    연결도 임계치 ThresConn(v) = AvgConnn
    * AvgNet(v)
Else
    연결도 임계치 ThresConn(v) = AvgConnn
    * AvgNet(v) * TotConn(v)/ AvgConnc
    
```

3. 가변 연결도 임계치 설정에 의한 다단 분할 알고리즘
본 다단 분할 알고리즘의 구성은 그림 1과 같다.

Threshold based Multilevel Partition Algorithm	
INPUT	: G ₀ (V ₀ , E ₀)
VARIABLES	: m = Number of levels P _i = Intermediate bipartitioning
OUTPUT	: P ₀ = (X ₀ , Y ₀) = Final bipartitioning
1. while 1 do 2. G _{i+1} (V _{i+1} , E _{i+1}) = Match(G _i). 3. if G _{i+1} == G _i then break . 4. Set i = i + 1. 5. endwhile 6. Let m = i. P _m = Partition(G _m , Null). 7. for i = m - 1 down to 0 do 8. P _i = Partition(G _i , P _{i+1}). 9. endfor 10. return P ₀	

그림 1. 제안된 연결도 임계치에 의한 다단 분할 알고리즘

Fig. 1. The proposed multilevel partition algorithm using connectivity threshold.

그림 1에서와 같이 본 알고리즘은 계층구조를 형성하는 단계와 계층 구조의 최상위레벨에서부터 반복적으로 분할 개선을 수행하는 단계로 구성된다. 계층 구조의 형성 단계에서 매칭이 더 이상 발생하지 않으면 계층구조 형성 단계를 종료하는 방법을 제시하였다. 분할 개선 단계에서는 최상위 레벨에서는 임의 초기 분할을 이용하였고 이를 제외한 그 이하 레벨에서는 바로 직상위 레벨의 분할 결과를 초기 분할로 취하여 분할을 수행한다.

본 알고리즘에서 사용한 매칭 단계의 구성은 그림 2와 같다.

Procedure Match	
INPUT	: $G_i(V_i, E_i)$
VARIABLES	: AvgConn _c = 평균 셀당 총 연결도 AvgConn _n = 평균 네트당 연결도
OUTPUT	: $G_{i+1}(V_{i+1}, E_{i+1})$
<ol style="list-style-type: none"> 1. Construct random permutation Π of V_i. 2. Calculates AvgConn_c and AvgConn_n. 3. while $j < V_i$ do 4. Set $k = k+1$. Add $v_{\Pi(j)}$ to V_{i+1}. 5. if exists $w = \text{FindMatch}(v_{\Pi(j)}, \text{AvgConn}_c, \text{AvgConn}_n)$ then 6. add w to V_{i+1}. 7. Set $j = j + 1$. 8. endwhile 9. return $G_{i+1}(V_{i+1}, E_{i+1})$ 	

그림 2. 매칭 프로시저의 구성
Fig. 2. The matching procedure.

각 셀에 대한 최대 매칭 셀을 찾기 전에 회로 전체의 평균 셀당 총 연결도 및 평균 네트당 연결도를 계산한 후 매칭율을 결정 짓는 FindMatch 프로시저로 전달한다. 본 알고리즘에서는 이러한 연결도 정보를 이용하여 매칭 연결도 임계치를 계산하고 이를 이용하여 매칭을 제한하여 매칭율이 회로 연결 상태에 따라 자율적으로 조절되는 기능을 갖는다. 이 프로시저의 구성은 그림 3과 같다.

FindMatch Procedure	
INPUT	: $v_{\Pi(j)}$ AvgConn _c = 평균 셀당 총 연결도 AvgConn _n = 평균 네트당 연결도
VARIABLES	: TotConn(v) = 한 셀의 총연결도 AvgNet(v) = 한 셀의 평균 공유 네트수 ThresConn(v) = 연결도 임계치
OUTPUT	: $w = \text{Matching}$ 셀
<ol style="list-style-type: none"> 1. Calculates $\text{conn}(v_{\Pi(j)}, w)$ for $w \in V_i$. 2. Calculates TotConn(v), AvgNet(v). 3. if AvgConn_c < TotConn(v) then ThresConn(v) = AvgConn_n * AvgNet(v) else then ThresConn(v) = AvgConn_n * AvgNet(v) * TotConn(v) / AvgConn_c 4. Find unmatched w that maximize $\text{conn}(v_{\Pi(j)}, w) / (\text{size}(v_{\Pi(j)}) * \text{size}(w))$ and $\text{conn}(v_{\Pi(j)}, w) > \text{ThresConn}(v)$ and $\text{size}(v_{\Pi(j)}) + \text{size}(w) < \text{ThresSize}$ 4. if exists w then return w else then Null 	

그림 3. 매칭 셀 계산 프로시저
Fig. 3. The Findmatch procedure for a cell.

최대 매칭 셀을 찾는 프로시저에서는 전달 받은 평균 셀당 총 연결도 및 평균 네트당 연결도와 각 셀의 연결관계에 의해 계산되어지는 한 셀의 총연결도 및 한 셀의 평균 공유 네트수를 이용하여 그 셀의 매칭 연결도 임계치를 계산한다. 매칭 후보 셀 중에 연결도 임계치 이하의 후보 셀은 매칭을 금지시킨다. 또한 매칭되는 두 셀의 크기 합이 일정 크기(ThresSize) 이상이면 매칭을 금지하였는데 이는 분할 개선의 성능 향상을 위하여 과도하게 큰 셀의 형성을 방지하는데 목적이 있다. 본 알고리즘에서 이러한 임계 크기를 분할 영역 제한 크기(balance tolerance)로 제한하였다. 임계 크기는 다음과 같이 정하였다.

$$\text{임의의 셀의 임계 크기 ThresSize} = \frac{A(V) \cdot r}{2}$$

여기서 A(V)는 분할 대상 회로의 셀 크기의 총합이고 r은 balance tolerance이다. 이러한 방식을 적용할 경우 확실적인 매칭율을 설정하는 것보다 대부분 높은 매칭율을 갖는다. 그림 4의 예와 같이 제안된 방법을 적용하는 경우에 있어 ML 방법(매칭율 0.5)에 비해 평균적으로 높은 매칭율을 나타내었고 이에 따라 계층구조 형성시간의 단축되었다.

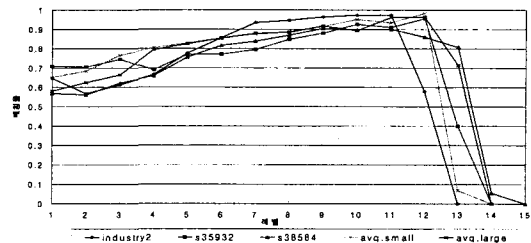


그림 4. 실험 회로에 대한 매칭율 변화
Fig. 4. The variances of matching ratio for experiments.

IV. 실험 및 고찰

본 알고리즘에서 이용하는 매칭 연결도 임계치의 계산은 별도의 추가 계산 없이 매칭을 진행시키면서 직접 계산할 수 있으므로 ML 다단 분할 방법과 동일한 time complexity를 갖는다.

본 알고리즘은 ACM/SIGDA에서 제공한 벤치마크 회로 23개에 대해 실험되었고 표 1에서는 본 논문에서 제안한 가변 연결도 임계치 설정에 의한 계층구조

형성과 HYIP^[11]에 의한 분할 개선을 적용한 분할기(TMLh)에 대하여 10회 수행한 결과를 GM^[4], HB^[5], PB^[9], GFM_t^[8], CL-LA3_f, CD-LA3_f, CL-PR_f^[3]과 최소 cutsize를 비교하였다. PB는 대표적인 클러스터링 기반의 분할 방법이며 GM과 HB는 다단 분할 방법, GFM_t는 two-phase 분할 방법이고 CL-LA3_f, CD-LA3_f, CL-PR_f는 클러스터 제거에 의한 분할 방법이다. 또한 대표적인 다단 분할 방법으로 알려진 ML_c와의 비교는 ML_c를 10회와 100회 실험한 결과와 비교하였다. 실험 결과 제안된 알고리즘이 실험한 결과 기존 분할 방식에 비해 약 10-40% 가량의 최소 cutsize 감소 효과가 있었고 기존의 다단 분할 방법 중에 가장 효율적인 방법으로 평가되는 ML^[10]에 비해 제안된 방법이 최소 cutsize에 있어서는 약 5%, 평균 cutsize에 있어서는 평균 20%이상의 성능 향상을 가져 왔다. 더욱이 제안된 방법을 10회 수행한 결과가 ML 방법을 100회 수행한 결과 보다 앞서는 성능을 보인점에 주목할 만하다.

표 2에서는 제안된 알고리즘의 안정성을 입증하기 위하여 본 알고리즘과 ML 알고리즘을 예제 회로에 대해 100회 수행한 후에 평균 cutsize를 비교하였다. 실험 결과에 의하면 본 알고리즘이 ML 알고리즘에 비해서 평균 cutsize에서 12-28%가량의 결과 향상이 있었다. 이는 본 알고리즘이 기존 방식에 비해 상당히 안정된 성능으로 수행됨을 알 수 있다.

표 2. ML알고리즘과의 평균 cutsize 비교
Table 2. Average cutsize comparison of TMLh and ML algorithm.

Circuits	Average Cut Size(100runs)							Improvement(%)						
	TML _c	ML _c (1.0)	ML _c (0.5)	ML _c (0.33)	ML _c (1.0)	ML _c (0.5)	ML _c (0.33)	ML _c (1.0)	ML _c (0.5)	ML _c (0.33)	ML _c (1.0)	ML _c (0.5)	ML _c (0.33)	
s9234	43	50	47	47	48	45	45	16.3	9.3	9.3	11.6	4.7	4.7	
Biomed	85	103	96	94	92	91	91	21.2	12.9	10.6	8.2	7.1	7.1	
s13207	57	77	72	71	76	71	68	35.1	26.3	24.6	33.3	24.6	19.3	
s15850	52	83	58	59	59	56	57	21.6	11.6	13.5	13.5	7.7	9.6	
ind2	181	213	207	207	197	196	202	17.7	14.4	14.4	9.8	8.3	61.3	
ind3	256	275	277	275	274	276	276	7.4	8.2	7.4	7.0	7.9	7.6	
s35932	44	46	49	49	46	45	46	4.5	9.1	11.4	9.1	2.2	9.1	
s38584	48	77	58	77	58	52	52	60.4	16.7	60.4	20.8	8.3	8.3	
Avqsm	131	182	147	148	194	159	156	38.9	12.2	13	48.1	21.4	19.1	
s38417	52	66	56	56	82	72	68	26.9	7.7	7.7	36.6	38.5	30.8	
Avqlg	133	183	148	148	200	183	157	37.5	11.3	11.3	50.4	22.6	18.1	
Golem3	1372	2008	1465	1413	2026	1462	1421	46.2	6.8	2.3	47.7	6.6	3.4	
Average Imprv(%)								27.8	12.2	15.5	24.6	13.3	16.0	

표 1. 기존의 분할 방법과 최소 cutsize 비교
Table 1. Comparison of TMLh and other partition algorithms.

Circuits	Cut Size										Improvement(%)								
	TML _h (10)	GM	HB	PB	GFM _t	CL-LA3 _f	CD-LA3 _f	CL-PR _f	ML _c (10)	ML _c (100)	GM	HB	PB	GFM _t	CL-LA3 _f	CD-LA3 _f	CL-PR _f	ML _c (10)	ML _c (100)
Balu	27	27		41	28	27	27	27	27	27	0.0		0.0	3.7	0.0	0.0	0.0	0.0	0.0
bm1	47	48				51	47	47	51	47	2.1			8.5	0.0		0.0	8.5	0.0
p1	47	47		53	51	51	47	51	52	47	0.0		12.7	8.5	8.5	0.0	8.5	10.6	0.0
i4	48	49				49	48	52	49	48	2.1			2.1	0.0		8.3	2.1	0.0
i3	56	62				56	57	57	58	56	10.1				0.0	1.8	1.8	3.6	0.0
i2	87	95				91	89	87	92	89	9.2			4.6	2.3	0.0	5.8	2.3	
i6	60	94				60	60	60	60	60	56.7				0.0	0.0	0.0	0.0	0.0
Struct	30	33		40	36	33	36	33	33	33	10.0		33.4	20.0	10.0	20.0	10.0	10.0	10.0
i5	71	104				80	74	77	72	71	46.5				12.7	4.2	8.5	1.4	0.0
19ks	104	106				104	104	104	108	106	1.9				0.0	0.0	0.0	3.9	1.9
p2	139	142		146	139	142	151	152	145	139	2.2		5.0	0.0	2.2	8.6	9.4	4.3	0.0
s9234	40	43	45	74	44	45	44	42	41	40	7.5	12.5	85.0	10.0	12.5	10.0	5.0	2.5	0.0
Biomed	83	83		135	92	83	83	84	84	83	0.0		62.7	10.8	0.0	0.0	1.2	1.2	0.0
s13207	53	70	62	91	61	66	69	71	55	55	32.1	17.0	71.7	15.1	24.5	30.2	34.0	3.8	3.8
s15850	43	53	46	91	46	71	59	56	56	44	23.3	7.0	111.6	7.0	65.1	37.2	30.2	30.2	2.3
ind2	165	177		193	175	200	182	192	174	164	7.3		17.0	6.1	21.2	10.3	16.4	5.5	-0.1
ind3	241	243		267	244	260	243	243	243	243	0.8		10.8	1.3	7.8	0.8	0.8	0.8	0.8
s35932	41	57	46	62	44	73	73	42	42	41	39.0	12.2	51.2	7.3	78.1	78.1	2.4	2.4	0.0
s38584	47	53	52	55	54	50	47	51	48	47	12.8	10.6	17.0	14.9	6.4	0.0	8.5	2.1	0.0
Avqsm	127	144		224	129	139	144	134	128	128	13.4		76.4		1.6	9.5	13.4	5.5	0.8
s38417	49	69		49	62	70	74	65	50	49	40.8		0.0	26.5	42.9	51.0	32.7	2.0	0.0
Avqlg	128	144		139	127	137	143	131	128	128	12.5		8.6		-0.8	7.0	11.7	2.3	0.0
Golem3	1332	2111		1629				1374	1346	1346	58.5		22.3						
Average Imprv(%)											16.9	11.9	36.6	10.1	14.0	12.3	9.2	4.9	1.0

표 3에서는 위에서 언급한 각 분할 방법의 수행 시간을 비교하였다. 본 알고리즘과 GM, CL-LA3f, CD-LA3f, CL-PRf, ML은 SUN SPARC5 Model 85에서 수행된 결과이고, PB는 DEC3000 Model 500 AXP에서 수행된 결과이고 GFM은 SUN SPARC 10에서 수행된 결과이다. 비록 서로 다른 종류의 workstation상에서 수행되었지만 거의 비슷한 성능으로 가정할 수 있다^[3]. 이를 고려하면 기존에 분할 알고리즘에 비해 비교적 적은 혹은 대등한 수행 시간으로 우수한 결과를 얻을 수 있었다.

선하기 위해 계층 구조를 진행 시키면서 매칭을 주어진 계층의 연결 상황에 따라 자율적으로 매칭을 제한하는 방법을 제안하였다. 이를 위해 회로의 평균 연결도와 매칭이 수행되는 각각의 노드의 현재 연결 상태를 고려한 가변적인 매칭 연결도 임계치를 정의하였고 이를 매칭 수행 시 효율적으로 적용하는 방법을 제시하였다. 또한 이러한 임계치의 적용을 통해 계층 구조의 계층이 자율적으로 결정될 수 있음을 보였고 실험을 통하여 제안된 방식이 기존 알고리즘에 비해 효율적이며 매우 안정적임을 입증하였다.

표 3. 분할 방법별 수행 시간
Table 3. Comparison of CPU times of various algorithm.

Circuits	CPU Time(s)							
	TML _r (10)	GM	PB	GFM	CL-LA3 _r	CD-LA3 _r	CL-PR _r	ML _r (10)
balu	12	14	16	25	32	31	34	17
bm1	12	12			37	47	36	18
p1	14	12	18	25	36	48	37	18
t4	26	21			81	106	114	41
t3	22	23			88	107	95	47
t2	27	26			99	124	109	45
t6	29	32			50	55	175	55
struct	18	27	35	32	45	54	75	35
t5	54	46			141	162	188	74
19ks	62	39			178	216	219	84
p2	73	53	137	61	167	210	353	90
s9234	82	58	490	186	175	270	264	97
biomed	145	95	711	371	231	362	572	172
s13207	156	102	2060	397	220	429	380	155
s15850	191	114	1731	530	267	543	576	189
ind2	514	245	1367	819	1129	1453	2127	502
ind3	612	299	761	861	1419	1944	1920	667
s35932	445	266	2627	1088	463	964	1085	427
s38584	442	397	6518	3463	748	1339	1950	490
avqsm	582	328	4099		1260	2507	2082	603
s38417	425	281	2042	1062	811	1733	1590	496
avqlg	698	417	4135		1430	3145	2126	666
golem3	11018	450	10823					10483
Sub	15659	3357						15471
total(s)	15427		37570					
	3129			8920				
	4641				9107	15849	16107	

V. 결론

본 논문에서는 대규모 집적회로 설계에 있어 효율적이고 안정된 분할을 위한 새로운 다단 분할 방법을 제안하였다. 기존에 제안된 다단 분할 방법은 분할 계층 구조 형성에 있어 인위적인 제한 조건에 의해 계층의 구조를 조정함에 따라 분할 대상에 따라 그 결과가 안정적이지 못한 단점이 있었다. 본 논문에서는 이를 개

참고 문헌

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

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

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

[4] C. J. Alpert, L. W. Hagen and A. B. Kahng, "A hybrid multilevel/genetic approach for circuit partitioning", Physical Design Workshop, 1996, pp. 100-105.

[5] S. Hauck and G. Borriello, "An evaluation of bipartitioning techniques", Proc. of the 16th Conf. On Advanced Research in VLSI, 1995, pp. 383-402.

[6] B. Hendrickson and R. Leland, "A Multilevel Algorithm for Partitioning Graphs", Proc. Supercomputing, 1995.

[7] G. Karypis and V. Kumar, "Multilevel Graph Partitioning Schemes", Proc. Intl. Conf. on Parallel Processing, vol. 3, pp. 10-13, 1995.

[8] L. T. Liu, M. T. Kuo, S. C. Huang, and C.-K. Cheng, "A Gradient Method on the

- initial Partition of Fiduccia-Mattheyses Algorithm”, Proc. Intl. Conf. Computer-Aided Design, 1994, pp. 63-67.
- [9] 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.
- [10] C. J. Alpert, J.-H. Huang and A. B. Kahang, “Multilevel Circuit Partitioning”, Proc. ACM/IEEE Design Automation Conf., 1997, pp. 530-533.
- [11] 임창경, 정경화, “하이브리드 버킷을 이용한 대규모 집적회로에서의 효율적인 분할 개선 방법”, 대한전자공학회 논문집, 35권, C편, 3호, 1998

 저 자 소 개

任 昌 慶(正會員) 第35卷 C編 第3號 參照
 현재 한양대학교 전자공학과 박사과
 정 재학 중

鄭 正 和(正會員) 第35卷 C編 第3號 參照
 현재 한양대학교 전자공학과 교수 재
 직