

한정된 그룹 이동에 의한 위상 기반 회로 분할 방법

(A Topology Based Partition Method by Restricted Group Migration)

南敏祐*, 崔然景**, 林鐘錫***

(Min Woo Nam, Yeun Kyung Choi, and Chong Suck Rim)

요 약

본 논문에서는 다중의 FPGA 칩과 연결 전용 칩으로 구성되어 있는 프로그래밍이 가능한 PCB(Programmable Circuit Board)를 대상으로 주어진 회로를 분할하는 새로운 회로 분할 방법을 제안한다. 여기서 칩들간에는 상호 연결 가능한 배선 위상이 정해져 있으며 사용할 수 있는 연결선의 수가 고정되어 있다. 그러므로 회로를 PCB상의 다중의 FPGA 칩으로 분할하기 위해서는 기존의 분할 방법과는 달리 칩들간의 연결선에 대한 제한 조건을 고려하여야 하며 이를 위하여 본 논문에서는 주어진 PCB의 모든 제한 조건을 고려한 분할 방법을 제안한다. 또한 분할 속도를 개선하면서 보다 좋은 분할 결과를 얻기 위하여 다단계의 클러스터 트리를 생성하여 계층적 분할을 수행한다. 다수의 벤치마크 회로에 대하여 실험한 결과 입력회로들은 주어진 제한 조건들을 모두 만족하면서 분할되었으며 기존의 다중 분할 방법과 비교한 결과에서 칩간의 연결선의 수가 최대 10 % 적게 사용되었다.

Abstract

In this paper, we propose a new multi-way circuit partitioning system that partition large circuits to programmable circuit boards which consist of FPGAs and interconnect components. Here the routing topology among the chips are predetermined and the number of available interconnections are fixed. Since the given constraints are difficult to be satisfied by the previous partition method, we suggest a new multi-way partition method by target restriction that considers all the constraints for the given board. To speed up, we construct a multi-level cluster tree for hierarchical partitioning. Experimental results for several benchmarks show that the our partition method partition them by satisfying all the given constraints and it used up to 10 % fewer interconnections among the chips than the previous K-way partition method.

* 正會員, 三星電子 技術總括 멀티미디어研究所
(Multimedia Lab. Corporate Technical Operations,
Samsung Electronics Co.)

** 正會員, 慶收大學 電子計算科
(Dept. of Computer Science, Kyungmin College)

*** 正會員, 西江大學校 컴퓨터學科
(Dept. of Computer Science and Engineering, Sogang
University)

接受日字: 1998年8月20日, 수정완료일: 1998年12月24日

I. 서 론

현재 ASIC(Application Specific Integrated Circuit) 분야에서 VLSI(Very Large Scale Integrated) 칩의 크기가 수십만 게이트에 이르고 있으며 시스템의 생명주기(life cycle)가 짧아짐에 따라서 설계 검증 단계의 시간을 줄이는 것이 매우 중요한 문제가 되고 있다.

FPGA(Field Programmable Gate Array)^[1]는 회로 동작을 검증하는 시간이 짧고, 설계 오류를 수정

한 이후 재구성(reconfiguration)이 가능하여 설계 검증을 위하여 널리 사용되고 있다. 그런데 수십만 게이트가 넘는 회로를 하나의 FPGA 칩으로는 구현하기가 곤란하므로 일반적으로 회로를 여러 개의 그룹으로 분할하여 각 그룹을 FPGA 칩에서 구현한 후 칩간을 연결하여 시험한다.

최근에는 설계 검증을 보다 편리하게 할 수 있도록 FPGA 칩이 이미 장착된 PCB들이 등장하였다^[2,3]. FPCB(Field Programmable Circuit Board)의 경우에는 FPGA 칩과 이들간을 연결하는 연결 전용 칩인 FPIC(Field Programmable Interconnect Component)TM 칩이 장착되어 있어서 FPGA 칩간의 연결을 쉽게 재프로그래밍할 수 있다^[2].

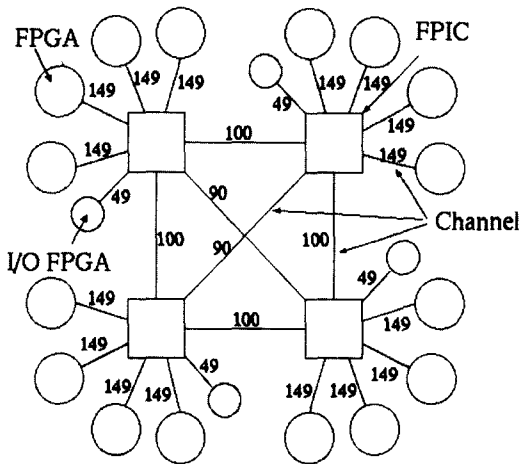


그림 1. 위상그래프
Fig. 1. Topology graph.

그런데 FPCB 상의 모든 칩들간에는 서로간의 연결을 위한 배선 위상이 정해져 있으며 사용 가능한 연결선들의 수도 고정되어 있다. 이들 보드 구조는 위상그래프(topology graph) $G=(L \cup I, E)$ 로 형식화할 수 있다^[4]. 여기서 L 과 I 는 각각 FPGA 칩과 연결 전용 칩을 나타내는 노드들의 집합이며 E 는 두 노드 간에 연결선을 나타내는 에지들의 집합이다.

그림 1에 AXB-AP4 보드^[2]를 형식화한 위상그래프를 보인다. AXB-AP4 보드는 4개의 FPIC 칩과 16개의 FPGA 칩, 그리고 4개의 I/O 구현용 FPGA (I/O FPGA) 칩으로 구성된 FPCB이다. 그림에서 칩들을 서로 구분하기 위하여 FPGA 칩은 원형으로 FPIC 칩은 사각의 노드로 나타내며 I/O FPGA는 작은 원으로 나타낸다. 여기서 회로가 분할되어 들어갈

수 있는 대상은 집합 L 에 해당하는 원으로 표현한 FPGA 칩들이며 이를 앞으로 '블록(block)'이라고 한다.

에지에 해당하는 두 노드간의 연결선들의 집합을 채널(channel)이라고 할 때 채널을 사용 가능한 네트의 수를 '채널 용량(channel capacity)'이라고 한다. 그림 1의 에지 옆에 표기된 숫자는 채널 용량을 나타낸다. 각 채널을 사용하는 네트의 수는 주어진 채널 용량 보다 작아야 하며 이를 '채널 용량 제한 조건(channel capacity constraint)'이라고 한다. 위상이 고정된 보드 구조에 주어진 회로를 분할하는 문제는 회로를 위상그래프의 블록으로 분할하는 것과 동일하다. 위상그래프 상의 분할 문제는 기존의 분할 문제와는 달리 채널 용량 제한 조건을 만족하여야 하므로 이를 위하여 새로운 분할 문제로 위상 기반 회로 분할 문제(topology-based circuit partition problem)가 제안되었다^[4].

위상 기반 회로 분할 문제는 다음의 조건을 만족하면서 채널을 사용하는 네트의 수를 최소화하도록 회로를 주어진 FPGA 칩으로 분할하는 문제이다. 즉 (1) 각 분할된 회로의 크기가 FPGA 칩의 크기 보다 작거나 같고(이를 앞으로 '크기 제한 조건(size constraint)'이라고 한다), (2) 각 칩의 입출력으로 사용되는 연결선의 수가 주어진 칩의 입출력 버퍼(buffer)의 수 보다 작거나 같아야 하며(이를 앞으로 '핀 제한 조건(pin constraint)'이라고 한다), 그리고 (3) 분할 후 칩들간의 연결을 위하여 사용하는 네트의 수가 사용 가능한 연결선의 수를 넘지 않도록 해야 한다.

그런데 지금까지의 대부분의 연구에서 분할 문제는 회로를 각 분할의 크기 제한 조건을 위배하지 않으면서 컷(cut) 수를 최소화하는 문제^[5-13]이거나, 분할의 대상이 되는 디바이스(device)의 크기 제한 조건과 핀 제한 조건을 만족하면서 회로를 분할하는 문제였다^[14-18].

첫 번째 분할 문제를 해결하는 가장 대표적인 이중 분할 방법으로는 FM^[7] 분할 방법을 들 수 있다. Sanchez^[9]는 FM 분할 방법을 확장하여 임의의 K 개의 블록으로 회로를 분할하는 방법(이하 'K-way 분할 방법'이라고 한다)을 제안하였다. K-way 분할 방법은 각 블록에서 사용하는 핀수를 비용 함수로 정의하고 레벨 이득(level gain)을 이용하여 회로 이동에 의한 영향을 고려하여 분할하였다.

디바이스의 제한 조건들을 고려한 분할 방법 중에서 Woo와 Kim^[18]은 K-way 분할 방법을 기반으로 각 FPGA의 크기와 핀 제한 조건을 만족하면서 사용하는 핀수의 합을 최소화하는 방법을 제안하였으며, Kuznar 등^[17]은 FPGA의 크기와 핀 제한을 만족하면서 디바이스의 비용을 최소화하기 위한 방법을 제안하였다. Ober 등^[19]의 경우에는 FPCB와 같은 보드를 위한 분할 방법을 제안하였으나 연결 전용 칩들간의 사용하는 연결선들의 수에 대한 정확한 계산이 힘들다.

결국 기존의 분할 방법들은 연결 전용 칩간의 연결선들에 대한 고려를 하지 않으므로 분할 결과에서 앞에서 설명한 (3)의 제한 조건이 만족되지 않을 수 있다. 그러므로 위상 기반 회로 분할 문제를 위해서는 새로운 효율적인 분할 방법을 개발할 필요가 있다.

본 논문에서 제안한 분할 방법은 주어진 회로를 AXB-AP4 보드의 FPGA들로 분할할 때, 모듈 이동(move)시 전체 모듈과 블록을 대상으로 하지 않고 일부의 모듈과 블록으로 한정하여 이들을 대상으로 수행하는 방법이다. 여기서 분할 문제의 비용과 이동을 위한 이득은 위상 기반 회로 분할 문제에 보다 적합하도록 새로운 함수를 정의하여 계산하였으며 수행시간을 줄이기 위하여 클러스터 트리(cluster tree)라고 하는 클러스터로 구성된 계층구조를 형성하여 계층적인 분할을 수행하도록 하였다.

제안한 분할 방법을 구현하여 여러 입력 회로에 대하여 분할한 결과 모든 회로들이 주어진 보드로 모든 제한 조건을 만족하면서 분할되었으며 블록의 크기나 핀수만을 고려한 분할 방법과 비교한 결과에서는 채널을 사용하는 네트의 수를 최대 10 % 더 줄이는 좋은 결과를 보였다.

본 논문은 서론을 포함하여 모두 4 장으로 구성된다. 다음 장에서는 논문에서 제안하는 위상 기반 회로 분할 방법에 대하여 설명한다. 제 3 장에서는 실험결과를 보이고, 마지막으로 제 4 장에서 결론과 향후 연구방향을 제시한다.

II. 한정된 그룹 이동에 의한 위상 기반 회로 분할 방법

이 장에서는 본 논문에서 제안한 한정된 그룹 이동에 의한 위상 기반 회로 분할 방법에 대하여 기술한

다. 먼저 1 절에서 제안한 분할 방법의 개요를 설명하고, 2 절에서는 위상 기반 회로 분할 방법에 적합하도록 새롭게 정의한 비용 함수와 사용하는 채널 수를 예측하기 위한 배선에 관하여 기술하고, 마지막으로 3 절에서는 분할 방법을 구체적으로 기술한다.

1. 분할 방법의 개요

본 논문에서 제안한 위상 기반 회로 분할 문제는 각 블록의 크기 제한 조건과 핀 제한 조건을 만족하며 블록간의 연결을 위하여 사용하는 채널의 채널 용량 제한 조건을 만족하도록 회로를 분할하는 문제이다. 그런데 위상그래프 $G=(L \cup I, E)$ 에서 에지 E 는 블록 L 과 연결 전용 칩 I 사이의 연결과 I 와 I 사이의 연결로 구성된 채널을 의미하는데, L 과 I 를 연결하는 채널 용량은 블록 L 의 핀 용량과 같거나 작게 되므로 채널 용량 제한 조건을 만족하는 분할 결과는 항상 핀 제한 조건을 만족한다. 그러므로 제안한 위상 기반 회로 분할 문제는 크기 제한 조건과 채널 용량 제한 조건을 만족하며, 블록간의 연결을 위하여 사용하는 채널 수를 최소화하는 문제로 정의할 수 있다.

이를 위하여 제안한 분할 방법은 기존의 K-way 분할 방법과 유사하게 모듈 이동에 의하여 분할을 수행한다. 그런데 위상 기반 회로 분할 문제에서 크기 제한 조건을 위배하지 않도록 주어진 회로를 분할하기 위해서는 블록의 용량을 넘지 않게 분할을 수행해야 하는데, 주어진 회로가 기술사상(technology mapping)되어 있는 경우에는 사용되는 LUT(Look Up Table)의 수를 통하여 회로의 크기를 알 수 있지만, 게이트 레벨의 회로를 분할하는 경우에는 필요한 LUT의 수를 알기 힘들다. 그러므로 이들 회로들은 MFFC(Maximum Fanout Free Cone)^[21] 클러스터링 방법을 통하여 사용하는 LUT의 수를 예측하여 그 크기를 이용하며 분할을 수행한다.

채널 용량 제한 조건을 위배하지 않도록 회로를 분할하기 위해서는 블록간을 연결하는 네트가 사용하는 채널을 정확하게 계산할 필요가 있으며 이를 이용하는 새로운 비용함수를 정의한다. 네트가 사용하는 채널들을 정확히 계산하기 위해서는 연결 전용 칩들간을 연결하는 네트에 대하여 실제로 배선이 수행한다. 또한 사용률을 초과하는 채널이 존재하지 않도록 기중치를 추가하여 모듈의 이동시 채널 용량을 초과하는 블록으

로의 이동을 제한한다.

네트의 배선은 많은 수행시간을 필요로 하므로 분할을 위한 계산 시간을 단축하기 위한 방법들이 필요하다. 이를 위하여 모듈들로 계층구조를 형성하여 이를 분할에 이용하고 모듈 이동시 전체 모듈 중에서 일부의 모듈을 선정하여 이를 선정한 일부의 블록에 대하여 이동하는 분할 방법을 구현한다.

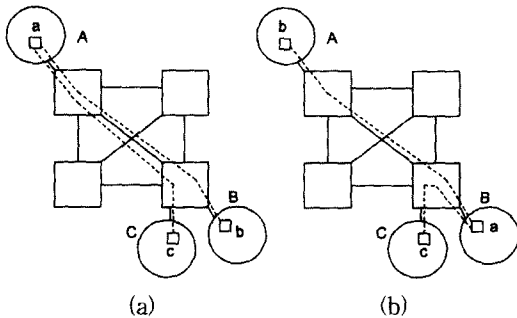


그림 2. 두 개의 분할 해
Fig. 2. Two Partition Solution.

그러므로 본 논문에서 제안한 분할 방법은 위상 기반 회로 분할 문제를 위하여 최적의 해를 빠른 시간에 얻기 위한 방법으로 이를 '한정된 그룹 이동에 의한 분할 방법'이라고 하며 이의 특징을 정리하면 다음과 같다.

- (1) 위상 기반 회로 분할 문제에 적합한 채널을 이용한 새로운 비용 함수를 사용한다.
- (2) 블록간의 연결을 위하여 사용되는 채널 수를 예측하기 위하여 배선을 수행한다.
- (3) 채널 용량 제한 조건을 만족하기 위해서 채널 사용률에 따른 가중치를 추가한다.
- (4) 전체 모듈과 블록 중에서 일부의 모듈과 블록을 선정하고 이들을 대상으로 모듈의 이동을 한정한다.
- (5) 클러스터들로 계층구조를 형성하고 계층적 분할 방법을 구현한다.

앞의 (1), (2), (3) 항은 위상 기반 회로 분할 문제의 제한 조건을 만족하기 위한 것이며, (4), (5) 항은 계산시간을 단축하기 위한 방법이다.

2. 비용 함수와 배선

위상 기반 회로 분할 문제에서 채널 수를 최소화하기 위해서는 블록간의 연결을 위하여 각 네트가 사용하는 채널 수의 합을 최소화하여야 한다. 그런데 기존의 K-way 분할 방법이나 Woo와 Kim의 분할 방법

에서 블록간을 연결하는 네트들이 각 블록에서 사용하는 핀수의 합을 비용 함수로 정의하였으며, 이득 계산을 회로 이동에 의한 핀수의 감소분으로 구성하였으므로^[9,18] 위상 기반 분할 문제에서는 그대로 적용하기 곤란하다.

그림 2에 두 개의 분할 해를 보인다. 그림에서 작은 사각형은 모듈을, 점선은 네트를 나타낸다. 그림 2의 (a), (b)에서 각 블록이 사용하는 핀수의 합은 네 개로 동일하지만 네트들이 사용하는 채널 수의 합은 (a)에서는 여섯 개이고 (b)에서는 다섯 개로 다르게 나타난다. 즉 핀수만을 고려하는 분할 문제에서는 (a), (b)가 동일한 분할 해를 의미하지만, 위상기반 분할 문제에서는 (b)가 더 좋은 결과임을 알 수 있다.

그러므로 위상 기반 회로 분할 문제를 해결하기 위한 새로운 비용 함수를 다음과 같이 정의한다. 여기서 N 을 전체 네트의 집합, n 은 임의의 네트라고 할 때 $CH(n)$ 은 네트 n 이 사용하는 채널의 수를 의미한다.

$$total\ cost = \sum_{n \in N} CH(n)$$

즉 비용 함수는 각 네트가 모듈의 연결을 위하여 사용하는 채널 수의 합이다.

비용 함수에서 각 네트가 사용하는 채널의 수를 계산하고, 채널 용량 제한 조건을 만족하는 지를 알기 위해서는 분할된 회로를 배선한다^[4]. 배선은 위상그 래프 $G=(L \cup I, E)$ 상에서 수행하는데, 블록 L 과 연결 전용 칩 I 를 연결하는 에지 E 는 유일한 경로(path)이므로 배선을 적용할 필요가 없고, I 와 I 사이의 에지 E 에 대하여 채널 사용률이 100%를 초과하는 경우 큰 가중치를 주어 다른 채널을 사용하여 배선되도록 하는 미로 배선(maze routing) 방법^[20]을 이용한다.

FM 분할 방법^[7]과 K-way 분할 방법^[9]에서는 네트의 연결 정보를 이용하여 이득의 갱신 작업을 수행하는데, 네트의 연결 정보는 네트 n 에 연결된 모듈들이 어느 블록에 몇 개씩 분포하는지를 나타내는 정보로 아래와 같이 표현할 수 있다.

$$n(M_{\ell_1}(n), M_{\ell_2}(n), \dots, M_{\ell_{|L|}}(n))$$

여기서 $M_{\ell_i}(n)$ ($\ell_i \in L, 1 \leq i \leq |L|$)은 네트 n 에 연결된 모듈 중 블록 ℓ_i 에 속한 모듈의 개수를 의미

한다.

네트 n 에 연결된 모듈의 이득은 $M_{i,j}(n)$ 의 값이 $0 \rightarrow 1$ 이 되거나 $1 \rightarrow 0$ 이 되는 두 경우에 변화하고, 배선의 수행 역시 이 경우에만 발생한다. 그림 1의 위상그래프와 같이 $|L|=16$ 일때 각 블록에 대해서 두 가지 경우에만 이득의 갱신 및 배선의 수행이 발생하므로 최대 32회의 상수값만큼 배선을 수행한다.

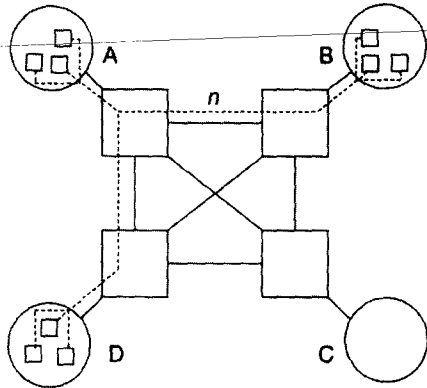


그림 3. 네트 n의 배선 결과
Fig. 3. Routing result for net n.

그림 3에 네트의 연결 정보를 통한 배선의 예를 보인다. 네트 n 의 연결 정보는 블록 A, B, D에 연결된 모듈들이 세 개씩 존재하고, 블록 C에는 연결된 모듈이 존재하지 않으므로 $n(3,3,0,3)$ 과 같이 나타낸다. 여기서 각 필드는 알파벳 순서에 의한 블록을 의미하고, 각 필드의 숫자는 블록에 포함된 모듈의 수를 의미한다.

그림 3에서 블록 A에 있던 모듈 한 개가 블록 C로 이동하는 경우에는 연결 정보가 $n(2,3,1,3)$ 이 되어 세 번째 필드에서 $0 \rightarrow 1$ 인 경우가 발생한다. 이 경우에는 블록 C로 새로운 연결이 필요하므로 네트 n 에 대하여 배선을 한다. 그러나 만일 그 모듈이 블록 B로 이동하는 경우에는 연결 정보가 $n(2,4,0,3)$ 으로 이미 블록 B에 네트 n 을 위한 연결선이 있으므로 배선을 할 필요가 없다.

배선은 초기 분할에서 각 네트에 대하여 한 번 수행하고 그후에는 연결 상태가 달라지는 네트들에 대해서만 수행하는데, 각 네트에 대한 배선은 I 의 개수를 $|I|$ 라고 하고, I 에서 서로 다른 I 를 연결하는 에지 수의 최대값을 $|E_{max}|$ 라고 할 때, 배선의 복잡도는 $O(|I| \times |E_{max}|)$ 이다. 그림 1의 위상그래프의

경우에는 $|I|=4$ 이고, $|E_{max}|=3$ 이므로 상수값의 복잡도를 갖는다.

3. 한정된 그룹 이동에 의한 분할 방법

이 절에서는 본 논문에서 제안한 한정된 그룹 이동에 의한 위상 기반 회로 분할 방법을 자세히 기술한다. 그림 4에서 분할 방법의 개요를 보인다. 먼저 회로를 클러스터링하여 형성된 클러스터들로 클러스터 트리를 구성한 후 클러스터 트리에서 원하는 클러스터들을 선택한다. 여기서 클러스터는 클러스터링된 회로들이거나 주입출력 또는 입력 회로 자체로 구성된 모듈 등으로 편의상 모두 '모듈'이라고 한다.

입력된 회로가 게이트 레벨의 회로인 경우 다음과 같이 클러스터 트리를 구성한다. 우선 회로를 MFFC 클러스터링 방법에 의하여 MFFC 단위의 클러스터를 생성한다. MFFC 클러스터를 구하는 방법은 기술사에서 LUT를 구하는 방법과 유사하므로 MFFC 클러스터의 수로 LUT 수를 예측할 수 있다. 만일 회로가 기술사상된 회로이면 이 과정을 생략할 수 있다.

Procedure partitioning_for_AP4

begin

입력 회로로 클러스터 트리 구성

클러스터 트리에서 모듈을 선택

초기 분할 수행

블록을 연결하는 네트들을 배선

while(Block_Select()==True){

대상 그룹 결정

while(Cost_Down()==True){

대상 그룹으로의 모듈 이동에 대한

이득 계산

각 블록을 위한 버킷을 구성

while(Module_Select()==True){

/* 최대 이득을 갖는 모듈 중 이동 가능한 것을 선택 */

선택된 모듈을 이동

인접한 모듈의 이득을 갱신

} 이득의 합이 최고인 이동들까지만

실질적으로 이동

}

end procedure partitioning_for_AP4

그림 4. 분할 방법의 개요

Fig. 4. Overview of Partition Method.

MFFC는 주출력(primary output)에 연결된 모듈들로부터 BFS(Breadth First Search) 방법을 이용

하여 찾는다. 주출력에서부터 주입력(primary input)이나 다른 주출력으로의 경로를 구성할 수 있는 모듈을 만날 때까지 방문할 수 있는 모든 모듈의 집합을 하나의 MFFC로 한다.

MFFC들을 대상으로 LRC(Local Ratio Cut) 클러스터링 방법^[22] 의하여 반복적으로 클러스터들을 구하여 클러스터 트리를 형성한다. LRC 클러스터링은 기존의 ratio cut 클러스터링을 수정한 방법이다. 기존의 방법이 임의로 선택한 모듈의 집합 M 과 \bar{M} 을 제외한 모든 모듈들의 집합 \bar{M} 를 대상으로 ratio cut 방법을 적용하여 클러스터를 구하는데 비하여, LRC 클러스터링은 임의로 선택한 모듈의 집합 M 과 \bar{M} 에 연결되어 있는 모듈들의 집합 M_c 를 대상으로 ratio cut 방법을 이용하여 클러스터를 구한다. 이때 상위 레벨의 클러스터들의 크기는 포함된 하위 클러스터들의 크기의 합으로 결정한다.

이렇게 형성된 클러스터 트리에서 모듈을 선택하여 초기 분할을 수행한다. 초기 분할은 클러스터 트리에서 가장 크기가 큰 클러스터들을 대상으로 모듈을 선택하여 초기 분할을 구한 후, 이들에 대하여 각각 제안한 분할 방법을 한 패스 적용하는 방법을 10 회 적용하여 이중 가장 좋은 분할 결과를 초기 분할로 사용하였다.

초기 분할 후에는 분할 결과에 대하여 전체 네트에 대하여 배선을 수행한다. 여기서의 초기 배선 결과는 이후의 분할 과정에서 모듈의 이동에 따른 이득 계산 시 사용되며 2.2 절에서 설명한 바와 같이 네트의 연결 상태가 변하는 경우에 대해서만 배선이 이루어진다.

배선 후에는 임의의 블록을 선택하여 분할을 수행할 이동대상그룹을 구한다. '이동대상그룹'이란 이동할 수 있는 모듈들의 집합을 말한다. 이동대상그룹들이 놓인 블록들의 집합에서 이동대상그룹들이 이동할 수 있는 블록들의 집합을 '이동블록'이라고 하며 이동대상그룹들은 이동블록들로의 이동만이 허용된다. 이동대상그룹은 먼저 하나의 블록을 선택하고(이 블록을 '선택한 블록'이라고 함), 이 블록에 놓인 모듈들과 이들과 네트로 연결된 다른 블록들에 놓인 모듈들의 집합을 합집합으로 한다.

그럼 5에 이동대상그룹의 예를 보인다. 그림에서 F 를 선택한 블록이라고 할 때 F^c 는 F 에 놓인 모듈들과 연결된 모듈들의 집합이 속한 F 를 제외한 블록들의 집합을 나타내고, \bar{F} 는 전체 모듈 중에서 F

와 F^c 를 제외한 나머지 블록들의 집합을 의미한다. 여기서 F 와 F^c 에 놓인 모듈들의 집합이 이동대상그룹이며, F 에 놓인 모듈들의 이동블록은 F^c , F^c 에 놓인 모듈들의 이동블록은 F 가 된다.

이동대상그룹을 정하기 위하여 블록을 선택할 때에는 채널 사용률이 가장 높은 채널에 연결된 블록을 우선적으로 선택한다. 그 이유는 사용률이 높은 채널들이 채널 용량 제한 조건을 위배할 가능성이 가장 크기 때문이다. 이와 같이 전체 블록을 한번씩 선택하여 이동에 의한 분할 과정을 수행하며 한 패스에서 블록은 중복 선택되지 않도록 한다.

한정된 그룹 이동에 의한 분할 방법에서 임의의 블록 F 를 선택하여 이동대상그룹에 대하여 이동블록만으로서의 이동을 수행하는 것은 F 에 연결된 채널의 채널 용량 제한 조건을 만족시키고 이 채널을 사용하는 네트의 수를 최소화하기 위한 방법이다. 이와 같이 이동대상그룹을 한정하여도 F 에 연결된 채널을 사용하는 네트의 수는 전체 F 에 포함된 모듈들이 F^c 로, F^c 에 포함된 모듈들이 F 로 이동함으로써 대부분 감소할 수 있기 때문이다. 따라서 한정된 그룹 이동에 의한 분할 방법을 수행하여도 실험 결과에는 큰 차이를 보이지 않으며, \bar{F} 에 대한 이동을 생략하므로 계산 시간을 단축하는 효과를 얻을 수 있다. 또한 모든 블록을 한번씩 선택하여 분할을 수행하므로 이동대상그룹으로 이동을 한정하는 것을 보완할 수 있다.

논문에서 제안한 한정된 그룹 이동에 의한 분할 방법은 임의의 블록 F 를 선택하고 F 에 연결된 채널의 채널 용량 제한 조건을 만족시키며 이 채널을 사용하는 네트의 수를 최소화하기 위한 방법이다. 전체 비용을 증가시키지 않으면서 F 에 연결된 채널을 사용하는 네트의 수를 감소시키는 이동은 F 에 포함된 모듈들이 F^c 로 이동하는 경우와 F^c 에 포함된 모듈들이 F 로 이동하는 경우가 대부분이다.

따라서 이 두 그룹으로 한정된 이동을 수행하여도 실험 결과에는 큰 차이를 보이지 않으며, \bar{F} 에 대한 이동을 생략하므로 계산 시간을 단축하는 이득을 얻을 수 있다. 또한 블록은 사용되는 최대의 블록 수만큼 돌아가면서 선택되어 분할을 수행하므로 각 블록에서는 한정된 그룹에서의 이동을 수행하지만 지역 최소값(local minimum)에 빠질 가능성은 줄어든다.

선택한 블록에 대하여 이동대상그룹과 이동블록들이

정해지면 이동대상그룹들에 대하여 이동블록으로의 이득을 계산한다. 모듈의 이득은 이동블록으로 이동하였을 때의 네트가 사용하는 채널 수의 감소분과 채널 사용률에 의한 가중치로 구성된다. 채널 수 외에 가중치를 추가로 적용하는 이유는 모듈의 이동 시 채널 용량 제한 조건을 위배하지 않는 이동을 우선적으로 수행하고, 그 반대의 경우를 억제하기 위함이다.

각 모듈은 이동대상그룹의 이동블록 각각에 대하여 이득을 가지는데, 모듈 m 이 블록 A에서 블록 B로 이동할 경우 이득 함수 $gain(m)_{A \rightarrow B}$ 는 다음과 같이 정의한다.

$$gain(m)_{A \rightarrow B} = (cost_A(m) - cost_B(m)) + (over_A(m) - over_B(m)) \times W$$

여기서 $cost_A(m)$ 은 모듈 m 이 블록 A에 있을 때, m 에 연결된 네트들이 사용하는 채널 수의 합을 의미하고, $cost_B(m)$ 는 모듈 m 이 블록 B에 있는 경우를 의미한다. $over_A(m)$ 와 $over_B(m)$ 는 각각 모듈 m 이 이동하기 전과 이동 후의 채널 용량을 초과하여 사용하는 채널의 수를, W 는 임의의 상수값을 나타낸다. $(cost_A(m) - cost_B(m))$ 은 채널 수의 감소분을, $(over_A(m) - over_B(m)) \times W$ 는 채널 사용률에 의한 가중치를 의미한다. 여기서 상수값 W 의 값이 1 보다 작으면 모듈의 이동에 미치는 이득값에 영향이 적고, 2 보다 큰 경우는 채널 용량을 초과하여 사용하는 채널 수만을 줄이는 결과가 되어 분할 결과에 나쁜 영향을 줄 수 있다. 그러므로 W 의 값을 1 또는 2의 값을 주어서, 채널 용량을 초과하는 네트의 수를 줄이는 이동에 대해서는 이득이 증가하여 우선적으로 수행하도록 한다.

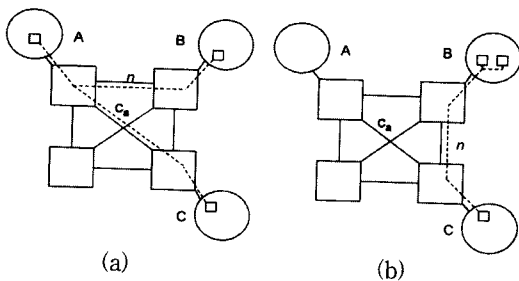


그림 6. 이득 계산의 예
Fig. 6. An example for calculating the gain.

그림 6에 모듈의 이동에 대한 이득 계산의 예를 보인다. 그림 6 (a)는 모듈 m 이 블록 A에서 블록 B로

이동하기 전의 상태를 나타내고, 그림 6 (b)는 모듈 m 이 이동한 후의 상태를 나타낸다. 그림 6 (a)에서 채널 C_a 는 이 채널을 사용하는 네트의 수가 채널의 용량을 초과하는 채널이라고 가정할 때, 모듈 m 의 이동에 의한 이득은 $(5-3)+(1-0) \times W$ 가 된다.

이득은 편리하게 관리하기 위하여 이동 블록에 포함된 블록마다 버킷을 구성한다. 버킷은 각 모듈의 이득을 관리하는 자료구조로써 어떤 모듈을 선택하여 어느 블록으로 이동할 것인지의 순서를 정하는 작업을 편하게 할 수 있도록 도와준다. 제안한 분할 방법에서는 버킷에 있는 모듈 중에서 이득이 가장 크면서도 각 블록의 크기 제한을 위배하지 않는 것을 선택하여 이동을 한다.

Max_{gain} 은 하나의 모듈이 이동함으로써 발생하는 최대 이득 값으로 버킷을 구성할 때 필요한 최대, 최소 값을 의미한다. 이때 컷 수만을 고려하는 FM 분할 방법에서는 $Max_{gain} = \text{'모듈의 최대 핀수'}$ 이고, 레벨 이득을 적용하지 않고 핀수만을 고려한 K-way 분할 방법에서는 $Max_{gain} = 2 \times \text{모듈의 최대 핀수} + 1$ 이다(여기서 '모듈의 최대 핀수'는 분할 대상이 되는 전체 모듈 중에서 가장 많은 핀을 보유한 모듈의 핀수를 말한다). K-way 분할 방법에서는 모듈의 이동에 따라 블록간의 핀수가 2 개씩 줄어들기 때문이다.

본 논문에서 제안한 위상 그래프상에서 채널을 고려한 분할 방법에서는 $Max_{gain} = \text{'최대 채널 변화 수'} \times \text{모듈의 최대 핀수} + \text{'최대 채널 변화 수'} \times W$ 로 계산한다. 여기서 '최대 채널 변화 수'는 하나의 네트에 대하여 사용하는 채널 수가 모듈 이동에 의하여 변할 수 있는 최대 값을 의미하고, W 는 이득 함수에서 사용한 가중치 상수값이다. 이는 모듈의 이동에 따라 핀수가 변하는 것이 아니라 사용되는 채널의 수가 변하므로 '최대 채널 변화 수 \times 모듈의 최대 핀수' 만큼 변하고, 채널의 사용률에 따라 가중치 W 를 이득 계산에 추가시켰으므로 이를 더한 값만큼이 Max_{gain} 이 된다. 임의의 모듈 m 의 이득 $gain(m)$ 은 $-Max_{gain} \leq gain(m) \leq Max_{gain}$ 의 범위를 가지며, 이때 한 버킷의 크기는 $2 \times Max_{gain} + 1$ 이 된다.

Max_{gain} 의 값이 FM 분할 방법이나 K-way 분할 방법 보다는 증가하였지만 최대 채널 변화 수가 상수 값이고, W 역시 상수값이므로 버킷을 관리하는 작업은 FM 방법에서와 같이 $O(p)$ 의 계산 복잡도를 가

진다.

선택된 모듈이 이동하면 그 모듈을 고정(Lock)시키고 이득을 갱신한다. 만일 이동한 모듈에 의하여 모듈에 연결된 네트들의 연결 상태가 변하면 이들 네트에 대하여 배선을 다시 수행하고 이동한 모듈과 인접한(incident) 모듈들에 대해서만 이득을 갱신한다. 그림 1의 위상그래프에 기반한 블록으로의 분할에서 이득의 갱신 작업은 2.2 절의 배선에서 설명한 것과 같이 최대 32 회의 상수 번 만큼 이루어 지므로 전체 계산 복잡도는 K-way 분할 방법에서와 같이 $O(p)$ 에 가능하다.

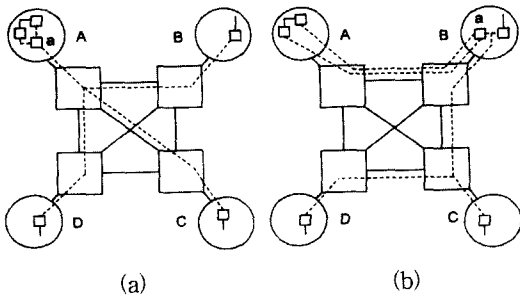


그림 7. 이동의 예
Fig. 7. An example of move.

그림 7에 모듈 이동의 예를 보인다. 그림 7 (a)는 모듈 a가 블록 A에서 B로 이동하기 전의 상태를 나타내며, (b)에서는 이동 후의 상태를 나타낸다. 점선으로 표현된 네트는 모듈 a에 연결된 네트로 모듈 a의 인접한 모듈과의 연결을 나타내며 다른 네트들은 실선으로 나타낸다. 그림에서 이동 전과 이동 후에 변화가 있는 네트는 점선으로 표현된 네트이므로 이와 연결된 모듈들에 대한 이득만 갱신하면 된다.

이동 가능한 모든 모듈에 대하여 이동이 종료되고 나면 이득의 합이 가장 큰 상태까지 실질적인 이동을 수행하여 하나의 분할 해를 얻는다. 임의의 블록을 선택한 후에 이동 그룹간의 모듈 이동에 의하여 하나의 분할 해를 구하기까지의 과정을 패스(pass)라고 한다. 분할 해를 얻는 과정은 더 이상 비용이 감소하는 해를 구하지 못할 때까지 반복한다.

이상에서 설명한 한 패스의 계산 복잡도는 전체적으로 K-way 분할 방법의 계산 복잡도와 비교하여 버킷을 관리하는 작업이 $O(p)$ 에 가능하고 이득을 갱신하는 작업에서 배선을 위한 추가 계산 과정만을 필요하므로 하므로 2.2 절에서 설명한 배선의 계산 복잡도

$O(|I| \times |E_{max}|)$ 가 추가 되어 $O(|I| \times |E_{max}| \times p)$ 의 계산 복잡도를 갖는다. 그러나 위상그래프 상에서 $|I|$ 와 $|E_{max}|$ 는 상수값을 가지고, 이득의 변화에 대한 버킷을 갱신하는 작업은 버킷이 양방향 연결 리스트(doubly linked list) 구조를 갖고 있어서 상수 시간에 가능^[9]하므로 한 패스의 계산 복잡도는 $O(p)$ 를 넘지 않는다.

모든 블록은 한번씩 선택하여 이동블록에 대하여 분할을 진행한다. 전체 블록을 한번씩 선택하여 분할한 후에는 처음 블록을 선택할 때의 비용이 최소인 해와 현재까지 분할 과정을 한 후의 비용이 최소인 해를 비교하여 비용이 현재 감소하였으면 다시 블록 선택 단계로 가서 분할과정을 반복하고 그렇지 않을 경우에는 모듈을 교체하여 분할을 반복한다.

선택한 모듈을 이용하여 분할 과정을 수행하며 더 이상 비용이 감소하는 분할 해가 존재하지 않으면 모듈의 하위 계층에 해당하는 클러스터들로 모듈을 교체한다. 교체된 모듈들은 자신의 상위 클러스터에 해당하는 모듈들에 의하여 얻은 가장 적은 비용을 갖는 현재까지의 가장 좋은 분할 해에서 그 상위 클러스터들이 속해 있던 블록으로 초기 분할되며 분할 방법이 다시 적용된다. 즉 클러스터 트리에서 하향식(top-down)으로 클러스터를 선택하여 계층적인 분할을 한다.

이와 같은 과정은 더 이상 교체할 모듈이 존재하지 않으며 더 이상의 비용이 감소하는 해가 없으면 종료한다. 모듈을 교체하여 모듈의 크기가 작아질수록 모듈의 수가 증가하지만 비용이 감소하는 모듈 이동이 적어지므로 계산시간은 감소한다.

III. 실험결과

본 장에서는 본 논문에서 제안한 분할 방법의 실험 결과를 보인다. 제안한 분할 방법은 C 언어를 사용하여 구현하였으며, 실험은 삼보 워크스테이션 SDT-820에서 수행하였다.

표 1에 실험에 사용된 ISCAS 벤치마크 회로들을 보인다. AXB-AP4 보드에서는 primary I/O 의 개수가 196개(49개x4)로 한정되어 있으므로, ISCAS 벤치마크 회로 중 primary I/O의 개수가 196개 이상인 회로에 대해서는 실험 대상에서 제외한다. Primary I/O 는 회로의 주입출력 수를 의미하고, 예측크기는

벤치마크 회로들이 게이트 레벨 회로이기 때문에 MFFC 클러스터링을 통하여 예측한 LUT 의 개수를 의미한다. 예측크기는 위상 기반 회로 분할 문제에서의 크기 제한 조건을 만족하는 결과를 위하여 필요하다.

이렇게 구성된 MFFC 클러스터에 대하여 LRC 클러스터링을 반복적으로 사용하였다. 이때 생성되는 각 클러스터의 크기와 LRC 클러스터링 방법의 반복횟수는 회로의 크기에 따라서 실험에 의하여 결정하였으며, 하나의 클러스터 크기가 블록의 크기를 넘지 않도록 하기 위하여 클러스터들은 최대 LUT 수와 최소 LUT 수를 정의하여 크기를 제한하였다.

실험은 그림 1의 위상그래프를 기반으로 회로를 열여섯 개의 블록으로 분할하는 것과 AXB-AP4 보드로 분할하는 것으로 두 가지 방식으로 수행하였다. 첫 번째 실험은 위상 기반 회로 분할 문제를 위하여 제안한 분할 방법이 기존의 분할 방법보다 효과가 있음을 보이기 위한 것이고, 두 번째 실험은 실제 보드를 사용하여 시제품을 제작할 경우 제안한 분할 방법이 효과가 있음을 보이기 위한 것이다.

표 1. 실험에 사용한 데이터들
Table 1. Data used for the experiments.

Data	Primary I/O	예측크기
C6288	64	1480
S5378	88	961
S9234	45	1071
S13207	156	1739
S15850	105	1671
S35932	359	3222

비교 대상으로 사용된 기존의 분할 방법은 위상그래프의 블록이 크기 제한 조건과 핀 제한 조건을 가지고 있으므로 기본적인 분할 방법은 Woo와 Kim이 제안한 분할 방법을 따르면서 제안한 분할 방법에서 사용한 블록의 선택과 한정된 그룹 이동을 추가로 적용한 방법(이하 'WK 분할 방법'이라고 한다)이다. 또한 비용 함수로 핀수를 사용하였으며, 배선은 수행하지 않았다. 실험 결과 중 WK의 분할 방법의 SUM(I)는 최종 분할 결과에 대해서 블록간의 연결에 사용되는 채널을 미로 배선 방법을 이용하여 구한 값이다. 이러한 WK 분할 방법을 비교 대상으로 삼은 것은 제안한

비용 함수가 더 효과가 있음을 보이기 위하여 기존의 방법에 약간의 수정을 가할 필요가 있었기 때문이다.

비용 함수에서 가중치 W 의 값은 채널 용량 제한 조건을 위배하는 채널들을 사용하는 네트를 우선적으로 제한하기 위하여 사용하는 것으로 많은 실험 결과 그 값이 1 또는 2인 경우에 좋은 분할 결과를 보였다. 일반적으로 입력 회로의 네트들이 복잡하게 연결되어 있는 경우에는 W 의 값을 2로 하였고, 그 외의 입력에 대해서는 1로 하였다.

표 2에 그림 1의 위상그래프 $G=(L, U, E)$ 에 대하여 16개 블록으로 분할하는 실험 결과를 보인다. 각 필드에서 'CH_{max}'는 블록 L 과 연결 전용 칩 I 간의 채널을 사용하는 네트 수의 최대값을 의미하고, 'CH_{min}'은 최소값을 의미한다. 'SUM(I)'는 연결 전용 칩 I 사이를 연결하는 네트들이 사용하는 채널 수의 합을, 'SUM(C)'는 전체 네트들이 블록간의 연결을 위하여 사용하는 채널 수의 합을, 그리고 'SUM(P)'는 블록 L 에서 사용된 핀수의 합을 나타낸다. 'Time(s)'은 분할을 수행하는데 걸린 시간을 나타내며 단위는 초(second)이다. 그리고 'Top'는 본 논문에서 제안한 분할 방법을 나타내며 'WK'는 WK 분할 방법을 나타낸다.

표 2. 열 여섯 블록으로의 분할
Table 2. Partition results for sixteen blocks.

Data	CH _{max}		CH _{min}		SUM(I)		SUM(C)		SUM(P)		Time(s)	
	Top	WK	Top	WK	Top	WK	Top	WK	Top	WK	Top	WK
C6288	71	94	36	18	231	336	1048	1161	817	825	1006	592
S5378	68	70	26	31	292	344	1119	1168	827	825	436	152
S9234	80	78	25	10	275	367	1080	1179	805	812	528	219
S13207	138	145	22	29	414	550	1748	1789	1334	1239	1028	549
S15850	82	87	33	35	390	489	1424	1588	1034	1099	1124	897
S35932	99	100	69	69	516	729	2232	2471	1716	1742	5946	1835

표 2에서 처럼 한정된 그룹 이동에 의한 위상 기반 회로 분할 방법이 배선 수행에 의한 부가적인 시간이 필요하며, 사용하는 총 핀의 수는 다소 높은 것도 있으나 전체적으로 SUM(I)에서는 최대 31%, SUM(C)에서는 최대 10% 적은 것으로 나타났다. 그러므로 제안한 분할 방법은 연결 전용 칩간의 채널을 고려하지 않고 핀수만을 비용 함수로 사용하는 기존의 분할 방법에 비하여 네트가 사용하는 채널의 수를 크게 줄일

수 있으며, 이는 위상 기반 회로 분할 문제에 대해서는 채널 수를 비용함수로 사용하는 것이 보다 좋은 분할 해를 얻을 수 있음을 의미한다.

두 번째 실험은 AXB-AP4 로 회로를 분할하는 것으로 그림 1의 위상그래프로 분할하는 것과 동일하다. 각 채널 용량 제한은 그림 1에 나타난 값을 사용하며 위상그래프의 각 블록을 위해서는 표 3에 보인 FPGA 칩을 사용한다. XC4010은 로직 구현용으로, XC4005는 I/O 구현용 FPGA 칩으로 사용한다. FPIC 칩으로 사용 가능한 모델은 APTIX사에서 제작한 AX1024R/AX1024D^[2]의 모델이 있는데, 이중 AX1024D는 diagnostic이 가능한 모델이고, AX1024R은 저가의 모델이다. 이중 본 실험에는 AX1024D를 사용한다. 각 디바이스에 대하여 사용 가능한 LUT 수는 각 블록의 크기 제한 조건이 된다. IOB 수는 핀 제한 조건으로 사용되어야 하나 2.1 절에서 설명한 바와 같이 채널 용량 제한이 만족되면 핀 제한 조건이 만족되기 때문에 생략할 수 있다.

표 3. AXB-AP4에서 사용 가능한 FPGA
Table 3. FPGAs available for AXB-AP4.

Device	LUT수	IOB수
XC4010	800	160
XC4005	392	112

표 4. AXB-AP4 보드에서의 분할 결과
Table 4. Partition result for AXB-AP4.

Data	칩 수	B _{max}	B _{min}	CR _{max}	CR _{min}	Time(s)
C6288	(3,2)	89%	7%	97%	15%	211
S5378	(2,2)	90%	32%	96%	75%	32
S9234	(2,1)	86%	47%	73%	60%	43
S13207	(13,4)	23%	11%	91%	36%	439
S15850	(4,3)	63%	40%	90%	77%	224

표 4에 AXB-AP4 보드에서의 실험결과를 보인다. 표에서 '칩수'는 사용한 로직 구현용 FPGA 칩 수와 I/O 구현용 FPGA 칩 수를 (로직 구현용 FPGA 칩 수, I/O 구현용 FPGA 칩 수)로 나타낸 것이며, 'B_{max}'는 가장 사용률이 높은 FPGA 칩의 사용률, 'B_{min}'은 가장 사용률이 낮은 FPGA 칩의 사용률을 나타낸다. FPGA 칩은 최대 사용률이 90% 를 넘을 수 없도록 상한을 정하였다^[17]. FPGA 칩의 최대 사

용률을 정한 이유는 MFFC 클러스터링을 통하여 게이트 레벨 벤치 마크 회로의 LUT 개수를 예측하여 '크기 제한 조건'을 만족하도록 하였으므로 예측크기와 기술 사상 이후의 LUT 개수의 오차를 감안하여 정한 것이다. 그런데 이러한 상한값은 FPGA 칩 내부의 배치와 배선 가능성을 고려하여 조정할 필요가 있을 것으로 본다.

'CR_{max}'는 FPGA와 FPIC 칩간을 연결하기 위해서 사용하는 채널 중에서 가장 사용률이 높은 것, 'CR_{min}'은 이 채널들 중에서 가장 사용률이 낮은 것을 나타낸다. Time(s)은 표 2에서와 동일하다. 표에서 처럼 제안한 분할 방법은 실험에 이용한 모든 회로에 대하여 AXB-AP4 보드의 제한 조건을 만족하면서 분할하였으며, C6288과 S13207를 제외한 모든 회로에 대하여 최적의 칩 수를 사용하며 분할되었다. C6288과 S13207의 경우에는 회로의 넷트들이 복잡하게 연결되어 있어서 채널 용량 제한 조건에 따라 여러 블록으로 분산되었기 때문이다.

이상의 실험을 통하여 제안한 분할 방법이 기존의 분할 방법보다 위상 기반 회로 분할 문제를 위하여 더 효율적인 분할 결과를 얻으며, 주어진 보드에서 각 제한 조건들을 만족하면서 분할할 수 있음을 보였다.

IV. 결론

본 논문에서는 주어진 회로를 FPGA 칩과 연결 전용 칩이 존재하는 위상이 고정된 보드로 분할하는 새로운 분할 방법인 한정된 그룹 이동에 의한 위상 기반 회로 분할 방법을 제안하였다. 제안한 분할 방법은 다른 분할 방법과는 달리 칩들간에 존재하는 채널 용량 제한 조건을 위배하지 않으면서 회로를 분할할 수 있도록 넷트가 사용하는 전체 채널 수를 비용 함수로 하였으며, 이를 이용하여 모듈의 이동에 대한 이득을 계산하였다. 분할의 속도를 개선하기 위하여 이동이 가능한 모듈의 집합인 이동대상그룹과 이동할 수 있는 칩들의 집합인 이동블록을 한정하여 수행하였으며 회로를 클러스터링하여 계층적인 분할 방법을 구현하였다. 다수의 벤치마크 회로에 대하여 실험한 결과 입력 회로들은 주어진 제한 조건을 모두 만족하면서 분할되었으며 기존의 WK 분할 방법과 비교한 결과에서는 칩간의 연결선의 수가 최대 10 % 적게 사용되어 FPCB 와 같은 보드로의 회로 구현을 위한 분할에 보

다 적합함을 보였다. 향후 다른 형태의 연결 구조를 가지는 보드에서도 적용할 수 있도록 제안한 분할 방법을 일반화하는 연구가 필요하다.

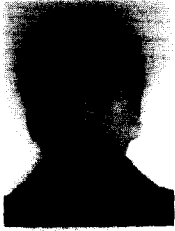
참 고 문 헌

- [1] Xilinx, Inc., The Programmable Gate Array Data Book, Xilinx, San Jose, 1994.
- [2] Aptix, Inc., System Data Book, Aptix, 1993.
- [3] V. Sankarasubramanian and D. Bhatia, "Multiway Partitioner for High Performance FPGA Based Board Architectures", ICCD, pp. 579-585, 1996.
- [4] Y. Choi, Y.S. Jung, C.S. Rim, "A Topology-Based Multiway Circuit Partitioning for ASIC Prototyping", Proc. 39th Midwest Symp. on Circuits and Systems, pp. 357-360, 1996.
- [5] C.J. Alpert, J. Huang and A.B. Kahng, "Multilevel Circuit Partitioning", 34th DAC, pp. 530-533, 1997.
- [6] C.J. Alpert and A.B. Kahng, "Multi-Way Partitioning Via Spacefilling Curves and Dynamic Programming", 31th DAC, pp. 652-657, 1994.
- [7] C.M. Fiduccia and R.M. Mattheyses. "A Linear-time Heuristic for Improving Network Partitions", 19th DAC, pp. 175-181, 1982.
- [8] B.M. Kernighan and S. Lin. "An Efficient Heuristic Procedure for Partitioning graph", Bell system technical journal, vol. 49, no. 2, pp. 297-307, Feb., 1970.
- [9] L.A. Sanchis. "Multiple-Way Network Partitioning", IEEE Trans. on Computers. vol. 38. no. 1. Jan, 1989.
- [10] M. Shih and E.S. Kuh, "Quadratic Boolean Programming for Performance-Driven System Partitioning", 30th DAC, pp. 761-765, 1993.
- [11] Y.C. Wei and C.K. Cheng, "Towards Efficient Hierarchical Designs by Ratio Cut Partitioning", ICCAD, pp. 298-301, 1989.
- [12] C.W. Yeh and C.K. Cheng, "A General Purpose Multiple Way Partitioning Algorithm", 28th DAC, pp. 421-426, 1991.
- [13] J.Y. Zien, P.K. Chan and M. Schlag, "Hybrid Spectral/Iterative Partitioning", ICCAD, pp. 436-440, 1997.
- [14] K. Roy and C. Sechen, "A Timing Driven N-way Chip and Multi-Chip Partitioner", ICCAD, pp. 240-247, 1993.
- [15] V.C. Chan and D. Lewis, "Hierarchical Partitioning for Field-Programmable Systems", ICCAD, pp. 428-435, 1997.
- [16] H. Krupnova, A. Abbara and G. Saucier, "A Hierarchy-Driven FPGA Partitioning Method", 34th DAC, pp. 522-525, 1997.
- [17] R. Kuznar, F. Brglez and B. Zajc, "Multi-way Netlist Partitioning into Heterogeneous FPGAs and Minimization of Total Device Cost and Interconnect", 31th DAC, pp. 238-243, 1994.
- [18] N.S. Woo and J. Kim, "An Efficient Method of Partitioning Circuits for Multiple FPGA Implementation", 30th DAC, pp. 202-207, 1993.
- [19] U. Ober and M. Glesner, "Multiway Netlist partitioning onto FPGA-based Board Architectures," European DAC, pp. 150-155, 1995.
- [20] C. Lee, "An Algorithm for Path Connections and Its Applications", IEEE Trans. on Electronic Computers, VEC-10, pp. 346-365, Sept., 1961.
- [21] J. Cong, Y.Ding, "On Area Depth Trade-off in LUT-Based FPGA Technology Mapping", 30th DAC, pp. 213-218, 1993.
- [22] N.C. Chou, C.K. Cheng. W.J. Dai and R. Lindelof, "Circuit Partitioning for Huge Logic Emulation systems", 31th DAC, pp. 244-249, 1994.
- [23] B. Krishnamurthy, "An Improved Min-Cut Algorithm for Partitioning VLSI Networks", IEEE Trans. on Computers, vol. 33, pp. 438-446, May, 1984.
- [24] G. Saucier, D. Brasen and J.P. Hiol,

“Partitioning with Cone Structures”,

ICCAD, pp. 236-239, 1993.

저 자 소 개



南 敏 祐(正會員)

1994년 서강대학교 전자계산학과 학사. 1996년 서강대학교 전자계산학과 석사. 현재 삼성전자 멀티미디어연구소 전임연구원



崔 然 景(正會員)

1991년 서강대학교 전자계산학과 학사. 1993년 서강대학교 전자계산학과 석사. 현재 서강대학교 컴퓨터학과 박사과정. 현재 경민대학 전자계산과 전임강사



林 鐘 錫(正會員)

1981년 서강대학교 전자공학과 학사. 1983년 한국과학기술원 전기전자 및 전자공학과 석사. 1989년 Univ. of Maryland 전기공학과 박사. 1989년 Univ. of Maryland 전기공학과 박사. 1983년 3월 ~ 1990년 8월 한국 전자통신연구소 연구원. 1990년 9월 ~ 현재 서강대학교 컴퓨터학과 부교수