

論文94-31A-10-16

동시 미로 배선 방법에 의한 새로운 FPGA 배선 방법

(A New FPGA Routing Method by Concurrent Maze Routing)

崔眞英*, 林鍾錫**

(Jin Young Choi and Chong Suck Rim)

要約

본 논문에서는 FPGA 배선을 위하여 기존의 전통적인 미로 배선 방법을 응용한 한번에 다수의 네트를 동시에 배선하는 새로운 배선 방법을 제안하고 이를 적용하여 symmetrical array 형태의 FPGA 배선에 사용할 수 있는 배선기 CMRF (Concurrent Maze Router for FPGA)를 소개한다. 제안한 배선 방법에서는 네트의 집합이 주어졌을 때 각각의 네트에 대하여 독립적으로 maze propagation과 backtracing을 수행한 후 네트 사이의 경쟁을 통하여 각 네트의 배선 경로를 동시에 선택한다. CMRF에서는 이러한 배선 방법을 사용하여 배선할 네트중에서 q(주어진 컴퓨터 환경에 따라 적당히 선택한 상수) 개의 네트씩 반복적으로 선택하여 이들을 동시 배선한다. 이러한 과정을 모든 네트의 배선이 완료되거나 또는 배선되지 않은 네트들에 대한 maze propagation이 모두 실패할 때까지 되풀이 한 후 들어내기 및 재배선 과정을 통하여 배선을 마무리 한다.

본 논문에서 제안한 배선 방법의 성능을 실험하기 위하여 임의로 생성한 10 개의 자료에 대하여 배선을 수행한 결과 q의 값이 클수록 배선률이 증가하였다. q인 경우 우리의 배선 방법은 일반적인 미로 배선과 유사하다. 그리고 우리의 CMRF를 Brown 등이 제안한 CGE 방법과 비교했을때 다섯 개의 모든 benchmark 자료에 대해서 CMRF가 CGE 방법에 비하여 보다 적은 갯수의 와이어 세그먼트를 갖는 연결 블록으로도 그들의 배선을 완료할 수 있었다.

Abstract

In this paper, we first propose a new FPGA routing method in which several nets are routed concurrently by applying the traditional maze routing method. We then introduce CMRF(Concurrent Maze Router for FPGA) which can be used for the routing of FPGAs of symmetrical array type by applying our new routing method. Given a set of nets, the proposed routing method performs the maze propagation and backtracing independently for each net and determines the routing paths concurrently by competition among nets. In CMRF, using this routing method, q nets are selected from the nets to be routed and they are routed concurrently, where q is the user given parameter determined by considering the computing environment. This process is repeated until either all the nets are routed or the remaining unrouted nets fail to their maze propagations. The routing of these nets are completed using the rip-up and rerouting technique.

We apply our routing method to ten randomly generated test examples in order to check its routing performance. The results show that as we increase the value of q, the routing completion rate increases for all the examples. Note that when q=1, our method is similar to the conventional maze routing method. We also compare CMRF with the CGE method which has been proposed by Brown et. al. For the five benchmark examples, CMRF complete the routing with less wire segments in each connection block than the wire segments needed in the CGE method for 100% routing.

*正會員, 金星社 中央 研究所 ASIC 센터 技術室
(ASIC Center, Central Research Lab.,
Gold Star Corp)

**正會員, 西江大學校 電子計算學科

(Dept. of Computer Science, Sogang Univ.)

接受日字 : 1993年 12月 22日

* 이 논문은 1993년도 교육부 학술 연구 조성비에 의하여 연구되었음.

1. 서론

FPGA(Field Programmable Gate Array)는 원하는 기능을 사용자 수준에서 칩에 프로그래밍할 수 있도록 구현되어 있는 레이아웃 형태이다.^[1] FPGA는 프로그래밍이 가능한 기본 모듈인 로직 블록(logic block)과 로직 블록간의 연결을 위한 배선 영역으로 구성된다. 로직 블록은 다양한 논리 함수를 구현할 수 있도록 기본 회로가 내재되어 있는 모듈로서 그 수가 정해져 있고 FPGA 전체에 규칙적으로 배치되어 있다. 배선 영역은 다양한 길이의 와이어 세그먼트(wire segment)와 사용자가 단락을 결정할 수 있는 와이어 세그먼트간의 연결을 위한 스위치(switch)로 이루어져 있다. 따라서 사용자는 칩 내의 로직 블록을 프로그래밍하고 이들의 핀(pin)간을 배선 영역에 주어진 와이어 세그먼트와 스위치를 통하여 연결함으로써 원하는 시스템을 구현할 수 있다.

FPGA는 사용자 수준에서 프로그래밍이 가능하도록 미리 기본 공정을 완료한 상태이기 때문에 그 구조가 다른 레이아웃 형태와는 달리 매우 독특하다. 우선 정해진 양의 로직 블록만을 가지고 있으며 각 로직 블록에 제한적인 논리 함수만을 구현할 수 있다. 또한, 배선 영역의 와이어 세그먼트와 스위치들 역시 그 양이 정해져 있고 로직 블록의 각 핀과 와이어 세그먼트 그리고 두 개의 세그먼트간의 연결 등이 자유롭지 않다. 따라서 FPGA에 적합한 설계 방법의 개발이 필요하게 되었고 본 논문에서는 FPGA 설계 과정 중 하나인 배선을 위한 새로운 방법을 제시하고자 한다.

FPGA는 로직 블록의 배치 구조와 배선 영역의 모양에 따라 여러가지 종류로 나눌 수 있는데^[1] 본 논문에서 다루는 FPGA 구조는 symmetrical-array 형태이다. Symmetrical-array 형태는 그림 1에 보인 바와 같이 CLB(configurable logic block)라고 불리는 로직 블록이 이차원 배열 구조로 규칙적으로 놓여 있으며 배선 영역이 이들 사이에 존재한다. CLB는 다양한 논리 회로를 필요에 따라 구현할 수 있는 FPGA의 기본 기능 모듈이다. 그림 1에서 보인 바와 같이 인접한 두 개의 CLB 사이에는 와이어 세그먼트들이 놓여 있는데 이 영역을 연결 블록(connection block)이라고 하며 CLB의 각 핀은 이 연결 블록 내의 스위치를 통하여 와이어 세그먼트와 연결된다. 그리고 와이어 세그먼트간의 연결은 스위치 블록(switch block)이라고 불리는 그림 1의 점선으로 둘러싸인 부분에 존재하는 스위치에 의해 이루어진다.

Symmetrical-array 형태를 갖는 상용의 FPGA로는 Xilinx의 3000 및 4000 계열의 FPGA 등을 들 수 있다.^[11] 이들의 배선 구조에는 그림 1에서의 와이어 세그먼트 외에도 long line, double line 또는 direct line 등으로 불리는 다른 형태의 와이어 세그먼트가 존재하나 본 논문에서는 이들을 다루지 않는다. 그런데 실제로 long line은 clock, reset 등과 같이 서로 연결하여야 할 핀들이 칩 전체에 널리 퍼져있는 네트를 위하여 존재하므로 이를 통한 배선은 따로 처리하는 것이 바람직하며 double line 또는 direct line 등은 필요한 경우 본 논문에서 제안한 배선 알고리즘의 배선구조에 쉽게 추가할 수 있다.

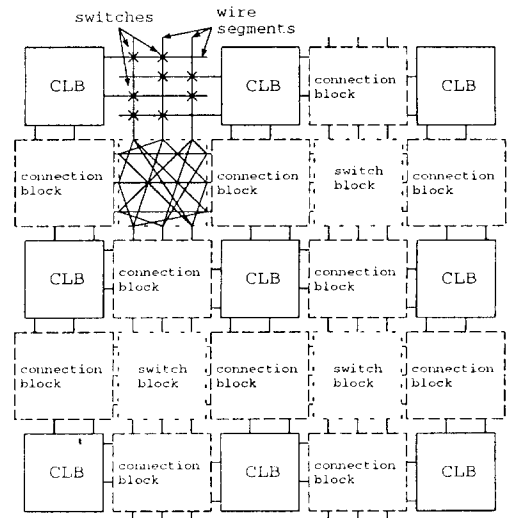


그림 1. Symmetrical-array 형태의 FPGA 구조
Fig. 1. FPGA architecture of symmetrical array type.

지금까지 기존의 PCB나 VLSI 설계를 위한 배선 문제를 해결하기 위해서 많은 배선 방법들이 개발되었는데^[6] 이러한 배선에서는 배선을 위한 수직 또는 수평선을 필요에 따라 임의로 만들 수 있다. 그러나 FPGA에서의 배선 문제는 각각의 네트가 로직 블록의 핀들의 집합으로 주어졌을 때 이들 핀들을 미리 주어진 적당한 와이어 세그먼트와 스위치를 사용하여 서로 연결하는 것이므로 이러한 방법은 적합하지 않다. 따라서 FPGA의 구조에 적합한 배선 방법의 개발이 필요하게 되었고 지금까지 이를 위하여 몇 가지 방법들이 제안되었다.^[1,2,7]

Symmetrical-array 형태의 FPGA 배선을 위해서 기존에 제안된 방법으로는 Brown 등이 제안한 CGE

(Coarse Graph Expansion) 방법^[2]과 Mikael Palczewski가 제안한 PPR(Plane Parallel Router)^[7] 등이 있다. CGE 방법에서는 먼저 모든 네트를 각각 두 개의 핀으로 구성된 네트들로 만들어서 이들에 대해 개략 배선^[8] (global routing)을 수행하고 그 결과를 이용하여 배선을 수행한다. 즉, 각 네트에 대한 개략 배선결과는 그 네트의 배선이 어떤 연결 블록과 스위치 블록들을 사용하여 배선하여야 할 지를 나타내며 이를 이용하여 개략 배선 경로에 포함된 모든 가능한 배선 경로를 나열한 확장 그래프(expanded graph)를 만든다. 다음, 확장 그래프의 루트 정점에서 단말 정점까지의 경로중 하나를 네트끼리의 상호 연관성을 고려하여 선택한다.

CGE 방법은 각 네트에 대해서 가능한 모든 배선 경로를 고려할 경우 대단히 많은 메모리를 요구하고 또한 네트들 사이의 경쟁을 하는 데에도 대단히 많은 시간을 요구한다. 이러한 문제 때문에 Brown 등은 모든 가능한 경로 중 일부분을 고려하여 배선 경로를 선택하되 만약 배선에 실패하면 좀더 많은 경로들을 고려하여 다시 배선 경로를 선택하는 방법을 사용하는데 이 경우 배선율이 떨어진다. 그리고 CGE 방법은 개략 배선 결과에 포함된 배선 경로만을 고려하기 때문에 배선에 실패했을 때 재배선이 대단히 어렵다. 또한 이 방법은 모든 네트를 각각 두 개의 핀으로 구성된 네트로 분할한 후 독립적으로 배선하기 때문에 필요 이상의 많은 와이어 세그먼트를 사용할 가능성이 있다 Palczewski에 의해 제안된 방법에서는 미로 배선 방법을 변형하여 배선을 수행한다. 전통적인 미로 배선 방법^[6]에 의하여 FPGA의 배선을 수행할 경우에는 하나의 와이어 세그먼트에서 스위치를 통하여 이것과 연결될 수 있는 모든 와이어 세그먼트를 탐색하는 형태로, 즉, 와이어 세그먼트 단위로 maze propagation을 수행한다. 그러나 Palczewski의 방법에서는 각 연결 블록의 와이어 세그먼트들을 동시에 고려하여 연결 블록 단위로 propagation을 수행한다. 즉, 하나의 핀에서 시작하여 그 핀이 속한 연결 블록의 연결 가능한 와이어 세그먼트들을 모두 마크(mark)하고 이 연결 블록과 스위치 블록을 통하여 도달할 수 있는 다른 연결 블록을 탐색한다. 만약 현재의 연결 블록에 마크되어 있는 와이어 세그먼트와 연결할 수 있는 다른 연결 블록의 와이어 세그먼트가 존재할 경우 그들을 모두 마크하고 같은 방법으로 계속 연결 블록 단위로 propagation을 수행한다. 이렇게 함으로써 전통적인 미로 배선 방법과 비교할 때 수행 시간을 보다 줄일 수 있으며 시작 핀에서부터 좀더 멀리 까지 propagation이 되도록 하였다. 그러

나 Palczewski가 제안한 방법은 수행 시간을 줄였지만 그보다 더 중요한 배선 가능성 항상 측면에서 볼 때 전통적인 미로 배선 방법이 갖는 문제점을 그대로 가지고 있다.

전통적인 미로 배선 방법이 갖는 문제점으로는 다음과 같은 두 가지를 들 수 있다. 먼저, 지금까지 알려진 대부분의 미로 배선 방법에서는 주어진 네트에 순서를 정하여 한번에 한 개씩 순서대로 네트를 배선한다. 이러한 방법은 일반적으로 먼저 배선이 이루어진 네트들의 배선결과가 나중에 배선될 네트의 배선에 방해줄 가능성이 있어 배선율을 낮추는 요인이 되며 따라서 배선될 네트의 순서를 정하는 것은 대단히 중요하다. 그러나 아직까지 이러한 네트의 순서를 정하는 방법이 확실히 알려져 있지 않아서 보통 길이가 짧은 네트 또는 긴 네트를 먼저 배선한다.^[6]

전통적인 미로 배선 방법이 갖는 또 다른 문제점으로는 배선 경로의 선택이다. 전통적인 미로 배선 방법에서는 하나의 네트에 대한 maze propagation을 완료한 후 backtracing에 의하여 길이가 가장 짧은 배선 경로를 선택한다. 그러나 이러한 배선 경로는 유일하지 않으며 일반적으로 다수가 존재하는데 이중 어느것을 선택하는가에 따라 차후 배선에 영향을 줄 수 있다. 이러한 영향을 고려하여 Brown 등^[3]은 미로 배선 방법을 개략 배선에 응용하면서 현재의 배선 밀도를 고려하여 propagation을 수행하고 이를 통하여 배선 경로를 선택하였고 Chiang 등^[4]은 min-max Steiner 트리를 새로 정의하여 이를 개략 배선에 적용하였다. 그러나 이들 방법 모두 네트를 하나씩 배선하는 형태이므로 이러한 문제들을 근본적으로 해결하는 것은 아니며 특히, 세부 배선(detailed routing)에서 이 문제들을 해결하기 위하여 본 논문에서와 같이 네트의 동시 배선을 시도한 방법은 찾아보기 어렵고 일반적으로 미배선된 네트가 존재할 경우 들어내기(rip-up)와 재배선(rerouting) 과정을 통하여 배선을 완료한다.^[6,9,10]

본 논문에서는 전통적인 미로 배선 방법이 갖는 이러한 문제점들을 해결하기 위하여 한번에 다수의 네트를 동시에 미로 배선하는 방법을 개발하여 이를 FPGA의 배선에 적용한 결과를 보인다. 이 방법에서는 먼저, 배선하고자 주어진 네트 중에서 동시에 배선할 q개의 네트를 선택한다. 여기서 q는 현재 사용하는 컴퓨터의 메모리의 크기 등을 고려하여 사용자가 결정한다. 다음, 이들 각각에 대해 독립적으로 maze propagation과 backtracing을 수행한 후 그 결과로 얻은 배선 가능한 최단 경로들 중에서 하나를 네트끼리의 경쟁을 통해 선택하도록 한다.

이러한 배선 방법은 q 개의 선택된 네트를 그 배선 순서를 정하지 않고 그들간의 경쟁을 통하여 배선하기 때문에 이들의 배선이 성공할 가능성이 크다. FPGA를 위해 실제로 구현한 배선기 CMRF(Concurrent Maze Router for FPGA)로 실험을 해본 결과 $q=1$ 로 즉, 한 개씩 네트를 배선하는 것 보다 q 를 충분히 크게 하여 가능한 많은 네트를 차례로 선택하여 동시에 배선하는 것이 보다 높은 배선률을 얻을 수 있었다. 여기서, $q=1$ 인 경우에도 CMRF는 배선 밀도를 고려하여 네트를 한 개씩 배선하므로 기존의 전통적인 미로 배선 방법 보다는 배선률이 높다. 또한 우리의 CMRF와 Brown 등^[2]의 CGE 방법과의 비교에서도 다수의 벤치마크 데이터에 대하여 100% 배선률을 위하여 CGE 방법이 필요로하는 배선 자원보다 상당히 적은량으로도 CMRF는 그 배선을 완료한다.

본 서론에 이어 2 장에서는 FPGA 배선을 위한 우리의 동시 미로 배선 방법을 설명하고 3 장에서는 이러한 동시 미로 배선 방법을 적용하여 개발한 배선기 CMRF에 대하여 기술한다. 그리고, 4 장에서는 CMRF를 다수의 벤치마크 자료에 적용한 결과를 보이고 마지막으로 5 장에서 결론을 맺는다.

II. FPGA를 위한 동시 미로 배선

본 장에서는 FPGA 배선을 위하여 본 논문에서 제안한 동시 미로 배선 방법에 대하여 기술한다.

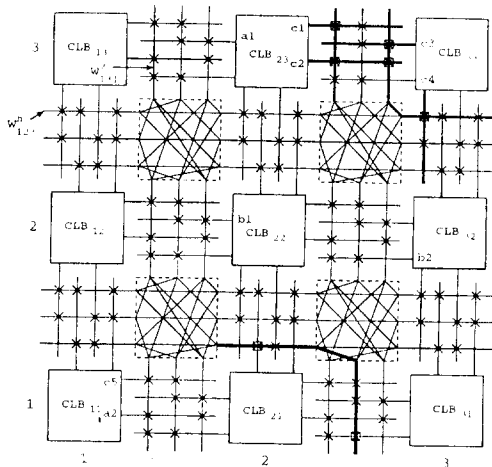


그림 2. 배선 방법을 설명하기 위한 예
Fig. 2. An example to explain our routing method.

본 장에서는 배선 방법의 설명을 위하여 그림 2에 보

인 예를 사용한다. 그림에서 굵은 선으로 표시한 와이어 세그먼트들과 정사각형으로 둘러싼 스위치들은 이미 배선에 사용한 것들이다. 본문에서 그림의 각 와이어 세그먼트들을 지칭하기 위하여 편의상 그들 각각에 고유의 이름을 붙인다. 먼저, 그림에서 보인 바와 같이 각 CLB를 그의 x, y 좌표에 따라 색인을 붙인다. 다음, 어떤 수평 와이어 세그먼트가 CLB_{ij} 위에 존재하는 연결 블록의 밑에서 k 번째 존재하는 와이어 세그먼트라면 이를 w_{ijk}^h 로 표시한다(예를들어 그림에서 w_{123}^h). 마찬가지로 어떤 수직 와이어 세그먼트가 CLB_{ij} 우측에 존재하는 연결 블록의 좌측에서 k 번째 존재하는 와이어 세그먼트라면 이를 w_{ijk}^v 로 표시한다(예를들어 그림에서 w_{31}^v). 그림 2에서는 세 개의 네트 $N_a = \{a_1, a_2\}$, $N_b = \{b_1, b_2\}$ 그리고 $N_c = \{c_1, c_2, c_3, c_4, c_5\}$ 가 배선을 위하여 주어지 있다.

일반적으로 배선하고자 하는 네트는 다중 핀(multi-pin) 네트이다. 이러한 다중 핀 네트의 배선에 미로 배선 방법을 적용할 경우 그 배선 방법은 다음과 같다.^[6] 먼저 그 네트에 속한 핀 하나를 시작점(source)으로 하고 나머지 다른 핀들을 목표점(targets)으로 하여 이중 하나를 만날때까지 maze propagation을 수행하고 backtracing에 의하여 시작점과 현재 만난 목표점간의 배선을 수행한다. 다음, 배선에 사용한 와이어 세그먼트들과 배선된 핀들을 모두 시작점으로 하고 같은 방법으로 이들 중 하나에 가장 가까운 핀까지 배선을 수행하며, 이러한 과정을 그 네트에 속한 모든 핀들이 서로 연결될 때까지 반복한다. 예를들어 그림 2에서 네트 N_c 는 이러한 방법으로 핀 c_1, c_2, c_3, c_4 간에 이미 배선이 완료된 상태이며 남은 일은 이들 핀들과 핀 c_5 와의 연결인데 이를 위한 시작점은 핀 c_1, c_2, c_3, c_4 외에도 와이어 세그먼트 w_{231}^h, w_{233}^h 그리고 w_{323}^h 등이 포함된다.

본 논문에서 소개하는 FPGA를 위한 배선기 CMRF에서도 이러한 배선 방법을 택하고 있으며, 따라서 각 네트의 미로 배선을 위한 시작점은 한 개 이상의 핀 또는 와이어 세그먼트들로 구성된다. 또한 다음 3 장에서 설명하는 바와 같이 CMRF에서는 배선에 실패한 네트를 들어내기 및 재배선을 통하여 그 배선을 완료하는데 이 때 들어내어진 네트들은 각각 그 배선 결과의 일부만이 제거된 후 재배선 된다. 이 경우 이러한 네트들의 배선을 위한 미로 배선에서는 시작점 뿐만 아니라 그 목표점도 한 개 이상의 핀 또는 와이어 세그먼트들로 구성된다.

배선하고자 하는 q 개의 네트와 그 각각에 대한 핀 또는 와이어 세그먼트들로 구성된 시작점과 목표점들의 집합이 주어졌다고 하자. 본 논문에서 제안한 알

고리증에서는 먼저 주어진 모든 네트에 대하여 각각 독립적으로 그의 목표점중 하나를 만날때까지 전통적인 미로 배선에서와 마찬가지로 방법으로 maze propagation을 수행한다.

주어진 q개의 네트중 하나를 N이라고 하자. 네트 N에 대한 maze propagation은 N의 시작점으로 표시된 모든 핀들과 와이어 세그먼트들을 0으로 마크함으로써 시작된다. 현재 i 값까지 마크를 했다면 i로 마크된 와이어 세그먼트와 연결 가능하고 다른 네트의 배선에 사용되지 않은 핀이나 와이어 세그먼트들중에서 아직 마크되지 않은 것들을 찾아 이들을 i+1로 마크한다. i+1로 마크된 핀이나 와이어 세그먼트들중에서 N의 목표점으로 표시된 것이 있으면 maze propagation을 중단하고 다음에 설명하는 backtracing 과정을 수행하며 그렇지 않으면 이러한 과정을 반복한다. 만약 i+1로 마크할 핀이나 와이어 세그먼트를 찾을 수 없다면 maze propagation을 중단하고 네트 N의 배선이 실패했음을 알린다. 또한, 마크값이 네트 N에 주어진 한계값을 넘어도 그 목표점을 만나지 못한다면 maze propagation을 중단하고 N의 배선이 실패했음을 알린다. 이것은 네트 N이 필요 이상으로 길게 배선되는 것을 방지하기 위함이다. Maze propagation에 실패한 네트의 배선은 다음 3장에서 설명한다.

Maze propagation에 성공한 네트, 즉 목표점을 만난 네트들에 대해서는 다시 backtracing을 각각 수행하여 본 논문에서 배선 그래프라고 부르는 방향 그래프(directed graph)를 구성한다. Backtracing은 maze propagation의 결과에 의존해서 이루어진다. 네트 N의 목표점중의 일부가 $d (> 0)$ 로 마크되어 있다고 하면 이러한 목표점으로부터 시작점을 향하여 backtracing을 시작한다. 먼저, $i=d$ 로 마크된 모든 목표점에 대하여 각각 하나씩의 정점을 생성한다. 목표점으로부터 시작하여 현재 마크가 $i (> 0)$ 인 핀 또는 와이어 세그먼트까지 방문하여 필요한 정점 및 방향에지를 생성하였다고 가정하자. 그러면 정점이 생성된 마크가 i 인 핀 또는 와이어 세그먼트와 연결이 가능하고 $i-1$ 로 마크된 모든 핀 또는 와이어 세그먼트들을 찾아 각각 하나의 정점을 생성하고 이들간의 연결 가능성을 방향에지로 표시한다. 즉, $i-1$ 로 마크된 핀 또는 와이어 세그먼트 w_1 과 i 로 마크된 핀 또는 와이어 세그먼트 w_2 와 연결이 가능하다면 w_1 에 해당하는 정점으로부터 w_2 에 해당하는 정점으로 방향에지를 연결한다. 다음, $i-1=0$ 이면 backtracing을 마치고 그렇지 않으면 $i=i-1$ 로 하고 위의 과정을 반복한다. 이렇게 maze propagation과 backtracing

을 통해 만들어진 그래프를 배선 그래프라고 부른다. 그림 3에 그림 2에 보인 네트들에 대하여 구성된 배선 그래프를 각각 보인다. 이러한 예에서 앞으로 우리는 네트 N_a, N_b, N_c 에 대한 배선 그래프를 각각 G_a, G_b, G_c 라고 하고 어떤 와이어 세그먼트 w 에 대응하는 정점이 $G_i(i=a,b,c)$ 에 존재한다면 이를 v_w^i 로 표시한다.

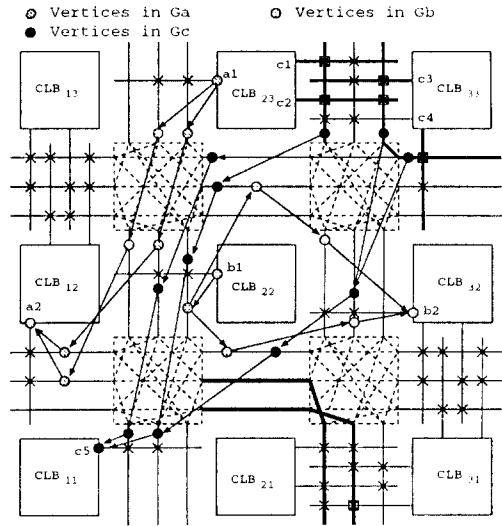


그림 3. 그림 2의 네트들에 대한 배선 그래프
Fig. 3. Routing graphs for the nets in Fig. 2.

배선 그래프의 정점들중 들어오는 에지의 수(in-degree)가 0인 정점 즉, 시작점에 해당하는 정점을 시작 정점(source vertex)이라고 하고 나가는 에지의 수(out-degree)가 0인 정점 즉, 목표점에 해당하는 정점을 목표 정점(target vertex)이라고 부른다. 배선 그래프의 임의의 시작 정점에서 목표 정점까지의 경로를 배선 경로라고 부른다. 네트 N에 대한 배선 그래프의 임의의 배선 경로는 N의 한 시작점으로부터 목표점까지의 최단거리 배선을 나타낸다. 즉, 배선 경로상의 정점에 해당하는 와이어 세그먼트와 에지에 해당하는 스위치를 이용하여 네트 N을 배선할 수 있다. 특히, 구성된 배선 그래프는 전통적인 미로 배선 방법에 의하여 생성되었기 때문에 배선 그래프의 모든 배선 경로의 길이는 동일하다.

하나의 배선 그래프의 시작 정점에서 동일한 거리에 있는 정점들의 집합을 동일 거리 정점의 집합이라고 하고 이에 속한 정점에 부여된 마크값이 h 라면 이 집합을 거리 h 의 동일 거리 정점의 집합이라고 한다. 예를들어 그림 3에서 G_c 의 정점의 집합 $\{v_{a1}^{c1}, v_{a2}^{c2}, v_{a3}^{c3}\}$

은 거리 2의 동일 거리 정점의 집합이다. 어떤 네트 N에 대한 목표점들이 maze propagation 후에 d로 마크되었다면 네트 N의 배선 그래프에는 모두 d+1개의 동일 거리 정점의 집합이 존재한다. 또한 배선 그래프의 에지는 거리 차이가 1인 두 동일 거리 집합의 정점 사이에만 존재한다. 따라서, 배선 그래프의 임의의 배선 경로는 각 동일 거리 정점의 집합에서 정확히 한 개씩의 정점을 포함하고 있다.

주어진 q개의 네트에 대한 배선은 이들에 대하여 각각 구성된 배선 그래프에서 한 개씩의 배선 경로를 선택하는 것과 동일하다. 그런데 두 개의 서로 다른 네트의 배선에 하나의 와이어 세그먼트를 동시에 사용할 수 없으므로 각 와이어 세그먼트에 대응하는 정점은 배선 경로에 많아야 한번만 포함되어야 한다. 본 장의 나머지 부분에서는 이러한 조건을 만족하는 배선 경로 선택 방법을 설명한다.

주어진 q개의 네트에 대한 배선 그래프는 각각 독립적으로 구성된다. 따라서 서로 다른 두 개의 배선 그래프가 동일한 와이어 세그먼트에 대응하는 정점을 각각 포함할 수 있다. 이 경우 우리는 두 배선 그래프가 서로 겹친다고 말한다. 만약 q개의 배선 그래프가 서로 겹치지 않았다면 각 배선 그래프에서 임의로 하나의 배선 경로를 선택함으로써 배선을 수행할 수 있다. 그러나 서로 겹친 배선 그래프에 대해서는 배선 경로를 임의로 선택할 수 없다. 따라서, 우리는 먼저 구성된 배선 그래프에서 적당한 정점을 제거하여 이들이 서로 겹치지 않도록 한다. 예를 들어 그림 3에서 배선 그래프 G_b 와 G_c 에 와이어 세그먼트 w_{123}^* 에 대응하는 정점 v_b^{*123} 와 v_c^{*123} 가 각각 포함되어 있으므로 G_b 와 G_c 는 서로 겹쳐있다. 따라서 이 두 정점중 하나를 제거하여 배선 그래프 G_b 와 G_c 간의 겹침을 제거한다. 이 밖에도 와이어 세그먼트 w_{222}^h , w_{213}^h 그리고 w_{222}^* 등에 해당하는 정점들이 배선 그래프 G_b 와 G_c 에 각각 포함되어 있으므로 이들중 하나씩을 배선 그래프에서 제거하여 겹침을 제거하여야 한다.

배선 그래프의 겹침을 제거하는 과정은 이를 위한 가중치를 정의하여 배선 그래프의 모든 정점에 부여하고 가중치가 큰 정점부터 배선 그래프의 겹침이 모두 제거될 때까지 하나씩 제거하는 방법을 사용한다. 이를 위하여 먼저 모든 배선 그래프의 각 정점의 가중치를 0으로 초기화 하고 아래에 설명하는 방법에 따라 정점의 가중치를 계산한다.

어떤 배선 그래프 G에서 하나의 정점 v가 속해 있는 동일 거리 정점의 집합을 E라고 하고 $|E|=h$ 라고 하자. 만약 $h>1$ 이면 배선 그래프 G에서 v를 제거하더라도 E에는 h-1 개의 정점이 남게되고 따라서 G-

h에는 최소한 h-1 개의 서로 다른 배선 경로가 여전히 존재한다. 즉, 배선 그래프 G에서 선택할 배선 경로에 정점 v가 포함되어야 할 필요도는 h가 클수록 작아져 대략 h-1에 반비례한다고 할 수 있다. 그런데 만약 h가 1이라면 배선 그래프 G에서의 모든 배선 경로는 정점 v를 포함하므로 이의 필요도는 ∞ 라고 할 수 있다. 예를 들어 그림 3에서 와이어 세그먼트 w_{123}^* 에 대응하는 G_c 의 정점 v_c^{*123} 가 속한 동일 거리 정점의 집합은 $\{v_c^{*123}, v_c^{*123}, v_c^{*123}\}$ 으로 그 크기가 3인데 비하여 같은 와이어 세그먼트에 대응하는 G_b 의 정점 v_b^{*123} 는 그 자체만으로 G_b 에서 하나의 동일 거리 집합이 된다. 그런데, G_b 와 G_c 간의 겹침 제거에서 정점 v_b^{*123} 가 제거될 경우 G_b 에는 가능한 배선 경로가 존재하지 않게 되므로 가능한한 제거되어서는 안되며 따라서 v_b^{*123} 의 가중치는 ∞ 로 설정한다. 한편, 정점 v_c^{*123} 의 경우에는 비록 이 정점이 제거되더라도 G_c 에는 여전히 최소한 두 개의 배선 경로가 존재하며 따라서 정점 v_c^{*123} 의 가중치는 이 갯수의 역인 1/2로 설정한다. 이러한 가중치 설정 방법을 일반화 하면 다음과 같다.

어떤 와이어 세그먼트 w에 대응하는 정점을 포함한 배선 그래프를 G_1, G_2, \dots, G_m 이라고 하고, $G_i (1 \leq i \leq m)$ 에 포함된 이러한 정점을 v_i^* 라고 하며 이 정점이 속해있는 동일거리 정점의 집합의 크기를 h_i 라고 하자. 우리는 각 정점 $v_i^* (1 \leq i \leq m)$ 에 대하여 가중치 $1/(h_i-1)$ 을 다른 모든 정점 $v_j^* (1 \leq i \leq m, j \neq i)$ 에 더한다. 즉, v_i^* 의 가중치는 $\sum_{1 \leq k \leq m, k \neq i} 1/(h_k-1)$ 이며 이는 G_i 에서 하나의 배선 경로를 선택할 때 다른 배선 그래프의 배선 경로 구성에 필요한 와이어 세그먼트를 가능한한 포함시키지 않기 위함이다. 만약 정점 v_i^* 가 겹침 제거를 위하여 G_i 에서 제거된다면 다른 정점 $v_j^* (j \neq i, 1 \leq j \leq m)$ 의 가중치에서 $1/(h_i-1)$ 을 뺀다.

배선 그래프의 겹침 제거는 다음과 같은 과정에 의한 것이다. 먼저 가중치가 가장 큰 정점 v를 그것에 연결된 에지들과 함께 그들이 속한 배선 그래프에서 제거한다. 그리고 v와 인접하고 그 마크값이 v보다 하나 작은 정점중에서 이러한 제거 결과로 나가는 에지가 존재하지 않게 된 정점들 역시 제거하며 이러한 작업을 반복적으로 더이상 제거할 정점이 없을 때까지 계속한다. 마찬가지로 v와 인접하고 그 마크값이 v보다 큰 정점중에서 들어오는 에지가 존재하지 않게된 정점들을 반복적으로 제거한다. 이와같이 추가로 정점을 제거하는 이유는 이러한 정점들은 더이상 배선 경로에 포함될 수 없기 때문에 이들을 미리 제거하여 겹침 제거 작업을 보다 빨리 끝내기 위해서이다. 겹

침 제거는 모든 정점의 가중치가 0 이 될 때까지 위의 과정을 반복함으로써 이루어 진다. 그런데, 이러한 과정에서 어떤 배선 그래프는 그의 모든 정점이 제거될 수가 있는데 이 경우에는 그 배선 그래프에 대한 네트의 배선을 포기한다. 이러한 과정을 그림 3에 적용한 결과를 그림 4에 보인다.

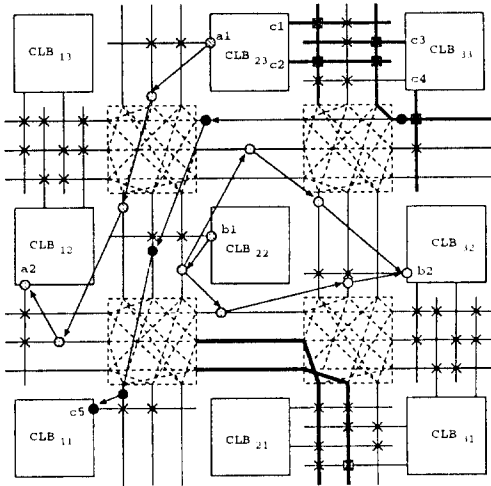


그림 4. 그림 3의 배선 그래프에 대한 겹침 제거 결과
 Fig. 4. The overlap removal results among the routing graphs in Fig. 3.

그런데, 이 예에서는 와이어 세그먼트 w_{123}^v 에 대응하는 정점들중 하나를 먼저 제거하면(즉, G_c 의 정점 $v^{(a)}$ 의 제거) 동일한 가중치를 갖는 정점들이 많아서 이들의 제거에 여러가지 선택이 있을 수 있으나 그림 4는 앞으로의 설명을 위하여 가장 적당한 겹침 제거 결과 하나를 보인 것이다.

배선 그래프 사이의 겹침을 제거한 후에는 각 배선 그래프에서 하나씩의 배선 경로를 선택한다. 그런데 이러한 배선 경로는 임의로 선택할 수 있지만 차후 다른 네트의 배선을 쉽게하기 위하여 연결 블록의 배선 밀도를 고려하여 선택한다. 이를 위하여 배선 밀도를 고려한 가중치를 각 정점에 부여한다. 즉, 어떤 배선 그래프의 한 정점 v 에 대응하는 와이어 세그먼트가 속한 연결 블록에 t 개의 와이어 세그먼트가 존재하고 이중 p 개가 이미 배선에 사용되었다면 v 의 가중치는 p/t 로 한다. 이러한 가중치에 의하여 그 값이 가장 큰 정점을 선택하여 모든 배선 그래프에 정확히 각각 하나의 배선 경로가 남을 때까지 계속 하나씩 제거한다. 그런데 선택된 정점이 이것이 속한 동일

거리 정점의 집합의 유일한 정점이라면 이 정점은 배선 경로에 꼭 포함되어야 하므로 제거하지 않는다. 그리고 앞의 겹침 제거에서와 마찬가지로 하나의 정점을 제거할 때마다 그 정점을 제거함으로써 배선 경로에 포함될 수 없는 다른 정점들도 역시 모두 제거한다.

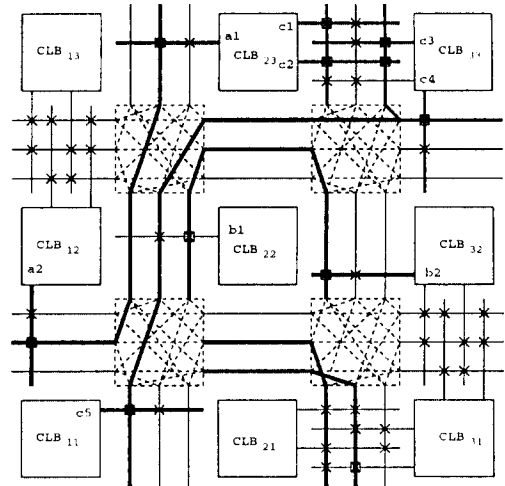


그림 5. 그림 2의 예에 대한 최종 배선 결과
 Fig. 5. The final routing result for the example in Fig. 2.

배선 시간을 줄이기 위하여 이러한 배선 경로 선택 과정에서는 앞의 겹침 제거 과정에서와는 달리 하나의 정점을 제거할 때마다 매번 다른 정점들의 가중치를 갱신하지는 않는다. 실제로 정점들의 가중치는 다음과 같이 갱신한다. 만약 현재 가장 큰 가중치 값을 갖는 정점이 이것이 속한 동일 거리 정점의 집합의 유일한 정점이라면 이 정점에 대응하는 와이어 세그먼트는 앞으로 배선에 사용될 것이 분명하다. 따라서 이 와이어 세그먼트가 속한 연결 블록의 다른 와이어 세그먼트에 대응하는 정점들이 있을 경우 이들의 가중치를 $1/t$ 만큼 높인다.

이러한 배선 경로 선택 방법을 그림 4에 적용하면 네트 N_b 에 대한 배선 경로중 와이어 세그먼트 w_{123}^h 에 대응하는 정점을 지나는 배선 경로가 제거되어 그림 5와 같은 배선 결과를 얻는다. 그런데, 그림에서 보이는 바와 같이 CLB_{21} 과 CLB_{22} 의 위에 존재하는 연결 블록에는 각각 한 개의 사용되지 않은 와이어 세그먼트가 존재하게 되는데 이와 같이 우리의 배선 경로 선택 방법은 배선을 골고루 분산시켜주는 효과가 있다.

III. CMRF

본 장에서는 앞의 II 장에서 기술한 동시 미로 배선 방법을 사용한 FPGA를 위한 새로운 배선기 CMRF에 대하여 기술한다. CMRF는 C언어로 구현되었으며 그 흐름도는 그림 6에 보인 바와 같이 모두 6개의 과정으로 되어 있다. 이중 maze propagation과 backtracing을 수행하여 배선그래프를 구성하는 과정 그리고 배선 그래프 사이의 겹침을 제거하는 과정과 배선 경로를 선택하는 과정 등은 앞 장에서 설명하였으며 따라서 본 장에서는 이들을 제외한 나머지 과정에 대하여 설명한다.

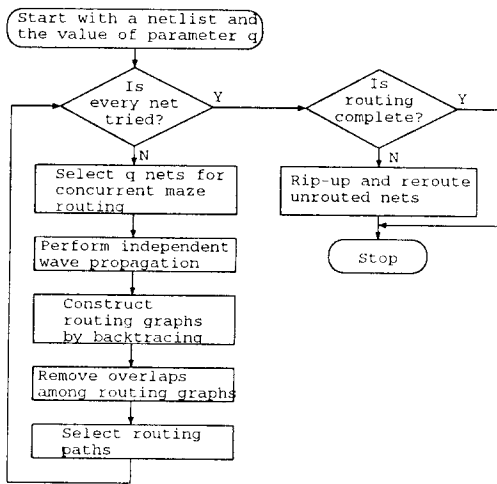


그림 6. CMRF의 주요 흐름도
Fig. 6. Flowchart of CMRF.

CMRF에서는 먼저 현재 미배선된 네트중에서 한번에 q 개씩의 네트를 선택하여 이들에 대한 동시 미로 배선을 수행한다. 이를 위한 준비 작업으로서 선택된 네트에 대하여 각각 시작점과 목표점을 결정하는 작업이 선행된다. 이러한 동시 미로 배선 과정은 모든 네트의 배선이 완료되었거나 또는 현재 배선이 아직 완료되지 않은 네트들에 대한 maze propagation이 모두 실패할 때까지 계속 반복된다. 그리고 이 과정에서 maze propagation에 실패한 네트들은 들어내기 및 재배선 과정에 의하여 배선을 시도하는데 이 과정은 나중에 설명한다. 또한 다중핀 네트의 경우에는 이 네트가 현재 선택되어 동시 미로 배선 과정에서 그 배선에 성공하였다고 하더라도 이 네트에 대한 배선이 완료된 경우가 아니면 나중에 다시 선택되어 그 배선이 완료되도록 한다.

동시 미로 배선을 위하여 선택되어지는 네트의 갯수인 q 값은 1부터 네트의 총 갯수 사이의 어떠한 값도 가질 수 있다. 만약 q 가 1이면 네트를 하나씩 선택하여 미로 배선하는 전통적인 배선 방법과 유사하다고 할 수 있다. 다만 전통적인 방법과 한가지 다른 점은 우리의 배선 방법은 현재 배선중인 네트에 대한 배선 경로를 선택하는데 있어서 그 배선 경로의 와이어 세그먼트가 속한 연결 블럭의 배선 밀도를 고려한다는 점이다. 이것은 Palczewski의 방법에서도 고려되지 않았다.^[7]

한편, 만약 q 가 주어진 네트의 총 갯수라면 동시에 모든 네트를 배선하고자 함을 의미한다. 이 경우 네트간의 배선 순서를 고려할 필요가 없어 배선물은 향상될 것이나 배선에 필요한 정보를 저장하는데 많은 메모리가 요구되며 네트간의 상호 관계를 조사하는 등의 일이 많아져 오랜 수행시간이 필요하게 된다. 따라서, q 값은 주어진 컴퓨터 환경에 따라 적절히 선택한다. 실제로 16M 메모리를 장착한 SUN SPARC IPX에서 CMRF를 수행시킬 때 q 값은 입력 자료에 따라 차이가 있지만 대체로 20정도가 가장 적절하였다.

주어진 컴퓨터 환경을 고려할 때 일반적으로 q 값은 주어진 네트의 갯수 보다 상당히 작다. 이 경우 어떤 네트들을 먼저 선택하여 동시 배선할 것인가가 높은 배선률을 위하여 대단히 중요하다. 만약 선택된 네트들이 서로의 배선 결과에 영향을 주지 않는다면 배선시 네트 상호간의 연관성을 고려하여 배선하는 우리의 동시 미로 배선 방법은 전통적인 미로 배선 방법과 비교해서 별로 이익이 없다. 따라서 우리는 핀들이 서로 한곳에 모여있는 네트들을 가능한한 선택함으로써 배선중에 네트 상호간에 많은 영향을 주도록 한다. 즉, 각 네트에 속한 핀들의 중심점을 구하고 그 중심점들이 한데 모여있는 q 개의 네트들을 선택한다.

어떤 네트 N 이 maze propagation에 실패했다면 그것은 그 네트의 시작점과 목표점간에 연결 가능한 배선 경로가 존재하지 않음을 의미한다. 이 경우 일반적으로 이미 배선된 다른 네트의 배선 결과를 들어내어(rip-up) 네트 N 의 배선을 위한 경로를 확보하여 N 을 배선한 후 들어낸 네트들을 재배선(reroute)하는 방법을 사용한다.^[6] 우리의 CMRF에서도 배선에 실패한 네트들에 대하여 이러한 들어내기 및 재배선 과정을 통하여 배선을 완료하는데 앞으로 이 과정을 간단히 설명한다.

Maze propagation에 실패하여 들어내기 및 재배선에 의하여 배선하고자 하는 네트를 N 이라고 하자. 네트 N 과 들어낸 네트들을 성공적으로 배선하기 위

해서는 들어낼 네트들을 선택할 때 그들의 재배선 가능 여부를 고려하여야 한다. 그러나 이를 고려하기는 대단히 어려워 본 논문에서는 가능한 적은 수의 네트를 들어내도록 하였다. 먼저 들어낼 네트를 선택하기 위하여 앞의 2 장에서 설명한 바와 같이 네트 N에 속한 핀과 와이어 세그먼트를 각각 시작점과 목표점으로 분류하여 표시하고 시작점으로부터 maze propagation을 수행한다. 그런데 이 경우에는 2 장에서와는 달리 와이어 세그먼트가 다른 네트의 배선에 사용되었는지의 여부에 관계없이 maze propagation을 수행한다.

Maze propagation을 수행한 후에는 2 장에서 설명한 방법과 마찬가지로 backtracing에 의하여 배선 그래프를 만든 후 그래프의 각 에지에 다음과 같은 방법으로 가중치를 부여한다. 배선 그래프의 한 에지 $e=v_i \rightarrow v_j$ 를 고려하자. 여기서, w_i 와 w_j 를 각각 정점 v_i 와 v_j 에 대응하는 와이어 세그먼트라고 하자. 만약, w_i 와 w_j 가 각각 서로 다른 네트의 배선에 사용되었다면 에지 e 의 가중치를 2로 하고 동일한 네트의 배선에 사용되었다면 이의 가중치를 1로 한다. 또한, 만약 w_j 가 아직 배선에 사용되지 않았다면 e 의 가중치는 0으로 한다. 이러한 방법을 배선 그래프의 모든 에지에 적용하여 그들의 가중치를 결정한다. 이러한 가중치는 가능한 적은 수의 네트만이 그들의 배선에 사용한 와이어 세그먼트들을 포기하는 효과가 있다.

네트 N에 대한 배선 그래프의 모든 에지에 대한 가중치가 결정되면 배선 그래프의 임의의 시작점에서 목표점에 이르는 최단거리 경로를 찾아 이 경로의 정점들에 대응하는 와이어 세그먼트 중에서 다른 네트의 배선에 사용된 와이어 세그먼트들을 그 배선 결과로부터 제거한다. 그리고 제거된 각 와이어 세그먼트를 배선에 사용하였던 네트의 배선 그래프를 조사하여 이 와이어 세그먼트에 대응하는 정점을 제거한 후, 이 정점에 인접한 정점중 시작 또는 목표 정점이 아니고 단 한 개의 에지만이 연결되어 있을 경우 이들과 이들에 대응하는 와이어 세그먼트 역시 제거하며, 이러한 과정을 더이상 제거할 세그먼트가 존재하지 않을 때까지 반복한다. 이러한 과정을 거친 후 네트 N과 배선 결과의 일부가 들어내어진 네트들을 우리의 동시 미로 배선 방법을 사용하여 함께 배선한다. 만약 이 과정에서 배선에 실패한 네트가 존재하면 차후에 이 네트에 대하여 들어내기 및 재배선 과정을 다시 수행한다. 우리의 CMRF에서는 하나의 네트에 대하여 최대 5번까지 들어내기 및 재배선 과정을 시도할 수 있도록 하였으며 그 이후에도 배선에 실패하면 배선이 불가능하다고 판단하여 이 사실을 출력한다.

IV. 실험 결과

본 장에서는 여러 실험 입력에 대하여 우리의 CMRF를 사용하여 배선한 결과를 보인다. CMRF는 C언어로 작성되었으며 실험은 SUN SPARC

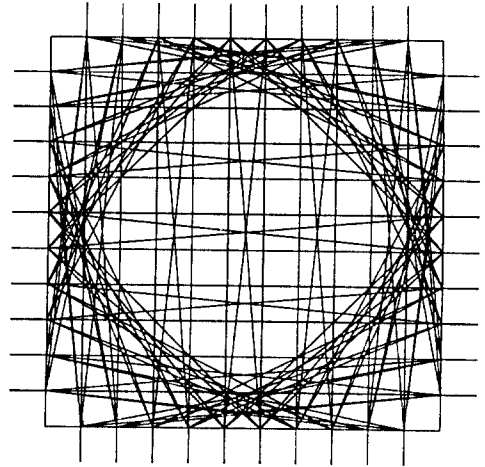


그림 7. q값에 따른 배선률 향상을 조사하기 위하여 사용한 스위치 블럭의 구조
Fig. 7. The switch block structure used in our routability testing for the various values of q.

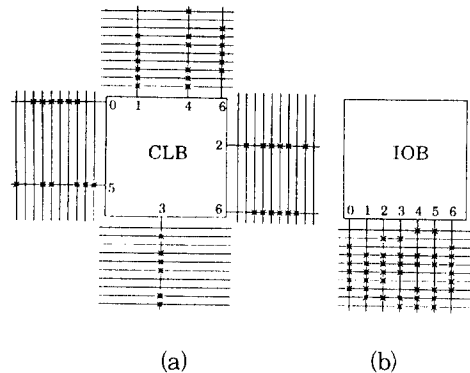


그림 8. Q 값에 따른 배선률 향상을 조사하기 위하여 사용한 연결 블럭의 구조 (a) CLB와의 연결 (b) IOB와의 연결
Fig. 8. The connection block structure used in our routability testing for the various values of q. (a) Connections with CLB, (b) Connections with IOB.

station IPX에서 수행하였다.

먼저 우리의 동시 미로 배선 방법에서 동시에 처리하는 네트의 갯수 q 의 영향을 알아보기 위하여 임의로 생성한 각각 120 개의 two-terminal 네트로 구성된 10 개의 실험 입력에 대하여 q 값을 각각 1, 20 그리고 40 으로 했을 때의 배선률을 조사하였다. 이 실험은 우리의 동시 미로 배선 방법의 우수성을 조사하기 위한 것이므로 CMRF에서 들어내기 및 재배선 과정을 생략하고 단지 그 전 단계인 모든 네트에 대한 동시 미로 배선 과정을 수행한 후의 배선률을 조사하였다. 실험에 사용한 FPGA의 구조는 100 개의 CLB가 10×10 배열을 이루고 있는 형태이고 그 사이의 연결 블럭에는 각각 10 개의 와이어 세그먼트가 존재한다. 또한, 연결 블럭과 스위치 블럭의 구조는 Brown 등^[2]이 사용한 형태로서 그림 7과 그림 8에 각각 보인다. 여기서 그림 8의 (a)와 (b)는 각각 CLB와 IOB(input output block)의 각 핀이 연결 블럭의 어떤 와이어 세그먼트와 연결이 가능한지를 보여주고 있으며 CLB와 CLB 그리고 CLB와 IOB 사이에 존재하는 연결 블럭은 이들을 조합하여 구성한다. 예를들어 그림 9에 이러한 FPGA 좌상부의 배선구조를 보인다.

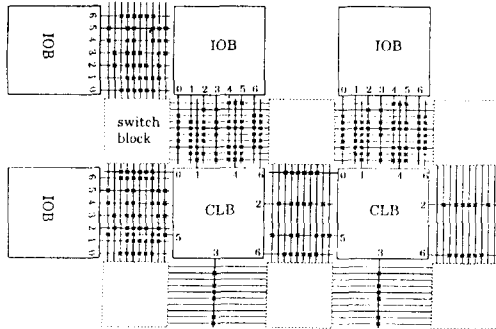


그림 9. FPGA의 좌상부 구조

Fig. 9. The structure of the left upper part of the FPGA.

표 1에 이러한 우리의 실험 결과를 보인다. 표에서 보인 바와 같이 모든 입력 자료에 대해서 S가 클수록 배선율이 높아졌다. 그리고, 이 실험에서의 평균 배선율은 S=1일 때 91.3%, S=20일 때 93.4%, S=40일 때 96.0%이다. 여기서, S=1인 경우에도 네트를 한 개씩 배선하지만 각 연결 블럭의 배선 밀도를 고려하여 배선 경로를 각각 선택하므로 기존의 전통적인 미로 배선 방법^[6]에 비하여 그 배선률이 더 높으리라고 짐작된다. 따라서, 본 논문에서 제시하는

동시 미로 배선 방법은 네트를 하나씩 배선하는 기존의 미로 배선 방법에 비하여 우수함을 알 수 있으며, 이러한 높은 배선률은 차후 들어내기 및 재배선 과정에서의 노력을 줄일 수 있으므로 일반적으로 전체 배선을 보다 효율적으로 수행하여 주는 효과가 있다.

표 1. $q = 1, q = 20, q = 40$ 일 때 결과
Table 1. Completion rates for $q = 1, q = 20, q = 40$.

	q=1		q=20		q=40	
	배선된 네트 수	배선율 (%)	배선된 네트 수	배선율 (%)	배선된 네트 수	배선율 (%)
Data1	108	90.0	112	93.3	117	97.5
Data2	106	88.3	110	91.7	115	95.8
Data3	110	91.7	112	93.3	115	95.8
Data4	113	94.2	113	94.2	118	98.3
Data5	110	91.7	112	93.3	114	95.0
Data6	107	89.2	111	92.5	112	93.3
Data7	112	93.3	115	95.8	116	96.7
Data8	108	90.0	109	90.8	113	94.2
Data9	112	93.3	115	95.8	116	96.7
Data10	110	91.7	112	93.3	117	97.5

이제 Brown 등이 제안한 CGE 방법과 우리의 CMRF에 의한 결과를 서로 비교한다. W 를 각 연결 블럭에 주어진 와이어 세그먼트의 갯수라고 하자. 그리고, F_c 를 CLB의 각 핀에서 스위치를 통하여 연결 가능한 와이어 세그먼트의 갯수라고 하고 F_s 를 하나의 스위치 블럭에서 한 와이어 세그먼트와 연결 가능한 와이어 세그먼트의 갯수라고 하자. Brown 등은 이러한 세 개의 변수를 사용하여 symmetric array 형태의 FPGA의 배선 구조를 매개 변수화 하였다.^[2] 즉, 그들의 CGE 방법을 실험하기 위하여 구현한 FPGA 배선 프로그램을 보면 CLB의 총 갯수, W , F_c 그리고 F_s 의 값이 주어지면 프로그램 내에서 이에 따른 연결 블럭 및 스위치 블럭을 구성함으로써 FPGA 배선 구조를 자동 생성하며 따라서 FPGA 배선 방법의 실험에 매우 효과적이다. 예를들어 앞의 q 값에 따른 배선률을 조사하기 위하여 사용한 그림 7과 그림 8의 스위치 블럭과 연결 블럭은 $W=10$, $F_s=6$ 그리고 $F_c=6$ 인 경우에 해당한다.

실험에 사용한 다섯 개의 자료는 Rose로부터 제공 받은 것으로 그들의 주요 사양은 표 2에 보인바와 같다. 여기서, 각 자료에 대하여 네트의 총 갯수를 N 이라 하고 모든 네트의 핀 수를 t_1, t_2, \dots, t_N 이라 할 때, connection 수는 $\sum_{i=1}^N (t_i - 1)$ 이며 이들은 각각 하나의 네트를 완전히 배선하는데 필요한 최소의 미

로 배선 시도 횟수를 나타낸다. 그리고, 우리의 실험에서는 한번의 동시 미로 배선하기 위하여 선택할 네트의 갯수인 q 의 값으로 20을 사용하였는데 이는 실험에 의하여 가장 적당한 값으로 결정하였다.

표 2. Rose의 benchmark 자료
Table 2. Benchmark data by Rose.

	CLB 갯수	네트 수	Connection 수
BUSC	13×12	151	392
DMA	18×16	213	771
BNRE	22×21	352	1257
DFSM	23×21	420	1422
Z03	27×26	608	2135

표 3. 100% 배선을 위한 최소의 W값
Table 3. Minimum value of W for 100% routing.

	CMRF($q=20$)	CGE
BUSC	8	11(10)
DMA	8	12(10)
BNRE	10	14(12)
DFSM	8	13(10)
Z03	11	15(13)

CMRF와 CGE 방법과의 비교를 위하여 먼저 수행한 실험은 F_s 와 F_c 가 주어졌을때 각 benchmark 자료의 네트들을 100% 배선하는데 필요한 각 연결 블럭내의 와이어 세그먼트의 갯수 W 의 최소값이다. $F_s=6$ 이고 $F_c=0.6W$ 인 경우에 대한 실험 결과는 표 3에 보인 바와 같다. 그런데 표 3에서 보인 바와 같이 CGE 방법에 의한 결과에는 Rose로부터 제공받은 프로그램을 실제로 수행하여 얻은 결과와 참고문헌 [2]에서 제시한 W 값과는 다소 차이가 있다.¹⁾ 그 이유는 알 수 없으나, 어쨌든 우리의 CMRF가 CGE 방법보다 작은 W 값(즉, 보다 적은 수의 와이어 세그먼트)으로도 그 배선을 완료할 수 있음을 알 수 있다.

표 4는 $F_s=6$, $F_c=0.6 \times W$ 에 대하여 CGE 방법에서 100% 배선 가능한 최소의 W 값을 사용했을때 우리의 CMRF와 CGE 방법에 의한 결과를 비교한

것이다. 표에서 보인 바와 같이 CMRF에 의한 배선 결과가 CGE 방법에 의한 결과보다 배선에 사용된 와이어 세그먼트의 갯수가 적다. 또한, 배선을 수행하는데 있어서 CMRF는 CGE 방법에 비하여 매우 적은량의 메모리를 사용한다. 수행시간의 관점에서 보면 CMRF가 몇 개의 입력 자료에 대하여 CGE보다 많은 시간이 필요한데, CGE 방법을 사용하기 위하여서는 그 전 단계로 개략배선 과정이 필요하다는 것을 감안하여야 한다. 그런데 만약 W 값이 작아서 배선이 어려운 경우에는 CGE 방법은 CMRF보다 훨씬 많은 수행시간을 요구한다(표 4에서 BNRE). 참고로, CGE 방법에 대한 실험은 Rose^[2]로부터 그들의 프로그램을 제공받아 CMRF의 실험에 사용한 것과 동일한 컴퓨터에서 수행하였다.

표 4. CGE 방법과 CMRF와의 비교
Table 4. Comparative results by the CGE method and our CMRF.

	W	CGE			CMRF		
		사용한 wire segment 갯수	수행 시간 (sec)	사용한 memory (kbytes)	사용한 wire segment 갯수	수행 시간 (sec)	사용한 memory (kbytes)
BUSC	11	1181	14.4	2564	1056	36.7	728
DMA	12	2354	23.8	4000	2216	60.9	1561
BNRE	14	4009	471.1	14735	3648	181.6	3125
DFSM	13	4250	241.9	9260	4041	223.0	3819
Z03	15	7132	440.9	17788	6775	471.2	6133

표 5. 들어내기 및 재배선 단계 이전까지 CMRF에 의한 결과

Table 5. The routing results by CMRF without the routing completion using the rip-up andrerouting stage.

	W	연결에 성공한 connection 수	배선에 사용된 wire segment 갯수	수행 시간 (sec)
BUSC	8	388	1106	22.1
DMA	8	764	2271	56.3
BNRE	10	1247	4070	161.6
DFSM	8	1407	4234	184.4
Z03	11	2124	7033	442.8

마지막으로 $F_s=6$, $F_c=0.6 \times W$ 인 경우 우리의 CMRF로 100% 배선이 가능한 최소의 W 에 대하여 배선을 수행한 결과를 표 5와 표 6에 보인다. 여기서, 표 5는 들어내기 및 재배선 이전까지의 결과이고 표 6는 최종 결과이다. 이러한 결과의 한 예로서 그림 10에 BUSC의 배선 결과를 보인다.

1) CGE 방법의 경우, 각 자료에 대한 첫번째 W 값은 Rose로부터 제공받은 프로그램을 실제로 수행하여 얻은 결과이고 괄호안의 값은 참고 문헌 [2]에서 제시한 결과이다.

표 6. CMRF에 의한 최종 배선 결과
Table 6. The final routing results by CMRF.

	배선에 사용된 wire segment 갯수	재배선 횟수	수행 시간 (sec)	Memory (kbyte)
BUSC	1131	11	25.9	936
DMA	2308	17	58.7	1730
BNRE	4162	24	176.4	3723
DFSM	4373	42	211.4	3716
Z03	7081	15	461.2	6987

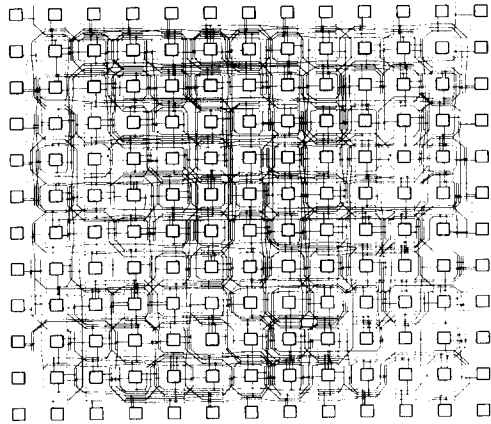


그림 10. CMRF에 의한 입력 회로 BUSC의 배선 결과

Fig. 10. The routing result of the circuit BUSC by CMRF.

V. 결론

본 논문에서는 다수의 네트를 동시에 배선하는 새로운 배선 방법을 제안하고 이를 이용하여 symmetrical array 형태의 FPGA 배선에 사용할 수 있는 배선기 CMRF를 소개 하였다. CMRF에서는 배선하고자 하는 네트를 다음과 같은 과정을 통하여 배선한다. 먼저 동시에 배선될 네트들을 상호 배선 결과에 미치는 영향을 고려하여 선택한다. 다음 선택된 각각의 네트에 대해 maze propagation과 back-tracing을 통해서 최단 거리 배선 경로를 모아 놓은 배선 그래프를 만든 후, 각 네트의 배선 그래프에서 배선 그래프들 사이의 상호 연관성을 고려하여 배선 경로를 선택한다. 그리고 배선에 실패한 네트에 대해서는 들어내기 및 재배선 과정을 통하여 배선을 완료한다. 이러한 CMRF로 한번에 다수의 네트씩 동시에 배선하는 경우를 한번에 하나의 네트씩 배선하는

경우와 비교해 본 결과 모든 실험 자료에 대해서 높은 배선율을 얻었다.

그리고 Brown 등이 사용한 다섯 개의 benchmark 입력에 대해 실험한 결과 CMRF는 그들의 CGE 방법보다 세 개 이상 적은 갯수의 와이어 세그먼트를 갖는 연결 블록을 사용하여도 그들의 배선을 완료할 수 있었다. 또한, 본 논문에서 사용한 FPGA의 배선 구조는 Xilinx의 3000 계열과 유사한데 4000 계열과 같은 배선 구조에도 우리의 동시 미로 배선 방법을 사용할 수 있으며 이들에 추가로 존재하는 direct line 또는 double line 등과 같은 특별한 와이어 세그먼트가 있는 경우도 우리의 방법을 적용할 수 있다. 마지막으로 본 논문에서 제안한 동시 미로 배선 방법은 symmetrical array 형태의 FPGA 뿐만 아니라 row-based array 형태^[1,5]의 FPGA 배선에도 쉽게 적용할 수 있다. 그리고 이러한 CMRF에서의 배선 방법을 토대로 배선 지연을 고려한 효율적인 배선 방법에 대한 연구를 현재 진행 중이다.

감사의 글

우리의 CMRF를 실험하는데 사용한 benchmark 자료와 CGE 방법에 의한 배선 프로그램을 제공하여 주신 Toronto 대학의 J. Rose 교수에게 감사를 드립니다.

參考文獻

- [1] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic, *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, 1992.
- [2] S. Brown, J. Rose and Z. G. Vranesic, "A Detailed Router for Field-programmable Gate Arrays," *IEEE Trans. on CAD*, Vol. 11, No. 5, May 1992, pp. 620 - 628.
- [3] A. D. Brown and M. Zwolinski, "Lee Router Modified for Global Routing," *Computer-Aided Design*, Vol. 22, No. 5, pp. 296 - 300, 1990.
- [4] C. Chiang and M. Sarrafzadeh, "Global Routing Based on Steiner Min-Max trees," *IEEE Trans. on CAD*, Vol. 9, No. 12, Dec. 1990, pp. 1318 - 1325.

- [5] J. Greene, V. Roychowdhury, S. Kaptanoglu and A. El Gamal, "Segmented Channel Routing," Proc. 27th DAC, pp. 567 - 572, June 1990.
- [6] T. Ohtsuki, *Layout Design and Verification, Advances in CAD for VLSI*, Vol. 4, North Holland, 1991.
- [7] M. Palczewski, "Plane Parallel A*Maze Router and Its Application to FPGAs," 29th DAC, pp. 691 - 697, 1992.
- [8] J. Rose, "Parallel Global Routing for Standard Cells," IEEE Trans. on CAD, Vol. 9, No. 10, pp. 1085 - 1095, Oct. 1990.
- [9] H. Shin and A. Sangiovanni-Vincentelli, "Mighty: A 'Rip-up and Reroute' Detailed Router," Proc. ICCAD-86, 1986, pp. 2 - 5.
- [10] P. Tzeng and Carlo H. Sequin, "Codar: A Congestion-Directed General Area Router," ICCAD-88, 1988, pp. 30 - 33.
- [11] *The Programmable Gate Array Data Book*, Xilinx Inc., 1991.

 著者紹介



崔眞英(正會員)

1992年 2月 한국 과학기술대학 전자계산학과 학사. 1994年 2月 서강대학교 전자계산학과 석사. 1994年 3月 ~ 현재 금성사 중앙연구소 ASIC 센터 기술실. 주관

심 분야는 ASIC 시스템 설계, VLSI 테스트 등임.



林鍾錫(正會員)

1981年 서강대학교 전자공학과 학사. 1983年 한국 과학기술원 전기 및 전자공학과 석사. 1989年 Univ. of Maryland, College Park, 전자공학과 박사. 1983年 3月 ~ 1990年 8月 한국 전자통신연구소 연구원. 1990年 9月 ~ 현재 서강대학교 전자계산학과 부교수

신연구소 연구원. 1990年 9月 ~ 현재 서강대학교 전자계산학과 부교수