

論文 96-33A-4-18

대칭형 FPGA의 새로운 배선구조와 배선 알고리즘

(A New Routing Architecture for Symmetrical FPGA and its Routing Algorithm)

嚴洛雄*, 曹漢鎭*, 朴仁學*, 慶宗旻**

(Nak Woong Eum, Han Jin Cho, In Hag Park, and Chong Min Kyung)

요 약

대칭형 FPGA(Field Programmable Gate Array)의 새로운 배선구조와 제안된 구조에 적합한 배선 알고리즘을 제안한다. 본 배선구조는 선분화된 채널을 채택하여 다양한 길이의 배선선분을 제공함으로써 스위치소자 사용을 줄인다. 그리고 서로 다른 두 네트가 스위치모듈 내의 동일 트랙을 사용할 수 있게 하는 스위치소자 배치를 갖는다. 배선 알고리즘은 한 네트에 대해 모든 배선 가능한 트랙을 찾은 후, 연속스위치를 가장 적게 사용하거나 배선에 활용되지 않는 부분이 최소인 트랙을 선택한다. 배선 알고리즘의 성능을 안정화하기 위해 배선을 네트가 갖는 배선선분의 양, 예측되는 배선 길이, 핀의 수를 고려하여 배선 순서로 사용한다. 실험결과, 채널밀도보다 필요한 트랙 수가 증가하지 않고도 교차스위치 수가 기존보다^{[1][13]} 40 % 정도로 줄었다.

Abstract

This paper presents a new symmetrical routing architecture for FPGA and an efficient routing algorithm for the architecture. The routing architecture adopts the segmented wires and the improved switch modules. Segmented wires construct routing channels which pass through the chip in vertical and horizontal directions. To maximize the utility of a track, a track in each switch module can be separated in two part using a programmable switch to route two different net. The proposed routing algorithm finds all assignable tracks for a given net and selects the best track from assignable tracks to minimize the number of programmable switches and the unused portion of the wire segments. In order to stabilize the performance of the algorithm, the routing order is defined by weighted sum of the number of wire segment, the length of wire segment, and the number of pin. Experimental results show that routability is improved dramatically and the number of crossing switches are reduced about 40 % compared with the previous works^{[1][13]}.

I. 서 론

FPGA는 PLD(Programmable Logic Device)의 사용자 구현성(user-programmability)과 MPGA

(Mask Programmable Gate Array)의 고집적성을 절충한 IC(Integrated Circuit)이다. FPGA의 사용자 구현성은 기본적으로 스위치소자에 의해 얻어진다. 스위치소자는 배선자원과 논리 소자를 연결하기 위해 필요하며, FPGA로 논리회로를 구현하기 위해서는 많은 스위치소자가 사용된다. 이에 따라 면적, 단락저항값 및 기생전용량값이 적을수록 우수한 스위치소자로 분류되며 적용 분야에 따라서 비휘발성(non-volatility) 및 재프로그래밍성(re-programmability) 등도 요구된다. 상용 FPGA에서 SRAM(Static RAM), Anti-fuse, (E)EPROM 등이 스위치소자로서 사용되

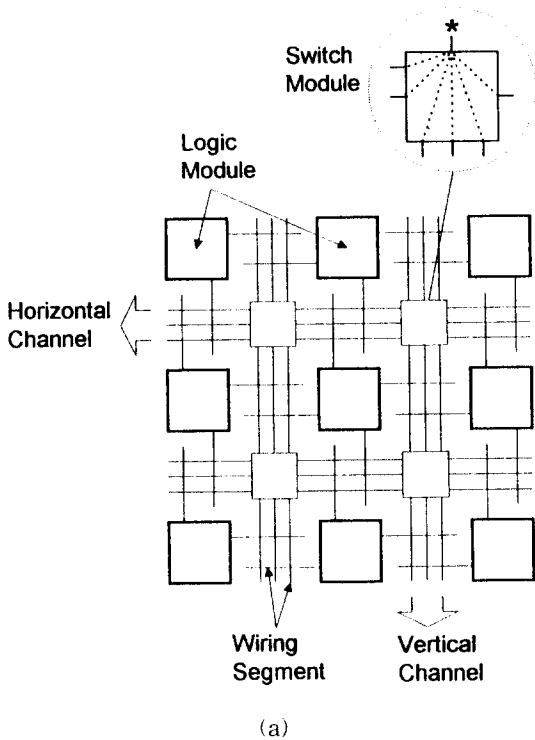
* 正會員, 韓國電子通信研究所 自動設計研究室
(Design Automation Section, Elec. and Tele. Research Institute)

** 正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Elec. Eng., Korea Advanced Institute of Science and Technology)

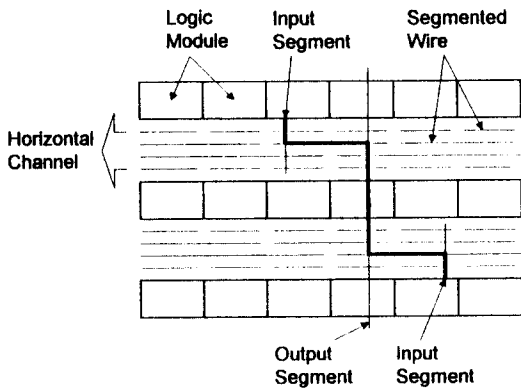
接受日字: 1995年6月29日, 수정완료일: 1996年2月24日

고 있다. 현재 가장 우수한 전기적 특성을 갖는 Anti-fuse의 경우에도 그 단락저항값은 50-80Ω 정도이므로^[1] 스위치소자 사용을 최소화하는 것이 필요하다.

과 그들간의 접속을 담당하는 배선자원으로 구성된다. 대표적인 회로구조(architecture)로서 논리모듈과 배선 자원이 이차원 대칭으로 분포된 대칭형 구조와^[2] 수평의 배선채널만을 갖는 일자형 구조가^[3] 있다. 대칭형 FPGA는 그림 1 (a)와 같이 배선자원으로서 스위치모듈(switch module) 및 배선선분(wiring segment)을 갖는다. 스위치모듈 사이의 배선선분은 단위 길이로 구성되며, 그림에서 별표를 한 배선선분은 스위치모듈에 의해 6개의 배선경로를 가질 수 있다. 일자형 FPGA는 그림 1 (b)와 같이 선분화된 배선선분(segmented wire)의 수평채널을 갖는다. 배선선분 사이에는 스위치소자가 존재하며 그들의 길이는 대칭형 FPGA에 비해 다양하다. 대칭형 FPGA는 수평과 수직의 채널을 모두 지니므로 논리모듈의 상하좌우에 편을 갖지만 일자형 FPGA는 수평채널만을 지니므로 각 논리모듈의 상하에만 편을 갖는다. 이상에서 배선선분은 금속선이다.



(a)



(b)

그림 1. FPGA의 대표적인 회로구조 (a) 대칭형 (b) 일자형

Fig. 1. The representative FPGA architecture. (a) Symmetrical (b) Row-based

FPGA는 논리를 구현하는 논리모듈(logic module)

새로운 FPGA 칩의 개발 과정은 스위치소자 및 회로구조 연구 분야로 나누어진다. 스위치소자 연구 분야는 새로운 공정 기술을 개발하며, 회로구조 연구 분야에서는 논리모듈 및 배선구조를 고안하거나 개선한다. FPGA를 이용한 논리 회로 설계는 배치 및 배선의 일반적인 레이아웃 발생과정을 따르며, 배선은 다시 개략 배선(global routing)과 상세배선(detailed routing)으로 나누어진다. FPGA의 배치문제는 MPGA나 표준셀(Standard Cell)과 유사하므로 기존의 알고리즘이 변형되어 적용될 수 있다^[4]. 배선은 스위치소자와 배선 자원을 이용하여 논리모듈간을 연결하면서 지연시간 최소화를 도모한다. 신호의 지연은 스위치소자, 논리모듈, 배선선분 등에 의해 발생한다. 스위치소자 및 배선선분은 칩의 제조 시에 결정되므로 배선자원을 효율적으로 이용하는 회로구조 및 레이아웃 알고리즘이 연구되고 있다.

본 논문은 대칭형과 일자형 회로구조를 절충하여 스위치소자 사용량을 줄일 수 있는 새로운 대칭형 배선구조를 제안한다. 배선구조 개선을 통해 얻어지는 효과를 논하며 이를 평가하기 위해 동일한 개략배선 정보를 이용하여 상세배선을 수행한다. 다음 II장에서 FPGA의 기존 배선구조와 알고리즘을 고찰하고, III장에서 새로운 배선구조 그리고 IV장에서 새로운 배선구조를 위한 상세배선 알고리즘을 제안한다. V장에서 실험결과를 보이고 마지막으로, VI장에서 결론을 내린다.

II. 기존 배선구조와 배선 알고리즘의 고찰

MPGA에서는 필요한 배선경로를 금속선을 사용하여 임의로 형성할 수 있지만, FPGA에서는 칩에 존재하는 배선선분과 스위치소자만으로 배선을 수행해야 한다. 일자형 FPGA에서 한 네트가 동일 채널의 배선에 사용할 수 있는 스위치소자 수를 K 라고 할 때, 그의 배선문제는 K 선분화 채널배선(K -segmented channel routing)으로 정의된다^[5-7]. K 의 값이 증가하거나 배선선분의 길이가 짧아질수록 배선은 쉬워진다. 그러나 배선에 사용되는 스위치소자가 증가하므로 누적된 저항성분에 의해 회로의 동작특성은 낮아진다. 균일장 선분(Uniform Length Segmentation) 모델의 일자형 FPGA는 일정한 길이의 배선선분이 채널에 균일하게 분포된 형태를 갖는다. 이것은 K 선분화 채널 배선문제에서 K 값이 1인 경우이다. 논리모듈의 각 핀에 수직의 전용 배선선분이 존재하므로 수직제약(vertical constraint)이 발생하지 않는다. 이 문제는 left-edge 알고리즘으로 배선하는 것이 충분하다고 알려져 있으며^[5], K 값이 2 이상인 경우는 NP-complete 문제임이 증명되었다^[7].

균일장 선분 모델을 사용하는 칩에서 길이가 긴 네트를 배선하려면 스위치소자를 이용하여 다수의 배선 선분을 연결해야 한다. 이로 인해 지연시간이 증가하므로 회로구조의 설계 단계에서 길이가 긴 배선선분도 채택하는 것이 요구된다. 사용되는 스위치소자 수를 줄이기 위해 긴 배선선분을 과도하게 사용하면 배선에 활용되지 않는 부분이 증가하고 그들의 기생정전용량 때문에 신호의 지연시간이 커진다. 단순히 긴 배선선분을 사용하여 스위치소자 수를 줄인 회로구조는 지연시간 최적화를 얻을 수 없을 뿐 아니라 트랙 사용량 및 칩 면적을 증가시킬 수 있다. 따라서 회로의 평균 배선 길이 및 논리모듈의 평균 핀 수 등을 고려한 배선선분의 선분화에 대한 연구가 회로구조 설계에 적용되고 있다^[8-9].

대칭형 회로구조의 배선에 Maze^[10]와 BFS(Breadth First Search)^[11]를 이용한 연구가 있다. Maze를 이용한 기법^[10]은 개략배선의 과정 없이 직접 배선을 수행한다. BFS를 채용한 CGE(Coarse Graph Expansion)^[11]은 상세배선 알고리즘으로서 개략배선 정보를 받아들인 후, 모든 네트의 배선경로를 BFS로 동시에 탐색하여 비용이 적은 순서로 신호선을

트랙에 할당한다. 각 네트의 모든 배선경로를 탐색하기 위해서는 많은 메모리와 계산이 필요하므로 CGE는 pruning 기법을 사용한다. Pruning은 배선경로를 탐색하는 각 단계에서 이용 가능한 배선경로의 일부만을 선택하는 것을 의미하는데, 일부의 배선경로만을 고려하여 상세배선을 시도하므로 배선이 실패할 수 있다. 만약 배선이 실패하면 수행한 배선의 일부를 삭제하고 이전단계보다 더 많은 배선경로를 선택하여 재배선을 시도한다.

일자형 회로구조의 배선문제는 각 채널별로 독립적인데 비해서 대칭형 회로구조의 배선문제는 분리할 수 없다. 대칭형 회로구조에서는 논리모듈의 상하좌우 측에 존재하는 핀들로 인해 채널이 수평과 수직으로 교차하므로 채널의 배선문제는 서로 종속적이다. 채널이 교차하는 지점의 스위치모듈은 대칭형 회로구조의 설계 및 레이아웃 문제를 더욱 복잡하게 한다. 그림 1(a)에서와 같이 스위치모듈에 접하는 각 배선선분들에 독특한 배선패턴이 설정되므로 각 배선선분에서 배선할 수 있는 경로가 매우 다양하기 때문이다. 그래서 CGE는 다중핀(multi-pin) 네트를 처리하지 못하고 그들을 이중핀(two pin) 네트로 분리하여 배선을 수행한다.

스위치모듈의 구조 연구를 통해 대칭형 FPGA의 배선구조를 개선할 수 있다^[12-14]. Y. Sun^[12]은 고정된 길이의 배선자원을 수평과 수직채널에 분포시키고 그들의 교차점에 선택적으로 스위치소자(crossing switch)를 한 개씩 두었다. 이 구조는 기존의 스위치모듈과 달리 스위치소자의 수에 비해 다양한 배선경로를 제공할 수 있으므로, CGE가 사용하는 구조에 비해 총 스위치 수를 대폭 줄일 수 있었다. 그러나 배선선분들을 결합시킬 수 있는 수단을 제공하지 않았으므로 개략배선 과정에서 정의한 경로가 상세배선에 일치하지 않는 문제를 지녔다. 이를 해결하기 위해 Y. Sun은 일자형 배선구조의 선분화 채널과 유사하게 배선선분들 사이에 스위치소자를 채택하였다^[13]. 이러한 스위치를 연속스위치(continuing switch)라 하며 스위치모듈의 가장자리를 따라 분포시켰다. 그림 2는 Y. Sun이 사용한 스위치모듈 구조 및 배선구조를 보여준다^[13]. 그림에서 ○는 연속스위치를 나타내고 ×는 교차스위치를 나타낸다. K. Zhu^[14]는 스위치소자의 설계문제를 다루었다. 단일 스위치모듈에 대해 임의로 발생시킨 배선요구의 조건을 선형 방정식으로 모델링 하여 스위

치소자의 분포를 구하였다.

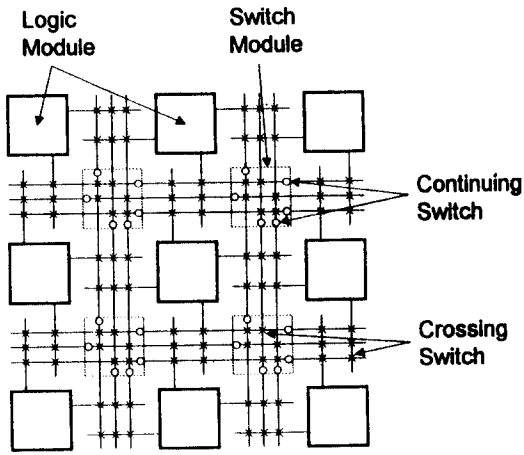


그림 2. Y. Sun의 배선구조
Fig. 2. Y. Sun's routing architecture.

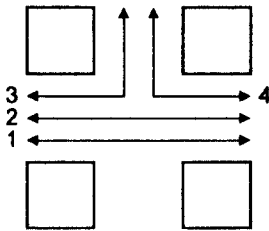


그림 3. 채널밀도가 3인 배선요구의 예
Fig. 3. An routing requirement, channel density is 3.

Y. Sun의 스위치모듈에서는 한 트랙을 두 네트가 공유할 수 없다. 그림 2에서 보이는 바와 같이 스위치 모듈내의 각 수평선분이나 수직선분은 스위치모듈을 관통하고 있으므로, 한 네트가 어떤 트랙을 사용한다면 그 트랙에는 다른 네트를 위해 사용할 수 있는 배선선분이 더 이상 존재하지 않는다. 따라서 채널밀도보다 많은 트랙이 필요한 경우가 발생할 수 있다. 예를 들어 개략배선에 의해 그림 3과 같이 네트 1, 2, 3 및 4의 배선경로가 얻어졌음을 가정한다. 이 경우 채널밀도는 3으로 계산된다. Y. Sun의 배선구조에서 상세배선을 수행하면 그림 4 (a)와 같은 결과가 얻어진다. 즉, 네트 3이나 4가 사용하는 트랙을 서로 공유할 수 없으므로 4개의 트랙이 필요하다. 이와 같이 Y. Sun의 배선구조에서는 상세배선 과정에서 트랙 사용량이 많아질

수 있으므로 개략배선에서 정의한 채널밀도가 여전히 부합하지 않는다. 그림에서 ●와 ×는 각각 사용된 연속스위치와 교차스위치를 나타낸다.

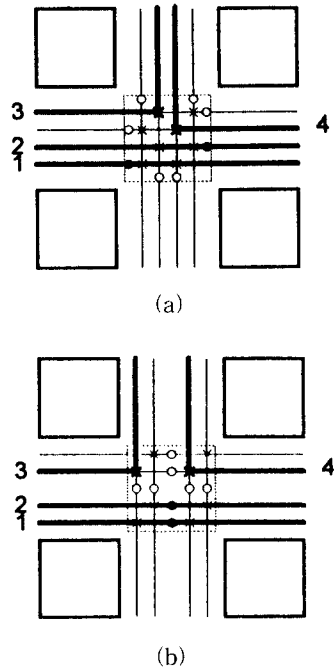


그림 4. 채널밀도와 배선에 필요한 트랙 수의 비교
(a) Y. Sun의 구조에서 그림 3의 배선요구 처리, 필요한 트랙 수 = 4
(b) 제안한 구조에서 동일한 배선요구 처리, 필요한 트랙 수 = 3

Fig. 4. The required number of tracks versus channel density.

- (a) Routing on Y. Sun's architecture for the requirements shown in Fig 3, the number of required tracks is 4.
- (b) Routing on the proposed architecture for the same requirements, the number of required tracks is 3.

III. 새로운 대칭형 배선구조

본 논문에서 추구하는 배선구조 개선의 주목적은 배선에 필요한 총 스위치의 수를 증가시키지 않으면서도 채널밀도보다 많은 트랙을 사용하지 않도록 하는 것이다. 앞에서 논의한 바와 같이 Y. Sun의 배선구조는 교차스위치와 연속스위치로서 총 스위치 수를 감소시켰지만 상세배선에 필요한 트랙 수가 채널밀도보다 증가하는 문제를 지닌다. 이를 해결하기 위해 본 논문의 배

선구조는 연속스위치를 스위치모듈의 내부에 채택한다. 이처럼 단순한 구조 변경만으로 수평선분과 수직선분들은 스위치모듈 내에서 두부분으로 분리될 수 있다. 분리된 선분들은 보다 긴 선분을 형성하기 위해 연속스위치를 이용하여 이어질 수 있고, 각각 서로 다른 네트의 배선을 위해 사용될 수도 있다. 제안한 구조를 이용하여 그림 3의 배선요구를 처리한 예는 그림 4 (b)와 같다. 그림과 같이 연속스위치가 스위치모듈의 내부에 존재하면 네트 3과 4와 같은 구부러진 네트의 트랙 공유가 가능할 뿐만 아니라 네트 1이나 2와 같이 스위치모듈을 관통하는 배선요구도 처리할 수 있다. 따라서 상배선에서 필요한 트랙의 수는 채널밀도와 같은 3이다. 구조 개선으로 서로 다른 네트가 스위치모듈 내의 동일한 트랙을 공유할 수 있으므로 필요한 트랙의 수는 채널밀도보다 증가하지 않는다.

트랙의 수를 D 그리고 스위치모듈내의 총 교차스위치의 수를 C 라고 할 때, 교차스위치를 둘 수 있는 경우의 수는 $w \times D \times C$ 이다. 본 논문은 대각선 방향으로 교차스위치를 분포시킨 경우를 다루고 그에 대한 배선 알고리즘을 제시한다.

인접한 두 논리모듈의 중심 사이 거리를 배선선분의 기본 길이로 정한다. 길이가 긴 신호선을 위해 기본길이의 2의 멱승에 해당하는 배선선분들을 둔다. 일반적으로 길이가 짧은 네트들이 길이가 긴 네트에 비해 칩에 많이 존재하므로¹⁵⁾, 배선선분의 길이에 따른 빈도를 길이에 반비례하도록 한다. 배선선분의 길이를 기본 길이보다 크게 만드는 방법은 배선선분이 관통하는 스위치모듈의 해당 트랙에서 연속스위치를 제거하는 것이다. 긴 배선선분이 많을 수록 칩이 갖는 연속스위치의 수는 줄어들지만 배선이 어려워진다¹⁶⁻¹⁷⁾.

스위치모듈 내에서 한 트랙이 만날 수 있는 교차스위치의 수를 Q 라고 한다. 스위치모듈의 교차스위치 위치를 Q 에 따라 변수화하기 위해 $W = \{w_m \mid m = 1, 2, \dots, Q\}$ 를 정의한다. w_m 은 $\{0, 1, \dots, D-1\}$ 의 값을 가질 수 있다. 위의 기호를 사용하여 수직트랙 i 가 수평트랙과 이루는 교점 중에 교차스위치가 놓이는 위치를

$$\{(i, (i+w_m) \bmod D) \mid i = 0, 1, 2, \dots, D-1 \text{ and } m = 1, 2, \dots, Q\}$$

와 같이 정의한다. 여기서 mod는 modulo 연산을 의미한다. 이 정의에 따라 w_i 가 0인 경우에 교차스위치는

스위치모듈의 좌하점과 우상점을 연결하는 대각선상에 놓인다. 그림 5 (a)는 $D = 6, Q = 1, w_1 = 0$ 인 스위치모듈을 그림 5 (b)는 $D = 6, Q = 3, w_1 = 0, w_2 = 2, w_3 = 3$ 인 스위치모듈을 보인다. 그리고 그림 5 (b)는 단위 길이보다 긴 배선선분을 제공하기 위해 연속스위치의 일부가 삭제된 예를 보여준다. 이와 같이 D, Q 및 W 그리고 채널의 선분모델이 주어지면 배선구조를 구성할 수 있다.

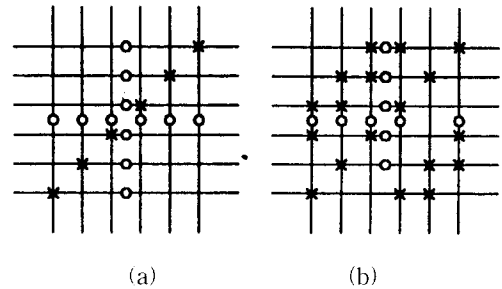


그림 5. D, Q 및 W 에 의해 변수화되는 스위치모듈 구조

(a) $D=6, Q=1, w_1=0$ (b) $D=6, Q=3, w_1=0, w_2=2, w_3=3$

Fig. 5. The structure of switch module is parameterized by D, Q and W .

(a) $D=6, Q=1, w_1=0$ (b) $D=6, Q=3, w_1=0, w_2=2, w_3=3$

제안한 배선구조에서 D 와 Q 가 같은 값을 갖는 경우, 수평선분과 수직선분이 만나는 모든 위치에 교차스위치가 존재하므로 채널배선문제는 서로 독립적이다. 그러나 칩이 갖는 총 스위치 수가 많으므로 활용되지 않는 스위치의 비율이 매우 높다. 따라서 Q 의 값을 가능한 낮추는 것이 필요하다. Q 의 값이 D 보다 적어지면 각 채널의 배선문제는 종속적으로 된다. 즉, 한 채널을 배선하면서 다른 채널에 미치는 영향을 고려해야 하므로 배선문제가 복잡해진다. 다음 장에서는 Q 의 값이 D 보다 적을 때 배선을 효과적으로 수행하는 알고리즘을 제안한다.

IV. 배선 알고리즘

배선 알고리즘을 전개하기 전에 몇 가지 기호를 정의한다. V_j 와 H_j 는 원점으로부터 각각 I 와 J 번째의 수직 및 수평채널을 의미한다. 모든 채널에서 X 와 Y 가

증가하는 방향으로 만나는 각 트랙의 이름을 $\{ t_i \mid i = 0, 1, \dots, n \}$ 으로 부여한다. 각 채널이 가지는 트랙 수는 $n+1$ 개이며, $D = n+1$ 이다. 이 정의에 따라 수평 채널 H_J 의 가장 낮은 트랙은 $t_0(H_J)$, 가장 위의 트랙은 $t_n(H_J)$, 그리고 수직채널 V_I 의 가장 왼쪽 트랙은 $t_0(V_I)$, 가장 오른쪽 트랙은 $t_n(V_I)$ 로 표현한다. V_I 와 H_J 가 교차해서 이루는 스위치모듈을 $S_{I,J}$ 로 나타낸다. 기호에 대한 예는 그림 6과 같다.

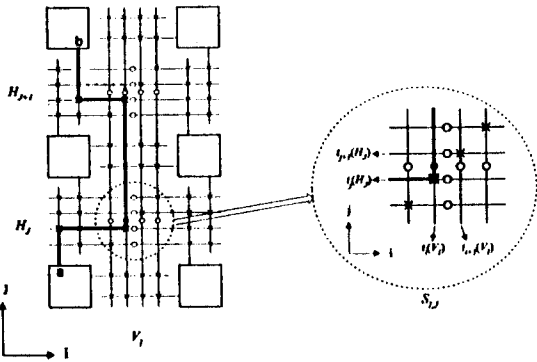


그림 6. 트랙 할당 문제
Fig. 6. Track assignment problem.

본 논문의 배선구조에서 채널배선 문제는 서로 종속적이다. 이것은 스위치모듈 내의 각 배선선분이 가질 수 있는 교차스위치 수가 제한되므로 발생한다. 예를 들어 $Q = 2$ 이고 $w_1 = 0$ 인 그림 6의 배선구조 상에서 핀 a를 b와 연결하고자 한다. $S_{I,J}$ 의 수평트랙 $t_j(H_J)$ 를 사용한다면 $S_{I,J-1}$ 에서 가능한 수평트랙은 $t_j(H_{J-1})$ 밖에 없다. 만약, $t_j(H_{J-1})$ 에 이미 다른 신호선이 할당되었다면 $t_j(H_J)$ 를 통한 핀 a와 b의 배선은 불가능하다. 이 관찰은 네트가 존재하는 모든 수평 및 수직채널에서 동시에 트랙 할당을 수행해야 하는 것을 보인다. 본 논문의 상세배선 알고리즘은 한 네트에 대해 채널당 한 개의 트랙만을 사용한다. 하나의 네트를 위해 동일한 채널에서 여러 트랙을 사용하면 트랙 위치를 변경하기 위해 스위치소자가 추가로 필요하기 때문이다. 아래에서 트랙 할당 문제의 해석적인 해법을 설명한다.

위에서 정의한 Q 와 $W = \{w_1, w_2, \dots, w_m, \dots, w_Q\}$ 가 주어지고 수평트랙 $t_i()$ 에 임의의 네트를 할당할 때, 다른 모든 수평채널과 수직채널에서 이 네트가 사용할 수 있는 트랙은 다음과 같이 정의된다.

$$T_H(i) = \{ t_{(i+w_1) \bmod D}(H_J), t_{(i+w_2) \bmod D}(H_J), \dots, t_{(i+w_m) \bmod D}(H_J), \dots, t_{(i+w_k) \bmod D}(H_J) \text{ for all } J \}$$

$$T_V(i) = \{ t_{(i-w_1+D) \bmod D}(V_I), t_{(i-w_2+D) \bmod D}(V_I), \dots, t_{(i-w_m+D) \bmod D}(V_I), \dots, t_{(i-w_k+D) \bmod D}(V_I) \text{ for all } I \}$$

$T_H(i)$ 와 $T_V(i)$ 은 각각 수평트랙과 수직트랙의 집합을 의미하고 I 와 J 는 네트가 속하는 모든 수평트랙과 수직트랙을 포함한다. t 의 인덱스가 D 보다 큰 경우는 modulo 연산을 수행하고 0 보다 작은 경우에는 D 를 더한 후 modulo 연산을 수행한다. 그래서 인덱스는 항상 스위치모듈 내의 좌표를 갖는다.

위의 수식에 따라 그림 6에서 사용할 수 있는 트랙을 구해 보자. 그림 6은 $D = 4, Q = 2$ 그리고 $w_1 = 0$ 이므로, 트랙 $t_1(H_J)$ 에 네트가 할당되는 경우에 $T_H(i) = \{t_1(H_{J-1})\}$ 그리고 $T_V(i) = \{t_1(V_I)\}$ 와 같이 구해진다. 이 결과로부터 트랙 $t_1(H_{J-1})$ 과 $t_1(V_I)$ 가 미사용된 경우에는 배선이 가능함을 알 수 있다.

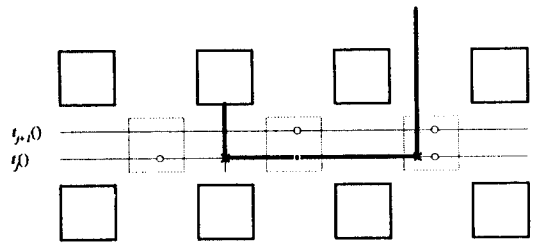


그림 7. 가능한 트랙 중에서 트랙 선택
Fig. 7. Track selection from the feasible tracks.

한 채널의 동일한 트랙을 이용하면서 네트의 중요도에 따라 배선경로를 순차적으로 찾는다. $T_H(i)$ 와 $T_V(i)$ 에 속하는 어떤 트랙을 사용하더라도 배선이 가능하지만 배선의 성능을 높이기 위해서 보다 유리한 것을 선택하는 것이 필요하다. 예를 들어 그림 7에서 트랙 $t_j()$ 와 $t_{j-1}()$ 이 어떤 네트에 대해 사용할 수 있는 트랙이라고 가정한다. 트랙 $t_{j-1}()$ 은 $t_j()$ 에 비해 연속스위치를 하나 더 사용하며 배선 중에 활용되지 않는 부분도 많다. 따라서 $t_j()$ 에 네트를 할당하는 것이 바람직하다. 배선 알고리즘은 위의 두 요소 중에 연속스위치의 사용량을 먼저 평가하도록 구성되었다. 연속스위치의 사

용량이 같은 경우는 배선에 활용되지 않는 부분이 적은 트랙을 선택한다.

배선 순서는 결과에 영향을 미칠 수 있으므로 실험을 통해 다음과 같이 네트의 배선비용 f 를 정하였다.

$$f = C_1 \times R_1 + C_2 \times R_2 + C_3 \times R_3$$

단,

R_1 = 네트를 포함하는 채널의 수,

R_2 = 네트의 배선길이,

R_3 = 네트가 갖는 핀의 수,

$C_1 > C_2 > C_3$: weight.

네트가 여러 채널에 걸쳐 존재할수록 기 수행된 배선의 영향을 많이 받는다. 이를 우선적으로 처리하기 위해 C_1 이 가장 크다. R_1 은 R_2 와 차원이 다르고 R_2 에 비해 적은 값을 가지므로 C_1 은 칩의 가로 및 세로 길이 중에서 큰 값으로 정하였다. 배선비용 f 의 첫째 항에 의해 구부러진 네트를 갖는 네트는 순서대로 모두 처리된다. 배선비용 f 의 첫째 항의 값이 같을 때 둘째 항과 마지막 항이 추가로 사용된다. 네트의 배선길이는 개략배선에서 정의된 값을 사용한다. 실험에서 $C_2 = 1.5$ 이고 $C_3 = 1$ 일 때 가장 좋은 결과가 얻어졌다. 배선 알고리즘은 다음과 같다.

Detailed_Route()

```
{
  read global routing information;
  for each net {
    calculate cost value;
  }
  sort the cost value in non-increasing order;
  for each sorted net {
    find the feasible tracks  $T_H()$  and  $T_V()$ ;
    for each feasible tracks {
      evaluate the number of continuing switches and
      unused portion of the wire segments;
    }
    select the best track from the feasible tracks;
    assign net to the track in each channel;
  }
}
```

본 알고리즘은 한 네트에 대해 채널마다 한 트랙만을 할당하므로, 트랙 할당 상황을 나타내기 위해 채널당 트랙 수 만큼의 메모리가 필요하다. 수평 및 수직채

널의 총 수를 A , 채널의 트랙 수를 D 라고 할 때 트랙 할당 정보를 표현하기 위해 $O(AD)$ 의 메모리가 필요하다. 배선비용을 크기순서로 배열하는 데는 $O(N \log N)$ 의 시간이 필요하다. 여기서 N 은 회로가 갖는 네트 수이다. 어떤 네트에 대한 $T_H()$ 와 $T_V()$ 의 계산은 그를 포함하는 채널의 모든 트랙을 탐색한다. $T_H()$ 와 $T_V()$ 의 계산은 칩과 회로의 크기에 선형적으로 비례한다. 연속 스위치의 수와 배선 중에 활용되지 않는 부분을 계산하는 것은 할당 가능한 트랙에 대해서만 수행하므로 계산 시간을 상수로 볼 수 있다. 이상에서 상세배선 알고리즘의 전체 시간복잡도는 $O(AN)$ 이다.

V. 실험결과

제한한 알고리즘을 C 언어로 SUN SPARC-10에서 구현하였다. 표 1은 실험에 사용한 회로들의 네트 수, 논리모듈 수, 사용한 칩의 크기를 보여준다. 스위치 모듈 내에서 신호선이 구부러진 것을 계산하여 각 회로에서 가장 큰 값을 Jog_{max} 로 나타낸다. 칩이 갖는 연속스위치는 선분모델에 따라 결정된다. 비교문헌^[11]^[13]에서 이에 관한 자료가 제공되지 않으므로 100% 배선을 위해 필요한 스위치모듈내의 교차스위치 수와 최소 트랙 수를 비교하였다. 실험에서 사용한 개략배선 정보는 CGE가 사용한 것과 동일하다.

표 1. 실험에 사용한 회로
Table 1. Experimental circuits.

회로명	네트 수	논리모듈 수	원판 크기	Jog_{max}
BUSC	392	109	13×12	7
DMA	771	224	18×16	8
BNRE	1257	362	22×21	8
DFSM	1422	401	23×22	7
Z03	2135	586	27×26	7

구부러진 네트가 많을 수록 스위치모듈은 많은 교차스위치를 필요로 한다. 그래서 Jog_{max} 는 한 트랙이 스위치모듈 내에서 갖는 교차스위치의 수인 Q 와 최소트랙 수에 영향을 미친다. 표 2의 실험결과에서 보인 바와 같이 배선을 완결하기 위해 필요한 최소 트랙의 수

는 문헌^[11]과 비슷하지만, 문헌^[13]에 비해서는 크게 개선되었다. 문헌^[11]과 비슷한 이유는 개략배선의 정보가 동일하기 때문이고 문헌^[13]에 비해 개선된 것은 III장에서 설명한 바와 같이 서로 다른 두 네트가 스위치모듈 내의 트랙을 공유할 수 있기 때문이다. Q의 값은 문헌^[11]에 비해 평균적으로 50% 이상 감소되었으며, 문헌^[13]에 비해서는 회로에 따른 편차가 있기는 하지만 크게 감소됨을 알 수 있다. 실험 회로의 J_{ogmax} 는 7이나 8로 일정한데도 불구하고 문헌^[13]은 회로에 따라 Q의 변화가 심하다. 하나의 스위치모듈이 갖는 교차스위치 수를 표 2의 마지막 열에서 보이고 있다. 이 값은 D와 Q를 곱하여 구한 것인데, 괄호 속에 있는 수치는 우리의 실험결과를 1로 하여 문헌^[11] [13]이 사용하는 교차스위치 수를 나타낸다. 교차스위치 수가 비교문헌에 비해 평균적으로 60 % 정도가 감소되는 것을 보여 준다.

표 2. 실험결과의 요약
Table 2. Summary of the experimental results.

회로명	트랙당 교차스위치 수(K)			최소 트랙 수(D)			스위치모듈내의 교차스위치 수		
	[11]	[13]	Ours	[11]	[13]	Ours	[11]	[13]	Ours
BUSC	6	4	2	10	11	9	60(3.3)	44(2.4)	18(1)
DMA	6	5	3	10	13	10	60(2.0)	65(2.2)	30(1)
BNRE	6	6	2	12	15	12	72(3.0)	90(3.8)	24(1)
DFSM	6	3	3	10	16	10	60(2.0)	48(1.6)	30(1)
Z03	6	8	3	13	16	13	78(2.0)	128(3.3)	39(1)

표 3. 실험에 사용한 W 및 수행시간
Table 3. W values used in the experiments and run time.

회로명	w_1	w_2	w_3	수행시간(sec.)
BUSC	0	2	-	0.3
DMA	0	3	5	0.9
BNRE	0	3	-	2.1
DFSM	0	2	8	3.0
Z03	0	5	10	5.6

표 3은 실험에 사용한 W와 수행시간을 보인다. 실험회로에 대해 다양한 W 값을 가지고 실험을 수행하였다. W 값의 변화에 따른 배선율의 변동을 관찰하였는데 그 차이는 미소하였다. 배선율에 가장 큰 영향을 미치는 요소는 Q와 D이다. BUSC와 BNRE에 대해서는 Q가 2 일 때 100 % 얻어졌고 나머지 회로는 Q가 3 일 때 모든 네트를 배선할 수 있었다. 수행시간은 앞에서 논의한 바와 같이 회로와 칩의 크기에 선형적으로 증가하였다. Z03에 대한 배선 결과를 그림 8에서 보인다.

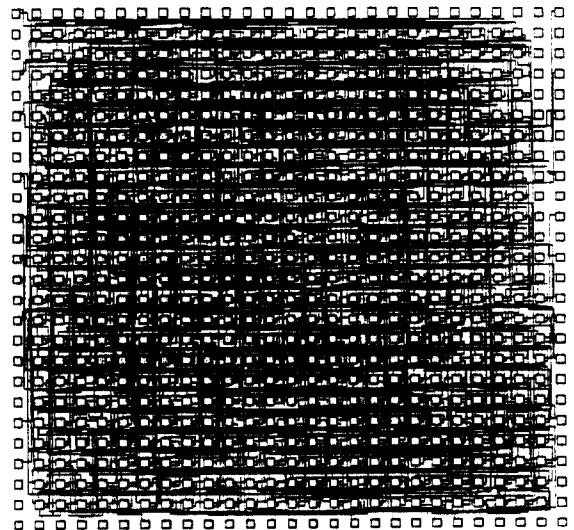


그림 8. Z03에 대한 배선
Fig. 8. Detailed routing of Z03.

VI. 결론

본 논문에서 새로운 대칭형 FPGA의 배선구조를 보이고 이 구조에 적합한 상세배선 알고리즘을 제안하였다. 본 배선구조와 알고리즘을 기존의 연구와 비교했을 때, 교차스위치가 크게 감소하며 Q가 2 내지 3의 적은 값으로도 100 % 배선을 수행할 수 있었다. 트랙 수가 증가하지 않으면서도 적은 스위치의 수로 배선을 수행할 수 있는 것은 스위치모듈내의 트랙 공유가 가능하고 실험회로의 구부러진 네트가 스위치 수에 비해 많지 않기 때문이다. 그리고 본 배선구조는 연속스위치를 통한 선분화 채널을 형성하므로 칩이 갖는 총 스위치 수가 줄어 신호의 지연시간이 크게 개선될 것이 예상

된다.

현재 상세배선 알고리즘은 재배선(rip-up-reroute)의 과정을 갖지 않는다. 재배선의 과정이 없으므로 미결선이 발생하면 단순히 Q 의 값을 증가시키면서 미결선이 없어지는 최소의 Q 를 찾는다. Q 의 값을 증가하여도 100 % 배선이 얻어지지 않으면 D 를 증가한다. 성능을 평가하기 위한 실험에서는 비교적 Q 의 값이 크지 않아도 충분한 배선율이 얻어졌지만 본 알고리즘을 실제회로에 적용하기 위해서는 재배선의 과정이 추가로 필요하다. 재배선의 한 방안은 미결선이 발생한 네트가 속하는 채널 중에서 부분 채널밀도가 가장 높은 영역에서 배선비용이 동일한 일부 네트의 트랙을 입의로 교환하는 것이다. 그래서 충돌이 일어나는 네트들을 모두 삭제하고 삭제된 네트에 대해서 제안한 배선 알고리즘을 다시 적용한다.

참 고 문 헌

- [1] S. D. Brown, R. J. Francis, J. Rose and Z. G. Vranesic, *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, p 20, 1992.
- [2] H. C. Hsieh, et. al, "Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays," in *Proceedings of Custom Integrated Circuits Conference*, pp. 31.2.1-31.2.7, 1990.
- [3] A. El Gamal, et. al., "An Architecture for Electrically Configurable Gate Arrays," *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 2, pp. 394-398, 1990.
- [4] S. K. Nag and R. A. Rutenbar, "Performance-Driven Simultaneous Place and Route for Row-Based FPGAs," in *Proceedings of the 31st Design Automation Conference*, pp. 301-307, 1994.
- [5] J. Green, V. Roychowdhury, S. Kaptanoglu and A. El Gamal, "Segmented Channel Routing," in *Proceedings of the 27th Design Automation Conference*, pp. 567-572, 1990.
- [6] K. Roy, "A Bounded Search Algorithm for Segmented Channel Routing for FPGA's and Associated Channel Architecture Issues," *IEEE Trans. on CAD*, Vol. 12, No. 11, pp. 1695-1705, 1993.
- [7] W. N. Li, "The Complexity of Segmented Channel Routing," *IEEE Trans. on CAD*, Vol. 14, No. 4, pp. 518-523, 1995.
- [8] K. Zhu and D. F. Wong, "On Channel Segmentation Design for Row-Based FPGAs", in *Proceedings of the International Conference on CAD*, pp. 26-29, 1992.
- [9] M. Pedram, B. S. Nobandegani and B. T. Preas, "Design and Analysis of Segmented Routing Channels for Row-based FPGA's," *IEEE Trans. on CAD*, Vol. 13, No. 12, 1994.
- [10] M. Palczewski, "Plane Parallel A* Maze Router and its Application to FPGAs," in *Proceedings of the 29th Design Automation Conference*, pp. 691-697, 1992.
- [11] S. D. Brown, J. Rose and Z. G. Vranesic, "A Detailed Router for Field-Programmable Gate Arrays," *IEEE Trans. on CAD*, Vol. 11, No. 5, pp. 620-628, 1992.
- [12] Y. Sun, T. C. Wang, C. K. Wong, C. L. Liu, "Routing for Symmetric FPGAs and FPICs," in *Proceedings of the International Conference on CAD*, pp. 486-490, 1993.
- [13] Y. Sun and C. L. Liu, "Routing in a New 2-Dimensional FPGA/FPIC Routing Architecture," in *Proceedings of the 31st Design Automation Conference*, pp. 171-176, 1994.
- [14] K. Zhu, D. F. Wong and Y. W. Chang, "Switch Module Design with Application to Two-Dimensional Segmentation Design," in *Proceedings of the International Conference on CAD*, pp. 480-485, 1993.
- [15] S. D. Brown, J. Rose and Z. G. Vranesic, "A Stochastic Model to Predict the Routability of Field-Programmable Gate Arrays," *IEEE Trans. on CAD*, Vol. 12, No. 12, pp. 1827-1838, 1993.
- [16] Y. L. Wu and D. Chang, "On the NP-completeness of Regular 2-D FPGA Routing Architectures and A Novel Solution," in *Proceedings of the International Conference on CAD*, pp. 362-366, 1994.

[17] T. Karnik and S. M. Kang, "An Empirical Model For Accurate Estimation of Routing Delay in FPGAs," in *Proceedings of the*

International Conference on CAD, pp. 328-331, 1995.

— 저 자 소 개 —



嚴 洛 雄 (正會員)

1962년 9월 6일생. 1984년 2월 경북대학교 전자공학과 학사, 1987년 2월 한국과학기술원 전기 및 전자공학과 석사, 1987년 2월-현재 한국전자통신연구소 선임연구원, 1992년 3월-현재

한국과학기술원 전기 및 전자공학과 박사과정 재학 중. 관심분야는 VLSI CAD 알고리즘, FPGA 설계 등임.

朴 仁 學 (正會員) 第 32卷 A編 第11號 參照

현재 한국전자통신연구소 선임 연구원

曹 漢 鎮 (正會員) 第 32卷 A編 第11號 參照

현재 한국전자통신연구소 선임 연구원

慶 宗 旻 (正會員)

1975년 2월 서울대학교 전자공학과 학사, 1977년 2월 한국과학기술원 전기 및 전자공학과 석사, 1981년 2월 한국과학기술원 전기 및 전자공학과 박사, 1981년 4월 -1983년 1월 AT&T Bell Lab 초빙연구원, 1989년 2월-1989년 12월 독일 Karlsruhe 대학교 객원 교수, 1983년 2월-현재 한국과학기술원 전기 및 전자공학과 교수. 관심분야는 마이크로프로세서 설계, CAD 알고리즘, VLSI 회로 및 시스템 설계, 컴퓨터 그래픽스 등임.