

論文 2002-39SD-8-10

배선 길이 최소화를 위한 그룹화된 스캔 체인 재구성 방법

(A Grouped Scan Chain Reordering Method for Wire Length Minimization)

李政桓*, 林種錫*

(Jung Hwan Lee and Chong Suck Rim)

요약

대규모 VLSI 시스템을 설계하는 경우 스캔 플립플롭(이하 셀)을 채택한 스캔 테스트 방법을 사용하여 IC 칩의 테스트를 용이하게 한다. 이러한 경우 스캔체인에서의 스캔 셀들의 연결 순서는 물리적 설계과정인 셀들의 배치가 완료된 후 결정하여도 무방하다. 본 논문에서는 이러한 사실을 이용하여 스캔 셀간의 연결선의 길이가 작도록 이들의 순서를 조정하는 방법을 제안한다. 특히 본 논문에서 제안하는 방법은 스캔 셀들이 클럭 도메인별로 그룹화되어 있을 경우 이들의 순서를 결정하기 위하여 새롭게 제시되는 방법으로 기존의 재구성 방법에 비하여 약 13.6%의 배선길이를 절약할 수 있다. 또한, 스캔 셀 순서에 대한 여러 다양한 제약에 대하여 효율적으로 셀들의 순서를 재구성할 수 있다.

Abstract

In order to design a huge VLSI system, the scan testing methodology by employing scan flip-flops(cells) is a popular method to test those IC chips. In this case, the connection order of scan cells are not important, and hence the order can be determined in the very final stage of physical design such as cell placement. Using this fact, we propose, in this paper, a scan cell reordering method which minimizes the length of wires for scan chain connections. Especially, our reordering method is newly proposed method in the case when the scan cells are grouped according to their clock domains. In fact, the proposed reordering method reduces the wire length about 13.6% more than that by previously proposed reordering method. Our method may also be applicable for reordering scan chains that have various constraints on the scan cell locations due to the chain grouping.

I. 서론

대규모 순차 회로의 테스트를 위하여 스캔 플립 플롭(이하, 스캔 셀)을 사용한 스캔 설계가 널리 사용되고 있다.^[1] 스캔 설계에서는 스캔 셀들을 사용하여 주어진 순차 회로를 그의 테스트를 위하여 조합회로 형태로 바꾸어 주며, 아울러 스캔 셀들을 직렬로 연결하

여 테스트용 데이터를 입력하고 그 테스트 결과를 출력 받는다. 여기서, 이러한 직렬 연결된 스캔 셀들을 스캔 체인이라고 부른다.

하나의 스캔 체인내의 서로 인접한 스캔 셀들 간의 연결길이는 가급적 짧게 하여야 한다.^[2] 이는 스캔 체인 셀간의 스큐(skew) 등의 문제로 인한 테스트시 오동작을 방지하기 위함이다. 그러나, 스캔 설계를 마친 후 물리 설계(physical design) 단계인 배치 및 배선 단계를 거치면, 멀리 떨어진 연결선이 존재하는 것을 피하기 어려우며 인접한 스캔 셀 사이에도 그 위치가 서로 멀리 떨어질 가능성이 있다. 일반 연결선의 경우에

* 正會員, 西江大學校 컴퓨터學科

(Dept. of Computer Science, Sogang University)

接受日字:2001年3月26日, 수정완료일:2002年7月24日

는 배치를 다시 수행하기 전에는 이를 수정하기가 어려우나, 스캔 셀의 경우 스캔 체인에서 이들의 위치를 서로 바꿀 수 있으므로 스캔 셀간의 연결이 짧도록 그들의 순서를 다시 조정할 수 있다. 이렇게 스캔 체인에서 스캔 셀간의 순서를 바꾸어 연결선의 길이를 짧게 하기 위한 문제를 '스캔 체인 재구성 문제'라고 한다. 스캔 체인 재구성을 통하여 추가로 얻을 수 있는 이득은 스캔 셀간의 연결 거리가 짧아지게 되므로 배선 복잡도(congestion)가 작아져 IC 칩 전체 배선의 성공 가능성을 보다 높게 할 수 있다.

스캔 체인 재구성 문제에 관한 여러 연구 결과가 지금까지 발표되었다. Freuer와 Koo는 스캔 체인 재구성 문제가 Travelling Salesman Problem(TSP)과 유사한 문제임을 밝히고 이 문제를 symmetric TSP로 변환하여 해결하는 방법을 제시하였다.^[8] 그러나, 그들의 방법은 방문 도시가 2배로 늘어나 수행 시간이 클 것으로 짐작되며 실제 그들의 실험 결과는 보이지 않았다. Aroa 등은 레이아웃에 기반을 둔 스캔 체인 재구성 방법을 제안하였다.^[2] 그러나, 작은 수의 플립 플롭을 포함하는 스캔 체인을 재구성하는 것도 많은 수행 시간이 걸려 규모가 큰 회로에 대해서는 적합해 보이지 않는다. Nakamura는 스캔 체인 재구성 문제를 풀기 위하여 TSP의 해결을 연속적으로 반복 수행하는 방법을 사용하였다.^[11] 논문에서는 구현의 방법이 자세히 설명되어진 것은 아니며, 실제 플립 플롭의 수가 적은 회로들에 대한 결과만을 보였다. Hirech 등은 PDEF 파일을 통해 얻어진 스캔 셀들의 물리적인 좌표를 이용하여 스캔 체인 재구성 방법을 실용적인 면에서 제안하였다.^[5]

시스템의 규모가 커짐에 따라 하나의 스캔 체인내의 셀들을 한 개 이상의 클럭으로 구동하여야 할 경우가 있을 수 있다.^[13] 이 경우 스캔 체인의 셀들을 그 구동 특성이 동일한 것끼리 각각 모아서 한 개 이상의 그룹으로 분류할 수 있는데 이러한 스캔 체인을 '그룹화된 스캔 체인'이라고 부른다. 그룹화된 스캔 체인을 재구성하는 방법으로 최근 Makar가 발표한 방법이 있다.^[10] Makar는 플립플롭들의 그룹별 연결은 물론 계층별 연결도 고려하였으나, 그의 방법은 스캔 체인 내의 그룹 순서를 바꿀 수 없는 경우만을 고려한 것이다.

본 논문에서는 보다 일반화된 그룹화된 스캔 체인을 재구성하는 방법을 제안한다. 즉, 그룹화된 스캔 체인에서 그룹의 순서를 조정할 수 있는 경우 이러한 스캔

체인을 재구성하는 방법을 제안한다. 제안한 방법은 TSP를 위한 휴리스틱인 2-OPT, 3-OPT를 응용한 방법으로, 그룹 순서를 조정할 수 없도록 입력을 제한할 경우 Makar의 방법을 그대로 구현하여 그 결과를 비교하였을 때 연결 거리는 평균 13.6% 정도 개선되었다.

서론에 이어 본 논문의 제 2 장에서는 그룹화된 스캔 체인 및 그 재구성 문제를 정의하고 제 3 장에서는 그룹화된 스캔 체인을 재구성하는 방법을 제안한다. 제 4 장에서는 실험 결과를 기술하며 마지막으로 제 5 장에서 결론을 보인다.

II. 그룹화된 스캔 체인

1. 스캔 셀의 그룹화

대규모 시스템의 스캔 테스트를 위한 설계에서 스캔 체인의 클럭 소스를 하나 이상 사용할 수도 있다. 이는 스캔 체인의 셀들의 수가 많아 지역적으로 넓게 분포될 경우 하나의 클럭 소스로 인하여 야기되는 클럭 스큐가 커져 스캔 체인의 정상적인 동작을 기대할 수 없을 경우가 있기 때문이다. 하나의 스캔 체인에서 클럭 스큐가 유사하여 임의의 순서로 배열하여도 스캔 체인의 동작이 보장되는 스캔 셀들의 모임을 '클럭 도메인(domain)'이라고 한다.^[10]

스캔 체인 하나는 여러 개의 클럭 도메인으로 구성될 수 있다. 스캔 체인이 멀티 클럭으로 동작할 경우 각 클럭에 연결된 셀들이 하나씩의 클럭 도메인을 구성할 수 있다. 단일 클럭으로 동작시킬 경우에도 일부 셀들은 클럭의 rising 에지에서 나머지 셀들은 falling 에지에서 동작시킬 수 있으며 이들 각각 하나씩의 클럭 도메인을 형성한다. 또한 동일 클럭 소스로 동작되되 gated 클럭형태로 구동되는 셀들이 있을 경우에도 이들은 하나의 클럭 도메인으로 생각할 수 있다.

같은 클럭 도메인에 속한 셀들의 순서는 클럭 도메인의 정의에 따라 임의의 순서로 배열하여도 스캔 체인의 동작에 문제가 없다. 그러나, 다른 클럭 도메인에 속한 셀이 한 클럭 도메인의 셀들의 배열 중간에 삽입될 수 없으며, 클럭 도메인의 순서도 클럭 스큐의 형태에 따라 미리 고정되어 서로 바꾸지 못할 수 있다. 그러나, 멀티 클럭 소스의 경우 두 클럭 도메인의 스큐에 따라 그 순서를 바꿀 수도 있으며, 동일 클럭의 경우 작동 에지만 다를 경우에도 클럭 도메인의 배열 순서

를 바꿀 수 있다. 즉, 스캔 체인의 일반적인 형태는 체인내의 셀들이 하나 이상의 클록 도메인으로 분류되어 있으며, 클록 도메인의 배열 순서는 경우에 따라 바꿀 수 있거나 또는 바꿀 수 없을 수도 있다. 그리고, 클록 도메인내의 셀들은 임의의 순서로 배열될 수 있다. 그런데, 앞 장에서 언급한 Marak의 방법은 클록 도메인의 순서가 고정되었을 경우로 한정하여 그 응용범위가 넓지 않으며, 따라서, 스캔 설계자에 의해서 주어질 수 있는 여러 다양한 형태의 스캔 체인을 효과적으로 재구성하는 새로운 방법의 개발이 필요하다.

본 논문의 나머지 부분에서는 동일 클록 도메인에 속한 셀들을 편의상 '스캔 그룹'이라고 정의하여 부르며, 한 개 이상의 클록 도메인으로 구성된 스캔 체인을 '그룹화된 스캔 체인'이라고 표기한다.

2. 그룹화된 스캔 체인 재구성 문제

스캔 체인은 주입력(Primary Input(PI)), 주출력(Primary Output(PO)), 그리고 스캔 체인에 속한 셀들의 집합 R 로 구성되어 있으며, 이미 셀들의 배치가 완료되었다고 가정하여 PI 와 셀, 셀과 셀, 그리고 셀과 PO 사이의 연결 거리가 주어진다. 이러한 스캔 체인을 재구성할 경우 그 구성형태는 PI , R 의 순열, PO 등의 순서로 이루어지는데 스캔 체인 재구성 문제는 가능한 배열 중 전체 연결 거리의 합이 최소인 배열을 구하는 것이다.

그룹화된 스캔 체인 재구성 문제의 목적도 마찬가지로 구해진 배열의 인접한 셀간 연결거리의 전체 합을 최소화시키는 것이다. 그러나 주어진 스캔 그룹, 스캔 그룹간의 순서 제한 등의 스캔 체인에 입력으로 주어진 제약 조건을 만족시켜야 한다. 이러한 셀 순서에 관한 제약 조건을 좀더 수식적으로 기술하면 다음과 같다.

스캔 체인 C 가 n 개의 스캔 그룹 g_1, g_2, \dots, g_n 으로 구성되어 있다고 가정하자. 각 스캔 그룹은 스캔 셀들로 구성되어 있으며 동일 스캔 그룹내의 스캔 셀들은 임의의 순서도 무방하나 다른 스캔 그룹의 셀이 중간에 끼어 있지 않도록 연속적으로 배치되어야 한다. 스캔 그룹의 순서에 대한 제한 조건은 다음과 같이 표식된다:

$$P = [G_1, G_2, \dots, G_m], \text{ 여기서 } G_i = (g_{i_1}, g_{i_2}, \dots, g_{i_n}).$$

위의 식에서 스캔 그룹의 그룹, 즉, G_1, G_2, \dots, G_m 은

그 순서를 그대로 유지하여야 함을 의미하며, 각 G_i 내의 스캔 그룹들은 그들 사이에 순서를 임의로 바꾸어도 무방하다는 의미이다.

예를 들어 여섯 개의 스캔 그룹 g_1, g_2, \dots, g_6 가 다음과 같은 그룹순서에 관한 제약조건 $P = [(g_2), (g_1, g_3), (g_4, g_5, g_6)]$ 으로 주어 졌다면, 그룹 g_2 는 스캔 체인의 맨 앞에 항상 위치하여야 하며 그 다음 g_1 또는 g_3 가 올 수 있다. 마지막으로, g_1 과 g_3 다음에 g_4, g_5, g_6 가 임의의 순서로 배열되면 된다. 예를 들어 $[g_2, g_3, g_1, g_5, g_4, g_6]$ 의 형태로 구성된 스캔 체인은 주어진 제약 조건을 만족하나, $[g_2, g_3, g_4, g_1, g_5, g_6]$ 는 만족하지 않는다. 만일, 그룹간의 제한조건이 $P = [(g_1, g_2, \dots, g_6)]$ 로 주어진다면 그룹간의 순서 제한이 존재하지 않는다는 의미이며, 따라서 임의의 스캔 셀 그룹 배열도 가능하다.

이러한 기본적인 설명을 바탕으로 그룹화된 스캔 체인의 재구성문제를 정의하면 PI, PO, R, P 가 주어졌을 때 셀 간의 연결거리의 전체 합이 최소가 되면서도 주어진 제약조건 P 를 만족하도록 스캔 체인의 스캔 셀 순서를 결정하는 것이다.

주어진 스캔 체인 전체가 하나의 스캔 그룹으로 구성되어 있을 경우 스캔 체인 재구성 문제는 TSP와 유사하다. 즉, PI, PO, R 의 셀들을 정점으로한 완전 그래프(complete graph)를 구성하고 두 정점간의 에지 비용을 이에 대응하는 두 셀간의 연결거리라고 할 때, 스캔 체인 재구성 문제는 PI 부터 시작하여 PO 까지 R 의 각 셀에 대응하는 모든 정점을 한번씩 방문하는 방문중 경유한 에지 비용의 합이 최소인 경로를 찾는 문제이다. 이 문제는 PI 와 PO 에 해당하는 정점 사이에 비용이 $-M$ 인 에지를 삽입하면 일반 TSP의 해는 스캔 체인 재구성 문제의 해와 동일하다. 여기서, M 은 매우 큰 값이다. 그러나, TSP는 NP-hard 문제[4]이므로 이를 해결하기 위하여서는 일반적으로 휴리스틱 방법을 사용한다.

그룹화된 스캔 체인 재구성 문제의 경우 각 스캔 그룹에 대해서 스캔 그룹내의 시작 셀과 마지막 셀이 결정되면 그룹 내에서의 재구성은 TSP 휴리스틱을 이용하여 해결할 수 있다. 그러나, 두 개의 그룹 g_i 와 g_j 를 연결하기 위하여 g_i 에서의 마지막 셀과 g_j 에서의 시작 셀을 결정하는 문제는 쉽게 해결할 수 없다. 그

이유는 그룹간의 연결을 짧게 할 경우 그룹내 스캔 셀들의 연결 길이가 증가할 수 있으며, 반대로 그룹내 스캔 셀들의 연결 길이를 짧게 할 경우 스캔 그룹간의 연결길이가 증가할 수 있기 때문이다.

예를 들어 두 개의 스캔 그룹 g_1 , g_2 로 구성된 스캔 체인이 있다고 가정하자. g_1 , g_2 의 스캔 셀의 개수를 각각 k_1, k_2 라고 하고 $P = [(g_1, g_2)]$ 라고 하자. 그리고 스캔 그룹의 시작 셀과 마지막 셀이 정해졌을 경우 최적의 스캔 그룹내 스캔 셀의 순서를 정할 수 있는 알고리즘 A 가 있다고 가정하자. 이 상황에서 최적의 재구성 해를 계산시간을 고려하지 않고 구하고자 한다면, 스캔 그룹 g_1 의 셀중 모든 가능한 시작 셀과 마지막 셀을 선택하여 알고리즘 A 를 수행하고 그룹 g_2 에 대해서도 마찬가지로 A 를 수행하여 전체 연결길이가 가장 작은 셀 배열을 구한다면 알고리즘 A 를 $O(k_1^2 + k_2^2)$ 만큼 수행하여야 한다. 그룹수가 많을 경우 이 수행 회수는 더욱 더 증가할 것이고 이러한 예를 통하여 그룹화된 스캔 체인의 재구성 문제가 해결하기 아주 어려운 문제라고 여길 수 있다.

이어지는 다음 장에서는 그룹화된 스캔 체인의 재구성 문제를 스캔 그룹의 순서를 먼저 결정하고 다음 각 스캔 그룹내의 스캔 셀 순서를 정하는 두 단계에 걸친 스캔 셀 재구성 방법을 제안 기술한다.

III. 그룹화된 스캔 체인 재구성

제 2 장에서 언급한바와 같이 본 논문에서 제안하는 재구성 방법은 두 단계로 구성된다. 먼저, NN(Nearest Neighbor) 탐색[7]을 적용하여 스캔 그룹의 순서를 결정한 후, TSP 휴리스틱인 2-OPT, 3-OPT 방법^[11]을 적용하여 각 스캔 그룹내의 스캔 셀들의 순서를 결정한다. 물론, 스캔 그룹 순서와 스캔 그룹내 스캔 셀들의 순서를 분리하여 고려하는 것은 최적화된 스캔 체인 재구성 결과를 얻는데 장애가 될 수 있다. 그러나 스캔 그룹은 클록 스큐 등을 고려하여 구성되어 있기 때문에 각 스캔 그룹의 셀들은 지역적으로 몰려 있는 경향이 있어 NN 탐색에 의한 그룹순서가 어느 정도 정확하며, 스캔 그룹내의 스캔 셀들의 순서를 결정할 때 그룹간의 연결을 고려하여 스캔 그룹내의 시작 셀과 마지막 셀을 선택할 수 있게 하였으므로 비교적 우수한 결과를 얻을 수 있다.

이어지는 3.1, 3.2 절에서는 스캔 그룹 순서를 결정하는 과정과 스캔 그룹내의 스캔 셀들의 순서를 결정하는 방법을 자세히 기술한다.

1. 스캔 그룹 순서 결정

NN 탐색^[7]은 스캔 체인 재구성 문제에 적용할 수 있다. 먼저 주어진 스캔 체인의 PI 부터 시작하여 PI 와 가장 가까운 스캔 셀 c 를 선택하여 PI 와 연결될 스캔 셀로 결정한다. 다음, c 와 가장 가까운 아직 선택되지 않은 스캔 셀을 선택하여 c 의 다음 스캔 셀로 결정한다. 이러한 과정을 모든 셀들의 순서가 결정될 때 까지 반복한 후 마지막으로 PO 를 스캔 체인의 마지막 순서로 결정한다.

NN 탐색에 의한 스캔 체인 재구성 방법으로는 일반적으로 최적화된 결과를 얻기 어렵다. 특히, 마지막으로 선택된 셀 c_i 과 PO 와의 거리가 클 수 있는데, 이는 c_i 과 PO 와의 연결은 이 연결 외의 다른 선택이 없기 때문이다. 그러나, NN 탐색은 신속하게 그리 나쁘지 않은 재구성 결과를 얻을 수 있어, 스캔 그룹 순서를 결정하는데 사용할 수 있다. 스캔 그룹 순서는 스캔 셀들의 상세한 순서를 정한다기 보다는 그룹의 상대적인 위치를 고려하여 개략적인(global한) 스캔 셀들의 순서를 구함으로써 결정할 수 때문이다.

스캔 그룹의 순서는 NN 탐색으로 스캔 체인의 스캔 셀 순서를 구하는 과정을 통하여 얻어진다. 스캔 그룹에 대한 제한이 다음과 같다고 하자:

$$P = [G_1, G_2, \dots, G_m], \text{ 여기서 } G_i = (g_{i1}, g_{i2}, \dots, g_{in}).$$

먼저 스캔 체인의 PI 부터 시작하여 G_1 에 속한 스캔 그룹의 스캔 셀 중 가장 가까운 스캔 셀 c_{s1} 을 선택한다. 스캔 셀 c_{s1} 가 G_1 의 스캔 그룹 g_{h1} 에 속한다고 하면 g_{h1} 에 속한 모든 스캔 셀들의 순서를 스캔 셀 c_{s1} 을 시작 셀로 하여 NN 탐색으로 결정한다. 이 과정이 완료되면 G_1 에 속한 g_{h1} 를 제외한 나머지 그룹의 셀중에서 g_{h1} 의 마지막으로 선택된 스캔 셀과 가장 가까운 스캔 셀을 선택하여 다음 시작 스캔 셀로 정하고 그 스캔 셀이 속한 그룹의 모든 스캔 셀들의 순서를 NN 탐색으로 결정한다. 이와 같은 과정을 반복하여 G_1 에 속한 모든 스캔 그룹에 속한 스캔 셀들의 순서를 결정하면, G_1 에 속한 스캔 그룹의 순서가 동시에

결정된다.

G_1 의 스캔 셀 순서와 스캔 그룹의 순서가 결정되면 이때 마지막으로 선정된 스캔 셀을 시작으로 G_2 의 모든 스캔 셀들의 순서를 G_1 에서와 마찬가지로 결정하며 이를 통하여 G_2 의 스캔 그룹 순서를 정하게 된다. 그리고 같은 과정을 G_3, G_4, \dots 등의 순서로 차례로 적용하여 스캔 셀의 순서 및 스캔 그룹의 순서를 결정한다. 그리고, 이때 결정된 스캔 셀의 순서는 다음 2 단계에서 수행하는 스캔 그룹내의 스캔 셀 순서를 최적화 하는데 초기 순서로 사용된다.

2. 스캔 그룹 셀 재구성

스캔 체인 재구성의 제 2 단계에서는 제 3.1 절에서 기술한 제 1 단계 방법에서 생성된 스캔 그룹의 순서를 유지하면서 각 스캔 그룹내의 스캔 셀 순서를 재조정하여 연결길이를 보다 최소화시킨다. 이를 위하여 첫 단계의 실행시 구성한 스캔 셀 순서를 초기 순서로 하고 각 스캔 그룹별로 2-OPT와 3-OPT를 우리의 목적에 맞게 변형하여 적용한다. 이어지는 소절에서는 2-OPT와 3-OPT 방법을 간단히 설명하고 다음 소절에서는 이를 이용한 각 스캔 그룹내 스캔 셀의 순서를 결정하는 방법을 기술한다.

(1) 2-OPT와 3-OPT

2-OPT와 3-OPT 휴리스틱은 TSP의 해를 구하는 방법으로서 가장 많이 알려져 있는 방법으로 이를 변형한 여러 다른 방법이 제안되었다.^[7] 2-OPT, 3-OPT의 입력으로는 이차원 평면상에 놓여진 정점들이며 임의의 한 정점에서 다른 모든 정점으로의 방문이 가능하다고 가정한다(즉, complete 그래프). 두 정점사이의 거리는 그들간의 Euclidean 거리로 정의되나 본 논문에서는 이들간의 rectilinear 거리로 간주한다.

TSP에 대한 방문 경로 하나가 주어졌을 때 2-OPT와 3-OPT는 이 경로에서 각각 두 개 또는 세 개의 에지를 보다 거리가 짧은 동 수의 다른 에지로 교체하는 방법이다. 그림 1과 그림 2에 이러한 에지 교체 과정을 설명한다.

그림 1은 2-OPT 과정을 보인 것으로 에지 BA와 CD를 기존 경로에서 제거한 후 에지 CB와 DA를 추가하여 새로운 경로를 얻는 것이다. 이때 D에서 B로 향하는 기존 경로는 새로운 경로에서는 B에서 D로 방향이 바뀌게 되나 경로 길이에는 영향을 주지 않는다.

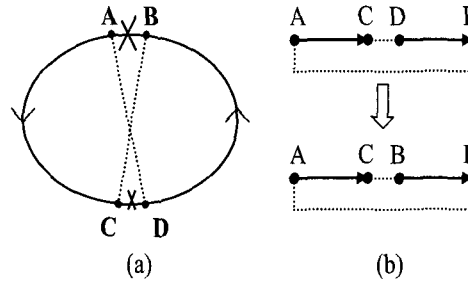


그림 1. 2-OPT 이동의 예
Fig. 1. 2-OPT Movement.

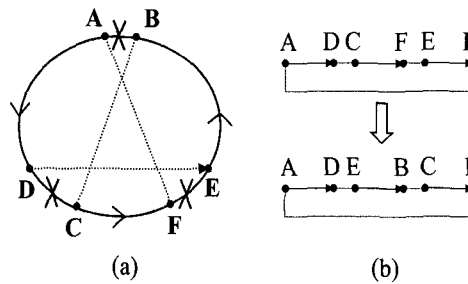


그림 2. 3-OPT 이동의 예
Fig. 2. 3-OPT Movement.

그림 2는 3-OPT 과정을 보인 것이다. 3-OPT는 기존 경로에서 세 개의 에지(즉, 그림에서 BA, DC, FE)를 제거하고 세 개의 다른 에지(즉, 그림에서 DE, BC, FA)를 추가하는 과정이다. 그런데 이 경우에는 기존 경로의 방향은 바뀌지 않는다.

이러한 2-OPT와 3-OPT 과정은 TSP의 해를 얻기 위하여 그리디(greedy)하게 더 이상 개선이 없을때까지 반복적으로 이들을 사용하거나 또는 시뮬레이티드 어닐링(simulated annealing)에 의한 방법에서의 이동 방법으로 사용할 수 있다. 본 논문에서는 반복적 개선 방법을 사용하는데, 2-OPT의 경우 4 개의 정점(스캔 셀), 3-OPT의 경우에는 6 개의 정점을 일련의 순서에 의하여 차례로 선정하여 새로운 경로의 전체 길이가 작아지는 경우에만 에지 교체를 수행하는 방법을 사용한다. 여기서, 에지를 교체할 경우 전체 거리가 작아지는지를 빠르게 조사하는 방법이 참고문헌 [3]에서 제안되었는데, 이를 사용할 경우 보다 빠르게 2-OPT와 3-OPT 과정을 반복적으로 적용할 수 있다.

(2) 스캔 그룹의 스캔 셀 순서 재구성

하나의 스캔 그룹에서 스캔 셀 순서를 재구성하는 문제는 TSP와 다소 차이가 있다. TSP는 사이클

(cycle), 즉, 한 정점에서 모든 다른 정점을 한번씩 경유하여 자기 자신으로 되돌아오는 경로를 구하는 문제임에 반하여 우리의 문제는 스캔 그룹의 한 정점에서 출발하여 모든 다른 정점을 한번씩 방문하는 경로를 구하는 것이다. 2-OPT와 3-OPT는 주어진 사이클을 개선하는 방법으로 우리의 문제에 그대로 적용하기는 곤란하다.

이러한 문제는 현재 순서를 구하고자하는 스캔 그룹에 대하여 시작 셀과 마지막 셀이 정해져 있다면 쉽게 해결할 수 있다. 즉, 시작 셀 c_s 와 마지막 셀 c_l 사이의 연결 거리를 다른 모든 두 정점사이의 연결거리와는 다르게 $-M$, M 은 매우 큰 수, 으로 설정하면 2-OPT와 3-OPT를 적용한 후 얻어진 사이클에서 c_s 와 c_l 은 항상 인접한 형태가 된다. 따라서, c_s 로부터 인접한 c_l 의 반대 방향으로 사이클을 따라 c_l 까지 얻어진 스캔 셀들의 순서를 구하면 된다.

주어진 스캔 체인이 n 개의 스캔 그룹으로 구성되어 있고 앞에서 기술한 스캔 그룹 순서를 정하는 단계를 거쳐 얻어진 스캔 그룹 순서를 g_1, g_2, \dots, g_n 이라고 하자. 스캔 그룹 g_i 에 속한 스캔 셀들의 순서를 정하기 위해서는 이들 스캔 셀 중 시작 스캔 셀과 마지막 스캔 셀을 결정하는 것이 필요하다. 그러나 스캔 그룹 g_i 의 시작 및 마지막 스캔 셀을 결정하는 것은 g_i 와 인접한 스캔 그룹 g_{i-1} 과 g_{i+1} 의 스캔 셀 순서와 무관하지 않다. 즉, g_i 의 내부 연결거리만을 고려하여 시작(마지막) 스캔 셀을 결정할 경우 결정된 시작(마지막) 스캔 셀과 스캔 그룹 g_{i-1} (g_{i+1})의 마지막(시작) 스캔 셀과 멀리 떨어지게 되어 스캔 체인 전체 연결거리를 작게 하는데 나쁜 결과를 초래할 수 있다. 그런데, 더욱 문제가 되는 것은 각 스캔 그룹의 시작 스캔 셀과 마지막 스캔 셀이 정해지지 않았기 때문에 모든 스캔 그룹 전체를 고려하여 각 스캔 그룹의 시작 스캔 셀과 마지막 스캔 셀들을 결정하기가 대단히 어렵다는 점이다. 한편, 제 1 단계 과정에서 수행한 NN 탐색에 의하여 스캔 셀들의 대략적인 순서가 결정되어 있는데 이를 토대로 각 스캔 그룹의 시작 및 마지막 스캔 셀을 결정할 수도 있으나 이는 그리 좋은 방법이 아니다.

이러한 문제를 해결하기 위하여 본 논문에서 제안하는 방법은 다음과 같다. 스캔 그룹 g_i 의 스캔 셀들을 재구성할 때 스캔 체인의 전체 스캔 셀 순서가 초기

```

입력:  $PI, PO$ , 제 1 단계에서 구한 스캔 체인의 스캔 셀
      순서 및 스캔 그룹 순서  $g_1, g_2, \dots, g_n$ .
출력: 최적화된 스캔 체인의 스캔 셀 순서.
1.  $C_{old} =$  매우 큰 값;
2.  $C_{new} =$  초기 스캔 셀 순서에 대한 스캔 셀간 거리 합;
3. while(  $C_{old} > C_{new}$  ) {
4.    $C_{old} = C_{new}$ ;
5.   for (  $i=1, 2, \dots, n$  ) {
6.     if (  $i==1$  )  $c_s = PI$ ;
7.     else  $c_s = g_{i-1}$ 의 마지막 셀;
8.     if (  $i==n$  )  $c_l = PO$ ;
9.     else  $c_l = g_{i+1}$ 의 시작 셀;
10.     $c_s, c_l$ 을 각각 시작 및 마지막 스캔 셀로 고정된
        상태에서,
11.    do {
12.      3-OPT 과정을 통한  $g_i$ 의 스캔 셀 순서 재구성;
13.      2-OPT 과정을 통한  $g_i$ 의 스캔 셀 순서 재구성;
14.       $C_{new}$ 는 단계 11, 12에서 계속 갱신된다.
15.    } while ( 연결거리 단축이 있는 동안 );
16.  }
17. }
    
```

그림 3. 그룹내 스캔 셀 재구성을 통한 스캔 체인의 최적화 과정
 Fig. 3. Scan chain optimization procedure by scan cells reordering within each group.

순서로 주어진다. 이는 제 1 단계에서 정한 순서일 수도 있고 최적화 과정 중 중간단계의 스캔 셀 순서일 수도 있다. 이러한 스캔 셀 순서에서, 스캔 그룹 g_{i-1} 에 속한 스캔 셀 중 마지막 스캔 셀 $c_{g_{i-1}}$ 과 스캔 그룹 g_{i+1} 의 시작 스캔 셀 $c_{g_{i+1}}$ 을 각각 시작 스캔 셀과 마지막 스캔 셀로 하고 스캔 그룹 g_i 의 스캔 셀들의 순서를 $c_{g_{i-1}}, c_{g_{i+1}}$ 과 함께 2-OPT와 3-OPT를 적용하여 결정한다. 즉, g_i 에 속한 스캔 셀들의 시작 스캔 셀과 마지막 스캔 셀을 그룹 g_{i-1} 과 g_{i+1} 의 현재 상태에 따라서 유동적으로 결정하고 아울러 스캔 그룹 g_i 의 스캔 셀 순서를 최적화 하는 것이다. 그리고 이러한 과정을 반복적으로 모든 스캔 그룹에 대하여 더 이상 개선이 없을 때까지 반복 수행하여 모든 스캔 그룹의 시작 스캔 셀과 마지막 스캔 셀을 결정함과 동시에 스캔 그룹내의 스캔 셀의 순서를 최적화하고 스캔 그룹간의

연결거리 또한 자연스럽게 고려하게 된다.

그림 3에 이러한 스캔 그룹내의 셀들의 순서를 정하는 과정을 보인다. 그림에서 3-OPT와 2-OPT를 함께 사용한 것은 두 방식이 모두 휴리스틱 방법이므로 보다 나은 결과를 얻기 위하여 상호 보완할 수 있도록 하기 위함이다.

IV. 실험 결과

본 논문에서 제안한 그룹화된 스캔 체인 재구성 방법을 C 언어를 사용하여 Solaris 환경에서 수행되도록 구현하였다. 실험에 사용된 컴퓨터는 Sun Ultrasparc 60이다. 본 장에서는 먼저 실험에서 사용한 입력 자료에 대하여 간단히 설명하고, 다음 기존 연구 결과와의 비교 그리고 자체 실험 결과 등을 기술한다.

1. 실험을 위한 입력자료

제안한 재구성 방법을 평가하기 위하여 국내 모 기업에서 설계한 자료 중 코어의 크기와 스캔 체인의 개수 그리고 각 체인에 포함된 스캔 셀들의 위치 등만 포함된 자료를 PDEF 형태로 구하여 이를 토대로 실험을 수행하였다. 따라서, 실험에 사용한 시스템의 기능 및 기타 다른 정보를 알지 못하는 상태이므로, 스캔 체인의 셀들을 임의로 분할 하여 스캔 그룹을 생성하였으며, 스캔 그룹에 관한 위치 제약도 임의로 구성하였다. 표 1에 실험에서 사용한 자료의 특성을 보인다. 기본적으로 스캔 셀의 개수가 각각 775, 650, 600개인 설

계 자료를 스캔 셀 그룹 생성 그리고 그룹의 순서 제약 등에 따라 총 9 개의 시험자료를 생성하여 실험을 수행하였다.

본 논문에서 제안한 방법의 입력 조건과 동일한 대상을 다른 기존 발표된 방법 및 실험 결과가 없기 때문에 스캔 그룹을 다루되 그 순서가 고정된 스캔 체인을 재구성하는 방법인 Makar의 방법과 비교를 수행하였는데 이를 위하여 생성한 자료는 $c_testi, i = 1, 2, \dots, 5$, 이다. 그리고 $g_testi, i=1, 2, 3, 4$, 는 그룹순서를 바꿀 수 있는 제약조건을 갖도록 생성한 자료이며 이는 순전히 제안한 방법을 실험하기 위한 자료이다.

2. 스캔 체인 재구성 결과

본 논문에서 제안한 그룹화된 스캔 체인의 재구성 방법의 성능을 시험하기 위하여 제안한 방법을 최근 Markar가 제안한 방법과 그 결과를 비교하였다. 그러나 앞장에서 언급한 바와 같이 Markar가 해결하고자 하는 문제는 본 논문에서 고려한 문제보다 다소 제한된 형태이다. 즉, Markar는 스캔 체인의 스캔 셀들이 그룹화 되어 있기는 하되 그 그룹 순서가 이미 고정되어 있는 경우 스캔 셀들의 순서를 정하는 방법을 제안하였다. 그의 방법은 NN 탐색에 기반한 방법으로 스캔 셀들을 NN 탐색에 의하여 하나씩 선택하여 순서를 정하되 이 순서가 그룹 순서를 위반하지 않도록 정하는 방법으로, 우리의 그룹순서를 결정하는 방법과 유사하다. 그런데, 실제 그의 프로그램을 구하지 못하여 발표된 논문의 내용을 토대로 직접 구현하여 비교를 수행

표 1. 입력 데이터들의 특성

Table 1. Input data characteristics.

회로	스캔 그룹 수	스캔셀 수	그룹 순서 제한
c_test1	1	775	$[(g_1)]$
c_test2	2	775	$[(g_1), (g_2)]$
c_test3	3	600	$[(g_1), (g_2), (g_3)]$
c_test4	6	775	$[(g_1), (g_2), \dots, (g_6)]$
c_test5	10	650	$[(g_1), (g_2), \dots, (g_{10})]$
g_test1	2	775	$[(g_1, g_2)]$
g_test2	3	600	$[(g_1, g_2), (g_3)]$
g_test3	6	775	$[(g_1, g_2), (g_3, g_4, g_5), (g_6)]$
g_test4	10	650	$[(g_1, g_2, g_3), (g_4, g_5, g_6), (g_7, g_8), (g_9, g_{10})]$

하였다.

표 2. Makar의 방법과 제안된 방법과의 비교
Table 2. Comparative results between ours and Makar's.

회로	초기 연결 길이	Makar의 방법		제안한 방법		Makar 방식 대비 감소비율
		연결 길이	수행시간(초)	연결 길이	수행시간(초)	
c_test1	69900.5	39414.6	9.24	33507.2	8.64	15%
c_test2	61569.1	37889.1	9.27	31708.6	4.20	17%
c_test3	76312.7	43573.9	9.40	40288.1	1.60	8%
c_test4	69911.6	48633.4	9.42	41981.7	1.12	14%
c_test5	74011.1	52750.1	9.61	45807.3	0.66	14%

표 2에 본 논문에서 제안한 방법과 Markar의 방법에 의한 결과를 보인다. 표에서 초기 연결 길이는 스캔 셀들을 입력파일에서 입력한 순서대로 스캔 체인을 구성하였을 때의 연결길이이다. Markar의 방법은 전체 셀을 대상으로 NN 탐색을 수행하는 방식이고 회로의 셀 수가 서로 유사하기 때문에 비슷한 수행시간이 소요된다. 그러나 본 논문에서 제안한 방법은 그룹 순서를 구하고 각 그룹내의 스캔 셀들의 최적화를 수행하기 때문에 같은 전체 스캔 셀 수에 대하여 스캔 그룹의 수가 증가할수록 그 수행시간은 짧아진다. 또한 전체 연결 길이도 Markar의 방법에 비하여 평균 약 13.6% 정도의 개선된 결과를 얻었다.

표 3에 그룹순서를 바꿀 수 있는 경우에 대한 실험 결과를 보인다. 표에서는 전체 연결거리가 어떻게 달라졌는지를 보이고 아울러 결정된 그룹 순서도 보였다. 비록 이 결과를 비교할 만한 기존 다른 발표된 방법이

없이 정확한 평가는 하기 어렵지만 표 2의 비교결과를 볼 때 표 3에 보인 결과도 우수한 결과일 것이라고 짐작된다.

마지막으로 그림 4와 5에 회로 g_test1에 대한 최적화 실행 전과 후의 스캔 체인 연결 상태를 보인다.

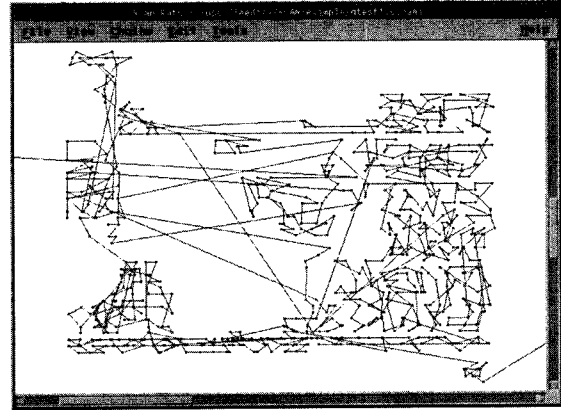


그림 4. 스캔 체인 g_test1의 초기 스캔 셀 배열 순서
Fig. 4. Initial scan cell orders of g_test1.

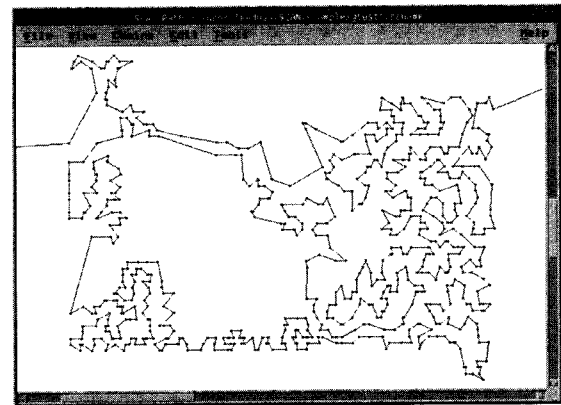


그림 5. 스캔 체인 g_test1의 최적화 후 스캔 셀 연결 순서
Fig. 5. Scan cell reordering result after optimization.

표 3. 스캔 그룹 순서를 바꿀 수 있을 경우에 대한 실험 결과
Table3. Experimental results when scan groups can be reordered.

회로	초기 연결길이	연결길이	그룹 정보	최종 그룹 순서	수행시간
g_test1	61569.0	40874.5	{(1,2)}	(2,1)	4.74s
g_test2	76312.7	39847.2	{(1,2),(3)}	(1,2,3)	1.95s
g_test3	69900.5	40235.2	{(1,2),(3,4,5),(6)}	(1,2,4,5,3,6)	1.36s
g_test4	74011.1	44602.4	{(1,2,3),(4,5,6),(7,8),(9,10)}	(2,1,3,4,6,5,7,8,9,10)	0.82s

V. 결 론

본 논문에서는 멀티 클록 등 여러 이유로 스캔 셀들이 그룹화 된 경우의 스캔 체인 최적화 문제를 다루었다. 여러 다양한 형태로 주어질 수 있는 스캔 그룹의 순서에 관한 제약조건을 위반하지 않는 범위에서 인접한 스캔 셀간의 연결 거리의 합을 작게 하기 위한 스캔 체인 재구성 방법을 제안하였다. 제안한 방법은 스캔 그룹 순서가 고정된 경우만 다루는 기존 방법과 그 결과를 비교하였을 때 평균 13.6% 정도 연결 길이가 감소하였다. 또한, 본 논문은 저자의 조사로는 그룹화된 스캔 체인 재구성 문제를 처음 제안하고 그 해결 방법을 제시한 논문이라고 할 수 있으며, 제안한 재구성 방법이 다루는 스캔 체인 형태는 시스템의 규모가 커짐에 따라 좀더 유용하게 활용될 수 있을 것이다.

본 논문에서 제안한 재구성 방법을 그룹 순서를 먼저 결정하고 다음 그룹내 스캔 셀 순서를 재구성하여 연결 길이의 합이 작은 스캔 체인을 구하는 것이다. 그런데, 앞으로 이 두 과정을 통합하여 그룹 순서와 동시에 스캔 셀의 순서를 동시에 구하는 좀더 우수한 재구성 방법이 필요하다고 할 수 있다. 이를 위하여 시뮬레이션 어닐링 또는 유전자 알고리즘 등의 기법을 사용한 재구성 방법 등을 생각 할 수도 있을 것이다.

참 고 문 헌

- [1] M. Abramovici, M. A. Breuer., & A. D. Friedman, Digital Systems Testing and Testable Design, *IEEE Press*, 1990.
- [2] R. Aroa & A. Albicki, "Computer-Aided Scan Path Design For Self-Testing Chips," *Proc. of 29th Midwest Symp. on Circuits and Systems*, pp. 301~304, Aug 1987.
- [3] K. D. Boese., A. B. Kahng., & S. Muddu. "A New Adaptive Multi-Start Technique for Combinatorial Global Optimizations", *Operations Research Letters*, 16(2), Sept. 1994, pp. 101~113.
- [4] M. R. Garey & D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W, H, Freeman, 1979.
- [5] M. Hirech., J. Beausang., & X. Gu, "A New Approach to Scan Chain Reordering Using Physical Design Information," *Proc. of ITC*, 1998.
- [6] D. S. Johnson, "Local Optimization and the Traveling Salesman Problem," in Proc. of the 17th Int'l Colloquium on Automata Languages and Programming, pp. 446~460, July 1990.
- [7] S. Lin & B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research*, 21, pp. 498~516, 1973.
- [8] Freuer, M and C. Koo, "Method for Rechainning shift Register Latches which Contain more than one Physical Book," *IBM Technical Disclosure Bulletin* Vol. 25, No. 9, pp. 4818~4820, Feb. 1983.
- [9] E. L. Lawler., J.K. Lensta., A. H. G. Rinnoy Kan., & D.B. Shmoys, eds., *The Traveling Salesman Problem*, John Wiley & Sons, 1985.
- [10] S. Makar, "A layout-Based Approach for Ordering Scan Chain Flip-Flops," *Proc. of ITC*, 1998.
- [11] K. Nakaumra, et al., "Scan path's Wire Length Minimization and Its Short Path Error Correction," *NEC Res. And Develop* Vol. 38, No.1, pp. 22~27, Jan. 1997.
- [12] Barbagallo, S. et al. "Scan Insertion Criteria for Low Design Impact," *Proc. of 14th VLSI Test Symposium*, pp. 26-31, 1996.
- [13] J. Schmid & J. Knablein, "Advanced Synchronous Scan Test Methodology for Multi Clock Domain ASICs," *Proc. of 17th VLSI Test Symposium*, 1999.

저 자 소 개

李 政 桓(正會員)

1974년 6월 27일 생. 1999년 2월 서강대학교 전자계산
학과(공학사). 2001년 2월 서강대학교 컴퓨터학과(공학
석사). 2001년 1월~현재 SK C&C IMT 팀 근무. <주
관심분야: CAD, CAD Algorithm, ASIC(통신),
Wireless Network>

林 種 錫(正會員) 第39卷 SD編 3號 參照