# 비트스트림 분석을 활용한 고속의 FPGA 역공학

최소연, 이동훈, 유호영*
충남대학교 전자공학과
e-mail : *sychoi.cas@gmail.com, dhlee.cas@gmail.com, hyyoo@cnu.ac.kr*

## Fast FPGA Reverse Engineering using Bitstream Analysis

Soyeon Choi, Donghun Lee, and Hoyoung Yoo*
Department of Electronics Engineering
Chungnam National University

## Abstract

FPGAs are commonly used in various industrial fields, but methods such as reverse engineering have been proposed as a method for attacking FPGAs. Previously reverse engineering techniques essentially require a mapping table representing a correlation between a netlist and a bitstream, but have a disadvantage in that it takes a very long time to generate a mapping. The clock regions that are distinguished in hardware are clearly appeared in the bitstream. In this paper, we propose a method to reduce the time required to create a mapping tables by simultaneously generating a mapping table for all clock domains. Depending on the number of clock domains, the time required for mapping table generation can be reduced by as little as 8 times and as much as 24 times. The mapping table generation time for Artix-7 A100T, Kintex-7 K355T, and Virtex-7 V355T take 8, 12, and 18 times shorter than previous work.

## I. 서론

FPGA는 ASIC을 이용한 설계 대비 개발 시간과 비용이 적게 든다는 장점으로 다양한 산업 분야에 사용되고 있다 [1]. FPGA를 사용한 산업이 늘어남에 따라 FPGA에 구현된 회로를 공격하는 방법 또한 늘어났으며, 대표적으로 부채널 공격 (side channel attack), 역공학 (reverse engineering) 등이 제안되었다 [2].

FPGA의 역공학은 회로 정보를 포함한 비트스트림을 탈취하여 회로의 넷 리스트를 복원하는 방법이다. 가장 많은 연구가 진행된 Xilinx FPGA의 기존의 역공학 기법은 비트스트림과 넷 리스트 사이의 상관관계를 나타내는 매핑 테이블을 이용한다. 가장 먼저, FPGA의 기본 구성 단위인 타일 별로 매핑 테이블을 확보하고, 매핑 테이블을 이용해 비트스트림을 넷 리스트로 복원한다.

기존의 역공학 기법들은 FPGA 타일별로 매핑 테이블을 순차적으로 생성한다 [2]. 규모가 작은 FPGA의 경우 구성 타일의 수가 수 백개 정도로 매핑 테이블을 모두 확보하는데 수 백일의 시간이 소요되나, 규모가 큰 FPGA의 경우 구성 타일의 수가 수 십만개 정도로 매핑 테이블 확보에만 수십 년의

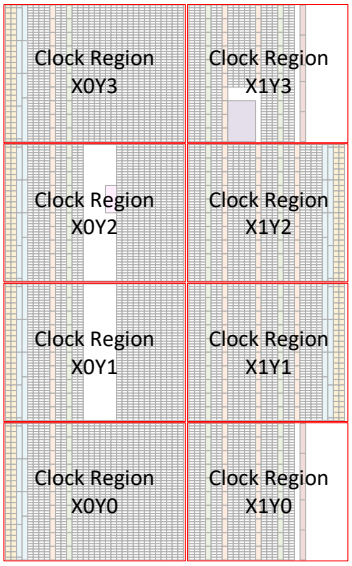


그림 1. Artix-7 A100T의 클럭 영역

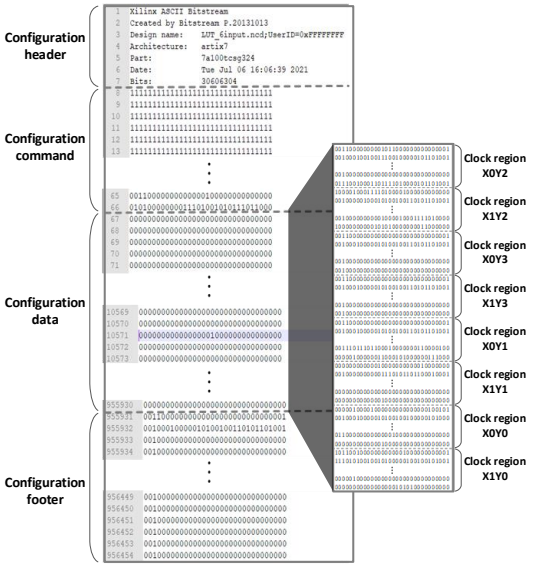


그림 2. 비트스트림 구성

시간이 필요하다. 매핑 테이블 생성 이후 수행되는 역공학 과정은 FPGA의 규모가 커지더라도 수 분 이내에 종료된다. 따라서 역공학 기법을 구현하려면 매핑 테이블 확보에 소요되는 시간을 줄이는 것이 중요하다. 따라서 본 논문에서는 비트스트림을 분석하여 서로 다른 클럭 영역 (clock region)에 있는 타일의 매핑 테이블을 동시에 생성하여 매핑 테이블 생성에 필요한 시간을 줄이고자 한다.

## II. 분석

### 2.1 FPGA 구조

Xilinx FPGA는 타일 단위로 구성되며, 타일은 내부 회로의 구성에 따라 그 종류가 입출력 타일, 논리 회로 구성 타일, 내부 메모리 타일, 클럭 타일 등으로 구분된다. 클럭 타일은 외부로부터 클럭을 입력 받아 다른 타일들로 배분해주는 역할을 수행한다. 이때, 하나의 클럭 타일이 영향을 줄 수 있는 타일의 수와 영역이 클럭 영역 (clock region)으로 고정된다. 하나의 FPGA에는 클럭 타일의 수만큼 클럭 영역이 있으며 FPGA의 종류에 따라 그 수는 최소 8에서 최대 24로 정해진다 [3]. 그림 1은 Xilinx 사의 FPGA 중 Artix-7 계열의 A100T FPGA의 타일 단위 구성도 위에 클럭 영역을 나타낸 것으로 총 8개의 클럭 영역으로 나뉜다.

### 2.2 비트스트림 분석

Xilinx FPGA의 비트스트림의 구성은 그림 2와 같이 구성 헤더 (configuration header), 구성 명령 (configuration data), 구성 데이터 (configuration data), 구성 푸터 (configuration footer)이다. 구성 헤더는 비트스트림 생성에 사용한 EDA 툴 정보, 비트스트림 생성 일시, FPGA 종류, 비트 수에 대한 정보가 기술된다. 구성 명령어는 IDCODE, 구성 데이터의 길이 등을 설정하는 명령어와 그 값으로 구성된다. 구성 데이터는 FPGA의 모든 타일에 대한 정보를 나타낸다. 구성 푸터는 구성 데이터의 CRC 결과, FPGA에 비트스트림이 끝났음을 알려주는 DESYNC를 알리는 명령어와 그 값으로 구성된다.

모든 타일에 대한 정보가 비트스트림에 나타나는 위치를 분석한 결과 클럭 영역 별로 비트스트림에 나타나는 위치가 구분된다. Artix-7계열의 A100T FPGA의 8개 클럭 영역은 비트스트림에서 그림 2와 같이 구분된다. 8개의 클럭 영역은 비트스트림에 순차적으로 나타나지는 않으나 각 영역별로 나타나는 위치는 명확히 구분된다.

## III. 제안하는 방법

기존의 역공학 기법 [2]은 각 타일별로 내부에 구성된 프로그래밍 가능한 포인트를 위한 매핑 테이블이 필요하다 [3]. 매핑 테이블을 만들기 위해서는 먼저 그 타일에 포함된 모든 프로그래밍 가능한 포인트 ($p$)와, 그 포인트의 구성 옵션($op$)을 모두 파악해야 한다. 그 다음, 그 포인트 ($p$)를 사용하지 않은 회로의 비트스트림과 구성 옵션 ($op$)을 사용하도록 설계된 회로의 비트스트림을 생성한다. 마지막으로 그 포인트 ($p$)를 사용하지 않은

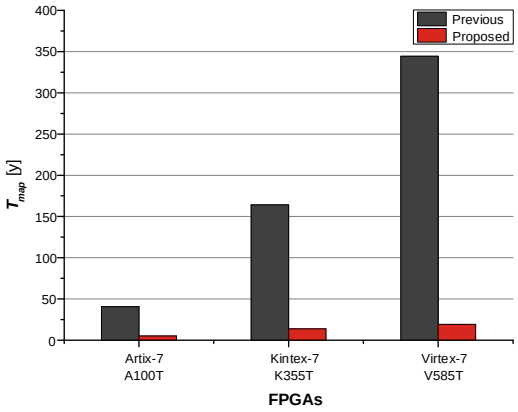


그림 3. 비트스트림 구성

비트스트림과 그 포인트의 ($p$) 구성 옵션 ($op$)을 사용한 비트스트림을 비교한다. 그 결과, 그 포인트 ($p$)의 구성 옵션 ($op$)에 대한 비트 표현을 파악하여 매핑 테이블을 생성할 수 있다.

기존 역공학 기법 [2]은 FPGA에 구성된 모든 타일에 대해 순차적으로 앞의 매핑 테이블 과정을 반복한다. 이 경우 모든 타일의 매핑 테이블 확보에 소요되는 시간 $T_{map}$은 식 (1)과 같이 나타낼 수 있다. 이때, $\mathbf{T}$는 FPGA에 구성된 타일의 집합, $\mathbf{P}$는 그 타일의 프로그래밍 가능한 포인트 ($p$)의 집합, $N_o$는 그 포인트의 구성 옵션 ($op$)의 수를 의미하고, $T_{BIT}$는 비트스트림 1개 생성하는데 필요한 시간을 의미한다.

$$T_{map} = \sum_{t \in \mathbf{T}} \sum_{p \in \mathbf{P}} (N_o \cdot T_{BIT}), \quad (1)$$

FPGA와 비트스트림 구조 분석을 통해 클럭 영역에 따라 비트스트림에 나타나는 위치가 명확히 구분되는 사실을 알 수 있다. 따라서, 각 클럭 영역이 비트스트림에 나타나는 위치를 알고 있다면, 모든 클럭 영역의 매핑 테이블을 동시에 생성하는 것이 가능하다. 따라서 전체 매핑 테이블 생성에 필요한 시간을 식으로 표현하면 식 (2)와 같으며, 이때 $C$는 클럭 영역의 수를 의미한다.

$$T_{map} = \sum_{t \in \mathbf{T}} \sum_{p \in \mathbf{P}} (N_o \cdot T_{BIT}) / C, \quad (2)$$

## V. 실험 결과

Xilinx ISE Design Suite을 이용하여 Artix-7 family의 A100T, Kintex-7 family의 K355T, Virtex-7 family의 V355T의 매핑 테이블 생성에 필요한 시간을 기존 방법 [2]과 본 논문에서 제안하는 방법을 적용한 경우에 대해 비교하였다. 기존의 방법[2] 으로 매핑 테이블을 생성할 경우 매핑 테이블 생성에 A100T는 각각 40년, K355T는 163년, V355T는 344년이 필요하다. 세 가지 FPGA A100T, K355T, V355T는 각각 클럭 영역의 수가 각각 8, 12, 18개 존재하므로 제안하는 방법을 적용하면 기존 연구 [2] 대비 각각 8배, 12배, 18배 짧은 시간인 5.1년, 13.7년, 19.1년만에 전체 매핑 테이블을 확보하여 역공학을 수행할 수 있다.

## Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 FPGA 하드웨어 구조에 기반하여 비트스트림 파일을 분석하고, 그 결과에 따라 역공학에 필수적인 매핑 테이블 생성에 되는 시간을 줄이는 방법을 제안하였다. 제안하는 방법을 적용하면 기존 방법 [2] 대비 최소 8배에서 최대 24배까지 매핑 테이블 생성시간을 줄일 수 있다.

다만, 제안하는 방법을 이용하여 매핑 테이블 생성에 필요한 시간을 줄였다고 하더라도 여전히 하나의 FPGA 칩 역공학에 필요한 매핑 테이블 확보에 수 년의 시간이 소요되므로 이를 줄이는 방법에 대한 연구가 필요하다.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A5A80 26986), Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NO. 2021R1I1A3055806).

## 참고문헌

[1] H. Yu, H. Lee, S. Lee, Y. Kim and H. M. Kim, "Recent advances in FPGA reverse engineering," Electronics, vol. 7, no. 10, 2018.

[2] S. Choi, J. Park and H. Yoo, "Reverse engineering for Xilinx FPGA chips using ISE design tools," *Journal of Integrated Circuits and Systems*, vol. 6, no. 1, 2020.

[3] Xilinx, "7 series FPGAs Clocking Resources (UG471)," July 30, 2018