매핑 규칙 분석 기반의 LUT 매핑 테이블 생성 기법

최소연, 유호영*
충남대학교 전자공학과

LUT Mapping Table Generation Method based on Mapping Rule Analysis

Soyeon Choi and Hoyoung Yoo*
Department of Electronics Engineering
Chungnam National University
E-mail: soyeonchoi@cnu.ac.kr, *hyyoo@cnu.ac.kr

Abstract

In this paper, we propose a method that reduce the number of bitstream files to generate a mapping table for a 4-input LUT by analyzing the mapping rules. In order to extract circuit from the bitstream file, mapping tables for 4-input LUTs are essential. Several methods have been proposed, but it takes a long time to generate 17 or 8 bitstream files. Therefore, in this paper, we propose a method to generate mapping table using only two bitstream files by analyzing the mapping rule of the 4-input LUT. By using the proposed method, the time required to generate a complete mapping table is reduced by more than 78% compared to the previous research. Therefore, it is possible to determine whether the circuit is damaged faster than the previous methods.

I. 서론

FPGA 는 구현된 회로의 재 구성이 가능한 집적회로로 시간적, 비용적 측면의 이점을 가지고 있어 다양한 산업 분야에서 사용되고 있다 [1]. SRAM 기반의 FPGA 는 외부에 넷 리스트 (netlist)를 비트스트림 형태로 저장하기 위한 비 휘발성 메모리가 필요하다. 외부 메모리에 저장된 비트스트림은 공격자로부터 탈취되어 LUT (Look-Up Table)에 구현된 회로 정보가 공격받을 수 있다. 이때, LUT 의 회로를 수정하거나 수정 여부를 파악하기 위해 HDL 로 구현된 회로와 FPGA 의 LUT 사이의 규칙을 알아내기 위한 매핑 테이블 생성 방법이 필수적이므로 이에 대한 연구가 다수 진행되었다 [2, 3].

가장 먼저 제안된 LUT 매핑 테이블 생성 방법 매핑 테이블을 생성하는 방법은 HDL 작성 단계에서 LUT 에 저장되는 초기 값 INIT 값을 one-hot 으로 설정한 비트스트림을 비교하는 것이다 [2]. 이 방법으로 하나의 4-입력 LUT 에 대한 매핑 테이블을 만들기 위해서는 그림 1 과 같이 16 개의 비트스트림을 생성해야 하므로 시간이 오래 걸린다는 단점이 있다. 이후, 넷 리스트 파일과 HDL 파일을 이용하여 매핑 테이블을 생성하는 방법이 제안되었으나, 이 방법으로 하나의 4-입력 LUT 의 매핑 테이블을 생성하기 위해서는 그림 1 과 같이 8 개의 비트스트림이 필요하다 [3].

본 논문에서는 4-입력 LUT 의 매핑 규칙을 분석해 1 개의 비트스트림으로 하나의 4-입력 LUT 의 매핑 테이블을 생성하는 방법을 제안한다.

II. 제안하는 LUT 매핑 테이블 생성 방법

4-입력 LUT 의 매핑 규칙은 4 개의 입력 신호로 가능한 모든 조합을 통해 각 입력 신호의 비트 패턴 [3]에 따라 매핑 규칙이 결정되므로 총 24 가지 존재한다. 매핑 규칙에 의해 비트스트림에서 INIT 의 $2^n(n = 0, 1, 2, 3)$ 번째 비트인 1 번째, 2 번째, 4 번째, 8 번째의 비트가 나타

이 논문은 2021 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2021R1I1A3055806).

# of bitstream files for [2] = 16

File	LUT value	File	LUT value
Verilog	INIT = 16'b0000_0000_0000_0001	Verilog	INIT = 16'b0000_0001_0000_0000
Verilog	INIT = 16'b0000_0000_0000_0010	Verilog	INIT = 16'b0000_0010_0000_0000
Verilog	INIT = 16'b0000_0000_0000_0100	Verilog	INIT = 16'b0000_0100_0000_0000
Verilog	INIT = 16'b0000_0000_0000_1000	Verilog	INIT = 16'b0000_1000_0000_0000
Verilog	INIT = 16'b0000_0000_0001_0000	Verilog	INIT = 16'b0001_0000_0000_0000
Verilog	INIT = 16'b0000_0000_0010_0000	Verilog	INIT = 16'b0010_0000_0000_0000
Verilog	INIT = 16'b0000_0000_0100_0000	Verilog	INIT = 16'b0100_0000_0000_0000
Verilog	INIT = 16'b0000_0000_1000_0000	Verilog	INIT = 16'b1000_0000_0000_0000

# of bitstream files for [3] = 8

File	LUT value	File	LUT value
Verilog	INIT = 16'b0000_0000_0000_0010	Netlist	LUT = A_1
Verilog	INIT = 16'b0000_0000_0000_0100	Netlist	LUT = A_2
Verilog	INIT = 16'b0000_0000_0001_0000	Netlist	LUT = A_3
Verilog	INIT = 16'b0000_0001_0000_0000	Netlist	LUT = A_4

# of bitstream files for proposed = 1

File	LUT value
Verilog	INIT = 16'b0000_0000_1000_1010

그림 1. 4-입력 LUT 의 매핑 규칙 분석

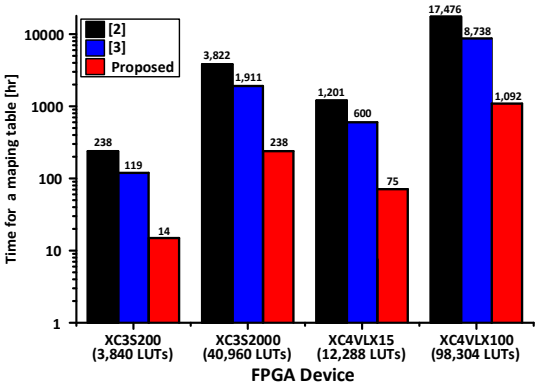

그림 2. FPGA 에 따른 매핑 테이블 생성 시간 비교

나는 위치는 항상 1 번째, 2 번째, 4 번째, 8 번째로 고정되고, 2^n 비트 단위로 매핑 규칙이 반복되는 것을 알 수 있다. 이때, 매핑 규칙의 2^n–1 (n = 1, 2, 3)번째 비트인 1 번째, 3 번째, 7 번째 비트의 위치는 입력 신호의 비트 패턴 [3]에 의해 매핑 규칙마다 다르게 나타나는 것을 확인하였다. 따라서 그림 1 에서 보이는 바와 같이 1 번째, 3 번째, 7 번째 비트는 모두 1 이고, 나머지 비트는 0 으로 고정한 16'h0000_0000_1000_1010 를 INIT 값으로 설정하여 생성한 비트스트림만 이용하여 4-입력 LUT 에 대한 매핑 규칙을 찾을 수 있다.

기존 연구에서 제안된 방법과 본 논문에서 제안하는 방법으로 LUT 매핑 테이블을 만들기 위해서는 나머지 요소를 모두 동일하게 설정한 상태에서 INIT 의 모든 비트가 0 으로 설정된 $\mathbf{B}_0$ 와 그림 1 에 나타난 비트스트림 파일들을 비교하여야 한다. $\mathbf{B}_0$ 와 비교하면 모든 비트스트림은 INIT 값만 다르므로 두 비트스트림에서 값이 다르게 나타난 비트의 위치를 추출하여 LUT 의 매핑 테이블을 생성할 수 있다.

III.실험 결과

기존의 연구와 [2, 3] 본 논문에서 제안하는 4-입력 LUT 매핑 테이블을 생성하는 방법을 이용했을 때 소요되는 시간과 매핑 테이블을 이용하여 LUT 에 구현된 회로를 복원하였을 때 그 정확도를 비교하였다. 실험을 위해 Xilinx 사의 FPGA 가운데 4-입력 LUT 로 FPGA 가 구성된 Spartan-3 의 XC3S200, XC3S2000 칩과 Virtex-4 의 XC4VLX15, XC4CLX100 칩을 사용하였으며, EDA 툴은 Xilinx 사의 ISE Design Suite 14.7 버전을 사용하였다.

그림 2 는 기존의 LUT 매핑 테이블 생성 방법과 제안하는 LUT 매핑 테이블 생성 방법을 사용할 때에 소요되는 시간을 4 종류의 FPGA 칩에 대해 비교한 결과이다. 제안하는 방법을 사용하면 [2]에서 제안된 방식 대비 소요시간이 89% 감소되며, [3]에서 제안된 방식 대비 78% 감소된다. 이때, 제안하는 방법으로 생성된 매핑 테이블은 기존의 방법 [2, 3]으로 생성된 매핑 테이블과 완벽히 동일하다.

V. 결론

본 논문에서는 매핑 규칙 분석을 통한 새로운 4-입력 LUT 매핑 테이블 생성 방법을 제안한다. 제안하는 4-입력 LUT 매핑 테이블 생성 방법을 통하여 기존의 4-입력 매핑 테이블 생성 방법 [2, 3]보다 생성에 소요되는 시간을 78% 이상 줄일 수 있다. 따라서 제안하는 방법으로 비트스트림에서 LUT 에 구현된 회로를 복원하는 시간을 단축시키면서 완벽한 회로를 복원할 수 있다.

참고문헌

[1] S. Drimer, "Volatile FPGA design security–a survey," Computer Laboratory, University of Cambridge, 2008.

[2] P. Swierczynski, M. Fyrbiak, P. Koppe and C. Paar, "FPGA Trojans Through Detecting and Weakening of Cryptographic Primitives," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 8, pp. 1236-1249, Aug. 2015.

[3] S. Choi, H. Yoo, "Fast Logic Extraction of LUT from Bitstream in Xilinx FPGA," MDPI Electronics, vol. 9, no. 7, pp. 1-11, July 2020.