Extraction of ROM Data from Bitstream in Xilinx FPGA

Soyeon Choi
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
soyeonchoi@cnu.ac.kr

Jieun Yeo
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
jeyeo.cas@gmail.com

Hoyoung Yoo
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
hyyoo@cnu.ac.kr

Abstract— **Recently, many researches have investigated efficient reverse engineering methods to restore Programmable Logic Points (PLPs) and Programmable Interconnect Points (PIPs) in SRAM-based Field Programmable Gate Arrays (FPGAs). However, the research on the restoration of Programmable Content Points (PCPs) such as memory data are rarely studied. In this paper, we propose an efficient reverse engineering method to recover Read Only Memory (ROM) data, which is essential for the implementation of modern digital circuits. First, we analyze the FPGA hardware resources mapped to Xilinx primitive library of ROM, and next the proposed reverse engineering process is explained using mapping relation between ROM data and hardware resources. As an example, XC3S50 FPGA of Xilinx Sparatan-3 family is utilized, and the process of restoring the SBOX of AES (Advanced Encryption Standard) is provided as a practical application.**

Reverse Engineerkng; ROM; Bitstream; FPGA; Xilinx

I. INTRODUCTION

Due to low development cost and reconfigurable property, FPGAs are widely used in various fields such as communication, consumer electronics, and defense applications [1]. Among various types of FPGAs including Flash-based and Fuse-based FPGAs, SRAM-based FPGAs are the most commonly used, and Xilinx and Intel have supported most of volumes in the FPGA market. In general, SRAM-based FPGA consists of Input/Output Block (IOB), Block RAM (BRAM), and Configurable Logic Block (CLB) as shown in Fig. 1(a). CLB is the most important block since it can reconfigure logical operation by selectively operating the internal logics of CLB. CLB is composed of several sub-blocks called SLICEs as shown in Fig. 1(b). Fig. 1(b) shows the structure of XC3S50 of Xilinx Spartan-3, which consists of four SLICEs, two SLICEL and two SLICEM. More precisely, SLICEL consists of pairs of 4-input LUT (LUT4), MUX, Carry Chain, and Flip-Flops (FFs) to provide basic logic circuit configuration, and SLICEM includes additional components associated with Shift Register Logics (SRL) and Read Access Memory (RAM) to support further complex circuit configuration. Note that we exemplify XC3S50 of Xilinx Sparatan-3 family thoroughly in this paper to provide a concise explanation.

Due to the fact that the SRAM memory is volatile, the internal circuit configuration in SRAM-based FPGAs disappears when the power is turned off [2]. To resolve this problem, a nonvolatile external memory is inevitably required to store the netlist of the RTL design outside the SRAM-based FPGA. The information embedded in the netlist is typically categorized into Programmable Logic Points (PLPs), Programmable Interconnect Points (PIPs), and Programmable Content Points (PCPs). The programmable points stored in external memory used to implement the target RTL design on the FPGA are transferred in the form of a bit-stream. Reverse engineering is the entire process to extract the bit-stream while transferring it from the external memory to the FPGA and to recover it by figuring out PLPs, PIPs, and PCPs [3].

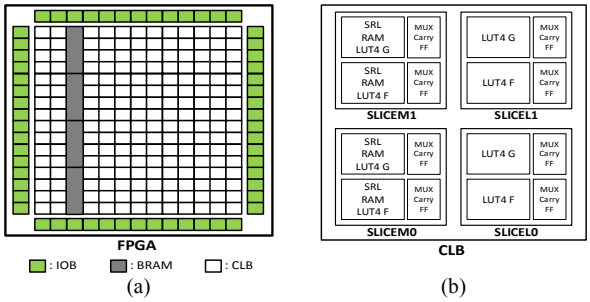

Figure 1. (a) Xilinx FPGA structure and (b) CLB structure for XC3S50.

Recently, many researches have investigated efficient reverse engineering for PLPs and PIPs of the SRAM-based FPGA, but the researches on the restoration of PCP such as memory data are rarely studied. In this paper, we propose an efficient reverse engineering process of ROM data that essential for the implementation of modern digital circuits. There are three main contributions of this paper; 1) Analysis on the mapping relationship between the ROM primitive and the internal hardware for Xilinx FPGA. 2) We propose the process of reverse engineering the ROM data of Xilinx FPGA using the mapping relation. 3) To bring a practical example, the proposed reverse engineering is exemplified using S-BOX in AES encryption algorithm.

II. PROPOSED METHOD

In general, FPGA manufactures such as Xilinx provides primitive libraries written in hardware description language (HDL) to facilitate RTL implementation on the FPGA [4]. For instance, the primitive library for ROM is provided in Xilinx as ROMi×1, where i indicates the size of ROM. The stored bit is accessed depending on the ROM access address. There are five types of primitive libraries [5] for ROM in Xilinx XC3S50;

TABLE I. UTILIZED FPGA HARDWARE RESOURCES FOR EACH PRIMITIVE LIBRARY OF ROM

Primitive	# CLBs	# LUTs	SLICE	LUT
ROM16×1	1	1	SLICEM0	LUT G
ROM32×1	1	2	SLICEM0	LUT G, LUT F
ROM64×1	1	4	SLICEM0, SLICEM1	LUT G, LUT F
ROM128×1	1	8	SLICEM0, SLICEM1, SLICEL0, SLICEL1	LUT G, LUT F
ROM256×1*	2	16	SLICEM0, SLICEM1, SLICEL0, SLICEL1	LUT G, LUT F

* ROM256×1 uses 2 vertically consecutive CLBs.

ROM16×1, ROM32×1, ROM64×1, ROM128×1, and ROM256×1.

To analyze the FPGA hardware resources used to implement each primitive library [4] of ROMs, we synthesized and implemented each ROM primitive, and then analyzed all the generated bit-stream as follows;

STEP 1) Bit-stream generation for each primitive
STEP 2) Search for ROM location
STEP 3) Extraction of ROM data for target bit-stream
STEP 4) Rearrangement of ROM data

First, an individual ROM primitive is synthesized twice with different initial values, all zeros and all ones. Except for the initial values, the other hardware resources maintain the same to distinguish the location of the target primitive ROM [4]. Second, the two bit-streams generated from Xilinx design suits are compared and the proposed method determines the ROM location where the ROM data appears in the bit-stream. Next, given the obscured bit-stream, the ROM data is extracted in the ROM location determined in STEP 2. Finally, the ROM data is rearranged using [5], since ROM data is not arranged in the natural order. The proposed method can restore any type of ROM that are implemented using Xilinx primitive library [4].

Through STEP1 and STEP4, it is newly found that all ROMs are internally assigned into the LUTs of SLICEs. Since Xilinx Spartan-3 has a 4-input LUT as a basic structure as shown in Fig. 1(b), one LUT4 in Xilinx Spartan-3 can store 16-bit ROM data. According to the iterative analysis, we found that the utilized LUTs for each ROM primitive is deterministic, and Table 1 summarizes the utilized FPGA hardware resources according to the ROM type. As shown in Table 1, ROM16×1utilizes a single LUTG in SLICEM0, and ROM32x1 utilizes two LUTG and LUTF in SLICEM0, and so forth. It is noticeable that LUT is configured internally by Xilinx design suite, and ROM primitive is instantiated by RTL designers.

To bring a practical example, we recovered S-BOX of Advanced Encryption Standard (AES) algorithm that is the most widely used encryption algorithm. In AES algorithm, constant values called S-BOX are used to ensure the confidentiality of messages, and S-BOX composed of 8-bit 256 data as shown in Fig. 2. Since S-BOX does not change on encryption and decryption, S-BOX is normally implemented as ROM. More precisely, 8 ROM256×1 is necessary for implementing S-BOX with 8-bit 256 data. The upper part in Fig. 2 shows the process of creating the bitstream by implementing the S-BOX of AES on the FPGA, and the lower part in Fig. 2 depicts the process of reverse engineering the S-BOX of AES. According to the experiments, when

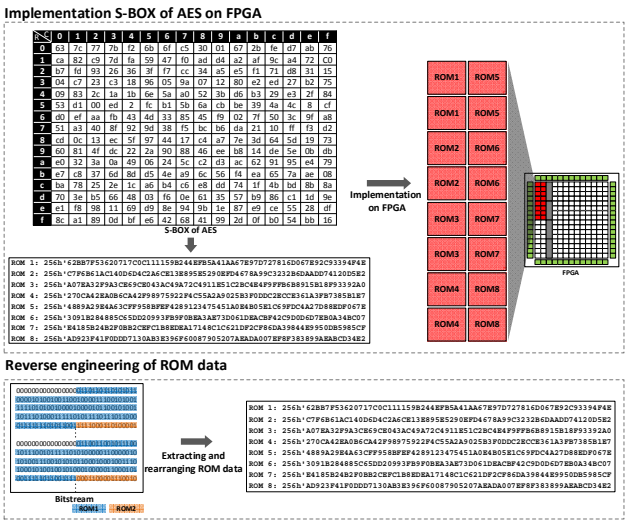

Figure 2. Example of S-BOX in AES algorithm.

implementing S-BOX of AES on the XC3S50 FPGA, a single ROM256×1 is utilized as two LUTs existed in vertically consecutive CLBs as described in Table 1, and total 8 ROM256x1 are utilized as 16 LUTs within 8 CLBs. As a result, the proposed reverse engineering can restore ROM data of S-BOX in AES completely given the unknown bit-stream.

III. CONCLUSION

This paper presents an efficient reverse engineering method target for ROM data. The mapping relation between Xilinx FPGA primitives and FPGA hardware resources are analyzed, and using the mapping relation, the proposed reverse engineering process is explained with Xilinx XC3S50 as an example. Due to the lack of page limit, the reverse engineering process associated with S-BOX in AES is briefly described. The proposed reverse engineering can be further applicable to any type of primitive library of ROM data.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019M3F3A1A01074449)

REFERENCES

[1] P. Lysaght, B. Blodget, J. Mason, J. Young and B. Bridgford, "Invited Paper: Enhanced Architectures, Design Methodologies and CAD Tools for Dynamic Reconfiguration of Xilinx FPGAs," 2006 International Conference on Field Programmable Logic and Applications, Madrid, 2006, pp. 1-6.

[2] O. Heron, T. Arnaout and H. -. Wunderlich, "On the reliability evaluation of SRAM-based FPGA designs," International Conference on Field Programmable Logic and Applications, 2005., Tampere, 2005, pp. 403-408.

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

[4] Xilinx. Spartan-3 Libraries Guide for HDL Designs. CA: Xilinx, 2013.

[5] 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.