CYENG 225 – SP 23: Microcontroller Essentials for Cyber Applications
Instructor: Dr. Shayan (Sean) Taheri
Gannon University (GU)

Assignment 5: Chapter 5 - Hardware Root of Trust

Total Points: 100**; and Deadline:** February/15/2023, 11:59 PM.

Note – Cheating and Plagiarism: Cheating and plagiarism are not permitted in any form and they cause certain penalties. The instructor reserves the right to fail culprits.

Deliverable: All of your responses to the questions of assignment should be included in a single compressed file to be uploaded to the Gannon University (GU) – Blackboard Learn environment.

Question 1. Provide short answers (i.e., no more than five lines on average with the font size of 12) for the following items. The grade for each item is **10 points**.

1. Specify the processes that require cryptographic keys to form the **Root of Trust (ROT)**. Explain each of those processes briefly.
2. Discuss how an executing software on a processor can be **Attested Remotely**. Provide a figure that reflects your explanations.
3. Explain how **Signatures** made by a trusted party can be employed to authenticate codes and data.

Question 2. Select and implement PUF defensive solutions and evaluate the results: A physical unclonable function (PUF) is a physical object that for a given input and conditions (i.e., computing challenge), provides a physically defined "digital fingerprint" output (i.e., computing response). The fingerprint serves as a unique identifier for security operations. Complete the following Steps. The grade for this question is **70 points**.

A. Implement **the following three PUF circuits** using **Verilog hardware description language (HDL)**. The implementation of the third circuit in VHDL is provided for your guidance (i.e., "**PUFCircuit3_VHDL.zip**"). General samples are given for your kind reference and understanding of the implementation processes.
B. Perform the procedure from the "**EXPERIMENT #1: Introduction to Xilinx's FPGA Vivado HLx Software**" laboratory assignment for your implementations.
C. Include the following items in your submitting package:
   - Provision of the achieved results from your implementations in the **Steps A and B**.
   - All files of your implementations in the **Steps A and B**.
   - Provide a report that includes: (**1**) your overall understanding and conclusions from completing the experiments; (**2**) the interesting points and the challenges that you faced in this laboratory; and (**3**) the screenshots for all of the major steps in your experiments.

Refer to the "**PUF Circuit 1**" paper for more information about the following circuit.

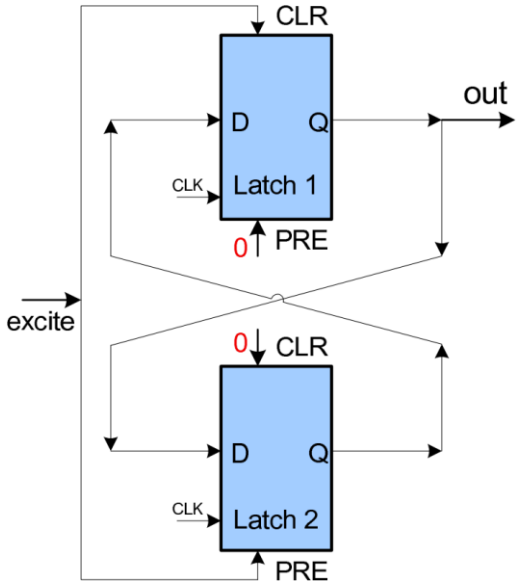

Figure 1: PUF Circuit 1.

Refer to the "**PUF Circuit 2**" paper for more information about the following circuit.

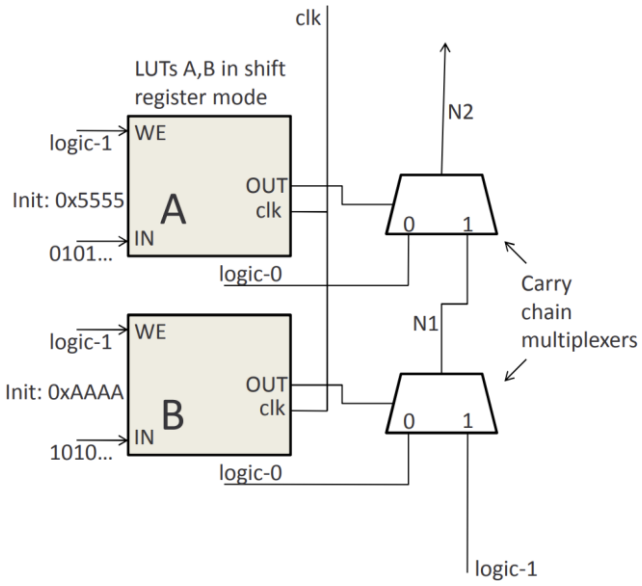

Figure 2: PUF Circuit 2.

Refer to the "**PUF Circuit 3**" paper for more information about the following circuit.

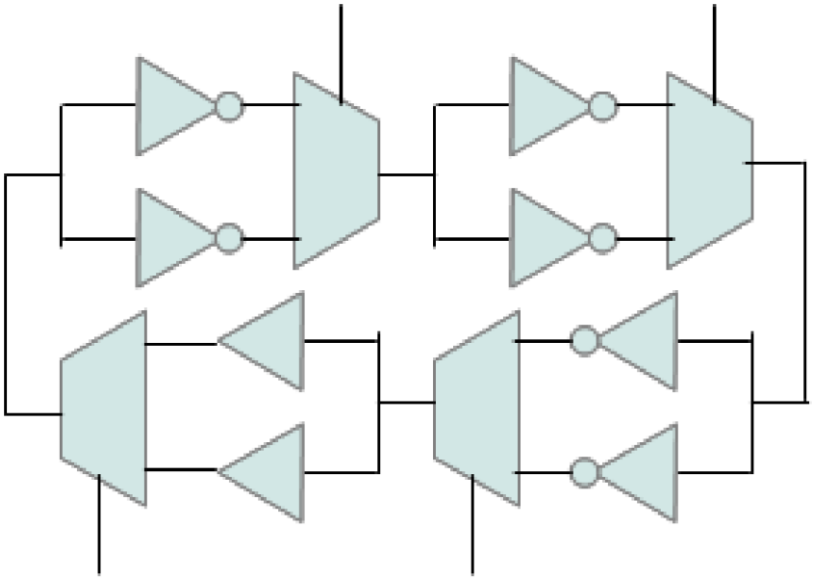

Figure 3: An 8-stage configurable ring oscillator (CRO).

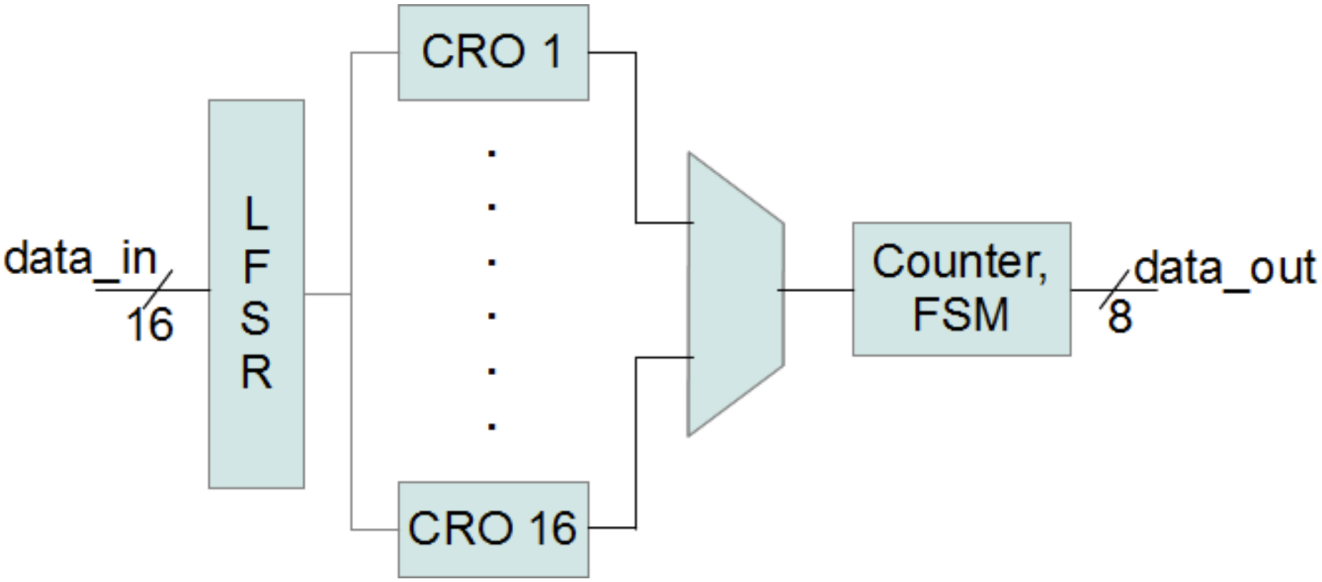

Figure 4: PUF Circuit 3.

Resources:

1. stnolting/fpga_puf: Technology-agnostic Physical Unclonable Function (PUF) hardware module for any FPGA. (github.com)
2. Crimsonninja/senior_design_puf: Repository to store all design and testbench files for Senior Design (github.com)
3. rahuls321/Hardware-Security: Related papers (github.com)
4. FAU-LS12-RC/CHOICE-PUF: CHOICE, is a novel class of FPGA-based PUF designs with tunable uniqueness and reliability characteristics. By the use of addressable shift registers available on an FPGA, CHOICE provides a wide configuration space to obtain a device-specific PUF response without sacrificing randomness. (github.com)
5. xiangyun-wang/Implementation-of-Asymmetric-Delay-based-PUF-on-FPGA (github.com)
6. CarlosCraveiro/PUFTRandGen: PUF based True Random Number Generators written in Verilog (github.com)
7. simoasnaghi/FPGA_PUF: This is a simple implementation of an Arbiter PUF made for an FPGA Artix-7. Project part of the course Embedded Systems, part of the Electronics Engineering Master course at Politecnico di Milano. (github.com)
8. andreaaspesidev/puf-fpga (github.com)
9. renaturation/DNN_PUF_FPGA (github.com)
10. Justin5567/Ring-Oscillator-PUF: Implement Ring Oscillator PUF on Xilinx FPGA (github.com)
11. zona8815/Arbiter-PUF: Use Verilog to design Arbiter PUF, then do Floorplan to make inter-variation approach 50%. (github.com)
12. sebanti10/APUF: The VHDL design for Arbiter PUF (github.com)
13. AlfonsoDiMartino/Reed-Muller-Correction_Code_on_PUF_Response: VHDL implementation of Reed-Muller Coder and Decoder for SSD exam. (github.com)
14. eriksargent/PUF-lab: FPGA implementation of a physical unclonable function for authentication (github.com)
15. scluconn/LPN-based_PUF: FPGA implementation of a cryptographically secure physical unclonable function based on learning parity with noise problem. (github.com)
16. IFM-Ulm/ro-pr-fw: Framework based on Partial Reconfiguration for chip characterization utilizing ring-oscillator PUFs (github.com)
17. canaknesil/4x4-apuf: An FPGA Implementation of Arbiter PUF with 4x4 Switch Blocks (github.com)
18. RuochenDai78/EEE5716_Intro_to_hardware_securityRing-Oscillator-PUF-design-and-optimization-for-FPGA: In this paper, a series of Ring Oscillators has been wired in different ways to form Ring Oscillator PUFs to find out the RO PUF which has a better quality. Thus, three different routing methods are proposed and developed , software automatic routing design, 1D and 2D hard-macro array design. And the quality factors including uniqueness, reliability and uniformity are analyzed (github.com)
19. mdkul22/ucsd-ece268-puf: Creating a MAC based PUF for FPGAs (github.com)
20. kamat900/puf-1: Hardware PUF FPGA implementation (github.com)
21. oliver132/FPGA-PUF: FPGA VHDL implementation of a Physical Unclonable Function (github.com)
22. RHESGroup/FPGA-based-PUF: Physical unclonable function for SEcube (github.com)
23. LinkZyy/fpga_puf_axi (github.com)
24. pkpio/dual_core_PUF_with_PDL_and_ethernet: Implementation of a dual core adder based PUF on FPGA. To be submitted for the Design Automation Conference' 14 (github.com)
25. rxg161230/Physically-Unclonable-Functions-for-Hardware-Security: PUF is a digital Fingerprint used to prevent semi-conductor device designs of a particular company to be stolen, copied or

remade by the Foundry or any other company. This project was to analyze which out of Arbiter or Butterfly PUFs work the best for Hardware Security. (github.com)
26. eknoes/puf-lab (github.com)
27. chethan2807/puf_rng_64_bit: This project implements a 64-bit Ring oscillator based PUF on Arty A7 FPGA (github.com)
28. scluconn/DA_PUF_Library: Defense/Attack PUF Library (DA PUF Library) (github.com)
29. salaheddinhetalani/PUF: Design and evaluation of an arbiter-delay-based Physically Unclonable Function (Secure Hardware Design Assignment) (github.com)