Cryptography on Hardware Platforms (WS 2023/24)

Assignment 2

Soft deadline for Task 2: 12th January
The final deadline for all tasks: 19th January

In this exercise, you will implement a physical True Random Number Generator (TRNG) on the Xilinx PYNQ-Z2 platform.

The total points for Assignment 2 are 40, including 10 points for the individual oral defense.

Review class notes on random number generation and the video on the practical tutorial before starting this assignment.

Open the Vivado project associated with this assignment and create a Vitis workspace (using the provided Vitis code for the assignment) to make all hardware and software implementations.

Task 1: Implementation of a TRNG (Total 15 points)

In this task, you will implement a Ring Oscillator (RO)-based TRNG in the FPGA that achieves a high entropy level during statistical testing. The TRNG should use timing jitter as the entropy source, as taught in the lectures.

In the Vivado project, open the TRNG_RO( ) module. You will find an unoptimized (and low-quality) reference implementation inside it. You may use it as a starting point for implementing the high-entropy TRNG.

Your TRNG description must be present within the module "TRNG_RO( )". Do not change the interface of the module.

```
(* keep_hierarchy = "true" *)
module TRNG_RO(en, rng_reg, rng_ready);
input en;
output reg [63:0] rng_reg;
output rng_ready;

// Your TRNG description.

endmodule
```

Brief and high-level description of the execution flow: In the Vivado project, the TRNG_RO( ) module is present inside the TRNG_wrapper( ) module. The provided TRNG_wrapper( ) does necessary jobs related to communications with the Cryptoprocessor. When the SW part (code in Vitis workspace) wants to obtain random data from the FPGA, it sends Instruction code 18 to the Cryptoprocessor. After that, TRNG_wrapper( ) generates

512*64 random bits (i.e., 512 64-bit words) using TRNG_RO( ). The random words are stored in the BRAM so the SW part can read them in its buffer array. The project has already implemented the necessary steps for interfacing TRNG_RO( ). In this task, you should focus on implementing the TRNG_RO( ) module.

On the processing system (i.e., the ARM processor in Vitis), perform entropy evaluation of the generated random data under the non-iid assumption. **Perform necessary modifications to the TRNG hardware so that the TRNG-generated data has a min-entropy of at least 0.80.**

The C++ codes for the entropy testing are already in the Vitis_code folder. The random data size is set to 1,000,000 bits for a meaningful entropy evaluation.

Task 2: On-chip statistical testing (Total 15 points)

Soft deadline: 12th January. Upload a report on the implementation method for Task 2.

You will implement the Markov estimate in FPGA to measure the min-entropy of your TRNG.

The Markov estimation method is described with an example on page 51 of NIST Special Publication 800-90B. Additionally, see the explanation provided in the Assignment-2 tutorial video.

Your Markov estimation implementation in HW must be present within the module "statistical_test( )". Do not change the interface of the module.

```
module statistical_test(clk, stat_trng_rst,
                        enable_TRO, random_reg, rng_ready,
                        stat_error, done,
                        debug_out);
```

In the Vivado project, you have been provided with a reference implementation of the frequency test that checks if there is any significant difference between the number of 1s and 0s in a string of 1M bits. You should replace this reference implementation (which is frequency test) with your Markov estimation implementation.

When the Vitis software wants to perform a Markov test in the FPGA, it sends Instruction code 19 to the Cryptoprocessor. During the statistical test, the above module calls TRNG_RO( ) for obtaining random bits. The test requires 1M random bits. At the completion of the test, statistical_test( ) sends done=1 signal to the Cryptoprocessor. If the min-entropy using Markov estimate is found to be lower than 0.8, the module sets stat_error=1. Otherwise, when the entropy is found to be higher than 0.8, stat_error=0.

Optimization hints: The description of the generic Markov estimate is not at all implementation-friendly. You should develop a much simpler description of the Markov estimate assuming that the number of bits is fixed to 1M and the required classification is binary (i.e., min-entropy is above or below 0.8). Also, note that in the FPGA there is no memory to store 1 million bits. You need to simplify the estimation to implement it in a small area/memory. A small approximation error (<0.1) from the accurate Markov estimate in the

entropy estimation is acceptable. **You can simulate this module as this module performs deterministic operations.**

Submission guidelines

- The final deadline for submitting your project (all tasks) is 19th January
- The soft deadline for submitting Task 2 is 12th January
- Upload the Vivado project and Vitis workspace to the git repository of your team by the deadlines.
- Upload your short 1 or 2 page report to the git repository of your team by the final deadline. The report should have:
    - Summary and explanation of your design choices strategy.
    - Optimizations techniques that you used.
    - Results of on-chip statical testing.
    - Implementation results (throughput bits/sec and area count).
        - Number of LUTs, FFs, DSPs, BRAMs and Area metric.

Marking scheme

10 points are reserved for the individual oral defense of Assignment 2.

Task 1 (15 points)

- You get 12 points in Task 1 if you have a working TRNG implementation satisfying the requirements on min entropy

- You get 0 to 3 points based on how well your implementation of Task 1 performs in terms of resource requirements, speed, and code quality.

Task 2 (15 points)

- You get 12 points in Task 2 if you have a working implementation of the Markov estimation in hardware.

- You get 0 to 3 points based on how well your implementation of Task 2 performs in terms of resource requirements, speed, and any additional features such as innovations.

You will be working in teams of two. We expect each of you to contribute equally to the assignment. Individual defense of the assignment will take place a week after the submission deadline. The defense has 10 points. You will be asked questions related to the project of Assignment 2.