A Self-Calibrating True Random Number Generator

Adriaan Peetermans, Miloš Grujić, Vladimir Rožić and Ingrid Verbauwhede
imec-COSIC, KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
Email: {adriaan.peetermans, milos.grujic, vladimir.rozic, ingrid.verbauwhede}@esat.kuleuven.be

***Abstract*—True Random Number Generators (TRNGs) are essential in all security systems. Unfortunately, large design effort is required to ensure that a TRNG design on a Field-Programmable Gate Array (FPGA) generates a sufficient entropy density at its output. This design effort relates to the fact that for each FPGA family a manual placement and routing procedure has to be executed. On top of this often comes the additional effort of finding a suitable location inside the target FPGA. This searching procedure has to be repeated for every device separately.**

In this demo, we show the working of a novel entropy source for the Coherent Sampling Ring Oscillator (COSO) based TRNG. This entropy source eliminates the need for any manual intervention during the implementation process. It generates two oscillating signals that can be matched with a precision of a few picoseconds. A controller regulates this entropy source based on some predefined bounds on the period length difference of the two oscillating signals.

I. Demo setup

The demo setup together with the Graphical User Interface (GUI) and a block diagram of the COSO-TRNG [1] are shown in Fig. 1. The TRNG uses a configurable architecture as proposed in [2]. The GUI enables the user to:

- Modify the calibration parameters used by the controller.
- Monitor live TRNG output data and random byte histogram.
- Assess the generated entropy by monitoring basic statistical test results.
- Monitor the throughput of the TRNG.

II. Demo working

The GUI is connected with a *Xilinx Spartan 6* FPGA that contains the TRNG. The controller inside the FPGA drives the reconfigurable Ring Oscillators (ROs) to produce a counter value within the specified range. The least significant bit of this counter value is used as the random bit. The ROs are reconfigurable thanks to a technique often used in Application Specific Integrated Circuit (ASIC) designs (e.g. a TERO-TRNG [3]). For each RO stage, multiple candidate stages are implemented and a control signal enables a sequence of stages that produce the required oscillating frequency.

The controller remains active during operation to dynamically adjust the RO configuration in case of a change in operating conditions. The controller notifies the end user when no configuration can be found that matches the specified counter range. By changing the parameter bounds, a user can explore the trade-off between the throughput and the statistical quality of the produced random bits.

Acknowledgements

This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work is supported in part by the Hercules Foundation AKUL/11/19, and by the European Commission through the Horizon 2020 research and innovation programme Cathedral ERC Advanced Grant 695305. Adriaan Peetermans is funded by an FWO fellowship and Vladimir Rožić is an FWO postdoctoral researcher.

References

[1] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for fpgas," in *Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field-Programmable Gate Arrays*. ACM, 2004, pp. 71–78.

[2] A. Peetermans, V. Rožić, and I. Verbauwhede, "A highly-portable true random number generator based on coherent sampling," in *Proceedings of the 2019 International Conference on Field-Programmable Logic and Applications (FPL)*, September 2019.

[3] K. Yang, D. Blaauw, and D. Sylvester, "An all-digital edge racing true random number generator robust against pvt variations," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1022–1031, 2016.

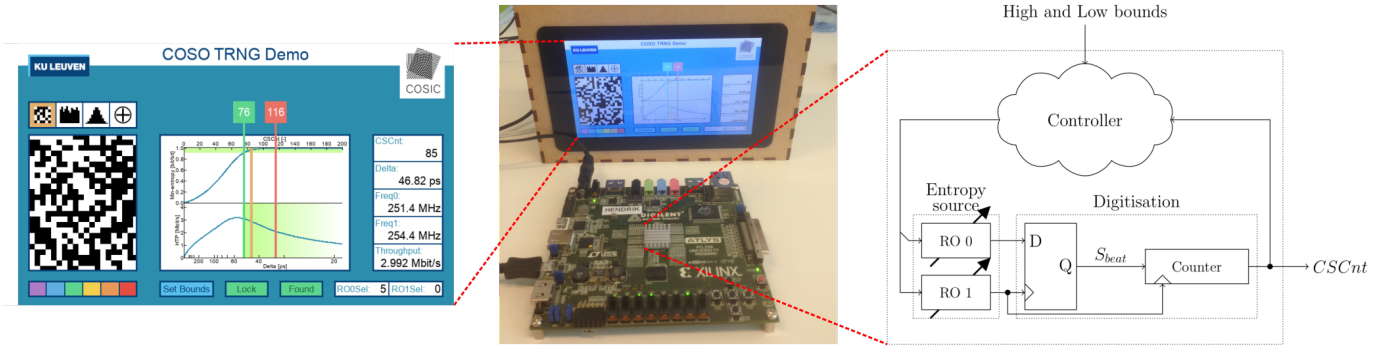

Fig. 1. Demo setup.