An open-source multi-FPGA modular system for fair benchmarking of true random number generators

Viktor Fischer, Patrick Haddad, Florent Bernard

AN OPEN-SOURCE MULTI-FPGA MODULAR SYSTEM FOR FAIR BENCHMARKING OF TRUE RANDOM NUMBER GENERATORS

Viktor Fischer, Florent Bernard

Hubert Curien Laboratory, UMR 5516 CNRS,
Jean Monnet University Saint-Etienne
18, rue Pr. Lauras, 42000 Saint-Etienne, France
email: (fischer, florent.bernard)@univ-st-etienne.fr

Patrick Haddad

STMicroelectronics
Advanced System Technology
Av. Coq, 13790 Rousset , France
email: patrick.haddad@st.com

ABSTRACT

True Random Number Generators (TRNG) are cryptographic primitives that exploit intrinsic noise sources in electronic devices. Their quality is linked to the underlying technology, activity of the neighboring circuitry and device environment (temperature, power supply, electromagnetic emanations). Consequently, when comparing TRNGs, they should be tested in identical technology, system architecture and operating conditions. We present a unified hardware platform and related open source tools aimed at fair benchmarking of TRNGs implemented in different FPGA technologies. The platform is accessible remotely. Designers can download related tools from the web site and they can upload their configuration bitstream to the remote FPGA and download random data generated in the same hardware and in the same conditions as other concurrent designs and state-of-the-art generators. The proposed tools were approved in many applications and they guarantee safe acquisition of random sequences at data rates of up to 400 Mbits/s.

1. INTRODUCTION AND MOTIVATIONS

Random number generation is a critical issue in numerous cryptographic applications: generation of initialization vectors, challenges, nonces and confidential keys. Random number generators are classified into two main categories: Deterministic Random Number Generators (DRNG) and True Random Number Generators (TRNG). While the DRNG is based on an algorithmic process, the TRNG exploits noisy analog phenomena in electronic devices to produce random bit sequences.

FPGAs are widely used for implementation and evaluation of cryptographic primitives, algorithms and protocols [1]. Many new TRNGs were recently implemented in FPGAs as well [2]. However, their quality is mostly evaluated using evaluation boards designed by FPGA vendors with different objectives. These boards are unsuitable for TRNG evaluation because they include many unnecessary components (and thus undesired additional noise sources) and switching power supplies. When using these boards, the quality of TRNG is very often evaluated by designers using common batteries of statistical tests, without taking into account underlying technology, external noise sources and operating conditions. One of rare examples is the approach presented in a recent work [3], in which the authors studied the influence of the package temperature and the FPGA core voltage on a TRNG implemented in two different devices. The authors highlighted that the bias of the raw binary sequence depended on the device family, temperature and FPGA core voltage. However, they did not discuss in their paper the impact of external noise sources on the generator.

The security in random number generation is related to the unpredictability of the generated bit sequences. In order to guarantee the unpredictability, the entropy per bit at the generator output should be as close to one as possible. However, some generators that seem to be unpredictable can be manipulated, while giving more or less predictable results [4]. This can have catastrophic consequences on the security of the cryptographic system relying on confidentiality of encryption keys generated in the generator. In order to be robust against manipulations, environmental fluctuations and aging, the generator output should depend only on intrinsic noise sources (e.g. thermal noise). We can imagine the following scenario: the attacker could replace the standard noisy power supply of the card used during TRNG evaluation by a low noise battery power supply and thus reduce the entropy per bit and to make TRNG output predictable. For this reason, when evaluating the generator principle, all noise sources, which do not come from the generator should be reduced to a minimum. This very important condition is neglected very often in scientific papers presenting and evaluating new TRNG principles [2].

From the above mentioned analysis it follows that in order to compare TRNG principles and their implementation in different FPGA families as fairly as possible, the evaluation boards should be identical, they should contain only necessary components and should operate in the same con-

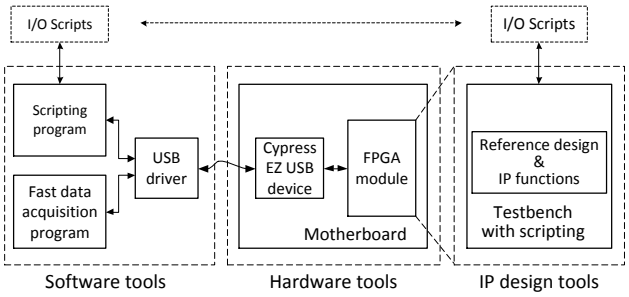

Fig. 1. The Evariste II toolkit structure

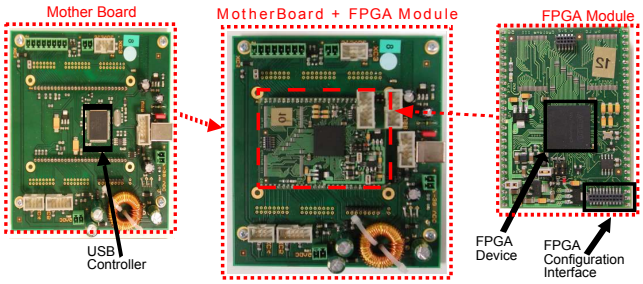

Fig. 2. Hardware part of the Evariste II system

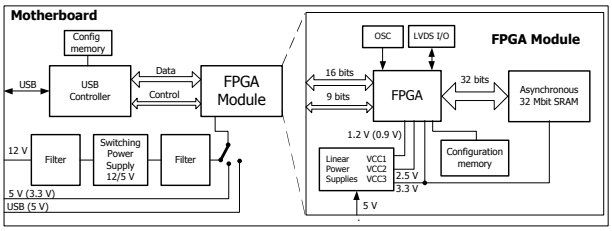

Fig. 3. The Evariste II hardware configuration

ditions. This is for example clearly not the case in recent papers [5], [6], [7], [8], [9], in which five different "noisy" evaluation boards were used. This way, the claimed performance of TRNGs evaluated in the cited papers was specific to individual boards and operating conditions.

In this paper, we propose an open-source multi-FPGA modular system named Evariste II, which is composed of a set of hardware, software and design tools. The goal of this system is to offer designers a common platform, accessible via Internet, for implementing, testing and comparing TRNG designs, while maintaining the comparisons of principles and their behavior in different technologies as fair as possible. All the necessary documents and tools including source codes can be accessed at the Evariste II web page [1].

The paper is organized as follows: Section 2 presents the main parts of the Evariste II toolkit. Section 3 describes current and future use of the system. Section 4 concludes the paper and describes future steps concerning Evariste II evolution.

2. THE EVARISTE II TOOLKIT

The system Evariste II is an open source toolkit (see Fig. 1) aimed at evaluation of TRNG in reconfigurable hardware. It consists of a set of hardware, software and design tools making the development and fair evaluation of TRNG easier. The Evariste II system has two main objectives:

- To provide unified and easily accessible TRNG design and evaluation environment, guaranteeing identical operating conditions for tested TRNG principles.
- To reduce environmental perturbations (e.g. external noise sources such as switching power supplies) that can modify the operation of the TRNG under test.

2.1. The hardware

The dedicated hardware constitutes the core of the Evariste II system. It consists of a motherboard and five types of FPGA modules having identical architecture and topology (see Fig. 2). All FPGA modules are built around an FPGA device from one of five selected FPGA families: Altera Cyclone III, Altera Arria II, Xilinx Spartan 3, Xilinx Virtex 5 and Actel (Microsemi) Fusion. New modules containing new families are under development. The proposed hardware architecture has two main advantages:

- By implementing the same TRNG structure in five different FPGA families, but in topologically identical modules, the impact of the FPGA technology on TRNG performance can be precisely studied.
- By implementing different TRNG principles in the same FPGA devices and modules, performance of different TRNGs, but in identical conditions can be fairly compared.

The block diagram of the hardware configuration is depicted in Fig. 3.

Motherboard

The motherboard features the Cypress EZ USB interface controller CY 7C68013A – 100 ACX with its configuration memory. It can be powered from a 12 V battery, 5 V external power supply or the USB bus and it delivers the power to the FPGA module.

FPGA modules

Five hardware modules having the same topology are available. They contain: one FPGA device (Altera Cyclone III EP3C25F256-C8, Altera Arria II GX EP2AGX45CU17C6,

[1] http://labh-curien.univ-st-etienne.fr/wiki-evariste-ii

Xilinx Spartan 3 XC3S700AN-FFG484, Xilinx Virtex 5 XC5-VLX30T-FFG323 or Actel Fusion M7AFS600 FGG256X2 FPGA), one configuration memory (except for the Fusion FPGA module), one configuration interface, one 16 MHz quartz oscillator and two connectors for the Low Voltage Differential Signaling (LVDS) data interface . Moreover, in order to reduce external noise sources that can modify the quality of generated bitstreams, only linear power supplies and high quality low pass ferrite filters are used for powering FPGAs.

The choice of FPGA families was motivated by the objective of proposing a set of modules representing different vendors, technologies and application categories. Indeed, according to their vendors, Altera Cyclone III and Xilinx Spartan 3 devices are intended for cost-sensitive applications, Actel Fusion for extended temperature military type applications, Xilinx Virtex 5 for high-performance general logic applications and Altera Arria II for transceiver-based and fast embedded applications.

For proper TRNG evaluation, data acquisition from the TRNG under test must be perfectly controlled. The speed of the transfer must be high enough (at least as high as the generator output bit rate) and no data must be lost or repeated. The data interface and validation of its design is thus very important. Two solutions are possible in the Evariste II system: 1) for low speed generators (up to 48 Mbits/s), the available USB interface can be used; 2) for faster generators (transfers of up to 400 Mbits/s were tested), the generator output must be first saved in the RAM block and then transfered to the PC. For both solutions, we propose reliable designs independent from generator clock and output bit rate.

2.2. Typical TRNG design

The structure of a typical top level design structure is depicted in Fig. 4. The top level entity is composed of:

- Three basic blocks (*cypress_if*, *sequencer* and *pll1*) contained in a USB controller interface block – these blocks should remain unchanged.
- TRNG controller block (*applic_ctrl*) – this block should be adapted by the user to the given TRNG structure.
- TRNG core and its wrapper (*applic_wrp*) – this block should be entirely redesigned by the user.

Naturally, more PLLs can be used for the TRNG implementation if required.

The size of basic modules in different FPGA devices is presented in Tab. 1. We can conclude that the surface is negligible comparing to the size of all FPGA devices used in the system. The space remaining in all families is largely sufficient for implementing common TRNG designs. Note that given values could be slightly different for concrete applications, depending on complexity of the TRNG design.

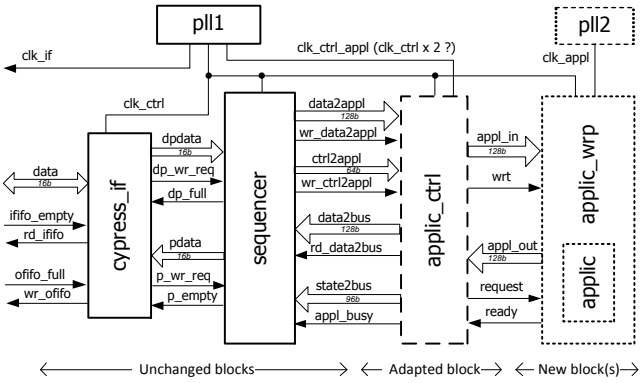

Fig. 4. Typical top level VHDL entity structure

	cypress_if		sequencer		applic_ctrl		applic_wrp	
Device	LC	Reg	LC	Reg	LC	Reg	LC	Reg
CycloneIII	40	31	495	644	23	15	143	137
Arria II	37	31	146	644	19	15	141	137
Spartan 3	46	31	479	644	15	15	270	137
Virtex 5	43	31	412	644	12	15	269	137
Fusion	24	31	828	644	26	15	780	137

Table 1. Typical area for basic modules implemented in FPGA in Logic Cells and registers, (LC = Logic Cells, i.e. LEs for Cyclone III, ALUTs for Arria II, LUTs for Spartan 3, LUTs for Virtex 5 or tiles for Fusion FPGA)

One could argue that the sequencer and surrounding logic could have non negligible impact on the generator. For this reason, we placed two LVDS inputs/outputs on each module. If needed, the generator implemented in one FPGA module can have only two outputs – a random data output and a strobe signal validating the random bits. Both signals can be transmitted to the destination FPGA module containing embedded RAM block and USB interface. The practical speeds of the random data transfers across LVDS interface that were verified very thoroughly attained up to 400 Mbits/s.

2.3. Scripting commands and tools

The proposed scripting tools are optimized for easy to use and flexible control of developed hardware functions in simulation testbenches and hardware tests. The scripts serve for sending 64-bit control words in control packets and 128-bit data blocks in data packets and for receiving 96-bit state words and 128-bit data blocks. Two scripting tools are available:

- An interpreter included in the VHDL testbench file, which instantiates and stimulates the Design Under Test (DUT) in simulations,
- A software interpreter *script.exe* (a console application) running on the host PC and accessing the corresponding hardware via USB bus.

Consequently, the same scripts can be used in VHDL simulations and in hardware tests. The concept of the two scripting environments in conjunction with the fast acquisition software application guarantees high design productivity, flexibility and fast and reliable data interface. For more details about the communication between the system and the host computer using packets, a wiki page is available on the web site.

2.4. Fast acquisition software

Last but not least, the Evariste II system contains a fast data acquisition software that is needed for reliable acquisition of huge random sequences. Using this software, the user can create the same packet structures as those that were tested and approved in scripts. The software can create automatically as many files with parameterized size as needed, while reporting their creation time and bit rate.

3. CURRENT AND FUTURE USE OF THE SYSTEM

We suppose the next design strategy, while using the system. The designer will first download the Evariste II design tools. He will design the TRNG core and its wrapper according to the application controller interface. Next, he will add new control commands (if needed) to the application controller. Finally, he will write scripts and simulate the design. Once the simulation results are as expected, he will launch the *script.exe* program interpreting the same scripts. If the response of the hardware is the same as that of the VHDL model, the high-speed real-time acquisition program can be executed for acquiring sufficient amount of data.

All the software and IP function source codes and design tools can be downloaded from the web site after simple user identification. Next, the user can access the hardware either in the context of a scientific cooperation, purchase it from the Micronic company or it can access the hardware remotely via Internet.

Clearly, the remote access has some advantages: 1) it extends the open-source principle to the hardware availability; 2) it guarantees the same operating conditions for all tested and compared generators.

The server application and the hardware is accessible via above mentioned web page.

4. CONCLUSION AND PERSPECTIVES

We presented a unified hardware platform and related open source tools aimed at fair benchmarking of TRNGs implemented in selected FPGA technologies. The platform is accessible remotely. Designers can download related tools and upload their configuration bitstream to the remote FPGA and download random sequences generated in the same hardware and in the same conditions as other concurrent designs and state-of-the-art generators. The proposed tools were approved in many applications and they guarantee safe acquisition of random bitstreams at data rates of up to 400 Mbits/s.

The next version of the system will include communications between the host PC and the FPGA modules using optical fibers. The generator hardware will be placed in a Faraday cage in order to be perfectly isolated from external environment. A professional Faraday cage was already purchased and will be soon installed. This will create an unprecedented opportunity for correct and fair TRNG testing.

5. REFERENCES

[1] V. Fischer and F. Bernard, *Security Trends for Fpga's: From Secured to Secure Reconfigurable Systems*. Springer, 2011, ch. 5. True Random Number Generators in FPGAs, pp. 101–135.

[2] V. Fischer, "A closer look at security in TRNGs design," in *Proceedings of Constructive Side-Channel Analysis and Secure Design – COSADE'12*, ser. LNCS, vol. 7275. Springer-Verlag Berlin Heidelberg, 2012, pp. 167–182.

[3] C. Hochberger, C. Li, M. Raitza, and M. Vogt, "Influence of operating conditions on ring oscillator-based entropy sources in FPGAs," in *Field Programmable Logic and Applications, FPL'12*, 2012, pp. 555–558.

[4] P. Bayon, L. Bossuet, A. Aubert, V. Fischer, F. Poucheret, B. Robisson, and P. Maurine, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *Proceedings of Constructive Side-Channel Analysis and Secure Design – COSADE'12)*, ser. LNCS, vol. 7275. Springer-Verlag Berlin Heidelberg, 2012, pp. 151–166.

[5] M. Varchola and M. Drutarovsky, "New high entropy element for fpga based true random number generators," in *Cryptographic Hardware and Embedded Systems, CHES 2010*. Springer, 2010, pp. 351–365.

[6] K. Wold and C. Tan, "Analysis and enhancement of random number generator in fpga based on oscillator rings," in *Proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig'08)*, 2008, pp. 385–390.

[7] 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.

[8] S. Yoo, B. Sunar, D. Karakoyunlu, and B. Birand, "A robust and practical random number generator," 2007.

[9] M. Thamrin, I. Ahmad, and M. Hani, "A true random number generator for crypto embedded systems," in *Regional Postgraduate Conference on Engineering and Science*. School of Postgraduate Studies, UTM, 2006, pp. 253–256.