

Platform for Testing and Evaluation of PUF and TRNG Implementations in FPGAs

Marek Laban^{*†}, Milos Drutarovsky^{*}, Viktor Fischer[‡], and Michal Varchola[†]

^{*}Department of Electronics and Multimedia Communications

Technical University of Kosice Park Komenskeho 13, 04120 Kosice, Slovak Republic

[†]MICRONIC, Sliacska 2/C, 83102, Bratislava, Slovak Republic

[‡]Univ. Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien
UMR 5516, F-42023, Saint-Etienne, France

Email: laban@micronic.sk, milos.drutarovsky@tuke.sk, fischer@univ-st-etienne.fr, varchola@micronic.sk

Abstract—Implementation of cryptographic primitives like Physical Unclonable Functions (PUFs) and True Random Number Generators (TRNGs) depends significantly on the underlying hardware. Common evaluation boards offered by FPGA vendors are not suitable for a fair benchmarking, since they have different vendor dependent configuration and contain noisy switching power supplies. The proposed hardware platform is primarily aimed at testing and evaluation of cryptographic primitives across different FPGA and ASIC families. The modular platform consists of a motherboard and exchangeable daughter board modules. These are designed to be as simple as possible to allow cheap and independent evaluation of cryptographic blocks and namely PUFs. The motherboard is based on the Microsemi SmartFusion 2 SoC FPGA. It features a low-noise power supply, which simplifies evaluation of vulnerability to the side channel attacks. It provides also means of communication between the PC and the daughter module. Available software tools can be easily customized, for example to collect data from the random number generator located in the daughter module and to read it via USB interface. The daughter module can be plugged into the motherboard or connected using an HDMI cable to be placed inside a Faraday cage or a temperature control chamber. The whole platform was designed and optimized to fulfill the European HECTOR project (H2020) requirements.

I. INTRODUCTION

Nowadays, we live in an information society where information appears mostly in a digital form. An electronic mail is used more often than a traditional mail, documents are stored in a digital form more than on a paper and information is often very expensive. Cryptography has become increasingly important to ensure data security.

In the framework of the information security politics of the European Union, a project called HECTOR [1] was recently accepted for funding. The project emerged from the scientific cooperation of TRUDEVICE partners: KU Leuven and Jean Monnet University Saint-Etienne. HECTOR should bridge basic algorithmic approaches with hardware-level security implementations. It requires to evaluate in a fair way many hardware dependent cryptographic primitives, in many different technologies. A flexible platform for testing and evaluation of primitives implemented on various Field Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC) devices was therefore needed.

II. CRYPTOGRAPHY AND RANDOMNESS

Cryptography applies mathematical methods to ensure information security requirements such as data confidentiality, integrity, and authentication, but also authentication of devices and subjects [2]. It uses cryptographic primitives to build cryptographic protocols. Cryptographic primitives like random number generators (RNGs) and physical unclonable functions (PUFs) extract randomness from the underlying hardware. Although other cryptographic primitives like symmetric- or asymmetric-key ciphers, and one way functions are deterministic, their implementation in hardware can leak confidential information and it is therefore hardware dependent, too.

A. Random Number Generators

Random numbers play an important role in information security applications. They are used as confidential keys, initialization vectors, nonces in authentication protocols, and masks in side-channel attack countermeasures.

RNGs generate sequences of random numbers (bits or vectors of bits), which have to fulfill two fundamental requirements: statistical quality and unpredictability. The unpredictability of generated numbers can be based on a computational hardness or on some exploitation of a random physical process. In the first group of generators called the deterministic RNGs, it must be computationally difficult to guess the following or preceding output values of the generator from the current output value. In the second group of RNGs called true RNGs (TRNGs) the unpredictability comes from an unpredictable and non manipulable random physical process like thermal noise. Although better from the point of view of unpredictability, TRNGs are usually slower and have lower statistical quality than their deterministic competitor.

B. Physically Unclonable Functions

Physical unclonable functions (PUFs) [5] are pieces of hardware exploiting randomness in the device manufacturing process. They are based on physical characteristics which are reliable and unique for each chip, difficult to predict, and easy to evaluate. PUF cannot be replicated, even if the full design is known [6]. PUFs can be used to authenticate hardware or to generate hardware dependent confidential keys.

C. Hardware Dependence of TRNGs and PUFs

Both TRNGs and PUFs depend significantly on the underlying hardware. TRNGs use dynamic random process like electric noises to generate random numbers. PUFs use random phenomena appearing during the manufacturing process of logic devices. Both TRNGs and PUFs are influenced by surrounding hardware, which must be designed very carefully.

D. HECTOR Project

HECTOR (Hardware Enabled Crypto and Randomness) is a European cooperative research project. The main objective of this project is to close the gap between basic algorithmic approaches and hardware-level security implementations [1]. The main goal of the project is to study, design and implement RNGs and PUFs with demonstrable entropy guarantees and quality metrics. This includes on-the-fly entropy estimation and evaluation of robustness against physical attacks, which is needed in the security evaluation and certification process.

III. PLATFORMS DEDICATED TO EVALUATION OF CRYPTOGRAPHIC PRIMITIVES

The hardware dedicated to evaluation of cryptographic primitives must fulfill special requirements, especially from the point of view of electric noises, electro-magnetic interference and robustness of the design. Several solutions are currently available.

The Research Center for Information Security (RCIS) of AIST and Tohoku University developed the Side-channel Attack Standard Evaluation BOard (SASEBO) [7] as a research project funded by METI (Ministry of Economy, Trade and Industry, Japan). Several SASEBO boards aimed at evaluation of cryptographic functions implemented in FPGAs, ASICs, and Smartcards are available. The boards were designed essentially as platforms for evaluation of SCA attacks. They contain mostly two FPGAs: one as a target of evaluation (ToE), and the second one controls the target. Unfortunately, the ToE cannot be separated from the control FPGA and cannot be placed remotely in a hostile environment (a temperature controlled chamber or a chamber with a strong electro-magnetic field). The second disadvantage of the SASEBO boards in the context of the HECTOR project is, that only a limited choice of FPGA devices is available. This argument is valid also for SAKURA boards, which are successors of the SASEBO boards. Last but not least, the SASEBO (and SAKURA) boards are complex and thus expensive and consequently, not suitable for evaluation of PUFs, where a large amount of devices must be tested.

Another platform, a Flexible Open-source BOard for Side-channel analysis (FOBOS) [8] was designed by the George Mason University for conducting side-channel attacks on FPGAs. The platform consists of two different boards, one is used as a control board and another one as a device under test (DUT). Both cards are connected together by a module, which is called a bridge connector. The advantage of this platform is that it uses commercially available boards – it is therefore cheaper. But the use of commercially available

general purpose evaluation boards is also the main disadvantage of this solution: these boards were not intended for SCA evaluation purposes, they contain many redundant components and switching power supplies generating significant electronic noise.

Evariste III [9], is a platform aimed at development and evaluation of cryptographic functions and primitives in reconfigurable hardware. The platform was developed by the Jean Monnet University in cooperation with the MICRONIC company. It is a modular platform containing daughter boards (featuring target FPGAs or ASICs) and a motherboard containing USB data interface device. Three daughter boards designed for Evariste III and several other daughter boards designed for older platform Evariste II are available. The Evariste III modules contain connectors for SCA measurements. The main disadvantage of the Evariste III (and the Evariste II) system is that the modules cannot be used remotely, and that they are relatively expensive since they contain power supplies. Last but not least, the high speed data memory, which is needed for high-speed data acquisition, is available only on few modules (which are thus more expensive).

IV. HECTOR HARDWARE TOOLS

The main motivation for designing the HECTOR evaluation platform was to design hardware, which would be optimized for a thorough, but still easy evaluation of cryptographic primitives implemented in FPGA and ASIC devices. The designed modular platform consists of a *motherboard* and four types of interchangeable *daughter boards*. Evaluated cryptographic primitives are implemented in daughter boards with hardware resources significantly reduced and data are stored, processed and transmitted to the PC using the motherboard featuring large choice of peripherals and interfaces. The daughter boards can be connected to the motherboard remotely and can be thus placed in a hostile environment during attacks.

A. Daughter Board

The HECTOR daughter boards are designed to allow evaluation of primitives across different FPGA families and ASICs. The selected architecture has two main advantages. First, the daughter modules contain only the necessary hardware components, which minimizes their impact on the behavior of the target primitive. Second, the module is simple and thus cheaper, i.e. a huge number of modules can be manufactured to test PUFs.

Three types of daughter modules featuring Altera Cyclone V, Xilinx Spartan-6, and Microsemi SmartFusion 2 FPGA were designed. The fourth type of the daughter module will feature an ASIC, which will be designed in the framework of the project.

Selected devices represent recent FPGA families of main FPGA vendors. The daughter modules contain small programming connector and an additional connector with input/output pins, quartz oscillator, micro-miniature coaxial (MMCX) connectors available for side-channel attacks (SCA) implementation (triggering and measurement), and a SATA connector. The

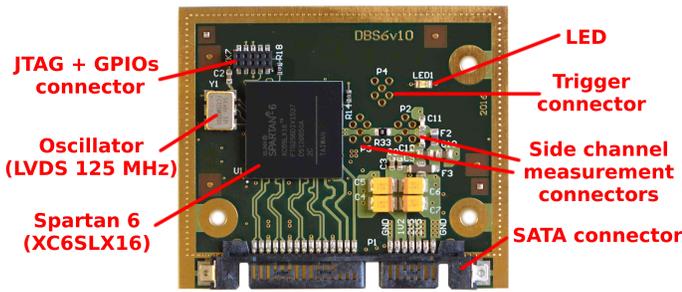


Fig. 1: Spartan-6 HECTOR daughter board layout

SATA connector is used to power the board and to transfer data between the daughter board and the motherboard (see Fig. 1). The SATA connector is used mainly for its good signal integrity and mechanical features. The SATA interface protocol is not supported by the hardware. Instead, four LVDS (low voltage differential signaling) signal couples, three single ended wires and power supply voltages are present on the connector. The daughter boards contain high quality power filters. To reduce the cost and the electric noise, all power regulators are placed on the motherboard.

B. Motherboard

The main task of the motherboard is to control daughter modules, to read and eventually to process data from the modules and to ensure data transfers to the PC. The board uses USB interface to communicate with the PC and a variety of connectors for plugging in different daughter modules. (see Fig. 2).

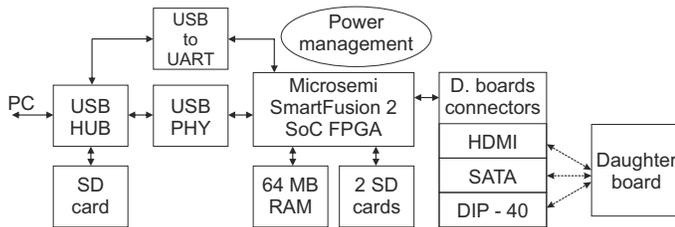


Fig. 2: Motherboard hardware block diagram

The motherboard is based on the Microsemi SmartFusion 2 M2S025 FPGA. The SmartFusion 2 (SF2) is a system on chip (SoC) FPGA device, which integrates a flash-based FPGA fabric and a 166 MHz ARM Cortex-M3 processor. The time-critical parts of the system can be processed by the fabric and the communication protocol can be implemented in the Microcontroller Sub-System (MSS). The SF2 can be programmed through the JTAG interface by FlashPro programmer (both FPGA and MCU). The programming interface can be also used for debugging the firmware using SoftConsole or other tools like IAR or KEIL.

The SmartFusion 2 M2S025 device features:

- 27696 logic elements (4 LUT + DFF),
- 34 math blocks, 6 PLLs, MSS running at 166 MHz,
- MSS 256 kB eNVM and 64 kB eSRAM,

- 267 total user I/Os.

The HECTOR motherboard features synchronous low-power 512 Mb (64MB) DDR SDRAM memory. It runs at 166 MHz, for a total theoretical bandwidth over 5.3 Gbps. It is provided as a flexible volatile memory for user applications.

The communication between the SF2 and the PC is ensured via USB by two data channels. The first one, the virtual COM port, is designed to exchange control packets between the motherboard and the PC using a simple UART protocol. The communication is controlled by the FTDI device FT232RL. The UART protocol is supported by many operating systems, therefore no special driver is necessary.

The second channel is designed to provide reliable high-speed data transfers using the USB mass storage class interface, which is natively supported by operation systems, too. It is ensured by the USB physical layer circuit (USB3300), which creates an intermediate interface between the SF2 and the USB differential wires.

Both USB ports are connected to the USB HUB (USB2640), which ensures reliable data transfers to the PC. An additional Micro SD card reader with a mass storage class interface is connected to the USB HUB, too. The SC card can contain for example some installation software.

Two additional micro SD card slots placed on the motherboard can be used to save and to transfer data to other devices.

The motherboard is powered by an external 5 V power supply. Only linear regulators are used on the board due to their low noise compared to noisy switching regulators. This low noise feature is very important for fair TRNG and PUF evaluation, but also for evaluation of robustness against side channel attacks.

In order to reduce the cost of daughter boards, the boards are powered from dedicated voltage regulators, which are placed on the motherboard. Three of them are user-configurable by micro switches. The configurable regulators were used to ensure compatibility across various daughter boards, which usually require different power supplies (e.g. the power voltage of the FPGA core may be different). The whole set of power supplies is properly filtered to avoid any interference or noise.

The motherboard provides three connectors for connecting additional board (HECTOR daughter boards or Evariste II modules):

- High-definition multimedia interface (HDMI) connector,
- Serial - ATA (SATA) connector,
- Zero insertion force (ZIF) connector.

The daughter boards can be plugged directly into the motherboard using the SATA connector or remotely using an HDMI cable.

The SATA connector ensures easy, reliable, low cost and low noise connection. An optional aluminium lid can be used to protect the daughter board from surrounding electro-magnetic fields, if necessary.

Connection of the daughter board via the HDMI cable can be useful when the tested device should be placed in a temperature controlled chamber or a Faraday cage. To make

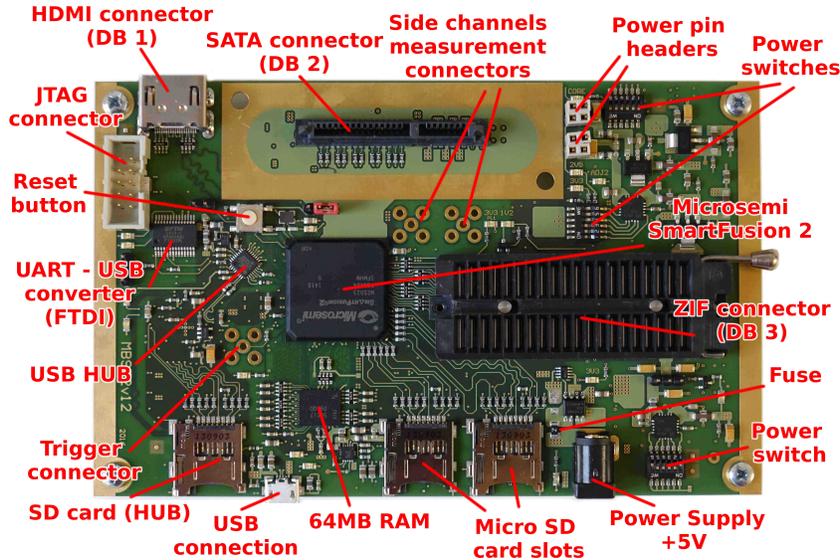


Fig. 3: Layout of the HECTOR motherboard featuring the SmartFusion 2 FPGA device

the connection easier, the same data signals are present on both SATA and HDMI connectors. However, in the case of a remote use of daughter boards, the power must be provided from an external power source connected to the available HDMI-to-SATA adapter module.

The zero insertion force (ZIF) connector, which is available on the motherboard is dedicated to additional boards, such as:

- Evariste II modules [9],
- Expansion boards (e.g. with switches and LEDs).

V. CONFIGURATION OF THE SYSTEM AND REFERENCE DESIGNS

A set of tools is provided in several reference designs: the user applications running on the PC, and the motherboard hardware and firmware adapted to various user applications placed in daughter boards. The main components of the system are depicted in Fig. 4.

Four reference designs are currently available:

- Simple control of the daughter board inputs and outputs,
- Simple data processing module placed in the daughter board and accessed from the PC via motherboard,
- High-speed data acquisition from a daughter board,
- Simple TRNG.

The proposed software tools and configuration of the system vary depending on the application. For the sake of place, we will briefly present only the fourth reference design in Section VI-C.

VI. SOFTWARE TOOLS

Two groups of software tools are available: the PC application software and the motherboard firmware.

A. PC Application Software

The main task of the PC application software is to provide the user API to the motherboard via USB interface. It uses a virtual COM port (VCP) to transfer commands and the state word and a USB mass storage interface to transfer data. The proposed communication protocol ensures reliable control of the motherboard by exchanging command and state packets. Created data files can be accessed directly from the PC.

To ensure flexibility and system independence, the software running on the PC is developed in a TCL language. Only a TCL interpreter is needed to read user scripts. The TCL interpreter is usually installed during the FPGA design software installation (e.g. Quartus or Libero). The script can be very easily edited and adapted to user requirements.

B. Motherboard Firmware

The motherboard firmware runs on the microcontroller subsystem (MSS) inside the FPGA device and it reads/writes to internal MSS peripherals, user peripherals placed in the FPGA fabric and external memory placed on the motherboard. The firmware is written in the C programming language.

The MSS part has the following roles:

- Reception of commands from the PC,
- Control of the device under test by sending commands to the FPGA fabric,
- Maintenance of the file system for data transfers (files can have up to 32 MB).

As mentioned earlier, the motherboard uses two communication channels: the UART for commands, status and small data (32-bit data blocks); and the USB mass-storage device to transfer large files (up to 30 MB in size).

The main role of the *Packetizer-Depacketizer* (PD) block is to maintain a communication protocol based on transferring

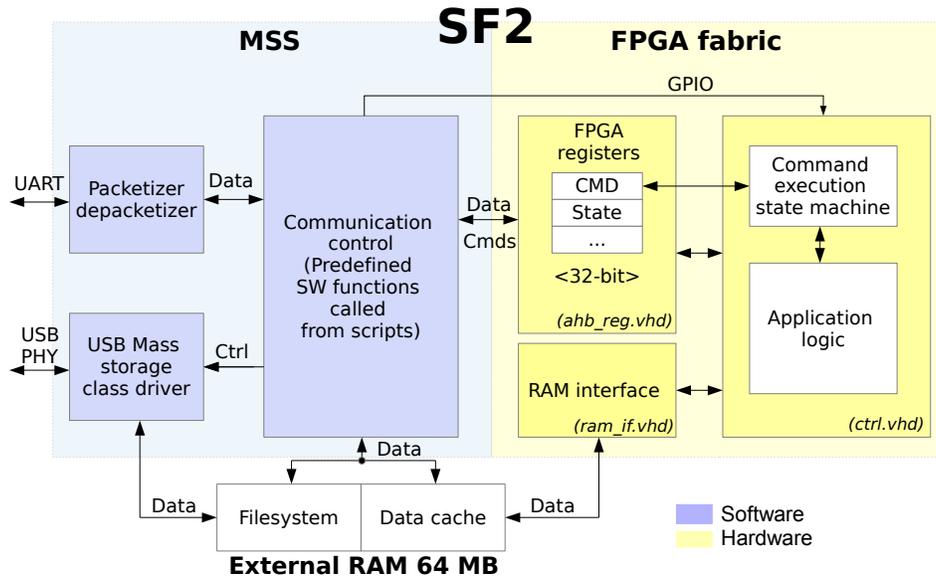


Fig. 4: Internal system block diagram

small 12-byte packets via UART. In this communication, the PC is a master device, which initializes every communication. The communication is based on the command-response pairs where each command generates a response. There are few types of commands – each having its own header and predefined contents (see Fig. 5). PD recognizes a header of the packet, collects appropriate data and generates response packets.

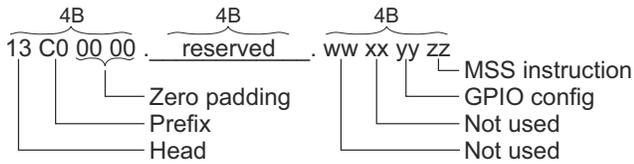


Fig. 5: Example of the command packet

The *USB mass storage class disk drive* is based on the reference design delivered by Microsemi. The disk is located in the RAM of the motherboard. The memory space is divided to two halves. The first half is used for the file system and the second half is used as a cache reserved for data acquisitions. The file system is based on the open-source library called FatFs. It is a generic FAT/exFAT file system module aimed at small embedded systems implementations.

A *Communication Control* block contains predefined functions, which manage operations in the FPGA and response to received packets. This block provides a simple way of how to insert new instructions. Predefined functions (instructions) are called by packets in following steps:

- Wait for the new instruction (idle state),
- Get new instruction from the PD,
- Execute instruction and send response to the PD.

The MSS features many channels to communicate with FPGA. There are 8+2 *GPIOs* which can be directly connected to the output of the device or can be used as a reset or common signals for the FPGA fabric instances.

An AHB lite bus is used to transfer small data and commands between the MSS and FPGA fabric parts. The control block is based on the AHB lite protocol specification. It enables MSS to write or read to/from *FPGA registers* without the need of any special routines.

The *RAM interface* implemented in FPGA can write up to 32 MB data blocks to the external RAM on the bit rate up to 400 Mbps. It uses the AHB bus to directly write data to the external RAM (DMA access).

The *command execution state machine* is implemented in the FPGA fabric. It is a simple state machine, which executes received commands. The command is transferred to the 32-bit command register block situated in the FPGA fabric.

C. TRNG Reference Design

We use a *Simple TRNG reference desing* with a PLL TRNG [4] implemented in the daughter board (see Fig. 6) as an example. The application requires fast acquisition system with big storage capacity. Generated data need to be stored in an external RAM to be sent to the host PC via USB.

The TRNG implemented in the daughter board uses just one data signal and one strobe signal. The output of the TRNG (before or after the decimator) can be selected using the GPIO. The reset signal is also connected. The daughter board can be connected to the SATA or HDMI slot. The connector is selected from the TCL script using a multiplexer placed in the FPGA fabric.

The serial output of the generator is converted to 32-bit words in a *S/P converter*. This converter also solves the synchronization problem between two independent domains

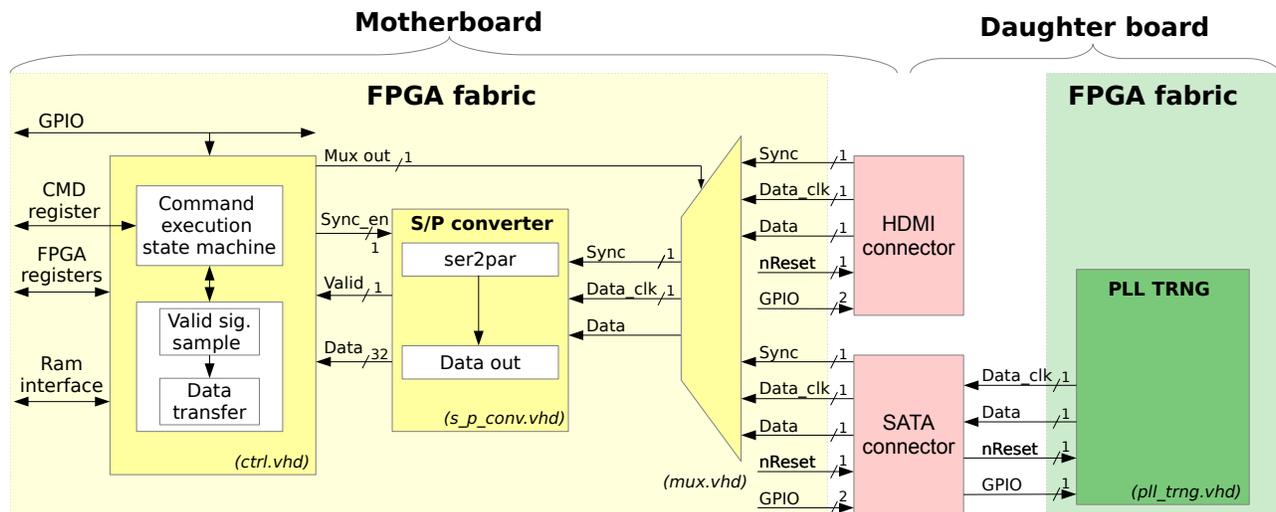


Fig. 6: Example of TRNG acquisition system implementation

(the TRNG clock domain and the acquisition system clock domain).

The control block just samples the output of the converter and saves it to the external RAM. At the end of the data acquisition, data are copied from the acquisition half of the memory to the file system in the second half of the memory.

The acquisition system is controlled by the host PC executing the following steps:

- 1) Send the Control packet to create the file system,
- 2) Send the Fabric packet to choose the connector (SATA/HDMI),
- 3) Send the Read file packet to run operation and set the size and the name of the required file,
- 4) Repeatedly send the Control packet to get the state of the operation,
- 5) Send the Control packet to mount the HECTOR disk,
- 6) Copy the file from the mass storage device (HECTOR disk).

VII. CONCLUSION

The HECTOR evaluation platform was designed following the project requirements and it reflects the needs of all HECTOR partners. It is a unique and powerful tool set particularly suitable for testing and evaluation of cryptographic primitives. However, the platform is sufficiently flexible to be adapted to a variety of other applications.

Comparing to existing platforms aimed at evaluation of cryptographic primitives and side-channel attacks, the HECTOR evaluation platform is unique in the following crucial points:

- Because of the simplicity of the daughter modules, a huge number of boards can be manufactured and tested. This is particularly useful for PUF evaluation.
- The number of components on the daughter board is limited to a strict minimum to minimize undesirable effects on the target of evaluation.

- Thanks to the simple serial interface existing between the motherboard and daughter boards, the target device can be placed remotely in a hostile environment, in order to perform active attacks,

All the IP functions and software tools from the reference designs are open-source. According to the HECTOR consortium agreement, the HECTOR evaluation boards can be used by third parties for an educational dissemination purposes.

ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 644052. This work was also supported by the Slovak Research and Development Agency under the contract No. APVV-15-0692.

REFERENCES

- [1] HECTOR project web page, available: <https://hector-project.eu/>
- [2] Menezes, Oorschot, Vanstone, *Handbook of Applied Cryptography*, CRC Press, October 1996.
- [3] Laban, M., *Development tools for evaluation of cryptographic primitives implemented in reconfigurable hardware*, Master thesis, Technical University of Kosice, Kosice, May 2016, pp. 1-91.
- [4] V. Fischer and M. Drutarovsky, *True Random Number Generator Embedded in Reconfigurable Hardware*, in *Cryptographic Hardware and Embedded Systems*, 4th International Workshop CHES 2002, pp. 415-430, Springer-Verlag, Aug. 1315, 2002.
- [5] R. Maes, P. Tuyls, I. Verbauwhede, *Intrinsic PUFs from Flip-flops on Reconfigurable Devices*, Katholieke Universiteit Leuven: ESAT-COSIC, WISSec 2008, available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.459.2788&rep=rep1&type=pdf>
- [6] Aaron Mills, Iowa State University, *Design and evaluation of a delay-based FPGA physically unclonable function*, available online: <http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=3416&context=etd>
- [7] National institute of AIST, *Side-Channel Attack Standard Evaluation Board SASEBO*, available online: <http://satoh.cs.ucc.ac.jp/SASEBO/en/>
- [8] Veleglati, Kaps *Towards a Flexible, Opensource BOard for Side-channel analysis (FOBOS)*, available online: <https://cryptography.gmu.edu/fobos/>
- [9] Laboratoire Hubert Curien, *Evariste wiki page*, available online: http://labh-curien.univ-st-etienne.fr/wiki-evariste/index.php/Main_Page