Table des matières

1. Introduction

This page presents the architecture of a Physical Unclonable Function (PUF) based on Transient Effect Ring Oscillators (TERO) and its implementation with FPGAs. The TERO PUF is based on the comparison between the numbers of oscillations of two elementary TERO cells. All implementation files are available at the end of this page. The TERO-PUF has been encapsulated inside the Evarist III System. This modular hardware system is presented in details on: https://labh-curien.univ-st-etienne.fr/wiki-evariste/index.php/Main_Page

2. The Evarist System

2.1 Description general of the system

The Evarist III system is a generic platform which is possible to implement a complex system without take care of the communication interface with the computer. Indeed, this system provides all the communication part between the external environment (computer) and the application implemented.

Nevertheless, a protocol needs to be defined in order to adapt the communication interface with the application. Figure 1 shows the overall system.

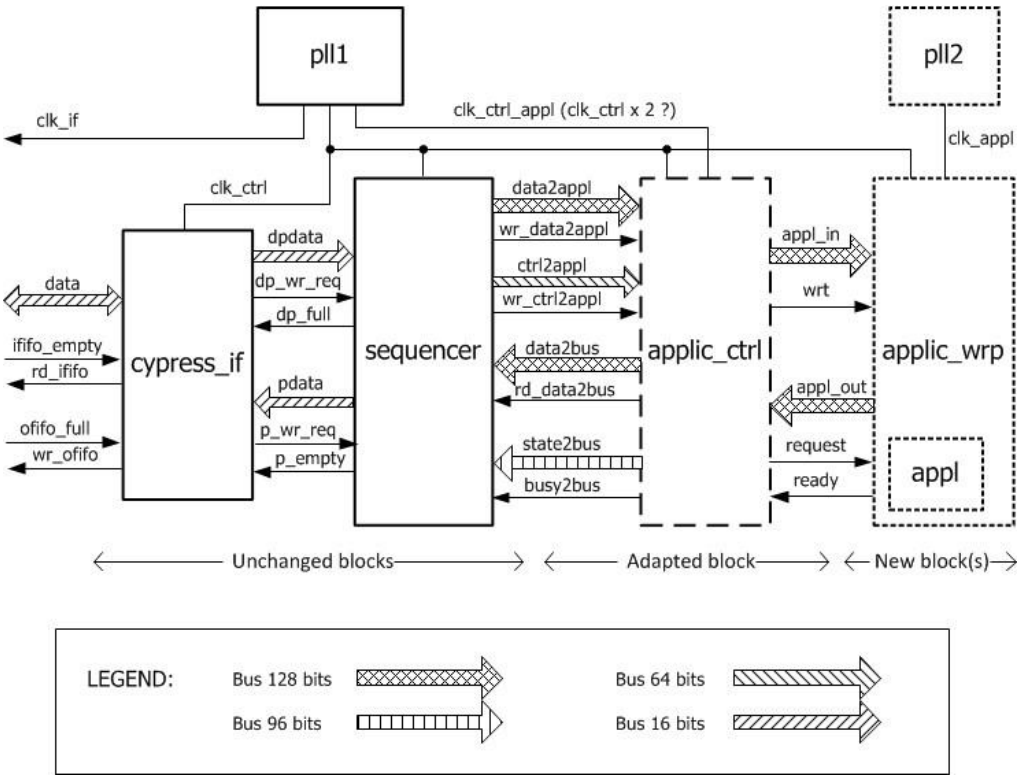

Fig. 1 Global view of the Evarist system extracted from the Evarist website

The parts of the system which the user can modify are only the application controller (*applic_ctrl*), the application wrapper (*applic_wrp*) and the generation of the clock for the application part (*pll2*). All other parts represent the interface between the external environment

(computer) and the application. The next subsection presents the implementation of the TERO-PUF inside the Evarist system without detail of the basic TERO cell.

3. The TERO-PUF implementation

3.2 The application wrapper (applic_wrp.vhd)

The application wrapper contains the core of the TERO-PUF implementation which is composed of:

- Two blocks of 128 TERO cells,
- Two 16-bit counters to count the number of oscillations of TERO cells,
- Two selectors to choose which are the two TERO cells started together,
- Two multiplexors in order to choose the right signal as clock for the counters.

The Figure 2 shows the schematic of the application wrapper.

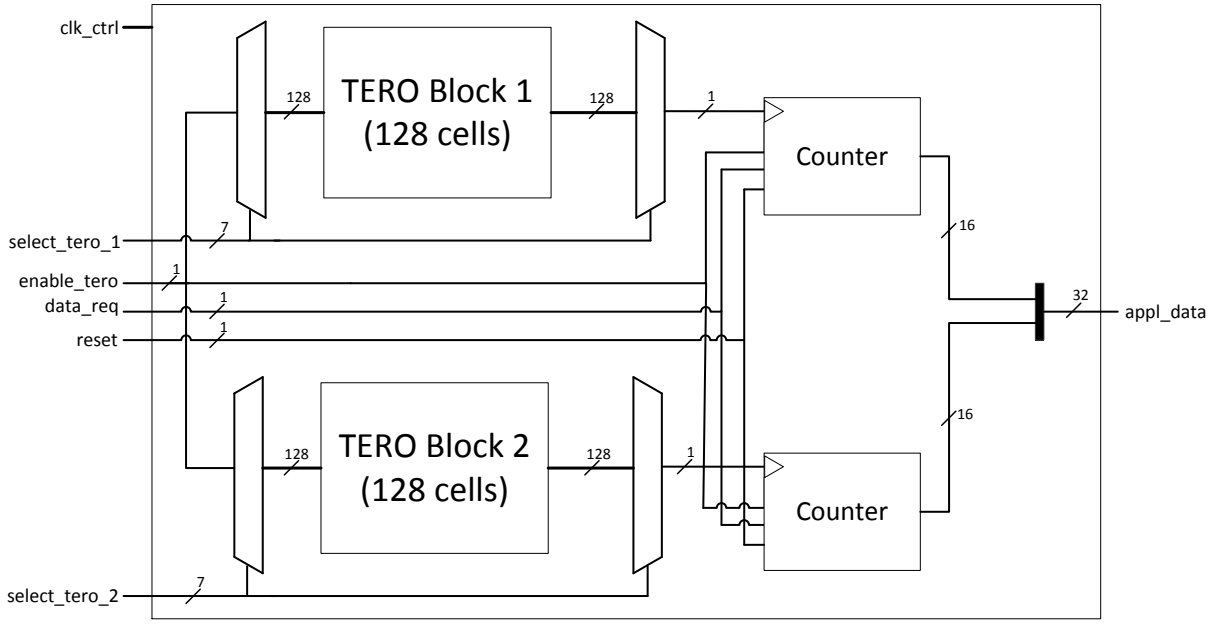

Fig. 2 Application wrapper schematic

This Figure shows clearly that the two blocks of TERO cells are completely separated inside the FPGA. In addition, by selecting only one TERO cell in each block, only two cells are able to oscillate at the same time.

3.2.1 The application controller (applic_ctrl.vhd)

This part of the system is a finite state machine which controls the execution of the TERO-PUF and its interaction with the rest of the system. It is the application controller which contains the communication protocol defined in order to send commands to the TERO-PUF such as configuration (time of the acquisition and challenge), acquisition of data and reset of the application.

The Figure 3 shows the finite state machine which controls the application wrapper. In addition, the interactions with the application are showed. Most of state transitions are controlled by the user who sends commands to the system (*MODE_CONFIG*, *MODE_START* and *MODE_IDLE*). Transition between START and READ are controlled by the system itself but the user can change the parameter CNT_MAX by sending a new acquisition time during a configuration step.

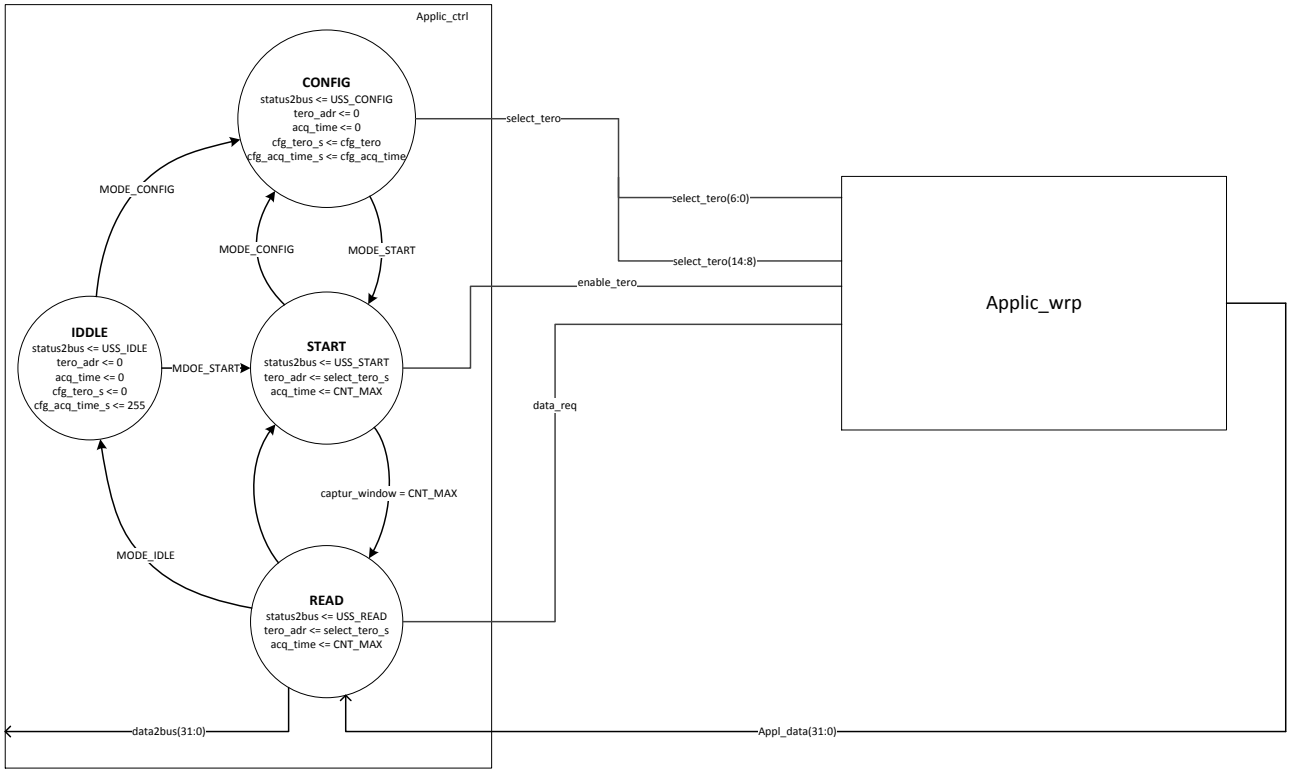

Fig. 3 Application control FSM

3.3 Communication protocol

The communication protocol is described in the Figure 4. The communication protocol allows to choose one TERO cell in each block separately and to change the acquisition time. The command word must contain all configuration parameters written in hexadecimal.

Ctrl2appl(63:0) : X000000XXXXXXXXX

35:20 -> cfg_acq_time : XXXXXXXXXXXXXXX
19:4 -> cfg_tero : 0XXXXXXX0XXXXXXX
10:4 -> select_tero_1
18:12 -> select_tero_2
2:0 -> mode2appl : 00XX
0 – IDLE
3 – Config
7 - Data acquisition

Example :

C 000000000500F093

- mode : config
- tero : 9 of block 1 and 15 of block 2
- acquisition window : 80 clock cycles

Fig. 4 Communication protocol

4. Example of script

The whole system is controllable by script. The Figure 5 shows an example of documented script. In addition, the communication protocol is also described in this Figure.

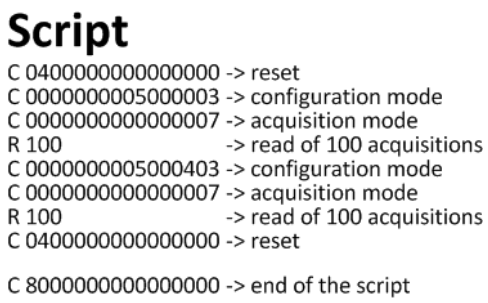

Fig. 5 Example of script

The example of script shows the correct protocol to perform acquisition using the communication protocol:

1. Start by a reset
2. Configure the acquisition
3. Ask for a number of acquisition
4. End the communication

It is possible to perform several acquisitions without doing a reset each time but by reconfigure the acquisition and start the acquisition (step 2 and 3). The number following the **R** command must be less than 128 but it is possible to chain this command without reconfiguration to read more than 128 times the number of oscillation of the selected TERO cells:

R 100
R 100
R 100 …

5. List of files

5.1 Files for Xilinx

The following list presents the list of files used to implement the TERO-PUF inside Xilinx Spartan 6 FPGAs

5.1.1 Non modified files (Evarist system)

- cypress_if.vhd
- elapsed.vhd
- sequencer.vhd

5.1.2 Modified files (Application and package)

- transfer2usb.vhd: Top level file of the project
- applic_ctrl.vhd: File which control the correct execution of the application
- applic_wrp.vhd: File containing the application

- gen_clk_wrp.vhd: File which generate the clock for the application
- lab_hc_pkg.vhd: File containing all types for the project
- Tero_cell_S6_SLX4_7inv.nmc: Hard macro of the Tero cell

5.1.3 Generated file

- transfer2usb.ucf: Placement file, modified to place all of the 256 TERO cells

5.1.4 Other File

- TERO_inst.vhd: VHDL implementation of the TERO cell

5.2 Files for Altera Cyclone V

The following list presents the list of files used to implement the TERO-PUF inside Altera Cyclone V FPGA

5.2.1 Non modified files (Evarist system)

- cypress_if.vhd
- elapsed.vhd
- sequencer.vhd
- pll1cycV.vhd

5.2.2 Modified files (Application and package)

- test_cycloneV.vhd: Top level file of the project
- applic_ctrl.vhd: File which control the correct execution of the application
- applic_wrp.vhd: File containing the application
- gen_clk_wrp.vhd: File which generate the clock for the application
- lab_hc_pkg.vhd: File containing all types for the project
- lut_and.vhd: LUT based *AND* gate
- lut_not.vhd: LUT based *NOT* gate
- tero_chain_cycV.vhd: File containing one branch of the TERO cell
- tero_cycV.vhd: File containing the full TERO cell

5.2.3 Generated file

- test_cycloneV.qsf: Placement file, modified to place all of the 256 TERO cells