International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue (4) July, Technovision-2014, ISSN 2249–071X

Technovision-2014: 1st International Conference at SITS, Narhe, Pune on April 5-6, 2014

Configurable Ring Oscillator for FPGA Chip Identification

Kavita C. Mugali
Department of E & TC Engineering
Sinhgad Academy of Engineering, Kondhwa, Pune, India
Email: kavitamugali@yahoo.com

Prof. Minakshee M. Patil
Department of E & TC engineering
Sinhgad Academy of Engineering, Kondhwa, Pune, India
Email: minaksheepatil.me@gmail.com

Abstract **– Hardware security and intellectual property protection have become very important for vendors. Instead of storing identification information in the device, the physical unclonable functions (PUF) are widely used for identification. There are various techniques for PUF implementations. PUFs extract secretes from physical characteristics of integrated circuits. They have the unique property of generating volatile chip specific variation signatures at runtime. It protects the ICs against attacks by offering robust security shield. The generated ID should be unique and repeatable. The Xilinx Spartan-3 field programmable gate array is used for implementing the design.**

Keywords **– Configurable Ring Oscillator, Physical Unclonable Function, Field Programmable Gate Array.**

I. INTRODUCTION

Chip identification of integrated circuits (IC) has become very popular in the vendors of intellectual property (IP) providers as they have to worry about the security of their designs and products. No one has guaranty their ICs are foolproof. So generating a unique binary string as identification no (ID) for produced chips of same design is desirable in embedded systems. It has wide range of applications including public key cryptography, digital right management, integrated circuit counterfeit detection and prevention, to identify nodes on networks. Field programmable gate arrays (FPGA) are commonly used for system on chip applications and embedded system designs and are the main platform to implement the design. So they need the facility of chip identification.

Many FPGAs have such features. Vertex-II pro platform FPGAs have advanced triple DES (data encryption standard) security embedded in them [1]. In CoolRunner-II CPLD the two technology schemes DualEDGE and DataGATE confuse attackers with double data rate operation and input signal locking under internal macrocell control. Xilinx Spartan-3 FPGAs have "Device DNA" facility that protects the design from cloning.

The traditional approach is to store the ID information in volatile or non-volatile memory in an FPGA. Storing digital information in a device in a way that is resistant to physical attacks is different and expensive. Adversaries can physically extract secrete keys from EEPROM. Also the EEPROM adds additional complexity to manufacturing. So instead of storing identification information in a device, physical unclonable function (PUF) can be used to extract the physical variation in the chip to distinguish it from other. The chip ID can be calculated by converting mismatch in delay, voltage or current values of the array of circuit structures of identical design into binary string. The process of averaging and thersholding is applied on the extracted variation and chip ID is generated.

The chip IDs thus generated should be unique and repeatable to avoid ID collisions between devices. The device should return the same value of ID every time. The ring oscillator (RO) PUF can be used to generate the ID. The mismatch in transistor delays gives random output from group of ROs which are then converted into binary sting by averaging and thresholding. The configurable ROs give better output when used with runtime re-initialization scheme.

The rest of the paper is as follows. Section II summaries the previous work on chip ID generation and physical unclonable functions (PUF). The architecture of chip ID generator circuit is described in Section III. The results are presented in Section IV. Section V describes conclusion.

II. BACKGROUND

PUFs are innovative circuits and enable low cost authentication of individual ICs. There are various types of PUF representations. The summary of relevant work is given below.

A. PUF on ASICs

Lofstrom et al [3] developed integrated circuit identification (ICID) that extracts unique and repeatable information from the randomness inherent in silicon processing. This technique can be used with any standard submicron CMOS process. ICID uses an array of addressable MOSFETS with common gate and source. Drain currents are different due to device mismatch, which causes sequence of random voltages across the load. These sequences of random but repeatable voltages are used to construct the unique identification. The device fabricated using 0.35 µm single-poly N-well process s used. A 112 bit ID is generated with less than 4% drift with less than 10-7 error rate with wide range of power supplies, biases, clock frequencies, temperature.

International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue (4) July, Technovision-2014, ISSN 2249–071X

B. PUF on FPGAs

FPGA-based PUF architectures are classified in types: memory-based, logic-based, arbiter-based and ring oscillator-based.

1) Memory-based PUF: The static RAM cell can be used to produce the ID bits. It is the best known intrinsic PUF based on standard available components. Gaujardo et al. showed that 4% bits vary over time over temperature range - 20o c to 80o c [4].

2) Logic-based PUF: Anderson [5] implemented the PUF in Vertex-5 FPGA using configurable logic block (CLB) in it. The 6-input look up tables (LUT) in the two slices of CLB, four flip-flops and other arithmetic circuitry is utilized to implement the PUF design. The vertical chain of multiplexers called the carry chains intended for fast arithmetic operations are also used. A 128-bit signature is generated on Vertex-5 FPGAs on Xilinx XUPV5-LX110T development boards. Six such boards are tested. The bit flip rate of only 3.6% is observed under high temperature conditions.

3) Arbiter-based PUF: The arbiter based PUF consists of two parallel multiplexer chains feeding a flip-flop. The input transition travels through a series of 2-input/2-output switches. Each switch is configured to be either a cross or straight connection based on its selection bit. The difference between the top and bottom path delays is compared by the arbiter and the response bit is generated. Suh and Devadas [13] used this technique and achieved 0.7% unstable bit rate at room temperature and fixed supply voltage [6].

4) RO-based PUF: The RO based PUF uses differences in the periods between similar ROs. RO is encapsulated in the hard macro with fixed layout. In comparison with the arbiter based PUF, the RO based PUF achieves better performance [6]. Maiti and Schaumont [7] proposed configurable RO which achieves higher reliability in RO based PUF. This approach is more efficient in terms of hardware cost. The common centroid arrangement is used to better counter spatial correlations [2].

III. DESIGN

A. One-Bit Generation

The proposed bit generation scheme consists of a 2 × 2 array of four ROs which are placed in common centroid layout as shown in Fig. 1 to mitigate correlations due to spatial process variation as in [8]. Such an arrangement is called as cell. The cells are arranged in overlapped fashion instead of individual one as in [2]. This improves the resource efficiency by a factor of four. So according to this design 9 × 9 RO array is required instead of 16 × 16 RO array to generate 64-bit ID.

A timer having system clock of frequency fclk is used to measure the number of rising edges of RO, NRO, over a period of Ntimer cycles. The frequency of particular RO is then given by

$$N_{RO} = (f_{RO} \times N_{timer}) \div f_{clk} \quad (1)$$

If NA, NB, NC and ND are the counter values of four ROs A, B, C and D respectively, then the residue can be calculated as

$$R_i = (N_A + N_D) - (N_B + N_C) \quad (2)$$

If Ri is positive then the 0 bit is generated by the cell, otherwise it is 1. It is also called as polarity. Due to transistor delay variations, a random output of cell *i*, R*i*, is obtained from difference in period of ROs with the same layout but different spatial locations. The R*i* is normally distributed with the expected value E(R*i*) of 0.

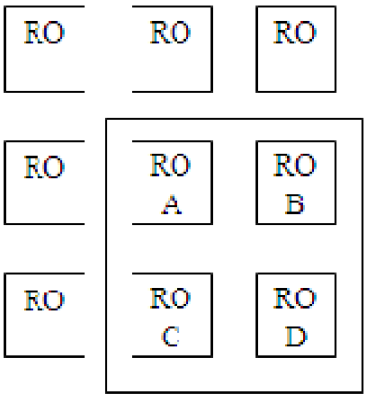

Fig.1. One- bit ID generation (Block diagram of cell).

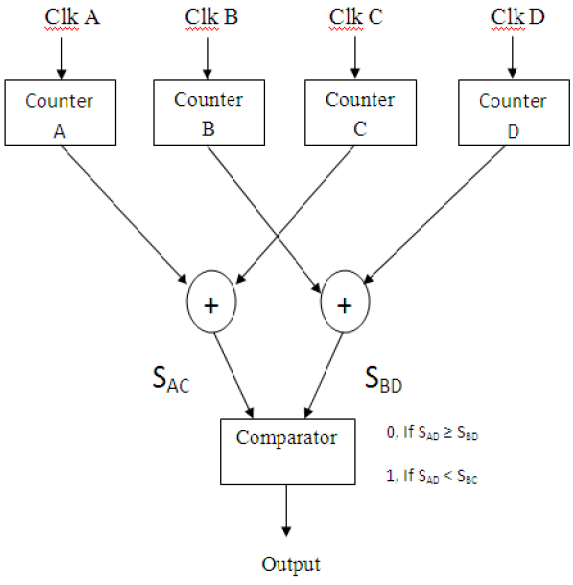

Fig.2. One bit computation.

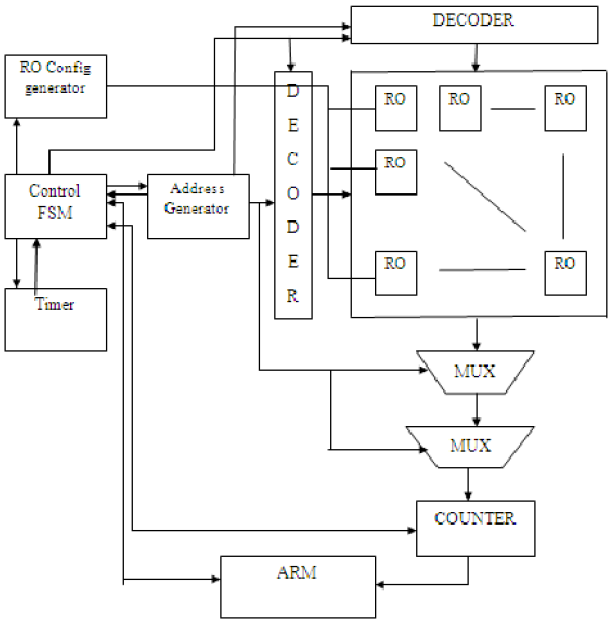

Fig.3. Architecture of chip ID generator.

B. Architecture

Fig. 3 shows the architecture of the chip ID generator. The measurement circuitry consists two main parts. An array of identical ring oscillators and the controller to measure and compare their differential delays. The 9 × 9 RO array provides 8 × 8 cells. The cells are arranged in spatially overlapped fashion instead of individual ones. This arrangement saves the significant logic resources and also maintains good statistics for ID generation. These cells can generate 64 separate bits. (i = 0,…, 63) The address generator selects the single RO with the help of two decoders. A 4-bit global RO configuration signal is sent to each RO. The configuration only affects the operating RO as only one RO can be activated for a given time interval. The finite state machine (FSM) controls an internal timer and address generator so that the ID can be generated sequentially. Two levels of multiplexers route the output of selected RO to the counter. Each RO is connected to the clock input of counter and the number of RO clock periods occurring during a timer interval is measured. The handshaking signals connect the timer to the ARM processor and the residue is calculated according to (2). The handshaking signals include the one bit enable signal to start the timer operation.

The post processing to facilitate different experiments with ID generator is implemented on an external ARM processor in software.

C. Configurable RO

Fig. 4 shows the circuit for configurable RO. The RO is implemented in Xilinx Spartan-3 FPGA. The design can be easily ported to different FPGA families. The four-stage RO is designed with each stage occupying two Xilinx logic elements (LE) within a slice and a multiplexer is used to choose a single path. The entire RO occupies a single Xilinx configurable logic block (CLB). By selecting different values of S0 – S3, 16 different configurations can be selected. Logic and interconnect delay mismatch in paths changes the frequency of ROs.

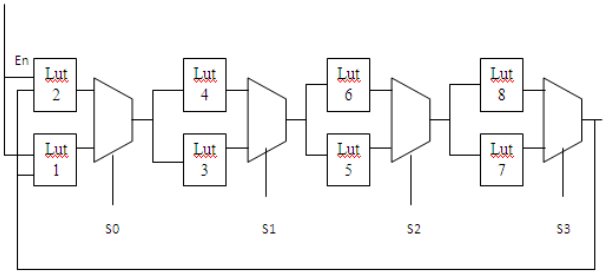

Fig.4. Configurable ring oscillator.

D. Configuration Initialization

The configuration bits of each RO need to be initialized on power up for generation of stable ID. One approach can be to determine the configurations when the FPGA is powered up the first time and to store them in non-volatile memory. But there is possibility of information leakage when the digital information is stored in memory. The communication channel is required to transfer the configuration bits to chip for ID generation process. Adversaries can extract this information leading to modeling attacks [9].

The other approach gives better results by analyzing three types of cells. Cell #1 produces negative Ri values for all configurations (negative polarity), Cell #2 produces positive values (positive polarity) and Cell #3 produces both positive and negative polarity values (hybrid polarity) for all configurations. The configuration with maximum Ri values are selected for ID generation to maximize the stability. The residues over all possible configurations are added and the sum, SR is calculated for all configuration values which are denoted as c.

$$S_R = \sum_{c=0\times0}^{0\times f} R_i(c) \qquad (3)$$

When the residue for all configurations has the same polarity, then the best configuration among them can be determined by the largest absolute value. The polarity of a residue of the particular cell does not change even if the configuration changes with time. In case of hybrid polarity cells, the threshold value can be applied to SR. when it gets difficult to determine whether the particular cell has positive or negative polarity then divide the configurations into two halves, 0000 to 0111 and 1000 to 1111 and also divide the threshold value. Thus the polarity can be easily identified by checking whether the residue in first half is larger or smaller than the new threshold value.

Table I: Logic Utilization

Resource	Consumption	Total	Percentage
Number of slice flipflops	31	7168	0
Number of 4-input LUTs	793	7168	11
Number of GCLKs	1	8	12

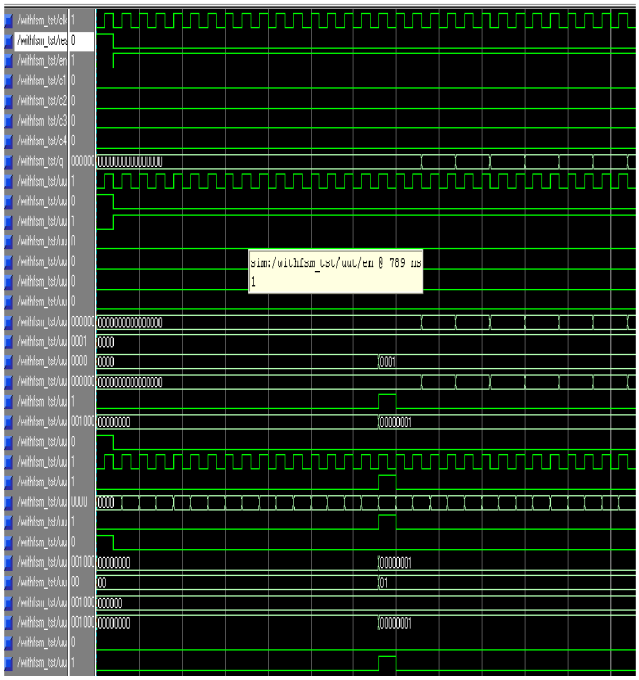

Fig.5. Simulation results on FPGA

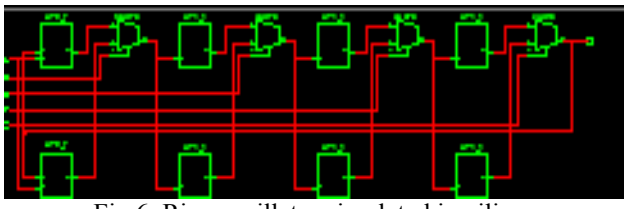

Fig.6. Ring oscillator simulated in xilinx

If the polarity still cannot be determined then the above process can be applied iteratively until the polarity is identified.

E. Run-Time Re-initialization

RO frequency gets affected by temperature and supply voltage. As they change, the |Ri| values in the selected configuration may change. So the dynamic re-initialization technique can be used to improve the repeatability of chip ID. The re-initialization is done by tracking the temperature and voltage variations in the circuit.

IV. RESULTS

A. Hardware and Software Resources

The design is implemented on a custom board with a Xilinx Spartan-3 FPGA (xc3s400-PQ208) and a NXP LPC 2148 ARM processor. Xilinx ISE Design Suite 10.1 and µVision v4 are used respectively for FPGA design and ARM C compilation.

B. Results

Table I summarizes the resource consumption in FPGA. The design is simulated on FPGA with T flip-flop with clock frequency of 4 MHz. Fig. 5 and 6 show the simulation results on FPGA.

V. CONCLUSION

The robust chip id can be calculated using configurable ring oscillators, power up initialization and re-initialization techniques with improved stability and repeatability. The design can be implemented with minimum resource utilization. The design can also be implemented in ASICs.

REFERENCES

[1] A. Telikepalli, “Is your FPGA design secure?,” Xilinx *XCELL*, pp.25-32, Fall, 2003.

[2] H.Yu, P. Leong, H. Hinkelmann, L. Moller, M. Glesner, and P. Zipf, ”Towards a unique FPGA-based identification circuit using process variations,” in *Proc. Int. Conf. Field Program. Logic Appl. (FPL)*, 2009, pp. 397-402.

[3] K. Lofstrom, W. R. Daasch, and D. Taylor, “IC identification circuit using device mismatch,” in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, 2000, pp. 372-373.

[4] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, “Physical unclonable functions and public- key crypto for FPGA IP protection,” in *Proc. Int. Conf. Field Program. Logic Appl. (FPL)*, 2007, pp. 189-195.

[5] J. Anderson, “A PUF design for secure FPGA-based embedded systems,” in *Proc. Int. Conf. Field Program. Logic Appl. (FPL)*, 2009, pp. 1-6.

[6] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proc. 44th Annu.Design Autom. Conf.(DAC)*, 2007, pp. 9-14.

[7] A. Maiti and P. Schaumont, “Improving the quality of a physical unclonable function using configurable ring oscillators,” in *Proc Int. Conf. Field Program. Logic Appl. (FPL)*, 2009, pp. 703-707.

[8] Y. Su, J. Holleman, and B. Otis, “A 1.6 pj/bit 96% stable chip-id generating circuit using process variation,” in *IEEE Int. Solid-State Circuits Conf. Dig. Techn. Papers (ISSCC)*, 2007, pp. 406-611.

[9] U. Ruhmair, F. Schnke, J. Solter, G. Dror, S. Devadas, and J. u. Schmidhuber, “Modeling attacks on physical unclonable functions,” in *Proc 17th ACM Conf. Comput. Commun. Security*, 2010, pp. 237-249.