**Chip Identification Generator by Butterfly-PUF with RO**

**CHAUHAN MAULIK B.**¹, J.K.KASTHURI BHA²

¹²Department Of ECE, SRM University, kattankulathur, Chennai, INDIA

¹cmaulik26@gmail.com, ²kasthuriroshan@gmail.com

---

**ABSTRACT**

Physically unclonable functions (PUF) are commonly used in applications such as hardware security and intellectual property protection. Various BPUF implementation techniques have been proposed to translate chip-specific variations into a unique binary string. Counterfeiting of goods in general and of electronic goods in particular is a growing concern with a huge impact on the global economy, the society, and the security of its critical infrastructure. BPUFs will be used as the hardware from which the key is extracted and can be considered as the intrinsic electronic fingerprint or biometric of a device. An implementation on a Xilinx Spartan-3e family chip set.

**Keywords:** Butterfly physically unclonable functions (BPUF), linear feedback shift register (LFSR), Pseudo Random Sequence Generator (PRSG), physically unclonable functions (PUF), ring oscillator (RO).

---

**[1] INTRODUCTION**

Chip identification, in which unique binary strings are associated with integrated circuits of the same design, has a wide range of applications including digital intellectual property protection, integrated circuit counterfeit detection/prevention, and public-key cryptography. Field-programmable gate array (FPGA) is a mainstream hardware implementation platform and need to be equipped with chip identification capabilities. Uniqueness and repeatability is the major factor in chip ID generation. Many of the chip identification process, physically unclonable functions (PUF) are used generate chip ID. The PUF performance based on the Ring Oscillator (RO) characteristics. Repeatability of Chip ID affect by temperature and voltage variation. By controlling these we can improve designing level of RO. Uniqueness and repeatability also affect by inter-chip and intra-chip variations. It can be maintained by introduce the Pseudo Random Sequence Generator concept (PRSG). Today’s commercial FPGAs already contain such features works. Example, in Xilinx Virtex-7 devices, a bits stream can be encrypted using a secret key. When the bit stream is downloaded, hardware (h/w) decrypts the bit stream. The bit stream only operates correctly if the device was programmed with the same key.

Xilinx also provides “DNA of Device” in Spartan-3e series FPGAs to protecting designs from cloning to stop the make the building of the circuits, which can be used to implement design which only operates with a specific ID. Ring oscillator are often used for generate the PUF IDs. Chip IDs generated in this way should be unique and repeatable. Uniqueness is required to avoid ID collisions between devices, while repeatability is necessary to ensure that a given device returns the same value every time. We use the term unstable to describe a chip ID with low repeatability.

Physical unclonable function is a function that is embodied in a physical structure and is easy to evaluate but hard to predict. One common method is to use a cell consisting of two or more ROs. Due to transistor delay variations, a random output for cell, can be obtained from the difference in period of ROs with the same layout but different spatial locations. The contributions of this work are summarized as follows.

- A cell which uses a number of ring oscillators with slightly different, configurable path-delays. They are arranged in a spatially overlapped fashion, saving significant logic resources while maintaining good statistics for ID generation.
- A power-up initialization and dynamic re-initialization process which selects and stores paths with the largest. Re-initialization serves to improve repeatability in the presence of varying voltage and temperature.

---

**[2] BACKGROUND**

PUFs are generally used for the hardware security research community since they were proposed in 2001 [3], [4]. Various PUF implementations on both application-specific integrated circuits (ASICs) and FPGAs have been reported. A general discussion of summary works is given in the following sub-sections.

**2.1 PUF on ASICs**

Lofstrom et al. [5] used arrays of addressable NMOS transistors loaded with a common resistive load. Drain current (id) mismatch caused the voltage (v) across the load to be different for different transistors in the particular array. By addressing the transistors in the array sequentially, a sequence of voltage was generated and successive values converted to a binary sequence via an auto-zeroing comparator to form an ID. A 112-bit ID circuit was shown to have a drift of less than 4% over a wide supply voltage (V) and temperature (T) range. Su et al. [6] reported on an improved circuit which used cross-coupled logic gates to simultaneously generate Amplify and digitize transistor mismatch. This circuit was able to produce a 128-bit, 4% unstable ID using only
1.6 pJ/bit. Helinski et al. proposed another PUF design based on measured equivalent variations of resistance in the power distribution system.

2.1 PUF on FPGA
FPGA-based PUF implementations can be categorized into the given following types: memory-based, logic-based and arbiter-based.

1) Memory-Based PUF: Guajardo et al. utilized the initialization state of SRAM cells in an FPGA and showed that they had valuable statistical properties for producing an ID [7],[8]. His experimental results showed that 4% of the start-up bits from the same RAM changed with times. Over a 20° to 80 degree temperature range, bit strings had a maximal fractional Hamming distance of 12% compared to a reference at 20°C. Holcomb et al. [9] proposed to a Fingerprint Extraction and Random Numbers in SRAM extraction system that harvests static identity and randomness from existing volatile CMOS memory without requiring any particular extra circuitry.

2) Logic-Based PUF: Patel et al. counting depends on variation on glitches and the output of a combinational multiplier to generate unique identification [10]. They found that 6 out of 64 bits are changed with the particular temperature range. Anderson used an FPGA’s carry chain to implement a PUF [11]. On average, 3.6% of bits are changed in high temperature (T).

Fig.1 an arbiter delay circuit. the circuit creates two delay Paths, layout length for each input X, and produces an output Y based on which path is faster.

3) Arbiter-Based PUF: Fig.1, illustrates a silicon PUF delay circuit based on MUX and an arbiter. The circuit has a multiple-bit input X and computes a 1-bit output Y based on the relative delay difference between two paths with the same length of layout. The input bits determine the delay paths by controlling the MUX. Here, a pair of MUX controlled by the same input box.

Fig.2 one-bit ID generation,(a) block diagram of a cell. (b) 1-bit ID calculation
To evaluate the output for a particular input, a rising signal is given to both paths at the same time, the signals race through the two delay paths, and the arbiter (latch) at the end decides which signal is faster. The output is one (1) if the signal to the latch data input (D) is faster, and zero (0) otherwise. Because the PUF circuit is rather simple, attackers can try to construct a precise timing model and learn the parameters from many input-output (I/O) pairs [7]. To prevent these model-building attacks, the PUF circuit output can be obfuscated by XOR operation, multiple outputs or a PUF output can be used as one of the MUX control signals.

3 PRINCIPAL OF OPERATION
3.1 Single-Bit Generation
Our proposed bit generation is achieved via a 2*2 RO array. The four ROs are placed in a common centroid layout, as show in Fig. 2(a) to mitigate correlations due to spatial process variations on the die. The effect of removing spatial correlation by adopting a common centroid layout has been shown in [13]. Such an arrangement is called a “cell” in this work and generates a single bit. We adopt an overlapped cell composition rather than the disjoint one used in our previous work [2]. This serves maintaining the Integrity of the Specifications to improve the resource efficiency of the design by a factor of four. As an example, to generate a 64-bit ID, the new scheme requires a 9*9 RO array compared to 16*16, and the randomness of the generated bits is not compromised.

4 IMPLEMENTATION
4.1 Architecture
Fig. 3 illustrates the architecture of our chip ID generator design. It includes a 9*9 RO array providing 8*8 cells. That can generate 64 individually bits. The circuitry to generate the chip ID consists of two main parts: an array
of identical ring oscillators, and a controller to measure and compare their differential delay. A total of 64 individual ring oscillator cells, are selected via decoding logic and the ID generation circuitry is shared. A finite state machine (FSM) controls an internal timer and address generation so, that the ID can be generated sequentially. In the physical layout, the control circuitry is separated from the ring oscillator (RO) array to reduce the chance of coupling between the two. As common in analogue layout, dummy cells were also used around the outside of the array so that each of the ring oscillator cells has identical neighbouring elements.

4.2 Configurable RO
A ring oscillator is a device composed of an odd number of NOT gates whose output oscillates between two voltage levels, representing true and false. The simple inverter diagram shown in the fig.4. The NOT gates, or inverters, are attached in a chain. the output of the last inverter is fed back into the first. Because a single inverter computes the logical NOT of its input, it can be shown that the last output of a chain of an odd number of inverters is the logical NOT of the first input. This final output is asserted a finite amount of time after the first input is asserted, the feedback of this last output to the input causes oscillation. A circular chain composed of an even number of inverters cannot be used as a ring oscillator; the last output in this case is the same as the input. However, this configuration of inverter feedback can be used as a storage element; it is the basic building block of SRAM.

The stages of the ring oscillator are often differential stages that are more immune to external disturbances. This renders available also non-inverting stages. A ring oscillator can be made with a mix of inverting and non inverting stages, provided the total number of inverting stages is odd. The oscillator period is in all cases equal to twice the sum of the individual delays of all stages.

A real ring oscillator only requires power to operate; above a certain threshold voltage, oscillations begin spontaneously. To increase the frequency of oscillation, two methods are commonly used. Firstly, the applied voltage may be increased; this increases both the frequency of the oscillation and the current consumed. The maximum permissible voltage applied to the circuits limits the speed of a given oscillator. Secondly, making the ring from a smaller number of inverters results in a higher frequency of oscillation given a certain power consumption.

4.3 Operation work
In work, we have to add the pairs of the NOT gate in the Ring oscillator and making the different types of the behavior of the each Ring oscillator. Adding the each part of the ring oscillator in pairs of the NOT gate in the each blocks. The simple logic shown by adding the NOT gate in fig.4, after that, each block gives to the input signals. They generates different time frequency signals those each RO’s are generating different-different signals with time delays. In the fig.5, whole system is there according to the block the ID bit generates. The use of the adding pairs of the odd numbers of NOT gate getting signals. Ring oscillators are widely used in ICs to generate clocks and characterize performance.

---

![Fig. 3 block diagram of chip ID generator architecture](image)

![Fig.4 ring oscillator adding of NOT gate](image)
4.4 Linear Feedback Shift Register
An LFSR is a shift register that when clocked, advances the signal through the register from one bit to the next most-significant bit. Some of the outputs are combined in Exclusive-OR configuration to form a feedback mechanism. Simply we said like that a Linear feedback shift register is linearly taking the value of the feedback and passed to the input port. A linear feedback shift register can be formed by performing exclusive-OR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input. Also the LFSR is based on the primitive polynomials. So based upon the polynomials we select the how the feedback path is given towards the input port and making the 64-bit LFSR. A maximal-length LFSR produces the maximum number of PRSG (Pseudo Random Sequence Generator) patterns possible and has a pattern count equal to \( 2^n - 1 \), where \( n \) is the number of register elements in the LFSR. Using the 64-bit LFSR we can produce the different Chip ID bits.

4.5 Physically unclonable functions (PUF)
Physical unclonable function is a function that is embodied in a physical structure and is easy to evaluate but hard to predict. Further, an individual PUF device must be easy to make but practically impossible to duplicate, even given the exact manufacturing process that produced it. In this respect it is the hardware analog of a one-way function.

Rather than embodying a single cryptographic key, PUFs implement challenge–response authentication. When a physical stimulus is applied to the structure, it reacts in an unpredictable (but repeatable) way due to the complex interaction of the stimulus with the physical microstructure of the device.

The block diagram shown in fig.6 This exact microstructure depends on physical factors introduced during manufacture which are unpredictable (like a fair coin). The applied stimulus is called the challenge, and the reaction of the PUF is called the response. PUFs can be implemented with a very small hardware investment. Unlike a ROM containing a table of responses to all possible challenges, which would require hardware exponential in the number of challenge bits, a PUF can be constructed in hardware proportional to the number of challenge and response bits. The internal structure of the PUF function is given into the Fig.8, it shows the how the bits come into the input bit, directly generated the random IDs. Unclonability means that each PUF device has a unique and unpredictable way of mapping challenges to responses, even if it was manufactured with the same process as a similar device, and it is infeasible to construct a PUF with the same challenge–response behaviour as another given PUF because exact control over the manufacturing process is infeasible. Mathematical Unclonability means that it should be very hard to compute an unknown response given. In other
words, given the design of the PUF system, without knowing all of the physical properties of the random components. The combination of physical and mathematical unclonability renders a PUF truly unclonable. Different sources of physical randomness can be used in PUFs. A distinction is made between PUFs in which physical randomness is extremely introduced and PUFs that use randomness that is intrinsically present in a physical system.

4.5 Butterfly PUF
Butterfly PUF are inherently difficult to implement on FPGA due to the delay skew present between a pair of circuit elements that are required to be symmetric in these PUFs. This static skew is an order of magnitude higher than the delay variation due to random process variation. The Butterfly PUF, proposed by Kumar et.al [17], is a technique that aims to emulate the behaviour of an SRAM PUF [16].

However, the functionality of this PUF is based on the delay variations of interconnects. A BPUF cell employs two cross-coupled latches, and exploits the random assignment of a stable state from an unstable state that is forcefully imposed by holding one latch in preset while the other in clear mode by an excite signal (Fig. 8). The final state is determined by the random delay mismatch in the pair of feedback paths and the excite signal paths due to process variation. A BPUF cell is presented. For a functional BPUF, the pair of nets AB/AC as well as XY/XY need to be symmetric along with the latches. In the Fig.9, block diagram of the BPUF circuit is given.

The function of this block diagram is when we apply the enable signal and according to apply input data value (data_IN). data is loaded into the circuit. According to the matching of the clk, enable and the rst signals the chip IDs generation at the IPout(7:0). In that port generated IDs will be the uniqueness property. Also, the port of Tb is the bi-directional, so according to that port we know which the IDs gives to which device and stored into the memory.

5 RESULTS
We implemented the system on a Xilinx ISE Design Suite 14.2 used for FPGA design. Also, the check the different possibilities of the to block the making of the cloning of the IDs. In the given respective figures show the simulation results of the different block circuits. In the table I. and table II. show the estimated value of the function and how much the device utilized in the system. Comparison table show the number of the sliced reduced. Also, Number of 4 input LUTs also reduced by the 5%. Also, in table III. and IV. show the power calculation of the chip ID architecture. Table V. show the utilization summary of the BPUF. Table VI. Show the total time regarding to route onto the chip system.

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of slices</td>
<td>169</td>
<td>768</td>
<td>22%</td>
</tr>
<tr>
<td>Number of flip-flops</td>
<td>164</td>
<td>1536</td>
<td>10%</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>312</td>
<td>1536</td>
<td>20%</td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>39</td>
<td>124</td>
<td>31%</td>
</tr>
<tr>
<td>Number of BRAMs</td>
<td>1</td>
<td>4</td>
<td>25%</td>
</tr>
<tr>
<td>Number of GCLKs</td>
<td>2</td>
<td>8</td>
<td>25%</td>
</tr>
</tbody>
</table>

**TABLE 5. TOTAL POWER**

<table>
<thead>
<tr>
<th>Source</th>
<th>Voltage</th>
<th>Current (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vccint</td>
<td>1.200</td>
<td>0.005</td>
</tr>
<tr>
<td>Vccaux</td>
<td>2.500</td>
<td>0.007</td>
</tr>
<tr>
<td>Vcc25</td>
<td>2.500</td>
<td>0.002</td>
</tr>
</tbody>
</table>

**TABLE 4. POWER CALCULATION**
[6] SIMULATION

6.1 Configurable RO
In the configurable RO the simulation result of this is shown in to the fig.10. The output result of the configurable RO show That, after the particular timing the ID will generate and the generated IDs are the uniqueness and the properties of the repeatability. Also, the here the condition of the circuit depends on the reset condition (rs=’0’ or ‘1’).

![Fig. 10 When rs=1](image1)

![Fig. 11 When rs=0 & en=1](image2)

6.2 Linear Feedback Shift Register (LFSR)
The Linear Feedback Shift Register (LFSR) is shown in the fig.11, with the 64-bit. Applying the condition of the reset and the enable bit. When we apply the enable bit ‘1’. So the run simulation. After that the when the condition s reverse means that the reset set to ‘0’, due to enable ‘1’ the IDs generated.

![Fig. 12 Butterfly-puf system](image3)

6.3 Implemented Design
When we used the BPUF on the circuit due to the coupling of the inverter the blocking of the repeatable(‘cloning’) chip IDs. if we try to change the condition of input value so, the according to the law we have to change the both the initial bits and the seed values. Condition of the implementing design is, we have to added the seed value with initial bits for the comparison of the bits IDs. In fig.13 shown the both input value changes. Then the new ID will produce. Otherwise the only the same ID shown in result.

CONCLUSION
Physically unclonable functions are circuits that leverage process variations to compute a unique signature for a fabricated IC. BPUFs have varied applications, including anti counterfeiting, hardware security, and cryptography. In this paper, we proposed the first FPGA-specific PUF and BPUF design – one that takes advantage of the FPGA logic and routing part. Compared with prior work, our design is easy to implement and incorporate into a surrounding design. Measured results on target on xc7a100t device, demonstrate the BPUF signature uniqueness and its reliability. The total time of the PUF take 2.63ns and for the BPUF 1.89ns. The BPUF the Future work will involve the development of FPGA flows that employ BPUF signatures for IP protection/licensing and anti-counterfeiting.

ACKNOWLEDGEMENTS
The authors would like to thank the SRM University, Chennai for accessing Cadence lab for helping with the design of this architecture design.
REFERENCES


