An FPGA Chip Identification Generator Using Configurable Ring Oscillator

Haile Yu[1], Philip H.W. Leong[2] and Qiang Xu[1]

[1] *Department of Computer Science and Engineering, The Chinese University of Hong Kong*
{hlyu,qxu}@cse.cuhk.edu.hk
[2] *School of Electrical and Information Engineering, University of Sydney*
philip.leong@sydney.edu.au

***Abstract*—An improved chip identification (ID) generator, otherwise known as a physically unclonable function (PUF) is described. Similar to previous designs, a cell, i, is used to obtain a measure of the difference in period of four ring oscillators and obtain the residue R_i, a random variable. Experiments show it is normally distributed with a mean of 0. A binary output value of 0 or 1 assigned depending on the sign of R_i. When $|E(R_i)|$ is large, this scheme consistently gives the same output. Unfortunately, when it is small, the repeatability is compromised, particularly when variations in operating conditions such as supply voltage and temperature are also taken into account, which is a common problem for all previous works. To address this problem, we propose a cell with configurable ring oscillators together with an orthogonal re-initialisation scheme. Together, these two techniques maximise repeatability by causing the distribution of the mean of different R_i's to change from normal to bimodal. We implement this design in the Xilinx Spartan-3e FPGA. Nine FPGA chips are tested, and experimental results show that the new method significantly enhances reliability of ID generation and tolerance to environmental changes. Bit flip rate is reduced from 1.5% to approximately 0 at a fixed supply voltage and room temperature. Over the $20-80°$C temperature range, and a 25% variation in supply voltage, the bit flip rate is reduced from 1.56% to 3.125×10^{-7}, which is a 50000$\times$ improvement.**

I. Introduction

Associating a unique identification (ID) string with an integrated circuit is necessary for chip identification and generation of keys in authentication systems. It has a wide range of applications including: key generation in public-key cryptographic systems; identification in smart cards; keyless entry systems; RF-IDs; identifying nodes in systems and networks; and digital intellectual property protection. Traditional techniques include writing a unique number to a random access memory or non-volatile memory, and post-fabrication modification of an integrated circuit using lasers or fuses.

Chip IDs can be obtained from the process variation associated with integrated circuit manufacture. Differences in the delay, voltage or current values of an array of identical circuit structures with different spatial locations have a random variation which can be extracted, averaged and thresholded to produce a binary output. 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 adjective *unstable* to describe a chip ID with low repeatability.

Several FPGA-based PUF Designs have been proposed. Guajardo et al utilized the initialisation state of static RAM cells in an FPGA and showed that they had good statistical properties for producing an ID [1]. Anderson used the carry chain to implement the PUF [2]. Suh and Devadas [3] used both path pairs and ring oscillators on FPGAs to generate an ID for cryptographic applications. Similar to [3], our previous work employed an array of ring oscillators on an FPGA [4]. A random output for cell i, R_i, is obtained from the difference in period of ROs with the same layout but different spatial locations. We show experimentally that R_i is normally distributed with an expected value of 0. Previous designs converted this to an ID bit by averaging a number of R_i readings to reduce noise and then applying a threshold equal to the expected value of R_i, $E(R_i)$. This threshold is chosen to ensure an equal distribution of 0's and 1's. Using this scheme the average number of unstable bits was reduced from 5.3% to 0.9% at 20°C. Within the range 20°C to 60°C, the percentage of unstable bits was less than 2.8%.

The contributions of this work are as follows:

- A scheme to change the distribution of the expected value of all R_i values $E(R_i)$ from normal to a bimodal one. This reduces the probability of its value being near the threshold and greatly improves the repeatability of the chip ID.
- A cell with a number of ring oscillators having slightly different, configurable delay paths. These cells can be used in an overlapped fashion, saving significant logic resources.
- An initialisation and re-intialisation process which selects and stores the path with the largest $|E(R_i)|$. Re-initialisation is shown in the paper to improve repeatability, particularly with varying temperature and voltage.

Experimental results show that the R_i's generated have the desired bimodal distribution and, after thresholding, the resulting IDs have improved statistical properties over a wide range of temperature and voltage.

The rest of this paper is organised as follows. An analysis of the chip ID generation process is given in Section II and the circuit design described in Section III. Experimental results and a statistical analysis are presented in Section IV. Finally, conclusions are drawn in Section V.

II. Principle of Operation

A. One-bit Generation

Bit generation is achieved via a 2×2 RO array. The four ROs are placed in a common centroid layout to mitigate correlations due to spatial process variations on the die. Such an arrangement is called a "cell". We adopt an overlapped cell composition rather than the disjoint one used in our previous work [4]. This served 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 a 16×16.

A timer driven by a 10 MHz system clock, f_{clk} is used to measure the number of rising edges of the RO, N_{RO}, over a period of N_{timer} cycles. The frequency of the RO is hence given by

$$N_{RO} = \frac{f_{RO}}{f_{clk}} \times N_{timer} \tag{1}$$

In this work, we use $N_{timer} = 2000$. The value of f_{RO} ranges from 170 MHz to 190 MHz at room temperature, resulting in an N_{RO} in the range of 34000 to 38000.

As mentioned earlier, four ROs arranged in a square are used. If N_A, N_B, N_C and N_D are the counter values for ROs *A*, *B*, *C* and *D* respectively, the residue is calculated as:

$$R_i = (N_A + N_D) - (N_B + N_C) \tag{2}$$

If R_i is positive, the bit generated by this cell is 0, otherwise, it is 1.

B. Sources of Instability

The R_i values of all cells were analysed and it was observed that R_i has a Gaussian distribution with 0 as mean value. Statistically, since the mean is zero, the most frequently occurring residues are close to this value, making it likely to become unstable. A simple averaging scheme was adopted to reduce this problem, which effectively eliminates unstable bits with a bias on either positive side or negative side. However, there are still scenarios in which R_i is close to zero. The averaging scheme fails to generate repeatable chip IDs in this case.

III. Implementation

A. Architecture

Figure 1 illustrates the architecture of our chip ID generator design. The central part is a 9×9 RO array. As stated in subsection II-A, four configurable ROs in a common centroid layout are used for each cell. Therefore, a 9×9 RO array provides 8×8 cells and this can be used to generate 64 separate bits ($i = 0, \ldots, 63$).

The address generator together with the two decoders select a single RO to operate over a given time interval. A 4-bit global RO configuration signal is also sent to all of the ROs. At any given time only one RO can be activated and hence the configuration only affects the working RO. Two levels of multiplexers are used to route the output of the selected RO to the counter. Handshaking signals are used to indicate to the ARM processor when the time interval has elapsed and the residue is calculated in software according to equation 2.

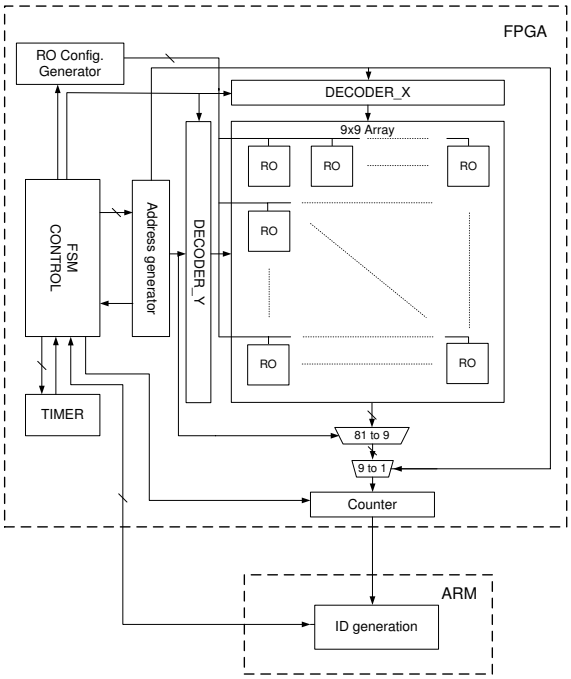

Fig. 1. Block diagram of chip ID generator.

To facilitate different experiments with the ID generator, postprocessing is implemented on an external ARM processor in software. The postprocessing could also be included in an on-FPGA processor or finite state machine.

B. Configurable RO

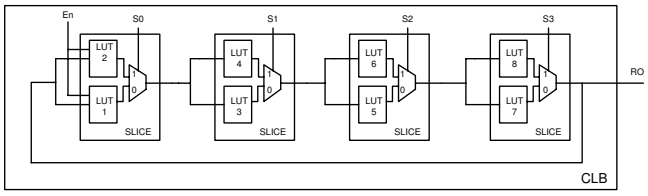

Fig. 2. Circuit for the configurable RO.

The circuit implementation of the configurable RO is shown in figure 2. A 4-stage RO is used where three of the stages are non-inverting and the final one is inverting. Each uses two Xilinx logic elements (LEs) within a slice and a multiplexer used to choose the signal path. The entire RO occupies a single Xilinx configurable logic block (CLB). It can be seen that 16 unique configurations are possible in the design. Logic and interconnect delay mismatch in the paths of the different configurations change the frequency of the RO.

Due to the observed systematic variations, generating R_i from ROs using different configurations would lead to correlated outputs. Instead, we use the same configuration for all 4 ROs, and choose the one with the largest $|E(R_i)|$. This scheme

employs one configurable RO to achieve a similar result to choosing from 16 normal ROs as done in Suh and Devadas's work [3]. This results in an improvement in area efficiency of up to a factor of 16.

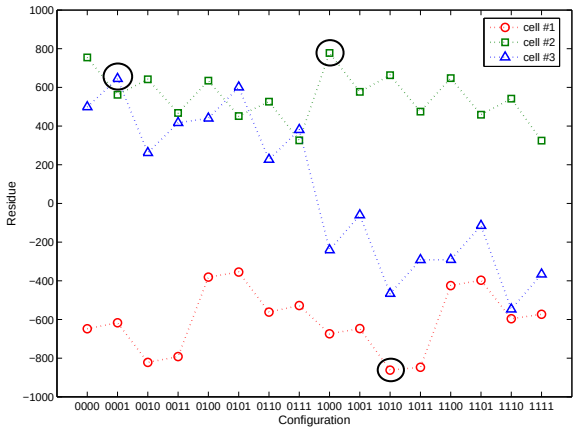

Fig. 3. Residues for all configurations from 3 cells.

In figure 3 three types of configuration patterns are shown. Cell #1 produces negative R_i values for all configurations, cell #2 all positive, and cell #3 has both positive and negative R_i values. The circles indicate the configurations with maximum $|R_i|$ which maximize the stability of the bit generated.

C. Chip ID Generation

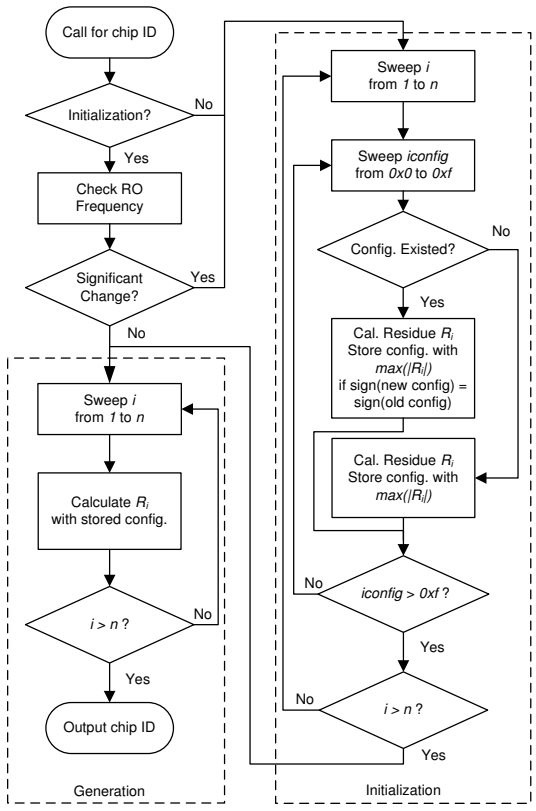

Fig. 4. Flowchart showing chip ID generation process.

Figure 4 describes the proposed process for chip ID generation. It can be divided into two phases, *initialisation* and *generation*. During initialisation, all configurations of all cells are swept to determine using which generates the largest $|E(R_i)|$. The information is stored for chip ID generation.

RO frequency is strongly affected by temperature and supply voltage [5]. As the temperature and supply voltage change, $|R_i|$ in the selected configuration may decrease and potentially become unstable. We propose a dynamic re-initialisation technique further improve reliability. Re-initialisation could be triggered by tracking the absolute RO frequency from time or an embedded sensor to track temperature or voltage variations [6] [7]. It can be run periodically using use one RO in the array as such a sensor. A re-initialisation is invoked to find a configuration with larger $|R_i|$ than the previous one with the same polarity if applicable. If the current configuration still remains the best one, no modification is made. Otherwise, a new best configuration is stored.

IV. Results

A. Statistical Analysis

1) Cell Configurations: Although the distribution is not uniform, strong biases towards a particular configuration were not evident.

2) One/Zero Ratio: The measured one-to-zero ratios of the 9 chips are listed in table I. On average, the one/zero ratio is 1.06, confirming an equal likelihood for each value.

TABLE I
One/Zero Ratio

Chip No.	1/0 ratio	Chip No.	1/0 ratio
1	0.88	6	1.06
2	1.06	7	0.94
3	1.00	8	1.13
4	1.20	9	1.13
5	1.13		

3) Hamming Distance: The Hamming distances between all pairs of chip IDs are summarized in table II The average value is 30, which is 47% of the bit width. This is very close to the ideal of 50% for independent IDs.

TABLE II
Hamming distance matrix.

	1	2	3	4	5	6	7	8	9
1	0	31	28	29	30	35	33	32	28
2	31	0	27	28	29	28	22	29	23
3	28	27	0	35	32	35	27	34	26
4	29	28	35	0	33	24	34	35	31
5	30	29	32	33	0	25	37	30	32
6	35	28	35	24	25	0	28	33	31
7	33	22	27	34	37	28	0	25	29
8	32	29	34	35	30	33	25	0	34
9	28	23	26	31	32	31	29	34	0

Taken together, the Hamming distance analysis and one/zero ratio demonstrate the generation scheme has very good statistical properties. It shows that the configuration selection scheme and overlapped cell composition, do not come at the cost of reduced randomness. In fact, the statistical properties are improved compared with our previous work [4].

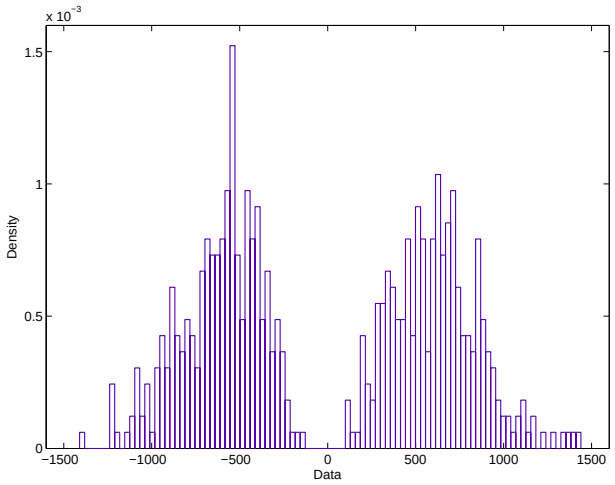

Fig. 5. Overall $E(R_i)$ distribution over all cells and all chips with configuration selection.

4) Stability Analysis: Figure 5 shows the $E(R_i)$s distribution over all chips and all cells. The configuration scheme modifies the distribution from Gaussian to a bimodal one and the occurrence of residues whose absolute values are close to zero are eliminated.

A bit flip occurs when a cell generates an R_i with sign opposite to the mean value. We define the *"bit flip rate"* P_{bf} as the number of occurrences of bit flips N_{bf} divided by the total number of bits generated N_{all}.

$$P_{bf} = \frac{N_{bf}}{N_{all}} \tag{3}$$

Experimentally, under the normal operating condition ($1.2V$, 20°C), we did not detect any bit flip for all nine chips over 50000 ID generations, which means P_{bf} is less than 3.125×10^{-7} by equation 3. In fact, bit flips were never recorded under normal operating conditions in any of our testing. The bit flip rate by measurement is approximately 0.

B. Environmental Influences

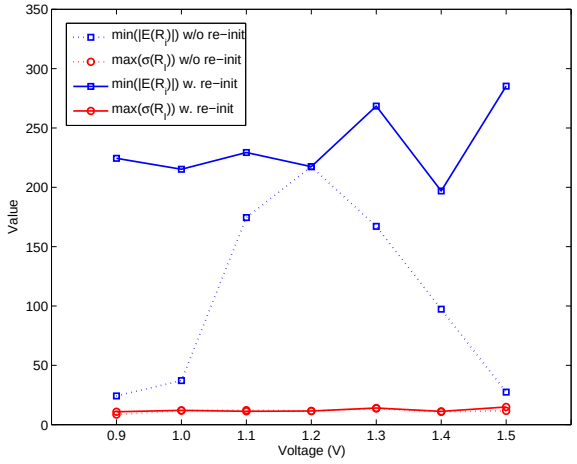

Fig. 6. Effect of re-initialisation.

To verify effectiveness of re-initialisation, configurations are initialised under normal operating condition (1.2 V and 20°C), then chip IDs are generated under 0.9 V and 80°C, which can be regarded as the worst-case operating condition for our tests. Without re-initialisation, one bit is constantly flipped as the sign residue is changed. Even without considering bit flips introduced by other cells, bit flip rate is $\frac{1}{64} = 0.0156$ by equation 3. In this case, a unique ID generation cannot be achieved even with post-processing. Using re-initialisation, as shown in figure 6, $min|E(R_i)|$ can still remain similar to that of the normal condition, which implies the bit flip rate should be similar. Measurement shows in the worst case, one bit flip occurs every 50000 repetitions, the rest of chips maintain the same ID over this range. Re-initialisation improves stability at least 50000×.

V. Conclusion

Chip identification has been widely used for digital system security and intellectual properties protection. For this purpose, the designers utilize intrinsic process variation of semiconductor device to produce a binary ID. However, unstable bits pose a challenge to this technique. To address this issue, we have demonstrated that a chip ID generation method with configurable RO and adaptive re-initialisation can considerably improves repeatability, meanwhile maintains high cost-efficiency. Results show that a very stable ID generation is achieved over a wide range of operating condition.

Since our design was completely implemented using standard digital circuits, it can also be implemented in an ASIC. As future work, the authors would like to develop more parallel schemes to speed up chip ID generation and introduce countermeasures to side-channel attack.

References

[1] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *CHES '07: Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 63–80.

[2] J. Anderson, "A PUF design for secure FPGA-based embedded systems," in *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*, jan. 2010, pp. 1 –6.

[3] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *DAC '07: Proceedings of the 44th annual Design Automation Conference*. New York, NY, USA: ACM, 2007, pp. 9–14.

[4] H. Yu, P. Leong, H. Hinkelmann, L. Moller, M. Glesner, and P. Zipf, "Towards a unique FPGA-based identification circuit using process variations," in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, 31 2009-Sept. 2 2009, pp. 397–402.

[5] G. Quenot, N. Paris, and B. Zavidovique, "A temperature and voltage measurement cell for VLSI circuits," in *Euro ASIC '91*, 27-31 1991, pp. 334 –338.

[6] S. Velusamy, W. Huang, J. Lach, M. Stan, and K. Skadron, "Monitoring temperature in FPGA based SoCs," in *ICCD '05: Proceedings of the 2005 International Conference on Computer Design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 634–640.

[7] K. M. Zick and J. P. Hayes, "On-line sensing for healthier FPGA systems," in *FPGA '10: Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. New York, NY, USA: ACM, 2010, pp. 239–248.