IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 6, Ver. IV (Nov.-Dec .2016), PP 33-43
www.iosrjournals.org

Realization of MUX-Based PUF for low power applications

Yarlagadda Tejaswini[1] Raghavaiah B[2]
[1]PG scholar, Department of ECE, CEC, Chirala, AP, India
[2]Associate professor, Department of ECE, CEC, Chirala, AP, India

Abstract: *In the process of extracting information, which is unique for every chip, physical unclonable functions (PUF) use variations in silicon fabrication process. There have been many recent approaches to improve security related applications by using PUF's. It is well known that, for pointing the security threat, the PUF uses a very strong spatial correlation in the process of silicon fabrication.The uncontrollable randomness in the process of manufacturing variations provides necessary strength to store the secrete keys in the integrated circuits (IC's). The PUF's can be used as authenticators for devices and for key generation in security applications. Various multiplexer-based PUF's which includes original MUX PUF, feed forward, modified feed forward multiplexer and de-multiplexer PUF's are presented.The inter chip and intra chip variations quantify the performance for number of stages, noise variance environmentally an also the arbiter skew for different PUF's. PUF's are more reliable if they have less intra chip variations. PUF's are more random if their response bit is 0 0r 1 having same probabilities.*

Keywords: *Puf, Mux, Verilog, Xilinx Ise.*

I. Introduction

The uncontrollable randomness in the process of manufacturing variations provides necessary strength to store the secrete keys in the integrated circuits (IC's). The PUF's can be used as authenticators for devices and for key generation in security applications. Various multiplexer-based PUF's which includes original MUX PUF, feed forward, modified feed forward multiplexer and de-multiplexer PUF's are presented. The inter chip and intra chip variations quantify the performance for number of stages, noise variance environmentally an also the arbiter skew for different PUF's. Trust is the sociological concept expressing the positive belief that a person or a system will behave as expected. In our day to day life, we constantly and often implicitly on other parities we put pour trust, e.g.: when we drive a car, we trust that the car will function as expected, that the breaks will work and car goes right when we turn the steering right. We also trust the other people who are driving cars around us are qualified while driving a car and paying attention to traffic. When we pay in advance we believe that we receive the genuine product we paid for. The shop owner trusts that we will pay for all products we carry out on the other hand. Our entire society is depends on such trust relations among people and systems and it would not last very long when no one or nothing could be trusted. However, we are not in an ideal world and it would be very naïve to think that everyone is intrinsically trust worthy.

Many parties have external motives to behave them in trustworthy manner, e.g. the shop and the bank won't get many customers when they are not be trusted, and the csr owners will primarily drive carefully for their own safety. Some parties cannot be trusted at all; we immediately thimk of criminals and terrorists, but this can also include distinguishable employees, envious colleges or nosy neighbors or even normally honest people who are tempted to abuse a situation when it presents itself. In our non ideal world we need systems that induce guarantee or even enforce trustworthiness of parties. Security is a means to enable trust which is called as call security., security is either based on physical protection and prevention measures, on observation and detection of untrusted elements, or on legal and other reprimands of trust violations, and often on combining of all these techniques from the past, and to a large extent still today. In order to keep money secure banks will store in vault.The access to this vault is moreover strictly limited to the bank's employees and protocols are in place to keep other people away (detection). Finally, by law, trying to rob a bank is also a criminal act for which one will be prosecuted when caught (legal reprimands). In our modern world, the techniques are no longer sufficient to enable trusted interactions, due to i) the nature of these interactions, and the scale of the possible threats. In order to provide these physical security objectives, we cannot rely on mathematical reductions anymore. Instead, we need to develop physical techniques and primitives which, based on physical reasoning, can be trusted to withstand certain physical attacks and can hence provide certain physical security objectives. We call such primitives physical roots of trust.Possible candidates of physical roots of trust are true Random Number Generators or TRNGs harvest random numbers from truly physical sources of randomness and can therefore be trusted to produce highly random keys for cryptographic purposes. Design styles for digital silicon circuits have been developed which minimize and ideally eliminate certain physical side channels.

Physical unclonable functions (PUF’s) are novel security primitives that have a capability of storing secrete keys in physical objects. PUF’s generate signatures depend on the unique intrinsic unctrollable physical features, which can be then used for hardware authentication or the generation of secrete keys. PUF’s extracts information from complex properties of a physical material rather than storing them in non volatile memory. It is nearly impossible to predict, clone or duplicate PUF’s furthermore; an adversary cannot easily mount attack to counterfeit. The secrete information without changing the physical randomness.Based on these advantages, PFU’s can efficiently and reliable generate volatile secrete keys to cryptographic operations and enable light and cost efficient authentications for IC’s.

II. Types Of MUX PUF

There are two types of MUX PUF’s and several subtypes, with their own applications and security features. A major type is the silicon PUF, which exploit the delay variations of circuit components to generate its own unique signature for each IC.

Silicon MUX PUF

Silicon PUF’s can be integrated into chips vary conventionally, since these are implemented with digital logic and no need of any special fabrication. The MUX PUF’s include silicon MUX PUF, ring oscillator PUF, SRAM PUF, and butterfly PUF. The silicon MUX PUF is the strong PUF that is unclonable because of its variations in manufacturing process, and can accommodate many possible challenge- response pairs (CRP’s). as illustrated in Figure, MUX PUF each challenge creates two paths through the circuit that are excited simultaneously.The output is generated based on the delay variations between the two paths. Depending on the challenge bit, one of the arbiter MUX in each stage act as switch for straight or cross propagation Of the rising edge signals, the MUX PUF is connected with the last stage of the two paths. Every MUX is designed equivalently because of the variations in the manufacturing process. Finally, the arbiter digital value is obtained by translating the analog timing difference. For instance, if rising edge arrives to the top of the input of arbiter earlier then the arriving to the bottom input, the output is one; otherwise zero if it reaches to the bottom path first.The output response depends on the applied challenge bits will be permanent for every IC after fabrication or only vary in a small range due to environmental variations.

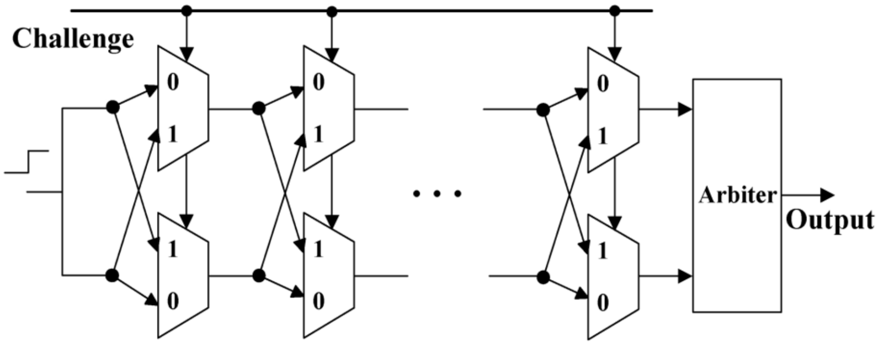


Figure.1: Silicon MUX Physical Unclonable Function

Depending on the transistor length, width, gate oxide thickness, doping density body bias, metal width, metal thickness, and inter level dielectric thickness (ILD) variations, manufacturing randomness changes. These manufacturing variations lead to significant amount of variability for MUX-based PUF’s which are sufficient to generate unique challenge-response pairs for every integrated circuit by comparing the delays of two paths.

Feed-Forward MUX PUF

For improving security to add non-linearity into PUF a fee forward structure is proposed. In feed forward MUX PUF, arbiter (FF arbiter) from intermediate stage is used for challenge to a subsequent stage. Fig 2.3 shows the result of intermediate stage as the selection signal for later mux stage one from the structure of feed forward MUX PUF. The numerical modeling complexity attacks from this structure increases. However, the reliability of a PUF has been degraded since some signals of the MUX may also be affected by environmental variations.

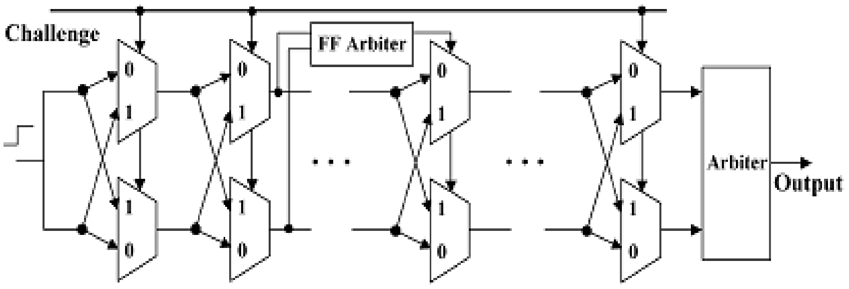


Figure.2: Feed-Forward MUX PUF

MUX-based Reconfigurable PUFs:

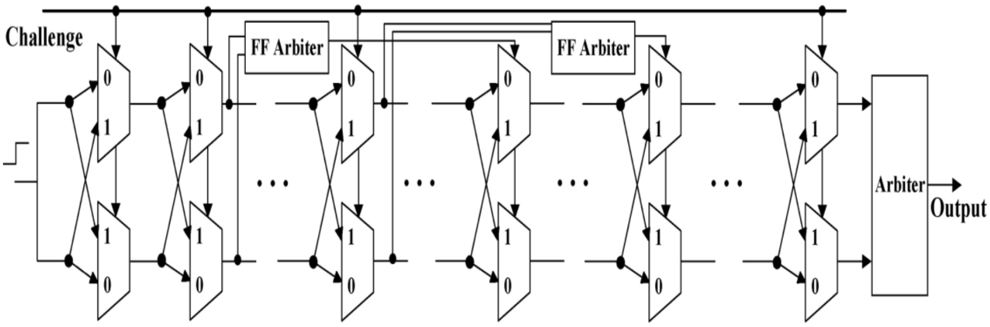
MUX based reconfigurable PUF's: based on the PUF MUX and feed forward MUXOUF variants, several novel reconfigurable PUF's are proposed.

Logic-Reconfigurable Feed-Forward MUX PUF:

Reconfigurable PUFs satisfy the updatable key requirement for PUF based authentication systems. Furthermore, reconfigurability improves the security against modeling attacks by limiting the information leaked for each configuration. Logic reconfigurable feed forward MUX PUF: the feed forward MUX PUF arbiter form intermediate stage used as challenge to a subsequent stage. There are three different types in feed forward MUX PUF. They are feed forward overlap (FFO) feed forward cascade (FFC), feed forward separate (FFS).These structures are classified by interconnections of various feed forward patterns in these PUFs. The performance of feed forward patterns in these feed forward MUX PUF depends on locations and the number of feed forward paths as shown below. The three feed forward structures are described below.

Feed-Forward Overlap (FFO):

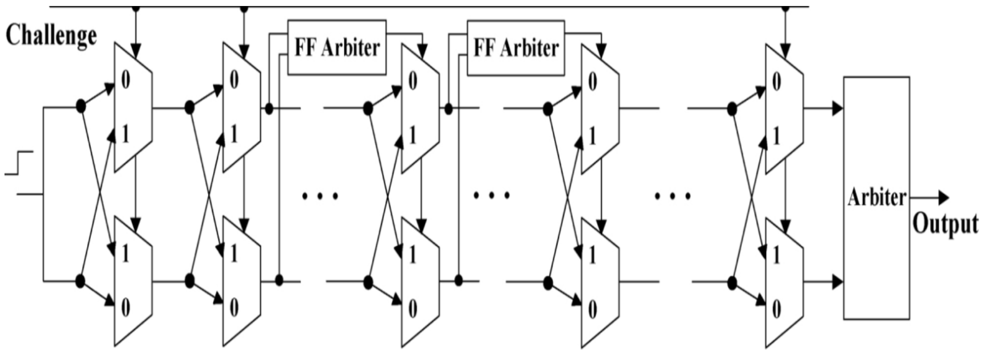
This structure has at least one stage overlap between two feed-forward paths Figure 3: Feed forward overlap MUX PUF. / Feed-Forward Cascade (FFC): The starting stage of another feed-forward path will be the ending stage of a feed-forward path as shown in figure 4



Figue. 3: Feed forward overlaps MUX PUF

Feed-Forward Cascade (FFC):

The starting stage of another feed-forward path will be the ending stage of a feed-forward path as shown in figure 4



. **Figure.4:** Feed forward Cascade MUX PUF

Feed-Forward Separate (FFS):
Different feed-forward paths are separate; thus, between the two paths no stage overlap exists as shown in figure 6

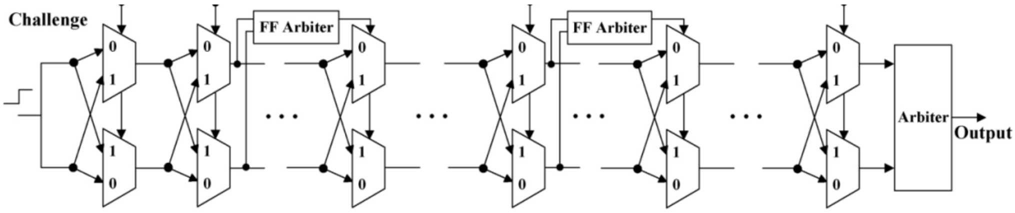


Figure.6: Feed-Forward Separate MUX PUF

Feed forward Cascade MUX PUF Feed-Forward Separate (FFS): Different feed-forward paths are separate; thus, between the two paths no stage overlap exists as shown in Figure 6: Feed-Forward Separate MUX PUF We have simulated these three feed forward structures [19],[20]. Based on this property, we have proposed a logic reconfigurable, which can be configured to any of these three different structures are feed-forward MUX PUF (i.e., FFO, FFC, and FFS)[19][20].

Novel Reconfigurable

In order to add reconfigurable property into general MUX based silicon PUFs, we must make the challenge-response pairs (CRPs) reconfigurable, which can be used to update the database for an authentication system. Two categories are done based on the classification: Make the challenge-response pairs reconfigurable directly, by adding some extra circuits into the structure, but without configuring the main PUF circuit.This can be achieved by utilizing some techniques to preprocess the challenge before applying to PUF or pre-process the response before using it for authentication. Make the PUF circuit reconfigurable, therefore the challenge response pairs will be reconfigurable as well. We propose several novel non-FPGA reconfigurable PUFs implementations for the above two categories, which would be more suitable for practical use than FPGA-based techniques.Furthermore, we address the reliability and the security of the PUF performance, as some information of the hidden secrets that an adversary can take advantage of may leak out during reconfigurations. 2.6.1 Reconfigurable Challenge and/or Response Structures The reconfigurable structures of PUF are built on Physical Unclonable Function and the prior work, which can also be applied to various types of silicon PUFs as well as other challenge-response based PUFs.

Our goal is to develop reconfigurable PUF which is a PUF with a mechanism to transform it into a new PUF with an unpredictable and uncontrollable even if the challenge-response behavior of the original PUF is already known. Additionally, besides of original one the new PUF inherits all the security properties. An early reconfigurable design PUF in the literature treated some challenge bits as the configure data. As an example, the last 10 bits of a 100-bit challenge can be fixed as the configure data, leaving only 90 bits for actual challenge. A user can update the CRPs by applying another 10-bit stream to the last 10 stages of the PUF. However, it is very clear that the reconfigured PUF will have high correlation between different configurations and will be vulnerable to attacks, as this method is similar to adding a certain time or introducing an interval between the two rising edge signals.Even worse, the performance of the PUF will be greatly degraded, if the cumulative variations in the last 10 stages are relatively large. Due to these disadvantages, this architecture of reconfigurable PUFs cannot generate unpredictable challenge-response behaviors. Intuitively, adding reconfigurable elements before the challenges applied to the PUF can definitely make the PUF reconfigurable.
The performance of the original PUF will be preserved at the same time,. The main structure of this type of reconfigurable PUF is shown in Figure 7.

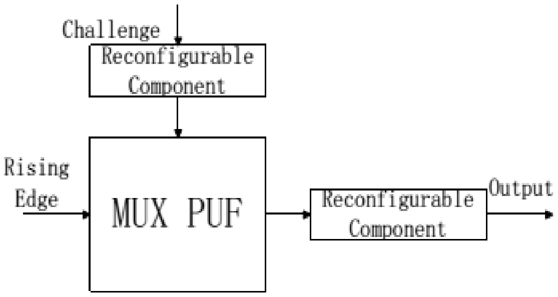


Figure.7: Reconfigurable Challenge and Response PUF Structure

PUF with Hash Function:

"one-way" function is nothing but Hash function, which means the hash value for a given message is easy to compute, but hard to find a message with a given hash. Due to the random property of hash function, we can generate a reconfigurable PUF that is employ with a hash function as the reconfigurable element . This reconfigured very easily, such as by adding several different lengths of 0's at the end of every challenge as shown in below figure 2.8. Additionally, the security of PUF can be increased, due to the "one-way" property of hash function.Many hash algorithms have been investigated and developed in the last years Currently, the SHA-1 algorithm is the National institute of Standards and technology (NIST) secure hash standard. In past years several reconfigurable hash function unit architectures have been published. In fact, this structure has already been named as Controlled Physical Unclonable Function , which was described as adding control logic to a PUF structure to prevent an adversary from accessing the PUF directly.

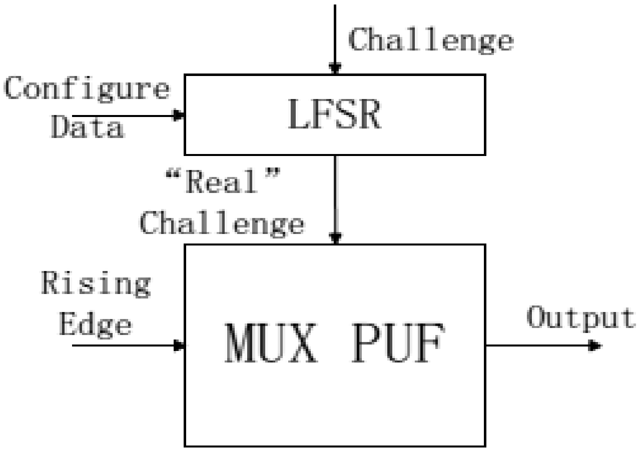


Figure.8: PUF Structure of Using LFSR to Configure the Challenge

PUF Structure of Using LFSR to Configure the Challenge Instead of doing a simple hash before the challenges we can consider adding another control logic, which would make the CRPs updatable. Several reconfiguration methods are propose.Re-ordering the challenge stream by certain rules. By using the re-configurability of these reconfigurable hash function implementations the hash function are reconfigurable.

PUF with Output Recombination

It is extremely hard for an adversary to model the PUF due to the property of hash function, even after we configure it several times, since unpredictable output will be given. 2.6.3 PUF with Output Recombination Another idea is to add an extra reconfigurable component to preprocess arbiter before using it as an authentication key.One simple example is to use two parallel MUX PUFs to update the CRPs, as shown in Figure 5. In this case, the signal (rising edge) will propagate through 4 paths which are selected by challenges. Then we can select two of the four paths using the configure data and forward to the arbiter to generate the response. If we use a 2 parallel MUX PUFs we will have a total of 12 possible combinations. Therefore, we can reconfigure this architecture 12 times. However, there will be very high correlations among these 12 different combinations. For example, conclude that path 1 will be faster than path 3 if we know that path 1 is faster than path 2, and path 2 is faster than path 3. Therefore, there should be some constraints for the preprocessing, N paths based on their arrival time by decreasing the total number of possible cases for ordering. Therefore, log2 (N!) independent bits can be produced by N paths. In order to meet the practical application needs we can increase the number of parallel PUFs to obtain more possible combinations.If we want to achieve the entropy limit as log2 (N!), we need to choose the output comparison pairs adaptively, which would increase the design complexity and fabrication area significantly. However, there will be a problem by employing this structure, since the preprocessing component after the last stage also has variations, which will affect the performance of the PUF. To solve this problem, we can add pre-processing components after the arbiters, as in structure of Figure 6. If we use N parallel MUX-based PUF, we will need 2N-1 arbiters, where we only compare the neighbor paths. As the comparison pairs are non-cyclic there is no correlation between the output bits of the arbiters and this is a concept borrowed from ring oscillator PUF. Therefore, this structure as shown in figure can update its challenge-response behavior in an unpredictable manner.

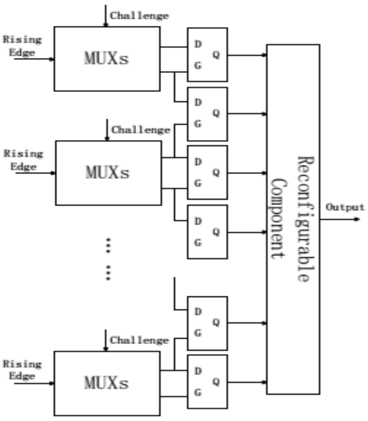


Figure.9: block diagram of puf mux implimentation

MUX/Demux PUF

Another MUX-based logic reconfigurable PUF is the MUX/DeMUX PUF, which alters the PUF logic by using DeMUX. DeMUX makes the original MUX PUF and enables the circuit to select the direction of the propagating signals, reconfigurable. A basic structure is shown in Figure 10.

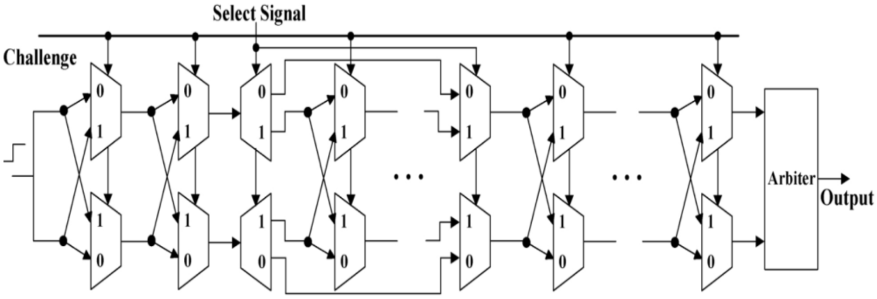


Figure 10: MUX/DeMUX PUF

Some stages can be skipped by the DeMUX instead of propagating the rising edge signal successively, which allows the challenge response behavior to be reconfigurable. Modified Feed-Forward MUX PUFs Modified feed-forward MUX PUF structure shown in Fig 11.which is motivated by our statistical analysis results.

Modified Feed-Forward MUX PUFs

Modified Feed-Forward MUX PUFs Modified feed-forward MUX PUF structure shown in Fig 11. which is motivated by our statistical analysis results. In this structure, arbiter from an intermediate stage is input as the challenge bit to two consecutive late MUX stages for the output of a feed-forward. By employing this modified feed-forward path, while the same level of security will be retained so that the reliability of the feed-forward PUF structure can be improved. By using several modified feed forward paths in a PUF circuit the complexity of the modified feed-forward MUX PUFs can be improved. Note if we need to increase the number of MUX stages to N+2M for the modified feed forward structure if we want to maintain the length of challenge bits as N,, compared to N+M of the standard feed forward PUF, where M represents feed-forward paths number. Additionally, both the standard feed forward MUX PUF and the modified feed forward MUX PUF include arbiters.

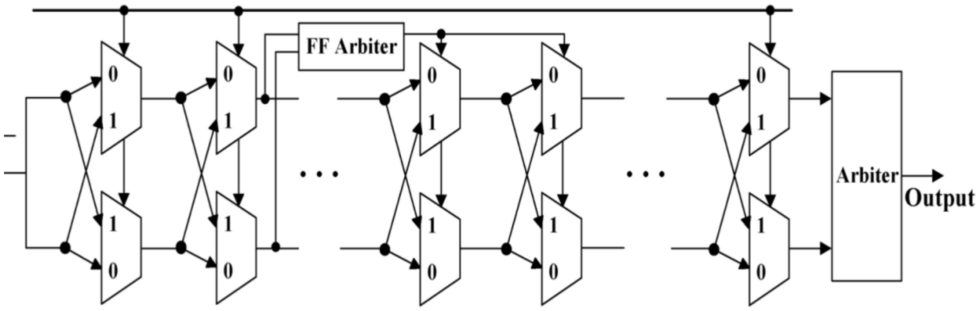


Figure 11: Modified feed-forward MUX PUF

Different Types of Modified Feed-Forward MUX PUFs

Different Types of Modified Feed-Forward MUX PUFs Modified feed-forward overlap (MFFO) are also known to be Modified Feed-Forward MUX PUFs. The modified feed-forward MUX PUFs can also be classified as, modified feed-forward cascade (MFFC), and modified feed-forward separate (MFFS) , respectively. Interchip and Intrachip behavior are observed in. These three different structures also have different. Additionally, while returning the high security modified feed paths also use in logic reconfigurable feed forward MUX PUF for improving reliability. The structure of feed forward MUX PUF is as shown in fig.12 has two FF arbiter and output is added as a challenge bit to two intermediate MUX stage.

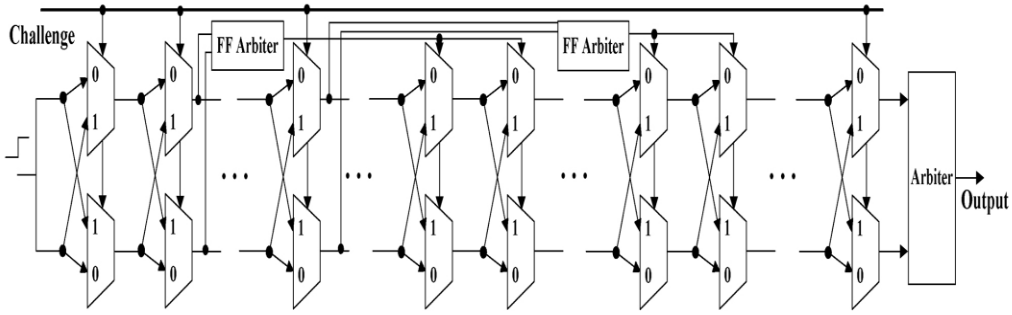


Figure 12: Modified Feed Forward MUX PUF overlap Structure

The figure shows the cascade structure of modified feed-forward MUX PUF. Fig.13 has two FF arbiters whose output is added as challenge bit to two immediate MUX stages.

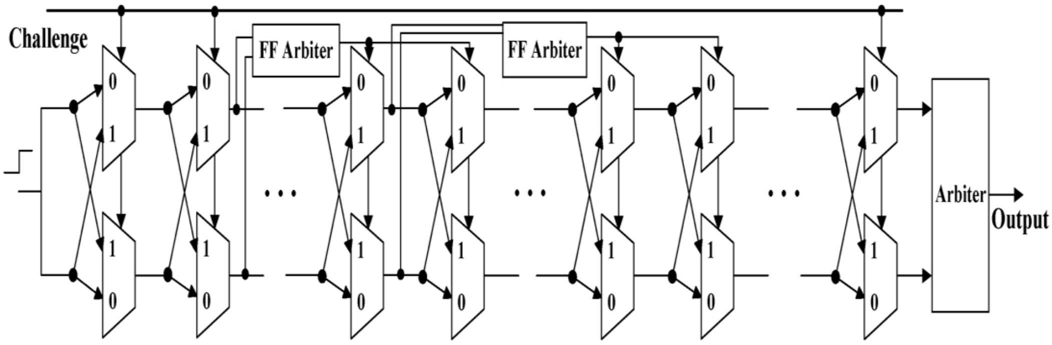


Figure 13: Modified Feed Forward MUX PUF Cascade Structure

The modified feed forward MUX PUF separate structure as shown in figure.14 output is added to two MUX stages where the structure has two separate FF arbiters w.

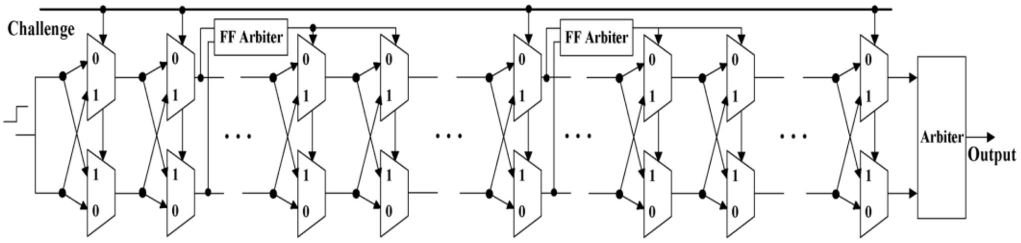


Figure.14: Modified Feed Forward MUX PUF Separate Structure

III. Arbiter Design

Modified feed forward MUX PUF separate structure 3 arbiter design whenever more master requests the bus at the same time. In traditional shared bus architecture, connection of resources happens. For the crossbar and partial crossbar architecture, resource connection when more than one master tries to access slave. The slave IP's are associated with arbiters which can be accessed by the master. For obtaining the prescribed states, a counter is used, which can produce different states to accesses different slaves. These counters are also used as clock pulses and called as counter clock pulses. The sequence of states in a counter may fallow binary count.If a counter follows binary sequence than it is said to be a binary counter. The n-bit counter can count up to 2^n -1 binary counts and consist of n flip flops. Random numbers are much used in creating encryption keys for various applications in communication channels and in security applications. Encodes and decoders are used for communicating in noisy channels. In set of numbers having same probability and uniformly distributed for each number is called as a random number. There are various methods of generating the random patterns like counters and shift registers. These shift registers are very fast generation of binary patterns. For creating truly random patterns, shift registers uses m- sequences are well suited. The minimum length feedback polynomials like 8, 16, and 32 bit LFSR are implemented on FPGA based on PNRG. By changing the feedback polynomial, runtime length and randomness cab changed. For understanding the utilization of memory and speed requirements, 8, 16 and 32 bits are implemented on FPGA by using verilog.

The simulation problem in long bit LFSR is identified and comparisons are made based on simulation and synthesis results. The LFSR is a shift register in which the MSB is the XOR result of any two bits in the sequence which helps on generating all the other patterns. The linear feedback shift register (LFSR) is as shown in the figure.12. LFSR is a shift register whose input is driven by a XOR gate, which helps to produce all the other input combinations i.e patterns. The values in the shift register follows chancing states for producing the other values. An input value is produced called as seed value is given as input and by the XOR operation for this seed value gives the remaining patterns. For checking the transmission errors, the mathematics used is much related to the CRC cyclic redundancy check. In a 16 bit Fibonacci LFSR, the register cycles reaches a maximum number of 65535 states excluding all-zeros state for CRC, are primitive polynomial corresponding to top feedback numbers. In fig (16,14,13,11) are the taps. The rightmost bit is the XOR'd with the output bit and passed the result to the left most bit. For the LFSR there is a possibility of having more than one maximum length value. If the tap sequence, in an n-bit LFSR, is (n, A, B, C, 0), the corresponding 'mirror' sequence is [n, n - C, n - B, n - A, 0] for the 0 corresponds of x0 = 1 term. The counterpart [32, 30, 29, 25, 0] of the tap sequence [32, 7, 3, 2, 0]. Both give a maximum-length sequence. Some example C code is below: shift registers having their previous states as inputs are called as LFSR's. An initial value for the LFSR is known as the seed value . for the next states the values generated by using a logic function that which helps to generating the next states. In LFSR the basic function is chosen as exclusive-or (XOR) logic function which can be simply implemented by using an XOR gate. All the other cycles / states or produced by shifting the bits in the shift register. However with well chosen feedback function the LFSR can produce the patterns that are random to each other implementing LFSE base on the pseudo random number sequence generator. One advantage of LFSR is that it is similar in both hardware and software. FSR produces a maximum length of an m sequence (i.e. it cycles through all possible 2n -1 states within the shift register except the state where all bits are zero), unless it contains all zeros, in which case it will never change.

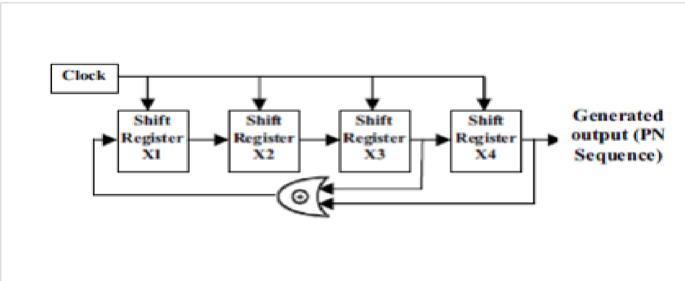


Figure.15: Block diagram of LFSR

The feedback polynomial or characteristic polynomial is the one in which the polynomial must be 1's or 0's for the co-efficients. For example, the feedback polynomial is X32 + X30 + X11 + X5 + X 1. 8-bit LFSR with maximum length feedback polynomial X8 + X6 + X5 + X4 + X1 generates 28 -1 = 255 random outputs for if the taps are at the 32nd, 30th, 11th and 5th bits, then , which is verified from the simulation waveform. The circuit diagram for 8-bit LFSR with maximum length polynomial is shown in Fig. 16.

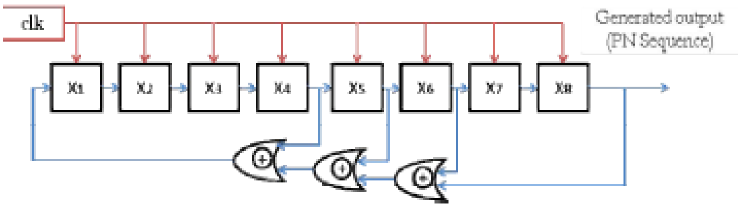


Fig.16: 8- bit LFSR.

The timing simulation is as shown in figure 2.16 40 ns to 5140 ns. Beyond this time period repeating random output is obtained. By avoiding the frequent transiting of logic the logic levels of primary inputs that improves correlation among the successive vectors. We just need two input pins as test enable and clock for activating and generating the patterns as well as for designing the hardware. The XOR operation is based on the polynomial x8+x+1. As in the conventional LFSR, the number of transitions can also assist the fault detections in patterns by the intermediate vectors along with aiding in reducing. The technique for producing a low power patterns for BIST, a low power test pattern generator uses a 9-bit LFSR schematic as shown in figure 4.1. The seed value is assigned as (01001011) for the 9-bit LFSR by using the verilog testbench coding. The output of the first flip-flop is 0 and the output of the 8th flip-flop is 1. The exclusive-or of the 8th-flip-flop (logic 1 in this case) and the first flip-flop(logic 0 in this case) is input (1 EXOR 0 = 1 into the first D flip-flop. LFSR is 1010 for the new pattern in the first four bits. Note along with the first 4 bits of the LFSR the shaded register is clocked. So, output of 4th flip flop is 0 in this case and is driven by the shaded flip flop. The input of shaded flip flop is the seed value for the 4th flip flop, the output of 4th flip flop is 0 in this case. so at the end of first clock 0 will be appear at the end of next Shared flip flop. In other words the 4th output is now stored in shaded register and is used in the next steps.

IV. Simulation Results

Rtl Schematic

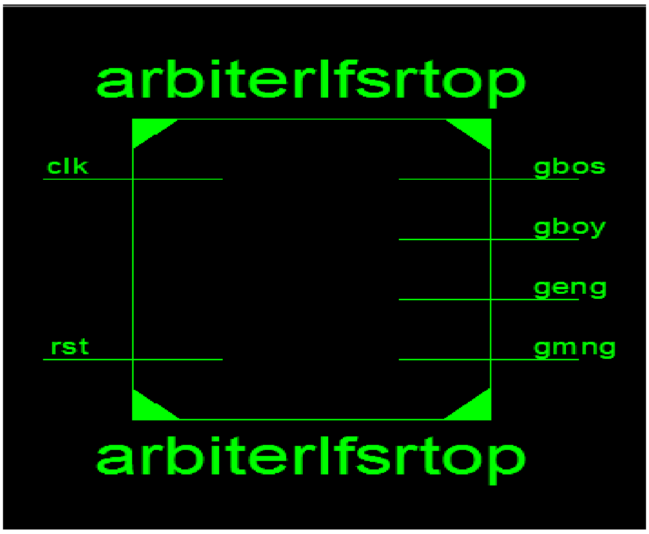


Figure.17 Rtl Schematic

Internal View Of Rtl Schematic

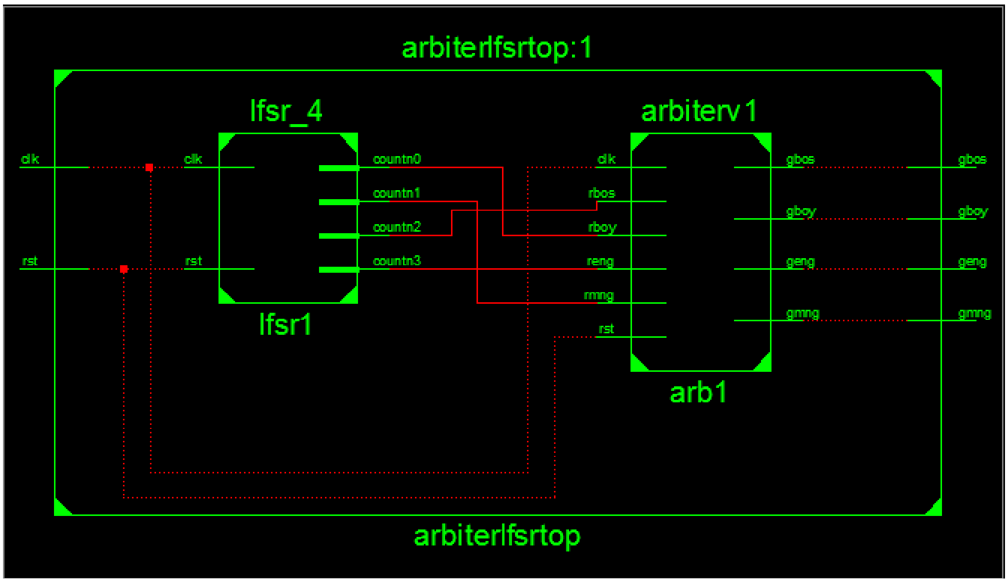


Figure.18 internal view of rtl schematic

Technology Schematic:

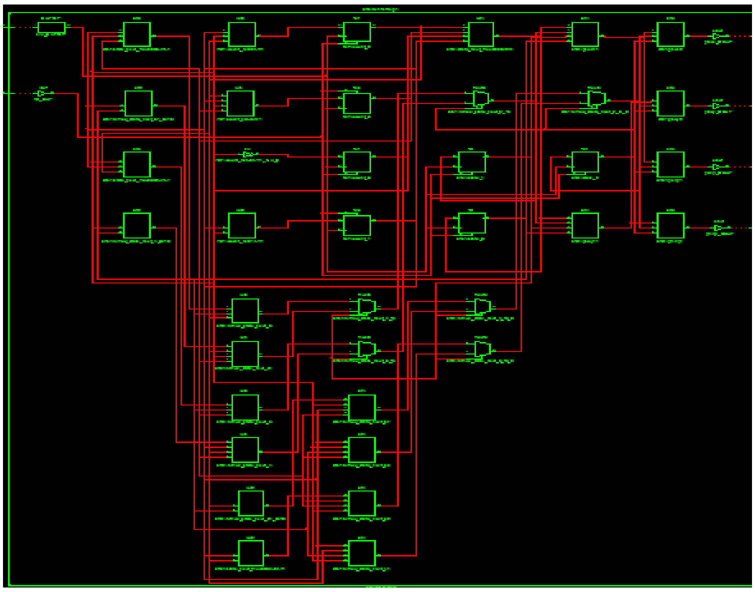


Figure.19: Technology schematic

Comparison Table:

Table: Comparison Table

	NO of 4 input LUT'S	POWER(mw)	DELAY(ns)
EXISTING	25	0.20382	9.400
PROPOSED	22	0.17936	9.370

V. Conclusion

A systematic statistical approach to quantitatively evaluate various types of MUX-based PUFs is presented. Three performance indicators—reliability, uniqueness, and randomness—to compare the performances of these MUX-based PUFs are defined. These indicators are also validated by the corresponding simulation results. The experimental results show that the proposed statistical analysis approach effectively reflects the characteristics of various PUF designs. We have also proposed a novel modified MUX PUF structure using LFSR, which has better reliability than the standard feed forward MUX PUF. Future work will be directed toward the evaluation of MUX-based PUFs from a security perspective by various types of modeling attacks. We also would like to verify our findings with fabricated PUFs in future.By comparing both existing and proposed system of MUX based LFSR, power reduction will be reduced and increase the delay. These experimental results validate the correctness of our statistical analysis. Overall, all the MUX-based PUF structures can be used as reliable secret keys for authentication and identification within certain error tolerance, as the PUFs exhibit sufficient gaps between the minimum of the interchip variations and the maximum of intrachip variations.

References

[1] R.Pappu, B.Recht, J. Taylor, and N.Gershenfeld, "Physical one-way functions," Science, vol. 297, no. 5589, pp. 2026–2030, 2002.

[2] B.Gassend, D.Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in Proc. ACM Conf. Comput. Commun. Security, 2002, pp. 148–160.

[3] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled physical unclonable functions," in Proc. 18th Annu. Comput. Security Appl. Conf., 2002, pp. 149–160.

[4] B. ˇSkori´c, S. Maubach, T. Kevenaar, and P. Tuyls, "Information-theoretic analysis of capacitive physical unclonable functions," J.Appl. Phys., vol. 100, no. 2, p. 024902, 2006.

[5] B. ˇSkori´c, "On the entropy of keys derived from laser speckle; statistical properties of Gabor-transformed speckle," J. Opt. A: Pure Appl. Opt., vol. 10, no. 5, p. 055304, 2008.

[6] A. Maiti, J. Casarona, L. McHale, and P. Schaumont, "A large scale characterization of RO-PUF," in Proc. IEEE Int. Symp. HOST, 2010,pp. 94–99.

[7] R. Maes, P. Tuyls, and I. Verbauwhede, "Statistical analysis of silicon PUF responses for device identification," in Proc. SECSI Workshop, 2008.

[8] Z. C. Jouini, J. Danger, and L. Bossuet, "Performance evaluation of physically unclonable function by delay statistics," in Proc. IEEE 9th Int. NEWCAS, Jun. 2011, pp. 482–485.

[9] Z. Tariguliyev and B. Ors, "Reliability and security of arbiter-based physical unclonable function circuits," Int. J. Commun. Syst., 2012.

[10] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," in Proc. Int. Conf. ReConFig, 2010, pp. 298–303.

[11] I. Kim, A. Maiti, L. Nazhandali, P. Schaumont, V. Vivekraja, and H. Zhang, "From statistics to circuits: Foundations for future physical

[12] unclonable functions," in Towards Hardware-Intrinsic Security. Berlin, Germany: Springer, 2010, pp. 55–78.

[13] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," IEEE Trans. Very Large Scale (VLSI) Syst., vol. 13, no. 10, pp. 1200–1205, Oct. 2005.

[14] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in Proc. CHES, 2007, pp. 10–13.

[15] S. Kumar, J. Guajardo, R. Maesyz, G. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in Proc. HOST, 2008, pp. 67–70.

[16] U. R¨uhrmair, H. Busch, and S. Katzenbeisser, "Strong PUFs: Models, constructions, and security proofs," in Towards Hardware-Intrinsic Security. Berlin, Germany: Springer, 2010, pp. 79–96.

[17] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlation in a pert-like traversal," in Proc. IEEE Int. Conf.Comput.-Aided Design Integr. Circuits Syst., 2003, pp. 621–625.

[18] J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits

[19] with identification and authentication applications," in Proc. IEEE Int. Conf. Computer.-Aided Design Integr. Circuits Syst., 2003, pp. 621–625.

[20] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, and J. Schmidhuber, Modeling attacks on physical unclonable functions," in Proc. Conf. RFID Security, 2010, pp. 237–249.

[21] Y. Lao and K. K. Parhi, "Novel reconfigurable silicon physical unclonable functions," in Proc. Workshop FDSCPS, 2011, pp. 30–36.

[22] Y. Lao and K. K. Parhi, "Reconfigurable architectures for silicon physical unclonable functions," in Proc. IEEE Int. Conf. Electro Inf. Technol., 2011, pp. 1–7.

TextBook

Physically Unclonable Functions: Constructions, Properties and Applications Roelmaes.