Microprocessor Based Physical Unclonable Function

Sudeendra kumar K, Sauvagya Sahoo, Abhishek Mahapatra, Ayas Kanta Swain, K.K.Mahapatra
kumar.sudeendra@gmail.com, sauvagya.nitrkl@gmail.com, kmaha2@gmail.com
National Institute of Technology, Rourkela

Abstract- **Research on Physical Unclonable Functions (PUF) is well established topic in the field of hardware security. PUF is useful in many security applications like IC metering, IP protection and cryptographic key generation. The PUF circuits proposed in the past are dedicated circuits which are extra overhead in terms of area and power. Utilizing the existing circuit structures like microprocessor, power rails, etc to design PUF can be seen in recent literature. In this paper, we propose a PUF topology based on microprocessor and CRP generation method. We present the interim result in terms of hamming distance to prove sufficient randomness in path delays in the hardware multiplier of OpenMSP430 microprocessor which can be exploited to design the PUF. The simulation and statistical analysis technique is also discussed.**

Keywords: **Physical Unclonable function, Microprocessor, Hardware security.**

I. INTRODUCTION

Security has become one of the prime concerns in modern chip design. The major security problems are: - reverse engineering, side-channel attacks, hardware Trojans, IP violations and counterfeiting [1]. Counterfeiting is a major threat, which affects both revenue and reputation of the chip vendor. The chip vendors categorize semiconductor applications as: - Automotive, Consumer electronics, Communication and networking, Biomedical and Industrial control. With the ubiquitous networking happening due to IoT based products and services, the above mentioned applications will undergo significant re-engineering. In the process of this re-engineering, the growth and revenue of chip makers will mainly depend on the success of their products and services in IoT domain. The core tenets in the IoT design are: - agility, scalability, cost and security. The known challenges in an IoT ecosystem are: - Identification or authentication for addressing an IoT node, choosing a right connectivity technique, maintaining data compliance across network and security.

Security is of prime importance, because it is a part of challenge and also it is one of the core tenets in the IoT design. In the list of security issues, the problem of identification and authentication of an IoT node in the network of millions of nodes is very important. The conventional cryptography based authentication techniques may not support millions of nodes and vulnerable to different varieties of side channel attacks. Physical Unclonable Function (PUF) is a promising security primitive, which is widely investigated to solve the security problems in hardware security and IoT domain [2]. PUF circuits are used in hardware metering schemes against IC counterfeiting, chip authentication and in designing IP protection techniques [3] [4]. PUF produces a unique response for an input challenge. The input challenge and corresponding response obtained from PUF circuit is called challenge-response pair (CRP). The large number of CRP's are collected from each chip produced. CRP's are unique to a given chip, which can be used for identification of chip and also further can be used in the identification of IoT node, in which chip is used. The unique CRP for a chip is derived from the process variation that occurs during chip fabrication. With complete knowledge of PUF circuit, it is impossible to manufacture an identical circuit with same CRP's. The uniqueness in CRP's for every chip produced comes from process variation, which is hidden and distinct. PUF circuits are helpful in solving the security issues in hardware security and cryptography [2].

The circuits which pick up maximum process variation are selected to design the dedicated PUF circuits. Ring oscillator (RO) based PUF circuits are discussed widely in the literature [2]. Arbiters, SRAM cells and flip-flops based PUF circuits are found in literature [2] [5]. A majority of PUF circuits are dedicated structures, which occupy reasonably good amount of area on silicon and affect the power budget. And also, integration of PUF circuits into chip needs extra effort in terms of placement and routing. Some PUF implementations are proposed in the past using on-chip modules which are primarily not designed to use as PUF [6]. Typical examples are microprocessor PUF in [7]. Microprocessor is used as PUF in a separate mode called PUF-mode. In the similar way, there are several PUF implementations use SRAM cells, Scan or Design for Testability (DFT) structures [6], power rails [8], clock networks [9] etc. In this paper, we focus on: -

- Survey and comparative analysis of all PUF implementations which use existing on-chip structures like processor, memory, DFT etc.
- Analysis of different microprocessor based PUF implementations.
- Novel technique to design a PUF using microprocessors.

The organization of the paper is as follows: - Section II discusses background and motivation to design intrinsic PUF using available on-chip components or sub-modules. Section III introduces the openMSP430 microprocessor. Section IV presents a novel technique to use microprocessor as PUF. Finally section V concludes the paper.

II. BACKGROUND

PUF circuits found in research literature are implemented need dedicated structures which tax the silicon area and power consumption. In design effort, special care is required in placement and routing of dedicated PUF structures. Dedicated PUF structures may also increase number of I/O pins, which is a crucial resource. So using the existing modules, which are primarily required for functionality or testing of the chip can be used as PUF circuits. This separate class of PUF circuits has advantage in terms of silicon area and power consumption. Most of the PUF circuits are based on delay. Ring Oscillator PUF (RO-PUF) [5] is a common delay PUF produces CRP's by comparing the frequency of two ring oscillators. The widely discussed PUF structure after RO-PUF is Arbiter PUF. Arbiter PUF derives its randomness from the delay of two paths in cascaded switches, which generates unique CRPs [2] [5]. Both RO-PUF and Arbiter PUFs are dedicated structures. The PUF structures which are capable of producing large number of CRPs are known as Strong PUFs. The PUF with small number of CRPs are called Weak PUFs [2].

The quality of PUF is decided by security properties: - uniqueness, uniformity and reliability. Uniqueness is an ability of PUF to generate a unique response in a specific chip among the group of chips of same type for same challenge. Uniqueness is measured using hamming distance. Inter-chip variation of PUF response is measured using uniqueness. An ideal value of uniqueness is 50%. Reliability of PUF is ability to generate the same response, when same stimulus is applied repeatedly. The ideal value of reliability is 100%. Reliability of PUF CRP's is affected by temperature, supply voltages and aging. Uniformity of the PUF is an estimation of the proportion of 1's and 0's in the response of PUF. The ideal value of uniformity is 50%. Uniformity is calculated using percentage of the hamming weight of the response. The details on calculation of security properties can be found in [2] [5].

Motivation to Build PUF based on microprocessor: - Microprocessor is a versatile module in modern day System on Chip (SoC) devices. When microprocessor is not used as PUF circuit, it can be used several other purposes like any other processor. The delays in microprocessor can be exploited to design an efficient PUF circuit. There are several sections of microprocessor in which delays are sensitive to process variation and randomness in delays can be used to generate the quality CRPs. Few examples of such sensitive sections are: - memory to processor interfacing, ALU of the processor (mainly combinational logic), MAC unit (Multiply and Accumulate unit) and memory management. Delays in microprocessor sections are sensitive to intra-die variations, which is helpful in generating the quality CRPs. The microprocessor is a large circuit and strong PUF can be designed.

Comparative Analysis of Different PUF Implementations using on-chip structures: - Several PUF implementations using on-chip structures can found in research literature. This class of PUF implementations can be classified as: -

- Microprocessor based PUF implementations [7] [10],
- Test structures based PUF implementations [6] [11],
- Memory/SRAM based PUF's [12] [13],
- Other delay based PUF implementations [14].

Microprocessor PUF: - A.Maiti et al, in [7], propose microprocessor PUF, which accepts assembly program as challenge and produce response based on the delay in the data path or control path. The delay value is captured by over-clocking the microprocessor operation. The instructions are characterized as input challenge to PUF, fail when clock frequency is increased. Based on the pass count between the passing frequency and failing frequency, response is generated. This microprocessor PUF is verified on 32-bit LEON3 processor using the SPARC instruction set on Xilinx Spartan 3E FPGA. The uniqueness of this PUF is of acceptable quality and very good reliability. J.Kong et al propose a PUF, by adding the arbiter circuits at the output of ALU of processor [10]. The randomness in ALU and arbiter circuit is used to generate the CRPs. Authors of [10], also proposes an algorithm which leverages aging phenomenon in chips to improve the inter-chip and intra-chip variation, which leads to better quality of CRPs. The proposed PUF and algorithm is verified in 45 nm technology.

Test structures based PUF: - B. Niewenhuis proposed PUF [15] based on the power-up states of scan chains. Scan chains are inserted into chip to make the post fabrication tests easy and cost effective. The randomness required for PUF is derived from the scan chain power up states. This PUF implementation is verified in 65nm CMOS process. Y.Zheng in [11], propose a PUF by exploiting the path delay variation of scan flip-flops to generate the CRPs. The scan chains which are spread across the chip provided the large pool of scan paths to create large number of CRPs. Circuit simulation results for 1000 chips show high uniqueness of 49% at room temperature and this PUF is sensitive to environmental variations like temperature etc. PUF is also validated on FPGA and results at room temperature are presented. Y.Zheng in [6] proposes an extension to [6] called DScanPUF. DScanPUF is based on delay measurement structure consisting of PLL and multiple clock delay lines to measure the delays in scan path. Based on the delays, CRPs are generated. DScanPUF is validated on 31 FPGA chips and results show good security properties and circuit simulations are performed at 45nm CMOS process. F.Saqib in [16], presents the PUF design based on the embedded test structure called REBEL. REBEL is similar to logic analyser to perform the analysis of temporal behavior of signals from emerging paths of the logic core. REBEL based PUF is called as HELP (Hardware Embedded deLay PUF), which extracts the randomness from the path delays of the logic core. The security properties of HELP are evaluated across temperature and supply voltage variations.

Memory/SRAM based PUF: - PUF's are designed using randomness derived from the inner node voltages of storage circuits like SRAM and flip-flop. RESP (Retrofitted Embedded SRAM PUF) [13] utilizes the voltage scaling induced access failures in SRAM array to generate the

challenge-response pairs. RESP extracts the randomness from write access failures under scaled supply voltage from a set of SRAM cells. The access failures occur in only few cells, due to device level process variation. After writing initial values into the SRAM cell at the scaled supply, the content of SRAM cells is read out to create the CRPs of that chip. Initial values and voltage levels will act like challenges, which make large number of challenges that can be fed into SRAM to generate unique responses for each chip produced. The MECCA (Memory Cell based Chip Authentication) PUF proposed in [12] leverages the randomness from the read/write access failures based on the word line duty cycle of the SRAM cells. The large number of challenge response pairs can be generated using word line controllability. The MECCA PUF contains a SRAM array and programmable delay generator. The input challenge decides the number of SRAM cells and selected cells are written with logic '0' or logic '1' into the cells with standard write duty cycle. The word line duty cycle is reduced using programmable delay circuit. The write operation is performed with reduced duty cycle and finally values are read out from the cells to create the response equal to number of cells. MECCA PUF is simulated for 1000 chips with 10% inter-die variations. The simulation results show the high uniqueness with 49.9% and good reproducibility.

Other delay based PUF: - Suzuki proposes delay based PUF called Glitch PUF [14]. Glitch PUF exploits glitches from delay variations between the gates and pulse propagation of each gate. The Glitch PUF behaves differently on each individual chip due to process variation. A random challenge is fed into the logic core and glitch wave forms are acquired and converted to response bits. Any reasonable size digital logic core can be used as Glitch PUF. In [14], AES (Advanced Encryption Standard) is used to generate the glitches. The security properties of glitch PUF are comparable with other standard PUF circuits.

The comparison of all the intrinsic PUF's discussed above is analysed in Table I. The observations are: -

- Except SRAM PUF, all other PUF topologies are based on delay.
- Accurate delay measurement and conversion of delay into response bits is crucial.
- The PVT performance of delay circuit or voltage regulator (RESP PUF) is crucial for the performance of the PUF.
- In most of the intrinsic PUFs, the CRP collection is complex. The exhaustive characterization of both PUF and additional circuit added to handle challenge creation is required before enrolment.
- The quality of PUF and number of CRPs depends upon the size of the circuit. The higher the size of the circuit, more CRPs can be generated in delay PUF.
- Generally, SRAM based PUF, are considered as weak PUF. The voltage and temperature variations will affect the SRAM PUF CRP reliability.
- The quality of test structure based PUF depends upon the number of scan chains in the design. Careful PUF characterization is required before enrolment of CRPs, to get reliable CRPs. With less number of scan chains (in smaller designs) the PUF will suffer from low intra-die and inter-die variations.

Based on the above observations, size of circuit is important to get the quality PUF with large number of CRP. An 8/16-bit microprocessors makes a good candidate for designing PUF. Microprocessor based PUF implementations are not explored well. The two processor based PUF implementations methods are discussed below: -

- In microprocessor PUF described in [7], delay variability in data path and control path is used to generate the PUF response. Depending on the instruction running on the processor, particular set of data path and control path are activated. The paths will have variable delay for every chip produced. The path delay depends upon the instruction and its operands being executed. Increase in clock frequency leads to setup and hold violations. This will make the instruction fail. Each instruction with different operands will fail at different frequencies. The point of failure is called frequency failure point (FFP). Before reaching the point of failure, instruction will fail partially at different frequency points. Between the full successful execution of instruction and complete failure point, the instruction will fail partially, when over-clocked. Setup and hold a failure due to over-clocking occurs through a region and instruction does not fail at one point. The instruction is executed for 'n' number of times at different frequencies and number of successful executions of instruction is recorded in the region. The number of successful executions is called pass count (pc). Based on pc, the response is generated.
- An important PUF topology is Glitch PUF in [14]. Glitch PUF is carefully designed using extensive simulation studies on delay. The numerous SDF files (Standard Delay Format) are created with small variations in delay parameters, which reflect the randomness in the process variations for simulations. Based on the accurate simulations, occurrence of glitches is determined and challenges are designed accordingly. Glitches are captured and converted into response bits.

The concepts used in microprocessor PUF in [7] and glitch PUF [14] can be further extended to build better PUF circuits. An exhaustive path delay analysis of the complete microprocessor through statistical static timing analysis (SSTA) and designing PUF circuit based on the path delays is not yet explored. A complete timing analysis from fetching the instruction and operands from memory to final execution and its modelling is important in designing PUF. In this paper, we perform the SSTA on hardware multiplier of OpenMSP 430 microprocessor and using the path delays of the multiplier, new PUF is proposed.

TABLE I
COMPARISON OF DIFFERENT INTRINSIC PHYSICAL UNCLONABLE FUNCTIONS

Intrinsic PUF Implementation	Validation	Need for additional circuit	Complexity in operation	Security Properties	Challenges with this technique
Microprocessor PUF [7]	FPGA	On board precise high speed pulse generator	Characterising the each instruction of microprocessor for different levels of over-clocking makes the input challenge design difficult.	Each instruction will have separate security values. Reliability is better than uniqueness. Reliability =98% and Uniqueness =38%	Choosing a right instruction to use as input is a challenge. Calibration is required for response collection.
Processor PUF [10]	Circuit Simulation (ASIC)	Arbiter at the output of ALU of the processor.	This paper does not mention the suitable method for CRP collection.	Uniqueness = 43% and Reliability = 98%. Properties are validated on circuit simulations based on Hamming Distance (HD).	Choosing the aging based input vectors for challenge to PUF is challenge. Post silicon characterization is also complex.
Scan PUF [11]	FPGA and Circuit Simulation (ASIC)	NA	Relatively simple operation.	Comparable with standard PUF implementations.	Low Inter-die and Intra die variations.
DScan PUF [6]	FPGA and Circuit simulation	On-chip precise programmable PLL/Clock generator required	CRP collection is complex. Responses are first stored in memory and read out. Challenge design is based on PLL characterization.	Uniqueness is high (49%) and Reliability is good for voltage fluctuations and temperature.	Challenge design is not clear. Location of scan path is part of challenge. Careful challenge design based on timing analysis data is followed.
Hardware Embedded Delay PUF (HELP) [16]	FPGA	On-chip precise programmable PLL/Clock generator required	CRP collection is complex. CRP quality will vary on the performance of Data Collection Engine (DCE). The DCE characterization is required prior CRP collection.	Hamming distance is presented in paper. Security properties are not explicitly discussed.	Sufficiently large logic core is required for better PUF or strong PUF implementation
RESP (Retrofitted Embedded SRAM PUF) [13]	ASIC	Programmable reference generator and voltage regulator	Input challenges depend upon the precision of voltage generator and regulator.	Uniqueness =49% for standard voltage and uniqueness reduces when operating voltage is decreased.	Low Inter-die and Intra die variations. Using only voltage as challenge will lead to this problem. Resolution of voltage generator is crucial.
MECCA (Memory Cell based Chip Authentication) PUF [12]	ASIC	Programmable delay generator	PVT analysis of delay generator is crucial	Uniqueness is 49%. Reliability at normal operating conditions is 87%.	Voltage variations lead to unreliable CRPs.
Glitch PUF [14]	FPGA	Glitch generator Delay circuit and Memory.	Careful design and analysis of glitch acquisition module is crucial.	Hamming distance is presented in paper. Security properties are not explicitly discussed	Low inter-chip and intra chip variations. Jitter corrections and shape judgements. Size of logic core should be reasonably large.

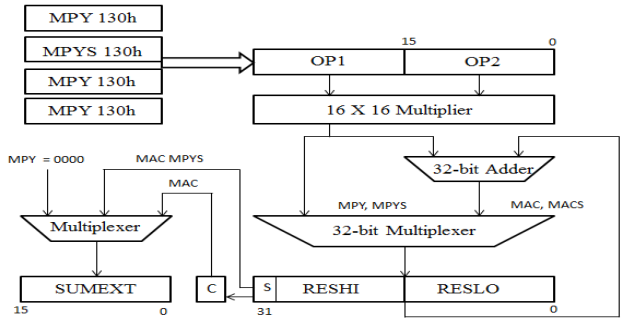

Fig. 1.OpenMSP430 Microprocessor Hardware Multiplier

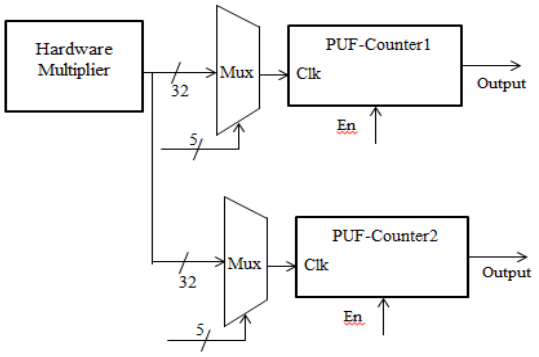

Fig. 2.Proposed Microprocessor PUF

III. OPENMSP430 MICROPROCESSOR

The openMSP430 is instruction cloned design which is compatible with Texas Instruments MSP 430 microcontroller. It is Von Neumann architecture with single address space for both instruction and data. The complete design and gcc tools (compiler and debugger) are available in open source [17]. Design is silicon proven multiple times and it is suitable for both FPGA and ASIC implementation. The microprocessor core is fully compatible to connect memory cores and RAM/ROM of any size up to 32Kb. The components of microprocessor are: - ALU, serial debug interface, Memory backbone for memory management, clock module, SFR, Watchdog and 16X16 hardware multiplier. All features of original MSP 430 are implemented in hardware multiplier of OpenMSP430. We use hardware multiplier to build the PUF circuit.

Hardware multiplier: - Hardware multiplier is a peripheral and does not interfere with CPU activities. The multiplier registers supports unsigned and signed multiplication, unsigned and signed multiply accumulate. The block diagram of hardware multiplier is shown in figure 1. Hardware multiplier has got two 16-bit operand registers: - OP1 and OP2 and three register to store result: - RESLO, RESHI and SUMEXT. RESLO stores lower word and RESHI stores the high word of the result and SUMEXT stores information about the result. The operand OP1 has four addresses, used to select the different modes of multiplication. Writing the first operand to address will select the type of multiply operation. Writing the second operand into appropriate register initiates the multiplication. Multiplication is performed on the values stored in OP1 and OP2 and result is stored in RESLO, RESHI and SUMEXT. Repeated multiplication operations can be performed without reloading OP1, if the value in OP1 is used for successive multiplications. More details on hardware multiplier can be found in datasheet of MSP 430 [18].

IV. PROPOSED MICROPROCESSOR PUF

The block diagram of proposed microprocessor PUF is shown in figure 2. The proposed PUF uses the hardware multiplier, two 32-bit multiplexers, two 32-bit PUF-counters. The output of hardware multiplier block consists of 32-bits, which is connected to multiplexer. The multiplexer chooses the any one output of hardware multiplier block of microprocessor to drive the PUF-counter. The output of hardware multiplier is connected to the PUF-counter. The size of PUF-counter is 32-bits. The central portion of the PUF-counters output from 9^{th} bit to 24^{th} bit is stored in general purpose general purpose register of the microprocessor and taken out at the output ports. The 32-bit response is PUF response is generated out of two PUF counters.

Challenge Design and Response collection in the proposed PUF: - The each multiplexer has got 5 select lines to choose the output of hardware multiplier. The input to the select lines makes a part of the challenge to PUF circuit. The other portion of the challenge is assembly program which triggers the different paths of the hardware multiplier. The response generated from the counter are stored in general purpose registers of the microprocessor for further processing. The development of assembly program as a challenge to PUF is crucial for CRP generation. The randomness in the process variation reflected in interconnect delays in data path of hardware multiplier will be different for every chip fabricated. The Multiply-Accumulate instruction mode (MPY and MPYS mode) has got the inherent feedback in the functionality. Using MPY mode, data can be looped several times. This feature is used effectively in challenge design to PUF.

Implementation and Results: - The openmsp430 design from opencores is modified to operate in two modes: - normal mode and PUF mode. The 10 select lines and 'En" pin are added to the input/output pins of the microprocessor design. The hardware multiplier output section is modified to add the multiplexer and PUF-counter. The additional circuits are active only during PUF mode and in normal mode circuits are detached from the output of hardware multiplier.

Statistical STA of hardware multiplier: - Static Timing Analysis (STA) is deterministic and analysis is based on fixed delays for all timing arcs in the design. Statistical modeling of variations of all parameters like cell timing models, interconnect parasitics is required to perform Statistical Static Timing Analysis (SSTA). This means that timing models should be described in terms of mean and standard deviation for both global and local parameters. The interconnect parasitics described in terms of mean and standard deviation is used in delay calculations. Every delay is represented by a mean and standard deviation. SSTA process combines the delays of timing arcs to calculate the path delay, which is also in terms of mean and standard deviation. SSTA maps the standard deviation with respect to independent parameters (both process and interconnect related) to calculate the overall standard deviation of path delay. The slack is also obtained as statistical variable with its nominal value and standard deviation. Timing windows for noise and crosstalk are also modelled statistically and added to SSTA. Based on the path slack distribution, SSTA calculate and report the all relevant statistical parameters of slack for every path of interest [19].

A complete openMSP430 microprocessor is simulated using assembly programs in Cadence NcSim. Synopsys Design Compiler is used to synthesize the microprocessor using TSMC65nm library. Synopsys PrimeTime is used for STA. Cadence Encounter is used for place and route the design.

The Hamming distance analysis is performed on the bits generated for a given challenge from the output of hardware multiplier, to ensure the randomness in the output of hardware multiplier. The procedure is based on the statistical analysis of path delays: -

- The VCD (Value Change Dump) files for gate level netlist of the design are captured for different challenges (assembly programs) during gate level simulations.

- Post layout parasitics (SPEF) files are extracted from place and route. Both VCD and SPEF files are used in Synopsys PrimeTime for STA.
- Design is synthesized at 50MHz and STA is performed. Further, SSTA is performed and mean and standard deviation of delays in different paths of hardware multiplier is studied. Based on this, effect of global and local variations on delays is analysed. Based on the statistical data, the timing constraints file (SDC and SDF) used in synthesis are changed accordingly to reflect the actual delays of chips after fabrication. The timing constraints are varied in SDC and SDF between -3σ to +3σ from the standard values. In this experiment, 64 different SDC files are created and different gate level netlist is created with each SDC. The STA analysis is performed on each gate level netlist to calculate path delays in hardware multiplier.
- Path delay values are collected from Primetime is processed to convert them into bits. The delay value of each path is compared with standard value (50 MHz delay values) to generate the bits. The hamming distance performed for all 64 instances (64 different SDC used for synthesis) with standard 50 MHz SDC. The average hamming distance is 42%.

This confirms the randomness in the hardware multiplier can be used to design PUF and output of hardware multiplier can be used to drive the PUF-counter to get the more stable response.

V. Conclusion and Future Work

In this paper, we present the microprocessor based intrinsic PUF. We have discussed intrinsic PUF topologies and comparative analysis is also presented. The inherent path delays in the large combinational circuits in the microprocessor can be exploited to design PUF. Statistical STA of Path delays in hardware multiplier in openmsp430 microprocessor used in this experiment prove that sufficient random variation in path delays exists. The interim results are presented in this paper and our future work is: -

- Analyse the microprocessor (openmsp430) completely through statistical STA for designing better quality PUF circuit.
- Characterization of instruction set for better challenge design
- FPGA implementation and large scale CRP collection.
- Analysis of PUF quality using security metrics like uniqueness, uniformity and reliability.

References

[1] M. Tehranipoor and C. Wang, "*Introduction to Hardware Security and Trust*", Newyork, NY, USA; Springer-2011.

[2] C.Herder et al, "Physical Unclonable Functions and Applications: A tutorial", Proceedings of the IEEE, Volume: 102, Issue: 8, Aug. 2014.

[3] Abhishek Basak, et al, "Security Assurance for System-on-Chip Designs With Untrusted IPs", IEEE Trans. Information Forensics and Security, 1515-1528, June-2017.

[4] X. Wang, Y. Zheng, A. Basak, and S. Bhunia, "IIPS: Infrastructure IP for Secure SoC Design," IEEE Transaction on Computers, 2014.

[5] C.H.Chang et al, "A Retrospective and a Look Forward: Fifteen Years of Physical Unclonable Function Advancement", IEEE Circuits and Systems Magazine, Volume 17, Issue-3, 2017.

[6] Y.Zheng, et al, "DScanPUF: A Delay-Based Physical Unclonable Function Built Into Scan Chain", **:** IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 24, Issue: 3, March 2016.

[7] Abhranil Maiti, Patrick Schaumont, A novel microprocessor-intrinsic Physical Unclonable Function, 22nd International Conference on Field Programmable Logic and Applications (FPL), 2012.

[8] Helinski, "A Physical Unclonable Function Derived from the Power Distribution System of an Integrated Circuit", Ph. D dissertation submitted to University of New Mexico, December-2010.

[9] Y. Yao et al, "ClockPUF: Physical Unclonable Functions based on clock networks", Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013.

[10] J.Kong, "Processor-Based Strong Physical Unclonable Functions With Aging-Based Response Tuning" IEEE Transactions on Emerging Topics in Computing, Volume: 2, Issue: 1, March 2014.

[11] Yu Zheng et al, "ScanPUF: Robust ultralow-overhead PUF using scan chain" 18th Asia and South Pacific Design Automation Conference (ASP-DAC), 2013.

[12] Aswin R Krishna et al, "MECCA: a robust low-overhead PUF using embedded memory array" 13th International conference on Cryptographic hardware and embedded systems (CHES-2011).

[13] Yu Zheng et al, "RESP: A robust Physical Unclonable Function retrofitted into embedded SRAM array" 50th ACM/EDAC/IEEE Design Automation Conference, 2013.

[14] D.Suzuki, "The glitch PUF: a new delay-PUF architecture exploiting glitch shapes", 12th International conference on Cryptographic Hardware and Embedded Systems (CHES'10), 2010.

[15] B. Niewenhuis et al, "SCAN-PUF: A low overhead Physically Unclonable Function from scan chain power-up states", International Test Conference, 2013.

[16] J.Aarestad et al, "HELP: A Hardware-Embedded Delay PUF", IEEE Design & Test, Volume: 30, Issue: 2, April 2013.

[17] www.opencores.org.

[18] http://www.ti.com/lit/ds/symlink/msp430g2253.pdf.

[19] J.Bhasker, Static Timing Analysis, http://www.springer.com/in/book/9780387938196 Springer-2010.