Noname manuscript No.
(will be inserted by the editor)

Novel Randomized Placement For FPGA Based Robust ROPUF with Improved Uniqueness

Arjun Singh Chauhan · Vineet Sahula · Atanendu Sekhar Mandal

Received: date / Accepted: date

Abstract The physical unclonable functions (PUF) are used to provide software as well as hardware security for the cyber-physical systems. They have been used for performing significant cryptography tasks such as generating keys, device authentication, securing against IP piracy, and to produce the root of trust as well. However, they lack in reliability metric. We present a novel approach for improving the reliability as well as the uniqueness of the field programmable gated arrays (FPGAs) based ring oscillator PUF and derive a random number, consuming very small area ($< 1\%$) concerning look-up tables (LUTs). We use frequency profiling method for distributing frequency variations in ring oscillators (RO), spatially placed all across the FPGA floor. We are able to spot suitable locations for RO mapping, which leads to enhanced ROPUF reliability. We have evaluated the proposed methodology on *Xilinx -7* series FPGAs and tested the robustness against environmental variations, e.g. temperature and supply voltage variations, simultaneously. The proposed approach achieves significant improvement (i) in uniqueness value upto 49.90%, within 0.1% of the theoretical value (ii) in the reliability value upto 99.70%, which signifies that less than 1 bit flipping has been observed on average, and (iii) in randomness, signified by passing NIST test suite. The response generated through the ROPUF passes all the applicable relevant tests of NIST uniformity statistical test suite.

Malaviya National Institute of Technology
JLN Marg, Jaipur, Rajasthan, India
Tel.: +91-9785324861
E-mail: 2015rec9055@mnit.ac.in

Malaviya National Institute of Technology
JLN Marg, Jaipur, Rajasthan, India
Tel.: 0141-2713336
E-mail: sahula@ieee.org

Cognitive Computing Group, CEERI
Pilani, Rajasthan, India
E-mail: atanu@ceeri.ernet.in

Keywords Hardware Security, PUF, Ring Oscillator, FPGA, Random Number Generator, Biased & Randomized Placement, NIST Statistical Test, K-means Clustering.

1 Introduction

Nowadays, information security, integrity and privacy are the primary attention for each software and hardware subsystems, since the compromised subsystems introduce vulnerability by getting exploited due to various loopholes. Most of these hardware subsystems have been designed by appropriating application specific integrated circuits (ASIC) and field programmable gated arrays (FPGA). FPGA based security solutions are distinguished due to reconfigurable architectures. To secure the information, cryptographic primitives have been used to advance an essential level of information security and integrity. The primary requirement for the cryptographic primitive is to protect the secret keys against various security threats, i.e. key stealing, random guessing, etc. The earlier approaches to protect secret keys against random guessing attacks attempt to increase the entropy of secret key and keep it high enough to pass randomness benchmarks. These random keys can be stored in secure non-volatile memory (NVM) of the hardware device, and appropriated during cryptographic operations.

Nevertheless, these keys can not resist against physical attacks, i.e. side channel parameter extraction, device tempering, etc [12] [13]. These attacks can extract at least partial information of the secret key and predict the remnant. However, hardware-enabled random number generation (RNG) methods provide a suitable alternative, instead of storing it. Moreover, physical unclonable function (PUF) has been utilized to produce random information from the hardware, unlike NVM based keys.

The physical unclonable function has been used to generate a random unique identification sequence by employing stochastic process variability of the silicon-based chips. An electronic circuit is used to produce a n bit binary unique identification sequence, each from a device. These identification sequences are appropriated to provide an essential level of hardware security against IC overbuilding, IP piracy and are also used to protect against device tempering in hardware, i.e. ASIC and FPGA [38] [25]; moreover, the same can also be used for hardware-based cryptographic keys generation [43], hardware device authentication [17] [36] and hardware-based random number generation [3]. Numerous methods have been proposed to produce hardware based random number generations, i.e. linear feedback shift registers [29], power supply based RNG [45], ring oscillator based RNG [15], etc.

FPGA based security solutions are quite prominent and widely adopted due to reconfigurability. Therefore, lightweight and secure cryptographic algorithms have been implemented on FPGA, and employed for the IoT security [24].

The work related to unclonable function has been started with physical one-way function. The one-way function has been used to produce random optical patterns by appropriating the optical characteristics of a material. These optical characteristics have an approximately zero probability for regeneration, without knowing the exact angle information [36]. The silicon-based physical unclonable function has been introduced to produce random information by employing a digital circuit, which is termed as arbiter based PUF design [17] [11]. The arbiter PUF design performs a delay comparison between two symmetrically routed path, where each path consists of the equal number of logic gates. The design exploits the random manufacturing variations and derives a random signature from it. There are many more PUF designs have been proposed to produce random binary sequences, i.e. ring oscillator [43], latch and flip-flops enabled [42] [51] [28], static random access memory (SRAM) [20], etc.

PUF can be categorized in mainly two types on the basis of challenge-response pairs (CRP); 1) Strong, and 2) Weak. A weak PUF typically consists of less number of CRP. However, the strong PUF has many number of CRP. The typical application of the strong PUF is the device authentication, where a large number of CRP requires. The example of strong PUF is arbiter based, optical based, etc. The weak PUF typically used for cryptographic key generation, i.e. ring oscillator PUF (ROPUF), Latch enabled, SRAM based, etc.

The ring oscillator based PUF has been designed by employing multiple ring oscillator, and each ring oscillator produces different frequency due to its manufacturing variations. These frequencies have been used to generate random unique identification sequences. ROPUF design is less affected by routing skew as compared to arbiter based design. However, ring oscillators are more prone to uncontrolled environmental noises, i.e. voltage (V), temperature (T) and device aging (λ) [4]. These noises revise the internal propagation delay of ring oscillators; consequently, the frequency variation appears to be significant, which causes unreliable response bit generation or response bit flipping.

The security of the cryptographic algorithms is dependent on the secure key, and the keys should be reliable and reproducible at different environmental conditions. Therefore unreliable cryptographic key can corrupt the entire message after encryption, which is hard to decrypt at the other end, e.g. secure hash algorithm (SHA) can change 50% of message bits with a single bit flipping in secret key. Similarly, high bit flipping rate can increase the false acceptance rate (FAR) and false rejection rate (FRR) during device authentication application. Therefore reliable response generation through PUF is a relevant and essential problem to be solved. However, the uniqueness enhancement can reduce the FAR and FRR, although randomness can improve the unpredictability of the secure key.

2 Related Work

Author Devadas et al. have started extensive work related to ring oscillator PUF [43]. After that, various researchers have proposed numerous numbers of research articles to address and solve the ring oscillator PUF related problems. The response produced by the ROPUF deficits with unreliable response bits. The increment in unreliable response bits leads to produce misidentification errors, therefore inter-PUF and intra-PUF hamming distances should keep as large as possible. There are many approaches has been proposed to improve the frequency response of the ring oscillator PUF and the improvement in frequency response can improve both distances.

Devadas et al. have introduced the very first ROPUF reliable design, and the approach provides a significant improvement in the frequency response. The author improves the ring oscillator selection logic based on the frequency difference from a group of k ring oscillators. A pair of ring oscillator has been selected out of k ring oscillators to maximize the frequency difference. The approach provides a significant improvement in reliability, but the area utilization has been increased by k times.

The other significant work relating to reliability improvement is the configurable ring oscillator (CRO), which was proposed by Maiti et al. [34]. The author proposed a new design consists of configurable ring oscillators. These ring oscillators are configured in such a way that frequency difference between selected ring oscillator should be maximized. The approach improves the reliability at $90nm$ technology,

and the reliability is $> 99.2\%$ in the presence of environmental variations. There are other implementations of the CRO have been proposed to address the reliability and uniqueness metric issues [55] [41].

There is a remarkable contribution made to correct bit error rate (BER) by introducing post-processing techniques, some of these are the error correction techniques, i.e. temporal majority voting (TMV), Bose-Chaudhuri- Hocquenghem (BCH) codes, etc [6] [7]. There are other techniques such as fuzzy extraction, which has been used to provide strength to the device authentication in the presence of environmental noise [23] [31]. These approaches require extra resources on the hardware and also requires spare computation time for processing.

The FPGA based ROPUF design has limited access to reduce the effect of environmental noise, therefore some authors proposed LUT level modification to enhanced the ROPUF performance. The author suggested a LUT based self compare structure, which can generate 256 bit response by tuning the LUT delay lines [16]. The approach provides an improvement in uniqueness, but the fine-tuning of delay line originates unreliable response bits. Moreover, the author proposed an adaptive tuning circuit, which can reduce unreliable response bits, but the tuning design itself requires extra hardware on FPGA. The author Wei Yan et al. proposed an approach based on phase calibrated ring oscillators, the phase calibration technique has been employed to measure fast and accurately RO frequencies. The response generated through phase calibrated ROPUF design has passed all NIST statistical tests along with very less unreliable bits $< 1\%$ in the presence of the temporal environmental variations [52].

The ROPUF reliability enhancement has been accomplished by increasing the frequency difference. Apart from the CRO and 1-out-of-k approach, there are numerous methods have been proposed to improve the ROPUF reliability. These approaches extract the ring oscillator frequency variation using on-chip frequency monitors and choose a pair of ring oscillators, which provides a significantly large frequency difference. The logic for ring oscillator selection and the frequency difference enhancement are different for each approach. The authors have proposed methods related to temperature awareness frequency measurement [10] [44]. These approaches prefer a ring oscillator pair which has better temperature stability. Moreover, the reliability improvement is significant, but the approach is limited to temperature variations, and it also requires some mechanism to perform temperature characterization of ring oscillators. The authors F. Kodýtek et al. present an approach to improve the randomness characteristics using the entropy of the counter bit position, while counter is subjected to the ring oscillator frequency. The method possess significant randomness and passes NIST statistical methods, however the approach shows high bit flipping in the presence of voltage variation [26].

The other works are related to grouping based approach, where a group or specific order has chosen with several frequencies, and these frequencies are selected in such a manner that the frequency difference has been improved [27] [54]. The author defines a threshold frequency (f_{th}), which is the safe limit for frequency difference and the approach search for the ring oscillators, which provides at least f_{th} frequency difference. The selection of f_{th} is uncertain, and it depends on the characterization accuracy, environmental variations and device family.

The characterization phase for all of these approaches is almost the same, except the accuracy and device type. The accuracy and computation time for the characterization phase relies on the counter enable duration, system clock frequency, number of inverters in ring oscillators, and routing propagation delay. Some of the quite prominent frequency monitoring methods have been proposed for accurate frequency measurement [32] [22] [21]. These methods have been implemented for different device families.

The author Chauhan et al. presented an approach, which improves the reliability of ROPUF design by increasing frequency difference among ring oscillators [8]. The authors have proposed, the enhancement in frequency difference using a new characterization phase, biased placement and k-means clustering. The overall reliability improvement is apparent due to the frequency difference enhancement. However, the other parameters such as uniqueness and uniformity diminish, due to biased placement. Furthermore, the approach is limited to temperature variations. Later, the approach has been modified to improve the randomness and uniqueness along with reliability [9]. The authors additionally, employ the randomized placement to enhance the uniqueness, whereas randomness is improved using random frequency allocation with an LFSR design. The sequences produced by modified ROPUF design have passed NIST statistical tests for randomness. However, the reliability certainly get reduced for the extreme voltage conditions. Furthermore, the routing validation step increases ROPUF synthesis and implementation efforts, making it a comparatively time-consuming process.

The rest of the paper is organized as follows. The problem motivation and current novel contribution are discussed in section 3 and 3.1, respectively. In section 4 we have discussed ring oscillator design, modeling and effect of measurement uncertainties. The proposed approach and the experimental results are presented in sections 5 and 6, respectively. The manuscript has concluded in section 7.

3 Motivation

The ring oscillator PUF is considered a weak PUF, since it can only produce a small number of challenge-response pairs. Therefore, it is widely used for cryptographic key generation, where cryptographic core obfuscates the CRP correlation. Some of the researchers have utilized weak PUF with strong PUF and pseudo random number generator (PRNG) to increase the obfuscation, and thus are able to improve the resistance against machine learning attacks [47] [30]. The PUFs are vulnerable to machine learning attacks, which are capable of replicating challenge-response correlation by employing intelligent machine learning (ML) approaches. Similar, to the ML attacks, a genetic algorithm (GA) based attack has been explored to predict the response of ROPUF design. These approaches can predict CRP with a prediction rate of upto 96% [35] [40].

The authors proposed an improved version of the existing ROPUF design in order to make it as a strong PUF. Further, the reliable PUF-ID generator is used to provide challenges instead of applying challenges directly to the PUF. Here, the reliable PUF-ID introduces obfuscation between input and output and thus reduces the CRP correlation. However, the environmental variations have not been considered with the approach [18]. Similarly, the authors have proposed dual-mode PUF to improve the ML resistance [47], where the approach utilizes two modes; i) counting and ii) state stabilization. However, the obfuscation increases the sensitivity to environmental noise. Thus authors in [47], report a $> 13\%$ bit flipping occurrence due to temperature variations. They do not consider the effect of supply voltage variations. As the supply voltage variation modifies the ring oscillator pulse width nonlinearly, the effect of such variation is more significant as compared to effect due to temperature variation.

These attempts accommodate ways to secure device authentication against ML attacks by combining multiple PUFs, which could lead to increased sensitivity to environmental variations. Hence, a single but more reliable PUF is preferable in order to improve false rejection ratio (FRR) as well as false acceptance ratio (FAR). The other solution is to utilize the error correction schemes, albeit at the cost of FPGA chip area overhead [46], constraining how large a number of PUFs could be realized on the remaining area.

3.1 Our Contribution

The work proposed in this paper is substantial extension of authors' work for ROPUF reliability and uniqueness enhancement using randomized and biased placement on FPGA [9] [8]. Following are our additional contributions in this manuscript.

- The ring oscillator frequency difference is maximized using augmented clustering-based approach. We propose an improved K-means based grouping approach by novel approach for global maximum based inter-centroid frequency selection. This novel augmentation in algorithm results in improving the frequency difference characteristics of the ROPUF.
- The further improvement in the minimum pairwise frequency difference has been obtained by employing novel scheme for centroid relocation based difference maximization. The proposed approach provides the enhancement in the frequency difference whenever the number of ring oscillators in ROPUF increase.
- The randomness evaluation has been performed using NIST statistical tests. We have introduced a controllable randomness factor to configure the randomness of the proposed ROPUF, which also we have validated by employing NIST statistical tests and minimum entropy analysis.
- We propose the elimination of routing validation scheme in design flow. Alternately, we effect modifications in hardware design, which altogether omits synthesis & implementation and thus saves time/efforts.
- We have included a detailed analysis of the improved k-means clustering, overall design time, minimum entropy. We have performed experimentation on increased the number of hardware (FPGA devices) to achieve higher confidence in the results.

4 Ring Oscillator Physical Unclonable Functions

The ring oscillator PUF design consists of M number of ring oscillators divided into two groups as shown in Figure 1, where each group consists of $\frac{M}{2}$ ring oscillators and these ring oscillators are placed over the entire FPGA chip. Presuming that manufacturing variation modifies the frequency of each ring oscillator. A pair of ring oscillator has been selected, each from a group using the input challenge (C), and finally pulse difference has been evaluated using a comparator design. The response (R) is the binary output generated through the comparator design.

4.1 Ring Oscillator Delay Modeling

A typical ring oscillator has been designed with the odd number of inverter and enable logic element. The propagation delay τ_g of a logic element due to manufacturing variation can be assumed as the multivariate Gaussian distribution, and it is dependent on the process parameters, i.e. the threshold voltage (V_{th}), the channel length (L), width (W), etc.

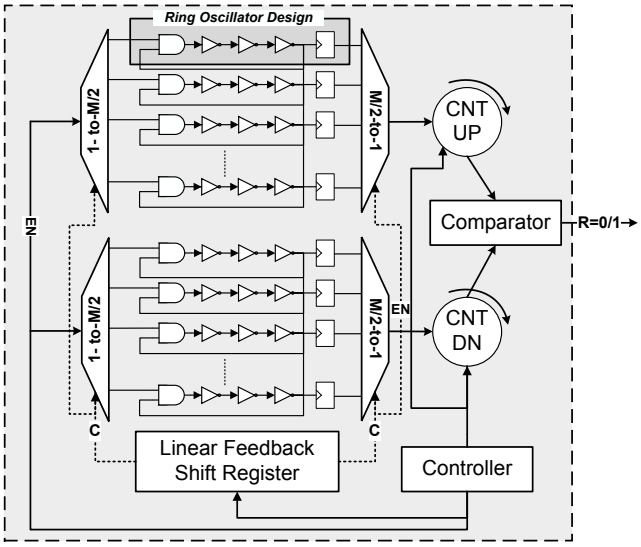


Fig. 1: The ring oscillator based PUF design consists of M ring oscillators. The input and output can be represented as a tuple of challenge-response pair (C, R). The challenges have applied using a maximal length LFSR design.

The τ_g is composed of mainly three components; 1.) average delay component τ_g^{avg}, 2.) systematic delay component τ_g^s and, 3.) random delay component τ_g^r. The propagation delay of i^{th} logic element is the aggregation of all three delay components and it can be represented as (1).

$$\tau_{g_i} = \tau_{g_i}^{avg} + \tau_{g_i}^s + \tau_{g_i}^r \tag{1}$$

The propagation delay (d) of n logic element based ring oscillator is manifested in (2).

$$d = \left(\sum_{i=1}^{n} \left(\tau_{g_i}^{nom} + \tau_{g_i}^s + \tau_{g_i}^r \right) \right) + d_w \tag{2}$$

Here, d_w is the routing path delay of the ring oscillator. The average value appears due to the effective value of process parameters. The systematic delay component arises due to non-uniformity during the fabrication, whereas the random delay component is present due to irregular doping concentration in transistors. Therefore each ring oscillator produces a distinct frequency and further it is used to generate random signatures.

4.2 ROPUF response modeling

The ring oscillator PUF design produces random numbers by performing comparison in between a pair of the ring oscillator. A pair of digital counters has been utilized to capture ring oscillator pulses (α) for t_{on} duration, therefore the propagation delay (d), and the RO frequency (f) at an instance is manifested in (3).

$$d = \frac{t_{on}}{2\alpha}\ , f = \frac{\alpha}{t_{on}} \tag{3}$$

The delay difference between i^{th} and j^{th} ring oscillators can be evaluated using (4).

$$\Delta d = -\frac{t_{on}\Delta\alpha}{2\alpha_i\alpha_j}\ , \Delta f = \frac{\Delta\alpha}{t_{on}} \tag{4}$$

From (4), we can conclude that, $\Delta F \propto \Delta\alpha \propto \Delta d$. Therefore the ROPUF response can be evaluated in terms of either count difference $(\Delta\alpha)$ or frequency difference (Δf), which is manifested in (5).

$$R = \begin{cases} 0, & \text{if } \Delta f \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

A noise (ϵ) has been introduced during count value extraction from counter, it can be expressed using (6).

$$\Delta f_\epsilon = \Delta f_{\epsilon_m} + \Delta f_{\epsilon_t} \tag{6}$$

Here, Δf_{ε_t} and Δf_{ε_m} are the environmental noise and measurement noise. The error-free frequency difference $(\Delta\overline{f})$ can be expressed as the aggregation of systematic $(\Delta\overline{f}_{sys})$, random $(\Delta\overline{f}_{rand})$ and wire frequency difference $(\Delta\overline{f}_w)$. Therefore the actual frequency difference (Δf) can be expressed as (7).

$$\Delta f = \Delta\overline{f}_{sys} + \Delta\overline{f}_{rand} + \Delta\overline{f}_w + \Delta f_{\varepsilon_t} + \Delta f_{\varepsilon_m} \tag{7}$$

The unreliable response is referred as the frequency difference (Δf) sign change for same input during difference conditions.

4.3 Effect of Measurement Error

The frequency difference measurement error Δf_{ϵ_m} is random in nature and it can be modeled as $\mathcal{N}(0, \sigma^2_{\varepsilon_m})$, here mean is assumed to be zero and σ_{ε_m} is the standard deviation for measurement error. The maximum frequency difference measurement error $(\Delta f_{\epsilon_m}^{max}) = 6\sigma$ (range is $\pm 3\sigma$) for 99.7% confidence interval. Therefore highly deviated ring oscillators increase bit flipping possibility by increasing measurement error.

4.4 Effect of Environmental Error

The environmental variations modifies the ring oscillator frequency difference. The source of environmental variation is temperature (T), supply voltage (V) and device aging (λ). Rewriting (7) in (8).

$$\Delta f = \Delta \overline{f} + \Delta f_{\epsilon_t} + \Delta f_{\epsilon_m} \tag{8}$$

From (8), it is clear that reliable condition (no frequency sign change) can be achieved by either (9) or (10).

$$|\Delta \overline{f}| > (|\Delta f_{\varepsilon_t} + \Delta f_{\varepsilon_m}|) \tag{9}$$

$$\Delta \overline{f} * (\Delta f_{\varepsilon_t} + \Delta f_{\varepsilon_m}) > 0 \tag{10}$$

The condition (9) itself is a complete one, but if (9) does not satisfy then (10) can be checked for stability.

5 Proposed Methodology

The proposed approach has been used to increase the frequency difference (Δf) such that all the ring oscillator pairs should satisfy (9), because the effect of noise (Δf_ϵ) in frequency difference can be compensated when the frequency difference is comparatively large. Therefore highly separated ring oscillator frequencies are preferred to improve the reliability.

Algorithm 1 Work flow of the proposed approach

Phase-0 : Constraint Creation
begin
  1: Place & Route Constraint Creation for Characterization
end
Phase-1 : Frequency Variation Profiling
begin
  1: Capture Frequency Data using On-chip Frequency Monitors
  2. Erroneous Frequency Rejection (Post Process)
end
Phase-2 : ROPUF Creation
begin
  1: Clustering Based Grouping.
  2: Centroid relocation for difference maximization.
  3: Ring oscillator group assignment.
  4: Randomized placement of ring oscillators.
  5: Create constraint files.
end

The proposed approach is a two-phase approach, where the first phase extracts the process variation of the entire FPGA chip using $\mathcal{Z}$ ring oscillator instances. Each ring oscillator instance is placed to a single slice. During the second phase, we have selected M ring oscillator instances out of $\mathcal{Z}$ instances, in order to achieve a significant frequency difference, which improves the overall reliability, here $Z \gg M$. The manual placement & routing constraints have been included to fix the placement and routing of ring oscillators. Since we have used fixed placement and routing, therefore no other logic can occupy the allocated resources.

The flow of the proposed approach is shown in Algorithm 1. All the constraints related to the first phase has been created during a pre-processing phase, which is termed as phase-0 or constraint creation phase.

5.1 Phase-0 : Constraint Creation

The characterization phase requires to place $\mathcal{Z}$ number of ring oscillator to the entire FPGA chip, each ring oscillator requires a single slice for implementation (four slices for logic element and single shift register for latch). A software interface has been designed to automate the characterization constraint creation phase using *MATLAB*, *TCL* & *VIVADO* [48], as shown in Figure 2. The preprocessing phase requires to perform only once for the same type of FPGA chips; for example, all Nexys-4DDR FPGA requires only one constraint creation phase.

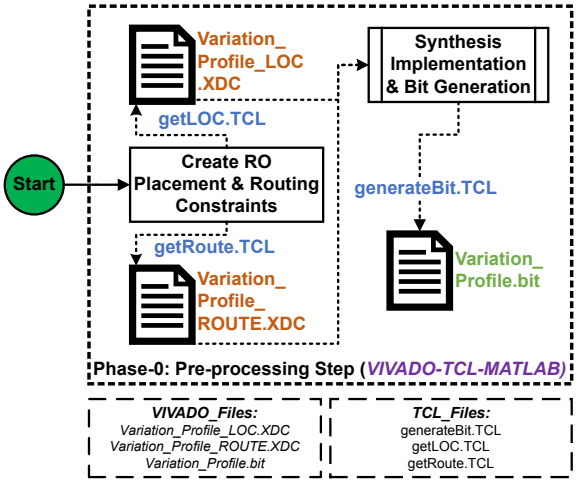


Fig. 2: *MATLAB*, *TCL* and *VIVADO* interface has been used for device constraint creation and bit-file generation for characterization phase. There are two *VIVADO* constraint files, and one bitstream file has been generated using three *TCL* scripts.

5.1.1 Placement & Routing

The ring oscillators have been placed over the entire FPGA region, and *XDCMacro* with detailed routing techniques are used to provide control over place & route. Ring oscillators are constrained not to occupy the central region of FPGA fabric because we have appropriated it for other required circuitry.

The architecture of the Xilinx-7 series FPGA consists of configurable logic blocks (CLB), where each CLB consists of mainly two types of slices, i.e. L and M, and each one provides different routing paths [49]. The placement location of each slice affects the routing path. There are mainly four possible configurations of the slice according to the physical location of the slice w.r.t. switch box. The slice connected to top side is always a *L* and the bottom connected slice can be *L* or *M*. The slice connected to top-left, bottom-left, top-right and bottom-right is labeled as TL, BL, TR and BR, respectively, The different configurations of CLB are mentioned in Figure 3.

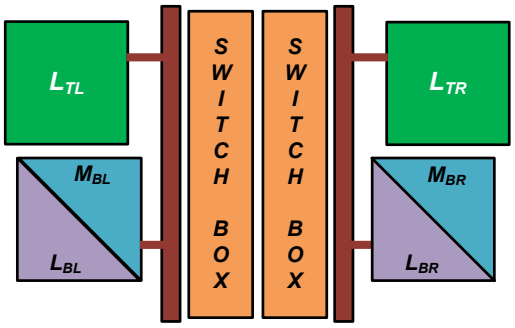


Fig. 3: Configurable logic block (CLB) architecture for *Xilinx-7* series FPGA device with different configurations. Here, $L_{TL}, L_{TR}, L_{BL}, L_{BR}$ represents the slice-L at top-left, top-right, bottom-left, bottom-right. Similarly, M_{BL} and M_{BR} represents the slice-M at bottom-left and bottom right locations.

The other slice classification is based on the routing delay of the ring oscillators. There are three possible sets related to the routing delay, i.e. L_{12}, L_3 and M. The set L_{12} contains slice-L which is placed at top side (i.e. L_{TL} and L_{TR}). Similarly, set L_3 contains L_{BL} and L_{BR} locations, and set M contains M_{BL} and L_{BR}.

$$L_{12} = L_{TR} \bigcup L_{TL},\ L_3 = L_{BL} \bigcup L_{BR} \text{ and } M = M_{BL} \bigcup M_{BR} \tag{11}$$

The routing path of each set should produce different routing delay.

5.2 Phase-1: Frequency Variation Profiling

The frequency variation profiling phase requires two steps; 1) Data extraction and 2) Post processing. The data extraction phase extracts the frequency, and the collected frequencies have been processed to remove the data transmission error.

5.2.1 Data Extraction

The first step for the proposed approach is data extraction by employing on-chip monitors, which can be performed with the help of hardware-software co-interface. The hardware part consists of an FPGA device, and the software interface has been designed *MATLAB*. The data transmission is performed by employing standard *UART* communication protocol with a data transmission speed of 115200 bits/seconds. The extracted count (α_{ro}) data is converted to frequency data (F_{ro}) using (3). The mean value and standard deviation of the collected frequency data have been represented as F_{ro_μ} and F_{ro_σ}, respectively. All the collected data will be discarded after the completion of the second phase, to minimize the risk of information leakage.

5.2.2 Erroneous RO Rejection

The second step performs the post-processing that has been used to remove the highly deviated ring oscillator frequencies. These highly deviated frequencies are termed as "erroneous RO frequencies". Moreover, we have rejected these frequencies and the obtained error-free ring oscillator frequencies (F) using (12).

$$F = F_{ro}\left(\frac{f^x_{ro_\sigma}}{f^x_{ro_\mu}} \leq Th\right), \forall x \epsilon F_{ro} \tag{12}$$

Here, F_{ro} is the ring oscillator frequencies obtained from the characterization phase and (F) represents the error-free ring oscillator frequencies. The mean and standard deviation value of x^{th} RO frequency is represented as $f^x_{ro_\mu}$ and $f^x_{ro_\sigma}$, respectively.

The parameter Th is defined as the maximum allowable value of the normalized standard deviation. We have used normalized standard deviation with mean value to reduce the effect of mean value on the standard deviation. We have analyzed the data of 54 FPGA chips and observed that the maximum of 1% ring oscillator frequencies are showing bit flipping error and some frequencies are showing high standard deviation, which can produce unreliable response bits. Therefore, we have fixed Th such that 5% of the total frequencies are discarded and not used for further processing.

The effect of threshold variation (Th) on the number of the erroneous ring oscillator is depicted in Figure 4. We have achieved a threshold value $Th = 0.002$ for 5% error band. However, the increment in Th value reduces the $\%$ erroneous frequencies and vice versa.

The selected error-free ring oscillator frequencies are represented as F, the mean values are F_μ, the standard deviation values are F_σ and the number of error-free frequencies are represented as $\overline{\mathcal{Z}}$

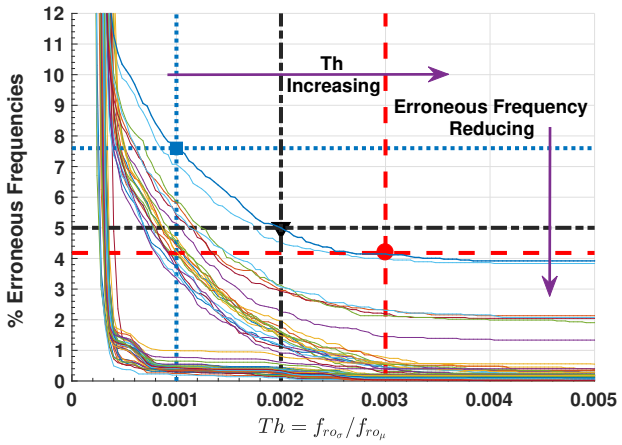


Fig. 4: The effect of threshold selection on the number of erroneous ring oscillator for 54 FPGA chips. The rejection threshold is fixed at $Th = 0.002$, which can reject a maximum of 5% ring oscillator instances.

5.2.3 Frequency Variation Results

We have extracted a total of 11264, 5696 and 3520 ring oscillators count data from each *Nexys-4 DDR*, *Basys-3* and *Zybo* FPGA chips respectively at room temperature. We have collected a total of 32 frequency samples for each ring oscillator. Figure 5 shows the heat map for the mean (F_μ) and standard deviation (F_σ) of frequencies extracted from all three type FPGA chips.

The mean frequency span $F_{\mu_{span}}$ for *Nexys-4 DDR*, *Basys-3* and *Zybo* chips are 64.78MHz, 54.28MHz and 54.71MHz, respectively. Similarly, standard deviation span $F_{\sigma_{span}}$ are 249.4KHz, 235.24KHz and 229.4KHz for *Nexys-4 DDR*, *Basys-3* and *Zybo* FPGA chips, respectively.

5.2.4 Effect of Routing Bias

The reduction in routing delay increases the frequency and vice-versa. Figure 6 shows the effect of routing delay variation on all three possible placement location sets for *Basys-3* FPGA device. There are three different distributions, and each represents a slice type. The statistical distribution parameters have been evaluated with the help of empirical normal distribution of frequencies $\mathcal{N}(\mu, \sigma^2)$. The highest mean frequency 418.10 MHz has been achieved by selecting locations from set L_{12}, and the minimum mean frequency 402.3 MHz is achieved by selecting locations from slice-M. The maximum frequency range for biased placement and unbiased placement are 47.11 and 32.51 MHz, respectively. The improvement in the frequency range for biased placement is about 1.43 times of the unbiased placement.

The ring oscillator implementation to all possible locations (i.e. L_{12}, L_3 and M) introduces routing bias, which increases frequency variation range. Therefore routing delay variation with biased placement certainly improved the frequency difference.

5.3 Phase-2: ROPUF Creation

The obtained set of error-free ring oscillator frequencies (F) are randomly distributed across the frequency scale, and we have to select M ring oscillators out of $\overline{\mathcal{Z}}$ number of ring oscillators to increase frequency difference. Therefore second phase adaptively searches for M suitable locations out of $\overline{\mathcal{Z}}$, which maximizes the pairwise frequency difference (PFD). The frequency difference for a pair of ring oscillator is expressed as (13).

$$\Delta f = \Delta\overline{f} + \Delta f_{\varepsilon_m} + \Delta f_{\varepsilon_t} \tag{13}$$

Therefore, the essential condition required to avoid unstable response generation for each pair is manifested in (14).

$$|\Delta\overline{f}| \geq max\left(|\Delta f_{\varepsilon_m} + \Delta f_{\varepsilon_t}|\right) \tag{14}$$

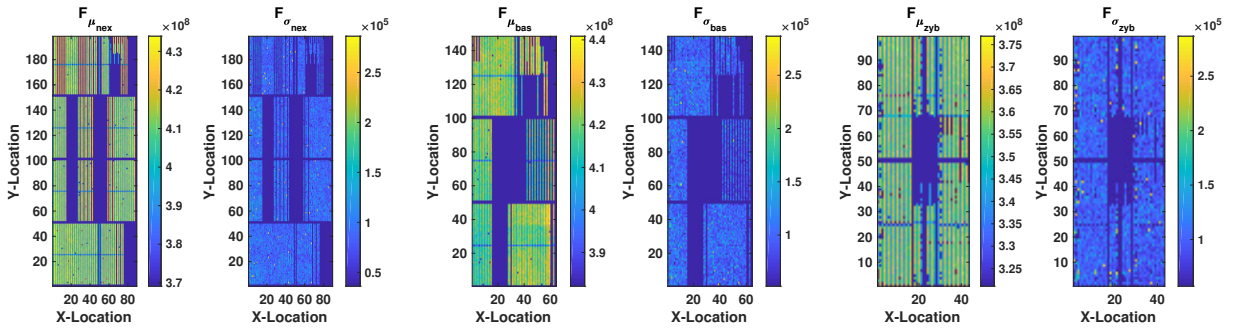
The behavior of each ring oscillator is uncertain, therefore the estimation of $max\left(|\Delta f_{\varepsilon_m} + \Delta f_{\varepsilon_t}|\right)$ is difficult. Consequently, we have designed an adaptive approach to increase $\Delta\overline{f}$, such that it can compensate both environmental error as well as measurement error. Although measurement error is limited by employing erroneous RO rejection technique, still there are chances that aggregation of both errors can produce unreliable response bits.

The approach, we are using is based on grouping such that it increases the inter-group frequency difference. The step used to create ROPUF place and route constraints are mentioned in Figure 7.

We are using K-means clustering based approach which has been used to create groups, such that the intra-group distance should be minimized [2]. The reduction in intra-group distance may improve the inter-cluster distance. There are mainly two reasons to select K-means based clustering, which are as follows.

- The number of frequencies to be selected (M) is fixed, therefore the number of clusters is the same as the number of frequencies to be selected.
- K-means clustering is sensitive to outliers (frequencies which are far apart from the mean value), and inclusion of noisy frequencies significantly improved the frequency difference.

We are assuming that K-means centroid frequencies ($\overline{C}$) during each iteration seems not to resemble at the frequency



(a) **Nexys-4 DDR (Artix-7)** (b) **Basys-3 (Artix-7)** (c) **Zybo (ZYNQ-7000)**

Fig. 5: Frequency variation profiling data evaluated for three different FPGA chips. Here, $F_{\mu_{dtype}}$ and $F_{\sigma_{dtype}}$ are the mean and standard deviation value heat-map for $dtype$ FPGA device. Here, $dtype$ is nex, bas and zyb for *Nexys-4DDR*, *Basys-3* and *Zybo*, respectively.

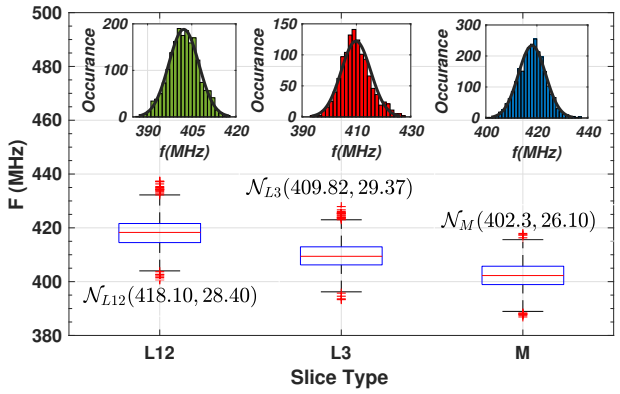


Fig. 6: The measured frequency for different Slice locations after applying erroneous RO rejection on Basys-3 FPGA chip. The empirical mean and standard deviation for each type of slice have been evaluated using $\mathcal{N}(\mu, \sigma^2)$.

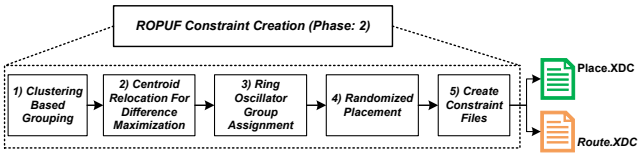


Fig. 7: ROPUF Constraint creation phase 2, this phase requires five steps to create place & route constraints.

scale (F), therefore we are choosing the i^{th} frequency which is near to the centroid ($\overline{c}_i$) using (15).

$$\overline{c}_i = F(\text{indexof}(min|F - \overline{c}_i|)), \forall i \in 1, 2, \ldots, \overline{\mathcal{Z}} \tag{15}$$

The inter-group frequency difference is further improved using centroid relocation based approach. The centroid relocation based approach shift the centroid and searches for the frequencies, which increases minimum pairwise frequency difference (χ). The minimum pairwise frequency difference (χ) is defined as the minimum frequency difference among $\binom{M}{2}$ possible pairs, and it can be evaluated using (16).

$$\chi = min(|\Delta\overline{f}_i|), \forall i \in \{1, 2, \ldots \binom{M}{2}\}; \tag{16}$$

The metric χ provides a worst-case analysis of frequency difference for the proposed approach. Five steps have been used to generate ROPUF place & route constraints. As we have modified the ring oscillator design by inserting a latch circuit, therefore we have omitted the routing path validation scheme from the previous work [9]. The following steps have been used during the second phase to select suitable ring oscillator locations, which improves χ.

5.3.1 Clustering Based Grouping

K-means clustering approach works based on expected maximization approach, which is used to minimize the intra-cluster distance [14], and it may improve the inter-cluster distance. We are searching for the set of frequencies, which improves the minimum pairwise frequency difference (χ). We have added a function in the K-means clustering to test the minimum pairwise frequency difference (χ) during each iteration and stored the centroid values, which provides the highest χ among iterations. The convergence of the improved algorithm is the same as the standard K-means algorithm based on intra-cluster distance. The steps require to maximize χ has been explained in Algorithm 2.

Initialization K-means approach sensitive to the initialization, thus we have applied different centroid initialization techniques, i.e. Linearly spaced frequency vector, Uniform density based initialization, K-means based initialization (K-means++) and Random initialization.

Algorithm 2 Modified K-means based grouping approach

Initialization:
begin
  $\overline{C}$ = **generateInitialCentroid**(F);
end
K-means Clustering
begin
  1: Calculate inter centroid minimum distance (χ) in between all centroid $\bar{c}_j, j \in 1, 2, \ldots, M$ and store χ in variable β_p and store the corresponding centroid value in a list ℓ_p.
  2: Compute distance ψ_i for i^{th} frequency instance f_i, $i \in 1, 2, \ldots, \mathcal{Z}$ with all centroids $\overline{C}$.
  3: Label i^{th} instance with j such that ψ_i is minimum.
  4: Replace centroid $\bar{c}_j, j \in 1, \ldots, M$ with average of the frequencies labeled with j.
  5: Find nearest existing frequency for each $\bar{c}_j$ from F and store it in list ℓ_c.
  6: Calculate χ in between all elements of list ℓ_c and store it in β_c.
  7: if $\beta_c > \beta_p$ **then**
    (i) $\ell_p = \ell_c$ and β_p to β_c
  end
  else
    (i) No modifications.
  end
  8: Repeat Steps (2) to (8) until $\bar{c}_j$ does not becomes fixed or the maximum number of iterations k_{max}.
  9: The optimized centroid list is ℓ_p and maximized χ is β_p.
end

- **Linearly spaced centroid:** The linearly spaced vector is generated in the ring oscillator frequency range, where two consecutive frequencies have the same frequency difference.
- **Uniform density based centroid:** Two consecutive frequencies have the same number of instances.
- **K-means based initial centroid:** K-means itself has been used for initial centroid generation.
- **Random initial centroids:** Choose initial centroid uniformly at random in a provided frequency range.

The effect of seeding on χ is shown in Figure 8. The linearly spaced centroids based initial vector improves the χ for $M \leq 32$, but the performance degrades for $M > 32$. Similarly, the performance for K-means based initialization provide better χ for $M \geq 64$.

Intra-group Minimization The intra group minimization has been performed with the help of distance function (ψ). The minimum distance $\psi_{i_{min}}$ for i^{th} frequency instance with all centroids $\overline{C}$ has been evaluated using (17).

$$\psi_{i_{min}} = min\left(\|c_j - f_i\|\right), \forall \bar{c}_j \in \overline{C} \tag{17}$$

K-means based approach minimizes ψ during each iteration and for each frequency instance. The assessment of K-means clustering has been performed by employing mean

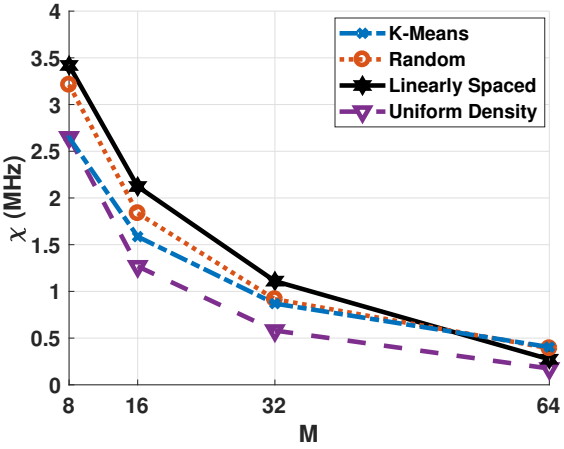


Fig. 8: The variation of minimum pairwise frequency difference (χ) for different seed initialization methods and ring oscillator instances (M).

intra-cluster distance (MICD) [37]. The mean intra-cluster distance (MICD) for j^{th} centroid can be determined using (18).

$$\begin{aligned} \text{MICD}_j &= \frac{1}{N_j} \sum_{x \in F_{\bar{c}_j}} \|x - \bar{c}_j\| \\ \overline{MICD} &= \frac{1}{M} \sum_{j=1}^{M} (\text{MICD}_j) \end{aligned} \tag{18}$$

Here $F_{\bar{c}_j}$ is the set of ring oscillator frequencies relating $\bar{c}_j$ centroid frequency, N_j are the number of elements in j^{th} cluster and $\overline{MICD}$ is the mean of MICD.

Inter-group Maximization The inter-cluster minimum distance (ICMD) has been used to evaluate the performance of K-means clustering [37]. The metric ICMD is same as minimum pairwise frequency difference (χ) for K-means clustering. The evaluation of χ has performed during each iteration, and χ value along with the centroids has been stored in β and list ℓ, respectively.

K-means convergence is the same as earlier to minimize intra-cluster distance, but we have modified the algorithm in such a way that it also checks the centroids, which can improve the χ. The algorithm converges with the minimization of the mean intra-cluster distance.

The proposed approach dependent on number of ring oscillator instances (M), therefore we have assessed the performance of K-means by employing $\overline{MICD}$ and χ evaluation metric. The effect of improved K-means on χ and $\overline{MICD}$ is depicted in Figure 9. $\overline{MICD}$ metric improvement is similarly as standard K-means, but we are searching for the global maximum of χ, which is achieved at iteration #16.

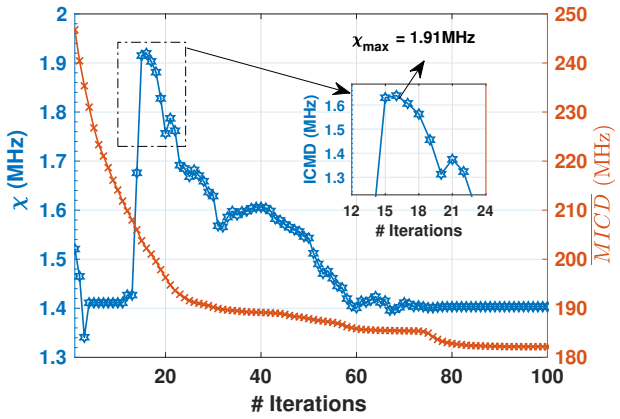


Fig. 9: The effect of improved K-means on χ during iterations. The maximum χ_{max} is achieved at iteration #16, therefore optimized centroids are centroids available at iteration #16.

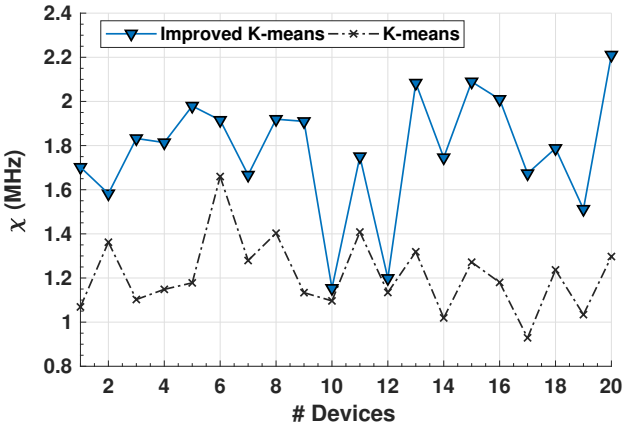


Fig. 10: The performance evaluation of improved K-means with the K-means in terms of minimum pairwise frequency difference (χ) on 20 *Nexys-4DDR* FPGA devices and the ROPUF configuration is $M16$.

We are storing the centroid values belongs to the iteration #16 and consider these centroids as optimized centroids.

The effectiveness of the improved K-means based grouping approach has been evaluated by comparing it with some other frequency selection techniques, i.e. Mean based selection, Median based selection, Random frequency selection and K-means based selection [9].

- **Mean based selection:** Select the M frequency data points such that the neighbor frequencies have an equal distance.
- **Median based selection:** Select M frequency data points such that the difference between two consecutive frequencies has the same number of ring oscillator instances.
- **Random frequency selection:** Select M frequency instances, uniformly at random from all the available frequency instances.

We have improved K-means clustering by including inter-centroid global maximum selection technique along with mean based initialization. The frequency difference increment is evaluated by comparing the improved K-means approach with the other approaches. We have performed a comparison of improved K-means with previous K-means for 20 *Nexys-4 DDR* FPGA devices, as shown in Figure 10. We have found that improved K-means provides an average improvement of 46.4% as compared to K-means [9].

The comparison with other approaches with improved K-means are shown in Figure 11, which signifies that the improved K-means based approach increase the χ among all methods. The second best performance is the mean based frequency selection procedure. The proposed approach provide same frequency difference for $M \leq 16$ and it improves the χ for $M > 16$.

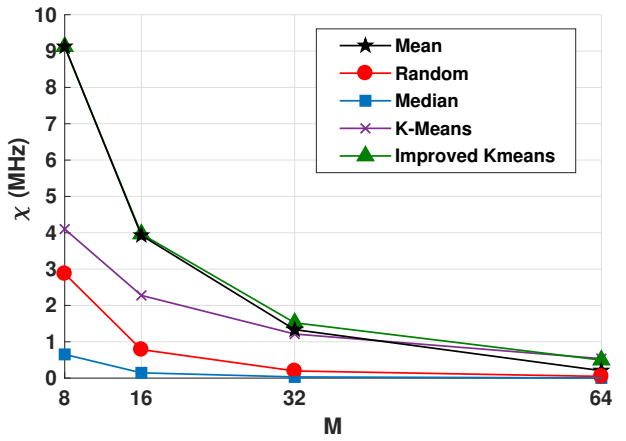


Fig. 11: The comparative performance evaluation of the improved K-means with other frequency selection approaches in terms of χ and the variation of χ with M.

5.3.2 Relocating Centroid For Difference Maximization

The centroid obtained through the modified K-means algorithm increases frequency difference but the difference is randomly distributed, and minimum pairwise frequency difference can be increased by shifting selected frequency points, therefore we are proposing centroid relocation based difference maximization approach as shown in Algorithm 3.

The approach has been used after getting centroids from a modified K-means clustering approach. The approach evaluates frequency difference and checks the possible movement direction, If the frequency difference in the left side is larger than right side then shifting has to be performed in the left direction, else the direction will be the right side. The shifting can reduce the high-frequency difference and

Algorithm 3 Centroid relocation for difference maximization

Input: Sorted ring oscillator frequency mean values = ν
Number of centroids = M
Maximum Iterations = max_{iters}
K-means centroid list = ℓ_p
Pseudocode:
begin
1: Perform sorting operation in ascending order on K-means centroid list ℓ_p and stored it in ℓ_s .
2: Evaluate successive pairwise centroid frequency difference $\Delta\ell_{s_k}, k = (i, j), i \in 1, 2, \ldots 2M-1$ and $j = i+1$.
3: Find the minimum value $\Delta\ell_{s_{min}}$ and its index k_m from frequency difference list $\Delta\ell_{s_k}$. Assign threshold th to $\Delta\ell_{s_{min}}$.
4: Select the neighbor frequency difference $\Delta\ell_{s_{k_{m-1}}}$ and $\Delta\ell_{s_{k_{m+1}}}$.
5: Check neighbor frequency difference, **if** $\Delta\ell_{s_{k_{m-1}}} > \Delta\ell_{s_{k_{m+1}}}$ then centroid movement direction $dir = L$ **else** $dir = R$. L = Left and R = Right.
6: if $dir = L$ then shifting element is ℓ_{s_i} from sorted frequency data ν, else shifting elements is ℓ_{s_j}.
7: if $dir == L$ **then**
(i) Find the index of ℓ_{s_i} and $\ell_{s_{i-1}}$ from frequency data ν, and store it in $\hbar_R$ and $\hbar_L$, respectively.
(ii) Search new $\overline{\ell}_{s_i}$ from $\nu_{\hbar_R}$ to $\nu_{\hbar_L}$ such that the index increment factor p should satisfy $|\nu_{\hbar_L} - \nu_{\hbar_{R-p}}| > th$, $p \in 1, 2, \ldots, p_{max}$.
(iii) If p does not satisfy (ii), then reverse the direction for search $dir = R$ and goto step (6).
(iv) New centroid value $\overline{\ell}_{s_i}$ is $\nu_{\hbar_{R-p_{max}}}$.
end
else
(i) Find the index of ℓ_{s_j} and $\ell_{s_{j+1}}$ from frequency data ν, and store it in $\hbar_L$ and $\hbar_R$, respectively.
(ii) Search new $\overline{\ell}_{s_j}$ from $\nu_{\hbar_L}$ to $\nu_{\hbar_R}$ such that the index increment factor p should satisfy $|\nu_{\hbar_R} - \nu_{\hbar_{L+p}}| > th$, $p \in 1, 2, \ldots, p_{max}$.
(iii) If p does not satisfy (ii), then reverse the direction for search $dir = L$ and goto step (6).
(iv) New centroid value $\overline{\ell}_{s_j}$ is $\nu_{\hbar_{L+p_{max}}}$.
end
8: Update current centroid ℓ_{s_i} or ℓ_{s_j} with $\overline{\ell}_{s_i}$ or $\overline{\ell}_{s_j}$, respectively.
9: Repeat step (2) - (9) until location varies or till maximum number of iterations max_{iter}.
end

increases the low-frequency difference. The approach is iterative and performed until the possible movement locations does not exhaust.

The improvement in χ is evaluated for 20 Nexys-4 DDR FPGA devices for $M = 8, 16, 32$ and 64 ring oscillators. Figure 12 shows the improvement in χ and the improvement is significant. Moreover, the average improvement for $M = 8$ and $M = 16$ are less than 10%, but the improvement for $M = 32$ and $M = 64$ are 27.69% and 50.37%. The improvement is higher for a large value of M because, for large ring oscillators the K-means based approach provides small χ and centroid shifting improvement is comparatively significant.

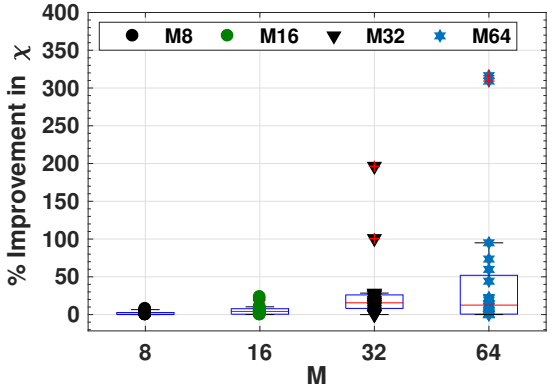


Fig. 12: The % improvement in the χ by employing centroid relocation based approach and the effect of M value on χ.

5.3.3 Ring Oscillator Group Assignment

The hardware design comprises of two different groups; lower groups and upper groups, where all ring oscillator instances of the lower group (LG) should be compared to the ring oscillators at the upper group (UG). Each group contains $\frac{M}{2}$ ring oscillator frequencies, such that each group contains some frequencies in ordered and some frequencies in random trend. The ratio of κ decides that the number of ring oscillator frequencies are randomly allocated with total frequencies. The possible value of parameter κ can be evaluated using (19).

$$\kappa = \frac{i}{2^{x-1}}, \forall i \in \{0, 1, \ldots, 2^{x-1}\}, x = log_2\left(\frac{M}{2}\right) \quad (19)$$

Let's suppose that $M = 16$, then $x = 3$ and the possible randomness ratio $\kappa = \{0, 0.25, 0.5, 0.75, 1\}$. Therefore out of M ring oscillator frequencies, $\kappa \times M$ frequencies are randomly assigned and remaining $(1 - \kappa) \times M$ are orderly assigned. The group assignment affects the uniformity, and two NIST statistical tests, i.e. frequency and block frequency test can provide a validation of uniformity, therefore we are validating ring oscillator assignment performance with NIST statistical tests.

5.3.4 Randomized Placement

The allocated ring oscillators frequencies follow the sorting order consequently we have performed group based randomization. We are randomly indexing selected ring oscillator frequencies. The ring oscillators are randomly placed,

which deforms the output sequences in such a manner that the sequences should avoid creating the same response patterns on different hardwares. The randomized placement effectively removes the patterns, and it also reduces the chances for appearing the same sequence, therefore it enhanced the uniqueness as well as it also improves the randomness of the responses. The effect of randomized placement has been evaluated by employing the uniqueness metric and minimum entropy.

5.3.5 Create Constraint Files

The randomized placement provides the selected locations for each ring oscillators. We have extracted the routing path information from the characterization constraints and create placement and routing constraint files. The constraint creation is automated through *MATLAB* and *TCL* scripting.

5.4 Algorithm Computation time

The overall computation time is bifurcate in two parts; 1.) Frequency characterization phase (t_{p1}) and 2.) ROPUF constraints creation (t_{p2}). The synthesis implementation and bitstream generation time is not included for the fair assessment.

5.4.1 Frequency Characterization Time (t_{p1})

The frequency characterization time (t_{p1}) is approximated linearly varied with the number of ring oscillator instances ($\mathcal{Z}$) and the number samples for each ring oscillators (m) during characterization process. The ring oscillator enable pulse duration for each ring oscillator is fixed at $122.87\mu sec$, and the data transmission time is considerable extensive as compared to enable pulse duration. The average time requires to measure a single sample of a ring oscillator instance is about $3msec$. We have unfolded circuit two times to reduce computation time. Moreover, it increases the characterization area, but the area will be unoccupied after the completion of the characterization phase.

5.4.2 ROPUF Constraint Creation Time (t_{p2})

Apart from the characterization, the second phase also requires time to produce ROPUF place and route constraints. K-means clustering has a computation time complexity of $\mathcal{O}(n \log n)$ [2]. Similarly, the computation time requires for the centroid relocation based approach has been evaluated using *MATLAB* delay function. Figure 7 shows the computation time dependency on the number of iterations for the centroid relocation approach, and it is evident that average computation time increases with the increment in the number of average iterations. The computation time for constraint creation is much smaller than characterization ($t_{p2} \ll t_{p1}$), therefore it can be managed with characterization.

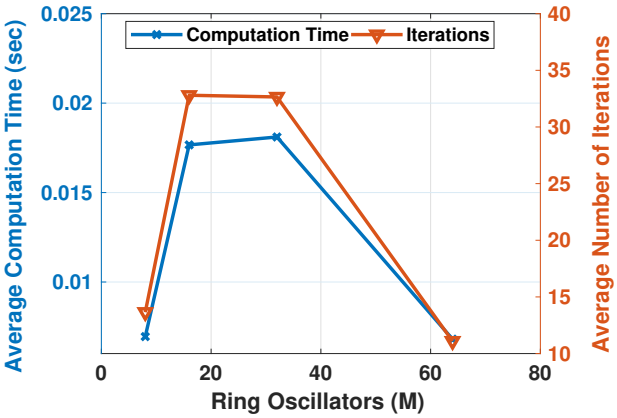


Fig. 13: The effect of M value on average computation time based on *MATLAB* software and average number of iteration for centroid relocation approach.

6 Experimental setup & Evaluation Metric

The proposed method has been evaluated using standard quality metrics, i.e. uniqueness and reliability [33]. The reliability and uniqueness metric have been evaluated by using intra-chip hamming distance HD_{intra} and inter-chip hamming distance HD_{inter}. Hamming distance can be evaluated using (20).

$$\text{HD}(x, y) = \sum_{z=1}^{k} |R_{x_z} - R_{y_z}| \tag{20}$$

Here, R_x and R_y are two k bit response sequences. The randomness has been evaluated using NIST random statistical suite. Moreover, we have used a maximum of 54 FPGA devices (i.e. 24 *Nexys-4 DDR*, 10 *Basys-3* and 20 *Zybo*), and a single ROPUF has been implemented on a device, but the number of ROPUF per device can be increased by utilizing the remaining slices. We have used four different configurations, i.e. $M8$, $M16$, $M32$ and $M64$, whereas these configurations can produce 15, 63, 255 and 1023 bits long responses, respectively.

6.1 Experimental Setup

We have designed "Hardware-Software" based co-interface for the automation of the characterization and ROPUF constraints creation phase by employing TCL and MATLAB

scripts. The temperature variation is achieved with a temperature environment enclosure (*Espec SH-241 Temperature & Humidity Chamber*) as shown in Figure 14, and the temperature range is fixed in between $-5°C$ to $75°C$ with a step size of $10°C$. Similarly, supply voltage variation has been achieved by modifying one *Basys-3* FPGA device, and the voltage variation range has kept in between $0.9V$ to $1.1V$ with a step size of 20mV. The real-time temperature and the supply voltage has been monitored for verification, using *XADC* interface and *TCL* scripts. The reference operating conditions (T_{ref}=35$°C$ and V_{ref}=1000mV) has been used for the evaluation perspective.

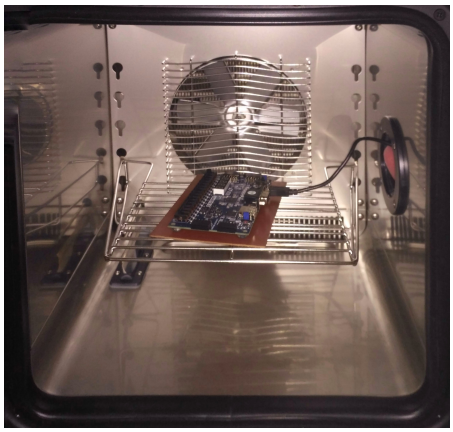


Fig. 14: Temperature variation enclosure, *"Espec SH-241 Temperature & Humidity Chamber"* with a maximum variation range from $-40°$ C to $150°C$.

The proposed approach is completed in two different phases. The first phase is used to collect frequency data using on-chip frequency monitors. In order to extract frequency variations, we have designed and interfaced an IP-Core for characterization, termed as "VMAP_zybo" as shown in Figure 15. The Zynq processing system IP-core has interfaced with AXI and UART-Lite for data extraction and data transmission [50]. Similarly, we have implemented a standard UART protocol for *Artix-7* FPGA devices.

6.2 Randomness Test

The randomness of the response is evaluated by applying the NIST statistical test suite [5]. "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" revision 1-A, has been used to test the randomness based on χ^2 statistical distribution. The suite consists of a total of 15 possible tests, and the applicability of each test is dependent on the input sequence length. Furthermore, each sequence should have at least 100 bits to

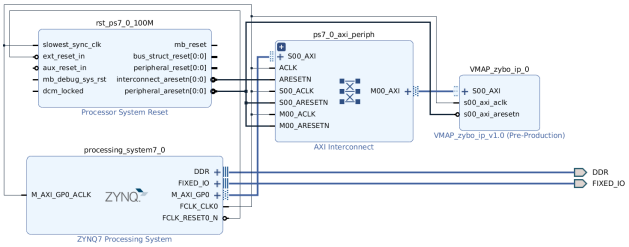


Fig. 15: Ring oscillator characterization and variation MAP creation using VMAP_zybo IP-Core, interfaced with AXI and *ZYNQ-PS* (processing system).

apply 9 basic tests. The authors have validated the randomness effect on small length sequences [26] [53] [56]. However, at least one million bit sequences required to apply all 15 statistical tests. We have applied NIST tests to M16 and M32 configuration only, however, the remaining configurations M4 and M8 do not provide the appropriate sequence length to apply the test.

We can apply nine out of fifteen and ten out of fifteen tests for $M32$ and $M64$ configuration, respectively. The list of applicable tests according to sequence length is as follows. The frequency, block frequency, runs, longest runs of ones in a block, serial, approximate entropy, DFT, and cumulative sums, whereas each test has its own significance. For example, two out of these nine tests are used to validate the frequency of 1's and 0's in the sequence, which provides a better assessment of the uniformity.

We have produced ROPUF sequences by applying a random initial LFSR seed and test the generated sequences for NIST statistical tests. We have produced a total of 54 sequences, each from an FPGA device with a single seed value, and out of 54 at least 51 sequences (at least 94.44% sequences) should have to be passed for a confidence interval $\alpha = 0.01$.

The effect of randomness factor κ has been validated, and the obtained results are shown in Figure 16. There are only two possible κ values provides 100% passing rate for $M32$ and two possible κ values for $M64$ configurations. For a small value of κ, frequency and cumulative sum tests have been failed, because of the over-uniform responses. As the κ value increases, the uniformity reduces, and randomness increases, which improves the results and further increment in κ produces responses with over-randomness, which leads to failed most of the tests. The summarized results for applicable sequences are presented in Table 1. There are two possible sequences for M32 and two possible sequences for M64 has passed all the NIST statistical tests.

Table 1: NIST statistical tests results variations with applicable κ and M values

Test Name	**M32** ($\kappa = 0.375$)		**M32** ($\kappa = 0.5$)		**M64** ($\kappa = 0.3125$)		**M64** ($\kappa = 0.375$)	
	P-Value	**% Proportion**	**P-Value**	**% Proportion**	**P-Value**	**% Proportion**	**P-Value**	**% Proportion**
Frequency	0.03517	100.00 ✔	0.07572	98.14 ✔	0.02054	100.00 ✔	0.15376	96.30 ✔
Block Frequency	0.41902	100.00 ✔	0.65793	100.00 ✔	0.09657	98.14 ✔	0.09731	100.00 ✔
Cumsum Forward	0.07572	100.00 ✔	0.02074	98.14 ✔	0.05898	100.00 ✔	0.09657	96.30 ✔
Cumsum Reverse	0.01560	100.00 ✔	0.15376	98.14 ✔	0.28966	100.00 ✔	0.19168	96.30 ✔
Runs	0.41902	100.00 ✔	0.12232	100.00 ✔	0.07572	100.00 ✔	0.35048	98.14 ✔
Longest Runs of 1's	0.39706	100.00 ✔	0.81653	100.00 ✔	0.95583	98.14 ✔	0.31908	98.14 ✔
Entropy	0.08558	100.00 ✔	0.57490	100.00 ✔	0.01304	100.00 ✔	0.01451	98.14 ✔
Serial	0.53414	100.00 ✔	0.28966	98.14 ✔	0.17187	100.00 ✔	0.13728	96.30 ✔
Serial	0.17186	100.00 ✔	0.17186	100.00 ✔	0.10879	100.00 ✔	0.65793	98.14 ✔
DFT	***NA***	***NA***	***NA***	***NA***	0.02695	100.00 ✔	0.45593	100.00 ✔

✔ : Passed Test
NA: Not applicable
Block frequency test block length (**M**): 20 ,
Max entropy block length (m_e): $m_e < \lfloor log_2(n) \rfloor - 5$
Serial test block length (m_s): $m_s < \lfloor log_2(n) \rfloor - 2$
Confidence interval (α_{test}) : 0.01

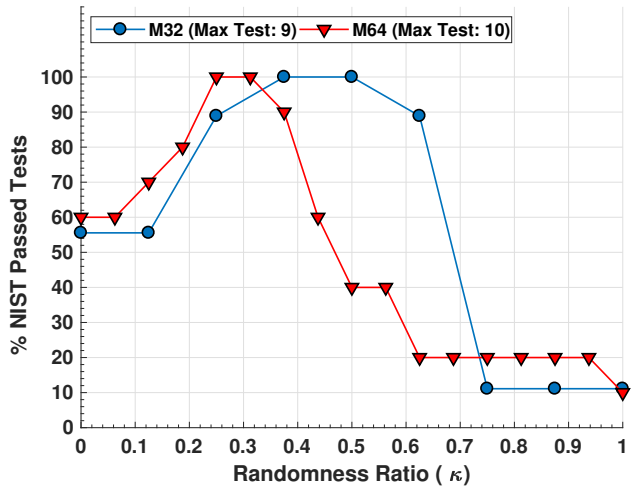


Fig. 16: NIST statistical test pass percentage with the randomness ratio (κ) variation. The number of passed tests have $\kappa = 0.375, 0.5$ for M32 and $\kappa = 0.3125, 0.375$ for M64 configuration.

6.2.1 Minimum Entropy

The random sequences should have high entropy to pass the NIST statistical test for approximate entropy. Moreover, the entropy extracted through the ring oscillator can be expressed as $log_2(M!)$ [43]. Since, we are using M32 and M64 design, which can provide a maximum of $log_2(32!)$ and $log_2(64!)$ bit entropy.

The proposed approach uses a randomized placement and an LFSR design. Therefore, these two techniques certainly improve the minimum bound of the entropy. Whereas, we can measure lower bound of entropy ($\mathcal{H}^i_{\mathcal{LB}}$) for i^{th} bit position and full lower bound of entropy ($\mathcal{H}_{\mathcal{FLB}}$) of the whole sequence using (21) [19].

$$\mathcal{H}^i_{\mathcal{LB}} = -log_2\left[\frac{max(b^i_1, b^i_0)}{q}\right]$$
$$\mathcal{H}_{\mathcal{FLB}} = \frac{1}{k}\sum_{i=1}^{k}\mathcal{H}^i_{\mathcal{LB}} \tag{21}$$

Here, b^i_1 and b^i_0 are the number of 1's and 0's in a binary sequence on many hardware (q) for the i^{th} bit location and k is the maximum response bits extracted from single ROPUF instance. The full minimum entropy ($\mathcal{H}_{\mathcal{FLB}}$) for 54 devices has been evaluated using (21) and depicted in Figure 17 for both M32 and M64 design with all possible κ configurations.

The mean value of $\mathcal{H}_{\mathcal{FLB}}$ for M32 and M64 are 0.8519 and 0.8564, respectively, with all κ variation. However, the maximum theoretical value is 1 when the number of 1's and 0's are equally distributed over each bit position. Therefore we have achieved a good minimum entropy value for both M32 and M64 hardware designs.

6.3 Reliability Evaluation

PUF can produce a random signature from a digital circuit. Moreover, the digital circuit uses the device manufactur-

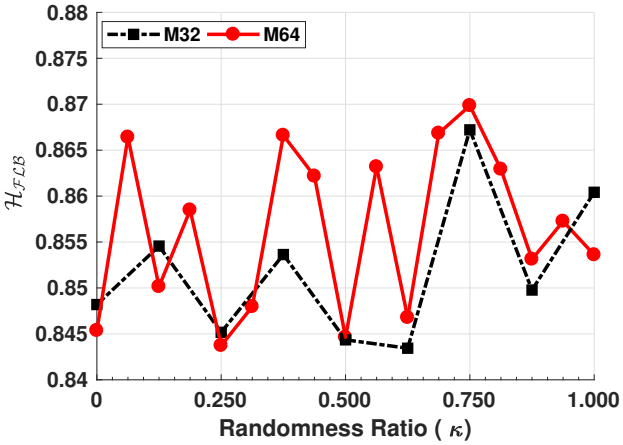


Fig. 17: Full minimum entropy $\mathcal{H}_{\mathcal{FLB}}$ for M32 and M64 with different κ configurations

.

ing variation, which is uncontrolled and sensitive to environmental noise. Consequently, frequencies of ring oscillator may vary, which produces random bit flipping. The proposed approach increases the minimum pairwise frequency difference (χ) and presuming that it may improve reliability.

The reliability is defined for PUF response variation under different environmental conditions (e). The reliability can be evaluated in terms of intra-chip hamming distance (HD_{intra}), which is defined as the number of response (R) bit flipping at different environmental conditions out of k bits. The variation has been observed by comparing response (R_e) with the golden responses (R_g). The golden response is the response which can be extracted at reference environmental variations. The intra-chip hamming distance for i^{th} PUF can be evaluated using (22).

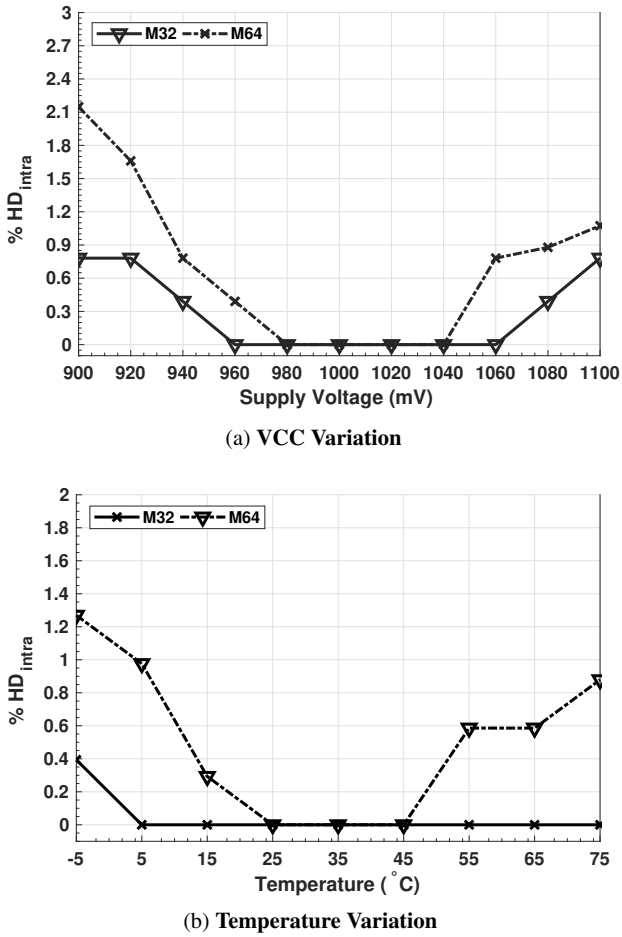
$$\text{HD}_{intra_i} = \frac{1}{k}\sum_{j=1}^{e} HD(R_{g_i}, R_{i,j}); \tag{22}$$

The reliability metric r_i of i^{th} device can be evaluated using (23).

$$r_i = \left[1 - \left(\frac{\text{HD}_{intra_i}}{e}\right)\right] \tag{23}$$

Here, R_{g_i} and $R_{i,j}$ both are the k bit response for the i^{th} device. R_{g_i} is the reference or golden response and $R_{i,j}$ is the response obtained in the presence of the j^{th} environmental condition.

The proposed improved K-means with centroid relocation enhanced the minimum pairwise frequency difference (χ), therefore reliability improvement is apparent. The evaluation metric HD_{intra} in the presence of supply voltage and temperature variation is manifested in Figure 18. The worst



(a) **VCC Variation**

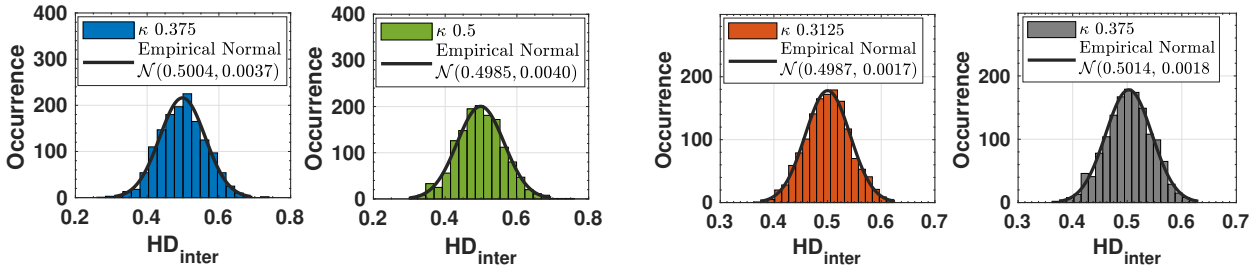
(b) **Temperature Variation**

Fig. 18: The effect of environmental variations (temperature and supply voltage) on intra chip hamming distance (HD_{intra}), measured for a single *Basys-3* FPGA device.

case reliability for $M32$ and $M64$ have been improved by 1.18% and 0.78% in the presence of voltage variation.

The sequences obtained through $M8$ and $M16$ provide a response length of 15 and 63 bits, therefore the reliability for these approaches are far better than $M32$. The increment in the M value can reduce minimum pairwise frequency difference (χ), consequently, the reliability has been reduced. Therefore, we are considering the large value of M, i.e. $M32$ and $M64$, which have already passed NIST statistical test, and assumed that reliability for $M8$ and $M16$ is better than $M32$ as well as $M64$.

The average reliability obtained through the Figure 18 has been summarized in Table 2. The average reliability improvement for M32 and M64 are 0.35% and 1.14%, respectively.



(a) $M32$ (b) $M64$

Fig. 19: Inter-chip hamming distance (HD_{inter}) for $M32$ with $k = 0.375, 0.5$ and $M64$ with $k = 0.3125, 0.375$.

Table 2: % The summarized reliability results in the presence of environmental variations. Here, r_{max}, r_{min} and r_{avg} are the maximum, minimum and average reliability values

M	VCC Variation			Temp Variation		
	r_{max}	r_{min}	r_{avg}	r_{max}	r_{min}	r_{avg}
32	1	0.9921	0.9961	1	0.9961	0.9980
64	1	0.9785	0.9890	1	0.9873	0.9936

6.4 Uniqueness

The uniqueness is quite an important metric to ensure that two devices are not producing the same responses for the same applied challenges. The uniqueness can be evaluated using inter-chip hamming distance (HD_{inter}). The inter-chip hamming distance is evaluated in between q number of different devices. The inter-chip variation appears due to the inter-chip random variation. The inter-chip hamming distance (HD_{inter}) in between q number of different chip can be evaluated using (24).

$$\text{HD}_{inter} = \frac{1}{k}\sum_{i=1}^{q-1}\sum_{j=i+1}^{q}\text{HD}(R_i, R_j) \tag{24}$$

Here, R_i and R_j are the k bit responses and $i \neq j$. The theoretical value of the interchip hamming distance is $\text{HD}_{inter} = 0.5$. The uniqueness metric (u) is evaluated using (25).

$$u = \frac{2 \times HD_{inter}}{q(q-1)} \tag{25}$$

Figure 19 represents the inter chip hamming distance (HD_{inter}) for $M32$ and $M64$ configuration. Figure 19a and 19b represents the inter-chip hamming distance for M32 with $\kappa = 0.375, 0.5$ and M64 with $\kappa = 0.3125, 0.375$. The distribution obtained through inter-chip hamming distance looks like Gaussian distribution, therefore we have fit the distribution with standard normal distribution and empirically determined the statistical parameters for inter-chip hamming distance. The mean value for each distribution is near to theoretical value 0.5.

The randomized placement deforms the sorting order of the placement locations, therefore the output response is random, and the responses obtained from the different devices are randomized for the same challenge (C). The improvement in the uniqueness is significant, and the uniqueness value is near to theoretical value. The obtained average uniqueness value has a maximum deviation of 0.22% as compared to the theoretical value.

6.5 Comparison

The proposed method is examined along with some of the existing approaches, and a comparison has been presented in Table 3. The reliability metric is the quite essential in order to evaluate the reproducibility of the sequences because the non-reproducible sequences can not be utilized for device authentication application. Moreover, the standalone ROPUF can not defend against ML attacks and modeling attacks.

All the existing approaches in the group have reliability $> 99\%$ except [26], which is not considered for reliability comparison. The approach [16] has the highest reliability, whereas our method occupies the second position among the group. However, this approach [16] doesn't take voltage variations into consideration. We have considered both supply voltage and the temperature variation, and even the temperature range is higher by $15°C$ compared to the method in [16]. Moreover, our approach provides 100% reliability in the same temperature variation range, as mentioned in [16]. The method in [18] provides reliable responses in the presence of both temperature and supply voltage variation, however, it requires post-processing to achieve that much reliability. The approaches [1] [8], provide reliable response with no environmental variations. The approaches [34] [43], provide better reliability as compared to all other approaches, except our approach, in the presence of environmental variations. Therefore if we are considering the effect of envi-

Table 3: Comparison of the group of PUF characteristics of the proposed scheme with those of earlier schemes

Method	$\%r_{avg}$	$\%u$	k	**Resources**	**Resources/bit**		**FPGA**	**Tech.**	ΔT	$\frac{\Delta V}{V_{ref}}$	NIST ($\frac{App.}{Total}$)
						Norm					
ROPUF [43] ⋆	99.52	46.15	128	1024 ROs	8.0000	63.79	Vertex-4	90nm	**100**$°C$	20%	**?**
Maiti-CRO [34]	99.14	47.31	127	512 Slices	4.0314	32.14	Spartan-3E	$90nm$	$40°C$	**40**%	**?**
Improved ROPUF [26]	95%	49.32	150	1200 Slices	8.0000	63.79	Spartan3E	90nm	**?**	0.83%	$\frac{8}{15}$ ✔
Self Compare [16] †	**99.80**	49.07	**256**	**4 LUTs**	**0.0039**	0.032	Spartan-6	45nm	$65°C$	**?**	**?**
Compact RO [1]	99.16	47.13	**256**	32 Slices	0.1250	0.996	Spartan-6	45nm	**?**	**?**	✖
PUF-ID [18] ‡	99.40	45.60	128	128 Slices	1.0000	7.974	**Artix-7**	**28nm**	$75°C$	20%	**?**
Biased ROPUF [8]	99.05	35.83	**256**	32 Slices	0.1250	0.996	**Artix-7**	**28nm**	$40°C$	**?**	✖
Previous Work [9]	99.35	49.83	255	32 Slices	0.1254	1.000	**Artix-7**	**28nm**	$80°C$	20%	$\frac{9}{15}$ ✔
Proposed Work (M32)	99.70	**49.90**	255	32 Slices	0.1254	1.000	**Artix-7**	**28nm**	$80°C$	20%	$\frac{9}{15}$ ✔

? = Failed/Not reported. k = Maximum Response Length. $\Delta T = T_{max} - T_{min}$ (Temperature Variation Range).
$\Delta V = V_{max} - V_{min}$ (Supply Voltage Variation Range). † = Required resources excluding adaptive tuning circuits.
⋆ = Assuming that 1-RO requires 1 Slice. ‡ = Post processing. V_{ref} = Reference voltage. **Norm**= Normalized Resources/bit.
NIST = NIST test status. ($\frac{\text{Applicable}}{\text{Total}}$) and (✔ = Passed, ✖ = Failed)

ronmental variations, then our approach shows superiority among the considered approaches in the group.

Our proposed approach requires 32 slices in order to produce 255 bit CRPs. The method in [16] provide minimum utilization of the slices, however, the area for adaptive tuning circuit on hardware requires extra circuitry, which is not reported in the paper. The methods [1] [8] [9] occupy similar number of slices in comparison to our present approach. Furthermore, the resources required to produce a single bit response is approximately the same as reported in [8] [9].

The approach in [43], provides significantly good reliability with the comparatively large environmental variation range. However, the method consumes 32 times extra slices to implement ring oscillators in order to produce the same number of response bits. The approach in [34], provides a reliable response, but the uniqueness is less and area utilization is significantly large i.e. approximately 30 times.

The randomized placement provides the highest uniqueness value in the group of existing PUF design. The method in [16] and [26] provide a uniqueness value $> 49\%$. However, [16] requires extra computation time and resource utilization for adaptive tuning, and the method in [26] has poor reliability. Apart from [16] and [26], no other approach is able to achieve that much of uniqueness value.

Finally, one of the strengths of the present approach is *randomness*. The proposed approach passes all nine applicable NIST statistical test by employing random group allocation of the ring oscillator. We have included a configurable randomness factor κ, which allows passing all nine tests for two randomness factor. The approaches in [26] passed eight tests out of nine applicable tests. Apart from these, no other method in the group is bale to pass the NIST statistical tests.

In summary, the proposed approach has these features, which show the superiority over the existing approaches- i) excellent reliability despite environmental variations, ii) enhanced uniqueness, iii) improved randomness, iv) hardware/area efficient and v) implemented on recent technology (28nm). All these features are simultaneously available in the cost of one additional phase for characterization, which can be accomplished with auto characterization framework at the time of PUF creation.

6.6 Security Threats

As the proposed approach is engaged in two phases, therefore, the information leakage during PUF creation can produce serious threats. Moreover, the PUF creation environment should be secure. If an attacker is able to extract the frequency information, then it is possible for him to evaluate centroid frequency by employing the proposed approach. The randomized placement reduced the chances to predict responses bits because the frequency ordering and the placement is random. Moreover, the ROPUF architecture is the same as earlier; therefore, it can not resists against the modeling attacks, i.e. GA based modeling [35] and quicksort

based modeling [39]. However, the proposed approach enhanced the performance of all three essential features such as reliability, uniqueness and randomness. Therefore it can be used with some other true random number generator (TRNG) or pseudo-random number generator (PRNG) to improve security performance.

6.7 Application Scenario

Since the standalone ROPUF can not be used for device authentication as it can be attacked using quicksort as well as the genetic algorithm. The proposed ROPUF design can be used with cryptographic approaches with suitable post-processing scheme. The authors S. Vyas et al. have presented the effect of unreliable response bits on hardware utilization for the error correction circuity [46]. Any error correction circuitry consumes extra hardware as well as it reduces the entropy of the response bits [31]. For example, BCH decoder circuitry consumes more hardware utilization with the increment in BER. Therefore, the less worst-case unreliable response bits can reduce the area overhead for the error correction circuitry.

Table 4: Worst case unreliable response bit comparison with other approaches and the suitable BCH configurations.

Method	**Resp (k)**	**Wost Case Unreliable bits**	**Suitable BCH** (n, t)
[34]	128	7	BCH(127,7)
[18]	128	4	BCH(127,4)
[9]	255	6	BCH(255,6)
This work	255	4	BCH(255,4)

n = Block length, t = Number of bits for correction

The comparison of worst-case unreliable bits and the suitable error correction circuitry has mentioned in Table 4. By considering BCH codes, the proposed ROPUF can provide approximately twice response bits with an equal amount of unreliable bits as compared to [18].

7 Conclusion & Future Work

The proposed method using novel augmentations provide significant improvement in reliability as well as uniqueness, simultaneously despite voltage as well as temperature variations. The proposed approach has been primarily implemented and validated on *28nm-technology Xilinx* FPGA. The randomness for the proposed method passes NIST statistical tests for randomness facilitated by randomized placement. The approach requires two-phases- the first phase requires one-time additional efforts (trusted environment) to perform frequency pre-characterization on FPGA; whereas the second phase is utilized for RO selection in order to maximize frequency difference.

The ring oscillator PUF design proposed in this paper can be used along with with other circuitry like TRNG/PRNG to improve the machine learning resistivity. The work on secure device authentication protocol with error correction circuitry, and the reliability examination against device aging is currently being pursued by us.

Acknowledgements We also gratefully acknowledge SMDP-C2SD project (ODRC No. 1000110086) project funded by *Ministry of Electronics & IT, Government of India* for technical support.

We are thankful to Mr Bharat Kasyap, MNIT, Jaipur to provide support for experimental setup design.

References

1. Anandakumar, N.N., Hashmi, M.S., Sanadhya, S.K.: Compact implementations of FPGA-based PUFs with enhanced performance. In: Proc. of 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), pp. 161–166 (2017)
2. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007)
3. Bakiri, M., Guyeux, C., Couchot, J.F., Oudjida, A.K.: Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses. Computer Science Review **27**, 135 – 153 (2018)
4. Barbareschi, M., Natale, G.D., Bruguier, F., Benoit, P., Torres, L.: Ring oscillators analysis for security purposes in Spartan-6 FPGAs. Microprocessors and Microsystems **47**, 3 – 10 (2016)
5. Bassham III, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., Heckert, N.A., Dray, J.F., Vo, S.: Sp 800-22 rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Tech. rep., NIST, Gaithersburg, MD, United States (2010)
6. BÃűhm, C., Hofer, M., Pribyl, W.: A microcontroller SRAM-PUF. In: Proc. of 2011 5th International Conference on Network and System Security, pp. 269–273 (2011)
7. Bolotnyy, L., Robins, G.: Physically unclonable function-based security and privacy in RFID systems. In: Proc. of Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07), pp. 211–220 (2007)
8. Chauhan, A.S., Sahula, V., Mandal, A.S.: Novel placement bias for realizing highly reliable physical unclonable functions on FPGA. In: Proc. of 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1–6 (2018)
9. Chauhan, A.S., Sahula, V., Mandal, A.S.: Novel randomized biased placement for FPGA based robust random number generator with enhanced uniqueness. In: Proc. of 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), pp. 353–358 (2019)

10. Chi-En Yin, Gang Qu: Temperature-aware cooperative ring oscillator PUF. In: Proc. of 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 36–42 (2009)
11. Daihyun Lim, Lee, J.W., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **13**(10), 1200–1205 (2005)
12. Delvaux, J., Verbauwhede, I.: Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. In: Proc. of 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 137–142 (2013)
13. Delvaux, J., Verbauwhede, I.: Fault injection modeling attacks on 65 nm arbiter and ro sum pufs via environmental changes. IEEE Transactions on Circuits and Systems I: Regular Papers **61**(6), 1701–1713 (2014)
14. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of The Royal Statistical Society, Series B **39**(1), 1–38 (1977)
15. Dichtl, M., Golić, J.D.: High-speed true random number generation with logic gates only. In: P. Paillier, I. Verbauwhede (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, pp. 45–62. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
16. Gan, J., Zhou, J., Wang, N.: A FPGA-based RO PUF with LUT-based self-compare structure and adaptive counter time period tuning. In: Proc. of 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2018)
17. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS '02, pp. 148–160. ACM, New York, NY, USA (2002)
18. Gu, C., Hanley, N., O'neill, M.: Improved reliability of FPGA-based PUF identification generator design. ACM Trans. Reconfigurable Technol. Syst. **10**(3), 20:1–20:23 (2017)
19. Gu, C., Liu, W., Hanley, N., Hesselbarth, R., O'Neill, M.: A theoretical model to link uniqueness and min-entropy for PUF evaluations. IEEE Transactions on Computers **68**(2), 287–293 (2019)
20. Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, CHES '07, pp. 63–80. Springer-Verlag, Berlin, Heidelberg (2007)
21. Herkle, A., Mandry, H., Becker, J., Ortmanns, M.: In-depth analysis and enhancements of RO-PUFs with a partial reconfiguration framework on Xilinx Zynq-7000 SoC FPGAs. In: Proc. of 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 238–247 (2019)
22. Hesselbarth, R., Wilde, F., Gu, C., Hanley, N.: Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs. In: Proc. of 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 126–133 (2018)
23. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Transactions on Computers **58**(9), 1198–1210 (2009)
24. Johnson, A.P., Chakraborty, R.S., Mukhopadhyay, D.: A puf-enabled secure architecture for FPGA-Based IoT applications. IEEE Transactions on Multi-Scale Computing Systems **1**(2), 110–122 (2015)
25. Kanuparthi, A., Karri, R., Addepalli, S.: Hardware and embedded security in the context of internet of things. In: Proceedings of the 2013 ACM Workshop on Security, Privacy & Dependability for Cyber Vehicles, CyCAR '13, pp. 61–64. ACM, New York, NY, USA (2013)
26. Kodýtek, F., Lórencz, R., Buček, J.: Improved ring oscillator PUF on FPGA and its properties. Microprocessors and Microsystems **47**, 55 – 63 (2016)
27. Kömürcü, G., Pusane, A.E., Dündar, G.: Dynamic programming based grouping method for RO-PUFs. In: Proceedings of the 2013 9th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), pp. 329–332 (2013)
28. Kumar, S.S., Guajardo, J., Maes, R., Schrijen, G., Tuyls, P.: Extended abstract: The butterfly PUF protecting IP on every FPGA. In: Proc. of 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 67–70 (2008)
29. Lewis, T.G., Payne, W.H.: Generalized feedback shift register pseudorandom number algorithm. Journal of ACM (J. ACM) **20**(3), 456–468 (1973)
30. Ma, Q., Gu, C., Hanley, N., Wang, C., Liu, W., O'Neill, M.: A machine learning attack resistant multi-PUF design on FPGA. In: Proc. of 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 97–104 (2018)
31. Maes, R., Maes, R., Darmstadt, T.U.: Physically unclonable functions: Constructions, properties and applications. In: Ph.D. thesis, KU Leuven, Ingrid Verbauwhede (promotor) (2012)
32. Maiti, A., Casarona, J., McHale, L., Schaumont, P.: A large scale characterization of RO-PUF. In: Proc. of 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 94–99 (2010)
33. Maiti, A., Gunreddy, V., Schaumont, P.: A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions, pp. 245–267. Springer New York, New York, NY (2013)
34. Maiti, A., Schaumont, P.: Improved ring oscillator PUF: An FPGA-friendly secure primitive. Journal of Cryptology **24**(2), 375–397 (2011)
35. Nozaki, Y., Yoshikawa, M.: Security evaluation of ring oscillator puf against genetic algorithm based modeling attack. In: L. Barolli, F. Xhafa, O.K. Hussain (eds.) Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 338–347. Springer International Publishing, Cham (2020)
36. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. Science **297**(5589), 2026–2030 (2002)
37. Rokach, L., Maimon, O.: Clustering Methods, pp. 321–352. Springer US, Boston, MA (2005)
38. Rostami, M., Koushanfar, F., Karri, R.: A primer on hardware security: Models, methods, and metrics. Proceedings of the IEEE **102**(8), 1283–1295 (2014)
39. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS '10, pp. 237–249. ACM, New York, NY, USA (2010)
40. Saha, I., Jeldi, R.R., Chakraborty, R.S.: Model building attacks on physically unclonable functions using genetic programming. In: Proc. of 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 41–44 (2013)
41. Sahoo, S.R., Kumar, S., Mahapatra, K.: A modified configurable RO PUF with improved security metrics. In: Proc. of 2015 IEEE International Symposium on Nanoelectronic and Information Systems, pp. 320–324 (2015)
42. Simons, P., van der Sluis, E., van der Leest, V.: Buskeeper PUFs, a promising alternative to D flip-flop PUFs. In: Proc. of 2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 7–12 (2012)
43. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: Proc. of 2007 44th ACM/IEEE Design Automation Conference, pp. 9–14 (2007)
44. Tang, B., Lin, Y., Zhang, J.: Improving the reliability of RO PUF using frequency offset. In: Proc. of 2014 International Conference on Field-Programmable Technology (FPT), pp. 338–341 (2014)
45. Tehranipoor, F., Karimian, N., Yan, W., Chandy, J.A.: A study of power supply variation as a source of random noise. In: Proc. of 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), pp. 155–160 (2017)

46. Vyas, S., Dumpala, N.K., Tessier, R., Holcomb, D.E.: Improving the efficiency of PUF-based key generation in FPGAs using variation-aware placement. In: Proc. of 2016 26th International Conference on Field Programmable Logic and Applications (FPL), pp. 1–4 (2016)
47. Wang, Q., Gao, M., Qu, G.: A machine learning attack resistant dual-mode PUF. In: Proceedings of the 2018 on Great Lakes Symposium on VLSI, GLSVLSI '18, pp. 177–182. ACM, New York, NY, USA (2018)
48. Xilinx-Inc: Vivado design suite tcl command reference guide. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2013_4/ug835-vivado-tcl-commands.pdf (2013). Accessed 22.07.2019
49. Xilinx-Inc: Vivado design suite user guide using constraints. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_4/ug903-vivado-using-constraints.pdf (2013). Accessed 22.07.2019
50. Xilinx-Inc: Zynq-7000 SoC technical reference manual. https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf (2013). Accessed 22.07.2019
51. Yamamoto, D., Sakiyama, K., Iwamoto, M., Ohta, K., Takenaka, M., Itoh, K.: Variety enhancement of PUF responses using the locations of random outputting RS latches. Journal of Cryptographic Engineering **3**(4), 197–211 (2013)
52. Yan, W., Jin, C., Tehranipoor, F., Chandy, J.A.: Phase calibrated ring oscillator PUF design and implementation on FPGAs. In: Proc. of 2017 27th International Conference on Field Programmable Logic and Applications (FPL), pp. 1–8 (2017)
53. Yin, C., Qu, G.: Improving PUF security with regression-based distiller. In: Proc. of 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2013)
54. Yin, C., Qu, G., Zhou, Q.: Design and implementation of a group-based RO PUF. In: Proc. of 2013 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 416–421 (2013)
55. Yu, H., Leong, P.H.W., Xu, Q.: An FPGA chip identification generator using configurable ring oscillator. In: Proc. of 2010 International Conference on Field-Programmable Technology, pp. 312–315 (2010)
56. Zhang, Q., Liu, Z., Ma, C., Li, C., Zhang, L.: FROPUF: How to extract more entropy from two ring oscillators in FPGA-Based PUFs. In: R. Deng, J. Weng, K. Ren, V. Yegneswaran (eds.) Security and Privacy in Communication Networks, pp. 675–693. Springer International Publishing, Cham (2017)