In-Memory Unified TRNG and Multi-Bit PUF for Ubiquitous Hardware Security

Aashu Singhal
Electrical engineering
IIT Gandhinagar

Abdul Qadir Ronak
Electrical engineering
IIT Gandhinagar

Pupul Dalbehra
Electrical engineering
IIT Gandhinagar

***Abstract*—This project implemented an SRAM architecture that incorporates in-memory generation of dynamic and multi-bit static entropy. The architecture integrates a true random number generator (TRNG) and a physically unclonable function (PUF) within a standard bitcell and periphery, maintaining compatibility with memory compiler designs. The TRNG generates bits based on the discharge of bitlines caused by cumulative column-level leakage. Two PUF bits per accessed bitcell are uniquely derived from the bitline discharge rate.**

I. Introduction

In an era defined by the relentless evolution of technology, securing sensitive information has become an increasingly paramount concern. As our reliance on electronic systems grows, the need for robust and reliable methods of safeguarding data has spurred the development of innovative security features. Among these, Hardware Physical Unclonable Functions (PUFs) and True Random Number Generators (TRNGs) have emerged as pivotal components in enhancing the security landscape.

PUFs are specialized electronic components that exploit the inherent physical variations within hardware to create unique, virtually unclonable signatures. These distinctive identifiers offer a potent defense against unauthorized access and counterfeiting, making PUFs a cornerstone in the realm of hardware security.

On the other hand, TRNGs play a crucial role in cryptographic applications by providing a source of truly unpredictable random numbers. In a digital world where the strength of cryptographic algorithms depends on the unpredictability of keys, TRNGs become indispensable for ensuring the robustness of encryption and authentication protocols.

An intriguing synergy exists between PUFs, TRNGs, and Static Random-Access Memory (SRAM). SRAM, a fundamental component of electronic memory systems, serves as the foundation for the implementation of PUFs and TRNGs. Its unique characteristics make it an ideal host for in-memory generation of random numbers and security keys, offering an efficient solution for secure key generation and unpredictable random number generation.

A salient attribute of SRAM that can be leveraged for pervasive entropy and security applications lies in the intrinsic variations present within SRAM structures. These variations manifest as both static and dynamic entropy. The disparities introduced during the chip manufacturing process serve as a stable static entropy source, providing constant and chip-specific entropy variations. This static entropy can be harnessed for the generation of Physically Unclonable Function (PUF) keys. Simultaneously, the variations induced by temperature fluctuations and supply voltage deviations act as a dynamic entropy source. This dynamic entropy can be effectively utilized for the generation of random numbers in cryptographic applications.

In the pursuit of True Random Number Generators (TRNGs), a fundamental principle unifies a diverse array of methodologies, all centered around the extraction of intrinsic randomness or unpredictability from natural processes. These approaches span various domains: physical processes exploit the inherent variability in phenomena like thermal or atmospheric noise, quantum techniques hinge upon the intrinsic unpredictability of quantum processes, radioactive decay capitalizes on the stochastic nature of decay events, chaos-based methodologies navigate the intricate complexities within chaotic systems, and memristor-based approaches harness distinctive electronic properties for random sequence generation. Collectively, these methods converge on a common objective—ensuring the generation of genuine randomness for TRNGs by tapping into the inherent stochasticity of diverse natural and physical phenomena.

II. CIRCUIT IMPLEMENTATION AND ANALYSIS

The complete circuit implementation consists of three major segments: the SRAM memory, the TRNG segment and the PUF segment. The circuit is implemented on a 28nm technology node.

A. SRAM

We are implementing a 64×64 SRAM memory module for the demonstration of both TRNG and PUF, as both are integrated inside a single SRAM memory. For TRNG, we use a single column from the array, and for the PUF, we use two adjacent columns from the array. Fig.2 and Fig.3 represent the whole SRAM memory and the SRAM column respectively.

1) SRAM Cell: We have implemented an 8T cell structure for making the SRAM, instead of the conventional 6T structure. The difference is that separate word lines are designated for read and write operations to prevent contention between them. Since we are working with noise, the noise margins play an important role in our implementation, and the benefit of the 8T cell structure is that the read and write margins

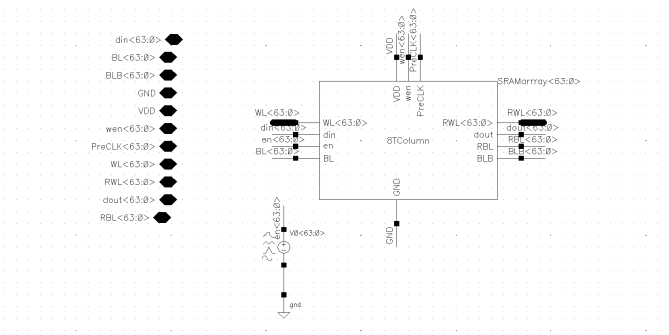

Fig. 1. SRAM memory unit

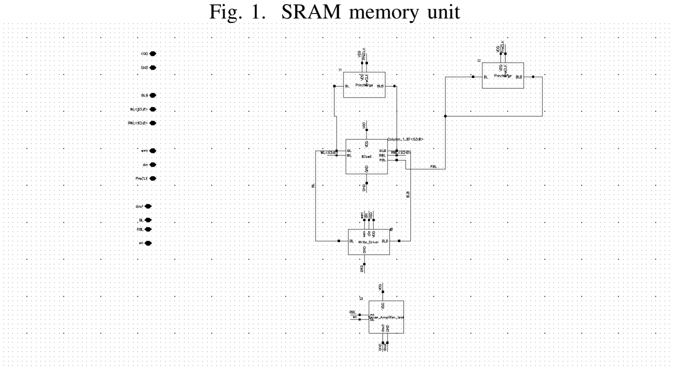

Fig. 2. SRAM Column 64x1

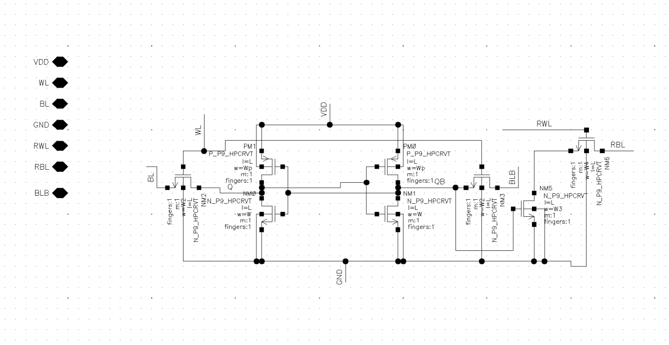

Fig. 3. SRAM Cell

are independent of each other. Each individual SRAM cell consists of 8 transistors. All PMOS and NMOS components are designed with a minimum width of 100 nm and a length of 30 nm. The SRAM cell is represented in Fig.3

2) Pre-charge Circuit: This circuit is employed to pre-charge all the bit lines before read and write operations. The activation and deactivation of the circuit are determined by the Pre-clk signal, facilitating the charging of the bitlines accordingly. Additionally, the circuit features an equalizer transistor, which ensures a consistent and equal voltage level between the two bitlines in case of a write operation. All PMOS components are configured with a width of 350 nm and a length of 30 nm. The precharge component is represented in Fig.4

3) Sense Amplifier: The SRAM incorporates a single-ended sense amplifier with a reference voltage set at 800mV. This amplifier detects subtle voltage variations in RBL(Read Bitline), amplifying the small variation to expedite the SRAM reading process. It enhances the sensitivity to voltage differentials and contributes to faster and more efficient read operations by accurately detecting and amplifying the stored data. The Sense amplifier design is represented in Fig.5

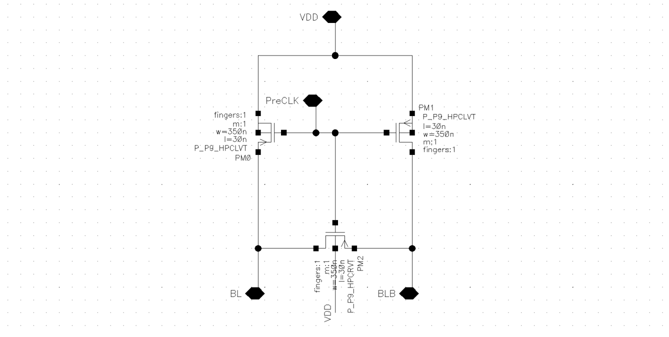

Fig. 4. Precharge Circuit

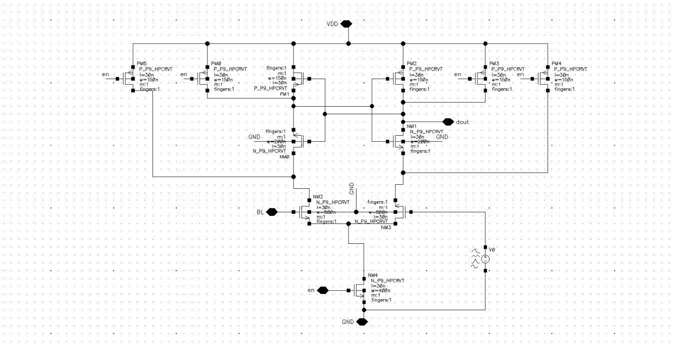

Fig. 5. Sense Amplifier

4) Write-driver: This circuit is responsible for writing data to the write bitline and write bitline bar. It incorporates a write-enable signal to initiate the writing process, and the input signal, denoted as ”din,” specifies the data to be written.The Write Driver is represented in Fig.6

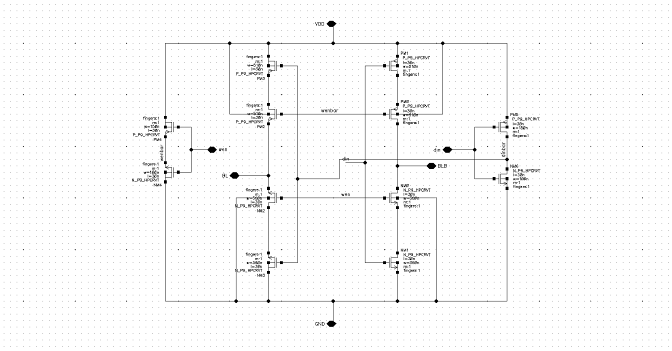

Fig. 6. Write Driver

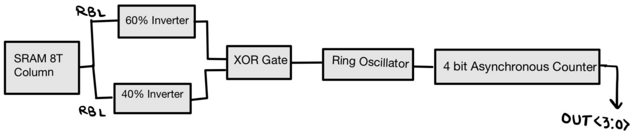

Fig. 7. Peripheral Circuitry of TRNG

B. True Random Number Generator(TRNG)

The TRNG segment consists of two parts: the entropy generation part and the digitization part. Both the parts are integrated with SRAM.

1) Dynamic Entropy Generation: The discharge rate of the bitline in SRAM can be used to generate dynamic entropy by harnessing inherent random noise during bitline capacitance discharge at very low transistor current. Furthermore, the combined contributions of leakage and current noise from bitcells on the same bitline exploit multiple sources of randomness simultaneously, thereby enhancing overall randomness. In TRNG generation, we utilize the cumulative leakage current on the read bitline from the bitcell access and pull-down (driver) transistors in an SRAM column. The leakage current in each cell depends mainly on factors such as temperature and supply voltage variations. To generate dynamic entropy, we initiate by pre-charging the read bitline and the bitline capacitance, followed by disabling all wordlines. Subsequently, the cumulative bitline leakage current from all bitcells discharges the read bitline (RBL), taking a random time to complete. This gradual and variable bitline discharge rate is transformed into a random pulse width, commencing when the bitline voltage reaches 60% of the supply voltage and concluding when it reaches 40%. The random pulse is then directed to a time-to-digital converter to digitize the random pulse width obtained through bitline discharge, resulting in a 4-bit true random number. The complete transformation of bitline discharge to the 4-bit number is shown in Fig. 8

The different components used in the TRNG peripheral circuit are:

60% and 40% inverter: The 60% inverter is a low-skew inverter with PMOS width of 100 nm and an NMOS width of 500 nm, both with a length of 30 nm. The design is optimized to initiate the rising edge of the output at 540 mV, corresponding to 60% of the 0.9V (Vdd). It ensures the prompt activation of the enable signal as soon as the voltage reaches 60% Vdd.

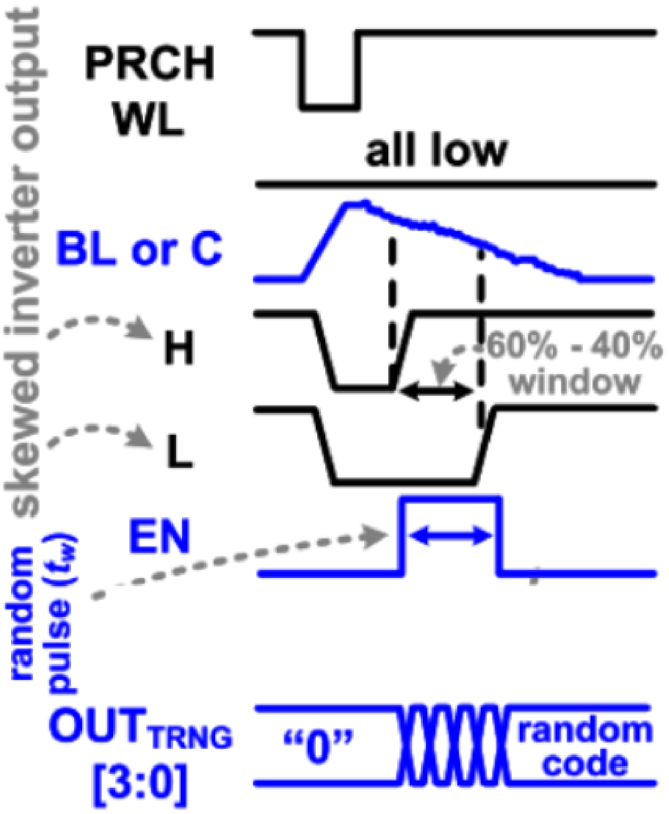

Fig. 8. Dynamic Entropy Generation and Digitization

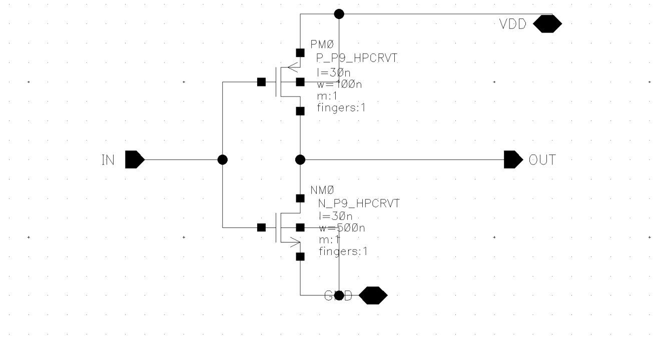

Fig. 9. 60% inverter

The 40% inverter is a high-skew inverter with a PMOS width of 800 nm and NMOS width of 100 nm, each featuring a length of 30 nm. The design is calibrated to initiate the falling edge of the output at 360 mV, constituting 40% of the 0.9V (Vdd). It ensures the rapid deactivation of the enable signal as soon as the voltage descends to 40% Vdd. The skewed gates are represented in the Fig. 9 and 10 respectively

The output of the skewed inverter is given to a **XOR gate** which produces signal based on the time difference between the outputs of skewed gates, giving us a pulse between 60% and 40% level of bitline discharge.

2) Digitization Circuit: The TRNG output, derived from the discharge rate attributed to the leakage current, is created through a Time-to-Digital Converter (TDC) based on a ring

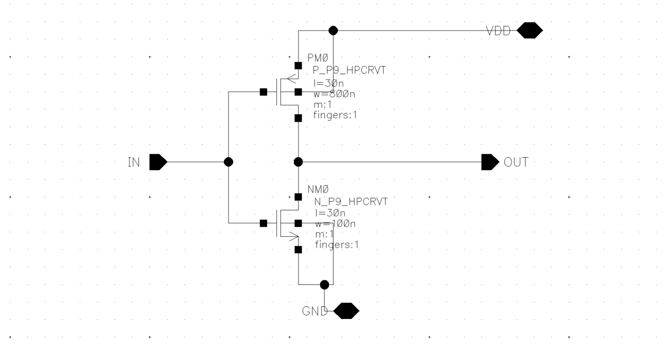

Fig. 10. 40% inverter

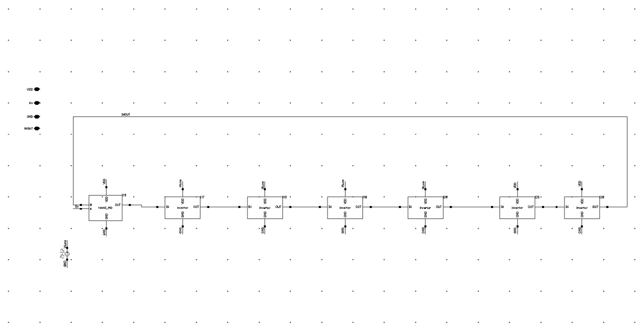

Fig. 11. RO circuit

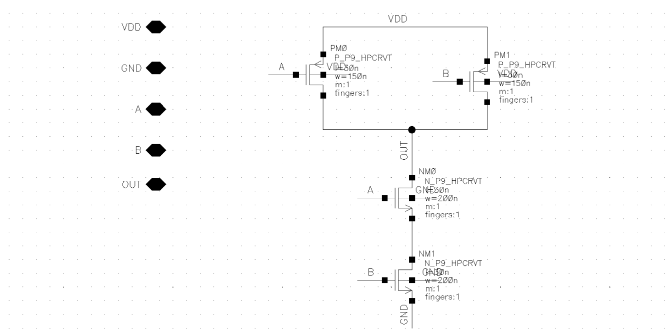

Fig. 12. RO NAND gate

oscillator. The frequency of the ring oscillator is determined by the number of stages, as well as voltage and temperature variations. Activation of the ring oscillator is governed by an enable signal directed to the NAND gate within the oscillator. The oscillations produced by the ring oscillator serve as the clock signal for a 4-bit asynchronous counter, operating as a converter. The random pulse acquired from the XOR gate acts as the enable signal for the ring oscillator, generating a sequence of random cycles that, in turn, produces a random number through the counter. Thus producing a 4-bit random number using the random pulse generated due to the discharging of the read bitline.

Ring Oscillator: A conventional ring oscillator is employed to generate the clock signal for the counter. The oscillator comprises seven stages, with one stage being a NAND gate and the remaining six stages being simple inverters arranged in a cascaded configuration. It is essential to maintain an odd number of stages for proper clock generation. Predominantly, the frequency of the clock depends on the number of stages in the ring oscillator, along with temperature and supply voltage variations. Specifically, we have chosen seven stages to ensure sufficient time to avoid any timing violations in the counter. The NAND gate within the oscillator incorporates an enable signal from the XOR gate, allowing the ring oscillator to be activated only when enabled. The NAND gate that is used is a normal NAND gate, having minimum PMOS widths and NMOS widths of 150 nm and 200 nm, respectively. The Ring oscillator circuit along with the NAND gate used is represented in Figures 11 and 12 respectively

4-b asynchronous counter: The 4-bit asynchronous counter generates a 4-bit random number by utilizing the clock signal obtained from the ring oscillator. Additionally, the counter includes a reset signal that, upon activation, swiftly resets the output, classifying it as an asynchronous counter due to its capacity to reset autonomously, irrespective of the clock signal. Figure 13 represents the counter circuit The Figure 14 represents the circuitry for the **D Latch** used in the counter, comprising cross-coupled NAND gates, an inverter, and clk serving as an input. The NAND gate used, is made stronger to drive the further components by setting the PMOS width and the NMOS widths at 300 nm and 400 nm, respectively. These widths are chosen to minimize the delay between input and output signals and, at the same time ensuring that the increase in width does not lead to a significant expansion in the overall area occupied by the gate.

The D-latches are configured in a Master-slave combination to construct a flip-flop. Transmission gates of minimum sizes are used for passing the values of reset and input. Four flip-flops are joined to make the 4-bit counter. All the inverters used in the asynchronous counter circuit are minimum-size inverters, with Pmos of size 150nm and NMOS of size 100nm. Figure 15 represents the circuit for Flip-flop.

The counter generates a 4 bit random value based on the random pulse given to the ring oscillator, which is formed due

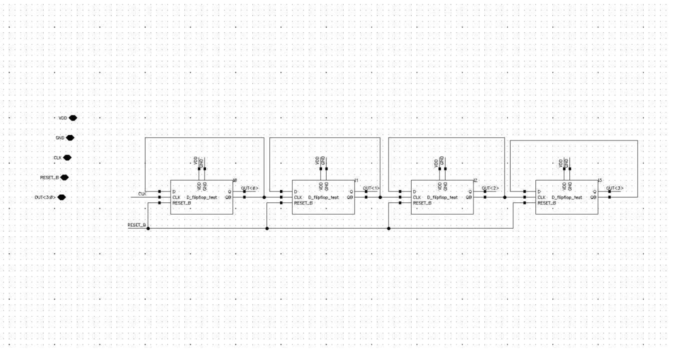

Fig. 13. 4-b asynchronous counter circuit schematic

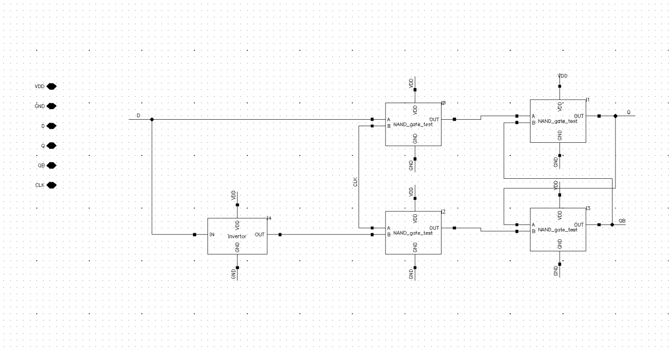

Fig. 14. D-latch

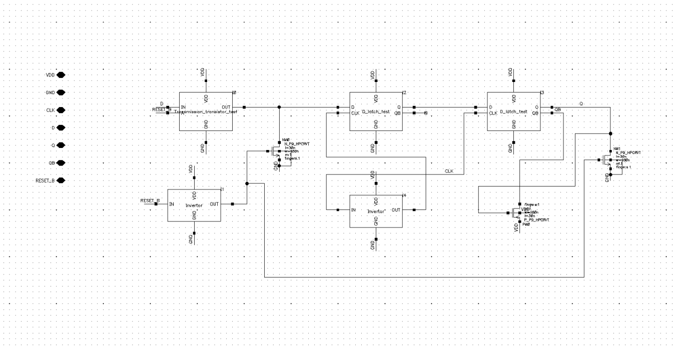

Fig. 15. D Flip-flop

to the dynamic entropy in the circuit because of the leakage and noise contributions.

C. Physically Unclonable Function(PUF)

1) Entropy Generation and Digitization of the Entropy: To harness the unique identification capabilities of an SRAM cell for physically unclonable function (PUF) generation, it is crucial to minimize the impact of noise on the SRAM source. Instead, the primary factor influencing the PUF characteristics should be the inherent internal variations within the chip arising from process variations. These variations, stemming from imperfections in the manufacturing process, result in subtle discrepancies in the electrical characteristics of individual SRAM cells, creating a unique cryptographic signature for each chip. This can be accomplished by eliminating the variations due to voltage and temperature fluctuations and random noises in the memory. For this, we use the read current from two different columns. We precharge the bitlines, and any one wordline is activated, and the bitline discharge time difference is evaluated in two adjacent cells. The time difference is independent of the common process variations as well as temperature and voltage fluctuations. This mechanism can be used to generate multiple PUF bits by a simple mathematical process. In this project, we have shown the generation of two PUF bits. Multi-bit PUF is generated by digitizing the time difference in discharge rate between the two cells. The digitization is done through a delay-based TDC, made of delay-inducing buffers and time-arbiters. The two bits are generated

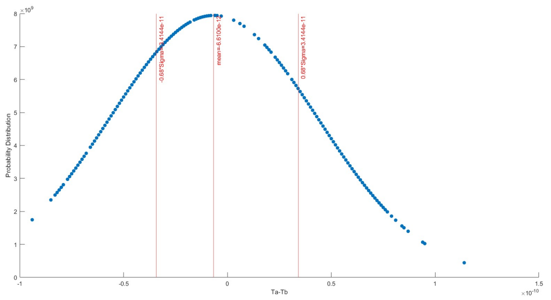

Fig. 16. Gaussian distribution of bitline discharge rate difference

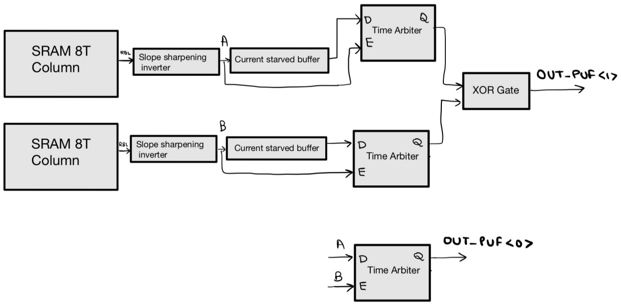

Fig. 17. PUF Peripheral Circuitry

by dividing the variation in tA - tB into 4 regions. We slice the probability distribution function of the tA - tB into four regions to digitize the difference. The LSB of PUF is decided by comparing it with 0. The MSB is decided on the basis of comparing the difference, tA - tB, to specific delay thresholds. We take the point +/- 0.68σ as the delay thresholds, which divides the Gaussian into four regions, digitizing it to a 2-bit number. For PUF[1], the delay lines of 0.68σ are implemented using a delay-inducing buffer and subsequently given to time arbiters, where it is assigned to 0 if it is inside the Gaussian lobe, i.e. between the two delay thresholds and to 1 otherwise. The threshold points chosen are suitable for passing the test and proper entropy generation. Figure 16 shows the Gaussian distribution of the time difference between two adjacent cells.

2) Digitization circuit:

Time Arbiter: It is a cross-coupled NAND gate designed to assess the time difference between two signals. Its role includes generating the Least Significant Bit (LSB) of the PUF output bit and serving as one of the stages in determining the Most Significant Bit (MSB) of the PUF output bit. Figure 18 represent circuit for time arbiters.

Delay Inducing Buffer: This buffer is utilized to introduce a precise delay in signals A and B. It consists of two inverters, with one of the inverters having multiple pull-up and pull-down transistors adjusted according to the desired delay. The most

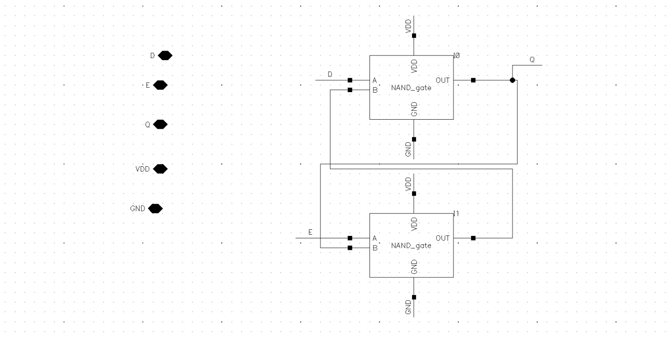

Fig. 18. Time Arbiter

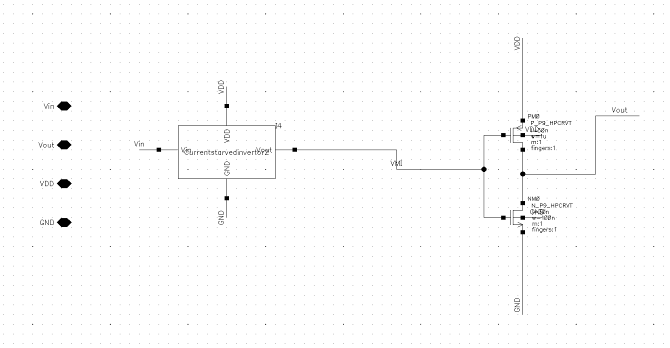

Fig. 19. Delay Buffer

significant bit (MSB) of the PUF output bit is decided by the delay introduced by these buffers. Subsequently, the output from this buffer is directed into the time arbiter. Figure 19 represents circuit for the delay inducing buffer

III. Results

A. *TRNG Results*

Figure 20 shows the waveforms obtained from the TRNG circuit.

B. *PUF Results*

Figure 21 shows the waveforms obtained from the TRNG circuit.

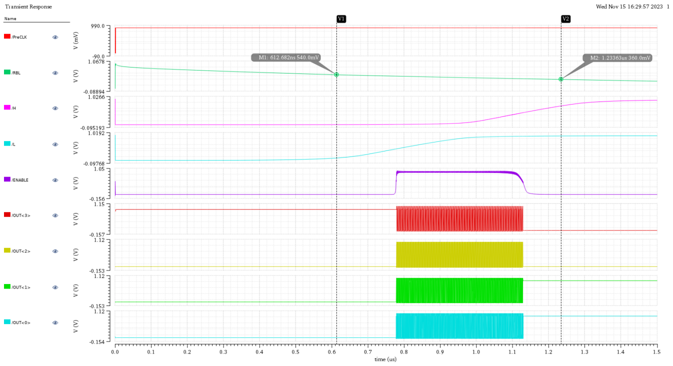

Fig. 20. Result Waveform of TRNG circuit

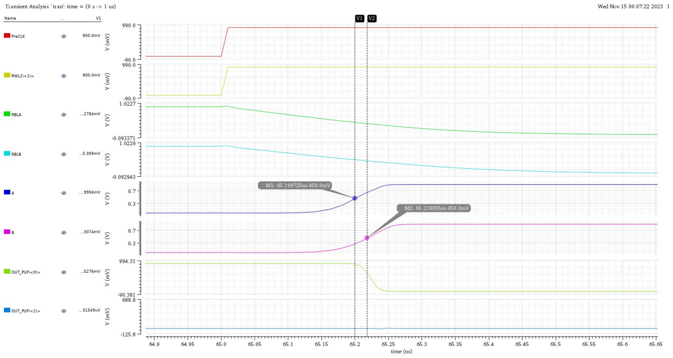

Fig. 21. Result Waveform of PUF circuit

Temperature(C)	Voltage(V)	min-entropy	avg-p-value
-25	1.05	0.8280	0.8299
27	0.9	0.8526	0.4747
100	0.75	0.8527	0.3070

TABLE I
NIST TEST FOR OUT[0]

IV. Statistical Characterization of TRNG

The statistical quality of the output bitstream was performed based on 3000 Monte Carlo Evaluations. The statistical quality of the TRNG bitstream is evaluated using the NIST 800-90B test, by calculating its min-entropy. A min-entropy value ¿0.84 from tests across all bits and all conditions, with some exceptions. It also passes most tests of NIST 800-22b with an average p-value of 0.53 across all conditions. Table I-IV shows min-entropy values and average-p values obtained from both NIST tests for 4 TRNG bitS OUT[0], OUT[1], OUT[2] and OUT[3] respectively. Tables V-VII shows test results of NIST 800-22B

V. Statistical Characterization of PUF

The statistical quality of the output bitstream was performed based on 1500 Monte Carlo Evaluations. The statistical quality of the PUF bitstream is evaluated using the shannon entropy and ber test. It passes both the test. Table X-X1 shows shannon entropy and ber percentage values obtained from the tests for 2 PUF bitS PUF[0], PUF[1].

Temperature(C)	Voltage(V)	min-entropy	avg-p-value
-25	1.05	0.4812	0.6174
27	0.9	0.8688	0.4704
100	0.75	0.8449	0.5117

TABLE II
NIST TEST FOR OUT[1]

Temperature(C)	Voltage(V)	min-entropy	avg-p-value
-25	1.05	0.4466	0.6174
27	0.9	0.8091	0.5499
100	0.75	0.8551	0.6991

TABLE III
NIST TEST FOR OUT[2]

Temperature(C)	Voltage(V)	min-entropy	avg-p-value
-25	1.05	0.7075	0.5629
27	0.9	0.8622	0.3186
100	0.75	0.8637	0.5782

TABLE IV
NIST TEST FOR OUT[3]

monobit test	0.8003	PASS
frequency within block test	0.5216	PASS
runs˙test	0.2286	PASS
longest run ones in a block test	0.2275	PASS
binary matrix rank test	0.2707	PASS
dft test	0.2457	PASS
non overlapping template matching test	0.9993	PASS
overlapping template matching test	0	FAIL
maurers universal test	0	FAIL
linear complexity test	0	FAIL
serial test	0.5485	PASS
approximate entropy test	0.6817	PASS
cumulative sums test	0.7656	PASS
random excursion test	0.0095	FAIL
random excursion variant test	0.0131	PASS

TABLE V
NIST TEST 800-22 FOR OUT[0]

monobit test	0.2549	PASS
frequency within block test	0.0730	PASS
runs˙test	0.2134	PASS
longest run ones in a block test	0.7957	PASS
binary matrix rank test	0.6038	PASS
dft test	0.1468	PASS
non overlapping template matching test	0.9067	PASS
overlapping template matching test	0	FAIL
maurers universal test	0	FAIL
linear complexity test	0	FAIL
serial test	0.5531	PASS
approximate entropy test	0.5795	PASS
cumulative sums test	0.2744	PASS
random excursion test	0.3365	PASS
random excursion variant test	0.1558	PASS

TABLE VI
NIST TEST 800-22 FOR OUT[1]

monobit test	1	PASS
frequency within block test	0.2146	PASS
runs˙test	0.7043	PASS
longest run ones in a block test	0.8893	PASS
binary matrix rank test	0.1156	PASS
dft test	0.7716	PASS
non overlapping template matching test	0.9651	PASS
overlapping template matching test	0	FAIL
maurers universal test	0	FAIL
linear complexity test	0	FAIL
serial test	0.4487	PASS
approximate entropy test	0.6288	PASS
cumulative sums test	0.8231	PASS
random excursion test	0.0008	FAIL
random excursion variant test	0.3959	PASS

TABLE VII
NIST TEST 800-22 FOR OUT[2]

monobit test	0.5279	PASS
frequency within block test	0.0526	PASS
runs˙test	0.3362	PASS
longest run ones in a block test	0.1589	PASS
binary matrix rank test	0.2636	PASS
dft test	0.5616	PASS
non overlapping template matching test	0.3295	PASS
overlapping template matching test	0	FAIL
maurers universal test	0	FAIL
linear complexity test	0	FAIL
serial test	0.5432	PASS
approximate entropy test	0.6054	PASS
cumulative sums test	0.3681	PASS
random excursion test	0.3619	PASS
random excursion variant test	0.2336	PASS

TABLE VIII
NIST TEST 800-22 FOR OUT[3]

Min entropy	PASS
Independance Test Binary	PASS
GOODNESS OF FIT TEST BINARY	PASS
LRS TEST	PASS
IID Permutation test	PASS

TABLE IX
NIST TEST 800-90B FOR OUT[3:0]

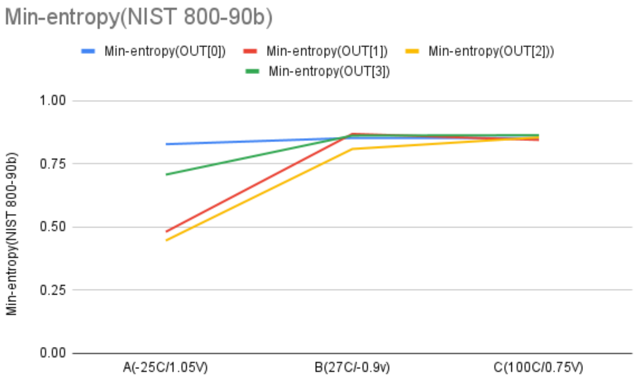

Fig. 23. Min-entropy(NIST 800-90b)

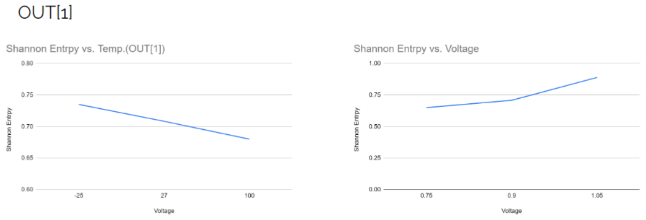

Fig. 24. Shannon Entropy vs. temp and Shannon Entropy vs. voltage for PUF[1]

Voltage	Temperature	Shannon Entropy	BER Percentage
0.75	27	0.9954	10
0.9	-25	1	2
0.9	27	0.9998	0
0.9	100	0.9988	0
1.05	27	1	7.33

TABLE X
STATISTICAL CHARACTERISATION OF PUF[0]

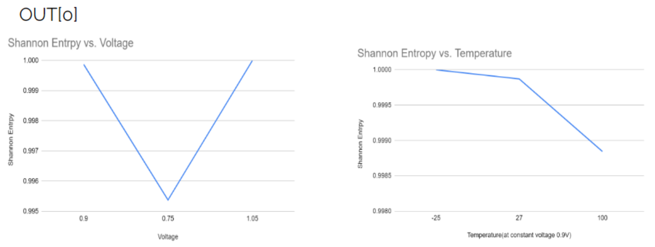

Fig. 25. Shannon Entropy vs. temp and Shannon Entropy vs. voltage for PUF[0]

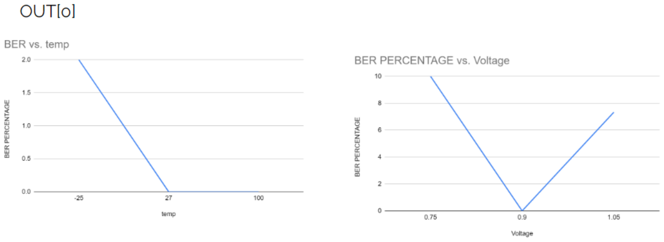

Fig. 26. BER vs. temp and BER vs. voltage for PUF[0]

VI. Conclusion

In this study, we implemented a unified SRAM with both static and dynamic entropy generation, enabling secure key generation. The dynamic entropy generation is based on a true random number generator (TRNG), while the static entropy generation is based on a physically unclonable function (PUF). Both the TRNG and the PUF share the same operating principle and enable extensive circuit reuse across functions. The proposed design also eliminates the need for additional circuitry, reducing the system integration effort and eliminating physical attack points. The unified architecture delivers cryptographic-grade randomness across all operating points under both TRNG and PUF operation. The insensitivity of the entropy against the data pattern stored allows flexible usage of portions of each bank for read/write. In view of the pervasive nature of SRAMs in today's systems on chip, the proposed in-memory unified TRNG and multi-bit PUF makes entropy generation ubiquitous in next-generation systems down to ultra-low-power applications.

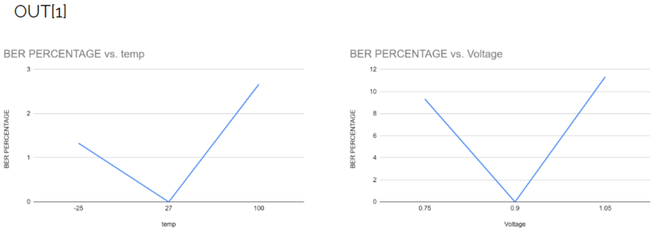

Fig. 27. BER vs. temp and BER vs. voltage for PUF[1]

Voltage	Temperature	Shannon Entropy	BER Percentage
0.75	27	0.6500	9.33
0.9	-25	0.7351	1.33
0.9	27	0.7084	0
0.9	100	0.6800	2.66
1.05	27	0.8893	11.33

TABLE XI
Statistical Characterisation of PUF[1]

References

1. S. Taneja, V. K. Rajanna and M. Alioto, ”In-Memory Unified TRNG and Multi-Bit PUF for Ubiquitous Hardware Security,” in IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 153-166, Jan. 2022, doi: 10.1109/JSSC.2021.3125255.

2. D.Jhonston [Github]https://github.com/usnistgov/SP800-90B˙EntropyAssessment/tree/master/bin(accesssed 13 Nov. 2023)