Securing low power embedded processors for IoT applications against power analysis attacks

A report on

“Integration of RISC-V with Modified Collatz Conjecture Countermeasure for AES IP”

Submitted by:

Prof. Roy Paily Palathinkal & team

Department of Electronics & Electrical Engineering
Indian Institute of Technology, Guwahati

Date: 31/03/2025

INTRODUCTION

The integration of secure cryptographic functions into modern processors is essential for various applications. This report explores the integration of a modified Collatz Conjecture-based countermeasure protected AES (Advanced Encryption Standard) Intellectual Property (IP) on a RV32IM RISC-V processor, aiming to enhance the security of the encryption system. The addition of a UART module facilitates the display and verification of results. The main objectives of this work are to design a seamless interface between the AES IP and the processor, ensuring secure and efficient data handling, as well as to implement a hardware pause technique to manage the processor's operation during AES computations.

The RISC-V RV32IM Processor

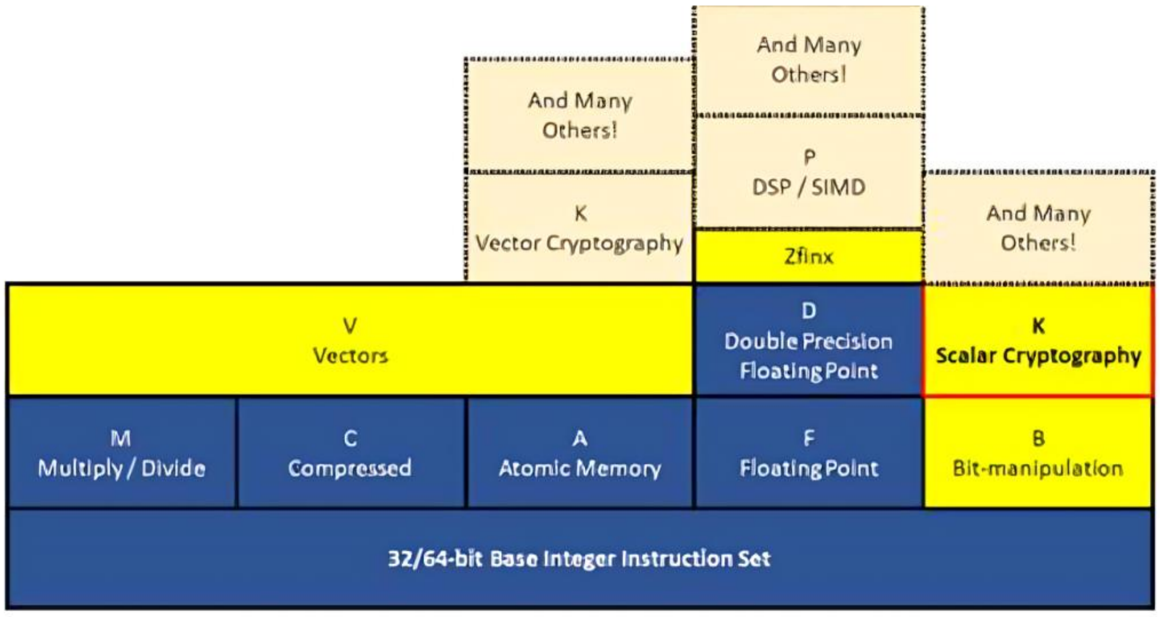


Figure 1: Extension of RISC-V

RISC-V is an open standard Instruction Set Architecture (ISA) based on the principles of Reduced Instruction Set Computing (RISC). The basic RISC-V ISA, as shown Fig 1, is available in 32-bit and 64-bit versions, which are RV32I and RV64I, respectively. The 32-bit processor denotes that its addresses and data are of a 32-bit size. The basic RISC-V ISA can be extended with some additional extensions for multipliers, DSP applications, etc. RV32IM indicates that this ISA supports 32-bit base instructions and multiplication and division instructions. In the base integer instruction RISC-V ISA shown in table 1, different types of instruction formats are available, like R-type, I-type, S-type and U-type. All are of 32-bit fixed length, and they must be positioned in memory in a 4-byte order. If the target address is not 4-byte aligned, an instruction address misalignment exception is raised. These RISC-V instructions can be divided into four different categories as shown Fig 2.

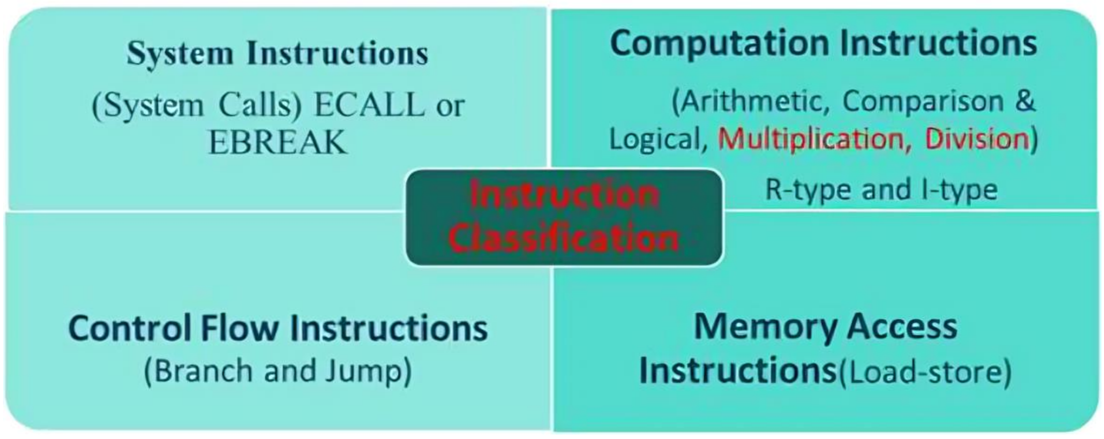

System Instructions
(System Calls) ECALL or EBREAK
Computation Instructions
(Arithmetic, Comparison & Logical, Multiplication, Division)
R-type and I-type
Instruction Classification
Control Flow Instructions
(Branch and Jump)
Memory Access Instructions(Load-store)

Figure 2: Classification of RISC-V instructions

The RV32IM variant includes integer operations, multiplication, and division instructions. The processor employs a pipeline for executing integer instructions, excluding load and store operations. These memory-access instructions are handled separately with an additional memory read/write stage, decoupling them from the main pipeline. To optimize performance and reduce critical path delay, most arithmetic operations are designed for single-cycle execution. However, multiplication and division operations necessitate a multi-cycle approach due to their inherent complexity. The design incorporates a Baugh-Wooley multiplier optimized to complete its operation within two clock cycles.

Figure. 3 illustrates the micro-architecture of a high-performance 4-stage pipelined RISC-V (RV32IM) processor. The first stage, Instruction Fetch (IF), fetch instructions and updates the program counter (PC). It employs a 2-bit Dynamic Branch Predictor (DBP) with a Branch History Table (BHT) to track branching behavior, and make predictions assuming future behaviour will be the same. If the prediction goes wrong, it updates the BHT with new values and stalls the pipeline all the way while re-fetching the instructions. The second stage, Instruction Decode (ID), includes a decoder and a register bank with dual read/write ports to prevent stalls while multiple register writes occurs and thereby enhance performance. The third stage, Instruction Execute (IE), consists of an ALU for arithmetic, logic, and address calculations, a Baugh-Wooley multiplier, a radix-16 divider, comparators, barrel shifters, a data controller, CSR and a branch unit. The ALU is also used for memory read/write address and branch target calculations, optimizing resource utilization. The final stage, Register Write (RW), writes execution results or memory-loaded data back to registers by load-store unit.

The Baugh-Wooley multiplier, shown in Figure. 4, is a specialized circuit designed to efficiently multiply two signed numbers in two's complement format. This architecture is particularly efficient in handling the sign bits during the multiplication process. By employing a specific arrangement of partial products and carry-save adders, the Baugh-Wooley multiplier can produce the correct product in a relatively short amount of time. This makes it a suitable choice for high-performance digital systems where multiplication speed is critical

.

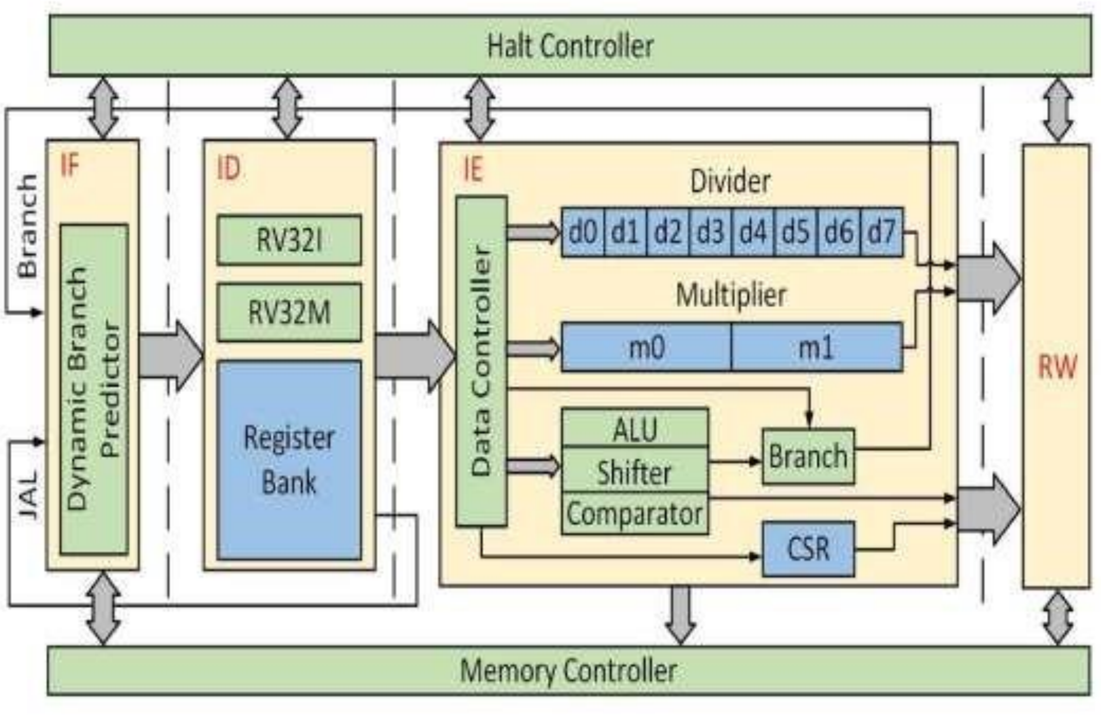


Figure 3: Proposed Micro-Architecture of 4-stage pipelined RISC-V (RV32IM)

Key characteristics of the Baugh-Wooley multiplier:

Handles signed numbers: Unlike traditional array multipliers, it can directly multiply signed numbers in two's complement form.

Efficient: It often provides a good balance between speed, area, and power consumption compared to other multiplier architectures.

Regular structure: The algorithm lends itself to a regular and modular implementation, making it suitable for hardware design.

Key Steps

1. **Partial Product Generation:**
   - The multiplicand is multiplied with each bit of the multiplier, similar to traditional multiplication.
   - However, the signs of the partial products are adjusted based on a predetermined pattern to account for the two's complement representation.
2. **Partial Product Reduction:**
   - The generated partial products are added together using carry-save adders to reduce the number of bits in each stage.
   - This step is critical for improving the multiplier's speed.

3. **Final Addition:**
   - The reduced partial products are added together to form the final product.
   - Depending on the implementation, this can be done using a carry-lookahead adder or a ripple-carry adder.

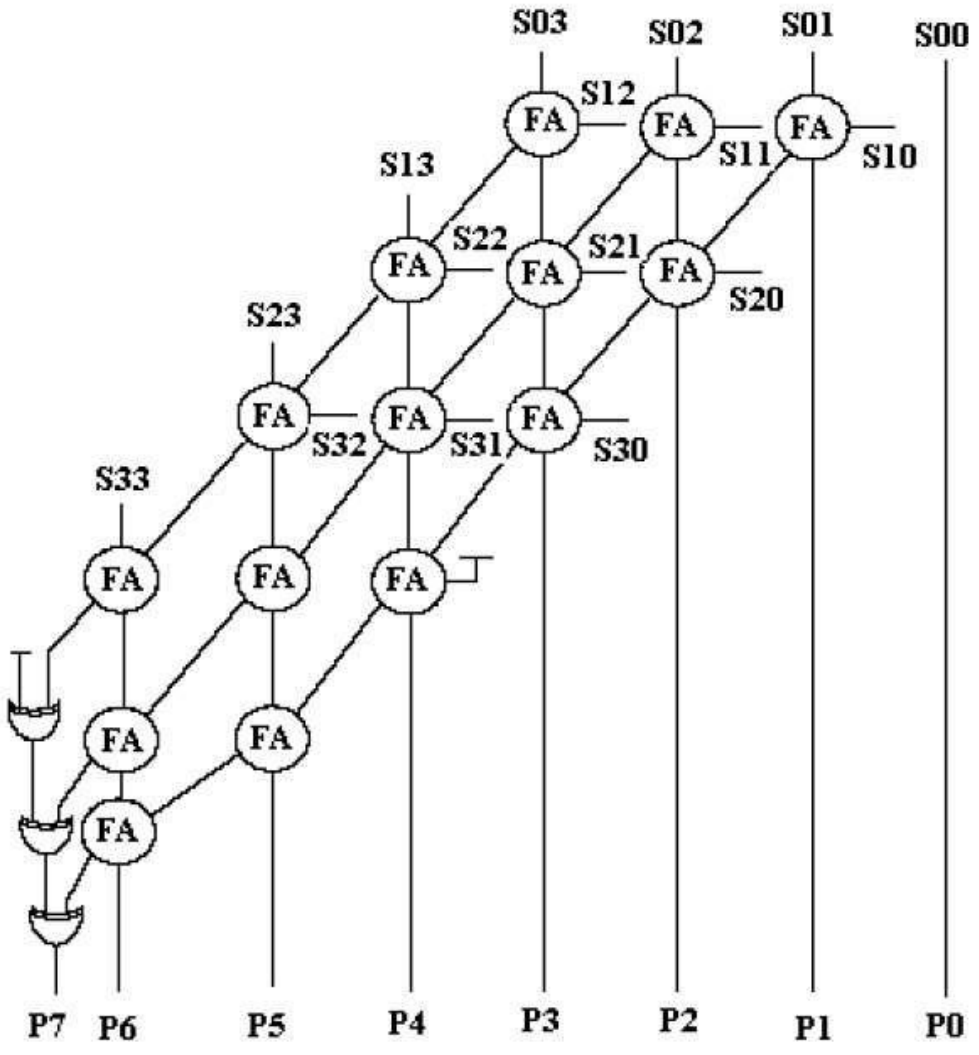


Figure 4: Baugh-Wooley multiplier

Advantages:

- Efficient handling of signed numbers
- Regular structure for hardware implementation
- Good balance of speed, area, and power
- Additionally, a 32-bit radix-16 divider has been implemented to execute division tasks in eight clock cycles.

AES IP Integration

AES IP stands for **Advanced Encryption Standard Intellectual Property**. It's essentially a hardware implementation of the AES encryption algorithm. This IP core is integrated into various devices and systems to provide robust data encryption and security. A sequential AES design implemented in Verilog operates in 12 clock cycles, completing one round per cycle. This design balances throughput and area, making it suitable for hardware implementations. Sequential designs mitigate the high critical path delay of combinational designs and enhance overall throughput by pipelining the rounds. AES is a symmetric block cipher which works on a 128-bit data at the sender's and receiver's terminal. It is a type of Substitution Permutation Network (SPN) implementable in 10, 12 and 14 rounds for key lengths of 128-bit, 192-bit and 256-bit, respectively. The scope of this thesis limits to AES-128, implying a 128-bit key length. The PN is facilitated by four round operations which involve AddRoundKey performing the whitening step, SubBytes fulfilling the substitution step, and ShiftRows and MixColumns accomplishing the permutation step. For an AES adopting a 10-round strategy for encryption, a remarkable feature is that the last round is bereft of the MixColumns step. The AES flow mechanism is represented in Figure. 5, with its round operations and key scheduling operations, described as follows.

Round operations

The round operations are performed on the original plaintext to eventually transform it to an unrelatable ciphertext. The deeper the plaintext gets into the algorithm, or the more the round operations act upon it, the lesser is its correlation between the intermediate round outputs and the plaintext. The 128-bit plaintext is stored in the form of a 4×4 state matrix with each matrix element representing a byte. A four-step strategy is adopted as a part of the round operation: -

1. SubBytes

An acronym for Substitution of Bytes, the SubBytes step, as the name suggests, substitutes each byte of the plaintext with another predecided byte. This operation is performed on the basis of a Substitution Box (S-box) which has precomputed alternative for every possible value of a byte. The S-box entries have been calculated using Galois Field (GF) mathematics to provide nonlinearity to the SubBytes operation. This step involves 16 S-box operations owing to the 16 bytes in state matrix.

2. ShiftRows

The AES round operation is a permutation step which cyclically shifts the rows of the state matrix. Leaving the first row unchanged, the second row is shifted right by one place, the third row is shifted by two, and the fourth row is shifted by three positions.

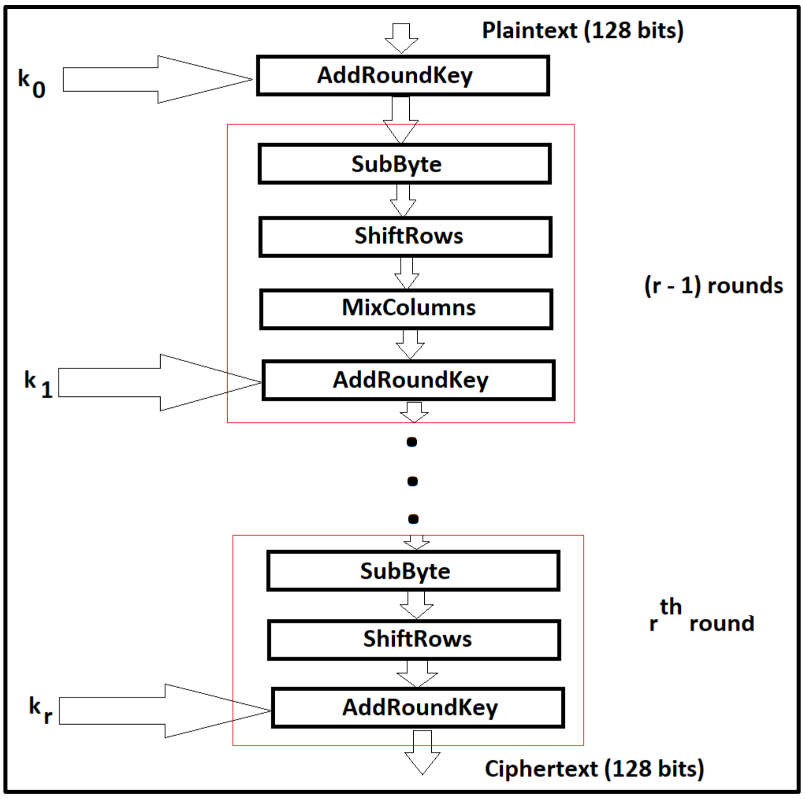


Figure 5: Advanced Encryption Standard (AES) design flow

3. **MixColumns**

This is the second permutation step with operations on the columns of the state matrix. The elements of the column in the matrix is considered as a four-term polynomial over GF(28) and multiplied with a fixed polynomial,

a(x) = 03x 3 + 01x 2 + 01x + 02 --------------------- (1)

In order to restrict the degree of the resultant polynomial to 3, a multiplication modulus of (x 4 + 1) is used. The MixColumns operation can be represented as:

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

4. **AddRoundKey**

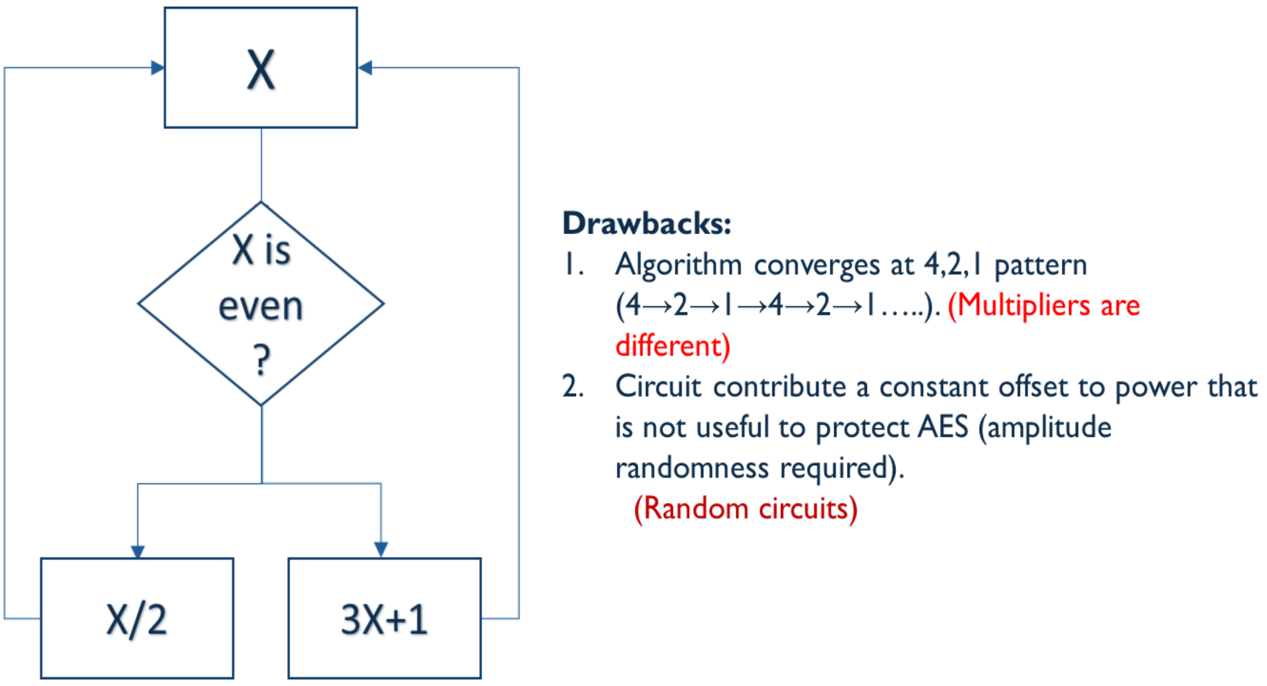
This is the first step of an AES algorithm which involves XORing of the plaintext with the key generated for each round from secret cipher key using key scheduler.

Vulnerability of AES

Despite the valiant efforts in the AES algorithm to secure data, it still remains vulnerable. IoT edge devices employing AES are an easy target of side-channel attackers. Out of the various side-channel parameters, this work focusses on ‘power leakage’, hence the attacks are called “Power Analysis Attacks (PAA)”. As the attackers intend to study the power information liberated by the cryptographic operations in the device, SubBytes, the most power intensive step of AES is the target. The SubBytes executes 16 S-boxes to substitute 16 bytes of data, resulting in 75% of the total AES power consumption. The attackers take hints from this AES round operation where majority of the power spending is concentrated. The objective of the attacker is to retrieve the secret cipher key employed in the algorithm, obtaining which reverse engineering can be easily performed to obtain the original messages (plaintexts). To address this vulnerability to PAA, a small countermeasure based on Modified Collatz Conjecture is designed.

Modified Collatz Conjecture Based Countermeasure

For the countermeasure design, we need a random power generating source to disrupt the AES power trace patterns. This is crucial because the correlation of AES power patterns with the AES operations performed and data used for encryption is most significant part of all Correlation Power Analysis (CPA) attacks. We designed modified Collatz conjecture for this purpose. The modified Collatz conjecture is based on the famous unsolved famous mathematical problem.



Drawbacks:

1. Algorithm converges at 4,2,1 pattern (4→2→1→4→2→1…..). (Multipliers are different)
2. Circuit contribute a constant offset to power that is not useful to protect AES (amplitude randomness required).
   (Random circuits)

Figure 6: Collatz conjecture without modification

The Collatz conjecture queries whether repeatedly applying two straightforward arithmetic operations will ultimately lead every positive integer to 1. Named after mathematician Lothar Collatz, who proposed it in 1937, the conjecture has been validated for all positive integers up to 2.95 × 1020. However, a comprehensive proof remains elusive. In the sequence, if the previous term is even, then the next term is the half of the previous term, else it is 1 plus 3 times previous number. This is called collatz sequence. The Collatz conjecture is: *Regardless of the initial positive integer* *selected, this process will ultimately reach the number 1.* If the hypothesis is incorrect, it can only mean that there is a starting number that results in a sequence that doesn't include 1. Such a sequence would either grow infinitely or enter a repeating cycle that does not include 1. There isn't a sequence like that. That's why it's a conjecture. This happens because, in sequence, always a number that is power of 2 occurs which divides down to 1 and then goes into the cycle. It has two drawbacks: first, the cycle 4,2,1 is evident; second, if its corresponding circuit produces a power signature primarily due to addition operations, it can be compared to a constant power offset to the AES power, which is easily nullified during statistical testing. And secret key will be easily revealed using PAA

Modified Collatz Conjecture

The main modification of the proposed modified Collatz Conjecture countermeasure design, shown in Figure.7, is that we have merged 3 more sequences with the Collatz sequence that are similar to it but just differ by the coefficient in the case where the previous term is odd. We can infer that these coefficients are 4, 5 and 6. Logically, multiplication with 4 is 2 times left shift, 5 is 2 times left shift with an addition operation with itself and 6 is an addition of 2 different left shifted operands (2x + 4x). They will have 4 different circuits and which operation to choose is decided by an intermediate value of each round of the AES design. Hence, these different circuits on hardware will disrupt AES power traces by different amounts solving both of our problems of constant offset and ending with 4,2,1 sequence repetition

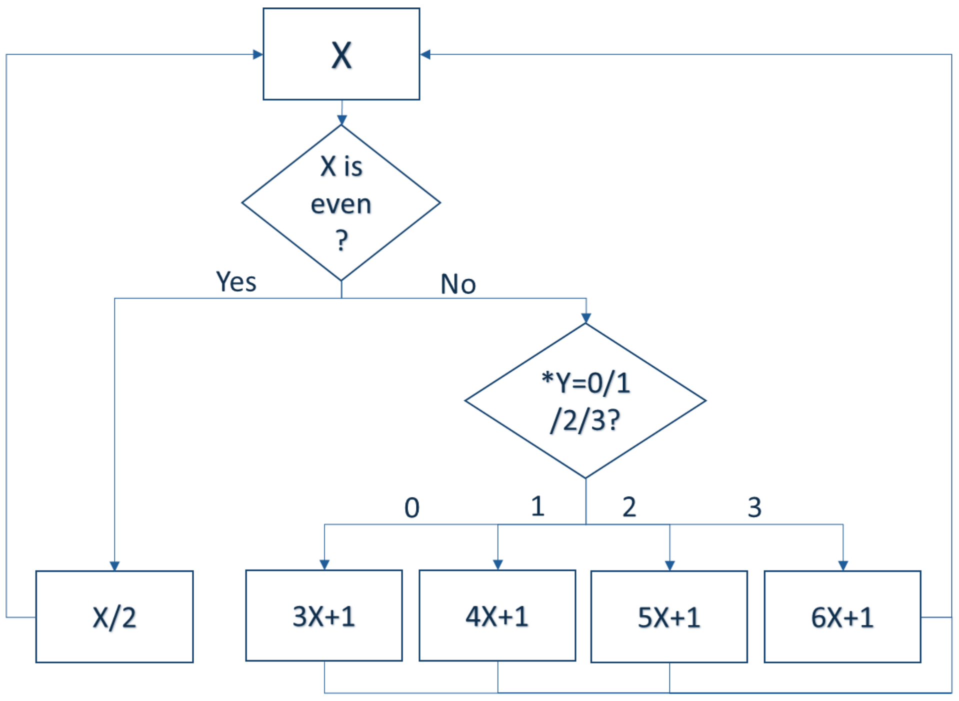


Figure 7: Proposed modified Collatz conjecture

Integration of RISC-V With Modified Collatz Conjecture Countermeasure Protected AES IP

This work integrates the modified Collatz Conjecture countermeasure integrated AES with RISC-V as an address-enabled hardware accelerator IP as shown in Figure.8. This integration of AES hardware IP reduces the number of clock cycles, energy consumption, and encryption time compared to the software AES implementation on RISC-V. The AES module is enabled when the CPU's data address is 32'h00000030. Given the 32-bit CPU and 128-bit AES plaintext, the input is provided over four clock cycles. Addresses 30h, 34h, 38h, and 3ch correspond to the four 32-bit segments of the AES plaintext. The drdy and krdy signals indicate readiness for operation. After 42 clock cycles, the AES outputs the ciphertext and the Dvld signal. The ciphertext is stored in the D Cache at addresses 50h, 54h, 58h, and 5ch.

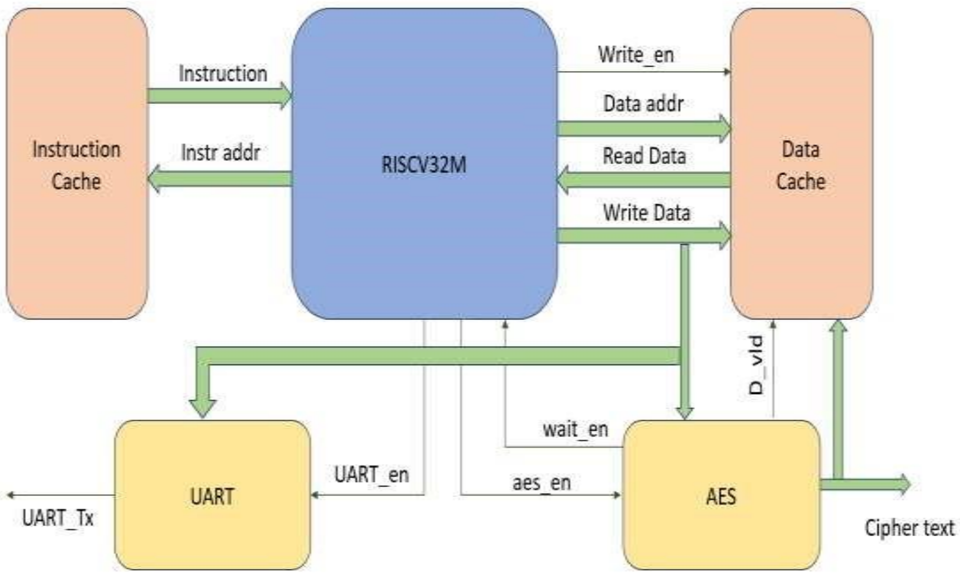


Figure 8: The RV32IM-AES-Integration

UART (Universal Asynchronous Receiver/Transmitter) is a simple yet widely used protocol for serial communication between two devices, essential for debugging and result verification in embedded systems. Unlike synchronous communication, UART operates asynchronously, using start and stop bits instead of a shared clock signal to synchronize data transmission. It transmits data serially, one bit at a time, requiring only two wires: TX (transmit) and RX (receive). The UART module is activated when the CPU's data address is 32'h000001f0, while for other addresses, the Data Cache (D Cache) is enabled.

The hardware pause technique pauses the CPU during AES operation. The wait_enable signal is set high at the start of AES computation (address 3ch) and remains high until the ciphertext is stored. The wait_enable signal interacts with the CPU's program counter, halting instruction fetches until AES processing completes.

While AES is running, the RISC-V processor's multiplier and ALU perform random operations on data generated by the AES-based random number generated by the modified Collatz

Conjecture-based countermeasure. This generates additional random power fluctuations along with additional noise power by the modified Collatz Conjecture countermeasure, masking the power consumption patterns of AES and enhancing security. In this design, the AES block is replaced with an AES module integrated with the modified Collatz Conjecture-based countermeasure.

Experimental Results and Verification

Simulation environments were used to verify the integration and functionality of the processor, AES IP, and UART module.

FPGA Results

Table. 1: FPGA Results

Design	LUTs	FF	Delay(ns)	Max.Freq(MHz)
RV32IM	6471	2585	12.841	77.87
RV32IM + AES IP	7907	3121	12.841	77.87

Table 1 highlights the FPGA results. Comparing a 4-stage RV32IM with an AES IP core integrated shows a 22.22% increase in LUTs and a 20.73% increase in flip-flops. Notably, the delay and frequency remain unaffected by this integration.

ASIC Results

Table. 2: ASIC Results

Design (RISC-V core **)**	**Area(μm^2)**	**Power (**mW)	**Delay(ns)**	**Max. Freq(MHz)**
RV32IM	63199.1	0.294	6.94	144.09
RV32IM + AES IP	80967.6	0.341	6.94	144.09
RV32IM + AES IP + CM	80621.3	0.357	6.94	144.09

Comparing a 4-stage RV32IM with an AES IP core integrated in Table2: shows a 0.42% decrease in area and a 4.69% increase in power. Notably, the delay and frequency remain unaffected by this integration.

In a 12-clock AES design, power leakage primarily comes from the 16 S-boxes running in parallel, which occupy a large area and contribute significantly to leakage power. To address this, we redesigned AES with a 42-clock cycle, reducing the number of S-boxes running in parallel to 4. This not only decreases the area but also allows for a smaller Collatz Conjecture-based countermeasure to effectively mask AES power consumption with lower area overhead. As a result, the Table.2 highlights the Modified Collatz Conjecture-secured RISC-V achieves a negative area overhead of **x%** compared to an unprotected 12-clock AES-integrated RISC-V, with a power overhead of **y%**. The countermeasure does not impact the critical path delay or maximum operating frequency. Additionally, based on the last two bits of the random number generated by the Modified Collatz Conjecture, 1 to 4 false clock cycles are randomly inserted in each AES round operations. This variability makes it difficult for an attacker to determine the exact timing of each round operation, resulting in each plaintext being encrypted with a different number of cycles and providing variable throughput.

Power Traces Pattern

Figure9.a illustrates the power traces of an unprotected AES, where distinct power patterns are visible for each plaintext since the leakage power is solely due to AES encryption. While integrating the AES IP accelerator with RISC-V, additional noise power from RISC-V partially obscures the AES power, as shown in Figure 9.b, though AES power traces remain slightly visible. In contrast, Figure 9.c presents the power traces of the Modified Collatz Conjecture countermeasure-secured RISC-V. The results demonstrate the effectiveness of the proposed countermeasure, as the AES power pattern is completely altered, with peaks spread across the time axis. The combined effect of the countermeasure power and RISC-V noise power successfully masks the AES power traces, significantly enhancing security.

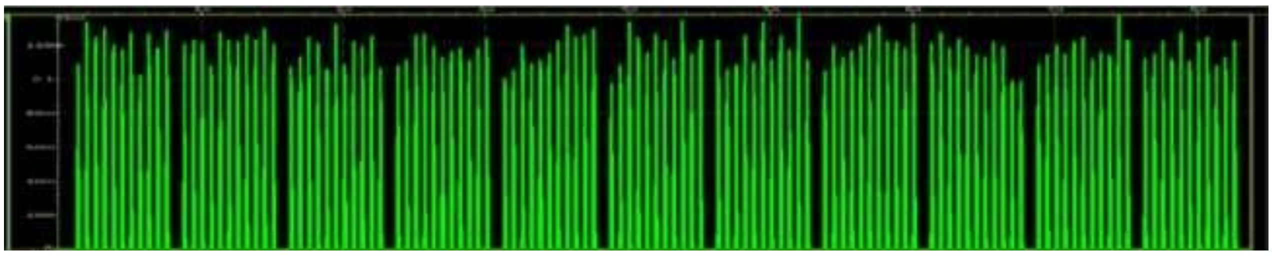


Figure 9.a: 12-clock AES (unprotected)

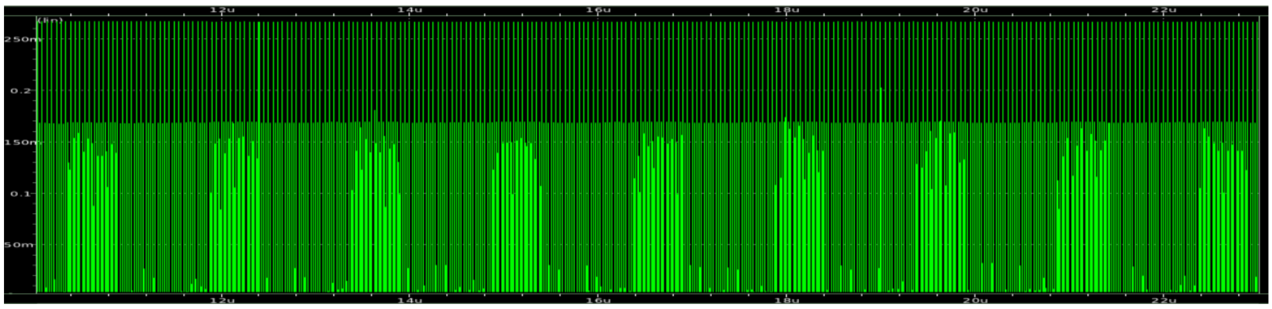


Figure 9.b: RV32IM + 12-Clock AES IP (unprotected)

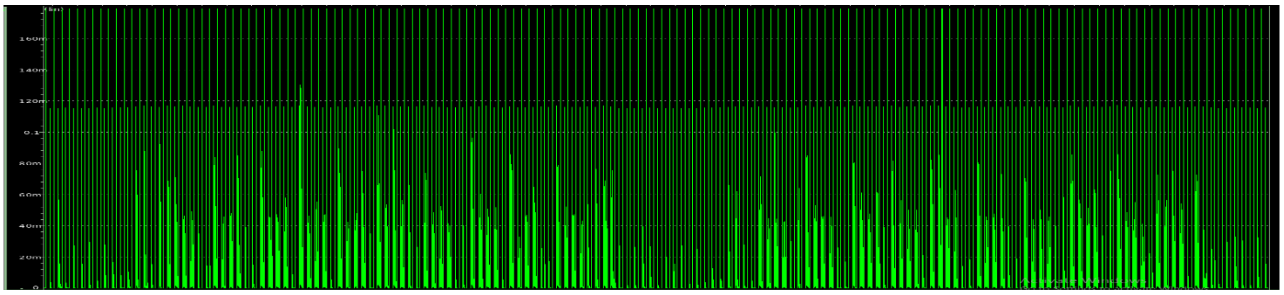


Figure 9. c: Modified Collatz Conjecture countermeasure-secured RISC-V

Power Analysis Attack

Power Analysis Attack (PAA) is a technique used to analyze power consumption to extract cryptographic keys. Correlation Power Analysis is a specific type of PAA that uses statistical methods to find correlations between power consumption and key data. The security level of an IoT edge device under PAA can be judged on the basis of quantitative and qualitative evaluations of some hardware security metrics. The judgement can be passed on the basis of key recovery attacks and information leakage assessments. The former category is an attempt to retrieve the secret cipher key embedded in the device, whereas the latter is a quantitative analysis, if the information siphoned off the device by an attacker is sufficient to label the device as secure or not. Measurements To Disclose (MTD) is a key recovery scheme, and Signal-to-Noise Ratio (SNR) falls in both the categories. Each metric is explained as under.

Measurements To Disclose (MTD)

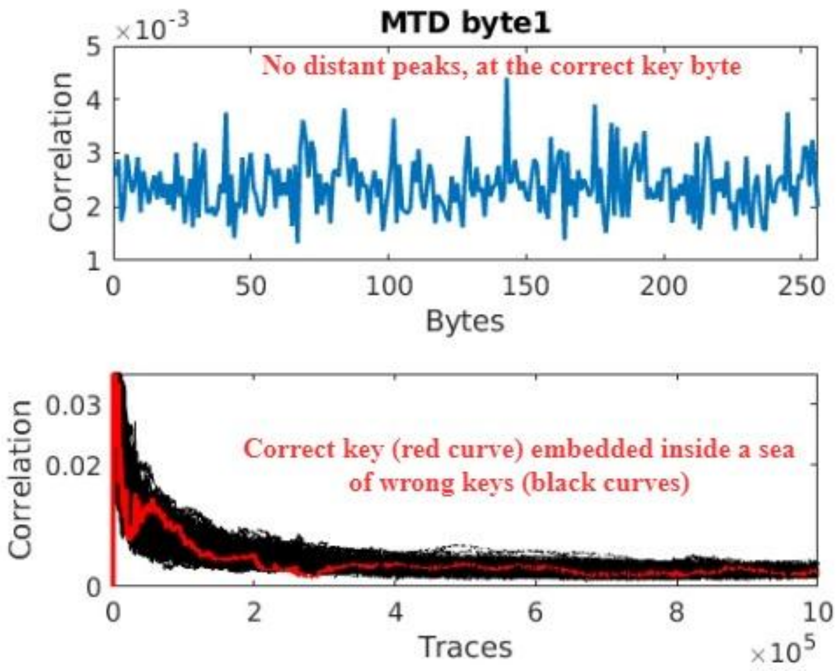


Figure 10: MTD for 1 million traces for modified Collatz Conjecture protected AES countermeasure-secured RISC-V

The number of plaintexts required to recover the key refers to the amount of plaintext data needed for the attack to successfully determine the encryption key. In the MTD plot of the unprotected design, the correct key byte (red line) stands out distinctly from the incorrect keys (sea of black lines), with a distant peak in the correlation coefficient graph at correct key byte, indicating key recovery. Figure 10 presents the MTD for 1 million traces of the Modified Collatz Conjecture countermeasure-protected AES IP integrated with RISC-V. The blue graph shows that the correlation coefficient is insufficiently high to form distant peaks for the correct key. Similarly, the second graph demonstrates strong security, as the correct key byte (red line) remains indistinguishable from incorrect guesses, effectively preventing key recovery (sea of black lines).

Signal-to-noise ratio (SNR)

In the context of Power Analysis Attacks (PAAs), the Signal-to-Noise Ratio (SNR) measures the ratio of signal to noise in a given measurement. A higher SNR indicates easier detection of signal components within noise, implying higher leakage in cryptographic design.

$$SNR = \frac{\sigma^2\ signal}{\sigma^2\ \ noise} \qquad \text{----------------------------(2)}$$

In power analysis attacks, the exploitable power consumption P$_{exp}$, consists of data-dependent power *Pdata* and operation-dependent power Pop, which contain relevant information for an attacker. Additionally, power traces include switching noise *Psw.noise*, and electrical noise *Pel:.noise* which are not exploitable in an attack scenario.

$$\frac{\sigma^2\ Pexp}{\sigma^2\ \ Psw{:}noise + \sigma^2\ \ Pel{:}noise} \qquad \text{----------------------------(3)}$$

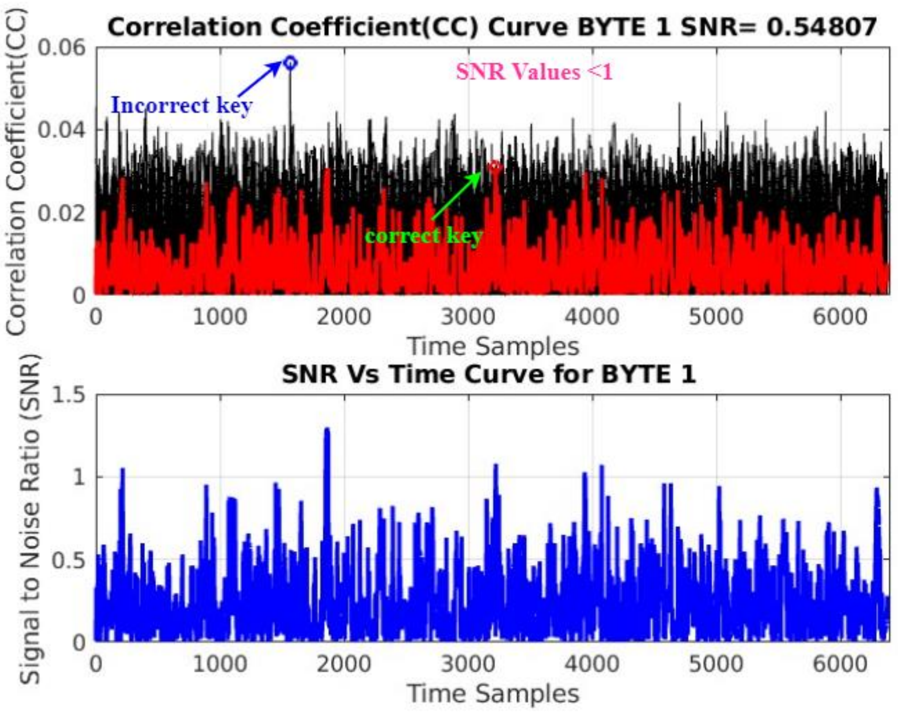


Figure 11: SNR for modified Collatz Conjecture protected AES secured RISC-V

The SNR plot for unprotected designs shows the correct key (marked with a red circle) exhibits the highest correlation, surpassing incorrect key guesses (marked with a blue circle), with SNR values greater than 1, indicating that the design is susceptible to leakage.

Figure 11 illustrates the SNR for the Modified Collatz Conjecture countermeasure-protected AES integrated with RISC-V. The results demonstrate the effectiveness of this countermeasure against power analysis attacks. The graph shows that incorrect key guesses (blue circle) exhibit the highest correlation, surpassing that of the correct key (red circle), with SNR values below 1, the design effectively reduces leakage, enhancing security.

Optimized Layout of Modified Collatz Conjecture Protected AES Secured RISC-V

The design is developed in ASICs using UMC 65nm technology. After functional verification of the RTL code in Vivado, it is synthesized into a gate-level netlist using Synopsys Design Compiler (DC). The synthesized netlist then undergoes physical design step including floor-planning, placement, and routing using Cadence Innovus. Before generating the final post-layout netlist, timing analysis is conducted to ensure zero setup and hold violations, as shown in Figure 13.b. Additionally, a Design Rule Check (DRC) and connectivity issues are performed to confirm that the optimized layout of the Modified Collatz Conjecture countermeasure-protected AES-secured RISC-V, shown in Figure 12, meets manufacturing constraints. The final optimized layout has no timing violations (setup and hold), DRC errors, or connectivity issues as in Figure 13.b. Post-layout simulation is carried out using Synopsys VCS with the optimized netlist and power analysis is performed using Synopsys PrimeTime PX. The power traces are sampled at a rate of 1 GSa/s and exported to MATLAB for security metric evaluation.

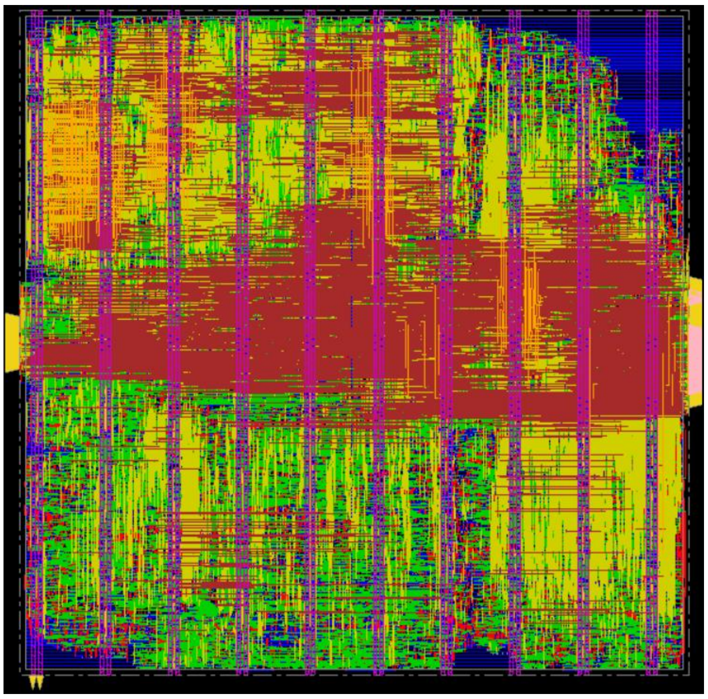
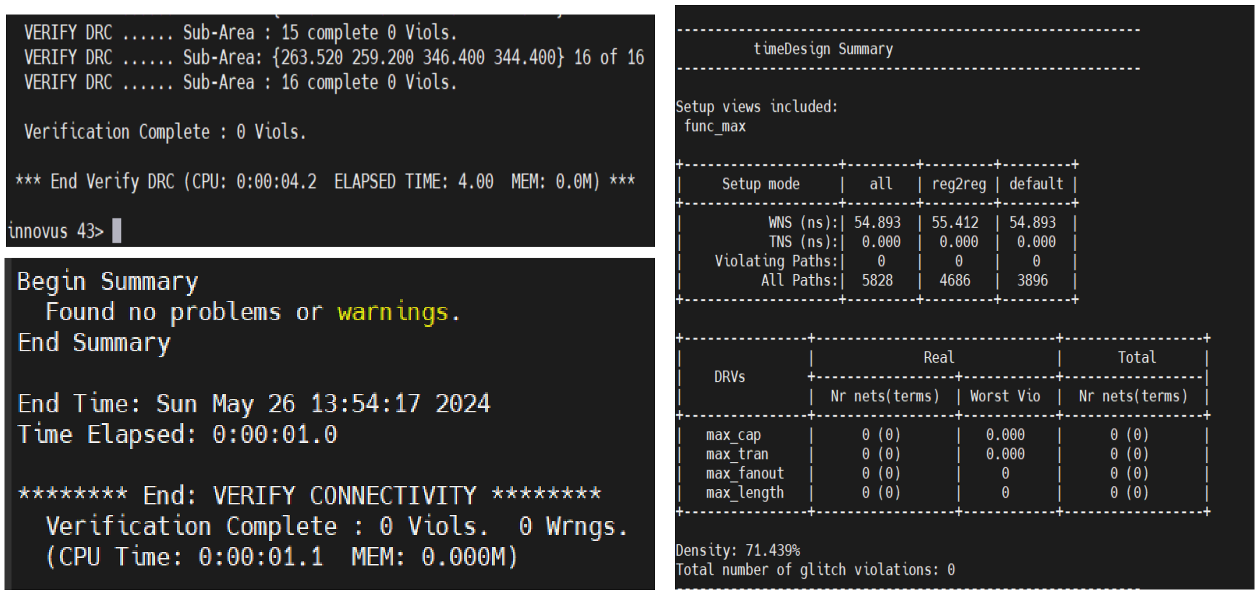


Figure 12: Optimized Layout of modified Collatz Conjecture protected AES secured RISC-V



```
 VERIFY DRC ...... Sub-Area : 15 complete 0 Viols.
 VERIFY DRC ...... Sub-Area: {263.520 259.200 346.400 344.400} 16 of 16
 VERIFY DRC ...... Sub-Area : 16 complete 0 Viols.

 Verification Complete : 0 Viols.

*** End Verify DRC (CPU: 0:00:04.2  ELAPSED TIME: 4.00  MEM: 0.0M) ***

innovus 43> 
```

```
Begin Summary
  Found no problems or warnings.
End Summary

End Time: Sun May 26 13:54:17 2024
Time Elapsed: 0:00:01.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:01.1  MEM: 0.000M)
```

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 func_max

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):| 54.893  | 55.412  | 54.893  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|  5828   |  4686   |  3896   |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |    0 (0)         |   0.000    |     0 (0)        |
|   max_tran     |    0 (0)         |   0.000    |     0 (0)        |
|   max_fanout   |    0 (0)         |     0      |     0 (0)        |
|   max_length   |    0 (0)         |     0      |     0 (0)        |
+----------------+------------------+------------+------------------+

Density: 71.439%
Total number of glitch violations: 0
```

Figure 13: Zero Timing violations, DRC errors, connectivity issues of Optimized Layout

Conclusion

The integration of the RV32IM processor with a modified Collatz Conjecture-based countermeasure in a 12-clock AES IP was successful, with secure and efficient data handling and processing, verified through simulations. This countermeasure enhance security significantly with minimum area and power overheads highly suitable for securing low power battery driven IoTe devices or embedded processors.

References

1. V. Rijmen and J. Daemen, “Advanced Encryption Standard,” Proceedingsof Federal Information Processing Standards Publications, National Institute of Standards and Technology, pp. 19–22, 2001.

2. S. Mangard, E. Oswald, and T. Popp, Power analysis attacks: Revealing thesecrets of smart cards. Springer Science & Business Media, 2008, vol. 31.

3. P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in AnnualInternational Cryptology Conference. Springer, 1999, pp. 388–397.

4. A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic, “The RISCV instruc- tion set manual, volume I: User-level ISA, version 2.0,” EECS Dept., Univ. Cali- fornia at Berkeley, Berkeley, CA, USA, Rep. UCB/EECS-2014-54, May 2014. Available: http://www2.eecs. berkeley.edu/Pubs/TechRpts/2014/EECS-2014-
54.html

5. The RISC-V Instruction Set Manual Volume II: Privileged Architecture Version 1.9.1 Andrew Waterman Yunsup Lee Rimas Avizienis David A. Patterson Krste Asanovi´c Electrical Engineering and Computer Sciences University of California at Berkeley Technical Report No. UCB/EECS2016-161 http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS2016-161.html November 4, 2016

6. Krste Asanovi´c Rimas Avizienis Jonathan Bachrach Scott Beamer David Biancolin Christopher Celio Henry Cook Daniel Dabbelt John Hauser Adam Izraelevitz Sagar Karandikar Ben Keller Donggyu Kim John Koenig Electrical Engineering and Computer Sciences University of California at Berkeley ”The Rocket Chip Generator” Technical Report No. UCB/EECS2016-17 http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html April 15, 2016

7. D. Bellizia, S. Bongiovanni, P. Monsurr, G. Scotti, A. Trifiletti, and F. B. Trotta, “Secure double rate registers as an RTL countermeasure against power analysis attacks,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 7, pp. 1368–1376, 2018.

8. B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, “Mutual information analysis,” in International Workshop on Cryptographic Hardware and Embedded Systems, pp. 426- 442, Springer, 2008.

9. B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi, et al., “A testing methodology for side-channel resistance validation,” in NIST non-invasive attack testing workshop, vol. 7, pp. 115-136, 2011