Circuit and Layout Level Optimization of Logic Locking Circuits to Reduce Area

[1]Mahesh G Shet, [2]Dr. Meghana Kulkarni, [3]Tejswini Gull
[1]PG Scholar, [2]Assistant Professor, [3] Assistant Professor,
[1]Electronics and Communication Engineering,
[1]Visvesvaraya Technological University, Belagavi, Karnataka

Abstract : Hardware security has become a crucial issue in contemporary VLSI design due to the growing complexity of integrated circuits as well as the growing threat of intellectual property (IP) piracy reverse engineering and hardware tampering.
An efficient hardware security method is logic locking which safeguards a circuit by requiring a secret key for proper operation. In this project a transistor-level logic-locking scheme is designed and implemented using Cadence Virtuoso on the C17 benchmark circuit. Initially XOR-based key insertion was used to implement a baseline logic-locked circuit that would only function properly when the valid key was applied.
Using transmission-gate logic structures and traditional CMOS an unoptimized design was created. Transistor-efficient gate structures were then used in an optimized design to minimize hardware overhead. Under a 1 V supply voltage transient analysis was used to simulate and validate both designs. Transistor count power consumption and layout area were used to assess the performance of both the optimized and unoptimized circuits.
By reducing the number of transistors from 60 to 32 the optimized design was able to reduce the hardware complexity by 46. 67 percent. The layout area was reduced from 33. 649 µm² to 20. 90 µm² resulting in a 37. 88 percent area reduction while the maximum power consumption was lowered from 994 µW to 400 µW representing a 59. 76 percent reduction.
Functional verification verified that the circuit generates a tampered output when the wrong key is used and the correct output when the right key is applied. The findings show that the suggested optimized logic-locking method greatly lowers silicon area and power consumption while successfully improving hardware security.
As a result the developed architecture offers a workable and effective solution for low-power secure VLSI applications.

1. INTRODUCTION

1.1 Importance of hardware security in VLSI

Highly complex integrated circuits (ICs) used in contemporary electronic devices like computers cellphones automobiles medical equipment and defense applications have been made possible by the quick development of Very Large Scale Integration (VLSI) technology. Maintaining hardware security has grown in importance as semiconductor manufacturing has developed into a global process involving various design firms fabrication facilities and testing facilities. Protecting integrated circuits and electronic systems from malicious attacks unauthorized modifications reverse engineering intellectual property (IP) theft and hardware Trojans is the main goal of hardware security. Hardware-based attacks can impact the entire system and are more difficult to detect than software vulnerabilities.

Therefore it is crucial to protect hardware throughout the design manufacturing and deployment stages. Intellectual property (IP) piracy or the unlawful duplication or reuse of circuit designs by unauthorized parties is a major concern in VLSI design. This can lower semiconductor companies competitive edge and cause significant financial losses. Hardware Trojans which are malicious modifications added to a circuit to alter its functionality leak private information or cause system failures pose a serious threat as well. Critical infrastructure military and aerospace systems are particularly vulnerable to these attacks. Concern over counterfeit integrated circuits is also on the rise.

Fake or recycled chips could find their way into the supply chain posing a risk to security and reliability. Hardware security techniques support supply chain trust by verifying the legitimacy of ICs. Effective security measures are also required because adversaries can study chip layouts and obtain sensitive design information through reverse engineering attacks. Several hardware security strategies such as Logic Locking IC Camouflaging Split Manufacturing Physical Unclonable Functions (PUFs) and Hardware Trojan Detection techniques have been introduced to address these issues. While IC camouflaging conceals the actual logic implemented on the chip logic locking protects circuits by requiring a secret key for proper operation. For secure key generation and device authentication PUFs offer distinct identifiers. Hardware security is now a basic design requirement in contemporary VLSI systems not an optional feature. Secure hardware protects important intellectual property while guaranteeing data confidentiality system integrity and dependable operation. Hardware security in VLSI will become increasingly important as electronic systems continue to play a significant role in daily life making it a vital area of semiconductor industry research and development.

1.2 What is Logic Locking? Why is it needed?

Logic Locking is a hardware security technique used to protect integrated circuits (ICs) from piracy, reverse engineering, overproduction, and unauthorized use. In this method, additional logic gates called key gates are inserted into a circuit. The circuit functions correctly only when the correct secret key is applied. If an incorrect key is used, the circuit produces wrong outputs or malfunctions.

The correct key is typically programmed by the chip designer or authorized user after fabrication, ensuring that untrusted foundries cannot use or copy the design.

Why is Logic Locking Needed?

1. Protection Against IP Piracy
   - Prevents attackers from copying and reusing proprietary circuit designs.

2. Prevention of IC Overproduction
   - Stops untrusted fabrication facilities from manufacturing extra chips without authorization
   .
3. Defense Against Reverse Engineering
   - Makes it difficult for attackers to understand the circuit functionality by analyzing the chip.

4. Secure Supply Chain
   - Ensures that only authorized parties can activate and use the integrated circuit.

5. Protection from Hardware Attacks
   - Increases resistance against malicious attempts to tamper with or clone the design.

6. Maintains Design Ownership
   - Helps semiconductor companies safeguard their investment in research and development.

Conclusion

Logic Locking is an effective hardware security technique that secures VLSI circuits by requiring a secret key for correct operation. It plays a crucial role in protecting intellectual property, preventing unauthorized chip production, and enhancing trust in the semiconductor supply chain.

2 LITERATURE SURVEY

2.1 Summary of Literature Survey

The literature review focuses on hardware security issues brought on by the globalization of semiconductor manufacturing and design. In contemporary VLSI systems security risks like intellectual property (IP) piracy

reverse engineering IC overbuilding counterfeiting and hardware Trojan insertion have grown to be serious obstacles.

Numerous hardware protection techniques including split manufacturing logic locking watermarking fingerprinting and IC camouflaging have been put forth to reduce these dangers. Logic locking is one of these strategies that has drawn a lot of attention as a successful way to prevent unauthorized use and duplication of integrated circuits.

Adding more key-controlled gates to a circuit design—usually XOR or XNOR gates—is known as logic locking. The circuit only operates properly when the right secret key is supplied otherwise it generates inaccurate outputs. This method helps protect designs from unauthorized chip overproduction reverse engineering and intellectual property piracy. Because of its simple implementation and demonstrated efficacy XOR/XNOR-based logic locking is further highlighted in the literature.

A circuits design can be made more difficult for attackers to decipher comprehend or replicate by strategically placing key gates. Although logic locking may result in additional overhead in terms of area power consumption and performance studies have shown that it can significantly increase hardware security.

Although logic locking provides robust protection optimization is needed to lower the related implementation costs according to the reviewed research. Therefore the focus of this project is to apply XOR-based logic locking to a C17 benchmark circuit and optimize the transistor-level implementation to reduce silicon area and power consumption while maintaining security and proper circuit operation.

2.2 Existing Technologies

Logic locking is a widely used hardware security technique that protects integrated circuits from IP piracy, reverse engineering, and unauthorized overproduction. Existing logic locking methods can be implemented at three levels: logic level, transistor level, and layout level.

At the logic level, security is achieved by inserting key-controlled gates such as XOR/XNOR gates into the circuit netlist. Common techniques include Random Logic Locking (RLL), SARLock, Anti-SAT, and SFLL. These approaches are relatively simple to implement and can provide effective protection, but they often lead to additional area and power overhead.

At the transistor level, logic locking is incorporated directly into CMOS circuits using key-controlled transistors, transmission gates, multiplexers, or threshold-voltage-based structures. This approach generally provides stronger security and increases the difficulty of reverse engineering, although it also adds design complexity.

The work presented in this project falls into this category, where XOR-based logic locking is implemented and optimized at the transistor level using CMOS and Transmission Gate logic.At the layout level, security features are introduced during physical design through methods such as IC camouflaging, dummy contacts, dummy vias, and split manufacturing. These techniques conceal the actual functionality of the circuit and offer strong resistance to reverse-engineering attacks. However, they typically require more area and increase fabrication complexity and manufacturing cost.

In summary, logic-level techniques are easier to implement, transistor-level techniques offer a balance between security and design overhead, and layout-level techniques provide the strongest protection against physical attacks. For this reason, transistor-level optimization of logic locking is a practical approach for improving hardware security while reducing power consumption and layout area.

2.3 Limitations in Existing Technologies

1. Overhead Area.
   - More security circuitry and key gates expand the chips total area.

2. Power usage.
   - More switching activity is introduced by additional logic components which raises power consumption.

3. degradation of performance.
   - Adding key gates may slow down the circuit and increase propagation delay.

4. susceptibility to attacks.
   - A lot of traditional logic locking methods are vulnerable to removal attacks SAT-based attacks and reverse engineering techniques.

5. Design complexity has increased.
   - More design work and verification procedures are needed to implement and verify locked circuits.

6. Overhead in Layout.
   - During physical design security structures may result in bigger layouts and more intricate routing.

7. manufacturing expenses.
   - The complexity and cost of fabrication are increased by advanced protection techniques like split manufacturing and camouflaging.

8. major issues with management.
   - In logic locking systems secure key generation distribution and storage continue to be significant challenges.

2.4 Objectives

1. To study hardware security threats such as IP piracy, reverse engineering, and IC overbuilding.
2. To understand the principles and applications of logic locking in VLSI security.
3. To implement XOR-based logic locking on the C17 benchmark circuit at the transistor level.
4. To design an unoptimized logic-locked circuit using Transmission Gate-based NAND gates and CMOS-based XOR gates.
5. To design an optimized logic-locked circuit using CMOS-based NAND gates and Transmission Gate-based XOR gates.
6. To perform schematic simulation and functional verification of both designs in Cadence Virtuoso.
7. To develop physical layouts for the optimized and unoptimized circuits.
8. To perform DRC (Design Rule Check) and LVS (Layout Versus Schematic) verification.
9. To analyze and compare power consumption and layout area of both implementations.
10. To demonstrate that transistor-level optimization can reduce the overhead introduced by logic locking while maintaining circuit functionality and security.

3 . METHODOLOGY

1. Baseline Circuit Design
The C17 benchmark circuit was selected as the reference circuit and implemented at the schematic level in Cadence Virtuoso. Functional verification was performed through transient simulations, and the initial power consumption and layout area were recorded.

2. Transistor Count Reduction
To improve hardware efficiency, transistor-level optimization was carried out. Conventional gate implementations were replaced with transistor-efficient structures to reduce the overall transistor count while preserving the original functionality of the circuit.

3. Logic Locking Implementation
An XOR-based key gate was inserted into the benchmark circuit to introduce logic locking. A key input (**K**) was added such that the circuit produces the correct output only when the valid key is applied. Any incorrect key results in tampered output behavior.

4. Circuit-Level Optimization
The locked circuit was further optimized by replacing the CMOS-based XOR implementation with a transmission-gate-based XOR structure. This optimization reduced transistor count, power consumption, and hardware overhead without affecting the security functionality.

5. Layout Design
The optimized schematic was converted into a physical layout using Cadence Virtuoso. Standard VLSI layout practices were followed during device placement, routing, and floorplanning.

6. Layout Optimization
Layout optimization techniques such as compact transistor placement, diffusion sharing, and efficient routing were applied to minimize the silicon area and improve layout efficiency.

7. Performance Analysis and Comparison
The unoptimized and optimized designs were compared based on:

- Transistor Count
- Power Consumption
- Layout Area

8. Security and Overhead Analysis
Finally, the effectiveness of the logic-locking technique was evaluated by analyzing the trade-off between security and hardware overhead. The circuit successfully prevented correct operation under an incorrect key while maintaining low power consumption and reduced silicon area.

4 .CIRCUIT DISRICPTION

4.1 C17 Benchmark Circuit

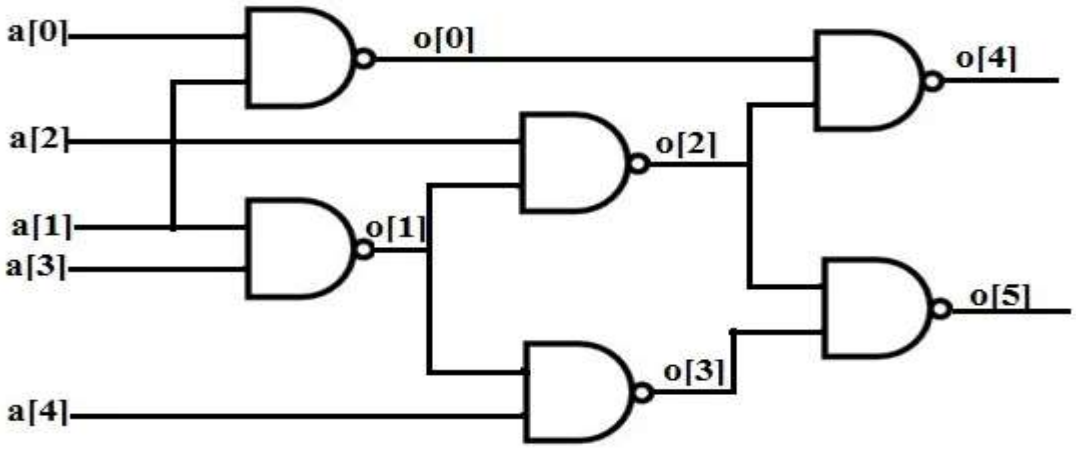

Figure No.1: Logical Circuit of C17 Benchmark

The C17 benchmark circuit is the smallest combinational benchmark circuit from the ISCAS'85 (International Symposium on Circuits and Systems 1985) benchmark suite. It is widely used in digital design, VLSI testing, hardware security, logic locking, fault analysis, ATPG, and optimization research because of its simple structure and well-defined functionality.

Parameter	Value
Benchmark Name	C17
Benchmark Suite	ISCAS'85
Circuit Type	Combinational
Primary Inputs	5
Primary Outputs	2
Logic Gates	6
Gate Type	2-input NAND Gates
Internal Nodes	4

Table No.1: Parameters and values of C17 Bench mark circuit

For the C17 benchmark circuit shown in the figure, the gate-level logical expressions are:
Intermediate Nodes

$$o[0] = \overline{a[0] \cdot a[2]}$$
$$o[1] = \overline{a[1] \cdot a[2]}$$
$$o[2] = \overline{o[0] \cdot o[1]}$$
$$o[3] = \overline{o[1] \cdot a[4]}$$

Final Outputs

$$o[4] = \overline{o[0] \cdot o[2]}$$
$$o[5] = \overline{o[2] \cdot o[3]}$$

Substituting all intermediate terms:

$$\boldsymbol{o[4] = (a[0]\bar{a}[2]) \cdot \left((a[0]\bar{a}[2]) \bar{\cdot} (a[1]\bar{a}[2])\right)}$$

$$\boldsymbol{o[5] = \left((a[0]\bar{a}[2]) \bar{\cdot} (a[1]\bar{a}[2])\right) \cdot \left((a[1]\bar{a}[\bar{2}]) \cdot a[4]\right)}$$

Standard C17 Benchmark Notation:
If inputs are represented as $N1, N2, N3, N6, N7$:

$$N10 = N1\bar{N}3$$
$$N11 = N3\bar{N}6$$
$$N16 = N2\bar{N}11$$
$$N19 = N1\bar{1}N7$$
$$N22 = N10\bar{N}16$$
$$N23 = N16\bar{N}19$$

These are the pure logical expressions typically included in a VLSI/logic-locking project report for the C17 benchmark circuit.

Significance in Hardware Security:

The C17 benchmark is extensively used in hardware security research because:

- Small circuit size simplifies analysis.
- Suitable for implementing and evaluating logic locking techniques.
- Enables measurement of power, delay, and area overhead.
- Useful for testing SAT attacks and key-based security mechanisms.
- Ideal for transistor-level implementation and layout optimization studies.

For the **C17 benchmark circuit**, the truth table (Inputs: a0, a1, a2, a3, a4 and Outputs: o4, o5) is:

a0	a1	a2	a3	a4	o4	o5
0	0	0	0	0	1	1
0	0	0	0	1	1	1
0	0	0	1	0	1	1
0	0	0	1	1	1	1
0	0	1	0	0	1	1
0	0	1	0	1	1	1
0	0	1	1	0	1	1
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	0	0	1	1	1
0	1	0	1	0	1	1
0	1	0	1	1	1	1
0	1	1	0	0	1	0
0	1	1	0	1	1	1
0	1	1	1	0	1	0
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	0	0	1	1	1
1	0	0	1	0	1	1
1	0	0	1	1	1	1
1	0	1	0	0	1	0
1	0	1	0	1	1	0
1	0	1	1	0	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	0	1	1	1
1	1	0	1	0	1	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	0	1	0	1
1	1	1	1	0	0	0
1	1	1	1	1	0	1

Table No.2: Truth Table For C17 Bench Mark Circuit

4.2 C17Bench mark Logic Locking Implementation

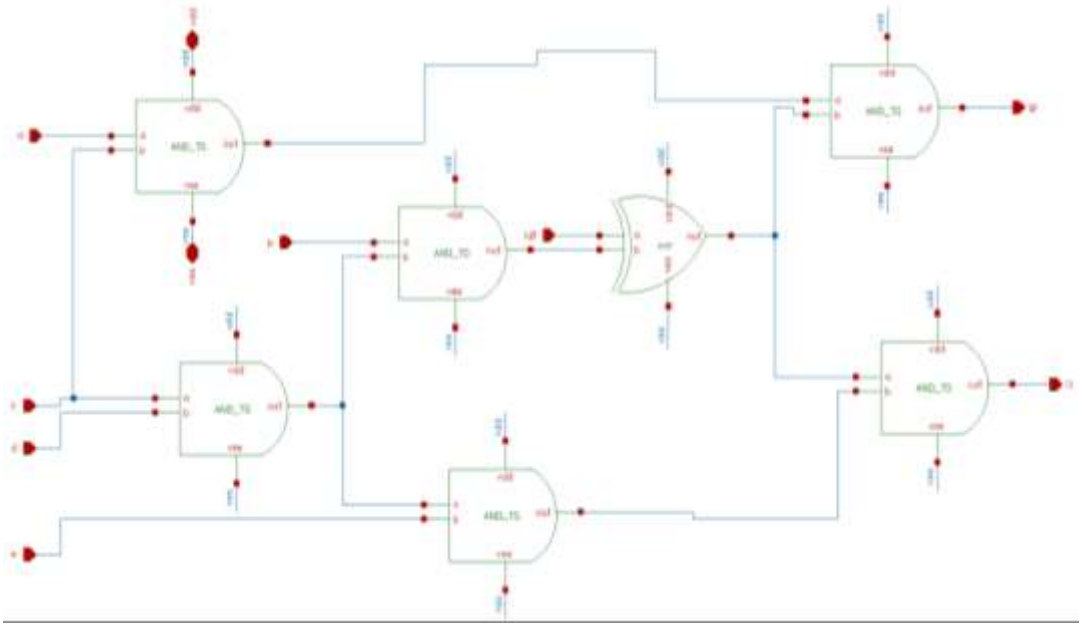

Figure No.2: C17 Bench Mark logic locking implementation circuit

Key Input Behaviour Description

The proposed logic-locked circuit incorporates a single key input (k_0) through an XOR gate to protect the original functionality of the circuit. The key input is inserted at an internal node corresponding to the signal bcd. The output of this node is XORed with the key bit before being propagated to the remaining logic stages.

When the **correct key value ($k_0 = 0$)** is applied, the XOR gate acts as a transparent element, allowing the original signal to pass unchanged. As a result, the circuit produces the intended outputs

$$\boldsymbol{i0} = \left(\boldsymbol{a}[\boldsymbol{0}]\bar{\boldsymbol{a}}[\boldsymbol{2}]\right) \cdot \left(\left(\boldsymbol{a}[\boldsymbol{0}]\bar{\boldsymbol{a}}[\boldsymbol{2}]\right) \bar{\cdot} \left(\boldsymbol{a}[\boldsymbol{1}]\bar{\boldsymbol{a}}[\boldsymbol{2}]\right)\right)$$

$$\boldsymbol{i1} = \left(\left(\boldsymbol{a}[\boldsymbol{0}]\bar{\boldsymbol{a}}[\boldsymbol{2}]\right) \bar{\cdot} \left(\boldsymbol{a}[\boldsymbol{1}]\bar{\boldsymbol{a}}[\bar{\boldsymbol{2}}]\right)\right) \cdot \left(\left(\boldsymbol{a}[\boldsymbol{1}]\bar{\boldsymbol{a}}[\bar{\boldsymbol{2}}]\right) \cdot \boldsymbol{a}[\boldsymbol{4}]\right)$$

However, when an **incorrect key value ($k_0 = 1$)** is applied, the XOR gate inverts the internal signal bcd. This inversion propagates through the subsequent logic gates, altering the output responses and causing the circuit to generate incorrect results.

Thus, the key input directly controls the functionality of the circuit. Only the correct key restores the original behavior, while any incorrect key corrupts the output logic. Since the XOR output is connected to multiple logic paths, a single wrong key bit affects both outputs simultaneously, enhancing the security of the design against unauthorized usage and reverse engineering. This demonstrates the effectiveness of logic locking in protecting integrated circuits by ensuring that the circuit functions correctly only when the proper secret key is applied.

5. IMPLEMENTATION IN CADENCE VIRTUOSO

5.1 Tool Overview

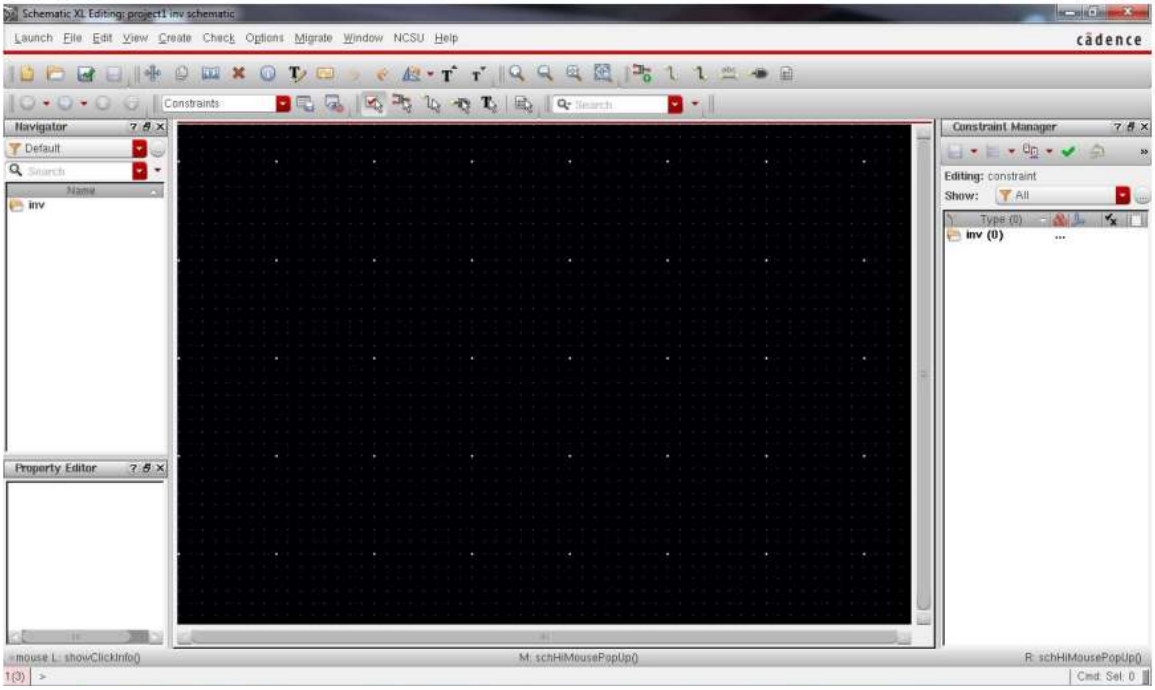

Figure No.3: EDA Tool Environment of Cadence virtuoso

Cadence Virtuoso is an industry-standard Electronic Design Automation (EDA) tool widely used for the design, simulation, verification, and layout of analog, mixed-signal, RF, and custom digital integrated circuits. It provides a complete environment for transistor-level IC design from schematic creation to physical layout and post-layout verification.

Key Features

1. Schematic Design

- Creation of transistor-level and gate-level circuit schematics.
- Supports custom CMOS circuit design.
- Provides extensive component libraries and parameterized cells (PCells).
- Enables hierarchical design methodology.

2. Circuit Simulation

- Integrated with **Cadence Spectre Simulator** for accurate circuit analysis.
- Supports:
  - DC Analysis
  - AC Analysis
  - Transient Analysis
  - Noise Analysis
  - Parametric Sweeps
  - Monte Carlo Simulations

3. Layout Design

- Physical implementation of integrated circuits using process design rules.
- Manual and assisted layout editing.
- Supports:
- CMOS Layout Design
- Device Matching Techniques
- Common-Centroid Structures

- Interdigitated Layouts
- Guard Rings and Shielding

4. Verification Tools

- **Design Rule Check (DRC):** Verifies compliance with fabrication rules.
- **Layout Versus Schematic (LVS):** Ensures layout matches the schematic.
- **Parasitic Extraction (PEX):** Extracts parasitic resistance and capacitance for accurate post-layout simulations.

5. Analog and Mixed-Signal Design

- Supports operational amplifiers, ADCs, DACs, PLLs, RF circuits, and custom digital blocks.
- Facilitates co-simulation of analog and digital components.

Design Flow in Cadence Virtuoso

1. Circuit Specification
2. Schematic Design
3. Functional Simulation
4. Layout Creation
5. DRC Verification
6. LVS Verification
7. Parasitic Extraction (PEX)
8. Post-Layout Simulation
9. Tape-Out for Fabrication

Advantages

- Industry-standard platform for custom IC design.
- High simulation accuracy.
- Comprehensive verification environment.
- Supports advanced semiconductor technology nodes.
- Widely adopted in VLSI, analog, RF, and mixed-signal industries.

5.2 Unoptimized Design

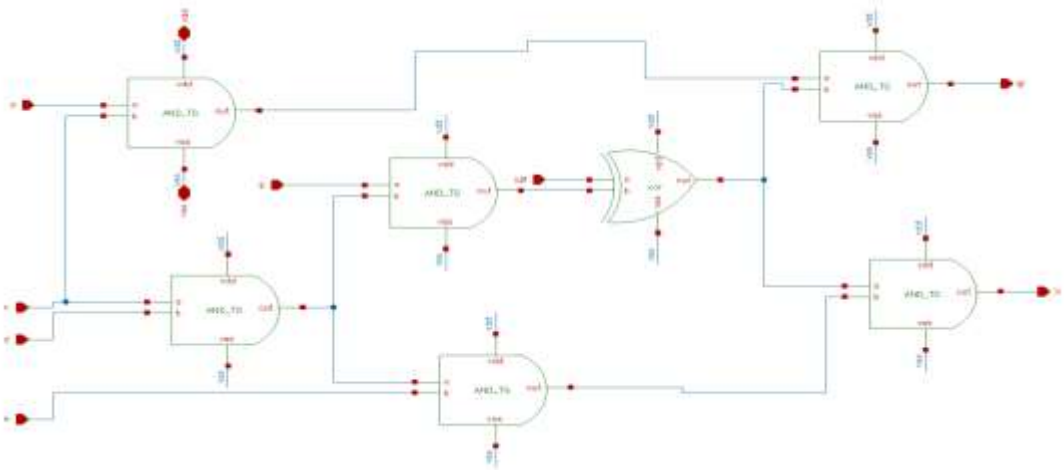

Figure No.4: Unoptimized Design in Cadence virtuoso

Transmission Gate (TG)-based NAND gates and a CMOS-based XOR key gate were used in Cadence Virtuoso to implement the unoptimized logic-locked circuit. Each NAND gate in this design is implemented with eight transistors. The NAND gate section requires 48 transistors in total because six NAND gates are used. Conventional CMOS logic is used to implement the XOR key gate which calls for an extra 12 transistors. Thus there are 60 transistors in the entire circuit. The transmission-gate-based implementation guarantees proper logic operation and keeps the outputs voltage swing full. However the number of transistors produced by this method is rather high. The number of transistors increases parasitic capacitances routing complexity and silicon area requirements. As a result compared to a more optimal design the circuit uses more dynamic and leakage power. From the standpoint of layout more transistors take up more physical space which leads to a larger layout footprint and longer interconnects. Higher propagation delay and power dissipation are caused by these factors. As a result even though the unoptimized design effectively uses the XOR key gate to demonstrate logic locking functionality its efficiency is comparatively low in terms of power consumption transistor count and layout area. This unoptimized implementation highlights the gains made through transistor-level optimization techniques in terms of area reduction lower power consumption and improved overall design efficiency. It also acts as a baseline design against which the optimized circuit can be assessed.

5.3 optimized Design

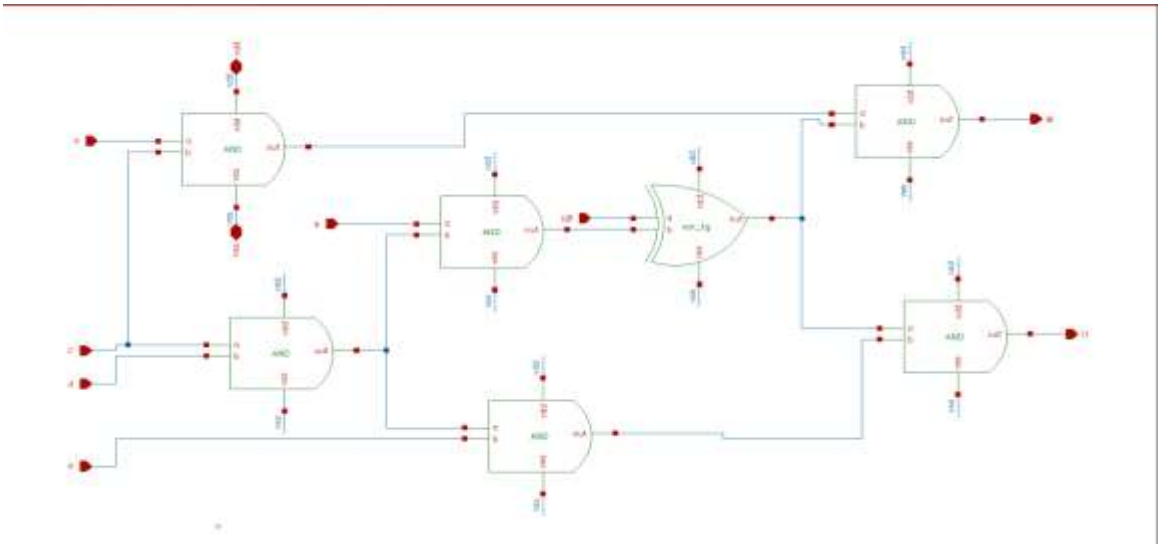

Figure No.5: optimized Design in Cadence virtuoso

A Transmission Gate (TG)-based XOR key gate and standard CMOS-based NAND gates were used to implement the optimized logic-locked circuit in Cadence Virtuoso.

With just four transistors needed for each of the six NAND gates in this design the NAND network is made up of 24 transistors. Eight transistors are needed to implement the XOR key gate using a transmission-gate structure. As a result only 32 transistors are used in the entire optimized circuit which is a considerable decrease over the unoptimized design.

A smaller layout area and lower parasitic capacitances are directly caused by the fewer transistors. The dynamic power consumption is also decreased because fewer transistors are switched during circuit operation. Additionally the compact layout leads to shorter interconnect lengths which enhances circuit performance and further reduces parasitic effects.

The optimized implementation takes up significantly less silicon area from a physical design standpoint which makes it more appropriate for VLSI applications that are power-sensitive and have limited space. Without sacrificing the logic-locking mechanisms functionality the reduced transistor count also makes routing simpler and improves design efficiency. The optimized design achieves significant improvements in power consumption layout area and transistor efficiency when compared to the unoptimized circuit.

Therefore by preserving the necessary logic-locking functionality while drastically lowering resource consumption and enhancing overall VLSI design metrics the suggested optimized implementation shows a more effective hardware security solution.

6. LAYOUT DESIGN

6.1 Unoptimized

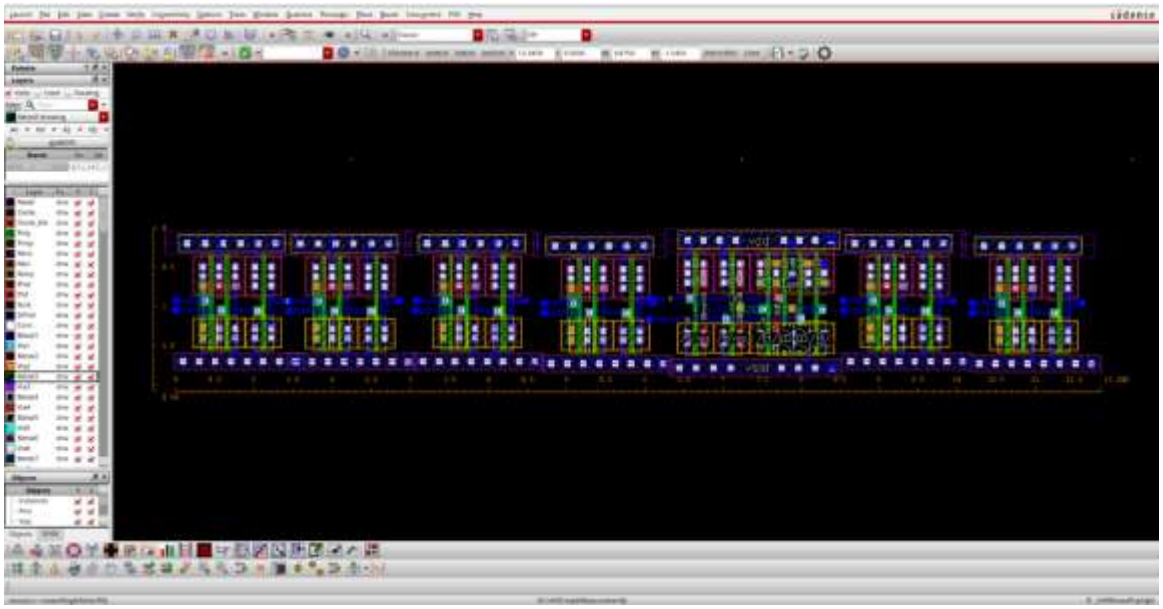

Figure No.6:Layout of Unoptimized Design in Cadence virtuoso

The layout area is calculated as:

$$\text{Area} = \text{Width} \times \text{Height}$$
$$\text{Area} = 16.1\ \mu m \times 2.09\ \mu m$$
$$\text{Area} = 33.649\ \mu m^2$$

Result:
Layout Area = 33.649 µm²
(Approximately 33.65 µm² when rounded to two decimal places.)
The obtained layout area reflects the hardware resources required to implement the logic-locked circuit using the selected transistor-level architecture. The relatively larger area is primarily due to the increased transistor count and the use of conventional CMOS logic structures, which require additional routing and device placement space. A larger layout area directly translates to higher silicon utilization and fabrication cost. Therefore, area optimization is an important design objective in VLSI systems, particularly for low-power and resource-constrained applications. The calculated area serves as a baseline for comparison with the optimized design, where a significant reduction in silicon area was achieved through transistor-efficient logic implementation techniques

6.2 Optimized Layout

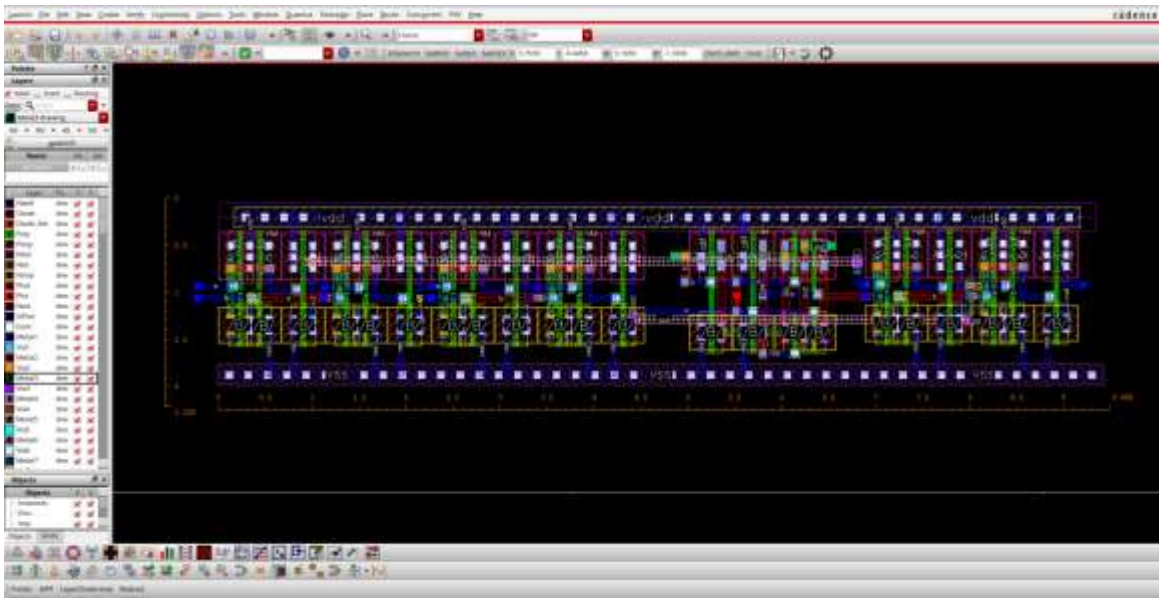

Figure No.7:Layout of optimized Design in Cadence virtuoso

The layout area is:

$$\text{Area} = \text{Width} \times \text{Height}$$
$$\text{Area} = 9.48\,\mu m \times 2.205\,\mu m$$
$$\text{Area} = 20.9034\,\mu m^2$$

Result
Layout Area = 20.9034 μm²
Rounded to two decimal places:

$$20.90\,\mu m^2$$

If you're comparing this with your unoptimized layout area of 33.649 μm², the area reduction is:

$$\frac{33.649 - 20.9034}{33.649} \times 100 = 37.88\%$$

Area reduction ≈ 37.88%.
Hence, the total layout area occupied by the optimized circuit is 20.9034 μm², which can be rounded to 20.90 μm² for reporting purposes.
The reduction in layout area is a direct consequence of the optimization methodology adopted in this work. By replacing conventional CMOS-based logic structures with transistor-efficient implementations, the total number of transistors required for the circuit was significantly reduced. This enabled a more compact placement of devices and minimized routing complexity, resulting in lower silicon utilization. A smaller layout area is highly desirable in VLSI design because it reduces fabrication cost, improves chip density, and allows more functionality to be integrated within a given silicon die.
To evaluate the effectiveness of the optimization, the layout area of the optimized design was compared with that of the unoptimized circuit. The unoptimized implementation occupied 33.649 μm², whereas the optimized implementation occupied only 20.9034 μm². The reduction in area is calculated as:

$$\text{Area Reduction} = \frac{33.649 - 20.9034}{33.649} \times 100$$
$$\text{Area Reduction} = 37.88\%$$

This result indicates that the proposed optimization technique successfully reduced the silicon area by approximately 37.88% while preserving the functionality and security features of the logic-locked circuit. The achieved area savings demonstrate the suitability of the optimized design for low-area and resource-constrained hardware security applications.

7. SIMULATION AND RESULTS

7.1 Simulation Setup

- Supply voltage (Vdd) = 1V
- Stop Time = 400n
- input waveforms= Digital (BCD)
- simulation type = transient

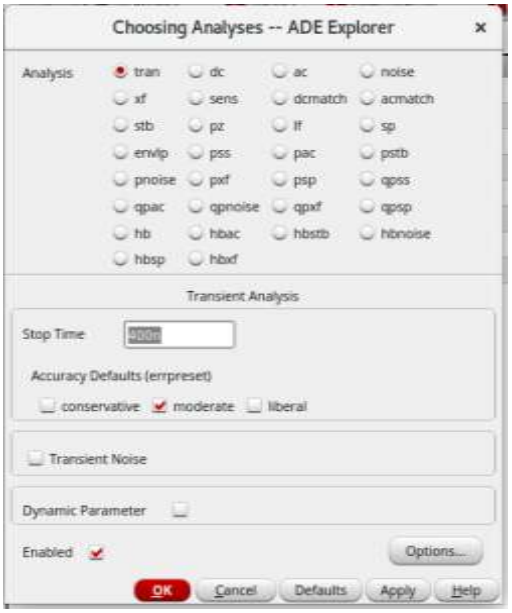

Figure No.8: Analysis Setup for Testbench in Cadence virtuoso

The proposed logic-locked C17 benchmark circuit was designed and simulated in Cadence Virtuoso to evaluate its power performance. A 1 V supply voltage (VDD) was applied to ensure low-power operation while maintaining correct logic functionality. The simulation was performed using Transient Analysis with a stop time of 400 ns, allowing the circuit behavior to be observed over multiple switching cycles.

The input signals were applied as digital BCD waveforms, generating various logic transitions at the circuit inputs. These transitions enabled the analysis of dynamic switching activity and its impact on power consumption. During the simulation, the power waveform was monitored to capture instantaneous power dissipation and peak power values.

The transient analysis results show that power consumption varies with the input switching activity. Whenever the input signals change state, charging and discharging of internal capacitances occur, resulting in power spikes. By comparing the optimized and unoptimized implementations, the effectiveness of transistor-level optimization can be evaluated in terms of reduced switching power and improved energy efficiency.

The simulation setup provides a realistic assessment of circuit performance under dynamic operating conditions and verifies that the optimized logic-locking approach achieves lower power consumption while preserving the intended functionality and security of the design.

7.2 Unoptimized Waveforms

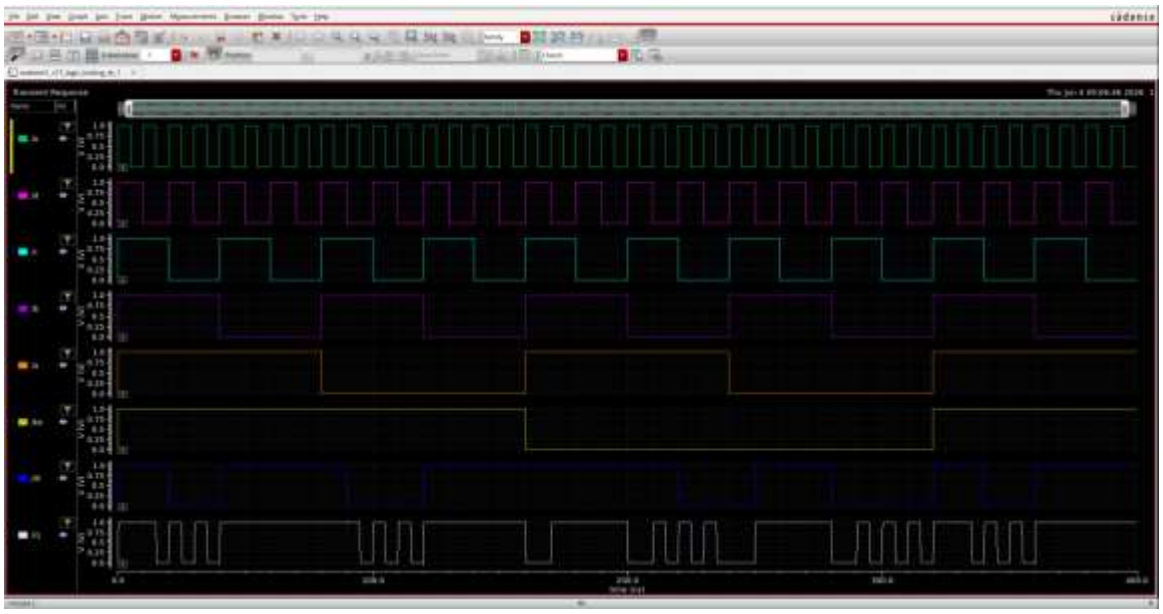

Figure No.9: Unoptimized Design Testbench output in Cadence virtuoso

Figure No.10: Unoptimized Design power consumption in Cadence virtuoso

The transient power analysis of the unoptimized C17 benchmark circuit was performed in Cadence Virtuoso. From the simulation waveform, the circuit exhibits a **maximum power consumption of 944 µW (0.944 mW)** during switching activity. The higher power consumption is mainly due to the use of conventional CMOS-based logic gates and a less optimized transistor arrangement, which increases both dynamic switching power and internal node capacitances. During input transitions, significant power spikes are observed because of charging and discharging of parasitic capacitances. These results indicate that the unoptimized design occupies more area and consumes more power, highlighting the need for transistor-level optimization techniques to improve overall efficiency while maintaining the security features provided by logic locking.

7.3 Optimized Waveforms

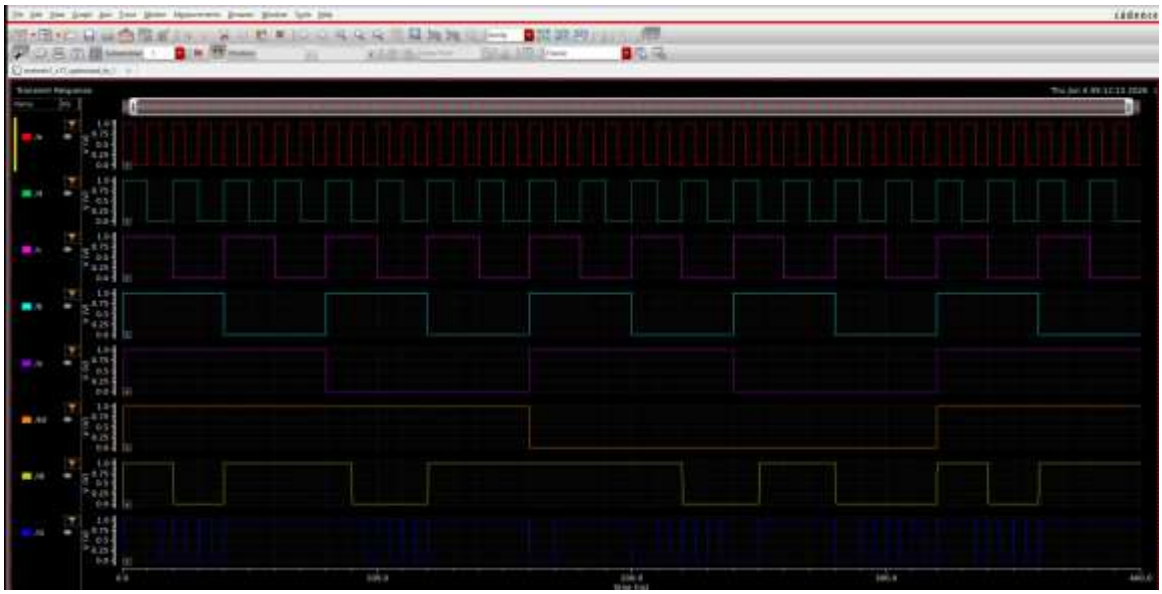

Figure No.11: optimized Design Testbench output in Cadence virtuoso

The circuit uses a key-controlled XOR-based logic-locking mechanism. The key input (K) determines whether the circuit produces the correct functionality or a tampered output.

- When K = 0 (Correct Key):
  - The XOR gate becomes transparent to the original logic.
  - The output (Y) follows the actual output of the benchmark circuit.
  - The circuit operates normally and produces the intended functionality.
- When K = 1 (Incorrect Key):
  - The XOR gate inverts the original logic value.
  - The output (Y) becomes the complement of the correct output.
  - The circuit functionality is altered (tampered), preventing an unauthorized user from obtaining the correct result.

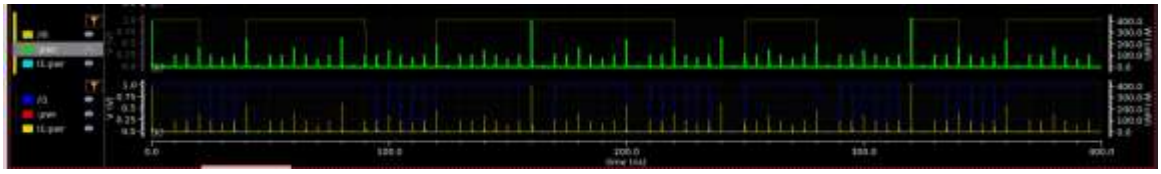

Figure No.12: optimized Design power consumption in Cadence virtuoso

The optimized logic-locked C17 benchmark circuit was simulated using Cadence Virtuoso to evaluate its power performance. From the transient power waveform, the circuit achieved a **maximum power consumption of 400 µW (0.400 mW)**. The reduction in power consumption was achieved through transistor-level optimization, where CMOS NAND gates were implemented using conventional CMOS logic while the XOR key gate was implemented using transmission-gate logic, resulting in a lower transistor count and reduced switching capacitance. Consequently, the charging and discharging currents during logic transitions were minimized, leading to lower dynamic power dissipation. The waveform also shows fewer and smaller power spikes compared to the unoptimized design, indicating improved power efficiency. These results demonstrate that the optimized implementation successfully reduces power consumption while preserving the functionality and security provided by logic locking.

8. COMPARATIVE ANALYSIS

Parameters	Unoptimized	Optimized	Difference	Diff Percentage
Transistor counts	60	32	28	46.66%
Power(max)	994uW	400uW	594uW	59.7%
Area	33.649 µm²	20.90 μm^2	12.749 μm^2	37.88%

Table No.3: Comparative Analysis for the parameters like Transistors count, maximum power, Area

The optimized logic-locking circuit achieved a substantial reduction in hardware complexity by decreasing the transistor count from 60 to 32 transistors, resulting in a 46.67% reduction. This reduction was primarily achieved by replacing conventional CMOS-based logic structures with more transistor-efficient transmission-gate-based implementations.

Power analysis shows that the maximum power consumption decreased from 994 µW to 400 µW, yielding a 59.76% reduction in power dissipation. The lower transistor count and reduced switching activity contribute significantly to this improvement, making the optimized design more suitable for low-power VLSI applications.

Layout analysis further demonstrates the effectiveness of the optimization. The layout area was reduced from 33.649 µm² to 20.90 µm², corresponding to a 37.88% area reduction. This compact layout not only minimizes silicon utilization but also improves the overall design efficiency and scalability for larger integrated circuits.

9.CONCLUSION & FUTURE SCOPE

Conclusion:

In this project a transistor-level logic-locking method for hardware security was designed and implemented using Cadence Virtuosos C17 benchmark circuit. The main goal was to maximize hardware resources while safeguarding the integrated circuit against unauthorized use reverse engineering and intellectual property theft. Initially CMOS-based XOR and NAND gate structures were used to create an inefficient logic-locked circuit. Performance metrics significantly improved after transmission-gate-based logic was used to create an optimized design. The optimized circuit achieved a 46. 67 percent reduction in hardware complexity by reducing the number of transistors from 60 to 32. Additionally the maximum power consumption dropped from 994 µW to 400 µW which translated into a power dissipation reduction of 59. 76 percent. Additionally the layout area was shrunk from 33. 649 µm² to 20. 90 µm² resulting in a 37. 88 percent reduction in silicon area. The logic-locking schemes operation was confirmed by transient simulation results. When the right key (K = 0) was used the circuit generated the desired result. On the other hand an incorrect key (K = 1) prevented unauthorized operation by tampering with the output. The suggested optimized logic-locking technique offers improved security while preserving low power consumption and decreased area overhead as shown by the successful implementation simulation and layout verification. All things considered the project demonstrates that transistor-level logic locking can efficiently enhance hardware security with low resource consumption making it appropriate for contemporary low-power VLSI systems.

Future Scope:

1. Locking using multiple keys.

➢ To expand the key space and strengthen defense against brute-force attacks several key inputs can be used in place of a single key bit.

2. Superior Benchmark Circuits.

➢ The suggested approach can be used to assess scalability and performance in intricate designs on larger ISCAS and ITC benchmark circuits.

3. Logic locking that is resistant to attacks.

➢ Advanced locking techniques can be incorporated to protect against machine learning-assisted attacks removal attacks and SAT-based attacks.

4. Methods of Hybrid Security.
➢ Multiple layers of protection can be achieved by combining logic locking with split manufacturing hardware obfuscation and IC camouflaging.

5. Node scaling of technology.
➢ To examine the effects on power area and security the design can be applied in advanced technology nodes (65 nm 45 nm 28 nm and below).

6. Security architectures with low power.
➢ To reduce leakage power and dynamic power while upholding security requirements more optimization strategies can be investigated.

7. Fabrication and Physical Design.
➢ The design can undergo full ASIC implementation which includes timing analysis place-and-route and fabrication-ready verification.

8. Security Optimization with AI Assistance.
➢ The best locking locations and key-gate insertion techniques can be automatically determined for enhanced security and performance using machine ML techniques.

REFERENCES

1. Analysis of C17 Benchmark Circuit Using Modified Logic Locking Technique
Authors: Smitha M. U and Meghana Kulkarni
Published in: 2025 International Conference on Smart & Sustainable Technology (INCSST) Karnataka, India. July 04-06, 2025
Year: 2025
This paper analyzes the C17 benchmark circuit with a modified logic locking technique that locks both a secret key and a hidden Boolean expression, and conclud…This paper analyzes the C17 benchmark circuit with a modified logic locking technique that locks both a secret key and a hidden Boolean expression, and concludes that using a NAND gate as the key gate reduces power by 12.25% and delay by 2.15% compared to standard XOR-based locking.

2. Interleave Lock: An SAT Attack Resistant Logic Lock for Logic Circuits
Authors: Yang Zeng , Xiaole Cui and Juncheng Pu
Published in: International Symposium of Electronics Design Automation
Year: 2025
This paper proposes a lightweight logic locking method called "Interleave Lock" that combines both logic obfuscation and routing obfuscation into switch boxes …This paper proposes a lightweight logic locking method called "Interleave Lock" that combines both logic obfuscation and routing obfuscation into switch boxes to protect circuits from SAT attacks, while consuming significantly less area, power, and delay compared to existing methods like the Interlock.

3. Breaking Analog Locking Techniques
Authors: Nithyashankari Gummidipoondi Jayasankaran,, Adriana Sanabria-Borbó
Jiang Hu and Jeyavijayan Rajendran
Published in: Ieee transaction in very large scale integration(VLSI) cicuit
Year: 2020
This paper demonstrates that analog locking techniques (which protect analog ICs using secret keys controlling bias currents/voltages) are vulnerable to attack, and proposes SMT (Satisfiability Modulo Theory)-based attacks that can successfully break five analog and three mixed-signal locking techniques regardless of key size, in under a second in most cases.