Sense and React: Self-Destructive Polymorphic Mechanism Against Voltage Tampered Active Physical Attacks

Sourav Roy, Andrew Cannon, Luis de la Mata, Rabin Yu Acharya, Tasnuva Farheen,
Shahin Tajik, *Member, IEEE,* and Domenic Forte, *Senior Member, IEEE*

Abstract*—Secrets such as cryptographic keys and obfuscation keys are used in modern computing systems to protect the sensitive and private information as well as intellectual property (IP). During typical operation, they are stored in volatile memories, e.g., registers and SRAMs, which are vulnerable to active physical attacks whereby environmental parameters such as temperature, system clock, supply voltage, etc. are manipulated to extract information. A common way to protect assets against such attacks are sensors that detect active physical attacks and trigger the destruction of secrets. Often, this requires several thousand clock cycles to accomplish. On top of that, the detection and destruction mechanisms are implemented as separate circuitry, which can be identified and disabled by an attacker. In this article, active physical attacks based on supply voltage manipulation are considered. Storage elements, specifically latches and registers, are designed to change their behavior with supply voltage manipulation and automatically destroy their stored data in – an *integrated* sense and response countermeasure. The ability of an electronic circuit to change its behavior under different environmental conditions is known as polymorphism and such circuits are called polymorphic circuits. In the proposed designs, genetic algorithm (GA) is used to optimize polymorphic gates designed using two separate approaches namely multi threshold null convention logic (MTNCL) and voltage-controlled polymorphism termed in this article as Non-MTNCL. These polymorphic gates are used to design polymorphic latches and registers and both approaches are compared using power, performance, area overhead, reliability criteria and application in cryptographic benchmarks. It is observed that while the GA-optimized MTNCL-based implementation has 75% less area overhead, the GA-optimized Non-MTNCL implementation is 14% more reliable according to simulation results. Apart from the simulations, proof-of-concept is further provided with an FPGA implementation.

***Index Terms*—hardware security, polymorphism, polymorphic latch, polymorphic registers, genetic algorithm, MTNCL polymorphic circuits, Active physical attack.**

I. Introduction

Ensuring data privacy where data is transferred into volatile memory storage can pose a significant challenge as it gives attacker opportunity to launch passive side-channel attacks leveraging unintentional emissions from caches, latches, flip-flops, and/or registers [1], [2], [3]. Active physical attacks such as optical probing [4], fault injection [5], [6], [7], [8] can pose increased threat as attacker can purposefully manipulate the environment specially the supply voltage. For example, in case of thermal laser stimulation (TLS) attack, a preliminary step is to lower the supply voltage to a brownout level. Brownout voltage [9] refers to the specific voltage level below which the IC's logical gates can no longer operate reliably. However, the brownout voltage is still above state element data retention voltage (DRV) [10]. DRV is the minimum voltage below which a memory cell has negative Static Noise Margin (SNM) and loses its state. At brownout voltage, the memory cells retain their data but the circuit-based countermeasures to attacks such as memory zeroization are deactivated. Such successful attacks have been demonstrated where the encryption keys stored in FPGA memory is revealed using thermal laser stimulation [11], [12]. These type of attacks are equally applicable to ASICs. In addition to lowering the supply voltage, attackers can also modulate the supply voltage at a known frequency and freeze volatile memory contents at a specific time by freezing the system clock. This devastating attack is called laser logic state imaging (LLSI). It provides attacker with unlimited number of probes to read out values at any point and time of interest [13], [14]. Voltage glitch-based fault injection can be an effective method to steal cryptographic keys [5], [15]. It is imperative to detect such voltage tampering to protect sensitive assets stored in memories.

Existing countermeasures such as LED-based backside thinning detector [16] or backside tamper detector [17] is designed to detect physical attacks that require chip backside manipulation but doesn't address voltage glitch attacks. Another method uses ring oscillators (ROs) to measure frequency mismatch for detecting environmental disturbances [18]. Other circuit-based detectors, like clock freeze and voltage modulation detectors [19], [20], on-chip monitors (OCMs) [21], and local electromagnetic (LEMA) detectors [22], add significant area and power overhead. All these methods also require separate response mechanisms, such as self-destruction of memory which is invasive [23], or memory-zeroization [24] which is slow and allows attackers time to steal secrets. In addition, detectors may be disabled by lowering supply voltage and even in the presence of detectors successful glitch attack can be implemented [25].

To address the need for a comprehensive, reliable, and integrated sense-and-destroy solution for voltage tampering based attacks, a self-destructive latch [26] was recently proposed by us. It is accomplished by constructing the latch out of voltage-controlled polymorphic NOR/NAND gates and a polymorphic buffer/always-off gate. The polymorphic latch operates normally when the supply voltage is within the IC's specification and erases the latch's data when the voltage drops below a specific threshold. The threshold is easily calibratable by changing the sizes of only a few transistors and can be used to cater to different applications and attack conditions. Further, since the latch only destroys data when its local supply voltage is affected, the self-destruction works equally well against non-invasive (less precise) and semi-invasive (more precise) voltage-manipulation attacks. In this article, prior work is extended by incorporating a new NAND/NOR polymorphic gate, optimization of transistor sizes using genetic algorithm (GA) and extending the concept of self-destruction to registers along with establishing the proof-of-concept in cryptographic benchmarks.

Contributions. Contributions of this work are summarized as follows:

- Optimization of two NAND/NOR polymorphic gates from [26] which is MTNCL-based and from [27] which is voltage-controlled polymorphism-based, termed in this work as Non-MTNCL-based. Optimization is done by sizing transistors using GA.

S. Roy, A. Cannon, T. Farheen, and D. Forte are with the Department of Electrical and Computer Engineering, University of Florida (email: sourav.roy@ufl.edu, andrew.cannon@ufl.edu, tasnuvafarheen@ufl.edu, and dforte@ece.ufl.edu)

L. de la Mata is with the Department of Electrical and Computer Engineering, Comillas Pontifical University (email:201808137@alu.comillas.edu)

R. Yu Acharya is with the Intel Corporation (email: rabin.yu.acharya@intel.com)

S. Tajik is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (email:stajik@wpi.edu)

- Addition of a discharge path for faster response time for the MTNCL-based design.
- Latches are designed using both optimized MTNCL-based and Non-MTNCL-based NAND/NOR gates and their post-layout power, performance, and area as well as their reliability across process variation and temperature are compared between them and also with the non-optimized design from [26].
- Creation of a polymorphic registers using all polymorphic latch designs mentioned above and comparison using cryptographic benchmarks
- Proof-of-concept of polymorphic latches in an FPGA with two different implementations: 1-LUT and 2-LUT.

The rest of the work is organized as follows. In Section II, background on attack mechanisms, state-of-the-art countermeasures, and different ways of designing polymorphic circuits and genetic algorithm-based optimization technique used in this work is provided. In Section III, a threat model with assumptions on attacker capabilities is described. In Section IV, the designs of NAND/NOR gate are presented and optimizations implemented in those designs are explained. In Section V, both GA-optimized MTNCL-based and GA-optimized Non-MTNCL-based latches are presented with layouts. In Section VI, the extension to polymorphic registers is explained with layout. Section VII describes our FPGA implementations for a polymorphic latch. In Section VIII, simulation results with power, performance, area (PPA) and reliability characterization of the latches and registers is presented along with benchmark simulations and FPGA results. Finally, in Section IX, conclusions are summarized.

II. Background

A. Attack Vectors

In this article, active physical attacks that involve supply voltage manipulation are considered. In the case of laser based attacks such as thermal laser stimulation (TLS) and laser logic state imaging (LLSI), attacker also freezes the system clock along with supply voltage manipulation. In the case of voltage fault injection (VFI) attack, the attacker injects an under-voltage for a short amount of time (at least few hundred nanoseconds) [28] at an appropriate moment. The injected fault corrupts the system output and by comparing the corrupted output with golden output, attackers can figure out secret information. Attack steps of TLS, LLSI and VFI are summarized below.

1) Thermal Laser Stimulation (TLS): In case of TLS, attacker scans the region of interest with a laser with wavelength of about $1300\mathrm{nm}$ which causes local heating and generates current at the drain of 'ON' transistors. The attack vectors of TLS attack are as follows [29]:

- For the data to be preserved in the volatile memory cells, the *clock must be frozen* for the attack duration. A frozen clock does not let the sequential elements such as latches and registers change state.
- In order to disable circuit-based defensive measures, *supply voltage is lowered* approximately $\frac{V_{DD}}{2}$ or brownout voltage which is above the volatile memory's DRV.

2) Laser Logic State Imaging (LLSI): LLSI was developed as a fault analysis technique to detect faults by observing static signals [13]. Attackers can use it with malicious intent to steal secrets from a chip [14]. In LLSI attacks, a laser with wavelength of about $1100\mathrm{nm}$ is used to scan the region of interest in the target chip. At this wavelength, the laser energy is above silicon band-gap and transparent to bulk silicon. Thus, the reflected light intensity is channel dependent which can be used to distinguish between transistors in 'ON' or 'OFF' states. The attack vectors of LLSI attack are as follows:

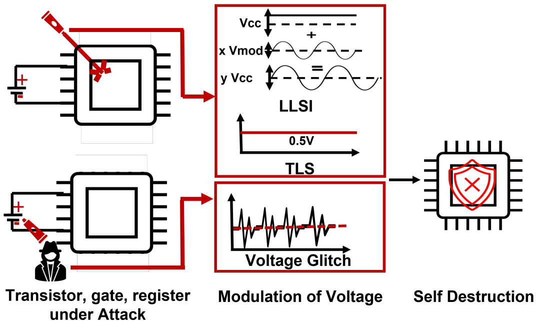


Fig. 1: (left) Different types of voltage manipulation: voltage modulation (LLSI), voltage lowering for thermal laser stimulation (TLS) and voltage fault injection (VFI); (right) Desired self destruction of sensitive on-chip data as a defense mechanism.

- Similar to TLS, the *clock is frozen* to keep the sequential elements from changing state.
- In order to set the capture frequency of spectrum analyzer, *supply voltage is modulated at a known frequency* where the amplitude of modulation has to be large enough to get clear distinguishing signature of 'ON' and 'OFF' transistors but not large enough so that the gates or registers lose their original states.

3) Voltage Fault Injection (VFI) Attack: Noninvasive attacks such as VFI or voltage glitching are inexpensive and require little to no technical knowledge. A software-managed VFI setup can be used to automate the attack and reduce complexity [5]. Fault injection attacks have compromised even trusted execution environments (TEEs) [6] which are specially designed with security in mind. Using a software-based framework, a circuit can be triggered at the appropriate time to inject faults, and the fault results can be analyzed using software designed to extract assets by analyzing circuit behavior under faults. The attack vectors of the VFI attack are as follows:

- *Fault injection setup* to introduce sudden temporary voltage glitch.
- Design a *mathematical model* based on the target for fault analysis and a timing controller circuitry for triggering voltage faults at appropriate time.
- Extract secret information based on fault behavior using *fault analysis*.

Fig. 1 illustrates these attack vectors and the common threat model is explained in Section III.

B. Existing Countermeasures

Several countermeasures have been developed so far for detecting attempt of environmental manipulation such as voltage glitching, under-voltage, clock freeze, probing attempt etc. The countermeasures are categorized in Table I in terms of what type of attacks those are applicable to and the nature of the countermeasures. Invasive, Semi-invasive and non-invasive attacks are indicated with letters **I**, **SI** and **NI** respectively. Sense, response and prevent are expressed using letters **S**,**R** and **P** respectively as the nature of the countermeasures. Circuit based countermeasures such as ring oscillators (ROs) network [18], clock freeze and voltage modulation detectors, impedance based backside tamper detector [17], on-chip monitors (OCMs) and local electromagnetic (LEMA) detectors [19], [20], [21], [22] are mainly detectors that detect physical attack attempt and requires separate circuitry for response mechanism. These circuit-based detectors have to be distributed over the whole chip and result in high area and power overhead. Response mechanisms such as self-destruction [23] requires complex fabrication steps and zeroization [24] can be too slow to react before attack subsides. These countermeasures require separate circuitry for sense and response. Circuit based detectors can be disabled by lowering the supply voltage to brownout level where although detectors are disabled, memories still retain data [30], [24] for the attacker to read-out. Moreover, attackers with reverse engineering capability can locate the detectors and can disable the communication between detector and response mechanism using circuit edit [31], [32]. Prevention based countermeasure such as Nanopyramid structures fabricated in the metal layers have been proposed to obfuscate reflected beams during laser-based attack [33] but this approach suffers lower device reliability due to higher metal complexity and susceptibility to electromigration [34]. Again, this countermeasure is only effective against laser based attack from chip backside and not effective against voltage-based fault injection attacks. As a result, in this work, a simple, fast, polymorphic sense and react based solution is proposed.

C. Design of Polymorphic Gates

Polymorphic gates are logic gates that change behavior with changes in environmental factors. Voltage, temperature, and light are some of the factors that alter the functionality of polymorphic gates. NASA's Jet Propulsion Laboratory first introduced polymorphic gates in 2001 [35]. Since then, numerous high-performance polymorphic gates with multiple functions have been designed [36]. Polymorphic gate design methods enable logic gates, such as NAND/NOR, to switch functionality based on external conditions like voltage, temperature, or control signals. Voltage-controlled polymorphism [27], [37], [38] leverages voltage-dependent transistor behavior for reconfigurability but may face reliability issues under varying power conditions. Temperature-controlled polymorphism [38] exploits thermal properties for functionality switching but is sensitive to environmental fluctuations. Control signal-based polymorphism [39], [38] uses external signals to reconfigure gates, offering precise control but requiring additional circuitry. Evolutionary hardware design [40] employs algorithms to optimize polymorphic behavior, offering innovative solutions but requiring extensive computational resources. Process variation-based designs exploit manufacturing variations for polymorphism, though they may lack consistency. Each method has unique strengths, such as adaptability and security, but also challenges, including reliability, complexity, and scalability. These approaches are difficult to use when porting to different technology nodes in practice. Recently, a multi threshold null convention logic (MTNCL) based polymorphic circuit design mechanism that is sensitive to supply voltage was proposed [41], [42]. In this method, a control transistor pair controls whether the pull-up network or the pull-down network will be activated depending on the supply voltage. MTNCL is a methodology to design asynchronous circuitry that uses dual rail logic. In this method, threshold voltage drop of a diode connected NMOS transistor is used to create polymorphism with supply voltage change.

D. Genetic Algorithm for Optimization

The genetic algorithm (GA) is a powerful optimization technique inspired by the principles of natural selection in biological evolution [43]. Genetic Algorithms (GAs) are often better than Particle Swarm Optimization (PSO) [44], Bacterial Foraging Optimization (BFO), and Ant Colony Optimization (ACO) for optimization because they balance exploration (searching the entire solution space) and exploitation (refining good solutions) more effectively. GAs use crossover and mutation to explore diverse solutions, helping them avoid getting stuck in local optima, which is a common issue with PSO, BFO, and ACO. Additionally, GAs are robust in noisy or changing environments and can handle complex, high-dimensional problems better, making them a versatile and reliable choice for many optimization tasks. GA is particularly effective in optimizing the design of polymorphic gates by iteratively improving a set of candidate solutions [45]. In this article we apply GA to optimize the transistor sizes in both MTNCL-based [26] and voltage-controlled Non-MTNCL-based [27] designs, aiming to achieve the required self-destructive behavior while minimizing the overall circuit size. The transistors were sized using trial and error method in $45nm$ technology node in [26] and in this work GA is used to find the minimum transistor sizing that reliably produces desired polymorphism phenomena. The voltage-controlled polymorphic gate in [27] is designed at older technology node and higher supply voltage and in this work, GA is used to minimize the transistor sizing for $45nm$ technology node. The optimization process is as follows:

Pre-execution Decisions. Before executing the GA, it is crucial to select which transistors to evolve, as the number of transistors involved directly affects the algorithm's runtime. Before executing the algorithm, population size, tournament size, mutation rate, the range of transistor widths and lengths must be configured. Table II lists the final values of these variables that showed better and faster results and therefore the values used in the genetic algorithm (GA).

Genetic Algorithm Processes. The GA involves several key processes. The steps and pseudo-code for them are given below.

1) *Initial Population Generation*: The initial population of transistor sizes is generated randomly within predefined limits for widths and lengths.

Algorithm 1 Initial Population

```
for $i \leftarrow 0; i < populationSize; i++$ do
    $widths \leftarrow randomSizes(widthRange)$
    $lengths \leftarrow randomSizes(lengthRange)$
    $population[i] \leftarrow [widths, lengths]$
end for
```

2) *Fitness Evaluation*: Each circuit in the population is evaluated using a fitness function that measures how well their associated transistor sizes meet the design requirements.
3) *Selection*: A tournament selection function is used to choose the best candidates for the next generation. This function samples fitness values, selects the best circuits from these samples, and designates them as parents.

Algorithm 2 Tournament Selection

```
$tournament1 \leftarrow randomSample(fitnessValues)$
$bestCandidate1 \leftarrow maxFitness(tournament1)$
$tournament2 \leftarrow randomSample(fitnessValues)$
$bestCandidate2 \leftarrow maxFitness(tournament2)$
```

4) *Crossover and Mutation*: The crossover function generates a new population by combining the transistor sizes of selected parents. A mutation function introduces random changes to maintain genetic diversity and avoid local minima.

Algorithm 3 Crossover

```
for $j \leftarrow 0; j < numberOfTransistors; j++$ do
    $widths[j] \leftarrow random(candidate1[j], candidate2[j])$
    $lengths[j] \leftarrow random(candidate1[j], candidate2[j])$
    if $random(0, 1) \leq mutationRate$ then
        $widths[j] \leftarrow random(minWidth, maxWidth)$
        $lengths[j] \leftarrow random(minLength, maxLength)$
    end if
end for
$child \leftarrow [widths, lengths]$
```

TABLE I: Existing Countermeasures. I: Invasive attack, SI: Semi-invasive attack, NI: Non-invasive attack, S: Sense-based, R: React-based, P: Preventive. S+R: Sense and React integrated

Countermeasure	I	SI	NI	S	R	P	S+R	Notes
Impedance based backside tamper detector (BackMon) [17]		✓		✓				Uses change in impedance due to backside tampering using on-chip impedance monitoring circuit
Light emitting diode (LED)-based backside thinning detector [16]		✓		✓				Detects backside thinning required for laser-based attacks, requires extra fabrication steps
Network of ring oscillators (ROs), PUFMon [18]		✓	✓	✓				Measures frequency mismatch to determine laser-based or FIB-based probing attempt and fault injection.
Clock freeze and voltage modulation detectors [19], [20]		✓		✓				Detects disturbances in the clock signal and supply voltage to detect laser based probing attempt.
On-chip monitors (OCMs) and local electromagnetic (LEMA) detectors [21], [22]		✓	✓	✓				Detects supply voltage disturbance through on-chip sensors and detects electromagnetic probing attempts through LEMA. Requires separate response mechanism.
Inductive impulse Self-destruction [23]	✓	✓	✓	✓	✓			Sensor detects physical attack attempt and generates high-voltage impulse to permanently destroy cryptographic processor.
Memory zeroization [24]	✓	✓	✓		✓			Generates a zeroization request and leverages the memory built in self test (MBIST) architecture to destroy memory data.
Nanopyramid structure [33]		✓				✓		Nanopyramid structures fabricated in metal layers scatters reflected laser beam and provides obfuscation.
Polymorphic latch and registers: This work		✓	✓				✓	Changes behavior under attack condition such as voltage manipulation and immediately destroys sensitive assets.

TABLE II: Variables values for the GA.

Population Size	Tournament Size	Mutation Rate	Width range (nm)	Length range (nm)
300	4	0.3	120–10000	45–10000

Algorithm 4 Genetic Algorithm

```
$population \leftarrow firstPopulation()$
while $bestFitness > 0.1$ do
    for $i \leftarrow 0; i < populationSize; i++$ do
        $fitness[i] \leftarrow getFitness(population[i])$
    end for
    $bestFitness \leftarrow max(fitness)$
    for $i \leftarrow 0; i < populationSize; i++$ do
        $bestCandidates \leftarrow tournamentSelection(fitness)$
        $child \leftarrow crossoverFunction(bestCandidates)$
        $population.append(child)$
    end for
end while
```

The complete GA repeats the above steps 2–4 until a stopping criteria is met. In Algorithm 4, the algorithm stops when the best candidate achieves a fitness score below 0.1.

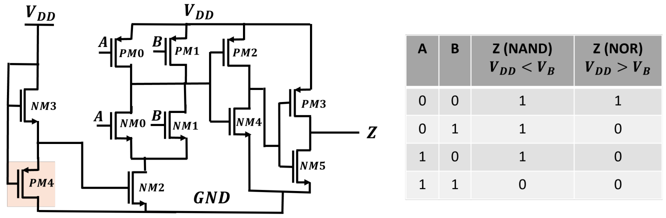
III. Threat Model

When mounting different types of active physical attacks, the attacker actively tries to manipulate environment such as supply voltage, system clock, temperature, etc. in the attempt to get unauthorized access or to steal secret information from the chip. The first step for the attacker is to localize point or region of interest (POI/ROI) through reverse engineering. The next step is to formulate a threat model which gives the attacker information about method and timing of environmental manipulation for carrying out the attack. The attack may be active physical attack, such as laser assisted attacks which requires tens of hours [4] if the complete layout is available to the attacker and requires days if reverse engineering has to be done from scratch to reveal the POI/ROI. In case of fault injection attacks, localization of voltage regulator ICs is easily identifiable [15] but precise timing and duration of fault has to be maintained for a successful fault injection. The attack duration is usually in the range of μs [46], [15], [5] but it can be as low as hundreds of ns [28]. Finally, the attack is carried out at the POI/ROI. Here, the adversary is likely to be the end-user with access to the fabricated and deployed chip. The threat model may involve manipulation of supply voltage and/or system clock as explained in attack vectors of TLS, LLSI and VFI attacks in Section II – note that the common thread among all of them is voltage supply manipulation.

IV. Proposed Polymorphic NAND/NOR Gate

In this article, two methodologies of polymorphic NAND/NOR designs are proposed. In the first methodology, MTNCL-based polymorphic NOR/NAND gate from [26] is used and the transistor sizes are optimized using GA for $45\mathrm{nm}$ technology node. In the second methodology, a voltage controlled Non-MTNCL NAND/NOR design is chosen from [27] and the transistor sizes are optimized using GA for $45\mathrm{nm}$ technology node. Both designs are GA-optimized.

1) GA-optimized MTNCL-based Polymorphic NOR/NAND Gate: The polymorphic NOR/NAND gate presented here is an optimized version of the gate presented in [26]. Here, twofold optimization has been achieved. Firstly, the GA is applied to minimize the transistor sizes keeping the same polymorphism threshold of $780mV$. The transistor sizes of original version presented in [26] and the optimized version presented in this work are provided in Table III. Secondly, an additional PMOS transistor is added which is connected to the threshold drop-configured NMOS transistor, improving the response time of the latch to an attack. This change is highlighted in Fig. 2 and the width of added PMOS transistor is shown in Table III. Genetic algorithm optimizes the width of $NM2$ and $NM3$ from $1800\mathrm{nm}$ to $810\mathrm{nm}$ in the $45\mathrm{nm}$ technology node where the length of all transistors for previous and proposed designs is set at $45\mathrm{nm}$. The width of additional PMOS transistor $PM4$ is $2100\mathrm{nm}$. The overall effect of both optimizations is area increase but faster response time. The area increase of overall optimization in the latch design using this NOR/NAND gate is about 8.5% but it leads to more than $3000\times$ faster response. The response time is defined as the time needed for the polymorphic latch or register to enter forbidden state after supply



A	B	Z (NAND) $V_{DD} < V_B$	Z (NOR) $V_{DD} > V_B$
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Fig. 2: MTNCL-based polymorphic NOR/NAND gate with PMOS transistor $PM4$ (colored in pink) to improve response time by providing a discharge path for $NM2$'s gate voltage when $NM3$ is 'OFF'.

TABLE III: Transistor sizing for MTNCL-based NOR/NAND gate presented in [26] and the GA-optimized MTNCL-based design presented in this work. The length of all the transistors is the minimum value of $45nm$.

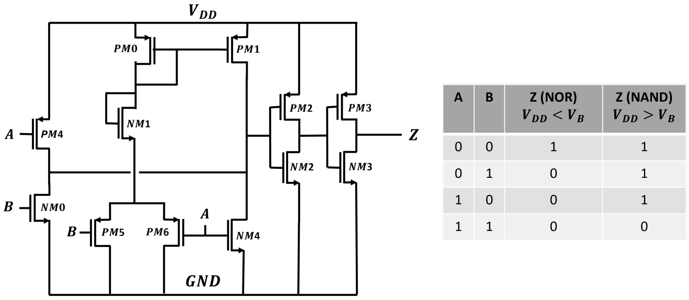
Transistors	Width (nm) in [26]	Width (nm) in this work	Response time in [26]	Response time in this work
NM0, NM1	600	600	Latch:	Latch:
NM2, NM3	1800	810	$364\mu s$	$112ns$
NM4, NM5	120	120		
PM0, PM1	120	120	Register:	Register:
PM2, PM3	120	120	$355\mu s$	$117ns$
PM4	NA	2100		

voltage is lowered to carry out active physical attack. The forbidden state occurs when both outputs Q and its complement $\overline{Q}$ enters the same logic level.

The PUN of the NOR/NAND gate consists of PMOS transistors $PM0$ and $PM1$ connected in parallel representing the PUN of a 2-input NAND gate while the PDN of parallel NMOS transistors $NM0$ and $NM1$ depicts the PDN of a 2-input NOR gate. A buffer is connected at the output which is the connection point between PUN and PDN. The polymorphism is established through two NMOS transistors $NM2$ and $NM3$ where $NM3$ is connected in a threshold-drop configuration and providing gate voltage to $NM2$. When the supply voltage is lower than a threshold V_B, the gate voltage of $NM2$ is not high enough to turn it on and quickly discharges through $PM4$. Thus, the PDN is disconnected and PUN ensures NAND behavior. However, when the supply voltage is above the threshold V_B, both $NM2$ and $NM3$ are fully on and NOR functionality is ensured through PDN. To ensure this, pull-down transistors are sized about five times compared to pull-up transistors. The truth table of the operation is provided in the right hand side of Fig. 2.

2) GA-optimized Non-MTNCL-based Polymorphic NAND/NOR Gate: The GA-optimized Non-MTNCL polymorphic NAND/NOR gate schematic shown in Fig. 3 was originally presented in [27]. The polymorphic NAND/NOR gate presented in [27] adopts $0.5\mu m$ technology node. Due to differences in technology nodes and device scaling it is not possible to utilize the transistor sizing from [27] and instead GA approach is used to optimize the transistor sizing for the technology node used in this article. This circuit's operation can be explained as follows:

- When inputs A and B are of the same logic value (either logic high or low), it behaves like an inverter. In case of both inputs are logic low, $NM0$ and $NM4$ are off and output is logic high through $PM4$ and $PM1$. When both inputs are logic high, $NM0$ and $NM4$ are on and drive the gate output to logic low. This behavior is same for both NAND and NOR case. Polymorphic behavior occurs when inputs have different logic as shown in the truth table of Fig. 3.



A	B	Z (NOR) $V_{DD} < V_B$	Z (NAND) $V_{DD} > V_B$
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

Fig. 3: GA-optimized Non-MTNCL polymorphic NAND/NOR gate. The gate functions as NAND for V_{DD} greater than V_B and functions as NOR for V_{DD} less than V_B.

TABLE IV: Transistor sizing for GA-optimized Non-MTNCL-based NAND/NOR gate.

Transistors	Width (nm)	Length (nm)
NM0	120	450
NM1	800	45
NM2, NM3	120	45
NM4	120	6000
PM0	120	3000
PM1	3000	145
PM2, PM3, PM4	120	45
PM5, PM6	2900	120

- When A is logic low and B is high, both $NM0$ and $PM1$ are on. Depending on the supply voltage, the output either settles at logic low through $NM0$ or logic high through $PM1$. If the supply voltage is above a certain threshold V_B (assuming suitable sizes for the transistors), $PM1$ wins the race and output settles at logic high.
- When A is logic high and B is logic low, both $NM4$ and $PM1$ are on and again $PM1$ wins the tug of war if the supply voltage is above the threshold V_B and drives the gate output to logic high. If supply voltage is below the threshold V_B, gate output becomes logic low through $NM4$.

Thus, in the case of GA-optimized Non-MTNCL NAND/NOR gate, it operates as a NAND gate under normal operating condition and behaves as a NOR gate at lower voltages than the threshold. Note that this is the opposite polymorphic behavior compared to the MTNCL-based polymorphic gate described before. The threshold for polymorphism depends on transistor sizing specially of transistors $PM1$ and $NM4$. The transistor sizing is optimized using GA and the final values are shown in Table IV. In this work, three designs for NAND/NOR gate are considered and used to implement latches and registers: the MTNCL NOR/NAND gate from [26], the GA-optimized (twofold optimization: GA for sizing, discharge path for response time) MTNCL NOR/NAND gate and GA-optimized Non-MTNCL NAND/NOR gate. The designs are summarized in Table V.

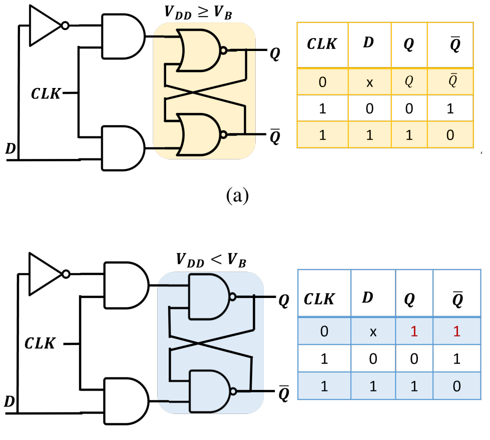
V. Proposed Polymorphic Latch

A. GA-optimized MTNCL-based Polymorphic Latch Design

The configuration and transformation of the polymorphic NOR/NAND gates in the latch design are shown in Fig. 4(a). Under normal supply voltage the latch holds data but under supply voltage manipulation when the voltage drops below a threshold V_B, it enters forbidden state ($Q = \overline{Q}$). This happens when the system clock, CLK is low irrespective of the input, as highlighted in Fig. 4(b). This causes meta-stability in the latch and it is expected to settle to a random value when the supply voltage returns to normal operating

TABLE V: Polymorphic NAND/NOR gates used in this article to design latches and registers. Comparative analysis is done in terms of power, performance, area, temperature and process variation.

Design Name	Technology Node	Transistor Sizing Method	PMOS in Discharge Path	Polymorphic Gate Design Method
MTNCL-based Polymorphic NOR/NAND [26]	45nm	Manual	No	MTNCL [41]
GA-optimized MTNCL-based Polymorphic NOR/NAND		GA	Yes	MTNCL [41]
GA-optimized Non-MTNCL-based Polymorphic NAND/NOR		GA	N/A	Voltage-controlled [27]



CLK	D	Q	$\bar{Q}$
0	x	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

(a)

CLK	D	Q	$\bar{Q}$
0	x	1	1
1	0	0	1
1	1	1	0

(b)

Fig. 4: MTNCL-based polymorphic NOR/NAND D-latch and truth tables for (a) normal operation ($V_{dd} > V_B$) and (b) attack condition ($V_{dd} \leq V_B$). For latter, the polymorphic gates change from NOR behavior (yellow) to NAND behavior (blue) and the latch enters the forbidden state (red) when clock is logic low.

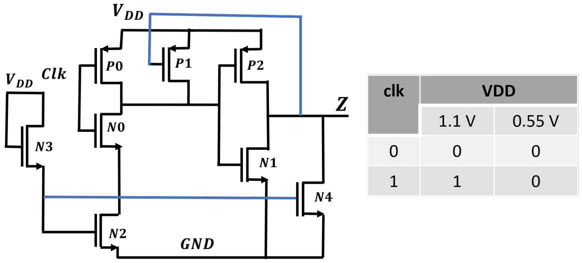
condition, effectively "self-destructing" its data. The latch layout is done and post-layout simulation is used to perform measurements in later sections. The polymorphic gate that forces the clock input of the latch to a low value is discussed in Section V-C.

B. GA-optimized Non-MTNCL Based Latch Design

The polymorphic latch for GA-optimized Non-MTNCL design is chosen to be the NAND latch instead of the NOR latch as under normal operating condition, the GA-optimized Non-MTNCL polymorphic NAND/NOR gate operates as a NAND gate. Under normal operation when supply voltage V_{DD} is above the threshold V_B the polymorphic NAND/NOR gate in the latch operates as a NAND gate and the latch latches the data like a standard latch. But under attack conditions, when supply voltage falls below the carefully chosen threshold V_B, the polymorphic gate changes its behavior to NOR and the latch enters forbidden state when the clock is logic low. Post layout simulation is performed for results presented in Section VIII-B1 and are compared with MTNCL-based polymorphic latch.

C. GA-optimized Polymorphic Clock Buffer/Always-off Gate and Layout

As discussed previously and indicated in the truth table in Fig. 4, the polymorphic latch only enters the forbidden state when *CLK* input is 0. Since it is possible for the system clock to be frozen in the 1 state during an attack, the latch may still retain its data under attack conditions. To counter this, a supplementary polymorphic gate is proposed, which can be incorporated into the clock tree to force the system clock to 0 when the supply voltage is lowered as per the attack conditions. That is, the proposed polymorphic gate behaves as a clock buffer when the supply voltage is higher than the threshold V_B, passing the clock signal without change. However, when the supply voltage is lower than V_B, the gate behaves as an always-off gate and outputs a constant logic 0 value.



clk	VDD	
	1.1 V	0.55 V
0	0	0
1	1	0

Fig. 5: Polymorphic clock buffer/always-off gate. The gate passes *CLK* to Z unaltered at 1.1V, and outputs 0 to Z at 0.55V.

TABLE VI: Transistor sizing for clock buffer/always-off gate.

Transistors	Width (nm)	Length (nm)
N0, N3, N4	120	45
P0, P1	120	45
N2	400	45
N1, P2	700	45

The schematic representation of the polymorphic buffer/always-off gate is shown in Fig. 5. This gate, like the polymorphic NOR/NAND gate used for the self-destructive latch, was designed following the approach discussed in [41]. This polymorphic clock buffer is implemented with NOR/XNOR as its two logical functions, with output inversion. A keeper configuration is used to feed the second input to the PUN, removing the need for inverting *CLK*. The functionality of this gate is discussed in further detail in [26].

The GA method discussed in Section II-D was applied to find optimal transistor sizing for the polymorphic buffer/always-off gate. Transistor N2's width was reduced from 1800nm to 400nm, while N1 and N2's widths were increased to 700nm each. All other transistors were reduced to minimal sizing. This improvement reduces the area of the clock buffer and significantly improves its noise characteristics. With the optimal transistor sizing, shown in Table VI, peak-to-peak output ripple is limited to 3.7mV in the always-off mode. The layout of the gate is shown in Fig. 6 and has a measured area of 4.62μm^2.

VI. Proposed Polymorphic Register

The polymorphic register is designed by connecting two polymorphic latches back to back as shown in Fig. 7. In the case of MTNCL-based polymorphic register, the latches used are GA-optimized MTNCL-based latches and in the case of Non-MTNCL polymorphic register, the latches used are GA-optimized Non-MTNCL latches. The layout of GA-optimized MTNCL-based register and GA-optimized Non-MTNCL-based register are given in Fig. 8 and Fig. 9, respectively. Post layout simulations for the registers are performed and compared in Section VIII-C1.

VII. Implementation of Polymorphism in FPGA

Active physical attacks can also be used against FPGAs. Thus, this article proposes a polymorphic latch design that can be utilized for

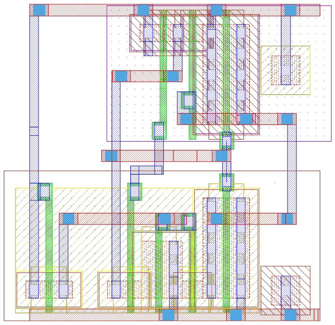


Fig. 6: Layout for clock buffer/always-off gate. The area of the layout is measured as 2.1µm × 2.2µm (4.62µm^2).

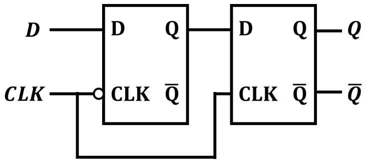


Fig. 7: Illustration of the construction of a register from two latches.

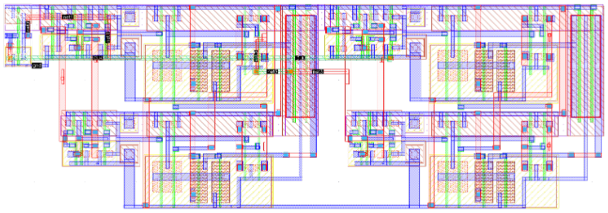


Fig. 8: Layout view of optimized MTNCL-based polymorphic register with two MTNCL-based latches connected in master slave configuration. Areas are measured as 5.5µm × 4.0µm for latch and 11.8µm × 4.0µm for register.

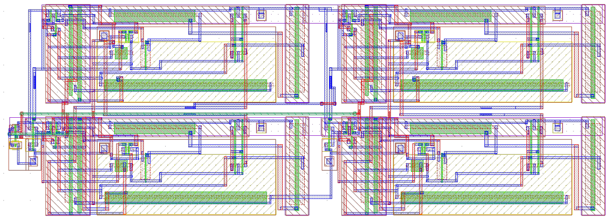


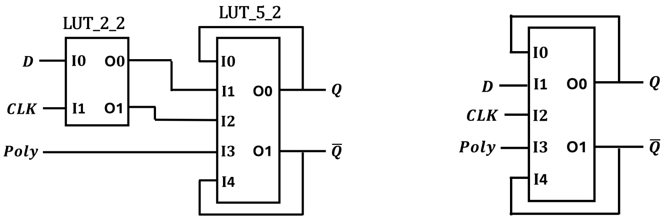
Fig. 9: Layout view of GA-optimized Non-MTNCL-based polymorphic register with two Non-MTNCL-based latches connected in master slave configuration. Areas are measured as 10.5µm × 7.8µm for latch and 22.1µm × 7.8µm for register.

data protection within an FPGA. A look-up-table (LUT) approach for implementing polymorphic latches in reconfigurable circuits is presented in this section. The design can be implemented with 2 LUTs or 1 LUT to meet design needs. While the LUTs themselves cannot exhibit sense-and-response polymorphic behavior, they can be combined with on-chip sensors to effectively counter physical attacks. The following subsections outline the approach of implementing a polymorphic latch by means of explicit LUT instantiation. These methods are synthesized using Xilinx Vivado 2022.2.

A. 2-LUT Polymorphic Latch

The polymorphic latch design can be implemented within an FPGA by mapping the gates from Fig 4 into two LUTs. First, the logical behavior of the inverter and AND gates is collapsed into a single two-input, two-output LUT (LUT_2_2). The inputs to this LUT are the CLK and D signals, and the outputs each feed into the inputs of polymorphic NAND/NOR gates. The two polymorphic NAND/NOR gates are mapped to a single five-input, two-output LUT (LUT_5_2). Two of the inputs to this LUT, one for each polymorphic gate, are the outputs of the LUT_2_2. A third input is a primary input, $Poly$, which is used to control whether the outputs use NAND or NOR behavior. This polymorphic control signal is used to trigger the forbidden state, and can be connected to an external trigger source such as a voltage or temperature sensor. The outputs of the LUT_5_2 are representative of the latch signals Q and $\overline{Q}$. These outputs are also fed back in loop configuration to the final two inputs of the LUT.

The delay in the paths between $Q/\overline{Q}$ and the LUT inputs controls the state in which the 2-LUT polymorphic latch returns when the $Poly$ signal is eventually deasserted – referred to here as the restoration state of the latch. When the Q feedback path has greater delay than the $\overline{Q}$ feedback path, the latch restores to a 1/0 output state. On the contrary, higher delay in the $\overline{Q}$ feedback path corresponds to a 0/1 restoration state. By adding delays to these feedback paths, the designer can control what the latch data restores to after the forbidden state is exited. A schematic representation of the 2-LUT polymorphic latch is shown in Fig. 10(a).



(a) (b)

Fig. 10: Implementation of polymorphic latch in FPGA: (a) 2-LUT implementation (b) 1-LUT implementation.

B. 1-LUT Polymorphic Latch

A much more efficient implementation of the polymorphic latch can be achieved by use of a single LUT. This 1-LUT latch is constructed by programming a five-input, two-output LUT (LUT_5_2) with the state table of the polymorphic latch. Two of the LUT inputs are connected to the CLK and D signals. The $Poly$ signal, a control signal used to switch between normal operation and the forbidden state, is also a LUT input. Finally, two inputs are reserved to feed the outputs, Q and $\overline{Q}$, back in as inputs. The schematic representation of the 1-LUT polymorphic latch is shown in Fig. 10(b).

An example state table for the 1-LUT polymorphic latch is shown in Table VII. Two methods of customization exist for this latch. Firstly, the forbidden state can be customized on a per-latch basis to be either $Q = \overline{Q} = 1$ or $Q = \overline{Q} = 0$. This is controlled by changing the bits programmed into the LUT truth table for the $Poly = 1$ rows – this corresponds to the **yellow** row in Table VII. The second method of customization is the state in which the 1-LUT polymorphic latch returns when the $Poly$ signal is deasserted – the restoration state of the latch. This is controlled by changing the bits programmed for the rows in the LUT truth table where $Poly = 0$ and $Q = \overline{Q}$. Thus, when a latch instance is in the forbidden state, but $Poly$ is

TABLE VII: State table of 1-LUT polymorphic latch. The forbidden state is highlighted in yellow. The restoration states are highlighted in green. 'X' denotes a don't care.

$Poly$	Q_-	$\overline{Q}_-$	CLK	D	Q_+	$\overline{Q}_+$
0	0	0	X	X	0	1
0	0	1	0	X	0	1
0	0	1	1	0	0	1
0	0	1	1	1	1	0
0	1	0	0	X	1	0
0	1	0	1	0	0	1
0	1	0	1	1	1	0
0	1	1	0	X	1	0
0	1	1	1	0	0	1
0	1	1	1	1	1	0
1	X	X	X	X	1	1

deasserted, the contents are restored to a specific, customizable value. The restoration states –one for the $Q = \overline{Q} = 1$ forbidden state and one for the $Q = \overline{Q} = 0$ forbidden state – are highlighted as **green** in Table VII.

Easy control of the restoration states of the 1-LUT latch, especially compared to the 2-LUT latch's need for controlling feedback delays, is a definite advantage of this approach. Another major advantage of the 1-LUT latch is that, the forbidden state can be entered depending only on the $Poly$ signal being 1. The 1-LUT latch therefore eliminates the requirement of the CLK being in the 0 state.

C. Supporting Circuitry for LUT-based Latches

While the 1-LUT and 2-LUT polymorphic latches can implement destructive logic-based memory in FPGAs, they must be combined with on-chip or external sensors. Sensors should be instantiated to drive the $Poly$ signal high when the conditions of a physical attack are detected. For the experiments in this article, we utilize the Xilinx XADC IP as a temperature and voltage sensor (TVS). The XADC can be configured as a TVS by making use of the programmable internal voltage (VCCINT) alarm and the programmable temperature alarm. These signals are each driven high when their respective conditions exceed some user-programmed value. For example, the VCCINT alarm can be programmed to 0.9V to counter LLSI attack. By making use of a TVS, voltage can be monitored and used to trigger the LUT-based polymorphic latches.

For the 2-LUT latch, the CLK input must be held at 0 in order for the forbidden state to be entered. The latch clock source should be routed through clock gating logic, such as an AND gate with an inverted input, so that the CLK signal is not passed when the alarm is triggered.

VIII. Results and Discussion

A. Simulation Setup

Cadence Virtuoso is used for simulation purposes and all the designs were implemented in $45nm$ technology node using the $gpdk045$ library.

To get accurate simulation results for each cell, parasitic extraction is performed using Quantus Extraction Solution version 22.1.0-p089. This calculates all net resistances and capacitances, as well as all MOSFET channel resistances between drain and source regions. Extracted cells are then simulated using Cadence Virtuoso ADE to verify that they exhibit normal behavior under 1.1 V operation and polymorphic behavior under 0.55 V operation.

Cadence Virtuoso ADE is also used to measure the performance, power, and area (PPA) characteristics of each latch and register. Transient simulations are run to characterize the timing requirements of setup and hold time. Similarly, we utilize transient simulations to reproduce various device states (switching activity, storing of 0/1 data), which are then used to extract nominal delays and power measurements. The areas of the latch and register are also determined from the final layout view.

To assess the reliability of the latch and register, Monte Carlo analysis is conducted in Cadence Virtuoso's ADE XL simulation environment. Both process variation and transistor mismatch are accounted for in the Monte Carlo analysis, at operating points of 1.1V and 0.55V. The 1.1V operating point results indicate functional yield, while 0.55V operating point results reveal the ability of each design to reliably react to attack conditions. Finally, a temperature sweep is performed in Cadence Virtuoso ADE to analyze the resilience of the latch and register to temperature fluctuations. The worst-case destruction delay for both the latch and register is determined during the temperature sweep.

B. Polymorphic Latch Simulation

1) Power, Performance, and Area (PPA) Overhead: The results of parasitic extraction allow the latch to be simulated with inclusion of all relevant RC delays. The ADE simulation tool in Cadence Virtuoso is then used to conduct transient simulation, allowing the power, performance, and are (PPA) characteristics to be determined. The PPA results are extracted for all designs – standard NOR latch, standard NAND latch, the MTNCL-based polymorphic latch from [26], GA-optimized MTNCL-based polymorphic latch and the GA-optimized Non-MTNCL-based polymorphic latch. The results of this analysis are shown in Table VIII. All PPA measurements are simulated under nominal conditions at 27°C and standard 1.1V operation.

CLK2Q Delay: The clock-to-Q and clock-to-$\overline{Q}$ timings are obtained by analyzing the propagation delay from a rising clock edge to valid data being latched to the output.

D2Q Delay: The D-to-Q and D-to-$\overline{Q}$ timings are similarly obtained by measuring the time for data to be latched to the output when the clock is held high and the D input changes.

Setup Time is measured by determining the earliest that data can arrive relative to the falling clock edge and still be propagated to Q within 5% of nominal D-to-Q delay.

Hold Time is measured by determining how long data that arrives at the minimum setup time must be held after the falling clock edge in order to be properly latched. If data can be properly latched even if it changes before the falling clock edge then the hold time is negative.

Power: Power measurements are collected by running transient simulations up to 10µs in length. Minimum static power refers to the power consumption of the latch when there is no switching activity and the latch does not store any data. Average power refers to the average power consumption when the device experiences regular switching activity. Peak power consumption is the highest recorded power value observed during signal transitions in simulation.

As can be observed in Table VIII, the MTNCL-based polymorphic latch carries significantly lower PPA overhead than the Non-MTNCL-based polymorphic latch. The area of the MTNCL-based latch is over 3× lower than that of the Non-MTNCL-based latch. Additionally, the timing measurements indicate that the MTNCL-based latch can easily operate at GHz speeds, while the Non-MTNCL-based latch can only support a clock in the MHz range. Both latches have similar power characteristics, which are higher than standard NAND and NOR latches. With these measurements in hand, it is suggested to utilize the MTNCL-based latch for timing or area-sensitive application and the Non-MTNCL-based latch for reliability-sensitive application as will be explained in process variation simulations.

TABLE VIII: Latches: power, performance, and area (PPA) comparison. (1.1V, 27°C).

Parameters	NOR Latch	NAND Latch	MTNCL-based Polymorphic Latch in [26]	GA-optimized MTNCL-based Polymorphic Latch	Non-MTNCL-based Polymorphic Latch
Area	7.8 µm^2	7.1 µm^2	20.3 µm^2	22.0 µm^2	80.8 µm^2
$CLK2Q$ delay (rise/fall)	170 ps / 95 ps	76 ps / 109 ps	320 ps / 178 ps	257 ps / 174 ps	739 ps / 1065 ps
$CLK2\overline{Q}$ delay (rise/fall)	177 ps / 90 ps	71 ps / 106 ps	298 ps / 180 ps	248 ps / 190 ps	746 ps / 1023 ps
$D2Q$ delay (rise/fall)	171 ps / 117 ps	70 ps / 131 ps	321 ps / 205 ps	251 ps / 187 ps	681 ps / 1112 ps
$D2\overline{Q}$ delay (rise/fall)	200 ps / 95 ps	93 ps / 103 ps	326 ps / 186 ps	260 ps / 184 ps	767 ps / 1008 ps
Setup Time	79 ps	64 ps	155 ps	140 ps	1050 ps
Hold Time	4 ps	8 ps	23 ps	18 ps	-134 ps
Minimum Static Power	32 pW	47 pW	82 pW	201 pW	3 uW
Average Power	904 nW	808 nW	25 uW	25 uW	6.5 uW
Peak Power	73 uW	79 uW	141 uW	130 uW	77 uW

TABLE IX: Latches: Reliability analysis under worst case of normal operation (V_{DD} lowered by 10% at 84°C) and under attack condition (V_{DD} lowered to $0.55V$) with Monte Carlo simulation.

Design	Input Pattern	$V_{DD} = 1V$ at 84°C		$V_{DD} = 0.55V$	
	D	Data Retained	Data Destroyed	Data Flipped	Data Retained
NOR Latch	High	100%	0%	0%	100%
	Low	100%	0%	0%	100%
NAND Latch	High	100%	0%	0%	100%
	Low	100%	0%	0%	100%
Polymorphic Latch from [26]	High	100%	86.5%	2%	11.5%
	Low	100%	86.5%	3.5%	10%
GA-optimized MTNCL-based Polymorphic Latch	High	100%	85%	0%	15%
	Low	100%	73%	10.5%	16.5%
GA-optimized Non-MTNCL-based Polymorphic Latch	High	100%	100%	0%	0%
	Low	100%	100%	0%	0%

The addition of the PMOS $PM4$ to the NOR/NAND gates of the GA-optimized MTNCL-based polymorphic latch increases its area cost by 8.5% compared to the MTNCL-based latch in [26] as shown in Table VIII. However, this PMOS improves the response time of the latch to destroy data by quickly discharging the gate charge of $NM2$ and allowing the pull-down network to switch the circuit functionality. While the polymorphic latch from [26] is capable of destroying its stored data in 364 µs at 27°C, the GA-optimized MTNCL-based polymorphic latch does that in 112 ns. This represents about more than 3000× improvement in the response time for the GA-optimized MTNCL latch over the non-optimized one in [26].

2) Impacts of Temperature: The performance of the latch is measured across a range of temperatures to verify stability and to measure the worst-case time to enter the forbidden state. According to the threat model, countermeasures to active physical attacks should react as quickly as possible to prevent data from being extracted. The temperature simulations are performed for all temperature points between 0°C and 84°C. First, the simulation latches data into the latch to verify standard operation. Then, the supply voltage is lowered from 1.1V to 0.55V while the CLK signal is held at 0. The response time is measured as the time from the drop in voltage to the forbidden state ($Q = \overline{Q}$) being entered.

The GA-optimized MTNCL-based polymorphic latch is stable and exhibits the correct 1.1V behavior for all temperature points. Furthermore, the latch exhibits a worst-case delay of 125ns at 34°C while the room temperature delay is 112ns. The Non-MTNCL-based polymorphic latch similarly enters the forbidden state with a worst-case delay of 146ns at 84°C with the room temperature delay of 84ns. However, the Non-MTNCL-based polymorphic latch with the listed transistor sizings fails to operate normally at temperatures less than 9°C. It should be noted that this is dependent on transistor sizing: the operational range of the Non-MTNCL-based latch can be easily extended to lower temperatures by increasing the sizing of transistors $PM1$ and $NM1$. A minimum operational temperature of 0°C can be achieved by increasing the width of $PM1$ to 4.7µm and the width of $NM1$ to 2µm. Thus, the Non-MTNCL-based latch forces a trade off between operational temperature range and area.

With these results, it is confirmed that both the GA-optimized MTNCL and Non-MTNCL latch designs can reliably protect against voltage manipulation-based active physical attacks, even under variation of device temperature. Additionally, with respective worst-case response times of 125ns and 146ns, the MTNCL-based and Non-MTNCL-based latches are proven to be able to react quick enough to counteract voltage manipulation-based attacks such as LLSI, TLS, or VFI.

3) Effects of Process, Voltage, and Temperature (PVT) Variation: The latches must be able to latch data at 1V (10% lower than nominal voltage to account for sudden noise-based lowering) and enter the forbidden state at 0.55V at elevated temperature of 84°C representing the worst case normal operation scenario. Monte Carlo simulations are run in Cadence Virtuoso with 400 simulation points, including both process variation and transistor mismatch, to verify the reliability of the latches and the results are summarized in Table IX.

First, the Monte Carlo simulation is run at worst case condition. This simulation tests whether each latch can capture both 0 and 1 states, and effectively represents functional yield at worse case. Under 1V and 84°C operation, all latch designs are able to latch and hold data for 100% of simulation points. This is expected for standard latches, but also illustrates that the polymorphic latches are stable and can be reliably deployed in a design.

The second Monte Carlo simulation is run under the the conditions of a simulated attack. Both input sequences for input D logic $high$ and logic low is latched at 1V, and then the supply voltage is lowered to 0.55V with the CLK signal locked to 0. The output of the latch is monitored for the forbidden state where both outputs Q and its complement $\overline{Q}$ enters the same logic level. This test captures the ability of the latch to effectively destroy data under lowered supply voltage. The results of this Monte Carlo simulation are shown in Table IX. The standard NOR and NAND latches do not have destruction capability and retains data 100% of the times. Consequently, the data in the standard latches is completely susceptible to attacks according to the threat model. The GA-optimized MTNCL-based latch enters the forbidden state for 85% of test points for logic $high$ and 73% of test points for logic low inputs. Thus the GA-optimized MTNCL latch destroys data 79% of the times on an average. Further, on an average, 5% of test points show the data state of the latch being flipped, effectively obfuscating the original data. The attacker will not be able to determine whether the latch flipped the data or retained its state. Although compared to the latch presented in [26], proposed latch in this work is able to destroy data about 7.5% less number of times on average, the proposed optimized latch is about 3000 times faster in response time, which is crucial for security applications and makes the yield loss justified. The Non-MTNCL-based latch is more stable in the face of process variation and mismatch, entering the forbidden state for 100% of test points irrespective of input sequence. Also, the response time of Non-MTNCL-based design is comparable to that of MTNCL-based design although the Non-MTNCL design has much larger internal timing overhead in terms of setup time, hold time and clock to output delays. The Non-MTNCL-based polymorphic latch, while being more costly in terms of area and timing overhead, should therefore be considered for applications where it is critical that *all* data is reliably destroyed.

C. Polymorphic Register Simulation

1) Power, Performance and Area (PPA) Overhead: The simulation setup for characterizing the polymorphic registers is similar to that of the polymorphic latches. Cadence Virtuoso's ADE and ADEXL simulation environments are used for transient simulation of RC extracted layouts. All simulations are conducted under nominal conditions at 27°C and standard 1.1V operation. The PPA measurements for the standard and polymorphic registers are demonstrated in Table X. The measured characteristics are the same as those of the latches, listed in Section VIII-B1, with the exception of $D2Q$ Delay - this timing arc is not defined for registers and is omitted. The calculation of setup time and hold time for registers is different from latches.

Setup Time is measured by determining the earliest that data can arrive relative to the rising clock edge and still be propagated to Q within 5% of nominal CLK-to-Q delay.

Hold Time is measured by determining how long data that arrives at the minimum setup time must be held after the rising clock edge in order to be properly latched. If data can be properly latched even if it changes before the rising clock edge then the hold time is negative.

According to Table X, the GA-optimized MTNCL-based polymorphic register has significantly lower PPA overhead compared to the GA-optimized Non-MTNCL polymorphic register. Also, much like the latch, the timing measurements indicate that the MTNCL-based register is capable of supporting sub-ns clock periods permitting operation frequency in GHz range. The Non-MTNCL-based register struggles with timing and carries large delays as well as long setup and hold times limiting operating frequency in few hundred MHz. With comparable power requirements, it is suggested to utilize the MTNCL-based register where timing and area overhead are of concern. But as it will be observed in the Monte Carlo analysis, the Non-MTNCL-based polymorphic register should be chosen where reliability is paramount.

2) Impacts of Temperature: The performance of the registers is measured at various temperatures to obtain time-to-destruction metrics and to verify robust operation under fluctuating conditions. These temperature experiments are identical to those conducted for the latches. Temperatures between 0°C and 84°C are used as the simulation test points. The simulation transitions the clock to store some data in the latch and verify the output state. The supply voltage is lowered from 1.1V to 0.55V while the CLK signal is held at 0. The delay is measured as the time from the drop in voltage to the forbidden state being entered.

The GA-optimized MTNCL-based polymorphic register, at 1.1V supply voltage, clocks data correctly for all temperature points. The worst-case delay of this register is 125ns at 36°C. The room-temperature delay is 117ns. The worst-case delay of the GA-optimized Non-MTNCL-based register is 146ns at 84°C. The delay at room temperature is 83ns. This register, like its latch counterpart, only functions properly for temperatures 9°C or higher. The constituent latches can be resized as discussed before, to achieve stability at lower temperatures.

These results solidify that, like the polymorphic latches, both the MTNCL-based and Non-MTNCL-based registers can reliably protect against physical attacks at various temperature points. The reported response times are sufficient to destroy data before an attacker can read it out.

3) Reliability over PVT Variations: Reliability simulations are run for the registers, like for the latches, to verify that the device can withstand process variation and mismatch. The results are captured in Table XI.

Similar to the latch simulation first Monte Carlo simulation is done at worse case of normal operation ($1V$ and 84°C). All register designs are able to clock and hold data for 100% of the simulation points. The polymorphic registers thus do not have any normal yield issues even at worst case related to transistor mismatch or process variation and can be reliably be deployed in a design.

The second Monte Carlo simulation simulates attacks within our attack model. Both input sequences logic $High$ and logic Low is registered at $1V$, and then the supply voltage is lowered to 0.55V with the CLK at 0. The output of the register is monitored for the forbidden state. If both outputs Q and $\overline{Q}$ enters the same logic, it is termed as forbidden state. If the output enters forbidden state data is considered destroyed.

According to Table XI at attack condition, normal NOR and NAND registers do not have ability to destroy data and they retain data for all the cases. The GA-optimized MTNCL-based register enters the forbidden state for 68.6% of test points on average if both input sequences for D are considered. Logic flips on an average of 9.1% and data is not destroyed for 22.3% of the times. Logic flip denies the attacker of original data but the data retained condition poses potential yield loss. Compared to the register using MTNCL-based latches from [26], the GA-optimized MTNCL-based register is less reliable by 10%, have 8.5% more area overhead but has a response time of 117 ns compared to 355 µs which makes the GA-optimized MTNCL-based version a practical choice in security applications. The longer response time for [26]-based register design makes it impractical for security applications as it gives attacker ample time to carry out their attack justifying the 10% more yield loss and 8.5% more area overhead for the GA-optimized MTNCL-based register presented in this article.

The GA-optimized Non-MTNCL register enters the forbidden state for 100% of Monte Carlo test points, showing high reliability. Moreover, the response time of the GA-optimized Non-MTNCL-based register comparable to the response time of GA-optimized MTNCL-based register. The only drawback is the 2.5× area overhead compared to the MTNCL-based register design. This indicated that despite increased area overhead cost, the non-MTNCL-based design is suitable for highly sensitive applications where *all* bits of data must be destroyed.

TABLE X: Registers: power, performance, and area (PPA) comparison. (1.1V, 27°C).

Parameters	NOR Register	NAND Register	Register using Latches from [26]	MTNCL-based Polymorphic Register	Non-MTNCL-based Polymorphic Register
Area	18.2 µm^2	17.7 µm^2	45.1 µm^2	47.2 µm^2	170.9 µm^2
$CLK2Q$ delay (rise/fall)	184 ps / 99 ps	76 ps / 109 ps	320 ps / 188 ps	246 ps / 176 ps	768 ps / 1072 ps
$CLK2\overline{Q}$ delay (rise/fall)	193 ps / 98 ps	72 ps / 107 ps	320 ps / 184 ps	249 ps / 189 ps	775 ps / 1099 ps
Setup Time	255 ps	78 ps	427 ps	275 ps	1510 ps
Hold Time	-74 ps	-10 ps	-93 ps	-83 ps	330 ps
Minimum Static Power	2.7 nW	2.4 nW	46.5 uW	46 uW	12 uW
Average Power	2.2 uW	1.5 uW	52 uW	51 uW	28 uW
Peak Power	124 uW	116 uW	273 uW	185 uW	187 uW

TABLE XI: Registers: Reliability analysis under worst case of normal condition (V_{DD} lowered by 10% at 84°C) and under attack condition (V_{DD} lowered to $0.55V$) with Monte Carlo simulation.

Design	Input Pattern	$V_{DD} = 1V$ at 84°C	$V_{DD} = 0.55V$		
	D	Data Retained	Data Destroyed	Data Flipped	Data Retained
NOR Register	High	100%	0%	0%	100%
	Low	100%	0%	0%	100%
NAND Register	High	100%	0%	0%	100%
	Low	100%	0%	0%	100%
Polymorphic Register using latch from [26]	High	100%	79.5%	7.5%	13%
	Low	100%	79.25%	7.25%	13.5%
GA-optimized MTNCL-based Polymorphic Register	High	100%	70.75%	7.75%	21.5%
	Low	100%	66.5%	10.5%	23%
GA-optimized Non-MTNCL-based Polymorphic Register	High	100%	100%	0%	0%
	Low	100%	100%	0%	0%

TABLE XII: Overhead in terms of area, timing, power and corruption rate in terms of hamming distance (HD) achieved using proposed MTNCL-based and Non-MTNCL-based registers in crypto modules: AES 128, DES and RSA.

Benchmark		No. of Polymorphic Registers	Area Increase	Maximum Clock Frequency	Power Consumption Increase	Corruption Rate in HD (Avg. of 100 random plaintexts)
AES 128	MTNCL-based Register	128	7.9%	2 GHz	320%	53%
	Non-MTNCL-based Register	128	42%	580 MHz	256%	58%
DES 64	MTNCL-based Register	56	9%	1.94 GHz	164%	42%
	Non-MTNCL-based Register	56	45%	570 MHz	131%	47%
RSA 512	MTNCL-based Register	48	0.1%	1.29 GHz	161%	45%
	Non-MTNCL-based Register	48	0.7%	490 MHz	129%	50%

D. Simulation of Corruption in Cryptographic Benchmarks

Three cryptographic benchmarks AES-128 [47], DES-64 [48] and RSA-512 [49] are chosen and both MTNCL-based and Non-MTNCL-based polymorphic countermeasures are simulated by replacing the key registers by polymorphic registers. Here, iterative implementation of AES-128 and DES-64 is used where every clock cycle one round operation is performed. AES-128 undergoes 10 rounds, with a voltage glitch injected in the 8th round, while DES-64 has 16 rounds with a glitch in the final round. For RSA-512, a pipelined implementation with 16-bit registers is used, and the glitch is introduced at the start of the Montgomery multiplication step. In AES-128, both state and key are 128 bits, and 128 key registers are replaced by polymorphic registers. DES-64 has a 64-bit state, a 56-bit key leading to 56 polymorphic register replacements. In RSA-512, the message, public exponent, and modulus are each 16 bits, with a total of 48 polymorphic register replacements.

The area, power increase, maximum allowable frequency and corruption rate of resulting ciphertext in percent of hamming distance (HD) are provided in Table XII. The area increase with Non-MTNCL registers is significantly larger than with MTNCL registers across all benchmarks. In RSA, state and key registers occupy minimal area compared to the whole implementation, so replacing them with polymorphic versions incurs little overhead. However, replacement with polymorphic registers leads to notable power overhead in all cases, though Non-MTNCL registers have a comparatively lower impact. MTNCL registers support gigahertz operation due to their small setup time of $275ps$ compared to the larger setup time of $1510ps$ for Non-MTNCL registers. Maximum achievable frequency also depends on the critical path delay of the benchmark implementations.

The corruption rate is measured as the percent change in the HD between glitch-free ciphertext and glitch-induced ciphertext for 100 samples. The corruption rate is higher for Non-MTNCL registers in all benchmarks. In AES-128, the glitch propagates to the ciphertext after the 10th round through substitute byte, mix column, and add round-key operations. In DES-64, the glitch's avalanche effect is smaller but still affects the ciphertext through the final permutation step. In RSA-512, corruption spreads through Montgomery multiplication, affecting the ciphertext. Corruption increases as more registers are replaced with polymorphic ones. If area and timing constraints exist, MTNCL registers are preferable, whereas Non-MTNCL registers are suitable for lower power consumption and higher corruption rates. Power can be reduced further by decreasing the number of polymorphic registers at the cost of corruption rate.

E. Adjusting Threshold for Polymorphism

For the simulation purposes, the transistor sizes are chosen in such a way that the threshold for polymorphism is $780mv$. However, the proposed polymorphic latches and registers must be adaptable to various attack models, such as those presented in this work. The voltage at which that gate switches between NOR and NAND behavior, denoted previously as V_B, should be customizable by the designer. For example, to counter a TLS attack, the V_B should be set

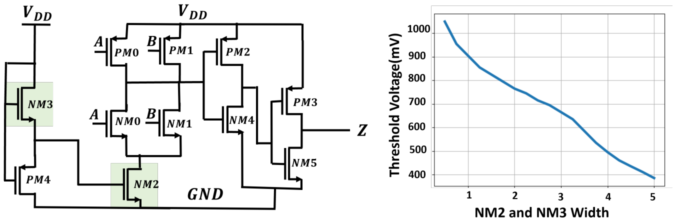


Fig. 11: Polymorphic voltage thresholds for various sizes of $NM2$ and $NM3$ pull-down transistors for MTNCL-based gate.

somewhere between the supply voltage and the brownout voltage. In this work, the countermeasures use a 45nm technology node with a nominal supply voltage of 1.1V and a brownout voltage of 550mV. The polymorphic threshold voltage for this case can be set in the range of 700-800mV. However, for the same technology, an LLSI attack might modulate with a peak-to-peak voltage of 400mV. To counter this, the polymorphic voltage V_B would need to be set between 900mV and 1V. In the case of a voltage glitch attack, the voltage drop is large enough that any threshold voltage V_B can be considered. However, for a voltage glitch, the countermeasure must react within $200ns$ - it was proven in the temperature sweeps that both MTNCL-based and non-MTNCL-based designs are capable of this where the designs from [26] cannot meet this criteria.

In the case of the MTNCL-based polymorphic latch and register, the polymorphic threshold V_B can be controlled by modifying the sizing of $NM2$ and $NM3$, the two gating NMOS devices in the pull-down network of each NOR/NAND gate. $NM2$ and $NM3$ should be equally sized, and their size is inversely proportional to the resulting V_B. Fig. 11 demonstrates the resultant polymorphic threshold voltage V_B for various sizes of $NM2$ and $NM3$. It can be seen that the decrease in polymorphic voltage is approximately linear with increase in transistor size. From Fig. 11, it is evident that $NM2$ and $NM3$ width of about 3.5 µm is suitable for TLS attack countermeasure where a width of about 0.5 µm is suitable as LLSI countermeasure. However, as $PM4$ is used to quickly release gate charge from $NM2$, it may be desirable to also increase the width of $PM4$ when the size of $NM3$ becomes larger. $NM3$'s larger gate will contain more charge, so $PM4$ should be accordingly scaled to increase discharge current and keep the response time of the countermeasure low.

In the case of Non-MTNCL-based polymorphic latches and registers, the NAND/NOR gate exhibits both an upper and a lower threshold. According to Fig. 12, the transistor $NM1$ predominantly affects the upper threshold, while $PM4$ has a greater influence on the lower threshold. Similar to the NOR/NAND gate, the upper threshold is inversely proportional to the width of $NM1$. Conversely, the lower threshold is directly proportional to the width of $PM4$. Fig. 12 illustrates the relationship between the widths of the relevant transistors and the upper and lower polymorphic threshold voltages of the design.

An width of 200 nanometers for $PM4$ and 2.5 micrometers for $NM1$ gives a polymorphism threshold of about $730mV$ according to Fig. 12 which is sufficient to protect against TLS. For the LLSI countermeasure, the width of $NM1$ should remain the same, whereas the width of $PM4$ needs to be increased to approximately 280 nanometers to meet the requirements. Different polymorphism thresholds can be achieved depending on the design needs by sizing $NM1$ and $PM4$ accordingly.

F. FPGA-Based Polymorphic Latch Results

The 1-LUT and 2-LUT polymorphic latches are tested in silicon using a Digilent Arty A7-35T development board, which makes use of the Xilinx Artix-35T FPGA. Synthesis, implementation, and bitstream generation are carried out in Xilinx Vivado 2022.2.

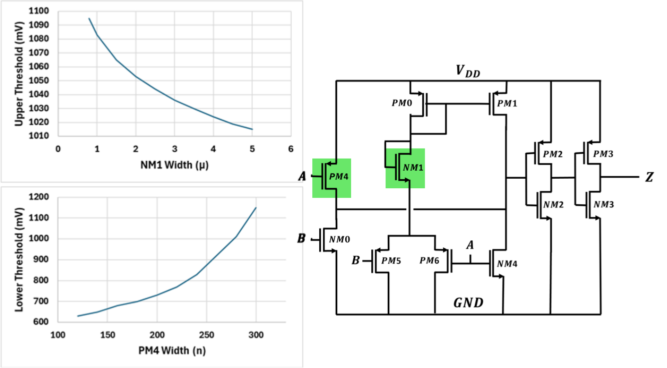


Fig. 12: Polymorphic voltage threshold for various sizes of $NM1$ and $PM4$ for GA-optimized Non-MTNCL polymorphic gate.

TABLE XIII: Simulation performance of 2-LUT and 1-LUT FPGA-based polymorphic latches.

Parameters	2-LUT	1-LUT
Resource Cost	2x LUT_6_2	1x LUT_6_2
$CLK2Q$ Delay (rise/fall)	2.37 ns / 2.43 ns	2.03 ns / 1.86 ns
$CLK2\overline{Q}$ Delay (rise/fall)	3.60 ns / 2.09 ns	2.03 ns / 1.86 ns
$D2Q$ Delay (rise/fall)	2.24 ns / 2.24 ns	1.79 ns / 1.62 ns
$D2\overline{Q}$ Delay (rise/fall)	3.30 ns / 1.79 ns	1.79 ns / 1.62 ns
Time-to-Destruction $(CLK = 0 \rightarrow Q = \overline{Q})$	2.09 ns	1.86 ns

First, the post-implementation timing simulation verifies both FPGA versions of the polymorphic latch. This uses the actual placement and routing information from the implementation results to carry out timing-accurate simulations. Table XIII compares the post-implementation simulation performance of the latches. Note that as 6-input and 2-output LUTs (LUT_6_2) are the core element of Xilinx logic slices, resource cost is noted accordingly. For example, the LUT_2_2 and the LUT_5_2 required for the 2-LUT latch are each implemented by Vivado as a LUT_6_2 with the extra inputs.

Because the 1-LUT latch has better timing characteristics, including a faster time-to-destruction, the 1-LUT latch is favorable when performance is needed. The 1-LUT latch also requires fewer resources than the 2-LUT latch. Furthermore, each instance of the 1-LUT latch can be individually customized to have either a $Q = \overline{Q} = 1$ or a $Q = \overline{Q} = 0$ forbidden state, and the restoration states can be customized as well. Finally, the 1-LUT latch is favorable due to the fact that it does not need the CLK to be stopped in order for the forbidden state to be triggered. The 2-LUT latch should only be considered if it is important for the latch to be easily disguised by placing the constituent LUTs far apart from each other. Otherwise, it is highly recommended to implement polymorphic latches in FPGA using the 1-LUT approach.

IX. Conclusion

In this work, polymorphic latches and registers have been proposed for protection against laser-based probing and non-invasive voltage fault injection attacks. The polymorphism-based response mechanism is also integrated with detection in the form of destruction of stored data upon detection of attack attempt. Compared to conventional countermeasures, this approach is faster in response after attack detection which does not give attacker enough time to carry out attack and steal sensitive information. Compared to other circuit-based detection countermeasures, this approach leads to reduced area and power overhead as it uses existing intellectual properties (IPs) already present in design (latches and resisters) with minimal modification. The proof-of-concept is provided in silicon using FPGAs. The next step involves polymorphic latch and register design improvement for area, power overhead and utilization of polymorphism for a wider variety of hardware security applications.

X. Acknowledgements

This effort was sponsored in part by NSF under grant numbers 2117349, 2150122, and 2150123. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

[1] M. E. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptogr.*, vol. 4, p. 15, 2020.

[2] M. Lipp, A. Kogler, D. F. Oswald, M. Schwarz, C. Easdon, C. Canella, and D. Gruss, "Platypus: Software-based power side-channel attacks on x86," *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 355–371, 2021.

[3] Y. Wang, R. Paccagnella, E. T. He, H. Shacham, C. W. Fletcher, and D. Kohlbrenner, "Hertzbleed: Turning power side-channel attacks into remote timing attacks on x86," *IEEE Micro*, vol. 43, pp. 19–27, 2023.

[4] S. Tajik, H. Lohrke, J.-P. Seifert, and C. Boit, "On the power of optical contactless probing: Attacking bitstream encryption of fpgas," *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:5060320

[5] C. Bozzato, R. Focardi, and F. Palmarini, "Shaping the glitch: Optimizing voltage fault injection attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, pp. 199–224, 2019.

[6] R. Buhren, H. N. Jacob, T. Krachenfels, and J.-P. Seifert, "One glitch to rule them all: Fault injection attacks against amd's secure encrypted virtualization," *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[7] L. Zussa, J.-M. Dutertre, J. Clédière, and A. Tria, "Power supply glitch induced faults on fpga: An in-depth analysis of the injection mechanism," *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*, pp. 110–115, 2013.

[8] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria, "Electromagnetic transient faults injection on a hardware and a software implementations of aes," *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 7–15, 2012.

[9] H. Lohrke, "Laser-based attacks on secure integrated circuit extracting and protecting sensitive information," Ph.D. dissertation, Technische Universitaet Berlin, 2019.

[10] J. Wang and B. H. Calhoun, "Canary replica feedback for near-drv standby vdd scaling in a 90nm sram," *2007 IEEE Custom Integrated Circuits Conference*, pp. 29–32, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:9111195

[11] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, and J.-P. Seifert, "Key extraction using thermal laser stimulation: A case study on xilinx ultrascale fpgas," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 573–595, 2018.

[12] T. Krachenfels, T. Kiyan, S. Tajik, and J.-P. Seifert, "Automatic extraction of secrets from the transistor jungle using laser-assisted side-channel attacks." in *USENIX Security Symposium*, 2021, pp. 627–644.

[13] B. Niu, G. M. E. Khoo, Y.-C. S. Chen, F. Chapman, D. Bockelman, and T. Tong, "Laser logic state imaging (llsi)," in *Proceedings from the 40th International Symposium for Testing and Failure Analysis (ISTFA 2014)*, 2014, p. 65.

[14] T. Krachenfels, F. Ganji, A. Moradi, S. Tajik, and J.-P. Seifert, "Real-world snapshots vs. theory: Questioning the t-probing security model," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1955–1971.

[15] Z. Chen, G. Vasilakis, K. Murdock, E. Dean, D. F. Oswald, and F. D. Garcia, "Voltpillager: Hardware-based fault injection attacks against intel sgx enclaves using the svid voltage scaling interface," in *USENIX Security Symposium*, 2021.

[16] C. Boit, S. Tajik, P. Scholz, E. Amini, A. Beyreuther, H. Lohrke, and J.-P. Seifert, "From ic debug to hardware security risk: The power of backside access and optical interaction," *2016 IEEE 23rd International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pp. 365–369, 2016.

[17] T. Mosavirik and S. Tajik, "Backmon: Ic backside tamper detection using on-chip impedance monitoring," *IACR Cryptol. ePrint Arch.*, vol. 2024, p. 631, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269589167

[18] S. Tajik, J. Fietkau, H. Lohrke, J.-P. Seifert, and C. Boit, "Pufmon: Security monitoring of fpgas using physically unclonable functions," *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 186–191, 2017.

[19] S. Roy, T. Farheen, S. Tajik, and D. Forte, "Self-timed sensors for detecting static optical side channel attacks," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2022, pp. 1–6.

[20] T. Farheen, S. Roy, S. Tajik, and D. Forte, "A twofold clock and voltage-based detection method for laser logic state imaging attack," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 1, pp. 65–78, 2022.

[21] M. Nagata, T. Miki, and N. Miura, "On-chip physical attack protection circuits for hardware security," in *2019 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2019, pp. 1–6.

[22] N. Miura, D. Fujimoto, D. Tanaka, Y.-i. Hayashi, N. Homma, T. Aoki, and M. Nagata, "A local em-analysis attack resistant cryptographic engine with fully-digital oscillator-based tamper-access sensor," in *2014 symposium on VLSI circuits digest of technical papers*. IEEE, 2014, pp. 1–2.

[23] S. Tada, Y. Yamashita, K. Matsuda, M. Nagata, K. Sakiyama, and N. Miura, "Design and concept proof of an inductive impulse self-destructor in sense-and-react countermeasure against physical attacks," *Japanese Journal of Applied Physics*, vol. 60, no. SB, p. SBBL01, 2021.

[24] A. Srivastava and P. Ghosh, "An efficient memory zeroization technique under side-channel attacks," in *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. IEEE, 2019, pp. 76–81.

[25] A. Askeland, S. Nikova, and V. Nikov, "Who watches the watchers: Attacking glitch detection circuits," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2024, pp. 157–179, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264557967

[26] A. Cannon, T. Farheen, S. Roy, S. Tajik, and D. Forte, "Protection against physical attacks through self-destructive polymorphic latch," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[27] R. Ruzicka, L. Sekanina, and R. Prokop, "Physical demonstration of polymorphic self-checking circuits," *2008 14th IEEE International On-Line Testing Symposium*, pp. 31–36, 2008.

[28] C. O'Flynn, "Fault injection using crowbars on embedded systems," *IACR Cryptol. ePrint Arch.*, vol. 2016, p. 810, 2016.

[29] H. Lohrke, "Laser-based attacks on secure integrated circuits," Ph.D. dissertation, Technische Universität Berlin, 2019.

[30] D. Nedospasov, J.-P. Seifert, C. Helfmeier, and C. Boit, "Invasive puf analysis," in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2013, pp. 30–38.

[31] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J. S. Krissler, C. Boit, and J.-P. Seifert, "Breaking and entering through the silicon," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 733–744.

[32] H. Wang, D. Forte, M. M. Tehranipoor, and Q. Shi, "Probing attacks on integrated circuits: Challenges and research opportunities," *IEEE Design & Test*, vol. 34, no. 5, pp. 63–71, 2017.

[33] H. Shen, N. Asadizanjani, M. Tehranipoor, and D. Forte, "Nanopyramid: An optical scrambler against backside probing attacks," in *ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis*. ASM International, 2018, p. 280.

[34] M. A. Korhonen, P. Bo/Rgesen, K.-N. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[35] A. Stoica, R. Zebulum, and D. Keymeulen, "Polymorphic electronics," in *Evolvable Systems: From Biology to Hardware: 4th International Conference, ICES 2001 Tokyo, Japan, October 3–5, 2001 Proceedings 4*. Springer, 2001, pp. 291–302.

[36] J. Nevoral, R. Ruzicka, and V. Simek, "Cmos gates with second function," in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2018, pp. 82–87.

[37] L. Sekanina, R. Ruzicka, Z. Vaíek, V. Simek, and P. Hanáek, “Implementing a unique chip id on a reconfigurable polymorphic circuit,” *Inf. Technol. Control.*, vol. 42, pp. 7–14, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:32060146

[38] L. Sekanina, L. Starecek, and Z. Kotásek, “Novel logic circuits controlled by vdd,” *2006 IEEE Design and Diagnostics of Electronic Circuits and systems*, pp. 83–84, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:30797850

[39] R. Ruzicka, “New polymorphic nand/xor gate,” in *Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science - Volume 7*, 2007, p. 192–196. [Online]. Available: https://api.semanticscholar.org/CorpusID:61209346

[40] L. Sekanina, “Evolutionary design of gate-level polymorphic digital circuits,” in *EvoWorkshops*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:14102065

[41] C. Bernard, W. Bryant, R. Becker, and J. Di, “Design of asynchronous polymorphic logic gates for hardware security,” in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2021, pp. 1–5.

[42] J. Di and C. Bernard, “Asynchronous polymorphic logic gate design,” U.S. Patent 11 095 287B1, 8 17, 2021.

[43] S. Katoch, S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimedia Tools and Applications*, vol. 80, p. 8091–8126, 2021.

[44] A. McNulty, B. Ombuki-Berman, and A. Engelbrecht, “A comparative study of evolutionary algorithms and particle swarm optimization approaches for constrained multi-objective optimization problems,” *Swarm and Evolutionary Computation*, vol. 91, p. 101742, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650224002803

[45] H. Liang, W. Luo, and X. Wang, “Designing polymorphic circuits with evolutionary algorithm based on weighted sum method,” in *Evolvable Systems: From Biology to Hardware. ICES 2007. Lecture Notes in Computer Science*, vol. 4684. International Conference on Evolvable Systems (ICES), 2007, pp. 30–38.

[46] D. R. E. Gnad, F. Oboril, and M. B. Tahoori, “Voltage drop-based fault attacks on fpgas using valid bitstreams,” *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–7, 2017.

[47] H. Hadipour. (2019) Vhdl implementation of aes algorithm. [Online]. Available: https://github.com/hadipourh/AES-VHDL

[48] A. Clement. (2015) Data encryption standard - vhdl. [Online]. Available: https://github.com/audeberc/DES-VHDL

[49] E. C. Villar and J. C. Villar. (2011) High performance rsa 512 bit ipcore. [Online]. Available: https://opencores.org/projects/rsa$_5$12

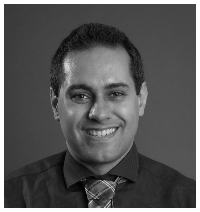
SOURAV ROY received a B.S. degree in electrical and electronics engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh with honors in 2011, and the M.S. degree in electrical, electronics and communication engineering from the Osaka University, Osaka, Japan in 2015. Currently he is pursuing Ph.D. in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. His research interest include hardware security, including chip backside attack detection and prevention, hardware analog trojan detection and prevention and chip counterfeit detection and prevention.

ANDREW CANNON received the B.S. and M.S. degrees in electrical engineering from the University of Florida, Gainesville, Florida, USA in 2023 and 2024, respectively. He is currently a Product Development Engineer at Advanced Micro Devices in Austin, TX, USA. His research focus is hardware security, primarily the design of polymorphic countermeasures against physical attacks.

LUIS DE LA MATA received the B.S. degree in telecommunication engineering from the Comillas Pontifical University, Madrid, Spain in 2022. He is currently pursuing an M.S. degree in telecommunication engineering and cybersecurity at the same university. Alongside his studies, he works as a Cybersecurity Consultant at Cross Mind Technology in Madrid, Spain. His research interests focus on hardware security and cybersecurity, particularly the application of AI techniques to enhance or develop security solutions in these domains.

RABIN YU ACHARYA currently works as a Reliability R&D Engineer at Intel. He graduated from University of Florida in 2023 where his thesis was on ”Design and Application of Evolutionary Algorithms for Hardware Security”. His research interests include reliability analysis, device characterization, AI-based hardware primitive design, and electronic design automation”.

TASNUVA FARHEEN received the B.S. degree in chemical engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 2018. Currently she is a Ph.D. student in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. Her research interests focus on the domain of hardware security, including physical backside attacks prevention and detection, anti-reverse engineering, SEM image analysis, FPGA implementation, simulation modeling and device fabrication.



SHAHIN TAJIK received the B.S. degree in electrical engineering from the K. N. Toosi University of Technology in 2010, and the M.S. and Ph.D. degrees in electrical engineering from the University of Berlin, in 2013 and 2017, respectively. He is currently an Assistant Professor with the Electrical and Computer Engineering Department of Worcester Polytechnic Institute (WPI), Worcester, MA. His field of research mainly includes non-invasive and semi-invasive side-channel analysis, Physically Unclonable Functions (PUFs), machine learning, FPGA security, and designing anti-tamper mechanisms against physical attacks. He has served as a reviewer for IEEE and ACM journals as well as a technical program committee member of many hardware security conferences, including Conference on Cryptographic Hardware and Embedded Systems (CHES), Symposium on Hardware Oriented Security and Trust (HOST), and Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC).

DOMENIC FORTE received the B.S. degree from the Manhattan College, Riverdale, NY, USA, in 2006, and the M.S. and Ph.D. degrees from the University of Maryland at College Park, College Park, MD, USA, in 2010 and 2013, respectively, all in electrical engineering. He is currently a Professor with the Electrical and Computer Engineering Department, University of Florida, Gainesville, FL, USA. His research interests include the domain of hardware security, including the investigation of hardware security primitives, hardware Trojan detection and prevention, electronics supply chain security, and anti-reverse engineering. He was a recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE), the Early Career Award for Scientists and Engineers (ECASE) by the Army Research Office (ARO), the NSF Faculty Early Career Development Program (CAREER) Award, and the ARO Young Investigator Award. His research has also been recognized through multiple best paper awards and nominations.