PolyWorm: Leveraging Polymorphic Behavior to Implant Hardware Trojans

Nimisha Limaye, *Graduate Student Member, IEEE,* Nikhil Rangarajan, *Member, IEEE,* Satwik Patnaik, *Member, IEEE,* Ozgur Sinanoglu, *Senior Member, IEEE,* and Kanad Basu, *Senior Member, IEEE*

Abstract—Owing to the various challenges looming over the integrated circuit (IC) supply chain for the sub-nm technology nodes, researchers are looking into emerging devices to design the next generation of high performance and low-power chips. The magnetoelectric spin-orbit (MESO) switch, developed by Intel™, is a front-runner in this search and is currently in a state of advanced experimental research. To ensure faster time-to-market and reap the existing supply chain's benefits, it is envisioned that the upcoming MESO devices will be coupled and hybridized with the existing complementary metal-oxide-semiconductor (CMOS) industrial framework. However, adopting the existing CMOS framework and incorporating it's outsourced supply chain increases exposure to various security threats such as design intellectual property (IP) piracy, overproduction of ICs, and insertion of hardware Trojans (HT) to leak information or cause denial-of-service. This paper proposes and investigates a stealthy HT insertion technique, *PolyWorm*, leveraging the polymorphic capabilities of MESO gates in hybrid MESO-CMOS architectures. We evaluate the efficacy of *PolyWorm* on ITC-99 benchmarks and demonstrate two use-case scenarios, viz., corrupting the intended design functionality and leaking the secret key from a cryptographic core. We also present a low-footprint domain wall-based trigger for our polymorphic Trojan to evade structural and power-based testing.

Index Terms—Hardware Trojans, emerging devices, polymorphic logic gates, magnetoelectric spin-orbit (MESO), domain wall device

1 Introduction

The aggressive dimensional scaling of complementary metal-oxide-semiconductor (CMOS) technology nodes has inundated power density constraints in modern microprocessors. Feature size miniaturization beyond the 7 nm node is intractable owing to numerous challenges ranging from increased gate leakage, more pronounced quantum mechanical effects resulting in reduced gate control, and thermal management issues that can drastically minimize the lifetime [1], [2]. The recent announcement by Intel™ to delay its next-generation 7 nm chips until the year 2022 exemplifies the challenges concerning yield and manufacturing complications [3]. Attempts at mitigating the plateauing of Moore's law include the adaptation of new transistor topologies, like the nanosheet transistor, a modification of today's FinFETs, which have the potential to pave the way for the 3 nm node in the near future [4]. Other research avenues include transitioning from the traditional two-dimensional (2D) chip layout to three-dimensional (3D) integrated circuit (IC) technology, to allow heterogeneous integration, reduction in wirelength and power budget with smaller die-outlines [5].

Arguably, one of the most prominent directions to alleviate these challenges is materials- and device-space exploration in the post-CMOS realm. The semiconductor industry invests heavily in novel computing paradigms driven by non-conventional devices from the spin and quantum domains. A promising candidate to arise out of this material and device research is the *magnetoelectric spin-orbit (MESO)* switch, developed by Intel™ [6]. Features like ultra-low power operation, low switching energy, small area footprint, competitive delay metrics, high integration density, and compatibility with existing CMOS processes have propelled the prototyping, testing, and characterization of this device [6]. Further, Intel™ has showcased MESO logic families [7] at the circuit- and system-level, which can pave the way for future MESO chiplets. At the experimental side, READ and WRITE blocks of the MESO device have also been demonstrated [8], with further ongoing research to achieve fabrication maturity.

It is widely regarded that any novel device of the future must be built on the existing industrial CMOS framework to be economically feasible [9]. Along the same lines, it is fair to assume that any commercial product borne out of the current device research would be a hybrid CMOS contraption. With a relatively straight-forward integration with CMOS in the back-end-of-line [6], it is anticipated that hybrid MESO-CMOS circuits, such as the ones presented in [7], might not be a distant reality [10]–[13]. *Hence, there is a need to investigate the possible vulnerabilities in the supply chain of such hybrid CMOS circuits.* Typically, the CMOS IC supply

- *Nimisha Limaye is with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: nimisha.limaye@nyu.edu).*
- *Nikhil Rangarajan and Ozgur Sinanoglu are with the Division of Engineering, New York University Abu Dhabi, Abu Dhabi, 129188 UAE (e-mail: nikhil.rangarajan@nyu.edu; ozgursin@nyu.edu).*
- *Satwik Patnaik was with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, USA. He is currently with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: satwik.patnaik@tamu.edu).*
- *Kanad Basu is with the Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: kanad.basu@utdallas.edu).*

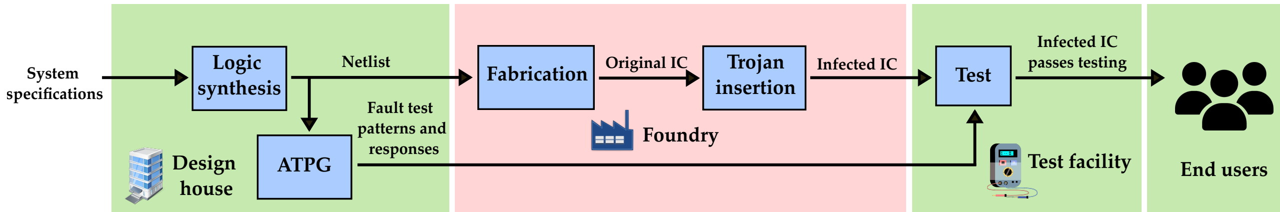


Fig. 1. Threat model landscape considered in this work for the insertion of hardware Trojans. Foundry (shown in red) is considered untrusted and is assumed to possess capabilities required for inserting a stealthy malicious hardware Trojan in the underlying design. The design house, test facility, and the end users (shown in green) are trusted entities.

chain faces hardware Trojan (HT) insertion threats with the inclusion of an untrusted foundry; this threat landscape can be extended to prevail in the context of hybrid MESO-CMOS circuit fabrication as well. In this case, the untrusted foundry can modify the gate-level netlist, which it receives from the design house, to insert the malicious Trojan. The untrusted foundry is also capable of obtaining the test patterns used to structurally test the chip – same as the test patterns given by the design house to the trusted test facility. All the other supply chain entities, viz., design house, test facility, and the end user, are assumed to be trusted, as illustrated in Fig. 1.

In this paper, we demonstrate *PolyWorm*, an HT insertion technique in hybrid CMOS designs composed of MESO gates. The MESO device's polymorphic capabilities aid the attacker in masking a potentially malicious piece of hardware as part of the original design, or in other words, "*a wolf in sheep's clothing*." The attack models and case studies developed in this paper apply to any novel polymorphic logic device compatible with CMOS processes, without loss of generality.[1] The contributions are enumerated as follows.

1) We investigate the security vulnerabilities of hybrid MESO-CMOS ICs from the perspective of an untrusted foundry, for the first time. To that end, we demonstrate how a malicious employee in an untrusted foundry can insert stealthy, malicious HT (*PolyWorm*) in the presence of other trusted entities (test facility and end users).
2) We design a stealthy trigger for *PolyWorm* using domain wall (DW)-based counters with low power and area footprint to circumvent power-based detection. Further, we identify the optimal circuit nets for stealthy trigger placement based on the frequency of net activation during IC testing, which helps thwart structural testing-based detection. Finally, we incorporate the payload using the existing MESO logic to avoid overheads and determine a transformation scheme for the payload gates, depending on the level of the trigger signal.
3) We verify the efficacy of *PolyWorm* by analyzing the output corruption and Hamming distance on Trojan activation on ITC-99 benchmarks.
4) We demonstrate two practical applications of *PolyWorm* using suitable case studies that showcase denial-of-service in a *Gaussian blur filter* image processing IP and key retrieval attack from the *Advanced Encryption Standard (AES)* cipher.

1. For instance, the semiconductor-based reconfigurable magneto-logic gates proposed in [14] and experimentally demonstrated in [15] could also be viable candidates for the proposed attack.

The organization of the paper is as follows. Section II provides a background and summary of the prior work in the field of HTs. In Section III, we provide the threat model considered for this work which outlines the resources and the abilities of an attacker. Section IV introduces the MESO device model and demonstrates the implementation of complex logic gates using MESO. Section V describes the construction and working of the *PolyWorm* trigger and payload. Section VI explains the experimental setup, provides results, and highlights the resiliency of *PolyWorm* against Trojan detection techniques. Section VII demonstrates the applicability of the *PolyWorm* technique on practical applications which includes denial-of-service and key retrieval. In Section VIII, we provide a discussion on possible countermeasures to thwart our *PolyWorm* technique and finally present concluding remarks in Section IX.

2 Background on Hardware Trojans

An HT is a malicious piece of hardware that is deliberately inserted in a design, at some point in its supply chain, to cause unwanted behavioral changes during deployment. The purpose of such a malicious introduction could be manifold, ranging from functional modification, degradation of performance to denial of service, and leaking of secret keys [18]. An HT comprises two essential components, the trigger and the payload [19]. The trigger continually monitors specific signal nets or events in the design and generates a signal once a predefined condition is satisfied. This trigger signal then activates the Trojan payload, which executes the intended malicious activity. An attacker can design an HT as an always-ON circuit without a trigger; however, such instances become more prone to detection.

2.1 Taxonomy for Hardware Trojans

Authors in [16] present a comprehensive classification of HTs according to their physical, activation, and action characteristics. **Physical characteristics** refer to the particular mode of the hardware implementation of the HT. For instance, the functional category consists of HTs constructed by the addition or removal of gates/transistors, whereas the parametric category encompasses HTs that involve modifying existing wires and logic [17]. **Activation characteristics** define whether an external entity/signal activates the HT or when particular conditions are satisfied on the internal nodes of the underlying design. Finally, **action characteristics** are concerned with the modus operandi and target of the HT. This detailed taxonomy of HTs, expounded in [17], is

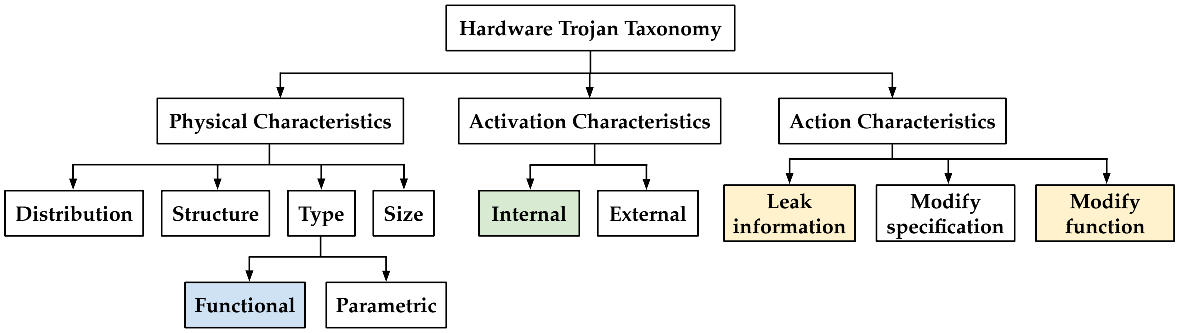


Fig. 2. Classification of hardware Trojans, according to [16], [17]. In this work, we focus on Trojans with the highlighted characteristics.

shown in Fig. 2. In this work, we consider a functional Trojan triggered using internal activation for leaking information (secret keys) and modifying functionality, as shown in Fig. 2.

2.2 Prior Work

Existing research in HT has mainly focused on novel trigger and payload mechanisms, design and performance optimization of HTs, stealthy HT placement techniques, and HT detection and prevention countermeasures. Various HT approaches already exist in the literature, as outlined next.

A differential fault-analysis-based HT, targeting cryptographic IP cores like the Advanced Encryption Standard (AES), was proposed in [20]. Chakraborty *et al.* [21] explored the implications of a software-based HT insertion attack on field-programmable gate array (FPGA) chips. The attack exploited the dynamic reconfigurability of FPGAs to modify the configuration bit-streams of the chip to partially reconfigure specific gates and wires to create an HT. Ender *et al.* [22] proposed a side-channel-based HT model for attacking application-specific integrated circuits (ASIC) by modifying transistor parameters without adding or removing logic. It utilizes a secure masked hardware implementation, where the modified transistors do not leak any detectable side-channel signals under regular operation unless they are operated at a specific frequency (trigger). The authors in [23] showcased a class of HT attacks on hardware neural networks, aimed at causing incorrect classification by changing the network weights and connections.

HT detection mechanisms were also proposed to deter such HT insertion techniques. Researchers have proposed several detection mechanisms to deter HT insertion techniques. Mechanisms include (i) designing the IP with high placement density [20], (ii) path delay fingerprinting [19], (iii) logic testing [24], (iv) side-channel analysis [25], and (v) statistical techniques like multiple excitation of rare logic conditions at internal nodes [26].

In contrast to prior CMOS-based HT insertion research, this work considers the HT threat in emerging devices, particularly in MESO-based hybrid CMOS circuits, for the first time.[2]

2. The *PolyWorm* technique focuses specifically on the Trojan threat in emerging MESO-CMOS hybrid circuits. However, the methodology, results, and conclusions demonstrated in this work apply to any CMOS-compatible polymorphic logic device without losing generality. It is also applicable to purely CMOS-based polymorphic circuits realized using look-up tables (LUTs) or multiplexers.

3 Threat Model

An attacker aiming to insert a malicious Trojan in a MESO-CMOS hybrid circuit has access to the following resources.

1) The attacker must be able to modify the GDSII file pre-fabrication. He/she resides at an untrusted foundry.
2) He/she must be able to extract the reverse-engineered netlist from the GDSII file. Using this netlist, he/she can obtain the exact test patterns to be used by the trusted test facility during structural testing of the chip.

With these resources, an attacker can successfully insert a HT in a MESO-CMOS hybrid design, and can consequently trigger the Trojan in-field to leak secret data or cause denial-of-service, by colluding with an end user.

4 MESO: Device Model and Complex Logic

In this section we present a brief description of the polymorphic spin device that will be leveraged in this work, namely the MESO device demonstrated by Manipatruni *et al.* [6].

4.1 Construction and Operation

MESO device works on the principle of the inverse Rashba-Edelstein effect (IREE) and inverse spin Hall effect (ISHE) exhibited by large spin-orbit coupling (SOC) materials [27], [28] and the magnetoelectric (ME) field-based switching effect prevalent in multiferroics [29]. Large SOC materials with a high spin-to-charge conversion efficiency, including topological insulators like Bi_2Se_3, 2D materials like MoS_2, and heavy metals like Pt, Pd, Ta, etc. are preferred for generating the output signals, which are in the electrical current domain. Multiferroics such as $BiFeO_3$, which possess a room-temperature ME effect, enable an all-electrical manipulation of the magnetization state of a connected ferromagnet. These two phenomena used in conjunction can implement the write and readout stages of the MESO logic.

The operation of the MESO device, shown in Fig. 3, is as follows. In the write stage, an input charge current flowing in the non-magnetic interconnect, in the $\pm\hat{x}$ direction, creates a charge separation and subsequently sets up an ME field along $\mp\hat{z}$ in the ME capacitor layer (purple). The ME field then switches the magnetization of the ferromagnet layer (blue) along $\mp\hat{y}$. Hence, the direction of the input charge current dictates the final orientation of the ferromagnet's state. In the read stage, the voltages on the terminals V_+ and V_- are turned on. This results in a

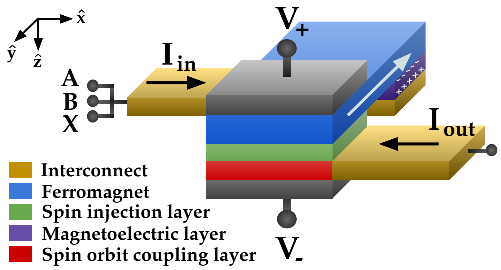


Fig. 3. Construction of the MESO device. An input charge current in the interconnect layer induces a magnetoelectric field in the magnetoelectric layer, which then switches the free ferromagnet. Voltages applied on the V_+/V_- terminals result in spin injection into the spin-orbit coupling (SOC) layer, which transduces the spin-polarized current into an output electric current.

spin-polarized current generated by the spin injection layer getting injected into the SOC layer. The SOC layer is then responsible for transducing the spin-polarized current into an output electric current (through IREE and ISHE), carried in the output interconnect layer. The polarity of the voltages on terminals V_+ and V_-, and the final orientation of the magnetization of the ferromagnet determine the polarity of the spin current generated in the spin injection layer, and hence the direction of the output electric current.

4.2 Complex Logic and Polymorphic Reconfiguration

The logic implemented is dictated by the direction of the output electric current relative to the input current. The logic levels "0" and "1" correspond to "-I" and "+I", in terms of the output current's directionality. In the case of Fig. 3, the output current direction is opposite to that of the input current; hence, the MESO device functions as an INV (here, B and X are dummy wires and the output is $\overline{A}$). *The functionality of this MESO INV can be morphed into a BUF, post-fabrication and during run-time, by interchanging the polarity of voltages on terminals V_+ and V_-*. To implement complex logic gates such as NAND/NOR, the primary inputs A and B along with a control signal X are fed into the input interconnect layer. Here, X acts as the tie-breaking signal and its polarity determines the logic operation. $X = +I$ results in NAND operation, whereas $X = -I$ furnishes a NOR. Note that here again, interchanging the polarity of voltages on terminals V_+ and V_- morphs NAND $\leftrightarrow$ AND and NOR $\leftrightarrow$ OR. Hence, the control signal X and the polarity of V_+ and V_- essentially act as knobs, which allows one to *reconfigure the gate functionality on-the-fly*. Figure 4 shows a NAND/NOR gate constructed using the charge-equivalent circuit representation of MESO [6], and Fig. 5 highlights the corresponding polymorphic transformation from NOR $\rightarrow$ NAND upon changing the polarity of the control signal X. The configuration of the various voltage and control inputs required to obtain different logic functionalities in the MESO device is presented in Fig. 6.

After understanding the construction and working of the reconfigurable MESO device, we next look into the application of this device in our *PolyWorm* technique.

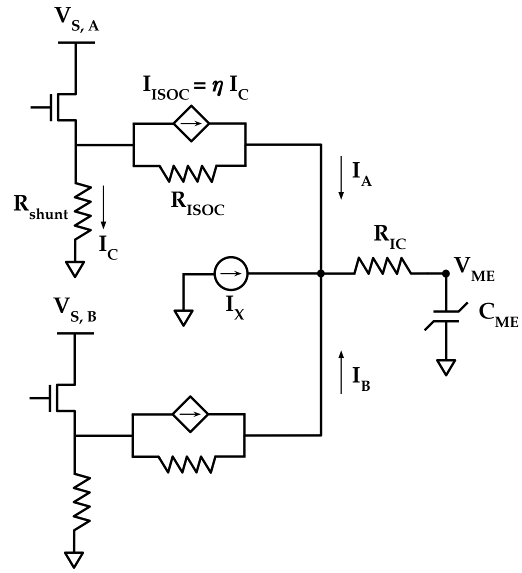


Fig. 4. Polymorphic NAND/NOR gate constructed with MESO charge equivalent circuit model.

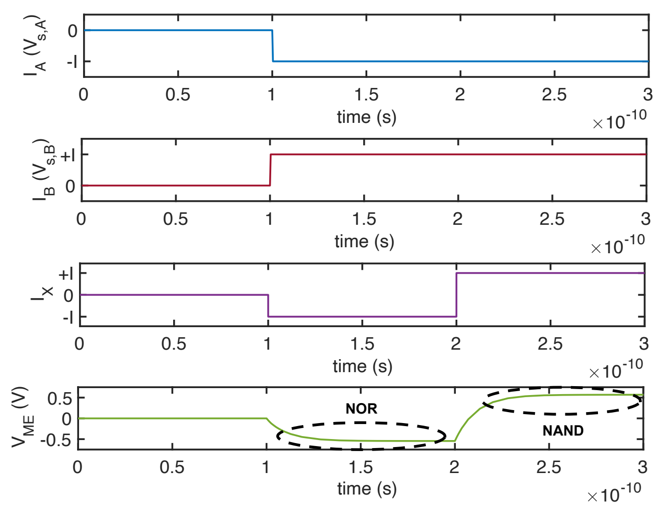


Fig. 5. Polymorphic functional transformation (obtained from behavioral simulations) of a MESO logic gate from NOR $\rightarrow$ NAND, on changing the control signal X from "-I" to "+I".

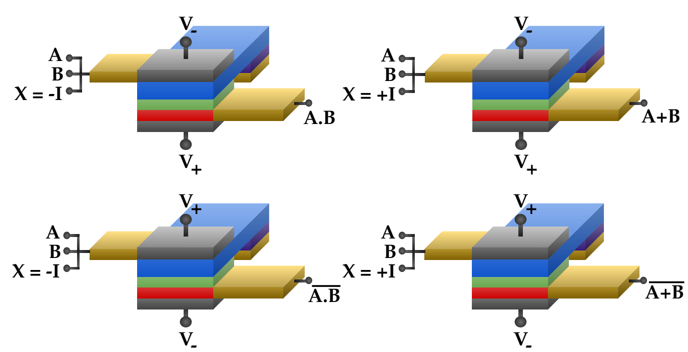


Fig. 6. Input and control signal configurations required to implement AND/OR/NAND/NOR logic functionalities using a MESO device. Signals A and B are logic inputs, and X is a control input.

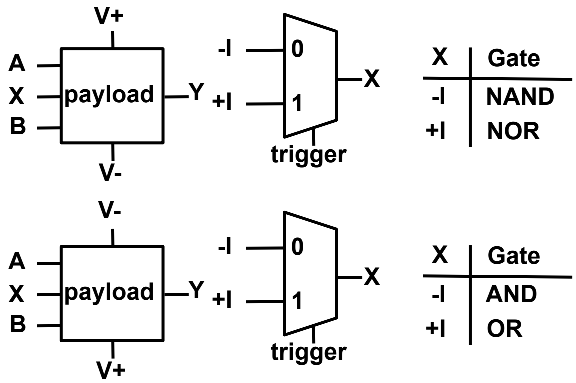


Fig. 7. Polymorphic gate configurations based on trigger value.

5 PolyWorm: Construction and Working

PolyWorm comprises a trigger and a payload circuitry as shown in Fig. 7. We incorporate the benefits of run-time reconfigurable behaviour of MESO devices by utilizing the existing MESO gates in the design as payload. For trigger circuit, we incorporate another emerging device, viz., domain wall device. Based on the trigger activation, MESO NAND gate can transform into MESO NOR gate and MESO AND gate into MESO OR gate as shown in Fig. 7. In this section, we dissect the trigger and payload components of the Trojan used in our *PolyWorm* technique.

5.1 Design of Trojan Trigger

Domain wall trigger design. For the purpose of activating *PolyWorm* when certain conditions are met in the circuit, we design a domain wall (DW) device-based trigger [31].[3] The DW device is used to construct a counter, as shown in Fig. 8(a). It consists of a long strip of a material that can support the formation and motion of a DW, for instance Permalloy nanowires [32], CoFe nanoconstrictions [33], ferroelectrics like $BiFeO_3$ [34], and Pt/Co/Pt epitaxial trilayers [35] etc. The DW strip forms the free layer of a Magnetic Tunnel Junction (MTJ) arrangement that is positioned at one end of the strip. The operation of this trigger is explained next. The DW, which is initially near the right end of the strip (adjoining terminal T1), starts moving towards the left whenever the wire connected to T1 is activated. The distance by which the DW moves on each activation, or its step length, depends on the DW velocity in the strip and the duration for which the input wire is activated. After a certain number of steps, the DW reaches the left end of the strip and the MTJ now has an anti-parallel configuration, as seen in Fig. 8(a). Hence, there is a spike in the MTJ junction resistance, which represents the firing of the trigger signal.

Considering an application of ATPG-generated test vectors at a frequency of 1.68 GHz, the node T1 can get activated once every ∼0.6 ns, assuming all of the testing patterns activate this particular wire T1. For a DW strip composed of Pt/Co/Pt trilayer, which can support DW velocity ∼20 m/s, a 10-bit counter (counting 0-1023), for example, can be constructed with a strip of length ∼12.19 μm. Here, the DW velocity and the frequency of node activation decides the length of the DW strip, for a fixed bit counter. The DW motion and the corresponding count at various positions of the DW, for a 10-bit counter constructed with a 12.19 μm Pt/Co/Pt strip, is illustrated in Fig. 8(b). We note here that the DW strip's material selection rationale is decided by its DW velocity, which directly affects the length of strip required and, hence, area overheads for the trigger. In material systems like Pt/Co/M trilayers (nonmagnetic layer M $\equiv$ Pt, Ir, Cu, Al), the DW velocity can be tuned by engineering the Dzyaloshinskii-Moriya interaction (DMI) strength, as demonstrated in [35]. This allows for optimization over the material-space for a given ATPG frequency and the required counter length, as shown in Fig. 9.

We note here that the DW structure used for the counter in Fig. 8(a), is a simple MTJ arrangement composed of standard ferromagnetic layers. The insertion of such a modified MTJ-DW structure by the attacker is not prohibitive since it uses materials and processes compatible with the MESO fabrication-capable foundry. Further, to account for any yield and reliability issues that may cause the DW trigger to remain dormant even upon activation, the attacker could fabricate redundant DW structures to trigger the same payload. Such a redundant scheme would not exacerbate the Trojan overheads or detectability, owing to the minimal area and power metrics for the DW counter (Table 2).

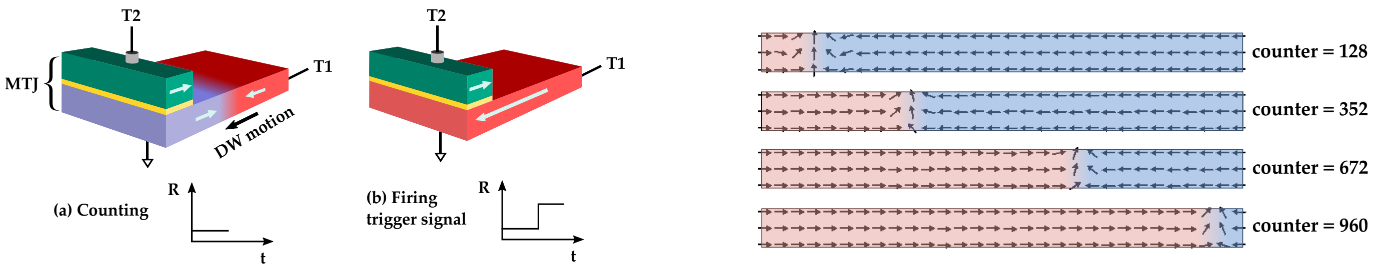
Trigger placement. Here, we focus on stealthy trigger placement of the DW device in the hybrid MESO-CMOS design. A stealthy trigger placement is critical to evade detection during the testing phase at a trusted test facility. An attacker at the foundry with access to test patterns can utilize them to construct its trigger. For each test pattern to be applied during structural testing, the attacker tabulates the respective values observed at the internal nodes of the design. The number of times a logic "1" is observed at a particular net (wire activation) is recorded and accordingly the counter for that net is constructed. Depending on the area overhead an attacker can expend, a net is chosen as the trigger and a counter equal to $\lceil log_2 n \rceil$ bit is constructed; n is the number of times the chosen net is activated.

Example. Figure 10 shows the original ISCAS-85 `c17` benchmark with five inputs and two outputs. Using an ATPG tool (e.g., *Synopsys Tetramax*), we compute test patterns which maximizes the fault coverage.[4] Next, we compute the bit-value for all internal nets for the obtained test patterns. Table 10(b) shows the test patterns and corresponding bit-values for each net in the `c17` benchmark. We observe that net `n2` receives logic "1" five times, i.e., it is activated the most (five times) as compared to the other nets, for the pre-determined set of test patterns. Next, we choose net `n2` as our trigger net and thus construct a 3-bit counter using DW of length ∼95 nm, which will activate only when `n2` observes logic "1" eight times.

Complex Trigger Design. The simple DW counters presented so far in this paper are representative examples designed to minimize the visibility and detectability of the trigger in terms of area and power overheads. However, these

3. Note that the baseline technology from the design house comprises a hybrid MESO-CMOS architecture, and the attacker inserts the DW trigger in this baseline circuit.

4. Fault coverage is defined as the percentage of faults detected by the ATPG tool over total faults in the design.



(a) (b)

Fig. 8. (a) Domain wall strip with MTJ structure at one end, to implement the trigger. The motion of the domain wall from left to right, on the application of input at node T1, represents the counting. Once the domain reaches the right end, the MTJ is in the anti-parallel state, which is marked by a jump in the junction resistance. This represents the firing of the trigger signal. (b) The DW moves from one end to the other, each time the input node of the trigger is activated. This is analogous to a digital counter counting steps, or in this case, the number of inputs applied. For a 10-bit counter, the motion of the DW and the corresponding count is as shown above.

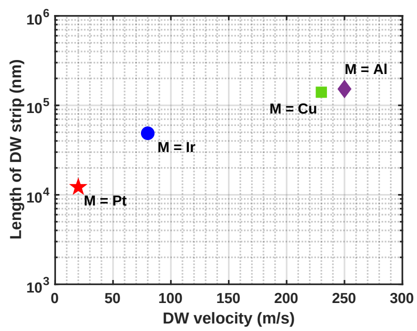
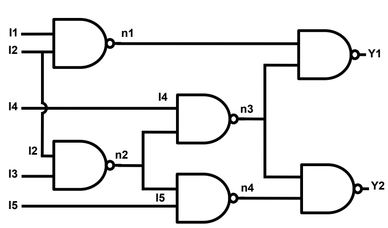


Fig. 9. Length of DW strip required to implement a 10-bit counter using various trilayers Pt/Co/M, where M $\equiv$ Pt, Ir, Cu, or Al. The ATPG frequency considered is 1.68 GHz [30].



Test Pattern	n1	n2	n3	n4
11011	0	1	0	0
00110	1	1	0	1
10000	1	1	1	1
11000	0	1	1	1
01111	1	0	1	1
00101	1	1	1	0
count	4	5	4	4

(a) (b)

Fig. 10. (a) ISCAS-85 `c17` benchmark. (b) Net values tabulated for each of the pre-determined test patterns.

are by no means the only trigger configurations possible with spin-based structures. Complicated trigger constructions, capable of matching control sequences, can provide the attacker extensive control over the activation, though at the cost of increased overheads and risk of detection. Note that such pattern matching arrangements require additional components like memory and comparators. The hybrid MESO-CMOS circuit style in conjunction with control sequence-based alternative triggers opens up possibilities for novel and unique triggering mechanisms for the attacker. For instance, a single DW trigger structure can activate two very different Trojan actions with two control sequences, all in the same payload. Here, one control sequence may program the MESO payload with specific functionality. The other control sequence might program the same payload as something else by leveraging the polymorphic nature of the MESO payload. However, the design and analysis of such complex trigger-payload configurations are beyond the scope of this paper.

5.2 Design of Trojan Payload

MESO gate-based payload. A MESO gate is configured into logic gates based on the polarity of voltages on terminals V_+ and V_- and the value at the control signal (X). Figure 6 shows the different logic configurations of the MESO gate. We utilize the existing MESO gates in the hybrid MESO-CMOS design as payload to lower the area overhead due to Trojan insertion. The payload delivers its targeted action by reconfiguring gates after trigger activation, which can either result in output corruption or can be leveraged for advanced key stealing attacks as demonstrated in further sections.

Example. Consider Fig. 11(a), where `c17` is constructed using hybrid MESO-CMOS gates; 50% of the total gates are changed to MESO (shown in green). An attacker can then choose all or any of these MESO gates as payload gates. The DW-based counter's output is then fed to the control signal X of the MESO payload gates to switch its logic functionality on Trojan activation. Using the transformations shown in Fig. 6, MESO NAND gates from Fig. 11(b) on trigger activation are converted into NOR gates as shown in Fig. 11(c), thereby resulting in output corruption.

For a generic circuit, any of the original MESO gates can be converted to payload gates (see Fig. 7). When the MUX select signal goes from low (0) to high (1) upon trigger activation, the NAND (AND) gates get transformed into NOR (OR) gates. For the reverse transformation, a second MUX with inverted trigger signal can be used.

6 Experimental Setup and Results

In this section, we verify the efficacy of our *PolyWorm* HT framework on the largest benchmarks from the ITC-99 test suite [36], for 20 individual trials. We provide the benchmark statistics in Table 1. We utilize existing metrics to quantify the effect of our Trojan on these benchmarks. Next, we define these metrics used in our analysis.

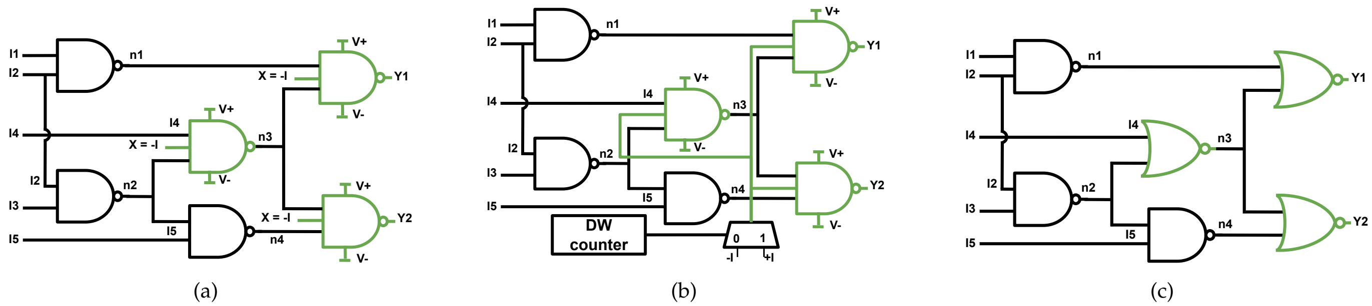


Fig. 11. (a) ISCAS-85 `c17` hybrid MESO-CMOS benchmark. The green gates are MESO-based and black ones are from CMOS domain. Any number of the three MESO gates could be leveraged for the payload. (b) ISCAS-85 `c17` example with DW counter. The DW counter drives a MUX, which decides the control signal *X* ("+I" or "-I") feeding into the MESO gates. (c) ISCAS-85 `c17` benchmark post-trigger activation. In this example, all the initial MESO NAND gates have been converted to NOR gates by changing *X* from "-I" to "+I".

TABLE 1
Benchmark statistics and number of test patterns achieving high fault coverage using stuck-at-fault model

	# PIs	# POs	# Gates	# TPs	FC (%)
b14_C	277	245	2,616	312	99.99
b15_C	487	449	4,197	420	99.94
b20_C	524	512	6,103	487	99.99
b21_C	524	512	5,915	450	99.98
b22_C	769	757	9,140	437	99.98
b17_C	1,454	1,445	13,526	530	99.89
b18_C	3,357	3,342	36,131	665	99.95
b19_C	6,666	6,669	70,175	979	99.86

PIs: Primary Inputs, POs: Primary Outputs, TPs: Test Patterns, FC: Fault Coverage

6.1 Definitions

- **Corruption** is a metric which structurally tests the mismatch between the corrupted netlist and the correct netlist and informs of the maximum corruption that can be achieved.
- **Hamming distance (HD)** is a metric that averages the bit-wise mismatch between the observed output response and the correct output response. It heavily depends on the input patterns applied. HD can be considered a sub-set of corruption.
- **Output error rate (OER)** is a metric that assesses the incorrect functionality over a large number of input patterns; even one-bit mismatch for each of these applied input patterns can result in 100% OER.

6.2 Setup

We model the MESO logic gates using CMOS gates as shown in Fig. 12, since industrial tools like Synopsys™ and Cadence™ do not yet support system-level simulations of the MESO device. This approximation in modeling is not prohibitive as we test only the functional equivalence and not the performance. To construct an AND/OR MESO gate, we utilize the AND, OR CMOS gates and 2:1 Multiplexer (MUX). The select line of the MUX corresponds to the control signal (*X*) of the MESO device. To construct hybrid MESO-CMOS designs, we randomly choose 10% of the total CMOS gates and convert them into MESO gates (using the

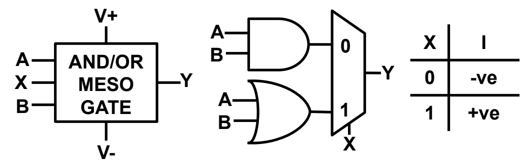


Fig. 12. MESO AND/ OR logic gate modeled with a CMOS circuit using CMOS AND, OR, and 2:1 Multiplexer. Select line chooses between the CMOS gates based on the control signal (*X*).

above-mentioned modeling). We use *Synopsys Tetramax* to generate test patterns for the modeled benchmarks.[5]

We conduct our Hamming distance and output corruption analysis on the considered benchmarks using *Synopsys VCS* and *Cadence Conformal LEC*. Area and power of the original and Trojan-infected designs is computed using *Synopsys Design Compiler* for NCSU FreePDK15 FinFET library. All these EDA tools are executed on a 128-core Intel Xeon processor running at 2.2 GHz with 794 GB RAM.

6.3 Domain wall-based vs. CMOS-based trigger

We motivate the selection of DW trigger as opposed to a CMOS trigger by providing a comparison between the two using a 10-bit counter. Low-energy DW devices have been shown to operate at low voltages ~100 mV, inducing write currents in the range ~100 μA, and resulting in a write power to the tune of ~10 μW [37], [38]. The area for a 10-bit DW counter is computed as 12,190×50 nm^2 for a 50 nm wide strip (see Fig. 9). To construct a 10-bit counter using CMOS gates, 10 flip-flops are utilized which incur an area overhead of ~21.04 um^2. Table 2 highlights the advantage of using a DW-based trigger over its CMOS counterpart, with respect to evading area-based Trojan detection approaches. Further, the power consumption of the DW-based trigger is also significantly less than that of the CMOS-based trigger as can be seen from Table 2. Both, the CMOS and DW-based counters do not affect the critical path of the circuit, thus bypassing performance checks carried out to detect Trojans. Along with the DW counter, the Trojan trigger includes two additional MESO-based MUXes to guide the programming of the Trojan MESO gates. For larger benchmarks,

5. Test patterns are generated considering only the logic inputs of the MESO gate (A, B), and not the control signal (*X*) or voltage terminal inputs.

TABLE 2
Area and power comparison between CMOS-based and Domain Wall (DW)-based 10-bit counter

	CMOS counter	DW counter
Area	21.04 um^2	$\sim$0.6 μm^2
Power	0.22 mW	$\sim$10 μW [37], [38]

the overhead of these two MUXes will be minimal, thereby hindering detection, since these MUXes do not lie in the critical path.

6.4 Trojan evaluation on ITC-99 benchmarks

We conduct security analysis of *PolyWorm*-infected benchmarks to assess the resiliency of our scheme; whether it can circumvent detection by structural testing and power-based testing at trusted test facilities.

Evasion from test-pattern-based detection: An attacker at the foundry can compute the test patterns for the Trojan-free design provided by the design house. He/she can then apply these test patterns to the Trojan-infected design to confirm the non-triggering of the embedded Trojan. The size of the counter chosen as a trigger must be larger than the activation count of the candidate trigger net. We compute the activation count for the generated test patterns. We observe that with such conditions on the size of the counter, involuntary Trojan activation was successfully evaded in 100% of the cases, thereby circumventing test-pattern-based Trojan detection. Further, we select the size of the counter and the trigger net based on the test patterns used to structurally test the chip, irrespective of the test coverage. Even if any design has low test coverage, our approach is successful since we only need to bypass the trusted test facility. We select the net which is activated the least number of times and choose a counter size greater than the most activated net (rounded up to closest power of 2) to ensure maximum tolerance limit between the number of times a net is activated during testing and the number of steps counted before the trigger fires. We observe that the maximum activation count is 978 for `b19_C`, hence choosing a 10-bit counter (see Table 2 for area and power overheads) for even the most activated net in the circuit provides a tolerance limit of $\sim$50 random test patterns. Note that, to increase the tolerance limit ($\sim$1070), the size of the counter can be increased to 11-bit.

Alternatively, the attacker can choose the size of the counter to account for all the test patterns while still selecting the least activated net. Table 1 outlines the number of test patterns required to achieve a fault coverage of more than 99.85%. We observe that the maximum number of test patterns required to test any of the ITC-99 benchmarks is 979 (b19_C), and hence, we choose a counter size of 10-bit to ensure low detectability with minimal overheads. Additionally, we also showcase the effect of random testing on our Trojan activation, wherein a designer chooses 100 random test patterns.[6] We observe that even with random testing, the Trojan was not activated. Note, the untrusted foundry can also simulate such a random testing and accordingly increase the counter size to evade detection at test facility.

6. The designer can increase the number of random patterns, which increases the test cost.

TABLE 3
Output corruption (%) for over 100,000 input patterns, where 20, 40, 60, 80, and 100% of MESO gates are used as Trojan payload

	20 %	40 %	60 %	80 %	100 %
b14_C	77.36	79.57	91.45	92.94	93.88
b15_C	85.02	85.93	87.74	91.25	91.76
b20_C	86.38	91.86	93.51	94.60	94.73
b21_C	83.52	87.37	93.15	95.11	95.31
b22_C	88.98	93.73	96.14	96.57	96.57
b17_C	77.07	79.33	79.81	81.49	83.67
b18_C	72.91	81.63	84.61	84.83	87.19
b19_C	79.49	85.55	90.83	91.82	94.11
Average	**81.34**	**85.62**	**89.65**	**91.08**	**92.15**

TABLE 4
Hamming distance (HD) over 100,000 input patterns, where 20, 40, 60, 80, and 100% of MESO gates are used as Trojan payload

	20 %	40 %	60 %	80 %	100 %
b14_C	4.09	6.97	9.95	12.55	14.32
b15_C	5.00	9.00	12.16	15.88	19.00
b20_C	5.47	11.41	12.75	18.71	21.49
b21_C	5.05	8.07	12.68	14.74	16.99
b22_C	4.68	7.96	10.51	13.50	15.66
b17_C	4.72	8.99	14.17	16.70	19.90
b18_C	3.8	7.55	10.75	13.03	16.77
b19_C	4.77	9.69	13.9	16.75	19.75
Average	**6.26**	**11.61**	**16.15**	**20.31**	**23.98**

Evasion from power-based detection: Since the power consumption difference between the original (Trojan-free) circuit and the Trojan-infected circuit is negligible ($\sim$10 μW overhead), an analysis of the total power consumption of the chip would not be able to conclusively pinpoint the presence of a Trojan. The variations in the MESO and CMOS fabrication process, which can lead to increased leakage power [39], would essentially mask these minute deviations in power, and it would be extremely difficult for the trusted entities to discriminate such process and temperature variations from the Trojan signature.[7]

Effect of Trojan: For ITC-99 benchmarks, we provide corruption numbers in Table 3 to determine the upper limit on the HD. The increasing trend of converting more MESO gates into Trojan payloads is coherent with the increase in the output corruption. HD is computed by applying 100,000 input patterns and averaging them. Table 4 provides the average HD numbers for the largest ITC-99 benchmarks. Further, the OER, post Trojan triggering, is 100% for all the benchmarks under consideration.

7 PolyWorm: Applications

In this section, we examine two particular case studies related to the primary objectives of *PolyWorm*, viz. denial of service and leaking of secret keys. These are achieved through Trojan induced corruption on a *Gaussian blur filter* image processing application and Trojan-aided key leakage in a pipelined *AES* design. The setup for these IPs is adopted from OpenCores [41], [42].

7. Note that, although the resolution of state-of-the-art multimodal Trojan detection apparatus is less than 0.05 μW/m^2 [40], the variability in the MESO-CMOS process is expected to be comparable to the DW-based counter power, and hence, render such testing ineffectual.

TABLE 5
Error introduced in a 32-bit adder and 32-bit multiplier due to *PolyWorm* activation. Z_{corr} and Z_{Trojan} refer to the result of the operation from the correct and Trojan-infected circuits, respectively

	A	B	Z_{corr}	Z_{Trojan}	Deviation
Adder	1.54×10^{7}	2.75×10^{8}	2.90×10^{8}	2.91×10^{8}	1.06×10^{6}
Multiplier	1.54×10^{7}	2.75×10^{8}	4.22×10^{15}	1.56×10^{19}	1.56×10^{19}

7.1 Case study I: Denial-of-Service

The Gaussian blur filter considered here, performs the Gaussian smoothing operation on the input image. It uses the Gaussian function shown in Eq. 1 to reduce white noise in the image.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}, \tag{1}$$

where σ is the standard deviation and x is distance from the origin in one dimension.

We obtain a 32-bit floating point multiplier from OpenCores [41] and construct the Gaussian blur filter using this multiplier. We only corrupt the multiplier used for normalizing the pixels of the original image, i.e., we embed *PolyWorm* in this multiplier and observe the corruption due to Trojan activation on the final filtered image. Note that *PolyWorm* can be inserted in any component of the filter, eg. multiplier or adder. We choose multiplier in this case study since it exacerbates the corruption of the output image (See Table 5). The corruption is re-producible, i.e., for the same set of input patterns, we get the same error from *PolyWorm* activation. We follow the steps for identifying the trigger net and accordingly construct the counter using DW. All the MESO gates in the multiplier are converted into *PolyWorm* payload gates. Thus, when the trigger is activated, all these MESO gates will change their original state, i.e., NAND $\rightarrow$ NOR, AND $\rightarrow$ OR, NOR $\rightarrow$ NAND, and OR $\rightarrow$ AND.

Results: Figure 13 presents a comparison between the original image with white noise, the correct Gaussian blurred image and the image obtained from the *PolyWorm*-infected design. The last image is produced using a *PolyWorm*-infected 32-bit floating point multiplier. Here, each pixel computation is passed through the corrupted multiplier. Hence, we observe a more distorted image, rather than a shift in the degree of Gaussian blurring through a variation in σ as per Eq. 1. We observe positive deviation for some pixels and negative deviation in others, with respect to the correct multiplier output. Due to reproducibility of error introduced by the Trojan-infected multiplier, we obtain some traces of the correct operation as opposed to a completely distorted image; we observe a non-uniform contrast image.

7.2 Case study II: Leaking the Secret Key

We demonstrate key leakage using *PolyWorm* on a pipelined version of AES taken from OpenCores [42]. There are 10 rounds in the AES algorithm as shown in Fig. 14. The first nine rounds contain a SubBytes operation, a ShiftRows operation, a MixColumns operation, and an AddRoundKey operation. The last round omits the MixColumns operation. The key for each round is generated from the main AES key using a Key Scheduler. Thus, in each round a new key

(a) (b) (c)

Fig. 13. We observe the effect of Trojan-infected multiplier on an image processing application (Gaussian blur filter). (a) Original image with white noise, (b) Correct Gaussian blur filtered image. (c) Trojan-infected image.

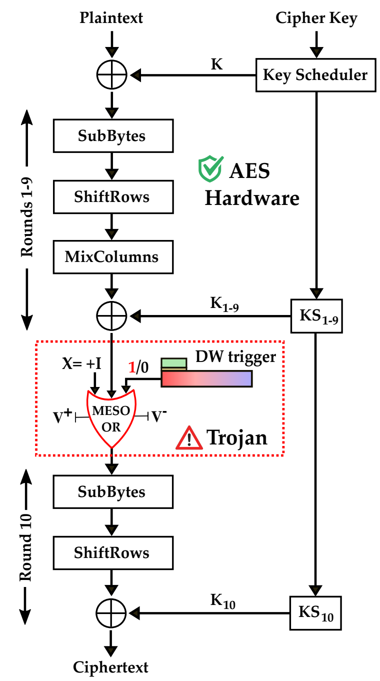


Fig. 14. Pipelined AES architecture with MESO-based payload and DW trigger inserted before the 10th round [43]. The Trojan here passes a chosen plaintext to the final round, upon activation.

is used. All these operations within the rounds cause an avalanche effect, i.e., if one bit in the plaintext or the key is changed, the ciphertext completely changes.

Implementation and results: The pipelined AES implementation from OpenCores is synthesized with NCSU FreePDK15 FinFET library. Since the attacker who resides in the foundry has access to the gate-level netlist as opposed to the behavioral RTL, she/he needs to first analyze this netlist to extract information about the AES rounds. The first nine rounds are pipelined, whereas the last round is excluded from the pipeline. Thus, the attacker can identify the beginning of the last round to insert the Trojan payload [43], as shown in Fig. 14. This payload is implemented using a MESO OR gate, which acts as a buffer as long as the DW-based trigger is dormant.[8] The area and power footprint of the trigger and payload are 0.028% and 0.03% respectively as shown in Table 6, thus evading detection through area

8. We note here that although the full polymorphic potential of MESO logic is not utilized here, it is still advantageous to use here owing to its negligible power and area footprints.

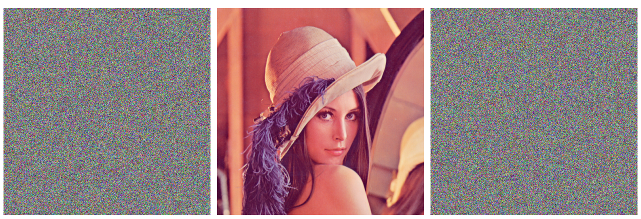
Algorithm 1: Key leakage algorithm using Trojan for pipelined AES implementation.

Input: Cipher text (CT)
Output: Secret Key (SK)

K_n ▷ n^{th} round key
$\{Wn_3, Wn_2, Wn_1, Wn_0\}$ ▷ Key k_n distributed into four 32-bit words
SB ▷ SubBytes operation
SR ▷ Shift register operation
ARK ▷ Add round key operation
Rot ▷ Rotation in the key expansion algorithm
$RCON$ ▷ Round constant in key expansion
TI ▷ Trojan input = all ones

K_{10} = $\{Wn_3, Wn_2, Wn_1, Wn_0\}$ = SR(SB(TI)) $\oplus$ CT;
for $i \leftarrow$ *9 to 0* **do**
  $Wi_3 = W(i+1)_3 \oplus W(i+1)_2$;
  $Wi_2 = W(i+1)_2 \oplus W(i+1)_1$;
  $Wi_1 = W(i+1)_1 \oplus W(i+1)_0$;
  $Wi_0 = W(i+1)_0 \oplus SB(Rot(Wi_3)) \oplus RCON$;

SK = $\{W0_3, W0_2, W0_1, W0_0\}$;



(a) (b) (c)

Fig. 15. We observe the effect of Trojan-infected AES to recover the secret key. (a) Encrypted image, (b) Decrypted image with recovered key. (c) Decrypted image with random key.

and power analysis. When the trigger is activated, the MESO OR gate will always pass a "1" to the final round. Post-activation, we use Alg. 1 to successfully leak the AES encryption key by means of a chosen plaintext attack (all 1s). Further, the Trojan infected AES design successfully evades detection for the 200 test patterns generated for the Trojan-free AES design. In Fig. 15, we illustrate this key retrieval process using an image. First, the image is encrypted with 128-bit AES, resulting in Fig. 15a. Then, we decipher the keys using the above-mentioned technique and decrypt the image to obtain Fig. 15b. Finally, Fig. 15c shows the result of decrypting the image with a random incorrect key.

8 Possible Countermeasures

In this section, we provide a brief discussion on possible countermeasures, which a designer can employ to prevent insertion of the proposed polymorphic Trojan.

8.1 Security-aware MESO Placement

Proper placement of the MESO payload gates is crucial for achieving high output corruption upon activation of the trigger. The design house may target this requirement and develop a security-aware MESO placement protocol to restrict usage of existing MESO gates as Trojan payload for achieving high output corruption on Trojan activation. However, this strategy would work only if the attacker solely utilizes the existing MESO gates in the original design as payload. Note that the attacker in the untrusted foundry also has the option of inserting additional MESO payload gates to circumvent any security-aware MESO placement. Nevertheless, the number of these new MESO gates that can be inserted is limited, depending on the area and power-based Trojan detection schemes utilized.

TABLE 6
Area and power overheads of Trojan payload inserted in AES pipelined implementation, computed at iso-performance

Circuit	Area (cells)	Area (μm^2)	Power (mW)
AES	123,729	41,916.68	103.21
AES_Trojan	123,858	41,928.65	103.24
Overhead	**129**	**0.028%**	**0.03%**

The extra 129 cells in the Trojan-infected AES design correspond to 128 MESO OR gates and 1 DW cell. The area and power of one MESO OR gate is 0.014μm^2 and 0.2μW, respectively [6], [44]. The metrics for the DW counter are shown in Table 2.

Note here that the main advantage of hybrid spin-CMOS architectures lies in the flexibility that emerging spintronic gates afford in synthesizing logic [44], [45]. For instance, MESO technology can directly implement majority logic in a single device and thus can greatly simplify the constrained synthesis of any target logic. But if the security-aware placement protocol of the designer places MESO gates only at those logic cones that do not propagate to or affect the primary outputs (to reduce output corruption on Trojan activation), then these advantages are mitigated and the purpose of the hybrid architecture is defeated.

8.2 Dynamic MESO Logic

Recall that in this paper, we focus specifically on hybrid MESO-CMOS architectures with static MESO gates, which are meant to retain their functionality post-fabrication. That is, the original hybrid circuit is designed to encompass only static MESO gates, but the attacker exploits the innate polymorphism of MESO logic to realize reconfigurable payload gates. This is in contrast to the dynamic style of hybrid MESO-CMOS logic [44], in which the MESO devices are intended to be used as reconfigurable gates by the design house itself. The difference between static and dynamic MESO logic styles, from the perspective of the designer, arises due to the need for a dedicated control and signal routing block to achieve proper reconfiguration of all the MESO gates in the dynamic architecture [44]. This additional circuit cost essentially necessitates static MESO logic design, forcing designers to use the dynamic style only when required. In this work, the attacker in the foundry inserts these additional peripherals to convert the existing static MESO gates to dynamic payload gates.

For dynamic MESO-CMOS architectures, the designer can leverage the reconfigurability of the MESO device to prevent an attacker in the foundry from obtaining the same test patterns as the trusted test facility. The designer then sends this dynamically configured MESO-CMOS design to the untrusted foundry and generates test patterns for some dummy static configuration to share with the trusted test facility. An untrusted foundry without knowing this dummy static configuration cannot produce the same test patterns as the test facility. This deters the foundry from effectively choosing a detection-resistant net for trigger placement. As such, the insertion of polymorphic Trojans in dynamic MESO-CMOS architectures is reserved for future work, and we also invite the device and security community to follow-up on this discussion.

9 Conclusion

Emerging spin devices, which exhibit advantages like low power, high integration density, and non-volatile data retention, offer an alternative to the conventional semiconductor technology, which faces challenges with continual scaling. MESO device also showcases reconfigurability of logic gates post-fabrication. Although this finds many applications, including logic reduction and IC camouflaging, attackers can abuse this property to perform malicious activities.

In this work, we highlight how such a vulnerability (polymorphic nature of MESO gates) can be exploited by an untrusted foundry to insert a hardware Trojan, *PolyWorm*, in hybrid MESO-CMOS ICs. *PolyWorm* exploits the innate functional polymorphism in MESO devices to cloak itself as a harmless logic gate from the original design IP. By utilizing the existing logic gates to implement the Trojan payload, we lower the area overheads and make it inconspicuous. We also present an efficient construction of the trigger using a low-power and low-area domain wall device to evade detection through power-based testing. The trigger is placed at a stealthy location to avoid exposure during structural and delay-based testing analyses. We also demonstrate the efficacy and stealth of *PolyWorm* on eight largest ITC-99 benchmarks. Using case studies on denial of service in an image processing IP and leaking of the secret key in an AES cipher, we hope to motivate further research into the security of hybrid CMOS architectures of the near future.

References

[1] N. Z. Haron and S. Hamdioui, "Why is CMOS scaling coming to an END?" in *International Design and Test Workshop*. IEEE, 2008, pp. 98–103.

[2] T.-C. Chen, "Overcoming research challenges for CMOS scaling: Industry directions," in *International Conference on Solid-State and Integrated Circuit Technology Proceedings*. IEEE, 2006, pp. 4–7.

[3] "Intel's next-generation 7nm chips delayed until 2022," https://www.bbc.com/news/technology-53525710, accessed: 2020-08-18.

[4] P. Ye, T. Ernst, and M. V. Khare, "The last silicon transistor: Nanosheet devices could be the final evolutionary step for Moore's law," *IEEE Spectrum*, vol. 56, no. 8, pp. 30–35, 2019.

[5] M. Sadaka, I. Radu, and L. Di Cioccio, "3D integration: Advantages, enabling technologies & applications," in *International Conference on Integrated Circuit Design and Technology*. IEEE, 2010, pp. 106–109.

[6] S. Manipatruni, D. E. Nikonov, C.-C. Lin, T. A. Gosavi, H. Liu, B. Prasad, Y.-L. Huang, E. Bonturim, R. Ramesh, and I. A. Young, "Scalable energy-efficient magnetoelectric spin–orbit logic," *Nature*, vol. 565, no. 7737, pp. 35–42, 2019.

[7] H. Liu, S. Manipatruni, D. H. Morris, K. Vaidyanathan, D. E. Nikonov, T. Karnik, and I. A. Young, "Synchronous Circuit Design With Beyond-CMOS Magnetoelectric Spin–Orbit Devices Toward 100-mV Logic," *Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 5, no. 1, pp. 1–9, 2019.

[8] C.-C. Lin, T. Gosavi, D. Nikonov, Y.-L. Huang, B. Prasad, W. Choi, I. Groen, J.-Y. Chen, D. Mahendra, H. Liu *et al.*, "Experimental demonstration of integrated magneto-electric and spin-orbit building blocks implementing energy-efficient logic," in *International Electron Devices Meeting (IEDM)*. IEEE, 2019, pp. 37–3.

[9] G. I. Bourianoff, P. A. Gargini, and D. E. Nikonov, "Research directions in beyond CMOS computing," *Solid-State Electronics*, vol. 51, no. 11-12, pp. 1426–1431, 2007.

[10] "Intel's MESO transistor promises vast leap in AI processing power," https://venturebeat.com/2019/02/21/intels-meso-transistor-promises-vast-leap-in-ai-processing-power/, accessed: 2020-08-18.

[11] "Intel's fundamentally new MESO architecture could arrive in a few years," https://www.extremetech.com/computing/286163-intels-fundamentally-new-meso-architecture-could-arrive-in-a-few-years, accessed: 2020-08-18.

[12] "Intel looks beyond CMOS to the future of logic devices," https://newsroom.intel.com/news/intel-looks-cmos-future-logic-devices/#gs.dqw8ah, accessed: 2020-08-18.

[13] "Bringing energy-efficient MESO technology a step closer to reality," https://devicematerialscommunity.nature.com/posts/65094-cic-nanogune-and-intel-bring-the-meso-technology-a-step-closer-to-reality, accessed: 2020-08-18.

[14] H. Dery, P. Dalal, L. Sham *et al.*, "Spin-based logic in semiconductors for reconfigurable large-scale circuits," *Nature*, vol. 447, no. 7144, pp. 573–576, 2007.

[15] R. Ishihara, Y. Ando, S. Lee, R. Ohshima, M. Goto, S. Miwa, Y. Suzuki, H. Koike, and M. Shiraishi, "Gate-Tunable Spin xor Operation in a Silicon-Based Device at Room Temperature," *Physical Review Applied*, vol. 13, no. 4, p. 044010, 2020.

[16] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *International Workshop on Hardware-Oriented Security and Trust*. IEEE, 2008, pp. 15–19.

[17] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE design & test of computers*, vol. 27, no. 1, pp. 10–25, 2010.

[18] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: Lessons learned after one decade of research," *Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 1, pp. 1–23, 2016.

[19] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *International workshop on hardware-oriented security and trust*. IEEE, 2008, pp. 51–57.

[20] S. Bhasin, J.-L. Danger, S. Guilley, X. T. Ngo, and L. Sauvage, "Hardware Trojan horses in cryptographic IP cores," in *Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE, 2013, pp. 15–29.

[21] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, "Hardware Trojan insertion by direct modification of FPGA configuration bitstream," *Design & Test*, vol. 30, no. 2, pp. 45–54, 2013.

[22] M. Ender, S. Ghandali, A. Moradi, and C. Paar, "The first thorough side-channel hardware Trojan," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 755–780.

[23] J. Clements and Y. Lao, "Hardware Trojan attacks on neural networks," *arXiv preprint arXiv:1806.05768*, 2018.

[24] R. S. Chakraborty, S. Paul, and S. Bhunia, "On-demand transparency for improving hardware Trojan detectability," in *International Workshop on Hardware-Oriented Security and Trust*. IEEE, 2008, pp. 48–50.

[25] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. G. Wolff, C. A. Papachristou, K. Roy, and S. Bhunia, "Hardware Trojan detection by multiple-parameter side-channel analysis," *Transactions on computers*, vol. 62, no. 11, pp. 2183–2195, 2012.

[26] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: A statistical approach for hardware trojan detection," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2009, pp. 396–410.

[27] M. Isasa, M. C. Martínez-Velarte, E. Villamor, C. Magén, L. Morellón, J. M. De Teresa, M. R. Ibarra, G. Vignale, E. V. Chulkov, E. E. Krasovskii *et al.*, "Origin of inverse Rashba-Edelstein effect detected at the Cu/Bi interface using lateral spin valves," *Physical Review B*, vol. 93, no. 1, p. 014420, 2016.

[28] E. Saitoh, M. Ueda, H. Miyajima, and G. Tatara, "Conversion of spin current into charge current at room temperature: Inverse spin-Hall effect," *Applied physics letters*, vol. 88, no. 18, p. 182509, 2006.

[29] P. Borisov, A. Hochstrat, X. Chen, W. Kleemann, and C. Binek, "Magnetoelectric switching of exchange bias," *Physical Review Letters*, vol. 94, no. 11, p. 117203, 2005.

[30] "T2000," https://www.advantest.com/products/ic-test-systems/t2000, accessed: 2020-06-15.

[31] J. Grollier, D. Querlioz, and M. D. Stiles, "Spintronic nanodevices for bioinspired computing," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 2024–2039, 2016.

[32] S. S. Parkin, M. Hayashi, and L. Thomas, "Magnetic domain-wall racetrack memory," *Science*, vol. 320, no. 5873, pp. 190–194, 2008.

[33] M. Tsoi, R. Fontana, and S. Parkin, "Magnetic domain wall motion triggered by an electric current," *Applied physics letters*, vol. 83, no. 13, pp. 2617–2619, 2003.

[34] P. Sharma, D. Sando, Q. Zhang, X. Cheng, S. Prosandeev, R. Bulanadi, S. Prokhorenko, L. Bellaiche, L.-Q. Chen, V. Nagarajan *et al.*, "Conformational domain wall switch," *Advanced Functional Materials*, vol. 29, no. 18, p. 1807523, 2019.

[35] F. Ajejas, V. Křižáková, D. de Souza Chaves, J. Vogel, P. Perna, R. Guerrero, A. Gudin, J. Camarero, and S. Pizzini, "Tuning domain wall velocity with Dzyaloshinskii-Moriya interaction," *Applied Physics Letters*, vol. 111, no. 20, p. 202402, 2017.

[36] F. Corno, M. S. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *Design & Test of Computers*, vol. 17, no. 3, pp. 44–53, 2000.

[37] J. A. Currivan, Y. Jang, M. D. Mascaro, M. A. Baldo, and C. A. Ross, "Low energy magnetic domain wall logic in short, narrow, ferromagnetic wires," *Magnetics Letters*, vol. 3, pp. 3 000 104–3 000 104, 2012.

[38] S. Fukami, M. Yamanouchi, K.-J. Kim, T. Suzuki, N. Sakimura, D. Chiba, S. Ikeda, T. Sugibayashi, N. Kasai, T. Ono *et al.*, "20-nm magnetic domain wall motion memory with ultralow-power operation," in *International Electron Devices Meeting*. IEEE, 2013, pp. 3–5.

[39] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Design Automation Conference*, 2003, pp. 338–342.

[40] A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1792–1805, 2014.

[41] "Floating point adder and multiplier," https://opencores.org/projects/fpuvhdl, accessed: 2020-09-12.

[42] "Aes-128 encryption," https://opencores.org/projects/aes-128_pipelined_encryption, accessed: 2020-09-12.

[43] A. Jain and U. Guin, "A Novel Tampering Attack on AES Cores with Hardware Trojans," in *International Test Conference in Asia*. IEEE, 2020, pp. 1–6.

[44] N. Rangarajan, S. Patnaik, J. Knechtel, R. Karri, O. Sinanoglu, and S. Rakheja, "Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2020.

[45] N. Rangarajan, S. Patnaik, J. Knechtel, O. Sinanoglu, and S. Rakheja, "Spin-based reconfigurable logic for power-and area-efficient applications," *IEEE Design & Test*, vol. 36, no. 3, pp. 22–30, 2019.

Nimisha Limaye is a Ph.D. candidate at the Department of Electrical and Computer Engineering at New York University, USA and also a Global Ph.D. Fellow with New York University Abu Dhabi, UAE. Her research interests include hardware security and in particular logic locking, scan locking, and hardware Trojan. She received B.E. in Electronics and Telecommunications Engineering from University of Mumbai, India in 2015 and M.S. in Computer Engineering from New York University, USA in 2017.

Nikhil Rangarajan is a Postdoctoral Associate at the Division of Engineering, New York University Abu Dhabi, United Arab Emirates. He has Ph.D. and M.S. degrees in Electrical Engineering from New York University, NY, USA. His research interests include spintronics, nanoelectronics, device physics and hardware security.

Satwik Patnaik received the B.E. degree in electronics and telecommunications from the University of Pune, India, the M.Tech. degree in computer science and engineering with a specialization in VLSI design from the Indian Institute of Information Technology and Management, Gwalior, India, and the Ph.D. degree in Electrical engineering from Tandon School of Engineering, New York University, Brooklyn, NY, USA in September 2020.

He is currently a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. His current research interests include hardware security, trust and reliability issues for CMOS and emerging devices with particular focus on low-power VLSI Design. Dr. Patnaik received the Bronze Medal in the Graduate Category at the ACM/SIGDA Student Research Competition held at ICCAD 2018, and the Best Paper Award at the Applied Research Competition held in conjunction with Cyber Security Awareness Week, in 2017.

Ozgur Sinanoglu is a professor of electrical and computer engineering at New York University Abu Dhabi. He obtained his Ph.D. in Computer Science and Engineering from University of California San Diego. He has industry experience at TI, IBM and Qualcomm, and has been with NYU Abu Dhabi since 2010. During his Ph.D. he won the IBM Ph.D. fellowship award twice. He is also the recipient of the best paper awards at IEEE VLSI Test Symposium 2011 and ACM Conference on Computer and Communication Security 2013. Prof. Sinanoglu's research interests include design-for-test, design-for-security and design-for-trust for VLSI circuits, where he has more than 200 conference and journal papers, and 20 issued and pending US Patents. Prof. Sinanoglu is the director of the Center for CyberSecurity at NYU Abu Dhabi. His recent research in hardware security and trust is being funded by US National Science Foundation, US Department of Defense, Semiconductor Research Corporation, Intel Corp and Mubadala Technology.

Kanad Basu received his Ph.D. from the department of Computer and Information Science and Engineering, University of Florida. His thesis was focused on improving signal observability for post-silicon validation. Post-Ph.D., Kanad worked in various semiconductor companies like IBM and Synopsys. During his Ph.D. days, Kanad interned at Intel. Currently, Kanad is an Assistant Professor at the Electrical and Computer Engineering Department of the University of Texas at Dallas. Prior to this, Kanad was an Assistant Research Professor at the Electrical and Computer Engineering Department of NYU. He has authored 2 US patents, 2 book chapters and several peer reviewed journal and conference articles. Kanad was awarded the "Best Paper Award" at the International Conference on VLSI Design 2011. Kanad's current research interests are hardware and systems security.