Satisfiability Attack-resilient Camouflaged Multiple Multivariable Logic-in-Memory Exploiting 3D NAND Flash Array

SUBMISSION DATE / POSTED DATE

10-11-2022 / 11-11-2022

CITATION

Sahay, Shubham; Swaroop, Bhogi Satya; Saxena, Ayush (2022): Satisfiability Attack-resilient Camouflaged Multiple Multivariable Logic-in-Memory Exploiting 3D NAND Flash Array. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.21532455.v1

DOI

10.36227/techrxiv.21532455.v1

Satisfiability Attack-resilient Camouflaged Multiple Multivariable Logic-in-Memory Exploiting 3D NAND Flash Array

Bhogi Satya Swaroop, *Member, IEEE,* Ayush Saxena, and Shubham Sahay, *Member, IEEE*

***Abstract*— Logic-in-memory implementations have attracted significant attention recently for energy efficient in-situ processing of big data in this era of IoT. However, the emerging memory technologies such as RRAMs, PCMs, STT-MRAMs, etc. are still immature and exhibit significant spatial and temporal variations limiting the yield and the size of crossbar arrays available for implementing logic functions. Considering the technological maturity, ultra-high density and ultra-low cost of 3D NAND flash memory, in this work, we have proposed a novel methodology to exploit 3D NAND flash memory for realizing any logic function in sum-of-product form (SOP) with ≤177 literals/inputs and $\leq 2^{14}$ minterms parallelly. Moreover, all the logic functions realized using the proposed technique appear same at the layout level rendering the logic-in-memory implementation utilizing the 3D NAND flash memory an innate camouflaging property and an inherent immunity against security vulnerabilities in the semiconductor supply chain. We have also evaluated the resiliency of the proposed technique against reverse engineering attacks such as SAT attacks, ATPG attacks and brute force attacks on ISCAS'85 and ISCAS'89 benchmark circuits. Our results indicate that the proposed logic-in-memory implementation facilitates complete obfuscation of the logic function without introducing any area overhead and exhibits a strong resiliency against reverse engineering.**

***Index Terms*—— 3D NAND Flash, logic-in-memory, Reverse engineering, IC camouflaging, SAT attack.**

I. INTRODUCTION

The unprecedented growth in the interconnected cyber-physical systems in this era of Internet-of-things (IoT) and rapid advancements in emerging sectors such as social media, personalized health care, finance, online education, etc. have led to a surge in the asynchronous data. Analyzing and processing such Big-data with high energy-efficiency is a challenge for the conventional von-Neumann computing systems. The physical isolation of the memory and processing units operating at different speed degrades the performance of the von-Neumann computing primitives owing to the large data transfer. This von-Neumann bottleneck not only increases the energy dissipation but also reduces the computing speed.

Recently, compact and highly energy-efficient processing-in-memory primitives exploiting co-located storage and computational blocks were introduced to minimize the data transfer and circumvent this von-Neumann bottleneck. Since logic gates are used in almost every aspect of processing and computation, implementation of logic gates within the memory block may accelerate the computations significantly while drastically reducing the energy consumption and area. Therefore, logic-in-memory implementations exploiting volatile SRAM [1] and DRAM [2] cells were proposed.

While a NOR gate was realized with inputs stored in two 8T SRAM cells, the NAND gate implementation requires time-constrained read operation and skewed inverters [1]. Moreover, an IMPLY logic (IMP) gate was also realized using a voltage divider implementation exploiting 8T SRAM cell with additional voltage sources and skewed inverters [1]. Furthermore, an 8^{+}T SRAM cell (9 transistors) with appropriately sized sense amplifier-based NAND and NOR logic gates were also realized [1]. Although these universal gate implementations appear promising, the complex logic gates such as XOR are derived from these basic gates which results in an increased latency and energy dissipation and reduced throughput. Furthermore, utilization of 8T and 8^{+}T SRAM cells, skewed inverters and additional voltage sources lead to a large area overhead and complex routing.

Moreover, a bitwise OR and AND logic was implemented in [2] with inputs stored in two rows of 1T-1C DRAM array. However, the output of these logic operations overwrites the input data stored in DRAM cells necessitating redundant storage of input data leading to an increase in the area and energy dissipation. Furthermore, the input/output data cannot be retrieved after power-off since SRAM and DRAM are volatile memories.

Recently, several in-memory logic implementations based on compact emerging non-volatile memories (NVM) such as memristors [3]-[4], RRAM [5], PCM [6] etc. were proposed. Although the standby power dissipation (which dominates the energy landscape for CMOS designs) is eliminated in NVM based logic implementations, they can only support limited fan-ins (typically 2) per stage. Moreover, the emerging NVM technology is not mature and exhibits large spatial and temporal variations, complex fabrication process and limited yield despite the recent developments in the integration of emerging NVMs with the CMOS technology [3]-[6].

Recently, 3D NAND flash memory has emerged as the

The authors are with the Department of Electrical Engineering, Indian Institute of Technology Kanpur, 208016, India (e-mail: ssahay@iitk.ac.in).

This work was partly supported by by the C3iHub, IIT Kanpur and Startup Research Grant (SRG) from SERB (SERB/EE/2021358).

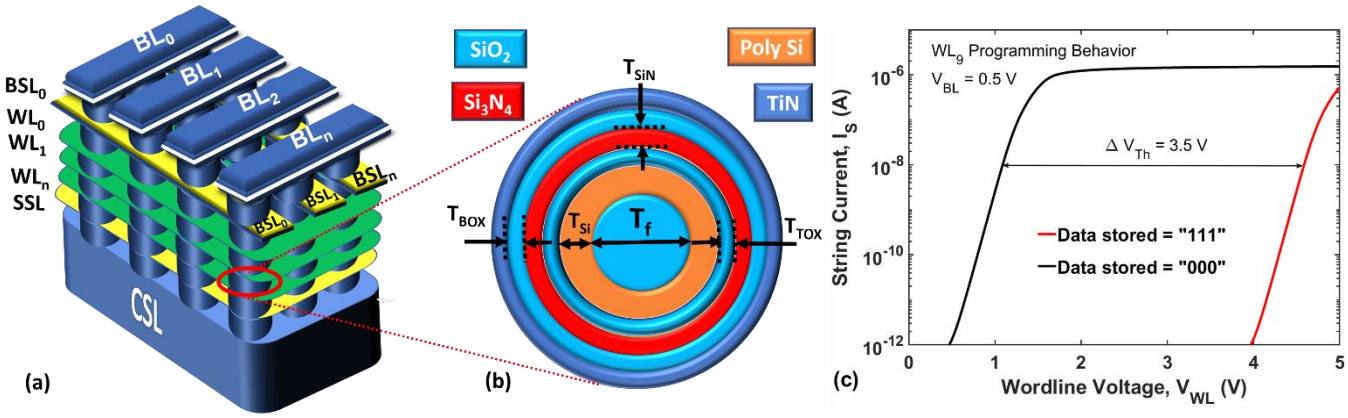

Fig. 1 (a) Bird's eye view of the gate-all-around (GAA) charge trap (CT) Macaroni body 3D NAND flash memory array, (b) cross-sectional view of the 3D NAND flash cell and (c) string current characteristics of the 3D NAND cell located at WL_9 for extreme programmed state (111) and the erased state (000).

mainstream technology for data storage ranging from small USB drives to solid state drives in data warehouses and gigantic cloud storage. Considering the high technological maturity, ultra-high density and ultra-low cost of the 3D NAND flash memory [7], it becomes imperative to analyze their potential for logic-in-memory implementations. Although 3D NAND flash memory array has been exploited for implementation of in-memory vector-by-matrix multiplication accelerators [8]-[12], to the best of our knowledge, their application for logic implementation has not been explored so far.

To this end, in this work, we have proposed a methodology to implement logic functions within the mature and ultra-dense commercial 3D NAND flash memory array. Any Boolean logic in sum-of-products (SOP) form can be implemented by encoding inputs as bit line (BL) voltages and threshold voltages of the 3D NAND flash cells in the strings and sensing the output accumulated on the input capacitance of the sense amplifier. Such an implementation enables realization of logic functions with ≤177 literals/inputs and $\leq 2^{14}$ minterms parallelly considering the state-of-art 3D NAND flash memory array with 176-word line (WL) layers and page size of 16 KB. This is highly advantageous since realizing such high fan-in logic using CMOS technology not only requires large number of complimentary MOSFETS and multi-stage design but also results in large latency, area overhead, high power dissipation and routing complexity. Moreover, the throughput of the proposed implementation is significantly high as compared to the logic gate implementations utilizing other NVMs [3]-[6]. Using an experimentally calibrated behavioral compact model for 3D NAND flash memory [13], with the aid of HSPICE simulations, we have implemented the basic logic gates (AND, OR, NAND, NOR, XOR and XNOR) and performed extensive analysis of their delay and energy dissipation. Furthermore, a novel system-level reconfigurable architecture for fast and energy-efficient implementation of logic functions using pre-programmed 3D NAND flash cells and a content addressable memory (CAM) has also been proposed.

Moreover, the integrated circuit (IC) manufacturing process involves a large number of steps including chip design, manufacturing, testing, packaging and supply of final product. The entire supply chain is dispersed across the globe to reduce cost and design time. Furthermore, due to the economic considerations while sustaining the foundries, many semiconductor giants have gone fabless increasing their dependence on the global supply chain. Recently, the semiconductor supply chain has been exposed to several security vulnerabilities such as reverse engineering, IC piracy, IP piracy, trojan insertion, counterfeiting, etc. due to the involvement of malicious parties in the supply chain. Reverse engineering (RE) [14]-[15] involves de-packaging the IC and delayering and imaging it layer by layer to extract the gate level netlist and is considered as the most serious threat to the semiconductor industry since the design technology and IP used in the IC can be identified and the functionality may be inferred. Once the design is identified, the malicious attacker may initiate counterfeiting or manufacturing the same product and supplying at a reduced cost.

Recently, several techniques such as IC camouflaging, logic locking, split manufacturing, etc. have been proposed to thwart the RE attacks. IC camouflaging [16] is a hardware obfuscation technique where the designers introduce dummy elements/contacts in different logic gates to realize the same physical layout. Although addition of dummy contacts yields similar layout for NAND, NOR, and XOR gates in [17] which may deceive the reverse engineer, it also leads to an increased area overhead.

The proposed logic-in-memory technique using ultra-dense 3D NAND flash memory array exhibits an innate camouflaging property whereby all the realized logic functions with ≤177 literals/inputs and $\leq 2^{14}$ minterms appear similar at the layout level. Hence, it provides an inherent strong resilience against the reverse engineering techniques such as SAT attacks, ATPG attacks and brute force attacks without additional area overhead. Moreover, compared to the prior IC camouflaging techniques which may camouflage only NAND/NOR/AND/OR gates, the proposed implementation provides efficient means to camouflage all the logic functions and is less vulnerable to the SAT attacks performed on ISCAS'85 [18] and ISCAS'89 [19] benchmark circuits.

The manuscript is organized as follows: the structure and operating principle of 3D NAND flash memory is described in section II. The logic-in-memory implementation methodology based on 3D NAND flash array is discussed in section III and the performance metrics are evaluated in section IV. The reconfigurable system-level architecture to further enhance the performance of logic-in-memory implementation using 3D NAND flash array is described in

TABLE I
PARAMETERS USED FOR 3D NAND FLASH MEMORY ARRAY

Parameters	Value
Core filler diameter (T_f)	35 nm [13]
Tunnel oxide thickness (T_{TOX})	4 nm [13]
Blocking oxide thickness (T_{BOX})	4.5 nm [13]
Nitride layer thickness (T_{SiN})	5 nm [13]
Active channel thickness (T_{si})	10 nm [13]
Spacer thickness (t_{sp})	50 nm [13]
Page Size	16 KB [21]

section V. The camouflaging property of the proposed 3D NAND flash-based logic-in-memory implementation is discussed in section VI and its resiliency against the SAT attack, ATPG attack and brute-force attack is analyzed in sections VII, VIII and IX, respectively and the conclusions are drawn in section X.

II. STRUCTURE AND OPERATING PRINCIPLE

The three-dimensional view of 3D NAND Flash memory array [20] is as shown in Fig. 1(a). The cylindrical pillars (strings) consist of Macaroni body, vertical channel charge-trap (CT) devices with oxide/nitride/oxide (O/N/O) in gate stack. The Flash cells are located at the intersection of cylindrical pillar and word line (WL) plates. Bit lines (BL) and Bit selector transistor lines (BSL) are used to access a particular string and WL is used to select a particular flash cell in that string. All the strings share the same common source line (CSL). The exact dimensions of 3D NAND flash cell shown in Fig. 1 are given in Table I.

The schematic view of flash cell array considering the behavioral compact modeling approach used in [13] is shown in Fig. 2. The BL and BSL are orthogonal to each other and are used for selecting a particular string within the array. WL is implemented as a metal plate to select a particular layer of the array. Each TLC flash cell considered in this work can exhibit eight different threshold voltages. The threshold voltage of each cell can be programmed to the desired state using incremental step pulse programming/erase (ISPP/E) technique [22].

To implement any Boolean logic, the logic function is first converted into its SOP form and the inputs are applied as bit line voltages and threshold voltages of the flash cells. A high threshold voltage corresponding to the extreme programmed state (111) is treated as logic '0' and a low threshold voltage corresponding to the erased state (000) is treated as logic '1'. A voltage of 0.5 V on BL represents logic '1', and the BL is grounded to implement logic '0'. One literal of a minterm is applied on the BL and the remaining literals of that minterm are encoded as threshold voltages of the flash cells in that string. Therefore, one string is used to encode one minterm of the Boolean logic expressed in the SOP form. To perform computation, a read voltage (V_{read} = 2.5 V) is applied to the WL of the flash cells encoding the inputs, and a pass voltage (V_{pass} = 8 V) is applied to the remaining WLs to reduce the series resistance of the flash cells along the string. The current flowing through the string depends on the applied bit line voltage and the threshold voltage state of the programmed flash cells and encodes the output as the voltage accumulated on the input capacitance of the sense amplifier (C_{CSL}). A high voltage (0.5 V) on the C_{CSL} is considered as output logic '1'

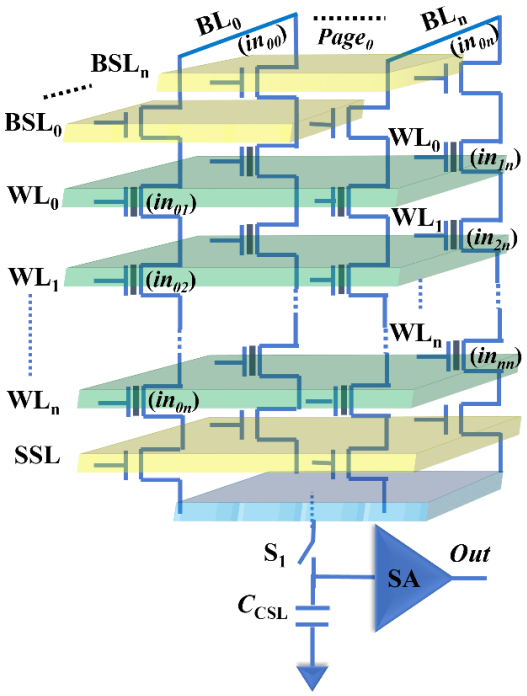

Fig. 2 Schematic representation of the 3D NAND flash memory array used for implementation of logic gates utilizing the behavioral compact modeling approach proposed in [13].

and a low voltage (0 V) on the C_{CSL} is considered as output logic '0'. The voltage on the C_{CSL} can then be fed as the input of the subsequent stages for cascading of logic gates.

III. LOGIC-IN-MEMORY IMPLEMENTATION

For proof-of-concept demonstration of logic-in-memory implementation utilizing 3D NAND Flash memory array, we have designed two-input universal gates like NAND and NOR and other useful logic functions like OR, AND, XOR and XNOR using the proposed approach. The compact form of these implementations (where BSL and SSL are omitted for simplicity) are shown in Fig. 3. From Fig. 3, we observe that logic gate implementations with only one minterm such as N-input AND, NOR, etc. require only one string of 3D NAND flash memory array. However, logic gate implementations of N-bit NAND, OR, XOR, XNOR, etc. which consist of N-minterms in their SOP expression need N strings of 3D NAND flash memory array. Considering the ultra-high density of the commercial 3D NAND flash memories with ≤ 176 WL layers and page size exceeding 16 KB, we can real1ize logic functions with ≤ 177 inputs (literals) and ~ 2^{14} minterms parallelly. Realizing logic functions with such high fan-ins (inputs) exploiting single-stage static CMOS design technique not only requires a large number of complementary MOSFETs (atleast N-pMOSFETs and N-nMOSFETs for N fan-ins) but also results in a significantly high delay, large area and high dynamic power dissipation. Furthermore, even input signal routing is a complex task while realizing such CMOS circuits with large fan-ins. Therefore, logic functions with large inputs are typically implemented using multi-stage CMOS circuits with large latency. However, the proposed approach facilitates a simple, energy-efficient and single-stage implementation of even complex logic functions with large fan-ins or minterms

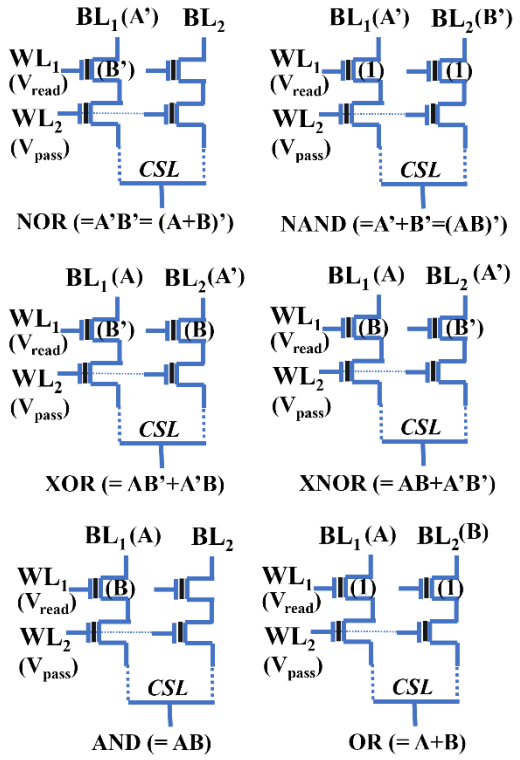

Fig. 3 Implementation of logic gates using 3D NAND Flash.

with a high throughput. Moreover, the proposed approach can also be extended to implement multiplexer (MUX) based logic designs.

IV. Results And Discussion

To analyze the efficacy of the proposed logic-in-memory implementation utilizing the 3D NAND flash memory array, we have performed rigorous analysis of the performance metrics such as energy and delay for different logic gates designed in section III using the experimentally calibrated behavioral compact model of 3D NAND flash [13] with the aid of HSPICE [23] simulations. The behavioral compact model in [13] utilizes the BSIM CMG 110.0.0 model to mimic the cell behavior and the model parameters have been tuned to reproduce the experimental characteristics of a 3D NAND flash string with 10 WLs [24-25]. Moreover, the model also effectively captures the parasitic capacitances pertaining to the string in the 3D NAND flash architecture [13]. The methodology utilized for estimating the performance of the proposed logic-in-memory implementation utilizing 3D NAND flash memory array is described in detail in the following sub-sections.

A. Energy calculation

For performing logic-in-memory operations utilizing 3D NAND flash, first, the inputs have to be encoded as threshold voltages of the flash cells in the string. To encode the inputs as threshold voltage, we need to apply a programming pulse which leads to the write energy dissipation.

For a threshold voltage (V_t) shift of 3.5 V (Fig. 1(c)), the programming voltage (V_{pgm}) required is 20 V [22]. The write energy (E_{write}) can then be calculated as:

$$E_{write} = \Delta q . V_{pgm} \quad (1)$$

where Δq is the change in the amount of charge stored in the nitride layer, and is given by:

$$\Delta q = C_q . A . V_{t-shift} \quad (2)$$

Table II
Energy Consumed By Different Logic Gates

In		Out	XOR			XNOR		
A	B	O	Write energy	Read energy	Total energy	O	Write energy	Read energy	Total energy
0	0	0	0.16 fJ	0	0.16 fJ	1	0.16 fJ	30 fJ	30.16 fJ
0	1	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	0	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	1	0	0.16 fJ	0	0.16 fJ	1	0.16 fJ	30 fJ	30.16 fJ
In		**Out**	**OR**			**AND**		
A	**B**	**O**	**Write energy**	**Read energy**	**Total energy**	**O**	**Write energy**	**Read energy**	**Total energy**
0	0	0	0.16 fJ	0	0.16 fJ	0	0.16 fJ	0	0.16 fJ
0	1	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	0	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	1	1	0.16 fJ	30 fJ	30.16fJ	1	0.16 fJ	30 fJ	30.16 fJ
In		**Out**	**NAND**			**NOR**		
A	**B**	**O**	**Write energy**	**Read energy**	**Total energy**	**O**	**Write energy**	**Read energy**	**Total energy**
0	0	1	0.16 fJ	30 fJ	30.16fJ	1	0.16 fJ	852 fJ	30.16 fJ
0	1	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	0	1	0.16 fJ	30 fJ	30.16fJ	0	0.16 fJ	0	0.16 fJ
1	1	0	0.16 fJ	0	0.16 fJ	0	0.16 fJ	0	0.16 fJ

where A is the curved surface area and C_q is the capacitance per unit area between the charge trap nitride layer and the active polysilicon layer. C_q can be derived as [22]:

$$C_q = \frac{\varepsilon_{ox}}{T_{TOX}+\frac{\varepsilon_{ox}}{2\varepsilon_{SiN}}T_{SiN}} \quad (3)$$

where ε_{ox} is the permittivity of the tunnel oxide layer, ε_{SiN} is the permittivity of the charge trap nitride layer, T_{TOX} is the thickness of the tunnel oxide and T_{SiN} is the thickness of the charge trap nitride layer (Fig. 1(b)). Utilizing equations (1)-(3) and the structural parameter values from Table I [13], the write energy is obtained as E_{write} = 0.16 fJ for programming one input.

Once the inputs are encoded as the threshold voltages of the flash cells in 3D NAND string, the output of the logic gate is obtained by applying a read voltage (V_{WL} = 2.5 V) on WLs corresponding to the programmed cells and a pass voltage (V_{pass} = 8 V) to the remaining cells in the string and a bit line voltage corresponding to one input of the logic gate (V_{BL} = 0.5 V for logic '1' and V_{BL} = 0 V for logic '0'). The string current during this read process gets accumulated as the voltage on the input capacitance of the sense amplifier (C_{CSL} = 60 fF) and a read energy given by $E_{read} = C_{CSL} \times V_{BL}^2$ = 30 fJ is consumed in the process.

The write energy (E_{write}), the read energy (E_{read}) and the total energy dissipated by the widely used two-input logic gates for different input combinations are reported in Table II. As can be observed from Table II, the worst-case energy consumption of all the logic functions realized using the proposed methodology is same. Moreover, the worst-case (assuming all inputs of a single string need to be programmed) energy dissipation of an *N*-input logic function implemented utilizing this scheme can be obtained as $(N-1) \times E_{write} + E_{read}$. For a 177-input logic-in-memory implementation exploiting 3D NAND flash array with 176 layers, the worst-case energy dissipation is 58.32 fJ which is significantly lower than the multi-stage CMOS implementation.

B. Delay Calculation

The delay of the proposed logic-in-memory implementation utilizing 3D NAND flash memory depends significantly on the critical input (applied on the bit line BL). The delay is obtained as the temporal difference between 50% of the

TABLE III
DELAY FOR DIFFERENT WL LAYERS AND PULSE DURATION

No. of layers	Worst case t_{PLH}	Worst case t_{PHL}	Worst case delay
Input pulse width = 10 µs			
16	102.82ns	16.75ns	59.78ns
64	157.24ns	62.74ns	109.99ns
128	237.96ns	134.07ns	186.01ns
176	293.47ns	187.64ns	240.55ns
Input pulse width = 100 µs			
16	23.58ns	18.57ns	21.07ns
64	78.63ns	66.45ns	72.54ns
128	150.53ns	131.48ns	141ns
176	203.37ns	181.13ns	192.25ns
Input pulse width = 1 ms			
16	32.23ns	18.73ns	25.48ns
64	62.78ns	68.26ns	65.52ns
128	302.71ns	133.04ns	217.87ns
176	413.19ns	182.76ns	297.97ns

critical input voltage and 50% of the output voltage accumulated on the common source line.

We have performed an extensive analysis of the delay of different logic gates considering a read voltage V_{read} = 2.5 V for the programmed WLs and a pass voltage V_{pass} = 8 V for the unprogrammed WLs for 3D NAND flash strings with 16, 64, 128 and 176 WL layers (or flash cells in a string). Moreover, the time period of the critical input BL pulse is taken as 10 µs considering the typical random read time (t_R) of commercial 3D NAND flash memory [26]-[28]. The worst-case delay for logic-in memory implementation utilizing 3D NAND flash array with 16, 64, 128, and 176 WLs in the string for critical input pulse width of 10 µs, 100 µs and 1 ms are mentioned in table III. The worst-case delay increases with the number of WLs in the string. It may be noted that we have decoupled the programming time i.e., the time required to encode the inputs as the threshold voltage of the flash cells (typically 2 ms per input) from the read time required for performing in-memory logic operation while analyzing the delay following the approach used in [3]-[4].

V. RECONFIGURABLE SYSTEM-LEVEL ARCHITECTURE

We also propose a novel reconfigurable system-level architecture for ultra-fast implementation of different logic gates with different input combinations as shown in Fig. 4. In the proposed architecture, we exploit the ultra-high density of 3D NAND flash memory and pre-program different logic gates with different input combinations in different strings across the 3D NAND flash array in the form of logic gate planes (Fig. 4) and use a content addressable memory (CAM) (which can also be implemented using a 3D NAND flash [29]) to search the strings encoding the logic function to be performed.

The inputs and the logic function to be performed are encoded and given as a search query vector to the CAM. The CAM stores the information regarding the location of the logic gate planes performing a particular logic operation on a set of inputs in the pre-programmed 3D NAND flash array. The flash cells within different strings encoding a logic function are programmed in the 3D NAND flash array such that the output of the logic function is obtained as the voltage on C_{CSL} when input literals are applied on bit lines (BLs) and bit select lines

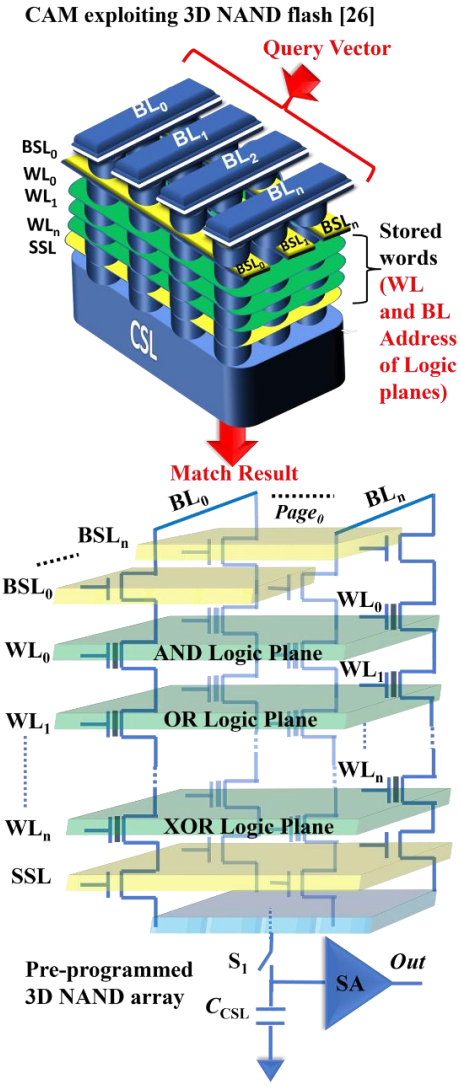

Fig. 4. System level architecture for improving the speed.

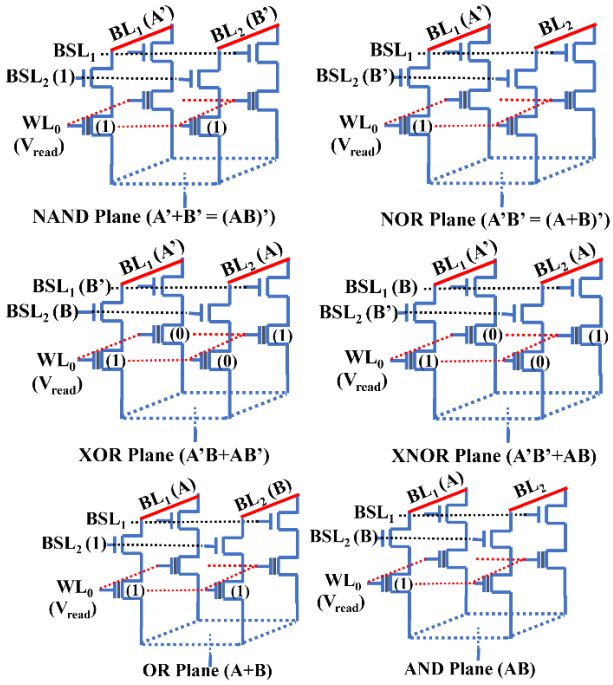

Fig. 5. The logic plane schematic for the pre-programmed 3D NAND flash array used in the system level architecture proposed to increase the speed of 3D NAND flash-based login-in-memory implementation.

(BSLs) during the read process. A read voltage V_{read} = 2.5 V is applied to the WL layer in which the flash cells are pre-

programmed according to the logic operation and a pass voltage of V_{pass} = 8 V is applied to the remaining WLs to obtain the output. For instance, as shown in Fig. 5, we can realize the widely used logic functions in our proposed architecture by applying the inputs on BLs and BSLs and pre-programming the threshold voltage of flash cells in different strings such that each string encodes one minterm of the logic function. Moreover, we can reconfigure the same 3D NAND flash array for different logic operations by re-programming the threshold voltage of flash cells in different strings.

The CAM returns a match result which is used to fetch the logic output from the pre-programmed 3D NAND flash array if the query vector encoding the input combination and logic function is found within the CAM array. However, if there is a mismatch and the query vector corresponding to a particular combination of logic function and inputs is not found in the CAM array, the 3D NAND flash array performing logic-in-memory computation is updated to include the missing logic function and the set of inputs. Such an approach can significantly increase the speed of the 3D NAND flash-based logic-in-memory implementations.

VI. Immunity Against Reverse Engineering Attacks

As discussed in the introduction section, reverse engineering, which involves de-packaging and delayering of the IC and subsequent imaging of different layers to extract the information regarding the underlying design, poses a serious challenge to the semiconductor design companies. Although layout level IC camouflaging techniques [17] increase the resilience against RE attacks, they offer limited number of gates with similar layout (NAND, NOR, XOR) and introduce a significant delay, power consumption and area overhead which restricts widespread (100%) obfuscation while satisfying the design constraints. Therefore, there is an urgent need for a camouflaging technique which does not introduce significant delay/power/area overhead and provides complete (100%) logic obfuscation for enhanced immunity against RE attacks. Since all the logic functions with ≤177 literals/inputs and $\leq 2^{14}$ minterms appear similar at the layout level in the proposed logic-in-memory utilizing 3D NAND flash array, our implementation may provide an efficient means to thwart the RE attacks without introducing any overheads.

To analyze the potential of the proposed logic-in-memory implementation utilizing 3D NAND flash as a logic camouflaging technique, we have performed various reverse engineering attacks on different ISCAS'85 [18] and ISCAS'89 [19] benchmark circuits with logic gates replaced by the camouflaged logic gates utilizing the ultra-dense 3D NAND flash memory.

VII. Resilience To Satisfiability (SAT) Attacks

The efficiency of a circuit camouflaging technique is evaluated by analyzing its resiliency against SAT attacks [30]. The inherent assumptions while performing the SAT attacks are: the reverse engineer has access to (a) functional chip and (b) RE tools which facilitate de-packaging, delayering, imaging of the layout and extraction of gate level netlist. Moreover, (c) the RE attacker can differentiate between the layouts of camouflaged and non-camouflaged gates and (d) the attacker also possesses the list of functions that can be implemented using the camouflaged cell.

To identify the function of all camouflaged gates in the circuit, the reverse engineer extracts all the input/output patterns from a functional IC and applies these patterns on another functional IC while analyzing the outputs. The critical step while de-camouflaging the IC using SAT attacks is to generate input patterns that help in resolving the functionality of the camouflaged gate using least number of iterations. This is achieved utilizing two basic principles of VLSI design for test (DFT) used for analysis of manufacturing defects in ICs [31-32]: justification and sensitization. While justification refers to the process of controlling one or more inputs of the gate to obtain a certain output, sensitization involves application of inputs to the gate in such a way that the output is controlled only by the input to be sensitized. For instance, application of '0' as one input to an AND gate justifies its output to '0' while application of '1' as an input to the AND gate forces the output to depend only on the other input increasing its sensitivity.

To analyze the efficacy of the proposed implementation for IC camouflaging, we have utilized the (fast) incremental SAT solver tool [30], [33]. Although the proposed camouflaging technique is capable of realizing any logic function with ≤177 literals/inputs and $\leq 2^{14}$ minterms without introducing any area overhead in the layout, the incremental SAT solver tool supports analysis of camouflaged gates with only two inputs. For logic gates with only two inputs, the total number of distinguishable truth tables (functions) which may be realized is 16. Considering this limitation of the tool, we restricted the number of logic functions that may be realized using the camouflaged gate to 16. The incremental SAT solver utilizes a mux-based modelling approach for the camouflaged gate as shown in Fig. 6. In the mux-based modelling approach, the control vectors (P_i, $P_{i+1,....}$) dictate the logic function realized by the camouflaged gate. The SAT solver tries to identify the control vectors which yield correct input/output patterns (obtained via the RE of functional IC) in an iterative manner to identify the logic function realized by the camouflaged gates [30],[33].

We used the incremental SAT solver on ISCAS'85 and ISCAS'89 benchmark circuits with all the logic gates replaced by the proposed logic gate implementation utilizing 3D NAND flash memory. Although the SAT solver was able to resolve the functionality of the six camouflaged gates and break the simple c17 circuit of the ISCAS'85 benchmark circuits [18] within 39 secs, it was not able to break (decode) the larger circuits like c432 and c499 benchmarks even after running the tool with an inexhaustive resource space for more than 15 hours. This clearly indicates the significant increase in the immunity of the benchmark circuits with large number of camouflaged logic gates against SAT attacks. Moreover, we have compared the resiliency of the proposed implementation against de-camouflaging using SAT attacks on ISCAS'85 benchmark circuits with recent camouflaging techniques in Table IV. The inherent camouflaging of the proposed logic-in-memory architecture utilizing ultra-dense 3D NAND flash memory exhibits a significantly reduced vulnerability to SAT attacks as compared to the recent obfuscation techniques based on CMOS [17] and 2D hetero-structures [34].

Although we have replaced all the logic gates of the ISCAS'85 benchmark circuits with the mux-based camouflaged gates while performing SAT attacks, the area overhead introduced by the other obfuscation techniques restrict such an approach while designing logic circuits. To reduce the area overhead, only 5% of the logic gates were

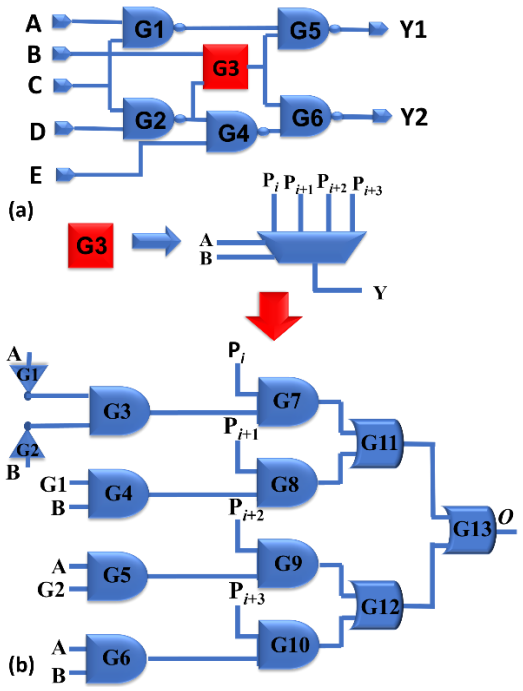

Fig. 6. (a) c17 circuit of ISCAS'85 circuit. G3 gate is replaced by a Mux-based model of camouflaged gate with an internal structure shown in (b).

camouflaged in the CMOS-based camouflaging technique [17] and silicon nanowire (SiNW) [35] based obfuscation technique. Although the 2D hetero structure based camouflaging technique [34] may be used to camouflage all the logic gates owing to the significantly reduced area overhead, the camouflaged gates can only implement NAND/NOR/AND/OR gates. On the other hand, the inherent camouflaging technique offered by the proposed logic-in-memory implementation using 3D NAND flash memory allows realization of all the logic gates without introducing area overhead facilitating complete obfuscation and alleviates the need for hybrid designs. Moreover, the proposed implementation exhibits same delay and energy dissipation for all the logic operations. Therefore, it is also expected to show high robustness against side-channel attacks.

VIII. Resilience to ATPG Attacks

The automatic test pattern generation (ATPG) tool [36] relies on 'activation' and 'propagation' of faults. For efficient detection of single fault models such as stuck-at-fault utilizing ATPG attacks, there should be a clean path between the inputs and the outputs. i.e., the gate being attacked should be connected to either primary inputs or the primary outputs through standard logic gates. Recently, the ATPG tools have also been exploited for de-camouflaging obfuscated circuits [37]. During the ATPG attack, the IC is considered as a black box and the outputs corresponding to different input patterns generated by the ATPG are analyzed. Each camouflaged gate is replaced by a standard logic gate and the test patterns are generated according to the expected functionalities. For instance, for a 2-input camouflaged gate which may exhibit 16 different logic functions, the total number of possible input patterns is 64 ($2^2 \times 16$). Furthermore, if 'm' camouflaged gates are connected in series, they are replaced by a single dummy gate with 'm+1' inputs. Moreover, if each camouflaged gate exhibits 'N' different logic functions, the total number of input patterns to be generated increases exponentially to $2^{m+1} \times N^m$. Since the proposed logic-in-memory implementation exploiting ultra-dense 3D NAND flash memory can realize any logic function with ≤177 literals/inputs and $\leq 2^{14}$ minterms, the number of test patterns required for ATPG attack would be significantly large. Moreover, the proposed technique facilitates full (100%) camouflaging without introducing any significant overhead and mitigates the possibility of having clean paths for activation and propagation of faults. Therefore, the ATPG tool failed to de-camouflage even the smallest c17 benchmark circuit indicating the strong resilience of the proposed technique against ATPG attacks.

TABLE IV
DeCamouflaging Efficiency Of SAT Attacks on Different ISCAS Benchmark Circuits [34]

ISCAS benchmark circuit	FinFET based camouflaged circuit	2D hetero structures based camouflaged circuit	Our proposed 3D NAND based camouflaged circuit
c17	yes	Yes	yes
c432	yes	Yes	no
c499	yes	Yes	no
c1908	no	no	no
c2670	yes	no	no
c3540	yes	no	no
c7552	no	no	no

IX. Resilience To Brute Force Attack

Brute-force attack, which relies on enumeration of all possible combinations of logic gates to realize a particular function, is considered as the ultimate solution to de-camouflage an obfuscated circuit. For the two input gates in the ISCAS'85 benchmark circuits, the proposed implementation may realize at least 16 different functionalities. Considering the smallest c17 benchmark circuit which consists of 6 gates, the number of possible logic combinations is huge (16^6 = 16,777,216). Furthermore, for complex benchmark circuits such as c7552 consisting of 2362 gates, the possible logic combinations increase exponentially (16^{2362}). This makes the brute force attack computationally complex, resource intensive and time consuming. Therefore, the attacker may not be able to resolve the camouflaged gates in a reasonable time with limited resources. Furthermore, since the proposed implementation facilitates realization of more than 16 logic functions (any logic function with ≤177 literals/inputs and $\leq 2^{14}$ minterms), the possible combinations are expected to be significantly large increasing its robustness against the brute force attack.

X. Conclusions

In this work, we have proposed a novel architecture for realizing multiple multi-variable logic-in-memory utilizing 3D NAND flash memory array. The proposed implementation is compact, highly energy-efficient and enables realization of any logic function with ≤ 177 literals/inputs and ≤ 2^{14} minterms. We have also proposed a novel reconfigurable system level architecture exploiting a CAM to further improve the efficiency of logic-in-memory implementation. Moreover, we have also demonstrated an innate camouflaging technique utilizing the proposed architecture which facilitates complete obfuscation of logic gates and benchmark circuits without introducing any area overhead while exhibiting a strong resilience to SAT attacks, ATPG attacks and brute force attacks. We believe that this work is an important step in the

direction of exploiting ultra-dense 3D NAND flash memories for secured processing-in-memory architectures which are immune against reverse engineering. Our results may provide the incentive for experimental demonstration of 3D NAND flash memory based multiple multi-input logic-in-memory primitives.

REFERENCES

[1] A. Agrawal, A. Jaiswal, C. Lee, and K. Roy, "X-sram: Enabling in-memory boolean computations in cmos static random-access memories," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4219–4232, Jul 2018. doi: 10.1109/TCSI.2018.2848999.

[2] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Fast bulk bitwise and and or in dram," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 127–131, May 2015. doi: 10.1109/LCA.2015.2434872.

[3] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic—memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 61, no. 11, pp. 895–899, Sep 2014. doi: 10.1109/TCSII.2014.2357292.

[4] S. Gupta, M. Imani, and T. Rosing, "Felix: Fast and energy-efficient logic in memory" *IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE*, pp. 1–7, Nov 2018. doi: 10.1145/3240765.3240811.

[5] S. Shirinzadeh, M. Soeken, P.-E. Gaillardon, and R. Drechsler, "Logic synthesis for rram-based in-memory computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1422–1435, Sep 2017. doi: 10.1109/TCAD.2017.2750064.

[6] C.-Y. Wen, J. Li, S. Kim, M. Breitwisch, C. Lam, J. Paramesh, and L. Pileggi, "A non-volatile look-up table design using pcm (phase- change memory) cells" *Symposium on VLSI Circuits-Digest of Technical Papers. IEEE*, pp. 302–303, Jun 2011.

[7] H. Kim, S.-J. Ahn, Y. G. Shin, K. Lee, and E. Jung, "Evolution of nand flash memory: From 2d to 3d as a storage market leader," *IEEE International Memory Workshop (IMW)*, pp. 1–4, May 2017. doi: 10.1109/IMW.2017.7939081.

[8] M. Bavandpour, S. Sahay, M. R. Mahmoodi, and D. Strukov, "Mixed-signal vector-by-matrix multiplier circuits based on 3d-nand memories for neurocomputing," *Design, Automation & Test in Europe Conference & Exhibition*, pp. 696–701, Mar 2020. doi: 10.23919/DATE48585.2020.9116401.

[9] P. Wang, F. Xu, B. Wang, B. Gao, H. Wu, H. Qian, and S. Yu, "Three-dimensional nand flash for vector–matrix multiplication," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 988–991, Dec 2018. doi: 10.1109/TVLSI.2018.2882194.

[10] H. T. Lue, P. K. Hsu, M. L. Wei, T. H. Yeh, P. Y. Du, W. C. Chen, K. C. Wang, and C. Y. Lu, "Optimal design methods to transform 3D NAND flash into a high-density, high-bandwidth and low-power nonvolatile computing in memory (nvCIM) accelerator for deep-learning neural networks (DNN)," In *2019 IEEE International Electron Devices Meeting (IEDM),* pp. 38-1, 2019. doi: 10.1109/IEDM19573.2019.8993652

[11] H. T. Lue, W. Chen, H. S. Chang, K. C. Wang, and C. Y. Lu, "A novel 3D AND-type NVM architecture capable of high-density, low-power in-memory sum-of-product computation for artificial intelligence application," In *IEEE Symposium on VLSI Technology*, pp. 177-178, 2018. doi: 10.1109/VLSIT.2018.8510688.

[12] M. Bavandpour, S. Sahay, M. R. Mahmoodi and D. Strukov, "3D-aCortex: An Ultra-Compact Energy-Efficient Neurocomputing Platform Based on Commercial 3D-NAND Flash Memories", *IOP Neuromorphic Computing and Engineering*, vol. 1, pp. 014001, July 2021. doi: 10.1088/2634-4386/ac0775

[13] S. Sahay and D. Strukov, "A behavioral compact model for static characteristics of 3d nand flash memory," *IEEE Electron Device Letters*, vol. 40, no. 4, pp. 558–561, 2019. doi: 10.1109/LED.2019.2901211.

[14] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proc. of IEEE/ACM Design Automation Conference*, pp. 333–338, Jun 2011.

[15] Silicon Zoo, "The layman's guide to ic reverse engineering," http://siliconzoo.org/tutorial.html.

[16] J. P. Baukus, L. W. Chow, R. P. Cocchi, and B. J. Wang, "Method and apparatus for camouflaging a standard cell based integrated circuit with micro circuits and post processing," US Patent no. 20120139582, 2012.

[17] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp.709–720, Nov 2013. doi: 10.1145/2508859.2516656.

[18] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the iscas-85 benchmarks: A case study in reverse engineering," *IEEE Design & Test of Computers,* vol. 16, no. 3, pp. 72–80, Jul 1999. doi: 10.1109/54.785838.

[19] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1929–1934, May 1989. doi: 10.1109/ISCAS.1989.100747.

[20] R. Micheloni, S. Aritome, and L. Crippa, "Array architectures for 3-d nand flash memories," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1634–1649, May 2017. doi: 10.1109/JPROC.2017.2697000.

[21] J.-W. Park, D. Kim, S. Ok, J. Park, T. Kwon, H. Lee, S. Lim, S.-Y. Jung, H. Choi, T. Kang, et al., "30.1 a 176-stacked 512gb 3b/cell 3d-nand flash with 10.8 gb/mm 2 density with a peripheral circuit under cell array architecture," in *IEEE International Solid-State Circuits Conference (ISSCC),* vol. 64, pp. 422–423, Feb 2021. doi: 10.1109/ISSCC42613.2021.9365809.

[22] W.-C. Chen, H.-T. Lue, Y.-H. Hsiao, T.-H. Hsu, X.-W. Lin, and C.-Y. Lu, "Charge storage efficiency (cse) effect in modeling the incremental step pulse programming (ispp) in charge-trapping 3d nand flash devices," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 5–5, Dec 2015. doi:10.1109/IEDM.2015.7409635.

[23] *HSPICE User Guide*: *Basic Simulation and Analysis*, Synopsys, Inc., Mountain View, CA, USA,2018.

[24] D. Resnati, A. Mannara, G. Nicosia, G. M. Paolucci, P. Tessariol, A. S. Spinelli, A. L. Lacaita, and C. M. Compagnoni, "Characterization and modeling of temperature effects in 3-D NAND flash arrays— Part I: Polysilicon-induced variability*," IEEE Trans. Electron Devices*, vol. 65, no. 8, pp. 3199–3206, Aug. 2018. doi: 10.1109/TED.2018.2838524.

[25] G. Malavena, A. L. Lacaita, A. S. Spinelli, and C. M. Compagnoni, "Investigation and compact modeling of the time dynamics of the GIDL-assisted increase of the string potential in 3-D NAND flash arrays," *IEEE Trans. Electron Devices*, vol. 65, no. 7, pp. 2804–2811, Jul. 2018. doi: 10.1109/TED.2018.2831902

[26] S. Sahay, M. Klachko, and D. Strukov, "Hardware security primitive exploiting intrinsic variability in analog behavior of 3-d nand flash memory array," *IEEE Transactions on Electron Devices*, vol. 66, no. 5, pp. 2158–2164, Jul 2019. doi: 10.1109/TED.2019.2903786.

[27] Micron Technology. Micron 3D NAND Flash Memory. Accessed: Dec. 23, 2018. [Online]. Available: https://www.micron.com/~/media/documents/products/..flyer/3d_nand_flyer.pdf.

[28] Samsung V-NAND Technology. Accessed: Dec. 23, 2018. [Online]. Available: https://www.samsung.com/us/business/oem-solutions/pdfs/VNAND_technology_WP.pdf.

[29] H. Yang, P. Huang, R. Han, Y. Xiang, Y. Feng, B. Gao, J. Chen, L. Liu, X. Liu, and J. Kang, "A novel high-density and low-power ternary content addressable memory design based on 3d nand flash," in *IEEE Silicon Nanoelectronics Workshop (SNW)*, pp. 29–30, Jun 2020. doi: 10.1109/SNW50361.2020.9131662.

[30] C. Yu, X. Zhang, D. Liu, M. Ciesielski, and D. Holcomb, "Incremental sat-based reverse engineering of camouflaged logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 10, pp. 1647–1659, Jan 2017. doi: 10.1109/TCAD.2017.2652220.

[31] M. Bushnell and V. Agrawal, Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits. Springer Science & Business Media, 2004, vol. 17.

[32] M. Abramovici, M. A. Breuer, A. D. Friedman, et al., Digital systems testing and testable design. Computer science press New York, 1990, vol. 2.

[33] https://ycunxi.github.io/ Incremental-SAT-DeCam/

[34] A. Wali, S. Kundu, A. J. Arnold, G. Zhao, K. Basu, and S. Das, "Sat-isfiability attack-resistant camouflaged two-dimensional heterostructure devices" *ACS nano*, vol. 15, no. 2, pp. 3453–3467, Jan 2021. doi: 10.1021/acsnano.0c10651.

[35] Q. Alasad, J.-S. Yuan, and P. Subramanyan, "Strong logic obfuscation with low overhead against ic reverse engineering attacks," *ACM Transactions on Design Automation of Electronic Systems (TODAES),* vol. 25, no. 4, pp. 1–31, Jun 2020. doi: 10.1145/3398012.

[36] Tools, Synopsys Test. "TetraMAX ATPG." (2003).

[37] D. Vontela and S. Ghosh, "Methodologies to exploit atpg tools for de-camouflaging," in *18th International Symposium on Quality Electronic Design (ISQED)*, pp. 250–256, Mar 2017. doi: 10.1109/ISQED.2017.7918324.