Fault Modeling and Test Generation for Technology-Specific Defects of Skyrmion Logic Circuits

Ziqi Zhou, Ujjwal Guin, Peng Li, and Vishwani D. Agrawal
Department of Electrical and Computer Engineering
Auburn University, AL, USA
{ziqi.zhou, ujjwal.guin, peng.li, agrawvd}@auburn.edu

***Abstract*—This paper advances the recent results on testing skyrmion logic circuits, which recently gained popularity as an emerging technology. A skyrmion circuit differs significantly from the existing CMOS circuit in physical structure and operation mechanisms. The previous work identified 19 defect types and modeled them as either a stuck-at fault, no-fault causing no error, or a technology-specific defect requiring special consideration. The previous work was limited to those defects that map onto single stuck-at faults. The present work addresses the class of technology-specific defects that were not discussed before. Our defect mapping onto an analyzable fault model uses extensions of fault equivalence and fault dominance principles. We model the defects as transition faults whose test generation is supported in the logic-level EDA systems. All such defects require two-pattern tests, except one defect, missing annihilation notch of OR gate, that needs three patterns. These require test generation for constrained stuck-at fault, generally available in EDA systems. The reported results show that majority of the defects of skyrmion-based circuits can be detected using the proposed test generation approach; few exceptions are defects that map through dominance onto faults rendered redundant due to the circuit structure.**

***Index Terms*—Skyrmion, micromagnetic logic, fault model, technology-dependent fault, stuck-at fault, stuck-open fault.**

I. Introduction

The economics of cost per transistor has led to technology scaling, which refers to the shrinking of device and interconnect geometries on integrated circuits. Besides increasing the transistor density and the resulting reduction in the cost per transistor, scaling has bonuses of higher speed and reduced power. Gordon Moore based his predictions on this observation [1]–[3]. However, continued scaling has given rise to other effects of higher static power (due to higher leakage), dropping yields (due to process variation), poor reliability, and rising fabrication costs.

For the past few years, technologists have been predicting the end of scaling. Of course, these predictions are a direct consequence of what has happened to the CMOS technology. The chip technology has advanced in three directions to combat the difficulties arising from scaling. First, improved fabrication methods have recently brought the feature size down to 2 nanometers [4]. Second, new geometries, such as 3-D device structures of nanosheet [5] or finFET [6] and others [7], have evolved. Third, a shift from semiconductors to other materials and physical phenomena, such as magnetic skyrmion [8], [9], carbon nanotube (CNT) [10], and topological insulator [11], to construct switching devices has shown new possibilities.

This article focuses on the third category, generally referred to as emerging technologies. For skyrmion [8], [9] circuits, we discuss the technology-specific defects that have been formulated before [12] but their analyzable fault models and test methods, as developed here, have never been addressed before. The main contribution of this work is contained in Sections V through VII, as outlined in the next paragraph.

This paper is organized as follows. The background of the skyrmion-based design is provided in Section II. Components of skyrmion hardware and basic gate structures are outlined. In Section III, we list nineteen defects of skyrmion logic as extracted from the existing literature. Section V first summarizes and then enhances the previous results on mapping defects onto stuck-at faults of equivalent logic gates. The previous work used fault equivalence for the defect to fault mapping. That, however, left out certain defects. The present enhancement completes the mapping using fault dominance. The section continues with defect mapping onto nonclassical faults (the novelty of this paper). It ends with an automatic test pattern generation (ATPG) algorithm. In Section VI, we discuss the experimentation on benchmark circuits, demonstrating that just using stuck-at faults may not be sufficient. Finally, we conclude the paper in Section VII.

II. Background

Recent researches have brought skyrmion into practical domain [13]. Skyrmion evolved from a concept mathematically proposed by British nuclear physicist Tony Hilton Royle Skyrme in 1962 [14]. In the next twenty years, the concept gave rise to a pseudoparticle called skyrmion, a stable two-dimensional pattern of the magnetic field whose movement is electrically controllable. It can be created, moved, and annihilated by magnetic fields and low electrical current pulses [15]. It has a diameter varying from tens of nanometers to a few microns [8], [16]. Skyrmions can provide an ideal platform for implementing novel logic and memory designs [8], [14].

The nanometer diameter, room-temperature stability, current-controlled motion, topological charge, and symmetry protection against large defects make skyrmions promising candidates for beyond-Moore systems even though the magnetic Hall effect causes their non-linear motion pose some challenges [17], [18]. Over the years, various skyrmion logic gates have been proposed. They utilize effects of skyrmion movement resulting from the spin-orbit torque-induced motion [19], [20], skyrmion

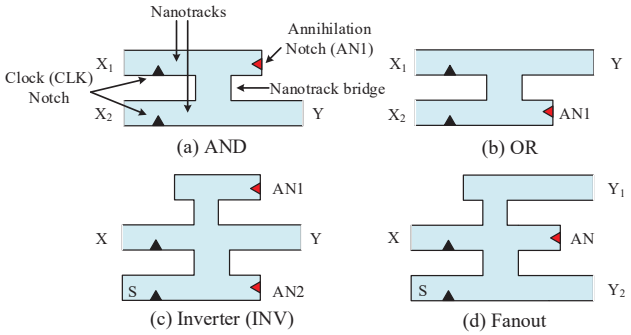


Figure 1: Structures of skyrmion gates (inputs denoted by X's and outputs by Y's) [12] for (a) AND gate (b) OR gate (c) Inverter (INV), and (d) Fanout. S in the lower nanotracks of INV and fanout is a single skyrmion source implemented by a magnetic tunnel junction (MTJ) [32], pulsed every clock cycle.

Hall effect [21]–[25], skyrmion-edge repulsion [26]–[28], and voltage control of magnetic anisotropy effect [29]–[31].

In this paper, we have used the same logic gates that were introduced in a recent paper [12]. Figure 1 shows basic two-input AND and OR, inverter, and fanout structures. Each consists of two or more nanotracks with a junction, making the gate a transversely H-shape structure. The blue triangle on the inputs side is a clock notch to synchronize the input skyrmions so that the output of the gate is evaluated correctly based on the skyrmion-skyrmion interaction. The clock notch has the same material as the ferromagnetic layer. Note that the voltage-controlled magnetic anisotropy (VCMA) structure can be used to synchronize the skyrmion [33]. When a standard current is applied, the clock notch can hold/block the skyrmion movement. When a high current pulse is applied, the skyrmions at the inputs can simultaneously cross the notches. At the end of a nanotrack, the red triangle is an annihilation notch, which eliminates any arriving skyrmion. For the inverter and fanout, we need to add a source S, where a skyrmion is injected every clock cycle. One can find the detailed functionality of each gate in the literature [12].

Since the nanotracks form a planar circuit, it requires a crossover component where interconnects cross and to achieve it in skyrmion-based circuits, an additional element magnetic tunnel junction (MTJ) is required. An MTJ consists of two layers of magnetic metal separated by an ultra-thin insulating layer [32]. This structure is positioned above the nanotrack where skyrmions are to be generated, or existing ones are to be converted into electrical impulses. To generate a skyrmion, the MTJ is supplied a voltage, and when no voltage is applied, the MTJ produces a voltage when it senses a skyrmion. The crossover is implemented by two MTJ's placed on one of the interconnects, on either side of the other interconnect, with an external wire electrically connecting the MTJ's.

III. Defects in Skyrmion Logic Structures [12]

The recent paper [12] identifies 19 defects for skyrmion gates. Those are listed in Table I. Because of structural differences, each defect is relevant only to certain gates. For example, defect T_1, a break in the nanotrack of input X_1 or X, applies to all gates. But

Table I: Defects in skyrmion gates [12]. Applicability of a defect to a specific gate is shown by checkmark (✓). Inputs are X's and outputs, Y's. AN's are annihilation notches.

Defect			Relevant gates			
Name	Location	Condition	AND	OR	INV	Fanout
T_1	X_1/X track	Break	✓	✓	✓	✓
T_2	X_2 track	Break	✓	✓	✓	✓
T_3	Y track	Break	✓	✓	✓	✓
T_4	AN1 track	Break	✓	✓	✓	✓
T_5	AN2 track	Break			✓	✓
T_6	X_1/X track	Void	✓	✓	✓	✓
T_7	X_2 track	Void	✓	✓	✓	✓
T_8	Y track	Void	✓	✓	✓	✓
T_9	AN1 track	Void	✓	✓	✓	✓
T_{10}	AN2 track	Void			✓	✓
T_{11}	AN1 notch	Missing	✓	✓	✓	
T_{12}	AN2 notch	Missing			✓	
T_{13}	AN notch	Missing				✓
T_{14}	X_1 clk notch	Missing	✓	✓	✓	✓
T_{15}	X_2 clk notch	Missing	✓	✓	✓	✓
T_{16}	Nanotracks	Bridge	✓	✓	✓	✓
T_{17}	Bridge	Broken	✓	✓	✓	✓
T_{18}	Bridge	Broken			✓	✓
T_{19}	MTJ	Missing			✓	✓

the defect T_{12}, i.e., missing $AN2$ annihilation notch, is found only in an inverter. That paper also examined the equivalence between the defects and stuck-at faults of two-input logic gates. The results of defect-to-fault mapping are shown in Table II.

Certain terms in Table I need clarification. A *void* is a defective nanotrack through which skyrmion moves. In the skyrmion technology nanotracks are used for interconnects as well as for the internal structure of logic gates. The effect of a void varies depending on the speed of skyrmion. When the speed is low, the skyrmion stops before the void. But a skyrmion moving at a high speed vanishes upon collision with the void. Another term *bridge* refers to a connection between two nanotracks. It can be part of a gate design or a defect where no bridge was intended. The former is the location of a *broken* bridge defect. The latter is the defect condition of a short between nanotracks. In either case a skyrmion will cross over from one to the other nanotrack.

Table II shows how skyrmion defects map onto equivalent stuck-at faults. This defect-based fault modeling guarantees that a traditional ATPG tool can generate patterns for testing the skyrmion-based circuit. For example, defect T_1 of AND gate is equivalent to the output Y stuck-at-0 (sa0). Because of fault equivalence, T_1 is also equivalent to X_1 sa0 and X_2 sa0. The principle of fault equivalence, used here, says that two faults are equivalent if the truth tables of the two corresponding faulty circuits are identical [34]. Note that the two faults are not in the same structure, but are in two different circuits, one a logic gate model and the other a skyrmion implementation of the same function. The circuits were simulated for an exhaustive set of inputs. The skyrmion gates were simulated using the micromagnetic simulator MuMax3 [35].

We also observe that several defects, e.g., T_3 in fanout, T_4 in AND, OR, and INV, T_5 in INV, and T_9 in OR, do not produce any output error. Hence, they are classified as "no-fault" or "NF"

Table II: Mapping of skyrmion gate defects onto equivalent stuck-at faults in logic gates [12].

Gate Type	Defect																		
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}
AND	$Y/0$	$Y/0$	$Y/0$	NF		$Y/0$	$Y/0$	$Y/0$	$X_1/1$		$*$			$Y/0$	$Y/0$	$Y/0$	$X_1/1$		
OR	$X_1/0$	$X_2/0$	$Y/0$	NF		$X_1/0$	$X_2/0$	$Y/0$	NF		$*$			$*$	$*$	$*$	$X_2/0$		
Inverter	$Y/1$	$Y/0$	$Y/0$	NF	NF	$Y/1$	$Y/0$	$Y/0$	$Y/1$	$Y/1$	$*$	$*$		$Y/1$	$Y/1$	$*$	$Y/0$	$Y/1$	$Y/0$
Fanout	$X/0$	$Y_2/0$	NF	$Y_1/0$	$Y_2/0$	$X/0$	$Y_2/0$	$Y_2/1$	$Y_1/0$	$Y_2/0$			$*$	$Y_2/0$	$Y_2/0$	$*$	$Y_2/1$	$Y_1/0$	$Y_2/0$

Notation: $Y/0$ is Y stuck-at-0, NF is "no-fault", and a blank cell indicates that defect is not relevant to the gate type as shown in Table I. Asterisk (*) marks the unresolved mapping of technology-specific defects. T_9 of AND, which would have appeared as *, has been corrected as equivalent to $X_1 sa1$.

Table III: Exhaustive simulation (truth tables) of 2-input AND gate under no-fault, defect T_9 and various stuck-at fault states. Gate with T_9 was simulated by MuMax3 [35].

Inputs X_1, X_2	Output Y for defect T_9 and stuck-at faults					
	No-fault	T_9	X_1 sa1	X_2 sa1	Y sa0	Y sa1
0 0	0	**0**	**0**	0	0	1
0 1	0	**1**	**1**	0	0	1
1 0	0	**0**	**0**	1	0	1
1 1	1	**1**	**1**	1	0	1

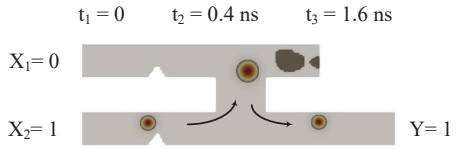


Figure 2: Micromagnetic simulation (MuMax3 [35]) of AND gate with defect T_9 - void in annihilation track. T_9 is detectable by 01, which is a test for X_1 sa1.

in Table II. More details can be found in the original paper [12].

IV. Classical Faults

Stuck-at fault model is often referred to as *classical faults*, placing all other faults into the non-classical category. While the classical fault mapping was done in an earlier paper [12] (see Table III of that paper), defect T_9 - missing annihilation notch of AND gate - was left out. It is correctly shown in Table II above as equivalent to X_1 sa1. The defect to fault equivalence is established as follows.

The present Table III gives results of exhaustive simulation of a two-input AND gate, $Y = X_1 \cdot X_2$ for no-fault and faulty states. The skyrmion version with T_9 was simulated using the MuMax3 micromagnetic simulator [35]. Identical truth tables (bold-face columns) indicate equivalence of defect T_9 and X_1 sa1. Test is 01. The micromagnetic simulation in Figure 2 shows that the skyrmion from X_1 goes up through the connecting channel, is repulsed by the void, and moves to the output nanotrack to produce a faulty response of 1.

V. Technology-Specific Faults

Table II leaves several defects with unresolved fault mapping. These are marked with asterisk (*) and are the focus of the present work. Previously, almost 100% stuck-at fault coverage for benchmark circuits was shown [12]. However, in Section VI we will find that the defect coverage can be significantly lower for the skyrmion version of the same circuit (see Table VII). We will call those unresolved defects as technology-specific defects and map them onto classical or non-classical fault models, mostly, though not always, onto transition faults requiring two patterns. The technology-specific faults in Table II form two groups, one requiring a single test pattern and the other, two or more test patterns. In the following, we discuss each group.

Table IV: Exhaustive simulation of an inverter under no-fault, defect T_{16} and stuck-at fault states. Gate with T_{16} was simulated by MuMax3 [35].

Input X	Output Y for defect T_{16} and stuck-at faults					
	No-fault	T_{16}	X sa0	X sa1	Y sa0	Y sa1
0	1	0	1	0	0	1
1	0	1	1	0	0	1

A. Defect Mapping by Fault Dominance

Similar to the equivalence mapping of defects on fault models, we can use dominance mapping. The principle of *fault dominance* states: For two faults F_1 and F_2, if all tests of F_2 detect F_1, although F_1 may have tests that do not detect F_2, then F_1 is said to dominate F_2 [34]. Thus, F_2 can be safely targeted to derive a test for F_1. However, F_1 can dominate yet another fault, say F_3, which can be an alternate target for finding a test for F_1. In general, F_1 may dominate a set of faults and any test for faults in this set is a test for F_1.

Two asterisked defects from Table II, T_{16} in INV (inverter), and T_{16} in fanout, still map onto stuck faults. We illustrate the mapping of the first of these. The other is done similarly.

Table IV shows exhaustive simulation result for an inverter (INV). The first column shows input combinations. Columns 2 through 7 show the output for no-fault, defect T_{16} and stuck-at faults. T_{16} is a bridging defect between two input tracks. The truth table of INV with T_{16}, i.e., the T_{16} column in Table IV does not match with any of the stuck-fault columns. Thus, no equivalence is found. However, partial matching of columns indicates that defect T_{16} dominates all four stuck-at faults. Thus, a test for any stuck-at fault of INV will detect T_{16}. As an example, we show the micromagnetic simulation by MuMax3 [35] for input $X = 1$ in Figure 3. $Y = 1$ at the output indicates detection of the defect. The test $X = 1$ is easily derived by a logic-level ATPG if either fault X sa0 or Y sa1 is targeted.

B. Defect Mapping onto Nonclassical Faults

Fault models other than the stuck-at faults are generally referred to as *nonclassical* faults. There are several fault models supported for test generation and fault simulation with logic-level circuit description in a commercial tool like Synopsys TestMax [36]. These faults can be stuck-at, bridging, path delay faults, transition delay faults, etc. A single test pattern can detect a stuck-at or a bridging fault, whereas two

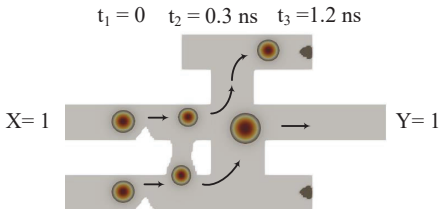


Figure 3: Micromagnetic simulation by MuMax3 [35] for inverter (Figure 1c) with defect T_{16} - a bridging defect between tracks. T_{16} is detectable by 1, which is a test for X sa0 or Y sa1.

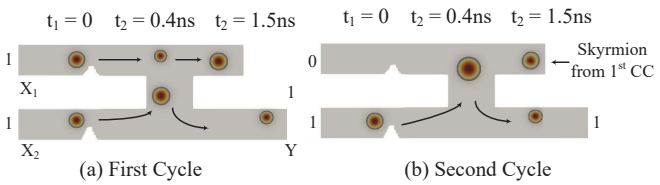


Figure 4: Micromagnetic simulation by MuMax3 [35] for AND gate with defect T_{11} - missing annihilation notch at the right end of top nanotrack (Figure 1a). T_{11} can be detected by pattern sequence 11 and 01 (Example 1).

test patterns are required for a delay (e.g., path or transition) fault. Also, supported is a *constrained or conditional* stuck-at fault, where a stuck-at fault is detected while specified signals must have given values. This generates a single test pattern.

The previous test methodology [12] for skyrmion logic circuits was based on single pattern tests, such as those generated for classical fault models (e.g., stuck-at fault). Thus, existing EDA tools could be used to test the defects without extra effort. However, such tests cannot cover all skyrmion defects. Single pattern tests cannot detect such defects as missing annihilation notch, missing clock notches, and extra bridging defects between two input tracks. These defects need more than one input pattern to observe the faulty response. A convenient and readily available way would be to continue to use the existing EDA tool, and take advantage of generating patterns for other supported fault models in addition to stuck-at faults.

We extend the previous method of exhaustive single-pattern simulation of defective gates to simulation of exhaustive set of pattern-pairs. Skyrmion gates are then compared with faulty logic gates to establish equivalence and dominance between defects and modeled nonclassical faults. Results of this simulation are shown in Tables V and VI, where only the defects not covered by the single-pattern tests are included. Table V has two-input AND and OR gates and Table VI has single-input structures, inverter (INV) and two-output fanout.

Table V shows the two-pattern exhaustive simulation result. The second column indicates the type of defects. None indicates the fault-free gate. StoR and StoF, respectively, indicate that the output Y of the gate has a slow-to-rise and slow-to-fall transition delay fault. $T_{\#}$ is a defect of skyrmion gate as listed in Table I. Note that we only simulate a subset of defects, because the rest of the defects are already detected or covered by stuck-at faults. Columns 3 through 17 show the simulation results of 15 input combinations. Note that we are only considering two-input gates and did not simulate 00-00 input combination, because such input will cause the output to be 0 and no defect can be detected in the skyrmion technology. In this technology, the presence of a skyrmion pseudoparticle represents logic 1 state and absence of skyrmion is logic 0. Thus, for a 00 input, the entire structure of the gate will have no activity and hence it will be impossible to detect any defect.

Table VI gives simulation result for input pairs 0-1, 1-0, 1-1 and 0-0 for single-input structures. Other definitions are similar to Table V.

Examining the simulation data in Tables V and VI, we find that a test for slow-to-fall transition fault can detect missing annihilation notch of AND gate with input pattern-pair [11, 01]. A closer examination of Table V reveals that if all transition faults are detected, then we can ensure that all asterisked defects of Table II will be detected excerpt for just one defect. That defect is the missing annihilation notch defect (T_{11}) in the skyrmion OR gate, which requires more than two patterns to complete the test. We will return to this test after discussing two-pattern tests. Following examples illustrate multi-pattern tests for defect detection.

Example 1: Two-Pattern Test: Consider defect T_{11} of AND gate. For its detection by a test of transition fault (slow-to-rise or slow-to fall), we examine the upper half of Table V. The "None" row shows outputs of fault-free circuit and the next three rows, those of faulty and defective circuits. Note that the first output bit for T_{11} always matches with the fault-free gate indicating that this defect is not detectable by a single pattern. Detection by the second pattern requires that output bits should differ. But we also need a dominated transition fault, whose output bit matches the T_{11} outputs. These conditions are satisfied by 11-01 and 11-10 pattern-pairs for slow-to-fall (StoF) transition fault. Thus, if a StoF fault at the output of AND is targeted then the ATPG will produce a pattern-pair that also detects the defect T_{11}.

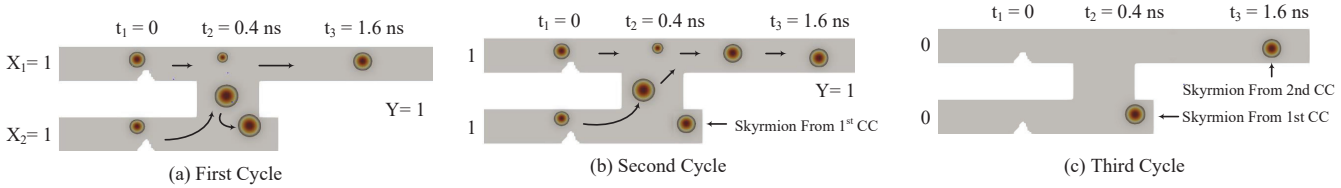
Figure 4 shows how a two-pattern test [11, 01] generated for a slow-to-fall transition fault at the output of an AND logic gate detects the T_{11} defect in the skyrmion version of the AND gate. In the first cycle, input 11 produces a correct output $Y = 1$. Because of the missing annihilation notch, a skyrmion is now left in the upper nanotrack. In the second cycle, input 01 produces a faulty output $Y = 1$ because of the skyrmion-skyrmion repulsion due to the leftover skyrmion.

On rare occasions the circuit structure may make the targeted fault redundant. This does not necessarily make the defect untestable. As can be verified from Figure 4, the pattern-pair [10, 01], which is not a test for the dominance-identified transition fault, is a test for T_{11} of AND gate. This is the reason the industry uses equivalence and not dominance for fault collapsing [34].

Example 2: Three-Pattern Test: The procedure of Example 1 can map all defects in Tables V and VI, except T_{11} of OR gate, which requires a special consideration. Here we must analyze the skyrmion gate structure resulting in a three-pattern test [11, 11, 00], with fault-free outputs [1, 1, 0], only differing in the third pattern to detect the defect. To generate this test by an ATPG tool we would use two faults, a sa1 at OR gate output to produce 00 pattern and then a sa0 at the output with both inputs constrained

Table V: Two-input AND and OR logic (None, StoR and StoF) and skyrmion (None, T_{11}, T_{14}, T_{15} and T_{16}) gates simulated for exhaustive set of input pattern-pairs. StoR and StoF are slow-to-rise and slow-to-fall transition faults at gate output.

Gate	Defect	Input pattern-pair														
		00-01	00-10	00-11	01-00	01-01	01-10	01-11	10-00	10-01	10-10	10-11	11-00	11-01	11-10	11-11
AND	None	0-0	0-0	0-1	0-0	0-0	0-0	0-1	0-0	0-0	0-0	0-1	1-0	1-0	1-0	1-1
	StoR	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	1-0	1-0	1-0	1-1
	StoF	0-0	0-0	0-1	0-0	0-0	0-0	0-1	0-0	0-0	0-0	0-1	1-1	1-1	1-1	1-1
	T_{11}	0-0	0-0	0-1	0-0	0-1	0-1	0-1	0-0	0-1	0-1	0-1	1-0	1-1	1-1	1-1
OR	None	0-1	0-1	0-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1
	StoR	0-0	0-0	0-0	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1
	StoF	0-1	0-1	0-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1	1-1
	T_{11}	0-1	0-1	0-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1
	T_{14}	0-1	0-1	0-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-1	1-1	1-11	1-1
	T_{15}	0-1	0-1	0-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-1	1-1	1-11	1-1
	T_{16}	0-1	0-1	0-1	1-0	1-1	1-1	1-1	1-0	1-1	1-1	1-1	1-1	1-1	1-11	1-1



(a) First Cycle (b) Second Cycle (c) Third Cycle

Figure 5: Micromagnetic simulation of the OR gate with defect T_{11} - missing annihilation notch, using MuMax3 [35]. T_{11} is detected by a three-pattern sequence, 11, 11, and 00 (Example 2).

Table VI: Inverter and fanout logic (None, StoR and StoF) and skyrmion (None, T_{11}, T_{12} and T_{13}) elements simulated for all possible input pattern-pairs. StoR and StoF are slow-to-rise and slow-to-fall transition faults at gate output.

Gate	Defect	Input pattern-pair			
		0-1	1-0	1-1	0-0
INV	None	1-0	0-1	0-0	1-1
	StoR	1-0	0-0	0-0	1-1
	StoF	1-1	0-1	0-0	1-1
	T_{11}	1-0	0-1	0-1	1-1
	T_{12}	1-0	0-1	0-1	1-1
FANOUT	None	00-11	11-00	11-11	00-00
	StoR	00-00	11-00	11-11	00-00
	StoF	00-11	11-11	11-11	00-00
	T_{13}	00-11	11-00	11-11	00-01

to 1 to produce 11 pattern. Note that the OR gate is embedded in a larger circuit at the time of ATPG. The final test is constructed by first duplicating the second test applying 11 to OR and then following it with the first test 00, resulting in the required sequence.

Figure 5 shows the micromagnetic simulation for defect T_{11} in an OR gate. We need to apply a three pattern sequence [11, 11, 00] to detect T_{11} - missing annihilation notch. The first two-patterns set up the necessary conditions to propagate the incorrect response to the output. In the first cycle, input pattern 11 is applied. The skyrmion that is supposed to be destroyed by the annihilation notch remains in the gate. In the second cycle, upon application of 11 again, another skyrmion from the lower track moves to the output track and two skyrmions now exist there. One skyrmion produces $Y = 1$ output and the other is left over. Thus, one skyrmion exists in the output track when in the third cycle input 00 is applied. As a result, the faulty response 1 is observed instead of the correct 0, indicating the manifestation of the defect.

C. Test Pattern Generation

We propose an algorithm that allows an existing commercial EDA tool to generate patterns for skyrmion defects. Algorithm 1 uses the logic-level netlist (C) as an input. The output of this algorithm consists of test patterns. It begins by initializing stuck-at fault test patterns P_s, transition fault test patterns P_t and gate information list L_g (Line 1). From the netlist C it identifies the type for each gate and stores the gate information to list L_g during preprocessing (Line 2). Then iterates over elements in the gate list L_g, adding the mapped stuck-at faults and transition faults to lists FL_s and FL_t (Lines 3-6). After all gates are covered, it applies ATPG to the stuck-at fault list FL_s to generate stuck-at fault patterns P_s (Line 7). Then repeats similar steps to generate transition fault patterns (Line 8). This algorithm can use any current commercial EDA tool, and can be realized using a simple Tcl script. It has a high degree of scalability. Although not elaborated here, simple steps for the three-pattern tests of Example 2 can be included since they only require stuck-at and constrained stuck-at fault processing available in EDA tools.

VI. Results and Discussion

A previous attempt [12] at defect detection used stuck-at-faults (SAF) only. However, ensuring the 100% SAF coverage only detects the defects that map onto SAFs. Several other defects (see Table II) cannot be detected using SAF patterns. As a result, a new coverage metric is required that reflects the detection of all possible defects present in a skyrmion circuit. We define the defect coverage as the ratio of detected defects to the total number of defects. We compute the coverage of defects T_1 through T_{19} detected by ATPG test patterns generated by targeting analyzable faults obtained by defect mapping. This section presents the results.

To evaluate the effectiveness of the proposed test generation process, we used Synopsys tools, Design Compiler [37] for

Table VII: Defect coverage of benchmark circuits implemented in skyrmion technology.

Circuit	Number of gates (2-input AND, OR, and NOT gates)	Number of defects according to mapping procedures				Defect coverage (%)		
		Defects mapped onto SAFs (stuck-at faults)			Mapped onto TFs (transition faults)	Equivalent SAFs	Total SAFs	Total SAFs and TFs
		Equivalence	Dominance	Total SAFs				
c17	8	85	2	87	15	83.3	85.3	100.0
c432	295	3,153	123	3,276	652	80.3	83.4	100.0
c499	841	9,403	413	9,816	1,690	81.7	85.3	100.0
c880	516	5,681	200	5,881	977	82.8	85.8	100.0
c1355	916	10,392	428	10,820	1,667	83.2	86.7	100.0
c1908	764	8,706	388	9,094	1,435	82.7	86.4	100.0
c3540	1,698	19,056	762	19,818	3,163	83.0	86.2	100.0
c6288	4,160	46,997	1,888	48,885	7,605	83.2	86.5	100.0

Algorithm 1: Test Pattern Generation Algorithm for skyrmion Circuits.

Input : The netlist of a circuit (C)
Output : Test pattern set (P_S and P_T)

$P_S, P_T \leftarrow \phi$; $L_g \leftarrow \phi$;
Read netlist C, identify the type of each gate and add the gate information to list L_g;
for *each gate* G *in* L_g **do**
    Add all stuck-at faults to fault list, FL_S ;
    Add all transition delay faults to fault list, FL_T;
end
$P_S \leftarrow ATPG(FL_S)$;
$P_T \leftarrow ATPG(FL_T)$;
return P_S, P_T;

synthesis, and TestMAX ATPG [36] for test pattern generation. We used Synopsys 32nm SAED32 EDK Generic Library [38] for synthesis and test generation.

Table VII shows defect coverage analysis of ISCAS'85 benchmarks [39]. First column gives circuit name. Second column lists the gate count for skyrmion circuit consisting of 2-input AND, 2-input OR, and NOT gates. Columns 3 and 4 show the equivalent and dominant stuck-at faults (SAFs) derived from defect mapping. Columns 5 and 6 are the total SAFs and transition-delay faults (TFs), respectively. Columns 7 through 9 list defect coverages of equivalent SAF tests, total SAF tests, and total SAF and TF tests, respectively, assuming that all faults are detectable. This assumption makes the coverages in Table VII upper bounds. Let us consider the benchmark circuit, c432, consisting of 295 gates. After skyrmion gate defect mapping, we get 3,153 equivalent and 123 dominant stuck-at faults, resulting in a total of 3,276 stuck-at faults. Detecting these 3,153 and 3,276 stuck-at faults provides defect coverages of 80.3% and 83.4%, respectively, for the skyrmion implementation of c432. Finally, the defect coverage rises to 100% after all stuck-at and delay faults are detected. We observe similar defect coverages for all benchmark circuits.

The result is not surprising because the mapping of defects T_1 through T_{19} onto analyzable faults allows detection of all defects with few exceptions. Note that in the previous paragraph we assumed that all targeted faults were detected. In reality, however, a small number of faults may become redundant due to circuit topology. If the mapping was equivalence-based, then the defect would be considered redundant. But, for dominance mapping, the defect may still have other possible tests. See the remark at the end of Example 1 in Section V-B.

VII. Conclusion

This paper advances the research on skyrmion circuits that belong in an emerging technology. We propose a new detection method based on the existing skyrmion circuit. We map defects to analyzable fault models using an extension of the principles of fault equivalence and fault dominance. We model defects as stuck-at and transition faults, for which we can generate the test patterns from logic-level EDA tools. According to our calculations, the test coverage by the proposed method to detect defects can potentially reach 100 percent.

Defects in skyrmion circuits are classified into three categories: (1) defects that are completely mapped onto stuck-at faults; (2) defects not completely mapped as equivalent to stuck-at faults, but detectable through fault dominance by stuck-at fault tests; (3) defects detectable by transition fault tests, requiring two or in a special case three-pattern tests.

The defect-oriented test procedure illustrates that not all defects are truly represented by the classical stuck-at fault model. Hence, the often-used methods of covering all stuck-at faults may not be as reliable as the presented procedure of mapping defects on various fault models using the equivalence and dominance principles.

Since defect mapping requires exhaustive technology simulation only at the single gate level, the computational complexity is manageable. Once the defects are mapped onto analyzable fault models, test generation and simulation are possible with available tools. The method is applicable to any new or emerging technology once a technology-specific simulator is available.

In the present-day design environment, test tools such as test pattern generator and fault simulator work at the logic gate level. Thus, our defect mapping method will work well because it can readily find the appropriate faults in the logic level circuit model. Also, since the fault modeling is done at the gate level, its applicability extends to all circuits, combinational, sequential, asynchronous, etc., for which tools and methodologies exist.

Acknowledgment

Authors thank Martin Keim for meaningful comments on the paper. This work was supported by the National Science Foundation (NSF) under grant CNS-1755733.

REFERENCES

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits," in *Electronics*, vol. 38, no. 8, April 19, 1965.

[2] ——, "Progress in Digital Integrated Electronics," in *IEEE International Electron Devices Meeting (IEDM) Technical Digest*, 1975, pp. 11–13.

[3] ——, "Lithography and the Future of Moore's Law," in *Proc. SPIE*, vol. 2437, May 1995.

[4] D. Johnson, "Big Blue Gets Small > IBM's 2-nanometer Chip is World's First," *IEEE Spectrum*, p. 7, Aug. 2021.

[5] P. Ye, T. Ernst, and M. V. Khare, "The Last Silicon Transistor," *IEEE Spectrum*, pp. 30–35, Aug. 2019.

[6] T. Perry, "The Father of FinFETS," *IEEE Spectrum*, pp. 46–51, May 2020.

[7] J. Radu, "Atom-Thick Transistors," *IEEE Spectrum*, pp. 44–49, Feb. 2020.

[8] A. Fert, V. Cros, and J. Sampaio, "Skyrmions on the Track," *Nature Nanotechnology*, vol. 8, no. 3, pp. 152–156, 2013.

[9] X. Zhang, M. Ezawa, and Y. Zhou, "Magnetic Skyrmion Logic Gates: Conversion, Duplication and Merging of Skyrmions," *Scientific Reports*, vol. 5, no. 1, pp. 1–8, 2015.

[10] M. M. Shulaker, G. Hills, N. Patil, H. Wei, H.-Y. Chen, H.-S. P. Wong, and S. Mitra, "Carbon Nanotube Computer," *Nature*, vol. 501, pp. 526–530, Sep. 2013.

[11] C. Choi, "A Beginner's Guide to Topological Materials," *IEEE Spectrum*, pp. 6–8, Jul. 2021.

[12] Z. Zhou, U. Guin, P. Li, and V. D. Agrawal, "Defect Characterization and Testing of Skyrmion-Based Logic Circuits," in *Proc. IEEE 39th VLSI Test Symposium (VTS)*, 2021, pp. 1–7.

[13] S. L. Zhang, G. van der Laan, and T. Hesjedal, "Direct Experimental Determination of the Topological Winding Number of Skyrmions in Cu2OSeO3," *Nature Communications*, vol. 8, no. 1, p. 14619, Feb 2017. [Online]. Available: https://doi.org/10.1038/ncomms14619

[14] T. H. R. Skyrme, "A Unified Field Theory of Mesons and Baryons," *Nuclear Physics*, vol. 31, p. 556–569, 1962.

[15] K. M. Song, J.-S. Jeong, B. Pan, X. Zhang, J. Xia, S. Cha, T.-E. Park, K. Kim, S. Finizio, J. Raabe *et al.*, "Skyrmion-Based Artificial Synapses for Neuromorphic Computing," *Nature Electronics*, vol. 3, no. 3, pp. 148–155, 2020.

[16] W. Legrand, D. Maccariello, N. Reyren, K. Garcia, C. Moutafis, C. Moreau-Luchaire, S. Collin, K. Bouzehouane, V. Cros, and A. Fert, "Room-Temperature Current-Induced Generation and Motion of Sub-100 nm Skyrmions," *Nano Letters*, vol. 17, no. 4, pp. 2703–2712, Apr 2017. [Online]. Available: https://doi.org/10.1021/acs.nanolett.7b00649

[17] G. Chen, "Skyrmion Hall Effect," *Nature Physics*, vol. 13, no. 2, pp. 112–113, 2017.

[18] T. Dohi, S. DuttaGupta, S. Fukami, and H. Ohno, "Formation and Current-Induced Motion of Synthetic Antiferromagnetic Skyrmion Bubbles," *Nature Communications*, vol. 10, no. 1, pp. 1–6, 2019.

[19] W. Jiang, P. Upadhyaya, W. Zhang, G. Yu, M. B. Jungfleisch, F. Y. Fradin, J. E. Pearson, Y. Tserkovnyak, K. L. Wang, O. Heinonen *et al.*, "Blowing Magnetic Skyrmion Bubbles," *Science*, vol. 349, no. 6245, pp. 283–286, 2015.

[20] G. Yu, P. Upadhyaya, X. Li, W. Li, S. K. Kim, Y. Fan, K. L. Wong, Y. Tserkovnyak, P. K. Amiri, and K. L. Wang, "Room-Temperature Creation and Spin-Orbit Torque Manipulation of Skyrmions in Thin Films with Engineered Asymmetry," *Nano Letters*, vol. 16, no. 3, pp. 1981–1988, 2016.

[21] J. Zang, M. Mostovoy, J. H. Han, and N. Nagaosa, "Dynamics of Skyrmion Crystals in Metallic Thin Films," *Physical Review Letters*, vol. 107, no. 13, p. 136804, 2011.

[26] J. Iwasaki, M. Mochizuki, and N. Nagaosa, "Current-Induced Skyrmion Dynamics in Constricted Geometries," *Nature Nanotechnology*, vol. 8, no. 10, p. 742, 2013.

[22] N. Nagaosa and Y. Tokura, "Topological Properties and Dynamics of Magnetic Skyrmions," *Nature Nanotechnology*, vol. 8, no. 12, p. 899, 2013.

[23] W. Jiang, X. Zhang, G. Yu, W. Zhang, X. Wang, M. B. Jungfleisch, J. E. Pearson, X. Cheng, O. Heinonen, K. L. Wang *et al.*, "Direct Observation of the Skyrmion Hall Effect," *Nature Physics*, vol. 13, no. 2, pp. 162–169, 2017.

[24] G. Chen, "Spin-Orbitronics: Skyrmion Hall Effect," *Nature Physics*, vol. 13, no. 2, pp. 112–113, 2017.

[25] K. Litzius, I. Lemesh, B. Krüger, P. Bassirian, L. Caretta, K. Richter, F. Büttner, K. Sato, O. A. Tretiakov, J. Förster *et al.*, "Skyrmion Hall Effect Revealed by Direct Time-Resolved X-Ray Microscopy," *Nature Physics*, vol. 13, no. 2, pp. 170–175, 2017.

[27] X. Zhang, G. Zhao, H. Fangohr, J. P. Liu, W. Xia, J. Xia, and F. Morvan, "Skyrmion-Skyrmion and Skyrmion-Edge Repulsions in Skyrmion-Based Racetrack Memory," *Scientific Reports*, vol. 5, p. 7643, 2015.

[28] Y. Zhang, C. Tang, P. Li, and U. Guin, "CamSkyGate: Camouflaged Skyrmion Gates for Protecting ICs," in *Design Automation Conference (DAC)*, 2022.

[29] T. Maruyama, Y. Shiota, T. Nozaki, K. Ohta, N. Toda, M. Mizuguchi, A. Tulapurkar, T. Shinjo, M. Shiraishi, S. Mizukami *et al.*, "Large voltage-induced magnetic anisotropy change in a few atomic layers of iron," *Nature Nanotechnology*, vol. 4, no. 3, p. 158, 2009.

[30] P. Upadhyaya, G. Yu, P. K. Amiri, and K. L. Wang, "Electric-Field Guiding of Magnetic Skyrmions," *Physical Review B*, vol. 92, no. 13, p. 134411, 2015.

[31] X. Liang, J. Xia, X. Zhang, M. Ezawa, O. A. Tretiakov, X. Liu, L. Qiu, G. Zhao, and Y. Zhou, "Antiferromagnetic skyrmion-based logic gates controlled by electric currents and fields," *arXiv preprint arXiv:1909.10709*, 2019.

[32] S. Mangin, D. Ravelosona, J. Katine, M. Carey, B. Terris, and E. E. Fullerton, "Current-Induced Magnetization Reversal in Nanopillars with Perpendicular Anisotropy," *Nature materials*, vol. 5, no. 3, pp. 210–215, 2006.

[33] M. Weisheit, S. Fähler, A. Marty, Y. Souche, C. Poinsignon, and D. Givord, "Electric Field-Induced Modification of Magnetism in Thin-Film Ferromagnets," *Science*, vol. 315, no. 5810, pp. 349–351, 2007.

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

[35] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. Van Waeyenberge, "The Design and Verification of Mumax3," *AIP Advances*, vol. 4, no. 10, p. 107133, 2014. [Online]. Available: http://doi.org/10.1063/1.4899186

[36] TestMAX ATPG, Synopsys, https://www.synopsys.com/implementation-and-signoff/test-automation/testmax-atpg.html.

[37] DC Ultra: Concurrent Timing, Area, Power, and Test Optimization, https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html.

[38] Synopsys 32/28nm Generic Library for Teaching IC Design, https://www.synopsys.com/community/university-program/teaching-resources.html.

[39] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Targeted Translator in FORTRAN, Special Session on ATPG and Fault Simulation," in *Proc. 1985 IEEE Int. Symp. on Circuits and Systems (ISCAS'85)*, Jun. 1985, pp. 663–698, *Benchmark circuit netlists are available at* http://www.pld.ttu.ee/∼maksim/benchmarks/iscas85/.