Towards Systematic Design of 3D pNML Layouts

Robert Perricone*, Yining Zhu†, Katherine M. Sanders*, X. Sharon Hu*, and Michael Niemier*
*Department of Computer Science and Engineering, University of Notre Dame
Notre Dame, IN 46556, USA, Email: {rperrico,ksander8,shu,mniemier}@nd.edu
†Department of Information Science and Communication Engineering, Zhejiang University
Hangzhou, Zhejiang Province, China, Email: {3110105028}@zju.edu.cn

***Abstract*—Nanomagnetic logic (NML) is a "beyond-CMOS" technology that uses bistable magnets to store, process, and move binary information. Compared to CMOS, NML has several advantages such as non-volatility, lower power consumption, and radiation hardness. Recently, NML devices with perpendicular magnetic anisotropy (pNML) have been experimentally demonstrated to perform logic operations in three dimensions. 3D pNML layouts provide additional benefits such as simplified signal routing and greater integration density. However, designing functional 3D pNML circuits can be challenging as one must consider the effects of fringing magnetic fields in three dimensions. Furthermore, the current process of designing 3D pNML layouts is little more than a trial-and-error-based approach, which is infeasible for larger, more complex designs. In this paper, we propose a systematic approach to designing 3D pNML layouts. Our design process leverages a machine learning-inspired prediction approach that examines the effects of varying individual device parameters (e.g., length, width, etc.) and predicts functional configurations.**

I. Introduction

Nanomagnetic logic [1] (NML) uses bistable, single-domain nanomagnets to store, process, and propagate binary data via fringing field interactions [1], [2]. NML is inherently non-volatile, radiation-hard, and low-power when compared to charge-based devices (e.g., CMOS). Of recent interest, devices with perpendicular magnetic anisotropy (PMA) (i.e., out-of-plane magnetization states) capable of performing common data operations (e.g., propagation [3], [4], fanout [5], and majority logic [6], [7]) have been experimentally demonstrated. Furthermore, PMA NML (referred to here as pNML) devices have been employed in the fabrication of 3D layouts that perform logic operations within and across functional layers [8], [9]. Compared to traditional 2D layouts, 3D layouts can achieve higher integration density, reduced footprint, and simplified signal routing [9], [10]. However, utilizing additional functional layers increases the complexity of the layout design process as one must consider 3D fringing field interactions.

The authors of [10] introduced a process for deriving 3D pNML layouts. At a high level, this process involved strategically arranging magnets in 3D space based on a desired functionality. Micromagnetic simulations [11] are then leveraged to determine the logical correctness of the layout. If the simulations reveal that the layout does not function as expected, the designer either makes "best-guess" adjustments to the geometry of the layout (e.g., magnet size, spacing, functional layer, etc.) or restarts the design process. However, using trial-and-error to determine the necessary adjustments is a tedious endeavor. The designer generally makes small (10s of nanometers) adjustments to their layout and then waits (hours) for simulation results. While this may be feasible for simple layouts, it is impractical for larger, more complex ones. Possible geometric adjustments are numerous and lengthy simulation times induce a strong desire for a more informed, systematic approach.

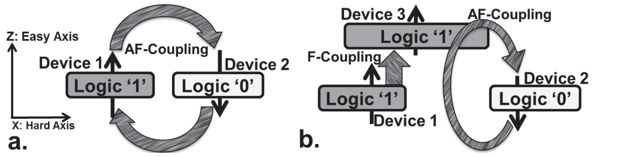

Fig. 1. (a) AF-coupling occurs between pNML devices in the same plane (2D). (b) AF-coupling or F-coupling can occur between pNML devices in different planes (3D) depending on overlap [10].

In this paper, we propose a more efficient layout design process that leverages a machine learning-inspired prediction approach based on regression analysis (Sec. III). Regression analysis has been shown to be a useful technique for efficient design optimization, especially for large design spaces [12], [13]. Here, the goal of our prediction-based approach is to find a functional layout by sampling and simulating a fraction of the design space. This approach involves three key components: design space sampling of geometric parameters, regression analysis of each geometric parameter sampled, and prediction of functional layout configurations.

II. Background

Most of the initial work with NML devices has focused on devices with in-plane magnetic anisotropy (iNML) [1], [2]. More recently, devices with perpendicular magnetic anisotropy (pNML) [14], [15] – illustrated in Fig. 1(a) – have emerged and have several distinct advantages when compared to iNML devices [16], such as being irregularly shaped while remaining single-domain (e.g., L-shape in [7], [17]). Furthermore, pNML devices can be arbitrarily sized, and given the ability to modify its shape, the process of signal routing can be much simpler. Of particular interest, the ability for pNML to leverage additional functional layers to create a 3D signal crossing [8] and perform 3D logic operations [9] has been demonstrated experimentally. Moreover, recent work has suggested 3D designs can perform logic operations within a fraction of the area of 2D designs [9], [10].

The preferred magnetization state of pNML devices depends on the anti-ferromagnetic (AF) coupling and/or ferromagnetic (F) coupling exhibited from their placement in relation to neighboring magnets (Fig. 1(b)). Furthermore, pNML can leverage focused ion beam (FIB) irradiation to establish a direction of dataflow as shown in Fig. 2(a). Essentially, FIB placed on a pNML device (e.g., an edge) increases the influence of fringing fields.

To compute and propagate data, it is necessary to "clock" pNML circuits. Experimental [3], [4], [18] and simulation-based [15] studies suggest that appropriately irradiated pNML structures can be controlled by a uniform, homogeneous, oscillating, global clock field. For example, consider the AF-ordered line schematic in Fig. 2(b) [19]. Here, we (i) refer to the period of the sinusoidal, out-of-plane field as T_{pulse} with peak amplitude of H_{pulse}, and (ii) assume magnets are FIB

978-3-9815370-4-8/DATE15/©2015 EDAA

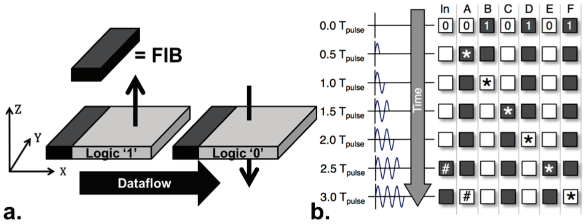

Fig. 2. (a) FIB used to establish dataflow direction for pNML devices (b) Clocking of pNML devices (assume left edge of each magnet is FIB irradiated). $*$ and # are used to show dataflow [19].

irradiated on their left edges to establish dataflow from left-to-right. After the first application of H_{pulse} (in the +z-direction), the magnet immediately adjacent to the flipped input will change state after $0.5T_{pulse}$. Successive applications of the field allow for multiple streams of data to propagate through the ensemble, which creates an inherent pipeline.

Switching occurs as H_{pulse} is of sufficient magnitude to eliminate the parallel-alignment of the pair at the input, but is not of sufficient magnitude to alter the state of properly aligned devices. Specifically, $H^c \leq H_{pulse} < H^{\overline{c}}$ where H^c and $H^{\overline{c}}$ are the field magnitudes necessary to switch the magnet to its correct or incorrect state, respectively. Clearly, a pNML design is functional if $H^c < H^{\overline{c}}$ (with $H_{pulse} := H^c$).

For pNML devices, the out-of-plane clocking field could be generated via inductor structures coupled with a capacitance in an LC oscillator [18], [19]. Other mechanisms have been proposed to clock circuits as well. For example, magnetostriction [20] could serve as a low-power, voltage-controlled alternative to current-based clocking schemes. Multiferroics [21] could control NML devices in a similar fashion. More recently, a clocking mechanism that leverages the Spin Hall effect has been proposed for pNML devices that may provide lower power clocking [22].

Finally, when designing pNML circuits, one must be aware of factors that may affect the behavior of a device. In particular, pNML devices are affected by fabrication variations [3] and thermal noise [23], which cause the switching point of a device to vary. In this paper, these effects are not considered, and we assume that only geometric parameters of the device (e.g., shape, FIB size and placement, etc.) and neighboring fringing fields affect the switching of a device. However, we do show in Sec. III-A where these considerations can be included in the layout design process.

III. Systematic pNML Circuit Design

In this section, we propose a systematic layout design process for pNML devices. We start by presenting an overview of the main elements of our pNML layout design process, which includes an algorithm for determining the logical correctness of both combinational and sequential logic layouts. We then discuss our proposed method of utilizing a machine learning-inspired prediction process to make the design process more efficient.

A. Design Process Overview

Our proposed layout design process, shown in Fig. 3, contains the necessary steps to achieve a functional 3D pNML layout. We summarize the steps of the process below, which also includes an algorithm for determining the logical correctness of both combinational and sequential logic layouts. (Note that steps 1-5 are similar to the process in [10], but we briefly discuss them for the sake of completeness.)

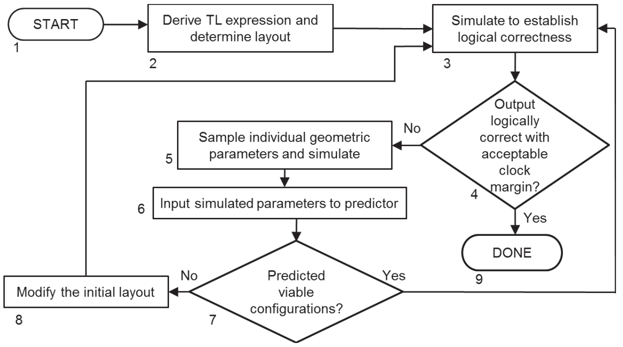
START
1
Derive TL expression and determine layout
2
Simulate to establish logical correctness
3
Output logically correct with acceptable clock margin?
4
No
Yes
Sample individual geometric parameters and simulate
5
Input simulated parameters to predictor
6
Predicted viable configurations?
7
No
Yes
Modify the initial layout
8
DONE
9

Fig. 3. Our proposed pNML layout design process.

Steps 1 and 2. The process begins with the designer deriving the threshold logic (TL) expression – essentially majority voting logic – for the desired circuit function [10]. The TL expression considers the coupling (F or AF) and "weight" (or influence) of the input magnets on the output. Once the designer has derived the TL expression, an initial design layout can be determined. The "weights" of each input in the TL expression give the designer an indication of the layout parameters (e.g., size, spacing, area overlap, etc.) for the input and output magnets.

Step 3. To reduce simulation time, the designer should use this step to determine the unique test cases for establishing functionality. For combinational logic, this can done as in [10] by excluding complementary and symmetrical input tests cases. For sequential logic however, the initial logic state of inputs and outputs may need to be considered (e.g., the initial state of the sum bit in a counter). Accordingly, a state transition diagram can be constructed and traversed to determine the unique input test cases [24], [25].

Step 4. The next two steps in the design process utilize micromagnetic simulations [11] to determine if the initial layout is logically correct. Simulations compute the applied field strength necessary – for a given input test case i – to switch an output magnet to its correct (H_i^c) and incorrect logic ($H_i^{\overline{c}}$) states. In [10], two conditions were established to determine the correctness of a combinational logic layout:

C1: $\forall i \in I \; (H_i^c < H_i^{\overline{c}})$

C2: $\min_{i \in I} \, (H_i^{\overline{c}}) - \max_{i \in I} \, (H_i^c) > \Delta_H$

where I is the set of test cases and Δ_H is the clock margin (i.e., field difference in C2) required to ensure logically correct operation given switching field distributions associated with fabrication variations, thermal noise, etc. While these conditions are appropriate for combinational logic layouts, they are too strict for sequential logic layouts (explained below). Furthermore, the second condition implies the first. We propose an algorithm (see Algorithm 1) to determine the correctness of both sequential and combinational logic layouts.

For Algorithm 1, we define the input B to be the set containing the H_i^c and $H_i^{\overline{c}}$ values for each input test case $i \in I$. To determine the logical correctness of a sequential logic layout, it is first necessary to determine if a clocking field magnitude exists such that the output can be placed into its correct state as necessary (per discussion in Sec. II). Test cases with the output magnet initially in its correct state are *not* considered when determining the magnitude of the clock as they do not impact the logical correctness of the

Algorithm 1 `isLogCorr`. Determine if pNML layout is logically correct

```
1:  Input: $B$ a set containing pairs $(H_i^c, H_i^{\overline{c}})$ for each input
    test case $i \in I$. $I$ also contains initial logic states for
    sequential logic layouts.
2:  Output: true if layout is logically correct, false
    otherwise
3:  $minIncorrect := \infty$
4:  $maxCorrect := 0$
5:  for all input test cases $i \in I$ do
6:    if the output of test $i$ is not initially in correct state then
7:      if $B(H_i^{\overline{c}}) < minIncorrect$ then
8:        $minIncorrect := B(H_i^{\overline{c}})$
9:      end if
10:     if $B(H_i^c) > maxCorrect$ then
11:       $maxCorrect := B(H_i^c)$
12:     end if
13:   else
14:     $C(H_i^{\overline{c}}) \leftarrow B(H_i^{\overline{c}})$ {$C$ contains each $H_i^{\overline{c}}$ for test cases
        where output initially in correct state}
15:   end if
16: end for
17: if $minIncorrect - maxCorrect < \Delta_H$ then
18:   return false
19: end if
20: $clock := maxCorrect$
21: for each input test case $i \in C$ do
22:   if $C(H_i^{\overline{c}}) \leq clock$ then
23:     return false
24:   end if
25: end for
26: return true
```

layout. Accordingly, Algorithm 1 extracts the cases where the output is initially correct and places them into the set C (line 14). The minimum values of $H_i^{\overline{c}}$ and maximum values of H_i^c are determined (lines 8 and 11) and we check if their difference (i.e., clock margin) is less than some Δ_H (line 17). If an appropriate clock margin exists, then the minimum value of H_i^c is chosen as the clock field magnitude (as it should minimize clock energy). Finally, on line 22, we check each of the excluded input cases (set C) to determine if the clock field magnitude would switch the output to its incorrect logic state. This is a relaxation of two conditions from [10] as a sequential logic layout can fail the two conditions, but still be functional. If the clock magnitude does not violate this condition, our sequential logic layout is functional. Note that for combinational logic layouts the above process is similar except C will always be the empty set.

Steps 6-9. While the TL expression determines the "weight" and coupling of devices, it is very difficult to predict how stray fields will impact the 3D layout, especially for each possible test case. Consequently, the initial layout will likely not be functional. If a layout is found to be non-functional (i.e., Algorithm 1 returns `false`), one must determine the necessary adjustments for properly "weighing" the inputs. The remaining steps in the design process provide our methodology for incrementally working towards a functional layout, if one exists. Our process begins by sampling and simulating individual geometric parameters in the design space (Step 6) and recording H_i^c and $H_i^{\overline{c}}$ for each test case. We then employ a machine learning-inspired approach that leverages regression analysis to predict the geometric parameter adjustments needed to achieve a functional layout (Steps 7 and 8). If the predictor fails to output any candidate adjustments, the designer should consider more substantial modifications to the initial layout (Step 9). In the following subsection, we discuss this prediction-based approach (Algorithm 2) in more detail.

B. Predicting functional pNML circuits

Our machine learning-inspired prediction approach is built on several empirical observations. First, the impact of most geometric parameters (e.g., magnet width, length, functional layer, etc.) on H^c and $H^{\overline{c}}$ tend to be somewhat linear. Second, as more parameters are explored, it becomes exceedingly difficult for a designer to determine the best combinations of adjustments that may lead towards a functional layout. Third, there is no guarantee that the trial-and-error-based approach will incrementally move towards a functional layout, or conversely, indicate when a new layout should be attempted. Based on these observations, we employ a prediction process that first quantifies the effects of individually sampled geometric parameters through regression analysis. Then, using the trends obtained by the regression analysis, a predictor estimates the necessary combinations of parameter adjustments to achieve a functional layout.

If an initial layout is determined to be logically incorrect, we then perform a design space sampling (Step 6, Fig. 3) to further iterate towards a functional design. More specifically, using the non-functional layout (Steps 4 and 5) as a starting point, how individual design parameters – (a) the length/width of a magnet, (b) the functional layer in which a magnet resides, (c) the area/location of FIB, (d) inter-magnet distance, and (e) magnet overlap between functional layers – impact H_i^c and $H_i^{\overline{c}}$ are quantified. Note that during the sampling process, the impact of layout geometry design parameters on H_i^c and $H_i^{\overline{c}}$ are considered independently of each other (i.e., only 1 is varied at a time). The range over which a particular design parameter is studied is typically determined by practical/physical constraints – e.g., pNML fringing field distribution (for determining layer height), lithographic resolution (for determining device-to-device spacing), etc. Given a particular design parameter, for each test case within a given range, switching field data (H_i^c and $H_i^{\overline{c}}$) of a target output magnet is recorded. These data sets then become inputs (Steps 7 and 8) to Algorithm 2 (described below).

The algorithm for the prediction process inputs three sets: (1) the set G containing all sampled geometric parameters, (2) the set S_g containing simulation data for each sampled geometric parameter $g \in G$, and (3) R_g containing the range over which a geometric parameter $g \in G$ can be adjusted. The first part of the prediction process is regression analysis (lines 8 and 13). A linear least squares regression function computes the slope $Q_{g,i}$ for every input test case $i \in I$ within each set of geometric parameter simulations $S_{g,i}$ (lines 10 and 11). Once the regression analysis completes, a brute-force search begins over the defined range (i.e., R_g) of all possible geometric parameter adjustments (lines 14 and 15). Towards making a prediction, each currently selected combination of geometric parameter adjustments (K_g on line 16) is first scaled by its slope and then collectively summed for each test case $i \in I$ (lines 19 and 20). These results are then added to the simulation results from the initial layout ($S_{0,i}$) to produce a predicted set B containing H_i^c and $H_i^{\overline{c}}$ for each test case. B

Algorithm 2 `predProcess`. Prediction of functional pNML Layouts

```
 1: Input: $G$ the set of sampled geometric parameters. $S_g$ the
    set of simulations for each geometric parameter $g \in G$.
    $R_g$ the set containing the range over which a geometric
    parameter $g \in G$ can be adjusted.
 2: Output: $P$ the set of predicted combinations of geometric
    parameter adjustments for a logically correct layout.
 3: {Let $S_{g,i}$ index a set of $n$ pairs $(H_i^c, H_i^{\bar{c}})$ for each input
    test cases $i \in I$ where $n = |S_g|$}
 4: {Let $S_{0,i}$ index the pairs $(H_i^c, H_i^{\bar{c}})$ for each input test case
    $i \in I$ from the initial layout simulation}
 5: $Q \leftarrow \emptyset$ {$Q$: set of slopes from regression analysis}
 6: $B \leftarrow \emptyset$ {$B$: set of pairs $(H_i^c, H_i^{\bar{c}})$ for each input test case
    $i \in I$ to be input to algorithm 1}
 7: $P \leftarrow \emptyset$
 8: for all $g \in G$ do
 9:    for all test cases $i \in I$ do
10:       $Q_{g,i}(H_i^c) \leftarrow$ leastSquares($S_{g,i}(H_i^c), n$)
11:       $Q_{g,i}(H_i^{\bar{c}}) \leftarrow$ leastSquares($S_{g,i}(H_i^{\bar{c}}), n$)
12:    end for
13: end for
14: for each $g \in G$ do
15:    select a $d \in R_g$
16:    $K_g \leftarrow d$ {K: a set of geometric parameters to test}
17:    $x := |G|$
18:    for all test cases $i$ in $I$ do
19:       $B(H_i^c) \leftarrow (\sum_{g=1}^{x} Q_{g,i}(H_i^c) \cdot K_g) + S_{0,i}(H_i^c)$
20:       $B(H_i^{\bar{c}}) \leftarrow (\sum_{g=1}^{x} Q_{g,i}(H_i^{\bar{c}}) \cdot K_g) + S_{0,i}(H_i^{\bar{c}})$
21:    end for
22:    if $isLogCorr(B)$ then
23:       {isLogCorr is algorithm 1}
24:       $P \leftarrow K$ {K's adjustments predict correct design}
25:    end if
26: end for
```

is then input to Algorithm 1 to determine if the parameter adjustments predict a functional layout. If the predicted H_i^c and $H_i^{\bar{c}}$ values in B constitute a functional layout, the set of combinations K is added to the set of candidate geometric adjustments P (line 24). Once all possible combinations have been considered, the algorithm terminates and returns P.

From the output of Algorithm 2, the designer then selects a set of adjustments and simulates the new layout to verify its correctness. If the new layout is determined to be non-functional, the designer can either select another set of parameter adjustments or sample additional geometric parameters (Step 6). Finally, if no functional layout is found and the design space has been sufficiently sampled, the designer should select the "closest-to-functional" set of parameter adjustments found or redesign the layout (Step 9) and repeat the design process from step 4.

IV. Conclusion

In this work, we have proposed an efficient systematic process for the design of functional 3D pNML layouts. Our primary goal was to find a functional layout by sampling and simulating a fraction of the design space. To this end, we leverage a machine learning-inspired prediction technique (Algorithm 2) that, through regression analysis, quantifies the effects of individual geometric parameters on a target output magnet. Our predictor then searches all possible combinations of parameter adjustments and outputs those found to be logically correct. Furthermore, we developed an algorithm for determining the logical correctness of both combinational and sequential logic layouts (Algorithm 1). Future work involves designing pNML layouts for target applications and testing our designs through fabrication.

References

[1] A. Imre *et al.*, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.

[2] M. T. Niemier *et al.*, "Nanomagnet logic: progress toward system-level integration," *J. of Physics: Cond. Mat.*, vol. 23, no. 49, p. 493202, 2011.

[3] I. Eichwald *et al.*, "Nanomagnetic logic: Error-free, directed signal transmission by an inverter chain," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 4332–4335, 2012.

[4] S. Breitkreutz *et al.*, "Nanomagnetic logic: Demonstration of directed signal flow for field-coupled computing devices," in *European Solid-State Device Research Conference*, 2011, pp. 323–326.

[5] S. Breitkreutz *et al.*, "Controlled reversal of co/pt dots for nanomagnetic logic applications," *Journal of Applied Physics*, vol. 111, no. 7, 2012.

[6] S. Breitkreutz *et al.*, "Majority gate for nanomagnetic logic with perpendicular magnetic anisotropy," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 4336–4339, 2012.

[7] S. Breitkreutz *et al.*, "Experimental demonstration of a 1-bit full adder in perpendicular nanomagnetic logic," *IEEE Transactions on Magnetics*, vol. 49, no. 7, pp. 4464–4467, 2013.

[8] I. Eichwald *et al.*, "Towards a signal crossing in double-layer nanomagnetic logic," *IEEE T. on Mag.*, vol. 49, no. 7, pp. 4468–4471, 2013.

[9] I. Eichwald *et al.*, "Majority logic gate for 3d magnetic computing," *Nanotechnology*, vol. 25, no. 33, p. 335202, 2014.

[10] R. Perricone *et al.*, "Design of 3d nanomagnetic logic circuits: a full-adder case study," in *Design, Automation Test in Europe Conference Exhibition*, March 2014.

[11] M. J. Donahue *et al.*, *OOMMF User's guide, Version 1.0, Interagency Report NISTIR 6367*.

[12] L. Deng *et al.*, "Accurate models for estimating area and power of fpga implementations," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 1417–1420.

[13] M. Zuluaga *et al.*, ""smart" design space sampling to predict pareto-optimal solutions," in *International Conference on Languages, Compilers, Tools and Theory for Embedded Systems*, 2012, pp. 119–128.

[14] M. Becherer *et al.*, "Magnetic ordering of focused-ion-beam structured cobalt-platinum dots for field-coupled computing," *IEEE Transactions on Nanotechnology*, vol. 7, no. 3, pp. 316–320, 2008.

[15] X. Ju *et al.*, "Nanomagnet logic from partially irradiated co/pt nanomagnets," *IEEE T. on Nano.*, vol. 11, no. 1, pp. 97–104, 2012.

[16] M. Niemier *et al.*, "Systolic architectures and applications for nanomagnet logic," in *IEEE Silicon Nanoelectronics Work.*, 2012, pp. 1–2.

[17] S. Breitkreutz *et al.*, "1-bit full adder in perpendicular nanomagnetic logic using a novel 5-input majority gate," *Joint European Magnetic Symposium, available: http://www.jems2013.org/usb/*, August 2013.

[18] M. Becherer *et al.*, "Towards on-chip clocking of perpendicular nanomagnetic logic," *Solid-State Electronics*, 2014.

[19] X. Ju *et al.*, "Systolic pattern matching hardware with out-of-plane nanomagnet logic devices," *IEEE T. on Nano.*, vol. 12, no. 3, pp. 399–407, 2013.

[20] M. S. Fashami *et al.*, "Magnetization dynamics, bennett clocking and associated energy dissipation in multiferroic logic," *Nanotechnology*, vol. 22, no. 15, p. 155201, 2011.

[21] Y.-H. Chu *et al.*, "Electric-field control of local ferromagnetism using a magnetoelectric multiferroic," *Nature materials*, vol. 7, no. 6, pp. 478–482, 2008.

[22] D. Bhowmik *et al.*, "Spin hall effect clocking of nanomagnetic logic without a magnetic field," *Nature nanotechnology*, 2013.

[23] S. Breitkreutz *et al.*, "Nanomagnetic logic: compact modeling of field-coupled computing devices for system investigations," *Journal of Computational Electronics*, vol. 10, no. 4, pp. 352–359, 2011.

[24] K. T. Cheng and A. S. Krishnakumar, "Automatic functional test generation using the extended finite state machine model," in *Proceedings of the 30th International Design Automation Conference*, 1993, pp. 86–91.

[25] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines-a survey," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1090–1123, August 1996.