The Dissertation Committee for Jaeyoung Park
certifies that this is the approved version of the following dissertation:

Probabilistic Design for Emerging Memory and Nanometer-Scale Logic

Committee:

Michael Orshansky, Supervisor

Mattan Erez

Jacob Abraham

Zhigang (David) Pan

Chankyung Kim

Probabilistic Design for Emerging Memory and Nanometer-Scale Logic

by

Jaeyoung Park

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2018

Dedicated to my family

Acknowledgments

Most importantly, I would like to express my very special gratitude to my supervisor, Prof. Michael Orshansky. Without his continuous support and insightful guidance, I would have never accomplished my research results. He has inspired me with enthusiasm for research. Also, his generosity kept me moving forward. I would also like to thank my committee members, Prof. Mattan Erez, Prof. David Z. Pan, Prof. Jacob Abraham, and Dr. Chankyung Kim, for their valuable advice and discussions. I have been honored to work with and thankful to my fellow students: Kareem Ragab, Meng Li, Mukund Kalyanaraman, Xiaodan Xi, and Ye Wang. Special thanks to Mike Thomson and Tianhao Zheng, for their collaboration on my research. I am greatly indebted to my parents for their selfless support and care. Last but by no means least, I am heartily thankful for having my wife, Seri An, in my life. I owe my deepest gratitude to her whose warm encouragement enabled me to go on with my research. I would also like to thank God.

Probabilistic Design for Emerging Memory and Nanometer-Scale Logic

Jaeyoung Park, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Michael Orshansky

As semiconductor technology has scaled down, the impact of stochastic behavior in very large scale integrated circuits (VLSI) has become an ever-more important concern. This dissertation investigates two distinct classes of problems that require the use of probabilistic methods and models: (1) Modeling and exploiting stochastic behavior in advanced memory technologies; (2) Probabilistic modeling of faults due to on-chip voltage variation.

This dissertation first investigates the unique physics-level stochasticity of spin-transfer torque magnetic RAM (STT-RAM). The write process of STT-RAM is stochastic: specifically, the write time of a bitcell varies significantly. The worst-case approach, which uses the longest write pulse duration, guarantees a successful write; however, it introduces significant energy overhead due to excessive margins since the average write pulse duration is far shorter than the worst-case pulse duration. This dissertation develops novel circuit techniques to exploit the stochastic properties of STT-RAM write operation for energy savings by moving away from

the worst-case approach to dynamic strategies while maintaining the required low error rate. The first contribution is a variable energy write (VEW) architecture that effectively exploits the wide distribution of write time to greatly reduce energy via a mechanism that checks the instantaneous state of the bitcell and deactivates the write current once the correct value has registered. The second contribution is a multiple attempt write (MAW) strategy that utilizes the asymptotic temporal stochastic independence of repeated switching events to achieve a dramatic reduction in energy. The proposed architectures are evaluated using a compact STT-RAM cell model. Analysis indicates that VEW succeeded in reducing the write energy by 94.7% with approximately 1% relative area overhead under an efficient design methodology compared with the conventional designs relying on the worst case approach. MAW reduced the overall write energy by 94.6% with approximately 0.05% relative area overhead.

This dissertation then addresses the problem of probabilistic modeling of faults due to on-chip voltage variations. The power supply voltage variation can increase gate delay, resulting in timing faults on near-critical paths. These low-level faults ultimately propagate to architecture and application levels, often leading to critical system failures. Developing an accurate fault model and injection tool that generates and propagates faults from circuit- to gate-level is important for accurately predicting the resulting system failures. This is challenging since the model needs to accurately capture the physical characteristics at the circuit level that define the likelihood of a fault and use that information to guide the injection with the proper probability. At the same time, the analysis and fault injections need to

be computationally manageable to allow analysis of realistic systems under realistic workloads. The conventional fault models rely on either Monte Carlo sampling or time-consuming runtime simulation using the worst-case voltage drop. To overcome simulation overheads of runtime circuit-level simulation, a novel two-phase approach is proposed. The main idea is that circuit characterization can be done before simulation. The result of pre-characterization is used at runtime via a form of look-up to enable gate-level efficiency. The two-phase methodology is time-efficient but may require high memory unless the look-up tables are carefully optimized. This dissertation also develops the fault probability estimation based on workload-specific voltage distribution, rather than a fixed worst-case voltage. The proposed methodology is implemented on an OpenSPARC design targeting on a $32\,\mathrm{nm}$ technology node. Analysis indicates the proposed fault modeling and injection flow reduces runtime overhead by $24\times$ compared to the previously best-known gate-level fault simulator while having circuit level accuracy.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

For several decades, Moore's Law has dictated the scaling of transistor dimensions as well as the level of the voltage at which integrated circuits operate [12]. According to the International Technology Roadmap for Semiconductors (ITRS), by 2019, the minimum gate length will be $5\,\mathrm{nm}$ and the supply voltage (Vdd) will be around 0.77V [13]. Scaling has led to significant benefits in terms of speed improvement and reduced power consumption in VLSI designs. Scaling has also brought a number of challenges. Smaller transistors are more vulnerable to variability of nanometer technology. At lower operating voltages, the systems are more sensitive to fluctuations of the power supply voltage. New memory devices exhibit entirely new physics-level stochasticity that makes older design principles obsolete.

This dissertation focuses on two areas which call for the development of modeling and design techniques based on probabilistic principles. The first topic concerns the design principles that exploit stochastic behavior in spin-transfer torque magnetic RAM (STT-RAM) memory technologies. The second thrust of this dissertation is on probabilistic modeling of faults arising due to on-chip voltage variation. This chapter provides a brief introduction to the problems addressed by this dissertation.

1.1 Design Methods for Addressing the Stochasticity of STT-RAM

Novel memory device technologies such as STT-RAM hold significant promise in providing new capabilities and opportunities for low power systems. STT-RAM has been touted as a candidate for universal memory technology that may be able to provide integration density close to DRAM, the non-volatility of Flash memory, fast read speed close to SRAM, and practically zero standby power. However, serious challenges remain in terms of reliable and high-yielding commercial manufacturing of large systems utilizing STT-RAM technology [14]. The STT-RAM write process is inherently stochastic and the actual time to complete a write varies dramatically, with the distribution having a very long tail. This stochasticity of switching time is temporal, leading to variation in transition time even for a single cell. As a result, conservatively guaranteeing a reliable write requires maintaining the write current for a duration much longer than that required for an average write to complete. In other words, the worst-case approach that uses the longest pulse duration guarantees a successful write; however, it introduces a significant energy penalty since the average write duration is far shorter than the worst case duration.

1.1.1 Variable Energy Write STT-RAM Architecture

The first technique that this dissertation develops is a design approach that exploits write stochasticity to significantly reduce the write energy in STT-RAM. With the proposed *variable-energy writes* (VEWs), the write current of each indi-

vidual cell is terminated once that cell's state matches its desired write value. This is in contrast to the conventional approach, which fixes the write duration of all writes to match the expected worst-case delay for a given level of reliability. Since STT-RAM has a wide distribution of write time, the difference between a write pulse needed to achieve the typically required error rate, 1.5×10^{-7} [15] and the mean pulse duration is almost $20\times$. This means that substantial energy savings could be achieved for a large fraction of switching events if the current pulse is terminated once the specific switching occurred. These energy savings can be achieved by moving away from the traditional worst-case approach towards a technique that detects the write completion of each bit and turns off the write current. To realize variable-energy writes, a circuit is developed that continuously monitors the state of each STT-RAM cell and disables its write current when it senses the desired state has been achieved.

Due to the presence of significant process variations and low tunnel magneto-resistance (TMR) characteristic, the VEW circuit can be sub-optimal. This is because the bitcell of STT-RAM needs to be designed to have the minimum area to achieve the maximum array density. The minimum-size cell has maximal parameter variability, and the resulting variation in the parameters of both the MOSFET and the MTJ greatly affects the functionality of the memory cell. More importantly, the low TMR of the MTJ, e.g. of about 70%, yields insufficient read margin of the initial design described above. To address this limitation, a variation-tolerant version of VEWs peripheral circuitry is developed. The main idea of the proposed *variation-tolerant VEW* (VT-VEW) is to detect voltage transitions rather than to

sense a specific voltage level. This avoids the need to detect the absolute resistance (voltage) which is changed by the process variations.

The advantages of the proposed VEW and VT-VEW architectures are significant: they are predicted to reduce write energy by 94.7% and 71.6% compared with the conventional designs relying on the worst case approach.

1.1.2 Multiple Attempt Write STT-RAM Architecture

While VEW and VT-VEW require relatively small area overhead if column-multiplexing is considered, they still require circuitry for monitoring and terminating operations. A different approach is developed based on a hypothesis that an MTJ can be written by a current pulse of some magnitude and duration over multiple attempts even when the likelihood of a successful write on each attempt is low. Write-verify-rewrite with adaptive period (WRAP) [8] and low-current probabilistic writes (LCPW) [16] exploit an iterative write algorithm for reducing write energy by lowering individual write currents. Lowering write current leads to a high error rate; however, the writing eventually succeeds with multiple trials.

The proposed *multiple attempt write (MAW)* architecture is also built on the principle of applying multiple write attempts to realize low-energy writes. The pulse duration is fixed at a value that results in an unacceptably high probability of error. Even though the probability of a write-fail on any individual attempt is high, the writing ultimately succeeds on the following attempts. This allows MAW to achieve a high overall write success rate while reducing the needed energy. This work makes several contributions. The first contribution is a self-validating write

circuit that allows bit-wise validation and deactivation with near-zero energy and area overhead. In contrast, WRAP and LCPW require a comparator for bit-wise validation and a sizable amount control logic for deactivation. The second contribution is the formalization of the notion of re-randomization time as the interval needed for sufficient randomness to develop such that subsequent write attempts become stochastically independent. The value of re-randomization time is quantified via accurate numerical simulation.

Compared to WRAP and LCPW, the write energy is reduced by $2.1\times$ because the proposed architecture does not introduce energy overheads of validation and deactivation.

1.2 Efficient Fault Injection Flow

Power supply variation is one of the major sources of hardware faults. Developing a fault model and injection tool that generates and propagates faults from circuit- to gate-level is important for accurately predicting an resulting system failures. This is challenging because the model needs to accurately capture the physical characteristics at the circuit level that define the likelihood of a fault and use that information to guide the injection with the proper probability. At the same time, the analysis and fault injections need to be computationally manageable to allow analysis of realistic systems under realistic workloads. The conventional fault models rely on either Monte Carlo sampling or time-consuming runtime simulation using the worst-case voltage drop.

This dissertation presents a hybrid gate-level fault simulator capable of cap-

turing fault correlations through circuit-level analysis. The gate-level simulator is used to check fault propagation while on-demand circuit-level analysis is used for fault generation. To overcome simulation overhead of runtime circuit-level analysis, a novel two-phase approach is proposed. The main idea is that circuit characterization can be done before simulation. The result of pre-characterization is used at runtime via a form of look-up to enable gate-level efficiency. The two-phase methodology is time-efficient, but may require high memory unless the look-up tables are carefully optimized.

This dissertation develops an efficient characterization methodology enabling the two-phase flow. It is based on jointly capturing and compactly storing a set that includes a fault-inducing input, the impacted outputs, and the probabilities of the corresponding fault-events. This is enabled by a novel combination of static timing analysis (for finding near-critical paths) and an automatic test-pattern generation (ATPG) tool to check sensitization conditions. This dissertation also develops the fault probability estimation based on workload-specific voltage distribution, rather than a fixed worst-case voltage.

The contributions of this work are: (1) an efficient fault modeling and injection flow that enables on-demand transistor-accurate fault injection with workload-specific voltage distribution to allow gate-level reliability analysis; (2) a two-phase approach comprised of a pre-characterization and a runtime fault injection that enables accuracy and efficiency; (3) an efficient pre-characterization methodology to jointly capture and compactly store a set that includes a fault-inducing input, the impacted outputs, and the probabilities of the corresponding fault-event.

Chapter 2

Variable-Energy Write STT-RAM

This chapter[1] addresses the modeling of stochasticity of the STT-RAM write and develops a variable-energy write (VEW) architecture to handle the stochasticity for reducing write energy while maintaining low error rate.

2.1 Stochastic Switching of STT-RAM

STT-RAM holds significant promise in providing new capabilities and opportunities for low power systems. STT-RAM has been touted as a candidate for universal memory technology that may be able to provide integration density close to DRAM, the non-volatility of flash memory, fast read speed close to SRAM, and practically zero standby power [17–19]. As shown in Fig. 2.1, STT-RAM has higher endurance compared to Flash, Ferroelectric RAM (FeRAM) and Phase

[1]This chapter is an extended work of publications: **Jaeyoung Park**, Tianhao Zheng, Mattan Erez, and Michael Orshansky, "Variation-Tolerant Write Completion Circuit for Variable-Energy Write STT-RAM Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 99, pp. 360 - 367, 2015. Jaeyoung Park's contribution of this paper is a circuit design and energy analysis of the proposed technique.

Tianhao Zheng, **Jaeyoung Park**, Michael Orshansky, and Mattan Erez, "Variable-Energy Write STTRAM Architecture with Bit-Wise Write-Completion Monitoring," *in Proc. of IEEE International Symposium on Low Power Electronics and Design*, pp. 229 - 234, 2013. Jaeyoung Park's contribution of this paper is a circuit design of the proposed technique.

change memory (PRAM), and STT-RAM write time is much faster than that of flash memory. In addition, the cell size is small and the read latency is tenths of nano seconds, which is comparable to DRAM and PRAM. According to the International Technology Roadmap for Semiconductors issued, STT-RAM has the potential to be the next-generation memory device. At the $16\,\mathrm{nm}$ technology node, STT-RAM cell designs with very competitive characteristics – an area of $20F^2$, read time of 5ns, and write time of 35ns – are feasible [20]. Yet, serious challenges remain in terms of reliable and high-yielding commercial manufacturing of large systems utilizing STT-RAM technology. One of the dominant challenges for the adoption of STT-RAM is the fundamental stochasticity of write operations. The STT write process is inherently stochastic and the actual time to complete a write varies dramatically, with the distribution having a very long tail. This stochasticity of switching time is temporal, leading to variation in transition time even for a single cell. As a result, conservatively guaranteeing a reliable write requires maintaining the write current for a duration much longer than that required for an average write to complete. In other words, the worst-case approach, which uses the longest pulse duration, guarantees a successful write; however, it introduces a significant energy penalty since the average write duration is far shorter than the worst case duration.

This chapter addresses the details of the stochasticity of the STT-RAM write and a potential of a probabilistic design method that exploits write stochasticity to significantly reduce the write energy in the STT-RAM design.

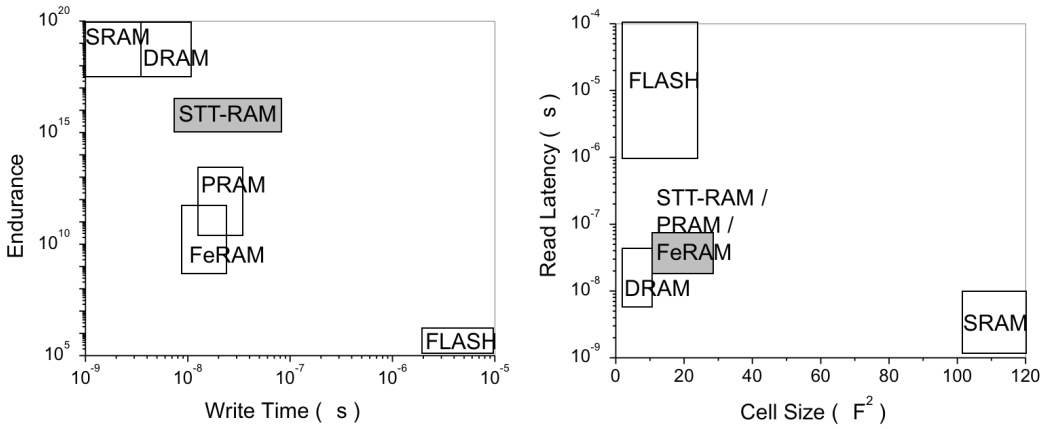


Figure 2.1: Performance comparison of current memory technologies.

2.1.1 Fundamentals of STT-RAM

An STT-RAM's bitcell is comprised of 1 MOSFET and 1 magnetic tunnel junction (MTJ) in the conventional 1T-1MTJ configuration. The MTJ is a storage component and an access transistor connected to the MTJ is used to control its operation as shown in Fig. 2.2 [9–11]. Write '0' and write '1' operations proceed by turning on the access transistor and injecting a high write current in one of two directions from source line (SL) to bitline (BL) or vice versa.

The MTJ consists of two layers of ferromagnetic material separated by a dielectric layer. The two magnetic layers have their own spin directions, with one layer pinned to a fixed polarization and the second layer being free. The spin of the free layer (FL) can be switched from one orientation to its opposite by apply-

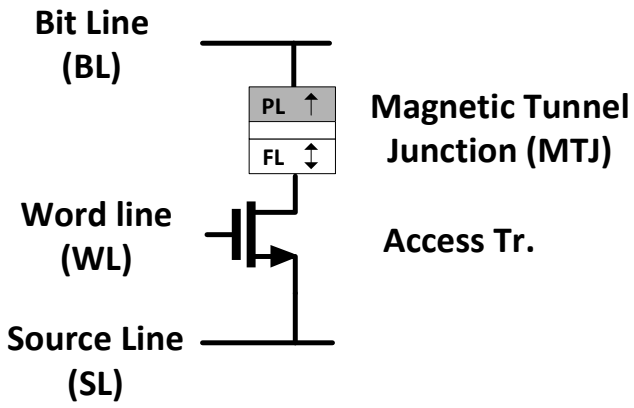


Figure 2.2: 1T-1MTJ STT-RAM's bitcell configuration.

ing a current pulse through the MTJ while the spin is set to one orientation in the pinned layer (PL). Fig. 2.3 shows a common device design that has a CoFeB-MgO interface. CoFeB is a ferromagnetic material and MgO is used as a dielectric layer. The MTJ can be in one of two states, *anti-parallel* (*AP*) and *parallel* (*P*). In the AP state the free and pinned layers have opposite spin directions and in the P state both have the same spin direction. Each of these two states exhibits a distinct resistance corresponding to storing a binary '0' or '1' (e.g., *P* and *AP*).

2.1.2 Switching Model

The change of MTJ state occurs when the current passing through the junction exceeds a certain minimum magnitude and is maintained for sufficient time. The process of MTJ write is a process of alignment of the magnetic orientation of the regions of the ferromagnetic layer. There are two distinct physical mechanisms

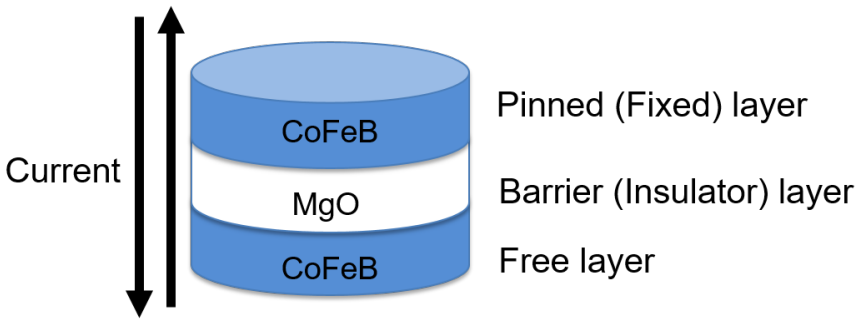


Figure 2.3: Block diagram of a perpendicular MTJ.

that govern MTJ switching, which depend on the magnitude of injected current: thermally activated switching and fast precessional switching. The thermally activated switching regime holds for currents at or below a certain *critical current* (I_{C0}), defined at zero Kelvin. The thermally activated switching process is relatively slow, with mean switching times of several nanoseconds to tens of nanoseconds, and is also highly stochastic. The average current required for switching (I_C^{therm}) depends on the write pulse duration (T_{wr}). The following deterministic model is often used to describe this relationship, despite the inherent stochasticity of the write process [20]:

$$I_C^{therm}\left(T_{wr}\right) = I_{C0}\left\{1 - \frac{1}{\Delta}\ln\left(\frac{T_{wr}}{\tau_0}\right)\right\} \tag{2.1}$$

where Δ is the thermal stability factor and τ_0 is the inverse of the thermal attempt frequency (with a typical value of $1\,\mathrm{ns}$ [20, 21]).

The second mechanism, fast precessional switching, is very rapid, typically occurring within $1\,\mathrm{ns}$, and shows less stochasticity. However, activating this switching process requires a current that is much larger than I_{C0}. The average current required in fast precessional switching also depends on the write pulse duration and can be described as:

$$I_C^{prec}\left(T_{wr}\right) = I_{C0} + \frac{C\ln\left(\pi/2\theta\right)}{T_{wr}} \tag{2.2}$$

where C and θ represent the relaxation time and initial angle between the free layer and reference layer, respectively [20].

2.1.3 Stochastic Switching Model

The time needed for the MTJ to switch is stochastic and the distribution is quite wide [21, 22]. The wider part of the distribution is dominated by the thermally activated switching [23]. The above-mentioned deterministic model has an essential limitation to consider the stochasticity in the switching process and describe the mean switching behavior rather than the entire distribution. A probability model for the thermally activated switching duration has been proposed [21]. The model describes the switching probability $p_{sw}\left(t, I\right)$, which is the probability of switching occurring for a pulse duration t at current I:

$$p_{sw}(t, I) = 1 - exp\left\{ -\frac{t}{\tau_0} exp\left[-\Delta\left(1 - \frac{I}{I_{C0}}\right)\right]\right\} \tag{2.3}$$

where Δ is the thermal stability factor, τ_0 is the inverse of the thermal attempt frequency, and I_{C0} is the critical current at zero Kelvin.

In the fast precessional switching regime, the stochasticity of switching time is lower. The ratio of the switching time standard deviation to its mean is in the range from 0.2 to 1 [22]. The same ratio has the value of 0.08 for moderately wide pulses in the fast precessional switching regime [21]. Unfortunately, in this regime, the exact closed-form model for the switching time probability distribution is not available. However, empirical measurements suggest that the form of the distribution is asymmetric Gaussian [22]. Therefore, this asymmetric Gaussian model is adopted for experiments in the case of asymmetric writes where the current through the MTJ is $2.06I_{C0}$, in our design (Fig. 2.4).

Importantly, in the thermal switching regime (e.g., $I = 0.95I_{C0}$), the model predicts that the pulse duration required to reach a high write success probability is much longer than the pulse duration corresponding to a switching probability of 50%. Fig. 2.4 shows the width of the distribution by plotting $1 - P_{sw}$, which can be thought of as the write error probability. To include the above stochastic behavior, a model of [1] which is the first model that can accurately predict the stochasticity of MTJ switching within a circuit simulator is adopted.

2.2 Variable-Energy Writes

Because of a wide distribution of the switching, the traditional approach which uses the pulse that guarantees writes for all cells wastes energy since the

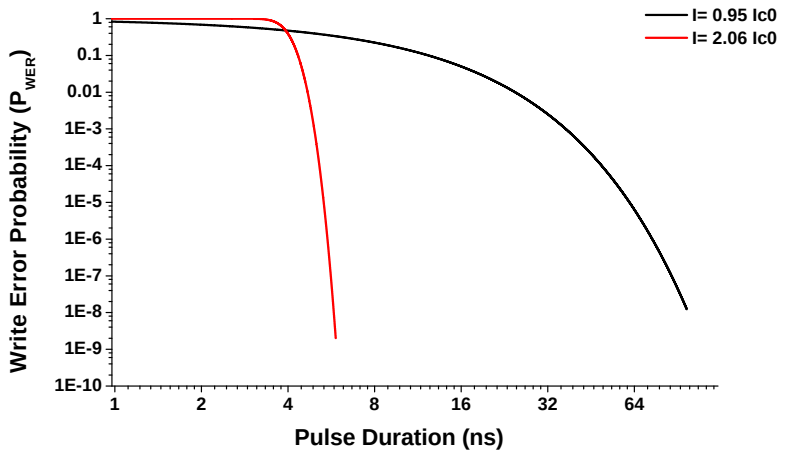


Figure 2.4: Probabilistic model of write error probability as a function of write pulse duration; shown for several currents.

average write time is far shorter than the worst-case pulse duration. This means that the excess margining from the worst-case approach is not efficient and show a potential gain of a probabilistic design method that exploits the write stochasticity. The variable energy write (VEW) architecture is developed as an effective way to exploit the stochasticity of write time. Write energy is reduced by utilizing a mechanism that: (1) monitors the instantaneous state (resistance) of the MTJ, and (2) deactivates the write current once the correct value being written has registered, as shown in Fig. 2.5. The monitoring circuit compares the MTJ resistance to a reference value and can thus determine the stored value and detect the change of resistance on a switch. The shutdown circuit terminates the write current once the correct value is registered. In addition to saving energy when the write switches the

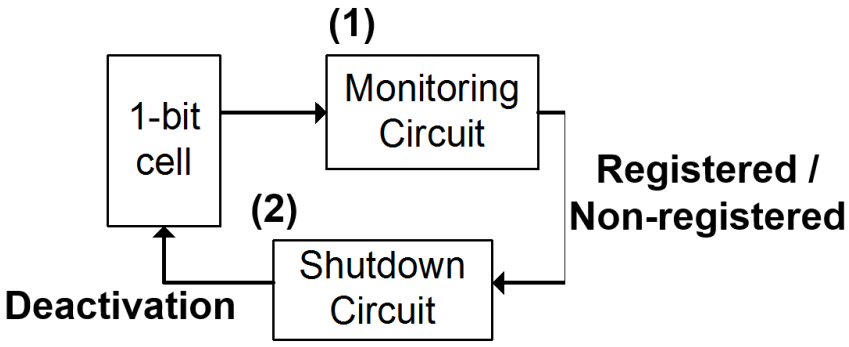


Figure 2.5: Block diagram of the proposed variable energy write architecture.

MTJ state, this design inherently shuts down the write current when the value being written equals the value already stored.

2.2.1 Circuit Design of Variable Energy Write (VEW)

Fig. 2.6 shows a schematic view of the proposed circuit comprised of the monitoring sub-circuit and the shutdown sub-circuit. The monitoring sub-circuit, whose schematic is further expanded in an inset, tracks the bitline voltage at node SW_M. Depending on the direction of the write, the voltages corresponding to the *AP* and *P* states are significantly different. Therefore, to detect both transitions, the monitoring sub-circuit contains two comparators, whose thresholds are set to 680 mV for $P \rightarrow AP$ and to 390 mV for $AP \rightarrow P$. A CMOS inverter is used as a comparator. The switching threshold voltage of the inverter can be set to the mid-point between the *P* and *AP* write voltages by properly selecting the PMOS/NMOS sizing ratio.

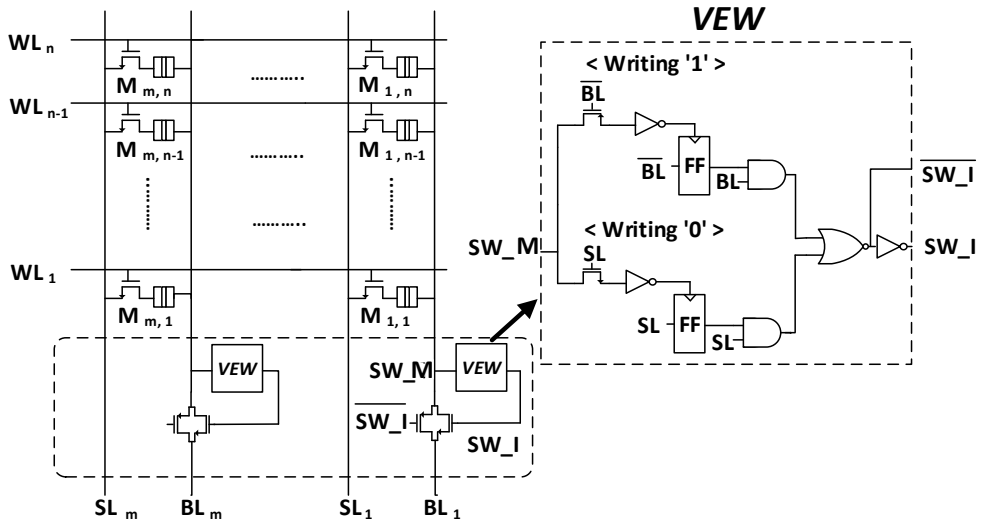


Figure 2.6: Block diagram of a memory array with the proposed VEW circuit.

Table 2.1: Key parameters of $17\,\mathrm{nm}$ MTJ [1–3].

Term	Definition	Value	Unit
e	Electron charge	1.6E-19	C
$\hbar$	Reduced Planck constant	1.05E-34	Js
α	Magnetic damping constant	0.027	
η	Thickness of the oxide barrier	1.3	*nm*
t_F	Thickness of the free layer	0.9	*nm*
W	Width of MTJ	40.0	*nm*
L	Length of MTJ	17.0	*nm*
H_d	Out-of-plane magnetic anisotropy	1.3	*T*
P	Percentage of tunnel current	0.56	
I_{c0}	Critical current at zero Kelvin	69.4	μA
Δ	Thermal stability factor	34	

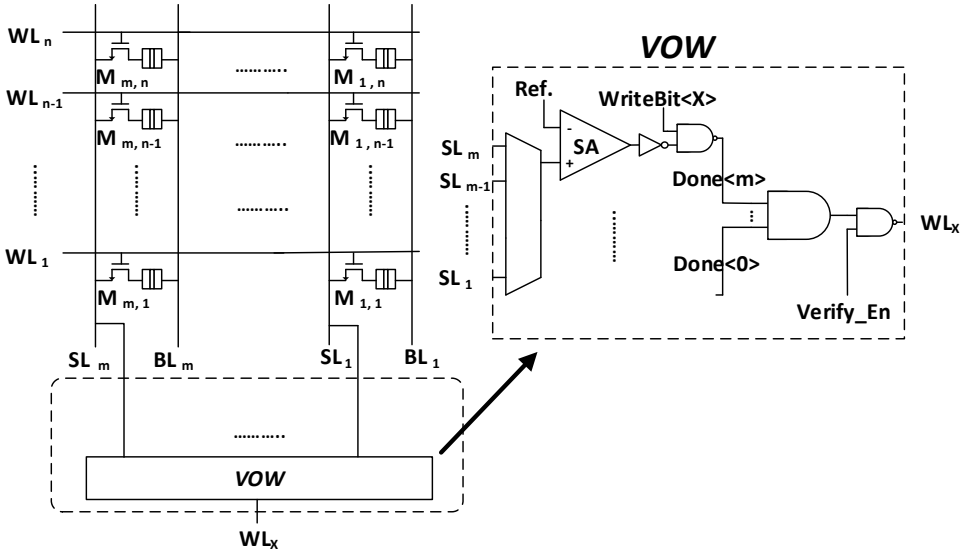


Figure 2.7: Block diagram of a memory array with VOW [8].

Consider a $P \rightarrow AP$ transition. At the start of the write operation the voltage is raised on the word line (WL) and the bitline (BL), and lowered on the source line (SL). The access transistor turns on and the current flows through the MTJ and the access transistor. If the current supplied by the NMOS is sufficiently large, the MTJ undergoes a state change. A state change modifies the MTJ resistance, which leads to a voltage change at the output node of the access transistor. The voltage change at node SW_M is detected by the comparator (the tuned inverter), which then turns the shutdown transmission gate off, terminating the pulse. The AND gate, which is used in the shutdown circuit, ensures that this monitoring circuit is only active when writing a logical '1'. Because we want to monitor transitions in both directions, an additional comparator needs must be used when writing a '0'.

This is because the absolute voltages at SW_M are lower when the current direction is reversed (SL to BL). Thus, two comparators are needed for each column. The flip-flop (FF) in the VEW circuit works as a delay element to prevent an unwanted feedback loop between the monitoring and reset circuits.

In an $AP \rightarrow P$ transition, the voltage is raised on the SL and lowered on the BL. Because the bitcell is not symmetric, the symmetric write current is only enabled if the WL voltage is boosted (e.g., $V_{WL} >$ Vdd). A state change occurs if the current is sufficiently large, leading to a voltage change at the output node of the access transistor. The voltage change is ultimately detected by the comparator, which then turns the shutdown transmission gate off.

The area overhead of the above design is 2 inverters, 2 AND gates, 1 OR gate per column, and 2 flip-flops. We estimate the area to be $1860F^2$ per column based on equivalent scaling estimation method [13, 24]. Note that the proposed write completion circuit controls each bit independently which maximizes write efficiency. This is in contrast to the VOW architecture of Bi et al. [8], which terminates writes at the granularity of an entire word (Fig. 2.7).

The proposed circuit is designed using the $16\,\mathrm{nm}$ Predictive Technology Model (PTM) MOSFET model and a compact MTJ model [1–3]. The MTJ parameters are derived from the $17\,\mathrm{nm}$ MTJ manufactured by Samsung [2], as detailed in Table 2.1. Simulation results of the proposed circuit are shown in Fig. 2.8, for a $P \rightarrow AP$ transition. While the conventional 1-bit cell without the write completion circuit consumes substantial power until the end of the clock period, the 1-bit cell with the VEW circuit minimizes power after the switching occurs at $2.8\,\mathrm{ns}$.

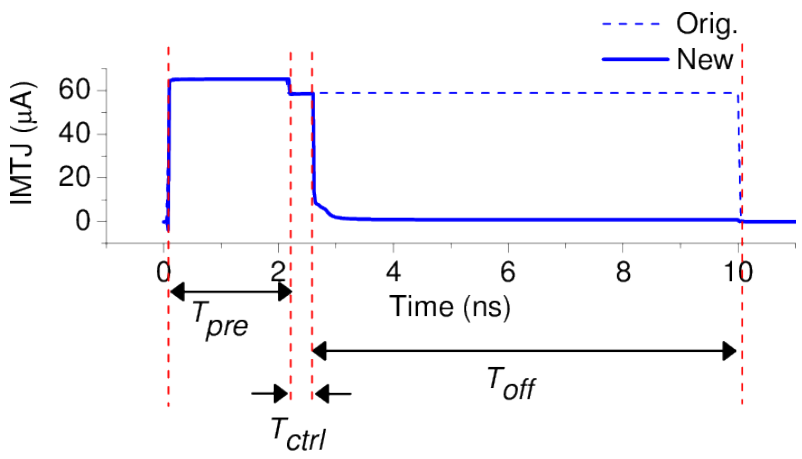


Figure 2.8: Simulated current of a $P \rightarrow AP$ transition for the conventional (original) and VEW techniques.

Fig. 2.9 shows simulation results for writing a '0' when a '0' is already stored in a cell ($P \rightarrow P$). The current through the MTJ (IMTJ) for the baseline circuit (dashed line) is kept high for the entire pulse duration. The MTJ current controlled by the proposed VEW circuit drops to zero within $1\,\mathrm{ns}$, which is the response time of the monitoring and control circuits. The circuit design is also validated in an array configuration. We use wire resistance and capacitance as specified by ITRS for a $16\,\mathrm{nm}$ process (resistivity: $22\,\mu\Omega$-cm; capacitance per unit length: 1.6pF/cm) [4]. The wire delay, simulated with Cadence Spectre, is less than 10ps.

2.2.2 Circuit Design of Variation-Tolerant VEW (VT-VEW)

The 1-bit cell needs to be designed to have the minimum area to achieve the maximum array density. The minimum-size cell has maximal parameter vari-

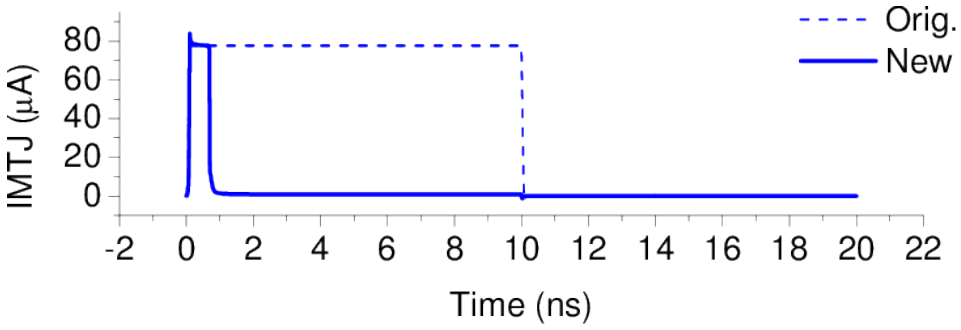


Figure 2.9: Simulation results of a 1-bit cell with the proposed write completion circuit writing '0' while already in the '0' state ($P \rightarrow P$).

ability, and the resulting variation in the parameters of both the MOSFET and the MTJ greatly affects the functionality of the memory cell. More importantly, the low tunnel magneto-resistance (TMR) of the MTJ, e.g. of about 70%, yields insufficient read margin the initial design described above. To address this limitation, a variation-tolerant version of VEW's peripheral circuitry is developed.

The reasons for the inadequate performance of the initial design under parameter variability is described first and circuit designs are then addressed. The sensing method that relies on the reference voltage performs well when the lower voltage (V_P) representing a specific state does not exceed the reference voltage under any circumstances. Similarly, the higher voltage (V_{AP}) representing the other state cannot be allowed to fall below the reference voltage. However, a statistical circuit simulation indicates that there is a substantial overlap between the V_{AP} and V_P voltages. This means that in many cases, a decision based on a fixed reference voltage will produce an error. This is illustrated in Fig. 2.10 that shows the distribu-

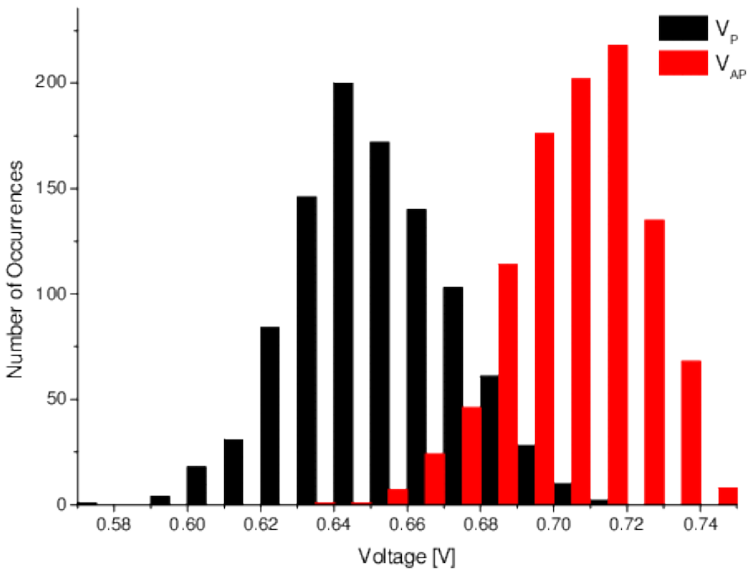


Figure 2.10: Distribution of voltage at node SW_M.

tion of state voltages as measured at the node, SW_M. The distribution is generated via a Monte Carlo simulation with the parameter-variability values collected from [4–6] and summarized in Table 2.2.

The main idea of the proposed variation-tolerant strategy is to detect voltage transitions rather than to sense a specific voltage level. This avoids the need to detect the absolute resistance (voltage). Reliance on transition-detection means that this circuit will only be able to reduce energy for a write operation that involves state transitions. Transition-detection can be achieved by monitoring the difference between the initial and the final voltages on a node and can be realized via a double-sampling strategy. The initial voltage set by one state (e.g., the P state) is stored

Table 2.2: Key parameters for corner simulations [4–6].

CMOS Parameters	Mean	$3\sigma/\mu$
Gate length	16 nm	9.6%
Oxide thickness	10Å	4.0%
Threshold voltage (N/P)	$392\,\mathrm{mV}$ / $398\,\mathrm{mV}$	22.8%
MTJ Parameters	**Mean**	$3\sigma/\mu$
Length	$17\,\mathrm{nm}$	9.0%
Thickness of free layer	$0.9\,\mathrm{nm}$	5.0%
Thickness of oxide barrier	$1.3\,\mathrm{nm}$	4.0%

Data from [4] , [5] , [6]

on a capacitor by a sample-and-hold (S/H) circuit at the start of the write operation. The initial voltage is then compared with the final voltage set by the other state (e.g., the AP state). The voltage difference is amplified by a differential amplifier, which then turns on an inverter. Finally, the 1-bit cell is terminated by an inverter.

The critical advantage of this method is that since the initial voltage of a cell is used as the reference, the difference between the initial and final voltages in any cell is given by the voltage determined by the value of TMR. The ability of this method to be insensitive to parameter variation can also be seen from the same perspective: since the impact of parameter variability is to modify both the initial and final voltages by the same amount, relying on double-sampling eliminates the impact of parameter variability entirely. Finally, because the typical time-scale over which the environmental parameters, namely temperature and supply voltage, exhibit variation is much larger than the switching time of an MTJ, the detrimental impact of these sources of variation is also eliminated.

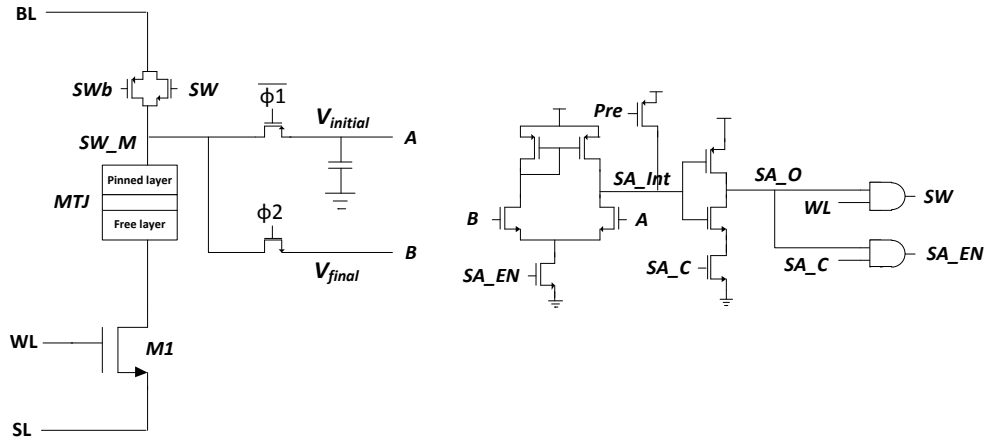


Figure 2.11: The schematic of the proposed variation-tolerant VEW circuit.

Fig. 2.11 shows a schematic view of the variation-tolerant implementation. This circuit is comprised of the monitoring sub-circuit and the shutdown sub-circuit. The monitoring circuit samples the bitline voltage at node SW_M and stores it on a capacitor. It continues tracking the voltage at node SW_M and compares it with the stored voltage on the capacitor using the differential amplifier. Note that the leakage-induced voltage change during the sampling phase is very small (several μV).

Consider the write '1' operation. In the nominal case, the P state produces the output voltage of 0.65V at node SW_M and this voltage is stored in the capacitor. The AP state produces the output voltage of 0.71V at node SW_M after switching. The voltage difference of $60\,\mathrm{mV}$ is amplified by the amplifier with a gain of 21dB. The output of the differential amplifier turns on an inverter. The voltage change

ultimately deactivates the 1-bit cell by terminating switches on the BL and turning off the differential amplifier via SA_EN.

Fig. 2.12 shows the waveforms at each node of the proposed circuit for a $P \rightarrow AP$ transition. The waveforms show the change of the current across the MTJ and the voltage at SW and SA_EN as the state of the MTJ changes. At the start of the write operation, the voltage is raised on the word line (WL) and the bitline (BL) and lowered on the source line (SL). The access transistor turns on, and the current flows through the MTJ and the access transistor. The voltage on SW_M, introduced by the current through the MTJ, is sampled and stored in a capacitor on node A. The voltage on SW_M is tracked on node B after the first sampling on node A. When the transition occurs, the voltage on SW_M and B change. The voltage change is detected by the differential amplifier that turns on an inverter. AND gates, which are used in the shutdown circuit, ensure that this monitoring circuit is only active in the evaluation phase. Eventually, the signals SW and SA_EN are lowered and the 1-bit cell and the monitoring sub-circuit are shut off.

To validate the functionality and energy savings of VT-VEW, a Monte-Carlo simulation experiment is performed with the stochastic model of the MTJ [1]. We use Cadence Spectre circuit simulator to simulate the energy consumption of the VT-VEW. Fig. 2.13 shows a symbol view of the MTJ model written in VerilogA in Cadence Spectre. The switching time varies to represent the stochastic behavior of an MTJ as shown in Fig. 2.14. Fig. 2.14 shows the waveforms on a 1-bit cell that represents the change of the current across an MTJ and the state change of the MTJ in the 1-bit cell.

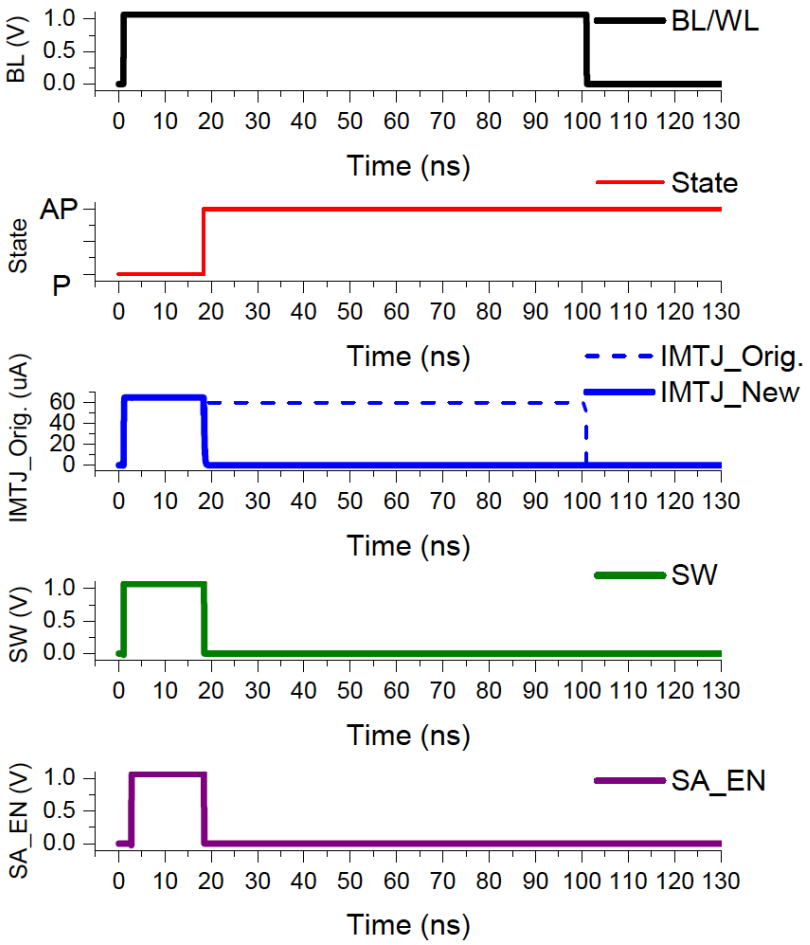


Figure 2.12: The waveforms for a $P \rightarrow AP$ transition.

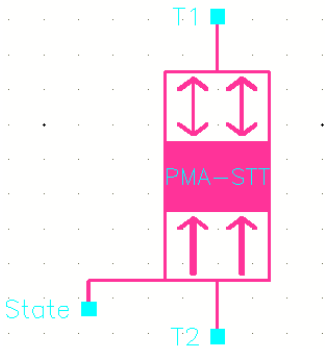


Figure 2.13: Symbol view of an MTJ model written in VeriloA in Cadence Spectre.

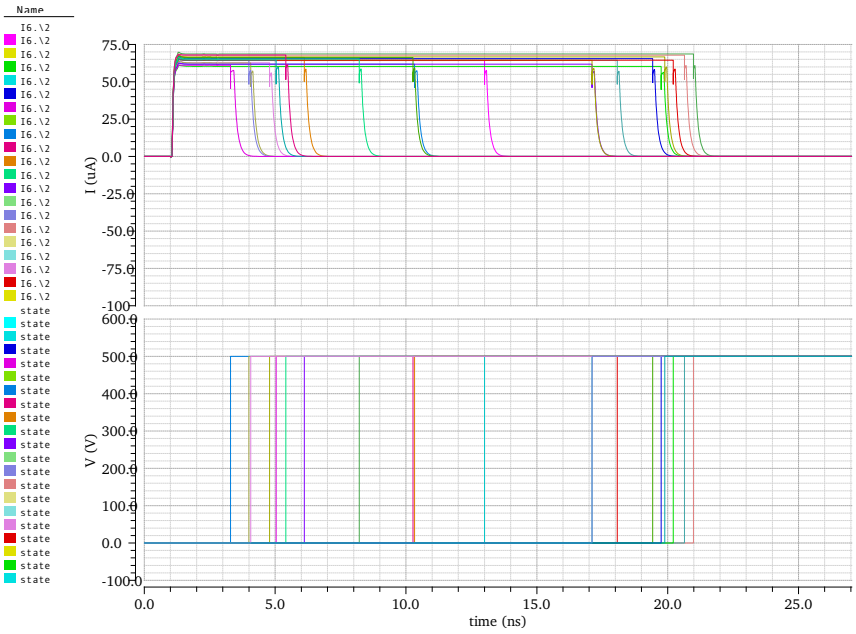


Figure 2.14: The waveforms of a 1-bit cell for a $P \rightarrow AP$ transition with a stochastic MTJ model using Cadence Spectre circuit simulator.

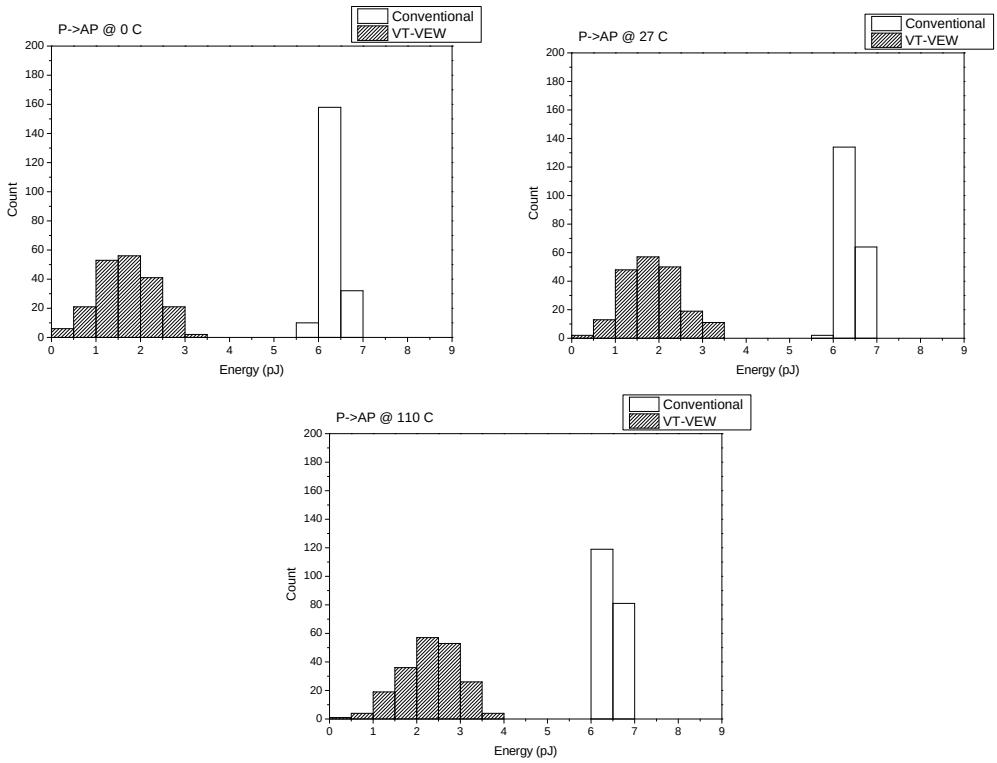


Figure 2.15: Energy of the conventional write with WL boosting and the VEW for a $P \rightarrow AP$ transition for several temperature values: $0\,^{\circ}\mathrm{C}$, $27\,^{\circ}\mathrm{C}$, and $110\,^{\circ}\mathrm{C}$.

Fig. 2.15 shows the energy consumption of a 1-bit cell for the conventional write with the energy consumption of the variation tolerant VEW circuit for a $P \rightarrow AP$ transition by Monte-Carlo simulation. Energy for the conventional write ranges from 5.5pJ to 7pJ. Energy variation is mainly due to process variations. It is caused by the fact that the applied pulse duration is always $100\,\mathrm{ns}$. VEW changes the applied pulse duration adaptively by changing the switching time of the MTJ. As a result, the energy of VEW shows a wide variation. When the switching in the MTJ occurs in the first few nanoseconds, the energy consumption is very low. In contrast, the energy is large in the extreme cases. On average, the energy saving is approximately 71% for a $P \rightarrow AP$ and is 77% for the $AP \rightarrow P$ transition. These results are summarized in Table 2.3. Overall, the energy saving is 73%.

The proposed variation-tolerant design is also tolerant to temperature variation. Fig. 2.15 shows energy consumption across the wide temperature range for the $P \rightarrow AP$ transition. Energy consumption of the conventional write ranges from 5.5pJ to 7pJ and energy consumption of the proposed variation tolerant VEW is below 4pJ in all cases. The summary of the average energy consumption values for a wide range of temperatures is shown in Table 2.3.

We compare VEW with the related techniques such as the verify-on-write (VOW) [8], early-write termination (EWT) [25], and the self-termination technique [26] in terms of energy savings. Note that the cell structure is changed for the self-termination technique requiring an additional word-line and resulting in increased array size [26]. VEW has an improvement of nearly a factor of 2-3 in terms of energy over the state-of-the-art write completion architectures. Because

Table 2.3: Energy of writes for a range of temperatures.

	$P \rightarrow AP$		$AP \rightarrow P$	
	Conventional	VT-VEW	Conventional	VT-VEW
$0\,^{\circ}\mathrm{C}$	6.31pJ	1.69pJ	7.34pJ	1.71pJ
$27\,^{\circ}\mathrm{C}$	6.43pJ	1.86pJ	7.52pJ	1.77pJ
$110\,^{\circ}\mathrm{C}$	6.46pJ	2.31pJ	7.72pJ	1.76pJ

the conventional VEW architecture is beneficial in all four transition cases and thus produces greater energy savings. Energy consumption of VEW and the prior work is shown in Table 2.4. VT-VEW results in smaller energy savings because it only saves energy during $P \rightarrow AP$ and $AP \rightarrow P$ transitions. Reliance on transition-detection means that this circuit is able to reduce energy for a write operation that involves state transitions. However, VT-VEW handles variations in both process parameters and temperature whereas VEW becomes fragile even for relatively small levels of variability. Note that in this evaluation the energy consumption of VT-VEW is the same as of the conventional write scheme for $P \rightarrow P$ and $AP \rightarrow AP$. This is also used in evaluations of VOW and EWT.

The area of the design is estimated to be $990F^2$. For comparison, the area of the conventional VEW design is $1860F^2$. The FFs can be removed because the monitoring circuit is turned off during the termination phase. Therefore, the unwanted feedback loop cannot occur in the termination phase eliminating the need in the FFs to break the feedback loop. In addition, only one monitoring circuit is required for both directions in this design because the proposed circuit works as a transition detector and the proposed detector can sense both transition directions.

The area overhead of the proposed circuit is also estimated in a sub-array. A sub-array is comprised of 512 rows and 512 columns, and the area of a 1-bit cell is $20F^2$ [4, 27]. Under these assumptions, the area overheads of VEW and VT-VEW are 18.2% and 9.7%, respectively. The relative area overhead of VEW and VT-VEW becomes 9.1% and 4.8% when the number of rows is 1024. In addition, the relative area overhead is reduced with column multiplexing, which is often used in STT-MRAMs. Recent work reports the use of 5- and 6-level column multiplexing [11, 28]. With 5-level column multiplexing, the relative area overhead becomes approximately 1% for the proposed VEW circuits. Therefore, under an efficient design methodology, the proposed techniques result in small overall area overhead.

2.3 Experimental Results

The energy saving achieved by applying the variable-energy write technique is evaluated in this chapter. We also compare VEW's energy savings with the related techniques. We compute the average bitcell write energy using a SPICE simulation and, then use it to evaluate the savings in a memory system on SPEC benchmarks [29, 30].

2.3.1 Evaluation Methodology

To evaluate the energy consumption of VEW, we consider a write operation as a three-stage process (depicted in Fig. 2.8). The first stage, T_{pre}, is the time after the write starts until the MTJ switch occurs (previous value held in the cell). The second stage, T_{ctrl}, is the time between the switch and the write current is termi-

nated; this is the reaction time of our monitoring and shutdown circuits. The third stage, T_{off}, is the time after shutdown until the end of the pre-determined pulse duration, which equals the time required to meet a given maximum error rate target. Without VEW, there are two stages, T_{pre} as before and T_{post}, which is the time after the switch with the current still on. The write energy is then calculated using the power consumed in each stage, as shown in Eq. 2.4 and Eq. 2.5 for baseline and VEW, respectively.

$$E_{baseline} = V_{DD}(T_{pre}I_{pre} + T_{post}I_{post}) \tag{2.4}$$

$$\begin{aligned} E_{VEW} = V_{DD}(T_{pre}I_{pre} + T_{ctrl}I_{ctrl} + T_{off}I_{off}) \\ +P_{wcc}(T_{pre} + T_{ctrl} + T_{off}) \end{aligned} \tag{2.5}$$

P_{wcc} in Eq. 2.5 is the power of the write completion circuit, which is the overhead of the technique. The circuit parameters, including its power consumption and reaction time (T_{ctrl}) are obtained using SPICE simulation ($V_{dd} = 1.05V$). We compute the overall pulse duration to match a desired error probability as determined by the probabilistic model shown in Eq. 2.3. We also calculate the expected durations T_{pre}, T_{off}, and T_{post} by applying the model of Eq. 2.3.

There is an important difference in current values that are supplied by the access transistor to the MTJ in the course of a normal operation of the cell. Depending on the value being written to the cell the bitline voltage is set high and the

source line voltage is set low, or vice versa. The effective V_{gs} of the access transistor is different in the two cases because of asymmetric bitcell configuration, resulting in significantly different current values of $2\times$ or more, unless word-line boosting is used in one of the two cases [31]. The difference in the delivered current leads to a significant asymmetry of MTJ switching times. It is beneficial for the overall energy minimization to use word-line boosting to make the distributions symmetric.

We also compare VEWs with the related verify-on-write (VOW) [8] technique and with early-write termination (EWT) [25]. To directly compare all three techniques within the same context, the same MTJ parameters are used (Table 2.1). We also use the same basic components for the write-completion circuit, but modify their application to mimic VOW and EWT. To mimic VOW, we remove the boosting write circuit because VOW was designed assuming asymmetric writes. Also, write current pulse is terminated an entire 64-bit word at a time, rather than each individual bit. We use Monte Carlo simulation to obtain the expected word-completion time, which is the expected longest duration write in each word (we measure the actual number of $P \rightarrow AP$ writes in each word). We use the write-duration distributions for $0.95I_{C0}$ and $2.06I_{C0}$ as described in Chapter 2.1.3 Without boosting, the circuit required $V_{DD} = 1.73V$. We also simulate an augmented version of VOW that can work with boosting (symmetric writes) by utilizing our monitoring circuit and using the Monte Carlo methodology assuming 50% of the bits in a word switch state. To mimic EWT, we only utilize our monitoring and shutdown circuits for those bits that do not change their state in each word-granularity write.

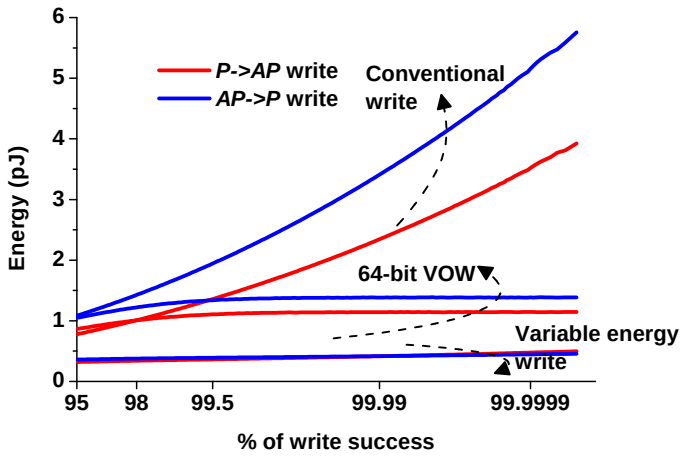


Figure 2.16: Per-bit energy of $P \rightarrow AP$ and $AP \rightarrow P$ writes: value-modifying writes.

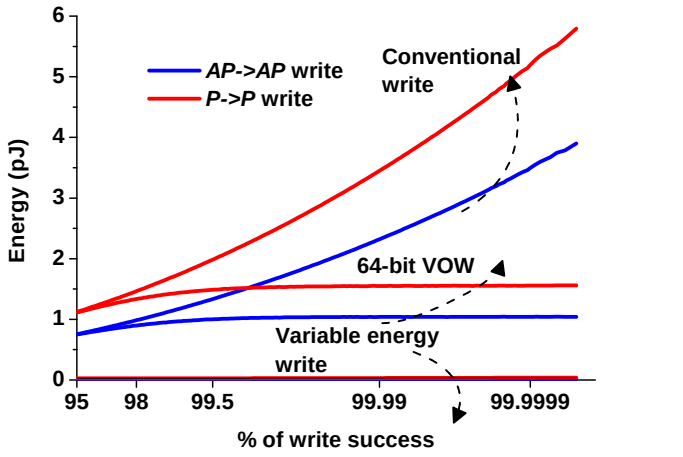


Figure 2.17: Per-bit energy of $P \rightarrow P$ and $AP \rightarrow AP$: value-maintaining writes.

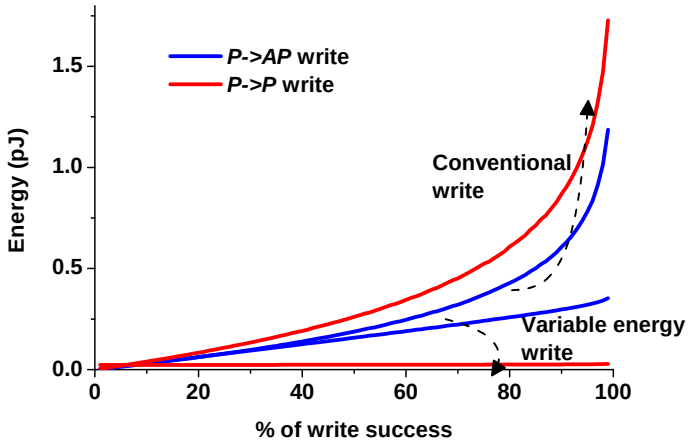


Figure 2.18: Per-bit energy of $P \rightarrow AP$ and $P \rightarrow P$ writes across a large range of write success probabilities.

2.3.2 Program-Level Energy

In addition to the validation results presented in Chapter 2.2.2, we now present the expected energy savings of using VEWs in a memory system using SPEC simulations [29, 30]. We compare the energy of conventional, VOW and VEWs across a range of write error probabilities: from an unacceptably high error rate of 5% to our target error rate of 1.5×10^{-7} [15]. These results are summarized in Fig. 2.16 and Fig. 2.17, which show the expected energy per bit for the cases of switching writes and writes that maintain already stored values ($P \rightarrow P$ and $AP \rightarrow AP$), respectively. Note that the horizontal axis in each figure represents

success probability, rather than error rate. For the value-maintaining writes, the error probability is zero, however, we use the same horizontal scale, where each success rate point corresponds to a certain pulse duration.

In all cases and across this entire error probability range, VEWs dramatically improve write energy compare with both the conventional baseline and a VOW with 64-bit sub-block(same as [8]). We assume 50% bits within one sub-block will change in each write in VOW. For the target error rate (1.5×10^{-7}), the savings are 87.3% for a $P \rightarrow AP$ write and 92.0% for a $AP \rightarrow P$ write. The energy savings are even more significant when no switching occurs and are 99.3% and 99.5% for $P \rightarrow P$ and $AP \rightarrow AP$ writes, respectively. To have a global view of how VEW compare with the conventional baseline write technique of fixed pulse duration, we show the expected write energy per bit across a wider range of write-success probabilities in Fig. 2.18.

Finally, to put the energy savings into the context of a memory system, the energy model is applied to the memory write traffic of SPEC CPU 2006 benchmarks. We capture a memory trace form the integer and floating point applications with the largest number of memory accesses with PIN [32], assuming an on-chip memory hierarchy with a single x86 Out-of-Order core, 32KB L1, 256KB L2 and a 1MB 16-way set-associative last-level cache. The simulator uses the cache configuration to filter memory access. We also consider that some data bits are unchanged and take advantage of this to reduce the write energy for the SPEC benchmarks. Fig. 2.19 shows the relative energy of VEWs and the previously proposed early write termination and verify-on-write designs [8, 25].

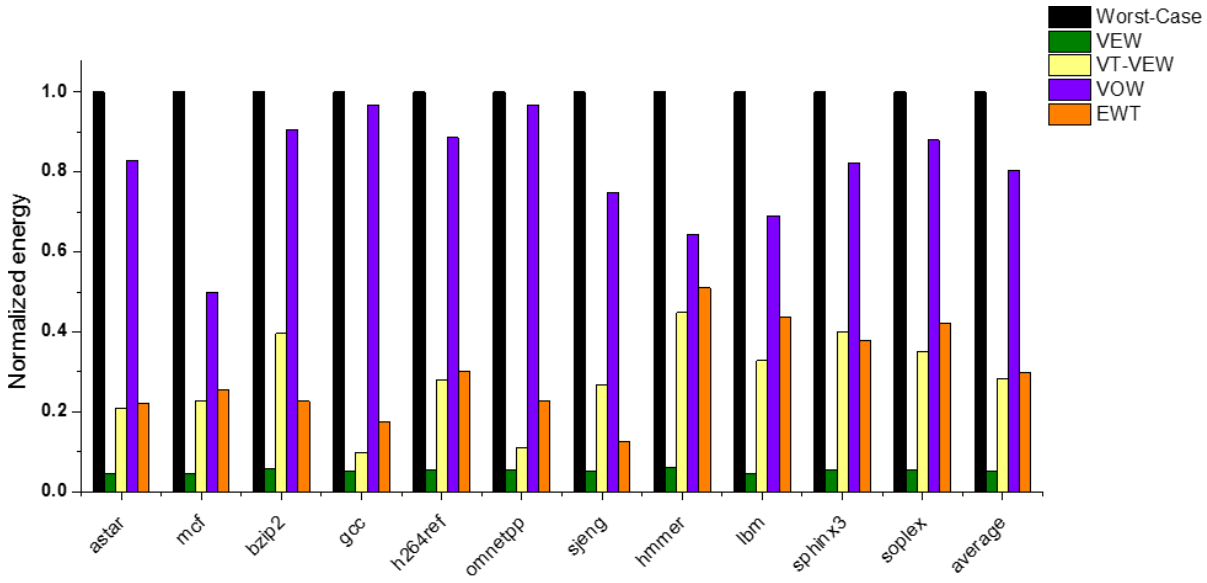


Figure 2.19: Overall write energy of worst-case VEWs, VOW, and EWT approach (normalized to the conventional worst-case write) by SPEC 2006 benchmarks

EWT is only able to reduce the energy of value-maintaining writes, and the word granularity shut down limited the benefit from VOW. VEW decreases total write energy consumption by 94.7% on average, while VOW and EWT only achieve 27.0% and 70.2% reductions, respectively. This is because VEW is very robust as they account for all 4 possible state combinations and monitoring and shutdown is applied for every bit independently. The energy savings are more significant in the cases when cells already store the desired values because write operations are terminated instantaneously.

VT-VEW reduces write energy by 71.6%, on average, as shown in Fig. 2.19. This energy reduction is due to two factors. First, VT-VEW greatly saves write energy in the cases when states are changed. The energy savings are 71.1% ($6.4pJ \rightarrow 1.9pJ$) for $AP \rightarrow P$ and 90.8% ($19.3pJ \rightarrow 1.8pJ$) for $P \rightarrow AP$ cases. The energy saving by VT-VEW is more significant for a $P \rightarrow AP$ case because a write '0' (AP) operation consumes more energy. The write '0' operation requires high voltage because of asymmetry of V_{gs} as described in Chapter 2.3.1. Second, the wordline boosting allows more energy reduction in the write '0' operation. Therefore, VT-VEW reduces energy more significantly on applications that have more write '0' operations and state-change writes.

2.4 Discussion

We now discuss the suitable application niche for the VEW techniques. An important consideration is the fact that the VEW circuit itself consumes non-negligible energy and therefore presents an overhead that needs to be carefully con-

Table 2.4: Performance summary and comparison with the related techniques.

	This work		[25]	[8]	[26]
	VEW	VT-VEW	EWT	VOW	Self-Termination
Technology	16 nm	16 nm	45 nm	45 nm	90 nm
Cell structure	1T-1MTJ	1T-1MTJ	1T-1MTJ	1T-1MTJ	2T-1MTJ[a)]
Area overhead	$1860F^2$	$990F^2$	13.44 µm ^{2}	-[b)]	$1582F^2$
Termination	$P \rightarrow P, P \rightarrow AP$ $AP \rightarrow AP, AP \rightarrow P$	$P \rightarrow AP$ $AP \rightarrow P$	$P \rightarrow P$ $AP \rightarrow AP$	$P \rightarrow AP$	$P \rightarrow P, P \rightarrow AP$ $AP \rightarrow AP, AP \rightarrow P$
Energy saving	94.7%	71.6%	70.2%	27.0%	69.1%[c)]
Detection circuit	Tuned inverter	S/H & Diff. Amp	Diff. Amp. & Sense Amp.	Sense Amp.	Tuned inverter
Variation-tolerant	No	**Yes**	No	No	No

[a)]A shared and a non-shared access transistors−This introduces area overhead by additional lines such as word and intermediate lines.

[b)]Not described in [8].

[c)]Average energy saving ratio from [26].

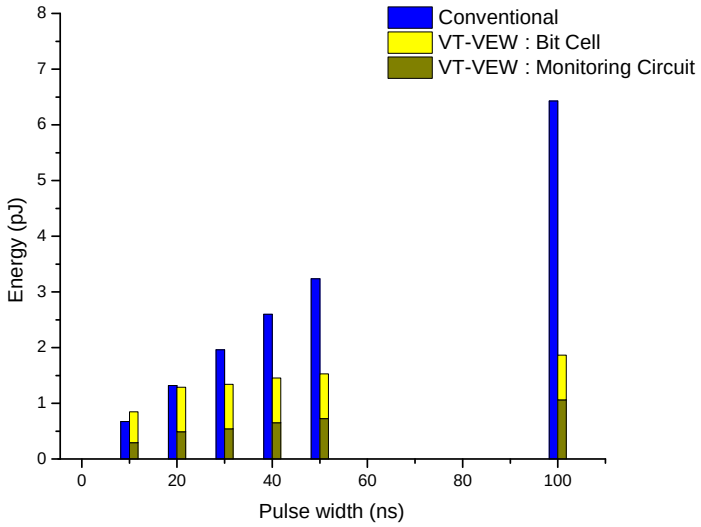


Figure 2.20: Energy consumption of VEW, decomposed into energy needed for a 1-bit write and for the monitoring circuit, and of the conventional architecture for $P \rightarrow AP$ at different pulse widths.

sidered. The overhead means that when the energy savings are not substantial, the benefits of VEW are limited. Specifically, for a short write pulse, the overall energy is small and thus the overhead of the VEW circuit can be larger than the savings it produces. With increasing write pulse duration, the energy savings of the VEW increase. Fig. 2.20 shows the 1-bit cell write energy for a conventional scheme and for VEW at different write pulse durations for a $P \rightarrow AP$ transition. We observe that for the $100\,\mathrm{ns}$ pulse, there is a 71.6% energy reduction enabled by VT-VEW. However, when the pulse is reduced to $20\,\mathrm{ns}$, the energy savings disappear. For the $10\,\mathrm{ns}$ pulse, VEW, in fact, consumes higher energy than the conventional write, illustrating the impact of the overhead of additional circuitry. Note that the VEW is effective in reducing write energy even at $10\,\mathrm{ns}$ if the state of the cell is not changed ($P \rightarrow P$ and $AP \rightarrow AP$) because VEW deactivates the write current at once.

A second important point is that at the very short write durations, there is little chance to exploit stochastic behavior because the distribution is very narrow. Taken together with the above-discussed impact of the overhead, we arrive at the conclusion that VEW is likely to be effective when an application operates in the regime of longer write durations. This is where the potential to exploit the stochastic write behavior is the biggest. Therefore, the VEW techniques may be more appropriate for the "slower" memory applications of STTs, such as for SCMs, which tend to have longer write time requirements.

In addition to energy overhead, area overhead needs to be minimized. VEW and VT-VEW consume relatively small area overhead when column-multiplexing is considered, but still introduce area overhead. Circuits for state change detection

and bit-line termination introduce area overhead.

2.5 Summary

This chapter presents a novel variable-energy write STT-RAM architecture with a write-completion monitoring. In the proposed architecture, the write process is continuously monitored and is terminated as soon as the MTJ reaches the required state. We also developed a variation-tolerant circuit for fast state change detection and evaluated it using a stochastic MTJ model targeting an implementation in a $16\,\mathrm{nm}$ technology node. The proposed technique has no significant area overhead and is easy to integrate into memory arrays. Analysis indicates that at the required write-error rate the proposed VEW and VT-VEW architectures reduce write energy by 94.7% and 71.6%, respectively.

Chapter 3

Multiple Attempt Write STT-RAM

This chapter[1] addresses the modeling of stochasticity of STT-RAM write and develops a multiple attempt write (MAW) architecture as an alternative design strategy to handle the stochasticity of STT-RAM write time. While VEW and VT-VEW introduce relatively small area overhead if column-multiplexing is used, it is desirable to further reduce the area overhead.

In this chapter, a different approach is proposed which is based on a hypothesis that an MTJ can be written by a current pulse of some magnitude and duration over multiple attempts, even when the likelihood of a successful write on each attempt is low. Write-verify-rewrite with adaptive period (WRAP) [8] and low-current probabilistic writes (LCPW) [16] exploit an iterative write algorithm for reducing write energy by lowering individual write currents. Lowering the write current leads to a high error rate; however, the writing ultimately succeeds on the following trials.

The proposed multiple attempt write (MAW) is a technique that is also built

[1]This chapter is an extended work of a publication: **Jaeyoung Park** and Michael Orshansky, "Multiple Attempt Write Strategy for Low Energy STT-RAM," *in Proc. of Great Lakes Symposium on VLSI*, Boston, MA, 2016. Jaeyoung Park's contribution of this paper is a circuit design and energy analysis of the proposed technique.

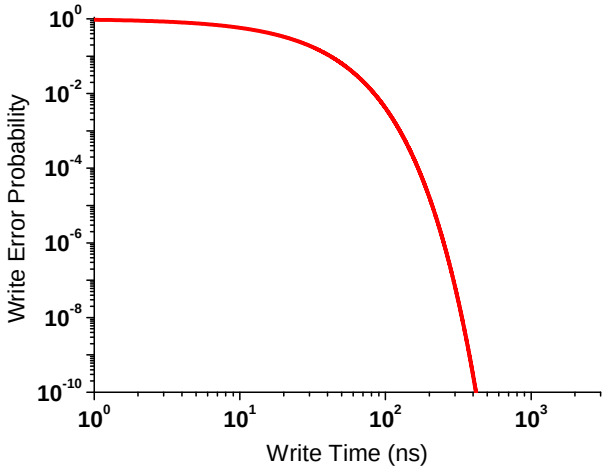


Figure 3.1: Stochastic distribution of write times

on the principle of applying multiple attempts. This research makes several contributions. The first contribution is a self-validating write circuit that allows bit-wise validation and deactivation with near-zero energy and area overhead. In contrast, WRAP and LCPW require a comparator for bit-wise validation and sizable control logic for deactivation. Because these are needed on every column of a memory array, a significant overhead is introduced. The second contribution is the formalization of the notion of re-randomization time as the interval needed for sufficient randomness to develop such that subsequent write attempts become stochastically independent. We quantify the value of re-randomization time via accurate numerical simulation.

We evaluate the proposed architecture using a compact STT-RAM cell model targeting an implementation in a $10\,\mathrm{nm}$ technology node. The experimental result

indicates that the overall write energy is reduced by 94.6% compared to the conventional worst-case write. Compared to WRAP and LCPW the write energy is reduced by $2.1\times$ with a 0.05% relative area overhead.

3.1 Modeling Stochastic STT-RAM Writes

In this chapter, we describe the STT-RAM write process, explain the opportunity for reducing energy, and discuss the stochastic model and the re-randomization process. We first revisit the stochastic STT-RAM write operation and then describe the re-randomization process. The proposed research started from an observation of the unique stochastic characteristic of the STT-RAM write operation. The core component in an STT-RAM cell an MTJ and an access transistor which is connected to the MTJ to control its operation as shown in Fig. 3.2 [9–11].

An access transistor is connected to the MTJ to control its operation. Write '0' and write '1' operations proceed by turning on the access transistor and injecting a high write current in one of two directions. Consider a write '1' operation. If an incoming data is '1', the voltage is raised on the bitline (BL) and lowered on the source line (SL). The access transistor turns on, and the current flows through the MTJ and the access transistor. If the current supplied by the access transistor is sufficiently large, the MTJ undergoes a state change.

As seen in Fig. 3.1, the time needed for the MTJ to switch is stochastic and the distribution is quite wide [21, 22]. The wider part of the distribution is dominated by the so-called thermally activated switching [23]. A probability model for the thermally activated switching duration has been proposed [21]. The model de-

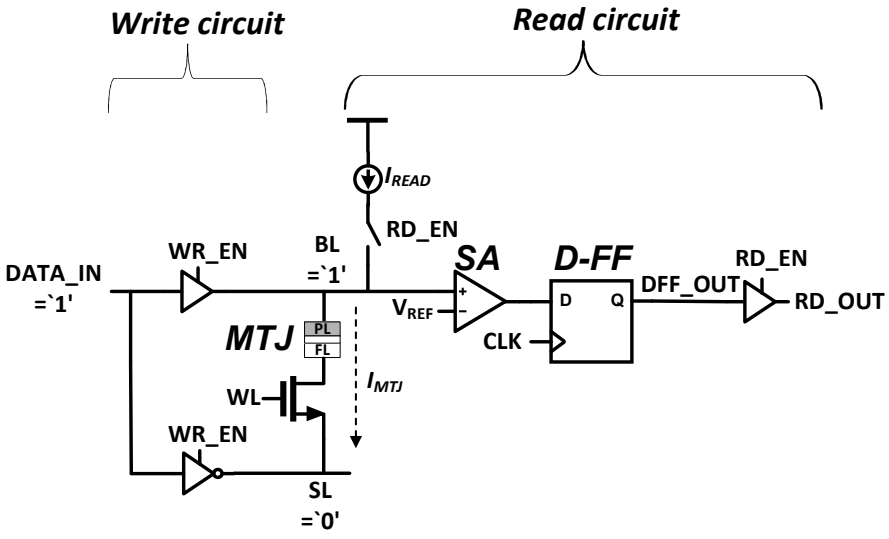


Figure 3.2: A 1-bit STT-RAM cell with conventional write and read circuits [9–11].

scribes the switching probability $p_{sw}\,(t, I)$, which is the probability of switching occurring for a pulse duration t at current I:

$$p_{sw}(t, I) = 1 - exp\Bigg\{ -\frac{t}{\tau_0} exp\bigg[-\Delta\bigg(1 - \frac{I}{I_{C0}}\bigg)\bigg]\Bigg\} \tag{3.1}$$

where Δ is the thermal stability factor, τ_0 is the inverse of the thermal attempt frequency, and I_{C0} is the critical current at zero Kelvin. Because of a wide distribution, the traditional approach which uses the pulse that guarantees writes for all cells wastes energy since the average write time is far shorter than the worst-case pulse duration. The critical observation that we exploit is that the stochastic behavior of writes of the same cell occurs when write events are interleaved with

sufficient thermal re-randomization processes [1, 2, 23]. This means that a certain time is required for the MTJ to develop sufficient randomness to ensure that writes are stochastically independent and are characterized by different switching durations. This time is called as the re-randomization time (τ_r) in this research and defined as the time needed to de-correlate the switching times of subsequent write events. At non-zero temperatures, thermal energy leads to the spontaneous agitation of magnetization, producing randomization of the initial angle (θ_0), which is the angle between the magnetic moment and easy anisotropy axis [23]. Differences in the initial magnetization angle lead to differences in switching time. Therefore, at above-zero temperature, the switching time of each write attempt is changed.

To evaluate re-randomization time, the detailed magnetics numerical simulator OOMMF [33] is used. OOMMF is a solver of the Landau-Lifshitz-Gilbert equation [34] describing the magnetization dynamics in a solid:

$$\frac{d\mathbf{M}}{dt} = -\gamma \mathbf{M} \times \mathbf{H_{eff}} + \lambda \mathbf{M} \times (\mathbf{M} \times \mathbf{H_{eff}}) \tag{3.2}$$

where γ is the gyromagnetic ratio, α is the Gilbert damping and $\mathbf{H_{eff}}$ is the effective field, including the anisotropy field, the demagnetizing field and the external applied field.

OOMMF is commonly used to model the time-domain behavior of a magnetic field in ferromagnetic materials. OOMMF is used to simulate the dynamics of magnetization of an MTJ device and the evolution of the magnetic field due to temperature. The simulation experiment is as follows. A common MTJ device design

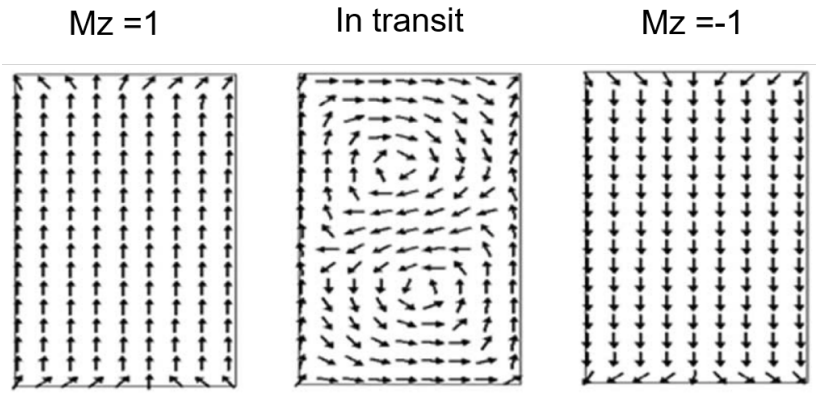


Figure 3.3: Orientations of spins for magnetization on Z-direction (Mz).

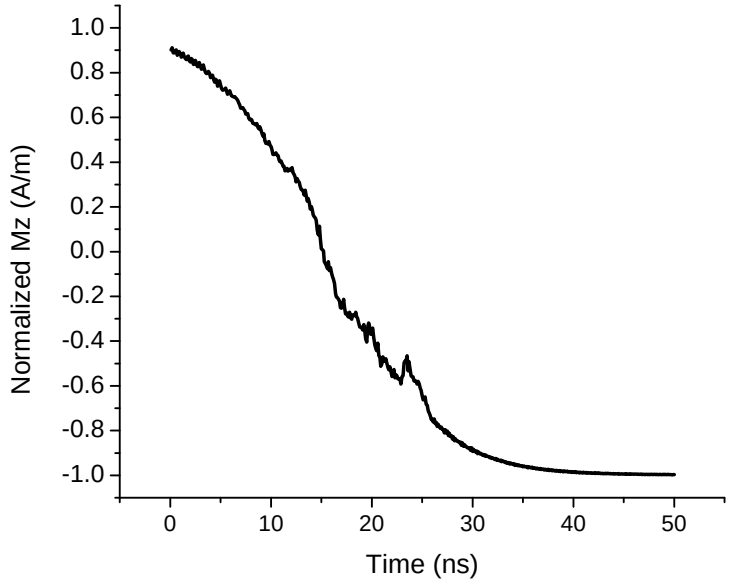


Figure 3.4: Change of Magnetization on Z-direction (Mz) over time by applying current.

that has a CoFeB-MgO interface is used [7] as shown in Fig. 2.3. The normalized magnetization on the Z-direction (Mz) is simulated, which represents the macro-spin angle in a free layer, over time and at various temperatures. The simulations are conducted without any applied current to only study the dependence of spins on temperature. When all spins in a free layer are up, the value of Mz is 1. If all spins are down, the value of Mz is -1 as shown in Fig. 3.3. Fig. 3.4 shows Mz changes over time when a sufficient current is applied to change the state of the MTJ. At 250-400K, which is the temperature range used in commercial electronic devices, Mz shows significant fluctuation over time as shown in Fig. 3.5. The modified angle becomes an initial angle for a subsequent write attempt, leading to different switching times. In Fig. 3.6, depending on the initial Mz, the switching time varies significantly.

To find the re-randomization time, the autocorrelation function for Mz(t) simulated at 300K at various time intervals is simulated (Fig. 3.7). Autocorrelation remains low down to the limit of OOMMF's time resolution ($0.2\,\mathrm{fs}$). This indicates that at the sub-fs scale the subsequent values of Mz are random. We observe that sub-fs is, for current technologies, significantly below the typical gate delay and is easily accrued within MAW during a validation step without any effort. The autocorrelation remains low up to $1\,\mathrm{fs}$ which is below the typical gate delay. This indicates that even at this scale the subsequent values of Mz are random.

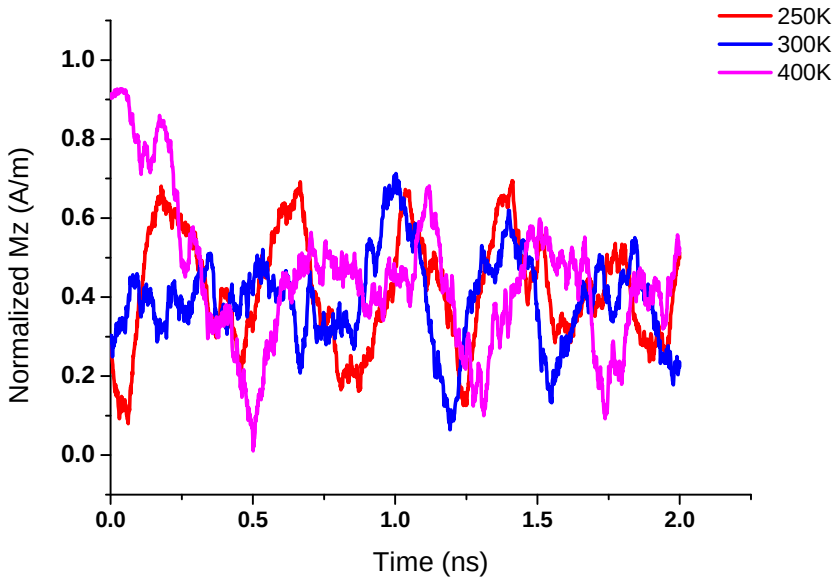


Figure 3.5: Magnetization on Z-direction (Mz) over time obtained via numerical simulations at various temperature.

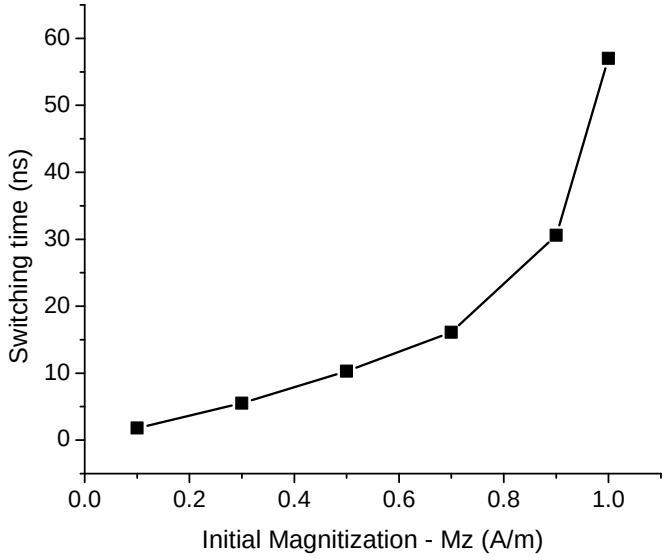


Figure 3.6: The simulated switching time as a function of initial magnetization.

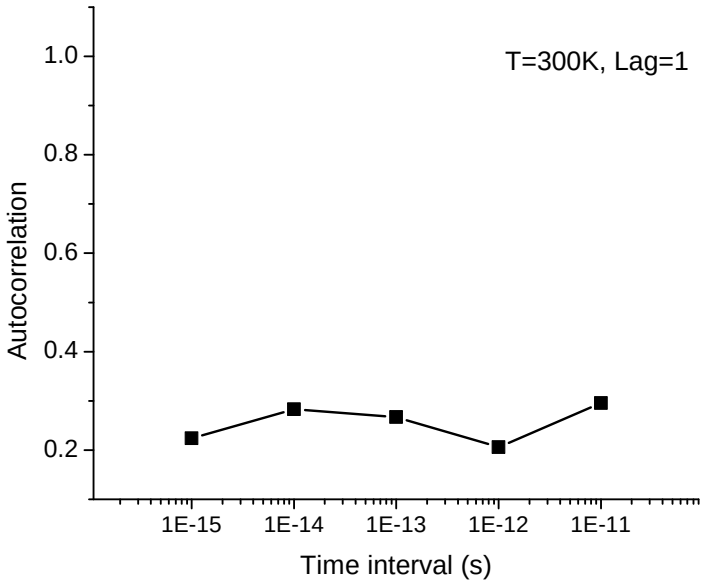


Figure 3.7: The autocorrelation function of Mz.

3.2 Multiple Attempt Write

In this chapter, a multiple attempt write (MAW) algorithm is introduced to overcome the wide distribution of write time by exploiting the re-randomization process. Both the MAW algorithm and its implementation are described.

MAW utilizes a multiple-iteration mechanism that: (1) writes to a cell, and (2) validates that the correct value has registered. If the cell is not written correctly, the MAW algorithm attempts to write the cell again until the cell is written successfully. Once the correct value has registered, the write current is deactivated, eliminating unnecessary energy consumption. Even though the probability of a write-fail on any individual attempt is high, the writing ultimately succeeds on the following attempts. That allows MAW to achieve high overall write success rate while reduc-

ing the needed energy. In contrast to the VEW that that continuously checks the instantaneous state of the MTJ and deactivates the write current, the MAW architecture validates whether the correct value has registered after each write attempt. Fig. 3.8 shows a conceptional comparison of the MAW with VEW. We assume that in both cases switching occurs at 30ns. The write current is deactivated at 30ns in the VEW architecture while the MAW stop writing after the second attempt at 40ns.

In Fig. 3.9 and Fig. 3.10 we provide the waveforms produced by MAW and those of the conventional (worst-case) write scheme. Fig. 3.9 and Fig. 3.10 shows the waveforms for the incoming data (DATA_IN), the current through an MTJ (the write current pulse) for both (a) the conventional worst-case approach and (b) the MAW approach. The applied pulse duration (DATA_IN) is fixed at $80\,\mathrm{ns}$ for the conventional strategy and at $20\,\mathrm{ns}$ for MAW. Assuming that in both cases switching occurs at $38\,\mathrm{ns}$: at that time a cell is written successfully and the write current is no longer necessary. Despite that, in the conventional approach, the write current (I_{MTJ}) continues to be drawn until $80\,\mathrm{ns}$. MAW deactivates the write current after the second attempt because MAW reads out the cell content and validates the state of the MTJ before each write attempt. In this example, a higher than 40% energy reduction is achieved by cutting the unnecessary write current assuming the validation energy overhead is negligible.

To maximize the advantages of MAW, the proposed circuit needs to minimize the energy cost of validation. What is the best way to realize the strategy of writing, validation, and current-deactivation? One possibility is to use a digital comparator (e.g., XOR gate). This is a strategy of WRAP and LCPW. The com-

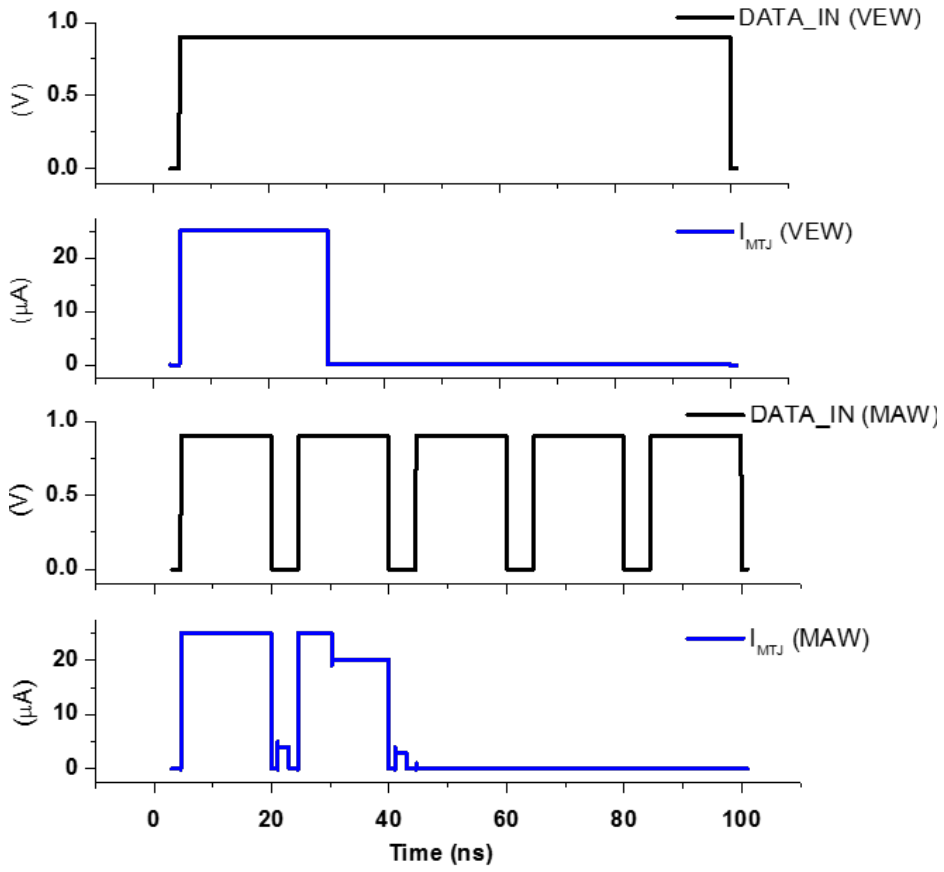


Figure 3.8: Conceptional comparison of the MAW with the VEW architecture.

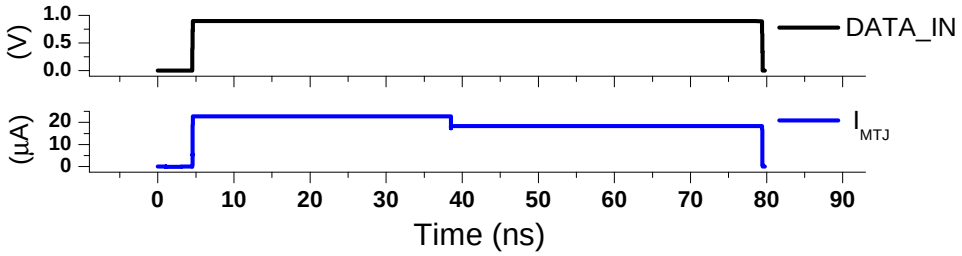


Figure 3.9: Waveforms of the conventional worst-case approach for the write '1' operation.

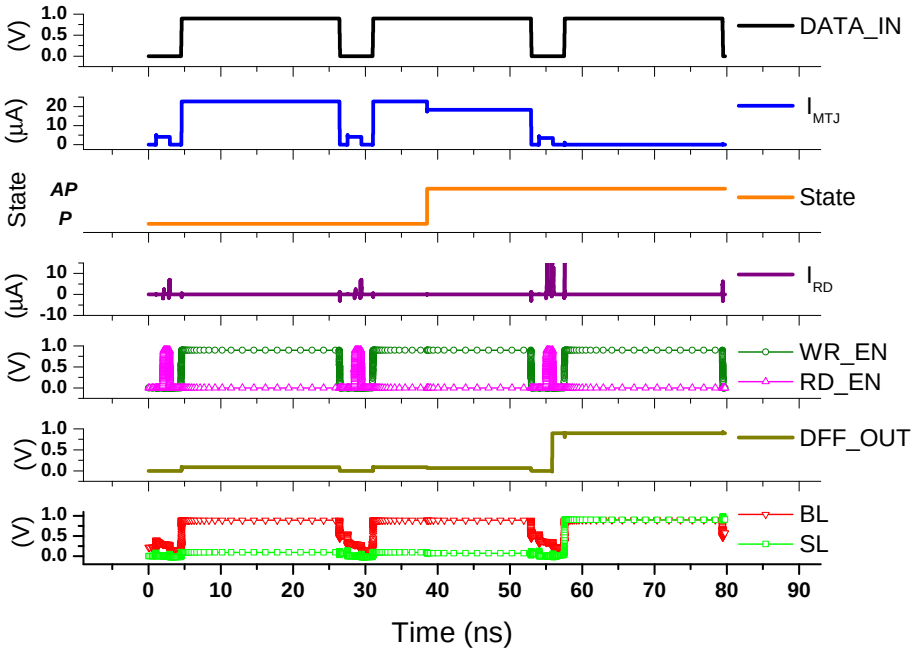


Figure 3.10: Waveforms of MAW for the write '1' operation.

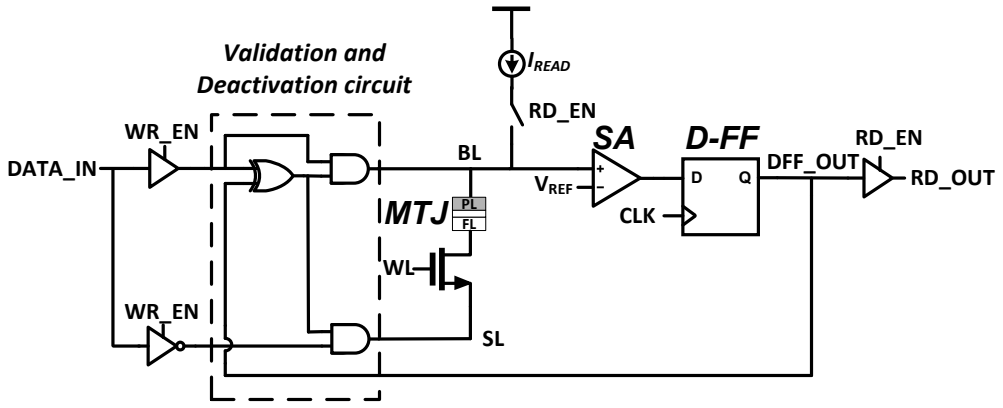


Figure 3.11: The validation and deactivation circuits of WRAP and LCPW. LCPW additionally requires a write buffer.

parator output could then disable the write as shown in Fig. 3.11. However, as the results show, the additional circuits for validation consume non-negligible area and energy. The area overhead can be significant as the additional circuits need to be placed on every column for bit-wise validation. The area cost of WRAP and LCPW is 8.2% and 13.5%, respectively. In contrast, the proposed solution requires no extra area.

To minimize the overhead, the proposed architecture is to modify the MTJ cell directly to allow a self-validation/deactivation process. The self-validation/deactivation process is from an important observation that a deactivation process is accomplished if both the bit line (BL) and the source line (SL) have the same value. Consider a write '1' operation. If an incoming data is '1', the voltage is raised on BL and lowered on SL. The write '1' operation is deactivated if BL is '1' as shown in Fig. 3.12.

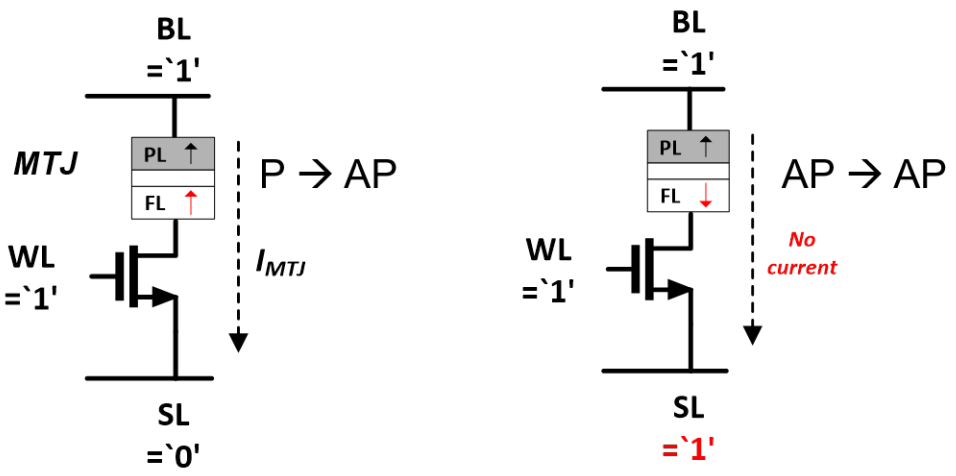


Figure 3.12: BL and SL values for the write ‘1’ and the deactivation operations.

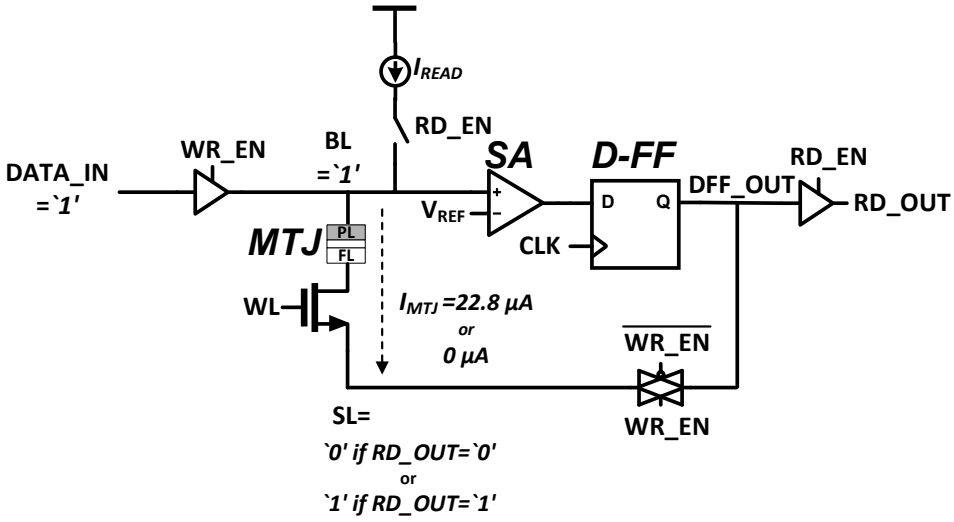


Figure 3.13: The self-validation write circuit (write ‘1’ operation).

Table 3.1: BL and SL conditions for all write and deactivation operations.

	DATA_IN	BL	SL	Write current
Write '1'	1	1	0	$BL \rightarrow SL$
	1	1	1	Deactivated
Write '0'	0	0	1	$SL \rightarrow BL$
	0	0	0	Deactivated

This is accomplished if the value stored on the cell (i.e., DFF_OUT) is delivered to SL by a read before a write. The cell content available on SL is compared with the incoming data on BL during a write. Fig. 3.13 shows a schematic view of the proposed circuit. SL is connected to DFF_OUT via a switch. DFF_OUT becomes equal to the cell content after a cell is read out via the sense amplifier and D-FF. Note that this utilizes the standard read infrastructure. Now, the write current is determined by SL's value. Consider a write '1' operation. If SL is 0, the write current is passed through the cell from BL to SL. If SL is 1, the write current is deactivated since both BL and SL are 1. SL and BL conditions of write '1' and '0' operations are summarized in Table 3.1.

Fig. 3.14 shows the write operation sequence by the MAW with the proposed circuit for the write '1' operation. It starts with a read operation to propagate the current value of a cell to SL. If the DFF_OUT is 1, no current is passed through the cell since both ends have the same value. If DFF_OUT is 0, MAW attempts to write. In the second attempt, the cell is read out again, and it continues or stops writing, depending on the read-out value. These read and write operations proceed until the cells is written successfully.

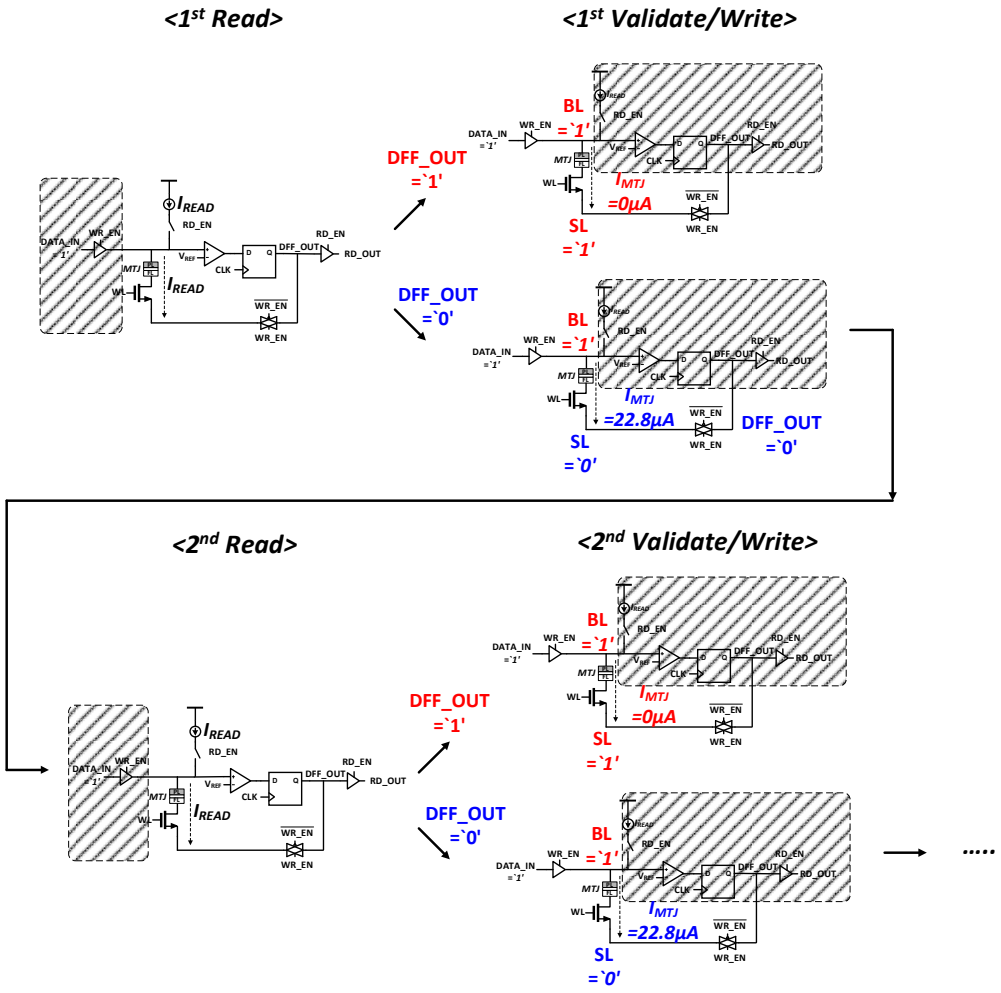


Figure 3.14: The MAW operation sequence for the write ‘1’ operation.

The waveforms at each node of the proposed circuit are obtained using SPICE simulation [35] as shown in Fig. 3.10. The waveforms show the change of the current across an MTJ, and the voltage on DFF_OUT and SL as the state of the MTJ changes. Note that the MTJ is set to *P* state at the beginning. As the MAW sequence is read/validate/write, the proposed sequence starts with a read operation. At the start of the first read operation, the voltages on DFF_OUT and SL are lowered since the MTJ is in *P* state. The voltage on BL is raised by DATA_IN. During the first write operation, the write current (I_{MTJ}) is passed through the cell. Switching does not occur in the first write attempt; therefore, DFF_OUT remains low during the second read operation. The state transition occurs in the middle of the second write attempt. The state change is captured in the third read operation and modifies the voltage on DFF_OUT. The voltage on SL is raised in the third write attempt. Therefore, the write current is deactivated in the third write attempt because both BL and SL are high.

3.3 Experimental Results

In this chapter, energy savings achieved by applying MAW is evaluated. First, the average bitcell write energy is calculated and then evaluated the savings on SPEC benchmarks [29, 30].

3.3.1 Bitcell Write Energy

We designed a 1-bit cell with the proposed MAW write circuit using the $10\,\mathrm{nm}$ Predictive Technology Model (PTM) MOSFET model and a compact MTJ

Table 3.2: Key parameters of a perpendicular MTJ [1, 3, 7].

Parameter	Value	Unit
Interface Material	CoFeB-MgO	
Intrinsic critical current	24	μA
Thermal stability factor	58	
Tunnel Magnetoresistance ratio (TMR)	$\sim$100	$\%$
Diameter of MTJ	20	*nm*
Out-of-plane magnetic field	0.4	*T*
Product of resistance and area (RA)	12	$\Omega\,\mu\mathrm{m}^2$

model [1–3] to verify functionality and extracted single bit write energy using SPICE simulation. MTJ parameters are derived from a perpendicular MTJ manufactured by Tohoku University [7], as detailed in Table 3.2. We also implemented WRAP and LCPW to extract their area and energy overheads.

We first extract the 1-bit write energy of MAW, WRAP, and LCPW. We use the MTJ parameters of Table 3.2 and the same basic components, such as a write driver. The power supply voltage is 0.9V and the write current across an MTJ (I_{MTJ}) is 22.8μA. The single attempt write energy is 0.27pJ at the $13\,\mathrm{ns}$ pulse duration. To compute the *expected (average)* write energy of MAW, a simple probabilistic model is derived. The average write energy is the expected value of the cost of performing Bernoulli trials with error probability, p, until the first success [36]. The single-attempt error probability can be found from Eq. 3.1 as $p = 1 - p_{sw}\,(t, I)$. If *X* is the number of trials until the first success, $p(X = k) = p^{k-1}(1 - p)$. The expected energy is found by summing up the geometric series when e is the cost of an individual write attempt including the contribution from read energy:

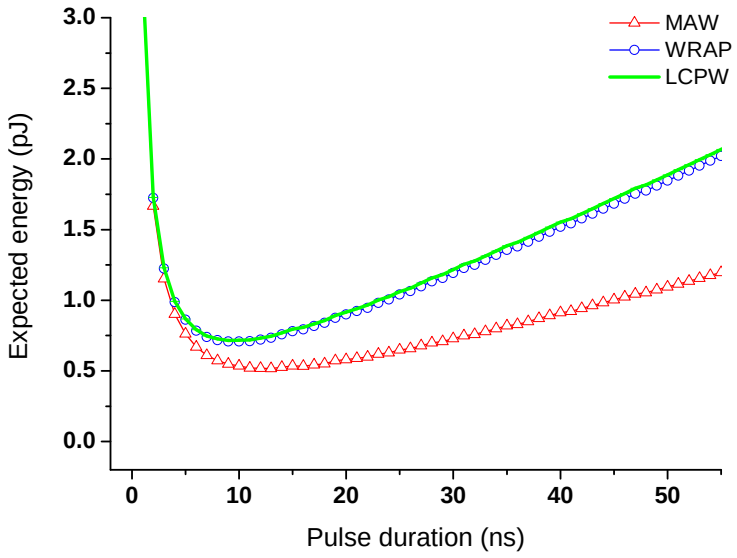


Figure 3.15: Expected write energy of MAW, WRAP, and LCPW at various write pulse durations.

$$E(energy) = e\sum_{k=1}^{n} kp^{k-1}(1-p) = e(\frac{1-p^n}{1-p} - np^n) \tag{3.3}$$

The expected energy is plotted to find the optimal pulse duration and the minimum total energy. Fig. 3.15 shows the expected energy of WRAP, LCPW, and MAW at different write pulse durations. For short pulses, the number of attempts needs to be increased, leading to higher energy. The minimum expected energy of MAW is 0.52pJ. It occurs at the pulse of $13\,\mathrm{ns}$ and requires 1.9 write attempts on average and 22 write attempts in the worst case.

The average power of additional circuits of WRAP and LCPW is $14.4\mu W$ and $15.3\mu W$, respectively. The minimum expected energy of WRAP and LCPW

are 0.70pJ and 0.72pJ, respectively. The write energy of the conventional approach is 5.86pJ, where the pulse duration is set to $286\,\mathrm{ns}$ to achieve the typically required error rate (1.5×10^{-7}) [15]. Note that the worst-case latencies of WRAP, LCPW, and MAW are increased by additional read latency since a read operation is inserted between each write attempt. The read circuitry is designed to enable $1\,\mathrm{ns}$ read time at 0.01pJ energy. The total read energy overhead is 3.8% (0.22pJ/5.86pJ) and the latency overhead is 7.7% ($22\,\mathrm{ns}$/$286\,\mathrm{ns}$). The longer latency does not impact performance under a write scheme utilizing a write buffer. Write buffers are typically used to mitigate the impact of long latency [16, 37, 38].

We estimate the area costs of WARP, LCPW, and MAW. The area cost of WARP is one XOR gate and two AND gates per column. The area cost of LCPW is one XOR gate, two AND gates, two inverters, and one flip-flop. The area overheads of WRAP and LCPW are estimated to be $840F^2$ and $1380F^2$, respectively. If the area of a 1-bit cell is $20F^2$ ($4F \times 5F$) and a sub-array is comprised of 512 rows and 512 columns, the area overhead of WRAP and LCPW is 8.2% and 13.5%, respectively. The area cost of MAW is one CMOS switch that connects DFF_OUT to SL (Fig. 3.13). However, one inverter, which is originally connected to SL in the conventional write driver (Fig. 3.2), is removed. Because the area of a CMOS switch is generally similar to the area of an inverter, the proposed solution requires no extra silicon area. Two metal lines are added for control signals, as illustrated in Fig. 3.16. The area overhead of MAW is estimated to be $10F^2$ per column ($2 \times 1F \times 5F$). The relative area overhead is a 0.05% in a sub-array comprised of 512 rows and 512 columns.

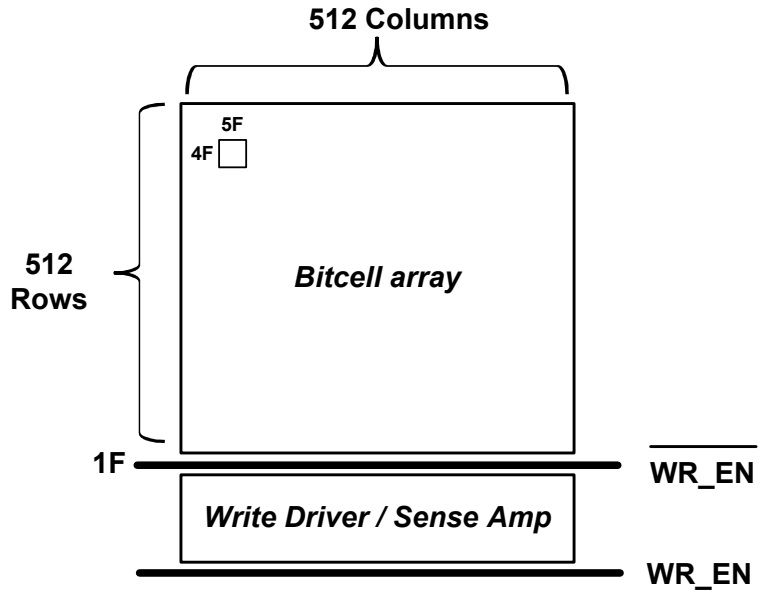


Figure 3.16: Block diagram of a memory array with additional control signals.

3.3.2 Program-Level Energy

The energy savings of MAW are evaluated in a memory system using SPEC benchmarks [29, 30]. The expected write energy is used for a bit for $P \rightarrow AP$ and $AP \rightarrow P$ write cases. In cases when a cell already stores the desired values ($P \rightarrow P$ and $AP \rightarrow AP$), only the read energy is required because of MAW's read-first sequence. In contrast, WRAP and LCPW use write energy for the first write attempt even for $P \rightarrow P$ and $AP \rightarrow AP$ cases because of their write-read-verify sequence.

To put the energy savings into the context of a memory system, we apply our energy models to the main memory write traffic of SPEC CPU 2006 benchmarks. Memory traces from integer and floating point applications are used with the largest

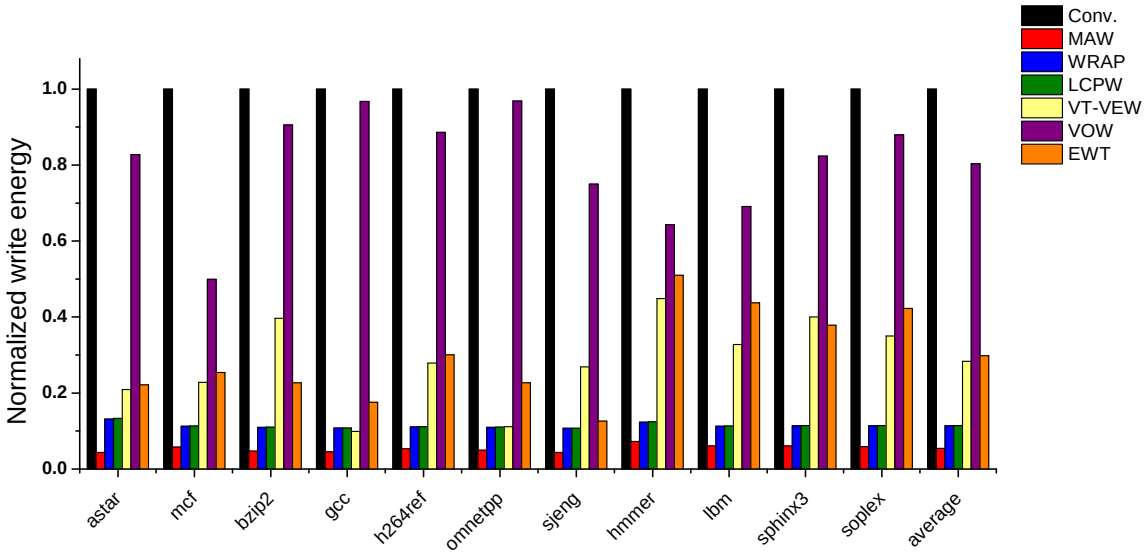


Figure 3.17: Overall write energy of the conventional worst-case, MAW, WRAP, LCPW, VT-VEW, VOW, and EWT approach (normalized to the conventional worst-case write) by SPEC 2006 benchmarks.

number of memory accesses with PIN application programming interface [32], assuming an on-chip memory hierarchy with a single X86 core, 32KB L1, 256KB L2 and 1MB 16-way set-associative last-level cache. Fig. 3.17 shows the relative energy consumption of MAW and earlier techniques. Write energy is normalized to the conventional write. MAW reduces energy by more than 95% in 7 applications. In applications that have a higher fraction of non-redundant write ($P \rightarrow AP$ and $AP \rightarrow P$) such as shpinx3 and hmmer, MAW produces slightly lower savings because it saves more energy on redundant write cases ($P \rightarrow P$ and $AP \rightarrow AP$). On average, MAW reduces write energy by 94.6%. WRAP and LCPW only achieve 88.6% and 86.5% reduction, respectively. Thus, compared to the best previously known architecture, MAW achieves a $2.1\times$ larger energy reduction.

This improvement is due to two factors. The first is that MAW introduces less energy overhead for validation in contrast to other techniques. The second is that MAW's read-first sequence allows energy reduction in the cases when cells already store desired values significantly. The energy saving on redundant write cases is more than 99% because only read energy is used (0.01pJ). The energy saving on non-redundant write is relatively small: 89.14% ($4.80pJ \rightarrow 0.52pJ$) for $P \rightarrow AP$ and 90.97% ($5.85pJ \rightarrow 0.53pJ$) for $P \rightarrow AP$ cases. Therefore, MAW reduces energy more for applications that have fewer transitions. All performance metrics are summarized in Table 3.3.

Table 3.3: Performance summary and comparison with the related techniques.

	MAW	WRAP [8]	LCPW [16]	**VEW** [39]	**VT-VEW** [40]	VOW [8]	EWT [25]
Technology	$10\,\mathrm{nm}$	$45\,\mathrm{nm}$	-[a)]	$16\,\mathrm{nm}$	$16\,\mathrm{nm}$	$45\,\mathrm{nm}$	$45\,\mathrm{nm}$
Addi. circuit	2 metal lines	Comparator Comb. circuits	Comparator Comb. circuits, Buffer	Tuned inverter, FF Comb. circuits	Diff. Amp. Comb. circuits	Sense Amp. Comb. circuits	Diff. Amp. Comb. circuits
Area overhead	$10F^2$	$840F^2$	$1380F^2$	$1860F^2$	$990F^2$	$860F^2$	$13.44\,\mu\mathrm{m}^2$
Energy saving	94.6%	88.6%	88.5%	94.7%	71.6%	27.0%	70.2%

[a)] Not described in references.

3.4 Summary

This chapter presents a multiple attempt write architecture to reduce average write energy. In the proposed architecture, a cell is validated before each write attempt and the write operation is deactivated once the correct value has registered. A novel circuit is developed for the write operation based on self-validation. Analysis indicates that at the required write-error rate the proposed architecture reduces write energy by 94.6%. The MAW has an improvement of a factor of 2.1 in terms of energy over the state-of-the-art architectures with a 0.05% relative area overhead.

3.5 Appendix: Details of Magnetics Numerical Simulation via OOMMF

This chapter presents the details of magnetics numerical simulation via OOMMF. Object Oriented MicroMagnetic Framework (OOMMF) is a software developed for micromagnetic numerical simulations at the National Institute of Standards and Technology (NIST) in 1999. This software utilizes an Landau-Lifshitz (LL) ordinary differential equation (ODE) to predict the magnetization dynamics in a solid, implementing a simple first-order forward Euler method and several RungeKutta methods [41–43]. The various forms of the LL equation are commonly used in micromagnetics to model the effects of a magnetic field on ferromagnetic materials. However, the exact closed-form solution of the equation is only available if boundary and initial conditions are provided. Alternatively, a numerical approximation is used with one of two common approaches, finite-differential method (FDM) and finite-element method (FEM) [42, 43]. OOMMF uses the FDM method to solve the LL equation for estimating the magnetization dynamics in a solid.

We used OOMMF to simulate the dynamics of magnetization of an MTJ device and the evolution of the magnetic field due to temperature. Specifically, we used Theta evolver module of OOMMF, which is an extension module of OOMMF that models finite temperature effect via a differential equation of Langevin [33, 44, 45].

The time evolver implements a simple first order forward Euler method with

step size control on the Landau-Lifshitz differential equation (LL) [46, 47]:

$$\frac{d\mathbf{M}}{dt} = -|\bar{\gamma}|\,\mathbf{M} \times \mathbf{H}_{\text{eff}} - \frac{|\bar{\gamma}|\alpha}{\mathbf{M_s}}\,\mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{\text{eff}}) \tag{3.4}$$

where $\mathbf{M}$ is the magnetization that represents all forces acting on the magnetic moment, $\mathbf{H}_{\text{eff}}$ is the effective field, γ is the gyromagnetic ratio, and α is the damping constant. This is illustrated in Fig. 3.18 that shows magnetization precession movements with the damping.

There are no temperature effects in the evolver. In order to take temperature into account, Theta evolver adds a highly stochastic fluctuating field term ($\mathbf{h}_{\text{fluct}}$) representing the irregular influence of temperature in the LL equation [33, 45]. The LL equation then becomes the stochastic Landau-Lifshitz differential equation (SLL):

$$\frac{d\mathbf{M}}{dt} = -|\bar{\gamma}|\,\mathbf{M} \times (\mathbf{H}_{\text{eff}} + \mathbf{h}_{\text{fluct}}) - \frac{|\bar{\gamma}|\alpha}{\mathbf{M_s}}\,\mathbf{M} \times [\mathbf{M} \times (\mathbf{H}_{\text{eff}} + \mathbf{h}_{\text{fluct}})] \tag{3.5}$$

where $\mathbf{h}_{\text{fluct}}$ follows a Gaussian distribution that includes numerous interactions with the underlying crystal lattice. Because $\mathbf{h}_{\text{fluct}}$ has no correlation in time, space, and axis direction in the most magnetic devices, the macro-spin fluctuates at random over time as a white noise. We used OOMMF to simulate the normalized magnetization on the Z-direction (Mz), which represents the macro-spin angle in a free layer, over time. Without temperature, Mz stays constant while Mz shows

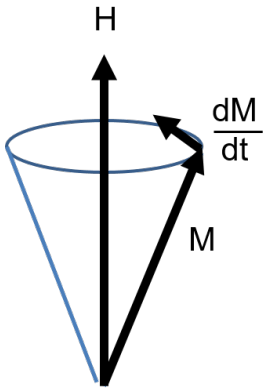


Figure 3.18: Magnetization precession movements with the damping.

significant fluctuation with temperature over time as shown in Fig. 3.19. When all spins in a free layer are up, the value of Mz is 1. If all spins are down, the value of Mz is -1.

However, several limitations remain to quantify the exact value of the re-randomization time because OOMMF reaches a computational limit. The smallest value in OOMMF is 2.2204×10^{-16} since OOMMF follows IEEE 754 compatible floating point [48]. Also, Theta evolver only computes Mz correctly above 1K [33]. Thus, we performed simulations down to 10K and 1fs scale. At 10K, Mz still shows significant fluctuation over time as shown in Fig. 3.20.

We find that autocorrelation remains low down to the limit of OOMMF's time resolution above OOMMF's temperature limit. This indicates that at sub-fs scale the subsequent values of Mz are random within the temperature range used in commercial electronic devices (250-400K). Even though we are unable to find

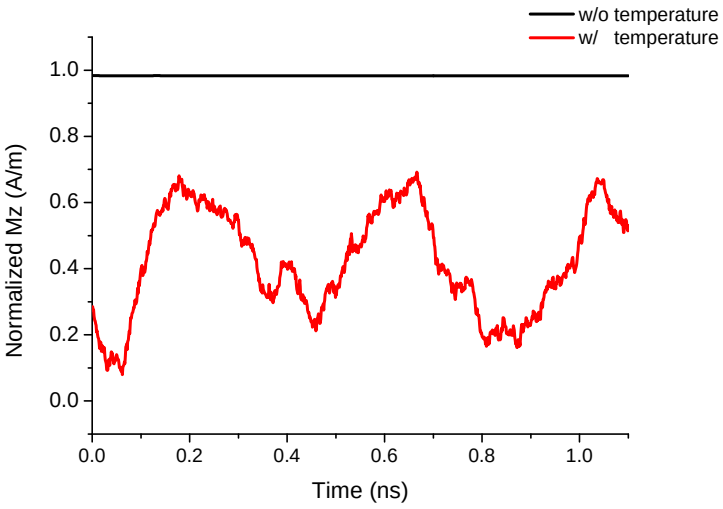


Figure 3.19: Magnetization on Z-direction (Mz) over time obtained via numerical simulations.

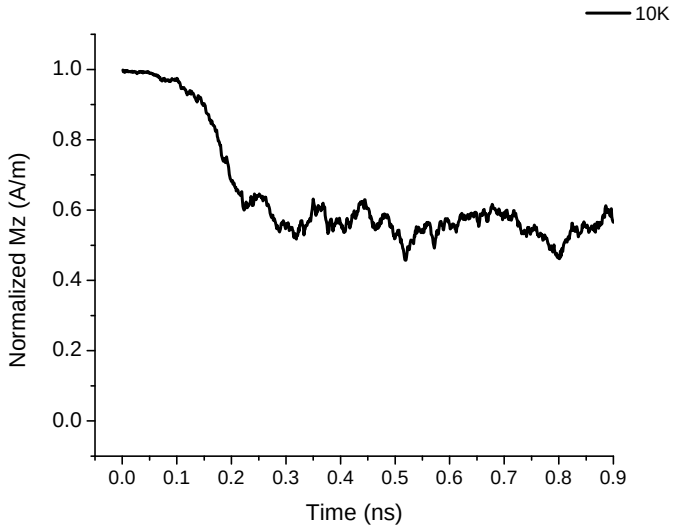


Figure 3.20: Magnetization on Z-direction (Mz) over time at 10K.

its exact value, we conclude that sub-fs is significantly below the typical gate delay and is easily accrued within MAW during a validation step without any effort. The following codes show a MIF file which is used for the OOMMF simulation.

Key codes for OOMMF simulation: MIF (Micromagnetic Input Format)

```
# MIF 2.1

# Setup constant values

set pi [expr 4* atan(1.0)]

set mu0 [expr 4* $ pi* 1e-7]

set gt [expr 1/650e3]

set gt2 [expr 1/500e3]


# The number of stages to execute

set NumStageCount [expr 300]

# Unit time for each stage (sec unit)

set DeltaTStage [expr 1e-11]

set StopTime [expr $ NumStageCount* $ DeltaTStage]


# Determine an MTJ size

Specify Oxs_ImageAtlas:top {
```

```
        xrange { 0 40e-9}
        yrange { 0 18e-9}
        zrange { 12e-9 14e-9}
viewplane xy
        image elp_spin.bmp
colormap
{ black cotop
white nm}
}

# Determine time evolver, temperature, initial seed
Specify Xf_ThermSpinXferEvolve:evolve [subst {
        alpha 0.1
         mp { 0 0 1}
        J 1e12
        P 0.4
        Lambda 2
        J_profile Jprofile
        J_profile_args total_time
do_precess 1
```

```
temperature 300

uniform_seed 111

} ]


# Apply current after 20ns to flip spin

proc Jprofile { t } {

        set scale 0.0;

        if { $ t¿20e-9} {

                set scale -0.2

        }

        return $ scale

}
```

Chapter 4

Efficient Fault Injection Flow

Voltage and temperature variations are major sources of hardware faults [49–51]. Voltage droops and high temperature increase gate delay, resulting in timing faults on near-critical paths. Voltage variation is caused by a variety of sources. First, voltage droop at the board- and package-levels tends to be relatively small and has long time constants ($\sim$100ns), thus, affecting many cycles. However, this component of voltage droop can be typically mitigated through the use of off-chip decoupling capacitors [52, 53]. Second, voltage droop caused by the behavior of the on-chip power grid tends to be larger and has shorter time constants ($\sim$1ns). The large voltage droop increases gate delay significantly, resulting in timing faults. The most effective mitigation technique is to rely on the on-chip decoupling capacitors. With the proper deployment of decoupling capacitors, the majority of voltage droop events do not exceed a single clock cycle [54–56]. We build our work on the premise that the multi-cycle droop events are handled by the above design strategy and focus on the class of single-cycle droop events.

These timing faults by voltage droops and high temperature are likely to propagate to architecture and application states, often leading to critical system failures. Developing a fault model and injection tool that generates and propagates

faults from circuit- to gate-level is important for accurately predicting the resulting system failures. This is challenging since the model needs to accurately capture the physical characteristics at the circuit level that define the likelihood of a fault and use that information to guide the injection with the proper probability. At the same time, the analysis and fault injections need to be computationally manageable to allow analysis of realistic systems under realistic workloads.

At the physical level, the faults generated by voltage droops and temperature, manifest timing faults at the latching flip-flops of near-critical paths. Circuit-level simulators can identify such timing faults by analyzing near-critical paths. However, the cost of such simulations is high, and they can only be used for a small fraction of designs and/or cycles making them infeasible for system-level analysis: $>$100 days per cycle on a circuit which has a realistic chip size [57, 58].

The problem of fault analysis has been widely studied. Various gate-level fault simulators have been developed [50, 59–61]. These gate-level simulators typically select fault-injection location (flops) at random, but the fault-injection time (a cycle) is determined when a transition is observed on the flops. A gate-level simulator is called for selected modules to check a transition on the flops because timing faults are only captured by the flops with a transition. A one-cycle delay is added to the selected flops of the modules and fault values are injected into the flops when a transition is observed. After the injection, the analysis continues to determine wheather the fault has been masked or a system failure has occurred. The challenge of a gate-level simulation utilizing commercial tools, such as Synopsys VCS, is that they are slow: $\sim$1s per cycle for OpenSPARC core [61, 62].

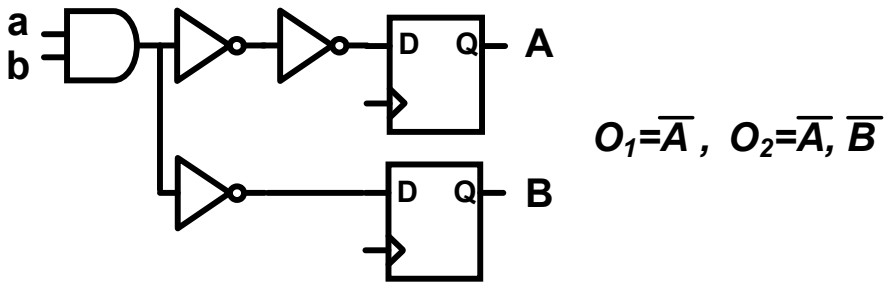


Figure 4.1: An example circuit for describing possible path-outputs.

Also, this approach can produce false injections because faults are injected randomly into a possible set of targets. This may be appropriate for sources of faults that are independent of each other. However, typically, it is unreasonable to model voltage-induced timing faults in a set of flops as independent. For example, if a voltage droop leads to a timing fault on a path in a given module then a longer path within the same module will also experience a fault. This is true under the assumption that gates within a single module are impacted by voltage in a correlated manner. This is illustrated in Fig. 4.1: the possible path-outputs are $(\overline{A})$ and $(\overline{A}, \overline{B})$, however, $(\overline{B})$ never occurs since the delay of path a-A is always greater than the delay of path a-B. Note that because of the importance of linking an output to a specific path we use a term path-output to refer to the latching flip-flop of a given path. However, a random fault injection can generate a fault on (B), which is a *false injection*.

To resolve the above correlation of realistic faults, we develops a hybrid gate-level fault simulator capable of capturing fault correlations through circuit-

level timing analysis. A gate-level simulator is used to check fault propagation while on-demand circuit-level analysis is used for fault generation. To overcome simulation overhead of runtime circuit-level analysis, a novel two-phase approach is proposed. The main idea is that circuit characterization can be done before simulation. The result of pre-characterization is used at runtime via a form of look-up to enable gate-level efficiency.

The two-phase methodology is time-efficient but may require high memory unless the look-up tables are carefully optimized. We develop an efficient characterization methodology enabling the two-phase flow. It is based on jointly capturing and compactly storing a set that includes a fault-inducing input, the impacted outputs, and the probabilities of the corresponding fault-events. We enable this by a novel combination of static timing analysis (for finding near-critical paths) and automatic test-pattern generation (ATPG) tool to check sensitization conditions. We develop the fault probability estimation based on workload-specific delay variation due to voltage and temperature distributions. The proposed two-phase methodology can be extended to model fault mechanisms beyond the traditionally studied particle-induced soft errors to include errors due to voltage droops, temperature, and aging.

In summary, the contributions of this research are as follows:

- We develop an efficient fault injector that enables on-demand transistor-accurate fault injection with workload-specific distributional properties to allow gate-level reliability analysis;

- We propose a two-phase approach comprised of a pre-characterization and a runtime fault injection that enables accuracy and efficiency;

- We propose an efficient pre-characterization methodology to jointly capture and compactly store a set that includes a fault-inducing input, the impacted outputs, and the probabilities of the corresponding fault-event.

4.1 Two-Phase Fault Injection Flow

In this chapter, we describe the proposed tool flow comprised of pre-characterization and runtime fault-output generation to achieve both high accuracy and efficiency. The main idea is that circuit-level behavior can be compactly captured through pre-runtime characterization and later be efficiently used at runtime. We also demonstrate that the two-phase flow is correct in the sense that it converges to the system failure rate defined by a direct Monte-Carlo simulation.

The proposed flow estimates the system failure rate r by:

$$r = \frac{\sum_{i=1}^{M} SLE(I_{t_i})}{N} \tag{4.1}$$

in which *N* denotes the total number of cycles simulated and *M* denotes the number of faults injected. Assume those *M* faults are injected at cycles $t_1, t_2 \ldots, t_M$. The $SLE(I)$ represent the system-level failure rate defined on the given input I and can be characterized jointly by the gate-level simulation and corresponding flip-flop (FF) fault patterns FP_j as:

$$SLE(I) = \sum_j u(I, FP_j) P(FP_j) \tag{4.2}$$

Here $u(I, FP)$ is an indicator variable denoting whether there is a system failure for given input I and flipped FF sets FP.

4.1.1 Pre-Characterization

Now we show the innovative pre-characterization flow for calculating the module-level fault probability. The idea behind pre-characterization is that we can compactly capture the most likely fault-producing behaviors and store this description thus avoiding the need for run-time analysis. That is to say, we want to evaluate only module-level fault probabilities that are not negligible.

Clearly, the feasibility of this approach depends on the resulting representation being small: if the pre-characterization table is exponentially large in some circuit parameter then the flow will not work. The key innovation of our work is the problem formulation leading to a compact look-up table.

The challenge is as follows. We seek a replacement to a runtime circuit-level characterization flow, which performs the following tasks: identifies impacted outputs along with their fault probabilities for a given cycle-pair that defines two sets of inputs to a block on subsequent cycles. At first glance, it seems that creating a table that lists possible combinations of inputs is infeasible since we would need to have 2^{2n} rows for an n-input block. However, in reality, we only need inputs that sensitize near-critical paths because timing faults only occur on the near-critical

paths such that the actual number of fault-producing input pairs is a small fraction of the total. First, only a small fraction of all paths leads to a fault for Vdd values above a certain voltage (V_{MIN}). Second, only a small fraction of the near-critical paths are fault-producing paths; a large fraction of paths are false paths, i.e., they are not sensitized by any inputs.

The goal of the pre-characterization is to identify fault-producing input pairs, corresponding path-outputs, and their probabilities. SPICE simulation can identify both the likely fault-producing paths and input pairs at the same time. However, running SPICE simulation to pre-characterize complete modules of realistic size is not feasible. We identify fault-producing paths and inputs separately. We statically find near-critical paths using static timing analysis first and identify inputs that sensitize the paths using ATPG. The ATPG is typically used to identify test patterns in conventional stuck-at-fault and delay-faults testing. The test pattern for the timing fault is a pair of consecutive input vectors that: (1) produces a signal transition at the input node of a path, and (2) sensitizes the path for a signal propagation to the end-point flip-flop of the path. Once a list of paths and a netlist of a circuit are supplied, the ATPG algorithm decides whether the paths can be sensitized by any input and identifies these inputs. If it fails, the path is a false path. The ATPG's complexity is linear in the number of paths; therefore, it is only effective if the number of paths is limited. Even in large designs, such as an OpenSPARC core, the number of paths is manageable. For example, we synthesized all modules of OpenSPARC core with *Synopsys 32nm EDK* standard cell library and found near-critical paths using static timing analysis [63, 64]. The number of paths is less than 1 million and

the number of near-critical paths is approximately 10% if the voltage drop is less than 0.35Vdd.

It is important to capture fault probability in a way that makes it dependent on both circuit and workload characteristics. The workload properties are represented by the probability density function of supply voltage, $f^w(v)$. A timing fault occurs at the capturing flop if the delay of a path exceeds setup time, producing negative slack. We first consider droop-induced faults. We refer to the voltage that induces negative slack on a path as *Critical voltage* (*V**): above the critical voltage, no faults occur because the increased delay is not sufficient to cause a fault. Different paths have different amounts of timing slack and therefore introduce different *V**. Because delay can be modeled as a linear function of voltage with reasonable accuracy, the critical voltage can be computed based on slack values derived from two STA runs at two supply voltages. This is illustrated in Fig. 4.2. For each path, the fault probability is the fraction of voltage distribution below *V**:

$$P_{\pi_j} = \int_{v_{\min}}^{v_j} f^w(v)dv \tag{4.3}$$

where π refers to a path, $f^w(v)$is the supply voltage distribution, and *V** is the critical voltage.

The voltage distribution, $f^w(v)$, is obtained empirically for each workload of interest. We utilize a commercial power network analysis tool, *Synopsys PrimeRail*, that performs static IR drop analysis. The computed Vdd map is predominantly impacted by the parasitic resistance of metal wires constituting the on-chip power

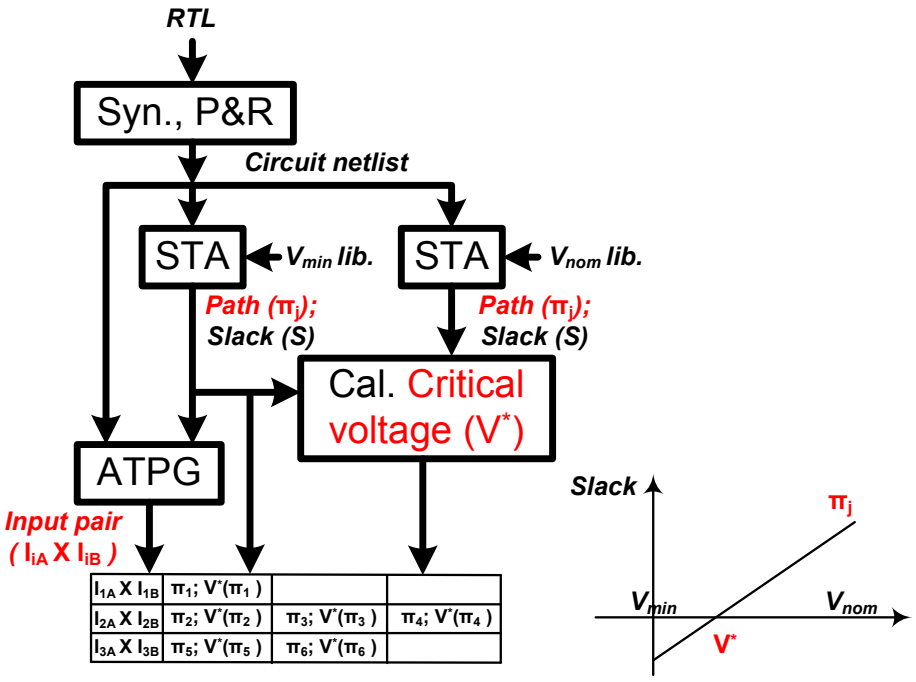
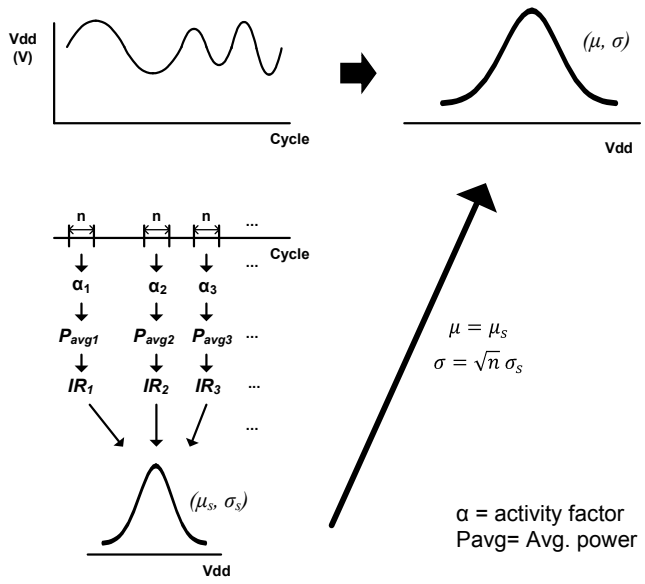


Figure 4.2: Timing and input analysis for fault-producing paths and input pairs that sensitize the paths.

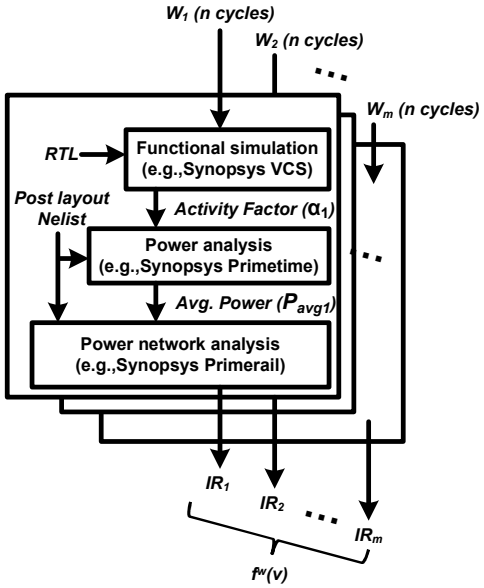
distribution network under a specific average power dissipation. Fig. 4.3 shows our methodology and a tool flow to derive $f^w(v)$. The following steps are done for each workload of interest: (1) A switching activity factor is obtained through analysis of randomly selected cycles. At each cycle, the switching activity is estimated; the average switching activity is averaged over *n* cycles. (2) Average power is computed based on the obtained switching activity using static power analysis (e.g., *Synopsys Primetime*). (3) The power estimate is used by the power network analysis tool operating on a post-layout netlist to produce the IR drop map. By running the flow *m* times, the mean and the standard deviation of the distribution of IR drop over a space of workload-phases is finally computed.

The flow used to characterize temperature-induced faults uses a similar strategy. The flow is shown in Fig. 4.4. It is also built around a linear model of delay on temperature which allows using STA checks at two extreme temperatures to identify *Critical temperature*, a point at which the slack of a path goes negative. The only difference compared to the flow for deriving $f^w(v)$ is that a thermal simulator replaces the power network analysis tool. A thermal simulator uses the average power consumption to estimate on-chip temperature. By running the flow *m* times with a workload, we obtain *m* temperature values at different cycles and ultimately a temperature distribution for the workload. CDF of temperature below the critical temperature can be thought of as fault probability.

The entire pre-characterization tool flow and the resulting look-up table are illustrated in Fig. 4.5 and Table 4.1, respectively. The table consists of input pairs that sensitize near-critical paths, path-output (the end-point flip-flops) of the near-



(a) Methodology



(b) Tool flow

Figure 4.3: Voltage analysis to estimate workload-specific voltage distributions.

Table 4.1: Error profile look-up table (*edb*).

Input Pair	[a)]Path-output 1;Probability	...	Path-output n;Probability
$I_i(0)$, $I_i(1)$	$O_{i,1}$; $P_{i,1}$	...	$O_{i,n}$; $P_{i,n}$

[a)]path-outputs: outputs on end-point flip-flops of near-critical paths

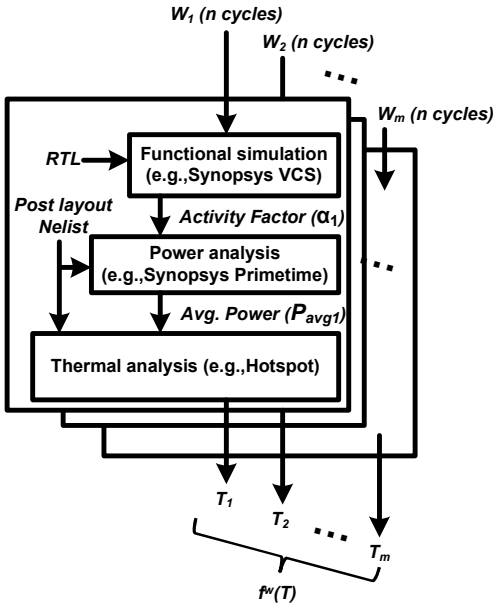


Figure 4.4: Thermal analysis to estimate workload-specific temperature distributions: Tool flow.

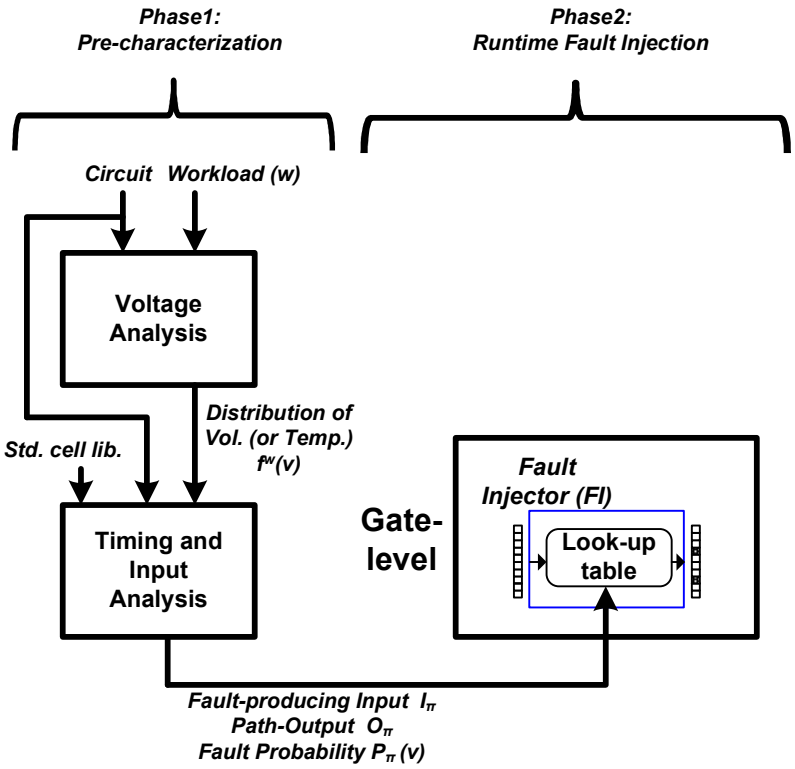


Figure 4.5: Overview of the proposed tool flow.

critical paths, and their probability. The near-critical paths are analyzed by STA, and their probabilities are calculated by Eq. 4.3. The input pairs for the paths are identified by ATPG. Table 4.1 is derived from a table in Fig. 4.2. Paths are replaced with path-outputs of the paths, and critical voltages are replaced with their probabilities using Eq. 4.3.

4.1.2 Runtime Fault Injection

We now describe the runtime fault injector (FI) based on the pre-characterized lookup tables. The fault injection flow is illustrated in Fig. 4.5, Fig. 4.6 and Algorithm 1. The injection process is based on a gate-level simulation, and that is halted at a randomly-determined cycle of interest. The gate-level simulator extracts the inputs and outputs of combinational blocks for the cycle of interest. If a fault is produced, it then traces fault propagation until the fault is either masked or a system failure occurs. Once a pair of inputs (*input(0)*, *input(1)*) is supplied to the FI, the FI checks whether it belongs to the set of input pairs defined as critical and capable of inducing one or more sensitizable paths from the pre-characterized look-up table (*edb*). For such inputs, the tool identifies the corresponding path-outputs and injects faults based on their probabilities as outlined in Algorithm 1. If fault probabilities are small, a large fraction of simulations would not inject any faults, leading to wasted simulation effort. With the pre-characterized look-up table, we could remove the redundant non-injection part in runtime injection whereas the other runtime fault simulators cannot. Because the runtime fault simulators do not know whether the given condition can produce faults or not until the simulation is

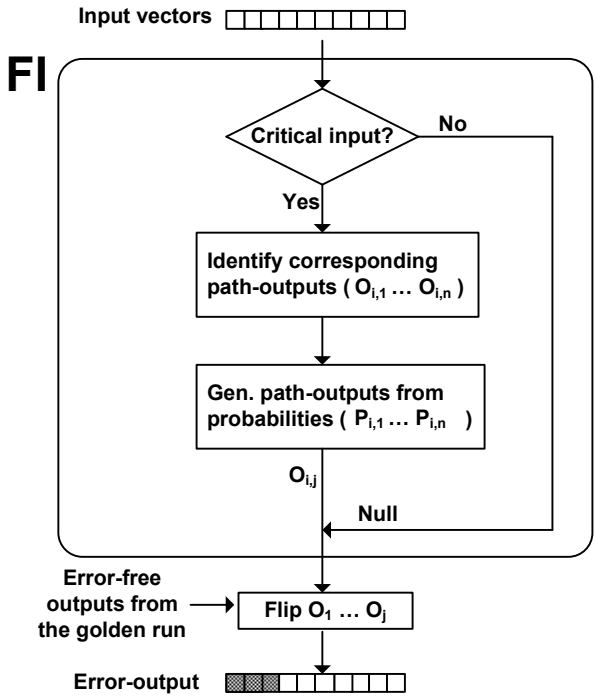


Figure 4.6: Fault injection flow.

finished. Even if the table-based injection can reduce the runtime simulation effort greatly by removing the non-injection part, a system failure probability by the fault injections should be unchanged. Therefore, we describe a mathematical equality of the system failure probability by the proposed fault injection flow with a reference injection flow with direct Monte-Carlo.

We first present the evaluation of the probabilities $P(FP_i)$ of flipped FF set and show the equivalence of the proposed two-phase flow with Direct Monte-Carlo.

The evaluation proceeds as follows. Consider a look-up table where an input pair corresponds to a set of paths and the unique FFs these paths lead to. Let

F_i represent a fault on FF ending path *i*. We can easily (in linear time in the number of paths) find the fault probability of each distinct FF. However, we also need to compute probabilities of multi-FF fault patterns, e.g. $(F_1 = 1, F_2 = 1)$. We use the following argument to *efficiently* compute these set probabilities from individual FF (path) probabilities. Suppose the matched line in the table contains n paths to n outputs $FF_1, FF_2, \ldots FF_n$. Without loss of generality, we can sort those n paths according to their critical voltages in the monotonic increasing manner: $v_1 \leq v_2 \leq \ldots \leq v_n$. Accordingly, their fault probabilities are also monotonic increasing: $p_1 \leq p_2 \leq \ldots \leq p_n$. This means if a fault on path i occurs, faults on superordinate paths $(e.g., i+1, i+2, \ldots, n)$ also occur. Therefore, possible fault patterns on $F_1, F_2, \ldots F_n$ are $(0, 0, \ldots, 0, 0)$,$(0, 0, \ldots, 0, 1)$, $(0, 0, \ldots, 1, 1)$, ..., and $(1, 1, \ldots, 1, 1)$, where $F_i = 1$ means a fault i occurs on $(F_1, F_2, \ldots, F_{n-1}, F_n)$. Let's index the fault patterns (*FP*) as FP_0 to FP_n. The probability of the fault pattern *i* is $P(FP_i) = p_{n+1-i} - p_{n-i}$ as illustrated in Fig. 4.7. In this manner, we can compute probabilities of all patterns.

Recall that our flow estimates total system level failure rate via:

$$r^{'} = \frac{\sum_{i=1}^{M} \sum_{j} u(I_{t_i}, FP_j) P(FP_j)}{N} \tag{4.4}$$

On the other hand, the direct Monte-Carlo flow evaluates the total system level failure rate via:

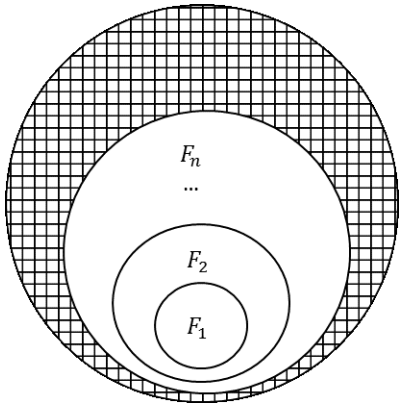


Figure 4.7: Diagram of probabilities of fault patterns.

$$r = \frac{\sum_{i=1}^{N} u(I_i, FP(V_i))}{N} \tag{4.5}$$

Now we show that these two estimates are equivalent in the sense that

$$E[r] = E\left[\frac{\sum_{i=1}^{N} u(I_i, FP(V_i))}{N}\right] = E\left[\frac{\sum_{i=1}^{M}\sum_{j} u(I_{t_i}, FP_j)P(FP_j)}{N}\right] = E[r^{'}] \tag{4.6}$$

in which the expectation is defined over the joint distribution of input I and voltage V.

Before showing the equivalence, we first re-write $r^{'}$ in a form similar to r:

$$r^{'} = \frac{\sum_{i=1}^{M}\sum_{j} u(I_{t_i}, FP_j)P(FP_j)}{N} = \frac{\sum_{i=1}^{N}\sum_{j} u(I_i, FP_j)P(FP_j)}{N} \tag{4.7}$$

This is true because at cycles without fault injection, gate-level simulation never produces system level failures at all: $u(I_i, FP_j) = 0$ when $i \notin \{t_1, t_2, \ldots t_M\}$.

Although voltage V does not directly show up in $r^{'}$, probabilities $P(FP_j)$ are derived based on the voltage distribution so that they are already encoded with the voltage information, as we discussed above. Now both estimates r and $r^{'}$ are similar time averages of functions defined over the joint distribution of input I and voltage *V*. They only differ in the items being summed up. According to the law of large number, $E[r]$ and $E[r^{'}]$ exist and converges to $E[u(I, FP(V))]$ and $E[\sum_j u(I, FP_j)P(FP_j)]$ respectively. Note that those expectations are taken over both input and voltage domain. In particular, for the fixed input I, by the definition of expectation.

$$E_V[u(I, FP(V))] = \sum_j u(I, FP_j)P(FP_j) \tag{4.8}$$

where the expectation is over voltage only and the summation is over all possible fault patterns FP_j's of FF sets. This is identical to the inner sum of $r^{'}$. By taking the expectation over inputs *I* on both sides, we got

$$E[u(I, FP(V))] = E\left[\sum_j u(I, FP_j)P(FP_j)\right] \tag{4.9}$$

which implies our desired result that

Fault Injection (FI)

Inputs: look-up table (*edb*) and input vectors (*input(0), input(1)*)

Output: Path-output (O_j)

1: **procedure** FI

2: sample *input(0), input(1)*

Algorithm 1 Direct Monte-Carlo flow

3: sample Vdd
4: find impacted FFs w. circuit simulator;
5: repeat step 3 and 4 until the impacted FFs are found
6: flip corresponding bits of FFs;

Algorithm 2 The proposed flow

7: read *edb*;
8: **for** *i=0 ; i* $<$ *sizeof(edb(:,0)) ; i* $\leftarrow$ *i+1* **do**
9: **if** *edb(i,0), edb(i,1) == input(0), input(1)* **then**
10: generate path-outputs from O_1 to O_j using P_j;
11: **end if**
12: **end for**
13: flip corresponding bits of path-outputs;
14: **end procedure**

$$E[r] = E[r^{'}] \tag{4.10}$$

The statements do not require independence assumption of *I* and *V*, so it holds for any joint distribution on V and *I*. Fault injections using the direct Monte-Carlo flow and the proposed flow are outlined in Algorithm 1 and 2.

4.2 Experimental Results

We present experimental results of the proposed fault injection flow on a core of OpenSPARC T1 design in this chapter. Four programs – matrix multiply, sort, bzip2, and prime – are selected for the experiments. We first present a pre-characterization result of OpenSPARC T1 core targeting an implementation in a 32nm technology node and show runtime fault injection results using the pre-characterized look-up table. We report computed system failure rates over four programs and runtime overheads by the proposed fault injection. We also compare failures coverages the runtime overheads with the previously best-known gate-level fault injectors.

4.2.1 Pre-Characterization Result of OpenSPARC core

We first synthesized all modules of a core of OpenSPARC T1 and performed the placement and routing with *Synopsys 32nm EDK* standard cell library [64]. The OpenSPARC T1 is an open-source version of UltraSPARC T1 processor. The OpenSARC design is comprised of eight 4-threaded cores and each core has an instruction fetch unit (IFU), an execution unit (EXU), a load/store unit (LSU), and a floating point unit (FFU). The OpenSPARC has more than 30 million gates and 200 million transistors.

We used *Synopsys Design Compiler*, *IC Compiler*, and *Primetime* for synthesizing, placement & routing, and static timing analysis, respectively [65–67]. Synthesizing, placement and routing are performed to target 10% timing guardband for a clock period (0.833 ns) at nominal Vdd of 1.05V. The number of paths and the

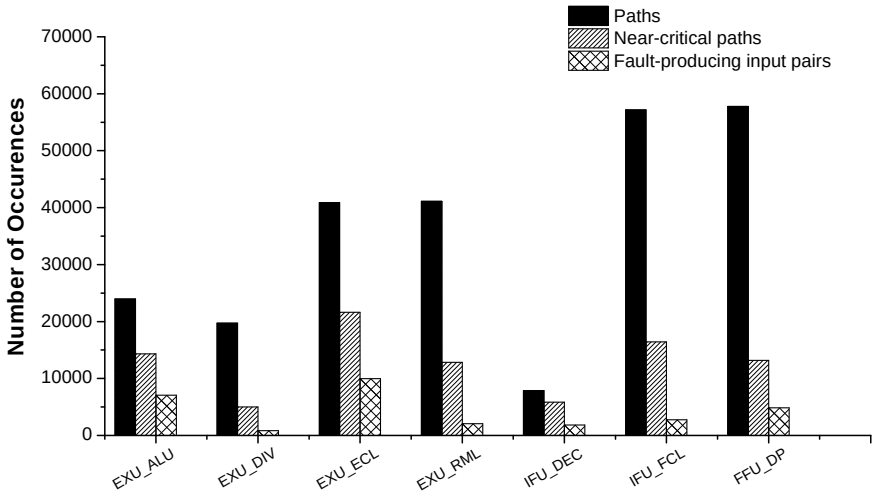


Figure 4.8: Paths, near-critical paths, and input pairs of OpenSPARC core implemented on 32nm technology.

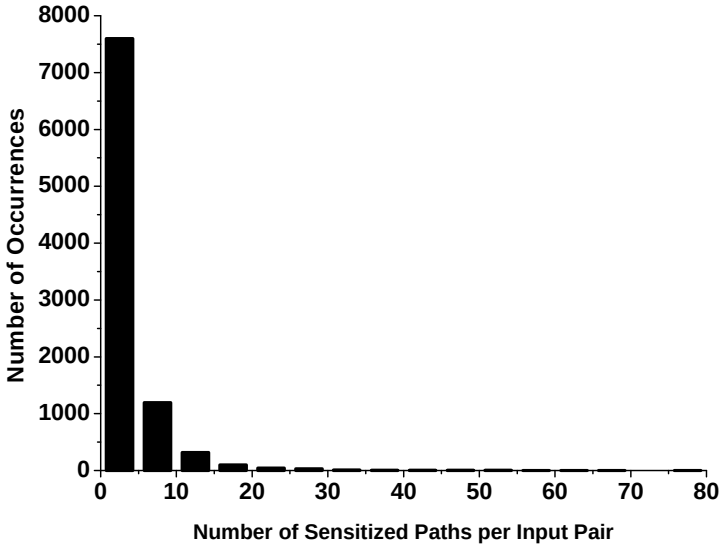


Figure 4.9: The number of paths that are sensitized by the same input pair.

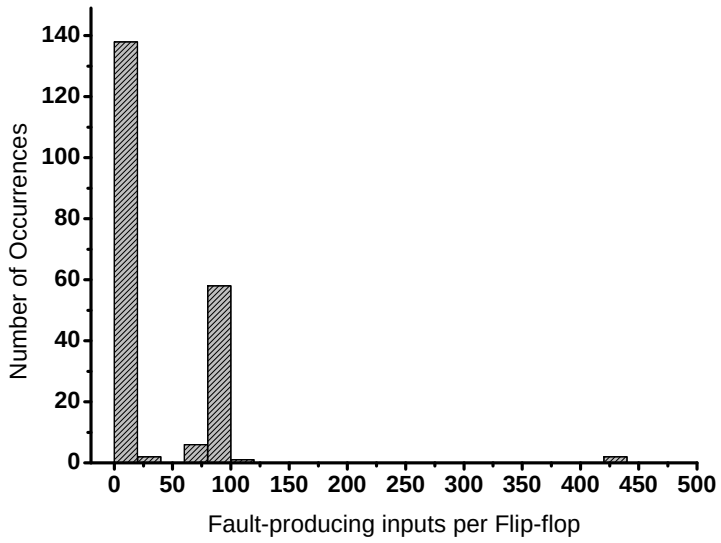


Figure 4.10: The number of fault-producing inputs per FF.

number of near-critical paths are 889,238 are 89,240, respectively. Seven of 233 modules show paths that induce negative slack at the minimum supply voltage. The other modules, which are mostly buffers and repeaters, do not produce near-critical paths. We used *Synopsys Tetramax* for ATPG to find input pairs that sensitize the near-critical paths [68]. A list of near-critical paths and a netlist of a circuit are supplied for the ATPG to identify inputs that sensitize the given paths. Fig. 4.8 shows near-critical paths and input pairs identified by the ATPG. The number of near-critical paths and inputs are 89,240 and 29,357, respectively. The number of paths that are sensitized by the same input varies from 1 to 75 as shown in Fig. 4.9. This means that multiple fault injections can occur at a cycle. Therefore, a random fault injection, which does not consider a correlation of faults, may produce *false injection* as described above. The number of fault-producing inputs per FF also

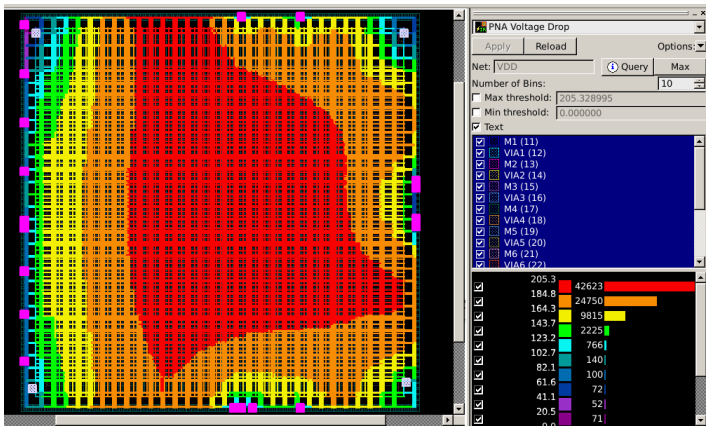


Figure 4.11: Voltage drop map of OpenSPARC core.

varies from 1 to 436 as shown in Fig. 4.10.

We also extracted workload-specific Vdd and temperature distributions for four programs. We first computed activity factors while programs are running on OpenSPARC using *Synopsys VCS*. A switching activity factor is obtained through analysis of randomly selected cycles. At each cycle, the switching activity is estimated and averaged over 100 cycles. 10 average powers are obtained by *Synopsys Primetime* with a post-layout netlist. *Synopsys Primerail* and *Hotspot* are used to obtain IR drop and temperature maps with power consumption information as shown in Fig. 4.11 and Fig. 4.13 [66, 69, 70]. High IR drop and temperature are illustrated in red, and blue shows low IR drop and temperature. By running these tools 10 times with average powers, we obtained workload-specific voltage distributions.

Fig. 4.12 and Fig. 4.14 show voltage and temperature distribution of mod-

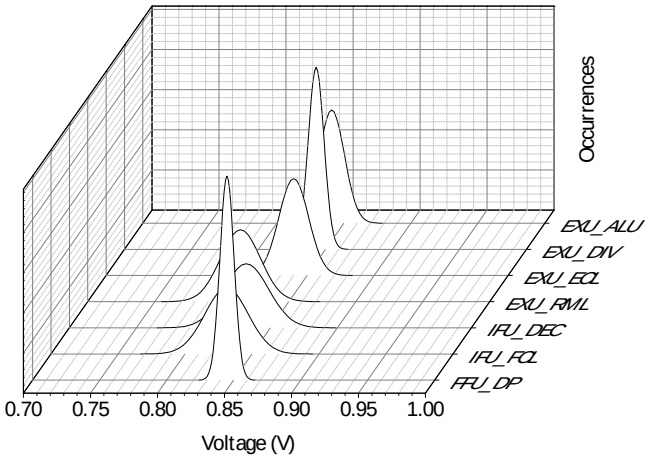


Figure 4.12: Voltage distribution of modules for matrix multiply.

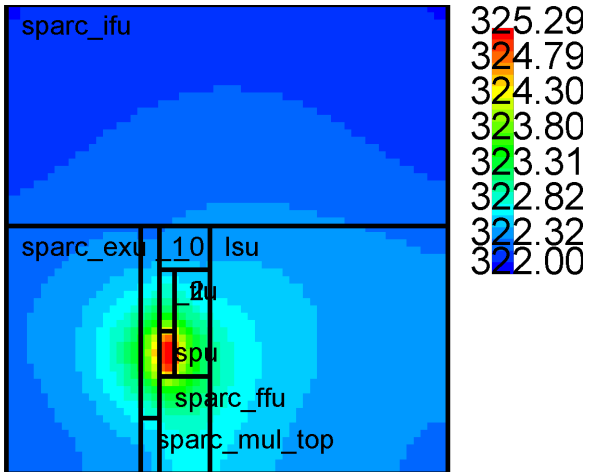


Figure 4.13: Temperature map of OpenSPARC core.

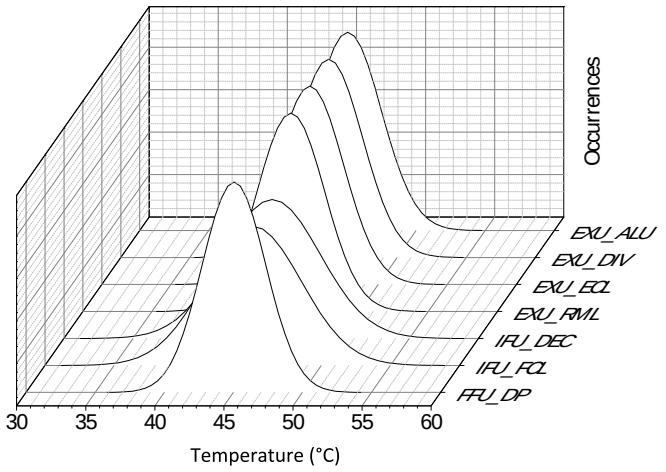


Figure 4.14: Temperature distribution of modules for matrix multiply.

ules for matrix multiply. From the above pre-characterization, we derived a look-up table (*edb*). The table consists of input pairs that sensitize near-critical paths, path-output (the end-point flip-flops) of the near-critical paths, and their probabilities.

4.2.2 Runtime Fault Injection Results for Four Programs

We now present runtime fault injection results using the pre-characterized look-up table. The injection process is based on a gate-level simulation that is halted at a randomly-determined cycle. The gate-level simulator extracts the inputs and outputs of the combinational blocks for the cycle of interest. The fault injector is called and the input vector is examined. If the inputs belong to the set of input pairs contained in the look-up table, the injector identifies the corresponding path-outputs and generates faults based on the probabilities of their occurance. Ultimately, the

Table 4.2: System failure rates for 4 programs.

Program	matrix multiply	sort	bzip2	prime
System failure rate (10^{-7})	0.45	0.82	1.02	0.46

faulty outputs are obtained by flipping the values of the outputs on the golden run. If a fault is produced, the injector traces it to a checkpoint to see whether the fault is masked or a system failure has occurred.

The simulation experiment is as follows. The inputs are sampled 1 billion times and examined with a look-up table. The fault injection flow only continues when the inputs are fault-producing. If the input vector is fault-producing, the simulator traces it to a checkpoint for a 1,000 cycles to assess error masking/system failure. Table 4.2 shows computed system failure rates for four programs. The system failure rate varies from 0.45×10^{-7} to 1.02×10^{-7}. This is because the supplied inputs vary over programs.

We also compare the results produced by the proposed fault simulator with those from a conventional gate-level simulator. To mimic the injection methodology of the conventional gate-level simulator, we first halt gate-level simulation at a randomly-determined cycle of interest and select modules to analyze. After that, flops are selected at random and a one-cycle delay is added to the selected flops. Because a timing fault is not captured by a flop if there is no transition at the input node (D) of the flop, the conventional gate-level fault injector monitors the selected flops to check a transition. Fault values are only injected into the flops if a tran-

sition occurs. After the injection, the analysis continues to determine whether the fault has been masked or a system failure has occurred.

To have a basis for comparison of the two approaches, we use our injector to analyze fault behavior at the same set of cycles. The injector is called to identify the corresponding path-outputs and the faults are injected according to the procedure described above. Note that the key difference is that our flow only injects faults into the flops of near-critical paths, that are truly responsible for faults, while the gate-level simulator injects faults on the randomly selected flops.

Outcomes from the fault injection methodologies are compared to a golden fault-free run. System failures by both fault injection methodologies are categorized as one of the following: detected unrecoverable error (DUE), Output match, silent data corruption (SDC), Hung, or Masked. As shown in Fig. 4.15 the most frequent category is Masked (above 50% of all cases). The second highest category is DUE, followed by output match and SDC. The hung case is not observed in the simulations. The complete summary of the results is described in Table 4.3. We observed the same order of occurrences is observed from both fault injection methodologies (Masked - DUE - Output match - SDC - Hung). This shows that the overall tendency remains unchanged over fault injectors. This tendency is also shown in the other literature [61, 62, 71].

To compare runtime overheads, we continuously injected faults up to 10 million cycles. Simulation time varies over cycles in the proposed fault injectors because the proposed fault injector checks whether the supplied inputs belong to the set of input pairs in the look-up table and stop searching if a corresponding fault is

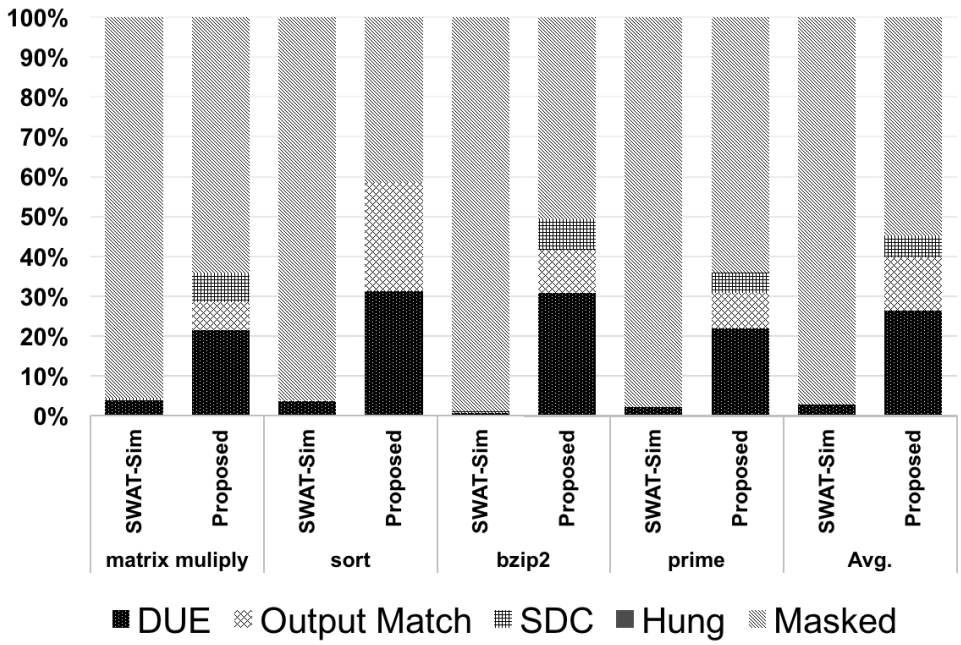


Figure 4.15: System failure coverage by fault injections from two fault injection methodologies.

injected. Fault simulators run on Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz PC, resulting in the runtime of 16.3 minutes (0.1 ms per cycle) while a gate-level simulator is required 6.7 hours (2.4 ms per cycle). The proposed fault injector achieves 24$\times$ runtime reduction compared to a gate-level fault simulation. The cost of the proposed fault injector is a memory usage for the look-up table. All performance comparisons with a gate-level fault simulator are summarized in Table 4.3.

Gate-level fault simulators select FFs to inject faults at random, but faulty cycles are identified by monitoring transitions at the selected flops in the gate-level

Table 4.3: Performance summary and comparison.

		Gate-level sim.	Proposed
Accuracy	Fault injection location	Random	FFs on near-critical paths
	Injection Time (cycle)	At cycle of transitions	At cycle input that sensitizes the path
Efficiency (Runtime overhead)		2.4 ms per cycle	0.1 ms per cycle
Memory Usage		No	Yes

simulators. However, without detailed circuit-level timing analysis, faults can be injected into FFs on non-critical paths (e.g., *false injection*) by the gate-level simulator. Therefore, the proposed fault simulator is more realistic for the fault injection compared to gate-level fault simulators because the accurate circuit-level analysis is performed to identify fault-producing flops and cycles.

4.3 Summary

We present an efficient fault injector that enables on-demand transistor-accurate fault injection with workload-specific distributional properties to allow gate-level reliability analysis. We propose a novel two-phase tool flow to perform the pre-characterization and the runtime fault injection. The proposed simulator ensures circuit-level accuracy by identifying and injecting faults without significant runtime overhead. We compare the proposed fault injector with the best previously known fault injector. Experiments show the overall tendency remains unchanged from two fault injections in terms of the frequency of system failure occurrences. The fault injector reduces runtime by 24$\times$ compared to the best previously known fault injector with gate-level accuracy.

Chapter 5

Conclusions and Future Work

This dissertation investigated two classes of problems that require the use of probabilistic methods and models: (1) Modeling and exploiting stochastic behavior in advanced memory technologies; (2) Probabilistic modeling of faults due to on-chip voltage variation.

This dissertation first investigated the unique physics-level stochasticity of STT-RAM. The write process of STT-RAM is stochastic: specifically, the write time of a bitcell varies significantly. This dissertation developed novel circuit techniques to exploit the stochastic properties of STT-RAM write operation for energy savings by moving away from the worst-case approach to dynamic strategies while maintaining the required low error rate. The first contribution is a variable energy write (VEW) architecture that effectively exploits the wide distribution of write time to greatly reduce energy via a mechanism that checks the instantaneous state of the bitcell and deactivates the write current once the correct value has registered. The second contribution is a multiple attempt write (MAW) strategy that utilizes the asymptotic temporal stochastic independence of repeated switching events to achieve a dramatic reduction in energy. Analysis indicates that the proposed write architectures reduce the write energy with no significant area overhead.

This dissertation then addressed the problem of probabilistic modeling of faults due to on-chip voltage variations. The power supply voltage variation can increase gate delay, resulting in timing faults on near-critical paths. Developing an accurate fault model and injection tool that generates and propagates faults from circuit- to gate-level is important for accurately predicting the resulting system failures. This dissertation developed a hybrid gate-level fault simulator capable of capturing fault correlations through circuit-level analysis. The gate-level simulator is used to check fault propagation while on-demand circuit-level analysis is used for fault generation. To overcome simulation overhead of runtime circuit-level analysis, a novel two-phase approach is proposed. The main idea is that circuit characterization can be done before simulation. The result of pre-characterization is used at runtime via a form of look-up to enable gate-level efficiency. Analysis indicates the proposed fault modeling and injection flow reduces runtime overhead by 24$\times$ compared to a fast gate-level fault simulator while having circuit level accuracy.

Although the results of the dissertation show the effectiveness of the proposed STT-RAM architectures, several limitations remain. One obvious limitation is that the proposed STT-RAM architectures are evaluated through simulation. Fabrication would provide a more reliable way to evaluate the effectiveness of the proposed write architectures. Several foundries plan to start offering STT-RAM as an alternative for embedded flash. Future research should focus on implementing the proposed probabilistic write approaches once the STT-RAM technologies by the foundries become available to designers.

In this dissertation, fault injection has studied at circuit- to gate-levels. Fu-

ture research should focus on full system analysis covering also μ*Arch*- to software levels. Such an approach will provide a better understanding of how faults propagate through an entire system, offering direct quantitative insight into the resilience characteristics of systems in realistic failure environments.

Appendix

Appendix 1

List of publications

1. **Jaeyoung Park** and Michael Orshansky, "Abnormal ESD failure mode with low-voltage turn-on phenomenon of LDMOS output driver," *in Proc. of IEEE International Reliability Physics Symposium*, Anaheim, CA, 2012.

2. Tianhao Zheng, **Jaeyoung Park**, Michael Orshansky, and Mattan Erez, "Variable-Energy Write STTRAM Architecture with Bit-Wise Write-Completion Monitoring," *in Proc. of IEEE International Symposium on Low Power Electronics and Design*, pp. 229 - 234, 2013.

3. **Jaeyoung Park**, Tianhao Zheng, Mattan Erez, and Michael Orshansky, "Variation-Tolerant Write Completion Circuit for Variable-Energy Write STT-RAM Architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 99, pp. 360 - 367, 2015.

4. Xiaodan Xi, Meng Li, **Jaeyoung Park**, and Michael Orshansky, "Design Optimization of a Strong PUF Based on Nonlinearity of MOSFET Subthreshold Operation," *TECHCON 2015*, Austin, TX, 2015

5. Bongjun Kim, **Jaeyoung Park**, Michael Geier, Mark C. Hersam, and Ananth Dodabalapur, "Voltage-Controlled Ring Oscillators Based on Inkjet Printed Carbon Nanotubes and Zinc Tin Oxide," *Voltage-Controlled Ring Oscillators Based on Inkjet Printed Carbon Nanotubes and Zinc Tin Oxide*, no. 22, pp. 12009 - 12014, 2015.

6. **Jaeyoung Park** and Michael Orshansky, "Multiple Attempt Write Strategy for Low Energy STT-RAM," *in Proc. of Great Lakes Symposium on VLSI*, Boston, MA, 2016.

7. **Jaeyoung Park**, "Area-efficient STT/CMOS non-volatile flip-flop," *in Proc. of IEEE International Symposium on Circuits and Systems*, Baltimore, MD, 2017

8. **Jaeyoung Park** and Young Y. Yim, "Two-phase Read Strategy for Low Energy Variation-tolerant STT-RAM," will be published in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.

Bibliography

[1] Yue Zhang, Weisheng Zhao, Yahya Lakys, J Klein, Joo-Von Kim, Dafiné Ravelosona, and Claude Chappert. Compact modeling of perpendicular-anisotropy cofeb/mgo magnetic tunnel junctions. *IEEE Trans. Electron Devices*, 59(3):819–826, 2012.

[2] Woojin Kim, JH Jeong, Y Kim, WC Lim, JH Kim, JH Park, HJ Shin, YS Park, KS Kim, SH Park, et al. Extended scalability of perpendicular stt-mram towards sub-20nm mtj node. In *IEDM*, 2011.

[3] Wei Zhao and Yu Cao. Predictive technology model for nano-cmos design exploration. *ACM J. on Emerging Techn. in Comp. Sys. (JETC)*, 3(1):1, 2007.

[4] ITRS. International technology roadmap for semiconductors. URL: http://www.itrs.net/Links/2012ITRS/Home2012.htm, 2012.

[5] Zhenyu Sun, Xiuyuan Bi, and Hai Li. Process variation aware data management for stt-ram cache design. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 179–184. ACM, 2012.

[6] Xingsheng Wang, Andrew R Brown, Binjie Cheng, and Asen Asenov. Statistical variability and reliability in nanoscale finfets. In *Electron Devices Meeting (IEDM), 2011 IEEE International*, pages 5–4. IEEE, 2011.

[7] H Sato, T Yamamoto, M Yamanouchi, S Ikeda, S Fukami, K Kinoshita, F Matsukura, N Kasai, and H Ohno. Comprehensive study of cofeb-mgo magnetic tunnel junction characteristics with single-and double-interface scaling down to 1x nm. *IEDM Tech. Digest*, 3:1–3, 2013.

[8] Xiuyuan Bi, Zhenyu Sun, Hai Li, and Wenqing Wu. Probabilistic design methodology to improve run-time stability and performance of stt-ram caches. In *ICCAD*, 2012.

[9] M Hosomi, H Yamagishi, T Yamamoto, K Bessho, Y Higo, K Yamane, H Yamada, M Shoji, H Hachino, C Fukumoto, et al. A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-ram. In *IEDM*, 2005.

[10] H. Li et al. An overview of non-volatile memory technology and the implication for tools and architectures. In *DATE*, 2009.

[11] Yiran Chen, Hai Li, Xiaobin Wang, Wenzhong Zhu, Wei Xu, and Tong Zhang. A 130 nm 1.2 v/3.3 v 16 kb spin-transfer torque random access memory with nondestructive self-reference sensing scheme. *Solid-State Circuits, IEEE Journal of*, 47(2):560–573, 2012.

[12] Gordon E Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff. *IEEE solid-state circuits society newsletter*, 20(3):33–35, 2006.

[13] Semiconductor Industry Association et al. International technology roadmap for semiconductors (itrs), 2014 edition. 2014.

[14] Jian-Gang Zhu. Magnetoresistive random access memory: The path to competitiveness and scalability. *Proceedings of the IEEE*, 96(11):1786–1798, 2008.

[15] DC Worledge, G Hu, PL Trouilloud, DW Abraham, S Brown, MC Gaidis, J Nowak, EJ O'Sullivan, RP Robertazzi, JZ Sun, et al. Switching distributions and write reliability of perpendicular spin torque mram. In *IEDM*, 2010.

[16] Nikolaos Strikos, Vasileios Kontorinis, Xiangyu Dong, Houman Homayoun, and Dean Tullsen. Low-current probabilistic writes for power-efficient stt-ram caches. In *Computer Design (ICCD), 2013 IEEE 31st International Conference on*, pages 511–514. IEEE, 2013.

[17] Seungbum Hong, Orlando Auciello, and Dirk Wouters. *Emerging non-volatile memories*. Springer, 2014.

[18] Anurag Nigam, Clinton W Smullen, Vidyabhushan Mohan, Eugene Chen, Sudhanva Gurumurthi, and Mircea R Stan. Delivering on the promise of universal memory for spin-transfer torque ram (stt-ram). In *Low Power Electron-*

ics and Design (ISLPED) 2011 International Symposium on, pages 121–126. IEEE, 2011.

[19] Richard F Freitas and Winfried W Wilcke. Storage-class memory: The next storage system technology. *IBM Journal of Research and Development*, 52 (4.5):439–447, 2008.

[20] Arijit Raychowdhury, Dinesh Somasekhar, Tanay Karnik, and Vivek De. Design space and scalability exploration of 1t-1stt mtj memory arrays in the presence of variability and disturbances. In *Electron Devices Meeting (IEDM), 2009 IEEE International*, pages 1–4. IEEE, 2009.

[21] Zhitao Diao, Zhanjie Li, Shengyuang Wang, Yunfei Ding, Alex Panchula, Eugene Chen, Lien-Chang Wang, and Yiming Huai. Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *Journal of Physics: Condensed Matter*, 19(16):165209, 2007.

[22] Xiaobin Wang, Yuankai Zheng, Haiwen Xi, and Dimitar Dimitrov. Thermal fluctuation effects on spin torque induced switching: Mean and variations. *Journal of Applied Physics*, 103(3):034507–034507, 2008.

[23] AV Khvalkovskiy, D Apalkov, S Watts, R Chepulskii, RS Beach, A Ong, X Tang, A Driskill-Smith, WH Butler, PB Visscher, et al. Basic principles of stt-mram cell operation in memory arrays. *Journal of Physics D: Applied Physics*, 46(7):74001–74020, 2013.

[24] Andrew B Kahng. Design-based equivalent scaling to the rescue of moores law.

[25] Ping Zhou, Bo Zhao, Jun Yang, and Youtao Zhang. Energy reduction for stt-ram using early write termination. In *ICCAD*, 2009.

[26] Daisuke Suzuki, Masanori Natsui, Akira Mochizuki, and Takahiro Hanyu. Cost-efficient self-terminated write driver for spin-transfer-torque ram and logic. *Magnetics, IEEE Transactions on*, 50(11):1–4, 2014.

[27] Brent Keeth, R Jacob Baker, Brian Johnson, and Feng Lin. *DRAM Circuit Design: Fundamental and High-Speed Topics*, volume 13. Wiley-IEEE Press, 2007.

[28] Noboru Sakimura, Tadahiko Sugibayashi, Ryusuke Nebashi, Hiroaki Honjo, Shinsaku Saito, Yuko Kato, and Naoki Kasai. A 250-mhz 1-mbit embedded mram macro using 2t1mtj cell with bitline separation and half-pitch shift architecture. In *Solid-State Circuits Conference, 2007. ASSCC'07. IEEE Asian*, pages 216–219. IEEE, 2007.

[29] Kaivalya M Dixit. The spec benchmarks. *Parallel computing*, 17(10):1195–1209, 1991.

[30] John L Henning. Spec cpu2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):1–17, 2006.

[31] Denny D Tang and Yuan-Jen Lee. *Magnetic memory: fundamentals and technology*. Cambridge University Press, 2010.

[32] Chi-Keung Luk, Robert Cohn, Robert Muth, Harish Patil, Artur Klauser, Geoff Lowney, Steven Wallace, Vijay Janapa Reddi, and Kim Hazelwood. Pin: building customized program analysis tools with dynamic instrumentation. In *PLDI*, 2005.

[33] M. Donahue and DG Porter. Oommf user guide, version 1.0, interagency report nistir 6376, national institute of standard and technology, gaithersburg, md, 1999. *URL: http://math. nist. gov/oommf*, 1999.

[34] T. Gilbert et al. Anomalous rotational damping in ferromagnetic sheets. In *Conference on Magnetism and Magnetic Materials*, 1955.

[35] L. Meares et al. *Simulating with SPICE (Simulation Program with Integrated Circuit Emphasis)*. Intusoft, 1988.

[36] A. Papoulis et al. *Probability, random variables, and stochastic processes*. McGraw-Hill Education, 2002.

[37] B. Govindarajalu. *Computer Architecture and Organization: Design Principles and Applications*. McGraw-Hill, 2004. ISBN 9780070532366.

[38] G. Sun et al. A novel architecture of the 3d stacked mram l2 cache for cmps. In *HPCA*, 2009.

[39] Tianhao Zheng, Jaeyoung Park, Michael Orshansky, and Mattan Erez. Variable-energy write stt-ram architecture with bit-wise write-completion monitoring. In *in IEEE International Symposium Low Power Electronics and Design (ISLPED)*, pages 229–234, 2013.

[40] Jaeyoung Park, Tianhao Zheng, Mattan Erez, and Michael Orshansky. Variation-tolerant write completion circuit for variable-energy write stt-ram architecture. *IEEE Transactions on Very Large Scale Integration (VLSI) systmes*, 51(99):360–367, 2015.

[41] Sang-Koog Kim. Micromagnetic computer simulations of spin waves in nanometre-scale patterned magnetic elements. *Journal of Physics D: Applied Physics*, 43(26):264004, 2010.

[42] Saeed Moaveni. *Finite element analysis theory and application with ANSYS, 3/e*. Pearson Education India, 2011.

[43] Subir Sachdev. Handbook of magnetism and advanced magnetic materials. 2006.

[44] WF Van Gunsteren and HJC Berendsen. Algorithms for brownian dynamics. *Molecular Physics*, 45(3):637–647, 1982.

[45] Grazvydas Ziemys, Stephan Breitkreutz-v. Gamm, Gyorgy Csaba, Doris Schmitt-Landsiedel, and Markus Becherer. Experiment-based thermal micromagnetic simulations of the magnetization reversal for ns-range clocked nanomagnetic logic. *AIP Advances*, 7(5):056625, 2017.

[46] T. L. Gilbert. A Lagrangian formulation of the gyromagnetic equation of the magnetization field. *Phys. Rev.*, 100:1243, 1955.

[47] L. Landau and E. Lifshitz. On the theory of the dispersion of magnetic permeability in ferromagnetic bodies. *Physik. Z. Sowjetunion*, 8:153–169, 1935.

[48] V Rajaraman. Ieee standard for floating point numbers. *Resonance*, 21(1): 11–30, 2016.

[49] Hari Cherupalli, Rakesh Kumar, and John Sartori. Exploiting dynamic timing slack for energy efficiency in ultra-low-power embedded systems. In *ACM SIGARCH Computer Architecture News*, volume 44, pages 671–681. IEEE Press, 2016.

[50] Ulf Schlichtmann, Veit B Kleeberger, Jacob A Abraham, Adrian Evans, Christina Gimmler-Dumon, Michael Glas, Andreas Herkersdorf, Sani R Nassif, and Norbert Wehn. Connecting different worldstechnology abstraction for reliability-aware design and test. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–8. IEEE, 2014.

[51] Joseph Sloan, David Kesler, Rakesh Kumar, and Ali Rahimi. A numerical optimization-based methodology for application robustification: Transforming applications for error tolerance. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 161–170. IEEE, 2010.

[52] Keng L Wong, Tawfik Rahal-Arabi, Matthew Ma, and Greg Taylor. Enhancing microprocessor immunity to power supply noise with clock-data compensation. *IEEE Journal of Solid-State Circuits*, 41(4):749–758, 2006.

[53] Vijay Janapa Reddi and David Brooks. Resilient architectures via collaborative design: Maximizing commodity processor performance in the presence of variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(10):1429–1445, 2011.

[54] Howard H Chen, J Scott Neely, Michael F Wang, and Gricel Co. On-chip decoupling capacitor optimization for noise and leakage reduction. In *Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings. 16th Symposium on*, pages 251–255. IEEE, 2003.

[55] Tasreen Charania, Ajoy Opal, and Manoj Sachdev. Analysis and design of on-chip decoupling capacitors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(4):648–658, 2013.

[56] SK Nithin, Gowrysankar Shanmugam, and Sreeram Chandrasekar. Dynamic voltage (ir) drop analysis and design closure: Issues and challenges. In *Quality*

Electronic Design (ISQED), 2010 11th International Symposium on, pages 611–617. IEEE, 2010.

[57] Michał Rewieński. A perspective on fast-spice simulation technology. In *Simulation and Verification of Electronic and Biological Systems*, pages 23–42. Springer, 2011.

[58] Premkishore Shivakumar, Michael Kistler, Stephen W Keckler, Doug Burger, and Lorenzo Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on*, pages 389–398. IEEE, 2002.

[59] Joseph Sloan, David Kesler, Rakesh Kumar, and Ali Rahimi. A numerical optimization-based methodology for application robustification.

[60] Raghuraman Balasubramanian and Karthikeyan Sankaralingam. Understanding the impact of gate-level physical reliability effects on whole program execution. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, pages 60–71. IEEE, 2014.

[61] Andrea Pellegrini, Robert Smolinski, Lei Chen, Xin Fu, Siva Kumar Sastry Hari, Junhao Jiang, Sarita V Adve, Todd Austin, and Valeria Bertacco. Crashtest'ing swat: accurate, gate-level evaluation of symptom-based resiliency solutions. In *Proceedings of the conference on design, automation and test in Europe*, pages 1106–1109. EDA Consortium, 2012.

[62] Man-Lap Li, Pradeep Ramachandran, Ulya R Karpuzcu, Siva Kumar Sastry Hari, and Sarita V Adve. Accurate microarchitecture-level fault modeling for studying hardware faults. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pages 105–116. IEEE, 2009.

[63] Sun Microsystem. Opensparc t1 microarchitecture specification. 2009.

[64] R Goldman, K Bartleson, T Wood, K Kranen, V Melikyan, and E Babayan. 32/28nm educational design kit: Capabilities, deployment and future. In *Microelectronics and Electronics (PrimeAsia), 2013 IEEE Asia Pacific Conference on Postgraduate Research in*, pages 284–288. IEEE, 2013.

[65] Design Compiler, R User, and Modeling Guide. Synopsys. *Inc., see http://www. synopsys. com*, 2000.

[66] IC Synopsys. Compiler user guide, 2013.

[67] PX PrimeTime. User guide version c-2009.06. *Synopsys, June*, 2009.

[68] Synopsys User Manual. Tetramax atpg user guide. *Version X-2005.09*, pages 249–264, 2005.

[69] PrimeRail User Guide. Synopsys inc, 2013.

[70] Wei Huang, Shougata Ghosh, Karthik Sankaranarayanan, Kevin Skadron, and Mircea R Stan. Hotspot: Thermal modeling for cmos vlsi systems. *IEEE Transactions on Component Packaging and Manufacturing Technology*, pages 200–205, 2005.

[71] Hyungmin Cho, Shahrzad Mirkhani, Chen-Yong Cher, Jacob A Abraham, and Subhasish Mitra. Quantitative evaluation of soft error injection techniques for robust system design. In *Proceedings of the 50th Annual Design Automation Conference*, page 101. ACM, 2013.

Vita

Jaeyoung Park received his B.S. and M.S. degrees from Hanyang University in 2005 and 2008, respectively. His master research topic was on electro static discharge (ESD) circuit design of high voltage and RF devices. He worked as an ESD engineer for Dongbu Hitek for 3 years from 2008. He joined the Electrical and Computer Engineering department in University of Texas at Austin as a Ph.D. student in fall 2011. He is now working for Qualcomm Technologies, Inc.

Permanent address: jypark22@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.