Waseda University Doctoral Dissertation

Study on Spin-Transfer Torque Magnetic Tunnel Junction Modeling for Magnetic Memories and Spiking Neural Networks

Haoyan LIU

Graduate School of Information, Production and Systems
Waseda University

October 2023

Abstract

Over the past decades, several emerging memory devices have been introduced with the aim of enhancing circuit performance. Spin-transfer torque magnetic tunnel junction (STT-MTJ) is one of the most promising emerging memories and has found widespread applications because of its excellent physical properties, scalability potential and compatibility with CMOS technology.

To leverage the potential of STT-MTJ, it is necessary to develop circuit simulation models within electronic design automation (EDA) tools. For different types of applications, there are varying requirements for STT-MTJ simulation models. These requirements encompass various aspects such as model accuracy, CPU time, memory requirement and information on magnetization dynamics. The primary objective of this article is to develop appropriate circuit simulation models for STT-MTJ tailored to different types of applications. These models will be used to simulate various applications and validate their utility.

For STT-MTJ, the stochastic Landau-Lifshitz-Gilbert-Slonczewski (s-LLGS) equation provides the rigorous description of the dynamics in magnetization. However, the model based on solving the equation leads to long simulation times and high memory requirements since it requires calculating a random Langevin field at each timestep and using a smaller timestep for accurate results. The s-LLGS equation based model costs an unacceptable CPU time for large-scale circuits simulations such as spin-transfer torque magnetoresistive random-access memory (STT-MRAM). Under this situation, a circuit simulation model based on switching probability which includes several equations to describe the switching characteristics for different current conditions is proposed. Compared with the s-LLGS equation based model, it significantly reduces simulation time. Unlike other circuit simulation models based on switching probability, the proposed model addresses a limitation of that there is no physical model for a certain range of injection current by introducing a framework that connects existing equations which were derived from two extreme physical condition.

However, the switching probability based model cannot be used for the applications which need the information on magnetization change such as spiking neural networks (SNNs). For this requirement, a simulation model based on the Fokker-Planck equation (FPE) which is the

master equation of the s-LLGS equation is proposed. The FPE is solved by the finite difference method (FDM) and the impact of the solvers on computational efficiency and accuracy is analyzed. A framework is proposed which traces dynamics of a particular STT-MTJ's theta between the magnetic moments of the free and the pinned layers and achieves performance comparable to that of the s-LLGS equation is achieved. The proposed model represents the first model for STT-MTJ based on the FPE to be implemented into a defacto-standard circuit simulator HSPICE.

The two proposed models show excellent accuracies when compared with experimental data and they demonstrate high CPU time efficiency in comparison with other models for large scale circuit simulations. The models were applied to two different applications to verify their effectiveness in practical use. Specifically, these applications are STT-MRAM and a spiking neural network (SNN) that is proposed by the author. The two applications utilize different characteristics of STT-MTJ. The results show that the proposed models have been applied well in different types of circuits. Furthermore, the simulation results validate that the proposed SNN implements an automatic reset function on the neuron and can be used in recognition tasks. It achieves a low-power SNN with an energy consumption of 0.23pJ per synaptic operation, which is a significant improvement compared with existing designs.

The dissertation contains seven chapters as follows:

Chapter 1 provides an introduction to the underlying principles and historical development of MTJ, as well as an overview of its basic physical properties. This chapter also introduces an overview of the s-LLGS model and its limitations in circuit simulation.

Chapter 2 describes the switching probability models which includes the Sun model and Néel-Brown model for the injection current much larger and smaller than the critical current, respectively. A problem that there is no physical model around the critical current is explained. A mathematical framework is proposed to calculate the switching characteristics in the vicinity of the critical current based on two existing models. The model for the resistance and Joule heating effect are also shown. The model is compared with experimental data and the results validate the effectiveness.

In Chapter 3, we present a circuit simulation model based on the FPE that utilizes the finite difference method (FDM) for solving the FPE. We analyze various forms of FDM and demonstrate that the implicit form of FDM offers the best trade-off between computation time and accuracy. We also propose a framework for extracting theta information of the magnetization from the probability density distribution of the theta. The simulation results obtained from the FPE based model are in good agreement with both experimental data and the results of the s-LLGS model.

Chapter 4 shows two specific applications that are applied by the two different circuit simulation models. The first application utilizes the resistance characteristics of STT-MTJ and focuses on the simulation of 4T2MTJ STT-MRAM. The both models were evaluated for their performance in simulating STT-MRAM, and the results demonstrate their effectiveness. The second application demonstrates the use of theta information in STT-MTJs to design an SNN with STT-MTJ-based auto-reset neurons and synapses. The simulation results have confirmed the effectiveness of the proposed model based on the FPE for analyzing the circuit operations in the SNN. Additionally, the simulation confirms the efficient implementation of the proposed SNN neuron and the overall SNN system has achieved a low energy which can be reduced to 0.23pJ/SOP. Overall, these specific applications provide further evidence of the practical applicability and versatility of the proposed circuit simulation models.

Chapter 5 shows a comparison between proposed models and existing circuit simulation models, examining their basic characteristics and CPU time requirements. The results indicate that the proposed switching probability based model is unique in solving the intermediate regime problem and accommodating time-varying currents. The proposed FPE model is the first circuit simulation model based on the FPE equation. The comparison of CPU time across circuits of varying sizes and complexities indicates that the switching probability based model requires the least amount of CPU time. Moreover, the FPE based model can provide the same information as the s-LLGS equation based model but with only 3% of the CPU time. At last Chapter 5 concludes the dissertation.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

CMOS	Complementary Metal-Oxide-Semiconductor
MTJ	Magnetic Tunnel Junction
TMR	Tunneling Magnetoresistance
$\mathbf{R_p}$	Low Resistance State of MTJ
$\mathbf{R_{ap}}$	High Resistance State of MTJ
$\mathbf{D(+,E_f)}$	Density of Majority Spin States at The Fermi Energy
$\mathbf{D(-,E_f)}$	Density of Minority Spin States at The Fermi Energy
TMRR	Tunneling Magnetoresistance ratio
Pol	Spin Polarizations
MRAM	Magnetoresistive Random-Access Memory
STT-MTJ	Spin-Transfer Torque Magnetic Tunnel Junction
PMA	Perpendicular Magnetic Anisotropy
VCMA	Voltage-Controlled Magnetic Anisotropy
SOT	Spin-Orbit Torque
SHE	Spin Hall Effect
$\mathbf{I_{cp}}$	Critical Current for Parallel to Anti-parallel Switching in MTJ
$\mathbf{I_{cap}}$	Critical Current for Anti-parallel to Parallel Switching in MTJ
$\boldsymbol{\alpha}$	Gilbert Damping Coefficient
$\boldsymbol{\gamma}$	Gyromagnetic Ratio
e	Electronic Charge
$\boldsymbol{\mu_B}$	Bohr Magneton
$\boldsymbol{\eta_p}$	Spin polarization for Parallel State
$\boldsymbol{\eta_{ap}}$	Spin polarization for Anti-parallel State
$\boldsymbol{\xi}$	Thermal Stability Factor
E	Energy Barrier
$\mathbf{k_B}$	Boltzmann Constant
T	temperature in Kelvin
DNN	Deep Neural Network
SNN	Spiking Neural Networks
BEOL	Back-end-of-the-line
EDA	Electronic Design Automation

s-LLGS	Stochastic Landau-Lifshitz-Gilbert-Slonczewski
$\overrightarrow{\boldsymbol{M}}$	Magnetization
$\overrightarrow{\boldsymbol{H_{eff}}}$	Effective Field
$\boldsymbol{M_S}$	Saturation Magnetization
$\hbar$	Reduced Planck Constant
Vol	Volume of the Free Layer
$\vec{\boldsymbol{p}}$	Unit Vector in the Direction of Magnetization in the Pinned Layer
$\boldsymbol{\Delta t}$	Simulation Timestep
$\mathbf{X_i(t)}$	Zero Mean Standard Gaussian Distribution for the x, y and z axes
$\mathbf{T_c}$	Curie Temperature
$\boldsymbol{\rho}$	Probability Distribution
$\mathbf{H_k}$	Anisotropy Field
$\mathbf{H_{ex}}$	External Field
$\mathbf{H_D}$	Demagnetization Field
$\mathbf{H_{app}}$	Applied Field
K	Anisotropy Coefficient
$\mathbf{t_{fl}}$	Thickness of the Free Layer
$\boldsymbol{\mu_0}$	Vacuum Permeability
$\mathbf{N_i}$	Demagnetization Factor for the x, y and z axes
WER	Write Error Rate
P	Switching Probability
$\boldsymbol{\tau_0}$	Inverse of Attempt Frequency
$\boldsymbol{\tau_D}$	Relaxation Time
$\mathbf{t_{ox}}$	Thickness of the Oxide Layer
F	Fitting Factor
Area	Area of the MTJ Surface
$\boldsymbol{\varphi_B}$	Energy Barrier Height of the Oxide Layer
$\boldsymbol{T_{heat}}$	Temperature During the Heating Period
$\boldsymbol{D_{heat}}$	Time Duration of the Heating Period
$\boldsymbol{V_{mtj}}$	MTJ voltage
$\boldsymbol{J_{mtj}}$	Current Density of MTJ

$\boldsymbol{C_v}$	Heat Capacity Per Unit Volume
$\boldsymbol{t_{MTJ}}$	Thickness of MTJ
$\boldsymbol{T_{cool}}$	Temperature During the Cooling Period
$\boldsymbol{D_{cool}}$	Time Duration of the Cooling Period
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
BL	Bit Line
FPE	Fokker-Planck Equation
G	Grid Number
PL	Power Line
WL	Word Line
LIF	Leaky Integrate-and-fire
PTM	Predictive Technology Model
MNIST	Modified National Institute of Standards and Technology
BSNN	Binarized Spiking Neural Network
GND	Ground
u	Membrane Potential
$\boldsymbol{\tau_m}$	The Time Constant of Neuron
r	Firing Rate
n	Probability Number of Spikes During a Time Interval of Length Δt
STDP	Spiking-timing-dependent plasticity
$\boldsymbol{A^+}$	Learning Rate of Potentiation
$\boldsymbol{A^-}$	Learning Rate of Depression
$\boldsymbol{\tau_+}$	Time Constant of Potentiation
$\boldsymbol{\tau_-}$	Time Constant of Depression
SOP	Synaptic Operation

Chapter 1

Introduction

1.1 Magnetic tunnel junction history

The complementary metal-oxide-semiconductor (CMOS) technology is approaching the physical limitation these years. In this situation, emerging memory devices are used to continuously improve the performance. Magnetic tunnel junction (MTJ) which is based on tunneling magnetoresistance (TMR) becomes an important candidate for the next generation memory device. The structure of MTJ is shown in Fig. 1.1. [1] The device consists of three layers: one oxide layer (MgO) and two ferromagnetic layers (CoFeB). The directions of magnetizations in the two ferromagnetic layers are constant (pinned layer) or variable (free layer). When the magnetizations of the two ferromagnetic layers change relative direction, the TMR effect causes a significant change in the electrical tunneling resistance of a thin oxide layer (with a typical thickness of several nanometers) sandwiched between them. [2] When the magnetizations of the free layer and pinned layer in an MTJ are parallel (P state), the device is in a low resistance state (R_p). Conversely, when the magnetizations are anti-parallel (AP state), the device is in a high resistance state (R_{ap}).

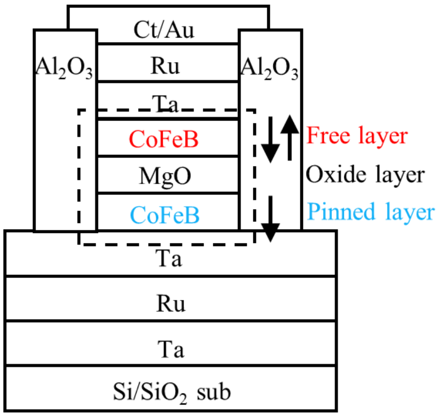


Fig. 1.1. Structure of magnetic tunnel junction.

In 1975, Julliere first observed TMR in thin oxide films. [2] The Julliere model explains this by a two-current model which is shown in Fig. 1.2. This model assumes the conservation of spin in the tunneling process, and it divides the total current into two independent partial currents. One partial current is attributed to the spin-up electrons, while the other partial current is attributed to the spin-down electrons. Unfilled states in the second layer and spin-aligned electrons from the first layer determine the current. This is the underlying reason for the two resistance states to be exhibited by the MTJ. The conductance of MTJ in P and AP state can be expressed as: [3]

$$R_p^{-1} \propto D_1(+,E_f)D_2(+,E_f) + D_1(-,E_f)D_2(-,E_f) \tag{1.1}$$

$$R_{ap}^{-1} \propto D_1(+,E_f)D_2(-,E_f) + D_1(-,E_f)D_2(+,E_f) \tag{1.2}$$

where $D(+,E_f)$ is the density of states for electrons with majority spin state at the Fermi energy (E_f) and $D(\text{-},E_f)$ is the density of states for electrons with minority spin states at the Fermi energy (E_f). The footnotes 1 and 2 represent the pinned layer and free layer in MTJs, respectively. The TMR ratio (TMRR) which is defined as the ratio between the maximum and minimum resistance of an MTJ can be expressed by the spin polarization:

$$TMRR = \frac{R_{ap} - R_p}{R_p} = \frac{2Pol_1Pol_2}{1 - Pol_1Pol_2} \tag{1.3}$$

$$Pol_i = \frac{D_i(+,E_f) - D_i(-,E_f)}{D_i(+,E_f) + D_i(-,E_f)} \qquad (i = 1,2) \tag{1.4}$$

where Pol_1 and Pol_2 are the spin polarizations of the pinned layer and free layer in MTJs. The spin polarization refers to the difference in the density of spin states at the Fermi level.

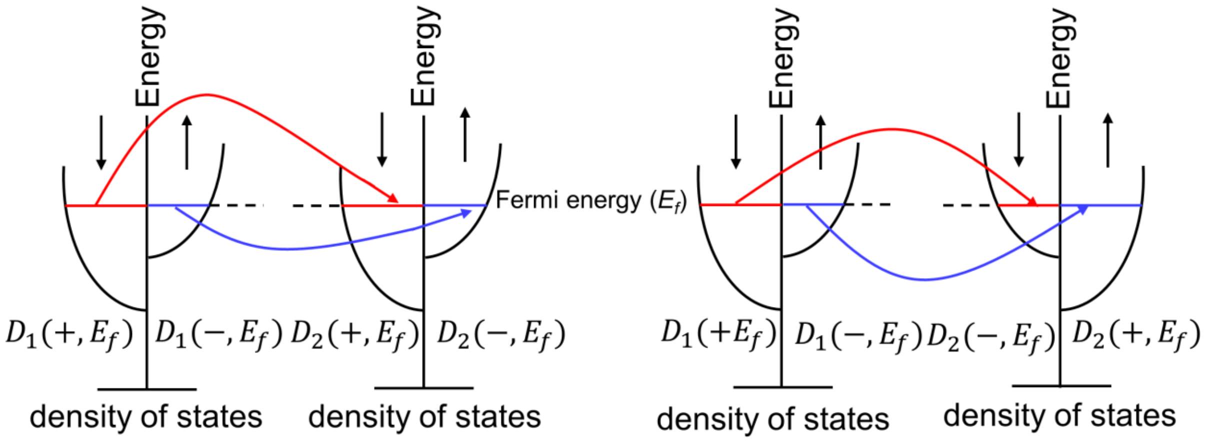


Fig. 1.2. Two-current model (Julliere model) for tunneling magnetoresistance.

However, the limited resistance change of approximately 14% for the high and low states has restricted the broader recognition of this phenomenon. By changing the oxide layer materials to Al_2O_3, the TMR ratio increases and the maximum value can reach 70.4% in 2004. [4] At the same year, Yuasa used MgO as the oxide layer and resulted in 180% TMR ratio. [5] Nowadays, the CoFeB/MgO/CoFeB structure is widely used as MTJ materials.

MTJs can be divided into three generations based on their switching mechanisms. The first generation MTJs use the magnetic fields to switch the magnetic moment of free layer. The only commercially available first-generation MTJ product is the Toggle magnetoresistive random-access memory (MRAM), which is manufactured by Freescale Semiconductor. [6]-[7] The field switching doesn't have wear-out effects which lead to a strong write endurance. However, the use of field switching in MRAMs is limited by the requirements of high switching current and complex control, which constrains the scalability of MRAMs to smaller technology nodes. [8] The example of structure of the field-switching MTJ is shown in Fig. 1.3. Cladding is a technique that involves adding permeable ferromagnetic material at write line sides which is far from the MTJ. This technique helps to focus the magnetic field towards the side without the permeable ferromagnetic material, where the MTJ is located. It can realize a stronger magnetic field to switch the state of MTJ.

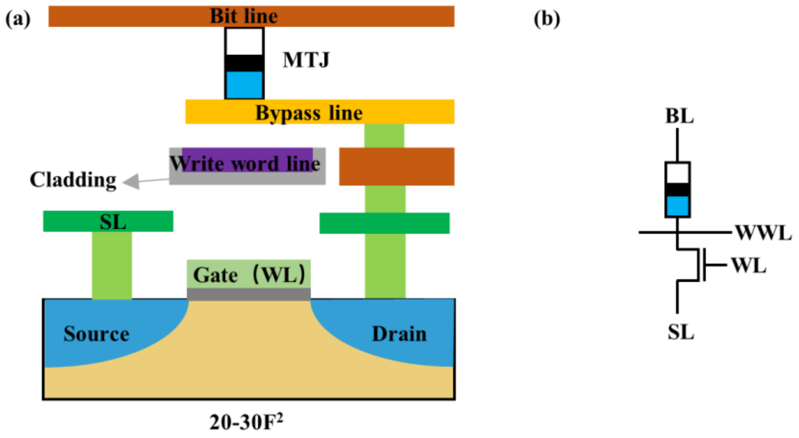


Fig. 1.3. (a) and (b) Structure of field-switching MTJ and its equivalent circuits.

Under the circumstances, the spin-transfer torque effect which was theoretically predicted independently by Berger [9] and Slonczewski [10] is used to design the second generation MTJ. Spin-transfer torque effect means when current flows through the ferromagnetic layer, the current will be polarized. Then the polarized current will exert a torque to the next ferromagnetic layer. [9]-[10] Compared with field-switching magnetic tunnel junctions, spin-transfer torque magnetic tunnel junctions exhibit the potential for significant reduction in both the required switching current and the size of the cell. [8] The structure of spin-transfer torque MTJ (STT-MTJ) is shown in Fig. 1.4. Unlike the field-switching MTJ, the STT-MTJ does not require a write word line to provide the switching field for magnetic switching which results in the size of STT-MTJ cell which can be reduced to $6F^2$ (F stands for a minimum feature size). STT-MTJ also has two categories: in-plane and perpendicular STT-MTJ. The key distinction between these two structures lies in the relative orientation of easy axis with respect to the injection current direction. In MTJs, the orientation of the magnetic moment of the fixed and free layers is the easy axis which determines the magnetic configuration. When the easy axis lies in perpendicular to injection current direction, the MTJ is in-plane MTJ. When the easy axis is oriented parallel to the direction of injection current, the MTJ is perpendicular MTJ. The perpendicular structure uses the materials with perpendicular magnetic anisotropy (PMA) and can lead to sufficient energy barrier, lower switching current and high TMR ratio. [1] Everspin announced their 1Gb commercial STT-MRAM in 2021. [11]

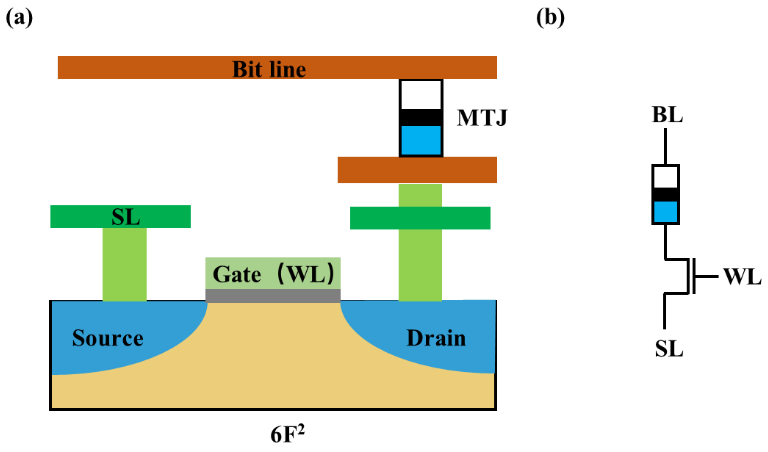


Fig. 1.4. (a) and (b) Structure of spin-transfer torque MTJ and its equivalent circuits.

Even if the switching current have been reduced to tens of microamps [1], people still explore the third generation MTJ which has smaller switching current. The physical phenomenon just like voltage-controlled magnetic anisotropy (VCMA) [12]-[14], spin-orbit torque switching (SOT) [15]-[17] and spin Hall effect (SHE) [18] are used to improve the performance. However, there still are many problems for each technology about the materials, the compatibility with high-density memory or reliability problems like parameter drift. [8] Therefore, the third generation of MTJ is still under active investigation and development.

1.2 MTJ characteristics

Figure 1.5 illustrates the structure of a perpendicular STT-MTJ and the corresponding I-V characteristics curve. STT-MTJs have two ferromagnetic layers and one oxide layer in between. The magnetization in the pinned layer keeps stable. The magnetization of the free layer changes its direction with precessional motion accompanied when spin-polarized electrons are injected. And this motion is subject to fluctuation caused by microscopic thermal agitation from environment such as phonons, conducting electrons and nuclear spins, being well described by a Brownian motion model. [19] The change in the magnetization of the free layer in an STT-MTJ is determined by the joint effects of spin-polarized current injection and thermal fluctuation, which give rise to the stochastic switching behavior of the device. The I_c in the Fig. 1.5 is the critical current which is defined as

$$I_{cp} = \frac{2\alpha\gamma e}{\mu_B \eta_p}\xi \qquad (P\ to\ AP) \tag{1.5}$$

$$I_{cap} = \frac{2\alpha\gamma e}{\mu_B \eta_{ap}}\xi \qquad (AP\ to\ P) \tag{1.6}$$

$$\xi = \frac{E}{k_B T} \tag{1.7}$$

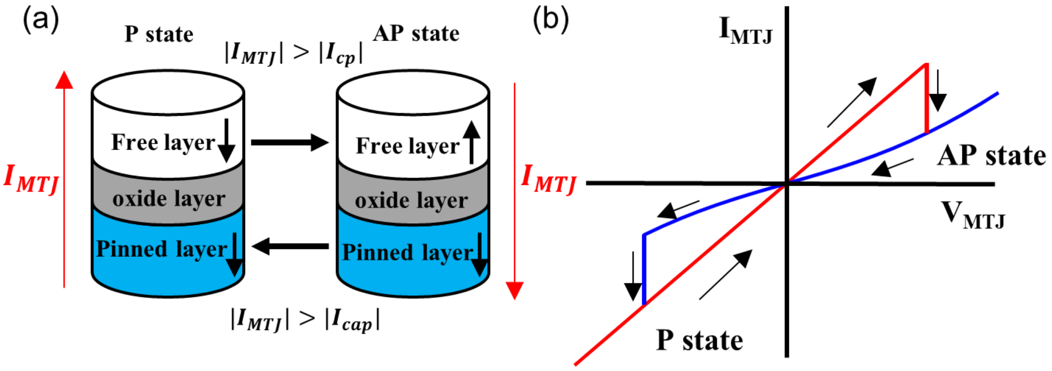


Fig. 1.5. (a) Structure of perpendicular STT-MTJ. (b) I-V characteristics of STT-MTJ.

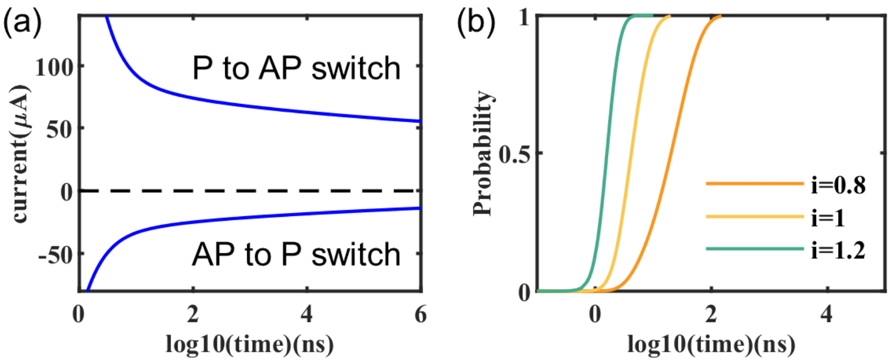


Fig. 1.6. (a) Switching characteristics of STT-MTJ for constant switching probability. (b) Switching characteristics of STT-MTJ. (i=I/I_c)

where α, γ, e, μ_B, η_p, η_{ap}, ξ, E, k_B and T are the Gilbert damping coefficient, the gyromagnetic ratio, the electronic charge, the Bohr magneton, the spin polarization factor for P state, the spin polarization factor for AP state, the thermal stability factor, the energy barrier, the Boltzmann constant and the temperature, respectively. The resistance of an STT-MTJ is determined by the relative orientation of the magnetizations in the pinned layer and the free layer.

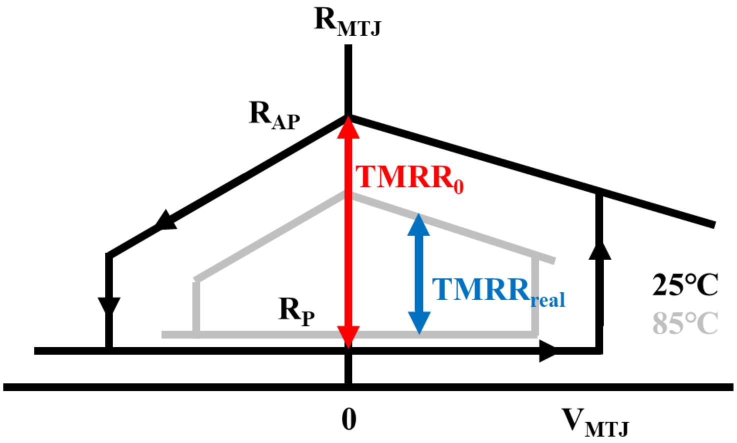


Fig. 1.7. Temperature-dependent R-V curve.

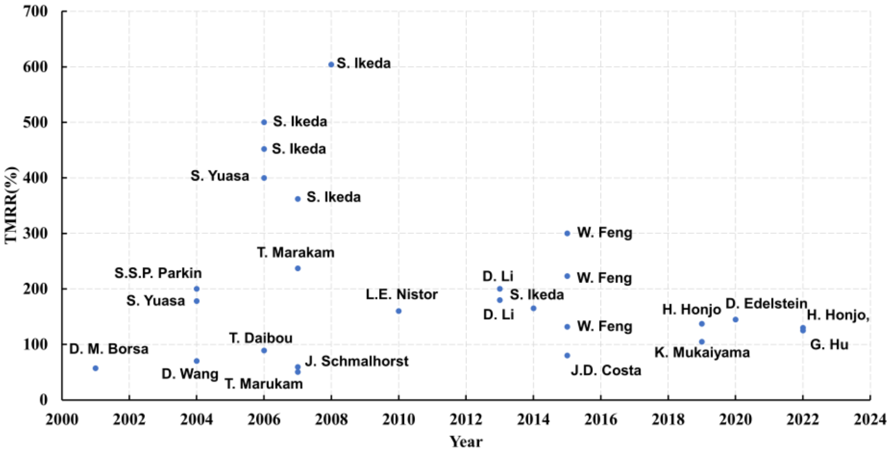


Fig. 1.8. Survey of TMR ratio.

The relationship between the injected current and the corresponding time required to switch the STT-MTJ in a certain probability is illustrated in Fig. 1.6 (a) and Fig. 1.6 (b) depicts the relationship between the switching probability and time for constant input currents. The direction of the injected current required for switching the MTJ from the P state to AP state is opposite to that required for switching it from the AP state to the P state. To make the MTJ from the P state to the AP state, the current direction must be from the pinned layer to the free layer and vice versa.

Figure 1.7 shows the temperature-dependent R-V curve. The TMRR of an STT-MTJ is affected by both the temperature and the voltage across the two terminals of MTJ. The low resistance R_P keeps constant. The $TMRR_0$ corresponds to the maximum TMRR observed when the voltage across the two terminals is 0. As the voltage across the terminals increases, the TMRR decreases. Meanwhile, the increase in temperature will also lead to a decrease in the TMRR. Most of the MTJs' TMR ratios are in the range of 100% ~ 300% [20] and the largest value of the TMR ratio can reach 604% in the room temperature [21]. Figure 1.8 shows a survey of TMRR. [20], [22]-[26].

1.3 MTJ applications and circuit simulation models with issues

The primary application of MTJs is as the core device in MRAM technology. Due to the favorable physical properties of MTJs, MRAM has become an important candidate of next generation non-volatile memories. Characteristics of conventional and emerging memories are shown at Table 1.1.[27]-[29] The colors red, yellow, and green in the table indicate the performance of memories levels of excellent, moderate, and poor, respectively.

Table 1.1. Comparison of conventional and emerging memories

Types Features	SRAM	DRAM	Flash(NAND)	FeRAM	PCM	RRAM	STT-MRAM
Cell size (F^2)	120-200	60-100	4-6	6-40	4-12	4-10	6-50
Nonvolatile	NO	NO	YES	YES	YES	YES	YES
Endurance (cycles)	$>10^{15}$	$>10^{15}$	10^5	$>10^{15}$	10^9	10^9	$>10^{15}$
Read time(ns)	~1	3	100	20~80	~10	~10	~5
Write time(ns)	~1	10	10^6	100	~10	~10	~5
Power	~fJ	~10fJ	~100pJ	1pJ	10pJ	1pJ	<1pJ

Because of the good physical characteristics, STT-MTJ also has been widely used in many fields except for MRAM, such as the weights in deep neural networks (DNN) [30], and computing in memory [31]. They are all based on the resistance property of the STT-MTJ. Based on the stochastic switching property, it is used in spiking neural networks (SNN) [32], random number generators [33], and stochastic computing [34]. Fig. 1.9 shows the typical circuits of each application which includes MRAM [35]. Nevertheless, there exist certain challenges in MTJ-based circuit. The most significant of these challenges include achieving a favorable power-stability tradeoff, the fabrication process, and managing the back-end-of-the-line (BEOL) thermal budget. [36]

To utilize a device in the circuit simulation, a circuit simulation model for the device in the electronic design automation (EDA) tools is necessary. The existing devices are mainly used by two methods in the EDA tools. The standard devices just like resistor, capacitor and MOSFET have already been programmed in the source code of the EDA tools as the built-in models. For emerging devices like MTJs, phase change memory devices, and memristors, there are no standard models. To simulate emerging devices which are implemented in CMOS circuits, it is necessary to import a model to be used in the circuit simulations.

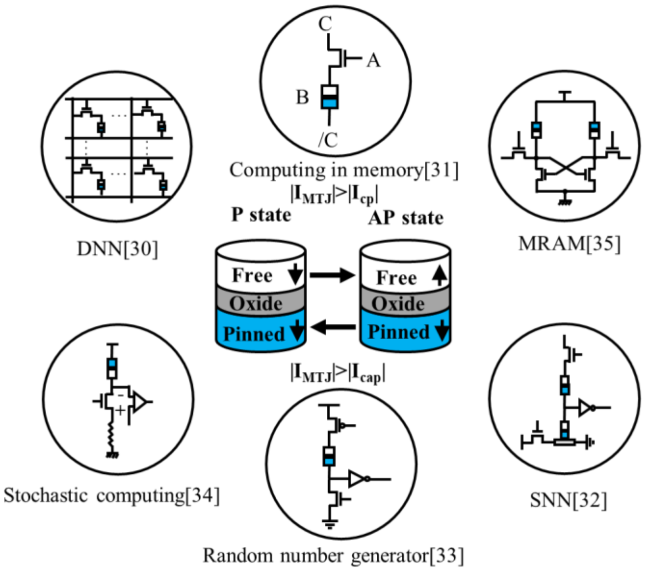


Fig. 1.9. STT-MTJ structure and its applications.

An appropriate device circuit simulation model should meet the following criteria: have high accuracy, describe device characteristics as comprehensively as possible to enable its use in various applications, and minimize resource consumption, such as CPU time and memory usage. In comparison with other emerging devices, the switching mechanism of MTJ is fully characterized by fundamental physical theories, which facilitates the development of simulation models. [37] The clear underlying mechanism facilitates the analysis and the improvement of device performance, providing a solid foundation for further development and optimization of these devices.

1.4 Stochastic Landau-Lifshitz-Gilbert-Slonczewski model

1.4.1 Landau-Lifshitz equation

To utilize the MTJ in the circuit design, it is necessary to have a simulation model in the EDA tools. However, there is no standard model for MTJ in the EDA tools. Therefore, modeling is an important work for MTJ. The most important characteristics of MTJ modeling is the switching. The orbital angular momentum and the spin angular momentum of atomic nuclei and bound electrons provide the magnetic moments. It is explained that all the magnetic moments of nanometric size of free layer or pinned layer provides a single domain. [37] Stochastic Landau-Lifshitz-Gilbert-Slonczewski (s-LLGS) model describes the dynamic characteristics of magnetization ($\vec{M}$) by this assumption. In 1935, Landau and Lifshitz developed the Landau-Lifshitz (LL) model which explains that an effective field ($\overrightarrow{H_{eff}}$) leads to precession of magnetization in the ferromagnetic material. This model also incorporates damping which is proportional to the perpendicular component of $\overrightarrow{H_{eff}}$ with respect to the magnetization. [38] The LL equation is shown as follows:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma_L \vec{M} \times \overrightarrow{H_{eff}} - \frac{\alpha\gamma}{M_S} \vec{M} \times (\vec{M} \times \overrightarrow{H_{eff}}) \qquad (1.8)$$

where γ, α, and M_S are the gyromagnetic ratio, the damping factor and the saturation magnetization, respectively.

1.4.2 Landau-Lifshitz-Gilbert equation

Experimental evidence suggests that magnetization decay occurs within a finite time period, despite the predictions of the LL model. This indicates that dissipative effects play a significant role in magnetization dynamics. The electron scattering is the main underlying mechanism for the damping. [39]-[40] Gilbert replaced the damping term in the LL model with a term that depends on the time derivative of the magnetization and incorporated the dissipative effect into the equation. [41]-[42] The Landau-Lifshitz-Gilbert (LLG) equation is shown as follows:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \overrightarrow{H_{eff}} + \frac{\alpha}{M_S} \vec{M} \times \frac{\partial \vec{M}}{\partial t} \tag{1.9}$$

It is worth noting that when the damping in a system is weak, the LL equation and the LLG equation become equivalent. In comparison to the LL equation, the LLG equation provides a better description of magnetization dynamics in the presence of strong damping. [42]

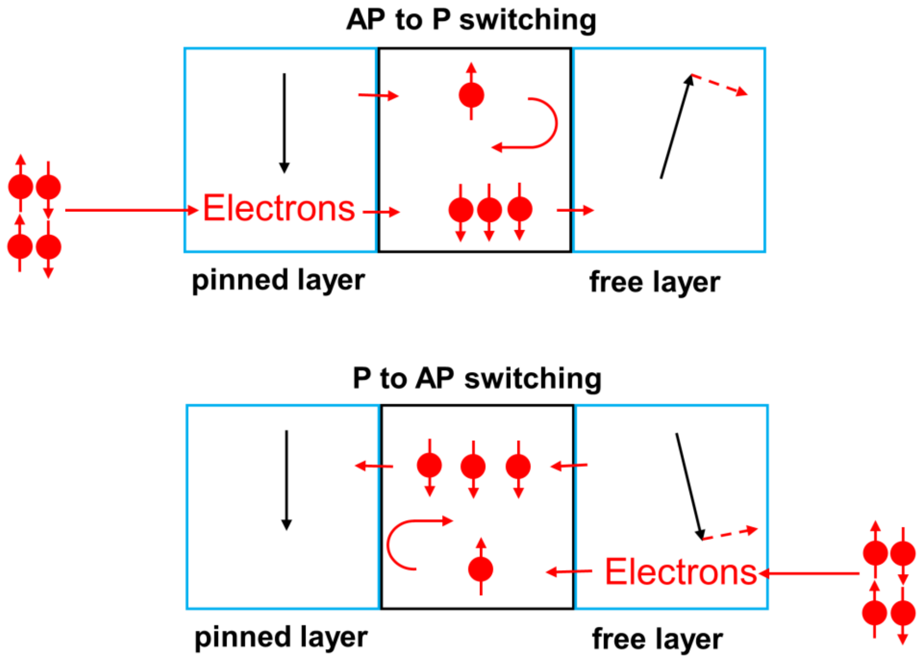


Fig. 1.10. State switching for STT-MTJ based on spin-transfer torque effect.

1.4.3 Landau-Lifshitz-Gilbert-Slonczewski equation

During flowing through the ferromagnetic material in STT-MTJ, the current will be polarized to the same direction as the magnetization direction. As a result of the conservation of angular momentum, the spin angular momentum of spin-polarized electrons is transferred to the magnetic moment of the free layer when these electrons flow from the pinned layer. This transfer induces a change in the spin angular momentum of the free layer, which drives the magnetization to align parallel with that of the pinned layer. When electrons flow from the free layer, the situation differs slightly. Electrons with an opposite spin polarization direction are reflected back into the free layer. The reflected electrons generate a torque on the magnetic moment of the free layer, driving it to align antiparallel with that of the pinned layer. [10] The schematic diagram of the processes is shown in Fig. 1.10. The spin-transfer torque exerted on the free layer can be calculated using the equation proposed by Slonczewski, which leads to the LLGS equation: [40]

$$\frac{d\vec{M}}{dt} = -\gamma\vec{M} \times \overrightarrow{H_{eff}} + \frac{\alpha}{M_S}\vec{M} \times \frac{d\vec{M}}{dt} + \frac{\gamma\eta\hbar I}{2eM_S^2 Vol}\vec{M} \times \left(\vec{M} \times \vec{p}\right) \tag{1.10}$$

where η, $\hbar$, I, e, Vol and $\vec{p}$ are the spin polarization factor of the incident current, the reduced Planck constant, the incident current, the electronic charge, the volume of the free layer and the unit vector in the direction of magnetization in the pinned layer, respectively. The three terms in the right side of the equation respectively represent the precession, the damping and the spin-transfer torque. Figure 1.11(a) elucidate the effects of individual terms on magnetization changes and Figure 1.11(b) illustrates an example of time-dependent magnetization variation under a spin-transfer torque.

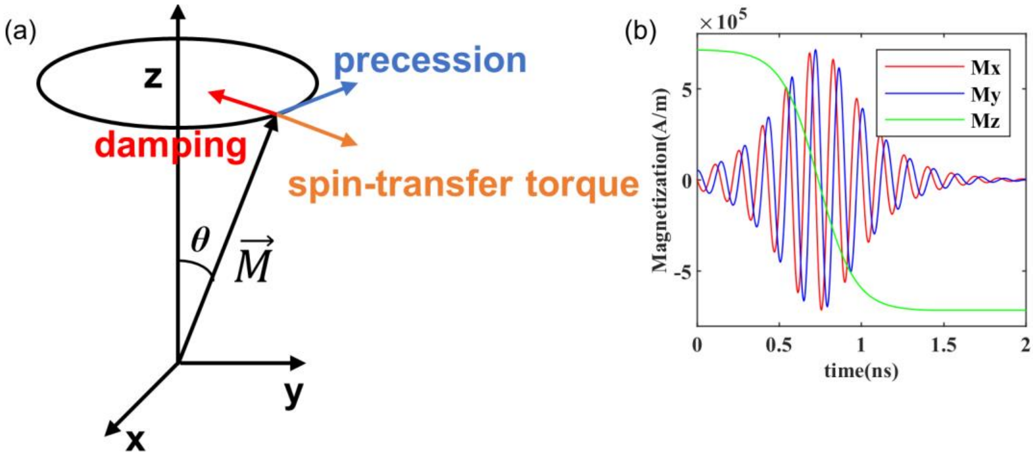


Fig. 1.11. (a) The effects of individual terms in the LLGS equation on magnetization changes. (b) MTJ switching dynamics.

1.4.4 Thermal effects

Thermal fluctuation has a significant impact on the spin dynamics in the magnetic system at the nanoscale. Figure 1.12 shows the magnetization switching trajectories with and without the thermal effect. The effect of finite temperature makes the spin do the random motion. The thermal fluctuation can be modeled by the Langevin field, which is proposed by Brown in 1963. [37] The stochastic-LLGS (s-LLGS) model can be expressed as:

$$\frac{d\vec{M}}{dt} = -\gamma\vec{M} \times (\overrightarrow{H_{eff}} + \overrightarrow{H_{th}}) + \frac{\alpha}{M_S}\vec{M} \times \frac{d\vec{M}}{dt} + \frac{\gamma\eta\hbar I}{2eM_S^2Vol}\vec{M} \times (\vec{M} \times \vec{p}) \quad (1.11)$$

$$\overrightarrow{H_{th}} = \sqrt{\frac{2\alpha k_B T}{\gamma M_S Vol\Delta t}} X_i(t) \qquad (i = x, y, z) \qquad (1.12)$$

where k_B, T, Δt and $X_i(t)$ are the Boltzmann constant, the temperature in Kelvin, the simulation timestep and zero mean standard Gaussian distribution for the x, y and z axes, respectively.

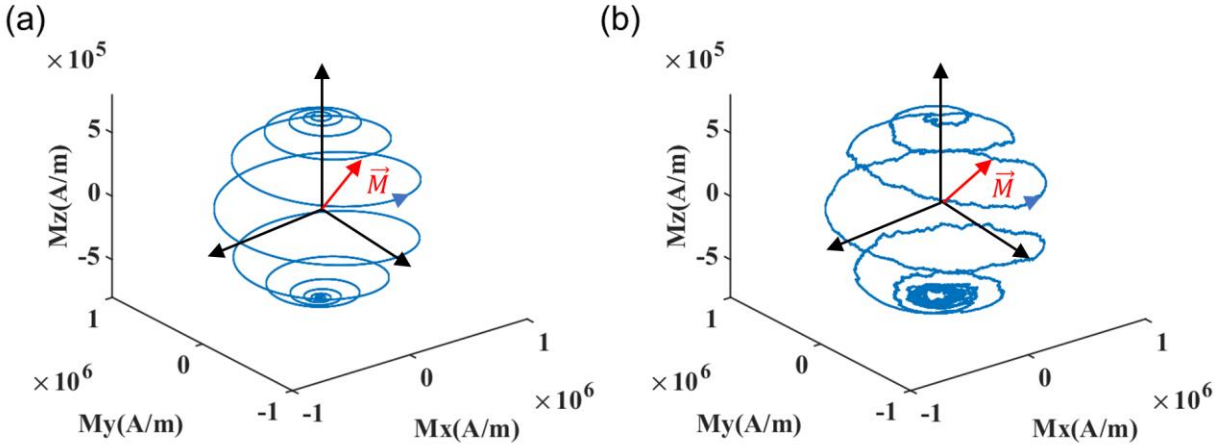


Fig. 1.12. (a) Magnetization switching trajectory without thermal effect. (b) Magnetization switching trajectory with thermal effect.

It is worth noting that the theory only works when $T \ll T_C$.T_c is the Curie temperature and T_c is 1040℃ for CoFeB [44]. When temperature is larger than the Curie temperature, the ferromagnetic materials will transfer to paramagnetic material where the magnetization is greatly influenced by external fields. [45]

Meanwhile, the thermal fluctuation also determines the initial magnetization probability distribution. The initial magnetization probability distribution for the parallel and the anti-parallel states are respectively modeled by:

$$\begin{cases} \rho_{pini}(\theta) = \dfrac{sin\theta exp\,(-\xi sin^2\theta)}{\int_0^{\frac{\pi}{2}} sin\theta exp\,(-\xi sin^2\theta)d\theta} & (\theta < \dfrac{\pi}{2}) \\ \rho_{pini}(\theta) = 0 & (\theta > \dfrac{\pi}{2}) \end{cases} \qquad (1.13)$$

$$\begin{cases} \rho_{apini}(\theta) = 0 & (\theta < \frac{\pi}{2}) \\ \rho_{apini}(\theta) = \frac{sin\theta exp(-\xi sin^2\theta)}{\int_{\frac{\pi}{2}}^{\pi} sin\theta exp(-\xi sin^2\theta) d\theta} & (\theta > \frac{\pi}{2}) \end{cases} \quad (1.14)$$

where ξ is the thermal stability factor. [46]

1.5 Effective field

The total effective field of perpendicular MTJ includes the anisotropy field (*H*$_k$), the external field (*H*$_{ex}$) and the demagnetization field (*H*$_D$):

$$\overrightarrow{H_{eff}} = \overrightarrow{H_k} + \overrightarrow{H_{ex}} + \overrightarrow{H_D} \quad (1.15)$$

Sometimes, the external field is also named as applied field (*H*$_{app}$). In magnetic systems, the anisotropy field can arise from various sources, including interfacial or magnetocrystalline effects. When present, this anisotropy field can provide enough interfacial anisotropy energy to overcome the demagnetization field along the easy axis of the magnet, thereby maintaining the magnetization in the easy axis direction. [1] The *H*$_k$ is calculated by: [47]

$$\overrightarrow{H_k} = \frac{2K}{t_{fl}\mu_0 M_S}\overrightarrow{m_z} \quad (1.16)$$

Where *K*, t_{fl}, μ_0 and $\overrightarrow{m_z}$ are the anisotropy coefficient, the thickness of the free layer, the vacuum permeability and the unit vector of the magnetization among z axes. The total external magnetic field is represented by $\overrightarrow{H_{ex}}$. For the perpendicular STT-MTJ, the $\overrightarrow{H_{ex}}$ is generally assumed to be zero by default.

For s-LLGS equation, one assumption is that there is only one magnetic moment in the ferromagnetic layer. However, there are many magnetic moments in the ferromagnetic layer and the magnetic field of these magnetic moments interacts with each other. The magnetostatic interaction results in the parallel alignment of spins, forming a single domain. This phenomenon is modeled by the demagnetization field. It is toward to reduce the total spin torque in the ferromagnetic materials and lead to the shape anisotropy. It is modeled by: [48]

$$\overrightarrow{H_D} = -M_S N_i \overrightarrow{m_i} (i = x, y \; and \; z) \quad (1.17)$$

where *N*$_i$ and $\overrightarrow{m_i}$ are the demagnetization factor for the x, y and z axes and the unit vector of the magnetization for the x, y and z axes. The demagnetization factors satisfy the relation N_x+N_y+N_z=1. The demagnetization factor of right-circular cylinder which is shown in Fig.1.13 can be calculated by [49]

$$N_x = N_y = \frac{\frac{2n}{\sqrt{\pi}}}{2\frac{2n}{\sqrt{\pi}} + 1} \tag{1.18}$$

$$N_z = \frac{1}{2\frac{2n}{\sqrt{\pi}} + 1} \tag{1.19}$$

The length is inversely proportional to N. Therefore, the demagnetization field is considered only existing in the z axes by assuming N_x=N_y=0 and N_z=1. [50] The demagnetization field can be rewrite as:

$$\overrightarrow{H_D} = -M_S N_z \overrightarrow{m_z} \tag{1.20}$$

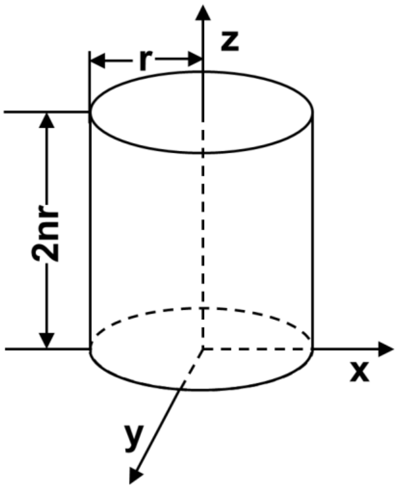


Fig. 1.13. Schematic of right-circular cylinder ferromagnetic materials.

1.6 Problems in stochastic LLGS model

The s-LLGS model is the most rigorous physical model that describes the entire trajectory of magnetization during switching. It should be noted that while the s-LLGS model offers high accuracy in circuit simulations, it comes at a cost of increased computational demands. Specifically, for each simulation time step, a random Langevin field must be generated, which requires additional computational resources and memory allocation. As a result, longer simulation times and larger memory requirements may be necessary to achieve the desired level of accuracy using the s-LLGS model. One example is about the write error rate (WER) which is the probability of non-switching. The WER needs to be 10^{-9} or lower in the STT-MRAM which means more than 10^9 times Monte Carlo simulation. [51] The number of required Monte Carlo simulations is unacceptably high, resulting in several days or more of simulation time. [52] Under the circumstances, it needs some other types of models to simulate the STT-MTJ based circuits in the EDA tools.

1.7 Motivations and Proposals

In this research, I focus on making a general purpose STT-MTJ circuit simulation model and costing less CPU time and memory with high accuracy.

The fundamental concern of the general purpose STT-MTJ model lies in its switching dynamics. The existing STT-MTJ switching models are in three categories: stochastic Landau-Lifshitz-Gilbert-Slonczewski model, switching probability model and Fokker-Planck equation model. However, each method has each own problems that limit them to be used as a sole circuit simulation model.

The switching probability model cannot work for all the current situations, which limits it to be used for the varying current. [53] To overcome this problem, a framework which connects the existing physical models which describe the switching characteristics in the very large and small current situations is introduced, as shown in Chapter 2. It is based on the conference paper in VLSI-DAT 2019 [54] and the journal article in IEICE 2020 [55]. I proposed a mathematic method to connect the existing physical models to obtain the switching curve which is close to the experimental results and used an interpolation method to obtain the switching time from the curve. The circuit simulation model is implemented by Verilog-A language which is a standard programming language to model the device in the HSPICE. The simulation results validate that my switching probability based model can work for all the current situations and only cost CPU time 20% larger than the original switching probability based model in large scale circuits.

Even if the problem was solved that the switching probability based model cannot work for all the current situations, it still cannot work for all kinds of applications because of some fundamental limitations. Specifically, the switching probability based model simplifies the resistance to two state (parallel and antiparallel) and cannot model the excitation and the relaxation in magnetization of the free layer in an STT-MTJ. As a result, it cannot be used for the applications that rely on magnetization changing information, such as spiking neural networks (SNNs). The s-LLGS model describes the magnetization changing. However, the large CPU time and memory requirements limit it to be used in large-scale circuit simulations. Meanwhile, this kind of model is hard to obtain the overall characteristics, i.e., switching probability, of an STT-MTJ. [52]

To overcome the problems mentioned above, a novel circuit simulation model based on the Fokker-Planck equation (FPE) is proposed for STT-MTJ devices, which is shown in Chapter 3. The FPE based model was reported in the conference paper in NANO 2022 [56]. The model is programmed by using Verilog-A. This is the first work in which FPE is solved to describe the switching characteristics as a circuit simulation model. I proposed an theta obtaining framework which specifies the most possible magnetization trace from the probability density distribution of theta. The theta obtaining framework can obtain the same information as the s-LLGS model and makes the FPE based circuit simulation model work for all kinds of applications. The simulation results validate that the model can fit the experimental data well and make the error rate less than 10%. It reduces the CPU time consumption to 1/30 compared with the original s-LLGS based model in large-scale circuit simulations.

Table 1.2. Brief Comparison among STT-MTJ circuit simulation models

	Applications		Accuracy	CPU time
	STT-MRAM	SNN		
s-LLGS based model	○	○	High	Long
Conventional switching probability based model	○	×	Low	Short
Proposed switching probability based model	○	×	High	Short
Proposed FPE based model	○	○	High	Medium

A brief comparison among the proposed model and existing models are shown in Table 1.2. The comparison shows the proposed switching probability based model has higher accuracy than the conventional ones. Meanwhile, it costs less CPU time than the s-LLGS or the FPE based model, which makes it usable for some applications. The FPE based model can work for wide range of applications and cost less CPU time than the s-LLGS based model. The two proposed models are used for simulating MTJ/CMOS hybrid circuits in two applications: an STT-MRAM and a proposed SNN, which are shown in Chapter 4. The proposed STT-MTJ based low power binarized SNN (BSNN) was reported in the journal article JJAP 2023 [57]. I proposed a neuron which uses the magnetization to integrate the input spikes and its relaxation to express the leak mechanism in the leaky integrate-and-fire (LIF) neuron. The 4T3MTJ structure makes it possible to be automatically reset to the initial state during outputting a spike. The BSNN is simulated by the FPE based model in HSPICE and reduces the energy consumption as low as 0.23pJ/SOP.

Chapter 2

Switching probability model

2.1 Switching probability model

Switching probability model is a statistical model which describes the probability of the magnetization in the free layer changing the state from one to the other one. The model considers the impact of the external conditions such as incident current and the material properties such as interfacial anisotropy energy. Depending on the incident current and the critical current, the switching behavior can be classified into three regimes: (1) the adiabatic precessional regime in which the injected current is significantly greater than the critical current ($I >> I_c$). (2) the thermal activation regime in which the injected current is much smaller than the critical current ($I << I_c$). (3) the intermediate regime in which the injected current is close to the critical current ($I \approx I_c$). [58] In the adiabatic precessional regime, the switching characteristics can be described using Sun model, [59]-[60] while the Néel-Brown model [61] is suitable for the thermal activation regime which is often used for checking the read disturb. However, there is currently no physical model available to describe the switching characteristics in the intermediate regime. [53] Figure 2.1 shows the performance of the two models in the whole region comparing with the experimental data from [58].

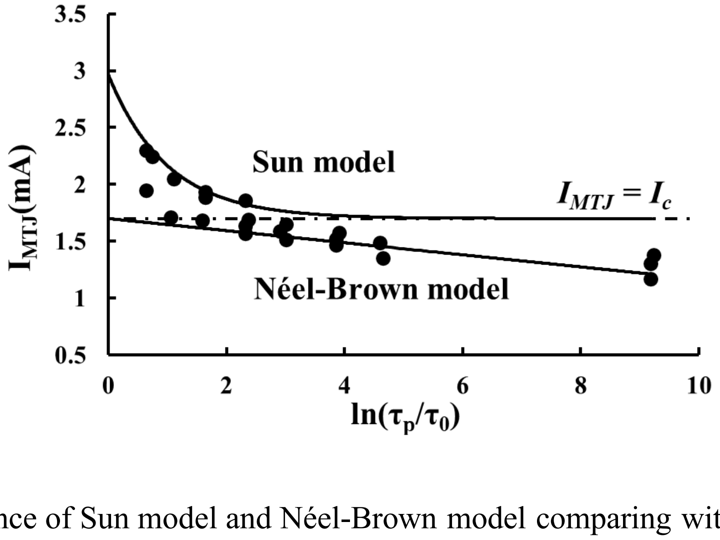


Fig. 2.1. Performance of Sun model and Néel-Brown model comparing with the experimental data from [58].

The switching probability can be expressed by the Néel-Brown relaxation-time formula which is shown as: [61]

$$P(t) = 1 - e^{(-\frac{t}{\tau})} \tag{2.1}$$

where τ is the relaxation time constant. The value is initially obtained using the Arrhenius equation, which does not account for the spin-transfer torque effect. The Arrhenius equation directly provides the value as: [62]

$$\tau = \tau_0 e^{(-\frac{E}{k_B T})} \tag{2.2}$$

where the τ_0, E, k_B and T are the inverse of attempt frequency, the energy barrier, the Boltzmann constant and the temperature, respectively. The form with spin-transfer torque effect can be derived from the s-LLGS equation [61]. The s-LLGS equation can be rewritten by using a field ($\overrightarrow{H_S}$) to represent the contribution of the spin-transfer torque.

$$\frac{1}{\gamma}\frac{d\vec{m}}{dt} = \vec{m} \times (-\overrightarrow{H_{eff}} + \overrightarrow{H_{th}} + \frac{\alpha}{M_S}\vec{m} \times (-\overrightarrow{H_{eff}} + \overrightarrow{H_S})) \tag{2.3}$$

$\overrightarrow{H_{eff}}$ is the effective field in the s-LLG equation. The equation can be simplified by using a new effective field which is shown in Eq. 2.4.

$$\begin{aligned}\widetilde{H_{eff}} &= (-H_k - H_{ex} + H_S)\overrightarrow{m_z} \\ &= (H_k + H_{ex})(\frac{H_S}{H_k + H_{ex}} - 1)\overrightarrow{m_z} \\ &= (\frac{H_S}{H_k + H_{ex}} - 1)\overrightarrow{H_{eff}}\end{aligned} \tag{2.4}$$

The Eq. 2.3 can be rewritten as:

$$\frac{1}{\gamma}\frac{d\vec{m}}{dt} = \vec{m} \times (-\overrightarrow{H_{eff}} + \overrightarrow{H_{th}} - \frac{\alpha}{M_S}(1 - \frac{H_S}{H_k + H_{ex}})\vec{m} \times \overrightarrow{H_{eff}}) \tag{2.5}$$

The damping factor can be written as:

$$\begin{aligned}\tilde{\alpha} &= (1 - \frac{H_S}{H_k + H_{ex}})\alpha \\ &= (1 - \frac{I}{I_c})\alpha\end{aligned} \tag{2.6}$$

Whether or not the spin-transfer torque effect is considered, the Langevin field remains constant. Based on this, we can obtain the equation:

$$\overrightarrow{H_{th}} = \sqrt{\frac{2\alpha k_B T}{\gamma M_S Vol\Delta t}} X_i(t)$$

$$= \sqrt{\frac{2\tilde{\alpha} k_B \tilde{T}}{\gamma M_S Vol\Delta t}} X_i(t) \tag{2.7}$$

The temperature should also be a fictitious temperature to keep the Langevin field keep constant based on the effective damping factor. The relaxation time constant with the fictitious temperature can be written as:

$$\tau = \tau_0 e^{(-\frac{E}{k_B \tilde{T}})} = \tau_0 e^{(-\frac{E}{k_B T}(\frac{\tilde{\alpha}}{\alpha}))} = \tau_0 e^{(-\frac{E}{k_B T}(1-\frac{I}{I_c}))} \tag{3.8}$$

Finally, by substituting Eq. 2.1 into Eq. 2.8, the switching probability in the thermal activation regime can be expressed as:

$$P(t) = 1 - exp\left\{-\frac{t}{\tau_0} exp\left[-\frac{E}{k_B T}\left(1-\frac{I}{I_c}\right)\right]\right\} \; (I \ll I_C) \tag{2.9}$$

Several different models have been proposed to describe the switching behavior in the adiabatic precessional regime, such as those presented in references [53], [59]-[60]. However, the Sun model is the most commonly used to describe the switching characteristics in this regime. The first Sun model is used for the in-plane STT-MTJ which is expressed as:[60]

$$\tau = \frac{\tau_D \, ln\left(\frac{\frac{\pi}{2}}{\theta_0}\right)}{\left(\frac{I}{I_c} - 1\right)} \tag{2.10}$$

where τ_D is the relaxation time. Combine it with the Néel-Brown relaxation-time formula, the switching probability can be expressed by:

$$P(t) = 1 - exp\left[-\frac{\frac{t}{\tau_D}\left(\frac{I}{I_c} - 1\right)}{ln\left(\frac{\frac{\pi}{2}}{\theta_0}\right)}\right] \tag{2.11}$$

However, it is not work well for the perpendicular STT-MTJ. In this situation, Sun proposed another model which is shown as following:

$$P(t) = \exp\left(-\frac{4E}{k_B T}\exp\left(-\frac{2t\left(\frac{I}{I_c}-1\right)}{\tau_D}\right)\right) \tag{3.12}$$

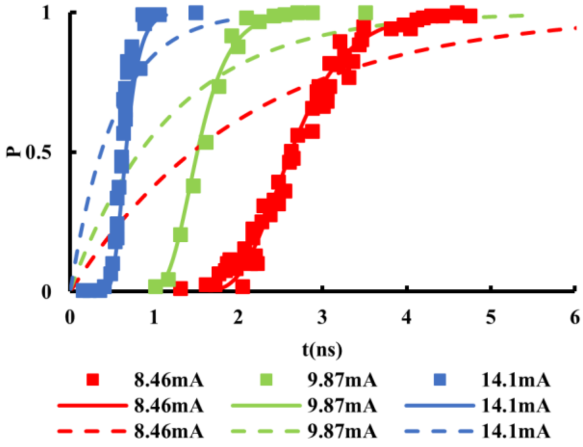


Fig. 2.2. Comparison among experimental data, SUN model and the newer model. Solid line represents newer model, dotted line represents SUN model and ■ represents experimental data. (Experimental data is from [59])

Figure 2.2 compares these two models with the experimental data and the results indicate that the Eq. 2.12 is more consistent with the experimental data.

2.2 The problems in the switching probability model

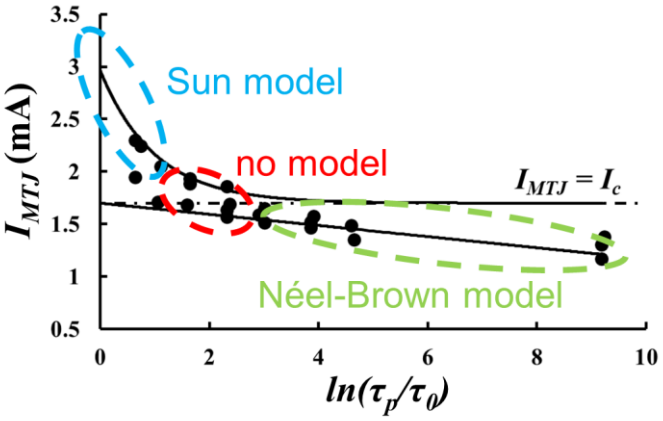


Fig. 2.3. The applied range differs among various models.

Nevertheless, none of these models can accurately depict the magnetic behavior of materials in the intermediate regime, which poses a significant problem in the field of MTJ modeling. [53] In practical applications, the adiabatic precessional region is the main region of interest. However, the intermediate regime is also used in some applications and this regime is important to handle the time varying current. The switching characteristics are greatly affected by the time varying current. The intermediate problem makes it difficult to handle the varying current. I simulated the switching characteristics for ideal and non-idea current based on the LLGS model. In Fig. 2.4, we can see that switching happens at 1.7ns for ideal current pulse. For non-ideal current pulse, switching happens at 2.2ns. There is 23% switching time difference between ideal and non-ideal current pulse. Some existing models [63]-[65] have employed various forms of model in the adiabatic precessional regime. They determine whether a switch occurs by comparing the precomputed switching time with the actual current input time. This approach makes it impossible to handle time varying currents even when using a single model.

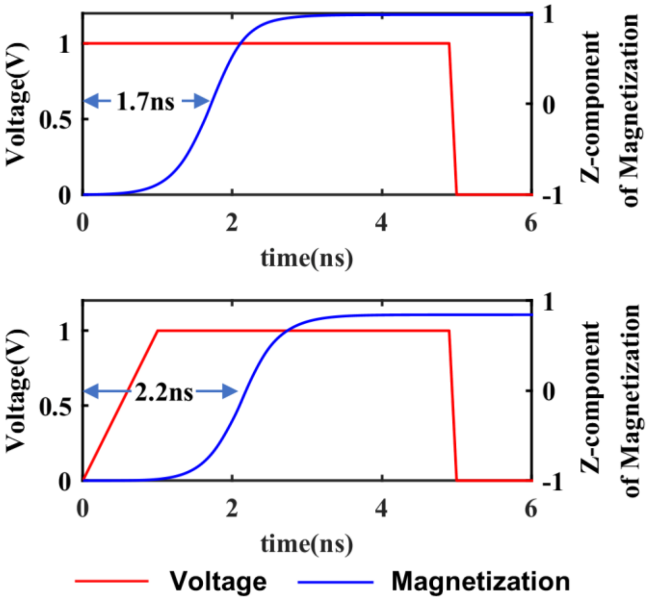


Fig. 2.4. Switching characteristic of ideal current pulse and non-ideal current pulse.

The Néel-Brown model and Sun model cannot work for the intermediate regime. Figure 2.5 illustrates a comparison among the switching occurrences in a single physical model and the experimental data. The critical current is 6.55mA in this simulation. For Sun model, there is a 31% deference between Sun model and experimental data when I=1.07I_c and a difference larger than 100% when I a little bit larger than I_c. The Sun model tends towards infinity in the vicinity of the critical current region. It fits the experimental data well starting from 1.15I_c where switching time is 5ns. As for only using Néel-Brown model, there is also a l difference larger than 100% when I a little bit larger than I_c. The results show that, it is hard to use the sole existing model to describe the switching characteristic around the critical current. In summary, accurately modeling MTJs is a challenging task that cannot be achieved using a single model or by directly switching between the two physical models.

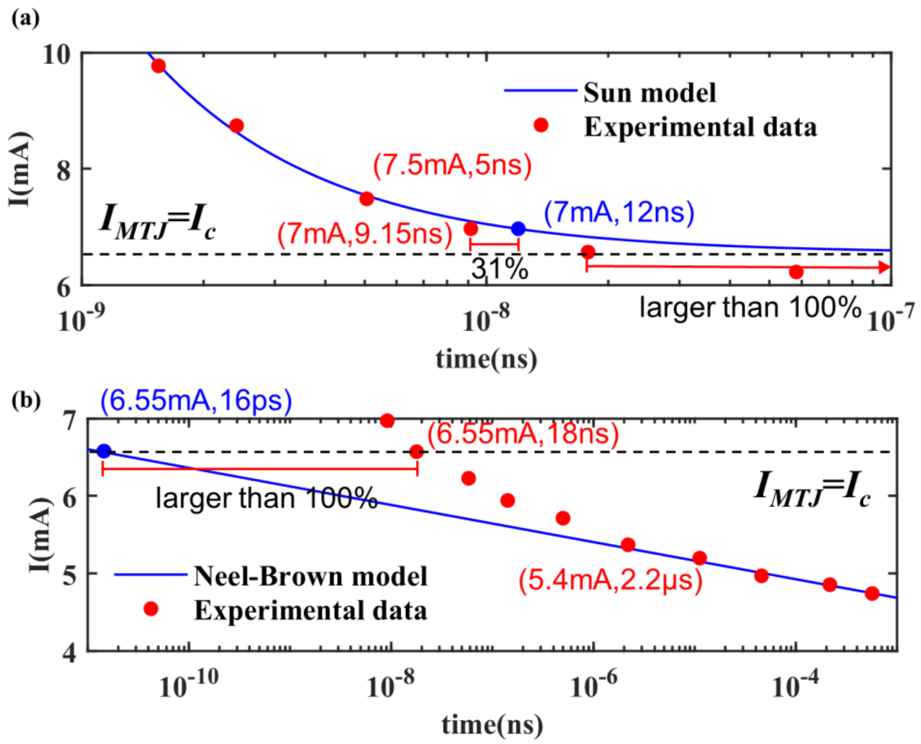


Fig. 2.5. Switching characteristic for different physical models. (a) Only Sun model (b) Only Néel-Brown model. (Experimental data is from [59])

2.3 Proposed switching probability model

2.3.1 Framework to get the switching probability

We proposed a mathematical framework to solve the problem of the intermediate regime. [54]-[55] The fundamental concept is to utilize the two types of models separately in their respective regimes and integrate them in the intermediate regime. The Néel-Brown model mainly describes the effect of thermal fluctuations, while the Sun model describes the dominance of the spin-transfer torque. The intermediate regime can be considered as the two kinds of effect work together.

The process of each step of the framework is illustrated in Fig. 2.6, with the same color scheme used in (a) and (b). In the first step, a model is chosen to calculate the switching time (t_0) based on the current value. Because the Sun model trends to infinity around the critical current, we set a small positive constant value (λ) to select a suitable physical model. When the incident current is 1+λ times larger than the critical current, the t_0 is equal to:

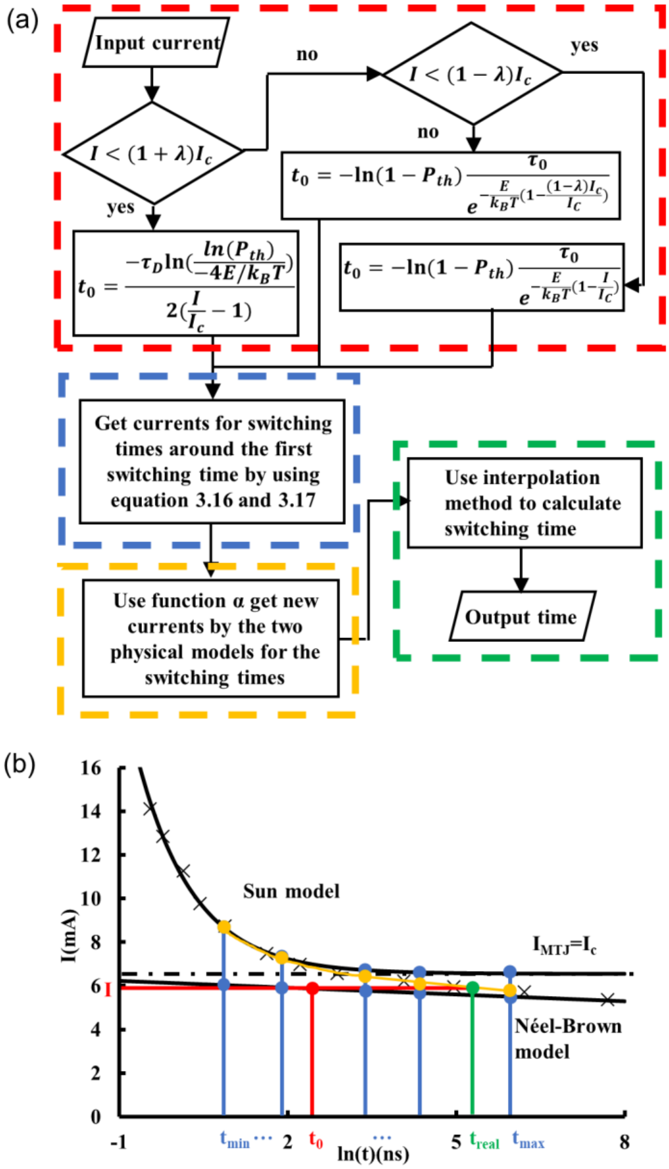

(a)
Input current
$I < (1+\lambda)I_c$
no
yes
$I < (1-\lambda)I_c$
no
yes
$t_0 = \frac{-\tau_D \ln(\frac{ln(P_{th})}{-4E/k_BT})}{2(\frac{I}{I_c}-1)}$
$t_0 = -\ln(1-P_{th})\frac{\tau_0}{e^{-\frac{E}{k_BT}(1-\frac{(1-\lambda)I_c}{I_c})}}$
$t_0 = -\ln(1-P_{th})\frac{\tau_0}{e^{-\frac{E}{k_BT}(1-\frac{I}{I_c})}}$
Get currents for switching times around the first switching time by using equation 3.16 and 3.17
Use function α get new currents by the two physical models for the switching times
Use interpolation method to calculate switching time
Output time
(b)
I(mA)
16
14
12
10
8
6
4
2
0
I
Sun model
$I_{MTJ}=I_c$
Néel-Brown model
-1
t_{min} ⋯ 2
t_0
⋯
5
t_{real}
t_{max}
8
ln(t)(ns)

Fig. 2.6. (a) Flow chart of the switching probability model in the intermediate region. (b) Schematic diagram of the switching probability model in the intermediate region. × represents experimental data which are from [59].

$$t = \frac{-\tau_D ln(\frac{ln(P_{th})}{-4E/k_BT})}{2(\frac{I}{I_c}-1)} \tag{2.13}$$

where P_{th} is a predetermined switching probability. If the incident current is 1-λ times smaller than the critical current, the t_0 is equal to:

$$t = -ln(1 - P_{th})\frac{\tau_0}{e^{-\frac{E}{k_B T}(1-\frac{I}{I_c})}} \tag{2.14}$$

Otherwise, when the incident current is between 1+λ and 1-λ times critical current, t_0 can be calculated as follows:

$$t = -ln(1 - P_{th})\frac{\tau_0}{e^{-\frac{E}{k_B T}\left(1-\frac{(1-\lambda)I_c}{I_c}\right)}} \tag{2.15}$$

This step is illustrated in red color in Fig. 2.5. The second step obtains several time points around the t_0 from t_{min} to t_{max} which are the values related to t_0. The models can be rewritten as a relationship between the incident current and time:

$$\mathrm{I_{Néel}} = \mathrm{I_c}\left(1 + \frac{\ln\left(-\frac{\ln(1-\mathrm{P_{th}})\,\tau_0}{\mathrm{t}}\right)}{\frac{\mathrm{E}}{\mathrm{k_B T}}}\right) \tag{3.16}$$

$$\mathrm{I_{new}} = \mathrm{I_c}\left(1 - \frac{\tau_D \ln\left(\frac{\ln(\mathrm{P_{th}})}{-4\mathrm{E}/\mathrm{k_B T}}\right)}{2\mathrm{t}}\right) \tag{3.17}$$

The current values corresponding to the time points can be calculated for the Néel-Brown model and Sun model. This step is illustrated in blue or in Fig. 2.6.

The next step involves using a function to smoothly connect the two models. The function is shown as following:

$$f = \frac{1}{1 + e^{-A(ln(t)-B)}} \tag{3.18}$$

where A and B are fitting parameters. We chose switching time as the parameter to connect the models due to the fact that the Sun model's calculation results in the thermal activation regime are negative, which is physically impossible. Meanwhile, we use the natural logarithm form to express time in the calculation. The reason is that the switching time for different incident current has several orders of magnitude differences which makes it hard to directly use the linear time. The real current corresponding with the time can be calculated by:

$$I_{real} = fI_{Néel} + (1 - f)I_{Sun} \tag{3.19}$$

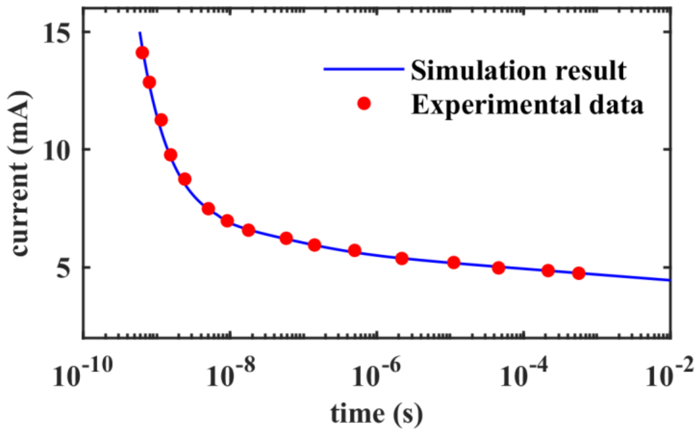


Fig. 2.7. The numerical calculation results obtained from framework for I-t curve compared with experimental data taken from [59].

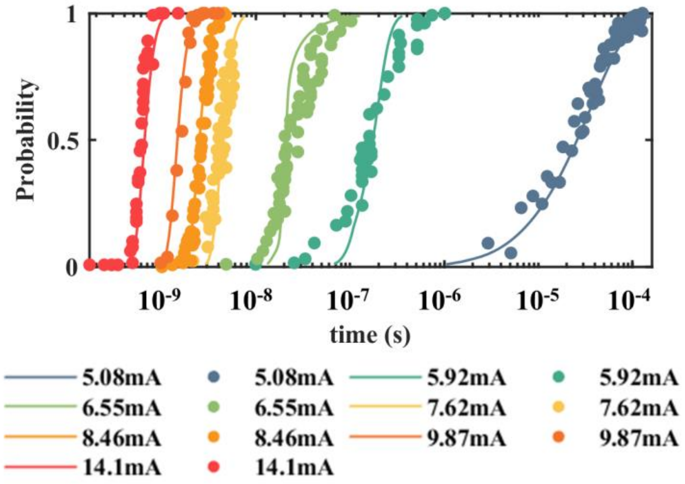


Fig. 2.8. Numerical calculation results of switching probability compared with experimental data taken from [59]. Solid line represents calculation results and ● represents experimental data.

This step is illustrated in orange color in Fig. 2.6. The orange curve in the Fig.15 can be thought as the real switching curve of the STT-MTJ. The last step is obtaining the real switching time (t_{real}) by the real switching curve. Since the connection function is a function of time, switching time can be calculated by a interpolation method. R By repeating the framework for each timestep, the switching curve can be obtained. The simulation results obtained using Matlab are shown in Fig. 2.7 and Fig. 2.8. In Fig. 2.7, we simulated the relationship between the incident current and the switching time at a switching time of 50%. The simulation results of the proposed framework are consistent with the experimental data from [59].

It is equally important for the model to have the ability to operate under time varying current conditions. My approach involves using an effective time to accumulate the contribution of each timestep. The value of effective time is between 0 and 1. For each timestep the effective time is update by:

$$t_{eff}(t+\Delta t) = t_{eff}(t) + \frac{\Delta t}{t_{real}} \tag{2.20}$$

Switching is determined to occur when t_{eff} equals 1. Using this method, the model can be applied to time varying currents.

2.3.2 Interpolation method

There are several interpolation methods just like linear interpolation, polynomial interpolation and cubic interpolation. Here, we compare the linear interpolation method and cubic interpolation method. The simplest interpolation method is linear interpolation. If we get two points (x_0, y_0) and (x_1, y_1), the value at x_2 can be evaluated by:

$$y_2 = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 \tag{2.21}$$

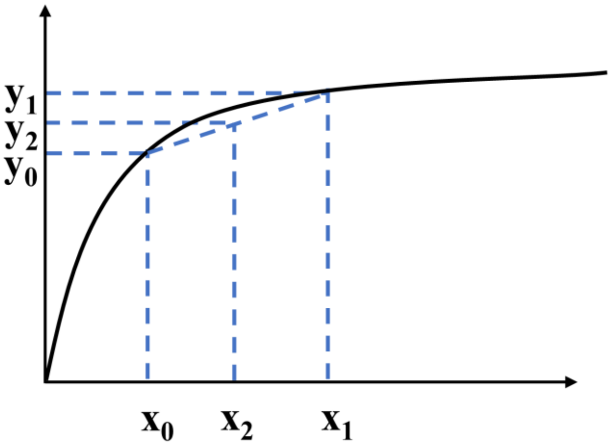


Fig. 2.9. Schematic diagram of the linear interpolation.

The next interpolation method is cubic spline interpolation. [66] With this method, greater accuracy can be achieved by selecting more points. The result can be evaluated by:

$$y = \begin{cases} S_0(x) = a_0x^3 + b_0x^2 + c_0x + d_0\ (x_0 < x < x_1) \\ \vdots \\ S_{n-1}(x) = a_{n-1}x^3 + b_{n-1}x^2 + c_{n-1}x + d_{n-1}\ (x_{n-1} < x < x_n) \end{cases} \tag{2.22}$$

For every region, it uses a third-order polynomial to get the result. Figure 2.10 shows how this interpolation method work.

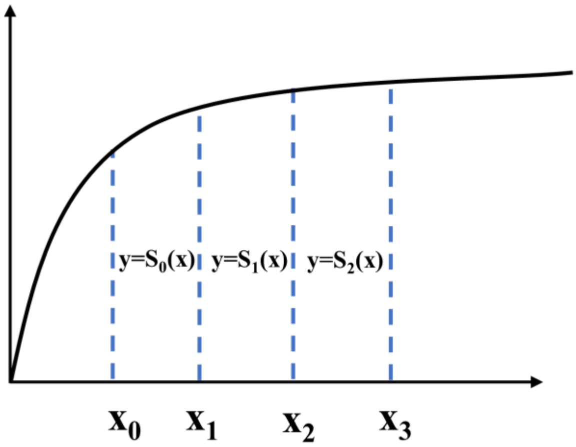


Fig. 2.10. Schematic diagram of the cubic interpolation.

Boundary conditions is S(x), the first derivative and the second derivative of every S(x) is continuous. It means $S_{i-1}(x_i) = S_i(x_i)$, $S_{i-1}'(x_i) = S_i'(x_i)$ and $S_{i-1}''(x_i) = S_i''(x_i)$. Assume $z_i = S''(x_i)$. The linear spline is simply expressed as

$$S_i''(x) = z_i \frac{x - x_{i+1}}{x_i - x_{i+1}} + z_{i+1} \frac{x - x_i}{x_{i+1} - x_i} \tag{2.23}$$

We introduce $h_i = x_{i+1} - x_i$. After twice integration, we can obtain:

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - x_i)^3 + \frac{z_i}{6h_i}(x_{i+1} - x)^3 + C_i(x - x_i) + D_i(x_{i+1} - x) \tag{3.24}$$

Based on the $S_i(x) = y_i$ and $Si(x_{i+1}) = y_{i+1}$, the Eq. 2.24 can be expressed as:

$$\mathrm{S_i}(x) = \frac{z_{i+1}}{6h_i}(x - x_i)^3 + \frac{z_i}{6h_i}(x_{i+1} - x)^3$$

$$+(\frac{y_{i+1}}{h_i}-\frac{z_{i+1}}{6}h_i)(x-x_i)+(\frac{y_i}{h_i}-\frac{h_i}{6}z_i)(x_{i+1}-x) \tag{2.25}$$

By substituting the above equation into the continuity equation ($S_i'(x_i) = S_{i-1}'(x_i)$) and solving, we obtain the following equation:

$$6(b_i - b_{i-1}) = h_{i-1}z_{i-1} + 2(h_{i-1} - h_i)z_i + h_i z_{i+1} \tag{2.26}$$

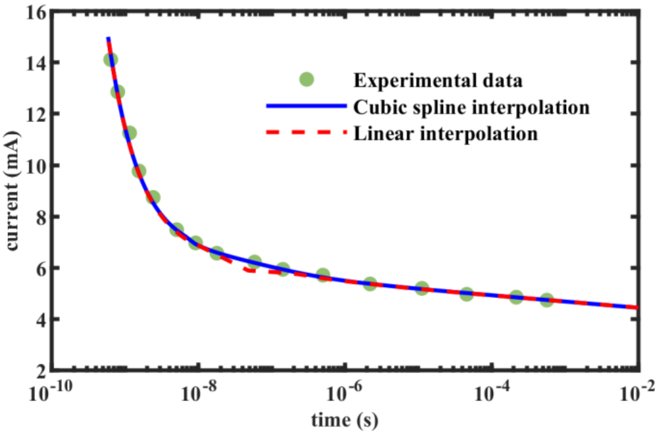


Fig. 2.11. Comparison of cubic spline interpolation, linear interpolation and experimental data from [59].

Using Eq. 2.26 and applying the boundary conditions z_0=z_n=0, the value of z_i can be determined. We tried the two different interpolation methods and the results are shown in Fig. 2.11. The experimental data is taken from [59]. Linear interpolation requires less simulation time but has poor stability and exhibits a glitch around the critical current. In contrast, cubic spline interpolation does not exhibit a glitch and fits the experimental data well. Considering both the demand for accuracy and simulation time in the simulation task, we can determine which interpolation method to use specifically.

2.3.3 Other physical models

The resistance of STT-MTJ depends on the theta between the magnetizations of the pinned layer and free layer. However, the switching probability based STT-MTJ circuit simulation model can provide information about changes in the resistance state. In this situation, the resistance of STT-MTJ is simplified to two resistance state: low resistance (R_p) and high resistance (R_{ap}). The low resistance can be expressed by [63]

$$R_P = \frac{t_{ox}}{F \times Area \times \sqrt{\varphi_B}} exp(1.025t_{ox} \times \varphi_B) \tag{2.27}$$

Where t$_{ox}$, F, Area and φ$_B$ mean thickness of the oxide layer, a fitting factor calculated from the resistance-area product (RA), the area of the MTJ surface and energy barrier height of the oxide layer, respectively. The high resistance state is calculated by a predefined TMR ratio:

$$R_{ap} = (1 + TMRR)R_p \tag{2.28}$$

The value of TMR is not constant and can be influenced by the voltage applied to the two ends of the STT-MTJ. The voltage dependent TMRR can be calculated by: [63]

$$\mathrm{TMRR(V)} = \frac{\mathrm{TMRR_0}}{1 + \frac{\mathrm{V}^2}{\mathrm{V_h^2}}} \tag{2.29}$$

where *TMRR$_0$* and *V$_h$* are the *TMRR* at 0V and the bias voltage for *TMRR* = *TMRR$_0$/2*, respectively. The critical current is calculated by:

$$I_{cp} = \frac{2\alpha\gamma e}{\mu_B \eta_p}\xi \qquad (P\ to\ AP) \tag{2.30}$$

$$I_{cap} = \frac{2\alpha\gamma e}{\mu_B \eta_{ap}}\xi \qquad (AP\ to\ P) \tag{2.31}$$

The spin polarization factor η of parallel and anti-parallel can be separately calculated by: [67]

$$\eta_p = \left(-4 + \left(Pol^{-\frac{1}{2}} + Pol^{\frac{1}{2}}\right)^3 /4\right)^{-1} + \frac{Pol/2}{1 + Pol^2} \tag{2.32}$$

$$\eta_{ap} = \left(-4 + 0.5\left(Pol^{-\frac{1}{2}} + Pol^{\frac{1}{2}}\right)^3 /4\right)^{-1} + \frac{Pol/2}{1 - Pol^2} \tag{2.33}$$

Many parameters of STT-MTJs are affected by temperature such as thermal stability factor, critical current, relaxation time and TMR ratio. The temperature dependent TMR ratio can be calculated using the following equation: [53] [64]

$$TMRR(T) = TMRR(0K) - kT \tag{2.34}$$

Where *TMRR(0K)* and k are the TMR ratio at 0K and the TMR ratio slope with temperature, respectively. The temperature variation caused by the flowing current is modeled as two processes. When a current flows, the resulting temperature increase due to the current heating effect can be calculated by the following equations: [53] [64]

$$T_{heat} = T_{room} + \Delta T_{max}\left[1 - exp\left(-\frac{D_{heat}}{\tau_{th}}\right)\right] \tag{2.35}$$

$$\Delta T_{max} = \frac{V_{mtj}J_{mtj}t_{ox}}{\lambda} \tag{2.36}$$

$$\tau_{th} = \frac{C_v t_{MTJ} t_{ox}}{\lambda} \tag{2.37}$$

where T_{heat}, T_{room}, D_{heat}, V_{mtj}, J_{mtj}, λ, C_v and t_{MTJ} are the temperature during the heating period, the room temperature, the time duration of the heating period, the MTJ voltage, the current density of MTJ, the thermal conductivity, the heat capacity per unit volume and the thickness of MTJ. When no current flows through the MTJ, the resulting temperature decrease can be calculated using the following equation: [53] [64]

$$T_{cool} = T_{room} + (T_{heat} - T_{room})\exp\left(-\frac{D_{cool}}{\tau_{th}}\right) \tag{2.38}$$

where T_{cool} and D_{cool} are the temperature during the cooling period and the time duration of the cooling period.

The thermal model also suffers from the issue that it cannot be directly applied to situations where the current is changing with time. We use a similar method to solve this by setting an effective *D_{heat}*. For each timestep when current flows and $T_{heat} > T_{room}$, we calculate the effective D_{heat} by:

$$D_{heateff} = -\tau_{th} ln(\frac{T_{room} - T_{heatlast}}{\Delta T_{max}} + 1) \tag{2.39}$$

where $T_{heatlast}$ is the temperature at last timestep. Then, the T_{heat} of this timestep can be calculated based on $D_{heateff} + \Delta t$.

2.4 Validation results

For making the model compatible with the HSPICE, we implement the model in Verilog-A language. To implement the devices model, there are mainly two methods: subcircuit and Verilog-A. Comparing with the subcircuit method, the Verilog-A method is more flexible especially for the devices with complex physical theory.

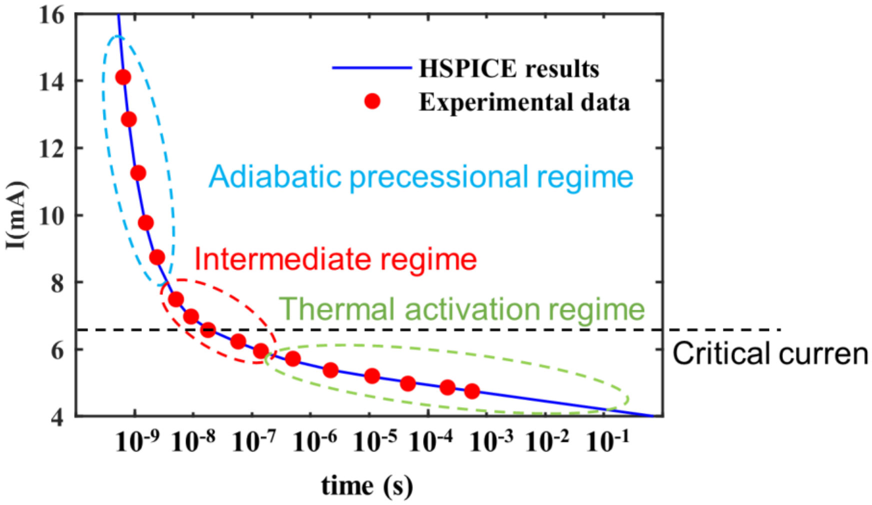


Fig. 2.12. The HSPICE results obtained from framework for I-t curve compared with experimental data taken from [59].

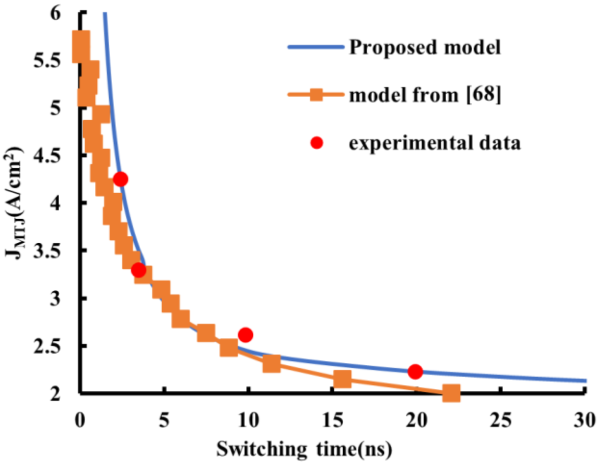


Fig. 2.13. HSPICE simulation results of model accuracy comparison. (The experimental data and the model results are taken from [68]).

The model is validated in two aspects. Firstly, the model is validated for individual devices. In this simulation, one STT-MTJ was directly connected to a current source. The cubic spline interpolation is used in all the simulation. The relationship between the incident current and switching time for a constant switching probability which is equal to 50% is shown in Fig. 2.12 and Fig. 2.13. Meanwhile, a LLGS model simulation result is also shown in Fig. 2.13 and compare with our model. Compared to experimental data, the average error rate of my model is 8.96%, whereas that of the s-LLGS based model [68] is 25.5%. The relationship between the switching time and probability for a constant incident current is shown in Fig. 2.14. Compared with the experimental data from [59], our model is consistent with the experimental results and effectively captures the switching behavior in the intermediate regime.

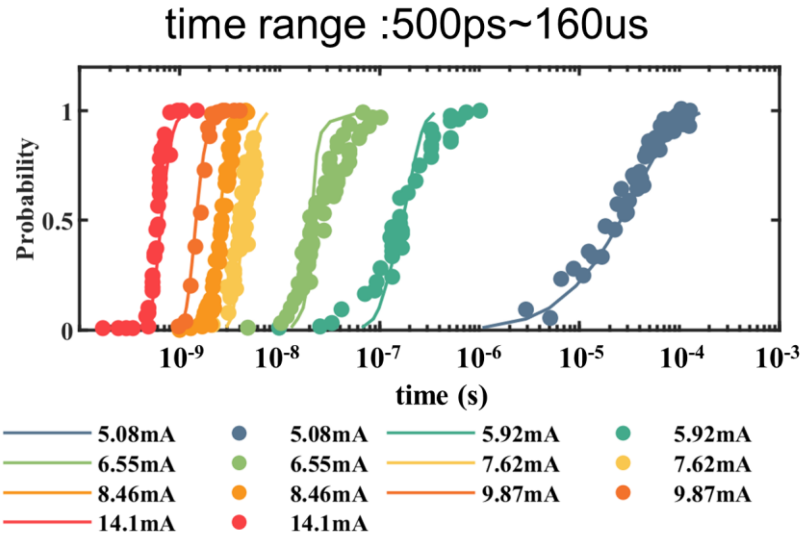


Fig. 2.14. HSPICE simulation results of switching probability versus ln(t) compared with the experimental data taken from [59]. Solid line with × represents HSPICE simulation results and ● represents experimental data.

The thermal model was validated by simulating the switching characteristics under different temperatures and the results are shown in Fig. 2.15. Meanwhile, the Fig. 2.15 also shows the voltage dependence of the TMR ratio. Among the voltage range applied to the MTJ, the resistance of anti-parallel state shows a significant decrease. The temperature dependent TMR ratio is also shown in Fig. 2.15. The TMR ratio exhibits a decreasing trend with increasing temperature.

A time varying input simulation is although necessary. Figure 2.16 illustrates the switching time for constant and time varying inputs. Despite the input voltage reaching the same value of 0.5V, a significant difference is observed in the switching time. Specifically, the switching occurs at 1.77ns with constant input, while it occurs at 2.49ns for time varying input.

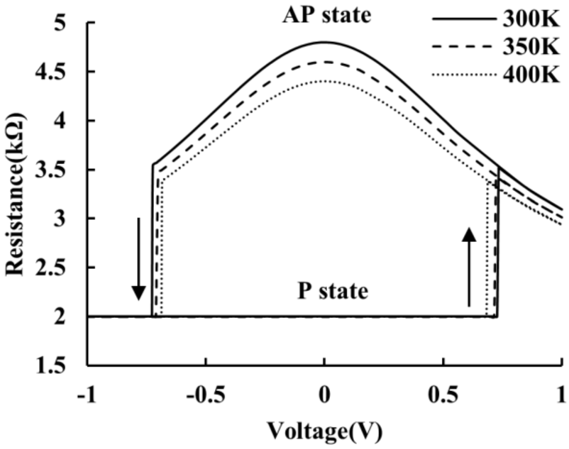


Fig. 2.15. HSPICE simulation results of MTJ resistance versus voltage between two terminals of MTJ for different temperature.

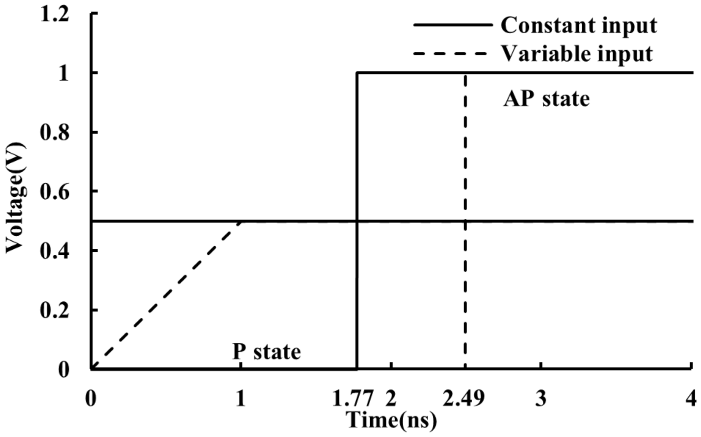


Fig. 2.16. HSPICE simulation results of constant and time varying current.

Secondly, the model is simulated with other standard devices just like metal oxide semiconductor field effect transistor (MOSFET) and capacitor to validate the compatibility. The structure of 1T1C1MTJ circuit is shown in Fig. 2.17. I realize a single AP state writing and the results of the current and resistance state changing are shown in Fig. 2.18. Simulated results are consistent with expectations. The results shows that my switching probability based model can work with other standard devices in the EDA tools. The practical applications of switching probability based MTJ model in more complex applications will be demonstrated in Chapter 6.

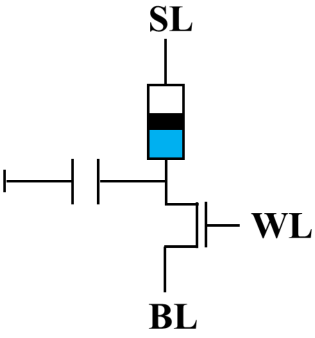


Fig. 2.17. 1T1C1MTJ structure.

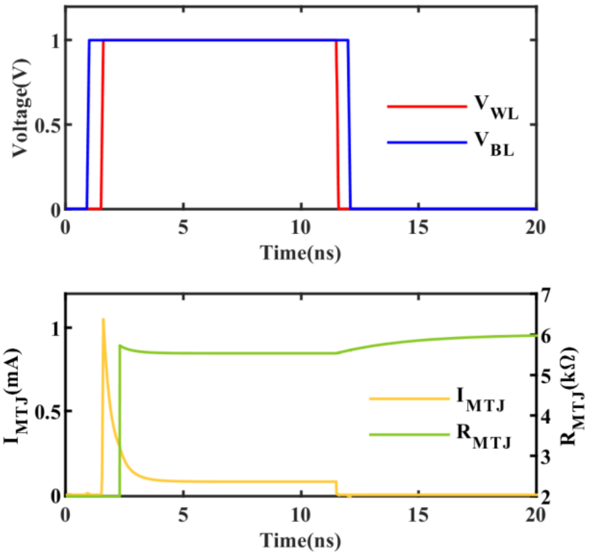


Fig. 2.18. HSPICE simulation 1T1CMTJ structure with P to AP switch. The V_{SL} keeps 0 during the simulation.

The comparison between this model and other different MTJ simulation models is presented in Chapter 5. However, even if we solve the problem of lacking a physical model in the intermediate regime, the use of switching probability is still limited in some applications as it cannot provide information regarding the variation of theta. Under the circumstances, we proposed a general-purpose STT-MTJ simulation model based on the Fokker-Planck equation which is introduced in Chapter 4.

Chapter 3

Model based on Fokker-Planck equation

3.1 Fokker-Planck equation

Fokker-Planck equation (FPE) is used for describing the probability density function changing of the velocity of a particle based on the thermal fluctuations. It has been used for describing the Brownian motion. [69] A FPE for the stochastic motion of the spin moment can be derived if the random field is assumed to be sufficiently short-correlated. [70] The thermal fluctuations can be modeled as white noise with a frequency of approximately 10^{-13}s, while the response time of the single domain particle is approximately 10^{-10}s. [37] This satisfies the condition that the random thermal force has a much shorter correlation time than the response time of the magnetization. For STT-MTJ, the FPE is a master equation of the Langevin equation or the s-LLG equation which describes the MTJ's probability density *ρ(θ, t)* of magnetization under the influence of thermal fluctuations, as in the Brownian motion. It was firstly derived from the s-LLG equation which is the LLG equation with thermal fluctuations by Brown in 1963. [37] The FPE can be written as: [71]

$$\frac{\partial \rho}{\partial t} = -\sum_{i=1}^{2} \frac{\partial}{\partial x_i}(A_i \rho) + \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{1}{2} \frac{\partial^2}{\partial x_i \partial x_j}(B_{ij}\rho)\ (x_1 = \theta, x_2 = \varphi) \tag{3.1}$$

$$A_i = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \langle \Delta x_i \rangle \tag{3.2}$$

$$B_{ij} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \langle \Delta x_i \Delta x_j \rangle \tag{3.3}$$

where ρ is the probability density ($\rho(\theta)\Delta\theta$ is the probability for θ to take the value $[\theta, \theta + \Delta\theta]$), Δx_i is the x_i changing in Δt and the statistical averages $\langle\ \ \rangle$ are evaluated by the characteristic of Langevin field (Eq. 1.12). The A and B can be deviated from s-LLG equation and expressed as: [37]

$$A_1 = -\frac{\gamma}{1+\alpha^2}\left(\alpha H_\theta + H_\varphi + \frac{\alpha k_B T}{M_S Vol} cot\theta\right) \tag{3.4}$$

$$A_2 = \frac{\gamma}{(1+\alpha^2) sin\theta}(-H_\theta + \alpha H_\varphi) \tag{3.5}$$

$$B_{11} = \frac{2\alpha\gamma k_B T}{(1+\alpha^2) M_S Vol} \tag{3.6}$$

$$B_{12} = B_{21} = 0 \tag{3.7}$$

$$B_{22} = \frac{2\alpha\gamma k_B T}{(1+\alpha^2)M_S Volsin^2\theta} \tag{3.8}$$

If we consider the effect of spin-transfer torque as a field, it is easy to obtain the s-LLGS equation based 2-D FPE:

$$\frac{\partial\rho}{\partial t} = \frac{\gamma}{1+\alpha^2}\{-\frac{\partial}{\partial\theta}[(H_{eff\varphi} + \alpha H_{eff\theta} + \frac{\eta\hbar I}{2eM_S V}sin\theta + \frac{\alpha k_B T}{M_S Vol}cot\theta)]\rho - \frac{\partial}{\partial\varphi}[(\alpha H_{eff\varphi}$$

$$+H_{eff\theta} + \frac{\alpha\eta\hbar I}{2eM_S V})]\rho + \frac{\partial^2}{\partial\theta^2}\frac{\alpha k_B T}{M_S Vol}\rho + \frac{\partial^2}{\partial\varphi^2}\frac{\alpha k_B T}{M_S Volsin^2\theta}\rho\} \tag{3.9}$$

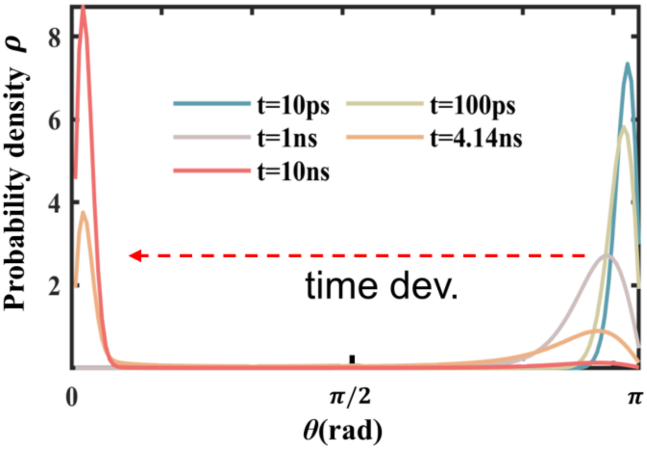


Fig. 3.1. Probability density distribution of theta for different time.

Due to the cylindrical symmetry of the perpendicular STT-MTJ, the 2-D FPE can be reduced to 1-D form and lead to a better computational efficiency. [72] It is worth to note that 2-D FPE method is required when dealing with fields that are not aligned with the easy axis. The 1-D FPE is expressed as: [43]

$$\frac{\partial\rho}{\partial t} = -\frac{1}{\tau_D}\frac{\partial}{\partial\theta}\{[(\frac{I}{I_c} - \frac{H_{app}}{H_k})sin\theta - cos\theta sin\theta + \frac{cot\theta}{2\xi}]\rho\} + \frac{1}{2\xi\tau_D}\frac{\partial^2\rho}{\partial\theta^2} \tag{3.10}$$

$$\tau_D = \frac{1+\alpha^2}{\alpha\gamma H_k} \tag{3.11}$$

$$\xi = \frac{E}{k_B T} \tag{3.12}$$

The method to calculate the initial theta distribution is the same as shown in Chapter 2.1. Figure 3.1 shows an example about probability density distribution of theta (θ) changing based on a constant input current. The results show that the distribution of theta has been shifted from being around π to being around 0.

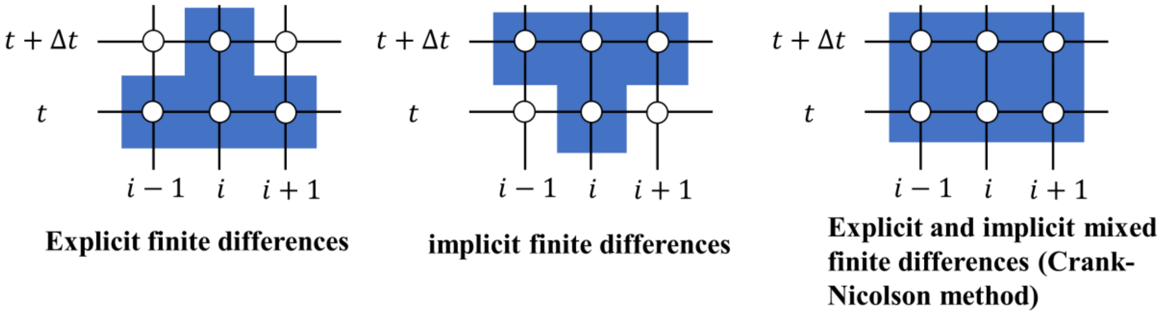
3.2 Problems in Fokker-Planck equation based model

By solving the FPE, we can obtain the switching probability as the time-dependent probability density of theta. However, even though the Fokker-Planck equation does not have the intermediate regime problem, it requires several times longer simulation time compared with the switching probability model. It makes models based on the Fokker-Planck equation less attractive in the applications of large-scale circuits. Additionally, it only calculates the probability density of theta distribution rather than the theta changing information, which limits its applicability. To address this, we propose a framework that can obtain the theta changing information from the probability density of theta.

3.3 Proposed Fokker-Planck equation based model

3.3.1 Numerical method for solving Fokker-Planck equation

To implement a Fokker-Planck equation-based model, the key aspect is to identify a suitable method to solve the Fokker-Planck equation. We use the finite difference method (FDM) [73] to solve the FPE. There are mainly three forms of FDM: the explicit method, the implicit method and the Crank-Nicolson method which can be thought of as a mixture of the explicit and implicit methods. Figure 3.2 shows the basic stencils of each form for a 1D problem. i means the position of the grid and t means time.



Explicit finite differences

implicit finite differences

Explicit and implicit mixed finite differences (Crank-Nicolson method)

Fig. 3.2. The explicit, implicit and Crank–Nicolson stencils for a 1D problem.

While the explicit method can significantly reduce simulation time, it is not sufficiently stable to guarantee the convergence in the simulation and therefore cannot be used in practice. In this situation, we mainly analyze the performance of the implicit method and Crank-Nicolson method. The equations of the implicit and Crank-Nicolson method are shown as following:

$$\begin{cases} \dfrac{\partial \rho_i}{\partial \theta} = \dfrac{\rho_{i+1}^{t+\Delta t} - \rho_{i-1}^{t+\Delta t}}{2\Delta\theta} \\ \dfrac{\partial^2 \rho_i}{\partial \theta^2} = \dfrac{\rho_{i+1}^{t+\Delta t} - 2\rho_i^{t+\Delta t} + \rho_{i-1}^{t+\Delta t}}{(\Delta\theta)^2} \end{cases} \tag{3.13}$$

$$\begin{cases} \dfrac{\partial \rho_i}{\partial \theta} = \dfrac{\left(\rho_{i+1}^{t+\Delta t} - \rho_{i-1}^{t+\Delta t}\right) + \left(\rho_{i+1}^{t} - \rho_{i-1}^{t}\right)}{4\Delta\theta} \\ \dfrac{\partial^2 \rho_i}{\partial \theta^2} = \dfrac{\left(\rho_{i+1}^{t+\Delta t} - 2\rho_i^{t+\Delta t} + \rho_{i-1}^{t+\Delta t}\right) + \left(\rho_{i+1}^{t} - 2\rho_i^{t} + \rho_{i-1}^{t}\right)}{2(\Delta\theta)^2} \end{cases} \tag{3.14}$$

Based on the implicit method, the FPE can be rewritten as:

$$\rho_i^t = -(f_{i-1} + b)\rho_{i-1}^{t+\Delta t} + (1 + 2b)\rho_i^{t+\Delta t} + (f_{i+1} - b)\rho_{i+1}^{t+\Delta t} \tag{3.15}$$

$$f_i = \frac{I}{I_c} sin\,\theta_i - cos\theta_i\, sin\,\theta_i - \frac{H_{app}}{H_k} cos\theta_{\mathrm{i}} + \frac{cos\theta_i}{2\xi sin\theta_i} \tag{3.16}$$

$$a = \frac{\Delta t}{2\tau_D \Delta\theta} \tag{3.17}$$

$$b = \frac{\Delta t}{4\tau_D \xi \Delta\theta^2} \tag{3.18}$$

Based on the Crank-Nicolson method, the FPE can be rewritten as:

$$\begin{aligned} &(-cf_{i-1} + d)\rho_{i-1}^t + (1 - 2d)\rho_i^t + (cf_{i+1} + d)\rho_{i+1}^t \\ &= (cf_{i-1} - d)\rho_{i-1}^{t+\Delta t} + (1 + 2d)\rho_i^{t+\Delta t} - (cf_{i+1} + d)\rho_{i+1}^{t+\Delta t} \end{aligned} \tag{3.19}$$

$$c = \frac{-\Delta t}{4\tau_D \Delta\theta} \tag{3.20}$$

$$d = \frac{\Delta t}{2\tau_D \xi \Delta\theta^2} \tag{3.21}$$

In fact, the Eq. 3.15 and Eq. 3.19 are the systems of linear equations with a tridiagonal matrix as the coefficient matrix. These special systems of linear equations can be solved by using the

Thomas method [74]. The Thomas method is that if a function *Ax=B* satisfy that A is a tridiagonal matrix:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & a_n & b_n \end{pmatrix} \tag{3.22}$$

The variable x can be calculated by:

$$\begin{cases} d_1 = b_1 \\ u_i = \dfrac{c_i}{d_i} \qquad (i = 1,2 \ldots, n-1) \\ d_{i+1} = b_{i+1} - a_{i+1}u_i \end{cases} \tag{3.23}$$

$$\begin{cases} y_1 = \dfrac{B_1}{d_1} \\ y_i = \dfrac{B_i - a_i y_{i-1}}{d_i} \end{cases} (i = 2,3,\ldots, n) \tag{3.24}$$

$$\begin{cases} x_n = y_n \\ x_i = y_i - u_i x_{i-1} \end{cases} (i = n-1, n-2, \ldots, 1) \tag{3.25}$$

Based on these equations, we can calculate the value of $\rho^{t+\Delta t}$. The switching probability can be calculated as:

$$P(t) = \int_0^{\frac{\pi}{2}} \rho(\theta, t) d\theta \; (AP \; to \; P) \tag{3.26}$$

$$P(t) = \int_{\frac{\pi}{2}}^{\pi} \rho(\theta, t) d\theta \;\; (P \; to \; AP) \tag{3.27}$$

To analyze the performance of each method, the first thing is measuring the stability for different grid numbers. We simulated the two methods using different timestep (Δt) and grid number (G) in Matlab. We simulate the MTJ switching time for timestep equal to 10ps and 100ps and compare them with the switching time for timestep equal to 1ps. The simulation result is shown in Fig. 3.3. The results indicate that while the difference in percentage of P for different timesteps is greater for the implicit method than for the Crank-Nicolson method, the absolute values remain within an acceptable range.

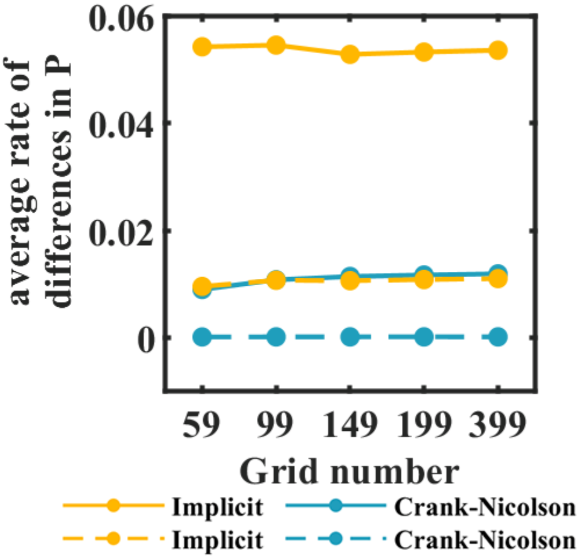


Fig. 3.3. Stability analysis for the implicit method and the Crank-Nicolson method of FDM. The solid line represents the difference in switching probability between Δt =1ps and Δt =100ps. The dotted line represents the difference in switching probability between Δt =1ps and Δt =10ps.

We evaluated the accuracy of the FPE results by comparing them to experimental data [59] and calculated the error rate (error rate=$(t_{experiment}^{P=50\%} - t_{FPE}^{P=50\%})/t_{experiment}^{P=50\%}$) as the averaged difference between the two results during the switching process. The timestep was set to 1ps. In Figure 3.4, it is observed that the results reach stability when the grid number is greater than or equal to 149. For this particular sample, the error rate can be reduced to less than 10% when the grid number equal to 149. Meanwhile, we have computed the total CPU times for the two methods in the simulation which entails multiple switching events of a single MTJ spanning hundreds of microseconds. The results show that, in the same simulation conditions, the Crank-Nicolson method consumes 40% more CPU time compared to the implicit method. Based on the above simulations, we conclude that the optimal configuration for the underlying solver is the implicit method with a grid number of 149.

In the case where we assume the external field to be zero, the Fokker-Planck equation-based model only requires three parameters (I_c, τ_D and ξ), which simplifies the process of parameter fitting using experimental data in the Matlab. Figure 3.5 shows an example of fitting result with the experimental data from [75]. The write error rate is a critical parameter in memory design which is the non-switching probability (1-*P*). In comparison with the s-LLGS-based model, the FPE-based model can obtain a write error rate of less than 10^{-6} in a single simulation. The results indicate that our model fits well with the experimental data even when the write error rate is very low. The parameter we set are I_c=68μA, ξ=44 and τ_D=0.25ns.

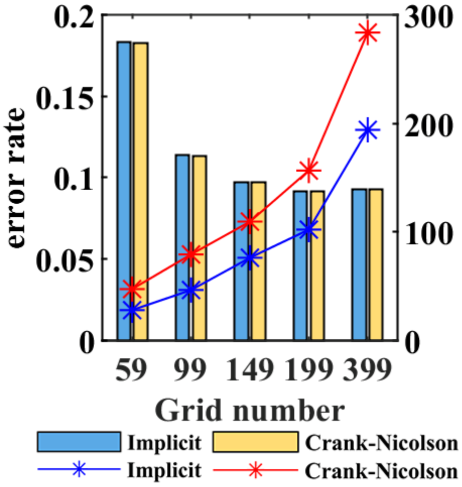


Fig. 3.4. Stability and simulation time comparison with the grid number changed. The bars represent the error rates and the asterisks represent the simulation times.

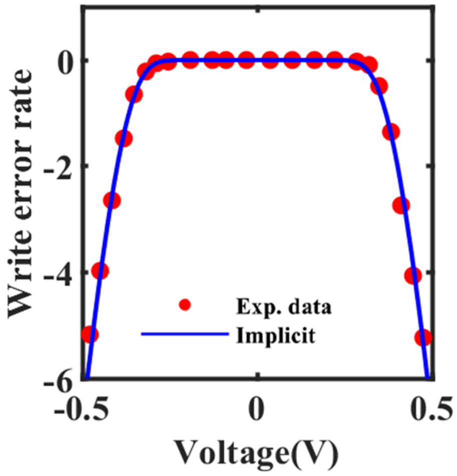


Fig. 3.5. Numerical calculation results of the FPE for write error rate compared with the experimental data [75].

3.3.2 Framework to obtain the theta (θ)

In some applications such as spiking neural network neurons, it is crucial to have access to information about the dynamic changes in the theta between the magnetic moments of the free and pinned layers. In order to overcome the limitation of FPE based models that do not provide theta information, we have developed a framework to obtain the changes in theta. [42] The FPE offers the probability density of theta as opposed to the switching probability in the switching probability model. This feature makes it is possible to monitor of the theta change process in an individual switching event of STT-MTJ for a particular device that is defined by a cumulative probability of the switching event. Figure 3.6 demonstrates the procedure for obtaining information on theta. A number *P* is predefined which can be considered to specify a specific STT-MTJ from the time-dependent cumulative probability. It specifies the most possible trace of magnetization for the device whose switching probability equals to the P at the last time step of the simulation. For different application scenarios, the P can be a constant value for all the STT-MTJs or different random numbers for the STT-MTJs in order to describe the extreme case or the reality. An example of P = 0.5 is shown in Fig. 3.6. The value of theta can be thought of as the cumulative probability of theta being equal to 1-P, which is 0.5 in this case. By implementing this framework for each timestep, we can obtain the trace of theta.

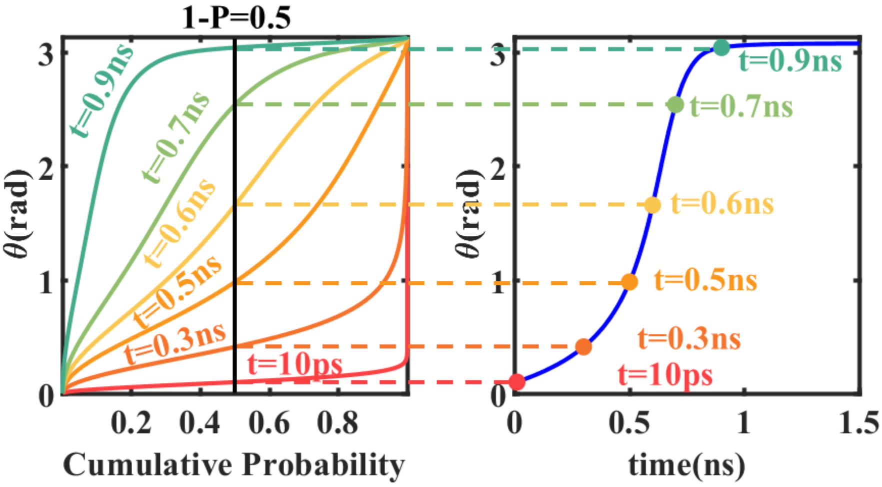


Fig. 3.6. Schematic diagram for obtaining θ from the FPE solutions for the probability density of θ. The curves with the different colors represent those for *t*=0.01, 0.3, 0.5, 0.6, 0.7 and 0.9ns from the bottom to the top, respectively.

We compared the performance of my model with the s-LLGS based model. A 10000 times

Monte Carlo simulation is done for the s-LLGS model and compare the average trace of the theta with the trace obtained by the framework we proposed. We averaged the Monte Carlo simulation cases that reached $\pi/2$ at the same time as our model for different switching probabilities. The comparison results are presented in Fig. 3.7 and demonstrate that our framework agrees with the s-LLGS model. The parameters we used are I_c=8.26mA, ξ=80 and τ_D=0.254ns.

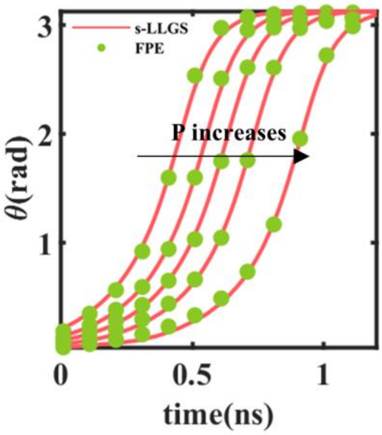


Fig. 3.7. Comparison between numerical calculation results of the FPE and the averaged s-LLGS solutions. From left to right, the lines represent the switching probability P = 0.1, 0.3, 0.5, 0.7 and 0.9, respectively.

We set two modes to adapt to different applications: one includes the theta obtaining framework, and the other does not. Different modes require different resistance models. The resistance model for the mode includes the theta obtaining framework (theta mode) is expressed by: [76]

$$R^{-1}(\theta) = (R_p^{-1} + R_{ap}^{-1})/2 + \frac{(R_p^{-1} - R_{ap}^{-1})cos\theta}{2} \tag{3.28}$$

For the mode which does not include the theta obtaining framework (probability mode), the resistance is considered only to have two different resistance states, rather than being a function of theta. The low resistance state (R_p) is calculated by: [63]

$$R_P = \frac{t_{ox}}{F \times Area \times \sqrt{\varphi_B}} exp(1.025t_{ox} \times \varphi_B) \tag{3.29}$$

The high resistance state (R_{ap}) is calculated by the TMR ratio:

$$R_{ap} = (1 + TMRR)R_p \tag{3.30}$$

In fact, the resistance model of the probability mode is the same as the switching probability model.

Thermal fluctuations affect the initial theta distributions, so we update the random number P to account for this phenomenon when the system is at thermal equilibrium. The stability of the system is measured by the value S which is defined as:

$$S = \frac{\sum_{i=0}^{n} |\rho(\theta_i, t + \Delta t) - \rho(\theta_i, t)|}{\Delta t} \tag{3.31}$$

where n is the total grid number for θ from 0 to π. Because the timestep in the HSPICE is not a constant value just like in the Matlab, the S is corresponding with the Δt for each timestep in HSPICE. If S is smaller than the predetermined critical value, the system is regarded as being in thermal equilibrium and a new random number is generated. At the same time, the probability density is reset to the corresponding thermal equilibrium state.

4.4 Validation results

The model is implemented by Verilog-A and simulated in HSPICE. The equations in the implicit format made it challenging to use the built-in '*idt*' function which is commonly used for integration in Verilog-A. In this situation, I constructed an architecture that uses the Thomas method to solve the linear system of equations described in Eq. 3.15.

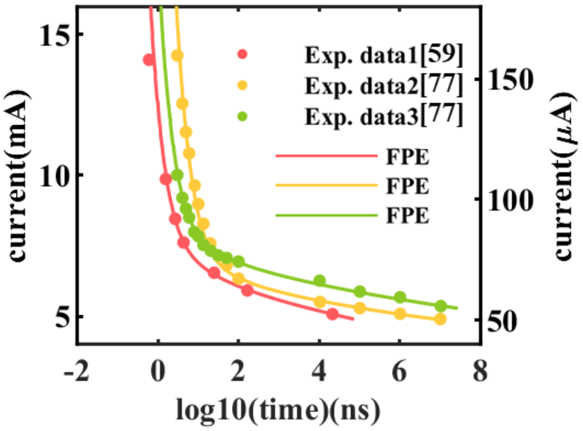


Fig. 3.8. HSPICE simulation results of switching time based on the FPE model for different incident currents compared with the experimental data with the switching probability equal to

50% [59][77]. The unit of data1 is mA. The unit of data2 and data3 is μA.

The model is validated for individual devices. In this simulation, one STT-MTJ was directly connected to a current source. The first validation is for a constant switching probability. The switching time for different incident currents are simulated with the P equal to 50%. The parameters we use are I_c=8.26mA, 80.4μA and 93.2μA, ξ=80, 116 and 111 and τ_D=0.254ns, 1.38ns and 0.42ns for experimental data1, experimental data2 and experimental data3, respectively. The current changes across a broad range including the adiabatic precessional regime, the thermal activation regime and the intermediate regime. The FPE-based model only requires three simulation parameters to match experimental results. The Fig. 3.8 shows that the model work well in all the regimes and consistent with the experimental data [59][77].

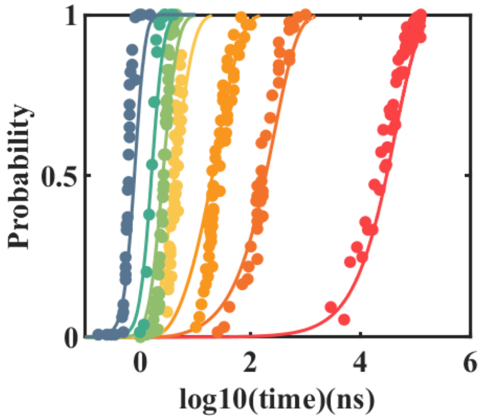


Fig. 3.9. HSPICE simulation results of switching probability based on the FPE model for constant incident currents compared with the experimental data [59]. Solid lines represent HSPICE results and data points represent the experimental data. From left to right, the incident currents are 14.1mA, 9.87mA, 8.46 mA, 7.62 mA, 6.55 mA, 5.92 mA and 5.08 mA, respectively.

The model Is also validated for t"e re'ationship between probability and the time for a constant incident current. Fig. 3.9 shows that the model fitting the experimental data [59] well. Furthermore, even if the results are not compared directly with the changing of the switching probability distribution, validating the correction of the model is sufficient since the probability is the summation of the switching probability density for each time step. The parameters I set are I_c=8.26mA, ξ=80 and τ_D=0.254ns. Figure 3.10 illustrates the resistance characteristics of both modes along with the theta transition process in the angle mode with the predetermined number is 0.9. A constant voltage is input start from t=0. Different resistance modes result in different switching speeds, as dynamic theta causes a change in current.

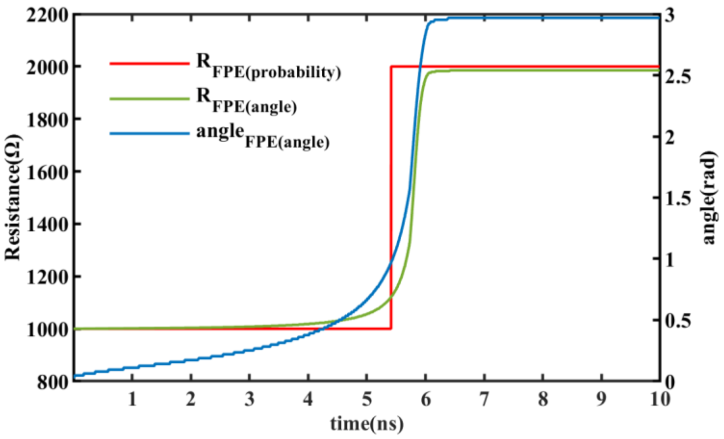


Fig. 3.10. HSPICE simulation results of angle and resistance characteristics for different resistance modes.

In order to validate the FPE based model can work with other standard devices in the EDA tools, I simulate a simple 1T1C1MTJ circuit. The structure of 1T1C1MTJ circuit is shown in Fig. 3.11. I realize a single P state writing and the results are shown in Fig. 3.12. The mode of FPE based model is set as theta mode. The parameters I set are I_{cap}=80μA, ξ=100 and τ_D=0.15ns. The switching probability is set as 80%. The results shows that my FPE based model can work with other standard devices in the EDA tools. The practical applications of FPE based MTJ model in more complex applications will be demonstrated in Chapter 4.

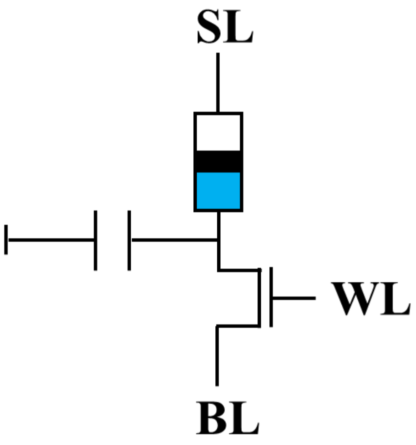


Fig. 3.11. 1T1C1MTJ structure.

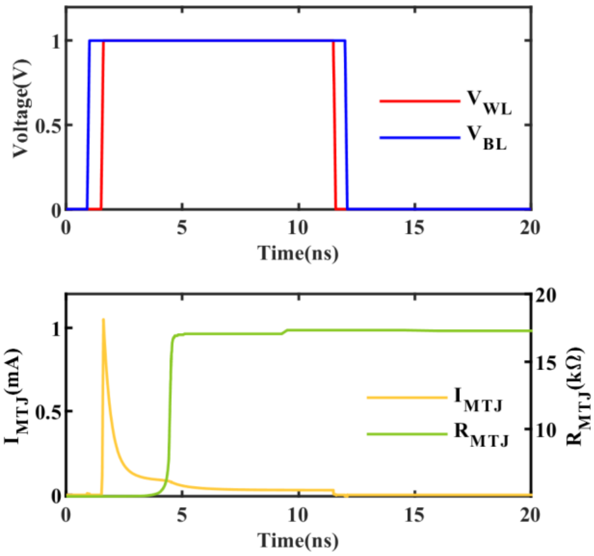


Fig. 3.12. HSPICE simulation 1T1CMTJ structure with P to AP switch. The V_{SL} keeps 0 during the simulation.

Chapter 4

Model application to spiking neural networks

4.1 Introduction

Due to its resistance and stochastic properties, STT-MTJs have been used in various applications. In this chapter, I have utilized STT-MTJs in two distinct types of applications. The first application involves utilizing the resistance property of STT-MTJs in STT-MRAM, whereas the second application involves the magnetization changing and resistance properties of STT-MTJs in a spiking neural network (SNN) which I proposed. [57]

4.2 STT-MRAM

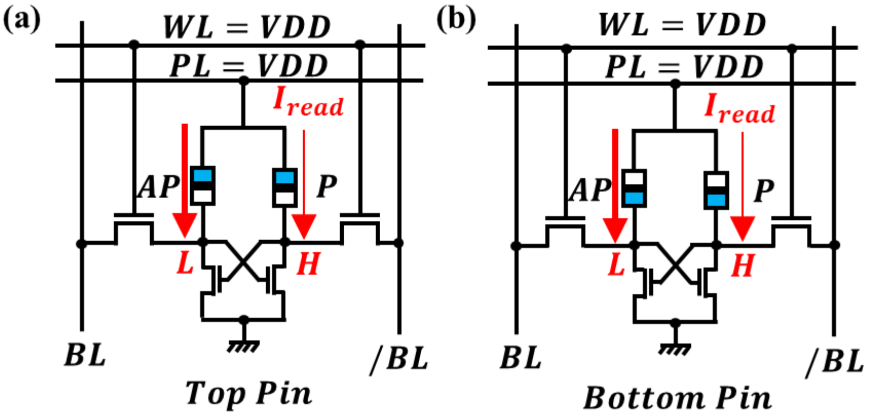


Fig. 4.1. The top pinned (a) and the bottom pinned (b) structures of the 4T2MTJ MRAM cells with power-on operation, respectively. The bule color represents the pinned layer.

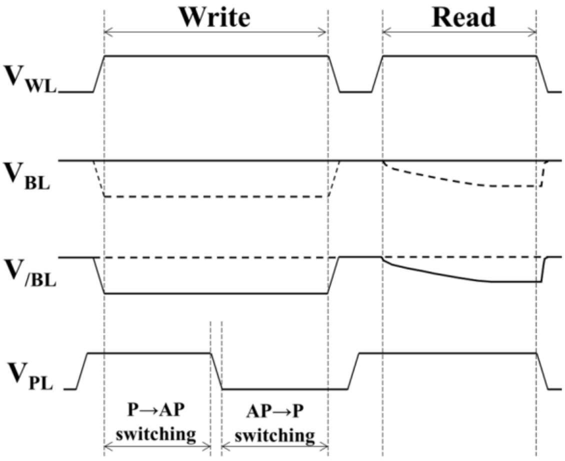


Fig. 4.2. Read/write operations of the 4T2MTJ cell.

There are many kinds of STT-MRAM cell. Here we deal with the 4T2MTJ cell [35], [78], again. The structure of the 4T2MTJ cell is shown in Fig. 4.1. The control signals of read and write operations are shown in Fig. 4.2. For the write operation, it needs two steps to separately switch the two MTJs because the different state switching needs different directions of input current. For the read operation, it needs to pre-charge the BL and /BL. After that, the voltage of the node which connects the high resistance will decrease during the read operation. In Fig. 4.2, we used dotted line and solid line to show the signals of writing different information into the cell and different results of read operation.

A noteworthy point is the type of STT-MTJ connection in this structure. There are two types of connections, namely the top-pinned structure and the bottom-pinned structure. The top-pinned structure refers to the configuration in which the pinned layer is connected to the power line (PL), while the bottom-pinned structure refers to the configuration in which the free layer is connected to the PL. During the read operation, the data in the STT-MTJs is loaded to the latch circuit. The node connecting to the P state STT-MTJ can reach a high voltage. A large current flows through the STT-MTJ at the AP state from the PL to the ground, whereas only a very small current flows through the STT-MTJ at the P state. The state changing of STT-MTJ depends on not only the intensity and duration of the current but also the direction of the current. To ensure that data is not corrupted in the STT-MTJ during wake-up operations, the 4T2MTJ cell must adopt the top-pinned structure because a large current flowing from the free layer to the pinned layer can switches the AP state to the P state.

I simulate the 4T2MTJ MRAM by my switching probability based model and the probability mode of the FPE based model in the HSPICE. The mode assumes that the STT-MTJ only has two resistance states. The simulation results are shown in Fig. 4.3 and Fig. 4.4. The top figure of Fig. 4.3 and Fig. 4.4 shows the control signals. At the beginning and the end, the read operation was performed twice to verify the effectiveness of the write operation in between.

For switching probability based model, I perform a simulation with switching probability equal to 50%. The switching of STT-MTJ takes place in the black dotted circles in the bottom figure of Fig. 4.3. For FPE based model, total 20 times simulation results of the BL and /BL under the same condition are depicted in the bottom figure of Fig. 4.4. The circuit demonstrates stochastic characteristics during the write operation due to the stochastic property of STT-MTJs. The switching of STT-MTJs takes place at various time points, as shown by the black dotted circles in the bottom figure of Fig. 4.4. The fact that the results of the two read operations were opposite to each other demonstrates the successful execution of the write operations.

The simulation results in the Fig. 4.4 prove that our FPE based model can work with standard CMOS devices in the EDA tools. Because of the resistance property, the STT-MTJ can be used in memory.

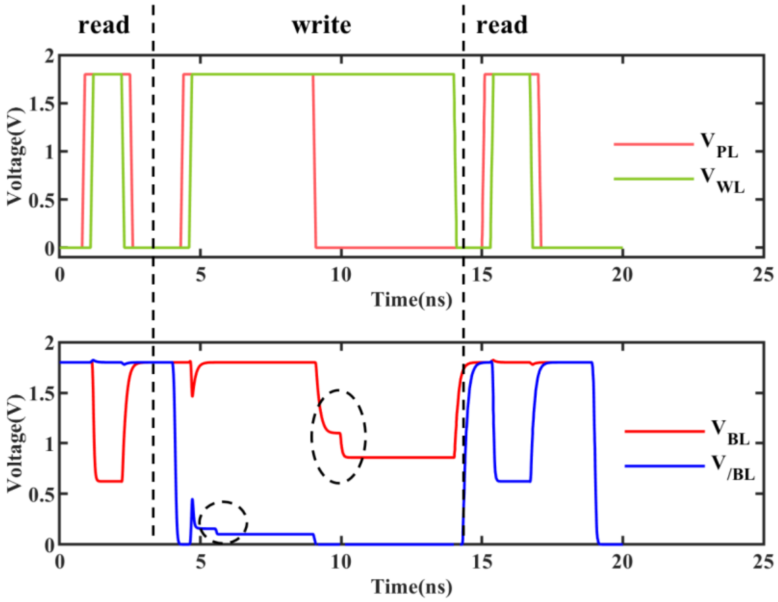


Fig. 4.3. 4T2MTJ STT-MRAM cell Simulation based on switching probability based model.

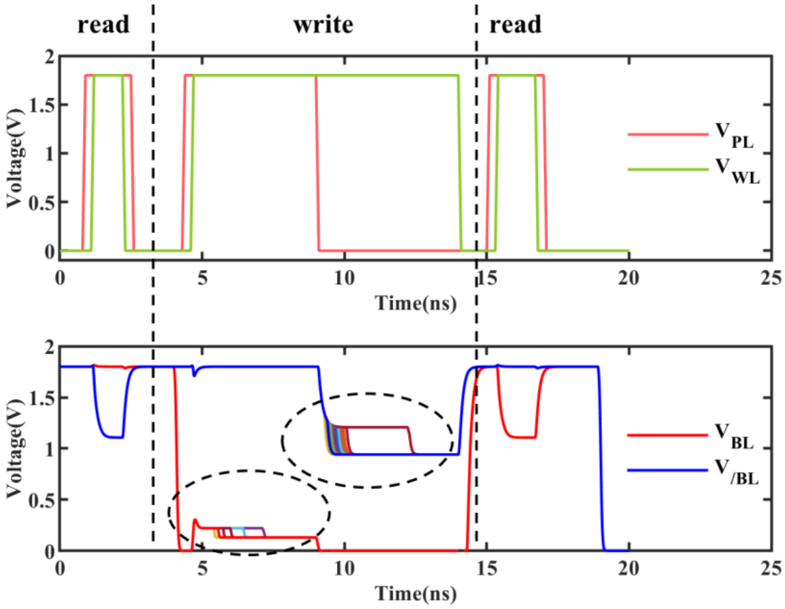


Fig. 4.4. 4T2MTJ STT-MRAM cell Simulation based on FPE based model.

4.3 Spiking neural networks (SNNs)

I propose an STT-MTJ based leaky integrate-and-fire (LIF) neuron with auto-reset function. I also propose an STT-MTJ based synapse which solves the problem of small *TMRR* and resistance. An SNN s designed to validate the performance of the proposed neuron and synapse, and the results indicated that our design achieved high energy efficiency.

4.3.1 Basic of SNN

In the past decade, there have been significant advancements in deep neural networks (DNNs) which are now extensively applied across diverse domains such as image recognition [79], autonomous driving [80], and board games [81]. They have demonstrated a performance level comparable to, or in some cases, surpassing that of humans.[81]-[82] Despite significant advances in deep neural networks (DNNs), the computational demands for performing inference remain huge, as a vast number of multiply-accumulate operations are required. For instance, AlphaGo's victory over the human world champion in the game of Go was achieved at the cost of consuming energy orders of magnitude higher than that of the human player. While several approaches, such as the use of computing-in-memory architecture, have been proposed to reduce the energy consumption by minimizing data transmission between memories and processors, the power consumption of DNNs still remains significantly higher than that of the human brain, which consumes only 20 watts. [83]

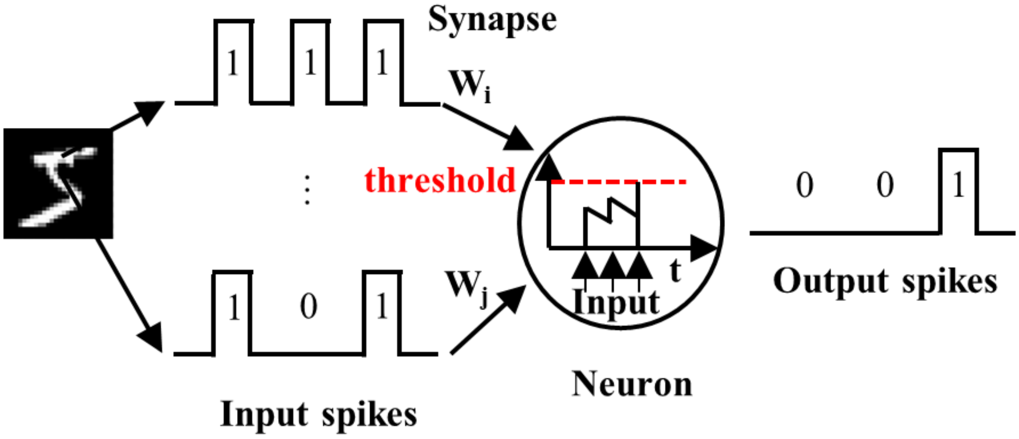


Fig. 4.5. The basic process of spiking neural network.

The inefficiency of DNNs has led to increasing interest in SNNs which are inspired by biological neural networks. SNNs are event-driven, which is a key difference from deep neural networks. [84] Neurons communicate via digital pulses instead of analog values. This approach capitalizes on the temporal representation of information and is expected to improve efficiency while consuming less energy. [85] SNNs offer distinct advantages for tasks that require real-time processing, such as speech and motion analysis. [84]

Figure 4.5 provides a simple example of information processing in SNNs, where interconnected neurons communicate via synapses. Based on an encoding method, the information in the input pixels transfers to some discrete spikes. Spiking neurons integrate incoming spikes and generate output spikes to the next layer. The rules dictating spike generation differ depending on the specific neuron type. The recognition result is determined based on the output neuron with the highest firing rate or the first spike. The final recognition results are determined by either the firing rate of the neurons in the output layer or the sequence of their firing. While the example demonstrates the application of SNNs to static image recognition, it is worth noting that SNNs are more advantageous in handling time-related events, such as dynamic image processing.

Neurons play a crucial role in SNNs because they receive, process and transfer the information in the SNNs. There are several existing neuron models such as the Hodgkin-Huxley (HH) model [86], the leaky integrate-and-fire (LIF) model [87], and the Izhikevich model [88]. Although their characteristics may differ, the core of these models is to accumulate different time-varying input signals and output when a threshold is reached.

Some studies have attempted to leverage the performance advantages of spiking neural networks (SNNs) by utilizing non-von Neumann architectures such as in-memory computing. [89]-[97] The traditional complementary metal-oxide-semiconductor (CMOS) technology requires the large size capacitors to integrate the spikes, which makes it hard to increase the density of the circuit. [98] The research progress on novel memory devices has made it possible to design capacitorless SNNs and provides better energy efficiency. [99] Under the circumstances, I utilize the STT-MTJ to implement the SNNs.

4.3.2 STT-MTJ based neuron

The LIF neuron is one of the most widely used neuron models in current applications. The performance and the equivalent circuit of LIF neuron are shown in the Fig. 4.6. The *u(t)* is expressed as: [87]

$$\frac{du}{dt} = -\frac{u(t)}{\tau_m} + \frac{J(t)}{C} \tag{4.1}$$

where u, R, J and τ_m (=RC) are a membrane potential, a resistance, the incident current into the circuit and a time constant of the neuron, respectively. The neuron receives the input spikes and generates an output spike when the membrane potential u reached the V_{th}. The membrane potential will reset to the initial value after the neuron fires the output spike. The reset mechanism is not shown in the Eq. 4.1 and Fig. 4.6 (a).

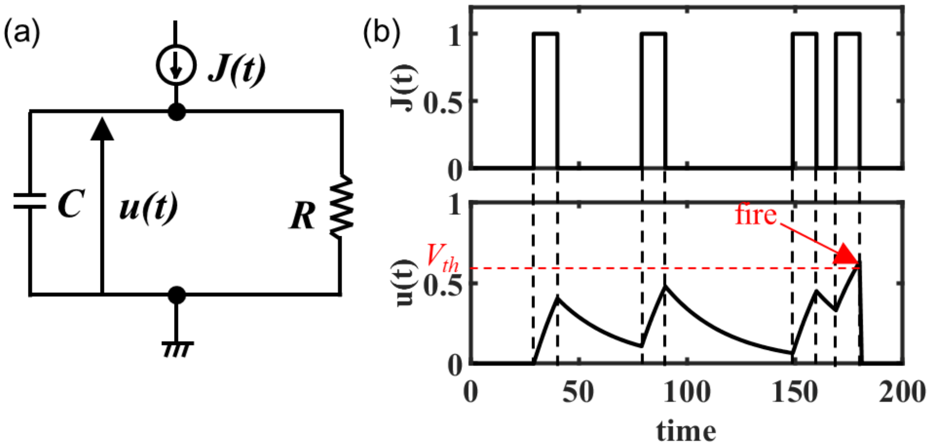


Fig. 4.6. (a) A simplified equivalent circuit of a leaky integrate-and-fire (LIF) neuron. (b) The response of the LIF neuron to incident currents. The spike firing with reset mechanism shown in (b) is not modeled in (a).

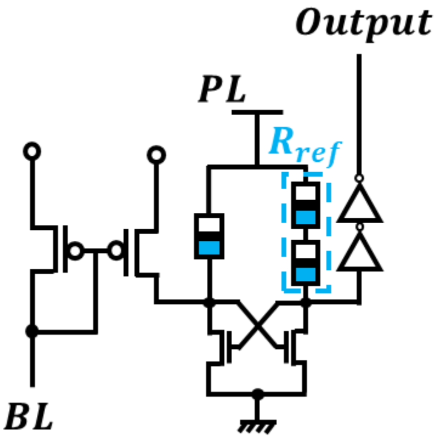


Fig. 4.7. Structure of the auto-reset neuron.

The magnetization switching process is similar to the membrane potential in the LIF neuron, which makes it possible to use STT-MTJ as the core device in the LIF neuron design. The magnetization in STT-MTJ replaces the membrane potential. Comparing with the s-LLGS equation, the term $-u(t)/\tau_m$ can be replaced by the terms associated with precession and damping, which enables the magnetization to return to its initial state before it reaches half π. The π/2 can be considered as the threshold voltage (V_{th}) in the LIF neuron. The $J(t)/C$ can be replaced by the spin-transfer torque term which makes the magnetization change the direction.

The structure of the auto-reset neuron I proposed is shown in the Fig. 4.7. The relationship between the R_{ref} and the STT-MTJ which is shown in the left determined the output signal. In order to implement this, the R_{ref} is set to the value between R_p and R_{ap} (R_p<R_{ref}<R_{ap}). When $TMRR_0$ is set as 200% in this work, it can use two STT-MTJs in series as the reference. The relationship satisfies R_p<R_{ref}=$2R_p$ <R_{ap}=$3R_p$. Three STT-MTJs and two cross-coupled NFETs constitute the core part of the neuron. The initial state of the STT-MTJ is P state. The magnetization in the STT-MTJ will be switched by the input from the BL. When the STT-MTJ switches to the AP state, the neuron will output a high voltage which can be consider as the fire.

Different from the 4T2MTJ cell, the LIF neuron does not need the top pinned structure to keep the information during the read operation. In the opposite, the bottom pinned structure can provide better performance. In LIF neuron, a reset operation is necessary after one output spike is generated. For the neuron which is shown in the Fig. 4.7, it means that the STT-MTJ should be reset to the P state. Under the circumstances, I utilized the bottom pinned structure to realize the state of STT-MTJ automatically resets to the P state after firing rather than keeps AP state, which is exactly the error in read operation of 4T2MTJ.

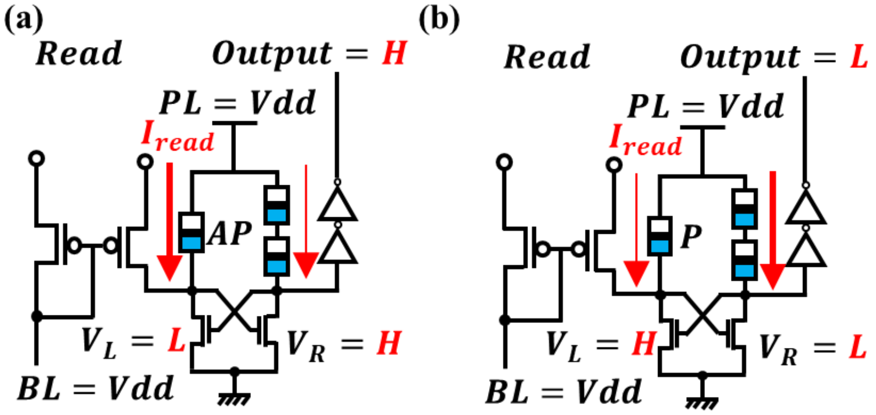


Fig. 4.8. Read operations of auto-reset neuron.

Table 4.1. Parameter of STT-MTJ used in BSNN

Parameters	**Symbol**	**Value**
Critical current (P to AP)	*Icp*	93.1μA
Critical current (P to AP)	*Icap*	37.4μA
TMR ratio	$TMRR_0$	200%
Resistance of P state	R_p	6.5kΩ
Relaxation time	τ_D	0.2ns

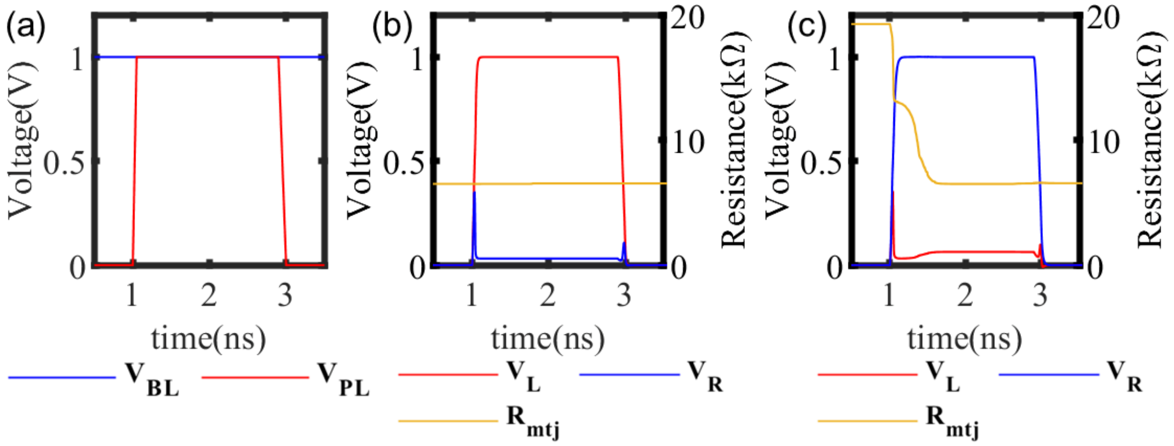


Fig. 4.9. HSPICE simulation results of read operations. (a) The control signals of auto-reset neuron for read operation. (b) Results of read operation when STT-MTJ is AP state. (c) Results of read operation when STT-MTJ is P state.

The read operations based on different STT-MTJ states are shown in the Fig. 4.8 and the HSPICE simulation results of the read operation are presented in Fig. 4.9. V_L and V_R are used to define the storage nodes. The parameters of STT-MTJ are shown in Table 4.1. The simulation is conducted using the 45nm technology node, while the CMOS transistor simulations are based on the Predictive Technology Model (PTM) model [100]. The output of the neuron is V_{dd} when the STT-MTJ is in the AP state, indicating logic 1. A large enough current flows through the STT-MTJ, switching the state to the P state. In the same read operation, the output can remain constant even if the state of STT-MTJ is changed because the information has already been latched. The output of the neuron is 0V when the STT-MTJ is in the P state, indicating logic 0, and the resistance remains constant. Because the V_L is at V_{dd}, there is only a tiny current flowing the STT-MTJ. This tiny current has practically no impact on the magnetization changing. The simulation results shown in Fig. 4.9 demonstrate that the auto reset neuron can generate spikes depending on the state of STT-MTJ and automatically reset to its initial state while maintaining the output.

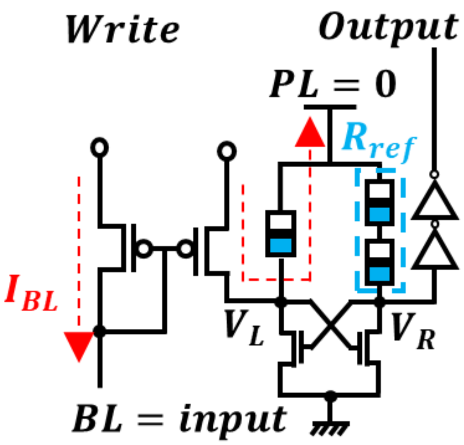


Fig. 4.10. Write operation of auto-reset neuron.

The write operation is shown in Fig. 4.10. A current mirror is used to generate the input current and isolate the synapse array during the read operation. A simulation was conducted to validate the performance of the auto-reset neuron by observing changes in the resistance of the STT-MTJ for different input currents. The Fig. 4.11 shows that the neuron integrates the input information, which leads to that the STT-MTJs switch the states in different write operations for different input currents. Additionally, the relaxation of magnetization to the thermal equilibrium state, observed in the simulation results, can be considered as a leakage in the LIF neuron.

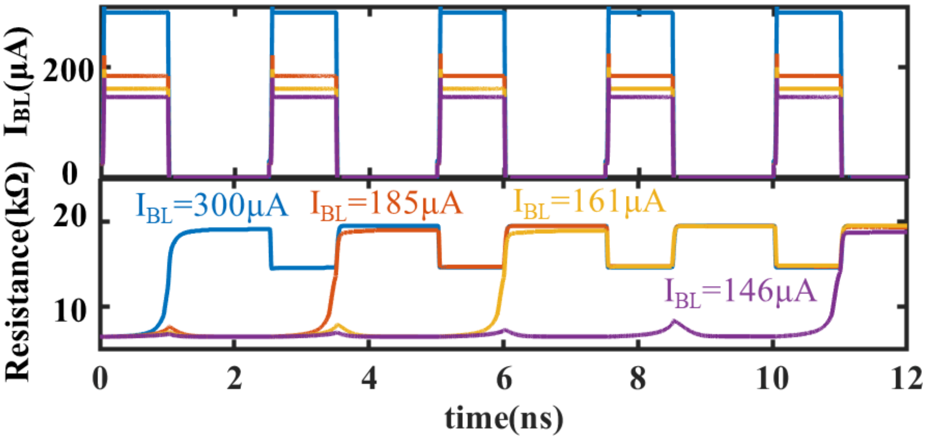


Fig. 4.11. HSPICE simulation results for write operations of auto-reset neuron with different input currents. The voltage signal V_{PL} is maintained at 0V throughout the simulation.

The auto-reset neuron mainly has three merits. Firstly, the removal of independent reset cycles enhances the throughput of SNNs. The second merit of my design is the elimination of the circuits used to detect spike occurrence and control the reset operation, which is especially beneficial for neurons that employ emerging devices requiring a reversal in current direction for resetting their state. To illustrate this, we compare with [94] which utilized the spin-orbit torque MTJ (SOT-MTJ) to implement SNN neuron. The structure of the SOT-MTJ and neuron form [94] are shown in the Fig. 4.12. The direction of the current flows through the heavy metal determines the SOT-MTJ switching. The control signals comparison is shown in Fig. 4.13. When compared to my proposed neuron design, the neuron presented in [94] not only requires an additional independent reset operation and control circuit, but also necessitates a latch circuit to preserve the spike until the completion of the write operation for the neuron in the next layer.

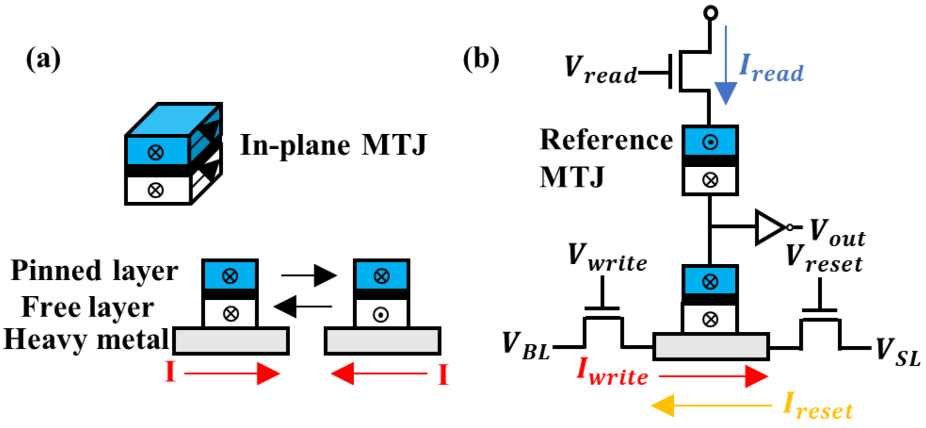


Fig. 4.12. (a) Structure of spin-orbit torque MTJ. (b) Neuron structure from [93].

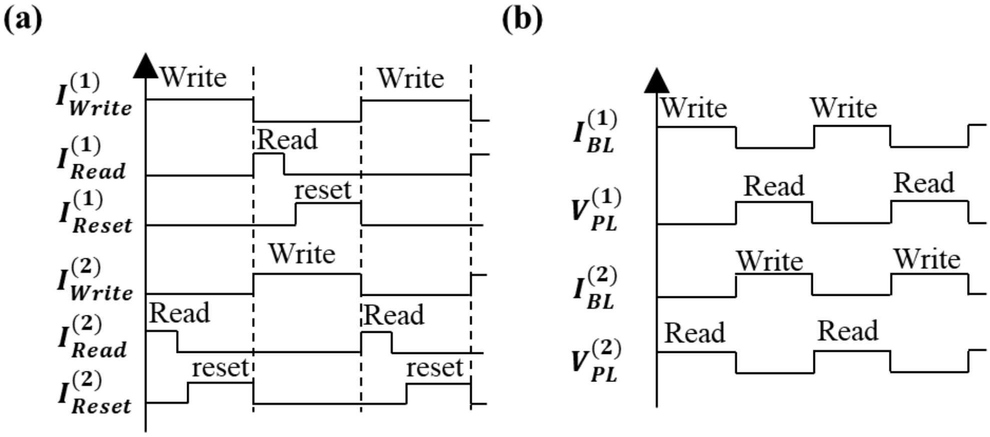


Fig. 4.13. (a) Control signals of the neuron from [94]. (b) Control signal of the auto-rest neuron.

The third merit is that the auto-reset neuron reuses the energy for the read operation. The energy consumed during read operations cannot be avoided. Whether or not an independent reset operation is performed, there will always be a current flowing through the STT-MTJ during read operations. By repurposing this current to reset the state of the STT-MTJ, the overall energy consumption can be reduced.

4.3.3 STT-MTJ based synapse

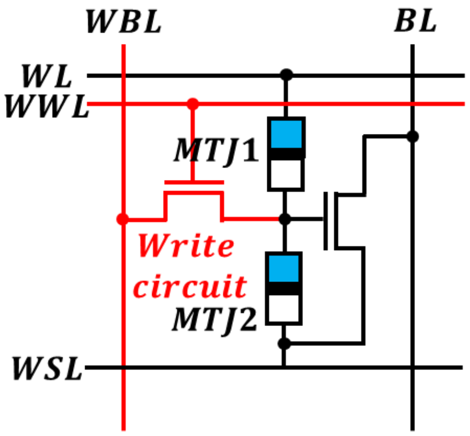


Fig. 4.14. Structure of the proposed STT-MTJ based synapse with a write circuit.

As I introduce in the previous chapters, the value of *TMRR* is only at the range of 100%~300%. It is not a very large value for the synapse comparing with other devices just like RRAM. Meanwhile, the absolute resistance value is also small. For large-scale SNNs, the effective resistance of the synapse array may be less than several thousand ohms, which poses challenges to signal reading and processing. The operational amplifiers with negative feedback can solve the problem. However, the long delay time which avoids the oscillation limits it is used with STT-MTJ based neuron because the relaxation time of STT-MTJ is only about 1ns.

Under the circumstances, I proposed a STT-MTJ based synapse to solve the problems. The structure of the 2T2MTJ synapse is shown in Fig. 4.14. Based on the state of two STT-MTJs, the NFET can be in three different conducting states. The relationship between the states of STT-MTJs and the conducting state is shown in Fig. 4.15. In this work, I implemented a binarized spiking neural network and I only used two different conducting state (0.3V_{dd} and 0.7V_{dd}) to implement binarized weights. The conducting state with 0.3V_{dd} is considered as the logic 0 and the conducting state with 0.7V_{dd} is considered as the logic 1.

(a)

	R_p	R_{ap}
R_p	0.5Vdd	0.7Vdd
R_{ap}	0.3Vdd	0.5Vdd

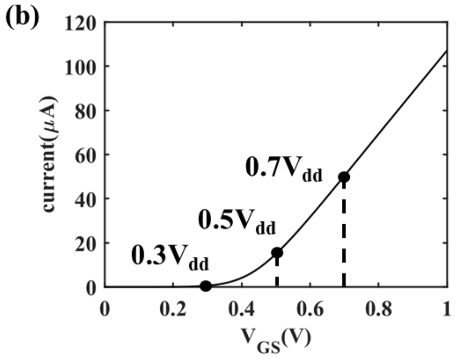


Fig. 4.15. (a) The relationship between the states of STT-MTJs and the voltage at the node between two STT-MTJs. (b) Conduction properties of the NFET at the three different gate voltages realized by the resistance combinations for the two STT-MTJs in the synapse shown in (a).

As shown in Fig. 4.7, the output current generated by the current mirror in the neuron. A comparison between 2T2MTJ and 1T1TMJ is shown in Fig. 4.16. The structure of 1T1TMJ is shown in Fig. 4.16 (b). The resistance of P state in STT-MTJ is 6.5kΩ. The output current is a function of the weight equal to logic 1. I assumed total 10 WL is activated at the same time in the simulation. Because the lack of operational amplifier, the voltage of BL is changing, which leads to the nonlinearity between the number of synapses with logic 1 and the current input to the neuron. The output current range of the 2T2MTJ synapse is from 0 to 110μA. The range difference of 1T1MTJ structure is only 23μA. Even if the resistance of P state is set as a larger value (50kΩ), the output current range is still much smaller than the 2T2MTJ synapse. This suggests that achieving sufficient precision to distinguish inputs within a single write operation is challenging. Especially in situations where operational amplifiers cannot be used.

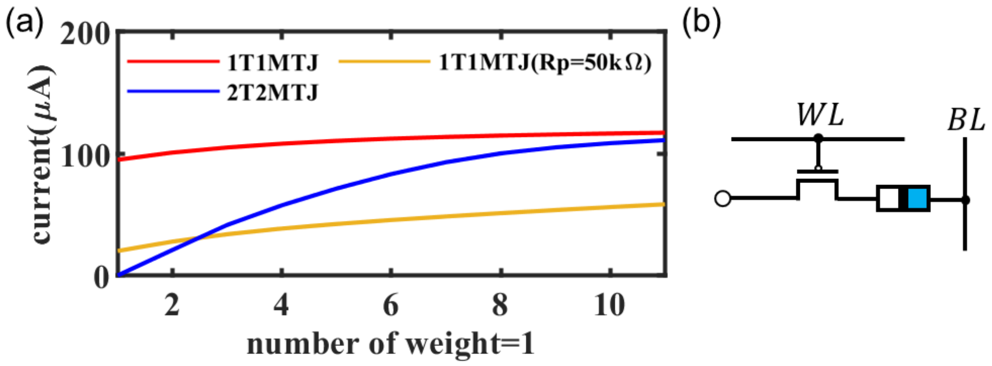


Fig. 4.16. The current flowing through the left STT-MTJ in the neuron for different

multiply-accumulate (MAC) values (the number of the synapses with weight “+1”) when total number of the synapses is 10 for the proposed synapse (blue line). The other two lines (red and orange) are for the 1T1MTJ synapse for Rp=6.5 kΩ and 50 kΩ, respectively. (b) Structure of the 1T1MTJ synapse.

4.3.4 Spiking neural network implementation

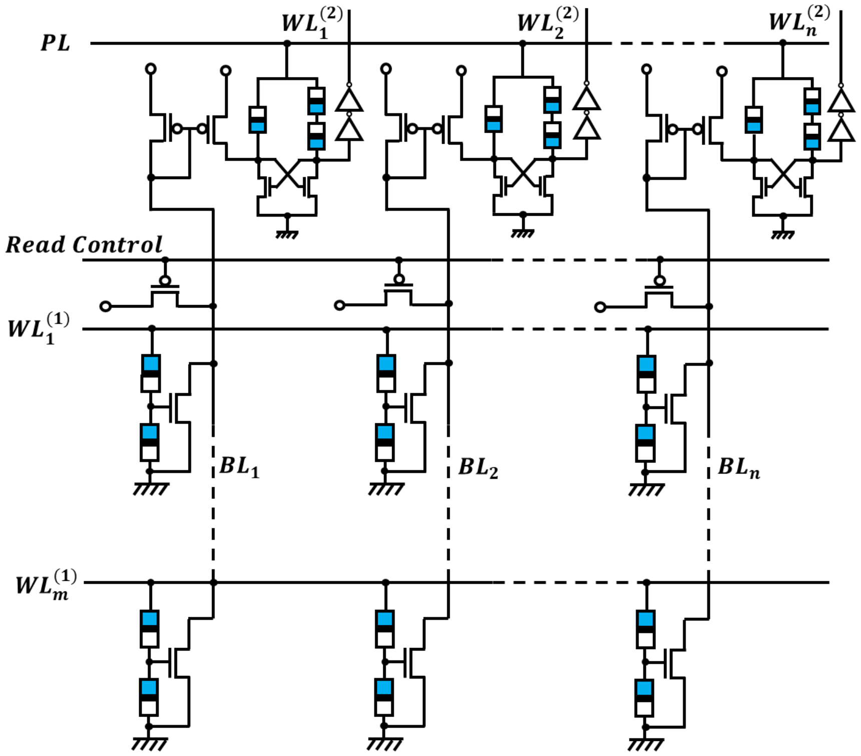


Fig. 4.17. The structure of the single layer BSNN with the auto-reset neurons and the 2T2MTJ synapses. The write circuits of the synapses are not shown in the structure.

This section validates the performance of the proposed neuron and synapse at a system level. I implemented a binarized spiking neural network (BSNN) on the standard MNIST dataset [101], which is used for handwritten digit recognition. The frame information of the handwritten digits is transferred to discrete spikes by the Poisson encoding [102]. The Poisson encoding is expressed as

$$P_{poisson} = \frac{(r\Delta t)^n}{n!} e^{-r\Delta t} \tag{4.2}$$

where r, Δt and n are firing rate, timestep and the probability number of spikes during a time interval of length Δt. By mapping pixel intensities onto different firing rates, we obtain Poisson encoding probabilities for each input point. At each timestep, I compare the probabilities of Poisson encoding with randomly generated numbers between 0 and 1 to determine whether to generate an input spike.

I created a single-layer BSNN with 784 input neurons and 400 output neurons. The structure of the BSNN is shown in Fig. 4.17, where *m*=784 and *n*=400. The write circuit of the synapse is not shown in this figure. During the read operation, the read control signal is set to ground (GND) to isolate the synapse array from the neurons. During the write operation, it is set to *Vdd*. Because of the auto-reset function, the BSNN does not require additional external circuits for resetting. The Poisson encoding is a type of rate encoding method. Therefore, the recognition results are determined by comparing the firing rates of $WL_i^{(2)}$ shown in Fig. 4.16. The neuron with the highest firing rate is considered the winner and the corresponding digit which is predetermined during the training phase to be used as the recognized result. This step implements in a post-processor of software in MATLAB.

The BSNN is trained by the spiking-timing-dependent plasticity (STDP) algorithm [103]. The algorithm adjusts the weights of synapses by the relative timing of output spikes and input spikes. The classical STDP algorithm is expressed as: [104]

$$\begin{cases} \Delta\omega = A^{+}e^{-\frac{\Delta t}{\tau_{+}}} & \Delta t > 0 \\ \Delta\omega = A^{-}e^{\frac{\Delta t}{\tau_{-}}} & \Delta t < 0 \end{cases} \tag{4.3}$$

Where A^{+}, A^{-}, τ_{+} and τ_{-} are the learning rate of potentiation/depression and time constant of potentiation/depression. The process is performed like shown in Fig. 4.18.

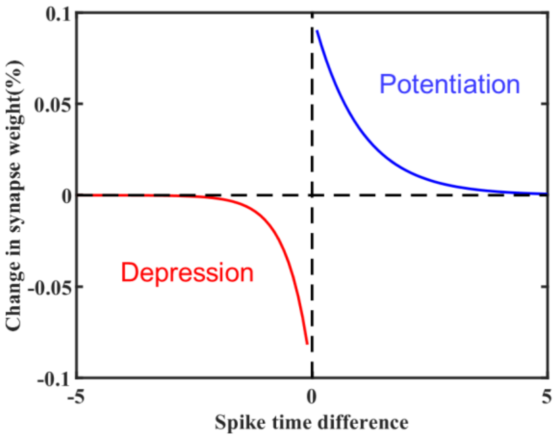


Fig. 4.18. Schematic diagram of spiking-timing-dependent plasticity algorithm

The algorithm is for the full precision weights and I directly binarized the trained weights.

The training result of the weight matrix after binarization is shown in Fig. 4.19. Here I only show the weights of the first 100 neurons as an example. Each digit is composed of 28×28 grids. Based on the weight matrix, it is easy to find each neuron separately represents a digit. I determine the success of this recognition by comparing the digit of the sample with the digit corresponding to the neuron with the highest firing rate. BSNN can reach 82% test accuracy in the software with the standard LIF neuron and linear binarized synapse. In this simulation, I used total 20 cycles of read and write operations. The duration of read and write operation is 1ns and there is a 0.25ns transition between the read and write operations.

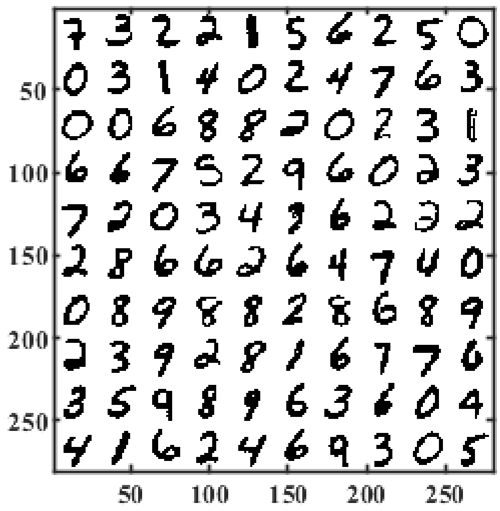


Fig. 4.19. Binarized weight matrix (28×28) of 100 neurons in a 10 by 10 grid.

The simulation results of a single output neuron are shown in Fig. 4.20. Figure 4.20 (a) and (b) are simulated based on the same situation and stochastic characteristics are shown. The output of a single neuron demonstrates that my design has effectively achieved the intended functionality. The input spikes are integrated by the neuron during the write operation which sets the voltage of the $V_{PL(last\ layer)}$ to V_{dd}. When the state of STT-MTJ changes to the AP state, the neuron outputs a spike and resets to the initial state during the read operation which sets the voltage of the $V_{PL)}$ to V_{dd}.

To better demonstrate how SNN performs recognition tasks, Figure 4.21 presents an example of the output voltages of different neurons in an image recognition task. I selected ten neurons representing different digits to show their output when recognizing the digit “1”. The input figure is shown in the left of the Fig. 4.21. The weight matrix for each neuron is displayed either at the top or bottom of the output figure. It can find that the neuron which represents "1" has the highest firing rate.

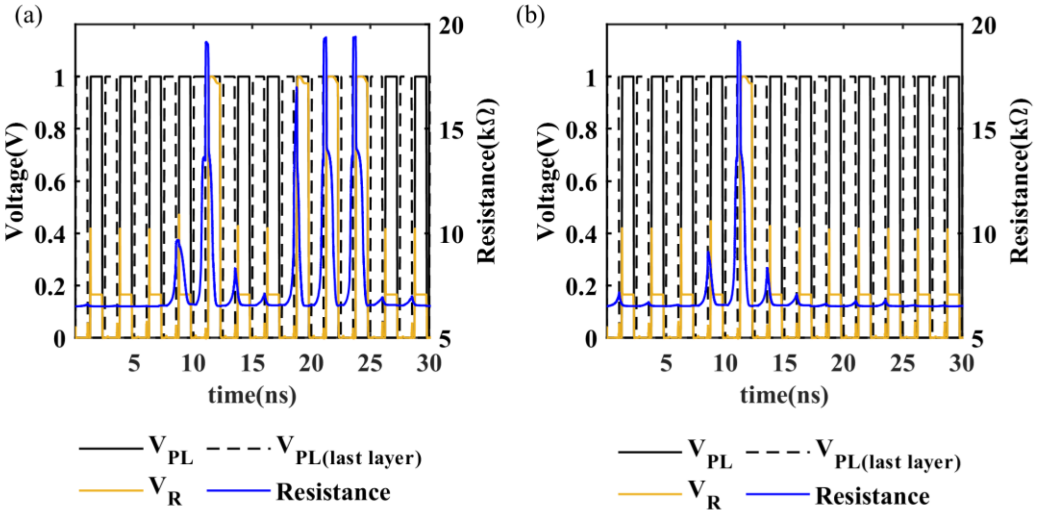


Fig. 4.20. Simulation results of a single output neuron in the BSNN. (a) and (b) are simulated based on the same situation.

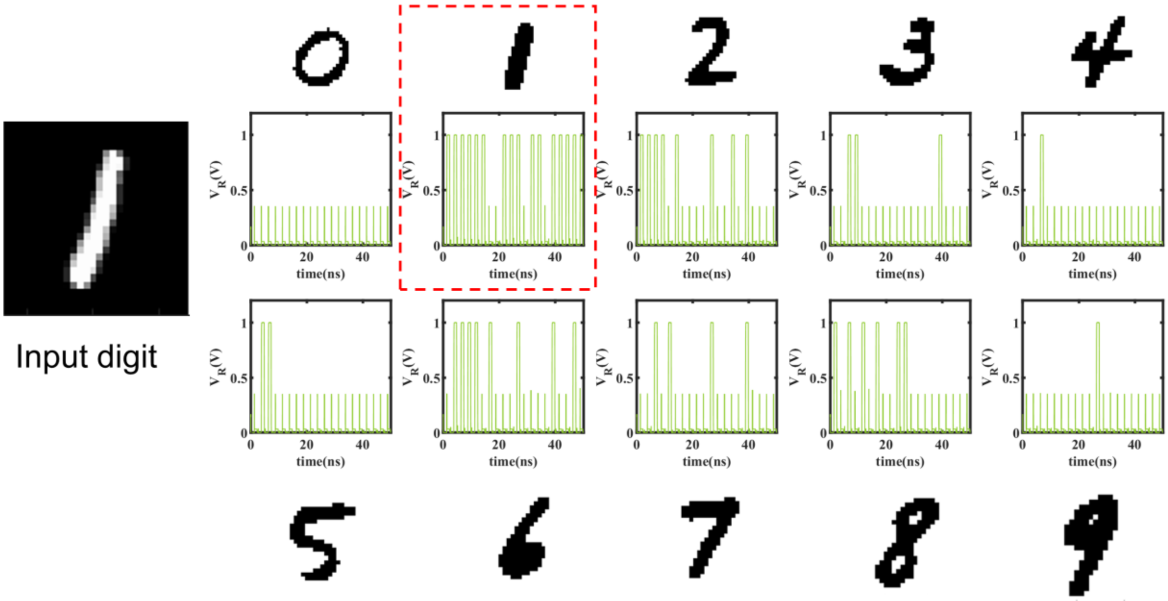


Fig. 4.21. Simulation results of 10 neurons which represent digit "0" to "9". The neuron represents the digit "1" has the highest firing rate. The input figure is shown in the left. The weight matrix for each neuron is displayed either at the top or bottom of the output figure.

The accuracy of the recognition was 75% for 1200 patterns. The accuracy loss is from the mismatch between the training algorithm and the hardware design. The training is based on the standard LIF neuron and full precision synapse.

There are some methods can improve the accuracy such as using a suitable training algorithm based on the characteristic of STT-MTJ based BSNN, increasing the size of the network. Adding some techniques including winner-take-all [105] or classification layer [106] also works for increasing the accuracy.

I tried to add one more SoftMax readout layer. The amount of output spikes for each neuron is counted and used as the input to the SoftMax readout layer. The fully connected SoftMax readout layer is trained on the whole dataset. The training uses the Adam optimizer and cross-entropy loss function. [105] With the SoftMax readout layer, the test accuracy can increase to 88% in the software and 83.71% for the HSPICE results (1200 patterns). The accuracy is enough to prove that my neuron and synapse can be used in the SNNs.

4.3.5 Energy efficiency

The energy efficiency of neural networks is a key performance indicator. For SNNs, people use the synaptic operation (SOP) to fairly compare the energy efficiency among different SNNs. The SOP is defined as a source neuron generates a spike which then flows through a non-zero synapse and is added to the membrane potential of the neuron in the next layer. It is unfair to directly compare the total inference energy, because the accuracy level or the datasets are variable in different SNNs. Under the circumstances, energy per SOP which means the energy per fundamental operational unit us a fairer metric to compare the energy efficiency among different SNNs. The merits of energy per SOP are it minimize the impact of weight values or the network size which have huge impact on the accuracy of SNNs.

There is another similar indicator to evaluate the energy efficiency of the neuron: energy per spike. Energy per spike is defined as the total energy consumption includes the read, write and reset operation for a neuron to generate a spike to the next layer. It evaluates the energy consumption of a fundamental operation unit in a neuron. The mainly difference between the energy per spike and energy per SOP is that the energy per spike only evaluate the energy in neuron and guarantee one spike is generated.

The resistance of STT-MTJ is greatly affects the power consumption of the BSNN. When the R_P of STT-MTJ is set as 6.5kΩ, the energy consumption of the SNN is 0.23pJ/SOP. If the R_P of STT-MTJ increases to 7.5kΩ, the total energy can reduce to 0.19pJ/SOP. The limitation of the STT-MTJ resistance is the read current of auto-reset neuron. The read current must be

Table 4.2. Performance Comparison with Other Works

	This work	***TED 2016[95]***	***TCAS-I 2020 [106]***	***ISCAS 2021 [90]***	***TETCI 2018 [97]***	***ASSCC 2020 [91]***	***ISSCC 2022 [92]***
Technology Node (nm)	45	45	90	28	45	65	28
Circuit Type	STT-MTJ based	SOT-MTJ based	CMOS	CMOS	Memristor based	CMOS	CMOS
Model type	FPE	s-LLGS	-	-	-	-	-
Network Size[$]	784-400/ 784-400-10	784-6c5-2s-12c5-2s-10o	784-400/ 784-400-10	-	784-6c5-2s-12c5-2s-10o	256-128-128-128-10	(256)-256-16
Voltage (V)	1	1	1.4	0.71/0.9	1	0.5-1	0.5-0.8
Bit Width of Weight	1	1	1	8	1	1	8
Auto-reset	YES	NO	NO	NO	NO	NO	YES
Frequency (MHz)	400	-	37.5	31.5/252	24	0.07	13-115
Accuracy for MNIST Dataset	75%/ 83.71%	94.6%	65.9%/ 92.3%	-	97.84%	91.8%[#]	87.3%[##]
Neuron Energy (pJ/spike)	0.42[!]	>0.08[!!]	8.4	-	0.75	-	-
Neuron and synapse Energy (pJ/SOP)	0.23	-	-	0.34 @ 0.71V	-	1.5 @ 0.5V	5.3 @ 0.5V
Normalized Energy* (pJ/SOP)	0.23	-	-	1.025	-	1.7941	27.93

[$] c, s and o in the Network Size column stand for convolutional kernel, subsampling kernel and output, respectively.

[#] The dataset is Google Speech Command Dataset (4 Keywords).

[##] The dataset is IBM DVS Gestures.

[!] The neuron energy includes the energy for the current mirror, the neuron core (2T3MTJ) and the two inverters shown in Fig. 6.6.

[!!] The energy is estimated by the currents which are written in the papers[94]-[95]. The energy of the write operation is only for the core device with the resistance state changed and does not include the energy for the external circuits.

[*] Since 28nm technology node is not shown in the paper [107], we used 32nm node for the normalization.

enough to switch the STT-MTJ from AP state to P state in one read operation. For this work, 7.5kΩ is the maximum resistance of STT-MTJ which guarantees the switch can happen in 1ns read operation. Here, I assume that the STT-MTJs in the neurons and synapses are same.

In fact, the resistance in the synapse can be larger than 7.5kΩ and leads to a better energy efficiency. The comparison among my work and state-of-the-art is shown in Table 4.2. It is unfair to compare the energy consumption in different technology nodes. In this table, I used a scaling equation to normalize the energy consumption to the 45nm technology node. [107] The scaling equations are expressed as:

$$E_x = \frac{E_{Fx}}{E_{Fy}} E_y \tag{4.4}$$

$$E_F = a_{e2} V_{dd}^2 + a_{e1} V_{dd} + a_{e0} \tag{4.5}$$

where E_x is the normalized energy, E_y is the original energy, E_F is the energy factor and V_{dd} is the supply voltage. The coefficients $a_{e0} - a_{e2}$ are shown in Table 4.3 and the values are from paper [107].

Table 4.3. Value of coefficients $a_{e0} - a_{e2}$

Type		Node	a_{e0}	a_{e1}	a_{e2}
Bulk	-	90nm	1.975	-4.398	3.755
High-k	HP	45nm	0.1539	-0.3107	1.018
		32nm	0.1701	-0.4341	0.8367
	LP	45nm	0.2767	-0.362	1.103
		32nm	0.471	-0.7823	0.9559

In comparison to other works, my design exhibits superior operational frequency and energy efficiency. There are mainly three factors in my design leading to a higher energy efficiency. The first factor contributing to the higher energy efficiency of my design is the utilization of magnetization to represent the membrane potential in the standard LIF neuron. The STT-MTJ provides moderate energy efficiency (the switching energy is about 1pJ) and two resistance states, which makes it possible to design low-power SNNs. [108] The second factor is that the characteristics of STT-MTJ switching can happened in about 1ns and there are no operational amplifiers with negative feedback in the BSNN. This leads to a high-frequency operation and a high energy efficiency. The last factor is that the auto-reset technology in the neuron can reuse the read energy and reduce the energy consumption.

It should be acknowledged that the energy per fundamental operational units and accuracy may still have some correlations, but determining and measuring these correlations can be complex and difficult. While the duration of the write operation may impact the energy consumption per SOP, it is not always directly correlated with accuracy. In this BSNN, If the duration of the write operation is adjusted by a factor of 10%, either increased or decreased, the test accuracy in the software will decrease to 80.5% or 81.6% from the current test accuracy of 82%, respectively. It proves that there exists an optimal value for the write operation duration, which corresponds to the training algorithm employed in the network. Because of lacking suitable training algorithm, it is difficult to analyze the how the energy consumption impacts the accuracy in my design. However, a comparison based on the fundamental operational unit is sufficient to demonstrate that my design exhibits a significantly higher energy efficiency compared to existing designs, thereby providing a competitive advantage.

Chapter 5

Conclusions

5.1 Review of MTJ dynamic models

There are many existing MTJ simulation models based on different switching models [53]-[55], [56], [63]-[65], [109]-[122]. This chapter presents a review of different models and compares their time efficiency. The review of switching models is shown in Table 5.1. Here, I compare the original switching model rather than the improved one. S-LLGS equation has the highest accuracy and switching probability model requires least CPU time. The FPE approach provides a more balanced solution compared to the other two models. From the information provided and application range, the s-LLGS model performs the best, while the probability model has a narrower application range.

Table 5.1. Comparison among different switching models

Type of switching model	Accuracy	Speed	Memory Requirement	Information	Applied for all currents
s-LLGS	High	Low	High	Angle	Yes
Switching probability	Low	High	Low	Switching probability	No
FPE	Medium	Medium	Medium	Switching probability	Yes

5.2 Review of MTJ circuit simulation models

The summary of different MTJ circuit simulation models is shown in Table 5.2. The comparison of the models from seven different points of view: switching model, implementation, stochastic term, temperature effect, time varying current, intermediate regime problem-solving, and information obtained. The switching model and the Implementation are the type of the dynamic model and the implementation method of the model. The Langevin term which necessitates random number generation at each timestep is indicated by the stochastic term. The temperature effect denotes the incorporation of

Table 5.2. Comparison among different MTJ circuit simulation models

Model	**Switching model**	**Implementation**	**Stochastic term**	**Temperature effect**	**Time varying current**	**Intermediate regime problem solving**	**Information obtained**
My work [54]-[55]	Switching probability	Verilog-A	No	Yes	Yes	Yes	Switching probability
My work [56]	FPE	Verilog-A	No	Yes	Yes	-	Switching probability and angle
Panagopoulos [68]	s-LLGS	Sub-circuit	No	Yes	Yes	-	Angle
Nigman [109]	s-LLGS	Sub-circuit	Yes	No	Yes	-	Angle
Panagopoulos [110]	s-LLGS	Sub-circuit	No	Yes	Yes	-	Angle
Lee [111]	s-LLGS	Verilog-A	Yes	No	Yes	-	Angle
Torunbalci [112]	s-LLGS	Sub-circuit	Yes	No	Yes	-	Angle
Garcia-Redondo [113]	Modified s-LLGS	Verilog-A	No	No	Yes	-	Switching probability and angle
	s-LLGS	Verilog-A	Yes	No	Yes	-	Angle
Kim [114]	s-LLGS	Sub-circuit	No	Yes	Yes	-	Angle
Rose [115]	s-LLGS	Verilog-A	Yes	No	Yes	-	Angle
Xu [116]-[117]	LLGS	Sub-circuit	No	No	Yes	-	Angle
Ahmed [118]	LLGS	Verilog-A	No	Yes	Yes	-	Angle
Zhang [63]	Switching probability	Verilog-A	No	No	No	No	Switching probability
Yang [119]	Switching probability	Verilog-A	No	Yes	No	Yes	Switching probability
Rose [53]	Switching probability	Verlog-A	No	Yes	Yes	No	Switching probability
Rose [120]	Switching probability	Verilog-A	No	No	Yes	No	Switching probability
Sakimura [121]	Switching probability	NS-SPICE	No	No	Yes	No	Switching probability
Zhang [65]	Switching probability	Verilog-A	No	No	No	No	Switching probability
Wang [64]	Switching probability	Verilog-A	No	Yes	No	No	Switching probability
Wang [122]	Switching probability	Verilog-A	No	No	Yes	No	Switching probability

temperature-dependent TMR ratio or Joule heating effect. The time varying current column specifies the model's capability to handle changing incident current over time.
The s-LLGS based model can provide the most accurate magnetization trajectory. However, it requires generating multiple random numbers for each axis to implement Langevin field, which leads to a higher consumption of CPU time and memory. Furthermore, it only provides angle information, which means it is necessary to perform a large number of Monte Carlo simulations for some applications such as write error rate of STT-MRAMs. There are still a few models intending to reduce computational load. Nigman [109] proposed to use the thermal noise of the resistor for modeling the thermal fluctuation of STT-MTJ. Garcia-Redondo [113] uses an effective Langevin field which is a summation of Langevin field of each time step. The effective Langevin field is calculated in the Python based on FPE. This method can avoid the random number generation and provide switching probability. The comparison of simulation times between this model and my models is shown in chapter 5.2.

For the switching probability based models, the main problem is the intermediate regime problem. Yang [119] solved the intermediate regime problem by a self-defined function to calculate the requirement of switching time. However, it uses a fitting variable k which is not constant fitting parameters, which limits it to be used for time varying current. Meanwhile, based on different types of the switching probability model, some other models can't handle the time varying current, either. One of the more special methods to handle the time varying current is that Sakimura [121] utilized a customized NS-SPICE to enable the model to be used with time varying current. Compared to other works, my switching probability model has the merit of solving the problem of lacking a physical model in the intermediate regime. Moreover, the model is able to work with time varying current.

The Fokker-Planck equation (FPE) based model presented in my work is the first to be developed for circuit simulation purposes. Other existing works have only used FPE to fit the parameters [113]-[114] or to analyze the performance of MTJ device [46], [52], [72], [123]. By using the framework proposed in this study, the FPE-based model can obtain switching probability and angle information, which makes it a general-purpose model. Compared with switching probability-based models, the FPE-based model does not suffer from intermediate regime inaccuracies. The number of parameters required to be fitted in the FPE-based model [56] is much smaller compared with the number of parameters required to be fitted in the switching probability based models [54]-[55] which use mathematical methods to solve the intermediate region problem. Furthermore, compared with the s-LLGS model, it requires less CPU time and memories.

5.3 Time efficiency comparison

STT-MTJs have found widespread application in large-scale circuits such as neural networks and STT-MRAMs. The efficiency of CPU time is a crucial factor to consider while selecting the appropriate model for such applications. In this section, we provide a comparative analysis of various models across different categories. All simulations were performed using HSPICE.

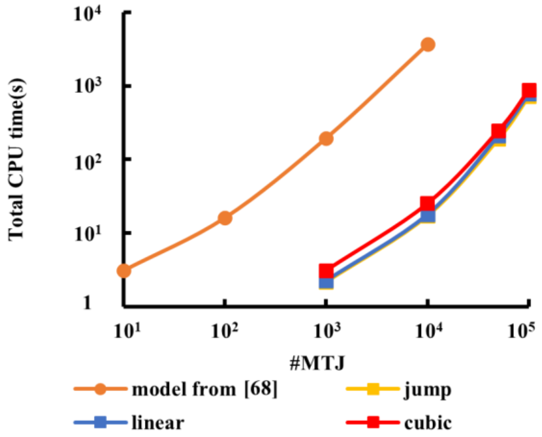


Fig. 5.1. HSPICE simulation results of model computing time comparison.

Firstly, I compared the simulation times based on the reversal of a single STT-MTJ's magnetic moment. The STT-MTJs are connected with independent voltage sources and switch the states simultaneously. For my switching probability model, I am comparing the time efficiency of two different interpolation methods: linear interpolation and cubic interpolation. Additionally, I am comparing my model with one that directly switches (jumps) between the Sun model and Néel-Brown model (which I have coded myself), as well as a s-LLGS equation based model from paper [68]. The comparison results are shown in Fig. 5.1. The total number of the STT-MTJs is changed in a large range from 10^1 to 10^5. The simulation time required for the s-LLGS model is orders of magnitude larger than that of the switching probability based model. In comparison to model that directly switching between the Sun model and Néel-Brown model, my model based on cubic spline interpolation solves the intermediate regime problem at the cost of approximately 20% higher CPU time consumption when simulating a circuit with 10^5 STT-MTJs.

A comparison which includes the FPE based model with the same circuit is shown in Fig. 5.2. The time is normalized by the CPU time of the switching probability model with 1 STT-MTJ and 1 voltage source. Each simulation was repeated 10 times and the average results were obtained for each simulation time. Compared with the s-LLGS based model, the FPE model can reduce CPU time by approximately 40% for a single MTJ structure. Among the models compared in the HSPICE simulation, the switching probability based model had the lowest CPU time consumption. However, the number of required parameters in the preprocessing step for fitting experimental data is larger than that of the FPE based model. With the assumption that there is no external field (H_{ex}=0), FPE based model only needs to fit 3 parameters (I_c, ξ, and τ_D), while my switching probability based model needs to fit 6 parameters (A, B, τ_0, I_c, ξ, and τ_D). More parameters to fit means a significantly increased amount of preparation work prior to the simulation. Additionally, the absence of theta information limits the applicability of my switching probability based model.

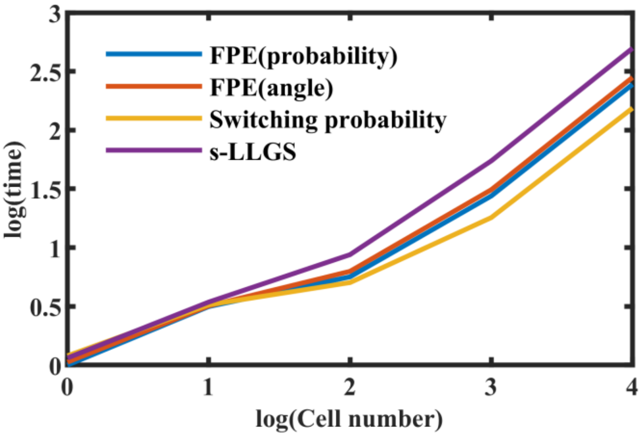


Fig. 5.2. Simulation results of the computing time comparison among the switching probability based model [55], s-LLGS based model, and FPE model [56] with a single STT-MTJ structure.

Given that various models may display different sensitivities to circuit complexity, I conducted simulations under increasingly complex circuit configurations to facilitate a more comprehensive comparison. We used a 4T2MTJ memory cell array to compare the CPU times. The structure and control signals of the 4T2MTJ memory cell array are shown in Fig. 5.3 and Fig. 5.4, respectively. The simulation results of a single 4T2MTJ cell in the FPE-based model are shown in chapter 6.1. The array is set to have 100 rows and the numbers of columns ranges from 2 to 500. It is worth to note that the STT-MRAM array is just an example to compare the CPU time required for different complexity circuits. In actuality, the primary utilization of such large-scale circuits is in neural networks. The s-LLGS based model we used in this comparison is from paper [113]. This model has two modes: constant thermal noise mode which uses the effective Langevin field and thermal mode which uses the random Langevin field. The absolute tolerance which is a simulation parameter in Verilog-A for this model is set as 10^{-6}.

The simulation includes one write operation to switch the states of STT-MTJs in 4T2MTJ memory cells and one read operation on the first column followed by checking the writing results. Meanwhile all other cells are in the state of power on in which V_{PL} is V_{dd} and V_{WL} is 0.

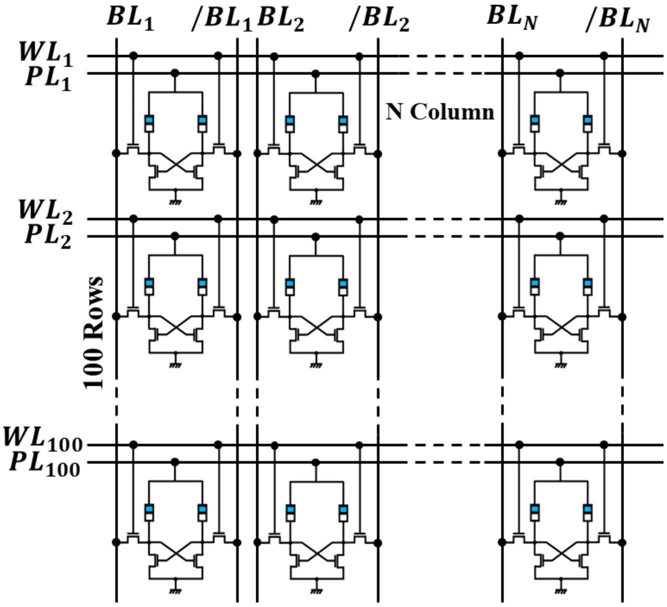


Fig. 5.3. Structure of 4T2MTJ array.

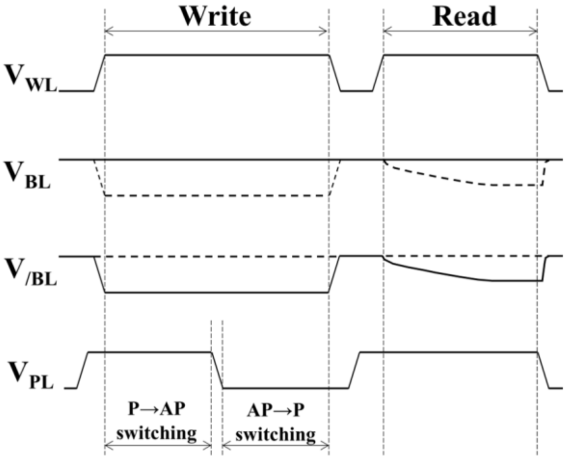


Fig. 5.4. Read/write operations of the 4T2MTJ cell.

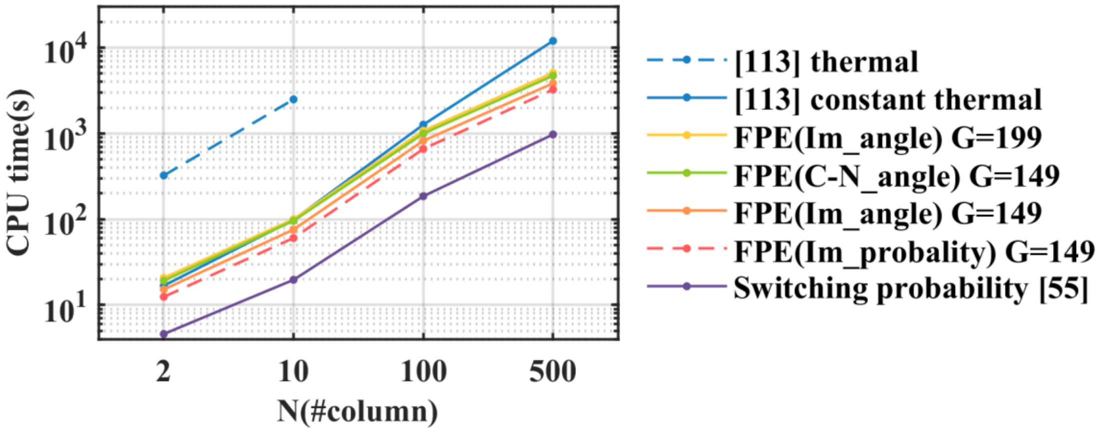


Fig. 5.5. CPU time comparison among different MTJ models. G stands for grid number. I_m means the implicit method and C-N means the Crank- Nicolson method.

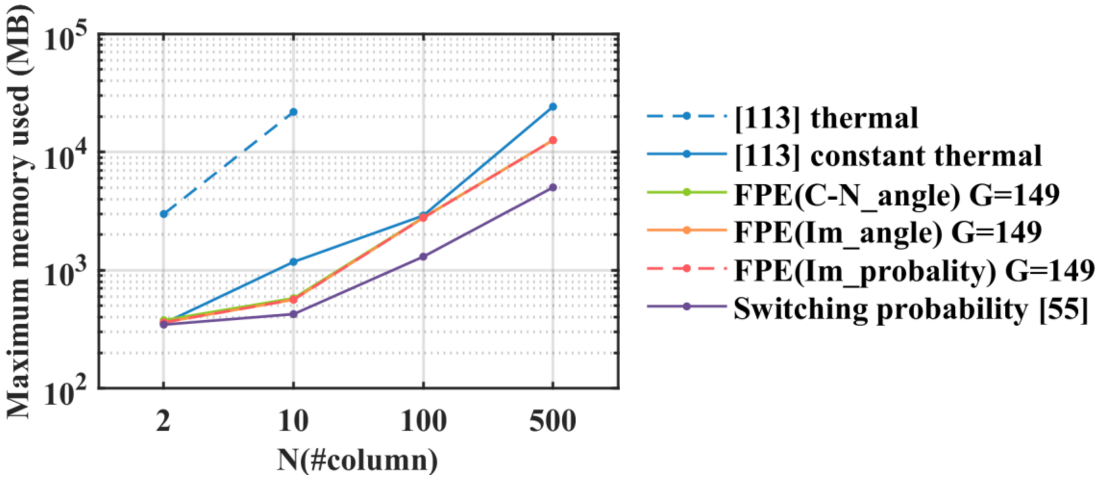


Fig. 5.6. Maximum memory consumption comparison among different MTJ models. G stands for grid number. I_m means the implicit method and C-N means the Crank- Nicolson method.

The comparison results are shown in Fig. 5.5. Here, each simulation also was repeated 10 times as the single STT-MTJ simulation and the average results were obtained for each simulation time. The impacts of underlying solver in the FPE model are also compared in Fig. 5.5. For a large-scale circuit, the s-LLGS model with a random Langevin field consumes several orders of magnitude longer CPU time. My FPE based model can reduce the total CPU time by 97% from the thermal mode of the model [113] when N=10. Moreover, the memory requirement poses a significant challenge to the utilization of this method in large-scale circuits. The switching probability model shows the same performance as the single STT-MTJ simulation which requires the least amount of CPU time. Compared with the constant thermal noise mode of the s-LLGS based model, the theta mode of the implicit method demonstrates higher time efficiency as the circuit complexity increases. When N = 500 and G (grid number) = 149, my model can reduce CPU time consumption by 68%. Despite the use of the Crank-Nicolson method in our FPE based model, it still demonstrates better time efficiency compared with the constant thermal mode of the s-LLGS based model.

Meanwhile, I have also compared the maximum memory usage of different models. The results are shown in Fig. 5.6. The maximum memory used by different modes of the proposed FPE-based model is almost the same. In contrast, the original s-LLGS model consumes approximately 38 times more memory than the proposed FPE-based model when N=10. For N=500, my model is capable of reducing the maximum memory consumption by 48%. Notably, the switching probability-based model exhibits the lowest memory consumption across all N situations.

5.4 Conclusion

In this dissertation, I presented two kinds of spin-transfer torque magnetic tunnel junction simulation models. The switching probability based model solved the intermediate regime problem and has the capability to handle time varying currents. The FPE based model proposed a new framework to obtain the information on the theta between the magnetizations in the free layers as a function of time. The effectiveness of the models had been validated by comparing with experimental data. Comparing with other models, the Fokker-Planck equation based model has higher time efficiency while providing the same kinds of information. The models are capable of meeting the requirements of various types of applications which use STT-MTJ devices. I employed STT-MRAM and the BSNN that I designed as examples to demonstrate the practicality of the proposed model.

In Chapter 1, I discussed the development history and basic characteristics of STT-MTJs, as well as their applications. I introduced the basic physical model of an MTJ’s switching dynamics, which is the Stochastic Landau-Lifshitz-Gilbert-Slonczewski model (s-LLGS). Additionally, I provided an explanation on how to calculate the effective fields in STT-MTJs.

In Chapter 2, a switching probability based STT-MTJ simulation model was proposed. The switching probability is deviated from the s-LLGS equation with some assumptions, which makes it can only work in some current condition. The main problem with probability-switching models is the lack of a physical model in the intermediate region. The problem was solved by using a framework that connects two existing models, each suitable only for regimes far greater or smaller than the critical current. Meanwhile, I also used an effective time which depends on the current other than the actual time to handle the time varying current. The validation results show that the model works in the intermediate regime with time varying current. Compared with some of the s-LLGS based model, this model consumes less CPU time with a higher accuracy.

In Chapter 3, a Fokker-Planck equation (FPE) based STT-MTJ simulation model was proposed. The FPE is also deviated from the s-LLGS equation which can be thought of as the master equation of the s-LLGS equation. I proposed a new framework to obtain the magnetization changing information from the probability density distribution. This framework makes the model usable for all kinds of applications. The FPE was solved by the finite difference method. I conducted a detailed comparison of three forms of finite difference method and selected the optimum forms which balance the accuracy and computational load. The model's validity had been confirmed by comparing it with both the s-LLGS model and experimental data.

In Chapter 4, I applied the proposed FPE based model to two kinds of applications which use different properties of STT-MTJ. One is the STT-MRAM with 4T2MTJ memory cells, which was to observe the resistance property and stochastic switching. The other was the proposed binaried spiking neural network (BSNN) which utilizes the magnetization changing to realizes LIF neuron in SNN. The 4T3MTJ structure implement the auto-reset function. Meanwhile, the 2T2MTJ synapse provides a writing current with a large signal and high-frequency operations. The BSNN shows a high potential for low-power SNN implementations. It is shown that the energy is reduced as low as 0.23pJ/SOP.

In Chapter 5, I compared my models with other existing models. The comparison results shows that the proposed switching probability model is the only one which solves the intermediate regime problem and can handle time varying current. The proposed FPE based model is the first one using the FPE to make STT-MTJ simulation model. Compared with the directly switch model, the Néel-Brown model and Sun model, the proposed switching probability based model consumes roughly the same amount of time which is much smaller than the s-LLGS based model. Compared with the s-LLGS based models, the proposed Fokker-Planck equation based model can reduce the CPU time by 97% while providing same kinds of information in large-scale circuit simulation.

Appendix

A. Switching probability based circuit simulation model code (Verilog-A)

```
/// MTJ model 2022/04/23
//Author: Haoyan LIU
//email:liuhaoyan@ruri.waseda.jp
//Switching probability based STT-MTJ circuit simulation model

`include "constants.vams"
`include "disciplines.vams"

module MTJ(out, in);
inout in, out;
electrical in;//pinned layer
electrical out;//free layer
electrical stat,resis,probability,cur;
//Define nodes
branch(in, out) mtj;

//function control switch
parameter real rswitch = 0 from (-inf:inf); //switch to control the RP calculation
parameter real Iswitch = 0 from (-inf:inf); //switch to control the Ic calculation
parameter real Tswitch = 0 from (-inf:inf); //switch to control the heat mode

//basic parameter definition
parameter real ini = 1 from (-inf:inf); //initial state
parameter real Pth = 0.63 from (-inf:inf); //predetermined switching probability
parameter real E1 = 18900 from (-inf:inf); //energy barrier
parameter real Troom = 300 from (-inf:inf); //room temperature
parameter real tau0 = 24e-12 from (-inf:inf); //inverse of attempt frequency
parameter real tau_relax = 0.26e-9 from (-inf:inf); //relaxation time
parameter real pi = 3.141592 from (-inf:inf); //pi
parameter real Rp1 = 2e3 from (0:inf); //resistance of P state
parameter real TMR01 = 2 from (-inf:inf); //TMR ratio at 0V and 0K
parameter real VH = 0.8 from (-inf:inf); // Voltage for TMRR=TMRR/2
parameter real F = 332.2 from (-inf:inf); //fitting factor
parameter real tox = 0.85e-9 from (-inf:inf); //thickness of oxide layer
parameter real dia = 65e-9 from (-inf:inf); //diameter of MTJ
parameter real phiB = 0.4 from (-inf:inf); //energy barrier height of the oxide layer

//critical current calculate parameter definition
parameter real alpha = 0.027 from (-inf:inf);//damping factor
parameter real el = 1.69e-19 from (-inf:inf);//electron charge
parameter real muB = 9.27e-24 from (-inf:inf);//Bohr Magneton
parameter real mu0 = 1.26e-6 from (-inf:inf);//Vacuum permeability
parameter real hbar = 1.05e-34 from (-inf:inf); //reduced Planck constant
parameter real Hk = 1.13e5 from (-inf:inf); //Anisotropy field unit:T
parameter real Ms = 4.56e5 from (-inf:inf); //Saturation magnetization:A/m
parameter real polar = 0.56 from (-inf:inf); //polarization factor
parameter real tfl = 1.3e-9 from (-inf:inf);//thickness of free layer

//Temperature parameter definition
parameter real thicks = 10e-9 from (-inf:inf); //thickness of MTJ
parameter real Cv = 3.5e6 from (-inf:inf); //the heat capacity per unit volume
parameter real lambda = 1.5 from (-inf:inf); //the thermal conductivity
parameter real k = 0.002 from (-inf:inf); //TMRR slope with temperature

//critical current definition
parameter real Icp1 = 60e-6 from (-inf:inf); //critical current for P to AP switch
parameter real Icap1 = -50e-6 from (-inf:inf); //critical current for AP to P switch

//Parameter of alpha function
parameter real pa = 4.2 from (-inf:inf); //fitting factor
parameter real pb = 0.9 from (-inf:inf); //fitting factor
parameter real pc = 1.01 from (-inf:inf); //fitting factor

//variable define
integer sta;
genvar i;
real Pp,Pap,R;
real t,t8,t9;
real curre;
real k1,P,kk;
real time,timelast,timeperiod;
real Rp,Icp,Icap;
real gp,gap,Energy,gamma,Vol,polar1;
real E,TMR0,T,tauth,Tcool,Theat,Tmax,ticool,tiheat,tiheat1;
real
h[1:21],b[1:21],q[1:20],u[1:20],c[1:20],d[1:20],z[1:22],fun[1:22],x[1:22],z[1:22],part[1:4];
real
cubicparlarge[1:22]={-100,-10,-5,-1,-0.5,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.2,1.5,1.7,2,3,5,1
0,500};
real
cubicparsmall[1:22]={-10,-7,-5,-1,-0.5,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.2,1.5,1.7,2,3,5,10,
500};
real debug1;

real tini,tinipre,tiniabs,count,tini_1;
real t1[1:22],t1ex[1:22],Isum[1:22];
integer position;


analog begin
//initial step
   @(initial_step) begin
// Parameter initialization
```

```
        debug1 = 0;
        Pp = 0;
        Pap = 0;
        time = 0;
        timelast = 0;
        timeperiod = 0;
        curre = 0;
        z[1]=0;
        z[22]=0;
        t = 0;
        T = Troom;
        E = E1/Troom;
        P = -ln(1-Pth);
        k1 = -tau_relax * ln(ln(Pth)/-4/E)/2;
        kk=-tau_relax * ln(ln(Pth)/-4/E)/2;
        TMR0 = TMR01-k*T;
        sta = ini;
//RP calculation
        if (rswitch == 0)
                Rp = Rp1;
        else
                Rp = tox/F/(dia*dia*pi/4)/sqrt(phiB)*exp(1.025*tox*sqrt(phiB));
//Ic calculation
        if (Iswitch == 0)
        begin
                Icp = Icp1;
                Icap = Icap1;
        end
        else
        begin
                Vol = tfl*dia*dia*pi/4;
                gamma = 2*muB/hbar;
                Energy = mu0*Ms*Vol*Hk/2;
                polar1 = sqrt(1/polar)+sqrt(polar);
                gp = 1/(-4+polar1*polar1*polar1);
                gap = 1/(-4+polar1*polar1*polar1*0.5);
                Icp = 2*alpha*gamma*el*Energy/muB/gp;
                Icap = 2*alpha*gamma*el*Energy/muB/gap;
        end
//Joule effect calculation
        if (Tswitch == 1)
        begin
                tauth = Cv*thicks/lambda*tox;
                Theat = Troom;
                Tcool = Troom;
                ticool = 0;
                tiheat = 0;
        end
    end
//main part
```

```
    V(vgnd) <+ 0.0;
    time = $abstime;
    timeperiod = time-timelast;
    timelast = time;
// Resistance calculation
    if(sta == 0)
        R = Rp;
    else
        R = Rp*(1+abs(TMR0)/(1+V(mtj)*V(mtj)/VH/VH));
// current calculation
    curre = V(mtj)/R;
//Joule effect calculation
    if (Tswitch == 1)
    begin
            if (curre==0)
            begin
                    ticool=-ln((Tcool-Troom)/(Theat-Troom))*tauth;
                    Tcool=Troom+(Theat-Troom)*exp(-(ticool+timeperiod)/tauth);
                    T=Tcool;
            end
            else
            begin
                    Tmax=V(mtj)*curre/(dia*dia*pi/4)*thicks/lambda;
                    tiheat=-ln(1-(Theat-Troom)/Tmax)*tauth;
                    Theat=Troom+Tmax*(1-exp(-(tiheat+timeperiod)/tauth));
                    T=Theat;
            end
            TMR0 = TMR01-k*T;
    end
    if(sta == 0)
        R = Rp;
    else
        R = Rp*(1+abs(TMR0)/(1+V(mtj)*V(mtj)/VH/VH));
    curre = V(mtj)/R;
//switching time calculate
//p state
    count=0;
// Framework to obtain switching time
    if (curre > 0)
    begin
            debug1 = 0;
            if (curre >= pc*Icp)
            begin
                    tinipre=k1/(curre/Icp-1);
            end
            if ((curre >= 0.85*Icp)&&(curre < pc*Icp))
            begin
                    tinipre=P*tau0/exp(-E*(1-0.85));
            end
            if (curre < 0.85*Icp)
            begin
                    tinipre=P*tau0/exp(-E*(1-curre/Icp));
            end

            tini=ln(tinipre*1e9);
            if (tini > 2)
                    //if (curre > pc*Icp)
                            tini = 2;
            tiniabs=abs(tini);
            position=22;
        //data points
            if (curre > 1.5*Icp)
            begin
                    for(i=1;i<=22;i=i+1)
                    begin
                            t1[i]=tini + cubicparlarge[i] * tiniabs;
                            t1ex[i]=exp(t1[i])*1e-9;
                            Isum[i]=Icp*(k1/t1ex[i]+1)   *(1-   1/(1+exp(-(t1[i]-pa)*pb)))   +
Icp*(1+ln(P*tau0/t1ex[i])/E) * 1/(1+exp(-(t1[i]-pa)*pb));
                            if ((curre>=Isum[i])&&(count==0))
                            begin
                                    position=i;
                                    count=1;
                            end
                    end
            end
            else
            begin
                    for(i=1;i<=22;i=i+1)
                    begin
                            t1[i]=tini_1 + cubicparsmall[i] * tiniabs; //t2
                            t1ex[i]=exp(t1[i])*1e-9; //t3
                            Isum[i]=Icp*(k1/t1ex[i]+1)    *(1-1/(1+exp(-(t1[i]-pa)*pb)))    +
Icp*(1+ln(P*tau0/t1ex[i])/E) * 1/(1+exp(-(t1[i]-pa)*pb));
                            if ((curre>=Isum[i])&&(count==0))
                            begin
                                    position=i;
                                    count=1;
                            end
                    end
            end
            if (position>=2)
            begin
                    position=position-1;
            end
// interpolation way
// parameter prepare
            for (i=1;i<=21;i=i+1)
            begin
                    h[i]=Isum[i+1]-Isum[i];
                    b[i]=(t1[i+1]-t1[i])/h[i];
            end
            for (i=1;i<=20;i=i+1)
            begin
                    q[i]=2*(h[i]+h[i+1]);
                    u[i]=6*(b[i+1]-b[i]);
            end
            for (i=1;i<=20;i=i+1)
            begin
                    if (i==1)
                    begin
                            c[i]=h[i]/q[i];
                            d[i]=u[i]/q[i];
                    end
                    else
                    begin
                            c[i]=h[i]/(q[i]-c[i-1]*h[i]);
                            d[i]=(u[i]-d[i-1]*h[i])/(q[i]-c[i-1]*h[i]);
                    end
            end
            for (i=21;i>=2;i=i-1)
            begin
                    z[i]=d[i-1]-c[i-1]*z[i+1];
            end
            part[1]=z[position+1]/6/h[position]*(curre-Isum[position])*(curre-Isum[position])*(
curre-Isum[position]);
            part[2]=z[position]/6/h[position]*(Isum[position+1]-curre)*(Isum[position+1]-curre)
*(Isum[position+1]-curre);
            part[3]=(t1[position+1]/h[position]-z[position+1]/6*h[position])*(curre-Isum[positio
n]);
            part[4]=(t1[position]/h[position]-h[position]/6*z[position])*(Isum[position+1]-curre
);
            t = part[1]+part[2]+part[3]+part[4];
      end
//ap state
      if (curre <= 0)
      begin
            debug1 = 0;
            if (curre <= pc*Icap)
            begin
                    tinipre=k1/(curre/Icap-1);
            end
            if ((curre <= 0.85*Icap)&&(curre > pc*Icap))
            begin
                    tinipre=P*tau0/exp(-E*(1-0.85));
            end
            if (curre > 0.85*Icap)
            begin
                    tinipre=P*tau0/exp(-E*(1-curre/Icap));
            end
```

```
    tini=ln(tinipre*1e9);
    if (tini > 2)
        //if (curre > pc*Icap)
            tini = 2;
    tiniabs=abs(tini);
    position=22;
  //data points
    if (curre > 1.5*Icap)
    begin
        for(i=1;i<=22;i=i+1)
        begin
            t1[i]=tini + cubicparlarge[i] * tiniabs;
            t1ex[i]=exp(t1[i])*1e-9;
            Isum[i]=Icap*(k1/t1ex[i]+1) *(1- 1/(1+exp(-(t1[i]-pa)*pb))) + Icap*(1+ln(P*tau0/t1ex[i])/E) * 1/(1+exp(-(t1[i]-pa)*pb));
            if ((curre<=Isum[i])&&(count==0))
            begin
                position=i;
                count=1;
            end
        end
    end
    else
    begin
        for(i=1;i<=22;i=i+1)
        begin
            t1[i]=tini_1 + cubicparsmall[i] * tiniabs; //t2
            t1ex[i]=exp(t1[i])*1e-9; //t3
            Isum[i]=Icap*(k1/t1ex[i]+1) *(1- 1/(1+exp(-(t1[i]-pa)*pb))) + Icap*(1+ln(P*tau0/t1ex[i])/E) * 1/(1+exp(-(t1[i]-pa)*pb));
            if ((curre<=Isum[i])&&(count==0))
            begin
                position=i;
                count=1;
            end
        end
    end
    if (position>=2)
    begin
        position=position-1;
    end


// interpolation way
// parameter prepare
    for (i=1;i<=21;i=i+1)
    begin
        h[i]=Isum[i+1]-Isum[i];
        b[i]=(t1[i+1]-t1[i])/h[i];
        end
        for (i=1;i<=20;i=i+1)
        begin
                q[i]=2*(h[i]+h[i+1]);
                u[i]=6*(b[i+1]-b[i]);
        end
        for (i=1;i<=20;i=i+1)
        begin
                if (i==1)
                begin
                        c[i]=h[i]/q[i];
                        d[i]=u[i]/q[i];
                end
                else
                begin
                        c[i]=h[i]/(q[i]-c[i-1]*h[i]);
                        d[i]=(u[i]-d[i-1]*h[i])/(q[i]-c[i-1]*h[i]);
                end
        end
        for (i=21;i>=2;i=i-1)
        begin
                z[i]=d[i-1]-c[i-1]*z[i+1];
        end
        part[1]=z[position+1]/6/h[position]*(curre-Isum[position])*(curre-Isum[position])*(
curre-Isum[position]);
        part[2]=z[position]/6/h[position]*(Isum[position+1]-curre)*(Isum[position+1]-curre)
*(Isum[position+1]-curre);
        part[3]=(t1[position+1]/h[position]-z[position+1]/6*h[position])*(curre-Isum[positio
n]);
        part[4]=(t1[position]/h[position]-h[position]/6*z[position])*(Isum[position+1]-curre
);
        t = part[1]+part[2]+part[3]+part[4];
  end

//Switching judgement
  t9=t;
  t=exp(t)*1e-9;
  if (sta == 0)
  begin
    if (curre > 0)
        Pp = Pp+1/t*timeperiod;
    else
        Pp = Pp;
  end
  else if (sta == 1)
  begin
    if (curre <= 0)
        Pap = Pap+1/t*timeperiod;
    else
        Pap = Pap;
    end
    if (sta == 0)
    begin
        if (Pp >= 1)
        begin
                sta = 1;
                Pp = 0;
        end
        else
        begin
                sta = 0;
                R = Rp;
        end
    end
    else if (sta == 1)
    begin
        if (Pap >= 1)
        begin
                sta = 0;
                Pap = 0;
        end
        else
        begin
                sta = 1;
                R = Rp*(1+abs(TMR0)/(1+V(mtj)*V(mtj)/VH/VH));
        end
    end
//Output
    I(mtj) <+ V(mtj)/R;

end
endmodule
```

B. Fokker-Planck equation based circuit simulation model code (Verilog-A)

```
// MTJ model 2022/01/18
//Author: Haoyan LIU
//email:liuhaoyan@ruri.waseda.jp
//Fokker-Planck equation based STT-MTJ circuit simulation model

`include "constants.vams"
`include "disciplines.vams"

module MTJ(out, in);
inout in, out;
electrical in;//pinned layer
electrical out;//free layer
//Define nodes
branch(in, out) mtj;

//basic parameter define
parameter real partten = 1 from (-inf:inf); //mode selector: 1 switching probability mode; 2
angle mode.
parameter real ini = 0 from (-inf:inf); //initial state selector: 0 Parallel state; 1 Anti-parallel
state
parameter real Ep = 100 from (-inf:inf); //Energy barrier for P to AP switch
parameter real Eap = 36 from (-inf:inf); //Energy barrier for AP to P switch
parameter real Hk = 0.245 from (-inf:inf); //Anisotropy field unit:T
parameter real Happ = 0 from (-inf:inf); //External feild or Applied field unit:T
parameter real pi = 3.1416 from (-inf:inf); //pi
parameter real Troom = 300 from (-inf:inf); //room temperature
parameter real Icp = 80e-6 from (-inf:inf); //critical current for P to AP switch
parameter real Icap = 30e-6 from (-inf:inf); //critical current for AP to P switch
parameter real VH = 0.8 from (-inf:inf); //Parameter for TMR ratio
parameter real TMR0 = 2.5 from (-inf:inf); //TMR ratio for 0V and 0K
parameter real Rp = 5e3 from (-inf:inf); //Resistance for Parallel state
parameter real rnum = 0.8 from (-inf:inf); //The random number to control the probability
//parameter real bstep = 10p from (-inf:inf); //Control the minimum timestep

//variable define
real control;
genvar i;
real time,timelast,timeperiod,curre,Rmtj,P,state,Rap,Gp,Gap,Pth;
real P_total,tauD,dtheta,par1,par2p,par2ap,positionpre,control,stable;
integer position;
real P_stable;
real Pd [1:199],Ic[1:199],Pdini[1:199];
real Pdlast [1:199];
real theta [1:199];
real f0[1:199];
real f[1:199];
real a[1:199],b[1:199],c[1:198],d[1:199],pu[1:198],y[1:199];

analog begin
// initial step
   @(initial_step) begin
// Parameter initialization
            P_stable = 0;
            time = 0;
            timelast = 0;
            timeperiod = 0;
            state=ini;
            Pth = rnum;
            Rmtj = 0;
            Rap = Rp*(1+TMR0);
            Gp=1/Rp;
```

```
    Gap=1/Rap;
    tauD = 0.15e-9;
    dtheta = pi / 100.0 / 2.0;
    P_total = 0;
    P = 0;
// Initial angle distribution calculation
    if (state<0.5)
    begin
        for(i=1;i<=100;i=i+1)
        begin
            theta[i] = i * pi / 100.0 / 2.0;
            P_total = P_total + 2 * Ep *sin(theta[i]) * exp(-Ep * sin(theta[i]) *
sin(theta[i])) * pi / 100 / 2;
        end
        for(i=1;i<=100;i=i+1)
        begin
            Pdlast[i] =  2 * Ep *sin(theta[i]) * exp(-Ep * sin(theta[i]) *
sin(theta[i])) / P_total;
            f0[i] =
-cos(theta[i])*sin(theta[i])-Happ/Hk*cos(theta[i])+cos(theta[i])/sin(theta[i])/2/Ep;
        end
        for(i=101;i<=199;i=i+1)
        begin
            theta[i] = i * pi / 100.0 / 2.0;
            Pdlast[i] =   0;
            f0[i] =
-cos(theta[i])*sin(theta[i])-Happ/Hk*cos(theta[i])+cos(theta[i])/sin(theta[i])/2/Ep;
        end
        position=1;
    end
    else
    begin
        for(i=100;i<=199;i=i+1)
        begin
            theta[i] = i * pi / 100.0 / 2.0;
            P_total = P_total + 2 * Eap *sin(theta[i]) * exp(-Eap * sin(theta[i])
* sin(theta[i])) * pi / 100 / 2;
        end
        for(i=1;i<=99;i=i+1)
        begin
            theta[i] = i * pi / 100.0 / 2.0;
            Pdlast[i] = 0;
            f0[i] =
-cos(theta[i])*sin(theta[i])-Happ/Hk*cos(theta[i])+cos(theta[i])/sin(theta[i])/2/Eap;
        end
        for(i=100;i<=199;i=i+1)
        begin
            Pdlast[i] =  2 * Eap *sin(theta[i]) * exp(-Eap * sin(theta[i]) *
sin(theta[i])) / P_total;
            f0[i] =
-cos(theta[i])*sin(theta[i])-Happ/Hk*cos(theta[i])+cos(theta[i])/sin(theta[i])/2/Eap;
                    end
                    position=199;
          end
          if (state==0)
          begin
                    for(i=1;i<=199;i=i+1)
                    begin
                              Pdini[i] = Pdlast[i];
                    end
          end
          else
          begin
                    for(i=1;i<=199;i=i+1)
                    begin
                              Pdini[i] = Pdlast[200-i];
                    end
          end
   end
// main part
   //$bound_step(bstep);
   //timestep calculation
   time = $realtime;
   timeperiod = time-timelast;
   timelast = time;
// Resistance calculation
   if (partten==0)
   begin
          if(state==0)
          begin
                    Rmtj = Rp;
          end
          else
          begin
                    Rmtj = Rp*(1+(TMR0/(1+V(mtj)*V(mtj)/VH/VH)));
          end
   end
   else
   begin
          Rmtj=1/(1/2.0*(Gp+Gap)+1/2.0*(Gp-Gap)*cos(theta[position]));
   end
// Current calculation
   curre = V(mtj) / Rmtj;
// Tomas method
   par1 = timeperiod / 2 / tauD / dtheta;
   par2p = timeperiod / 2 / tauD / Ep / dtheta / dtheta;
   par2ap = timeperiod / 2 / tauD / Eap / dtheta / dtheta;
   P = 0;
   for(i=1;i<=199;i=i+1)
   begin
```

```
          if (i<=100)
          begin
                    f[i] = curre/Icp*sin(theta[i]) + f0[i];
          end
          else
          begin
                    f[i] = curre/Icap*sin(theta[i]) + f0[i];
          end
          if(i==1)
          begin
                    a[1]=0;
                    b[1]=1+par2p+par1*f[1];
                    d[1]=b[1];
                    y[1]=Pdlast[1]/d[1];
          end
          else if(i==199)
          begin
                    a[i]=-par1*f[i-1]-par2ap;
                    b[i]=1+par2ap-par1*f[i];
          end
          else
          begin
                    if(i>100)
                    begin
                              a[i]=-par1*f[i-1]-par2ap;
                              b[i]=1+2*par2ap;
                    end
                    else
                    begin
                              a[i]=-par1*f[i-1]-par2p;
                              b[i]=1+2*par2p;
                    end

          end
end
for(i=1;i<=198;i=i+1)
begin
          if(i>100)
          begin
                    c[i]=par1*f[i+1]-par2ap;
          end
          else
          begin
                    c[i]=par1*f[i+1]-par2p;
          end
          pu[i]=c[i]/d[i];
          d[i+1]=b[i+1]-a[i+1]*pu[i];
          if(i>1)
          begin
                    y[i]=(Pdlast[i]-a[i]*y[i-1])/d[i];
```

```
       end
  end
  y[199]=(Pdlast[199]-a[199]*y[198])/d[199];
  Pd[199]=y[199];

//switch control for different modes of the model
  if (partten==1) //theta mode
  begin
        if ((position<100)&&(curre>0))
        begin
              P_stable=0;
        end
        if ((position>=100)&&(curre<0))
        begin
              P_stable=0;
        end
        positionpre = 0;
        positionpre = positionpre + Pd[199]*dtheta;
        control = 0;
       //find the theta position
       if((Pd[199]*dtheta>Pth)&&(control==0))
       begin
              position = 199;
              control=1;
       end
        for(i=198;i>=1;i=i-1)
        begin
              Pd[i]=y[i]-pu[i]*Pd[i+1];
              if (control == 0)
              begin
                     positionpre = positionpre + Pd[i]*dtheta;
                     if(positionpre>Pth)
                     begin
                            position = i;
                            control=1;
                     end
              end
       end
       if ((position<100)&&(curre<=0)&&(P_stable==0))
       begin
              stable=0;
              for(i=1;i<=99;i=i+1)
              begin
                     stable=stable+abs(Pd[i]-Pdlast[i]);
              end
              if(stable>0)
              begin
                     stable=timeperiod/stable;
              end
              if(stable>=10e-9)
        begin
            for(i=1;i<=199;i=i+1)
            begin
                Pdlast[i] = Pdini[i];
            end
            P_stable=1;
        end
    end
    if ((position>=100)&&(curre>=0)&&(P_stable==0))
    begin
        stable=0;
        for(i=100;i<=199;i=i+1)
        begin
            stable=stable+abs(Pd[i]-Pdlast[i]);
        end
        if(stable>0)
        begin
            stable=timeperiod/stable;
        end
        if(stable>=10e-9)
        begin
            for(i=1;i<=199;i=i+1)
            begin
                Pdlast[i] = Pdini[200-i];
            end
            P_stable=1;
        end
    end
    if(P_stable==0)
    begin
        for(i=199;i>=1;i=i-1)
        begin
            Pdlast[i] = Pd[i];
        end
    end
end
else //switching probability mode
begin
    for(i=198;i>=1;i=i-1)
    begin
        Pd[i]=y[i]-pu[i]*Pd[i+1];
        if(state<0.5)
        begin
            if(i<=100)
            begin
                P = P + Pd[i]*dtheta;
            end
        end
        else
        begin
```

```
              if (i==198)
              begin
                  P = P + Pd[199]*dtheta;
              end
              if(i>=100)
              begin
                  P = P + Pd[i]*dtheta;
              end
          end
          Pdlast[i] = Pd[i];
      end
      Pdlast[199] = Pd[199];
      P=1-P;
      if(state==0)
      begin
          if(P>rnum)
          begin
              state=1;
          end
      end
      else
      begin
          if(P>1-rnum)
          begin
              state=0;
          end
      end
  end
//Output
  I(mtj) <+ curre;
end
endmodule
```

C. Simulation example (HSPICE netlist)

```
// 1T1C1MTJ structure simulation example
//Author: Haoyan LIU
//email:liuhaoyan@ruri.waseda.jp

.hdl "./probability.va" //using switching probability model
//.hdl "./FPE.va" //using FPE model
.include "./hspice/bu40n1.mdl"
.lib "./hspice/bu40n1.skw" NT
.lib "./hspice/bu40n1.skw" PT
.include "./hspice/cmos45.pm"
.option RUNLVL = 6;
.probe V(*)
.option psf=2
```

```
.option plot=1
.save
.param Vdd=1

Vbl bl 0 PWL 0 0 0.9ns 0 1ns Vdd 12ns Vdd 12.1ns 0
Vsl sl 0 PWL 0 0
Vwl wl 0 PWL 0 0 1.5ns 0 1.6ns Vdd 11.5ns Vdd 11.6ns 0
M1 in g co 0 nmos45 L=50n W=500n
C1 co 0 1p
X1 sl co MTJ ini=0 //Switching probability based model, 'ini' controls initial state of MTJ: ini=0: P state; ini=1: AP state
//X1 0 co MTJ ini=0 pattern=1 //FPE based model, 'pattern' controls the mode. Pattern=1: theta mode; Pattern=0: probability mode
.tran 1p 30ns START=1e-14 uic
.end
```

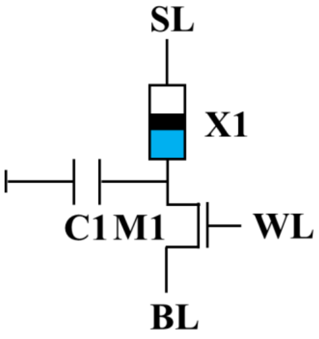


Fig. C1. 1T1C1MTJ structure.

Bibliography

[1] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. D. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature Materials*, vol. 9, no. 9, pp. 721-724, Sep. 2010.

[2] M. Julliere, "Tunneling between ferromagnetic films," *Physics Letters A*, vol. 54, no. 3 pp. 225-226, Sep. 1975.

[3] H. Itoh, and J. Inoue, "Theory of Tunnel Magnetoresistance," *Journal of the Magnetics Society of Japan*, vol. 30, no. 1, pp.1-37, 2006.

[4] D. Wang, C. Nordman, J. M. Daughton, Z. Qian, and J. Fink, "70% TMR at room temperature for SDT sandwich junctions with CoFeB as free and reference Layers," *IEEE Transactions on Magnetics*, vol. 40, no. 4, pp. 2269-2271, July 2004.

[5] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, and K. Ando, "Giant room-temperature magnetoresistance in single-crystal Fe/MgO/Fe magnetic tunnel junctions," *Nature Materials*, vol. 3, no. 12, pp. 868-871, Oct. 2004.

[6] B. N. Engel, J. Akerman, B, Butcher, R. W. Dave, M. DeHerrera, M. Durlam, G. Grynkewich, J. Janesky, S. V. Pietambaram, N. D. Rizzo, J. M. Slaughter, K. Smith, J. J. Sun, and S. Tehrani, "A 4-Mb toggle MRAM based on a novel bit and switching method," *IEEE Transactions on Magnetics*, vol. 41, no.1, pp. 132-136, Jan. 2005.

[7] T. W. Andre, J. J. Nahas, C. K. Subramanian, B. J. Garni, H. S. Lin, A. Omair, and W. L. Martino, "A 4-Mb 0.18-μm 1T1MTJ toggle MRAM with balanced three input sensing scheme and locally mirrored unidirectional write drivers," *IEEE Journal of Solid-State Circuits*, vol. 40, no.1, pp. 301-309, Jan. 2005.

[8] D. Apalkov, B. Dieny, and J. M. Slaughter, "Magnetoresistive Random Access Memory," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1796-1830, Oct. 2016.

[9] L. Berger, "Emission of spin waves by a magnetic multilayer traversed by a current," *Physical Review B*, vol. 54, no. 13, pp. 9353-9358, Oct. 1996.

[10] J. C. Slonczewski, "Current-driven excitation of magnetic multilayers," *Journal of Magnetism and Magnetic Materials*, vol. 159, no. 1-2, pp. L1-L7, June 1996.

[11] J. J. Sun, M. DeHerrera, B. Hughes, S. Ikegawa, H, K, Lee, F. B. Mancoff, K. Nagel, G. Shimon, S. M. Alam, D. Houssameddine, and S. Aggarwal, “Commercialization of 1Gb Standalone Spin-Transfer Torque MRAM”, *IEEE International Memory Workshop (IMW)*, pp. 1-4, Dresden, Germany, May 2021.

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

[13] M. Endo, S. Kanai, S. Ikeda, F. Matsukura, and H. Ohno, “Electric-field effects on thickness dependent magnetic anisotropy of sputtered MgO/$Co_{40}Fe_{40}B_{20}$/Ta structures,” *Applied Physics Letters*, vol. 96, no. 21, May 2010.

[14] J. Zhu, J. A. Katine, G. E. Rowlands, T. Chen, Z. Duan, J. G. Alzate, and P. Upadhyaya “Voltage-Induced Ferromagnetic Resonance in Magnetic Tunnel Junctions,” *Physical Review Letters*, vol. 108, no.19, p. 197203, May 2012.

[15] K. L. Wang, X. Kou, P. Upadhyaya, F. Fan, Q. Shao, G. Yu, and P. K. Amiri, “Electric-Field Control of Spin-Orbit Interaction for Low-Power Spintronics,” *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1974- 2008, Oct. 2016.

[16] S. Lee, and K. Lee, “Emerging Three-Terminal Magnetic Memory Devices,” *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1831-1843, Sep. 2016.

[17] M. DC, R. Grassl, J. Chen, M. Jamall, D. R. Hickey, D. Zhang, Z. Zhao, H, Li, P. Quarterman, Y. Lv, M. Li, A. Manchon, K. A. Mkhoyan, T. Low, and J. Wang, “Room-temperature high spin–orbit torque due to quantum confinement in sputtered $Bi_xSe_{(1–x)}$ films,” *Nature Materials*, vol. 17, no. 9, pp. 1831-1843, Oct. 2016.

[18] L. Liu, C. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman “Spin-Torque Switching with the Giant Spin Hall Effect of Tantalum,” *Science*, vol. 336, no. 6081, pp. 555-558, May 2012.

[19] J. L. García-Palacios, and F. J. Lázaro, “Langevin-dynamics study of the dynamical properties of small magnetic particles,” *Physical Review B*, vol. 58, no. 22, p. 14937, Dec. 1998.

[20] D. Almasi, H. Homayoun, H. Salmani, and H. Mahmoodi, "Comparative analysis of hybrid Magnetic Tunnel Junction and CMOS logic circuits," *IEEE International System-on-Chip Conference (SOCC)*, pp. 259-264, Seattle, WA, USA, Sep. 2016.

[21] S. Ikeda, J. Hayakawa, Y. Ashizawa, Y. M. Lee, K. Miura, H. Hasegawa, M. Tsunoda, F. Matsukura, and H. Ohno, “Tunnel magnetoresistance of 604% at 300K by suppression of Ta diffusion in CoFeB/MgO/CoFeB pseudo-spin-valves annealed at high temperature,” *Applied Physics Letters*, vol. 93, no. 8, pp. 082508, Aug. 2008.

[22] K. Mukaiyama, J. W. Jung, H. Sepehri-Amin, S. Kasai, T. Furubayashi, T. Ohkubo, and K. Hono, “Over 100% magnetoresistance ratio at room temperature in magnetic tunnel junctions with $CuGaSe_2$ spacer layer,” *Applied Physics Letters*, vol. 114, no. 17, p. 172402, Apr. 2019.

[23] H. Honjo, S. Ikeda, H. Sato, M. Yasuhira, and T. Endoh, “Effect of surface modification treatment of buffer layer on thermal tolerance of synthetic ferrimagnetic reference layer in perpendicular-anisotropy magnetic tunnel junctions,” *Journal of Applied Physics*, vol. 126, no. 11, p. 113902, Sep. 2019.

[24] D. Edelsterin, M. Rizzolo, D. Sil, A. Dutta, J. DeBrosse, M. Wordeman, A. Arceo, I. C. Chu, J. Demarest, E. R. J. Edwards, E. R. Evarts, A. Gasasira, G. Hu, M. Iwatake, R. Johnson, V. Katragadda, T. Levin, J. Li, Y. Liu, C. Long, T. Maffitt, S. McDermott, S. Mehta, V. Mehta, D. Metzler, J. Morillo, Y. Nakamura, S. Nguyen, P. Nieves, V. Oaim R. Patlolla, R. Pujari, R. Southwick, T. Standaert, O. van der Straten, H. Wu, C. Yang, D. Houssameddine, J. M. Slaughter, and D. C. Worledge, “A 14 nm Embedded STT-MRAM CMOS Technology,” *International Electron Devices Meeting (IEDM)*, p.11.5, San Francisco, CA, USA, 2020.

[25] G. Hu, C. Safranski, J. Z. Sun, P. Hashemi, S. L. Brown, J. Bruley, L. Buzi, C. P. D’ Emic, E. Galligan, M. G. Gottwald, O. Gunawan, J. Lee, S. Karimeddiny, P. L. Trouilloud, and D. C. Worledge, “Double spin-torque magnetic tunnel junction devices for last-level cache applications,” *International Electron Devices Meeting (IEDM)*, p. 10.2, San Francisco, CA, USA, 2022.

[26] H. Honjo, K. Nishioka, S. Miura, H. Naganuma, T. Watanabe, T. Nasuno, T. Tanigawa, Y. Noguchi, H. Inoue, M. Yasuhira, S. Ikeda, and T. Endoh, “25 nm iPMA-type Hexa-MTJ with solder reflow capability and endurance>10^7 for eFlash-type MRAM,” *International Electron Devices Meeting (IEDM)*, p. 10.3, San Francisco, CA, USA, 2022.

[27] A. Gerbegiorgis, H. A. D. Nguyen, J. Yu, R. Bishnoi, M. Taouil, F. Catthoor, and S. Hamdioui, “A Survey on Memory-centric Computer Architectures,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no.4, pp.1-50, Oct. 2022.

[28] J. Boukhobza, S. Rubini, R. Chen, and Z. Shao, “Emerging NVM: A Survey on Architectural Integration and Research Challenges,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 23, no.2, pp. 1-32, Mar. 2018.

[29] S. Bertolazzi, P. Bondavalli, S. Roche, T. San, S. Choi, L. Colombo, F. Bonaccorso, and P. Samori, “Nonvolatile Memories Based on Graphene and Related 2D Materials,” *Advanced Materials*, vol. 31, no. 10, p. 1806663, Mar. 2019.

[30] K. Lee, J. Kim, C. Kim, D. Ham, J. Lee, J. Lee, H. Jeong, and H. Yoo, “A 1.02-μW STT-MRAM-Based DNN ECG Arrhythmia Monitoring SoC With Leakage-Based Delay MAC Unit,” *IEEE Solid-State Circuits Letters*, vol. 3, pp. 390-393, Sep. 2020.

[31] H. Zhang, W. Kang, K. Cao, B. Wu, Y. Zhang, and W. Zhao, “Spintronic Processing Unit in Spin Transfer Torque Magnetic Random Access Memory,” *IEEE Transactions on Electron Devices*, vol. 66, no. 4, pp. 2017-2022, Apr. 2019.

[32] G. Srinivasan, A. Sengupta, and K. Roy, “Magnetic tunnel junction enabled all-spin stochastic spiking neural network,” *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 530-535, Lausanne, Switzerland, 2017.

[33] Y. Qu, J. Han, B. F. Cockburn, W. Pedrycz, Y. Zhang, and W. Zhao, “A true random number generator based on parallel STT-MTJs,” *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp.606-609, Lausanne, Switzerland, 2017.

[34] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, “Integer factorization using stochastic magnetic tunnel junctions,” *Nature*, vol. 573, no. 7774, pp.390-393, Sep. 2019.

[35] T. Ohsawa, F. Iga, S. Ikeda, T. Hanyu, H. Ohno, and T. Endoh, “High-Density and Low-Power Nonvolatile Static Random Access Memory Using Spin-Transfer-Torque Magnetic Tunnel Junction,” *Japanese Journal of Applied Physics*, vol. 51, p. 02BD01, Feb. 2012.

[36] A. Chen “A review of emerging non-volatile memory (NVM) technologies and applications,” *Solid-State Electronics*, vol. 125, pp. 25-38, July 2016.

[37] W. F. Brown, “Thermal Fluctuations of a Single-Domain Particle,” *Physical Review*, vol. 130, no. 5, pp.1677-1686, June 1963.

[38] L. Landau, and E. Lifshits, “On the theory of the dispersion of magnetic permeability in ferromagnetic bodies,” *Physikalische Zeitschrift der Sowjetunion*, vol. 8, pp. 153-169,

1935.

[39] V. Kamberský, "On the Landau–Lifshitz relaxation in ferromagnetic metals," *Canadian Journal of Physics*, vol. 48, no. 24, pp. 2906-2911, Dec. 1970.

[40] V. Kamberský, "On ferromagnetic resonance damping in metals," *Czechoslovak Journal of Physics B*, vol. 26, no. 12, pp. 1366-1383, 1976.

[41] T. L. Gilbert, "A phenomenological theory of damping in ferromagnetic materials," *IEEE Transactions on Magnetics*, vol. 40, no. 6, Nov. 2004.

[42] J. Xiao, A. Zangwill, and M. D. Stiles, "Macrospin models of spin transfer dynamics," *Physical Review B*, vol. 72, no. 1, p. 014446, July 2005.

[43] H. Liu, d. Bedau, J. Z. Sun, S. Mangin, E. E. Fullerton, J. A. Katine, and A. D. Kent, "Dynamics of spin torque switching in all-perpendicular spin valve nanopillars," *Journal of Magnetism and Magnetic Materials*, vol. 358-359, pp. 233-258, May 2014.

[44] K. Nagasaka, L. Varga, Y. Shimiza, S. Eguchi, and A. Tanaka, "The temperature dependence of exchange anisotropy in ferromagnetic / PdPtMn bilayers," *Journal of Applied Physics*, vol. 87, no. 9, pp. 6433-6435, May 2000.

[45] D. A. Garanin, "Fokker-Planck and Landau-Lifshitz-Bloch equations for classical ferromagnets," *Physical Review B*, vol. 55, no. 5, pp. 3050-3057, Feb. 1997.

[46] W. H. Butler, T. Mewes, C. K. A. Mewes, P. B. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, "Switching Distributions for Perpendicular Spin-Torque Devices within the Macrospin Approximation," *IEEE Transactions on Magnetics*, vol. 48, no. 12, pp. 4684-4700, Dec. 2012.

[47] H. Lee, A. Lee, S. Wang, F. Ebrahimi, P. Gupta, P. K. Amiri, and K. L. Wang, "Analysis and Compact Modeling of Magnetic Tunnel Junctions Utilizing Voltage-Controlled Magnetic Anisotropy," *IEEE Transactions on Magnetics*, vol. 54, no. 4, pp.1-9, Apr. 2018.

[48] J. A. Osborn, "Demagnetizing Factors of the General Ellipsoid," *Physical Review*, vol. 67, no. 11-12, pp. 351-357, June 1945.

[49] M. Sato, and Y. Ishii, "Simple and approximate expressions of demagnetizing factors of uniformly magnetized rectangular rod and cylinder," *Journal of Applied Physics*, vol. 66, no. 3, pp. 983-985, July 1989.

[50] A. H. Lone, S. Shringi, K. Mishra, and S. Srinivasan, “Cross-Sectional Area Dependence of Tunnel Magnetoresistance, Thermal Stability, an0d Critical Current Density in MTJ,” *IEEE Transactions on Magnetics*, vol. 57, no. 2, p. 4400410, Feb. 2021.

[51] N. Xu, F. Chen, D. Apalkov, W. Qi, J. Wang, Z. Jiang, W. Choi, and D. Kim, “Rare-Failure Oriented STT-MRAM Technology Optimization,” *IEEE Symposium on VLSI Technology*, pp. 187-188, Honolulu, HI, USA, June 2018.

[52] J. Song, H. Dixit, B. Behin-Aein, C. H. Kim, and W. Taylor, “Impact of Process Variability on Write Error Rate and Read Disturbance in STT-MRAM Devices,” *IEEE Transactions on Magnetics*, vol. 56, no. 12, p. 3400411, Dec. 2020.

[53] R. D. Rose, M. Lanuzza, M. d’Aquino, G. Carangelo, G. Finocchio, F. Crupi, and M. Carpentieri, “A Compact Model with Spin-Polarization Asymmetry for Nanoscaled Perpendicular MTJs,” *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4346-4353, Oct 2017.

[54] H. Liu, and T. Ohsawa, “User-friendly compact model of magnetic tunnel junctions for circuit simulation based on switching probability,” *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, China, Apr. 2019.

[55] H. Liu, and T. Ohsawa, “Compact Model of Magnetic Tunnel Junctions for SPICE Simulation Based on Switching Probability,” *IEICE Transactions on Electronics*, vol. 104, no. 3, pp.121-127, Mar. 2021.

[56] H. Liu, and T. Ohsawa, “Spin-Transfer Torque Magnetic Tunnel Junction Model Based on Fokker-Planck Equation for Stochastic Circuit Simulations,” *IEEE International Conference on Nanotechnology*, Palma de Mallorca, Spain, pp. 287-290, July 2022.

[57] H. Liu, and T. Ohsawa, “A Binarized Spiking Neural Network Based on Auto-Reset LIF Neurons and Large Signal Synapses Using STT-MTJs”, *Japanese Journal of Applied Physics*, vol. 62, no. 4, p. 044501, Apr. 2023.

[58] T. Aoki, Y. Ando, D. Watanabe, M. Oogane, and T. Miyazaki, “Spin transfer switching in the nanosecond regime for CoFeB/MgO/CoFeB ferromagnetic tunnel junctions,” *Journal of Applied Physics*, vol. 103, no. 10, p. 103911, May 2008.

[59] D. Bedau, H. Liu, J. Z. Sun, J. A. Katine, E. E. Fullerton, S. Mangin, and A. D. Kent, “Spin-transfer pulse switching: From the dynamic to the thermally activated regime,” *Applied Physics Letters*, vol. 97, no. 26, p.262502, Dec. 2010.

[60] K. Yagami, A. A. Tulapurkar, A. Fukushima, and Y. Suzuki, “Inspection of intrinsic critical currents for spin-transfer magnetization switching,” *IEEE Transactions on Magnetics*, vol. 41, no. 10, pp. 2615-1617, Oct. 2005.

[61] Z. Li and S. Zhang, “Thermally assisted magnetization reversal in the presence of a spin-transfer torque,” *Physical Review B*, vol. 69, no. 13, p. 134416, Apr. 2004.

[62] M. OSACI, “The Néel Magnetic Relaxation Dynamics in Nanoparticle Systems - Phenomenological Approach versus Stochastic Approach,” *Acta Physica Polonica A*, vol. 129, no. 3, pp. 323-328, Mar. 2016.

[63] Y. Zhang, W. Zhao, Y. Lakys, J. O. Klein, J. V. Kim, D. Ravelosona, and C. Chappert, “Compact Modeling of Perpendicular-Anisotropy CoFeB/MgO Magnetic Tunnel Junctions,” *IEEE Transactions on Electron Devices*, vol. 59, no. 3, pp. 819-826, Mar. 2012.

[64] Y, Wang, H. Cai, L. A. B. Naviner, Y. Zhang, J. O. Klein, and W. S. Zhao, “Compact thermal modeling of spin transfer torque magnetic tunnel junction,” *Microelectronics Reliability*, vol.55, no. 9-10, pp. 1649-1653, Aug. 2015.

[65] Y. Zhang, B. Yan, W. Kang, Y. Cheng, J. O. Klein, Y. Zhang, Y. Chen, and W. Zhao, “Compact Model of Subvolume MTJ and Its Design Application at Nanoscale Technology Nodes,” *IEEE Transactions on Electron Devices*, vol. 62, no. 6, pp. 2048-2055, June 2015.

[66] S. A. Dyer, and J. S. Dyer, “Cubic-spline interpolation. 1,” *IEEE Instrumentation & Measurement Magazine*, vol. 4, no. 1, pp. 44-46, Mar. 2001.

[67] G. D. Fuchs, I. N. Krivorotov, P. M. Braganca, N. C. Emley, A. G. F. Garcia, D. C. Ralph, and R. A. Buhrman, “Adjustable spin torque in magnetic tunnel junctions with two fixed layers,” *Applied Physics Letters*, vol. 86, no. 15, p.152509, Apr. 2005.

[68] G. Panagopoulos, C. Augustine, and K. Roy, “A framework for simulating hybrid MTJ/CMOS circuits: Atoms to system approach,” *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, 2012.

[69] S. Chandrasekhar, “Stochastic Problems in Physics and Astronomy,” *Reviews of Modern Physics*, vol. 15, no. 1, pp. 1-89, Jan. 1943.

[70] R. Kubo, and N. Hashitsume, “Brownian Motion of Spins,” *Progress of Theoretical Physics Supplement*, vol. 46, pp. 210-220, June 1970.

[71] M. C. Wang, and G. E. Uhlenbeck, “On the Theory of the Brownian Motion II,” *Reviews of Modern Physics*, vol. 17, no. 2-3, pp. 323-342, Apr. 1945.

[72] Y. Xie, B. Behin-Aein, and A. W. Ghosh, “Fokker-Planck Study of Parameter Dependence on Write Error Slope in Spin-Torque Switching,” *IEEE Transactions on Electron Devices*, vol. 64, no. 1, pp. 319-324, Jan. 2017.

[73] C. Grossmann, H. Roos, and M. Stynes, “Finite Difference Methods” in Numerical treatment of partial differential equations, Universitext. Springer, Berlin, Heidelberg, 2007, pp. 23-124.

[74] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, “Solution of Linear Algebraic Equations,” in Numerical Recipes: The Art of Scientific Computing, 3rd. ed., New York, Cambridge University Press, 2007, pp. 56-57.

[75] J. J. Nowak, R. P. Robertazzi, J. Z. Sun, G. Hu, J-H Park, J. Lee, A. J. Annunziata, G.P. Lauer, R. Kothandaraman, E. J. O’Sullivan, P. L. Trouilloud, Y. Kim, and D. C. Worledge, “Dependence of Voltage and Size on Write Error Rates in Spin-Transfer Torque Magnetic Random-Access Memory,” *IEEE Magnetics Letters*, vol.7, pp. 1-4, Mar. 2016.

[76] J. Zhu, and C. Park, “Magnetic tunnel junctions,” *Materials Today*, vol. 9, no. 11, pp. 36-45, Nov. 2006.

[77] G. Hu, J. J. Nowak, M. G. Gottwald, S. L. Brown, B. Doris, C. P. D’Emic, P. Hashemi, D. Houssameddine, Q. He, D. Kim, J. Kim, C. Kothandaraman, G. Lauer, H. K. Lee, N. Marchack, M. Reuter, R. P. Robertazzi, J. Z. Sun, T. Suwannasiri, P. L. Trouilloud, S. Woo, and D. C. Worledge, “Spin-Transfer Torque MRAM with Reliable 2 ns Writing for Last Level Cache Applications,” *IEEE International Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 2019.

[78] T. Ohsawa, H. Koike, S. Miura, H. Honjo, K. Kinoshita, S. Ikeda, T. Hanyu, H. Ohno, and T. Endoh, “A 1 Mb Nonvolatile Embedded Memory Using 4T2MTJ Cell With 32 b Fine-Grained Power Gating Scheme,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 6, June 2013.

[79] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.

[80] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving:

Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58443-58469, 2020.

[81] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneeershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1026-1034, 2015.

[83] K. Meier, "Special report: Can we copy the brain? - The brain as computer," *IEEE Spectrum*, vol. 54, no. 6, pp. 28–33, Jun. 2017.

[84] K. Yamazaki, V. Vo-Ho, D. Bulsara, and N. Le, "Spiking Neural Networks and Their Applications: A Review," *Brain Sciences*, vol.12, no. 7, p.863, June 2022.

[85] D. Lee, M. Kwak, K. Moon, W. Choi, J. Park, J. Yoo, J. Song, S. Lim, C. Sung, W. Banerjee, and H. Hwang, "Various Threshold Switching Devices for Integrate and Fire Neuron Applications," *Advanced Electronic Materials*, vol. 5, no. 9, p. 1800866, Feb. 2019.

[86] A. L. Hodgkin, and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, 1952.

[87] W. Gerstner, and W. M. Kistler, Spiking neuron models: single neurons, populations, plasticity, Cambridge University Press, Cambridge, 2002.

[88] E. M. Izhikevich, "Simple models of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569-1572, Nov. 2003.

[89] I. Yeo, M. Chu, and B. Lee, "A power and area efficient CMOS stochastic neuron for neural networks employing resistive crossbar array," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1678-1689, Dec. 2019.

[90] Y. Kuang, X. Cui, Y. Zhong, K. Liu, C. Zou, Z. Dai, D. Yu, Y. Wang, and R. Huang, "A 28-nm 0.34-pJ/SOP Spike-Based Neuromorphic Processor for Efficient Artificial Neural Network Implementations," *IEEE International Symposium on Circuits and Systems*

(ISCAS), Daegu, Korea, May 2021.

[91] D. Wang, P. K. Chundi, S. J. Kim, M. Yang, J. P. Cerqueira, J. Kang, S. Jung, S. Kim, and M. Seok, "Always-On, Sub-300-nW, Event-Driven Spiking Neural Network based on Spike-Driven Clock-Generation and Clock- and Power-Gating for an Ultra-Low-Power Intelligent Device," *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Hiroshima, Japan, S17-3, 2020.

[92] C. Frenkel, and G. Indiveri, "ReckOn: A 28nm Sub-mm2 Task-Agnostic Spiking Recurrent Neural Network Processor Enabling On-Chip Learning over Second-Long Timescales," *IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, pp.1-3, San Francisco, CA, USA, 2022.

[93] D. Zhang, L. Zeng, Y. Zhang, W. Zhao, and J. O. Klein, "Stochastic spintronic device based synapses and spiking neurons for neuromorphic computation," *IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp.173-178, Beijing, China, 2016.

[94] A. Sengupta, P. Panda, P. Wijesinghe, T. Kim, and K. Roy, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific reports*, vol. 6, no. 1, pp.1-8, July 2016.

[95] A. Sengupta, M. Parsa, B. Han, and K. Roy, "Probabilistic deep spiking neural systems enabled by magnetic tunnel junction," *IEEE Transactions on Electron Devices*, vol. 63, no. 7, pp. 2963-2970, July 2016.

[96] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning," *Scientific reports*, vol. 6, no.1, pp. 1-13, 2016.

[97] P. Wijesinghe, A. Ankit, A. Sengupta, and K. Roy, "An All-Memristor Deep Spiking Neural Computing System: A Step Toward Realizing the Low-Power Stochastic Brain," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no.5, pp. 345-358, Oct. 2018.

[98] D. Kim, W. Yi, J. Choi, K. Ashiba, J. Baek, H. Jun, J. Kim, and J. Park, "Double MgO-Based Perpendicular Magnetic Tunnel Junction for Artificial Neuron," F*rontiers in Neuroscience*, vol. 14, Apr. 2020.

[99] F. Liang, I. Wang, and T. Hou, "Progress and Benchmark of Spiking Neuron Devices and

Circuits," *Advanced Intelligent Systems*, vol. 3, no. 8, p. 2100007, June 2021.

[100] [online] Available: ptm.asu.edu.

[101] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[102] J. Kim, H. Kim, S. Huh, J. Lee, and K. Choi, "Deep neural networks with weighted spikes," *Neurocomputing*, vol. 311, no. C, pp. 373-286, Oct. 2018.

[103] P. U. Diehl, and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, no. 99, pp. 1-9, Aug. 2015.

[104] P. Panda, J. M. Allred, S. Ramanathan, and K. Roy, "ASP: Learning to Forget With Adaptive Synaptic Plasticity in Spiking Neural Networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 51-64, Mar. 2018.

[105] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Competitive STDP-based spike pattern learning," *Neural computation*, vol. 21, no. 5, pp. 1259-1276, May 2009.

[106] M. Koo, G. Srinivasan, Y. Shim, and K. Roy, "sBSNN: Stochastic-Bits Enabled Binary Spiking Neural Network With On-Chip Learning for Energy Efficient Neuromorphic Computing at the Edge," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 8, pp. 2546-2555, Mar. 2020.

[107] A. Stillmaker, and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration*, vol. 58, pp. 74-81, Jun. 2017.

[108] V. Milo, G. Malavena, C. M. Compagnoni, and D. Ielmini, "Memristive and CMOS Devices for Neuromorphic Computing," *Materials*, vol. 13, no. 1, p. 166, Nov. 2020.

[109] A. Nigam, C. W. Smullen, V. Mohan, E. Chen, S. Gurumurthi, and M. R. Stan, "Delivering on the promise of universal memory for spin-transfer torque RAM (STT-RAM)," *IEEE/ACM International Symposium on Low Power Electronics and Design*, Fukuoka, Japan, pp. 121-126, Aug. 2011.

[110] G. D. Panagopoulos, C. Augustine, and K. Roy, "Physics-Based SPICE-Compatible Compact Model for Simulating Hybrid MTJ/CMOS Circuits," *IEEE Transactions on Electron Devices*, vol. 60, no. 9, pp. 2808-2814, Sep. 2013.

[111] H. Lee, A. Lee, S. Wang, F. Ebrahimi, P. Gupta, P. K. Amiri, and K. L. Wang, “Analysis and Compact Modeling of Magnetic Tunnel Junctions Utilizing Voltage-Controlled Magnetic Anisotropy,” *IEEE Transactions on Magnetics*, vol. 54, no. 4, pp.1-9, Apr. 2018.

[112] M. M. Torunbalci, P. Upadhyaya, S. A. Bhave, and Y. Camsari, “Modular Compact Modeling of MTJ Devices,” *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4628-4634, Oct. 2018.

[113] F. Garcia-Redondo, P. Prabhat, and M. Bhargava, “A Fokker-Planck Solver to Model MTJ Stochasticity,” *IEEE 51st European Solid-State Device Research Conference (ESSDERC)*, Grenoble, France, pp. 263-266, Sep. 2021.

[114] J. Kim, A. Chen, B. Behin-Aein, S. Kumar, J. Wang, and C. H. Kim, “A technology-agnostic MTJ SPICE model with user-defined dimensions for STT-MRAM scalability studies,” *IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, CA, USA, pp. 1-4, Sep. 2015.

[115] R. D. Rose, M. Lanuzza, F. Crupi, G. Siracusano, R. Tomasello, G. Finacchio, M. Carpentieri, and M. Alioto, “A Variation-Aware Timing Modeling Approach for Write Operation in Hybrid CMOS/STT-MTJ Circuits,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 3 pp. 1086-1095, Mar. 2018.

[116] Z. Xu, K. B. Sutaria, C. Yang, C. Chakrabarti, and Y. Cao, “Compact modeling of STT-MTJ for SPICE simulation,” *European Solid State Device Research Conference (ESSDERC)*, Bucharest, Romania, pp. 338-341, Sep. 2013.

[117] Z. Xu, C. Yang, M. Mao, K. B. Sutaria, C. Chakrabarti, and Y. Cao, “Compact modeling of STT-MTJ devices,” *Solid-State Electronics*, vol. 102, pp. 76-81, Dec. 2014.

[118] I. Ahamed, Z. Zhao, M. G. Mankalale, S. S. Sapatnekar, J. Wang, and C. H. Kim, “A Comparative Study Between Spin-Transfer-Torque and Spin-Hall-Effect Switching Mechanisms in PMTJ Using SPICE,” *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 3, pp. 74-82, Dec. 2017.

[119] X. Yang, Y. Zhang, Y. Zhang, and P. Wang, “A Universal Compact Model for Spin-Transfer Torque-Driven Magnetization Switching in Magnetic Tunnel Junction,” *IEEE Transactions on Electron Devices*, vol. 69, no. 11, pp. 6453-6458, Nov. 2022.

[120] R. D. Rose, M. Lanuzza, F. Crupi, G. Siracusano, R. Tomasello, G. Finocchio, M.

Carpentieri, and M. Alioto, “A Variation-Aware Timing Modeling Approach for Write Operation in Hybrid CMOS/STT-MTJ Circuits,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 3, pp. 1086-1095, Mar. 2018.

[121] N. Sakimura, R. Nebashi, Y. Tsuji, H. Honjo, T. Sugibayashi, H. Koike, T. Ohsawa, S. Fukami, T. Hanyu, H. Ohno, and T. Endoh, “High-speed simulator including accurate MTJ models for spintronics integrated circuit design,” *IEEE International Symposium on Circuits and Systems (ISCAS)*, Seoul, Korea, pp. 1971-1974, May 2012.

[122] H. Wang, W. Kang, Y. Zhang, and W. Zhao, “Modeling and Evaluation of Sub-10-nm Shape Perpendicular Magnetic Anisotropy Magnetic Tunnel Junctions,” *IEEE Transactions on Electron Devices*, vol. 65, no. 12, pp. 5537-5544, Dec. 2018.

[123] M. Tzoufras, “Switching probability of all-perpendicular spin valve nanopillars,” *AIP Advances*, vol. 8, no. 5, p. 056002, May 2018.

Publications

Journal papers

[1] H. Liu, and T. Ohsawa, “Compact Model of Magnetic Tunnel Junctions for SPICE Simulation Based on Switching Probability,” *IEICE Transactions on Electronics*, vol. 104, no. 3, pp.121-127, Mar. 2021, doi: 10.1587/transele.2020ECP5011.

[2] H. Liu, and T. Ohsawa, “A Binarized Spiking Neural Network Based on Auto-Reset LIF Neurons and Large Signal Synapses Using STT-MTJs”, *Japanese Journal of Applied Physics*, vol. 62, no. 4, p. 044501, Apr. 2023, doi: 10.35848/1347-4065/acc9f4.

International Conference Papers

[1] H. Liu, and T. Ohsawa, “User-friendly compact model of magnetic tunnel junctions for circuit simulation based on switching probability,” *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, China, Apr. 2019, doi: 10.1109/VLSI-DAT.2019.8741646.

[2] H. Liu, and T. Ohsawa, “Spin-Transfer Torque Magnetic Tunnel Junction Model Based on Fokker-Planck Equation for Stochastic Circuit Simulations,” *IEEE International Conference on Nanotechnology*, Palma de Mallorca, Spain, pp. 287-290, July 2022, doi: 10.1109/NANO54668.2022.9928695.

Acknowledgements

I would like to express heartfelt gratitude to everyone who has provided assistance during my four-year doctoral journey. Without all of you, I would not have been able to successfully complete my studies.

First and foremost, I would like to express my gratitude to my advisor Professor Ohsawa. He not only guided my research but also demonstrated how to plan and balance life and study. His dedication to work and his kindness towards people will benefit me for a lifetime. The time we spent working together and traveling will always be cherished as my most precious memories. I feel extremely honored to have been his student.

Second, I would like to thank Professor Takahata, Professor Ikehashi and Professor Natsui, for their invaluable suggestions for my work. Their expertise in the field and thoughtful critiques significantly improved the quality of this work.

Third, I would like to thank Professor Ikenaga, Professor Kimura, Professor Makino, Professor Shinohara, Professor Yamasaki, Professor Yoshimasu, Professor Kakitsuka and all other Professors at IPS for their comments and reviews. The comments help me improve my dissertation and presentation.

I would like to thank all the members in our laboratory, who offered valuable discussions and encouragement during this research journey. I am also deeply grateful for their companionship, which allowed me to experience wonderful times together.

Finally, I am extremely grateful for the support of my family during this time. Their love and companionship have kept me in high spirits throughout my doctoral journey.