11 Stateful STT-MRAM-Based Logic for Beyond–Von Neumann Computing

Hiwa Mahmoudi, Thomas Windbacher, Viktor Sverdlov, and Siegfried Selberherr

CONTENTS

11.1 INTRODUCTION

The exponential growth of the semiconductor industry has successfully proceeded for about four decades, supported by continued improvement of complementary metal–oxide–semiconductor (CMOS) technology. For the next several years, there will be no apparent substitutes for the CMOS technology and its future development has already been charted by the International Technology Roadmap for Semiconductors (ITRS) [1]. However, fundamental physical limitations such as leakage, high power densities, process variability, and soaring costs will bring the scaling of the classical CMOS devices to an end [2,3]. Therefore, besides exploring and introducing new materials, device structures, and design technologies, investigating possible alternative technologies to replace or at least to supplement CMOS [4–9], is important to further enhance the performance of logic devices and circuits [10]. Right now there are many different devices under investigation with widely varying performance parameters, for example, energy, speed, area, and so on [11]. Unfortunately, their level of maturity is often insufficient for practical purposes and we believe that combining CMOS and spintronic devices is the most promising solution for the near future. For example, distributing nonvolatile memory elements over a CMOS logic circuit is expected to address some of these limitations by providing ultralow power and fast operation as this method eliminates static leakage (standby) power dissipation and reduces interconnection delay [12].

Spintronic devices, especially MgO-based magnetic tunnel junctions (MTJs), are strong candidates to replace CMOS-based memory due to their nonvolatility and compatibility with CMOS technology [13]. Despite the advantages of high speed and unlimited endurance, the first generation of MTJs, which utilized Oersted fields for magnetization switching, were unfavorable in terms of scalability and energy consumption. By using the spin-transfer torque (STT) [14,15] switching technique, the second generation of MTJs (STT-MTJ) eliminated the need for current lines adjacent to memory cells, which were required previously for generating a switching field. Thus, by using the same lines for reading and writing operations, the STT-MTJs are more scalable and allow for lower switching energies [16,17]. Magnetoresistive random-access memory (MRAM) with STT-MTJs as memory elements combines the speed of static RAM (SRAM), the density of dynamic RAM (DRAM), and the nonvolatility of flash memory and has all the characteristics of a universal memory [18]. Furthermore, MTJ technology is attractive for building logic configurations that combine nonvolatile memory and logic circuits (so-called logic-in-memory architecture) to overcome the leakage power issue [19,20].

In this chapter, we concentrate on *stateful* logic architectures, which realize intrinsic nonvolatile logic-in-memory by using nonvolatile memory elements simultaneously as latches and logic gates. Recently, it has been shown that the fundamental Boolean logic operation material implication (IMP) is naturally realized in a simple circuit combining a conventional resistor and two TiO_2 memristive switches [21]. In Section 11.2, stateful memristive implication logic gates are presented. It is shown that due to the error accumulation and low endurance of the memristive gates, STT-MTJ devices are preferable to build up stateful logic circuits, as they do not show error accumulation and exhibit almost unlimited endurance. In Section 11.3, STT-MTJ-based reprogrammable gates and implication gates are studied. Due to the easy integration of MTJs on top of a CMOS circuit, the MTJ-based logic gates can be generalized to large-scale nonvolatile circuits. In Section 11.4, it is shown how the functionality of STT-MRAM cells can be extended by including stateful logic operations. This inherently provides large-scale nonvolatile logic-in-memory architectures with zero standby power and allows shorter interconnection delays by eliminating the need for data transfer between separate memory and logic units. Thus, the Von Neumann architecture is no longer a prerequisite. MRAM-based logic is also well suited for parallel nonvolatile computations, as will be shown by an example.

11.2 MEMRISTIVE IMPLICATION LOGIC GATES

11.2.1 Material Implication

Material implication (IMP) is a fundamental two-input Boolean logic operation ($s \rightarrow t$) that reads "*s* implies *t*" or "if *s*, then *t*" and is equivalent to "(NOT *s*) OR *t*" ($\overline{s} + t$), as shown in Table 11.1. The operations IMP and NIMP (negated IMP) form a computationally complete logic basis in combination with any operation from the sets C and C', respectively, for which $C = \{\text{NOT, FALSE, XOR, NIMP}\}$ and $C' = \{\text{NOT, TRUE, XNOR, IMP}\}$, and are therefore able to compute arbitrary Boolean functions.

In addition to the AND, OR, and NOT operations, the IMP operation has been classified by Whitehead and Russell as one of the four basic logic operations [22]. However, by modeling Boolean logic with circuits built with relays and switches, Shannon founded modern digital electronics only based on AND, OR, and NOT operations [23]. Since then, the IMP operation has been ignored in digital electronics. Only recently, it was demonstrated that memristive switches intrinsically enable the IMP operation in a crossbar array [21].

11.2.2 TiO_2 Memristive Switches

The memristor or memory resistor is a fundamental circuit element predicted by Chua in 1971 [24]. However, the first physical realization of a memristor was demonstrated only in 2008 [25], based on ionic transport in a metal–insulator–metal structure (TiO_2 memristor). Shortly after that, STT memristors were proposed [26], based on the STT-induced magnetic domain wall motion [27]. The memristor can be thought of as a passive programmable resistor that preserves its electrical resistance when the power is turned off. The most obvious application of memristors is nonvolatile memory. Since a memristor has a behavior similar to synapses, another potential application is to use them in neuromorphic systems [28,29]. Furthermore, as the fourth basic passive circuit element, a memristor is also suited for analog circuit applications [30–32]. Its current–voltage (I–V) characteristics display a hysteretic

TABLE 11.1
Material Implication, IMP, and NIMP Truth Tables

State	$s\ t$	$s \rightarrow t$	$\overline{t \rightarrow s}$
1	0 0	1	0
2	0 1	1	1
3	1 0	0	0
4	1 1	1	0

behavior confined to the first and the third quadrants and pinched at the origin. Under large voltages, however, the memristor operates as a latch or digital switch between the two states characterized by low and high resistances.

Recently, the realization of the IMP logic operation in a memristive circuit including a conventional resistor and two TiO_2 memristive switches was demonstrated in [21]. The use of memristive devices for new computational architectures enables the application of the same elements as latches and logic gates called *stateful* logic [21], which inherently realizes a logic-in-memory architecture, allows the extension of nonvolatile electronics from memory to logical computing applications, and opens the door for a new computational paradigm.

Figure 11.1a shows a TiO_2 memristive device that consists of a TiO_2 thin film sandwiched between two platinum (Pt) electrodes. The thin film is divided into a (highly conducting) doped region and an (insulating) undoped region. The internal resistance of the device is equal to the sum of the variable resistance on each region.

$$R_{\text{int}} = R_{\text{doped}} + R_{\text{undoped}} \tag{11.1}$$

A simple linear memristor model [25] describes memristor resistance (memristance) as

$$R_{\text{int}} = \frac{w(t)}{w_{\max}} R_{\text{off}} + \left(1 - \frac{w(t)}{w_{\max}}\right) R_{\text{on}}, \tag{11.2}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\mu_v R_{\text{on}}}{w_{\max}} i(t) \tag{11.3}$$

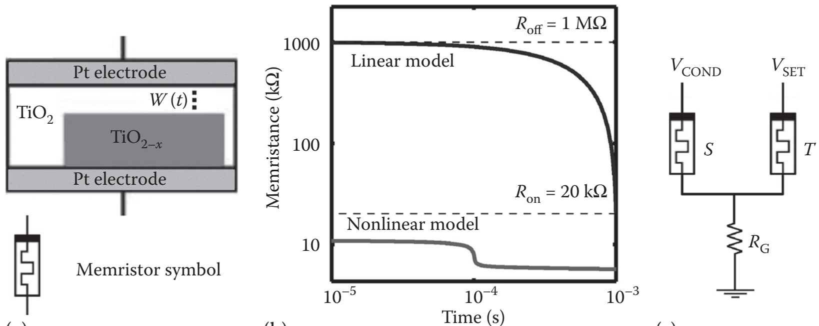


FIGURE 11.1 (a) TiO_2 memristive device and the circuit symbol of the memristor. (b) The memristance profile of the TiO_2 memristive device during a high-to-low resistance switching according to a simple linear model and the more advanced nonlinear model. (From Pickett, M.D., Strukov, D.B., Borghetti, J.L., Yang, J.J., Snider, G.S., Stewart, D.R., and Williams, R.S., *J. Appl. Phys.*, 106, 074508, 2009.) (c) The circuit topology of the memristive stateful implication logic gate. (From Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., and Williams, R.S., *Nature*, 464, 873–876, 2010.)

where:

R_{off} and R_{on} = maximum and minimum resistances, respectively
w and w_{max} = thickness of the insulating undoped region and its maximum possible value, correspondingly
μ_v = average mobility of dopants (oxygen vacancies) in TiO_2

According to this simplified model, the internal state variable w linearly changes with respect to charge flowing through the memristor. However, it was shown that the TiO_2 memristive devices have a more complex switching behavior, arising from ionic motion and also the modulation of an effective tunneling resistance with applied voltage (current) [33,34].

Figure 11.1b shows the dependence of the memristance during high-to-low resistance switching as it follows from the linear and the nonlinear model presented in [33]. Due to a high voltage level applied (V = 1.5 V), the tunneling effect through the insulating undoped region dominates the memristor I–V characteristics [33]. Therefore, during the switching, the total resistance is even lower than R_{on}, in contrast to the behavior predicted by the linear model according to Equation 11.2. This voltage-dependent memristance switching dynamics strongly affects the electrical properties of the device under voltage levels (>0.5 V) required for logic applications. Therefore, for modeling and optimizing the memristive stateful logic gates, the nonlinear model of the memristive devices has to be used [35]. In a high-voltage switching regime, the memristor acts as a bipolar two-resistance-state switch suited for storing binary data. A positive voltage (V_{OFF}) places the device in a high-resistance state (HRS; R_H), while a negative voltage (V_{ON}) places it in a low-resistance state (LRS; R_L).

11.2.3 Memristive Stateful Logic

Figure 11.1c shows the circuit topology of the memristive implication logic gate combining two TiO_2 memristors, S, and T, with a conventional resistor R_G. The initial resistance states of the source (S) and target (T) memristors (denoted by the logic variables s and t, respectively) are the logic inputs of the gate. The final resistance state of T after performing the logic operation (t') is the logic output of the gate. Performing the logic operation ($t' = s \rightarrow t$) involves simultaneous application of two negative voltage pulses, V_{SET} and V_{COND}, to the noncommon terminals of S and T. V_{COND} is a negative voltage with smaller amplitude than V_{SET}. Therefore, the voltage drop on S is smaller than V_{ON} and it remains unchanged after the operation for any input patterns. However, depending on the resistance state of S, the voltage V_{COND} changes the voltage level on the common terminal of S and T to modulate the voltage drop on the target memristor T. This provides a conditional switching behavior in T, which is shown in Table 11.2. In fact, the negative voltage pulse V_{SET} enforces a high-to-low resistance switching of T only when both memristors are initially in the HRS (State 1). The voltage V_{SET} has a higher amplitude than V_{ON} as it must compensate for the voltage drop on R_G.

According to Table 11.2, depending on the logical definitions for the memristor low (LRS) and high (HRS) resistance states, LRS ≡ logic 1 and HRS ≡ logic 0 or vice versa; the realized conditional switching behavior corresponds to the IMP or NIMP

TABLE 11.2
Realized Conditional Switching Behavior Is Equivalent to the Operation IMP or NIMP Depending on the Definitions for the HRS and LRS as Logical 0 and 1

	Implication Operation (Conditional Switching)		HRS ≡ 0, LRS ≡ 1		HRS ≡ 1, LRS ≡ 0	
			$t' = s \to t$		$t' = \overline{t \to s}$	
State	***s t***	***s′ t′***	***s t***	***t′***	***s t***	***t′***
1	HRS HRS	HRS LRS	0 0	**1**	1 1	**0**
2	HRS LRS	HRS LRS	0 1	1	1 0	0
3	LRS HRS	LRS HRS	1 0	0	0 1	1
4	LRS LRS	LRS LRS	1 1	1	0 0	0

(negated IMP) operation (Table 11.1). In accordance with the convention of Shannon, if we define HRS ≡ 1 and LRS ≡ 0, the logic output of the implication gate corresponds to the NIMP operation as

$$\left\{t' = t\,\mathrm{NIMP}\,s\right\} \equiv \overline{t \to s} \equiv \left\{t' = t \cdot \overline{s} = t\,\mathrm{AND}\,\overline{s}\right\}, \tag{11.4}$$

where t' is the final state of the variable t after the operation. In combination with the low-to-high resistance switching that corresponds to the TRUE operation (writing logic 1) according to the above definition, the NIMP operation forms a complete logic basis to compute any Boolean function. Therefore, it enables stateful logic operations by memristive devices used simultaneously as nonvolatile memory and logic gates [21]. Stateful universal NOR and NAND operations can be performed in three and five sequential steps, respectively, as

$$\begin{aligned}
&\text{TRUE: } a = 1\\
&\text{NIMP: } \overline{a \to b} \equiv \left\{a' = a \cdot \overline{b} = \overline{b}\right\}\\
&\text{NIMP: } \overline{a \to c} \equiv \left\{a' = a \cdot \overline{c} = \overline{b} \cdot \overline{c} = \overline{b + c} = b\,\mathrm{NOR}\,c\right\}
\end{aligned} \tag{11.5}$$

$$\begin{aligned}
&\text{TRUE: } a = 1\\
&\text{NIMP: } \overline{a \to b} \equiv \left\{a' = a \cdot \overline{b} = \overline{b}\right\}\\
&\text{NIMP: } \overline{c \to a} \equiv \left\{c' = c \cdot \overline{a} = c \cdot b\right\}\\
&\text{TRUE: } a = 1\\
&\text{NIMP: } \overline{a \to c} \equiv \left\{a' = a \cdot \overline{c} = \overline{c \cdot b} = b\,\mathrm{NAND}\,c\right\}
\end{aligned} \tag{11.6}$$

where *a* (*a′*) represents the initial (final) logic variable equivalent to the resistance state of a third memristor storing the logic result of intermediary logic steps and the final result of stateful NAND and NOR operations.

As explained before, by applying the negative voltage pulses V_{SET} and V_{COND}, a (desired) high-to-low resistance switching (shown in bold in Table 11.2) is enforced in T only in State 1. However, the current flowing through the memristors tends to decrease their electrical resistances and changes their internal state variable *w*. This phenomenon is called *state drift* [36], and its accumulation after a specific number of sequential (N)IMP operations causes an undesired switching event either in S or T. In fact, although the TiO_2 memristive switches are used for binary data storage, they act as analog circuit elements as the parameter *w* changes continuously [33,34]. Therefore, the memristive implication logic architecture requires states to be refreshed to avoid undesired switching events. In MTJ-based logic gates [37–39] the need for refreshing circuits is eliminated as the MTJ has a bistable (parallel/antiparallel) magnetization configuration.

11.3 MTJ-BASED STATEFUL LOGIC GATES

11.3.1 Background

The basic MTJ structure consists of a free and a pinned ferromagnetic layer separated by a tunneling oxide. The magnetization of the free layer has a bistable configuration and can be switched between a parallel and an antiparallel state compared to the fixed magnetization direction of the pinned layer. The electrical resistance of the device depends on the relative orientation of the magnetization directions. An antiparallel alignment results in a high-resistance state (HRS; R_{AP}) of the MTJ, while the parallel alignment places it in a low-resistance state (LRS; R_P). The MTJ resistance modulation is described by the tunneling magnetoresistance (TMR) ratio, defined as $TMR = (R_{AP} - R_P)/R_P$. The tunneling oxide is usually MgO. Due to additional spin filtering, MgO-based MTJs exhibit a high TMR ratio, which facilitates reading-out of the MTJ resistance state via the TMR effect [40,41]. Compared to the first generation of MTJs, which utilized an Oersted field for switching, the STT switching technique exhibits pure electrical read/write operations and has significant advantages with respect to scalability and energy consumption [16]. This makes the STT-MTJ a suitable candidate for a universal memory that combines the advantages of CMOS compatibility, nonvolatility, high switching speed, high integration density, unlimited endurance, and scalability.

Recently, the realization of MTJ-based nonvolatile logic gates has been demonstrated: the MTJ devices are used simultaneously as nonvolatile memory cells and main computing elements (logic gates) [37–39]. In [37,38], reprogrammable logic gates (Figure 11.2a and b) realize the basic Boolean logic operations AND, OR, NAND, NOR, and the Majority operation. The MTJ-based logic gates reported in [39] are designed on the conventional voltage-controlled IMP circuit topology (Figure 11.2c) and also a beneficial current-controlled topology (Figure 11.2d) to realize stateful implication operations. Compared to the TiO_2 memristive switches, MTJs provide a higher number of cycles for reversibly and reliably switching (so-called endurance),

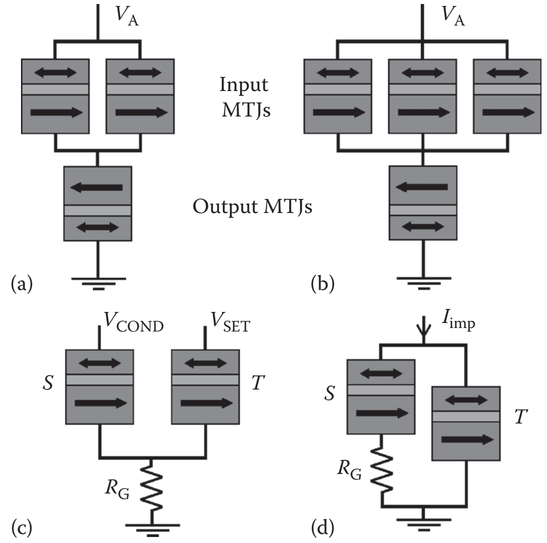


FIGURE 11.2 MTJ-based two-input (a) and three-input (b) reprogrammable logic gates. (c) VC-IMP and (d) CC-IMP logic gates. The resistance states of the target (T) and the source (S) MTJs act as the inputs (*t* and *s*) and the final logic state of T (t') is the output of the implication gates.

which is still a major challenge for memristors to be used as universal memory cells or computing elements [42]. In fact, MTJs show at least three orders of magnitude higher endurance [43]. Furthermore, the bistable resistance state of the MTJs eliminates the need for refreshing circuits, which is required in memristive computation due to the state drift phenomenon, as explained before. According to the simulation results from [35,36], without refreshing the states in a memristive implication gate, state drift can cause one bit error in less than ten sequential logic steps.

11.3.2 Analysis and Modeling

11.3.2.1 Reprogrammable Gates

Two-input and three-input reprogrammable logic gates [37,38] are shown in Figure 11.2a and b, respectively. Using these gates, the basic Boolean logic operations are executed in two sequential steps including an appropriate preset operation (parallel or antiparallel state) in the output MTJ and then applying a voltage pulse (V_A) with a proper amplitude to the gate. Depending on the logic states of the input MTJs, the preset in the output MTJ, and the voltage level applied to the gate, a conditional switching behavior in the output MTJ is provided that corresponds to a particular logic operation [38].

Table 11.3 illustrates how the AND, OR, NAND, and NOR operations are performed employing a two-input reprogrammable gate in two steps. The variables *s* and *t* show the logic states of the input MTJs and *y* represents the logic state of the output MTJ. In order to perform a logic operation, first a preset of $y = 1$ (putting in

TABLE 11.3
Realized Conditional Switching Behavior Is Equivalent to the AND, OR, NAND, and NOR Operations Using the Two-Input Reprogrammable Gate

Input Patterns			$y' \leftarrow s$ AND t		$y' \leftarrow s$ OR t		$y' \leftarrow s$ NAND t		$y' \leftarrow s$ NOR t	
			Step 1	**Step 2**	**Step 1**	**Step 2**	**Step 1**	**Step 2**	**Step 1**	**Step 2**
State	s	t	y	y'	y	y'	y	y'	y	y'
1	LRS (0)	LRS (0)	HRS (1)	**LRS (0)**	HRS (1)	**LRS (0)**	LRS (0)	**HRS (1)**	LRS (0)	**HRS (1)**
2	LRS (0)	HRS (1)	HRS (1)	**LRS (0)**	HRS (1)	HRS (1)	LRS (0)	**HRS (1)**	LRS (0)	LRS (0)
3	HRS (1)	LRS (0)	HRS (1)	**LRS (0)**	HRS (1)	HRS (1)	LRS (0)	**HRS (1)**	LRS (0)	LRS (0)
4	HRS (1)	HRS (1)	HRS (1)	HRS (1)	HRS (1)	HRS (1)	LRS (0)	LRS (0)	LRS (0)	LRS (0)

Notes: The variable y represents the logic (resistance) state of the output MTJ and the desired switching events are indicated by boldface type.

the HRS) or $y = 0$ (putting in the LRS) is performed in the output MTJ (Step 1). In Step 2, a proper voltage level ($V_A < 0$ or $V_A > 0$ with optimized amplitude according to Figure 11.3) is applied to the gate to enforce the desired (high-to-low or low-to-high) resistance switching events in the output MTJ to execute the logic operations AND/OR or NAND/NOR. Compared to the (N)AND operation, the (N)OR operation requires a lower voltage amplitude ($|V_A|$), as it must enforce a desired switching event only if both input MTJs are in the LRS (State 1). For the (N)AND operation, the switching events are enforced not only in State 1, but also when only one of the inputs is in the LRS (State 2 and State 3). These switching events in State 2 and State 3 are desired switching events for the (N)AND operation, while they are undesired events for the (N)OR operation.

For given MTJ device characteristics, the values of the circuit parameter V_A have to be optimized to ensure a reliable conditional switching behavior of the output MTJ for any possible input patterns. Indeed, for any logic operation performed by the reprogrammable gates, this optimization is required to maximize/minimize the switching probability in the output MTJ ($P \to 1$ or $P \to 0$) when it is a

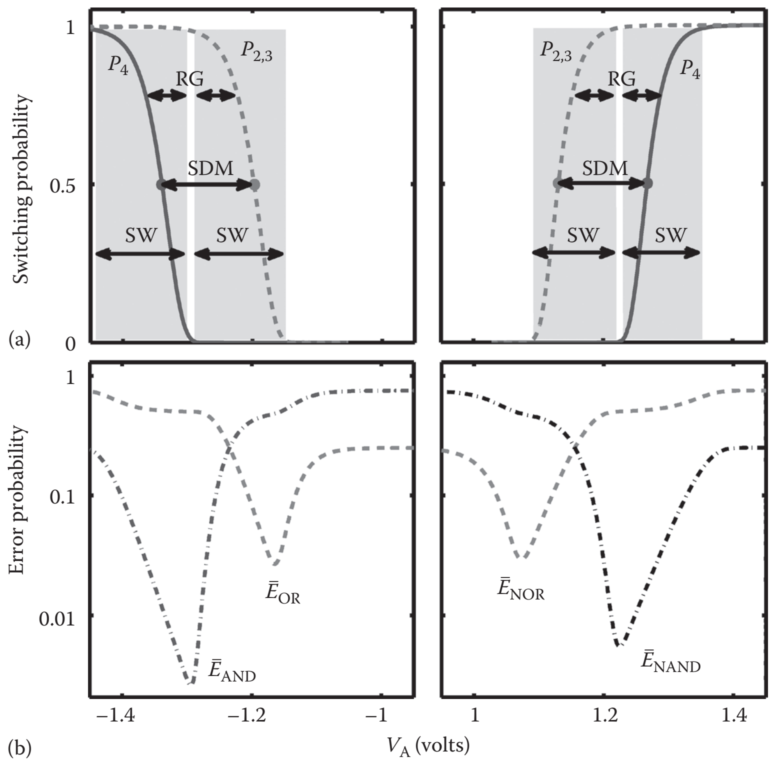


FIGURE 11.3 (a) The switching probabilities of the nearest desired ($P_{2,3}$) and undesired (P_4) switching events shown for the AND (*left side*) and NAND (*right side*) operations. (b) The average error probabilities for the basic reprogrammable operations as a function of V_A. MTJs are characterized as TMR = 200%, $\Delta = 40$, I_{C0} (AP → P) = 325 μA, and $R_P = 1.8$ kΩ.

desired/undesired switching event in Step 2. Therefore, for the reliability analysis [44], the error probability of a given input state (State i) is defined as $E_i = 1 - P_i$ ($E_i = P_i$) for a desired (undesired) switching event, where P_i is the switching probability of the output MTJ in State i. It should be noted that the input MTJs are left unchanged and their switching probabilities are negligible as the current flowing through the output MTJ splits between the inputs, and their currents are smaller than the critical current level required for the STT switching.

It is obvious that a reliable logic behavior of an operation is ensured only when the logic gate exhibits correct functionality for all input patterns. Therefore, by assuming equal incidence probabilities for all input patterns, we obtain the average error probability ($\bar{E}_\mathrm{b}$) of a basic logic operation b implemented by the reprogrammable gate as

$$\bar{E}_\mathrm{b} = \frac{1}{2^n} \sum_{i=1}^{2^n} E_\mathrm{i}, \qquad (11.7)$$

where n is the number of input MTJs. According to Equation 11.7 and by using Table 11.3, the average reliabilities of the two-input (N)OR and (N)AND operations are obtained as

$$\bar{E}_\mathrm{OR} = \bar{E}_\mathrm{NOR} = \frac{1}{4} \sum_{i=1}^{4} E_\mathrm{i} = \frac{1}{4}\left[\left(1 - P_1\right) + P_2 + P_3 + P_4\right], \qquad (11.8)$$

$$\bar{E}_\mathrm{AND} = \bar{E}_\mathrm{NAND} = \frac{1}{4} \sum_{i=1}^{4} E_\mathrm{i} = \frac{1}{4}\left[\left(1 - P_1\right) + \left(1 - P_2\right) + \left(1 - P_3\right) + P_4\right], \qquad (11.9)$$

where P_i is the switching probability of the output MTJ in State i. It is important to note that for the AND and OR operations P_i represents the probability for antiparallel-to-parallel (AP-to-P) magnetization switching, while it is the probability for the P-to-AP switching in the case of NAND and NOR operations. In order to calculate P_i for different input patterns of various logic operations, we use the theoretical model [45] for the thermally activated switching regime (switching time $t > 10$ ns), which has been proved experimentally in [46] and is given as

$$P = 1 - \exp\left\{-\frac{t}{\tau_0} \exp\left[-\Delta\left(1 - \frac{I}{I_{\mathrm{C0}}}\right)\right]\right\}, \qquad (11.10)$$

where:

t = current pulse duration

$\tau_0 \sim 1$ ns

I = current flowing through the MTJ

I_{C0} = critical high-to-low (or low-to-high) resistance switching current extrapolated to τ_0 [47]

Δ = thermal stability factor and is equal to E/k_BT, where E is the energy barrier between the parallel and the antiparallel magnetization states of the MTJ, k_B is the Boltzmann constant, and T is the temperature

In order to calculate the current flowing through each MTJ, the voltage-dependent effective TMR model [48] is used, which determines the $R-V$ characteristics of the MTJs in the HRS as

$$R_{AP} = \left(1 + \text{TMR}_{\text{eff}}\right) R_P = \left(1 + \frac{\text{TMR}_0}{1 + V^2/V_h^2}\right) R_P \tag{11.11}$$

TMR_0 and TMR_{eff} are the TMR ratio under zero and nonzero bias voltage (V) across the MTJ, and V_h is the bias voltage equivalent to $\text{TMR}_{\text{eff}} = \text{TMR}_0/2$, respectively.

Figure 11.3 shows different values of P_i and $\bar{E}_b$ for different logic operations for the two-input reprogrammable gate as a function V_A. For a voltage level V_A optimized within the reliable gap (RG in Figure 11.3a), which is opened between the switching windows (SWs) of the nearest desired ($P_{2,3}$) and undesired (P_4) switching events, the average error probability is minimized. Figure 11.3b illustrates that the (N)AND operation exhibits a better error probability than the (N)OR operation. It can be shown that the modulation in the total resistance at the input MTJs is higher, when the input resistance in State 2 (or State 3) is compared to its value in State 4 rather than its value in State 1. As a result, the current flowing through the output MTJ has a higher modulation, when State 2 (or State 3) is compared to State 4 rather than to State 1. Therefore, the (N)AND operation provides a more reliable behavior than the (N)OR operation.

11.3.2.2 Implication Gates

Similar to the memristive stateful implication gate (Table 11.2) in the voltage- and current-controlled implication (CC-IMP) gates (Figure 11.2c and d), the logic operation (N)IMP is realized based on a conditional switching in the target MTJ (T). Depending on the initial resistance states of the source and the target MTJs, an AP-to-P STT switching event is enforced in the target MTJ only, when both MTJs are initially at antiparallel (high-resistance) states (State 1). For the other input patterns (States 2, 3, and 4), the resistance states of the MTJs are left unchanged, as shown in Table 11.2. In the MTJ-based voltage-controlled implication (VC-IMP) gate (Figure 11.2c), the logic operation is executed by simultaneously applying the voltage pulses V_{COND} and V_{SET}. As $|V_{COND}| < |V_{SET}|$, the voltage drop on S is smaller than the critical voltage level required for STT switching, and thus S is left unchanged. The resistance state of S provides a voltage modulation across T through R_G. Due to this modulation, T switches when S is in the HRS (State 1), but remains unchanged when S is in the LRS (State 3).

In the CC-IMP gate (Figure 11.2d), the logic operation is performed by applying the current pulse I_{imp} to the gate [39]. I_{imp} is applied in a direction that tends to enforce AP-to-P switching events for both MTJs. The current I_{IMP} is split between S and T inversely proportional to the total resistance of each branch. The current split depends on the input pattern as the resistance value of each branch is a function of the logic state of its MTJ. According to Table 11.2, there are four possible AP-to-P switching events containing State 1 and State 3 for T and State 1 and State 2 for S. However, the only desired switching event is the AP-to-P switching event in T in State 1, and the other three are undesired events. In State 1, both S and T are in the HRS. However, due to R_{G}, a majority part of the current I_{IMP} flows through T. This current is higher than the critical current required for AP-to-P switching. But when S is in the LRS (State 3), the current flowing through T is decreased below the critical level required for AP-to-P switching as the resistance (current) of the other branch is decreased (increased). Because of R_{G}, the current flowing through S is always smaller than the critical current level required for AP-to-P STT switching and thus S is left unchanged in State 1 and State 2.

It is worth mentioning that unlike in the memristive implication gate, the effects of the voltages or currents during the logic operations, which tend to enforce undesired switching events in memory cells, are not accumulated due to the intrinsic damping of the MTJs' free layer [49]. Therefore, the magnetization direction of the free layer can relax to its initial state when there is enough time (in the range of sub-nanoseconds [50]) between sequential (N)IMP operations. In comparison, in memristors, the internal state variable w drifts after each (N)IMP operation and the errors of sequential logic steps are accumulated. Thus, refreshing circuits are required to avoid the error due to state drift accumulation.

The reliability of the (N)IMP operation in State 1 is proportional to the multiplication of the probability of the desired switching event in T (P_{t1}) and the term $1 - P_{\mathrm{s1}}$, where P_{s1} is the probability of the undesired switching event in S. Therefore, in State 1, the error probability (E_1) is

$$E_1 = 1 - P_{\mathrm{t1}}\left(1 - P_{\mathrm{s1}}\right) \tag{11.12}$$

In States 2 and 3, there are only undesired switching events (P_{s2} and P_{t3}) in S and T, respectively. Therefore, the error probabilities are given by

$$E_2 = P_{\mathrm{s2}}, \quad E_3 = P_{\mathrm{t3}} \tag{11.13}$$

When both MTJs are in the LRS (State 4), there is no possible switching event and the error probability E_4 is zero. Similar to the reprogrammable gate, the average error probability of the (N)IMP operation ($\bar{E}_{\mathrm{IMP}}$) is obtained by using Equation 11.7.

$$\bar{E}_{\mathrm{IMP}} = \frac{1}{4}\left(1 - P_{\mathrm{t1}} + P_{\mathrm{s1}}P_{\mathrm{t1}} + P_{\mathrm{s2}} + P_{\mathrm{t3}}\right) \tag{11.14}$$

From a circuit point of view, the parameters (I_{imp} and R_{G} in the CC-IMP and V_{COND}, V_{SET}, and R_{G} in the VC-IMP gates) can be optimized to minimize the error

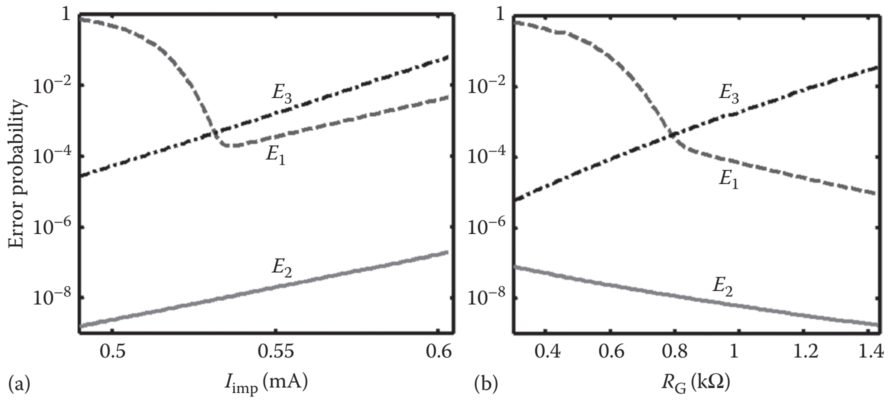


FIGURE 11.4 The error probabilities E_i for different input states of the CC-IMP logic gate as a function of I_{imp} (a) and R_G (b) plotted for MTJ devices with TMR = 250%, $\Delta = 40$, I_{C0} (AP → P) = 325 μA, and $R_P = 1.8$ kΩ.

probability $\overline{E}_{\text{IMP}}$ for given MTJ device characteristics. Figure 11.4a shows the error probabilities E_i for different input states of the CC-IMP gate as a function of I_{imp} for a fixed R_G. I_{imp} has to be high enough to enforce a desired switching of T in State 1. However, there is an optimum I_{imp}, as increasing I_{imp} increases the probabilities of possible undesired switching events in both T and S in States 1, 2, and 3. In the CC-IMP gate, R_G provides a structural asymmetry that increases the current flowing through T compared to S, when both MTJs are in the HRS (State 1). Therefore, increasing R_G reduces the error probability E_1, as it increases (decreases) the probability of the desired (undesired) switching event P_{t1} (P_{s1}), as shown in Figure 11.4b. However, its maximum value is limited by E_3. In State 3, S is in the LRS and thus the current flowing through S is higher than in State 1. Therefore, the current flowing through T is decreased to below the critical current required for the STT switching. Because higher R_G decreases the effective resistance modulation of its corresponding branch (the source branch comprising R_G and S), it increases the error probability E_3 (Figure 11.4b).

11.3.3 Results and Discussion

The logic implementation using MTJ-based logic gates relies on a conditional switching behavior provided by state-dependent current modulations on the output (target) MTJs. These modulations are caused by the changes in the MTJs' resistances for different initial logic states. According to Equation 11.11, the resistance modulation between the HRS and LRS in the MTJ with antiparallel and parallel magnetization alignments is proportional to the TMR ratio of the MTJs. Therefore, from a device point of view, the average error probabilities of all MTJ-based operations are expected to decrease with increasing TMR ratio, which is, therefore, considered as the most important device parameter for reliability [44].

For the reprogrammable gate, the width of the RG opened between the switching probabilities (P_i in Figure 11.3) is enlarged for a higher TMR ratio, as the difference between the different input states ($i=1$, 2, 3, or 4) originates from the modulation between the HRS and the LRS of the MTJs (Table 11.3). It is obvious that the difference between the HRS and LRS states of one input MTJ causes a higher current modulation in the output MTJ when the reprogrammable gate has two input MTJs compared to the three-input reprogrammable gate. Thus, the two-input gate exhibits a more reliable logic behavior [44]. Moreover, it has been demonstrated that the CC-IMP enables a more energy-efficient and reliable implementation [39]. Accordingly, in the following we employ only the two-input reprogrammable and the CC-IMP logic gates for the performance analysis and comparison of the reprogrammable and implication logic architectures.

In the CC-IMP gate, a higher R_G reduces the error probabilities in State 1, however, its value is limited by the required current modulation in State 3, as explained before (Figure 11.4b). Figure 11.5a shows the two dominant error probabilities (E_1 and E_3) for two different TMR values. It illustrates that a higher TMR has a negligible effect on E_1, but it decreases E_3 significantly. In fact, the current modulation between State 1 and State 3 relies on the MTJ resistance modulation described by the TMR ratio, and a higher TMR provides a higher modulation and thus allows higher values of R_G for the CC-IMP circuit parameter design.

In order to investigate the effect of the MTJ device parameters on the reliability of the MTJ-based logic gates, one has to optimize the circuit parameters (V_A for the reprogrammable gate [Figure 11.3b] and I_{imp} and R_G for the CC-IMP gate [Figure 11.5b]) in order to find and compare the minimum possible error probabilities of the gates for the same MTJ technology. Figure 11.6a compares the average error probabilities ($\bar{E}$) of different logic operations using the CC-IMP gate ((N) IMP operation) and a two-input reprogrammable logic gate (AND, OR, NAND, and NOR operations) as a function of TMR ratio with optimized circuit parameters for each TMR value [44]. As expected, it shows that the error decreases with increasing

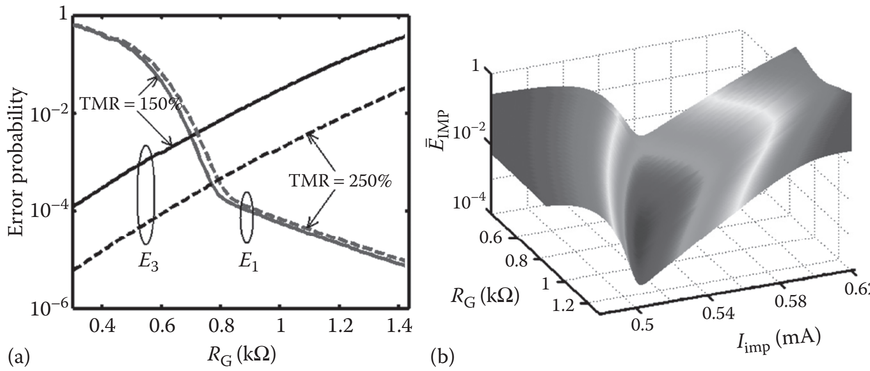


FIGURE 11.5 **(See color insert)** (a) Dominant error probabilities (E_1 and E_3) for different TMR values. (b) Circuit parameter optimization in the CC-IMP gate.

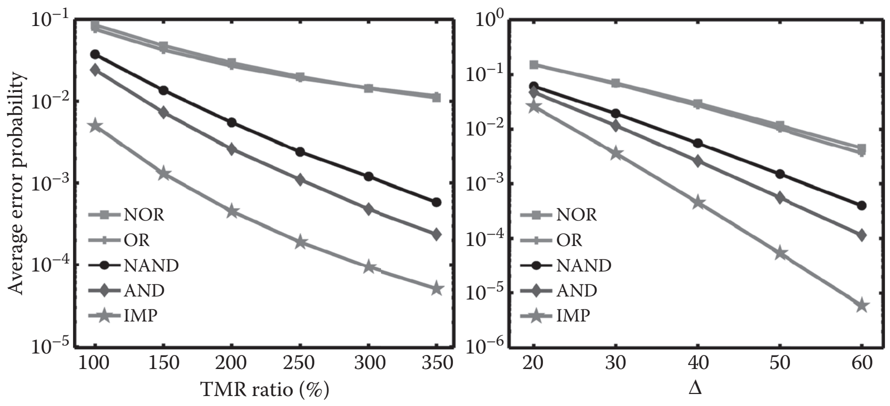


FIGURE 11.6 Average error probabilities $(\bar{E})$ for the basic operations of the CC-IMP and the two-input reprogrammable gates as a function of the (a) TMR ratio (with $\Delta = 40$) and (b) thermal stability factor Δ (with TMR = 200%).

TMR ratio. Furthermore, for the same MTJ device characteristics, the basic operation implemented in the implication logic architecture is more reliable than the basic operations based on the reprogrammable gate.

As mentioned before, the TMR ratio is considered as the most important device parameter for the reliability of the conditional switchings in MTJ-based logic gates. However, other MTJ device parameters also affect the reliability [51]. According to Equation 11.10, in the thermally activated switching regime, the dominant term for the switching probability calculation is $\exp[-\Delta(1 - I/I_{C0})]$. The modulation of the term I/I_{C0} depends on the TMR ratio and its impact has been studied before. Nevertheless, a higher Δ magnifies the effect of this modulation. Therefore, for all MTJ-based logic operations, a higher Δ decreases the error probabilities, as shown in Figure 11.6b. The effect of the pulse durations (t) is negligible as compared to the internal exponential term in Equation 11.10. Furthermore, the error values are independent of the absolute values of I_{C0} and R_P, as the computations can be generalized by normalizing all current and resistance values to I_{C0} and R_P, respectively. In Section 11.4, we describe how the one-transistor/one-MTJ (1T/1MTJ) cell can be employed to realize hybrid CMOS/MTJ logic. As the 1T/1MTJ cell is the basic structure in the STT-MRAM architecture [46], this topology allows extension of the functionality of the STT-MRAM cells to perform reprogrammable and implication logic operations.

11.4 HYBRID CMOS/MTJ TECHNOLOGY

In the common STT-MRAM architecture (Figure 11.7), the 1T/1MTJ cell contains one MTJ to store binary data and an access transistor to control the current flowing through the MTJ. The cells are coupled in parallel between the current-carrying source lines (SLs) and bit lines (BLs), as shown in Figure 11.7. The gate terminals of the access transistors are coupled to the word lines (WLs) in order to apply proper voltage signals to a specific MTJ for read/write operations (memory mode) through

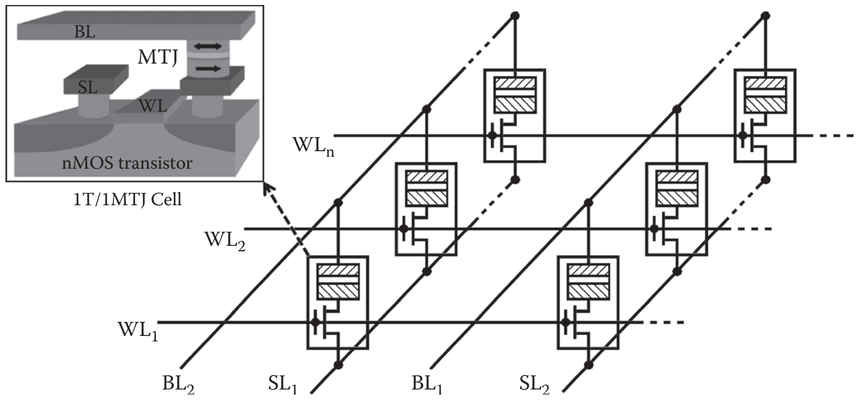


FIGURE 11.7 The common STT-MRAM architecture based on the 1T/1MTJ structure. (From Hosomi, M., Yamagishi, H., Yamamoto, T., Bessho, K., Higo, Y., Yamane, K., Yamada, H. et al., *IEEE International Electron Devices Meeting (IEDM) Technical Digest*, pp. 459–462. 5–5 December, Washington, DC, 2005.)

the SL and the BL in the MRAM array. For the read operation, a select voltage and a read current are applied to the specific WL and BL. The reading current flows through the selected MTJ, and by sensing the generated voltage difference between the SL and the BL, the resistance (logic) state of the selected MTJ is sensed. The read current must be low enough to prevent an undesired switching, which is referred to as a read disturbance. During a write operation, a select voltage and a write current (voltage) are applied to the specific WL and BL. According to the polarity of the current (voltage) applied to the current-carrying lines (SL and BL), the AP-to-P or P-to-AP switching is enforced in the selected MTJ depending on the desired binary data (logical 0 or 1). In the following, it will be shown how STT-MRAM arrays and MTJ logic gates can be combined to provide large-scale nonvolatile logic-in-memory architectures without the need for CMOS logic units.

11.4.1 MRAM-Based Reprogrammable Logic Arrays

Figure 11.8 shows two STT-MRAM arrays that are connected in series. In the logic mode, the access transistors of the 1T/1MTJ cells are used to simultaneously select two MTJs (inputs) in one array and one MTJ (output) in the other array. Due to the serial connection of the arrays, the current flowing through the output MTJ has the reverse polarity direction compared to the input MTJs. Therefore, the three simultaneously selected MTJs (two inputs and one output) form the circuit topology required for the two-input reprogrammable gate are shown in Figure 11.2a. By applying the voltage difference V_A to the BLs of the arrays, the desired switching (Step 2 in Table 11.3) is enforced in the output MTJ. Depending on the specified basic logic operation (AND, OR, NAND, or NOR), V_A has to be optimized, as shown in Figure 11.3. An accordant preset (Step 1) is performed in the output MTJ beforehand by selecting the corresponding access transistor and applying the write current/voltage signal to the BL and SL of the output array, as in the common write operations in the memory mode.

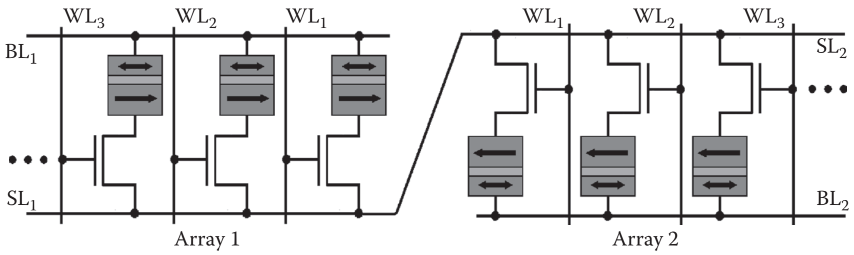


FIGURE 11.8 Proposed MRAM-based reprogrammable logic architecture including two common STT-MRAM arrays connected in series. (From H. Mahmoudi, T. Windbacher, V. Sverdlov, S. Selberherr, *Proceedings of the International Conference on Nanoscale Magnetism*, p. 208. 2–6 September, Istanbul, Turkey, 2013.)

Compared to the MTJ-based reprogrammable circuits (Figure 11.2a and b), which require extra wiring for generating a current-induced Oersted field for independent access to the input MTJs [38], the MRAM-based implementation enables independent STT writing of the input MTJs by using the access transistors. This brings significant advantages related to scalability and energy consumption [16]. However, the nonzero ON resistance of the access transistors decreases the effective TMR of the 1T/1MTJ cells by about 10% [52]. According to Figure 11.6a, this increases the average error probabilities by a factor of <2. Therefore, MTJs with higher TMR ratios are required for a reliable MRAM-based reprogrammable logic implementation. The effect of the channel resistance of the transistors can be taken into account by using the following equation [53] coupled with (11.10) and (11.11):

$$R_{\text{on}} = \frac{V_{\text{DS}}}{\mu_n C_{\text{ox}} \dfrac{W}{L}\left[\left(V_{\text{GS}} - V_{\text{TH}}\right)V_{\text{DS}} - \dfrac{V_{\text{DS}}^2}{2}\right]}. \tag{11.15}$$

where:

$V_{\text{GS}}(V_{\text{DS}})$ = voltage difference between the gate (drain) and the source of the access transistor

μ_n = mobility of electrons

C_{ox} = gate oxide capacitance per unit area

$W(L)$ = channel width (length)

The transistors are supposed to operate in the triode (ohmic) region ($V_{\text{GS}} > V_{\text{TH}}$ and $V_{\text{DS}} < V_{\text{GS}} - V_{\text{TH}}$).

For the MRAM-based reprogrammable implementation of more complex Boolean logic functions, a sequence of basic logic operations including AND, OR, and so on has to be constructed. As an example, we consider the implementation of the exclusive OR (XOR) function, which is a fundamental logic function in arithmetic circuits.

The output of the XOR function (a_1 XOR a_2) is logic 1 only if one of the inputs is 1 and can be expressed as $a_1 \cdot \overline{a_2} + \overline{a_1} \cdot a_2$ or $(a_1 + a_2) \cdot (\overline{a_1 \cdot a_2})$. It can be shown that the design based on the latter expression requires a minimum of steps (six steps) for implementation using the MRAM arrays as

$$\begin{aligned} \text{Preset: } & b_1 = 1 \\ \text{OR: } & b_1 \leftarrow a_1 + a_2 \\ \text{Preset: } & b_2 = 0 \\ \text{NAND: } & b_2 \leftarrow \overline{a_1 \cdot a_2} \\ \text{Preset: } & a_3 = 1 \\ \text{AND: } & a_3 \leftarrow b_1 \cdot b_2 \equiv a_1 \, \text{XOR} \, a_2 \end{aligned} \tag{11.16}$$

where:

a_i and b_i = logic variables equivalent to the resistance states of the MTJs in Arrays 1 and Array 2, respectively

a_1 and a_2 = input variables stored in two MTJs in Array 1 and the final result (a_3) is written in an MTJ in Array 1

There are two intermediate basic operations on a_1 and a_2 (OR and NAND), the results of which (b_1 and b_2) are stored in two arbitrary MTJs in Array 2 for performing the final basic operation (AND). b_1 and b_2 are the inputs of the final operation and Array 2 (Array 1) acts as the input (output) array.

As the output of one operation can be used as the input data for the next logic stage, complex Boolean logic functions are designed by executing a well-defined set of subsequent basic operations. Furthermore, the MTJs can be selected arbitrarily (two in the input array and one in the output array). The computation framework in the MRAM architecture is flexible and not localized as in the MTJ circuits shown in Figure 11.2. MRAM-based logic extends the functionality of the STT-MRAM architecture to perform nonvolatile logic by eliminating the need for data transfer between separated memory and logic units. This also allows a shift away from the Von Neumann architecture to shorten the interconnection delay.

In order to analyze the reliability of a complex Boolean logic function (f), which is implemented as a sequence of the basic reprogrammable logic operations, we define the average error probability of f as

$$\bar{E}_{\mathrm{f}} = 1 - R(f) = 1 - \prod_{i=1}^{n_{\mathrm{f}}} \left[1 - \bar{E}_{\mathrm{b}}(i)\right], \tag{11.17}$$

where:

$R(f)$ = reliability of f

n_{f} = total number of required basic logic operations for implementing f

$\bar{E}_{\mathrm{b}}(i)$ = average error probability of the ith basic logic operation

By assuming the optimized V_A for each basic operation, we use the minimum $\bar{E}_b(i)$ values (Figure 11.3) for calculating $\bar{E}_f$. For example, by using Equation 11.17 and Figure 11.6a for TMR = 300%, the average error probability of the XOR function, which is designed following the steps shown in Equation 11.16, is obtained as

$$\bar{E}_{\mathrm{XOR}} = 1 - \left(1 - \bar{E}_{\mathrm{OR}}\right)\left(1 - \bar{E}_{\mathrm{NAND}}\right)\left(1 - \bar{E}_{\mathrm{AND}}\right) \simeq 2 \times 10^{-2} \tag{11.18}$$

For the sake of higher reliability, one can design a complex logic function based solely on the AND and NAND operations, as these are more reliable compared to the OR and NOR operation in the reprogrammable implementation. However, the reliability-based design increases the number of required basic logic operations for implementation and, thus, increases the computation time and the energy consumption. For example, the reliability-based design of the XOR function requires the following steps in the reprogrammable MRAM-based logic architecture:

$$\begin{aligned}
&\text{Preset: } b_1 = 1\\
&\text{AND: } b_1 \leftarrow a_1 \cdot a_2\\
&\text{Preset: } a_3 = 0, b_2 = 1\\
&\text{NAND: } a_3 \leftarrow \overline{b_1 \cdot b_2} \equiv \overline{b_1}\\
&\text{Preset: } b_1 = 0\\
&\text{NAND: } b_1 \leftarrow \overline{a_1 \cdot a_3}\\
&\text{Preset: } b_2 = 0\\
&\text{NAND: } b_2 \leftarrow \overline{a_2 \cdot a_3}\\
&\text{Preset: } a_3 = 0\\
&\text{NAND: } a_3 \leftarrow \overline{b_1 \cdot b_2} \equiv a_1 \,\text{XOR}\, a_2
\end{aligned} \tag{11.19}$$

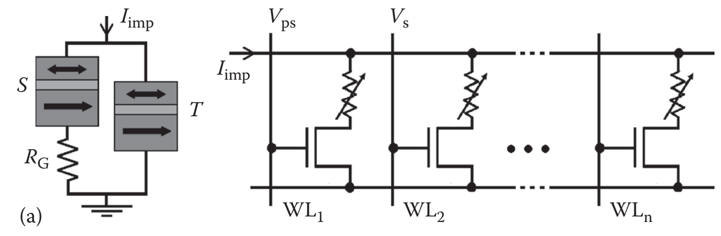
The average error probability of the XOR for this design is about $\simeq 5 \times 10^{-3}$, which is four times smaller than that of the design with minimized steps. However, the number of sequential steps and, therefore, the computation time and the energy consumption are approximately doubled.

11.4.2 MRAM-Based Implication Logic Arrays

In this subsection, we show that by using the access transistors of the 1T/1MTJ cells as voltage-controlled resistors, the CC-IMP gate can be implemented in a STT-MRAM array without the need for extra hardware. Compared to the VC-IMP gate (Figure 11.2c), the MTJ-based CC-IMP gate (Figure 11.2d) provides higher performance [54]. However, its structural asymmetry caused by R_G makes the generalization of the CC-IMP gate to a large-scale implication logic circuit more problematic. In fact, as R_G is connected in series to S, S (T) can be used only as source (target) MTJ for the implication operations, and the logic result stored in T cannot be used as a source input for the next implication operation. Therefore, intermediate read/write operations are required to read the data stored in a target MTJ and to write it to a source MTJ, which increases complexity, energy consumption, and delay.

This problem can be addressed by an innovative solution [54] using two simultaneous voltage pulses on the WLs of a STT-MRAM array with different amplitudes. Figure 11.9a shows the MTJ- and the MRAM-based CC-IMP circuit topologies. In the MRAM array, the structural asymmetry required for the CC-IMP is provided, when the select and preselect voltage signals (V and V_{ps}) are applied to two arbitrary WLs. As $V_{ps} < V_s$, the transistors exhibit different channel resistances and the required structural asymmetry is provided by the preselected transistor showing a higher resistance, which acts as R_G. The logic operation is performed by simultaneously applying the current I_{imp} to the common BL, and V_s and V_{ps} to the WLs of the target and the source 1T/1MTJ cells, respectively. The logic result is stored as the final resistance state of the selected (target) MTJ, which can be used now as a source input by preselect in the next operations.

Figure 11.9b shows the required circuit signals to implement the universal NOR operation ($a_3 \leftarrow a_1$ NOR a_2) in three steps including one TRUE and two NIMP operations, as explained in Equation 11.5. According to Equation 11.17, the reliability of the implication-based NOR is then obtained as $\bar{E}_{NOR} = 1-(1-\bar{E}_{IMP})^2$, which is $\simeq 1.9\times10^{-4}$ for TMR = 300%. In the MRAM-based implication logic framework,



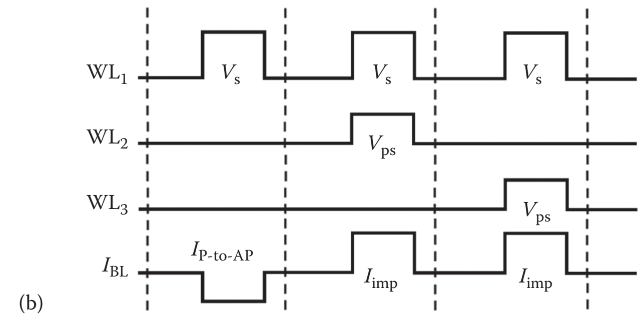


FIGURE 11.9 (a) MTJ- and MRAM-based implication logic architectures. (b) Circuit signals for performing the universal NOR operation in MRAM-based implication logic architecture.

complex logic functions are implemented by using subsequent FALSE (TRUE) and IMP (NIMP) operations, as in each array only one operation can be performed at a time. Regardless of the number of inputs, only two extra memory elements [55] are needed to compute all Boolean logic functions with maximum $n-2$ inputs in an array with n 1T/1MTJ cells.

It is important to note that when the (N)IMP operation is executed, the target data is not available anymore, as the (N)IMP result is written into the target MTJ. Therefore, as long as the data are used only as the source data in the subsequent operations, multiple logic fan-out is not required. However, when the data have to be used after being the target data of an operation, implication-based NOT/COPY operations are executed to keep the data available. As a result, when multiple fan-out is required, a set of FALSE (TRUE) and IMP (NIMP) operations are performed to execute NOT and COPY operations in the implication MRAM array (Figure 11.9a). This allows information to be copied from the source MTJ (which could be the target MTJ of the previous operation) to an arbitrary target MTJ in the array without the need for intermediate sensing. As an example, in the implication logic, the XOR function can be designed as [21]

$$a_1\,\mathrm{XOR}\,a_2 \equiv \left(a_1\,\mathrm{IMP}\,a_2\right)\,\mathrm{IMP}\,\left(a_2\,\mathrm{NIMP}\,a_1\right) \tag{11.20}$$

where, in the MRAM logic architecture, its implementation ($a_3 \leftarrow a_1$ XOR a_2) comprises the following steps:

$$\begin{aligned}
&\text{TRUE: } a_3, a_4 = 1\\
&\text{NIMP: } \overline{a_3 \rightarrow a_1} \equiv \left\{a_3' = a_3 \cdot \overline{a_1} = \overline{a_1}\right\}\\
&\text{NIMP: } \overline{a_4 \rightarrow a_2} \equiv \left\{a_4' = a_4 \cdot \overline{a_2} = \overline{a_2}\right\}\\
&\text{NIMP: } \overline{a_2 \rightarrow a_1} \equiv \left\{a_2' = a_2 \cdot \overline{a_1}\right\}\\
&\text{NIMP: } \overline{a_4 \rightarrow a_3} \equiv \left\{a_4' = a_4 \cdot \overline{a_3} = \overline{a_2} \cdot a_1\right\}\\
&\text{TRUE: } a_1 = 1\\
&\text{NIMP: } \overline{a_1 \rightarrow a_2} \equiv \left\{a_1' = a_1 \cdot \overline{a_2} = 1 \cdot \left(\overline{a_2 \cdot \overline{a_1}}\right) = \overline{a_2} + a_1\right\}\\
&\text{NIMP: } \overline{a_1 \rightarrow a_4} \equiv \left\{a_1' = a_1 \cdot \overline{a_4} = \left(\overline{a_2} + a_1\right) \cdot \left(a_2 + \overline{a_1}\right)\right\}\\
&\text{TRUE: } a_3 = 1\\
&\text{NIMP: } \overline{a_3 \rightarrow a_1} \equiv \left\{a_3' = \overline{a_1} = a_2 \cdot \overline{a_1} + \overline{a_2} \cdot a_1 \equiv a_1\,\mathrm{XOR}\,a_2\right\}
\end{aligned} \tag{11.21}$$

According to Equation 11.17, as the implementation includes seven NIMP operations, the reliability of the implication-based XOR is obtained as $\bar{E}_{\text{XOR}} = 1-(1-\bar{E}_{\text{IMP}})^7$, which is $\simeq 6.5\times10^{-4}$. This is about one order of magnitude smaller than that of the most reliable design with the reprogrammable architecture for the same MTJ device characteristics, although both implementations include 11 operations.

11.4.3 Performance Comparisons

In order to perform a fair comparison between the reprogrammable and the implication logic architectures, we assume the same device (MTJ and transistor) characteristics for both implementations and optimize the circuit parameters with respect to their minimum error probabilities of the basic logic operations. We calculate the average error probabilities of the same Boolean logic functions using the reliability analysis method explained before and also the energy consumptions of these functions by using the MTJ SPICE model presented in [47].

Figure 11.10a shows the energy consumptions of implication- and reprogrammable-based implementations of some basic Boolean logic operations. Due to the mismatch between the intrinsic logic functions of the gates, the implication-based implementation requires more steps and thus more energy (by an average factor of ~1.4) to implement the basic operations. However, even for the intrinsic functions of the reprogrammable gate, the implication logic exhibits about one to three orders of magnitude higher reliability than the reprogrammable logic architecture (Figure 11.10b). In order to see the performance at larger circuits, the energy consumptions and the average error probabilities of more complex Boolean functions including an XOR, a half adder, and a full adder are compared in Figure 11.11a and b. We have used only AND and NAND operations to provide the most reliable design for the reprogrammable architecture. Nevertheless, Figure 11.11b shows that the implication-based implementation of more complex functions exhibits about two orders of magnitude higher reliability than the most reliable design with the

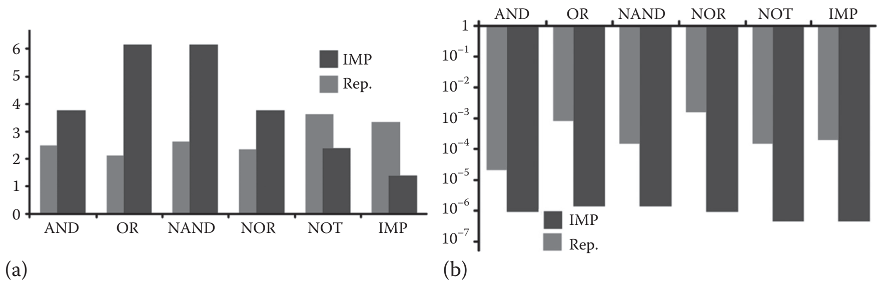


FIGURE 11.10 (a) Normalized energy consumption and (b) minimum average error probabilities plotted for MRAM-based implication (IMP) and reprogrammable (Rep.) implementations of some basic Boolean logic operations. The energy is normalized by the amount of energy required for MTJ AP-to-P switching, which is equal to 18 pJ for pulse duration of $t = 50$ ns in our simulations.

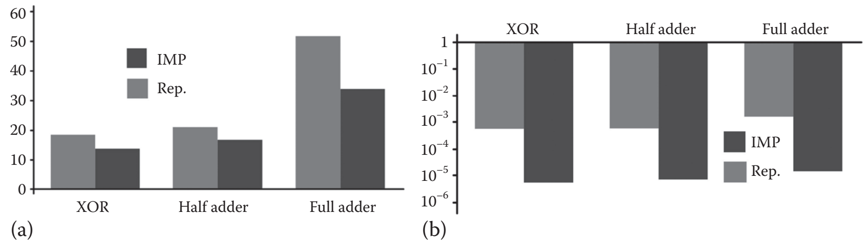


FIGURE 11.11 (a) Normalized energy consumption and (b) minimum average error probabilities plotted for XOR, half adder, and full adder logic functions. For the sake of reliability, reprogrammable implementations are designed based on AND and NAND operations.

reprogrammable architecture. Furthermore, for complex logic functions that are not inherently covered by the gates, the implication logic architecture performs better also with respect to power consumption (Figure 11.11a). In combination with 0 and ‘1 writing operations, both reprogrammable-based AND–NAND and implication-based IMP-NIMP logic functions form complete logic bases. Thus, any Boolean logic function can be computed in a series of subsequent steps using these architectures. We believe combining implication and (N)AND-based reprogrammable frameworks in an MRAM arrays is a possible direction in designing large-scalable MTJ-based logic circuits featuring a minimized number of logic steps and optimized error, delay, and power consumption.

11.4.4 Parallel MRAM-Based Computation

Parallelization of several MRAM arrays can be used to perform simultaneous operations on the same WLs to decrease the number of required serial steps. For example, the required steps to implement the XOR function presented in Equation 11.21 can be performed in parallel in the MRAM structure shown in Figure 11.12. By applying the single/dual mode voltage signals to the WLs (Figure 11.9b) to execute TRUE/NIMP (FALSE/IMP) operations, the corresponding MTJs are selected or preselected in all arrays. Therefore, by applying relevant current signals ($I_{\text{P-to-AP}}$ or I_{imp}) simultaneously to all current-carrying lines (SLs and BLs), the computations are performed in parallel. This significantly reduces the total time needed for implementing XOR functions on binary data (a_i and a_j) stored in the ith and jth MTJs of each array.

For more complicated applications in which intermediate results have to be used as the input of next logic steps (e.g., n-bit full adders, where $n > 1$), only some parts of the computations can be performed in parallel. As an example, we consider the MRAM-based implementation of a full adder, which is a basic element of arithmetic circuits. As is well known, it adds three binary inputs (a_1, b_1, and c_{in}) and produces two binary outputs: sum ($s = a_1$ XOR b_1 XOR c_{in}) and carry ($c_{\text{out}} = [a_1$ AND $b_1]$ OR $[c_{\text{in}}$ AND $\{a_1$ XOR $b_1\}]$). Figure 11.13 shows a two-bit full adder in which the carry output from the first full adder ($c_{\text{out}-1}$) is connected to the carry input of the second full adder ($c_{\text{in}-2}$). Therefore, it is not possible to perform all computations in parallel. We assume that a_1 (b_1) and a_2 (b_2) are stored in the first and second MTJs in

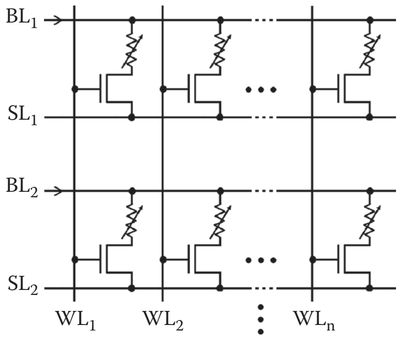


FIGURE 11.12 Coupled MRAM arrays based on the common STT-MRAM architecture suited for parallel MRAM-based computations. (From Hosomi, M., Yamagishi, H., Yamamoto, T., Bessho, K., Higo, Y., Yamane, K., Yamada, H. et al., *IEEE International Electron Devices Meeting (IEDM) Technical Digest*, pp. 459–462. 5–5 December, Washington, DC, 2005.)

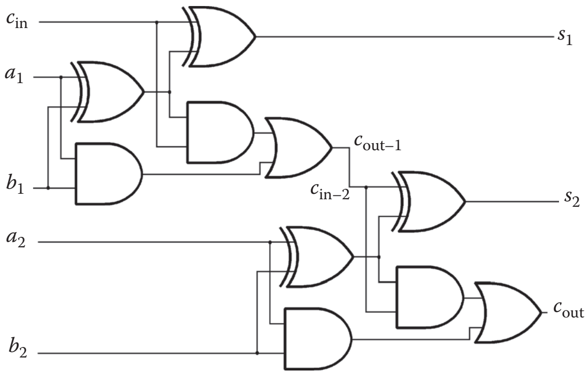


FIGURE 11.13 Logic diagram of a two-bit full adder.

the MRAM Array 1 (2). The XOR functions between a_1 and a_2 (b_1 and b_2) can be performed in parallel, as explained before. Afterward, one must continue the calculations for c_{out-1} (c_{in-2}). This part of the calculations is performed only in Array 1 and cannot be parallelized. Then, a read/write operation is required to read out c_{out-1} and write into an MTJ in Array 2, which is in the same WL as the MTJ that holds c_{in} in Array 1. After that, the XOR functions are performed in parallel to calculate s_1 and s_2. Finally, the calculations are continued to compute c_{out} in Array 2. As a result, by parallelization of the XOR functions, the total calculation time required for the MRAM-based implementation of a two-bit full adder is decreased by about 40%. The same method can be employed to improve the run-time performance of the MRAM-based reprogrammable logic arrays.

In contrast to this computation scheme, the Von Neumann architecture consists of physically separated memory and logic units. Here, the computation requires continuous data transfer between these units over one global bus. This communication increases the delay and limits the performance of the computing system. Since MRAM-based computing systems merge logic and memory, the necessary communication between separate units is largely decreased. It also features a simple circuit structure and delocalizes computational execution.

11.5 CONCLUSION

We have described MTJ-based nonvolatile logic circuits capable of stateful logic operations. In these circuits, MTJs serve simultaneously as memory and main computing elements (logic gates). Unlike the memristive stateful gates, the MTJ logic devices do not show error accumulation and exhibit almost unlimited endurance. A reliability analysis of MTJ-based logic operations was presented and it has been shown that the implication logic architecture, an up-to-now ignored Boolean logic based on material implication, significantly improves the reliability of the MTJ-based logic compared to the reprogrammable logic architectures based on the common Boolean logic (N)AND and (N)OR operations.

Because of the easy integration with CMOS, the MTJ-based logic gates are generalizable to large-scale nonvolatile circuits based on 1T/1MTJ STT-MRAM memory arrays. The presented MRAM-based computing system is computationally complete, has a simple circuit structure (STT-MRAM), delocalizes computational execution and eliminates the need for intermediate circuitry compared to most nonvolatile logic-in-memory architectures previously proposed. It also enables parallel nonvolatile computations and, therefore, it is suited for large-scale integration of complex logic functions and opens an alternative path toward zero-standby-power systems, shifting away from the Von Neumann architecture by eliminating the need for data transfer between separate memory and logic units.

ACKNOWLEDGMENT

This work was supported by the European Research Council through the grant #247056 MOSILSPIN.

REFERENCES

1. International Technology Roadmap for Semiconductor, Chapter PIDS, 2011. Available at: http://www.itrs.net/.
2. V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, Limits to binary logic switch scaling—A Gedanken model, *Proc. IEEE*, 91:1934–1939, 2003.
3. N. S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan, Leakage current: Moore's law meets the static power, *Computer*, 36(12):68–75, 2003.
4. J. A. Hutchby, G. I. Bourianoff, V. V. Zhirnov, and J. E. Brewer, Extending the road beyond CMOS, *IEEE Circ. Dev. Mag.*, 18:28–41, 2002.
5. A. Ney, C. Pampuch, R. Koch, and K. H. Ploog, Programmable computing with a single magnetoresistive element, *Nature*, 425:485–487, 2003.
6. J. G. Wang, H. Meng, and J. P. Wang, Programmable spintronics logic device based on a magnetic tunnel junction element, *J. Appl. Phys.*, 97:10D509, 2005.

7. B. Behin-Aein, D. Datta, S. Salahuddin, and S. Datta, Proposal for an all-spin logic device with built-in memory, *Nat. Nanotechnol.*, 5(4):266–270, 2010.
8. A. C. Seabaugh and Q. Zhang, Low-voltage tunnel transistors for beyond-CMOS logic, *Proc. IEEE*, 98:2095–2110, 2010.
9. B. Dellabetta and M. J. Gilbert, Performance characteristics of strongly correlated bilayer graphene for post-CMOS logic devices, in *Proceedings of Silicon Nanoelectronics Workshop*, pp. 1–2. 13–14 June, Honolulu, HI, 2010.
10. D. Bouvet, L. Forró, A. M. Ionescu, Y. Leblebici, A. Magrez, K. E. Moselund, G. A. Salvatore, N. Setter, and I. Stolitchnov, Materials and devices for nanoelectronic systems beyond ultimately scaled CMOS, in *Nanosystems Design and Technology*, pp. 23–44. Springer, 2009.
11. D. E. Nikonov and I. A. Young, Overview of beyond-CMOS devices and a uniform methodology for their benchmarking, *Proc. IEEE*, 101:2498–2533, 2013.
12. S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, T. Endoh, H. Ohno, and T. Hanyu, MTJ-based nonvolatile logic-in-memory circuit, future prospects and issues, in *Proceedings of Design Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 433–435. 20–24 April, Nice, 2009.
13. B. N. Engel, J. Akerman, B. Butcher, R. W. Dave, M. DeHerrera, M. Durlam, G. Grynkewich, et al., A 4-Mb toggle MRAM based on a novel bit and switching method, *IEEE Trans. Magn.*, 41(1):132–136, 2005.
14. J. C. Slonczewski, Current-driven excitation of magnetic multilayers, *J. Magn. Magn. Mater.*, 159:L1–L7, 1996.
15. L. Berger, Emission of spin waves by a magnetic multilayer traversed by a current, *Phys. Rev. B Condens. Matter*, 54:9353–9358, 1996.
16. C. Chappert, A. Fert, and F. N. V. Dau, The emergence of spin electronics in data storage, *Nat. Mater.*, 6:813–823, 2007.
17. K. L. Wang, J. G. Alzate, and P. K. Amiri, Low-power non-volatile spintronic memory: STT-RAM and beyond, *J. Phys. D Appl. Phys.*, 46:074003, 2013.
18. C. Augustine, N. Mojumder, X. Fong, H. Choday, S. P. Park, and K. Roy, STT-MRAMs for future universal memories: Perspective and prospective, in *Proceedings of the 28th International Conference on Microelectronics (MIEL)*, pp. 349–355. 13–16 May, Serbia, 2012.
19. W. Zhao, E. Belhaire, C. Chappert, F. Jacquet, and P. Mazoyer, New non-volatile logic based on spin-MTJ, *Phys. Status Solidi A*, 205:1373–1377, 2008.
20. M. Natsui, D. Suzuki, N. Sakimura, R. Nebashi, Y. Tsuji, A. Morioka, T. Sugibayashi, et al., Nonvolatile logic-in-memory array processor in 90 nm MTJ/MOS achieving 75% leakage reduction using cycle-based power gating, in *Proceedings of the International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 194–195. 17–21 February, San Francisco, CA, 2013.
21. J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, Memristive switches enable stateful logic operations via material implication, *Nature*, 464(7290):873–876, 2010.
22. A. Whitehead and B. Russell, *Principia Mathematica*, Cambridge University Press, New York, 1910.
23. C. E. Shannon, A symbolic analysis of relay and switching circuits, master's thesis, Massachusetts Institute of Technology (MIT), 1940.
24. L. Chua, Memristor–The missing circuit element, *IEEE Trans. Circ. Theor.*, 18(5):507–519, 1971.
25. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, The missing memristor found, *Nature*, 453(7191):80–83, 2008.
26. X. Wang, Y. Chen, H. Xi, H. Li, and D. Dimitrov, Spintronic memristor through spin torque induced magnetization motion, *IEEE Electr. Dev. Lett.*, 30(3):294–297, 2009.

27. G. Tatara and H. Kohno, Theory of current-driven domain wall motion: Spin transfer versus momentum transfer, *Phys. Rev. Lett.*, 92:086601, 2004.
28. Y. V. Pershin, S. L. Fontaine, and M. D. Ventra, Memristive model of amoeba learning, *Phys. Rev. E*, 80(2):021926, 2009.
29. S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, *Nano Lett.*, 10(4):1297–1301, 2010.
30. Y. N. Joglekar and S. J. Wolf, The elusive memristor: Properties of basic electrical circuits, *Eur. J. Phys.*, 30(4):661–675, 2009.
31. Y. V. Pershin and M. D. Ventra, Practical approach to programmable analog circuits with memristors, *IEEE Trans. Circ. Syst. I*, 57(8):1857–1864, 2010.
32. H. Mahmoudi, V. Sverdlov, and S. Selberherr, Influence of geometry on the memristive behavior of domain wall spintronic memristors and its applications for measurement, *J. Supercond. Nov. Magn.*, 26:1745–1749, 2013.
33. M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, Switching dynamics in titanium dioxide memristive devices, *J. Appl. Phys.*, 106:074508, 2009.
34. S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, TEAM: ThrEshold Adaptive Memristor Model, *IEEE Trans. Circ. Syst. I*, 60:211–221, 2013.
35. H. Mahmoudi, V. Sverdlov, and S. Selberherr, State drift optimization of memristive stateful IMP logic gates, in *Proceedings of the 15th International Workshop on Computational Electronics (IWCE)*, pp. 243–244. 22–25 May, Madison, WI, 2012.
36. S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, Memristor-based IMPLY logic design procedure, in *Proceedings of IEEE International Conference on Computer Design*, pp. 142–147. 9–12 October, Amherst, MA, 2011.
37. A. Lyle, J. Harms, S. Patil, X. Yao, D. Lilja, and J. P. Wang, Direct communication between magnetic tunnel junctions for nonvolatile logic fan-out architecture, *Appl. Phys. Lett.*, 97:152504, 2010.
38. A. Lyle, S. Patil, J. Harms, B. Glass, X. Yao, D. Lilja, and J. P. Wang, Magnetic tunnel junction logic architecture for realization of simultaneous computation and communication, *IEEE Trans. Magn.*, 47:2970–2973, 2011.
39. H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, Implication logic gates using spin-transfer-torque-operated magnetic tunnel junctions for intrinsic logic-in-memory, *Solid State Electron.*, 84:191–197, 2013.
40. S. S. P. Parkin, C. Kaiser, A. Panchula, P. M. Rice, B. Hughes, M. Samant, and S.-H. Yang, Giant tunnelling magnetoresistance at room temperature with MgO (100) tunnel barriers, *Nat. Mater.*, 3:862–867, 2004.
41. E. Chen, D. Apalkov, Z. Diao, A. Driskill-Smith, D. Druist, D. Lottis, V. Nikitin, et al., Advances and future prospects of spin-transfer torque random access memory, *IEEE Trans. Magn.*, 46:1873–1878, 2010.
42. J. J. Yang, M.-X. Zhang, J. P. Strachan, F. Miao, M. D. Pickett, R. D. Kelley, G. Medeiros-Ribeiro, and R. S. Williams, High switching endurance in TaO_x memristive devices, *Appl. Phys. Lett.*, 97:232102, 2010.
43. J. J. Yang, D. B. Strukov, and D. R. Stewart, Memristive devices for computing, *Nat. Nanotechnol.*, 8:13–24, 2013.
44. H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, Reliability analysis and comparison of implication and reprogrammable logic gates in magnetic tunnel junction logic circuits, *IEEE Trans. Magn.*, 49:5620–5628, 2013.
45. Y. Higo, K. Yamane, K. Ohba, H. Narisawa, K. Bessho, M. Hosomi, and H. Kano, Thermal activation effect on spin transfer switching in magnetic tunnel junctions, *Appl. Phys. Lett.*, 87:082502, 2005.

46. M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, et al., A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM, in *IEEE International Electron Devices Meeting (IEDM) Technical Digest*, pp. 459–462. 5–5 December, Washington, DC, 2005.
47. J. D. Harms, F. Ebrahimi, X. F. Yao, and J. P. Wang, SPICE macromodel of spin-torque-transfer-operated magnetic tunnel junctions, *IEEE Trans. Electron Dev.*, 57(6):1425–1430, 2010.
48. 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 Trans. Electron Dev.*, 59:819–826, 2012.
49. Y. Huai, Spin-transfer torque MRAM (STT-MRAM): Challenges and prospects, *AAPPS Bull.*, 18:33–40, 2008.
50. H. Kronmüller, General micromagnetic theory, in *Handbook of Magnetism and Advanced Magnetic Materials*, Wiley, Chichester, 2007.
51. H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, Study on reliability-based optimization of spin-transfer torque magnetic tunnel junction implication logic gates, *Adv. Mat. Res.*, 854:89–95, 2013.
52. R. Beach, T. Min, C. Horng, Q. Chen, P. Sherman, S. Le, S. Young, et al., A statistical study of magnetic tunnel junctions for high-density spin torque transfer-MRAM (STT-MRAM), in *IEEE International Electron Devices Meeting (IEDM) Technical Digest*, pp. 306–308. 15–17 December, San Francisco, CA, 2008.
53. B. Razavi, *Fundamentals of Microelectronics*, vol. 1. Wiley, Hoboken, NJ, 2009.
54. H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, MRAM-based logic array for large-scale non-volatile logic-in-memory applications, in *Proceedings of the 2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 26–27. 15–17 July, Brooklyn, NY, 2013.
55. E. Lehtonen, J. H. Poikonen, and M. Laiho, Two memristors suffice to compute all Boolean functions, *Electron. Lett.*, 46(3):239–240, 2010.