Majority-based Synthesis for Nanotechnologies

Luca Amarú, Pierre-Emmanuel Gaillardon, Giovanni De Micheli
Integrated Systems Laboratory (LSI), EPFL, Switzerland

***Abstract*—We study the logic synthesis of emerging nanotechnologies whose elementary devices abstraction is a majority voter. We argue that synthesis tools, natively supporting the majority logic abstraction, are the technology enablers. This is because they allow designers to validate majority-based nanotechnologies on large-scale benchmarks. We describe models and data-structures for logic design with majority-based nanotechnologies and we show results of applying new synthesis algorithms and tools. We conclude that new logic synthesis methods are required to achieve a fair assessment on emerging nanotechnologies.**

I. Introduction

The arrival of post-CMOS nanotechnologies has brought new devices whose logic models are different from traditional transistors. In this scenario, new logic abstractions and synthesis techniques are key to exploit at best the enhanced functionality of emerging nanodevices.

In this paper, we focus on the synthesis of new digital technologies whose elementary device abstraction is a majority voter. Specific examples for these nanotechnologies include, but are not limited to, silicon nanowire [1], [2], graphene [3], [4], resistive random-access memory [5], [6], spin-wave devices [7], [8], quantum-dot cellular automata [9], nanomagnets [10], DNA-logic [11], and many others [12]–[14]. We revisit logic synthesis in light of its enabling role in the selection of majority-based nanotechnologies. We first survey a selected pool of nanodevices behaving as majority voters. Next, we focus on logic synthesis, where we present novel data structures and optimization techniques based on majority logic. Experimental results, over two representative nanotechnologies, show that majority logic synthesis not only generates better circuits than standard synthesis tools but also enables a larger gain versus CMOS. This serves as an empirical demonstration of potentially advantageous nanotechnologies that cannot emerge without the leading support of logic synthesis.

The remainder of this paper is organized as follows. Section II surveys some promising post-CMOS devices behaving as majority voters. Section III describes synthesis models for majority logic natively fitting the functionality of the devices presented in Section II. Section IV shows experimental results for some of the nanotechnologies introduced in Section II using the logic synthesis techniques presented in Section III. Section V is a conclusion.

II. Majority-based Nanodevices

In this section, we review some promising nanodevices in post-CMOS technologies. We focus on resistive RAM and spin-wave nanotechnologies. Regardless of using charge-based or non-charge-based physical phenomena to carry digital information, all considered devices inherently implement a majority voter. The majority functionality is a key asset for these nanodevices in addition to their intrinsic performances.

A. Spin-Wave Devices

Spin Wave Devices (SWDs) are promising beyond-CMOS candidates. Unlike charge-based technologies, SWDs use spin as information carrier that propagates in waves [7], [8]. This physical mechanism enables ultra-low power operation, two orders of magnitude lower than current CMOS [7], [8].

Besides the extremely low-power consumption of SWD logic, which is a key technological asset, the use of wave computation in digital circuits can enhance its logic expressive power. SWD logic computation is based on the interference of spin waves. Depending on the phase of the propagating spin waves/signals, their interference is constructive or destructive. The final interference result is translated to the output via *Magneto-Electric* (ME) cells. In this scenario, an inverter is simply a waveguide with length equal to 1.5× of the spin wavelength (λ_{SW}). In this way, the information encoded in the phase of the SW signal arrives inverted to the output ME cell, Fig. 1(a). The actual logic primitive in SWD technology is the majority voter, which is implemented by the symmetric merging of three waveguides Fig. 1(b). Here, the length of each waveguide is 1.0× the spin wavelength. In the majority voter structure, the spin wave signal at the output is determined by the majority phase of the input spin waves.

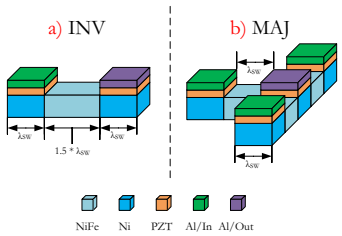

Fig. 1: Primitive gate areas and designs for SWD technology. All distances are parameterized with the spin wave wavelength λ_{SW} [16].

B. Resistive RAM

The *Resistive RAM* (RRAM) technology is receiving widespread research attention as candidate for high-density and low-cost storage. RRAM store information as an internal resistive state, which can be either a *Low Resistance State* (LRS) or a *High Resistance State* (HRS) [5], [6]. Their internal resistance state of the device, Z_n, can be modified by applying a positive or a negative voltage. This is a function of the previous resistance state Z, the voltage at the terminal P and the voltage at the terminal Q. The functionality of RRAM can be summarized by a state machine [5], [6], as shown in Fig. 2. Further details can be found in [6]. Transition occurs only for

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

the conditions $P = 0, Q = 1$, i.e., $V_{PQ} < 0$ so $Z \to 0$ and $P = 1, Q = 0$, i.e., $V_{PQ} > 0$ so $Z \to 1$. Thus, it is possible

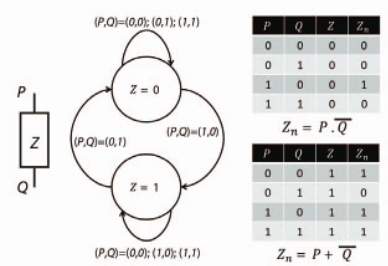

Fig. 2: Resistive majority operation with RRAM devices [15].

to express Z_n as the following:

$$\begin{aligned} Z_n &= (P.\overline{Q}).\overline{Z} + (P + \overline{Q}).Z \\ &= P.Z + \overline{Q}.Z + P.\overline{Q}.\overline{Z} \\ &= P.Z + \overline{Q}.Z + P.\overline{Q}.Z + P.\overline{Q}.\overline{Z} \\ &= P.Z + \overline{Q}.Z + P.\overline{Q} \\ &= M_3(P, \overline{Q}, Z) \end{aligned}$$

where M_3 is the majority Boolean function with 3 inputs.

C. Logic Abstraction and Model for Post-CMOS Devices

The aforementioned nanodevices inherently realize majority gates as primitive building block for computation. Note that majority voting is the basic computational brick also for nanotechnologies other than the ones surveyed in this paper. For example, DNA-logic [11], graphene reconfigurable gates [17], quantum-dot cellular automata [9], etc., are promising majority-based nanotechnologies.

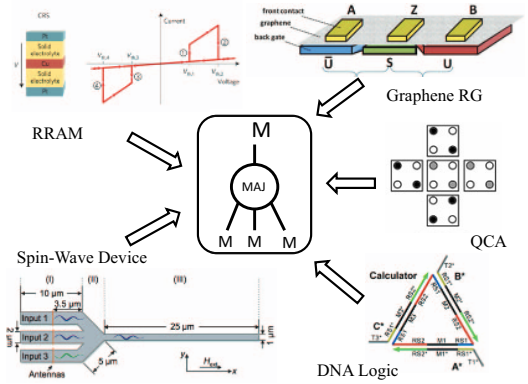

Fig. 3: Common logic abstraction for SWD, RRAM, Graphene reconfigurable gates, QCA and DNA logic.

The far reaching consequence of embedding majority operators as atomic functions in nanotechnologies is the ability to implement arithmetic logic with few physical resources.

III. Majority Logic Synthesis

In this section, we review data structures and optimization algorithms based on majority logic. We describe *Majority-Inverter-Graph* (MIG) [18]–[20] as a logic representation form and we define optimization routines running on it.

A. Majority-Inverter Graph

A MIG is a data structure for Boolean function representation and optimization. An MIG is defined as a logic network that consists of 3-input majority nodes and regular/complemented edges [18]–[20]. This model can be extended to wider majority operators but we consider here only 3-input majority nodes for the sake of simplicity.

MIGs can efficiently represent Boolean functions thanks to the expressive power of the majority operator. Indeed, a majority operator can be configured to behave as a traditional conjunction (AND) or disjunction (OR) operator. In the case of 3-input majority operator, fixing one input to 0 realizes an AND while fixing one input to 1 realizes an OR. As a consequence of the AND/OR inclusion by MAJ, traditional *AND/OR/INV Graphs* (AOIGs) are a special case of MIGs and MIGs can be easily derived from AOIGs. Two examples of MIG representations derived from their optimal AOIGs are depicted by Fig. 4. AND/OR operators are replaced node-wise by MAJ-3 operators with a constant input.

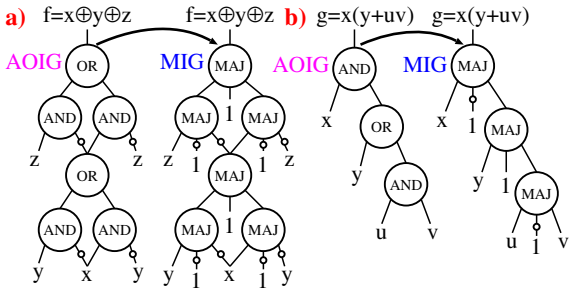

Fig. 4: Examples of MIG representations (right) for (a) $f = x \oplus y \oplus z$ and (b) $g = x(y + uv)$ derived by transposing their optimal AOIG representations (left). Complement attributes are represented by bubbles on the edges.

Intuitively, MIGs are at least as compact as AOIGs. However, even smaller MIG representation arise when fully exploiting the majority functionality, i.e., with non-constant inputs [20]. Later, we will show two smaller and lower depth MIGs for the two examples in Fig. 4.

B. Majority-Inverter Graph Manipulation and Optimization

We are interested in compact MIG representations because they translate in smaller and faster physical implementations. In order to manipulate MIGs and reach advantageous MIG representations, a dedicated Boolean algebra was introduced in [18]. The axiomatic system for the MIG Boolean algebra, referred to as Ω, is defined by the five following primitive transformation rules.

$$\Omega \begin{cases} \textbf{Commutativity} - \boldsymbol{\Omega.C} \\ M(x,y,z) = M(y,x,z) = M(z,y,x) \\ \textbf{Majority} - \boldsymbol{\Omega.M} \\ \begin{cases} \text{if}(x = y)\text{: } M(x,y,z) = x = y \\ \text{if}(x = y')\text{: } M(x,y,z) = z \end{cases} \\ \textbf{Associativity} - \boldsymbol{\Omega.A} \\ M(x,u,M(y,u,z)) = M(z,u,M(y,u,x)) \\ \textbf{Distributivity} - \boldsymbol{\Omega.D} \\ M(x,y,M(u,v,z)) = M(M(x,y,u),M(x,y,v),z) \\ \textbf{Inverter Propagation} - \boldsymbol{\Omega.I} \\ M'(x,y,z) = M(x',y',z') \end{cases} \quad (1)$$

Some of these axioms are inspired by median algebra and others by the properties of the median operator in a distributive lattice. A strong property of MIGs and their algebraic framework is about reachability. It has been proven that, by using a sequence of transformations drawn from Ω, it is possible to traverse the entire MIG representation space [20]. In other words, given any two equivalent MIG representations, it is possible to transform one into the other by just using axioms in Ω. This results is of paramount interest to logic synthesis because it guarantees that the best MIG, for a given target metric, can always be reached. Unfortunately, deriving a sequence of Ω transformations is an intractable problem. As for traditional logic optimization, heuristic techniques provide here fast solutions with reasonable quality [22]. An efficient depth optimization heuristic for MIGs consists of local Ω rules iterated over the critical path [18]. Here, the rationale driving local transformations is to push up variables with largest depth.

As previously anticipated, by using the MIG algebraic framework it is possible to obtain better MIGs for the examples in Fig. 4. Fig. 5 shows the new MIG structures, which are

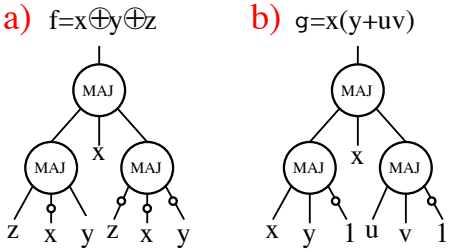

Fig. 5: Optimized MIGs for (a) $f = x \oplus y \oplus z$ and (b) $g = x(y + uv)$.

optimized in both depth (number of levels) and size (number of nodes). These MIGs can be reached using a sequence of Ω axioms starting from their unoptimized structures. We refer the reader to paper [20] for an in-depth discussion on MIG optimization recipes.

IV. Majority-based Design for Nanotechnologies

In this section, we experiment the efficacy of MIGs in the synthesis of emerging nanodevices. We target two different emerging technologies taken from the devices surveyed in Section II. Note that many other nanodevices may benefit from the presented majority synthesis methodologies [12]–[14].

A. SWD-MIG

In SWDs, the logic primitive is a majority voter. In this scenario, we use the MIG data structure to represent and synthesize SWD circuits. The intrinsic correspondence between

TABLE I: Cost Functions for MIGs Mapped onto SWDs

MIG Element	SWD Gate	Area Cost	Delay Cost
Majority node	Majority Gate	4	1
Complemented edge	Inverter Gate	1	1

MIG elements and SWDs makes MIG optimization naturally extendable to obtain minimum cost SWD implementations. For this purpose, *ad hoc* cost functions are assigned to MIG elements during optimization as per Table I. These cost functions are derived from the SWD technology implementation of majority and inverter gates [16].

For the sake of clarity, we comment on our proposed MIG-based SWD synthesis flow by means of an example. The logic function in this example is $g = x \cdot (y + u \cdot v)$. This function is initially represented by the MIG in Fig. 4(b), which has a SWD delay cost of 4 and an SWD area cost of 14. By using Ω transformations, it is possible to reach the optimized MIG depicted by Fig. 5(b). Such an optimized MIG counts the same number of nodes and complemented edges of the original one but one fewer level of depth. In this way, the associated area cost remains 14 but the delay is reduced to 3. After the optimization, each MIG element is mapped onto its corresponding SWD gate. Fig. 6 depicts the SWD mapping for the original (a) and optimized (b) MIGs.

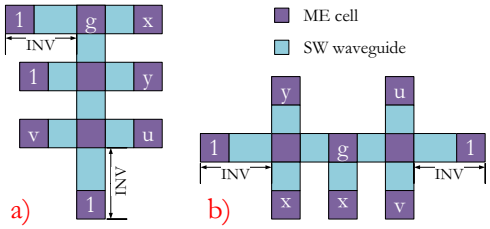

Fig. 6: SWD circuits implementing function $g = x \cdot (y + u \cdot v)$. (a) Standard circuit (b) MIG optimized circuit. ME stands for Magneto-Electric cell and SW for Spin Wave waveguide.

The circuit in Fig. 6(b) features roughly the same area occupation as the one in Fig. 6(a) but shorter input-output path. As per the cost functions in Table I, the achieved speed-up is roughly 25%. Including the physical models in [16], the refined speed-up becomes 18.2%.

We validate hereafter the efficiency of our proposed MIG-based SWD synthesis flow for larger circuits [16]. We also provide a comparison reference to 10-nm CMOS technology.

In MIG-based SWD synthesis, we employed the *MIGhty* MIG optimizer [18]. As traditional-synthesis counterpart, we employed ABC tool [21] with optimization commands *resyn2* and produced in output an *AND-Inverter Graph* (AIG). The AIGs mapping procedure onto SWDs is in common with MIGs: AND nodes are simply mapped to MAJ gates with one input biased to logic 0. For advanced CMOS, we used a commercial synthesis tool fed with a standard-cell library produced by a 10-nm CMOS process flow. The circuit benchmarks are taken from the MCNC suite.

TABLE II: Summarizing performance results of SWD and CMOS Technologies

Technology	ADP Product (a.u.)	Gain vs CMOS	Gain vs AIG
CMOS	9707.06	-	-
SWD - AIG	526.25	18.45 ×	-
SWD - MIG	432.59	22.44 ×	1.22×

Table II summarizes the performance of the benchmarks in the *Area-Delay-Power* (ADP) product to better point out the significant improvement MIG synthesis brings to light. SWD circuits synthesized via MIGs have 1.30× smaller ADP than SWD circuits synthesized via traditional AIGs. This is thanks to the SWD delay improvement enabled by MIGs. As compared to the 10-nm CMOS technology, SWD circuits synthesized by MIGs have 17.02× smaller ADP, offering an

ultra-low power, compact SWD implementation with reduced penalty in delay.

B. RRAM-MIG

In Section II, we have seen that RRAM elementary devices can realize a majority voting operation in addition to storing the result. Therefore, the MIG data structure is the native logic abstraction for RRAM in-memory computation.

To demonstrate the efficacy of the RRAM-MIG coupling, we map a lightweight cryptography block cipher [23] on a RRAM array using MIG-based design techniques [15]. The target cryptography block cipher is PRESENT, originally introduced in [23]. The main operations in the PRESENT encryption algorithm are *addRoundKey*, *sBoxLayer*, *pLayer*, and *KeyUpdate* [23].

For the sake of brevity, we give here details only on the *sBoxLayer* operation. We refer the interested reader to [23] for details on the other operations. The *sBoxLayer* operates on 4-bit long strings. Each string is processed individually by a 4-input, 4-output combinational Boolean function, called operator S. In order to map S into the RRAM memory array, we use MIG representation and optimization. The optimization goal is to reduce the number of majority operations.

The S operator is nothing but a Boolean function with primary inputs pi_0, pi_1, pi_2, pi_3 and primary outputs po_0, po_1, po_2, po_3. After optimization, the MIG for such function counts 11 nodes and 4 levels. Each majority node is mapped into a set of primitive RRAM memory/computing instructions. The S operator requires a total of 38 cycles for its operation in the RRAM array.

Using an analogous MIG-mapping approach, all the PRESENT encryption operations can be performed directly on the RRAM array.

The overall performance of the MIG-based PRESENT implementation on the RRAM array has been estimated considering a RRAM technology aligned with the ITRS 2013 predictions. More precisely, we assume a write time of 1 ns and a write energy of 0.1 fJ/bit. Table III summarizes the number of M_3 instructions and *Read/Write* (R/W) cycles required by the different operations of the PRESENT cipher.

TABLE III: Experimental Results for RRAM-MIG Synthesis PRESENT Implementation Performances

Operation	Instructions (#M 3)	Cycles (#R/W)
Key copy	80	720
Cipher copy	64	576
AddRoundKey	448	4032
sBoxLayer	608	5472
pLayer	64	576
KeyUpdate	760	6840
	Instructions	Cycles
PRESENT Block	58 872	455 184
	Energy	Throughput
	(pJ)	(kbps)
PRESENT Block	5.88	120.7

The total number of primitive majority instructions for the encryption of a 64-bit cipher text is 58872 [15]. The total energy required for one block encryption operation is 5.88 pJ. The total throughput reachable by the system is 120.7 kbps, making it competitive to silicon implementations [23].

V. Conclusions

In this paper, we investigated the relation between logic synthesis and emerging nanotechnologies. We reviewed a class of promising nanodevices whose logic abstraction is a majority voter. We demonstrated that new logic synthesis techniques, natively supporting this abstractions, are the technology enablers as they allow designers to validate nanotechnologies on large-scale benchmarks. New research in logic synthesis is becoming essential to permit a fair evaluation on nanotechnologies with different logic abstractions than standard CMOS.

Acknowledgment

This research was partially supported by ERC-2009-AdG-246810 and SNSF-200021-146600.

References

[1] M. De Marchi *et al.*, *Polarity control in Double-Gate, Gate-All-Around Vertically Stacked Silicon Nanowire FETs*, Proc. IEDM, 2012.
[2] O. Turkyilmaz, *et al.*, *Self-checking ripple-carry adder with ambipolar silicon nanowire FET*, IEEE International Symposium on Circuits and Systems (ISCAS), 2013.
[3] Heejun Yang *et al.*, *Graphene Barristor, a Triode Device with a Gate-Controlled Schottky Barrier*, Science 336, 1140 (2012).
[4] F. Schwierz, *Graphene transistors*, Nature nanotechnology, 2010, 5.7: 487-496.
[5] E. Linn, R. Rosezin, C. Kügeler, R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Materials*, 9, 2010.
[6] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, R. Waser, "Beyond von Neumann–logic operations in passive crossbar arrays alongside memory operations," *Nanotechnology*, 23(305205), 2012.
[7] T. Schneider, *et al.*, *Realization of spin-wave logic gates*, Applied Physics Letters 92.2 (2008): 022505.
[8] Khitun, Alexander, and Kang L. Wang. "Nano scale computational architectures with Spin Wave Bus." Superlattices and Microstructures 38.3 (2005): 184-200.
[9] I. Amlani, et al. *Digital logic gate using quantum-dot cellular automata*, Science 284.5412 (1999): 289-291.
[10] A. Imre, et al. *Majority logic gate for magnetic quantum-dot cellular automata*, Science 311.5758 (2006): 205-208.
[11] Li, Wei, et al. "Three-input majority logic gate and multiple input logic circuit based on DNA strand displacement." Nano letters 13.6 (2013): 2980-2988.
[12] K. Bernstein *et al.*, *Device and Architecture Outlook for Beyond CMOS Switches*, Proceedings of the IEEE, 98(12): 2169-2184, 2010.
[13] D. Nikonov, I. Young, "Benchmarking of Beyond-CMOS Exploratory Devices for Logic Integrated Circuits", IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, Volume:PP , Issue: 99, 2015.
[14] Jongyeon Kim et al., "Spin-Based Computing: Device Concepts, Current Status, and a Case Study on a High-Performance Microprocessor," Proceedings of the IEEE , vol.103, no.1, pp.106,130, Jan. 2015
[15] P.-E. Gaillardon, L. Amaru, A. Siemon, E. Linn, A. Chattopadhyay, G. De Micheli, "Computing Secrets on a Resistive Memory Array", (WIP poster) Design Automation Conference (DAC), San Francisco, CA, USA, 2015.
[16] Zografos, Odysseas, et al. "System-level assessment and area evaluation of Spin Wave logic circuits." Nanoscale Architectures (NANOARCH), 2014 IEEE/ACM International Symposium on. IEEE, 2014.
[17] Sandeep Miryala et al., *Exploiting the Expressive Power of Graphene Reconfigurable Gates via Post-Synthesis Optimization*, Proc. GLVSLI'15.
[18] L. Amaru, P.-E. Gaillardon, G. De Micheli, *Majority-Inverter Graph: A Novel Data-Structure and Algorithms for Efficient Logic Optimization*, Proc. DAC, 2014.
[19] L. Amaru, P.-E. Gaillardon, G. De Micheli, *Boolean Logic Optimization in Majority-Inverter Graphs*, Proc. DAC, 2015.
[20] L. Amaru, P.-E. Gaillardon, G. De Micheli, *Majority-Inverter Graphs: A New Paradigm for Logic Optimization*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99): 1-14. 2015.
[21] ABC synthesis tool - available online.
[22] G. De Micheli, *Synthesis and optimization of digital circuits*, McGraw-Hill Higher Education, 1994.
[23] A. Bogdanov et al., PRESENT: An Ultra-Lightweight Block Cipher, CHES Tech. Dig., 2007.