

New Logic Synthesis as Nanotechnology Enabler

This paper investigates the relation between logic synthesis and emerging nanotechnologies, and shows how new logic synthesis techniques can enable the identification of the full potential of a given nanotechnology.

By LUCA AMARÚ, *Student Member IEEE*, PIERRE-EMMANUEL GAILLARDON, *Member IEEE*, SUBHASISH MITRA, *Fellow IEEE*, AND GIOVANNI DE MICHELI, *Fellow IEEE*

ABSTRACT | Nanoelectronics comprises a variety of devices whose electrical properties are more complex as compared to CMOS, thus enabling new computational paradigms. The potentially large space for innovation has to be explored in the search for technologies that can support large-scale and high-performance circuit design. Within this space, we analyze a set of emerging technologies characterized by a similar computational abstraction at the design level, i.e., a binary comparator or a majority voter. We demonstrate that new logic synthesis techniques, natively supporting this abstraction, are the technology enablers. We describe models and data-structures for logic design using emerging technologies and we show results of applying new synthesis algorithms and tools. We conclude that new logic synthesis methods are required to both evaluate emerging technologies and to achieve the best results in terms of area, power and performance.

KEYWORDS | Beyond-complementary metal-oxide semiconductor (CMOS); CAD for nanotechnology; CMOS; emerging devices; enhanced functionality; logic expressive power; logic synthesis; nanotechnology

I. INTRODUCTION

The strong interaction between *Electronic Design Automation* (EDA) tools and *complementary metal-oxide semiconductor* (CMOS) models contributed substantially to the advance-

ment of modern digital electronics. The continuous downscaling of CMOS *field effect transistor* (FET) dimensions enabled the semiconductor industry to fabricate digital systems with higher circuit density at reduced costs. To keep pace with technology, EDA tools are challenged to handle both digital designs with growing number of components and device models of increasing complexity. Logic synthesis is a core EDA task that matured under these conditions. Nevertheless, whereas the downscaling of CMOS technology is requiring more complex physical design models, the logic abstraction of a transistor as a switch has not changed even with the introduction of 3-D FinFET technology.

The arrival of post-CMOS nanotechnologies has brought new computational paradigms and new logic devices. The former are exemplified by quantum computing [1], adiabatic computation [2] and neurocomputing [3] while the latter relate to device models which are different from traditional transistors. For both of them, new logic abstractions and synthesis techniques are key to automatically designing large computing systems. As the purpose of this paper is to delve into new digital logic models and synthesis tools, we neglect considerations related to analog computing and other nondigital computational paradigms. We concentrate instead on new digital technologies whose elementary devices have an enhanced native functionality as compared to standard (MOS) transistors. In this context, the term elementary is equivalent to atomic, in its original Greek interpretation as indivisible. Thus, we focus on a promising class of nanotechnologies whose elementary device abstraction is either a Boolean comparator or a majority voter. Specific examples for these nanotechnologies include, but are not limited to, silicon nanowires [4]–[7], carbon nanotubes [8], [9], graphene [10]–[14], nanorelays [15], [16], resistive random-access memory [17], [18], spin-wave devices [32], organic FETs [25]–[31], quantum-dot cellular automata [36], nanomagnets [37], reversible logic

Manuscript received September 15, 2014; revised April 30, 2015 and July 2, 2015; accepted July 16, 2015. Date of publication August 20, 2015; date of current version October 26, 2015. This work was supported in part by ERC-2009-AdG-246810, SNSF-200021-146600, and by the NSF, STARNet SONIC.

L. Amarú, P.-E. Gaillardon, and G. De Micheli are with the EPFL, I&C Faculty, Integrated Systems Laboratory, Lausanne 1015, Switzerland (e-mail: luca.amaru@epfl.ch; pierre-emmanuel.gaillardon@epfl.ch; giovanni.demicheli@epfl.ch).

S. Mitra is with the Electrical Engineering and Computer Science Department, Stanford University, Palo Alto, CA 94305 USA (e-mail: subh@stanford.edu).

Digital Object Identifier: 10.1109/JPROC.2015.2460377

0018-9219 © 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

[38], [39], molecular computing [43], [44], and many others [46], [150], [151].

Despite the wide availability of choices, there is a large uncertainty on which nanotechnology will prevail and when. As device characteristics and abstractions evolve, logic synthesis no longer only plays the original role of mapping functionality to netlists of transistors. Novel logic abstractions and synthesis techniques, capable of fully utilizing the expressive power of nano devices, are essential to unlock the true value of the candidate nanotechnologies, otherwise unseen by plain synthesis methodologies. Thus logic synthesis turns into a technology enabler in addition to being a technology supporter. Similar to logic synthesis, other EDA tasks are increasingly important to emerging technologies. Imperfection-immune design techniques overcome intrinsic limitations of carbon nanotubes [49], [56]. Variability-aware methodologies handle physical variations at the nanoscale regime [57], [58]. Without these EDA techniques, many emerging technologies would be prematurely disregarded.

In this paper, we revisit logic synthesis in light of its enabling role in the selection of post-CMOS technologies. We first survey a selected pool of nanodevices, featuring enhanced logic functionality, exemplified by Boolean comparators and majority voters. Next, we focus on logic synthesis, with novel circuit models based on the biconditional and majority logic connectives. Experimental results, over six representative nanotechnologies, show that dedicated logic synthesis not only generates better circuits than standard synthesis tools but also enables a larger gain versus CMOS. This is indeed an empirical demonstration of potentially advantageous nanotechnologies that cannot emerge without the leading support of logic synthesis.

Overall, this paper addresses the general question: *Why should EDA bother about emerging nanotechnologies?* We argue that new methods and tools are necessary for the progress of electronics, as we have reached a multiforking point of technology where choices are tougher than ever.

The remainder of this paper is organized as follows. Section II justifies the interest of EDA in emerging technologies through practical considerations and examples. Section III surveys some promising post-CMOS devices. Section IV describes logic synthesis models natively fitting the functionality of the devices presented in Section III. Section V shows experimental results for six different nanotechnologies from Section III using the logic synthesis techniques presented in Section IV. Section VI discusses the outcomes of the current study. Section VII is a conclusion.

II. EDA MEETS EMERGING TECHNOLOGIES

There are emerging technologies that are real. Electronic systems entirely based on nonconventional technologies (different from CMOS) have been fabricated recently, comprising several hundreds of elementary devices. Fig. 1 depicts

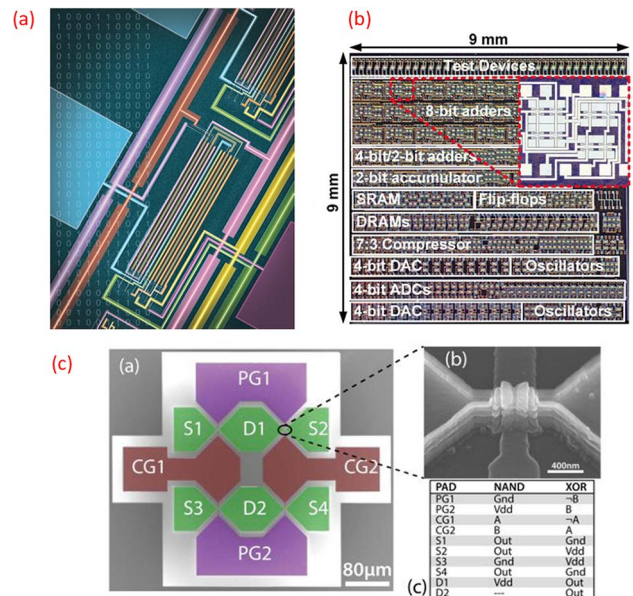


Fig. 1. (a) Real systems built with emerging technologies. First carbon nanotube computer [64]. (b) Memory/logic circuits realized with nanorelays on a chip [16]. (c) Unate/binate configurable logic gates fabricated with SiNWs [5].

some of these systems. A partial image of the first carbon nanotube-based computer [64] is shown in Fig. 1(a). Several memory and logic blocks realized with electromechanical relays on a chip [16] are depicted by Fig. 1(b). Configurable unate/binate logic gates based on vertically-stacked SiNWs [5] are shown in Fig. 1(c). Other concrete emerging technologies demonstrations also exist. To enable these systems, advance in processing and fabrication techniques alone is not sufficient. In order to make these demonstrations real, emerging technologies need EDA techniques.

EDA is an engineering domain consisting of methods and tools used to design complex electronic systems. Starting from a *high-level description* of a logic system, a typical EDA flow operates on logic abstractions of the system and produces a GDSII file (planar geometric shapes) as final result [120]. The main steps involved in a design flow are *high-level synthesis*, *logic synthesis* and *physical design*. In parallel to, or after, these steps, verification and testing methods make sure that the final electronic product operates correctly. The three design steps are subject to area, delay and power minimization metrics. Most contemporary EDA techniques are fine tuned for the silicon CMOS technology.

With the rise of nanotechnologies, EDA is not business as usual. Here the challenge is not just to support a nanotechnology but also to identify early its potential for next-generation computing systems. Indeed, specialized EDA techniques can provide the means for a nanotechnology to happen and to be competitive. In this section, we show actual examples where the early interaction between EDA and emerging nanodevices is key to success. First, we

discuss on how logic synthesis can harness the enhanced functionality of nanodevices. Second, we consider physical design methods immune to imperfections, demonstrated vital to carbon nanotube technology. Finally, we deal with variability in nanoscale devices from an EDA perspective.

In this work, we focus on the role of logic synthesis as nanotechnology enabler. As in the eighties logic synthesis made semicustom design real, we believe that today logic synthesis will contribute substantially to decide the winner nanotechnology for computing beyond CMOS.

A. Harnessing Enhanced Functionality Devices

In the quest for a CMOS technology successor, a plethora of different nanotechnologies are under consideration. Some of them mainly offer an enhanced performance over CMOS while others offer an enhanced functionality over standard switches. The drawback of enhanced functionality devices is usually an extra implementation cost reducing the performance. Nevertheless, enhanced functionality nanodevices are more advantageous than other candidates because their logic expressive power permits to build complex systems with fewer physical resources than standard switches [138]. For instance, spin-wave devices, RRAM-based switches, nanomagnetic logic and quantum-dot cellular automata have a majority/minority gate as circuit primitive that is more expressive than NAND/NOR gates in CMOS [136]. In these, and all others, enhanced functionality nanodevices the increased logic expressivity is a pivotal asset to be harnessed via *logic synthesis*. Otherwise, the true value of a nanotechnology might be unseen. We will provide a detailed discussion on this topic in the next two sections of this paper.

B. Imperfection-Immune Design

Many emerging nanotechnologies, such as carbon nanotubes and bottom-up self-assembled silicon nanowires, are inherently highly subject to imperfections. The reason behind the the presence of imperfections is twofold. On the one hand, it is difficult to control material properties at the scale of few nanometers. On the other hand, it is complicated to have a device-precise fabrication on an integrated circuit with billions of nanodevices. Nevertheless such nanotechnologies are being seriously explored to build highly energy-efficient systems of the future. The inherent imperfections must be overcome before such nanotechnologies can be harnessed with practical benefits to society.

For example, *carbon nanotube field-effect transistors* (CNFETs) represent a significant departure from today's silicon ICs. Fundamental limitations inherent to *carbon nanotube* (CNT) technology pose major obstacles to VLSI demonstration and previous CNT processing techniques alone are inadequate to overcome these challenges. However, the synergy between CNT processing and EDA techniques, referred to as the imperfection-immune paradigm [49]–[51], [53], [54], addresses these challenges. Thanks to this approach, experimental demonstration of CNFET-

based digital systems has been made possible, e.g., all-digital sensor interface circuits [67], [68], and the first microprocessor built entirely using CNFETs [64]–[66].

C. Coping With Variability

Most integrated circuits built with nanoscale devices face serious variability issues. In older technologies, physical parameters extracted from local geometries, such as length and width of a transistor, were sufficient to run accurate simulation/design interactive tasks. When scaling down to nanoscale feature sizes, it is more difficult to control and predict accurately the behavior of a device.

Solutions different than circuit overdesign [69] exist for the variability problem. One example is to construct computing machines purposely exposing hardware variations to various layers of the system stack, including software [70]. In general, hardware variations are peculiar to a technology. Indeed, in different technologies the variability problem manifests with characteristic issues. For example, CNFETs are subject variations in CNT type (metallic or semiconducting CNT) [71], CNT diameter [74], CNT density [72], CNT alignment [51] and CNT doping. In this scenario, EDA techniques help in overcoming these issues by: 1) variation-aware design of aligned-active layouts; and 2) by process-design coexploration and cooptimization. The aligned-active design technique [73] achieves more than an order of magnitude reduction in CNT count failure probability by aligning the active regions of all CNFETs along the direction of the aligned CNTs. The CNT process-design coexploration [75], [76] accurately identifies the most important processing parameters, in conjunction with CNFET circuit sizing, to achieve high energy-efficiency while satisfying circuit-level noise margin constraints.

The ones above are relevant examples on where EDA meets nanotechnologies. In this work, we particularly focus on the enabling role of logic synthesis for nanotechnologies. For this reason, we do not provide a more detailed discussion on physical design or other EDA techniques for nanotechnologies. However, we refer the interested reader to [64]–[68], [76] for recent accomplishments enabled by physical design for CNT nanotechnologies.

By focusing on new logic synthesis methods, we will unveil the hidden potential of enhanced functionality nanodevices.

III. SURVEY OF ENHANCED FUNCTIONALITY NANODEVICES

In this section, we review a set of promising nanodevices in post-CMOS technologies. Using conventional or nonconventional physical phenomena to carry digital information, all the considered devices inherently implement either a two-input comparator or a majority voter. Such enhanced functionality is a key asset for those nanodevices in addition to their intrinsic performances. We limit our survey to devices already supported by some experimental

demonstration. Finally, we present two common logic abstractions for these devices, useful to drive dedicated logic synthesis techniques.

A. Silicon Nanowire FETs

Silicon NanoWires FETs (SiNWFETs) are considered a near-term nanodevice technology. This is because of their ultimate mono-dimensional semiconductor properties combined with the vast experience and investment in Si technologies and its compatibility with traditional CMOS implementation. Fabrication technologies for SiNW have been the object of recent investigation through bottom-up [77]–[80] and top-down [81], [82] approaches.

Top-down, i.e., lithography-based, SiNW technologies are credible for VLSI circuits because of the accurate control of device size and geometry [48], [83] as well as of the proven capability of Si technology to scale-up to billions of devices per chip. In addition, top-down fabrication techniques are capable of fabricating arrays of stacked nanowires [4], [6], [7]. Vertically-stacked NWFETs exploiting gate-all-around structure are the most advanced extension of FinFETs. Considered as a longer term solution, bottom-up integration can lead to impressive density of integration and will push even further the limits of Si technologies, even though they require complex transfer procedures of pregrown nanowires to a final substrate [77]–[80].

SiNWFETs can be doped and used to implement standard complementary logic [47]. However, it is important to note that, at advanced technology nodes, SiNWFETs exhibit an ambipolar behavior, i.e., superposition of *n*- and *p*-type carriers. Instead of suppressing the ambipolar phenomenon by additional complex process steps, an alternative approach that controls the ambipolar properties by an electrostatic mean has been recently developed [4]. By using a double-gate structure, it is possible to select on-line which carrier species prevails in conduction and suppress the other [4], [6], [7].

Fig. 2 depicts the conceptual sketch and fabrication pictures of vertically-stacked double-gate SiNWFET devices from [4]. There, source/drain regions are Schottky contacts, responsible for the device ambipolarity. The second gate, commonly called polarity gate, tunes the Schottky barriers choosing the channel carriers type, whereas the central gate modulates the amount of carriers flowing into the channel. In such a double-gate device, the *n*- or

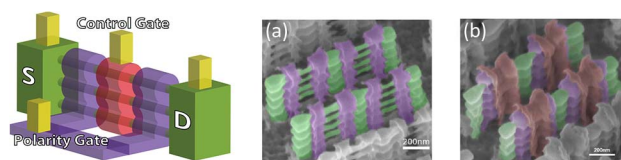


Fig. 2. Vertically stacked double-gate SiNWFET conceptual structure and fabricated devices from [4].

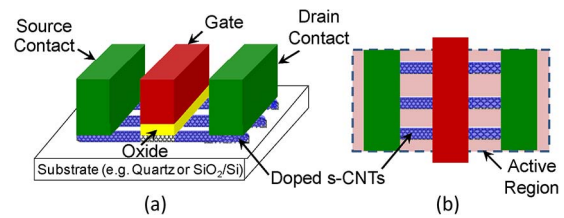


Fig. 3. CNFET structure: (a) 3-D view and (b) 2-D layout view.

p-polarity is electrically configurable via the second gate. The benefit deriving from this feature is twofold. First, the electrostatic doping of the transistor suppresses the need for dopant, hence simplifying its fabrication process. Second, the on-line polarity configuration enhances the logic functionality of the device. This second aspect can be seen as a promising alternative to Moore's Law, whose functionality of device is enriched rather than having its dimensions scaled.

B. Carbon Nanotube FETs

CNTs have drawn considerable attention in electronics due to their superior electrical, thermal, and mechanical properties [90]. CNTs are hollow cylindrical nanostructures made up of carbon atoms. Depending on the detailed arrangement of carbon atoms (also known as CNT chirality [90]) CNTs can be metallic (*m*-CNT) or semiconducting (*s*-CNT). An *m*-CNT has zero or near-zero bandgap, and an *s*-CNT has nonzero bandgap. *s*-CNTs are promising channel materials for building CNFETs [91], [92] that can serve as extensions to silicon MOSFETs. With recent advances in CNT synthesis (especially through chemical vapor deposition techniques), arrays of well-aligned single-walled CNTs can be grown and patterned at wafer scale [52]. These CNTs can be used to fabricate CNFETs on the growth substrate itself, or, they can be transferred onto a target substrate before CNFET fabrication [52]. The imperfection-immune design paradigm discussed in Section II-B is key here to enable the exploitation of CNFET advantageous properties in VLSI systems.

Fig. 3 shows a typical device structure of CNFETs. The CNTs in the device act as transistor channels that can be modulated by a gate. The source and drain regions of CNTs are heavily doped, and the gated regions can be undoped. CNFET devices can act as Schottky-barrier-modulated transistors. In [8], a CNFET concept using dual gates is presented to control the ambipolar behavior deriving from the Schottky-barriers. The ambipolar control is achieved by means of electrostatic doping through the second gate. Fig. 4 shows the device sketch and its fabrication picture from [8]. In this device, the Schottky barrier height is modulated by the fringing gate field at the carbon nanotube-to-metal contact, allowing the device polarity to be set via the back gate voltage. Additionally, the off-state performance and sub-threshold swing are improved.

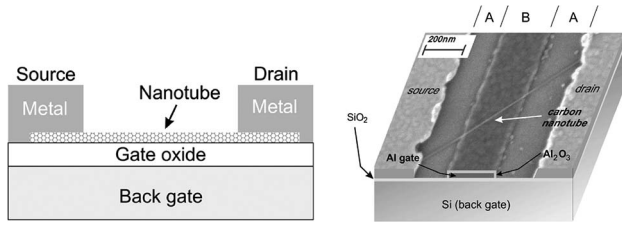


Fig. 4. CNFET device structure and fabrication picture from [8].

Combined with the previously mentioned CNFET benefits, the ambipolar control increases the interest in carbon-based electronics.

C. Resistive RAM

Multitude of emerging *nonvolatile memories* (NVM) are receiving widespread research attention as candidates for high-density and low-cost storage. NVMs store information as an internal resistive state, which can be either a *low resistance state* (LRS) or a *high resistance state* (HRS) [19]. Among the different types of NVMs, Redox-based resistive RAM (RRAM) is considered a leading candidate due to its high density, good scalability, low power and high performance [20], [21]. A different and arguably more tantalizing aspect of RRAMs is their ability to do primitive Boolean logic. The possibility of in-memory computing significantly widens the scope of the commercial applications. To undertake a logic computation, RRAM-based switches are needed. *bipolar resistive switches* (BRS) [22] and *complementary resistive switches* (CRS) [23] have been presented for this purpose. BRS and CRS are devices with two terminals, denoted P and Q . BRS can be used in ultra-dense passive crossbar arrays but suffer from the formation of parasitic currents which create sneak paths. This problem can be alleviated by constructing a CRS device, which connects two BRS anti-serially [23]. For the sake of clarity, we report in Fig. 5 the CRS device conceptual structure proposed in [23] and its sweep properties. Their internal resistance state of the device, Z , can be modified by applying a positive or a negative voltage V_{PQ} . The functionality of BRS/CRS can be summarized by a state machine, as shown

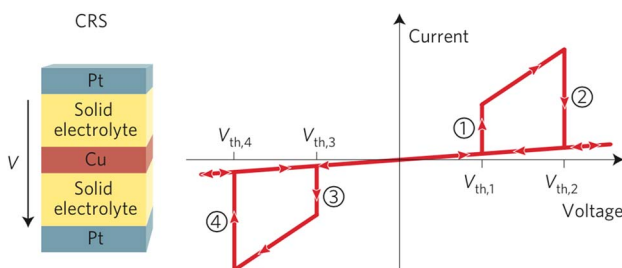


Fig. 5. CRS conceptual structure and sweep properties from [23].

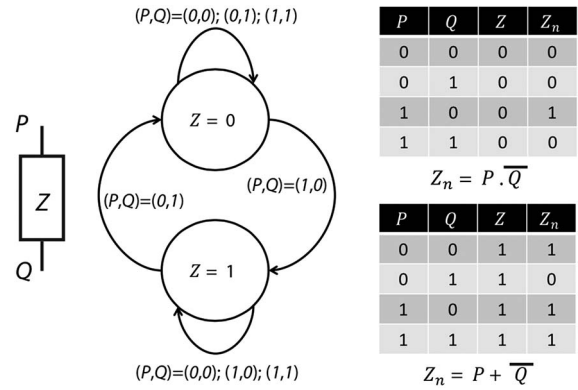


Fig. 6. Resistive majority operation with BRS/CRS devices [139].

in Fig. 6. Further details can be found in [24]. Transition occurs only for the conditions $P = 0, Q = 1$, i.e., $V_{PQ} < 0$ so $Z \rightarrow 0$ and $P = 1, Q = 0$, i.e., $V_{PQ} > 0$ so $Z \rightarrow 1$. By denoting Z as the value stored in the switch and Z_n the value stored after the application of signals on P and Q , it is possible to express Z_n as the following:

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

where M_3 is the majority Boolean function with 3 inputs. A 3-input majority Boolean function is evaluated to be *true* if at least 2 of its inputs are *true*.

The aforementioned resistive RAM technology enables an in-memory computing system, which exploits the same devices to perform both standard storage and computing operations, such as majority voting.

D. NEMS

NEMS or simply nanorelays, are electrostatically actuated mechanical switches [61]. The good properties of nanorelays are: 1) very low on-state intrinsic resistance (0.5 Ω); and 2) virtually infinitely large off-state resistance [60]. On the other hand, the key hurdles of nanorelays are: 1) long switching time (hundreds of nanoseconds); 2) relatively short lifetime (10^8 switching cycles); and 3) limited scalability of minimum feature size [59], [60]. Nanorelays can be fabricated by top-down approaches using conventional lithography techniques or bottom-up approaches using carbon nanotubes or nanowire beams [60].

Nanorelays are a promising alternative to CMOS for ultralow-power systems [59]–[63] where their ideally zero leakage current (consequence of the large off-resistance) is a key feature to be harnessed.

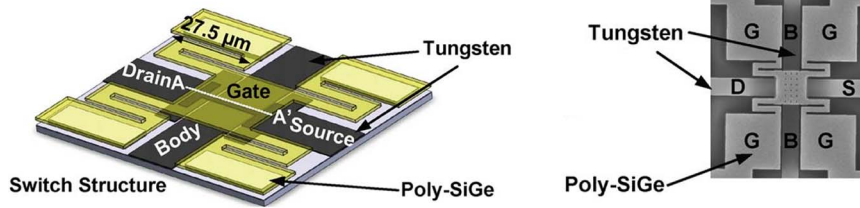


Fig. 7. Four-terminals nanorelay structure and fabrication image from [16].

Different nanorelay structures for logic have been proposed in the literature. Most of them are based on electrostatic actuation and they implement different switching (logic) functions depending on their number of terminals and device geometry. Mechanical contacts (connections) are enforced via electric fields between the various terminals. Two-terminals (2T) and three-terminals (3T) nanorelays are simple devices useful to solve preliminary process challenges. Trading off simplicity for functionality, four-terminals (4T) and six-terminals (6T) nanorelays are more expressive and desirable for compact logic implementations.

In [16], a 4T NEM relay is proposed consisting of a movable poly-SiGe gate structure suspended above the tungsten body, drain, and source electrodes. Fig. 7 shows the 4T relay conceptual structure and a fabrication microphotograph. When a voltage is applied between the gate structure and the body electrode a corresponding electric field arises and the relay is turned on by the channel coming into contact with the source and drain electrodes.

In [15], a 6T NEM relay is realized by adding an extra body (*Body2*) and an extra source (*Source2*) contacts to the previous 4T NEM relay. Fig. 8 shows the 6T relay conceptual structure and a fabrication microphotograph. The two body contacts are designed to be biased by opposite voltages. Either *Source1* or *Source2* to *Drain* connection is controlled by the gate to body positive or negative voltage and its corresponding electric field polarity.

Because of the electrostatic forces among the different terminals, the 6T NEM relay naturally acts as a logic multiplexer driven by a bit comparator.

When technological challenges of NEMS are overcome, 4T and 6T nanorelays can be superior devices for

next-generation ultra low-power systems thanks to their zero-leakage current and increased logic expressivity.

E. 2-D Materials FETs

Introduced as a material science revolution, 2-D crystals, e.g., graphene, MoS₂, *h*-BN [11], [84], can bring a new push in semiconductor devices. Among them, graphene is a very thin, nearly transparent, sheet of one atom-thick carbon material. From an electronics perspective, graphene conduction and scaling properties are highly desirable within a FET device. Indeed, graphene is: 1) more conductive than copper; and 2) allows short-channel effects to be suppressed thanks to its thin channel region [11]. Unfortunately, graphene material still needs to face important challenges. The major crux here is the intrinsic absence of bandgap [10], [11], that prevents a well-defined binary logic state. Other 2-D materials, such as MoS₂, have a band gap [84] but present other fundamental issues (device performance at scaled dimensions, interface between 2-D crystal and dielectric, etc.) [85].

At present, several 2-D material FET structures have been proposed and fabricated with varying success in addressing the material physics issues. Recently, a promising approach emerged from the exploitation of Schottky barriers at the interface between graphene and semiconductors [10]. Fig. 9 shows the device structure and fabrication microphotograph from [10]. By adjusting the gate voltage of this device, it is possible to achieve a large modulation on the device current (on/off ratio of 10⁵) thanks to the control on the graphene-silicon Schottky barrier.

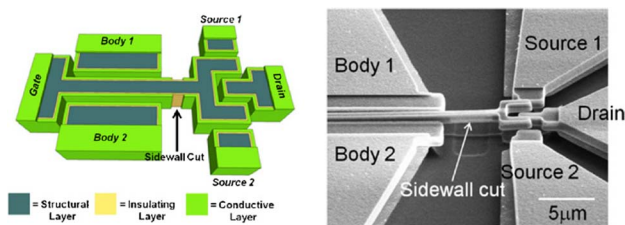


Fig. 8. Six-terminals nanorelay structure and fabrication image from [15].

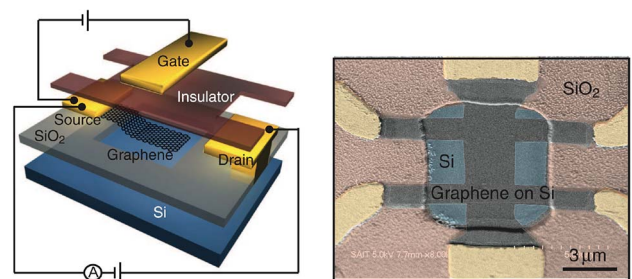


Fig. 9. Graphene FET device (barristor) structure and fabrication microphotograph from [10].

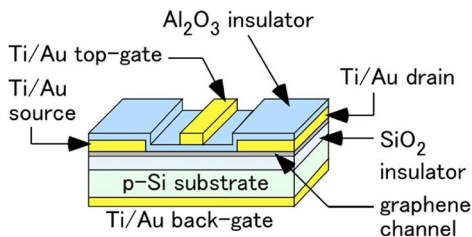


Fig. 10. Graphene FET devices from [13], featuring polarity control.

As for SiNWFETs and CNFETs, the Schottky barriers in graphene FETs can be engineered and used for polarity control via a double gate structure. This option is demonstrated experimentally in [12]–[14], [86]. The devices in [13] and [14] are reported in Figs. 10 and 11, respectively. In both proposals, the back gate is used to switch the device polarity while the top gate serves to modulate the channel current. The physics behind the polarity control in graphene FET is similar to the phenomena previously presented for SiNWFETs and CNFETs.

An alternative strategy to implement graphene switches is to exploit its intrinsic properties rather than modifying them, that is, one can effectively steer carriers across pristine sheets of graphene by means of external voltage potentials. This is conceptually equivalent to building P-N junctions [87]. The P-N junctions, when properly connected, can implement reconfigurable logic gates [88]. A graphene reconfigurable logic gate, 3-D view given in Fig. 12(top-a), consists of four layers [89]. The bottom layer has three metal back-gates (U, S, and U) isolated from each other by oxide. A thick oxide layer is grown on top of these three back-gates, while a graphene sheet is deposited on top. Three front metal-to-graphene contacts (A, Z, B) are then connected to the graphene sheet. The central front-contact Z acts as an output pin and the other two front-contacts, namely A and B, will serve as inputs to the reconfigurable logic gate. The graphene material adapts its doping profile based on the polarity of the back-gate potentials. Depending on the doping configuration, the input carriers injected at the two front-contacts A and B eventually reach the output front-contact Z. The reconfigurable

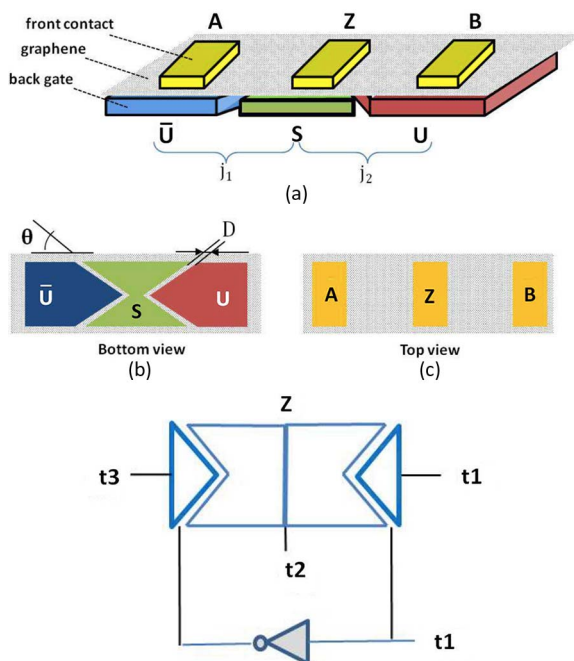


Fig. 12. Structure of reconfigurable graphene logic gate and majority gate implementation from [141].

gate structure can realize expressive primitive functions, most notably the MAJ function with a single device [141], [142]. Fig. 12(bottom) shows the reconfigurable gate implementation of the majority function, i.e., $f(t1, t2, t3) = \text{MAJ}(t1, t2, t3) = t1 \cdot t2 + t2 \cdot t3 + t3 \cdot t1$.

The research on graphene devices is quite active in both academia and industry. When reaching a sufficient level of maturity, graphene digital technology is one of the most desirable successor to CMOS.

F. Spin-Wave Devices

Spin wave devices (SWDs) are digital devices where information transmission happens via spin waves instead of conventional carriers (electrons and holes). The SWD physical mechanism enables ultra-low power operation, almost two orders of magnitude lower than the one of state of the art CMOS [35].

SWDs operate via propagated oscillation of the magnetization in an ordered magnetic material [33]. That oscillation (spin wave) is generated, manipulated and detected through a synthetic multiferroic component, commonly called magneto-electric (ME) cell [34]. The characteristic size of spin-wave devices is the spin wavelength, whose values may range from 30 nm up to 200 nm [35].

On top of the extremely low power consumption of SWD logic, which is a key technological asset, the employment 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

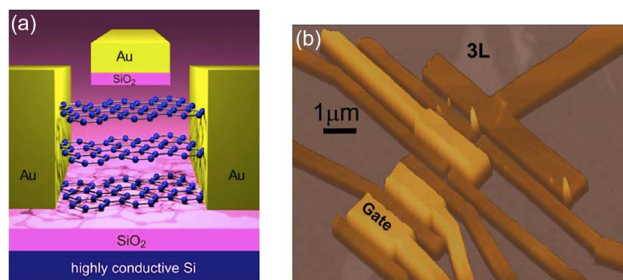


Fig. 11. Graphene FET devices from [14], featuring polarity control.

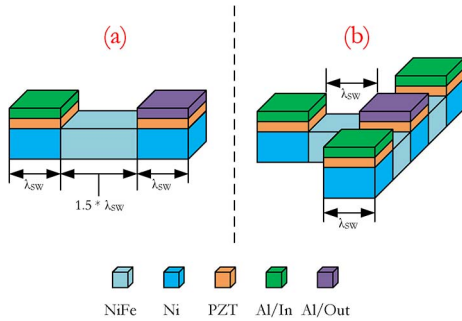


Fig. 13. Primitive gate areas and designs for SWD technology. All distances are parameterized with the spin wave wavelength λ_{SW} [136].

constructive or destructive. The final interference result is translated to the output via magneto-electric cells. In this scenario, an inverter is simply a waveguide with length equal to $1.5 \times$ 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. 13(a). The actual logic primitive in SWD technology is the majority voter, which is implemented by the symmetric merging of three waveguides Fig. 13(b). Here, the length of each waveguide is $1.0 \times$ 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. For the sake of clarity, we report, in Fig. 14, the SWD majority gate physical geometry proposed in [143]. In that structure, spin waves are: 1) excited by the antennas; 2) interfere in the combiner; and 3) are emitted in the output waveguide [143].

From a technological standpoint, there are still several challenges to be addressed before building a real SWD computing system [136]. When and if these challenges will be overcome, the exploitation of SWD logic expressive power, i.e., the native majority voting, will be essential to provide efficient and competitive SWD digital systems.

G. Reversible Logic

The study of reversible logic has received significant research attention over the last few decades. This interest is motivated by the asymptotic zero power dissipation ideally achievable by reversible computation [38]. Reversible logic finds application in a wide range of emerging technologies

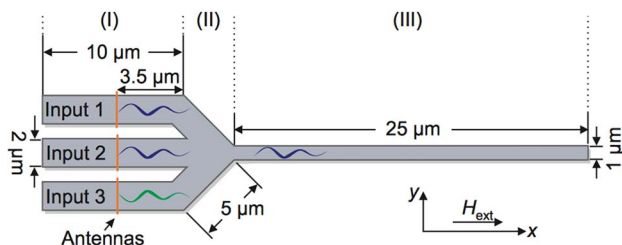


Fig. 14. Geometry of the SWD majority gate proposed by [143].

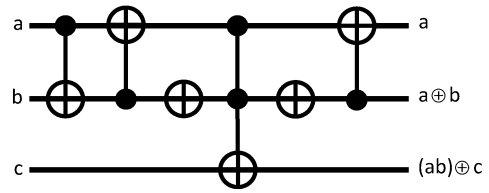


Fig. 15. Reversible circuit made of Toffoli, CNOT and NOT reversible gates.

such as quantum computing [39], optical computing [40], superconducting devices [144], [145] and others [41].

A logic function $f : \mathbb{B}^{n_i} \rightarrow \mathbb{B}^{n_o}$ is reversible if and only if it represents a bijection. This implies that:

- the number of inputs is equal to its number of outputs (i.e., $n_i = n_o$);
- it maps each input pattern to a unique output pattern.

A reversible function can be realized by a circuit $G = g_1, g_2, \dots, g_d$ comprised of a cascade of reversible gates g_i , where d is the number of gates [42]. Several different reversible gates have been introduced including the Toffoli gate [146], the Fredkin gate [147], and the Peres gate [148]. In accordance to the common approach in reversible circuit design, we focus on Toffoli gates. Toffoli gates are universal, i.e., all reversible functions can be realized by means of this gate type alone [146].

A Toffoli gate has a target line t and control lines $\{c_1, c_2, \dots, c_n\}$.¹ Its behavior is the following: If all control lines are set to the logic value 1, i.e., $c_1 \cdot c_2 \cdot \dots \cdot c_n = 1$, the target line t is inverted, i.e., t' . Otherwise, the target line t is passed through unchanged. Hence, the Boolean function of the target line can be expressed as $(c_1 \cdot c_2 \cdot \dots \cdot c_n) \oplus t$. All remaining signals (including the signals of the control lines) are always passed through unchanged. A Toffoli gate with a single control line is called CNOT gate. A Toffoli gate with no control lines is a NOT gate.

For the sake of clarity, we report in Fig. 15 an example of reversible circuit made of Toffoli reversible gates. We follow the established drawing convention of using the symbol \oplus to denote the target line and solid black circles to indicate control connections for the gate. An \oplus symbol with no control lines denotes a NOT gate.

Primitive Toffoli gates have been demonstrated in various emerging technologies. For the sake of clarity, we report briefly hereafter on a recent superconducting-based implementation of a Toffoli gate [145]. It consists of three superconducting transmon qubits coupled to a microwave resonator. Fig. 16 depicts its conceptual implementation. First (I), resonant microwave pulses are applied to the qubits on the corresponding gate lines. Second, the Toffoli gate computation (II) happens via three flux pulses and

¹Toffoli gates have originally been introduced with just two control lines. However, their functionality is commonly extended to n control lines.

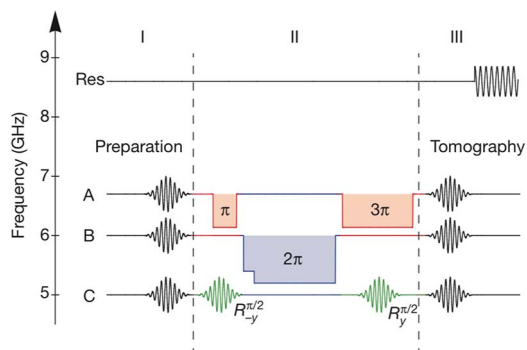


Fig. 16. Pulse sequence used for the implementation of the superconducting Toffoli gate from [145].

resonant microwave pulses. Finally, the output measurement (III) consists of microwave pulses that turn the qubit states to the desired measurement axis, and a subsequent microwave pulse applied to the resonator is used to perform a joint dispersive read-out [145].

Whether finally realized in one emerging technology or the other, reversible circuits must exploit at best the logic expressive power of reversible gates. Being the Toffoli gate the most known reversible gate, harnessing the biconditional connective embedded in its functionality is of paramount importance.

H. Logic Abstraction and Model for Post-CMOS Devices

A digital device is traditionally modeled as a switch whose on/off state is driven by a logic function, commonly called *switching function*. For CMOS FETs, the switching function is just an *inversion* (or *buffer* depending on a chosen polarity convention). For emerging nanodevices, the switching function is usually more complex. Let us consider the devices surveyed so far.

Double-gate SiNWFETs, CNFETs, graphene FETs and organic FETs can be engineered to allow device polarity control. The switching function of these devices is biconditional on both gates (polarity and control) values. Indeed, to be in the on-state, the polarity gate must be either set to logic 1 (*n-type*) and the control gate set to logic 1, or the polarity gate set to logic 0 (*p-type*) and the control gate set to logic 0. This means that a path between source and drain terminals is enabled when both gates assume the same logic value, disabled otherwise. In other words, these devices operate as a switch driven by a *single bit comparator*. Also, the 4T and 6T nanorelays previously presented operate similarly. The source to drain connection in these nanorelays is controlled by the gate to body voltage sign and amplitude. In the binary domain, this corresponds to a *bit comparator* between the gate and body logic values. Finally, also reversible logic gates, such as Toffoli gates, embed the biconditional connective in their operation. Indeed, bicon-

ditional (XNOR) operations are easily reversible while other logic operations, such as conjunctions and disjunctions, are not. Fig. 17 depicts the common logic abstraction for those comparator-intrinsic nanodevices.

The remaining nanodevices surveyed, such as SWD, RRAM and graphene reconfigurable gates, operate using different physical phenomena than standard FETs. For example, SWD uses spin waves as information carrier while CRS logic behavior depends on the previous memory state. In those nanotechnologies, the circuit primitive is not anymore a standard switch but a three-input majority voter. Note that there are other nanotechnologies where majority voters are the circuit primitive. Quantum-dot cellular automata is one well-known voting-intrinsic nanotechnology [36]. Also, DNA strand displacement recently showed the capability to implement voting logic [149]. For the sake of brevity, we do not give details on QCA and DNA voting-intrinsic nanotechnologies. However, it is important to keep in mind that majority voting is the basic computational brick also for other nanotechnologies than the ones surveyed in this work. Fig. 18 depicts the common logic abstraction for these voting-intrinsic nanodevices.

The far reaching consequence of embedding biconditional and majority operators as atomic functions in nanotechnologies is the ability to implement arithmetic function with few physical resources. The logic models in Figs. 17 and 18 serve as input to synthesis techniques targeting comparator-intrinsic and voting-intrinsic nanotechnologies.

IV. NEW LOGIC SYNTHESIS FOR NANOTECHNOLOGIES

In this section, we present new logic synthesis models for nanodevices. We first give a brief overview on logic synthesis. Then, we discuss how advances in logic synthesis will relate to emerging nanotechnologies. Finally, we introduce biconditional and majority logic models as native design abstraction for the nanotechnologies surveyed in the previous section.

A. Brief Overview

The field of logic synthesis originally started from the classic work on switching theory [96]. It then took a sharp turn in the eighties with the advent of *application-specific integrated circuits* (ASICs), enabled by *very-large-scale integration* (VLSI), demanding for advanced synthesis techniques and tools. At that point, multilevel logic synthesis emerged along with other important techniques, e.g., binary decision diagrams [97]–[99], retiming [100], technology mapping [101], sequential equivalence checking [103], etc. Based on these techniques, YLE [106] and academic MIS [105] synthesis systems became rapidly essential for digital system design. Two-level synthesis tools, like Espresso [104], kept being used inside multilevel synthesis tools for local optimization. Later on, in the nineties, other

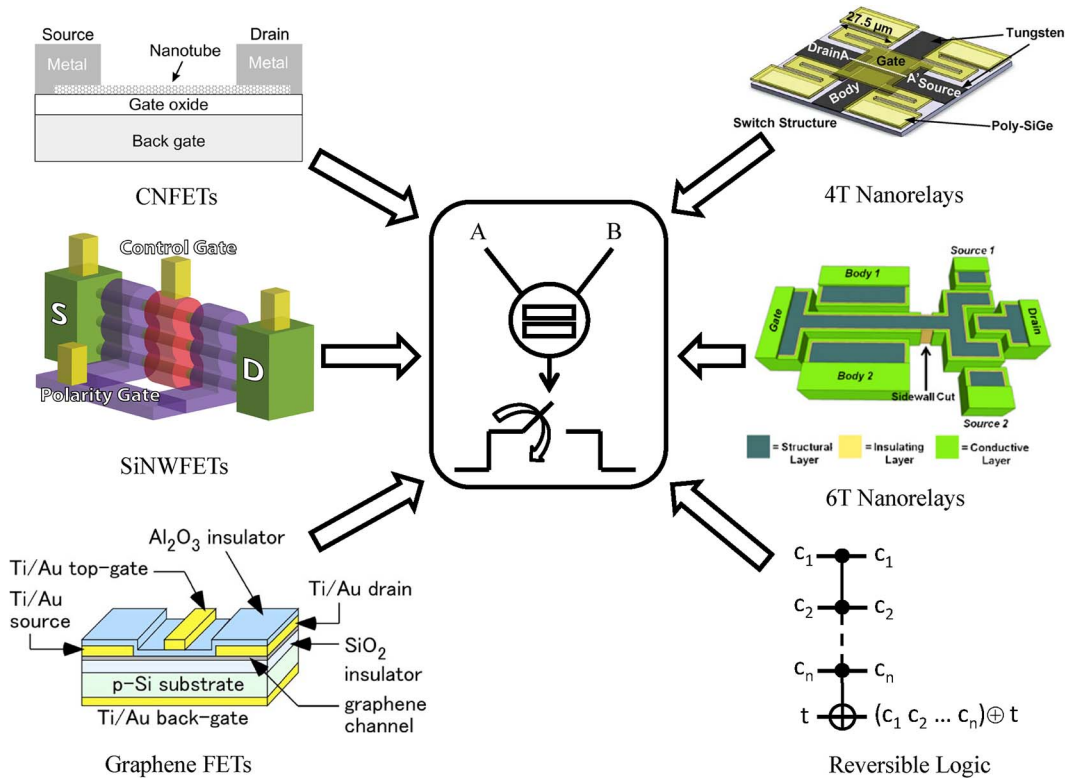


Fig. 17. Common logic abstraction for SiNWFETs, CNFETs, graphene FETs, reversible logic and nanorelays. Logic model: switch driven by a comparator.

important problems required the attention of logic synthesis, such as testing interaction with synthesis, power consumption, delay variations (due to the interconnects) etc. Advanced algorithms capable to deal with these issues were based on don't cares [107], image computation [108], symbolic manipulation [109], [110] etc. The SIS [111] academic synthesis tool arose in those years and remained a synthesis standard for a long period. With the new millenium, the main directions for logic synthesis shifted to scalability, design closure between logic and physical synthesis, verification during synthesis, etc. This was when modern synthesis methods arose, including *And-Inverter Graphs* (AIGs) (popularized in [114] and [119]), decomposed BDDs [112], [113], SAT-based Boolean optimization [116], disjoint support decomposition [117], [118], etc. Nowadays, the state-of-art academic synthesis tool is ABC [114], [119] that includes a large collection of synthesis algorithms running on various data structures.

B. Motivation

During all its evolution, the key factor behind the success of logic synthesis was the ability to optimize and map *hardware description languages* onto negative unate logic gates, efficient in CMOS. Along these lines, logic synthesis supported CMOS technology evolution over the four past decades.

In the future, we do believe that logic synthesis will not just support nanotechnologies but also enable their rise.

In this work, we show how proper logic models and synthesis techniques unlock the full potential of the emerging nanodevices presented in Section III. Brought together under two expressive logic models, the previously studied nanodevices embed the biconditional connective or the majority voter functionality in their logic operation. We present new logic representation forms based on these logic connectives. Thanks to an explicit logic commonality, such logic representation forms provide a natural and native design abstraction for the previously studied nanodevices.

C. Biconditional Binary Decision Diagrams: Representing and Manipulating Comparator-Intrinsic Nanotechnologies

The automated design of comparator-intrinsic nanotechnologies, such as the ones introduced in Section III, demands for a data structure naturally embedding the comparator functionality in the binary domain, i.e., the biconditional logic connective. For this purpose, we propose a canonical variant of the popular binary decision diagrams replacing Shannon's expansion with the *biconditional expansion*.

1) *Brief Background on BDDs*: To support logic manipulation and optimization tasks, it is important to represent

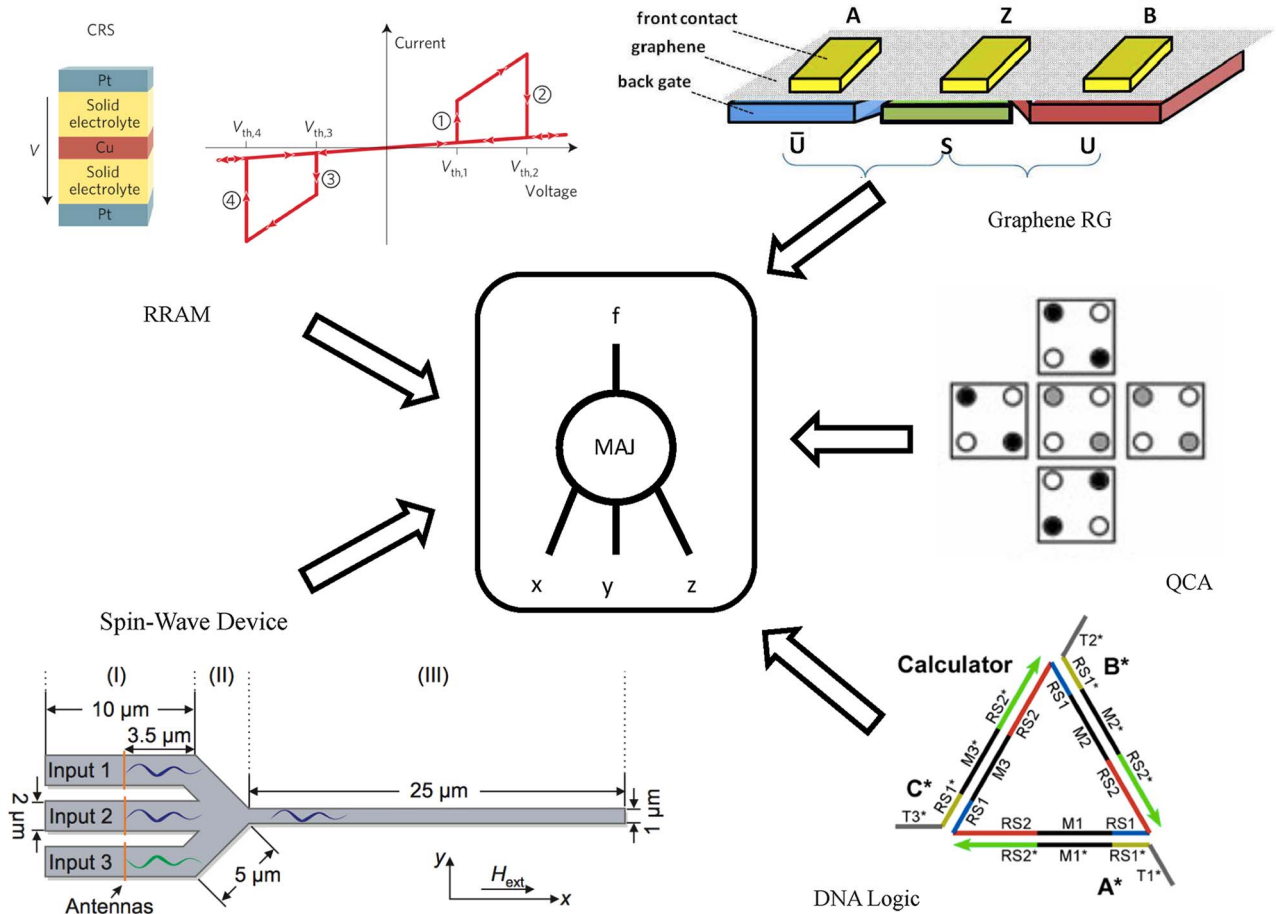


Fig. 18. Common logic abstraction for SWD, RRAM, Graphene reconfigurable gates, QCA and DNA logic. Logic model: majority voter.

logic functions efficiently. *binary decision diagrams* (BDDs) [97]–[99] are a popular data structure for this purpose. Original BDDs are driven by Shannon’s expansion to decompose a Boolean function until the constant logic values are encountered. Reduced and ordered BDDs [99] are unique for a given variable order, i.e., canonical. BDDs enable efficient logic manipulation [121], [122]. However, BDDs face limitations in terms of representation size. In fact, BDDs can be exponential-sized for some functions, e.g., multipliers, hidden weighted bit function etc. Canonical extensions of BDDs have been investigated in literature to improve the representation compactness. One notable example is *Kronecker functional decision diagrams* (KFDDs) [123] utilizing both Davio’s expansions and Shannon’s expansion. Still, some functions can take exponential representation size with KFDDs [123]. Even though canonical decision diagrams can face scalability issues, they provide a small and easily manipulable representation for many practical logic functions.

We are interested here in an extension of BDDs, where the Shannon’s expansion is replaced by a *biconditional expansion*, to support natively the functionality of emerging nanodevices. The corresponding decision diagram is called

biconditional binary decision diagram (BBDD), recently introduced in [124]–[126]. We review the basics of BBDDs in the following.

2) *Biconditional Expansion*: Logic expansions, also called decompositions, are the driving core of various types of decision diagrams. We consider a novel logic expansion, called *biconditional expansion*, examining two variables per time rather than one, in order to produce novel compact decision diagrams.

Definition: The *biconditional expansion* is a two variable expansion defined $\forall f : \mathbb{B}^n \rightarrow \mathbb{B}$, with $n > 1$, as

$$(v, w, \dots, z) = (v \oplus w) \cdot f(w', w, \dots, z) + (v \odot w) \cdot f(w, w, \dots, z) \quad (1)$$

with v and w distinct elements in the support for function f .

As per the *biconditional expansion* in (1), only functions that have two or more variables can be decomposed.

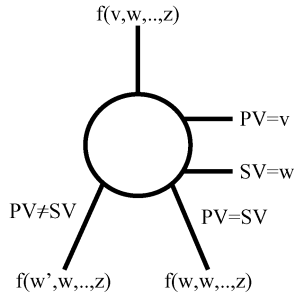


Fig. 19. BBDD nonterminal node.

Indeed, in single variable functions, the terms $(v \oplus w)$ and $(v \odot w)$ cannot be computed. In such a condition, the *biconditional expansion* of a single variable function can reduce to Shannon's expansion by fixing the second variable w to logic 1. With this boundary condition, any Boolean function can be fully decomposed by *biconditional expansions*.

3) *BBDD Structure and Ordering*: BBDD are driven by the *biconditional expansion*. Each nonterminal node in a BBDD has the branching condition *biconditional* on two variables. We call these two variables the *primary variable (PV)* and the *secondary variable (SV)*. An example of a BBDD nonterminal node is provided by Fig. 19. We refer hereafter to $PV \neq SV$ and $PV = SV$ edges in a BBDD node simply as \neq -edges and $=$ -edges, respectively.

To achieve *ordered BBDDs* (OBBDDs), a variable order must be imposed for PVs and a rule for the other variables assignment must be provided. The *chain variable order (CVO)* addresses this task. Given a Boolean function f and a variable order $\pi = (\pi_0, \pi_1, \dots, \pi_{n-1})$ for the support variables of f , the CVO assigns PVs and SVs as

$$\begin{cases} PV_i = \pi_i \\ SV_i = \pi_{i+1} \end{cases} \quad \text{with } i = 0, 1, \dots, n-2; \quad \begin{cases} PV_{n-1} = \pi_{n-1} \\ SV_{n-1} = 1. \end{cases} \quad (2)$$

CVO Example: From $\pi = (\pi_0, \pi_1, \pi_2)$, the corresponding CVO ordering is obtained by the following method. First, $PV_0 = \pi_0$, $PV_1 = \pi_1$, and $SV_0 = \pi_1$, $SV_1 = \pi_2$ are assigned. Then, the final boundary conditions of (2) are applied as $PV_2 = \pi_2$ and $SV_2 = 1$. The consecutive ordering by couples (PV_i, SV_i) is thus $((\pi_0, \pi_1), (\pi_1, \pi_2), (\pi_2, 1))$.

The CVO is a key factor enabling unique representation of ordered biconditional decision diagrams. For the sake of clarity, we first consider the effect of the CVO on *complete OBBDDs* and then we move to generic reduced BBDDs.

Definition: A *complete OBBDD* of n variables has 2^{n-1} distinct internal nodes, no sharing, and 2^n terminal 0-1 nodes.

Lemma 4.1: For a Boolean function f and a variable order π , there exists only one *complete OBBDD* ordered with $CVO(\pi)$.

Proof: We refer the reader to [126] for a formal proof. The rationale behind the proof is the following. A complete OBBDD is exponential sized and consequently must have 2^n distinct paths. Each path points to the entry of a truth table build with the same variable order π . This makes a one-to-one correspondence between truth tables and complete OBBDDs. As a truth table under a fixed variable order π is canonical, so is a corresponding complete OBBDD.

We refer hereafter to OBBDDs as to BBDDs ordered by the CVO.

4) *BBDD Reduction*: In order to improve the representation efficiency, OBBDDs should be reduced according to a set of rules. We report hereafter BBDD reduction rules, and we discuss the uniqueness of the so obtained ordered and reduced BBDDs.

Reduction Rules: The straightforward extension of OBBDD reduction rules [99] to OBBDDs, leads to *weakly reduced OBBDDs* (ROBBDDs). This kind of reduction is called *weak* due to the partial exploitation of OBBDD reduction opportunities. A *weak ROBBDD* is an OBBDD respecting the two following rules:

- R1)** it contains no two nodes, root of isomorphic subgraphs;
- R2)** it contains no nodes with identical children.

In addition, the OBBDD representation exhibits supplementary interesting features enabling further reduction opportunities. First, levels with no nodes (empty levels) may occur in OBBDDs. An empty level is a level in the decision diagram created by the CVO but containing no nodes as a result of the augmented functionality of the *biconditional expansion*. Such levels must be removed to compact the original OBBDD. Second, subgraphs that represent functions of a single variable degenerates into a single DD node driven by the Shannon's expansion followed by the sink terminal node. The degenerated node functionality is the same as in a traditional BDD node. Single variable condition is detectable by checking the cardinality of the support set of the subgraph.

Formally, a *strong ROBBDD* is an OBBDD respecting **R1** and **R2** rules, and in addition:

- R3)** it contains no empty levels;
- R4)** subgraph representing single variable functions degenerates into a single DD node driven by the Shannon's expansion.

For the sake of simplicity, we refer hereafter to a single variable subgraph degenerated into a single DD node as a BDD node.

Fig. 20 depicts *weak* and *strong* ROBBDDs for the function $f = a \cdot b + (a \oplus b) \cdot (c \odot d)$. The *weak* ROBBDD is forced to allocate 4 levels (one for each variable) to fully represent the target function resulting in 5 internal nodes. On the other hand, the *strong* ROBBDD exploits

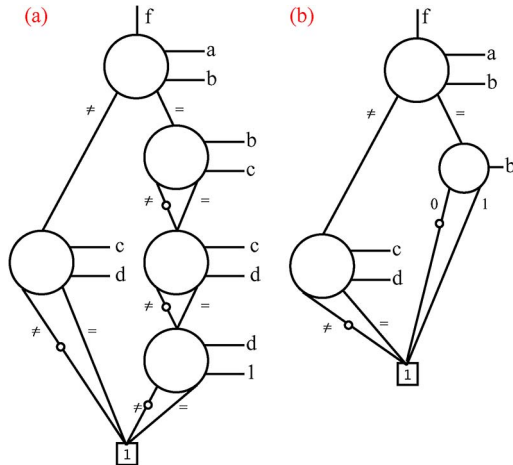


Fig. 20. Function to be represented: (a) $f = a \cdot b + (a \oplus b) \cdot (c \odot d)$, weak ROBDD for f and (b) strong ROBDD for f .

reduction rule **R4**. It collapses the $=$ -branch after the top level into a single BDD node because it represents a function of a single variable. Immediately after that, rule **R3** suppresses 2 empty levels created by rule **R4**. The final reduced diagram counts 3 levels of depth and 3 internal nodes.

Canonicity: Weak and strong reduced OBBDDs are canonical, as per:

Lemma 4.2: For a given Boolean function f and a variable order π , there exists only one weak ROBDD.

Proof: We refer the reader to [126] for a formal proof. The rationale behind this proof is the following. Weak ROBDDs are obtained by applying reduction rules **R1** and **R2**, in any combination, to a complete OBBDD until no other **R1** or **R2** rule can be applied. The iterative reduction of general decision diagrams, based on rules **R1** and **R2**, reaches a unique structure. This preserves the initial complete OBBDD uniqueness. ■

Theorem 4.3: A strong ROBDD is a canonical representation for any Boolean function f .

Proof: We refer the reader to [126] for a formal proof. The rationale behind this proof is the following. Strong ROBDDs can be directly derived by applying reduction rules **R3** and **R4**, in any combination, to weak ROBDDs until no other **R3** or **R4** rule can be applied. In [126] it is shown that any sequence of reductions drawn from $\{\mathbf{R3}, \mathbf{R4}\}$, that continues until no other reduction is possible, reaches a unique strong ROBDD structure, preserving the uniqueness property of the starting weak ROBDD. ■

5) *BBDD Complemented Edges:* Being advantageously applied in modern ROBDDs packages [115], complemented edges indicate to invert the function pointed by an edge. The canonicity is preserved when the complement attribute is allowed only at 0-edges (only logic 1 terminal node available). Reduction rules **R1** and **R2** continue to be valid with complemented edges. Similarly, ROBDDs can use complemented edges only at \neq -edges, with also only logic 1 terminal node available, while maintaining canonicity.

For the sake of simplicity, we refer hereafter to BBDDs as to canonical ROBDDs with complemented edges, unless specified otherwise.

6) *BBDD Examples:* Fig. 21 depicts the BBDDs for some logic functions of interest in today's designs. In Fig. 21(a) a 6-bit parity function is represented with 3 nodes and 3 levels. Note that rule **R3** eliminated half of the levels originally allocated for the BBDD of the 6-bit parity function. This is thanks to the expressiveness of a BBDD node handling two variables per time. Fig. 21(b) and (d) show the BBDDs for mixed AND/OR-XOR intensive functions that frequently appear in datapath circuits. In Fig. 21(c), a 3-input majority function is represented with 3 nodes and 3 levels. Together with the parity-check, the majority function is a basis for binary addition representing the carry propagation.

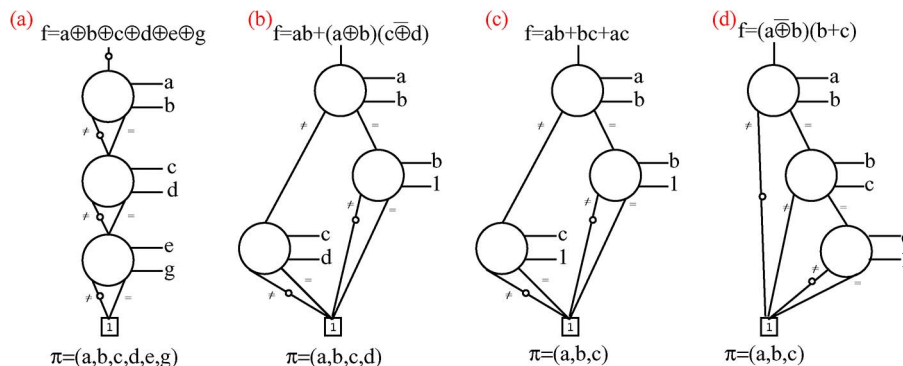


Fig. 21. Examples for BBDD representation of 3 to 6 variables logic functions. (a) Parity, (b) and (d) mixed AND/OR-XOR, and (c) majority functions are considered.

7) *BBDD Manipulation*: So far, we showed that, under ordering and reduction rules, BBDDs are unique and potentially very compact. In order to exploit such features in real-life tools, a practical theory for the construction and manipulation of BBDDs is needed. This theory is presented in [126] and implemented in the software manipulation package available online at [127]. We report hereafter the key concepts enabling efficient BBDD manipulation.

Rationale for Construction and Manipulation of BBDDs: DDs are usually built starting from a netlist of Boolean operations. A common strategy employed for the construction task is to build bottom-up the DD for each element in the netlist, as a result of logic operations between DDs computed in the previous steps. In this context, the core of the construction task is an efficient Boolean operation algorithm between DDs. In order to make such approach effective in practice, other tasks are also critical, such as memory organization and reordering of variables. With BBDDs, the same construction and manipulation rationale is followed, but with specialized techniques taking care of the *biconditional expansion*.

Considerations to Design an Efficient BBDD Package: Nowadays, one fundamental reason to keep decision diagrams small is not just to successfully fit them into the memory, that in a modern server could store up to 1 billion nodes, but more to maximize their manipulation performance. Following this trend, BBDD manipulation algorithms and data structures are designed aiming to minimize the runtime while keeping under control the memory footprint. The key concepts unlocking such target are: 1) *unique table* to store BBDD nodes in a *strong canonical form*; 2) recursive formulation of Boolean operations in terms of *biconditional expansions* with relative *computed table*; 3) memory management to speed up computation; and 4) *chain variable reordering* to minimize the BBDD size.

The variable order in BBDDs, as in other canonical structures, determines the final representation size. The BBDD package automatically finds a good order during construction. While building the BBDD, *chain variable reordering* is continuously applied to a partially built BBDD in order to keep the decision diagram small. Even though this strategy may not reach the global optimum, it typically produces efficient results.

We refer the interested reader to [126] for more details on BBDD manipulation algorithms and software implementation.

D. Majority-Inverter Graphs: Representing and Manipulating Voting-Intrinsic Nanotechnologies

The automated design of voting-intrinsic nanotechnologies, such as the ones introduced in Section III, demands for a data structure naturally embedding the majority functionality. For this purpose, we review *majority-inverter graphs* (MIGs) [128], a new type of homogenous logic network, where all nodes represent the three-input majority function, empowered by a sound and complete Boolean algebra.

1) MIG Logic Representation:

Definition: An MIG is a homogeneous logic network with an indegree equal to 3 and each node representing the majority function. In an MIG, edges are marked by a regular or complemented attribute.

To determine some basic logic representation properties of MIGs, we compare them to the well-known *AND/OR/inverter graphs* (AOIGs) (which include AIGs). In terms of representation expressiveness, the elementary bricks in MIGs are *majority operators* while in AOIGs are conjunctions (AND) and disjunctions (OR). It is worth noticing that a majority operator $M(x, y, z)$ behaves as the conjunction operator $AND(x, y)$ when $z = 0$ and as the disjunction operator $OR(x, y)$ when $z = 1$. Therefore, majority is actually a generalization of both conjunction and disjunction. Recall that $M(x, y, z) = xy + xz + yz$. This property leads to the following theorem.

Theorem 4.4: MIGs \supset AOIGs.

Proof: We refer the reader to [128] for a formal proof. The rationale behind this proof is the following. An AOIG node is always a special case of an MIG node, with the third input biased to logic 0 or 1 to realize an AND or OR, respectively. On the other hand, an MIG node is never a special case of an AOIG node, because the functionality of the three input majority cannot be realized by a unique AND or OR. ■

As a consequence of the previous theorem, MIGs are at least as good as AOIGs but potentially much better, in terms of logic representation compactness. Indeed, in the worst case, one can replace node-wise AND/ORs by majorities with the third input biased to a constant (0/1). However, even a more compact MIG representation can be obtained by fully exploiting its node functionality rather than fixing one input to a logic constant.

Fig. 22 depicts a MIG representation example for $f = x_3 \cdot (x_2 + (x'_1 + x_0)')$. The starting point is a traditional AOIG. Such AOIG has 3 nodes and 3 levels of depth, which is the best representation possible using just AND/ORs. The first MIG is obtained by a one-to-one replacement of AOIG nodes by MIG nodes. A better MIG representation is

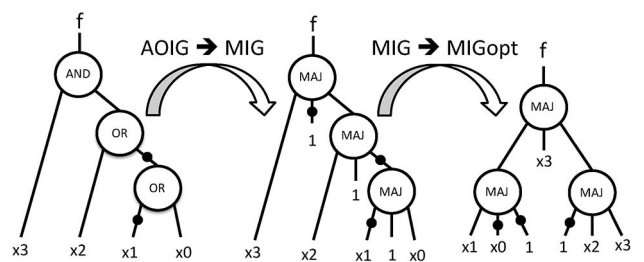


Fig. 22. Examples of MIG representations (right) for $f = x_3 \cdot (x_2 + (x'_1 + x_0)')$ derived by its optimal AOIG representation. Complement attributes are represented by bubbles on the edges.

possible by taking advantage of the majority function by a simple example depicted in Fig. 22. In this way, one level of depth is saved with the same node count. How to automatically derive such optimized MIG representation will be discussed later on in this paper.

MIGs inherit from AOIGs some important properties, like universality and AIG inclusion. This is formalized by the following:

Corollary 4.5: MIGs \supset AIGs.

Proof: MIGs \supset AOIGs \supset AIGs \implies MIGs \supset AIGs. ■

Corollary 4.6: MIG is an universal representation form.

Proof: MIGs \supset AOIGs \supset AIGs that are universal representation forms [120]. ■

So far, we have shown that MIGs extend the representation capabilities of AOIGs. However, we need a proper set of manipulation tools to handle MIGs and automatically reach compact representations. For this purpose, we report hereafter a new Boolean algebra, based on MIG primitive operators.

6) *MIG Boolean Algebra:* We show a novel Boolean algebra, defined over the set $(\mathbb{B}, M', 0, 1)$, where M is the majority operator of three variables and $'$ is the complementation operator. The following set of five primitive transformation rules, referred to as Ω , is an *axiomatic system* for $(\mathbb{B}, M', 0, 1)$. All the variables belong to \mathbb{B}

$$\Omega \left\{ \begin{array}{l} \mathbf{Commutativity} : \Omega.C \\ M(x, y, z) = M(y, x, z) = M(z, y, x) \\ \mathbf{Majority} : \Omega.M \\ \left\{ \begin{array}{l} \text{if } (x = y) : M(x, x, z) = M(y, y, z) = x = y \\ \text{if } (x = y') : M(x, x', z) = z \end{array} \right. \\ \mathbf{Associativity} : \Omega.A \\ M(x, u, M(y, u, z)) = M(z, u, M(y, u, x)) \\ \mathbf{Distributivity} : \Omega.D \\ M(x, y, M(u, v, z)) = M(M(x, y, u), M(x, y, v), z) \\ \mathbf{Inverter Propagation} : \Omega.I \\ M'(x, y, z) = M(x', y', z'). \end{array} \right. \quad (3)$$

Axiom $\Omega.C$ defines a commutativity property. Axiom $\Omega.M$ declares the intended majority decision threshold. Axiom $\Omega.A$ is an associative law extended to ternary operators. Axiom $\Omega.D$ establishes a distributive relation over majority operators. Axiom $\Omega.I$ expresses the interaction between M and complementation operators. It is worth noticing that $\Omega.I$ does not require operation type change like De Morgan laws, as it is well known from self-duality [130].

Note that there are other possible axiomatic systems for $(\mathbb{B}, M', 0, 1)$. For example, one can show that in the presence of $\Omega.C$, $\Omega.A$ and $\Omega.M$, the rule in $\Omega.D$ is redundant. Here, we consider $\Omega.D$ as part of the axiomatic system for the sake of simplicity.

Derived Theorems: Several other complex rules, formally called theorems, in $(\mathbb{B}, M', 0, 1)$ are derivable by Ω . Among them, three derived rules from Ω are of particular interest to logic optimization. We refer to them as Ψ and are described hereafter. In the following, the symbol $z_{x/y}$ represents a replacement operation, say replace x with y in all its appearance in z

$$\Psi \left\{ \begin{array}{l} \mathbf{Relevance} - \Psi.R \\ M(x, y, z) = M(x, y, z_{x/y}) \\ \mathbf{Complementary Associativity} - \Psi.C \\ M(x, u, M(y, u', z)) = M(x, u, M(y, x, z)) \\ \mathbf{Substitution} - \Psi.S \\ M(x, y, z) \\ = M(v, M(v', M_{v/u}(x, y, z), u), M(v', M_{v/u'}(x, y, z), u')). \end{array} \right. \quad (4)$$

The first rule, relevance ($\Psi.R$), replaces reconvergent variables with their neighbors. The second rule, complementary associativity ($\Psi.C$), deals with variables appearing in both polarities. The third rule, substitution ($\Psi.S$), extends variable replacement to the nonreconvergent case. In the following, we show how Ψ rules can be derived from Ω .

Theorem 4.7: Ψ rules are derivable by Ω .

Proof: **Relevance** ($\Psi.R$): Let S be the set of all the possible primary input combinations for $M(x, y, z)$. Let $S_{x=y}$ ($S_{x=y'}$) be the subset of S such that $x = y$ ($x = y'$). Note that $S_{x=y} \cap S_{x=y'} = \emptyset$ and $S_{x=y} \cup S_{x=y'} = S$. According to $\Omega.M$, variable z in $M(x, y, z)$ is only relevant for $S_{x=y}$. Thus, it is possible to replace x with y' (x/y') in all its appearance in z , preserving the original functionality.

Complementary Associativity ($\Psi.C$)

$$M(x, u, M(u', y, z)) = M(M(x, u, u'), M(x, u, y), z) \quad (\Omega.D)$$

$$M(M(x, u, u'), M(x, u, y), z) = M(x, z, M(x, u, y)) \quad (\Omega.M)$$

$$M(x, z, M(x, u, y)) = M(x, u, M(y, x, z)) \quad (\Omega.A)$$

Substitution ($\Psi.S$): We set $M(x, y, z) = k$ for brevity

$$k = M(v, v', k) = (\Omega.M)$$

$$M(M(u, u', v), v', k) = (\Omega.M)$$

$$M(M(v', k, u), M(v', k, u'), v) = (\Omega.D)$$

$$\text{Then, } M(v', k, u) = M(v', k_{v/u}, u) \quad (\Psi.R)$$

$$\text{and } M(v', k, u') = M(v', k_{v/u'}, u) \quad (\Psi.R).$$

Recalling that $k = M(x, y, z)$, we finally obtain: $M(x, y, z) = M(v, M(v', M_{v/u}(x, y, z), u), M(v', M_{v/u'}(x, y, z), u'))$. ■

V. LOGIC SYNTHESIS EXPERIMENTS FOR NANOTECHNOLOGIES

In this section, we experiment the efficacy of BBDDs and MIGs in the synthesis of emerging nanodevices. We target six different emerging technologies taken from the devices surveyed in Section III. Note that many other nanodevices may benefit from the presented majority/biconditional synthesis methodologies [150], [151], however, a precise evaluation of their performance is out of the scope of the current study.

We first explain the basic synthesis strategies employed for BBDDs and MIGs. Then, we present the details for each technology and show the corresponding results.

A. BBDD-Based Synthesis

When representing a logic circuit with BBDDs, its arithmetic structure is highlighted thanks to the biconditional expansions. If the target circuit is arithmetic in nature, then BBDDs are a preferable representation for logic. Moreover, representing a function with BBDDs already provides a better ground for mapping onto comparator-native devices. Owing to these facts, we employ a BBDD package [127] to represent combinational digital logic.

Depending on the target nanotechnology, we either use BBDD for a direct one-to-one mapping into nanodevices or as a frontend to a standard synthesis tool. In the first case, we partition the initial design to better handle the complexity and increase the efficiency of the one-to-one mapping of BBDD elements into nanodevices. Being canonical, BBDDs might not be compact for large functions. Instead, by partitioning the design in medium/small blocks, the BBDD representation stays compact and leads to advantageous results. In the second case, the BBDD representation is kept if it reduces the *And/Or-Inv* graph representation complexity, i.e., the number of nodes and the number of logic levels. Starting from a simpler description, the synthesizer can reach higher levels of quality in the final circuit, especially when such description naturally matches a target technology functionality.

More details on BBDD-based synthesis are given for each specific nanotechnology.

B. MIG-Based Synthesis

MIGs enable compact logic representation and powerful logic optimization. They already showed very promising results for traditional CMOS technology [128], [129]. Moreover, if the target technology natively realizes the MIG primitive function, i.e., a majority voter, the use of MIGs in circuit synthesis produces superior results. For this reason, we use a MIG optimizer, called *MIGhty* [128], to synthesize circuits in voting-intrinsic nanotechnologies.

Depending on the target nanotechnology, we either use MIGs for a direct one-to-one mapping into nanodevices or as a frontend to a standard synthesis tool. In both cases, no

prepartitioning is strictly required as MIG are not canonical per se, thus they scale efficiently with the design size.

More details on MIG-based synthesis are given for each specific nanotechnology.

C. Results

We present results for six different nanotechnologies. Three of them are comparator-intrinsic nanotechnologies and the other three are voting-intrinsic nanotechnologies. Results for BBDD/MIG-based synthesis are presented and compared to a reference design flow, showing the importance of dedicated logic synthesis techniques.

The benchmarks used among these case-studies differ due to the specific maturity/computing-ability of a target nanotechnology. This is not a limitation because the purpose of this study is not to provide an extensive comparison among nanotechnologies but to show how dedicated logic synthesis can be decisive in generating better implementations for each nanotechnology. For this reason, we encourage the reader in considering intratechnology results and not intertechnology comparisons for this work. We refer the interested reader to [46] for an extensive intertechnology comparison beyond CMOS.

1) *SiNWs-BBDD*: Silicon nanowires are a quite mature emerging nanotechnology. Indeed, they are considered the natural evolution of present FinFETs. Thanks to top-down SiNW maturity and already proved fabrication [4], we can show results down to physical design for this case study.

We focus on controllable-polarity SiNWFETs, realizing natively a comparator functionality within a single device. Our final implementation goal is a complete telecommunication decoder design. We run design experiments with the help of commercial synthesis and physical layout tools. We target the highest speed possibly achievable. To map the target design in a semicustom style, we built a standard cell library containing traditional logic cells (negative unate gates) and comparator-intrinsic logic cells (rich in binate terms) relying on SiNWFETs expressive power. We built the same library also for traditional CMOS technology. Both CMOS and SiNWFET technologies are evaluated at the 22-nm node.

The target design considered in this case-study is an *Iterative Decoder for Product Code* taken from [133]. The original design consists of 8 main modules, among them 2 are sequential, one is the top entity, and 6 are potentially arithmetic intensive. Due to the iterative nature of the decoder, no pipelining is applied in the combinational modules to reduce logic depth. Using BBDDs, we process the 6 arithmetic intensive modules and we keep the BBDD-restructured circuits if their size and depth are decreased as compared to a standard *And/Or-Inv* graph representation. For the sake of clarity, we show an example of BBDD-restructuring for the circuit *bit_comparator*. Fig. 24(a) depicts the logic network before processing (*And/Or-Inv* graph) and Fig. 24(b) illustrates the equivalent

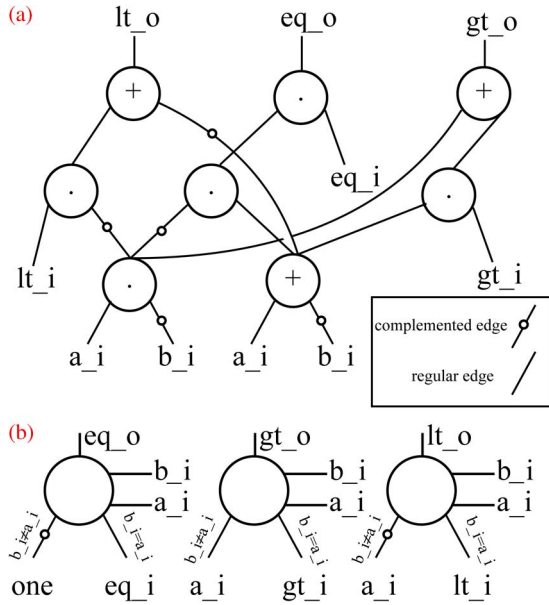


Fig. 24. Representations for the bit comparator circuit in [133] (inverters are bubbles in edges). (a) Original circuit (b) BBDD rewriting, BDD nodes are omitted for the sake of illustration.

circuit after BBDD-restructuring. BDD nodes due to rule R4 are omitted for simplicity. This is because their final logic mapping is just a wire connecting to the corresponding input variable. An advantage in both size and depth is reported. Table 1 shows the remaining results. BBDD-restructuring is favorable for all modules except *ext_val* that instead is more compact in its original version. The representation overhead in *ext_val* happens because BBDDs, being canonical, do not always guarantee a more compact representation than other general logic networks.

Three functionally equivalent versions of the *Iterative Decoder for Product Code* are produced during our experiments. The first one is synthesized, placed and routed in CMOS technology by commercial tools. The second one is

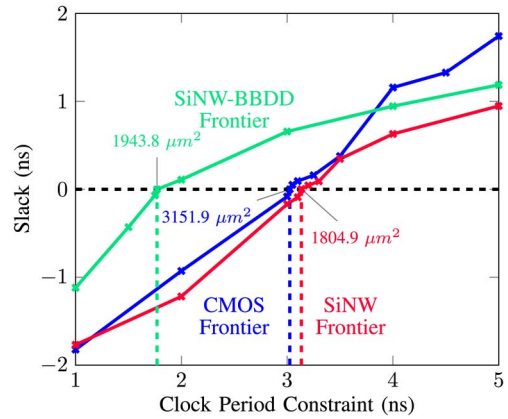


Fig. 25. Target versus obtained frequency curves and frequency frontiers for CMOS, SiNW-standard and SiNW-BBDD designs.

synthesized, placed and routed in SiNWFET technology by commercial tools. The third one is synthesized, placed and routed in SiNWFET technology by BBDD-rewriting as front-end to commercial synthesis and physical design tools.

The maximum clock period is determined by sweeping the clock constraint between 1 ns (1 GHz) and 5 ns (200 MHz) and repeating the implementation process. Fig. 25 shows the postPlace & Route slack vs. target clock constraint curves. Vertical lines highlight the clock constraint barriers for standard-SiNW (red), CMOS (blue) and BBDD-SiNW (green) designs. In the following, we report the shortest clock period achieved.

Table 1 depicts the implementation results, i.e., area, clock frequency and *Energy-Delay Product* (EDP) obtained at the maximum reachable clock frequency. The CMOS design reaches 331 MHz of clock frequency with area occupancy of 4271 μm^2 and EDP of 13.4 nJ.ns. The SiNWFET version, synthesized with plain design tools, has a slower clock frequency of 319 MHz and a larger EDP of 14.2 nJ.ns, but a lower area occupancy of 2473 μm^2 . The final SiNWFET design, synthesized with BBDD-enhanced

Table 1 Experimental Results for SiNWs-BBDD Synthesis

Case Study for Design: <i>Iterative Product Decoder</i>							
Optimization via BBDD-rewriting							
Logic Circuits	Type	I/O		BBDD-rewriting		Original	
		Inputs	Outputs	Nodes	Levels	Nodes	Levels
adder08_bit.vhd	Comb.	16	9	16	8	78	19
bit_comparator.vhd	Comb.	5	3	3	1	8	3
comparator_7bits.vhd	Comb.	14	3	21	7	58	14
fulladder.vhd	Comb.	3	2	2	1	9	4
ext_val.vhd	Comb.	16	8	674	16	173	29
twos_c_8bit.vhd	Comb.	8	8	20	8	29	8
ser2par8bit.vhd	Seq.	11	64	-	-	-	-
product_code.vhd	Top	10	4	-	-	-	-

Post Place and Route results in 22-nm CMOS and 22-nm SiNWFET Technologies								
SiNWFET-BBDD			SiNWFET			CMOS		
Area (μm^2)	Clk (MHz)	EDP (nJ.ns)	Area (μm^2)	Clk (MHz)	EDP (nJ.ns)	Area (μm^2)	Clk (MHz)	EDP (nJ.ns)
2643	565	8.7	2473	315	14.2	4271	331	13.4

synthesis techniques, attains the fastest clock frequency of 565 MHz and the lowest EDP of 8.7 nJ.ns with a small 2643 μm^2 of area occupancy.

If just using a standard synthesis tool suite, SiNWFET technology shows similar performances to CMOS, at the same technology node. This result alone would discard the SiNWFET technology from consideration because it brings no advantage as compared to CMOS. However, the use of BBDD abstraction and synthesis techniques unlock the real value of SiNWFETs, that are indeed capable to produce a faster and more energy efficient realization than CMOS for the *iterative product code decoder*.

2) *Reversible-Circuits-BBDD*: The efficiency of reversible circuits, whether they are finally implemented in quantum computing or other nanotechnologies, strongly depends on the capabilities of reversible synthesis techniques. Due to the inherent complexity of the reversible synthesis problem, several heuristics are proposed in the literature. Among those, the ones based on decision diagrams offer an attractive solution due to scalability and ability to trade-off diverse performance objectives.

Reversible circuit synthesis based on decision diagrams essentially consists of two phases. First, the generation of decision diagrams is geared towards efficient reversible circuit generation. This typically involves nodes minimization or other DD complexity metric reduction. Second, node-wise mapping is performed over a set of reversible gates.

The current standard for DD-based reversible synthesis uses binary decision diagrams generation via existing packages [115] and a custom node-wise mapping. However, standard BDDs do not match the intrinsic functionality of popular reversible gates that are comparator(XOR)-intensive. Instead, BBDDs are based on the biconditional expansion which natively models reversible XOR operations. In this way, BBDDs enable a more compact mapping into common reversible gates, such as Toffoli gates [134]. Fig. 26 depicts the efficient mapping of a single BBDD node into reversible gates. The additional reversible gates w.r.t. a traditional BDD mapping are marked in gray. As one can

notice, two extra gates are required. However, when comparing the functionality of BBDD nodes w.r.t. BDD nodes, it is apparent that more information is encoded into a single BBDD element. This is because the BBDD core expansion examines two variables per time rather than only one. Consequently, the node count reductions deriving from the use of BBDDs overcompensate the slight increase in the direct mapping cost w.r.t. BDDs.

The novel reversible synthesis flow uses BBDD logic representation and minimization using the package [127] and a final one-to-one mapping of BBDD nodes as depicted by Fig. 26. As reference flow, we consider the traditional BDD-based reversible synthesis approach. To validate the BBDD effectiveness, we run synthesis experiments over reversible benchmarks taken from the RevLib online library [135]. In this context, we estimate the implementation cost using the *Quantum-Cost* (QC) [134]. Table 2 shows the reversible synthesis results. Out of 26 benchmark functions studied, 20 reported improved QC and 13 reported improvement in QC as well as line count. A closer study reveals that some benchmark functions, e.g., *plus63mod4096*, contain major contribution from nonlinear subcircuits, which are represented in more compact form by BDD. This translates to better performance in BDD-based synthesis. Nevertheless, future improvement in BBDD construction heuristics may bridge also this gap.

These results provide a fair perspective on the efficacy of BBDDs in reversible synthesis for emerging nanotechnologies.

3) *Nanorelays-BBDD*: Six-terminal nanorelays can realize complex circuit primitives within a single device. For instance, a single nanorelay inherently behaves as a Boolean comparator driving a multiplexer. Such logic expressive power is a key asset to be exploited together with the switching properties of those devices.

To assess the potential of nanorelays in VLSI, a BDD-based synthesis flow has been presented in [15]. It first partitions a design in subblocks and then creates BDDs for those subblocks. For each local BDD, a one-to-one mapping strategy generates a netlist of nanorelays implementing the target logic function. Indeed, the functionality of each BDD node can be realized by a single nanorelay device. We consider this as the reference design flow for nanorelays.

From the analysis we performed in Section III, we know that nanorelays can implement much more complex Boolean functions than just 2:1 multiplexers. Indeed, the functionality of these nanorelays is naturally modeled by a BBDD node. For this reason, we propose a novel design flow based on BBDDs to take full advantage of the nanorelays expressive power. Analogously to the BDD design flow, the design is first prepartitioned if necessary. Then, local BBDDs are built and each BBDD node is mapped into a single nanorelay device.

We first test the BBDD-design flow against the MCNC benchmark suite. Table 3 shows the number of relays and

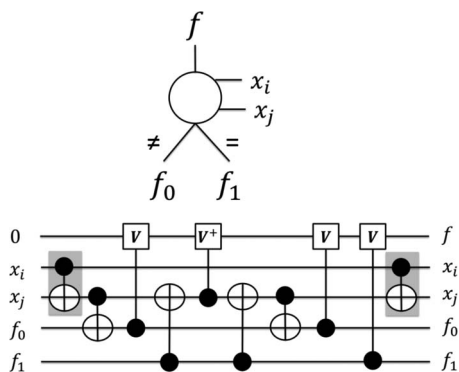


Fig. 26. Reversible circuit for a BBDD node [134].

Table 2 Results for Reversible Circuit Synthesis Using BBDDs Versus Traditional BDDs

Benchmark		BDD			BBDD			Improvement (%)	
Name	I/O	Line	QC	Runtime (in seconds)	Line	QC	Runtime (in seconds)	Line	QC
4mod5_8	4/1	7	24	< 0.01	6	10	0.01	14.28	58.33
decod24_10	2/4	6	27	< 0.01	6	23	0.02	0	14.81
mini-alu_84	4/2	10	60	< 0.01	8	42	0.03	20.00	30.00
alu_9	5/1	7	29	0.01	7	25	0.02	0	13.79
rd53_68	5/3	13	98	< 0.01	13	81	0.03	0	17.34
mod5adder_66	6/6	32	292	< 0.01	32	269	0.05	0	7.76
rd73_69	7/3	13	217	< 0.01	15	117	0.04	-15.38	46.08
rd84_70	8/4	34	304	< 0.01	31	256	0.04	8.82	15.79
sym6_63	6/1	14	93	< 0.01	11	49	0.02	21.43	47.31
sym9_71	9/1	27	206	< 0.01	22	124	0.06	18.52	39.81
cycle10_2_61	12/12	39	202	0.09	25	183	0.03	35.89	9.41
cordic	23/2	52	325	0.06	50	222	0.02	3.84	31.69
bw	5/28	87	943	0.11	78	645	0.03	10.35	31.60
apex2	39/3	498	5922	0.24	744	5242	9.30	-49.39	11.48
seq	41/35	1617	19632	1.14	2440	18366	27.78	-50.89	6.45
spla	16/46	489	5925	0.10	788	5315	1.16	-61.15	10.30
ex5p	8/63	206	1843	0.24	251	1682	1.1	-21.85	8.74
e64	65/65	195	907	0.04	192	826	1.14	1.54	8.93
ham7_29	7/7	21	141	< 0.01	18	153	0.03	14.29	-8.51
ham15_30	15/15	45	309	0.25	43	573	0.06	4.44	-85.44
hwb5_13	5/5	28	276	0.01	30	238	0.02	-7.14	13.77
hwb6_14	6/6	46	507	< 0.01	49	488	0.06	-6.52	3.75
hwb7_15	7/7	73	909	< 0.01	102	978	0.12	-39.73	-7.59
hwb8_64	8/8	112	1461	0.01	189	1831	0.35	-68.75	-25.33
plus63mod4096_79	12/12	23	89	0.08	28	186	0.16	-21.74	-108.99
plus127mod8192_78	13/13	25	98	0.21	31	210	0.02	-24.00	-114.28

the number of relays on the critical path. It compares these numbers with the corresponding numbers in [15] and shows the BBDD to BDD ratio for the different benchmark circuits. We also provide the ratios for the number of relays on the critical path. The BBDD design flow results in an average reduction in NEM relays of 24%. This is due to the compactness of the BBDD representation relative to the BDD representation. Since BBDDs require less nodes than BDDs, BBDD circuits require less NEM relays. Furthermore, the BBDD design flow enables us to obtain circuits with shorter critical paths. On average the critical path length is reduced by 12%. This decrease in the critical paths is due to the BBDD reduction rules, which can be leveraged to decrease the height of BBDDs more than the reduction rules for their BDD counterparts.

Table 3 Total Number of Relays, the Number of Relays on the Critical Path, and Ratios Compared to [15] (MCNC Benchmark Circuits)

Circuit Name	Relays	Levels	R. Ratio [15]	L. Ratio [15]
alu4	599	14	0.77	1.00
apex4	992	8	0.90	0.89
des	3130	18	0.78	1.00
ex1010	1047	10	0.94	0.91
ex5p	283	8	0.92	1.00
misex3	846	14	1.29	1.00
pdv	865	14	0.35	0.88
spla	691	16	0.82	1.00
8-b adder	28	9	0.19	0.53
16-b adder	56	17	0.31	0.52
8 × 8 multiplier	14094	16	1.05	1.00
Average	2057.36	13.09	0.76	0.88

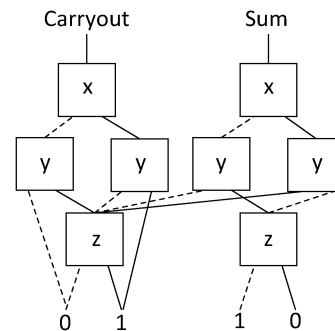


Fig. 27. Nanorelay implementation of a full-adder using a BDD-based design approach [15].

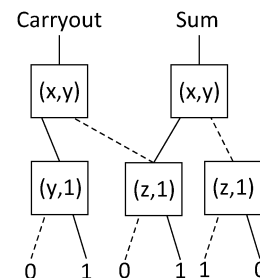


Fig. 28. Nanorelay implementation of a full-adder using a BBDD-based design approach. Dotted lines represent ≠-edges and solid lines are =-edges.

Table 4 Comparison of BDD-Based Versus BBDD-based Synthesis of an 8×8 Array Multiplier

Mech. Delays	BBDD-Relays	BDD-Relays	Ratio BBDD/BDD
2	2491	4129	0.60
3	1367	2186	0.63
4	647	875	0.74
5	434	590	0.74
6	407	533	0.76
Average	1069.2	1662.6	0.64

We also compare the BBDD-approach in the case of synthesizing an 8×8 array multiplier. In [15] the BDD-based approach is tested on such a multiplier implemented using a carry-save adder tree followed by a ripple carry adder. We represent the same multiplier, but using BBDDs instead of BDDs. The main source of advantage here is that BBDDs represent more compactly full-adders and half-adders as compared to BDDs. Figs. 27 and 28 depicts the nanorelay implementation of a full-adder using BDD and BBDD approaches, respectively. Each squared box in the figures represents a six-terminal nanorelay device. We show that this compact representation allows us to implement the multiplier using a smaller number of NEM relays. Table 4 shows the corresponding results. It is possible to see that the BBDD design flow requires a smaller number of relays. On average, reduce the number of relays is reduced by 36% w.r.t. the BDD design flow. Furthermore, as the number of mechanical delays decreases, so does the ratio of the number of relays required by the BBDD representation versus the BDD representation.

These results show the impact of a dedicated logic abstraction to design a comparator-intrinsic nanotechnology, such as nanorelays.

4) *SWD-MIG*: Spin-wave technology enables ultra-low power operation, almost two orders of magnitude lower than state-of-the-art CMOS. However, it has been estimated that the delay performance of SWD will not be adequate to compete with regular CMOS, due to its intrinsically large switching and propagating delays. In order to improve the SWD delay performance, we have to leverage the native logic primitives spin wave logic offer. In SWDs, the logic primitive is a majority voter. Standard synthesis techniques are inadequate to harness this potential. However, the novel MIG data structure previously introduced naturally matches the voting functionality of SWD logic. For this reason, we use MIGs to represent and synthesize SWD circuits. The intrinsic correspondence between MIG elements and SWDs makes MIG optimization naturally extendable to obtain minimum cost SWD implementations.

Table 5 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

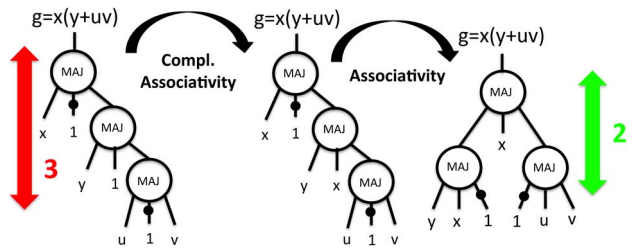


Fig. 29. Optimization of the MIG representing the function $g = x \cdot (y + u \cdot v)$. Initial MIG counts 3 nodes and 3 levels. Final MIG counts 3 nodes and 2 levels.

For this purpose, *ad hoc* cost functions are assigned to MIG elements during optimization as per Table 5. These cost functions are derived from the SWD technology implementation of majority and inverter gates in Fig. 13.

For the sake of clarity, we comment on our proposed MIG-based SWD synthesis flow by means of an example. The objective function in this example is $g = x \cdot (y + u \cdot v)$. This function is initially represented by the MIG in Fig. 29(left), 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. 29(right). 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. 30 depicts the SWD mapping for the original (a) and optimized (b) MIGs.

As one can visually notice, the circuit in Fig. 30(b) features roughly the same area occupation as the one in Fig. 30(a) but shorter input-output path. Following the theoretical cost functions employed, the achieved speed-up is roughly 25%. Including the physical models and assumptions presented in [136], the refined speed-up becomes 18.2%.

We validate hereafter the efficiency of our proposed MIG-based SWD synthesis flow for larger circuits [137]. We also provide a comparison reference to 10-nm CMOS technology.

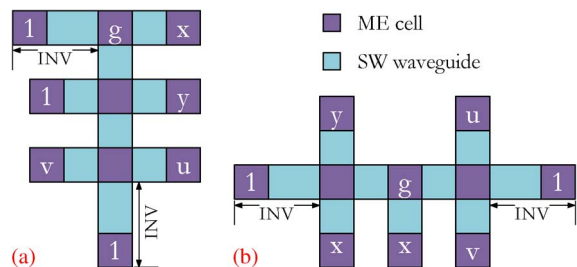


Fig. 30. SWD circuit implementing function g , (a) from example in Fig. 29(left), (b) from example in Fig. 29(right) which is optimized in size and depth.

Table 6 Experimental Results for SWDs-MIG Synthesis

Benchmarks	I/O	SWD technology - MIG			SWD technology - AIG			CMOS Technology - Commercial Tool		
		A (μm^2)	D (ns)	P (μW)	A (μm^2)	D (ns)	P (μW)	A (μm^2)	D (ns)	P (μW)
C1355	41/32	16.95	5.81	0.12	13.88	5.81	0.10	36.27	0.39	68.06
C1908	33/25	16.13	7.30	0.09	12.81	7.9	0.07	32.68	0.53	61.19
C6288	32/32	77.57	26.05	0.12	70.93	28.43	0.11	131.94	1.32	425.21
bigkey	487/421	152.50	3.14	2.11	170.99	3.14	2.34	238.85	0.32	262.50
my_adder	33/17	9.42	6.11	0.07	5.00	10.28	0.04	17.83	0.44	23.94
cla	129/65	36.57	7.60	0.21	32.21	11.77	0.19	72.49	0.62	88.48
dalu	75/16	50.47	6.71	0.31	39.17	9.39	0.25	46.59	0.36	34.63
b9	41/21	5.60	2.24	0.08	5.60	2.54	0.08	5.92	0.09	4.73
count	35/16	6.36	2.54	0.11	4.67	6.11	0.09	8.90	0.32	6.56
alu4	14/8	47.81	4.62	0.42	49.22	4.62	0.43	87.20	0.34	72.39
clma	416/115	433.59	12.96	1.37	450.15	14.15	1.42	231.69	0.51	177.82
mm30a	124/120	41.57	30.52	0.06	35.70	37.66	0.05	68.40	1.68	47.19
s38417	1494/1571	319.86	7.01	1.92	319.86	7.9	1.88	609.94	0.53	740.73
misex3	14/14	45.84	4.33	0.43	44.14	4.62	0.41	78.02	0.26	59.34
Average	212/176	90.02	9.07	0.53	89.60	11.02	0.53	119.05	0.55	148.06

In MIG-based SWD synthesis, we employed the MIGhty MIG optimizer [128]. As traditional-synthesis counterpart, we employed ABC tool [114] 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.

The cost functions for MIG optimization are taken from Table 5. In addition to the direct cost of SWD gates, our design setup takes also into consideration the integration in a VLSI environment given input and output overhead, as presented in [137]. The final synthesis values presented hereafter are comprising all these costs.

Table 6 shows all synthesis results for SWD and CMOS technologies. We summarize in Table 7 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 $1.30\times$ 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 $17.02\times$ smaller ADP, offering an ultra-low power, compact SWD implementation with reduced penalty in delay.

Results showed that MIG synthesis naturally fits SWD technology needs. Indeed, MIGs enhanced SWD strengths (area and power) and alleviated SWD issues (delay).

Table 7 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	$18.45\times$	-
SWD - MIG	432.59	$22.44\times$	$1.22\times$

5) *RRAM-MIG*: The possibility of in-memory computing for RRAM technology can increase the intelligence of many portable electronic devices. However, to fully exploit this opportunity, the primitive Boolean operation in RRAM technology needs to be fully understood and natively handled by design tools.

Previously, we have seen that RRAM elementary devices can realize a majority voting operation on top of storing the corresponding 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 [140] on a RRAM array using MIG-based design techniques [139].

The target cryptography block cipher is PRESENT, originally introduced in [140]. We briefly review its encryption mechanism hereafter.

PRESENT Encryption: A PRESENT encryption consists of 31 rounds, through which multiple operations are performed on the 64-bit plaintext and finally produces a 64-bit ciphertext. The rounds modify the plaintext, which is referred as STATE internally. The operation of the cipher components are *addRoundKey*, *sBoxLayer*, *pLayer*, and *KeyUpdate* [140].

For the sake of brevity, we give here details only on the *sBoxLayer* operation. We refer the interested reader to [140] for details on the other operations. The *sBoxLayer* operation divides the 64-bit word into 16 parts of 4-bit each. Each 4-bit portion is the 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.

S Operator Mapping: 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 . For the sake of brevity, we only represent in Fig. 31 the MIG representation for po_0 that consists of 11 majority nodes. Then, each majority

of 20% and an area improvement of 30% as compared to the standard synthesis flow.

On average over a larger set of circuit benchmarks [141], a majority-based synthesis flow improves reconfigurable graphene implementations by 17% in area and 8.17% in performance, at the same time.

These results confirm that also graphene technology needs *ad-hoc* logic synthesis techniques to unlock its true potential.

VI. DISCUSSION

Emerging technologies are expected to push the computational performance of digital systems beyond the limits of CMOS. While the advantage of some post-CMOS candidate is obvious at the device characteristics level, e.g., speed, area and energy, other technologies need intelligent design methods to extract a less evident, but not smaller, potential. This is the case for functionality-enhanced devices that might be bulkier than competitors but realize complex functions with fewer physical resources. Without exploiting their increased expressive power, many of these devices can be prematurely discarded from consideration. In this scenario, logic synthesis is asked to make the best use possible of a device functionality, to unlock its true value. We have seen that, using standard design methods, emerging technologies like SiNWFETs, nanorelays, reversible logic, RRAM, graphene and spin-wave devices do not reach their full potential and may not be competitive to CMOS. However, this may give wrong assessment on an emerging nanotechnology. For instance, specialized synthesis techniques enable faster SiNWFET circuits than in CMOS, e.g., reaching a clock frequency of 565 MHz (SiNW) rather than just 331 MHz (CMOS). This

possibility was not evident if just using standard design methods for SiNWFET circuits.

Our results already demonstrate that the attention of logic synthesis in emerging nanotechnologies is vital to achieve a fair evaluation. In a general sense, dedicated synthesis approaches do not just support post-CMOS candidates but also enable their rise.

VII. CONCLUSION

In this work, we investigated the relation between EDA and emerging nanotechnologies, mainly from a logic synthesis standpoint. We reviewed a class of promising nanodevices whose logic abstraction is either a Boolean comparator or a majority voter. We demonstrated that new logic synthesis techniques, natively supporting these abstractions, are the technology enablers as they allow designers to validate nanotechnologies on large-scale benchmarks. Even though our experiments focused on a limited class of nanodevices and design benchmarks, the obtained results bring up a broader point. New research in logic synthesis is becoming essential to permit a fair evaluation on nanotechnologies with different logic abstractions than standard CMOS. ■

Acknowledgment

The authors would like to thank their scientific collaborators for helping generating the experimental results, in particular: Mr. O. Zografos and the IMEC team, Prof. A. Chattopadhyay and the NTU team, Dr. S. Miryala and the PoliTO team, Dr. E. Linn and the RWTH team, and Mr. W. J. Haaswijk. The authors would also like to thank Prof. A. Burg and Prof. H.-S. Philip Wong for valuable discussions.

REFERENCES

- [1] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proc. Roy. Soc. London. A. Math. Phys. Sci.*, vol. 400, no. 1818, pp. 97–117, 1985.
- [2] E. Farhi et al., *Quantum Computation by Adiabatic Evolution*, 2000, arXiv preprint quant-ph/0001106.
- [3] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science and Engineering*. New York, NY, USA: McGraw-Hill Higher Education, 2000.
- [4] M. De Marchi et al., "Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs," presented at the IEDM, 2012.
- [5] M. De Marchi et al., "Configurable logic gates using polarity controlled silicon nanowire gate-all-around FETs," *IEEE Electron Device Lett.*, vol. 35, no. 8, pp. 880–882, Aug. 2014.
- [6] A. Heinzig, S. Slesazek, F. Kreupl, T. Mikolajick, and W. M. Weber, "Reconfigurable silicon nanowire transistors," *Nano Lett.*, vol. 12, pp. 119–124, 2011.
- [7] T. Ernst, "Controlling the polarity of silicon nanowire transistors," *Science*, vol. 340, p. 1414, 2013.
- [8] Y.-M. Lin et al., "High-performance carbon nanotube field-effect transistor with tunable polarities," *IEEE Trans. Nanotechnol.*, vol. 4, no. 5, pp. 481–489, May 2005.
- [9] J. Appenzeller, "Carbon nanotubes for high-performance electronics—Progress and prospect," *Proc. IEEE*, vol. 96, no. 2, pp. 201–211, Feb. 2008.
- [10] H. Yang et al., "Graphene barristor, a triode device with a gate-controlled schottky barrier," *Science*, vol. 336, p. 1140, 2012.
- [11] F. Schwierz, "Graphene transistors," *Nature Nanotechnol.*, vol. 5, no. 7, pp. 487–496, 2010.
- [12] S.-L. Li et al., "Complementary like graphene logic gates controlled by electrostatic doping," *Small*, vol. 7, no. 11, pp. 1552–1556, 2011.
- [13] N. Harada, K. Yagi, S. Sato, and N. Yokoyama, "A polarity-controllable graphene inverter," *Appl. Phys. Lett.*, vol. 96, 2010, Art. ID. 012102, DOI: 10.1063/1.3280042.
- [14] M. F. Craciun, S. Russo, M. Yamamoto, and S. Tarucha, "Tuneable electronic properties in graphene," *Nano Today*, vol. 6, no. 1, pp. 42–60, 2011.
- [15] D. Lee et al., "Combinational logic design using six-terminal NEM relays," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 5, pp. 653–666, May 2013.
- [16] M. Spencer et al., "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI applications," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 308–320, Jan. 2011.
- [17] R. Fackenthal et al., "16 Gb ReRAM with 200 MB/s Write and 1 GB/s Read in 27 nm Technology," presented at the ISSCC Tech. Dig., 2014.
- [18] S.-S. Sheu et al., "4 Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160 ns MLC-access capability," presented at the ISSCC Tech. Dig., 2011.
- [19] G. W. Burr et al., "Overview of candidate device technologies for storage-class-memory," *IBM J. Res. Develop.*, vol. 52, no. 4/5, pp. 449–464, 2008.
- [20] R. Fackenthal et al., "A 16 Gb ReRAM with 200 MB/s write and 1 GB/s read in 27 nm technology," presented at the ISSCC Tech. Dig., 2014.

- [21] S.-S. Sheu et al., "A 4 Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160 ns MLC-access capability," presented at the ISSCC Tech. Dig., 2011.
- [22] H.-S. P. Wong et al., "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.
- [23] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Mater.*, vol. 9, no. 5, pp. 403–406, 2010.
- [24] E. Linn, R. Rosezin, S. Tappertzshofen, U. Böttger, and R. Waser, "Beyond von Neumann-logic operations in passive crossbar arrays alongside memory operations," *Nanotechnology*, vol. 23, 2012, Art. ID. 305205.
- [25] S. Iba et al., "Control of threshold voltage of organic field-effect transistors with double-gate structures," *Appl. Phys. Lett.*, vol. 87, no. 2, 2005, Art. ID. 023509.
- [26] T. Anthopoulos et al., "Ambipolar organic field-effect transistors based on a solution-processed methanofullerene," *Adv. Mater.*, vol. 16, no. 2324, pp. 2174–2179, 2004.
- [27] E. J. Meijer et al., "Solution-processed ambipolar organic field-effect transistors and inverters," *Nature Mater.*, vol. 2, no. 10, pp. 678–682, 2003.
- [28] T. B. Singh et al., "Organic inverter circuits employing ambipolar pentacene field-effect transistors," *Appl. Phys. Lett.*, vol. 89, no. 3, 2006, Art. ID. 033512.
- [29] S.-Y. Min et al., "Large-scale organic nanowire lithography and electronics," *Nature Commun.*, vol. 4, p. 1773, 2013.
- [30] J. Cornil et al., "Ambipolar transport in organic conjugated materials," *Adv. Mater.*, vol. 19, no. 14, pp. 1791–1799, 2007.
- [31] M.-H. Yoon et al., "Organic thin-film transistors based on carbonyl-functionalized quaterthiophenes: High mobility n-channel semiconductors and ambipolar transport," *J. Amer. Chem. Soc.*, vol. 127, no. 5, pp. 1348–1349, 2005.
- [32] T. Schneider et al., "Realization of spin-wave logic gates," *Appl. Phys. Lett.*, vol. 92, no. 2, 2008, Art. ID. 022505.
- [33] A. Khitun and K. L. Wang, "Nano scale computational architectures with Spin Wave Bus," *Superlattices Microstruct.*, vol. 38, no. 3, pp. 184–200, 2005.
- [34] A. Khitun et al., "Non-volatile magnonic logic circuits engineering," *J. Appl. Phys.*, vol. 110, Aug. 2011, Art. ID. 034306.
- [35] D. E. Nikonov et al., "Overview of beyond-CMOS devices and a uniform methodology for their benchmarking," *Proc. IEEE*, vol. 101, no. 12, pp. 2498–2533, Dec. 2013.
- [36] I. Amlani et al., "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, no. 5412, pp. 289–291, 1999.
- [37] A. Imre et al., "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.
- [38] C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Develop.*, vol. 17, no. 6, pp. 525–532, 1973.
- [39] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [40] R. Cuykendall and D. R. Andersen, "Reversible optical computing circuits," *Optics Lett.*, vol. 12, no. 7, pp. 542–544, 1987.
- [41] R. C. Merkle, "Reversible electronic logic using switches," *Nanotechnology*, vol. 4, pp. 21–40, 1993.
- [42] A. Barenco et al., "Elementary gates for quantum computation," *Phys. Rev.*, vol. 52, no. 5, pp. 3457–3467, 1995.
- [43] P. Ranjit, "Molecule for electronics: A myriad of opportunities comes with daunting challenges," *J. Nanomater. Molecular Nanotechnol.*, vol. 1, no. 1, 2012.
- [44] A. Bandyopadhyay et al., "Massively parallel computing on an organic molecular layer," *Nature Phys.*, vol. 6, no. 5, pp. 369–375, 2010.
- [45] S. R. Kumar and P. Chatterje, "Approximate reasoning on a DNA-chip," *Int. J. Intell. Comput. ybern.*, vol. 3, no. 3, pp. 514–553, 2010.
- [46] K. Bernstein et al., "Device and architecture outlook for beyond CMOS switches," *Proc. IEEE*, vol. 98, no. 12, pp. 2169–2184, Dec. 2010.
- [47] S. D. Suk et al., "High performance 5 nm radius twin silicon nanowire mosfet (tsnwfet): Fabrication on bulk si wafer, characteristics, reliability," presented at the IEDM Tech. Dig., 2005.
- [48] S. Bangsaruntip et al., "High performance and highly uniform gate-all-around silicon nanowire MOSFETs with wire size dependent scaling," presented at the IEDM Tech. Dig., 2009.
- [49] S. Bobba et al., "Design of compact imperfection-immune CNFET layouts for standard-cell-based logic synthesis," in *Proc. Design Autom. Test Eur. (DATE)*, 2009, pp. 616–621.
- [50] S. Mitra et al., "Imperfection-immune VLSI logic circuits using carbon nanotube field effect transistors," in *Proc. Design Autom. Test Eur. DATE*, 2009, pp. 436–441.
- [51] N. Patil et al., "Digital VLSI logic technology using carbon nanotube FETs: Frequently asked questions," in *Proc. 46th Annu. Design Autom. Conf.*, 2009, pp. 304–309.
- [52] N. Patil et al., "Wafer-scale growth and transfer of aligned single-walled carbon nanotubes," *IEEE Trans. Nanotechnol.*, vol. 8, no. 4, pp. 498–504, Jul. 2009.
- [53] H. Wei et al., "Carbon nanotube imperfection-immune digital VLSI: Frequently asked questions updated," in *Proc. Int. Conf. Comput.-Aided Design*, 2011, pp. 227–230.
- [54] J. Zhang et al., "Robust digital VLSI using carbon nanotubes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 4, pp. 453–471, Apr. 2012.
- [55] S. Bobba et al., "Design of compact imperfection-immune CNFET layouts for standard-cell-based logic synthesis," presented at the Conf. Design, Automation and Test Eur., Eur. Design Autom. Assoc., 2009.
- [56] S. K. Bobba, "Design Methodologies and CAD for Emerging Nanotechnologies," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2013.
- [57] S. Bobba et al., "System-level benchmarking with yield-enhanced standard cell library for carbon nanotube VLSI circuits," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 4, p. 33, 2014.
- [58] J. Zhang, L. Wei, N. Patil, A. Lin, H. Wei, H.-S. P. Wong, and S. Mitra, "Robust digital VLSI using carbon nanotubes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. Keynote Paper*, vol. 31, no. 4, pp. 453–471, 2012.
- [59] O. Loh and H. Espinosa, "Nanoelectromechanical contact switches," *Nature Nanotechnol.*, vol. 7, no. 5, pp. 283–295, 2012.
- [60] Nano-Electro-Mechanical Switches, ITRS, white paper, 2008.
- [61] V. Pott et al., "Mechanical computing redux: Relays for integrated circuit applications," *Proc. IEEE*, vol. 98, no. 12, pp. 2076–2094, Dec. 2010.
- [62] P. Sharma, J. Perruisseau-Carrier, C. Moldovan, and A. Ionescu, "Electromagnetic performance of RF NEMS graphene capacitive switches," *IEEE Trans. Nanotech.*, vol. 13, no. 1, pp. 70–79, Jan. 2014.
- [63] D. Weinstein and S. A. Bhawe, "The resonant body transistor," *Nano Lett.*, vol. 10, no. 4, pp. 1234–1237, 2010.
- [64] M. Shulaker et al., "Carbon nanotube computer," *Nature*, vol. 501, no. 7468, pp. 526–530, 2013.
- [65] M. Shulaker et al., "High-performance carbon nanotube field-effect transistors," presented at the IEEE Int. Electron Devices Meeting, San Francisco, CA, USA, Dec. 2014.
- [66] M. Shulaker et al., "Monolithic 3D integration of carbon nanotube FETs, resistive RAM, silicon FETs," presented at the IEEE Int. Electron Devices Meeting, San Francisco, CA, USA, Dec. 2014.
- [67] M. Shulaker et al., "Experimental demonstration of a fully digital capacitive sensor interface built entirely using carbon nanotube FETs," in *Proc. Int. Solid State Circuits Conf.*, 2013, pp. 112–113.
- [68] M. M. Shulaker et al., "Sensor-to-digital interface built entirely with carbon nanotube FETs," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 190–201, 2014.
- [69] K. Jeong, A. B. Kahng, and K. Samadi, "Impacts of guardband reduction on design process outcomes: A quantitative approach," *IEEE Trans. Semicond. Manuf.*, vol. 22, no. 4, pp. 552–565, Nov. 2009.
- [70] P. Gupta et al., "Underdesigned and opportunistic computing in presence of hardware variability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 8–23, Jan. 2013.
- [71] J. Zhang, N. Patil, and S. Mitra, "Probabilistic analysis and design of metallic-carbon-nanotube-tolerant digital logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 9, pp. 1307–1320, Sep. 2009.
- [72] J. Zhang et al., "Carbon nanotube circuits in the presence of carbon nanotube density variations," presented at the 46th Annu. Design Autom. Conf., 2009.
- [73] J. Zhang et al., "Carbon nanotube correlation: Promising opportunity for CNFET circuit yield enhancement," presented at the 47th Annu. Design Autom. Conf., 2010.
- [74] H. Wei et al., "Efficient metallic carbon nanotube removal readily scalable to wafer-level VLSI CNFET circuits," presented at the Symp. VLSI Technol., Honolulu, HI, USA, Jun. 2010.
- [75] G. Hills et al., "Rapid exploration of processing and design guidelines to overcome carbon nanotube variations," presented at the 50th Annu. Design Autom. Conf., 2013.
- [76] G. Hills et al., "Rapid co-optimization of processing and circuit design to overcome carbon nanotube variations," *IEEE Trans.*

- Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1082–1095, Jul. 2015.
- [77] A. Heinzig, S. Slesazek, F. Kreupl, T. Mikolajick, and W. M. Weber, “Reconfigurable silicon nanowire transistors,” *Nano Lett.*, vol. 12, pp. 119–124, 2011.
- [78] J. Xiang et al., “Ge/Si nanowire heterostructures as high-performance field-effect transistors,” *Nature*, vol. 441, pp. 489–493, 2006.
- [79] G. Yu and C. M. Lieber, “Assembly and integration of semiconductor nanowires for functional nanosystems,” *Pure Appl. Chem.*, vol. 82, pp. 2295–2314, 2010.
- [80] H. Yan et al., “Programmable nanowire circuits for nanoprocessors,” *Nature*, vol. 470, pp. 240–244, 2011.
- [81] C. Dupre et al., “15 nm-diameter 3D stacked nanowires with independent gates operation,” presented at the IEDM Tech. Dig., 2008.
- [82] L. K. Bera et al., “Three dimensionally stacked SiGe nanowire array and gate-all-around p-MOSFETs,” presented at the IEDM Tech. Dig., 2006.
- [83] W. W. Fang et al., “Vertically stacked silicon nanowire array channel CMOS transistors,” *IEEE Elec. Dev. Lett.*, vol. 28, no. 3, pp. 211–213, Mar. 2007.
- [84] B. Radisavljevic, A. Radenovic, J. Brivio, V. Giacometti, and A. Kis, “Single-layer MoS₂ transistors,” *Nature Nanotechnol.*, vol. 6, p. 147, 2011.
- [85] W. Xin-Ran et al., “Field-effect transistors based on two-dimensional materials for logic applications,” *Chinese Phys. B*, vol. 22, no. 9, 2013, Art. ID. 098505.
- [86] S. Nakaharai et al., “Electrostatically-reversible polarity of dual-gated graphene transistors with the ion irradiated channel; toward reconfigurable CMOS applications,” in *Proc. Techn. Digest 2012 IEEE Int. Electron Devices Meeting (IEDM)*, 2012, p. 72.
- [87] V. V. Cheianov, V. Falko, and B. L. Altshuler, “The focusing of electron flow and a veselago lens in graphene p-n junctions,” *Science*, vol. 315, no. 5816, pp. 1252–1255, 2007.
- [88] S. Tanachutiwat, J. U. Lee, W. Wang, and C. Y. Sung, “Reconfigurable multi-function logic based on graphene p-n junctions,” in *Proc. Design Autom. Conf. (DAC)*, Jun. 2010, pp. 883–888.
- [89] C.-Y. Sung and J. U. Lee, “Graphene: The ultimate switch,” *IEEE Spectrum*, vol. 49, no. 2, pp. 32–59, Feb. 2012.
- [90] R. Saito, G. Dresselhaus, and M. Dresselhaus, *Physical Properties of Carbon Nanotubes*. London, U.K.: Imperial College Press, 1998.
- [91] S. J. Tans et al., “Room-temperature transistor based on a single carbon nanotube,” *Nature*, vol. 393, no. 49, 1998.
- [92] A. Bachtold, “Logic circuits with carbon nanotube transistors,” *Science*, vol. 294, no. 5545, pp. 1317–1320, 2001.
- [93] A. D. Franklin et al., “Sub-10 nm carbon nanotube transistor,” *Nano Lett.*, vol. 12, no. 2, pp. 758–762, 2012.
- [94] L. Wei, D. J. Frank, L. Chang, and H.-S. P. Wong, “A non-iterative compact model for carbon nanotube FETs incorporating source exhaustion effects,” in *Proc. 2009 IEEE Int. Electron Devices Meeting*, 2009, pp. 917–920.
- [95] L. Chang, “Technology optimization for high energy-efficiency computation,” in *Short Course IEEE Intl Electron Devices Meeting (IEEE, 2012)*.
- [96] E. J. McCluskey, “Minimization of boolean functions,” *Bell Syst. Techn. J.*, vol. 35, no. 6, pp. 1417–1444, 1956.
- [97] C. Y. Lee, “Representation of switching circuits by binary-decision programs,” *Bell Syst. Techn. J.*, vol. 38, no. 4, pp. 985–999, 1959.
- [98] S. B. Akers, “Binary decision diagrams,” *IEEE Trans. Comput.*, vol. C-27, no. 6, pp. 509–516, Jun. 1978.
- [99] R. E. Bryant, “Graph-based algorithms for Boolean function manipulation,” *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.
- [100] C. E. Leiserson and J. B. Saxe, “Optimizing synchronous systems,” *J. VLSI Comput. Syst.*, vol. 1, no. 1, pp. 41–67, Spring, 1983.
- [101] K. Keutzer, “Kurt, DAGON: Technology binding and local optimization by DAG matching,” presented at the Design Autom. Conf. (DAC), 1987.
- [102] L. Benini and G. De Micheli, “A survey of Boolean matching techniques for library binding,” in *Proc. ACM TODAES*, Jul. 1997, vol. 2, no. 3, pp. 193–226.
- [103] O. Coudert, C. Berthet, and J. C. Madre, “Verification of synchronous sequential machines based on symbolic execution,” in *Proc. Workshop Autom. Verification Methods Finite State Mach.*, 1989, vol. 407, pp. 365–373.
- [104] R. L. Rudell and A. Sangiovanni-Vincentelli, “Multiple-valued minimization for PLA optimization,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 5, pp. 727–750, Jun. 1987.
- [105] R. K. Brayton et al., “MIS: A multiple-level logic optimization system,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 6, pp. 1062–1081, Jun. 1987.
- [106] R. K. Brayton et al., “The yorktown silicon compiler,” presented at the ISCAS, 1985.
- [107] M. Damiani and G. DeMicheli, “Observability don't care sets and boolean relations,” in *Proc. IEEE Int. Conf. Comput.-Aided Design*, 1990, pp. 502–505.
- [108] H. Savoj et al., “Extracting local don't cares for network optimization,” presented at the ICCAD, 1991.
- [109] S. Malik et al., “Symbolic minimization of multilevel logic and the input encoding problem,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 11, no. 7, pp. 825–843, Jul. 1992.
- [110] G. De Micheli, “Symbolic design of combinational and sequential logic circuits implemented by two-level logic macros,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 5, no. 4, pp. 597–616, Apr. 1986.
- [111] E. Sentovich et al., “SIS: A system for sequential circuit synthesis,” ERL, Dept. EECS, Univ. California, Berkeley, CA, USA, UCB/ERL M92/41, 1992.
- [112] C. Yang and M. Ciesielski, “BDS: A BDD-based logic optimization system,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 7, pp. 866–876, Jul. 2002.
- [113] N. Vemuri, P. Kalla, and R. Tessier, “BDD-based logic synthesis for LUT-based,” *FPGAs, ACM Trans. TODAES*, vol. 7, pp. 501–525, Oct. 2002.
- [114] Berkeley Logic Synthesis and Verification Group, “ABC: A System for Sequential Synthesis and Verification,” *ABC Synthesis Tool*. [Online]. Available: <http://www.eecs.berkeley.edu/~alanmi/abc/>
- [115] CUDD, CU Decision Diagram Package Release 2.5.0. [Online]. Available: <http://vlsi.colorado.edu/~fabio/CUDD/>
- [116] A. Mishchenko et al., “Using simulation and satisfiability to compute flexibilities in Boolean networks,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 5, pp. 743–755, May 2006.
- [117] V. Bertacco and M. Damiani, “The disjunctive decomposition of logic functions,” presented at the ICCAD, 1997.
- [118] S. Plaza and V. Bertacco, “STACCATO: Disjoint support decompositions from BDDs through symbolic kernels,” presented at the ASPDAC, 2005.
- [119] R. Brayton and A. Mishchenko, “ABC: An academic industrial-strength verification tool,” presented at the CAV, 2010.
- [120] G. De Micheli, *Synthesis and Optimization of Digital Circuits*. New York, NY, USA: McGraw-Hill Higher Education, 1994.
- [121] K. S. Brace, R. L. Rudell, and R. E. Bryant, “Efficient implementation of a BDD package,” presented at the Design Autom. Conf. (DAC), 1990.
- [122] R. Rudell, “Dynamic variable ordering for ordered binary decision diagrams,” presented at the ICCAD, 1993.
- [123] R. Drechsler et al., “Ordered Kronecker functional decision diagrams a data structure for representation and manipulation of Boolean functions,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 10, pp. 965–973, Oct. 1998.
- [124] L. Amaru, P.-E. Gaillardon, and G. De Micheli, “An efficient manipulation package for biconditional binary decision diagrams,” presented at the DATE, 2014.
- [125] L. Amaru, P.-E. Gaillardon, and G. De Micheli, “Biconditional BDD: A novel canonical representation form targeting the synthesis of XOR-rich circuits,” presented at the DATE, 2013.
- [126] L. Amaru, P.-E. Gaillardon, and G. De Micheli, “Biconditional binary decision diagrams: A novel canonical representation form,” *IEEE J. Emerg. Sel. Topics Circuits Syst. (JETCAS)*, vol. 4, no. 4, Dec. 2014.
- [127] BBDD Package. [Online]. Available: <http://lsi.epfl.ch/BBDD>
- [128] L. Amaru, P.-E. Gaillardon, and G. De Micheli, “Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization,” presented at the Design Autom. Conf. (DAC), 2014.
- [129] L. Amaru, P.-E. Gaillardon, and G. De Micheli, “Boolean logic optimization in majority-inverter graphs,” presented at the Design Autom. Conf. (DAC), 2015.
- [130] T. Sasao, *Switching Theory for Logic Synthesis*. Berlin, Germany: Springer-Verlag, 1999.
- [131] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Feb. 1996.
- [132] A. Picart and R. Pyndiah, “Adapted iterative decoding of product codes,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'99)*, 1999, vol. 5.
- [133] An Iterative Decoder for Product Code—From Open Cores. [Online]. Available: http://opencores.org/project_product_code_iterative_decoder
- [134] A. Chattopadhyay et al., “Reversible logic synthesis via biconditional binary decision diagrams,” in *Proc. ISMVL*, vol. 15.

- [135] RevLib is an Online Resource for Benchmarks Within the Domain of Reversible and Quantum Circuit Design. [Online]. Available: <http://www.revlib.org>
- [136] O. Zografos *et al.*, "System-level assessment and area evaluation of Spin Wave logic circuits," presented at the IEEE/ACM Int. Symp. Nanoscale Architectures (NANOARCH), 2014.
- [137] O. Zografos *et al.*, "Majority logic synthesis for spin wave technology," presented at the IEEE 17th Euromicro Conf. Digital Syst. Design (DSD), 2014.
- [138] L. Amar, G. Hills, P.-E. Gaillardon, S. Mitra, and G. De Micheli, "Multiple independent gate FETs: How many gates do we need?," presented at the Asia South Pacific Design Autom. Conf. (ASP-DAC 2015), Chiba/Tokyo, Japan, 2015.
- [139] P.-E. Gaillardon *et al.*, "Computing secrets on a resistive memory array," presented at the Design Autom. Conf. (DAC), San Francisco, CA, USA, 2015.
- [140] A. Bogdanov *et al.*, "PRESENT: An ultra-lightweight block cipher," presented at the CHES Tech. Dig., 2007.
- [141] S. Miryala *et al.*, "Exploiting the expressive power of graphene reconfigurable gates via post-synthesis optimization," presented at the GLVSLI'15.
- [142] V. Tenace, A. Calimera, E. Macii, and M. Poncino, "One-pass logic synthesis for graphene-based Pass-XNOR logic circuits," presented at the 52nd Annu. Design Autom. Conf. (DAC), 2015.
- [143] S. Klingler *et al.*, "Design of a spin-wave majority gate employing mode selection," *Appl. Phys. Lett.*, vol. 105, no. 15, 2014, Art. ID. 152410.
- [144] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Reversible logic gate using adiabatic superconducting devices," *Sci. Rep.*, vol. 4, p. 6354, 2014.
- [145] A. Fedorov *et al.*, "Implementation of a Toffoli gate with superconducting circuits," *Nature*, vol. 481, no. 7380, pp. 170–172, 2011.
- [146] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, W. de Bakker and J. van Leeuwen, Eds. Berlin, Germany: Springer-Verlag, 1980, p. 632, technical memo MIT/LCS/TM-151, MIT Lab. for Comput. Sci.
- [147] E. F. Fredkin and T. Toffoli, "Conservative logic," *Int. J. Theoret. Phys.*, vol. 21, no. 3/4, 1982, Art. ID. 219253.
- [148] A. Peres, "Reversible logic and quantum computers," *Phys. Rev. A*, no. 32, pp. 3266–3276, 1985.
- [149] W. Li *et al.*, "Three-input majority logic gate and multiple input logic circuit based on DNA strand displacement," *Nano Lett.*, vol. 13, no. 6, pp. 2980–2988, 2013.
- [150] D. Nikonov and I. Young, "Benchmarking of beyond-CMOS exploratory devices for logic integrated circuits," *IEEE J. Exploratory Solid-State Comput. Devices Circuits*, vol. 1, pp. 3–11, Dec. 2015.
- [151] J. Kim *et al.*, "Spin-based computing: Device concepts, current status, a case study on a high-performance microprocessor," *Proc. IEEE*, vol. 103, no. 1, pp. 106–130, Jan. 2015.

ABOUT THE AUTHORS

Luca Amarú (Student Member, IEEE) received the B.S. degree in electronic engineering from the Politecnico di Torino, Torino, Italy, in 2009. In 2011, he received the joint M.S. degree in electronic engineering from the Politecnico di Torino and the Politecnico di Milano, Milano, Italy. Since 2011, he has been working toward the Ph.D. degree in computer science at the EPFL, Lausanne, Switzerland.

In 2014, he was a Visiting Researcher at Stanford University, Palo Alto, CA, USA. His research interests include electronic design automation, logic synthesis, formal verification and beyond CMOS technologies.

Mr. Amarú received the Best Presentation Award at the FETC 2013 Conference and a Best Paper Award Nomination at ASP-DAC 2013 conference. He has been serving as a TPC Member for DSD'14-15 conferences and is Reviewer for several IEEE journals.



Pierre-Emmanuel Gaillardon (Member, IEEE) received the B.Eng. degree from CPE-Lyon, France, in 2008, the M.Sc. degree in electrical engineering from INSA Lyon, France, in 2008, and the Ph.D. degree in electrical engineering from CEA-LETI, Grenoble, France, and the University of Lyon, France (2011).

He works as a Research Associate at the Laboratory of Integrated Systems (LSI), for the EPFL, Lausanne, Switzerland. Starting in January 2016, he will assume an Assistant Professor position within the Electrical and Computer Engineering (ECE) Department at The University of Utah, Salt Lake City, UT, USA. Previously, he was a Research Assistant at CEA-LETI, Grenoble, France, and a Visiting Research Associate at Stanford University, Palo Alto, CA, USA. His research interests include the development of reconfigurable logic architectures and circuits exploiting emerging device technologies and novel EDA techniques.

Dr. Gaillardon is recipient of the C-Innov 2011 Best Thesis Award and the Nanoarch 2012 Best Paper Award. He is an Associate Editor of the IEEE TRANSACTIONS ON NANOTECHNOLOGY. He has been serving as a TPC Member for many conferences, including DATE'15-16, VLSI-Soc'15, CMOS-ETR'13-15, Nanoarch'12-15, and ISVLSI'14-15 conferences, and is a Reviewer for several journals and funding agencies.



Subhasish Mitra (Fellow, IEEE) received the Ph.D. in electrical engineering from Stanford University, Palo Alto, CA, USA.

He directs the Robust Systems Group in the Department of Electrical Engineering and the Department of Computer Science of Stanford University, where he is the Chambers Faculty Scholar of Engineering. Before joining Stanford, he was a Principal Engineer at Intel. His research interests include robust systems, VLSI design, CAD, validation and test, emerging nanotechnologies, and emerging neuroscience applications. His X-Compact technique for test compression has been key to cost-effective manufacturing and high-quality testing of a vast majority of electronic systems, including numerous Intel products. X-Compact and its derivatives have been implemented in widely-used commercial Electronic Design Automation tools. His work on carbon nanotube imperfection-immune digital VLSI, jointly with his students and collaborators, resulted in the demonstration of the first carbon nanotube computer, and it was featured on the cover of NATURE. The US National Science Foundation presented this work as a Research Highlight to the US Congress, and it also was highlighted as "an important, scientific breakthrough" by the BBC, Economist, EE Times, IEEE Spectrum, MIT Technology Review, National Public Radio, New York Times, Scientific American, Time, Wall Street Journal, Washington Post, and numerous other organizations worldwide.

Dr. Mitra received the Presidential Early Career Award for Scientists and Engineers from the White House, the highest U.S. honor for early-career outstanding scientists and engineers, the ACM SIGDA/IEEE CEDA A. Richard Newton Technical Impact Award in Electronic Design Automation, "a test of time honor" for an outstanding technical contribution, the Semiconductor Research Corporation's Technical Excellence Award, and the Intel Achievement Award, Intel's highest corporate honor. He and his students published several award-winning papers at major venues: the IEEE/ACM Design Automation Conference, the IEEE International Solid-State Circuits Conference, the IEEE International Test Conference, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE VLSI Test Symposium, the Intel Design and Test Technology Conference, and the Symposium on VLSI Technology. At Stanford, he has been honored several times by graduating seniors "for being important to them during their time at Stanford." He has served on numerous conference committees and journal editorial boards. He served on DARPA's Information Science and Technology Board as an invited member. He is a Fellow of the ACM.



Giovanni De Micheli (Fellow, IEEE) received the Nuclear Eng. degree from the Politecnico di Milano, Milano, Italy, in 1979, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1980 and 1983, respectively.

He is a Professor and the Director of the Institute of Electrical Engineering and of the Integrated Systems Centre at EPF Lausanne, Lausanne, Switzerland. He is a Program Leader of the Nano-Tera.ch program. His research interests include several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips and 3-D integration. He is also interested in heterogeneous platform design including electrical components and biosensors, as well as in data processing of biomedical information. He is author of "Synthesis and Optimization of Digital Circuits," (McGraw-Hill, 1994), coauthor and/or coeditor of eight other books and of more than 500 technical articles. His citation h-index is 85 according to Google Scholar.

He is a Fellow of ACM and a Member of the Academia Europaea. He is Member of the Scientific Advisory Board of IMEC and STMicroelectronics.



He is the recipient of the 2012 IEEE/CAS Mac Van Valkenburg award for contributions to theory, practice, and experimentation in design methods and tools and of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He received also the Golden Jubilee Medal for outstanding contributions to the IEEE Circuits and Systems Society in 2000, the D. Pederson Award for the best paper on the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN (CAD)/INTEGRATED CIRCUITS AND SYSTEMS (ICAS), in 1987, and several Best Paper Awards, including Design Automation Conference (DAC) in 1983 and 1993, Design, Automation, and Test in Europe (DATE) in 2005, and Nanoscale Architectures in 2010 and 2012. He has been serving IEEE in several capacities, namely: Division 1 Director during 2008–2009, the CoFounder and President Elect of the IEEE Council on Electronic Design Automation during 2005–2007, the President of the IEEE CAS Society in 2003, an Editor-in-Chief of the IEEE TRANSACTIONS ON CAD/ICAS from 1987 to 2001. He has been the Chair of several conferences, including Memocode in 2014, DATE in 2010, Public Health Conferences in 2006, IEEE International Conference on Very Large Scale Integration in 2006, DAC in 2000, and IEEE International Conference on Computer Design in 1989.