PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

rsta.royalsocietypublishing.org

Research

Cite this article: Gaillardon P-E, Amarù LG, Bobba S, De Marchi M, Sacchetto D, De Micheli G. 2014 Nanowire systems: technology and design. *Phil. Trans. R. Soc. A* **372**: 20130102. http://dx.doi.org/10.1098/rsta.2013.0102

One contribution of 11 to a Discussion Meeting Issue 'Beyond Moore's law'.

Subject Areas:
electrical engineering, nanotechnology, computer-aided design

Keywords:
nanosystems, nanoelectronics, nanowire transistors, controllable polarity, regular arrays, logic synthesis

Author for correspondence:
Giovanni De Micheli
e-mail: giovanni.demicheli@epfl.ch

Nanowire systems: technology and design

Pierre-Emmanuel Gaillardon, Luca Gaetano Amarù, Shashikanth Bobba, Michele De Marchi, Davide Sacchetto and Giovanni De Micheli

EPFL, Lausanne, Switzerland

Nanosystems are large-scale integrated systems exploiting nanoelectronic devices. In this study, we consider *double independent gate, vertically stacked nanowire field effect transistors* (FETs) with gate-all-around structures and typical diameter of 20 nm. These devices, which we have successfully fabricated and evaluated, control the ambipolar behaviour of the nanostructure by selectively enabling one type of carriers. These transistors work as switches with electrically programmable polarity and thus realize an *exclusive or* operation. The intrinsic higher expressive power of these FETs, when compared with standard *complementary metal oxide semiconductor* technology, enables us to realize more efficient logic gates, which we organize as tiles to realize nanowire systems by regular arrays. This article surveys both the technology for double independent gate FETs as well as physical and logic design tools to realize digital systems with this fabrication technology.

1. Introduction

Nanosystems are integrated systems exploiting *nanoelectronic* devices. Extreme miniaturization has multiple positive effects, including better electronic properties (e.g. performance) and lower cost. In particular, this work considers *silicon nanowire* (SiNW) technology as a possible replacement/enhancement of current device technologies and design issues for integrated *nanowire systems*. The interest in exploring new technological

Royal Society Publishing

approaches to very-large-scale *system-on-chip* (SoC) design stems from the physical limitations and the costs of current manufacturing technologies and from the desire to use more efficient devices, still within the realm of silicon manufacturing. The downscaling of the physical features of *field-effect transistors* (FETs) has successfully produced better and cheaper devices. Nevertheless, current semiconductor technologies have succeeded mainly along two avenues: *fully depleted silicon on insulator* [1] and *FinFET* [2,3] technologies. The latter (also called TriGate technology) is a major departure from planar semiconductor manufacturing: better transistor charge control is achieved at the price of a more complex three-dimensional fabrication process. Within the quest of future technologies, we describe here *vertically stacked silicon nanowire field effect transistors* (SiNWFETs) [4] as a promising extension to the FinFETs.

An *SiNW* is a thin wire of silicon material, with a diameter ranging from some nanometres to some tenths of nanometres. Transistors are formed by surrounding a segment of the wire by an insulator (such as SiO_2 or HfO_2) and then by a coaxial conducting material (gate), thus forming a so-called *gate-all-around* (GAA) transistor. This structure yields an excellent electrostatic control of the transistor channel, consisting of the nanowire itself under the gate. As a measurable result, the transistor gives a higher I_{on}/I_{off} ratio (i.e. current ratio of the conducting over non-conducting device) [5].

At advanced technology nodes, an increasingly larger number of devices are affected by Schottky contacts at the source and drain interfaces. Hence, devices face an ambipolar behaviour, i.e. each device exhibits *n*- and *p*-type characteristics simultaneously because of the possible flow of electrons and holes in the channel. This phenomenon is often suppressed in most technologies because of the desire to create unipolar transistors, i.e. devices with a specific type of carriers: electrons for *n*-type and holes for *p*-type transistors. Nevertheless, an important recent breakthrough has shown [6,7] that it is of high interest to control the ambipolar phenomenon through *programmable polarity* devices. Indeed, by engineering the source and drain contacts and by constructing independent *double-gate* (DG) structures, the device polarity can be electrostatically programmed to be either *n*- or *p*-type at run time. The functionality of a transistor with controllable polarity is an *exclusive or* (EXOR) of the logic signals on both gates. Thus, the fundamental switching primitive, the DG-SiNWFET, is intrinsically more expressive in terms of logic when compared with standard *complementary metal oxide semiconductor* (CMOS) transistors. In other words, while regular transistors act as switches, the DG-SiNWFETs act as comparators.

The potential advantage of this powerful logic primitive may be offset by the interconnect complexity. This trend is not a surprise for nanosystems in general, including scaled CMOS. Regularity is one of the key features to increase the yield of integrated circuits at advanced technology nodes [8], while keeping the routing complexity under control. Therefore, nanowire systems can be realized as regular arrays of elementary logic blocks, called *sea of tiles* (SoT) [9]. Thanks to a novel symbolic layout methodology, a desired logic function can be mapped onto an array of logic tiles, thereby enabling the automatic placement of digital circuits onto a SoT organization.

In a similar vein, a logic design has to be mapped efficiently onto the SiNW primitives. These primitives can support the realization of both unate and binate functions. Note that CMOS logic primitives are inherently inverting, thus privileging the realization of negative unate functions. Hence, logic synthesis and algorithms supporting the mapping of architectural-level specification into DG-SiNWFET netlists are mandatory.

This paper aims at surveying the main results associated with DG-SiNWFETs from technology to physical design and to logic synthesis. The remainder of the paper is organized as follows. In §2, we present our DG-SiNWFET technology and its circuit-level features. In §3, we introduce means of describing regular transistor arrangements to mitigate the impact of the additional gate, and summarize the associated physical design methodology. In §4, we describe the basis for a new logic synthesis flow, whereas, in §5, we derive the potential of the approach for arithmetic and fault-tolerant architectures. Section 6 concludes this work.

2. Technology overview

Here, we introduce the technology of DG-SiNWFETs and the associated circuit structures.

(a) Transistors with controllable polarity

The ambipolar conduction phenomenon is observable in several nanoscale FET devices (45 nm node and below), including silicon [10], carbon nanotubes [11] and graphene [12]. The control of the ambipolarity allows us to adjust the device polarity online. Such transistors, i.e. with a controllable polarity, have been experimentally fabricated in several novel technologies, such as carbon nanotubes [13], graphene [14] and SiNWs [15,16]. To the best of our knowledge, Sacchetto *et al.* [17] and De Marchi *et al.* [18] were the first to fabricate and test successfully SiNW transistors with independent individual control. They introduced DG-SiNWFETs where one gate controls the polarity (i.e. type of carrier, *n* or *p*), whereas the other gate controls the carrier flow in the channel. The operation of these FETs is enabled by the regulation of Schottky barriers on source/drain junctions through the additional gate.

In particular, De Marchi *et al.* [18] fabricated vertically stacked SiNWFETs, featuring two *gate-all-around* electrodes (figure 1). Vertically stacked GAA SiNWs represent a natural evolution of FinFET structures, providing better electrostatic control over the channel and consequently superior scalability properties [18].

In the device, one gate electrode, the *control gate* (CG), acts conventionally by turning *on* and *off* the device. The other electrode, the *polarity gate* (PG), acts on the side regions of the device, in proximity of the *source/drain* (S/D) Schottky junctions, switching the device polarity dynamically between *n*- and *p*-type (figure 2). The input and output voltage levels are compatible, resulting in directly cascadable logic gates. It should be noted that owing to the device geometries, the two gates are not identical from a size standpoint. Indeed, the PG is roughly two times bigger than the CG, leading to differences in their timing responses. Such a behaviour can be easily compensated at the design level by assigning the signal with the lowest frequency/switching activity to the slowest gate terminal.

Thanks to their one-dimensional structure, DG-SiNWFETs demonstrate remarkable electrostatic performances. Figure 2 depicts the subthreshold slopes of 64 mV/Dec and 70 mV/Dec for the *p*-type and *n*-type parts of the characteristic, respectively, hence competing with the most advanced FinFET technologies [3]. In addition, the one-dimensional electrostatic control over the channel coupled to the use of a Schottky barrier-based injection mechanism enables very low *off-current* densities of a few pA per μm when compared with few tens of pA per μm for low-power FinFETs [3]. These combined facts qualify the presented device technology as *high-performance low-standby-power* technology.

(b) Logic operations with double-gate field effect transistors

Digital circuits using these transistors can exploit both gates as inputs, thereby enabling the design of compact cells that implement XOR more efficiently than in CMOS. Indeed, in the context of digital operations, DG-SiNWFETs realize intrinsically an XOR characteristic, because the transistor is ON when PG = CG, i.e. $\overline{\mathrm{PG} \oplus \mathrm{CG}} = 1$, and consequently is OFF when $\mathrm{PG} \oplus \mathrm{CG} = 1$. Figure 3 presents a pseudo-logic XOR gate. The device in the *pull-down* network is polarized by means of the PG. In the case of the *n*-type polarization, the characteristics of a pseudo-logic inverter are obtained (green). In the *p*-type polarization, a buffer is obtained (blue). As shown in the inset truth table, an XOR function can be implemented by a single transistor and a pull-up.

The unique feature of this device of being polarized electrostatically was first used to build a reconfigurable logic cell [6], and later used to define a static XOR-intensive logic family [7]. In particular, a full-swing two-input XOR gate can be achieved by using a complementary pull-up and parallel transistors to avoid threshold drops. The XOR and XNOR implementations,

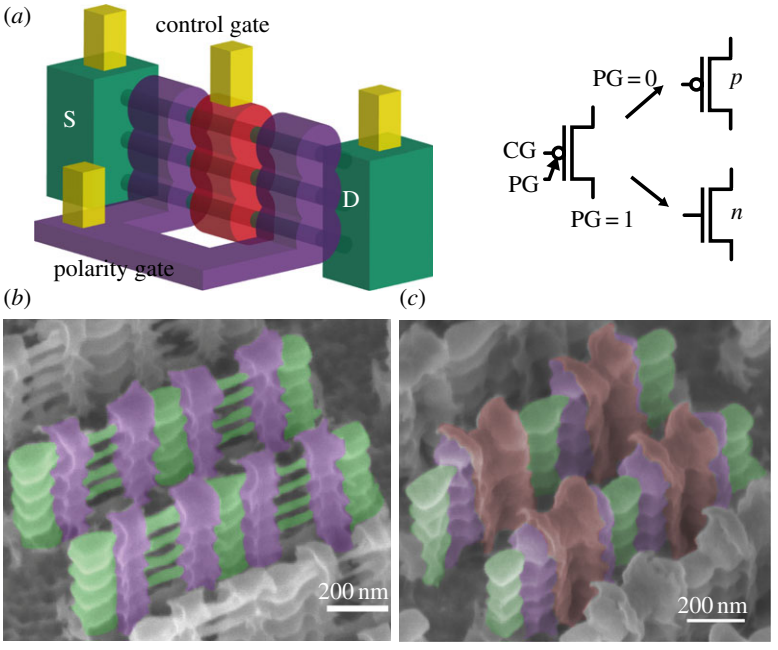

Figure 1. Three-dimensional sketch of the SiNWFET featuring two independent gates and its associated symbol (*a*). Tilted SEM views of an array of fabricated devices before creation of the control gate (*b*) and after addition of the polarity gates (*c*). S/D pillars and nanowires (green), PG (violet) and CG (red) are shown [18].

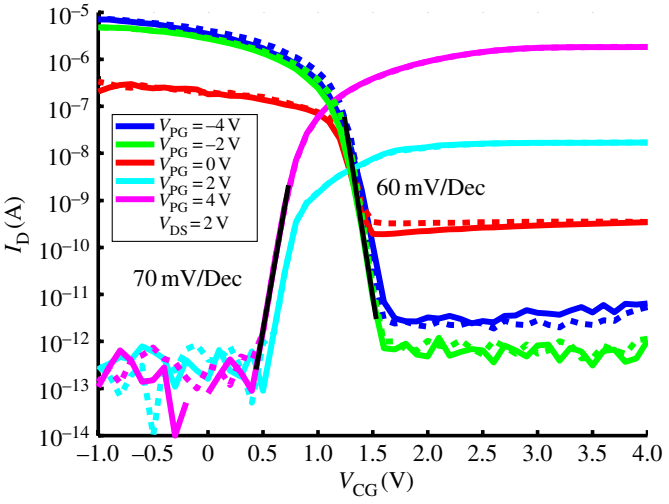

Figure 2. I_{DS}–V_{CG} logarithmic plot of a measured device for several V_{PG} voltages. Curves extracted at $V_{DS} = 2$ V [18].

reported in figure 4, require four transistors, whereas the traditional full-swing static CMOS implementation uses eight transistors [20].

Various families of logic gates can be designed for DG-SiNWFETs. In particular, one can extend the principle shown in figure 4 to design arbitrary combinational logic functions. Alternatively, fewer transistors can be used by either using a dynamic (or resistive) load, or by correcting the reduced swing owing to threshold drops by using an output buffer. Examples of realizations of arbitrary functions are shown in figure 5.

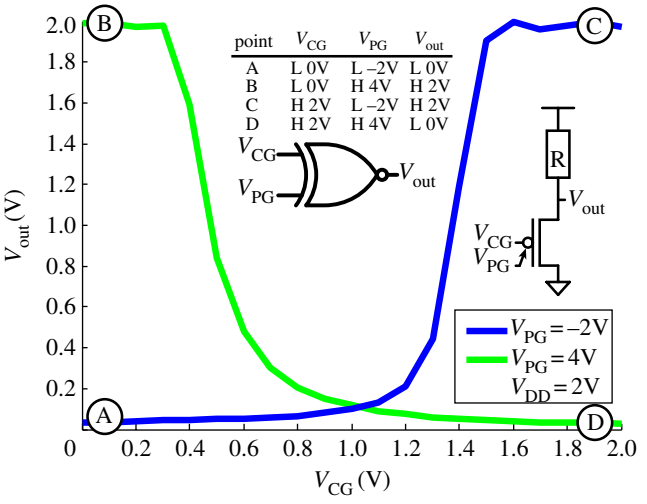

Figure 3. Pseudo-logic XOR characteristic obtained using a single SiNWFET with controllable polarity [19].

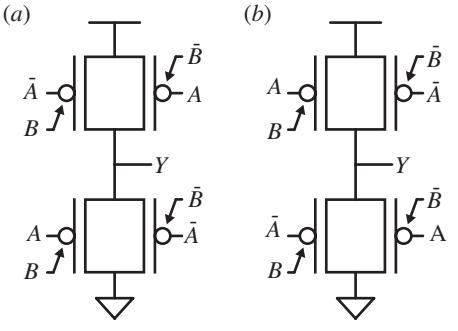

Figure 4. Two-input XOR (*a*) and XNOR (*b*) gates built with DG-SiNWFETs [7].

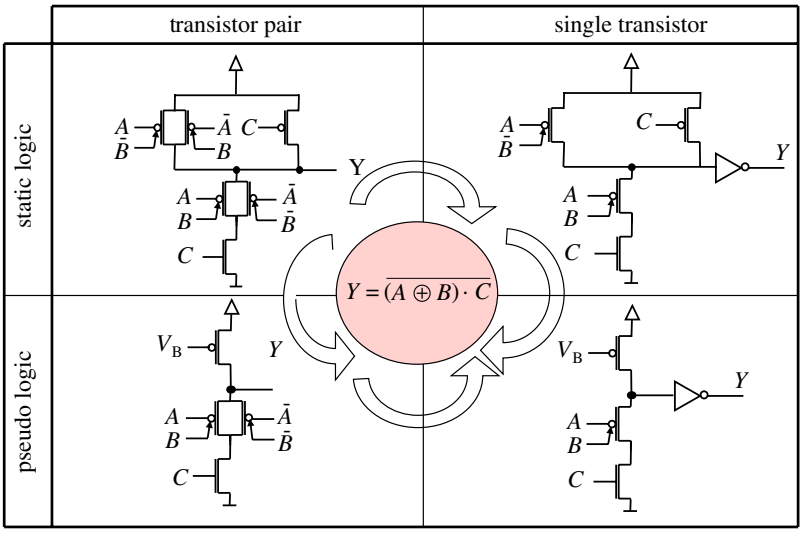

Figure 5. Various implementations of the function $Y = \overline{(A \oplus B) \cdot C}$. (Online version in colour.)

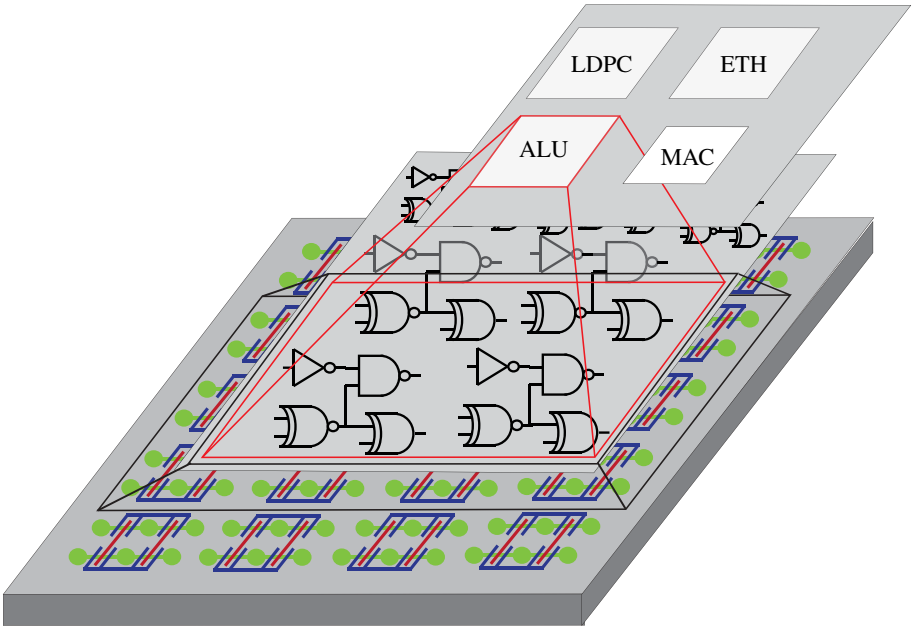

Figure 6. Conceptual representation of a regular *sea of tiles*. Tiles are configured to realize logic functions that are part of a complex system such as a processor [19]. (Online version in colour.)

3. Sea of tiles: how to deal with the routing congestion

Regular layout fabrics have the advantage of higher yield as they maximize layout manufacturability. In this section, we describe a novel architecture, called SoT, which is an array of logic tiles that are uniformly spread across the chip. The concept is illustrated in figure 6. Each tile is a template that can be wired to implement an elementary logic gate, such as a NOR, NAND, XOR, DFF or more generally a single-output combinational logic function. Note first that functions realized in ambipolar technology are not restricted to be unate. It is important to stress that the choice of logic tile (or tiles) to use in an array is important, as larger tiles can implement more complex functions, but waste devices for smaller functions, as in the case of gate arrays.

(a) Towards a regular gate arrangement

Layout regularity is one of the key features required to increase the yield of integrated circuits at advanced technology nodes [8]. Various regular fabrics have been proposed throughout the evolution of the semiconductor industry, with some recent approaches explained in [8, 21,22]. In gate-array fabric style, a sea of prefabricated transistors is customized to obtain a desired logic gate. The customization of generic gate arrays comes at a large area cost as well as routing overhead, thereby increasing the performance gap between application-specific integrated circuits (ASICs) and gate arrays. However, strict design rules, at 22 nm technology node and beyond, have led to ASIC cell layouts with arrays of gates with a constant gate pitch, which resemble a sea-of-gates layout style. In Bobba *et al.* [9], a logic *tile* was defined as a fixed pattern of prefabricated transistor pairs grouped together. Uncommitted tiles can then be mapped to logic cells by connecting the gates and the S/D free terminals.

(b) Layout techniques

To enable the compact implementation of functions with the proposed transistors, we use a novel symbolic-layout technique, called *dumbbell–stick diagrams* [9].

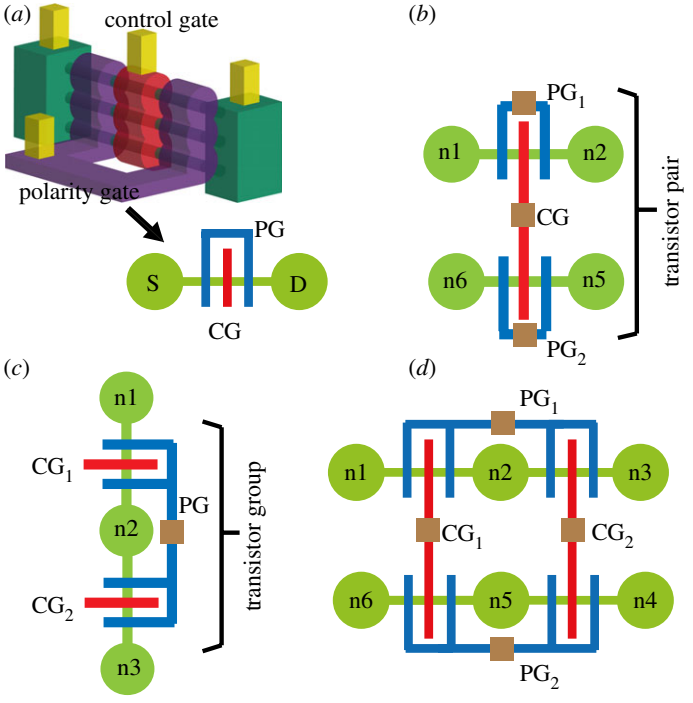

Figure 7. Dumbbell–stick diagram (*a*), transistor pairing (*b*), transistor grouping (*c*) and logic tile (*d*).

(i) Dumbbell–stick diagram

Similar to the CMOS stick diagrams [23], *dumbbell–stick diagrams* abstract the topology of logic gates with DG FETs technology. They are a convenient means for designing compact layouts and for minimizing the cell routing complexity. Figure 7*a* shows the dumbbell–stick diagram and how it is inspired by the physical shape of the device. The suspended SiNWs between the source and drain contacts form the basic dumbbell. The CG and the PG constitute the sticks. From this representation, we introduce the notion of transistor pairing and transistor grouping. *Transistor pairing* (figure 7*b*) helps in aligning the CGs of the complementary transistors in the pull-up and pull-down networks, whereas with *transistor grouping* (figure 7*c*) PGs of adjacent transistors are connected together. A logic tile is defined as an array of transistor pairs, with contiguous S/D pairs. Pairing and grouping reduce the number of *input* pins to the tile. A *tile*, consisting of two transistor pairs grouped together, is depicted in figure 7*d*. This simple tile is very effective in realizing logic primitives.

(ii) Layout technique for simple unate logic gates

Unate logic functions (e.g. NAND, NOR, AOI, etc.) with controllable-polarity devices are obtained by biasing the PGs of the *pull-up-network* (PUN) and *pull-down-network* (PDN) to G_{ND} and V_{DD}, respectively. Hence, all the transistors in the PUN (and PDN) are grouped together (i.e. PGs of the stacked transistors are connected together). The personalization of the tile is reminiscent of the methods used for CMOS cells, which determine an optimum sequence of pairs with a minimum number of gaps [24]. Figure 8*a* shows an example of a two-input NAND gate with the PGs biased to either G_{ND} or V_{DD}. Figure 8*b* shows its equivalent dumbbell–stick diagram.

(iii) Layout technique for simple binate logic gates

In the case of binate functions such as the XORs, the PGs in the PUN (and PDN) cannot be grouped, because they require independent inputs. An efficient implementation of a two-input

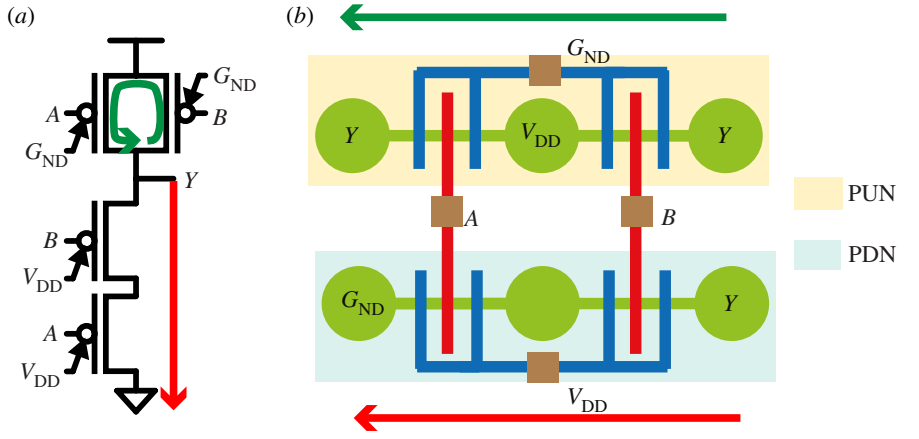

Figure 8. Schematic of a static two-input NAND gate (*a*) and its equivalent dumbbell–stick diagram (*b*).

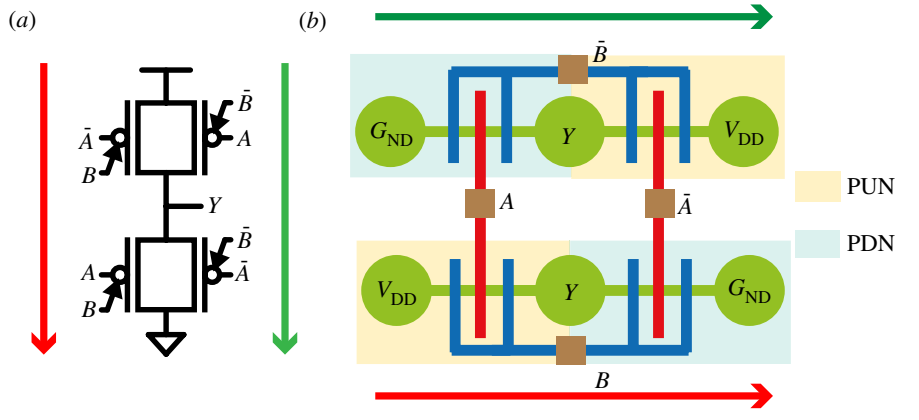

Figure 9. Schematic of a static two-input XOR gate [7] (*a*) and its equivalent dumbbell–stick diagram (*b*).

XOR is shown in figure 9, where gates with similar polarity are grouped together to reduce routing. From the dumbbell–stick diagram, we can observe that the PUN and PDN are placed next to each other, which is possible with DG-SiNWFET technology as the transistors are field controlled to make them *p*-type or *n*-type. More complex cell designs have been proposed which leverage upon embedded XOR functionality of DG FETs [7,25,26].

(iv) Layout technique for sequential elements

Sequential elements can still be efficiently mapped onto a set of tiles. Indeed, sequential elements often embed transmission gates that can be grouped together. Figure 10 illustrates a *D flip–flop* (DFF) mapped onto an array of tiles. In this implementation, we can observe that the two transmission gates in the master (slave) stage are physically mapped onto tile_1 (tile_3), efficiently compacting the overall mapping of the circuit. The inverters in the master, slave and output stages of the DFF are mapped onto tile_2, tile_4 and tile_5, respectively. The inverting stage of the clock signal is not depicted.

4. Logic synthesis

Here, we summarize models and methods for performing effectively logic synthesis and mapping into an SoT.

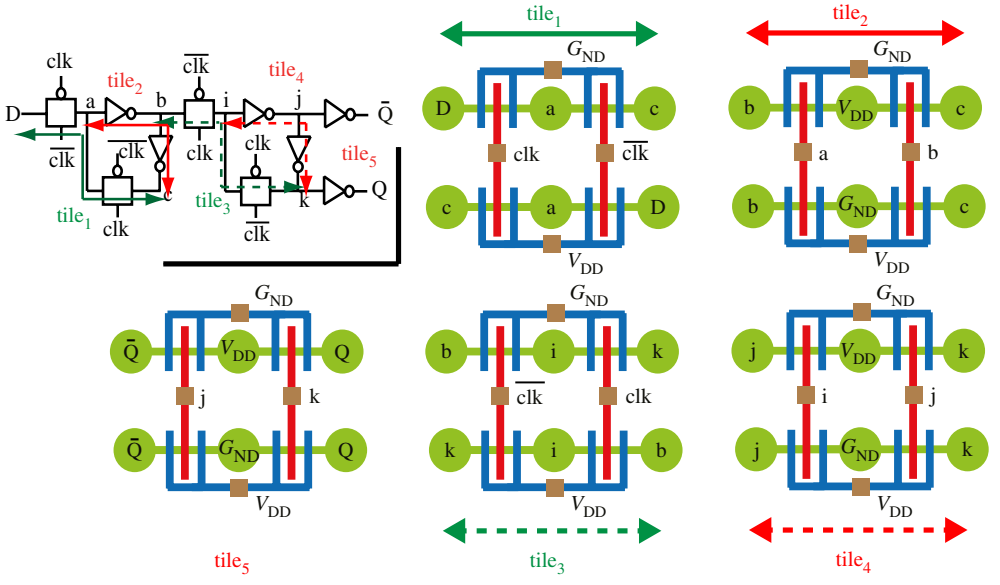

Figure 10. D flip–flop mapped on a regular set of tiles [19].

Transistors with controllable polarity intrinsically embed the XOR logical connective and thus enable the realization of XOR operators with the same ease as NAND/NORs. The original logic synthesis methods [27–29], which are the basis for current commercial tools, use NAND/NOR representations and tend to be less effective for XOR-rich circuits, such as arithmetic operators and data paths. Other methods (e.g. BDS [30]) use *binary decision diagrams* (BDDs) to fully represent, manipulate and decompose logic functions. Thanks to the advantageous BDD-based XOR-decomposition techniques, BDS efficiently synthesizes XOR-intensive circuits. In the following, we show a formalism that is directly applicable to logic circuits to be implemented with XOR primitives, such as those based on DG-SiNWFETs. In particular, we introduce a novel BDD extension, called *biconditional binary decision diagrams* (BBDDs), that presents the advantage of directly supporting the behaviour of DG-SiNWFETs. Such a representation is canonical and demonstrates powerful properties when coupled to one-pass synthesis methodologies.

(a) Biconditional binary decision diagrams

This section summarizes BBDDs. First, it presents the core logic expansion that drives BBDDs. Then, it gives ordering and reduction rules that make *reduced and ordered BBDDs* (ROBBDDs) canonical. A detailed description is given in [31].

(i) Biconditional expansion

In standard BDDs, each *non-terminal* node represents a Shannon expansion:

$$f(x, y, \ldots, z) = x.f(1, y, \ldots, z) + \bar{x}.f(0, y, \ldots, z).$$

In BBDDs, the Shannon expansion is replaced by the *biconditional expansion:*

$$f(x, y, \ldots, z) = (x \oplus y).f(\bar{y}, y, \ldots, z) + (\overline{x \oplus y}).f(y, y, \ldots, z).$$

Note that the *biconditional expansion* is a special case of the (x_i, p)-decomposition in [32] that extends the Shannon expansion. Note that only functions with two or more variables can be decomposed by a biconditional expansion. Indeed, in single variable functions, the XOR and XNOR terms cannot be computed. In such a condition, the biconditional expansion of a single

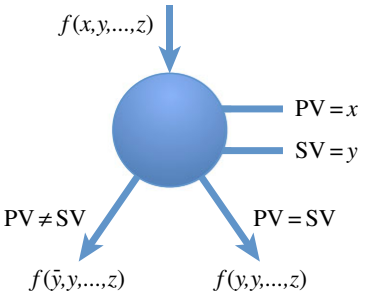

Figure 11. BBDD *non-terminal* node [31]. (Online version in colour.)

variable function reduces to a Shannon expansion by setting the second variable y to logic 1. With this boundary condition, any Boolean function can be fully represented in terms of biconditional expansions.

(ii) Biconditional binary decision diagram structure and ordering

A BBDD is a BDD driven by the *biconditional expansion* in place of Shannon's expansion. Each non-terminal 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 *non-terminal* node is provided in figure 11. We refer hereafter to PV = SV and PV = SV edges in a BBDD node simply as the $\neq$ -edges and = -edges, respectively.

To achieve OBBDDs, a variable order must be imposed for PVs and a rule for the other variables assignment must be provided. We use the following *chain variable order* (CVO) to address this task. Given a Boolean function f and an order $\pi = (\pi_0, \pi_1, \ldots \pi_{n-1})$ of the inputs, PVs and SVs are ordered as

$$\begin{cases} \mathrm{PV}_i = \pi_i \\ \mathrm{SV}_i = \pi_{i+1} \end{cases} \quad \text{with } i = 0, 1, \ldots, n-2; \quad \begin{cases} \mathrm{PV}_{n-1} = \pi_{n-1} \\ \mathrm{SV}_{n-1} = 1. \end{cases}$$

Note that if we swap π_i with π_j in the initial order π, owing to some reordering operation, this simply translates through the CVO as PV_i exchanged with PV_j and SV_{i-1} with SV_{j-1}.

Example: from $\pi = (\pi_0, \pi_1, \pi_2)$, the corresponding CVO ordering is obtained by the following method. First, $\mathrm{PV}_0 = \pi_0$, $\mathrm{PV}_1 = \pi_1$ and $\mathrm{SV}_0 = \pi_1$, $\mathrm{SV}_1 = \pi_2$ are assigned. Then, $\mathrm{PV}_2 = \pi_2$ and $\mathrm{SV}_2 = 1$. The consecutive ordering by pairs $(\mathrm{PV}_i, \mathrm{SV}_i)$ 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 structures. We refer to ordered binary biconditional decision structures as BBDDs ordered by the CVO.

(iii) Biconditional binary decision diagram reduction

As in the case of OBDDs, also OBBDDs can be reduced to improve the representation efficiency, according to a set of rules. The straightforward extension of OBDD reduction rules [4] to OBBDDs corresponds to the iterated merging of isomorphic subgraphs.

Moreover, the OBBDD can be further reduced by eliminating levels with no nodes. Last, subgraphs that represent functions of a single variable can be collapsed into a single BDD node. Reduced OBBDDs are canonical [31].

(b) One-pass logic synthesis

One pass synthesis (OPS) [33] is a logic synthesis methodology where logic optimization and technology mapping phases are combined in a single step carried out through a common data structure, e.g. BDDs. To target XOR-rich functions, we use BBDDs as data structure.

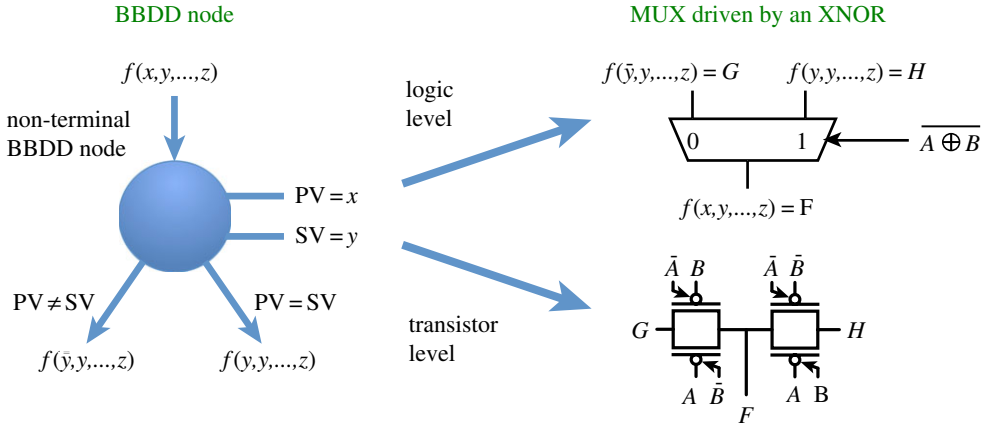

Figure 12. BBDD node corresponding logic gate and realization in ambipolar and CMOS technologies [31]. (Online version in colour.)

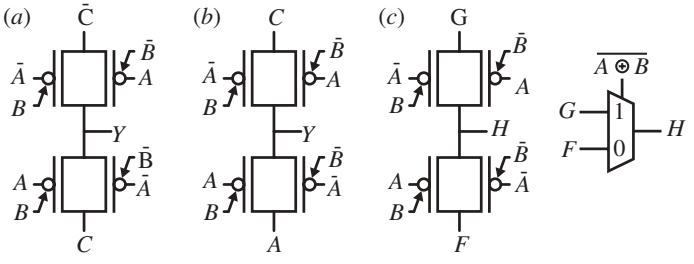

Figure 13. Transmission-gate three-input XOR (*a*), three-input majority logic gate (*b*) and generalized arithmetic gate (MUX-XNOR) (*c*).

In BBDD-based OPS, logic optimization corresponds to the ROBBDD construction. Note that most of the algorithms for ROBDD construction, e.g. BUILD, APPLY [34], etc., can be adapted to ROBBDDs, hence to support the *biconditional expansion* in place of Shannon's expansion. Standard dynamic variable reordering algorithms can be applied also with the CVO (figure 12).

5. System-level design issues

The combination of the DG-SiNWFETs technology and BBDS-based synthesis can be applied to the design of both data path and control circuits. In particular, it enables the compact impact implementation of arithmetic functions and opens novel horizons in terms of testing and online fault detection.

(a) Compact arithmetic operators

DG-SiNWFETs enable the efficient design of parity circuits. Besides the efficient full-swing four-transistor XOR gate realization, shown in figure 4, a three-input XOR realization [26] leverages pass-transistor logic, as depicted in figure 13*a*. Note that in static CMOS, the same gate has 10 devices in place of 4 here [20].

Inspired by this last structure, a four-transistor three-input majority logic gate [35] is shown in figure 13*b*. This gate relies on the pass-transistor implementation of the MAJ(A, B, C) function rewritten as

$$\mathrm{MAJ}(A, B, C) = A \cdot (\overline{A \oplus B}) + C \cdot (A \oplus B).$$

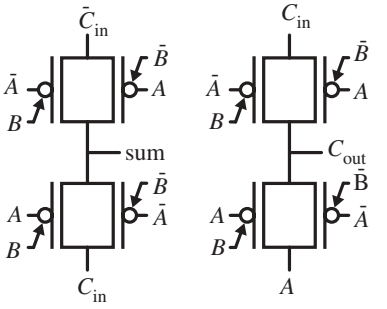

Figure 14. Full-adder implementation with eight controllable polarity devices.

Note that in static CMOS, the same gate has 10 devices in place of 4 [20]. Moreover, the four DG-SiNWFETs configuration (of figure 13*a*) can be generalized to the MUX-like structure depicted in figure 13*c*. Its functionality corresponds to a multiplexer driven by an XNOR operation between A and B, selecting between two external signals F and G. With different assignments of F and G, it is possible to implement three-input MAJ($F = A$, $G = C$), three-input MIN($F = A'$, $G = C'$), three-input XOR($G = C'$, $F = C$) and two-input XOR($G = 1$, $F = 0$) logic gates. Therefore, this four-transistor structure can be seen as a generalized arithmetic gate.

The *full-adder* (FA) is a widely used arithmetic circuit that supports the addition of two binary numbers. It is represented by the following three-input two-output logic function:

$$\text{sum}(A, B, C_{in}) = A \oplus B \oplus C_{in}$$

and

$$C_{out} = \text{MAJ}(A, B, C_{in}) = A.\overline{(A \oplus B)} + C_{in}.(A \oplus B).$$

Controllable polarity transistors offer an advantageous implementation for both the *sum* and C_{out} functions using two generalized arithmetic gates. Therefore, the full-adder is competitively realized by eight devices, input inverters apart, as depicted by figure 14. The corresponding static (transmission gate) CMOS version has 28 (14) transistors [20].

(b) Self-checking computation

Among online testing strategies, self-checking circuits offer an efficient way of testing circuits without adding redundant voter circuitry such as in *triple modular redundancy* [36]. The most used self-checking technique is the parity prediction scheme [37]. Parity computation relies largely on the XOR operation, and therefore its implementation with the DG-SiNWFET technology can be fairly effective. The design of a self-checking ripple-carry adder has been introduced in [35] and is shown in figure 15.

The adder includes one-bit adders with complemented carry, double-rail checkers and parity generation trees. The complemented carry can be included within the existing FA structure, thanks to a compact minority operator. Indeed, only four extra transistors are required, whereas static CMOS design style needs 12 extra transistors. The parity-generation tree includes cascaded compact two-input XORs. Unfortunately, the compact four-transistor XOR implementation enabled by DG-SiNWFETs does not provide the fault-secure property. Indeed, in the case of a fault on the PGs, there exist some conditions where all the transistors take the same polarity, therefore leading to undetermined levels at the output. For this reason, in [35], a few parts of the circuit (the double-rail checkers) are still implemented using a traditional static CMOS implementation to guarantee the self-checking property. Nevertheless, the use of DG-SiNWFETs opens new opportunities also for fault-tolerant architectures.

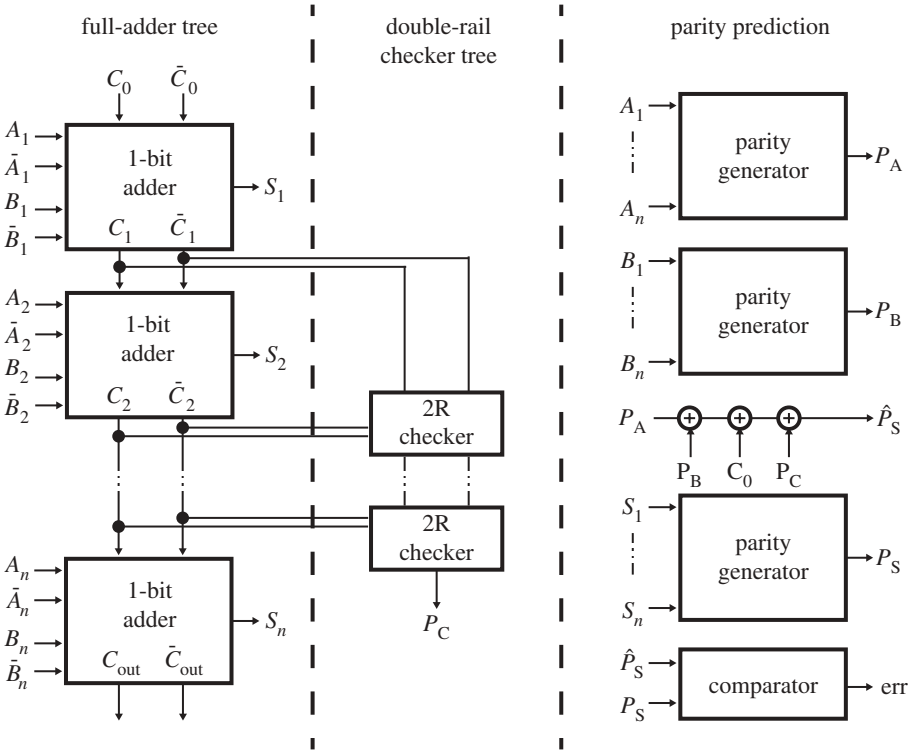

Figure 15. Self-checking *n*-bit adder using carry-checking parity-prediction scheme [36].

6. Conclusion

We have presented here a complete design framework for nanoelectronic computational systems that leverage DG-SiNWFET technology. This framework includes semiconductor process development, device and circuit design, models and design tool research as well as architecting overall systems. In particular, we have shown the synergy of research results coming from novel device fabrication with circuit and architectural design. This research aims at achieving scalable arrays of nanodevices within regular arrangements, as a way to mitigate wiring variability. Last but not least, we have shown the challenges in design automation for nanotechnologies at various levels of abstraction.

Funding statement. This research is supported by the ERC senior grant no. NanoSys ERC-2009-AdG-246810.

References

1. Barral V *et al.* 2007 Strained FDSOI CMOS technology scalability down to 2.5 nm film thickness and 18 nm gate length with a TiN/HfO_2 gate stack. *IEDM Tech. Dig.* 61–64. (doi:10.1109/IEDM.2007.4418863)
2. Auth C *et al.* 2012 A 22 nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors. In *Proc. 2012 Symp. on VLSI Technology*, pp. 131–132. (doi:10.1109/VLSIT.2012.6242496)
3. Jan C-H *et al.* 2012 A 22 nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications. *IEDM Tech. Dig.* 3.1.1–3.1.4. (doi:10.1109/IEDM.2012.6478969)
4. Suk SD *et al.* 2005 High performance 5 nm radius twin silicon nanowire MOSFET (TSNWFET): fabrication on bulk Si wafer, characteristics, and reliability. *IEDM Tech. Dig.* 717–720. (doi:10.1109/IEDM.2005.1609453)

5. Bangsaruntip S *et al.* 2009 High performance and highly uniform gate-all-around silicon nanowire MOSFETs with wire size dependent scaling. *IEDM Tech. Dig*. 1–4. (doi:10.1109/IEDM.2009.5424364)
6. O'Connor I *et al.* 2007 CNTFET modeling and reconfigurable logic-circuit design. *IEEE Trans. Circuits Syst. I* **54**, 2365–2379. (doi:10.1109/TCSI.2007.907835)
7. Ben Jamaa MH, Mohanram K, De Micheli G. 2009 Novel library of logic gates with ambipolar CNTFETs: opportunities for multi-level logic synthesis. *DATE Tech. Dig*. 622–627. (doi:10.1109/DATE.2009.5090742)
8. Jhaveri T, Pileggi L, Rovner V, Strojwas AJ. 2007 Maximization of layout printability/manufacturability by extreme layout regularity. *J. Micro/Nanolith. MEMS.* **6**, 031011.
9. Bobba S, De Marchi M, Leblebici Y, De Micheli G. 2012 Physical synthesis onto a sea-of-tiles with double-gate silicon nanowire transistors. In *Proc. 49th Annual Design Automation Conf.*, pp. 42–47. (doi:10.1145/2228360.2228369)
10. Colli A, Pisana S, Fasoli A, Robertson J, Ferrari AC. 2007 Electronic transport in ambipolar silicon nanowires. *Phys. Stat. Sol. B* **244**, 4161–4164. (doi:10.1002/pssb.200776154)
11. Martel R, Derycke V, Lavoie C, Appenzeller J, Chan KK, Tersoff J, Avouris Ph. 2001 Ambipolar electrical transport in semiconducting single-wall carbon nanotubes. *Phys. Rev. Lett.* **87**, 256805. (doi:10.1103/PhysRevLett.87.256805)
12. Geim AK, Novoselov KS. 2007 The rise of graphene. *Nat. Mater.* **6**, 183–191. (doi:10.1038/nmat1849)
13. Lin Y-M, Appenzeller J, Knoch J, Avouris P. 2005 High-performance carbon nanotube field-effect transistor with tunable polarities. *IEEE Trans. Nanotechnol.* **4**, 481–489. (doi:10.1109/TNANO.2005.851427)
14. Harada N, Yagi K, Sato S, Yokoyama N. 2010 A polarity-controllable graphene inverter. *Appl. Phys. Lett.* **96**, 12102. (doi:10.1063/1.3280042)
15. Appenzeller J, Knoch J, Tutuc E, Reuter M, Guha S. 2006 Dual-gate silicon nanowire transistors with nickel silicide contacts. *IEDM Tech. Dig*. 1–4. (doi:10.1109/IEDM.2006.346842)
16. Heinzig A, Slesazeck S, Kreupl F, Mikolajick T, Weber WM. 2011 Reconfigurable silicon nanowire transistors. *Nano Lett.* **12**, 119–124. (doi:10.1021/nl203094h)
17. Sacchetto D, De Micheli G, Leblebici Y. 2012 Ambipolar gate-controllable SiNW FETs for configurable expressive capability. *IEEE Electron Dev. Lett.* **33**, 143–145. (doi:10.1109/LED.2011.2174410)
18. De Marchi M, Sacchetto D, Frache S, Zhang J, Gaillardon P-EJM, Leblebici Y, De Micheli G. 2012 Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs. *IEDM Tech. Dig*. 8.4.1–8.4.4. (doi:10.1109/IEDM.2012.6479004)
19. Gaillardon P-E, Amaru LG, Bobba S, De Marchi M, Sacchetto D, Leblebici Y, De Micheli G. 2013 Vertically stacked double gate nanowire FETs with controllable polarity: from devices to regular ASICs. *DATE Tech. Dig*. 625–630. (doi:10.7873/DATE.2013.137)
20. Rabaey JM, Chandrakasan AP, Nikolic B. 2003 *Digital integrated circuits: a design perspective*. Upper Saddle River, NJ: Prentice Hall.
21. Lin Y-W, Marek-Sadowska M, Maly W. 2009 Transistor-level layout of high-density regular circuits. In *Proc. 2009 Int. Symp. on Physical Design*, pp. 83–90. (doi:10.1145/1514932.1514954)
22. Ran Y, Marek-Sadowska M. 2006 Designing via-configurable logic blocks for regular fabric. *IEEE Trans. VLSI* **14**, 1–14. (doi:10.1109/TVLSI.2005.863196)
23. Weste N, Harris D. 2004 *CMOS VLSI design: a circuit and systems perspective*. Boston, MA: Pearson/Addison Wesley.
24. Uehara T, Vancleemput W. 1981 Optimal layout of CMOS functional arrays. *IEEE Trans. Comput.* **C-30**, 305–312. (doi:10.1109/TC.1981.1675787)
25. De Marchi M, Ben Jamaa MH, De Micheli G. 2010 Regular fabric design with ambipolar CNTFETs for FPGA and structured ASIC applications. In *Proc. IEEE/ACM Int. Symp. on Nanoscale Architectures*, pp. 65–70. (doi:10.1109/NANOARCH.2010.5510923)
26. Zukovski A, Xuebei Y, Mohanram K. 2011 Universal logic modules based on double-gate carbon nanotube transistors. *DAC Tech. Dig*. 884–889.
27. Brayton RK, Rudell R, Sangiovanni-Vincentelli A, Wang AR. 1987 MIS: a Multiple-level logic optimization system. *IEEE Trans. CAD* **6**, 1062–1081. (doi:10.1109/TCAD.1987.1270347)
28. Sentovich E *et al.* 1992 *SIS: a system for sequential circuit synthesis*. Berkeley, CA: ERL, Dept. EECS, Univ. California. UCB/ERL M92/41.
29. ABC logic synthesis tool. See http://www.eecs.berkeley.edu/alanmi/abc/.

30. Yang C, Ciesielski M. 2002 BDS: a BDD-based logic optimization system. *IEEE Trans. CAD* **21**, 866–876. (doi:10.1109/TCAD.2002.1013899)
31. Amarù L, Gaillardon P-E, De Micheli G. 2013 Biconditional BDD: a novel canonical BDD for logic synthesis targeting XOR-rich circuits. *DATE Tech. Dig*. 1014–1017. (doi:10.7873/DATE.2013.211)
32. Bernasconi A, Ciriani V, Trucco G, Villa T. 2009 On decomposing Boolean functions via extended cofactoring. *DATE Tech. Dig*. 1464–1469. (doi:10.1109/DATE.2009.5090894)
33. Drechsler R, Gunther W. 2002 *Toward one-pass synthesis*. Berlin, Germany: Springer.
34. Bryant RE. 1986 Graph-based algorithms for Boolean function manipulation. *IEEE Trans. Comput.* **C-35**, 677–691. (doi:10.1109/TC.1986.1676819)
35. Turkyilmaz O, Clermidy F, Amarù LG, Gaillardon P-E, De Micheli G. 2013 Self-checking ripple-carry adder with ambipolar silicon nanowire FET. In *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 2127–2130. (doi:10.1109/ISCAS.2013.6572294)
36. Graham P, Gokhale M. 2004 *Nanocomputing in the presence of defects and faults: a survey. Nano, quantum and molecular computing*. Berlin, Germany: Springer.
37. Nicolaidis M. 1993 Efficient implementations of self-checking adders and ALUs. In *Proc. 23rd Int. Symp. on Fault-Tolerant Computing*, pp. 586–595. (doi:10.1109/FTCS.1993.627361)