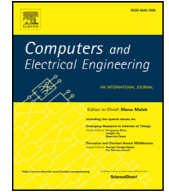




Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compelecengMajority-Logic, its applications, and atomic-scale embodiments[☆]Behrooz Parhami^a, Dariush Abedi^b, Ghassem Jaberipur^{b,c,*}^a Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, United States^b Department of Computer Science and Engineering, Shahid Beheshti University, Tehran 19839-63113, Iran^c School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ARTICLE INFO

Article history:

Received 25 June 2019

Revised 20 January 2020

Accepted 21 January 2020

Keywords:

Algorithms implemented in hardware

Cellular arrays and automata

High-speed arithmetic

Logic design styles

Low-power design

Performance analysis and design aids

ABSTRACT

Today's computing is increasingly data-intensive, heralding the age of big data. With greater data volumes, come the needs for faster processing, greater storage capacity, and expanded communication bandwidth, all of which imply the expenditure of more energy. Thus, energy efficiency, already a major design consideration, will assume broader significance in the coming years. As important as storage and communications are, our focus in this paper is on better technology to reduce the computation (logic manipulation) power. We review majority logic, a special case of threshold logic, show how a number of common arithmetic/logic operations can be performed using the majority-gate primitive, and review an impressive array of atomic-scale logic technologies that are particularly efficient in realizing the majority or minority function. We conclude that a combination of orders of magnitude energy reduction by virtue of the technology used and implementation strategies that lead to comparable complexity in terms of majority gates when contrasted with currently used circuit primitives (AND, OR, XOR, NOT, mux) leads to energy-efficient realization of arithmetic/logic functions suitable for use in the age of big data.

© 2020 Published by Elsevier Ltd.

1. Introduction

Talk of big data pervades not only our technical exchanges but also our discourse on social media and at the dinner table. Like many other attributes in the modern world, data production has risen exponentially since the advent of digital computing in the mid-20th-century [1]. There are many definitions of, and concerns with, big data. The exponential rise of data volume is, of course, a main component of all definitions. Less obvious are the other so-called "V"s (Velocity, Variety, Veracity, Value), data attributes used to characterize the age of big data.

- *Volume*: Necessitates massive storage facilities and methods to ensure data availability and reliability.
- *Velocity* (in production or rate of change): Creates the need for high-speed data access and processing.
- *Variety*: Implies the use of flexible and heterogeneous storage and data-processing resources.
- *Veracity*: Requires methods to ensure that inevitable errors in data do not propagate beyond control.

[☆] This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. G. Glan Devadhas.



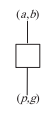


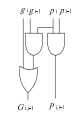
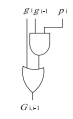
* Corresponding author.

E-mail addresses: parhami@ece.ucsb.edu (B. Parhami), d_abedi@sbu.ac.ir (D. Abedi), jaberipur@sbu.ac.ir (G. Jaberipur).

Table 1
Acronyms.

Acronym	Description	Acronym	Description
BK	Brent-Kung	LF	Ladner-Fischer
CLA	Carry look-ahead	MTJ	Magnetic tunnel junction
CDP	Critical delay path	NBM	Nano-scale bar magnets
CSTG	Charge-sharing threshold gate	NML	Nanomagnetic logic
CMTG	Current-mirror threshold gate	PMLA	Programmable majority logic arrays
CZ	Clock zone	PUM	Partially-utilized majority
DNA	Deoxyribonucleic acid	QCA	Quantum-dot cellular automata
FA	Full-adder	SET	Single-electron tunneling
FUM	Fully-utilized majority	TPL	Tunneling phase logic
HC	Han-Carlson	STMG	Spin torque majority gate
IC	Integrated circuit	STO	Spin torque oscillator
KS	Kogge-Stone	TM	Twin majority

Table 2
Symbols.

Symbol	\mathcal{M}					
Description	Majority function	Majority gate	Twin majority (Fig. 4)	(p, g) logic $p = a \vee b$ $g = a \wedge b$		

- *Value*: Calls for elaborate provisions to ensure data privacy, integrity, and longevity.

It is evident that big data needs significant processing capabilities, substantial storage capacities, and expanded communication bandwidths, all of which imply the expenditure of more energy. Thus, energy efficiency, already a dominating concern for a couple of decades, as computing bifurcated into the use of battery-operated mobile devices on the user side and energy-hungry server farms and supercomputers in the cloud, will become even more important in the coming years. The number of operations that can be performed per watt-hour of energy is thus important at all points of the system-size spectrum.

We are thus attacking the energy efficiency problem from multiple fronts. The algorithmic front will give us lower-complexity, and thus faster, application programs (both through cleverer computational schemes and via reduction of precision to the bare minimum required [2]), better-compressed data, and lower-redundancy methods for ensuring data integrity and longevity. The data management front will lead to better-organized and load-balanced storage and distribution schemes for faster and less communication-intensive data access. Finally, better technology will produce a variety of advances, which will include higher-capacity and lower-waste batteries, energy-frugal nonvolatile storage, and inherently lower-power logic circuits (our focus in this paper) for data manipulation.

An important aspect of our effort to render logic circuits as energy-efficient as possible is making more effective use of majority gates, the novel circuit elements at the heart of our approach. Prior strategies for incorporating majority elements in logic circuits end up using the inputs to a majority element partially in many cases (dubbed “partially-used majority” or PUM), in effect deploying majority gates to replace other, less capable, circuit elements. One feature of our methodology is striving to achieve full use of the capabilities of majority gates to the extent possible (“fully-used majority,” or FUM, in our nomenclature). Tables 1 and 2 contain the acronyms and symbols used throughout the paper.

In this paper, we review the history of threshold logic and its special cases of majority/minority gates in Section 2, an important starting point, given that either majority or minority function, along with inversion, forms a universal set capable of synthesizing any logic function. In Section 3, we touch upon our prior work in designing carry networks for fast adders, using majority gates and possibly inverters as the sole building blocks, and briefly discuss how other arithmetic operations can be synthesized in a similar fashion. We next present, in Section 4, details of a number of emerging technologies that are atomic-scale, and thus low-power by nature, and share the attribute of having an affinity with dense and efficient realization of majority gates. Section 5 describes architectural differences between CMOS-free and hybrid-CMOS circuit design within the discussed technologies. We conclude the paper in Section 6.

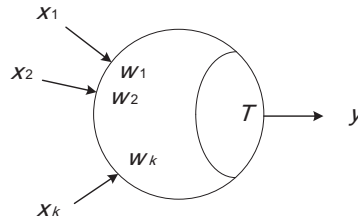


Fig. 1. Threshold logic element with arbitrary input weights.

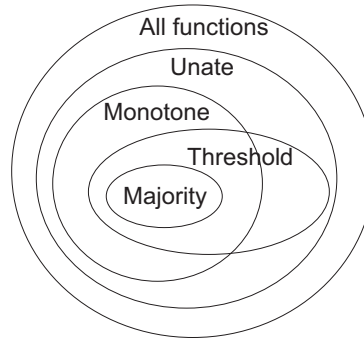


Fig. 2. Relationships among some classes of Boolean functions.

2. Threshold logic and majority gates

Threshold logic is nearly as old as digital computing and has been revisited from time to time, as a steady stream of new technologies arrived on the scene [3]. The neural networks and neuronlike computational elements of the early years of threshold logic evolved into capacitance- and inductance-based solutions [4] and several other alternate embodiments.

A generic threshold logic element/gate (see Fig. 1) has k binary inputs x_1, x_2, \dots, x_k , assigned the real weights w_1, w_2, \dots, w_k . It also has a real threshold value T and a binary output y , which assumes the value of 1 if and only if $\sum x_j w_j \geq T$.

Not every Boolean function can be realized by a threshold gate, but each realizable function corresponds to multiple weights-threshold sets. Good examples are found among voting or data-fusion functions specified by agreement sets [5]. As a specific instance, consider the voting function of the UN Security Council, which requires all 5 permanent members and at least 4 of the 10 non-permanent members to consent for a resolution to pass. A threshold realization of this voting function may have weights of 1 for each non-permanent member and 11 for each permanent member (selected on the basis of making a permanent member’s opinion more important than all 10 non-permanent members combined), with the threshold T set to 59. Alternatively, the weights can be chosen as 1 and 7, respectively, along with $T = 39$. Given multiple options such as the above, optimal realizations may be sought for each implementation technology.

Majority function with equally weighted inputs is a special case of threshold logic, with k odd, $w_1 = \dots = w_k = 1$, and $T = (k + 1)/2$. Often, the term “majority element/gate” is used for the special 3-input variant. CMOS realization of a majority gate with three binary inputs a, b , and c yields $\mathcal{M}(a, b, c) = (a \vee b)c \vee ab$. In certain technologies, majority elements can be realized directly and, thus, efficiently. However, it is also possible to realize \mathcal{M} elements via direct replacement of the AND and OR pairs in the expression above with their equivalents in the target technology.

The 3-input majority function is defined by the arithmetic expression $\mathcal{M}(a, b, c) = (a + b + c + 1)/3$, and it can be viewed as the median function. The median interpretation of the majority function allows us to use the axiomatically defined median algebra [6] to prove new results and derive various relationships.

Majority is a (positive) unate function, meaning that $\mathcal{M}(a,b,c)$ is an increasing function of each of the three input variables. As such, \mathcal{M} isn’t universal by itself. In fact, not even all unate functions can be realized by using majority elements only (see Fig. 2). However, the set composed of \mathcal{M} and logical inversion is provably universal, given that each of the binary operations AND and OR can be realized by a majority gate, as in Eqs. (1) and (2), respectively.

$$a \wedge b = \mathcal{M}(a, b, 0) \tag{1}$$

$$a \vee b = \mathcal{M}(a, b, 1) \tag{2}$$

Given the inefficiency of using the more powerful majority elements as AND or OR gates, we need synthesis procedures to directly generate efficient circuits based on majority gates (and other elements, when majority gates alone will not do). The required optimizations are generally technology-dependent [7].

3. Majority-based arithmetic circuits

As previously noted, any logical circuit defined in terms of AND and OR gates can be directly mapped into an equivalent \mathcal{M} -based circuit via Eqs. (1) and (2). However, such naïve mapping does not fully utilize the capabilities of an \mathcal{M} gate; hence the designation partially-utilized majority (PUM), in contrast to fully-utilized majority (FUM) gates [8]. As examples of the full utilization of majority gates in synthesizing arithmetic circuits, we discuss two designs from our earlier work [8], and provide three new FUM-based designs; namely the CLA logic block with blocking factor 4, along with Brent-Kung (BK) and Han-Carlson (HN) parallel prefix carry generation networks.

3.1. FUM-based CLA logic

The conventional carry look-ahead logic with blocking factor 4 is ruled by Eqn. set 3, for the i th 4-bit block. The corresponding carry signals can be expressed as in Eqn. set 4, where 8 AND and 8 OR gates are required for circuit realization. A direct mapping of such circuit to an \mathcal{M} -based one will utilize 16 PUMs, where the c_{i+j} -to- c_i ($1 \leq j \leq 4$) critical delay path (CDP) passes through two PUMs.

$$\begin{aligned} G_{i+1:i} &= g_{i+1} + p_{i+1}g_i \\ G_{i+2:i} &= g_{i+2} + p_{i+2}G_{i+1:i} \\ G_{i+3:i} &= g_{i+3} + p_{i+3}g_{i+2} + P_{i+3:i+2} G_{i+1:i} \end{aligned} \quad (3)$$

$$\begin{aligned} c_{i+1} &= g_i + p_i c_i \\ c_{i+2} &= G_{i+1:i} + P_{i+1:i} c_i \\ c_{i+3} &= G_{i+2:i} + P_{i+2:i} c_i \\ c_{i+4} &= G_{i+3:i} + P_{i+3:i} c_i \end{aligned} \quad (4)$$

The same carry signals can be expressed, as in Eqn. set 5, with 10 FUMs and only one within the CDP.

$$\begin{aligned} c_{i+1} &= \mathcal{M}(a_i, b_i, c_i) \\ c_{i+2} &= \mathcal{M}(A_{i+1:i}, B_{i+1:i}, c_i) \\ c_{i+3} &= \mathcal{M}(A_{i+2:i}, B_{i+2:i}, c_i) \\ c_{i+4} &= \mathcal{M}(A_{i+3:i}, B_{i+3:i}, c_i) \end{aligned} \quad (5)$$

The auxiliary carry-support signals $A_{i+j:i}$ and $B_{i+j:i}$, for $1 \leq j \leq 3$, were previously introduced in [8], and are reproduced here in Eqn. set 6.

$$\begin{aligned} A_{i+1:i} &= \mathcal{M}(a_{i+1}, b_{i+1}, a_i) \\ A_{i+2:i} &= \mathcal{M}(\mathcal{M}(a_{i+2}, b_{i+2}, a_{i+1}), \mathcal{M}(a_{i+2}, b_{i+2}, b_{i+1}), a_i), \\ A_{i+3:i} &= \mathcal{M}(\mathcal{M}(\mathcal{M}(a_{i+3}, b_{i+3}, a_{i+2}), \mathcal{M}(a_{i+3}, b_{i+3}, b_{i+2}), a_{i+1}), \mathcal{M}(\mathcal{M}(a_{i+3}, b_{i+3}, a_{i+2}), \mathcal{M}(a_{i+3}, b_{i+3}, b_{i+2}), b_{i+1}), a_i) \\ B_{i+1:i} &= \mathcal{M}(a_{i+1}, b_{i+1}, b_i), \\ B_{i+2:i} &= \mathcal{M}(\mathcal{M}(a_{i+2}, b_{i+2}, a_{i+1}), \mathcal{M}(a_{i+2}, b_{i+2}, b_{i+1}), b_i), \\ B_{i+3:i} &= \mathcal{M}(\mathcal{M}(\mathcal{M}(a_{i+3}, b_{i+3}, a_{i+2}), \mathcal{M}(a_{i+3}, b_{i+3}, b_{i+2}), a_{i+1}), \mathcal{M}(\mathcal{M}(a_{i+3}, b_{i+3}, a_{i+2}), \mathcal{M}(a_{i+3}, b_{i+3}, b_{i+2}), b_{i+1}), b_i) \end{aligned} \quad (6)$$

Fig. 3 depicts a TM-based implementation of Eqn. set 3, where TM stands for twin-majority gate that consists of two FUMs, as illustrated in Fig. 4 (also reproduced from [8]).

3.2. FUM-based parallel prefix adders

In our previous relevant work [8], we provided for Ladner-Fischer (LF) and Kogge-Stone (KS) parallel prefix adders whose only building blocks were FUM gates and showed the advantages of these designs relative to prior relevant works that used many PUMs within their parallel prefix architecture. In this paper, we provide two other FUM-based parallel prefix architectures realizing the well-known Brent-Kung (BK) and Han-Carlson (HC) adders, shown in Figs. 5 and 6, respectively, where the conventional BK and HC parallel prefix networks are also illustrated.

Fig. 7 depicts the QCA layout for our FUM-based realizations of the four conventional parallel prefix architectures. The corresponding figures of merit for the new designs introduced in this paper are presented in Table 3, alongside those of previous designs by us and other researchers.

The following points are worth noting:

- The LF design is the fastest.
- The KS design is the slowest, despite having one fewer FUM on its CDP. As evident from Fig. 7, this is due to it having twice as many cells and the fact that the number of cells per clock zone is limited.
- The BK design uses the least total number of \mathcal{M} gates and least number of cells leading to the least consumed area, while it is the second fastest next to the LF design.
- The CDP of all the four proposed designs travel through one or two fewer \mathcal{M} gates than those of the two previous works.
- Area and number of utilized cells of the new LF, HC, and BK designs are much lower than similar measures of the previous work and the proposed KS design.
- The delay measures of the new designs (except for the KS adder) are also lower than or equal to those of previous works.

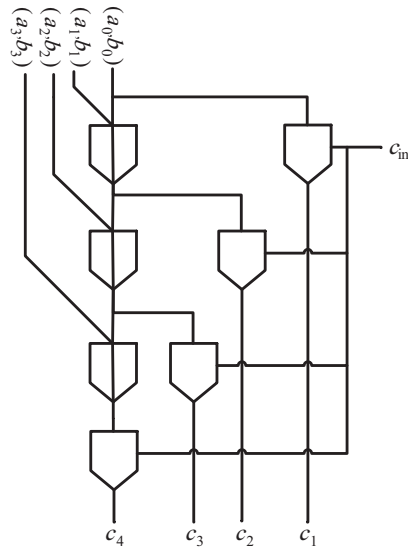


Fig. 3. FUM-based CLA logic with blocking factor 4.

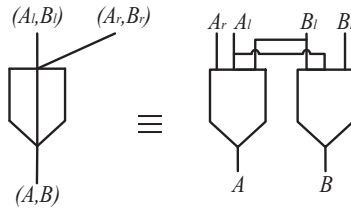


Fig. 4. Notation for, and structure of, the TM gate [8].

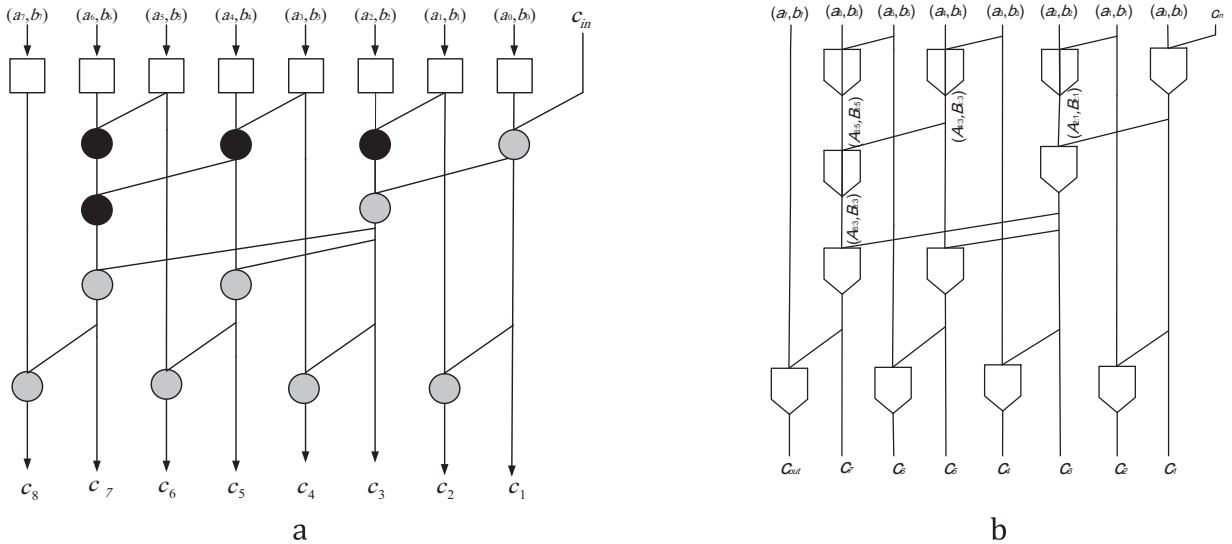


Fig. 5. BK parallel prefix network; conventional (a) and FUM-based relaxation (b).

4. Technologies for atomic-scale logic

Moore's-Law scaling of integrated circuits, a fixed staple of the IC industry for several decades, will likely end in the early 2020s [11], as the feature size approaches a small multiple of the size of a silicon atom.

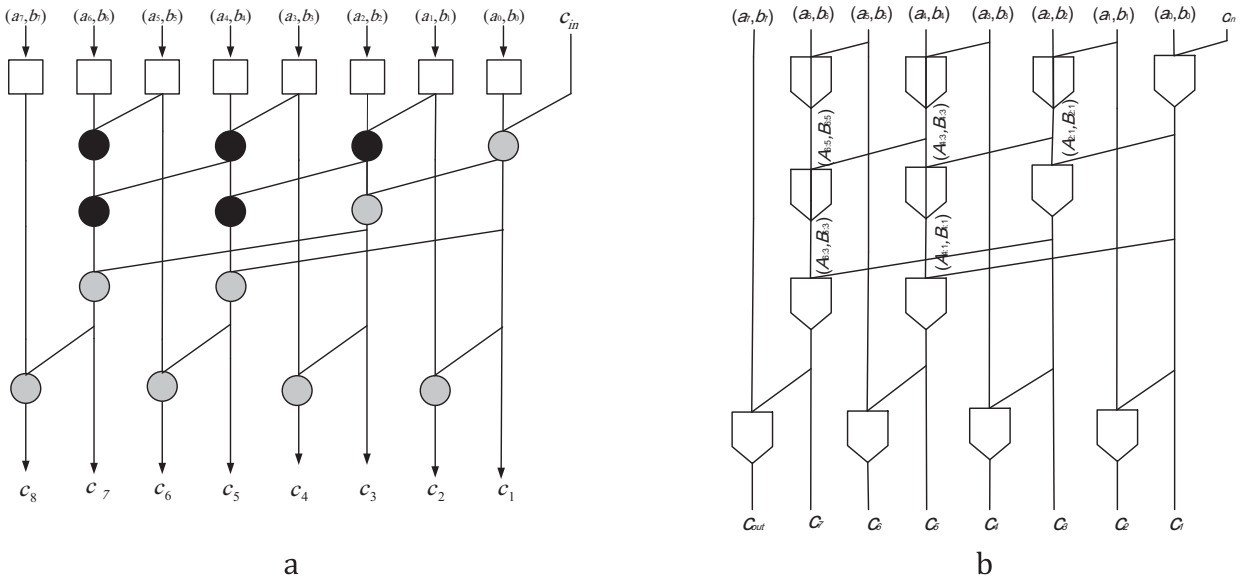


Fig. 6. HC parallel prefix network; conventional (a) and FUM-based realization (b).

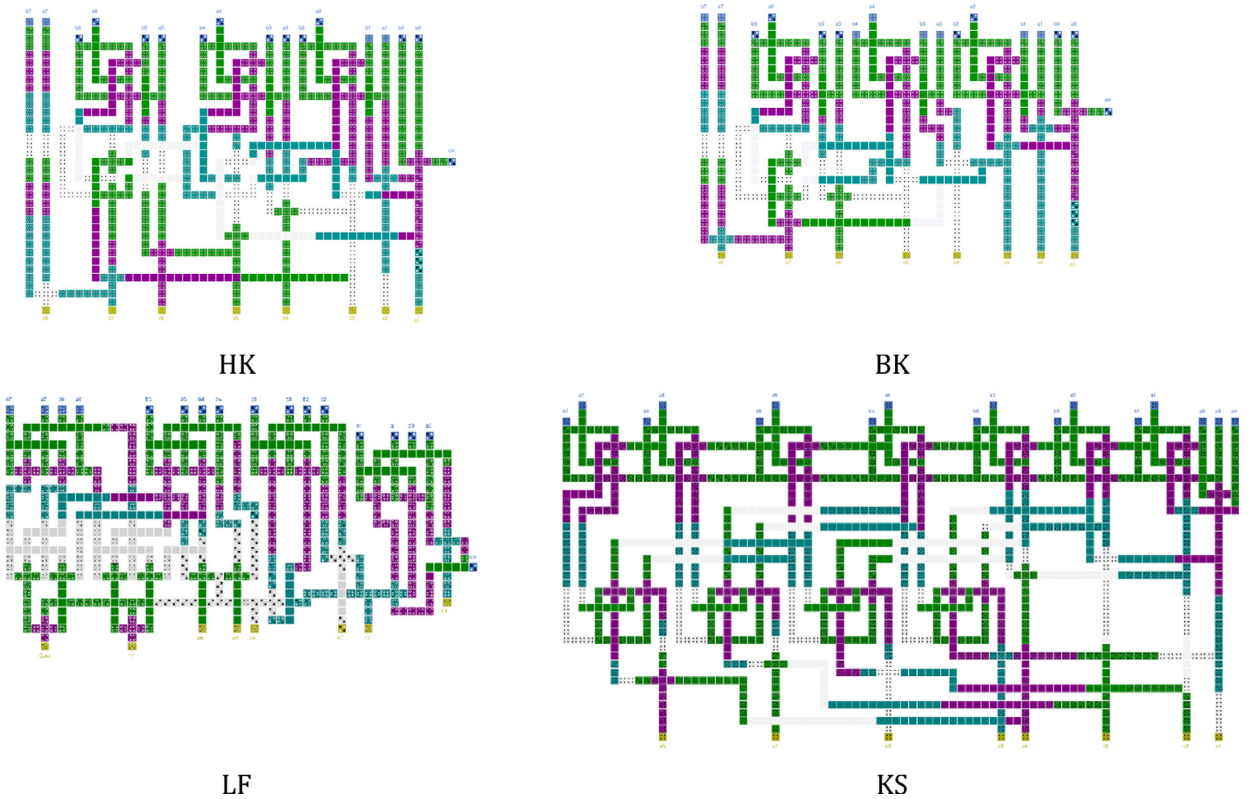


Fig. 7. QCA layouts of 8-bit LF, KS, HC, BK carry networks.

For further performance improvements, we must enter the realm of atomic-scale computation, where physical phenomena, notably quantum tunneling [12], may present roadblocks to fast, reproducible, and reliable computation. Post-quantum cryptography is a notable example for moving the research agenda to the new atomic-scale computing age.

There are a number of new emerging technologies that can pave the way for the aforementioned computations. Majority or minority elements serve as the basic gate in most of these technologies, some of which also use CMOS components (e.g., magnetic tunnel junction), while others are totally CMOS-free (e.g. quantum-dot cellular automata).

Table 3
Figures of merit for six PPNs.

PPN	Area (μm^2)	# Cells	Delay (CZ)	CDP	# \mathcal{M}		
					FUM	PUM	Total
LF	0.67	776	6	4	20	0	20
KS	1.51	1550	10	3	30	0	30
HC	0.82	839	8	4	18	0	18
BK	0.63	650	7	4	16	0	16
[9]-LF	1.77	1994	9	5	7	28	35
[10]	1.01	1143	8	5	10	9	19

After providing separate brief introductions to these technologies, we discuss fundamental architectural differences between CMOS-free and hybrid-CMOS majority gate technologies.

4.1. CMOS-free technologies

There are new emerging majority gate technologies, where circuits can be realized without any CMOS components to help in cascading majority elements, devising memory elements, or wiring complications. We provide, below, brief introductions for five selected technologies in separate subsections.

4.1.1. Quantum-dot cellular automata (QCA)

Four electron place-holders or “dots” characterize the basic QCA cell. Two injected electrons can take up positions in the four dots according to the slash or the backslash configuration (Fig. 8, from [13]). Figs. 9 and 10 show the QCA realization of the \mathcal{M} gate, and an inverter, respectively, both borrowed from [13]. The two gates constitute a complete logic set, given



Fig. 8. QCA cell configurations (left to right): Null, “1,” and “0”.

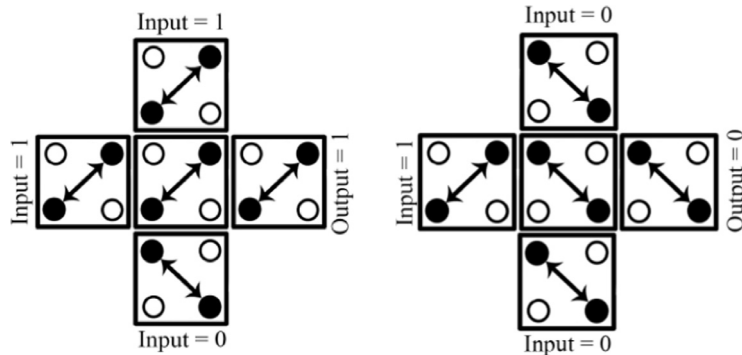


Fig. 9. Two-input QCA \mathcal{M} gate (left to right): $\mathcal{M}(1, 1, 0) = 1$ and $\mathcal{M}(0, 1, 0) = 0$.

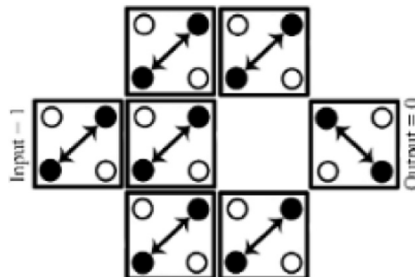


Fig. 10. QCA Inverter, designed with robustness in mind.

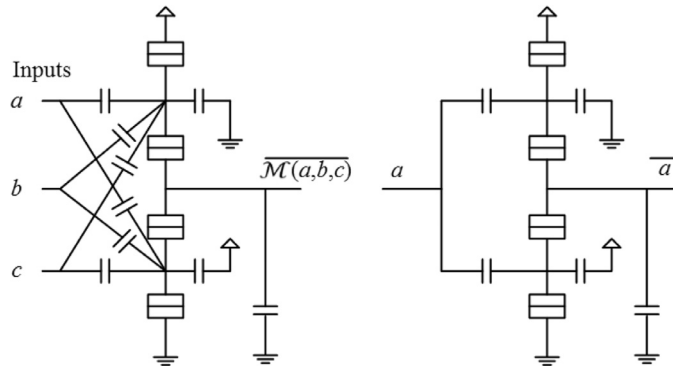


Fig. 11. SET circuits for Min (left) and inverter (right) [14].

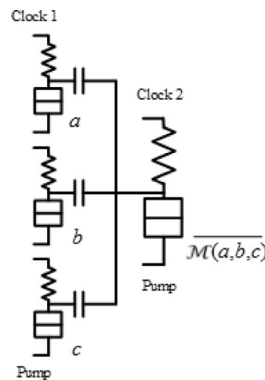


Fig. 12. Basic TPL gate.

the realizability of AND and OR functions in terms of majority gate. Mapping a 7-gate full-adder (FA) directly to QCA leads to the use of 7 partially utilized \mathcal{M} gates, whereas we can realize the same functionality with 3 fully-utilized \mathcal{M} gates and 2 inverters.

4.1.2. Single-electron tunneling (SET)

Single-carrier electronics, allowing for controlled transfer of individual electrons based on the phenomenon of single-electron tunneling, can be viewed as the ultimate in compactness and energy efficiency.

As in any new technology, we need to show the ability to build feasible logic gates in order to use the single-electron method for computation. This ability has been demonstrated for a minority element [14], shown along with an inverter in Fig. 11.

4.1.3. Tunneling phase logic (TPL)

As we see in Fig. 12, several capacitively coupled inputs feed a load capacitance in TPL, so as to realize the 3-input minority function [15] if some conditions are met. The output of a minority element is the inverse of one input when the other two inputs are opposites of each other, thus the element can also serve as an inverter. The same structure can form 3-input NAND and NOR gates.

4.1.4. Nano-scale bar magnets (NBM)

Certain difficulties with the QCA technology directed attention to NBMs. Use of magnets for computation isn't new per se, because it dates back to the early days of digital technology, but NBM magnets have microscopic size. The magnetic directions can be horizontal (on the surface) or vertical (perpendicular to the surface), with the latter option bearing some advantages.

Extreme energy efficiency and latch-free pipelining (resulting from the built-in non-volatile storage capability) constitute the main benefits of computing with nanomagnets.

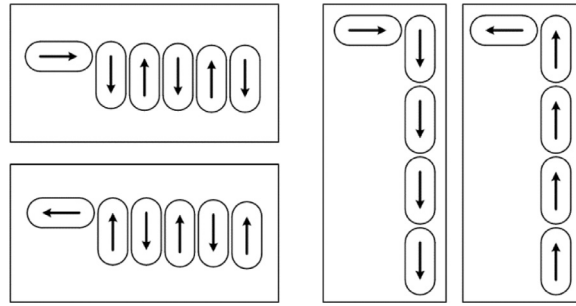


Fig. 13. Two types of nanomagnet wires.

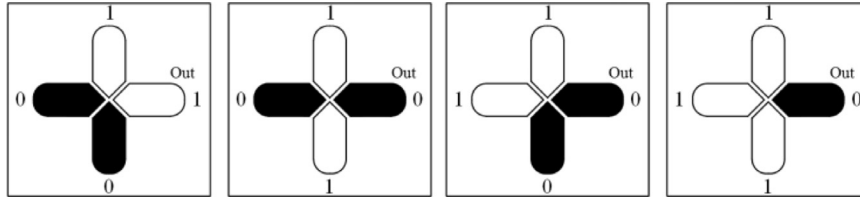


Fig. 14. Voting with nanomagnets.

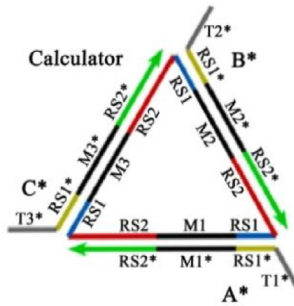


Fig. 15. DNA-based majority [16].

However, the speed of magnetic computing does not compete well with silicon-based technologies. Two types of nanomagnet-based wires are shown in Fig. 13, while Fig. 14 depicts four instances of voting resulting in a minority output (inverted majority value).

4.1.5. Deoxyribonucleic acid (DNA)

Neural brain computation in humans and animals is based on biological embodiments of the majority function [16]. DNA provides another new technology of interest, which can be used to realize 3-input majority gates [16]. Operation of these biological majority gates is based on DNA strand displacement mechanism. For example, Fig. 15 depicts the structure of a DNA \mathcal{M} gate.

4.2. Hybrid-CMOS technologies

The majority and inverter gates are realized in a CMOS-free manner in these technologies. However, circuit implementation, in general, requires CMOS components such as sense amplifiers and memory elements. Four selected technologies in this category are briefly introduced below.

4.2.1. Magnetic tunnel junction (MTJ)

MTJ, a new spintronics technology, utilizes devices with two (free and fixed) ferromagnetic thin-film layers that are separated by an oxide-tunneling barrier. The fixed layer's magnetization is not easily changeable, whereas the free layer can change magnetization readily to become aligned with, or line up in opposition to, the fixed layer. A bit is represented by the resulting low or high junction resistance [17]. Fig. 16 shows how two of these elements are combined to form an \mathcal{M} gate.

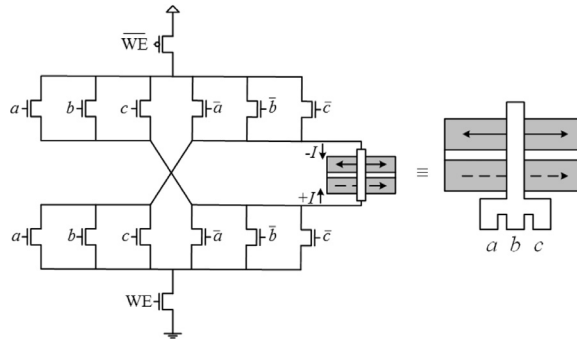


Fig. 16. Majority gate in MTJ logic [17].

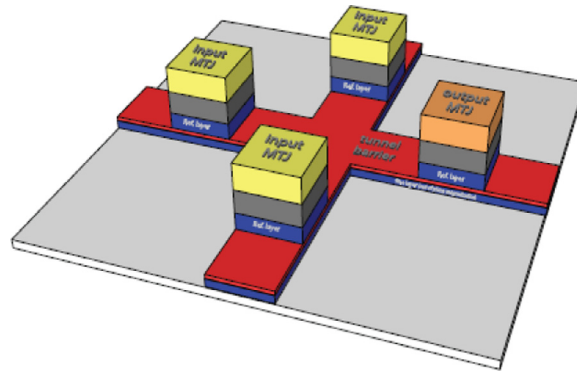


Fig. 17. Schematic of STMG. The three MTJ inputs are visible in yellow, while the output is the orange block.

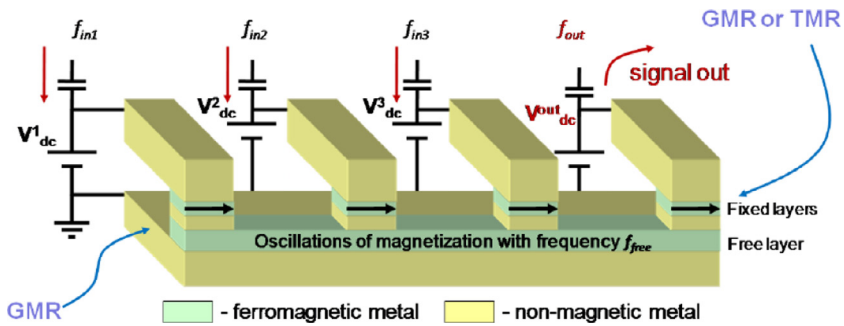


Fig. 18. Spin torque oscillator (STO) logic.

4.2.2. Spin torque majority gate (STMG)

In the MTJ-based spin torque technology, a 3-input STMG is constructed by four MTJs that are realized in spin transfer torque (STT) technology. The three \mathcal{M} -gate inputs (see Fig. 17) are written to three of the STT-MTJs, and the fourth one provides the output, which is represented by the magnetization orientation of the MTJ freelayer [18].

There are a few other spin-based technologies that efficiently implement majority gates and are not MTJ-based; namely spin torque oscillator, spin wave device, all spin logic, the first of which is briefly described below.

4.2.3. Spin torque oscillator (STO) logic

Fig. 18 depicts a 3-input current mode \mathcal{M} gate in spin torque oscillator (STO) technology, where current drivers are realized with CMOS cells, and giant magnetoresistance (GMR) is used as the substance for the three inputs and the output of the \mathcal{M} gate.

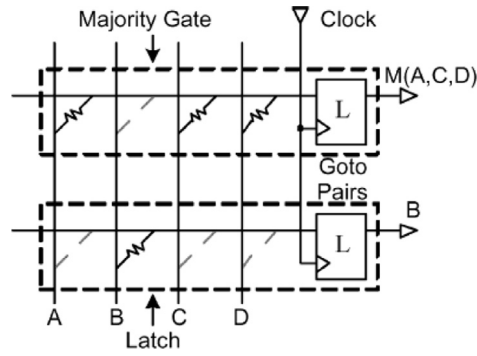


Fig. 19. Molecular switches can be used as the input resistors to a Goto pair. Since these switches are programmable, the result is a form of PMLA [20].

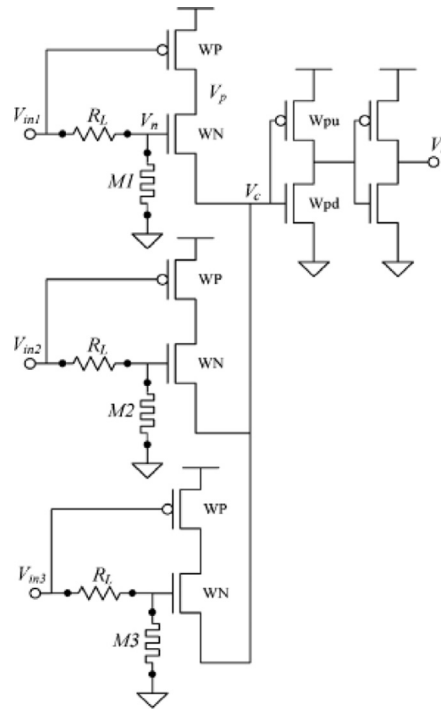


Fig. 20. Subthreshold CMOS-memristor neural circuit consisting of the synaptic circuit and a buffering stage which amplifies the summed result of the synaptic circuits [21].

4.2.4. Memristors

Useful in many applications, Memristors [19] and memristive devices are novel structures; basically variable resistors whose resistance changes with device history. While memristive devices are used primarily for storage (where resistance reflects the stored data), they can also be used as functional blocks, such as analog circuits, neuromorphic systems, and logic circuits, either by themselves or integrated with CMOS structures to perform the logical operations.

Three schemes are available for realizing majority logic with memristors; namely, the programmable majority logic arrays (PMLAs) that take advantage of the Goto pair [20], an interesting circuit that can restore signals at the nanoscale level (Fig. 19). Another one is the charge-sharing threshold gates (CSTGs), illustrated in Fig. 20. These devices take advantage of synaptic voltage dividers and CMOS circuitry for amplifying the voltage swing ΔV_n and providing an output that can be summed together with the outputs of other synapses [21]. Finally, Fig. 21 depicts a circuit of the current-mirror threshold gates (CMTGs), where a 3-input threshold gate of this type requires 14 transistors ($2h + 8$, more generally, for h inputs) [21].

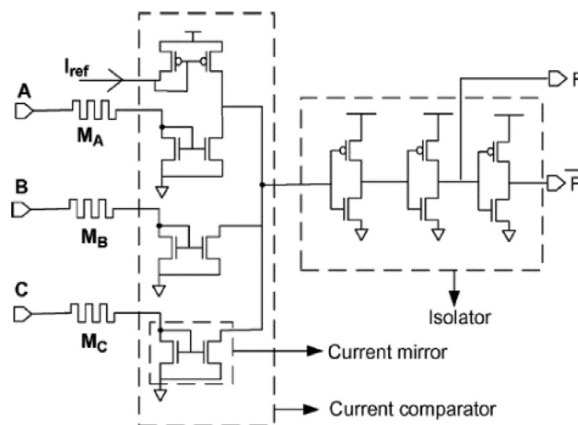


Fig. 21. A three-input CMTG using memristors as weights and I_{ref} as the threshold [21].

5. Architectural considerations in CMOS-free and hybrid-cmos circuit design

Circuit design issues include particular problems such as element cascading, fan-out, crossovers, and feedback. There are different ways and means for facing these difficulties in the CMOS-free platforms that are briefly addressed below in connection with QCA, SET, TPL, DNA, and NML nanotechnologies. However, the CMOS components in hybrid-CMOS technologies, help to overcome the aforementioned problems in the same way that are handled in pure CMOS.

5.1. Cascading

The QCA, SET, and NML technologies can be considered as cascade-friendly, with supporting evidence provided by the designs of a QCA 32-bit adder [13], a SET 16-bit adder [22], and an NML 32-bit adder [23]. In CMOS-free MTJ designs and newly developed DNA nanotechnology, however, we have not encountered any cascading of \mathcal{M} gates that is longer than two. On the other hand, combination of CMOS components and atomic scale elements may overcome the cascading problems.

5.2. Fan-out

Regarding the fan-out metric in different CMOS-free nanotechnologies, experience indicates [15] that fan-out depends on the coupling capacitance. However, the highest fan-out degree in the work just cited is 3. The highest encountered fan-out of 3 also applied to NML [23]. It appears that higher fan-out should be achievable via increasing the required energy for the clocking field [23]. QCA is capable of providing for higher fan-out than the aforementioned three nanotechnologies via incorporating the wire cells in different clock zones [24].

5.3. Crossovers

TPL, MTJ, and SET that use metallic wires realize the required crossovers via multiple layer constructs, as in CMOS and, by extension, hybrid-CMOS. In the NML, coplanar crossover is realized via four nanomagnets (two per each wire), where in the event of reliable performance of the junction no signal loss will occur [25]. Besides the multi-layer crossovers in QCA, coplanar crossovers are possible, via rotated cells or through associating clocks to vertical and horizontal wire cells with 180° phase difference.

5.4. Feedback

The feedback capability is essential for realization of sequential circuits. This problem is easily solved in NML, MTJ, and QCA, where each cell inherently acts like a delay element (memory), and it can be arranged like a chain to implement loops. However, in TPL and SET, the required feedback elements (i.e., flip-flops or latches) are actually implemented as in [26]. Regarding hybrid-CMOS designs such as MTJ-CMOS, it is usually the case that designers take advantage of MTJ as the memory elements required for feedback [27].

5.5. Case study: full-adder-based circuits

Full-adder-based circuits in CMOS, hybrid-CMOS, and CMOS-free technologies have been extensively studied, where cascading, fanout, crossover, and feedback issues can be problematic.

Four FA architectures in different technologies are depicted in Fig. 22, where the NMI, QCA, and SET FAs allow for easy cascading of majority gates, while, as pointed out in Section 5.1, cascading in MTJ is possible via sense amplifier interfaces.

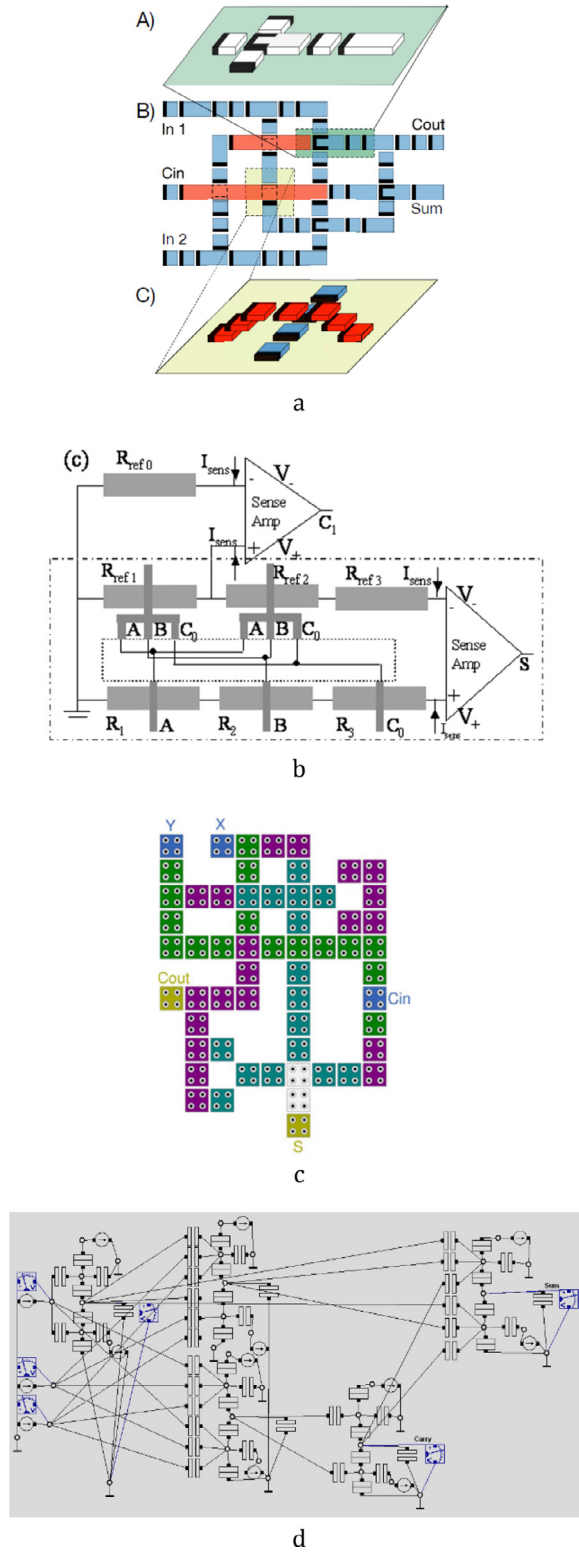


Fig. 22. M -based FA architectures in a. NML [28], b. MTJ [29], c. QCA [13], and d. SET [30] technology.

6. Conclusion and future work

Over three-quarters of a century, design of arithmetic and other functional circuits required for digital computation has undergone changes in response to technological evolution, from electromechanical relays, through vacuum tubes and discrete transistors, to various degrees of circuit integration. Emerging nanotechnologies call for another transition that may prove more difficult and disruptive to the industry than the previous changes.

It has been our purpose in this paper to present the needed background for understanding and dealing with emerging majority-friendly nanotechnologies and to offer examples of early successes in adapting the design of arithmetic circuits to them. It appears that majority (or 2-out-of-3 agreement), extending both OR (1-out-of-2) and AND (2-out-of-2) functions of standard gates, is a capability that arises rather naturally in many different domains, so we can expect additional new technologies (biologically-based or otherwise) to support its efficient realization.

We are continuing work in the domain of arithmetic circuits: Making our current designs more efficient with respect to various figures of merit and extending our approach to other arithmetic operations. These efforts must be augmented by fine-tuning and discovering new ways of dealing with the problems of cascading, fan-out, crossovers, and feedback, which we outlined in Section 5 of the paper. While success is in no way guaranteed, the record of the computer industry, and scientists/engineers who support it, is stellar in overcoming obstacles and making progress beyond even the boldest predictions.

We also plan to investigate whether one can leverage the majority function provided by the circuits under consideration to realize fault tolerance in triple-modular redundancy schemes, applying the redundancy in space, time, or a combination of the two. Given that the majority voting function needs only one \mathcal{M} -element, overhead will be quite minimal beyond triplication in time or space, making fine-grain replication and voting quite feasible. Such fine-grain replication would offer greater improvements in reliability compared with module-level replication and voting. It is unknown at this time whether the reliability gain from triplication is worth the circuit cost and the partial loss of energy-efficiency, which is one of main motivations for pursuing majority-friendly logic circuits.

One class of circuits that are useful in implementing arithmetic circuits is that of parallel counters/compressors. A full-adder is a (3, 2)-counter, receiving three single-bit inputs and generating their 2-bit sum. Any $(n, \log_2 n)$ -counter can be implemented by using FAs in efficient assemblies. However, just as in standard CMOS the use of certain larger building blocks, notably (4, 2)-counters, leads to better speed, cost, and power-use parameters, one may be able to identify other efficient higher-level primitives for majority-based circuits. We hope to be able to explore the virtually-endless design space in this area.

Declaration of Competing Interest

None.

Acknowledgements

Dr. Jaberipur's research was supported in part by two sources: IPM under Grant CS1398-2-03 and Shahid Beheshti University.

References

- [1] Denning PJ, Lewis TG. Exponential laws of computing growth. *Commun ACM* 2016;60(1):54–65.
- [2] Mittal S. A survey of techniques for approximate computing. *ACM Comput Surv* 2016;48(4):62.
- [3] Muroga S. *Threshold logic and its applications*. Wiley; 1971.
- [4] Beiu V, Quintana JM, Avedillo MJ. VLSI implementation of threshold logic: a comprehensive survey. *IEEE Trans. Neural Networks* September 2003;14(5).
- [5] B. Parhami, *Dependable computing: a multilevel approach*, forthcoming book, available on-line: https://www.ece.ucsb.edu/~parhami/text_dep_comp.htm.
- [6] Isbell JR. Median algebra. *Trans. American Mathematical Society* August 1980;260(2):319–62.
- [7] Kong K, Shang Y, Lu R. An optimized majority logic synthesis methodology for quantum-dot cellular automata. *IEEE Trans. Nanotechnology* 2010;9(2):170–83.
- [8] Jaberipur G, Parhami B, Abedi D. A formulation of fast carry chains suitable for efficient implementation with majority elements. In: *Proc. IEEE 23rd Symposium on Computer Arithmetic (ARITH)*; 2016. p. 8–15.
- [9] Pudi V, Sridharan K. Low complexity design of ripple carry and brent-kung adders in qca. *IEEE Trans. Nanotechnology* January 2012;11(1):105–19.
- [10] Pudi V, Sridharan K. New decomposition theorems on majority logic for low-delay adder designs in quantum dot cellular automata. *IEEE Trans. Circuits and Systems II* October 2012;59(10):678–82.
- [11] Volger D. The roadmap to 5nm: convergence of many solutions needed. *Semiconductor Industry Association 2016 on-line document*, retrieved on October 4.
- [12] Razavy M. *Quantum theory of tunneling*. World Scientific; 2003.
- [13] Abedi D, Jaberipur G, Sangsefidi M. Coplanar full adder in quantum-dot cellular automata via clock-zone based crossover. *IEEE Trans. Nanotechnology* May 2015;14(3):497–504.
- [14] Iwamura H, Akazawa M, Amemiya Y. Single-Electron majority logic circuits. *IEICE Trans. Electronics* January 1998(1):42–8 Vol. E81-C.
- [15] Fahmy HAH, Kiehl RA. Complete logic family using tunneling-phase-logic devices. In: *Proc. 11th Int'l Conf. Microelectronics*; 1999. p. 153–6.
- [16] Li W, Yang Y, Yan H, Liu Y. Three-Input majority logic gate and multiple input logic circuit based on dna strand displacement. *Nano Lett.* 2013;13:2980–8.
- [17] Bromberg DM, Morris DH, Pileggi L, Zhu J-G. Novel stt-mtj device enabling all-metallic logic circuits. *IEEE Trans. Magnetics* November 2012;48(11):3215–18.
- [18] Nikonov DE, et al. Proposal of a spin torque majority gate logic. *Electron Device Letters, IEEE Aug.* 2011;32(8):1128–30.

- [19] Chua LO. Memristor—The missing circuit element. *IEEE Trans. Circuit Theory Sep.* 1971;18(5):507–19.
- [20] Kvatinsky S, Satat G, Wald N, Friedman EG, Kolodny A, Weiser UC. Memristor-Based material implication (IMPLY) logic: design principles and methodologies. *IEEE Trans. VLSI Systems* October 2013;22(10):2054–66.
- [21] Rose GS, Rajendran J, Manem H, Karri R, Pino RE. Leveraging memristive systems in the construction of digital logic circuits. *Proceedings of the IEEE* June 2012;100(6):2033–49.
- [22] Sulieman, Beiu V. Characterization of a 16-bit threshold logic single-electron technology adder. In: *Proc. IEEE International Symp. Circuits and Systems*; 2004. pp. III-681.
- [23] Alam M, et al. Experimental progress of and prospects for nanomagnet logic (NML). In: *Proc. IEEE Silicon Nanoelectronics Workshop*; 2010.
- [24] Yadavalli KK, Orlov AO, Timler JP, Lent CS, Snider GL. Fanout gate in quantum-dot cellular automata. *Nanotechnology* 2007;18(37):375401.
- [25] Pulecio Javier F, Bhanja Sanjukta. Magnetic cellular automata coplanar cross wire systems. *J Appl Phys* 2010;107(3):304–8.
- [26] Lageweg Casper, Cotofana Sorin, Vassiliadis Stamatis. Single electron encoded latches and flip-flops. *IEEE Trans Nanotechnol* 2004;3(2):237–48.
- [27] Oosawa S, Konishi T, Onizawa N, Hanyu T. Design of an STT-MTJ based true random number generator using digitally controlled probability-locked loop. In: *IEEE 13th International Conf. New Circuits and Systems (NEWCAS)*; 2015. p. 1–4.
- [28] Cairo F, Turvani G, Riente F, Vacca M, Breitzkreutz-V Gamm S, Becherer M, Graziano M, Zamboni M. Out-of-plane nml modeling and architectural exploration. In: *15th International Conference on Nanotechnology (IEEE-NANO)*; 2015. p. 1037–40.
- [29] Meng H, Wang J, Wang JP. A spintronics full adder for magnetic CPU. *IEEE electron device letters* 2005;26(6):360–2.
- [30] Sulieman MH, Beiu V. On single-electron technology full adders. *IEEE Trans Nanotechnol* 2005;4(6):669–80.

Behrooz Parhami (PhD, UCLA, 1973; MS, Oregon State University, 1970; BSEE, Tehran University, 1968) is Professor of Electrical and Computer Engineering, and former Associate Dean for Academic Personnel, College of Engineering, at University of California, Santa Barbara, where he teaches and does research in computer arithmetic, parallel processing, and dependable computing. A Life Fellow of IEEE, a Fellow of IET and British Computer Society, and recipient of several other awards (including a most-cited paper award from *J. Parallel & Distributed Computing*), he has written six textbooks and more than 300 peer-reviewed technical papers. Professionally, he has served on journal editorial boards (including multiple IEEE Transactions) and conference program committees and is also active in technical consulting.

Dariush Abedi received his B.S. degree in hardware computer engineering from the Department of Computer Engineering and Information Technology, Sadjad University of Technology, Mashhad, in 2011, and the M.S. degree in computer engineering from the Department of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, in 2015. He is currently pursuing the Ph.D. degree in Computer Architecture from Shahid Beheshti University. His-research interests include Computer Arithmetic, Nano-technology and Embedded System Design.

Ghassem Jaberipur, is a Professor of Computer Engineering in the Department of Computer Science and Engineering of Shahid Beheshti University, Tehran, Iran. He received his BS in electrical engineering and PhD in computer engineering from Sharif University of Technology in 1974 and 2004, respectively, (where he's been recognized as one of the 50 distinguished graduates for years 1966–2016), MS in engineering from UCLA in 1976, and MS in computer science from University of Wisconsin, Madison, in 1979. His-main research interest is in computer arithmetic. dr. Jaberipur is also affiliated with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), in Tehran, Iran.