Keynote

The Emerging Majority: Technology and Design for Superconducting Electronics

Giovanni De Micheli
EPFL, 1015 Lausanne, Switzerland

Editor's notes:
Emerging computing technologies will enable design of energy-efficient circuits and systems in the future. This article presents a tutorial on the role of emerging technologies in designing green computing systems.
—Partha Pratim Pande, Washington State University

Electronic circuits are ubiquitously used in computing, communication, and consumer applications and are affecting everyone's life. The rise of artificial intelligence (AI) has fueled the design and manufacturing of new integrated circuits and processors. It is expected that the demand for performance of electronic circuits is going to increase without a foreseeable limit. Conversely, the energy consumption per task (e.g., performing an electronic transaction) cannot increase significantly from current levels and indeed it should decrease. In 2020, about 5% of the world energy consumption (and related CO_2 emissions) was spent in data servers and storage, and it is projected to be between 8% and 21% by 2030 [1] according to policies and innovations introduced in the current decade. Thus, environmental protection requirements motivate the research into *green electronics* and in all various means to make the cyberspace energy requirements sustainable. complementary metal–oxide–semiconductor (CMOS) technology, the main workhorse in electronic systems, is showing increasingly higher fabrication costs and challenges in downscaling transistor dimensions, with marginal improvements in energy consumption at smaller technology nodes. It is well perceived that CMOS technology has reached the plateau.

The search for emerging technologies that can replace and/or supplement CMOS has been ongoing for over a decade. In 2019, Shulaker et al. [2] described the design, fabrication, and successful test of a 16-bit RISC V processor in carbon nanotube (CNT) technology. CNTs can outperform silicon in speed, power consumption, and compactness due to the intrinsic higher mobility of the material when compared to silicon. Two-dimensional electronic materials [3] have shown to be interesting contenders for future electronic technologies. While applications remain still

Digital Object Identifier 10.1109/MDAT.2021.3095046
Date of publication: 6 July 2021; date of current version: 6 December 2021.

limited (e.g., only a single-bit processor has been fabricated in MoS_2), 2-D layers of functional materials can be used (such as CNTs) to form 3-D stacks for high-performance energy-efficient computing. A growing family of computing devices leverage optics and wave propagation/interaction (e.g., spin wave and plasmonics) and exploits the interference of waves to perform computation. Recent efforts in superconducting electronics (SCEs) circuit design have shown the ability of meeting unprecedented design objectives, in terms of performance and energy efficiency, despite the energy cost of cooling down the devices.

From a design standpoint, it is important to discern two important characteristics of any emerging technology. The former is related to the competitive advantage over CMOS, the application domain, and the compelling reason to use it. In view of the accumulated experience in CMOS technology, it will still take time to establish new technologies (or technology layers) than can compete with and/or displace CMOS. Most likely these new technologies will complement CMOS by supporting the realization of application-specific accelerators. The latter characteristics is the ease of design, that is, the availability of design tools and methods to deal with new emerging technologies. For any technology to emerge, design flows have to be available to design, verify, and evaluate scalable architectures before setting up manufacturing lines. It is yet unclear how much current design flows will have to change, and whether commercial electronic design automation (EDA) providers will support emerging technologies and which ones.

Recent research has shown that several emerging technologies have devices that can be modeled by *majority functions*, that is, odd-fanin logic gates that yield a TRUE output when a numerical majority of the inputs is TRUE (Figure 1). This model is valid for at least three families of circuits: 1) double-gate controlled-polarity transistors in silicon, CNTs, and 2-D materials such as WSe_2 [4]; 2) technologies based on wave propagation and interaction, such as spin-wave and plasmonics; and 3) SCE circuits. For these families of devices, it has been shown that better designs (in terms of performance, power, area—PPA) can be achieved with models, synthesis, and verification algorithms exploiting the majority paradigm in logic synthesis

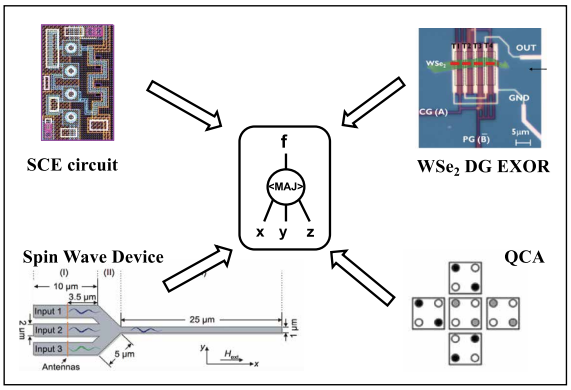

Figure 1. Several emerging technologies can be modeled by majority functions.

[5], [6]. Although applications to various emerging technologies was shown before [7], the benefits of using majority-logic synthesis algorithms and tools for superconducting circuits was mentioned but never described in detail in the literature.

This work outlines the rationale and current status of SCE, as a technology with strong potentials to support high-performance large-scale computing within a limited energy budget. Majority-based logic design and optimization is then described in its generalities, as well as its applications to the SCE design. The joint potentials of both the manufacturing and design technologies make SCE a strong contender for the realization of computing systems.

Superconducting electronics

SCE is a branch of engineering that leverages computation at few degrees Kelvin (typically 4K) where resistive effects can be neglected and where switching is achieved by Josephson junctions (JJs). Current difficulties in downscaling CMOS have made SCE quite attractive for the following reasons. First, the technology can match and extend present performance requirements, for example, ALU prototypes have been shown to run at and above 50-GHz clock rates. Second, information theory states that the minimum transition energy per bit is $E_{\text{bit}} = kT \ln 2 = 4 \times 10^{-21}$ J (at 300 K). In practice, CMOS circuits require energy in the range 10^5–10^6 E_{bit}. SCE devices manipulate flux quanta $\phi = h/2e$ with energy 2×10^{-19} J or equivalently 5×10^3 kT ln 2 at 4 K. Thus, SCE circuits can be realized to operate much closer to the minimum energy limit and roughly two orders of magnitude better when compared to CMOS. This prediction is confirmed by prototypes [8]. Third,

today current superconductor circuits are designed in a 250-nm technology, much easier to realize in integrated fashion (when compared to 5-nm CMOS) and with a potential horizon of a 10–50× possible downscaling, thus projecting one or two decades of further improvement. Finally, although cooling energy is a major drawback for SCE, cryocooling efficacy is expected to improve as well. Therefore, SCE is a strong candidate for high-performance large system designs in the coming decade.

It is important to note that SCE differs significantly from quantum computing (QC). Indeed, SCE operates at 4K, where resistive effects vanish and it is based on superconductive inductive loops and JJs. Logic design follows the principles (but differs in the practice) of classic Boolean logic. QC operates at 10 mK for noise reasons and leverages superposition and entanglement provided by the quantum properties of the material. Thus, QC design is based on a different (reversible) logic abstraction and on logic gates with different properties. Nevertheless, QC requires interfacing to classical host computers through channels that may exploit SCE and/or CryoCMOS [9] circuits. As a result, SCE design is also relevant to the design and engineering of QC, beside other self-standing applications.

IBM took a strong effort in SCE in the 1970s with the objective of building computers that would outperform the currently available technology. The circuits utilized JJ exhibiting hysteresis in their resistive states (i.e., resistive and superconductive). The JJ acts as a switch that can be set and reset by applying a current. A logic TRUE is associated with the JJ in its resistive state, and a logic FALSE with its superconductive state. This effort faded in the mid-1980s, because of various drawbacks, including the choice of materials and the latching operation of logic [10].

Likharev and Semenov [10] brought back strong interest in SCE by proposing rapid single flux quantum (RSFQ) circuits. In these circuits, the logic values (TRUE, FALSE) are represented by the presence or absence of single flux quantum (SFQ) pulses called *fluxons* with $\phi = h/2e = 2 \times 10^{-15}$ Wb corresponding approximately to a 2 mv pulse lasting 1 ps. Junctions are DC biased and when a pulse is applied to the junction, it can be sufficient to drive the current level over its threshold and to generate another pulse that can be propagated through the circuit. This type of behavior is often called Josephson transmission line (JTL) and it is the basic operational principle of RSFQ circuits that conditionally propagate flux pulses. A specific feature of RSFQ circuits is that logic gates are clocked and that the overall circuit is pipelined. The RSFQ technology evolved in many directions. Energy-efficient SFQ (eSFQ and ERFSQ) [11] and low-voltage RSFQ (LV-RSFQ) [12] employ specific bias networks and low supply voltages, respectively, to reduce the power consumption. Dynamic SFQ (DSFQ) logic [13] introduces self-resetting gates that ease the clocking requirements. Various realizations of ALUs have been reported, with deep-pipelined, wave-pipelined, and asynchronous operation.

SCE circuit design has several peculiarities and constraints that may vary in different SCE families. We highlight two constraints. First, each gate is triggered by a clock or bias signal in conjunction with the logic input signal. Thus, circuits operate in *pipelined* mode, and input to logic gates have to be present simultaneously, thus requiring that logic inputs have the same *logic depth* or distance from the primary inputs. A circuit with such a property is said to be *balanced*. Second, logic gates generate pulses that cannot sustain multiple fanouts and thus *splitters* have to be used. As a result, SCE design requires specific electronic design tools. The Coldflux project [14], under the auspices of the IARPA Supertools program, has addressed design EDA problems for SCE, including automatic circuit balancing [15], [16] and splitter insertion [17]. Researchers at Synopsys recently published the results of a full synthesis of a 4-bit AMD 2901 microcontroller from RTL code to layout in an ERFSQ standard cell library from Hypres [18].

Recent research work has addressed technologies that target low-energy consumption. This can be achieved by using adiabatic mode of operation and AC power (i.e., alternating current supply). Two technologies are particularly relevant: reciprocal quantum logic (RQL) [8] researched and developed at Northrop Grumman, and adiabatic quantum flux parametron[1] (AQFP) [20] pursued at Yokohama National University (YNU) in Japan. We describe just the latter in more detail.

The fundamental element in AQFP is the clocked buffer, as shown in detail in Figure 2. Two loops, involving each a JJ and an inductor, are used to store logic information in terms of flux quanta depending on the direction of an input current signal and the magnetic coupling to other inductors. When an input and the

[1] A parametron is a resonant circuit with a nonlinear reactive element [19].

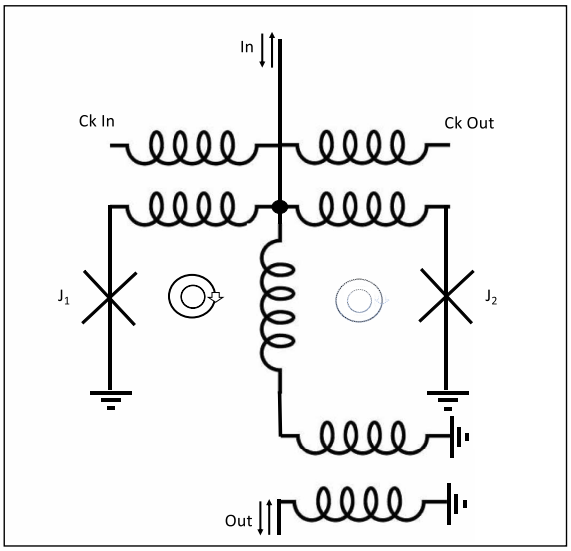

Figure 2. Buffer in AQFP technology. The buffer forwards the down/upward input current signal to the output representing TRUE/FALSE information, respectively. The input has to arrive when the JJ loops are activated by the clock signal. Thereafter, either the left or right loop stores the TRUE or FALSE information, respectively.

supply trigger are present, an output current pulse is generated. The direction of the current pulse encodes the logic value TRUE or FALSE. A buffer can be made into an inverting buffer by switching the terminals of the output coupled inductor. Thus, inversion comes at no cost in this technology. It was shown [21], [22] that the "parallel combination" of three AQFP buffers yields a majority gate (Figure 3), which is the basic logic primitive of this technology. The 2-input logic AND and OR gates can be realized by modifying one buffer (of the majority gate) so that a small imbalance in the loop design yields always a logic FALSE or TRUE as output, respectively. Based on these principles, a simple and modular cell library can be built from buffers (regular, inverted, or modified) and branch cells (used to join and split signals) [22]. A full EDA flow from an HDL description to a cell-based physical design has been created by researchers at YNU [23]. In particular, tools for logic synthesis need to address balancing, majority-based synthesis (described in the next section), and splitter insertion [24]–[26].

Majority logic

The majority function, denoted by <a,b,c>, has been studied extensively since the 1960s, as it

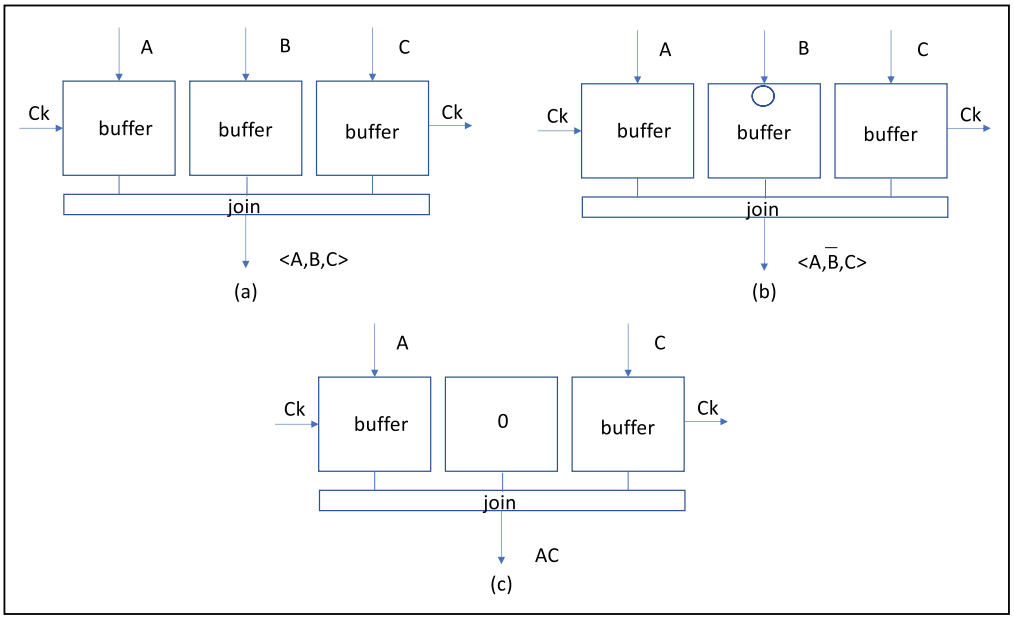

Figure 3. (a) Majority gate consisting of three buffers and a passive inductive-based join block. (b) Majority gate with inverted input. Inversion is hard-coded in a cell by exchanging the terminals of the output coil. (c) AND gate realized in pulse-based logic by using a modified buffer that yields always the FALSE value. An OR gate is realized similarly.

Commutativity: $\langle x, y, z\rangle = \langle y, x, z\rangle = \langle z, y, x\rangle$

Majority: *if* $(x = y)$, $\langle x, y, z\rangle = x$ = *y else* $\langle x, y, z\rangle = z$

Associativity: $\langle x, u, \langle y, u, z\rangle\rangle = \langle z, u, \langle y, u, x\rangle\rangle$

Distributivity: $\langle x, y, \langle u, v, z\rangle\rangle = \langle\langle (x, y, u\rangle, \langle x, y, v\rangle, z\rangle$

Self-duality: $\langle x, y, z\rangle' = \langle x', y', z'\rangle$

Figure 4. Axioms of majority algebra.

corresponds to the carry function in adders and plays a role in arithmetic unit design. Knuth stated that majority "is probably the most important ternary operation in the entire universe, because it has amazing properties that are continuously being discovered and rediscovered" [27]. Despite the theoretical interest, the use of majority and ternary functions in modern logic synthesis was limited for many years. It is quite interesting to note that the realization of new devices during the last decade (e.g., polarity-controlled transistors in silicon and other materials) led to revisiting ternary operators [28] and the majority function. In turn, the creation of new algorithms and tools based on the majority function has further enabled emerging technologies, such as SCE.

The *majority paradigm* in logic synthesis [29], [30] is based on a formulation of a new Boolean algebra using the majority and complementation operators. The algebra can be expressed in terms of five axioms (commutativity, associativity, distributivity, majority, and self-duality) and was shown to be sound and complete (Figure 4) [30]. From a theoretical standpoint, algorithms based on the majority paradigm enable the search for an optimum solution in a connected design space, which provides the existence of a path to the optimum (even though such path may be hard to find and with over-polynomial length, as the problem is computationally intractable). From a practical standpoint, tools based on the majority algebra were shown to achieve circuits 15% better in delay in average when compared to other methods after physical design in ASICs [30]. This fact was also validated on commercial tools.

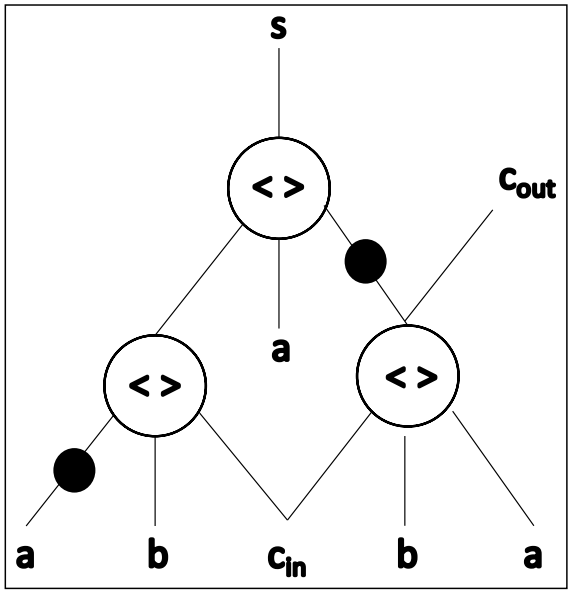

Figure 5. MIG representation of a full-add circuit. Majority is denoted by <> and bubbles on edges represent inversions.

Logic optimization starts by casting a circuit representation in terms of a logic network (LN) graph, where nodes are combinational functions and edges may represent complementation by bubbles. Majority inverter graphs (MIGs) are LNs where nodes are majority-n functions (Figure 5). Traditionally, MIGs are restricted to $n=3$, but extensions to odd $n>3$ for all (or some) nodes is possible. Typical design objectives are the area and latency reduction of the LN model that relate to the (weighted) number of nodes in the graph model and to the critical path length, which are called *size* and *depth*, respectively.

Logic optimization of MIGs can be achieved by algebraic and by Boolean methods, as in classical logic synthesis. The former methods [29] entail an *algebraic rewriting* of MIGs through the use of rules that relate to the fundamental axioms of majority logic and combination thereof. These methods have shown to be highly effective in reducing the size and depth of logic circuits, also on large-scale CMOS benchmarks (Figure 6). The latter approach, the Boolean methods, is still under investigation and related to recent results in *Boolean substitution*. In simple terms, a computational node is simplified by using a new input from the LN that has already computed "a part" of the function. Another Boolean optimization method is based on exploiting the error-correction property of the majority function. For example, subnetworks that fanin to a majority gate can be simplified by introducing "errors" (i.e., variations from the original specs), as long as these errors are masked by the majority operation [30].

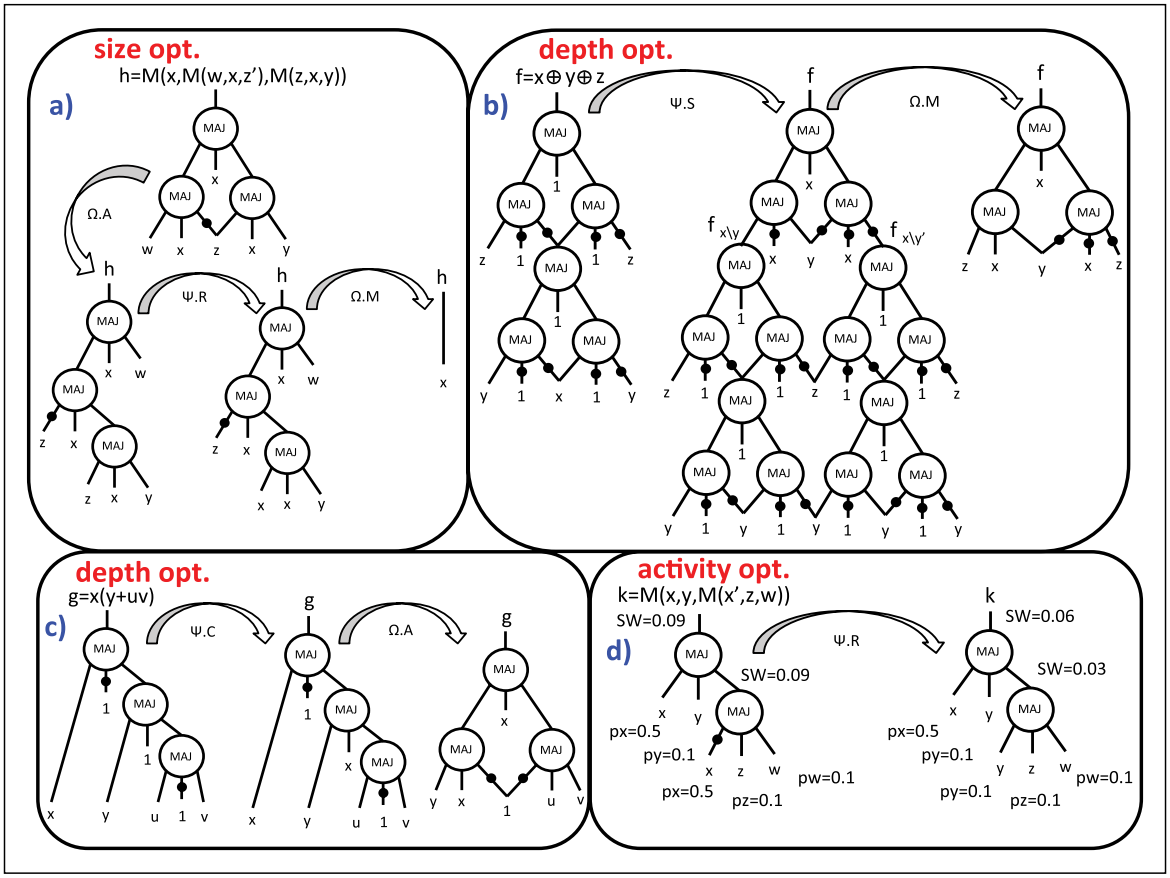

Figure 6. Example of algebraic rewriting using MIGs. (a) Sequence of transformations leading to a minimum area circuit. (b) and (c) Depth optimization. (d) Power estimate optimization. Source: [30].

Logic SCE circuits in some families, like AQFP and RQL, realize the majority function in a simple and direct way. Thus, the majority abstraction is the native model that can be used fruitfully for their design and optimization. Nevertheless, a direct application of majority logic synthesis tools to SCE is not straightforward, as SCE circuits operate in pipelined mode and the circuit has to be balanced. Thus, restructuring logic circuits arbitrarily, by adding/removing stages to/from logic paths, may reduce logic complexity but introduce computational errors.

A simple solution to SCE circuit synthesis is to structure the LN according to an arbitrary synthesis method (while optimizing a desirable objective), determine the longest path from each input to any node, and eventually add buffer nodes to selected edges so that any path to any node is as long as the longest path. This technique, called *padding*, can be refined by using *retiming* to shift the buffers and reduce their count while preserving correct operation [15]. A dynamic programming algorithm for the optimization of buffers in technology mapping was recently presented [16]. In practice, the number of buffers used in padding may be very high, and experience shows that it is important to reduce the depth of the circuit as a first objective, as all I/O path lengths are reduced as a side effect.

Overall, logic synthesis for SCE circuits is complicated by two constraints: 1) generate a balanced network and 2) support multiple fanout through splitters. Specific features of some SCE technologies can make logic synthesis easier or harder at the same time. For example, AQFP gates can realize the majority function with input and/or output complementation. Thus, any logic function that is negation-permutation-negation (NPN)-equivalent to the majority function can be implemented by

an AQFP majority gate. In other words, the presence of inversions (i.e., bubbles on the LN graph) is inconsequential.

Fanout management requires specific attention in all SCE technologies, as output pulses are small enough that cannot be subdivided on wires in a straightforward way. Splitters are small cells that duplicate signals and need to be inserted at multiple fanout points in a similar vein as buffer trees are used in CMOS. Although some technologies (e.g., RFSQ, eSFQ, and ERSFQ) do not require splitters to be buffered, AQFP requires clocked buffers in conjunction with splitters. This complicates logic synthesis because logic optimization, fanout management, and network balancing have to be considered simultaneously. Exact and heuristic formulations for the overall problem are being studied. Some researchers addressed the splitter and buffer insertion in AQFP [24]–[26], while others considered a flow where the LN is reduced first for depth using algebraic methods, followed by Boolean substitution and splitter insertion [26]. Overall, the logic synthesis problem is not yet solved completely because: 1) the interaction between logic optimization of balanced circuits is not yet fully coupled with splitter/buffer inversion; 2) the potential of using wide (i.e., $n > 3$) majority gates is still untapped; and 3) there is still not a consensus on how to architect registers (that can be observable/controllable for other reasons, including testing), even though each buffer is itself a register. In other words, much of the potentials of AQFP, as for other SCE technologies, can be leveraged by sequential logic synthesis that can cope with register assignment and appropriate register clocking to both ease the interface to combinational logic in terms of balancing and combine locally synchronous operation with asynchronous communication among logic blocks [31].

Wave pipelining in SCE

Wave pipelining (WP) [32] is a technique to speed up the computation by allowing two or more waves of signals to propagate in between two registers. In a WP circuit, the clock frequency of the registers can be higher than the maximum propagation delay, to capture wave-fronts of data as they propagate from the source to the sink register. It is quintessential that the waves do not mix, which implies that I/O paths need to have the same delay, or to mismatch by a small quantity that eventually poses a bound on the clock frequency. Wong pioneered algorithms for designing WP circuits [33] and applied them to the design of floating-point processing units in BiCMOS technology. These algorithms consisted of two techniques: a rough tuning step that inserts a minimal number of delay padding elements and a fine-tuning algorithm that adjusts tail currents to equalize delays. There are examples of RSFQ ALU designs that exploit WP [34]. In standard (synchronous) RSFQ, the clock triggers the computation at logic gates, however, in asynchronous wave-pipelined RSFQ, signals are held so that a logic stage does not start operating until all signals from the previous stage are available. This obviates the local clocking [35] and enables multiple data waves to propagate simultaneously. As the overall performance is limited by the signal arrival-time mismatches, then SCE can benefit from path delay equalization as in CMOS WP [32] and furthermore WP path balancing can be combined with majority logic synthesis transformations, to achieve correct and optimal SCE digital circuits.

Outlook

Recent realizations of SCE circuits have shown remarkable performances. For example, Ke et al. [36] showed the realization of a low-power 8-point, 7-bit fast Fourier transform (FFT) processor running at 47.8 GHz consuming 5.3 mW in SFQ technology. An AQFP adiabatic processor has been realized [37] with switching energy at 1.4 zJ per operation and a 5 GHz AC clock. Even by considering a 1000× energy loss in cryocooling, this realization is still two orders of magnitude more efficient when compared to 7-nm CMOS according to [37]. These very positive results make us very optimistic about the potentials of SCE as a superconducting technology, especially for low-energy high-throughput computation. Nevertheless, scaling up SCE design is challenging, as the support of EDA tools is still in its infancy.

Majority-based logic synthesis has shown to be very effective in various domains, such as CMOS and spin-wave circuits. In the former case, its power has been validated by both academic and commercial tools. Libraries of algorithms for logic optimization are publicly available, such as the *mockturtle* library (https://github.com/lsils/mockturtle). The application of the majority-based synthesis paradigm to SCE

circuit design—in combination with buffering and splitting—is still under study now, but it is expected that the native majority model of SCE circuits will lead to faster and more compact designs. Indeed, the combination of advanced design algorithms and tools with SCE can yield a key solution to *exascale computing*, which will empower the scientific community to address important scientific and societal challenges

From a broader point of view, many electronic systems process a large amount of data that comes from various sources, such as sensors. The inherently inexactness of input data lead us often to design systems that are tolerant to input variations. The majority-based paradigm fits well the design of digital systems where data has to be considered in aggregate, and where decisions are taken—at the logic and system levels—on a majority of inputs. In other words, this design style leads to more robust design and operation.

The evolution of the computational landscape requires a shift in paradigm, to deliver high performance within a limited energy budget. New emerging technologies, such as SCE, may be the key to perform such a leap forward. Nevertheless, the complexity of designing in this technology requires models, algorithms, and tools that fit the circuits and that enable their optimization. Majority-based designs, combined with superconducting technology, can be the key to our computing future. ■

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF) grant "Supercool: Design methods and tools for superconducting electronics" under Grant 200021_1920981.

References

[1] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–166, Sep. 2018.

[2] G. Hills et al., "Modern microprocessor built from complementary carbon nanotube transistors," *Nature*, vol. 572, no. 7771, pp. 595–602, Aug. 2019.

[3] G. V. Resta et al., "Devices and circuits using novel 2-D materials: A perspective for future VLSI systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1486–1503, Jul. 2019.

[4] P. E. Gaillardon et al., "Nanowire systems: Technology and design," *Philos. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 372, Mar. 2014, Art. no. 20130102.

[5] O. Chen et al., "Adiabatic quantum-flux-parametron: Towards building extremely energy-efficient circuits and systems," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, Art. no. 10514.

[6] E. Testa et al., "Logic synthesis for established and emerging computing," *Proc. IEEE*, vol. 107, no. 1, pp. 165–184, Jan. 2019.

[7] L. Amaru et al., "New logic synthesis as nanotechnology enabler," *Proc. IEEE*, vol. 103, no. 11, pp. 2168–2195, Aug. 2015.

[8] Q. P. Herr et al., "Ultra-low power superconducting logic," *J. Appl. Phys.*, vol. 109, May 2011, Art. no. 103903.

[9] B. Patra et al., "Cryo-CMOS circuits and systems for quantum computing applications," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 309–321, Jan. 2018.

[10] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, Mar. 1991.

[11] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 760–769, Jun. 2011.

[12] M. Tanaka et al., "Low-energy consumption RSFQ circuits driven by low voltages," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, Jun. 2013, Art. no. 1701104.

[13] G. Krylov and E. G. Friedman, "Asynchronous dynamic single-flux quantum majority gates," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 5, pp. 1–7, Aug. 2020.

[14] C. J. Fourie et al., "ColdFlux superconducting EDA and TCAD tools project: Overview and progress," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–7, Aug. 2019.

[15] N. Katam and M. Pedram, "Logic optimization, complex cell design, and retiming of single flux quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 7, Oct. 2018, Art. no. 1301409.

[16] G. Pasandi and M. Pedram, "A dynamic programming-based, path balancing technology mapping algorithm targeting area minimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[17] T. Jabbari et al., "Splitter trees in single flux quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, pp. 1–6, Aug. 2021.

[18] L. Amaru et al., "First demonstration of a superconducting electronics microcontroller RTL-to-GDSII flow," in *Proc. GOMAC-Tech-Government Microcircuit Appl. Crit. Technol. Conf.*, 2021, pp. 1–4.

[19] E. Goto, "The parametron, a digital computing element which utilizes parametric oscillation," *Proc. IRE*, vol. 47, no. 8, pp. 1304–1316, Aug. 1959.

[20] N. Takeuchi et al., "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Superconductor Sci. Technol.*, vol. 26, no. 3, Mar. 2013, Art. no. 035010.

[21] C. L. Ayala et al., "Majority-logic-optimized parallel prefix carry look-ahead adder families using adiabatic quantum-flux-parametron logic," *IEEE Trans. Appl. Supercond.,* vol. 27, no. 4, pp. 1–7, Jun. 2017.

[22] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Adiabatic quantum-flux-parametron cell library adopting minimalist design," *J. Appl. Phys.*, vol. 117, no. 17, May 2015, Art. no. 173912.

[23] C. L. Ayala et al., "A semi-custom design methodology and environment for implementing superconductor adiabatic quantum-flux-parametron microprocessors," *Superconductor Sci. Technol.*, vol. 33, no. 5, May 2020, Art. no. 054006.

[24] R. Cai et al., "A majority logic synthesis framework for adiabatic quantum-flux-parametron superconducting circuits," in *Proc. Great Lakes Symp. VLSI*, May 2019, pp. 189–194.

[25] R. Cai et al., "A buffer and splitter insertion framework for adiabatic quantum-flux-parametron superconducting circuits," in *Proc. IEEE 37th Int. Conf. Comput. Design (ICCD),* Nov. 2019, pp. 429–436.

[26] E. Testa et al., "Algebraic and Boolean optimization methods for AQFP superconducting circuits," in *Proc. ASPDAC*, Tokyo, Japan, Jan. 2021, pp. 779–785.

[27] D. E. Knuth, *The Art of Computer Programming*, vol. 4. Reading, MA, USA: Addison-Wesley, 2011.

[28] D. S. Marakkalage et al., "Three-input gates for logic synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Oct. 20, 2020, doi: 10.1109/TCAD.2020.3032625.

[29] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization," in *Proc. 51st Annu. Design Autom. Conf. Design Autom. Conf. (DAC),* San Francisco, CA, USA, Jun. 2014, pp. 1–6.

[30] L. Amaru, P.-E. Gaillardon, and G. De Micheli, "Majority-inverter graph: A new paradigm for logic optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 5, pp. 806–819, May 2016.

[31] G. Krylov and E. G. Friedman, "Globally asynchronous, locally synchronous clocking and shared interconnect for large-scale SFQ systems," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, pp. 1–5, Aug. 2019.

[32] W. P. Burleson et al., "Wave-pipelining: A tutorial and research survey," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 6, no. 3, pp. 464–474, Sep. 1998.

[33] D. Wong, G. De Micheli, and M. Flynn, "Algorithms for designing high-performance digital circuits using wave pipelining," *IEEE Trans. CAD/ICAS*, vol. 12, no. 5, pp. 25–46, Jan. 1993.

[34] T. V. Filippov et al., "20 GHz operation of an asynchronous wave-pipelined RSFQ arithmetic-logic unit," *Phys. Procedia*, vol. 36, pp. 59–65, 2012.

[35] Z. Deng, N. Yoshikawa, J. Tierno, S. Whiteley, and T. van Duzer, "Asynchronous circuits and systems in superconducting RSFQ digital technology," in *Proc. Symp. Asynchronous Circuits Syst.*, Apr. 1998.

[36] F. Ke et al., "Demonstration of a 47.8 GHz high-speed FFT processor using single-flux-quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, pp. 1–5, Aug. 2021.

[37] C. L. Ayala et al., "MANA: A Monolithic Adiabatic iNtegration Architecture microprocessor using 1.4zJ/op superconductor Josephson junction devices," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2020, pp. 1–2.

Giovanni De Micheli is a Professor at EPFL, Lausanne, Switzerland. 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 heterogeneous integration of circuits and sensors. He is a Fellow of ACM and IEEE, a member of the Academia Europaea, and an International Honorary member of the American Academy of Arts and Sciences.

■ Direct questions and comments about this article to Giovanni de Micheli, EPFL, 1015 Lausanne, Switzerland; giovanni.demicheli@EPFL.ch.