

FELECTRONIC DESIGN AUTOMATION (EDA) is essential for the design of large-scale microelectronic systems. In this article, EDA methodologies, techniques, and algorithms used to develop superconductive computing systems are reviewed. The semicustom standard cell-based design flow, common in conventional CMOS circuits, is widely adopted in modern superconductive digital circuits. Differences and issues in CAD flows as compared to CMOS design methodologies are highlighted. The most common stages of these design flows, from high-level simulation to physical layout, are described. These stages are grouped into three areas: simulation/modeling, synthesis/place and route, and verification. Modern approaches and tools for superconductive circuits are reviewed for each of these areas.

Design Automation of Superconductive Digital Circuits

A review.



GLEB KRYLOV, JAMIL KAWA, AND EBY G. FRIEDMAN

Digital Object Identifier 10.1109/MNANO.2021.3113218
Date of current version: 13 October 2021

INTRODUCTION

As the scaling of CMOS systems becomes progressively more expensive, multiple technologies have been proposed to improve performance and power efficiency. Cryogenic electronics specifically targets large-scale stationary systems, greatly offsetting the energy required by cryogenic refrigeration with a higher energy efficiency [1]. Modern superconductive electronics (SCE) based on Josephson junctions (JJs), such as rapid single-flux quantum (RSFQ) [2], adiabatic quantum-flux parametron (AQFP) [3], and reciprocal quantum logic (RQL) [4], promise at least two orders of magnitude improvement in energy efficiency as compared to conventional semiconductor-based supercomputers [1].

While state-of-the-art niobium-based fabrication technologies can produce more than 1 million JJs per die [5], practical superconductive circuits are significantly less complex. The physical properties of superconductive materials and devices are now well understood; considerable engineering effort is, however, required to ensure large-scale SCE systems are practical and cost-effective. To support the increasing complexity of these systems, advances in EDA algorithms, tools, and methodologies are necessary [6]. These efforts are the primary focus of this article. SCE has recently reached the complexity at which a semicustom standard cell design flow [7] is preferable to a custom design process in terms of computational effort, design time, and cost. A semicustom design flow enables the automation of many stages of the design process.

Common elements of a typical EDA flow are shown in Figure 1. Specific processes, methodologies, algorithms, and tools exist for the different layers of abstraction: register transfer level (RTL), logic, circuit, layout, and device. Within each layer, the design is automatically synthesized with specialized algorithms or manually evaluated based on simulation and modeling tools. The design is verified to ensure logical and functional correctness and mitigation of failure mechanisms, such as flux trapping [8], based on specialized verification tools. These verification tools also employ

SCE has recently reached the complexity at which a semicustom standard cell design flow is preferable to a custom design process in terms of computational effort, design time, and cost.

simulation and modeling capabilities to extract relevant information from the circuit. These stages are common in most integrated circuit (IC) design processes; each individual flow is a complex process composed of several different stages [9]. In this article, the individual stages within modern semicustom EDA flows are reviewed within the context of superconductive digital electronics. In a standard cell design methodology, well-characterized blocks perform specific logic functions. The standard cell design process and related cell libraries for superconductive ICs are described in the section “Cell Library Design and Characterization.”

SIMULATION AND MODELING

Simulation and modeling tools are essential for both the manual and automated design of microelectronic circuits. These tools were among the first EDA tools to be developed. The design of a complex functional block begins with a description of a circuit at a high level of abstraction, such as at the RTL. Tools and

techniques for RTL simulation of superconductive circuits are described in the section “RTL Design and Simulation.” To design and characterize each cell within a standard cell library, a dynamic circuit simulator is used. A physical layout of a circuit contains many parasitic components. These elements are not considered during early phases of the design process and can degrade circuit operation. Because of the minimal use of resistive elements in SCE, the inductance, in particular, can greatly affect the operation and behavior of superconductive circuits. These topics are discussed in later sections.

SYNTHESIS AND PLACE AND ROUTE

Synthesis tools enable fast and efficient design of large-scale circuits, which would be prohibitively complex during a manual design process. In logic synthesis, an RTL description of a circuit is converted into equivalent logic elements from an existing cell library. This process enables fast conversion and optimization from a high-level description into individual circuit components to produce a large-scale

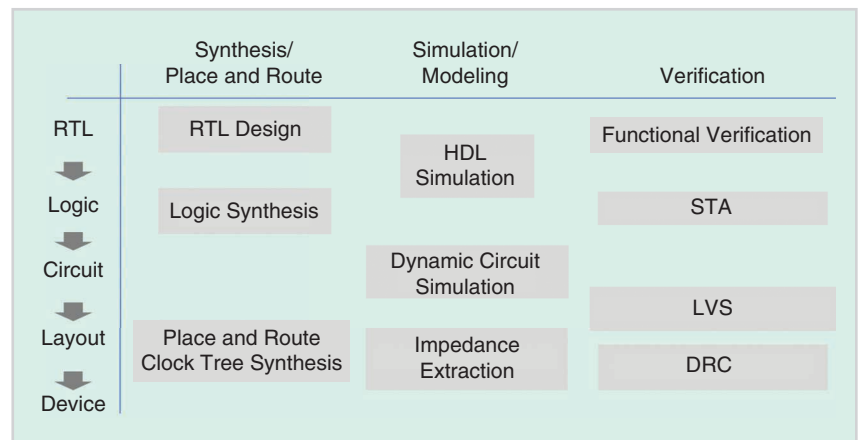


FIGURE 1 EDA tools for ICs. IC: integrated circuit; HDL: hardware description language; STA: static timing analysis; LVS: layout versus schematic; DRC: design rule check.

SystemVerilog models provide multiple benefits, among which are compatibility with industrial tools, modular reusable design, and integration with existing verification methodologies.

system. Automated logic synthesis for superconductive circuits is discussed in the section “Logic Synthesis.” A synthesized circuit consists of standard cells and the connections between those cells. To lay out the physical geometry of the cells and route the connections between the cells while satisfying specific design rules, automated place-and-route (APAR) tools are utilized. APAR techniques are reviewed in the section “Layout Synthesis.”

VERIFICATION

An IC is a complex system that inevitably contains a variety of logic and layout errors. As the fabrication of an IC is often costly in terms of both resources and time, a significant effort is directed at identifying, fixing, and/or tolerating errors during the different stages of the design process. A multigigahertz IC in particular can exhibit narrow timing margins. To ensure correct timing for the logic gates and flip-flops and prevent race conditions, timing analysis tools are necessary. Timing analysis tools for superconductive circuits are described in the section “Timing Analysis.”

Functional verification detects errors in the RTL description of a circuit. This step utilizes hardware description language (HDL) simulators and specialized verification methodologies. To detect the errors introduced during the physical layout stage of the design process, design rule check (DRC) and layout versus schematic (LVS) tools are used. Existing verification techniques and tools for these steps are reviewed in the section “Verification and Testability.”

CELL LIBRARY DESIGN AND CHARACTERIZATION

Each cell within a standard cell library is developed and characterized only once,

followed by reuse of these cells throughout multiple circuit design processes. In this section, existing standard cell libraries and characterization methodologies for superconductive circuits are described. Many industrial and academic superconductive standard cell libraries have been developed for internal use. Some of them are briefly reviewed here.

Early cell libraries were primarily developed for manual or semimanual design of RSFQ circuits. The cell library developed at Stony Brook University is one of the earliest cell libraries developed for RSFQ circuits [10]. The circuits are optimized to provide ease of interconnectivity with other cells in the library. This capability minimizes the redistribution of bias currents among the connected cells, which can affect the circuit behavior and timing characteristics. Physical layouts are available for most cells, although the layouts are based on a mature 3.5- μm Hypres fabrication process [10]. Most of the cells have been experimentally verified. The Stony Brook cell library remains one of the few open source RSFQ cell libraries, and is a foundation for a significant portion of the circuit development process in modern RSFQ circuits. Another openly available RSFQ cell library is sourced by the Ilmenau University of Technology [11] and targets the Fluxonics foundry [12].

The CONNECT cell library was developed in collaboration between Nagoya University and National Electrical Code (NEC) [13], in which cells are also optimized for interconnectivity by minimizing the bias current redistribution. The cells are characterized, and timing information is available. The physical layouts are based on an NEC niobium fabrication process [14]. An RSFQ cell library has been developed

at the National Institute of Advanced Industrial Science and Technology (AIST), also in Japan [15], using a relatively advanced 10 kA/cm² AIST fabrication process. These cells [16] are scaled and optimized to utilize less area while preserving functionality.

The architecture of modern SCE cell libraries, as compared to earlier cell libraries, primarily targets automated cell placement and interconnect routing tools. This focus on design automation places several major requirements on the cell structure. The cell library must satisfy specific routing rules and allocate sufficient space for routing. The characteristics of the cells must be extracted to support automated timing analysis.

A common approach to the automated placement and routing of medium-scale integration and large-scale integration (LSI) CMOS circuits is row-based placement [7]. These techniques, commonly utilized in LSI CMOS circuits, are also applicable to modern superconductive circuits. Modern RSFQ and AQFP circuits adopt a row-based standard cell placement methodology with channel routing. The standard cells are arranged in rows, while the space between these rows is reserved for the signal routing channels [7]. While each cell occupies a different area depending upon the function, a uniform cell height is maintained to support standard height rows.

A major difference between CMOS and RSFQ circuits is the two different types of interconnects—passive transmission lines (PTLs) and active Josephson transmission lines (JTLs)—used in RSFQ circuits. The interconnect in CMOS technology is composed of metal lines directly connecting logic gates [17]–[19]. RSFQ circuits can be abutted, where the output of a gate directly connects to the input of another gate. JTLs in RSFQ circuits are active elements and are abutted to the cells driving and receiving SFQ signals. JTLs can therefore be characterized in a similar manner to any other logic cell within a cell library. PTLs are composed of strip-lines with a specific characteristic impedance, and require a driver and receiver to operate [20]. The driver and receiver are characterized as active logic gates. These

gates are connected by a (typically long) passive metal stripline or microstripline. The interconnect delay for a PTL is related to the speed of light (in the medium) in these long metal lines [2]. A typical PTL delay is on the order of 1 ps per 100 μm .

An RSFQ cell library for a modern Massachusetts Institute of Technology Lincoln Laboratory SFQ5ee fabrication process [21] has recently been developed by Hypres [22]. This library supports both RSFQ and energy-efficient SFQ (ERSFQ) [23], [24] bias distribution by separating each cell into multiple components. The core portion of the cells contains most of the circuitry. The bias component contains the relevant elements: a bias resistor in the case of conventional RSFQ bias networks and a bias JJ with a large inductor in the case of ERSFQ logic [25], [26]. Another portion of the cell contains tracks for routing PTLs and bias lines for use by an automated router.

A methodology for extracting the timing parameters of standard library cells has been proposed [27] in which the cell delays for different combinations of preceding and subsequent cells are extracted in a standard format. These delays can be included within modern industrial CMOS static timing analysis (STA) tools, as further discussed in the section “Timing Analysis.” A methodology is proposed in [25] for parameterizing these standard cells. This methodology allows the cells to be rotated and flipped—common operations during the design process—while maintaining the bias and PTL tracks.

A cell library for the automated layout of AQFP circuits has also been developed [28] based on standard cells and is integrated into a semicustom design flow. Certain steps within the design flow, such as circuit simulation and layout placement, are performed by industrial tools. Other steps, such as retiming, utilize custom tools specifically developed for AQFP. A bottom-up approach to building an AQFP cell library has been proposed [29] in which only four basic elements are used: branch, buffer, constant, and

The phase is a natural parameter characterizing the state of a JJ, allowing the state, signals, and behavior of a circuit to also be characterized by the phase.

NOT. All AQFP cells are composed from these elements, greatly simplifying the cell library design process.

A tool for the automated extraction of the timing parameters of AQFP cells has also been developed [30]. This tool determines setup/hold time characteristics and delay information by utilizing a dynamic circuit simulation capability. The timing information is included within an industry standard format for use by timing analysis tools.

RTL DESIGN AND SIMULATION

Electrical circuit simulation exhibits high accuracy but low computational efficiency. When the size of a circuit exceeds several thousand active elements, the circuit simulation process becomes prohibitively long. A solution to this complexity issue is to simulate the circuit at higher levels of abstraction with simplified models. The RTL abstraction layer models the transfer of data between registers with functional operations on the data. This abstraction level is often described in an HDL. RTL simulation is generally technology independent; the logic gates and flip-flops described by an HDL can be based on different technologies. HDL simulation of RSFQ circuits, however, generally differs from HDL simulation of CMOS circuits since the signals are pulse based. In RQL and AQFP circuits with ac clock signals, a different representation of the signals is used [3], [31]. In addition, the extraction of timing parameters suitable for HDL models is an important issue. In this section, existing research on HDL models of superconductive circuits is reviewed.

The logic simulation of RSFQ circuits was first demonstrated in 1993 [33]. Although this approach enabled

behavioral and timing simulation of RSFQ circuits, it utilized a CMOS-oriented model structure specific to a proprietary simulation tool. The personal superconductor circuit analyzer (PSCAN) circuit simulator [34] utilizes an internal HDL, SFQHDL, to describe the intended circuit behavior. This feature enables automated margin analysis and parameter optimization of RSFQ gates.

RTL-level models of RSFQ circuits based on a general-purpose HDL were proposed in 1997 [32], [35], [36]. Both Verilog HDL [32] and the very-high-speed integrated circuit hardware design language (VHDL) [35], [36] can be used. In this approach, gate-level HDL models of RSFQ circuits are developed for each logic gate and flip-flop. SFQ data and clock pulses are modeled as simple rectangular pulses [37]. An example of an HDL simulation of a half adder is shown in Figure 2. The HDL simulation operates with logic states rather than voltages and currents. The internal structure of the gates is not modeled, which reduces the computational complexity. Only the state of the gates is stored. The behavior of the cell is modeled as a finite-state automaton (FSA), where an input pulse causes a transition between states and/or generates an output pulse [38].

RSFQ gates exhibit specific timing requirements. Transitions between the states of an FSA occur within a specific time window. To produce accurate simulation results, the timing characteristics are included within the HDL models. These characteristics include the output delay, setup time, and hold time. The timing characteristics are extracted from circuit simulations. Many additional factors exist that can change these characteristics, such as parasitic impedances,

temperature, bias current fluctuations, and fabrication process variations. Moreover, these characteristics are frequently data dependent.

Multiple ways to integrate these factors into HDL models have been proposed [38]–[40]. The timing characteristics of individual cells are initially determined. The parasitic elements are extracted and back-annotated into the circuit design flow [38]. The next step is to extract the delay of combinations of gates and more complex circuits [39]. The probabilistic nature of the delay of RSFQ cells is typically modeled as a normal Gaussian distribution [40]. Monte Carlo simulations are performed to determine the delay of the more complex blocks.

Tools and methodologies for the automated extraction of HDL models and timing characteristics from circuit netlists have recently been proposed [41],

[42]. In this process, SFQ pulses are applied at the circuit inputs, allowing different states within the flux storage loops to be identified. This capability enables the automated extraction of an FSA representation of a specific circuit. Critical timing characteristics are also extracted by applying input pulses in a binary search pattern to verify the output states [42].

A different method for representing signals in an HDL is necessary for RQL and AQFP logic. In RQL, information is represented as the presence or absence of two reciprocal SFQ pulses, while the clock is a multiphase ac sinusoidal signal [31]. The clock signal in a VHDL model of RQL is composed of three regions based on the magnitude and direction: positive pulse propagation, negative pulse propagation, and no propagation [31]. The total change

in phase of the JJs within the RQL gates is zero. The change in phase is nonzero only during the time between the arrival of the positive and negative SFQ pulses comprising the signal. This property enables a natural translation of RQL signals into an HDL. Logic “zero” corresponds to the absence of activity. In the case of a logic “one,” positive and negative SFQ pulses correspond, respectively, to the positive and negative edges of a conventional CMOS-like signal [31].

A similar approach to RQL is used in HDL models of AQFP logic. The direction of the excitation current determines the HDL state. A negative current corresponds to logic “zero,” a positive current to logic “one,” and the absence of a current corresponds to the high-impedance Z state [43].

The SystemVerilog language has recently been proposed to model the characteristics of RSFQ and AQFP circuits in HDL [43], [44]. SystemVerilog models provide multiple benefits, among which are compatibility with industrial tools, modular reusable design, and integration with existing verification methodologies. Moreover, this approach supports the HDL simulation of hybrid RSFQ/AQFP systems.

CIRCUIT SIMULATION

Circuit simulators capable of analyzing circuits with JJs are reviewed in this section. Dynamic circuit simulators for SCE operate with voltages, currents, and phases at different nodes and can produce highly accurate timing and waveform properties, while exhibiting a high computational complexity. An example waveform of a circuit simulation of a JTL is shown in Figure 3, where the input and output voltages of a two-stage JTL, connected to a clock signal, are shown.

Two different methods are used to simulate superconductive circuits. In one approach, the node voltage is the fundamental variable of a dynamic simulator. This approach is commonly adopted in dynamic simulators for conventional electronic circuits based on the original Spice simulator [45]. Spice-based tools for SCE include JSpice3 [46], JSIM [47], and WRSpice [48]. These simulators are

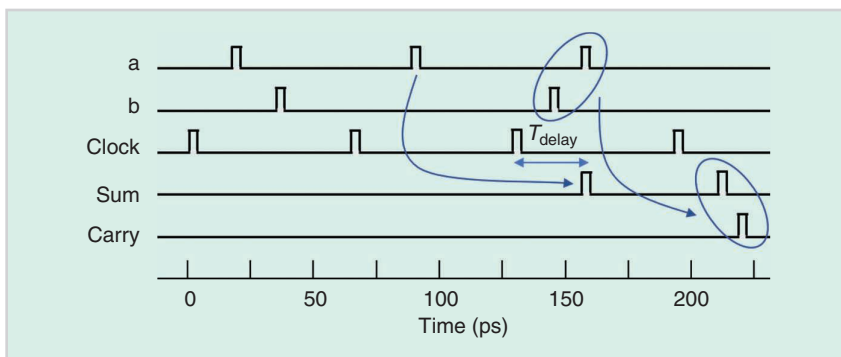


FIGURE 2 HDL simulation of an RSFQ half adder [32].

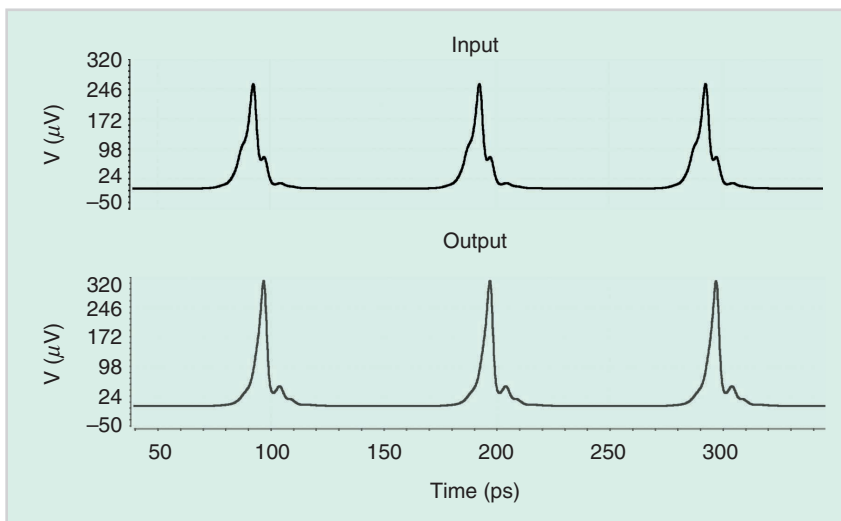


FIGURE 3 Circuit simulation of a two-stage JTL. The voltages across the input JJ and output JJ are shown.

in common use and include a variety of methods to speed up the time required to simulate JJ-based circuits [48].

Certain Spice-like simulators support JJ-based circuits by using a physics-based device model. Verilog-A is an HDL (see the section “RTL Design and Simulation”) commonly used to describe an electronic device based on the physical equations characterizing the behavior of a junction. A Verilog-A model of a JJ is typically based on the resistively and capacitively shunted junction (RCSJ) model [49]. The Verilog-A model enables simulation of JJs with the same industrial simulators used for CMOS circuits, such as Synopsys HSPICE [50] and Spectre [51]. However, the use of an external model within a standard circuit simulator can be computationally slower than using a model embedded within the simulator because of the absence of enhancements specific to JJ-based circuits [52].

Among the advantages of this type of simulator is compatibility with existing circuit simulation tools. Another important benefit is multimode simulation, combining both superconductive and semiconductor devices (mixed technologies). This capability can be beneficial, for example, in RSFQ/CMOS interface circuits. Among the disadvantages of existing simulation tools is the difficulty in performing dc analyses as the dc operating point depends upon the phase of the JJs. Modern versions of WRspice [48] and JoSIM [53] support dc analysis based on the phase.

A different approach to simulate superconductive circuits is to treat the phase of the node rather than the voltage as the fundamental variable. The phase is a natural parameter characterizing the state of a JJ, allowing the state, signals, and behavior of a circuit to also be characterized by the phase. The most widely used simulators utilizing this approach are PSCAN [34], [54] and PSCAN2 [55]. Another recently developed simulator, JoSIM [53], operates with both voltages and phases. No studies, however, exist comparing these approaches in terms of relative computational speed and/or accuracy [52].

An important feature of a superconductive circuit simulator is whether the

A distinctive advantage of complex microwave simulators, such as HFSS and Sonnet, is the capability to extract frequency-dependent impedances.

microscopic tunnel junction model of a JJ [56], [57] is supported. Although the RCSJ model is sufficiently accurate for most modern digital superconductive circuits, the Werthamer model [56] targets digital circuits with externally shunted JJs, where the shunt resistance and inductance are small. In addition, voltage variations across a JJ, on a picosecond time scale, are typically assumed to be small in comparison to the gap voltage [58]. Significant differences exist between the microscopic Werthamer model and the RCSJ model for junctions with high damping [49]. The microscopic Werthamer model enables simulation of deeply scaled, unshunted JJs with a high critical current density [57]. PSCAN2 [55], JoSIM [53], WRspice [50], and Synopsys HSpice [59] currently support embedded microscopic JJ models.

INDUCTANCE EXTRACTION

On-chip inductance is a critical design parameter in superconductive circuits. Small variations in the inductance can produce incorrect circuit operation. For simple structures, the inductance can be estimated based on fabrication process characteristics. However, most practical circuits include inductors of complex geometry. To design circuits with these inductors, accurate extraction tools are necessary. In this section, these tools and relevant techniques are described.

Inductance extraction can be broadly separated into two categories: 2D and 3D [52]. 2D extraction tools are typically significantly faster and less accurate than 3D tools. As described in an important review on inductance extraction [52], 2D methods are not commonly used in the design of modern superconductive circuits.

An intermediate step between 2D and 3D simulation is 2.5D [60] or planar 3D [52] simulation. This type of 2.5D analysis is used in the Sonnet field solver [61], [62]. In this 2.5D simulation system, a 3D circuit geometry is separated into conducting surfaces, which are partitioned into 2D segments. Field equations are solved for these segments based on a surface impedance model using the method of moments [63]. This type of analysis is faster than 3D analysis [52], while providing good accuracy for most structures. This simulator, however, exhibits significantly lower accuracy for the narrow submicrometer lines expected in next-generation superconductive circuits [60].

The Ansys high-frequency structure simulator (HFSS) tool is also used to extract impedances within superconductive circuits [64]. This tool combines three-dimensionality with the finite-element method [65]. A distinctive advantage of complex microwave simulators, such as HFSS and Sonnet, is the capability to extract frequency-dependent impedances.

A commonly used 3D inductance extraction tool in SCE is FastHenry [66]. This field solver was originally developed and widely used for conventional CMOS circuits [67]. In FastHenry, a conductor is divided into segments and subdivided into filaments—the partial element equivalent circuit method. Filaments and terminal sources form an equivalent circuit from which a complex impedance matrix is extracted. This tool, with additional modifications, is capable of simulating superconductive structures. In [68], an additional term corresponding to the kinetic inductance is added to the basic governing equations. In [69], the London equations and the two-fluid

These tables allow quick estimation and detection of timing violations without requiring a circuit to be simulated on a nodal basis.

model [70] are used to support super-conductive structures. FastHenry produces highly accurate results for simple structures [71]; the tool, however, is computationally expensive. Extracting the inductance of complex geometries requires a fine mesh composed of many filaments and can require a prohibitively long computational time.

One of the most popular inductance extraction tools suitable for large and complex structures is InductEx [72]. Initial versions of this tool are based on a FastHenry engine [73]. InductEx utilizes a novel segmentation algorithm, in which the edges of the geometry are divided into finer segments than the simpler regions. Multiple modifications have also been introduced to improve the speed of the analysis process. For example, a novel field solver has been developed utilizing cuboid segments [74]. The latest versions utilize a tetrahedral mesh and other enhancements to improve both accuracy and speed [75].

A different methodology is used in the 3-D-MLSI tool [76], [77], in which individual currents are produced from stream functions. The London and

Maxwell equations are described in terms of these stream functions, and the resulting expression is solved using the finite-element method [78]. This approach is computationally efficient and enables inductance extraction of complex structures [79].

LOGIC SYNTHESIS

The behavioral description of a circuit is converted during logic synthesis into a gate-level netlist. Multiple differences exist in RSFQ circuits as compared to conventional CMOS circuits, requiring modifications to existing tools [80], [81]. In this section, existing approaches to logic synthesis for SCE are discussed.

LOGIC REPRESENTATION

One of the first methodologies for automated synthesis of RSFQ circuits is the top-down binary decision diagram (BDD) methodology [82]. A BDD is an acyclic directed graph consisting of decision nodes and terminal nodes [83]. A Boolean function can be represented as a binary decision tree, as shown in Figure 4(a), where each variable corresponds to a decision node, while the

terminal nodes correspond to the state of the Boolean function. A different path within a graph exists for each combination of variables, X_1 , X_2 , and X_3 . A BDD, shown in Figure 4(b), is a reduced representation of a binary decision tree where the redundant nodes and edges are omitted.

In a BDD-based synthesis methodology, the cell library contains only one logic gate: a binary switch [82]. In RSFQ technology, this switch is based on a D flip-flop (DFF), which does not require significant area. Other elements in the cell library provide connections between the switches. BDD transformations, however, exhibit high computational complexity and are therefore not feasible for large-scale circuits.

A commonly used, academically developed synthesis tool, ABC [84], is applicable to RSFQ circuits. ABC converts the behavioral description of a circuit into an intermediate representation: the AND-inverter graph (AIG). An AIG is an acyclic directed graph consisting of conjunction nodes, terminal nodes, and edges that can contain the inversion operation. An example AIG is shown in Figure 5, where each node represents the conjunction operation on a corresponding child node, and the inversion operations are represented by black dots on the edges. Specific logic transformations are performed on the AIG representation [85] to enhance the synthesized circuit. The ABC tool is widely used as a platform for synthesis optimization.

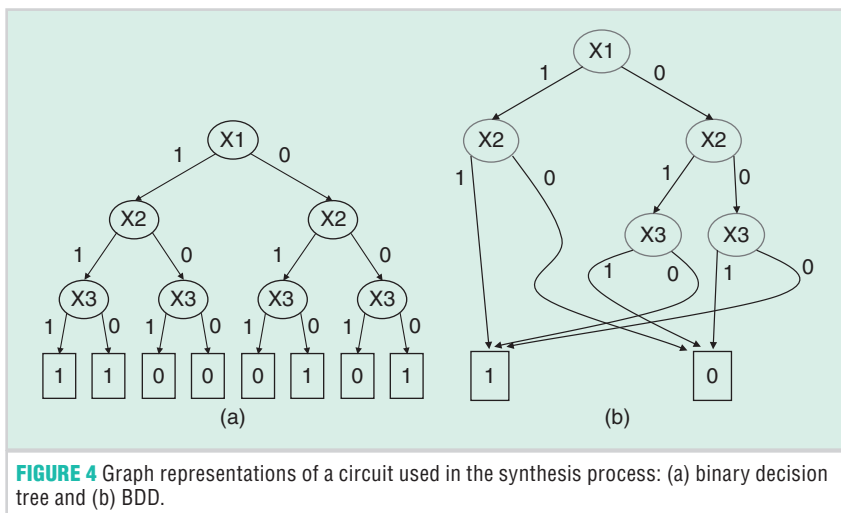


FIGURE 4 Graph representations of a circuit used in the synthesis process: (a) binary decision tree and (b) BDD.

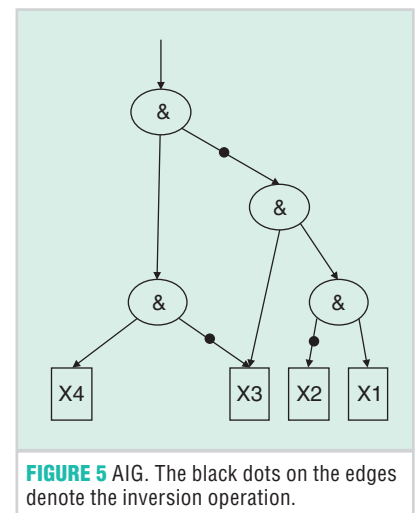


FIGURE 5 AIG. The black dots on the edges denote the inversion operation.

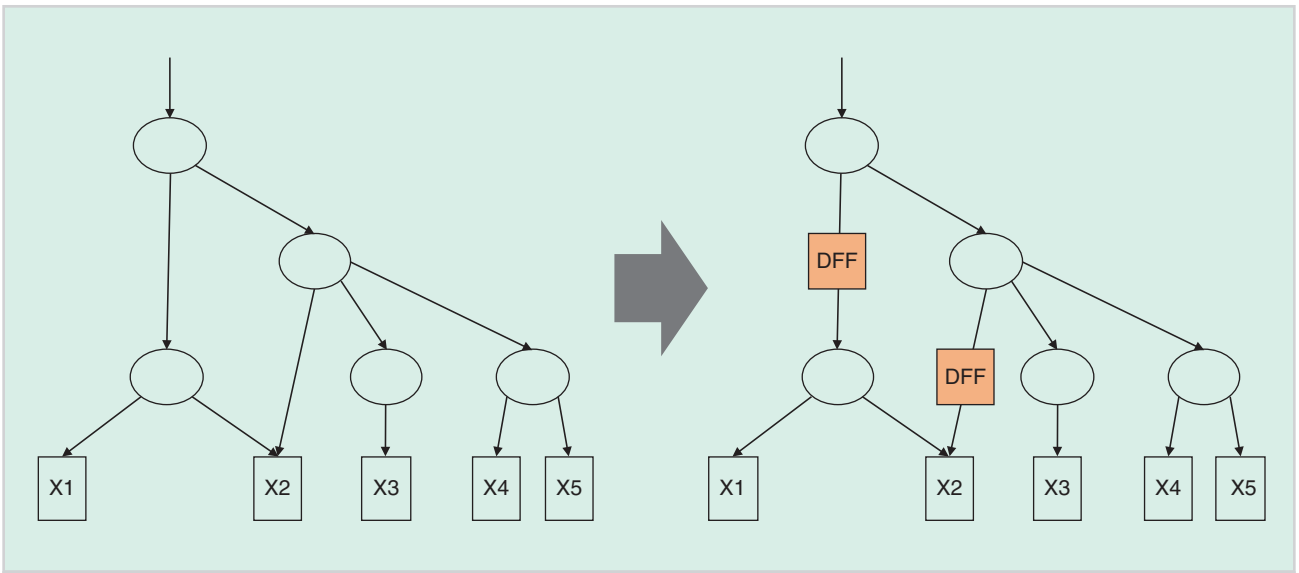


FIGURE 6 Insertion of path-balancing DFFs into RSFQ logic paths.

A modification of the AIG representation of a Boolean function is the majority-inverter graph (MIG) [86]. In a MIG, three input majority nodes are used rather than conjunction nodes. This format enables a more natural representation of logic circuits with efficient majority elements, such as AQFP [87] and dynamic SFQ [88] circuits. The MIG representation also supports specific optimizations that exploit the properties of the majority function [89].

AQFP logic is also compatible with standard CMOS synthesis tools, such as Yosys [90]. AQFP logic exhibits an efficient inversion operation and limited fan-out. Additional transformations of the netlist are therefore necessary [91]. A synthesis methodology for AQFP circuits has also been proposed [92]. In this methodology, an And-Or-Inverter (AOI) graph is converted into a majority-minority graph, similar to the MIG. As AQFP technology features efficient majority gates, this approach reduces the number of JJs as compared to an AOI representation and also lowers the delay due to a shallower logic depth.

PATH BALANCING

Multiple changes are necessary for ABC to support the synthesis of RSFQ circuits, first proposed in [93]. As previously described, most logic gates in RSFQ technology are individually clocked [94]. If different inputs of a gate are at differ-

ent logic depths, an erroneous output is produced. To balance the depth of all of the gate inputs, path-balancing DFFs are added. As shown in Figure 6, DFFs are inserted into those input paths that exhibit a shallow logic depth. These flip-flops operate as dummy gates, performing no functional operation other than delaying a data pulse by a single clock period. This process is referred to as path balancing.

In a complex RSFQ logic path, many path-balancing DFFs are needed, requiring significant additional area. Multiple techniques and algorithms have been developed to reduce the number of these DFFs [95]. One such technique is retiming [96], [97], in which logic gates

and path-balancing DFFs are rearranged to reduce the number of flip-flops. The effect of retiming on the total number of path-balancing DFFs is shown in Figure 7. A technique to convert a CMOS netlist into an RSFQ netlist has been demonstrated [98]. This technique also utilizes retiming to reduce the number of flip-flops.

LAYOUT SYNTHESIS

The synthesis process produces a symbolic description of a synthesized circuit [99]. At this stage, the circuit is composed of standard cells and connections between these cells. The physical topology of the individual cells is laid out; the geometric descriptions of the elements

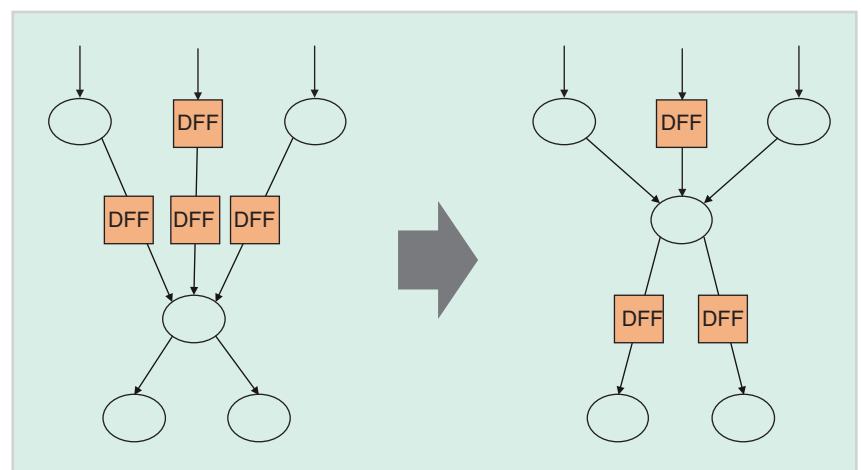


FIGURE 7 Retiming of a logic path with path-balancing DFFs.

The effectiveness of these tools to enable large-scale superconductive digital systems is greatly improving.

are assigned to specific on-chip locations by a placement tool. The necessary connections between cells are determined by a routing tool using interconnect elements. In this section, automated placement (placer) and routing (router) tools and related algorithms for superconductive circuits are described.

The initial placement of the cells representing the synthesized circuit utilizes a geometric representation of the physical cells that contains the information associated with the cell height and width, and the exact location of all of the inputs, outputs, and bias pins. These abstractions of the physical cells also contain necessary information about regions the router can use to route and regions that are prohibited to the router (blockages) to ensure the connectivity constructed by the router is legitimate and does not create shorts between signals. After the initial seed placement, the placement is finely optimized to minimize the length of the wiring between the cells. This process is based on routines that assign different weights to different criteria.

APAR FOR RSFQ

One of the first automated APAR methodologies for RSFQ systems was developed [100] to lay out an 8-bit general-purpose RISC processor composed of 20,000 cells. This methodology utilizes PTL interconnect for routing, and an H-tree topology [138] for the clock distribution network. The placer is based on the Fiduccia-Mattheyses heuristic [101] to recursively partition a circuit while minimizing the number of connections between partitions. By placing the connected cells near each other, the wire length and, therefore, the delay are reduced, thereby increasing the clock frequency. Routing in this methodology

can be performed by any standard commercial CMOS routing tool [102].

A routing tool specifically targeting RSFQ circuits is proposed in [103] and used to route an 8-bit microprocessor, where a PTL interconnect is exclusively utilized for routing. This tool is based on the A* algorithm [104], widely used for CMOS routing and, in general, for minimizing the cost of a path within a graph. The tool also adjusts the cost of these paths to decrease the number of vias and corners along a path.

The delay of the JTL interconnect strongly depends upon variations in the manufacturing process characteristics. Variations in the bias current, JJ size, or inductance can change the delay by several picoseconds. This effect can be unacceptable in a multigigahertz system. The PTL interconnect also exhibits small variations in the delay, which primarily depend upon the length of the line [20], [105]. It can therefore be desirable to insert a delay by increasing the length of the PTL rather than adding a JTL delay element. In [106], this property is exploited in a routing tool based on integer linear programming (ILP) [107]. This technique is similar to wire snaking, commonly used in CMOS circuits for delay balancing [108].

An extension of this work is described in [109], where simulated annealing (SA) [110] rather than ILP is used to decrease the routing time. Segments of the PTL interconnect—the delay matching elements—are inserted to balance the delay of the different paths. The SA algorithm is also used for automated placement, where the length of the wires and the delay matching elements are minimized. In [111], this approach is improved by rearranging the delay matching elements. This technique reduces the minimum width of the routing channels.

An APAR methodology and a tool are proposed in [112] based on an HL-tree clock distribution network [113]. In this approach, the cells are grouped by increasing logic level. Cells within each group are abutted, decreasing both the area and the total wire length. In this methodology, the open source router tool [114], based on the maze routing algorithm [115], is used for routing, while the SimPL algorithm [116] is used for global placement.

The delay and area of a PTL interconnect are greater than those of a JTL interconnect for short-distance routing [20]. This property is due to the relatively large delay of the driver and receiver required to interface with the passive stripline. A mixed approach for routing RSFQ circuits has also been proposed [117], utilizing both JTLs and PTLs. A place-and-route methodology for data-driven, self-timed RSFQ circuits has also been proposed [118]. This methodology utilizes commercial CMOS-based APAR tools to synthesize and lay out asynchronous RSFQ circuits.

APAR FOR AQFP

AQFP circuits are typically placed in a row-based standard cell topology. Each row corresponds to a different logic level, and is synchronized by a different clock phase.

AQFP circuits utilize a different type of interconnect than RSFQ logic. The current waveforms used for signaling in this technology propagate within standard metal wires, similar to those in CMOS circuits. Multiple restrictions exist when routing these wires, as long interconnect segments introduce attenuation. In [119] and [120], an APAR methodology for AQFP circuits is proposed. In this approach, a genetic algorithm [121] is used to place cells while reducing the number of long interconnect segments. Buffers are inserted into the remaining long lines. The left-edge algorithm [122] is used for channel routing. This methodology has been successfully used to automatically place and route a 16-bit AQFP adder [119].

TIMING ANALYSIS

SCE systems operate at extremely high clock frequencies and exhibit small gate

delays [123]. The timing analysis process in these circuits is essential for high-speed operation. In this section, timing analysis and related techniques for SCE are reviewed.

Conventional CMOS circuits utilize different techniques for timing analysis. Dynamic timing analysis (DTA), which operates at an abstraction layer above dynamic circuit simulation, simulates a system at the behavioral level. Although this approach can produce accurate results, it is computationally expensive. This analysis is similar to verification, which is described in greater detail in the section “Verification and Testability.”

STA is much faster than DTA. During the STA process, the expected delay of different gates and logic paths is compared to the minimum and maximum allowed delays. The delay of each standard cell—gates and flip-flops—is extracted and compiled into a lookup table (LUT). Certain cell parameters are evaluated to consider manufacturing variations in the fabrication process. The delay of the interconnect lines including the dependence on the load is also included within the LUTs. These tables allow quick estimation and detection of timing violations without requiring a circuit to be simulated on a nodal basis.

TIMING REFERENCE

Existing CMOS-based STA tools can be adapted to RSFQ circuits utilizing PTL routing; inclusion of the PTL driver/receiver within the standard cells further simplifies this process. However, multiple differences related to timing exist between RSFQ logic and conventional CMOS circuits. One of these differences is signaling for the clock and data. RSFQ circuits utilize SFQ voltage pulses for signaling. Although these pulses exhibit a quantized area Φ_0 , the magnitude and duration of these pulses can vary for different gates. Transient noise in RSFQ circuits can appear similar to an SFQ pulse, with the primary difference being the area of the waveform. It is therefore difficult to determine the precise moment when a pulse arrives or is generated—an important issue in the timing analysis of SFQ systems.

One method to determine the timing of a pulse is to monitor the phase

of the input/output JJs within a gate. A pulse incoming to a gate produces a 2π change in the phase of the input JJ. Conversely, the output pulse produces the same change in the phase of the output JJ. The duration of the complete 2π change varies depending upon the damping characteristics and bias conditions [6]. In general, however, it is characterized by a steep slope and a settling time. A common technique to set the precise moment of switching is to use a specific fraction of the 2π change, e.g., 75% [52].

This technique is shown in Figure 8, where both the voltage and phase of a switching JJ are shown. These fractional changes in the phase of a JJ are less dependent on the circuit parameters and settling time, exhibiting smaller variations. This approach, however, cannot be used if the internal structure of the gate is not accessible. In this case, the phase of the JJs within the interconnect can be used. In the case of a JTL interconnect, for example, the phase of the first and last junction of the interconnect can serve as a temporal reference. For a PTL interconnect, the phase of the driver JJs and receiver JJs can be used.

A difference of several picoseconds can occur in the gate timing characteristics extracted from the phase change of the output junction of a gate rather than the peak output voltage. While

both methods are used, it is important to extract timing characteristics in a consistent manner.

TIMING CONSTRAINTS

Another important distinction of RSFQ logic is that most logic gates require a clock signal. Because of this feature, the logic cells are treated as sequential elements. Standard CMOS timing concepts, such as the setup time and hold time, are therefore redefined for RSFQ circuits [32], [124].

Multiple critical timing constraints exist in RSFQ logic circuits [42]. These timing constraints are illustrated in Figure 9. One constraint is the minimum separation time between the input pulses and the clock pulse. If an input pulse arrives too late, the clock signal produces an output based on the incorrect state of the gate, as shown in Figure 9(a). This clock-after-data [27] timing constraint is similar to the CMOS setup time constraint [17]. Another critical timing constraint is the minimum separation time between the clock pulse and any incoming input pulses during the next clock period, as shown in Figure 9(b). If an input pulse arrives too early, the state of the logic gate does not change, producing an error. This data-after-clock [27] separation time is similar to the CMOS hold time constraint [17].

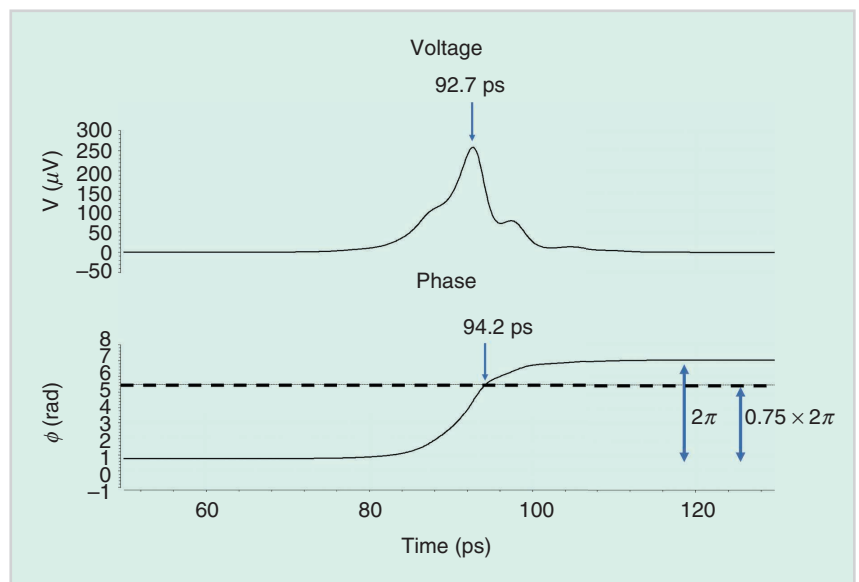


FIGURE 8 Pulse timing based on the phase change of a JJ. The voltage and phase across a JJ are shown during switching. The dashed horizontal line denotes a 75% increase in the phase of the JJ.

A different set of critical timing constraints exists for asynchronous circuits and is related to the minimum separation time between input pulses. An example of a circuit function affected by this timing constraint is the merger. If the separation time is violated, one pulse is produced rather than two, as shown in Figure 9(c). These critical timing constraints are different for each gate, and some of these constraints do not exist for certain gates.

STA TECHNIQUES

Specific algorithms and techniques exist for the STA of RSFQ circuits. A statistical approach for timing analysis has been proposed [125]. In this methodology, variations in the output delay are based on a statistical distribution of several fabrication parameters. In [126], algorithms

for estimating the path delay and timing slack of different logic paths are presented. Based on the timing slack, the minimum system clock period is determined. An STA tool is described in [127], where the critical paths are detected. The timing slack is determined based on the worst-case cell delays and the length of the PTL interconnect between cells.

A similar definition of a critical timing window exists for AQFP circuits [43], where both data and clock are represented as current waveforms. This feature is similar to CMOS signaling and timing constraints, increasing compatibility with standard CMOS timing analysis methodologies [17].

VERIFICATION AND TESTABILITY

The circuit under development is modified during each step of the design flow

(see Figure 1). From logic synthesis to layout and fabrication, many different tools and steps introduce changes into a circuit, which can affect functionality and/or introduce errors. To negate the likelihood of errors, verification is necessary. In this section, verification of both the logical circuit structure and the physical layout is discussed.

The operation of a circuit is initially described in a target specification. These specifications are manually converted into an RTL description. To verify the correctness of this conversion process, confirming that the RTL matches the original specifications, functional verification is performed. As the RTL design process for superconductive circuits is similar to the design process of conventional CMOS circuits, many CMOS-compatible techniques are used. Until recently, the complexity of superconductive circuits was not sufficiently high to require rigorous functional verification. For the 8-bit FLUX-1 processor, composed of approximately 66,000 JJs, functional verification was partially performed for individual blocks using VHDL [128]. These blocks were also individually fabricated and tested before being combined into a single IC. HDL verification has also been used for AQFP [43]. For prospective large-scale superconductive processors, more rigorous functional verification is necessary.

The synthesis process converts an RTL description of a system into an optimized netlist suitable for layout. During this process, errors can be introduced by the synthesis tools. To detect these errors, formal equivalence checking (FEC) is performed [129]. FEC techniques verify that the RTL description is equivalent to the netlist. Alternatively, two netlists can be compared before and after certain transformations. A logical equivalence checking methodology has been developed for RSFQ circuits [130]. A tool combining formal and functional verification has also been proposed for RSFQ circuits [131]. These tools verify any fan-out restrictions and the correctness of the path-balancing process [132]. Upon completion, the circuit is functionally verified using the industry standard Universal Verification Methodology

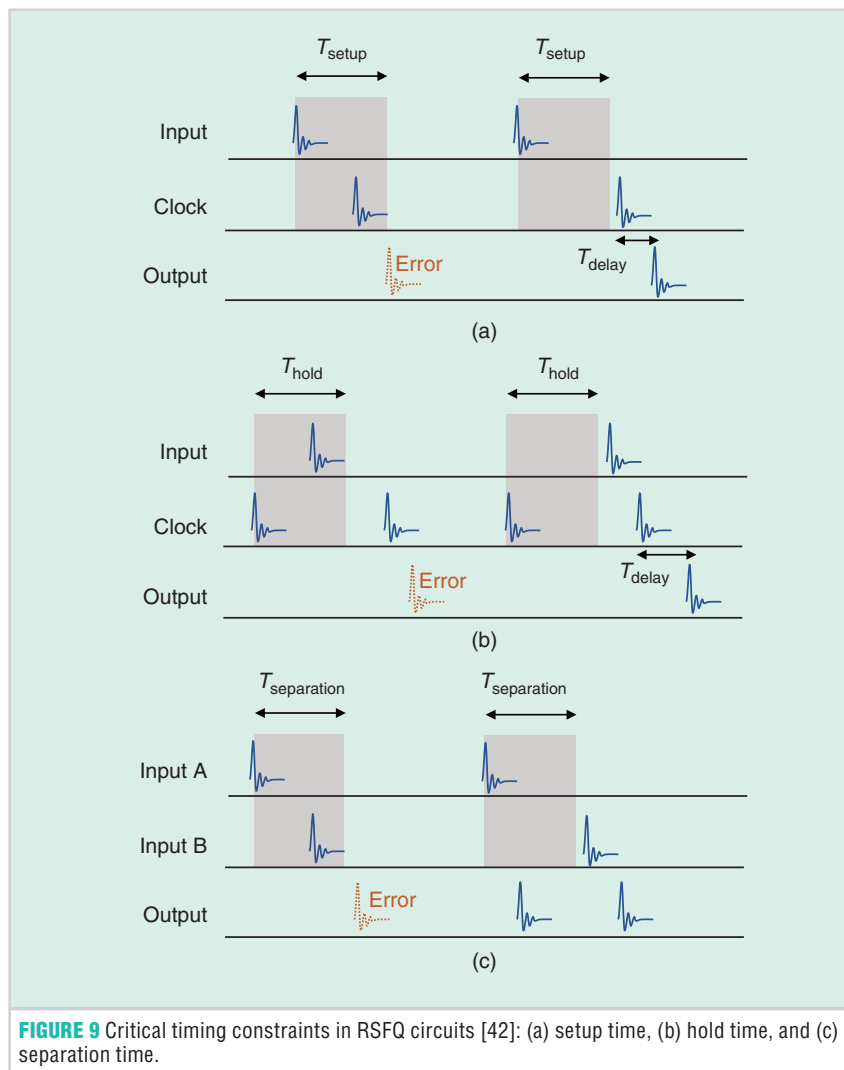


FIGURE 9 Critical timing constraints in RSFQ circuits [42]: (a) setup time, (b) hold time, and (c) separation time.

(UVM) framework [133]. The UVM framework utilizes SystemVerilog features to enable an efficient design process and to facilitate reuse of the verification environment [134].

Once the synthesized netlist is laid out, additional errors can be introduced. Specialized verification steps are therefore required. A DRC verifies that the layout does not violate the geometric spacing rules of the fabrication process. This verification process evaluates the spacings, sizes, and other geometric properties of the different physical structures. An LVS check confirms the equivalence of the layout with the logic netlist. This process extracts a netlist from the physical layout and compares this netlist with the intended logic netlist. Both DRC and LVS are applied to superconductive circuits [51], [135]; Synopsys IC Validator combines these capabilities [51]. For these verification steps, many compatible CMOS methods and tools are used.

Defects can be introduced into a verified circuit during the fabrication process. Parameter variations and environmental conditions can affect the timing characteristics and lead to incorrect operation, particularly in multigigahertz superconductive VLSI systems. To evaluate fabricated circuits, design-for-testability features are included within superconductive circuits [136], [137]. These features enable easier detection and localization of errors during the debug and testing processes.

CONCLUSION

An EDA flow for the design and analysis of superconductive circuits is described in this article. This discussion includes a general background and describes issues specific to superconductive circuits and systems. For each step of the design flow, tools and algorithms are discussed, and sources for more information are provided. Existing standard cell libraries for superconductive circuits and related cell library design and characterization techniques are reviewed. For automated synthesis, methodologies and algorithms are described for logic synthesis and place and route. For simulation and modeling, RTL simulation based on HDLs,

dynamic and static circuit simulators, as well as inductance extraction tools are reviewed. For the verification process, timing analysis methodologies and related timing constraints suitable for modern superconductive circuit families are discussed, and verification approaches are described.

The existing EDA tools and techniques for SCE are highly immature as compared to CMOS EDA tools. Significant research efforts are, however, currently directed at improving and developing algorithms and design methodologies that target superconductive circuits. The effectiveness of these tools to enable large-scale superconductive digital systems is greatly improving.

ACKNOWLEDGMENT

The effort is supported by the U.S. Department of Defense Agency–Intelligence Advanced Research Projects Activity through the U.S. Army Research Office under contract W911NF-17-9-0001. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

ABOUT THE AUTHORS

Gleb Krylov (gleb@urgrad.rochester.edu) is with the Department of Electrical and Computer Engineering, the University of Rochester, Rochester, New York, 14627, USA.

Jamil Kawa (jamil@synopsys.com) is with Synopsys Inc., Mountain View, California, 94043, USA.

Eby G. Friedman (friedman@ece.rochester.edu) is with the Department of Electrical and Computer Engineering, the University of Rochester, Rochester, New York, 14627, USA.

REFERENCES

- [1] M. A. Manheimer, "Cryogenic computing complexity program: Phase 1 introduction," *IEEE Trans. Appl. Superconductivity*, vol. 25, no. 3, pp. 1–4, June 2015.
- [2] 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. Superconductivity*, vol. 1, no. 1, pp. 3–28, Mar. 1991.
- [3] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Superconductor Sci. Technol.*, vol. 26, no. 3, p. 035010, Jan. 2013.

- [4] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, "Ultra-low-power superconductor logic," *J. Appl. Phys.*, vol. 109, no. 10, p. 103903, May 2011.
- [5] S. K. Tolpygo et al., "Advanced fabrication processes for superconductor electronics: Current status and new developments," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–13, Aug. 2019.
- [6] G. Krylov and E. G. Friedman, *Single Flux Quantum Integrated Circuit Design*. Berlin: Springer-Verlag, 2022.
- [7] A. B. Kahng, J. Licnig, I. L. Markov, and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Netherlands: Springer-Verlag, 2011.
- [8] K. Jackman and C. J. Fourie, "Flux trapping analysis in superconducting circuits," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 4, pp. 1–5, June 2017.
- [9] K. Gaj, Q. P. Herr, V. Adler, D. K. Brock, E. G. Friedman, and M. J. Feldman, "Toward a systematic design methodology for large multigigahertz rapid single flux quantum circuits," *IEEE Trans. Appl. Superconductivity*, vol. 9, no. 3, pp. 4591–4606, Sept. 1999.
- [10] Stony Brook University. SUNY RSFQ Cell Library. [Online]. Available: <http://www.physics.sunysb.edu/Physics/RSFQ/Lib/contents.html>
- [11] Technische Universität Ilmenau. RSFQ—Cell Library. [Online]. Available: <https://www.alt.tu-ilmenau.de/it-tet/forschung/supraleitende-hochgeschwindigkeits-elektronik/rsfq-cell/cells/>
- [12] S. Anders et al., "European roadmap on superconductive electronics—Status and perspectives," *Phys. C, Superconductivity*, vol. 470, no. 23, pp. 2079–2126, Dec. 2010.
- [13] S. Yorozu, Y. Kameda, H. Terai, A. Fujimaki, T. Yamada, and S. Tahara, "A single flux quantum standard logic cell library," *Phys. C, Superconductivity*, vols. 378–381, pp. 1471–1474, Oct. 2002.
- [14] S. Tahara, H. Numata, S. Yorozu, Y. Hashimoto, and S. Nagasawa, "Superconducting technology for digital applications using niobium josephson junctions," *IEICE Trans. Electron.*, vol. 83, no. 1, pp. 60–68, Jan. 2000.
- [15] M. Maezawa, M. Ochiai, H. Kimura, F. Hirayama, and M. Suzuki, "Design and operation of RSFQ cell library fabricated by using a 10-KA/cm² Nb technology," *IEEE Trans. Appl. Superconductivity*, vol. 17, no. 2, pp. 500–504, June 2007.
- [16] M. Maezawa, F. Hirayama, and M. Suzuki, "Design and fabrication of RSFQ cell library for middle-scale applications," *Phys. C, Superconductivity*, vols. 412–414, pp. 1591–1596, Oct. 2004.
- [17] E. Salzman and E. G. Friedman, *High Performance Integrated Circuit Design*. New York: McGraw-Hill, 2012.
- [18] M. El-Moursy and E. G. Friedman, *On-Chip Inductive Interconnect Design Methodologies*, VDM Verlag Dr. Muller Aktiengesellschaft & Company, 2009.
- [19] J. Rosenfeld and E. G. Friedman, "Design methodology for global resonant H-tree clock distribution networks," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 15, no. 2, pp. 135–148, Feb. 2007.
- [20] T. Jabbari, G. Krylov, S. Whiteley, E. Mlinar, J. Kawa, and E. G. Friedman, "Interconnect routing for large-scale RSFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–5, Aug. 2019.
- [21] S. K. Tolpygo et al., "Advanced fabrication processes for superconducting very large-scale integrated circuits," *IEEE Trans. Appl. Superconductivity*, vol. 26, no. 3, pp. 1–10, Apr. 2016.
- [22] A. Inamdar, D. Amparo, B. Sahoo, J. Ren, and A. Sahu, "RSFQ/ERSFQ cell library with improved circuit optimization, timing verification, and test characterization," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 4, pp. 1–9, June 2017.
- [23] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Superconductivity*, vol. 21, no. 3, pp. 760–769, June 2011.

- [24] G. Krylov and E. G. Friedman, "Design methodology for distributed large scale ERSFQ bias networks," *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, vol. 28, no. 11, pp. 2438–2447, Nov. 2020.
- [25] S. S. Meher, C. Kanungo, A. Shukla, and A. Inamdar, "Parametric approach for routing power nets and passive transmission lines as part of digital cells," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–7, Aug. 2019.
- [26] G. Krylov and E. G. Friedman, "Bias distribution in ERSFQ VLSI circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, Oct. 2020, pp. 1–5.
- [27] D. Amparo, M. Eren Çelik, S. Nath, J. P. Cerqueira, and A. Inamdar, "Timing characterization for RSFQ cell library," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–9, Aug. 2019.
- [28] 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, p. 054006, Mar. 2020.
- [29] Y. He et al., "A compact AQFP logic cell design using an 8-metal layer superconductor process," *Superconductor Sci. Technol.*, vol. 33, no. 3, p. 035010, Feb. 2020.
- [30] C. L. Ayala, O. Chen, and N. Yoshikawa, "AQF-PTX: Adiabatic quantum-flux-parametron timing extraction tool," in *Proc. IEEE Int. Superconductive Electron. Conf.*, Aug. 2019, pp. 1–3.
- [31] O. T. Oberg, "Superconducting logic circuits operating with reciprocal magnetic flux quanta," Ph.D. dissertation, Univ. of Maryland, College Park, 2011.
- [32] K. Gaj, C.-H. Cheah, E. G. Friedman, and M. J. Feldman, "Functional modeling of RSFQ circuits using Verilog HDL," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 3151–3154, June 1997.
- [33] A. Krasniewski, "Logic simulation of RSFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 3, no. 1, pp. 33–38, Mar. 1993.
- [34] S. V. Polonsky, V. K. Semenov, and P. N. Shevchenko, "PSCAN: Personal superconductor circuit analyser," *Superconductor Sci. Technol.*, vol. 4, no. 11, pp. 667–670, Nov. 1991.
- [35] P. Bunyk, A. Y. Kidiyarova-Shevchenko, and P. Litskevitch, "RSFQ microprocessor: New design approaches," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 2697–2704, June 1997.
- [36] H. Toepfer, T. Harnisch, J. Kunert, S. Lange, and H. F. Uhlmann, "Formal description of the functional behavior of RSFQ logic circuits for design and optimization purposes," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 3630–3633, June 1997.
- [37] F. Matsuzaki, N. Yoshikawa, M. Tanaka, A. Fujimaki, and Y. Takai, "A behavioral-level HDL description of SFQ logic circuits for quantitative performance analysis of large-scale SFQ digital systems," *Phys. C, Superconductivity*, vols. 392–396, pp. 1495–1500, Oct. 2003.
- [38] P. Bunyk and P. Litskevitch, "Case study in RSFQ design: Fast pipelined parallel adder," *IEEE Trans. Appl. Superconductivity*, vol. 9, no. 2, pp. 3714–3720, June 1999.
- [39] S. Intiso, I. Kataeva, E. Tolkacheva, H. Engseth, K. Platov, and A. Kidiyarova-Shevchenko, "Time-delay optimization of RSFQ cells," *IEEE Trans. Appl. Superconductivity*, vol. 15, no. 2, pp. 328–331, June 2005.
- [40] A. K. Kasperek, "32-bit superconductor integer and floating-point multipliers," Ph.D. dissertation, Stony Brook Univ., New York, 2012.
- [41] L. C. Müller and C. J. Fourie, "Automated state machine and timing characteristic extraction for RSFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 24, no. 1, pp. 3–12, Feb. 2014.
- [42] C. J. Fourie, "Extraction of DC-biased SFQ circuit verilog models," *IEEE Trans. Appl. Superconductivity*, vol. 28, no. 6, pp. 1–11, Sept. 2018.
- [43] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "HDL-based modeling approach for digital simulation of adiabatic quantum flux parametron logic," *IEEE Trans. Appl. Superconductivity*, vol. 26, no. 8, pp. 1–5, Dec. 2016.
- [44] R. N. Tadros, A. Fayyazi, M. Pedram, and P. A. Beerel, "SystemVerilog modeling of SFQ and AQFP circuits," *IEEE Trans. Appl. Superconductivity*, vol. 30, no. 2, pp. 1–13, Mar. 2020.
- [45] L. W. Nagel, and D. O. Pederson, "SPICE (Simulation Program with Integrated Circuit Emphasis)," Univ. of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr. 1973.
- [46] S. R. Whiteley, "Josephson junctions in SPICE3," *IEEE Trans. Magnetics*, vol. 27, no. 2, pp. 2902–2905, Mar. 1991.
- [47] E. S. Fang and T. V. Duzer, "A Josephson integrated circuit simulator (jsim) for superconductive electronics application," in *Proc. IEEE Int. Superconductive Electron. Conf.*, pp. 407–410, June 1989.
- [48] S. R. Whiteley. WRspice reference manual. Whiteley Research Inc. [Online]. Available: <http://www.wrcad.com/manual/wrsmmanual.pdf>
- [49] K. K. Likharev, *Dynamics of Josephson Junctions and Circuits*, Gordon and Breach Science Publishers, 1986.
- [50] S. R. Whiteley and J. Kawa, "Progress toward VLSI-capable EDA tools for superconductive digital electronics," in *Proc. IEEE Int. Superconductive Electron. Conf.*, July 2019, pp. 1–3.
- [51] V. Adler, C.-H. Cheah, K. Gaj, D. K. Brock, and E. G. Friedman, "A cadence-based design environment for single flux quantum circuits," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 3294–3297, June 1997.
- [52] K. Gaj, Q. P. Herr, V. Adler, A. Krasniewski, E. G. Friedman, and M. J. Feldman, "Tools for the computer-aided design of multigigahertz superconducting digital circuits," *IEEE Trans. Appl. Superconductivity*, vol. 9, no. 1, pp. 18–38, Mar. 1999.
- [53] J. A. Delpoit, K. Jackman, P. I. Roux, and C. J. Fourie, "JoSIM – Superconductor SPICE simulator," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–5, Aug. 2019.
- [54] S. Polonsky, P. Shevchenko, A. Kirichenko, D. Zinoviev, and A. Rylyakov, "PSCAN'96: New software for simulation and optimization of complex RSFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 2685–2689, June 1997.
- [55] P. Shevchenko, "PSCAN2 superconductor circuit simulator." [Online]. Available: <http://www.pscan2sim.org/documentation.html>
- [56] N. R. Werthamer, "Nonlinear self-coupling of Josephson radiation in superconducting tunnel junctions," *Phys. Rev.*, vol. 147, pp. 255–263, July 1966.
- [57] A. Odintsov, V. Semenov, and A. Zorin, "Specific problems of numerical analysis of the Josephson junction circuits," *IEEE Trans. Magnetics*, vol. 23, no. 2, pp. 763–766, Mar. 1987.
- [58] A. De Lustrac, P. Crozat, and R. Adde, "A picosecond Josephson junction model for circuit simulation," *Revue de Physique Appliquée*, vol. 21, no. 5, pp. 319–326, May 1986.
- [59] R. Freeman, J. Kawa, and K. Singhal, "Synopsys' journey to enable TCAD and EDA tools for superconducting electronics," in *Proc. Government Microcircuit Appl. Critical Technol. Conf.*, Mar. 2020.
- [60] S. K. Tolpygo et al., "Inductance of circuit structures for MIT LL superconductor electronics fabrication process with 8 niobium layers," *IEEE Trans. Appl. Superconductivity*, vol. 25, no. 3, pp. 1–5, June 2015.
- [61] *Sonnet User's Guide*, Sonnet Software Inc. [Online]. Available: <https://www.sonnetsoftware.com/support/downloads/guide.pdf>
- [62] J. C. Rautio and R. F. Harrington, "An electromagnetic time-harmonic analysis of shielded microstrip circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 35, no. 8, pp. 726–730, Aug. 1987.
- [63] A. R. Kerr, "Surface impedance of superconductors and normal conductors in EM simulators," *Nat. Radio Astronomy Observatory, Green Bank, WV, Electronics Division Internal Rep.* no. 302, Feb. 1996.
- [64] "3D Electromagnetic field simulator for RF and wireless design," Ansys Inc., Canonsburg, PA. Accessed: Oct. 7, 2021. [Online]. Available: <https://www.ansys.com/products/electronics/ansys-hfs>
- [65] K. U-Yen, K. Rostem, and E. J. Wollack, "Modeling strategies for superconducting microstrip transmission line structures," *IEEE Trans. Appl. Superconductivity*, vol. 28, no. 6, pp. 1–5, Sept. 2018.
- [66] M. Kamon, M. J. Tsuk, and J. K. White, "FASTHENRY: A multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microw. Theory Techniques*, vol. 42, no. 9, pp. 1750–1758, Sept. 1994.
- [67] I. P. Vaisband, R. Jakushokas, M. Popovich, A. V. Mezhiba, S. Köse, and E. G. Friedman, *On-Chip Power Delivery and Management*, 4th ed. Berlin: Springer-Verlag, 2016.
- [68] B. Guan, M. J. Wengler, P. Rott, and M. J. Feldman, "Inductance estimation for complicated superconducting thin film structures with a finite segment method," *IEEE Trans. Appl. Superconductivity*, vol. 7, no. 2, pp. 2776–2779, June 1997.
- [69] S. R. Whiteley. FastHenry 3.0wr, Sunnyvale, CA. Accessed: Oct. 7, 2021. [Online]. Available: http://www.wrcad.com/ftp/pub/README_FASTHENRY
- [70] F. London and H. London, "The electromagnetic equations of the superconductor," *Proc. Roy. Soc. London. A, Math. Phys. Sci.*, vol. 149, no. 866, pp. 71–88, Mar. 1935.
- [71] G. Krylov and E. G. Friedman, "Inductive noise coupling in superconductive passive transmission lines," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2021, pp. 727–731.
- [72] C. J. Fourie, O. Wetzstein, T. Ortlev, and J. Kunert, "Three-dimensional multi-terminal superconductive integrated circuit inductance extraction," *Superconductor Sci. Technol.*, vol. 24, no. 12, p. 125015, Nov. 2011.
- [73] C. J. Fourie, C. Shawawreh, I. V. Vernik, and T. V. Filippov, "High-accuracy InductEX calibration sets for MIT-LL SFQ4ee and SFQ5ee processes," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 2, pp. 1–5, Jan. 2017.
- [74] K. Jackman and C. J. Fourie, "Fast multicore FastHenry and a tetrahedral modeling method for inductance extraction of complex 3D geometries," in *Proc. IEEE Int. Superconductive Electron. Conf.*, July 2015, pp. 1–3.
- [75] K. Jackman and C. J. Fourie, "Tetrahedral modeling method for inductance extraction of complex 3-D superconducting structures," *IEEE Trans. Appl. Superconductivity*, vol. 26, no. 3, pp. 1–5, Apr. 2016.
- [76] M. M. Khapaev, "inductance extraction of multilayer finite-thickness superconductor circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 49, no. 1, pp. 217–220, Jan. 2001.
- [77] M. M. Khapaev, A. Y. Kidiyarova-Shevchenko, P. Magnelin, and M. Y. Kupriyanov, "3D-MLSI: Software package for inductance calculation in multilayer superconducting integrated circuits," *IEEE Trans. Appl. Superconductivity*, vol. 11, no. 1, pp. 1090–1093, Mar. 2001.
- [78] M. M. Khapaev, M. Y. Kupriyanov, E. Goldobin, and M. Siegel, "Current distribution simulation for superconducting multi-layered structures," *Superconductor Sci. Technol.*, vol. 16, no. 1, pp. 24–27, Nov. 2002.
- [79] M. M. Khapaev and M. Y. Kupriyanov, "Inductance extraction of superconductor structures with internal current sources," *Superconductor Sci. Technol.*, vol. 28, no. 5, p. 055013, Apr. 2015.
- [80] H. Kumar, T. Jabbari, G. Krylov, K. Basu, E. G. Friedman, and R. Karri, "Toward increasing the difficulty of reverse engineering of RSFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 30, no. 3, pp. 1–13, Apr. 2020.

- [81] T. Jabbari, G. Krylov, and E. G. Friedman, "Logic locking in SFQ technology," *IEEE Trans. Appl. Superconductivity*, vol. 31, no. 5, pp. 1–5, Aug. 2021.
- [82] N. Yoshikawa and J. Koshiyama, "Top-down RSFQ logic design based on a binary decision diagram," *IEEE Trans. Appl. Superconductivity*, vol. 11, no. 1, pp. 1098–1101, Mar. 2001.
- [83] S. B. Akers, "Binary decision diagrams," *IEEE Trans. Comput.*, vol. C-27, no. 6, pp. 509–516, June 1978.
- [84] ABC: A System for Sequential Synthesis and Verification. Berkeley Logic Synthesis and Verification Group. [Online]. Available: <http://www.eecs.berkeley.edu/texts/citildealanmi/abc/>
- [85] J. A. Darringer, W. H. Joyner, C. L. Berman, and L. Trevillyan, "Logic synthesis through local transformations," *IBM J. Res. Develop.*, vol. 25, no. 4, pp. 272–280, July 1981.
- [86] L. Amarú, P. Gaillardon, and G. De Micheli, "Majority-inverter graph: A new paradigm for logic optimization," *IEEE Trans. Comput.-Aided Design Integrated Circuits Syst.*, vol. 35, no. 5, pp. 806–819, May 2016.
- [87] K. Inoue, N. Takeuchi, K. Ehara, Y. Yamanashi, and N. Yoshikawa, "Simulation and experimental demonstration of logic circuits using an ultra-low-power adiabatic quantum-flux-parametron," *IEEE Trans. Appl. Superconductivity*, vol. 23, no. 3, p. 1301105, June 2013.
- [88] G. Krylov and E. G. Friedman, "Asynchronous dynamic single flux quantum majority gates," *IEEE Trans. Appl. Superconductivity*, vol. 30, no. 5, pp. 1–7, Aug. 2020.
- [89] L. Amarú, P. Gaillardon, A. Chattopadhyay, and G. De Micheli, "A sound and complete axiomatization of majority- n logic," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2889–2895, Sept. 2016.
- [90] C. Wolf. "Yosys Open Synthesis Suite." Accessed: Oct. 7, 2021. [Online]. Available: <http://www.clifford.at/yosys/>
- [91] Q. Xu, C. L. Ayala, N. Takeuchi, Y. Murai, Y. Yamanashi, and N. Yoshikawa, "Synthesis flow for cell-based adiabatic quantum-flux-parametron structural circuit generation with HDL back-end verification," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 4, pp. 1–5, Jan. 2017.
- [92] M. Pedram and Y. Wang, "Design automation methodology and tools for superconductive electronics," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2018, pp. 1–6.
- [93] N. Katam, A. Shafaei, and M. Pedram, "Design of complex rapid single-flux-quantum cells with application to logic synthesis," in *Proc. IEEE Int. Superconductive Electron. Conf.*, June 2017, pp. 1–3.
- [94] G. Krylov and E. G. Friedman, "Globally asynchronous, locally synchronous clocking and shared interconnect for large-scale SFQ systems," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–5, Aug. 2019.
- [95] G. Pasandi and M. Pedram, "PBMap: A path balancing technology mapping algorithm for single flux quantum logic circuits," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 4, pp. 1–14, Nov. 2019.
- [96] X. Liu, M. C. Papaefthymiou, and E. G. Friedman, "Retiming and clock scheduling for digital circuit optimization," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 21, no. 2, pp. 184–203, Aug. 2002.
- [97] T. Soyata, E. G. Friedman, and J. H. Mulligan Jr., "Incorporating interconnect, register, and clock distribution delays into the retiming process," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 16, no. 1, pp. 105–120, Jan. 1997.
- [98] N. Kito, K. Takagi, and N. Takagi, "Conversion of a CMOS logic circuit design to an RSFQ design considering latching function of RSFQ logic gates," *IEEE Trans. Appl. Superconductivity*, vol. 25, no. 3, pp. 1–5, June 2015.
- [99] G. Krylov and E. G. Friedman, "Partitioning RSFQ circuits for current recycling," *IEEE Trans. Appl. Superconductivity*, vol. 31, no. 5, pp. 1–6, Aug. 2021.
- [100] Y. Kameda, S. Yorozu, and Y. Hashimoto, "A New design methodology for single-flux-quantum (SFQ) logic circuits using passive-transmission-line (PTL) wiring," *IEEE Trans. Appl. Superconductivity*, vol. 17, no. 2, pp. 508–511, June 2007.
- [101] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. ACM/IEEE Des. Automat. Conf.*, June 1982, pp. 175–181.
- [102] S. Whiteley, E. Mlinar, G. Krylov, T. Jabbari, E. G. Friedman, and J. Kawa, "An SFQ digital circuit technology with fully-passive transmission line interconnect," in *Proc. Applied Superconductivity Conf.*, Nov. 2020.
- [103] M. Tanaka et al., "Automated passive-transmission-line routing tool for single-flux-quantum circuits based on A* algorithm," *IEICE Trans. Electron.*, vol. E93.C, no. 4, pp. 435–439, Apr. 2010.
- [104] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, July 1968.
- [105] T. Jabbari, G. Krylov, S. Whiteley, J. Kawa, and E. G. Friedman, "Repeater insertion in SFQ interconnect," *IEEE Trans. Appl. Superconductivity*, vol. 30, no. 8, pp. 1–8, Dec. 2020.
- [106] N. Kito, K. Takagi, and N. Takagi, "Automatic wire-routing of SFQ Digital circuits considering wire-length matching," *IEEE Trans. Appl. Superconductivity*, vol. 26, no. 3, pp. 1–5, Jan. 2016.
- [107] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY: Dover, 1998.
- [108] V. F. Pavlidis, I. Savidis, and E. G. Friedman, *Three-Dimensional Integrated Circuit Design*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2017.
- [109] N. Kito, K. Takagi, and N. Takagi, "A fast wire-routing method and an automatic layout tool for RSFQ digital circuits considering wire-length matching," *IEEE Trans. Appl. Superconductivity*, vol. 28, no. 4, pp. 1–5, Jan. 2018.
- [110] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [111] P. Cheng, K. Takagi, and T. Ho, "Multi-terminal routing with length-matching for rapid single flux quantum circuits," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Jan. 2018, pp. 1–6.
- [112] S. N. Shahsavani, A. Shafaei, and M. Pedram, "A placement algorithm for superconducting logic circuits based on cell grouping and super-cell placement," in *Proc. ACM/IEEE Des., Automat. Test Europe Conf.*, Mar. 2018, pp. 1465–1468.
- [113] S. N. Shahsavani, T. Lin, A. Shafaei, C. J. Fourie, and M. Pedram, "An integrated row-based cell placement and interconnect synthesis tool for large SFQ logic circuits," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 4, pp. 1–8, Mar. 2017.
- [114] T. Edwards, Open Circuit Design. [Online]. Available: <http://opencircuitdesign.com/grouter/index.html>
- [115] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, no. 3, pp. 346–365, Sept. 1961.
- [116] M. Kim, D. Lee, and I. L. Markov, "SimPL: An effective placement algorithm," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 31, no. 1, pp. 50–60, Jan. 2012.
- [117] T. Dejima, K. Takagi, and N. Takagi, "Placement and routing methods based on mixed wiring of JTTLs and PTLs for RSFQ circuits," in *Proc. IEEE Int. Superconductive Electron. Conf.*, July 2019, pp. 1–3.
- [118] S. Nath, K. English, A. Derrickson, A. Haslam, and J. F. McDonald, "An automatic placement and routing methodology for asynchronous SFQ circuit design," *IEEE Trans. Appl. Superconductivity*, vol. 30, no. 3, pp. 1–10, Apr. 2020.
- [119] Y. Murai, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Development and demon-
- stration of routing and placement EDA tools for large-scale adiabatic quantum-flux-parametron circuits," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 6, pp. 1–9, June 2017.
- [120] T. Tanaka, C. L. Ayala, Q. Xu, R. Saito, and N. Yoshikawa, "Fabrication of adiabatic quantum-flux-parametron integrated circuits using an automatic placement tool based on genetic algorithms," *IEEE Trans. Appl. Superconductivity*, vol. 29, no. 5, pp. 1–6, Feb. 2019.
- [121] J. Lienig and K. Thulasiraman, "A genetic algorithm for channel routing in VLSI circuits," *Evol. Comput.*, vol. 1, no. 4, pp. 293–311, Dec. 1993.
- [122] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 1, no. 1, pp. 25–35, Jan. 1982.
- [123] T. Jabbari, G. Krylov, J. Kawa, and E. G. Friedman, "Splitter trees in single flux quantum circuits," *IEEE Trans. Appl. Superconductivity*, vol. 31, no. 5, pp. 1–6, Aug. 2021.
- [124] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 16, no. 2, pp. 247–276, June 1997.
- [125] M. E. Çelik and A. Bozbeý, "Statistical timing analysis tool for SFQ cells (STATS)," in *Proc. IEEE Int. Superconductive Electron. Conf.*, July 2013, no. PA23, pp. 1–3.
- [126] T. Kawaguchi, K. Takagi, and N. Takagi, "Static timing analysis of rapid single-flux-quantum circuits," in *Proc. Workshop on Synthesis Syst. Integration Mixed Inf. Technol.*, Oct. 2016, pp. 341–345.
- [127] J. A. Delpert and C. J. Fourie, "A static timing analysis tool for RSFQ and ERSFQ superconducting digital circuit applications," *IEEE Trans. Appl. Superconductivity*, vol. 28, no. 5, pp. 1–5, Mar. 2018.
- [128] M. Dorojevets, "Architecture and design of an 8-bit FLUX-1 superconductor RSFQ microprocessor," *Int. J. High Speed Electron. Syst.*, vol. 12, no. 2, pp. 521–529, June 2002.
- [129] S.-Y. Huang and K.-T. T. Cheng, *Formal Equivalence Checking and Design Debugging*. New York: Springer Science & Business Media, 2012.
- [130] A. Fayyazi, S. Nazarian, and M. Pedram, "qEC: A logical equivalence checking framework targeting SFQ superconducting circuits," in *Proc. IEEE Int. Superconductive Electron. Conf.*, Aug. 2019, pp. 1–3.
- [131] A. D. Wong, K. Su, H. Sun, A. Fayyazi, M. Pedram, and S. Nazarian, "VeriSFQ: A semi-formal verification framework and benchmark for single flux quantum technology," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, Mar. 2019, pp. 224–230.
- [132] G. Krylov and E. G. Friedman, "Wave pipelining in DSFQ circuits," *IEEE Trans. Appl. Superconductivity*, to be published. doi: 10.1155/2008/738983.
- [133] *IEEE Standard for Universal Verification Methodology Language Reference Manual*, IEEE Standard 1800.2-2020 (Revision of IEEE Standard 1800.2-2017), pp. 1–458, Sept. 2020.
- [134] I. Stotland, D. Shpagilev, and N. Starikovskaya, "UVM based approaches to functional verification of communication controllers of microprocessor systems," in *Proc. IEEE East-West Des. Test Symp.*, Oct. 2016, pp. 1–4.
- [135] R. M. C. Roberts and C. J. Fourie, "Layout-versus-schematic verification for superconductive integrated circuits," *IEEE Trans. Appl. Superconductivity*, vol. 25, no. 3, pp. 1–5, June 2015.
- [136] G. Krylov and E. G. Friedman, "Test point insertion for RSFQ circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2017, pp. 2022–2025.
- [137] G. Krylov and E. G. Friedman, "Design for testability of SFQ circuits," *IEEE Trans. Appl. Superconductivity*, vol. 27, no. 8, pp. 1–7, Dec. 2017.
- [138] T. Jabbari, E. G. Friedman, and J. Kawa, "H-tree clock synthesis in RSFQ circuits," in *Proc. Baltic Electron. Conf.*, Oct. 2020, pp. 1–5. doi: 10.1109/BEC49624.2020.9277224.