

Results From the ColdFlux Superconductor Integrated Circuit Design Tool Project

Coenrad J. Fourie^{ID}, Senior Member, IEEE, Kyle Jackman^{ID}, Member, IEEE, Johannes Delpot^{ID}, Member, IEEE, Lieze Schindler^{ID}, Member, IEEE, Tessa Hall^{ID}, Pascal Febvre^{ID}, Senior Member, IEEE, Lucas Iwanikow^{ID}, Olivia Chen^{ID}, Senior Member, IEEE, Christopher L. Ayala^{ID}, Senior Member, IEEE, Nobuyuki Yoshikawa^{ID}, Senior Member, IEEE, Mark Law^{ID}, Fellow, IEEE, Thomas A. Weingartner^{ID}, Yanzhi Wang^{ID}, Peter Beerel^{ID}, Sandeep Gupta^{ID}, Haipeng Zha^{ID}, Sasan Razmkhah^{ID}, Mustafa Altay Karamuftuoglu^{ID}, Arash Fayyazi, Mingye Li, Murali Annavaram, Shahin Nazarian, and Massoud Pedram^{ID}, Fellow, IEEE

Abstract—In five and a half years, the ColdFlux project under the IARPA SuperTools program pushed the boundaries of digital and analog superconductor electronic design automation (S-EDA) tools. The SuperTools program demanded significant beyond-state-of-the-art deliverables in four main areas: RTL synthesis, architectures, and verification; analog design and layout synthesis; physical design and test; and technology CAD and cell library design. Through the work of academic groups scattered over four continents, the ColdFlux effort forged into a powerful set of open-source and commercial S-EDA tools unlike any before, rivaled only by a commercial toolchain from Synopsys under the same SuperTools umbrella. We present an overview of the tools from where we started to the eventual project deliverables. These

Manuscript received 20 April 2023; revised 29 July 2023; accepted 14 August 2023. Date of publication 18 August 2023; date of current version 14 September 2023. This work was supported by the Office of the Director of National Intelligence, Intelligence Advanced Research Projects Activity, SuperTools Program, via the U.S. Army Research Office under Grant W911NF-17-1-0120. (Corresponding author: Coenrad J. Fourie.)

Coenrad J. Fourie, Kyle Jackman, Johannes Delpot, and Tessa Hall are with Stellenbosch University, Stellenbosch 7602, South Africa (e-mail: coenrad@sun.ac.za; kjackman@sun.ac.za; jdelpot@sun.ac.za; 19775539@sun.ac.za).

Lieze Schindler is with Stellenbosch University, Stellenbosch 7602, South Africa, and also with the Institute of Advanced Sciences, Yokohama National University, Yokohama 240-0067, Japan (e-mail: liezeschindler@gmail.com).

Christopher L. Ayala and Nobuyuki Yoshikawa are with the Institute of Advanced Sciences, Yokohama National University, Yokohama 240-0067, Japan (e-mail: ayala-christopher-pz@ynu.ac.jp; yoshikawa-nobuyuki-gt@ynu.ac.jp).

Pascal Febvre and Lucas Iwanikow are with the Institute of Microelectronics, Electromagnetism and Photonics and the Microwave and Characterization Laboratory, Université Savoie Mont Blanc, 73000 Chambéry, France (e-mail: pascal.febvre@univ-smb.fr; lucas.iwanikow@univ-smb.fr).

Olivia Chen is with the Department of Computer Science, Tokyo City University, Setagaya 158-8557, Japan (e-mail: oliviach@g.tcu.ac.jp).

Mark Law and Thomas A. Weingartner are with the University of Florida, Gainesville, FL 32611 USA (e-mail: law@ece.ufl.edu; t.weingartner@ufl.edu).

Yanzhi Wang is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: yanzhi.wang@northeastern.edu).

Peter Beerel, Sandeep Gupta, Haipeng Zha, Sasan Razmkhah, Mustafa Altay Karamuftuoglu, Arash Fayyazi, Mingye Li, Murali Annavaram, Shahin Nazarian, and Massoud Pedram are with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90007 USA (e-mail: pabeerel@usc.edu; sandeep@usc.edu; hzh@usc.edu; razmkhah@usc.edu; karamuft@usc.edu; fayyazi@usc.edu; mingye@usc.edu; annavara@usc.edu; shahin.nazarian@usc.edu; pedram@usc.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TASC.2023.3306381>.

Digital Object Identifier 10.1109/TASC.2023.3306381

include powerful simulation and extraction engines, magnetic field and flux trapping analysis, advanced clocking methods, multi-chip interface extraction and verification, unified multi-layer design-rule compliant track blocks for automated place and route of both rapid single flux quantum (RSFQ) and adiabatic quantum-flux-parametron (AQFP) cells, models and tools for validation and test, multi-bit single flux quantum (SFQ) cells, architecture innovations for full CPU designs and more. Comprehensive cell libraries and a process design kit (PDK) were developed with the ColdFlux tools. The AQFP cell library features a logically rich collection of 80+ cells, including 3- and 5-input logic gates, signal-driving boosters, and refined RSFQ-to-AQFP interfaces, while the RSFQ library has 30+ cells. Finally, we discuss how the full toolchain enables and enhances the superconductor IC design process.

Index Terms—Compact model, electronic design automation tools, flux trapping, inductance extraction, moats, superconductor electronics (SCE).

I. INTRODUCTION

THE ColdFlux project, which ran in parallel with a development project by Synopsys, Inc., under the Intelligence Advanced Research Projects Activity (IARPA) SuperTools program [1], was focused on the development of both front-end and back-end superconducting electronic design automation (S-EDA) and technology computer-aided design (TCAD). SuperTools was initiated after many papers, and technology assessments over the years identified critical shortcomings in the software toolchain used for superconductor digital integrated circuit (IC) design and verification [2], [3], and hardware development projects such as the IARPA Cryogenic Computing Complexity Program (C3) [4] exposed the limits of large-scale superconductor circuit design without dedicated tools.

The SuperTools program set very ambitious goals. A set of tools had to be developed to enable very large scale integration design and verification of superconductor electronics (SCE) as a step toward developing energy-efficient scalable high-performance computers. As a proof-of-concept demonstration, the ColdFlux team would undertake the design of a 64-bit reduced instruction set computer (RISC) microprocessor with the tools and cell libraries developed under the project. Four technical focus areas were mandated, and augmented by fabrication and testing, as illustrated in Fig. 1.

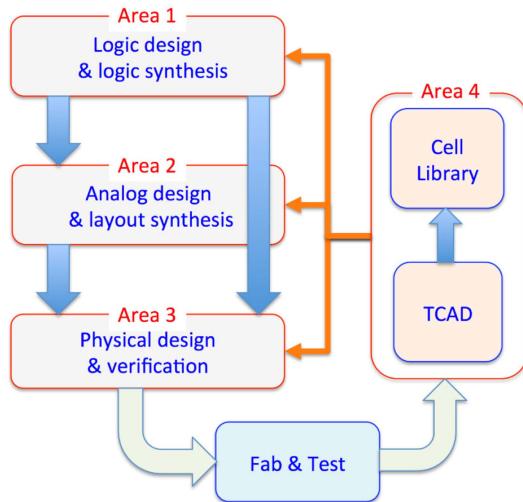


Fig. 1. SuperTools focus areas and process flow.

In the absence of commercially driven development that powered semiconductor design tool development, SuperTools provided access to vast resources in terms of physicists, engineers, and applied mathematicians in applied superconductivity, as well as fabrication runs to test concepts and tools. The SuperTools program enabled research on the most effective design methods and allowed us to convert these to software tools that mesh together to form an effective toolchain, *can function outside of the laboratory*, can be maintained and expanded, and can adapt to evolving technologies.

To keep this article to a reasonable length, we mainly focus on results here, referencing all the relevant research outputs that stemmed from ColdFlux to allow further reading.

A spirit of friendly competition existed between ColdFlux and the parallel project headed by Synopsys, with sharing of ideas leading to better progress. The Synopsys project also delivered a rich set of results and publications, and the reader is advised to explore that, starting with results on a full arithmetic logic unit [5].

This article concludes with directions for future S-EDA development from the ColdFlux tool suite.

II. COLDFLUX STRUCTURE

A. Performers

Tool development for the ColdFlux project was managed over four continents, six groups initially (seven later), more than 50 postgraduate students, and tens of research and development personnel. The project was split into:

- 1) a “front-end” under the University of Southern California with the participation of Northeastern University and Yokohama National University for all high-level synthesis, placement, routing, clocking, timing verification, and other system-level design tools;
- 2) a “back-end” under Stellenbosch University, for all physical level tools from TCAD (performed by the University of Florida) and Josephson junction (JJ) modeling (performed

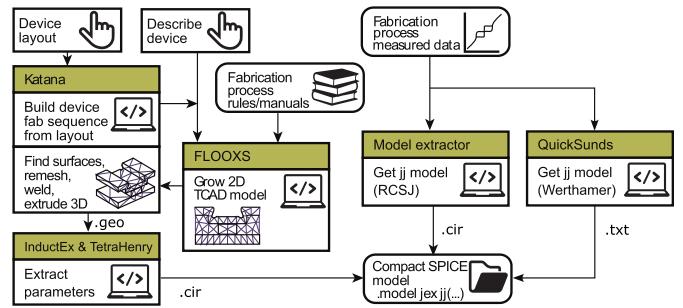


Fig. 2. ColdFlux tool flow diagram at the device (TCAD) level.

by the University of Savoie Mont Blanc), as illustrated in Fig. 2, to electrical simulation, cell characterization and optimization, layout extraction, layout-versus-schematic (LVS) verification, parameterized cell layout, and compact simulation model extraction, as illustrated in Fig. 3;

- 3) cell library development by Stellenbosch University [rapid single flux quantum (RSFQ)], Yokohama National University, and Tokyo City University [adiabatic quantum flux parametron (AQFP)].

Nearly all tools and libraries are available upon request at the ColdFlux repository [6] as finalized within the SuperTools program. Further updates may be found at appropriate GitHub repositories referenced appropriately in the subsequent text.

B. Test and Evaluation

A team at the National Institute of Standards and Technology (NIST) tested chips fabricated under ColdFlux to find RSFQ cell operating margins, verified transmission over different lengths of passive transmission line interconnect, and gathered large datasets to verify flux trapping analysis and compact simulation model fidelity.

A team at the Lawrence Berkeley National Laboratory (LBNL) produced high-quality artifacts to test ColdFlux tools: an RISC-V RV32 (Sodor) core and memory, an RISC-V RV64 (Rocket) core, and a multithreaded/time-skewed parameterized RISC-V core.

Teams at NIST and MIT Lincoln Laboratory (MITLL) tested circuit simulation and parameter extraction tools. MITLL also provided data on flux trapping events in relation to moat placement.

III. DEVELOPMENT DECISIONS AND CODE PRACTICE

A. Development Decisions

To meet budget and time constraints and accommodate the ColdFlux team’s academic nature, some decisions were taken at the start of the project. ColdFlux results are predominantly open-source, so many of these design decisions were influenced by the availability of open-source modules.

- 1) Development would have a strong research component to allow postgraduate student development and to uncover the answers to open questions, such as how to model trapped fluxons in a circuit simulation.

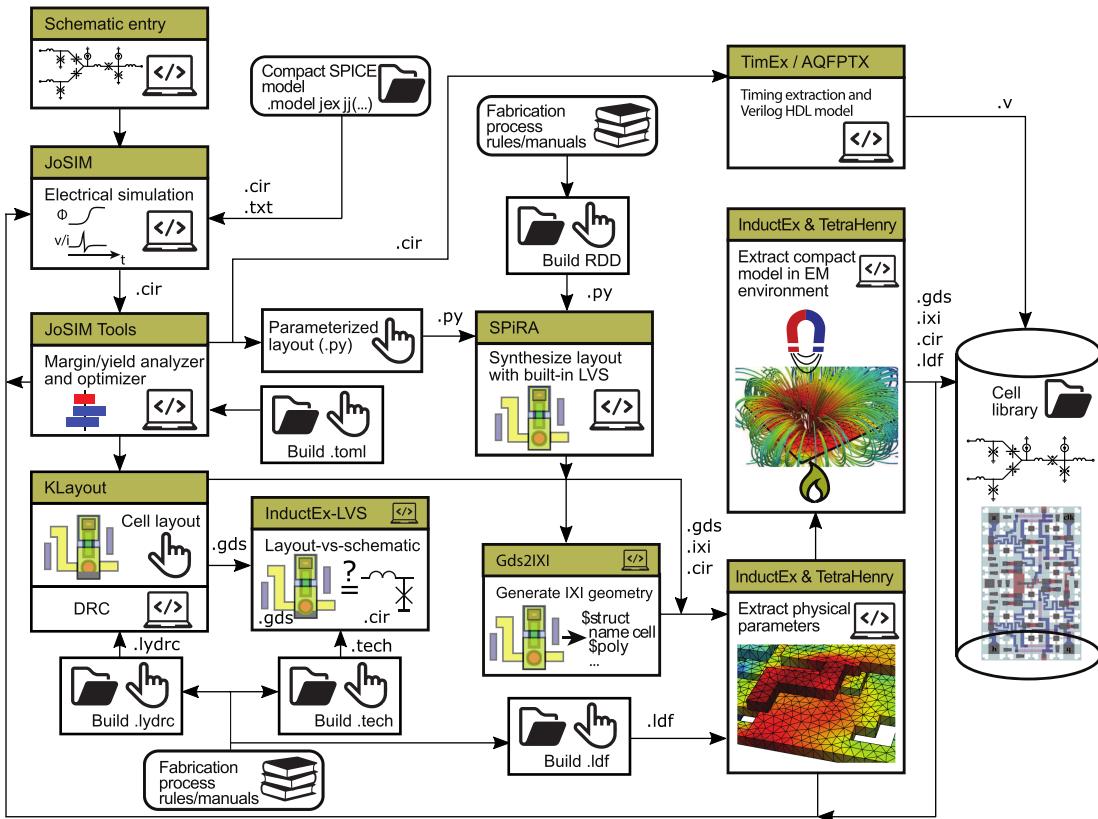


Fig. 3. ColdFlux tool flow diagram at the circuit (physical) level.

- 2) The cell libraries would be limited to RSFQ as a dc-biased logic family and AQFP as an ac-biased logic family.
- 3) Hardware description language (HDL) modeling would be in Verilog.
- 4) Cells would be developed for row-based placement [7], with a “standard” cell that supports direct connection to an abutted cell through inductors, complemented by a “PTL” cell with integrated passive transmission line (PTL) drivers and receivers to enable routing over large distances for automated routing.

It was evident from the results of a preceding seedling project [7] that ColdFlux could not simply adapt tools for semiconductor IC design, but that many new concepts would have to be researched, developed, and evaluated.

A more detailed description of the design decisions and the uncertainties facing the development team at the start of the project is given in [8].

B. Code Practice

Management of a project to develop software modules over four continents in an academic environment, where postgraduate students commit to two or three years of research and development *aligned with a tangible output in terms of a postgraduate thesis*, is complex.

We opted for a strategy where tools would be developed as stand-alone modules, with most tool modules reduced in function to enable one or two developers to develop, code, debug, and maintain a module. At the physical level, modules

are all compiled as binary files that take files and command line parameters as inputs and write outputs to files. In this way, interfacing modules with each other reduced the complexity of correctly translating input/output files.

Open-source modules are available on platforms such as GitHub, with user manuals and examples supplied for all modules. Internal development notes and reports were kept to document tool development decisions.

Most proof-of-concept modules were developed in Python due to wide familiarity with the amongst postgraduate students and the relative ease of implementation. At the physical level, all the modules that are resource intensive were coded in C++ or Pascal.

C. Platforms

The SuperTools program mandated support for the CentOS 7 Linux operating system, so that a modular approach worked well. CentOS 7 and the Linux Kernel 3.10 aged badly toward the end of the program, which complicated tool and library maintenance. The tools work well when compiled on Red Hat Enterprise Linux 8 and 9 or any recent Ubuntu Linux release. It is recommended that open-source tools are compiled from source under the most recent version of Ubuntu Linux (22.10 at the time of writing).

The commercial tools developed under ColdFlux are precompiled for Linux (from Kernel 3 and up), Windows 10 and up, and macOS (11 and up, at the time of writing).

D. Execution

At the start of the project, tools and cell libraries were developed in parallel, so that cells were designed with archaic tools or by hand.

As new tools were phased into the ColdFlux design chain, they were immediately applied to cell library development to allow intragroup feedback and debugging. Toward the end of the project, all the tools outside of the project were removed from the cell design process, with the exception of the layout tool KLayout [9].

Fabrication runs provided by SuperTools were used to:

- 1) Verify library cell operation and compare against tool predictions;
- 2) Test extraction precision and improve modeling fidelity and tool calibration;
- 3) Verify flux trapping analysis assumptions and methods.

The long turnaround time for fabrication runs, along with tight deadlines for the various phases, few chip slots, and a project stipulation that fabrication should be used to verify device parameters rather than large-scale digital circuits, meant that the completed toolchain could not be applied to a full digital circuit and fabrication cycle. Even so, many digital logic cells, initially designed without the benefit of the ColdFlux toolchain, were included on various chips through the course of the project. Operating margins and other measured parameters provided feedback for future designs and tool development. *Without the benefit of data from the fabrication runs, the ColdFlux toolchain would have been considerably less capable.*

IV. REGISTER-TRANSFER-LEVEL (RTL) SYNTHESIS, ARCHITECTURES, AND VERIFICATION

A. qPALACE Tool Suite Integration and Process Design Kit (PDK)

qPALACE (Physical and Logical Aware Compiler Engine for single-flux-quantum logic [10]) is a tool suite that receives a high-level design in Verilog HDL or Berkeley Logic Interchange Format (BLIF) formats and maps it to a single-flux-quantum (SFQ) chip. The mapping process is composed of different steps, e.g., synthesis (behavioral and logic synthesis), followed by technology mapping, placement (global and detailed), clock tree synthesis, and routing. In addition, other processes are required within the design flow, e.g., timing characterization, post placement and routing (P&R) timing analysis, design for test (DFT), automatic test pattern generation (ATPG), power analysis, and format converters. Fig. 4 shows the flow of the qPALACE Tool Suite.

The design tools can be implemented independently of each other, each referred to as *qTools* in this document. The functionality of qPALACE is to connect and incorporate different qTools within a flow. Therefore, given an SFQ technology, qPALACE enables the designer to have a simple and reliable process of mapping an input description to a final SFQ chip and generating the required reports, while reserving access to the configuration inputs and generated outputs of each qTool. The list of qTools

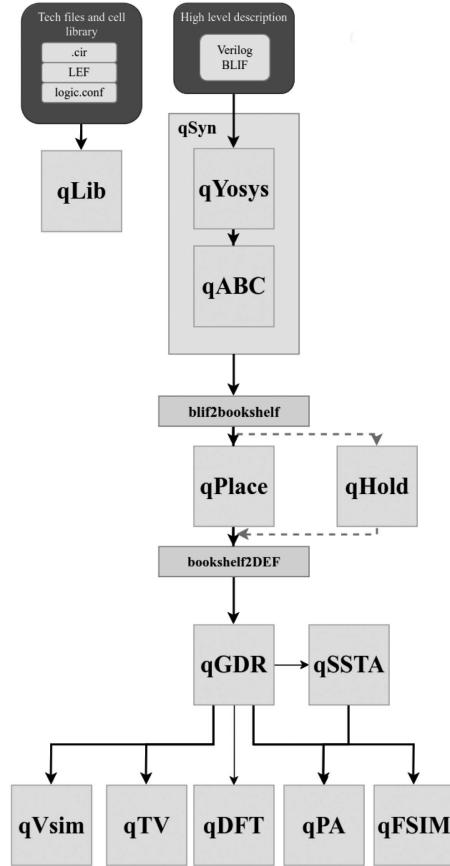


Fig. 4. ColdFlux tool flow diagram at system level.

TABLE I
NAME AND DESCRIPTION OF EACH QTOOL IN THE QPALACE TOOL SUITE

qTool	Description
qLib	Generates the required formats from the input technology library provided by the manufacturer through translation and simulation
qYosys	Parses high-level Verilog/BLIF descriptions and behavioral synthesis
qABC converters	SFQ specific logic synthesis, mapping and verification blif2bookshelf: converts the BLIF format to the bookshelf, and bookshelf2def: converting the bookshelf format to the Design Exchange Format (DEF).
qPlace	Placement and clock-tree synthesis
qGDR	Global and detailed routing
qSSTA	Statistical static timing analysis
qVSim	Post-routing simulation
qHold	Fixes hold time violations considering process variation
qTV	Timing validation for post routing netlists
qPA	Power analysis
qFSIM	Fault simulation and test pattern generation
qDFT	Fast fault simulation for BIST performance evaluation

in the final version of qPALACE released under ColdFlux, with short descriptions, is provided in Table I.

The qPALACE Tool Suite is accompanied by a PDK [6], to provide the most up-to-date information about installation, and an extensive user manual for each of the qTools. Moreover, it includes the relation of the qTools to the overall design process

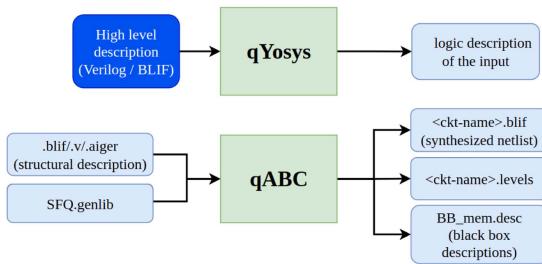


Fig. 5. qSyn tool includes several subtools such as qYosys and qABC.

and in each section the description of a set of process-related configuration files and examples of how to use each within the appropriate tools is provided.

B. High-Level and RTL Synthesis

1) *Rapid Single Flux Quantum*: Synthesis and translation of the high-level architecture definition to the gate level are done by qSyn. It consists of two main tools: qYosys and qABC. The qYosys tool is a framework for RSFQ Verilog RTL synthesis. qABC is responsible for gate-level translation and path balancing. The flow for the qSyn tool with its inputs and outputs and associated format is shown in Fig. 5.

The first tool developed for synthesis was SFQmap [11]. This novel technology mapping tool provides optimization methods to minimize the circuit depth and path balancing overhead first and then minimize the worst case stage delay of mapped SFQ circuits. Compared with the state-of-the-art technology mappers, the SFQmap reduces the depth and path balancing overhead by an average of 14% and 31%, respectively.

We then developed a path balancing technology mapping algorithm [12], a new algorithm for generating a mapping solution for a given Boolean network such that the average logic level difference among the inputs of each gate in the network is minimized. Path balancing technology mapping is required in SFQ circuits to guarantee the correctness of the operation, and is beneficial in complementary metal–oxide–semiconductor (CMOS) circuits to reduce hazard issues. We developed an algorithm for path balancing technology mapping based on dynamic programming that generates optimal solutions for RSFQ circuits with tree structure, and acts as an effective heuristic for circuits with general directed acyclic graph structure. Experimental results show that our path balancing technology mapper reduces the balancing overhead by up to 2.7 times and with an average of 21% compared to the state-of-the-art academic technology mappers.

Next, we developed balanced factorization and rewriting algorithms to reduce the path balancing overhead [13]. Experimental results on a set of 15 benchmark circuits show that a combination of balanced factorization and rewriting algorithms reduces the path balancing overhead by an average of 63% and area by up to 23% compared to state-of-the-art logic synthesis tools. Solutions were improved by designing a new dynamic programming-based algorithm for technology mapping, where our proposed method decreased the total area, static power consumption, and path balancing overhead of SFQ circuits by large factors [14]. Experimental results showed that this algorithm can reduce the

circuit area by up to 111% and by 26.3% on average when compared to state-of-the-art technology mappers. The synthesis output for some of our larger test benches is demonstrated in Table II.

The first column of Table II shows the name of the test bench. The definition file is parsed with qYOSYS, and then qABC uses the defined technology library to generate a cell level logic with the total number of logic cells shown in the second column. All our SFQ logic cells are clocked; therefore, a clock tree branch is needed for each logic cell. D flip-flops (DFFs) are used for path balancing to ensure that inputs come at the right clock cycle to the clocked gates. The number of DFFs is shown in the third column. The numbers of pulse splitters for the data and clock paths are shown in the fifth and sixth columns. The seventh column shows the number of JJs needed for circuit design, and the last column shows the circuit depth in logic levels. While not affecting the throughput, the circuit depth is important for the latency. If we assume that the number of logic cells is N , the number of path balancing DFFs is P , and the circuit depth is D , the following can be concluded by curve fitting.

- 1) Sequential: $P = 188.6 + 2.812 \times N - 45.82 \times D$
- 2) Combinational: $P = 132.4 + 4 \times N - 60 \times D - 0.289 \times N \times D + 4.63 \times D^2$
- 3) Clock_Path_Splitters = $N + P - 1$.

It is evident that as depth increases, the number of path balancing DFFs drops at the cost of higher latency.

2) *Adiabatic Quantum Flux Parametron*: Earlier work on developing an AQFP RTL-to-GDS top-down flow to assist in the implementation of adiabatic microprocessors is described in [15]. It is based on a customized Cadence environment using a four-phase AQFP cell library implemented in the Japanese high-speed standard process available at the National Institute of Advanced Industrial Science and Technology [16]. The design environment is composed of HDL gate-level models with dynamic timing verification [17], rudimentary combinational logic synthesis flow based on Yosys [18], and a physical synthesis flow using a genetic-algorithm-based place and route [19], [20]. The largest successfully operating circuit using a mix of both manual design and the aforementioned tools was a 4-bit adiabatic superconductor microprocessor comprising over 20k JJs with execution components operating up to 2.5 GHz [21]. To move beyond this scale and achieve a more automated flow, improvement to the design environment was necessary.

In the SuperTools program, the logic synthesis flow has been improved to take advantage of majority logic and the optimization of buffer and splitter insertion [22], [23]. This was a key improvement as AQFP logic natively operates on majority logic primitives, and it has been shown that circuits expressed in majority logic can yield better quality of results, particularly for arithmetic circuits [24]. Majority optimization alone has resulted in up to 60% improvement in circuit area and delay in many benchmarks including the ISCAS'85 benchmarks.

Furthermore, the genetic-algorithm-based place and route flows developed before SuperTools were generally extremely slow and had very limited circuit scale capacity. This has been much improved using more analytical approaches in [25] and [26] as part of the SuperTools program. Circuit benchmarks that

TABLE II
QPALACE GATE-LEVEL SYNTHESIS RESULTS

Circuit Name	Logic Cells	DFFs	Latches	Logic Splitters	Splitters	Total Cells	JJs	Max Depth
KSA-32bit	526	455	0	489	980	2450	18 278	11
CGRA2Core	165 666	6768	53 787	231 675	517 138	3 611 307	24	
Counter128	1171	2350	128	1133	3648	8430	59 925	13
Integer_divider8	3679	4137	0	3221	7815	18 852	136 923	41
EPFL_Log2	62 995	230 818	0 53 617	293 812	641 242	4 391 301	212	
EPFL_Mult	78 827	136 587	0	66 256	215 413	497 083	3 496 507	99
EPFL_Divider	148 230	671 999	0	125 276	820 228	1 765 733	12 027 387	631
C6288	28 993	32 981	0	25 136	87 109	149 083	1 079 909	64
C7552	1822	3179	0	1433	5 000	11 444	80 922	21
Mult-16bit	2936	4556	0	2445	7 491	17428	122 811	39
RISCV-32bit ¹	126 931	253 406	9 644	114 308	389 980	894 269	6 341 284	25
RISCV-32bit ²	57 919	179 826	2 732	49 678	240 476	530 631	3 669 260	52
RISCV-32bit-Sodor ³	49 515	108 950	2 503	41 558	159 825	361 209	2 537 923	35
RISCV-64bit-Rocket ⁴	145 707	463 234	4 562	125 698	613 502	1 352 703	9 338 053	62

¹Synthesized for depth efficiency from Py RTL python library with memory masked (black boxed).

²Synthesized for area efficiency from Py RTL python library with memory masked (black boxed).

³The 32-bit Sodor core RISCV CPU.

⁴The 64-bit Rocket core RISCV CPU.

would typically need more than a day to complete are now able to converge to a better quality of results in a matter of a few to several minutes. The timing-aware placement done at both the global level and the detailed level by taking into account the clock skew accumulated along the meandering power-clock networks has been shown to increase the maximum operating frequencies by up to 40% with a negligible increase in wirelength (1%) [26].

Finally, the AQFP synthesis approaches before SuperTools only considered the realization of combinational logic. In the SuperTools program, we developed a sequential logic synthesis methodology based on the development of the quantum-flux-parametron latch (QFPL), which can be used as a nondestructive-read-out (NDRO) to serve as the architectural state registers of any given sequential logic circuit from finite-state machines to pipelined datapaths [27]. This methodology has been successfully applied to N -bit counters (up to 32-bit) as well as a 16-bit MIPS microprocessor and has been integrated into the ColdFlux suite of AQFP top-down tools. It has also revealed further directions in buffer reduction by as much as 75% by taking advantage of the operating principles of the NDRO.

C. RTL Simulation and Verification

Design verification is a critical part of the chip design process. The main purpose of design verification is to ensure that the final design meets all the functional and performance requirements as stated in the system specifications. CMOS technologies have a robust set of tools and technologies for the verification of CMOS circuits. Before SuperTools, the SFQ design process lacked the verification frameworks necessary for it to grow into a viable alternative to CMOS technology [28], [29], [30], [31].

Under the ColdFlux project, we developed and delivered a suite of verification frameworks for SFQ technology, namely, qEC, qMC, VeriSFQ, and qVSIM. These frameworks rely on simulation, formal, and semiformal verification methodologies.

More precisely, qEC is a logical equivalence checking (LEC) framework for SCE [28], [32]. Our LEC framework is compatible with existing CMOS technologies as well as able to check unique features of SCE.

qMC is based on model checking (MC) using formal properties defined for SFQ design functionality [33]. qMC constructs a SystemVerilog test bench using formal assertions to verify the SFQ-specific properties of a circuit. It then produces system correctness results using MC. qMC is built on top of the already established back-end verification engines for MC of CMOS circuits, namely, Yosys-SMTBMC and EBMC.

VeriSFQ is a semiformal verification framework for SFQ circuits based on the Universal Verification Methodology (UVM) [29]. VeriSFQ is a tool for SFQ logic circuit and gate-level characteristics such as fan-out, path balancing, gate-level pipeline, and input-to-output delay verification. Following that, we introduced qCG, our machine-learning-based UVM-compliant coverage-directed test generation verification engine. qCG's verification engine learns to improve the quality of results by reducing verification time and increasing coverage. To ensure the pulse integrity of clock signals, SFQ fan-out, and path balancing features, embedded datapath and coverage meters are integrated into qCG.

Finally, qVSIM is a simulation-based post-layout verification framework utilizing a test pattern generation engine. It creates valid test benches by sampling millions of points drawn from a large valid space generated from a satisfiability modulo theory formula. qVSIM estimates a suitable frequency for circuit operation by deploying a static timing analyzer and produces golden results for the evaluation of circuit operation.

D. Datapath Synthesis and Architectural Optimization

We designed the register file, HiPerRF [34], based on the High-Capacity Destructive ReadOut (HC-DRO) cells [35]. While using the DRO-based design, we still keep the nondestructive property of the register file. We also implemented

an RISC-V simulator to evaluate our design under the microarchitectural scope while considering the gate-level pipeline. Although HC-DRO has a different data representation, HiPerRF has corresponding translation circuits so that it can be used in any traditional CPU design without modification. HiPerRF reduced the JJ count by 56.1% and the static power consumption by 46.2% compared with an NDRO-based register file design.

V. ANALOG DESIGN AND LAYOUT SYNTHESIS

A. Electrical Simulation Engine

1) *Existing Simulation Engines*: Electrical simulation engines were arguably the most mature tools for SCE circuit analysis at the start of SuperTools. Where standard SPICE (Simulation Program with Integrated Circuit Emphasis) engines lack support for the JJ, a number of JJ-capable simulators exist. Some of these, such as COMPASS [36], are not available anymore. PSCAN [37], [38] has been available since 1991 and uses a modified nodal phase method. It supports the microscopic tunneling model as one option for JJ simulation. PSCAN only supported inductive coupling in two-inductor transformers and could, thus, not model inductors coupled to multiple other inductors—an essential requirement for the analysis of circuits such as AQFP gates. This shortcoming was addressed when PSCAN was rewritten in Python and released as open-source software PSCAN2 [39]. Although PSCAN2 is a powerful simulation engine that runs more than an order of magnitude faster than PSCAN, it is not widely used, probably because of the lack of user manuals or example sets, and it uses dimensionless units, which slightly complicates the mapping of parameter values.

The most popular electrical simulators for SCE electronics have intrinsic support for the resistively and capacitively shunted junction (RCSJ) model of the JJ [40], [41]. JSPICE3 [42] and its direct successor, WRspice [43], use an intrinsic RCSJ model. WRspice supports a standard piecewise-linear model, an analytic exponentially derived approximation, and a fifth-order polynomial expansion model for quasi-particle resistance. WRspice was a commercial simulator but was released as open-source software when its creator, Dr. Stephen Whiteley, joined Synopsys under the SuperTools program where he helped transfer JJ model support to the powerful Hspice simulation engine.

The Josephson integrated circuit simulator (JSIM) [44] is a lightweight voltage-based simulator for both analog and digital SCE simulation. It was designed to operate on systems without large random access memory, which limits efficiency. It is also limited to passive circuit elements and the JJ (through to the RCSJ model) with a piecewise-linear quasi-particle resistance. A modified version that includes limited thermal noise analysis support was released as JSIM_n [45], but requires a script running under Linux to convert a noiseless simulation deck into one with Johnson (thermal) noise included.

2) *Josephson Simulator (JoSIM)*: The JoSIM [46], [47] was conceived under the ColdFlux project as a simulation engine that would exploit modern coding methods for improved speed and

larger circuit support (with an initial aim of one million circuit components) than existing superconductor circuit simulators.

At its core, the JoSIM is set apart from other simulators by the provision for two analysis modes for the solution of linear circuit equations: a modified nodal voltage analysis mode, such as that used in traditional SPICE engines, and a modified nodal phase analysis mode. The two modes require different modified nodal analysis stamps.

The JoSIM includes intrinsic support for noise and has an application programming interface, through which tool modules such as margin analyzers and optimizers can access JoSIM functionality.

With a C++ implementation, the JoSIM runs on Windows, Linux, and macOS and exceeds the SuperTools goal for simulation size by an order of magnitude: at the time of writing, it easily handles 100 million components—the typical component count for a circuit with 10 million JJs—on a desktop computer with 128 GB of RAM.

Finally, the support of phase sources in phase mode allows a circuit designer to model flux trapping in compact simulation models of SCE circuits, *something that was not possible with any known tools at the start of the SuperTools program*.

We have used JoSIM to do bit-error rate analysis [48], while SuperTools Test and Evaluation teams have used the JoSIM to analyze SQUID arrays [49].

The final JoSIM version produced for the SuperTools program is available on GitHub [50].

B. Layout Synthesis

Layout synthesis methods were developed to define circuit layouts as scripts with the aid of Python-based parameterized cells, with LVS verification baked into the synthesis process. The tool SPIRA [51], which is available as open-source software [52], was developed in Python. For version 2.1 of the ColdFlux RSFQ cell library, every cell was described as a Python script for SPIRA so that it could be synthesized directly from the script for a given set of layout parameters without the need to ship a layout artwork file such as GDS.

Layout synthesis became less important as a stronger LVS tool, InductEx-LVS, matured toward the end of ColdFlux.

C. JOINUS: JOsephson INterface Utility Software

We developed JOINUS [53] as a graphic user interface to integrate the simulation engines and the layout synthesis environment. JOINUS can read circuit netlists, simulate them with several engines, add noise, draw I - V curves, calculate bit error rate for bias, temperature, and frequency and embed different add-ons. It was designed to embed some software that are needed to perform SFQ simulations in a single tool, mostly for beginners with circuits of limited complexity in a first step. JOINUS integrates a netlist editor with highlighting features and syntax analysis to recognize a range of usual errors. It has its own margin and yield simulation routines that rely on algorithms calling engines like JoSIM to assess when digital circuits work or not. It can also automatically add thermal noise to resistive elements for simulations. It has its own integrated plotting tool

that can show the time-domain, frequency-domain, and margin analysis. It is also possible to call InductEx [54], [55], [56], [57] to extract netlist parameters once the routine loads the corresponding layout and technology files and to back-annotate netlists once extraction is done and edit the GDS file by calling KLayout software. By integrating all the most used tools during a design sequence, JOINUS simplifies the learning curve for new entrants into the field of superconductor digital electronics.

D. Process Design Kit

A PDK was developed that contains the ColdFlux RSFQ and AQFP cell libraries and the setup files for all the tools that operate on the MITLL SFQ5ee process.

A comprehensive single volume document [6] was compiled (and kept updated) to form part of the PDK. It serves as a manual to guide the design and operation of the ColdFlux PDK for the MITLL SFQ5ee process. The manual is divided into sections that describe each tool within the overall PDK superstructure. It provides the current information about these tools and their relation to the overall design process. In addition, this manual provides in each section a set of process-related configuration files and examples of how to use each within the appropriate toolset.

The tools included are InductEx (and its engines), JoSIM, WRSpice, and Xic (used in the design chain before JoSIM and the KLayout environment were ready), Adiabatic Quantum-Flux-Parametron Timing eXtraction (AQFPTX), qPALACE, SPiRA, qIDE, qSynthesizer, and FLOOXS. Each section begins with an introduction section, configuration files, prerequisite libraries or packages, and program execution instructions. Some sections include examples of command line instructions, synthesized and mapped circuits, and waveforms for correct functionality of the circuits.

VI. PHYSICAL DESIGN AND TEST

A. Place-and-Route and Clock Tree Synthesis and Routing

Conventional place-and-route algorithms and tools matured over the years for CMOS technology cannot be deployed directly for SFQ circuits. Besides general similarities between the placement process for CMOS and SFQ technologies, new tools are required that use algorithms that take into account specific constraints of superconductor digital circuits [58]. We developed such tools under ColdFlux.

To target splitter delays, we first designed a novel clock tree synthesis algorithm that results in a fully balanced clock tree structure, i.e., a placement solution with an identical number of clock splitters from the clock source to all the sink nodes. Moreover, overlaps among the clock splitters and placement blockages are removed by deploying a mixed-integer-linear-programming-based algorithm that minimizes the clock skew simultaneously. To address the imbalanced topologies, a new version of the clock tree synthesis algorithm was presented that reduces the clock skew and the number of clock splitters in the clock network by 56% and 37%, respectively, compared with a fully balanced clock tree solution [58].

We extended the clock tree synthesis algorithm and developed a low-cost timing-uncertainty-aware synchronous clock tree topology generation algorithm for SFQ logic circuits [59]. The proposed method considers the criticality of the datapaths in terms of timing slacks as well as the total wirelength of the clock tree and generates a (height) balanced binary clock tree using a bottom-up approach and an integer linear programming (ILP) formulation. The statistical timing analysis results for ten benchmark circuits show that the proposed method improves, on average, the total wirelength and the total negative hold slack by 4.2% and 64.6%, respectively, compared with a state-of-the-art wirelength-driven balanced topology generation approach.

We developed a novel clustering-based placement algorithm for the SFQ logic circuits [7], [60], following a balanced topology generation approach. In these circuits, nearly all cells receive a clock signal, and a placement algorithm that ignores the clock routing cost will not produce high-quality solutions. Our approach addresses this issue by minimizing the signal nets' total wirelength and the clock routing area overhead simultaneously. Furthermore, constructing a perfect H-tree in SFQ logic circuits is not a viable solution due to the resulting very high routing overhead and the infeasibility of building exact zero-skew clock routing trees. Instead, a hybrid clock tree must be used whereby higher levels of the clock tree (i.e., those closer to the clock source) are based on H-tree construction, whereas lower levels of the clock tree follow a linear (i.e., chain-like) structure. Our approach can reduce the overall half-perimeter wirelength by 15% and area by 8% compared with state-of-the-art techniques.

We also presented a dual-clock architecture for realizing RSFQ circuits that removes all path balancing DFFs, resulting in a huge reduction in total area, node and JJ count, and power consumption. This type of architecture is called dual clock architecture since it requires two types of clock. One is called the slow clock. It resets the state of the control cells. The other is the fast clock for cell operation. We insert two levels of cells in the original circuit as control cells: one level (called the repeat band) uses NDRO cells to capture the input signal every slow clock and to repeat that signal during the slow clock period; the other level (called mask band) uses AND cells to capture the output signal when the slow clock signal arrives. By repeating the input signal, we can remove all the path balancing DFFs in the circuit. The drawback is a degradation of the peak throughput of the circuit, which can be overcome by performing partial path balancing in the circuit [61]. The P&R result of qPALACE is shown in Table III.

Furthermore, we extended the dual clock method in the synthesis of sequential circuits. Except for the primary inputs, we also insert mask and repeat bands in the state output and input of the sequential loop. We also designed another architecture called multithreading architecture to capture the input signal and release the output signal in parallel. It contains a scheduler to apply parallel input signals into the circuit in serial and another scheduler to capture output sequences and release them in parallel [62].

Furthermore, we developed TDP-ADMM [63], a novel timing-driven global placement approach utilizing the

TABLE III
QPALACE P&R RESULTS

Circuit Name	Number of Logic Cells	Number of Splitters	Circuit Depth	Clock Period (ps)	Power Consumption (mW)	Area (mm ²)
C432	1395	2644	18	160	15.0	48.6
C499	586	1262	10	120	6.7	22.2
Counter64	401	2448	23	163	14.5	50.1
Divider4	479	754	11	103	4.8	17.8
Divider8	7832	11 412	41	252.9	73.5	245.7
KSA32	981	1512	11	148	9.7	37.6
Mult8	1438	2485	39	115	14.3	47.5
S526	451	697	26	81	4.4	15.7
Wallace_tree	17 074	41 790	41	377	213.5	550.4
Mult_32bits						

alternating direction method of multipliers (ADMM) targeting SCE circuits. TDP-ADMM models the placement problem as an optimization problem with constraints on the maximum wirelength delay of timing-critical paths and employs the ADMM algorithm to decompose the problem into two subproblems: one minimizing the total wirelength of the circuit and the other minimizing the delay of timing-critical paths of the circuit. An iterative process generates a placement solution that simultaneously minimizes the total wirelength and satisfies the setup time constraints. Compared to a state-of-the-art academic global placement tool, TDP-ADMM improves the worst and total negative slack for seven SFQ benchmark circuits by an average of 26% and 44%, respectively, with an average overhead of 1.98% in terms of total wirelength.

1) *Advanced Clocking*: We also developed an advanced multiphase clocking methodology for reducing the number of required path balancing buffers, while enabling multithreaded computation [64]. Gate-level clock-phase assignments can exploit the differing arrival times of different clock phases to remove the necessity of some path balancing buffers in designs. Our developed ILP searches for the optimal clock phase assignment to each gate for a given number of clock phases. Compared to fully path balanced approaches, our method on average reduces path balancing buffer insertion by 55.5% for two clock phases and up to 95.5% for ten clock phases. Post clock tree synthesis and place-and-route results show that the decrease in registers yields a decrease in total gate area by 40.6% and clock tree wirelength by 54.9% with two clock phases, and by 69.6% and 69.8% with ten clock phases, respectively. In addition to having lower overhead, a key benefit of our approach is that it requires no fast clock. In particular, the clock frequency of the proposed multiphased clocks is the same as the throughput of the circuit, avoiding the need to synthesize and route a high-speed clock.

2) *Clocking AQFP*: AQFP circuits are generally clocked by a four-phase clock network distributed in a meandering structure comprising two ac lines and a dc line [16]. The two ac lines provide two sinusoidal excitation currents in quadrature, and when combined with the dc offset provided from the dc line, four distinct clock phases are generated to provide a power clock to all AQFP logic cells. One of the key limitations of this clocking approach is that data propagation is limited to only four levels of logic per clock cycle (one logic gate per phase). This ultimately

results in very large latencies in terms of clock cycles for deep levels of logic. To overcome this limitation, recent efforts on low-latency clocking methods have been developed, namely, delay-line clocking [65] and power-dividing clocking [66]. Both approaches exploit the fact that the AQFP is intrinsically a relatively fast switching primitive with a propagation delay on the order of a few picoseconds, which is very short compared to the much longer excitation clock phase period of 50 ps in conventional four-phase clocking at 5 GHz. In low-latency clocking, the effective clock phases operate closer to the intrinsic delay of the AQFP. These shorter clock phases are generated through microwave delay lines inserted between logic rows [65] or provided directly from on-chip power dividers [66]. In principle, the AQFP circuits using these approaches still benefit from the same extremely low-energy switching dissipation because the operating frequency still remains the same.

Reduction of buffering by the aforementioned low-latency clocking approaches was also investigated [67]. Because these approaches effectively create many (more than four) phases per clock cycle, for a given data launching clock phase, there are a number of suitable data capturing clock phases. It is, thus, not necessary to buffer data along every phase. We investigated latency and buffer reduction for a number of benchmark circuits including adders, multipliers, decoders, and shifter/rotators of varying data word sizes up to 64-bit. We compared the latency, buffer usage, and JJ usage for these benchmarks for clock networks with 4, 8, 12, 16, and 20 phases. When the number of clock phases in a single cycle increases x times, we saw that both the latency and buffer usage decrease by about x times in sufficiently large circuits.

Moving forward, it would be necessary to conduct a detailed timing analysis combined with accurate modeling of data signal propagation. Such a timing analysis would allow designers to determine what is the maximum number of clock phases per cycle that can be applied to a given circuit. In addition, more investigation is needed to physically realize the low-latency clocking approaches at chip scale. Initial work on a global microwave H-tree clock distribution network with delay-line clocking applied to the local circuits has been done in [68], but has yet to be shown operating for large circuits.

Recent progress in routing optimization has also been made outside of SuperTools [69].

B. Power Analysis Tool

Under ColdFlux, we developed a simulation-based power analysis tool that caters to both RSFQ and AQFP circuits. The tool's workflow commences with a cell power characterization stage, wherein we measure the power behavior of each logic cell, buffer, and splitter under various applied input patterns using JoSIM, our circuit simulator of choice. The resulting power data are then stored in a comprehensive power characterization table, encompassing both dynamic and static power consumption values for each input pattern. Next, for a given SFQ circuit, we estimate the circuit's power consumption utilizing a Monte Carlo method, as proposed in [70]. In each iteration, we generate a random primary input pattern and perform logic simulations to derive corresponding input patterns for each cell within the circuit. By referring to the power characterization table, we can determine the power consumption value for each cell and sum up the individual power consumption of all cells in the netlist, including logic cells, buffers, and splitters. The power consumption data for the circuit in this iteration is collected in a list. For subsequent iterations, we generate new test patterns, updating the power consumption dataset. During these iterations, we keep track of the mean and standard deviation of the circuit's power consumption based on the applied input patterns. The iteration process continues until the power consumption dataset converges, signifying that further iterations are no longer necessary. Once convergence is achieved, the process terminates, and we report the circuit's power consumption as the final result. This simulation-based power analysis tool provides an efficient and reliable means to estimate the power consumption of both RSFQ and AQFP circuits, significantly enhancing our understanding and enabling the optimization of SFQ circuits' energy efficiency.

C. Fault Simulation, Test Pattern Generation, and Built-in Self-Test (BIST) Tools

We developed a new clean-slate method to derive fault models from many simulation results [71]. We first select each logic cell in the given cell library—AND, XOR, DFF, INV (NOT), and OR—as the cell under study (CUS). For each CUS, we create many netlists by using each logic cell in the library as the driver of a CUS input and each logic cell in the library as the load on the CUS output. Furthermore, for each netlist obtained above, which we call a CUT, we also consider another version where we insert a splitter between the driver cell and the CUS. Since we worked with five logic cells, this approach creates a total of 250 CUT netlists.

For each CUT netlist described above, Monte Carlo sampling is performed to apply process variations to the critical current of JJ, resistance, and inductance, to obtain many versions of each CUT. This approach yields a total of 100 000 versions of CUT netlists. Each CUT is then simulated for a comprehensive set of patterns to identify the netlists where one or more logic errors are observed at the CUS output.

We developed and used Inductive Fault Model Extraction, an inductive method that analyzes a small number of failing CUTs to analyze the logic errors and identify the root cause. The method identifies all other CUT netlists that exhibit identical

combinations of logic errors and root causes and catalog the associate failure as a fault type. For RSFQ cells, our method catalogs fault models for more than 99% of failing cases and develops completely new fault models—overflow, pulse-escape, and pattern-sensitive—in addition to the more usual stuck-at faults [71].

Cells are then characterized under process variations to identify delay excitation conditions, sensitization conditions, and conditions for propagation of the logic errors caused by process variations. We addressed several radically new phenomena in RSFQ technology, especially the existence of single-pattern delay tests and the need to propagate delayed values via multiple pipeline stages [72], [73]. For this, we developed a completely new ATPG paradigm that utilizes these new phenomena to select target delay subpaths and generate test patterns that are guaranteed to excite the worst case delay along each target delay subpath.

We developed a new timing analysis method that allows larger increases in clock-to-Q delay, i.e., timing bleed, whenever the data input arrives late [74], [75]. Conventional setup time was defined as the point where the increment in clock-to-Q delay is lower than 10%, so that there is low probability that a logic error happens in the node under test. We also defined the soft setup time as the point where the cell starts to experience a larger than normal clock-to-Q delay, while hard setup time was defined as the point where the cell generates a logic error and the circuit has a high probability of failure. Taking timing bleed into consideration, we developed a method for selecting path delay faults by identifying the subset of paths for which the delay can exceed the clock period under the main cause of delay faults for RSFQ, namely, extreme process variations. We showed that this dramatically reduces the number of delay tests required due to the characteristics of gate-level pipelined design, a necessary requirement for RSFQ. We also extended our method to be the first ATPG to generate tests for RSFQ-specific static fault models [76]. Experimental results show that we can detect more than 98% of the faults with less than 100 patterns [76].

Based on the fault model we developed above, we developed our BIST/DFT method that enables high-quality testing of RSFQ logic. We first identified the test requirements for RSFQ logic by analyzing every aspect of RSFQ testing, especially by identifying the barriers to achieving high coverage of static and delay faults, challenges in terms of special test application requirements, causes of high test data volume (which impacts test time and cost), and the limitations of the external test equipment. In particular, we showed that the key requirements of DFT for RSFQ pertain to interfacing with much slower external test equipment and the achievement of high fault coverage for logic blocks with feedback as well as long cascades of non-feedback blocks. (Interestingly, special characteristics of RSFQ one-pattern delay testability and fine-grained pipelining make delay testing a much smaller challenge for RSFQ relative to CMOS.)

We designed our new scan DFT approach to address the above challenges. Due to the unique characteristics of RSFQ logic, the basic principle behind our scan design for RSFQ is unique: it does not use the multiplexer-based scan cells used in

most CMOS designs (which are predominantly flip-flop based), nor does it use the level-sensitive scan design (developed for latch-based designs). Also, our scan design controls scan testing in new ways. We present our scan DFT design top-down and demonstrate its correctness via extensive simulations.

Based on the scan chain designed above, we designed a scan-based BIST structure by further extending our scan chain design by integrating pseudorandom pattern generator and single-input signature register into our scan chain design to enable at-speed test for the chip under test [77].

In order to evaluate the performance of BIST and DFT of the circuits, we emulate the circuit of a random pattern generator and generate K random patterns. Then, we apply each random pattern to the circuit to perform fault simulation to see the coverage of the pattern set we generated. Finally, we emulate the behavior of single input signature compressor and extract the compressed result from the compressor to check the coverage of the given pattern set.

We developed qTV to validate the circuit via JoSIM simulation. For a cell, the parameter values used by our timing analysis tool do not capture all the details when the cell works in a large circuit. Hence, qTV constructs a JoSIM simulation script based on the postrouting netlist and simulates it using JoSIM. Finally, qTV performs logic simulation to generate the golden result and uses it to check the JoSIM simulation results.

For instances where very large circuits result in long simulation runtimes, we divide the circuit into small test windows, as shown in [78]. Each test window is then simulated in a topological flow to validate the functional correctness of the given circuit.

D. Timing Analysis

We developed TimEx [79] to extract timing models for dc-biased RSFQ and energy-efficient RSFQ (ERSFQ) cells from JoSIM electrical simulations. For this, we developed the concept of flux signatures to identify every possible state—including error states—of a logic circuit. Version 2.05 (May 2020) of TimEx is available as open-source software [80].

We furthermore developed different methods to efficiently find the conditional probability density function (PDF) of the minimum workable clock period of SFQ circuits in view of manufacturing-induced process variations and present qSSTA, a statistical static timing analysis tool targeting SFQ circuits [81], [82]. Following a grid-based correlation model, qSSTA represents the spatial correlation of SFQ gates at different positions with respect to process parameters. By approximating the timing characteristics of SFQ gates in a linear model, qSSTA can estimate the clock period as a normal random variable. Furthermore, process variations that generally result in extra delays in CMOS circuits can result in functional errors in SFQ circuits. qSSTA derives the closed form of the conditional PDF of the clock period under the scenario, where all SFQ gates in the circuit work correctly. Compared to Monte Carlo simulations on lookup tables, experimental results show that the average percentage errors are 0.89% for the mean values, 8.04% for the standard

deviation, and 0.61% for the 98-percentile point, whereas the runtime of qSSTA is 83% faster on average.

In the case of timing characterization for AQFP logic cells, a tool called AQFPTX is used [83]. The adiabatic switching of AQFP logic cells means that there are no clear abrupt switching events typically used in other gate-level characterization approaches. A first attempt in the special treatment of this unique timing characteristic was started in [84], which used custom nonindustry standard timing definitions. In the SuperTools program, the ColdFlux team adopted industry standard timing parameters to better align with the overall top-down design flow through the development of AQFPTX. Because the logic cells are adiabatic, the timing definitions are all clock frequency dependent; thus, a set of timing parameters are provided for a number of clock frequencies with 5 GHz being the nominal standard. A standard delay format (SDF) file is generated after the timing characterization is finished. This SDF file can then be used in conjunction with a digital simulation tool that can run a gate-level simulation of an AQFP circuit via HDL modeling of logic gates used in the circuit [17]. The modular nature of AQFPTX also makes it easy to support other standard or custom timing information formats in the future. The tool is available through the ColdFlux repository as AQFPTX v1.3 [6].

E. Cell Optimization

We developed qCS, a stand-alone cell-level optimization tool, to analyze and optimize component values of superconductor-based cells such as RSFQ and AQFP using a graphical user interface (GUI). The tool supports the JSIM and JoSIM circuit simulators.

qCS has built-in capabilities such as critical margin calculation, parametric yield analysis, and critical margin range optimization. The tool utilizes a hybrid cell optimization methodology [85], [86] employing the Automatic Niching Particle Swarm Optimization and Fireworks Algorithm. Thus, its balanced characteristics of exploration and exploitation can increase the chances of finding better cell design values. During the optimization process, the tool is also capable of restarting the optimization while using the best result as an initial point. Its novel centering-favored margin calculation as an objective function utilizes both lower- and upper-bound margin sides to provide robust solutions against process variations.

The tool generates detailed output files for the analysis. Each session can be exported and imported. The generated files are independent of the operating system. Thus, the files initially exported on a system with Windows be imported to a system with CentOS 7 and vice versa.

We also implemented a Distance-to-Failure-Maximization optimization method [87] in JoSIM-Tools as one of three optimizers available in the ColdFlux tool suite. A center-of-gravity method was implemented in a stand-alone tool, Optimum, that was still only used in-house by the time that ColdFlux was completed.

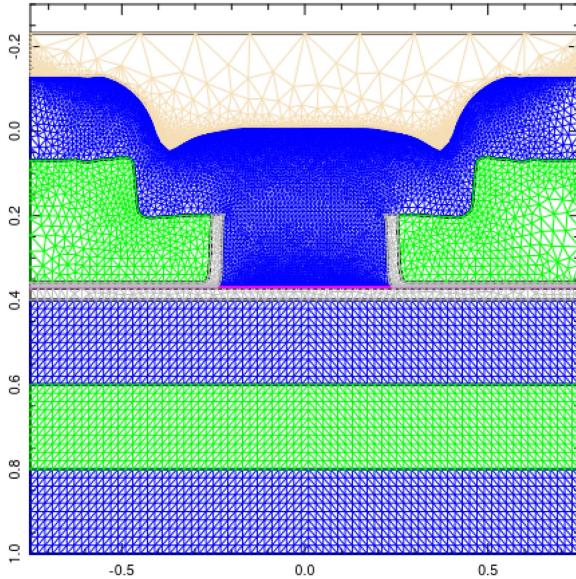


Fig. 6. FLOOPS generated finite-element mesh for SCE process flow.

VII. TECHNOLOGY CAD AND CELL LIBRARY DESIGN

A. Process and Device Modeling

The Florida Object-Oriented Superconductor Simulator (FLOOSS) was developed to meet the specific needs of SCE. The FLOOSS simulator was developed to augment the existing Florida Object-Oriented Process Simulator (FLOOPS) and Florida Object-Oriented Device Simulator (FLOODS). At the start of the SuperTools project, no coupled process and device simulator existed for the SCE. To start, a process simulator containing process steps unique to SCE had to be developed [88]. A device simulation approach also had to be developed based on the physics of JJ devices.

FLOOPS/FLOOSS solves the moving boundary problem using the level-set method (LSM). In an LSM simulation, a boundary representing the initial surface is propagated using a physics-based velocity function. A finite-element mesh is generated once the boundary has propagated for the specified amount of time. Many thin-film processing steps utilized in semiconductor processing are utilized in SCE, such as SiO₂ deposition and etching, chemical mechanical polishing, and niobium and aluminum sputtering. Process steps with existing semiconductor models were adapted for SCE. Novel processing steps had to be created from scratch, such as metal anodization and aluminum metal oxidation. An example of an SCE process flow simulation is shown in Fig. 6. The process simulator itself could be used as a stand-alone tool for investigating the geometric properties of SCE thin-film process steps, or the finite-element mesh it produces can be used in the FLOODS/FLOOSS tool.

Two approaches are taken to simulate JJ electrical properties in FLOODS/FLOOSS. The first is a semiclassical approach that models the room temperature resistance of a JJ similar to the room temperature measurements made on JJs. The semiclassical approach directly couples to the process simulator since it extracts conductance from a device scale finite-element mesh [89].

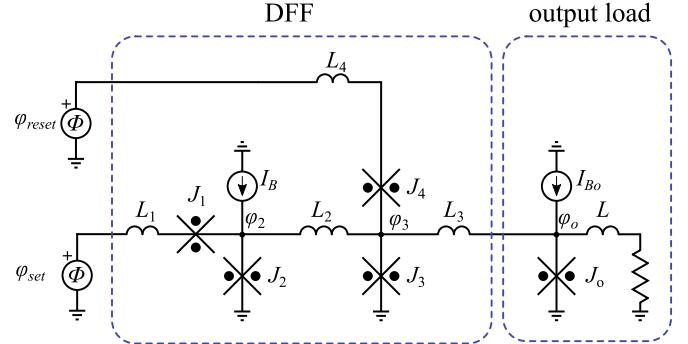


Fig. 7. Basic RSFQ DFF with input phase sources and output load.

[90]. Barrier layer thickness computed by the process simulator is used in device simulation to compute a local conductivity from the low-voltage ohmic tunneling condition. The simplicity of this approach enables the large-scale statistical simulation of thousands of JJs to investigate variability. The second approach is a direct atomistic *ab initio* simulation using the Bogoliubov–de Gennes (BdG) Hamiltonian [91], [92]. The BdG module simulates the superconducting properties of an arbitrary junction type on a finite-element grid, where the nodes represent atomic positions. Insulators are modeled using a spatially defined barrier height parameter, ferromagnetic layers are simulated using a magnetization parameter, and normal metals are simulated as normal regions with an absence of barrier height and magnetization. Current–phase relationships are primarily computed, and voltage can be applied to the system using a spatial chemical potential field. The drawback of the BdG solution method is that simulation resources grow rapidly with the size of the physical region, which limits structures that can be simulated on a typical personal computer to a few hundred atoms.

B. Parameterized Cell Libraries

1) *Rapid Single Flux Quantum*: At the start of SuperTools, a formalized design method for RSFQ circuits had not been published yet. Experienced circuit designers could assemble a circuit schematic from SQUID loops and use phase concepts to estimate working parameter values, but students or engineers new to the field could not study such methods as circuit theory from the publicly available literature.

Under ColdFlux, we formalized the design process for RSFQ circuits through the use of circuit equations in the phase domain [93], [94] and applied this to cell library development [93].

We showed that circuit component values—such as those of a basic RSFQ DFF shown in Fig. 7—can be described entirely in terms of parameterized equations, with the design steps detailed in [95].

The parameterized cell can now be adjusted for any required standard critical current I_C (which was chosen as 250 μA for the ColdFlux cell library). For example, the RSFQ DFF equations [95] for arbitrary values of the Stewart–McCumber parameter β_C , critical current I_C , and bias current fraction a (which was chosen as 0.7 for the ColdFlux cell library), and

B1	:	66	[***** *****	*****]	51
B2	:	64	[***** *****	*****]	43
B3	:	70	[***** *****	*****]	46
B4	:	62	[***** *****	*****]	44
IB2	:	56	[***** *****	*****]	72
L1	:	90	[***** *****	*****]	90
L2	:	59	[***** *****	*****]	90
L3	:	90	[***** *****	*****]	90
L4	:	90	[***** *****	*****]	90
Critical margin: 43% ['B2+']							

Fig. 8. Margins of the parameterized DFF for all the possible input combinations.

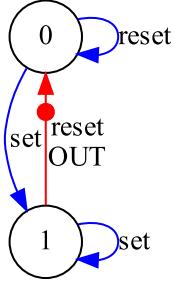


Fig. 9. Mealy state diagram of the RSFQ DFF. The two states are “0” and “1,” lowercase labels represent inputs, and the uppercase label with a filled circle represents an SFQ output.

with junction capacitance C scaled for I_C , are

$$I_{C(J2)} = I_{C(J3)} = I_C$$

$$I_{C(J1)} = I_{C(J4)} = 0.9I_C$$

$$I_B = aI_C$$

$$\begin{aligned} R_{\text{shunt}(J2)} &= R_{\text{shunt}(J3)} = \sqrt{\frac{\beta_C \Phi_0}{2\pi I_C C}} \\ R_{\text{shunt}(J1)} &= R_{\text{shunt}(J4)} = \sqrt{\frac{\beta_C \Phi_0}{2\pi I_C C(0.9)}} \\ L_1 &= L_4 = \frac{\Phi_0}{4I_C} - \frac{\Phi_0}{2\pi(0.9I_C)} \\ L_2 &= \frac{\Phi_0}{I_C} \\ L_3 &= \frac{\Phi_0}{4I_C}. \end{aligned}$$

The margins of the parameterized circuit, analyzed here without any parasitic elements for simplicity, are shown in Fig. 8.

The DFF remains fully functional, with the critical margin above 40%, when I_C is changed between 50 and 500 μA —and beyond—although this represents the range of interest.

One of the ways in which the ColdFlux modules make design easier is the use of flux signatures, obtained with TimEx from JoSIM simulation models, to exhaustively analyze each loop in a circuit under test in order to find all the possible states. A Mealy state diagram of a circuit under test then reveals hidden or *error* states that indicate undesired operation. The Mealy state diagram of the RSFQ DFF is shown in Fig. 9.

The ColdFlux RSFQ cell library [96] was designed to allow row-based place and route. Every cell was designed and laid out

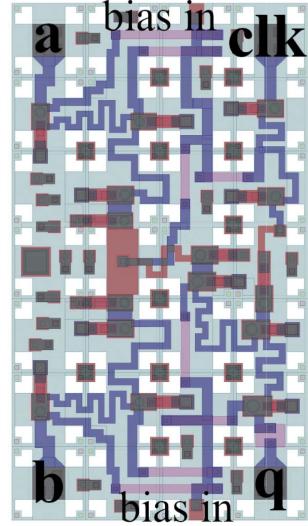


Fig. 10. Layout of a two-input RSFQ OR gate with integrated PTL drivers and receivers and covering 4×7 track blocks.

for two options: inductive interconnect between abutted cells and PTL interconnect between cells that are not directly adjacent to each other. All the PTL interconnects are routed as stripline with conductors in layer M1 or M3 and respective ground planes on M0 and M2, and M2 and M4, respectively. PTL characteristic impedance is 5.4Ω , and the phase velocity is around $96 \mu\text{m}/\text{ps}$. Under ColdFlux, we also investigated modeling of these PTL structures [97], [98], [99], [100].

In order to meet the design rule constraints of the MITLL SFQ5ee process [101], [102], a standard track block of $10 \mu\text{m} \times 10 \mu\text{m}$ was designed [103], and all the cells laid out to fit an integer number of track blocks in width and height.

The RSFQ OR2T cell (two-input OR gate with integrated PTL input receivers and output drivers) from version 3.0 of the ColdFlux RSFQ cell library [104] is shown in Fig. 10. It covers 4×7 track blocks for a total size of $40 \mu\text{m} \times 70 \mu\text{m}$, with pins to M3 and M1 at every input and output, and M5 bias input tabs at the top and the bottom.

The final RSFQ cell library at the completion of ColdFlux consists of 35 cells—four interface, thirteen logic, eight buffer, and ten interconnect cells. Cell specifications are listed in Table IV.

The SuperTools project required the development of cell libraries by each team to use with the respective toolchains. For a slightly different approach to cell library design, with smaller layout footprints, a different moat strategy and routing on layers M2 and M3 without a ground plane layer in between the different transmission line, see the work done at Hypres [105] for the Synopsys design flow [106]. In the Hypres library, an innovative layout framework allows switch-out of the bias structure to toggle cell technology between RSFQ and ERSFQ.

2) *Adiabatic Quantum Flux Parametron*: Throughout the project, the standard AQFP library has been significantly expanded and improved through the use of the tools developed during the SuperTools program. The cell design is loosely based on the work from [16] but implemented using the MITLL

TABLE IV
COLDFLUX RSFQ CELL LIBRARY SUMMARY

Standard Cells	Crit. Margin	Dimensions	Static Power	PTL Cells	Crit. Margin	Dimensions	Static Power
ALWAYS0_SYNC_NOA	-	70x20 μm	910 nW	ALWAYS0T_SYNC_NOA	-	70x10 μm	-
ALWAYS0_SYNC	-	70x20 μm	1370 nW	ALWAYS0T_SYNC	-	70x10 μm	-
ALWAYS0_ASYNC_NOA	-	70x10 μm	455 nW	ALWAYS0T_ASYNC_NOA	-	70x10 μm	-
ALWAYS0_ASYNC	-	70x20 μm	910 nW	ALWAYS0T_ASYNC	-	70x10 μm	-
JTL	65.8%	70x20 μm	901 nW	JTTL	28.6%	70x20 μm	913 nW
SPLIT	48.3%	70x20 μm	1370 nW	SPLITT	29.8%	70x30 μm	1510 nW
MERGE	21.9%	70x30 μm	1840 nW	MERGET	24.2%	70x50 μm	2370 nW
PTLTX	59.3%	70x20 μm	910 nW				
PTLRX	28.2%	70x20 μm	1270 nW				
AND2	33.8%	70x50 μm	2980 nW	AND2T	21.9%	70x50 μm	3340 nW
OR2	34.0%	70x40 μm	2970 nW	OR2T	26.1%	70x40 μm	3480 nW
XOR	21.2%	70x40 μm	2530 nW	XORT	17.0%	70x50 μm	3260 nW
NOT	31.2%	70x40 μm	1710 nW	NOTT	24.0%	70x40 μm	2070 nW
XNOR	18.0%	70x60 μm	3560 nW				
DFF	37.0%	70x30 μm	1940 nW	DFFT	22.9%	70x30 μm	2180 nW
BUFF	43.2%	70x30 μm	2150 nW	BUFFT	32.3%	70x20 μm	858 nW
NDRO	20.8%	70x40 μm	2590 nW	NDROT	23.6%	70x50 μm	4060 nW
DCSFQ	38.8%	70x20 μm	-	DCSFQ-PTLTX	37.2%	70x20 μm	-
SFQDC	27.0%	70x40 μm	-	PTLRX-SFQDC	29.1%	70x40 μm	-

SFQ5ee process. Both the dc-SQUID of the AQFP and the output transformer exist above the M4 ground plane, resulting in a relatively large cell area. The layers below M4 are completely dedicated to PTL interconnects. This is in contrast with the cell design proposed in [107], which opted to keep the dc-SQUID above M4 and stack the output transformer below it resulting in a compact footprint but with the tradeoff of introducing more routing obstacles below M4 and having less flexibility in parameterizing the internal structures due to the compactness. The larger cell area in this work allows for fewer constraints for future parameterization and autogeneration of cell layouts.

The AQFP library includes 81 main cells and additional subcells used to construct the main cells. Subcells are designed to have a direct connection with other AQFP cells or have a single PTL connection at either the input or output ports. Subcells are meant to be used for the construction of larger AQFP cells, such as the AND2 cells, and can also be used for manual test circuit setup. Main cells are constructed through subcells and are designed to be connected through PTLs. This is illustrated in Fig. 11. The main cells within the AQFP library include the following:

- 1) **subcells:** bfr, bfrL, inv, const0, const1;
- 2) **fan-outs:** spl2, spl2L, spl3, spl3L;
- 3) **current boosters:** boost1, boost2f2 (2 fan-out elements), boost2f4 (4 fan-out elements);
- 4) **memory elements:** storage_gate, qfpl (QFP latch), ndro_qfpl (nondestructive read-out QFP latch), ndro_fb (NDRO with feedback);
- 5) **two-input AND logic:** AND2 with all 4 input combinations (pp, pn, np, nn);

- 6) **three-input AND logic:** AND3 with all 8 input combinations (ppp, ppn through to nnn);
- 7) **two-input OR logic:** OR2 with all 4 input combinations (pp, pn, np, nn);
- 8) **three-input OR logic:** OR3 with all 8 input combinations (ppp, ppn through to nnn);
- 9) **three-input majority logic:** MAJ3 with all 8 input combinations (ppp, ppn through to nnn);
- 10) **five-input majority logic:** MAJ5 with all 32 input combinations (ppppp, ppppn through to nnnnn);
- 11) **hybrid interfaces and readout:** rsfq2aqfp, aqfp2rsfq, qdc.

Note that AQFP logic gates can perform inversion of inputs directly without any need for a discrete inverter; thus, we adopted a “p” and “n” notation to refer to each data input terminal as positive (direct) or negative (inverting), respectively, for each of the Boolean logic gates. The cells appended with an “L,” such as “bfrL” or “spl2L,” indicate that the cells are designed to provide a larger output current for intermediate transmission lines (up to 0.8 mm). The current booster cells are used for long transmission lines up to 1.7 mm.

The library includes netlists, schematics, symbols, layout, digital simulation, and LVS confirmation files for each cell. The cell layouts incorporate the standard track routing architecture developed during the ColdFlux project. Both the RSFQ and AQFP cell libraries delivered for the project implement the standard track routing architecture to ensure interoperability between the libraries. This also improves the interface with placement-and-routing routines developed during the project. Toward the latter end of the SuperTools program, a detailed investigation on flux trapping effects on the AQFP buffer subcell

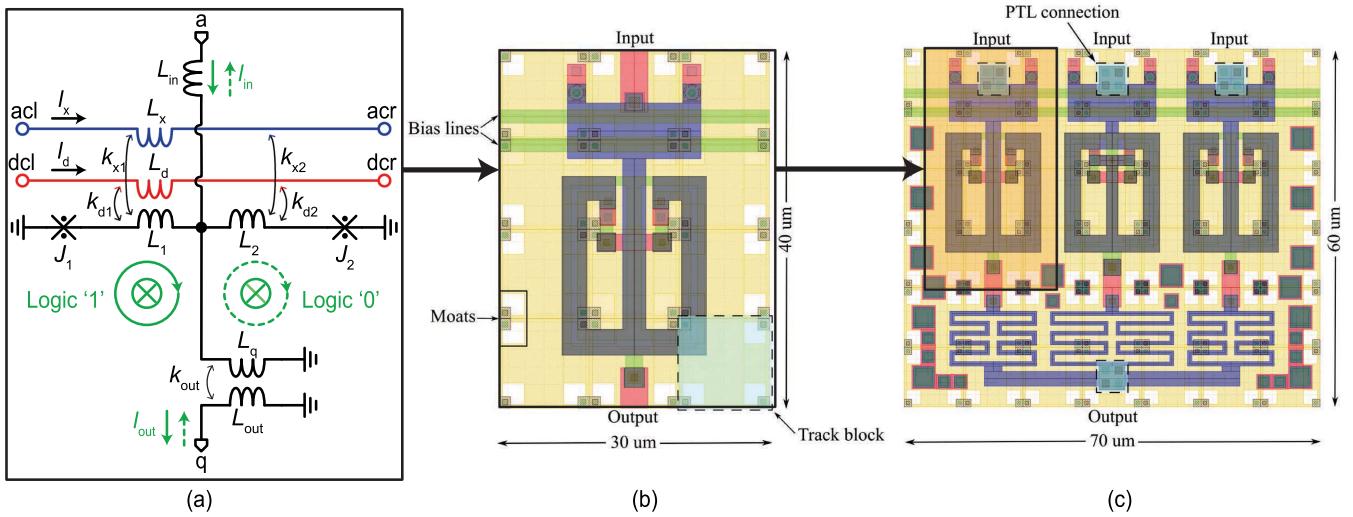


Fig. 11. Hierarchical assembly of an AQFP logic gate for the ColdFlux AQFP cell library using the MITLL SFQ5ee $100\text{-}\mu\text{A}\cdot\mu\text{m}^{-2}$ process. (a) JJ-level schematic of the AQFP buffer, a typical subcell of the library. (b) Physical layout of the AQFP buffer. Note that it is built using a $10 \times 10 \mu\text{m}$ track block. (c) Completed MAJ3_ppp logic gate where three buffers are connected in parallel using an inductor-based merging network. In this version, the I/Os have minor modifications to directly connect to PTLs. Also, the three buffer subcells partially overlap each other to create a more compact layout. (a) AQFP buffer (schematic). (b) AQFP buffer (layout). (c) AQFP MAJ3 (layout).

was conducted, and it showed sufficient performance even in the presence of multiple fluxons trapped in the moats. The details of the finalized version of the AQFP cell library used in this research program are described in [108], and the library is available to download from the ColdFlux repository [6].

In the future, we intend to investigate more compact AQFP structures especially by eliminating the large output transformer through direct coupling or through new π -JJ rf-SQUIDS that behave as negative couplers. A more detailed flux trapping analysis will also be conducted on the full set of AQFP logic gates, which is now possible through the compact model development done in ColdFlux.

C. Layout Parameter Extraction

Layout parameter extraction was already mature for inductance extraction with the InductEx tool suite at the start of ColdFlux [54], [55], [56], [57], but has been improved substantially over the course of the project.

1) *Acceleration of Numerical Methods:* Under ColdFlux, the aim was to drastically improve the maximum size of models that can be extracted. One research direction was the development of methods to improve the efficiency of numerical electromagnetic field solvers that use a multilevel fast multipole algorithm (MLFMA) solver to reduce required memory and accelerate the solution. An alternative method, multilevel adaptive cross-approximation (MLACA) solver with singular value decomposition recompression [109], [110] was developed to replace MLFMA in a solver such as FastHenry [111], where it was shown to require less memory than FastHenry's MLFMA for the same solution accuracy and offered control over accuracy as a speed tradeoff. However, rapid advances in the TetraHenry engine under ColdFlux negated the gains of MLACA through substantial speed improvements in MLFMA. The same MLFMA

acceleration methods were used to speed up magnetic field calculation significantly [112].

2) *Reengineered Meshing:* At the start of the SuperTools program, InductEx already supported multilayer models for full-gate layout extraction with interleaved cuboid meshes [57] and tetrahedral meshes [113]. Although our prior work meant that solution speed was fast [114], with typical cells extracted in tens of seconds to a few minutes, multilayer circuit models required many segments—in excess of several hundreds of thousands, due to the nature of our earlier mesh generation algorithms.

Under ColdFlux, we improved 3-D modeling and meshing significantly. Polygons are now smoothed with the Ramer–Douglas–Peucker algorithm to lower the vertex count and to allow for smoother, more uniform meshes. We also added hybrid meshing to the TetraHenry engine, which allows sheet current approximation and the use of hybrid meshes that contain cuboid, tetrahedral, and triangular mesh segments. The meshed model of a shunted JJ that demonstrates the different meshing options is shown in Fig. 12.

With the inclusion of elevation change over the surface area of a layer (although that is not required for the planarized MITLL SFQ5ee process), smoothed polygons, and support for any via-to-metal and metal-to-metal contact configuration, InductEx is now able to model a standard ColdFlux library cell with around 10 000 or fewer mesh segments. At the time of writing, InductEx easily handles large circuit models with 10 000 000 segments on a computer with 128-GB RAM.

3) *Improvement to Mutual Inductance Fidelity:* At the start of ColdFlux, InductEx used process definition files that were calibrated for mesh segment sizes of $2 \mu\text{m}$ so that layouts with linewidths down to $1 \mu\text{m}$ would be handled adequately [115]. However, experimental measurements on the weak coupling between very narrow stripline layouts by SuperTools T&E partners were shown [116] to cause significant overestimation of the mutual inductance by InductEx.

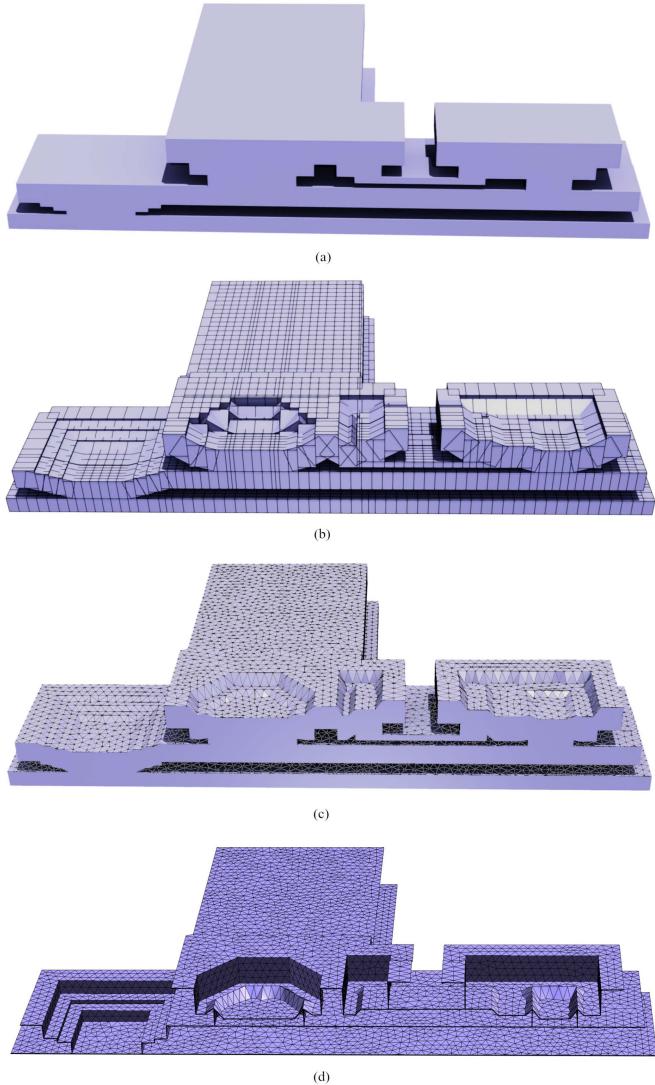


Fig. 12. Cross section of an InductEx model of a shunted JJ with (a) no elevation change, (b) a cuboid mesh, (c) a tetrahedral mesh, and (d) a triangular mesh with elevation change.

The cause of this overestimation comes from modeling. When lines are much narrower than the segment size—in this case down to $0.25 \mu\text{m}$, or eight times narrower than the segment size—the ground plane segments are far bigger than the linewidth. The return current is then modeled inaccurately beneath lines. Where the ground plane segments overlap both lines, the coupling is then created artificially. The InductEx model for such a circuit is shown in Fig. 13.

The easy solution is to decrease the maximum segment size to $0.25 \mu\text{m}$, but the resource cost for large layouts is prohibitive. A more elegant solution is to cast “shadows” from every object to the nearest ground planes above and below and to create mesh elements that have edges on the shadow boundaries, as is shown in Fig. 14. Furthermore, the addition of narrow segments around the outside of every conductor with a width equal to the penetration depth results in much better modeling of the edge current distribution and improves the accuracy of mutual

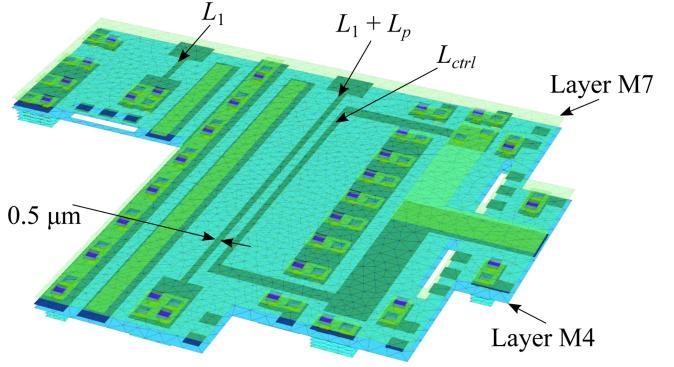


Fig. 13. InductEx model of a differential-arm inductance measurement SQUID for the MITLL SFQ5ee 10-kA · cm $^{-2}$ process with L_1 and L_{ctrl} in layer M5.

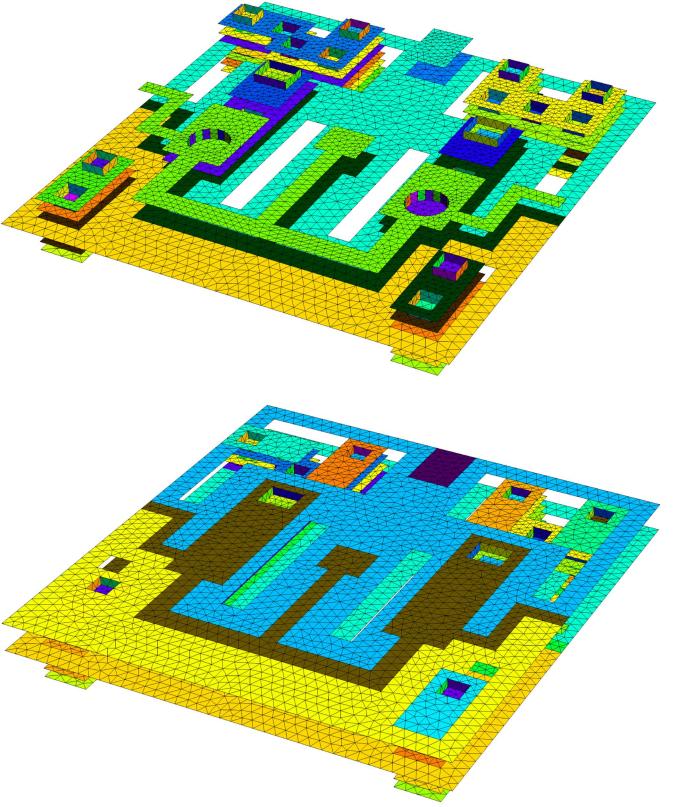


Fig. 14. Rendered image of the 3-D inductance model created by InductEx for an MITLL SFQ4ee JTL layout. The top image shows the model without the skyplane in M7, with shadow casting to the ground plane M4 visible. The bottom image shows the model with the M7 skyplane and the shadows cast on it included.

inductance extraction. All these methods were added to InductEx under ColdFlux.

The root-mean-square error (RMSE) results between measurement and calculation are shown in Table V. The mutual inductance differs between about 30% of the self-inductance for half of the structures, where the coupling is between overlapping striplines on different layers and 9% of the self-inductance where the coupling is between adjacent lines, as shown in Fig. 13. The table includes RMSE results when the mutual inductance is normalized to self-inductance. Shadow casting and edge slicing

TABLE V
RMSE RESULTS FOR VERY NARROW COUPLED STRUCTURES IN THE MITLL SFQ5EE PROCESS

Meshing method	RMSE of L	RMSE of M	RMSE of M normalized to L
Normal mesh	7.61%	29.5%	3.46%
Shadow casting	4.89%	8.19%	3.19%
Shadow casting and edge slicing	2.69%	4.66 %	1.47%

TABLE VI
SUMMARY OF INDUCTEX RESULTS FOR INDUCTANCE EXTRACTION VERSUS MEASUREMENT WITH THE CUBOID AND TETRAHEDRAL SEGMENT OPTIONS OF INDUCTEX

Figure of merit	FFH	TTH
Average error for 56 tests	1.8 %	2.4 %
Results within 15 % tolerance	100 %	100 %
Results within 10 % tolerance	100 %	98 %
Results within 5 % tolerance	98 %	89 %

bring self-inductance and normalized mutual inductance within an RMSE of 3%. Crucially, this is for lines with width down to 0.25 μm , while segment size is 2 μm —a significant result.

These mutual inductance experiments confirm that errors in extracted inductance arise almost solely from modeling.

After the implementation of high-fidelity mutual inductance modeling and compact model extraction tools, it was applied to the analysis and improvement of AQFP layouts [117].

4) *Calibrated Results:* InductEx was calibrated against measured results for inductance and mutual inductance results provided by MITLL. Errors between calculated and measured results with the calibrated process definition files but without high-fidelity modeling (shadow casting and edge slicing) are shown in Table VI.

The test results, individually, are listed in Table VII.

With the latest available self and mutual inductance test structures from MITLL, and with high-fidelity modeling enabled in InductEx, calculated versus measured results when triangular segments are used are shown in Table VIII. A summary of the results by tolerance band is shown in Table IX.

D. Design Rule Checking

During the ColdFlux project, we developed a comprehensive Design Rule Check (DRC) script tailored for the MITLL SFQ5ee process. This versatile script can be executed within KLayout [9] or directly from a command line terminal, providing users with flexible interaction options. To enhance usability, we have integrated a GUI that simplifies the execution of the script and visualization of the results, as shown in Fig. 15. A notable feature of the DRC script is its support for hierarchical mode, also known as “deep” mode, which accelerates DRC checks for large-scale layouts and significantly streamlines complex layout designs. The performance of the DRC script has been extensively tested on layouts with up to 10 million JJs.

TABLE VII
PER-TEST RESULTS FOR INDUCTANCE EXTRACTION VERSUS MEASUREMENT WITH THE CUBOID AND TETRAHEDRAL MESH OPTIONS OF INDUCTEX

Inductance structure	Width μm	FFH (%)	TTH (%)
M1-M0	0.35	-2.08	-1.80
M1-M0	0.5	-0.83	-0.13
M1-M0	0.7	1.04	2.23
M1-M0	1	1.79	2.48
M1-M0	2	-0.61	0.56
M0-M1-M2	0.35	-1.72	-1.59
M0-M1-M2	0.5	0.93	1.63
M0-M1-M2	0.7	2.11	3.30
M0-M1-M2	1	-0.18	0.54
M0-M1-M2	2	2.51	3.66
M2-M3-M4	0.25	-3.55	-10.7
M2-M3-M4	0.35	-3.24	-8.19
M2-M3-M4	0.5	-1.58	-4.74
M2-M3-M4	0.7	-2.34	-4.21
M2-M3-M4	1	-1.65	-3.49
M2-M3-M4	2	-3.64	-4.46
M5-M4	0.35	-2.08	-5.20
M5-M4	0.5	1.17	-0.94
M5-M4	0.7	2.89	1.66
M5-M4	1	-0.64	-1.8
M5-M4	2	0.88	-0.39
M5-M4	4	0.45	0.40
M4-M5-M7	0.35	-3.81	-7.07
M4-M5-M7	0.5	1.18	-1.01
M4-M5-M7	0.7	2.69	1.30
M4-M5-M7	1	0.72	-0.53
M4-M5-M7	2	1.66	0.57
M4-M5-M7	45	2.24	2.44
M6-M4	0.35	-2.42	-2.27
M6-M4	0.5	-0.52	-0.31
M6-M4	0.7	-0.16	0.12
M6-M4	1	-0.89	-1.41
M6-M4	2	-0.56	-0.90
M6-M4	4	-1.15	-1.10
M6-M6-M7	0.35	-4.53	-4.36
M6-M6-M7	0.5	9.27	10.0
M6-M6-M7	0.7	1.06	1.31
M6-M6-M7	1	-2.75	-1.93
M6-M6-M7	1.4	0.39	1.71
M4-M7	0.35	-1.48	-1.54
M4-M7	0.5	0.97	0.94
M4-M7	0.7	0.42	0.39
M4-M7	1	4.43	3.59
M4-M7	1.4	2.55	1.39
M5-M7	0.35	-3.85	-6.5
M5-M7	0.5	-1.15	-2.89
M5-M7	0.7	2.46	1.23
M5-M7	1	-1.76	-3.02
M5-M7	2	1.17	-0.91
M5-M7	4	0.77	-0.18
M6-M7	0.35	-2.73	-2.50
M6-M7	0.5	0.58	1.25
M6-M7	0.7	-0.12	0.81
M6-M7	1	0.36	0.87
M6-M7	2	1.11	2.35
M6-M7	4	0.21	2.31

E. LVS Verification

In the initial phase of the SuperTools ColdFlux project, there was an absence of an LVS verification tool suitable for large-scale superconducting electronic circuit layouts. As a consequence of the project, we have developed InductEx-LVS, a comprehensive LVS tool specifically tailored for superconductor circuit layouts, as shown in Fig. 16. InductEx-LVS facilitates large-scale LVS analysis, offering meticulous error reporting in the

TABLE VIII

PER-TEST RESULTS FOR INDUCTEX INDUCTANCE EXTRACTION VERSUS MEASUREMENT ON COUPLED INDUCTORS WITH WIDTHS FROM 0.5 TO 1.0 μm WHEN HIGH-FIDELITY TRIANGULAR MESH MODELS WITH SHADOW CASTING AND EDGE SLICING ARE USED

Inductance	Ground	Measured L (pH)	Measured M (pH)	Calculation difference L (%)	Calculation difference M (%)
M0-M0	M1	14.7	1.35	0.9	7.5
M1-M1	M0, M2	12.0	0.26	-2.6	0.8
M2-M2	M1, M3	12.0	0.32	0.6	2.8
M3-M3	M2, M4	11.7	0.37	-1.9	4.5
M5-M5	M4, M7	12.7	0.16	0.3	7.9
M6-M6	M4, M7	10.0	0.24	4.5	6.6
M0-M0	M1	21.7	8.21	2.9	5.6
M1-M1	M0, M2	19.5	7.00	-0.4	4.8
M2-M2	M1, M3	19.1	7.12	3.5	4.7
M3-M3	M2, M4	19.1	7.38	1.9	4.5
M5-M5	M4, M7	20.2	6.95	1.6	2.6
M6-M6	M4, M7	17.9	7.38	4.1	2.8
M6-M5	M4, M7	10.4	3.21	1.2	4.5
M5-M6	M4, M7	12.6	3.19	1.6	4.9
M3-M2	M1, M4	13.1	4.44	3.2	3.9
M2-M3	M1, M4	13.8	4.45	2.5	3.6
M1-M2	M0, M3	13.3	4.01	0.1	4.2
M0-M1	M2	19.0	7.10	4.3	7.2
M6-M5	M4, M7	18.0	10.4	3.7	3.6
M5-M6	M4, M7	20.1	10.5	2.1	3.4
M3-M2	M1, M4	20.2	11.4	3.1	2.1
M2-M3	M1, M4	21.0	11.4	2.5	1.9
M1-M2	M0, M3	20.6	10.7	1.5	3.4
M0-M1	M2	25.3	13.9	4.8	5.7

TABLE IX

SUMMARY OF INDUCTEX RESULTS FOR INDUCTANCE AND MUTUAL
INDUCTANCE EXTRACTION VERSUS MEASUREMENT WITH TRIANGULAR
MESHS AND HIGH-FIDELITY MODELS

Figure of merit	Inductance (L)	Mutual inductance (M)
Average error for 24 tests	1.8 %	4.3 %
Results within 15 % tolerance	100 %	100 %
Results within 10 % tolerance	100 %	100 %
Results within 5 % tolerance	100 %	75 %

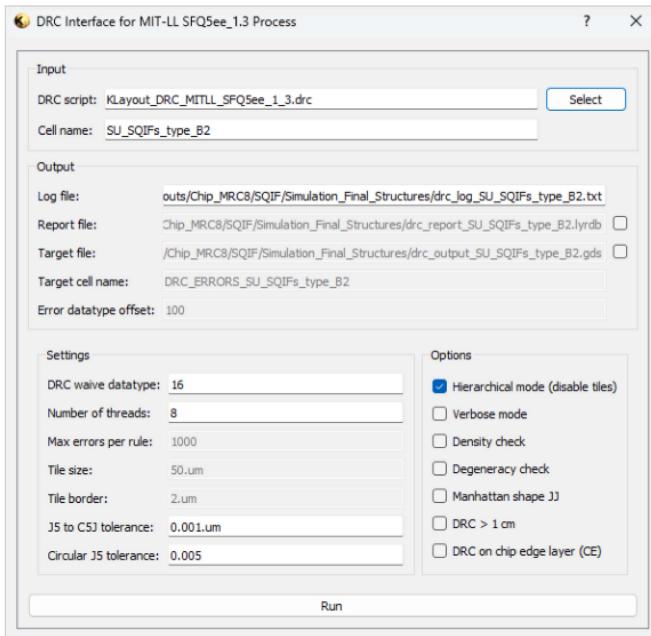
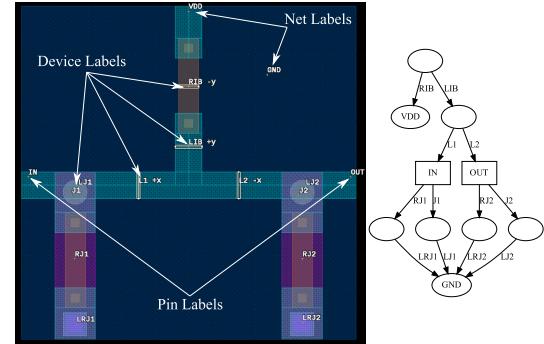
Fig. 15. Graphical interface for performing DRC in *KLayout*.

Fig. 16. Layout of a JTL cell features LVS labels that define the locations of pins, nets, JJs, and resistors. In the extracted layout graph, vertices represent pins and nets, while edges correspond to devices such as inductors, resistors, and junctions.

process. To bolster the tool's performance, we have optimized graph extraction through the implementation of quadtrees and refined the graph isomorphism algorithm. These improvements have substantially decreased simulation times and memory consumption. In addition, we have ensured compatibility between the latest AQFP and RSFQ cell libraries and InductEx-LVS, while also incorporating compact model extraction capabilities. The efficacy of the tool has been demonstrated by successful testing on a 1 cm \times 1 cm chip layout and a 10-million-JJ test case.

F. Compact Simulation Models

During the SuperTools ColdFlux project, we developed tools for compact model extraction to enhance the accuracy of superconductor logic circuit simulations, including AQFP

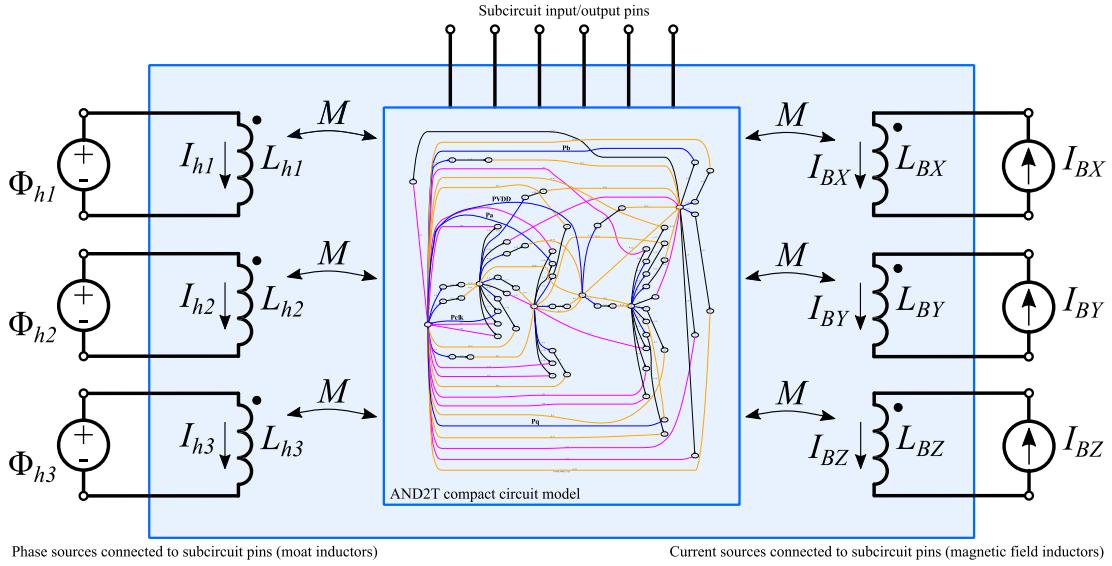


Fig. 17. Equivalent compact circuit model of an AND2T cell with moat (fluxon) and external magnetic field inductors, extracted with InductEx-LVS.

circuits [115]. Traditional hand-designed netlists often lack complete mutual inductances, causing discrepancies between the design schematic and the layout. By integrating compact simulation model extraction into the InductEx toolchain, we achieved a more accurate simulation model that includes all mutual inductances, allowing for improved verification of circuit performance after layout. Compact model extraction can be incorporated as the final step in cell library characterization for superconductor logic circuits. The compact models incorporate both fluxon (moat) and external magnetic field inductors, as shown in Fig. 17. These inductors are coupled to all the inductors within the compact circuit model. Although not shown in Fig. 17, the compact circuit model can also be extracted to include first- and second-order gradient field components.

1) *Effects of Quasi-Particles*: Electrical circuit simulators such as JSIM or earlier versions of JoSIM use the RCSJ model for JJs. The RCSJ model only accounts for the dynamics of Cooper pairs. It is sufficient for the simulation of most digital operations, as long as clock frequencies are below about a third of the gap frequency of superconductors. However, at higher frequencies or closer to the critical temperature T_c , as well as for analog devices like Josephson on-chip oscillators, quasi-particles play a role that cannot be neglected. In practice, the RCSJ model has variants in JSIM or JoSIM, which partially account for the nonlinear subgap resistance observed in the $I-V$ curves of JJs. However, they do not take into account the full electrodynamics of JJs that occurs when Cooper pairs are broken into quasi-particles, either by thermal effects or by incoming more energetic photons. Most of these effects on the tunneling properties of JJs have been described first by Werthamer [118] and then by Larkin and Ovchinnikov [119] and Harris [120]. We first developed a “Werthamer simulator” (WSIM) whose objective was to improve the model of JJ tunneling to account for the presence of quasi-particles in the presence of RF sinusoidal signals applied to the junction. The WSIM is able to calculate $I-V$ curves as well as current and voltage waveforms in the time

domain. It is based on analytical formulas, which make it fast but limited to only specific incoming waveforms reaching the junction.

In order to extend the capabilities of WSIM in the time domain, we developed QuickSunds (QUAsI-particle and Cooper pairs Kernel-based SUpercoNDucting Simulator), which has the advantage over WSIM to work with any time-domain voltage or phase waveform applied to the junction. This is done at the price of a longer simulation time due to numerical techniques associated with the fitting of the frequency kernels that describe the behavior of JJs from the Bardeen–Cooper–Schrieffer (BCS) theory [121] in the presence of both Cooper pairs and quasi-particles. Fig. 18 shows the interface of QuickSunds on macOS. A Windows 10 version is also available. Once the frequency kernels are determined, QuickSunds provides a normalized set of parameters embedded in a compact model that can be directly used to run time-domain simulations from the JoSIM, which embeds the set of equations necessary to interpret the compact model accounting for quasi-particle effects, based on the works of Werthamer, Larkin and Ovchinnikov, and Harris.

QuickSunds needs to be run only one time for a given set of current density, superconducting materials, and temperature of operation. It can save data on files, including the *.model* syntax. An example of compact model derived by QuickSunds is shown in Fig. 19. Since values are normalized, the same model can be used for all junctions on a chip, which are usually made of the same materials. Nevertheless, it is possible to have different models on the same chip, for instance, if some junctions work at a different temperature because of thermal effects.

Fig. 20 shows the response of a JJ biased over its critical current I_c ($= 100 \mu\text{A}$) to produce an on-chip clock signal at 540 GHz. The junction is externally shunted such that its McCumber parameter is $\beta_c = 1$ to avoid any $I-V$ curve hysteresis for correct digital operation. Its area and corresponding capacitance were adapted to keep I_c constant. Three cases were studied for the three current densities of MITLL process, 10,

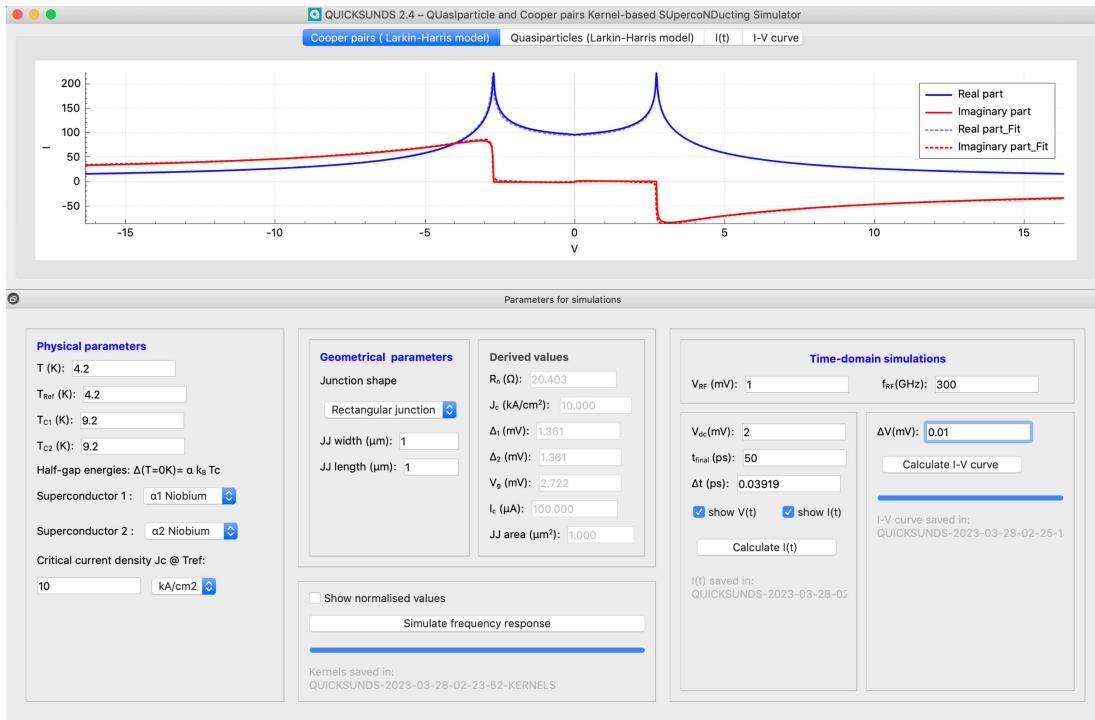


Fig. 18. QuickSunds macOS interface showing frequency kernels for Cooper pairs.

```
QUICKSUNDS-2023-03-28-02-49-17-COMPACT-MODEL.txt
.model JJQP TJ(Vg=2.742mV, CAP=50pF, Rn=6.55,
Icrit=0.314159mA, T=4.2k, -2.06379, 3, 0.100687, 0.00360531, 0.0258884, 3, -2.08
063, 0.10101, 0.0261455, 0.0036138, 3, 1.64512, -0.0389763, 0.000952943, -0.00176
694, -1.64682, -3, 0.0403927, 0.00181476, -0.000948671, 3, 1.8489, 0.336196, 0.01,
0.06879, 1.85055, 3, 0.33927, 0.0692147, 0.01, -1.1, -0.57424, -1.05774, -0.999989
, -1.00273, 0.580184, 1, 1, 0.05979, 1, 0.0282, 0.999998, -3, 2.98002, 0.392979, 0.006
98935, 0.0561122, -2.02931, -3, 2.63608, 0.0178252, 2.07954, 3, -3, 1.15405, 0.0013
2506, 0.0212481, -3, -1.883, 3, -0.00292417, 0.783285, 2.91808, 3, 1.02858, 0.01551
74, 0.134881, 3, 0.662506, 1.37015, 0.0318261, 0.531364, -1.1, 1.1, -0.813143, -1.0
0093, -0.992877, 1.1, 1.1, 1.1, 1.00139, 1.08323)
```

Fig. 19. Compact model produced by QuickSunds to be used as a JJ model in the JoSIM.

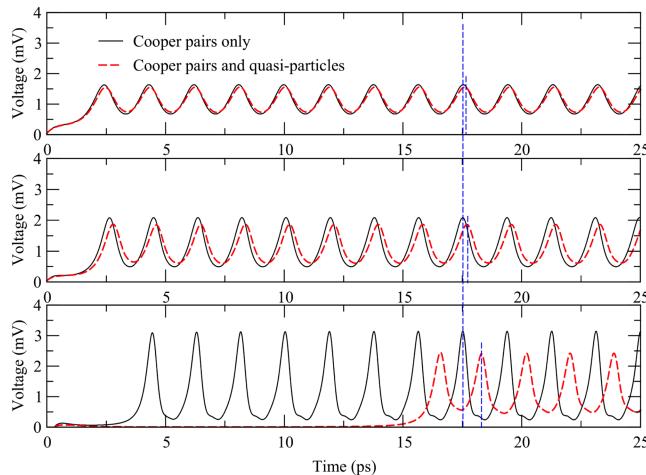


Fig. 20. Response of a $\beta_c = 1$ JJ with $I_c = 100 \mu\text{A}$ overbiased to produce a pulse train at 540 GHz, with and without considering quasi-particles. $J_c = 10 \text{ kA/cm}^2$ (top), 20 kA/cm^2 (middle), and 50 kA/cm^2 (bottom). The vertical dashed blue lines illustrate the delay observed in simulations when quasi-particles are taken into account.

20, and 50 kA/cm^2 , which, respectively, correspond to $R_n I_c$ voltages of 0.81, 1.15, and 1.81 mV. One can observe that the 540-GHz frequency is too high for the lowest critical current density, resulting in a signal that tends toward the waveform of a sinusoidal signal due to low-pass filtering by the junction's capacitance. The situation is intermediate for 20 kA/cm^2 , while clear SFQ pulses can be seen for the current density of 50 kA/cm^2 that corresponds to a higher $R_n I_c$ product. The influence of quasi-particles is negligible for $J_c = 10 \text{ kA/cm}^2$ and limited for $J_c = 20 \text{ kA/cm}^2$. It is clearly visible for the highest current density of 50 kA/cm^2 , resulting in a reduction of the pulse amplitude due to resistive losses caused by additional quasi-particles (shorter pulses result in more energy at higher frequencies that break more Cooper pairs), and a delay in the production of pulses, that can cause synchronization issues at higher clock frequencies if these effects are not taken into account during design simulations.

2) *External Magnetic Fields:* We have made significant advancements in magnetic field simulation for superconducting electronic circuits. We introduced robust and repeatable post-layout verification methods for analyzing the effects of static magnetic fields on circuit operation, enabling designers to assess circuit layouts for static magnetic field tolerance [122]. Furthermore, we developed an adaptive fast multipole algorithm to accelerate the computation of magnetic fields surrounding current-carrying superconducting volumes, employing a hierarchical tree of cubic cells and vector spherical harmonics to approximate the gradient of Green's function [112]. The algorithm demonstrated its ability to calculate trapped flux magnetic fields and magnetic fields around type-II superconducting microstrips,

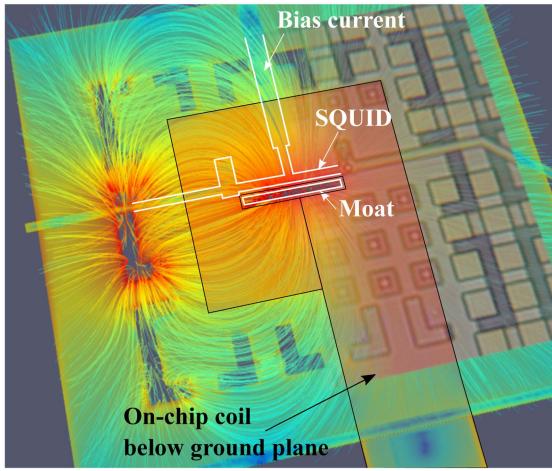


Fig. 21. Overlay of microscope photograph and InductEx simulation of the magnetic field created by a trapped fluxon for a test SQUID.

with the overall complexity found to increase linearly with the number of evaluation points [112]. We also improved InductEx and TTH to efficiently evaluate superconducting gradiometer layouts, presenting numerical methods for analyzing and extracting the magnetic coupling between orthogonal components of a gradient magnetic field and the inductive coils of first- and second-order gradiometers.

3) *Flux Trapping*: Several tools have been developed under the ColdFlux project for flux trapping analysis in superconducting ICs. These tools focus on analyzing the effects of external magnetic fields, trapped flux on circuit operation [122], and the use of moats to create low-energy locations for flux trapping [123]. Numerical simulation tools have been created to extract compact models for magnetic flux trapped in moats and their coupling to superconducting structures, with validation through experiments [123], [124], as shown in Fig. 21. Advances in InductEx and the TetraHenry numerical engine have enabled analysis of the coupling of trapped flux in moats to superconductor circuit structures, such as AQFP cells, and their influence on circuit performance [125]. The project has also investigated the impact of moat placement and geometry on circuit performance and provided design rules for optimizing moat configuration [124]. In addition, the project has improved the accuracy of simulating trapped flux in AQFP superconductor logic circuits through full-circuit inductance extraction and compact model extraction, ensuring better circuit performance verification after layout [117].

4) *Superconducting PTLs*: At higher current densities, SFQ pulses produced by JJ switching events are shorter, and some Cooper pairs of the superconducting propagation medium can be broken, resulting in absorption and dispersion. In the RF domain, particularly in radio astronomy, this effect is well known and led in the past to the use of other superconducting materials, like NbTiN instead of Nb, to alleviate this issue. For SFQ circuits, the main consequence of this effect is that SFQ pulses are broadened, and their amplitude decreases with the distance of propagation. After 1 mm, it may happen that they are not able to trigger the receiving JJ on the other end of the transmission line. To

mitigate these effects, the first thing is to know how pulses really propagate on the lines. Due to the nonlinear properties of the complex conductivity of superconductors derived from the BCS theory, which also accounts for kinetic inductance effects, such calculations are nontrivial in the time domain. SuperLink3.1 (see Fig. 22) solves this issue and is able to see SFQ pulse dispersion and absorption occurring along superconducting lines transmission lines from a frequency domain approach, followed by inverse Fourier transforms for time-domain analysis.

VIII. RESULTS

The ColdFlux project was ultimately successful in delivering a set of tools that are vastly more capable than the loose collection of design methods that existed at the start of the SuperTools program. In order to showcase the capabilities, we present some results achieved with the ColdFlux tools. Results on individual tools have been presented throughout this text. We briefly summarize the results as measured against the project milestone requirements here for analog design and synthesis (see Table X) and for physical design and verification (see Table XI).

IX. APPLICATION

A. ColdFlux Tools Used Outside the Project

One measure of the success of a design tool development project is arguably the uptake of the tools by end users other than those directly involved with the project.

On the physical level, the JoSIM has rapidly replaced JSIM as the preferred simulation engine in SCE digital design groups from the USA through China to Japan, so much so that its user manual has been translated into Japanese. It is now used widely for SFQ circuit simulation [126], [127] and recently even for the simulation of oscillatory neural networks [128].

The parameter extraction and verification tools InductEx and TetraHenry are very widely used for digital circuit layout verification—including memory design [129]—but toward the end of ColdFlux, a significant user base has evolved into analog applications such as JJ-based amplifiers [130], SQUID and SQIF analysis [49], [131], [132], negative-inductance SQUID applications [133], SQUID-based calorimeters [134], [135], SQUID gradiometers [136], SQUID probes [137], and quantum electronics (specifically for filter design in quantum annealing systems [138] and the extraction of weak coupling to qubits).

A number of institutions expressed interest and started using the AQFP top-down flow for their own circuits or projects. These include LBNL (USA) and Kyushu University (Japan).

B. Platforms

The commercial modules developed under ColdFlux are used by end users on Linux platforms (about 45%), Microsoft Windows 10 (about 45%), and macOS (about 10%). Of Linux users, almost all use Red Hat Enterprise Linux 7 or 8, with the rest mostly using CentOS 7.

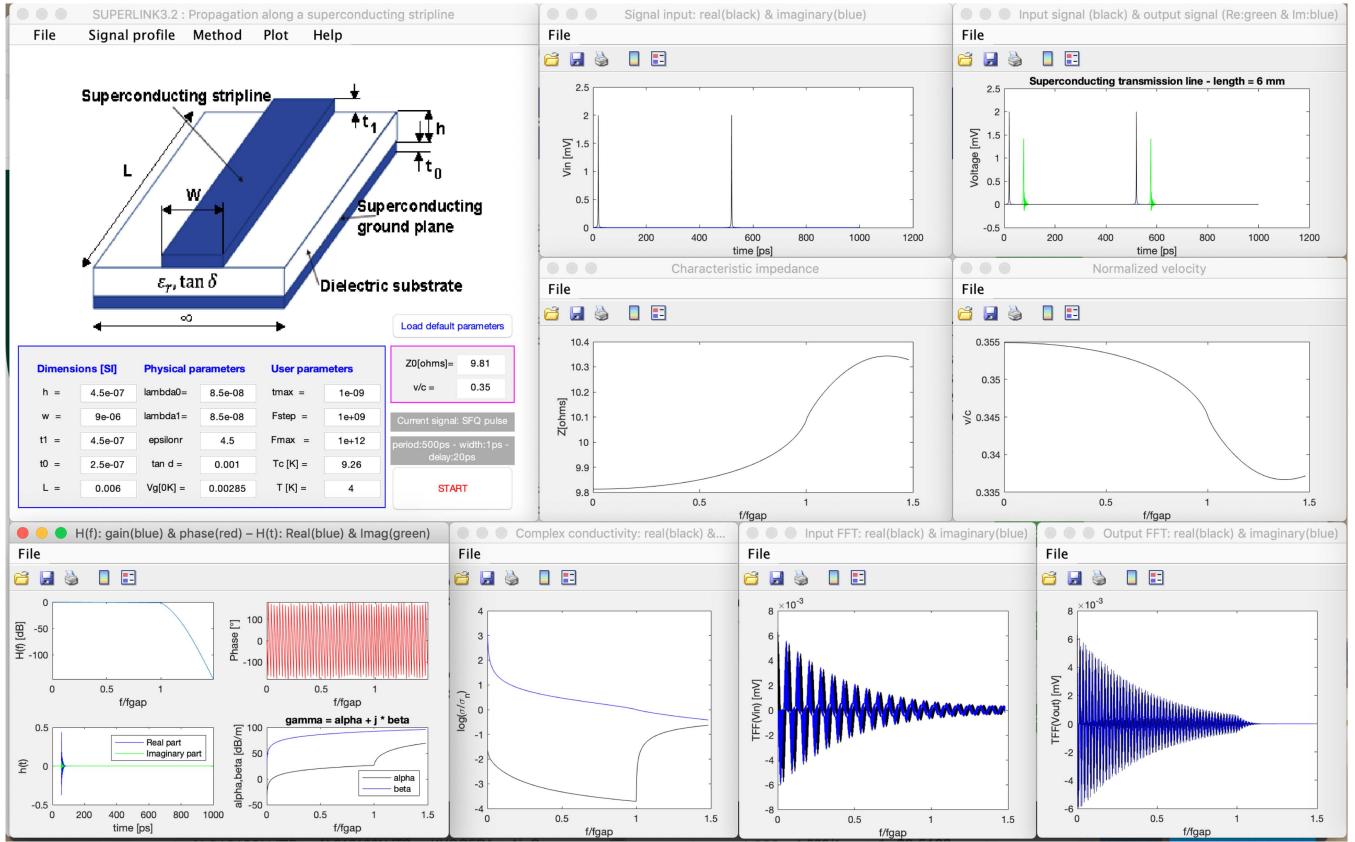


Fig. 22. SuperLink GUI for macOS, along with displayed results. All the results can be saved in ASCII files for postprocessing.

TABLE X
TECHNICAL FOCUS AREA 2: ANALOG DESIGN AND SYNTHESIS

Milestones/Tasks	Figure of Merit	Phase 1	Phase 2A	Phase 2B	Phase 3	ColdFlux result
Circuit simulator, layout synthesis tools, and timing, yield and power analysis tools	Design complexity (JJ count)	≥ 10 ⁴ JJs	≥ 3 × 10 ⁴ JJs	≥ 10 ⁵ JJs	≥ 10 ⁶ JJs	≥ 10 ⁷ JJs
	Clock frequency, maximum (GHz)	≥ 20	≥ 30	≥ 50	≥ 100	≥ 100

TABLE XI
TECHNICAL FOCUS AREA 3: PHYSICAL DESIGN AND VERIFICATION

Milestones/Tasks	Figure of Merit	Phase 1	Phase 2A	Phase 2B	Phase 3	ColdFlux result
Automated place-and-route (P&R) tools, circuit optimization tools, and verification tools	Device parameter extraction types, tolerance band (%), fraction	L, ± 20%, 0.9	L, ± 15%, 0.9	L, M, R, J, ± 10%, 0.9	L, M, R, J, C, Z, ± 10%, 0.95	L, M, R, J, C, Z, ± 10%, ≥ 0.95
	Circuit parameter extraction device count, area [μm^2]	-	-	≥ 30, ≥ 5000 μm^2	≥ 60, ≥ 10 000 μm^2	≥ 1000, ≥ 100 000 μm^2
	P&R timing tolerance (%)	-	-	< 10%	< 5%	< 5%
	P&R interconnect area, area reduction (relative to baseline)	-	-	≤ 1 cm ² baseline	≤ 4 cm ² 20%	≤ 4 cm ² 20%

X. CONCLUSION

The ColdFlux project fired up a research and development effort that spanned continents and brought together groups with different fields of experience. A complete toolchain was developed that was used in-house to design cell libraries and

synthesize large-scale digital systems. The toolchain and its use scenarios have been widely disseminated in publications [8], [139], [140].

The ColdFlux project had a deep academic impact and delivered 30 Ph.D. degrees and 17 master's degrees. A further 15 Ph.D. students, eight master's students, and nine undergraduate

students were supported by and contributed to the ColdFlux project. At the time of writing, 118 journal articles and conference papers had already been published on this work, with more in progress.

Since the conclusion of the ColdFlux project, we have continued to maintain the tools. The goal is for us to maintain the tools and improve the tools' interface based on feedback from users and through follow-up projects. We are also planning to add new tools in future, especially for the qPALACE and InductEx tool suites.

Under qPALACE, the ability of multiphase and dual clocking will be included in synthesis and place and routing tools. The synthesis tool qYosys will support more advanced Verilog features, and the routing tool and hold-time violation tool will be improved to support more complex circuits. The expanded qPALACE tool will also support compound cells for dense and high-throughput logic circuits. The tools will be used for designing large-scale circuits such as neural network accelerators, Josephson-based Ising machines, and modular multipliers for fully homomorphic cryptography.

Planned additions to the InductEx tool suite include a GUI and support for the design and analysis of more complicated analog structures with a focus on resonators in quantum systems, and on SQUID arrays. We also intend to expand the JoSIM to include more JJ models and lossy transmission lines.

Other future improvements not yet planned include true 3-D meshing for the TCAD tools and support for clockless dynamic SFQ circuits [141], [142] with resistive loops in TimEx.

ACKNOWLEDGMENT

The authors would like to thank D.S. Holmes for general comments, feedback, and suggestions on improvements over almost all aspects of the toolchain, as well as M. Heiligman, M. Sheu, and J. Sirevicius for driving the SuperTools program. The authors would also like to thank the test and evaluation partners for valuable inputs to tool development and debugging and for performing experimental tests that validated the tools or indicated directions for improvement, in particular, M. Castellanos-Beltran, A. Sirois, P. Hopkins, P. Dresselhaus, and D. Olaya from the NIST for performing the measurements and for their valuable feedback, A. Wynn and S. Tolpygo from the MITLL for fabrication and valuable discussions on magnetic rule checking, J. Shalf, G. Michelogiannakis, D. Vasudevan, D. Lyles, A. Butko, and F. Fatollahi-Fard from the LBNL for valuable feedback on synthesis tools, and T. J. Mannos, M. P. Frank, S. B. Kaplan, R. M. Lewis, J. Rose, and S. J. Valancius from Sandia National Laboratories for parameter extraction verification and feedback. For their work on the development and verification of ColdFlux tools and library cells, guiding students on the project or providing in-house counsel, the authors would like to thank M. S. Abrishami, R. S. Bakolo, M. M. Botha, F. China, E. S. Cishugi, J. A. Coetzee, H. Cong, A. P. Davies, J. F. De Villiers, H. F. Herbst, Y. Hironaka, D. Ito, N. Katam, F. Ke, P. Le Roux, H. Li, T.-R. Lin, W. Luo, C. C. Maree, K. Meyer, I. Mohanty, S. Nazar Shahsavani, M. Nazemi, B. A. P. Nel, G. Pasandi, E. Patrick, P. J. Peiser, N. Pohkrel, R. Saito,

H. Shen, R. Tadros, T. Tanaka, R. Van Staden, M. Sun, B. H. Venter, E. Verburg, F. Wang, R. Yahia, U. Yilmaz, and B. Zhang. For outside counsel, the authors would like to thank W. Hunt, O. Mukhanov, V. Semenov, and I. Sutherland. Lucas Iwanikow would like to thank the Centre National d'Etudes Spatiales, the Agence de l'Innovation de Défense, and the Direction Générale de l'Armement for his Ph.D. thesis support.

REFERENCES

- [1] IARPA SuperTools Program, 2016. [Online]. Available: <https://www.iarpa.gov/index.php/research-programs/supertools>
- [2] C. J. Fourie and M. H. Volkmann, "Status of superconductor electronic circuit design software," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, Jun. 2013, Art. no. 1300205.
- [3] C. J. Fourie, "Digital superconducting electronics design tools—Status and roadmap," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 5, Aug. 2018, Art. no. 1300412.
- [4] M. A. Manheimer, "Cryogenic computing complexity program: Phase 1 introduction," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 3, Jun. 2015, Art. no. 1301704.
- [5] A. Inamdar, S. S. Meher, B. Chonigman, A. Sahu, J. Ravi, and D. Gupta, "50 GHz operation of RSFQ arithmetic logic unit designed using the advanced design flow and the dual RSFQ/ERSFQ cell library," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1300908.
- [6] ColdFlux Repository. [Online]. Available: <https://coldflux.usc.edu/protected-content/>
- [7] 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. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1302008.
- [8] C. J. Fourie et al., "ColdFlux superconducting EDA and TCAD tools project: Overview and progress," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300407.
- [9] KLayout. [Online]. Available: <https://www.klayout.de>
- [10] M. Pedram, "qPALACE: A suite of EDA tools for synthesis and physical design optimization of single flux quantum logic circuits," in *Proc. 33rd Int. Symp. Supercond.*, 2020.
- [11] G. Pasandi, A. Shafaei, and M. Pedram, "SFQmap: A technology mapping tool for single flux quantum logic circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.
- [12] G. Pasandi and M. Pedram, "PBMap: A path balancing technology mapping algorithm for single flux quantum logic circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 4, Jun. 2019, Art. no. 1300114.
- [13] G. Pasandi and M. Pedram, "Balanced factorization and rewriting algorithms for synthesizing single flux quantum logic circuits," in *Proc. Great Lakes Symp. VLSI*, 2019, pp. 183–188.
- [14] 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 Des.*, 2019, pp. 1–8.
- [15] C. L. Ayala et al., "A semi-custom design methodology and environment for implementing superconductor adiabatic quantum-flux-parametron microprocessors," *Supercond. Sci. Technol.*, vol. 33, no. 5, 2020, Art. no. 054006.
- [16] N. Takeuchi et al., "Adiabatic quantum-flux-parametron cell library designed using a 10 kA cm⁻² niobium fabrication process," *Supercond. Sci. Technol.*, vol. 30, no. 3, 2017, Art. no. 035002.
- [17] 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. Supercond.*, vol. 26, no. 8, Dec. 2016, Art. no. 1301805.
- [18] 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. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1301905.
- [19] Y. Murai, C. L. Ayala, N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Development and demonstration of routing and placement EDA tools for large-scale adiabatic quantum-flux-parametron circuits," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 6, Sep. 2017, Art. no. 1302209.

- [20] 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. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1301706.
- [21] C. L. Ayala, T. Tanaka, R. Saito, M. Nozoe, N. Takeuchi, and N. Yoshikawa, "MANA: A monolithic adiabatic iNtegration architecture microprocessor using 1.4-zJ/op unshunted superconductor Josephson junction devices," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1152–1165, Apr. 2021.
- [22] R. Cai et al., "A majority logic synthesis framework for adiabatic quantum-flux-parametron superconducting circuits," in *Proc. Great Lakes Symp. VLSI*, 2019, pp. 189–194.
- [23] R. Cai et al., "IDE development, logic synthesis and buffer/splitter insertion framework for adiabatic quantum-flux-parametron superconducting circuits," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2019, pp. 187–192.
- [24] C. L. Ayala, N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa, "Majority-logic-optimized parallel prefix carry look-ahead adder families using adiabatic quantum-flux-parametron logic," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1300407.
- [25] Y.-C. Chang, H. Li, O. Chen, Y. Wang, N. Yoshikawa, and T.-Y. Ho, "ASAP: An analytical strategy for AQFP placement," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2020, pp. 1–7.
- [26] P. Dong et al., "TAAS: A timing-aware analytical strategy for AQFP-Capable placement automation," in *Proc. IEEE/ACM 59th Des. Autom. Conf.*, 2022, pp. 1321–1326.
- [27] R. Saito, C. L. Ayala, O. Chen, T. Tanaka, T. Tamura, and N. Yoshikawa, "Logic synthesis of sequential logic circuits for adiabatic quantum-flux-parametron logic," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1301405.
- [28] A. Fayyazi, M. Munir, A. Gopikanna, S. Nazarian, and M. Pedram, "A logic verification framework for SFQ and AQFP superconducting circuits," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 9, Dec. 2021, Art. no. 1303211.
- [29] 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. 20th Int. Symp. Qual. Electron. Des.*, 2019, pp. 224–230.
- [30] S. Nazarian, A. Fayyazi, and M. Pedram, "qCG: A low-power multi-domain SFQ logic design and verification framework," in *Proc. IEEE 37th Int. Conf. Comput. Des.*, 2019, pp. 446–449.
- [31] R. N. Tadros, A. Fayyazi, M. Pedram, and P. A. Beerel, "Systemverilog modeling of SFQ and AQFP circuits," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 2, Mar. 2020, Art. no. 1300513.
- [32] A. Fayyazi, S. Nazarian, and M. Pedram, "Logic verification of ultra-deep pipelined beyond-CMOS technologies," 2020, *arXiv:2005.13735*.
- [33] M. Munir, A. Gopikanna, A. Fayyazi, M. Pedram, and S. Nazarian, "qMC: A formal model checking verification framework for superconducting logic," in *Proc. Great Lakes Symp. VLSI*, 2021, pp. 259–264.
- [34] H. Zha, N. K. Katam, M. Pedram, and M. Annavaram, "HiPerRF: A dual-bit dense storage SFQ register file," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit.*, 2022, pp. 415–428.
- [35] N. K. Katam, H. Zha, M. Pedram, and M. Annavaram, "Multi fluxon storage and its implications for microprocessor design," *J. Phys.: Conf. Ser.*, vol. 1559, no. 1, 2020, Art. no. 012004.
- [36] V. K. Semenov, A. A. Odintsov, and A. B. Zorin, "Automation of numerical analysis of circuits with Josephson tunnel junctions," in *Superconducting Quantum Interference Devices and Their Applications*. Berlin, Germany: De Gruyter, 1986, pp. 71–76.
- [37] S. Polonsky, V. Semenov, and P. Shevchenko, "PSCAN: Personal superconductor circuit analyser," *Supercond. Sci. Technol.*, vol. 4, no. 11, 1991, Art. no. 667.
- [38] 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. Supercond.*, vol. 7, no. 2, pp. 2685–2689, Jun. 1997.
- [39] P. Shevchenko, "PSCAN2—Superconductor circuit simulator," 2020. [Online]. Available: <http://pscan2sim.org/>
- [40] W. C. Stewart, "Current-voltage characteristics of Josephson junctions," *Appl. Phys. Lett.*, vol. 12, no. 8, pp. 277–280, 1968.
- [41] D. E. McCumber, "Effect of ac impedance on dc voltage-current characteristics of superconductor weak-link junctions," *J. Appl. Phys.*, vol. 39, no. 7, pp. 3113–3118, 1968.
- [42] S. Whiteley, "Josephson junctions in SPICE3," *IEEE Trans. Magn.*, vol. 27, no. 2, pp. 2902–2905, Mar. 1991.
- [43] S. Whiteley, "WRspice," Feb. 2020. [Online]. Available: <http://www.wrcad.com/wrspice.html>
- [44] E. S. Fang, "A Josephson integrated circuit simulator (JSIM) for superconductive electronics application," in *Proc. Extended Abstr. Int. Supercond. Electron. Conf.*, 1989, pp. 407–410.
- [45] J. Satchell, "Stochastic simulation of SFQ logic," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 3315–3318, Jun. 1997.
- [46] J. A. Delport, "Simulation and verification software for superconducting electronic circuits," Ph.D. dissertation, Stellenbosch Univ., Stellenbosch, South Africa, 2019.
- [47] J. A. Delport, K. Jackman, P. Le Roux, and C. J. Fourie, "JoSIM—superconductor SPICE simulator," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300905.
- [48] T. Hall, J. A. Delport, and C. J. Fourie, "Determination of the bit error rate due to thermal noise using JoSIM superconducting circuit simulator and the Monte Carlo method," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1302505.
- [49] N. B. Ferrante, S. A. E. Berggren, S. T. Crowe, and B. J. Taylor, "Linearity improvements and large inductance spread methods in SQA design," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1600903.
- [50] JoSIM. [Online]. Available: <https://github.com/JoeyDelp/JoSIM/>
- [51] R. van Staden, J. A. Delport, J. A. Coetzee, and C. J. Fourie, "Layout versus schematic with design/magnetic rule checking for superconducting integrated circuit layouts," in *Proc. IEEE Int. Supercond. Electron. Conf.*, 2019, pp. 1–3.
- [52] SPIRA. [Online]. Available: <https://github.com/rubenvanstaden/spira>
- [53] S. Razmkhah and P. Febvre, "JOINUS: A user-friendly open-source software to simulate digital superconductor circuits," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 5, Aug. 2020, Art. no. 1300807.
- [54] C. J. Fourie, O. Wetzstein, T. Ortlepp, and J. Kunert, "Three-dimensional multi-terminal superconductive integrated circuit inductance extraction," *Supercond. Sci. Technol.*, vol. 24, no. 12, 2011, Art. no. 125015.
- [55] C. J. Fourie, O. Wetzstein, J. Kunert, H. Toepfer, and H.-G. Meyer, "Experimentally verified inductance extraction and parameter study for superconductive integrated circuit wires crossing ground plane holes," *Supercond. Sci. Technol.*, vol. 26, no. 1, 2013, Art. no. 015016.
- [56] C. J. Fourie, O. Wetzstein, J. Kunert, and H.-G. Meyer, "SFQ circuits with ground plane hole-assisted inductive coupling designed with InductEx," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, Jun. 2013, Art. no. 1300705.
- [57] C. J. Fourie, "Full-gate verification of superconducting integrated circuit layouts with InductEx," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 1, Feb. 2015, Art. no. 1300209.
- [58] S. N. Shahsavani, B. Zhang, and M. Pedram, "A timing uncertainty-aware clock tree topology generation algorithm for single flux quantum circuits," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, 2020, pp. 278–281.
- [59] S. N. Shahsavani and M. Pedram, "A minimum-skew clock tree synthesis algorithm for single flux quantum logic circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 8, Dec. 2019, Art. no. 1303513.
- [60] 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. Des., Autom. Test Eur. Conf. Exhib.*, 2018, pp. 1465–1468.
- [61] G. Pasandi and M. Pedram, "An efficient pipelined architecture for superconducting single flux quantum logic circuits utilizing dual clocks," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 2, Mar. 2020, Art. no. 1300412.
- [62] M. Li, B. Zhang, and M. Pedram, "Striking a good balance between area and throughput of RSFQ circuits containing feedback loops," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1302606.
- [63] S. N. Shahsavani and M. Pedram, "TDP-ADMM: A timing driven placement approach for superconductive electronic circuits using alternating direction method of multipliers," in *Proc. IEEE/ACM 57th Des. Autom. Conf.*, 2020, pp. 1–6.
- [64] X. Li, M. Pan, T. Liu, and P. A. Beerel, "Multi-phase clocking for multi-threaded gate-level-pipelined superconductive logic," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2022, pp. 62–67.
- [65] N. Takeuchi, M. Nozoe, Y. He, and N. Yoshikawa, "Low-latency adiabatic superconductor logic using delay-line clocking," *Appl. Phys. Lett.*, vol. 115, no. 7, 2019, Art. no. 072601.
- [66] Y. He, N. Takeuchi, and N. Yoshikawa, "Low-latency power-dividing clocking scheme for adiabatic quantum-flux-parametron logic," *Appl. Phys. Lett.*, vol. 116, no. 18, 2020, Art. no. 182602.
- [67] R. Saito, C. L. Ayala, and N. Yoshikawa, "Buffer reduction via n-phase clocking in adiabatic quantum-flux-parametron benchmark circuits," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 6, Sep. 2021, Art. no. 1302808.

- [68] Y. He et al., "Low clock skew superconductor adiabatic quantum-flux-parametron logic circuits based on grid-distributed blocks," *Supercond. Sci. Technol.*, vol. 36, no. 1, 2022, Art. no. 015006.
- [69] R. Yang, S. Yang, J. Ren, X. Gao, W. Yan, and Z. Wang, "A local optimization method for single flux quantum logic circuits design utilizing synchronizer IPs," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 3, Apr. 2023, Art. no. 1301308.
- [70] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 1, no. 1, pp. 63–71, Mar. 1993.
- [71] M. Li, F. Wang, and S. Gupta, "Data-driven fault model development for superconducting logic," in *Proc. IEEE Int. Test Conf.*, 2020, pp. 1–5.
- [72] F. Wang and S. Gupta, "Automatic test pattern generation for timing verification and delay testing of RSFQ circuits," in *Proc. IEEE 37th VLSI Test Symp.*, 2019, pp. 1–6.
- [73] F. Wang and S. K. Gupta, "An effective and efficient automatic test pattern generation (ATPG) paradigm for certifying performance of RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 5, Aug. 2020, Art. no. 1300711.
- [74] F. Wang and S. Gupta, "Timing verification for rapid single-flux-quantum (RSFQ) logic: New paradigm and models," in *Proc. IEEE Int. Supercond. Electron. Conf.*, 2019, pp. 1–3.
- [75] F. Wang, B. Zhang, M. Pedram, and S. Gupta, "Static timing analysis (STA) with timing bleed: Certifying much higher performance for rapid single flux quantum (RSFQ) logic," *J. Phys.: Conf. Ser.*, vol. 1559, no. 1, 2020, Art. no. 012003.
- [76] M. Li, F. Wang, and S. Gupta, "Methods for testing path delay and static faults in RSFQ circuits," in *Proc. IEEE 40th VLSI Test Symp.*, 2022, pp. 1–7.
- [77] M. Li, Y. Lin, and S. Gupta, "Design for testability for RSFQ circuits," in *Proc. IEEE 41st VLSI Test Symp.*, 2023, pp. 1–7.
- [78] F. Wang and S. Gupta, "Multi-cell characterization: Developing robust cells and abstraction for rapid single flux quantum (RSFQ) logic," in *Proc. IEEE Int. Test Conf.*, 2019, pp. 1–10.
- [79] C. J. Fourie, "Extraction of dc-biased SFQ circuit Verilog models," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 6, Sep. 2018, Art. no. 1300811.
- [80] TimEx. [Online]. Available: <https://github.com/sunmagnetics/TimEx>
- [81] B. Zhang, M. Li, and M. Pedram, "qSSTA: A statistical static timing analysis tool for superconducting single-flux-quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 7, Oct. 2020, Art. no. 1301612.
- [82] B. Zhang, F. Wang, S. Gupta, and M. Pedram, "A statistical static timing analysis tool for superconducting single-flux-quantum circuits," in *Proc. IEEE Int. Supercond. Electron. Conf.*, 2019, pp. 1–5.
- [83] C. L. Ayala, O. Chen, and N. Yoshikawa, "AQFPTX: Adiabatic quantum-flux-parametron timing extraction tool," in *Proc. IEEE Int. Supercond. Electron. Conf.*, 2019, pp. 1–3.
- [84] C. L. Ayala et al., "Timing extraction for logic simulation of VLSI adiabatic quantum-flux-parametron circuits," *IEICE Tech. Rep.*, vol. 115, no. 242, pp. 7–12, 2015.
- [85] M. A. Karamuftuoglu, S. N. Shahsavani, and M. Pedram, "Margin and yield optimization of single flux quantum logic cells using swarm optimization techniques," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 1, Jan. 2023, Art. no. 1300110.
- [86] M. A. Karamuftuoglu, S. Nazar Shahsavani, and M. Pedram, *Margin Optimization of Single Flux Quantum Logic Cells*. Cham, Switzerland: Springer, 2023, pp. 105–133.
- [87] P. le Roux and C. J. Fourie, "Distance-to-failure-maximization optimization algorithm for SFQ logic cells," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 7, Oct. 2020, Art. no. 1301405.
- [88] N. Pokhrel, T. Weingartner, R. J. Burwell, E. E. Patrick, and M. E. Law, "Simulating the fabrication of Nb/Al-O/Nb Josephson junction for superconductive electronics application," in *Proc. IEEE Int. Supercond. Electron. Conf.*, 2019, pp. 1–4.
- [89] T. Weingartner, N. Pokhrel, M. Sulangi, L. Bjorndal, E. Patrick, and M. E. Law, "Modeling process and device behavior of Josephson junctions in superconductor electronics with TCAD," *IEEE Trans. Electron Devices*, vol. 68, no. 11, pp. 5448–5454, Nov. 2021.
- [90] L. Bjorndal, N. Pokhrel, T. Weingartner, P. Leger, E. Patrick, and M. Law, "Modeling the effects of niobium surface roughness on electrical conductivity of Nb/Al-AlO_x/Nb Josephson junctions," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1100805.
- [91] M. A. Sulangi, T. A. Weingartner, N. Pokhrel, E. Patrick, M. Law, and P. J. Hirschfeld, "Disorder and critical current variability in Josephson junctions," *J. Appl. Phys.*, vol. 127, no. 3, Jan. 2020, Art. no. 033901.
- [92] M. A. Sulangi et al., "Critical currents in conventional Josephson junctions with grain boundaries," *J. Appl. Phys.*, vol. 130, no. 14, Oct. 2021, Art. no. 143901.
- [93] L. Schindler, "The development and characterisation of a parameterised RSFQ cell library for layout synthesis," Ph.D. dissertation, Dept. Electr. Electron. Eng., Stellenbosch Univ., Stellenbosch, South Africa, 2021.
- [94] L. Schindler and C. J. Fourie, "Application of phase-based circuit theory to RSFQ logic design," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 3, Apr. 2022, Art. no. 1300512.
- [95] C. J. Fourie, "Inductance in superconductor integrated circuits," D.Eng. dissertation, Dept. Electr. Electron. Eng., Stellenbosch Univ., Stellenbosch, South Africa, 2023.
- [96] L. Schindler, J. A. Delport, and C. J. Fourie, "The ColdFlux RSFQ cell library for MIT-LL SFQ5ec fabrication process," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 2, Mar. 2022, Art. no. 1300207.
- [97] P. le Roux, K. Jackman, J. A. Delport, and C. J. Fourie, "Modeling of superconducting passive transmission lines," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1101605.
- [98] L. Schindler, P. le Roux, and C. J. Fourie, "Impedance matching of passive transmission line receivers to improve reflections between RSFQ logic cells," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 2, Mar. 2020, Art. no. 1300607.
- [99] H. F. Herbst, P. Le Roux, K. Jackman, and C. J. Fourie, "Improved transmission line parameter calculation through TCAD process modeling for superconductor integrated circuit interconnects," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 7, Oct. 2020, Art. no. 1100504.
- [100] P. le Roux, C. Fourie, S. Razmkhah, and P. Febvre, "Accurate small signal simulation of superconductor interconnects in SPICE," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1303006.
- [101] S. K. Tolpygo et al., "Advanced fabrication processes for superconducting very large-scale integrated circuits," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 3, Apr. 2016, Art. no. 1100110.
- [102] S. K. Tolpygo et al., "Advanced fabrication processes for superconductor electronics: Current status and new developments," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1102513.
- [103] C. J. Fourie, C. L. Ayala, L. Schindler, T. Tanaka, and N. Yoshikawa, "Design and characterization of track routing architecture for RSFQ and AQFP circuits in a multilayer process," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 6, Aug. 2020, Art. no. 1301109.
- [104] ColdFlux RSFQ cell library v3.0. [Online]. Available: <https://github.com/sunmagnetics/RSFQLib/tree/master/RSFQLib>
- [105] S. S. Meher et al., "Superconductor standard cell library for advanced EDA design," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1300807.
- [106] A. Inamdar et al., "Development of superconductor advanced integrated circuit design flow using Synopsys tools," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1301907.
- [107] Y. He et al., "A compact AQFP logic cell design using an 8-metal layer superconductor process," *Supercond. Sci. Technol.*, vol. 33, no. 3, 2020, Art. no. 035010.
- [108] L. Schindler, C. L. Ayala, K. Jackman, C. J. Fourie, and N. Yoshikawa, "Adopting a standard track routing architecture for next-generation hybrid AC/DC-biased logic circuits," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1302905.
- [109] B. A. P. Nel and M. M. Botha, "MLACA with modified grouping strategy for efficient superconducting circuit analysis," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1100705.
- [110] B. A. P. Nel and M. M. Botha, "An efficient MLACA-SVD solver for superconducting integrated circuit analysis," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 7, Oct. 2019, Art. no. 1303310.
- [111] M. Kamon, M. J. Tsuk, and J. K. White, "FastHenry: A multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microw. Theory Techn.*, vol. 42, no. 9, pp. 1750–1758, Sep. 1994.
- [112] K. Jackman and C. J. Fourie, "Multipole accelerated magnetic field calculations for superconducting circuits," *Supercond. Sci. Technol.*, vol. 32, no. 1, 2019, Art. no. 015011.
- [113] K. Jackman and C. J. Fourie, "Tetrahedral modeling method for inductance extraction of complex 3-D superconducting structures," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 3, Apr. 2016, Art. no. 0602305.
- [114] K. Jackman and C. J. Fourie, "Fast multicore FastHenry and a tetrahedral modeling method for inductance extraction of complex 3D geometries," in *Proc. 15th Int. Supercond. Electron. Conf.*, 2015, pp. 1–3.
- [115] C. J. Fourie, C. Shawareh, I. V. Vernik, and T. V. Filippov, "High-accuracy InductEx calibration sets for MIT-LL SFQ4ee and SFQ5ee processes," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 2, Mar. 2017, Art. no. 1300805.

- [116] S. K. Tolpygo, E. B. Golden, T. J. Weir, and V. Bolkhovsky, "Mutual and self-inductance in planarized multilayered superconductor integrated circuits: Microstrips, striplines, bends, meanders, ground plane perforations," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 5, Aug. 2022, Art. no. 1400331.
- [117] C. J. Fourie and K. Jackman, "High-fidelity circuit simulation of AQFP circuits through compact models extracted from layout," *J. Phys.: Conf. Ser.*, vol. 2323, no. 1, Aug. 2022, Art. no. 012034.
- [118] N. Werthamer, "Nonlinear self-coupling of Josephson radiation in superconducting tunnel junctions," *Phys. Rev.*, vol. 147, no. 1, 1966, Art. no. 255.
- [119] A. Larkin and Y. N. Ovchinnikov, "Tunnel effect between superconductors in an alternating field," *Sov. Phys. JETP*, vol. 24, no. 11, pp. 1035–1040, 1967.
- [120] R. E. Harris, "Cosine and other terms in the Josephson tunneling current," *Phys. Rev. B*, vol. 10, no. 1, 1974, Art. no. 84.
- [121] L. Iwanikow and P. Febvre, "Time-domain simulator of Josephson junctions based on the BCS theory," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1800605.
- [122] C. J. Fourie and K. Jackman, "Software tools for flux trapping and magnetic field analysis in superconducting circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1301004.
- [123] K. Jackman and C. J. Fourie, "Flux trapping experiments to verify simulation models," *Supercond. Sci. Technol.*, vol. 33, no. 10, 2020, Art. no. 105001.
- [124] C. J. Fourie and K. Jackman, "Experimental verification of moat design and flux trapping analysis," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1300507.
- [125] C. J. Fourie, N. Takeuchi, K. Jackman, and N. Yoshikawa, "Evaluation of flux trapping moat position on AQFP cell performance," *J. Phys.: Conf. Ser.*, vol. 1975, no. 1, 2021, Art. no. 012027.
- [126] J. Lei et al., "Design and implementation of two-stage time-to-digital converter based on rapid single-flux-quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 2, Mar. 2023, Art. no. 1100107.
- [127] M. Roncken, E. Esimai, V. Ramanathan, W. Hunt, and I. Sutherland, "State access for RSFQ test and analysis," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1303907.
- [128] R. Cheng, C. Kirst, and D. Vasudevan, "Superconducting-oscillatory neural network with pixel error detection for image recognition," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1801107.
- [129] A. E. Madden, J. C. Willard, R. Loloe, and N. O. Birge, "Phase controllable Josephson junctions for cryogenic memory," *Supercond. Sci. Technol.*, vol. 32, no. 1, Nov. 2018, Art. no. 015001.
- [130] Y. Somei, H. Shimada, and Y. Mizugaki, "Enhanced operation frequencies of bipolar double-flux-quantum amplifiers fabricated using 10 kA cm^{-2} Nb/A_xNb integration process," *Jpn. J. Appl. Phys.*, vol. 60, no. 7, 2021, Art. no. 073001.
- [131] Q. Zhang et al., "Geometric dependence of washer inductance for NbN DC SQUIDS," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 7, Oct. 2018, Art. no. 1601704.
- [132] A. Labb   et al., "Effects of flux pinning on the DC characteristics of meander-shaped superconducting quantum interference filters with flux concentrator," *J. Appl. Phys.*, vol. 124, no. 21, 2018, Art. no. 214503.
- [133] H. Li et al., "Principle and experimental investigation of current-driven negative-inductance superconducting quantum interference device," *Supercond. Sci. Technol.*, vol. 30, no. 3, 2017, Art. no. 035012.
- [134] S. Kempf et al., "Demonstration of a scalable frequency-domain readout of metallic magnetic calorimeters by means of a microwave SQUID multiplexer," *AIP Adv.*, vol. 7, no. 1, 2017, Art. no. 015007.
- [135] S. Kempf et al., "Design, fabrication and characterization of a 64 pixel metallic magnetic calorimeter array with integrated, on-chip microwave squid multiplexer," *Supercond. Sci. Technol.*, vol. 30, no. 6, 2017, Art. no. 065002.
- [136] D. Xu et al., "Low-noise second-order gradient SQUID current sensors overlap-coupled with input coils of different inductances," *Supercond. Sci. Technol.*, vol. 35, no. 8, 2022, Art. no. 085004.
- [137] Y.-P. Pan et al., "Characterization of scanning SQUID probes based on 3D nano-bridge junctions in magnetic field," *Chin. Phys. Lett.*, vol. 37, no. 8, 2020, Art. no. 080702.
- [138] Y. He, S. Michibayashi, N. Takeuchi, and N. Yoshikawa, "Sharp-selectivity in-line topology low temperature superconducting bandpass filter for superconducting quantum applications," *Supercond. Sci. Technol.*, vol. 33, no. 3, Feb. 2020, Art. no. 035012.
- [139] C. J. Fourie, "Electronic design automation tools for superconducting circuits," *J. Phys.: Conf. Ser.*, vol. 1590, no. 1, 2020, Art. no. 012040.
- [140] C. Pegrum, "Modelling high- T_C electronics," *Supercond. Sci. Technol.*, vol. 36, no. 5, 2023, Art. no. 053001.
- [141] S. V. Rylov, "Clockless dynamic SFQ and gate with high input skew tolerance," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300805.
- [142] S. Rylov et al., "Superconducting VLSI logic cell library using dc-powered clockless dynamic SFQ gates and ASIC-style layout template," *IEEE Trans. Appl. Supercond.*, vol. 33, no. 5, Aug. 2023, Art. no. 1303207.