Tools for the Computer-Aided Design of Multigigahertz Superconducting Digital Circuits

Kris Gaj, Quentin P. Herr, Victor Adler, Andy Krasniewski, *Fellow, IEEE*, Eby G. Friedman, and Marc J. Feldman

***Abstract*— The realization of large integrated circuits depends upon the application of computer-aided design (CAD) tools. This paper summarizes the results of a survey of CAD tools targeting superconducting digital electronics. Five categories of tools: circuit simulators, circuit optimizers, layout tools, inductance estimators, and logic simulators are discussed in detail. Within each category, a comparison of several currently available CAD tools is presented, and a tool which has been adapted for use or developed at the University of Rochester is discussed in greater detail. In addition, tools for timing analysis as well as integrated design environments that permit the effective data interchange among various tools and support libraries of design models are discussed. Future tools for timing optimization, automated logic synthesis, and automated layout synthesis are shown to be necessary for the design of superconducting circuits at the very large scale of integration (VLSI) level of integration. Trends regarding changes in the requirements for effective CAD tools are discussed, and expected improvements to existing tools and features of new tools currently under development are presented.**

***Index Terms*— CAD, inductance extraction, layout, optimization, RSFQ, simulation, superconducting electronics.**

I. Introduction

SUPERCONDUCTING digital electronics (SDE) compares favorably with all existing semiconductor technologies when both speed and power consumption are considered [1], [2]. A major effort today in SDE focuses upon *Rapid Single Flux Quantum (RSFQ)* logic. Medium- to large-scale RSFQ circuits have been reported to operate with clock frequencies above 10 GHz, while 100-GHz RSFQ circuits should be feasible within the foreseeable future.

The complexity of semiconductor very large scale of integration (VLSI) circuitry is a wonder of modern technology. Tens of millions of active transistors and additional passive elements are systematically organized and fashioned such that the overall result of their individual operation provides a unified and coordinated function. This complexity is far beyond the ability of a human mind to comprehend in all its detail. Rather, the development of semiconductor VLSI circuits has been made possible by utilizing a large variety of tools and methodologies which fall under the general name of *computer-aided design (CAD)*. The development of CAD tools and design methodologies is a major research specialty in the field of semiconductor integrated circuits.

The current situation in *superconducting* digital electronics CAD is far less advanced. The level of integration of RSFQ circuits, although primarily limited by the relative immaturity of the process technology, is also distinctly limited by the available design tools. Future progress will depend on the development of more sophisticated CAD tools targeted to SDE. These tools are the subject of this paper; a companion paper treats design methodologies for RSFQ circuits [3].

All superconducting logic schemes can be roughly classified as either flux-based, such as RSFQ, or voltage-state, such as MVTL. Voltage-state logic is a natural emulation of semiconductor technology in that data is encoded by steady voltage levels. RSFQ logic is a new concept in which logic states "zero" and "one" are encoded using voltage pulses instead of voltage levels.

Conventional CAD tools used in the semiconductor industry cannot be directly applied to the design of superconducting digital circuits. At the circuit and physical levels, the main obstacles include a different basic active component (transistor for semiconductor logic families versus Josephson junction for superconducting technology), a different basic passive component (capacitor versus inductor), and different passive and active interconnects (metal RC lines with buffers versus Josephson transmission lines and microstrips). At the logic and system levels the main problems include a different suite of basic gates, different synchronization schemes, and significantly different levels of influence of gates on each other. For SFQ-based logic, an additional problem is a different convention for representation of logic states.

Although certainly there are difficulties to adapting semiconductor CAD tools for voltage-state superconducting logic, there are many more complications in adapting them for RSFQ. Several complete integrated design environments have been developed for superconducting voltage-state logic, for instance [4] and [5], and there are many examples of circuits consisting of approximately 20 000 Josephson junctions which have been demonstrated [4].

The development of RSFQ logic began with the intuitive design of basic gates [6]–[9] and only recently has reached a

Manuscript received April 25, 1998; revised October 30, 1998.

K. Gaj is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA.

Q. P. Herr is with TRW Space & Electronics Group, D1/1039, Redondo Beach, CA 90278 USA.

V. Adler is with Sun Microsystems, Palo Alto, CA 94303 USA.

A. Krasniewski is with the Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland.

E. G. Friedman, and M. J. Feldman are with Department of Electrical Engineering, University of Rochester, Rochester, NY 14627 USA.

Publisher Item Identifier S 1051-8223(99)03354-0.

level of complexity where circuits with many tens of RSFQ gates are successfully demonstrated at multigigahertz clock frequencies [10]–[14]. The initial progress utilized tools and methods developed specifically for RSFQ logic, which allow minimal automation. Together with the increasing complexity and growing variety of RSFQ circuits currently under development, design methods have evolved and new tools have been created to satisfy this need.

In the early stages of the development of RSFQ logic (1985–1992), the only tools in use were a *circuit simulator* (such as JSpice [15], PSCAN [16], or JSim) and a simple *layout editor* (such as AutoCAD or Magic). Beginning in 1993, *circuit optimizers* (such as MALT [17]) and *inductance estimators* (such as Lmeter [18]) were introduced to increase the yield of superconducting circuits and accelerate the design process, and the capabilities of circuit simulators and layout editors were significantly enhanced. Also beginning in 1992, research on logic level simulation of RSFQ circuits was initiated [19]. This early research effort led to the development of more efficient tools for simulating RSFQ circuits at the logic level [20].

Until very recently, tools for designing RSFQ circuits have been developed as separate components. These tools could not exchange data between each other and did not conform to standards used by CAD vendors within the semiconductor circuits industry. Several groups have made an effort to develop their own environments for the design of RSFQ circuits, based on a combination of commercially available, public-domain, and in-house developed tools [21]–[23]. Nevertheless, only a few coherent and integrated environments have been developed to date.

In part because of these deficiencies, the most complex RSFQ circuits are composed of roughly 2000 Josephson junctions and even at this scale they have required a considerable and tedious effort to realize [12], [24]. Further progress, to higher clock rates and greater levels of integration complexity, will require significantly more sophisticated CAD tools. Nevertheless, the recent development of the rather small-scale circuits have prepared the way for RSFQ circuits of drastically ambitious performance and complexity which are now envisioned.

In 1997, a large trial project began to explore the possibility of building a petaflops (10^{15} floating point operations per second) scale supercomputer based on RSFQ logic, under the title *Hybrid Technology MultiThreading (HTMT) Architecture* [25], [26]. In principle, a petaflops system based on RSFQ logic and other advanced technologies may be realizable within a ten-year time frame, far earlier than would be possible assuming the evolutionary progress of classical semiconductor-based technologies [2], [25]. A single RSFQ processor required for the HTMT architecture is expected to contain several million Josephson junctions. This complexity is far beyond the ability of currently available CAD tools targeted for developing medium-scale RSFQ circuits. Not only must tools for logic level simulation be adapted and enhanced, but also new tools for timing optimization, automated logic synthesis, and automated layout synthesis will need to be developed.

Current progress in the area of CAD tools for superconducting digital electronics is driven by two primary issues:

- practical industrial-based projects aimed at developing medium-scale specialized circuits for niche applications, such as analog-to-digital converters [24], [27], time-to-digital converters (timing digitizers) [28], [29], and digital filters [11], [12], [30];
- long range projects (such as HTMT) to apply RSFQ to high performance high complexity general-purpose computing [2], [25], [26], [31], [32].

In the first case, targeting for medium-scale applications, the existing tools discussed in this paper offer improved accuracy and design efficiency sufficient for the majority of these applications. In the latter case, no tools and methodologies are sufficient to deal with the complexity of this target problem. Although currently the main obstacle on the way to the development of very large-scale superconducting digital circuits is the immature fabrication technology, the lack of appropriate tools and methodologies may soon become a significant factor limiting further progress, even if other technical challenges were to be solved.

In Section II, the basic design flow for SDE circuits is reviewed. In Section III, a survey of existing tools for the design of superconducting digital circuits is presented. This survey is followed in Section IV by a discussion of future tools required for the development of superconducting VLSI circuits. Trends regarding changes in design requirements, expected improvements to existing tools, and new tools under development are summarized in Section V.

II. Basic Design Flow

A. Established Tools

The basic design flow for small- to medium-scale superconducting digital circuits, illustrating the basic tools used during the design process, is shown in Fig. 1. The first stage of the design process, creating a junction-level schematic of the circuit from its functional description, has remained unautomated and highly intuitive. After the initial structure of the circuit and values of the components are determined, the circuit is captured by the *schematic editor*. The schematic editor is often integrated into a circuit simulator or layout editor, but it can also work as a stand-alone tool or within a larger CAD environment. The netlist of the circuit is generated automatically by the schematic editor. The exact format of the netlist is determined by the requirement of the circuit simulator. The most typical format accepted by the majority of circuit simulators is the Spice notation [33]. For simple superconducting circuits, it is possible to create the netlist of the circuit manually and avoid the use of a schematic editor, but this process becomes cumbersome even for medium-scale circuits.

In the next step, the designer chooses test input stimuli sufficient to verify whether the circuit operates correctly. These stimuli can be either defined within a schematic editor as voltage or current sources, or can be described in a textual notation as a part of the netlist. The circuit is then simulated,

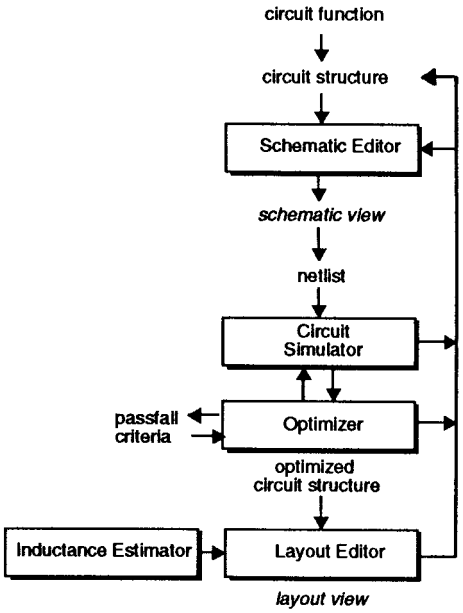


Fig. 1. Design flow for small- to medium-scale superconducting digital circuits.

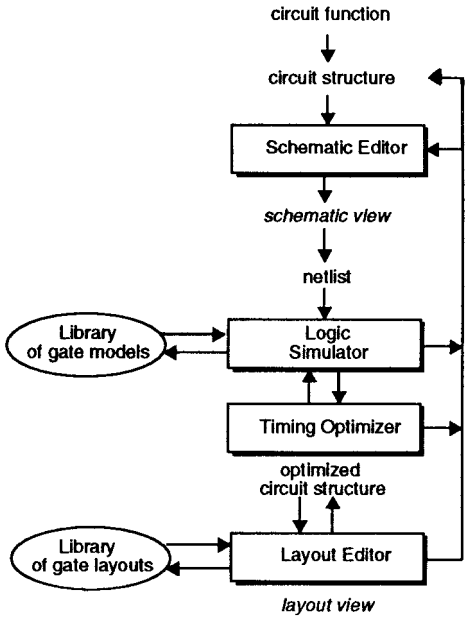


Fig. 2. Design flow for large-scale superconducting digital circuits.

verified, and modified iteratively until the correct operation for all input stimuli is obtained. In each iteration, the source of failure must be both determined and corrected. This process may be simplified and quickened if the schematic editor and the circuit simulator are closely integrated, and the circuit simulator has features permitting localizing errors within a circuit.

Optimizing the circuit starts once the set of parameters for which the circuit operates correctly is determined. The purpose of the *optimization* process is to determine the optimum values of the parameters describing the components in the circuits (e.g., the values of resistances, inductances, critical currents, and bias currents), for which the circuit has the highest probability to operate correctly after fabrication despite variations in the fabrication process. Optimization is often the most time-consuming part of the design process. After the optimization step is completed, the new parameter values are introduced into the schematic.

The next step is to create a layout of the circuit corresponding to the optimized schematic. For superconducting circuits, this is done manually using a *layout editor*. The layout editor typically includes a *design rule checker (DRC)* responsible for finding any violations of the process design rules, such as the minimum spacing between two layers. The DRC must be calibrated to work for a specific superconducting fabrication process (e.g., the Hypres, ten level, $Nb/Al_2O_3/Nb$ trilayer process [34]).

Modern layout editors include a more powerful verification tool, *layout versus schematic (LVS)* verification, integrated into the layout process. The function of LVS verification is to compare the original schematic of the circuit with the equivalent circuit schematic extracted from the layout. In this way, electrical design errors created during layout editing can be detected and fixed before the circuit is sent to fabrication. Electrical errors reported by LVS may also result from extracting parasitic components, such as parasitic inductances. If these parasitic inductances are meaningful, e.g., their values are greater than some predetermined threshold, the parasitic impedances must be added to the schematic. Once these parasitic inductances are included within the circuit schematic, the figures of merit used during optimization (such as the critical margin or expected yield of the circuit) may degrade. These figures of merit must be recomputed, and if the change is meaningful, the circuit must be reoptimized, followed by modifications to the circuit layout. The entire procedure may need to be repeated several times until no significant differences between the layout and the schematic are detected by the LVS checker, and the circuit is found to be sufficiently robust against parameter variations.

B. Emerging Tools and Extensions

For more complex superconducting digital circuits the basic design flow used for small-scale circuits is not satisfactory. The computer time required to simulate the circuit, and the number of parameters which must be included in optimization, increases to the point where it is inefficient or even computationally infeasible to directly scale up the small circuit design procedure. This problem was confronted long ago in semiconductor digital electronics. Large semiconductor circuits are designed at the logic level rather than at the device level. In this approach, the circuit is constructed using a limited set of previously developed building blocks. These blocks can be basic logic gates or they can be larger subcomponents. The basic design flow for large-scale superconducting circuits is illustrated in Fig. 2. Compared to the design process for small-scale circuits, the primary difference is the replacement of a circuit simulator by a logic simulator and the replacement of a yield optimizer by a timing optimizer.

Logic simulation for superconducting circuits can be performed using *logic simulators* developed for semiconductor

electronics. For superconducting voltage state logic the use of existing logic simulators is straightforward because the same convention is used to represent the logic states and both technologies use a similar suite of basic gates. For RSFQ logic, existing logic simulators with their libraries of standard gates cannot be applied directly. The main problems are:

- the different conventions used to represent logic states based on pulses rather than voltage levels;
- a different suite of basic gates including clocked RSFQ elementary gates instead of combinational semiconductor gates;
- the strong influence of RSFQ gates on each other that may affect timing parameters and thus accuracy of the simulation.

Despite these difficulties, existing simulators have been adapted to work with RSFQ logic. This was achieved by creating model libraries of RSFQ gates that accurately describe the behavior of RSFQ gates and at the same time can be simulated using traditional semiconductor-based simulation tools. The most straightforward technique for creating such models is the use of a *hardware description language (HDL)* such as Verilog HDL or VHDL [20].

To be useful for timing analysis of a large circuit, RSFQ gate models must include information about timing parameters, such as the propagation delay, the hold time, and the setup time [19], [35], [36]. Values of these parameters must be found using circuit (junction) level simulation. This is straightforward, but nevertheless, doing all of the simulations required to manually determine the hold time or setup time is both time-consuming and cumbersome. Therefore, either existing circuit simulators must be extended to permit these timing parameters to be determined or specialized tools closely integrated with the circuit simulators need to be developed to accomplish the same purpose.

Logic simulation by itself can be used to verify the correct function of the circuit and to detect any gross timing errors. Nevertheless, it does not provide the designer with information about the optimum values of the interconnect delays within the circuit. To obtain these values, a *timing optimizer* must be developed. This tool requires information about the circuit structure, assumed timing scheme, circuit layout limitations, and variations of timing parameters of the basic cells to calculate the optimum interconnect delays. To be completely general, the tool would need to accept both synchronous and asynchronous timing schemes.

Versions of all of these aforementioned tools have been developed but are not presently in widespread use. The primary reason is the small number of research groups involved in the development of large-scale RSFQ circuits, while most research groups deal with only small-scale circuits and highly repetitive structures. Another primary obstacle is the lack of a generally accepted theory regarding how best to properly model timing in RSFQ circuits.

C. Future Tools

When the complexity of superconducting circuits grows to a VLSI level, such as a million junctions on a single chip, these emerging tools will be grossly inadequate. One primary concern is layout editing. Even for circuits with thousands of Josephson junctions, manual editing is grueling and excessively time-consuming. For larger circuits, the time necessary for editing quickly becomes prohibitive. The only solution is to develop tools for *automated layout synthesis.* Unfortunately, the existing tools appropriate for semiconductor circuits cannot be easily adapted because of differences in implementation and in the characteristics of interconnects between these two technologies.

The other area where automation will certainly become indispensable for VLSI circuits is logic synthesis. Current methods, based mainly on designer intuition, do not scale well for larger circuits with little regularity or a large variety of gate types. Tools for *automated logic synthesis* of semiconductor circuits cannot be easily calibrated to RSFQ logic. The primary obstacle is that RSFQ gates are clocked, not combinational, and that the synthesis process should include the clock distribution network.

None of these tools have been developed to date for superconductors. Researchers have hardly begun to investigate the feasibility of calibrating semiconductor CAD tools for use in automated logic and layout synthesis.

III. Survey of Existing SDE CAD Tools

In this section, the results of a survey of superconducting digital electronics design tools is presented (Tables I–VI). Five basic categories of existing SDE design tools, currently in use by major SDE university and industry groups worldwide, have been considered. Although some of the tools are specific to RSFQ logic, note that the majority of the tools can be used for other types of superconducting logic. A significantly expanded version of this survey has been published on the World Wide Web in the form of interactive tables [37]. It is intended that the survey will be continued in the future and that the web site will be updated as the technology evolves. The reader is encouraged to study the more detailed version of the tables available on the web, updated to represent the most recent state of SDE CAD technology.

In this paper, the features of all major existing tools in each of the five categories of SDE CAD tools are reviewed and compared. Then, as an example, the particular tools developed, calibrated, and/or adapted for use at the University of Rochester are discussed in greater detail. It should be understood that although the Rochester group has made an effort to select or develop the best tool in each category, the final solutions described in this article have been clearly influenced by historical, economical, and educational factors, apart from purely technical considerations. No superiority of Rochester tools over equivalent tools developed or used by other groups is implied, and the detailed description of the Rochester toolset should be treated as an example of an integrated design environment for superconducting digital circuits, rather than a set of recommendations for use by other SDE groups.

TABLE I
FEATURES OF EXISTING CIRCUIT SIMULATORS

	JSpice	WRSpice	Spice 3f4	HSpice	Eldo	PSCAN	JSim	JULIA	WinS
Vendor	Whiteley Research, Inc.	Whiteley Research, Inc.	UC Berkeley and U. of Karlsruhe	Avant! Corp.	ANACAD Computer Systems	SUNY Stony Brook	UC Berkeley	P. Shevchenko	S. Kaplunenko
JJ Model	internal	internal	internal	*external*	*external*	internal	internal	internal	internal
Basic variable	voltage	voltage	voltage	voltage	voltage	*phase*	voltage	*phase*	*phase*
Additional types of analysis									
operating range	2D	2D	2D	?	–	2D	1D	2D	–
Monte-Carlo	–	+	+	+	+	–	–	–	–
mixed-mode	–	+	–	–	+	–	–	–	–
noise	–	+	+	–	+	–	+	–	–
Input/Output Interface									
Schematic Editor	sced	Xic/sced	–	–	from Powerview	Cadence Composer, OrCAD, Xic/sced	DERA extensions	Cadence Composer	+
Output Interface	nutmeg	IPLOT	nutmeg	+	Xelga	RODEO	XGRAPHscp	+	+
Scripts	+	+	+	+	+	+	–	+	+
Interactive Simulation	–	–	–	–	–	+	–	+	+
Optimization (built-in or built-upon)									
Optimizer	MALT	MALT	ABAK	–	XOPT	COWBOY	rtry	+	+
Documentation									
Manuals	+	+	+	+	+	+/-	+	+/-	–
Articles	[15]	–	–	–	[38]	[16], [23]	–	–	–

A. Circuit Simulator

1) Survey of Existing Tools: A variety of circuit simulators has been either extended to work for superconducting circuits or has been written exclusively to simulate superconducting logic, as shown in Table I. The first group includes modifications of Spice (such as Jspice3 [15], [33], WRSpice, Spice 3f4, HSpice), and extensions of other circuit simulators such as Eldo [38]. The second group includes original simulators, such as JSim developed at the University of Berkeley, PSCAN developed at Moscow State University [16], rewritten and extended at SUNY Stony Brook [23], WinS written by Kaplunenko, and JULIA developed by Shevchenko. Numerous proprietary modifications have been added to JSim at DERA (Defense Evaluation and Research Agency) in the United Kingdom.

A primary difference between these two groups of simulators concerns user documentation, user support, and error handling. Simulators written exclusively for superconducting logic have been developed at universities for noncommercial purposes and are currently available in the public domain. To date, documentation for these simulators is scarce, user support is not guaranteed, and error handling is often inadequate.

In terms of accuracy and speed, there seem to be no fundamental differences among the simulators surveyed in Table I. The accuracy offered by these tools is adequate for the design of digital and mixed-signal superconducting circuits. No reliable speed comparison of these simulators has been performed. This comparison would require the development of a set of benchmarks and simulations performed on the same or equivalent type of machines. An additional complication is the different definition of parameters controlling the accuracy of the simulators. A comparison of speed can be meaningful only if these parameters are set to values corresponding to the same simulation accuracy. Default values of accuracy-related parameters are often different for various simulators.

One clear distinction between two groups of circuit simulators is the choice of a basic internal variable. In JSim, Eldo, and all Spice-based simulators, nodal voltages are used as basic variables; phases of Josephson junctions are derived from these voltages. On the other hand, in PSCAN, JULIA, and WinS all of the superconducting circuit equations are solved with phase as the basic variable; voltages are computed only after completing the Newton iterations required during the numerical analysis. The effect of this difference on the accuracy or speed of the simulators appears to be noncritical, but to the authors' knowledge, this difference has not been quantitatively assessed to date.

Additionally, some simulators, e.g., PSCAN, have a built-in microscopic ("Werthamar") model of Josephson junctions [39], [40], [16]. This feature may be necessary to correctly model submicron Josephson junctions in new process technologies, currently under development.

Another difference arises from the use of an internal versus an external Josephson junction model. An internal model is built into the source code of the simulator, while an external model is defined by the user using specialized notation and is invoked as a macro. Although both models are typically functionally equivalent, leading to the same simulation results and accuracy, the difference in speed can be substantial. The use of an external Josephson junction model (as in HSpice and Eldo) may slow down the simulation by one or even two orders of magnitude [41]. On the other hand, the capability

of using external models significantly simplifies the process of adapting commercial semiconductor circuit simulators for use with superconducting logic. The reduction in simulation speed may be further justified by additional simulation modes and by greater compatibility with the powerful capabilities of commercial simulators, such as Monte Carlo analysis or mixed-mode simulation.

In summary, all of the available circuit simulators appear to offer accuracy and speed sufficient for the transient analysis of small- and medium-scale superconducting circuits. The choice of simulator may depend on:

- additional operational modes, such as Monte Carlo analysis, mixed-mode simulation, or noise simulation [42], [43];
- an associated or built-in optimizer;
- documentation and user support;
- user interface;
- operating system and hardware platform;
- availability and price.

At the time of writing, *JSpice3*, *WRSpice*, and *PSCAN* are the most widely used circuit simulators by groups involved in the design of medium- to large-scale RSFQ circuits. JSim is the most popular simulator among groups dealing with the design of small-scale superconducting circuits and structures.

2) University of Rochester Tools: The basic circuit simulator used at Rochester is *JSpice3* from Whiteley Research, Inc. Its primary advantages include good documentation, user support, low price, high speed, and large user base. The simulator has an internal Josephson junction model, permits two-dimensional (2-D) margin analysis, and has a comprehensive script language and a user-friendly graphical user interface. JSpice3 is fully compatible with Spice3 and therefore can be used to simulate hybrid semiconductor–superconductor circuits. It is also compatible with its successor WRSpice. The Rochester in-house developed tools, such as MALT for circuit optimization and TAN for timing parameter extraction, have been based on JSpice3 and exploit the capabilities of its script language. Both of these tools work without modifications with WRSpice.

B. Optimizer

1) Survey of Existing Tools: As opposed to other tools such as simulators and layout editors, the existing circuit optimizers for superconducting digital circuits have primarily been developed at universities, have been written from scratch, and are either proprietary or available in the public domain. This background results from the fact that optimizing superconducting digital circuits, in particular RSFQ gates, is significantly different from optimizing semiconductor gates. The most important factors that contribute to this difference are:

- superconducting RSFQ gates are based on two-terminal Josephson junctions versus three-terminal transistors used in semiconductor logic; this characteristic complicates the design of a single gate, decreases the input and output impedance, and increases the sensitivity of the circuit to variations of the component parameters;
- certain parameters that are optimized such as junction sizes, inductors, or bias currents are unique to superconducting circuits;
- the fabrication process is significantly different, and parameter variations are larger in currently immature superconducting foundries as compared to sophisticated multibillion dollar semiconductor fabrication facilities.

Additionally, semiconductor-based digital circuits are rarely developed as purely custom circuits (exceptions include RAM's and microprocessors); instead, standard libraries are used to build most digital circuits. Therefore, it is typical that a designer of a semiconductor logic circuit may never use a circuit optimizer. On the contrary, RSFQ cells are custom designed for different applications. The first attempts to use the same basic RSFQ cells in multiple designs have only recently been accomplished by several industry groups, and robust libraries of such gates have yet to be developed [28], [44]. Therefore, efficient circuit optimization remains one of the critical steps in the development of RSFQ circuits.

The robustness of RSFQ gates can be reliably quantified by circuit *yield*, the ratio of the number of circuits operating correctly to the total number of fabricated circuits. If it is assumed that the random distribution of the parameters characterizing the fabrication process is close to Gaussian and that their standard deviations are well characterized, the yield of a circuit can be determined using a Monte Carlo analysis. This technique requires simulating the circuit multiple times with all parameters chosen at random according to their respective random distributions. Although this strategy for measuring circuit robustness is clearly the most straightforward and accurate, it is also computationally expensive.

Therefore, a simpler measure, called the *critical margin*, has been adapted by many groups [45]. Individual margins of a specific parameter are defined as the difference between the upper and lower limits of the operating parameter for which the circuit operates correctly and the nominal value of this parameter, while other parameters are held at their nominal values. The critical margin is the smallest of the individual margins calculated for all parameters. One advantage of using the critical margin as a measure of circuit robustness is that it can be efficiently computed, e.g., applying a binary search to each individual parameter. It provides an intuitively direct technique for optimizing the circuit by centering each parameter within its operating range [45]. The assumption is that when all parameters are centered, then the parameters are at their optimum values and the critical margin is that most likely to cause a circuit fault. This would in fact correspond to the maximum yield point if all the parameter effects were independent of each other, but this is never the case. Another advantage is the possibility of experimental verification of individual margins for one type of parameter—the bias current.

There is no proven correlation between circuit yield and critical margin. On the contrary, it has been shown that there exist hypothetical circuits for which the point of optimum yield is relatively far from the point of optimum critical margin [17]. Despite this, it is commonly believed that for practical RSFQ circuits these measures are positively correlated, and optimizing the circuit with respect to critical margin improves

TABLE II
FEATURES OF EXISTING CIRCUIT OPTIMIZERS

	COWBOY	rtry	WinS	MALT	XOPT	ABAK
Developers	A. Kirichenko	J. Satchell	S. Kaplunenko	Q. Herr, K. Gaj	T. Harnisch, J. Kunert, Tran Chi Hien	W. Benzing, G. de Meester, R. Koch
Simulator in use	PSCAN	JSim	WinS	Jspice3, WRSpice	Eldo	Spice 3f4
Optimization method						
Figure of merit	critical margin	critical margin	critical margin	*yield*	*yield*	*yield* or critical margin
Optimization method	heuristic	heuristic	heuristic	inscribed hyperspheares	Centers of Gravity Method	Centers of Gravity Method and other methods
No of params in a single iteration	arbitrary	arbitrary	arbitrary	8-9	arbitrary[1]	arbitrary[1]
Pass-fail criteria						
Specification	described manually in SFQHDL	textual or graphical	automatically generated, adjustable by user	automatically generated	described manually in HDL-A	described manually in C++
Supports localizing errors	+	–	–	–	+	+
User interface						
	textual	textual	*graphical*	textual	*graphical*	textual
Documentation						
Manuals	+	–	–	+	+ (in German)	+
Articles	[23], [54]	–	–	[17]	[49], [50]	[51]

[1] The computation time increases with the number of parameters or the optimization becomes less effective (smaller improvements of the figure of merit).

circuit yield. A quantitative analysis of this relationship would require defining a set of benchmark circuits, optimizing these benchmark circuits using both figures of merit, and comparing the yield of the final circuits using a Monte Carlo analysis.

Existing circuit optimizers, shown in Table II, can be divided into two distinct groups on the basis of the definition of the figure of merit being optimized. One group uses the critical margin as a figure of merit and includes COWBOY [23], rtry [46], and WinS. COWBOY is currently an integral part of the PSCAN simulation environment [23], while rtry is a stand-alone tool using JSim for simulation. The second group includes MALT developed at the University of Rochester [17], [47], [48], XOPT developed at the University of Ilmenau (Germany) [49], [50], and ABAK [51] developed at the University of Karlsruhe (Germany). All of these tools use yield as the figure of merit being optimized. MALT employs JSpice3 or WRSpice for simulation, XOPT is based on Eldo, and ABAK works with Spice 3f4.

Optimizers using yield as a figure of merit can be further divided depending on the type of optimization algorithm being used. ABAK and XOPT use statistical algorithms such as the *Center of Gravity Method (CGM)*. This algorithm determines an optimum set of nominal parameter values by:

- choosing a number of parameter sets around an initial (temporary) operating point;
- simulating the circuit for each set of parameters to determine for which of these sets the circuit operates correctly and for which it fails;
- moving the temporary operating point based on the average position of the pass–fail parameter sets.

MALT and XOPT use a deterministic algorithm called the *method of inscribed hyperspheres* (or simplicial approximation) developed by Director [52], [53]. The algorithm aims at inscribing the largest possible hypersphere within a multidimensional operating region of the circuit. After the algorithm terminates, the optimum value is chosen at the center of the inscribed hypersphere.

Heuristic algorithms are used to optimize the critical margin in COWBOY and rtry. These algorithms are highly efficient and have no clear limitations on the number of parameters. The advantages of the method of inscribed hyperspheres include its speed and accuracy. The disadvantages are the limitation on the number of parameters optimized in a single program run, and the requirement that the shape of the operating region be convex. These disadvantages can be easily overcome by properly choosing the parameter sets or transforming the parameters into a logarithmic space, as described in [17] and [47].

Statistical methods, in principle, can be applied to an arbitrary number of parameters being simultaneously optimized; nevertheless, the final result of the optimization procedure depends strongly on this number. As a result, it is difficult to assure sufficient accuracy (and thus close to the optimum choice of all operating parameters) and reasonable efficiency for a large number of parameters. For a small number of parameters, the method of inscribed hyperspheres is clearly more efficient. Further research, considering both optimization speed and post-optimization yield, is required to quantitatively compare both methods.

The other important distinction between these various optimization programs, independent of the figure of merit and optimization algorithm, is the manner of specifying the *pass–fail criteria* for a given circuit. Pass–fail criteria are used by the optimizer to distinguish between correct and incorrect circuit operation.

Two basic ways of describing pass–fail criteria are currently in use among available optimizers. In MALT and WinS, pass–fail criteria are generated automatically on the basis of the simulation results for an initially correct set of parameters. In COWBOY/PSCAN, XOPT, and ABAK, a special HDL is used to describe the correct operation of the circuit. Various languages are used for this purpose: PSCAN uses a custom-designed notation called SFQHDL [54], XOPT employs a subset of VHDL called HDL-A [50], and ABAK makes use of the general purpose object-oriented programming language C++ [51]. The exact rules for describing correct circuit behavior vary from notation to notation. Typically, a user must describe the exact order in which junctions switch within the circuit, thereby specifying the behavior of all internal nodes within the circuit.

Both approaches have their advantages and disadvantages. Automatic generation of pass–fail criteria is extremely fast and convenient. On the other hand, no support is provided to determine the correct set of parameters.

Manual specification of pass–fail criteria using HDL's may be cumbersome, particularly for medium to large-scale circuits. On the other hand, at least in principle, once properly prepared this specification can be used to search for the initial set of operating parameters and/or support localization of errors.

Additional factors in choosing a circuit optimizer include a user-friendly interface, documentation, and user base. Presently, only two optimizers are in use by more than one research group, *MALT* and *COWBOY/PSCAN*.

2) University of Rochester Tools: The *MALT* optimizer was developed at the University of Rochester from 1993 to 1994. Since then, MALT has been transferred to several university and industrial groups. It has also been extensively used within the Rochester group to design about 20 basic RSFQ cells [12], [17], [47].

A short justification of several decisions made during the development of MALT, and a short review of its features are discussed below. MALT consists of three separate programs written in C, where each program performs, respectively, the:

- automatic generation of pass–fail criteria (*init*);
- one-dimensional (1-D) and 2-D operating range analysis (*marg*);
- optimization (*opt*).

The C programs constituting MALT are supported by a set of JSpice3 scripts.

MALT was originally written to work with JSpice3 and is also compatible with the current version of WRSpice. The core of the optimization program (*opt*) can be adjusted to work with other circuit simulators.

MALT optimizes the yield of the circuit, using a deterministic method of inscribed hyperspheres. In the authors' opinion, MALT offers a better tradeoff between the optimization effectiveness and computational efficiency than other optimization tools currently available. MALT provides a more optimal result than programs based on critical margin optimization and is more efficient than statistical methods aimed at providing an equivalent optimal solution.

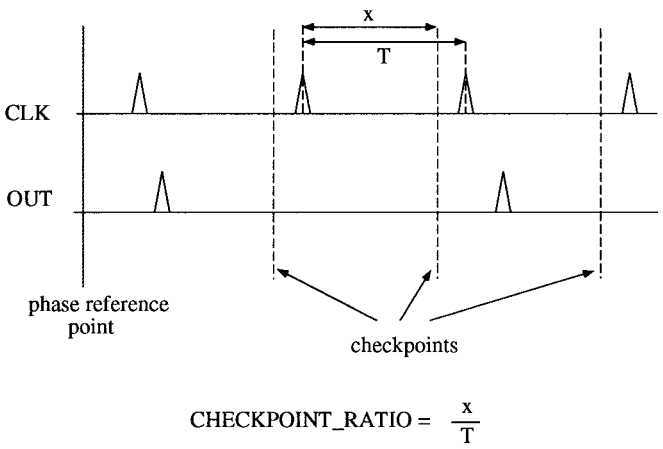


Fig. 3. MALT: Generating pass–fail criteria for *synchronous* circuits. Positions of checkpoints are determined on the basis of the CHECKPOINT_RATIO, which defines the relative position of each checkpoint in the clock cycle.

One of the distinct features of MALT is the capability to generate pass–fail criteria automatically on the basis of a single simulation of the circuit for correct operating parameters. This capability is performed by defining time windows when each specified junction must undergo a 2π phase change. The position of these windows is determined automatically on the basis of the simulation results for a correct set of operating parameters as well as a set of numerical parameters provided in the MALT configuration file [48]. The algorithm for determining the positions of these checkpoints (those points where the value of the phase at the output nodes is verified) is different for synchronous (clocked) and asynchronous circuits. For synchronous circuits, the checkpoints are set at a given percentage of a clock cycle, typically close to the end of the cycle, as shown in Fig. 3. By applying this convention, any realistic variation in the clock-to-output delay does not influence the outcome of the phase comparison. For asynchronous circuits, the checkpoints are set a certain time interval before and after the nominal position of the output pulse, as shown in Fig. 4. This procedure permits accommodating for limited variations in the position of the output pulses. This convention was later adapted in WinS, where additionally the user of the program can interactively change the positions of the checkpoints relative to the output waveforms.

Two main problems have been reported by users of MALT. First, the user interface is textual and requires preparing several input files for each circuit. Second, the selection of the circuit parameters subject to optimization during a particular optimization run must be done manually and intuitively on the basis of criticality and interrelations among the various parameters.

To address the first problem, MALTTool, a graphical user interface for all MALT programs is under development. This interface should greatly simplify the preparation of the input data and the interpretation of the final results.

The second problem, however, is intrinsic to this technique. The algorithm itself limits the number of parameters capable of being simultaneously varied during the optimization to eight or at most nine. Running on an Ultra Sparc, an eight-

TABLE III
FEATURES OF TOOLS FOR LAYOUT EDITING AND VERIFICATION

	Cadence Virtuoso and Diva	Xic	L-Edit	Magic	Kic
Vendor	Cadence Inc.	Whiteley Research Inc.	Tanner EDA	UC Berkeley	Whiteley Research Inc.
Extensions	SUNY Stony Brook, U. of Rochester, TRW, RSFQ Inc.				
Verification					
DRC	batch	batch and interactive	batch	interactive	–
ERC	+	+	–	–	–
Inductance extraction routine	Lmeter or inductance per square	Lmeter	–	INDEX	–
LVS - connectivity	+	+	–	–	–
LVS - parameter values	+	–	–	–	–
File formats					
Output files	GDSII, CIF	GDSIIr3, CIF, native, text mode GDS	GDSII, CIF	GDSII, CIF	GDSIIr3, CIF, native, text mode GDS
Tech files	SKILL	keyword/value pair format	?	unique	–

dimensional (8-D) optimization of a circuit containing 20 junctions requires about 10 min, a nine-dimensional (9-D) optimization requires from 1 to 2 h. Even the simplest RSFQ gates have more than eight or nine parameters. To cope with this problem, several parameters at a time can be optimized in an iterative process. This method is effective but has some limitations. The most critical parameters must be determined and included in each successive optimization. Otherwise, the figure of merit will change from one optimization to the next, and the set of parameter values will not converge to the optimum solution. Second, concavities may exist in the operating region, and the parameter values may not converge to the optimum. These problems can be overcome through the use of optimization techniques described in [17] and [47], which provide general guidelines for choosing proper sets of parameters for successive optimizations. Nevertheless, the final choice of the parameter sets is circuit specific and knowledge of the circuit operation can significantly enhance the efficiency of the iterative optimization.

C. Layout Tools

1) Survey of Existing Tools: Most of the layout tools used to design superconducting circuits (see Table III) are *commercial semiconductor* CAD tools. This outcome has occurred because semiconductor layout tools are sufficiently general to permit relatively easy calibration in order to accommodate different types of technologies. The calibration is often reduced to writing appropriate technology files, specific for a given fabrication process. In addition, layout tools are among the most complex and time-consuming to develop, and the current scope of SDE research and design effort worldwide does not justify development of such tools specifically for superconducting technology.

Price may be a decisive factor in selecting a layout tool. In particular, the high cost of the Cadence toolset, including Virtuoso (layout editor) and Diva (layout verification), is a deterrent to small industry research groups. On the other hand, the relatively low *educational* price together with the elaborate capabilities of Cadence have caused this toolset to become the most commonly used among university researchers. This situation is particularly disadvantageous for small companies that cannot afford Cadence, because they cannot anticipate that universities will calibrate other less expensive tools. Lower cost alternatives to Cadence are such commercial tools as Xic and L-Edit or public domain programs such as Magic and Kic.

The basic part of each layout tool, the layout editor, is difficult to compare. The most important feature, a user-friendly interface is often a matter of personal preference. The public domain Magic editor has the disadvantages that it only permits Manhattan structures and it applies old-fashioned pixel-oriented (versus object) graphics. Larger and more objective differences exist in the area of layout verification. These differences concern tools for

- DRC;
- electrical rule checking (ERC);
- LVS.

Most of the layout tools include a *DRC*. This capability is responsible for reporting violations of design rules such as the minimum spacing between two layers, the minimum width of a given layer, or the minimum overlap of one layer outside of the second layer. An *ERC* detects problems such as power/ground shorts and unconnected floating nodes. As similar rules must be verified for CMOS and other semiconductor processes, superconductive-based design and electrical rules can be easily incorporated into existing commercial semiconductor layout tools.

More difficult is the calibration of a final verification process—LVS comparison. The purpose of LVS verification is to compare the original circuit schematic with the schematic extracted from the circuit layout. In the first phase, a circuit extractor identifies all devices in the layout and the connections between the devices. As a result, a netlist of the extracted circuit is created. Optionally, values of the device parameters are calculated during extraction and included in the netlist. In the second phase, a netlist obtained from the original schematic is automatically matched with the netlist extracted from the layout and differences in connectivity and parameter values are reported.

Unmodified *semiconductor* tools can typically perform extraction of transistors, resistors, and capacitors, but lack the capability to extract inductances, particularly inductances of superconducting structures. As a result, special extensions have been added to Cadence Diva at SUNY Stony Brook and the University of Rochester. Stony Brook extensions employ Lmeter to extract inductor values, while the Rochester extensions use a simplified approach based on computing the number of squares constituting an inductor and multiplying it by an average inductance per square. The number of squares constituting corners, tees, and vias are adjusted according to the scaling factors used for resistances.

LVS comparison using Xic is limited to checking for correct circuit connectivity. An additional interface from RSFQ, Inc. permits calculating values of inductances using Lmeter. The authors are not aware of any other tool currently being calibrated to permit LVS verification of superconducting circuits. As a result, *Cadence* and *Xic* are the primary candidates for layout tools for the design of SDE circuits, with Cadence primarily being used by universities and large companies and Xic aimed at smaller industrial research groups.

2) University of Rochester Tools: The Cadence layout tool package was adapted at Rochester in 1994 due to its superiority over the previously employed Magic and L-Edit and its relatively low educational price for the entire toolset. This was the beginning of an effort to create an integrated environment for the design of large-scale RSFQ circuits. The Cadence toolset includes other tools such as a library manager, schematic editor, two logic simulators, and an interface to various circuit simulators, thereby providing an excellent framework for an integrated design environment.

Initially, technology files for the Hypres fabrication process were developed to permit using *Cadence Virtuoso* as a layout editor for superconducting circuits. These technology files have since been extended to enable DRC and ERC using *Cadence Diva*, based on the Hypres design rules [34]. Finally, parameter extraction and LVS verification based on Cadence Diva were also added and extensively tested in 1996 [21].

Two features of the extraction and LVS capabilities are worth mentioning. First, the precision of the parameter extraction is: 1% for junction areas, 2% for shunt and bias resistors, and 10% for significant inductors. Secondly, all significant inductors in the physical layout must be marked before extraction using a special dummy layer. This added information prevents extracting parasitic inductances not appearing in the schematic, which would make the comparison with an original schematic unsuccessful. The user must specify which parasitic inductances to extract, and these inductances are introduced manually into the circuit schematic. The dummy layer is also necessary due to the relative inaccuracy of the extraction procedure for small inductances. For typical parasitic inductances, the extraction error may easily exceed 50%. No components other than inductors require a dummy layer.

These extensions to the Cadence layout tools are available in the public domain. Future plans include increasing the accuracy of the inductance extraction procedure.

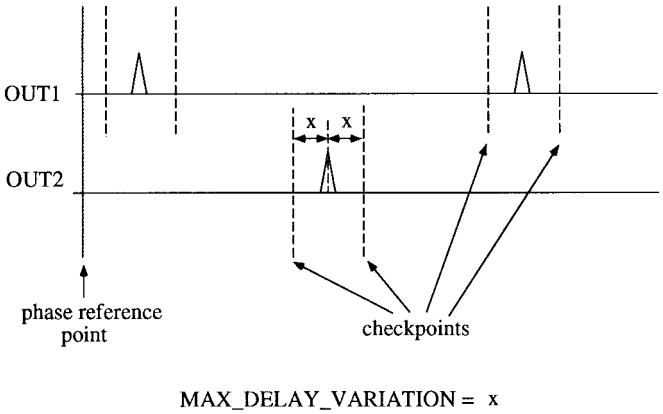


Fig. 4. MALT: Generating pass–fail criteria for *asynchronous* circuits. Positions of checkpoints are determined on the basis of MAX_DELAY_VARIATION, which determines the distance between each checkpoint and a nominal position of an SFQ pulse at the output of the circuit.

D. Schematic Editors

Schematic editors are usually associated with circuit simulators or layout tools. This function is provided at Rochester by the Cadence Composer. Customizing this editor for RSFQ logic requires the design of a library of symbols for the basic components of RSFQ circuits such as Josephson junctions, inductors, resistors, and voltage and current sources. One of the unique features of this schematic editor is the capability to export and import data in Electronic Design Interchange Format (EDIF) [55], which is the standard textual notation used to transfer schematics among various design environments.

E. Inductance Estimator

1) Survey of Existing Tools: There are a variety of tools for estimating the inductance of superconducting structures, as shown in Table IV. These tools include both commercially available packages, such as Maxwell and Sonnet em, as well as public domain tools such as Fast Henry and Lmeter.

The primary distinction between these inductance estimators is the type of algorithm used. Three-dimensional (3-D) methods are in principle more accurate, but are typically quite slow. Two-dimensional methods introduce some computational inaccuracies but are significantly faster than the 3-D methods. There also exist simple programs, such as *sline*, based on approximate 2-D analytical formulas [56]. These primitive tools can only be used for estimating the inductances of simple microstrip lines.

In 1996, a survey regarding the accuracy of existing tools for inductance estimation was initiated [57]. The results of this survey were published on the web and are summarized in Table V. The survey solicitation letter [57] specifies the geometry of seven common structures, which roughly comply with the Hypres design rules [34]. The results submitted by various researchers use ten different methods and programs. The 3-D programs, Fast Henry [58], Sonnet em [59], and Maxwell [60], are in close agreement. For microstrip lines, the results obtained using these programs differ from each other by no more than $\pm 1\%$; for three other structures, two corners

TABLE IV
FEATURES OF EXISTING INDUCTANCE ESTIMATORS AND EXTRACTORS

	Sline	Lmeter	ML	Fast Henry 2.0S	Sonnet em	Maxwell 3D, 2D	Icalc
Vendor	Whiteley Research Inc.	SUNY Stony Brook	M. Khapaev	Whiteley Research Inc.	Sonnet Software Inc.	Ansoft Corp.	U. of Rochester
Author	S. Whiteley	P. Bunyk, S. Rylov	M. Khapaev	S. Whiteley			B. Guan, P. Rott, D.K. Brock, M.J. Feldman
Method	2D, analytical	quasi-2D	3D	3D	3D-planar	3D	table look-up based on 3D precomputations
Computation time							
Microstrip	seconds	seconds	5-10 min[1]	5-10 min[2]	seconds[3]	~ 5 min (2D)[4]	instantenous
Corner, Tee	N/A	seconds	5-10 min[1]	15-20 min[2]	seconds[3]	~ 10 min (3D)[4]	instantenous
Via	N/A	seconds	5-10 min[1]	1-2 hrs[2]	seconds[3]	~ 10 min (3D)[4]	instantenous
Irregular Structure	N/A	1-2 min	5-10 min[1]	??	~ 15-20 min	~ 30 min (3D)[4]	N/A
Input format							
	interactively provided parameters	CIF file	SIF file	3D-mash (text file)	graphical view or GDSII + interactively provided parameters	graphical view + interactively provided parameters	structure type + inductance dimensions
Documentation							
Manuals	self-explanatory	+	–	+	+	+	self-explanatory
Articles		[18]	[64], [65]	–	[59]	[60]	[58]

[1] PC Pentium-166, 32 bit MS DOS extender
[2] Sun Sparc IPX
[3] HP7000, 128 MB RAM
[4] Sun Sparc20

and one via, the difference is no more than ± 0.02 pH; and for one more structure, "via with spreading," the difference does not exceed 0.04 pH. The difference between these results is substantial for only one structure, the crossover. This inconsistency is possibly due to a difference in interpreting the structure description.

Lmeter [18] provides less accurate results, including a 5% difference in the inductance of microstrip lines; less than 0.1 pH difference for the corners and the via; and less than 0.2 pH difference for "via with spreading" compared to the average results generated by the 3-D programs. Lmeter estimations should be sufficiently accurate for most applications, and the program can be further calibrated on the basis of 3-D simulations or test measurements based on a broad sample of experimental data. Several other inductance estimation programs are described in [61]–[65].

Despite its inaccuracies, *Lmeter* is the most widely used tool to estimate inductance. Its primary advantages are high speed and the ability to accept a description of the inductance structure in the form of a standard CIF file which is generated by most layout editors. Most 3-D programs are much slower, and only one, Sonnet em, accepts the standard GDSII file generated by many layout editors. Interfaces for Cadence Virtuoso and Xic have been written to extract inductances and provide an LVS comparison based on Lmeter.

2) University of Rochester Tools: The method used at Rochester to estimate and layout inductances combines the high accuracy of 3-D methods with instant access [58]. Fast Henry is used to precompute values of inductances for most typical widths and a large range of lengths of microstrip lines, for three basic combinations of layers in the Hypres fabrication process. The inductances of intermediate length microstrip lines are then found by linear interpolation. The same procedure is repeated for different dimensions of corners, tees, and vias. All precomputed results have been collected in a look-up table, and a graphical user interface (GUI) tool *Icalc* has been developed to access this table. Icalc determines the dimensions of a structure corresponding to the value of an inductor as well as a value of an inductance for a particular structure.

Icalc unlike Lmeter cannot be used to automatically extract inductance values from a geometric layout. The only automatic inductance extraction procedure currently implemented in the Rochester design environment is based on calculating the number of squares of a structure and multiplying the number of squares by an average inductance per square for a given combination of layers. Because of the inaccuracy involved in this procedure, small errors ($<10\%$) may remain unreported.

In the authors' opinion, the designer's errors introduced to inductances during layout editing are rare, and therefore their effect on a final design is less critical than that of the systematic errors introduced to all circuit inductors by using a less accurate 2-D algorithm to calculate the inductor dimensions.

F. Logic Simulator

1) Survey of Existing Tools: The capabilities and advantages of using logic level simulation in the design of large RSFQ circuits was first identified by Krasniewski [19]. From 1992 to 1993 he developed at the University of Rochester a library of RSFQ cells based on Design Works, a standard low-cost semiconductor logic simulator. As Design Works does not support behavioral models of gates in a HDL, RSFQ cells are modeled using an equivalent schematic description.

TABLE V
RESULTS OF THE INDUCTANCE STRUCTURE RODEO. (a) ABSOLUTE RESULTS AND (b) RELATIVE ERRORS WITH REGARD TO THE MOST LIKELY ANSWER

Source	Method	#1 μstrip (low L)	#2 μstrip (high L)	#3 corner (low L)	#4 corner (high L)	#5 crossover	#6 via	#7 via w. spreading
comments		**per 10 μm**	**per 10 μm**	**wrt #1**	**wrt #2**	**wrt #2**	**wrt #1**	**wrt #1**
M. Feldman	estimated, from Chang [56]	0.77	1.54	0.215	0.26		~0.30	
J. Zmuidzinas	analytic	0.72	1.60					
S. Whiteley	SLINE	0.79	1.665					
S. Whiteley	Fast Henry 2.0S	0.755	1.59					
P. Rott	Fast Henry 2.0S	0.77	1.62	0.20	0.24	0.38	0.305	-0.17
P. Rott	Fast Henry 2.0			0.21	0.23	0.35	0.28	-0.22
T. Kerr	Sonnet em	0.76	1.63	0.20	0.22	-0.15	0.31	-0.19
J. Du	Maxwell	0.76	1.595	0.215	0.21	0.22	0.295	-0.25
N. Dubash	Lmeter	0.725	1.525	0.19	0.235	0.16	0.49	-0.14
M. Khapaev	ML gp	0.73	1.93	0.19	0.23	0.12	0.31	-0.10
most likely answer	**(eyeball the above)**	**0.76**	**1.61**	**0.21**	**0.23**	**??**	**0.30**	**-0.21**

(a)

Source	Method	#1 μstrip (low L)	#2 μstrip (high L)	#3 corner (low L)	#4 corner (high L)	#5 crossover	#6 via	#7 via w. spreading
comments		**per 10 μm**	**per 10 μm**	**wrt #1**	**wrt #2**	**wrt #2**	**wrt #1**	**wrt #1**
M. Feldman	estimated, from Chang [56]	+ 1%	- 4%	0	+ 0.03		0	
J. Zmuidzinas	analytic	- 5%	- 1%					
S. Whiteley	SLINE	+ 4%	+ 3%					
S. Whiteley	Fast Henry 2.0S	- 1%	- 1%					
P. Rott	Fast Henry 2.0S	+ 1%	+ 1%	- 0.01	+ 0.01		0	+ 0.04
P. Rott	Fast Henry 2.0			0	0		- 0.02	- 0.01
T. Kerr	Sonnet em	0 %	+ 1%	- 0.01	- 0.01		+ 0.01	+ 0.02
J. Du	Maxwell	0 %	- 1%	0	- 0.02		0	- 0.04
N. Dubash	Lmeter	- 5%	- 5%	- 0.02	0		+ 0.19	+ 0.07
M. Khapaev	ML gp	- 5%	+ 20 %	- 0.02	0		+ 0.01	+ 0.11

(b)

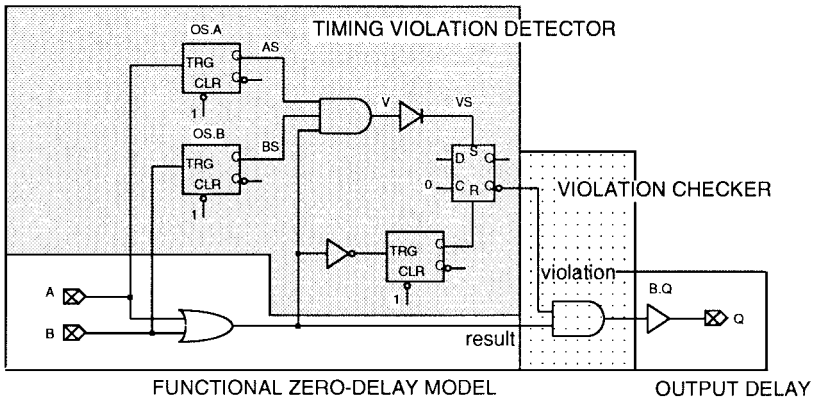


Fig. 5. URSULA: Equivalent schematic description of the confluence buffer composed of standard semiconductor gates.

The approach is to create a circuit consisting of classical semiconductor gates that would duplicate the behavior of a given RSFQ cell, as shown in Fig. 5. The circuit consists of two main parts, one part responsible for implementing a logic function and the second part for checking for violations of any hold and setup time constraints. A regular SFQ pulse is represented by a short rectangular pulse with a logic value "one." A rectangular pulse with a logic value "unknown" is generated at the output when a violation of a timing constraint occurs in the circuit, as shown in Fig. 6.

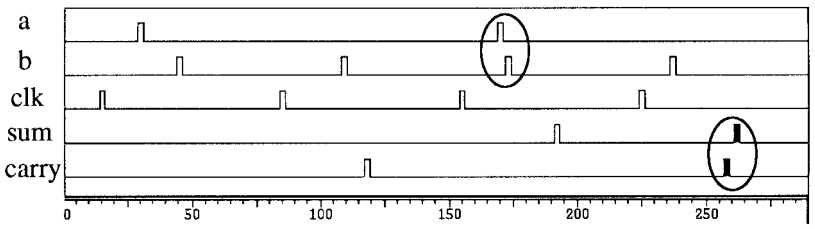


Fig. 6. HDL: Verilog-XL simulation of an RSFQ half-adder; *sum* $= a \oplus b$, *carry* $= a \cdot b$. Shaded pulses at the *sum* and *carry* output in the fourth clock cycle depict the violation of the minimum separation time between the pulses at the inputs a and b in the previous clock cycle.

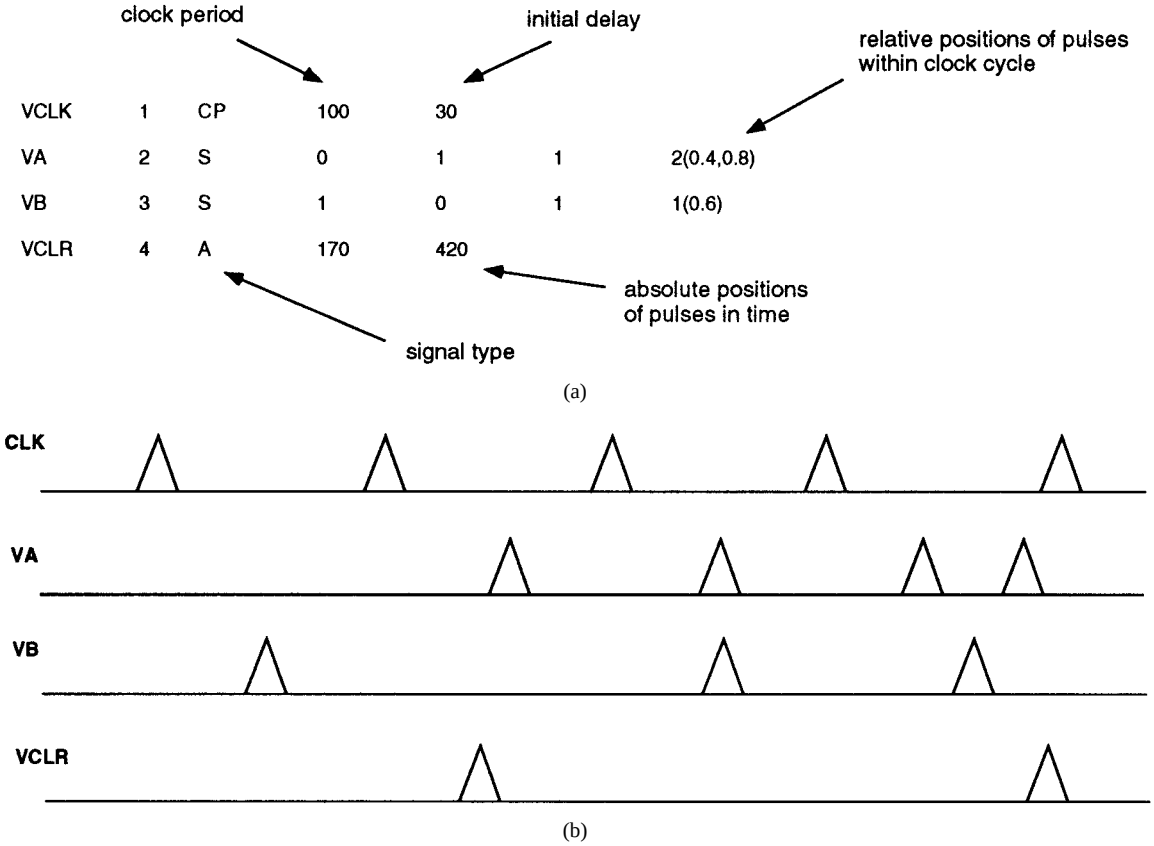


Fig. 7. SIG: (a) input sequences described using simplified notation for periodical (P), synchronous (S), and asynchronous (A) signals; (b) corresponding input waveforms.

The system consisting of the library of RSFQ gates and Design Works is the University of Rochester SUperconducting Logic Analyzer (URSULA) and was used successfully during the design of a digital filter [12]. The system was fully functional and offered sufficient modeling accuracy. Its main disadvantages include:

- difficulty in creating models for new gates;
- difficulty of transferring the cell library to other logic simulators;
- limitations imposed by Design Works and the MacIntosh operating system on the size of the simulated circuit.

A library of RSFQ cells was developed in Verilog HDL in 1995 [20] and in VHDL in 1997. To the authors' knowledge, these libraries constitute the first fully functional, accurate, and extensible solution for the logic simulation of RSFQ circuits. These models can be transferred to a variety of standard semiconductor-based simulation environments. Both of these Verilog and VHDL libraries are currently available in the public domain.

Two other groups are creating libraries of behavioral models of RSFQ gates. The SUNY Stony Brook group is designing a library in VHDL using Cadence Leapfrog VHDL as the primary logic simulator. A specific feature of these models, in contrast to the Rochester models, is the use of bias-dependent delays [66]. The University of Ilmenau group is designing a library in VHDL, using Powerview for logic simulation and Eldo for mixed analog-digital simulation [22]. Both of these libraries are currently under development and are at present not available in the public domain.

2) University of Rochester Tools: Very High Speed Integrated Circuit Hardware Description Language) (VHDL) and *Verilog HDL* are both standard HDL formats in common use today [67]–[70]. VHDL became the IEEE standard in 1987; Verilog in 1995. The entire top-down design process of most

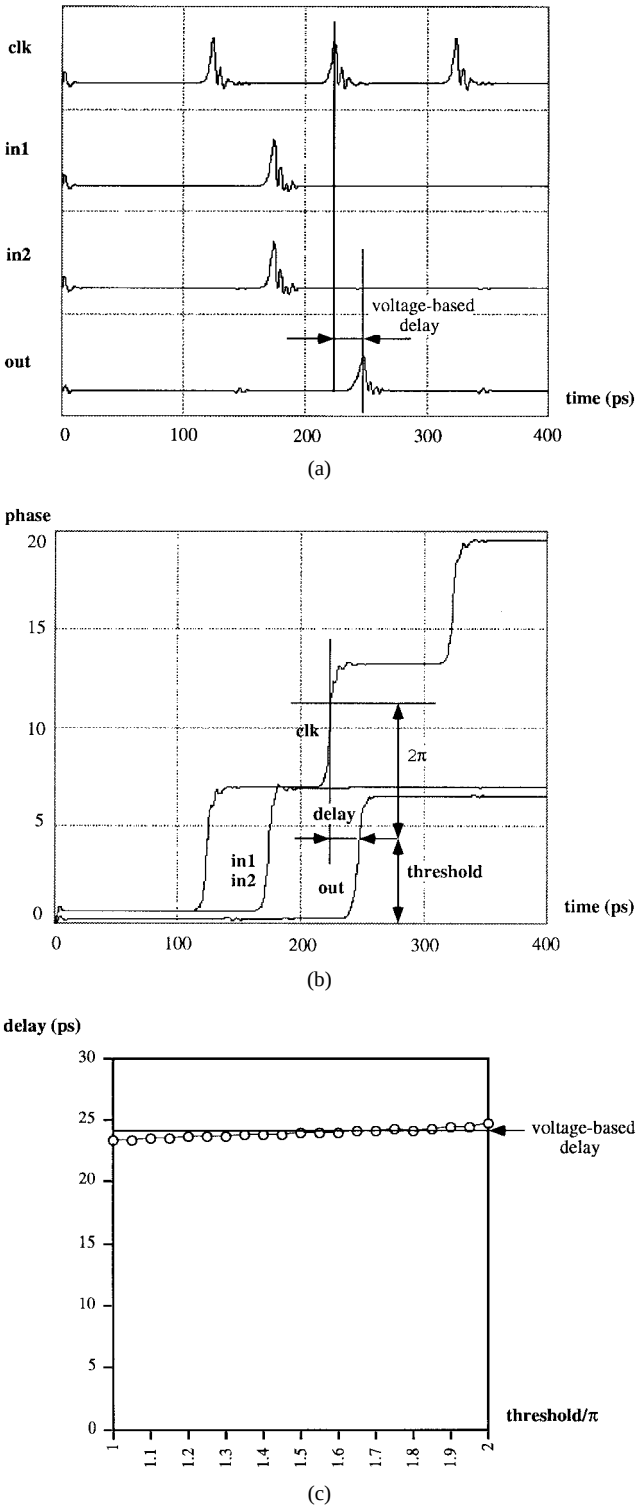


Fig. 8. TAN: Phase-based definition of timing parameters: (a) operation of the AND gate in voltage domain and a voltage-based definition of the clock-to-output delay; (b) operation of the AND gate in the phase domain and a phase-based definition of the clock-to-output delay; and (c) dependence of the phase-based delay on the choice of the phase threshold in the range from π to 2π.

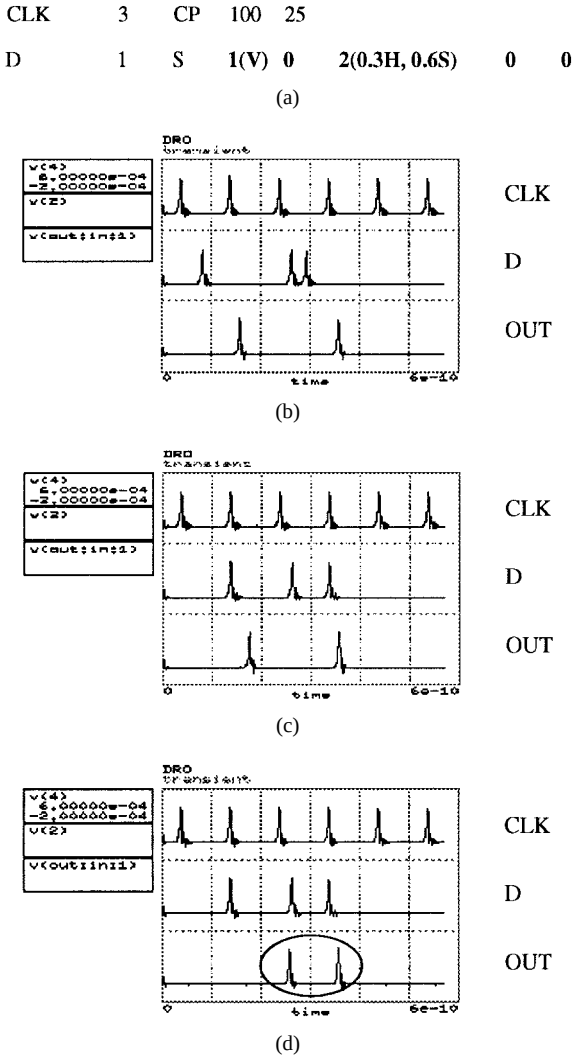


Fig. 9. TAN: Calculating the setup time for a DRO cell using TAN: (a) definition of input sequences for CLK and D inputs using SIG notation; pulses followed by a letter "S" change their position during the search for the setup time; pulses followed by "H" change their position during the search for the hold time; pulses followed by "V" change their position during both searches; (b) timing waveforms correspond to nominal positions of data pulses; (c) timing waveforms for the last position of D pulses before the setup time violation; and (d) timing waveforms for the first position of D pulses after setup time violation.

current large-scale and very large-scale *semiconductor* digital circuits is based on one of these HDL formats [71]–[73]. HDL's are employed to describe the behavior and structure of the circuit at all design levels for the purpose of functional simulation and automated logic synthesis [74], [75].

The behavioral models of RSFQ cells have been implemented at Rochester in both languages. Initially, Verilog HDL was chosen due to its relative simplicity, flexibility, and widespread use in the semiconductor industry. The development of libraries in VHDL soon followed, motivated by the fact that many government sponsored projects require or prefer the use of VHDL. Both languages are well supported within the Cadence toolset; *Verilog-XL* can be used to simulate circuits described in Verilog, and *Leapfrog-VHDL* can be used to simulate circuits composed of VHDL models. Many other simulators support the use of one or both of these HDL formats. These tools include Veriwell used by Hypres and Eldo used at the University of Ilmenau.

The Rochester models describe the behavior of RSFQ circuits in a manner *independent of the internal structure* of an RSFQ cell. The only implementation specific information

included in the model are the values of the timing parameters: the delay, the hold time, the setup time, and the minimum separation time [19], [35], [36]. These parameters can be easily changed without understanding the details of the model. The values of the timing parameters are extracted using circuit simulation. This procedure, if performed manually using a circuit simulator, is both cumbersome and time-consuming, particularly for extracting the hold and setup times. Therefore, a special program, the timing parameter extractor, TAN, has been developed at Rochester to simplify this calculation. TAN is described in detail in Section III-G2 and is available in the public domain.

Both the Verilog and VHDL models developed at Rochester permit two modes of operation. In one mode, the delays and other timing parameters are set to their nominal values. In the second mode, the timing parameters are set to the random values chosen according to a Gaussian distribution which corresponds to variations in the fabrication process. This mode can be used for Monte Carlo analysis of large-scale RSFQ circuits in order to verify the robustness of the timing scheme and the interconnect delays.

G. Extensions to Existing Tools Developed at the University of Rochester

Additional special purpose tools developed at Rochester are described below. The authors are unaware of any similar tools developed by other groups. The first two tools have been designed to extend certain features of a circuit simulator. Although these tools are aimed specifically at extending the capabilities of JSpice3, they might be adapted to work with other circuit simulators. No other circuit simulator contains similar options.

1) SIG: SIG is a preprocessor for JSpice3 that converts a description of an input sequence in a custom notation into a standard Spice notation. Input signals can be defined to be one of the following [see Fig. 7(a)]:

- periodical signals (such as a clock signal) described using a period and an initial delay;
- synchronous signals described using a sequence of zeros and ones (such as an input to an RSFQ clocked gate);
- asynchronous signals described as a sequence of points in time (such as an input to an RSFQ nonclocked gate).

SIG converts each input sequence into a Spice piecewise linear *(pwl)* notation with each pulse represented by a triangle with an area integrated over time equal to the fundamental flux of the magnetic field, as shown in Fig. 7(b). This triangular waveform, provided at the input of a multiple-stage JTL line, generates a sequence of appropriately spaced SFQ pulses. SIG is fully compatible with MALT and TAN and generates all relevant input files required by these programs.

2) TAN: The Timing ANalyzer (TAN) is a tool for calculating the timing parameters of a basic RSFQ gate, such as the propagation delay, the hold time, and the setup time as well as for estimating the standard deviation of these parameters as a function of variations in the fabrication process. TAN analyzes the output from JSpice3 and replaces this output with a sequence of numbers representing the approximate positions of all SFQ pulses in time. In superconducting circuits, the SFQ pulse corresponds to a Josephson junction rapidly increasing its phase by 2π. In TAN, the position of the pulse is defined as the moment when the phase equals to $3\pi/2$. Since all of the timing parameters depend upon the relative difference between the positions of two pulses rather than the absolute position of a single pulse, the choice of the $3\pi/2$ point rather then some other specification is immaterial (see Fig. 8).

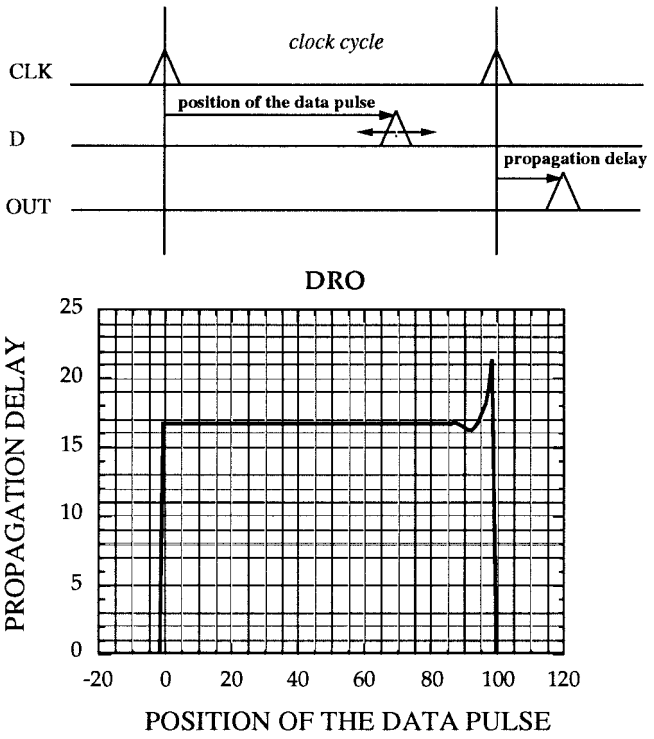


Fig. 10. TAN: Determining the dependence between the propagation delay and the position of the input pulse within a clock cycle for the DRO cell.

Calculating the hold and setup times for a synchronous SFQ cell requires a *sequence* of Spice simulations where the relative separation of the data pulse from the clock pulse is varied. TAN uses a *binary search algorithm* to do this. The user specifies a data pulse position giving correct circuit operation and a second data position giving incorrect circuit operation. Then each iteration of the binary search chooses the next position of the data pulse in the middle between two previously analyzed positions, and the circuit is resimulated and its function is verified. Based on this result, the respective boundary of the analyzed interval changes to the middle position, decreasing the interval by a factor of two. After several iterations the interval becomes smaller than the required precision of the binary search, and its middle position is returned as a final result. Timing waveforms corresponding to two final positions of the input pulses obtained using the binary search for the setup time of the destructive read-out (DRO) cell are shown in Fig. 9. The input test stimuli must be carefully chosen to represent an exhaustive test sequence.

In addition to computing nominal values of all timing parameters, TAN also permits computing the propagation delay as a function of the position of the data pulse within the clock cycle, as shown in Fig. 10. An additional capability is

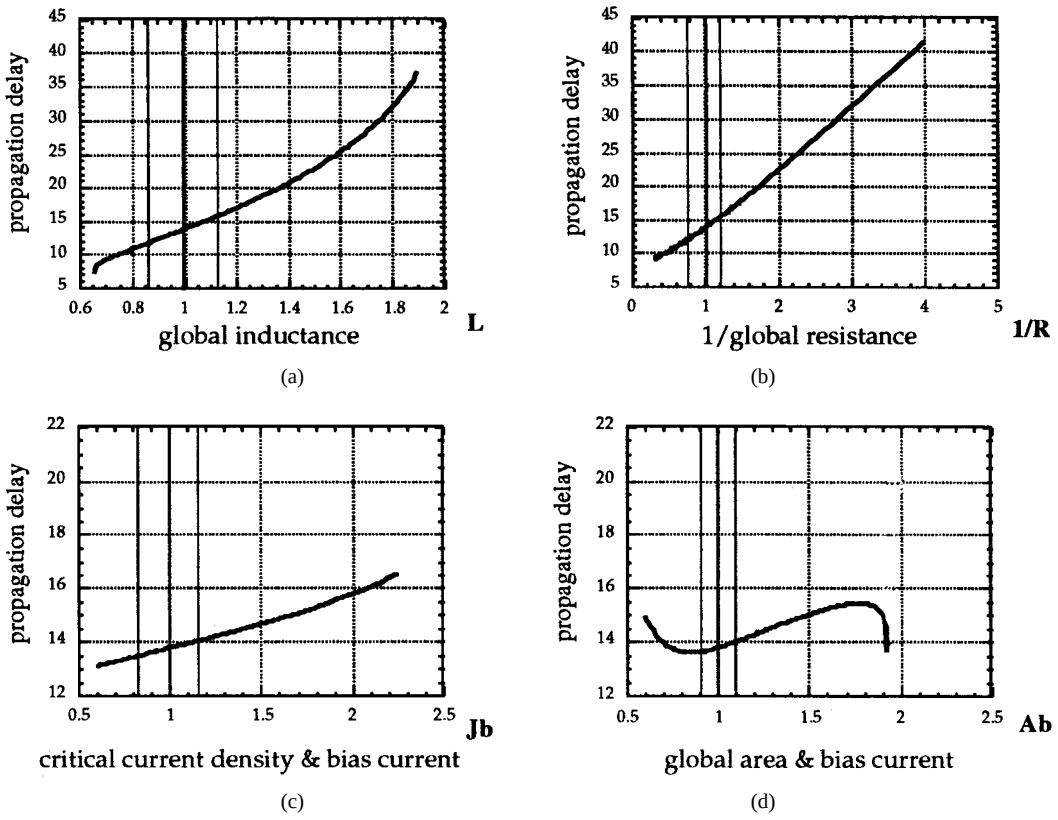


Fig. 11. TAN: Calculation of the dependence between the propagation delay and values of four normalized basic global parameters of the Hypres fabrication process for a DRO cell.

estimating the delay deviations due to process variations. The following four independent global parameters are assumed to have the primary influence on the timing parameters of the SFQ cells: the global inductance L, the global resistance R, the junction critical current density with the global bias current adjusted proportionally Jb, and the global junction area with the global bias current adjusted proportionally Ab [35], [76]. The delays of the SFQ gates in the multiparameter region determined by the process variations can be approximated as

$$\begin{aligned} delay_{\text{norm}} = {} & \alpha_L L_{\text{norm}} + \alpha_R / R_{\text{norm}} + \alpha_{Jb} Jb_{\text{norm}} \\ & + \alpha_{Ab} Ab_{\text{norm}} + \alpha_0 \end{aligned} \tag{1}$$

where X_{norm} denotes the value of the parameter X normalized to its nominal value X_{nom}, i.e., $X_{\text{norm}} = X/X_{\text{nom}}$. To determine the values of the *alpha coefficients* $\alpha_L \cdots \alpha_{\text{Ab}}$ in (1), TAN calculates the cell delays as a function of each of the four global parameters as shown in Fig. 11. The values of the coefficients are each found independently using least squares regression. The standard deviation for the delay is then computed as

$$\begin{aligned} & \sigma_{\text{delay}_{\text{norm}}} \\ & = \sqrt{\alpha_L^2 \sigma_{L_{\text{norm}}}^2 + \alpha_R^2 \sigma_{R_{\text{norm}}}^2 + \alpha_{Jb}^2 \sigma_{Jb_{\text{norm}}}^2 + \alpha_{Ab}^2 \sigma_{Ab_{\text{norm}}}^2}. \end{aligned} \tag{2}$$

3) CNET: A special custom software program, CNET, has been developed for determining the optimal interconnect delays within the clock distribution network and the individual data paths. The program is applicable to large *fully synchronous* RSFQ circuits with all interconnections among cells composed of standard JTL's and splitters.

The following assumptions have been made.

1) The interconnect delays in the clock and data paths for a given pair of cells are correlated and change proportionally as a function of the global process parameter variations.
2) The effects of local parameter variations are neglected.
3) No correlations are assumed between:
   a) the delays of the interconnected cells;
   b) the delays of the cells and interconnections;
   c) the delays of the cells and their respective hold, setup, and minimum separation times.

The program solves a set of inequalities for each pair of interconnected cells in order to minimize the clock period and maximize any timing violation margins in the presence of timing parameter variations, as shown in Fig. 12. This optimization strategy is based on results presented in [77] and [36] and extended to consider limitations imposed by the minimum separation time between two input data pulses.

The output of the program includes:

1) the maximum clock frequency;
2) the number of standard JTL stages that must be added to the clock and data paths;
3) a specification of the most critical data paths;
4) timing violation margins defined as the minimum possible interval between the data pulse and the point where the timing constraint violation appears (see Fig. 12).

TABLE VI
SURVEY OF BASIC TOOLSETS USED BY THE ACADEMIC AND INDUSTRIAL SDE RESEARCH GROUPS

Groups	Layout tool	Circuit simulator	Optimizer	Inductance estimator	Logic simulator
Universities					
SUNY Stony Brook	Cadence Virtuoso & Diva	PSCAN	COWBOY	Lmeter	Leapfrog VHDL
U. of Rochester	Cadence Virtuoso & Diva	JSpice3	MALT	Icalc, Fast Henry	Verilog XL, Leapfrog VHDL
UC Berkeley	Xic, Magic	HSpice, Jspice3, JSim	HSpice	Maxwell 2D, 3D	—
U. of Ilmenau (Germany)	Magic, LASI, Autocad	Eldo	XOPT	MFB, SCIM	Eldo/Powerview
U. of Karlsruhe (Germany)	Cadence Virtuoso & Diva	Spice 3f4	ABAK	Sonnet em	Verilog-XL
Goverment Agencies and Institutes					
DERA (UK)	L-Edit	jsim_n	rtry	Lmeter	Lard
Companies					
TRW	Cadence Virtuoso & Diva	JSpice3	MALT	Fast Henry, IE3D	Leapfrog VHDL
ETL (Japan)	??	PSCAN	COWBOY	Lmeter	—
Northrop-Grumman	Xic, Magic	PSCAN, WRSpice	COWBOY, MALT	Lmeter	—
Hypres	Xic	PSCAN, WRSpice	COWBOY, MALT	Lmeter	Veriwell
Conductus	Kic, Autocad	WinS, JSpice3	WinS, MALT	Lmeter	—

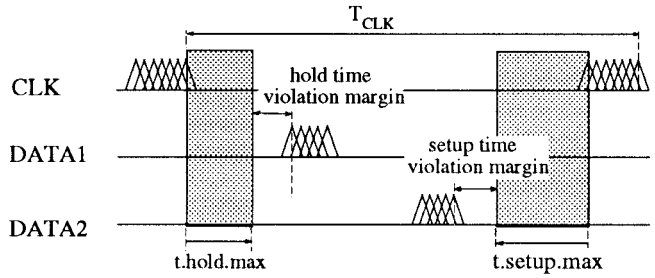


Fig. 12. CNET: Definition of the timing violation margins in the presence of variations in the fabrication process.

All outputs are obtained under the assumption that the timing parameters in the circuit remain within a 3σ range from their nominal values.

H. Toolsets

Due to the increasing complexity of RSFQ circuits, it has become clear that an integrated design environment is needed to assure adequate efficiency of the design process. Several such environments are presented in Table VI.

The most complete integrated design environments are based on the following tools for layout editing and circuit simulation:

1) Cadence and Jspice3;
2) Cadence and PSCAN;
3) Xic and WRSpice.

The first environment has been calibrated at the University of Rochester and independently at TRW. The second design environment has been developed at SUNY Stony Brook, and the third environment has been developed by Whiteley Research, Inc. and adopted by Hypres, Inc. and Northrop-Grumman.

IV. FUTURE TOOLS

Existing RSFQ CAD tools permit the development of small–to medium-scale circuits but do not offer any support for the design of large-scale and very large-scale circuits.

The following more advanced CAD tools remain to be developed.

A. Timing Optimizer

A *timing optimizer* is necessary to permit the interconnect delays in the clock and the data paths of a large circuit to be optimally determined. The optimum clock scheduling procedure is particularly complicated because active components such as Josephson Transmission Lines (JTL's) are used to implement interconnect delays. The delays of these interconnects are comparable in magnitude to the delays of logic gates. A timing optimizer must also consider a variety of clocking schemes and clock distribution network topologies developed specifically for RSFQ logic [36], [78]–[80]. Additional challenges for timing in the RSFQ technology include exceptionally high operating frequency and relatively large manufacturing induced parameter variations. The beta version of such an optimizer, called CNET, with limited capabilities and applicable to synchronous circuits only, has been developed at the University of Rochester and is described in Section III-G3.

B. Automated Logic Synthesizer

An *automated logic synthesizer* needs to be developed specifically for RSFQ logic, in order to deal with the completely different suite of basic gates, which does not include elementary combinational gates (all RSFQ gates are clocked). The synthesis process would preferably include the clock signal distribution. Additionally, the complexity of basic gates

differ in RSFQ technology; therefore, some gates may be preferred over others, according to criteria which differ from semiconductor logic.

C. Automated Layout Synthesizer

The development of an *automated layout synthesizer* for RSFQ logic is particularly challenging, in light of the use of active components for interconnects. Such interconnects have large delays and occupy a significant area on the chip. Therefore, any straightforward placement and routing scheme is likely to lead to multiple violations of timing constraints. As a result, a layout synthesizer for RSFQ logic must be integrated with the timing optimizer, and both invoked iteratively until a near optimum solution is determined.

Without the development of these prospective tools, it is difficult to imagine further progress toward very large-scale RSFQ technology, particularly its application to general purpose computing.

V. Trends in the Development and Use of SDE CAD Tools

Presently, no consensus exists regarding which particular SDE CAD tool is preferable at any stage of the design process. Certain features of existing tools often complement each other. An experience with using a certain tool and the time necessary to learn a new tool often outweigh advantages of switching to a new more advanced tool. As a result, even within the same company or university group, two or more tools of the same kind are used simultaneously by various designers and even by the same person. No strong attempt to standardize tools within the same institution has been observed.

The use of various tools may be advantageous in terms of exposing the designer to new experiences. However, it also has the disadvantage of making it more difficult to exchange designs when a close cooperation among designers from the same or cooperating institutions is required. This effect may be reduced by *standardizing data formats*. If the same standard data format can be used to store the information describing the circuit at each stage of the design process, then the data may be more easily exported from one tool and imported to another tool.

The following standard formats have been developed to facilitate the exchange of data among various *semiconductor* CAD tools:

- CIF (CalTech Intermediate Form) [81], [82] and GDS II (Calma GDS II Stream Format) [82] at the layout level;
- EDIF at the schematic level [55];
- Verilog HDL and VHDL at the behavioral level [67], [68].

The vast majority of existing layout tools used for the design of superconducting circuits support both CIF and GDSII. Tools for logic level simulation, adapted for RSFQ logic, such as Verilog XL, Leapfrog VHDL, Veriwell, and Eldo, also allow the transfer of behavioral models. Unfortunately, most schematic editors in use today do not support the reliable export and import of schematics using EDIF. The authors hope that this capability will be included in the new versions of schematic editors.

New trends can be observed in the development of circuit simulators. First, new types of analyzes targeted at supporting the design of superconducting digital and mixed-signal circuits have recently become available. These include Monte Carlo analysis, noise simulation [42], [43], error-rate simulation [83], and circuit animation [46]. Capabilities for mixed-mode simulation are being added to support the design of large digital and mixed-signal circuits. More circuit simulators will become closely integrated with the timing optimizers despite these optimizers being initially developed as stand-alone tools.

Together with the growing importance of logic simulation, it has become necessary to extract information about the timing parameters included in the behavioral models of the gates. In the authors' opinion, this function of timing analyzers (such as TAN) will need to be incorporated into the circuit simulators.

Further progress in optimization algorithms and their adaptation to superconducting digital circuits can be expected. It is likely that in the future, the figures of merit used during optimization will be extended to include both maximum circuit yield and optimum timing parameters. This will allow the maximum intrinsic speed of a gate to be achieved once the gate is optimized.

Although sophisticated tools for layout editing and verification of semiconductor circuits have been calibrated for superconducting electronics, the price limits their use to universities (which benefit from substantial educational discounts) and large companies. The authors expect that inexpensive tools targeted at superconducting electronics will become available and widespread among small-sized and medium-sized companies. These tools will possess most of the features of their expensive counterparts, including reliable extraction of connectivity and all component values and full LVS verification.

Accurate and fast inductance extractors are yet to be developed. These tools need to use a standard layout view of the circuit, either CIF or GDSII, as an input format. These tools should also combine the speed of existing 2-D inductance extractors while providing the accuracy of existing 3-D inductance estimators.

Finally, complete integrated design environments, with a straightforward design flow, supporting coherent and transferable design information and simplifying the use of standard cell libraries remain to be developed.

Acknowledgment

The authors would like to thank all participants of the SDE CAD tools survey, in particular, S. Whiteley, P. Dresselhaus, D. Zinoviev, S. Polonsky, S. Kaplunenko, M. Jeffery, Z. Du, R. Koch, M. Biehl, H. Toepfer, J. Satchell, and P. Shevchenko, who shared with the authors their knowledge and experiences regarding various tools they have developed, calibrated, or worked with. The authors appreciate the help and feedback they received from all participants of the working group on SDE CAD tools during the Workshop on Superconductive Electronics: Devices, Circuits & Systems in Winter Park, CO in September 1997. The authors are also grateful for valuable comments to the manuscript from Prof. K. Likharev,

H. Toepfer, and T. Harnisch. Special thanks belong also to our collaborators A. Herr, N. Vukovic, D. Brock, C.-H. Cheah, B. Guan, C. Mancini, and P. Rott, who participated in the design of the Rochester SDE CAD tools, devoted their time to debug these tools, used these tools for the first time, and gave the authors the feedback necessary to improve the capabilities of the Rochester SDE design environment.

REFERENCES

[1] K. K. Likharev, "Ultrafast superconductor digital electronics: RSFQ technology roadmap," *Czechoslovak J. Phys.*, Suppl. S6, vol. 46, 1996.

[2] K. K. Likharev, "Superconductors speed up computation," *Phys. World*, pp. 39–43, May 1997.

[3] K. Gaj *et al.*, "Toward a systematic design methodology for large multi-gigahertz rapid single flux quantum circuits," *IEEE Trans. Appl. Superconduct.*, to be published.

[4] M. Aoyagi *et al.*, "Chip layout design of a Josephson LSI circuit for examining high-speed operability by using a standard cell automatic placement and routing technique," *IEEE Trans. Appl. Superconduct.*, vol. 4, pp. 169–176, Sept. 1994.

[5] M. Khalaf, S. Whiteley, and T. Van Duzer, "A computer-aided design framework for superconductor circuits," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3341–3344, June 1995.

[6] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock frequency digital systems," *IEEE Trans. Appl. Superconduct.*, vol. 1, pp. 3–28, 1991.

[7] K. K. Likharev, "Rapid single-flux-quantum logic," in *The New Superconductor Electronics*, H. Weinstock and R. Ralston, Eds. Dordrecht: Kluwer, 1993, pp. 423–452.

[8] S. V. Polonsky *et al.*, "New RSFQ circuits," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 2566–2577, 1993.

[9] S. V. Polonsky, V. K. Semenov, and A. F. Kirichenko, "Single flux, quantum B flip-flop and its possible applications," *IEEE Trans. Appl. Superconduct.*, vol. 4, pp. 9–18, 1994.

[10] V. K. Semenov, Yu. Polyakov, and D. Schneider, "Preliminary results on the analog-to-digital converter based on RSFQ logic," in *CPEM Conf. Dig. Suppl.*, Braunschweig, Germany, June 1996, pp. 15–16.

[11] O. A. Mukhanov, P. D. Bradley, S. B. Kaplan, S. V. Rylov, and A. F. Kirichenko, "Design and operation of RSFQ circuits for digital signal processing," in *Proc. 5th Int. Superconductivity Electron. Conf.*, Nagoya, Japan, Sept. 1995, pp. 27–30.

[12] Q. P. Herr *et al.*, "Design and low speed testing of a four-bit RSFQ multiplier-accumulator," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3168–3171, 1997.

[13] O. A. Mukhanov, "RSFQ 1024-bit shift register for acquisition memory," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 3102–3113, Dec. 1993.

[14] A. M. Herr, C. A. Mancini, N. Vukovic, M. F. Bocko, and M. J. Feldman, "High-speed operation of a 64-bit circular shift register," *IEEE Trans. Appl. Superconduct.*, vol. 8, pp. 120–124, Sept. 1998.

[15] S. R. Whiteley, "Josephson junctions in SPICE3," *IEEE Trans. Mag.*, vol. 27, no. 2, pp. 2902–2905, Mar. 1991.

[16] S. V. Polonsky, V. K. Semenov, and P. N. Shevchenko, "PSCAN: Personal superconductor circuit analyzer," *Superconduct. Sci. Technol.*, vol. 4, pp. 667–670, 1991.

[17] Q. P. Herr and M. J. Feldman, "Multiparameter optimization of RSFQ circuits using the method of inscribed hyperspheres," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3337–3340, June 1995.

[18] P. I. Bunyk and S. V. Rylov, "Automated calculation of mutual inductance matrices of multilayer superconductor integrated circuits," *Ext. Abs. ISEC*, p. 62, 1993.

[19] A. Krasniewski, "Logic simulation of RSFQ circuits," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 33–38, Mar. 1993.

[20] K. Gaj, C.-H. Cheah, E. G. Friedman, and M. J. Feldman, "Functional modeling of RSFQ circuits using Verilog HDL," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3151–3154, June 1997.

[21] V. Adler, K. Gaj, D. K. Brock, C.-H. Cheah, and E. G. Friedman, "A Cadence-based design environment for single flux quantum circuits," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3294–3297, June 1997.

[22] H. Toepfer, T. Harnisch, J. Kunert, S. Lange, and H. F. Uhlmann, "Formal description of the functional behavior of RSFQ logic circuits for design and optimization purposes," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3630–3633, June 1997.

[23] 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. Superconduct.*, vol. 7, pp. 2685–2689, June 1997.

[24] V. K. Semenov, Y. A. Polyakov, and D. Schneider, "Implementation of oversampling analog-to-digital converter based on RSFQ logic," *Ext. Abs. ISEC*, pp. 41–43, 1997.

[25] *Proc. HTMT Project Meeting*, Univ. Delaware, July 21, 1997.

[26] Hybrid Technology Multithreaded Architecture Project Web Page. Available at http://www.cacr.caltech.edu/HTMT.

[27] D. L. Miller, J. X. Przybysz, A. H. Worsham, and A. H. Miklich, "Superconducting sigma-delta analog-to-digital converters," *Ext. Abs. ISEC*, pp. 38–40, 1997.

[28] O. A. Mukhanov and S. V. Rylov, "Time-to-digital converters based on RSFQ digital counters," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 2669–2672, June 1997.

[29] A. F. Kirichenko, O. A. Mukhanov, and S. V. Rylov, "Superconducting time-to-digital converters," *Ext. Abs. ISEC*, pp. 34–37, 1997.

[30] V. K. Semenov, Y. A. Polyakov, and A. Ryzhikh, "Decimation filters based on RSFQ logic/memory cells," *Ext. Abs. ISEC*, pp. 344–346, 1997.

[31] "Petaflops at SUNY stony brook," web page. Available at http://gamayun. physics.sunysb.edu/RSFQ/Research/PetaFLOPs.

[32] T. Sterling, P. Messina, and P. H. Smith, *Enabling Technologies for Peta(FL)OPS Computing*. Cambridge, MA: MIT, 1995.

[33] S. Whiteley, *JSPICE3—User's Manual*.

[34] *Hypres Niobium Process Flow and Design Rules*, Hypres, Inc., 175 Clearbrook Road, Elmsford, NY 10523 USA.

[35] K. Gaj, Q. P. Herr, and M. J. Feldman, "Analysis of timing requirements for basic RSFQ cells," presented at the *Applied Superconductivity Conf.*, Oct. 1994. Available http://www.ee.rochester.edu/users/sde/research/publications/asc94.

[36] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Processing*, vol. 9, pp. 247–276, 1997.

[37] K. Gaj, "Survey of SDE design tools." Available on the web under http://www.ee.rochester.edu/users/sde/cad/survey.html.

[38] H. Toepfer, H. Uhlmann, M. Knoll, H. Thiele, and M. Selent, "Design tools for parameter determination and simulation of integrated Josephson structures," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3345–3348, 1995.

[39] Odintsov, V. K. Semenov, and Zorin, "Specific problems of numerical analysis of the Josephson junction circuits," *IEEE Trans. Mag.*, vol. 23, pp. 763–766, 1987.

[40] O. Mukhanov, V. K. Semenov, and K. K. Likharev, "Ultimate performance of the RSFQ logic circuits," *IEEE Trans. Mag.*, vol. 23, pp. 759–762, 1987.

[41] M. Jeffery, private communication.

[42] J. Satchell, "Stochastic simulation of SFQ logic," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3315–3318, June 1997.

[43] H. Toepfer, T. V. Filippov, and H. F. Uhlmann, "RSFQ cells: Transient noise simulations," *Ext. Abs. ISEC*, pp. 317–319, 1997.

[44] D. K. Brock, private communication.

[45] C. Hamilton and K. Gilbert, "Margins and yield in single flux quantum logic," *IEEE Trans. Appl. Superconduct.*, vol. 1, pp. 157–163, Dec. 1991.

[46] J. Satchell, private communication.

[47] Q. P. Herr, "Bit errors and yield optimization in superconducting digital single-flux-quantum electronics," Ph.D. dissertation, Univ. Rochester, 1997.

[48] Q. Herr and K. Gaj, *MALT Reference Manual*. Available http://www.ee.rochester.edu/users/sde/research/software/malt/malt.reference.guide.html.

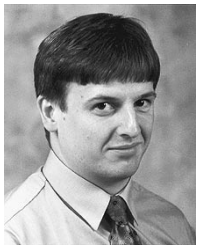
[49] T. Harnisch and H. F. Uhlmann, "Application of different design centering methods for yield optimization of cryoelectronic circuits," *Applied Superconductivity Inst. Phys. Conf.*, 1995, pp. 1717–1720.

[50] T. Harnisch, J. Kunert, H. Toepfer, and H. F. Uhlmann, "Design centering methods for yield optimization of cryoelectronic circuits," *IEEE Trans. Appl. Supercond.*, vol. 7, pp. 3434–3437, June 1997.

[51] R. Koch and W. Benzing, "Multi-dimensional parameter optimization for RSFQ circuit design," *Ext. Abs. ISEC*, pp. 305–307, 1997.

[52] S. Director and G. Hachtel, "The simplicial approximation approach to design centering," *IEEE Trans. Circ. Syst.*, vol. 27, pp. 363–372, July 1977.

[53] S. Director, W. Maly, and A. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*. Boston, MA: Kluwer, 1990.
[54] A. V. Rylyakov and S. V. Polonsky, "All-digital 1-bit RSFQ autocorrelator for radioastronomy applications: Design and experimental results," *IEEE Trans. Appl. Superconduct.*, Appendix, vol. 8, pp. 14–19, Mar. 1998.
[55] EDIF WWW Home Page at http://www.edif.org/index.html.
[56] W. H. Chang, "The inductance of a superconducting strip transmission line," *J. Appl. Phys.*, vol. 50, no. 12, pp. 8129–8134, Dec. 1979.
[57] M. J. Feldman, "Inductance calculation Rodeo, 1996–1997." Available on the web at http://www.ee.rochester.edu/users/sde/research/projects/Inductance/survey.html.
[58] B. Guan, M. J. Wengler, P. Rott, and M. J. Feldman, "Inductance estimation for complicated superconducting thin film structures with a finite segment method," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 2776–2779, June 1997.
[59] A. R. Kerr, "Surface impedance of superconductors and normal conductors in EM simulators," *Electronics Division Internal Rep. no. 302*, National Radio Astronomy Observatory, Charlottesville, VA, Feb. 19, 1996.
[60] Z. Du, S. R. Whiteley, and T. Van Duzer, "Inductance calculation of 3D superconducting structures," *Ext. Abs. ISEC*, pp. 293–295, 1997.
[61] P. H. Xiao, E. Charbon, A. Sangiovanni-Vincentell, T. Van Duzer, and S. R. Whiteley, "INDEX: An inductor extractor for superconducting circuits," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 2629–2632, Mar. 1993.
[62] G. Hildebrandt and F. H. Uhlmann, "Inductance calculation for integrated superconducting structures by minimizing free energy," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 2766–2769, June 1995.
[63] T. Nakazato and Y. Okabe, "Inductance computation of microscopic superconducting loop," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 3626–3629, June 1997.
[64] M. M. Khapaev, "Extraction of inductances of a multi-superconductor transmission line," *Superconduct. Sci. Technol.*, vol. 9, pp. 729–733, 1996.
[65] ———, "Extraction of inductances of plane thin film superconducting circuits," *Superconduct. Sci. Technol.*, vol. 10, pp. 389–394, 1997.
[66] P. Bunyk, A. Y. Kidiyarova-Shevchenko, and P. Litskevitch, "RSFQ microprocessor: New design approaches," *IEEE Trans. Appl. Superconduct.*, Appendix, vol. 7, pp. 2697–2704, June 1997.
[67] Standard 1076–1993, *IEEE Standard VHDL Language Reference Manual*, IEEE, 1994.
[68] IEEE 1364 Standard, *Verilog Hardware Description Language Reference Manual (LRM)*.
[69] D. Perry, *VHDL*. New York: McGraw-Hill, 1991.
[70] S. Palnitkar, *Verilog HDL—A Guide to Digital Design and Synthesis*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[71] A. M. Dewey, *Analysis and Design of Digital Systems with VHDL*. New York: PWS, 1997.
[72] J. R. Armstrong, *Chip-Level Modeling with VHDL*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[73] U. Golze, *VLSI Chip Design with the Hardware Description Language VERILOG*. New York: Springer-Verlag, 1996.
[74] D. Pellerin and D. Taylor, *VHDL Made Easy!* Englewood Cliffs, NJ: Prentice Hall, 1997.
[75] P. Ashenden, *The VHDL Cookbook*. Available at ftp://bears.ucsb.edu/pub/VHDL/comp.lang.vhdl.
[76] K. Gaj, Q. P. Herr, and M. J. Feldman, "Parameter variations and synchronization of RSFQ circuits," *Applied Superconductivity Conf.*, Institute of Physics, Bristol, U.K., 1995, Conf. Series #148, pp. 1733–1736.
[77] K. Gaj, E. G. Friedman, M. J. Feldman, and A. Krasniewski, "A clock distribution scheme for large RSFQ circuits," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3320–3324, 1995.
[78] K. Gaj, "Timing of large RSFQ circuits," collection of slides available at http://www.ee.rochester.edu/users/sde/research/projects/rsfq/timing/title.html.
[79] Z. J. Deng, N. Yoshikawa, S. R. Whiteley, and T. Van Duzer, "Data-driven self-timed RSFQ digital integrated circuit and system," *IEEE Trans. Appl. Superconduct.*, vol. 7, 1997.
[80] P. Patra, S. Polonsky, and D. Fussel, "Delay insensitive logic for RSFQ superconductor technology," *III Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1996, pp. 42–53.
[81] C. A. Mead and L. A. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1987, p. 115.
[82] S. M. Rubin, *Computer Aids for VLSI Design*. Reading, MA, Addison-Wesley, 1987, p. 275.
[83] J. Satchell, "Margins and errors in SFQ circuits," *Ext. Abs. ISEC*, pp. 44–46, 1997.



Kris Gaj received the M.Sc. and Ph.D. degrees in electrical and computer engineering from Warsaw University of Technology, Poland, in 1988 and 1992, respectively.

In 1991 he was a Visiting Scholar at Simon Fraser University in Vancouver, Canada, where he worked on analysis of various BIST techniques for VLSI digital circuits. From 1992 to 1993, he headed a research team at Warsaw University of Technology developing an implementation of the Internet standard for secure electronic mail, Privacy Enhanced Mail. From 1994 to 1998, he was with the Department of Electrical Engineering at the University of Rochester, where he worked on logic level design, design automation and timing analysis of superconducting digital circuits. In 1998, he joined George Mason University as an Assistant Professor. His current research interests include CAD tools, testing, cryptography, computer arithmetic, and hardware/software co-design. He is the author of the book on codebreaking.

Quentin P. Herr received the B.S. degree in physics from Case Western Reserve University in 1992 and the M.A. degree in physics and Ph.D. degree in electrical engineering from the University of Rochester.

His research interest at Rochester focused on single-flux-quantum electronics, including yield optimization, design for high-speed operation, and bit error-rate measurement of digital circuits. Since 1997, he has continued this research interest as a member of the Superconductor Electronics Organization, TRW, Inc., located in Redondo Beach, CA.

Victor Adler received the B.S. degree in electrical engineering and the B.A. degree in computer science from Duke University, Durham, NC, in 1992, the M.S. degree in electrical engineering from the University of Rochester, Rochester, NY, in 1993, and the Ph.D. degree in electrical engineering at the University of Rochester in 1998.

He was an IBM Watson Scholar and worked preprofessionally at IBM Microelectronics, Burlington, VT, between 1988 and 1992 in the areas of final module test, packaging, circuit macro development, and standard cell design. He has been a Teaching and Research Assistant at the University of Rochester since 1993. In 1997, he worked on clock skew scheduling at Intel Corp. In 1998, he joined Sun Microsystems. His research interests include design techniques for high-performance CMOS and superconductive technologies.

Andy Krasniewski (M'95-SM'91–F'96) received the M.S., Ph.D., and D.S. (Doctor of Science) degrees in electronic engineering from the Warsaw University of Technology, Poland, in 1974, 1983, and 1989, respectively.

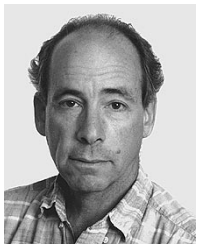
Since 1974 he has been with the Warsaw University of Technology, where he is currently a Professor and Associate Dean at the Faculty of Electronics and Information Technology. From 1990 to 1992, he was a Visiting Associate Professor at the Department of Electrical Engineering, University of Rochester, Rochester, NY. His research interests include testing and design for testability, fault-tolerant computing, and superconducting digital electronics.

Dr. Krasniewski is a member of the ACM, and a member of the AAUP.

Eby G. Friedman received the B.S. degree from Lafayette College in 1979 and the M.S. and Ph.D. degrees from the University of California, Irvine, in 1981 and 1989, respectively, all in electrical engineering.

From 1979 to 1991, he was with Hughes Aircraft Company, rising to the position of Manager of the Signal Processing Design and Test Department, responsible for the design and test of high-performance digital and analog IC's. He has been with the Department of Electrical and Computer Engineering at the University of Rochester since 1991, where he is a Professor and the Director of the High Performance VLSI/IC Design and Analysis Laboratory. His current research and teaching interests include high performance microelectronic design and analysis with application to high-speed portable processors and low-power wireless communications. He is the author of more than 100 papers and book chapters and the editor of three books in the fields of high-speed and low-power CMOS design techniques and the theory and application of synchronous clock distribution networks.

Dr. Friedman is a Member of the editorial board of *Analog Integrated Circuits and Signal Processing*, a member of the CAS BoG, and a Member of the technical program committee of a number of conferences. He was a Member of the editorial board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, Chair of the VLSI Systems and Applications CAS Technical Committee, Chair of the Electron Devices Chapter of the IEEE Rochester Section, Chair of the VLSI track for '96 and ISCAS'97, Technical Co-Chair of the 1997 IEEE International Workshop on Clock Distribution Networks, Editor of several special issues in a variety of journals, and a recipient of the Howard Hughes Masters and Doctoral Fellowships, an IBM University Research Award, an Outstanding IEEE Chapter Chairman Award, and a University of Rochester College of Engineering Teaching Excellence Award.



Marc J. Feldman received the Ph.D. degree in physics from the University of California, Berkeley, in 1975.

He worked at Chalmers University, Sweden, and at the NASA/Goddard Institute for Space Studies, New York City, in the development of superconducting receivers for radio astronomy observatories. He joined the faculty of Electrical Engineering at the University of Virginia in 1985, where he developed a variety of superconducting diodes for receiver applications. He is now Senior Scientist and Professor of Electrical Engineering at the University of Rochester. His current research activities are directed toward the development of ultra-high-speed large-scale digital circuits using superconducting single-flux-quantum logic.