Toward a Systematic Design Methodology for Large Multigigahertz Rapid Single Flux Quantum Circuits

Kris Gaj, Quentin P. Herr, Victor Adler, Darren K. Brock, Eby G. Friedman, and Marc J. Feldman

***Abstract*— Rapid single flux quantum (RSFQ) digital circuits have reached the level of medium- to large-scale of integration. At this level, existing design methodologies, developed specifically for RSFQ circuits, have become computationally inefficient. Applying mature semiconductor methodologies to the design of RSFQ circuits, one encounters substantial difficulties originating from the differences between both technologies. In this paper, a new design methodology aimed at large-scale RSFQ circuits is proposed. This methodology is based on a semiconductor semicustom design approach. An established design methodology for small-scale RSFQ digital circuits, based on circuit (junction-level) simulation and device parameter optimization, is used for the design of basic RSFQ cells. A library composed of about 20 basic RSFQ cells has been developed based on this approach. A novel design methodology for large-scale circuits, presented in this paper, is based on logic (gate-level) simulation and timing optimization. This methodology has been implemented around the Cadence integrated design environment and used successfully at the University of Rochester for the design of two large-scale digital circuits.**

***Index Terms*— CAD, design methodology, optimization, physical implementation, RSFQ superconducting electronics, synthesis, timing.**

I. Introduction

ONE of the primary problems in the development of large rapid single flux quantum (RSFQ) circuits [1], [2] is the lack of appropriate design methodologies that effectively utilize computer-aided design (CAD) tools [3] while providing direction for the development of new tools specific to this superconducting technology. It is controversial whether *superconducting* RSFQ circuits should be designed based on leveraging techniques developed for *semiconductor* circuits or whether completely new methodologies specific to RSFQ logic should be created. The proponents of the former approach note the analogies between both technologies, particularly strong at the system level, and stress the achievements and maturity of semiconductor technologies. The proponents of the latter approach stress the substantial differences between the two technologies, particularly strong at the circuit level, and the large difference in the operating speed, power consumption, and fabrication process. A combination of both strategies is clearly the most effective. This paper is intended to summarize recent developments and experiences describing design methodologies for SFQ circuits.

The main feature of the methodologies described in this paper is the application of different design flows and tools depending upon the size and functional complexity of the circuit. The design methodology for small-scale circuits is centered around circuit simulation and the optimization of device parameters. The design methodology for large-scale circuits is centered around logic (gate-level) simulation and optimization of the interconnect delays within the circuit.

The second important feature is the development of libraries composed of basic RSFQ cells, permitting the design of circuits of arbitrary complexity. This process of constructing large RSFQ circuits out of a general family of primitive gates has not been commonly accepted. The main obstacles to this approach are:

- difficulty of isolating RSFQ gates from each other;
- large uncertainty of delays and other timing parameters of RSFQ cells due to variations in the fabrication process and changes in the bias currents;
- use of Josephson transmission lines (JTL's) for interconnects;
- low fanout of RSFQ gates;
- lack of a well-established methodology for modeling the timing of RSFQ circuits;
- lack of tools for the timing analysis and timing optimization of RSFQ circuits;
- lack of tools to logically simulate RSFQ circuits.

Techniques for overcoming these obstacles are described in Sections III and IV.

The methodology presented in this paper, while still consistent with a *full custom* approach, supports the design of RSFQ circuits based on a *semicustom* standard cell approach. According to this application specific integrated circuits (ASIC) design methodology, a library of standard RSFQ gates is first created, permitting the multiple and repeated use of these cells for the design of large-scale digital circuits.

The basic methodology for the design of semiconductor circuits is summarized in Section II. The differences between RSFQ and semiconductor logic and their influence on the choice of RSFQ design methodologies are discussed as well. The methodology currently in common use for the design of basic RSFQ cells is presented in Section III. This overview is

Manuscript received May 21, 1998; revised March 9, 1999.

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, Redondo Beach, CA 90278 USA.

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

D. K. Brock is with HYPRES Inc., Elmsford, NY 10523 USA.

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

Publisher Item Identifier S 1051-8223(99)08092-6.

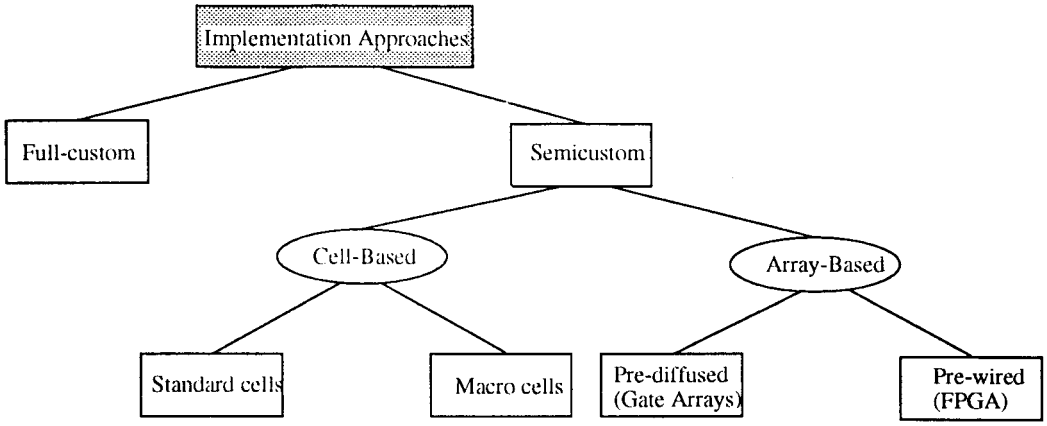


Fig. 1. Implementation approaches for the design of semiconductor digital circuits.

followed in Section IV by the description of the methodology used for the design of large-scale RSFQ circuits. In Section V, examples of applying these two methodologies to the design of various size RSFQ circuits are described, followed by some concluding comments in Section VI.

II. Superconductor Versus Semiconductor Design Methodologies

A. Semiconductor Design Methodologies

Several distinct methodologies have been developed for the design of large *semiconductor* digital circuits as shown in Fig. 1 [4]. The choice of methodology depends upon a number of conflicting factors such as performance (in terms of speed and power consumption), area, cost, production volume, and time-to-market. The methodologies and tools used in each of these various design methodologies have different cost/benefit tradeoffs with respect to these design goals.

In a *full custom* approach, the entire circuit is developed from the basic transistor level. Most stages of the design process are not fully automated, requiring manual optimization to achieve the highest performance. The design time is long, and the design is labor intensive, and therefore costly. These features may be justified if the circuit is produced in a sufficiently large volume (as in circuits such as microprocessors and memories) or the performance is the primary consideration, as in certain military applications.

In a *semicustom* approach, the circuit is composed of predefined structures such as standard gates or arrays of cells. Most stages of the design process are fully or partially automated, considerably reducing the design time and cost. However, the performance of the circuit is considerably degraded.

A very popular semicustom approach is a cell-based standard cell approach in which the circuit is built from a limited library of standard cells. The design flow for this methodology is shown in Fig. 2. The behavior of the circuit is described initially at a functional level, typically using a hardware description language (HDL), such as Verilog HDL [5] or VHDL [6]. The structure of the circuit is generated automatically from the behavioral model using automated logic synthesis. After the function of the circuit is verified using logic simulation, the timing is optimized using specialized analysis tools. The physical view of the circuit is created using tools for automated layout synthesis. The physical layout of the circuit is verified with design rule checking (DRC), electrical rule checking (ERC), and layout versus schematic (LVS) verification tools. The extraction of the circuit netlist from the layout permits parasitic components to be back annotated into the circuit simulation. The function of the circuit with extracted parasitics is verified next, and the timing is reoptimized if necessary. The entire design process is iterated until the required performance is achieved, and no meaningful discrepancies between the schematic of the circuit and the netlist extracted from the physical layout are flagged by the LVS verification process.

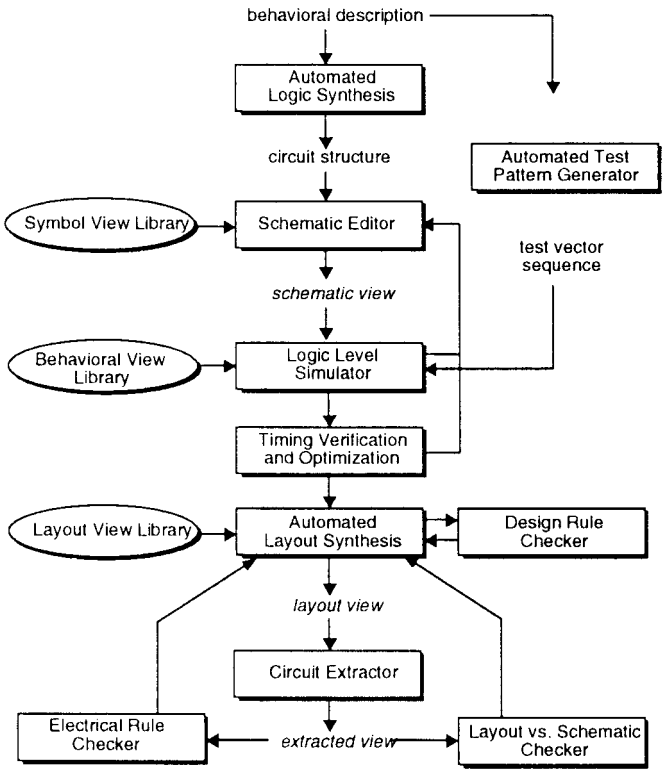


Fig. 2. Flow chart of the CAD process for semiconductor circuits based on standard-cell design methodology.

B. Adopting Semiconductor Semicustom Methodology to RSFQ Circuits

The design of large RSFQ circuits is currently based almost exclusively on a full custom methodology. This approach is justified by the immature state of RSFQ technology and by the niche applications of RSFQ circuits, such as time-to-digital converters [7] or decimation filters for analog-to-digital converters [8], where the advantages in performance, in terms of both speed and power, are of primary importance. Nevertheless, this design style also leads to long design times and significant design effort. Further development of RSFQ technology and its application to digital signal processing [8] and general purpose computing [9], [10] requires adopting a more labor efficient and less error-prone semicustom design methodology. Unfortunately, differences between superconducting RSFQ logic and traditional semiconductor technologies prevent the direct application of semiconductor methods and tools to the automated design of RSFQ circuits.

Semiconductor tools used to perform *automated logic synthesis*, *logic simulation*, and *timing analysis* are incompatible with the RSFQ logic convention and RSFQ suite of basic gates. RSFQ logic is based on pulses rather than voltage or current levels. The RSFQ convention for the representation of logic states requires that most of the logic components, including basic gates such as NOT, AND, and OR, are synchronous (clocked) rather than combinational as in semiconductor logic. These cells combine the logic function with a storage capability. The semiconductor counterpart is a combinational gate followed by a D flip-flop. As a result, the set of elementary RSFQ gates is substantially different from the set of basic semiconductor gates, and the relative complexity of corresponding cells in both technologies differ significantly. Due to these differences, tools for the automated logic synthesis of semiconductor circuits cannot be directly applied to the synthesis of RSFQ circuits. Calibrating these tools for RSFQ logic, if at all possible, would require a major effort. Particularly challenging is the automated design of the clock distribution network in synchronous RSFQ circuits.

The effective use of logic simulators for semiconductor circuits is based on a standard library of elementary semiconductor gates distributed by a simulator vendor or available from a semiconductor foundry. Currently, no such libraries exist for RSFQ technology. Nevertheless, a majority of standard simulators permit the creation of user-specified models of gates, permitting the development of entirely new model libraries. User specified models are most easily created using HDL's, such as Verilog HDL and VHDL. Both of these languages can be used to describe the functional behavior and detailed timing characteristics of an RSFQ gate [11].

Tools for the timing analysis and optimization of semiconductor circuits are not well suited for RSFQ logic. First, semiconductor tools are developed to deal primarily with *synchronous* circuits comprised of large blocks of *combinational logic* separated by registers. Secondly, timing constraints are more rigid for multigigahertz RSFQ circuits compared to subgigahertz semiconductor circuits. Finally, relative variations of timing parameters are larger in RSFQ due to the immaturity of the superconducting fabrication process.

Basic semiconductor tools for layout editing and verification are quite easily adapted to work with new technologies including superconducting technologies [3]. The calibration process typically includes writing a technology file in which the properties and rules of the fabrication process are described. This ease of calibration, however, does not apply to automated layout synthesis. Most interconnections among RSFQ gates are composed of active elements—JTL's. These circuits add substantial delay to the clock and data paths. In semiconductor technologies, most interconnections are composed of passive metal lines with extra components such as buffers or repeaters to control the behavior of the signals propagating through the interconnect. The different physical nature of interconnect and their different timing characteristics cause automated layout synthesis in both technologies to be based on a different set of rules and algorithms. In particular, the extremely strong interrelation between timing optimization and physical layout is of fundamental importance in RSFQ circuits.

Due to these aforementioned difficulties and the small to medium scale of RSFQ circuits developed to date, research on semicustom design methodologies for RSFQ circuits has only just started. The semicustom design style based on primitive libraries has only recently been demonstrated by Hypres Inc., where several medium-sized circuits have been developed from previously designed cells [7]. The purpose of this paper is to review the methodology development accomplished at the University in Rochester on adapting semiconductor design strategies to RSFQ logic and determine the directions in which this effort is expected to continue into the future.

III. Design Methodology for Small-Scale RSFQ Circuits

The design flow of a small-scale RSFQ circuit, such as a basic RSFQ cell, is shown in Fig. 3. The design process consists of three main phases: synthesis of the circuit structure, optimization of the circuit parameters, and physical implementation of the circuit layout. This design flow is commonly accepted and supported by multiple commercial and public domain CAD tools [3]. The primary phases of this well-established design methodology are reviewed below in order to introduce the design methodology for large-scale circuits discussed in Section IV. The CAD tools calibrated or developed at the University of Rochester which are utilized at particular phases of the design process for small-scale RSFQ circuits are also introduced in this section.

A. Synthesis

The synthesis of a small-scale RSFQ circuit begins with a description of the circuit function using a Mealy state transition diagram or present-state/next-state table. An example of a Mealy diagram for an AND gate is shown in Fig. 4. The circuit structure along with a set of near exhaustive input stimuli used to test the functional behavior of the circuit are derived from

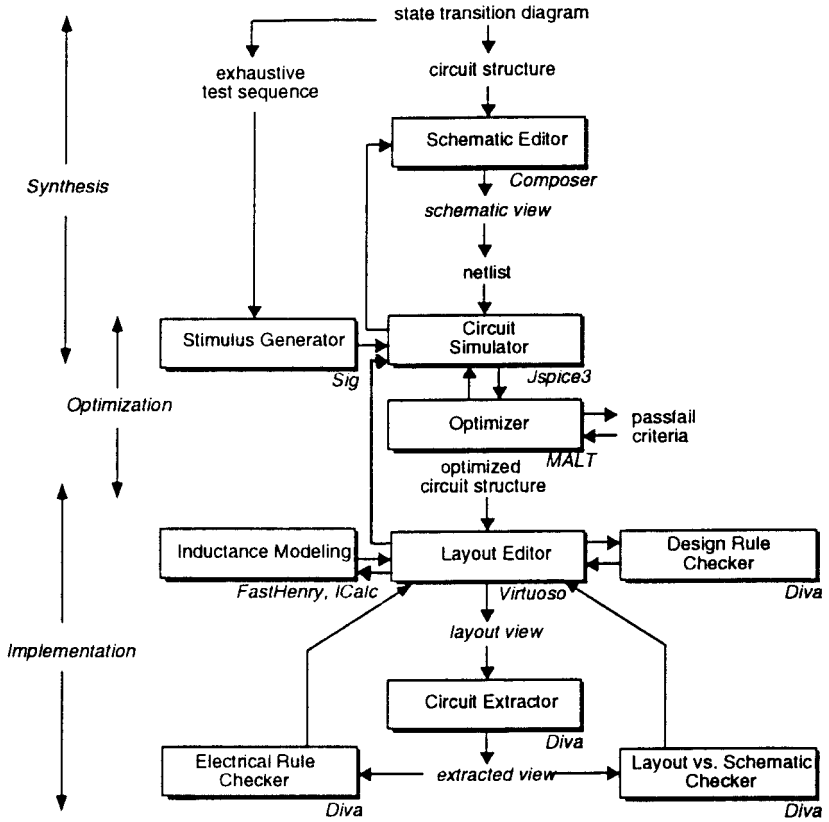


Fig. 3. Flow chart of the CAD process for small RSFQ circuits.

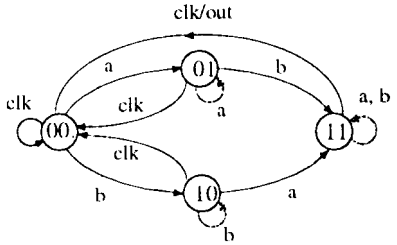


Fig. 4. Mealy state transition diagram of an AND gate.

the circuit function. This step is currently not automated and is primarily dependent on designer intuition.

The *schematic view* of the circuit is created from the circuit structure. At Rochester, the *Cadence Composer* schematic entry editor is used for this purpose, as shown in Fig. 5. The *netlist view* of the cell in JSpice3 notation is generated automatically through customized translation software. The input test stimuli are described initially in a simplified notation and translated automatically into JSpice3 format using SIG [3]. The netlist and the input stimuli together constitute the input deck for *JSpice3*. The circuit is simulated and modified iteratively to verify full functionality for all test sequences.

B. Optimization

The custom optimization package *MALT* developed at the University of Rochester is used to determine the optimal nominal values of the circuit device parameters that achieve the maximum yield [3], [12], [13]. Before the optimization procedure can begin, a pass–fail criteria is generated to permit distinguishing between sets of operating parameters that give correct and incorrect circuit functionality. These criteria are generated *automatically* by simulating the circuit for the set of initial operating parameters [3]. This pass–fail criteria only consider the externally observed behavior of the circuit, i.e., sequences of pulses at the inputs and outputs of the cell.

C. Implementation

After the optimum values of the device parameters are determined, the *layout view* of the cell is drawn manually, as exemplified by Fig. 6. At this level of abstraction, the circuit is described in terms of the physical geometric data used to produce the individual lithographic masks. The target technology used by the Rochester group is the low-temperature superconducting IC process provided by Hypres Inc., specifically, a four-metal layer, ten-level, $Nb/Al_2O_3/Nb$ tri-layer process [14]. The graphic layout environment supported by *Cadence Virtuoso* has been calibrated to support this technology. The physical dimensions of each circuit inductor are calculated via a custom graphical user interface, *Icalc* from the look-up tables precomputed by a modified three-dimensional inductance calculation package, *FastHenry* [3], [15]. This method permits a more accurate estimation of the physical dimensions of each inductor as compared with average inductance per square calculations.

Assuring that the physical layout meets the minimum design rule specifications of the fabrication process is accomplished by an automated *design rule checker (DRC)*. In an interactive

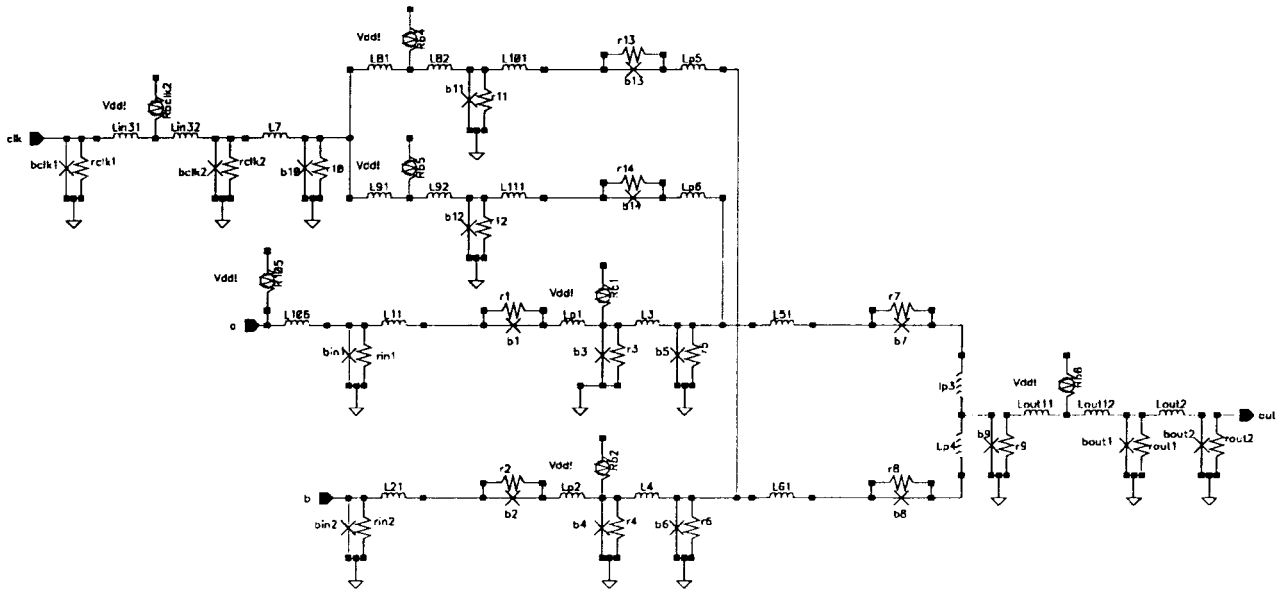


Fig. 5. Schematic diagram of the AND gate created using Cadence Composer.

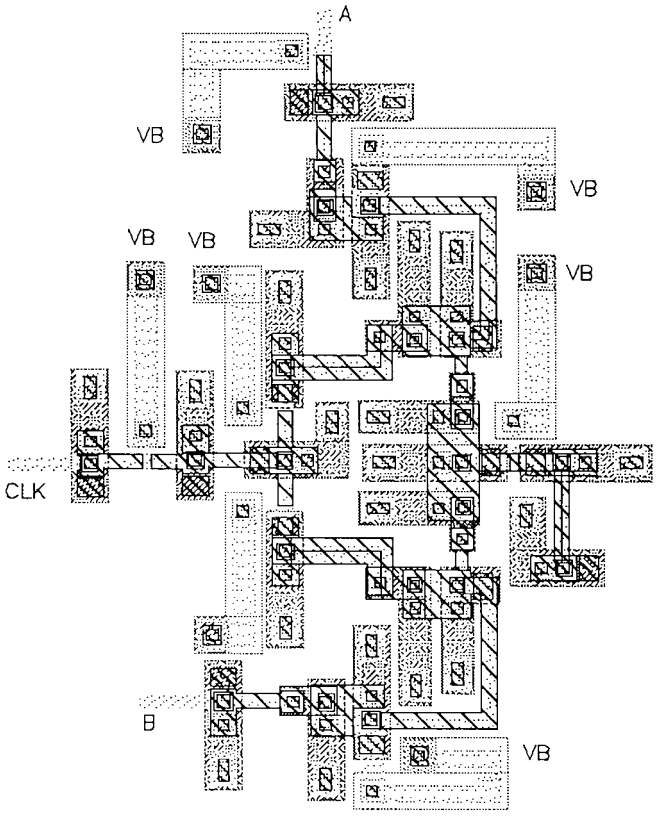


Fig. 6. Portion of the AND gate layout created using Cadence Virtuoso layout editor.

process, errors are flagged during physical layout, permitting the violations to be immediately corrected, as shown in Fig. 7. The design rules themselves are calibrated for same-layer spacing, different-layer spacing, minimum width, on-grid alignment, and layer enclosure [14].

Once the circuit has been physically laid out and verified using DRC, an *extracted view* of the cell can be generated automatically using the graphical data, as shown in Fig. 8. The device extraction process recognizes specific junctions, resistors, and inductors by Boolean combinations of mask layers using rules provided in a technology file. The location of each device is noted and a circuit netlist is created, describing the circuit connectivity. During the extraction process, the physical parameters of each device are attached to its instance using additional calibrated features of *Cadence Diva*. Two graphic layout layers have also been added to the standard ten-layer mask set in order to identify circuit inductors, delineating the desired inductors from the stray parasitic inductances present in all superconductive interconnect. After the preliminary layout of the basic cell is completed, parasitic inductances are marked with the additional mask layers and the schematic view updated accordingly.

The circuit extracted from the physical layout is verified by *electrical rule checking (ERC)* to assure that the circuit does not contain any electrical errors (e.g., power/ground shorts or unconnected floating nodes). ERC is a precursor to *layout-versus-schematic (LVS)* verification. With LVS, the extracted physical layout is checked directly against a schematic description by comparing the netlists generated from each corresponding view, permitting any errors to be flagged. As currently calibrated, the environment recognizes design errors with a precision of 1% for junction areas, 2% for shunt and bias resistors, and 10% for inductors.

If the parasitic inductances in the circuit are relatively large, the circuit may require being reoptimized while taking these inductances into account. The second iteration of the optimization procedure is typically sufficient to conclude the design process for a small-scale circuit.

IV. Design Methodology for Large-Scale Circuits

The established design methodology for small-scale RSFQ circuits described in the previous section cannot be easily adopted to large-scale RSFQ circuits, i.e., circuits composed

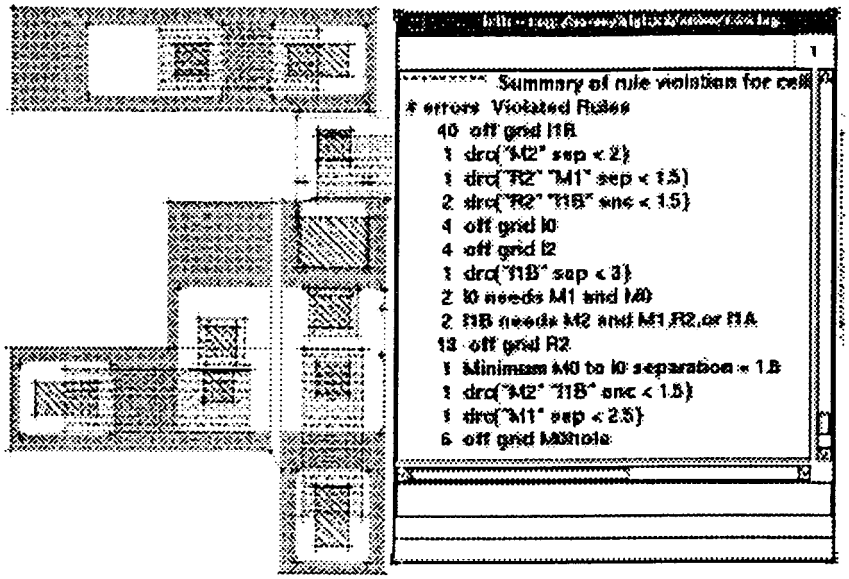


Fig. 7. DRC errors marked in the layout and explained in the information window of Cadence Diva.

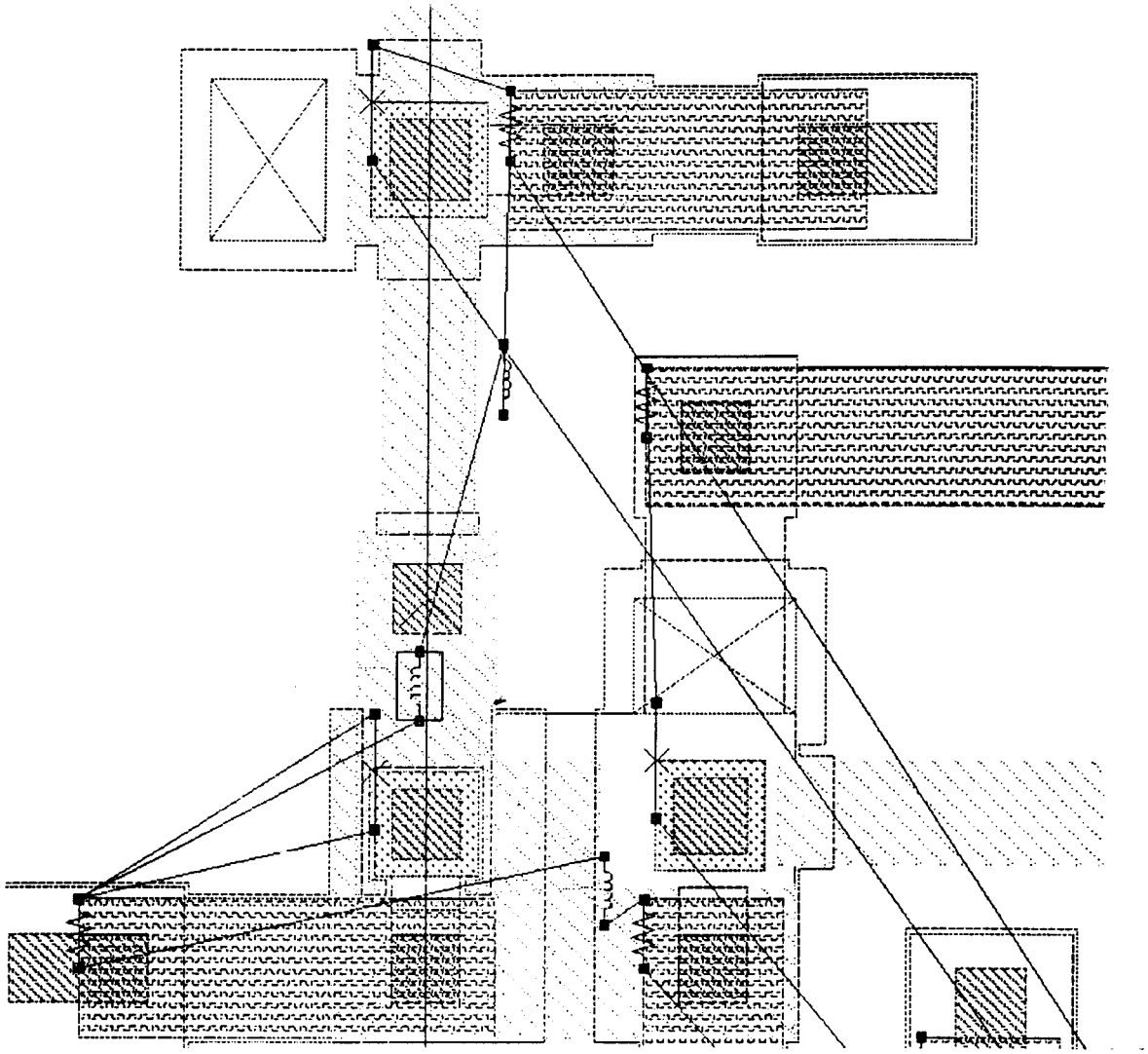


Fig. 8. Extracted view of the portion of AND gate layout obtained using Cadence Diva.

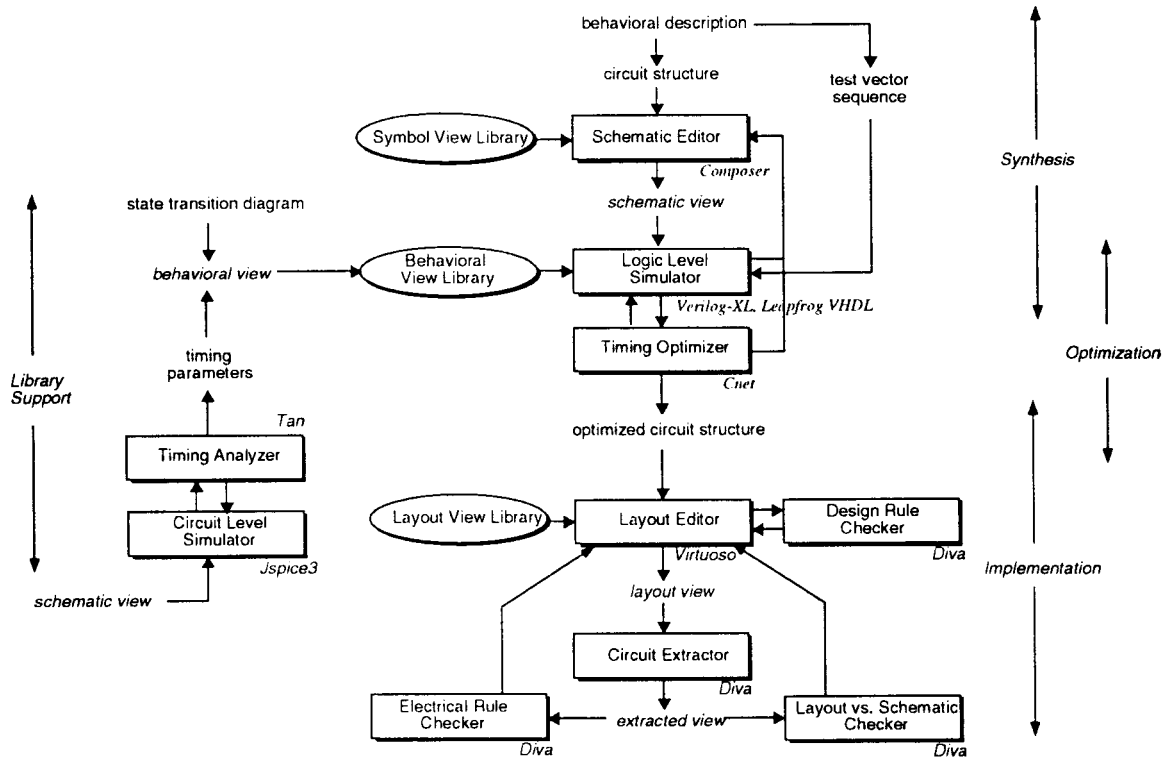


Fig. 9. Flow chart of the CAD process for large RSFQ circuits.

of thousands of Josephson junctions and hundreds of basic RSFQ gates. The main obstacle is the prohibitive time required to optimize a circuit, which is determined primarily by the computational time required to perform the circuit (or junction-level) simulation. Even assuming significant improvements in the simulation algorithms [16], the design process becomes extremely time consuming as soon as the circuit reaches a medium scale of integration with several hundred Josephson junctions. The computational time required to simulate a circuit is significant since the shapes of voltage, current, and phase waveforms in all nodes of the circuit are determined by solving a set of complex differential equations. Additionally, for many optimization algorithms [12], [13], the time required to optimize a circuit grows polynomially or even exponentially as a function of the number of parameters, leading to limitations on the number of parameters that can be optimized simultaneously.

In our methodology, a large RSFQ circuit is constructed out of basic gates, optimized separately. No device parameter optimization or even circuit (junction-level) simulation is required once the library of basic cells is created. A large-scale circuit is simulated at the gate level rather than the junction level using a logic instead of a circuit simulator. This choice reduces the simulation time by two to three orders of magnitude. The only goal of the optimization process is to choose the optimum timing scheme (e.g., synchronous versus asynchronous), and the optimum values of the interconnect delays. Interconnects among the gates are composed of standard JTL lines. The delays of these interconnects are controlled by changing the number of basic JTL stages comprising an interconnect. The timing optimization procedure determines the appropriate number of JTL stages for each interconnect in the circuit. The entire design flow used at Rochester for large-scale RSFQ circuits is shown in Fig. 9 and is discussed in detail in this section.

A. Library Support

The design of a large circuit in this methodology is based on the use of a library of basic gates. This library can be developed for the purpose of designing a specific large-scale circuit, leading to a full custom design approach, or it can be in common use for many circuits, leading to a semicustom standard cell approach. Each gate is represented in the library by four basic views: *schematic*, *layout*, *symbol*, and *behavioral*. Only the layout view of the gate may be modified when the cell becomes a part of a larger circuit, and the modifications should not change any of the inductances within the gate. An example is if the gate inputs and outputs are reoriented to better fit within the larger circuit. The other three views of the gate remain unchanged.

The *schematic* and *layout views* of each cell are created as part of the design process for a small circuit, as described in Section III. These views are sufficient for documenting and fabricating a single gate. When designing a larger circuit, the circuit schematic and its physical layout become prohibitively complicated. The design process can be simplified and made more efficient if both the schematic and layout views are represented hierarchically. For medium-scale circuits two levels of hierarchy are sufficient. At the higher logic level, the circuit is composed of basic cells, each represented by a *symbol view* of the cell. The symbol view is a graphical representation of the gate which includes information describing the gate

inputs and outputs. At the lower circuit level, each basic cell is composed of elementary devices such as Josephson junctions, inductors, resistors, and current sources. The circuit represented using hierarchical schematic can be simulated using a circuit simulator such as JSPICE. Nevertheless, the simulation time does not change compared to simulating a flat schematic. To substantially reduce the simulation time, a completely different simulation method must be applied. This new method, specifically logic simulation, was first introduced for RSFQ circuits in [17] and further improved in [11].

To make logic simulation possible, a behavioral view must be created for each basic gate in the library. The behavioral view describes the logic function and the timing characteristic of the gate. An acceptable way of creating a behavioral view, adapted from semiconductor circuit design, is the use of an HDL, such as Verilog HDL [5], [18] and VHDL [19], [6].

The Rochester design environment supports the use of both of these languages for behavioral modeling and simulation of RSFQ circuits [3], [11]. The initial version of a behavioral view is created directly from the cell specification (e.g., a Mealy diagram) and does *not* require any information about the internal structure of the circuit. This view is verified using one of the two Cadence simulators, Verilog-XL for Verilog HDL and Leapfrog VHDL for VHDL. A near exhaustive set of input sequences, including test sequences with timing violations (e.g., violations of hold or setup time), is used to analyze a circuit. Behavioral models of the RSFQ confluence buffer in Verilog HDL and VHDL are shown in Figs. 10 and 11, respectively. After the internal circuit structure is optimized, exact values of the timing parameters of the cell (i.e., the propagation delay, hold time, and setup time) specific for a given implementation of the cell can be determined from a circuit-level simulation. This process is automated by a program called the timing analyzer (TAN). TAN is a tool that enables calculating timing parameters of RSFQ cells, such as the hold time, the setup time, and the minimum separation time, and permits estimating the standard deviation of the propagation delays as a function of variations in the fabrication process [3]. These timing parameters and their estimated variations due to variations in the fabrication process are included in the behavioral model of the cell.

```
module conf_buff (in1, in2, out);
input
        in1, in2;
output
        out;
reg
        out;

parameter
        t_separation = 10,
        delay=15;

reg
        sep_clr;    // signal marking the end of the min separation zone
                    //  for the last pulse

integer
        last_in_time,       // time when the last t pulse appeared
        in_sep;             // number of pulses within an interval t_sep before
                            //  the last data pulse;

initial
        begin
                last_in_time = 0;
                in_sep = 0;
                sep_clr = 0;
                out = 0;
        end

always @(posedge in1)
        begin
                out <= #delay ((in_sep > 0) || (in2 == 1)) ? 1'bx : 1;
                out <= #(delay+2) 0;
                last_in_time <= $stime;
                in_sep <= in_sep + 1;
                sep_clr <= #(t_separation-1) 1;
        end

always @(posedge in2)
        begin
                out <= #delay ((in_sep > 0) || (in1 == 1)) ? 1'bx : 1;
                out <= #(delay+2) 0;
                last_in_time <= $stime;
                in_sep <= in_sep + 1;
                sep_clr <= #(t_separation-1) 1;
        end

always @(posedge sep_clr)
        begin
                in_sep <= in_sep - 1;
                sep_clr <= 0;
        end

endmodule
```

Fig. 10. Verilog HDL behavioral model of a confluence buffer.

B. Synthesis

The design process of a large-scale RSFQ circuit begins with a behavioral description. The description may be informal using a text specification or mathematical formulas, or more formal, e.g., based on a HDL. The design of the circuit structure is performed manually. Any automation of this process is yet to be developed. Gates already available in the library, or described in the literature, are used as basic building blocks of the circuit.

Once the circuit structure is specified, it is captured by the Cadence Composer schematic editor using a library of *symbol* views of basic RSFQ cells. The full circuit is simulated at the gate level using the Cadence Verilog-XL or Leapfrog VHDL logic simulators. These simulators make use of libraries of *behavioral* views of the RSFQ gates [3] developed according to the procedure described in the previous subsection. *Experiments have demonstrated that the logic (or gate-level) simulation is about two orders of magnitude faster than the circuit (or junction-level) simulation.*

Test sequences are chosen manually to verify the correct function of the circuit and to detect any violations of the timing constraints. In case of errors, the circuit structure is modified and the simulation is repeated. The process is terminated once the circuit exhibits correct logical functionality and no timing errors are detected.

Several obstacles, specific to RSFQ technology, must be overcome to make gate level synthesis of RSFQ circuits effective. The first obstacle is the small input and output impedance of the RSFQ gates. This small impedance makes

```
library ieee;
use ieee.std_logic_1164.all;
library std;
use std.textio.all;

entity confbuf is
      generic(t_separation     : time     := 10 ps;
                  delay              : time     := 15 ps);
          port(a, b: in std_logic;
                q: out std_logic := '0');
end confbuf;

architecture behavior of confbuf is

begin

process(a, b)
 -- time when the last clock pulse appeared
variable  last_in_time  : time := 0 ps;

begin
        if rising_edge(a) then

                if (now - last_in_time >= t_separation) then
                        q <= transport '1' after delay;
                else
                        q <= transport 'X' after delay;
                end if;
                q <= transport '0' after delay + 2 ps;

                last_in_time := now;

        end if;

        if rising_edge(b) then

                if (now - last_in_time >= t_separation) then
                        q <= transport '1' after delay;
                else
                        q <= transport 'X' after delay;
                end if;
                q <= transport '0' after delay + 2 ps;

                last_in_time := now;

        end if;

end process;

end behavior;
```

Fig. 11. VHDL behavioral model of a confluence buffer.

it difficult to connect two arbitrary gates without reoptimizing the device parameters at the interface between these gates. Two basic approaches can be used to circumvent this problem.

In the first approach, the device parameters at the input–output interfaces of the gate are chosen to yield zero input and output currents when the gate is connected to a standard JTL. This solution is, however, difficult to implement. RSFQ gates often have several (a minimum of two) internal states (see Fig. 4), which differ by the distribution of the internal currents. It is virtually impossible to choose the parameters of these devices in such a way as to yield zero input–output currents in all internal states of the gate. Additionally, choosing zero input–output currents may deteriorate the device margins in other parts of the circuit. A partial solution is to choose the device parameters so as to minimize the input–output currents in all possible states, although only when the effect on the critical margin of the circuit is negligible.

A second approach is simpler and more effective. The *core of the gate*, i.e., the part of the gate which is sufficient to perform the basic logic function, is extended before the optimization process with two JTL stages at each gate input and output. The optimization process changes the design parameters of the inner JTL's, leaving the outer JTL's basically unchanged. When two gates implemented with this procedure are connected, at least four JTL stages exist between the cores, providing excellent separation. Therefore, the critical margins of both of the gates remain essentially unaffected.

A drawback to this approach is that adding JTL's at the inputs and outputs of basic RSFQ gates adds delay, and the increased latency of the circuit is a disadvantage for certain applications such as contingent computation. It is important to note, however, that adding the same delay at the input and output of an RSFQ gate does *not* affect the other timing parameters such as the hold time, the setup time, and the minimum separation time. The *absolute minimum clock period* of a clocked RSFQ gate is the sum of the hold time, the setup time, and the minimum separation time [20] and does not depend on the gate delay. In certain timing schemes, such as *concurrent clocking* and *two-phase clocking* [21], the maximum clock frequency is determined (without taking parameter variations into account) only by the absolute minimum clock period of the slowest gate [21]. Adding input–output JTL's therefore does not necessarily reduce the maximum clock frequency of the entire circuit.

When the most simple and straightforward clocking scheme, *counterflow clocking* [21], is applied, the maximum clock frequency is dependent upon the gate delays. The minimum clock period is increased by the delay of two JTL stages (compared with the standard situation where only one JTL stage is used at each gate input and output). Based on current technology (minimum 3.5 μm junction size) the clock period will increase by 6 ps, compared to the minimum clock period of a large RSFQ circuit in the range of 50–100 ps. Therefore, the degradation in speed is in the range of 6–12%, a factor that only negligibly reduces the performance advantage of superconducting circuits over traditional semiconductor technologies.

For larger circuits, long-distance interconnects may consist of matching microstrip lines (MSL's), to date only used experimentally in one large circuit [22]. A more mature methodology will include a tradeoff analysis to determine in which circumstance the speed advantage of MSL's is more important than their greater area requirement.

A second problem with applying gate level synthesis to the design of RSFQ circuits is the influence of gate delay variations resulting from inaccuracies in the fabrication process on the operation of the circuit. This effect has often been overestimated in the past. Simulation results demonstrate that a 3σ standard deviation of the delay of a basic RSFQ gate, corresponding to the variations in Hypres fabrication process [14], is not larger than 20% [23]. Additionally, as shown in

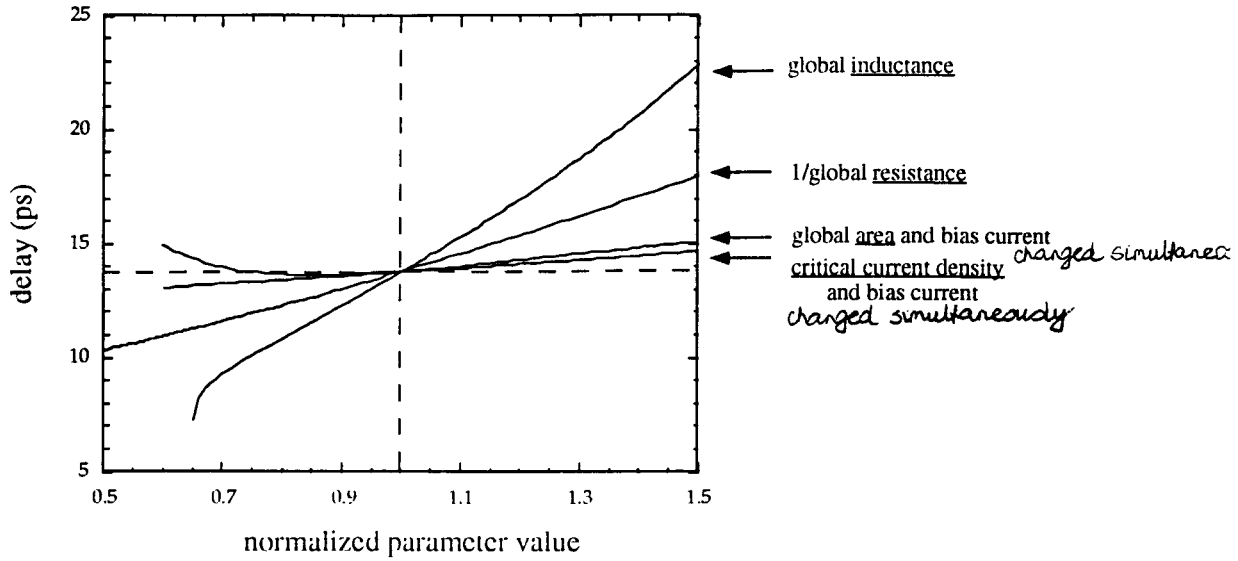


Fig. 12. Variations of the propagation delay as a function of the normalized global parameters for a DRO cell.

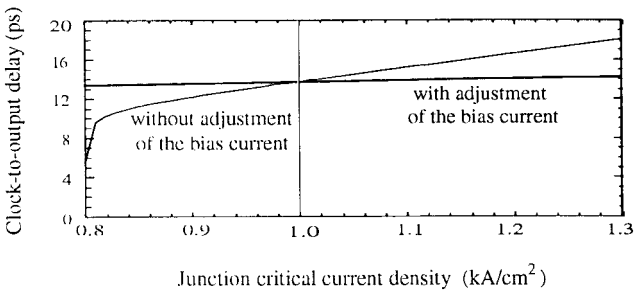


Fig. 13. Variations of the clock-to-output delay as a function of critical current density with and without a proportional adjustment of the global bias current for a DRO cell.

Fig. 12, in the region where the global parameters, such as the global inductance, global resistance, and global critical current, vary according to Hypres design rules, the delays are linearly dependent on these global parameters. Changing the bias current in proportion to the critical current density significantly reduces the maximum delay deviations, as shown in Fig. 13 [23]. To utilize this effect in the large-scale design methodology, before functional testing the bias current is set proportionally to the fabricated critical current density, which is measured using on-chip test structures. Setting the bias current proportionally to the actual value of the critical current density simplifies the experiments and makes timing analysis of the circuit at the gate level easier to perform.

A critical step in the synthesis of a multigigahertz digital circuit is the correct choice and implementation of the timing scheme [24]. The choice of synchronization strategy (synchronous versus asynchronous timing) is often determined by system level constraints. If this is not the case, the timing scheme is chosen after the general structure of the circuit has been developed, based on optimizing either circuit performance or area.

RSFQ logic permits the design of both synchronous and asynchronous circuits. Compared to semiconductor circuits, RSFQ synchronous circuits are more deeply pipelined and not well suited for zero skew clocking. Instead, several other clocking schemes with intentional nonzero clock skew, including counterflow, concurrent, and two-phase clocking, appear to be practical. Asynchronous RSFQ circuits have also been demonstrated, and several asynchronous timing schemes for RSFQ have been discussed in the literature [21]. Semiconductor circuits use synchronous timing almost exclusively [4]. Synchronous timing has also been successfully used in the majority of RSFQ circuits developed to date. In either synchronization approach, the timing of large-scale RSFQ circuits must consider relatively large parameter variations and multigigahertz signal frequencies. Certain features of various timing schemes for large-scale RSFQ circuits are summarized in Table I. The choice of timing scheme for a particular circuit should consider a variety of issues, such as the circuit speed, robustness to timing parameter variations, design simplicity, and area overhead.

C. Optimization

After the structure of an RSFQ circuit, including its synchronization circuitry, is determined, the interconnect delays must be carefully selected. Inappropriate delays in either the clock or data paths may cause the circuit to fail even if all the constituent gates operate correctly. Additionally, decreased yield, increased error rate, and/or a significantly reduced maximum operating clock frequency may occur. For these reasons, the design of the interconnect structures determines the ultimate speed and robustness of large-scale RSFQ circuits.

The procedure for optimizing the interconnect delays is very different depending upon the type of timing scheme chosen at the architectural or logic level. In particular, completely different optimization algorithms are used for asynchronous and synchronous timing schemes. The optimization procedure for several synchronous timing schemes developed at the University of Rochester are described below.

1) Counterflow Clocking: In the counterflow clocking scheme [21], the interconnect delay between clock inputs

TABLE I
FEATURES OF VARIOUS TIMING SCHEMES FOR LARGE RSFQ CIRCUITS. ALL CLOCK SCHEMES ARE DESCRIBED AT LENGTH IN [21] AND [24]. SOME ASSESSMENTS IN THE TABLE ARE CONTROVERSIAL AND DIFFERENT DESIGNERS MAY HAVE DIFFERENT OPINIONS

timing scheme	sync/ async	speed	robustness	design procedure simplicity	area overhead	suitable for linear Nx1 arrays	suitable for symmetric NxN arrays
counterflow	S	–	+	++	+	+	+/–
concurrent	S	+	–	– –	–	+	–
clock-follow-data	S	+	–	– –	– –	+	–
zero-skew	S	–/+	–	+	–/+	+	–
resynchronized	S	–	++	+	– –	+	+
two-phase	S	++	++	+	– –	+	+
dual-rail	A	+/–	+	+/–	– –	+	+
micropipelines	A	–	+	+/–	– –	+	+/–
delay insensitive	A	–/+	+	–/+	–	+	+
hybrid	A/S	– –	++	+	–	+	+

of two neighboring cells is determined on the basis of the minimum number of JTL stages necessary to cover the physical distance between these inputs. Special JTL's with extended inductor widths and lengths can be used to minimize the number of JTL stages between any two cells, thereby also minimizing the clock period.

The application of a counterflow scheme for medium-sized circuits almost guarantees that no low speed timing constraints (or race conditions) will occur. The majority of timing violations may appear only at high-speed, close to the maximum clock frequency. Exceptions to this rule include the following situations.

1) One of the cells in the circuit has a requirement on the minimum separation time between pulses at its two data inputs.
2) For a given pair of logically adjacent cells, the hold time of the second cell is exceptionally large (larger than the sum of the delay of the first cell and the delays of the clock and the data paths between the cells).
3) The clock distribution network contains two independent long branches leading to the clock inputs of two communicating cells. Variations in the delays of these two branches may cause a negative clock skew, which exceeds the intentional positive clock skew of the counterflow scheme [21].

The circuit can be checked for these conditions either manually or using the program CNET [3] developed at the University of Rochester. All of these potential sources of failure can be prevented by introducing additional delays into the data or clock paths within the circuit.

2) Concurrent and Clock-Follow-Data Clocking: Concurrent clocking and clock-follow-data clocking [21], [24], [25] are used when very high clock frequencies are required. The design and verification of the optimum clock distribution network with these schemes are much more complicated than for counterflow clocking.

The problem is to assure at the same time the maximum clock frequency and the maximum robustness of the circuit against race conditions. This process must consider both global and local parameter variations as described in [21]. This design effort requires the fabrication process to be statistically well characterized.

An analytical solution for these schemes which considers both global and local parameter variations has yet to be developed. An approximate solution for concurrent clocking which only considers global parameter variations is described in [25] and implemented in CNET [3].

To verify this solution, a Monte Carlo analysis integrated into a logic simulation is used. The timing parameters of each gate and interconnect are chosen randomly according to a gaussian distribution with the standard deviation determined on the basis of expected variations of the fabrication process. Results of this analysis are used to adjust delays within the most critical clock and data paths, and the analysis is repeated iteratively until a satisfactory circuit implementation is achieved.

3) Two-Phase Clocking: Two-phase clocking offers a higher maximum clock frequency than is possible with concurrent clocking [26]. Furthermore, the optimum delays in the clock distribution network are independent of variations in the fabrication process. A fully analytical solution to determine the optimum delays has been implemented in CNET. Some limitations on the type of the synchronous circuit which can use this scheme may apply. For example, shift registers are not effectively driven using two-phase clocking because double master–slave register stages need to be used instead of single register stages used in other clocking schemes.

D. Implementation

After circuit synthesis and optimization, a physical layout is constructed using the Cadence floorplanner and layout editor. The *layout* views of the basic cells are used for floorplanning. These views can be modified to adjust for any changes in the positions of the inputs and outputs in order to minimize the interconnect delays. If the interconnect delays obtained from the optimization process cannot be satisfied because of

TABLE II
FEATURES OF SELECTED RSFQ CELLS DEVELOPED AND MEASURED AT THE UNIVERSITY OF ROCHESTER: THE DESTRUCTIVE READ-OUT (DRO) CELL, THE AND GATE, TWO TYPES OF PARALLEL SHIFT REGISTERS (PSR1 AND PSR2), AND AN ADDER-ACCUMULATOR (AAC). NOTE THAT WHILE BOTH THE HOLD AND SETUP TIMES DEPEND UPON THE SOMEWHAT ARBITRARY POSITIONS AT WHICH THE CLOCK AND DATA ARE CONSIDERED TO ENTER THE CELL, THE SUM OF THESE MUST BE POSITIVE. THE MINIMUM CLOCK PERIOD LISTED IN THE TABLE IS SIMPLY THE SUM OF THE HOLD TIME AND THE SETUP TIME FOR THE CELL. IN REAL CIRCUITS OTHER EFFECTS SUCH AS PULSE REPULSION AND PARAMETER VARIATIONS [23] MAY DECREASE THE MAXIMUM OPERATING FREQUENCY

Name	No. of JJs		Axes of the Largest Hypersphere Inscribed within the Operating Region		Bias Current Margin		Timing Parameters [ps]			
	core	i/o	XIcb	XL	simu-lated	experi-mental	hold time	setup time	delay	Min. clock period
DRO	4	4	56%	41%	-41% +56%	-46% +67%	-3	8	9	5
AND	14	5	55%	40%	-55% +48%	-63% +31%	12	1	25	13
PSR1	14	6	53%	38%	-19% +39%	-30% +11%	-14	23	20	9
PSR2	16	4	53%	38%	-39% +34%	-27% +24%	14	-3	31	11
AAC	21	9	52%	38%	-22% +51%	-33% +25%	-43	48	13	20

physical constraints, the maximum speed of the circuit must be determined taking into account layout-derived values of interconnect delays. If this speed is unacceptable, the timing optimization process must be repeated.

Once the circuit is fully laid out, it is verified with DRC, parameter extraction, ERC, and LVS. This verification process is performed using the same tools and technology files as for a single cell; however, the circuit is verified hierarchically, such that if a single cell appears in the layout several times, its layout is verified only once. Note that the gate level schematic view of the circuit is used for both logic simulation, where each cell is represented by a behavioral model, and for layout-versus-schematic verification, where each cell is represented by a schematic view.

Another layout design goal is to minimize the total number of bias lines and pads. If the limitations on the maximum current are not exceeded, all bias currents are connected together or at least grouped into a small number of combined bias lines.

No tools for automated layout synthesis have been developed for RSFQ circuits. As a result, physical layout design is still the most cumbersome and time-consuming phase in the design of superconductive RSFQ circuits.

V. EXAMPLES OF APPLYING THIS DESIGN METHODOLOGY

A. Small Circuits

From 1994 to the present, about 20 basic RSFQ cells have been designed at the University of Rochester using the small circuit design methodology described in Section III. Selected features of these cells are summarized in Table II. Note the large yield (which is indicated by the large hypersphere axes) as well the large simulated and experimental bias current margins.

After these cells have been designed and manufactured, the timing parameters are extracted using TAN [3], and the behavioral models with timing information are described in Verilog HDL and VHDL [11]. These models are used in simulating and verifying larger circuits composed of these small basic cells.

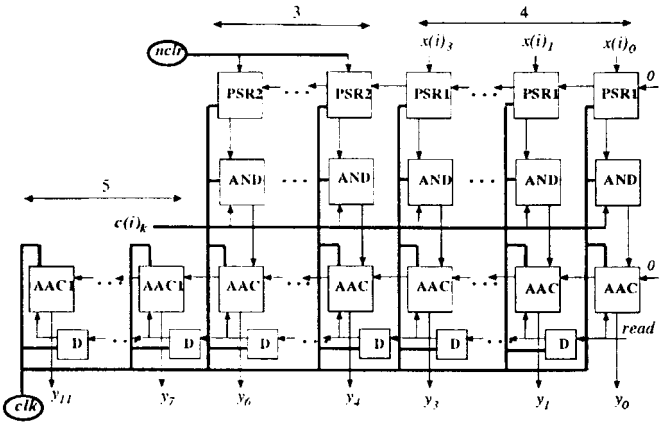


Fig. 14. Schematic diagram of a four-bit multiplier accumulator (MAC).

B. Medium-Scale Circuits

1) Multiplier-Accumulator Circuit (MAC): The methodology for developing large-scale RSFQ circuits has been demonstrated on the design of a four-bit multiplier accumulator, as described in [27]. The circuit contains approximately 1100 Josephson junctions, consisting of 38 synchronous RSFQ cells of six different types, as schematically shown in Fig. 14, and is one of the most complex RSFQ circuits verified experimentally to date. A microphotograph of this circuit is shown in Fig. 15. Detailed documentation describing the MAC is available on the web.[1]

The functional behavior of the multiplier-accumulator can be described using a simple formula, permitting an initial high-level diagram of the circuit to be created. The functional behavior of the circuit was first verified using URSULA [3], [17]. Design decisions were made regarding the specific function of

[1] http://www.ee.rochester.edu/users/sde/research/projects/rsfq/4bit.html.

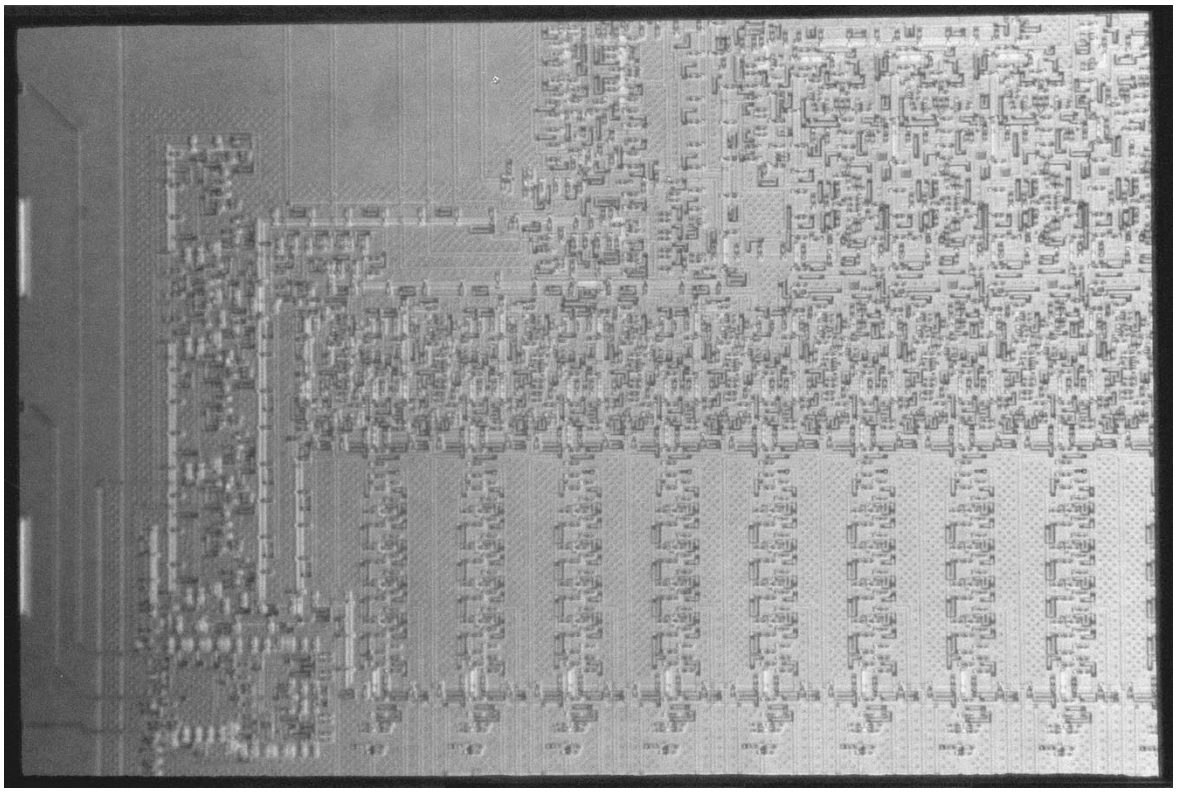


Fig. 15. Microphotograph of a four-bit multiplier accumulator (MAC) composed of 1100 Josephson junctions.

each of the cells constituting the MAC. These decisions were based on both the functional requirements of the entire circuit, as well as on the estimated difficulty of implementing each subfunctions in RSFQ. This approach resulted, for example, in replacing an initial design for an adder-accumulator cell described in [28], by an adder-accumulator based on a T1 flip-flop [27]. Similarly, a demultiplexer cell with small critical margin was replaced by much simpler parallel shift register cells of two types [27].

Most of the six basic cells comprising the MAC are composed of two or three elementary cells. These elementary cells are optimized individually using MALT. The core portion of each gate is supplemented with two JTL stages at each input and output. The parameters of these JTL's are varied during optimization. The elementary gates are connected into a larger cell by removing the outer JTL stages of each elementary cell and connecting the inner JTL's. The inner JTL's and the neighboring parts of the elementary cells are reoptimized using MALT. The necessary adjustments were typically quite small. The result of applying this procedure to an adder-accumulator, the most complex cell of the multiplier-accumulator, is shown in Fig. 16.

The next step in the design process is implementing the timing scheme. Counterflow clocking was chosen as the most simple and reliable solution. Concurrent clocking was considered and analyzed using logic simulation, but its physical implementation was abandoned for the following reasons:

- the design of this clocking scheme requires detailed and difficult to obtain knowledge of the variations in the fabrication process;

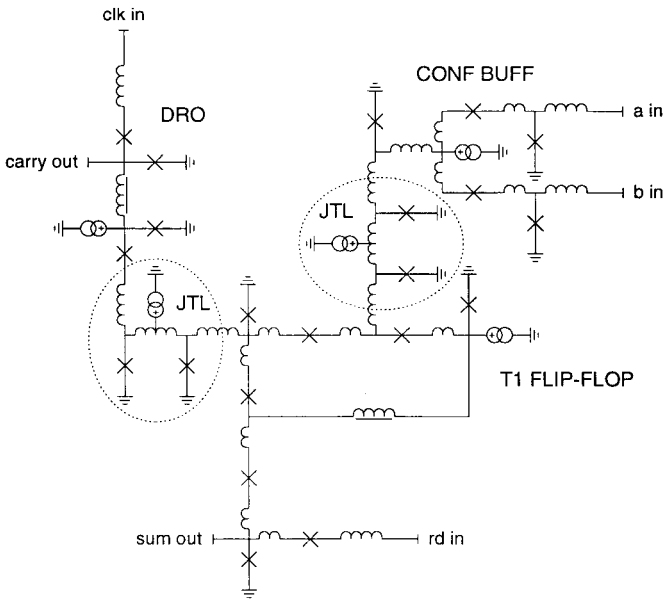


Fig. 16. Schematic of the adder-accumulator, the most complex cell within the multiplier-accumulator. Three elementary cells comprising the adder-accumulator (the DRO, confluence buffer, and T1 flip-flop) and the reoptimized interconnect JTL's between the cores of these elementary cells are noted on the schematic.

- extensive Monte Carlo analysis at the logic level would be required and this feature is not fully automated in the existing design environment.

The first version of the MAC with counterflow clocking was developed without prior timing analysis. Rather, all interconnections in the circuit were laid out using the minimum number

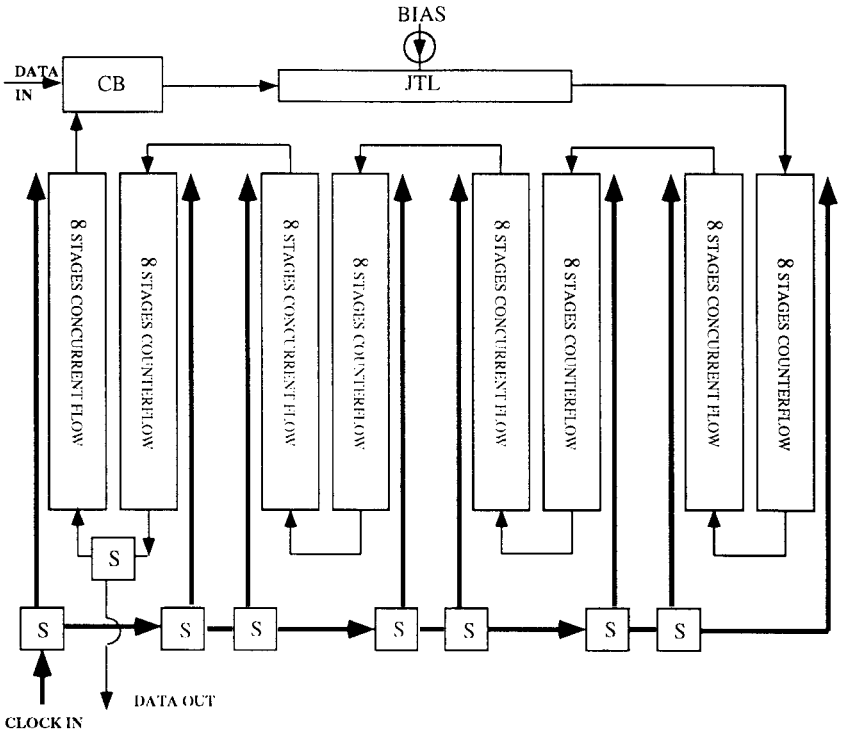


Fig. 17. Schematic diagram of a 64-bit circular shift register.

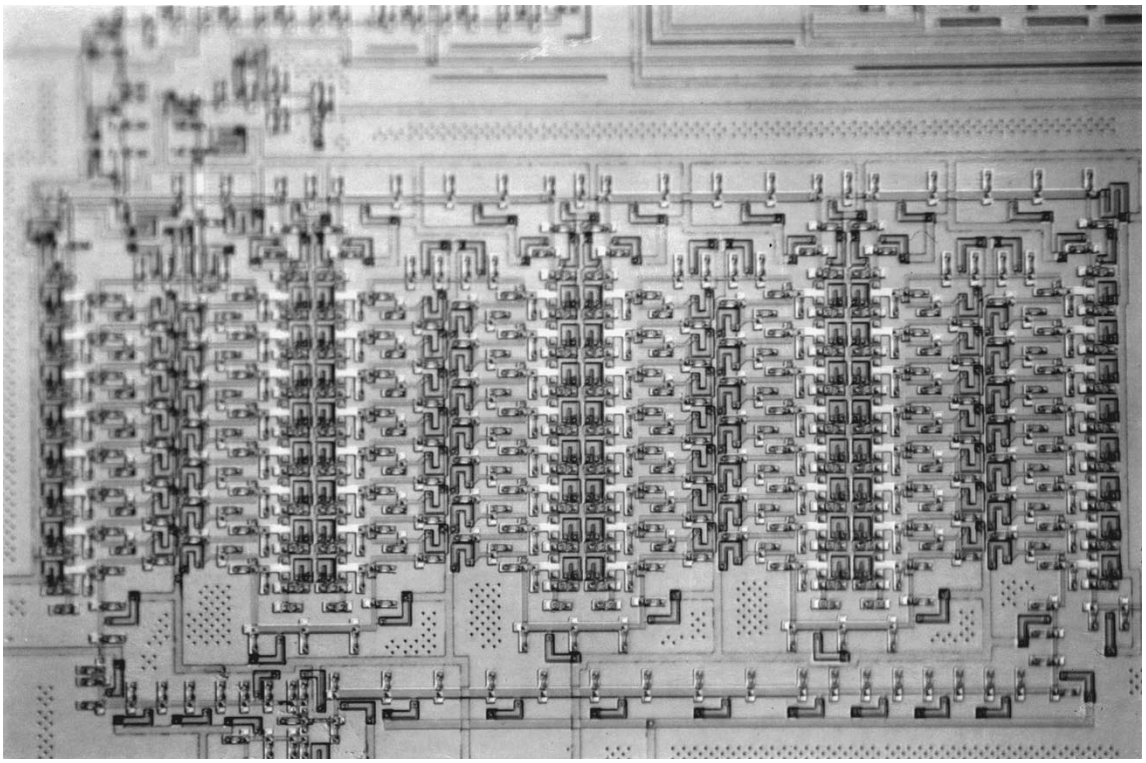


Fig. 18. Microphotograph of a 64-bit circular shift register (CSR-64), which has been experimentally demonstrated to operate correctly at a clock frequency up to 18 GHz.

of JTL stages necessary to physically cover the distance between the cells. The circuit was found to work for certain input sequences but produced errors for more complicated sequences which was attributed to a timing error.

Detailed timing analysis was performed using logic simulation with Verilog HDL models of RSFQ cells and Cadence Verilog-XL simulator [11]. The analysis revealed that the problem originated from a violation of the minimum separation

time between pulses at the two data inputs of the adder accumulator. This violation was independent of clock frequency and appeared only for specific test sequences. The error was easily fixed by introducing additional JTL stages at one of the adder accumulator inputs. The redesigned four-bit MAC was successfully tested at low speed with a global bias current margin of $\pm 5\%$.

Note that six basic cells were optimized separately. No additional reoptimization was necessary to connect these gates into a more complicated circuit structure. Furthermore, timing analysis at the logic level is crucial to properly determine the interconnect delays despite the simplicity of the clocking scheme.

2) CSR64: A second medium-scale RSFQ circuit developed according to this methodology is a 64-bit circular shift register. The details of the circuit are described in [29] together with the description and results of the high-speed testing. The circuit was confirmed to operate correctly up to a clock frequency of 18 GHz.

The circuit is composed of eight blocks, each containing eight shift register stages as shown in Figs. 17 and 18. Four blocks are clocked using counterflow clocking and the other four blocks using concurrent clocking.

Each individual stage of the shift register was designed according to the methodology used for small-scale circuits. The clock path of the register was treated as an integral part of the gate. The same circuit was used for both clocking schemes, with the direction of the clock path reversed. This strategy was possible because the timing constraints are more critical for concurrent clocking; therefore, the register stage optimized for this scheme would also operate correctly for the counterflow clocking scheme. The shift register stages were designed to be directly abutted to each other without any intermediate JTL stages.

The connections between the blocks were designed according to the methodology used for large-scale circuits. An appropriate number of JTL's was used to connect the blocks together along the data and the clock paths in such a way as to prevent timing errors at low speed, while maximizing the operational clock frequency. Logic simulation was applied at this stage to determine the proper number of JTL stages. The layout of the circuit was verified using LVS before fabrication.

VI. Summary

A comprehensive design methodology for the design of large-scale multigigahertz RSFQ circuits has been developed. The primary feature of this methodology is the application of different methods and tools for small and large-scale RSFQ circuits. The design of small-scale RSFQ circuits is centered around device parameter optimization and circuit simulation. The design of large-scale RSFQ circuits is focused on timing optimization and logic simulation.

This methodology for the design of large RSFQ circuits is based on the use of a semiconductor semicustom, standard-cell design methodology. Features specific to RSFQ logic include:

- a different set of basic gates constituting an RSFQ library;
- a large variety of RSFQ-based timing schemes;
- a complex timing optimization procedure;
- RSFQ-specific tools.

The detailed techniques presented in this paper are based on a semiconductor industry standard CAD commercial toolset, Cadence, which has been calibrated to operate with RSFQ logic. However, the methodology described in this paper is general and can be applied using a different set of commercial and public-domain CAD tools.

The methodology for small-scale circuits has been applied at the University of Rochester to the design of a library of over 20 elementary RSFQ gates. The methodology for large-scale circuits has been used to design two large-scale RSFQ circuits. Both of these circuits, a four-bit multiplier accumulator and a 64-bit circular shift register, are each one of the largest and most complex RSFQ circuits developed to date.

References

[1] 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.

[2] 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. Superconduct. Electron. Conf.*, Nagoya, Japan, Sept. 1995, pp. 27–30.

[3] K. Gaj *et al.*, "Tools for the computer-aided design of multi-gigahertz superconducting digital circuits," *IEEE Trans. Appl. Superconduct.*, vol. 9, pp. 18–38, Mar. 1999.

[4] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996, Ch. 11, pp. 629–692.

[5] S. Palnitkar, *Verilog HDL—A Guide to Digital Design and Synthesis*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[6] A. M. Dewey, *Analysis and Design of Digital Systems with VHDL*. PWS, 1997.

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

[8] V. K. Semenov, Y. A. Polyakov, and A. Ryzhikh, "Decimation filters based on RSFQ logic/memory cells," *Ext. Abs. ISEC'97*, 1997, pp. 344–346.

[9] P. Bunyk, A. Y. Kidiyarova-Shevchenko, and P. Litskevitch "RSFQ microprocessor: New design approaches," *IEEE Trans. Appl. Superconduct.*, vol. 7, pp. 2697–2704, June 1997.

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

[11] 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.

[12] 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.

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

[14] "Hypres niobium process flow and design rules," available from Hypres, Inc., 175 Clearbrook Road, Elmsford, NY 10523.

[15] 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.

[16] 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.

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

[18] U. Golze, *VLSI Chip Design with the Hardware Description Language VERILOG*. New York: Springer Verlag, 1996.

[19] D. Perry, *VHDL*, McGraw-Hill, 1991.

[20] K. Gaj, E. G. Friedman, and M. J. Feldman, "Analysis of timing requirements for basic RSFQ cells," presented at the *Appl. Superconduct. Conf. 1994*, Oct. 1994. Available at http://www.ee.rochester.edu/sde/research/publications/asc94.

[21] ———, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Processing*, vol. 9, pp. 247–276, 1997.

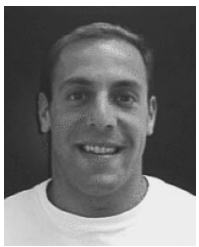
[22] V. K. Semenov, Y. A. Polyakov, and D. Schneider, "Implementation of oversampling analog-to-digital converter based on RSFQ logic," *Ext. Abs. ISEC'97*, pp. 41–43.
[23] K. Gaj, Q. P. Herr, and M. J. Feldman, "Parameter variations and synchronization of RSFQ circuits," in *Proc. Applied Superconductivity, Institute of Physics Conf.* Bristol, U.K., 1995, Series #148, pp. 1733–1736.
[24] Presentation "Timing of Large RSFQ Circuits." Available at http://www.ee.rochester.edu/users/sde/research/projects/rsfq/timing/title.html.
[25] 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.
[26] K. Gaj, E. G. Friedman, and M. J. Feldman, "Two-phase clocking for medium to large RSFQ circuits," *Ext. Abs. ISEC'97*, pp. 302–304.
[27] 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, June 1997.
[28] S. S. Martinet, D. K. Brock, M. J. Feldman, and M. F. Bocko, "Adder-accumulator cells in RSFQ logic," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3006–3009, 1995.
[29] 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.

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 (built-in self test) 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 a book on codebreaking.

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

His research interest at the University of Rochester, Rochester, NY, 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.



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 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. He was an IBM Watson Scholar and worked pre-professionally at IBM Microelectronics, Burlington, VT between 1988 and 1992 in the areas of final module test, packaging, circuit macro development, and standard cell design.

Darren K. Brock was born in Denton, TX, in 1968. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Rochester, Rochester, NY, in 1990, 1994, and 1997, respectively.

In 1997 he joined HYPRES, Inc. as a Member of Technical Staff, where he continued his graduate work on the design of rapid single flux quantum (RSFQ) electronics. He is currently Manager of Advanced Development at HYPRES, where his research focuses on CAD/CAE for digital and mixed-signal RSFQ circuits, including design methodologies, system-level architectures, design for testability, and cell retargeting.

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 ISCAS '96 and '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 both the B.A. and the M.A. degrees from the University of Pennsylvania, Philadelphia, in 1967 and the Ph.D. degree from the University of California, Berkeley in 1975, all in physics.

He worked at Chalmers University, Sweden, and at the NASA/Goddard Institute for Space Studies, New York City. He joined the Faculty of Electrical Engineering at the University of Virginia, Charlottesville, in 1985, where he established a laboratory to fabricate the highest quality Josephson junctions for receiver applications. He is now a Senior Scientist and Professor of Electrical and Computer Engineering at the University of Rochester. His current research interests include quantum-coherent computation, the development of ultra-high-speed digital circuits using superconducting single-flux-quantum logic, and the application of such circuits for useful purposes.