Low-Cost Superconducting Fan-Out With Cell I_C Ranking

Jennifer Volk, *Student Member, IEEE*, Georgios Tzimpragos, Alex Wynn, Evan Golden, and Timothy Sherwood, *Fellow, IEEE*

***Abstract*—Superconductor electronics (SCE) promise computer systems with orders of magnitude higher speeds and lower energy consumption than their complementary metal–oxide semiconductor (CMOS) counterparts. At the same time, the scalability and resource utilization of superconducting systems are major concerns. Some of these concerns come from device-level challenges and the gap between SCE and CMOS technology nodes, and others come from the way Josephson junctions (JJs) are used. Toward this end, we notice that a considerable fraction of hardware resources are not involved in logic operations, but rather are used for fan-out and buffering purposes. In this article, we ask if there is a way to reduce these overheads, propose the use of JJs at the cell boundaries to increase the number of outputs that a single stage can drive, and establish a set of rules to discretize critical currents in a way that is conducive to this assignment. Finally, we explore the design trade-offs that the presented approach opens up and demonstrate its promise through detailed analog simulations and modeling analyses. Our experiments indicate that the introduced method leads to a 48% savings in the JJ count for a tree with a fan-out of 1024, as well as an average of 43% of the JJ count for signal splitting and 32% for clock splitting in ISCAS'85 benchmarks.**

***Index Terms*—Circuit design, design methodology, digital superconductor electronics (SCE), superconductor circuit design.**

I. Introduction

THE performance and energy characteristics of superconductor electronics (SCE) put them in the spotlight as prime candidates for large-scale computing [1], quantum computing [2], and machine learning [3], [4], [5]. Their low computational density, however, is still a major roadblock ahead [6], [7]. That said, the computational density of SCE systems is predominantly a function of: the technology node, which governs the area of Josephson junctions (JJs) and transmission lines; the total number of JJs needed to implement the desired function; and the total number of required transmission lines [8]. In that regard, device researchers focus on the miniaturization of existing technology nodes [9], [10] while design automation experts concentrate on logic optimizations [11] and architects on joint logic and microarchitectural optimizations [12], [13], [14]. In this article, we adopt a different perspective on this problem by instead looking at reducing the number of JJs used for electrical purposes.

A close look at existing single flux quantum (SFQ)[1] benchmark designs reveals that 15%−33% of the total JJ count is taken up by splitters [18], [19]. Splitters act as amplifying Josephson transmission lines (JTLs) that deliver two copies of an input SFQ to its outputs and are essential for fan-out in SFQ circuit design [20]. On top of their hardware cost, multiple JJs per logic cell are typically used for buffering purposes, accounting for an additional ∼20% of the JJ count [21], [22].

The goal of this article is to reduce this overhead and make room for more logic cells and greater functionality within the same area. We observe that the JJs used for splitting and buffering perform related roles and can be "merged" by utilizing just one set of JJs to perform both tasks. In other words, we can integrate the capacity for fan-out—a term that is used in this work to refer to the number of outputs of a cell, irrespective of JJ sizes—into the logic cells themselves through a novel JJ-sharing technique. JJ sharing alludes to the process of replacing JJs in a splitter cell with JJs from surrounding cells. To achieve this integration, JJ sizes must be tuned on a case-by-case basis to match the desired critical currents (I_Cs) and meet fan-out needs—a process that is time-consuming at medium to large scales and can be prone to errors, especially under the complex interconnection properties of SFQ cells [6], [23]. We propose a new abstraction, dubbed cell I_C ranking,[2] to simplify the design of multiple-fan-out SFQ circuits through a fan-out-aware assignment of discrete baseline I_Cs to cells. In this context, a cell's baseline I_C is defined as the I_C of its buffer JJs and serves as a parametric scaling factor for all other JJs, inductors, and resistors in the cell [24]. We also derive a set of guidelines to systematically generate these assignments and provide a good starting point for further optimizations.

Manuscript received 15 June 2022; revised 22 October 2022, 19 January 2023, and 23 February 2023; accepted 27 February 2023. Date of publication 16 March 2023; date of current version 24 April 2023. This work was supported in part by the National Science Foundation under Grant 2006542 and Grant 1763699 and in part by the Under Secretary of Defense for Research and Engineering Air Force under Grant FA8702-15-D-0001. This paper was recommended by Associate Editor I. V. Vernik. *(Corresponding author: Jennifer Volk.)*

Jennifer Volk is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: jevolk@ucsb.edu).

Georgios Tzimpragos is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA.

Alex Wynn and Evan Golden are with the Massachusetts Institute of Technology Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02421 USA.

Timothy Sherwood is with the Department of Computer Science, University of California, Santa Barbara, CA 93106 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASC.2023.3256797.

Digital Object Identifier 10.1109/TASC.2023.3256797

[1]SFQ is used here to describe the Rapid SFQ (RSFQ) logic and its variants such as ERSFQ, eSFQ, etc., [15], [16], [17].

[2]For conciseness, we refer to cell I_C ranking as cell ranking from here on.

To validate this approach, we analyze how metrics such as area, energy, and delay are affected by the way cell ranks are assigned and demonstrate functional correctness, wide bias margins, and improved resource efficiency[3] for various example designs through a combination of detailed SPICE simulations[4] and higher-level models. Particularly, JJ sharing enables a fan-out of eight in a single stage with the use of JJs at the edges of cells—boundary JJs—for splitting with $\pm 37\%$ bias margins on average. The application of the introduced design rules leads to about 20% savings in the total JJ count for a simulated 2-bit Kogge–Stone adder (KSA) design. JJ sharing and cell ranking also give an estimated average savings of 10% in the total JJ count within ISCAS'85 benchmarks [26], [27]—43% of the JJ count for signal splitting and 32% for clock fan-out—even without taking into consideration the effects of path-balancing, and 47% for a tree with fan-out of 1024 (FO1024) compared to the use of conventional splitters.

Overall, the main contributions of this article are:

1) the use of boundary JJs for splitting purposes;
2) the abstraction of critical currents through cell ranking;
3) a set of rules that allows modular design at the gate level while enforcing circuit-level constraints; and
4) an analysis of cell rank as a new design trade-off parameter.

II. Reducing Electrical Redundancies

A. Splitting Problem

The poor scalability of active splitters, which are traditionally used to replicate SFQ pulses [20], has long been recognized as one of the key challenges in SCE [28], [29], [30], [31]. Moreover, high fan-out needs [32], especially in the case of conventional fully-synchronous SFQ designs [33], [34], [35], compound this problem. For example, in the FLUX processor, 77% of the instruction memory area goes to splitters, mergers, and transmission lines for address decoding [36]. A synthesized 8-bit general purpose RISC processor uses 21 065 splitters for clock distribution, accounting for nearly 16% of the total 400 000 JJ count—and this does not include splitting on the data path [31]. Moreover, a synthesized 4-bit KSA uses 58 clock splitters and 31 signal splitters, a cell count that exceeds the number of logic cells and path delay D-flip flops (DFFs) by 54% [37]. Obviously, if we could minimize these splitting overheads, we would open a significant amount of space for additional logic.

Before discussing splitting optimizations, let us take a look at the construction of a splitter to understand its functionality and constraints. Fig. 1 depicts the corresponding schematic. As can be seen, three JJs are needed to achieve an FO2. Each of the two legs of the splitter, colored in red and blue, acts as a JTL. The critical current of the input JJ is $\sqrt{2}$ times larger than the critical currents of the output JJs [20], the latter of which match the baseline critical currents of the preceding and succeeding SFQ cells. The difference in critical currents makes the entire splitter act as an amplifying JTL. In other words, the larger input JJ boosts incoming SFQ pulses so that they still meet the current requirements after the fan-out juncture to switch the smaller-sized output JJs.

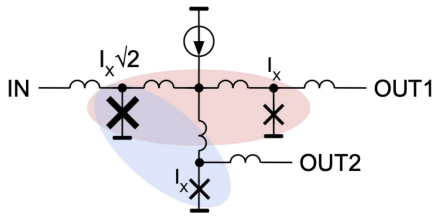

Fig. 1. An SFQ active splitter consists of three JJs and drives a fan-out of 2 (FO2). It does not perform any logic operation but rather acts as two JTLs, here red and blue, by directly passing SFQ from its input to both of its outputs. The sum of the critical currents of the two output JJs is $2I_X$, $\sqrt{2}$ times larger than that of the input, so the whole element is amplifying. Splitters can be chained directly, one after the other, with no degradation in bias margins.

Supporting a fan-out of N typically implies the use of a splitter tree that consists of $(N-1)$ FO2 splitters and at least $\log_2(N)$ stages. One way to improve this situation is by increasing the number of output ports per splitter cell [38]. Another improvement comes from a recently-proposed JJ sharing policy in which splitter output JJs are reused as input JJs for the next splitters [39]. In the case where splitters are followed by passive transmission lines (PTLs), further delay and area gains can be achieved by moving these PTLs from the outputs of the splitter cell to its center to replace the shared JJs [40]. These approaches are promising but require careful tuning of JJ critical currents, as well as the use of amplifying JTLs on their inputs and outputs for smooth integration with the rest of the circuit. However, replacing splitters with amplifying JTLs as a one-for-one substitution can hamper the gains made by the elimination of splitters. The goal of the proposed work is to instead extend the functionality of JJs within logic cells. We achieve this by applying JJ sharing to SFQ logic cells for the first time and introducing a systematic way to manage the shared JJs. In doing so, we reduce the cost of fan-out.

B. JJ Characterization

To enable JJ sharing, we look for redundancies in the logic cells. We start by examining the structure of conventional cells and categorizing JJs within them. In doing so, we pin down three common types: 1) those used in decision-making pairs [29], 2) blocking JJs, and 3) buffer JJs. An example OR cell [22] is shown in Fig. 2 with all JJs color-coded based on their functions. Decision-making pairs (magenta) act as comparators that consist of two JJs, in which one of the two JJs switches depending on the direction of bias current and incoming SFQ. Blocking JJs (gold) reject the passage of SFQ pulses in two ways. If they exist at the inputs, they prevent an extra SFQ from entering a superconducting quantum interference device (SQUID) while one is already circulating; e.g., in the case of the leftmost gold JJs in Fig. 2. If they exist at the output, they inhibit the backward

[3]For accurate area results, complete circuit layouts are necessary. However, considering that a greater number of JJs in a design also implies a greater number of inductors and resistors, JJ count is commonly used as a first-order estimate of resource efficiency [25].

[4]Our design netlists are available at https://github.com/UCSBarchlab/SFQ-Ranking.

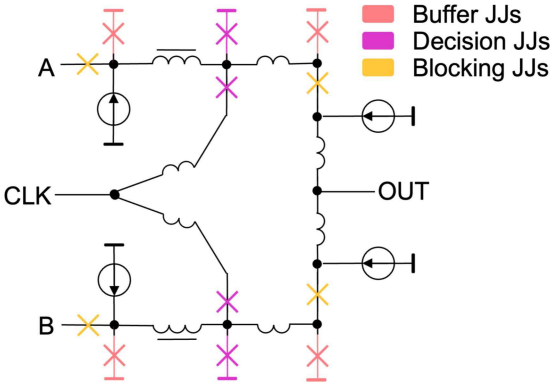

Fig. 2. An example SFQ OR cell, in which each JJ is color-coded to reflect its function in the cell. Pink JJs buffer input data and, in doing so, help maintain signal fidelity. Magenta JJs, indicating decision-making pairs, serve as comparators that evaluate the desired logical function. Finally, gold-blocking JJs are connected serially and perform two tasks: when they precede a SQUID in which an SFQ is currently stored, they prevent additional SFQ from entering the loop; and when they follow a SQUID or are close to the cell output, they prevent the backward propagation of SFQ from the output line toward the opposite input.

propagation of an SFQ toward the opposite input wire, as in the rightmost gold JJs. Finally, buffer JJs (pink) are inserted to maintain signal fidelity and typically serve to improve the electrical robustness of cells. In some cases, they also make up one side of a SQUID, with the decision-making JJ on the other side.

C. JJ Sharing

The ideal JJs for sharing sit on the edges of the cell and are wired in shunt configuration—as such, we call them "boundary JJs." In the examples that follow, both buffer JJs and decision-making pairs will be demonstrated for use in JJ sharing. Fig. 3 shows a typical circuit consisting of a DFF, two OR cells, and a splitter, with a total JJ count of 31. Fig. 4 shows the equivalent design after the proposed JJ sharing. As can be seen, the logic cells effectively merge with the splitter, thereby saving 2 JJs.

The above reassignment is made possible by the fact that the shared JJs have the same sizes before sharing. However, one 355 μA-JJ (in black) remains in Fig. 4, left over by the splitter. Convention states that because the baseline critical current of every cell in a library is static—for example, 250 μA [20], [21]—this JJ cannot be absorbed by the DFF that precedes it. Instead, we consider here a case where the baseline current can be flexibly assigned. For instance, if the DFFs decision-making pair I_C is scaled by 355 μA to come close to the I_C of the splitter's input JJ, then it can be shared for splitting. The result is shown in Fig. 5, in which all other JJs in the DFF are scaled by the same number.[5] By sharing from both sides, all of the JJs in the original splitter of Fig. 3 are absorbed to create a savings of three JJs, and splitting occurs directly at the output of the modified DFF. This modification's effect on bias margins depends on cell optimizations with different fan-out requirements and on the critical current gap between source and target cells. Cell-specific

[5]In practice, all cell inductors and resistors are inversely scaled by this value.

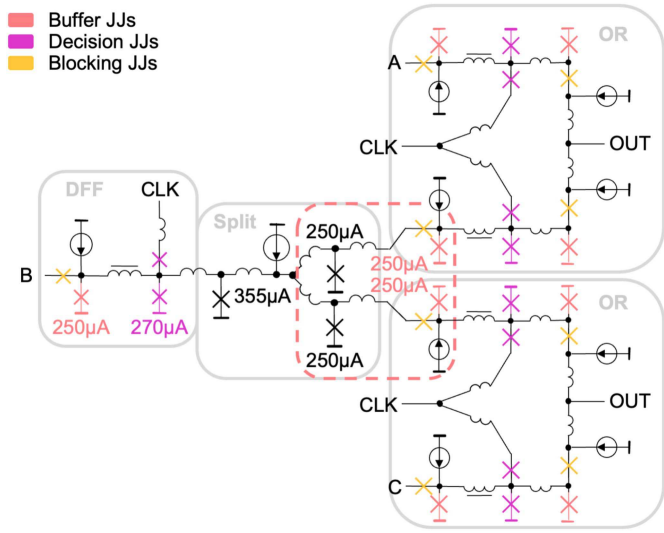

Fig. 3. A signal splits from a DFF to the inputs of two OR cells. Cells are outlined and labeled with their names. A conventional splitter enables an FO2 from the DFF. The JJs at the outputs of the splitter are redundant, as they have the same critical currents as those on the inputs of the two OR cells and perform a similar buffering function. Thus, they can be dropped and the OR cell's buffer JJs can be shared for splitting.

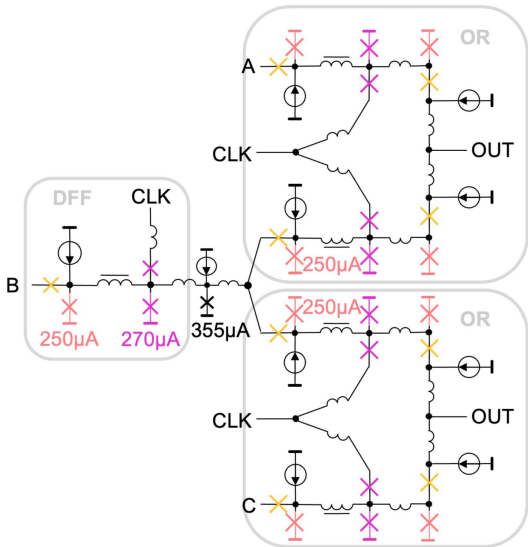

Fig. 4. The consequences of sharing the two OR cells' input JJs identified in Fig. 3. The splitter's output JJs are "merged" with the OR cell's buffer JJs so that only one of these sets is used while the splitting task is still accomplished.

optimizations are beyond the scope of this article, but basic design rules that guide more favorable connections are organized and presented in the next section.

III. I_C Abstraction Through Ranking

Even in small fan-out cases, critical current choices are important to ensure reliable connectivity. For example, an FO2 connection requires all of the JJs in the driving cell to be scaled by $\sqrt{2}$ [20]. Applying this approach to cells with larger fan-outs and baseline critical currents that are potentially different from

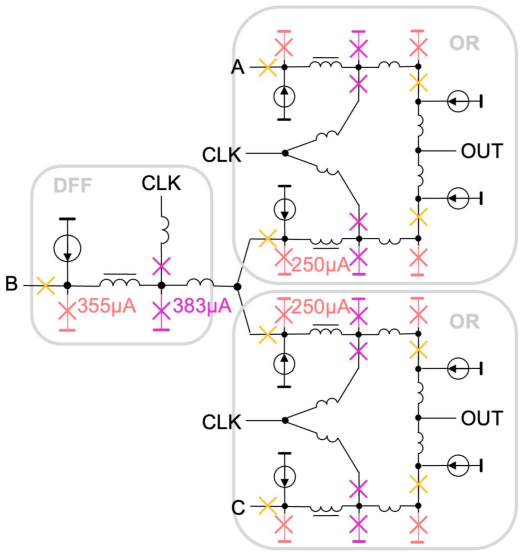

Fig. 5. The 355 μA-JJ in Fig. 4, left over from the splitter, is absorbed by the driving DFF through additional JJ sharing. As a prerequisite, the DFFs components must be scaled so that the baseline critical current matches that of the absorbed JJ. The JJ that sets the baseline critical current is the pink buffer JJ on the DFF input.

their neighbors,' however, turns every fan-out point into a unique calculation, which complicates the design process.

To simplify the procedure for connecting these cells, we create an abstraction in which critical currents are discrete. We call this abstraction ranking, wherein we select specific current values that have consistent separation within a range. Through ranking, we constrain the design options in a way that enforces the current ratio requirements of a connection. In this section, we define the relationship between the critical current of the driving boundary JJ, the critical current(s) of the target boundary JJ(s), and the maximum fan-out. We then detail the cost of various connections in terms of JJ count and finally compare and contrast three different rank-based design methodologies.

A. Ranking

A cell's baseline critical current defines its maximum fan-out capacity, and therefore, it is considered the primary identifying electrical characteristic. To relate critical current, which is analog, to fan-out capacity, which is discrete, we define the translation from one domain to the other.

Nominal values are chosen to start from 250 μA, as it is a popular baseline critical current for SFQ libraries [20], [21] and used in the prior example in Figs. 3 and 4. To discretize the range of critical currents, we take this as a central point and explore steps below and above it by multiplying by $(1/\sqrt{2})$ and $\sqrt{2}$, respectively. Going down, we reach 180, 125, 88, and 66 μA in turn, and finally, stop at 46 μA. Below this point, JJs become more prone to thermal errors [6], [41]. Applying the same approach to the other side, 353 μA is found after one step and 500 μA after two, which we choose here as our upper bound. In practice, however, the maximum JJ size is limited by the Josephson penetration depth [42] and practical shunt resistor size. In addition, it should be noted that the proposed JJ sharing and cell ranking techniques are independent of any particular selection of critical currents, range, and step size. From here on, we abstract the eight above-mentioned cell baseline critical currents with number labels, where the lowest baseline is assigned a label of 1 and the highest a label of 8.

In general, the number of ranks between the minimum and maximum critical currents with a rank step size p_r is

$$N_r \geq \frac{\log\left(\frac{I_{C,\mathrm{Max}}}{I_{C,\mathrm{Min}}}\right)}{\log(p_r)} + 1 \tag{1}$$

while the number of JTLs needed to amplify from a source critical current $\mathrm{I_S}$ to a target current $\mathrm{I_T}$ with an amplification step size p_a is

$$N_{\mathrm{JTL}} \geq \frac{\log\left(\frac{I_T}{I_S}\right)}{2\log(p_a)}. \tag{2}$$

The above inequalities stem from the fact that $N_r, N_{\mathrm{JTL}} \in \mathbb{N}$ and the observation that not every rank's nominal critical current value will necessarily match those chosen in an existing cell design. To add robustness and maximize interoperability between different library cells that may not match perfectly, critical current intervals can be assigned; however, such an assignment is beyond the scope of this article.

Fig. 6 shows a lookup table that describes the connectivity between various ranks for the range discussed above and $p_r = p_a \approx \sqrt{2}$. A cell with a higher rank, or larger baseline critical current, can be directly connected to any other with the same or lower rank. Consider an example in which four target cells' ranks must be found for a design that has an FO4 from a rank-6 source cell with a 250 μA baseline critical current. It can be concluded from the crosspoint of a rank-6 source cell and FO4 cell value in Fig. 6 that the target cells must be of rank-3.

Conversely, connecting a smaller-ranked cell to one with a higher rank necessitates the use of amplifying JTLs, the number of which is also dependent on their rank difference. Assume that it is now desirable to move one of those rank-3 cells back up to rank-6, likely in advance of a larger anticipated fan-out. In this case, the source is a rank-3 cell and the target is rank-6. The crosspoint of the rank-3 row and the rank-6 column in Fig. 6 indicate that two amplifying JTLs are needed for the connection. The first of the two JTLs will amplify from rank-3 to rank-4 and the second from rank-5 to rank-6. Note that an additional step in rank is gained (rank-4 to rank-5) in-between JTLs without introducing another amplifying JTL.

B. Design Methods

The rules discussed above can be carried out in more than one way depending on the designer's target goals. In this section, we use a DFF that fans out to four more DFFs as a running example to demonstrate three such design methods. In Section V, we elaborate on them and show how one can apply ranking to optimize for design reuse, energy, speed, or JJ count.

Target

Source Rank	Source I_C	Target Rank 8	7	6	5	4	3	2	1
		500μA	353μA	250μA	180μA	125μA	88μA	66μA	46μA
8	500μA	FO1	FO2	FO3	FO4	FO5	FO6	FO7	FO8
7	353μA	+1 JTL	FO1	FO2	FO3	FO4	FO5	FO6	FO7
6	250μA	+2 JTL	+1 JTL	FO1	FO2	FO3	FO4	FO5	FO6
5	180μA	+2 JTL	+2 JTL	+1 JTL	FO1	FO2	FO3	FO4	FO5
4	125μA	+3 JTL	+2 JTL	+2 JTL	+1 JTL	FO1	FO2	FO3	FO4
3	88μA	+3 JTL	+3 JTL	+2 JTL	+2 JTL	+1 JTL	FO1	FO2	FO3
2	66μA	+4 JTL	+3 JTL	+3 JTL	+2 JTL	+2 JTL	+1 JTL	FO1	FO2
1	46μA	+4 JTL	+4 JTL	+3 JTL	+3 JTL	+2 JTL	+2 JTL	+1 JTL	FO1

Amplification

Fig. 6. Rounded baseline critical current values for cells are abstracted into a property called "rank." This table depicts a rule set that dictates the connection type of a source cell (row) to the target cell(s) (column). The FO1 diagonal between green and gray/white cells marks a transitional line—to the right, the connection is *fan-out positive* and needs no additional hardware to achieve a fan-out up to the amount shown; to the left, the connection is fan-out negative: Each cell lists the number of amplifying JTLs needed to bridge the gap in rank. The hardware overhead required to amplify from a lower rank to a higher one increases sublinearly with the distance between ranks, as one additional step in rank can be gained *between* amplifying JTLs. Each JTL comprises two JJs.

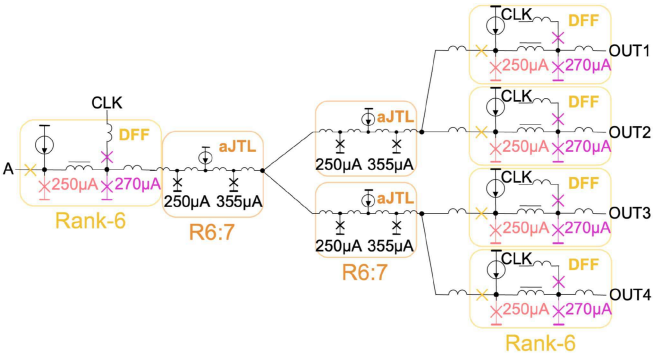

Fig. 7. Ranking affords three primary design methodologies, each of which is demonstrated with an example. A single-DFF source drives four DFF targets. Cells are outlined and labeled with their names and ranks. Amplifying JTLs modify the rank from input to output where needed and are labeled as R s:t, where s and t are the source and target ranks, respectively. The nominal rank is chosen to match the baseline used in the original library; here, it is 250 μA. Fan-out is achieved by using amplifying JTLs—in this case, they are arranged in a splitter-like tree structure.

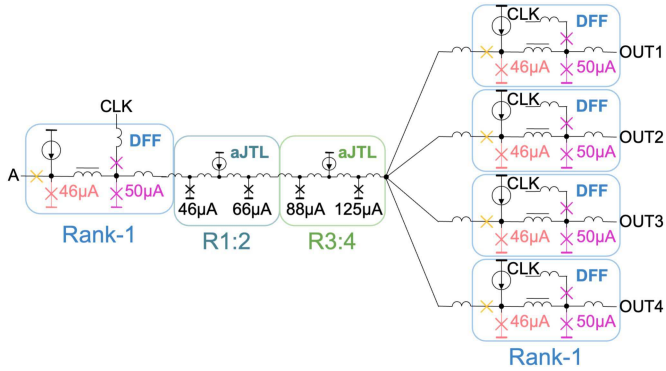

Fig. 8. The second rank-based design methodology. The baseline rank is 1 and is amplified only in advance of fan-out.

The first method requires the rank to remain the same for all logic cells. This enables the use of logic cells from existing libraries [20], [21] without modifications. An example is provided in Fig. 7, where cells of rank-6 make up the baseline. To achieve an FO4, three amplifying JTLs can be chained together in a tree structure, much like conventional splitters. Each JTL amplifies by one rank and fans out to two lines, saving three JJs compared to the conventional case.

The second method keeps the logic cells at the lowest rank, which provides faster JJ switching speeds and lower power consumption [6], [15]. Similar to the first case, the logic cell library needs no modifications after its initial design because the rank is consistent—only one copy of each cell is needed. However, because the starting rank is the lowest, it is likely to save more JJs at higher fan-outs compared to the first case. Fig. 8 depicts the resulting schematic. In this case, a two-stage JTL chain is needed before fan-out. The JJ count is five fewer than that using the splitter method.

The third method aims to reduce the number of amplifying JTLs used in the previous methods. Conventionally, superconductor cell libraries select a single baseline critical current to be used globally for all cells. However, adopting a more flexible assignment allows the fan-out point to shift closer to the output of the source logic cell. Applying this to the running example results in the circuit shown in Fig. 9, which begins with a rank-4 DFF that immediately splits to rank-1 DFFs and thus, in this case, removes all hardware overhead for splitting.

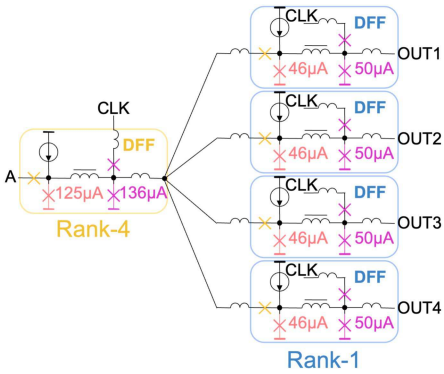

Fig. 9. The third rank-based design methodology is a flexible ranking scheme, in which the source cell and target cell(s) can take on any rank as long as the fan-out rules in Fig. 6 are upheld.

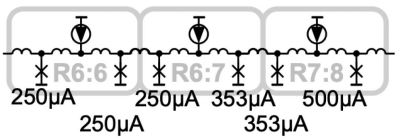

Fig. 10. An amplifying JTL chain—the first JTL stage serves as a non-amplifying buffer between test inputs and the circuit under test. Bias margins are simulated and found to be $+38.5\%/-50\%$.

IV. Evaluation

A. Evaluating the Basic Building Blocks

1) Bias Margin Analysis: The JTL chain is a fundamental building block of SFQ design and critical for the proposed approach. For this reason, various amplification chain configurations are simulated extensively, similarly to splitter chains, in the following experiments and their bias margins are compared. All designs are simulated in the Cadence design suite using models based on the MIT Lincoln Laboratory SFQ5ee 10 kA/cm^2 process and are terminated with strings of non-amplifying JTLs that match the designs' output impedances. Each JTL element consists of two externally-shunted JJs, four inductors, and a single resistor that biases both JJs. Each JTL chain that amplifies the rank from Rs to Rt is labeled with Rs:t. For a JTL that is not amplifying, $s = t$.

Our starting case is a JTL chain without amplification, designed with rank-6 JTLs. Bias margin simulations indicate an operating range of $+38.5\%/-65.4\%$. A chain that amplifies from a rank of six to a rank of eight with a step size of $\sqrt{2}$ [20], [43] is depicted in Fig. 10 and also has margins of $+38.5\%/-50\%$. The first JTL stage acts as a non-amplifying buffer between test inputs and the circuit under test. These two results serve as a reference for the ensuing tests.

To test fan-out directly from the JTLs, the above chain is used to split to three chains of rank-6 JTLs. This configuration is shown in Fig. 11 and has bias margins of $+38.5\%/-38.5\%$. By comparison, a splitter tree with the same fan-out, composed of directly-chained splitters that match Fig. 1 and have a current I_X of 250 μA, has bias margins of $+26.9\%/-30.8\%$.

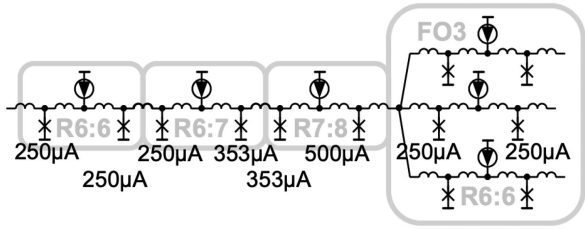

Fig. 11. An amplifying JTL chain leading to FO3—the first JTL stage serves as a non-amplifying buffer between test inputs and the circuit under test. Bias margins are simulated and found to be $+38.5\%/-38.5\%$.

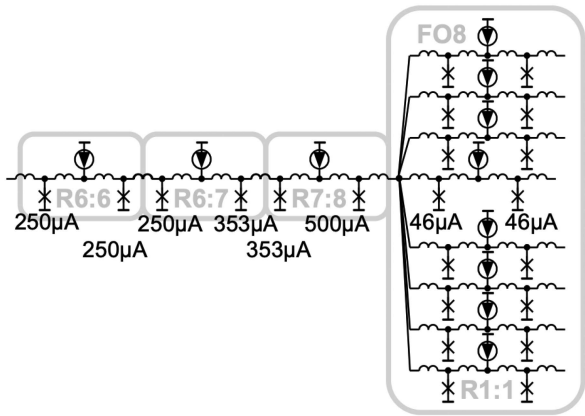

Fig. 12. An amplifying JTL chain leading to FO8—the first JTL stage serves as a non-amplifying buffer between test inputs and the circuit under test. Bias margins are simulated and found to be $+38.5\%/-53.8\%$.

To test the effects of additional fan-out, five more JTL chains are added onto the fan-out node and all of the output chains' baselines are reduced to rank-1 to achieve an FO8 in total. The schematic is shown in Fig. 12. The bias margins, in this case, are $+38.5\%/-53.8\%$. By comparison, a splitter tree with an FO8, composed of the same splitters described above, has bias margins of $+26.9\%/-26.9\%$.

Next, amplification is added from rank-1 to rank-8 to lead into the FO8, using a step size of $\sqrt{2}$ within the JTLs and between JTLs, yielding bias margins of $+42.3\%/-23.1\%$. This serves as a simulation of the longest possible amplification chain with the given rank range.

The above experiments show that bias margins are, for the most part, maintained after each progression, starting from the FO3 in Fig. 11. For the sake of modularizing the design, it is important to know whether it is possible to ultimately achieve the same fan-out with different cells while preserving those cells' respective bias margins. We test FO9 using the same JTL chains that amplify from rank-6 to rank-8, as shown in Fig. 13, and find that bias margins are $+38.5\%/-30.8\%$, which are similar to those of the FO3 case above and reveal that the design method indeed promotes modularity.

Bias margin simulation results for the above examples are summarized in Fig. 14.

2) Current Savings Analysis: According to the above-presented analyses, the proposed rank-based design methodology shows promise for better resource utilization without

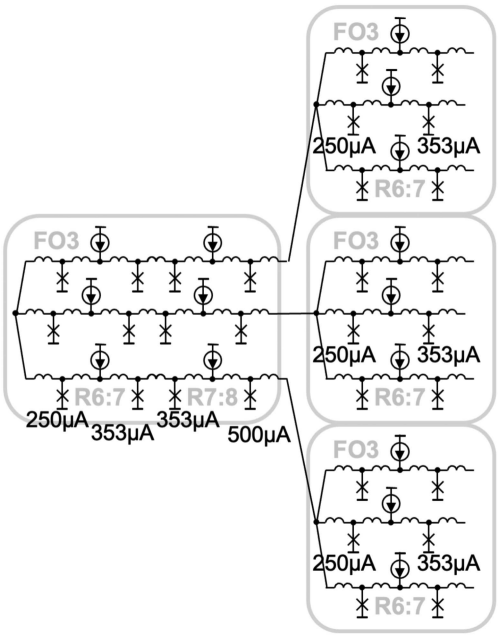

Fig. 13. An amplifying JTL chain leading to FO9, using R6:8 amplifying segments. Bias margins are simulated and found to be $+38.5\%/-30.8\%$.

Configuration	Ranking	Splitters
R6:8 (Fig. 10)	$+38.5\%/-50\%$	-
R6:8+FO3 (Fig. 11)	$+38.5\%/-38.5\%$	$+26.9\%/-30.8\%$
R6:8+FO8 (Fig. 12)	$+38.5\%/-53.8\%$	$+26.9\%/-26.9\%$
R1:8+FO8	$+42.3\%/-23.1\%$	$+26.9\%/-26.9\%$
R6:8+FO9 (Fig. 13)	$+38.5\%/-30.8\%$	$+30.8\%/-26.9\%$

Fig. 14. A summary of bias margins and comparison of various fan-out configurations between conventional splitters and ranking. The fourth entry refers to the configuration in which an amplifying JTL chain from rank-1 to rank-8 precedes a fan-out to 8.

bias margin degradation. However, it is not clear that the bias current is conserved, as different ranks impose different bias current requirements. To shed light on this, we calculate the bias current overhead for fanning out with an FO2, FO4, and FO8 from various logic cells and compare the ranking methodology to conventional splitting in terms of bias current savings. Our results are shown in Fig. 15. In this study, we consider two cases: in the first, we use flexible ranking, in which the source cell is of rank-6 and the target cells assume any rank that uses the smallest bias current overhead. In the second option, the source and target cell ranks match.

B. Building Rank-Based Circuits

Next, a 2-bit KSA is used as a comprehensible example to demonstrate how ranking can be used to simplify the design of circuits with flexible baseline I_Cs. The block diagram of the adder design is shown in Fig. 16. Synchronous AND, OR, XOR, and DFF cells are used for its implementation. Considering that the logic function of each cell is not relevant to ranking, however, cells are depicted as rectangles only labeled with their ranks Rs.

Cell	Fan-Out	Flex. Rank	Matched Rank
AND	*FO2*	59.2%	17.4%
	FO4	81.3%	23.9%
	FO8	68.9%	26.8%
OR	*FO2*	49.3%	14.5%
	FO4	74.4%	21.9%
	FO8	66.0%	25.6%
XOR	*FO2*	77.6%	22.8%
	FO4	91.2%	26.8%
	FO8	72.7%	28.3%
INV	*FO2*	41.8%	12.3%
	FO4	68.3%	20.1%
	FO8	63.1%	24.5%
DFF	*FO2*	77.6%	22.8%
	FO4	91.2%	26.8%
	FO8	72.7%	28.3%
AVG		70.4%	22.9%

Fig. 15. The percent savings in total bias current of various basic cell configurations with ranking versus conventional splitting. Two ranking options are considered: flexible ranking, in which the source cell is of rank-6 and the target cells take on any desired rank; and matched ranking, in which both the source and target cells are of rank-6.

As before, JTL chains that amplify the rank from Rs to Rt, necessary to meet the ranking rules from Fig. 6, are labeled as Rs:t. The required fan-out at the output of each cell is also labeled.

Assigning custom ranks to the fully-synchronous KSA design takes just five steps.

1) **Step ❶**: Find the stage with the most synchronous elements and assign the highest or lowest rank needed to minimize the number of additional JTLs on the clock line. Ascribe the highest rank if additional fan-out is needed after this stage, or if lower variability is needed [6]; ascribe the lowest rank if greater energy-efficiency or speed is desired. In the design of Fig. 16, the third stage has 6 synchronous components—more than any other stage. FO7 is possible with a clock that splits from rank-8 to rank-2 cells, or from rank-7 to rank-1 cells. We opt for the first case and assign rank-2 (R2) to all cells in this stage.
2) **Step ❷**: For any two cells that share a direct connection, FO1, assign the same rank to both. That is to say, chained FO1 connections will propagate the same rank value. In Fig. 16, we start at the stage found in ❶ and propagate the values over all FO1 connections to the left and right sides. As a consequence, all cells in the fourth and fifth stages in Fig. 16 are assigned rank-2. Additionally, the top cell in the second stage and the bottom cells in the first and second stages are assigned rank-2. Other cells that share an FO1 connection are noted to have the same ranks as their connected neighbors, which are not yet known.
3) **Step ❸**: Keep target cells the same rank if they share a source cell. In this case, the penultimate cell in the first stage is forced to be rank-2, the second cell in the second stage is forced to be rank-2, and the two middle cells in the first stage are forced to be the same rank, which is still unknown. The fourth cell in the second stage is

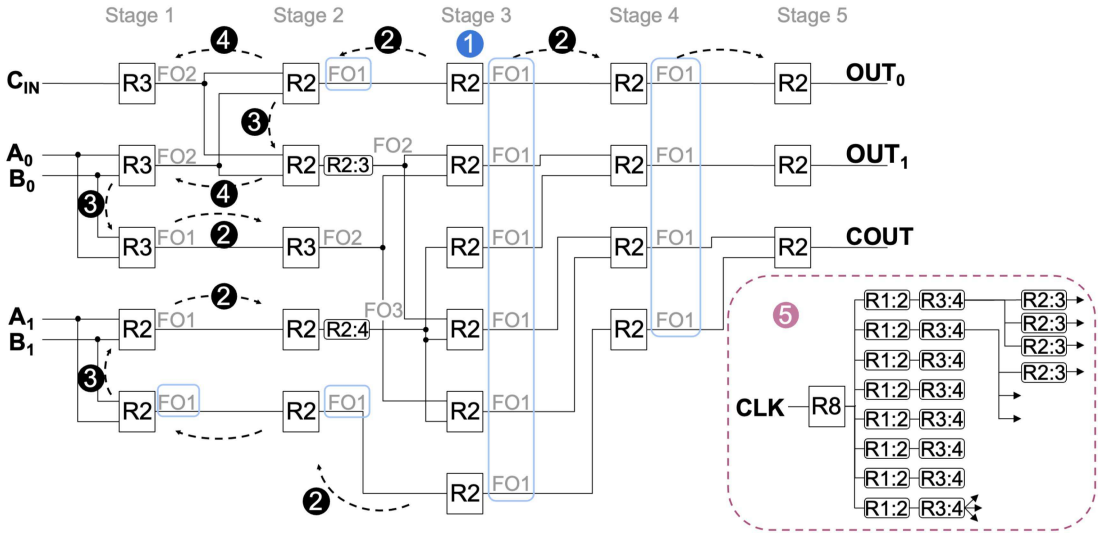

Fig. 16. Block diagram of a fully-synchronous 2-bit KSA designed and simulated in Cadence using the MITLL SFQ5ee 100 μA/μm^2 fabrication process models. Ranks are assigned to the design and clock tree using a five-step methodology. The proposed JJ sharing and flexible I_C assignment save 17.7% of JJs compared to the conventional splitting method.

also assigned rank-2, based on the sharing that occurred in Step ❷. The cells that have yet to be ranked at this point are the top three cells in the first stage and the third cell in the second stage.

4) **Step ❹**: To define the ranks of the remaining cells, we rely again on the table in Fig. 6 while considering the ranks and fan-outs of the cells that surround the ones in question. Amplifying JTLs are inserted in this step to electrically reinforce the connections between cells. This fills in the remaining unranked cells in this example: The top three cells in the first stage are assigned rank-3. This rank then propagates to the third cell in the second stage. Finally, amplifying JTLs are inserted after the second and fourth cells in the second stage to meet the FO2 and FO3 requirements of the connection between rank-2 source cells and rank-2 target cells. It is also now clear that inputs A_0 and B_0 should be sourced from rank-4 cells, and A_1 and B_1 should be sourced from rank-3 cells.
5) **Step ❺**: The last step is to design the clock tree using chains of amplifying JTLs that meet the fan-out and target cell rank requirements for every synchronous stage. In the shown example, we start with a rank-8 cell that fans out to 8, and then amplify each line to rank-4. Six of these rank-4 lines fan out to three rank-2 cells each, which covers the rank-2 cells in every stage from the second to the fifth while two are reserved for sharing amongst the remaining rank-2 and rank-3 cells in the first and second stages. To amplify from rank-2 to rank-3, one JTL is used per line, following the guidelines provided in Fig. 6.

To verify the functional correctness of the resulting design, analog simulations in the Cadence design suite are performed. The cells used are based on publicly available designs from Stonybrook [20], [22], [44], [45]. To satisfy the requirement

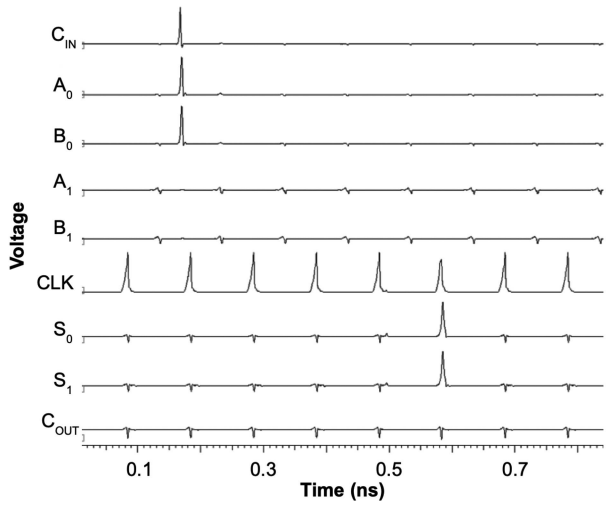

Fig. 17. Simulation waveform for the 2-bit KSA with ranking. Inputs are 01, 01, and 1 for A, B, and C_{IN}, respectively. The correct result, $C_{\text{OUT}} = 0, S_1 = 1$, and $S_0 = 1$, appears five clock cycles later.

for cells with flexible ranks, Stonybrook's designs are scaled to the required rank and individually tested before being stitched together. JTLs are inserted to enforce rank and prevent timing violations. An example waveform demonstrating the adder's functional behavior is shown in Fig. 17. The total JJ count for the design is 317 and includes logic cells, DC-to-SFQ converters, and JTLs. The bias margins are simulated and found to be $+19.2\%/-3.8\%$.

To quantify the gains compared with a traditional approach, we reimplement the same 2-bit KSA design, but this time without the proposed JJ sharing and cell ranking techniques. The achieved results indicate that a total of 385 JJs are needed for the traditional implementation, including 114 JJs for splitting. In other words, the proposed methodology leads to 17.7% JJ

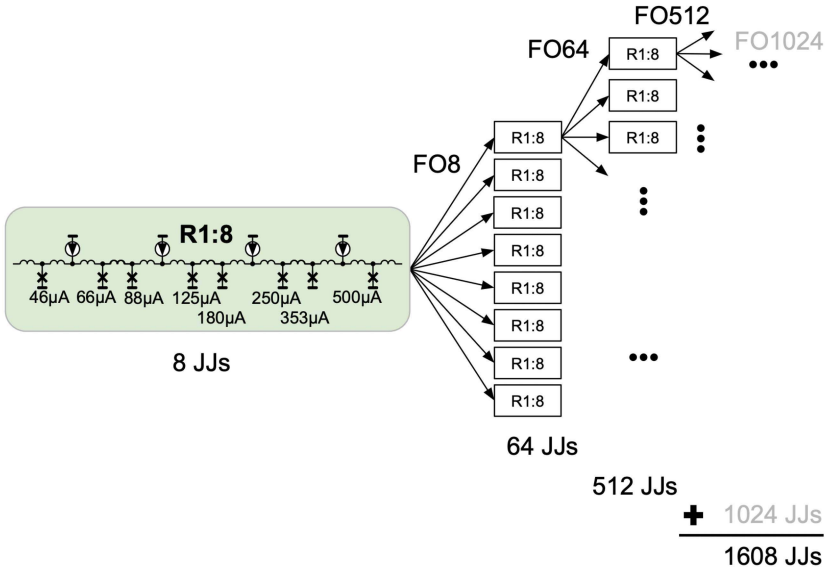

Fig. 18. An FO1024 tree using ranking. Chains of JTLs followed by an FO8 serve as the modular building block and are depicted here as rectangles with the label R1:8. Each building block amplifies from rank-1 to rank-8, as shown in the first chain, and costs eight JJs. This balanced tree costs 1608 JJs, whereas an FO1024 tree using conventional splitters costs 3069 JJs—a savings of 47.6%.

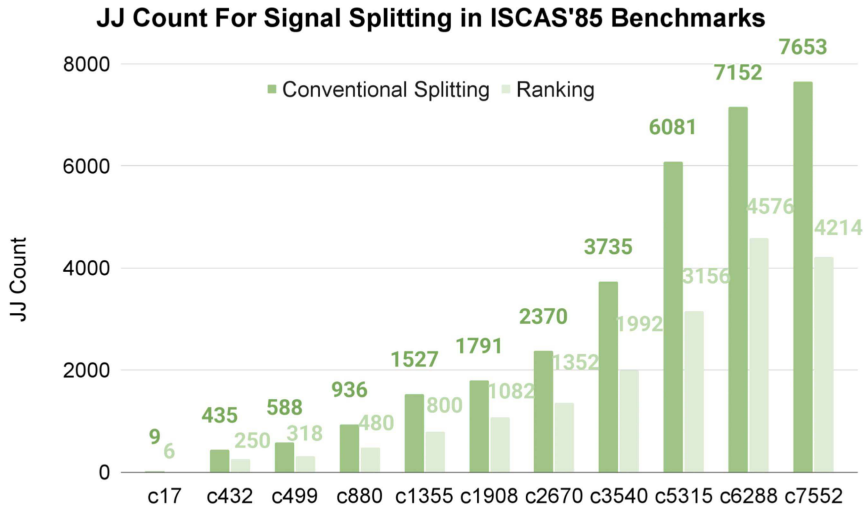

Fig. 19. The number of JJs required for signal fan-out in unmodified ISCAS'85 benchmark circuits is estimated for both conventional splitting and the proposed approaches. The average JJ savings with JJ sharing and cell ranking is 43.3%.

savings. The estimated bias margins of the traditional implementation are $+7.7\%/-23.1\%$, which are comparable to that of the rank-based version.

C. Modeling Larger Designs

Our next task is to move beyond functional testing and quantify the benefits of the proposed approach on a larger scale. To this end, we first use a clock tree with FO1024 as an example case and then analyze ISCAS'85 benchmark circuits.

Regarding the FO1024 tree, rank-1 source and target cells are assumed. For the construction of the tree, R1:8 amplifying JTL chains are used in a way that extends Fig. 12 and expands upon the fan-out achieved in Fig. 13. Our results indicate firstly that the design can be built with modularity, as the bias margins do not diminish beyond $+36.5\%/-19.2\%$. Second, the design, depicted in Fig. 18, uses 1608 JJs while a similar tree with conventional splitters costs 3069 JJs, resulting in a savings of 47.6%.

In the case of ISCAS'85 benchmarks [46], the fan-out requirements of each circuit are extracted by counting the fan-out for every signal in the corresponding Verilog file. The fan-out ranges from 1 to 16. In our analysis, we constrain fan-out per stage to FO8, as R1:8 JTL chains have shown to deliver satisfying bias margins. The JJ counts are shown in Fig. 19 and percent improvements in Fig. 20. A rank-based approach grants an

Benchmark	Improvement ($p_a = \sqrt{2}$)	Improvement ($p_a = 2$)
c17	33.3%	33.3%
c432	42.5%	50.3%
c499	45.9%	65.3%
c880	48.7%	60.5%
c1355	47.6%	55.1%
c1908	39.6%	47.6%
c2670	43.0%	51.9%
c3540	46.7%	56.5%
c5315	48.1%	58.7%
c6288	36.0%	53.9%
c7552	44.9%	56.0%
Average	43.3%	53.6%

Fig. 20. Percent improvements to JJ count for data signal splitting using ranking with $\sqrt{2}$ and 2 step sizes in ISCAS'85 benchmarks compared with the conventional splitter-based methodology.

average JJ savings of 43.3% for signal splitting, 32.3% for clock fan-out, and 10% for the total JJ count, even without taking into consideration the effects of path-balancing, which is expected to inflate these numbers further.

D. Increasing the Step Size

Finally, we analyze the effects of a parameter that so far has been constant: the step size. More specifically, a $\sqrt{2}$ amplification step size p_a has been assumed in all prior cases, following the principles of early SFQ [20] and microwave splitter design [43]. However, SPICE-level results indicate that a step of 2 within each JTL and a step of $\sqrt{2}$ between can potentially work well in the case of R1:8 JTL chains.[6] This arrangement, simulated with the same test setup and termination as in Section IV, has bias margins of $+23.1\%/-23.1\%$, which are somewhat diminished from those of a rank-1 to rank-8 chain with a step of $\sqrt{2}$: $+42.3\%/-23.1\%$. When combined with JJ sharing and cell ranking, the effects of intra-JTL p_a=2 on fan-out improvements to ISCAS'85 benchmarks are shown in Fig. 20. The results indicate an increase in average savings for signal splitting to 53.6%.

V. Design Trade-off Discussion

The trade-off space between circuit area, delay, power, and reliability is intricate and cell ranking serves as a new control to steer optimizations. In Section III-B, three design methodologies incorporating cell ranking were presented. In this section, we elaborate on the design trade-offs associated with each design methodology.

In the first methodology, shown in Fig. 7, all logic cells feature the same baseline current and, therefore, rank. Under this assumption, no change in existing cell designs and libraries is needed and splitting can be accomplished with cascaded amplifying JTLs. Note that JJ count instead of area is used to quantify resource efficiency improvements, as the provided results are simulation- and not fabrication-based. JJ count is frequently used as a first-order estimate of design complexity [25], [47] because it relates linearly to circuit area for closely-sized JJs and inductors, which is the case in this design methodology.

In the second, exemplified by Fig. 8, the lowest rank is picked as the baseline. A low rank comes with higher switching speeds, lower power dissipation, and smaller JJs. Smaller JJs do not directly imply a smaller area per cell, as JJ size is inversely related to the size of shunt resistors and inductors [6]. However, JJ count remains an important indicator of overall resource efficiency because it tracks component count. Regarding circuit reliability—a valid concern when fabricating large-scale designs—cells with baseline currents below $\sim$50 μA may suffer higher bit error rates at 4.2 K [6]. That said, the applicability of JJ sharing and cell ranking is independent of the above-discussed current values; thus, the minimum acceptable baseline current value for a particular design and process technology is configurable at design time.

In the third and final case, depicted in Fig. 9, the goal is to demonstrate the potential benefits of incorporating full flexibility in the cell ranking arrangement. The drawback of this approach is that the size of the corresponding cell libraries increases as multiple copies of the same gate, each with a different baseline current, are needed—this is similar to the case of complementary metal–oxide semiconductor (CMOS) cell libraries that feature various driving configurations. The advantage is that area, speed, energy, reliability, and fan-out overhead are left as free variables for potential optimization. The critical current range and step size serve as additional degrees of freedom; for example, small or large baseline critical currents can be incorporated or ignored.

In summary, each presented design methodology reveals a distinct set of relationships between rank and established metrics with various degrees of flexibility. The discussed trade-offs unwrap a plethora of optimization opportunities that are beyond the scope of this work. Nevertheless, the presented evaluation can function both as a high-level model and a good starting point for those optimizations while also providing strong evidence of the feasibility and foreseeable gains of JJ sharing and cell ranking.

VI. Conclusion

Advancements toward improving the computational density of superconductor systems commonly target the device, logic, and architecture levels. In this work, we take on a different approach and focus on cell abutment by maximizing the utility of JJs already present in each logic cell. The key idea behind our efforts is that boundary JJs can also be used for splitting purposes, thereby eliminating frequently-used splitter cells that account for up to $\sim$30% of the total number of JJs.

To facilitate this idea, we first proposed a JJ-sharing technique that examines the anatomy of existing SFQ cells and identifies JJs that are candidates for reuse. We secondly presented a ranking methodology that allows for the reliable stitching of SFQ cells with variable baseline critical currents in a way that abstracts analog design concerns. Our results indicate that such

[6]The first JTL amplifies from rank-1 to rank-3 (corresponding to the ranks in Fig. 6), the second from rank-4 to rank-6, and the third from rank-7 to rank-8 to compensate for the asymmetry given by this step size and critical current range.

designs are functionally correct, reduce current consumption, and exhibit satisfying bias margins. Rank-based JJ sharing is also shown to deliver 17.7% savings in the total JJ count of a 2-bit KSA, an average 43.3% savings for data signal splitting in ISCAS'85 benchmarks, and a 47.6% savings for a fan-out tree with FO1024, compared to the same designs with conventional splitting techniques. Moreover, we notice that a change in the amplification step size, from $\sqrt{2}$ to 2, can lead to an increase in average savings to 53.6% for data signal splitting in ISCAS'85 benchmarks at the cost of diminished bias margins. The integrity of this setup was evaluated through transient simulations, and further investigation is needed to identify additional limitations on this parameter.

Finally, the interplay between rank and various design metrics was analyzed in the context of three design methodologies. Each methodology offers different benefits and drawbacks, although greater flexibility in cell ranking tends to supply greater potential for optimization and, thus, greater potential for exploitation at the CAD-level.

ACKNOWLEDGMENT

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

REFERENCES

[1] D. S. Holmes and E. P. DeBenedictis, "Superconductor electronics and the international roadmap for devices and systems," in *Proc. IEEE 16th Int. Supercond. Electron. Conf.*, 2017, pp. 1–3.

[2] G. Li, H. Li, J.-S. Liu, and W. Chen, "Fabrication and characterization of superconducting RSFQ circuits," *Rare Met.*, vol. 38, no. 10, pp. 899–904, 2019.

[3] F. Feldhoff and H. Toepfer, "Niobium neuron: RSFQ based bio-inspired circuit," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1800505.

[4] H. He, Y. Yamanashi, and N. Yoshikawa, "Design of discrete Hopfield neural network using a single flux quantum circuit," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 4, Jun. 2022, Art. no. 1300604.

[5] K. Ishida et al., "SuperNPU: An extremely fast neural processing unit using superconducting logic devices," in *Proc. IEEE/ACM 53rd Annu. Int. Symp. Microarchitecture*, 2020, pp. 58–72.

[6] S. K. Tolpygo, "Superconductor digital electronics: Scalability and energy efficiency issues," *Low Temp. Phys.*, vol. 42, no. 5, pp. 361–379, 2016.

[7] S. K. Tolpygo and V. K. Semenov, "Increasing integration scale of superconductor electronics beyond one million Josephson junctions," *J. Phys.: Conf. Ser.*, vol. 1559, no. 1, 2020, Art. no. 012002.

[8] S. K. Tolpygo et al., "Superconductor electronics fabrication process with moNx kinetic inductors and self-shunted Josephson junctions," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 4, Jun. 2018, Art. no. 1100212.

[9] O. A. Mukhanov, S. V. Rylov, D. V. Gaidarenko, N. B. Dubash, and V. V. Borzenets, "Josephson output interfaces for RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 2826–2831, Jun. 1997.

[10] S. K. Tolpygo et al., "Properties of unshunted and resistively shunted Nb/ALOx-AL/Nb Josephson junctions with critical current densities from 0.1 to 1 mA/μm^2," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1100815.

[11] C. J. Fourie, "Electronic design automation tools for superconducting circuits," *J. Phys.: Conf. Ser.*, vol. 1590, no. 1, 2020, Art. no. 012040.

[12] G. Tzimpragos, J. Volk, A. Wynn, J. E. Smith, and T. Sherwood, "Superconducting computing with alternating logic elements," in *Proc. IEEE/ACM 48th Annu. Int. Symp. Comput. Architecture*, 2021, pp. 651–664.

[13] G. Tzimpragos et al., "Temporal computing with superconductors," *IEEE Micro*, vol. 41, no. 3, pp. 71–79, May/Jun. 2021.

[14] G. Tzimpragos et al., "A computational temporal logic for superconducting accelerators," in *Proc. 25th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2020, pp. 435–448, doi: 10.1145/3373376.3378517.

[15] O. Mukhanov, V. Semenov, and K. Likharev, "Ultimate performance of the RSFQ logic circuits," *IEEE Trans. Magn.*, vol. TMAG-23, no. 2, pp. 759–762, Mar. 1987.

[16] M. H. Volkmann, A. Sahu, C. J. Fourie, and O. A. Mukhanov, "Implementation of energy efficient single flux quantum digital circuits with sub-aJ/bit operation," *Supercond. Sci. Technol.*, vol. 26, no. 1, Nov. 2012, Art. no. 015002, doi: 10.1088/0953-2048/26/1/015002.

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

[18] M. Pedram, "Superconductive single flux quantum logic devices and circuits: Status, challenges, and opportunities," in *Proc. IEEE Int. Electron. Devices Meeting*, 2020, pp. 25.7.1–25.7.4. [Online]. Available: https://ieeexplore.ieee.org/document/9371914/

[19] H. Cong, M. Li, and M. Pedram, "An 8-bit multiplier using single-stage full adder cell in single-flux-quantum circuit technology," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 6, Sep. 2021, Art. no. 1303110. [Online]. Available: https://ieeexplore.ieee.org/document/9463754/

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

[21] L. Schindler, J. A. Delport, and C. J. Fourie, "The coldflux RSFQ cell library for MIT-ll SFQ5EE fabrication process," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 2, Mar. 2022, Art. no. 1300207.

[22] S. Polonsky et al., "New RSFQ circuits (Josephson junction digital devices)," *IEEE Trans. Appl. Supercond.*, vol. 3, no. 1, pp. 2566–2577, Mar. 1993.

[23] Z. J. Deng, N. Yoshikawa, S. Whiteley, and T. Van Duzer, "Data-driven self-timed RSFQ digital integrated circuit and system," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 3634–3637, Jun. 1997.

[24] L. Schindler and C. J. Fourie, "Application of phase-based circuit theory to RSFQ logic design," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 3, Apr. 2022, Art. no. 1300512.

[25] Cryogenic Electronics and Quantum Information Processing, "International roadmap for devices and sytems 2022 edition," Oct. 2022. Accessed: Oct. 16, 2022. [Online]. Available: https://irds.ieee.org/images/files/pdf/2022/2022_IRDS_CEQIP.pdf

[26] F. Brglez, "A neutral netlist of 10 combinational benchmark circuits and a translator in Fortran," in *Proc. Int. Symp. Circuits, Syst.*, 1985.

[27] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *IEEE Des. Test Comput.*, vol. 16, no. 3, pp. 72–80, Jul./Sep. 1999.

[28] F. Zokaee and L. Jiang, "Smart: A heterogeneous scratchpad memory architecture for superconductor SFQ-based systolic CNN accelerators," in *Proc. IEEE/ACM 54th Annu. Int. Symp. Microarchitecture*, 2021, pp. 912–924. [Online]. Available: https://doi-org.proxy.library.ucsb.edu:9443/10.1145/3466752.3480041

[29] C. Fourie, "Single flux quantum circuit technology and cad overview," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2018, pp. 1–6.

[30] G. Krylov and E. G. Friedman, "Design for testability of SFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 8, Dec. 2017, Art. no. 1302307.

[31] Y. Kameda, S. Yorozu, and Y. Hashimoto, "A new design methodology for single-flux-quantum (SFQ) logic circuits using passive-transmission-line (PTL) wiring," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 508–511, Jun. 2007.

[32] M. L. Schneider and K. Segall, "Fan-out and fan-in properties of superconducting neuromorphic circuits," *J. Appl. Phys.*, vol. 128, no. 21, Dec. 2020, Art. no. 214903, doi: 10.1063/5.0025168.

[33] T. Jabbari, E. G. Friedma, and J. Kawa, "H-tree clock synthesis in RSFQ circuits," in *Proc. IEEE 17th Biennial Baltic Electron. Conf.*, 2020, pp. 1–5.

[34] C.-C. Wang and W.-K. Mak, "A novel clock tree aware placement methodology for single flux quantum (SFQ) logic circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2021, pp. 1–9.

[35] T. Jabbari and E. G. Friedman, "Global interconnects in VLSI complexity single flux quantum systems," in *Proc. Workshop Syst.-Level Interconnect: Problems Pathfinding Workshop*, 2020, pp. 1–7.

[36] M. Dorojevets, P. Bunyk, and D. Zinoviev, "Flux chip: Design of a 20-GHz 16-bit ultrapipelined RSFQ processor prototype based on 1.75-/spl mu/m LTS technology," *IEEE Trans. Appl. Supercond.*, vol. 11, no. 1, pp. 326–332, Mar. 2001.

[37] C. J. Fourie, "Extraction of DC-biased SFQ circuit Verilog models," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 6, Sep. 2018, Art. no. 1300811.

[38] N. Katam, A. Shafaei, and M. Pedram, "Design of multiple fanout clock distribution network for rapid single flux quantum technology," in *Proc. IEEE 22nd Asia South Pacific Des. Automat. Conf.*, 2017, pp. 384–389.
[39] T. Jabbari, G. Krylov, J. Kawa, and E. G. Friedman, "Splitter trees in single flux quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1302606. [Online]. Available: https://ieeexplore.ieee.org/document/9395200/
[40] T. Yamada and A. Fujimaki, "A novel splitter with four fan-outs for ballistic signal distribution in single-flux-quantum circuits up to 50 Gb/s," *Japanese J. Appl. Phys.*, vol. 45, no. 3L, 2006, Art. no. L262.
[41] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, L. M. Johnson, M. A. Gouker, and W. D. Oliver, "Fabrication process and properties of fully-planarized deep-submicron Nb/Al-AlO_x/Nb Josephson junctions for VLSI circuits," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 3, Jun. 2015, Art. no. 1101312.
[42] T. Li, J. C. Gallop, L. Hao, and E. J. Romans, "Josephson penetration depth in coplanar junctions based on 2D materials," *J. Appl. Phys.*, vol. 126, no. 17, 2019, Art. no. 173901, doi: 10.1063/1.5124391.
[43] E. Wilkinson, "An N-way hybrid power divider," *IRE Trans. Microw. Theory Techn.*, vol. 8, no. 1, pp. 116–118, 1960.
[44] S. Polonsky, J. C. Lin, and A. Rylyakov, "RSFQ arithmetic blocks for DSP applications," *IEEE Trans. Appl. Supercond.*, vol. 5, no. 2, pp. 2823–2826, Jun. 1995. [Online]. Available: https://ieeexplore.ieee.org/document/403179/
[45] O. A. Mukhanov, S. V. Polonsky, and V. K. Semenov, "New elements of the RSFQ logic family," *IEEE Trans. Magn.*, vol. 27, no. 2, pp. 2435–2438, Mar. 1991.
[46] F. Brglez, "ISCAS85 combinational benchmark circuits," 2022. Accessed: Feb. 1, 2022. [Online]. Available: https://filebox.ece.vt.edu/mhsiao/iscas85.html
[47] N. Katam, A. Shafaei, and M. Pedram, "Design of complex rapid single-flux-quantum cells with application to logic synthesis," in *Proc. IEEE 16th Int. Supercond. Electron. Conf.*, 2017, pp. 1–3.

Jennifer Volk (Student Member, IEEE) received the B.S. degree in electrical engineering from UC Santa Cruz, Santa Cruz, CA, USA, in 2016, and the M.S. degree in electrical and computer engineering in 2021 from UC Santa Barbara, Santa Barbara, CA, where she is currently working toward the Ph.D. degree in electrical and computer engineering researching how to best utilize the quirks of superconductor electronics to benefit circuit and architecture design.

Georgios Tzimpragos received the M.S. degree in electrical and computer engineering from UC Davis, Davis, CA, USA, in 2016, and the Ph.D. degree in computer science from UC Santa Barbara, Santa Barbara, CA, in 2022. His alma mater is the National Technical University of Athens, Athens, Greece.

He is currently an Assistant Professor of Computer Science and Engineering with the University of Michigan, Ann Arbor, MI, USA. His research interests include logic-architecture co-optimizations for emerging applications and devices.

Alex Wynn received the B.A. degree in astronomy and physics from Boston University, Boston, MA, USA, in 2012, and the M.A. degree in physics from Vanderbilt University, Nashville, TN, USA, in 2014.

Since 2014, he has been working on the development of superconductor electronics technologies with MIT Lincoln Laboratory, Lexington, MA.

Evan Golden received the B.S. degree in engineering physics from the University of Colorado Boulder, Boulder, CO, USA, in 2017.

He is currently a Member of the Technical Staff with MIT Lincoln Laboratory, Lexington, MA, USA, and conducts research on superconducting electronics for classical and quantum computing. Before joining Lincoln Laboratory, he performed research at the University of Colorado and the National Institute for Standards and Technology.

Timothy Sherwood (Fellow, IEEE) received the B.S. degree in computer science and engineering from UC Davis, Davis, CA, USA, in 1998, and the M.S. and Ph.D. degrees in computer science and engineering from UC San Diego, San Diego, CA, in 2003.

He is currently a Professor of Computer Science with UC Santa Barbara, Santa Barbara, CA, specializing in the development of computing systems exploiting novel technologies. He is currently the Interim Dean of the College of Creative Studies.

Dr. Sherwood is a Fellow of the ACM.