Low-Cost Superconducting Fan-Out
with Repurposed Josephson Junctions

Jennifer Volk, Georgios Tzimpragos, Alex Wynn, Evan Golden and Timothy Sherwood

Abstract—Superconductor electronics (SCE) promise computer systems with orders of magnitude higher speeds and lower energy consumption than their complementary metal-oxide semiconductor (CMOS) counterpart. 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. Towards 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 paper, we ask if there is a way to reduce these overheads; propose the repurposing of JJs at the cell boundaries for fan-out; and establish a set of rules to discretize critical currents in a way that is conducive to this realignment. Finally, we demonstrate the accomplished gains through detailed analog simulations and modeling analyses. Our experiments indicate that the introduced method leads to a 48% savings in the JJ count in 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 fan-out in ISCAS’85 benchmarks.

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]–[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]–[14]. In this paper, 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) benchmark designs unveils 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 SFQ cell are typically used for buffering purposes, accounting for an additional ~20% of the JJ count [21], [22].

The goal of this paper 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 splitting into the logic cells themselves through a novel JJ borrowing technique. With JJ borrowing, we enable greater fan-out in a single stage. But to achieve this, JJ sizes must be tuned on a case-by-case basis to match the desired critical currents 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]. To make things simpler, we propose a new abstraction for designing multiple-fan-out SFQ circuits that is based on assigning discrete baseline critical currents, which are defined by the sizes of buffer JJs, according to their fan-out needs. We also derive a set of guidelines to systematically generate these assignments and provide a good starting point for further optimizations.

To validate this approach, we demonstrate functional correctness, wide bias margins, and improved resource efficiency for various example designs through a combination of detailed SPICE simulations and higher-level models. Particularly, JJ borrowing enables a fan-out of 8 in a single stage with the use of JJs at the edges of cells—boundary JJs—for splitting with ±40% 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 borrowing also gives an estimated average savings of 10% in the total JJ count within ISCAS’85 benchmarks [25], [26]—43.3% of the JJ count for signal splitting and 32.3% for clock fan-out—even without taking into consideration the effects of path-balancing, and 47.6% for a tree with fan-out of 1024 (FO1024) compared to the use of conventional splitters.

Overall, the main contributions of this paper are: (1) the repurposing of redundant boundary JJs for splitting purposes; (2) for accurate area results, complete circuit layouts are necessary. However, considering that inductor and resistor sizes scale inversely with junction size, JJ count is commonly used as a crude estimate of resource efficiency [24]; (3) Our design netlists are available at [https://github.com/UCSBarchlab/SFQ-Ranking].
Another improvement comes from a recently-proposed JJ increasing the number of output ports per splitter cell [38].

Splitters can be chained directly, one after the other, with no degradation in bias margins.

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 [27]–[30]. Moreover, high fan-out needs [31], especially in the case of conventional fully-synchronous SFQ designs [32]–[34], 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 [35]. A synthesized 8-bit general purpose RISC processor uses 21,065 splitters for clock distribution, accounting for 15.8% of the total 400,000 JJ count—and this does not include splitting on the data path [20]. 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 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’s 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.

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, substituting splitters with amplifying JTLs 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. To accomplish this, we minimize the fan-out cost through JJ borrowing and introduce a systematic way to manage the JJs that are borrowed. JJ borrowing, in contrast to other methods, encapsulates an approach that sources the JJs for splitting from existing logic cells, thereby mitigating the cost of splitters entirely.

**B. JJ Characterization**

To enable JJ borrowing, 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: those used in decision-making pairs [28], blocking JJs, and 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 backwards propagation of a SFQ towards the opposite input wire, as in the rightmost
Fig. 3. A signal splits from a D-flip flop (DFF) to the inputs of two OR cells. A conventional splitter enables 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, the OR cell’s buffer JJs can be borrowed for splitting.

Fig. 4. The consequences of borrowing the two OR cells’ input JJs identified in Figure 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.

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 Borrowing

The ideal JJs for borrowing 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 borrowing. 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 borrowing. 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 borrowed JJs have the same sizes before borrowing. 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 DFF’s decision-making pair $I_C$ is set to 355 µA to match the $I_C$ of the splitter’s input JJ, then it can be borrowed for splitting. The result is shown in Fig. 5, in which all other JJs in the DFF are scaled by the same number. After borrowing twice, all of the JJs in the original splitter of Fig. 3 have been 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 optimizations are beyond the scope of this paper, 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, a 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 their neighbors’, however, turns every fan-out point into a unique calculation, which complicates the design process.
We create an abstraction in which critical currents are discrete. 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 bias current, 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 starting from 250 µA, as it is a popular baseline critical current for SFQ libraries [20], [21] and used in the prior example in Fig. 3 and Fig. 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 \( \frac{1}{\sqrt{2}} \) and \( \sqrt{2} \) respectively. Going down, we reach 180 µA, 125 µA, 88 µA, and 66 µA in turn, and finally stop at 46 µA. Below this point, JJs become more prone to thermal errors [6]. Applying the same approach to the other side, 353 µA is found after one step and 500 µA after two. We choose 500 µA as the upper bound, but in practice the JJ diameter is limited by the Josephson penetration depth [31] and practical shunt resistor size. We number each step from one, the lowest current, to eight, the highest current, to assign their respective ranks.

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

\[
N_r \geq \left\lceil \frac{\log(I_{C,Max})}{\log(p_r)} \right\rceil + 1
\]  

(1)

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

\[
N_{JTL} \geq \left\lceil \frac{\log(I_{T}/I_{S})}{2 \log(p_a)} \right\rceil
\]

(2)

The above inequalities stem from the fact that \( N_r, N_{JTL} \in \mathbb{N} \) 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 assignment is beyond the scope of this paper.

Fig. 6 shows a lookup table that describes the connectivity between various ranks for the range discussed above and \( p_r = p_a = \sqrt{2} \). A cell with higher rank, or larger baseline critical current, can be directly connected to any one with a lower rank. Consider an example in which four target cells’ ranks must be found for a design that has a 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 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 additional 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 | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    |
|------|------|------|------|------|------|------|------|------|
| I_c  | 500µA| 353µA| 250µA| 180µA| 125µA| 88µA | 66µA | 46µA |
| FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +1 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +2 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +3 JTL | +2 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +3 JTL | +3 JTL | +2 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +4 JTL | +3 JTL | +3 JTL | +2 JTL | +2 JTL | +1 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +4 JTL | +4 JTL | +3 JTL | +3 JTL | +2 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +4 JTL | +4 JTL | +3 JTL | +3 JTL | +2 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |
| +4 JTL | +4 JTL | +3 JTL | +3 JTL | +2 JTL | +2 JTL | +1 JTL | FO1  | FO2  | FO3  | FO4  | FO5  | FO6  | FO7  | FO8  |

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 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 level 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 sub-linearly with the distance between ranks, as one additional step in rank can be gained between amplifying JTLs.

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 bias current, and finally compare and contrast three different rank-based design methodologies.
rank-6 column in Fig. 6 indicate that two amplifying JTLs are needed for the connection. The first of the two JTLs amplifies 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 used in more than one way depending on the designer’s target goals. Below, we use a DFF that fans out to four more DFFs as a running example and show how one can apply ranking to optimize for design reuse, energy, and JJ count.

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 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. Unlike the first case, however, because the starting rank is the lowest, it is more likely to save JJs at higher fan-outs compared to conventional splitting. Fig. 8 depicts the resulting schematic. In this case, only the initial JTL chain is needed before fan-out and just two JTLs are inserted. The JJ count has five fewer JJs than 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 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.

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 quantified and compared. Each JTL element consists of two JJs, four inductors, and a single resistor that biases both JJs. Each JTL chain that amplifies the rank from $R_s$ to $R_t$ is labeled with $R_s:t$. For a JTL that is not amplifying, $s = t$.

A JTL chain without amplification is designed with rank-6 JTLs. Bias margin estimates 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], [42] is depicted in Fig. 10 and also has margins of $+38.5\% / -65.4\%$. 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 $+38.5\%/−57.7\%$. 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 $\mu$A, has bias margins of $+30.8\%/−53.8\%$.

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 a FO8 in total. The schematic is shown in Fig. 12. The bias margins in this case are $+38.5\%/−46.2\%$. By comparison, a splitter tree with FO8, composed of the same splitters described above, has bias margins of $+26.9\%/−34.6\%$.

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 $+42.3\%/−46.2\%$. This serves as an estimate 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. The lower margin is diminished slightly after the increase to FO8, and again when adding the longer amplification chain. 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 $+38.5\%/−46.2\%$, which are similar to those of the FO3 case above and reveal that the design method indeed promotes modularity.

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

| Configuration          | Ranking    | Splitters |
|------------------------|------------|-----------|
| R6:8 (Fig. 10)         | $+38.5\%/−65.4\%$ | -         |
| R6:8+FO3 (Fig. 11)     | $+38.5\%/−57.7\%$ | $+30.8\%/−53.8\%$ |
| R6:8+FO8 (Fig. 12)     | $+38.5\%/−46.2\%$ | $+26.9\%/−34.6\%$ |
| R1:8+FO8               | $+42.3\%/−46.2\%$ | $+26.9\%/−34.6\%$ |
| R6:8+FO9 (Fig. 13)     | $+38.5\%/−46.2\%$ | $+26.9\%/−46.2\%$ |

2) Current Savings Analysis: According to the above presented analyses, the proposed rank-based design methodology shows promise for better resource utilization without bias margin degradation. However, it is not clear that 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 FO2, FO4, and FO8 from various logic cells and compare the ranking methodology to conventional splitting. Our results are shown in Fig. 15.
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 ranking that uses the smallest bias current overhead. In the second option, the source and target ranks match.

B. Building Rank-Based Circuits

Next, a 2-bit KSA is used as a comprehensible example to demonstrate how ranks can be used to simplify the design of circuits with flexible baseline $I_C$s. The block diagram of the adder design with counter-flow clocking 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. 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 1.** 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. 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. As before, JTLs are inserted to enforce rank and to 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, SFQ converters, and JTLs. The bias margins are estimated to be $+19.2%$/$−3.8%$.

2) **Step 2.** 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 value. In Fig. 16 we start at the stage found in 1 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 a FO1 connection are noted to have the same ranks as their connected neighbors, which are not yet known.

3) **Step 3.** Aim to keep target cells the same rank if they share a source cell. In this case, the two bottom cells in the first stage are forced to be rank-2, the second cell in the second stage is forced to be rank-2, and the two middle stages in the first stage are forced to be the same rank, which is still unknown. The fourth cell in the second stage is also assigned rank-2, based on the sharing that occurred in Step 3. The cells that have yet to be ranked at this point are the top three in the first stage and the third cell in the second stage.

4) **Step 4.** 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 ranks in this example: the top three cells in the first stage and the third cell in the second stage are all assigned rank-3. This rank 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 from the rank-2 source cells to the two rank-2 target cells. It is also now clear that inputs A0 and B0 should be sourced from rank-4 cells, and A1 and B1 should be sourced from rank-3 cells.

5) **Step 5.** The last step in the shown example is about the design of the clock tree. 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 then fan-out to three rank-2 cells each, while two are reserved for sharing amongst the remaining rank-2 and rank-3 cells. 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], [43], [44]. To satisfy the requirement for multi-rank cells, Stonybrook’s designs are scaled to the required rank and individually tested before being stitched together. JTLs are inserted to enforce rank and to 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 estimated to be $+19.2%$/$−3.8%$.

To quantify the gains compared to a traditional approach, we reimplement the same 2-bit KSA design, but this time without the proposed borrowing and ranking techniques. The achieved results indicate that a total of 385 JJs are needed for the tradi-
Fig. 16. Block diagram of a fully-synchronous 2-bit Kogge Stone Adder (KSA) designed in Cadence using the MITLL SFQ5ee 100 µA/µm² fabrication process. Ranks are assigned to the design using a five-step methodology. The clock line is designed to be counter-flow to help prevent timing violations. The proposed borrowing and flexible I_C assignment saves 17.7% of JJs compared to the conventional splitting method.

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_OUT = 0, S_1 = 1, and S_0 = 1, appears five clock cycles later.

Fig. 18. Block diagram of a fully-synchronous 2-bit Kogge Stone Adder (KSA) designed in Cadence using the MITLL SFQ5ee 100 µA/µm² fabrication process. Ranks are assigned to the design using a five-step methodology. The clock line is designed to be counter-flow to help prevent timing violations. The proposed borrowing and flexible I_C assignment saves 17.7% of JJs compared to the conventional splitting method.

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 +38.5% / −46.2%. Secondly, 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 [45], 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 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
Lastly, we analyze the effects of a parameter that so far has been constant: the step size. More specifically, a √2 amplification step size p_a has been assumed in all prior cases, following the principles of early SFQ [20] and microwave splitter design [42]. However, SPICE-level results indicate that a step of 2 within each JTL can potentially work equally well in the case of R1:8 JTL chains, which are simulated to have the same bias margins as a chain with a √2 amplification step: +42.3% / −53.8%. When combined with JJ borrowing and 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. Conclusion
Advancements towards improving the computational density of superconductor systems commonly target the device, logic,
Fig. 18. A FO1024 tree using ranking. Chains of JTLs followed by 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 a FO1024 tree using conventional splitters costs 3069 JJs—a savings of 47.6%.

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

| 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 to the conventional splitter-based methodology.

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 firstly proposed a JJ borrowing 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. Although flexibly designing with rank-based JJ borrowing comes with some overhead in the cell library, as multiple copies of each cell may be needed, we have also presented design methodologies that only require the use of amplifying JTLs. Our results indicate
that such designs are functionally correct, reduce current consumption, and exhibit satisfying bias margins. Rank-based JJ borrowing is also shown to deliver 17.7% savings in the total JJ count in 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 √2 to 2, can lead to an increase in average savings to 53.6% for data signal splitting in ISCAS’85 benchmarks. The integrity of systems based on a step size of 2 has so far been investigated at the transient simulation level only, and further investigation is needed to identify the limitations on this parameter.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grants No. 2006542 and 1763699 and the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. 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 2017 16th International Superconductor Electronics Conference (ISEC). IEEE, 2017, pp. 1–3.
[2] G. Li, H. Li, J.-S. Liu, and W. Chen, “Fabrication and characterization of superconducting rsfq circuits,” Rare Metals, vol. 38, no. 10, pp. 899–904, 2019.
[3] F. Feldhoff and H. Toepfer, “Niobium neuron: Rsfq based bio-inspired circuit,” IEEE Transactions on Applied Superconductivity, vol. 31, no. 5, pp. 1–5, 2021.
[4] H. He, Y. Yamashita, and N. Yoshikawa, “Design of discrete hopfield neural network using a single flux quantum circuit,” IEEE Transactions on Applied Superconductivity, 2021.
[5] K. Ishida, I. Umemura, M. Nakajima, K. Fuku, and M. Tanaka, S. Kawai, T. Tanikawa, and T. Inoue, “Supernpu: An extremely fast neural processing unit using superconducting logic devices,” in 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2020, pp. 58–72.
[6] S. K. Töpplö, “Superconductor digital electronics: Scalability and energy efficiency issues,” Low Temperature Physics, vol. 42, no. 5, pp. 361–379, 2016.
[7] S. K. Töpplö and V. K. Semenov, “Increasing integration scale of superconductor electronics beyond one million josephson junctions,” in Journal of Physics: Conference Series, vol. 1559, no. 1. IOP Publishing, 2020, p. 012002.
[8] S. K. Töpplö, V. Bolkhovsky, D. E. Oates, R. Rastogi, S. Zarr, A. L. Day, T. J. Wier, A. Wynn, and L. M. Johnson, “Superconductor electronics fabrication process with moxn kinetic inductors and self-shunted josephson junctions,” IEEE Transactions on Applied Superconductivity, vol. 28, no. 4, pp. 1–12, 2018.
[9] O. A. Mukhanov, S. V. Rylov, D. V. Gaidarenko, N. B. Dubash, and V. V. Borzenets, “Josephson output interfaces for rsfq circuits,” IEEE Transactions on Applied Superconductivity, vol. 7, no. 2, pp. 2826–2831, 1997.
[10] S. K. Töpplö, V. Bolkhovsky, S. Zarr, T. Wier, A. Wynn, A. L. Day, L. Johnson, and M. Gouker, “Properties of unshunted and resistively shunted nb/al/alx-al/nb josephson junctions with critical current densities from 0.1 to 1 ma/μm2,” IEEE Transactions on Applied Superconductivity, vol. 27, no. 4, pp. 1–15, 2017.
[11] C. J. Faurie, “Electronic design automation tools for superconducting circuits,” in Journal of Physics: Conference Series, vol. 1590, no. 1. IOP Publishing, 2020, p. 012040.
[12] G. Tzimpragos, J. Volk, A. Wynn, J. E. Smith, and T. Sherwood, “Superconducting computing with alternating logic elements,” 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), pp. 651–664, 2021.
[13] G. Tzimpragos, J. Volk, D. Vasudevan, N. Tsiskaridze, G. Michelogiannakis, A. Madhavan, J. Shalf, and T. Sherwood, “Temporal computing with superconductors,” IEEE Micro, vol. 41, no. 3, pp. 71–79, 2021.
[14] G. Tzimpragos, D. Vasudevan, N. Tsiskaridze, G. Michelogiannakis, A. Madhavan, J. Volk, J. Shalf, and T. Sherwood, A Computational Temporal Logic for Superconducting Accelerators. New York, NY, USA: Association for Computing Machinery, 2020, p. 435–448. [Online]. Available: https://doi.org/10.1145/3373376.3378517.
[15] O. Mukhanov, V. Semenov, and K. Likharev, “Ultimate performance of the rsfq logic circuits,” IEEE Transactions on Magnetics, vol. 23, no. 2, pp. 759–762, 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-ul/bit operation,” Superconductor Science and Technology, vol. 26, no. 1, p. 015002, nov 2012. [Online]. Available: https://doi.org/10.1088/0953-8042/26/1/015002.
[17] O. A. Mukhanov, “Energy-efficient single flux quantum technology,” IEEE Transactions on Applied Superconductivity, vol. 21, no. 3, pp. 760–769, 2011.
[18] M. Pedram, “Superconductive Single Flux Quantum Logic Devices and Circuits: Status, Challenges, and Opportunities,” in 2020 IEEE International Electron Devices Meeting (IEDM). San Francisco, CA, USA: IEEE, Dec. 2020, pp. 25.7.1–25.7.4. [Online]. Available: https://ieeexplore.ieee.org/document/9371914/.
[19] H. Kong, M. Li, and M. Pedram, “An 8-b Multiplier Using Single-Stage Full Adder Cell in Single-Flux-Quantum Circuit Technology,” IEEE Transactions on Applied Superconductivity, vol. 31, no. 6, pp. 1–10, Sep. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9453754/.
[20] K. Likharev and V. Semenov, “Rsfq logic/memory family: a new josephson-junction technology for sub-teraertz-clock-frequency digital systems,” IEEE Transactions on Applied Superconductivity, vol. 1, no. 1, pp. 3–28, 1991.
[21] L. Schindler, J. A. Delport, and C. J. Fourie, “The coldflux rsfq cell memory for mit-il rsfq fabrication process,” IEEE Transactions on Applied Superconductivity, vol. 32, no. 2, pp. 1–7, 2022.
[22] S. Polonsky, V. Semenov, P. Bunyk, A. Kirichenko, A. Kidiyarov-Shevchenko, O. Mukhanov, P. Shevchenko, D. Schneider, D. Zinoviev, and K. Likharev, “New rsfq circuits (josephson junction digital devices),” IEEE Transactions on Applied Superconductivity, vol. 3, no. 1, pp. 2566–2577, 1993.
[23] J. Deng, N. Yoshikawa, S. Whiteley, and T. Van Duzer, “Data-driven self-timed rsfq digital integrated circuit system,” IEEE transactions on applied superconductivity, vol. 7, no. 2, pp. 3634–3637, 1997.
[24] C. Electronics and Q. I. Processing, “International roadmap for devices and systems 2022 edition,” https://irds.ieee.org/images/files/pdf/2022/2022_IRDS_CEQIP.pdf October 2022, accessed: 2022-10-16.
[25] F. Biglitz, “A neutral netlist of 10 combinational benchmark circuits and a translator in fortran,” in Proc. 1985 ISCAS, 1985.
[26] M. C. Hansen, H. Yalcin, and J. P. Hayes, “Unveiling the iscas-85 benchmarks: A case study in reverse engineering,” IEEE Design & Test of Computers, vol. 16, no. 3, pp. 72–80, 1999.
[27] F. Zokaei and L. Jiang, “Smart: A heterogeneous scratchpad memory architecture for superconductor sfq-based systolic cnn accelerators,” in MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture, ser. MICRO ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 912–924. [Online]. Available: https://doi-org.proxy.library.ucsb.edu:9443/10.1145/3466752.3480041.
[28] C. Faurie, “Single flux quantum circuit technology and cad overview,” in 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2018, pp. 1–6.
[29] G. Krylov and E. G. Friedman, “Design for testability of sfq circuits,” IEEE Transactions on Applied Superconductivity, vol. 27, no. 8, pp. 1–7, 2017.
[30] Y. Kameda, S. Yorozu, and Y. Hashimoto, “A new design methodology for single-flux-quantum (sfq) logic circuits using passive-transmission-line (plt) wiring,” IEEE Transactions on Applied Superconductivity, vol. 17, no. 2, pp. 508–511, 2007.
[31] L. Schindler and K. Likharev, “Fan-out and fan-in properties of superconducting neuromorphic circuits,” Journal of Applied Physics, vol. 128, no. 21, p. 214903, Dec. 2020. [Online]. Available: http://aip.scitation.org/doi/10.1063/5.0025168.
[32] T. Jabbari, E. G. Friedma, and J. Kawa, “H-tree clock synthesis in rsfq circuits,” in 2020 17th Biennial Baltic Electronics Conference (BEC), 2020, pp. 1–5.
[33] C.-C. Wang and W.-K. Mak, “A novel clock tree aware placement methodology for single flux quantum (sfq) logic circuits,” in 2021 IEEE/ACM International Conference On Computer Aided Design (IC-CAD). IEEE, 2021, pp. 1–9.
[34] T. Jabbari and E. G. Friedman, “Global interconnects in VLSI complexity single flux quantum systems,” in Proceedings of the Workshop on System-Level Interconnect: Problems and Pathfinding Workshop, 2020, pp. 1–7.
[35] 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 Transactions on Applied Superconductivity, vol. 11, no. 1, pp. 326–332, 2001.
[36] 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 transactions on applied superconductivity, vol. 17, no. 2, pp. 508–511, 2007.
[37] C. J. Fourie, “Extraction of dc-biased sfq circuit verilog models,” IEEE Transactions on Applied Superconductivity, vol. 28, no. 6, pp. 1–11, 2018.
[38] N. Katam, A. Shafaei, and M. Pedram, “Design of multiple fanout clock distribution network for rapid single flux quantum technology,” in 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), 2017, pp. 384–389.
[39] T. Jabbari, G. Krylov, J. Kawa, and E. G. Friedman, “Splitter trees in single flux quantum circuits,” IEEE Transactions on Applied Superconductivity, vol. 31, no. 5, pp. 1–6, Aug. 2021. [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 journal of applied physics, vol. 45, no. 3L, p. L262, 2006.
[41] T. Li, J. C. Gallop, L. Hao, and E. J. Romans, “Josephson penetration depth in coplanar junctions based on 2d materials,” Journal of Applied Physics, vol. 126, no. 17, p. 173901, 2019. [Online]. Available: https://doi.org/10.1063/1.5124391
[42] E. Wilkinson, “An n-way hybrid power divider,” IRE Transactions on Microwave Theory and Techniques, vol. 8, no. 1, pp. 116–118, 1960.
[43] S. Polonsky, Jao Ching Lin, and A. Rylyakov, “RSFQ arithmetic blocks for DSP applications,” IEEE Transactions on Applied Superconductivity, vol. 5, no. 2, pp. 2823–2826, Jun. 1995. [Online]. Available: http://ieeexplore.ieee.org/document/403179/
[44] O. A. Mukhanov, S. V. Polonsky, and V. K. Semenov, “New elements of the rsfq logic family,” IEEE transactions on magnetics, vol. 27, no. 2, pp. 2435–2438, 1991.
[45] F. Brglez, “Iscas85 combinational benchmark circuits,” https://filebox.ece.vt.edu/~mhsiao/iscas85.html, 02 2022, accessed: 2022-02-01.