Multi-Input SFQ Multiplexers

Tejumadejesu Oluwadamilare, *Graduate Student Member, IEEE*, and Eby G. Friedman, *Life Fellow, IEEE*

***Abstract*—A multiplexer is a digital structure that controls signal flow, selecting among multiple inputs to produce a single output. A two-input to one output (2:1) multiplexer is commonly used in single flux quantum (SFQ) systems. When three or more inputs to one output are required, 2:1 SFQ multiplexers are typically cascaded to create a multiplexer tree. A multi-input SFQ multiplexer comprising both a three-input (3:1) and four-input (4:1) to one output SFQ multiplexer is presented. These multiplexers require fewer control signals without cascading multiple 2:1 multiplexers. This SFQ circuit function is achieved using an XOR gate, AND gate, NOR gate, non-destructive read out cells, splitters, and confluence buffers for the 3:1 multiplexer. The XOR gate in the 3:1 multiplexer is replaced with two gates that distinguish between the 01 and 10 control signals. The circuit topologies are presented, and the performance and control signals are compared with a conventional 2:1 multiplexer tree and an individual input control multi-input multiplexer. A combination of 2:1, 3:1, and 4:1 multiplexers is proposed to achieve higher performance, leveraging the advantages of each type of multiplexer. A speed increase of 36% and an 83.6% reduction in the number of control signals in a 54:1 multi-input SFQ multiplexer are demonstrated using the combined multiplexer tree.**

***Index Terms*—RSFQ, SFQ multiplexer, SFQ gates, multi-input SFQ multiplexer, SFQ XOR gates.**

I. Introduction

SINGLE flux quantum (SFQ) technology is a standard logic family in superconductive electronics [1]. The low power dissipation and high speed operation contribute to the growing popularity of SFQ circuits [1], [2]. A multiplexer (MUX) is an important digital structure in electronic systems to manage signal flow and decode addresses in SFQ-based processors. Multiplexers alter the flow of the digital signals by dynamically selecting one of the transmitted signals based on control signals. This approach makes a multiplexer highly useful in data processing and memory management [2], [3], [4]. In ALU operations, multiplexers select data signals to be processed. The control signals are a vital part of a multiplexer because of the role played in data selection. The control signals activate or deactivate some parts of the multiplexer to select an input signal. As SFQ devices scale, a large number of control signals are required in multiplexers with multiple inputs. In short distance on-chip interconnect, an SFQ control pulse is transferred through Josephson transmission lines (JTL). A JTL is an active transmission line which provides noise discrimination by regenerating SFQ pulses to maintain signal integrity [5]. The large number of control signals, however, increases the interconnect within a system.

A two-input multiplexer is presented in [2], [3], where a set and reset control signal selects either of the two data signals. When two or more data signals are connected to a single output, such as in a switch, a multi-input SFQ multiplexer is necessary. As the number of inputs in a multiplexer tree increases, additional control signals are required. To reduce both the number of control signals and the depth of the tree in a multi-input multiplexer, 3:1 and 4:1 multiplexers are proposed here.

The paper is structured as follows: a brief background and different types of multi-input multiplexers are discussed in Section II. The operating principles of the three-input and four-input multiplexer and a comparison of different multi-input multiplexer topologies are described in Section III. A multi-input multiplexer tree composed of different types of multiplexers is discussed in Section IV. Some conclusions are drawn in Section V.

II. SFQ Multiplexer

Digital multiplexers often feature systems where certain components operate synchronously while other components function asynchronously. Multiplexers are often treated as asynchronous circuits due to the independent global clock signals. Multiplexers operate at the data rate of the control signals [6] and are compatible with a globally asynchronous, locally synchronous (GALS) environment [6]. The operating principles of an SFQ multiplexer are discussed in this section. The operation of the two-input SFQ multiplexer is described in Section II-A. The multi-input multiplexer is discussed in Section II-B.

A. Two-Input SFQ Multiplexer

Two-input SFQ multiplexers are of two types: analog [2] and digital. The analog multiplexer is controlled by set and reset pulses which switch the state of the symmetric superconductive loop, *J2, J12, L, J10,* and *J7*, as shown in Fig. 1(a). The state of the loop determines which data signal at the input of the MUX appears at the output. The presence of flux in the loop represents state "1" while the absence of flux represents state "0." The "0" state is achieved using a reset pulse while the "1" state is established by a set pulse. When input pulses are applied to both *IN1* and *IN2* while the state is "1," *J7* transmits the received

Received 19 September 2024; revised 12 December 2024 and 16 January 2025; accepted 17 January 2025. Date of publication 4 February 2025; date of current version 25 February 2025. This work was supported in part by National Science Foundation under Grant 2124453 (DISCOVER Expeditions) and Grant 2308863, in part by the Department of Energy under FOA DE-FOA-0002950, and in part by the Singapore Ministry of Education under Grant MOE2019-T2-2-075. *(Corresponding author: Tejumadejesu Oluwadamilare.)*

The authors are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: toluwada@ur.rochester.edu; friedman@ece.rochester.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASC.2025.3538651.

Digital Object Identifier 10.1109/TASC.2025.3538651

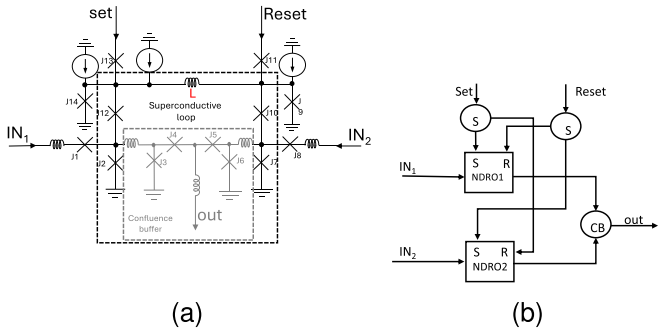

Fig. 1. Two-input multiplexer, (a) analog, and (b) digital, (CB - confluence buffer, S - splitter).

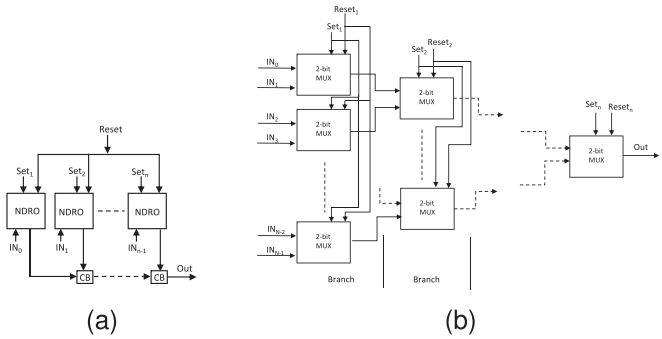

Fig. 2. Multi-input multiplexer, (a) individual input control, and (b) multiplexer tree.

pulse but the signal from *IN1* does not propagate to *J4*. This process occurs because the set signal induces a 2π phase shift in *J12* rather than *J2* which prevents *J2* from receiving sufficient bias current due to the large critical current I_c of *J2*. In contrast, when the state is "0" (indicating a reset signal), *J10* switches rather than *J7* and blocks the path of *IN2*. As a result, only *IN1* appears at the output.

As shown in Fig. 1(b), a digital MUX is composed of two nondestructive read out (NDRO) circuits, a confluence buffer, and two splitters. A digital SFQ MUX is also controlled by set and reset signals, where the set signal activates one of the NDROs and resets the other NDRO. The reset signal behaves in the opposite fashion, as depicted in Fig. 1(b). The confluence buffer in both types of multiplexers merges the data signals from each branch into a single output. The set and reset signals are only required when changing the path of the input signals at the ports and are not necessary for each individual SFQ pulse.

B. Multi-Input SFQ Multiplexer

A multi-input SFQ multiplexer has three or more inputs that are passed to a single output. One data signal at each input port is selected at a time. Several multi-input multiplexer topologies include individual input control and multiplexer trees, which are discussed below:

1) Individual Input Control: This method requires NDROs, confluence buffers, and splitters. Each NDRO is activated by a set signal. A common signal deactivates all of the NDROs to reset the entire multiplexer before another input is selected [7], as depicted in Fig. 2(a). The total number of control signals (N_c)

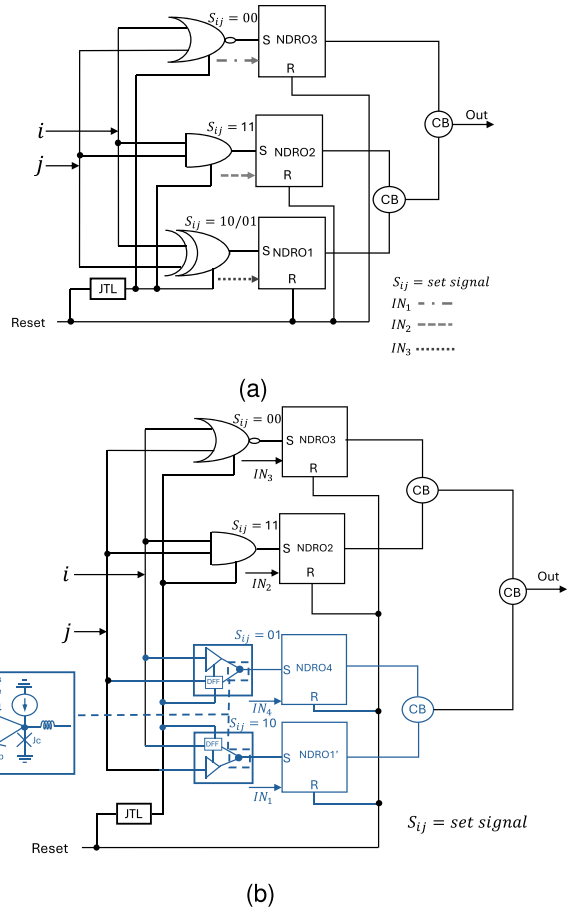

Fig. 3. Multi-input topologies, (a) three-input multiplexer, and (b) four-input multiplexer. The redundant logic selection for NDRO1 in the 3:1 MUX distinguishes between the control signals ij which are 01 and 10 to expand the 3:1 multiplexer to a 4:1 multiplexer.

required for this method is $N_c = N_i + 1$ where N_i is the number of inputs. Multiple control signals are necessary for large multi-input multiplexers. Individual input control is more suitable for multi-input multiplexers with fewer inputs.

2) Multiplexer Tree: This method is operated by set and reset signals. Two or more two-input multiplexers, as depicted in Fig. 3, are cascaded to achieve different size multi-input multiplexers, as depicted in Fig. 2(b) [8]. A set and reset signal are required at each branch to effectively control the multiplexer. Due to the fewer number of control signals, a multiplexer tree is preferable in a larger multi-input multiplexer. N_c of the MUX tree is $2\lceil \log_2(N_i) \rceil$ where the ceiling function $\lceil \cdots \rceil$ rounds a number up to the nearest integer.

III. Three-Input Multiplexer

The three-input multiplexer is discussed in this section. The operating principle is described in Section III-A. A comparison between an existing two-input multiplexer and a three-input multiplexer is provided in Section III-B.

A. Operating Principle of Three-Input Multiplexer

A three-input and four-input digital SFQ multiplexer are described in this subsection. The 3:1 multiplexer uses an XOR

gate, a NOR gate, an AND gate, three NDROs, two confluence buffers, splitters, and JTLs. Two control signals, i and j, and a reset signal select among the three data signals at the input ports of the multiplexer. The gates are clocked with a delayed reset signal. The delay is achieved using a JTL, allowing the i and j signals to reach the input of the gates before the arrival of the next clock pulse [9]. The output of the logic gates, S_{ij}, determines the set signal for the NDROs. When ij is 00, the NOR gate sends a pulse that activates NDRO3, selecting IN_1, as depicted in Fig. 3(a), while the other NDROs remain inactive. When ij is 11, the AND gate sends a pulse that activates NDRO2 and selects IN_2. IN_3 is selected when ij is either 10 or 01, enabling the XOR gate to send a pulse that turns on NDRO1. The output of each NDRO is merged by two confluence buffers, which read the selected signal at the output. After selecting each input signal, a reset signal deactivates the NDROs whenever a change of signal is required. The gates operate without a clock pulse but require a delayed reset signal. The data rate of the reset signal must match the data rate of the control signals, i and j, to ensure that the control signals arrive at the appropriate gate before the arrival of the clock pulse, which is the delayed reset signal.

Optimization of the redundant logic at the XOR gate in a 3:1 multiplexer expands this technique to a four-input multiplexer. The XOR gate generates the set pulse for the NDRO1 with either the 10 or 01 input control signal. To achieve a 4:1 multiplexer, the control combinations, 01 and 10, must be distinguished. This distinction requires two specialized XOR gates, each designed to handle one of these cases, as illustrated in Fig. 3(b). The specialized XOR gates require an inverter at one input and a path balancing DFF at the other input to ensure that the signal produced after the inverter and DFF are $-i, j$ or $i, -j$. A clockless AND gate, depicted in Fig. 3(a), only generates the set pulses for NDRO1' and NDRO4 when 11 is received from both the DFF and inverter. I_c of junction J_c of the clockless gate is larger to prevent switching unless two pulses are simultaneously received. Pulse $-i, j$ produces a set signal from a specialized XOR gate to activate NDRO4 while $i, -j$ activates NDRO1'. Aside from the 01 and 10 XOR gates and NDRO4 and NDRO1', all of the other connections are similar to the aforementioned 3:1 multiplexer.

The 3:1 and 4:1 SFQ multiplexer are both demonstrated using four different input signal frequencies, 20 GHz, 10 GHz, 5 GHz, and 13 GHz, respectively, at IN_1, IN_2, IN_3, and IN_4. The control signals i, j and the reset signal are received at a frequency of 1.67 GHz, as depicted in Fig. 4. Different input frequencies and delays differentiate the input signals at the output as the reset signal is received at different time intervals. Similarly, the control signals are set to 600 ps, allowing each input signal sufficient time to propagate before transitioning. Note that the multiplexers do not operate at these frequencies.

The input signals are highly dependent on the control signals, as previously stated. For the 3:1 MUX, after the i, j, and reset signals are received, a 20 ps delay occurs before the gates generate the set signals to activate the NDROs, as depicted in Fig. 5. This delay represents the setup time (T_{setup}) after arrival of the control signals. For the 4:1 MUX, T_{setup} is 23 ps. The difference in T_{setup} arises from the clockless AND gates in the

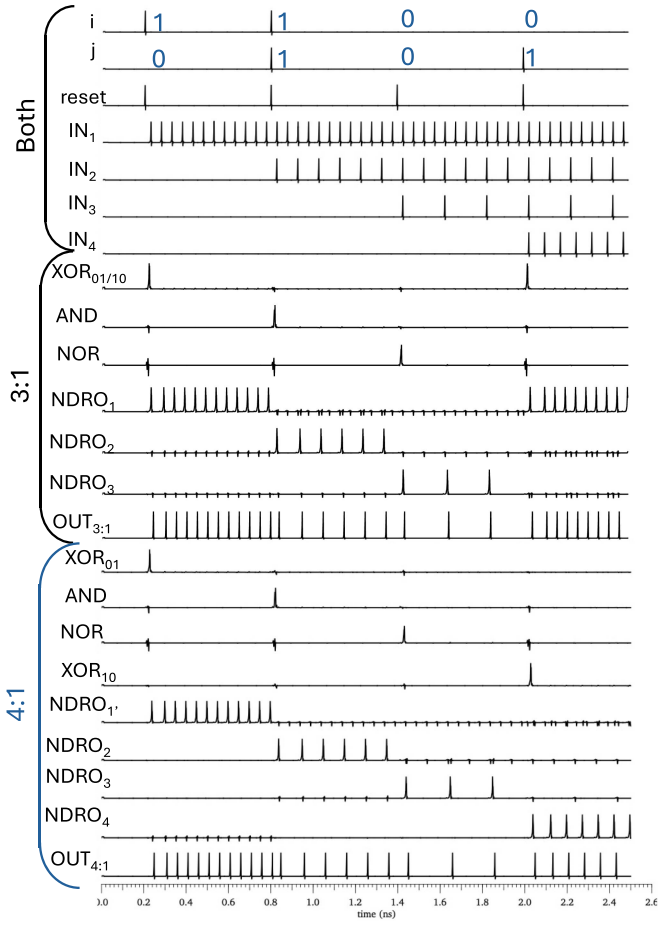

Fig. 4. Waveform of both 3:1 and 4:1 multiplexer. i, j and reset are the control signals.

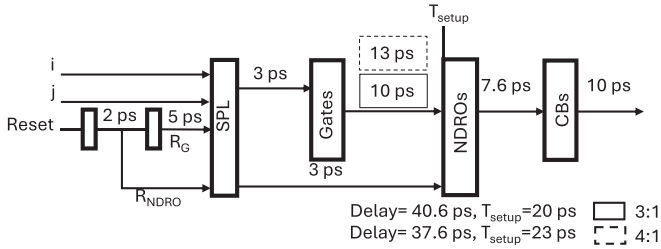

Fig. 5. Timing characterizations of 3:1 and 4:1 MUX. R_G is the delayed reset signal that clocks the gates while R_{NDRO} is the signal that resets the NDROs.

XOR gates that distinguish between 01 and 10. The delay of the multiplexers, measured from arrival of the control signal to the output, is 37.6 ps for the 3:1 MUX and 40.6 ps for the 4:1 MUX. These values are based on the 10 KA/cm^2 MIT Lincoln Laboratory SFQ5ee fabrication process [10].

B. Comparison of Multi-Input Multiplexers

The proposed 3:1 and 4:1 multi-input multiplexers are compared with existing multi-input multiplexers. The different multi-input multiplexers are the individual input control, 2:1 MUX, and proposed multiplexers. As listed in Table I, expressions are provided for key parameters such as the number of

TABLE I
EXPRESSIONS FOR DETERMINING NUMBER OF JJS, DELAY, AND CONTROL SIGNALS FOR ALL MULTIPLEXER TYPES

Parameters	Individual input control (IIC)	2:1	3:1/4:1
N_c	$N_i + 1$	$2\lceil \log_2(N_i) \rceil$	$2\lceil \log_{(3or4)}(N_i) + 1 \rceil$
N_m	N/A	$2^{\lceil \log_2(N_i) \rceil} - 1$	$\sum_{k=1}^{\lceil \log_{(3or4)}(N_i) \rceil} \left\lceil \frac{N_i}{(3or4)^k} \right\rceil$
S_{total}	$Ni - 1$	$2\left(N_m - \log_2(2^{\lceil \log_2(N_i) \rceil})\right)$	$\left(2 \cdot \sum_{k=1}^{\lceil \log_{(3or4)}(N_i) \rceil} \left(\left\lceil \frac{N_i}{(3or4)^k} \right\rceil - 1\right)\right) + (N_m - 1)$
N_{JJ}	$N_i(n_{NDRO}) + (n_s \cdot S_{total}) + n_{CB}(N_i - 1)$	$(n_{MUX} \cdot N_m) + (n_s \cdot S_{total})$	$(n_{MUX} \cdot N_m) + (n_s \cdot S_{total})$
ET_{Setup}	$D_s(N_i - 1)$	$D_s((\lceil N_i/2 \rceil) - 1)$	$D_s(N_i - 1)$
Delay	$ET_{Setup} + D_{NDRO} + D_{CB}(N_i - 1)$	$ET_{Setup} + D_m(\lceil \log_2(N_i) \rceil)$	$ET_{Setup} + D_m(\lceil \log_{3or4}(N_i) \rceil)$

n_s is the number of JJs in a splitter, n_{MUX} is the number of JJs in each MUX type, ET_{Setup} is the external transition speed when changing the input signal with multiple unit multiplexers, D_s is the splitter delay, n_{NDRO} is the number of JJs in an NDRO, and n_{CB} is the number of JJs in a confluence buffer.

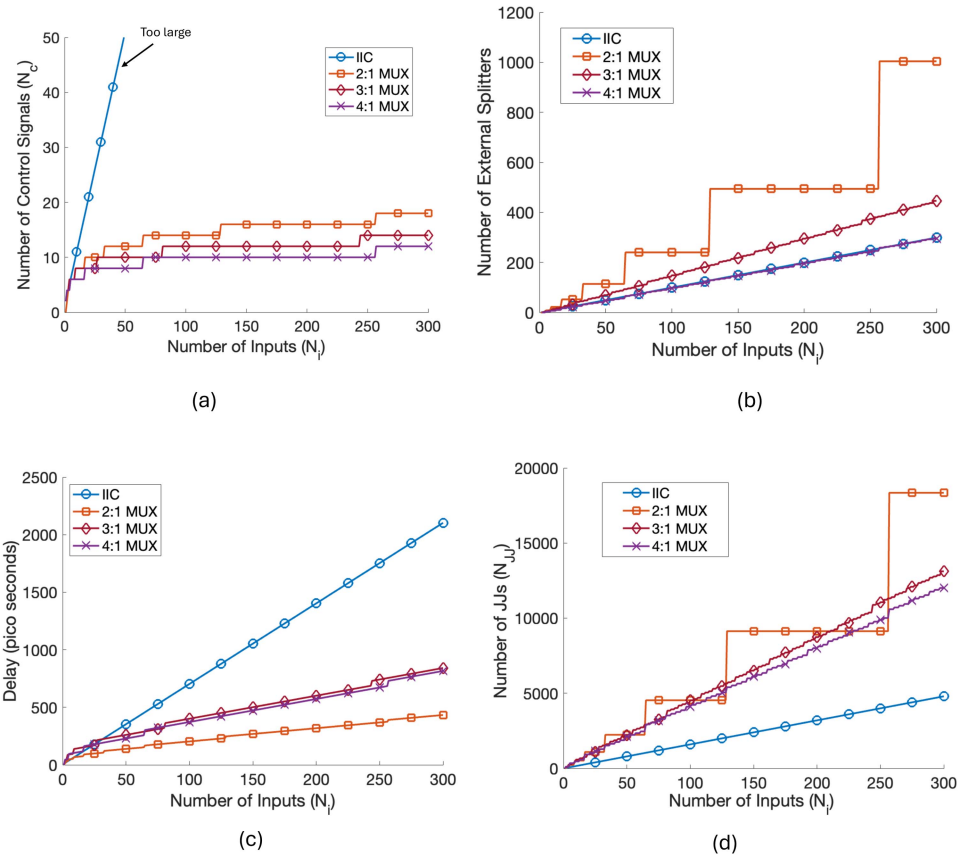

Fig. 6. Performance of individual input control, 2:1 MUX tree, 3:1 MUX tree, and 4:1 MUX tree for N_i multi-input signals. The following criteria are considered: (a) number of control signals, (b) number of external splitters, (c) delay, and (d) number of JJs.

JJs (N_{JJ}), delay, control signals, number of unit multiplexers in a MUX tree (N_m) where a unit is any single multiplexer type, and number of splitters in a MUX tree (S_{total}). These design characteristics apply across different types of multi-input multiplexers, including both individual multiplexers and entire MUX trees and are used to analyze the performance of each multi-input multiplexer, as depicted in Fig. 6. N_{JJ} is used to estimate the power dissipation which increases proportionally with the number of JJs.

For both the 3:1 and 4:1 MUX trees, N_m considers scenarios where the tree structure is either fully balanced (N_i is a power of three or four) or when N_i is not fully balanced. This approach minimizes the number of required unit MUXs when N_i is not a power of three or four, significantly reducing the number of unused ports. The value of N_m is determined by dividing the remaining inputs (N_i) by 3^k or 4^k, where k is the index of each branch, to determine the number of required multiplexers.

For instance, if $N_i = 40$,

- The number of branches for a 2:1, 3:1, and 4:1 MUX tree is $\lceil \log_{2,3,4}(N_i) \rceil$, which equates, respectively, to six, four, and three branches.
- Using the existing 2:1 MUX tree formula for N_m, 40 inputs require 63 unit MUXs, distributed across six branches as $32, 16, 8, 4, 2, 1$.
- In contrast, the proposed expression for a 3:1 MUX tree requires only 22 unit MUXs, distributed as $14, 5, 2, 1$ across four branches.
- Similarly, the proposed expression for a 4:1 MUX tree further reduces the number, requiring only 14 unit MUXs, distributed as $10, 3, 1$ across three branches.

The proposed expressions demonstrate effective management of unit MUXs, particularly when N_i is not a power of three or four. The reduction in the number of unit MUXs directly correlates to fewer splitters and a significant reduction in the number of JJs, making the proposed MUX tree significantly more efficient.

As shown in Fig. 6(a), N_c of both the proposed 3:1 and 4:1 MUX is fewer than the existing multi-input MUX. This lower N_c is due to the global reset signal. Several splitters are required to provide fanout for the control signals. The number of splitters is illustrated in Fig. 6(b), where the splitters used in the 4:1 multiplexer are comparable to the individual bit control which exhibits the lowest S_{total}. This situation occurs since fewer branches are utilized in the 4:1 multiplexer while individual input control exhibits lower S_{total} since only the reset signal requires splitters. As depicted in Fig. 6(c), the 2:1 multiplexer exhibits the lowest delay while the delay of the proposed multiplexer is less than the individual bit control. The 2:1 multiplexer exhibits less delay due to the use of additional control signals N_c. The maximum delay in the 2:1 MUX tree only occurs in the first branch since each branch is controlled by independent set and reset signals. Tradeoffs, therefore, exist between the delay and N_c. As shown in Fig. 6(c), individual bit control has the lowest N_{JJ} while the proposed multiplexers requires fewer JJs than the 2:1 multiplexers. The 2:1 multiplexer require fewer JJs for integer values of $log_2(N_i)$; however, when ceiling functions are applied, the number of JJs grows significantly. Power dissipation is estimated based on N_{JJ}, which, in multi-input multiplexers, increases proportionally with the number of inputs. These tradeoffs highlight the advantages of 3:1 and 4:1 multiplexers in terms of scalability and performance in multi-input large scale multiplexers.

IV. Combined Multi-Input Multiplexer Tree

Different types of multiplexer trees are explored in this section. Topologies are proposed to leverage the advantages of each type of MUX to achieve high performance multi-input multiplexer trees. The 4:1 MUX can substitute for the 2:1 MUX when a higher radix multiplexer is required, as $\lceil \log_4(N_i) \rceil = (\lceil \log_2(N_i) \rceil)/2$. The performance of the 3:1 multiplexer is comparable to the 4:1 MUX. This section, therefore, primarily focuses on the 3:1 MUX and 2:1 MUX.

The number of branch stages per two-input multiplexer and three-input multiplexer is therefore, respectively, n and m, where n is 2 and m is 1 for a twelve-input multiplexer tree, as depicted in Fig. 7. The number of inputs for a combined multiplexer tree is $2^n \times 3^m$, where $n + m$ is the total number of branches within a tree. Maintaining uniformity within a multiplexer tree is essential. Uniformity is maintained if each branch utilizes the same type of multiplexer. By combining both two-input and three-input multiplexers, MUX uniformity at each branch is ensured, as illustrated in Fig. 7. The multiplexer at each branch can either be a three-input MUX or a two-input MUX. This approach minimizes the number of control signals and ensures precise alignment of the pulse arrival times at the input ports across all branches. Placement of a similar MUX at each branch prevents timing skew between the control pulses and the data pulses.

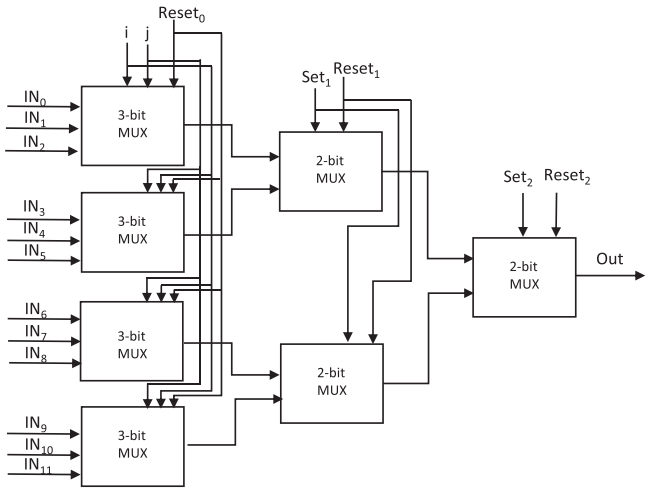

Fig. 7. Twelve-input multiplexer ($2^2 * 3^1$).

TABLE II
Comparison Between Combined Multiplexers Tree and Individual Bit Control With Fewer Inputs

Inputs	Branches $2^n * 3^m$	Combined MUX tree		Individual input control	
		Delay (ps)	N_c	Delay (ps)	N_c
6	$2^1 * 3^1$	52.6	5	42	7
12	$2^2 * 3^1$	85.2	7	72	13
18	$2^1 * 3^2$	105.2	7	102	19
24	$2^3 * 3^1$	103.8	9	137	25
36	$2^2 * 3^2$	135.2	9	192	37
54	$2^1 * 3^3$	183.8	9	287	55

The combined multiplexer tree is comprised of both two-input and three-input multiplexers. *n* and *m* in $2^n * 3^m$ represent, respectively, the number of branches for the 2:1 MUX and 3:1 MUX for each multi-input multiplexer tree.

A twelve-input multiplexer is used here to illustrate a combined multiplexer tree, as depicted in Fig. 7. Seven control signals are required for the twelve-input multiplexer tree, an improvement of 46% over individual input control, as described in Section II-B, where a twelve-input multiplexer requires 13 control signals. The delay of the combined multiplexer tree is also lower than a multiplexer with individual input control with a greater number of inputs. The combined 2:1 and 3:1 multiplexers enhance performance. For example, a 54:1 MUX tree exhibits a 36% reduction in the delay of the combined MUX tree and an 83.6% reduction in the number of control signals with individual input control, as listed in Table II.

V. Conclusion

A three-input and four-input SFQ multiplexer with fewer control signals are described in this paper. The three-input multiplexer utilizes SFQ logic gates to provide set signals to activate the NDROs connected to each input signal. The four-input multiplexer builds on this approach by utilizing redundant

logic in the XOR gates within a three-input multiplexer. This technique is extended to support four inputs while using the same number of control signals as a 3:1 multiplexer. The four-input multiplexer distinguishes between 01 and 10 logic inputs. The delay, number of control signals and number of JJs for the 3:1 and 4:1 multiplexer and existing multi-input multiplexer are compared. The 3:1 and 4:1 multiplexers, the focus of this work, excel in scalability and performance, reducing the number of control signals and delay while utilizing fewer JJs. These features make the 3:1 and 4:1 multiplexers effective for large scale multiplexers in high performance applications. A combined multiplexer tree, comprised of 2:1, 3:1, and 4:1 multiplexers, is also proposed, offering significant improvements in performance.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments and suggestions.

REFERENCES

[1] G. Krylov, T. Jabbari, and E. G. Friedman, *Single Flux Quantum Integrated Circuit Design*, 2nd ed. Cham, Switzerland: Springer, 2024.

[2] 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.

[3] T. Nakagawa et al., "Design and demonstration of high-functionality logic cells for miniaturization of SFQ switches," *Physica C: Supercond. Appl.*, vol. 463–465, pp. 1076–1079, Oct. 2007.

[4] 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.

[5] T. Jabbari, G. Krylov, S. Whiteley, E. Mlinar, J. Kawa, and E. G. Friedman, "Interconnect routing for large-scale RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1102805.

[6] G. Krylov and E. G. Friedman, "Globally asynchronous, locally synchronous clocking and shared interconnect for large-scale SFQ systems," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 3603205.

[7] C. Shawawreh et al., "Effects of adaptive DC biasing on operational margins in ERSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1301606.

[8] H. Myoren, Y. Yoshizawa, and T. Taino, "SFQ multiplexed signal processing circuits for STJ photon detector arrays," *IEEE Trans. Appl. Supercond.*, vol. 19, no. 3, pp. 413–416, Jun. 2009.

[9] M. Dorojevets, C. L. Ayala, N. Yoshikawa, and A. Fujimaki, "16-bit wave-pipelined sparse-tree RSFQ adder," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, Jun. 2013, Art. no. 1700605.

[10] S. K. Tolpygo et al., "Advanced fabrication processes for superconductor electronics: Current status and new developments," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1102513.