Wave Pipelining in DSFQ Circuits

Gleb Krylov, *Student Member, IEEE*, and Eby G. Friedman, *Fellow, IEEE*

***Abstract*—Dynamic SFQ (DSFQ) circuits are a promising form of asynchronous SFQ logic. The operation of DSFQ circuits, however, significantly differs from both CMOS logic and conventional synchronous RSFQ logic. Novel design methodologies are necessary to synthesize DSFQ circuits, while increasing performance and decreasing area. The path balancing process, essential for RSFQ circuits, is less important for DSFQ. Path delay balancing can, however, increase the performance of DSFQ circuits by enabling wave pipelining. In this article, several path balancing approaches for DSFQ circuits are evaluated and compared to equivalent RSFQ circuits. A partial path balancing methodology is described, where path balancing is primarily applied to the inputs of those gates with significant input skew. This methodology provides an effective tradeoff between system performance and area and enables wave pipelining in small to medium scale combinatorial DSFQ circuits without feedback, reducing the period between data waves.**

***Index Terms*—Single flux quantum (SFQ) logic, superconducting integrated circuits, superconducting logic circuits, wave pipelining.**

I. Introduction

In CONVENTIONAL RSFQ circuits [1], most logic gates require a clock signal to operate [2]. If the two inputs of a gate are connected to the gate outputs at different logic depths, an incorrect output will be produced. To prevent this error from occurring, path balancing (PB) elements are inserted into the faster logic paths. These path balancing elements, typically D flip flops (DFF), do not perform a logical operation but delay the signal by a clock cycle to synchronize the data paths.

The number of path balancing DFFs required by a typical circuit is significant, frequently exceeding the number of logic gates [3]. Moreover, these flip flops require both a bias current and a clock signal, further increasing the area and complexity of the clock and power distribution networks [4], [5]. Path balancing is, however, necessary to correctly operate RSFQ logic. Reducing the path balancing overhead during the synthesis process is an important design objective.

Wave pipelining [6] is a well-known technique in high performance circuits, where the pipeline stages operate at frequencies greater than the path delay should permit. The logic path between two registers is composed of two or more regions. Multiple data waves simultaneously propagate between these registers. This approach allows the next data wave to enter the combinatorial logic path before the previous data wave exits the same logic path, increasing the throughput and therefore the performance of the circuit.

Modern RSFQ circuits utilize wave pipelining in a limited fashion due to the large amount of clocked logic. Existing approaches utilize asynchronous AND/OR gates (or alternatively, clocked/resettable C elements) for computation, while handshaking signals propagate the data [7], [8]. The latency and throughput of these wave pipelined circuits are greatly improved as compared to circuits without wave pipelining. To initialize these intermediate asynchronous gates, a reset signal is required. The area overhead to apply wave pipelining utilizing resettable gates is comparable to the overhead of a clock distribution network. Self-resetting asynchronous RSFQ gates are an area efficient alternative; these gates, however, being a variation of a pulse merger [9], exhibit narrow timing margins.

Dynamic SFQ circuits [10] are a natural platform for RSFQ-based wave pipelining. The timing characteristics of DSFQ gates are adjustable over a wide range, and additional logic functions are available (e.g., a majority gate) [11]. Similar to CMOS circuits, DSFQ circuits can greatly benefit from wave pipelining while avoiding the overhead of path balancing and clock distribution networks inherent to standard RSFQ circuits. To date, no path balancing or wave pipelining methodologies has been developed for DSFQ circuits. These methodologies and related issues are the focus of this article.

This article is organized as follows. In Section II, path delay balancing of DSFQ circuits is introduced. In Section III, a partial path balancing methodology to enable wave pipelining in DSFQ circuits while reducing area is described. Finally, Section IV concludes this article.

II. Path Delay Balancing in DSFQ Circuits

Issues related to path delay balancing in DSFQ circuits are discussed in this section. In Section II-A, the concept of delay balancing is introduced. In Section II-B, a procedure to propagate inverters, necessary to correctly operate DSFQ circuits, is described. In Section II-C, the benchmark circuits used to evaluate the proposed techniques are presented, and an RSFQ version of these circuits is characterized. In Section II-A, the operation of DSFQ circuits without delay balancing is described.

A. Delay Balancing

The key characteristic of a DSFQ gate is the retention time—the time between the arrival of the last input pulse and the

Manuscript received 9 June 2021; revised 30 October 2021; accepted 28 November 2021. Date of publication 16 December 2021; date of current version 22 August 2022. This work was supported by the Department of Defense Agency – Intelligence Advanced Research Projects Activity through the U.S. Army Research Office under Contract No. W911NF-17-9-0001. This article was recommended by Associate Editor I. V. Vernik. (*Corresponding author: Gleb Krylov.)*

The authors are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: gleb.krylov@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.2021.3135956.

Digital Object Identifier 10.1109/TASC.2021.3135956

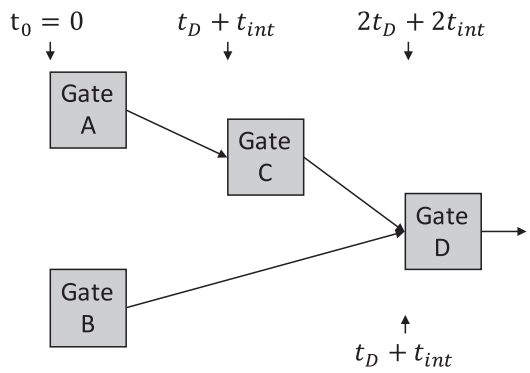

Fig. 1. Delay imbalance within DSFQ logic path. The pulse arrival time for the gates is highlighted. t_D and t_{int} are, respectively, the gate and interconnect delays.

self-reset of the gate. The retention time affects the tolerance of a circuit to input skew—the maximum skew between input signals of the same logic gate. An example of a DSFQ pipeline stage is schematically shown in Fig. 1. A stage is composed of multiple logic gates with different logic depths. The output of the gates at different logic depths (e.g., gate B and gate C in Fig. 1) exhibits a large difference in arrival time, which can exceed the input skew tolerance of gate D. A path balancing (or delay balancing) mechanism is therefore necessary for DSFQ circuits with significant input skews.

Since DSFQ-based combinatorial logic paths are not synchronized, inserting clocked DFFs is not a feasible approach for DSFQ. The delay elements are a means to equalize the arrival time of the signals among different logic paths. A delay element does not perform any function apart from adding delay to the signals along a path. A Josephson transmission line with reduced bias current and increased damping can be used as an area efficient delay element [12]. A set of standard delay elements is necessary for automated path balancing during the logic synthesis process [3]. These delay elements exhibit tunable delays which correspond to the delay of the other elements within a cell library.

DSFQ gates exhibit a tradeoff between the input skew tolerance and system throughput, both affected by the retention time. For increased throughput, it is desirable to decrease the retention time and therefore tolerance to the input skew. The primary source of delay uncertainty—input skew in DSFQ circuits—is caused by the difference in fanout among different input signals.

SFQ signals require an active splitter gate to drive multiple nets [13]. The most commonly used splitter is a binary splitter with cascaded binary splitter trees for fanouts greater than two. Splitter gates introduce significant delay, large as compared to the delay of the interconnect, comparable to the delay of the logic gates [13]. Splitter trees in a standard cell-based design flow use passive transmission lines (PTL) to connect individual splitter cells, which further increases the path delay since a PTL includes both a driver and receiver in each line [14], [15]. These features produce a significant difference in the delay between signals with differing fanout. If these signals are connected to a DSFQ gate, the gate needs to tolerate a large input skew to prevent errors. The effects of splitters on delay balancing is further discussed in Section II-D.

B. Inverter Propagation

An essential step in the synthesis of DSFQ circuits or conversion of RSFQ circuits into DSFQ is inverter propagation. DSFQ logic gates are asynchronous; however, no asynchronous inverter exists for DSFQ logic. This issue is alleviated by placing standard inverters at the boundaries of the combinatorial logic, and replacing standard flip flops with flip flops with complementary outputs (e.g., DFFC [16]) or similar cells.

To propagate inverters, De Morgan's laws of boolean logic are applied. The preceding AND gates are replaced with OR gates and vice versa, and the inverters are moved from the gate outputs to the gate inputs. The majority gates are not changed; only the inverters are moved.

A more complex case is when a gate exhibits multiple fanout, where both the inverted and noninverted output signals are used by other gates. In this case, the gate is duplicated, maintaining the original noninverting paths, while a new inverting path is introduced with appropriate logical modifications. While this redundant approach increases area by adding a duplicate gate, fewer splitters are required as compared to the original circuit.

C. Benchmark Circuits

DSFQ circuits, utilizing different approaches to path balancing, are compared to equivalent RSFQ circuits in the following sections. A subset of ISCAS 1985 benchmark circuits is considered here for this evaluation. These circuits are synthesized and mapped to a generic RSFQ/DSFQ cell library using the open source synthesis tool, ABC [17].

These mapped RSFQ/DSFQ circuits are path balanced. In DSFQ circuits, the inverters are moved to the input boundary of the logic, and integrated with flip flops using DFF cells with complementary outputs. Although these gates are significantly larger than conventional DFFs, the total number of these flip flops does not exceed the number of circuit inputs, which corresponds to approximately 10% to 20% of the gates in a circuit, as listed in Table I. The resulting Verilog netlists are combined with RSFQ and DSFQ SystemVerilog gate models and simulated using Synopsys VCS. The operation of the DSFQ circuits is compared to RSFQ to verify the correctness of the circuit modifications.

For both RSFQ and DSFQ, the number of gates, splitters (in both the logic paths and the clock distribution network), and inverters is determined for each of the benchmark circuits. The gates are assumed to be approximately equal in area. Although this assumption is not accurate, the total number of gates provides a high fidelity approximation of the total area required by the benchmark circuits. The timing characteristics of the gates are also simplified—the gate delay is assumed to be 8 ps for all gates, flip flops, and splitters with any fanout; while the interconnect delay is 6 ps between any gates. The characteristics of the RSFQ benchmark circuits are listed in Table I.

Note that the overhead of the path balancing process in conventional RSFQ circuits is significant, often exceeding the

TABLE I
CHARACTERISTICS OF ISCAS1985 BENCHMARK CIRCUITS MAPPED TO RSFQ CELL LIBRARY WITH PATH BALANCING

Benchmark circuit	Gates	Logic levels	Inputs	PB DFFs	Splitters	Clock splitters	Inverters	Latency (ps)	Total gates
c17	8	5	5	6	3	7	2	100	26
c432	369	27	36	634	140	368	60	540	1,571
c880	434	22	60	743	155	433	76	440	1,841
c1335	680	22	41	1,284	251	679	72	440	2,966
c2670	1,120	22	233	1,263	300	1,119	163	440	3,965

TABLE II
CHARACTERISTICS OF ISCAS1985 BENCHMARK CIRCUITS MAPPED TO A DSFQ CELL LIBRARY WITH AND WITHOUT SPLITTERS

	Without splitters				With splitters				
Benchmark circuit	Gates	Logic levels	Maximum latency, ps	Maximum input skew, ps	Gates	Splitters	Logic levels	Maximum latency, ps	Maximum input skew, ps
c17	6	4	56	14	6	2	6	84	42
c432	309	26	364	126	309	128	36	504	350
c880	358	22	308	168	358	162	32	448	378
c1335	608	20	280	168	608	244	30	420	364
c2670	957	18	252	168	957	417	25	350	196

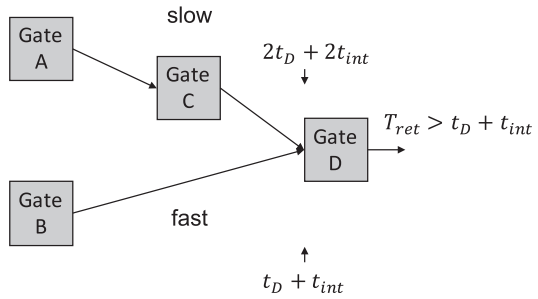

Fig. 2. Delay imbalance between DSFQ logic paths. The faster and slower signal paths are highlighted. The circuit operates correctly if the retention time is sufficient.

area occupied by the logic gates. Clock splitters are another significant contributor to the overall circuit area. The latency of the RSFQ circuits is proportional to the logic depth and clock period. For the benchmark circuits listed in Table I, the clock period is assumed to be 20 ps (50 GHz).

D. DSFQ Circuits Without Path Balancing

Unlike RSFQ circuits, DSFQ circuits, under specific conditions, produce the correct output without path balancing. If the two inputs of a DSFQ AND/majority gate are on different logic levels, as shown in Fig. 2, the faster signal path (shorter logic depth) changes the state of the gate, which persists within the gate during the retention time. If the slower signal path (deeper logic depth) arrives during this time, a correct output is produced by the gate. In the case of a DSFQ OR gate, the correct output is produced immediately upon arrival of the first input pulse, while the gate remains insensitive to the second input pulse during the retention time. Therefore, if all of the gates within a DSFQ circuit exhibit a sufficiently long retention time, path balancing is not required to maintain correct operation.

A retention time that is larger than necessary does not affect the operation of a circuit if the following datum is not expected during this time. A longer retention time increases the robustness of the circuits to variations in the input skew. The retention time also exhibits a negligible effect on the area of DSFQ circuits. Due to these properties, it is convenient to set the retention time equal to the maximum input skew for all DSFQ gates.

This approach is evaluated using benchmark circuits, as described in Section II-C. The properties of the DSFQ circuits without path balancing are listed in Table II. Observe that inclusion of the splitters greatly increases the maximum input skew of a circuit (sometimes by more than twofold). DSFQ circuit design and analysis which does not consider the splitters will produce incorrect results. The splitters should therefore not be neglected when evaluating the retention time or during the path balancing process.

As compared to the RSFQ circuits listed in Table I, DSFQ circuits require significantly less area and generally exhibit faster delay. Although DSFQ circuits can operate without path balancing, this approach results in a large input skew which necessitates a long retention time. The gates cannot accept another datum during this time, limiting the throughput. This limitation can be circumvented by using wave pipelining, as discussed in the following section.

III. PARTIAL PATH BALANCING AND WAVE PIPELINING

Approaches to enable wave pipelining in DSFQ circuits are discussed in this section. In Section III-A, necessary conditions for wave pipelining are discussed. In Section III-B, full path balancing (for all gates and splitters) is explored as an approach to wave pipelining. Partial path balancing is introduced in Section III-C.

A. Necessary Conditions for Wave Pipelining

To enable wave pipelining in DSFQ circuits, any uncertainty in the arrival time of the data signal for each gate should be managed. The primary sources of timing uncertainty in DSFQ gates are the difference in the logic depth of the input signals,

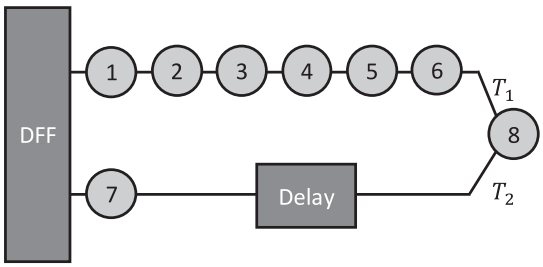

Fig. 3. DSFQ pipelined signal path supporting multiple data waves.

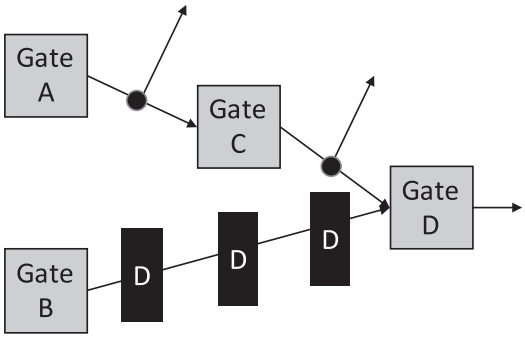

Fig. 4. Full path balancing in DSFQ circuits. The difference in path delays related to the gates and splitters (shown as circles) in the slower (top) path is balanced in the faster (bottom) path by delay elements D.

and the difference in the number and fanout of the splitters along these input paths. If these sources of uncertainty are mitigated, the difference in the interconnect length remains a source of input timing skew. This difference is typically small for a combinatorial circuit, as the gates are typically located in close proximity and the interconnect delays are small due to the high propagation velocity of the SFQ pulses within PTLs.

In wave pipelined DSFQ circuits, the following data wave can be initiated if any collisions with the existing data waves along the signal path are avoided. A new data wave is launched into the signal path when the data wave already present in the pipeline is located far from the input (for example, near gates 5 or 6 in Fig. 3). Any collision with the existing data wave would be avoided. The minimum time (or data skew) before the following data wave can be applied to the pipelined signal path is the largest retention time of all of the gates along a signal path. The objective is therefore to reduce the maximum input skew of all of the DSFQ gates within a circuit.

B. Full Path Balancing

A straightforward approach to reduce input skew is to utilize RSFQ-like path balancing. Any difference in the logic depth of the gate inputs is compensated by adding delay elements, with the delay equal to a typical gate delay or tuned to a specific delay within the slower path, as shown in Fig. 4. In addition, the delay of the splitters, as discussed in Section II-D, can also be compensated by adding delay elements. Although some of the successive delay elements can be merged, this process can produce collisions between the data or limit overall throughput.

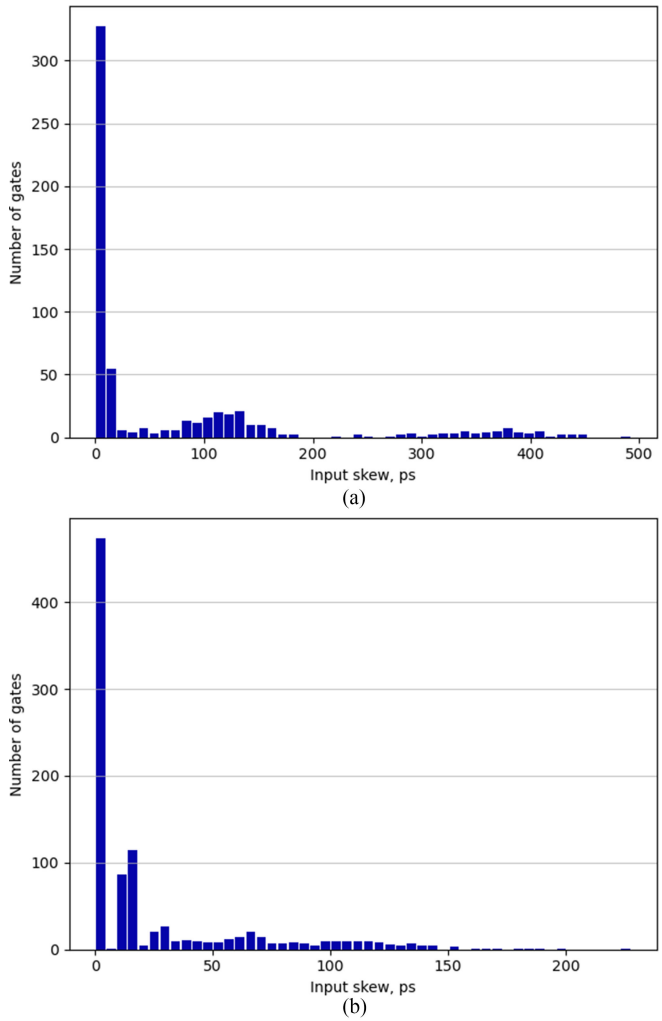

Fig. 5. Statistical distribution of input skews for DSFQ ISCAS 1985 benchmark circuits. (a) c1355 and (b) c2670.

This approach is applied to the benchmark circuits discussed in Section II-C. The results are listed in Table III. Note that full path balancing greatly reduces the minimum data skew, limiting any differences in the input arrival time to the difference in interconnect delay. As each logic level is a separate pipeline stage, this approach produces the ultimate throughput of a DSFQ circuit. This approach, however, exhibits significant area overhead as compared to conventional RSFQ circuits. This significant area overhead makes full path balancing impractical in DSFQ circuits. In addition, the benefits of higher retention times in DSFQ circuits are not exploited.

C. Partial Path Balancing

Each gate in a pipeline stage has a different input skew depending upon the preceding gates and splitters. The majority of gates exhibits zero or small input skew. Certain gates, however, exhibit significant or close to maximum skew. A statistical distribution of the input skews is shown in Fig. 5 for two different benchmark circuits. Variations in gate and interconnect delay are modeled

TABLE III
CHARACTERISTICS OF ISCAS 1985 BENCHMARK CIRCUITS MAPPED TO A DSFQ LIBRARY WITH FULL PATH BALANCING

Benchmark circuit	Gates	Logic levels	Splitters	PB elements (logic)	PB elements (splitters)	Maximum latency, ps	Maximum input skew*, ps	Total gates	Total gates (compared to RSFQ)	Total gates (compared to unbalanced DSFQ)
c17	6	6	2	3	5	76	0	16	-40%	+100%
c432	309	36	128	2,392	482	496	0	3,311	+111%	+658%
c880	358	32	162	3,001	743	440	0	4,246	+132%	+717%
c1335	608	30	244	2,744	1,092	412	0	4,688	+58%	+450%
c2670	957	25	417	4,258	670	342	0	6,302	+59%	+359%

* The input skew is due to the gates and splitters; any difference in interconnect delay also contributes to the input skew.

here as a Gaussian distribution with a mean of zero and a standard deviation of 2 ps per gate and interconnect stage.

In a DSFQ circuit, delay balancing between the input paths of gates with near zero or small skew does not increase performance while expending area. Applying delay balancing to the input paths of gates with significant input skew, however, reduces the maximum input skew in a circuit while increasing the throughput. Partial delay balancing is applied only to the input paths of these gates. The distribution of input skews is initially evaluated, and the target throughput is determined. The target throughput is the number of data waves simultaneously propagating within a logic path times the effective clock period. This number of simultaneously propagating data waves determines the target data skew T_{DATA}—the period of time between two consecutive data waves. The target data skew T_{DATA} sets the limit for the input skew within a circuit, as follows:

$$T_{\text{DATA}} \geq T_{\text{ret}} \geq T_{\text{skew}}^{\text{max}} \tag{1}$$

where $T_{\text{skew}}^{\text{max}}$ is the maximum input skew, and T_{ret} is the gate retention time. By reducing the input skew for a small number of outlier gates, the data skew T_{DATA} can also be decreased, increasing overall performance.

Based on the target T_{DATA}, those gates with an input skew $T_{\text{skew}} \geq T_{\text{DATA}}$ are delay balanced—delay elements are inserted into the faster path(s) to reduce the gate input skew. To accommodate multiple data waves, a specific number of delay elements are inserted to avoid data collisions. For N data waves, $N-1$ delay elements with a delay time $T_d = T_{\text{skew}}/N$ are inserted into the faster paths. A single datum is also stored in a DSFQ gate for $T_{\text{ret}} = T_d$. After completion of this process, the gate exhibits a small input skew and does not limit the overall data skew.

C. Case Study: Consider the DSFQ version of the ISCAS1985 benchmark circuit c880 with splitters, as listed in Table II. This circuit can operate without path balancing for a data skew $T_{\text{DATA}} \geq 408$ ps. A statistical distribution of the gate input skews for this circuit is shown in Fig. 6(a).

Based on this distribution and the target performance, T_{DATA} is arbitrarily set to 233 ps. Gates with the largest input skew are identified, and path balancing is applied. These large input skews are eliminated from the circuit, and the maximum input skew is T_{DATA}. The distribution of input skews in the modified circuit is shown in Fig. 6(b).

The resulting partially path balanced circuit can now operate at $T_{\text{DATA}} = 233$ ps. In this case, 160 standard delay elements (each element exhibits a delay of 8 ps and an interconnect segment exhibits a delay of 6 ps) are inserted—a significant improvement as compared to both RSFQ and DSFQ circuits

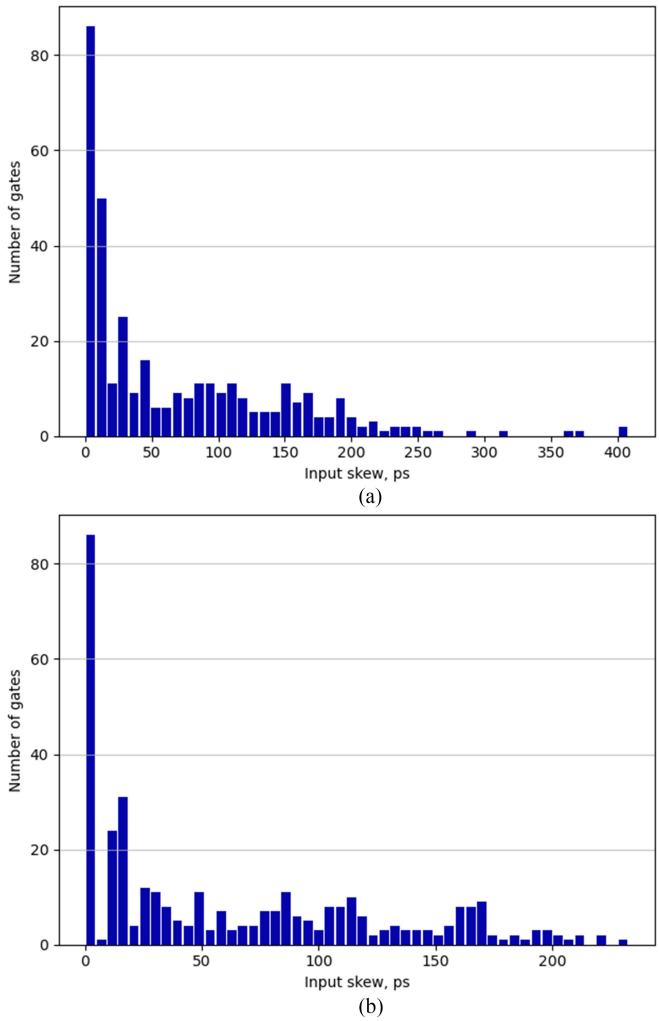

Fig. 6. Statistical distribution of input skews for DSFQ ISCAS1985 benchmark circuit c880. (a) Before partial path balancing and (b) after partial path balancing.

with full path balancing. Moreover, the total additional delay is approximately 2.2 ns, where only 17 discrete delay elements are required to avoid collisions between data waves.

Partial delay balancing increases the performance of DSFQ circuits by enabling wave pipelining, while requiring low area overhead as compared to RSFQ and DSFQ circuits with full path balancing. These results also highlight the need for efficient compact delay elements. Only a few discrete delay elements are required for path balancing. The remaining delay elements

could utilize a few elements with a large delay. To date, no compact SFQ circuits capable of producing a delay on the order of ~ 100 ps exist. Efficient delay elements would greatly reduce the area required for path balancing in DSFQ circuits, further incentivizing this SFQ circuit topology.

The phenomenon of delay dispersion is not considered in this article in detail. The interconnect delays are assumed here to primarily consist of driver and receiver gate delays, and variations in interconnect length are assumed to be small relative to the signal propagation speed in a PTL. In large scale DSFQ circuits with significant pipeline depth or feedback paths, delay dispersion due to variations in gate delays and differences in interconnect length complicates the insertion of delay buffers during logic synthesis. In these systems, inserted delays need to be adjusted postplacement, e.g., the bias currents of delay elements can be adjusted to account for difference in interconnect delays. Delay dispersion due to variations in gate delays in a fabricated circuit is a more complex problem common in asynchronous networks, and can require insertion of handshaking circuitry, purposeful reduction of pipeline depth via synchronous stages, or other techniques. Development of these techniques is an important next step to enable efficient large scale DSFQ circuits.

IV. Conclusion

DSFQ circuits can greatly reduce the area and complexity of clock and bias distribution networks in large scale superconductive circuits. The path balancing process, essential for RSFQ circuits, is less important for DSFQ circuits. Path delay balancing can, however, be used to increase the performance of DSFQ circuits by enabling wave pipelining. In this article, different path balancing approaches for DSFQ circuits are evaluated and compared to equivalent RSFQ circuits. Full delay balancing produces the highest throughput by enabling wave pipelining while minimizing the data skew. This approach, however, exhibits an impractical area overhead, and does not exploit the improved timing characteristics of DSFQ circuits. A partial path balancing methodology is described, where path balancing is primarily applied to the input paths of gates with significant input skew. This methodology enables wave pipelining of DSFQ circuits, reducing the time between data waves while requiring relatively small area, thereby providing an effective tradeoff between area and throughput.

Acknowledgment

The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

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

[2] G. Krylov and E. G. Friedman, *Single Flux Quantum Integrated Circuit Design*. Berlin, Germany: Springer, 2022.

[3] G. Pasandi and M. Pedram, "PBMap: A path balancing technology mapping algorithm for single flux quantum logic circuits," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 4, Nov. 2019, Art. no. 1300114.

[4] G. Krylov and E. G. Friedman, "Design methodology for distributed large scale ERSFQ bias networks," *IEEE Trans. Very Large Scale Integration Syst.*, vol. 28, no. 11, pp. 2438–2447, Nov. 2020.

[5] G. Krylov and E. G. Friedman, "Partitioning RSFQ circuits for current recycling," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1301706.

[6] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-pipelining: A tutorial and research survey," *IEEE Trans. Very Large Scale Integration Syst.*, vol. 6, no. 3, pp. 464–474, Sep. 1998.

[7] M. Dorojevets, C. L. Ayala, and A. K. Kasperek, "Data-flow microarchitecture for wide datapath RSFQ processors: Design study," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 787–791, Jun. 2011.

[8] 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.

[9] S. S. Meher *et al.*, "Superconductor standard cell library for advanced EDA design," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1300807.

[10] S. V. Rylov, "Clockless dynamic SFQ and gate with high input skew tolerance," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300805.

[11] G. Krylov and E. G. Friedman, "Asynchronous dynamic single flux quantum majority gates," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 5, Aug. 2020, Art. no. 1300907.

[12] B. Dimov *et al.*, "Tuning of the RSFQ gate speed by different Stewart-McCumber parameters of the Josephson junctions," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pp. 284–287, Jun. 2005.

[13] 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.

[14] 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.

[15] T. Jabbari, G. Krylov, S. Whiteley, J. Kawa, and E. G. Friedman, "Repeater insertion in SFQ interconnect," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 8, Dec. 2020, Art. no. 5400508.

[16] Stony Brook University. SUNY RSFQ Cell Library. [Online]. Available: http://www.physics.sunysb.edu/physics/rsfq/lib/contents.html

[17] ABC: A System for Sequential Synthesis and Verification. Berkeley Logic Synthesis and Verification Group. [Online]. Available: http://www.eecs.berkeley.edu/textasciitildealanmi/abc/