High-Capacity DFF Cells for Unary Computing Using Single-Flux Quantum Circuits

Zeyu Han, Nobuyuki Yoshikawa, *Fellow, IEEE*, and Yuki Yamanashi, *Member, IEEE*

***Abstract*—Due to their high-speed operation and low power consumption, single-flux quantum (SFQ) circuits are promising technologies for integrated circuits. However, current superconducting fabrication limitations impose strict area constraints, challenging the construction of large-scale systems. Unary computing (UC) encodes numerical values by concentrating all "1"s at the beginning of a bit sequence. This approach enables arithmetic operations with fewer logic gates, thereby improving the area efficiency. In this article, we propose a novel high-capacity D flip-flop (HC-DFF) cell for UC-based SFQ circuits that achieves high-density data registration while preserving UC encoding. The HC-DFF employs a cascaded storage loop structure in which each loop sequentially stores one flux quantum, avoiding the margin deterioration associated with storing multiple flux quanta in a single loop. Simulation shows that the proposed HC-DFF achieves a bias margin of –20.8% to +32.7% and supports read/write operations above 100 GHz. We fabricated the HC-DFF using a 10 kA/cm^2 Nb four-layer superconducting process and verified its operation through low-speed measurements. The HC-DFF offers enhanced scalability and stability while reducing the number of required Josephson junctions by 83.3% per bit compared to conventional DFF-based shift registers for saving UC signals. For short unary sequences of less than 31 bits, it retains an advantage over storing binary data with equivalent precision using conventional DFF-based shift registers.**

***Index Terms*—High-capacity D flip-flop (HC-DFF), single-flux-quantum circuit, unary computing (UC).**

I. Introduction

In the post-Moore era, single-flux quantum (SFQ) circuits have emerged as promising alternatives to semiconductor integrated circuits because of their high-speed operation and low power consumption [1]. These attributes have spurred the development of various SFQ-based high-performance computing units [2], [3], microprocessors [4], [5], [6], [7], [8], and neural network accelerators [9], [10], [11]. However, designing large-scale systems based on SFQ circuits remains challenging due to the strict area constraints imposed by current superconducting manufacturing limitations [12].

Various nonconventional computing paradigms have been proposed to address these area density challenges, such as stochastic computing [13], [14], [15] and temporal computing [16], [17]. In particular, unary computing (UC) [18] encodes values by concentrating all "1"s at the beginning of the bit sequence. By counting the number of "1"s in a bit sequence, UC can be regarded as a form of stochastic computing. In contrast, by encoding the timing of the final "1," UC can be interpreted as a form of temporal computing. Consequently, arithmetic operations, such as comparison, addition, and subtraction, can be implemented with fewer logic gates, reducing both area and energy consumption [19]. Although increasing the length of the unary sequence improves computational precision, it also increases computation time. Therefore, UC is typically used for low-precision operations (e.g., unary sequences shorter than 256 bits, equivalent to 8-bit binary precision). For instance, the unary median filter with a sequence length of 256 bits applied in image processing [20] and 2–8-bit unary matrix multiplication circuits in neural network accelerators [21], [22] have demonstrated higher energy efficiency than their conventional binary circuits.

Nevertheless, efficiently storing UC signals while preserving the concentration of "1"s at the beginning of the bit sequence during successive operations is challenging in UC-based SFQ circuits. Although a conventional D flip-flop (DFF)-based shift register (SR) can achieve this, it requires one DFF per bit, severely limiting storage efficiency. High-capacity destructive readout (HC-DRO) cells [23] offer a viable alternative by increasing the inductance in the storage loop to accommodate multiple flux quanta. However, due to inherent margin limitations, a single storage loop can only store up to three flux quanta. Despite this limitation, register files based on this HC-DRO have demonstrated high area and energy efficiency when integrated into processor designs [24].

In this article, we propose a novel high-capacity D flip-flop (HC-DFF) that enhances the storage capacity of HC-DRO while maintaining stability. The proposed HC-DFF employs a cascaded storage loop architecture in which each loop sequentially stores one flux quantum, avoiding the limitations of storing multiple flux quanta in a single loop. Low-speed measurements confirm the correct operation of the HC-DFF, and its performance is compared with alternative storage methods.

This article is organized as follows. Section II reviews key UC arithmetic circuits, motivates the need for an HC-DFF within

Received 18 April 2025; revised 22 July 2025; accepted 21 August 2025. Date of publication 25 August 2025; date of current version 8 September 2025. This work was supported in part by JSPS KAKENHI under Grant JP24KJ1148 and Grant JP25K01284 and in part by the Support Center for Advanced Telecommunications Technology Research. *(Corresponding author: Yuki Yamanashi.)*

Zeyu Han is with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan (e-mail: han-zeyu-mb@ynu.jp).

Nobuyuki Yoshikawa and Yuki Yamanashi are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan, also with the Institute of Advanced Sciences, Yokohama National University, Yokohama 240-8501, Japan, and also with the Institute for Multidisciplinary Sciences, Yokohama National University, Yokohama 240-8501, Japan (e-mail: nyoshi@ynu.ac.jp; yamanashi-yuki-kr@ynu.ac.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASC.2025.3602472.

Digital Object Identifier 10.1109/TASC.2025.3602472

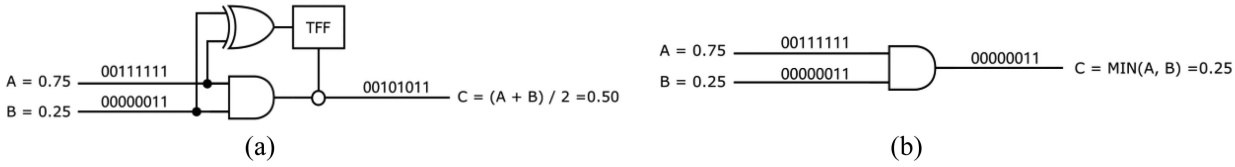

Fig. 1. Circuit diagram for (a) scaling adder and (b) minimum using UC, where the white circle represents the confluence buffer.

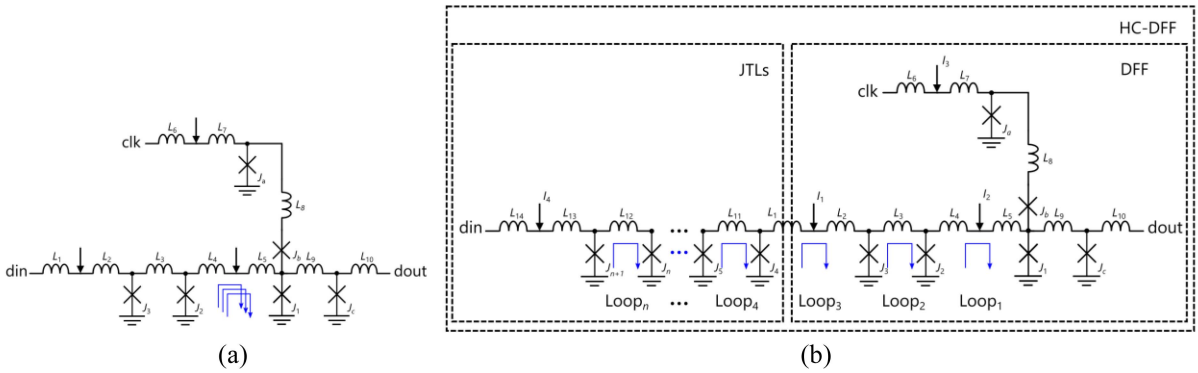

Fig. 2. (a) HC-DRO [23] circuit with a single storage loop capable of storing up to three flux quanta. (b) Proposed HC-DFF with cascaded multiple storage loops that can sequentially store flux quanta. Storage capacity scales with the number of cascaded JTL stages. Parameters in (b) are as follows: $I_1 = 138.89$ μA, $I_2 = 74.76$ μA, $I_3 = 203.25$ μA, $I_4 = 300.48$ μA, $L_1 = L_6 = L_{14} = 0.84$ pH, $L_2 = 4.48$ pH, $L_3 = 8.39$ pH, $L_4 = 1.08$ pH, $L_5 = 7.51$ pH, $L_7 = 1.81$ pH, $L_8 = 2.59$ pH, $L_9 = 5.24$ pH, $L_{10} = 1.96$ pH, $L_{11} = L_{12} = 4.78$ pH, $L_{13} = 2.47$ pH, $J_1 = J_4 = J_5 = \ldots = J_n = J_{n+1} = 213$ μA, $J_2 = 121$ μA, $J_3 = 130$ μA, $J_a = 164$ μA, $J_b = 202$ μA, $J_c = 207$ μA.

UC systems, and explains the HC-DFF operating principle. Section III presents low-speed experimental verification of the proposed HC-DFF and compares its performance with alternative storage methods. Finally, Section IV concludes the article and outlines directions for future research.

II. HC-DFF for Unary Computing

UC encodes values by concentrating all "1"s at the beginning of a bit sequence and representing numerical values by counting these "1"s. This method enables deterministic stochastic computing [25], [26]. For example, the scaling adder in Fig. 1(a) uses only three logic gates, where the number of "1"s in the output sequence represents the sum. UC can also encode values based on the timing of the final "1," which makes it suitable for temporal computing and dynamic programming algorithms [27]. Fig. 1(b) shows that a single AND gate can compare two input values to yield the minimum value, whereas OR gates can obtain the maximum value [28].

UC faces the challenge of keeping the output bit sequence in proper UC encoding when operations are performed in succession. For example, in the adder output shown in Fig. 1(a), although the number of "1"s represents the intended value, they are no longer concentrated at the beginning of the sequence. Therefore, re-encoding the output after each operation is essential. In addition, storing UC data only requires that the number of "1"s is correct and that all "1"s precede any "0"s. To address these challenges, we propose an HC-DFF that efficiently stores and re-encodes UC data during write and read operations.

Fig. 2(a) illustrates a DFF with increased storage capacity proposed in [23] by increasing the inductance in the storage loop ($J_2 - L_4 - L_5 - J_1$), which allows it to store multiple flux quanta. However, storing multiple flux quanta in a single loop narrows the operating margin and limits the maximum capacity to three. Because 3-bit storage is insufficient for UC systems, we propose a novel HC-DFF inspired by the back-pressure Josephson transmission line (JTL) [29], [30]. As shown in Fig. 2(b), our design uses cascaded storage loops to store flux quanta instead of relying on one enlarged loop. The parameters in Fig. 2(b) were determined using the parameter automation tool MarginX [31] to achieve the widest bias margin. The magnetic flux input from din is sequentially stored in Loop$_1$ through Loop$_n$ so that the storage capacity scales linearly with the number of cascaded JTL stages. During readout, the stored flux quanta are sequentially output with the clk signal.

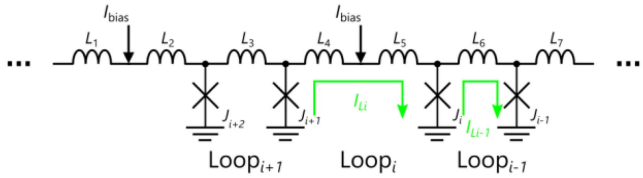

Fig. 3. Circuit diagram of Loop$_i$ with bias current I_{bias}, illustrating the circulating currents I_{Li} and I_{Li-1} induced by stored magnetic flux quanta in Loop$_i$ and Loop$_{i\text{-}1}$.

The JTL preceding the HC-DFF serves both signal transmission and storage functions. We define Loop$_i$ as an arbitrary loop within the cascaded JTL architecture, as depicted in Fig. 3. When no flux quantum is stored in Loop$_{i-1}$ ($J_i - L_6 - J_{i-1}$), the current flowing through J_i is

$$I_{J_i} = I_{\text{bias}}/2 < I_c \tag{1}$$

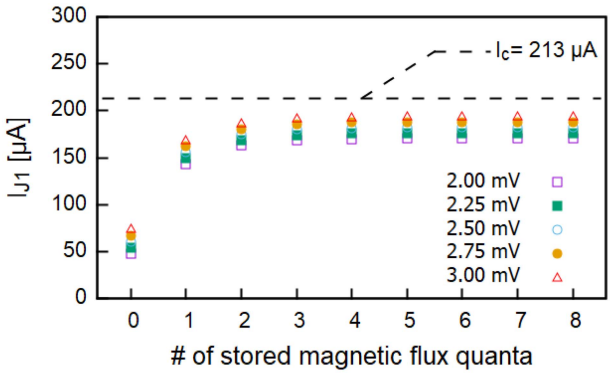

Fig. 4. Relationship between stored flux quanta and current flowing through J_1 at different bias voltages, with the black dashed line indicating the critical current value of 213 μA for J_1.

where I_{bias} is the bias current and I_c is the critical current of J_i. An input from the left then causes J_{i+1} to switch, which increases the current through J_i to

$$I_{J_i} = I_{\text{bias}}/2 + I_{Li} > I_c \tag{2}$$

where I_{Li} is the circulating current induced by J_{i+1} switching. Once I_{J_i} exceeds the critical current of J_i, the junction switches as in normal JTL, and the input is transmitted rather than stored in Loop$_i$ ($J_{i+1} - L_6 - L_4 - J_i$). When a flux quantum is stored in Loop$_{i-1}$, the current flowing through J_i becomes

$$I_{J_i} = I_{\text{bias}}/2 - I_{Li-1} < I_c \tag{3}$$

where I_{Li-1} is the circulating current induced by stored magnetic flux quanta in Loop$_{i-1}$. After a signal from the left causes J_{i+1} to switch, the current through J_i increases to

$$I_{J_i} = I_{\text{bias}}/2 + I_{Li} - I_{Li-1} \tag{4}$$

and since the parameters of each loop in the JTL are similar (i.e., $I_{Li} \approx I_{Li-1}$), we have

$$I_{J_i} \approx I_{\text{bias}}/2 < I_c. \tag{5}$$

Consequently, the critical current is not exceeded, and the flux quanta are stored in Loop$_i$.

In Fig. 2(b), storing a flux quantum in a left-side loop of the cascaded storage loops can affect the currents in the right-side loops. For example, the current in Loop$_1$ increases when a flux quantum is stored in Loop$_2$. However, as the stages between loops increase, this effect decays as a power law. Fig. 4 shows the change in the current through J_1 as the number of stored flux quanta increases, with the black dashed line indicating the critical current of 213 μA for J_1. When more than six flux quanta are stored, the increase in current through J_1 is less than 0.1 μA, ensuring that the critical current is not exceeded. Therefore, the capacity of this HC-DFF is determined solely by the number of JTL stages in the cascade. Except for Loop$_1$, Loop$_2$, and part of Loop$_3$, all other loops use the same parameters as the JTLs in our modified cell library [32], [33] for the AIST 10 kA/cm^2 Nb high-speed standard process (HSTP) [34], [35]. Because the current through J_1 is almost constant when more than six flux quanta are stored, adding additional cascaded loops does not affect the circuit margin. Consequently, the HC-DFF can scale to arbitrarily long sequences, limited only by available chip area. Furthermore, since UC targets only low-precision operations, this scalability does not incur excessive area overhead. This design optimizes circuit area by enabling conventional SFQ transmission circuits to serve as storage elements.

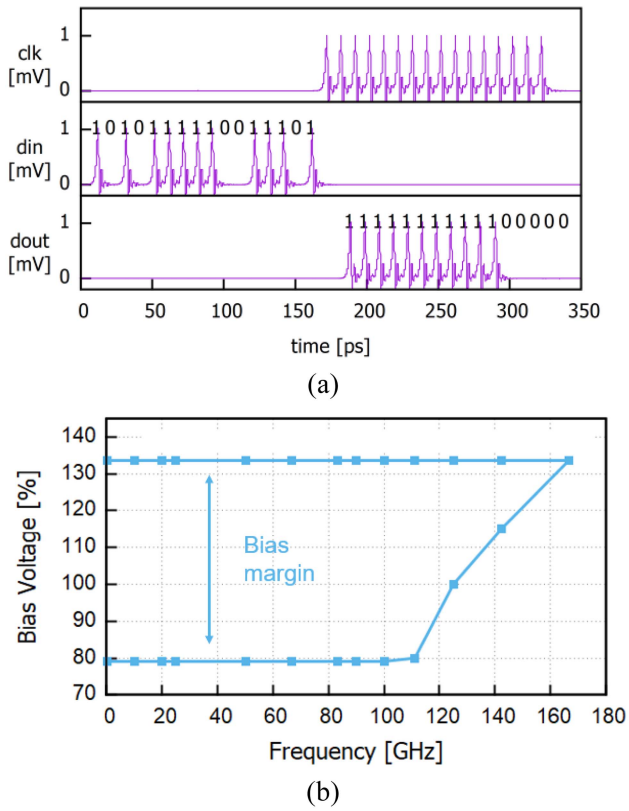

Fig. 5. (a) Simulation results of 16-bit HC-DFF, as shown in Fig. 2(b), at 100 GHz. (b) Simulation results of the normalized bias voltage margin. The designed bias voltage is 2.5 mV.

Fig. 5(a) shows the simulation results obtained with JoSIM [36], demonstrating the operation of the proposed 16-bit HC-DFF at 100 GHz. The design follows a cell-based methodology [37] and uses our modified cell library, adapted for HSTP. The simulation confirms that the 16-bit input data are successfully stored in the HC-DFF and that the "1"s shift to the beginning of the bit sequence upon readout. The HC-DFF achieves a bias margin of –20.8% to +32.7% . Fig. 5(b) shows the maximum read/write frequency of HC-DFF, which is limited by the JTL transmission speed in the simulation. In addition, the minimum interval between a write and the subsequent read is determined by the total delay of the cascaded JTLs, ensuring that the final written data have enough time to be stored in loop$_1$ in the worst case.

III. Experimental Details and Evaluation

We designed and implemented a 16-bit HC-DFF using the HSTP. A microphotograph of the fabricated 16-bit HC-DFF is presented in Fig. 6. The circuits were tested at 4.2 K using

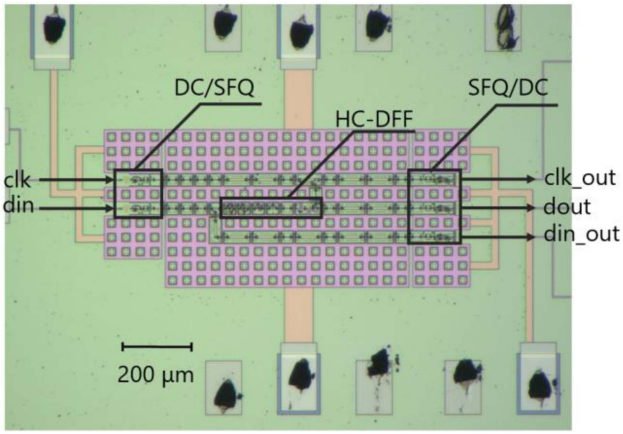

Fig. 6. Microphotograph of the 16-bit HC-DFF.

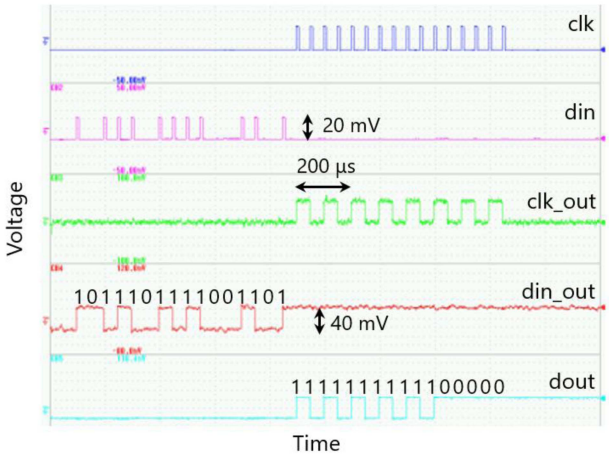

Fig. 7. Low-speed measurement result of 16-bit HC-DFF at 20 kHz.

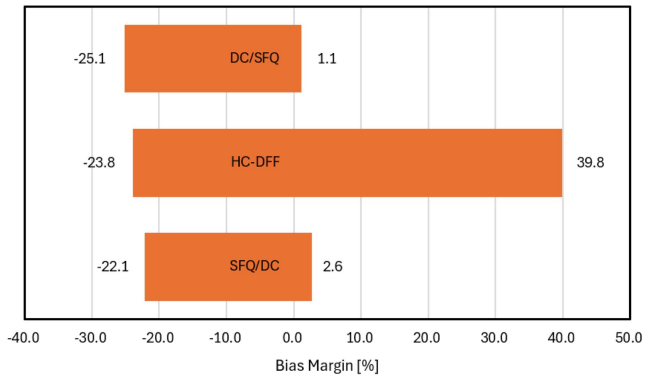

Fig. 8. Normalized bias margin obtained from low-speed measurements. The designed bias voltage is 2.5 mV.

a liquid helium bath for cooling. The device under test was evaluated using various input patterns. Fig. 7 shows an example of a waveform at 20 kHz, demonstrating that the 16-bit data are correctly stored in the HC-DFF when input sequentially and can be read out by the clk signal.

Fig. 8 shows the normalized bias margin from low-speed measurements at a bias voltage of 2.5 mV. Experimentally, the

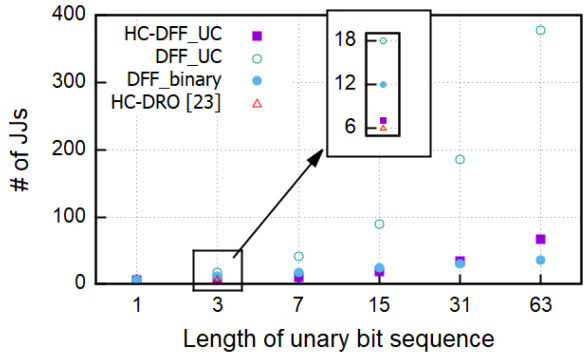

Fig. 9. Relationship between bit sequence length and required JJs. HC-DFF_UC denotes storing UC data using the proposed HC-DFF, while DFF_UC refers to storing UC data with a conventional DFF-based SRs. DFF_binary stores binary data with precision matched to the unary sequence (e.g., at 15 bits on the horizontal axis, and DFF_binary shows the number of JJs required for storing 4-bit binary data in a conventional DFF-based SR). HC-DRO [23] corresponds to the HC-DRO in [23], which can store up to three UC data. In DFF_UC and DFF_binary, only the JJs within the DFF are considered, excluding those in the data transmission line.

measured margin of HC-DFF exceeds the simulated range of [–20.8%, 32.7%]. This difference arises because our simulations flagged any deviation of internal flux storage as an error, such as a quantum residing concurrently in two loops, even though the circuit still produces correct outputs. These internal state differences were not detected in the test, and if they are ignored in the simulation, the simulated margin widens to [–47.1%, +32.7%]. Despite the upward shift in measured margin, it remains above ±20% . We attribute this shift to the on-chip resistance being about 36 % higher than the design value.

Increasing the capacity of the DFF improves the storage density of SFQ-based SRs. Fig. 9 compares the number of Josephson junctions (JJs) required to store data of various bit sequence lengths in HSTP using HC-DFF and conventional DFF-based SRs. HC-DFF_UC and DFF_UC denote the number of JJs needed to store the UC sequence in HC-DFF and conventional DFF-based SR, respectively. As shown in Fig. 9, each additional bit increases the number of JJs by one in HC-DFF, whereas conventional SRs require one extra DFF cell per bit. Even when considering only the JJs within the DFFs and excluding wiring costs, HC-DFF uses approximately 83.3% fewer JJs per bit than conventional SRs. DFF_binary represents the number of JJs consumed by conventional SRs when storing binary data with equivalent precision to the UC sequence. Because representing UC data with the same precision as an n-bit binary value requires a 2^n-bit sequence, HC-DFF_UC loses its advantage as the bit sequence length increases. However, for sequences shorter than 31 bits, HC-DFF requires fewer JJs than conventional SRs storing equivalent binary data. The red triangle in Fig. 9 indicates the previously designed HC-DRO [23] capable of storing up to 3-bit data. Although it uses the fewest JJs due to storing multiple flux quanta in a single loop, it lacks scalability.

Fig. 10 compares bias and critical margins of HC-DFFs for storing multiple flux quanta using a single storage loop and cascaded multiple loops, where the SL and ML denote HC-DFF

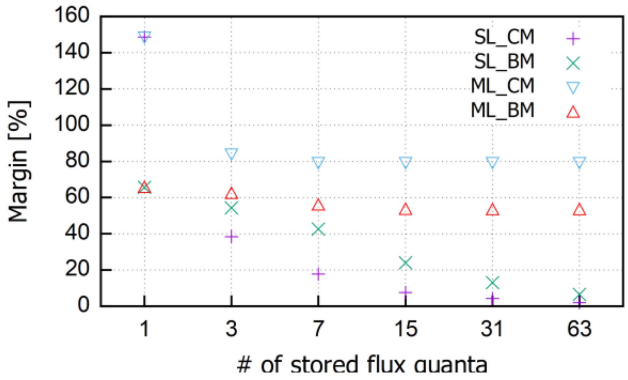

Fig. 10. Comparison of the margin between storing multiple flux quanta in a single loop and multiple loops. SL and ML denote HC-DFF with a single loop and multiple loops configurations; CM and BM denote critical margin and bias margin.

with a single loop and multiple loops configurations, respectively; CM and BM denote critical margin and bias margin. All margins were obtained after parameter optimization with MarginX [31]. As the number of stored flux quanta increases, the CM and BM of ML remain around 79.2% and 53.5%, respectively, while the CM and BM of SL steadily decline, dropping to only 1.9% and 6.5% at 63 flux quanta.

IV. Conclusion

We proposed a novel HC-DFF for high-density UC sequence storage and re-encoding. The correct operation was confirmed by low-speed measurements. The HC-DFF is implemented by cascading multiple storage loops, which can scale to any bit sequence length while maintaining a stable operating margin. These cascaded storage loops are realized using JTLs, allowing conventional JTL wiring to serve both data transmission and storage. Compared to DFF-based SRs that store binary numbers, HC-DFF offers advantages for low-precision data, consistent with the characteristics of UC signals.

Considering these properties, we believe that the future of the proposed HC-DFF can be used in two main research directions. The first involves leveraging the area efficiency of HC-DFF for low-precision storage in hybrid microprocessor architectures that combine unary and binary processing. Integrating encoders and decoders for data conversion between binary computation and unary storage, as demonstrated in [23] and [24], can improve overall system efficiency. The second direction focuses on extending the cascaded storage loop concept to other SFQ logic functions. Replacing the DFF in the HC-DFF with alternative SFQ logic gates could enhance performance in unary applications, such as neural networks and other low-precision tasks.

Acknowledgment

The circuits were fabricated in the Superconducting Quantum Circuit Fabrication Facility (Qufab) in the National Institute of Advanced Industrial Science and Technology.

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] X. Peng et al., "High-speed demonstration of bit-serial floating-point adders and multipliers using single-flux-quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 25, no. 3, Jun. 2015, Art. no. 1301106.

[3] G.-M. Tang, K. Takata, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, "4-bit bit-slice arithmetic logic unit for 32-bit RSFQ microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 1, Jan. 2016, Art. no. 1300106.

[4] Y. Yamanashi et al., "Design and implementation of a pipelined bit-serial SFQ microprocessor, CORE1β," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 474–477, Jun. 2007.

[5] G.-M. Tang et al., "Bit-slice butterfly processing units for 64-point RSFQ FFT processors," *IEEE Trans. Appl. Supercond.*, vol. 30, no. 1, Jan. 2020, Art. no. 1300106.

[6] F. Ke, O. Chen, Y. Wang, and N. Yoshikawa, "Demonstration of a 47.8 GHz high-speed FFT processor using single-flux-quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1300905.

[7] R. Kashima, I. Nagaoka, M. Tanaka, T. Yamashita, and A. Fujimaki, "64-GHz datapath demonstration for bit-parallel SFQ microprocessors based on a gate-level-pipeline structure," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 5, Aug. 2021, Art. no. 1301006.

[8] P. Qu et al., "An energy-efficient 32-bit bit-parallel superconducting SFQ specialized processor," *Superconductivity*, vol. 10, 2024, Art. no. 100099.

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

[10] Z. Li, Y. Yamanashi, and N. Yoshikawa, "Design of binary convolution operation circuit for binarized neural networks using single-flux-quantum circuit," *IEEE Trans. Appl. Supercond.*, vol. 32, no. 4, Jun. 2022, Art. no. 1300305.

[11] Z. Liu et al., "SUSHI: Ultra-high-speed and ultra-low-power neuromorphic chip using superconducting single-flux-quantum circuits," in *Proc. IEEE/ACM Int. Symp. Microarchit.*, 2023, pp. 614–627.

[12] P. Gonzalez-Guerrero et al., "Toward practical superconducting accelerators for machine learning using U-SFQ," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 20, no. 2, pp. 1–22, Jun. 2025.

[13] R. Qiu et al., "Design of a finite impulse response filter for rapid single-flux-quantum signal processors based on stochastic computing," *Superconductivity*, vol. 6, 2023, Art. no. 100045.

[14] Y. Yamanashi et al., "Stochastic matrix multiplier using superconductor single-flux-quantum circuit," *Supercond. Sci. Technol.*, vol. 37, no. 11, Oct. 2024, Art. no. 115024.

[15] K. Asaka et al., "Large fan-out single-flux-quantum stochastic number splitter utilizing frequency-synchronized superconductor random number generators," *Supercond. Sci. Technol.*, vol. 38, no. 4, Apr. 2025, Art. no. 045009.

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

[17] Z. Han et al., "High area efficiency binary neural processing elements with time-domain signals using SFQ circuit," *Supercond. Sci. Technol.*, vol. 37, no. 4, Apr. 2024, Art. no. 045014.

[18] S. Kak, "Generalized unary coding," *Circuits Syst. Signal Process.*, vol. 35, no. 4, pp. 1419–1426, Jul. 2016.

[19] A. H. Jalilvand, M. H. Najafi, and M. Fazeli, "Fuzzy-logic using unary bit-stream processing," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2020, pp. 1–5.

[20] M. H. Najafi, D. J. Lilja, M. D. Riedel, and K. Bazargan, "Low-cost sorting network circuits using unary processing," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 26, no. 8, pp. 1471–1480, Aug. 2018.

[21] D. Wu, J. Li, R. Yin, H. Hsiao, Y. Kim, and J. S. Miguel, "uGEMM: Unary computing for GEMM applications," in *Proc. ACM/IEEE 47th Annu. Int. Symp. Comput. Archit.*, 2020, pp. 377–390.

[22] P. Vellaisamy, H. Nair, D. Wu, S. Blanton, and J. P. Shen, "Exploration of unary arithmetic-based matrix multiply units for low precision DL accelerators," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2024, pp. 661–665.

[23] N. K. Katam et al., "Multi fluxon storage and its implications for microprocessor design," in *Proc. J. Phys., Conf. Ser.*, 2019, Art. no. 012004.

[24] H. Zha, N. K. Katam, M. Pedram, and M. Annavaram, "HiPerRF: A dual-bit dense storage SFQ register file," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit.*, 2022, pp. 415–428.

[25] D. Jenson and M. Riedel, "A deterministic approach to stochastic computation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2016, pp. 1–8.
[26] M. H. Najafi, D. Jenson, D. J. Lilja, and M. D. Riedel, "Performing stochastic computation deterministically," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 27, no. 12, pp. 2925–2938, Dec. 2019.
[27] P. Gonzalez-Guerrero et al., "Temporal and SFQ pulse-streams encoding for area-efficient superconducting accelerators," in *Proc. 27th ACM Int. Conf. Archit. Support Prog.. Lang. Oper. Syst.*, 2022, pp. 963–976.
[28] Z. Han, N. Yoshikawa, and Y. Yamanashi, "Small-area sorting network based on unary coding using single flux quantum circuits," *IEEE Trans. Appl. Supercond.*, vol. 35, no. 5, Aug. 2025, Art. no. 1400205.
[29] N. Takeuchi et al., "On-chip RSFQ microwave pulse generator using a multi-flux-quantum driver for controlling superconducting qubits," *Physica C*, vol. 470, no. 20, pp. 1550–1554, Nov. 2010.
[30] D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "Design and evaluation of multi-flux-quantum drivers using under-damped Josephson junctions," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 835–838, Jun. 2011.
[31] S. Matsuoka, N. Yoshikawa, and Y. Yamanashi, "MarginX: Simple and fast circuit parameter optimization tool for superconductor circuits," *IEICE Trans. Electron.*, vol. E108-C, no. 11, Nov. 2025, to be published, doi: 10.1587/transele.2024ECP5100.
[32] S. Yorozu et al., "A single flux quantum standard logic cell library," *Physica C*, vol. 378, pp. 1471–1474, Oct. 2002.
[33] Y. Yamanashi et al., "100 GHz demonstrations based on the single-flux quantum cell library for the 10 kA/cm^2 Nb multi-layer process," *IEICE Trans. Electron.*, vol. E93-C, no. 4, pp. 440–444, Apr. 2010.
[34] N. Takeuchi et al., "Adiabatic quantum-flux-parametron cell library designed using a 10 kA cm^{-2} niobium fabrication process," *Supercond. Sci. Technol.*, vol. 30, no. 3, Mar. 2017, Art. no. 035002.
[35] M. Hidaka and S. Nagasawa, "Fabrication process for superconducting circuits," *IEICE Trans. Electron.*, vol. E104-C, no. 9, pp. 405–410, Sep. 2021.
[36] J. A. Delport, K. Jackman, P. I. Roux, and C. J. Fourie, "JoSIM—Superconductor SPICE simulator," *IEEE Trans. Appl. Supercond.*, vol. 29, no. 5, Aug. 2019, Art. no. 1300905.
[37] N. Yoshikawa et al., "Cell-based top-down design methodology for RSFQ digital circuits," *Physica C*, vol. 357-360, pp. 1529–1539, Aug. 2001.