

32-Bit ALU with Clockless Gates for RSFQ Bit-Parallel Processor

Takahiro KAWAGUCHI[†], Member and Naofumi TAKAGI^{†a)}, Senior Member

SUMMARY A 32-bit arithmetic logic unit (ALU) is designed for a rapid single flux quantum (RSFQ) bit-parallel processor. In the ALU, clocked gates are partially replaced by clockless gates. This reduces the number of D flip flops (DFFs) required for path balancing. The number of clocked gates, including DFFs, is reduced by approximately 40 %, and size of the clock distribution network is reduced. The number of pipeline stages becomes modest. The layout design of the ALU and simulation results show the effectiveness of using clockless gates in wide datapath circuits.

key words: SFQ digital circuit, ALU, wide datapath circuit, bit-parallel processor, clockless gate

1. Introduction

Rapid single flux quantum (RSFQ) circuits [1] and their energy-efficient derivatives [2]–[4] are expected to be used for realizing high-performance and energy-efficient computing systems, which cannot be achieved using CMOS technologies. Substantial progress has been made in the field of superconductive electronics in the past decades. The current SFQ manufacturing technology is capable of accommodating over 800,000 Josephson junctions (JJs) on a die [5]. It is expected that a 32-bit/64-bit SFQ microprocessor will be developed in the near future [6].

Several SFQ microprocessors were studied [7]–[15]. Until now, most successfully demonstrated SFQ microprocessors are up to 8-bit bit-serial microprocessors [9], [10], [14], [15]. A 32-bit bit-serial microprocessor would require at least 320 ps for processing one word even if it is operated at 100 GHz, and it is not superior to existing high-end CMOS microprocessors. Therefore, it is desired to develop 32-bit/64-bit bit-parallel microprocessors. To develop such a microprocessor, it is necessary to establish a method for designing a wide datapath circuit.

In SFQ digital circuits, SFQ appears as a voltage pulse (SFQ pulse), and it is used as the carrier of information. Logic values of 1 and 0 is typically represented by the presence and absence of a data pulse, respectively, between two consecutive clock pulses. Ordinary logic gates are with a clock input and are referred to as clocked gates. A clocked gate has a latching function, which stores data until a clock pulse arrives. All clocked gates must be supplied with clock pulses. The zero-skew clocking scheme or concurrent-flow

clocking scheme are generally used in these gates [16]–[19].

In these clocking schemes, the number of clocked-gate stages must be the same in all paths from the input to the output of a circuit. A large number of clocked buffers, i.e., D flip flops (DFFs), must be inserted for this path balancing. In addition, clock pulses must also be supplied to these DFFs. The clocking schemes lead to a deep-pipelined circuit, which is referred to as a gate-level-pipelined circuit, where the number of pipeline stages (pipeline depth) is the same as the number of clocked-gate stages.

Although implementing a wide datapath circuit as a gate-level-pipelined circuit results in an extremely high throughput, it requires a large number of DFFs for path balancing and a large clock distribution network. One way to solve this problem is to use clockless gates. A well-known example of a clockless gate is an asynchronous AND gate [20], [21] which can be derived from a simpler Muller C-element (coincidence junction) [1]. A clockless AND gate and a clockless NIMPLY (not imply) gate based on a nondestructive read out (NDRO) were proposed [22], [23]. A clockless dynamic AND gate has been proposed in recent years [24]. Dorojevets et al. suggested the use of asynchronous AND gates and a hybrid wave pipeline for designing a wide datapath circuit [12] and proposed 8-bit arithmetic logic units (ALUs) [25], [26] and a 16-bit adder [27].

In this paper, we present the design of a 32-bit ALU using clockless gates. The ALU consists of 10 gate stages, four of which are composed of clockless gates. These stages do not require DFFs for path balancing. The number of clocked gates, including DFFs, is reduced by approximately 40 %. There are 6 pipeline stages. The partial replacement of clocked gates by clockless gates reduces the number of DFFs required for path balancing, and size of the clock distribution network. It also makes the number of pipeline stages modest. We create a layout design of the ALU with the clockless gates based on an NDRO [22] and clocked gates in the CONNECT cell library [28] for the AIST ADP2 fabrication process [29]. The ALU consists of 16,727 JJs. Its size is $2.475 \text{ mm} \times 7.080 \text{ mm} = 17.5230 \text{ mm}^2$. The estimated clock cycle is 92.7 ps, and the operating frequency is 10.8 GHz. The latency is 556.2 ps (6 cycles). The design has been verified using the static timing analysis and behavior abstraction tools in [30].

The rest of this paper is organized as follows: In Sect. 2, we explain the clockless gates based on the NDRO. In Sect. 3, we present the design of the 32-bit ALU with clockless gates. Section 4 concludes the paper.

Manuscript received July 29, 2021.

Manuscript revised October 18, 2021.

Manuscript publicized December 3, 2021.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.

a) E-mail: takagi@i.kyoto-u.ac.jp

DOI: 10.1587/transele.2021SEP0005

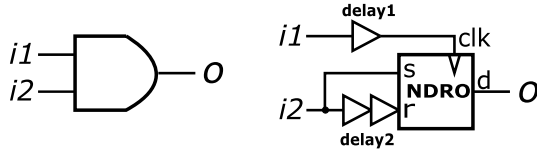


Fig. 1 Symbol and structure of a clockless AND gate

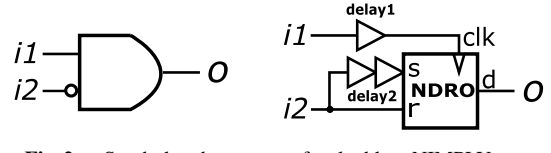


Fig. 2 Symbol and structure of a clockless NIMPLY gate

2. Clockless Gates

In the proposed ALU, we use the clockless AND gate and clockless NIMPLY gate based on an NDRO [22], [23].

An NDRO is widely used in RSFQ circuits. It has three inputs, clk , s , and r , and one output, d . There are two states, $s0$ and $s1$. The state is set to $s1$ by pulse arrival at s and reset to $s0$ by pulse arrival at r . At $s1$, a pulse is produced at d by pulse arrival at clk . At $s0$, no pulse is produced at d by pulse arrival at clk . The interval between the arrival times at clk , s , and r , must be sufficiently long.

The clockless AND gate consists of an NDRO and two delay elements, as shown in Fig. 1. It has two inputs, $i1$ and $i2$, and an output, o . $i1$ is connected to clk through delay element $delay1$. $i2$ is directly connected to s and r through delay element $delay2$. $delay2$ delays a pulse more than $delay1$. A pulse is produced at o only when pulses arrive at both $i1$ and $i2$. When pulses arrive at both $i1$ and $i2$, their arrival times must be close to each other.

A clockless AND cell was realized using the AIST 10-kA/cm² Advanced Process (ADP2) [29]. It consists of 25 JJs. Its size is 60 $\mu\text{m} \times 60 \mu\text{m}$. The delay from pulse arrival at $i1$ to pulse output is 20.9 ps. Note that the timing of the pulse output depends only on that of the pulse arrival at $i1$. The delay varies according to the bias current, similar to clocked gates. The delay shown above is the nominal delay at 100 % bias current. When pulses arrive at both $i1$ and $i2$, they must arrive within approximately 14 ps. If the first pulse arrives at $i1$, the second pulse must arrive at $i2$ within 14.3 ps. If the first pulse arrives at $i2$, the second pulse must arrive at $i1$ within 14.0 ps.

The clockless NIMPLY gate consists of an NDRO and two delay elements, as shown in Fig. 2. It calculates $i1 \wedge \bar{i}2$. The difference between the clockless NIMPLY and AND gates is the position of $delay2$. In the clockless NIMPLY gate, $delay2$ is inserted in the path from $i2$ to s instead of that from $i2$ to r . A pulse is produced at o only when a pulse arrives at $i1$ and no pulse arrives at $i2$. When pulses arrive at both $i1$ and $i2$, their arrival times must be close to each other.

A clockless NIMPLY cell was also realized. Its JJ count, size, and delay are the same as those of the AND gate. When pulses arrive at both $i1$ and $i2$, they must arrive within approximately 10 ps. If the first pulse arrives at $i1$, the second pulse must arrive at $i2$ within 10.0 ps. If the first pulse arrives at $i2$, the second pulse must arrive at $i1$ within 9.8 ps.

We implement a clockless XOR compound gate using

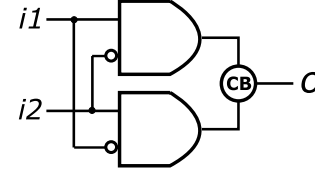


Fig. 3 Clockless XOR compound gate with clockless NIMPLY gates

two NIMPLY gates and a CB, as shown in Fig. 3. Its JJ count is 57 ($= 25 \times 2 + 7$). We directly connect the NIMPLY gates and CB.

3. 32-Bit ALU with Clockless Gates

Table 1 shows the major operations of the designed ALU. The ALU can perform arithmetic operations, i.e., ADD ($X + Y$), SUB ($X - Y$), and EQ ($X = Y?$), and bitwise logic operations, i.e., AND ($X \wedge Y$), OR ($X \vee Y$), XOR ($X \oplus Y$), NOR ($\bar{X} \vee \bar{Y}$), and XNOR ($\bar{X} \oplus \bar{Y}$). In addition, it can perform reverse SUB ($Y - X$), bitwise NAND ($\bar{X} \wedge \bar{Y}$), etc.

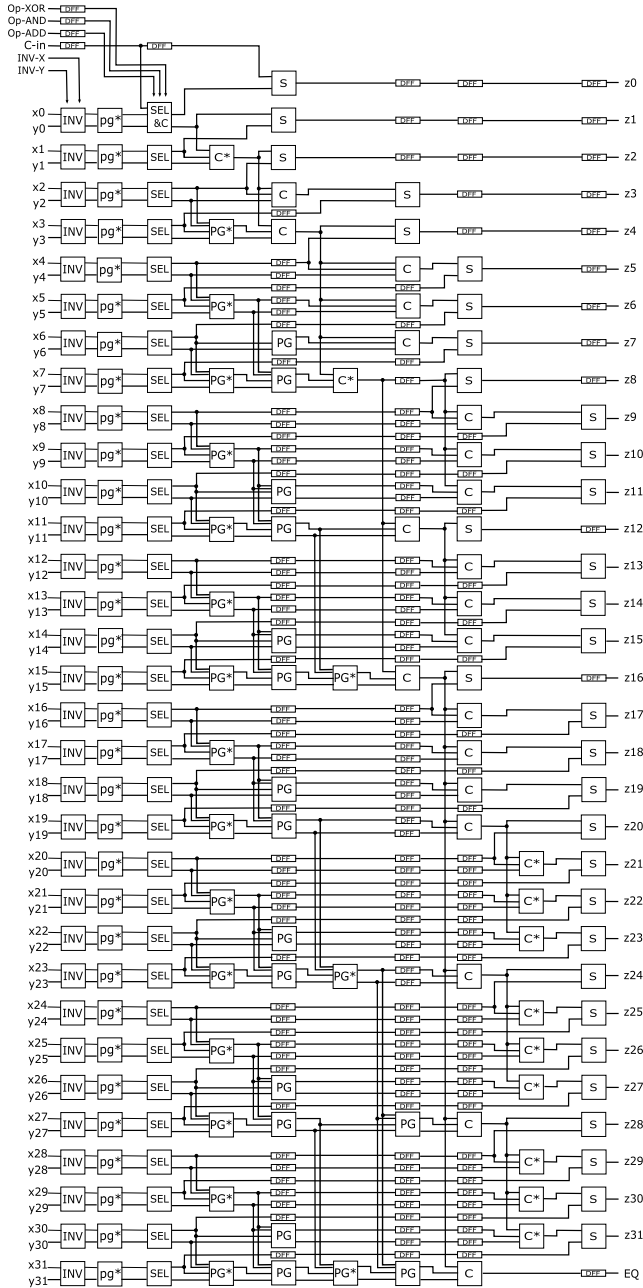
The arithmetic operations are more complex and hardware consuming than the logic operations. Therefore, an adder is used as the main component of the ALU. High-performance adders typically use prefix trees, which generate carries in $\log_2 n$ stages, where n is the number of bits of the datapath. For example, the Kogge-Stone adder [31] and Sklansky adder [32] were used in 4-bit/8-bit SFQ adders. However, the former requires numerous long wires, e.g., n wires spanning $n/2$ bit positions, and the latter requires numerous (maximum n) fanouts.

A parallel-prefix adder structure with a moderate number of long wires and fanouts is required for wide datapath SFQ adders. Similar to the 16-bit SFQ adder in [27], we develop an adder structure based on the sparse-tree adder [33]. In the sparse-tree adder, every four carries are calculated using a carry-merge tree (a prefix tree), and the sum is calculated using 4-bit conditional sum-generators in which the carry-merge logic is serially connected. In [27], a carry-skip adder was used instead of the conditional sum-generators. In our adder, every four carries are calculated using the prefix tree, and they are used to calculate the other carries in parallel similar to the Sklansky adder. An adder structure based on the sparse-tree adder is suitable for wider (e.g., 64-bit) SFQ ALUs. The adder structure must be tuned according to the data width.

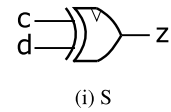
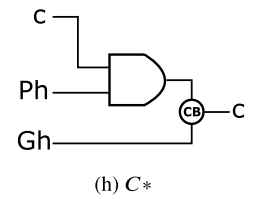
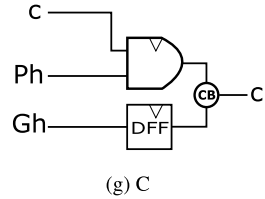
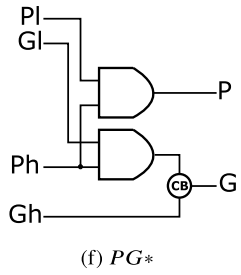
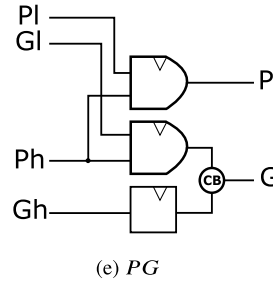
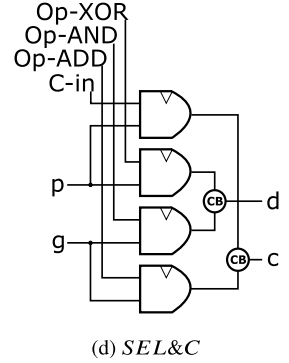
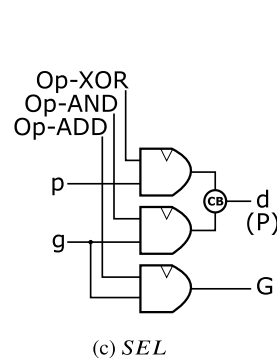
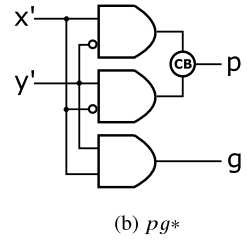
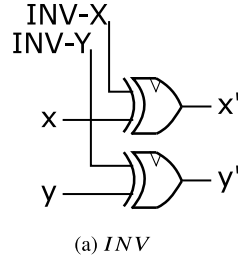
A structural diagram of the ALU is shown in Fig. 4. The ALU consists of nine types of blocks. The details of the blocks are shown in Fig. 5. Blocks pq^* , PG^* , and C^* consist of clockless gates. Block pq^* produces a 'propagation'

Table 1 ALU operations

Operation	Op-XOR	Op-AND	Op-ADD	Inv-X	Inv-Y	C-in
ADD	1	0	1	0	0	0
SUB	1	0	1	0	1	1
AND	0	1	0	0	0	0
OR	1	1	0	0	0	0
XOR	1	0	0	0	0	0
NOR	0	1	0	1	1	0
XNOR	1	0	0	0	0	0
EQ	1	0	0	0	1	1

**Fig. 4** Structural diagram of the ALU

signal, i.e., XOR of the inputs, and a 'generation' signal, i.e., AND of the inputs. As mentioned in the previous section, we can implement a clockless XOR compound gate

**Fig. 5** Details of the blocks

using two clockless NIMPLY gates and a CB. Block *SEL* produces the result for a specified logic operation. Block *PG* (*PG**) calculates the prefix operation, and block *C* (*C**) calculates a carry. Block *S* calculates a sum. Note that *c* is 0 for logic operations.

The ALU consists of 10 gate stages. The second, fourth, sixth, and ninth stages are composed of clockless gates. There are 6 pipeline stages. We target the operation of the ALU at 10 GHz. On this basis, we determine the gate stage that should be clockless so that the delay at each pipeline stage is balanced considering fanouts and wire

lengths.

In the second stage, in each pg^* , a clockless XOR compound gate is used instead of a clocked XOR gate. In the fourth, sixth, and ninth gate stages, PG^* and C^* are used instead of PG and C , respectively. In other words, clockless AND gates are used instead of clocked AND gates. These stages no longer require DFFs for path balancing; the number of DFFs is reduced by approximately 200. Overall, the number of clocked gates, including DFFs, is reduced by more than 300 (approximately 40 %) to 454.

We use the zero-skew clocking scheme for the clocked gates. Even though the concurrent-flow clocking scheme provides a higher operating frequency, clock skew accumulates. This leads to a large time difference in the clock cycle between the input and output of the circuit. This makes it difficult to control the clock cycle of the entire system. Note that the use of clockless gates significantly reduces the number of clocked gates and the DFFs to be supplied with clock pulses.

We create a layout design of the ALU using the clockless gates based on the NDRO and clocked gates in the CONNECT cell library [28] for the AIST ADP2 fabrication process [29]. The design is shown in Fig. 6. The number of JJs is 16,727. The size of the ALU is $2.475 \text{ mm} \times 7.080 \text{ mm}$ = 17.5230 mm^2 . The estimated clock cycle is 92.7 ps, and the operating frequency is 10.8 GHz. The latency is 556.2 ps (6 cycles). We have verified the design using the static timing analysis and behavior abstraction tools in [30].

Compared to the design with only clocked gates, 194 DFFs are removed, 79 clocked AND gates are replaced with clockless AND gates, and 32 clocked XOR gates are replaced with clockless XOR compound gates in the new design. Furthermore, 305 ($= 194 + 79 + 32$) splitters for clock distribution are removed. The number of JJs of a DFF with two PTL receivers (one for data input and the other for clock input) and a PTL driver is $14 (= 6 + 3 \times 2 + 2)$. The number of JJs of a clocked AND gate with three receivers and a driver is $25 (= 14 + 3 \times 3 + 2)$, and that of a clocked XOR gate with three receivers and a driver is $22 (= 11 + 3 \times 3 + 2)$. On the other hand, the number of JJs of a clockless AND gate with two receivers and a driver is $33 (= 25 + 3 \times 2 + 2)$, and that of a clockless XOR compound gate with four receivers and a driver is $71 (= 57 + 3 \times 4 + 2)$. For a clockless XOR compound gate, two splitters with a driver are required for splitting the data inputs into two NIMPLY gates. The number of JJs of a splitter with a driver is $5 (= 3 + 2)$. Therefore, the number of JJs in the designed ALU is 1,710 ($= 14 \times 194 + (25 - 33) \times 79 + (22 - (71 + 5 \times 2)) \times 32 + 5 \times 305$) less than that of the design with completely clocked gates.

We may reduce the JJ count further using a clockless dynamic AND gate [24] instead of the clockless AND gate based on the NDRO and by developing a clockless XOR gate. From the point of view of the JJ count, it may be better to not replace clocked XOR gates with clockless XOR compound gates.

We can change the number of pipeline stages by changing the number of stages of the clockless gates. For ex-

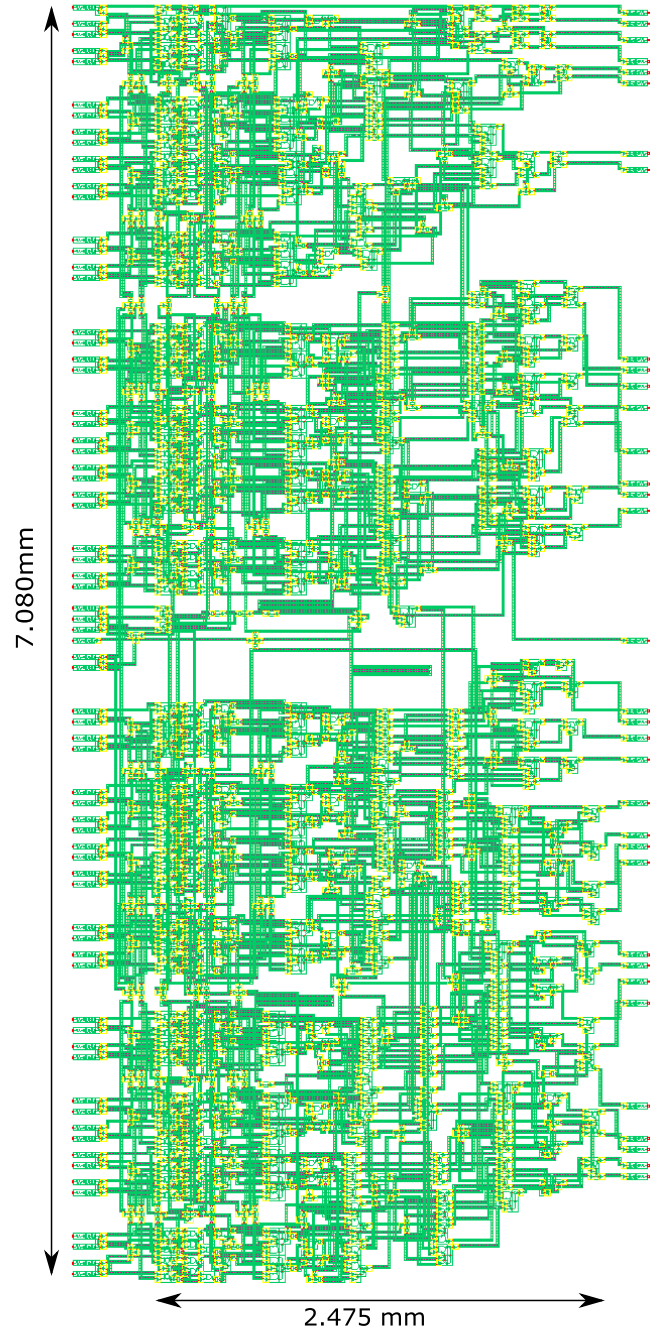


Fig. 6 Layout design of the ALU

ample, we can reduce the number of pipeline stages to 4 by utilizing clockless gates in the second, third, fifth, sixth, eighth, and ninth gate stages. The gate stage in which clockless gates will be used should be selected such that the delay at each pipeline stage is balanced considering fanouts and wire lengths. Furthermore, as the number of pipeline stages decreases, the level of clockless gates in each pipeline stage may increase, and the cycle time may increase. As the clockless gates have the limitation that the pulse arrival times at the inputs must be close to each other, the maximum level of clockless gates in each pipeline stage may be

limited by the timing jitter on paths.

4. Conclusion

We have shown the feasibility of realizing a wide datapath circuit using clockless gates by designing a 32-bit ALU. We partially replace clocked gates with clockless gates. This reduces the number of DFFs required for path balancing and size of the clock distribution network, and makes the number of pipeline stages modest.

One of the merits of using clockless gates in the development of bit-parallel processors is the flexibility in determining the number of pipeline stages of component circuits. The results of this study show that the use of clockless gates and the zero-skew clocking scheme for clocked gates is an effective method for designing wide datapath component circuits for 32-bit/64-bit microprocessors.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 18H05211 and also supported through the activities of VDEC, The University of Tokyo, in collaboration with Cadence Design Systems.

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.2–28, March 1991.
- [2] O.A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Supercond.*, vol.21, no.3, pp.760–769, June 2011.
- [3] D.E. Kirichenko, S. Sarwana, and A.F. Kirichenko, "Zero static power dissipation biasing of RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol.21, no.3, pp.776–779, June 2011.
- [4] M. Tanaka, A. Kitayama, T. Koketsu, M. Ito, and A. Fujimaki, "Low-energy consumption RSFQ circuits driven by low voltages," *IEEE Trans. Appl. Supercond.*, vol.23, no.3, #1701104, June 2013.
- [5] V.K. Semenov, Y.A. Polyakov, and S.K. Tolpygo, "AC-biased shift registers as Fabrication process benchmark circuits and flux trapping diagnostic tool," *IEEE Trans. Appl. Supercond.*, vol.27, no.4, #1301409, June 2017.
- [6] IARPA, "Cryogenic Computing Complexity (C3)," <https://www.iarpa.gov/index.php/research-programs/c3>
- [7] M. Dorojevets, P. Bunyk, and D. Zinoviev, "FLUX Chip: Design of a 20-GHz 16-bit ultrapipelined RSFQ processor prototype based on 1.75- μ m LTS technology," *IEEE Trans. Appl. Supercond.*, vol.11, no.1, pp.326–332, March 2001.
- [8] N. Yoshikawa, F. Matsuzaki, N. Nakajima, K. Fujiwara, K. Yoda, and K. Kawasaki, "Design and component test of a tiny processor based on the SFQ Technology," *IEEE Trans. Appl. Supercond.*, vol.13, no.2, pp.441–445, June 2003.
- [9] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, A. Fujimaki, H. Hayakawa, N. Yoshikawa, H. Terai, and S. Yoyozu, "A single-flux-quantum logic prototype microprocessor," *Tech. Digest of 2004 IEEE International Solid-State Circuits Conference (ISSCC 2004)*, pp.298–299, Feb. 2004.
- [10] Y. Yamanashi, M. Tanaka, A. Akimoto, H. Park, Y. Kamiya, N. Irie, N. Yoshikawa, A. Fujimaki, H. Terai, and Y. Hashimoto, "Design and implementation of a pipelined bit-serial SFQ microprocessor, CORE1 β ," *IEEE Trans. Appl. Supercond.*, vol.17, no.2, pp.474–477, June 2007.
- [11] Y. Nobumori, T. Nishigai, K. Nakamiya, N. Yoshikawa, A. Fujimaki, H. Terai, and S. Yorozu, "Design and implementation of a fully asynchronous SFQ microprocessor: SCRAM2," *IEEE Trans. Appl. Supercond.*, vol.17, no.2, pp.478–481, June 2007.
- [12] 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, June 2011.
- [13] G. 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, #1300106, Jan. 2016.
- [14] Y. Ando, R. Sato, M. Tanaka, K. Takagi, N. Takagi, and A. Fujimaki, "Design and Demonstration of an 8-bit Bit-Serial RSFQ Microprocessor: CORE e4," *IEEE Trans. Appl. Superconductivity*, vol.26, no.5, #1301205, Aug. 2016.
- [15] R. Sato, Y. Hatanaka, Y. Ando, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, "High-speed operation of random-access-memory-embedded microprocessor with minimal instruction set architecture based on rapid single-flux-quantum logic," *IEEE Trans. Appl. Supercond.*, vol.27, no.4, #1300505, June 2017.
- [16] M. Tanaka, H. Akaike, A. Fujimaki, Y. Yamanashi, N. Yoshikawa, S. Nagasawa, K. Takagi, and N. Takagi, "100-GHz single-flux-quantum bit-serial adder based on 10-kA/cm² niobium process," *IEEE Trans. Appl. Supercond.*, vol.21, no.3, pp.792–796, June 2011.
- [17] S.N. Shahsavani, T.-R. Lin, A. Shafaei, C.J. Fourie, and M. Pedram, "An integrated row-based cell placement and interconnect synthesis tool for large SFQ logic circuits," *IEEE Trans. Appl. Supercond.*, vol.27, no.4, #1302008, June 2017.
- [18] N.K. Katam and M. Pedram, "Logic optimization, complex cell design, and retiming of single flux quantum circuits," *IEEE Trans. Appl. Supercond.*, vol.28, no.7, #1301409, Oct. 2018.
- [19] S.N. Shahsavani and M. Pedram, "A minimum-skew clock tree synthesis algorithm for single flux quantum logic circuits," *IEEE Trans. Appl. Supercond.*, vol.29, no.8, #1303513, Dec. 2019.
- [20] O.A. Mukhanov and A.F. Kirichenko, "A superconductive high-resolution time-to-digital converter," *Proc. Extended Abstracts 7th Int. Supercond. Electron. Conf. (ISEC1999)*, pp.353–355, June 21–25, 1999.
- [21] S.B. Kaplan, A.F. Kirichenko, O.A. Mukhanov, and S. Sarwana, "A prescaler circuit for a superconductive time-to-digital converter," *IEEE Trans. Appl. Supercond.*, vol.11, no.1, pp.513–516, March 2001.
- [22] T. Kawaguchi, M. Tanaka, K. Takagi, and N. Takagi, "Demonstration of an 8-bit SFQ carry look-ahead adder using clockless logic cells," *Proc. 15th International Superconductive Electronics Conference (ISEC2015)*, pp.1–3, 2015.
- [23] T. Kawaguchi, M. Tanaka, K. Takagi, and N. Takagi, "Rapid single-flux-quantum logic circuits using clockless gates," *IEEE Trans. Appl. Supercond.*, vol.31, no.4, #1302407, June 2011.
- [24] S.V. Rylov, "Clockless dynamic SFQ AND gate with high input skew tolerance," *IEEE Trans. Appl. Supercond.*, vol.29, no.5, #1300805, Aug. 2019.
- [25] T. Filippov, M. Dorojevets, A. Sahu, A. Kirichenko, C. Ayala, and O. Mukhanov, "8-bit asynchronous wave-pipelined RSFQ arithmetic-logic unit," *IEEE Trans. Appl. Supercond.*, vol.21, no.3, pp.847–851, June 2011.
- [26] M. Dorojevets, C.L. Ayala, N. Yoshikawa, and A. Fujimaki, "8-bit asynchronous sparse-tree superconductor RSFQ arithmetic-logic unit with a rich set of operations," *IEEE Trans. Appl. Supercond.*, vol.23, no.3, #1700104, June 2013.
- [27] 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, #1700605, June 2013.
- [28] Y. Yamanashi, T. Kainuma, N. Yoshikawa, I. Kataeva, H. Akaike, A. Fujimaki, M. Tanaka, N. Takagi, S. Nagasawa, and M. Hidaka, "100 GHz demonstrations based on the single-flux-quantum cell library for the 10 ka/cm² Nb multi-layer process," *IEICE Trans. Electron.*,

- vol.E93-C, no.4, pp.440–444, April 2010.
- [29] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, M. Hidaka, H. Akaike, A. Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa, “New Nb multi-layer fabrication process for large-scale SFQ circuits,” *Physica C: Superconductivity and Its Applications*, vol.469, no.15–20, pp.1578–1584, Oct. 2009.
 - [30] T. Kawaguchi, K. Takagi, and N. Takagi, “A verification method for single-flux-quantum circuits using delay-based time frame model,” *IEICE Trans. Fundamentals*, vol.E98-C, no.12, pp.2556–2564, Dec. 2015.
 - [31] P. Kogge and H.S. Stone, “A parallel algorithm for the efficient solution of a general class of recurrence equations,” *IEEE Trans. Comput.*, vol.C-22, no.9, pp.786–793, Aug. 1973.
 - [32] J. Sklansky, “Conditional-sum addition logic,” *IRE Trans. Electron. Comput.*, vol.EC-9, no.2, pp.226–231, June 1960.
 - [33] S. Mathew, M. Anders, R.K. Krishnamurthy, and S. Borkar, “A 4-GHz 130-nm address generation unit with 32-bit sparse-tree adder core,” *IEEE J. Solid-State Circuits*, vol.38, no.5, pp.689–695, May 2003.



Takahiro Kawaguchi received his B.E. and M.IS degrees in information engineering from Nagoya University, Nagoya, Japan, in 2010 and 2012, respectively. He joined Kyoto University, Kyoto, Japan, as a Research Fellow in 2018. His current research interests include algorithms for the computer-aided design of SFQ integrated circuits.



Naofumi Takagi received his B.E., M.E., and Ph.D. degrees in information science from Kyoto University, Kyoto, Japan, in 1981, 1983, and 1988, respectively. He joined Kyoto University as an Instructor in 1984, and he was promoted to an Associate Professor in 1991. He moved to Nagoya University, Nagoya, Japan, in 1994, and he was promoted to a Professor in 1998. He returned to Kyoto University in 2010. His current research interests include computer arithmetic, hardware algorithms, and logic design.

Dr. Takagi received the Japan IBM Science Award and the Sakai Memorial Award of the Information Processing Society of Japan in 1995 and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology of Japan in 2005.