

## Design and implementation of stochastic neurosystem using SFQ logic circuits

著者	Kondo Taizo, Kobori Masayuki, Onomi Takeshi, Nakajima Koji
journal or publication title	IEEE Transactions on Applied Superconductivity
volume	15
number	2
page range	320-323
year	2005
URL	<a href="http://hdl.handle.net/10097/48237">http://hdl.handle.net/10097/48237</a>

doi: 10.1109/TASC.2005.849818

# Design and Implementation of Stochastic Neurosystem Using SFQ Logic Circuits

Taizo Kondo, Masayuki Kobori, Takeshi Onomi, and Koji Nakajima

**Abstract**—We propose a stochastic neurosystem using SFQ logic circuits and design the main components with the following functions: carrying out the multiplication of an input to a neuron on a synaptic weight value, integrating pulses to generate a membrane potential, and generating the output of a neuron. We simulate some circuits by JSIM and confirm their correct operation. We compare two methods of multipliers: using a comparator and using a divider. The multiplication using the divider is effective with respect to integration, and reduces the accumulation time  $N_a$  required for higher precision operations. We designed a 4-bit up/down counter assuming the NEC 2.5 kA/cm<sup>2</sup> Nb/AIO<sub>x</sub>/Nb standard process. We show that it is possible to compose the activation function circuit using a comparator.

**Index Terms**—Neural network, single flux quantum, stochastic logic, up/down counter.

## I. INTRODUCTION

APPLICATIONS of a neural network are expected in fields such as the pattern recognition and combinational optimization problems, tasks that conventional computers are not good at. A simple neural network is realizable in software simulation. However, it is not realistic with respect to calculation time and network scale when a neural network is used for applications in real-time processing. Although the number of neurons for the practical application is not obvious, the more the number of neurons increases, the higher the information processing a neural network can carry out. Therefore, it is necessary to integrate a neuron by implementing it as hardware.

A neural network using stochastic logic, which has been proposed by Kondo *et al.* [1], is suitable for this integration. The stochastic neurochips, including dozens of neurons, have been designed and fabricated using CMOS technology [2]–[4]. In stochastic logic, one can convert an analog value to occurrence probability of a pulse and carry out operations using the probability. A comparator is used for the conversion. The pulse sequence is generated when a value  $X$  is more than a random number  $R$  (Fig. 1(a)). The relationship between the probability of the pulse sequence generated and  $X$  is shown in Fig. 1(b). The pulse sequence has noise, which originates in the coding (coding noise). In some kinds of problems, the performance of the stochastic neural network is improved by the coding noise as compared to the deterministic neural network [1]. Encoding is performed by accumulating the pulse sequence for a certain period of time. This logic has merit in that complex operations are

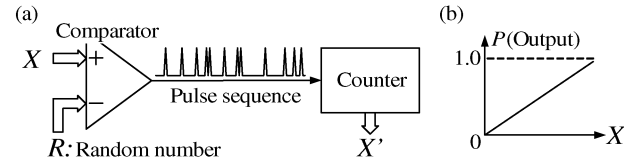


Fig. 1. (a) Coding and encoding circuits. Comparator outputs a pulse if  $X > R$ . (b) Relationships between the probability of the pulse sequence generated and  $X$ .

realizable with basic digital logic gates. For example, the multiplication of the pulse sequences is carried out with a single AND gate. Nemoto *et al.* have reported that the number of transistors for a neuron in stochastic logic is 1/5 of that in deterministic logic [2]. We require  $n^2$  synapses in the full-connection neural network including  $n$  neurons. The synaptic circuit in stochastic logic requires 1300 Josephson junctions (JJs) at a rough estimate. This number is about half of that in deterministic logic. However, it has demerit as operation becomes slow, because it is necessary to increase accumulation time for more precise operation.

Hence, we propose a stochastic neural network using single-flux-quantum (SFQ) logic circuits [5], [6]. SFQ logic circuits use a single flux quantum as an information carrier and are attractive because of their high speed operation and low power dissipation. Therefore, SFQ logic circuits are the candidates for the main data processor in the next generation. We expect to cancel the demerit of slow operation in stochastic logic with SFQ logic circuits, and SFQ logic circuits have good characteristics for stochastic logic because both are pulse logic.

In this paper, we report the design of necessary circuits in order to compose a stochastic neuron using SFQ logic circuits. We designed the circuit generating membrane potentials assuming the NEC 2.5 kA/cm<sup>2</sup> Nb/AIO<sub>x</sub>/Nb standard process. We investigate the difference between the performance and consumption area of a multiplier circuit using a comparator and that using a divider.

The numerical simulation is carried out by JSIM [7] and SCOPE [8] assuming the NEC 2.5 kA/cm<sup>2</sup> Nb/AIO<sub>x</sub>/Nb standard process.

## II. STOCHASTIC NEURON

We design a stochastic neuron on the basis of the system that has already been proposed [1]–[4]. Fig. 2(a) shows the model of a neuron. The discrete-time dynamics of a neural network are given by

$$u_i(t+1) = \sum_{j=1}^N w_{ij} x_j(t), \quad (1)$$

$$x_i(t) = f(u_i(t)), \quad (2)$$

Manuscript received October 4, 2004; revised November 30, 2004.

The authors are with the Intelligent Nano-Integration System Laboratory for Brainware Systems, Research Institute of Electrical Communication (RIEC), Tohoku University, Sendai 980-8577, Japan (e-mail: kondo@nakajima.riec.tohoku.ac.jp; m-kobori@nakajima.riec.tohoku.ac.jp; onomi@riec.tohoku.ac.jp; hello@riec.tohoku.ac.jp).

Digital Object Identifier 10.1109/TASC.2005.849818

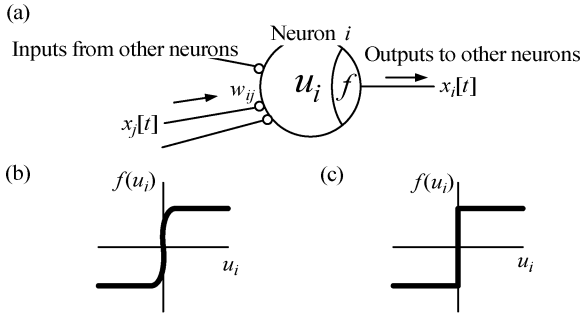


Fig. 2. (a) A neuron model, (b) sigmoid function, and (c) sign function.

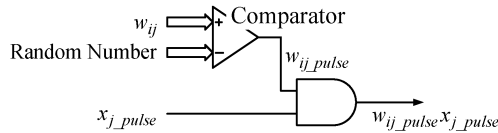


Fig. 3. Multiplication using comparator.

where  $w_{ij}$  is the synaptic weight from the  $j$ th neuron to the  $i$ th neuron,  $x_j$  is the neuron output,  $u_i$  is the membrane potential,  $f(u)$  is an activation function, and  $N$  is the number of neurons. The activation function is a sigmoid function, or a sign function, etc. (Fig. 2(b) and (c)). In order to realize a neuron which carries out such operations, we require the following circuits: one carrying out the multiplying of the output of a neuron by a synaptic weight value, one accumulating a pulse to generate membrane potential, and one generating the output of a neuron.

### A. Multiplier

We require a circuit for multiplying the synaptic weight  $w_{ij}$  on the output  $x_{j\_pulse}$  of a neuron. Two methods are able to realize this. One method is a multiplier using comparators; the other is one using dividers.

1) *Multiplier Using Comparator*: Fig. 3 shows multiplication using a comparator. The comparator outputs a pulse when  $w_{ij}$  is larger than a random number. In other words, the comparator converts  $w_{ij}$  to a pulse sequence generated in a probability proportional to the value. Multiplication is realized by implementing the logical AND operation between the pulse sequence and the output  $x_{j\_pulse}$  from another neuron. Therefore, the occurrence probability of the output  $w_{ij\_pulse}x_{j\_pulse}$  is given as follows:

$$\begin{aligned} P(w_{ij\_pulse}x_{j\_pulse}) &= P(w_{ij\_pulse}) \cdot P(x_{j\_pulse}) \\ &= \frac{w_{ij}}{w_{ij\_MAX}} P(x_{j\_pulse}), \end{aligned} \quad (3)$$

where  $w_{ij\_MAX}$  is the maximum value of the synaptic weight and the random number. This circuit requires a comparator and a random number generator.

- a) *Random number generator*: We use M-code as a random number even though it is only pseudo random number because it can be composed of simple circuits. Some M-code generators were designed and fabricated using SFQ logic circuits [9], [10]. We design a 4-bit M-code generator using XOR, Inverter, and 2 DFFs (D Flip-Flops). This circuit has the feature that no start signal is required and the operation margin is securable (Fig. 4). According to

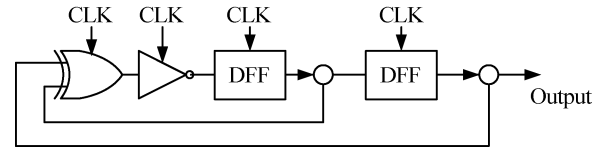


Fig. 4. 4-bit M-code generator.

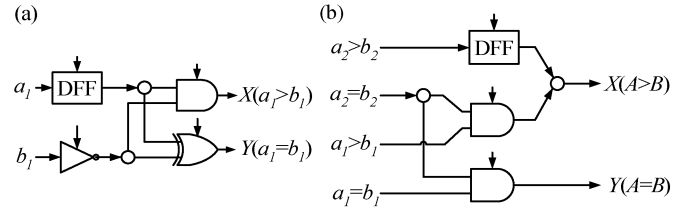


Fig. 5. (a) 1-bit comparator and (b) 4in-2out comparator.

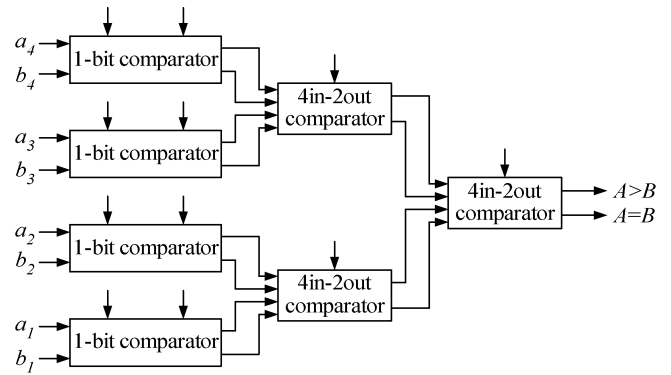


Fig. 6. 4-bit pipelined comparator.

the simulations by JSIM, bias margins are ranged from  $-32.5\%$  to  $+35.6\%$  at 10.0 GHz and the maximum operating speed is 18.9 GHz.

- b) *Comparator*: A multi-bit pipelined comparator can be assembled using 1-bit comparators (Fig. 5(a)) and 4in-2out comparators (Fig. 5(b)). When a 1-bit comparator compares  $a_1$  with  $b_1$ , it performs the following operations:

- If  $a_1$  is more than  $b_1$ , the output pulse  $X$  is generated.
- If  $a_1$  is equal to  $b_1$ , the output pulse  $Y$  is generated.
- If  $a_1$  is less than  $b_1$ , no output pulse is generated.

When  $A(= a_2, a_1)$  is compared with  $B(= b_2, b_1)$ , each bit is compared using a 1-bit comparator. Then, by using the outputs of two 1-bit comparators, a 4in-2out comparator carries out the following operations:

- If  $A$  is more than  $B$ , the output pulse  $X$  is generated.
- If  $A$  is equal to  $B$ , the output pulse  $Y$  is generated.
- If  $A$  is less than  $B$ , no output pulse is generated.

Fig. 6 shows a 4-bit pipelined comparator. The correct operation is verified by simulation, JSIM. The simulated upper and lower margins are  $+25.0\%$  and  $-22.2\%$  at 5.0 GHz, respectively.

- 2) *Multiplier Using Divider*: The multiplier (Fig. 7) is also realizable by using dividing circuits including T Flip-Flops and NDROs (Non Destructive Read Out) [6]. The input  $x_{j\_pulse}$  is divided into  $1/2$ ,  $1/4$ ,  $\dots$  by TFFs. We can change the ratio of the input  $x_{j\_pulse}$  and the output  $w_{ij\_pulse}x_{j\_pulse}$  by shifting the internal states of NDROs. When this circuit is used

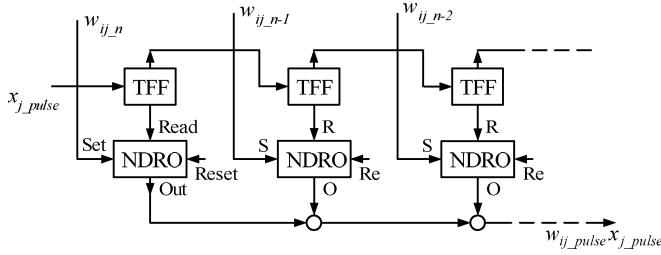


Fig. 7. Multiplication using divider.

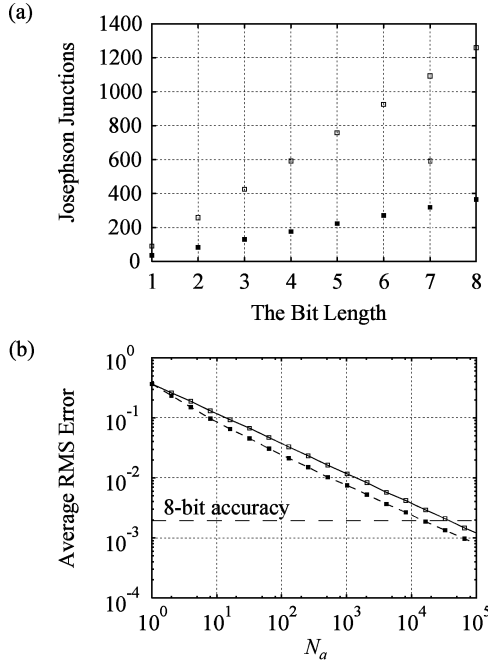


Fig. 8.  $\square$  and  $\blacksquare$  denote the multiplier using the comparator and that using the divider cases, respectively. (a) Comparison of the estimated number of JJs of both methods. In case of multiplier using comparator, CLK line is not included. (b) Comparison of the average RMS errors of the outputs of both methods.

as  $n$ -bit multiplier, the occurrence probability of the output  $w_{ij\_pulse}x_{j\_pulse}$  is obtained as follows:

$$P(w_{ij\_pulse}x_{j\_pulse}) = \left( \frac{1}{2}w_{ij\_n} + \frac{1}{4}w_{ij\_n-1} + \dots \right) P(x_{j\_pulse}), \quad (4)$$

where  $w_{ij\_n}$  is  $n$ th bit of  $w_{ij}$ . This circuit is a sequential circuit, and has  $2^n$  internal states.

3) *Comparison of Two Circuits:* The comparison of estimated number of JJs is shown in Fig. 8(a). The consumption area of a circuit is proportional to the number of JJs. Thus the multiplier using dividers has an advantage for integration over that using comparators. However, the multiplier using dividers is an asynchronous circuit making it difficult to adjust the timing, particularly when the bit length increases. On the other hand, the multiplier using comparators has a constant throughput regardless of the bit length since it is a pipelined circuit.

Fig. 8(b) shows the comparison of average root-mean-square (RMS) errors of the outputs  $w_{ij\_pulse}x_{j\_pulse}$  of both methods.

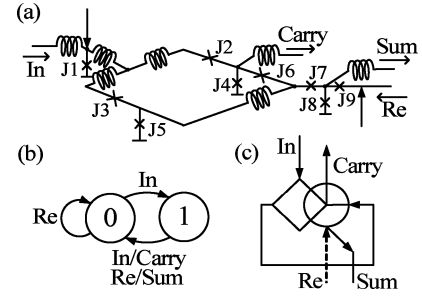


Fig. 9. Adder cell. (a) Circuit diagram, (b) Moore diagram, and (c) symbol.

The average RMS error is obtained by numerical simulation on the three following conditions. First, the error is defined as the difference between deterministic outputs. Second, each trial is carried out 10 000 times in each accumulation time  $N_a$ . Third,  $w_{ij}$  and  $x_{j\_pulse}$  are given at random in each trial. The output  $w_{ij\_pulse}x_{j\_pulse}$  is more precise with increasing  $N_a$ . Infinite  $N_a$  means deterministic. Therefore, the average RMS error is almost inversely proportional to  $N_a$  in both cases. The multiplier using the comparator has two coding noises, one from the input  $x_{j\_pulse}$ , and one from the pulse sequence  $w_{ij\_pulse}$ . The two coding noises influence the output  $w_{ij\_pulse}x_{j\_pulse}$ . In contrast, the multiplier using the divider has coding noise only in  $x_{j\_pulse}$ . Since the coding noise influences the output a little, the average RMS error of the multiplier using the divider is less than that using the comparator. 8 bits in general is required as operation accuracy for dealing with practical applications. In the case of the multiplier using the comparator,  $N_a$  is few ten thousands; in case of that using the divider,  $N_a$  is approximately half. Therefore, the performance of a neural network using dividers may improve with respect to the operation speed because the required  $N_a$  is reduced for higher precision operation.

## B. Up/Down Counter

The membrane potential  $w_i$  is generated by accumulation of the pulse sequence  $w_{ij\_pulse}x_{j\_pulse}$  from other neurons. An up/down counter, which satisfies this function, is designed using an adder cell [11], [12]. An adder cell is a serial input adder (Fig. 9) achieved by improving an ICF gate, which is the fundamental device of the Phase-Mode logic proposed by Nakajima *et al.* [5]. The up/down counter is easily assembled by connecting adder cells serially (Fig. 10(a)). This circuit has two inputs (Up, Down), several outputs ( $O_1, O_2, \dots, O_n$ ), and a reset input (Re).

Incrementing the value of the up/down counter occurs by sending a signal to the Up input. The signal to the Up input propagates only to the input of the adder cell which is the least significant bit (LSB) of the up/down counter. Thus, the value of the up/down counter is set to +1.

On the other hand, decrementing the internal state of the up/down counter occurs by sending a signal to the Down input. The signal to the Down input propagates to the inputs of every adder cell. In other words, the up/down counter receives "111...1" input. "111...1" is expressed as  $-1$  by two's complement. Hence, the value of up/down counter is set to  $-1$ .

The value of the up/down counter can be read out by sending a signal to the Re of every adder cell. The outputs of the multiplier  $w_{ij\_pulse}x_{j\_pulse}$  are brought to Up or Down according to

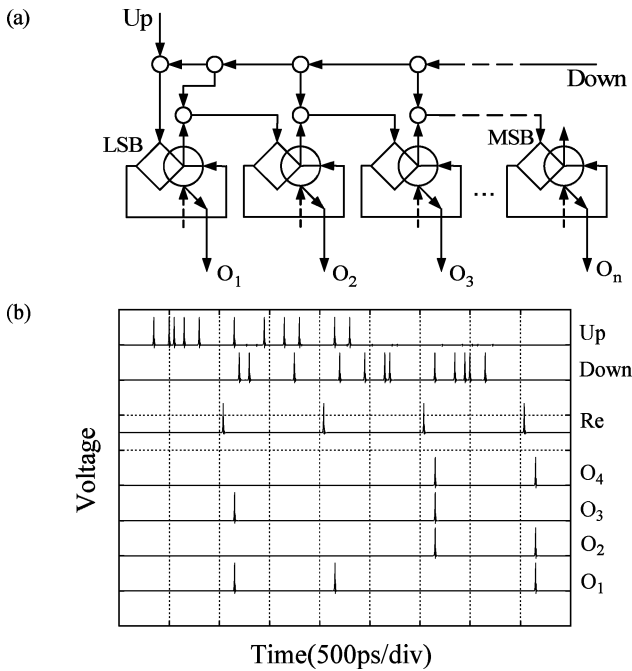


Fig. 10. (a) Block diagram of up/down counter. (b) Simulation result of the 4-bit up/down counter. When Re is applied, the output is +5, +1, -2, and -5 sequentially from a left, respectively.

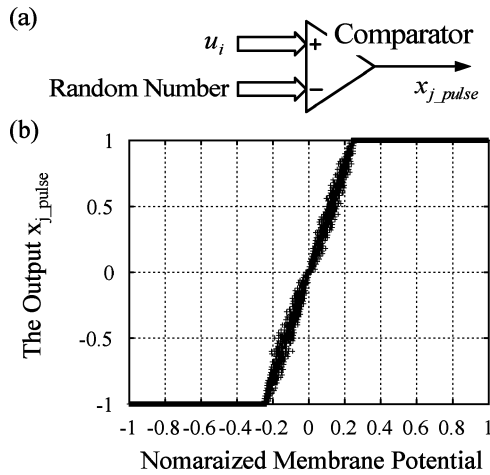


Fig. 11. Activation function circuit. (a) Block diagram. (b) Simulation result. The bit length of the membrane potential and that of random number are 9 bits and 7 bits, respectively.  $N_a$  is 50 and the trial is repeated 10 times. The membrane potential and the output  $x_{j\_pulse}$  are normalized by  $2^9$  and  $N_a$ , respectively.

the sign of  $w_{ij\_pulse}x_{j\_pulse}$ . The membrane potential is generated by the signal to the input Re after the up/down counter has counted the pulses for  $N_a$ .

We confirm the correct operation of a 4-bit up/down counter by JSIM simulation (Fig. 10(b)). We also designed the 4-bit up/down counter consisting of 337 JJs by the NEC 2.5 kA/cm<sup>2</sup> Nb/AIO<sub>x</sub>/Nb standard process. The designed total bias current is 40.01 mA.

### C. Activation Function

The activation function circuit is required to generate the outputs corresponding to the membrane potential. This circuit is assembled by applying the above-mentioned comparator

(Fig. 11(a)). The bit length of the comparator must correspond to the maximum value of the membrane potential. Here, the bit length of a random number is shorter than that of the membrane potential and sets up the threshold value. Fig. 11(b) shows the activation function, which is a monotonic increasing function. It is obtained when the bit length of the membrane potential and the random number are 9 bits and 7 bits, respectively. The noise decreases, and the stochastic output  $x_{j\_pulse}$  is close to the deterministic with increasing  $N_a$ .

### III. CONCLUSION

We propose the stochastic neuron using SFQ logic circuits and design the main components required for that. We simulate some circuits by JSIM and confirm their correct operation. We compare the comparator multiplier with the divider one. The divider multiplier has an advantage with respect to integration, and reduces the accumulation time  $N_a$  required for higher precision operations. We designed the 4-bit up/down counter assuming the NEC 2.5 kA/cm<sup>2</sup> Nb/AIO<sub>x</sub>/Nb standard process. The activation function circuit can be assembled by using a comparator.

### ACKNOWLEDGMENT

The authors gratefully acknowledge the Superconductive Circuits Fabrication Team of SRL-ISTEC and NEC Corp. for the fabrication of Nb chips.

### REFERENCES

- [1] Y. Kondo and Y. Sawada, "Functional abilities of a stochastic logic neural network," *IEEE Trans. Neural Netw.*, vol. 3, pp. 434–443, May 1992.
- [2] K. Nemoto, M. Kinjo, S. Sato, and K. Nakajima, "A nonmonotonic neurochip using stochastic logic," in *Proc. 2001 Int. Symp. NOLTA, 2001*, pp. 605–608.
- [3] S. Sato, K. Nemoto, S. Akimoto, M. Kinjo, and K. Nakajima, "Implementation of a new neurochip using stochastic logic," *IEEE Trans. Neural Netw.*, vol. 14, pp. 1122–1127, Sep. 2003.
- [4] S. Akimoto, A. Momoi, S. Sato, and K. Nakajima, "Implementation of continuous-time dynamics on stochastic neurochip," *IEICE Trans. Fundamentals*, vol. E87-A, no. 9, pp. 2227–2232, Sep. 2004.
- [5] K. Nakajima, H. Mizusawa, H. Sugahara, and Y. Sawada, "Phase mode Josephson computer system," *IEEE Trans. Appl. Supercond.*, vol. 1, pp. 29–36, Mar. 1991.
- [6] 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. Superconduct.*, vol. 1, pp. 3–28, Mar. 1991.
- [7] E. S. Fang and T. V. Duzer, "A Josephson integrated circuit simulator (JSIM) for superconductive electronics application," in *Extended Abstracts Int. Superconductive Electronics Conf.*, 1989, pp. 407–410.
- [8] N. Mori, A. Akahori, T. Sato, N. Takeuchi, A. Fujimaki, and H. Hayakawa, "A new optimization procedure for single flux quantum circuits," *Physica C*, pt. 2, vol. 357–360, pp. 1557–1560, 2001.
- [9] J. H. Kang, J. X. Przybysz, S. S. Martinet, A. H. Worsham, D. L. Miller, and J. D. McCambridge, "3.69 GHz single flux quantum pseudorandom bit sequence generator fabricated with Nb/AIO<sub>x</sub>/Nb," *IEEE Trans. Appl. Supercond.*, vol. 7, pp. 2673–2676, Jun. 1997.
- [10] A. Akahori, N. Takeuchi, N. Mori, Y. Suzuki, F. Furuta, A. Fujimaki, and H. Hayakawa, "Demonstration of 17 GHz operation of M-code generator based on SFQ with resettable latch," *IEEE Trans. Appl. Supercond.*, vol. 11, pp. 521–524, Mar. 2001.
- [11] S. V. Polonsky, V. K. Semenov, and A. F. Kirichenko, "Single flux quantum B flip-flop and its possible applications," *IEEE Trans. Appl. Supercond.*, vol. 4, pp. 9–18, Mar. 1994.
- [12] T. Onomi, Y. Horima, M. Kobori, I. Shimizu, and K. Nakajima, "Implementation of phase-mode arithmetic elements for parallel signal processing," *IEEE Trans. Appl. Supercond.*, vol. 13, pp. 583–586, Jun. 2003.