Design of Binary Convolution Operation Circuit for Binarized Neural Networks Using Single-Flux-Quantum Circuit

Zongyuan Li, Yuki Yamanashi, *Member*, and Nobuyuki Yoshikawa, *Senior Member*

***Abstract*—We design a binary convolution operation circuit (BCOC) using a single-flux-quantum circuit for high-speed and energy-efficient neural network. The proposed circuit is used for binary convolution operations using a convolution kernel size of 3×3, which accelerates the forward propagation process of a binary neural network (BNN). We analyze the binary convolution process and propose a bisection method for optimization. The BCOC is designed with a gate-level pipeline architecture and uses the bisection method for reduced number of pipeline stages. Thus, the circuit area of the BCOC is reduced by approximately 50% compared with that of a BCOC without the bisection method. We design the BCOC with 3270 Josephson junctions using a 10 kA/cm^2 Nb process. The measurement results show that the BCOC can perform binary convolution operations with a kernel size of 3×3. Compared to a CMOS circuit, BCOC increases the power efficiency by 3.9 times. In future research, we will build up a library of BNNs based on SFQ circuits to simulate various BNN structures.**

***Index Terms*—Single-flux-quantum (SFQ) circuit, binary convolution, superconducting integrated circuit.**

I. Introduction

A SINGLE-flux-quantum (SFQ) circuit is a superconducting integrated circuit that uses flux quanta to transmit binary information; it has ultrafast speeds and very low power characteristics [1][2]. Therefore, SFQ circuit technology is considered as a promising solution for the post-Moore era. Several computing units [3, 4], processors [5, 6], and architectures [7] have been proposed based on SFQ circuit technology. Accordingly, we summarize some of the unique characteristics of the SFQ circuit technology as follows.

- **Gate-level pipelining:** Almost all commonly used SFQ logic gates are clock-driven. This means that each logic gate itself includes computational and delay-flip-flop (DFF) parts. Therefore, architects can naturally apply gate-level pipelining without any overheads. Ultrahigh throughput processing is thus possible using gate-level pipelining.
- **Lack of memory technology:** During the design of SFQ circuits, shift-register (SR) memory are used for data storage. Currently, there are no mature random-access memory (RAM) or off-chip memory technologies. This renders SFQ circuits inadequate for applications that require large amounts of storage.
- **Difficulty in achieving multibit parallel computing:** Owing to the current low integration levels of superconductor circuit fabrication processes, it is difficult for SFQ circuits to perform parallel calculations. SFQ circuits use Josephson transmission lines (JTLs) for data transmission. The area of every four JTLs is approximately equal to that of one logic gate. Therefore, an increase in the number of parallel bits increases the circuit area.

Owing to these characteristics, not all circuits are suitable for implementation using SFQ technology. Therefore, we focus on binary convolutional neural networks (BNNs). BNNs were originally proposed to solve the processing speed and memory consumption problems caused by the use of large numbers of floating-point operations. The original intent for the BNN was to compress a convolutional neural network (CNN) such that it could be operated on resource-constrained hardware [8]. As a neural network model, a BNN is suited for implementation with gate-level pipeline structures. In BNNs, all data are binarized to "1" and "-1," so there is no need to perform floating decimal operations [9]. Therefore, we believe that using SFQ circuit technology to design BNNs can improve performance for very high operating frequency and very low power consumption while fully utilizing the available features of the SFQ circuits.

BNN usually consists of buffer parts, convolutional parts, pooling parts and fully connected parts, as shown in Figure .1. The convolution part is one of the most important assembly parts, which directly affects the performance and operation speed of the BNNs. In this paper, we first investigate the action principle of binary convolution circuit which is one of the basic operational units for implementing a convolutional layer in a BNN. We design a binary convolution operator circuit (BCOC) and propose the bisection method to optimize the circuit, which can perform binary convolution operations with a 3×3 kernel at

The authors are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan (e-mail: li-zongyuan-hv@ynu.jp; yamanashi-yuki-kr@ynu.ac.jp; nyoshi@ynu.ac.jp).

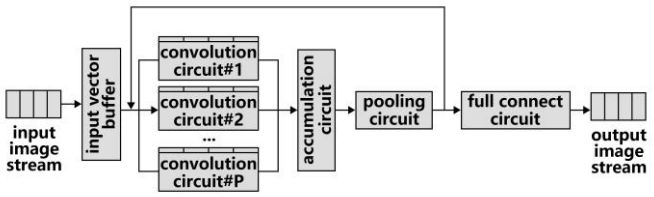


Fig. 1. Overview of the BNN. In this paper, we focus on the convolutional circuits

an operating frequency of 50 GHz. This demonstrates that we can perform binary convolution operations at a very high operating frequency, which can be used for accelerates the forward propagation process of a binary neural network (BNN). In future research, we will complete the circuit design for the other part of the BNNs. we will build up a library of BNNs based on SFQ circuits to simulate various BNN structures which includes different types of circuit parts for BNNs and find the most suitable BNN architecture for SFQ circuits.

II. Binary Convolutional Hardware Design

A. Binary Convolution

Given any two signals $f_1(t)$ and $f_2(t)$, the convolution operation is defined as

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau)\, d\tau\,. \tag{1}$$

From (1), convolution is a special type of integration over time τ [10]. For a signal, this means decomposition into a sum of infinitely many impulse signals. When processing an image using convolution, the input signal is a digital image, which is a two-dimensional discrete signal; hence, it is processed using two-dimensional convolution.

A single convolution operation is performed by multiplying and accumulating a convolution kernel of a certain size over a region of corresponding size in the feature map. The binary convolution operation is performed after binarizing the elements of the convolution kernel and input feature map using a specific function. The binarization value $B(x)$ for the analog input x is represented as

$$B(x) = \begin{cases} -1, & x \le 0 \\ +1, & x > 0 \end{cases}. \tag{2}$$

A single convolution operation between the kernel K of size $n \times n$ and feature map F can be represented as

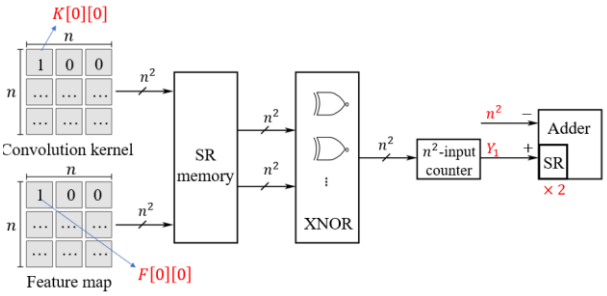
$$C = \sum_{i}^{n-1} \sum_{j}^{n-1} F[i][j] \times K[i][j], \tag{3}$$

where C is the convolution operation result, and i and j are the horizontal and vertical coordinates of the convolution kernel and feature map, respectively.

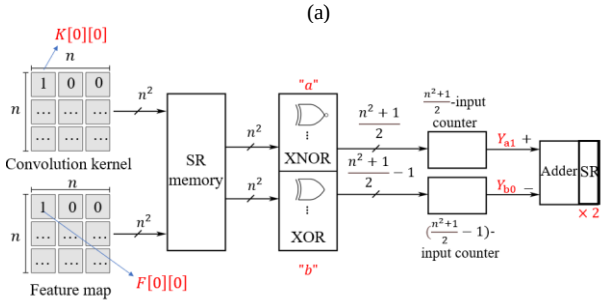
In hardware circuits, we usually use "0" to represent "-1" such that (2) can be written as

$$B_h(x) = \begin{cases} 0, & x \le 0 \\ +1, & x > 0 \end{cases}. \tag{4}$$

Meanwhile, binarization multiplication can be performed using an exclusive-NOR (XNOR) gate [11]. Then, (3) becomes

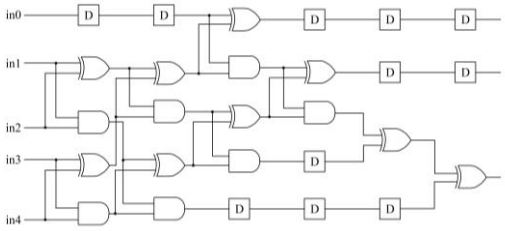


(a)

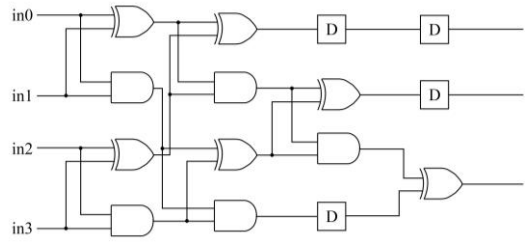


(b)

Fig. 2. Schematic of the single convolution operation (a) without and (b) with the bisection method.



(a) Five-input gate-level pipeline counter



(b) Four-input gate-level pipeline counter

Fig. 3. Block diagram of the (a) five-input and (b) four-input gate-level pipeline counters, where "D" is the delay-flip-flop (DFF). When the number of input parallel bits increases from 4 to 5, the number of stages increases from 4 to 6 and number of logic gates required increases from 15 to 27. Here, we ignore the number of transmission lines

$$Y_1 = \sum_{i}^{n-1} \sum_{j}^{n-1} B_h(F[i][j]) \odot B_h(K[i][j]), \tag{5}$$

where Y_1 is not the result of a single convolution operation but only the number of "1"s after the XNOR operation. The single convolution operation can thus be calculated as

$$C = Y_1 - Y_0, \tag{6}$$

where Y_0 is the number of "0"s after XNOR operation and is represented as

$$Y_0 = n^2 - Y_1. \tag{7}$$

From (6) and (7), we get

$$C = 2Y_1 - n^2. \tag{8}$$

TABLE I
COMPARISON OF THE TWO METHODS FOR A 3×3 CONVOLUTION KERNEL

Methods	Number of gates in counter	Number of gates in adder	Pipeline stages in counter	Pipeline stages in adder
Without bisection method	67	51	9	6
With bisection method	42	21	6	4

B. *Binary Convolutional Hardware Design*

As shown in (8), the main task of binary convolution is to process the results of the XNOR operation. The general method of handling this is to use a gate-level pipeline counter to count the number of "1"s in the result of the XNOR operation to replace the accumulation process, as shown in Fig. 2(a). Then, (8) is calculated using the shift and gate-level pipeline adders.

As described in the introduction, the area and number of stages of the gate-level pipeline circuitry increase with the number of parallel bits, which is shown in Fig. 3. Thus, we can reduce the number of gates required and increase the operational speed by reducing the number of parallel bits. For this purpose, we use the bisection method to process the results of the XNOR operations, which in turn reduce the number of parallel bits of the counter and adder.

The bisection method shown in Fig. 2(b) splits the results of the XNOR operation into two parts as "a" and "b". Because the number of elements in the convolution kernel is mostly odd (commonly used convolution kernel sizes are 3×3, 5×5, and 7×7), we add a "0" to the XNOR operation result to make it even, which is placed in part "b". Through this step, the number of elements in the two parts are equalized, which means that Y_{amax} and Y_{bmax} are represented as

$$Y_{amax} = Y_{bmax} = \frac{n^2+1}{2}. \tag{9}$$

Then, we use

$$Y_a = Y_{a1} - (Y_{amax} - Y_{a1}), \tag{10}$$

and

$$Y_b = -Y_{b0} + (Y_{amax} - Y_{a1}), \tag{11}$$

to calculate the results of the two parts, where Y_a is the result of part "a" and Y_b is the result of part "b". Here, Y_{a1} is the number of "1"s after the XNOR operation in part "a", and Y_{b0} is the number of "0"s after the XNOR operation in part "b". Further, the single convolution operation result can be calculated by

$$C = Y_a + Y_b = 2(Y_{a1} - Y_{b0}). \tag{12}$$

When not using the bisection method, we need to use an n^2-input counter to count the number of "1"s after the XNOR operation. When using the bisection method, we need to count the number of "1"s in part Y_a after the XNOR operation using a $\frac{n^2+1}{2}$-input and count the number of "0"s in part Y_b after the XNOR operation using a $(\frac{n^2+1}{2}-1)$-input counter. A comparison of the two methods is presented in Table I. This shows that using the bisection method can be effective at reducing the required hardware resources and number of pipeline stages. We believe that this method is applicable to any BNN convolution operation circuit.

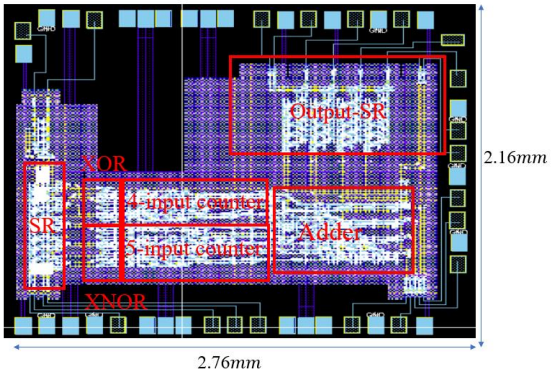


Fig. 4. Layout of the BCOC. The four-input counter in the upper part is used to calculate the number of "0"s in part "b". The five-input counter in the lower part is used to calculate the number of "1"s in part "a".

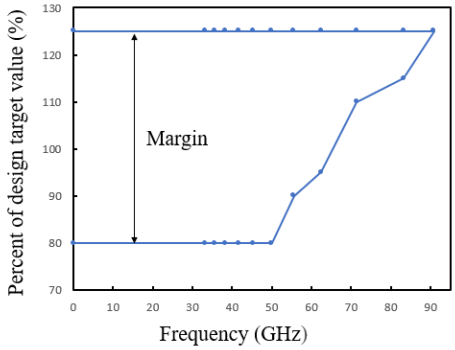


Fig. 5. Simulated dependence of the bias voltage margin on the input frequency; the designed bias voltage is 2.5 mV

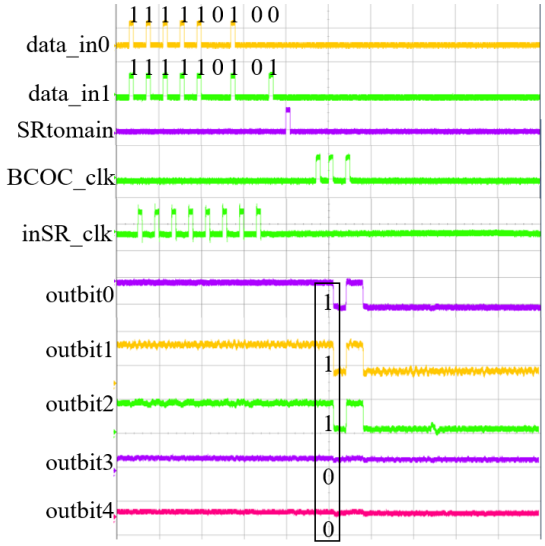


Fig. 6. Measurement results of the BCOC with low frequency (100 kHz). We input "111110101" and "111110100" to obtain the output $(00111)_2 = 7$. Here inSR_clk is the clock signal that writes data to the SR. SRtomain is the clock signal that reads data from the SR to the BCOC. BCOC_clk is the main clock signal for the BCOC circuit. Each BCOC_clk signal can perform one convolution operation.

TABLE II POWER CONSUMPTION AND SCALABILITY OF THE BCOC CIRCUIT WITH DIFFERENT CONVOLUTION KERNEL SIZE

Convolution kernel size	3×3	5×5	7×7
The number of SFQ gates	64	278	591
Dynamic power consumption (mW)	0.033	0.143	0.305
Static power consumption (mW)	0.8	3.475	7.387

TABLE III COMPARISON WITH CMOS IMPLEMENTATION

Methods	This paper	FBNA-CPU	FBNA-GPU	FINN-FPGA
Platform	SFQ	Intel 6700k	Nvidia GTX1070	Xilinx zc706
Power efficiency (GOPs/W)	Peak:2700[a]	0.76	9.25	684.7
	Aver-age:1080[b]			

[a]Here we consider a cooling cost of 400 times the total power consumption [16].

[b]Due to the lack of random storage in the SFQ circuit, typical processor performance is only about 40 percent of the peak.[7]

III. MEASUREMENT AND COMPARISON

We designed a binary convolution operation circuit (BCOC) using an SFQ circuit and the bisection method. This BCOC was used to perform binary convolutional operations with a kernel of size 3×3. All circuit components were designed using the National Institute of Advanced Industrial Science and Technology 10 kA/cm^2 Nb advanced process 2 (ADP2) [12, 13] and its cell library [14, 15].

The circuit used for measurement of the BCOC is shown in Fig. 4. The number of Josephson junctions in the BCOC is 3270. Because the result of the XOR gate is the opposite of the XNOR gate, we use the XOR gate instead of the XNOR gate in part Y_b . Thus, "1" is used to represent "0" in the counting circuit of part Y_b. The result of part b is calculated by taking the complement before addition operation.

Fig. 5. shows the simulated dependence of the bias voltage margin on the input frequency. Through simulation, we have successfully verified that the BCOC circuit can perform binary convolution operations at an operating frequency of 50 GHz. The measurement results with a low operation frequency are presented in Fig. 6. We input "111110101" and "111110100" as the convolution kernel and the feature input. Here "1" indicates "+1" and "0" represents "-1" in (2). We obtain the results $(00111)_2$, which is "7". The test results prove the correct operation of the circuit. Because of a design error in the outSR, the results of the high-speed test could not be read out. We will continue with the high-speed measurements of this circuit in future studies.

Table 2 shows the power consumption and scalability of the BCOC circuit. the power consumption of SFQ circuits typically includes static power consumption, dynamic power consumption. Methods to further reduce power consumption include the use of PTL for data transfer and the use of new techniques such as ERSFQ [17]. Table 3 shows the energy efficiency of the circuit. Compared to CMOS technology (CPU [18], GPU [18], FPGA [19]), our BCOC circuit is 3.9 times higher power efficiency. Even taking into account the lack of random storage in SFQ, a 1.6x improvement can be reached.

IV. CONCLUSION

We analyze the mechanism principle of binary convolution and prove that the binary convolution circuit can be optimized using the bisection method. Further, we designed and simulated a BCOC consisting of 3270 Josephson junctions. The BCOC was used to perform binary convolution operations at an operating frequency of 50 GHz. We have completed a low-speed test of the circuit and analyzed the scalability of the circuit. Compared to the CMOS circuit, the designed SFQ BCOC increases the power efficiency by 3.9 times. This is the most basic operational unit of the BNN and is the first step in our design of the SFQ-BNN. In future research, we intend to test and optimize the proposed circuit and as well as design the convolution layer of the SFQ-BNN based on the BCOC circuit. We will build up a library of BNNs based on SFQ circuits, in which BCOC is the first circuit unit.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP 18K04280. The circuits were fabricated in the clean room for analog-digital superconductivity (CRAVITY) of National Institute of Advanced Industrial Science and Technology (AIST) with the advanced process 2 (ADP2).

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 Transactions on Applied Superconductivity*, vol. 1, no. 1, pp. 3–28, March 1991.

[2] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 760–769, Jun. 2011.

[3] T. Kato et al., "60-GHz demonstration of an SFQ half-precision bit-serial floating-point adder using 10 kA/cm^2 Nb process," *IEEE 14th International Superconductive Electronics Conference-Cambridge, MA, USA*, pp. 56-58, 2013.

[4] I. Nagaoka, M. Tanaka, K. Inoue, and A. Fujimaki, "A 48GHz 5.6mW gate-level-pipelined multiplier using single-flux quantum logic," *2019 IEEE International Solid- State Circuits Conference*, pp. 460–462. 2019

[5] F. Ke et al., "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.

[6] R. Kashima et al., "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.

[7] K. Ishida et al., "SuperNPU: An Extremely Fast Neural Processing Unit Using Superconducting Logic Devices," *2020 53rd Annual IEEE/ACM*

International Symposium on Microarchitecture (MICRO), pp. 58-72, 2020.

[8] R. Andri, L. Cavigelli, D. Rossi, L. Benini, "YodaNN: An Ultra-Low Power Convolutional Neural Network Accelerator Based on Binary Weights," *IEEE Computer Society Annual Symposium on VLSI*, pp. 236-241, 2016.

[9] M. Courbariaux et al., "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1.," arXiv, arXiv:1602.02830, 2016.

[10] Y.S. Zhao et al., "Convolutional Integration based on Time and Frequency Domain," *Communications Technology* vol. 43, no, 11, pp. 165-168, 2010

[11] M. Rastegari, et al. "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks" *Computer Vision – ECCV 2016,* pp. 525–542, Oct. 2016.

[12] S. Nagasawa et al., "Nb 9-layer fabrication process for superconducting large-scale SFQ circuits and its process evaluation," *IEICE Trans. Electron.*, vol. E97-C, no. 3, pp. 132–140, Mar. 2014.

[13] M. Hidaka and S. Nagasawa, "Fabrication Process for Superconducting Circuits," IEICE Trans. Electron., vol. E104-C, no. 9, pp. 405–410, Sep. 2021.

[14] H. Akaike et al., "Design of single flux quantum cells for a 10-Nb-layer process," *Physica C*, vol. 469, no. 15–20, pp. 1670–1673, Oct. 2009.

[15] Y. Yamanashi et al., "100 GHz demonstrations based on the single-flux quantum cell library for the 10 kA/cm2Nb multi-layer process," *IEICE Trans. Electron.*, vol. E93-C, no. 4, pp. 440–444, Apr. 2010.

[16] S. Holmes, "Energy-Efficient Superconducting Computing—Power Budgets and Requirements," *IEEE Trans. Appl. Supercond.*, vol. 23, no. 3, Jun. 2013, Art. no. 1701610.

[17] D. Kirichenko, et al., "Zero static power dissipation biasing of RSFQ circuits," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 776–779, Jun. 2011.

[18] P. Guo, et al., "FBNA: A Fully Binarized Neural Network Accelerator," *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pp.51–513, 2018

[19] Y. Umuroglu, et al., "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference", *the 25th International Symposium on Field-Programmable Gate Arrays*, pp. 65-74, 2017.