The published version of this manuscript appeared in *IEEE Transactions on Applied Superconductivity*
19, No. 3, Part 1, 634 - 639 (2009)

Design, Implementation, and On-Chip High-Speed Test of an SFQ Half-Precision Floating-Point Adder

Heejoung Park, Yuki Yamanashi, Kazuhiro Taketomi, Nobuyuki Yoshikawa, Masamitsu Tanaka,
Koji Obata, Yuki Ito, Akira Fujimaki, Naofumi Takagi, Kazuyoshi Takagi, Shuichi Nagasawa

***Abstract*—We are developing a large-scale reconfigurable data-path (LSRDP) based on single-flux-quantum (SFQ) circuits to establish a fundamental technology for future high-performance computing systems. In the LSRDP, an SFQ floating-point adder (FPA) is one of the main circuit blocks as well as one of the most complicated circuit blocks. In the present paper, we designed and implemented an SFQ half-precision FPA and carried out on-chip high-speed tests. The data format of the half-decision FPA obeys the IEEE standard, in which two input data streams, an 11-bit significand and a 6-bit sign/exponent, are processed bit-serially. Floating-point addition is performed in three steps: (1) alignment and rounding of significands, (2) addition/subtraction of the significands, and (3) normalization of the result. We implemented an SFQ half-precision FPA using the SRL 2.5 kA/cm^2 niobium standard process. The size, power consumption, and total junction number are 5.86 mm × 5.72 mm, 3.5 mW, and 10,224, respectively. The simulated DC bias margin is ±20% at 20 GHz, which corresponds to the performance of 1.67 GFLOPS. We successfully confirmed the correct operation of the FPA, except for a read-out circuit for the significand, at 24 GHz by on-chip high-speed tests.**

***Index Terms*—SFQ circuits, superconducting integrated circuits, LSRDP, floating-point adder, shifter, normalizer**

I. Introduction

We are developing a large-scale reconfigurable data-path (LSRDP) based on single-flux-quantum (SFQ) circuits [1] to establish a fundamental technology for future high-performance computing systems. The LSRDP is composed of a large number of floating-point units (FPUs) connected by reconfigurable routing networks, as shown in Fig. 1 [2]. In the LSRDP, reputation loops in a program are directly mapped and calculated efficiently. The main advantage of the LSRDP is the reduction of the memory-wall problem in high-performance computing systems. The memory access rate is considerably reduced because data are directly transferred between FPUs without memory accesses. In the LSRDP, the SFQ floating-point adder (FPA) is one of the main circuit blocks, as well as one of the most complicated circuit blocks. In the present paper, we designed and implemented an SFQ half-precision FPA and carried out on-chip high-speed tests.

II. Floating-Point Numbers

Floating-point numbers are widely used in many scientific and engineering fields because the floating-point numbers can represent a wide range of values that are required in numerical calculation. However, the circuit implementation of floating-point numbers is very complex compared with the circuit implementation of fixed-point numbers. Floating-point numbers are generally given as follows:

$$\left(-1\right)^S \times F \times 2^E,$$

where S is the sign, F is a significand, and E is an exponent. There are many representations of floating-point numbers. In the present paper, we follow the IEEE standard [3], which is currently used in many floating-point processors. The formats

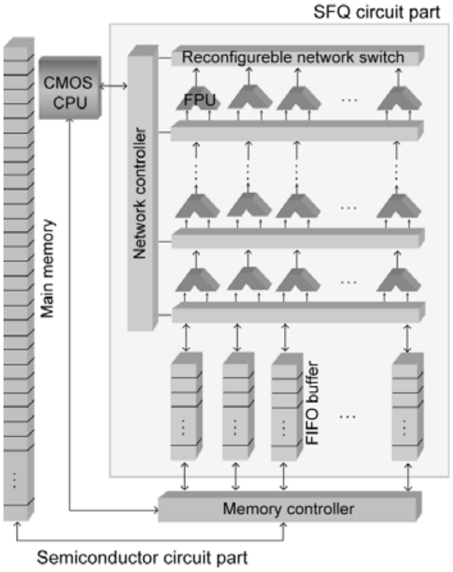


Fig. 1. Block diagram of a large-scale reconfigurable data path (LRDP) based on SFQ circuits.

Manuscript received August 26, 2008. This research was supported by CREST, Japan Science and Technology Agency.

H. Park, Y. Yamanashi, K. Taketomi and N. Yoshikawa are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama, Japan (phone: +81-45-339-4269; fax: +81-45-339-4269; e-mail: park@yoshilab.dnj.ynu.ac.jp).

M. Tanaka, K. Obata, Y. Ito, N. Takagi and K. Takagi is with the Department of Information Engineering, Nagoya University, Nagoya, 464-8603 Japan

A. Fujimaki is with the Department of Quantum Engineering, Nagoya University, Nagoya, 464-8603 Japan

S. Nagasawa is with Superconductivity Research Laboratory, Tsukuba, Japan.

TABLE I Bit Lengths of Floating-Point Numbers

	Sign (n_{S})	Exponent (n_{E})	Significand (n_{F})
Half-precision (16 bit)	1 bit	5 bit	11 bit
Single-precision (32 bit)	1 bit	8 bit	24 bit
Double-precision (64bit)	1 bit	11 bit	53 bit

*n_{S}: bit length of the sign
*n_{E}: bit length of the exponent
*n_{F}: bit length of the significand

TABLE II Effective Operation of Adder/Subtractor

Floating-point operation	Sign of numbers	Effective operation
ADD	Equal	ADD
	Different	SUBTRACT
SUBTRACT	Equal	SUBTRACT
	Different	ADD

of floating-point numbers are shown in Table I. The sign S is one bit, which is the sign of the floating-point number, where S = 1 denotes a negative sign. The significand F is the magnitude of the significand and includes a hidden bit. The exponent E is biased by $2^{n_{\mathrm{E}}-1}-1$ to simplify the calculation.

As shown in Fig. 1, a large number of FPUs are required for the LSRDP. We designed SFQ FPUs based on bit-serial architecture to implement circuits in a small area [4]. However, if whole data, including the sign, the significand, and the exponent, are input serially to the FPU, the performance is deteriorated because the data input time is proportional to the data length. To enhance the performance, we use two bit-serial data-paths. Fig. 2 shows the data format of the SFQ FPUs designed in the present study. We assigned one data-path for the significand and another for the sign and the exponent. The data are input to each circuit unit serially from the least significant bit (LSB) to the most significant bit (MSB).

III. Design of the SFQ Floating-Point Adder

A. Optimization of the Pipeline and the Performance

The FPA is the most complicated circuit blocks in an FPU. In general, floating-point addition is performed by several steps as follows [3]:

(1) **Subtract exponents.** The difference of two exponents is calculated.

(2) **Align significands.** The significand of the smaller number is shifted right by the difference of two exponents, so that its exponent corresponds to the larger exponent.

(3) **Addition (or subtraction) of two significands.** This calculation is a signed addition. The effective operation (add or subtract) is determined by the floating-point operation (ADD or SUBTRACT) and the signs of numbers to be calculated, as shown in Table II.

(4) **Produce the sign of the result.** The sign of the result depends on the sign of the numbers to be calculated, the operation, and the relative magnitude of the numbers. In general, the sign of the number having the larger exponent determines the sign of the result in the ADD operation. However, when the values of the two exponents are the same, the sign of the calculation results of the significand should be checked.

(5) **Normalization of the result.** If the significand of the result is not in normalized scientific notation, the significand is shifted to correspond to the normalized form, and the exponent is adjusted. (The MSB must be 1.)

(6) **Round.** The result is rounded to fit the data format.

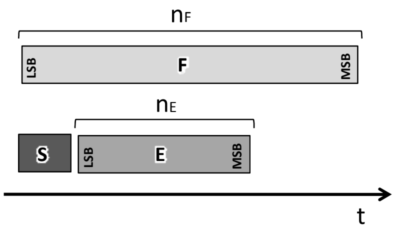


Fig. 2. Data format of the designed SFQ FPU, where S is the sign, F is the significand, and E is the exponent of a floating-point number.

If we design a bit-serial FPA according to the above calculation, the performance becomes $f/2n_{\mathrm{F}}$, where f is the clock frequency, because the bit length of the calculation result becomes twice that of the significand. In order to enhance the performance and simplify the circuit complexity, the rounding is restricted to “round-toward-0” and is carried out during the “align significands” step.

The operation of the adder/subtractor is predicted during the “align significands” step by comparing the magnitude of the significands of two numbers, so that the calculation result of the adder/subtractor is always positive.

Fig. 3 shows a block diagram of the SFQ bit-serial FPA. The main circuit components are shifters, an adder/subtractor, normalizers, and a controller. In the proposed design, the calculation is executed by three pipeline stages: (1) aligning and rounding the significands and prediction of the operation of the adder/subtractor, (2) addition (or subtraction), and (3) normalization. In the following subsection, we will present, in detail, the design of the circuit components for each pipeline stage.

B. Design of the Component Circuits

1) Aligning and rounding of significands and prediction of the sign of the adder/subtractor operation

First, the difference of two exponents, $E_{\mathrm{A}}-E_{\mathrm{B}}$, is calculated in order to align and round the significands. At the same time, the two significands are stored to each shifter. The difference of the two exponents is then sent to the shifters after being decoded to the control signals for the shifters. The decoder, the structure of which is similar to that of the decoder of SFQ memories, consists of a serial-parallel converter and tree-shaped binary switches composed of non-destructive read-out complementary flip-flops (NDROCs) [5].

The prediction of the sign of the adder/subtractor operation is achieved by comparing the magnitude of the significands and the exponents. According to the prediction of the sign, the adder/subtractor operation is determined so that the result of the calculation is always positive. To determine the sign, we examined every combination of magnitude of the significands and exponents, as listed in Table III. For example, in the case of

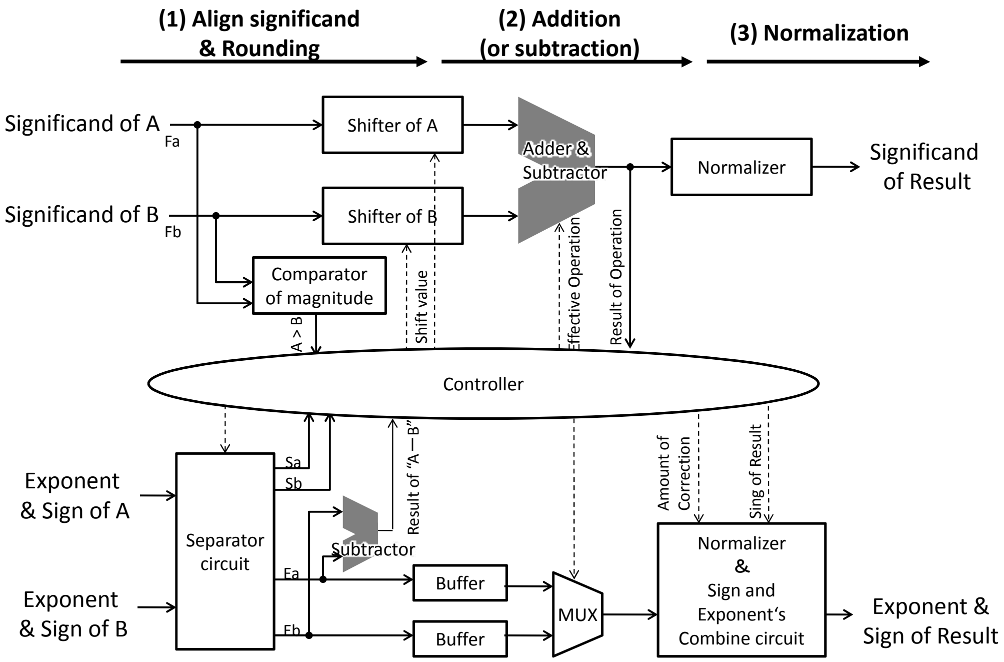


Fig. 3. Block diagram of a half-precision SFQ bit-serial floating-point adder.

$E_A = E_B$, the subtraction $(A - B)$ is calculated as $A + \bar{B} + 1$ when the prediction of the sign is positive and as $\bar{A} + B + 1$ when the prediction of the sign is negative. Note that when E_A is not equal to E_B, the subtraction $(A - B)$ is calculated by $A + \bar{B}$ or $\bar{A} + B$, because +1 for the smaller significand is rounded in the shift operation.

a) Shifter

For the shift operation of the significands, specific shifters are designed to enhance the performance. In this shifter, shifting and rounding operations are performed at the same time. Fig. 4(a) shows a block diagram of the 4-bit version of the shifter, which consists of a shift register (SR) composed of D2 flip-flops (D2FFs), confluence buffers (CBs) to combine output data, non-destructive read-out flip-flops (NDROs) to control the shift register, and delay flip-flops (DFFs) to send a reset signal to NDROs.

The D2FF has one data input ("Din"), two clock inputs ("Clk1" and "Clk2"), and two data outputs ("Dout1" and "Dout2") [6]. If an input SFQ pulse is applied to the D2FF, its internal state is set to "1". In this case, when an SFQ pulse is applied to "Clk1" ("Clk2"), an SFQ pulse is output from "Dout1" ("Dout2"), and the internal state is reset to "0". In the shift register composed of the D2FFs, as shown in Fig. 4(a), if clock pulses are applied to its "Clk2" terminal, it acts as a simple shift register. On the other hand, if clock pulses are applied to the "Clk1" terminal of one of the D2FFs, the data on the D2FF are extracted from "Dout1" and sent to "Data_out".

In the operation of the shifter shown in Fig. 4, the data are first input to the D2FF-based shift register from the LSB to the MSB. After the data are stored, a control signal for the shifter "Shift_*i*" is sent to one of the NDROs to enable it, and the output position of the shift register is selected. After that, when clock pulses are input to "Clk_in", the data are shifted in the shift register and output from the D2FF that is designated by the enabled NDRO. A timing chart of the shifter is shown in Fig. 4(b).

2) Addition (or subtraction)

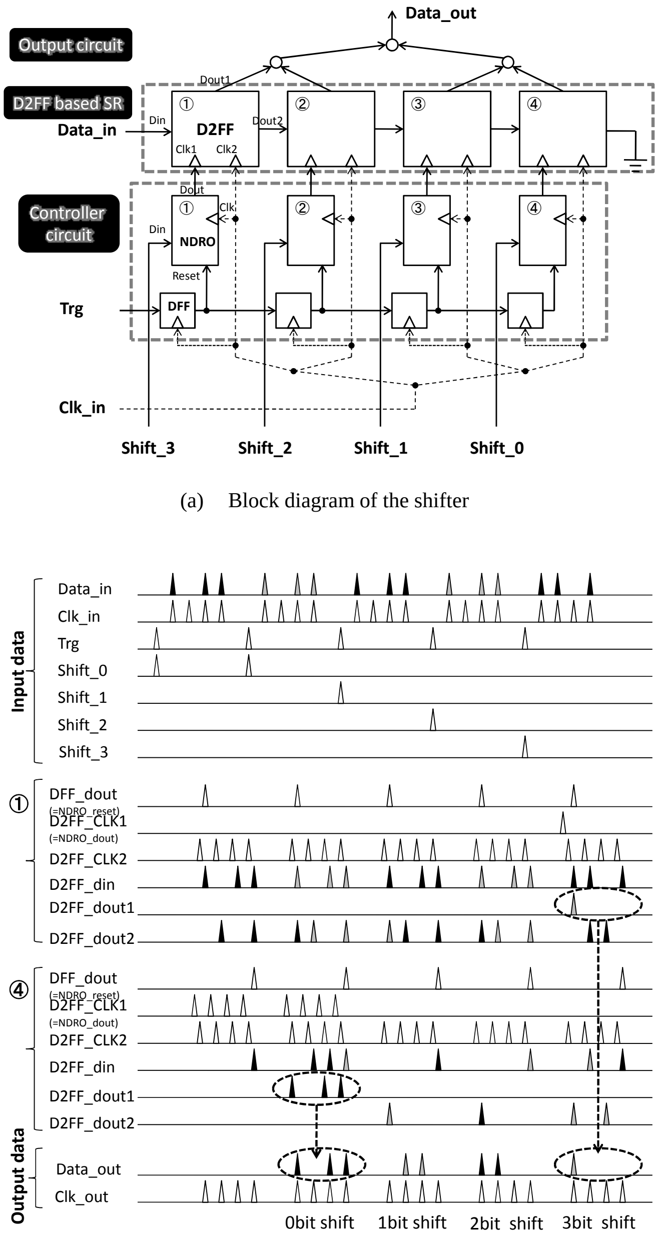
In this phase, the shifted significands are added (or subtracted) bit-serially according to the operation of the adder/subtractor, and the result is output to the normalizer. We used a bit-serial adder based on the state transition to increase the throughput of the operation, in which a feedback loop in the bit-serial adder is eliminated by translating the calculation to the state transition stored in an NDRO [6]. The subtraction is also carried out by using the adder after negating one of the input data and adding "1" to the datum.

During the addition and subtraction, the larger exponent is selected by a multiplexer (MUX), and the sign of the result is calculated.

TABLE III PREDICTION OF THE SIGN OF THE ADDER/SUBTRACTOR OPERATION AND OPERATION OF THE ADDER/SUBTRACTOR

Effective operation	Magnitude correlation of the exponents	Magnitude correlation of the significands	Prediction of the sign for the adder/subtractor	Operation of the adder/subtractor
ADD	*	*	+	$A + B$
SUBSTRACT (A − B)	$E_A > E_B$	*	+	$A + \bar{B}$
	$E_A < E_B$	*	−	$\bar{A} + B$
	$E_A = E_B$	$F_A \geq F_B$	+	$A + \bar{B} + 1$
	$E_A = E_B$	$F_A < F_B$	−	$\bar{A} + B + 1$

"*" means this situation does not affect the adder/subtractor operation. $\bar{A}$ denotes the complement of A.



(a) Block diagram of the shifter

(b) Timing chart of the shifter. The data are first input to the D2FF-based shift register from the LSB to the MSB. After the data are stored, a control signal for the shifter "Shift_*i*" is sent to one of the NDROs to enable it, and the output position of the shift register is selected. When clock pulses are applied to the shifter, the enabled NDRO sends clock pulses to "Clk1" of the D2FF just prior to the arrival of the clock pulses at "Clk2", while the other D2FF only obtains clock pulses at "Clk2". As a result, the data on the shift register are extracted through the D2FF designated by the enabled NDRO, while the data on the other D2FFs are just shifted rightward.

Fig. 4. Block diagram of the shifter and its timing chart.

3) Normalization

After the addition (or subtraction) of the significands, the format of the result is checked to determine whether it is normalized. If the result is not normalized, the position of "1" in the significand is checked for the normalization. To enhance the performance of the normalization, two specified normalizers for the significand and the exponent are used individually.

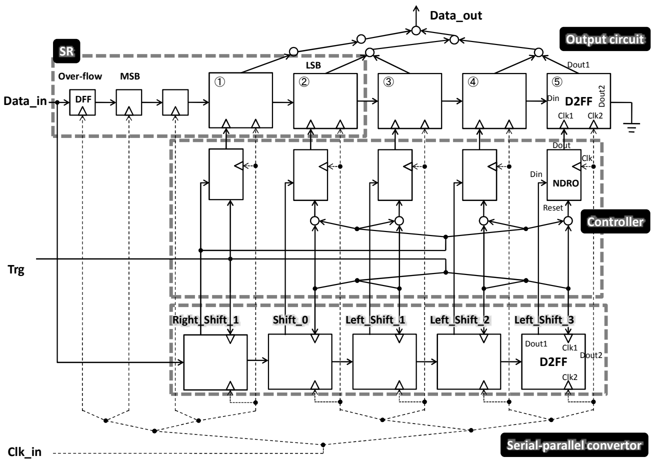


Fig. 5. Block diagram of the normalizer for the significand.

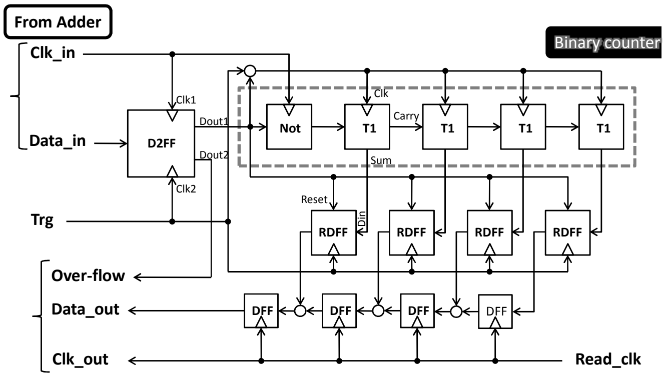


Fig. 6. Block diagram of the normalizer for the exponent.

a) Normalizer for the significand

The operation principle of the shifter described above is also applied to the design of the normalizer for the significand. The normalizer must perform not only right-shift operations, but also left-shift operations, because both overflow and underflow occur in calculations. Fig. 5 shows a block diagram of the normalizer of a 4-bit version. This normalizer can achieve shift operations ranging from a 4-bit left shift to a one-bit right shift. The normalizer consists of an 8-bit D2FF-based shift register (SR), the highest three bits of which are composed of DFFs, a 4-bit D2FF-based serial-parallel converter, and an NDRO array.

In the operation of the normalizer, 5-bit input data, including an overflow bit, are first stored on the D2FF-based shift register. The initial positions of the LSB, the MSB, and the overflow bit of the input data are indicated in Fig. 5. At the same time, a copy of the input data is also stored on the D2FF-based serial-parallel converter. A "Trg" signal is then applied to the "reset" terminals of NDROs to reset the D2FFs and to the "Clk1" terminals of the D2FFs of the serial-parallel converter to send the location of "1" of the input data to the NDROs and to enable them. After that, clock pulses are input to the normalizer. Since

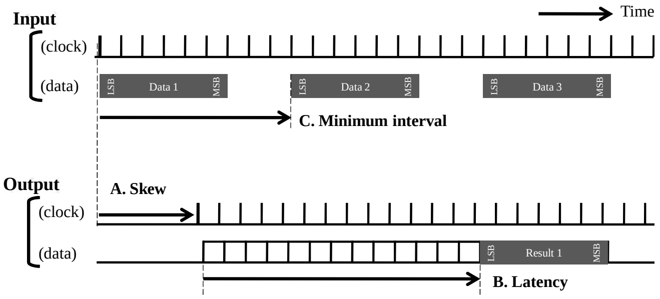


Fig. 7. Definition of timing parameters of processing units based on bit-serial architecture.
A. Skew: In the concurrent-flow or data-followed-clock clocking, clock signals propagate with data signals. This means that the clock signals arrive at the input and output terminals of the processing unit at different times.
B. Latency: Number of required clocks to calculate the results, i.e., the number of latch-gates (pipeline stages) in a data-path.
C. Minimum interval: Minimum clock cycles between input data. The minimum interval is necessary to prevent the data corrosion in a processing unit. The reciprocal of the minimum interval corresponds to the maximum throughput.

TABLE IV
ESTIMATION OF THE TIMING PARAMETERS AND THE JUNCTION NUMBER OF THE SFQ BIT-SERIAL FPA

	Minimum interval	Latency	# of JJs (not including wiring costs)
Half-precision	13	23	6K
Single-precision	26	49	12K
Double-precision	55	107	24K
Theoretical prediction	(n_F + 2)	(2 n_F + 2)	

The minimum interval and the latency are measured by clock cycles.

the D2FFs in the shift register that is clocked by the enabled NDROs extract the data into “Data_out”, the resultant output data have the normalized form. For example, if $(D_{\text{overflow}}\ D_4.\ D_3\ D_2\ D_1\ D_0)$ = (00.101) is applied to the normalizer, “1” is loaded on the D2FFs of the serial-parallel converter designated by “Left_Shift_1“ and “Left_Shift_3”, and sent to the “Clk1” terminals of the third and fifth D2FFs of the shift register. As a result, when clock pulses are input, (01.010) is output form the third D2FF of the shift register after being shifted left by one bit.

a) Normalizer for the exponent

Fig. 6 shows a block diagram of the normalizer for the exponent. The normalizer is composed of a D2FF, a not gate (Not), a binary counter composed of resettable toggle flip-flops (T1s), resettable delay flip-flops (RDFFs), and DFFs.

Initially, the result of the addition (or subtraction) is sent to the binary counter from the LSB to the MSB after being inverted by the Not gate to count the bit number of “0” in the input data. Since the last inputs of “1” in the data resets the counter, the resultant value in the counter corresponds to the number of final inputs of “0”. When a “Trig” pulse is applied to the normalizer, the counted results are read out by the RDFFs, and then sent to the DFFs. The D2FF is used to detect the overflow in the result.

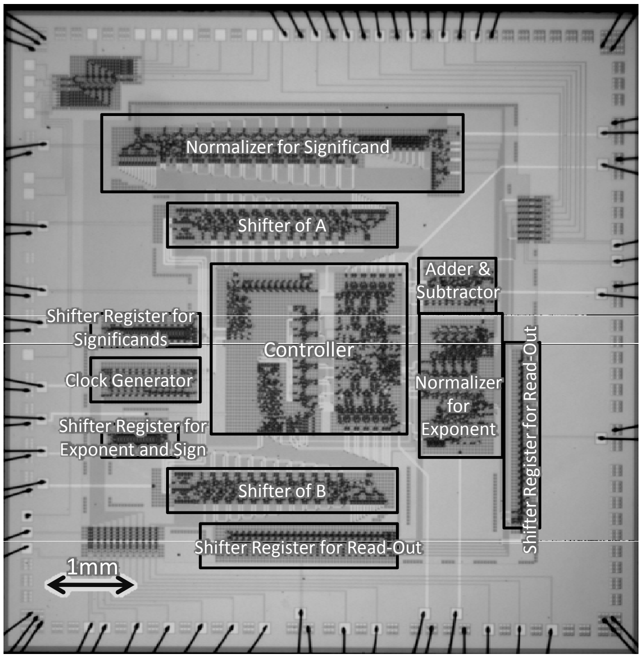

Normalizer for Significand
Shifter of A
Adder & Subtractor
Shifter Register for Significands
Controller
Clock Generator
Normalizer for Exponent
Shifter Register for Exponent and Sign
Shifter Register for Read-Out
Shifter of B
Shifter Register for Read-Out
1mm

Fig. 8. Chip photograph of the SFQ half-precision FPA

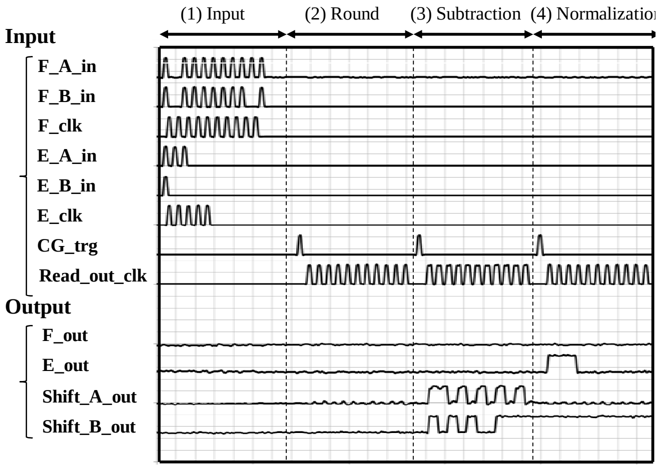


Fig. 9. Example of on-chip high-speed test results at 20 GHz. The upper eight waveforms are input signals, where the rising edges correspond to the input SFQ pulses. The lower four waveforms are output signals, where transitions are the output of SFQ pulses. In this test, a floating-point addition, -1.11111111101 x 2^{00011} + (-1.10111111101 x 2^{00000}) = -1.00010111100 x 2^{00100} is calculated. First, two 11-bit significands, F_A = (11111111101) and F_B = (10111111101), and two 6-bit exponents with signs, E_A = (000111) and E_B = (000001), are input from the LSB to the MSB at low speed. Then, three “CG_trg” pulses for three pipeline stages are input to a clock generator (CG), which provides high-speed clock pulses to the FPA. Finally, the calculation results of the significand, the exponent, and the two shifters, “F_out”, “E_out”, “Shift_A_out”, and “Shift_B_out” are read out by inputting clock pulses “Read_out_clk” for reading out the data. Except for “F_out”, correct results are obtained for Shift_A_out = (11111111101), Shift_B_out = (00010111111), and E_out = (001001).

IV. ESTIMATION OF THE PERFORMANCE AND SIZE OF THE SFQ BIT-SERIAL FPA

In the estimation of the performance of the bit-serial FPA, three timing parameters are considered: skew, latency, and

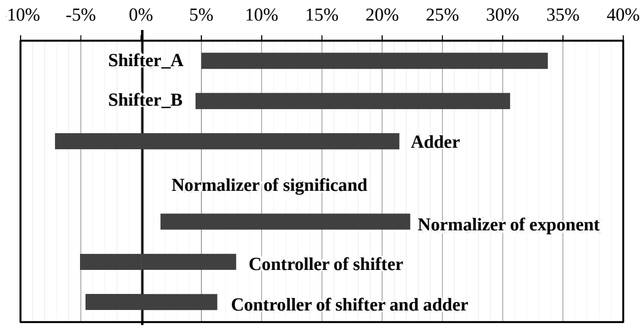


Fig. 10. DC bias margins of each circuit component at 20 GHz.

minimum interval. Fig. 7 illustrates the definitions of the three parameters. In the LSRDP, the minimum interval and the latency are very important parameters. The minimum interval determines the throughput of the system, and the latency corresponds to the number of the pipelined stage in each processing unit. The skew can be adjusted by inserting delays. Table IV lists the estimation of the two timing parameters and the junction number of SFQ bit-serial FPAs, including larger systems. The designed FPA has three pipeline stages, and their minimum intervals are estimated to be n_F + 2, where "2" is required for the extra clock cycles due to the calculation of the signed bit and the over-flow bit. The junction number is approximately proportional to the bit-length of the data.

V. Implementation and High-Speed Tests of SFQ Half-Precision Bit-Serial FPA

We have implemented an SFQ half-precision FPA using the SRL 2.5 kA/cm^2 niobium standard process [8] and the CONNECT cell library [9]. Fig. 8 shows a chip photograph of the SFQ half-precision FPA with a high-speed test system, which is implemented on an 8 mm die [10]. The size, power consumption, and total junction number are 5.86 mm × 5.72 mm, 3.5 mW, and 10,224 (without peripheral circuits for testing), respectively. The simulated DC bias margin is ±20% at 20 GHz, which corresponds to the performance of 1.67 GFLOPS (giga floating-point operations per second).

We have successfully confirmed full operations of the FPA at high speed up to 24 GHz, except the output of the significand ("F_out"). Fig. 9 shows an example of test results. We observed no output signal from "F_out" in the tested chips. Note that the test sequence in Fig. 9 confirms the correct function of almost all circuit components, including the shifters, the adder/subtractor, the controller, and the normalizer for the exponents. We believe that the circuit design of the normalizer for significands has no problem, because its design is almost the same as that of the shifter. The most probable reason for the malfunction in "F_out" is thought to originate from the long passive transmission line (PTL) wiring (14 mm in this design) between the normalizer for the significands and the shift register for reading out results.

Fig. 10 shows the DC bias margins of each circuit component at 20 GHz. An overlap of the DC bias margins is very small, approximately 2%. One of the main reasons for this is the discrepancy in the timing parameters of logic gates between simulated and fabricated circuits. Another reason is thought to be the large magnetic field induced by the large supply current and/or ground return current. These are important considerations for large-scale SFQ digital integrated circuits at present.

VI. Conclusion

We designed and implemented an SFQ half-precision bit-serial floating-point adder (FPA), which is one of main building blocks of SFQ large-scale reconfigurable data-paths (LSRDPs). In the design of the shifter and the normalizer of the FPA, we have proposed a novel architecture to increase the throughput. The resultant throughput is estimated to be n_{f} + 2, where n_{f} is the bit-length of the significand of the input floating-point number. We have successfully tested the full operation of the FPA at high speed, except for a read-out for the significand. Its maximum operation frequency was 24 GHz.

Acknowledgment

The authors would like to thank the CONNECT members at SRL-ISTEC, NICT, Nagoya University, and Yokohama National University.

References

[1] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction digital technology for sub-terahertz-clock-frequency digital systems," *IEEE Trans. Appl. Supercond.*, vol. 1, pp. 3–28, Mar. 1991.

[2] N. Takagi, K. Murakami, A. Fujimaki, N. Yosikawa, K. Inoue and H. Honda, "Proposal of a desk-side supercomputer with reconfigurable data-paths using rapid single-flux-quantum circuits," *IEICE Trans. Electron.*, Vol. E91-C, No. 3, pp. 350-355, Mar. 2008.

[3] M. D. Ercegovac and T. Lang, "Digital Arithmetic," Morgan Kaufmann Publishers, 2003.

[4] A. Fujimaki, Y. Takai, and N. Yoshikawa, "High-end server based on complexity-reduced architecture for superconductor technology," *IEICE Trans. Electron.*, vol. 85, pp. 612–616, Mar. 2002.

[5] K. Fujiwara, Y. Yamashiro, N. Yoshikawa, A. Fujimaki, H. Terai and S. Yorozu, "Design and High-Speed Test of (4 x 8)-bit single-flux-quantum shift register files," Supercond. Sci. Technol., vol. 16, pp. 1456-1459, 2003.

[6] T. Nishigai, M. Ito, N. Yoshikawa, K. Obata, K. Takagai, N. Takagai, A. Fujimaki, H. Terai, S. Yorozu, "Advanced design approaches for SFQ logic circuits based on the binary decision diagram," *IEEE Trans. Appl. Supercond.*, vol. 15, pp. 380-383, June 2005.

[7] M. Tanaka, Y. Kamiya, N. Irie, A. Fujimaki, Y. Yamanashi, A. Akimoto, H. Park, N. Yoshikawa, H. Terai and S. Yorozu, "A new design approach for high-throughput arithmetic circuits for single-flux-quantum microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 17, pp. 516–519, Jun. 2007.

[8] S. Nagasawa, Y. Hashimoto, H. Numata, and S. Tahara, "A 380 ps, 9.5 mWJosephson 4-Kbit RAM operated at a high bit," *IEEE Trans. Appl. Supercond.*, vol. 5, pp. 2447–2452, Jun. 1995.

[9] S. Yorozu, Y. Kameda, H. Terai, A. Fujimaki, T. Yamada and S. Tahara, "A single flux quantum standard logic cell library," *Physica C*, vol. 378-381, pp. 1471-1474, Oct. 2002

[10] Z. J. Deng, N. Yoshikawa, S. R. Whiteley and T. Van Duzer, "Data-Driven Self-Timed RSFQ High-Speed Test System," *IEEE Trans. Appl. Supercond.*, vol. 7, pp. 3830–3833, Jun. 1997.