

A New Design Approach for High-Throughput Arithmetic Circuits for Single-Flux-Quantum Microprocessors

Masamitsu Tanaka, Yuki Yamanashi, Yoshiaki Kamiya, Aya Akimoto, Naoki Irie, Hee-Joung Park, Akira Fujimaki, Nobuyuki Yoshikawa, Hirotaka Terai, and Shinichi Yorozu

Abstract—We propose a new design approach for high-throughput arithmetic circuits based on state transitions using single-flux-quantum (SFQ) logic circuits. Microprocessors have several complex interconnects in datapath including loops of data, to which only one SFQ pulse is allowed to be confined, and the loops can spoil the high-throughput nature of SFQ circuits. In our new approach, we regard an arithmetic circuit with loops as a sequential logic circuit, and we use nondestructive readout gates (NDROs) as storage elements of the internal state. We can eliminate the loops and achieve high throughput by translating calculations into transitions of the state stored in the NDROs. We have implemented a bit-serial adder with the proposed approach, and demonstrated 36-GHz operations using the niobium 2.5-kA/cm² standard process technology.

Index Terms—Arithmetic circuits, high throughput, microprocessor, single flux quantum (SFQ) logic.

I. INTRODUCTION

IMPULSE-SHAPED voltage pulses with a width of a few picoseconds transmit binary information in the single-flux-quantum (SFQ) logic circuits [1], and permit ultrahigh speed operations over several tens of GHz. In particular, multiple SFQ pulses can be sent out serially on a single interconnect, resulting extremely high throughput performance even if the circuits have very long wirings. This feature is significantly powerful in applications with a unidirectional data flow in conjunction with the flow-clocking scheme. Several component circuits toward the network router, which is a representative unidirectional data flow application, have been already demonstrated over 40 Gbps/ch [2], [3] using the niobium 2.5-kA/cm² standard process [4].

Manuscript received August 29, 2006. This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) through ISTEK as Collaborative Research and Superconductors Network Device Project.

M. Tanaka is with Nagoya University, Nagoya, Japan, and also with the Japan Society for the Promotion of Science (e-mail: masami_t@super.nuqe.nagoya-u.ac.jp).

Y. Yamanashi, A. Akimoto, H. Park, and N. Yoshikawa are with Yokohama National University, Yokohama, Japan (e-mail: yoshi@yoshilab.dnj.ynu.ac.jp).

Y. Kamiya, N. Irie, and A. Fujimaki are with Nagoya University, Nagoya, Japan.

H. Terai is with the National Institute of Information and Communications Technology, Kobe, Japan (e-mail: terai@nict.gr.jp).

S. Yorozu is with the Superconductivity Research Laboratory, International Superconductivity Technology Center, Ibaraki, Japan (e-mail: yorozu@istec.or.jp).

Digital Object Identifier 10.1109/TASC.2007.898714

In the microprocessor application [5], [6], in contrast, highly complex interconnects are required, and loops of data usually appear in the circuitry to perform arithmetic operations called datapath. The loops make it difficult to keep throughput high, because multiple SFQ pulses are not allowed to travel in each loop simultaneously. For instance, a serial adder, the core part of the arithmetical logic unit (ALU) of the microprocessors we demonstrated [7]–[9], has a loop path to calculate the carry signal from input data and the previous carry. The loop of the carry is occasionally the critical path of serial adder.

A simple, trivial approach is to reduce the number of Josephson junctions in the loop, by optimizing geometric wiring and physical pin alignments of logic gates. In the actual design, we experienced that the delay time of the loop path of the carry was improved from 63 ps to 25 ps by reduction of junctions. However, this simple approach has an obvious drawback of limitation in the number of removable junctions. We will be obligated to decrease throughput, as the circuit becomes more complex and the number of logic gates in the loop increases. In design of deeply pipelined bit-serial microprocessors, we must control the propagation of the carry signals in order to avoid the interferences between serial data such as overflow. We do not estimate that the serial adder with such a carry controller will work over 20 GHz as long as we take the simple approach.

At the gate-level design, an advanced approach to exclude loops from SFQ arithmetic circuits was proposed, called the “push-forward” design, by A. F. Kirichenko and O. A. Mukhanov [10]. By using a specific type of interaction of SFQ pulses, the high-throughput carry-save serial adders with a remarkably small number of Josephson junctions was demonstrated.

Here we propose another new design approach to achieve high-throughput performance in SFQ arithmetic circuits with excluding loops. Although this approach requires more Josephson junctions, it is suitable for implementing easy-to-extend circuits at the cell-based design with short turnaround times, because of usage of the standard SFQ logic gates and a logic simulator in a digital domain. We present the design scheme of serial adders based on the new approach as an example, and demonstrate the extensible, high-throughput bit-serial adder.

II. ARITHMETIC CIRCUITS BASED ON STATE TRANSITIONS

In the SFQ logic, unique, excellent nondestructive readout flop-flops can be constructed. An NDRO [1] is such an SFQ

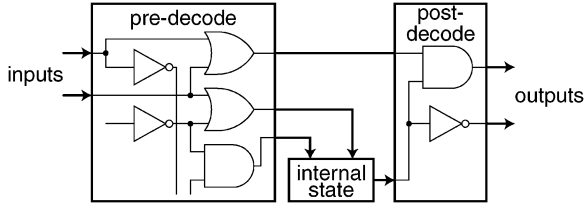


Fig. 1. Block diagram of our new design approach. The internal state is stored in SFQ nondestructive readout flip-flops.

logic gate, which can be rapidly set or reset. In our new design approach, we regard an arithmetic circuit as a sequential logic circuit, and store the internal state in NDROs. Our approach is made up of three steps illustrated in Fig. 1. At the first step, we convert the calculations into logical operations on the internal state. In other words, we decode inputs into the actions such as set, reset, or retention of the binary state. In general, the state transition also includes invert. The SFQ gate with these four functionalities is easily realized by modification of a toggle flip-flop (TFF). The second step is to update NDROs according to the results of the first step. Finally, we obtain outputs by decoding the outputs of NDROs and input signals. As a result, we can eliminate loops from the circuit. Each of steps can be processed in pipeline, namely the high-throughput nature of SFQ logic is fully utilized.

Here let us take serial adders for example of our approach. We select the carry signals as the internal state, because as mentioned in Section I, the part to calculate the carries makes a loop. For ease of explanation, we start with a bit-serial adder. Each condition of killing (k), propagating (p), and generating (g) the carry is calculated as below:

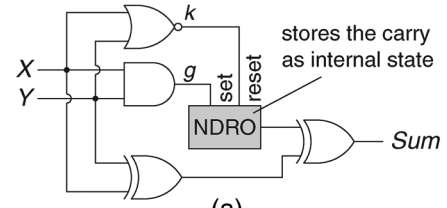
$$k = \overline{X + Y}, \quad p = X \oplus Y, \quad g = X \bullet Y \quad (1)$$

where X and Y are inputs; $+$, \oplus , and \bullet denote logical OR, XOR, and AND operations.

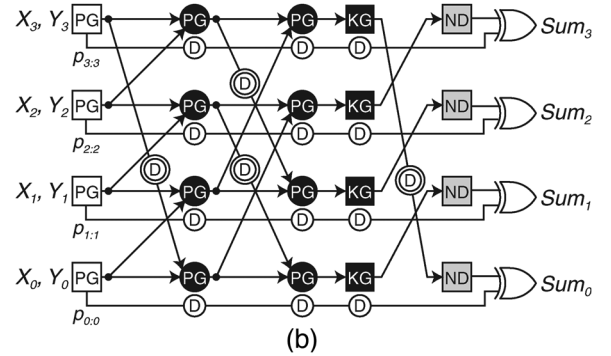
Fig. 2(a) shows the implementation of the bit-serial adder. We store the carry in an NDRO, and set or reset it in accordance with the conditions of k or g . The nondestructive readout functionality of the NDRO corresponds to the condition p . It is easily possible to control the carry externally by inserting confluence buffers before the NDRO. Although insertion of such a carry controlling circuitry makes the latency slightly larger, the throughput as high as 30 GHz or more is still expected.

This scheme is applicable to bit-slice adders, which perform addition by breaking a word into several contiguous bit groups, called slices, and by processing them serially. Because of the nature of high-speed operations of SFQ circuits, such a bit-slice computation can achieve better performance than a parallel one when data bits become much wider [11]. As well as carry-lookahead adders[12], we can compute the conditions of propagating and generating the carry signal of group of bits with numbers from i to j , denoted as $p_{i:j}$ and $g_{i:j}$ respectively, as follows:

$$p_{i:j} = p_{i:k-1} \bullet p_{k:j}, \quad g_{i:j} = g_{k:j} + g_{i:k-1} \bullet p_{k:j}. \quad (2)$$



(a)



(b)

Fig. 2. Implementations of bit-serial adder (a) and 4-bit-slice adder (b) based on state transitions. In the bit-slice adder, ND and D represent an NDRO and a D flop-flop; PG or KG surrounded with circles and squares are circuit elements to calculate conditions of p and g , or k and g , respectively.

In order to manipulate the NDRO storing the carry signal to the bit with number $j + 1$, $k_{0:j}$ and $g_{0:j}$ are required. Because p can be calculated most easily, it is better to obtain k not directly but indirectly by computation from p and g at the final stage of the first step: $k_{0:j} = \overline{p_{0:j} + g_{0:j}}$. We may calculate p and k at first. Fig. 2(b) illustrates the implementation of the bit-slice adder whose slice is four-bit wide. At the first pipeline stage, we calculate p and g of each bit by XOR and AND operations of inputs, respectively. We compute p and g using (2) in the next two stages, then calculate k and determine the state transition required for each NDRO. The final sum of each bit is obtained by XOR of the carry signal in each NDRO and the partial sum of X and Y calculated at the first stage as p . The D flip-flops (DFFs) on horizontal paths in Fig. 2(b) are introduced to make the number of pipeline stages equal.

The downward-sloping paths transmit the signals from the previous to the next slices, with delayed by a clock cycle by DFFs surrounded with double circles. It is also easily possible to break the chain of the carry signals between slices by replacing these DFFs with resettable ones.

III. DEMONSTRATION

We have designed the bit-serial adder based on state transitions shown in Fig. 2(a) using the CONNECT cell library [13]. We have not implemented the carry controlling circuitry. The target frequency is 30 GHz.

Fig. 3 is a microphotograph of the test circuit of the adder. The circuit includes shift registers and a ladder oscillator for on-chip testing [14]. The bit-serial adder itself is made up of 142 Josephson junctions on a $280 \times 320 \mu\text{m}^2$ area. The number of junctions and circuit size are almost the same as those of a conventional bit-serial cell-base-designed adder.

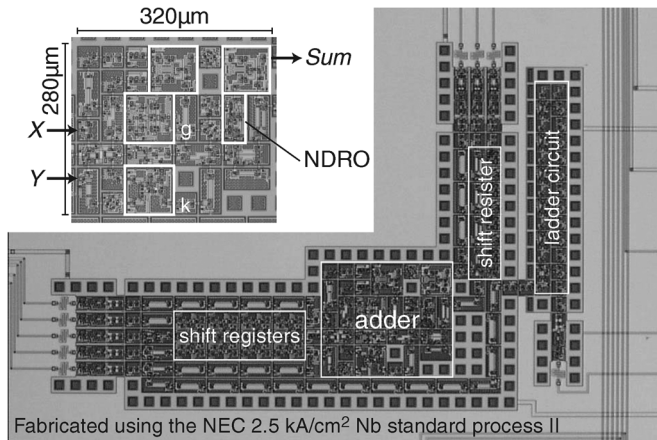


Fig. 3. Microphotograph of test circuit of the bit-serial adder. The circuit is made up of the designed adder and several components for on-chip testing. On the upper left corner, the closeup of the adder is located with the logic gates marked with frames.

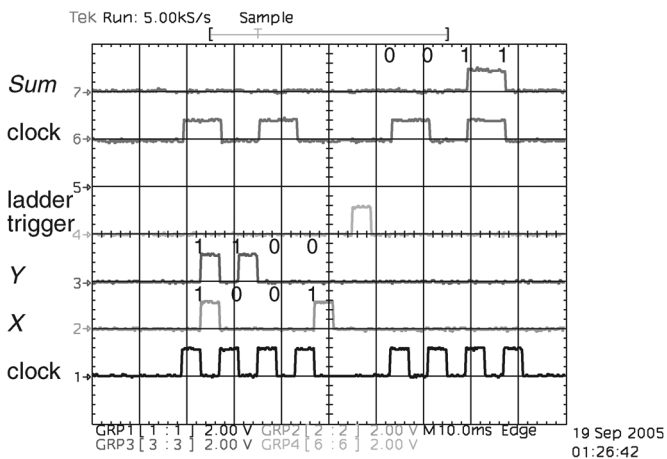


Fig. 4. A test result of the bit-serial adder. The data are 4-bit wide, and always started with the least significant bit (LSB). First, we have written data X , Y into the input shift registers, then triggered the ladder oscillator to make a high-speed calculation. At each rising edge of inputs, an SFQ pulse is generated. We have confirmed the correct operation by reading Sum from the output shift register, where transitions represent arrival of SFQ pulses.

Fig. 4 is one of test results. Here addition of two 4-bit integers has been successfully performed with high-speed clocks: $1001 + 0011 = 1100$. Note that all of the data in Fig. 4 start with the least significant bits.

The dependence of the operating range on the bias currents and clock frequencies is shown in Fig. 5. The area surrounded by two lines represents the operating range. As a result, we have found that the designed bit-serial adder could operate up to 36 GHz, as high as the maximum frequency expected for the bit-serial adder with minimized junctions in the loop by using the simple approach.

IV. CONCLUSION

We have proposed a new design approach for high-throughput SFQ arithmetic circuit. In the new approach, we regard the arithmetic circuit with loop paths as a sequential circuit. We store the internal state in the SFQ nondestructive readout flip-flops, and

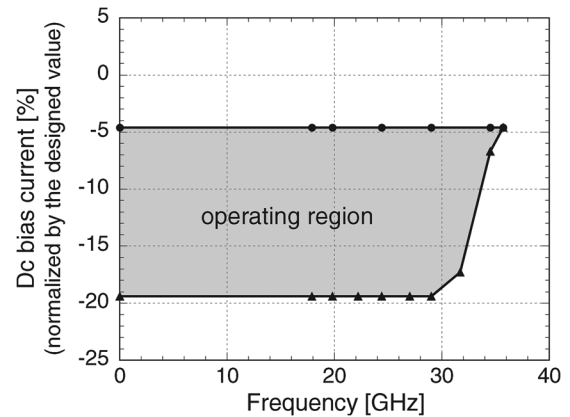


Fig. 5. The frequency dependence of the bias margin. The bias currents are normalized by the designed value.

focus on transitions of the state. By translating the calculations into state transitions, we eliminate the loops and achieve a high throughput performance.

We have described the serial adders as the applications of the new approach, and have implemented a bit-serial adder by using the niobium $2.5\text{-kA}/\text{cm}^2$ standard process. We have demonstrated its correct operations up to 36 GHz. The obtained operating frequencies of the newly designed bit-serial adder is not only as high as the maximum of those of the conventional approach, but also expected to be kept high even if we add some functional circuitries such as a controller of carry propagation.

ACKNOWLEDGMENT

The CONNECT cell library and tools are used for the demonstration. The authors would like to thank all the CONNECT members consisting of 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] Y. Kameda, S. Yorozu, Y. Hashimoto, H. Terai, A. Fujimaki, and N. Yoshikawa, "High-speed demonstration of single-flux-quantum cross-bar switch up to 50 GHz," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 1, pp. 6–19, March 2005.
- [3] T. Yamada, M. Yoshida, T. Hanai, A. Fujimaki, H. Hayakawa, Y. Kameda, S. Yorozu, H. Terai, and N. Yoshikawa, "Quantitative evaluation of the single-flux-quantum cross/bar switch," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pt. 1, pp. 237–324, June 2005.
- [4] S. Nagasawa, Y. Hashimoto, H. Numata, and S. Tahara, "A 380 ps, 9.5 mW Josephson 4-Kbit RAM operated at a high bit," *IEEE Trans. Appl. Supercond.*, vol. 5, pp. 2447–2452, Jun. 1995.
- [5] M. Dorjjevets, P. Bunyk, and D. Zinoviev, "FLUX chip: Design of a 20-GHz 16-bit ultrapipelined RSFQ processor prototype based on $1.75\text{-}\mu\text{m}$ LTS technology," *IEEE Trans. Appl. Supercond.*, vol. 11, pp. 326–332, March 2001.
- [6] A. Fujimaki, Y. Takai, and N. Yoshikawa, "High-end server based on complexity-reduced architecture for superconductor technology," *IEICE Trans. Electron.*, vol. E85-C, pp. 612–616, March 2002.
- [7] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, A. Fujimaki, H. Hayakawa, N. Yoshikawa, H. Terai, and S. Yorozu, "A single-flux-quantum logic prototype microprocessor," in *Technical Digest of IEEE International Solid-State Circuits Conference (ISSCC) 2004*, San Francisco, USA, February 2004.

- [8] M. Tanaka, T. Kondo, T. Kawamoto, Y. Kamiya, A. Fujimaki, H. Hayakawa, N. Nakajima, Y. Yamanashi, A. Akimoto, N. Yoshikawa, H. Terai, Y. Hashimoto, and S. Yorozu, "Demonstration of a single-flux-quantum microprocessor using passive transmission lines," *IEEE Trans. Appl. Supercond.*, vol. 15, no. 2, pt. 2, pp. 400–404, June 2005.
- [9] M. Tanaka, T. Kawamoto, Y. Yamanashi, Y. Kamiya, A. Akimoto, K. Fujiwara, A. Fujimaki, N. Yoshikawa, H. Terai, and S. Yorozu, "Design of a pipelined 8-bit-serial single-flux-quantum microprocessor with multiple ALUs," *Supercond. Sci. Technol.*, vol. 19, no. 5, pp. S344–S349, May 2006.
- [10] A. F. Kirichenko and O. A. Mukhanov, "Implementation of novel "push-forward" RSFQ carry-save adders," *IEEE Trans. Appl. Supercond.*, vol. 5, no. 2, pp. 3010–3013, June 1995.
- [11] H. Park, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, M. Tanaka, H. Terai, and S. Yorozu, "Design of bit-slice adders using RSFQ logic circuits," in *Appl. Supercond. Conf.*, Seattle, WA, Aug. 2006, 3EY03.
- [12] P. Bunyk and P. Litskevitch, "Case study in RSFQ design: Fast pipelined parallel adder," *IEEE Trans. Appl. Supercond.*, vol. 9, no. 2, pp. 3714–3720, June 1999.
- [13] 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, pt. 2, pp. 1471–1474, October 2002.
- [14] T. Yamada, A. Sekiya, A. Akahori, H. Akaike, A. Fujimaki, H. Hayakawa, Y. Kameda, S. Yorozu, and H. Terai, "On-chip test of the Shift register for high-end network switch based on cell-based design," *Supercond. Sci. Technol.*, vol. 14, no. 12, pp. 1071–1074, December 2001.