INVITED PAPER *Special Section on Innovative Superconducting Devices Based on New Physical Phenomena*

Towards Ultra-High-Speed Cryogenic Single-Flux-Quantum Computing

Koki ISHIDA[†a)], ***Student Member***, **Masamitsu TANAKA**[††], **Takatsugu ONO**[†††], ***and*** **Koji INOUE**[†††], ***Members***

SUMMARY CMOS microprocessors are limited in their capacity for clock speed improvement because of increasing computing power, i.e., they face a power-wall problem. Single-flux-quantum (SFQ) circuits offer a solution with their ultra-fast-speed and ultra-low-power natures. This paper introduces our contributions towards ultra-high-speed cryogenic SFQ computing. The first step is to design SFQ microprocessors. From qualitatively and quantitatively evaluating past-designed SFQ microprocessors, we have found that revisiting the architecture of SFQ microprocessors and on-chip caches is the first critical challenge. On the basis of cross-layer discussions and analysis, we came to the conclusion that a bit-parallel gate-level pipeline architecture is the best solution for SFQ designs. This paper summarizes our current research results targeting SFQ microprocessors and on-chip cache architectures.

key words: *single flux quantum (SFQ), cryogenic computing, microprocessor, cache memory, Josephson junction, low-power, high-performance, energy-efficient*

1. Introduction

Moore's Law, the observation in which the number of transistors in a chip doubles every 18 months, has so far been contributed to the evolution of computer system architectures, e.g., introducing manycore accelerators, using large on-chip caches, and increasing DRAM (or main memory) capacity. The growth of such hardware implementation increases the optimization opportunities available to software developers. Unfortunately, further transistor shrinking cannot be expected anymore, i.e., the end of Moore's Law will come. While device and manufacturing technologies have continued progressing, some researchers predict that transistors shrinkage may stop at around 2025 to 2030 because of physical or economic reasons. To tackle with such a critical problem, it is necessary to exploit emerging devices that have significant potential for performance and energy efficiency rather than relying upon conventional CMOS devices.

Cryogenic computing is a promising approach for achieving sustainable improvement in the post-Moore's era. Real implementations are available in the market as quantum computers [1], [2]. Although they can effectively be applied to specific purposes such as quantum annealing, there is a large gap regarding functionality between classical digital computing and the application-specific quantum acceleration. Bridging the gap is a critical issue for making cryogenic computing applicable to a wide range of emerging applications. Superconductor Single-Flux-Quantum (SFQ) logic [3] is a promising, practical VLSI technology for achieving the general purpose cryogenic computing, and some researchers have so far greatly been contributed to developing SFQ devices and logic design technologies; their physical designs and the successful operations of SFQ microprocessors have been demonstrated [4]–[7].

Although the SFQ microprocessors operate with outstanding clock frequency, e.g., several dozen GHz or even more than 100 GHz, unfortunately, their effective performance regarding "*program execution time*" is comparable or worse than that of state-of-the-art CMOS-based microprocessors. The fundamental problem existing at behind the SFQ microprocessors is the lack of optimization from the viewpoint of microarchitecture, i.e., the structure of the SFQ microprocessors does not fully exploit the potential of SFQ logic. To solve this issue, we have revisited the SFQ microprocessor/memory architectures, and this paper presents our previous research results that stand on a device/circuit/architecture level co-design [8]–[11]. This kind of cross-layer optimization is the key to realizing post-Moore's computing with emerging devices.

This paper is organized as follows. The device features and challenges of SFQ circuits are described in detail in Sect. 2. Section 3 presents our proposed architecture for SFQ microprocessors, and we evaluate it in Sect. 3.2. We also show SFQ cache architecture in Sect. 4, and the estimation results of the cache are shown in Sect. 4.3. Finally we conclude in Sect. 5.

2. Background

2.1 SFQ Circuits

Information processing in SFQ circuits is performed with magnetic flux-quanta stored in superconducting rings containing Josephson junctions (JJs). The presence (or absence) of a single flux quantum represents a logical '1' (or '0'), and a JJ acts as a switching device like a transistor, where an impulse-shaped voltage pulse, called an SFQ pulse, is gen-

Manuscript received October 1, 2017.
Manuscript revised January 5, 2018.

[†]The author is with Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan.
[††]The author is with Department of Electronics, Nagoya University, Nagoya-shi, 464–8603 Japan.
[†††]The authors are with Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819–0395 Japan.
a) E-mail: koki.ishida@cpc.ait.kyushu-u.ac.jp

DOI: 10.1587/transele.E101.C.359

erated only when an SFQ travels across a JJ.

In RSFQ logic [3], a timing reference signal, traditionally it has been called a "clock signal", is used to drive SFQ logic gates. In this paper, we refer to this signal as the **Gate Driving Clock (GDC)**. If an SFQ pulse is fed to an SFQ logic gate within a GDC cycle, the gate recognizes the input as logic information '1'; otherwise the input value is treated as '0'. On the other hand, the **System Driving Clock (SDC)** is used as a reference signal to drive a microprocessor's pipeline stages (or flip-flops), which are the same as the clock signals used in CMOS designs.

The maximum frequency of GDC depends on the *HoldTime*, *SetupTime*, and some delays of SFQ gates. *HoldTime* is a duration from an arrival time of a GDC pulse to that of an input data pulse. *SetupTime* is a duration from an arrival time of an input data pulse to that of the next GDC pulse. The GDC cycle time (GDC_{CT}) is given by Eq. (1),

$$GDC_{CT} = \max(HoldTime_{ng}, GateDelay_{pg} + WD) + SetupTime_{ng} + M \quad (1)$$

where, $HoldTime_{ng}$ and $SetupTime_{ng}$ are *HoldTime* and *SetupTime* of the next gate, respectively. $GateDelay_{pg}$ is the gate delay of the previous gate, whose output is connected to the next gate, M is the margin for fabrication variation, and WD is the delay caused by wire signal propagation from the previous gate to the next gate. If SFQ circuits employ concurrent-flow clocking and do not have feedback loops, delay elements called *skew* can be inserted into the GDC lines to keep their frequency at the maximum, i.e., $GateDelay_{pg}$ and WD are hidden by skew and the maximum GDC frequency depends on only *HoldTime* and *SetupTime* as shown in Eq. (2).

$$GDC_{CT} = (HoldTime_{ng} + M) + (SetupTime_{ng} + M) \quad (2)$$

2.2 State-of-the-Art SFQ Microprocessors

Some researchers have so far been designed SFQ microprocessors and demonstrated their ultra-high-speed, low-power operations [4]–[6], [12]. Also, by virtue of recent developments in fabrication technology with a multi-layered structure [13], [14] and some energy-efficient circuit technology [15]–[19], several research projects on SFQ microprocessors have been undertaken [20], [21], and an 8-bit microprocessor composed of more than 10,000 Josephson junctions successfully operates at 50 GHz [7]. In addition, inventions of several novel superconductor devices based on new physical phenomena have made the SFQ logic technology much more attractive, e.g., superconductor junctions with one or more ferromagnet layers show different electrical characteristics depending on the magnetic states, by which we can obtain high-density, low-power, cryogenic memory compatible with SFQ circuits [22], [23]. A thermally-assisted nano-structured device called a nanocryotron (nTron) showed the capability of voltage output in the sub-volt range [24], by which we can build very large-scale cryogenic memory by hybridizing Josephson and CMOS integrated circuits [25].

2.3 Challenges

Since we can expect ultra-high-speed, low-power cryogenic computing by exploiting SFQ microprocessors, the following fundamental issues remain.

1. Microprocessor core: SFQ microprocessors that have so far been demonstrated [4]–[7] follow the complexity-reduced (CORE) architecture [12]. Although the SFQ microprocessors such as CORE 1β successfully operates at a 25-GHz GDC frequency [4], the microprocessor's speed as a whole (SDC frequency) is 1.5 GHz, which does not produce outstanding performance improvement over current CMOS microprocessors. This is because the microarchitecture applied to the SFQ processors does not effectively exploit the characteristics of SFQ devices, i.e., bit-serial processing, and the pipeline structure of the prototyped CMOS microprocessors.
2. On-chip memory: The memory subsystem has an important role in computer systems, and it is well known that it has a strong impact on computer system performance. So, making a good balance between the microprocessor cores and memory subsystem is an important design optimization. A previous design that implemented an SFQ L1 cache has been prototyped in order to realize a high-speed memory [6]. The cache uses an SFQ shift register and operates in the bit-serial fashion, as does the microprocessor core, to reduce hardware complexity. However, such bit-by-bit fine-grained operations make the cache access time much longer, resulting in poor microprocessor performance. In addition, the large scale of the selector logic used to pick up referenced data strictly limits the scalability of the cache capacity.

So far, few studies have focused on architectures for SFQ microprocessors and memories. As explained in Sect. 2.2, several technical advancements indicate that the research on SFQ microprocessor architecture and memory hierarchy play a more important role in the development of SFQ-based full-scale computing. From the next section, we introduce our research attempts to answer the above mentioned two challenges.

3. Revisiting SFQ Microprocessor Architecture

In this section, we introduce a new SFQ microprocessor architecture that exploits the potential of SFQ devices. The details of this section have been discussed in [8].

3.1 Architectural Design Space Exploration

We have exploblack the architectural design space of SFQ

microprocessors in our previous work [8], [10]. By developing a performance model that reflects the impact of pipeline depth of a microprocessor's organization, we have analyzed the performance of bit-serial, bit-slice, and bit-parallel SFQ microprocessors with appropriate pipeline depth. As a result, we have reached the following conclusions.

- **Bit-parallel processing**: Unlike bit-serial or bit-slice operations that were applied to the previous SFQ microprocessor designs, the bit-parallel processing handles the microprocessor's word size at the same time (in parallel). There are at least three advantages: 1) no feedback loops appear within a word level in bit-parallel circuits and this feature makes it possible to maintain the maximum clock frequency as explained in Sect. 2.1, 2) the latency of word-size operation can be blackuced by exploiting bit-level parallelism, 3) the logic design becomes more simple by removing control circuits for bit-serial or bit-slice repeated operations.
- **Gate-level deep pipelining**: Gate-level pipelining is the most fine-grained pipeline structure, in which one pipeline stage consists of only one logic gate. In general, this extremely deep pipeline structure cannot be applied in CMOS designs because the area overhead of pipeline registers becomes greater. However, this disadvantage does not appear in SFQ designs because of its device features. SFQ logic gates inherently have a kind of latch function, so that pipeline registers are not needed[†]. Moreover, GDC can be used as SDC in the gate-level pipeline structure, blackucing the complexity of system-clock designs significantly.

On the basis of the above-mentioned architecture, we have decided to choose the gate-level bit-parallel datapath architecture for SFQ microprocessors to achieve 100-GHz clock (SDC) frequency operations. This structure cannot be applied to traditional CMOS based microprocessors because of the power-wall problem. In CMOS designs, power consumption increases with clock frequency, and 100-GHz operation is impractical because of heat problems. On the contrary, the dynamic power needed for an SFQ logic gate which uses more energy-efficient technology, such as ERSFQ, is about 1/10,000 of that needed for a CMOS logic gate [26], e.g., 0.01 μW even at 100 GHz operation. Therefore, such a fine-grained pipeline structure is suitable for SFQ microprocessors. However, in such deep pipeline structure, pipeline stalls caused by for instance by data dependency, branch pblackiction misses, cache misses, etc., significantly degrade the system performance. To avoid this issue, we have decided to apply the following architectural strategy.

- **Fine-grained multithreading**: In order to realize ultra-high-performance SFQ microprocessors by using gate-level pipelining, most pipeline stalls must be concealed. Current CMOS microprocessors adopt out-of-order execution to conceal pipeline stalls. However, in SFQ pulse logic, the timing adjustment of pulses is too critical to implement such complex circuits. Therefore, we use fine-grained multithreading that prepares as many threads as the number of pipeline stages[††] and switches the thread to be executed every clock cycles. The fine-grained multithreading can conceal all pipeline stalls that are caused by data hazard while keeping the hardware simple.

The register file in our fine-grained multithreaded SFQ microprocessor has a special function because the microprocessor requires as many register sets as the number of threads to maintain the architectural states of all threads. In addition, the output has to be switched every clock cycles for associated threads. On the other hand, shift register is the best choice for implementing a memory unit that can follow the ultra high-speed of SFQ circuits. So, we have decided to implement the register file by a circular buffer implemented by SFQ shift registers.

3.2 Evaluation of Bit-Parallel Microprocessor

3.2.1 Logic Design

The purpose of the logic design is to evaluate the effectiveness of the SFQ microprocessor on the basis of architectural design policies by estimating performance, area, and power consumption. In particular, in gate-level pipelining, the number of pipeline stages is determined by the number of logic gates included in the critical data path. The length of other data paths must be equalized to the critical data path by inserting latches. The number of the latches has a critical impact on circuit scale and power consumption of the SFQ microprocessor. So, it is necessary to know the number of pipeline stages to analyze this overhead. Therefore, we have designed a gate-level pipelined bit-parallel SFQ microprocessor by using Verilog HDL with gate-level entries (not RTL) and NC-Verilog is used for simulation.

The word size of the designed SFQ microprocessor is 8 bits, and 32-bit basic MIPS instructions, arithmetic operations (*add*, *addi*, *sub*), unconditional and conditional branches (*beq*, *bne*, *j*, *jr*, *jal*), and data transfer (*load*, *store*), are supported.

Figure 1 shows the block diagram of the designed 8-bit SFQ microprocessor [8]. We have extracted the pipeline depth requiblack to implement our gate-level pipeline microprocessor, and have found that 52 pipeline stages are needed. It is determined by the number of logic gates included in the critical data path. On the basis of the 8-bit SFQ microprocessor, we have estimated that a 64-bit SFQ microprocessor consists of 58 pipeline stages. We assume that the scale of microprocessor linearly increases with its bit width. We have confirmed the correct operations of all instructions

[†]Some DFFs are needed in order to align the stage of gates, and the scale of DFFs is the overhead of the pipeline structure.

[††]In fact, $\lceil skew_{all}/GDC_{CT} \rceil$ more threads are needed, where $skew_{all}$ is total skew of the critical path, in order to fill the pipeline with instructions of different threads.

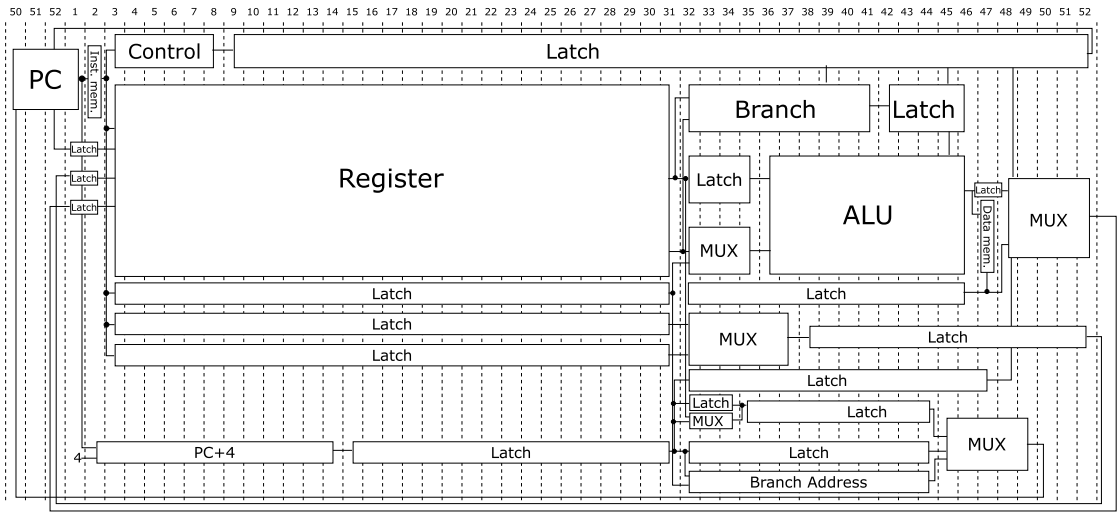


Fig. 1 Block diagram of the 8-bit SFQ microprocessor.

except for *store*. This is because we do not implement a way for initializing pipeline registers, and that let the undefined values propagate inside of the SFQ microprocessor's data path. However, the total impact on performance, power consumption, and area overhead are negligible.

3.2.2 Performance

In this section, we assume that each memory access can be completed in one clock (SDC) cycle in order to evaluate the potential of the designed SFQ microprocessor. The system-level clock cycle time (*CCT*) corresponds to GDC_{CT} in Sect. 2 because our proposed architecture supports gate-level pipelining. Furthermore, because we employ concurrent-flow clocking, we also assume that a gate delay and a wire delay can be hidden by skew. Therefore, *CCT* is given by Eq. (2) in Sect. 2. Each SFQ logic gate has its *HoldTime* and *SetupTime* and we use the longest *CCT*, which is 4.5 ps on the SFQ *XOR* gate for calculating clock frequency. These parameters are based on a 0.3μm Nb process, and we calculate them from the parameters of a 1.0μm Nb process [27] by applying the scaling rule for JJs[†]. This scaling can be applied up to 0.3μm because the SFQ pulses cannot be narrower less than that size [28]. On the basis of the parameters, we estimate that the clock frequency of the SFQ microprocessor can be 222 GHz. In this evaluation, we estimate M of Eq. (2) from past design results [4]–[6], [12]. However, because the scale of our SFQ microprocessor is quite larger than that of past design, it seems that more large margin is needed. Although there is the report about the impact of increasing jitter for *CCT* is not large [29], the detailed study on the basis of evaluating designed element circuits and prototype of the SFQ microprocessor are essential. In addition, developing scalable clocking techniques that are suitable for large microprocessors are future work.

[†]If we scale the JJs to $1/\alpha$, then the switching and signal propagation delays also become to be $1/\alpha$

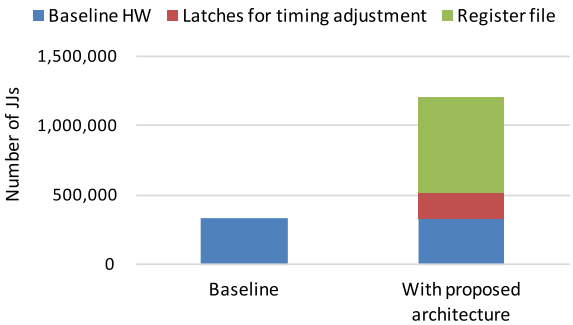


Fig. 2 Area breakdown of the 64-bit SFQ microprocessor [8].

3.2.3 Area

We estimate the area of the SFQ microprocessor on the basis of the number of JJs because we have not completed the layout design yet. The evaluation results show that the part of the estimated 64-bit microprocessor, i.e., without the wire and skew circuits, consists of about 1.2 M JJs and the whole microprocessor consists of about 3.5 M JJs[††]. Figure 2 shows an area breakdown of the 64-bit microprocessor [8]. Baseline hardware (HW) means an SFQ microprocessor without inserting latches for timing adjustment and a large number of register sets for fine-grained multithreading. The results show that the area of the proposed SFQ microprocessor is 3.58 times larger than the baseline HW. Moreover, the register file occupies 57.46% of the area because the fine-grained multithreading approach requires a lot of register sets as mentioned in Sect. 3.

3.2.4 Power Consumption

Power consumption of the SFQ microprocessor is given by Eq. (3),

[††]The number of JJs of wire and skew are estimated from past designs [30]

$$P = (\alpha \Phi_0 I_c f + V I_{bias}) \times N_{JJ}, \quad (3)$$

where α is switching probability, Φ_0 is magnetic flux of single quantum, I_c is the critical current of a JJ, f is clock frequency, V is voltage of bias current, I_{bias} is bias current, and N_{JJ} is the number of JJs in a whole microprocessor. In the worst case, i.e., $\alpha = 1$, the power consumption of the SFQ microprocessor is 1.46 W.

3.2.5 Comparison with CMOS Microprocessor Models

We compare the SFQ microprocessor and CMOS microprocessor models in terms of performance and power consumption to evaluate the effectiveness of the SFQ microprocessor. Because of many implementation differences, it is difficult to fairly compare the SFQ and CMOS microprocessors. So, this preliminary evaluation is a reference-level result.

For CMOS processor models, we assume two configurations: high-performance operation with high supply voltage (CMOS-HP) and low energy operation with low supply voltage (CMOS-LE). CMOS-HP operates at 5 GHz with 11 W, and CMOS-LE operates at 3.2 GHz with 3 W. Both of them can execute two instructions per clock cycle. We referblack to the parameters of the clock frequency, power consumption, and the number of pipeline stages from Cell Broadband Engine Synergistic Processor Element (90 nm) [31] for the CMOS microprocessor models. After these microprocessors, CMOS microprocessors tend to improve multi-thread performance by increasing the number of processing cores and decreasing each clock frequency to blackuce power consumption. In this evaluation, we use Cell processor's parameters because it is a represent microprocessor following in-order execution and a deep pipeline structure (26 stages) to achieve high clock frequency. It is essential to compare to state-of-the-art CMOS microprocessors in order to show the effectiveness of our proposal, and it is future work.

We evaluate performance with **Billion Instructions Per Second** (**BIPS**) to clear the effect of pipeline stalls. BIPS is calculated by the product of $1/CCT$ (i.e., clock frequency) and **Instructions Per Clock cycle** (**IPC**) and is given by Eq. (4),

$$BIPS = IPC/CCT \quad (4)$$

IPC is given by Eq. (5),

$$IPC = x/(1 + \omega \times (p - 1)), \quad (5)$$

where x is the number of executable instructions simultaneously, ω is the ratio of pipeline stalls, and p is the number of pipeline stages. In specifically, ω is the average rate of the number of pipeline stalls to pipeline stages, e.g., $\omega = 1$ means that $(p - 1)$ pipeline stages stall (only 1 instruction is in-flight). In the CMOS models, we assume ideal conditions in which $BIPS = 2 \times 1/CCT$ ($\omega = 0$). The comparison results are indicated in Fig. 3. The y-axis shows the performance ratio of the SFQ microprocessor to the CMOS models and the x-axis shows ω. Figure 3 shows that the performance ratios of our SFQ microprocessor to CMOS-HP and CMOS-LE is 22 times and 35 times higher, respectively [8]. However, if $\omega = 10\%$, the performance ratios of our SFQ microprocessor to CMOS-HP and CMOS-LE go down to 3 times and 5 times, respectively. Therefore, pipeline stalls should be hidden to maintain high performance.

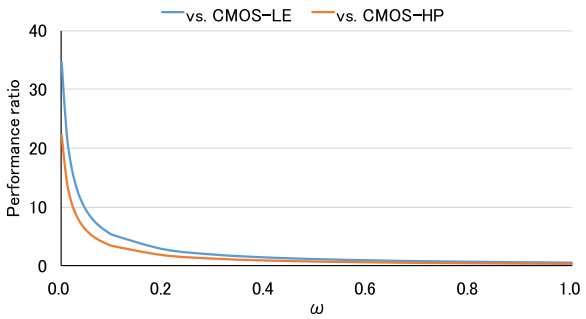


Fig. 3 BIPS performance comparison between SFQ and CMOS microprocessors [8].

The power consumption of CMOS-HP and CMOS-LE while running single-precision intensive applications is 11 W and 3 W, respectively. The power consumption of the SFQ microprocessor system including the power consumption of cryocooler, which is a specific cooling system for SFQ circuits, is given by Eq. (6).

$$P_{total} = P_{CPU} + P_{cooler} \quad (6)$$

We assume that the power consumption of cryocooler is 1,000 times higher than that of the microprocessor [32]. Because our SFQ microprocessor does not have a floating-point unit (or FPU), we estimate P_{CPU} on the basis of Eq. (7),

$$P_{CPU} = (P_{IntCPU} + P_{FPU} + P_{FPUoh}) \times \gamma, \quad (7)$$

where P_{IntCPU} is the power consumption of our SFQ microprocessor, P_{FPU} is the power consumption of 32-bit FPU, P_{FPUoh} is the overhead of an increase of pipeline stages when the FPU is included, and γ is the power consumption blackuction rate. In specifically, γ is the ratio of the power consumption of a certain energy-efficient SFQ logic gate to that of an RSFQ logic gate. We use $\gamma = 1/100$ on the assumption that our SFQ microprocessor employs ERSFQ [26], [33]. We assume that P_{FPU} linearly increases with its bit width and we calculate that the power consumption of the 32-bit FPU is 32 times greater than that of the bit-serial FPU [34]. The power consumption of the bit-serial FPU is 0.51 mW, and P_{FPU} is estimated to be 16.32 mW. P_{FPUoh} is given by Eq. (8),

$$P_{FPUoh} = (N_{DFF-GA} + N_{DFF-MT}) \times P_{1DFF}, \quad (8)$$

where N_{DFF-GA} is the number of additional DFFs to align the stage of gates when the FPU is included, and N_{DFF-MT} is the number of DFFs requiblack for additional register sets which store architectural states of added threads. N_{DFF-GA} and N_{DFF-MT} are given by Eqs. (9), and (10), respectively.

$$N_{DFF-GA} = BW_{total}$$

$$\times \max(0, (N_{Gate-FPU} - N_{Gate-ALU})) \quad (9)$$

$$N_{DFF-MT} = BW_{word} \times N_{reg} \times \max(0, (N_{Gate-FPU} - N_{Gate-ALU})) \quad (10)$$

Here, BW_{total} is the total bit width of the SFQ microprocessor's data path which need to be aligned the stage of gates when the FPU is included, $N_{Gate-FPU}$ and $N_{Gate-ALU}$ are the number of logic gates on the critical data path of each functional unit, respectively, BW_{word} is the bit width of a word data, and N_{reg} is the number of registers per a register set. Specifically, we assume that the FPU is placed in parallel with the ALU of our SFQ microprocessor, and we calculate BW_{total} by the sum of bit width of the multiplexer's input which selects write-back data, and the data paths which feedback from the last stage of the microprocessor to the top except for write-back data path. As a result, P_{FPUoh} is 0.31 W, and the power consumption of the SFQ microprocessor with the 32-bit FPU (i.e., $(P_{IntCPU} + P_{FPU} + P_{FPUoh})$) is estimated to 1.77 W. Therefore, P_{CPU} is 0.0177 W and the power consumption of the SFQ microprocessor system, P_{total} is estimated to be 17.7 W. The power consumption of the microprocessor without cryocooler is 1/620 and 1/170 of that of CMOS-HP and CMOS-LE, respectively. When the impact of cryocooler is taken into account, the power consumption is about 1.6 and 5.9 times larger than that of CMOS-HP and CMOS-LE, respectively. Although the SFQ microprocessor consumes more power than CMOS models, clock frequency per power of the SFQ microprocessor is 27.6 and 11.8 times faster than that of CMOS-HP and CMOS-LE, respectively. Thus, the results indicate that an SFQ microprocessor that uses our proposed architecture has the potential to outperform CMOS models with the balance between speed and power.

According to these evaluations, our SFQ microprocessor has a potential to achieve higher performance than CMOS microprocessors which follows similar architecture. Therefore, SFQ microprocessors seem to be suitable for coprocessors or accelerators which follow simple structure to achieve high clock frequency rather than a general purpose processor which needs more complex logic. To prove the correctness of our architectural direction, we have designed a bit-parallel gate-level pipelined ALU targeting 50-GHz operation. Although its word size is 8 bits, and it only supports simple functions, we have successfully demonstrated its 56-GHz operation at 1.6 mW. Now we are going to design the proposed bit-parallel gate-level pipelined multithreaded SFQ microprocessor to confirm its correct operation.

4. Revisiting SFQ Cache Memory Architecture

In this section, we introduce a new SFQ on-chip cache memory architecture that considers the characteristics of SFQ devices. The details of this section have been discussed in [9].

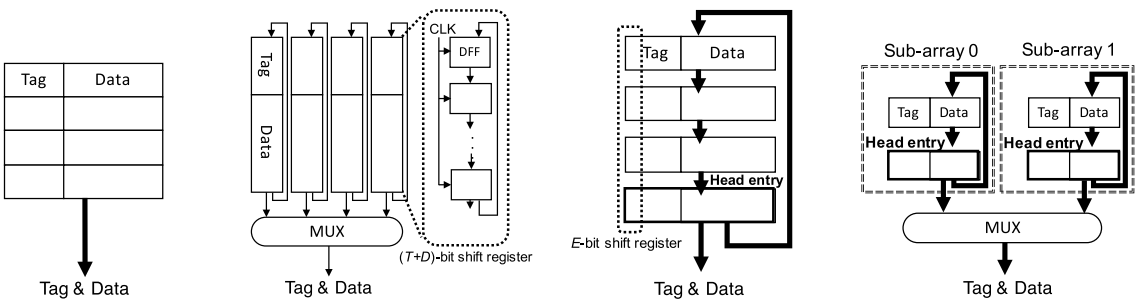
4.1 Bit-Parallel SFQ Cache Architecture

As explained in Sect. 2.3, the traditional SFQ cache operates in a bit-serial fashion to blackuce hardware complexity. As well as the microprocessor core discussed in Sect. 3, the cache architecture must also be revisited. In this paper, we assume to implement a direct-mapped organization.

Figure 4 shows the structure of a RAM-based traditional cache, a bit-serial SFQ cache, and our bit-parallel SFQ caches. In CMOS design, it is common to use the RAM-based cache. Although, in SFQ design, there are some RAM-based memory technologies such as Vortex Transition Cell (or VTC) [35], they are not fast enough to provide data to microprocessors which operate over 100 GHz. Therefore, shift-register-based memory technology is employed to achieve more fast access speed. Table 1 summarizes the access latency (in clock cycles) and the number of inputs of a multiplexer used for selecting the referenced the pairs of tag and data, where T and D describe the size of the tag and data in bits, E is the number of cache entries implemented by shift register, and S is the number of sub-arrays. Although there is no RAM-based cache which can operate over 100 GHz, i.e., one clock cycle time is ten ps, we

Table 1 Access latency and number of multiplexer's inputs of each cache.

Type of cache	RAM-based	Bit-serial	Bit-parallel	
			Unified	Sub-arrayed
Access latency [cc]	1	$T + D$	$1 \sim E$	$1 \sim E/S$
Number of multiplexer's inputs	N/A	E	N/A	S



(a) RAM-based cache (b) Bit-serial cache (c) Bit-parallel cache (d) Sub-arrayed cache

Fig. 4 Architecture models of each cache.

use the concept of RAM-based memory in order to explain our proposed cache architecture, and show its parameters as ideal cache structure.

As depicted in Fig. 4 (a), the RAM-based cache can complete its access in one clock cycle and no multiplexer is requiblack to select data. On the other hand, in the bit-serial SFQ cache as shown in Fig. 4 (b), it takes $T + D$ cycles to access the accessed data because all bits in each pair of tag and data are stoblack in a $(T + D)$-bit shift register. As a result, in the bit-serial cache, access latency is $T + D$ times longer. The multiplexer is used to choose a pair of tag and data pointed by the reference address, so this structure strictly limits the scalability in terms of the cache capacity. Since the bit-serial cache requires E of $(T + D)$-bit shift registers, we need to increase the parameter E for implementing a larger cache capacity, resulting in an unacceptable linear increase in the number of input ports on the multiplexer.

To solve the problem on the bit-serial structure, a naive bit-parallel SFQ cache memory architecture (the bit-parallel cache) can be consideblack as shown in Fig. 4 (c). In this cache, all bits with the same position in each pair of tag and data are stoblack in an E-bit shift register, so that $(T + D)$ of E-bit shift registers are requiblack. Since D is generally much larger than T and is independent of the number of cache entries, the cache can maintain the scalability. Instead of the multiplexer, the cache needs a control logic, called *shift controller*, to decide the number of shifts for accessing the referenced tag-data pair. Here, we call the cache entry connected to the output port of the memory array *head entry*. The output of the head entry is fed back to the input of the associated shift registers, so that it works as a circular buffer. The shift controller calculates distance *DIST*, which is defined as the number of cache entries existing between the head and the referenced entries, and attempts to repeat shift operations *DIST* times. As a result, the referenced tag-data pair moves to the head entry, so that we can read its contents. To implement this function, the shift controller needs to hold the index address information of the previous access, i.e., the index address associated with the tag-data pair existing in the head entry. Access latency of the bit-parallel cache varies from 1 to E.

Another alternative that can alleviate the worst case long access latency of the bit-parallel SFQ cache is to support a sub-arrayed structure as shown in Fig. 4 (d). The worst access latency of the sub-arrayed cache becomes shorter because the shift register array, e.g., one of the four-entry arrays in (c), is partitioned into several sub-arrays, e.g., two of the two-entry sub-arrays in (d). For example, if a cache consists of S sub-arrays, the worst access latency becomes $1/S$. However, the cache needs a multiplexer to select the referenced tag-data pair from the S of sub-array outputs. Since there is a trade-off between the multiplexer and sub-array access latency, deciding an appropriate value of S is a critical issue.

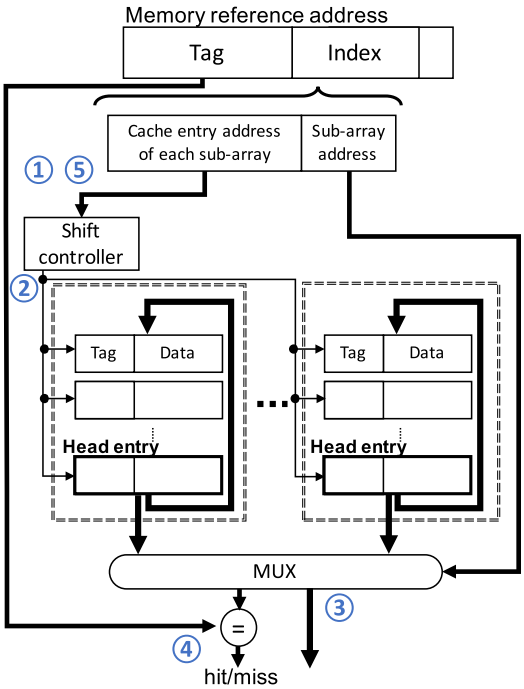


Fig. 5 High level structure of the sub-arrayed cache.

4.2 Operation Example

Figure 5 illustrates a high level structure of the bit-parallel sub-arrayed cache. Some parts surrounded by the broken line correspond to sub-arrays. The cache is a direct-mapped organization, and the high-order bits of the index correspond to the cache entry address in each sub-array, while the low-order bits are used to select an associated output (a tag-data pair) from sub-arrays. On a read, the cache works as follows.

1. *DIST* is calculated by obtaining the difference between the index in the referenced address and that for the head entry stoblack in the shift controller.
2. The shift controller generates as many pulses as *DIST* which are provided to all sub-arrays. An associated tag-data pair in each sub-array is shifted to the head entry.
3. The referenced tag-data pair is selected on the basis of the reference address by the multiplexer.
4. The cache returns a hit or miss signal on the basis of the result of tag comparison and a valid bit. On a hit, the read data is transferblack to the microprocessor.
5. The current index address of the head entry is stoblack in the shift controller to accept a next access.

4.3 Evaluation of the Bit-Parallel Cache

4.3.1 Methodology

In this section, we evaluate the access latency, power con-

sumption, and area of our proposed sub-arrayed cache architecture by modeling. Because the proposed caches are mainly constructed of three components, the shift controller, the shift register arrays, and the selector, the cache access latency T can be estimated by Eq. (11),

$$T = T_{sc}+T_{sra} + T_{sel}, \tag{11}$$

where T_{sc} is the time for calculating the number of shifts requiblack to access the referenced cache entry (the latency of the shift controller), T_{sra} is the time for shift operations (the latency of the shift register arrays), and T_{sel} is the time for selecting the referenced data (the latecy of the selector). T_{sc}, T_{sra}, and T_{sel} are given by Eqs. (12), (13), and (14), respectively.

$$T_{sc} = GDC_{CTsc} \times N_{Gate-sc} \tag{12}$$
$$T_{sra} = GDC_{CTsra} \times N_{Gate-sra} \tag{13}$$
$$T_{sel} = GDC_{CTsel} \times N_{Gate-sel} \tag{14}$$

Here, GDC_{CTsc}, GDC_{CTsc}, GDC_{CTsel} are the cycle time of gate driving clock applied to the shift controller, the shift register arrays, and the selector, respectively. They can be calculated on the basis of Eq. (1), which includes a wire delay and a gate delay[†], and $N_{Gate-sc}$, $N_{Gate-sra}$, and $N_{Gate-sel}$ are the number of logic gates on the critical data path of each functional unit, respectively. In this evaluation, we assume that each functional unit can operate at its own GDC, and the data transfer between each unit can be guaranteed by a timing adjustment. $N_{Gate-sc}$, $N_{Gate-sra}$, and N_{sel} are given by Eqs. (15), (16), and (17), respectively.

$$N_{Gate-sc} = 3 + 2 \times \lceil \log_2(B_{index} - \log_2 N_{sub}) \rceil \tag{15}$$
$$N_{Gate-sra} = 2^{B_{index}} - \log_2 N_{sub} - 1 \tag{16}$$
$$N_{Gate-sel} = 4 + \log_2 N_{sub} + \lceil \log_2 B_{tag} \rceil \tag{17}$$

Here, B_{index} is the bit length of the index, B_{tag} is the bit length of the tag, and N_{sub} is the number of sub-arrays. We evaluate the area and power consumption on the basis of the number of JJs. The number of JJs including wiring cost is estimated from past design results [30] because we have not done the layout design, and we assume that WD in Eq. (1) is 1.8 ps. Power consumption is given by Eq. (3) in Sect. 3.2.

In this evaluation, we regard an SFQ shift register memory presented in [36] as a traditional bit-serial cache explained in Fig. 4 (b) because of the lack of comparable design information for SFQ bit-serial caches, and we use it as a baseline. Since the baseline does not include the tag coincident comparison logic and the cache controller, the evaluation is pessimistic for our bit-parallel cache. The parameters of the evaluation target are as follows. The word size is 64 bits, the data size (or cache block size) is one word, the cache capacity is 2 KB, and the address length is 32 bits in which the high-order 21 bits are used as the tag and the low-order 8 bits (without 3 bits for selecting byte data) are for the index.

[†]These delays are not hidden by skew because the cache has feedback loops in shift register arrays.

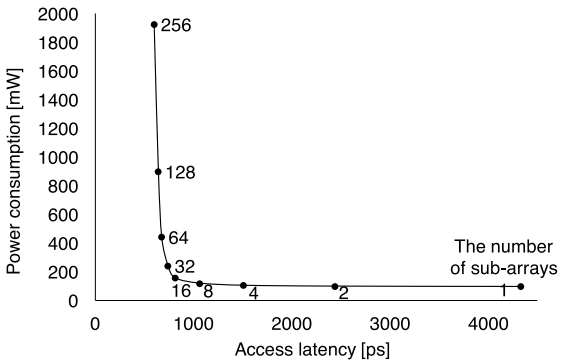


Fig. 6 Relationship between access latency and power consumption by varying number of sub-arrays [9].

Table 2 Breakdown of estimated latency, area, and power consumption of each functional unit.

Functional unit	Latency [ps]	#JJs	Power consumption [mW]
Shift controller	171.5	3,751	1.438
Shift register arrays	107.1	240,270	93.35
Selector	458.0	380,385	146.1

We use the latency values of 24.5 ps, 15.3 ps, and 22.9 ps, each of which is the longest GDC cycle time in each functional unit by assuming a 1.0 μ Nb process [27], as GDC_{CTsc}, GDC_{CTsra}, and GDC_{CTsel}, respectively. The dynamic power is estimated by assuming the worst case, i.e., the switching activity α is 1.0, whereas the static power is obtained by multiplying the average leakage power of a JJ and the total number of JJs requiblack to implement the cache. On the basis of the design report [36], we assume that the area of one JJ is $8.69 \times 10^{-4} mm^2$, and the access latency and area of the baseline are 1,333 ps and $265.9 mm^2$, respectively.

4.3.2 Results

Before discussing the estimated results, we have to decide an appropriate number of sub-arrays. Figure 6 shows the relationship between access latency and power consumption by varying the number of sub-arrays. If the number of sub-arrays increases, the access latency becomes shorter, while the power consumption becomes larger. This is because the scale of the multiplexer to pick up data from each sub-array's output becomes huge. Therefore, we have decided to focus on a 32 sub-arrayed cache which is well-balanced regarding performance and power consumption.

The estimated results for the access latency, area, and power consumption of the bit-parallel 32 sub-arrayed SFQ cache are shown in Table 2. The estimated access latency of the 32 sub-arrayed cache is 736.6 ps and is 1.8 times faster than the baseline. This advantage mainly comes from two aspects: the bit-parallel access scheme (i.e., blackucing the number of GDC cycles requiblack for an access) and the sub-arrayed structure (i.e., blackucing the time requiblack for reading out a stoblack set of data in a sub-array). How-

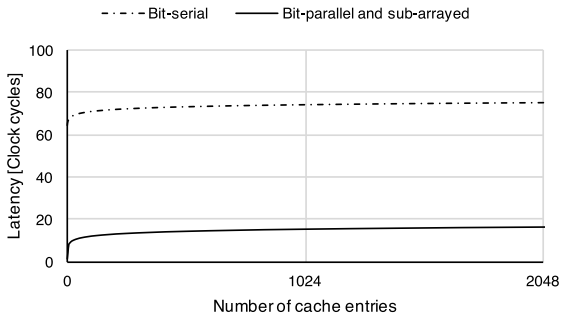


Fig. 7 Comparison between sub-arrayed cache and bit-serial cache in terms of access latency.

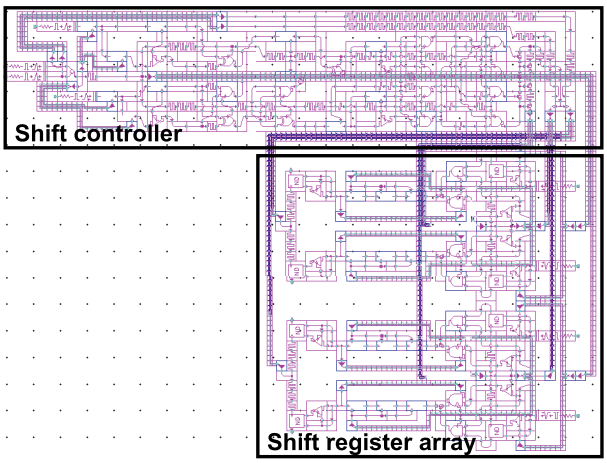


Fig. 8 Physical schematic of a shift controller and a shift register array.

ever, the number of JJs requiblack to implement our cache is 0.62 M, and the estimated area is 543.6 mm^2, 2.1 times larger than the baseline. Table 2 indicates that extra hardware (e.g., shift controller and multiplexer to select data from sub-arrays output) for supporting the bit-parallel access scheme and the sub-arrayed structure account for over 60%. The power consumption is estimated at 240.9 mW.

Figure 7 shows the comparison of access latency (clock cycles) when the cache capacity becomes larger. To simplify the sensitivity analysis, we have assumed that the number of cache entries included in each sub-array is fixed to eight. The result indicates that our proposed cache is much faster than the bit-serial cache even if the cache capacity becomes large scale.

4.4 Prototype Design

In addition to the evaluation, we have done the practical design of a prototype of the shift controller and the shift register array, which are the parts of the SFQ sub-arrayed cache. The purpose of the design is to confirm the operation of the main components of the sub-arrayed cache. There are only two components (the shift controller and the shift register arrays), and the selector has already been designed. This design has been done to confirm the feasibility of the shift controller, which is the most important unit for realizing random access by controlling the number of shifts in the sub-arrays. Figure 8 shows the circuit diagram of a scaled-down shift controller and shift register array. The shift register array consists of four entries in which each entry can be stoblack as 4-bit data. The controller consists of a 2-bit subtractor, a pulse generator and a buffer to save an address of previous access. We have designed the circuit by using the CONNECT cell library [27] and confirmed the correct operation of the shift controller by using the Verilog-XL simulator.

5. Conclusions

In this paper, we have introduced our recent research results focusing on ultra-high-speed cryogenic SFQ computing. To prove the correctness of our architectural direction, we have designed a bit-parallel gate-level pipelined ALU targeting 50-GHz operation. Although its word size is 8 bits, and it only supports simple functions, we have successfully demonstrated its 56-GHz operation at 1.6 mW. Our ongoing work is to design the proposed bit-parallel gate-level pipelined multithreaded SFQ microprocessor to confirm its correct operation. Another part of our future work is to develop a CMOS-SFQ hybrid memory sub-system and to integrate all of the components to construct a next generation latency-oriented computing platform.

Acknowledgments

This work was supported in a part by JSPS KAKENHI Grant Number JP16H02796.

References

[1] R. Harris, M.W. Johnson, T. Lanting, A.J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M.C. Thom, E.M. Chapple, J. Wang, B. Wilson, M.H.S. Amin, N. Dickson, K. Karimi, B. Macready, C.J.S. Truncik, and G. Rose, "Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor," Phys. Rev. B, vol.82, p.024511, July 2010.

[2] M. Johnson, M.H.S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A.J. Berkley, J. Johansson, P. Bunyk, E.M. Chapple, C. Enderud, J.P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M.C. Thom, E. Tolkacheva, C.J.S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, "Quantum annealing with manufactured spins," vol.473, pp.194–198, May 2011.

[3] 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.

[4] 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," Superconductor Science and Technology, vol.19, no.5, pp.S344–S349, 2006.

[5] 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, core 1β," IEEE Transactions on Applied Superconductivity, vol.17,

no.2, pp.474–477, 2007.

[6] M. Tanaka, Y. Yamanashi, N. Irie, H. Park, S. Iwasaki, K. Takagi, K. Taketomi, A. Fujimaki, N. Yoshikawa, H. Terai, and S. Yorozu, "Design and implementation of a pipelined 8 bit-serial single-flux-quantum microprocessor with cache memories," Superconductor Science and Technology, vol.20, no.11, pp.S305–S309, 2007.

[7] 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 Transactions on Applied Superconductivity, vol.27, no.4, pp.1–5, June 2017.

[8] K. Ishida, M. Tanaka, T. Ono, and K. Inoue, "Exploring design space of a single-flux-quantum microprocessor (In Japanese)," IPSJ Journal, vol.58, no.3, 2017 in press.

[9] K. Ishida, M. Tanaka, T. Ono, and K. Inoue, "Single-flux-quantum cache memory architecture," 2016 International SoC Design Conference (ISOCC), pp.105–106, Oct. 2016.

[10] T. Tsuhata, J. Yokota, K. Inoue, and M. Tanaka, "Architectural design space exploration of single-flux-quantum microprocessors," Superconducting SFQ VLSI Workshop for Young Scientists, pp.18–21, 2014.

[11] K. Ishida, M. Tanaka, T. Ono, and K. Inoue, "Logic design of a single-flux- quantum gate-level-pipelined microprocessor," Superconducting SFQ VLSI Workshop, pp.6–12, 2017.

[12] A. Fujimaki, Y. Takai, and N. Yoshikawa, "High-end server based on complexity-reduced architecture for superconductor technology," IEICE Transactions on Electronics, vol.E85-C, no.3, pp.612–616, March 2002.

[13] S. Nagasawa, K. Hinode, T. Satoh, M. Hidaka, H. Akaike, A. Fujimaki, N. Yoshikawa, K. Takagi, and N. Takagi, "Nb 9-layer fabrication process for superconducting large-scale sfq circuits and its process evaluation," IEICE Transactions on Electronics, vol.E97-C, no.3, pp.132–140, 2014.

[14] S.K. Tolpygo, V. Bolkhovsky, T.J. Weir, A. Wynn, D.E. Oates, L.M. Johnson, and M.A. Gouker, "Advanced fabrication processes for superconducting very large-scale integrated circuits," IEEE Transactions on Applied Superconductivity, vol.26, no.3, pp.1–10, April 2016.

[15] D.E. Kirichenko, S. Sarwana, and A.F. Kirichenko, "Zero static power dissipation biasing of rsfq circuits," IEEE Transactions on Applied Superconductivity, vol.21, no.3, pp.776–779, June 2011.

[16] M.H. Volkmann, A. Sahu, C.J. Fourie, and O.A. Mukhanov, "Implementation of energy efficient single flux quantum digital circuits with sub-aj/bit operation," Superconductor Science and Technology, vol.26, no.1, p.015002, 2013.

[17] M. Tanaka, M. Ito, A. Kitayama, T. Kouketsu, and A. Fujimaki, "18-ghz, 4.0-aj/bit operation of ultra-low-energy rapid single-flux-quantum shift registers," Japanese Journal of Applied Physics, vol.51, no.5R, p.053102, 2012.

[18] Q.P. Herra, A.Y. Herr, O.T. Oberg, and A.G. Ioannidis, "Ultra-low-power superconductor logic," Journal of Applied Physics, vol.109, no.10, p.103903, 2011.

[19] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," Superconductor Science and Technology, vol.26, no.3, p.035010, 2013.

[20] A. Herr, B. Konigsberg, R. Clarke, M. Vesely Jr., P. Farrell, P. Tschirhart, J. Egan, J. Strong, M. Alvarado, B. Song, K. Ogg, and Q. Herr, "Reciprocal quantum logic cpus for energy efficient high performance computing," International Superconductive Electronics Conference, 2017.

[21] A. Kirichneko, M. Miller, I. Vernik, O. Mukhanov, L. Albu, and G. Gibson, "Energy-ecient dual-port 32-word 8-bit ersfq register file," 13th European Conference on Applied Superconductivity (EUCAS 2017), 2017.

[22] I.V. Vernik, V.V. Bol'ginov, S.V. Bakurskiy, A.A. Golubov, M.Y. Kupriyanov, V.V. Ryazanov, and O.A. Mukhanov, "Magnetic josephson junctions with superconducting interlayer for cryogenic memory," IEEE Transactions on Applied Superconductivity, vol.23, no.3, pp.1701208–1701208, June 2013.

[23] A. Herr and Q. Herr, "Josephson magnetic random access memory system and method," Sept. 18 2012. US Patent 8, 270, 209.

[24] A.N. McCaughan and K.K. Berggren, "A superconducting-nanowire three-terminal electrothermal device," Nano Letters, vol.14, no.10, pp.5748–5753, 2014. PMID: 25233488.

[25] M. Tanaka, M. Suzuki, G. Konno, Y. Ito, A. Fujimaki, and N. Yoshikawa, "Josephson-cmos hybrid memory with nanocryotrons," IEEE Transactions on Applied Superconductivity, vol.27, no.4, pp.1–4, June 2017.

[26] O.A. Mukhanov, "Energy-efficient single flux quantum technology," IEEE Transactions on Applied Superconductivity, vol.21, no.3, pp.760–769, June 2011.

[27] Y. Yamanash, 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 Transactions on Electronics, vol.E93-C, no.4, pp.440–444, June 2010.

[28] A.M. Kadin, C.A. Mancini, M.J. Feldman, and D.K. Brock, "Can rsfq logic circuits be scaled to deep submicron junctions?," IEEE Transactions on Applied Superconductivity, vol.11, no.1, pp.1050–1055, 2001.

[29] P. Bunyk and P. Litskevitch, "Case study in rsfq design: fast pipelined parallel adder," IEEE Transactions on Applied Superconductivity, vol.9, no.2, pp.3714–3720, June 1999.

[30] M. Tanaka, K. Takata, R. Satoh, A. Fujimaki, T. Kawaguchi, Y. Ando, K. Takagi, N. Takagi, and N. Yoshikawa, "Design of rsfq microprocessors integrated with rams based on bit-serial processing," 7th Superconducting SFQ VLSI Workshop, 2014.

[31] B. Flachs, S. Asano, S.H. Dhong, H.P. Hofstee, G. Gervais, R. Kim, T. Le, P. Liu, J. Leenstra, J. Liberty, B. Michael, H.-J. Oh, S.M. Mueller, O. Takahashi, A. Hatakeyama, Y. Watanabe, N. Yano, D.A. Brokenshire, M. Peyravian, V. To, and E. Iwata, "The microarchitecture of the synergistic processor for a cell processor," IEEE Journal of Solid-State Circuits, vol.41, no.1, pp.63–70, Jan. 2006.

[32] H. Ogihara, Survey of Cryogenic Engineering (Japanese), Tokyo Denki University, 1999.

[33] D.E. Kirichenko, S. Sarwana, and A.F. Kirichenko, "Zero static power dissipation biasing of rsfq circuits," IEEE Transactions on Applied Superconductivity, vol.21, no.3, pp.776–779, June 2011.

[34] X. Peng, Q. Xu, T. Kato, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, and M. Hidaka, "High-speed demonstration of bit-serial floating-point adders and multipliers using single-flux-quantum circuits," IEEE Transactions on Applied Superconductivity, vol.25, no.3, pp.1–6, June 2015.

[35] S. Tahara and Y. Wada, "A vortex transitional ndro josephson memory cell," The Japan Society of Applied Physics, vol.26, no.9, pp.1463–1466, 1987.

[36] K. Fujiwara, Y. Yamashiro, N. Yoshikawa, A. Fujimaki, H. Terai, and S. Yorozu, "Design and high-speed test of (4 × 8)-bit single-flux-quantum shift register files," Superconductor Science and Technology, vol.16, no.12, pp.1456–1459, 2003.

Koki Ishida received his B.E. from Kyushu University, Japan, in 2016. He is currently a masters student in the Graduate School of Information Science and Electrical Engineering at Kyushu University. His research interests include the the computer architecture of ultra-fast/energy-efficient computing using SFQ devices. He is a member of the IEICE, IPSJ, and IEEE.

Masamitsu Tanaka received his M.E. and Ph.D. in electronics and information electronics from Nagoya University, Nagoya, Japan, in 2003 and 2006, respectively. He was a JSPS Research Fellow from 2005 to 2007. He joined the Graduate School of Information Science, Nagoya University, in 2007, and moved to the Graduate School of Engineering in 2010, where he is currently an assistant professor. In 2011 he was a research scholar at the University of California, Berkeley, CA, USA. His research interests include ultra-fast/energy-efficient computing using RSFQ circuits and logic design methodologies. He is a member of the IEICE, IEEE, Japan Society of Applied Physics, Cryogenics and Superconductivity Society of Japan, and Institute of Electrical Engineers of Japan.

Takatsugu Ono received his Ph.D. from Kyushu University, Japan, in 2009. He was a researcher for Fujitsu Laboratories Ltd., Kawasaki, Japan, and engaged in developing a server for a data center. He is currently an assistant professor in the Faculty of Information Science and Electrical Engineering at Kyushu University. His research interests include the areas of memory architecture, secure architecture, and supercomputing. He is a member of the IEICE, IPSJ, and IEEE.

Koji Inoue received his B.E. and M.E. in computer science from the Kyushu Institute of Technology, Japan, in 1994 and 1996, respectively. He received his Ph.D. from the Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan in 2001. In 1999, he joined Halo LSI Design & Technology, Inc., NY, as a circuit designer. He is currently a professor of the Department of I&E Visionaries, Kyushu University. His research interests include power-aware computing, high-performance computing, secure computer systems, 3D microprocessor architectures, multi/many-core architectures, nanophotonic computing, and quantum computing.