Demonstration of a Single-Flux-Quantum Microprocessor Using Passive Transmission Lines

M. Tanaka, T. Kondo, N. Nakajima, T. Kawamoto, Y. Yamanashi, Y. Kamiya, A. Akimoto, A. Fujimaki, H. Hayakawa, N. Yoshikawa, H. Terai, Y. Hashimoto, and S. Yorozu

***Abstract*—We have successfully demonstrated an 8-bit microprocessor using passive transmission lines based on single-flux-quantum LSI technology. In the microprocessor designed here, called CORE1α6, a simple bit-serial architecture with seven instructions was employed. In the CORE1α6, the floor plan was designed with consideration toward integration of a memory, and superconductive passive transmission lines (PTLs) were used to connect circuit blocks. Utilization of PTLs helped us reduce the propagation delay in long interconnections. The design flexibility of the floor plan was enhanced and the performance of the microprocessor was improved by 20% compared with our previous design. The CORE1α6 was composed of 6319 Josephson junctions and 15 PTLs with power consumption of 2.1 mW. We have confirmed the complete operations of the CORE1α6 by on-chip high-speed tests. The maximum clock frequencies for bit operation and instruction execution have been found to be 18 GHz and 1.2 GHz, respectively, where the performance corresponds to 240 million instructions per second (MIPS).**

***Index Terms*—Microprocessor, passive transmission line, single flux quantum (SFQ) logic.**

I. Introduction

THE single-flux-quantum (SFQ) logic circuit [1] features high-speed operation beyond several tens of GHz with extremely low-power consumption because it utilizes ballistic propagation of impulse-shaped voltage signals of SFQ. In the SFQ logic circuit, passive transmission lines (PTLs) are also available as interconnections. An SFQ travels in Superconductive PTLs at the speed of light without dissipation or dispersion. These excellent features indicate that the SFQ logic is a strong candidate for future LSI technology toward high-end digital applications, when we consider the crises of CMOS LSIs: the high power density and large interconnection delays. Several institutes have already started to study high-end SFQ microprocessors such as FLUX chip by TRW and State University of New York, Stony Brook [2].

We have been studying the microprocessors based on SFQ LSI technology for high-end network servers. As the first implementation, we have been developing an 8-bit SFQ microprocessor with simple, bit-serial architecture, called CORE1α. The main purposes of developing CORE1α are to demonstrate the feasibility of such a large-scale, complex digital system by using SFQ LSI technology, and to evaluate the bit-serial architecture through the actual design. The performance, power consumption and circuit size of the CORE1α microprocessor will be fundamental information for us to investigate architectures in the future, because the bit-serial processing employed here is a direct approach of our complexity-reduced (CORE) concept [3], where the high-throughput nature of SFQ logic is utilized to ease the system complexity.

We designed a prototype chip of CORE1α and succeeded in the demonstration so far [4], [5]. The prototype, called CORE1α5, was made up of minimal essential components without any consideration for a memory. The CORE1α5 operated at 15.2 GHz with power consumption of 1.6 mW. The performance was estimated to be 167 million instructions per second (MIPS).

As the next step to integrate a memory, we have designed a new version of the CORE1α microprocessor, CORE1α6. The purpose of this design is to modify the floor plan with consideration for an SFQ shift-register memory [6]. Although a memory is not implemented yet in the CORE1α6, the microprocessor has a space for it and some components to control data streams between the memory and registers. One of major differences from the CORE1α5 is utilization of PTLs to connect circuit blocks. We used Josephson transmission lines (JTLs) for all interconnections in the CORE1α5. Introduction of PTLs leads to the flexibility to design the floor plan of the CORE1α6, because an SFQ can propagate PTLs over ten times faster than JTLs. In addition, the CORE1α6 is designed to be tested with high-speed system clock. The CORE1α microprocessor uses two clocks: system clock and local clock. The former corresponds to the system cycles, which is utilized to control instruction execution. The later is for bit operations such as trans-

Manuscript received October 5, 2004. This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) through ISTEC as a Collaborative Research and Superconductors Network Device Project.

M. Tanaka, T. Kondo, T. Kawamoto, Y. Kamiya, A. Fujimaki, and H. Hayakawa are with Nagoya University, Nagoya, Japan (e-mail: masami_t@super.nuqe.nagoya-u.ac.jp).

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

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

Y. Hashimoto and S. Yorozu are with the Superconductivity Research Laboratory, International Superconductivity Technology Center, Ibaraki, Japan (e-mail: hasimoto@istec.or.jp).

Digital Object Identifier 10.1109/TASC.2005.849860

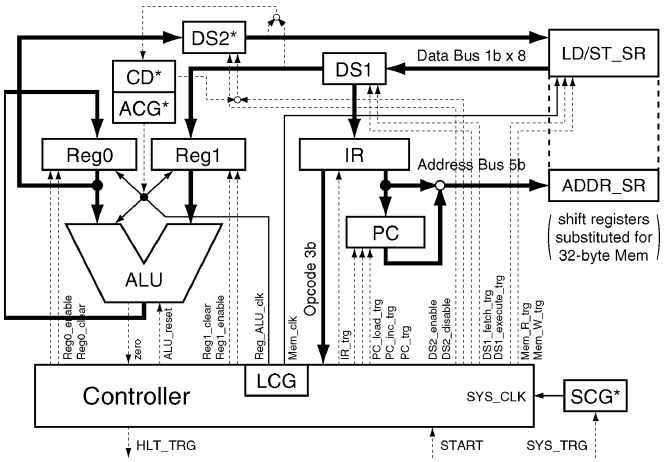

Fig. 1. The microarchitecture of the CORE1α6. The thick lines represent the paths for instruction and/or data. Excepting the address bus and the paths for opcode, the instruction and data are handled bit-serially. The thin solid lines are for the local clocks and the dashed lines for the control signals. The components marked asterisks are added newly at this version of the CORE1α microprocessor.

ferring and calculating bit-serial data. In the demonstration of the CORE1α5, we were obliged to estimate the performance based on simulation, because the system clocks were low speed, while we confirmed successful operation with 15.2 GHz local clock. Using high-speed system clocks together with high-speed local clocks, we can estimate the performance of the CORE1α6 experimentally.

In this paper, we describe the design of the CORE1α6 with these improvements, and report the results of the microprocessor by high-speed on-chip tests.

II. Design of the CORE1α6 Microprocessor

A. Modification of Floor Plan

Fig. 1 is a microarchitecture of the CORE1α6. The microprocessor is composed of a controller, a 5-bit program counter (PC), an 8-bit instruction register (IR), two 8-bit registers (Reg0, Reg1), a bit-serial ALU, data selectors (DS1, DS2), a clock detector (CD), and three clock generators (the local, system, and adjusting clock generators abbreviated as LCG, SCG, and ACG, respectively). Two shift registers substituted for a memory (the load/store and address shift registers as LD/ST_SR and ADDR_SR) are also integrated on the chip. LD/ST_SR is a 16-bit shift register that holds an instruction and data. The CORE1α6 loads an instruction and data into IR and Reg1, respectively, by switching DS1 in compliance with the controller. LCG triggered by the controller generates local clocks, which are required for the data transfer or register operations in the ALU. In the CORE1α microarchitecture, the local clock train is sent to the appropriate component, and then transmitted to the destination accompanying data. In other words, in the instruction fetch and the Load operation, the local clocks go to LD/ST_SR, and then arrive at IR and Reg1 with the instruction and data, respectively. In the Store operation, the local clocks flow from the Reg0 to LD/ST_SR with data.

TABLE I
Instruction Set of CORE1α6

Instruction	Name	Meaning	Opcode
HLT	Halt	Stop	000
ADD	Add	Reg0 $\leftarrow$ Reg0 + Reg1	111
LD	Load	Reg1 $\leftarrow$ Mem[address]	011
ST	Store	Mem[address] $\leftarrow$ Reg1	100
SKZ	Skip if zero	if (Reg0 $==$ 0) PC $\leftarrow$ Pc + 2	101
JMP	Jump	PC $\leftarrow$ address	001
MOVE	Move	Reg0 $\leftarrow$ Reg01	110

The target frequency of the local clocks is 16 GHz. ADDR_SR is a 5-bit shift register that contains an address pointed during a memory access. The following components are newly added at this version of the CORE1α microprocessor: DS2, CD and ACG to control data streams between the registers and a memory, and SCG to generate high-speed system clocks for on-chip testing. We will describe these components later.

The CORE1α microprocessor has seven instructions including data transfers (Load, Store), register operations (Add, Move), conditional/unconditional branches (Skip if zero, Jump), and Halt. Table I summarizes the instruction set of the CORE1α6. We employed minimal instruction set to CORE1α in order to keep the microprocessor small.

We designed the floor plan of these 14 components taking the integration of a memory into account. A memory is a relatively large component. Through designing the floor plan, we found several long interconnections over 1 mm appeared on the CORE1α6 chip. We estimated that the large delays of long interconnections would bring about 25% worse performance of the CORE1α6 compared with CORE1α5 because of lowered frequency of the system clocks.

To overcome this problem, we have employed PTLs to connect components in CORE1α6. In general, implementing superconductive PTLs is difficult because of mismatch of impedance between PTLs and Josephson junctions. Recently, reliable PTL interconnecting technology using microstrip line with characteristic impedance of 2 Ω has been developed in our cell library [7], [8]. Introduction of PTLs helps decrease propagation delays in large interconnections and enhance the design flexibility of the floor plan. Simultaneously, utilization of PTLs contributes to the reduction of the number of Josephson junctions and power consumption of the system. For instance, the propagation delay between DS1 and IR, which is the longest path on the chip, is reduced from 400 ps to 34 ps, and the power consumption of the microprocessor is decreased by greater than 10%. The estimated performance of the designed CORE1α6 is improved by 20% from the CORE1α5. This is because the number of system clock cycles per instruction is changed from 6 to 5 in the new design as keeping the frequency of the system clock 1 GHz. In other words, the CORE1α6 executes each instruction in 5 ns while the CORE1α5 does in 6 ns.

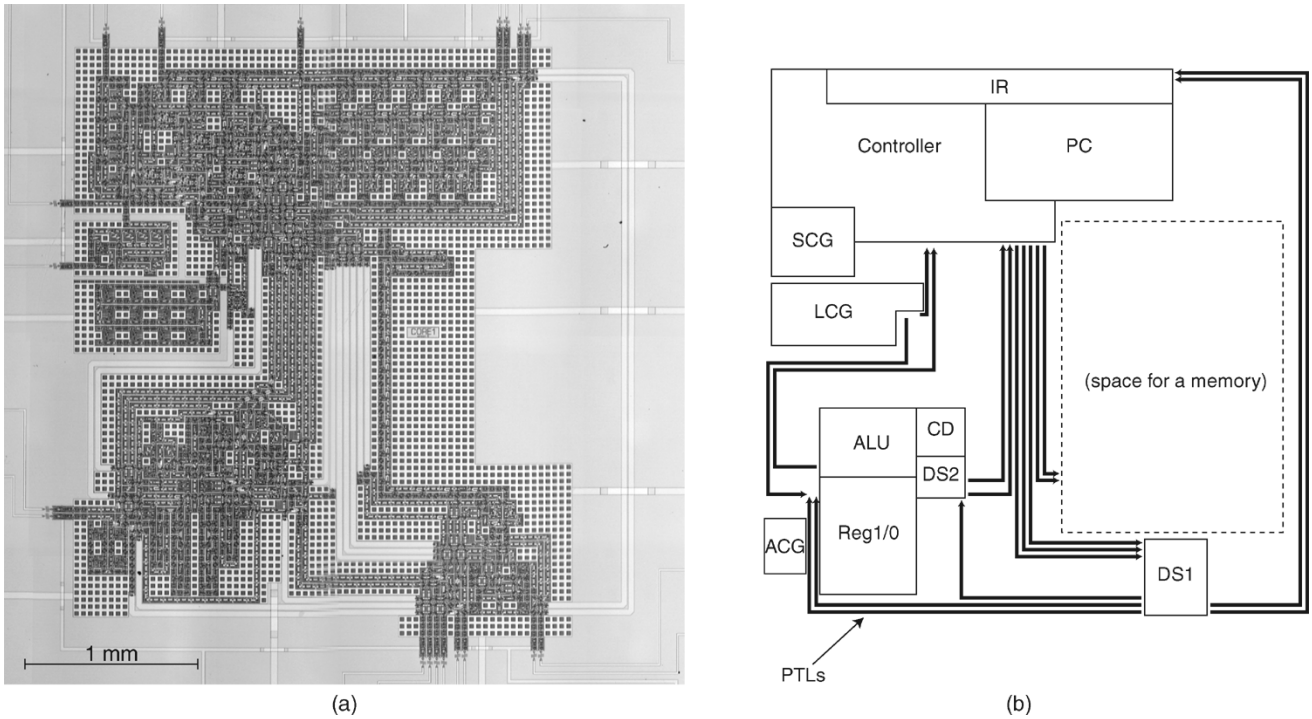

Fig. 2. The chip photograph (a) and floorplan (b) of CORE1α6 microprocessor. The size of the circuit is 3.4 mm $\times$ 3.2 mm. In the photograph, the white lines corresponding to the thick lines in the floorplan, are PTLs, which are superconductive microstrip lines with width of 34 μm. Their characteristic impedance and propagation delay are 2 Ω and 8.6 ps/mm, respectively. Although a memory is not implemented yet, the microprocessor has a space for it. The shift registers substituted for the memory are located at that space.

B. Additional Components

Here we describe some additional components of CORE1α6 marked asterisk in Fig. 1.

The system clock generator (SCG) generates 1-GHz system clock signals advancing instruction execution phases. We can examine the performance of the CORE1α6 experimentally by using SCG. SCG is designed to generate 1-GHz pulses when triggered by the external signal. The circuit is a combination of a 1/16 frequency divider and a 16-GHz programmable clock generator. The programmable clock generator is composed of a ring oscillator, a nondestructive read-out (NDRO) cell and a counter to control the ring [9], which is set to output five pulses. Utilization of the 16-GHz ring oscillator leads to the small area of SCG.

The data selector called DS2 works as a gate, which controls the local clock train and bit-serial data transfer from Reg1 to LD/ST_SR. When DS2 is switched to "enable" state for the Store instruction, DS2 passes the data. DS2 is a simple component composed of a pair of NDRO cells.

CD (clock detector) and ACG (adjusting clock generator) are added to adjust data format so as to fit a memory on the registers. In the CORE1α, both instruction and data are 8 bits wide and so is the memory. Eight local clocks are used for Load and Store operations. On the other hand, the register operations such as Add and Move requires 12 local clocks, because there are 12 stages at the 8-bit registers and 4-stage ALU, where we employed the synchronous design in order to reduce latencies of the registers and ALU [10]. In the CORE1α6, the local clock trains with appropriate length are obtained by the improved LCG, which is composed of a ladder circuit and a delayed flip-flop. It can be set to generate 8 or 12 SFQ pulses. Using this variable LCG, however, the data in registers shifts by 4 bits after Load or Store operation. CD and ACG are designed to correct this inconsistency. In the instructions of data transfer between the memory and registers, CD detects the local clocks and triggers ACG. ACG generates four additional local clocks and send it to the registers and ALU.

III. Test Results

We have designed the CORE1α6 using the CONNECT cell library [11], [12]. Fig. 2 is the photograph of the CORE1α6 chip, which was fabricated using NEC's Nb standard process [13]. The microprocessor is made up of 6319 Josephson junctions and 15 PTLs. The power consumption is 2.1 mW.

Functionality of the microprocessor has been tested on the chip with high-speed clocks. We confirmed the complete operations for every instruction by testing several test sequences.

Fig. 3 shows one of the results, where successful addition of two integers is done with sequential instructions of the Load, Move, Load, Add, Store. After initializing the microprocessor at the part of (0) in the figure, we wrote the Load instruction and an integer 0100 1101 into LD/ST_SR (1). In this figure, the data start with LSB. We executed the instruction by triggering SCG to generate a high-speed system clock train. The local clocks controlled by the system clocks are automatically generated on the chip when needed. These clocks are so fast that

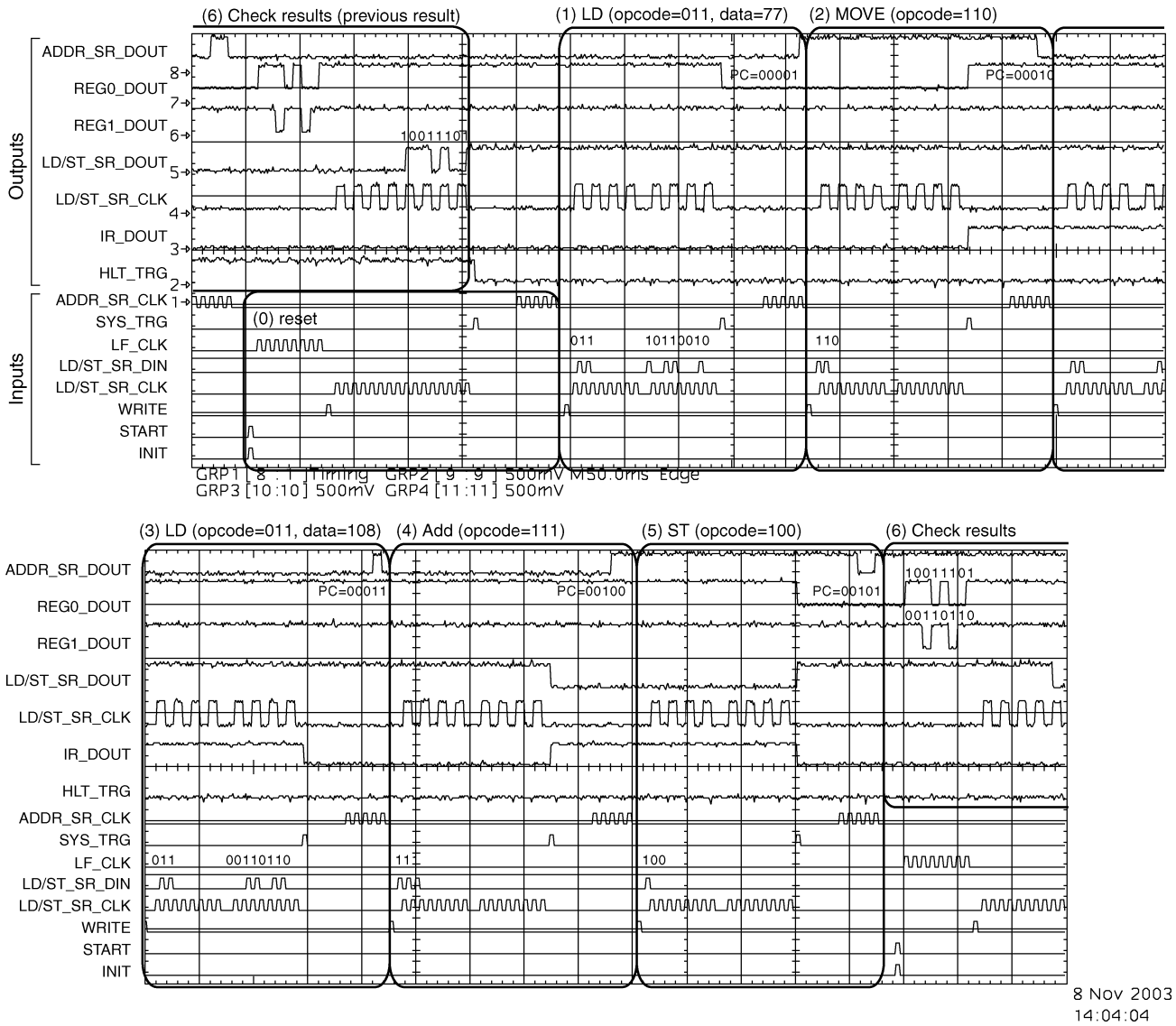

Fig. 3. The results of the addition of two integers with high-speed clocks. The eight waveforms at the bottom correspond to input signals, and an SFQ pulse is generated at each of their rising edges. The upper seven waveforms are outputs, whose signal transitions represent the arrival of the SFQ pulses. In this figure, data (LD/ST_SR_DIN, IR_DOUT, LD/ST_SR_DOUT, REG0_DOUT, and REG1_DOUT, which are the input or output signals of the corresponding components) start with LSB. On the contrary, an address (ADDR_SR_DOUT, which is an output of ADDR_SR) does with MSB. We can write an instruction and data to the LD/ST_SR by LD/ST_SR_DIN and LD/ST_SR_CLK after sending WRITE signal. To execute the instruction, we feed SYS_TRG, by which SCG generates high-speed system clocks. After each execution, we read ADDR_SR from ADDR_SR_DOUT with using ADDR_SR_CLK to check the address pointed by PC. The START and INIT, and LF_CLK (low-frequency clock) are used to initialize the microprocessors. The former two inputs reset the controller and other components, respectively. The latter flushes out the registers. The HLT_TRG is generated when the HLT instruction is executed.

we cannot observe them in Fig. 3. The integer was loaded into Reg1. Then we copied Reg1 to Reg0 with the Move instruction (2), loaded another integer 0110 1100 to Reg1 (3), and added the two integers (4) as well. Finally, we wrote back the result to LD/ST_SR with the Store (5), and send low-frequency clocks to read it. We can see the successful operation of the CORE1α6 by the correct result of the addition, 1011 1001, from both Reg0 and LD/ST_SR. (In Fig. 3, the result is confirmed at the upper left, which is obtained from the same test sequence executed before (1).)

We have examined the operating region of the CORE1α6 for the supplied dc bias current at different clock frequencies shown in Fig. 4. We can set the frequencies of local and system clocks independently by changing the bias currents for LCG and SCG, where the bias currents are supplied separately from each other and the other components. We have estimated the frequencies using the analog simulator [14] based on each supplied bias current. The test sequence used in Fig. 4 is an addition of two integers shown in Fig. 3. In this sequence, we have found that the successful bit operations were performed with the local clocks from 11 GHz to 18 GHz. As for the system clock, we have examined the maximum frequency of 1.2 GHz, where we confirmed the performance of 240 MIPS. At the target frequencies, 16 GHz for the local clock and 1 GHz for the system clock, the dc bias margin of the CORE1α6 ranges from -7.2% to $+2.9\%$ against the designed point.

Fig. 4 represents the typical tendency of the operating regions of CORE1α6. The maximum frequency of the system clock be-

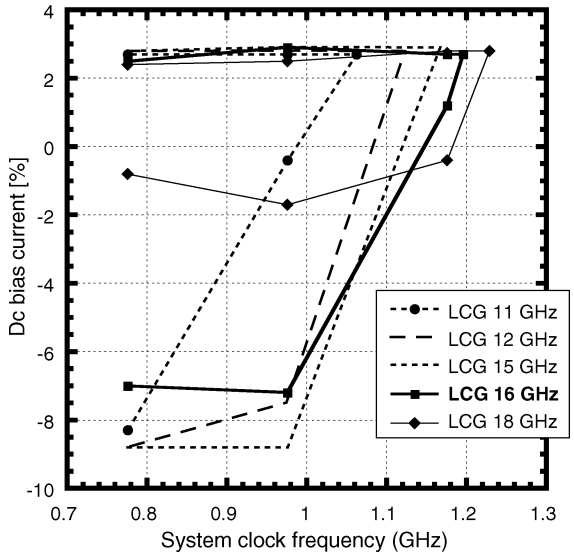

Fig. 4. The dependence of the operating range on the clock frequencies of both system and local clocks. The areas surrounded by upper and lower lines indicate the operating regions for different local clock frequencies. The dc bias currents are normalized by the designed value. The target frequencies of the system and local clocks are 16 GHz and 1.0 GHz, and their maximum frequencies were found to be 18 GHz and 1.2 GHz, respectively. At the target frequencies, the operating region ranges from -7.2% to $+2.9\%$.

comes higher as the local clock frequency faster. It suggests that the period between phases of instruction execution is shortened because of the time reduction in bit-serial operations. In addition, at 18-GHz local clocks, the microprocessor does not work with lower bias currents. In this case, we think that the lower margin is limited by the critical path of the local clocks, which is the data transferring path between the registers and ALU. These results agree with those of our simulation.

IV. Conclusion

We have succeeded in demonstration of the SFQ microprocessor using PTLs. The microprocessor demonstrated here was designed with the new floor plan with consideration toward integration of a memory. We have confirmed the complete operations of the microprocessor by on-chip high-speed tests. The maximum frequencies of the local and system clocks were found to be 18 GHz and 1.2 GHz, respectively, and the performance of 240 MIPS was examined experimentally. We have also demonstrated that the reduction of the delay in the long interconnection by using PTLs is highly effective in designing the SFQ microprocessor. Introduction of PTLs gave the flexibility for designing the floor plan, and improved the performance of the microprocessor by 20% compared with the previous design.

Acknowledgment

The authors used the CONNECT cell library and tools for designing the CORE1α6 chip. They 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] M. Dorojevets, P. Bunyk, and D. Zinoviev, "FLUX chip: Design of a 20-GHz 16-bit ultrapipelined RSFQ processor prototype based on 1.75-μm LTS technology," *IEEE Trans. Appl. Supercond.*, vol. 11, pp. 326–332, Mar. 2001.

[3] 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, Mar. 2002.

[4] M. Tanaka, F. Matsuzaki, T. Kondo, N. Nakajima, Y. Yamanashi, H. Terai, S. Yorozu, N. Yoshikawa, A. Fujimaki, and H. Hayakawa, "Prototypic design of the single-flux-quantum microprocessor, CORE1," *Supercond. Sci. Technol.*, vol. 16, no. 12, pp. 1460–1463, Dec. 2003.

[5] 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 *Tech. Dig. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 2004.

[6] 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," *Supercond. Sci. Technol.*, vol. 16, no. 12, pp. 1456–1459, Dec. 2003.

[7] Y. Hashimoto, S. Yorozu, Y. Kameda, and V. K. Semenov, "A design approach to passive interconnects for single flux quantum logic cells," *IEEE Trans. Appl. Supercond.*, vol. 13, no. 2, pp. 535–538, Jun. 2003.

[8] Y. Hashimoto, S. Yorozu, Y. Kameda, A. Fujimaki, H. Terai, and N. Yoshikawa, "Demonstration of passive interconnection between single-flux-quantum circuit blocks," in *Extended Abstracts 9th Int. Supercond. Electron. Conf.*, Sydney, Australia, Jul. 2003.

[9] M. Ito, N. Nakajima, K. Fujiwara, N. Yoshikawa, A. Fujimaki, H. Terai, and S. Yorozu, "Design and implementation of SFQ programmable clock generators," *Physica C*, pt. 2, vol. 412–414, pp. 1550–1554, to be published.

[10] M. Tanaka, T. Kondo, A. Sekiya, A. Fujimaki, H. Hayakawa, F. Matsuzaki, N. Yoshikawa, H. Terai, and S. Yorozu, "Component test toward single-flux-quantum processors," *Physica C*, pt. 2, vol. 392–396, pp. 1490–1494, Oct. 2003.

[11] S. Yorozu, Y. Kameda, H. Terai, A. Fujimaki, T. Yamada, and S. Tahara, "A single flux quantum standard logic cell library," *Physica C*, pt. 2, vol. 378–381, pp. 1471–1474, Oct. 2002.

[12] N. Yoshikawa, T. Nishigai, H. Kojima, K. Fujiwara, A. Fujimaki, T. Yamada, M. Tanaka, S. Yorozu, M. Hidaka, and H. Terai, "Magnetic shielding against DC bias current toward large-scale SFQ integrated circuits," in *Appl. Supercond. Conf.*, Jacksonville, FL, Oct. 2004.

[13] 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.

[14] E. S. Fang and T. V. Duzer, "A Josephson Integrated Circuit Simulator (JSIM) for superconductive electronics application," in *Extended Abstracts Int. Supercond. Electron. Conf.*, Tokyo, Japan, Jun. 1989.