Performance Evaluations of Finite Difference Applications Realized on a Single Flux Quantum Circuits-Based Reconfigurable Accelerator

Hiroaki Honda*, Farhad Mehdipour†, Hiroshi Kataoka*, Koji Inoue*, and Kazuaki J. Murakami*
* Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan
E-mail: {dahon, kataoka}@soc.ait.kyushu-u.ac.jp, {inoue, murakami}@ait.kyushu-u.ac.jp
† Center for Japan-Egypt Cooperation in Science and Technology, Kyushu University, Fukuoka, Japan
E-mail: farhad@ejust.kyushu-u.ac.jp

***Abstract*—Hardware accelerators integrating to general purpose processors are increasingly employed to achieve lower power consumption and higher processing speed, however, energy consumption of high performance accelerators has become a great issue on large scale parallel computer system. We have investigated the applicability of Single-Flux-Quantum (SFQ) circuits as a part of superconductivity technology in high-performance computing systems. Although it is possible to develop extraordinary low power processor by SFQ devices, conditional branch and loop back controls are difficult to be implemented by current SFQ technology. Therefore, we have proposed Reconfigurable Data-Path (RDP) accelerator which is avoiding those limitations of SFQ technology, while trying to get benefits of these circuits. In this research, we have implemented two-dimensional Heat (2D-Heat) and Finite Difference Time Domain (2D-FDTD) applications for investigating efficiency of using SFQ-RDP accelerator. According to performance evaluation results for above applications, execution times are 50.6 and 79.0 times smaller than those of the general purpose processor, and comparable with ones reported for GPU (Graphics Processing Units).Hardware accelerators integrating to general purpose processors are increasingly employed to achieve lower power consumption and higher processing speed, however, energy consumption of high performance accelerators has become a great issue on large scale parallel computer system. We have investigated the applicability of Single-Flux-Quantum (SFQ) circuits as a part of superconductivity technology in high-performance computing systems. Although it is possible to develop extraordinary low power processor by SFQ devices, conditional branch and loop back controls are difficult to be implemented by current SFQ technology. Therefore, we have proposed Reconfigurable Data-Path (RDP) accelerator which is avoiding those limitations of SFQ technology, while trying to get benefits of these circuits. In this research, we have implemented two-dimensional Heat (2D-Heat) and Finite Difference Time Domain (2D-FDTD) applications for investigating efficiency of using SFQ-RDP accelerator. According to performance evaluation results for above applications, execution times are 50.6 and 79.0 times smaller than those of the general purpose processor, and comparable with ones reported for GPU (Graphics Processing Units).**

I. Introduction

In various scientific areas such as fluid dynamics, computational chemistry, materials science, environmental issues and etc., complex numerical computations are indispensable which necessitate employing quite powerful computers. Providing high computational power to individual researchers is crucial for progress of the research and development. Large scale parallel computer systems with General Purpose Processors (GPPs) are often utilized as supercomputer. Although, continuing advances in manufacturing processes have made it possible for processor vendors to build increasingly faster, there is still a high demand to meet the required performance for specific applications. On the other hand, concept of green High Performance Computing (HPC) has been paid more attention in recent years for eco-friendly computing. In this point of view, computer accelerator is important for its computational efficiency per energy consumption. Examples of such accelerator are CSX600 PCI-X board [1], GRAPE-DR processor [2], Cell processor [3] which is heterogeneous multi-core processor and GPGPU for General Purpose Graphics Processing [4]. Especially, GPU is one noteworthy candidate base architecture for realization of exa-scale computer system [5].

For exa-scale or farther scale computing, utilization of new devices becomes another breakthrough to overcome the large energy consumption problem of large scale HPC. Superconducting Rapid Single-Flux-Quantum (RSFQ) circuit technology is expected as one of such next generation circuit technologies which enables ultra high-speed computation with ultra low-power consumption [6]. The basic component of RSFQ digital circuits is a superconducting loop with Josephson junctions. A SFQ pulse, which appears as an voltage pulse, is used as the carrier of information. The width of an SFQ pulse is several pico-seconds and the height is about 1 mV. Energy consumption for an operation switching is smaller and the gate switching speed is higher than those of CMOS circuits, respectively.

While SFQ device has advantages of high-speed switching and low energy consumption, there are a few disadvantages for SFQ technologies. As aforementioned, SFQ pulse is used as the carrier of information, hence each RSFQ logic gate is clocked and has a function of latch. In other words, latches can be implemented without additional costs. Further, RSFQ digital circuits are suitable for pipeline processing on streaming data. On the other hand, it is difficult to implement feedback loops and conditional branches. Moreover, practical SFQ memory device has never been developed yet, and SFQ on-chip memory can not be implemented at the present stage.

To overcome these disadvantages, we have proposed a SFQ Reconfigurable Data-Path (SFQ-RDP) processor as an accelerator for a high-performance computing system [7], as shown in Fig. 1. SFQ-RDP is based on a two-dimensional ALU array, and has a data path architecture. The input data is flowed only in a one direction and computed without any feedback from the output to inputs which is matching to the requirements of implementation by SFQ circuits. SFQ-RDP prototype processor with 2x3 integer ALUs is already fabricated [8] as shown in Fig. 2. Also, half-precision floating point adders and multipliers have already been implemented [9], [10]. Current RSFQ process technology is $1 \sim 2\mu m$, which means that the integration density is lower than that of current CMOS. However, it is expected that larger scale SFQ-RDP with double precision FPU processors can be developed soon with finer SFQ process technologies.

Because of very simple architecture of SFQ-RDP, we have to prove that it is possible to perform efficient computations if the architecture has following specifications:

1) 2D ALU-array without any loop back inside the chip
2) each ALU can implement only ADD or MUL
3) there is no on-chip memory.

In [11], design scheme for determining the architecture specifications has been proposed. In [13], software and hardware mechanisms for efficient data transfer have been introduced.

In this research, our main focus is on developing two-dimensional Heat (2D-Heat) and Finite Difference Time Domain (2D-FDTD) applications and evaluating their performance on the target hardware already developed and reported in [11] and [13]. Both HPC applications are based on finite difference methods. It will be shown it is possible to perform efficient computations by SFQ-RDP computing system. Performance evaluation results are compared with ones for the same applications implemented on the GPU.

In the next section, an overview of the SFQ-RDP processor will be displayed. In Section 3, we will explain features of target applications and how they are implemented on the SFQ-RDP processor. In Section 4, results of performance evaluations will be presented, and Section 5 will conclude the paper.

II. General Architecture and Specifications of the SFQ-RDP

Total SFQ-RDP system is constituted with SFQ-RDP accelerator chip, GPP and main memory which are connected through a shared bus to each other [11] as shown in Fig. 1. Generally, SFQ-RDP processor is a pipelined architecture comprising a two-dimensional array of Processing Elements (PEs) including a FPU and data transfer units between succeeding rows. PEs of every two consecutive rows are connected with flexible operand routing networks (ORNs) such that one PE can be connected through ORNs to a number of PEs in the next row. There are Streaming Memory Access Controllers (SMACs) at SFQ-RDP I/O ports so that each PE can be fed through SMAC and via ORN switches. Feedback

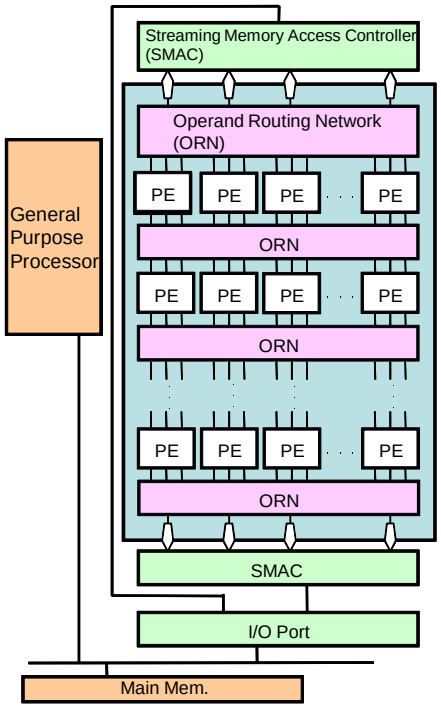

Fig. 1. A supercomputer architecture with SFQ-RDP processor

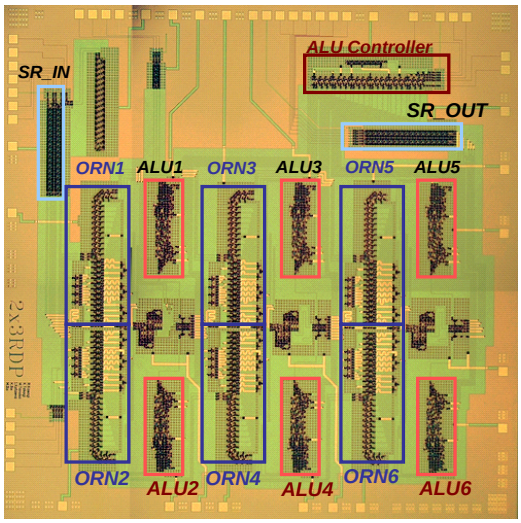

Fig. 2. A prototype 2x3 SFQ-RDP processor

data flow connections are not supported, which means that the flow of data in the PE arrays is unidirectional. The SFQ-RDP should be an adaptable accelerator, because it is aimed to target various scientific applications. In order to satisfy this requirement, the SFQ-RDP is featured with dynamically reconfiguring of the ORNs. Originally, an ORN consists of programmable switches. By means of setting the control signals provided with PEs and ORN switches, the function of the SFQ-RDP can be configured at run time. Such flexibility makes it possible to implement various Data Flow Graphs

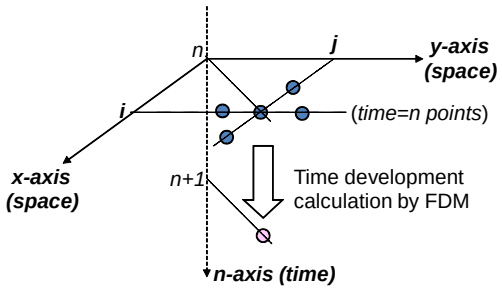

Fig. 3. 2D-Heat calculation

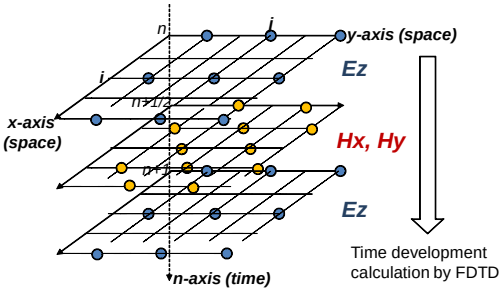

Fig. 4. 2D-FDTD calculation

(DFGs) on the PE array. A DFG extracted manually from a critical segments of target application program is mapped onto the two-dimensional PE array and executed to boost the performance. Since the cascaded PEs can generate a final result without temporally memorizing intermediate data, the number of memory load/store operations corresponding to spill codes can be reduced. Therefore, memory bandwidth required to achieve a high performance can be reduced as well. Furthermore, since a loop-body mapped onto the PE array is executed in a pipeline fashion, SFQ-RDP can provide a high computing throughput. For software implementation, SFQ-RDP architectural specifications including layout of ADD/MUL FPUs, ORN micro-architecture, RDP dimensions (width and height of 2D-PE array), numbers of Input/Output ports, configuration of each PE, and reconfiguration mechanism are obtained through a design process [11], [12].

Since SFQ-RDP processor implemented by superconductivity circuits has to be cooled to 4 Kelvine, special freezer is needed, thus energy consumption seems large for a small-scale computer. Therefore, SFQ-RDP accelerators can be used efficiently in the large computing system, where energy consumption of freezers are considered as negligible compared to total system.

III. Finite Difference Applications and Implementation on SFQ-RDP system

Nowadays, there are many applications which are the target of high-performance computing systems. We have already studied one-dimensional heat and vibrational applications based on Finite Difference Method (FDM) for proposing efficient SMAC mechanism by using SFQ-RDP system, and have obtained good performances [13]. Next, we will apply two-dimensional form of FDM based applications on SFQ-RDP system. This study can be easily extended to three-dimensional case.

A. 2D-Heat and 2D-FDTD applications

As shown in Fig. 3, Eq. (1) and (2) are the two-dimensional partial differential heat equation and corresponding discretized FDM equations, respectively.

$$\frac{df}{dt} = \kappa \left(\frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} \right) \tag{1}$$

$$\begin{aligned} f_{i,j}^{n+1} &= C_0 \times \left(f_{i-1,j}^{n} + f_{i+1,j}^{n} \right) \\ &\quad + C_1 \times \left(f_{i,j-1}^{n} + f_{i,j+1}^{n} \right) \\ &\quad + C_2 \times f_{i,j}^{n} \end{aligned} \tag{2}$$

Here, $f^n(i,j)$ is the heat function at position x_i, y_j and time t_n. $C_{0\sim2}$ correspond to thermal conductivity and assumed as constants. $f^{n+1}(i,j)$ is the heat function value at position x_i, y_j in the next time step, which is computed using current five point f^n values.

Eq. (3) and (4) are the 2D-FDTD equations shown in Fig. 4.

$$\begin{aligned} H_x^{n+1/2}\left(i, j+\frac{1}{2}\right) &= H_x^{n-1/2}\left(i, j+\frac{1}{2}\right) \\ &\quad + C_{HY}\left(E_z^n(i,j+1) - E_z^n(i,j)\right) \\ H_y^{n+1/2}\left(i+\frac{1}{2}, j\right) &= H_y^{n-1/2}\left(i+\frac{1}{2}, j\right) \\ &\quad + C_{HX}\left(E_z^n(i+1,j) - E_z^n(i,j)\right) \end{aligned} \tag{3}$$

$$\begin{aligned} &E_z^{n+1}(i,j) \\ &= E_z^n(i,j) \\ &\quad + C_{EY}\left(H_x^{n+1/2}\left(i, j+\frac{1}{2}\right) - H_x^{n+1/2}\left(i, j-\frac{1}{2}\right)\right) \\ &\quad + C_{EX}\left(H_y^{n+1/2}\left(i+\frac{1}{2}, j\right) - H_y^{n+1/2}\left(i-\frac{1}{2}, j\right)\right) \end{aligned} \tag{4}$$

To compute next E^{n+1} electric field function values, current E^n values and half-future $H^{n+1/2}$ magnetic field function values are required. These formulas are derived from electromagnetic wave equations [14]. Here, we assume C_{HY}, C_{HX} and C_{EY}, C_{EX} coefficients, which correspond to permeability and permittivity of media, are constant.

B. Implementations of applications on SFQ-RDP

To implement finite difference applications Eq. (2) and Eq. (3-4), we used loop-unrolling technique to generate larger DFGs as shown in Fig. 5 and Fig.6.

First for 2D-Heat case, "Original Code for GPP" corresponds to original implementation of 2D-Heat kernel code. Here, outside loop n and inner loops i, j correspond to time development and x, y space directions, respectively. This

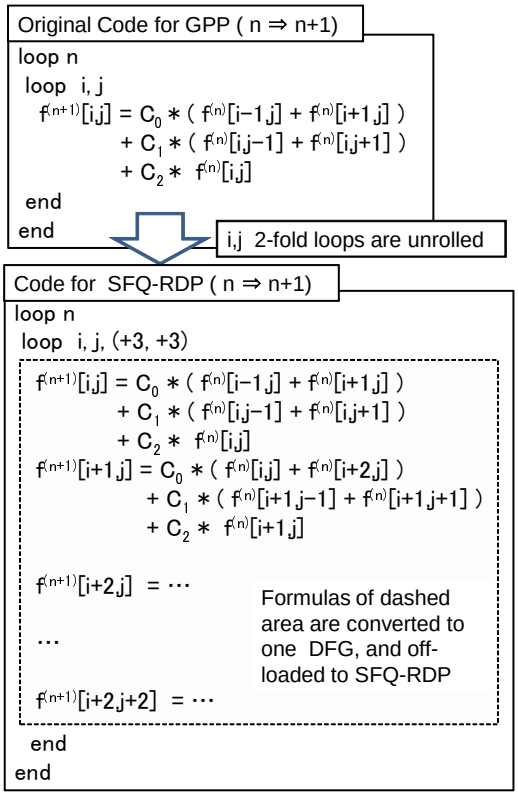

Fig. 5. Implementation of 2D-Heat to SFQ-RDP

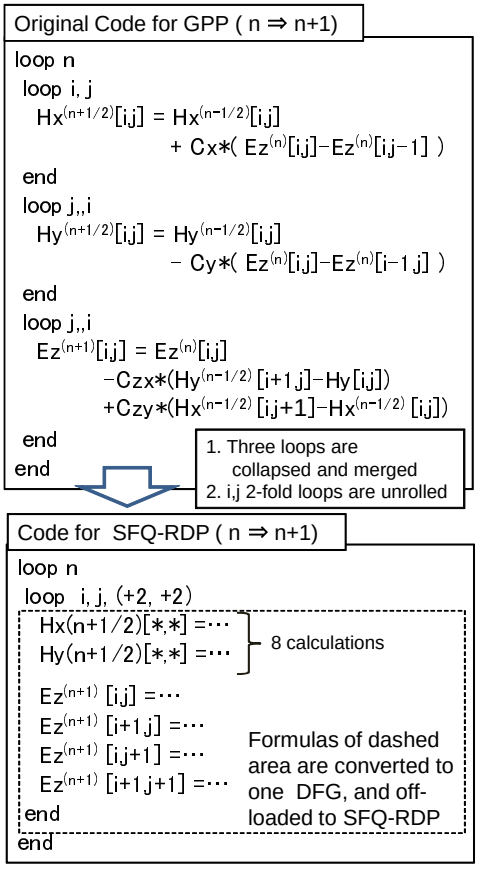

Fig. 6. Implementation of 2D-FDTD to SFQ-RDP

code is loop-unrolled three times for i, j indices for SFQ-RDP implementation. As a result, loop body part described in dashed area includes calculations with 63 operations, 21 inputs, and 9 outputs. This calculation part which is off-loaded to SFQ-RDP, is converted to DFG manually, and mapped onto SFQ-RDP's PE array by using a dedicated mapping tool and a configuration data set is generated for the run-time use [11]. Remaining loop controlling parts are computed in GPP.

Next for 2D-FDTD case, above implementation scheme is similar with 2D-Heat. For modifying original code, inner loops i, j are unrolled twice. As a result, loop body part described in dashed area, comprises calculations with 48 operations, 20 inputs, and 12 outputs. This DFG is also mapped onto the SFQ-RDP and a configuration data set is generated as well.

Obviously, the performances of both original implementations are depending on provided system memory bandwidth and the cache doesn't affect performance numbers significantly. Therefore, layout of main memory for describing f^n, H^n, and E^n arrays is modified in order to use DMA data transfer to a maximum extent. Two input and one output ports as well as double buffering scheme are used for input/output transactions between memory and SFQ-RDP.

IV. Performance Evaluations of Finite Difference Applications

A. Performance Evaluation Modeling

Execution times on SFQ-RDP is evaluated based on the below model. Total execution time T_{total} is obtained as Eq. (5).

$$T_{total} = T_{gpp} + T_{rdp} + T_{oh} \tag{5}$$

Here, total execution time is described as sum of GPP and SFQ-RDP execution times T_{gpp}, T_{rdp}, and overhead time T_{oh} for utilizing SFQ-RDP. SFQ-RDP execution time T_{rdp} is the sum of floating point calculation time T_{cal} and stall time for memory accesses T_{st}.

$$T_{rdp} = T_{cal} + T_{st} \tag{6}$$

As the SFQ-RDP has a pipelined structure, the execution time T_{cal} is estimated as follows:

$$T_{cal} = \sum_{i}^{n} \frac{C_i + H - 1}{f_{rdp}} \tag{7}$$

Here, SFQ-RDP is invoked n times and in each SFQ-RDP execution, number of input data lines of i^{th} execution is equal to C_i and number of rows of SFQ-RDP is equal to H. If there are no stalls for data transfer between memory and SFQ-RDP, input data will be available at every clock cycle. Since the

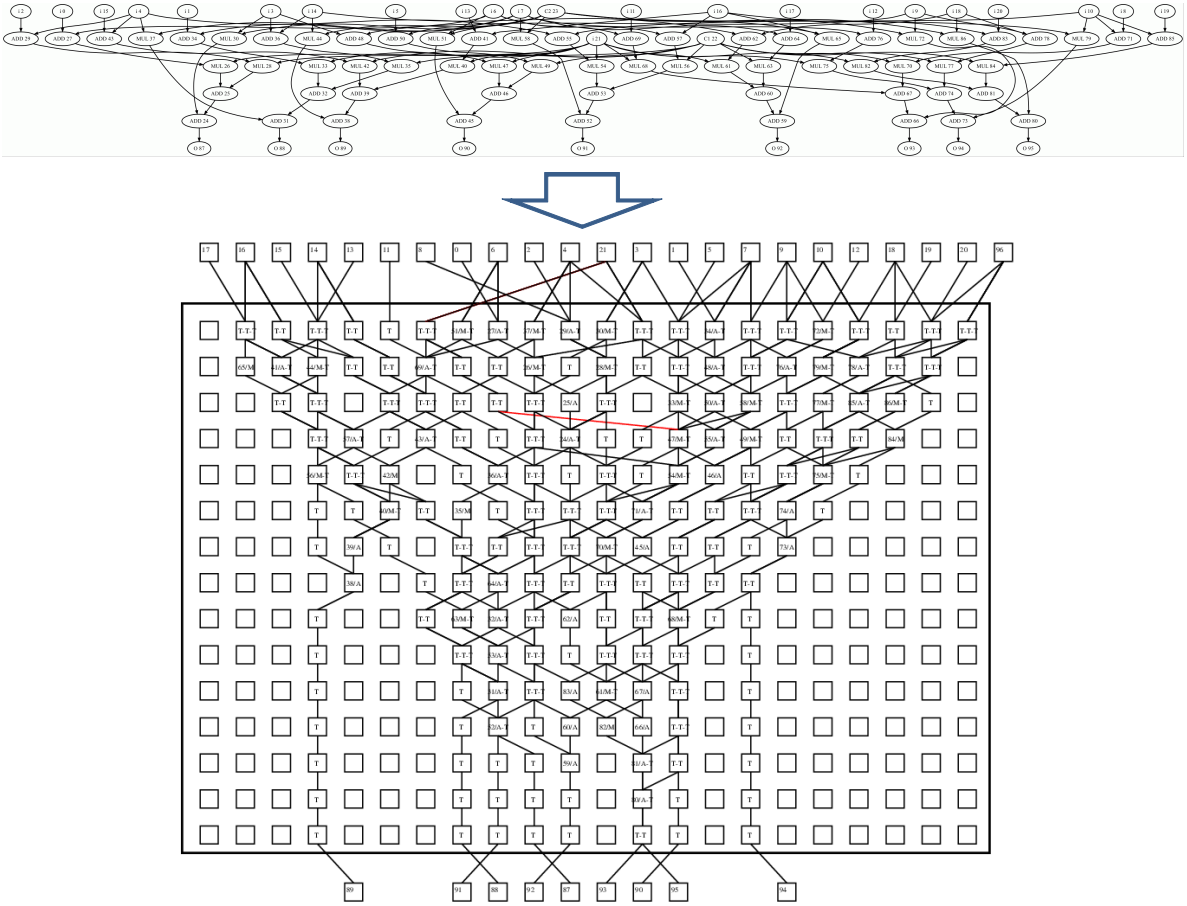

Fig. 7. Extracted 2D-Heat DFG and mapping onto SFQ-RDP

sum of C_i and H correspond to the number of clock cycles for performing calculations on SFQ-RDP, hence this value is divided by SFQ-RDP frequency f_{rdp} to calculate the execution time.

SFQ-RDP stall time T_{rdp} is achieved by sum of stall times spent for transferring data to every input and output rows as well as the first and last rows.

$$T_{st} = \sum_{i}^{n} \frac{2 \times Lat_{memrdp}}{f_{rdp}} + (\lceil \frac{BW_{rdpi}}{BW_{mem}} \rceil - 1) \times \frac{C_i}{f_{rdp}} \quad (8)$$

$$BW_{reqi} = (input_i + output_i) \times f_{rdp} \quad (9)$$

Here, Lat_{memrdp} corresponds to the memory latency from SFQ-RDP. If SFQ-RDP required bandwidth BW_{rdp} is larger than provided memory bandwidth BW_{mem}, stall time is proportional to BW_{rdp} and BW_{mem} ratio. Required bandwidth can be calculated based on the number of input and output ports ($input_i$ and $output_i$) corresponding to the number of required input and output data.

Overhead time T_{oh} is the sum of reconfiguration time T_{reci} and communication latency between GPP and SFQ-RDP T_{trai}.

$$T_{oh} = \sum_{i=1}^{n} \{T_{reci} + T_{trai}\} \quad (10)$$

Configuration parameters used for performance evaluation in next section are shown in Table I. For the sake of comparisons, assumptions on the amount of typical memory bandwidths which are supposed as 147.5GB/sec. and 158.0GB/sec. for 2D-Heat and 2D-FDTD, are obtained from reported implementations of these applications on counterpart GPU systems, respectively [15], [16].

B. Performance Evaluation Results

Execution times of 2D-Heat and 2D-FDTD applications with SFQ-RDP normalized with GPP execution times are shown in Fig. 8 with breakdown of each execution time. Execution times highly depend on provided system memory bandwidth. The larger memory bandwidth, the smaller stall time for application. For the maximum memory bandwidth cases, 2D-Heat and 2D-FDTD SFQ-RDP calculations are 26.2 and 79.0 times faster than GPP calculations, respectively. Largest execution time portions of both calculations are stall

TABLE I
CONFIGURATIONS OF GPP AND SFQ-RDP PROCESSOR

GPP	Processor type	Out-of-order
	Frequency	3.2GHz
	Inst. issue width	4 Inst./CC
	Inst. decode width	4 Inst./CC
	L1 data cache	64KB (128B Entry, 2Way, 2CC)
	L1 inst. cache	64KB (64B Entry, 1Way, 1CC)
	L2 unified cache	4MB (128B Entry, 4Way, 16CC)
	Latency of main mem.	300 CC
	L2 - main mem. bus width	64B
	L2 - main mem. freq.	64B
RDP	SFQ-RDP frequency	80 GHz
	Reconfiguration latency	30000 CC[(a)]
	Mem. Bandwidth	12.8, 102.4, 141.7, 157.0 GB/sec.
	No. of PEs in a row	22
	No. of PE rows	15

[a] Based on SFQ-RDP frequency (80GHz)

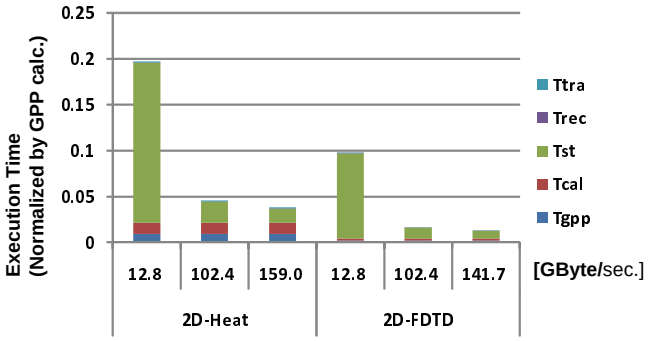

Fig. 8. Execution times of 2D-Heat and 2D-FDTD applications normalized by GPP executions

times correspond to data transfer between memory and SFQ-RDP. Because of 80GHz high frequency of SFQ-RDP, ratio of SFQ-RDP calculation times T_{cal} are small compared with T_{st}, and are about 30% and 80%, respectively. Both applications, T_{oh} for SFQ-RDP reconfiguration times are almost negligible because reconfigurations are needed only once at initialization of SFQ-RDP. 2D-Heat and 2D-FDTD performance numbers are 50.6 and 23.4 GFLOPS (single floating point calculations). While, reported GPU results are 63.0 and 31.4GFLOPS [15], [16]. Unfortunately, these results are smaller than GPU ones, however they are comparable to GPU results. As aforementioned, main loops of applications are unrolled to reduce the number of memory accesses, while the number of calculations doesn ’t change. Moreover, input/output data are rearranged in memory to use DMA transfer efficiently. Therefore, above remarkable performance numbers are achievable.

For one-dimensional Heat and Vibrational applications, we have obtained very high-performance values, namely 210.0 GFLOPS and 104.9 GFLOPS are achievable, respectively [13]. Current and previous results show that SFQ-RDP processor which is implemented by superconductivity circuits and has simple 2D-array architecture, can be used as an efficient accelerator for finite difference applications.

V. CONCLUSIONS

A high-performance computer comprising an accelerator referred as single-flux-quantum reconfigurable data-path (SFQ-RDP) with two-dimensional floating point array architecture implemented by superconducting circuits was introduced. To demonstrate effectiveness of SFQ-RDP system, two-dimensional Heat (2D-Heat) and Finite Difference Time Domain (2D-FDTD) applications are implemented on SFQ-RDP and performance evaluations are conducted. For 2D-Heat and 2D-FDTD, 50.6 and 79.0 times faster computation than general purpose processor are achievable, while these performance values are comparable to reported results for the GPU. Therefore, it is concluded that the SFQ-RDP accelerator can be used for practical scientific calculations especially based on finite difference methods.

ACKNOWLEDGMENT

The authors would like to thank Prof. A. Fujimaki, Prof. H. Akaike and Prof. M. Tanaka for kindly permitting us to use their processor picture and for their helpful discussion. The authors would also like to thank Prof. N. Takagi and Prof. K. Takagi of Kyoto University, Prof. N. Yoshikawa of Yokohama National University, and Dr. S. Nagasawa and Dr. M. Hidaka of International Superconductivity Technology Center for their helpful suggestions and discussion.

This research was supported in part by Core Research for Evolutional Science and Technology (CREST) of Japan Science and Technology Corporation (JST).

REFERENCES

[1] ClearSpeed processor. [Online]. Available: http://www.clearspeed.com/.
[2] J. Makino, K. Hiraki, and M. Inaba, “GRAPE-DR: 2-Pflops massively-parallel computer with 512-core, 512-Gflops processor chips for scientific computing,” in *SC07*. American Chemical Society, November 2007.
[3] Cell Broadband engine. [Online]. Available: http://cell.scei.co.jp/index.html.
[4] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A. Lefohn, and T.Purcell, “A survey of general-purpose computation on graphics hardware,” in *Computer Graphics Forum*, March 2007, pp. 80–113.
[5] B. Dally,“GPU Computing: To Exascale and Beyond,” International Conference for High Performance Computing 2010 (SC10), Nov.2010.
[6] K. Likharev and V. Semenov, “RSFQ logic/memory family: a new Josephson junction technology for sub-teraherz clock frequency digital systems,” *IEEE Trans. Appl. Supercond.*, vol. 1, no. 1, pp. 3–28, 1991.
[7] N. Takagi, K. Murakami, A. Fujimaki, N. Yoshikawa, K. Inoue, and H. Honda, “Proposal of a desk-side supercomputer with reconfigurable data-paths using rapid single flux quantum circuits,” *IEICE Trans. on Elec.*, vol. E91-C(3), pp. 350–355, 2008.
[8] A. Fujimaki, S. Iwasaki, K. Takagi, R. Kasagi, I. Kataeva, H. Akaike, M. Tanaka, N. Takagi, N. Yoshikawa, K. Murakami, “Demonstration of an SFQ-Based Accelerator Prototype for a High-Performance Computer,” Applied Superconductivity Conference 2008 (ASC08), Aug.2008.
[9] T. Kainuma, Y. Shimamura, F. Miyaoka, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, K. Takagi, N. Takagi, S. Nagasawa, “Design and Implementation of Component Circuit of an SFQ Half-Precision Floating-Point Adder Using 10-kA/cm2 Nb Process,” IEEE Trans. Appl. Supercond., Vol. 21, no. 3, pp. 827-830, Jun. 2011.
[10] Y. Shimamura, Y. Yamanashi, N. Yoshikawa, A. Fujimaki, N. Takagi, K. Takagi, “Design and implementation of SFQ Floating-Point Multiplier and Adder Using 10 kA/cm2 Nb Process,” Superconductivity Centennial Conference, Den Haag, Netherlands, Sep. 2011.
[11] F. Mehdipour, H. Honda, K. Inoue, H. Kataoka, and K. Murakami, “A design scheme for a reconfigurable accelerator implemented by single-flux quantum circuits,” Journal of Systems Architecture - Embedded Systems Design 57(1), pp.169-179, Jan. 2011.

[12] I. Kataeva, H. Akaike, A. Fujimaki, N. Yoshikawa, N. Takagi, and K. Murakami, “An operand routing network for an SFQ reconfigurable data-path processor,” *IEEE Trans. Appl. Supercond.*, vol. 19, no. 3, pp. 665–669, 2009.
[13] H. Kataoka, H. Honda, F. Mehdipour, K. Inoue and K. Murakami, “Reducing Preprocessing Overhead Times in a Reconfigurable Accelerator of Finite Difference Applications,” 2010 Symposium on Application Accelerators in High Performance Computing (SAAHPC’10), Jul. 2010.
[14] K. Yee, “Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media,” IEEE Trans. Antennas Propagat., 14, 4, pp.302-307, 1966.
[15] T. Aoki, and A. Nukada, “CUDA programming primer,”, (Japanese), Kougakusya, ISBN-10:4777514773, 2009.
[16] N. Takada, T. Shimobaba, N. Masuda, and T. Ito, “Speeding up of FDTD finite difference calculations by efficient use of GPU and shared memory,”, (Japanese), Proceedings of Forum of Information Science and Technology, 2009.