An Accelerator Based on Single-Flex Quantum Circuits for a High-Performance Reconfigurable Computer

Mehdipour, Farhad
Graduate School of Information Science and Electrical Engineering, Department of Informatics,
Kyushu University

Honda, Hiroaki
Institute of Systems, Information Technologies and Nanotechnologies

Kataoka, Hiroshi
Graduate School of Information Science and Electrical Engineering, Department of Informatics,
Kyushu University

Inoue, Koji
Graduate School of Information Science and Electrical Engineering, Department of Informatics,
Kyushu University

他

An Accelerator Based on Single-Flex Quantum Circuits for a High-Performance Reconfigurable Computer

Farhad Mehdipour[†], Hiroaki Honda[††], Hiroshi Kataoka[†], Koji Inoue[†] and Kazuaki Murakami[†]

[†]Graduate School of Information Science and Electrical Engineering, Department of Informatics, Kyushu University, Japan
[††]Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka, Japan
E-mail: {farhad, kataoka}@c.csce.kyushu-u.ac.jp, dahon@isit.or.jp, {inoue,murakami}@i.kyushu-u.ac.jp

Abstract

A large-scale reconfigurable data-path (LSRDP) processor based on single-flux quantum circuits has been proposed to overcome the barriers originating from the CMOS technology. LSRDP is integrated to a general purpose processor in a high-performance computing system to accelerate the execution of data flow graphs extracted from scientific applications. The LSRDP micro-architecture design procedure using a quantitative approach over benchmark data flow graphs and its specifications will be presented. A preliminary performance evaluation of the reconfigurable processor will be given in this paper as well.

1. Introduction

Nowadays, in various scientific areas such as quantum 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. 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.

Computer systems based on parallel computer clusters with general-purpose processors (GPPs) are often utilized for the high performance computing. Those parallel computers with GPPs account for a large share of the performance ranking in TOP500 [13]. On the other hand, a hybrid architecture comprising an accelerator augmented to a GPP might be chosen for special purpose computations. The accelerator should be designed to feature small size, high performance, and low power consumption. Recent examples of such accelerators are CSX600 PCI-X board [2], GRAPE-DR processor [1] and Cell processor (a heterogeneous multi-core processor) [6]. These accelerators commonly have single-instruction multiple-data stream (SIMD) mechanism for total architecture or functional units.

Generally as the most of computing systems are implemented by CMOS technology, there are some barriers in realizing powerful computing systems using this technology. The most important issues are high heat radiation, long interconnection delays and memory-wall problem [14].

As a solution, a desk-side tera-flop scale computer is introduced [12] which consists of a CMOS general purpose processor, a memory and a single-flux quantum (SFQ)-based Reconfigurable Large-Scale Data-Path processor (SFQ-LSRDP) as an accelerator (Fig. 1) [12]. Generally, a large memory bandwidth is demanded in conventional accelerators to perform calculations efficiently. Therefore, an on-chip memory is utilized for reduction of the required memory bandwidth. The proposed architecture is expected to be a 10TFLOPS desk-side computer with low electric power consumption and it is suitable for execution of the scientific applications demanding massive computations.

A SFQ circuit is based on the superconductor technology which includes low power consumption and high-speed compared to the CMOS circuits [12]. A SFQ logical circuit uses a 1mV extremely low-width pulse as an information carrier that is propagated at very high speed (up to light speed) in the circuit. Therefore, the SFQ circuit has a smaller switching energy and high switching speed in comparison with CMOS circuit. In addition, because the SFQ pulse propagates in the speed of light, its transmission speed is not limited to the latency time of electrical charge and discharge of CMOS gate capacitances. The main features of the SFQ technology can be summarized as follows:

- high-speed switching and signal transmission
- low power consumption
- compact implementation of a system (small area)
- no cost for latch
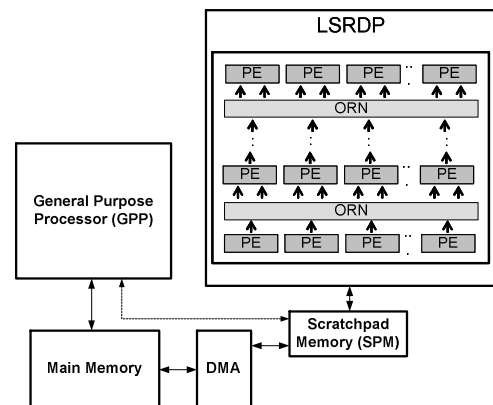- suitable for pipeline processing of data stream

Fig. 1. Overall architecture of the SFQ-LSRDP computer

1

One main component of the target architecture is a large-scale reconfigurable data-path (LSRDP) based on SFQ to address the following issues which usually come from CMOS technology:

- high electric power consumption
- high heat radiation and difficulties in high-density packing
- memory wall problem which limits the processing speed

LSRDP utilizes a data path comprising reconfigurable interconnections to connect several floating point processing units together. According to Fig. 1, the LSRDP is augmented to GPP as an accelerator. Executing the most frequently executed portions of applications (represented as data flow graphs) or in other word the part of applications demanding massive and time-consuming computations is the main responsibility of the LSRDP. In a general view, critical segments of applications are pulled-out and their corresponding configuration bit-streams are generated. During execution of application on the base processor, configurations relating to critical segments are loaded on LSRDP and executed to achieve higher performance and lower power consumption.

Developing necessary tools for compiling applications, generating data flow graphs (DFGs) and their configuration bit-streams as well as designing the LSRDP architecture are the main phases of implementation of target high-performance computer which will be discussed in the following sections. The main focus of this paper is on the design procedure for the LSRDP and presenting results of a preliminary evaluation of the designed architecture.

2. LSRDP General Architecture and Specifications

Fig. 1 displays the overall architecture of a high performance computer consisting of a GPP, LSRDP as the accelerator and memory elements. Generally, LSRDP is a pipelined architecture comprising a two-dimensional array of processing elements (PEs) such that one PE can be connected through operand routing networks (ORNs) to one or more PEs in the next row.

SFQ technology provides the LSRDP with a straightforward pipelined structure implementation. Each PE can be fed through input ports and the resultant of each PE can be transferred to one or more PEs in the next row via ORN switches. Reverse data flow connections are not supported, which means that the flow of data in the array is only in one direction. The LSRDP should be an adaptable accelerator, since it is aimed to target various scientific applications. In order to satisfy this requirement, the architecture is featured with dynamically reconfigurable PEs and ORNs. Originally, an ORN consists of programmable switches. Through configuring control signals provided with PEs and ORN switches, the function of LSRDP can be determined at run time. Such flexibility makes it possible to implement various DFGs on the array.

A data flow graph (DFG) extracted from a target application program is mapped onto the LSRDP array. 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 might decrease as well. Furthermore, since a loop-body mapped into the PE array is executed in a pipeline fashion, LSRDP can provide a high computing throughput.

Some assumptions and definitions on LSRDP architecture are presented here.

PE types: Each PE includes an FU for implementing desired operation and a TU (transfer unit) as a routing resource for transferring data to the next row. In LSRDP architecture, ORNs provide routing resources between succeeding rows. It means, to connect two PEs locating on inconsequent rows one or more transfer units should be utilized. Since a unique implementation of PEs is preferred in SFQ technology, it is supposed that each PE has a general architecture including a functional unit (FU) and a Transfer unit (TU) and it is possible to use an FU for implementing a transfer unit as well. In addition, each PE has three inputs (two inputs for FU and one for the TU) and two outputs (one from FU and another from TU).

Type and granularity of functional units: It is assumed that FUs can implement basic 64-bit double-precision floating point operations like e.g. ADD, SUB and MUL. Control instructions (branches) and direct memory accesses via PEs are not supported.

Layout: Layout of the LSRDP represents the type of FUs and their distribution. Three types of layouts are supposed for the LSRDP (Fig. 2). In a normal layout (type I), each FU can

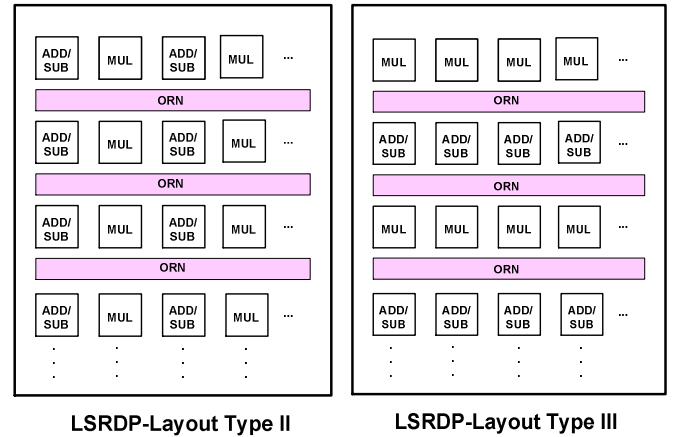**LSRDP-Layout Type II** **LSRDP-Layout Type III**
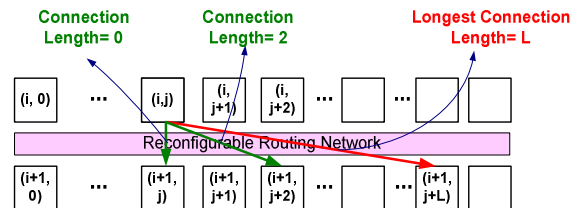
Fig. 2. LSRDP layout types

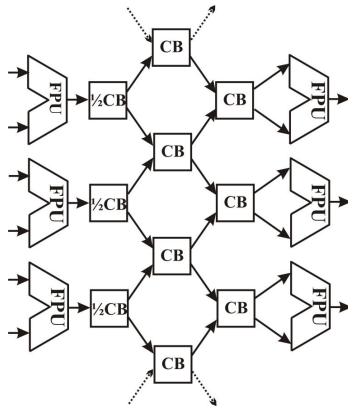Fig. 3. Definition of the connection length and the maximum connection length (MCL) on a piece of LSRDP architecture

Fig. 4.   Block diagram of a crossbar-based ORN

Fig. 5.   State diagram for LSRDP operation

implement any operation. In layout type II, each PE can implement only ADD/SUB or MUL. As in Fig. 2 (right-side, type III), only one type of operations is implemented in each row. Obviously, the implementation cost of a PE and entire LSRDP architecture based on layout types II and III would be less than that of type I.

Internal memory: 64-bit immediate registers are located in each PE in order to handle immediate values. The immediate values are transferred to the registers within configuration phase through a serial bit-stream.

LSRDP structures: Due to variety of DFGs generated from scientific applications, four architectures (small, medium, large and x-large) including different number of resources are designated and the input DFGs are classified into four groups, correspondingly.

Input/Output ports: Different numbers of input/outputs ports is assigned for various LSRDP architectures. The limitation on the number of ports depends on the available memory bandwidth, LSRDP operation frequency, width of data bus and the number of memory read/write channels.

LSRDP dimensions: Fig. 1 shows that LSRDP is a matrix of PEs in which the height and width of LSRDP are the number of rows and columns, respectively. The LSRDP height and width are two important parameters which are determined during the design procedure and directly affect the number of resources and area of LSRDP.

Operand routing network (ORN): PEs of each row are connected to the PEs in the next row through ORNs as routing resources. The maximum connection length (MCL) is defined as the maximum horizontal distance of two PEs located in two succeeding rows (Fig. 3). ORN size is determined base on the MCL value. The number of ORNs and their size affect the LSRDP area, energy consumption and its implementation cost as well.

ORNs' functionality is similar to a multiplexer however; ORNs are implemented as cross-bar switches. Here an implementation of an ORN is presented [5]. The requirements of an ORN are as follows:

  o each PE can be connected to one or more PEs in the next row;
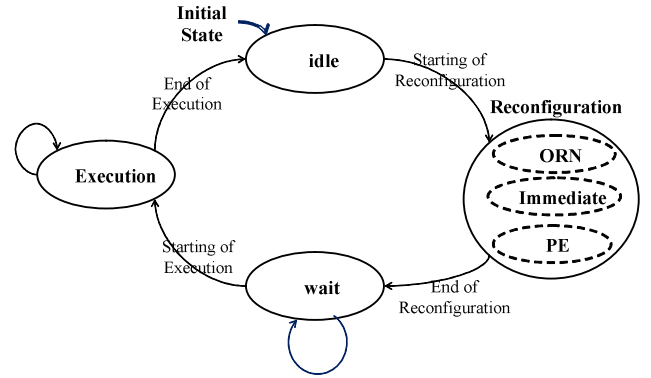 o connections exist among PEs located in the immediate

vicinity of each other and the maximum number of the connections, *N*, is odd
  o an FU output can be connected to either or both inputs of a PE in the next row.

To implement an ORN, crossbar switches are used as shown in Fig. 4. Due to the checker-type arrangement of the crossbars, this type of architecture is naturally suitable for the ORN with an odd number of connections per FU. In order to support a multicasting network, the crossbar switches must be capable of performing two more functions in addition to 'cross' and 'bar': multicasting of either of the inputs. ½ CB is a crossbar switch with only one input, from which data can be sent to either or both outputs. The crossbar-based ORN has a regular pipelined structure that does not limit the performance of the LSRDP and can be reconfigured on the fly. It can also be easily redesigned for any given complexity by adding a necessary number of extra rows of crossbars.

Reconfiguration mechanism: LSRDP is a reconfigurable hardware that can be configured within run-time using the bit-stream generated for DFGs. Fig. 5 shows the state diagram that represents the functionality of the LSRDP and how it is programmed and used in different stages. Upon reaching to a critical segment during application execution, a reconfiguration phase starts and the LSRDP configurable architecture including ORNs, immediate registers and PEs are reconfigured. Then, a DFG corresponding to the critical part of application might be executed iteratively. To eliminate or to reduce the reconfiguration overhead time a pre-configuration can be performed, therefore after finishing the configuration stage and before the LSRDP operation a wait state is required. In this manner, the reconfiguration phase would be overlapped with the GPP execution.

Fig. 6 shows the architecture of a PE and how it can be reconfigured during the configuration phase. Apart initializing immediate registers, the multiplexers, PEs and ORN micro-routing network should also be programmed using the configuration bits. According to Fig. 7 a serial chain is used for configuring immediate registers, PEs and ORNs. In order to configure each component, the configuration bit-stream is serially transferred to the configuration registers. It might take a hundreds of cycles with respect to the configuration bit-stream size which directly depends on the LSRDP specifications.
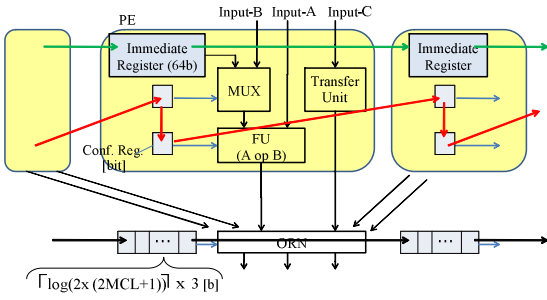
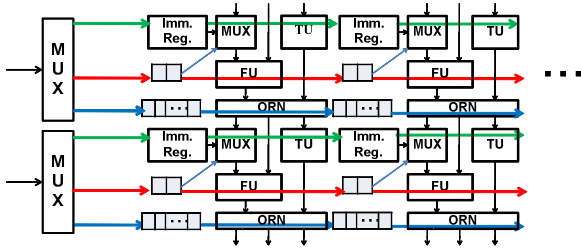Fig. 6.   Detailed architecture of a reconfigurable PE
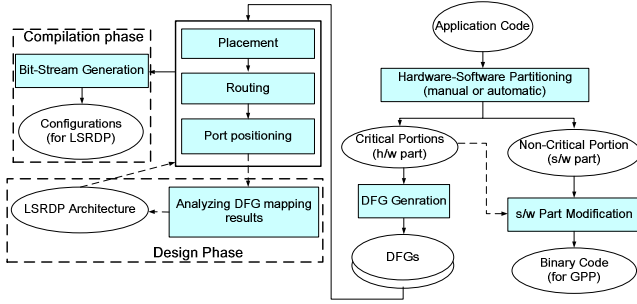

Fig. 7.   Reconfiguration structure of the LSRDP

Fig. 8.   A detailed outline of the proposed hw/sw compilation flow

External Memory: It is assumed that sixteen memory modules of 1800Mbps/pin are used in the memory array [7]. Each memory module uses one channel for input and the other channel for output. Data bus width for transferring data is 64bit and double precision data (8 bytes) is handled in the computations. Therefore, the data transfer rate is almost 24GB/s.

3.   LSRDP Design Procedure

3-1. Tool Chain

Fig. 8 shows the proposed tool chain which is used in the design and compilation phases. In the first stage, a hw/sw partitioning is performed on the input application manually. Critical segments of the code are isolated and the corresponding DFGs are generated. Considering the LSRDP architectural specifications, DFGs are mapped onto the LSRDP through placing DFG nodes on the PEs, routing interconnection as well as positioning input/output nodes on the proper ports. Placement and routing procedures should be iterated until a valid map satisfying the LSRDP constraints is

generated. Configurations' bit-stream corresponding to each one of DFGs can be generated after completion of the mapping stage. An executable code including non-critical segments of the application code and a piece of code for LSRDP interfacing ought to be produced. A part of compiler tools can be customized for utilizing in the LSRDP design phase as shown in Fig. 8.

3-2. DFG Extraction

Extracting critical portions of applications can be done manually or automatically by means of a sophisticated high-level profiling tool. In the former case, programmer needs to have a sufficient knowledge on the application and its detailed characteristics. On the other hand, automation of the hw/sw co-design methodology [11] brings with it the need to develop sophisticated high-level profiling tools e.g. *gprof* [3], HALT[15], ProfileME [4].

Four applications are attempted as benchmark scientific applications including: one-dimensional heat (referred as Heat) and vibration equations (Vibration), two-dimensional Poisson equation (Poisson) [9], and recursion calculation part of electron repulsion integral (ERI [8]) as a quantum chemistry application. All calculations consist of ADD, SUB, and MUL operations.

It is inefficient to use only small DFGs for the acceleration, therefore, larger DFGs are generated through combining the smaller ones together. For example, in heat equation, the extracted basic DFG can be shown as Fig. 9. By expanding that equation over the space and time dimensions, the final computation structure will correspond to DFG in Fig. 10. A similar DFG generation procedure is applicable to the basic DFGs of vibration and Poisson equations. Table 1 denotes the number of DFGs produced for each application.

3-3. Design Stages

Different methods can be used for determining LSRDP's detailed architectural specifications. One approach is to utilizes quantitative analysis of DFGs and extracting their properties. Various characteristics of DFGs are utilized during the design procedure. Other approaches like analytical or the combination of analytical and quantitative approach as well as design space exploration are applicable as well.

Fig. 8 shows a flow of design stages. As the design flow is an iterative procedure of gathering statistics and analysis of results, therefore, the designer should decide the priority of design parameters in the first step. Then, for determining each design parameter, DFGs should be mapped onto the LSRDP and outcome should be analyzed. There is no limitation in the initial architecture and the mapping process is performed without forcing any constraint. In the next stage, the results of mapping should be analyzed by the designer to decide an appropriate value for the intended parameter. This interactive process is repeated until fixing entire specifications of the architecture (the upward dashed line denotes that the process is iterative).
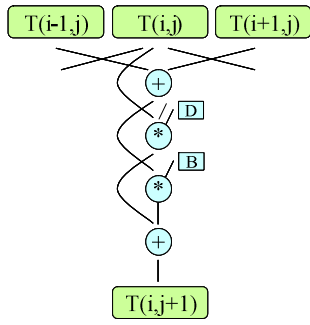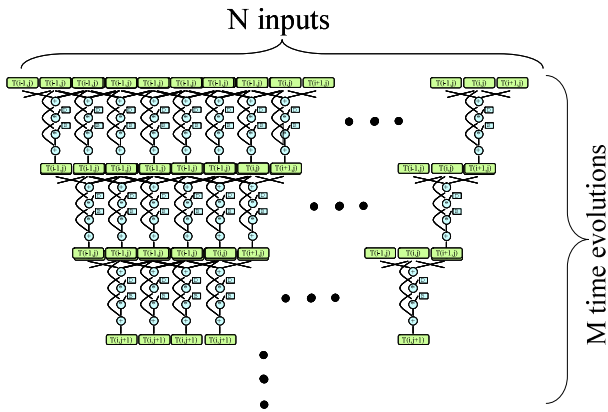
4

Fig. 9.   Data flow graph of basic heat equation

Fig. 10.   Combined data flow graph of Heat equation

Table 1.   Number of DFGs extracted from each application

Application	Heat	Vibration	Poisson	ERI
# of DFGs	6	7	3	8

3-4. DFG Classification

DFGs generated from various applications have different qualities in terms of size, number of inputs/outputs, connection complexity and etc. For each application at least one DFG is generated. Moreover, for some DFGs different versions are introduced to be able to meet different architectural specifications. Among various implementations of a DFG, usually a larger DFG is preferred due to higher achievable speedup.

Three factors including the total number of available PEs in the LSRDP and the number of input/output ports are considered as the main criteria for classifying DFGs. Four classes comprising Small (S), Medium (M), Large (L) and XLarge (XL) are identified. Table 2 shows the values determined for above parameters and the list of DFGs in each class. For the sake of coverage at least one DFG from each application is included in the classes. For an application which does not have any DFG in the specified range of parameters, a DFG from smaller class is used. The number of input/output ports in LSRDP architectures is determined based on the available communication bandwidth between the external memory array and the LSRDP [7], number of I/O channels in memories, data-bus width as well as LSRDP operating frequency.
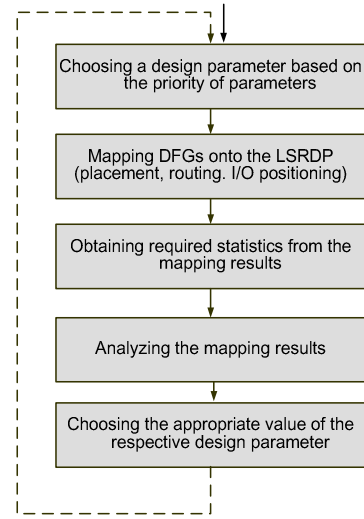

Fig. 11.   LSRDP Design flow (quantitative approach)

Table 2.   Various LSRDP configurations and a list of DFGs classified for each of LSRDP configurations

	# of PEs	# of Inputs	# of Outputs	# of DFGs
LSRDP-S	128	19	12	Heat (3) Poisson (1) Vibration (2) ERI (4)
LSRDP-M	512	19	12	Heat (1) Poisson (1) Vibration (1) ERI (4)
LSRDP-L	1024	38	24	Heat (2) Poisson (1) Vibration (2) ERI (5)
LSRDP-XL	> 1024	64	52	Heat (1) Poisson (1) Vibration (2) ERI (5)

3-5. Placement and Routing

During mapping process, firstly, DFG nodes are placed on appropriate positions (PEs) in the LSRDP. This is similar to the well-known placement problem [10]. Generally, minimizing the total connection length or the maximum connection length are main objectives, however in designing LSRDP, the mail goal is to minimize maximum connection length that directly impacts the ORN sizes.

Routing process is the next stage that establishes connections between the PEs in the LSRDP by means of ORNs and transfer units [10]. As aforementioned, it is supposed that each PE includes a transfer unit for transferring data as well as an FU for implementing operations. For each connection it is aimed to find a shortest path between the source and destination PEs. ORNs provide connection resources between two succeeding rows, therefore the connection length between two PEs in two succeeding rows should be less than the available connection lenghts provided by the ORNs. Fig. 12 shows how a sample DFG extracted from the vibration equations is mapped onto LSRDP.

3-6. I/O Positioning

The last step of mapping process is to locating input/ouput nodes of DFG on the appropriate ports around the LSRDP (Fig. 13). It is assumed that input and ports are located in the top and bottom borders, respectively. Between the first/last LSRDP rows and input/output ports, ORNs are onsidered as routing resources. In this step, the main objective is to reducing the connection legnth between ports and PEs.

3-7. Connection-length minimization

Connection length is calculated based on the distance of source and destination PEs of a net in horizontal direction (Fig. 3). Maximum ORN size is decided based on MCL value among whole DFGs. Since the number of ORNs strongly impacts the LSRDP various costs, reducing ORN size is an important challenge in the design procedure. Assuming *MCL* as the maximum connection length, the number of inputs required for an ORN supporting the longest connection is *2 x (2 x MCL+1)*. It is doubled because each PE has two outputs, one from FU and another from TU. Moreover, each ORN has three outputs to transfer the output data of each PE to the three inputs of some PEs in the next row. Fig. 14 sketches the structure of a 10 to 3 ORN assuming MCL equal to 2. In attempt to reduce ORN size and maximum connection length an exhaustive-search routing algorithm was developed which obtains the minimum MCL from any source to destination.

4. Experimental Results

The design procedure for LSRDP was accomplished using the DFGs extracted from scientific applications (Table 2) and considering various LSRDP architectures. Primary specifications of each LSRDP architecture i.e. the number of PEs and the number of input/output ports has been indicated in Table 3 as well. Other specifications of the LSRDP including its height (the number of rows) and width (the number of PEs in each row), the maximum connection length (MCL) and etc. were determined during the design process.

Table 3 shows the result obtained through the design procedure as well. Maximum and average values of width and height have been reported in the second and third columns. Fourth column denotes the total number of PEs required for implementing applications. The last column is the maximum and average numbers of extra TUs required for implementing DFGs.

To analyze the connection length and decide an appropriate size for ORNs, distribution of average connection lengths for LSRDP architectures were obtained. The graph in Fig. 15 shows that majority of connections have a small length and only a small fraction of them have a longer length more than the average connection length. Moreover, Table 4 shows the maximum connection length for various LSRDP architectures. With respect to the MCL values, the specifications of ORNs for various LSRDP architectures are calculated as in Table 4.
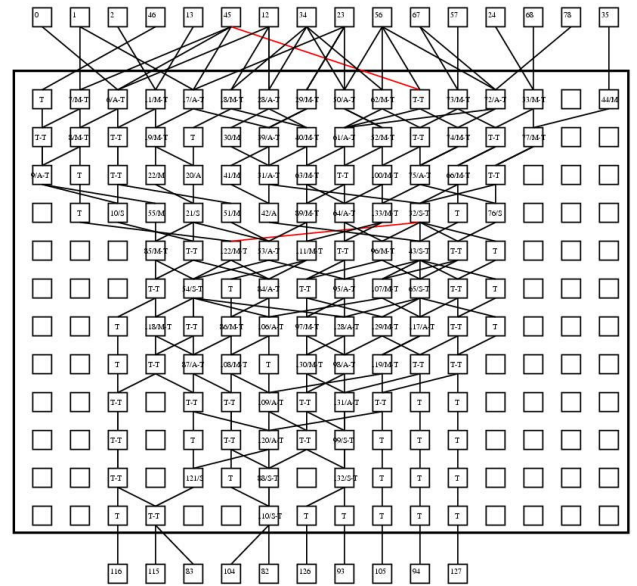

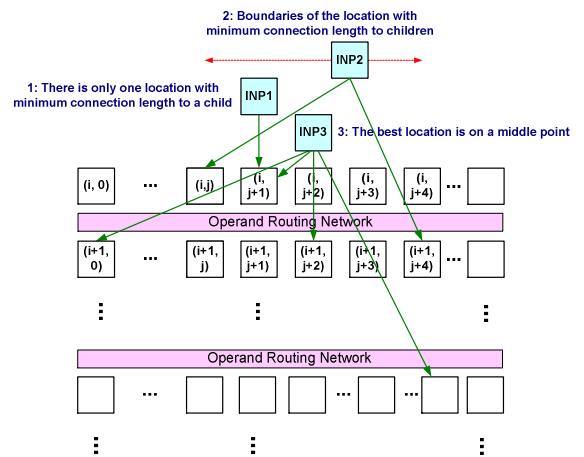Fig. 12.   Mapping process of a sample data flow graph onto LSRDP
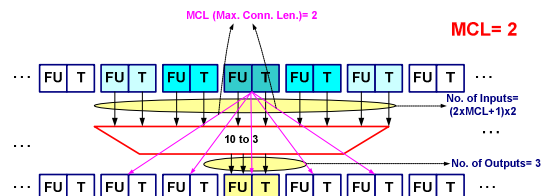

Fig. 13.   Input ports positioning

Fig. 14.   A 10 to 3 ORN with the maximum connection length of 2

In order to obtain an optimized structure including PEs with required functionality, three different layouts introduced in Section 2 were examined. We conducted our experiments considering three various layouts as underlying LSRDP architectures and analyzed the results of mappings. It is observed that for majority of input DFGs, the second layout type has comparable results to those of first layout. In fact, the result of mapping depends upon two important factors. The first one is the characteristic of the DFG (pattern of

6

operations) and the second one is the LSRDP layout. More similar operation patterns in the DFG and layout, better mapping result (minimized area as well as maximum connection length) is expectable.

To evaluate performance of the proposed LSRDP-based computer, a preliminary evaluation was accomplished based on simple analytical analysis and simulation. A base processor with characteristics displayed in Table 5 was used. Also, 0Table 6 shows configuration of target reconfigurable processor comprising GPP and LSRDP.

Two applications including Heat and Poisson equations were examined. Fig. 16 and Fig. 17 show performance on the GPP+LSRDP for those applications, respectively. Vertical axis denotes the normalized execution time (ratio of execution time on GPP+LSRDP to the execution time on GPP). In the figures, a breakdown of the execution time can be seen. For Poisson, in the 'Basic' bar, the largest portion of execution time (referred as 'Rearrange') is the time required for data rearrangement. Input/output data should be arranged in a proper order for the next execution step on the LSRDP. The second major fraction ('Stall') is the time elapsed for transferring data from scratchpad memory to main memory and vice versa. Fig. 16 depicts that the total execution time mainly consists of the GPP time ('GPP'), the LSRDP calculation time ('LSRDP') and communication time between the LSRDP and scratchpad memory ('Comm.').

To enhance the overall performance, a data reusing technique is employed to avoid the need for data rearrangement as well as frequently reloading data from the scratchpad memory. By using this technique the achievable speedup is improved considerably (Fig. 16). As a matter of fact, in the Poisson, by increasing DFG size, lower speedup is achieved. That is because of particular property of this application. A large or a small DFG of Poisson can be chosen to be mapped on the LSRDP; however by using smaller DFG, it is required to execute fewer instructions on the GPP rather than LSRDP.

Fig. 17 depicts the performance evaluation for the Heat application while data reusing is exploited. By using a larger DFG from the Heat application, total execution time decreases, hence performance rises. Total execution time mainly includes the GPP time, the LSRDP computation time and the communication time between LSRDP and scratchpad memory. According to the graphs, only a small fraction of the overall execution time is related to processing time on LSRDP and the main fraction concerns to various overhead times and execution time on GPP. Consequently, reducing above overhead times will strongly improve the achievable speedup.

5. Conclusion

A high-performance computer comprising an accelerator implemented by superconducting circuits was introduced. This computer is suitable for executing massive computational-intensive scientific applications. A quantitative design procedure was followed and a number of data flow graphs extracted from scientific application were subjected to the design procedure. Due to broad range of

Table 3. Results of the LSRDP design procedure

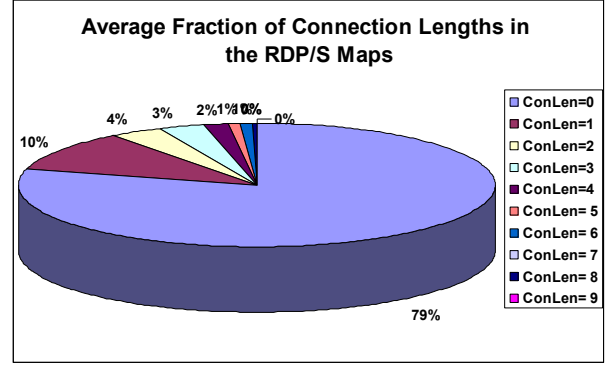|	Width (max/avg)	Height (max/avg)	Total # of PEs (max/avg)	Extra TUs (max/avg)
LSRDP-S	26/14.9	10/6.7	98/51.7	56/23.5
LSRDP-M	26/17.14	16/9.29	170/77.57	92/37.14
LSRDP-L	58/40	24/14.4	730/260.1	428/141.3
LSRDP-XL	122/45.25	25/12.38	1217/350.38	1065/240

Fig. 15. Fraction of connection lengths

Table 4. Specifications of ORNs

LSRDP	MCL (avg/max)	ORN Size # of Inps (avg/max), Outs	# of ORNs in Each RDP Row
LSRDP-S	4/9	18/38, 3	26
LSRDP-M	5/9	22/38, 3	26
LSRDP-L	9/19	38/78, 3	58

Table 5. Configuration of the base processor

Processor type	Out-of-order	
GPP operating frequency	3.2GHz	
Inst. issue width	4 instruction/cc	
Inst. decode width	4 instruction/cc	
Cache configuration	L1 data	64KB(128B Entry, 2way, 2cc)
	L1 instruction	64KB(64B Entry, 1way, 1cc)
	L2 unified	4MB(128B Entry, 4way, 16cc)
Latency of main memory	300cc	
L2 to main memory	Bus width	64 Bytes
	Freq	800 MHz

Table 6. Configuration of the reconfigurable processor (GPP+LSRDP)

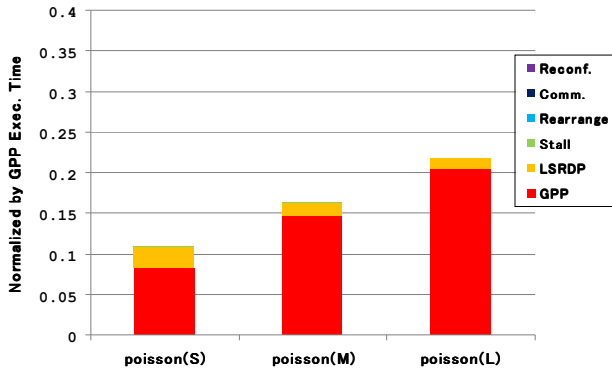| LSRDP operating frequency	80 GHz
Reconfiguration Latency	1cc
Latency SPM ←→LSRDP latency	1cc
Latency Main Memory ←→SPM	7500cc
Bandwidth SPM←→LSRDP	Max. 64 * 8 Bytes/cc
Bandwidth Main Memory←→ SPM	102.4GB/sec

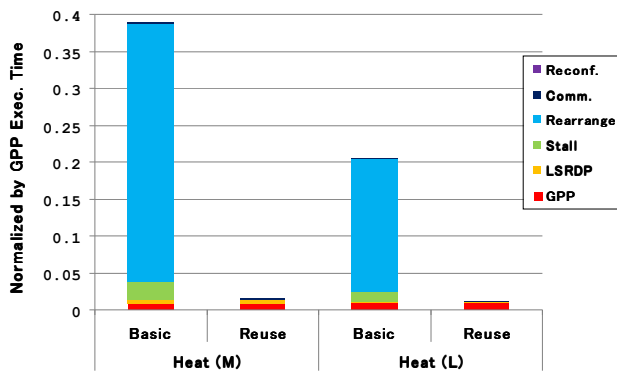Fig. 16.   Results of performance evaluation for Poisson application

Fig. 17.   Results of performance evaluation for Heat application

DFGs' characteristics, LSRDP architectures with different dimensionalities have been designed and evaluated. Evidences from experiments demonstrate that the high-performance computer equipped with SFQ-LSRDP is promising for resolving issues originated from CMOS technology as well as achieving considerable performances.

Acknowledgement

References

[1] Cell Broadband Engine, http://cell.scei.co.jp/index_j.html.

[2] ClearSpeed Processor, http://www.clearspeed.com/.

[3] R. F. Cmelik, SpixTools Introduction and User's Manual, Techincal Report SMLI TR-93-6, Sun Microsystems Laboratory, Mountain View, CA, February 1993.

[4] J. Dean, J. Hicks, C. Waldspurger, W. Weihl and G. Chrysos, ProfileMe: Hardware support for instruction-level profiling on out-of order processors, In Proceedings of International Symposium on Microarchitecture, December 1997.

[5] 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," 2008 Applied Superconductivity Conference (ASC 2008), 2EZ01, Chicago, Aug 2008.

[6] J. Makino, K. Hiraki and M. Inaba, GRAPE-DR: 2-Pflops massively-parallel computer with 512-core, 512-Gflops processor chips for scientific computing, SC07 2007.

[7] Memory Roadmap, http://tw.renesas.co

[8] S. Obara and A. Saika, Efficient recursive computation of molecular integrals over Cartesian Gaussian Functions, J. Chem. Phys., Vol.84, pp.3963, 1986.

[9] W.H. Press, B.P. Flannery, S.A. Teukolsky, and T.W. Vetterling, Numerical Recipes in C, Cambridge University Press, 1988.

[10] N. Sherwani, Algorithms for VLSI physical design automation, Kluwer-Academic Publishers, 1999.

[11] D.C. Suresh, W.A. Najjar, F. Vahid, J.R. Villarreal, G. Stitt, Profiling Tools for Hardware/Software Partitioning, Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems, 2003.

[12] 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., E91-C(3):350-355, 2008.

[13] TOP500 Supercomputer, http://www.top500.org/.

[14] W. Wulf and S. McKee "Hitting the Memory Wall: Implications of the Obvious," ACM SIGArch Computer Architecture News, 23 (1):20-24, March 1995.

[15] C. Young, The Harvard Atom Like Tool Manual (HALT), http://citeseer.nj.nec.com/121315.html.