

# Mapping Scientific Applications on a Large-Scale Data-Path Accelerator Implemented by Single-Flux Quantum (SFQ) Circuits

Mehdipour, Farhad

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

Honda, Hiroaki

Institute of Systems, Information Technologies and Nanotechnologies

Kataoka, Hiroshi

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

Inoue, Koji

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

他

<https://hdl.handle.net/2324/16810>

---

出版情報 : Proceedings of Design, Automation & Test in Europe. 2010, pp.993-996, 2010-03  
バージョン :  
権利関係 :

# Mapping Scientific Applications on a Large-Scale Data-Path Accelerator Implemented by Single-Flux Quantum (SFQ) Circuits

Farhad Mehdipour<sup>1</sup>, Hiroaki Honda<sup>2</sup>, Hiroshi Kataoka<sup>1</sup>, Koji Inoue<sup>1</sup>, Irina Kataeva<sup>3</sup>,  
Kazuaki Murakami<sup>1</sup>, Hiroyuki Akaike<sup>3</sup>, Akira Fujimaki<sup>3</sup>

<sup>1</sup> School of Information Science and Electrical Engineering, Department of Informatics, Kyushu University, Fukuoka, Japan

<sup>2</sup> Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka, Japan

<sup>3</sup> Department of Quantum Engineering, Nagoya University, Nagoya, Japan

Email: {farhad, dahon, kataoka}@c.scse.kyushu-u.ac.jp

**Abstract**— To overcome issues originating from the CMOS technology, a large-scale reconfigurable data-path (LSRDP) processor based on single-flux quantum circuits is introduced. LSRDP is augmented to a general purpose processor to accelerate the execution of data flow graphs (DFGs) extracted from scientific applications. Procedure of mapping large DFGs onto the LSRDP is discussed and our proposed techniques for reducing area of the accelerator within the design procedure will be introduced as well.

**Keywords**- Reconfigurable accelerator; single-flux quantum circuit; data flow graph; placement and routing

## I. INTRODUCTION

These days, 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. As a solution, a hybrid architecture comprising an accelerator augmented to a GPP can be exploited for special purpose computations [1][5]. 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 [9]. A desk-side tera-flop scale computer has been introduced 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) [9].

A SFQ circuit is based on the superconductor technology which includes low power consumption and high speed compared to the CMOS circuits. It 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. High-speed switching and signal transmission, low power consumption, compact implementation (small area), suitability for pipeline processing of data stream are the main features of the SFQ technology [4].

Developing necessary tools for compiling applications, generating data flow graphs (DFGs) and configuration bit-streams as well as designing and fabricating LSRDP architecture are the main phases of implementation of the target computer. In the architecture side, the main components of LSRDP as well as structure of routing resources which all are implemented by SFQ circuits will be discussed in this paper. As a basic component of the tool chain we will concentrate on the DFG mapping tool and

how it is exploited during the design procedure of the LSRDP. Basic specifications of the LSRDP architecture and constraints originating from the SFQ circuits ought to be taken into account within the tool development and LSRDP design procedure. Placing input nodes on appropriate locations is a key step in the placement procedure which highly impact the final cost which is based on the connection length. Connection length is also a key factor in turn which strongly impacts the routing resources area and overall area of the accelerator. A placement algorithm for input ports as well as an alternative to connection length measurement are proposed that can be effective in reducing the implementation cost. We evaluate the result of utilizing proposed techniques during the LSRDP design procedure. Moreover, a benchmark of computational-intensive scientific applications are introduced which are attempted for designing the target hardware.

## II. LSRDP GENERAL ARCHITECTURE AND SPECIFICATIONS

Fig. 1 displays the overall architecture of the proposed high-performance computer consisting of a GPP, an LSRDP as 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 a number of PEs in the next row. Data flow graphs are pulled-out from critical segments of applications and configuration bit-streams are generated for them. During execution of an application, configurations associated with the critical segments are loaded onto the LSRDP and executed in a pipeline fashion. The main specifications of the LSRDP architecture are presented here.

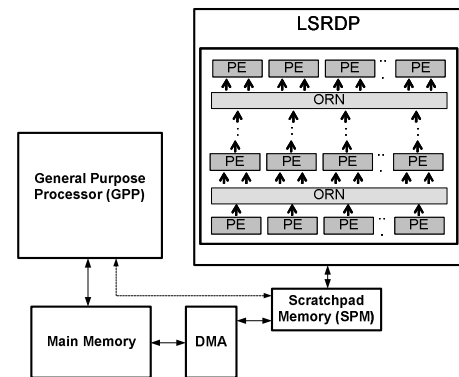


Fig. 1. Overall architecture of the SFQ-LSRDP computer

**Layout:** Layout of the LSRDP indicates the type of functional units and their distribution. Two types of layout are examined for LSRDP during the design procedure. In a normal layout (Layout-I), each FU can implement any operation including ADD/SUB and MUL. In Layout-II only one of operations ADD/SUB or MUL is implemented in each PE.

**Input/Output ports:** I/O ports are located on top and bottom borders of the LSRDP. 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.

**PE types:** Three PE architectures are examined for the LSRDP (Fig. 2). Most suitable PE is selected during the design procedure. The basic PE architecture includes an FU for implementing desired operation and a TU (transfer unit). As ORNs provide only routing resources between consecutive rows, TUs are utilized to connect two PEs locating on inconsecutive rows. 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). The second PE architecture has one additional TU for increasing the flexibility of routing and it has 4-inputs/3-outs. The third PE architecture has a similar architecture to the first one, the difference is in possibility of implementing two simultaneous TUs

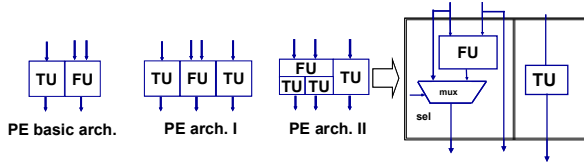


Fig. 2. Various PE structures

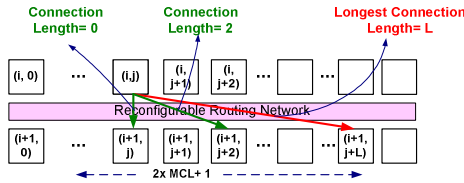


Fig. 3. Definition of the connection length and max. connection length (MCL)

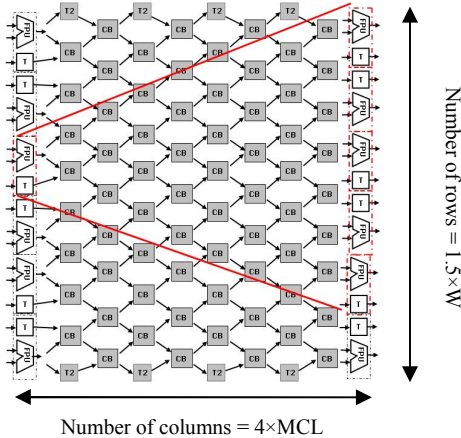


Fig. 4. Structure of a CB-based ORN for the PE basic arch. (MCL=2)

by the FU (totally three TUs). An additional mux should be used inside the PE to choose between FU's output and the input.

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

**Operand routing network (ORN):** PEs of each row are connected to the PEs in the next row through ORNs as routing resources. Fig. 3 shows the definition of the connection length and the maximum connection length (MCL) on a piece of LSRDP architecture. It can be seen that the connection length of two PEs is the horizontal distance between them. ORNs' functionality is similar to a multiplexer, however in the SFQ implementation; ORNs are composed of cross-bar switches (CBs). Similar to other components of the LSRDP, CBs are also implemented by means of Josephson Junctions as the basic elements of the SFQ circuits [3]. An ORN for the PE architecture consisting of one FUs and a TU (3-input/2-output) is presented in Fig. 4. 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.

A united ORN is implemented between each two rows rather than implementing separate ORNs for each PE. In Fig. 4 it is assumed that the MCL is equal to two, therefore, each PE in the left side (a row of PEs) can be connected to any of five ( $2 \times \text{MCL} + 1$ ) PEs in the right side (consecutive row). The number of PEs in a row, MCL size and the number of input/outputs for each PE are the main factors which affect the ORN size. In Fig. 4., assuming  $W$  as the number of PEs in each row (LSRDP width), the ORN will consist of  $2 \times W \times (4 \times \text{MCL})$  CBs.

### III. LSRDP DESIGN PROCEDURE AND MAPPING TOOL

We used a quantitative approach for designing the LSRDP architecture and determining its detailed architectural specifications. Whole the design procedure is an iterative process of gathering statistics and analyzing the results. First, DFGs are mapped onto the LSRDP and the outcome is analyzed. The mapping is performed without forcing any constraint except the constraints originated from LSRDP architecture e.g. unidirectional data flow over the PE rows, availability of routing resources between subsequent rows and etc. Then, the results of mapping are analyzed to decide an appropriate value for an intended parameter.

#### A. DFG Extraction

Four applications are attempted as scientific benchmark applications including: one-dimensional heat (referred as Heat) and vibration equations (Vibration), two-dimensional Poisson equation (Poisson) [7], and recursion calculation part of electron repulsion integral (ERI [6]) as a quantum chemistry application. All calculations consist of ADD, SUB, and MUL operations and DFGs have been extracted manually.

#### B. 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 [8]. Generally, minimizing the total connection length or the maximum connection length are main objectives, however in designing LSRDP, the main goal is to

minimize the maximum connection length (MCL) that directly impacts the ORN size and the LSRDP area as well.

Routing process is the next stage that establishes connections between the PEs by means of ORNs and transfer units [8]. 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 subsequent rows should be less than the available connection lengths provided by the ORNs.

### C. I/O Nodes Placement

Input/output nodes of the DFG should be located on appropriate input/output ports of the LSRDP. ORNs as routing resources exist between the first/last LSRDP rows and input/output ports. The main objective is to reduce the connection length between input/output ports and PEs in the first/last row of the LSRDP. Since DFG nodes are placed based on the location of their parents inside the LSRDP, placing input nodes is performed in a different manner from the placing output ports and it has more impact in the quality of final placement. Proper locations for output nodes can be determined based on the position of parent nodes which have already been placed.

In a simple placement algorithm for input nodes referred as *fan-out based placement*, I/O nodes are placed with respect to their fan-out or the total number of children. First, input nodes of DFG are prioritized with respect to their fan-out and then the placement algorithm looks for proper locations for them over the input ports to minimize the longest connection length.

Here a new algorithm (referred as *proximity factor-based placement*) for input node placement is proposed. The main intuition behind the proposed algorithm is to locate input nodes which have stronger connections to each other in a closer distance. We define a factor referred as proximity factor to represent the strength of forces that PEs can exert to each other.

**Proximity factor:** for each pair of input nodes  $i$  and  $j$  their proximity factor is calculated as:

$$p_{i,j} = \sum_{k \in S_{i,j}} \frac{1}{D_{k,i}} \quad (1)$$

while  $D_{k,i}$  is the distance of common descendant node  $k$  to the input ports  $i$  (and  $j$ ). The distance of a node to its related input port can be calculated as its ASAP (As soon as possible [2]) level of execution.  $S_{i,j}$  is the set of common descendants for the input nodes  $i$  and  $j$ . Larger number of common descendants with smaller distance to the parents pair result in larger proximity factor and therefore, corresponding nodes should be placed in a closer distance to each other. A heuristic algorithm is introduced below which employs the proximity factor for positioning input nodes of a DFG onto the LSRDP input ports.

#### Some Definitions:

$L$ : is the input ports array which stores the list of placed nodes such that the indexes indicate the location of corresponding input node.

$P$ : is the matrix of proximity factors for input node pairs.

$$P = \begin{bmatrix} \infty & p_{1,2} & \dots & p_{1,n} \\ p_{2,1} & \infty & \dots & p_{2,n} \\ \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & \infty \end{bmatrix} \quad (2)$$

$$p_{i,j} = p_{j,i} = \begin{cases} \infty & \text{if } i = j \\ \sum_{k \in S_{i,j}} \frac{1}{D_{k,i}} & \text{if } i \neq j \end{cases} \quad (3)$$

$l$  and  $r$ : denote the index of candidate locations in  $L$  for placing the under process input node.

$C_{l,m}$  and  $C_{r,m}$ : show the amount of proximity of an under process node  $m$  to the previously placed input nodes which have been located between  $l$  and  $r$  in array  $L$ . The node  $m$  will be placed in location  $l$  if  $C_{l,m}$  is larger than  $C_{r,m}$  otherwise, it will be located in location  $r$ .  $C_{r,m}$  and  $C_{l,m}$  are calculated as:

$$C_{l,m} = \sum_{i=l+1}^{r-1} \left( p_{i,m} \times \frac{1}{|i-l|} \right) \quad (4), \quad C_{r,m} = \sum_{i=l+1}^{r-1} \left( p_{i,m} \times \frac{1}{|r-i|} \right) \quad (5)$$

In (4) and (5),  $p_{i,j}$  is the proximity factor of node  $m$  and node  $i$  that has already been placed. The second term is the inverse of distance of candidate location ( $l$  or  $r$ ) to the location of node  $i$ . In this way, the strength of proximity to already placed nodes is examined and one of the candidate locations ( $l$  or  $r$ ) is chosen.

#### Placement alg.

1. Construct matrix  $P$  including the proximity factors for each pair of input nodes ( $n$  is the number of input nodes) and initialize  $L = \Phi$ .
2. Find node  $m$  with the highest proximity factor from the first row of matrix  $P(i=1)$  and place it in array  $L$  so that  $L[n/2] = m$ .
3. Initialize  $l$  and  $r$  to  $n/2-1$  and  $n/2+1$ , respectively.
4. Find the next node ( $m$ ) with the highest proximity factor from the first row of matrix  $P(i=1)$ .
5. Calculate  $C_{l,m}$  and  $C_{r,m}$  using (4) and (5).
6. if  $C_{l,m} > C_{r,m}$ :  
 $l = l+1, L[l] = m$  (node  $m$  is placed in the location  $l$ )  
else:  
 $r = r+1, L[r] = m$  (node  $m$  is placed in the location  $r$ )
7. if still there is unplaced input node, go to step 4.

### D. Connection-length minimization

The connection length between source and destination nodes is evaluated as the horizontal distance of two nodes (connection length =  $l_h = d_h$ ; horizontal distance). For two nodes located in two consecutive rows ( $d_v = 1$ ), the only possible way for routing is through the available ORN resources. On the other hand, for two nodes placed on inconsecutive rows (vertical distance:  $d_v > 1$ ), it is possible to use intermediate TUs for routing. In this way, the connection can be segmented to  $d_v-1$  connections between consecutive rows.

Considering above point, we use an alternative measurement for the connection length as  $l_{hv} = \lceil d_h/d_v \rceil$ . In Fig. 5 (left-side) the two definitions of connection length have been displayed. In right-side two connections (denoted by 1 and 2) can be seen so that  $d_{h1} = d_{h2} = 3$ , while they have different vertical distances ( $d_{v1} = 1, d_{v2} = 3$ ). For connection 1, connection length would be 3 and the only possible way for routing from *src* to *dest1* is via the ORN switches. On the contrary, for connection 2, it is possible to use available intermediate TUs and break it into three segmented connections from *src* to *dest2*. This results in reducing connection length to one. By using above new definition for connection length measurement, we modified the placement algorithm such that the vertical connection length becomes effective in calculating the cost function.

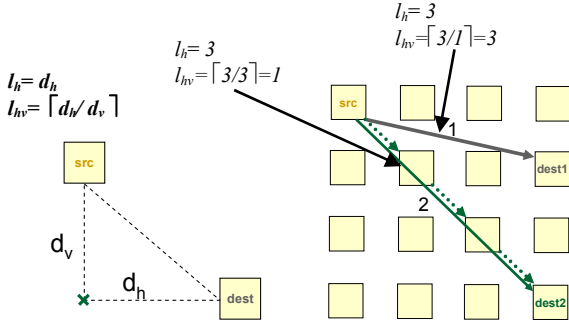


Fig. 5. Connection length definitions

#### IV. EXPERIMENTAL RESULTS

We conducted experiments to evaluate different LSRDP architectures exploiting the proposed mapping techniques. Input DFGs (including 10 DFGs) were extracted from the four target applications as mentioned in Section III-A. Three PE structures (Fig. 2) and two LSRDP layouts (introduced in Section II) were examined. Due to various number of inputs and outputs of the PE architectures, three ORN structures were designed and their area were calculated as in Table 1. In order to calculate PE area, it is assumed that the number of Josephson Junctions (JJs) required for implementing a TU or mux in a PE is almost %10 of a double-precision functional unit [3]. One basic requirement is to reduce the area of LSRDP because of limitations and difficulties in implementing large SFQ circuits.

Table 2 shows results of evaluations of various LSRDP architectures. We used both techniques presented in Section III-C for placing input nodes. In addition the MCL minimization technique introduced in Section III-D were exploited to reduce the MCL size. In Table 2, S1 denotes first strategy including fan-out based input node placement and using  $l_h$  as the cost of connection lengths. S2 stands for a strategy employing the proximity-factor based input node placement and  $l_{hv}$  for the connection length measurement. Results of experiments show the effectiveness of the proposed techniques in reducing MCL, ORN size as well as the overall LSRDP size. It is observed that implementing each CB needs around 550JJs and each FU i.e. ADD/SUB or MUL requires

Table 1. PE types and ORN area calculations

|     | # of inps/outs | PE Area (x FU) |           | ORN Area (x CBs)    |
|-----|----------------|----------------|-----------|---------------------|
|     |                | Layout-I       | Layout-II |                     |
| PE1 | 3/2            | 2.1            | 1.1       | 1.5x Wx (4 x MCL)   |
| PE2 | 4/3            | 2.2            | 1.2       | 2x Wx (6 x MCL+2)   |
| PE3 | 3/3            | 2.2            | 1.2       | 1.5x Wx (4 x MCL+1) |

Table 2. Evaluation results for various architectures

|                           |                | Layout-I |        | Layout-II |        |
|---------------------------|----------------|----------|--------|-----------|--------|
|                           |                | S1       | S2     | S1        | S2     |
| MCL                       | PE basic arch. | 14       | 6      | 15        | 12     |
|                           | PE arch. I     | 8        | 3      | 9         | 4      |
|                           | PE arch. II    | 10       | 4      | 12        | 7      |
| ORN size (overall) x CB   | PE basic arch  | 25116    | 12600  | 29250     | 24336  |
|                           | PE arch. I     | 30600    | 13680  | 38080     | 17680  |
|                           | PE arch. II    | 18696    | 8237   | 23520     | 14790  |
| No. of FUs (overall) x FU | PE basic arch  | 580      | 683    | 330       | 344    |
|                           | PE arch. I     | 634      | 713    | 384       | 384    |
|                           | PE arch. II    | 627      | 669    | 360       | 384    |
| Overall LSRDP Area (JJ)   | PE basic arch  | 36923K   | 34193K | 29200K    | 27040K |
|                           | PE arch. I     | 48083K   | 35995K | 36190K    | 25031K |
|                           | PE arch. II    | 35307K   | 31258K | 27266K    | 23451K |

around 40KJJs.

The LSRDP area is composed of the area of ORNs and PEs. Area of ORNs (first row) has been represented by the number of cross-bar switches and the area of PEs (second row) is estimated by the number of required FUs. Overall area of the LSRDP (the last row) has been reported in terms of the number of Josephson Junctions (JJs). Since each PE in Layout-I implements both ADD/SUB and MUL operations, it needs larger overall area regardless of the strategy of mapping. Comparing PE architectures, although PE arch. I gives smaller MCL size, but it results in larger overall area for both layout types. As PE arch. I has four inputs and three outputs which are more than those of two other PE architectures, therefore, ORN size would be bigger referring to Table 1. In Table 2 it can also be observed that the PE basic arch. and PE arch. II both obtain smaller overall area for LSRDP, however the MCL size for the first one is larger. Using mapping strategy S2, Layout-II and the PE arch. II the smallest area is achievable for the LSRDP.

#### V. CONCLUSION

A reconfigurable desk-side computer comprising an accelerator implemented by superconducting circuits has been introduced. This computer is suitable for executing massive computational-intensive scientific applications. Large DFGs were extracted from scientific applications and mapped onto the accelerator architecture to gain performance. A mapping tool has been developed in which the main goal is to reduce the overall routing resources size and the LSRDP area as well.

#### ACKNOWLEDGEMENT

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, <http://www.clearspeed.com/>.
- [2] G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994.
- [3] I. Kataeva et al., "An operand routing network for an SFQ reconfigurable data-path processor," IEEE Trans. Appl. Supercond., vol. 19, no. 3, pp. 665-669, 2009.
- [4] K. Likharev and V. Semenov, "RSFQ logic/memory family: a new Josephson junction technology for sub-terahertz clock frequency digital systems," IEEE Trans. on Appl. Supercond., vol. 1, no. 1, pp. 3-28, 1991.
- [5] J. Makino, K. Hiraki and M. Inaba, "GRAPE-DR: 2-Pflops massively-parallel computer with 512-core, 512-Gflops processor chips for scientific computing," Proc. of Supercomputing, 2008.
- [6] S. Obara and A. Saika, "Efficient recursive computation of molecular integrals over Cartesian Gaussian functions," J. Chem. Phys., vol.84, pp.3963, 1986.
- [7] W.H. Press, B.P. Flannery, S.A. Teukolsky, and T.W. Vetterling, Numerical recipes in C, Cambridge University Press, 1988.
- [8] N. Sherwani, Algorithms for VLSI physical design automation, Kluwer-Academic Publishers, 1999.
- [9] N. Takagi et al., "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.
- [10] W. Wulf and S. McKee "Hitting the memory wall: implications of the obvious," ACM SIGArch Computer Architecture News, 23 (1):20-24, 1995.