Evaluation of Various Routing Architectures for Multi-FPGA Boards

Sushil Chandra Jain, Shashi Kumar, Anshul Kumar
Department of Computer Science & Engineering,
Indian Institute of Technology, Delhi, Hauz Khas, New Delhi - 110 016
{scjain,shashi,anshul}@cse.iitd.ernet.in

Abstract

Multi-FPGA boards are being used for logic emulation, rapid prototyping, custom computing and low volume sub-system implementation. A key feature which characterizes these boards is their routing architecture(RA). Inter-FPGA connections in an RA can be of two types, namely fixed connections through direct wires and programmable connections through intermediate Field-Programmable Interconnect Devices. This paper presents an analytical approach for evaluating routing performance of an RA employing both types of connections . Our approach consists of two steps: 1) Generation of random interconnection requirement matrix for modeling real circuits with an orientation towards the available interconnection architecture. 2) Checking the routability of the generated matrix on the given RA.

The evaluation experiments for many RAs demonstrate that our methodology matches in results with purely experiment based approaches. However, it is more versatile in that it can evaluate new RAs. Also it provides more observability and controllability of architectural, circuit and performance parameters.

1 Introduction

Because of continuing increase in the capacity and speed of Field Programmable Gate Arrays (FPGAs), they have become cost effective implementation alternatives in the areas of reconfigurable computing, hardware emulation, rapid prototyping and low volume sub-systems. The capacity of a single FPGA is usually too small to implement practical systems, therefore, Multi-FPGA boards (MFBs) are commonly used. A large variety of MFBs have been developed by research laboratories[1, 2] and commercial companies[3, 4] to meet the requirement of larger capacity. A generic MFB is shown in Fig. 1 consisting of FPGAs, Field-Programmable Interconnect Devices(FPIDs), memories and their interconnection networks. The features which distinguish various MFBs include:

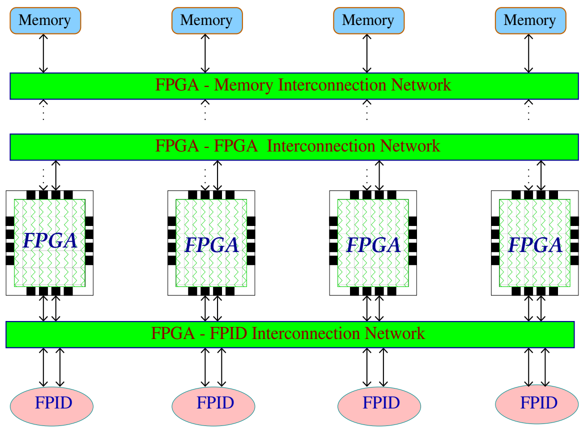

Figure 1. A Generic Multi-FPGA Board

1) Interconnection network among FPGAs, including fixed and programmable connections through FPIDs.

2) Type and size of local memory with each FPGA, and type and size of global shared memory among FPGAs.

3) Type and size of buses for connecting the MFB to a host processor.

4) Type and size of memory shared between MFB and the host.

5) The type of protocols used for MFB and host communication.

In this paper, we discuss and evaluate various options for the first feature, namely, the interconnection network or the Routing Architecture (RA) of an MFB. Criterion for evaluation of an RA depends upon the application of the MFB. We are concerned only with rapid prototyping or hardware emulation applications in this paper. For these applications, the main concern is the ability of the RA to satisfy the interconnection requirement of the circuits to be emulated. For mapping a large circuit on an MFB, it is first partitioned into parts such that each part can fit (in terms of logic resources required and I/O pins required) into a single FPGA. For a partitioned circuit to successfully map on an MFB, the RA of the MFB should satisfy its interconnection requirement.

Various researchers[5, 6, 7] have evaluated RA's of available MFB's . Most of the evaluations are experimental using some benchmark circuits. Because of the small set of benchmark circuits, the confidence level in the conclusions can not be very high. Also, the results reported critically depends on the specific circuit partitioner used. In contrast to this, we propose an analytical approach for evaluating a large number of RA's. We also propose a parameterized methodology for synthetically generating benchmark circuits partitioned for target RA.

The rest of this paper is organized as follows: In the next section, we discuss various RAs for MFB, the model for evaluation is developed in 3rd section. Section 4 elaborates the procedure for generating the test data. Results are shown in section 5, followed by conclusions in section 6.

2 Routing Architecture of MFBs

As shown in Fig. 1, an RA, generally, consists of two parts: 1) Fixed connections among FPGA. 2) Inter-FPGA connections through FPID types of chips. The cost of the MFB grows with the requirement of programmable connections. On one extreme, we can have an RA with only fixed topology among FPGAs. One/two dimensional meshes and many of its variations have been used for implementing fixed connections[2]. One interesting variation of mesh topology is known as One-Hop Mesh[8]. Without the availability of programmable interconnections, many partitioned circuits (partitions) may not map on MFB. On the other extreme, if all the interconnections among FPGAs are routed through programmable devices, then the routing requirement of all partitions can be met. Lewis et al have used this connection style in[1]. In this case, the cost of the required number of programmable devices may be very large. In general, the cost of the FPIDs depends on the fraction of FPGA pins which are routed through FPIDs along with the FPGAs-FPIDs topology. Therefore, there is a large design space for RA's for design of MFBs providing trade-off between the cost and percentage of circuits which can be prototyped. A compromise of having partially programmable pins as well as partially fixed pins is used in Hybrid RAs. Khalid et al[7] have used this approach in *Hybrid Complete Graph Partial -crossbar(HCGP)* architecture.

3 Evaluation Model of RAs

In this section, we develop a model for evaluation. The model is based on a set of equations for routability testing.

3.1 Evaluation Process

Fig. 2 shows the flow in our evaluation process. Input to the evaluation process is the routing architecture in terms of topology and the number of fixed connections among FPGA's and the programmable connection between an FPGA and the switch. The circuit is modeled by the interconnection requirements between all pairs of partitions. All connections are assumed to be bidirectional and no self connection is made. Routability checker module evaluates to find if the interconnection requirement is satisfied by the available routing resources. Statistics recorder module collects the success percentage of emulation of circuits on a particular routing architecture for large number of test data generated by inter-partition routing requirement generator.

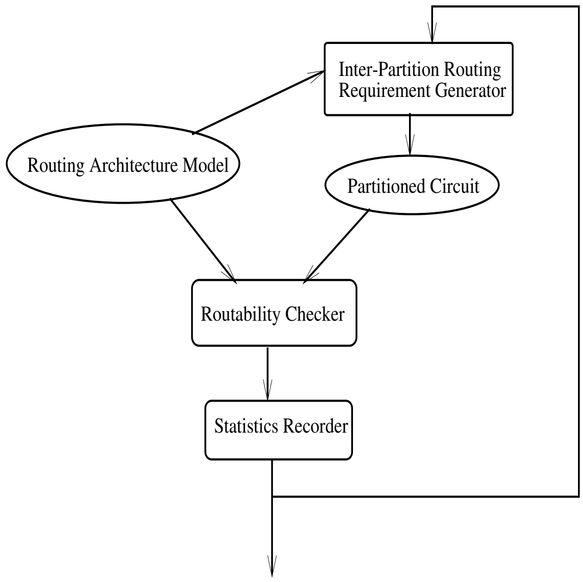

Figure 2. Flow of Evaluation Process

3.2 Architecture Modeling and Routability

Fixed and Programmable connections

Fixed connections among n FPGAs can be represented by an interconnection matrix X of size $n \times n$, where $X_{ij} =$ no. of wires connecting i^{th} and j^{th} FPGAs, for $i, j = 1 \ldots n$.

Interconnections between the switch and FPGAs can be represented by a vector S of size n, where $S_i =$ no. of wires connecting i^{th} FPGA to the switch.

Let N be the vector of size n denoting the no. of I/O pins available with FPGAs. That is, $N_i =$ no. of I/O pins on i^{th} FPGA.

The total number of wires connecting i^{th} FPGA to other FPGAs $(\sum_{i=1}^{n} X_{ij})$ or the switch (S_i) are limited by N_i. That is,

$$X \cdot U + S \quad \leq \quad N \tag{1}$$

where $\leq$ denotes element by element inequality and U denotes a vector of size n consisting of all 1's.

Switch Configurations

The switch can be configured to establish a pattern of interconnections among the FPGAs according to the requirement of a given circuit to be emulated. We represent a switch configuration by the matrix of the interconnections provided by it. That is, for a switch configuration L ,

L_{ij} = no. of connections between i^{th} FPGA and j^{th} FPGA provided by the switch in configuration X.

We assume that the switch is used to establish only point to point connections. That implies that the total no. of switched connections between i^{th} FPGA and other FPGAs ($\sum_{i=1}^{n} L_{ij}$) is limited by the no. of connections between i^{th} FPGA and the switch (S_i). That is,

$$L \cdot U \quad \leq \quad S \tag{2}$$

We assume that the switch is a full crossbar and non blocking, implying that $L \cdot U$ can be equal to S.

Routability of a circuit

The circuit to be emulated by an n FPGA board needs to be partitioned into a network of partitions such that

- no. of partitions $\leq n$,
- For $i = 1 \ldots n$, i^{th} partition is realizable in terms of logic capacity by i^{th} FPGA,
- For $i = 1 \ldots n$, the number of wires connecting i^{th} partition to other partitions $\leq N_i$.

After a circuit has been successfully partitioned meeting the above requirements, we can denote the interconnection requirements by an interconnection matrix A of size $n \times n$, where A_{ij} = no. of wires required between i^{th} and j^{th} partition .

The last condition of acceptable partitioning can be denoted by

$$A \cdot U \quad \leq \quad N \tag{3}$$

Then such a partition with interconnection matrix A can be emulated by multi-FPGA board with fixed FPGA-FPGA connection denoted by X and FPGA-Switch connections denoted by S if

$$\exists L \mid L \cdot U \quad \leq \quad S \tag{4}$$

and

$$X + L \quad \geq \quad A \tag{5}$$

where $\geq$ denotes element by element inequality.

Here we assume that each inter-partition connection is emulated by either a direct inter-FPGA connection or by a switched connection.

3.3 Interconnection matrices for specific architectures

We now look at matrix X for various common architectures for the fixed connections. First we make some common assumptions and define some symbols.

1) All FPGAs have same number of I/O pins. Let $N_i = N_x \ldots\ldots \forall i = 1 \ldots n$,

2) The fraction of I/O pins involved in programmable connections is same for all FPGAs and is denoted by P. Thus $S_i = P \cdot N_x \cdots \forall i = 1 \ldots n$. Consequently, the number of pins used for fixed connection, denoted by X_D is also the same for all FPGAs i.e. $X_D = (1 - P) \cdot N_x$,

3) The fixed interconnection network is symmetrical between all FPGAs. In other words, the graph representing the topology of fixed interconnections has same degree of all its node. Let us denote this degree by d.

Then X_{ij} can take only two values: either $\frac{X_D}{d}$ if i^{th} and j^{th} FPGAs are connected or 0 if they are not connected. Let us denote $\frac{X_D}{d}$ by X_N.

Now X_{ij} for different interconnection network topologies is defined as follows:

Linear Array with end around connection ($i.e.$ $d = 2$),

$$X_{ij} = \begin{cases} X_N, & if \ |i-j| = 1 \ or \ (n-1) \\ 0, & otherwise \end{cases} \tag{6}$$

Complete Graph ($i.e.$ $d = n - 1$),

$$X_{ij} = \begin{cases} X_N, & if \ i \neq j \\ 0, & otherwise \end{cases} \tag{7}$$

For the *Mesh* based topologies, we need to define, (x, y) co-ordinates of every node. These can defined as

$$x_i = |i-1| \bmod \sqrt{n}$$
$$y_i = |i-1| \ div \ \sqrt{n}$$

Here n is assumed as perfact square.

Toroidal Mesh ($i.e.$ $d = 4$),

$$X_{ij} = \begin{cases} X_N, & if \ x_i = x_j \ \& \ |y_i - y_j| = 1 \ or \ \sqrt{n}-1 \\ X_N, & if \ y_i = y_j \ \& \ |x_i - x_j| = 1 \ or \ \sqrt{n}-1 \\ 0, & otherwise \end{cases} \tag{8}$$

One-hop Mesh (Toroidal) ($i.e.$ $d = 8$),

$$X_{ij} = \begin{cases} X_N, & if \ x_i = x_j, |y_i - y_j| = 1, 2, \sqrt{n}-2 \ or \ \sqrt{n}-1 \\ X_N, & if \ y_i = y_j, |x_i - x_j| = 1, 2, \sqrt{n}-2 \ or \ \sqrt{n}-1 \\ 0, & otherwise \end{cases} \tag{9}$$

4 Inter-partition Connectivity Matrix Generation

In this section, we generate the inter-partition connectivity matrix modeling partitioned real circuits. Several partitioners try to orient the interconnections favouring target architecture. However, in presence of other objective functions, such as minimizing delay in the partitioned design, logic duplication etc., the partitioner may be able to only partially orient towards target architecture. We model this generation process in two steps. First, we generate topology independent matrices and then orient them towards a specific topology in the second step.

4.1 Topology Independent Partitioned circuit generation

Routability in the partitioned design is concerned with the number of wires connected within the partitions. Each *Topology Independent Partition*(TIP) will treat every other partition as its neighbour. We model this process as follows:

Let us try to model a circuit partitioned into n parts with w interconnecting wires. The wires are thrown within partitions one after the other. Each pair of partitions is equally likely to get the thrown wire. This implies that the number of wires A_{ij} between pair $i - j$ $(for\ \ any\ \ i, j)$ will follow binomial distribution. As the total number of pairs is $\frac{n(n-1)}{2}$, the probability of any pair getting a wire is given by

$$p = \frac{2}{n(n-1)} \tag{10}$$

and probability that the value A_{ij} of this distribution has value l is given by

$$Prob(A_{ij} = l) = \begin{pmatrix} w \\ l \end{pmatrix} p^l (1-p)^{(w-l)} \tag{11}$$

where the value of l lies between 0 and w.

Routability eq.(3) imposes limit on w i.e. $w \leq \frac{1}{2}\sum_{i=1}^{n} N_i$. Normally, a fraction of maximum possible wires is used in every MFB. We define a parameter representing the fraction of wires used as *average pin utilization* u, such that $u = \frac{w}{\frac{1}{2}\sum_{i=1}^{n} N_i}$ or $u = \frac{2*w}{n*N_x}$. It should be noted here, the value of A_{ij} must be much less than the number of pins N_x on the target FPGAs . Because p is very small, the probability of A_{ij} exceeding N_x is very small unless u is close to 1.

4.2 Orientation towards Topology

Topology oriented partition(TOP) can be obtained, when partitioner tries to favour neighbouring partitions with more wires than distant partitions. However, their success depends mainly on the nature of circuit and other objective functions. The nature of circuit can be mainly classified into 3 categories. First category represents *Random Circuits,* found in control dominated systems. The connections are likely to be distributed equally among all partitions. Partitioner can do very little orientation towards topology. Second category is of *Highly Structured(HS) Circuits,* found in data path oriented systems. The connections are highly localized and partitioner can take significant advantage in orientation. Third category is of *Medium Structured(MS) Circuits,* found in mixed type of systems. The connections are partially local and partially distributed to all other partitions. Partitioner can take only partial advantage of these connections.

We model the orientation by a multiplying factor for A_{ij}, which reduces wires exponentially with topological distance between the partitions. We refer to the parameter of exponential factor as *Topology Orientation Factor* α. The following equation describes how B_{ij}, the no. of connections between i^{th} and j^{th} partition after orientation, is computed for A_{ij}.

$$B_{ij} = A_{ij} \cdot \beta \cdot \exp^{(-\alpha\ \star\ dist(i,j))} \ldots\ldots \forall i, j \tag{12}$$

where

α determines the rate of decrease of connections,

$dist(i, j)$ denotes the architectural distance between i^{th} and j^{th} FPGA,

β is the normalizing factor which is used to keep the average pin utilization same as it was before orientation. The value of β can be obtained by the following formula:

$$\beta = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} \cdot \exp^{(-\alpha\ \star\ dist(i,j))}} \tag{13}$$

We observe that the random circuits can be modeled with $\alpha = 0$, MS Ckts with $0 < \alpha < 0.5$ and HS Ckts with $\alpha \geq 0.5$.

There are other circuit generation approaches reported in literature[9, 10]. The former approach generates circuits which are targeted for a single FPGA. Though the later targets larger circuits, it can be used for MFBs also. However, unlike our approach, it generates unpartitioned circuits. We do not need to model the partitioner, as the matrix A is directly generated after taking into account partitioning effect.

5 Evaluation Experiments and Results

This section presents the experimental setup and the results for different α for four different architectures namely *Hybrid Linear Array partial-Crossbar(HLAC), Hybrid Toroidal Mesh partial-Crossbar(HTMC)*, *Hybrid One-hop Mesh partial-Crossbar(HOMC)* and *Hybrid Complete Graph Partial-crossbar (HCGP)*.

5.1 Evaluation Experiments

For our experiments, we assumed a board with 25 FPGA's each having 400 pins. For each topology and for values of α in the range of 0 to 2, we generated interconnection matrices for utilization u in the range of $0..0.8$.

For each α, u pair, a set of 500 interconnection matrices was generated. For each generated circuit and for each architecture, we evaluated the minimum percentage of programmable pins such that 90% (or 450 out of 500) are successfully routed. The results for values of $\alpha = 0,\ 0.25\ \&\ 1$ are depicted in Fig. 3 to 7.

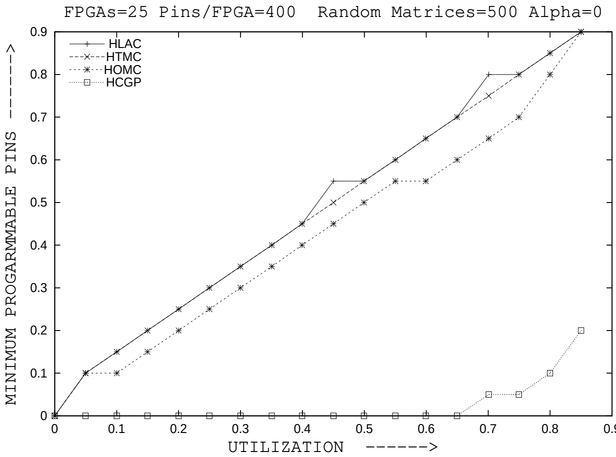

Figure 3. Results for random circuits with TIP

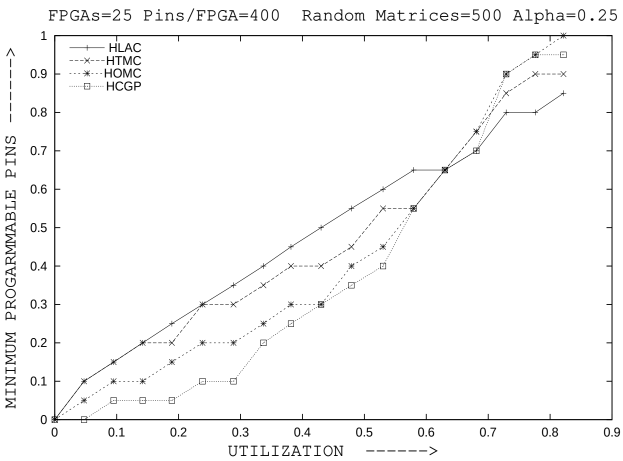

Figure 4. Results for MS Ckts with Linear Array TOP

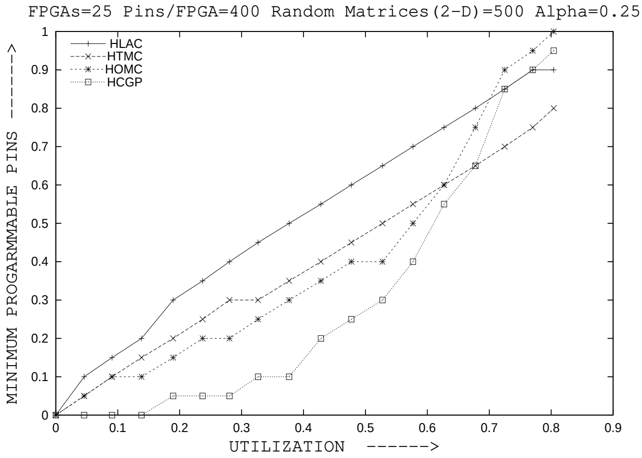

Figure 5. Results for MS Ckts with 2-D Mesh TOP

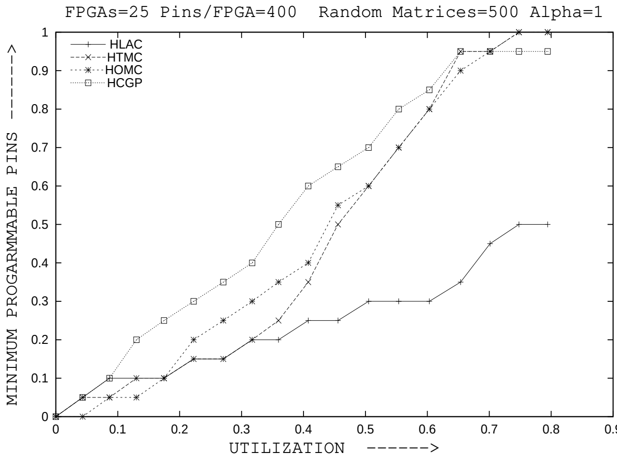

Figure 6. Results for HS Ckts with Linear Array TOP

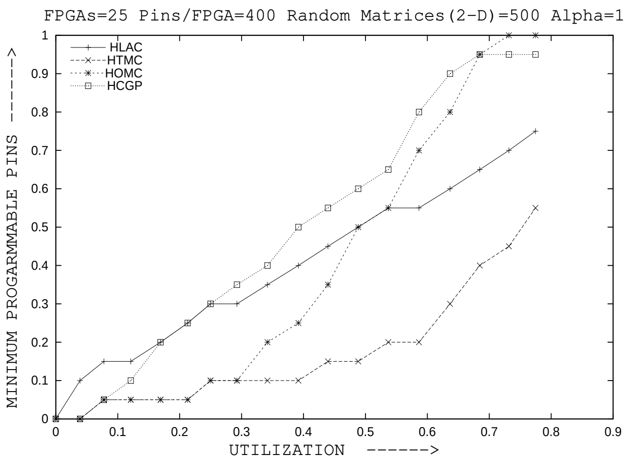

Figure 7. Results for HS Ckts with 2-D Mesh TOP

5.2 Results and Discussion

We have the following observations from the results of the experiments.

1. For random circuits ($\alpha = 0$), the HCGP topology has the best routability performance (Fig. 3). Our experiments show that less than 20% programmable pins are sufficient to route this category of circuits. This conclusion is in accordance with the conclusion reached by[7]. We observe that in most of the cases the percentage of programmable pins required are much smaller, 20% as compared to 40% in their study.

2. For highly structured circuits ($\alpha = 1$), the routability performance will depend on the match between the topology and the orientation (Fig. 6 & 7). This implies that linear array topology has the best performance if the partitioner has generated circuit with the orientation for the linear topology. This advantage increases with the increase in pin utilization. This is true for all topologies. However, the percentage of programmable pins required at higher utilization are in the range 40-50% for pin utilization close to 70%.

3. For medium structured circuits ($\alpha \simeq 0.25$), HCGP topology gives best performance for pin utilization upto 60% independent of the orientation of the partitioner (Fig. 4 & 5).

4. Based on the above results, it is possible to conclude that HCGP topology is the best choice for MFBs required for rapid prototyping of arbitrary circuits. Unfortunately, it is impractical to build boards, with many more than 25 FPGA's, using HCGP topology. Since scaling of HCGP topology is cost prohibitive.

5. It is expected that most of the large systems will be computation/data dominated, hence falling in the category of highly structured circuits. In this case, any MFB topology with the matching partitioner can give required routing performance. Our study shows that it is safe to keep 40% of the pins to be programmable.

6 Conclusions

We have presented an analytical technique to evaluate various interconnection architectures for MFBs, which is to be used for rapid prototyping. The basic strengths of our approach are:

• it matches in results with purely experimental approaches.

• it provides more accurate estimation of routing performance.

• it is more versatile in that it can evaluate new routing architectures.

• it provides more observability and controllability of architecture, circuit and performance parameters.

There are many interesting problems which are required to be investigated. In this paper, we have assumed that the required connection between two partitions is either implemented through a direct wire or is routed through a crossbar switch. It will be interesting to investigate the routability as well as delay performance, if we allow the connection to go through intermediate partitions (FPGAs) also.

Our analysis has been limited to routability analysis for rapid prototyping application of MFBs. Issues for evaluation will be different for other applications like using MFB as a co-processor. It will be important to optimize communication time between the host and MFB and communication time between various FPGAs on the board. It will be a useful future work to evaluate the communication time for various options in design of MFBs.

References

[1] D. Lewis *et al*, "The Transmogrifier-2: A 1 Million Gate Rapid Prototyping System," *IEEE Trans. on VLSI Systems*, vol. 6, no. 2, pp. 188–198, Jun 1998.

[2] S. Hauck, "The Roles of FPGAs in Reprogrammable Systems," *Proceedings of the IEEE*, vol. 86, no. 4, pp. 615–638, Apr 1998.

[3] "RIPP10 - The Reconfigurable Interconnect Peripheral Processor", Altera Corporation WWW Page : http://www.altera.com/html/programs/phd.html.

[4] "Spectrum: Reconfigurable Computing Plateform - Doc, Release 3.01", Giga Operations Corporations.

[5] M. A. S. Khalid and J. Rose, "Experimental Evaluation of Mesh and Partial-crossbar Routing Architectures for Multi-FPGA Systems," *IFIP IWLAS '97, Grenoble*, pp. 119–127, Dec 1997.

[6] Ubaid R. Khan *et al*, "Interconnect Networks and Routing Strategies in FPGA-based Hardware Emulators," *Journal of Microelectronic System Integration*, vol. 3, no. 2, 1995.

[7] M. A. S. Khalid and J. Rose, "A Hybrid Complete Graph Partial Crossbar Routing Architecture for Multi-FPGA Systems," *Proceedings of 6th ACM International Symposium on FPGAs,Monterey*, pp. 45–54, Feb 1998.

[8] S. Hauck, G. Borriello and C.Ebeling, "Mesh Routing Topologies for Multi-FPGA systems," *IEEE Trans. on VLSI Systems*, vol. 6, no. 3, pp. 400–408, Sep 1998.

[9] J. Darnauer and W.W-M. Dai, "A Method for Generating Random Circuits and its Application to Routability Measurement," *Proceedings of Fourth ACM/SIGDA International Symposium on FPGAs, Monterey*, Feb 1996.

[10] M. Hutton *et al*, "Characterization and Parameterized Random Generation of Digital Circuits," *Proceedings of the DAC, Las Vegas*, pp. 94–99, 1996.