

# Scalability Evaluation of a Hybrid Routing Architecture for Multi-FPGA Systems

Mohammed A. S. Khalid and Viktor Salitrennik

**Abstract**—Multi-FPGA systems (MFSs) are used as custom computing machines, logic emulators, and rapid prototyping vehicles. A key aspect of these systems is their programmable routing architecture which is the manner in which wires, FPGAs, and Field-Programmable Interconnect Devices (FPIDs) are connected. Several routing architectures for MFSs have been proposed and previous research has shown that the partial crossbar is one of the best existing architectures. A new routing architecture, called the Hybrid Complete-Graph and Partial-Crossbar (HCGP), was proposed by Khalid and was shown to provide superior speed and cost compared to partial crossbar. In this paper we address the issue of scalability of the HCGP routing architecture. The motivation for this work was to evaluate the suitability of the HCGP architecture for a future rapid prototyping system product that was being developed at Cadence. Experimental results show that the HCGP architecture is scalable and can be used with the state-of-the-art, high gate count FPGAs.

**Index Terms**—Partitioning, reconfigurable components, reconfigurable-computing, reconfigurable-systems, system-level.

## I. INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) are widely used for implementing digital circuits because they offer moderately high levels of integration and rapid turnaround time. Multi-FPGA systems (MFSs), which are collections of FPGAs joined together by programmable connections as illustrated in Figure 1, are used when the logic capacity of a single FPGA is insufficient, and when a quickly re-programmed system is desired. The typical applications of MFSs are for logic emulation [1], rapid prototyping [2], and reconfigurable custom computing machines [3].

The routing architecture of an MFS is the way in which the FPGAs, fixed wires, programmable interconnect chips are connected. The routing architecture has a strong effect on the

Mohammed A. S. Khalid is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada N9B 3P4 (corresponding author, phone: 519-253-3000, x2611; fax: 519-971-3695; e-mail: mkhalid@uwindsor.ca).

Viktor Salitrennik is with Cadence Design Systems, San Jose, CA 95134 USA (e-mail: viktor@cadence.com).

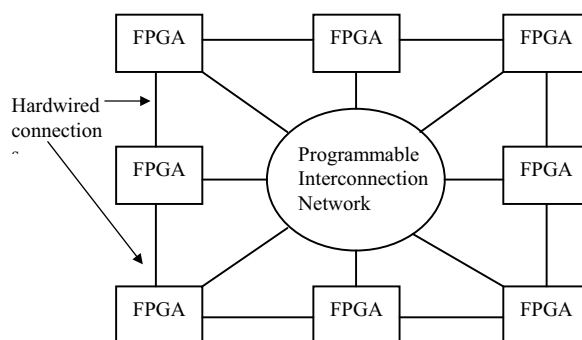


Figure 1. A Generic Multi-FPGA System

speed, cost, and routability of the system. Many architectures have been proposed and built and some research work has been done to empirically evaluate and compare different architectures. Previous research has shown that the partial crossbar [4] is one of the best existing architectures. A new routing architecture, called the Hybrid Complete-Graph and Partial-Crossbar (HCGP), was proposed by Khalid [5] and was shown to provide superior speed and cost compared to partial crossbar. The HCGP architecture uses a mixture of hardwired and programmable connections between the FPGAs whereas the partial crossbar uses only programmable connections. The speed and cost of the HCGP and partial crossbar architectures were compared experimentally, by mapping a set of 15 large benchmark circuits into each architecture. A customized set of partitioning and inter-chip routing tools were developed, with particular attention paid to architecture-appropriate inter-chip routing algorithms. Using the experimental approach, a key architecture parameter of HCGP, called percentage of programmable connections ( $P_p$ ), was also analyzed. Results showed that a  $P_p$  value 60% provided good routability for a variety of circuits. The HCGP architecture was licensed to Quickturn Design Systems (San Jose, California) and a U.S. patent was granted for this architecture [6].

In this paper we address the issue of scalability of the HCGP routing architecture. The previous experimental evaluation of this architecture and comparison to other architectures was done using relatively small FPGAs (compared to currently available FPGAs). This paper is organized as follows: in Section II we briefly describe the HCGP routing architecture. In Section III we discuss in detail

the scalability issues of the HCGP architecture and then describe the experimental procedure used to evaluate its scalability in Section IV. We present experimental results and conclude the paper in Section V.

## II. HCGP ARCHITECTURE DESCRIPTION

Ideally, all the FPGAs in an MFS should be connected using a single crossbar switch. Any connection between any set of FPGAs, irrespective of fanout, would be possible. Such a system would be always routable and would provide good speed. Unfortunately, such a crossbar switch is impractical for real systems because the size of the crossbar switch increases as square of the crossbar pins. The partial crossbar architecture [4] provides routability similar to that of a full crossbar (for real world netlists) at a much lower cost. The HCGP architecture provides lower cost and higher speed than

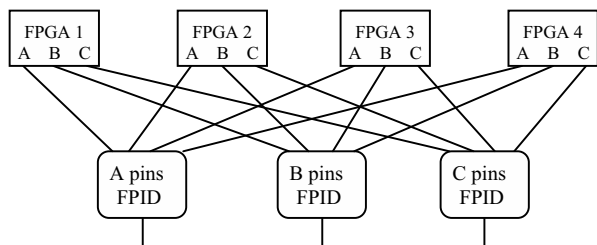


Figure 2. Partial Crossbar Architecture

the partial crossbar. In order to understand the main ideas behind the HCGP architecture, we first need to study the partial crossbar architecture. A partial crossbar using four FPGAs and three FPIDs is shown in Figure 2. The pins in each FPGA are divided into  $N$  subsets, where  $N$  is the number of FPIDs in the architecture. All the pins belonging to the same subset in different FPGAs are connected to a single FPID. Note that any circuit I/Os will have to go through FPIDs to reach FPGA pins. For this purpose, a certain number of pins per FPID are reserved for circuit I/Os. The number of pins per subset ( $P_t$ ) is a key architectural parameter that determines the number of FPIDs needed and the pin count of each FPID. The extremes of the partial crossbar architecture can be illustrated by considering a system with four FPGAs, and assuming 192 usable I/O pins per FPGA: a  $P_t$  value of 192 will require a single 768-pin FPID that acts as a full crossbar. A  $P_t$  value of 1 will require 192 4-pin FPIDs. Both of these cases are impractical. A good value of  $P_t$  should require low cost, low pin count FPIDs, and provide good routability.

The HCGP architecture for four FPGAs and three FPIDs is illustrated in Figure 3. The I/O pins in each FPGA are divided into two groups: hardwired connections and programmable connections. The pins in the first group connect to other FPGAs and the pins in the second group connect to FPIDs. The FPGAs are directly connected to each other using a complete graph topology, i.e. each FPGA is connected to

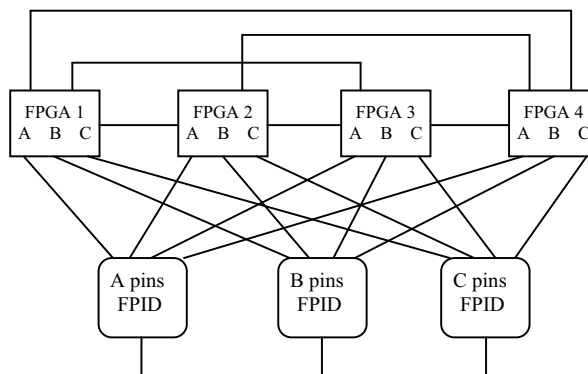


Figure 3. HCGP Architecture

every other FPGA. The connections between FPGAs are evenly distributed, i.e. the number of wires between every pair of FPGAs is the same. For programmable connections, the FPGAs and FPIDs are connected in exactly the same manner as in a partial crossbar. As in the partial crossbar, any circuit I/Os will have to go through FPIDs to reach FPGA pins. For this purpose, a certain number of pins per FPID are reserved for circuit I/Os. The direct connections between FPGAs can be exploited to obtain reduced cost and higher speed.

A key architectural parameter in the HCGP architecture is the percentage of programmable connections,  $P_p$ . It is defined as the percentage of each FPGA's pins that are connected to FPIDs (the remainder are connected to other FPGAs). The choice of a value of  $P_p$  involves tradeoffs between routability, speed, and cost. If  $P_p$  is too high it will lead to increased pin cost and lower speed, if it is too low it will adversely affect routability. If  $P_p$  is 0% the HCGP architecture degrades to a completely connected graph of FPGAs with no FPIDs used. If  $P_p$  is 100% the HCGP architecture degrades to a standard partial crossbar. Previous research [5] has shown that a  $P_p$  value of 60% is a suitable choice for obtaining good routability and speed at a reasonable cost.

## III. HCGP SCALABILITY ISSUES

The previous experimental evaluation of the HCGP architecture and comparison to other architectures [5] was done using relatively small FPGAs (compared to currently available FPGAs). The architecture can scale in three ways:

1. We can keep the FPGA logic and pin capacity constant and increase the total number of FPGAs.
2. We can keep the total number of FPGAs relatively small and increase the logic and pin capacity of each FPGA
3. We can increase both the total number of FPGAs and the logic and pin capacity of each FPGA.

The scalability issue 1 was addressed by using a hierarchical architecture such as the Hardwired Clusters Partial Crossbar (HWCP), proposed by Khalid [7]. Scalability issue 2 has not been explored so far for the HCGP architecture and is the subject of this paper. Note that scalability issue 3 is

a combination of scalability issues 1 and 2.

As FPGA logic and pin capacities continue to rise, it makes sense to use a limited number (say, 16 or less) of very high capacity FPGAs for creating MFSs that can be used for logic emulation or rapid prototyping of small to medium sized designs. This way we avoid the costs associated with using high pin count connectors and expensive boards for multi-board systems, that would be needed if we use many tens or a few hundreds of smaller FPGAs. For handling very large designs, processor-based emulators such as Cadence's Palladium are proving to be more effective than FPGA-based emulators [8].

#### IV. EXPERIMENTAL OVERVIEW

To evaluate the scalability of the HCGP architecture for large FPGAs, we first had to generate synthetic netlists similar to post-partition netlists produced for real multi-million gate designs. For the experiment we chose netlists consisting of 6, 8, 12, and 16 FPGAs. In order to resemble the real netlists, the netlist generation process was not completely random but followed some statistical patterns derived from real multi-million gate design netlists. First, consider the issue of the net fanout distribution in the synthetic netlist. We took real design partitioning results and collected statistical data on the nets distribution according to the fanout. On different types of real design partitioning results we determined typical distribution of nets connecting two FPGAs, three FPGAs, four FPGAs, etc. We reproduced the same distribution while randomly generating the connections in the synthetic netlists.

Second, post-partition netlists may vary on how evenly the connections are distributed between the FPGAs. A netlist may consist of FPGAs that have approximately the same number of connections to each other. In a more typical case there are clusters of tightly connected FPGAs, where there are more connections between FPGAs inside a cluster than between FPGAs that belong to different clusters. In our experiment we generated four types of netlists with different connection patterns. In the first pattern all FPGAs were connected to each other by approximately even numbers of nets. In the second pattern the netlists consisted of tightly connected two-FPGAs clusters. In the third pattern the netlists consisted of tightly connected three-FPGAs clusters. Finally, the last pattern included one cluster of two tightly connected FPGAs with the rest of the FPGAs connected to each other by approximately even numbers of nets. Note that this issue deals with the amount of "locality" in post-partition netlists. Replicating "locality" of real post-partition design netlists in synthetic netlists is a very elusive task and there has been little success in this respect in research efforts to date [9]. Fortunately, synthetic netlists produced using our approach are usually much more difficult to map compared to real netlists. Hence they yield a conservative evaluation of architecture and/or mapping CAD tools (rather than overly optimistic evaluation results).

Each netlist was derived using FPGA I/O pin utilization ranging from 50% to 100%. Then each generated netlist was sequentially mapped into HCGP architectures with  $P_p$  ranging from 0% to 100%. The numbers of FPGA and FPID I/O pins were assumed to be 1024 and 500 respectively. For every architecture, we tried to route the mapped netlist. We developed an architecture-specific router that restricted the number of chip hops for routing a net to one or two. A chip hop is defined as a pin-to-pin connection between two chips.

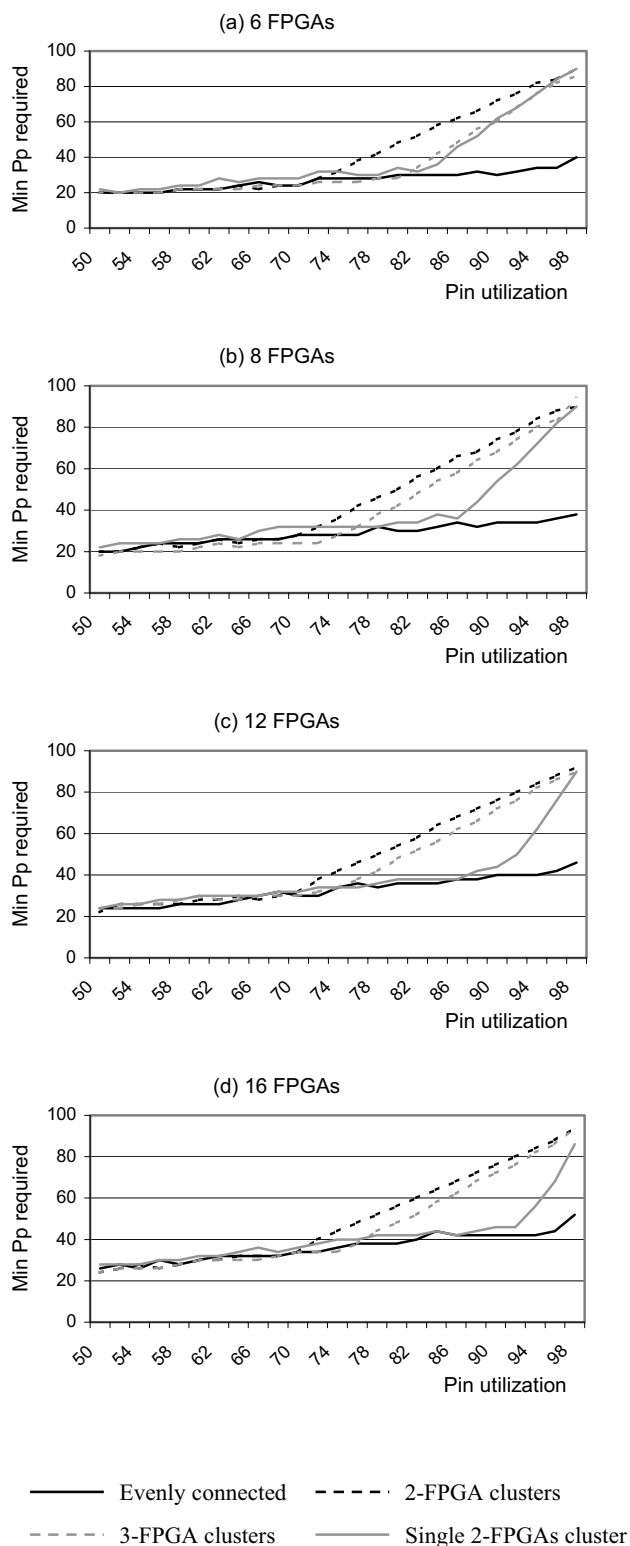
Hence, the routability of the HCGP architecture was evaluated for different combinations of (a)  $P_p$  value, (b) pin utilization per FPGA, (c) total number of FPGAs (varied from 6 to 16), and (d) FPGA interconnection pattern. The goal was to find a minimum value of  $P_p$  that provides routability for all cases depending on the I/O pin utilization.

#### V. RESULTS AND CONCLUSIONS

In this section, we present the experimental results obtained by mapping synthetic post-partition netlists to different configurations of the HCGP architecture. Recall from previous sections that our objective is to evaluate the routability of the HCGP architecture using large FPGAs. We are also interested in the value of  $P_p$  that results in routing completion in most cases.

The experimental results are shown in Figure 4 which consists of four graphs, each characterized by the number of FPGAs used in the HCGP architecture. We used synthetic post-partition netlists obtained using 6, 8, 12, and 16 FPGAs and mapped each to an HCGP architecture that used the same number of FPGAs. The FPGA pin utilization (shown on the X-axis) used in the synthetic netlist was varied from 50 to 100%. Each pin utilization case was mapped to the HCGP architecture using different values of  $P_p$  (shown on the Y-axis). There were four different types of netlists used: evenly connected FPGAs, collection of 2-FPGA clusters, collection of 3-FPGA clusters, and finally one 2-FPGA cluster with rest of the FPGAs evenly connected.

The results show that a  $P_p$  value of 60% is sufficient for achieving routing completion for all types of netlists provided we restrict the FPGA pin utilization to 82%. This is in agreement with previous research results [1] if we consider that in real design netlists, the *average* pin utilization per FPGA would likely be less than 80%. We have confirmed this assumption by pin utilization statistics collected on ten real designs. Recall that for netlists used in our experiments, FPGA pin utilization of 82% implies every single FPGA has 82% of its pins used. This is even more conservative than what would be expected in real design netlists.



**Figure 4. Experimental results: (a) for 6 FPGAs, (b) for 8 FPGAs, (c) for 12 FPGAs and (d) for 16 FPGAs**

We can conclude from the experimental results that the HCGP architecture is scalable using very large FPGAs, such as Xilinx Virtex II [10], and can be used to handle multi-million gate designs.

With improved fabrication technology, the logic capacity of FPGAs increases much faster than their pin capacity due to I/O pad placement limitations. Obviously FPGA logic utilization in HCGP, i.e. logic capacity, can be further improved if we use pin multiplexing on FPGAs to effectively increase the number of inputs and outputs per FPGA.

REFERENCES

- [1] Mentor Graphics, (2006) VStationPRO Datasheet. [Online] Available: <http://www.mentor.com>
- [2] C. Chang *et al*, "Implementation of BEE: a Real-time, Large-scale Hardware Emulation Engine," *Proc. of International Symposium on FPGAs*, 2003, pp. 91-99.
- [3] Timelogic Corp., (2004) DeCypher Bioinformatics Accelerator, [Online] Available: <http://www.timelogic.com>
- [4] M. Butts, J. Batcheller, and J. Varghese, "An Efficient Logic Emulation System," *Proc. of IEEE International Conference on Computer Design*, pp. 138-141, 1992.
- [5] M. A. S. Khalid and J. Rose, "A Novel and Efficient Routing Architecture for Multi-FPGA Systems," *IEEE Transactions on VLSI*, February 2000, Vol. 8, No. 1, pp. 30-39.
- [6] M. A. S. Khalid and J. Rose, "Multi-logic Device Systems Having Partial Crossbar and Direct Interconnection Architectures", *U.S. Patent No. 6604230*, issued August 5, 2003.
- [7] M. A. S. Khalid and J. Rose, "Hardwired-Clusters Partial-Crossbar: A Hierarchical Routing Architecture for Multi-FPGA Systems," *Proc. of the 1999 Sixth Reconfigurable Architectures Workshop (RAW'99)*, Springer, pp. 597-605, April 1999.
- [8] Cadence Design Systems, (2006) Palladium Datasheet. [Online] Available: <http://www.cadence.com>
- [9] M. Hutton, J. Rose and D. Corneil, "Automatic Generation of Synthetic Sequential Benchmark Circuits," in *IEEE Trans. on CAD*, Vol. 21, No. 8, August 2002, pp. 928-940.
- [10] Xilinx, Inc., (2006) Virtex II FPGAs Datasheet. [Online] Available: <http://www.xilinx.com>