Hardwired-Clusters Partial-Crossbar: A Hierarchical Routing Architecture for Multi-FPGA Systems

Mohammed A. S. Khalid[1] and Jonathan Rose

Department of Electrical and Computer Engineering
University of Toronto, Toronto, Ontario
CANADA M5S 3G4
{khalid, jayar}@eecg.toronto.edu

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 [Arno92] [Butt92] [Hauc94] [Apti96] [Vuil96] [Babb97] and previous research has shown that the partial crossbar is one of the best existing architectures [Kim96] [Khal97]. Recently, the Hybrid Complete-Graph Partial-Crossbar Architecture (HCGP) was proposed [Khal98], which was shown to be superior to the Partial Crossbar. In this paper we propose a new routing architecture, called the **H**ard**w**ired-**C**lusters **P**artial-Crossbar (HWCP) which is better suited for large MFSs implemented using multiple boards. The HWCP architecture is compared to the HCGP and Partial Crossbar and we show that it gives substantially better manufacturability. We compare the performance and cost of the HWCP, HCGP and Partial Crossbar architectures 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. We show that the HWCP architecture gives reasonably good cost and speed compared to the HCGP and Partial Crossbar architectures. Using our experimental approach, we also explore two key architecture parameters associated with the HWCP architecture to determine their best values.

1 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 [Brow92]. Multi-FPGA systems (MFSs), which are collections of FPGAs and memory joined 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 uses are for logic emulation [Apti96] [Quic96] [Babb97], rapid prototyping [Gall94] [Lewi97] and reconfigurable custom computing machines [Arno92] [Cass93] [Lewi97] [Vuil96] [Lewi98].

[1] Currently with Quickturn Design Systems, 55 West Trimble Road, San Jose, CA 95131-1013,USA

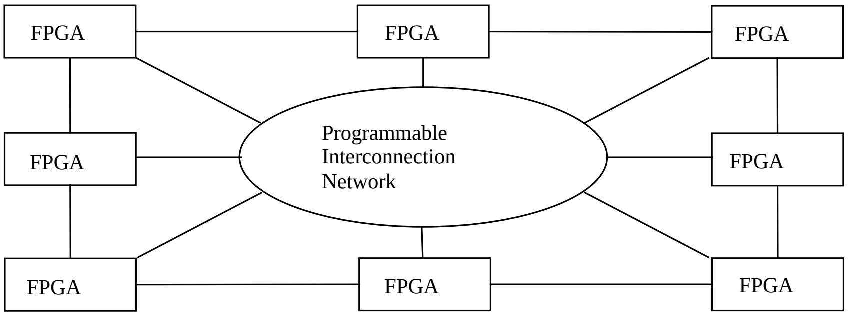

Figure 1 - A Generic Multi-FPGA System

The routing architecture of an MFS is the way in which the FPGAs, fixed wires, and programmable interconnect chips are connected. The routing architecture has a strong effect on the speed, cost and routability of the system. Many architectures have been proposed and built [FCCM] [Butt92] [Van92] [Apti96] [Babb97] [Lewi98] and some research work has been done to empirically evaluate and compare different architectures [Kim96] [Khal97]. These studies have shown that the partial crossbar is one of the best existing MFS architectures. Recently we proposed a new routing architecture for MFSs, called the HCGP [Khal98], that uses both hardwired and programmable connections to reduce cost and increase speed. We evaluated and compared the HCGP architecture and the partial crossbar architectures and showed that the HCGP architecture gives superior cost and speed.

The HCGP architecture gives excellent routability and speed for single board MFSs. The disadvantage of the HCGP, however, is that the wiring is very non-local and this becomes a major manufacturing problem when the FPGAs are spread across many boards - the number of connections required between the boards requires expensive, high pin-count connectors on each board. In this paper, we propose a new routing architecture called the Hardwired-Clusters Partial-Crossbar (HWCP), that lends itself to hierarchical implementations of large MFSs using FPGAs distributed across multiple boards. We show that the HWCP gives better manufacturability and reasonably good cost and speed compared to the HCGP and partial crossbar architectures.

This paper is organized as follows: In Section 2 we describe the partial crossbar, HCGP, and HWCP architectures. In Section 3 we define the metric used for measuring the manufacturability of the three architectures and compare them using this metric. To evaluate and compare the cost and speed of the three architectures we used an experimental approach which is described in Section 4. Real benchmark circuits were mapped to the three architectures using a customized set of partitioning, placement and inter-chip routing tools. The general CAD flow used is described in Section 4.1, the cost and speed evaluation metrics are defined in Section 4.2 and the benchmark circuits used are described in Sections 4.3. Experimental results and their analysis is presented in Section 5, and we conclude in Section 6.

2 Routing Architecture Description

In this Section we describe the partial crossbar, the HCGP and the HWCP architectures.

2.1 The Partial Crossbar Architecture

The partial crossbar architecture [Butt92] [Varg93] is used in logic emulators produced by Quickturn Design Systems [Quic96]. 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 number 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 (50) 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. For the above example, a P_t value of 12 will require 16 48-pin FPIDs. When we consider FPID pins required for circuit I/Os we will need to use 64 or 96-pin FPIDs that are commercially available [ICub97]. When choosing a value of P_t, we must ensure that number of usable I/Os per FPGA is evenly divisible by P_t or at least the remainder should be a very small number so that we can use such pins for routing high fanout inter-FPGA nets. In this work we set P_t = 17 which leaves five pins per FPGA to be used as global lines in the partial crossbar architecture. These global lines are used for routing global nets like *reset*, *clock* and other very high fanout nets in the circuit. Our previous research [Khal97] has shown that, for real circuits, the routability and speed of the partial crossbar is not affected by the value of P_t used. But this is contingent upon using an intelligent inter-chip router that understands the architecture and routes each inter-FPGA net using only

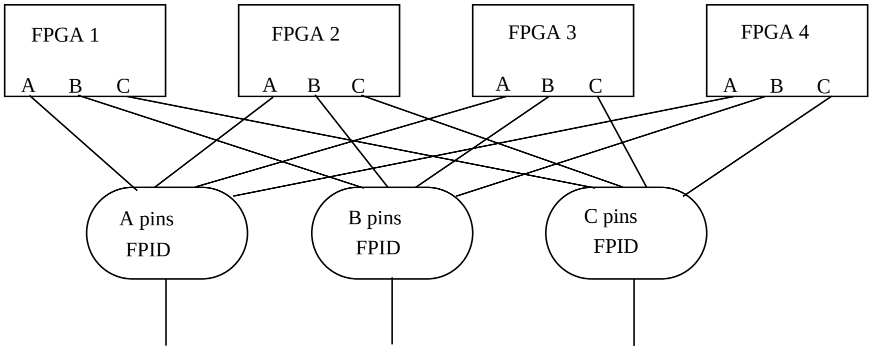

Figure 2 - The Partial Crossbar Architecture

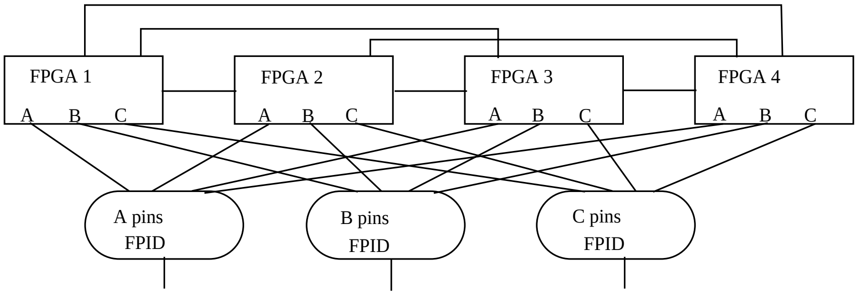

Figure 3 - The HCGP Architecture

two hops to minimize the routing delay. However, a practical constraint is that we should avoid using P_t values that require expensive or even unavailable high pin count FPIDs.

2.2 The HCGP Architecture

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 every other FPGA. The connections between FPGAs are evenly distributed, i.e. the number of wires between every pair of FPGAs is the same. 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 (50) are reserved for circuit I/Os.

The direct connections between FPGAs can be exploited to obtain reduced cost and better speed. For example, consider a net that connects FPGA 1 to FPGA 3 in Figure 3. If there were no direct connections as in the partial crossbar, we would have used an FPID to connect the two FPGAs. This will cost extra delay and two extra FPID pins.

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). If P_p is too high it

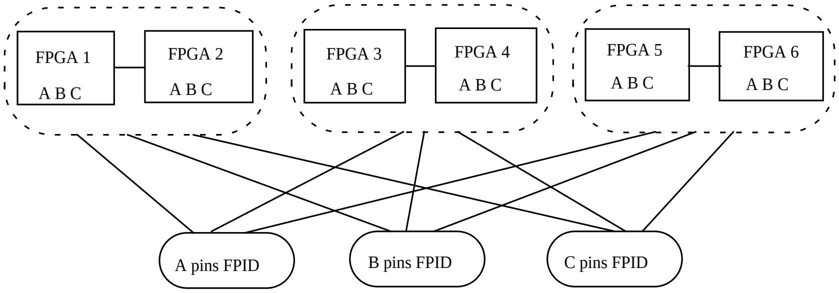

Figure 4 - The HWCP Architecture

will lead to increased pin cost, 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. Our previous research [Khal98] has shown that P_p = 60% gives good routability at the minimum possible pin cost for the HCGP architecture. Therefore we use this value of P_p when comparing the HCGP to the other architectures.

2.3 The HWCP Architecture

The motivation behind this architecture is to combine the routability and speed of the HCGP with easier manufacturability. All the connections in the HCGP architecture are non-local which leads to excessive board-level wiring complexity for single-board systems and requires expensive high pin count connectors when the FPGAs are spread out across multiple boards. Providing more local connections would mitigate this problem.

An example of the HWCP architecture using six FPGAs and three FPIDs is illustrated in Figure 4. The FPGAs are grouped into clusters whose size is represented by a parameter called the cluster size (C_s). In Figure 4 C_s = 2, which implies three clusters. The pins in each FPGA are divided into two groups: hardwired and programmable connections. The pins in the first group connect to the other FPGAs within each cluster and the pins in the second group connect to FPIDs. All the FPGAs within each cluster are connected to each other in a complete graph topology. In Figure 4 the pins assigned for programmable connections are divided into three subsets A, B, and C. The pins from different FPGAs belonging to the same subset are connected using a single FPID. As in the HCGP architecture, the percentage of programmable connections (P_p) is a key parameter in HWCP.

Notice that the MFS size in the HWCP architecture is restricted to be a multiple of C_s. In this research the HWCP architecture was explored for C_s values 2, 3, and 4.

3 Comparison of the Manufacturability of the Three Architectures

In this Section we propose a metric for evaluating and comparing the manufacturability of MFSs and compare the architectures using this metric.

3.1 Manufacturability Metric: Connector Pin Count

When implementing an MFS on a printed circuit board, proper design practices must be observed to ensure correct operation of the MFS. The main problems that need to be addressed are interconnect delays, crosstalk, ringing and signal reflection, and ground bounce. These problems become more acute as the system speed increases. Since MFSs usually run at low to medium speeds, these problems can be easily tackled by using proper design practices.

There is one major problem encountered when implementing an MFS using multiple boards connected by a backplane. Each board may contain one or more FPGAs and FPIDs, e.g. each board in the TM-2 system has 2 FPGAs and 4 FPIDs [Lewi98]. If the number of pins needed for connecting each board to the backplane is very high, expensive high pin count connectors are required. This problem is severe in MFS architectures with non-local wiring like the partial crossbar and HCGP. In fact, it is difficult to obtain (commercially) a pin connector that uses more than 800 pins [Iers97]. Therefore we use the number of pins required on the backplane connector on each printed circuit board as a metric for measuring the manufacturability of an MFS routing architecture (when the FPGAs and FPIDs are distributed across multiple boards). We call this the *Connector Pin Count* (CPC for short). The lower the CPC value for a given size and type of architecture, the easier it will be to manufacture. Given the MFS size, values of the architecture parameters, the number of boards used, and the number of FPGAs and FPIDs used per board, it is fairly easy to calculate the CPC value.

MFS Size (#FPGAs)	Connector Pin Count (CPC)		
	HWCP	HCGP	PXB
8	61	106 (+74%)	96 (+57%)
12	81	138 (+70%)	128 (+58%)
16	89	149 (+67%)	144 (+62%)
20	97	160 (+65%)	156 (+61%)
24	101	167 (+65%)	160 (+58%)
28	101	170 (+68%)	168 (+66%)
32	105	174 (+66%)	168 (+60%)
Average	**91**	**152 (+68%)**	**146 (+60%)**

Table 1: CPC Values for the Three Architectures for Different MFS Sizes

3.2 Manufacturability Comparison

Here we compare the manufacturability of the three architectures by calculating the CPC values for different MFS sizes ranging from 8 FPGAs to 32 FPGAs. To calculate CPC, we assume that the number of FPIDs used is equal to the number of FPGAs for all MFS sizes. We also assume that each board contains 4 FPGAs and 4 FPIDs. It is assumed that the FPGA used is the Xilinx 4013E-1(PG223) which has 192 usable I/O pins. We assume P_p = 60% for HCGP and HWCP, C_s = 4 for HWCP, and P_t = 17 for the partial crossbar. These parameter values were chosen after careful experimentation (further details are given in [Khal98] and Section 5.1).

Table 1 shows the CPC values obtained for the three architectures for different MFS sizes. Column 1 shows the MFS size (number of FPGAs used), column two shows the CPC value obtained for the HWCP, the HCGP and the partial crossbar architectures. Note that the CPC value for the HCGP architecture is 68% more on average compared to the HWCP architecture across different MFS sizes considered. The CPC value for the partial crossbar architecture is 60% more on average compared to the HWCP. The clear conclusion is that the HWCP architecture is easier to manufacture compared to the HCGP and partial crossbar architectures.

In addition to providing better manufacturability, an MFS routing architecture should give minimum possible cost and provide good routability and speed performance.

In the next Section we present the method used for experimentally evaluating and comparing MFS routing architectures. Although this method is targeted to only single-board systems, it still gives a good estimate of the cost, routability and speed of MFS routing architectures.

4 Experimental Overview

To evaluate the three routing architectures considered in this paper, we used the experimental procedure illustrated in Figure 5. Each benchmark circuit was partitioned, placed and routed into each architecture. Section 4.1 describes the general toolset used in this flow. The cost and speed metrics that we use to evaluate architectures are described in Section 4.2. A brief description of the 15 benchmark circuits used is given in Section 4.3.

4.1 General CAD Flow

The experimental procedure for mapping a circuit to an architecture is illustrated in Figure 5. We assume that the circuit is available as a technology mapped netlist of 4-LUTs and flip flops. First, the circuit is partitioned into a minimum number of sub-circuits using a multi-way partitioning tool that accepts as constraints the specific FPGA logic capacity and pin count. The FPGA used in our experiments is the Xilinx 4013E-1, which consists of 1152 4-LUTs and flip flops and 192 I/O pins. Multi-way partitioning is accomplished using a recursive bipartitioning procedure. The partitioning tool used is called *part*, which is based on the Fiduccia and Mattheyses partitioning algorithm [Fidu82] with an extension for timing-driven pre-clustering [Shih92]. The output of the partitioning step is a netlist of connections between the sub-circuits.

The next step is the placement of each sub-circuit on a specific FPGA in the MFS. Given the sub-circuits and the netlist of interconnections, each sub-circuit is assigned to a specific FPGA in the MFS. The goal is to place highly connected sub-circuits into adjacent FPGAs (if the architecture has some notion of adjacency) so that the routing

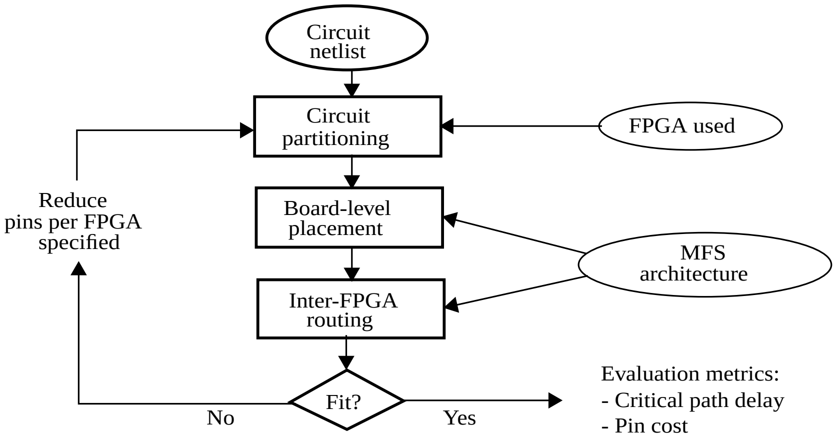

Figure 5 - Experimental Evaluation Procedure for

resources needed for inter-FPGA connections are minimized.

Given the chip-level interconnection netlist, the next step is to route each inter-FPGA net using the most suitable routing path. The routing path chosen should be the shortest path (use the minimum number of hops) and it should cause the least possible congestion for subsequent nets to be routed. Depending on the architecture, the routing resources available in an MFS could be wires that are direct connections between FPGAs, or wires that connect FPGAs and FPIDs.

If the routing attempt fails, the partitioning step is repeated after reducing the number of I/O pins per FPGA specified to the partitioner. This usually increases the number of FPGAs needed, and helps routability by decreasing the pin demand from each FPGA, and providing more “route-through” pins in the FPGAs which facilitate routing.

Note that in an actual MFS, the inter-FPGA routing step is followed by pin assignment, placement and routing within individual FPGAs. We need not perform these tasks because we are only interested in knowing the MFS size needed to fit the circuit. Our previous research has shown that we can afford to assign pins randomly for each FPGA without jeopardizing routability and speed [Khal95]. During recursive bi-partitioning, we restrict the logic utilization of each FPGA to be at most 70% to avoid placement and routability problems within individual FPGAs. Thus we ensure that if an inter-FPGA routing attempt succeeds, it is almost guaranteed that the subsequent pin assignment, placement, and routing steps will be successful for each FPGA in the MFS.

We have developed a specific router for each of the architectures compared. (We had attempted to create a generic router but found that it had major problems with different aspects of each architecture [Khal98a].)

4.2 Evaluation Metrics

To compare the three routing architectures we implement benchmark circuits on each and contrast the pin cost and post-routing critical path delay, as described below.

4.2.1 Pin Cost

The cost of an MFS is likely a direct function of the number of FPGAs and FPIDs: If the routing architecture is inefficient, it will require more FPGAs and FPIDs to implement the same amount of logic as a more efficient MFS. While it is difficult to calculate the price of specific FPIDs and FPGAs, we assume that the total cost is proportional to the total number of pins on all of these devices. Since the exact number of FPGAs and FPIDs varies for each circuit implementation (in our procedure above, we allow the MFS to grow until routing is successful), we calculate, for each architecture, the total number of pins required to implement each circuit. We refer to this as the *pin cost* metric for the architecture.

4.2.2 Post-Routing Critical Path Delay

The speed of an MFS, for a given circuit, is determined by the critical path delay obtained after a circuit has been placed and routed at the inter-chip level. We call this the *post-routing critical path delay.* We have developed an MFS static timing analysis tool (MTA) for calculating the post routing critical path delay for a given circuit and MFS architecture. The operation and modeling used in the MTA are described in [Khal98] hence we do not discuss it here for the sake of brevity.

4.3 Benchmark Circuits

A total of fifteen large benchmark circuits were used in our experimental work. An extensive effort was expended to collect this suite of large benchmark circuits which ranged in size from 2000 to 6000 4-LUTs and flip flops. A detailed description of the benchmark circuits can be found in [Khal98].

5 Experimental Results

In this Section we determine the effect of varying the value of P_p and C_s on the routability and speed of the HWCP architecture and compare the HWCP, the partial crossbar and HCGP architectures.

5.1 HWCP Architecture: Analysis of P_p and C_s

The cluster size C_s and the percentage of programmable connections P_p, are important parameters in the HWCP architecture. In this section we explore the effect of P_p on the routability of the HWCP architecture for three values of C_s (2, 3, and 4). Our objective is to determine suitable values of C_s and P_p that give good routability at the minimum possible pin cost.

We mapped the 15 benchmark circuits to the HWCP architecture for P_p values 20, 30, 40, 50 and 60 for each of the following cases: (a) $C_s = 2$, (b) $C_s = 3$ and (c) $C_s = 4$. The best routability results for the HWCP architecture (with the lowest possible P_p value to minimize the pin cost) are obtained when $C_s = 4$ and $P_p = 60\%$. The percentage of circuits that routed for $P_p \leq 60\%$ were 20% for $C_s = 2$, 60% for $C_s = 3$ and 80% for $C_s = 4$.

The clear conclusion from these results is that $C_s = 4$ and $P_p = 60\%$ are suitable choices for achieving good routability at the minimum possible pin cost in the HWCP architecture. Therefore we will use these parameter values when comparing the HWCP to the other two architectures.

Circuit	Number of FPGAs			Normalized pin cost			Normalized post-routing critical path delay		
	HWCP	Partial crossbar	HCGP	HWCP	Partial crossbar	HCGP	HWCP	Partial crossbar	HCGP
s35932	8	8	8	1.0	1.25	1.0	1.08	1.08	1.0
s38417	12	9	9	1.33	1.25	1.0	1.18	1.00	1.0
s38584	12	9	9	1.33	1.25	1.0	1.20	1.42	1.0
mips64	16	14	15	1.07	1.16	1.0	0.99	1.11	1.0
spla	20	18	18	routing failure	1.25	1.0	routing failure	1.16	1.0
cspla	20	18	18	routing failure	1.25	1.0	routing failure	1.18	1.0
mac64	8	6	6	1.33	1.25	1.0	1.15	1.34	1.0

Table 2: Comparison of HWCP, 1HCGP and Partial Crossbar Architectures

Circuit	Number of FPGAs			Normalized pin cost			Normalized post-routing critical path delay		
	HWCP	Partial crossbar	HCGP	HWCP	Partial crossbar	HCGP	HWCP	Partial crossbar	HCGP
sort8	16	12	14	1.14	1.07	1.0	1.15	1.07	1.0
fir16	12	10	10	1.20	1.25	1.0	1.19	1.43	1.0
gra	4	4	4	1.0	1.25	1.0	1.0	1.23	1.0
fpsdes	12	9	9	1.33	1.25	1.0	1.29	1.29	1.0
spsdes	8	8	8	1.0	1.25	1.0	1.15	1.21	1.0
ochip64	8	8	8	1.0	1.25	1.0	1.26	1.26	1.0
ralu32	16	9	14	routing failure	0.80	1.0	routing failure	1.21	1.0
iir16	8	6	6	1.33	1.25	1.0	1.11	1.05	1.0
Average	**12**	**10**	**10**	**1.17**	**1.20**	**1.0**	**1.15**	**1.20**	**1.0**

Table 2: Comparison of HWCP, 1HCGP and Partial Crossbar Architectures

5.2 Pin Cost and Speed Comparison of the Three Architectures

The 15 benchmark circuits were mapped to the HWCP, the HCGP and the partial crossbar architectures using the experimental procedure described in Section 4.1. The results are shown in Table 2, in which the first column shows the circuit name. The second column shows the number of FPGAs needed for implementing the circuit on each architecture (recall that we increase the MFS size until routing is successful). The third column shows the pin cost normalized to the number of pins used by the HCGP architecture and the fourth column shows the normalized critical path delay obtained for each architecture.

Inspecting Table 2, we can make several observations. First, the partial crossbar needs 20% more pins on average, and as much as 25% more pins compared to the HCGP architecture. Clearly, the HCGP architecture is superior to the partial crossbar architecture in terms of the pin cost metric. This is because the HCGP exploits direct connections between FPGAs to save FPID pins that would have been needed to route certain nets in the partial crossbar.

Table 2 also shows that the typical circuit delay is lower with the HCGP architecture: the HCGP gives significantly less delay for twelve circuits compared to the partial crossbar and about the same delay for the rest of the circuits. The reason is that the HCGP utilizes fast and direct connections between FPGAs, whenever possible. Another interesting observation is that even for the circuits where the HCGP needs more FPGAs compared to the partial crossbar, it still gives comparable or better delay value. This clearly demonstrates that the HCGP architecture is inherently faster due to the faster hardwired connections. It gives a significant speed advantage, especially when we use timing driven inter-FPGA routing.

Table 2 shows that the HWCP architecture failed to route three of the fifteen benchmark circuits. For the twelve circuits that routed on the HWCP, the pin cost is 17% more on average, and 33% in the worst case more compared to the HCGP architecture. The increase in pin cost is partly due to the fact that the HWCP required more FPGAs for some circuits to make the MFS size (measured by the number of FPGAs) a multiple of the cluster size (C_s = 4). The other reason is that the HCGP has superior routability compared to HWCP.

For the twelve circuits that routed on HWCP, the critical path delay is 15% more on average, and up to 29% more compared to the HCGP architecture. The reasons for this are: first the HWCP uses more FPGAs for some circuits, which implies more partitions and many more slow off-chip connections, some of which may lie on the critical path. Second, unlike the HCGP not all critical nets connecting pairs of FPGAs can be routed using hardwired connections in the HWCP.

Note that for the twelve circuits that routed, the average pin cost and speed results for the HWCP are comparable to the partial crossbar.

The results presented in this Section demonstrate that the HWCP architecture has the potential to provide good routability, speed, and manufacturability. Despite some routing failures, the routability results for HWCP are quite encouraging for the following reasons: first, we did not try higher cluster sizes (relative to the MFS size). The routability of the HWCP architecture improves with the increase in cluster size. Second, the partitioning and placement methods used for the HWCP architecture were not the best possible. Architecture-driven partitioning and placement methods may give better routability results.

6 Conclusions

In this paper we presented the Hardwired-Clusters Partial-Crossbar (HWCP), a new routing architecture suitable for large MFSs implemented using multiple boards. We show that the HWCP gives substantially better manufacturability compared to the HCGP and the partial crossbar architectures. Using an experimental approach, we evaluated and compared this architecture to the HCGP and the partial crossbar architectures and showed that it gives reasonably good pin cost and speed. To our knowledge, this is the first study of hierarchical architectures for board-level MFSs.

We explored key parameters (P_p and C_s) associated with the HWCP architecture and experimentally determined their best values (P_p = 60% and C_s = 4) for obtaining good routability for a variety of circuits.

This research is a first step in exploring routing architectures for large MFSs that use hundreds of FPGAs spread out across multiple boards. For a more rigorous investigation of such hierarchical architectures, we will need extremely large benchmark circuits and appropriate mapping CAD tools.

References

[Apti96] Aptix Corporation, Product brief: The System Explorer MP4, 1996. Available on Aptix Web site: http://www.aptix.com.

[Arno92] J. M. Arnold, D. A. Buell, and E. G. Davis, “Splash 2,” Proceedings of 4th Annual ACM Symposium on Parallel Algorithms and Architectures, pp. 316-322, 1992.

[Babb97] J. Babb et al, “Logic Emulation with Virtual Wires,” IEEE Trans. on CAD, vol. 16, no. 6, pp. 609-626, June 1997.

[Brow92] S. Brown, R. Francis, J. Rose, and Z. Vranesic, Field Programmable Gate Arrays, Kluwer Academic Publishers, 1992.

[Butt92] M. Butts, J. Batcheller, and J. Varghese, “An Efficient Logic Emulation System,” Proceedings of IEEE International Conference on Computer Design, pp. 138-141, 1992.

[Cass93] S. Casselman, “Virtual Computing and The Virtual Computer,” Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines, pp. 43-48, 1993.

[Chan93] P. K. Chan, M. D. F. Schlag, “Architectural Trade-offs in Field-Programmable-Device-Based Computing Systems,” Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines, pp. 152-161, 1993.

[Fidu82] C. M. Fiduccia, and R. M. Mattheyses, “A Linear-Time Heuristic for Improved Network Partitions”, Proc. of 19th ACM/IEEE Design Automation Conference, pp. 241-247, 1982.

[FCCM] Proceedings of IEEE Workshops/Symposia on FPGAs for Custom Computing Machines, 1992 to 1996.

[Gall94] D. Galloway, D. Karchmer, P. Chow, D. Lewis, and J. Rose, “The Transmogrifier: The University of Toronto Field-Programmable System”, CSRI Technical Report (CSRI-306), CSRI, University of Toronto, 1994.

[Hauc94] S. Hauck, G. Boriello, C. Ebeling, “Mesh Routing Topologies for Multi-FPGA Systems”, Proceedings of International Conference on Computer Design (ICCD’94), pp. 170-177, 1994.

[ICub97] I-Cube, Inc., The IQX Family Data Sheet, May 1997. Available at: www.icube.com.

[Khal95] M. A. S. Khalid and J. Rose, “The Effect of Fixed I/O Pin Positioning on The Routability and Speed of FPGAs,” Proceedings of The Third Canadian Workshop on Field-Programmable Devices (FPD’95), pp. 92-104, 1995.

[Khal97] M. A. S. Khalid and J. Rose, “Experimental Evaluation of Mesh and Partial Crossbar Routing Architectures for Multi-FPGA Systems,” Proceedings of the Sixth IFIP International Workshop on Logic and Architecture Synthesis (IWLAS’97), pp. 119-127, 1997.

[Khal98] M. A. S. Khalid and J. Rose, “A Hybrid Complete-Graph Partial-Crossbar Routing Architecture for Multi-FPGA Systems,” Proc. of 1998 Sixth ACM International Symposium on Field-Programmable Gate Arrays (FPGA’98), pp. 45-54, February 1998. This paper is available on the internet at “www.eecg.toronto.edu/~khalid/papers/fpga98.ps”.

[Kim96] C. Kim, H. Shin, “A Performance-Driven Logic Emulation System: FPGA Network Design and Performance-Driven Partitioning,” IEEE Trans. on CAD, vol. 15, no. 5, pp. 560-568, May 1996.

[Lewi97] D. M. Lewis, D. R. Galloway, M. Van Ierssel, J. Rose, and P. Chow, “The Transmogrifier-2: A 1 Million Gate Rapid Prototyping System,” Proceedings of FPGA’97, pp. 53-61, 1997.

[Lewi98] D. M. Lewis, D. R. Galloway, M. Van Ierssel, J. Rose, and P. Chow, “The Transmogrifier-2: A 1 Million Gate Rapid Prototyping System,” IEEE Trans. on VLSI Systems, vol. 6, no. 2, pp. 188-198, June 1998.

[Iers97] M. Van Ierssel, University of Toronto, Private Communication, 1997.

[Quic96] Quickturn Design Systems, Inc., System Realizer Data Sheet, 1996. Available on Quickturn Web site:http://www.quickturn.com.

[Shih92] M. Shih, E. S. Kuh, “Performance-Driven System Partitioning on Multi-Chip Modules,” Proc. of the Design Automation Conference, pp. 53-56, 1992.

[Varg93] J. Vargese, M. Butts, and Jon Batcheller, “An Efficient Logic Emulation System”, IEEE Trans. on VLSI Systems, vol. 1, no. 2, pp. 171-174, June 1993.

[Vuil96] J. E. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, and P. Boucard, “Programmable Active Memories: Reconfigurable Systems Come of Age,” IEEE Transactions on VLSI, Vol 4, No. 1, pp. 56-69, March 1996.

[Xili97] Xilinx, Inc., Product Specification: XC4000E and XC4000X Series FPGAs, Version 1.2, June 16, 1997. Available on Xilinx Web site: www.xilinx.com.