# Collaborative Routing Architecture for FPGA

Yaling Ma, *Student Member, IEEE,* and Mingjie Lin, *Student Member, IEEE*

***Abstract*— In this paper we present the *Collaborative Routing Architecture* (CRA), a routing architecture specially designed to achieve high efficiency in hardware and competitive delay performance for a FPGA. This is done by enabling *routing resource sharing* between different types: (1) Long interconnects can be constructed with short bypass interconnects without sacrificing delay performance. (2) Switch boxes and connection boxes both are embedded in the switching core of the routing modules. Therefore routing resources such as MUXs can be shared between them on a per-mapping basis. (3) The switching core in CRA can dynamically extend its switching capability, whereas in a conventional switch box, switch matrix is pre-determined and therefore static. These architectural features demonstrate significant performance improvements. Using the same logic placement, the CRA yields about 25% reduction in the minimum routing channel width, 20% improvement in overall delay performance for 20 largest MCNC benchmark circuits, when compared with a virtex-II style baseline FPGA.**

## I. Introduction

With the advent of sub-100nm CMOS technologies, the design and prototyping costs have become prohibitive for most ASICs, making FPGAs increasingly popular. Unfortunately, current FPGAs cannot meet the performance requirements of many applications due to their high programming overhead [1]. Previous studies such as [2] have found that programmable routing contributes about 60% of the total path delay in FPGAs and occupies as much as 90% of the FPGA area. Because routing architecture largely determines the overall performance of a FPGA and is critical to its logic density, designing a high-performance routing architecture for FPGA is the key to narrowing the performance gap between FPGAs and ASICs.

The routing architecture of almost all existing FPGAs, certainly the prevalent ones, are implemented as functionally separate blocks such as interconnects, switching boxes, and connection boxes [3] [4] (See Figure 1(a) for a schematic.). This approach has the advantage in design and implementation flexibility but makes routing resource sharing difficult, if not impossible. Unfortunately routing resource sharing can be critical to the overall performance of a FPGA in many cases. For example, some data-path intensive circuits may require large number of straight interconnects but relatively less number of switch points to make turns in the signal paths, therefore connection box and interconnects can be heavily used but switch boxes are significantly under-used. This inefficient use of routing resource can lead to larger minimum routing channel width required to successfully place/route a circuit.

In this paper, we propose a new routing architecture (See Figure 1(b) for a schematic.) that enables routing resource sharing and demonstrate through performance analysis that this new routing architecture can significantly improve the efficiency of routing resource usage and the overall delay performance.

The following section describes the proposed routing architecture in detail. In Section III, we study the performance benefits of the CRA over the Virtex-II style baseline and present the performance improvements in terms of routing resource efficiency and overall delay performance. Finally in Section IV, we summarize our main findings.

## II. Collaborative Routing Architecture

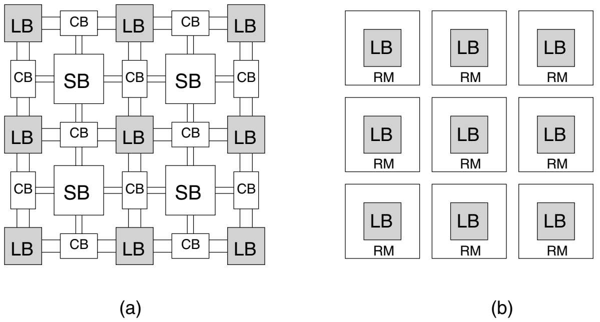


Fig. 1. Diagram of routing architecture. (a) Conventional island-style segmented routing architecture. (b) Proposed Collaborative Routing Architecture. (LB: Logic Block, SB: Switch Box, CB: connection Box, RM: Routing Module.)

The CRA consists of an array of routing modules (Fig. 2(a)), each of which contains a set of bypass interconnects along two perpendicular directions (Fig. 2(b)) and a switch core (Fig. 2(c)). The bypass interconnects serve as the fast signal paths connecting all logic blocks, while the switching core joins together different bypassing interconnects and is relatively slower compared with bypassing interconnects. Note long interconnects can be constructed by several short bypass interconnects without passing through switching core.

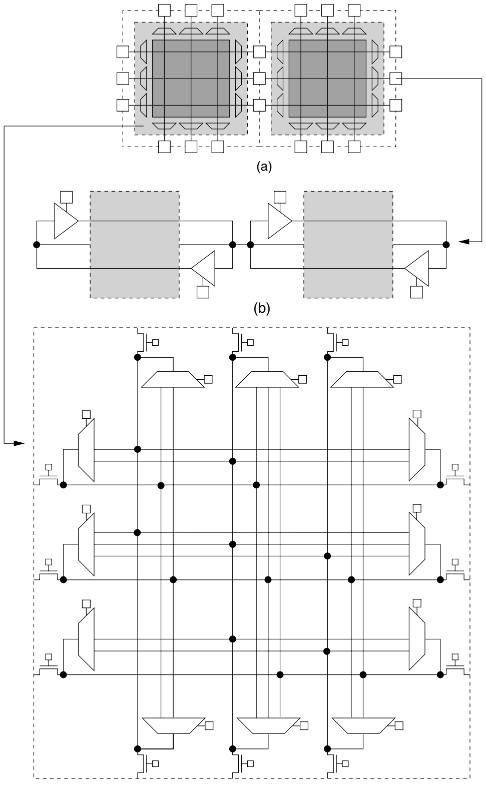
Each bypass interconnect has two uni-directional interconnects controlled by a properly sized tri-state buffer. Incoming signal for each routing module can either take bypass interconnects or enter the switching core to make signal turns. Because the CRA enables routing resource sharing and the routing software will always choose the signal path with the minimum possible delay cost, most routed signals will take fast bypass interconnects and enter switching core only when necessary. Outgoing signals from the routing module can take the bypass interconnects of the surrounding routing modules or directly connect to the input ports of the neighboring blocks.

The switching core is mainly composed of MUXs and its main function is to make signal turns, i.e., to switch signals entering from one side of the switching core to the other sides. Two parameters defines a switching core: the switching width

M. Lin is with the Electrical Engineering Department at Stanford University.

1-4244-0921-7/07 $25.00 © 2007 IEEE.

$W$ and the switching density $d$. $W$ denotes the numbers of possible interconnects connects to each side of the switching core and $d$ denotes the number of interconnects each incoming signal can be directly switched to. Figure 2(c) illustrates a switching core with $W = 3$ and $d = 3$. Each MUX in the switching core has four inputs and three configuration memory bits. Three inputs are for the signal switching and the fourth for an output port from the associated logic block. Five configuration patterns are needed. Four of them are used to select one of the inputs to the MUX and the fifth control pattern can generate a high-Z state for the output of the MUX. Typically, the number of the inputs to the MUXs needs to be $d + 1$.


(a)

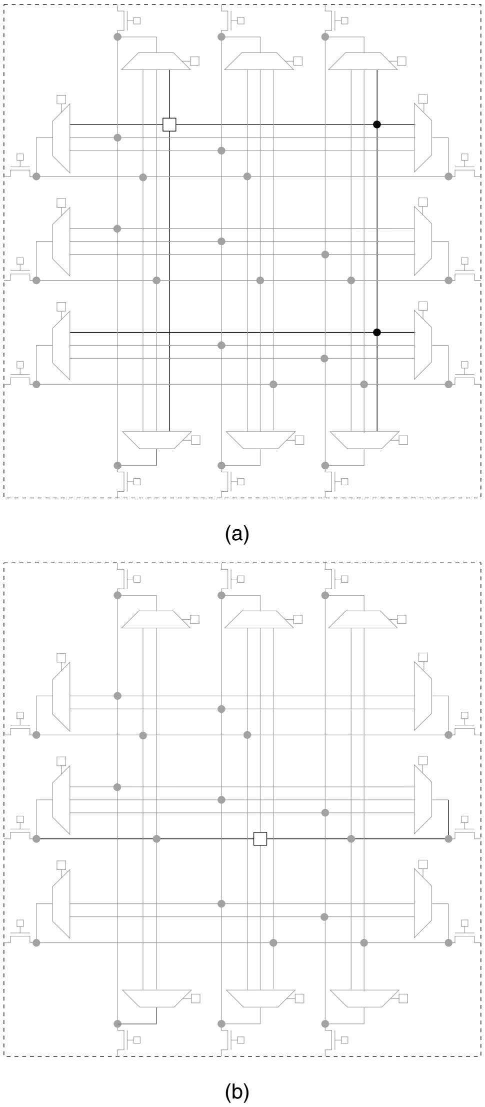
(b)

(c)

Fig. 2. (a) Logic schematic of a 3×3 routing module. (b) Bypassing interconnects. (c) Switching core.

*Embedded C-Box and S-Box*

In FPGA, connection boxes serve as signal interfaces between IO ports of logic blocks and interconnects, while switch box connects different interconnects and make signal turns. In CRA, switching core fulfills the functions of both switch and connection boxes but unlike an island-style segmented routing architecture, there is no separation between connection boxes and switch boxes. This design enables flexible routing resource sharing between switch box and connection box. Let $n_{\mathrm{i}}$ and $n_{\mathrm{o}}$ be the the number of the inputs and outputs in the associated logic block (16 and 4 in our study), each input port or output port can be connected to about $W/(n_{\mathrm{i}} + n_{\mathrm{o}})$ MUXs on each side of switching core. Figure 3(a) illustrates how an LB output connects to some bypass interconnects. Figure 3(b) illustrates how a bypass interconnect connects to an LB input. Figure 4(a) shows the switching capability of a $W = 3$ and $d = 3$ switching core.


(a)

(b)

Fig. 3. (a) Illustration of a signal path from an LB output to a bypass interconnect. (b) Illustration of a signal path from interconnects to an LB input.

*Dynamic Switching Matrix*

Switch box plays a central role in switching signals between different interconnects in an island-style FPGA. For almost all existing switch box design, the switch matrix defining possible switching patterns is pre-determined and therefore can not be changed during or after FPGA configuration phase, i.e., static. In the CRA, the switching core is functionally similar to the conventional switch box. However, instead of having static switching matrix, the switching core we propose can dynamically extend its possible switching patterns. In Section III, we will show that this dynamic extending feature can significantly reduce the minimum routing channel width needed for successfully routing benchmark circuits. Fig. 5(a) illustrates the expanded switching capability of a $3 \times 3$ switch

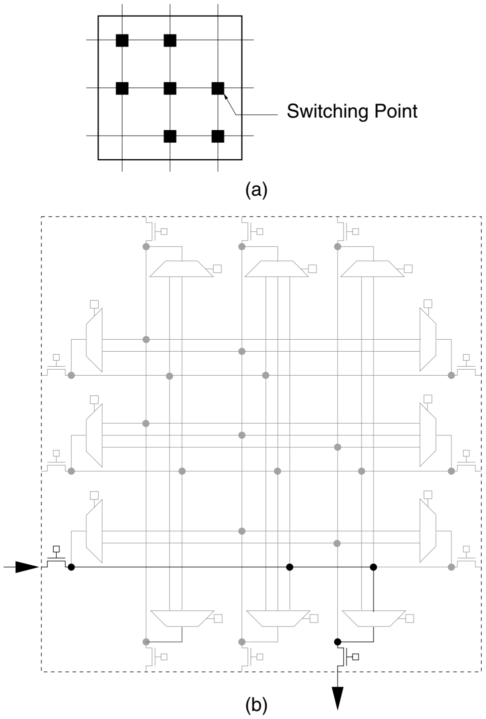


Fig. 4. (a) Switching capability of a routing block. (b) Example signal bend.

core with $W = 3$ and $d = 3$. In Fig. 5(b), we illustrate how to extend the switching capability of the same switching core simulate the more complete switch box shown. Note that the signal delay of switching core will increase if the passing signal takes the extended switching path.

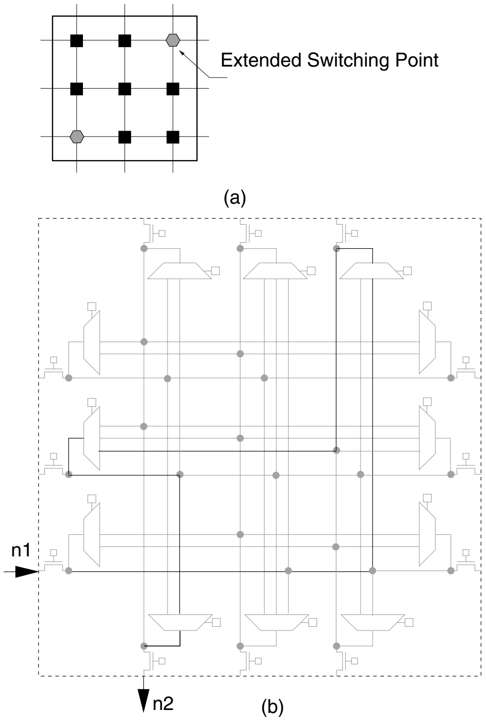


Fig. 5. Routing capability of switching core. (a) Direct routability. (b) Extended routability.

## III. Performance Results

We focus on two performance metrics: the minimum routing channel width ($W_{\min}$) that indicates the routing efficiency of a routing architecture and the system delay performance.

To make the performance comparison meaningful, we choose a Virtex-II style FPGA as our baseline (See [5] for more details.). For all delay simulations, we choose 65nm device and interconnect models from BPTM(Berkeley Predicative Technology Model). The transistor sizes used in the CRA are assumed to be: 6 for the tri-state buffers in bypass interconnects, 8 for the pass transistor switch in the switching core , and 6 for all MUXs. Another important aspect of the performance analysis is the CAD tools for the circuit mapping. For all three circuit mapping phases (technology mapping, placement, and routing), CAD software for the CRA have been largely rewritten based on VPR [6] by University of Toronto. Main modifications are made in the routing graph generation and the routing algorithms. Our baseline for delay comparison has a Virtex-II style routing architecture. All parameters related to the architecture is defined with details in [5]. We used the same methodology as in [5] to size all buffers. Our methodology for comparing delay between our baseline and the CRA is similar to the one in [5].

Using the Cadence GSCLib3.0 technology-independent library and Virtuoso tool we estimate the layout area for the logic block and buffers in the same manner as [5]. The area of routing module with $W = 72$ and $d = 3$ is estimated using custom layout. The area of an LB ($2256\lambda \times 2256\lambda$) is the same in both baseline FPGA and the proposed CRA. As in [5], The tile size for the baseline FPGA and for the CRA are estimated to be $4100\lambda \times 4100\lambda$ and $3955\lambda \times 3955\lambda$ respectively.

TABLE I
Routing resource of a CRA tile.

| | |
|---|---|
| Logic Block | Memory bits: 1049 |
| Bypass Interconnects | Tri-state buffers: 288<br>Memory Bits: 288 |
| Routing Module | Switches: 1440<br>Memory Bits: 1152 |
| Total | Switches: 1440<br>Tri-state buffers: 288<br>Memory Bits: 2489 |

The minimum routing channel width ($W_{\min}$) denotes the minimum width of routing channel required to successfully route a placed circuit. The smaller value of $W_{\min}$ means better routability of the considered routing architecture. In other words, for a fixed routing channel width, the routing architecture with smaller $W_{\min}$ normally can implement larger circuit designs. The first section of the Table II lists $W_{\min}$ for six MCNC benchmark circuits for a Virtex-II style baseline and a CRA. On average, the CRA can achieve about 30% reduction in $W_{\min}$.

One strength of CRA is its capability to share routing resource. Ideally, after placement, we want to route each signal with the fewest number of interconnects and the fewest number of switching points. One can imagine, should we have

TABLE II
ROUTING RESOURCE EFFICIENCY FOR VIRTEX-II STYLE ROUTING ARCHITECTURE AND THE CRA.

| Circuit | $W_{min}$ | | $\overline{L}$ | | $\overline{S}$ | |
|---|---|---|---|---|---|---|
| | Baseline | CRA | Baseline | CRA | Baseline | CRA |
| alu4 | 22 | 16 | 13.31 | 11.98 | 3.83 | 4.01 |
| apex2 | 29 | 20 | 12.43 | 10.76 | 3.18 | 3.56 |
| apex4 | 30 | 22 | 10.06 | 9.66 | 2.69 | 2.81 |
| ex1010 | 29 | 21 | 15.36 | 14.57 | 3.63 | 4.04 |
| misex3 | 25 | 17 | 11.94 | 10.35 | 3.11 | 3.45 |
| tseng | 16 | 12 | 10.62 | 9.89 | 3.04 | 3.37 |

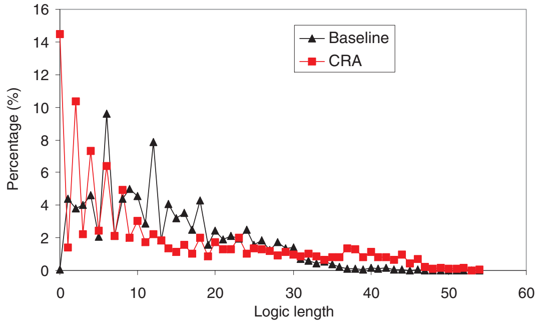


Fig. 6. Distribution of logic lengths for the ALU4 benchmark circuit mapped onto the baseline FPGA and the FPGA with the CRA.

unlimited routing resource ($W \rightarrow \infty$), each routed signal net would form a minimum Steiner tree and each source-sink pair will only pass one or zero switching point. Fig. 6 illustrate logic length distribution for a circuit example $ALU4$. There are two curves that plots the logic length distribution of all pin-to-pin signal path in Fig. 6. The first curve plots the baseline case with segmented interconnects and the second curve the CRA case. As one can see, the CRA's distribution is far more concentrated on the left side than that of the baseline, which means the majority of the pin-to-pin signal paths in a design routed on the CRA is very short without detouring.

Table II lists the average number of the logic length ($\overline{L}$) and the average number of the bends ($\overline{S}$) for all pin-to-pin signal paths for six typical benchmark circuits. Overall all, there are about 15% reduction in $\overline{L}$ and 5% increasing in $\overline{S}$ between the baseline and the CRA. The increasing of $\bar{S}$ may due to the fact that the CRA only has short bypass interconnects and longer interconnects have to be constructed with multiple bypass interconnects.

Several components of a FPGA contributes to the signal path delay. The main ones are logic block, interconnects, and switch box. Previous work has shown that signal delay through switching points in FPGA is significant. In this study, we assume the same logic block design and therefore the same logic block delay.

To compare the system performance of the proposed FPGA to that of the baseline, we use two metrics; the improvement in the geometric average of the pin-to-pin delays, and the improvement in critical path delay, which includes the LB delays along the path. By improvement here we mean the ratio of the delay in the baseline FPGA to that in a FPGA with the CRA. Results for the largest 20 MCNC benchmark circuits are plotted in Figures 7. Note that the improvements range between 1.13x and 1.37x for the geometric average and between 1.08x and 1.23x for the critical path delay. On average, there is a 20% delay improvement.

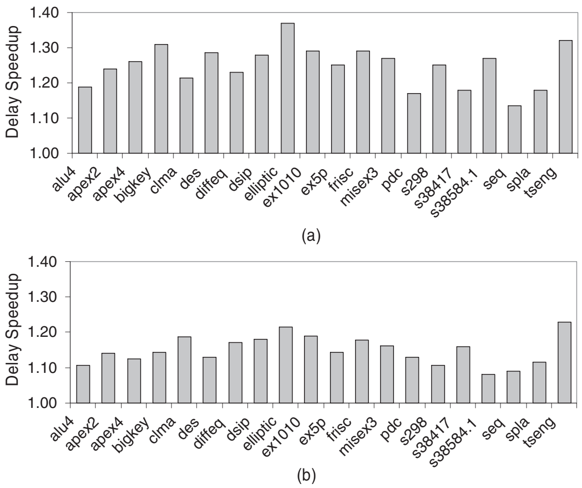


Fig. 7. Delay improvements of the CRA FPGA over the baseline FPGA for MCNC benchmark circuits. (a) The geometric average pin-to-pin delay. (b) The critical path delay. (All implemented in 65nm technology.).

## IV. CONCLUSION

As the performance gap between FPGA and cell-based design continues to widen rapidly, the need to drastically improve the performance of FPGA becomes increasingly urgent. Contrary to the conventional FPGA routing architecture design with segmented interconnects and sparse switch box, the Collaborative Routing Architecture enables routing resource sharing that significantly improves routing resource utilization and delay performance. We also have shown that this proposed CRA routing architecture can be readily implemented using the existing planar CMOS technology. We believe that additional performance improvements can be obtained by further optimizing this proposed architecture.

## REFERENCES

[1] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in *Proceedings of the 2006 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays*, pp. 21 – 30, 2006.

[2] V. George, *Low Energy Field-Programmable Gate Array*. PhD thesis, UC Berkeley, 2000.

[3] Xilinx, "Virtex-e 1.8v field programmable gate arrys datasheet," No. 2.2, Nov. 2001.

[4] D. Lewis, E. Ahmed, G. Baeckler, V. Betz, M. Bourgeault, D. Cashman, D. Galloway, M. Hutton, C. Lane, A. Lee, P. Leventis, S. Marquardt, C. McClintock, K. Padalia, B. Pedersen, G. Powell, B. Ratchev, S. Reddy, J. Schleicher, K. Stevens, R. Yuan, R. Cliff, and J. Rose, "The stratix II logic and routing architecture," in *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, pp. 14 – 20, 2005.

[5] M. Lin, A. El Gamal, Y.-C. Lu, and S. Wong, "Performance benefits of monolithically stacked 3D-FPGA," in *Proceedings of the 2006 ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays*, pp. 113 – 122, 2006.

[6] V. Betz and J. Rose, "Directional bias and non-uniformity in FPGA global routing architectures," in *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pp. 652 – 659, 1997.