

Architecture of Datapath-Oriented Coarse-Grain Logic and Routing for FPGAs

Andy Ye, Jonathan Rose, David Lewis

Department of Electrical and Computer Engineering, University of Toronto
Toronto, Ontario, Canada M5S 3G4
{yeandy, jayar, lewis}@eecg.utoronto.ca

Abstract

In this paper, we propose a new datapath-oriented FPGA architecture that utilizes coarse-grain logic and routing resources to increase the area efficiency of datapath circuits. Using a set of custom-built datapath-oriented CAD tools and a set of datapath benchmarks, we investigated several variants of our proposed architecture. We found that the architecture achieves the highest area efficiency when 40% to 50% of the total routing tracks are coarse-grain. Furthermore, comparing to conventional FPGA architectures, our datapath-oriented architecture uses about 10% less area to implement the same circuits.

1. Introduction

The past decade has seen a dramatic increase in the logic capacity of FPGAs which has brought FPGAs to use in ever-larger applications. Large applications, whether they are CPUs, graphics processors, digital signal processors, or packet switching networks, typically contain a greater amount of datapath logic, which is highly regular in structure. The efficient implementation of these highly regular structures has become an increasingly important issue to the overall area and performance of many FPGA applications.

Previous research [7][8][9][10][11][12][14][15][16] has shown that regularity-driven synthesis, placement and routing can be used to improve the density and speed of datapath circuits. The work of [6] also shows that further area savings can be achieved by incorporating datapath-specific features into regular FPGA architectures. One particularly compelling architectural feature is the coarse-grain routing resources suggested by [6]. Their basic notion was to amortize configuration bit area across multiple wires when these wires are data buses. In this paper we perform a detailed exploration of the area advantage of a number of variants of this architectural structure.

By way of a more complete introduction to the architectural concept of coarse-grain routing structures, we first review more traditional FPGA routing. Fig. 1 illustrates a typical FPGA, in which logic is implemented in logic blocks [1][3][4][5] that consist of tightly connected look-up tables (LUTs) such as the cluster shown in Fig. 2. Logic blocks are then connected together through programmable routing resources composed of input connection blocks, output connection blocks, switch blocks and routing tracks. These routing resources are made configurable by the programmable switches controlled by SRAM cells. In a typical FPGA, each switch is controlled by a unique set of configuration memory. We call these resources fine-grain routing resources.

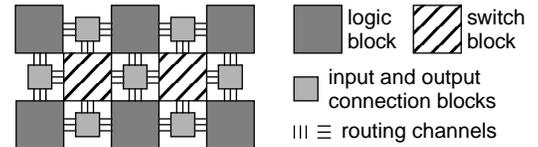


Figure 1: A Typical FPGA Architecture

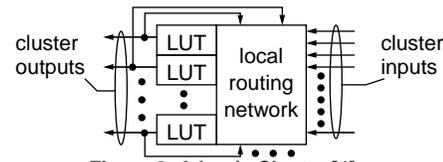


Figure 2: A Logic Cluster [4]

Coarse-grain routing tracks, on the other hand, are grouped together in groups of M , and switches associated with each group are collectively controlled by a single set of configuration memory bits. When routing datapath circuits, coarse-grain routing tracks can be more efficient at connecting a group of signals from a common source to a common destination and consequently achieve significant area savings.

Although coarse-grain tracks are more efficient at routing groups of signals that share a common source and a common destination, they are inefficient at routing individual signals. When a group of coarse-grain tracks are used to route a single signal, only one track in the group is utilized, wasting the other tracks. An efficient FPGA architecture for datapath circuits should, therefore, contain a mixture of fine and coarse-grain routing resources, as all application circuits are likely to contain both types of signals — signals that can be routed in groups and signals must be routed individually. In this paper, we investigate the question of how many coarse-grain routing tracks should be included in an FPGA targeting highly regular datapath circuits in order to achieve maximum area savings.

To the best of our knowledge, this important question has not been addressed by previous studies. In particular, the simple area model in [6] (which defines routing area as a linear function of the number of logic block I/Os) prevents such an in-depth study on routing architectures. In contrast, this paper uses a much more detailed area model based on [4]. Furthermore, our study also takes full account of the area inflation after datapath-oriented synthesis [18], which was ignored by many previous studies on datapath-oriented architectures, placement and routing tools.

The rest of this paper is divided into four sections. In Section 2, we give a complete description of a parameterized set of coarse-grain FPGA architectures that we explore. Section 3 describes the experimental methodology we use to explore the datapath archi-

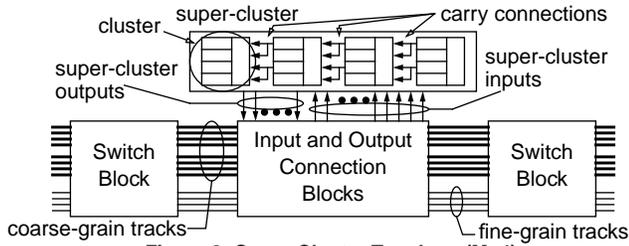


Figure 3: Super-Cluster Topology ($M=4$)

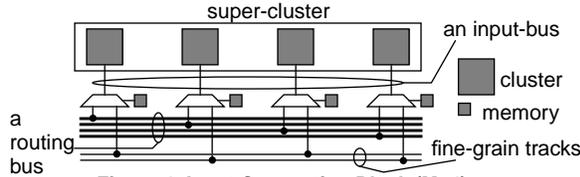


Figure 4: Input Connection Block ($M=4$)

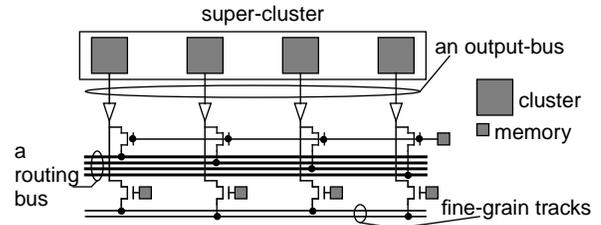


Figure 5: Output Connection Block ($M=4$)

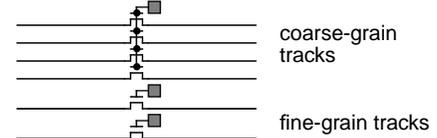


Figure 6: Switch Block ($M=4$)

ture. Section 4 presents experimental results on the percentage of coarse-grain tracks. We also compare the area efficiency of our proposed architecture with a conventional architecture. Section 5 presents concluding remarks.

2. A Coarse-Grain Datapath FPGA Architecture

In order to utilize the regularity of datapath circuits on coarse-grain routing resources, we need an FPGA architecture that can easily capture datapath regularity. Once captured, one should be able to easily map this regularity information onto coarse-grain routing resources. Since very few FPGAs are designed with datapath regularity in mind, we designed our own architecture (the datapath architecture) based on a conventional FPGA architecture (which we will call the “standard” architecture) described in [4]. In [4] a logic block is called a logic cluster. Each conventional cluster contains N four-input Look-Up Tables (LUTs). A cluster also contains a fully connected local routing network as shown in Fig. 2. In our architecture, a logic block is called a super-cluster which consists of M conventional clusters grouped together using the topology shown in Fig. 3. M is the granularity of our datapath FPGA architecture.

The super-cluster structure is motivated by the fact that datapath circuits often consist of many identical bit-slices and these bit-slices are the source of signal buses — regularly structured connections that map well onto the coarse-grain routing resources. Using our architecture, we implement portions of bit-slices in clusters. Then we group the clusters that implement identical portions of bit-slices together into super-clusters. By doing so, we can maximize the chance of capturing datapath buses onto inter-super-cluster connections without sacrificing the utilization of local routing networks in clusters. Once captured, these buses can then be efficiently routed through the coarse-grain routing resources in the global routing network.

The global routing resources of the datapath FPGA consist of both coarse-grain routing resources with a granularity value of M and conventional fine-grain routing resources. Each routing channel contains a fixed number of coarse-grain routing tracks and a fixed number of fine-grain routing tracks. Coarse-grain routing

tracks are grouped into M -bit wide buses. We call these buses routing-buses.

Within each super-cluster, special connections supporting arithmetic carry signals are provided. The number of super-cluster I/Os is equal to the total number of cluster I/Os in a given super-cluster; and each cluster I/O is directly connected to a super-cluster I/O. An input connection block is shown in Fig. 4. Each input pin can be connected to a fixed percentage, F_{c_if} , of fine-grain routing tracks. For each super-cluster, we group corresponding inputs of the M clusters together to form M -bit wide buses. We call these buses input-buses. Each input-bus is connected to a fixed percentage, F_{c_ic} , of routing-buses. An output connection block is shown in Fig. 5. Each output pin can be connected to a fixed percentage, F_{c_of} , of fine-grain routing tracks. As cluster inputs, we also group cluster outputs into M -bit wide buses. We call these buses output-buses. Each output-bus is connected to a fixed percentage, F_{c_oc} , of routing-buses.

As shown in Fig. 4 and Fig. 5, when connecting an input-bus/output-bus to a routing-bus, we connect the corresponding bits of each bus together. The programmable switches in each bus-to-bus connection of the output connection blocks share a single set of configuration memory.

As in conventional architectures, we assume all I/O pads are uniformly distributed on the boundary of our datapath FPGA. Each I/O pad is bi-directional — containing one input pin and one output pin. Both input pin and output pin have the same connection patterns to the routing tracks. Each pad pin can be connected to a fixed percentage, F_{c_pf} , of fine-grain routing tracks. M I/O pad input/output pins are grouped to form pad-input/output buses. Each bus is connected to a fixed percentage, F_{c_pc} , of routing buses. The pad-input/output bus to routing bus connections are similar to the cluster input-bus/output-bus to routing bus connections described above.

A switch block which resides at the intersection of all horizontal and vertical channels is shown in Fig. 6. It contains both fine-grain routing track connections and coarse to coarse-grain routing track connections. We assume that there are no connections between fine and coarse-grain routing tracks. We use the

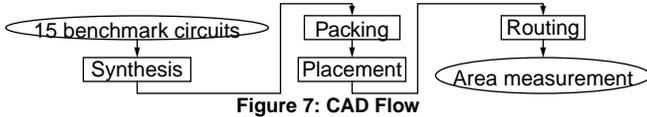


Figure 7: CAD Flow

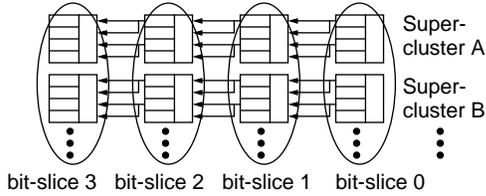


Figure 8: Packing (M=4)

disjoint topology [13] for both fine-grain connections and coarse-grain connections since this is one of the most efficient and widely used topology for conventional FPGAs. Each fine-grain routing track can be connected to Fs_f of other fine-grain tracks. Each coarse-grain routing-bus can be connected to Fs_c of other routing-buses. As shown in Fig. 6, when connecting two buses, we connect the corresponding bits of each bus together, and again there is sharing of configuration memory.

3. Experimental Methodology

We employ an experimental methodology to investigate the effect of varying the number of coarse-grain tracks on the area of the datapath architecture. We also compare the area efficiency of the datapath architecture against the standard architecture. Fig. 7 shows the CAD flow of our experiments. The 15 benchmark circuits are from the Pico-Java processor from SUN Microsystems [2]. The benchmark set covers all major datapath components of the processor. These circuits are synthesized into LUTs using a datapath-oriented synthesis process described in [18]. This synthesis process preserves the regularity of datapath circuits while attempting to minimize area.

The synthesized circuits are then packed into super-clusters using a new datapath-oriented packing tool that we have written based on the T-VPACK packing algorithm [4]. Our packing tool tries to pack every M adjacent bit-slices into a series of super-clusters. As shown in Fig. 8, portions of a bit-slice are mapped into a unique cluster for each super-cluster. The packer also utilizes the super-cluster level carry connections to minimize the delay of carry chains. The packed circuits are then placed using a placement algorithm modified from VPR [4]. The algorithm moves super-clusters as the basic unit if they contain grouped datapath slices. Otherwise, non-datapath clusters (that contain random logic) are optimized individually. The placed circuits are then routed using a datapath-oriented router, which is based on the VPR routing algorithm [4] and is modified to efficiently use coarse-grain routing resources. Using a set of specially designed cost functions, our router tries to balance the use of fine and coarse-grain routing resources based on congestion and timing constraints [17].

For all of our experiments, we set the granularity of the datapath architecture, M , to be four. This granularity was shown to be one of the most efficient by the study of [6]. It is also used by the

architecture described in [14]. As discussed in Section 2, the datapath architecture uses a disjoint switch block. As in many current commercial FPGAs, we set the Fs_f and Fs_c values to be three for all of our experiments. We also assume a fully buffered global routing architecture — all switches in our switch blocks are buffered switches.

To find the effect of varying the number of coarse-grain tracks on the area of the datapath architecture, we performed routing using several variants of the datapath architecture, each with a different number of coarse-grain tracks. For each circuit, we fix the total number of coarse-grain tracks that can be used and let the router search for the minimum number of fine-grain tracks that is needed to complete the routing. The number of fixed coarse-grain routing buses that we considered for each benchmark circuit is from 0 to 20 inclusively.

We define the track length, or the logical track length, to be the number of logic clusters that a routing track passes before being interrupted by a switch. For all of our experiments, we fixed the logical track length to be two for both coarse-grain and fine-grain tracks. We also fixed the cluster size, N , to be four. Both the track length of two and the cluster size of four were found to generate the best area results in [17]. To determine the area efficient values for Fc_{if} , Fc_{pf} , Fc_{of} , Fc_{ic} , Fc_{pc} , and Fc_{oc} , we set the number of coarse-grain tracks to be zero. We varied the design parameters Fc_{if} , Fc_{pf} , and Fc_{of} to find a combination of these three parameters that is the most area efficient. We then assume the same set of Fc_{if} , Fc_{pf} , and Fc_{of} will generate the most area efficient results for any percentage of coarse-grain tracks, when Fc_{ic} , Fc_{pc} , and Fc_{oc} are set to be equal to Fc_{if} , Fc_{pf} , and Fc_{of} , respectively.

To compare the area efficiency of a standard architecture with our datapath-oriented FPGA architecture, we also set the cluster size, N , to be four for the standard architecture. We again use a fully buffered global routing architecture. We varied several design parameters including L (logical track length), Fc_{input} (number of tracks that a cluster input connect to), Fc_{pad} (number of tracks that a pad I/O pin connect to), and Fc_{output} (number of tracks that a cluster output connect to) to find a set of design parameters that generate the best area result for the standard architecture. We also use the best available synthesis tool for the standard architecture instead of the regularity preserving datapath synthesis [18].

4. Experimental Results

Fig. 9 shows the total area vs. the percentage of total tracks that are coarse-grain in the datapath FPGA routing. We measured the area in terms of the number of equivalent minimum-width transistor area as described in [4]. For each benchmark circuit, we collected the area results from datapath architectures as described in Section 3. We then classify these architectures into eight groups based on the percentage of total tracks that are coarse-grain. The percentile ranges are (0%, 0%], (0%, 10%], (10%, 20%], (20%, 30%], (30%, 40%], (40%, 50%], (50%, 60%], and

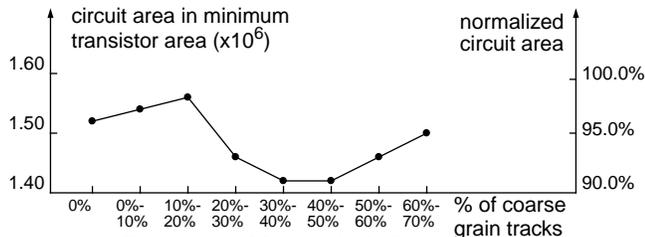


Figure 9: % of Coarse-Grain Tracks vs. Area

(60%, 70%]. Within each region, we first obtain the minimum area obtainable by each circuit. We then average these minimum area values across 15 benchmark circuits. The arithmetic average of the area values is then plotted against each percentile range.

Fig. 9 shows that as we start to add coarse-grain tracks to our routing fabric, we are differentiating our routing resources into two types. This differentiation reduces the routing flexibility and accounts for the initial increase in total area. As the number of coarse-grain tracks is increased to the 20% range, the benefit of coarse-grain tracks starts to outweigh the inflexibility in routing. As the result, the total area required decreases until it reaches the minimum when coarse-grain tracks account for between 40% to 50% of the total tracks. When we further increase the number of coarse-grain tracks, the number of coarse-grain tracks provided by the architecture starts to exceed the number of coarse-grain tracks required by the circuits. The router then starts to excessively use coarse-grain tracks for fine-grain routing. This reduces the efficiency of the datapath architecture past the 50% point.

Overall, the best area is achieved when coarse-grain tracks account for 40% to 50% of the total tracks, where the benchmark circuits use 6% less area comparing to architectures with no coarse-grain tracks. It is interesting to note that even though 94.6% of LUTs in our benchmark circuits belong to 4-bit wide datapath components [18], only 40% to 50% of coarse-grain tracks are needed. We found that many datapath component not only are connected by buses but also by a substantial amount of non-bus control signals, indicating that even highly regular circuits need many fine-grain tracks. The right hand axis of Fig. 9 also shows the area data normalized against the best standard architecture. All coarse-grain architectures performed better than the best standard architecture, where the 100% point represents the area of the standard architecture when implementing the same circuits. Even with no coarse-grain routing tracks, the datapath architecture is 3.6% smaller due to the more efficient datapath-oriented placement and routing. The best coarse-grain architecture is 9.6% smaller than the best standard architecture.

Finally, Fig. 10 shows that cluster size of four and logic track length of two are the best architectural choice for the datapath architecture. Here we measure area against track length and cluster size. The percentage of coarse grain tracks is set to be 50%.

5. Conclusion

In this paper we have proposed a datapath-oriented FPGA architecture with coarse-grain routing tracks. We used a set of datapath-oriented synthesis, packing, placement, and routing tools to investigate the effects of coarse-grain architectural variants on FPGA area for highly regular datapath circuits.

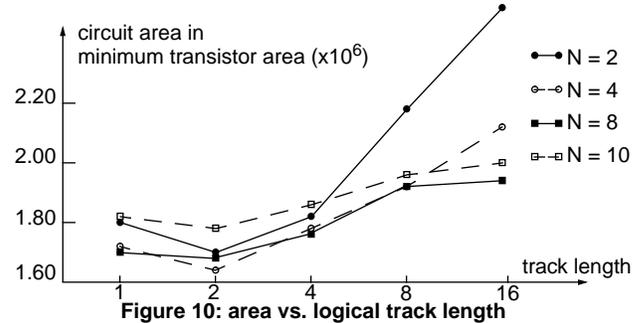


Figure 10: area vs. logical track length

ath-oriented synthesis, packing, placement, and routing tools to investigate the effects of coarse-grain architectural variants on FPGA area for highly regular datapath circuits.

We found that, in order to achieve the best area results, 40% to 50% of the total routing tracks should be coarse-grain despite the fact that, in our benchmark circuits, over 94% of LUTs are in regular datapath components. Furthermore, for cluster size of four, the best datapath architecture is 9.6% smaller than the best standard architecture.

References

- [1] Altera Data Sheet, Altera, 2002.
- [2] Pico-Java Processor Design Documentation, Sun Microsystems Inc., 1999.
- [3] Xilinx Datasheet, Xilinx, 2002.
- [4] V. Betz, J. Rose, A. Marquardt, **Architecture and CAD for Deep-Submicron FPGAs**, Kluwer Academic Publishers, 1999.
- [5] S. Brown, R. Francis, J. Rose, Z. Vranesic, **Field-Programmable Gate Arrays**, Kluwer Academic Publishers, 1992.
- [6] D. Cherepacha, D. Lewis, "DP-FPGA: an FPGA architecture optimized for datapaths", Proc. of Ninth Int. Conf. on VLSI Design, pp. 329-343, 1996.
- [7] T. J. Callahan, P. Chong, A. DeHon, J. Wawrzyniec, "Fast module mapping and placement for datapaths in FPGAs", Proc. of the ACM/SIGDA Sixth Int. Symp. on FPGA, pp. 123-132, 1998.
- [8] M. R. Corazao, M. A. Khalaf, M. Potkonjak, J. M. Rabaey, "Performance optimization using template mapping for datapath-intensive high-level synthesis", Trans. on CAD, pp. 877-888, August 1996.
- [9] A. Koch, "Structured design implementation — a strategy for implementing regular datapaths on FPGAs", Proc. of the ACM Fourth Int. Symp. on FPGA, pp. 151-157, 1996.
- [10] A. Koch, "Module compaction in FPGA-based regular datapaths", Proc. of the 33rd DAC, pp. 471-476, 1996.
- [11] T. Kutzschebauch, L. Stok, "Regularity driven logic synthesis", Proc. of IEEE/ACM Int. Conf. on CAD, pp. 439-446, 2000.
- [12] T. Kutzschebauch, "Efficient logic optimization using regularity extraction", Proc. of Int. Conf. on Computer Design, pp. 487-493, 2000.
- [13] G. Lemieux, S. Brown, D. Vranesic, "On Two-Step Routing for FPGAs," ACM Symp. on Physical Design, pp. 60-66, 1997.
- [14] A. Marshall, J. Vuillemin, B. Hutchings, "A reconfigurable arithmetic array for multimedia applications", Proc. of the ACM/SIGDA Seventh Int. Symp. on FPGA, pp. 135-143, February 1999.
- [15] A. R. Naseer, M. Balakrishnan, A. Kumar, "FAST: FPGA targeted RTL structure synthesis technique", Proc. of the Seventh Int. Conf. on VLSI Design, pp. 21-24, 1994.
- [16] A. R. Naseer, M. Balakrishnan, A. Kumar, "Direct mapping of RTL structures onto LUT-based FPGAs", Trans. on CAD, pp. 624-631, July 1998.
- [17] A. Ye, "FPGA architecture for datapath circuits", Ph.D. thesis in progress, University of Toronto, 2003.
- [18] A. Ye, J. Rose, D. Lewis, "Synthesizing datapath circuits for FPGAs with emphasis on area minimization", IEEE Int. Conf. on Field-Programmable Technology, pp. 219-227, December 2002.