

A New Switch Block for Segmented FPGAs

M. Imran Masud and Steven J.E. Wilton

Department of Electrical and Computer Engineering
University of British Columbia,
Vancouver, B.C., Canada,
{imranm|stevev}@ece.ubc.ca
<http://www.ece.ubc.ca/~stevev>

Abstract. We present a new switch block for FPGAs with segmented routing architectures. We show that the new switch block outperforms all previous switch blocks over a wide range of segmented architectures in terms of area, with virtually no impact on speed. For segments of length four, our switch block results in an FPGA with 13% fewer transistors in the routing fabric.

1 Introduction

An FPGA architecture consists of programmable logic elements and a programmable routing fabric. In commercial architectures, the routing consumes most of the chip area, and is responsible for most of the circuit delay. As FPGAs are migrated to more advanced technologies, the routing fabric becomes even more important [1]. Thus, there has been a great deal of recent interest in developing efficient FPGA routing architectures.

FPGA routing architectures consist of two components: fixed wires (tracks) and programmable interconnect between these tracks. A typical architecture is shown in Figure 2; the logic elements are surrounded by horizontal and vertical *channels*, each channel containing a number of parallel tracks.

At the intersection of each of these horizontal and vertical channels is a programmable interconnect block, often called a *switch block*. Each switch block programmably connects each incoming track to a number of outgoing tracks. Clearly, the flexibility of each switch block is key to the overall flexibility and routability of the device. Since the transistors in the switch block add capacitance loading to each track, the switch block has a significant effect on the speed of each routable connection, and hence the speed of the FPGA as a whole. In addition, since such a large portion of an FPGA is devoted to routing, the chip area required by each switch block will have a large effect on the achievable logic density of the device. Thus, the design of a good switch block is of the up-most importance.

Figure 1 shows three previous switch block architectures that have been proposed. In each block, each incoming track can be connected to three outgoing tracks. The topology of each block, however, is different.

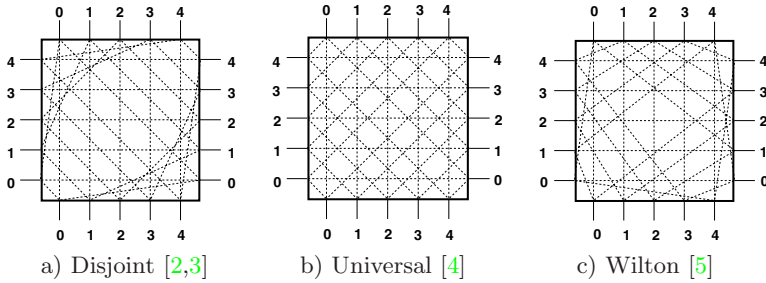


Fig. 1. Previous switch blocks.

Each of the blocks in Figure 1 was developed and evaluated assuming an architecture with only single-length wires (i.e. wires that only connect neighbouring switch blocks). Real FPGAs, however, typically have longer wires which connect distant switch blocks. Such a routing architecture is called a *segmented architecture*, and it is known that such architectures lead to a higher density and speed than an architecture with only single-length wires. Although each of the switch blocks in Figure 1 can be used in segmented architectures, this may not lead to the best density and speed. In particular, [6] showed that the Wilton switch block, while providing the best routability when used in a single-length architecture does not work as well as the Disjoint block in segmented architectures.

In this paper, we present a switch block designed for a segmented architecture. We show that it leads to significantly denser FPGAs than any other proposed switch block over a wide range of segmented architectures. This is important, since all commercial FPGAs rely on segmented routing of some sort, and we are unlikely to see any future FPGAs with only single-segment wires.

This paper is organized as follows. Section 2 describes the architecture we are targeting. Section 3 describes the new switch block, and Section 4 compares it to existing switch blocks.

2 Architectural Assumptions

We assume an island-style FPGA, in which each logic block is surrounded by vertical and horizontal routing channels, as shown in Figure 2.

Each logic block is assumed to be a cluster of four 4-input lookup tables and flip-flops. The logic cluster has 10 inputs and 4 outputs; each output can be fed-back to any of the lookup tables within the logic block. It is assumed that the four flip-flops are clocked by the same clock, and that this clock is routed on a dedicated FPGA routing track. Each of the other logic block inputs and outputs can be programmably connected to one-quarter of the tracks in a neighbouring channel.

Each routing channel consists of W parallel fixed tracks. We assume that all tracks are of the same length s . If $s > 1$, then each track passes through $s - 1$ switch blocks before terminating. If $s = 1$, each track only connects neighbouring switch blocks (an unsegmented architecture). A track can be connected to a perpendicular track at each switch block through which the track travels, regardless of whether the track terminates at this switch block or not. The starting and ending points of tracks within a channel are assumed to be staggered, so that not all tracks start and end at the same switch block. This architecture was shown to work well in [6,7], and is more representative of commercial devices than architectures considered in previous switch-block studies [3,4,5]. Figure 2 shows this routing architecture graphically for $W = 8, s = 4$.

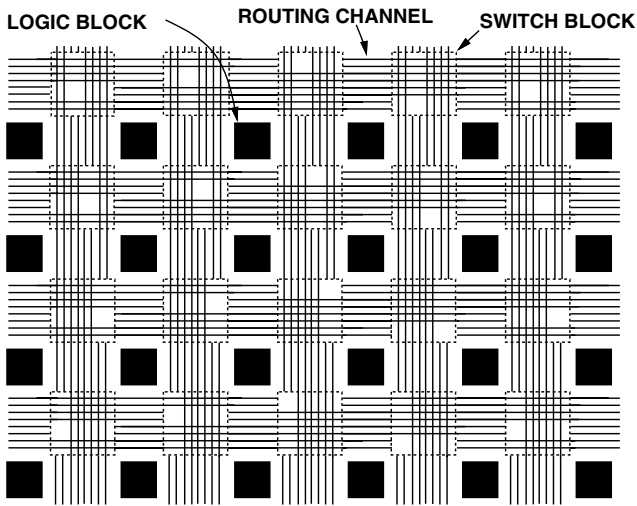


Fig. 2. Segmented Routing Arch. ($W = 8, s = 4$)

3 Switch Block Architectures

3.1 Previous Switch Blocks

Figure 1 shows three previously proposed switch blocks. In each case, each incoming track can be connected to three outgoing tracks. The difference between the blocks is exactly which three tracks each incoming track can be connected. In the Disjoint block, the connection pattern is “symmetric”, in that if the tracks are numbered as shown in Figure 1, each track numbered i can be connected to any outgoing track also numbered i [2,3]. This means the routing fabric is divided into “domains”; if a wire is implemented using track i , all segments that implement that wire are restricted to track i . It is known that this results in

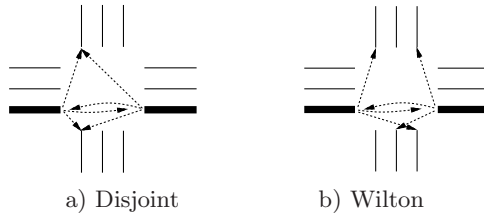


Fig. 3. Wire terminates at switch block

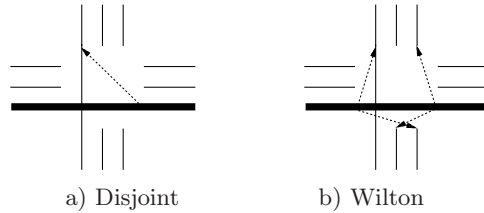


Fig. 4. Wire passes through switch block

reduced routability compared to the other switch blocks. In the universal block, the focus is on maximizing the number of simultaneous connections that can be made using the block [4]. This does not take into account interactions between neighbouring switch blocks. The Wilton switch block is similar to the Disjoint switch block, except that each diagonal connection has been “rotated” one track [5]. This eliminates the “domains” problem, and results in many more routing choices for each connection.

The next section will show that the Wilton block is the most efficient for single-length routing architectures. It is not, however, the best choice in an FPGA with longer segments. This is because it requires more switches than the Disjoint block in such an architecture. Consider a track that terminates at a switch block. Figure 3(a) shows that for a Disjoint block, two horizontal wire segments require 6 switches to connect straight across and diagonally up and down (uni-directional switches are assumed). Figure 3(b) shows the same thing for the Wilton switch block; again, 6 switches are required. Now consider a track that passes through a switch block (and hence has a length greater than 1). In the Disjoint switch block, 5 of the 6 switches are now redundant, as shown in Figure 4(a). In the Wilton switch block, however, only two are redundant. Thus, when a wire does not terminate at a switch block, the Disjoint switch block requires fewer switches than the Wilton block, and hence is smaller and faster. In [6], it is shown that this has a significant effect on the overall speed and density achievable in the device.

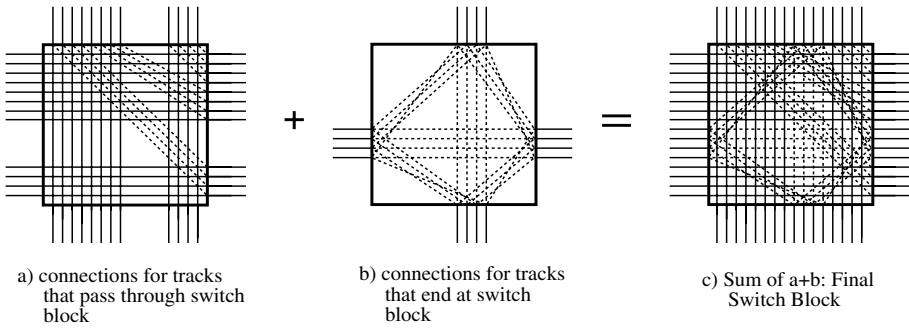


Fig. 5. New Switch Block ($W = 16, s = 4$)

3.2 New Switch Block

In this section, we propose a new switch block that combines the routability of the Wilton block and the implementation efficiency of the Disjoint block.

Figure 5 shows the new block for an FPGA with $W = 16$ and $s = 4$. The incoming tracks are divided into two subsets: those that terminate at this switch block and those that do not. Those tracks that do not terminate at the switch block are interconnected using a Disjoint switch pattern, as shown in Figure 5(a). Because of the symmetry of the Disjoint block, only one switch is required for each incoming track. The tracks that do terminate at the switch block are interconnected using a Wilton switch pattern, as shown in Figure 5(b). The two patterns can then be overlaid to produce the final switch block, as shown in Figure 5(c). Clearly, this pattern can be extended for any W and s .

Compared to the Wilton switch block, the new block requires fewer transistors. In the Wilton switch block, each track that does not terminate at the switch block requires 4 switches, as shown in Figure 4(b). The new switch block, however, only requires a single switch for each of these tracks. For the tracks that *do* terminate at this switch block, each block requires the same number of switches. Thus, we would expect the new switch block to be significantly smaller than the Wilton block in segmented routing architectures. As s increases, the number of tracks that terminate at each switch block decreases, meaning the new switch block is even more area efficient, compared to the Wilton block.

Compared to the Disjoint block, the new block has improved routability. As described above, the Disjoint block partitions the routing fabric into W subsets; all segments that make up a connection must be routed using the same subset. In the new switch block, the number of subset is reduced to s . Since there are fewer subsets, each subset is larger, and thus there are many more choices for each routing segment.

4 Experimental Results

In this section, we compare the proposed switch block to the existing switch blocks over a wide range of segmented architectures.

Nineteen large benchmark circuits were used. Each circuit was first mapped to 4-input lookup tables and flip-flops using Flowmap/Flowpack [8]. The lookup tables and flip-flops were then packed into logic blocks using VPACK [6] (recall each logic block contains four lookup tables and four flip-flops). VPR was then used to place and route each circuit [6]. For each circuit and each architecture, the minimum number of tracks per channel needed for 100% routability was found; this number was then multiplied by 1.2, and the routing was repeated. This “low stress” routing is representative of the routing performed in real industrial designs. Detailed area and delay models were then used to estimate the efficiency of each implementation [6].

Figure 6 shows area comparisons for each of the four switch blocks as a function of s (segmentation length). The vertical axis is the number of minimum-width transistor equivalents per tile in the routing fabric of the FPGA, averaged over all benchmark circuits (geometric average). Previous switch block papers use the number of tracks required to route each circuit as an area metric; our metric is more accurate since it includes the effects of different switch block sizes. In addition to the transistors in the programmable routing, each tile contains one logic block with 1678 minimum transistor equivalents [6], so the entire tile area can be obtained by adding 1678 to each point in Figure 6. As the graph shows, the new switch block performs better than any of the previous switch blocks over the entire range of the graph, except for $s = 1$, in which the new switch block is the same as the Wilton block. The best area results are obtained for $s = 4$; at this point, the FPGA employing the new switch block requires 13% fewer transistors in the routing fabric.

Figure 7 shows delay comparisons for each of the four switch blocks. The vertical axis is the critical path of each circuit, averaged over all benchmark circuits. A $0.35 \mu\text{m}$ process was assumed. Clearly, the choice of switch block has little impact on the speed of the circuit. If $s = 4$, the proposed switch block results in critical paths that are about 1.5% longer than in an FPGA employing the Subset switch block. However, this is the average over 19 circuits; in 9 of the 19 circuits, the proposed switch block actually resulted in faster circuits.

5 Conclusions

In this paper, we have presented a new switch block for FPGAs with segmented routing architectures. This new switch block combines the routability of the Wilton block with the area efficiency of the Disjoint block. Experimental results have shown that the new switch block outperforms all previous switch blocks over a wide range of segmented architectures. For segments of length 4, our switch block results in an FPGA with 13% fewer routing transistors. The speed performance of FPGAs employing the new switch block is roughly the same as that obtained using the previous best switch block.

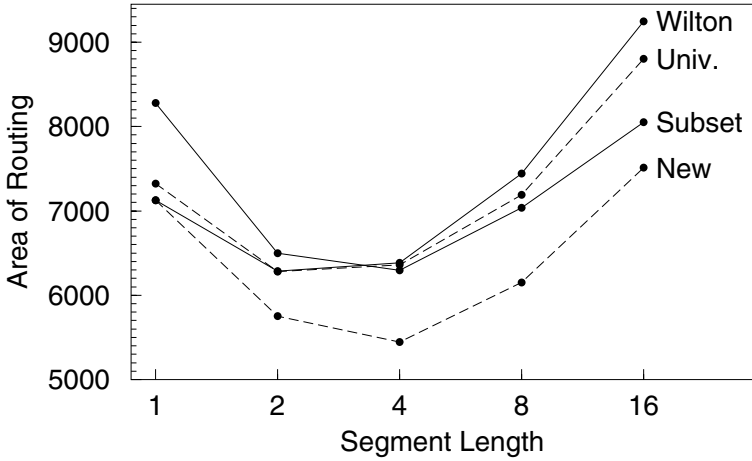


Fig. 6. Area results

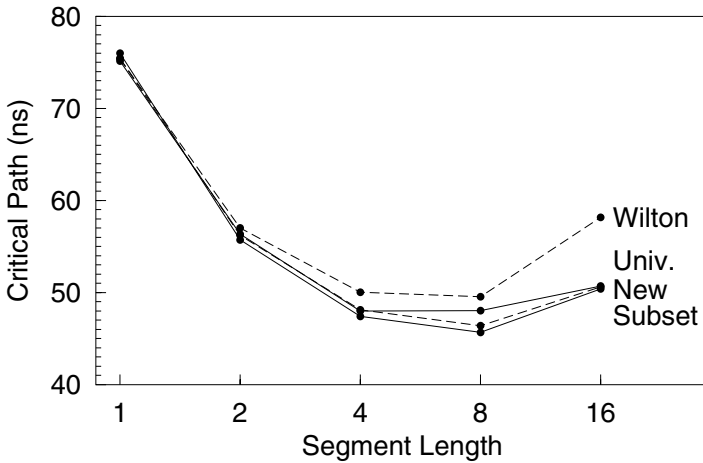


Fig. 7. Delay results

Acknowledgments

This work was supported by British Columbia’s Advanced System Institute, the Natural Sciences and Engineering Research Council of Canada, and UBC’s Centre for Integrated Computer Systems Research. The authors wish to thank Dr. Vaughn Betz for his helpful discussions and for supplying us with the VPR place and route tool.

References

1. J. Rose and D. Hill, "Architectural and physical design challenges for one-million gate FPGAs and beyond," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 129–132, Feb. 1997. 274
2. Xilinx, Inc., *The Programmable Logic Data Book*, 1994. 275, 276
3. G. G. Lemieux and S. D. Brown, "A detailed router for allocating wire segments in field-programmable gate arrays," in *Proceedings of the ACM Physical Design Workshop*, April 1993. 275, 276
4. Y.-W. Chang, D. Wong, and C. Wong, "Universal switch modules for FPGA design," *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, pp. 80–101, January 1996. 275, 276, 277
5. S. J. E. Wilton, *Architectures and Algorithms for Field-Programmable Gate Arrays with Embedded Memory*. PhD thesis, University of Toronto, 1997. 275, 276, 277
6. V. Betz, *Architecture and CAD for Speed and Area Optimizations of FPGAs*. PhD thesis, University of Toronto, 1998. 275, 276, 277, 279
7. V. Betz and J. Rose, "FPGA routing architecture: Segmentation and buffering to optimize speed and density," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Feb. 1999. 276
8. J. Cong and Y. Ding, "FlowMap: an optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, pp. 1–12, January 1994. 279