# A Comparison Between Mask- and Field-Programmable Routing Structures on Industrial FPGA Architectures

Luca Macchiarulo
Politecnico di Torino
C.so Duca degli Abruzzi 24
10129 Torino, Italy
+39 011 5644151
luca.macchiarulo@polito.it

Consolato F. Caccamo
Politecnico di Torino
C.so Duca degli Abruzzi 24
10129 Torino, Italy
+39 011 5644151
fabio.caccamo@polito.it

Davide Pandini
STMicroelectronics, Central R&D
via Olivetti 2
Agrate Brianza 20041, Italy
+39 039 6036437
davide.pandini@st.com

## ABSTRACT

In this paper we compare the routing architecture of island-style FPGAs based on field-programmable switch boxes with a mask-programmable routing structure, in order to assess its position in the design space of routing opportunities available to VLSI IC designers. Although the results presented in this work depend on a few implementation details that will be discussed in the paper, the mask-programmable routing structure shows a large area saving and delay improvement with respect to the field-programmable switch box. As a consequence, we believe that between the two bounds of the design space, i.e., ASICs and FPGAs, there are several hybrid architectural solutions trading off performances, power, area, and programmability, which in the future can be considered for different applications.

## Categories and Subject Descriptors

B.7.1 [Hardware]: Integrated Circuits – *Gate arrays*, *Advanced technologies, VLSI (very large scale integration).*

## General Terms

Performance, Design.

## Keywords

FPGA, mask-programmable, interconnect architectures, routing.

## 1. INTRODUCTION

As technology rapidly scales down into the nanometer range, standard cell-based ASIC designs are becoming affordable only for large volume applications with critical performance, area, and power constraints. Therefore, for low- up to mid-volume applications with less critical requirements, an increasing number of designs are implemented in FPGAs, whose most important characteristics are: better manufacturability due to more regular layout patterns, better timing predictability and signal integrity due to structured routing channels, re-programmability, faster turn-around-time and time-to-market, and reduced NRE costs. However, FPGAs still suffer from limiting factors such as higher cost per part, area occupation, power consumption, and reduced performances. These drawbacks with respect to standard cells, along with somewhat less mature CAD tools and design methodologies, still limit a wider spread among VLSI IC designers. Recently, programmable logic blocks and interconnect structures where logic functions and routing can be customized using via mask patterns instead of field-programmable CMOS active switches have been proposed [1]. As a consequence, new hybrid architectures that combine field-programmability with mask-customizable blocks are becoming more and more attractive for all the applications where performances are still very important, but cannot afford the typical ASIC NRE costs. Another important feature in FPGAs is the interconnect structure that connects the different logic blocks. Typically up to 90% of the total chip area is allocated to the routing resources necessary to implement complex hierarchical interconnect architectures. The conflicting requirements of ease of programming, speed of communication, and congestion control, make the task of FPGA interconnect design very challenging, and dependent on the overall design parameters (number and dimension of logic blocks, balance between fast long-range and short low-fanout interconnections, and power control). In *island-style* FPGA architectures, square-shaped logic blocks are interleaved across the chip as islands in a *sea-of-interconnects*, to which they can connect through programmable switches (normally implemented with pass-transistors or transmission-gates). The connection of a block output pin to all its fanouts is achieved through routing channels connected to each other by means of switch boxes. At every intersection of a horizontal and vertical channel there is a switch box, which is a set of programmable switches allowing the connection between some of the incoming wire segments with other outgoing wire segments on different sides. It is through the careful design of these elements that an FPGA interconnect structure can satisfy the demanding routing requirements of dense and high-performance VLSI designs.

In this work we propose a novel hybrid field- and mask-programmable architecture derived from the Xilinx Virtex™-II FPGA, based on a via-customizable routing structure that implements the switch box functionality. Replacing the field-programmable switch box with such mask-programmable structure achieves a significant saving in terms of total area, and a much better performance for an architecture directly derived from the Virtex™-II. This paper is organized as follows: Section 2 describes the Virtex™-II device family, while Section 3 presents the mask-programmable routing structure and outlines the hybrid field- and via-customizable FPGA architecture. In Section 4

*GLSVLSI'04*, April 26-28, 2004, Boston, Massachusetts, USA
Copyright 2004 ACM 1-58113-853-9/04/0004...$5.00.

important implementation issues are discussed, and results demonstrating the effectiveness of this approach are presented. Finally, Section 5 summarizes our conclusive remarks.

# 2. THE XILINX VIRTEX™-II FPGA

The Xilinx Virtex™-II product families are organized in a two-dimensional array of logic blocks that can be used to implement both combinational and synchronous circuits. Each logic block includes four similar slices, and each slice comprises two 4-LUTs, carry logic, arithmetic gates, multiplexers, and storage elements. Compared to older architectures, the Virtex™-II has reached a high level of sophistication in terms of functionality and performance. In modern FPGAs with medium logic granularity it is very unlikely that the classic basic structure of a simple k-LUT is maintained. On the contrary, while the logic block is still organized around a kernel of generic function generators, a series of improvements have been added to help with specific tasks, such as high-speed arithmetic or multiplexing. The Virtex™-II logic block structure will be analyzed more in detail in Section 2.1, and a complete description can be found in [2].

## 2.1 Logic Block

We estimated the Virtex™-II logic block size from the functional description contained in the Xilinx data sheets, and used the data from an industrial 0.13μm technology. We believe that this approach is fairly accurate with respect to the actual FPGA logic block structure. In order to evaluate our mask-programmable routing structure implemented in a Virtex™-II-like architecture on industrial test cases rather than performing only architectural explorations, we derived a realistic description of the original logic block using the limited technical information available. In the Virtex™-II architecture the logic block is a cluster of four *slices*, where each slice consists of two programmable function generators capable of implementing any 4-input Boolean function, fast carry logic, multiplexers and full adder. The area of each function generator was evaluated based on the multiplexer tree model. The SRAM programming bits are implemented with minimum size transistors, while the multiplexer tree is designed to match the overall maximum delay given by the Virtex™-II specifications. In order to estimate the total area of the logic block, missing any further layout and/or technology data, we considered the standard cell implementation of basic gates in a comparable technology for all the sparse logic (ANDs, ORs, XORs, MUXs), and performance-area trade-offs for the LUT tree. The local routing area estimation is more complicated, and neither the documentation nor the technical literature gives clear indications on this topic. However, the intra-block interconnect is fully populated, and we derived the internal interconnect area following the approach described in [6], where it is reported that Xilinx employs similar strategies.

## 2.2 Interconnect Structure

The FPGA type supported by VPR [3] is purely *island-style*. Such model is not completely coherent with the Virtex™-II architecture. However, it is possible to interpret this architecture as an *island-type* FPGA with the only change of a direct connection from the logic block to the switch block. We believe that such a change makes little difference in terms of routing. The Virtex™-II architecture consists of different routing resources: *long lines, hex lines, double lines*, and *direct connections* ([2]). While long lines cannot be described inside VPR, the other connections, with their staggered nature and their partial connectivity, can be well described in their topology. Furthermore, direct connections can be partially captured by introducing 1-spanning wires. The switch matrix must accommodate as much as 176 wires to be routed in all directions. A crucial point is to estimate the timing characteristics of the interconnections, in order to compare the field-programmable against the via-customizable routing structures. The basic programmable element in a Virtex™-II is the pass-transistor. Hence, it is possible to obtain a simplified electrical model, whose parameters are source and drain capacitances, together with a resistance $R_{MOS}$ estimated by means of circuit simulations (or electrical measurements) in the device working region. Accurate models of MOS transistors in 0.13μm technology were used for circuit simulations, to obtain resistance values for different device sizes. The choice of the resistance value was guided by [5] and by iterative experiments focused on optimizing area/delay trade-offs. The transistor found was about six times the minimum contactable dimension, approximately in the range suggested in [5]. After the pass-transistor size was determined, the switch box was designed. First, the *switch point* that is a cluster of six transistors (along with their SRAM programming bits) was built. Then, switch points were implemented in a regular square layout scheme. Since we did not want to assume any layout style, and did not have any information on this topic from the FPGA data sheets, we adopted the more conservative approach of summing the areas of the transistors necessary to generate a complete switch box, and added a small fraction to account for the switch box intra-routing. The switch point includes 6 pass-transistors of about six times the minimum size, plus 36 minimum size transistors (for the SRAM programming bits) for every switch point. This number is multiplied by the number of existing tracks, i.e., 176 in our simplified Virtex™-II architecture. By including a fraction of 20% for routing resources, we obtained a switch box of approximately 2.5 times the side of the logic block, confirming that our design is interconnect-dominated.

By knowing the logic block and switch box dimensions, we estimated length and pitch of the routing tracks. The only other missing information to extract the electrical parameters necessary for timing estimation was the wire widths and their spacing. Following the conclusions presented in [6] indicating the better performance of minimum width wires as a way to minimize parasitic capacitance[1], we made this choice.

# 3. MASK-PROGRAMMABLE ROUTING STRUCTURE

The basic field-programmable switch box described in Section 2.2 can be used to derive a mask-programmable routing structure. We chose to implement the mask-programmable routing structure with a crossbar, since in [8] it was demonstrated that a switch box is inferior to the crossbar in terms of area utilization, and as the number of routing tracks increases such as in a complex architecture like the Virtex™-II, the switch box area dominates the total design area. The crossbar structure is based on horizontal and vertical routing tracks on two adjacent metal layers. Each horizontal track can directly connect to every vertical track by

[1] Wire capacitance values were extracted with the academic tool FastCap [7].

means of a via, and each vertical track can connect to any horizontal track similarly. The typical crossbar architecture is shown in Figure 1.

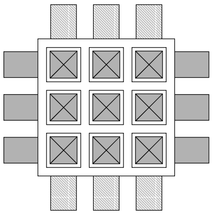

**Figure 1. Mask-programmable crossbar**

Another reason to choose a crossbar structure is related to the fact that for the Virtex™-II architecture, the number of connections makes the switch box larger than the logic block. Therefore, the design is interconnect-dominated rather than logic-dominated, and a more compact routing structure would alleviate the overall area problem. It is important to point out that, even though in this work we considered an *island-style* architecture and a tile similar to the Virtex™-II, a stacked solution, where the mask-programmable routing structure is implemented above the logic block[2] would also be possible [1]. In this case, the tile size will be dictated by the largest between the logic block and the connection block. Hence, a careful choice of the mask-programmable structure may significantly reduce the overall area of hybrid field- and mask-programmable devices. Since a crossbar is not logically equivalent to the FPGA switch box, in our experiments we achieved routability for both types of interconnect structures.

## 3.1 Buffered Architectures

There is a fundamental problem to consider when evaluating mask- vs. field-programmable routing solutions: should the routing structure be partially or totally buffered, and, in such case, how should buffering be implemented, and to which connections should it be applied? We will show in Section 4 that these critical issues impact the conclusions of this work. For our experiments, we decided to use two routing architectures, one fully buffered, and the other without buffers. For the unbuffered architecture, a trade-off between area and delay must be considered in the field-programmable case for the transistor sizes used inside the switch point. We followed the approach described in [5] for optimizing the area-delay product[3], and we obtained transistors that are about six times as large as the minimum size. In contrast, the transistor resistance value was obtained after circuit simulations carried out under different load conditions. The parasitic components of the mask-programmable structure were extracted using FastCap [7], while the via resistance was obtained from the 0.13μm technology data. For the buffered architecture the number of parameters increases, and the choice of one parameter may influence the value of others. Similarly, an area-delay trade-off was determined for the field-programmable structure. Although only a trial-and-error procedure guided by circuit simulations was pursued to identify the relevant parameter trends (propagation delay, output slope, etc.), the final transistor sizes were in excellent accordance to those indicated in a more systematic study [5]. The topology of the buffered structure considered was a couple of cascaded inverters of increasing size, followed by a large nMOS device. This approach can save routing resources and the number of transistors, even though it is not electrically optimal. All nMOS switches were boosted by a threshold voltage to compensate the slow and incomplete charge of its output voltage. Differently from the field-programmable switch box, it is possible to accommodate the buffers inside the channels *without* area penalty, since a simple couple of vias will either connect a buffer to a wire segment or bypass it, without wasting area or increasing wiring congestion. In our experiments, the maximal buffer size was given by the channel area divided by twice the number of tracks per channel (buffers are needed for both wire directions). The buffer characteristics were determined on the basis of such area values, and circuit simulations were performed to obtain the timing data.

# 4. RESULTS

In order to evaluate the mask-programmable routing structure described in Section 3, we followed a common design flow for the ISCAS benchmarks and for two industrial circuits. The flow for the ISCAS benchmarks was:

1. The BLIF description in 4-input functions was obtained to pack[4] the circuits into our hybrid mask-programmable Virtex™-II-like logic blocks;
2. VPR was run with four different architectures: MOS[5] with no buffering, VIA with no buffering, MOS with buffering and VIA with buffering, fixing the number of tracks per channel (176 to emulate the Virtex™-II interconnect matrix capability), and the number of rows and columns of the smallest Virtex™-II device that allows the design placement.

For the industrial circuits, the flow was more articulated:

A. The design was synthesized with FPGA Compiler II™ [4];

B. An *ad-hoc* developed parser was used to convert the synthesized netlist into the BLIF format compatible with T-VPack. This step had to deal with the simplified logic blocks of the VPR model (a single type of register vs. different storage elements in Virtex™-II, etc.).

After these two preliminary steps, the design flow was applied as usual. The synthesis with an industrial tool gave us the possibility of following a complete flow on a real Virtex™-II architecture (with a link to the Xilinx Foundation™ software), whose results were compared against the VPR results.

Table 1 reports the experimental results for the two routing structures both for the unbuffered and buffered case. The first column reports the benchmark names (STInd1 and STInd2 are the two industrial circuits from STMicroelectronics, Inc.), the second the dimension of the minimum Virtex™-II-like block array that allows at least one *place&route* completion on the different routing architectures[6]. Finally, the other columns report the critical path delay, expressed in nanoseconds, preceded by the critical path delay due to routing only. This value is the most important for our investigation, as in all experiments the logic blocks are identical. It is important to note that there is a

[2] Buffering is discussed in detail in Section 3.1.

[3] It is important to note that a correct evaluation of the transistor size depends on several factors such as reliability, power consumption, and printability that we could not take into account in this work.

[4] T-VPack [3] was employed to obtain 4-LUT clustered netlists.

[5] MOS is the field-programmable, while VIA is the mask-programmable architecture.

[6] Unrouted architectures have a hyphen in their critical path column.

significant difference between the field- and the mask-programmable architecture. The better performance of the via-customizable solution is straightforwardly explained by the large difference in the electrical parameters values, particularly the resistance of the programming elements: in the field-programmable (MOS) case, a pass-transistor introduces a non-negligible resistive path, where the mask-programmable (VIA) has a simple via whose resistance is orders of magnitude smaller. Another important conclusion we can deduce is the large unbalance between the weight of the interconnect factor with respect to the logic delay in determining the critical paths. In general, this indicates that a design is greatly unbalanced and needs more buffering efforts. Buffered architectures, either field- or mask-programmable, present a different perspective. Both cases are faster than the unbuffered architectures, but the relative difference is reduced. Such loss of performance can be explained by the fact that the only relevant difference in the electrical parameters is due to the resistance of the buffered switches, which are much more similar: for the MOS case we have a tristate structure, while for the VIA case we have a buffer with no tristate control, followed by a negligible (series of) via resistance(s). The question is: which of the two cases better resembles the "real life"? A reasonable answer is given by looking at the *place&route* results obtained with the Xilinx tools on the industrial examples. STInd2 has a critical path routing delay of 40ns. Besides the almost perfect match with our MOS buffered case (39.7ns obtained with VPR), it is clear that the impact of buffers cannot be overlooked if a realistic design is pursued.

Finally, the estimated area saving of the mask-programmable crossbar with respect to the field-programmable switch box is about 75%, which in turn reduces the overall area of the hybrid mask-programmable Virtex™-II-like architecture of about 60% (or more for architectures with fewer routing tracks). It is worth noting that the area estimations were carried out consistently, with the limited technological and architectural information available, and therefore, may not be extremely accurate, although we believe they are quite realistic.

## 5. CONCLUSIONS AND FUTURE WORK

In this work we have analyzed the trade-offs between performance, area, and programmability between a field- and a mask-programmable routing structure, and the potential impact on a complex industrial FPGA architecture like the Xilinx Virtex™-II. Our results show that a straightforward extension of a complex FPGA architecture with mask-programmable parts can be done, but the performance gain must be carefully analyzed on a more accurate FPGA representation than the one available. Future work will explore the different layout choices left to the designer of mask-programmable fabrics besides the use of one or more levels of vias, in order to evaluate the trade-offs between flexibility, routability, and regularity (that implies manufacturability and predictability).

## REFERENCES

[1] L. T. Pileggi, H. Schmit, A. J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, and K. Y. Tong, "Exploring Regular Fabrics to Optimize the Performance-Cost Trade-Off," in *Proc. Design Automation Conf.*, Jun. 2003, pp. 782-787.

[2] Virtex™-II Data Sheets and User Guide, Xilinx, Inc., June 2003.

[3] V. Betz and J. Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research," in *Proc. 7th Intl. Workshop on Field-Programmable Logic*, Aug. 1997, pp. 213-222.

[4] FPGA Compiler II™ User Guide, Synopsys, Inc., Nov. 2000.

[5] V. Betz and J. Rose, "Circuit Design, Transistor Sizing and Wire Layout of FPGA Interconnect," in *Proc. Custom Integrated Circuits Conference*, May 1999, pp. 171-174.

[6] V. Betz and J. Rose, "FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density," in *Proc. Intl. Symp. on FPGAs*, Feb. 1999, pp. 59-68.

[7] K. Nabors and J. White, "FastCap: a Multipole Accelerated 3-D Capacitance Extraction Program," *IEEE Trans. on Computer-Aided Design*, vol. 10, pp. 1447-1459, Nov. 1991.

[8] C. Patel, A. Cozzie, H. Schmit, and L. T. Pileggi, "An Architectural Exploration of Via Patterned Gate Arrays," in *Proc. ISPD*, Apr. 2003, pp. 184-189.

**Table 1. Benchmark features and results**

| Circuit | Array | Critical path (ns) | | Critical path (ns) | |
|---|---|---|---|---|---|
| | | MOS | VIA | MOS | VIA |
| bigkey | 40x32 | 77.7/78.7 | 22.8/23.8 | 6.26/6.95 | 4.2/4.8 |
| des | 48x40 | 150/152 | 38.8/41.7 | 13.4/15.5 | 9.15/11.2 |
| apex2 | 24x16 | 62.7/66.2 | - | 8.92/11.3 | 6.14/8.56 |
| apex4 | 24x16 | 68.8/71.4 | 25.5/28.5 | 8.27/10.3 | 5.62/7.71 |
| alu4 | 24x16 | 57.9/61.4 | 20.7/23.7 | 7.55/9.63 | 5.12/7.2 |
| elliptic | 32x24 | 100/104 | 32.8/39.8 | 11.8/16.0 | 7.21/12.1 |
| e64 | 16x8 | 23.2/24.7 | 10.8/12.8 | 3.63/4.67 | 2.26/3.65 |
| dsip | 40x32 | 121/123 | 29.8/30.8 | 9.41/10.1 | 6.66/7.36 |
| diffeq | 24x16 | 39.3/39.8 | 13.9/17.9 | 3.9/7.4 | 2.7/6.1 |
| clma | 40x32 | 1531/1541 | 310/315 | 32.6/36.8 | 21.4/25.9 |
| s298 | 24x16 | 103.4/109.4 | 43.4/49.9 | 13.8/18.0 | 9.3/13.8 |
| pdc | 32x24 | 169.2/173.2 | 48.2/52.2 | 15.9/18.3 | 10.7/13.2 |
| misex3 | 24x16 | 63.7/66.6 | 20.9/23.9 | 7.2/9.3 | 5.0/7.0 |
| frisc | 32x24 | 99.5/105.5 | 37.9/46.9 | 14.0/20.6 | 9.6/16.2 |
| ex5p | 24x16 | 69.9/73.4 | 39.8/42.8 | 8.7/11.2 | 6.0/8.4 |
| ex1010 | 32x24 | 122.5/125.5 | 41.3/44.8 | 13.9/16.6 | 9.4/12.2 |
| tseng | 24x16 | 27.0/27.5 | 9.4/14.4 | 3.1/6.9 | 1.8/6.0 |
| spla | 32x24 | 147.4/150.9 | 42.2/44.7 | 13.9/16.3 | 9.4/11.9 |
| seq | 24x16 | 67.6/70.1 | 45.0/47.0 | 8.6/10.7 | 5.8/7.9 |
| s38584 | 40x32 | 113.0/114.6 | 29.1/30.6 | 10.0/11.1 | 6.9/7.9 |
| s38417 | 40x32 | 105.7/109.7 | 32.9/36.9 | 11.2/14.3 | 7.6/10.8 |
| STInd1 | 40x32 | 75.5/77.0 | 23.5/27.0 | 8.43/10.8 | 5.84/8.27 |
| STInd2 | 40x32 | 384/423 | 132/192 | 39.7/81.3 | 27.1/69.4 |