Switch-Box Design for Synthesizable Coarse-Grain Arrays for System-on-Chip Applications

Sami Khawam[1]
[1] *School of Engineering and Electronic, The University of Edinburgh, UK, S.Khawam@ee.ed.ac.uk*

Tughrul Arslan[1,2]
[2] *Institute for System Level Integration, Livingston, UK T.Arslan@ee.ed.ac.uk*

Abstract

The use of synthesizable embedded FPGAs in a reconfigurable Systems-on-Chip (SoC) provides numerous improvements to ASIC designs due to the added flexibility and improvements in functionality. Such arrays have been proposed earlier by the authors and it was found that, as with all reconfigurable architectures, most of the power and area consumed is in the programmable interconnects mesh. This paper focuses on the design of optimized synthesizable Switch-Boxes that can be used in reconfigurable coarse-grain architectures. The paper compares different switch-boxes in terms of area, power, delay and routability and proposes new designs optimized for directional data-flow which are found to provide up to 47% less area and 22% less power with only an increase of 10% in routed wirelength and delays when compared to existing designs.

1. Introduction

Adding flexibility to large dedicated ASIC designs provides a number of significant advantages such as the ability to change the designs after fabrication for fixing errors or to support changing standards. Commercial embedded FPGAs such as [5] provide a good solution for achieving high flexibility and high throughput. However, embedded FPGAs suffer from high power and area consumption alongside difficulties in integrating them into the SoC architecture and the system design-flow. To overcome these problems the authors proposed domain-specific Reconfigurable Arrays (RA) in [1] having less flexibility than generic FPGAs, but providing a significant improvement in performance. The arrays are provided as synthesizable soft-cores to allow a better tuning of the array to the desired application domain and to simplify their integration into SoC software flow.

Nevertheless, it was measured that the island-style non-segmented programmable interconnects used in the arrays occupied up to 91% of the total array area and power consumption [1]. Such high ratios are usual for generic fine-grain FPGAs, however this is too high for the purpose of embedded coarse-grain arrays. The inefficiency occurs when trying to build synthesizable interconnects having the same functionality as typical interconnects found in FPGAs. The use of standard-cells libraries restricts the circuit design of the configurable switches since the pass-transistors used in typical FPGAs [7] have to be replaced by synthesizable cells such as tri-state buffers or multiplexers. This significantly increases the area, power consumption and delays: two tri-state buffers forming a bidirectional switch have nearly 8 times the area of a single pass-transistor. Similar area increases are found when trying to implement the configuration memory, as synthesizable alternative for SRAM-cells such as flip-flops or latches can occupy up to 2.7 times more area. As described in [4], a possible solution is to augment the standard-cell library with handcrafted FPGA-friendly cells. However, this reduces the portability of the array between different fabrication technologies.

The general symmetrical mesh of configurable switches [9] used in the array consists of connection-boxes and switch-boxes, with the latter one occupying around 42% of the total area and power consumption [1]. This paper proposes and investigates different designs of switch-boxes suitable for synthesis in reconfigurable coarse-grain architectures. The various designs are evaluated and compared in terms of power consumption, area, delays and routability.

2. Related Work

Published work on switch-boxes is mainly concentrated on non-synthesizable circuit designs and routability studies. Synthesizable interconnects were presented in [3] for small arrays of programmable look-up-tables; interconnects were based on directional switch blocks that allow only left-to-right data flow in order to prevent the occurrence of combinatorial feedback loops. The configurable switches were implemented using tri-state buffers. The Totem project [4] investigated the layout of synthesizable interconnects based on multiplexer and

demultiplexers for one dimensional arrays (not switch-boxes based).

The circuit design of non-synthesizable switch-boxes for generic FPGA has been studied in [7] [8]. In [7] switch-boxes using pass-transistors and buffers were compared and it was found that creating an interconnect structure containing both types of switches in an alternating arrangement provided delay savings. These results are not applicable for synthesizable arrays due to the use of pass-transistors. In [8] and [7] a switch-box based on multiplexers, suitable for synthesizable interconnects, was proposed for fine-grain FPGA. Reference [6] studied the effect of the switch-box style and size on the physical layout of the box circuit. In our study, we suppose that for the purpose of synthesizable switch-boxes the layout task should be left for the ASIC place and route tools, which would not provide enough optimizations as handcrafted layouts.

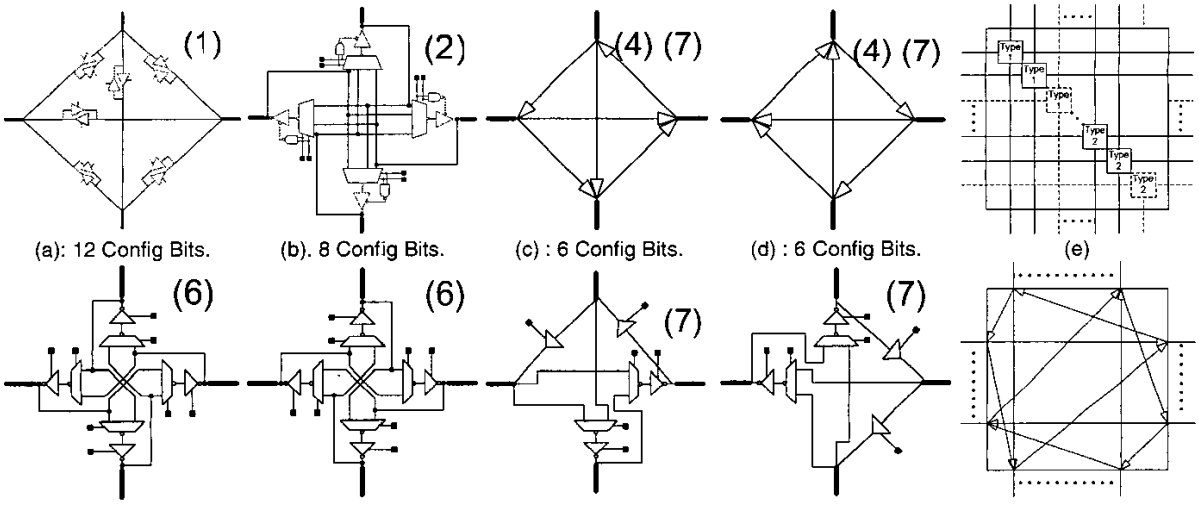
3. Proposed Switch-Boxes

Possible synthesizable elements that make programmable switches are tri-state buffers and multiplexers. Based on this, the following seven switch-box designs are proposed for evaluation:

(1) Full directions, tri-states
(2) Full directions, multiplexers
(3) Full directions, tri-states with reduced cfg memory
(4) Reduced directions, tri-states
(5) Reduced directions, tri-states with reduced cfg mem
(6) Reduced directions, multiplexers
(7) Reduced directions, multiplexers and tri-states

As shown in Figure 1(a) and Figure 1(b), designs (1) and (2) try to create bi-directional switches that connect any two sides together by using tri-state buffers or multiplexers. The switch-points shown have the same functionality as the basic switch made using pass-transistors in generic FPGAs; hence these switches have a relatively higher flexibility when compared to the rest of the switches proposed below. Since the area cost per configuration memory bit is high, area optimizations might be achieved by compressing the memory content: e.g. the number of configuration bits needed in switch (1) can be reduced by removing the redundant states since only 2 bits are required per side to select which of the 3 other sides, if any, has to be routed through. Hence, decoders are used in (3) and (5) to reduce the number of configuration bits to 8 and 4 respectively.

Depending on the placement of the components on the array the data flow can be more intense in some specific directions than others. This is especially true when routing for coarse-grain circuits where the direction of the data-flow is predictable, unlike the case of random logic circuits. Hence, switches (4)-(7) are proposed which favor some directions over the others. As shown in Figure 1(c) and Figure 1(d) for switch (4), two types of switch-points are proposed, each allowing connections only in specific directions. The two types of switch-points are both used in different ratios inside the switch-box as shown in Figure 1(e), which allows creating an overall switch-box that accepts more connections from left-to-right and top-to-bottom. As can be seen in Figure 1(f) and Figure 1(g), the use of 2-to-1 multiplexers in (6) allows the switch to have



(a): 12 Config Bits. (b). 8 Config Bits. (c) : 6 Config Bits. (d) : 6 Config Bits. (e)

(f): 8 Config Bits. (g): 8 Config Bits. (h): 6 Config Bits. (i): 6 Config Bits. (j)

Figure 1: (a): Configurable switch-point using tri-state buffers. (b): Switch using 3-to-1 multiplexers and buffers. (c), (d): Two possible types of unidirectional switches. (e): Combining the two types of switches in an S-box, and (j) is the resulting S-Box topology. (f), (g): Unidirectional switches using 2-to-1 multiplexers. (h), (i): Unidirectional switches using both tri-state buffers and 2-to-1 multiplexers. Number of configuration bits required shown below each switch-point.

a larger flexibility than the buffer-based switch (4). Finally, switch (7) uses both multiplexers and tri-state buffers to create a switch with the same functionality as (4). Switch (5) tries to further compress the configuration memory needed by (4), however, the flexibility is reduced as only two tri-state buffers are allowed to be on at the same time, which also decreases the routability of the design. It should be noted that switch-points with reduced directions are still able to do all the possible connections between two sides by using two tri-state buffers in a row, but this uses more resources and creates more switching activity in the wires, as measured below.

The standard subset switch-box topology shown in Figure 1(j) was used as it was initially measured to provide better routability results than other topologies like the Universal and Wilton s-boxes [2]. The boxes with full directions have a flexibility of Fs=3 (as defined in [9]). This value was initially chosen for simplicity and for creating interconnects that have the same functionality as the ones found in standard FPGAs. This flexibility measure does not apply to the switches with reduced directions, as these would have different values for each side.

4. Performance Evaluation

To compare these designs, an array with each type of switch-box was generated with UMC0.13 technology and the sample DCT circuit from [1] was mapped on it. Standard ASIC design tools were used to measure the area, power and timing of the implementation on each array.

The area of the switch-boxes can be split in two parts: the area needed for the actual switches and the area required by the configuration memory, as shown in Figure 2. The values shown are based on flip-flop configuration memory; other alternatives such as latches would provide slightly less area. The area measurements also include the overhead in the metal routing required, which varies due to changes in the number of wires inside each box.

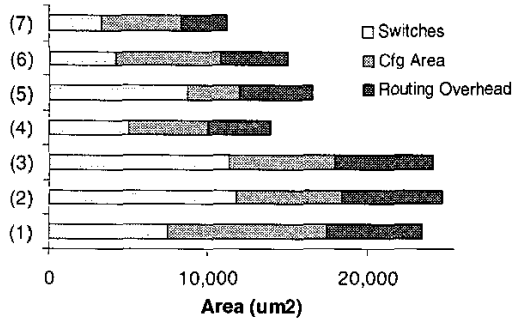


Figure 2: Area of the s-boxes for 12-bit words-tracks.

The relative power consumption and delays measured for each type of switch-boxes are shown in Figure 3. The power values measured are the average of all the switch-boxes in the array, while the delays represent the longest path in each implementation. Similarly, the routability of each switch type is measured using the total wirelength reported by the routing program (Table 1). It should be noted that the ratio of Type 1 and Type 2 blocks in switch-boxes with reduced directions (implementations (4), (5), (6) and (7)) has an effect on the routability of the design depending on the data-flow. The total wirelength was measured for different ratios and it was found that for implementations (4) and (7) the lowest wirelength is achieved when around 65% of the switch blocks are of Type 1. For switch (6) the minimum wirelength occurs when around 60% of the blocks are of Type 2.

As expected the highest areas and energies are consumed by the switch-boxes having full directions (1), (2) and (3). However, these boxes provide the lowest delays and best routability as the switches have a high flexibility which gives short interconnects. It can be clearly seen that the introduction of 3-to-1 multiplexers in implementation (2) is inefficient as it increases the total power consumed by 29%, the area by %5.2, and the delays by 37% when compared to tri-state buffers implementation in (1). Implementation (3) shows that no gain is achieved by compressing the configuration memory, as the area in (3) is 2.8% higher than in (1), due to the high area occupied by the decoding circuit. This decoder also results in a slight increase of 3% in power over (1), even though the decoder is not in the data path and hence does not get as much switching.

It should be noted that these results depend on the number of bits in the word track of the array. The result also depends on the design library and cell-geometries used: other libraries used (UMC0.18) showed results where (2) had up to 11% lower area than (1) for the same widths of tracks.

The switch-boxes with reduced directions have considerably less area than the full directions ones. These

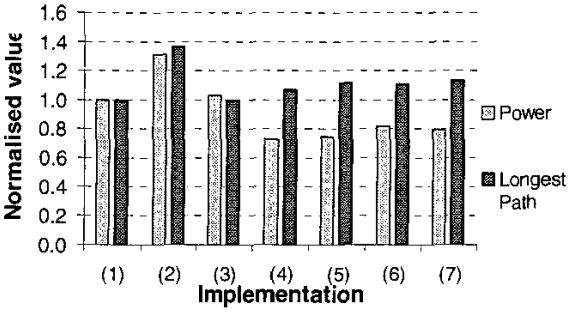


Figure 3: Relative power consumption and delays.

switch-boxes also show between 7% and 14% more delays than the full switch box due to the longer routings created. Implementation (4) has half the area used by (1) since the number of switches and configuration bits is halved. It also shows 27% less power than (1) since the load on the input lines has been reduced and we have less tristate drivers, hence less internal node switching.

In (5), for the chosen number of tracks, the area saving in configuration memory is less than the area occupied by the decoding circuit used, and hence (5) is 18% larger than (4). It should be noted that when the bit-width of the word track is increased, the number of configuration bits stay the same and only the area occupied by the switches is increased. It was measured that the use of compressed configuration memory as in (3) and (5) only offers area advantages for bit-widths below 8 and 4 respectively. The use of 2-to-1 multiplexers in (6) reduces the area taken by switches when compared to tri-state buffers in (4); however, more configuration bits are needed which makes the overall area of (6) 8% higher than (4). Furthermore, these 2-to-1 multiplexers in (6) do not add as much delays as the 3-to-1 multiplexers in (2). Implementation (7) has the lowest area, which is 20% smaller than (4), since the switches area is reduced by using 2-to-1 multiplexers and the number of configuration bits is kept the same. Switch-boxes (6) and (7) consume 8% to 12% more power than (4), while having around 20% less power than (1).

In terms of routability, implementations (4), (5) and (7) with optimized ratios have a wirelength around 12% higher than the implementations with full-directions. Using switch (6) with the optimized ratio we get only a 2% increase in wirelength over the full switch-boxes.

5. Results

From the above evaluations we can deduce that the compression of configuration data (as in (3) and (5)) only provides some area reductions for low widths of word-tracks. The use of 3-to-1 multiplexers (as in (2)) to implements full four-side switch blocks is inefficient as it increases the area, power and delays when compared to the use of tri-state buffers. Good results were achieved using switch-box with reduced directions ((4), (6) and (7)) when compared to full-directions switches.

The half-box based on tri-state buffers (implementation (4)) has a low area, power consumption and delays but a large wirelength. Using 2-to-1 multiplexers (as in (6)) allows big improvements in routability at a price of a slightly larger area, longer delays and higher power consumption. Finally, the lowest area is achieved by combining multiplexers and tri-state buffers in the box (as in (7)) which give low-power consumption but lower routability and longer delays.

The measurement and values shown in this paper greatly depend on the implementation and the data-flow used; however, they represent what can be achieved when making domain-specific design optimizations to coarse-grain reconfigurable architecture.

Table 1: Total wirelength for each implementation. For (4), (7), (5) and (6) the ratio of Type1/Type2 with the lowest wirelength is chosen.

Switch-Box	(1), (2), (3)	(4), (7)	(5)	(6)
Best wirelength	288	144	96	192

6. References

[1] S. Khawam, T. Arslan, F. Westall, "Synthesizable reconfigurable array targeting distributed arithmetic for system-on-chip applications", 2004 *Parallel and Distributed Processing Symposium, (RAW'04)*

[2] M. Imran Masud, Steven J.E. Wilton, "A New Switch Block for Segmented FPGAs", in *International Workshop on Field Programmable Logic and Applications* , Aug. 1999

[3] Kafafi N., Bozman K., Wilton S.J.E, "Architectures and Algorithms for Synthesizable Embedded Programmable Logic Cores", *ACM International Symposium on FPGA*, Monterey, CA, Feb 2003.

[4] Philips S., Hauck S., "Automatic layout of domain-specific reconfigurable subsystems for system-on-a-chip", *Proc. of the 10th ACM Int. Symp. on FPGA*, 2002.

[5] Bryant I., Tanurhan Y., "The Actel Embeddable FPGA Core", *Actel Corporation*, 2001

[6] Schmit H., Chandra V., "FPGA switch block layout and evaluation", *Proc. of the 10th ACM Int. Symp. on FPGA*, 2002.

[7] Lemieux G., Lewis D. "Circuit Design of Routing Switches", *ACM International Symposium on FPGA*, Monterey, CA, Feb 2002

[8] E. S. Ochotta, P. J. Crotty, C. R. Erickson, C.-T. Huang et al, "A novel predictable segmented FPGA routing architecture", *ACM International Symposium on FPGA*, Monterey, CA, Feb 1998

[9] Rose J., Brown S., "Flexibility of interconnection structures for field-programmable gate arrays", *Solid-State Circuits, IEEE Journal of*, Vol.26, Iss.3, 1990, Pages: 277- 282