

A Quantitative Study of The Benefits of Area-I/O in FPGAs

Herwig Van Marck, Jo Depreitere, Dirk Stroobandt and Jan Van Campenhout

University of Ghent
Electronics and Information Systems
St.-Pietersnieuwstraat 41
B-9000 Gent, Belgium

E-mail: {hvm,jdp,dstr,jvc}@elis.rug.ac.be

Abstract

Designs targeted for FPGAs are becoming increasingly larger and more complex. The need for I/O often surpasses the number of I/O pads that can be provided at the perimeter of the FPGA chip. As a result, these designs have to be implemented in larger FPGAs, the size of which is fixed by the number of I/O pads and not by the logic needed, reducing the performance of the implementation. Providing FPGA chips with I/O pads that are spread out across the whole chip area drastically reduces this problem. In this paper, we present a quantitative analysis of the impact of area-I/O in FPGAs.

1 Introduction

Over the past years *Field Programmable Gate Arrays* (FPGAs) have rapidly become widely accepted as an attractive means of implementing digital circuits. They provide designers with a flexible implementation medium. However, as a general rule of thumb, FPGAs can only achieve about 10% of the functionality of custom VLSI circuits of the same silicon area and fabrication technology [3]. As a result, large complex designs have to be partitioned into smaller parts and spread across multiple FPGAs.

Recent advances in interconnection technology have inspired the integration of multiple FPGAs into *Multi-Chip Modules* (MCMs) [5]. These modules create the illusion of one very large FPGA without having the low yield associated with the fabrication of very large VLSI chips. However, to maintain this illusion, the FPGAs require a massive amount of *Input/Output* (I/O). In this paper we will quantify and compare the traditional *perimeter-I/O* architecture, using only the chip edge to accommodate I/O pads, and the so called *area-I/O* architecture, placing I/O pads anywhere on

the chip. Although it seems obvious that area-I/O FPGAs are more suited for integration into larger systems, we will show that this largely depends on the designs being implemented.

This paper is structured as follows. First we introduce the basic ideas for area-I/O FPGAs. Then we proceed with an analysis of the main differences between area-I/O and perimeter-I/O FPGAs. Finally, we compare the performance of area-I/O versus perimeter-I/O FPGAs.

2 FPGAs for Multi-Chip Modules

Since the introduction by Xilinx in 1985, different types of FPGAs have become commercially available. Although each type has its own distinct features, the common characteristics are very much similar (see figure 1). The architecture of an FPGA consists of a two-dimensional array of *Configurable Logic Blocks* (CLBs), providing the logic processing power. These can be programmed to implement an arbitrary logic function of up to 5 variables.

In between the CLBs, there are routing channels providing programmable routing resources. They consist of a number of wire segments and programmable routing switches. The switches connect pins of the CLBs to wire segments or connect two wire segments with each other.

Switch matrices interconnect the wire segments of neighbouring routing channels. It must be noted that in most modern FPGA types so called “*long lines*” exist. These are segments that span multiple channels, bypassing some of the switch matrices to provide faster long interconnections.

External interconnections are implemented by *I/O blocks* (IOBs). These are programmable to provide inputs and/or outputs for the FPGA.

The most common way of making the actual electrical connection between the IOBs and the external world is done

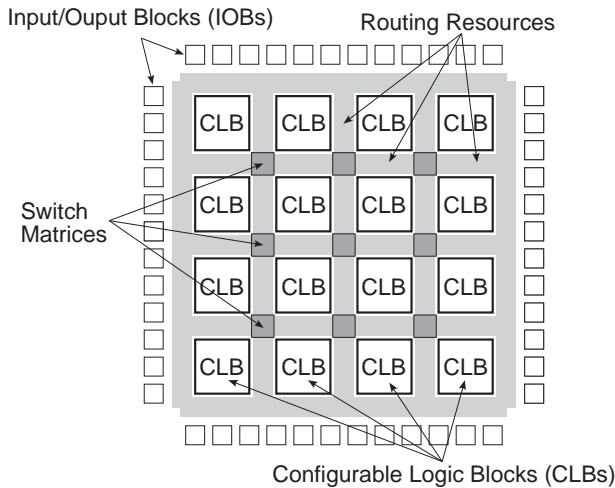


Figure 1. Overview of the common characteristics of FPGAs.

by *wirebonding*. This technique uses a thin wire (called a *wirebond*) to bridge the gap between an I/O pad on the FPGA chip and an external I/O pad (for example on a wafer in a MCM). Due to the restrictions on the length of wirebonds, the I/O pads have to be located at the perimeter of the FPGA. Fig. 2 shows how such a *perimeter-I/O FPGA* could be integrated on a wafer of a MCM.

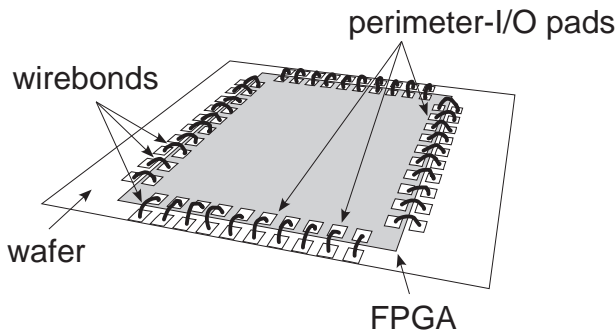


Figure 2. Integration of a perimeter-I/O FPGA in a MCM using wirebonding.

A major implication of the restriction of I/O pads to the perimeter of the chip is that the total number of I/O pads is limited. Due to growing pin requirements of large designs, this number of I/O pads in perimeter-I/O FPGAs is often found to be insufficient.

The obvious solution to this shortage of I/O pads is to use larger FPGAs. This means that the size of an FPGA is no longer determined by the number of CLBs required, but by the number of I/O pads. This phenomenon is called *pin limitation*. It has a negative effect on the performance of an

implementation since:

- the size of the FPGA has to increase to accommodate the required I/O pads, leaving large amounts of silicon unused;
- the length of interconnections in these FPGAs is higher, reducing the speed of an implementation (e.g., lower clock rates).

Another less than preferable solution to the pin limitation problem is to reduce the I/O pad requirements by encoding input and output signals (e.g., serial I/O instead of parallel I/O). In most cases this is only possible by also reducing the speed of an implementation.

A much more promising solution to alleviate pin limitation has recently become possible. It uses the so called *flip-chip* technique to make the electrical connection between the I/O pads on the FPGA chip and the external I/O pads. This technique uses tiny solderbumps to attach a chip (upside down) on a wafer. The I/O pads can now be placed anywhere on the chip, allowing a new FPGA architecture called an *area-I/O FPGA*. Fig. 3 shows how such an area-I/O FPGA could be integrated on a wafer of a MCM.

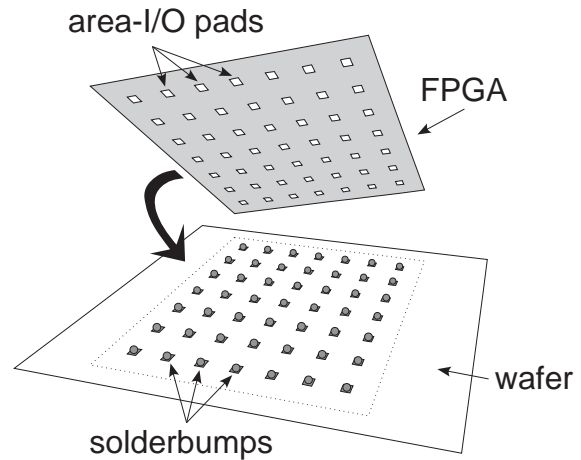


Figure 3. Integration of an area-I/O FPGA in a MCM using flip-chip.

The main advantages of area-I/O are:

- a larger number of I/O pads, eliminating the occurrence of pin limitation problems;
- smaller I/O pads, reducing the area requirements for I/O structures;
- reduced external interconnection lengths;
- reduced routing congestion.

In conclusion, it seems that perimeter-I/O FPGAs are not ideally suited for integration in MCMs due to the pin limitation phenomenon. It seems obvious that area-I/O FPGAs are a better alternative. In the following sections a more quantitative approach will be presented.

3 Analysis of Area-I/O FPGAs versus Perimeter-I/O FPGAs.

The previous section outlined the differences between area-I/O and perimeter-I/O FPGAs. In this section we will elaborate on these differences on a quantitative basis. Particularly, much attention will be paid on the pin limitation phenomenon, overlooked in a previous analysis of area-I/O FPGAs [5]. In fact, we will show that it is the pin limitation problem of perimeter-I/O FPGAs that makes area-I/O FPGAs superior for the use in FPGA MCMs.

We shall proceed as follows. First, we will briefly introduce the models used to quantify our claims. Then we will discuss the pin limitation phenomenon for both perimeter-I/O and area-I/O FPGA architectures. Finally, we will present a detailed quantitative comparison of both architectures.

3.1 Models

3.1.1 FPGA models

FPGAs consist of square lattices of $N \times N$ CLBs. Interconnections between these CLBs are assumed to follow the shortest Manhattan-style path through the routing channels. We express the length of these interconnections as the number of rows and columns between the CLBs. It is reasonable to do so—notwithstanding the fact that (e.g., by introducing area-I/O pads) the actual distance between these rows and columns may slightly vary—since in FPGAs the routing delays are largely determined by the number of programmable interconnections that must be traversed [8]. This number is proportional to the interconnection length expressed in rows and columns. Therefore, from now on, we express all dimensions in rows and columns or, equivalently, in lattice units.

Interfacing to the outside world is done by means of I/O pads in the IOBs. In perimeter-I/O FPGAs, these are located along the perimeter of the chip. We denote the pitch of these I/O pads by d_p (see Fig. 4). For the Xilinx FPGAs (XC30XX and XC40XX series) we find that $d_p = 0.5$ (I/O pad pitch is half the CLB pitch).

An alternative configuration is used in FPGAs with area-I/O. Here the I/O pads (and the IOBs) are distributed on a lattice, covering the whole chip. We denote the pitch of this lattice by d_a (see Fig. 5).

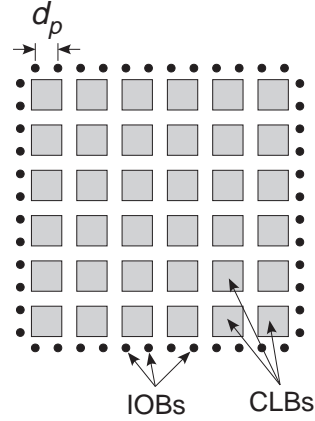


Figure 4. Perimeter-I/O FPGA ($d_p = 0.5$).

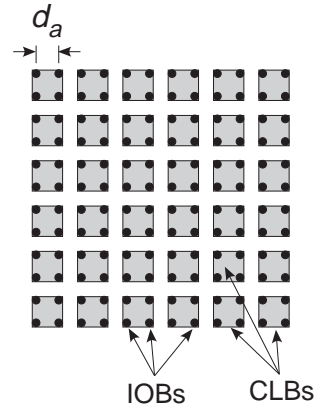


Figure 5. Area-I/O FPGA ($d_a = 0.5$).

3.1.2 Design

At this point, the difference between a *design* and its *implementation* should be appreciated. A *design* is merely a collection of interconnected logic gates. Some properties, like interconnection length, have no meaning for them. The *implementation* of a design, however, is the physical structure that results after placement and routing of the design in a target architecture. It is only then that the aforementioned properties get their meaning.

Before a design can be implemented in a FPGA it needs to be *technology mapped*. This process transforms the design to a collection of logic gates with fanin less than or equal to the fanin of the CLBs of the target FPGA. If needed, it must also be partitioned into smaller subdesigns (e.g., if it can not fit into one FPGA). After that, the design is ready to be placed and routed in the target FPGA. For the remainder of this paper the notation B will be used to denote the number of technology mapped logic gates in a design.

To quantify I/O related issues of FPGAs, it is important to understand what determines the I/O requirements of

a design. A quantitative description is given by *Rent's rule* [4, 9]. In short, Rent's rule states that, when a design is partitioned, there is a relationship between the (average) number of logic gates B_i in a (sub)design, and the (average) number of I/O connections (or *pins*) P_i it requires.

$$P_i = C B_i^r, \quad 0 \leq r < 1. \quad (1)$$

Here C denotes the average number of *terminals* (fanin plus fanout) per logic gate, and r is a constant called the *Rent exponent*. In [6] values of r between 0.47 and 0.75, and values of C between 3 and 5 are said to be observed. Also, r seems to serve as a quantitative measure for the interconnection complexity of a design. A high value of r indicates a complex design. Rent's rule corresponds more or less with the intuitive notion that a complex design requires a larger amount of I/O connections than a simple design. A typical complex design [6] is said to have a Rent exponent $r = 0.65$. An example illustrating Rent's rule on such a design is shown in Fig. 6. Note that if Rent's rule is applied

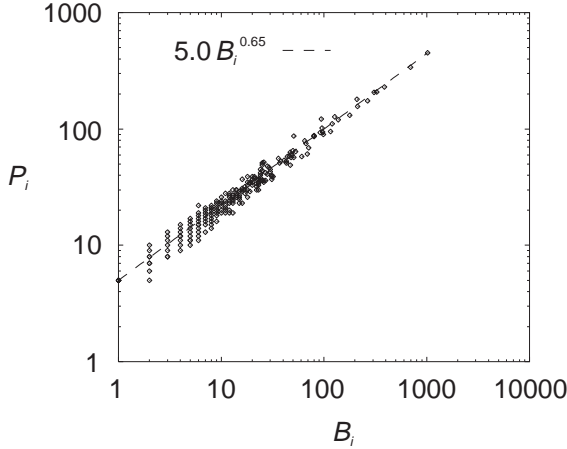


Figure 6. Illustration of Rent's rule showing the relationship between the number of pins P_i and the number of logic gates B_i when partitioning a design ($C = 5$, $B = 1024$, $r = 0.65$).

to the whole design ($B_i = B$), we will use the notation P instead of P_i .

3.2 Pin Limitation in Perimeter-I/O FPGAs

In this section we will show the quantitative impact that placing the I/O pads on the perimeter has on the design implementation size. We will show that the extent of this effect largely depends on the complexity of the design we want to implement.

Consider a design with B logic gates. We want to implement this design in an FPGA with B CLBs, satisfying

the logic block requirements of the design. This means the grid of CLBs in the FPGA has to be square with side N_p :

$$N_p = \sqrt{B}. \quad (2)$$

However, as we explained earlier, in perimeter-I/O FPGAs the number of I/O pads is limited by the perimeter of the FPGA chip. Since the pitch of the I/O pads is d_p , the number of I/O pads is given by

$$4 N_p / d_p. \quad (3)$$

The I/O requirements of a design are determined by its size and its complexity. These requirements are quantitatively described by Rent's rule (Eq. 1). Designs that are more complex (higher Rent exponents r), require more I/O. Above a certain complexity ($r > r_p$), the requirements will eventually exceed the number of available I/O pads on the FPGA (given by Eq. 3), and the implementation will be pin limited:

$$P \geq 4 N_p / d_p, \quad (4)$$

Substituting Eq. 1 and Eq. 2 in Eq. 4, this leads to

$$r \geq r_p = \frac{1}{2} + \log_B \left(\frac{4}{C d_p} \right). \quad (5)$$

This equation is depicted in Fig. 7. It shows us that for large designs (with a high number of logic gates B) the Rent exponent r_p for which the design becomes pin limited lies somewhere around 0.56. Since the Rent exponent r of designs varies from 0.47 and 0.75 it is clear that the complexity of a design plays an important role in the pin limitation behaviour. Consequently, for complex designs, perimeter-I/O chips will frequently suffer from an insufficient number of I/O pads.

For pin limited implementations the size of the FPGA N_p needs to be larger than given by Eq. 2. If the perimeter has to contain the required number of I/O pads, the following equation must be satisfied:

$$P = 4 N_p / d_p. \quad (6)$$

Using Eq. 2 and Eq. 5, and substituting Eq. 1 in Eq. 6, leads to the size N_p of the FPGA needed to provide both the required number of CLBs and the required number of I/O pads:

$$N_p = \begin{cases} \sqrt{B}, & r \leq r_p \\ \frac{d_p}{4} C B^r, & r \geq r_p, \end{cases} \quad (7)$$

depending on whether the implementation is pin limited or not (determined by Eq. 5).

In conclusion, it is clear that pin limitation is likely to occur when complex designs are implemented in perimeter-I/O FPGAs (see Eq. 5 and Fig. 7). As a consequence, the

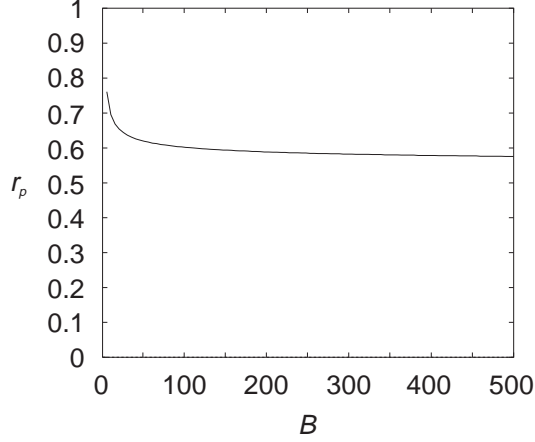


Figure 7. The maximum Rent exponent r_p for which an implementation of a design ($C = 5$) in a perimeter-I/O FPGA ($d_p = 0.5$) is not pin limited, as a function of the design size B .

FPGA size is no longer determined by the number of CLBs needed, but by the required number of I/O pads. We will show later that this has negative repercussions on the performance of such implementations.

3.3 Pin Limitation in Area-I/O FPGAs

We can perform the same calculations in the case of area-I/O FPGAs, to show that pin limitation is not likely to occur in this case, even with complex designs.

Consider again a design with B logic gates. The logic block requirements lead again to a square grid of CLBs in the FPGA with side N_a given by:

$$N_a = \sqrt{B}. \quad (8)$$

Since in area-I/O FPGAs the I/O pads can cover the whole chip with a pitch d_a , the number of I/O pads available is now given by

$$N_a^2 / d_a^2 \quad (9)$$

As with perimeter-I/O FPGAs, above a certain complexity ($r > r_a$), the requirements will exceed the number of available I/O pads on the FPGA (given by Eq. 9), and the implementation will be pin limited:

$$P \geq N_a^2 / d_a^2. \quad (10)$$

Substituting Eqs. 1 and 8 in Eq. 10, this leads to

$$r \geq r_a = 1 - \log_B (C d_a^2). \quad (11)$$

Note that this equation differs fundamentally from the equation determining pin limitation in the case of FPGAs with

perimeter-I/O (Eq. 5). A comparison between the pin limitation behaviour of perimeter-I/O FPGAs and area-I/O FPGAs is shown in Fig. 8. It shows that (for $C = 5$ and

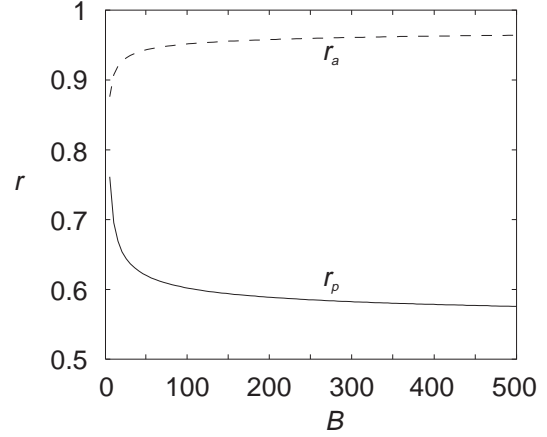


Figure 8. Comparison between r_p and r_a , the maximum Rent exponent for which an implementation of a design ($C = 5$) in an area-I/O FPGA ($d_a = 0.5$) is not pin limited, as a function of the design size B .

$d_a = 0.5$) an area-I/O FPGA implementation will never be pin limited, since the Rent exponent of a design is never as high as 0.9. It must be noted however that area-I/O FPGA implementation could be pin limited if the pitch d_a of the I/O pads is exceptionally high. Fortunately, for modern FPGAs d_a is small enough (e.g., $d_a = d_p = 0.5$) to prevent pin limitation from occurring. However, choosing d_a should be done carefully, ensuring that the number of I/O pads is useful. For example, for a complex design ($r = 0.7$) with $B = 1024$ logic gates and $C = 5$ terminals per logic gate it is sufficient to have (solving Eq. 11 for d_a):

$$d_a < \sqrt{\frac{B^{1-r}}{C}} = 1.2649.$$

Lower values for d_a will only result in area-I/O FPGAs with too many I/O pads, wasting valuable silicon.

In conclusion, area-I/O FPGAs will usually not suffer from pin limitation problems. As a consequence, the size of the FPGA will be determined by the CLB requirements of the designs we want to implement and, therefore, be smaller than the size of a corresponding perimeter-I/O FPGA.

3.4 Comparing the Performance of Area-I/O versus Perimeter-I/O FPGAs.

Using Eq. 11 and Eq. 5 enables us to compare Area-I/O versus Perimeter-I/O FPGAs.

3.4.1 Area Requirements

The area requirements of an implementation in a FPGA consist of three parts:

- the area requirements for the CLBs;
- the area requirements for the routing resources;
- the area requirements for the I/O structures.

As long as the implementations are not pin limited, the area requirements for the CLBs are the same for area-I/O and perimeter-I/O FPGAs, determined by the CLB requirements of the design. If, however, the (perimeter-I/O) implementation is pin limited, the size of the FPGA is dictated by the I/O requirements of the design. Note that this does not necessarily imply that the actual number of CLBs is higher. The extra silicon area could be left unused. Still, it is easier to include this area with the CLB area requirements.

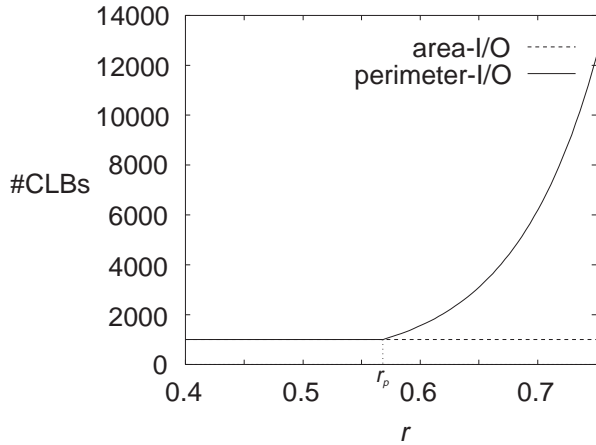


Figure 9. Area requirements (number of CLBs) for a perimeter-I/O FPGA ($d_p = 0.5$) versus an area-I/O FPGA ($d_a = 0.5$), as a function of the Rent exponent r of the design the FPGA needs to hold.

For the routing resources, things are a little bit more complicated. Due to the fact that in area-I/O FPGAs the IOBs are spread across the whole chip, we shall see later on that the routing requirements can be expected to be lower. On top of that, if a perimeter-I/O FPGA implementation is pin limited, the total amount of routing is also increased. Overall, the area requirements of the routing resources will be lower for area-I/O FPGAs, but this effect is on a much smaller scale than the differences in CLB area requirements.

The same applies to the area requirements of the actual I/O structures. We can assume that the number of I/O pads (both in the perimeter-I/O and the area-I/O case) equal the

number required by the design (by choosing the appropriate N_p and d_a). Since, the I/O pads for flip-chip (in area-I/O FPGAs) can be expected to be a bit smaller than the I/O pads for wirebonding (in perimeter-I/O FPGAs), the area requirements for the I/O pads will be slightly lower in the area-I/O FPGAs.

So, for our purpose, the area requirements of FPGAs can be approximated by the area needed for the CLBs (using Eq. 8 and Eq. 7). Fig. 9 shows a comparison between the area requirements (expressed in number of CLBs) for a perimeter-I/O and an area-I/O FPGA. It is clear that area-I/O implementations are only useful for complex designs ($r > r_p$).

3.4.2 Interconnection length

Much of the research on interconnection length estimation is based on work by Donath [1, 2]. It shows the relationship between the Rent exponent r of a design and the average interconnection length \bar{L} of its implementation in a square grid. The results presented in this section are based on our extensions of this work. For a more detailed discussion of the subject we refer to [7].

Fig. 10 shows the average interconnection length \bar{L} for perimeter-I/O and area-I/O FPGAs (for $C = 5$, $B = 1024$ and $d_p = d_a = 0.5$), as a function of the Rent exponent r of the design. Note that, as long as the perimeter-I/O im-

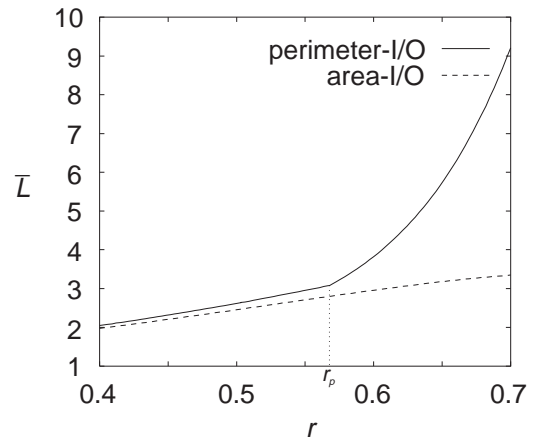


Figure 10. Average interconnection length \bar{L} for the implementation of a design ($C = 5$, $B = 1024$) in a perimeter-I/O FPGA ($d_p = 0.5$) and an area-I/O FPGA ($d_a = 0.5$), as a function of the Rent exponent r of the design.

plementation is not pin limited ($r \leq r_p$), the average interconnection length \bar{L} is slightly lower in the area-I/O case. This is due to the fact that the external interconnections (to the IOBs) are on average shorter. As soon as the perimeter-

I/O implementation is pin limited ($r > r_p$), the size of the FPGA is determined by the I/O requirements of the design. This results in a dramatic increase of \bar{L} , since both the internal (between CLBs) and the external (between CLB and IOB) are stretched in the larger perimeter-I/O FPGA. An example of such a pin limited implementation is shown in Fig. 11. In order to keep the interconnection length as low

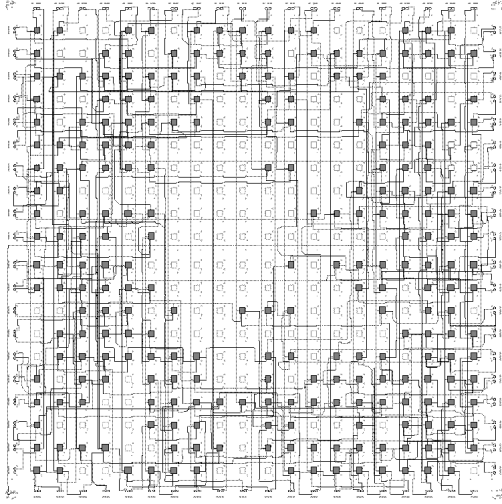


Figure 11. Pin limited implementation of a design ($r = 0.66$, $B = 256$, $C = 4$) in a perimeter-I/O FPGA (Xilinx XC4010), showing the “doughnut” shape of the used CLBs.

as possible the placement of the design results in the typical “doughnut” shape.

To compare both architectures, we define the relative interconnection length gain Γ as

$$\Gamma = \frac{\bar{L}_p - \bar{L}_a}{\bar{L}_p}, \quad (12)$$

where \bar{L}_p (\bar{L}_a) is the average interconnection length for a perimeter-I/O (area-I/O) implementation. Fig. 12 shows the gain Γ as a function of the Rent exponent r of the design. It clearly shows that, for complex designs only, area-I/O FPGA implementation results in large gains in interconnection length. For simple designs the gain is only marginal.

3.4.3 Routing Requirements

The routing requirements of a design are determined by the number of interconnections \bar{W} per routing channel. The average value \bar{W} can be calculated by dividing the average interconnection length by the number of routing channels in the FPGA. As such, the routing requirements are a combination of the area requirements of a design and the interconnection length of its implementation.

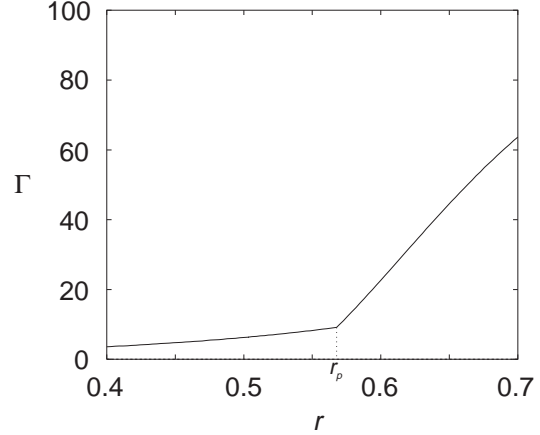


Figure 12. Relative interconnection length gain Γ (in %) of an implementation of a design ($C = 5$, $B = 1024$) in an area-I/O FPGA over a perimeter-I/O FPGA implementation, as a function of the Rent exponent r of the design.

Fig. 13 shows the average number of interconnections per routing channel \bar{W} , as a function of the Rent exponent r of a design. It shows that, as long as the perimeter-I/O implementation is not pin limited ($r \leq r_p$), the average number of interconnections per routing channel \bar{W} is slightly lower in the area-I/O case. This is again due to the fact that the external interconnections (to the IOBs) are on average shorter, consequently reducing the routing requirements. As soon as the perimeter-I/O implementation is pin limited ($r > r_p$), the picture changes. The increased number of routing channels reduces the routing requirements. Note that due to the “doughnut” shaped placement (see Fig. 11) the routing requirements in a pin limited perimeter-I/O FPGA are unevenly distributed.

4 Conclusions

Our analysis shows that providing FPGAs with area-I/O for integration in MCMs is a promising idea for the implementation of complex designs. Compared with traditional perimeter-I/O FPGAs, we demonstrated significant reductions in average interconnection length and area requirements. This is mainly due to the fact that area-I/O resolves the pin limitation problems of complex designs. For simple designs the gains are only marginal.

There are, of course, various cost factors that hinder or prevent the production of area-I/O FPGAs in the foreseeable future. Nevertheless, the maturing flip chip technology and, above all, the fact that the increased number of I/Os in area-I/O FPGAs enables complex designs to be

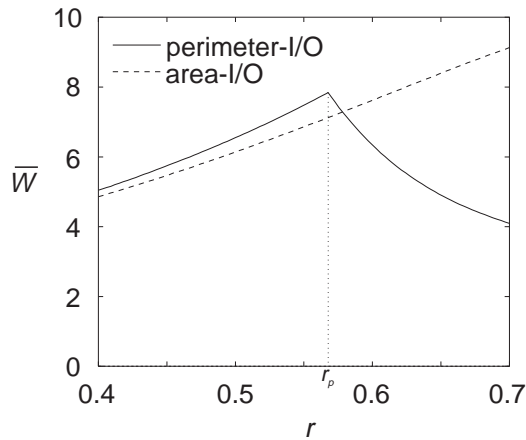


Figure 13. Average number of interconnections per routing channel \bar{W} for an implementation of a design ($C = 5$, $B = 1024$) in a perimeter-I/O FPGA ($d_p = 0.5$) and an area-I/O FPGA ($d_a = 0.5$), as a function of the Rent exponent r of the design.

implemented in large Multi-Chip Modules that are substantially more performant, in time, will make area-I/O FPGAs a valuable alternative to consider.

Acknowledgements

This research is carried out at the Department of Electronics and Information Systems at the University of Ghent, Belgium. The work of H. Van Marck and J. Depreitere is supported by an Inter University Attraction Poles research program (IUAP IV-13) on Photonic Interconnects initiated by the Belgian Government, Prime Minister's Service, Science Policy Office. D. Stroobandt is Research Assistant with the Fund for Scientific Research of Flanders, Belgium.

References

- [1] W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Trans. on Circ. & Sys.*, CAS-26:272 – 277, 1979.
- [2] W. E. Donath. Wire length distribution for placements of computer logic. *IBM Journal of Research and Development*, 25:152 – 155, 1981.
- [3] N. Howard, A. Tyrrell, and N. Allinson. The yield enhancement of Field-Programmable Gate Arrays. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(1):115–123, March 1994.
- [4] B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Trans. on Computers*, C-20:1469 – 1479, 1971.
- [5] V. Maheshwari, J. Darnauer, J. Ramirez, and W. Dai. Design of FPGA's with area I/O for field programmable MCM. In *Proceedings of the 1995 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 17–23, February 1995.
- [6] R. L. Russo. On the tradeoff between logic performance and circuit-to-pin ratio for LSI. *IEEE Trans. on Computers*, C-21:147 – 153, 1972.
- [7] D. Stroobandt, H. Van Marck, and J. Van Campenhout. An accurate interconnection length estimation for computer logic. In B. Werner, editor, *Proceedings of the 6th Great Lakes Symposium on VLSI*, pages 50–55, Los Alamitos, California, March 1996. IEEE Computer Society Press.
- [8] S. Trimberger. *Field-Programmable Gate Array Technology*. Kluwer Academic Publishers, 1994.
- [9] H. Van Marck, D. Stroobandt, and J. Van Campenhout. Towards an extension of Rent's rule for describing local variations in interconnection complexity. In S. Bai, J. Fan, and X. Li, editors, *Proceedings of the Fourth International Conference for Young Computer Scientists*, pages 136–141. Peking University Press, 1995.