# A Case Study on Rapid Prototyping of Hardware Systems: the Effect of CAD Tool Capabilities, Design Flows, and Design Styles

Apostolos Dollas Kyprianos Papademetriou Euripides Sotiriades
Dimitrios Theodoropoulos Iosif Koidis George Vernardos

Electronic and Computer Engineering Department
Technical University of Crete
Chania, 73100
Greece

Contact author: dollas@mhl.tuc.gr

## Abstract

*CAD tools are necessary for even the most rudimentary hardware designs when the target technology is FPGA or VLSI. This paper presents an experiment in the use of vendor-supplied vs. third-party tools, of high-level design (behavioral) vs. cell-based design, of technology tradeoffs among different vendors, and of design styles among designers. The experiment is a n-version hardware design project, in which many designs were made from common specifications, using different CAD tools and design styles. Some interesting conclusions can be drawn regarding the process, and these conclusions show new trends in terms of tool capabilities and desirable design flows.*

**KEYWORDS:** Hardware design, FPGA, CAD tools, Design flow.

## 1. Introduction and Motivation

In 1992, we had reported on a case study regarding the process of rapid system prototyping with incomplete CAD tools and inexperienced designers [1]. The conclusions were that rapid prototyping was indeed feasible with minimal CAD tools and inexperienced designers. The biggest challenges stated in that work for the (then) future were:

- Better subsystem reusability
- Better system level simulation capability, and,
- Better learning curve of the designers in developing new methodologies.

Over a decade later many parameters have changed. The omnipotence of CAD tools would make naïve any claim of non-trivial hardware design without a reasonable CAD tool suite. Hardware description languages are the dominant way to do hardware design, and simulation capabilities have progressed by leaps and bounds. Reusability has taken a life of its own with *intellectual property cores* and *parameterizable libraries*. Still, design styles and choice of design flows can affect the end result in a major way.

Based on these major changes in the design environment capabilities, it was time to revisit the experiment and do a new case study. Unlike the 1992 work, this case study is exclusively based on CAD tools, because this is how design is done today. A set of designs from common specifications were completed up to the point of post-place and route, including full timing analysis, and for different implementation vendor technologies. The designs were done by five engineers, all graduate students, with different levels of experience, ranging from two to seven years. Each designer completed several versions of the design and recorded a number of relevant results, including the quality of the design (in speed and area), the computer time that the CAD tools required to complete the design and their capability of compiling the design etc. These results were tabulated for the purpose of drawing conclusions.

In the process of this case study a large number of tools were used, including vendor-supplied tools from two major FPGA vendors, third party synthesis tools from two major vendors, a simulation tool from a major vendor, etc. Most designs were completed in VHDL but there was a set of experiments done in Verilog. As the purpose of this study was to determine trends, and because CAD tool capabilities change constantly, the specific brand names and release versions will not be mentioned in this paper, only their main characteristics (e.g. commercial synthesis tool). It should be noted though, that these are the latest (or near latest in some cases) releases of commercially distributed tools.

Section 2 briefly presents the experiment which we conducted and the parameters that were evaluated. Section 3 has the comparison between the two basic design styles. It presents results from experiments with multiple FPGAs of two different vendors. Section 4 describes the effects of synthesis CAD tools. Section 5 illustrates how the design style of each engineer effects on the quality of the results. Finally, Section 6 presents conclusions from this work.

## 2. Description of the Experiment

The design chosen for this work should lend itself to multiple instances, so that each designer would complete a number of different versions. It should be scalable, so that CAD tools would be put to test for small-scale vs. large-scale designs from substantially unchanged input files. It should also offer some regularity so that the tradeoff of hand placement vs. fully automatic design could be evaluated. Lastly, the end design should be understandable for purposes of analysis, so that the designers would have a very clear view of how an optimal design would be in terms of resources, and whether the tools achieved it or not. A problem with these characteristics is the Game of Life [3]: we have a two-dimensional array in which every cell gets its next value (0,1) as a function of its current value and a threshold of the 1's in its eight neighbors. Three versions of the specifications were made, depending on the size of the array:

- GL10 for a 10X10 array
- GL20 for a 20X20 array, and
- GL30 for a 30X30 array.

No wraparound of the edges was assumed, but in each case border cells with fixed values were assumed to be in place through hardwiring.

In terms of CAD tools, these tools were used in various combinations:

- ST1: Synthesis Tool of FPGA Vendor A
- ST2: Synthesis Tool of FPGA Vendor B
- ST3: Commercial Synthesis Tool of Vendor C
- ST4: Commercial Synthesis Tool of Vendor D
- PR1: Place and Route Tool of FPGA Vendor A
- PR2: Place and Route Tool of FPGA Vendor B
- SIM: Commercial Simulation Tool

Multiple FPGAs of two different major vendors were used as the target technology. The selection of the devices was based on the size of the technology. Four sizes were selected: 0,13um, 0,15um, 0,18um and 0,22um. The choice of feature size, given the typical die size for each manufacturing process, yields great similarity of available gates, and therefore is related to the usable logic area in each chip for FPGA vendors A and B.

- FPGA_A_013: 0,13um FPGA of Vendor A
- FPGA_A_015: 0,15um FPGA of Vendor A
- FPGA_A_018: 0,18um FPGA of Vendor A
- FPGA_A_022: 0,22um FPGA of Vendor A
- FPGA_B_013: 0,13um FPGA of Vendor B
- FPGA_B_015: 0,15um FPGA of Vendor B
- FPGA_B_018: 0,18um FPGA of Vendor B
- FPGA_B_022: 0,22um FPGA of Vendor B

The designers were categorized by their years of experience:

- D1: 7 years of experience
- D2: 7 years of experience
- D3: 3 years of experience
- D4: 3 years of experience
- D5: 2 years of experience

All designs had the constraints of a global reset, one clock input, unconstrained pin mapping, and a single clock cycle per result. Having the above in mind we will evaluate the experimental results with respect to parameters such as design style, synthesis tool, FPGA vendor and the combination between them.

## 3. Behavioral vs. Cell-based Design

A common approach for years has been that datapath is better done in structural design for better usage of the resources (including manual placement), whereas the control path is best done in behavioral design for easier changes. Several versions of the system were designed in both behavioral and cell-based HDL (VHDL or Verilog) designs. The term cell-based has the meaning of the design in which the functionality of the cell was constructed with glue logic and then repeated ("generate" function) into a structural code.

The more experienced engineers D1, D2 wrote cell-based HDL codes, while D3, D4, D5 preferred behavioral design style. Figures 1 and 2 present the results of the designs of all five engineers that were implemented by every synthesis tool and for both FPGA vendors. The bars of the graphs correspond to the five designs. Each group of bars corresponds to a synthesis tool and an FPGA device. Figure 1 has the results of designs GL10 and GL30 implemented in the FPGAs of Vendor A and Figure 2 has the same results for FPGA Vendor B.

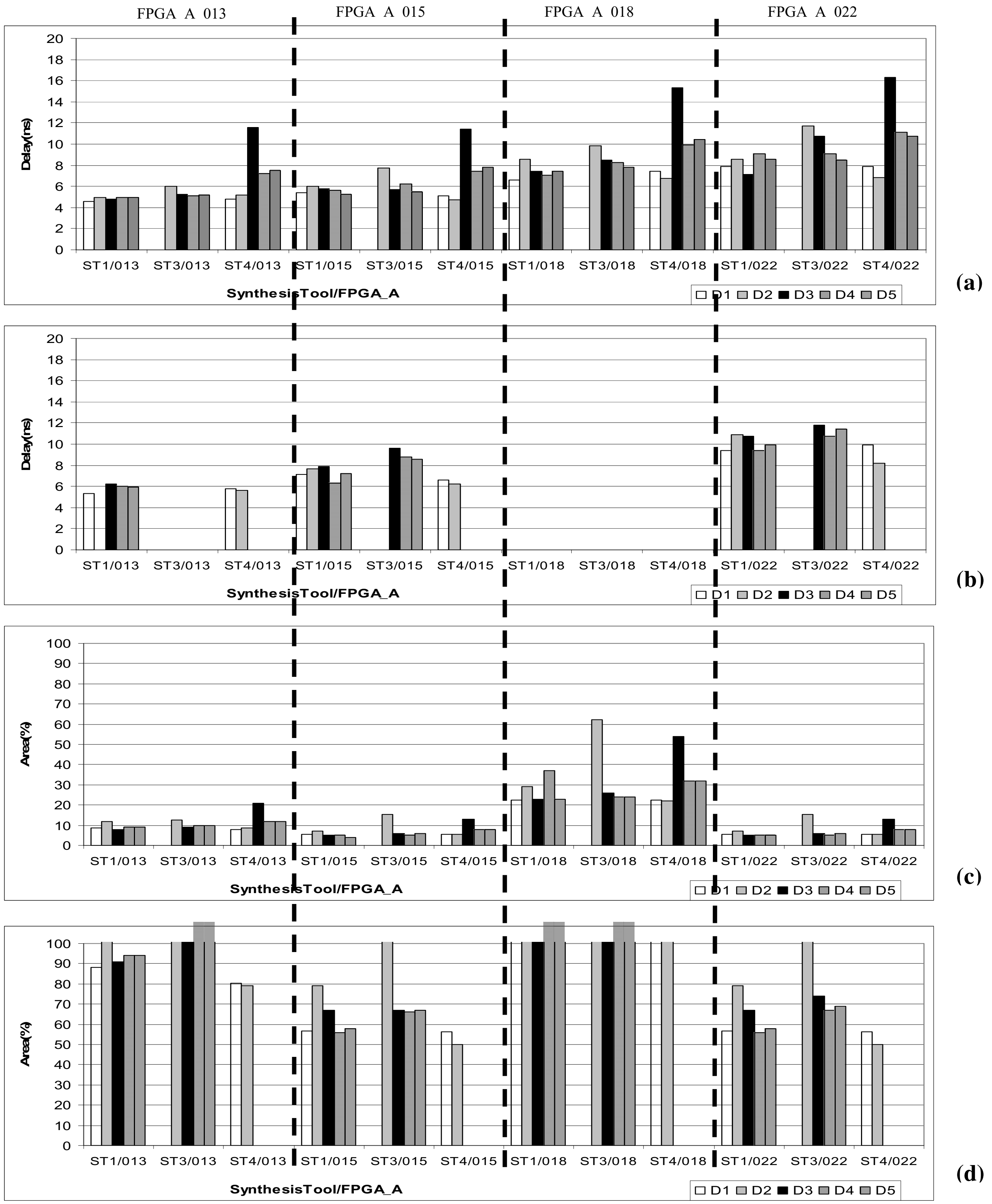

**Figure 1: Implementation of the five designs in FPGAs of Vendor A: (a) and (b) shows the speed results for GL10 and GL30 respectively, (c) and (d) shows the area results for GL10 and GL30 respectively.**

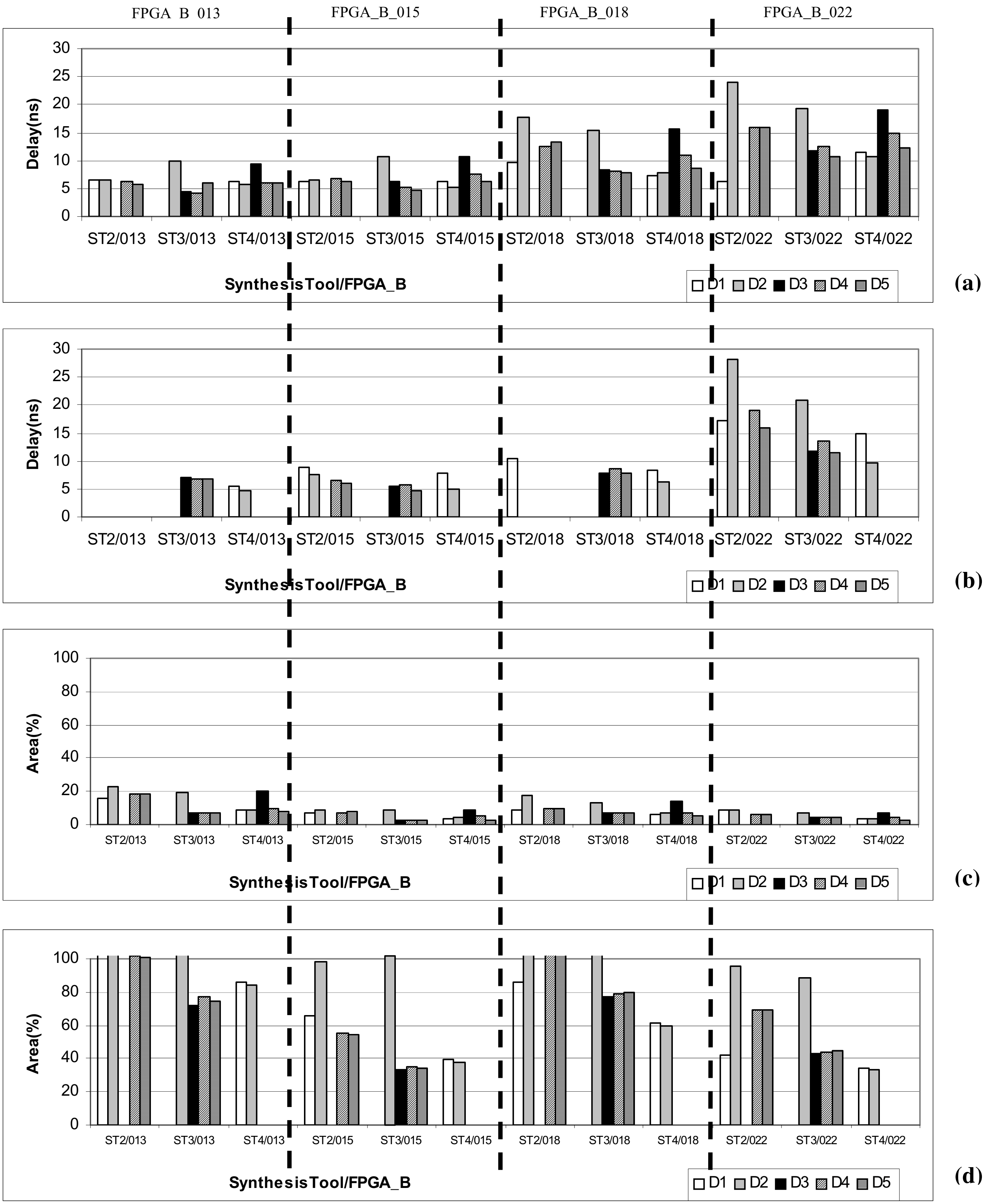

**Figure 2: Implementation of the five designs in FPGAs of Vendor B: (a) and (b) shows the speed results for GL10 and GL30 respectively, (c) and (d) shows the area results for GL10 and GL30 respectively.**

As shown in Figure 1 the use of synthesis tool ST1 has had a similar effect in all five designs. The use of ST3 affected in a different way the designs. More specifically the speed and area results between the behavioral designs are more-or-less similar and are quite satisfactory. On the other hand the implementation of D2's cell-based design produced (quite unexpectedly) the worst results. Moreover, the compilation of D1's design was not completed, due to a bug in the synthesis tool ("internal error", confirmed as a bug by the tool vendor). The implementation of the cell-based designs with ST4 gave similar results in all cases. The speed and area results were very satisfactory. The post place and route results of D4, D5 with ST4 were worse than the cell-based designs, while the results of D3 were the worst.

Figure 2 shows the results for FPGA Vendor B. The use of ST2 synthesis tool affected the designs in a strange manner. In some cases the less experienced designers D4, D5 produced better results than the rest, while in other cases D1 gave the best results. In every case D2's design had the worst results and D3's design couldn't be compiled due to a tool's "fatal error". The quality of all five designs when they were compiled with ST3 was effectively identical for either FPGA vendor as the target technology. The speed and area results between the behavioral designs are more-or-less similar and quite satisfactory. The implementation of the cell-based design of D2 produced the worst results with ST3. Finally, the results of cell-based designs that were synthesized with ST4 were slightly better than the behavioral designs. The compilation of the behavioral designs for size GL30 couldn't be completed as the tool got stuck on the mapping stage of PR2. Moreover, in every case the synthesis of D3's design with ST4 has had the worst results.

We observe in the graphs of both Figures that the peaks of each group of bars correspond to D2's design synthesized with ST3 and D3's design synthesized with ST4. Moreover, for FPGA Vendor B we observe that for most cases the peak belongs to D2's design when synthesized with ST2. These are the worst cases. Having the above in mind we draw the following conclusions between the two design styles:

*Conclusion (1): We may have similar results between behavioral and structural designs for some of the tools* e.g. ST1 tool produces similar results for all designs.

*Conclusion (2): Depending on the synthesis tool the designer must choose the appropriate design style (behavioral or structural)* e.g. for both FPGA Vendors and for every GL size the designs of D1, D2 produce much better results with ST4 than D3, D4, D5 do, while D3, D4, D5 produce much better results with ST3 than D1, D2 do. This conclusion was unexpected and merits some discussion. We see here that combinations of tools consistently perform better (or worse) than other combinations of tools, *vis a vis* the design style (behavioral vs. cell-based). In practice this means that certain design flows are practically guaranteed to be consistently better (or worse) than others. We will revisit this issue in Section 5.

*Conclusion (3): Compilation of structural designs produces results (although the design may potentially not fit in the target device) in contrast to behavioral design which may not pass the compilation of a tool* e.g. designs of D3, D4, D5 for GL30 can't be compiled by ST4 while compilation of D1, D2 was finished even for the FPGA device of Vendor A in which they did not fit.

## 4. The Effect of Synthesis Design Flows

One of the general assumptions to date is that third-party synthesis tools are much better than vendor supplied tools, and should be used if at all possible. It was therefore interesting to see what these tools could do, what was the quality of the results, and how the design time was affected.

| *Technology size* | *Device* | *Equivalent area units* | *Area ratio* |
|---|---|---|---|
| 0,13um | FPGA_A_013 | 11.891 | 1,072 |
| | FPGA_B_013 | 11.088 | |
| **0,15um** | **FPGA_A_015** | **1.900.000** | **0,950** |
| | **FPGA_B_015** | **2.000.000** | |
| 0,18um | FPGA_A_018 | 262.912 | 0,267 |
| | FPGA_B_018 | 985.882 | |
| 0,22um | FPGA_A_022 | 1.052.000 | 0,936 |
| | FPGA_B_022 | 1.124.000 | |

**Table 1: Equivalence of capacity between the FPGA devices of Vendors A and B.**

In order to compare the overall quality of third-party synthesis tools with FPGA vendor place and route tools vs. the FPGA vendors' tools alone, we selected two devices of the same technology feature size, one from each vendor, that have almost equivalent available logic. Table 1 has the comparison of the devices. The "Equivalent area unit" corresponds to the capacity of the device. It is expressed as the amount of the total system gates or logic units ("elements", "cells", "CLBs") of the device. The "Area ratio" is the ratio of area of FPGA_A to FPGA_B. We selected to compare the 0,15um devices because of their almost equivalent area. As it is shown their "Area ratio" is the closest to 1. Their difference in capacity is within 5%.

Figure 3 has the results of the six combinations of place-route and synthesis tools that were used by every designer, for the 0,15um FPGAs of the two vendors. The bars of the graphs correspond to the six combinations of the tools. Each group of bars corresponds to a designer and the two FPGAs. Figure 3(a) has the speed results and Figure 3(b) has the area results for all GL sizes.

1074-6005/04 $20.00 © 2004 IEEE

The use of synthesis tool ST3 in combination with PR2 produces much better results than the combination with PR1. That is observed for all designs except D1 and for all GL sizes. It has been mentioned already that ST3 did not successfully compile the design of D1. Also, ST4 has a higher impact on PR2 vs. PR1. The vendor-supplied tool ST1 produces significantly better results than the vendor-supplied tool ST2. The latter tool could not compile the design of D3. In conclusion we observe that the ST1 FPGA vendor tool is competitive to the third party tools ST3 and ST4, whereas FPGA vendor tool ST2 has consistently worse results than third party tools. We conclude that:

*Conclusion (4): Third-party synthesis tools produce much better results with PR2 (place and route tool of FPGA Vendor B) than with PR1 (place and route tool of FPGA Vendor A)* e.g. ST3 and ST4 tools work better with PR2 than with PR1.

*Conclusion (5): Vendor-supplied synthesis tool of FPGA Vendor A has better results than vendor-supplied synthesis tool of FPGA Vendor B.*

The combination of the two conclusions above means that third party tools are more important to be used with one FPGA vendor's technology than the other's.

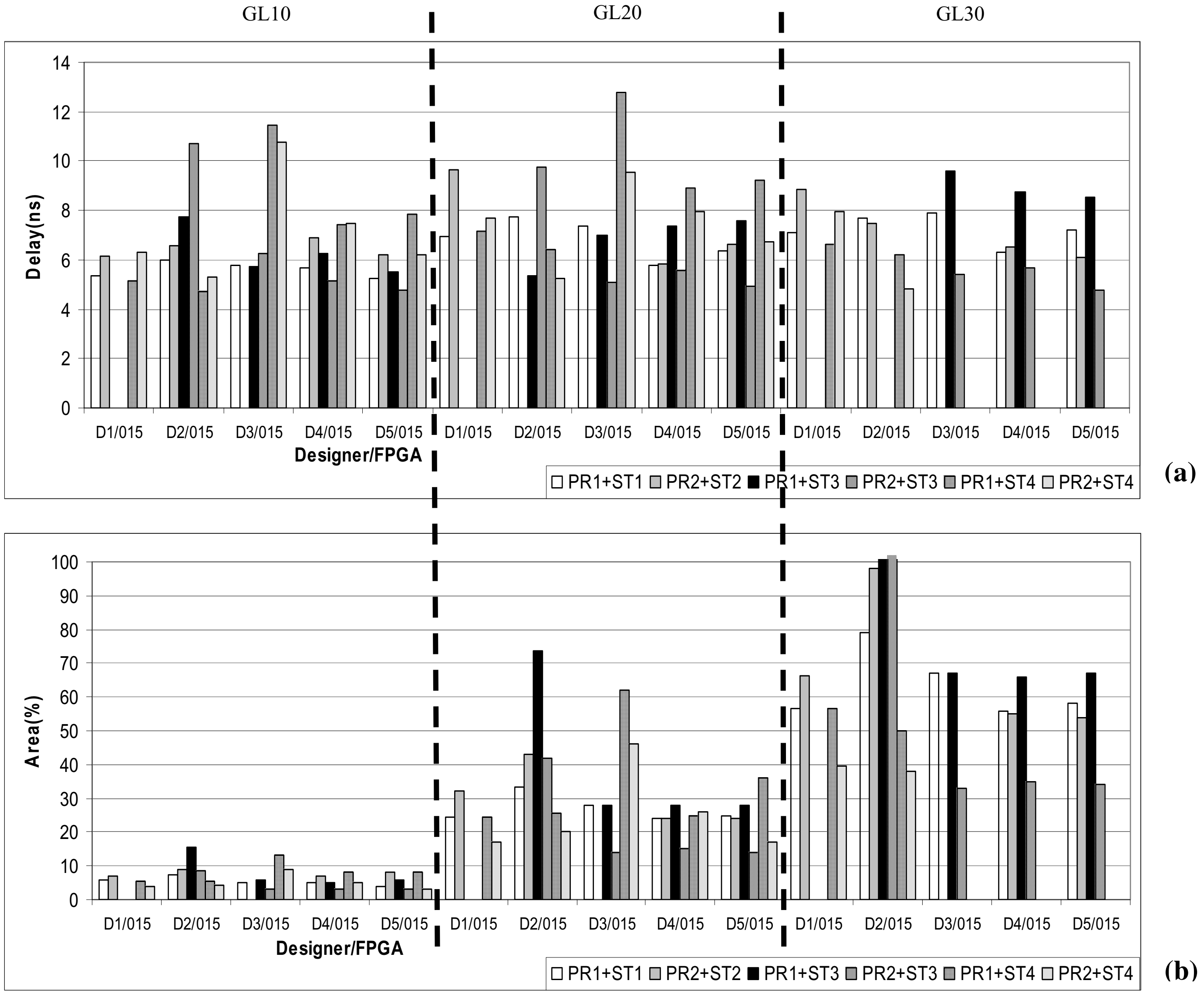

**Figure 3: Implementation of the five designs for the three GL sizes in 0,15um FPGAs of the two vendors: (a) shows the speed results, (b) shows the area results.**

*Conclusion (6): CAD Tools are not fully reliable even for simple designs* e.g. ST3 was not able to compile D1's design, ST2 was not able to compile D3's design, place and route tools got stuck during compilation of the GL30's behavioral designs.

*Conclusion (7): Third-party synthesis tools produce much better results with PR2 of FPGA Vendor B than the vendor-supplied synthesis tool does. FPGA Vendor's A-supplied synthesis tool works quite well.*

*Conclusion (8): Depending on the design style (behavioral or structural) the designer must choose the appropriate synthesis tool* e.g. cell-based designs work much better under ST4 than behavioral designs do, while the latter work much better with ST3 than cell-based designs. This conclusion is the same as conclusion (2) but from tool vs. design style perspective.

## 5. Designer Style Effects

One very interesting comparison to make was that of the designer style. More experienced designers generally preferred structural designs, probably as a result of what they were taught many years ago to be a sound practice, as well as their own self-confidence. Less experienced designers were more dependent on CAD tools to do a good job, and generally preferred behavioral (or higher level at any rate) designs. Interestingly enough, many behavioral designs ended up better than many of the structural designs, as a result of CAD tool quality.

In Section 3 we compared the results of behavioral vs. cell-based designs. We showed that the behavioral designs in many cases produce better results than cell-based designs, when the appropriate synthesis tool is used. We drew that conclusion after observation of Figures 1 and 2. In addition, Table 2 has best-case results of all designs for all GL sizes. It is shown that all behavioral designs produce quite good results, comparable to or better than cell-based designs, *but only if they were implemented with the appropriate synthesis tool in each case.* However, the comparison between the three behavioral designs indicates that implementation of the design of D3 has much worse results than those of D4 and D5 when synthesized by ST4. Moreover D3's design could not be compiled by ST3 at all. By contrast, the comparison between the cell-based designs indicates that the design of D2 produces much worse results with ST2 than D1's. We conclude that:

*Conclusion (9): The implementation of designs of less experienced engineers may have good results when the appropriate synthesis tools are used e.g.* implementation of the designs of D3, D4, D5 has much better results with ST3 than the designs of D1, D2 do.

*Conclusion (10): It is possible to observe substantial differences in the results between behavioral designs that are implemented with the same CAD tool* e.g. the design of D3 produces much worse results with ST4 than those of D4, D5.

*Conclusion (11): It is possible to observe substantial differences in the results between structural designs that are implemented with the same CAD tool* e.g. the design of D2 produces much worse results with ST2 than the design of D1.

## 6. Conclusions and Future Work

The specific results in the previous sections could be considered as tool-specific and therefore unimportant to the design community. Nonetheless, several trends emerge, which would certainly be useful to designers and educators alike. During the last few years the available CAD tools have been substantially improved, to the point of having direct comparisons between behavioral vs. structural designs, with behavioral designs often coming ahead. Nonetheless, it would be premature to proclaim the death of structural designs: all cases of incomplete designs due to

| | | | *GL10* | *Device/Tools* | *GL20* | *Device/Tools* | *GL30* | *Device/Tools* |
|---|---|---|---|---|---|---|---|---|
| *Cell-based Designs* | *D1* | *Delay (ns)* | 4,623 | A013/ST1+PR1 | 4,941 | A013/ST1+PR1 | 5,352 | B013/ST4+PR2 |
| | | *Area (%)* | 3,39 | B022/ST4+PR2 | 14,95 | B022/ST4+PR2 | 34,60 | B022/ST4+PR2 |
| | *D2* | *Delay (ns)* | 4,72 | A015/ST4+PR1 | 5,036 | B013/ST4+PR2 | 4,801 | B013/ST4+PR2 |
| | | *Area (%)* | 3,72 | B022/ST4+PR2 | 17,10 | B022/ST4+PR2 | 33 | B022/ST4+PR2 |
| *Behavioral Designs* | *D3* | *Delay (ns)* | 4,549 | B013/ST3+PR2 | 5,072 | B015/ST3+PR2 | 5,405 | B015/ST3+PR2 |
| | | *Area (%)* | 3 | B015/ST3+PR2 | 14 | B015/ST3+PR2 | 33 | B015/ST3+PR2 |
| | *D4* | *Delay (ns)* | 4,174 | B013/ST3+PR2 | 5,373 | A013/ST1+PR1 | 5,653 | B015/ST3+PR2 |
| | | *Area (%)* | 3 | B015/ST3+PR2 | 15 | B015/ST3+PR2 | 35 | B015/ST3+PR2 |
| | *D5* | *Delay (ns)* | 4,752 | B015/ST3+PR2 | 4,912 | B015/ST3+PR2 | 4,788 | B015/ST3+PR2 |
| | | *Area (%)* | 3 | B015/ST3+PR2 | 14 | B015/ST3+PR2 | 34 | B015/ST3+PR2 |

**Table 2: Best-case results of all designs for every GL size.**

CAD tools that got stuck come from behavioral designs. We therefore have a situation in which the results are either quite good, or not acceptable altogether. It should not be considered though that structural designs are without problems: one verified third party vendor tool bug was found in a structural design. Hence, CAD tools are great in general but they are far from sufficiently robust or sufficiently consistent in terms of their quality of results.

An additional surprising conclusion was that third party CAD tools consistently outperformed one FPGA vendor's tools but consistently matched the other FPGA vendor's tools. As a general observation we could note that FPGA vendor's tools are quite competitive to third party tools, which was not the case a few years ago.

We also noticed that specific design flows comprising of designer style (behavioral vs. structural), third party vs. FPGA vendor tools, and choice of FPGA vendor consistently produced better results. This conclusion was also somewhat surprising because they suggest that no matter what tool one chooses (FPGA vendor or third party) the quality of the resulting designs may well be related to the style of the designer (i.e. there are no consistently better tools, third party or otherwise).

Lastly, the combination of conclusions 10 and 11 means that the well-known adage that design style matters still holds today as much as it ever did.

In terms of future work, this study could be extended to less regular designs (e.g. large IP cores), in which notions of optimality of a design would not be possible to verify, but which may be more challenging designs to demonstrate tool capabilities.

## References

[1] A. Dollas, *Experimental Results in Rapid System Prototyping with Incomplete CAD Tools and Inexperienced Designers,* Proceedings, Second International IEEE Workshop on Rapid System Prototyping RSP-91, pp. 9-16, Computer Society Press, 1992.

[2] Y. Li, W. Chu, *Aizup – A Pipelined Processor Design and Implementation on Xilinx FPGA Chip*, Proceedings, 4th International Symposium on FPGA's for Custom Computing Machines (FCCM), pp. 98-106, Computer Society Press, 1996.

[3] M. Gardner, *Mathematical Games: the Fantastic Combinations of John Conway's New Solitary Game of Life,* in Scientific American, October 1970, pp. 120-123.