Loop Controller Area Estimation for Automatic Design Space Exploration

Tom Degryse

Supervisor(s): Dirk Stroobandt

Abstract— **High-level synthesis systems overcome the high design effort required to program an FPGA by translating an algorithm at the behavioral level into a synthesizable hardware description. At this higher level, loop transformations are used to improve the characteristics of the program. These transformations have a great impact on the resulting hardware, only known after the time-consuming synthesis steps. This hinders a fast design space exploration.**

In this work, we tackle this issue by estimating the performance of the hardware loop controller, an often overlooked component in other approaches. We present an equation based model to estimate the area of the loop controller during high-level synthesis. The presented approach is accurate enough to be useful during design space exploration. Due to its simplicity, the overhead of the estimations is minimal. The proposed methodology can easily be adapted to new FPGA design flows and architectures.

Keywords— **FPGA, Area estimation, High-level synthesis, Loop controller**

I. Introduction

FPGAs are widely used as hardware accelerators in todays embedded systems. A large class of algorithms can benefit from a hardware implementation thanks to the high degree of parallelism that is available inside an FPGA. However, the translation from a sequential software description into a hardware description language, such as VHDL, is a time-consuming and error-prone process. The introduction of parallelism is only one of the issues that the designer needs to overcome. To shorten the design time, the hardware design process is moved to a higher abstraction level. A high-level synthesis tool is then used to translate this high-level program representation into VHDL.

Programs are transformed at this higher abstraction level to obtain improved hardware implementations. One can for instance think about the automatic introduction of parallelism. These high-level transformations have a great impact on the quality of the resulting hardware, but their impact is only known after executing all the time-consuming synthesis steps. During design space exploration, a large number of different design variants is generated. As a result, one can not fully implement every possible solution during design space exploration.

This issue can be overcome by estimating the performance of the hardware implementation directly at the higher abstraction level. These estimations have to be very fast, because of the large number of design variants that need to be evaluated. Moreover the accuracy needs to be fairly good, so that the relative order of the design variants is maintained.

In this paper, we present a simple equation based model to estimate the FPGA resources needed to implement the loop controller. The loop controller is an often overlooked component in other estimation techniques, like [3] and [4], but it can have a great impact on the resulting hardware. We use the polyhedral model as the starting point for our estimations. Our area estimation model is fairly accurate, with an average relative error of not more then 8%. Moreover, we managed to keep the estimation overhead very low. As a result, this work is a step towards an automated design space exploration framework.

The rest of this paper is organized as follows. In the next section, a brief introduction to the polyhedral model is given. The paper continues with our proposed area estimation model in section III. Section IV presents the experimental results and section V summarizes this work and gives some future directions.

T. Degryse is with the Electronics and Information Systems Department, Ghent University (UGent), Gent, Belgium. E-mail: Tom.Degryse@elis.UGent.be .

II. Polyhedral model

In this work, we use the polyhedral model as an intermediate representation for the automatic hardware generation. Within this model, a single execution of a statement inside a loop or a loop nest is represented as a point in an n-dimensional space, with n the number of loops surrounding the statement. The set of all the points for which a certain statement is executed defines a bounded polyhedron, hence the name polyhedral model. The loop controller iterates through the different loops in the program and for triggers the individual statements.

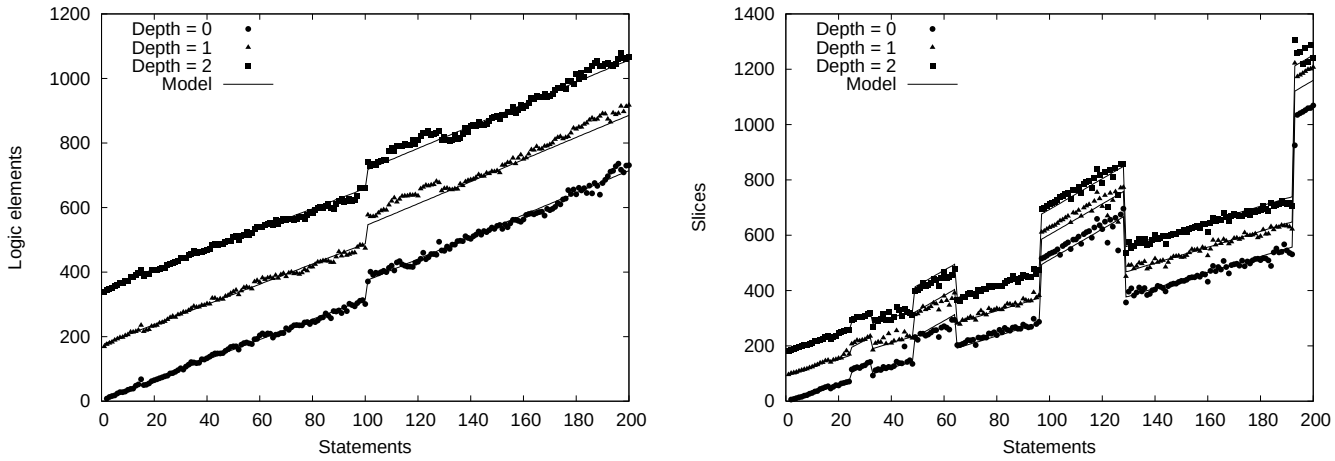
Within the polyhedral model, programs are transformed by performing simple matrix operations on the matrices that describe the polyhedra. These transformations only change the execution order of the statements and leave the statement implementations intact. The loop controller is thus the only component that changes by the program transformations.

Code generation from the polyhedral model is done with a tool called CLooG [1]. CLooG was extended to CLooGVHDL [2] to generate VHDL directly from a polyhedral representation.

III. Area estimation model

Our proposed estimation strategy uses empirical equations drawn from detailed observations of the synthesis results of a set of synthetically generated benchmarks. In the next step we fit the model obtained from the synthetic benchmarks to the SpecFP 2000 benchmarks. We try to minimize the average of the absolute value of the relative error. The reason we do not use the Root Mean Square Error fitting strategy is that this favors the larger, more complex benchmarks. Since these have lots of complex control structures inside them, they are not likely to be implemented in hardware. In the last step of our methodology, we validate the model on the PerfectClub benchmarks and on a real world example. We have developped an area estimation model for both Altera and Xilinx FPGAs.

In the first experiment, we synthesized a series of programs consisting of a varying number of statements without any con-



(a) Altera (b) Xilinx

Fig. 1. Parameters affecting the controller area for Altera (a) and Xilinx (b). The points indicate the measurements and the lines represent our models.

TABLE I
2D-IDWT DESIGN VARIANTS.

Design	Xilinx		Altera	
	Slices	Est.	LE	Est.
Row-Column	443	381	805	739
Line-Based	788	707	1460	1423
Block-Based	2858	3004	7002	4593
Average error		7.66%		15.02%

trol. The results for this experiment are shown in Fig. 1. In case of an Altera FPGA, there is a linear relationship between the number of statements and the FPGA resources needed by the loop controller. There only is a gap in the curve at 100 statements. As opposed to this, the area usage for Xilinx FPGAs is much more irregular. Two different sets of regions can be identified in the curve, each having a different area utilization characteristic. This is due to two different synthesis strategies used by the Xilinx synthesis tool. We were able to determine the bounds of these regions, so we could model the area utilization for Xilinx too.

We have repeated the experiment discussed above with a varying number of loops surrounding the statements. As you can see in Fig. 1 the shape of the curve is maintained for both Altera and Xilinx if the loop nest depth is increased. There is only an offset due to the logic needed to implement the loop iterators of each loop level in the program.

Although the area usage characteristics are very different for both Xilinx and Altera, we were able to model them fairly accurately, as indicated in Fig. 1. Other parameters, like the presence of conditionals in the program, have also been modeled, but will not be further discussed due to the page limitations for this paper.

IV. Validation

We have validated our estimation model with a set of loop controllers from the PerfectClub benchmark suite. The average relative error reported was about 8% for both Xilinx and Altera. Next to this, we have also applied our technique to a real world example. We have estimated the loop controller area usage of several design variants of the 2-dimensional inverse discrete wavelet transformation (2D-IDWT). As you can see in Table I, we were able to estimated the different design variants fairly accurately.

V. Conclusion and Future work

We have presented a model to estimate the FPGA resources needed by the loop controller during high-level synthesis. The model presented in this paper is accurate enough to be used during design space exploration. Due to its simplicity and the abstract level of estimating, the estimation overhead is minimal, which is critical when lots of design variants have to be evaluated. The presented methodology can easily be adapted to new FPGA design flows and architectures, like we showed with estimations for both Altera and Xilinx FPGAs.

Future work will focus on the data path during the estimations. For this purpose, the techniques presented in other papers could be used in conjunction with this work. Next to this, additional research is needed to move towards an automated design space exploration framework. We will also develop a similar model to estimate the clock frequency of the loop controller and the total number of registers needed.

Acknowledgments

Tom Degryse is supported by a BOF grant from Ghent University. Part of this research is funded by the IWT grant 060068.

References

[1] C. Bastoul. *Code generation in the polyhedral model is easier than you think*, In PACT, pages 7–16, 2004.

[2] H. Devos, K. Beyls, M. Christiaens, J. Van Campenhout, E. H. DHollander, and D. Stroobandt, *Finding and applying loop transformations for generating optimized FPGA implementations*, Transactions on High Performance Embedded Architectures and Compilers I, LNCS, 4050:159–178, 2007.

[3] R. Enzler, T. Jeger, D. Cottet, and G. Tröster. *High-level area and performance estimation of hardware building blocks on FPGAs*, In FPL, pages 525–534, 2000.

[4] D. Kulkarni, W. A. Najjar, R. Rinker, and F. J. Kurdahi. *Compile-time area estimation for LUT-based FPGAs*, ACM Transactions on Design Automation of Electronic Systems, 11(1):104–122, 2006.