Controller Estimation for FPGA Target Architectures During High-Level Synthesis*

Carsten Menn
FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
+ 49-721-9654-414
menn@fzi.de

Oliver Bringmann
FZI Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14
76131 Karlsruhe, Germany
+ 49-721-9654-455
bringman@fzi.de

Wolfgang Rosenstiel
Universität Tübingen
Sand 13
72076 Tübingen, Germany
+ 49 - 7071 - 29 - 75482
rosen@informatik.uni-tuebingen.de

ABSTRACT

In existing synthesis systems, the influence of the area and delay of the controller is not or not sufficiently taken into account. But the controller can have a big influence, especially, if a certain data-path realization requires a huge number of states and/or control signals. This paper presents a new approach on controller estimation during high-level synthesis for FPGA-based target architectures. The estimator, presented in this paper can be invoked after or during every synthesis-step, i.e. allocation, scheduling and binding, respectively. By considering the controller influence on the overall area of a design, design space exploration can be made more accurate and less error prone. We present an approach for estimating area of the controller based on information which are easily accessible during each step of high-level synthesis, so no explicit description of the controller, which usually will be generated after the binding, is necessary. This is particularly valuable in the allocation phase, where intensive design space explorations have to be done, based on fast and accurate estimates.

Categories and Subject Descriptors

J.6 [**Computer-Aided Engineering**]:*computer-aided design (CAD)*

General Terms

Design.

Keywords

Area Estimation, Controller, FPGA, High-Level Synthesis.

1 INTRODUCTION

High-level synthesis consists of several steps and during most steps some parameter, that have influence on various properties of the resulting hardware, need to be adjusted. The properties, that are important are the delay of the longest combinational critical paths, the latency and the area that it will occupy on the target architecture. High-level synthesis is very expensive in terms of the number of calculations that have to be made. Furthermore, area and delay constraints can be checked not before all synthesis steps have been performed. For this reason it is necessary to have estimation methods to be able to predict those properties without actually performing the synthesis steps. Fast and accurate estimates are especially needed during the allocation phase, where the examination and evaluation of a huge number of possible candidates has to be performed.

Our estimator predicts the expected data-path, based on the available information of the (partially-) synthesized design. Instead of building a real data-path, for estimation it is sufficient to have a model of the RT-structure which preserves the relevant data. The creation of this model is based on a heuristic function, which predicts the expected schedule and binding, if one or both are not already performed. Additionally, the area of the expected controller is estimated (which will be the content of this paper). The RT-model includes all kinds of multiplexors, i.e. those needed for resource sharing and those resulting from the control-structure. Information about the number and location of registers in the data-path is also included. Figure 1 depicts the design-flow from

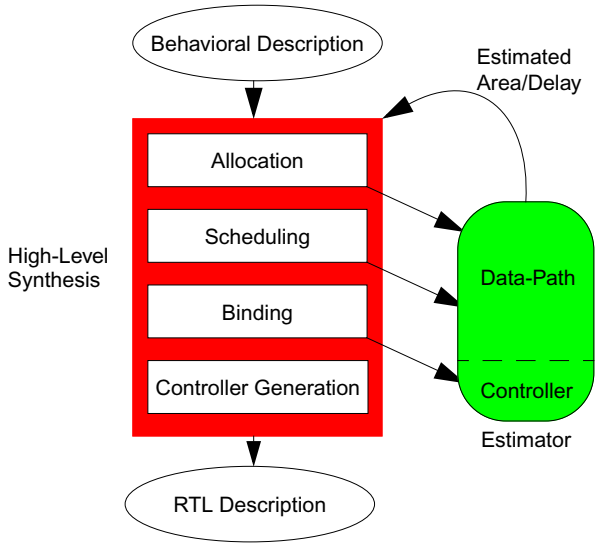

Figure 1. Estimation during high-level synthesis

ISSS'02, October 2-4, 2002, Kyoto, Japan.

* This work was partially funded by BMBF Speak-Project (Medea A508) and DFG-Priority-Program Rapid Prototyping RO1030/4-3

behavioral to RTL description together with an estimator which can be called after each step of high-level synthesis.
Many estimation techniques have been proposed so far and some considering the controller, but nearly all of them need at least a behavioral description of the controller. Because it is necessary to have accurate estimations even before scheduling and binding, where no such description is available, we propose an approach which is able to predict the area of a controller, based on information, which are available before the controller actually is generated.
After a brief overview of existing estimation approaches in Section 1.1, and the target architecture in Section 2, we will present our approach, where we first show, why a purely analytical approach can not sufficiently solve the problem and therefore, why we have to use experimental examination (Section 3). Afterwards we describe which parameters are available, or can easily be estimated during high-level synthesis and which of them are appropriate candidates to estimate the area of a controller (Section 4). In Section 5 we examine the dependencies between these parameters and the area and how to estimate the area of the controller. The results, which we have obtained are presented in Section 6. We will finish this paper with a summary and an outlook in Section 7.

1.1 Related Work

Several estimation approaches have been proposed in the past. Some focus on ASICs, some on FPGAs as target architecture, but only a few among all of them take the influence of the controller on the overall area of a design into account.
Nedjah and de Macedo Mourelle propose in [4] an approach to predict the number of CLBs required to implement the data-path and the controller, as well. Unfortunately their data-path is restricted to have exactly one general purpose functional unit. The number of CLBs required to implement the controller equals the number of flip-flops needed for the states, where they assume a maximum of 10 flip-flops will be sufficient and so they fix the size of the controller to be equivalent to a mean value of 6 flip-flops.
Xu and Kurdahi focus likewise on FPGAs as target architecture in [6], which is based on the previous work of Ramachandran and Kurdahi [5]. Their approach requires an already scheduled behavioral specification. After they have finished the binding, a gate level netlist is constructed, which is used to estimate the controller.
Quite impressive results on area estimation were presented by Mitra and Panda [3], but their work was restricted to standard cell target architectures. A PLA-based implementation of the controller is assumed by Katkoori and Vemuri in [1]. By our knowledge, their approach is the only one which does not require a description of the controller. Unfortunately the presented results are not convincing, because of large estimation errors.
None of the published approaches has ever estimated the area of a controller, implemented on a FPGA target architecture, without having at least a behavioral description of the controller.

2 TARGET ARCHITECTURE: FPGA

FPGAs (Field Programmable Gate Arrays) are digital circuits, which can be programmed by the user. They consist of a regular, symmetrical structure of single interconnected functional blocks. We used in our work FPGAs of the XC4000-series, manufactured by Xilinx Inc., but we are convinced of the usability of our estimation for other types of FPGAs as well (please see Section 7, where some examinations with the Virtex-series are described).
The functional blocks in the XC4000 series are named CLBs (Configurable Logic Block). In Figure 2 the general structure of one CLB is depicted..

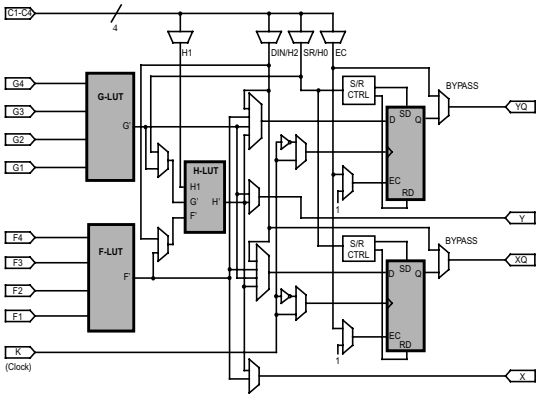

Figure 2. CLB of a XC4000 FPGA

Each CLB consists of three look-up tables (LUTs), where two of them (namely G and F) have four and the third (H) has three inputs. Each k-input-LUT is able to realize every arbitrary boolean function with k variables. Additionally, each CLB has two flip-flops, which enables the implementation of buffered outputs.
Although each CLB has nine inputs, it is not possible to realize every arbitrary boolean function with nine variables, but only a few of them. However, all boolean functions with a maximum of five inputs can be realized.

3 ANALYTICAL vs. EXPERIMENTAL APPROACH

A controller can be fully described by the following 6-tuple: $FSM = (I, O, S, \delta, \lambda, S_0)$, where *I* is a finite set of inputs, *O* a finite set of outputs and *S* a finite set of states. δ describes the state transition function, while λ describes the output function. The dedicated initial state is defined by S_0. Inputs, outputs and states, as well as the initial state are known or can easily be estimated, even during the allocation phase of high-level synthesis, whereas δ and λ are first available after the binding has been finished. However, we know that $N_f = \lceil ld(|S|) \rceil + |O|$ functions have to be implemented overall, which are depending on a maximum of $\lceil ld(|S|) \rceil + |I|$ variables.
The number of needed CLBs to implement all functions can now be calculated by: $N_{CLB} = N_f \times K$, where the cost factor *K* describes the cost for one function and depends on the number of dependent variables and the type of available LUTs. If the number of dependent variables exceeds the number of inputs of a LUT, *K* can not be determined, because not all boolean functions with more than five variables can be realized. To calculate *K* for those cases, it would be necessary to have exact knowledge about the boolean function to be realized, to determine whether one CLB is sufficient or not.

Legl et al. proposed in [2] to solve this problem by using the following formula:

$$N_{LUT} = \begin{cases} 1 & , n \le k \\ n-k+1 & , n > k \end{cases},$$

where *n* is the number of depending variables, *k* the number of inputs of one LUT and N_{LUT} the resulting number of needed LUTs. We know the maximum number of dependent variables, but not the exact number, required for each function. The LUTs used in XC4000 series have different number of inputs, therefore this approach does not solve our problem.

Another problem is the fact, that with the increasing complexity of a controller, the complexity of the interconnect is growing as well. Therefore some CLBs can not be used optimally. In some cases additional CLBs are needed, only to feed signals.

In summary one can say: An analytical approach is not possible if we assume that no explicit description of the controller is available. So we decided to solve the problem by using experimental examinations.

4 DETERMINATION OF APPROPRIATE PARAMETERS

As stated before, we assume, that no explicit controller description is available, as it applies to all phases in high-level synthesis (except the controller generation phase). The first question to be answered is: Which parameters supply sufficient information to get accurate estimations on the area of a controller? We have examined several different parameters and evaluated their applicability. In the following sections we will describe which on the one hand are easy to determine or to calculate and on the other hand lead to accurate estimates. Our goal is to determine a single formula, which can be calculated very fast and provide the desired accurate results.

Dependent on the synthesis steps, which have already been executed, our data-path estimator will be used to build a model of the RT-structure, i.e. the missing results will be estimated.

4.1 Number of Flip-Flops(a)

The number of flip-flops plays a dominating role for the area of a controller. Assuming, the states of the controller are binary encoded, the number of needed flip-flops can be easily determined by:

$$a = \lceil ld(|S|) \rceil \quad .$$

4.2 Number of Outputs(b)

After the allocation of functional units, the number of control-inputs and outputs is known for those units. Additional outputs are needed for the required multiplexors, either induced by the control structure (case, loop, etc.) or by resource sharing, and for registers. The set of outputs is defined as $O = \{O_1, O_2, \ldots, O_{|O|}\}$, where $\forall i \in \{1, \ldots, |O|\} : O_i = (o_{i1}, o_{i2}, \ldots, o_{i|S|})$. We define the next parameter *b* as:

$$b = |O| .$$

4.3 Number of Transitions(c)

The minimum number of transitions (N_T) of a controller is $|S|-1$. If the schedule requires a repetitive execution, than the minimum number of transitions is $|S|$. As soon as the controller has at least one input, the number of transitions will exceed the number of states of the controller. We are only interested on the number of additional transitions, so the parameter *c* is defined as:

$$c = \max\{0, N_T - |S|\} .$$

N_T is calculated in advance by our data-path estimator.

4.4 Number of Unused States(d)

With parameter *a*, we know the number of flip-flops, required to code $2^{a-1}+1$ till 2^a states. If the number of states is within these margins but less than 2^a, $2^a - |S|$ states are not used. The more flip-flops are available, the higher is the number of possible unused states. To be independent from the number of flip-flops, we define the next parameter d as:

$$d = \lceil ld(|S|) \rceil - ld(|S|) ,$$

where $d \in [0;1)$. A value of $d = 0$ means, that there are no unused states in the controller. A value of $d = 1$ is not possible, because this would imply, that the number of flip-flops is not minimal.

4.5 '0'/'1' Ratio in Output Signals(e)

Precise informations about the output-signals are not available before the binding step has been executed. The density of the '1's in each output can be calculated by determining the probability, how often a component will be selected. The signal on the output in these states is inverted in all other states, where the component is not selected. We will explain this by an example.

We assume an estimated scheduling and one arithmetic component which is able to execute additions and subtractions. We are interested in the '1' distribution probability of the „select“ input of this component. On the left side of Figure 3, a part of the data-flow graph is depicted. The marked area includes those operations which are estimated to be executed by the arithmetic component. In the middle of Figure 3, a part of the state-diagram is shown, which is the result of the estimated scheduling. It should be noted, that the number of states is known, but not the concrete mapping of operations to these states. Actually, this is not a problem, because the ratio between '0's and '1's is independent from this fact. According to the here used model, the number of executed operations on this component is invariant.

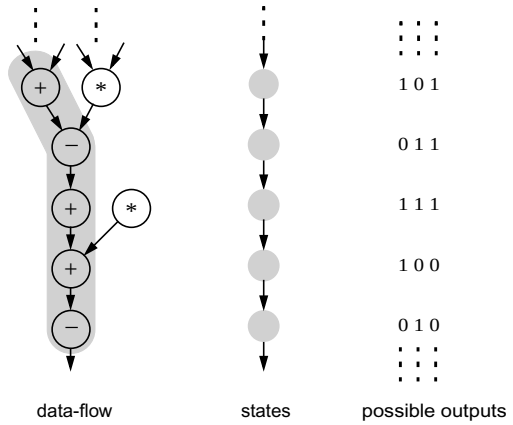

Figure 3. Determination of the '0'/'1' ratio

On the right of Figure 3, three possible bindings with the corresponding output are shown, where the ratio between '0's and '1's is always the same.

For the parameter *e*, we have to distinguish between Mealy- and Moore type controllers. This is necessary because the maximum number of bits in an output signal depends on the type of the controller. For a Mealy type controller, we define for each output O_i the parameter $e_{i,mealy}$ as follows:

$$e_{i,mealy} = \frac{1}{N_T} \cdot \sum_{j=1}^{N_T} o_{ij}\Big|_{o_{ij}=1}$$

For a Moore type controller the parameter $e_{i,moore}$ is defined similarly for each O_i :

$$e_{i,moore} = \frac{1}{|O|} \cdot \sum_{j=1}^{|O|} o_{ij}\Big|_{o_{ij}=1}$$

Now we can define the last parameter *e*, which describes the ratio between '0's and '1's for all outputs

$$O_i \in O :$$

$$e = \frac{1}{N} \sum_{i=1}^{N} e_{i,X} \quad \text{where } N = \begin{cases} N_T & \text{,if X = mealy} \\ |O| & \text{,if X = moore} \end{cases}$$

5 AREA ESTIMATION

To extract significant results out of the experimental examinations, it is necessary to have as much data available as possible. Therefore, we have developed a controller generator, which is able to build arbitrary behavioral descriptions of controllers, based on the parameters *a*, *b*, *c*, *d* and *e*. For each given set of these parameters, a huge number of possible controllers exists. We examined the common features among those controller, which are generated out of the same parameters. Additionally, we concentrate on the dependencies between each parameter and the number of required CLBs. This is done by setting all parameters, except the one under consideration, to a fixed value, whereas the value of the interesting one changes.

In the following we will show some of our examination results, regarding these dependencies. Because of marginal differences and for the sake of clarity, each curve in the following figures stays as a representative for a multitude of generated controllers with the same parameter set.

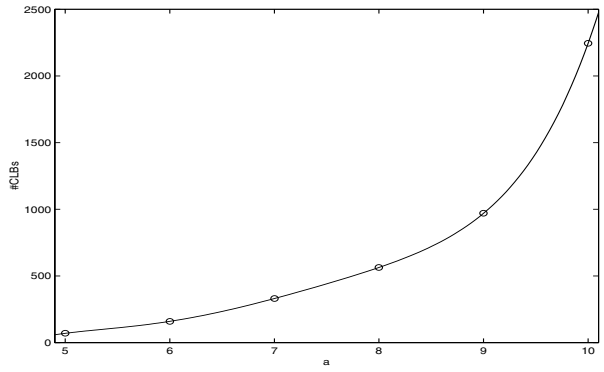

Figure 4. Dependency between area and parameter *a*

In Figure 4, we show the dependency between the number of flip-flops (parameter *a*) and the number of required CLBs for a controller. It can be seen by the shape of the curve, that it can be approximated by an exponential function.

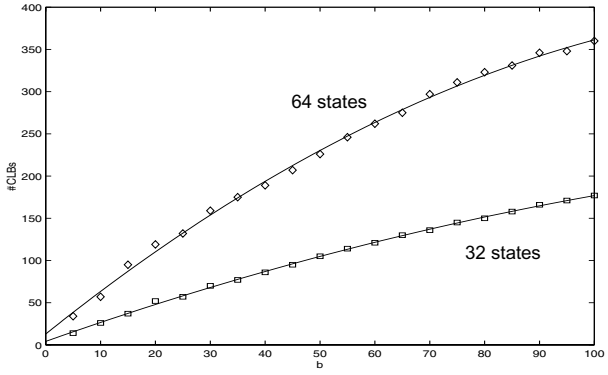

Figure 5. Dependency between area and parameter *b*

Two controller-types, one with 32, the other with 64 states, show examples of the dependency between the number of outputs (parameter *b*) and the number of CLBs, needed to implement the controller in Figure 5. This curve can be approximated by a root function.

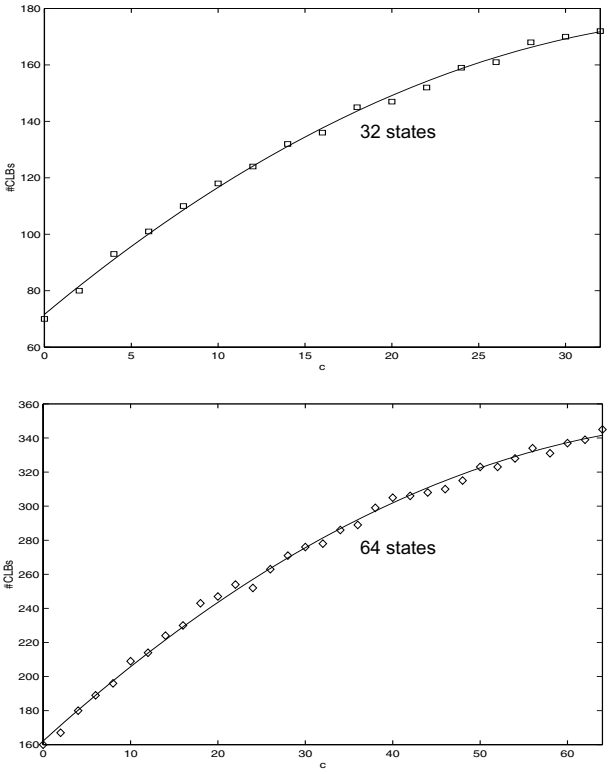

Figure 6. Dependency between area and parameter *c*

In Figure 6 the dependency between the number of CLBs and the number of transitions (parameter *c)* is depicted. (In the upper half for controller-types with 32, in the lower half with 64 states). As

in the case of parameter b, this curve can also be approximated by a root function.

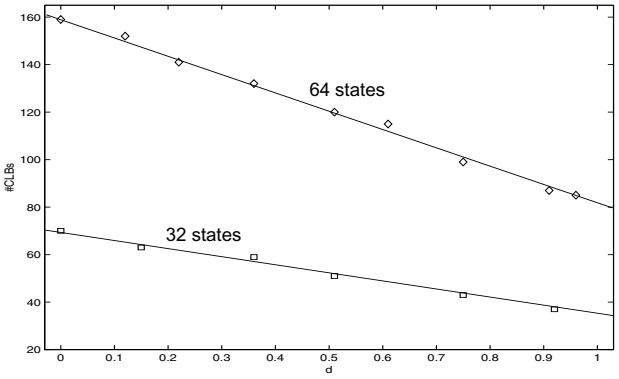

Figure 7. Dependency between area and parameter *d*

Nearly linear is the dependency between the number of CLBs and the number of unused states (parameter *d*). This is shown in Figure 7, again for controller-types with 32 and 64 states, respectively.

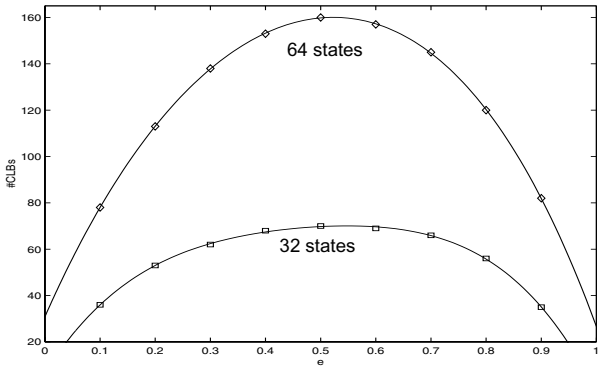

Figure 8. Dependency between area and parameter *e*

The dependency between the '0'/'1'- ratio (parameter *e*) and the resulting area is shown in Figure 8. In this case an approximation can be done by a parabola.

Because of the different shapes of the curves, describing the dependencies for each parameter alone, it is not possible to derive directly a formula like:

$$N_{CLB} = k_a \cdot a \otimes k_b \cdot b \otimes \ldots \otimes k_e \cdot e$$

to combine the dependencies for all parameter. Instead of this, first we put parameter *b* and *c* together by applying multiple linear regression. Then the result is combined together with parameter *a*, which leads to the following intermediate function:

$$\tilde{N}_{CLB} = f_1(a) + f_2(a)\sqrt{b} + f_3(a)\sqrt{c} + f_4(a)b + f_5(a)c + f_6(a)\sqrt{bc}$$

where the f_i are defined as follows:

$$f_i(a) = \sum_{j=1}^{8} k_{ij} \cdot a^{j-1}$$

The k_{ij} are the results from the linear regression and are not depicted here, due to space limitations.

In the following step the influence of the parameter *d* and *e* have to be integrated. We do this, by applying two correction functions $K_d(d)$ and $K_e(e)$ to $\tilde{N}_{CLB}$, which are defined as:

$$K_d(d) = 1 - 0,5d$$

$$K_e(e) = 3,9426e^3 - 6,3106e^2 + 3,8367e + 0,1720$$

The number of CLBs, required to implement a controller with the given parameter *a* to *e* can now be calculated as:

$$N_{CLB} = \tilde{N}_{CLB} \times K_d \times K_e$$

6 RESULTS

We tested the quality of our controller-estimation on seven algorithmic descriptions of different designs, where *diffeq* is a differential equation solver, *maha*, *ar*, end *ellip* are different kinds of filters. In addition to the last mentioned typical benchmark designs, we used *jpeg*, which is the compression algorithm of the jpeg-coding, as well as *subband* and *hybrid*, which are both processes of the MP3-coding algorithm.

To compare our results we synthesized the designs with two different HLS-systems. RT-synthesis was done by a widely used commercial synthesis tool, while the mapping to the FPGA was done by *xact* from Xilinx Inc. Table 1 shows the result we obtained for area estimation

.

Table 1. Results of area estimation

algorithm	HLS-system	N_{CLB}	estimated	error (abs)	error (rel)
diffeq	1	10	10	0	0
diffeq	2	11	12	+1	+9.1%
maha	1	15	15	0	0
maha	2	55	63	+8	+14.5%
ar	1	17	20	+3	+17.6%
ar	2	30	32	+2	+6.7%
ellip	1	33	29	-4	-12.1%
jpeg	1	36	28	-8	-22.2%
subband	2	202	226	+24	+11.9%
hybrid	2	1137	1245	+108	+9.5%

In Figure 9 these results are depicted graphically, for sake of clarity. The grey bars show the real, while the black bars show the estimated number of CLBs. The numbers in parentheses indicate, which HLS-system was used.

It is important to note, that the linear regression, performed as one part of our estimation approach, is only based on the automatically generated synthetic controllers. Whereas the controller of the benchmarks and the real applications have not been incorporated into the linear regression step. Since the obtained results

are very promising, our approach seems to be very effective in general.

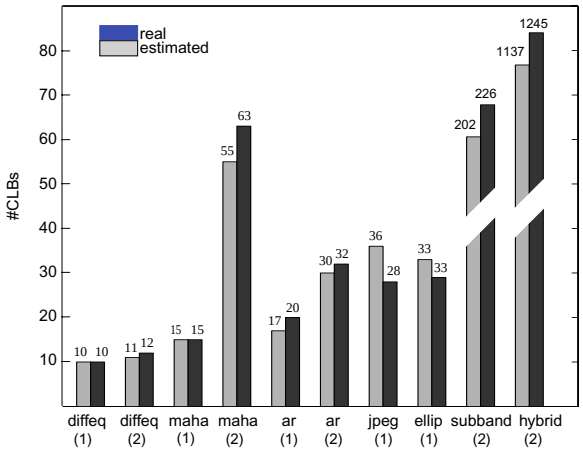

Figure 9. Estimated vs. real area

7 SUMMARY AND OUTLOOK

We presented a new approach on estimating the area of a controller during high-level synthesis, when FPGAs are used as target-architecture. This estimation can be done even during the allocation phase, because it depends on easy accessible data. It can be calculated very fast and is independent from the complexity of the controller.

We have made additional efforts to estimate the delay of a controller, based on the same parameters as used for area estimation. Till now it is possible to estimate the number of CLBs on the different paths through the controller. These estimates are quite accurate, but to know the number of CLBs on a path is not only sufficient to infer the delay from it. To get more accurate predictions, additional information about the number of switch-matrices, the type of wire (single-, double- or longline) and the distance between the CLBs have to be taken into account.

One obvious issue, that has to be worked on in the future is the examination of different types of FPGAs. Actually we start such examinations for the Virtex series from Xilinx Inc. The results obtained so far, show, that all dependencies between the parameter *a* to *e*, are the same in terms of the trend of the curves. Only the constants have to be adjusted, i.e. recalculating the k_{ij} and the correction functions $K_d(d)$ and $K_e(e)$.

Another issue to be examined in the future will be, to remove the restriction to binary-coded states. Especially the one-hot coding will be of interest, because their application reduces the complexity of the next-state logic, drastically.

8 REFERENCES

[1]S. Katkoori, R. Vemuri: *Accurate Resource Estimation Algorithms for Behavioral Synthesis*, 9th Great Lake Symposium on VLSI, pp. 338 - 339, 1999

[2]. Legl, C. Wurth, K. Eckl: *A Boolean Approach to Performance-Directed Technology Mapping for LUT-Based FPGA Designs*, Design Automation Conference 1996.

[3]. Mitra, P. R. Panda: *Estimating Complexity of Synthesized Designs from FSM Specifications*, IEEE Design & Test of Computers, pp. 30 - 35, March 1993.

[4]N. Nedjah, L. de Macedo Mourelle: *How many CLBs does your circuit need to be implemented?* 12th International Workshop on Rapid System Prototyping, pp. 174 - 179, 2001.

[5]. Ramachandran, F. J. Kurdahi: *Incorporating the Controller Effects During Register Transfer Level Synthesis*, European Design and Test Conference 1994.

[6]M. Xu, F. J. Kurdahi: *RTL Synthesis with Physical and Controller Information*, European Design and Test Conference 1997, pp. 299 - 303, 1997.