INTEGRATION, the VLSI journal 47 (2014) 461–475

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

High-level parameterizable area estimation modeling for ASIC designs

Ville Eerola*, Jari Nurmi

Tampere University of Technology, Department of Electronics and Communications Engineering, P.O. BOX 553, FIN-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 23 May 2013
Received in revised form
4 January 2014
Accepted 10 January 2014
Available online 23 January 2014

Keywords:
Gate-count estimation
Architecture exploration
VLSI circuits
Integrated circuit modeling
System-on-chip

ABSTRACT

Architectural design space exploration and early area budgeting for ASIC and IP block development require accurate high level gate count estimation methods without requiring the hardware being fully specified. The proposed method uses hierarchical and parameterizable models requiring minimal amount of information about the implementation technology to meet this goal. The modeling process flow is to: (1) create a block diagram of the design, (2) create a model for each block, and (3) sum up estimates of all sub-blocks by supplying the correct parameters to each sub-model. We discuss the model creation for a few parameterized library blocks as well as three communication blocks and a processor core from real IC projects ranging from 22 to 250 kgates. The average relative estimation error of the proposed method for the library blocks is 3.2% and for the real world examples 4.0%. The best application of this method is early in the design phase when different implementation architectures are compared.

1. Introduction

As both the complexity of Application Specific Integrated Circuit (ASIC) designs and demands for rapid time to market keep increasing, the importance of good estimates for the design complexity is growing. The gate count of a design affects its cost in several ways. The direct influence comes from the cost of the silicon area. The gate count also has a close relation to the device power consumption via the total capacitance accountable to the gates. The testability and test time of the device are affected by the complexity. Finally, the device yield is decreased with increasing area. In order to predict these, the device gate count or an other complexity measure is needed. In many cases the financial feasibility of an ASIC development project also depends on these estimates. It is very desirable that an accurate estimate of the device complexity could be made as early in the development as possible. Often, the first estimates would be needed even before any detailed system design has been made.

Early gate count estimation is desirable also because design optimization is more efficient at the early phases than later in the project. This would enable efficient hardware optimization at the architectural and system design levels. In many cases, the design contains several parameters which could be changed to alter its performance. However, at the same time, these alterations also have an effect on the silicon area required for the implementation. Having a gate count estimation model, which would follow the parameter changes, would allow more efficient high-level design space exploration and accurate planning of the design project.

The gate count estimation method presented in this paper has been developed to meet these demands. It can cope with rough and sketchy design specification since it does not depend on a formal and complete description of the device. It also allows having parameters to be included in the block descriptions, that will be taken into account when the gate count estimates are computed. Furthermore, the methodology requires a minimal amount of information about the implementation technology and keeps it separated from the design so that the implementation technology can easily be changed if needed. In the proposed methodology the gate count $\mathcal{G}$ for a block $\mathcal{B}$ is modeled using bottom-up parameterizable models.

The strength and differentiating factor in our methodology is to be free of requirements for strict design methodology or a formal description of the design to be estimated. This freedom allows gate count estimation to be done with incomplete designs.

The remaining of this paper is organized as follows: In Section 2 we will review some of the earlier work on the hardware area estimation field. In Section 3 we present an overview of the proposed gate count estimation methodology at a high level, describing the main steps involved. Next, in Section 4 we will explain the details involved in the gate count estimation model creation and explore some of the basic building blocks. In addition we will show the modeling process of some more complex building blocks: a multi-port register file, a CORDIC phase rotator and a complex mixer. In Section 5 we will go through four example cases from real world designs and present the results from the example cases and compare our estimates with actual gate counts. Furthermore, we will give some notes and summarizing

* Corresponding author.
E-mail addresses: Ville.Eerola@tut.fi (V. Eerola), jari.nurmi@tut.fi (J. Nurmi).

0167-9260/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.vlsi.2014.01.002

comments. Finally, Section 6 concludes the work and shortly discusses potential improvements to the methodology.

2. Related work

Several approaches to the design complexity estimation in area, gate count, or Field Programmable Gate Array (FPGA) resource usage have been proposed in the literature. The earliest approach for estimating the switch count of implementing Boolean functions was done by Shannon [1], who proposed that the *upper limit* of the complexity of Boolean functions is proportional to the exponential of the number of inputs. This was later applied to the gate count estimation for implementations of Boolean functions by Muller [2]. However, it is easy to see that these estimates cannot be applied to realistically complex circuits, due to the exponential growth of the estimate.

Kellerman [3] presented a formula for the computation of the area estimate and proposed that the area of a function is only depending on the number of conditions which must be differentiated by a one or a zero output. Other researchers, e.g. Cook and Flynn [4] and Pippinger [5], studied the relationship between the area complexity of Boolean functions and the entropy ($\mathcal{H}$). Cheng and Agrawal [6] expanded the entropy measure based estimation method to multi-output Boolean functions.

Nemani and Najm [7] pointed out problems in earlier work, which had been based on *randomly generated* Boolean functions with a small number of inputs. They showed how earlier estimates would greatly overestimate the gate count of real circuits and proposed that typical circuits are far from random in their structure. In their work, the authors developed a new *linear measure* of Boolean functions, which is dependent on the complexity of the on and off-sets of the function. Using that measure gave more realistic results when using typical circuits.

Büyükşahin and Najm [8] developed the estimation method further and used a Boolean network representation of the circuit instead of RTL level description to enable estimation at a higher level of abstraction. Still, use of these kinds of methods requires the knowledge of the accurate Boolean functions of the logic, which may not be available for all parts of the design at early phases of the design. For example, in many cases the accurate function for control logic is not known at the architecture design phase, even if the data processing would be well defined.

For the estimation of FPGA design complexity, a methodology was described by Enzler et al. [9]. Their methodology is based on capturing the features of the design on a data-flow graph (DFG) or block diagram level to a characterization vector consisting of the number and word lengths of adders, multipliers, etc. in addition to some other characteristics of each block. This is then mapped into the FPGA area and timing estimates. There is some similarity to our methodology, but our approach allows more freedom in the characterization as it is not tied to a fixed vector format containing counts of only a few characteristic functions such as adders, multipliers, and multiplexers. Our method allows freedom of selecting the functions to which the design is mapped to. This allows an optimal set of functions to be chosen for different classes of designs. It is also possible to increase the estimation accuracy by incrementally adding more basic blocks to be used in the estimation process over time.

It also seems apparent that many of the earlier efforts on the gate count or area estimation field seem to concentrate on integrating the algorithms into CAD tools for building more comprehensive frameworks like in [10].

Table 1 summarizes the above references with regard to the used design description (input) and resulting cost estimate (area) as well as shows the reported accuracy of the method, if available. Additionally, [11] provides some additional references and a good survey of the hardware characteristics estimation techniques related to hardware/software partitioning. In summary, most of the earlier work have been concentrated to finding a formula to map a small, combinatorial, accurately known logic function to a number related to the area of the silicon implementation. In contrast, our work aims to provide a more high-level tool for the area estimation, which would work on realistically complex designs early in the design process.

3. Gate count estimation methodology

We will now describe the estimation process at a high level. In the next section we will go into more detail of the process. The estimation methodology described in this paper is based on a bottom-up modeling approach. The models are built from parts at three levels:

1. Primitive library
2. Basic block library
3. Design level models

The lowest level of the model consists of a set of predefined primitives, which correspond directly to standard-cell gates of the implementation technology. The next level there is a pre-defined, but extensible, set of basic building blocks, such as adders, multipliers, and registers. The highest level consists of models that are specific to the design under estimation. The difference between the basic blocks and the design level models is that the basic blocks are re-usable and parameterized in such a way that they can be used in modeling of many different designs. On the other

Table 1
Area estimation references comparison.

Reference	Input	Cost (output)	Average accuracy	Notes
[1]	Boolean functions	Switch count	n/a	Theoretical work
[2]	Boolean functions	Gate count	n/a	Theoretical work
[3]	Boolean functions	Diode count	15% (random) / 33% (real)	
[4]	Boolean functions	Diode count	n/a	Theoretical work
[5]	Boolean functions	Gate count	n/a	Theoretical work
[6]	Boolean functions	Gate count	30%	
[7]	Boolean equations	Gate count	22%	Based on on/off sets of function
[8]	Boolean network	Gate count	24%	
[9]	Block diagram/DFG	FPGA cells	12%	Intermediate vector representation
[10]	VHDL	FPGA cells	3.5%	Intermediate CDFG representation
This work	Block diagram/DFG	Gate count	4.0%	Parameterizable/extendable models