ISSN 0002-306X.Изв. НАН РА и ГИУА. Сер. ТН. 2012. Т. LXV, № 3.

UDC 530.15: 004.032.2 COMPUTER SCIENCE AND INFORMATICS

L.A. MARTIROSYAN

A QUICK AREA ESTIMATION METHOD FOR RTL COMPILERS

A quick area estimation method for designs generated by RTL compilers which is based on polynomial interpolation is proposed. It is applied to in industrial RTL compilers. This paper shows that the processed method provides higher accuracy, than other quick area estimation methods for RTL compilers.

Keywords: RTL compiler, area, characteristic, parameter, polynomial interpolation.

Introduction. The use of memory cores in system-on-chip (SOC) designs is growing rapidly [1], and the design decisions, made early in the design process,effect the entire chip design process. Embedding many memory cores per chip helps create powerful SoCs suitable for today's memory-hungry applications, but it brings with it the problem of memory test and repair. Traditional approaches to memory test and repair do not effectively manage the complexity of today's SoCs and the soaring cost of test. To address these challenges, semiconductor IP vendors have introduced a new type of IP called infrastructure IP (IIP). IIP functions like mini testers embedded in a chip. Examples of IIP include built-in self-test (BIST), built-in repair-analysis (BIRA), built-in self-repair (BISR), and error correcting codes for embedded memories [2, 3]. Implementation of the infrastructure is done in a form of a hierarchy of RTL Compilers [4].

One of the most important quality characteristics of RTL descriptions generated by RTL compilers is the area [5]. The accurate area estimation method for RTL descriptions generated by RTL compilers is logic syntheses [6], While providers of the commercial synthesis tools are increasingly focusing on SoC design, their tools usually implement a top-down approach that requires the SoC designer to fully define the function of a developed system, repeatedly decompose coarse-grained functions into smaller subfunctions, and then map them into the available library HW cores. Using such a methodology, the physical design characteristics can be estimated only in the final stages of design [7]. In case of RTL compilers, when many instances are generated, this method [6] is time consuming and inefficient. A fast area estimation tool is useful for near-immediate feedback of design feasibility and the effect of circuit modification [8]. There are a lot of works related to this issue. Paper [8] represents a method for fast, high-level area estimation using the constant-delay paradigm and a zero-slack algorithm, but as the characterization phase becomes more complicated, the current method of hand analysis quickly becomes too tedious and error-prone. Paper [9] represents a method which is based on use of the Boolean network concept, and defines an area complexity measure which is invariant across the different BN (Boolean Network) representations of the same design. The average error of this

method is 24.5%. Paper [10] proposes a methodology to estimate the circuit area, minimum and maximum leakage current, and maximum power-up current, introduced by leakage reduction using sleep transistor insertion, for any given logic function. Compared to time-consuming logic synthesis and gate level analysis, the average errors for circuits from a leading industrial design project are 23.59%. Paper [11] offers a method for area estimation, which is based on transforming the multi-output function to an equivalent single-output function. The average error in area estimation was 21.07%.Paper [12] represents an equation-based macro-modeling technique for high-level area and power estimation on FPGAs. Paper [13] proposes area and power estimation models for IP core based FPGA implementations. Using curve fitting and non-linear regression methods, the models derived. [12, 13] methods provide enough higher accuracy for area estimation of IP core based FPGA implementations. Previous works represent methods which are typical for different design with given high-level description. In [14] and [15] the author offers a better way to estimate the area of RTL compilers than the above mentioned ones. Paper [14] represents a quick area estimation method, which is based on regularity of structure of schemes generated by RTL compilers. This method is based on approximation.The author mentions that a variety of approximation types is available: linear interpolation, polynomial interpolation, the least square estimation, etc. In [14, 15] the linear interpolation is considered. The maximum achieved estimation error is 15%. Our aim is to reduce the maximum error. We would like to explore how the results would be in case of polynomial interpolation. Hence, in this paper we represent a quick area estimation method which is based on polynomial interpolation and the maximum estimation error is 10 %. This method provides higher accuracy than [14, 15].

1. Methodology. Paper [4]mentions that theseRTL compilers are template based. They usually consist of templates that describe a parameterized hierarchy of modules and interfaces (interconnections) between them and a generation engine. The template input is a vector of input parameter values which defines features and a structure of design instances to be generated, and the output of the template is a RTL description with functionality corresponding to the given input.

Input parameters can be categorized by the following three types [4]:

· ***Functional***

These parameters control optional features/options of design. They affect the design structure by means of inclusion or exclusion of certain design components in the output RTL.

· ***External interface***

They parameterize HDL identifiers (module/wire/reg/instance) in a RTL description for customization of the generated RTL design to an external interface.

· ***Scalability***

These parameters affect the design structure by increase or decrease of certain designcharacteristics including but not limiting the register bit-width, number of

words (in memories), number of cores (e.g. in SoCs). The structural and functional changes of scheme are bringing to quality characteristics changes.

Paper [14] represents a quick area estimation method which is based on regularity of the structure of schemes generated by RTL compilers. There are functional dependencies between quality characteristic (in this case gate-count) of parameterized schemes generated by RTL compilers and input parameters. By finding out analytical representation we can estimate the area of scheme generated by RTL compilers for chosen values of parameters. As the analytical representation of function is unknown, [14] proposes to apply corresponding approximate function. The author mentions that a variety of approximation types is available: linear interpolation, polynomial interpolation, the least square estimation, etc. In [14, 15] papers the linear interpolation is considered. This paper shows that functional dependencies in case of some parameters are not linear. So for approximation of theme we use polynomial interpolation [16], which represents not linear dependence between argument and function. We added interpolation points to receive more exact approximate function. As a result, by adding interpolation points and applying polynomial interpolation, we processed a more accurate quick area estimation method for RTL compilers. The method has been applied to industrial RTL compilers and the maximum achieved generation error is 10%.

2.Test and Repair infrastructure for embedded memories. Fig. 1 shows a SoC which consists of STAR (Self Test and Repair) memory systems [2]. These test and repair systems are usually embedded on-chip to diagnose failed memory bits and repair the failed memory in real-time using the redundant resources (row or columns, or both) in the memory. It contains memories with wrappers, processors, server and fuse box. Each component has its own functional meaning [2, 3].

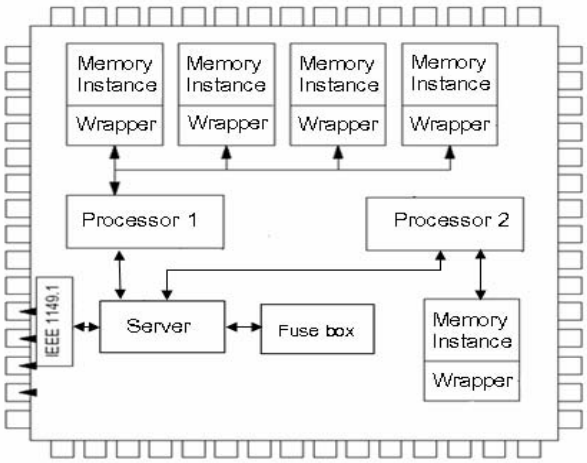

Fig.1. Test and Repair infrastructure

The wrapper contains address counters, registers, data comparators, and multiplexers [2]. It operates as an interface between processor and memory. The wrapper associated with each memory works with processor to perform memory test and repair as well as to allow normal memory functions.

Processor has key test and repair functions: a BIST(Built-In Self-Test)engine to create memory-specific test patterns; a BIST diagnostic engine to analyze and identify the failure; BIRA (Built-In Repair-Analysis), the repair and redundancy allocation logic with algorithms to reconfigure the memory rows; and columns to be topologically efficient post-repair. The processor enables test and repair possibility by using IEEE P1500 standard interface.

The fuse box stores memory reconfiguration signature. The contents of the fuse box correspond to the repair signature that is loaded into the corresponding memory for repair.

The server builds the top level design infrastructure utilizing low level processsors, wrappers and memories.

In hierarchy of RTL compilers the modules of processor, wrapper and server are represented in the form of separate RTL compilers for each type of memory. Each specific instance of the infrastructure is defined via assignment of specific values to parameters of compilers and generates of components and subcomponents for each level [4].

Paper [14] mentions that by analyzing the structure of design some regularities (number of words, number of bits, etc.) are found, and the area of STAR Memory systems have dependencies on some of the memory input parameters. This regularity allows to process quick area estimation method for RTL compilers.

3. Implementation details. The method described in this paper has been applied to processor and wrapper modules of STAR Memory Systems, as the method described in [14, 15]. So we also figure out those parameters that cause drastic variations in numbers of gates for STAR Memory Processor and Wrapper compilers.Gate count of RTL compilers depends on functional and scalability parameters of STAR Memory systems.

Experimental results are obtained through logic synthesis using Synopsys DC [6]. The sets of parameter values are used as interpolation points. Experimental results confirm supposition that in case of some parameters researched dependencies are not linear.Corresponding approximate functions have been got after polynomial interpolation [16].

The experimental results are represented in instance of wrapper (Fig. 2).

As a result of polynomial interpolation, the obtained approximate functions are embedded in area estimation script represented in TCL language [14, 15]. In order to verify the accuracy of developed methodology, we used the method represented in [14, 15] (Fig. 3).

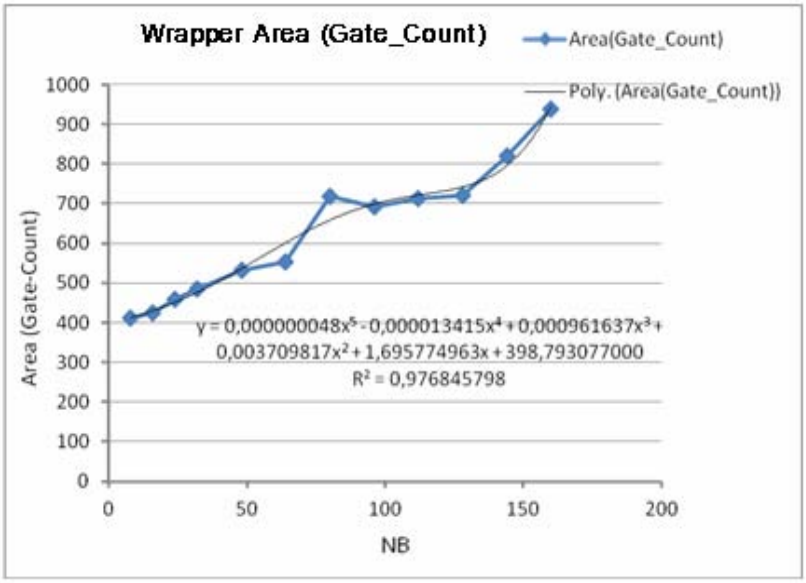

Fig.2. Wrapper's area dependency from NB parameter for SRAM type memory

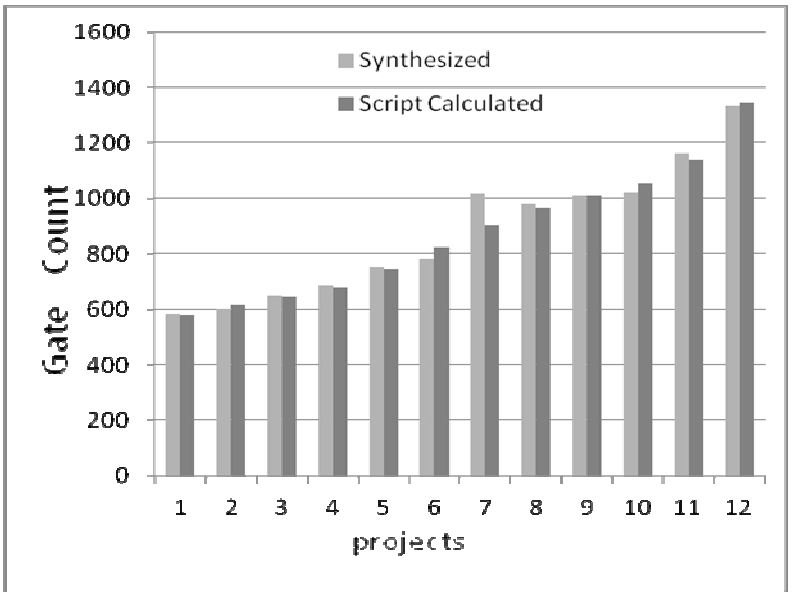

Fig.3. Wrapper area comparison chart in Synthesized and Script Calculated cases for SRAM type memory

Fig. 4 shows a script calculated (with polynomial and linear interpolation) and synthesized results comparison estimate, when the behavior of parameter is not linear.As it can be seen in those segments, where the behavior of parameter is not linear, polynomial interpolation makes more accurate approximation than linear interpolation. Hence, in those cases by using polynomial interpolation it will be possible to make more accurate area estimation.

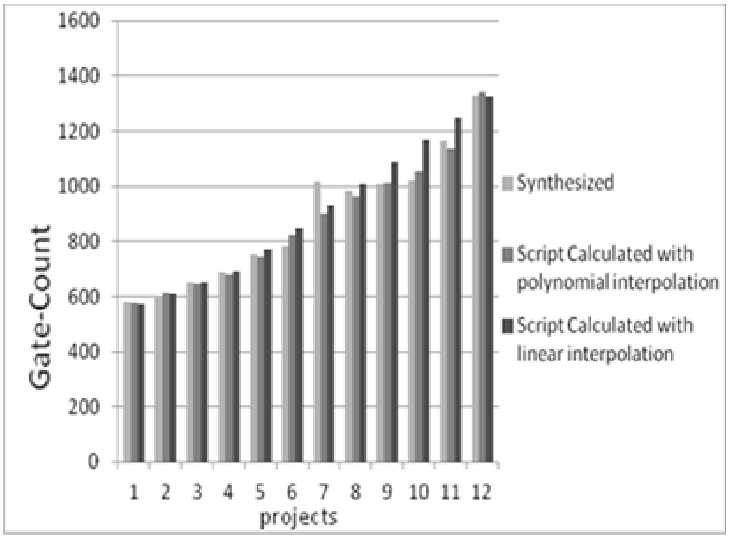

Fig.4. Wrapper area comparison chart in Synthesized and Script Calculated (with polynomial and linear interpolation) cases for SRAM type memory

Conclusion. The behavior of input parameters of RTL compilers is researched. It is shown, that in case of several parameters their behavior is not linear. A quick area estimation method is processed for RTL compilers of STAR memory systems, which is based on polynomial interpolation. This method compared with other quick area estimation methods [14, 15] for RTL compilers provides higher accuracy. The maximum generation error is achieved 10%.

REFERENCES

1. **Chin-Lung Su, Yi-Ting Yeh, Cheng-Wen Wu.** An Integrated ECC and Redundancy Repair Scheme for Memory Reliability Enhancement // 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05). - 2005.
2. **Zorian Y.**, **Shoukourian S.** Embedded-Memory test and repair: Infrastructure IP for SoC yield // IEEE Design and Test of Computers.-May-June 2003. - P. 58-66.
3. **Shoukourian S., Vardanian V., Zorian Y.** SoC yield optimization via embedded-memory test and repair infrastructure // IEEE Design and Test of Computers. - May, 2004.
4. **Margarian P. S.** An Approach for Automated Assertion Generation in Template Based RTL Compilers // Proceedings of the CSIT conference. - 2009. - P. 208-217.
5. **Kapoor B. and Murphy B**. Policy-Based RTL Design// IEEE 38th conference on Design automation. - 2001.
6. **Himanshu Bhatnagar.**Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler™Physical Compiler™ and PrimeTime®. Second edition. - Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2002. -355p.
7. **Damasevičius R.** Estimation of Design Characteristics at RTL Modeling Level Using SystemC // Information technology and Control. - 2006. - Vol.35. - P. 117-123.
8. **Sheets M.** Exploration of a Fast Delay/Area Estimation Method //EE219b Semester Project. - May 16, 2000.
9. **Buyuksahin K. M., Najm F. N.** High-Level Area Estimation // Low Power Electronics and Design: International Symposium. - 2002.
10. **Fei Li, Lei He, Basile J., Patel R. J., Ramamurthy H.** High-Level Area and Current Estimation // International workshop. - Turin, Italy. - 2003.- Vol. 2799. - P. 259-268.
11. **Nemani M., Najm F. N.** High-Level Area and Power Estimation for VLSI Circuit // IEEE/ACM International Conference, ISSN: 1092-3152. - San Jose, CA, USA, 1997.
12. **Tianyi Jiang, Xiaoyong Tang, Prith Banerjee.**Macro-models for High Level Area and Power Estimation on FPGAs // Proceedings of the 14th ACM Great Lakes symposium on VLSI. - New York, USA, 2004.
13. **Lanping Deng, Kanwaldeep Sobti, Chaitali Chakrabarti.** Accurate Models For Estimating Area and Power of FPGA Implementations // ICASSP. - 2008. - P. 1417-1420.
14. **Ter-Galstyan A., Mosikyan A**. An Approach for Quick Area Estimation of Compiler Genereted RTL // Proceedings of the CSIT conference, September 19-23, 2005.-Yerevan, Armenia, 2005. - P. 485-490.
15. **Ter-Galstyan A.** An Automation Method for Gate-Count Characterization of RTL Compilers // East-West Design & Test Workshop, September 15-19, 2006. - Sochi, Russia, 2006. - P. 313-316.
16. **Phillips G.M.** Interpolation and Approximation by Polynomia. - Springer, 2003. -327p.

SEUA (POLYTECHNIK).The material is received 09.02.2012.

Լ.Ա. ՄԱՐՏԻՐՈՍՅԱՆ

RTL ԿՈՄՊԻԼՅԱՏՈՐՆԵՐԻ ՄԱԿԵՐԵՍԻ ԱՐԱԳ ԳՆԱՀԱՏՄԱՆ ՄԵԹՈԴ

Ներկայացվում է RTL կոմպիլյատորների միջոցով գեներացված RTL նկարագրությունների մակերեսի արագ գնահատման մեթոդ, որը հիմնված է պոլինոմիալ ինտերպոլյացիայի վրա: Այն փորձարկվել է արդյունաբերական RTL կոմպիլյատորների վրա: Ցույց է տրվում, որ մշակված մեթոդը, համեմատած RTL կոմպիլյատորների մակերեսի գնահատման այլ մեթոդների հետ, ապահովում է ավելի բարձր ճշտություն:

Առանցքային բառեր. RTL կոմպիլյատոր, մակերես, բնութագիր, պարամետր, պոլինոմիալինտերպոլյացիա:

Л.А. МАРТИРОСЯН

МЕТОД БЫСТРОЙ ОЦЕНКИ ПЛОЩАДИ ДЛЯ RTL КОМПИЛЯТОРОВ

Предлагается метод быстрой оценки площади схем, генерируемых RTL компиляторами, который основан на полиномиальной интерполяции. Метод был применен в промышленных RTL компиляторах. Показано, что разработанный метод обеспечивает более высокую точность по сравнению с другими методами быстрой оценки площади для RTLкомпиляторов.

Ключевые слова: RTL компилятор, площадь, характеристика, параметр, полиномиальная интерполяция.