Evolutionary Approximation of Complex Digital Circuits

Zdenek Vasicek
Brno University of Technology
Faculty of Information Technology
IT4Innovations Centre of Excellence
Brno, Czech Republic
vasicek@fit.vutbr.cz

Lukas Sekanina
Brno University of Technology
Faculty of Information Technology
IT4Innovations Centre of Excellence
Brno, Czech Republic
sekanina@fit.vutbr.cz

ABSTRACT

Circuit approximation has been developed in recent years as a viable method for constructing energy efficient electronic systems. An open problem is how to effectively obtain approximate circuits showing good compromises between key circuit parameters – the error, power consumption, area and delay. The use of evolutionary algorithms in the task of circuit approximation has led to promising results; however, only relative simple circuit instances have been tackled because of the scalability problems of the evolutionary design method. We propose to replace the most time consuming part of the evolutionary design algorithm, i.e. the fitness calculation exponentially depending on the number of circuit inputs, by an equivalence checking algorithm operating over Binary Decision Diagrams (BDDs). Approximate circuits are evolved using Cartesian genetic programming which calls a BDD solver to calculate the fitness value of candidate circuits. The method enables to obtain approximate circuits consisting of tens of inputs and hundreds of gates and showing desired trade-off between key circuit parameters.

Categories and Subject Descriptors

B.0 [**Hardware**]: Logic Design—*General*; I.2.8 [**Computing methodologies**]: Artificial intelligence—*Problem Solving, Control Methods, and Search*

Keywords

Cartesian Genetic Programming, Combinational Circuit, Approximate Computing, Binary Decision Diagram

1. INTRODUCTION

In early fabrication technologies, Moore's law allowed doubling the number of transistors per square millimeter as well as increasing operation frequency every 18 – 24 months. While the density of transistors is still increasing, maximum frequency is roughly identical and the voltage is not decreasing accordingly. Hence the main challenge of modern chips is an efficient energy consumption management [1]. Several approaches to energy consumption reduction working at various levels have been developed recently.

For example, functional approximation can be employed to exploit inherent resilience of some applications to errors and reduce the power consumption. The idea is to implement a slightly different Boolean function to the original one providing that the error is acceptable and energy consumption is reduced adequately. The method starts with a fully functional circuit which is gradually modified by means of a problem specific heuristic in order to reduce the energy consumption, area and delay, and keep the error within predefined bounds. An evolutionary approach to circuit approximation has been proposed in [4]. However, because of the scalability problems of evolutionary design (especially the scalability of evaluation based on truth table representation), the approach allowed obtaining only relatively small approximate combinational circuits and 4-bit adders and multipliers.

The goal of this work is to introduce a method which can approximate more complex and general gate-level circuits. Introducing approximations to general logic could be dangerous in many cases (e.g. for controllers), but there is still an important class of circuits (such as circuits ensuring fault tolerance) in which the error (e.g. corresponding to the reliability level) can be exchanged for energy reduction. Contrasted to arithmetic circuits, in which it is natural to define the error, we do not suppose anything about the significance of output bits in our benchmark circuits. Hence the error is expressed as the Hamming distance between accurate circuit and a modified candidate circuit. The method is evaluated using a set of benchmark circuits that are difficult for the previous evolutionary approximation method, because they have too many primary inputs (27 – 50 inputs) and gates.

2. PROPOSED METHOD

The proposed method is based on Cartesian Genetic Programming [2] employing a population of $1 + \lambda$ individuals ($\lambda = 4$). The individuals are represented using a directed acyclic graph encoded by means of a string of integers of fixed length. New population is created by applying a point mutation operator modifying h integers ($h = 5$) of the parent individual.

Three design objectives are considered: circuit functionality (error), delay, and area. The functionality is evaluated using ROBDD (Reduced Ordered BDD) representa-

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '15 July 11–15, 2015, Madrid, Spain

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3488-4/15/07.

DOI: http://dx.doi.org/10.1145/2739482.2764657

tion. The electrical parameters are estimated using the parameters defined in the liberty timing file available for I3T25 technology (0.35 μm digital process). Delay of a gate is modeled as a function of its input transition time and capacitive load on the output of the gate. Delay of the whole circuit is determined as delay along the longest path. The area of a candidate circuit is calculated relatively to the area of a nand gate. It is assumed that power consumption is highly correlated with the area and hence it is sufficient to optimize for the area.

2.1 Hamming Distance Using BDDs

The merit of BDDs is that many operations for the manipulation of Boolean functions can be performed efficiently if the functions are represented by canonical ROBDDs. For example, *Sat-Count* operation computing the number of input assignments for which a Boolean function evaluates to one can be executed in linear time with respect to the BDD size.

The *Sat-Count* can be used to determine Hamming distance as follows. Let circuit C_A be a candidate circuit represented using CGP and C_B be a reference implementation (i.e. a fully functional circuit). Let us suppose that both circuits have k inputs denoted $x_1 \dots x_k$ and m outputs denoted $y_1 \dots y_m$ and $y'_1 \dots y'_m$, respectively. In order to determine the Hamming distance, a ROBDD has to be constructed by executing the *Apply* operation for every active gate G_i of C_A and C_B. As a result, a representation for $f \circ g$ is included into ROBDD provided that f and g are two operands of G_i (interpreted as pointers to appropriate ROBDD nodes) and $\circ$ is the logic function performed by G_i. Finally, XOR operation is applied to the corresponding primary outputs y_i and y'_i. The Hamming distance between truth tables of these circuits can be obtained by applying the *Sat-Count* operation on every output $z_i = y_i \oplus y'_i$ (i.e. a pointer to a ROBDD node) and summing up all the results. In the example shown in Fig. 1, *Sat-Count* will return 2 for z_1 and 0 for z_2, i.e. the Hamming distance is $0 + 2 = 2$. It can easily be checked that if $x \in \{0000, 0110\}$, the circuits provide different output values.

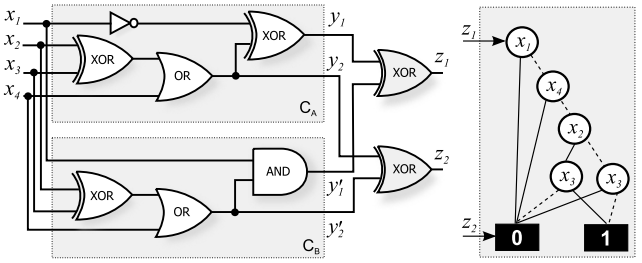

Figure 1: Hamming distance of two combinational circuits C_A and C_B using the equivalent ROBDDs

3. RESULTS AND CONCLUSION

In order to construct Pareto front, we follow the approach proposed in [3] in which a single-objective CGP is executed multiple times with different parameters. CGP is seeded with a fully functional circuit. We assume that Pareto front has to be constructed for ten different error levels $e_1 \dots e_{10}$ expressed as a percentage of the Hamming distance. A two-stage procedure is used. In the first stage, the goal is to evolve an approximate circuit showing desired error e_i providing that a 5% difference is tolerated with respect to e_i. In the second stage, the fitness function reflects not only the error, but also the area and delay. Each objective is normalized to the interval $< 0, 1 >$ and multiplied with weights w_e, w_a and w_d, respectively ($w_e = 0.12$, $w_a = 0.50$, $w_d = 0.38$). It is requested that the Error remains within 5% tolerance with respect to e_i. Candidate circuits violating this condition are discarded.

The proposed method was evaluated using the circuits having more than 25 inputs and consisting of more than 150 gates taken from LGSynth, ITC and ISCAS benchmark libraries. According to the experiments conducted on Intel Xeon X5670@2.93 GHz, the average time spent in the first stage is less than 1 second (hundreds to few thousands of generations are required to get an approximate circuit showing desired error). Compared to the truth table based representation, employing a BDD package in the fitness function enabled to enourmously reduce the fitness evaluation time. The average time needed to calculate the Hamming distance is in milliseconds for most cases.

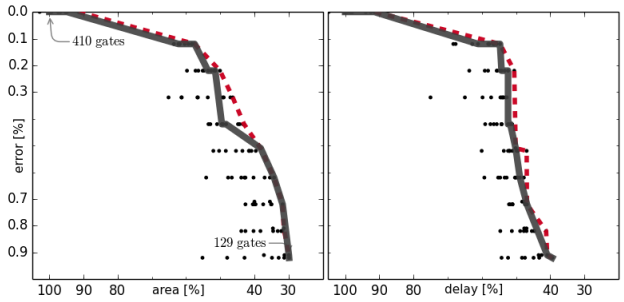

Figure 2: Pareto fronts for clmb circuit (46 inputs, 33 outputs)

The resulting Pareto front for clmb circuit calculated from 10 independent 30 minute evolutionary runs is shown in Fig. 2 (solid line). Two plots are presented – the best obtained area vs. error and delay vs. error, relatively to the fully functional optimized circuit. The result of a single CGP run is shown using a black dot. The local Pareto fronts in which the third objective (either area or delay) is ignored is shown using dash lines. The area was reduced by 38% (61%) by increasing the error to 0.1% (0.9%).

Acknowledgment

This work was supported by the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

4. REFERENCES

[1] M. Duranton, K. DeBosschere, A. Cohen, J. Maebe, and H. Munk. Hipeac vision 2015. Technical report, HiPEAC Network of Excellence, 2015.

[2] J. F. Miller. *Cartesian Genetic Programming.* Springer-Verlag, 2011.

[3] L. Sekanina and Z. Vasicek. Approximate circuits by means of evolvable hardware. In *2013 IEEE International Conference on Evolvable Systems*, Proceedings of the 2013 IEEE Symposium Series on Computational Intelligence (SSCI), pages 21–28. IEEE CIS, 2013.

[4] Z. Vasicek and L. Sekanina. Evolutionary approach to approximate digital circuits design. *IEEE Trans. on Evolutionary Computation – in press*, pages 1–13, 2015.