Logic Level Delay Modeling

Amit Narayan (anarayan@ic.eecs.berkeley.edu)

EE241 Interim Project Report

Logic Level Delay Modeling

Abstract

Performance optimization at the logic level requires accurate delay models which take into account both the gate-delay as well as the interconnect delay. Present day logic synthesis systems employ simplistic unit-delay model for the gates and largely ignore the effects of interconnect delay. In this paper we propose a new delay model which gives a more accurate estimate of the circuit delay. For gate delay, we model the factored form expression of a multilevel gate as an R-C network and calculate its Elmore delay. This calculation is implicitly done on the factored form expression and is efficient. Further, we identify the statistically significant parameters that effect the interconnect delay. We construct an explicit factor model for the interconnect delay based on these parameters. We believe that this is the first rigorous attempt to include interconnect delay during logic synthesis for performance.

1 Introduction

The design of a VLSI chip consists of a number of steps. At first, the design is specified at the behavioral level. This is followed by functional partitioning, architectural synthesis, logic synthesis, floorplanning, placement and routing and ends with a detailed specification of the layout mask level geometries. At each of these steps, various optimizations are done (manually or using CAD tools) to minimize area, increase performance or reduce power consumption. Typically, these optimizations proceed as a series of transformations to the design representation which are believed to improve the quality of the overall chip. Incorrect decisions made at a higher level of abstraction may lead the design so far away from the globally optimum solution that the subsequent stages of optimization may never be able to find it. In order to properly guide the optimization process, we need a model of the design which accurately reflects the cost of the final implementation taking into account the effects of subsequent steps of the design.

Performance optimization of circuits at the logic level is an important step in the automatic design of an integrated circuit. This step can be divided into two distinct phases. The first phase performs technology-independent transformations on a on the logic equations (which are described as a multilevel network) and the second phase performs performs technology-dependent mapping to implement the design using gates in the target technology. In this paper, we study the technology independent phase of performance optimization and present a model which can be used to predict the effects of various transformation on the overall delay of the circuit. The overall delay of a circuit can be divided into the following two components:

- active gate delay
- interconnect delay

Most of the present work has focussed on the active gate delays. The two most popular models are: 1) unit-delay model with a unit-fanout term and 2) library model. The unit-delay model besides being fast to evaluate has little justification. The delay under this model is expressed in arbitrary units making it difficult to relate it to the designer's constraints. Arbitrary units also make it difficult to compare the gate delay with interconnect delay. In the library model, the delay through a node is calculated by mapping it into cells in the target library. While this is more accurate, it takes much longer to run. Since this procedure has to be called many times during the optimization phase it can be prohibitively time consuming. As far as interconnect delay is concerned, very little attention has been paid so far towards rigorously modeling it at the logic level. But with decreasing feature sizes, this component of delay is becoming almost as important as the gate delay if not more. Ignoring this aspect of delay during logic synthesis may generate designs which are very difficult to route and hence have excessive interconnect delays. To avoid this, it is essential to incorporate the interconnect delay information during the logic synthesis phase.

In this work, we propose a more realistic model for the gate delay. We represent factored form expression for the gate function as an RC interconnect network and calculate its Elmore delay. Each transistor is replaced by a resistor and a capacitor. For a given technology, the values of these resistors and capapcitors are calculated by using standard statistical regression techniques. Not only does this allow us to model the delay more accurately but also we get the result in real time units so that it can be compared with the interconnect delay. Further, we identify the statistically significant parameters which effect the interconnect delay. Based on these parameters, we construct an explicit factor model for the interconnect delay. We believe that this is the first attempt which tries to rigorously model the interconnect delay at the logic level and gives a capability to make decisions which involve trade-off between the gate-delays and the interconnect delays.

In Section 2, we look at the previous work that address the problem of modeling gate and interconnect delay at the logic level and in Section 3 describe our approach.

2 Previous Work

Most commonly used logic synthesis tools model only the gate (or active cell) delays. The following three models are the most popular [?]:

- a unit delay model
- a unit fanout model
- a library model

In the unit delay model, the node expression is factorized into two input gates. Each two input gate is assumed to have a delay of one unit and the overall gate delay is found by adding the number of gates in the longest path from the node inputs to the outputs. The unit fanout model refines the unit delay model by adding a constant factor for each fanout of the node. As already mentioned, both of these models express delays in arbitrary units which cannot be easily related to the designer's constraints. In the library model, the delay through a node is calculated by mapping it into cells in the target library. This is a time intensive procedure which making it difficult to use it as an estimator in the inner loop of a performance optimization package where thousands of such calls need to be made. In [11] another model for estimating the gate delays for technolgy independent equations was proposed. In this a logarithmic model for the gate delay was proposed but again very little physical justification was provided for the use of such a model.

As we mentioned in the previous section, very little work has been done to model the interconnect delay at the logic level. This is mainly because very little information about the interconnect is available at the logic level. Some studies have been conducted which attempt to incorporate the effects of physical design at logic and higher levels of synthesis. We will briefly review some of them here.

The high-level layout area and delay estimation techniques can be broadly divided into two categories:

- analytical
- constructive

In analytical techniques, a mathematical model is used to estimate the layout phase. Constructive models keep a crude placement solution and use that to evaluate the layout cost. Most of the techniques [8],[5],[4],[3] deal with datapaths at the register-transfer level. In [8] a bit-slice architecture is assumed to model the datapath. A companion placement solution along with an analytical tool is used to estimate the overall floorplan and dimension of the bit-slice. Controller is assumed to have a random logic and the layout cost estimation is done using a companion placement solution. It is recognized that this technique is not very effective for random logic. In [5] analytical formulas based on the connectivity information is used with a companion placement solution to estimate the dimensions of physical layout. They try to model the standard cell methodology by dividing the process in two phases. In the first phase all the cells are placed in a single row. All possible orientations are tried and the one with least connection length is selected. The connection length is calculated by using the connectivity information of different modules. In the second phase, the single row design is folded to get the desired aspect ratio. Additional wiring needed to connect different rows is calculated by using an analytical model. In this technique, large number of different orientations have to be considered and hence is not practical when the number of components is large as at the logic level. Further, the accuracy of the model goes down as the number of components increases. In [3] propose a probabilistic model for area estimation of VLSI layouts. Based on Rent's rule, a geometric distribution for the wirelengths is assumed. A model is constructed for the standard cell design style and analytical expressions are derived for estimating the layout area.

At the logic level, attempts have been made to model the effect of layout during both technolgy independent and technology mapping phases. In [7] a technology mapper, LILY is implemeneted which takes the interconnect delay into account. The interconnect cost is estimated using a placement solution and is incorporated in the standard tree mapping algorithm. In [9] a fanout optimization algorithm is proposed which maintains the order of the fanouts to simplify routing. This order is generated using a companion solution. An $O(n^3)$ alogrithm is proposed which results in an improvement of about 8-14%. Similar idea is used in [1] to minimize routing factor during logic synthesis. The approach is based on lexicograhical expression of Boolean function controlling input dependency. In a lexicographic expression, for a given sum of products form, all the literals respect the same order in each product term. Maintaining this order in all expressions results in a simpler layout which takes less area. An average improvement of more than 20% was reported but the experiments were performed only on a few examples. In [6] various algorithms used in conventional logic synthesis like kernel extraction and technology decomposition are modified to incorporate a layout cost factor. This cost factor is evaluated using a companion placement solution and average improvements of about 27% in area and 12% in delay are reported. In [10] a heuristic to minimize the layout cost is proposed which doesn't employ a companion placement solution. Although impressive gains are reported for some cases, the experiments designed to obtain these are not very realistic. All the input pins are located on one side of the chip and all the output pins are located on the other side of the chip. Their method is based on minimizing the average fanout range and evenly distributing the fanout range throught the chip. A rigourous analysis of the effect of this cost function on the overall cost is not performed. In [2] an interesting approach of combining logic synthesis and routing is presented for FPGAs. Given a network, redundancy addition/removal techniques are used to identify alternative wires for every wire. Wires which have large number of alternative wires are given lower priority during routing and wires which don't have any alternatives are routed earlier. By combining synthesis and routing in this manner, they are able to complete routing of many circuits for the first time.

3 Our Approach

Our approach consists of accurately modeling both the gate as well as the interconnetc delay. We propose to do the following in this project

- For a given factored form expression of a node calculate the Elmore delay. This will be done by assuming an underlying transistor level implementation of this expression. This underlying transistor level circuit will be further approximated by an RC-circuit. We will use the standard switch level techniques to estimate the Elmpre delay. For a given factored form expression, this delay will be in terms of 'R' and 'C' of a given technology. Values of 'R' and 'C' will be evaluated using standard regression techniques for a given library. We hope that the Elmore delay for a given factored form expression can be done impicitly by just working on the expression and hence will not be prohibitively time consuming. Also, since a more realistic model is being used, it should be more accurate than the unit-delay model. Regression, ofcourse has to be performed only once and hence can be done off-line.
- We will experiment with different technologies and see the percentage of interconnect delay in the overall delay. We will try to identify at what device sizes does the interconnect effect starts to dominate.
- We will identify statistically significant logic level parameters using standard regression techniques.
- Finally, we will construct an explicit factor model for the interconnect delay using the statistically significant parameters identified in the previous phase.

References

[1] P. Abouzeid, K. Sakouti, G. Saucier, and F. Poirot. multilevel synthesis minimizing the routing factor. In *Proc. of the Design Automation Conf.*, pages 365–368, June 1990.

[2] S-C Chang, K-T Cheng, N-S Woo, and M. Marek-Sadowska. Layout Driven Logic Synthesis for FPGAs. In *Proc. of the Design Automation Conf.*, pages 308–313, June 1994.

[3] F. J. Kurdahi and A. C. Parker. Techniques for Area Estimation of VLSI Layouts. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 8(1):81–92, January 1989.

[4] T. A. Ly, W. L. Elwood, and E. F. Girczyc. A Generalized Interconnect Model for Data Path Synthesis. In *Proc. of the Design Automation Conf.*, pages 168–173, June 1990.

[5] M. Nourani and C. Papchristou. A Layout Estimation Algorithm for RTL Datapaths. In *Proc. of the Design Automation Conf.*, pages 285–291, June 1993.

[6] M. Pedram and N. Bhat. Layout Driven Logic Restructuring/Decomposition. In *Proc. of the Intl. Conf. on Computer-Aided Design*, pages 134–137, November 1991.

[7] M. Pedram and N. Bhat. Layout Driven technology Mapping. In *Proc. of the Design Automation Conf.*, pages 99–105, June 1991.

[8] C. Ramachandran, F. J. Kurdahi, D. D. Gajski, A. C.-H Wu, and V. Chaiyakul. Accurate Layout and Delay Modeling for System Level Design. In *Proc. of the Intl. Conf. on Computer-Aided Design*, pages 355–361, November 1992.

[9] H. Vaishnav and M. Pedram. Routability-Driven Fanout Optimization. In *Proc. of the Design Automation Conf.*, pages 230–235, June 1993.

[10] H. Vaishnav and M. Pedram. Minimizing the Routing Cost During Logic Extraction. In *Proc. of the Design Automation Conf.*, pages 70–75, June 1995.

[11] D. E. Wallace and M. S. Chandrasekhar. High-Level Delay Estimation for Technology-Independent Logic Equations. In *Proc. of the Intl. Conf. on Computer-Aided Design*, pages 188–191, November 1990.