13.5

Accurate Pre-layout Estimation of Standard Cell Characteristics

Hiroaki Yoshida
Zenasis Technologies, Inc.
Campbell, CA 95008-6900
hiroaki@zenasis.com

Kaushik De
Zenasis Technologies, Inc.
Campbell, CA 95008-6900
kaushik@zenasis.com

Vamsi Boppana
Zenasis Technologies, Inc.
Campbell, CA 95008-6900
vamsi@zenasis.com

ABSTRACT

With the advent of deep-submicron technologies, it has become essential to model the impact of physical/layout effects up front in all design flows [1]. The effect of layout parasitics is considerable even at the *intra-cell* level in standard cells. Hence, it has become critically important for any transistor-level optimization to consider the effect of these layout parasitics as an integral part of the optimization process. However, since it is not computationally feasible for the actual layout to be a part of any such optimization procedures, we propose a technique that estimates cell layout characteristics without actually performing the layout and subsequent extraction steps. We demonstrate in this work that it is indeed feasible to estimate the layout effects to get timing characteristics that are on average within about 1.5% of post-layout timing and that the technique is thousands of times faster than the actual creation of layout.

Categories and Subject Descriptors

B.7.2 [**Hardware**]: Integrated Circuits—*Design Aids*

General Terms

Algorithms, Measurement, Performance, Design

Keywords

Standard cell, cell characterization, transistor-level optimization

1. INTRODUCTION

Design optimization at the transistor-level is well-known and has been used successfully to achieve significant performance benefits above and beyond gate-level design optimization. The approaches range from transformations such as sizing [2][3], all the way to macro-cell based design methodologies [4]. More recently, transistor-level optimization techniques targeting a standard cell based design flow have also been proposed [5]-[7]. These optimization techniques take advantage of the recent progress in automated cell-layout solutions [8][9]. Such solutions are now being increasingly used in the creation of high-performance libraries and are equipped to handle a variety of manufacturing, design and cost considerations used in the creation of libraries [10].

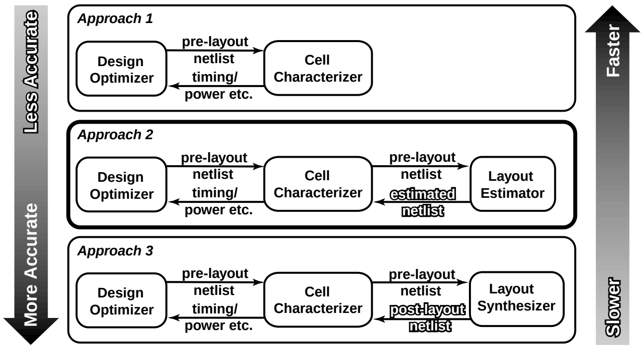


Figure 1: Transistor-level optimization approaches.

Traditionally, transistor-level optimization techniques such as [2] [11], have not attempted to account for the impact of layout parasitics, which has become increasingly important at the 130nm and at the 90nm process nodes [1]. The effect of these layout parasitics has become extremely important, even for circuits with a small number (∼10) of transistors, rendering any optimization without considering these effects to be impractical. In addition, reduced order device models such as switch-level (RC) models of transistors are becoming increasingly incapable of modeling deep submicron effects. This leaves detailed simulation, often at the BSIM3/4 level [12], or using detailed models built using simulation at that level, as the only reliable option for obtaining accurate circuit timing.

Hence, it has become critically important for any transistor-level optimization to consider the effect of these layout parasitics as an integral part of the optimization process. In other words, out of the design optimization approaches outlined in Figure 1, Approach 1 is not practical for deep submicron geometries. On the other hand, it is also not computationally feasible to wait for the actual layout to be a part of any such optimization procedures, as suggested by Approach 3 in Figure 1. Hence, we propose a technique that provides an accurate estimate of cell layout characteristics without actually performing the layout and subsequent extraction steps. This is Approach 2 outlined in Figure 1. We demonstrate in this work that it is indeed feasible to estimate the layout effects to get timing characteristics that are on average within about 1.5% of post-layout timing and the technique is thousands of times faster than the actual creation of layout.

DAC'04, June 7–11, 2004, San Diego, California, USA.

The rest of the paper is organized as follows. Section 2 introduces the layout model, circuit model and problem definition. Sections 3 and 4 describe the proposed statistical and constructive estimation techniques respectively and Section 5 presents experimental results on two industrial processes at the 130nm and 90nm process nodes. Conclusions are drawn in Section 6.

2. PRELIMINARIES

In this section, we first present our models of cells, layouts and circuits. We then describe several notions used throughout this paper. Finally, the problem addressed in this paper is formulated.

A *cell* refers to a typical standard cell. Figure 2 illustrates our cell layout model. A cell is of a single-height style, *i.e.*, P-type and N-type transistors are placed in diffusion regions as shown. No transistor stacking is allowed. The region between the two diffusion regions is called as the diffusion gap region. Maximum diffusion heights are either fixed or flexible, depending on the layout style. Transistors are connected by using either wires or diffusion.

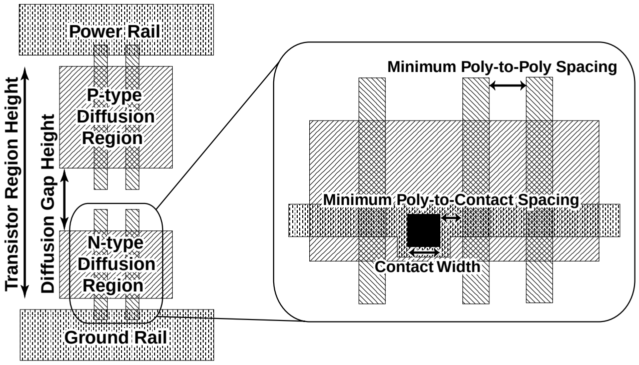
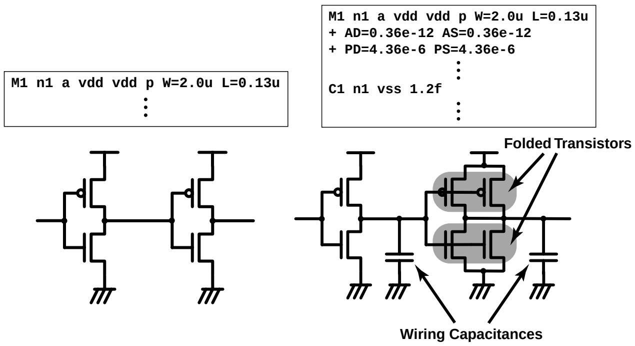


Figure 2: Our layout model.

Figure 3 shows our circuit models. A *pre-layout netlist* is a set of transistors and nets that connect transistors. Each transistor in a pre-layout netlist has its length and width. An *estimated netlist* is defined as a pre-layout netlist with the following modifications. Each transistor has the areas and perimeters of its drain and source diffusion regions in addition to its width and length. Each net has an associated grounded capacitance. An estimated netlist is functionally identical to the corresponding pre-layout netlist but can be structurally different due to transistor folding, *i.e.*, a wide transistor in a pre-layout netlist may be split into smaller parallel-connected transistors in the estimated netlist.



```
M1 n1 a vdd vdd p W=2.0u L=0.13u
          ⋮
```

```
M1 n1 a vdd vdd p W=2.0u L=0.13u
+ AD=0.36e-12 AS=0.36e-12
+ PD=4.36e-6 PS=4.36e-6
          ⋮
C1 n1 vss 1.2f
          ⋮
```

(a) Pre-layout netlist. **(b) Estimated netlist.**

Figure 3: SPICE-level and schematic-level descriptions of our circuit models.

A *Maximal Transistor Series (MTS)* is a maximal set of series-connected transistors. In a physical layout, an MTS is typically implemented as transistors that are connected to each other by diffusion. An *intra-MTS net* is a net that connects two transistors in an MTS and an *inter-MTS net* is a net that connects transistors in different MTSes. Figure 4 shows an example of an MTS and its physical implementation. MTS identification is an essential step in building a high-quality estimator.

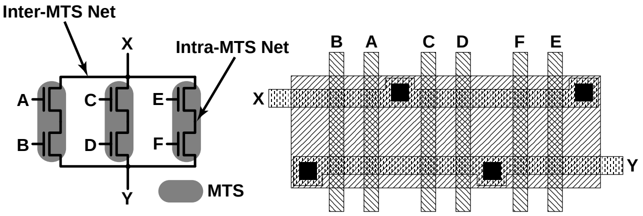


Figure 4: Maximal Transistor Series (MTS) and its physical implementation.

Cell characterization refers to the process of determining various characteristics (such as timing, power, input capacitances) of the cell. This process is used to create views/models of the cell that are used in various steps of the design flow. Of particular interest in this paper is the aspect of cell *timing* characterization. It refers to the process of creating models such as the non-linear delay model [13] based on detailed SPICE simulation of the transistor-level circuit representation of the cell. Cell timing represents four different timing characteristics, *i.e.*, cell rise, cell fall, transition rise and transition fall, for a pre-defined set of output loads and input slews, on every signal-carrying input-to-output path in the circuit.

Without loss of generality, *timing* $T(c)$ of a cell c can be defined as a delay or slew of an arc of the cell for a given output load and input slew. *Pre-layout timing* $T_{pre}(c)$, *estimated timing* $T_{est}(c)$ and *post-layout timing* $T_{post}(c)$ are the timing values that are obtained by characterizing a pre-layout netlist, estimated netlist and post-layout netlist, respectively.

The problem addressed in this paper can be formulated as follows: *Given a cell c and its pre-layout netlist, find an estimated timing such that the absolute difference of the estimated timing and post-layout timing:*

$$D(c) = \left|T_{est}(c) - T_{post}(c)\right| \qquad (1)$$

is minimized.

3. STATISTICAL ESTIMATOR

One can estimate post-layout timing from pre-layout timing based on statistical analysis of differences between pre-layout and post-layout timing. The *statistical estimator* estimates the post-layout timing of a cell by multiplying pre-layout timing by a predetermined scale factor S.

$$T_{est}(c) = S \cdot T_{pre}(c) \qquad (2)$$

S is specific to each technology and cell architecture, and is determined based on a small representative set of cells that are actually laid out and characterized with respect to timing. More specifically, the scale factor is calculated as follows:

$$S = \frac{1}{|C|} \sum_{c \in C} (T_{post}(c)/T_{pre}(c)) \qquad (3)$$

where C is a representative set of cells.

The advantage of this estimator is that it is applicable to any technology and cell architecture because it is formulated in a technology-independent manner. On the other hand, the disadvantage is its inaccuracy mainly due to the lack of consideration of the variation of layout characteristics. The experimental results in Section 5 show that this approach is not accurate enough. In the next section, we'll introduce the constructive estimator that takes such variations into account.

4. CONSTRUCTIVE ESTIMATOR

The central problem with the statistical estimator is that it cannot accurately capture the variation of layout characteristics present in different standard cells even from within the same standard cell library. In this section, we introduce the *constructive estimator* that takes such variations into account. The constructive estimator constructs an estimated netlist by applying the following transformations to a pre-layout netlist: a) folding each transistor, b) assigning diffusion areas and perimeters to each transistor, and c) adding a wiring capacitance to each net. Then estimated timing is obtained by characterizing the estimated netlist. These transformations are described in detail in the following subsections.

4.1 Transistor Folding

Since the height of a cell is fixed, a wide transistor in a pre-layout netlist is divided into smaller transistors to meet the cell height. Folded transistors are connected in parallel to preserve the original functionality as illustrated in Figure 3.

The folded transistor width W_f and the number of the folded transistors N_f are calculated as follows:

$$W_f(t) = W(t)/N_f(t) \tag{4}$$

$$N_f(t) = \lceil W(t)/W_{fmax}(t) \rceil \tag{5}$$

$$W_{fmax}(t) = \begin{cases} R(H_{trans} - H_{gap}) & \text{if } t \text{ is P-type} \\ (1 - R)(H_{trans} - H_{gap}) & \text{if } t \text{ is N-type} \end{cases} \tag{6}$$

where $W(t)$ is the width of a given transistor t, $\lceil x \rceil$ denotes the smallest integer greater than or equal to x, R is the ratio of heights of the P and N diffusions, H_{trans} is the height of a transistor region and H_{gap} is the height of a diffusion gap region as shown in Figure 2.

Our model allows two transistor folding styles, a *fixed P/N ratio* style and an *adaptive P/N ratio* style. In the fixed P/N ratio style, R is specific to a given technology and cell architecture. In the adaptive P/N ratio style, R is specific to a cell and is determined such that the width of the cell is minimized:

$$R = \sum_{t \in P(c)} W(t) \Bigg/ \sum_{t \in P(c) \cup N(c)} W(t) \tag{7}$$

where $P(c)$ is a set of P-type transistors in a cell c and $N(c)$ is a set of N-type transistors.

4.2 Diffusion Area and Perimeter

Given the width w and height h of the diffusion region of a transistor, the diffusion area A and perimeter P are calculated as follows:

$$A = wh, P = 2w + 2h \tag{8}$$

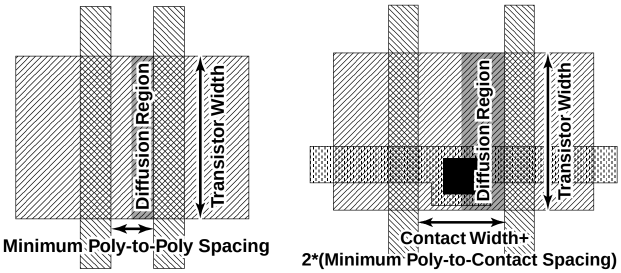
The height of a diffusion region is estimated as the width of the associated transistor t:

$$h = W(t) \tag{9}$$

The width of a diffusion region is estimated by using one of the following formulas depending on whether the net n that is associated with the diffusion is an intra-MTS net or an inter-MTS net.

$$w = \begin{cases} S_{pp}/2 & \text{if } n \text{ is intra-MTS net} \\ W_c/2 + S_{pc} & \text{if } n \text{ is inter-MTS net} \end{cases} \tag{10}$$

where S_{pp} is the minimum poly-to-poly spacing, W_c is the contact width and S_{pc} is the minimum poly-to-contact spacing. These are given as design rules as shown in Figure 2. Figure 5 illustrates the estimation of diffusion area and perimeter. It is important to note the role of an MTS in this computation. It is the MTS that typically controls diffusion sharing and hence controls the diffusion parasitics. This is the key to getting an accurate estimate of the diffusion parasitics.



(a) Intra-MTS **(b) Inter-MTS**

Figure 5: Estimation of diffusion area and perimeter.

Note that this transformation must be done after transistor folding because the widths of transistors may be different before and after transistor folding. It should also be noted that diffusion area and perimeter modeling should be made in conjunction with the transistor models of the target technology.

4.3 Wiring Capacitance

This transformation adds a wiring capacitance to each net in a pre-layout netlist. Intra-MTS nets are not considered because they are typically implemented in diffusion. Similar to diffusion area/perimeter estimation, this transformation must be done after transistor folding.

The capacitance $C(n)$ of a net n is estimated by the following formula:

$$C(n) = \alpha \sum_{t \in TDS(n)} \left| MTS(t) \right| + \beta \sum_{t \in TG(n)} \left| MTS(t) \right| + \gamma \tag{11}$$

where α, β and γ are constants, $TDS(n)$ is a set of transistors whose drain or source is connected to a net n, $TG(n)$ is a set of transistors whose gate is connected to a net n and $MTS(t)$ is an MTS that includes a transistor t. According to our experiments presented in Section 5, the above formula gives an excellent correlation to actual wiring capacitances. Again, it is important to note the importance of MTS in obtaining an accurate estimate of wiring capacitance. It is the MTS connectivity that typically dictates the length of the wire and hence the capacitance of the wire.

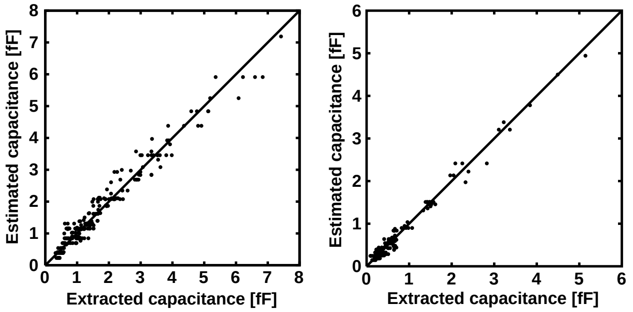
Formula (11) requires three constants α, β and γ to be determined in advance. These constants are determined by multiple regression analyses based on a small representative set of cells that are actually laid out. This calibration process has to be done only once for a given technology and cell architecture.

5. EXPERIMENTAL RESULTS

The proposed technique has been implemented within the framework of a standard cell characterization flow. Given a pre-layout netlist, cell timing is generated based on both *statistical* and *constructive* estimations and compared with post-layout timing. As

Table 1: Quality of proposed estimation techniques for two industrial standard cell libraries.

Feature Size [nm]	*#Cells*	*#Wires*	*No Estimation*		*Statistical Estimator*		*Constructive Estimator*	
			Avg. Abs. Diff. [%]	*Std. Dev. [%]*	*Avg. Abs. Diff. [%]*	*Std. Dev. [%]*	*Avg. Abs. Diff. [%]*	*Std. Dev. [%]*
130	57	276	8.85	4.08	3.60	2.76	1.55	1.79
90	53	221	8.81	4.80	4.10	3.35	1.52	1.40



(a) 130nm technology **(b) 90nm technology**

Figure 6: Extracted versus estimated capacitances.

discussed in Section 4, for the statistical technique, cell timing is generated based on calibrating simulation results from a pre-layout netlist. For the constructive technique, the estimated netlist (with transistor folding, diffusion area/perimeter and wiring capacitances) is simulated.

We conducted experiments on two different state-of-the-art standard cell libraries implemented in 130nm and 90nm technologies. These libraries were chosen at different process nodes and from different vendors in order to measure the effectiveness of the techniques across varying layout styles and design rules. The cells vary from simple cells such as an inverter to complex cells that consist of ~30 unfolded transistors. The simulator used in these experiments was *HSPICE*[14].

First, we present results that demonstrate the effectiveness of the constructive technique on estimating wiring capacitances. Wiring capacitances critically determine the quality of the constructive estimator due to their increased effects at the deep submicron geometries. Figure 6 shows scatter plots that compare extracted and estimated wiring capacitances for the cells in the 130nm and 90nm technologies, respectively. The extracted capacitance values are calculated from lumped C extracted netlists. The three constants α, β and γ in Formula (11) for these technologies are calculated by multiple regression analyses. This data shows the excellent correlation achieved by our wiring capacitance estimation technique.

Finally, Table 1 demonstrates the effectiveness of the estimators on overall cell timing for the two standard cell libraries under consideration. All four kinds of cell delays (cell rise, cell fall, transition rise, transition fall) are measured in this experiment. Columns 1, 2 and 3 show the feature size of the library, the number of cells used in this experiment and the number of wires whose capacitances are estimated in this experiment. The rest of the columns compare the quality of the cell timing for each of the proposed techniques. For example, for the 90nm technology library, if no estimation was used, the average of the absolute differences in timing is 8.85% and a standard deviation of 4.08%. The statistical approach produces an average absolute difference of 4.10% and a standard deviation of 3.35%. And finally, the best results are obtained by the constructive estimator, with an average absolute difference of 1.52% and a standard deviation of 1.40%. It is worth noting that the runtimes of the constructive estimators are very small, with typical overheads being less than 0.1% of typical SPICE simulation times.

6. CONCLUSIONS

We have demonstrated that it is feasible to estimate timing characteristics of transistor-level circuits in a standard-cell design framework to on average within about 1.5% of post-layout timing based on a fast and accurate *constructive* estimation technique. This technique solves a critical problem affecting transistor-level optimization techniques at deep submicron geometries. It has been implemented in an industrial software system and has been successfully used at today's leading edge process nodes.

7. REFERENCES

[1] *International Technology Roadmap for Semiconductors 2002 Update*, Semiconductor Industry Association, 2002.

[2] J. P. Fishburn and A. E. Dunlop, “TILOS: A posynomial programming approach to transistor sizing,” in *Proc. of IEEE International Conference on Computer-Aided Design*, pp. 326–328, Nov. 1985.

[3] A. Conn *et al.*, “Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 452–459, Jun. 1999.

[4] J. L. Burns and J. A. Feldman, “C5M — A Control- Logic Layout Synthesis System for High-Performance Microprocessors,” *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 1, pp. 14–23, Jan. 1998.

[5] R. Panda *et al.*, “Migration: A new technique to improve synthesized designs through incremental customization,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 388–391, Nov. 1998.

[6] M. Cote and P. Hurat, “Faster and Lower Power Cell-Based Designs with Transistor-Level Cell Sizing,” in *Closing the Gap Between ASIC & Custom*, pp. 225–240, Kluwer Academic Publishers, 2002.

[7] D. Bhattacharya and V. Boppana, “Design Optimization with Automated Flex-Cell Creation,” in *Closing the Gap Between ASIC & Custom*, pp. 241–268, Kluwer Academic Publishers, 2002.

[8] M. Guruswamy *et al.*, “CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 327–332, Jun. 1997.

[9] *abraCAD Documentation*, Synopsys, Inc., 2003.

[10] C. Bittlestone *et al.*, “Architecting ASIC Libraries and Flows in Nanometer Era,” in *Proc. of ACM/IEEE Design Automation Conference*, pp. 776–781, Jun. 2003.

[11] S. S. Sapatnekar *et al.*, “An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization,” *IEEE Trans. on Computer-Aided Design*, vol. 12, no. 11, pp. 1621–1634, Nov. 1993.

[12] W. Liu *et al.*, *BSIM3v3.2 MOSFET Model Users' Manual*, University of California, Berkeley, 1998.

[13] *Library Compiler User Guide: Modeling Timing and Power Technology Libraries*, Synopsys, Inc., 2003.

[14] *HSPICE Data Sheet*, Synopsys, Inc., 2003.