Gate Sizing Minimizing Delay and Power/Area

Gracieli Posser, Guilherme Flach, Gustavo Wilke, Ricardo Reis
{gposser,gaflach,wilke,reis}@inf.ufrgs.br

Universidade Federal do Rio Grande do Sul (UFRGS)
Instituto de Informática – PPGC/PGMicro
Av. Bento Gonçalves 9500       Porto Alegre, RS - Brazil

Abstract

In this work we present a gate sizing tool based on Geometric Programming. The optimization can be done targeting both, delay and power/area minimization. In order to qualify our approach, the ISCAS’85 benchmark circuits are mapped for 350nm and 45nm technologies using typical standard cell libraries. Next, the mapped circuit is sized using our tool and the result is compared to the original mapped circuit. The speed is increased by 21% and 4.5%, on average, for 45nm and 350nm technology, respectively, keeping the same area and power values of the sizing provided by standard-cells library. For power/area optimization, where the delay was restricted to the delay value found at delay minimization, the reduction was 28.2% in area and 27.3% in power consumption, on average, considering 45nm technology and 29.9% in area and 28.5% in power, on average, considering 350nm technology.

1. Introduction

One can make a circuit faster or consume less power by sizing its logic gates properly. Sizing a gate means that the transistors, which compose the gate, are made shorter or larger according to a scale factor.

There is an optimal scale factor for each gate, considering that increasing the size of the gate and, consequently, of the transistors, increases their ability to carry a load, reducing the time required for the gate switching its signal. However, by increasing the port size reduces its output resistance, and increases its input capacitance, giving its driver higher capacitive load.

The gate sizing tool developed in this work is able to handle CMOS circuits and can be configurable to several CMOS manufacturing technologies. Moreover, the optimization can be done in two ways: (1) delay minimization subject to an area constraint and (2) area minimization subject to a delay constraint. Elmore Delay [1] is used in order to model the gate sizing problem as a Geometric Program (GP) [2]. GP is a mathematical optimization problem that can be efficiently solved in polynomial time guaranteeing that the optimal solution is found if one exists.

The contributions of this work are:
- a gate sizing tool based on GP that can be used along with an automatic cell layout generation tool taking advantage of “continuous” gate sizing avoiding the rounding-off problems;
- a gate sizing configurable to several manufacturing technology by changing the technology parameters;
- a gate sizing tool targeting both, delay and area;

The paper is organized as follows. On section II, some related works are shown. We show the problem formulation on section III. The section IV presents the gate sizing development using GP. On section V, we show the results and, finally, on section VI, we present our conclusions.

2. Related Works

Gate-sizing problem has been studied in many papers using different ways to solve it. The most widely known is the logical effort method [4], which provides fast heuristics or design guidelines for solving the gate-sizing problem approximately. Linear Programming is used in [5] and [6]. In [7] and [8] is used Non-Linear Programming. [9] are concerned about scalability of the circuit, i.e., the execution time needed to size large circuits. The traditional gate sizing methodologies [10], [3], [2] use Elmore delay to model delay as posynomial functions allowing the gate sizing to be formulated as a Geometric Program.

This work combines the ideas of the works [3] and [2]. [3] presents a method for transistor sizing, where the sizing problem is modeled and solved by GP. The gates are modeled using the Switch-Level RC Gate Model. In this model, a gate is viewed as a set of RC trees one for each possible input vector and the gate delay is the maximum delay generated by its compound RC trees. The RC tree is built by replacing turned-on transistor by an equivalent resistance. Node capacitances is composed by source-to-bulk and drain-to-bulk capacitances. A load capacitance is connected to the output node of the gate.

Work [2] shows a gate sizing method, where a variable $X_i$ is associated for each logical gate. The scale factors are the optimization variables of the GP. The gate delay is estimated by a linear function on the scale
factors. Circuit area is the sum of the area of each port that make up the circuit. The path delays are given by the RC product and circuit delay is the maximum delay among all paths of the circuit.

3. Problem Formulation

This work supports two different formulations, delay minimization subject to an area constraint and area minimization subject to a delay constraint. The two formulations are shown below.

3.1. Minimizing Delay Subject to Maximum Area

When the objective is minimize delay, the optimization problem is formulated as following:

\[
\begin{align*}
\text{minimize} & \quad D = \max(D_1, \ldots, D_n) \\
\text{subject to} & \quad X_{\text{min}} \leq X_i \leq X_{\text{max}} \\
& \quad C_{\text{in}} \leq C_{\text{in}}^{\text{max}} \\
& \quad A \leq A_{\text{max}}
\end{align*}
\]

where \( D \) values are the delay of the circuit paths. The \( X_{\text{min}} \) and \( X_{\text{max}} \) are the minimum and maximum size of the gate. \( C_{\text{in}}^{\text{max}} \) is the maximum input capacitance acceptable to the circuit and \( A_{\text{max}} \) is the maximum circuit area.

3.2. Minimizing Area Subject to Maximum Delay

For this optimization, we make a change, the objective function becomes the area and delay becomes a constraint. The formulation is the following:

\[
\begin{align*}
\text{minimize} & \quad \text{Area} \\
\text{subject to} & \quad X_{\text{min}} \leq X_i \leq X_{\text{max}} \\
& \quad C_{\text{in}} \leq C_{\text{in}}^{\text{max}} \\
& \quad D \leq D_{\text{max}}
\end{align*}
\]

4. Gate Sizing Development

It can be said that the gate sizing problem is the problem of choosing the scale factors in order to find the minimum delay (area) subject to limits on the total area (delay) and others constraints.

Our gate sizing tool was developed as follow:
1) The logic gates are modeled using the Switch-Level RC Gate Model [3].
2) For each port is set a scale factor that multiplies the transistor widths.
3) Capacitance and resistance values used to calculate delay and power are obtained throughout SPICE simulations for PMOS and NMOS transistors. The capacitances that compose a gate are proportional to the scale factor and the driving resistance is approximately inversely proportional.
4) The delay is calculated using the Elmore delay model, which produces posynomial functions, enabling the problem solution by Geometric Programming.
5) Circuit delay is the maximum delay among all circuit paths.
6) The area of a scaled gate \( i \) is proportional to the gate scale factor \( X_i \), where, \( n \) is the number of gates of the circuit and \( A_{\text{base}}(Y_i) \) is the area base of the gate \( i \), as shown below:

\[
A_{\text{total}} = \sum_{i=1}^{n} X_i * A_{\text{base}}(Y_i)
\]

7) The power is calculated considering only switching. It is made using the equation (4), where, \( C_{\text{load}} \) is the load capacitance of the circuit. \( C_{\text{in}} \) is the input capacitance of each gate of the circuit. \( Vdd \) is 1.1V for 45nm technology and 3.3V considering 350nm technology. \( \alpha \) is the probability of switching, we considerate that the circuit switching 20% of the time and \( f \) is the clock frequency and it was set 500MHz for our test cases.

\[
P = (C_{\text{load}} + \sum_{i=1}^{n} C_{\text{in}} Y_i) * Vdd^2 * \alpha * f
\]

5. Results

[11] highlighted that once the minimum delay is achieved, a new optimization program may be performed targeting area minimization using the minimum delay as constraint. This allows the area to be further minimized since the former problem are not concerned in area minimization. This approach is used in this work. First, the delay is minimized. Then, the area is further reduced with no delay penalty.

We use a set of the ISCAS’85 benchmark circuits for our tests. Initially, the circuit was mapped using RTL Compiler tool from Cadence to 45nm library. Design Compiler from Synopsys was used to 350nm library. In both, only CMOS cells were considerate. These circuits are inserted to our sizing tool where the area, timing and power values were calculated. We considerate as load capacitance a value six times greater than the input capacitance of a gate not scaled, i.e., with scale factor one.
5.1. Results for 45nm Technology

Tab. 1 shows the values and reductions (R) considering the sizes found in a standard cell library (SC) and the sizes given by our gate sizing using Geometric Programming (GP). The gate sizing in Tab. 1 is a delay minimization, where the area is restricted to the same area of the circuit using standard cells. The circuit sized using our methodology (GP) obtained a reduction, on average, of 21% in delay, keeping the same area and power values of the sizing provided by standard-cells library.

Tab. 1 – Comparison results between standard cells (SC) sizing and sizing using Geometric Programming (GP) proposed in this work to 45nm minimizing delay subject to area

<table>
<thead>
<tr>
<th># Gates</th>
<th>Power (µW)</th>
<th>SC sizing</th>
<th>GP sizing</th>
<th>R (%)</th>
<th>Timing (ps)</th>
<th>SC sizing</th>
<th>GP sizing</th>
<th>R (%)</th>
<th>Area (µm²)</th>
<th>SC sizing</th>
<th>GP sizing</th>
<th>R (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C432</td>
<td>184</td>
<td>22.2</td>
<td>22.4</td>
<td>-0.9</td>
<td>718</td>
<td>666</td>
<td>7.3</td>
<td>210.4</td>
<td>210.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C499</td>
<td>403</td>
<td>58.3</td>
<td>58.4</td>
<td>-0.2</td>
<td>750</td>
<td>651</td>
<td>13.1</td>
<td>536.4</td>
<td>536.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1908</td>
<td>259</td>
<td>33.6</td>
<td>33.7</td>
<td>-0.3</td>
<td>472</td>
<td>425</td>
<td>10.0</td>
<td>304.3</td>
<td>304.3</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C880</td>
<td>232</td>
<td>31.4</td>
<td>31.1</td>
<td>1.1</td>
<td>451</td>
<td>330</td>
<td>26.8</td>
<td>281.0</td>
<td>277.4</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apex1</td>
<td>1728</td>
<td>239.8</td>
<td>239.5</td>
<td>0.1</td>
<td>673</td>
<td>504</td>
<td>25.2</td>
<td>2304</td>
<td>2296</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apex2</td>
<td>4110</td>
<td>527.1</td>
<td>523.6</td>
<td>0.7</td>
<td>863</td>
<td>650</td>
<td>24.7</td>
<td>5180</td>
<td>5145</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apex3</td>
<td>1939</td>
<td>254.3</td>
<td>251.9</td>
<td>0.9</td>
<td>687</td>
<td>507</td>
<td>26.3</td>
<td>2441</td>
<td>2413</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apex5</td>
<td>1942</td>
<td>264.6</td>
<td>258.3</td>
<td>2.4</td>
<td>662</td>
<td>431</td>
<td>34.9</td>
<td>2512</td>
<td>2446</td>
<td>2.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>1350</td>
<td>178.9</td>
<td>177.3</td>
<td>0.5</td>
<td>660</td>
<td>521</td>
<td>21.0</td>
<td>1721</td>
<td>1704</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 2 shows the values for gate sizing minimizing area/power subject to the delay to the minimum value found by the delay minimization, shown in Tab. 1. Tab. 2 shows area and power values given by delay minimization (Min. Delay) and by area minimization (Min. area) and their reductions. Area minimization allowed a reduction, on average, 28.2% in area and 27.3% in power consumption, considering the same delay value given by delay minimization, Tab. 1. This improvement is possible when there are multiple optimal points that minimize the delay, given an area budget. So, the area minimization problem is able to find the minimum area considering the minimal circuit delay.

Tab. 2 - Results for gate sizing minimizing area subject to delay (min. Area) compared to the values from delay minimization (min. Delay) for 45nm

<table>
<thead>
<tr>
<th># Gates</th>
<th>Min. Delay</th>
<th>Min. Area</th>
<th>Reduction (%)</th>
<th>Min. Delay</th>
<th>Min. Area</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C432</td>
<td>22.4</td>
<td>22.4</td>
<td>0.00</td>
<td>210.4</td>
<td>210.4</td>
<td>0.00</td>
</tr>
<tr>
<td>C499</td>
<td>58.4</td>
<td>58.4</td>
<td>0.00</td>
<td>536.4</td>
<td>536.4</td>
<td>0.00</td>
</tr>
<tr>
<td>C1908</td>
<td>33.7</td>
<td>33.7</td>
<td>0.00</td>
<td>304.3</td>
<td>304.3</td>
<td>0.00</td>
</tr>
<tr>
<td>C880</td>
<td>31.1</td>
<td>20.2</td>
<td>34.9</td>
<td>277.4</td>
<td>171</td>
<td>38.4</td>
</tr>
<tr>
<td>apex1</td>
<td>239.5</td>
<td>137.3</td>
<td>42.7</td>
<td>2296</td>
<td>1293.5</td>
<td>43.7</td>
</tr>
<tr>
<td>apex2</td>
<td>523.6</td>
<td>270.3</td>
<td>48.4</td>
<td>5145</td>
<td>2647.3</td>
<td>48.5</td>
</tr>
<tr>
<td>apex3</td>
<td>251.9</td>
<td>135.8</td>
<td>46.1</td>
<td>2413</td>
<td>1274.1</td>
<td>47.2</td>
</tr>
<tr>
<td>apex5</td>
<td>258.3</td>
<td>138.5</td>
<td>46.4</td>
<td>2446</td>
<td>1269</td>
<td>48.1</td>
</tr>
<tr>
<td>Avg.</td>
<td>177.3</td>
<td>102.1</td>
<td>27.3</td>
<td>1704</td>
<td>963.3</td>
<td>28.2</td>
</tr>
</tbody>
</table>

5.2. Results for 350nm Technology

Considering 350nm technology, the sized circuits using our gate sizing tool showed a reduction, on average, of 4.5% in delay, and area and power values are similar to the values presented by the circuits mapped to standard-cells library, as showed at Tab. 3.

Tab. 4 shows the values obtained for gate sizing minimizing area subject to delay found by minimizing delay, Tab. 3. Tab. 4 presents area and power values by delay minimization (Min. delay) and area minimization (Min. area) and their reduction (R) values. Area optimization allowed a reduction, on average, of 29.9% in area and 28.5% in power consumption.
6. Conclusion

Gate sizing using GP achieves better results compared with a circuit using commercial standard cell sizes selected by RTL Compiler.

In this paper, we solve a gate sizing problem using GP combining the works [3], [2]. To model the gates, we use a Switch-Level RC Gate Model. In our tool, the gate sizing problem can be formulated in two ways:
1) Aiming to minimize the circuit delay subject to a maximum area, or
2) Minimizing area subject to a delay restriction.

We showed that, for our test cases in 45nm and 350nm, the gate sizing tool produced better results compared with a circuit using commercial standard cell sizes selected by RTL Compiler. The tests show that our gate sizing tool using GP reduced the delay in 21%, on average, for 45nm and 4.5% for 350nm, considering delay minimization. For area minimization, considering 45nm technology, our gate sizing reduced the area in 28.2%, on average, and 27.3% in power consumption. For 350nm, the reduction in area was 29.9% and 28.5% in power consumption, on average.

Using an automatic cell generation tool, as ASTRAN [12], we can generate cells in the desired size and take advantage of the better results in timing, area and power, which is critical in recent technologies.

7. Acknowledgment

This work is partially supported by Brazilian National Council for Scientific and Technological Development (CNPq–Brazil) and Coordination for the Improvement of Higher Education Personnel (CAPES).

8. References


