MINIMIZING GATE CAPACITANCES WITH TRANSISTOR SIZING

Artur Wróblewski, Otto Schumacher, Christian V. Schimpfle and Josef A. Nossek

Munich University of Technology
Arcisstr. 21, 80333 Munich, Germany
e–mail: arwr@nws.ei.tum.de

ABSTRACT

In this paper a method for choosing appropriate transistor topology for use with transistor sizing is presented. In combinatorial blocks of static CMOS circuits transistor sizing can be applied for delay balancing in order to guarantee synchronously arriving signal slopes at the input of logic gates. Since the delay of a logic gate depends directly on transistor sizes, the variation of channel-widths and -lengths (W and L) allows to equalize different path delays without influencing the total propagation delay of the circuit. Thus, glitching can be avoided. To achieve optimal results, transistor lengths have to be increased, which results in both increased gate capacitances and area. Splitting the long transistors counteracts this negative influence and reduces the power dissipated. A program **GliMATS** for automated circuit optimization has been implemented. Experimental results show that significant power savings can be achieved with this method.

1. INTRODUCTION

Many approaches to transistor sizing have been presented in the past. A large number of them aim at area and power optimization under given delay constraints [1, 2, 4]. Since the substantial progress in development of deep submicron techniques, power dissipation has become the main limiting factor. This problem has been addressed in [6]. Unlike for most methods that focus on maximizing the speed of a circuit by variation of transistor widths, this method allows also the transisitor lengths to be variable. Reducing speed for delay balancing is only allowed for parts of the circuit that are not in the critical path. In [5] a method is presented, where all transistor widths outside the critical path are reduced in order to reduce the total capacitance of the circuit. However, delay balancing may not be possible if only the widths are variable because the limit here is the minimum feature size. Further speed reduction can then be achieved by increasing the transistor length [6]. In order to keep track of the conflicting design objectives like increasing transistor sizes for delay balancing, and at the same time reducing the total power consumption caused by charging capacitances, the method is formulated as a multiobjective optimization problem. Here we present further improvements to the method described in [6]. They include changes to the topology of the circuit where possible and aim at the reduction of gate capacitances. This makes further decrease in power dissipation possible. In the following we consider circuits in which increasing transisitor lengths is necessary. Decreasing of W to make the gate slower, which is the usual approach, results in smaller area and less power consumption. On the contrary increasing L provides slower gates, but influences both, the area and power dissipation negatively. Thus, increasing L represents the worst case approach to transistor sizing. Therefore, the power savings presented here reflect only the benefits of a delay balanced circuit due to reduced glitch activity. Of course **GliMATS** is not limited by this artificial constraint.

2. DELAY AND POWER MODELS

This section gives a short overview on delay and power modelling used. More detailed information can be found in [6]. The models used for the transistor sizing method presented here are defined at gate level. When modeling a circuit at gate level (*macromodeling*), the relatively large number of local parameters that describe every single transistor is reduced to a set of scale factors for each gate. This enables acceptable computation time for optimization of larger circuits. In the considered case the number of variables is reduced to one specific W and one specific L for each gate. If W and/or L are varied, all transistor widths and/or lengths within the gate are scaled by the same factor simultaneously.

2.1. Delay Model

The delay of a gate at position m can be split up into two parts [3, 4]: The step response delay $\tau_{s,m}$, which is independent of the input signal form, and $\tau_{in,m}$, which is the contribution caused by the finite input signal rise and fall

0-7803-6685-9/01/$10.00©2001 IEEE

times. The total delay τ_m is then approximated by

$$\tau_m = \tau_{in,m} + \tau_{s,m}. \qquad (1)$$

The goal of the optimization is the minimization of the number of glitches, which necessitates equalizing all path delays. However, the step response delay $\tau_{s,m}$ depends on the input transition. Therefore, the different paths can exactly be balanced for one specific transition only. Experiments have shown that the worst case delay is a good choice and is easy to formulate in the model. Furthermore, numerous simulations based on this model show, that even though the paths cannot be exactly balanced for all transitions, glitching can be eliminated in most cases.

According to the Elmore Delay Model used here, the delay of the gate considered can be described as:

$$\tau_m = f(W_{m-1}, L_{m-1}, W_m, L_m, W_{m+1}, L_{m+1}). \qquad (2)$$

The total delay of a path ν is the sum over all gate delays in this path:

$$\tau_\nu = \sum_{m=1}^{n} \tau_m, \qquad (3)$$

where n is the number of gates in the path.

2.2. Power Consumption Model

With the objective function (3) only the delay can be considered in the optimization procedure so far. In order to take account of the transistor size dependency of the short-circuit currents and the total capacitance of a circuit, an objective function for power consumed by gate m can be formulated as follows:

$$P_m = P_{m,cap} + P_{m,sc}, \qquad (4)$$

where $P_{m,cap}$ denotes the power consumed for charging the gate and drain/source capacitances and $P_{m,sc}$ denotes the short-circuit power consumption of gate m. Similiar to the delay, power can be described as:

$$P_m = f(W_{m-1}, L_{m-1}, W_m, L_m, W_{m+1}, L_{m+1}). \qquad (5)$$

The total power consumption in path ν can be formulated as:

$$P_\nu = \sum_{m=1}^{n} P_m, \qquad (6)$$

for a path with n gates.

3. MULTIOBJECTIVE OPTIMIZATION

In order to find a power optimal solution for W and L the designer is confronted with two conflicting design criteria: path balancing by transistor sizing, achieved by enlarging transistors, and low power consumption during charging capacitances which requires small transistors at the same time. In order to equalize all the path delays with respect to the critical path, every path requires individual optimization. Let τ_{crit} denote the critical path delay of the circuit. For every path ν

$$\min_{W,L} |\tau_\nu - \tau_{crit}| \qquad (7)$$

must be calculated to achieve path balancing. The path delay τ_ν is defined by (3). The power consumption according to (5) is minimized by

$$\min_{W,L}(P_\nu = \sum_{m=1}^{n} P_m). \qquad (8)$$

Equations (6) and (7) describe convex optimization problems in W and L. The multiobjective optimization problem is given by:

$$\min_{W,L}(S_\nu = w \cdot (\tau_\nu - \tau_{crit})^2 + (1-w) \cdot P_\nu). \qquad (9)$$

The weight factor w varies between 0 and 1, $w \in [0, 1]$. Results of the optimization are highly independent of the choice of w. Only values extremly close to 0 or 1 influence the result. In order to have a cost function, which is differentiable everywhere, $|\tau_\nu - \tau_{crit}|$ is replaced by its square. The upper and lower bounds of the transistor sizes are determined by the minimum feature size of the used technology and the user defined limits for the maximum available area for a single transistor. These additional constraints have to be considered separately. Assigning a value to w allows a solution to be chosen depending on which of the design objectives is more desired: low power consumption caused by the total capacitive load or balanced path delays. However, experiments have shown that for many circuits the best low power solution is obtained if $|\tau - \tau_{crit}| = 0$, i.e. for optimally balanced paths. This is usually given when $w = 0.5...1$.

4. MINIMIZING GATE CAPACITANCES

As mentioned before, the case considered in this paper is the one, when transistors are being made longer. This leads to larger channel resistance of the transistor and increases its gate capacitance. In the following we present two alternative ways of reducing this negative influence.

4.1. "Twin-Transistors"

So far the channel resistance as well as the gate capacitance

$$R_{ch} \sim r_{ch}\frac{L}{W}, \quad C_G \sim c_G LW \qquad (10)$$

are propportional to the channel length. On the other hand the delay is proportional to both, the channel resistance and gate capacitance. To increase the channel resistance without increasing the gate capacitance one has to be able to change them independently from each other. This is possible if the capacitance and the resistance are no longer part of one common transistor. To achieve that one can split the common transistor into two. The resistance can then be assigned to one of them, the capacitance to the other. The goal is to make the capacitance as small as possible. The reasonable approach is to make its transisitor minimum feature sized. This one will be responsible for switching. The length of the other transistor has to be dimensioned in a manner that satisfies the delay constraint given. The gate capacitance of this transistor has, of course, been increased, but it's not of importance anymore since its gate can be hard wired to the voltage supply. Thus, it has no influence on dynamic power dissipation. By splitting the transistor into two, both goals have been achieved. Despite increased resistance, the gate capacitance can be held minimal. The topology of "Twin-Transistors" is shown in Fig.1.

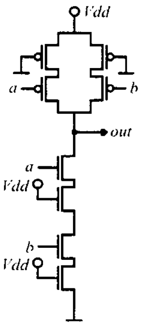

Figure 1: Topology of "Twin-Transistors"

4.2. "Merged-Transistors"

Introducing "Twin-Transistors" doubles the number of devices in the gate. Even if they can be placed in a area-saving way, together with additional wiring, the area taken is almost doubled. It's obvious that, within one block, the transistors, responsible for the increased delay, can be merged together. This considerably influences the data dependency of the gate delay. The range in which the delay varies becomes smaller and moves towards worst-case-delay. This is advantageous for the purpose of optimization. The topology of "Merged-Transistors" is shown in Fig.2.

The changes made to gate topology have their influence on power savings and area increase. Numerous simualtions have shown, that in combinatorial blocks of static CMOS circuits 50% to 90% of power is being dissipated due to glitch activity. For further considerations we will assume a mean value of 70%. In the following we try to estimate, how much power could be saved if glitching was elimianated completely. Let us consider "Twin-Transistors" first. For all additional transistors, that are not connected to power supply, additional drain/source capacitances of about 25% have to be taken into account. This results in increased power disspation by 12-17%. Thus power that can be saved drops to 65%. With "Merged-Transistors" there are no additional drain or source capacitances. A gate that has been modified in this manner does not dissipate more switching power than an usual one. In theory all 70% could be saved.

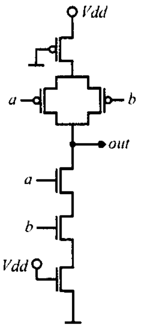

Figure 2: Topology of "Merged-Transistors"

The area increase is significant. A minimum size "Twin-Transistor" itself needs about 66% more area than a usual one. This number increases with the transistor length. Resulting average area increase is about 77%. Additional wiring could require even more space. For "Merged-Transistors" the number of additional transistors is significantly lower. But the ones used could be very long. The wiring is much less costly than in the case of "Twin-Transistors" and is comparable to that of a standard gate.

5. APPLICATIONS AND EXPERIMENTAL RESULTS

The proposed path balancing method has been tested on some example circuits, a few selected are shown here. They included array multipliers and a few combinational logic blocks (ISCAS'85 Benchmarks). They have been simulated with PowerMill before and after transistor optimization for glitch reduction. The different topologies have been tested in the optimization. For simulation 10000 random input vectors have been applied to each circuit. The results are summarized in Tables 1, 2 and 3.

Note that the percentage of power reduction due to the glitch elimination increases for larger arrays because of the snowball effect that glitches stimulate in these circuits. The CPU-time for the complete optimization of a 16 × 16 multiplier is about 7 minutes on an Ultra Sparc 10 workstation.

The results show significant power savings after **Gli-**

Circuit	not balanced	standard topology	power savings
4 × 4 Mult.	0.157	0.087	44%
8 × 8 Mult.	0.822	0.530	33%
16 × 16 Mult.	4.000	2.432	39%
c17	0.026	0.023	12%
c432	0.427	0.365	14%
c499	0.997	0.937	6%
c880	0.770	0.567	22%
c1908	0.935	0.837	10%

Table 1: Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for usual topology ($0.25\mu m, V_{dd} = 2.5V$, PowerMill simulations with 10000 random input vectors).

Circuit	not balanced	"Twin" topology	power savings
4 × 4 Mult.	0.157	0.092	41%
8 × 8 Mult.	0.822	0.412	49%
16 × 16 Mult.	4.000	2.000	50%
c17	0.026	0.018	30%
c432	0.427	0.260	39%
c499	0.997	0.695	30%
c880	0.770	0.417	45%
c1908	0.935	0.570	39%

Table 2: Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for "Twin" topology ($0.25\mu m, V_{dd} = 2.5V$, PowerMill simulations with 10000 random input vectors).

MATS has been applied. However, one must be aware that enlarging of the transistor lengths to increase the delay results in slower signal slopes which may lead to larger short circuit power consumption (this is considered in the results presented)

6. CONCLUSION

In this work two methods for improving the performance of the transisitor sizing algorithm presented in [6] have been introduced. By splitting long transistors into two a decrease in gate capacitances has been achieved. In case of "Twin"-topology a significant area increase has to be taken into account. The new version of the optimization software **GliMATS** is capable of handling all three topologies. It automatically reads the netlist of a circuit, builds the delay and power functions and starts multiobjective optimization. Depending on the chosen mode **GliMATS** can automatically introduce different topologies, where applicable, to achieve best power savings. The netlist of the optimized, delay balanced circuit with the new values of W and L for each gate is returned by the program. By applying this method glitching in a circuit can be reduced drastically. Experimental results show significant power savings after optimization.

Circuit	not balanced	"Merged" topology	power savings
4 × 4 Mult.	0.157	0.087	44%
8 × 8 Mult.	0.822	0.382	53%
16 × 16 Mult.	4.000	1.850	53%
c17	0.026	0.018	30%
c432	0.427	0.257	39%
c499	0.997	0.695	30%
c880	0.770	0.425	44%
c1908	0.935	0.567	39%

Table 3: Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for "Merged" topology ($0.25\mu m, V_{dd} = 2.5V$, PowerMill simulations with 10000 random input vectors).

7. REFERENCES

[1] M. Borah, R. M. Owens, and M. J. Irwin. Transistor Sizing for Low Power CMOS Circuits. *IEEE Trans. on Computer-Aided Design*, 15(6):665–671, June 1996.

[2] J. P. Fishburn and A. E. Dunlop. TILOS: A Posynomial Programming Approach to Transistor Sizing. *Proc. ICCAD*, pages 326–328, 1985.

[3] N. Hedenstierna and K. O. Jeppson. CMOS Circuit Speed and Buffer Optimization. *IEEE Transactions on Computer Aided Design*, CAD-6(2):270–281, March 1987.

[4] B. Hoppe, G. Neuendorf, D. Schmitt-Landsiedel, and W. Specks. Optimization of High-Speed CMOS Logic Circuits with Analytical Models for Signal Delay, Chip Area, and Dynamic Power Dissipation. *IEEE Trans. on Computer-Aided Design*, 9(3):236–247, March 1990.

[5] S. Trimberger. Automated Performance Optimization of Custom Integrated Circuits. *Proc. Int. Symp. on Circuits and Systems*, pages 194–197, 1983.

[6] A. Wróblewski, C.V. Schimpfle, and J. A. Nossek. Automated Transistor Sizing Algorithm For Minimizing Spurious Switching Activities in CMOS Circuits. *Proc. IEEE Int. Symp. on Circuits and Systems, ISCAS'2000, Geneva*, May 2000.