

Minimizing Spurious Switching Activities with Transistor Sizing*

ARTUR WRÓBLEWSKI^{a,†}, CHRISTIAN V. SCHIMPFLE^b, OTTO SCHUMACHER^c and JOSEF A. NOSSEK^a

^aMunich University of Technology, Arcisstr. 21, 80333 München, Germany; ^bTexas Instruments Deutschland, 85350 Freising, Germany; ^cInfineon Technologies AG, P.O. Box 800949, 81609 München, Germany

(Received 27 April 2001; Revised 11 June 2001)

In combinatorial blocks of static CMOS circuits transistor sizing can be applied for delay balancing as to guarantee synchronously arriving signal slopes at the input of logic gates, thereby avoiding glitches. Since the delay of logic gates depends directly on transistor sizes, their variation allows equalizing different path delays without influencing the total delay of the circuit. Unfortunately, not only the delay, but also power consumption circuits depend on the transistor sizes. 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.

Keywords: Glitches; Transistor sizing; VLSI; Low-power; CMOS

INTRODUCTION

Optimal sizing of MOS-transistors is a widely investigated method for the design of CMOS-circuits with restricted area, propagation delay or power consumption. A large number of the previously published approaches aim at area and power optimization under given delay constraints [1–3]. Optimal area utilization is still important, but since the substantial progress in development of deep submicron techniques, power dissipation has become the main limiting factor. The power consumption models often include only the power consumed for charging transistor gate- and drain/source-capacitances. The power models in Ref. [1] include also the dissipation caused by short-circuit currents that occur during transition when both P- and N-transistors of a CMOS-stage are conducting. Delay balancing for reducing the glitching activities in combinational Wallace-trees and array multipliers has been introduced in Ref. [4].

Unlike for most approaches that focus on maximizing the speed of a circuit by variation of transistor widths, the method presented here also allows the transistor 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 Ref. [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. 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. Furthermore, by introduction of changes to the topology of the circuit where possible, the reduction of gate capacitances can be achieved. This makes further decrease in power dissipation possible.

In the following we consider circuits in which increasing transistor 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 **GiMATS**, a program that has been implemented for automated circuit optimization, is not limited by this artificial constraint.

*Portions reprinted, with permission, from “Automated Transistor Sizing Algorithm For Minimizing Spurious Switching Activities in CMOS Circuits” ISCAS’00 ©2000 IEEE and “Minimizing Gate Capacitances with Transistor Sizing” ISCAS’01 ©2001 IEEE.

[†]Corresponding author. E-mail: arwr@nws.e-technik.tu-muenchen.de

DELAY AND POWER MODELS

The delay and power models used for the transistor sizing method presented here are defined at gate level. Although transistor level models may offer more degrees of freedom and allow individual sizing of each single transistor, it turns out to be more desirable to have a low dimensional optimization problem in order to be able to optimize larger circuits within acceptable computation time. 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. 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.

Delay Model

The macromodel delay has to be described for each type of logic gate separately. In the following, the delay model is derived exemplarily for a 2-input CMOS AND gate. The generalization to any other logic gate type is straightforward. The delay of a gate at position m can be split up into two parts [3,6]: The step response delay $\tau_{s,m}$, which is independent from the input signal form, and $\tau_{in,m}$, which is the contribution caused by the finite input signal rise and fall times. The total delay τ_m is then approximated by

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

The optimization method aims at the minimization of glitches, which necessitates equalizing all path delays. However, the step response delay $\tau_{s,m}$ depends on the input transition. For example, the step response delay of a 2-input AND gate in 0.25 μm technology with a certain load is 0.4 ns for the input transition 11 \rightarrow 00 and 0.75 ns for 00 \rightarrow 11. Therefore, the different paths can exactly be balanced for one special 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 although the paths cannot be exactly balanced for all transitions, glitching could be eliminated to a large amount. The step response delay $\tau_{s,m}$ is described with Elmore's delay model. It is assumed that there is a fixed ratio ρ between N- and PMOS transistor sizes

$$\left(\left(\frac{W_m}{L_m} \right)_p = \rho \left(\frac{W_m}{L_m} \right)_n \right) \quad (2)$$

to achieve $R_{n,m} = R_{p,m} = R_m$ for the channel resistances of N- and PMOS transistors. The drain/source and gate capacitances can then be written as

$$C_{d/s_n,m} = C_{d/s,m}, \quad C_{d/s_p,m} = \rho C_{d/s,m} \quad (3)$$

and

$$C_{g_n,m} = C_{g,m}, \quad C_{g_p,m} = \rho C_{g,m}. \quad (4)$$

All components can be formulated as functions of the channel length and width:

$$R_m = r \frac{L_m}{W_m} \quad (5)$$

$$C_{d/s,m} = c_{d/s} W_m \quad (6)$$

and

$$C_{g,m} = c_g W_m L_m \quad (7)$$

where r , $c_{d/s}$ and c_g denote process dependent constants. The load capacitance depends on the transistor sizes at position $m+1$ and the number of input transistor gates κ_{m+1} at position $m+1$, in such a way that

$$C_{L,m} = \kappa_{m+1} c_g W_{m+1} L_{m+1}. \quad (8)$$

For the 2-input AND gate follows

$$\begin{aligned} \tau_{s,m} = & rc_{d/s}(5 + 3\rho)L_m + 2rc_g(1 + \rho)L_m^2 \\ & + rc_g(1 + \rho)\frac{L_m}{W_m}W_{m+1}L_{m+1} \end{aligned} \quad (9)$$

corresponding to Fig. 1.

According to Ref. [3] the input dependent delay τ_{in} is given by:

$$\tau_{in} = \frac{1}{6} \tau_{r/f,m-1} \left(1 + \frac{2V_t}{V_{dd}} \right) \quad (10)$$

where $\tau_{r/f,m-1}$ is the rise/fall time of the input signal, V_t is the threshold voltage and V_{dd} the power supply voltage. If a chain of gates is considered, $\tau_{r/f,m-1}$ is the rise/fall time of the output signal of the previous gate at position $m-1$. In good approximation these signal ramps can be

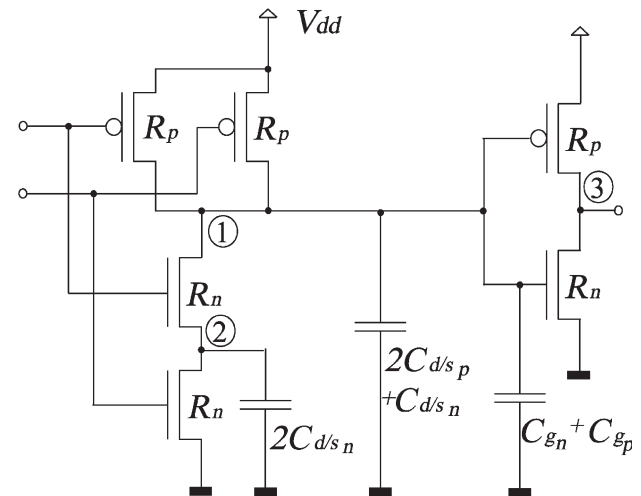


FIGURE 1 AND gate with parasitics for delay modeling.

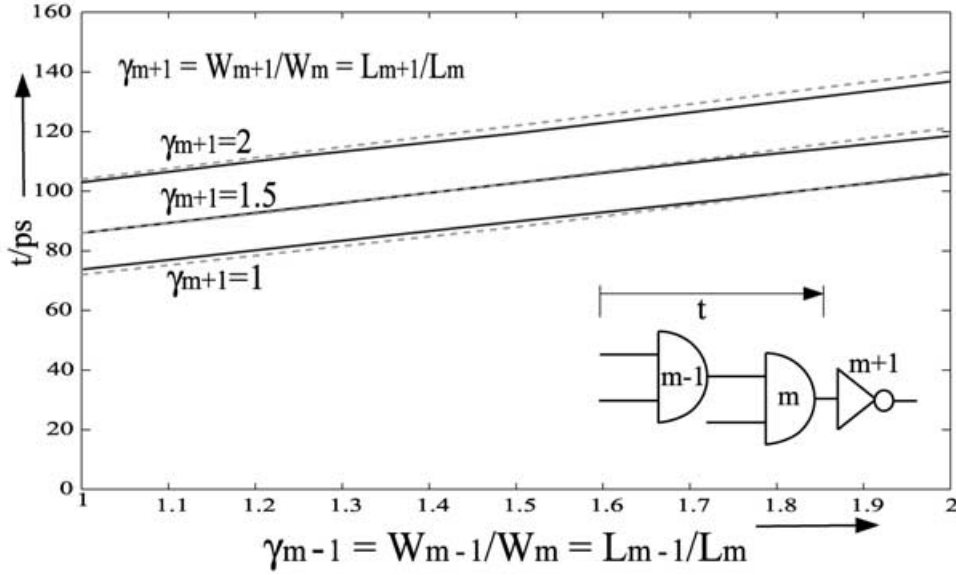


FIGURE 2 SPICE-simulated (dotted line) and calculated (solid line) delays t for different ratios $\gamma_{m-1} = W_{m-1}/W_m = L_{m-1}/L_m$ and $\gamma_{m+1} = W_{m+1}/W_m = L_{m+1}/L_m$.

considered to be quasi-linear, such that $\tau_{r/f,m-1}$ in Eq. (10) can be replaced by its linear approximation $\tau_{eff,m-1}$ which is given by:

$$\tau_{eff,m-1} = \frac{8}{3 \ln 3} \tau_{s,m-1} \left(1 - 0.27 \frac{V_t}{V_{dd}} \right). \quad (11)$$

For details see Ref. [3]. The input dependent delay can then be written as:

$$\begin{aligned} \tau_{m,in} &= \frac{4}{9 \ln 3} \left(1 + 1.73 \frac{V_t}{V_{dd}} - 0.54 \left(\frac{V_t}{V_{dd}} \right)^2 \right) \cdot \tau_{s,m-1} \\ &= K \cdot \tau_{s,m-1} \end{aligned} \quad (12)$$

where all technology dependent parameters are merged in the constant K . With Eqs. (1), (9) and (12) the total gate delay is given by:

$$\begin{aligned} \tau_m &= K \cdot (rc_{d/s}(5 + 3\rho)L_{m-1} + 2rc_g(1 + \rho)L_{m-1}^2 \\ &\quad + rc_g(1 + \rho) \frac{L_{m-1}}{W_{m-1}} W_m L_m) + rc_{d/s}(5 \\ &\quad + 3\rho)L_m + 2rc_g(1 + \rho)L_m^2 + rc_g(1 + \rho) \\ &\quad \times \frac{L_m}{W_m} W_{m+1} L_{m+1} \\ &= k_1 L_{m-1} + k_2 L_{m-1}^2 + k_3 \frac{L_{m-1}}{W_{m-1}} L_m W_m + k_4 L_m \\ &\quad + k_5 L_m^2 + k_6 \frac{L_m}{W_m} L_{m+1} W_{m+1}. \end{aligned} \quad (13)$$

All technology dependent parameters are merged in constants $k_{1..6}$. The total delay of a path number ν is

the sum over all gate delays in this path:

$$\tau_\nu = \sum_{m=1}^n \tau_m, \quad (14)$$

where n is the number of gates in the path.

For verification of the delay model the delay through a chain of two AND gates loaded with an inverter is considered. The circuit is shown in the lower right corner of Fig. 2.

The diagram in Fig. 2 shows the SPICE-simulated (dotted line) and calculated (solid line) delays from the primary inputs to the output of gate m for different loads (corresponding to W_{m+1} and L_{m+1}) and different W_{m-1} and L_{m-1} (W_m and L_m is held constant). The calculation with the proposed model shows good correspondence with the simulations.

Power Consumption Model

With the objective function [Eq. (14)] only the delay can be considered in an optimization procedure so far. In order to take into account also the transistor size dependency of the short-circuit currents and the total capacitance of a circuit, an objective function for the power consumption of gate number m can be formulated as follows:

$$P_m = P_{m,cap} + P_{m,sc}, \quad (15)$$

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 . For a 2-input AND, $P_{m,cap}$ can be written as:

$$P_{m,cap} = c_1 L_m + c_2 W_m L_m + c_3 W_{m+1} L_{m+1}, \quad (16)$$

where

$$c_1 = fV_{dd}^2(\alpha_1 c_{d/s}(1 + 2\rho) + 2\alpha_2 c_{d/s}), \quad (17)$$

$$c_2 = fV_{dd}^2 \alpha_1 c_g(1 + \rho) \quad (18)$$

and

$$c_3 = \alpha_3 fV_{dd}^2 (c_{d/s}(1 + \rho) + c_g k_{m+1}(1 + \rho)). \quad (19)$$

Factors α_1 , α_2 , and α_3 , denote the switching activities at nodes 1, 2, and 3, respectively (see the numbers in circles in Fig. 1), which are considered as constants over W and L .

The short-circuit power dissipation of a CMOS inverter according to Ref. [7] is given by:

$$P_{m,sc} = \tau_{eff,m-1} \frac{\beta_m}{12} f(V_{dd} - 2V_t)^3. \quad (20)$$

β_m is the transistor gain factor ($\beta_m \sim W_m/L_m$) and $\tau_{eff,m-1}$ is given in Eq. (11). For a 0.25 μm technology and $V_{dd} = 2.5\text{ V}$

$$P_{m,sc} \approx 0.00045f \frac{W_m}{L_m} \tau_{s,m-1}. \quad (21)$$

It can be shown experimentally, that neglecting the contribution of $P_{m,sc}$ has no negative influence on the results of the path balancing method even for complex gates. Therefore, it is reasonable to set $P_m = P_{m,cap}$. The total transistor size dependent power consumption in path number ν can be formulated as:

$$P_\nu = \sum_{m=1}^n P_m, \quad (22)$$

for a path with n gates.

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 for charging capacitances which requires small transistors at the same time. This problem usually can be solved with a non-linear programming method. A common approach to find a solution is to keep one of the design criteria within upper and lower bounds and find an optimal solution for the other one under these restrictions. The problem is to determine the upper and lower bounds if they are not previously known. Awkwardly chosen bounds may result in an unsolvable optimization problem. Therefore, not every single criterion is optimized while restricting all the others, but the weighted sum of all the design criteria [3]. 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 a solution of

$$\min_{W,L} |\tau_\nu - \tau_{crit}| \quad (23)$$

must be calculated to achieve path balancing. The path delay τ_ν is defined in Eq. (14). The power consumption according to Eq. (22) is minimized by

$$\min_{W,L} (P_\nu = \sum_{m=1}^n P_m). \quad (24)$$

Equations (23) and (24) describe convex optimization problems in W and L . The multiobjective optimization problem is given by:

$$\min_{W,L} (S_\nu = w \cdot (\tau_\nu - \tau_{crit})^2 + (1 - w) \cdot P_\nu). \quad (25)$$

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 extremely close to 0 or 1 influence the result. In order to obtain 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.

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 $w = 0.5 \dots 1$.

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.

“Twin-transistors”

So far the channel resistance as well as the gate capacitance

$$R_{ch} \sim r_{ch} \frac{L}{W}, \quad C_G \sim c_G L W \quad (26)$$

are proportional 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

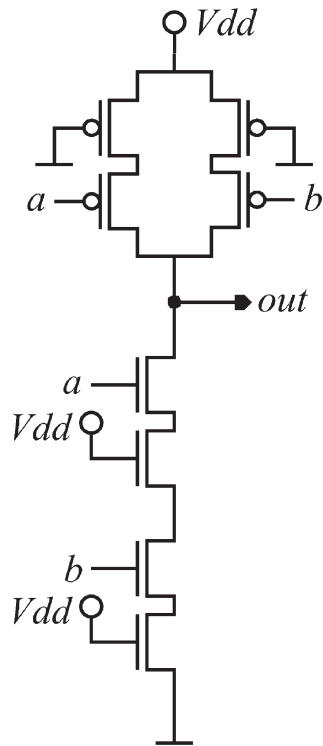


FIGURE 3 Topology of “Twin-Transistors”.

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 transistor 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.3.

“Merged-transistors”

Introducing “Twin-Transistors” doubles the number of devices in the gate. Even if they can be placed in an 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

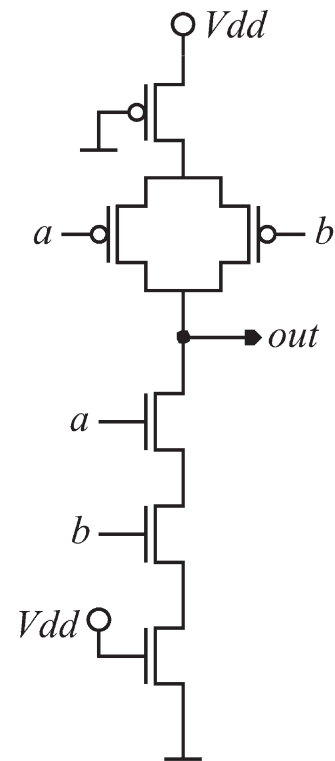


FIGURE 4 Topology of “Merged-Transistors”.

optimization. The topology of “Merged-Transistors” is shown in Fig. 4.

The changes made to gate topology have their influence on power savings and area increase. Numerous simulations have shown, that in combinatorial blocks of static CMOS circuits 50–90% 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 eliminated completely. Let us consider “Twin-Transistors” first. For all additional transistors that were not connected to power supply, additional drain/source capacitances of about 25% have to be taken into account. This results in increased power dissipation 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 a usual one. In theory all 70% could be saved.

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.

BALANCING OF THE PATH DELAYS

In order to reduce the glitches in a combinatorial circuit, it must be guaranteed that the signal slopes at a gates inputs arrive synchronously. This can be achieved by path balancing, i.e. by slowing down fast paths outside the critical path such that the total delay of the circuit is not affected.

In general the delay of a gate depends on the input transition. Therefore, exact path balancing at gate level is only possible for one special case of transition. As mentioned in “Delay model”, the delay which shall be synchronized with the method presented here is the worst case delay. For all the other transitions that cause different delays the paths will then be only approximately balanced. Circuit optimization for maximum speed under given constraints can be included but is not considered here. It is assumed that the circuit is already optimized to match certain delay constraints. As far as possible, reducing the widths of the transistors in the path, thereby, also reducing the total gate capacitance, is decreasing the path delay. Of course, this is limited by the minimum feature size of the technological IC-process. Further increase in delay is possible by increasing transistor lengths.

To explain the path balancing algorithm, we consider an example circuit pictured in Fig. 5. The gates of the circuit are numbered from 1 to 8 and their characteristic delays are also indicated. The black boxes are input and output registers. In the first step all gates in the critical path (gates 1–4) are marked as “fixed”, which means that their transistor sizes remain unchanged during optimization process. The delay of the critical path is denoted as $\tau_{crit} = \tau_{1-2-3-4}$. Then path 5-3-4 is chosen and the transistor sizes are optimized until $\tau_{5-3-4} = \tau_{crit}$. As the sizes of gates

3 and 4 are fixed, only sizes in gate 5 can be changed for this purpose. After this process it is guaranteed, that signals at the inputs of gate 3 arrive synchronously. Following that gate 5 is also marked as “fixed”. In the next step path 6-7-4 is equalized to the critical path, such that $\tau_{6-7-4} = \tau_{crit}$ and gates 6 and 7 are marked. Finally the same is done with path 8-7-4. Note that the total delay of the circuit, relevant for the clocking speed of the input and output registers remains the same, namely τ_{crit} . By treating gate sizes as fixed in paths that have already been balanced, the number of paths in the circuit that remain for optimization decreases rapidly, i.e. for a 3×3 array multiplier with 107 paths only 5 optimization steps are necessary.

The transistor sizing algorithm for the reduction of glitching activity is implemented in the program **GLiMATS** (**GLitch** **MIN**imization by **A**utomated **T**ransistor **S**izing). It allows optimizing the circuits automatically. **GLiMATS** processes a netlist of the circuit. The user can set a value for the weight factor w and specify which of the transistor topologies is to be used—standard, “Twin-Transistors” or “Merged-Transistors”. As the output **GLiMATS** produces netlist of the glitch minimized circuit. It is assumed that the circuit is already optimized to match eventually given timing constraints, so the critical path must not be manipulated in order to retain the required maximum delay. The input netlist to the path balancing program is given from this previous speed optimization.

GLiMATS starts at the primary inputs with building a node list which describes the dependencies of all nodes from their predecessors. The delay of every passed node and its predecessors is saved. After all output nodes are reached, the critical path delay is known. Then the algorithm starts at the output nodes and traces back

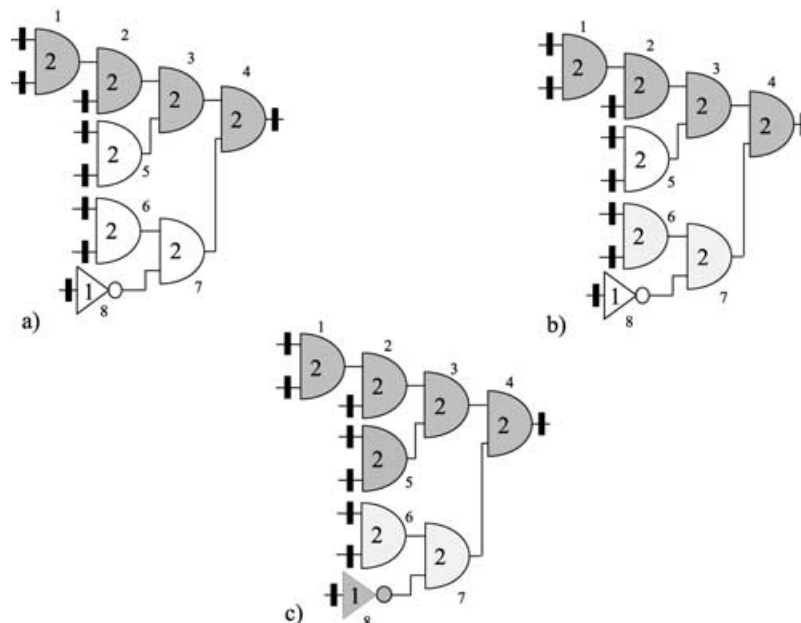


FIGURE 5 Optimization procedure.

TABLE I Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for usual topology ($0.25 \mu\text{m}$, $V_{dd} = 2.5 \text{ V}$, PowerMill simulations with 10,000 random input vectors)

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

to the primary inputs by choosing at each passed node the predecessor with the largest delay for the next node. The delay and power functions of this path are built according to Eqs. (14) and (22). The actual delay $\tau_{v,0}$ of the path is calculated. If $\tau_{v,0} \neq \tau_{crit}$, the path is optimized according to Eq. (25). For this purpose a MATLAB optimization routine is started. If after the optimization the length L_m of transistors in a gate is greater than a specified multiple of the minimal transistor length L_{min} given by technology constraints, then the topology of the transistor of the gate is changed or preserved according to user preferences. Of course, the given path has to be optimized again if the transistors have been modified. Once a path ν is optimized or if $\tau_{v,0} = \tau_{crit}$, all gates in this path are “marked”. In all further optimization steps for the yet non-balanced paths, the transistor sizes in “marked” gates are considered to be fixed and are not affected by the sizing algorithm. This complete procedure is repeated until all gates in the circuit are “marked”. Finally the netlist of the optimized, path delay balanced circuit is obtained. Note that it may not be possible to balance all the paths. For example, if one input of a gate is a primary input and another input of the same gate is the output of some other gate. Then the signal from the primary input cannot be further slowed down. Despite this case a circuit can be very well balanced as shown in “Applications and experimental results” section.

TABLE II Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for “Twin” topology ($0.25 \mu\text{m}$, $V_{dd} = 2.5 \text{ V}$, PowerMill simulations with 10,000 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 III Comparison of the power consumption in mW for circuits without and with path balancing by transistor sizing for “Merged” topology ($0.25 \mu\text{m}$, $V_{dd} = 2.5 \text{ V}$, PowerMill simulations with 10,000 random input vectors)

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

APPLICATIONS AND EXPERIMENTAL RESULTS

The proposed path balancing method has been tested on some example circuits, a few selected are shown here. They include array multipliers and combinational logic blocks (ISCAS’85 Bench-marks). The simulations have been performed with PowerMill before and after transistor optimization for glitch reduction. The different topologies have been tested in the optimization. For simulation 10,000 random input vectors have been applied to each circuit. The results are summarized in Tables I–III.

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 min on a Ultra Sparc 1 workstation.

Furthermore, for the 4×4 , 8×8 , and 16×16 array multipliers the increase of chip area has been estimated by measuring the area increase of a single cell due to transistor sizing and projecting this to the total chip area including the wiring. For the standard topology the expected area increase is between 15 and 31%. The additional silicon space needed is even greater for the “Twin-Topology”, but it’s significantly lower for the “Merged-Topology”.

To demonstrate the effect of path balancing by transistor sizing a logic circuit shown in Fig. 6 has been designed. If zero delay is assumed for all gates in this circuit, the output

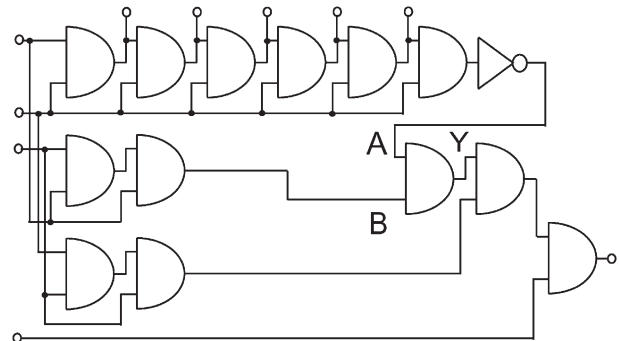


FIGURE 6 Logic circuit for demonstration of the path balancing method.

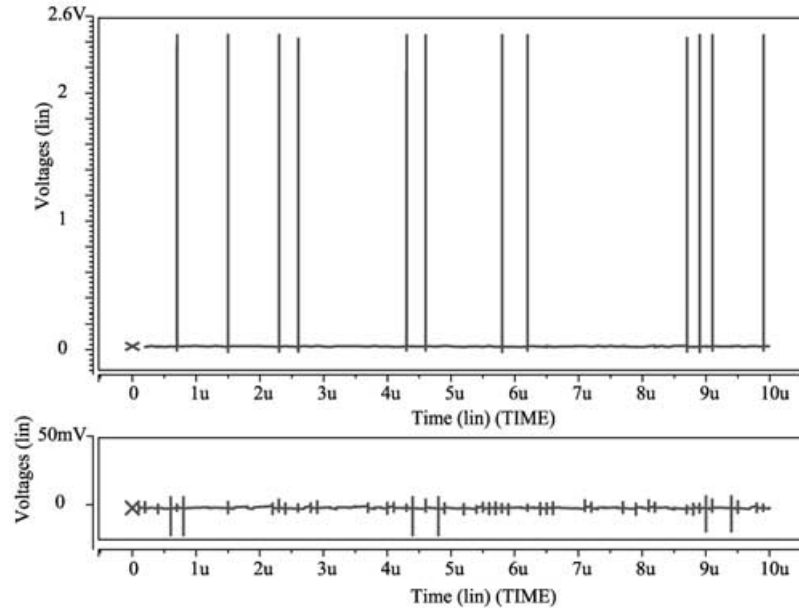


FIGURE 7 Signal *Y* of the circuit in Fig. 6 with unbalanced (top) and balanced path delays (bottom).

is always 0 regardless of the inputs. In a simulation with complex transistor models only glitches due to unbalanced paths are visible at the output.

The signal *Y* of the unbalanced circuit (all transistor sizes minimal) for random input signals is shown on top of Fig. 7.

The same input signals yield signal *Y* shown on the bottom of Fig. 7 for the circuit after optimization: glitches are completely eliminated. Note that the voltage axis are scaled differently. It results in power savings of 8%. Fig. 8 shows the slopes of the signals *A* and *B* before (top) and after (bottom) path balancing.

This figure demonstrates how signal *B* is delayed after transistor sizing in order to “wait” on signal *A*.

The results show significant power savings after **GiMATS** 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) and to an increase of the required chip area. A good application of the method would be in combination with pipelining, where register stages work as glitch barriers. In the combinational logic between two register stages glitching could be eliminated by the presented transistor optimization. In order to reduce design time for the different sized gates, module generators can be applied for automatic scaling of parameterized gate layouts.

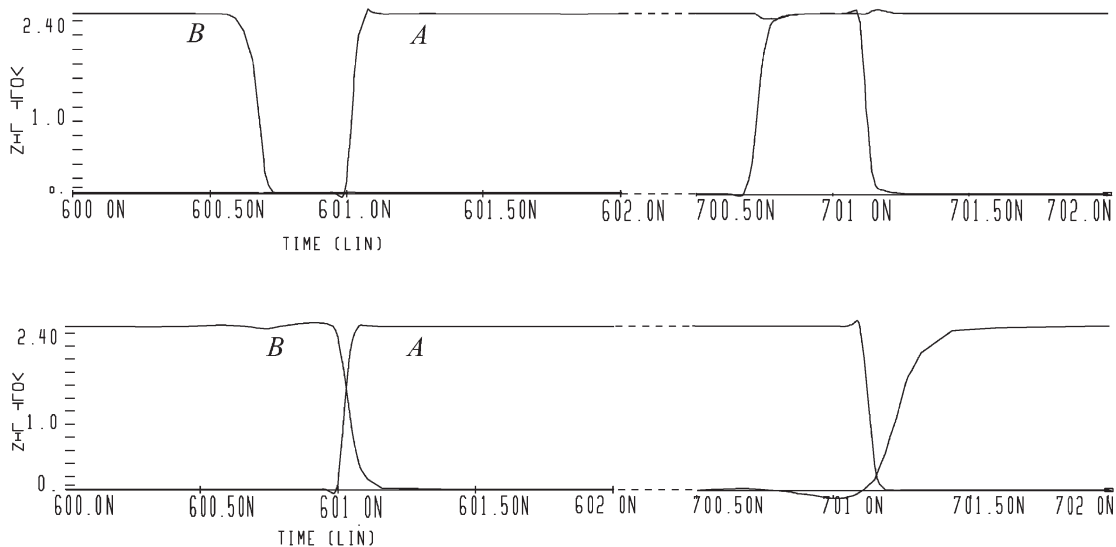


FIGURE 8 Example for signals *A* and *B* of the circuit in Fig. 6 with unbalanced (top) and balanced path delays (bottom).

CONCLUSION

In this work a method for transistor size optimization to achieve equal path delays in CMOS circuits has been presented. Delay and power consumption of a path can be modeled as functions of the transistor sizes W and L at gate level. With multiobjective optimization the path delay differences and the power consumed for charging capacitances can be minimized simultaneously. The solutions for W and L are restricted by upper and lower bounds, given by the minimum feature size and area limitations. 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 topology with “Merged”-transistors reduces the area increase and shows even better results in terms of power consumption. A tool—**GliMATS**—has been implemented that automatically reads a netlist of a circuit, builds the delay and power functions, starts multiobjective optimization and returns the netlist of the optimized, delay balanced circuit with the new values of W and L for each gate. **GliMATS** is capable of handling all three topologies. Depending on the chosen mode **GliMATS** can automatically introduce different topologies, where applicable, to achieve best power savings. By applying this method glitching in a circuit can be reduced drastically. Experimental results show significant power savings after optimization.

References

- [1] Borah, M., Owens, R.M. and Irwin, M.J. (1996) “Transistor sizing for low power CMOS circuits”, *IEEE Transactions on Computer-Aided Design* **15**(6), 665–671.
- [2] Fishburn, J.P. and Dunlop, A.E. (1985) “TILOS: A polynomial programming approach to transistor sizing”, *Proceedings of ICCAD*, 326–328.
- [3] Hoppe, B., Neuendorf, G., Schmitt-Landsiedel, D. and Specks, W. (1990) “Optimization of high-speed CMOS logic circuits with analytical models for signal delay, chip area, and dynamic power dissipation”, *IEEE Transactions on Computer-Aided Design* **9**(3), 236–247.
- [4] Sakuta, T., Lee, W. and Balsara, P.T. (1995) “Delay balanced multipliers for low power/low voltage DSP core”, *IEEE Symposium on Low Power Electronics* **October**, 36–37.
- [5] Trimberger, S. (1983) “Automated performance optimization of custom integrated circuits”, *Proceedings of International Symposium on Circuits and Systems*, 194–197.
- [6] Hedenstierna, N. and Jeppson, K.O. (1987) “CMOS circuit speed and buffer optimization”, *IEEE Transactions on Computer Aided Design* **CAD-6**(2), 270–281.
- [7] Veendrick, H.J.M. (1984) “Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits”, *IEEE Journal of Solid State Circuits* **SC-19**(4), 194–197.

Artur Wróblewski was born in Częstochowa, Poland in 1974. He studied electrical engineering at the Munich University of Technology where he received the Dipl.-Ing. degree in 1999. Currently he is working towards his Ph.D. at this university. His research interests include low power CMOS circuits and design of digital filters for mobile communication terminals.

Christian V. Schimpfle was born in Freising, Germany in 1968. He studied electrical engineering at the Munich University of Technology where he received the Dipl.-Ing. degree in 1995. From 1995 to 2000 he worked as a research assistant at the Institute for Circuit Theory and Signal Processing of the Munich University of Technology. In 2000 he received the Dr.-Ing. degree from the Munich University of Technology. He is currently working as a design engineer for analog integrated circuits at Texas Instruments, Freising, Germany. His research interests include methodologies for low power CMOS design and analog design.

Otto Schumacher has studied electrical engineering and information technology at Munich University of Technology where he received the Dipl.-Ing. degree in 2000. Currently, he is a design engineer for analogue and mixed signal circuits at Infineon Technologies AG, Munich. His fields of interest are low power integrated circuits.

Josef A. Nossek was born in 1947 in Vienna, Austria. He received the Dipl.-Ing. and Dr. degrees, both in electrical engineering, from the Technical University of Vienna, Austria, in 1974 and 1980, respectively. In 1974 he joined SIEMENS AG, Munich, Germany, where he was engaged in the design of passive and active filters for communication systems. Since 1989 he is Professor for Circuit Theory and Signal Processing at the Munich University of Technology, where he is conducting research in the areas of real-time signal processing and dedicated VLSI-architectures.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

