Optimization Potential of CMOS Power by Wire Spacing

Paul Zuber, Florian Helmut Müller et Walter Stechele
Institute for Integrated Systems, TU München
paul.zuber@tum.de

Abstract: In this work, we identify the power-optimal wire spacing as a *geometric program*. Its solution is a vector of individual distances between the wires. To quantify the optimization potential by this method we model the output of a grid based router with a set of parallel wires. A comparison of the power values before and after geometric optimization shows that the optimization potential lies well in the two digit percent zone for a representative circuit model in a 130nm process.

1 Introduction

Power consumption is one of the challenges in designing integrated circuits. Capacitive power still makes up the highest part of the power consumed in numerous integrated systems of today. The major fraction of the interconnect capacitances is increasingly caused by cross coupling wire segments within one metal layer as process geometries are scaling down. Therefore the distances between wire segments are good candidates for power optimization and build the focus of our paper.

Wire spacing, especially in combination with ordering, has been done in Electronic Design Automation for a long time. People attempt to space and order mainly bus wires for different objectives like power [MPS03] [ZR05], crosstalk [CKP01] [MPS03], area [Gro89], or timing [Moi04]. The latter work contains a more complete list.

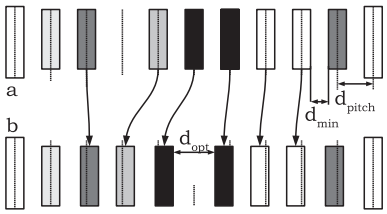

Figure 1: Illustration of bus wires (the darker the more active).

Our approach takes a step forward. The optimization is done by writing the wire spacing problem as a geometric program rather than developing a heuristic. Figure 1 shows an unoptimized bus (a) with an un-populated routing track. The idea is to place the wires off-grid (b), so that the unused space can be exploited: An individual distance is assigned to each wire pair based on their activities. (Cf. [ZWS05] for how to optimize any already detail-routed layout topologies.)

The paper continues with a brief illustration of CMOS power basics and the consequential formulation of power-optimal wire spacing as a geometric program. Section 3 quantifies the optimization potential. Section 4 will conclude the article.

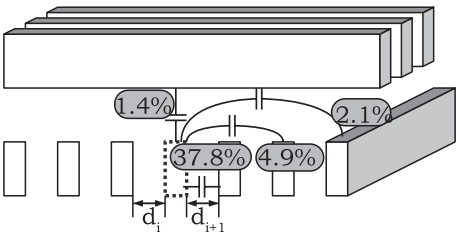

Figure 2: Capacitances in a 0.13μm process.

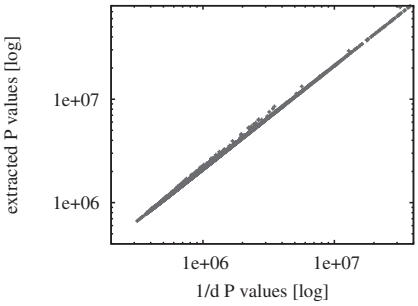

Figure 3: Correlation of power models.

2 Background

2.1 Power

The capacitive power consumption dissipated in the driving transistors of a circuit node i is the product of the node's capacitive load and a set of constant factors to be described shortly: $P_i = \kappa_i \cdot C_i$ with $\kappa_i := \alpha_{01,i} f_i V_{DD,i}^2$. The factor κ_i collects the variables 'toggle rate $\alpha_{01,i}$', 'frequency of clock domain of node i, f_i', and 'square of supply voltage of driving gate, $V_{DD,i}^2$'. Any of these factors has linear influence on power and can vary from node to node. However, we shall assume that $V_{DD,i}$ and f_i are constant from now on and therefore $\kappa \propto \alpha_{01}$.

2.2 Interconnect capacitance

It is widely understood that today the wire capacitance very often makes up the major part of a node's total capacitance in many ASIC circuits. Moreover, mainly the wire width is minimized as technology scales [SIA05][HMH01]. To avoid timing problems caused by too high resistances, but yet get acceptable integration densities, the wire thickness is kept relatively high. Backed by the decreasing pitch, this leads to an important implication for the interconnect capacitance: while in the past the highest fraction of the capacitance was caused by the coupling between different layers, now the capacitances within one layer dominate [WS02].

Figure 2 illustrates this situation for a wire (dotted) in a fully populated environment. The total self capacitance (100%) is broken down into capacitances to all other objects. The highest part (almost 2·38% for both sides) is the cross-coupling capacitance. We approximate this part with the plate capacitor formula: $C_i \propto \left(\frac{1}{d_i} + \frac{1}{d_{i+1}}\right)$. There is a strong correlation between the power values calculated with this expression and the power values calculated with extracted self capacitances, cf. Figure 3. So the applicability of this approximation seems to be rectified.

2.3 Power optimal wire spacing

Let us now consider N parallel wire segments of width w which must not get closer to each other than a technology dependent constant d_{min}. Let us further assume that we are given M possible routing tracks with pitch d_{pitch} on which a grid-based router placed the N wire segments. We can then formulate the power optimal wire spacing problem as a geometric program:

$$P_i \propto \sum_{n=1}^{N+1} \frac{(\kappa_{n-1} + \kappa_n)}{d_n} = \min! \quad (1)$$

$$d_n \geq d_{min} \qquad \forall n = 1 \ldots N+1 \quad (2)$$

$$\sum_{n=1}^{N+1} d_n \leq (M+1)d_{pitch} - N \cdot w \quad (3)$$

Cf. Figure 1 for an example with $N = 8$ and $M = 9$. At this point it should be noted that we model the whole scenario to be enclosed in between two static wires with the numbers 0 and $N + 1$ to avoid edge effects which could influence the results. The analogon on a chip could be power or shield wires. The right side of the area constraint (3) represents the available space. We generally assume $M > N$, or, should $d_{pitch} - w > d_{min}$, $M \geq N$, otherwise no freedom for optimization exists.

3 Example

3.1 Setup

To quantify the optimization potential we used FastCap [Nab05], a public domain capacitance extractor from MIT, a public domain geometric program optimizer [Lab00], and a typical 130 nm process. For a given N, we first chose a fixed set of switching activities, randomly generated by assuming a $1/x$ shaped distribution (similar to an actual microprocessor [Emb04]). We distributed the N wires to the M routing tracks and then had FastCap extract the wire capacitances. Providing the switching activities to the geometric optimizer (which used the simpler $1/d$ formula as capacitance model, as mentioned above) gave us an optimal wire spacing. Using this spacing and FastCap we obtained a second set of capacitances for the optimized case. It was shown that net ordering has influence on the optimization potential [ZR05]. Since this effect is not in the focus of this paper, it was eliminated by randomly permuting the distribution of wires to the routing tracks. From all those capacitances we could now calculate the power savings. Again, doing all the above a lot of times for different sets of switching activities gave us an average saving for given N and M. For FastCap we used N simple straight wires on layer metal2, spacing either derived from random allocation on M tracks or computed by the geometric optimizer.

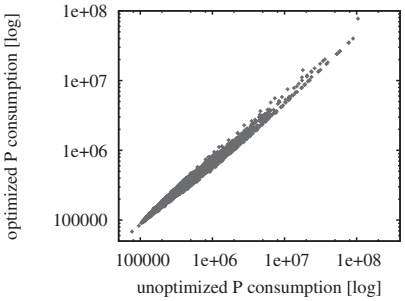

Figure 4: Optimized to unoptimized power values for N=16, M=20

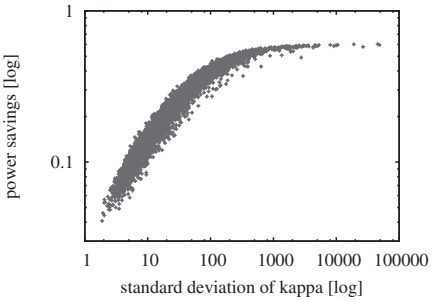

Figure 5: Power savings to standard deviation of toggle rates for N=64, M=65

3.2 Results

$N \backslash M$	N	N+1	1.25N
4	6.2	15.1	15.1
16	12.9	16.2	21.7
64	20.4	21.4	28.0

Table 1: Power savings [%].

The table shows the power savings $(P - P_{\text{opt}})/P$ for the parameters N and M. The basis for the comparison is a bus with N lines which were randomly assigned to M possible tracks. Thus, both the influence of the problem size (N) and the area (M) to the power optimization potential are reflected in the table. The optimized situation is derived after allowing the N tracks to run off-grid. FastCap extracted the self capacitances required to calculate the power. For example, if we are given 16 bus wires and 17 tracks, 16.2% capacitive switching power can be saved on average if we assigned individual spacings derived by geometric optimization.

3.3 Analysis

The savings in Table 1 are quite large. We assume this is because in our 130 nm process, the wire pitch less wire width is greater than the wire spacing, giving the geometric optimizer a lot of freedom ($s := (d_{pitch} - w)/d_{min} = 1.4$, cf. Section 2.3). We therefore also analyzed the results for processes with smaller s. Remarkably, the measured savings show the same orders of magnitudes even for $s = 1.0$. Regardless of N and s, the dependency of the savings on M is strictly concave. This means that a limited percentage of free area has to be present to exploit most of the optimization potential. For more numbers, refer to [Mül05].

Figure 4 shows unoptimized power values plotted against their optimized correspondents for thousands of instances with varying sets of κ and constant $N = 16$ and $M = 20$. The power savings aren't spread out very widely but instead occur at almost the same percentage for given unoptimized power usage, regardless of the variety of toggle rates. In Figure 5 we plotted the standard deviation of the κ against the resulting power savings, showing that one can expect specific power savings within a certain range for given standard deviation of the toggle rates. Overall, the power savings are not only quite significant but also appear to be highly predictable and show great potential in wire spacing.

4 Summary and Concluding Remarks

In this paper we formulated the wire spacing problem for low power as a *geometric program*. We assumed a set of parallel wires on equidistant tracks and approximated their wire-to-wire capacitances with the plate capacitor formula. A geometric programming solver was used to find the new individual distances between the wires. Even with very little extra routing space we can claim that the optimization potential for the capacitive power lies well in the two digit percent zone. A future work could address timing and crosstalk problems with the same wire spacing technique if the meaning of κ in (1) is changed.

References

[CKP01] J. Cong, C. Koh, and Z. Pan. Interconnect Sizing and Spacing with Consideration of Coupling Capacitance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6:1164–1169, 2001.

[Emb04] W. Embacher. *Analysis of Automated Power Saving Techniques using Power Compiler (TM)*. LIS Diploma Thesis, TU München, Germany, May 2004.

[Gro89] P. Groeneveld. Wire ordering for detailed routing. *Design & Test of Computers*, 6:6–17, 1989.

[HMH01] R. Ho, K. W. Mai, and M. A. Horowitz. The Future of Wires. *Proceedings Of The IEEE*, 89(4):490–504, April 2001.

[Lab00] Computational Optimization Laboratory. *A geometric programming solver, COPL_GP*. Internet: http://www.stanford.edu/∼yyye/Col.html, 2000.

[Moi04] K. Moiseev. *Net-Ordering for Optimal Circuit Timing in Nanometer Interconnect Design*. CCIT Report #506, Haifa, Israel, October 2004.

[MPS03] E. Macii, M. Poncino, and S. Salerno. Combining wire swapping and spacing for low-power deep-submicron buses. *Proceedings of the 13th ACM Great Lakes symposium on VLSI*, pages 198–202, 2003.

[Mül05] F. Müller. *Evaluation von Wire Spacing zur Verlustleistungsoptimierung*. LIS Student research project, TU München, Germany, June 2005.

[Nab05] K. Nabors. *FastCap*. MIT, 1992, 2005.

[SIA05] SIA. *International Technology Roadmap for Semiconductors*. Internet: http://public.itrs.net, 2005.

[WS02] A. Windschiegl and W. Stechele. Exploiting Metal Layer Characteristics For Low-Power Routing. *Power and Timing Modeling Workshop PATMOS*, 2002.

[ZR05] P. Zuber and M. Ritter. The Optimal Wire Order for Low Power CMOS. *Power and Timing Modeling Workshop PATMOS accepted*, 2005.

[ZWS05] P. Zuber, A. Windschiegl, and W. Stechele. Reduction of CMOS Power Consumption and Signal Integrity Issues by Routing Optimization. *Design, Automation & Test in Europe DATE*, March 2005.