An Automated Design Methodology for Layout Generation targeting Power Leakage Minimization

Cristiano Lazzari
ALGOS - INESC-ID
Lisbon, Portugal
Email: lazzari@inesc-id.pt

Adriel Ziesemer and Ricardo Reis
PPGC - UFRGS
Porto Alegre, Brazil
Email: {amziesemerj,reis}@inf.ufrgs.br

***Abstract*— With the advent of deep sub-micron technologies, power consumption has become one of the most important research areas in microelectronics. This paper presents a design methodology for power leakage reduction in deep sub-micron digital circuits associated with an automated layout generator. The methodology consists of finding the channel length of transistors in the non-critical paths. The sizing algorithm is basically divided in two steps. First, transistors in the most non-critical paths are sized and then a refinement phase is employed. Different from the standard cell methodology, where several versions of each cell must be inserted in the library before synthesis, in our methodology the layout is generated after the channel length of transistors are defined. Results show that power leakage was reduced to 63% in a set of combinational benchmarks, without timing penalties.**

I. Introduction

Mobile devices have taken a bigger slice of the electronics market. For this reason, power consumption becomes one of the most important areas of the microelectronics industry. High power dissipation shortens battery life, reduce circuits reliability and performance [1].

Designers have been concerned with the dynamic power consumption over all the technology evolution. On the other hand, power leakage was not taken into account due to its low amount in the total power consumption. However, static power is drastically increasing in deep sub-micron technologies (DSM) [2]. Furthermore, the power dissipation from chip leakage is approaching the dynamic power, and off-state sub-threshold leakage is expected to exceed the dynamic power consumption as technology drops below the 65nm feature size [3].

Dealing with these deep sub-micron challenges require careful planning [4] and emphasize the need for electronic design automation (EDA) tools able to automatically generate and validate integrated circuits. In addition, static power consumption must be addressed in the design of new systems.

In this work, we present a methodology targeting power leakage reduction in circuits by increasing the transistor channel length without insert timing penalties to the design. The technique consists basically of finding transistors in the non-critical paths and increasing their gate length. The layout is automatically generated, once the transistor length definition is performed.

This paper is organized as follows. Section II discusses some state-of-the-art techniques for power leakage reduction. The proposed gate length biasing algorithm is presented in Section III. Section IV gives an overview about the transistor level layout generation, use to generate the circuit layout. Results are presented in Section V and, finally, Section VI concludes the paper.

II. Techniques and Related Works

Many techniques have been proposed in the last years aiming at reducing the static power consumption of circuits. The power leakage problem has been addressed at design stage by various techniques such as *transistor stacking*, *sleep transistor insertion*, V_{DD} *assignment* and *gate length biasing*.

Some of these techniques such as the transistor stacking and the gate length biasing demand the design of new cells because the transistors structure is modified.

Sleep transistors can be applied in different granularities. High granularity allows to control the power leakage for small blocks (or even gates), but increase the design complexity and area-timing overhead. Low granularity presents small penalties to the circuit but the power leakage control may be reduced. The V_{DD} assignment technique involves a careful planning in order to accurately place cells with different V_{DD} voltages and route supply lines. These techniques are discussed in the following.

A. Transistor Stacking

Transistor stacking is the technique where transistors are duplicated in order to reduce the sub-threshold leakage when transistors are in standby mode [5]. Basically, transistors are connected in series, working as one transistor with larger channel length. Figure 1 shows an example of stacked transistor structure.

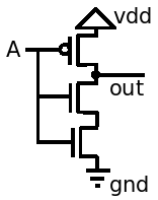

Fig. 1. An NMOS two-stack inverter reduces the sub-threshold leakage when input is set to logic “0”.

978-1-4244-5091-6/09/$25.00 ©2009 IEEE

Sometimes the stacked transistor widths are divided by two in order to ensure the same input load. Thus, the delay of the connected gate and the switching power remain unchanged. However, the gate delay is increased due to the current reduction and timing closure may not be achieved, as a consequence.

B. Sleep Transistors

The main idea in the insertion of *sleep transistors* is to cut the sub-threshold current when CMOS gates are in standby mode [6], [7]. The methodology seems to be very interesting because a whole block can be turned off, but the main drawback of this technique is related to delay penalties.

The activation time needed to change these transistors from "OFF" to the state "ON", may be longer than the response time of many cells in the design (specially those placed close to the primary inputs). Thus, extra time related to the transistors activation must be considered.

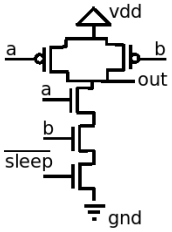

Fig. 2. An example of fine-grained sleep transistors.

Figure 2 illustrates an example of fine-grained sleep transistor. Usually sleep transistors do not control the shutoff of a single gate, but a set of gates in order to reduce the area overhead.

C. V_{DD} Assignment

V_{DD} assignment consists of assigning different supply power voltages to different cells in the circuit [8]. Thus, cells with lower V_{DD} voltage present a smaller power consumption.

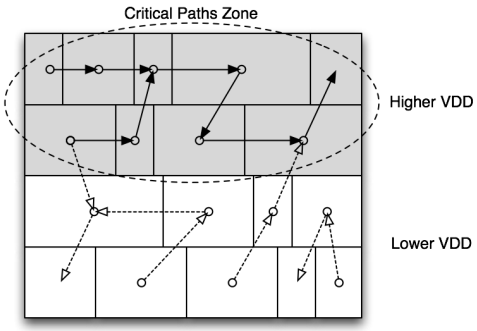

Fig. 3. An example of V_{DD} assignment where cells are grouped according the delay critically and the V_{DD} is appropriately assigned.

The V_{DD} assignment technique demands careful analysis in relation to the cells placement and supply lines routing. Cells must be grouped in order to reduce the supply routing, increasing the wire length and routing congestion. Figure 3 illustrates the V_{DD} assignment where cells are grouped according the delay critically and the V_{DD} is appropriately assigned.

Considering that the power planning is already a hard task in sub-micron technologies due to local hot spots, insufficient power supply and signal integrity problems, V_{DD} assignment inserts more challenges in the traditional design flow.

D. Gate Length Biasing

Gate length biasing consists of adjusting the channel length of transistors to reduce the power leakage [9], [1], [10], [11]. It is possible since leakage, in opposite to delay, increases with the reduction of the gate length. Thus, fastest paths can be sized while transistors in the critical paths may be maintained with the same length in order to deal with timing closure.

Gupta *et al.* [1] concludes that 10% is an upper bound for gate length biasing. In our work, we justify the gate length biasing with the Figure 4. These data were obtained by SPICE simulations with the predictive technology models presented in [12].

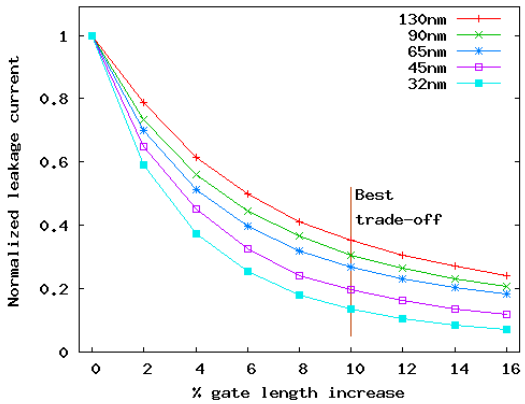

Fig. 4. Leakage current in deep sub-micron technology processes. Curves show the power reduction gain is very small when transistor channel length is larger than 10%.

Figure 4 shows the normalized leakage current in deep sub-micron technologies as a function of the transistor length biasing. We can highlight the behavior of the power leakage where, from a given gate length (more than 10% larger than the nominal), the efficacy of the technique is reduced. In other words, power leakage reduction gain is very small when transistor channel length is larger than 10%. Based on this experiment, we empirically decide to increase the transistors length in 10%, as proposed in [1].

III. The Gate Biasing Methodology

The gate biasing methodology presented in this paper is basically divided in three steps. First, the circuit described in a SPICE netlist is sized aiming at meeting circuit timing specifications. This step consists of analyzing critical paths and finding transistor widths according to the circuit structure and parasitics connected to them.

The second step is the basis of this paper. After transistor widths are defined, the channel length of the transistors in the non-critical paths is modified in order to reduce the power

leakage (gate length biasing). The last step is the layout generation with a transistor level layout generator.

Algorithm 1 : The proposed gate length biasing methodology.

Require: Netlist N, Design Rules D
Ensure: Layout L

1: $N_{size} \Leftarrow$ ApplyTransistorSizing(N); {Apply transistor sizing for timing [13]}
2: $\tau \Leftarrow$ getTiming(N_{size}); {Synopsys Pathmill[14]}
3: {Start gate length biasing}
4: $P \Leftarrow$ getAllPaths(N_{size}); {Synopsys Pathmill[14]}
5: $T \Leftarrow 80\%$
6: **while** $T \geq 0\%$ **do**
7: $G \Leftarrow$ getGatesInNonCriticalPaths(P, T);
8: $N_{new} \Leftarrow$ applyGateLengthBiasing(N_{size}, G);
9: $\tau_{new} \Leftarrow$ getTiming(N_{new}); {Synopsys Pathmill[14]}
10: **if** $\tau_{new} \geq \tau$ **then**
11: break;
12: **else**
13: $T \Leftarrow T$ - 10%;
14: **end if**
15: **end while**
16: $G \Leftarrow$ getAllGates(P) $\setminus$ G;
17: **for all** $g \in G$ **do**
18: $N_{temp} \Leftarrow N_{new}$
19: $N_{new} \Leftarrow$ applyGateLengthBiasing(N_{temp}, g);
20: $\tau_{new} \Leftarrow$ getTiming(N_{new}); {Synopsys Pathmill[14]}
21: **if** $\tau_{new} \geq \tau$ **then**
22: $N_{new} \Leftarrow N_{temp}$
23: **end if**
24: **end for**
25: {Finish gate length biasing}
26: $L \Leftarrow$ generateLayout(N_{new}, D); {Generate layout [15]}

Algorithm 1 presents the whole process. The timing optimization and the layout generation are not discussed in this paper. We refer the reader to verify the works published in [13] and [15] for further details. However, Section IV gives a brief overview about these techniques. The transistor sizing step is shown in Line 1 and the layout generation is shown in line 26.

The gate length biasing algorithm is basically divided in two parts. The first part is represented by the **while** command (line 6 to 15) where T is the amount of gates in the non-critical paths are modified. For example, the algorithm starts with all gates G in the 80% non-critical paths ($T = 80\%$) and reduction of 10% of these paths for each iteration. Command `applyGateLengthBiasing(`N_{size}`,`G`)` (line 8) increases the channel length of transistors in G of the netlist N_{size} and the timing is analyzed with the Synopsys Pathmill [14] (command `getTiming(`N_{new}`)`). If the circuit delay is greater than the initial circuit (line 10), T is reduced to 70% (line 13) and the process is repeated.

The second part is a refinement step (lines 16 to 24). The refinement step consists of increasing sequentially the gate length of all transistors not modified in the first step. If the delay increases, the gate length biasing for the gate g is discarded.

IV. The Transistor Level Layout Generator

A transistor level layout generation methodology is presented in [15]. This methodology consists of implementing the layout of a whole block based on a list of transistors. In other words, the methodology eliminates the use of standard cells in the design flow. The layout of every single gate is generated on demand.

In the transistor level layout generation methodology, timing and power are predicted before layout generation [13]. Then, when the circuit layout is generated, width and length of every transistor is already known, avoiding timing and power overestimation.

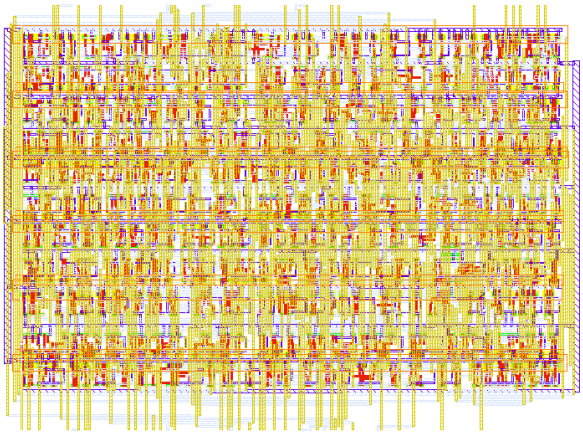

Fig. 5. Layout of the C1908 circuit generated by the transistor level design flow presented in [15].

Figure 5 shows an example of layout generated by the methodology presented in [15] where the layout is generated from a transistor netlist. Width and length of each transistor in the netlist was previously tailored before layout generation.

From the point of view of the transistor level layout generation, every technique presented in Section II can be applied. However, gate length biasing is used in our methodology because it can be entirely performed at layout level, without new transistors insertion or process modifications.

Although the gate length biasing methodology shown in this paper can be applied with the traditional standard cell design flow [16], new versions of each library cell must be implemented, validated and characterized, increasing the size of the library as a consequence. Differently, transistor level design implementation allows fast synthesis of circuits.

V. Experimental Results

We defined a 10% upper bound to transistor length sizing for our experiments. As shown in Section II-D, the power leakage gain is very small with larger transistor channel lengths.

Results in Table I show that the gate length biasing is very efficient on the reduction of the power leakage. The gate length biasing was applied to an average of 70% of the cells and resulting circuits spend around 63% of the initial power leakage in average. It is important to remark that the delay is not increased after gate length biasing.

TABLE I

THE GATE LENGTH BIASING TECHNIQUE FOR SOME ISCAS'85 BENCHMARKS WITH A $65nm$ TECHNOLOGY PROCESS.

Benchmark	Sized cells		Power leakage		
	#Sized/Total	(% Sized)	Before (μW)	After (μW)	After (%)
C432	139/209	66	3.5	2.1	60
C499	208/296	70	12.5	8.1	64
C880	290/359	80	10.6	5.9	55
C1355	247/446	55	10.3	6.6	64
C1908	245/372	65	14.4	11.0	76
C3540	526/704	74	20.1	12.3	61
Average power leakage after gate length biasing					63%

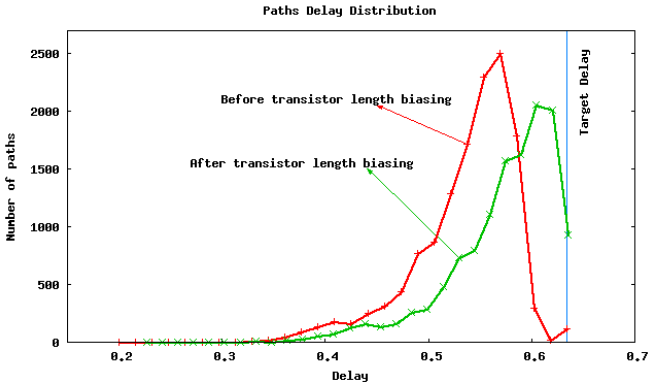

Fig. 6. Delay paths before and after gate length biasing for the circuit C1908.

Delay paths before and after the gate length biasing are shown in Figure 6. As expected, curves show a larger number of paths close to the target delay (critical delay) after the gate length biasing. These curves illustrate the delay paths shifting close to the critical delay due to the gate length biasing.

An important characteristic is the relation between variability and the large number of paths close to the target delay. Although selectively biased circuits show much less sensitivity to both intra and inter die variations [1], process variability may increase the delay of some paths causing a timing violation. For this reason, worst case corners are used at simulation time to reduce the impact of the variability in the circuit delay.

VI. CONCLUSION

This paper presents an automated methodology for layout generation targeting the reduction of static power consumption by increasing the channel length of transistors in non-critical paths. The proposed gate biasing methodology is applied in association with a transistor sizing tool (targeting circuit delay reduction) and an automatic layout generation tool. The proposed methodology can be directly applied to combinational blocks as a way to improve their power consumption.

Results show circuits consuming around 63% of power leakage in comparison with circuits where the technique was not applied. The area overhead is neglected due to the increasing of only 10% in the length of the some transistors. These results prove the efficacy of the methodology.

On going works include the research of methodologies to improve the selection of transistors such as statistical methods.

REFERENCES

[1] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Selective gate-length biasing for cost-effective runtime leakage control," in *Proceedings of the annual conference on Design automation, DAC, 2004*. New York, NY, USA: New York: ACM Press, 2004, pp. 327–330.

[2] G. E. Moore, "No exponential is forever: but ¨ forever¨ can be delayed!" in *IEEE International Solid-State Circuits Conference, ISSCC, 2003*, vol. 1, 2003, pp. 20–23.

[3] Kim, N. S. et al., "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, 2003.

[4] R. Otten, R. Camposano, and P. Groeneveld, "Design automation for deepsubmicron - present and future," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2002*. Washington, DC, USA: [S.l.]: IEE Computer Society, 2002, pp. 650–657.

[5] Narendra, S. et al., "Full-chip sub-threshold leakage power prediction model for sub-0.18 cmos," in *Proceedings of the international symposium on Low power electronics and design, ISLPED, 2002*. New York, NY, USA: New York: ACM Press, 2002, pp. 19–23.

[6] C. Long and L. He, "Distributed sleep transistor network for power reduction," in *Proceedings of the conference on Design automation, DAC, 2003*. New York, NY, USA: New York: ACM Press, 2003, pp. 181–186.

[7] Babighian, P. et al., "Post-layout leakage power minimization based on distributed sleep transistor insertion," in *Proceedings of the international symposium on Low power electronics and design, ISLPED, 2004*. New York, NY, USA: New York: ACM Press, 2004, pp. 138–143.

[8] Y. Lin and L. He, "Leakage efficient chip-level dual-vdd assignment with time slack allocation for fpga power reduction," in *Proceedings of the annual conference on Design automation, DAC, 2005*. New York, NY, USA: New York: ACM Press, 2005, pp. 720–725.

[9] N. Sirisantana, L. Wei, and K. Roy, "High-performance low-power cmos circuits using multiple channel length and multiple oxide thickness," in *ICCD '00: Proceedings of the 2000 IEEE International Conference on Computer Design*. Washington, DC, USA: IEEE Computer Society, 2000, p. 227.

[10] A. B. Kahng, S. Muddu, and P. Sharma, "Defocus-aware leakage estimation and control," in *Proceedings of the international symposium on Low power electronics and design, ISLPED, 2005*. New York, NY, USA: New York: ACM Press, 2005, pp. 263–268.

[11] S. Bhardwaj, Y. Cao, and S. Vrudhula, "Statistical leakage minimization through joint selection of gate sizes, gate lengths and threshold voltage," in *Proceedings of the conference on Asia South Pacific design automation, ASP-DAC, 2006*. New York, NY, USA: New York: ACM Press, 2006, pp. 953–958.

[12] W. Zhao and Y. Cao, "Predictive technology model for nano-cmos design exploration," *J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 1, p. 1, 2007.

[13] C. Santos, G. Wilke, C. Lazzari, J. L. Guntzel, and R. Reis, "A transistor sizing method applied to an automatic layout generation tool," in *Proceedings of the Symposium On Integrated Circuits and Systems Design, 2003*, 2003, pp. 303–307.

[14] www.synopsys.com, 2009.

[15] C. Lazzari, C. V. Domingues, J. L. Guntzel, and R. A. L. Reis, "A new macro-cell generation strategy for three metal layer cmos technologies," in *Proceedings of the 2003 VLSI-Soc*, Darmstadt, Germany, 2003, pp. 143–147.

[16] S. Saumil, G. Puneet, and K. Andrew, "Standard cell library optimization for leakage reduction," in *DAC '06: Proceedings of the 43rd annual conference on Design automation*. New York, NY, USA: ACM, 2006, pp. 983–986.