Logic Synthesis of Low-power ICs with Ultra-wide Voltage and Frequency Scaling

Yu Pu, Juan Echeverri, Maurice Meijer, Jose Pineda de Gyvez
Department of Digital Architectures Circuits and Signal Processing
NXP Semiconductors, Central R&D, The Netherlands
{yu.pu, juan.diego.echeverri.escobar, maurice.meijer, jose.pineda.de.gyvez}@nxp.com

***Abstract*— For low-power digital ICs with ultra-wide voltage and frequency scaling (e.g., from the nominal supply voltage to the sub/near-threshold regime), achieving design closure can be a big challenge, especially when speed limits are pushed at very different voltages. This paper shares a practical logic synthesis recipe that helps to fulfill tight timing constraints. Our method includes: *i*) synthesizing circuits at a high voltage; *ii*) over-constraining maximal transition time; *iii*) pruning standard cell library based on cell delay degradation factor across voltages. This approach shows effectiveness on an industrial 90nm low-power micro-controller.**

***Keywords*— logic synthesis; ultra-wide voltage and frequency scaling; ultra-low-power**

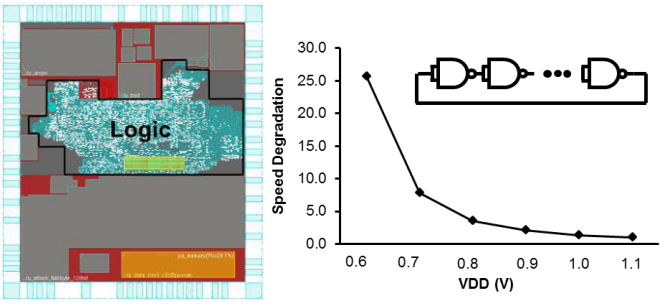
I. INTRODUCTION

Advanced EDA tools enable multi-corner multi-mode (MCMM) analysis and optimization [1]. However, the practice today is that, tools can support very limited mode and corner scenarios. The increase in the number of mode/corner scenarios leads to increased iterations hence larger difficulty in design closure. Ultra-wide voltage and frequency scaling, e.g., [2], exacerbates issues, as varying voltage and frequency results in more modes and corners than tools can handle.

Our design is an ultra-low-power microcontroller for applications with burst characteristics, i.e., it infrequently requires high performance and most of the time it only requires a near-standby mode. Ultra-wide range voltage and frequency scaling is applied to the logic part, because it occupies less than 20% of the total chip area (see the layout view in Figure 1(a)) but burns more than 70% of the total power. The aiming leakage profile requires the design to be implemented in a 90nm High V_T (HV_T) process technology. Logic synthesis is done at the slow-slow (SS) corner to reserve adequate margins for mass production. Figure 1(b) shows the speed degradation factor (normalized to the speed at 1.1V V_{DD}) of a ring oscillator consisting of an odd number of 2-input NAND gates. As seen, the threshold voltage at the SS corner of this process is around 0.8V, because beyond 0.8V the circuit speed drops linearly and below 0.8V the speed drops exponentially.

To maximize voltage scaling thereby power savings, the operating frequencies are pushed at both the nominal voltage and the ultra-low-voltage. Denoting $f_{max}(V_{DD})$ as the maximal frequency that can be achieved by synthesizing the micro-controller directly at V_{DD} (i.e., the maximum frequency for single V_{DD} scenario), we get $f_{max}(1.1V)$=106.09MHz (nominal mode), $f_{max}(0.9V)$=57.49MHz (near-threshold mode) and $f_{max}(0.65V)$=3.98MHz (sub-threshold mode). At 1.1V, 0.9V and 0.65V, our targeted operating frequencies are 100MHz, 50MHz and 3.6MHz. Considering the necessary margins for design variations in the back-end stage, the synthesized frequencies should be higher than the specified, so the timing constraints are actually close to the limits.

This design runs into difficulty with timing closure. As we will show in Section II, the behaviors of synthesis tool for timing optimization at a high voltage and an ultra-low-voltage are so conflicting that convergence in one scenario creates violations in other scenarios. The resulting "bouncing" effect causes failures to design convergence.



(a) (b)

Fig. 1: (a) Layout view of the design (b) Speed degradation factor with V_{DD} scaling in the 90nm HV_T process

II. CONFILICTING OPTIMIZATION PREFERENCES AT HIGH VOLTAGE AND LOW VOLTAGE

Synthesis tools use very complicate cost functions to concurrently optimize timing, area, power and signal integrity. To understand how tools behave at a high voltage and at an ultra-low-voltage, we did the following synthesis experiments, as listed in Table I:

a) Synthesizing at 0.65V V_{DD} for $f_{max}(0.65V)$. When V_{DD} scales to 0.9V and 1.1V, the speed losses compared to $f_{max}(0.9V)$ and $f_{max}(1.1V)$ are more than 10% and 30%.

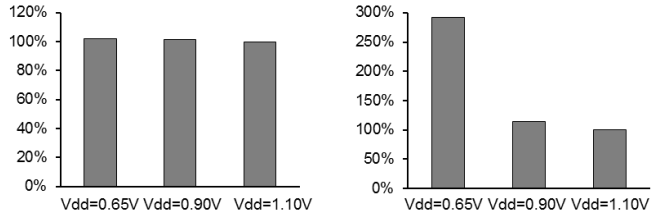
b) Synthesizing at 1.1V V_{DD} for $f_{max}(1.1V)$. When V_{DD} scales to 0.65V, the speed loss compared to $f_{max}(0.65V)$ is more than 10%. At 0.9V V_{DD}, no speed loss is observed compared to $f_{max}(0.9V)$.

c) Synthesizing at the intermediate 0.9V V_{DD} for $f_{max}(0.9V)$. At 0.65V and 1.1V, the speed losses compared to $f_{max}(0.65V)$ and $f_{max}(1.1V)$ are over 10% and 20%, respectively.

978-3-9815370-2-4/DATE14/©2014 EDAA

TABLE I. SYNTHESIS EXPERIMENTAL RESULTS

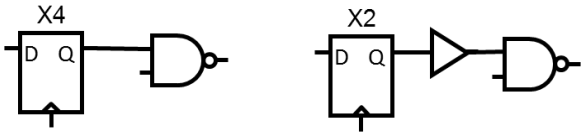
	0.65V (MHz)	Speed loss w.r.t. *f*max(0.65V)	0.9V (MHz)	Speed loss w.r.t. *f*max(0.9V)	1.1V (MHz)	Speed loss w.r.t. *f*max(1.1V)
a) synthesizing at 0.65V V_{DD}	3.98	0.00%	50.31	12.49%	69.41	34.58%
b) synthesizing at 1.1 V V_{DD}	3.52	11.70%	57.93	-0.76%	106.09	0.00%
c) synthesizing at 0.9V V_{DD}	3.58	10.09%	57.49	0.00%	82.42	22.31%
d) synthesizing at 1.1V V_{DD}, max transition time over-constraining	3.59	9.65%	57.49	0.00%	105.23	0.81%
e) synthesizing at 1.1V V_{DD}, library pruning	3.59	9.67%	57.77	-0.49%	105.69	0.38%
f) synthesizing at 1.1V V_{DD}, max transition time over-constraining, library pruning	3.66	7.89%	58.49	-1.74%	106.64	-0.52%



(a) logic area (b) area of inserted buffers

Fig. 2: Synthesis results normalized to results of 1.1V V_{DD} synthesis: (a) logic area; (b) area of inserted buffers

Figure 2(a) shows the logic areas synthesized at *f*max(V_{DD}). The areas are normalized to the area at 1.1V V_{DD} for *f*max(1.1V). Interestingly, in all three cases, the areas are similar, implying that the synthesis tool is capable of obtaining constant and high area efficiencies at very different V_{DD} points. However, the total area of inserted buffers at *f*max(0.65V) is almost 3X compared with that at *f*max(1.1V). Further analysis reveals the following trends: *i*) at a high voltage, the tool uses logic gate sizing more often than buffer insertion; *ii*) at an ultra-low-voltage, the transition time becomes exceedingly long, so buffer insertion is more effective and efficient than gate sizing. Figure 3 illustrates an example of optimizing a part of a logic path at 1.1V and 0.65V V_{DD}s by the tools.



(a) (b)

Fig. 3: Example: (a) Gate sizing is preferred at the nominal voltage; (b) Buffer insertion is preferred at ultra-low-voltages.

On one hand, the inserted buffers at the low voltages inevitably increase the logic depths of critical paths. When scaling to the nominal V_{DD}, it results in a severe speed loss, as clearly evidenced by experiment (*a*) and (*c*). On the other hand, gate sizing strategy adopted at the nominal voltage synthesis is not effective enough to address the significantly degraded transition time and gate delay at ultra-low-voltages. The two conflicting optimization preferences cause the design closure failure. By comparing the experimental results of *a*), *b*) and *c*), we conclude that, for our design, logic synthesis at a high voltage is more beneficial than at a low voltage.

III. MAXIMAL TRANSITION TIME OVER-CONSTRAINING AND STANDARD CELL LIBRARY PRUNING

To properly address the degradations of transition time and gate delay from 1.1V to 0.65V and to minimize the speed loss at 0.65V V_{DD} compared to *f*max(0.65V), the following three approaches were experimented, as also listed in Table I:

d) Synthesizing at 1.1V V_{DD} and over-constraining the maximal transition time. For example, by restricting the maximal transition time to be less than 240ps at 1.1V, we guarantee that the worst maximal transition time is less than 3ns at 0.65V.

e) Synthesizing at 1.1V V_{DD} and pruning standard cell library. By parsing and comparing the timing lookup tables in the liberty timing files characterized at 1.1V and 0.65V, we avoid using standard cells whose average gate delay degradation factors are more than 20X. In this way, around 1/10 of the standard cells are filtered out and the remaining subset of the cell library is allowed in logic synthesis.

f) Combining the approaches of *(d)* and *(e)* to restrict both the maximal transition time and gate delay degradations.

In this effort, speed is improved at 0.65V V_{DD}. For experiments *(d)* and *(e)*, the speed losses compared to *f*max(0.65V) are less than 10%. Finally, *(f)* reduces the speed loss to 7.9% at the cost of 10% increased logic area compared to synthesis at 1.1V V_{DD} for *f*max(1.1V). The total chip area (including analog IPs, memories and IO pads) is increased by 2%, which is acceptable.

IV. CONCLUSIONS

A practical logic synthesis recipe is presented for low-power ICs with ultra-wide voltage and frequency scaling. This approach includes: *i*) synthesizing circuits at a high voltage; *ii*) over-constraining maximal transition time; *iii*) standard cell library pruning based on gate delay degradation. The effectiveness of the approach is proven by an industrial microcontroller in a 90nm HV_T process.

REFERENCES

[1] A. Narayanan and S. Jilla, "Ultra-low power requires MCMM," http://low-powerdesign.com/article_mentor_NarayananJilla.htm

[2] Y. Pu, J. Pineda de Gyvez, H. Corporaal and Y. Ha, "An Ultra Low-Energy/Frame Multi-standard JPEG Co-processor in 65nm CMOS with Sub/Near Threshold Power Supply," IEEE Journal of Solid-State Circuits (JSSC), Vol. 45, No. 3, pp. 668-680, Mar. 2011.