ATLANTIS PRESS
Advances in Engineering Research (AER), volume 61
International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2017)

Using Design Compiler Topographical Technology for Modern Process

Qian Liang [1, a], Ligang Hou [1, b] and Jinhui Wang [2, c]

[1]VLSI and System Lab, Beijing University of Technology, Beijing 100124, China

[2]Department of Electrical and Computer Engineering, North Dakota State University, ND 58102, USA

[a]liangqian@emails.bjut.edu.cn, [b]houligang@bjut.edu.cn, [c]wangjinhui@bjut.edu.cn

Keywords: Interconnect Parasitics, Topographical, Physical Constraints.

Abstract. As the technology scales into deep submicron regime, accurate estimate of interconnect parasitics has become one of important factors on path delay calculation. Design Compiler Topographical technology leverages the Synopsys physical implementation solution to derive the "virtual layout" of the design, thus the tool can accurately predict and use real net capacitances instead of statistical net approximations based on wire load models (WLM). A synthesis method based on WLM mode and topographical mode for 8051 micro-controller in 90 nm technology is presented in this paper. Results show that the Design Compiler Topographical technology can accurately predict post-layout timing and ensure closed correlation to the final physical design.

1. Introduction

In ultra-deep submicron designs, interconnect parasitics have a major effect on path delays, thus accurate estimates of resistance and capacitance are critical in IC physical design [1]. Traditionally, Wire Load Models (WLM) are used to estimate the effect of wire length and fanout on the resistance, capacitance, and area of nets during RTL synthesis [2]. However, they are difficult to be compatible with modern process due to the probably of over-estimate or under-estimate. Design Compiler Topographical (DCT) technology provides solution to accurately predict post-layout timing, area, and power during RTL synthesis using the "virtual layout" of the design. This approach eliminates the need for overconstraining the design or using optimistic wire load models in synthesis. The accurate prediction of net capacitances drives Design Compiler to generate a netlist that is optimized for all design goals. It also results in a better starting point for physical implementation.

2. Design Compiler Topographical Technology

Design Compiler topographical synthesis performs placement-driven design mapping and optimization in order to achieve high quality of results (QoR), tight correlation between Design Compiler and IC Compiler, and improved routing [3]. It uses Synopsys' placement and optimization technologies to drive accurate timing prediction within synthesis, ensuring closed correlation to the final physical design.

2.1 Two-pass topographical synthesis flow.

Design Compiler topographical mode to IC Compiler is a two-pass flow. Initial synthesis with default physical constraints is performed during the first pass in order to generate the initial netlist. Synthesis with actual floorplan is performed during the second pass.

Fig 1 shows the steps needed to be performed during the two-pass topographical synthesis flow. During the first pass, we need to create the initial netlist in Design Compiler topographical mode for IC Compiler design planning and proceed to the design planning phase in IC Compiler in order to generate a floorplan with physical constraints. During the second pass, it uses floorplan constraints in Design Compiler topographical mode to create an optimazied netlist and performs detailed design closure in IC Compiler.

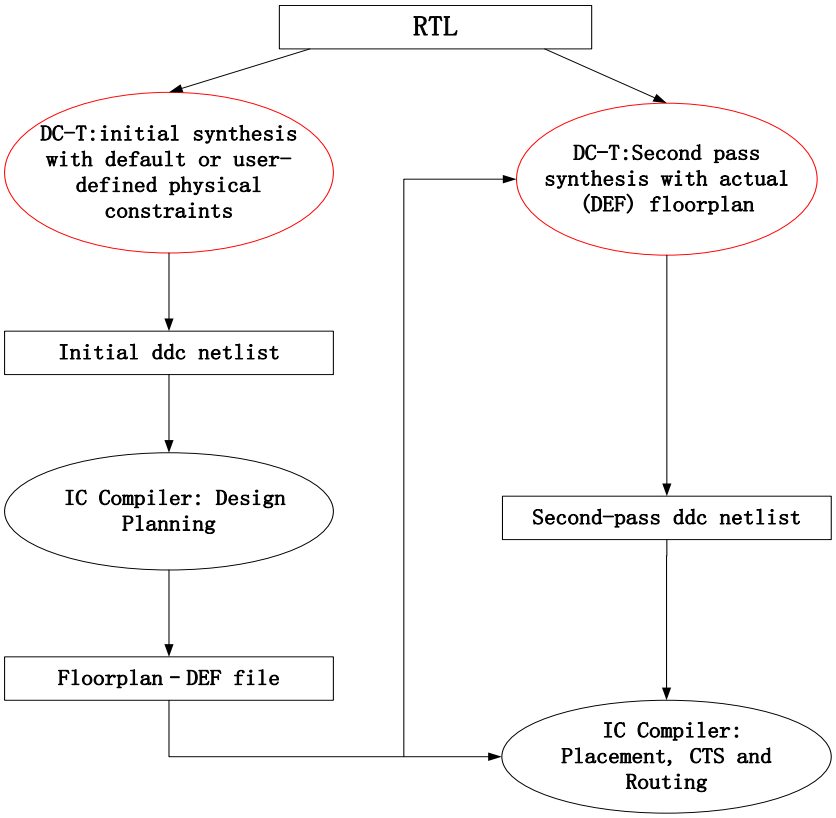

Fig. 1 Two-pass topographical synthesis flow

2.2 Inputs and outputs in topographical mode.

Fig 2 shows the inputs and outputs in Design Compiler topographical mode.

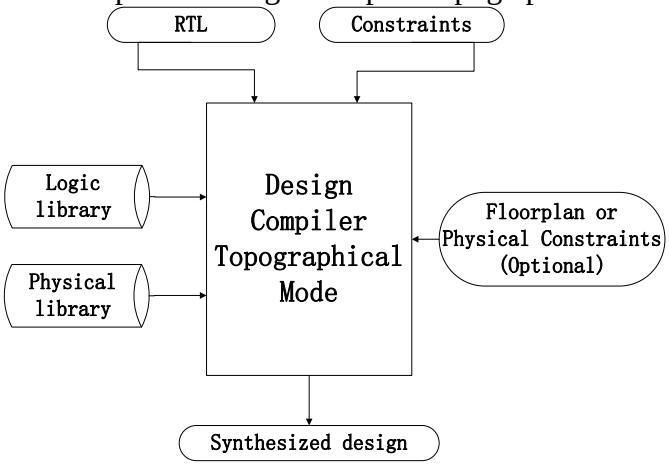

Fig. 2 Inputs and outputs in topographical mode

Table 1 and Table 2 describe the inputs and outputs in topographical mode.

Table 1. Inputs in topographical mode

Input	Description
Design	RTL or gate-level netlist
Constraints	Timing and optimization constraints
Logic library	Liberty format (.lib or .db)
Physical library	Milkyway format
Floorplan or physical constraints	High-level physical constraints that determine items such as core area and shape, port location, macro location and orientation, voltage areas, placement blockages, and placement bounds

Table 2. Outputs in topographical mode

Output	Description
.ddc	This format contains back-annotated net delays and constraints.
Milkyway	This format contains back-annotated net delays and constraints.
ASCII	This format does not contain back-annotated delays or Synopsys Design Constraints (SDC).

2.3 Floorplan physical constraints.

The prime reason for using floorplan constraints in topographical mode is to accurately estimate interconnect parasitics and improve timing correlation with the post-place-and-route tools, such as IC Compiler, by considering floorplanning information during optimizations. Design Compiler topographical mode supports high-level physical constraints such as die area, core area and shape, port location, macro location and orientation, keepout margins, placement blockages, preroutes, bounds, vias, tracks, voltage areas, and wiring keepouts, as Fig 3 shows.

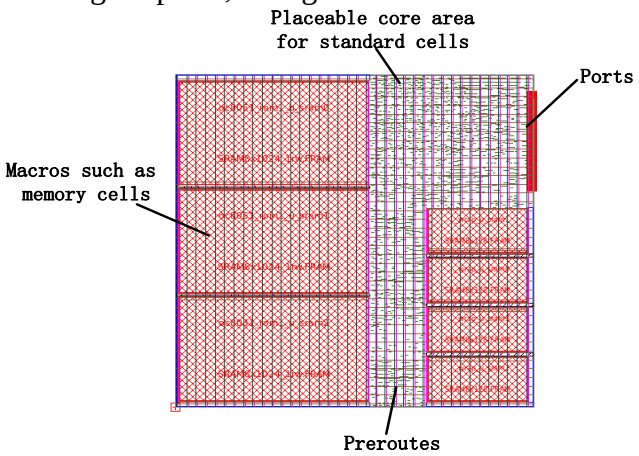

Fig. 3 Floorplan physical constraints

3. Using Design Compiler Topographical Technology

Based on 8051 micro-controller in 90nm technology, we perform synthesis with Design Compiler in WLM mode and topographical mode separately.

3.1 Basic synthesis flow.

Table 3. Basic synthesis flow

WLM	DCT
dc_shell	dc_shell –topographical
set search_path "$search_path ./libraries" set link_library "*max_lib.db" set target_library "max_lib.db"	
	create_mw_lib -technology $mw_tech_file -mw_reference_library $mw_reference_library $mw_lib_name open_mw_lib $mw_lib_name
read_verilog rtl.v source top_constraint.sdc	
set_wire_load_model -name "10x10"	extract_physical_constraints floorplan.def
compile	compile_ultra
report_constraints change_names -rules verilog -hierarchy write -format ddc -hierarchy -output top_synthesized.ddc	

Scripts that perform the basic synthesis flow in WLM mode and in topographical mode are showed in Table 3.

As we can see from Table 3, the most significant difference in WLM mode and topographical mode is that the former uses wire load models in logic library while the later uses floorplan or physical constraints instead.

3.2 Analysis of synthesis results.

Once synthesis is finished, we need to check the quality of results (QoR) by variable reports including timing, area, power, and so on.

Table 4. Timing report

Path Group	Number of violated paths		Worst negative slack	
	WLM	DCT	WLM	DCT
clk	361	53	-13.15	-0.03
in2reg	277	-	-12.41	-
reg2out	16	16	-13.26	-0.94
in2out	8	-	-12.53	-

Table 4 lists timing report in WLM mode and topographical mode. In order to analyze and optimize timing effectly, timing paths are divided into path groups based on the clock controlling the endpoint [4]. They are clk, in2reg, reg2out and in2out. Compared the timing report in these two modes, the number of violated paths are less and the worst negative slack is much smaller in topographical mode than in WLM mode.

Table 5. Area and power report

	WLM	DCT
area(um2)	606761.88	624982.81
power(pf)	2303798528	3959390464

Table 5 shows that the area and power in topographical mode become larger due to the compromise with the better timing results. Actually, Design Compiler topographical technology provides physical tools with more realistic results ensuring correlation and predictability.

4. Summary

This paper presents the Design Compiler Topographical technology and compares the results of synthesis in WLM mode and topographical mode. The results proves that the Design Compiler Topographical technology can accurately predict post-layout timing and lead to better correlation to the final physical design.

Acknowledgments

This work is supported by the Beijing Municipal Natural Science Foundation (No. 4152004), and National Natural Science Foundation of China (No. 61204040).

References

[1]. Synopsys. Design Compiler User Guide. Version G-2012.06, June 2012.
[2]. Chadha R, Bhasker J. Static Timing Analysis for Nanometer Designs [M]. Springer US, 2009.
[3]. Synopsys. Design Compiler and IC Compiler Physical Guidance Technology Application Note. Version G-2012.06, June 2012.
[4]. Synopsys. PrimeTime and PrimeTime SI User Guide. Version G-2012.06, June 2012.