King Fahd University of Petroleum and Minerals

Computer Engineering Department

COE 561
Digital Systems Design and Synthesis
(Course Activity)

Synthesis using Synopsys Design Compiler Tutorial
The Synthesis Flow (What, How & Why?)

Presented by
Mohammad IbrahimAl-Behwashi

Advisor
Dr. Aiman El-Maleh

Date 16-11-2006
Fall Semester (061)

Agenda

- **Introduction to Design Compiler**
- **The Synthesis Process Flow**
- **Explain Each Step in the Flow**
- **Apply an example of a combinational circuit under Design Analyzer**
- **Apply an example of a sequential circuit under Design Analyzer**
- **Show the commands of each step**

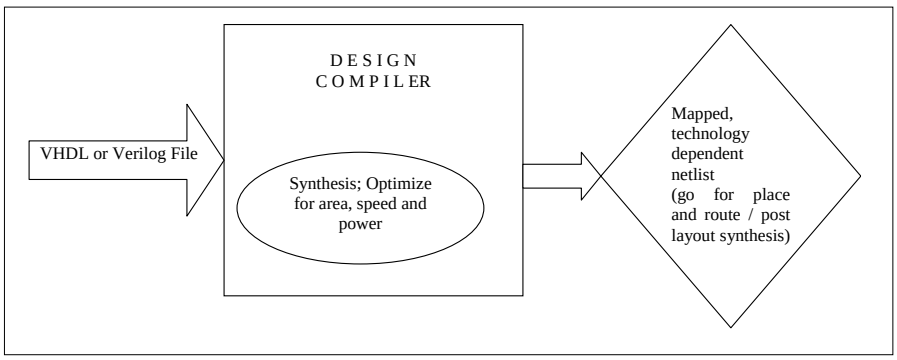
2

Introduction

- **The Design Compiler** is the core of the Synopsys synthesis software products. It includes tools that synthesis the HDL designs into optimized technology-dependent, gate level designs. It can optimize for speed, area and power.

3

Basic Synthesis Flow



A view of the Design Compiler Function

4

Synthesis, Optimization and Compilation

- **Synthesis**: is the process that generates a gate-level netlist for an IC design that has been defined using a Hardware Description Language (HDL). Synthesis includes reading the HDL source code an optimizing the design from that description.
- **Optimization**: is the step in the synthesis process that attempts to implement a combination of library cells that best meet the functional timing, and area requirements of the design.
- **Compile**: is the Design Compiler command and process that executes the optimization step. After reading in the design performing the necessary tasks, the compile command is invoked to generate a gate-level netlist for the design. `

5

Basic Synthesis Flow

1) Develop HDL Files
2) Specify Libraries
3) Read Design
4) Define Design Environment
5) Set Design Constraints
6) Optimize the Design
7) Analyze and Resolve the Design Problems

6

1) Developing HDL Files and 2) Specifying Libraries

- Design data management, design partitioning and HDL code style ➔ affect the synthesis and optimization processes
- Of course, use another program to do so. But you don't need to compile the HDL file.
- **Libraries**: link, target and symbol libraries
- **Link and target libraries:** define the semiconductor vendor's set for cells and related information, such as cell names, cell pin names, delay arcs, pin loading, design rules and operating conditions.
- **Symbol library:** defines symbols for schematic and viewing the design. (Needed if GUI is to be used).
- Showing an example of a library

7

Step (3): Reading, Analyzing and Elaborating Designs

- **Read**: ➔ Loading into the memory. (to see the designs loaded, use `list_designs`)
- **Analyze**: Reads an HDL source file. Checks it for errors. Creates HDL library objects in an HDL intermediate format.
- **Elaborate**: Creates a technology-independent design from the intermediate files produced during analysis.

8

Step(4): Defining the Design Environment

- Define the environment in which the design is expected to operate. This includes, operating conditions, wired load models and system interface characteristics.
- **Operating Conditions**: temperature, voltage, and process variations. Wire load models estimate the effect of wire length on design performance.
- **System interface characteristics** include input drives, in/out loads and fan-out loads.
- The environment model directly affects the design synthesis results.
- **Commands**: *`set_drive set_driving_cell set_set_load set_fanout_load`*
- Most technology libraries have predefined sets of operating conditions. Use: *`read_lib report_lib`* commands to list the operating conditions defined in a technology.

9

Follow Step(4): Defining the Design Environment

- **Example**: *`set_operating_conditions WCCOM`* for setting the operating conditions to the worst case commercial.
- **Defining Wire Load Models:** Wire load modeling allows you to estimate the effect of wire length and fanout on the resistance, capacitance, and area of the nest.
- **Example**: *`set_wire_load "10x10"`* . The model *`"10x10"`* is defined in the vendor's library.

10

Step(5): Setting the Design Constraints

(1) Design Rule Constraints

- **(a)** (***Implicit* design rules specified in the technology library**). Specified by the vendor and must not be violated in order to get a proper functioning of the fabricated circuit.
- **(b) Stricter Design Rules:** maybe specified by the user (*explicit* design rules)

(2) Optimization Constraints: Define timing and area optimization goals for Design Compiler. These constraints are user-specified

- **Design Compiler optimizes the synthesis of the design, in accordance with these constraints, but not at the expense of the design rule constraints. In other words, Design Compiler never violates the higher-priority design rules.**

11

Follow Step(5): Setting the Design Constraints (Design Rule Constraints)

- **Setting Fanout Load Constraints**: The maximum fanout load for a net is the maximum number of loads the net can drive.
- **Example** `Set_max_fanout 16`
- **Setting Capacitance Constraints**
- **Example** `Set_max_capacitance 3`

12

Follow Step(5): Setting the Design Constraints (Optimization Constraints)

- **Timing Constraints and Area Constraints**. (*For Power constraints, the Synopsys Power Compiler is used*).
- **Timing Constriants**: specify the required performance of the design
- **Steps for setting the timing constraints:**
- (a) Define the clock
- (b) Specify I/O timing requirements relative to the clock
- (c) Specify the combinational path delay requirements
- (d) Specify the timing exceptions

13

Follow Step(5): Setting the Design Constraints (Optimization Constraints – Timing Constriants Commands)

- **Example:** `create_clock clk1 –period 40`

`create_clock`	Defines the period and the waveform for the clock	`set_max_delay` `set_min_delay`	Defines maximum delay for combinational paths

`set_clock_latency`
`set_propagated_clock`
`set_clock_uncertainty`
	Defines clock delay	`set_false_path`	Specifies false paths

`set_input_delay`	Defines timing requirements for input ports relative to the clock period	`set_multicycle_path`	Specifies multicycle paths
`set_output_dlay`	Defines timing requirements for output ports relative to the clock period	`report_clock`	Informs about all clock sources in the design

14

Follow Step(5): Setting the Design Constraints (Optimization Constraints – Timing Constriants)

- (b) **Specifying I/O timing requirements relative to the clock**. When the signal will arrive to the port relative to the clock.
- Use *report_port* Command to list all I/O delays associated with ports.
- (c) **Specifying Combinational Path Delay Requirements**. For purely combinational delays that are not bounded by a clock period.
- **Example** *max_delay 30.0 all_outputs()*

15

Follow Step(5): Setting the Design Constraints (Optimization Constraints – Area Constraints)

- **Specify the maximum area for the current design.** **The unit used is the same as the one in the technology library**
- **Example** *max_area 100*

16

Where Are We?

1) **Develop HDL Files** ✓
2) **Specify Libraries** ✓
3) **Read Design** ✓
4) **Define Design Environment** ✓
5) **Set Design Constraints** ✓
6) **Optimize the Design**
7) **Analyze and Resolve the Design Problems**

17

Step (6): Design OpTMiZaTiOn

- Optimization is the Design Compiler synthesis step that maps the design to an optimal combination of specific target library cells, based on the design's functional, speed and area requirements.
- Please go through chapter (8) in the Design Compiler User Guide for more information
- compile *-map_effort high area_effort high*

18

References

- Design Compiler User Guide by Synopsys, Version 2002.05
- Other Useful Links:
- **http://www.ecs.syr.edu/faculty/ercanli/cse664/Synopsys.HTM**
- **http://www-ece.engr.utk.edu/~sowmyan/synthesis.html**
- **http://www.altera.com/support/software/eda_maxplus2/synopsys/compilers/vsynt.html**

19