Report 1: Design Compiler Synthesis

Tai Duc Nguyen
ECEC 574: ASIC Design I

January 21, 2020

---

1 Introduction

In this assignment, the Design Compiler (DC) software is used to generate the transistor-level logic of a D-Flip-Flop with various design constraints (clock frequencies, clock latency, input/output delay, etc.). Then, the PrimeTime software will do the timing analysis of the compiled design. The following section will consist of the DC synthesis code written in TCL.

2 DC Synthesis Code

Line 15 of the following piece of code lists the 2 design files (written in verilog), which sets up the inner-working of the DFF (dff.v) using the transistor definitions in s15850.v. Later in this assignment, the same DFF design will be evaluated along with 4 other transistor definitions (s1238.v, s35932.v, s386.v, and s9234.v).

```
 1
 2 ###### DC Synthesis Script #######
 3
 4 ## Give the path to the verilog files and define the WORK directory
 5
 6 lappend search_path ../src_assignment/
 7 define_design_lib WORK -path "work"
 8
 9 ## Define the library location
10 set link_library [ list /mnt/class_data/ecec574-w2019/PDKs/SAED32nm/lib/stdcell_rvt/
       db_ccs/saed32rvt_ss0p95v125c.db /mnt/class_data/ecec574-w2019/PDKs/SAED32nm/lib/
       stdcell_rvt/db_ccs/saed32rvt_ss0p95v25c.db /mnt/class_data/ecec574-w2019/PDKs/
       SAED32nm/lib/stdcell_rvt/db_ccs/saed32rvt_ss0p95vn40c.db]
11
12 set target_library [ list /mnt/class_data/ecec574-w2019/PDKs/SAED32nm/lib/stdcell_rvt/
       db_ccs/saed32rvt_ss0p95v25c.db ]
13
14 ## read the verilog files
15 analyze -library WORK -format verilog [list dff.v s15850.v]
16
17 elaborate   -architecture verilog -library WORK dff
18
19 ## Check if design is consistent
20 check_design  > reports/synth_check_design.rpt
21
```

```
## Create Constraints
create_clock clk -name ideal_clock -period 10
set_clock_latency -source 0.4 [get_clocks ideal_clock]
set_clock_uncertainty 0.05 [get_clocks ideal_clock]
set_clock_transition 0.1 ideal_clock

set_input_delay 2.0 [ remove_from_collection [all_inputs] d ] -clock ideal_clock

set_output_delay 2.0 [all_outputs ] -clock ideal_clock
set_max_area 0
set_load 0.3 [ all_outputs ]


## Compilation
## you can change medium to either low or high
compile -area_effort medium -map_effort medium


## Below commands report area , cell, qor, resources, and timing information needed to analyze the design.

report_area > reports/synth_area.rpt
report_cell > reports/synth_cells.rpt
report_qor  > reports/synth_qor.rpt
report_resources > reports/synth_resources.rpt
report_timing -max_paths 10 > reports/synth_timing.rpt

## Dump out the constraints in an SDC file

write_sdc  const/dff.sdc

## Dump out the synthesized database and gate-level-netlist
write -f ddc -hierarchy -output output/dff.ddc

write -hierachy -format verilog -output output/dff.v

exit
```

3 DC Logic Synthesis Results at 100MHz

Design	Area	Power	WNS	TNS	Cell count	Num of violating paths
s15850.v	6.855995 ns^2	0.1432 uW μW	0.00	0.00	1	0