Computer-Aided VLSI System Design
DFT Compiler Lab: Insert Scan Chain

Objectives:

In this lab, you will learn:

How to insert scan chain into a synthesized gate level design

Download Files from ~cvsd/CUR/Testing/DFT

1. Create a work directory and copy the lab files into it.
2. check if you have these files:

Filename	Description
ALU_syn.v	Synthesized gate level Verilog netlist for the simple ALU (no scan chain yet)
.synopsys_dc.setup	Synopsys Design Compiler setup file, which defines search paths, library name, etc.
constraints.tcl	The constraints that are used in the synthesis lab.
dft.tcl	Reference script in this lab.

3. Check the contents of these files.

Environment Setup

source /usr/cad/synopsys/CIC/synthesis.csh

Invoke DftCompiler

Dft Compiler is actually embedded in the Design Compiler.

To invoke Dft Compiler, you can do either one

dc_shell (command mode)

dv & (GUI mode)

I encourage everybody to use command mode because:

a. command mode helps you to keep a record of what you have done.

b. command mode runs more efficiently than GUI mode.

c. command mode helps you to lookup the manual/reference quickly.

In spite of the above advantages, command mode sometimes is not as good as GUI mode in terms of debugging the schematic problem. We will use command mode throughout this Lab. You are welcome to try the GUI mode by yourself.

NOTE: maybe you see some error message like “Error: current design not defined.” just ignore it for now.

STEP 1: Read Input Files

1. Please check there is no error message when starting the "dc_shell". If there are errors in the windows, please check the .synopsys_dc.setup. Type either one of these lines to read your gate level netlist (The circuit after synthesis).
   read_verilog ALU_syn.v
   read_file ALU_syn.v -format verilog

2. Set the working design to you top design. In this case, set ALU as the working design.
   current_design ALU

3. Resolve the design references and check if there is any errors.
   link
   check_design

4. Set the design constraints and check if the designs have any violations. The constraints.tcl is based on the constraints that you used in the synthesis lab.
   source constraints.tcl
   report_constraint -all_violators

5. To obtain a timing/area/power report of your original design, type (where ALU is your top design)
   report_area > ALU_syn.area_rpt
   report_timing > ALU_syn.timing_rpt
   report_power > ALU_syn.power_rpt

STEP 2: Select scan style

Define the default scan style for the insert_dft command if a scan style is not specified with the set_scan_style command. This variable must identify one of the following supported scan styles: multiplexed_flip_flop, clocked_scan, lssd, aux_clock_lssd, combinational, or none. You can skip this step because the default is multiplexed_flip_flop.

set test_default_scan_style multiplexed_flip_flop

STEP 3: Set ATE configuration and create test protocol

The timing of the test clock is based on the test_default_period, test_default_delay, test_default_strobe, and test_default_strobe_width variables.

```
set test_default_delay 0
set test_default_bidir_delay 0
set test_default_strobe 40
set test_default_period 100
```

To create a test protocol for a non-scan design, you can just type

```
create_test_protocol -infer_asynch -infer_clock
```

When -infer_asynch is specified, create_test_protocol infers asynchronous set and reset signals in the design, and places them at off state during scan shifting. When -infer_clock is specified, create_test_protocol infers test clock pins from the design, and pulses them during scan shifting.

(Optional) In this lab, DFT compiler can identify your clock and asynchronous reset automatically. Instead of automatic identification, you can also specify these signals by the set_dft_signal command. For example:

```
set_dft_signal -view existing_dft -type ScanClock -port clk \
    -timing [list 45 55]
set_dft_signal -view existing_dft -type reset -port reset \
    -active_state 0
create_test_protocol
```

STEP 4: Pre-scan Check

Check if there is any design constraint violations before scan insertion.

```
report_constraint -all_violators
```

Perform pre-scan test design rule checking.

```
dft_drc
```

Question 1: How many scan cells do you have? ________

Question 2: How many rule violation do you have? _______

Note: If there were violations, you should stop to fix your code.

STEP 5: Scan specification

This step tells the Dft Compiler how many scan chains you want. You can also specify the names of scan related pins (scan_enable, scan_in, scan_out). We will let Dft Compiler to choose the pin names for us.

set_scan_configuration -chain_count 1

STEP 6: Scan preview

This step checks your scan specification for consistency. Please type

preview_dft

Question 1: How many scan chains will you have? _____

Question 2: What are their pin names? __________________

STEP 7: scan chain synthesis

Stitch your scan cells into a chain. And do some more optimizations.

insert_dft

STEP8: Post-scan check

Check if there is any design constraint violations after scan insertion.

report_constraint -all_violators

Perform post-scan test design rule checking.

dft_drc

STEP 9: Reports

Report the scan cells and the scan paths

report_scan_path -view existing -chain all > ALU_syn_dft.scan_path
report_scan_path -view existing -cell all > ALU_syn_dft.scan_cell

To obtain a timing/area report of your scan_inserted design, type

report_area > ALU_syn_dft.area_rpt
report_timing > ALU_syn_dft.timing_rpt
report_power > ALU_syn_dft.power_rpt

Examine the report files of our scan_inserted design. Compare these reports with those of the non-scan design.

Question 1: How much is the area overhead of scan? ______%

Question 2: How much is the timing overhead of scan? ______%

STEP 10: Write out files

Write test protocol for later usage in the ATPG lab.

write_test_protocol -output ALU_syn_dft.spf

To output our scan-inserted netlist, type

write -hierarchy -format verilog -output ALU_syn_dft.v

write -hierarchy -format ddc -output ALU_syn_dft.ddc

The sdf (standard delay format) file is for timing analysis. Our next tool, Primetime, will need this file.

write_sdf -version 2.1 -context verilog ALU_syn_dft.sdf

Questions/Comments: Check your Verilog netlist.

1. What type of DFF before dft_compiler? ___________
2. What type of DFF after dft_compiler? _________
3. Which two pins did dft_compiler add? _________ and _________
4. Where is scan ouput pin? _________

dft_compiler share the scan_output pin with the functional output pin, so there is NO dedicated scan output pin. This is done for saving the number of pins. If you want dft_compiler to create a dedicate scan out pin, use this command before you insert the scan chain.

set_scan_configuration -create_dedicated_scan_out_ports true

We do not do this in this lab.

Checkpoints:

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Show your DFT results without any DRC violations.
2. Answer the questions in this lab.

END of LAB

Creator:
1st Edition: Chien-Mo Li, 2002
2nd Edition: Yu-Lin Chang, 2004
3rd Edition: Jui-Hsin Lai(Larry), 2008
4th Edition: Bing-Chuan Bai, 2010
5th Edition: Bing-Chuan Bai, 2011