Lab N° 3

Design for Testability

Objectives

- Realise the full synthesis flow
  - Start from an existing design
  - Use the synthesis tool « Design Compiler » from eSynopsys
  - Use the test tool test « Tetramax » from Synopsys

Design

- A 8-bit filter
  - Composed of an adder and two 8-bit register
  - Source fille in the directory
    - … /VHDL

Synthesis

- Tool **dc_shell-xg-t** (Design Compiler from Synopsys)
    - Necessary scripts are available in the directory
        - … /Synth
            - .synopsys_dc.setup
            - synthese_dc_shell.scr

Test

- Use Tetramax to generate test patterns
  - Comment and analyse the obtained results in terms of testable and non-testable faults
  - Comparer le TC avec et sans SCAN

DfT

- Modify the circuit by adding physically the scan path
  - Use the tool **dc_shell-xg-t**

Commands

- Read the file
  - `read_verilog <ckt.v>`
- Specify the top module
  - `current_design <top module name>`

Commands

- Specify the type of scan chain
  - FF type
    - set test_default_scan_style multiplexed_flip_flop
  - Number of scan chain
    - set_scan_configuration -chain_count <n>

Commands

- Add test input/output:
  - Test_mode
  - Test_si
  - Test_so
  - Test_se

  - Create_port <name> -direction <in | out | inout>

Commands

- Specify the signals that are controlling the scan chain:
  - Clock
  - Set|reset
  - Test_mode
  - Test_si
  - Test_so
  - Test_se

Commands

- set_dft_signal -view exist -type ScanClock -timing {45 55} -port < clk >
- set_dft_signal -view exist -type Reset -active_state 0 –port < reset >
- set_dft_signal -view spec -type Constant -active_state 1 -port < test mode >
- set_dft_signal -view spec -type ScanDataIn -port < test si >
- set_dft_signal -view spec -type ScanDataOut -port < test so >
- set_dft_signal -view spec -type ScanEnable –port < test se >

Commands

- Create the test protocol
  - create_test_protocol
- Add the scan chain
  - dft_drc
  - preview_dft
  - insert_dft
- Visualize the scan chain
  - report_scan_path

Commands

- Save the test protocol
  - `write_test_protocol -o < file.spf >`
- Save the modify circuit
  - `write -format verilog -hierarchy –output < ckt.v >`

Test

- Use Tetramax to check if the test protocol and the circuit with the scan chain are correct
- Generate test vectors and save them