TDDC33
Design for Test of Digital Systems
Lesson 1

Introduction to the LAB series

Zhiyuan He
Outline

- Organization
- LAB assignments
  - The scenario
  - Lab1: Test pattern generation
  - Lab2: Design for test
  - Lab3: Board test using boundary scan
- Tools
  - Leonardo spectrum
  - FlexTest
  - DFTAdvisor
  - Trainer1149
- Questions
• Contact information
  – Zhiyuan He
  – Email: zhihe@ida.liu.se start subject line with “TDDC33:”
  – Homepage: http://www.ida.liu.se/~zhihe/
  – Office: B 329:204
Organization

- Course homepage
  - http://www.ida.liu.se/~TDDC33/

- Lab web-pages

- Register in WebReg
  - https://www.ida.liu.se/webreg/TDDC33-2008/LAB1

- The labs are a mandatory part of the course and have to be solved individually.

- The lab assignments will to a large extent be solved at times not scheduled
Important dates

- Registration deadline: **2008-09-19**

- Deadlines for submitting lab reports:
  - Lab 1: **2008-09-12**
  - Lab 2: **2008-09-26**
  - Lab 3: **2008-10-10**

- Updates to returned lab reports must be handed in within **7 days** after receiving the notification emails

- Preliminary exam (dugga)
  - **2008-09-25, 13-15** (in the lab session)
What do you get for doing the labs?

- All points you gain will be counted into your mark of the final examination
- 0-1 point for each lab
- 0-7 point for the preliminary exam (dugga)
- Up to 40 points for the final exam (30 points for written exam + 10 points for labs)
Organization

- Examination of labs
  - Preliminary exam ("dugga") ~20 minutes
  - Oral presentation (prepare to answer questions and to show that you can handle the tools)
  - Written report (one for each lab) containing the results from the lab.
  - Use the Laboration report covers found by the printers.

- Personal experience of the lab. **Not graded.** Comments on the...
  - level of difficulty,
  - instructions, tools, effort and time, etc.
Outline

- Organization
- LAB assignments
  - The scenario and instructions
  - Lab1: Test pattern generation
  - Lab2: Design for test
  - Lab3: Board test using boundary scan
- Tools
  - Leonardo spectrum
  - FlexTest
  - DFTAdvisor
  - Trainer1149
- Questions
The board of directors at the company codenamed TDDC33 finally has realized the need of DFT. During their last meeting, they decided to use an external DFT consultant for the evaluation of different test approaches. They also want to take the opportunity to put forward some of the questions that they have regarding DFT. The questions that they have may be formulated like the following;

- Test pattern generation - Is it possible to write test patterns manually? Is the required effort the same for combinational and sequential circuits? What is the achieved fault coverage?
- DFT - How difficult is it to insert test points? What is the cost? How is boundary scan implemented? What test patterns are required? What type of faults can be detected using boundary scan?
Lab1 Test pattern generation

- **Objective**
  - To get experience and knowledge about test pattern generation for combinatorial and sequential circuits.

- **Input**
  - Two designs c17 and s27 described in VHDL

- **Assignment 1: Manual test pattern generation**
  - Number of test patterns used
  - Achieved fault coverage
  - Test patterns

- **Assignment 2: Automatic test pattern generation**
Objective
- To get experience and knowledge about different design for test techniques.

Input
- One sequential design s27 described in VHDL

Assignment 1: Manual test point insertion
- Number of test points added
- Achieved fault coverage before adding test points
- Achieved fault coverage after adding test points

Assignment 2: Automatic scan chain insertion
Lab3 Board Testing using Boundary Scan

- **Objective**
  - To get experience and knowledge about board testing using boundary scan.

- **Input**
  - A board design called TDDC33 consisting of two chips, c17 and s27.

- **Assignment 1: Design modification for Boundary Scan**
  - Draw a new design of the board where the Boundary Scan interface (TDI, TDO, TMS, and TCK) is added.

- **Assignment 2: Interconnect test**
  - Write an interconnect program that detects at least one fault for each of the following four types of faults: Stuck-at 1, Stuck-at 0, Wired-AND short, and Wired-OR short.
  - Verify the test program by introducing faults in the design.
Outline

- Organization
- LAB assignments
  - The scenario
  - Lab1: Test pattern generation
  - Lab2: Design for test
  - Lab3: Board test using boundary scan
- Tools
  - Leonardo spectrum
  - FlexTest
  - DFTAdvisor
  - Trainer1149
- Questions
The following information is also described in the Instruction.pdf available on the course web-page.

Core cells from AMS 0.35µm standard cell library (c35_CORELIB) will be used together with the test library c35_CORELIB.atpg

Tools and tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Tool</th>
<th>Tool vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesis</td>
<td>Leonardo Spectrum</td>
<td>(Mentor Graphics)</td>
</tr>
<tr>
<td>Design for test</td>
<td>DFTAdvisor</td>
<td>(Mentor Graphics)</td>
</tr>
<tr>
<td>Test pattern generation</td>
<td>FlexTest</td>
<td>(Mentor Graphics)</td>
</tr>
<tr>
<td>Boundary scan</td>
<td>Trainer1149</td>
<td>(Testonica Lab)</td>
</tr>
</tbody>
</table>
- **Initial Preparations**
  - Add the following modules
    
    ```
    module add prog/mentor/dft_2005
    module add prog/mentor/fpgadv
    ```
  - **Files**
    - `c35_CORELIB.atpg (Test library)`
    - `fflop.vhd (Description of a flip-flop)`
    - `gates.vhd (Description of the gates)`
    - `s27.vhdl (The design file)`
  - Set the environment variable `MODEL_TECH`
    
    ```
    setenv MODEL_TECH /sw/mentor/fpgadv/6.2/Modeltech/bin
    ```
  - Generate the work directory
    
    ```
    $MODEL_TECH/vlib work
    ```
  - Compile the vhdl files
    
    ```
    $MODEL_TECH/vcom -93 fflop.vhd
    $MODEL_TECH/vcom -93 gates.vhd
    $MODEL_TECH/vcom s27.vhdl
    ```
Leonardo Spectrum
- Used to synthesize a compiled design described in VHDL to a netlist (EDIF format)
- Start Leonardo Spectrum
  `leonardo &`
- Specify the input files (`library c35_CORELIB`, `design.vhdl`)
- Specify the output file (`design.edf`)
DFTAdvisor
- Used to insert scan-chains in the design
- Start DFTAdvisor
  `dftadvisor &`
- Specify the input files (library c35_CORELIB, design.edf)
- Save the new scanable design (design_scan.edf)
FlexTest
- Used both for fault simulation and test pattern generation.
- Start DFTAdvisor
  `flextest &`
- Specify the input files (library c35_CORELIB, design.edf)
- Manually write test patterns
Manually write test patterns, example

- Combinational design

```
SETUP =
  TEST_CYCLE_WIDTH = 1;
  DECLARE INPUT BUS "ibus" = "/INP(0)", "/INP(1)", "/INP(2)", "/INP(3)", "/INP(4)";
  DECLARE OUTPUT BUS "obus" = "/OUTP(0)", "/OUTP(1)";
END;
CYCLE_TEST =
  PATTERN = 0;
  CYCLE = 0;
  FORCE "ibus" "01100" 0;
  MEASURE "obus" "11" 1;
  PATTERN = 1;
  CYCLE = 0;
  FORCE "ibus" "10010" 0;
  MEASURE "obus" "00" 1;
END;
```
Test point insertion in VHDL
- We solve this task manually by modifying the VHDL file
  - Add ports in the ENTITY block
  - Add signals in the ARCHITECTURE block (if needed)
  - Modify the design such that the new ports is used to control and observe the “hard-to-test” parts of the design.
- Compile and synthesis the modified VHDL file
- Use automatic test pattern generation and fault simulation to observe the improvement in testability.
■ Trainer1149
  - Used for training and verifying test programs for board testing.
  - Verify that the current installed version of java is 1.6 or higher
    
    \texttt{java \textasciitilde version}
    
    \texttt{module add prog/jdk/1.6}
  
  - Start the \textit{Trainer1149} program
    
    \texttt{java \textasciitilde jar trainer1149.java}
  
  - Use the tool to write and verify an interconnect test program
Organization

LAB assignments
- The scenario
- Lab1: Test pattern generation
- Lab2: Design for test
- Lab3: Board test using boundary scan

Tools
- Leonardo spectrum
- FlexTest
- DFTAdvisor
- Trainer1149

Questions