VDF Assignment-2

Objective

The objective of this assignment is to gain hands-on experience on writing Verilog model, simulation, computing code-coverage, and logic synthesis

Instructions

- This assignment will be done individually
- Each student will submit only one PDF file containing all the answers [no ZIP file or other format accepted.]
- The focus should not only be on running the tool, but also on analysis/interpretation of the results.
- **Copying/Plagiarism is strictly prohibited. Institute copying/plagiarism policy will apply with no exception.**

Choice of Tools

- You can take the help of the tutorials shown in the following NPTEL videos in this assignment:

  https://www.youtube.com/watch?v=9Wzz--APeLU

  https://www.youtube.com/watch?v=c-cFxuH-HbE

  https://www.youtube.com/watch?v=Phcq_iDo3ss

- You can use any open-source tool or commercial tools available at IIITD (of Synospys, Cadence, etc.) in this assignment

- You can use the opensource tools [ICARUS for Verilog Simulation, COVERED for computing coverage, and YOSYS for synthesis].

FOR SIMULATION AND VIEWING WAVEFORM

Name of the tool: ICARUS Verilog
Where and how to download?
https://github.com/steveicarus/iverilog
download the GitHub project
How to install?
https://github.com/steveicarus/iverilog

./configure
make
sh autoconf.sh
make check
sudo make install

Note: look into the **README.txt (in the Github repository)** for more information

How to run?
1. open the terminal
2. iverilog -Wall -o test **fsm.v testbench.v (verilog files)**
3. vvp test
4. gtkwave **fsm.vcd (vcd file)**

Note: make sure gtkwave is already installed
sudo apt get update
sudo apt get install gtkwave

How to get more information?

WEBSITE: http://iverilog.icarus.com/

GITHUB: https://github.com/steveicarus/iverilog

MAN PAGE: https://linux.die.net/man/1/iverilog

USAGE MANUAL:
http://courses.cecs.anu.edu.au/courses/ENGN3213/Documents/VERILOG/iverilog_NOOR.pdf

MISCELLANEOUS: http://inf-server.inf.uth.gr/~konstadel/resources/Icarus_Verilog_GTKWave_guide.pdf
http://gtkwave.sourceforge.net/

FOR LINE COVERAGE:

Name of the tool: COVERED Verilog Code Coverage Analyzer
Where and how to download?
download the Github project
https://github.com/chiphackers/covered

How to install?
1. unzip

2. open the terminal inside the unzipped folder
3. ./configure
4. make
5. sudo make install

Note: look into the **INSTALL file (in the github repository)** for more information

How to run?
command:

1. **covered score -t** tb_fsm **-v** tb_fsm.v **-v** fsm.v **-vcd** fsm.vcd **-o** fsm.cdd

-t top-level module
-v filename
Name of specific Verilog file to score.
-vcd filename
Name of VCD dumpfile to score design with
-o database
Name of database to write coverage information to

2. **covered report -d v** fsm.cdd

to obtain the module-based verbose report

How to get more information?

Website: http://covered.sourceforge.net/

Github: https://github.com/chiphackers/covered

Manpage: https://linux.die.net/man/1/covered

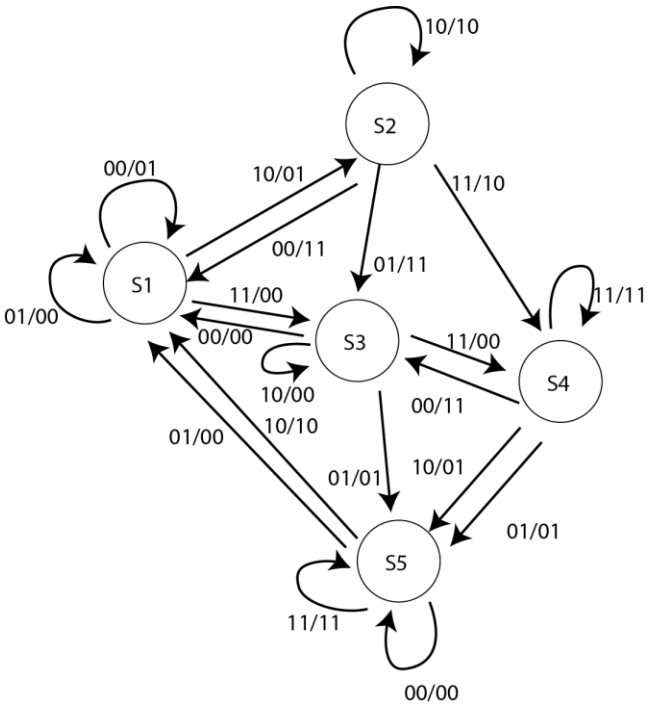
Name of the tool: YOSYS
Where and how to download?
GitHub: https://github.com/The-OpenROAD-Project/yosy
download the github project
How to install?
Follow the below link for detailed instruction on how to install Yosys (+ ABC)
GitHub: https://github.com/The-OpenROAD-Project/yosys
How to run?
cmd: yosys
//to launch the tool

Questions

1. Show a screenshot or snippet of the log file to prove that you have indeed run the tool **[If this is not given then your assignment will not be evaluated].**

2. Consider the state machine shown below.

(a) Write a Verilog model for the following state machine. Use binary encoding for the states. **[2 marks]**



The states are: S1, S2, S3, S4, and S5.

The input values are {00, 01, 10, 11}.

Output values are: {00, 01, 10, 11}.

The initial state is S1 (when *rst* is made 1).

(b) Write the testbench to test the Verilog model. **[3 marks]**

(c) Show the simulation output waveform and explain the correctness of the model. **[2 marks]**

(d) Show line coverage report for the above testbench. Explain the report. **[3 marks]**

(e) Synthesize the code written above using YOSYS. Use any technology library. Report the synthesized netlist and explain the result. **[4 marks]**

(f) Re-write the Verilog code using one-hot encoding. Synthesize this netlist using YOSYS. Use the same technology library as above. Report the synthesized netlist, compare the results with that obtained in the previous question, and explain your observations. **[6 marks]**