Indian Institute of Technology Roorkee
Department of Computer Science and Engineering
CSN-221: Computer Architecture and Microprocessors (Autumn 2016-2017)

(Instructor: Dr. Sudip Roy)
Date: Oct 24, 2016

Coding Project-1 (CP-1) Duration: 5 days

Instructions:

1. At most TWO students can make a group and perform each Lab assignment.
2. Objective of this Lab assignment: To make the student familiar with the programming language Verilog (modular programming in hardware description language or HDL) through implementation of some building blocks inside an ALU (arithmetic and logic unit). In this project, the student will learn how to write the Verilog code of a digital logic block, how to write the test-bench Verilog code to test his/her design (Verilog code of a digital logic block) and how to simulate the design in a software tool for simulation (called Xilinx ISE WebPACK Design Software) by checking the output for a given set of test input values.

- Problem Statements in CP-1:

1. Write the Verilog code (<alu8bit>.v) of an 8-bit ALU that can execute following four different operations:
   a. Addition
   b. Multiplication
   c. AND
   d. XOR

   As shown in Fig.1, A and B are the two 8-bit signed integers and input operand lines to the ALU, in which any one of the aforementioned operations is to be performed. The MSB (Most Significant Bit) of input integer indicates if the input number is positive or negative. If MSB = 0; then given number is positive, else negative. Opcode is the control line to the ALU that tells about which operation (+/*/AND/XOR) to be performed by the ALU and it comes from the instruction (binary bit string). Y is the output result obtained for the operation A Opcode B.

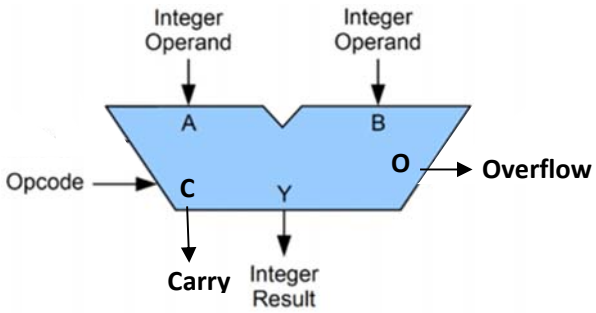

Fig.1: Arithmetic and Logic Unit (ALU)

In order to execute some instructions (format described in the next item), you need to design a register file with four registers each of 8-bit size.

Write three different Verilog modules adder8bit (8-bit full adder), mult8bit (8-bit multiplier), and8bit, xor8bit for addition, multiplication, AND and XOR, respectively. The 8-bit adder (8-bit RCA) is to be designed using eight 1-bit full-adders in a ripple-carry fashion and the 8-bit multiplier is to be designed following the Booth's multiplication algorithm (hence one input register is to be used for storing the product in the register-pair). C is the carry bit for two 8-bit integer addition.

Note: For the arithmetic operations check for the Overflow condition. Overflow during addition can occur, only when both input numbers are of same sign. If there is overflow after addition, then all output bits should be zero's while O bit should be set ON (1).

2. Write the test-bench Verilog code (<designfile_tb>.v) (one test-bench file for the top-module only) for your design under test (DUT). The test- bench will have instructions in the format as shown in Fig.2, where an instruction is of 26-bit binary bit string.

   The two MSBs are OpCode bits, which decide the operation to be performed on given A and B. Opcode may be of any one of the four bit-strings: 00 – Addition, 01 – ADD, 10 – XOR, 11 – Multiplication. Bits from [23 -16] and [15 - 8] are the two input operands (signed), where 23rd and 15th bits indicate whether the number is positive or negative. The eight LSBs [7 - 0] are the result (Y) of 8-bits for addition, ADD and XOR operation. For multiplication operation result should be stored in a register-pair.

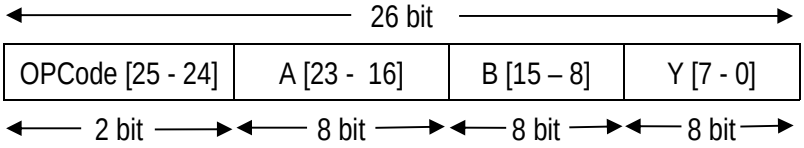

Fig.2: Input Instruction Format

3. Simulate your code in Xilinx ISE Design Suite tool and verify/check whether your design is giving correct output waveforms for the known instructions taken as input in the test-bench file.

- Instructions for Xilinx ISE Design Suite Tool:

  Software Tool Name: Xilinx ISE Design Suite 14.7 (WebPack) Download
  https://survey.xilinx.com/ss/wsb.dll/Xilinx/ISE_Download_Survey.htm?wsb5=14.7&wsb6=1&wsb7=Xilinx_ISE_DS_Win_14.7_1015_1.tar
  For downloading the software (6.17GB), you can choose the following in the link opened:
  - Student – ISE required for classes
  - WebPACK
  - Spartan-3E/-3A/-3AN
  - This is the first time

  Just follow the steps to install the WebPACK from this design suite. There is no need to download license file for the use of this software, as you will only simulate your Verilog code (no need to synthesis the design).

- Tasks to do:

  1. Write the Verilog code (<designfile>.v) of your design problem (if you have multiple such files, include all).
  2. Write the test-bench Verilog code (<designfile_tb>.v) (one test-bench file for the top-module only) for your design problem.
  3. The <designfile> depends on the problem you are working on. Your codes should have self-explanatory names to variables or modules you used.
  4. While writing a Verilog code, you MUST INCLUDE the following information (for example) in the .v file before coding the actual program:

     ## GroupID-1 (15114XXX_15114YYY) - Name1 & Name2
     ## Date: October 24, 2016
     ## <designfile>.v - State which digital block is coded in this file.
     …

Here you start writing your code.
…

5. Simulate your code in Xilinx ISE Design Suite tool and verify/check whether your design is giving correct output waveforms for the known input waveforms.
6. Take a screenshot (by PrintScreen key) of the timing diagrams (waveforms) obtained from the Xilinx ISE Design Suite tool for some test-cases.

- Submission Method (Strictly Follow These):

1. You must submit a zipped folder (<filename>.zip or <filename>.tar.gz) containing the following items:
   (a) The Verilog code file <designfile>.v (if you have multiple such files, include all).
   (b) The test-bench Verilog code file <designfile_tb>.v (one test-bench file for the top-module only).
   (c) The report file (<filename>.DOC or <filename>.PDF) should contain your details like Group-ID, Name(s) and Enrollment Number(s) of the Member(s). The report should contain the following:
      i. The basic description (in brief) of your design of the digital logic blocks for the problem you have been assigned.
      ii. The Verilog code (<designfile>.v) of your design problem (if you have multiple such files, include all) and the test-bench Verilog code (<designfile_tb>.v) (one test-bench file for the top-module only).
      iii. The screenshots (taken by PrintScreen key) of the timing diagrams (waveforms) showing the simulation of your design for verification of the output based on the input to your design.
2. Very Important: If the enrollment numbers of two members of your group are 15114XXX and 15114YYY, then replace <filename> with "15114XXX_15114YYY". Strictly follow this convention of you filename while submitting.
3. Submit your zipped folder (<filename>.zip or <filename>.tar.gz) through your account in Moodle. We have created a submission link in Moodle course site: http://moodle.iitr.ac.in/mod/assign/view.php?id=99
4. Hard deadline for Final submission in Moodle: October 29, 2016 (10:00 pm Indian Time).
5. For any submission after Final Deadline, 5 marks will be deducted for every 24 hours of extra time.
6. The key to success is starting early. You can always take a break, if you finish early.

- Evaluation Process:

1. We will simulate your codes contained in your zipped folder and will regenerate the results you had included in the report file.
2. Your submission will be checked with others' submission to identify any copy case. If we detect that your code is a copy (partially or fully) of other's code, then the total marks obtained by one group will be divided by the number of groups sharing the same Verilog code.
3. You may be asked to demonstrate and explain your submission after the final submission deadline.

References:

[1] David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, (Forth Edition), Morgan Kaufmann Publishers, 2008.
[2] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition, Prentice Hall Press, 2003.
[3] J. Bhaskar, Verilog HDL Synthesis- A Practical Primer, B.S. Publications, 2001.
[4] Video Lectures on Verilog Programming by Prof. Indranil Sengupta, Dept. of Computer Science and Engineering, IIT Kharagpur (follow first seven lectures), web: http://nptel.ac.in/courses/106105083/.
