DIGITAL DESIGN

ASSIGNMENT 1

Deadline: *22:30, Wednesday 14 October 2020*

Lab sessions&Location:

1. **Lychee Garden 6, Room 402 (Wednesday 14:00-15:50 pm)**
2. **Lychee Garden 6, Room 402 (Wednesday 16:20-18:10 pm)**
3. **Lychee Garden 6, Room 402 (Thursday 8:00-9:50 am)**
4. **Lychee Garden 6, Room 402 (Thursday 10:10~12:10 am)**

Teaching Assistant:

Wang Wei, email: wangw6@sustech.edu.cn

Hua ZhengChang, email: huazc@mail.sustech.edu.cn

PART 1: DIGITAL DESIGN THEORY

Provide answers to the following questions:

1. What is the exact number of bytes in a system that contains (a) 64K bytes, (b) 128M bytes, and (c) 6.4G bytes?
2. What is the largest binary number that can be expressed with 14 bits? What are the equivalent decimal and hexadecimal numbers?
3. Convert the decimal number 184 to binary in two ways: (a) convert directly to binary; (b) convert first to hexadecimal and then from hexadecimal to binary. Which method is faster?
4. Give the complement code of the following **decimal** numbers.
   a. 27904836
   b. 63325006
5. (a) Find the 16’s complement of C6EF.

   (b) Convert C6EF to binary.

   (c) Find the 2’s complement of the result in (b).

   (d) Convert the answer in (c) to hexadecimal, compare with the answer in (a).
6. Do the following conversion problems:

   (a) Convert decimal 19.625 to binary.

   (b) Calculate the binary equivalent of 4/3 out to eight places. Then convert from binary to decimal. How close is the result to 4/3?

   (c) Convert the binary result in (b) into hexadecimal. Then convert the result to decimal. Is the answer the same? Explain why.
7. Represent the decimal number 6503 in (a) BCD, (b) excess-3 code, (c) 8,4,-2,-1 code, and (d) 6311 code.
8. We can perform logical operations on strings of bits by considering each pair of corresponding bits separately (called **bitwise operation**). Given two eight-bit

strings A = 10100101 and B = 00011010, evaluate the eight-bit result after the following logical operations: (a) AND (b) OR (c) XOR (d) NOT A (e) NOT B (f) NAND (g) NOR **(represent the results in hexadecimal)**

PART 2: DIGITAL DESIGN LAB

INTRODUCTION

In this lab, you are required to use Vivado 2017.4 and Minisys Practice platform (xilinx FPGA chip artix 7 inside) to design a simple logic circuit: Do the addition on two signed 2bit numbers, do the simulation and verify its function by simulation. You should submit the description of the operation steps, the Verilog design, the wave form from the simulation, and the function verification results.

PREAMBLE

Before working on the coursework itself, you should master the following material. A separate tutorial document (on the Sakai site) has be provided to you which includes:

- Vivado: The Vivado software provides a complete design environment for system-on-a-programmable-chip (SOPC) design. Regardless of whether you use a personal computer or a Linux workstation, Vivado ensures easy design entry, fast processing, and straightforward device programming.
- Minisys Practice platform: a platform designed for Digital design ,Principles of Computer Organization and many other courses. This platform include FPGA chip ,storage chip and lots of Dial switches, LEDs for input and output.
- Verilog : standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems. It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits and mixed-signal circuits, as well as in the design of genetic circuits. http://www.verilog.com/

EXERCISE SPECIFICATION

In practical part, you will focus on how to use Vivado to do the design, simulation, generate the bitstream file which is used to program the FPGA chip (a part of the Minisys practice platform), you will also learn some basic concepts of Verilog. The steps you need to follow are:

1. Create an empty project.
2. Edit a design file (Verilog file) & add it to the project.
3. Edit a simulation file (Verilog file) & add it to the project.
4. Do the simulation using to verify if the function of design is ok. If not, modify your design and do the simulation again.
5. Do the synthesize.
6. Edit a constraints file (to define the Specifications of pins and the binding info between pins and the designed ports) & add it to the project.
7. Do the implementation.
8. Generate the bitstream file.
9. Turn on the board (Minisys & FPGA chip inside) and connect Vivado to the board
10. Program the device with the bitstream
11. Test the design on the board (using dial switch as input, using led to see the state of output)

In task1, you should focus on step 1~4. in task 2, you should focus on the step 1~11.

TASK1:

Create a project named as Lab2_Signed_Addition, design the source code to implement following function: get the addition of two input numbers, and output the sum. Both two input value ares 2bits, treated as signed number. Do the simulation and verify the function of this circuit design.

Note

(1): two input number here should be treated as singed number, which means if the number is $(11)_2$, the value is $(-1)_{10}$ if it’s a singed number. while it’s a unsigned number, the value is $(3)_{10}$.

Tips: Penultimate page of “Digital design lab2.pdf ” has shown you how to define a signed input. “Digital design lab2.pdf ” could be got from ”sources/lab/lab2/” of Sakai site “CS207-Prof.G”.

(2): there should be two inputs and three outputs (two of them are used to demonstrate the value of two operands, the 3rd one is used to demonstrate the value of sum) //following module is just a simplified demo.

```
module Lab2_Addition(addend, augend, addend_led, augend_led, sum_led);
input   addend;
Input   augend;
output   addend_led;
output   augend_led;
output   sum_led;
......
Endmodule
```

TASK2:

NOTIC：if you have completed the task2 in the classroom inspection, there is no need to repeat it on your report. record the truth table of (x+y)’ ,x’y’,(xy)’ and x’+y’ on the report is enough.

1) Do the design using data flow, block style and structured style respectively (While doing the design with structured style, It is optional to use primitive or encapsulated IP) to verify the following theorem (you can find the design on the lab3 and lab4 courseware as a reference):

DeMorgan: a) **(x+y)’ = x’y’** b) **(xy)’ = x’+y’**

2) Create a test bench, do the simulation to verify the function of the design.

3) Edit a constraints file (to define the Specifications of pins and the binding info between pins and the designed ports) & add it to the project.

4) Do the synthesis and implementation.

5) Generate bitstream file, program the device with the bitstream

6) Test the design on the board (using dial switch as input, using led to see the state of output).

TIPS:

1) Putting all the circuit ((x+y)’ , x’y’ ,(xy)’ and x’+y’) to one design file, and naming ports in different design file(corresponding to the different design style) with same name so that you can reuse the testbench file and constraint file with just a little modification (as in labs courseware, just modify the module name in your design file).

2) Naming the file:

a. For data flow design, the source file should be demorgan_df.v

b. For structured design, the source file should be demorgan_sd.v

3) All the 2 design could share the same test bench file because the number and name of ports are same while only name of module are different. If you want use test bench file of demorgan_df.v on demorgan_bd.bd ,just change the module name from demorgan_df to demorgan_bd is enough.

4) All the 2 design could also share the same constrain file because the number and name of ports are the same.

SUBMISSION

Submit your assignment report to the Sakai on *Corresponding site* by the deadline.

“CS207-Prof.G”

ASSESSEMENT

The full marks for this exercise is 100 and they are distributed as follows:

Theory: 35%

Question 1	3

Question 2	2
Question 3	2
Question 4	2
Question 5	2+1+2+1
Question 6	1+2+2
Question 7	8
Question 8	7
Total	35 marks

Lab: 65%

Task 1: Design in Verilog, the truth-table	5*2 marks
Task 1: Test bench in Verilog, simulation result	5*2 marks
Task 2: Design with data-flow style, design with structure-design style, the truth-table	5*3 marks
Task 2: Test bench in Verilog, simulation result	5*2 marks
Task 2: Constrains file and the description of the test result on Minisys practice board	5*2 marks
Problems and solutions	5*2 marks
Total	65 marks

The template for the report is provided in the next pages.

DIGITAL DESIGN

ASSIGNMENT REPORT

ASSIGNMENT ID : xxxx

Student Name: xxxx

Student ID: xxxx

PART 1: DIGITAL DESIGN THEORY

Provide your answers here:

PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

Describe the design of your system by providing the following information:

- *Verilog design (provide the Verilog code)*
- *Truth-table*

SIMULATION

Describe how you build the test bench and do the simulation.

- *Using Verilog(provide the Verilog code)*
- *Wave form of simulation result (provide screen shots)*
- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation.*

THE DESCRIPTION OF OPERATION

Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*

PART 2: DIGITAL DESIGN LAB (TASK2)

DESIGN

Describe the design of your system by providing the following information:

- *Verilog design while using data flow (provide the Verilog code)*
- *Verilog design while using structured design (provide the Verilog code)*

- *Truth-table*

SIMULATION

Describe how you build the test bench and do the simulation.

- *Using Verilog (provide the Verilog code)*
- *Wave form of simulation result (provide screen shots)*
- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation*

CONSTRAINT FILE AND THE TESTING

Describe how you test your design on the Minisys Practice platform.

- *Constraint file (provide the screen shots on the feature of a pin and the binding info between pins and the input /output ports)*
- *The testing result (provide the screen shots (at least 3 testing scene)) to show state of inputs and outputs along with the related descriptions.*

THE DESCRIPTION OF OPERATION

Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*