109-2 Digital System Design Homework #2

TA Information

馬咏治、羅翊誠

kane@access.ee.ntu.edu.tw; alan@access.ee.ntu.edu.tw

Problem 1. 8-bit arithmetic logic unit (ALU) (40%)

In problem 1, you have to design an 8-bit arithmetic logic unit (ALU). The input and output ports are defined in Figure 1. The functions of ALU are defined in Table 1.

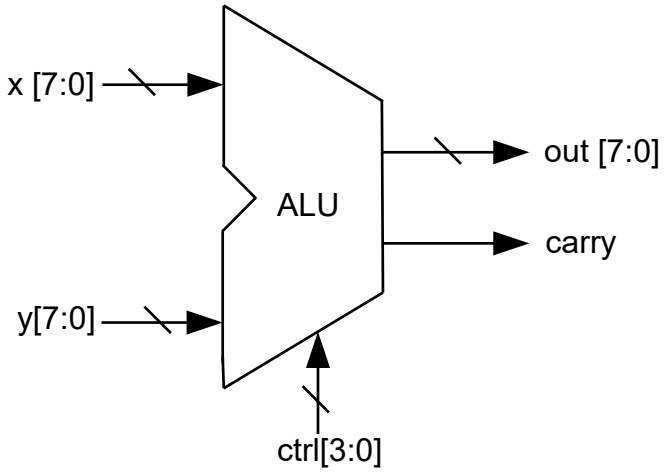

Figure 1

Table 1

Control Signal(ctrl)	Description	Function
0000	Add(signed)	out = x + y
0001	Sub(signed)	out = x - y
0010	Bitwise And	out = and(x, y)
0011	Bitwise Or	out = or(x , y)
0100	Bitwise Not	out = not(x)
0101	Bitwise Xor	out = xor(x , y)
0110	Bitwise Nor	out = nor(x , y)
0111	Shift left logical variable	out = y << x[2:0]
1000	Shift right logical variable	out = y >> x[2:0]
1001	Shift right arithmetic	out ={x[7],x[7:1]}
1010	Rotate left	out = {x[6:0] , x[7]}
1011	Rotate right	out = {x[0] , x[7:1]}
1100	Equal	out = (x==y)?1:0

1101	NOP (No operation)	out = 0
1110	NOP (No operation)	out = 0
1111	NOP (No operation)	out = 0

A. "carry" only needs to be considered in " **Add(signed)**", " **Sub(signed)**". As for the other functions, "carry" can be arbitrary.
B. carry is defined as the 9^{th} bit of the result of 8-bit signed addition
e.g. x=1001_0110 and y=0010_1101 → x+y=1_1100_0011 → carry=1

(1) (10%)
Use Verilog to implement the **RT**-level (use *continuous assignment*, ***assign***) model of the 8-bit ALU. Modify the "alu_assign.v" file, which contains the module name and input/output ports. Use the given testbench, "alu_assign_tb.v" to verify your design. Use the following command for simulation:

```
ncverilog alu_assign_tb.v alu_assign.v +access+r
```

(2) (10%)
Use Verilog to implement the **RT**–level (use *procedural assignment*, ***always*** *block*) model of the 8-bit ALU. The input and output ports are the same as the previous one. Modify the "alu_always.v" file, and use the given testbench, "alu_always_tb.v" to verify your design. Use the following command for simulation:

```
ncverilog alu_always_tb.v alu_always.v +access+r
```

(3) (20%)
The given two testbenches "alu_assign_tb.v" "alu_always_tb.v" don't check all the required functions of the ALU. You need to modify the two given testbenches and rename them to "alu_assign_tb2.v" and "alu_always_tb2.v", respectively (They are the same but for different modules!). Use your modified testbenches to verify if all functions of your design are correct. Show the waveform results and describe how you verify the correctness in your report. If how you execute your testbench is different from 1-(1) and 1-(2), please provide a README so that TA can correctly test your design.

Problem 2. 8x8 Register File (40%)

A register file consists of a set of registers that can be read or written. There are 8 registers in this register file, and the width of each register is 8-bits. The input and output ports are described in Figure 2.

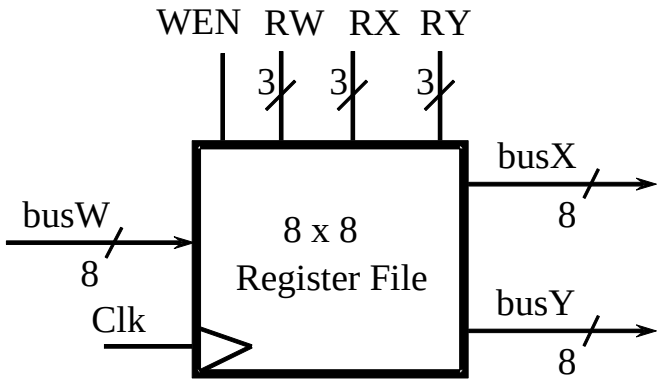

Figure 2

You must follow these specifications:

A. I/O Port Functionality
   (1) busW: 8 bits input data bus
   (2) busX、busY: 8 bits output data buses
   (3) WEN: active high write enable (WEN==1)
   (4) RW: select one of 8 registers to be written
   (5) RX: select one of 8 registers to be read, output on busX
   (6) RY: select one of 8 registers to be read, output on busY

B. Register File
   (1) 8 registers.
   (2) $r0~$r7
   (3) $r0=zero (constant zero, don't care any write operation)

C. Write Operation
   (1) The data on busW will be written into a specified register synchronously on positive edge of **Clk**
   (2) RW is the index of register to be written.

D. Read Operation
   (1) The register file behaves as a combinational logic block when reading
   (2) Read the data in the register file **synchronously**

(1) (20%)

Implement the register file in Verilog. Modify “register_file.v”, which contains the module name and input/output ports.

(2) (20%)

Write the testbench (“register_file_tb.v”) to verify your design. Show the waveform results and describe how you verify the correctness in your report. Use the following command for simulation:

```
ncverilog register_file_tb.v register_file.v +access+r
```

Problem 3. Simple Calculator (20%)

Combine the previous two designs (ALU, Register File) into a simple calculator unit. You can use this unit to execute some simple programs. The input and output ports are defined in Figure 3. And there is a control signal "Sel" to select which data is input to ALU.

(1) When Sel = 0, DataIn is passed to port x of ALU.

(2) When Sel = 1, the data loaded to port x of ALU is from the register file.

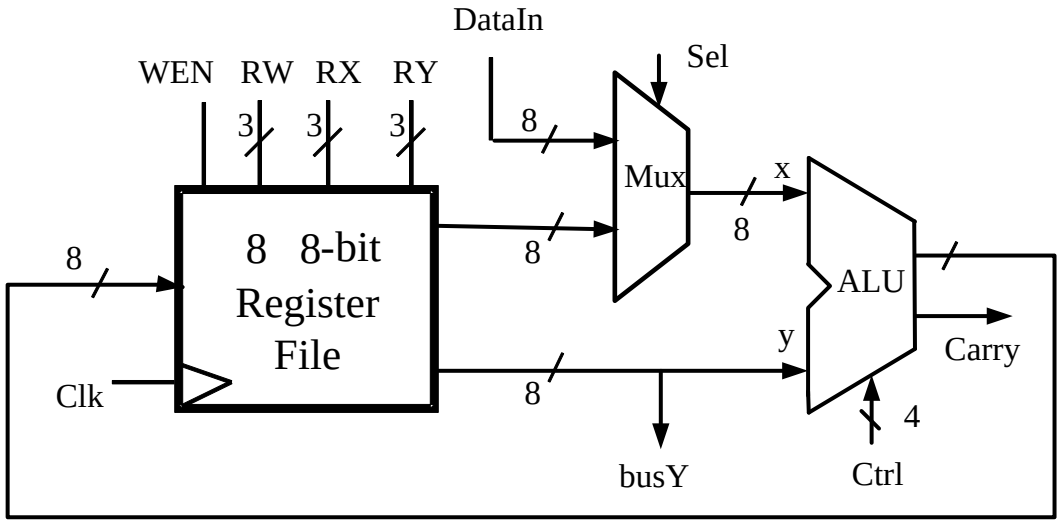

Figure 3

You must define the following ports in your design:

A. Input Port
   (1) Clk
   (2) WEN
   (3) RW
   (4) RX
   (5) RY
   (6) DataIn
   (7) Sel
   (8) Ctrl

B. Output Port
   (1) busY
   (2) Carry

(1) (20%)

Use the previous two modules to implement the simple calculator unit ("simple_calculator.v"). After finishing the calculator, use testbench ("simple_calculator_tb.v") to test the correctness of your design. Use the following command for simulation:

```
ncverilog simple_calculator_tb.v simple_calculator.v +access+r
```

In your report, you should include the following parts

(1) **ALU**: Screenshot the waveform result in nWave of alu_assign & alu_always, describe how you verify the correctness.

(2) **8x8 Register File**: Screenshot the waveform result in nWave of register_file, describe how you verify the correctness.

(3) **Describe what you found (bonus 5%):** Write down what you found. Feel free to share your experience. (Ex: some mistakes you spend a lot of time, your environment, naming method) Anything you feel is special. The points depend on your answers!

Submission requirements:

1. All the files need to be compressed as a single **ZIP file** and **uploaded to Ceiba**.
   Example of filename: DSD_HW2_b05901183.zip
   Your submitted file should include the following files:

```
DSD_HW2_b05901183/
    1-ALU/
        1_assign/
            alu_assign.v,    alu_assign_tb2.v
        2_always/
            alu_always.v,    alu_always_tb2.v
    2-RegisterFile/
        register_file.v,    register_file_tb.v
    3-SimpleCalculator/
        simple_calculator.v
    report_ HW2_b05901183.pdf
```

   Note that you have to replace b05901183 with your student ID number.

1. **Deadline: 2021/04/01 23:59**
2. **The homework will be graded ONLY IF the filename of your submission are correct!**
3. **Penalty for late submission: 20% per day**