Computer Organization – Project – Phase III

Email: iustCompOrg+4012@gmail.com

In this phase of the project you are going to design and implement execution modules (stage 2) of your CPU which are the ALU, MUL and SQRT units.

1) Based on the extracted fields from your decoding unit, determine whether the instruction needs source registers. Then read these registers by their index from the register file you implemented in phase I.

a. Construct the procedure for reading from the register file. Evaluate the input signals related to reading action:

`ReadEnable_1 / ReadIndex_1 / ReadEnable_2 / ReadIndex_2`

b. Monitor the result and make sure the registers are initialized correctly.

2) ALU: In this part you are tasked with designing the ALU and related circuits. The operation and the operands are determined by the control unit designed in the previous phase. The ALU must have the following input signals:

```
opcode (7 bits) / funct7 (7 bits) / funct7_valid / bus_rs1 (32
bits) /
bus_rs2 (32 bits) / immediate (32 bits)
```

and one output:

```
alu_output (32 bits)
```

and also one parameter for fixed point unit FLEN (FCVT function):

```
parameter FLEN = 10
```

For addition and subtraction operations use *Verilog*'s own operators '+' and '−'. (You don't need to implement any of the adder circuits you know)

You need to determine the validity of your second bus content; In fact, you'll need to design a mux to consider if immediate value is your second operand or if it is coming from either of the register files on `RS2_Bus`.

This ALU will support 3 instructions form the instruction memory:
`ADD / ADDI / FADD/ FCVT`

3) MUL : For Multiplication you can also use *Verilog*'s own operators '`*`'. But notice that your MUL module is separated from your ALU. The multiplier unit must have the following input signals:

```
opcode (7 bits) / funct7 (7 bits) / funct7_valid/ bus_rs1 (32
bits) /
bus_rs2 (32 bits)
```

and one output:

```
mul_output (32 bits)
```

Bonus point) For your ALU to support multiplication, implement the multiplier you designed in Verilog_HW1. (Remember this multiplier takes two 16-bit inputs and outputs a 32-bit value.)

4) SQRT: Four your ALU to support the square root operation, implement the square root calculator you designed in Verilog_HW2. You'll need to include your Fixed Point SQRT module in `SQRT_Unit.v` and add it to your design considering input and output signals required. (This source file `SQRT_Unit.v` will be released after Verilog_HW2 deadline)

SQRT unit parameters:

```
parameter WIDTH = 32 / parameter FBITS = 10
```

SQRT unit input signals:

```
CLK / Stage / opcode (7 bits) / funct7 (7 bits) / funct7_valid
/
bus_rs1 (32 bits)
```

SQRT unit output signals:

```
sqrt_output (32 bits) / sqrt_busy /
```

5) After you decoded the operation and the operands in your decoding circuit, and read the source registers from the register files, and executing the operation on the operands, it's time to write the result back in the register files.

Implement the register files write-back for the output of your ALU, index of the register to be written is determined by the control unit and decoding circuits. Evaluate the input signals and monitor the result and debug as needed:

```
WriteEnable / WriteIndex / WriteData
```

* Remember there are two separate register files for integer and fixed-point values and write-back process must be done accordingly.

* Register `x0` from the integer register file must be hardwired to zero, avoid any write procedures to this register. (i.e. abort the instruction if `WriteIndex` is set to zero)

[Hint: take a look at `RISCV_CPU_Datapath.pdf`. Block diagrams will help you with designing your CPU.]

<u>**Notes:**</u>

Send all your assignment related files (*Top module* and *Testbench* [.v] and [.vvp] and [.vcd] files) along with a detailed report in [.pdf] format all in one zip file to the email address of the class.

For this assignment you can work in groups of two, each participant must submit the assignment individually with their respective name and student number.

If you have any questions regarding this assignment, feel free to contact us.

Please submit your final project in the following format:

Name_StudentNumber_Project (BillGates_12345678_Project)

Good Luck!