Lab 7. General Purpose Register

Write code for general purpose register file. MIPS processor has a register file that contains 32 registers. Each register is 32-bit long. Lab task is to design General Purpose Register File.

We suggest having the following inputs:

clk, write_enable - 1-bit long

addrA, addrB, addrC

data_out_A, data_out_B, data_in_C – 32 bit long

Write Verilog code for 3 port general purpose register file. A port consists of an address and data input/output.

1) **Size of addresses**: If we have 32 registers, what should be size of address bus (addrA/B)? Give us the number of bits of addrA and addrB.

2) **Read Registers:** Implement logic in your Verilog code that allows us to read values stored in registers. 2 out of 3 ports should be output ports. (addrA, data_out_A, and addrB, data_out_B). Value of Register specified by addrA/B will be assigned to data_out_A/B. We suggest making Reading process independent of clock.

3) **Update Registers:** Implement logic in your Verilog code that allows us to update values stored in registers. 1 out of 3 ports should be input ports. (addrC, data_in_C). If write_enable is 1, value of data_in_c should be assigned to Register specified by addrC. Make sure that update happens at rising edge of the clock.

4) **Register 0:** In you code implement logic that makes sure that register $0 stays 0 all the time.

5) **Test Your design in Testbench:** Write testbench for your design. Generate Waveforms and explain in your reports why do you think your design works correctly.

HINTS: You are advised to create an array of registers: Syntax for the array is the following:

reg [(size of a register) – 1 : 0] NameOFArray [0 : (number of registers) – 1];

You can initialize the array values from text file with $readmemb command:

```
initial begin
        $readmemb("values.txt", NameOFArray);
end
```

Make sure that numbers in values.txt are written in binary and size of the numbers is the same as size of an array element. If you want to 1 will be written in a following manned:

00000000000000000000000000000001.