# Lab 07 – Worksheet

| Name: **Breeha and Namel** | ID: **08283 and 08327** | Section: |
|---|---|---|

## Task 1.

**Code: Design module & testbench**

Provide appropriately commented code for designed module & its testbench, code should contain meaningful variable naming.

**Add snippet or Copy paste the code written in the Editor window. Make sure the irrelevant area of the snip is cropped.*

**Design Module**

```
module registerFile(
   input [63:0] WriteData,
   input [4:0] RS1,
   input [4:0] RS2,
   input [4:0] RD,
   input RegWrite,
   input Clk,
   input Reset,
   output reg [63:0] ReadData1,
   output reg [63:0] ReadData2
   );
   reg [63:0] Registerx [31:0];
   integer index;
   initial
      begin
      //using loop to initialise the array of registers
         for (index=0;index<32;index=index+1)
            begin
               Registerx[index]=index;
            end
     end
   always@ (posedge Clk)
   begin
   //if regWrite==1 then assign write data to register RD (destination)
      if (RegWrite==1)
      begin
         Registerx[RD]=WriteData;
      end
   end
```

```
    always@(Registerx or Reset)
    begin
        if (Reset==1)
        begin
            ReadData1=0;
            ReadData2=0;
        end
        else
        begin
            ReadData1=Registerx[RS1];
            ReadData2=Registerx[RS2];
        end
   end
endmodule
```

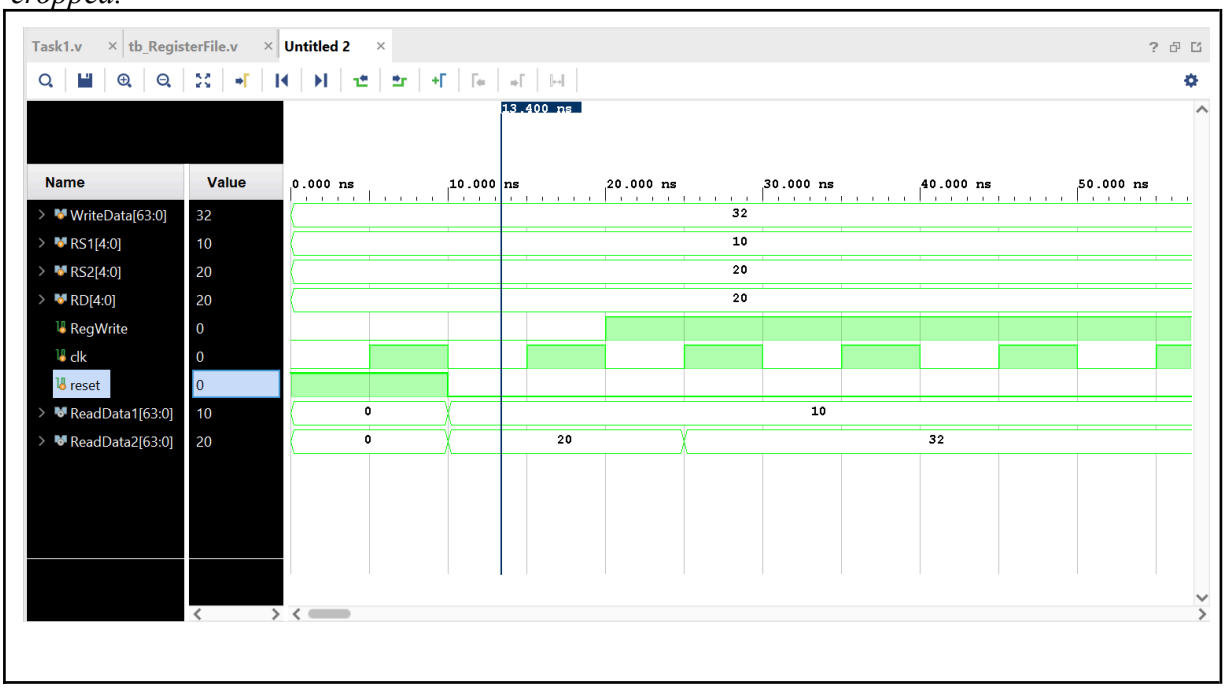
**Test Bench**

```
module tb_RegisterFile (
);
reg [63:0]WriteData;
reg [4:0]RS1;
reg [4:0]RS2;
reg [4:0]RD;
reg RegWrite, clk, reset;
wire [63:0]ReadData1;
wire [63:0]ReadData2;
registerFile regFile
(
WriteData,RS1, RS2,
RD,
RegWrite,
clk,
reset,
ReadData1,
ReadData2
);
initial
begin
clk = 0;
RegWrite = 0;
reset = 1;
// add x20,x10,x20
RS1 = 10; //ReadData1 reads value of register number 10 and the value is 11
RS2 = 20; //ReadData2 reads value in register number 20 and the value loaded is 21
```

```
WriteData = 64'd32; //This value 32 is given by user for now, the add instruction given above is just to show how does the process works
RD = 20; //The value of register number 20 was '21' . This 21 is the value or 'data'
#10 reset = 0;
#10 RegWrite = 1;//allow write in register 20
end
always
#5 clk=~clk;
endmodule
```

## Results (Waveforms)

**Add snip of relevant signals' waveforms. Make sure the irrelevant area of the snip is cropped.*



## Comments

**Observation/Comments on the obtained results/working of code.*

The module initializes registers with their indices, writes to a register on a positive clock edge if RegWrite is active, and supports simultaneous reading from two selected registers. According to their values, ReadData1 and ReadData2 take the required values from array of 32 registers in the module named Registerx

## Exercise

**Code: Design module & testbench**

Provide appropriately commented code for designed module & its testbench, code should contain meaningful variable naming.

**Add snippet or Copy paste the code written in the Editor window. Make sure the irrelevant area of the snip is cropped.*

**Design Module**

```
module top(
    input [31:0] instruction,
    output wire [63:0] ReadData1,
    output wire [63:0] ReadData2
    );
    wire [63:0] WriteData;
    wire RegWrite, clk,reset;

    wire [6:0] opcode; // assigned 7 bits to opcode
    wire[4:0] rd; // assigned 5 bits to rd
    wire [2:0] funct3; //assigned 3 bits to funct3
    wire [4:0] rs1; //assigned 5 bits to rs1
    wire [4:0] rs2; //assigned 5 bits to rs2
    wire [6:0] funct7; //assigned 7 bits to funct7

    //calling instruction parser module
    R_type R(instruction,opcode, rd, funct3, rs1, rs2, funct7);

    //calling registerFile module
    registerFile rg(WriteData,rs1, rs2, rd, RegWrite, clk, reset, ReadData1, ReadData2);

endmodule
```

**Test Bench**

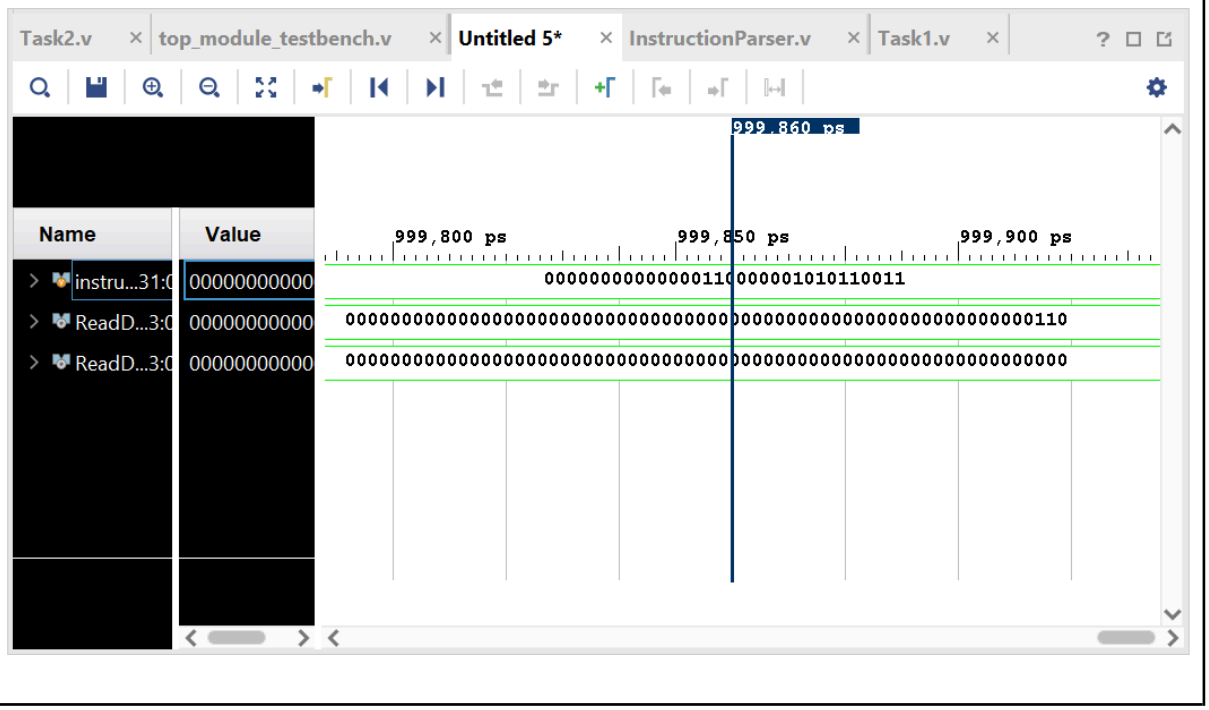
```
module top_module_testbench(
);
reg [31:0] instruction;
wire [63:0] ReadData1;
wire [63:0] ReadData2;

top tb(instruction, ReadData1, ReadData2);
```

```
initial
begin
 //giving binary of RISC V command : add x5,x6,x0
 instruction=32'b00000000000000011000000101011001 1;
 end
endmodule
```

## Results (Waveforms)

**Add snip of relevant signals' waveforms. Make sure the irrelevant area of the snip is cropped.*



## Comments

**Observation/Comments on the obtained results/working of code.*

Within the main module, the instruction decoder receives a 32-bit instruction and deciphers the `rs1`, `rs2`, and `rd` values from it. These extracted values are forwarded to the `registerFile` module, which subsequently produces `ReadData1` and `ReadData2`. These outputs, representing the register values specified in the RISC-V command instruction, are then outputted from the main module.

# Lab 7 – Register File

## Assessment Rubrics

| Task No. | LR 2<br>Code | LR 5<br>Results | AR 7<br>Report Submission |
|---|---|---|---|
| Task 1 | /30 | /20 | /20 |
| Task 2 | /20 | /10 | |
| Total Points | /100 Points | | |
| CLO Mapped | CLO 1 | | |

**Marks Distribution:**

*For description of different levels of the mapped rubrics, please refer the provided Lab Evaluation Assessment Rubrics.*

| # | Assessment Elements | Level 1: Unsatisfactory Points 0-1 | Level 2: Developing Points 2 | Level 3: Good Points 3 | Level 4: Exemplary Points 4 |
|---|---|---|---|---|---|
| LR2 | Program/Code/ Simulation Model/ Network Model | Program/code/simulation model/network model does not implement the required functionality and has several errors. The student is not able to utilize even the basic tools of the software. | Program/code/simulation model/network model has some errors and does not produce completely accurate results. Student has limited command on the basic tools of the software. | Program/code/simulation model/network model gives correct output but not efficiently implemented or implemented by computationally complex routine. | Program/code/simulation /network model is efficiently implemented and gives correct output. Student has full command on the basic tools of the software. |
| LR5 | Results & Plots | Figures/ graphs / tables are not developed or are poorly constructed with erroneous results. Titles, captions, units are not mentioned. Data is presented in an obscure manner. | Figures, graphs and tables are drawn but contain errors. Titles, captions, units are not accurate. Data presentation is not too clear. | All figures, graphs, tables are correctly drawn but contain minor errors or some of the details are missing. | Figures / graphs / tables are correctly drawn and appropriate titles/captions and proper units are mentioned. Data presentation is systematic. |

| | | | | | |
|---|---|---|---|---|---|
| **AR7** | **Report Content/Code Comments** | Most of the questions are not answered / figures are not labelled/ titles are not mentioned / units are not mentioned. No comments are present in the code. | Some of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Few comments are stated in the code. | Majority of the questions are answered, figures are labelled, titles are mentioned and units are mentioned. Comments are stated in the code. | All the questions are answered, figures are labelled, titles are mentioned and units are properly mentioned. Proper comments are stated in the code. |