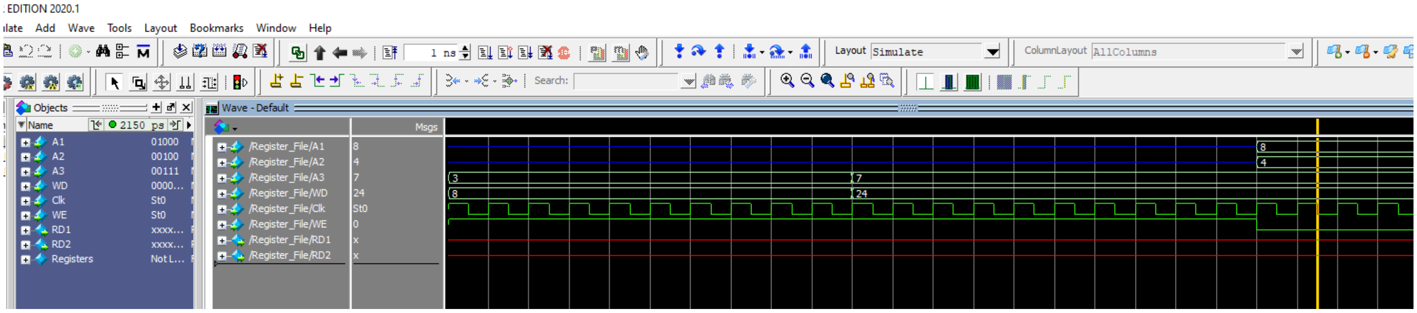
Objective of the Lab/Program

Design and Implement the MIPS Register file used for a Single Cycle Micro-architecture in Verilog HDL. The lab's focus will only be on the Register file component.

Source Code

```
//Register File Implementation: 32 registers each 32 bits wide, 1 write enable
//1 bit clock, write data 32 bits
module Register_File (A1, A2, A3, WD, WE, Clk, RD1, RD2);

    input [4:0]A1, A2, A3;
    input [31:0]WD;
    input [0:0]Clk, WE;

    output reg [31:0] RD1,RD2;

    reg[31:0] Registers [0:31];

    always @(posedge Clk)
    begin
        if(WE == 1)
        begin
            Registers[A3] <= WD; //write data into register A3
        end
    end

    always@(posedge Clk)
    begin
        RD1 <= Registers [A1];
        RD2 <= Registers [A2];
    end
endmodule
```

Screen shot of the simulation (waveforms) showing the results of the Register file operation. Show the results of Read and Write into any two registers.



Name	2150 ps
A1	01000
A2	00100
A3	00111
WD	0000...
Clk	St0
WE	St0
RD1	xxxx...
RD2	xxxx...
Registers	Not L...

	Msgs
/Register_File/A1	8
/Register_File/A2	4
/Register_File/A3	7
/Register_File/WD	24
/Register_File/Clk	St0
/Register_File/WE	0
/Register_File/RD1	x
/Register_File/RD2	x

Conclusion and References

After defining all the variables for input and output on the register two conditions used, one checked when the clock was ‘1’ and WE was ‘1’, while the other condition only checked for the clock to be ‘1’. Afterwards the program ran as intended.