DDCO Laboratory (UE23CS251A)

3rd Semester, Academic Year 2024-25 Date: 06/10/24

Week- Submission

Name: C Kaustubh	SRN: PES1UG23CS154	Section: C

Implement the following using the Verilog. After each experiment attach the following.

i. Verilog Code Screenshot.

ii. Verilog VVP Output Screen Shot.

iii. GTKWAVE Screenshot.

Q. 8 registers of 16-bits each with read and write functionality.

```
// 2 16-bit register file with read and write functionality

module reg_file (
    input wire clk,
    input wire reset,
    input wire wr,
    input wire [2:0] rd_addr_a, rd_addr_b, wr_addr,
    input wire [15:0] d_in,
    output reg [15:0] d_out_a, d_out_b
);

    reg [15:0] reg_array [7:0];

    always @(posedge clk) begin      You, 39 minutes ago • week5 registers
        if (reset) begin
            reg_array[0] <= 16'h0000;
            reg_array[1] <= 16'h0000;
            reg_array[2] <= 16'h0000;
            reg_array[3] <= 16'h0000;
            reg_array[4] <= 16'h0000;
            reg_array[5] <= 16'h0000;
            reg_array[6] <= 16'h0000;
            reg_array[7] <= 16'h0000;
        end else if (wr) begin
            reg_array[wr_addr] <= d_in;
        end
    end

    always @(*) begin
        d_out_a = reg_array[rd_addr_a];
        d_out_b = reg_array[rd_addr_b];
    end
endmodule
```

```
module reg_file_tb;

  reg clk, reset, wr;
  reg [2:0] rd_addr_a, rd_addr_b, wr_addr;
  reg [15:0] d_in;
  wire [15:0] d_out_a, d_out_b;

  reg_file uut (
    .clk(clk),
    .reset(reset),
    .wr(wr),
    .rd_addr_a(rd_addr_a),
    .rd_addr_b(rd_addr_b),
    .wr_addr(wr_addr),
    .d_in(d_in),
    .d_out_a(d_out_a),
    .d_out_b(d_out_b)
  );

  always #10 clk = ~clk;

  initial begin
    clk = 0;
    reset = 0;
    wr = 0;
    rd_addr_a = 3'b000;
    rd_addr_b = 3'b001;
    wr_addr = 3'b000;
    d_in = 16'h0000;

    $dumpfile("reg_file_tb.vcd");
    $dumpvars(0, reg_file_tb);

    reset = 1;
    #40;
    reset = 0;

    wr = 1;
    d_in = 16'h1234;
    wr_addr = 3'b000;
    #40;

    d_in = 16'h5678;
    wr_addr = 3'b001;
    #40;

    wr = 0;
    rd_addr_a = 3'b000;
    rd_addr_b = 3'b001;
    #40;

    $display("Read from reg 0: %h, Read from reg 1: %h", d_out_a, d_out_b);

    wr = 1;
    d_in = 16'h0ABC;
    wr_addr = 3'b010;
    #40;

    wr = 0;
    rd_addr_a = 3'b000;
    rd_addr_b = 3'b010;
    #40;

    $display("Read from reg 0: %h, Read from reg 2: %h", d_out_a, d_out_b);

    wr = 1;
    d_in = 16'hDEF0;
    wr_addr = 3'b000;
    #40;

    wr = 0;
    rd_addr_a = 3'b000;
    #40;

    $display("Read from reg 0 after overwrite: %h", d_out_a);

    $finish;
  end

endmodule
```

```
┌[® C Kaustubh from 🖳 DESKTOP-T2KDJ1A][⧖ 0s][▮ RAM: 10/21GB][🗓 Sunday at 12:50:29 PM][ main ≡ ✎ ?1]
└[D:\DDCO-Lab-UE23CS251A\week5]
└─Δ iverilog -o register reg_file.v lib.v register-tb.v
┌[® C Kaustubh from 🖳 DESKTOP-T2KDJ1A][⧖ 0.093s][▮ RAM: 10/21GB][🗓 Sunday at 12:50:32 PM][ main ≡ ✎ ?1]
└[D:\DDCO-Lab-UE23CS251A\week5]
└─Δ vvp register
VCD info: dumpfile reg_file_tb.vcd opened for output.
Read from reg 0: 1234, Read from reg 1: 5678
Read from reg 0: 1234, Read from reg 2: 9abc
Read from reg 0 after overwrite: def0
register-tb.v:89: $finish called at 320 (1s)
┌[® C Kaustubh from 🖳 DESKTOP-T2KDJ1A][⧖ 0.046s][▮ RAM: 10/21GB][🗓 Sunday at 12:50:35 PM][ main ≡ ✎ ?1]
└[D:\DDCO-Lab-UE23CS251A\week5]
└─Δ gtkwave reg_file_tb.vcd

GTKWave Analyzer v3.3.100 (w)1999-2019 BSI

[0] start time.
[320] end time.
```