CECS 225

Lab 5

Sotheanith Sok

1. Verilog module source

`timescale 1ns / 1ps

module regfile(clk, we3, ra1, ra2, wa3, wd3, rd1, rd2);

input clk; //clock signal

input we3; //write enable control signal

input [4:0] ra1, ra2, wa3; //Read address 1, 2, and Write address

input [31:0] wd3; //32-bit write data

output [31:0] rd1, rd2; //32-bit read data

reg [31:0] rf[31:0];

//three ported register file

// read two ports combinationally

// write thrid port on the falling edge of clk

// register 0 hardwired to 0

always @(negedge clk)

if (we3) rf[wa3] <=wd3;

assign rd1=(ra1 != 0) ? rf[ra1] : 0;

assign rd2=(ra2 != 0) ? rf[ra2] : 0;

endmodule

1. Verilog Test Fixture

`timescale 1ns / 1ps

module regfile\_Tester;

// Inputs

reg clk;

reg we3;

reg [4:0] ra1;

reg [4:0] ra2;

reg [4:0] wa3;

reg [31:0] wd3;

// Outputs

wire [31:0] rd1;

wire [31:0] rd2;

// Instantiate the Unit Under Test (UUT)

regfile uut (

.clk(clk),

.we3(we3),

.ra1(ra1),

.ra2(ra2),

.wa3(wa3),

.wd3(wd3),

.rd1(rd1),

.rd2(rd2)

);

always #5 clk=~clk;//clock pulse generator

initial begin

// Initialize Inputs

clk = 0;

we3 = 0;

ra1 = 0;

ra2 = 0;

wa3 = 0;

wd3 = 0;

//Test case 0

// Write SID in hex to register 10

// Read register 10 values on rd1 output

@(posedge clk) //wait until positive clock edge to set input values

we3 = 1; // enable a register write

wa3 = 10; //select register 10 as the destination register

wd3 = 32'h13256189; //lower 8 digits of SID in hex will be written to register 10

ra1 = 10; // select register 10 value to be put on rd1 output

#10;

//Test case 1

// Write hex SID multiplied by 3 to register 20

// Read register 20 values on rd2 output

@(posedge clk) //wait until positive clock edge to set input values

we3 = 1; // enable a register write

wa3 = 20; //select register 20 as the destination register

wd3 = 32'h13256189\*3; //lower 8 digits of SID in hex multiplied by 3 will be written to register 20

ra2 = 20; // select register 20 value to be put on rd2 output

#10;

//Test case 2

// Write SID multiplied by 30 to register 30

// Read register 30 values on rd2 output

@(posedge clk) //wait until positive clock edge to set input values

we3 = 1; // enable a register write

wa3 = 30; //select register 30 as the destination register

wd3 = 32’h13256189\*30; //lower 8 digits of SID in hex multiplied by 30 will be written to register 30

ra2 = 30; // select register 30 value to be put on rd2 output

#10;

//Test case 3

// Write SID divided by 30 to register 22

// Read register 22 values on rd1 output

@(posedge clk) //wait until positive clock edge to set input values

we3 = 1; // enable a register write

wa3 = 22; //select register 22 as the destination register

wd3 = 32'h13256189/30; //lower 8 digits of SID in hex divided by 30 will be written to register 22

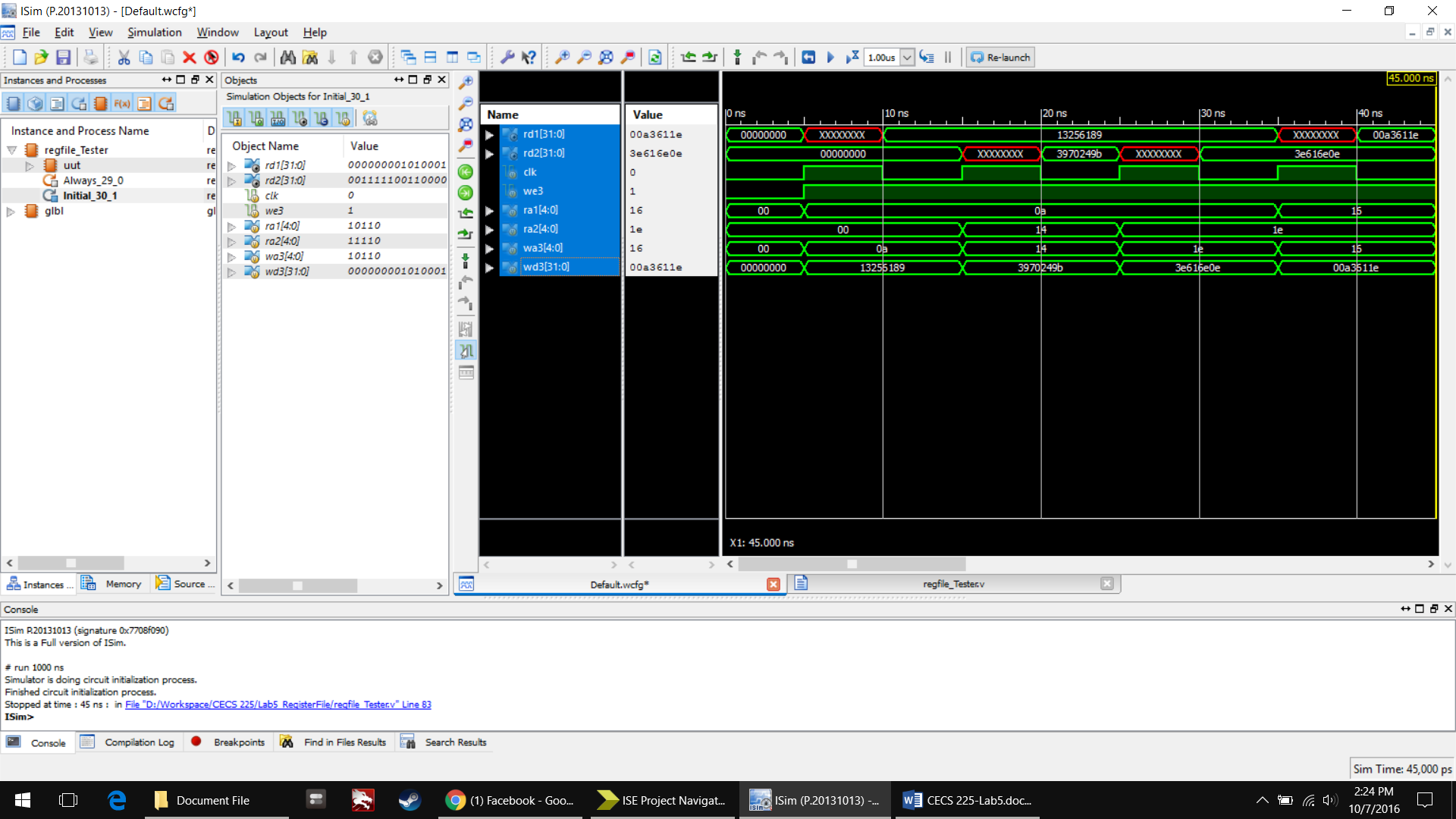
ra1 = 22; // select register 22 value to be put on rd1 output

#10;

$stop;

end

endmodule

1. Simulation Waveform
2. RTL Schematic Screenshot of regFile

