**Τμήμα Πληροφορικής**

**Μάθημα: Προηγμένες Εφαρμογές Ψηφιακής Σχεδίασης**

**Ονοματεπώνυμο : Ζαρκάδας Φώτης**

**ΑΕΜ: 5486**

**Εξάμηνο: 9ο**

**ΑΠΑΝΤΗΣΕΙΣ 2ης Σειράς ασκήσεων**

**Μέρος Ι**

Ο κώδικας rslatch.sv είναι:

module rslatch(input logic R,S,Clk,

output logic Q);

logic R\_g,S\_g,Qa,Qb /\* synthesis keep \*/;

assign R\_g = R && Clk;

assign S\_g = S && Clk;

assign Qa = ~(R\_g || Qb);

assign Qb = ~ (S\_g || Qa);

assign Q = Qa;

endmodule

Η εικόνα από τον rtl viewer είναι

Εικόνα που περιέχει διάγραμμα, γραμμή, στιγμιότυπο οθόνης, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation report είναι

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

O προσωμιωτής δείχνει

Εικόνα που περιέχει στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα

Χωρίς synthesis keep

O κώδικας rslatchnosynth.sv είναι

module rslatchnosynth(input logic R,S,Clk,

output logic Q);

logic R\_g,S\_g,Qa,Qb;

assign R\_g = R && Clk;

assign S\_g = S && Clk;

assign Qa = ~(R\_g || Qb);

assign Qb = ~ (S\_g || Qa);

assign Q = Qa;

endmodule

O RTL viewer δείχνει

Εικόνα που περιέχει διάγραμμα, στιγμιότυπο οθόνης, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation report είναι

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα

Η προσομοίωση

Εικόνα που περιέχει στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα

Μέρος ΙΙ

Ο κώδικας gdlatch.sv είναι

module gdlatch (input logic D,

input logic clk,

output logic Qa,Qb);

logic R,R\_g,S,S\_g;

assign S = D;

assign R = ~D;

assign R\_g = ~(R && clk);

assign S\_g = ~(S && clk);

assign Qa = ~(S\_g && Qb);

assign Qb = ~(R\_g && Qa);

endmodule

Σημείωση : το assign S = D Στην ουσία δε χρειάζεται (είναι περιττό)

Η εικόνα από τον RTL viewer είναι

Εικόνα που περιέχει στιγμιότυπο οθόνης, διάγραμμα, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation Report Είναι

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

Η προσομοίωση είναι

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, λογισμικό, λογισμικό πολυμέσων

Περιγραφή που δημιουργήθηκε αυτόματα

**Μέρος ΙΙΙ**

Ο κώδικας msdff.sv είναι

module msdff (input logic D,

input logic clk,

output logic Q,

output logic Qn);

logic Qm,nclk;

assign nclk = ~clk;

gdlatch Master(D,nclk,Qm);

gdlatch Slave(Qm,clk,Q,Qn);

endmodule

Η εικίνα από τον RTL viewer είναι

Εικόνα που περιέχει διάγραμμα, κείμενο, γραμμή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation report είναι

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

Η εικόνα από τη προσομοίωση είναι

Εικόνα που περιέχει στιγμιότυπο οθόνης, λογισμικό, λογισμικό πολυμέσων, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

**Μέρος IV**

Ο κώδικας lab2p4.sv είναι

module lab2p4(input logic D,clk,

output logic Qa,Qb,Qc);

dalwlatch latchD(D,clk,Qa);

pedff dffP(D,clk,Qb);

nedff dffN(D,clk,Qc);

endmodule

Ο κώδικας dalwlatch.sv είναι

module dalwlatch(input logic D, Clk,

output logic Q);

always\_latch

if (Clk) Q<=D;

endmodule

Ο κώδικας pedff.sv είναι

module pedff(input logic D,clk,

output logic Q);

always\_ff @(posedge clk)

Q <= D;

endmodule

O Κώδικας nedff.sv είναι

module nedff(input logic D,clk,

output logic Q);

always\_ff @(negedge clk)

Q <= D;

endmodule

Το RTL viewer δείχνει

Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

O κώδικας του testbench

testbenchp4.sv είναι

module testbenchp4();

logic D, clk;

logic Qa, Qb, Qc;

lab2p4 dut(D, clk, Qa, Qb, Qc);

initial begin

D=0; clk=1; #10;

D=0; clk=0; #20;

D=1; clk=0; #10;

D=1; clk=1; #20;

D=0; clk=1; #10;

end

endmodule

Η προσομοίωσει δείχνει

Εικόνα που περιέχει λογισμικό, λογισμικό πολυμέσων, λογισμικό γραφικών, κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation report

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

**Μέρος V**

Ο κώδικας regN.sv είναι

module regN

#(parameter N = 8)

(input logic [N-1:0] DataIn,

input logic clk,

input logic regWE,

input logic CLR,

input logic [2:0] Addr,

output logic [N-1:0] DataOut);

always\_ff @ (posedge clk, posedge CLR)

if (CLR) DataOut <= 0;

else if (regWE)

DataOut <= DataIn;

Endmodule

Ο RTL viewer δείχνει

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

Η προσομοίωση είναι

Εικόνα που περιέχει κείμενο, λογισμικό, λογισμικό πολυμέσων, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματαΟ κώδικας του tesbenchp5.sv είναι

module testbenchp5;

parameter N = 32;

logic [N-1:0] DataIn;

logic clk, regWE, CLR;

logic [N-1:0] DataOut;

regN DUT(DataIn,clk,regWE,CLR,DataOut);

always #15 clk = ~clk;

initial begin

clk = 0;

regWE = 0;

CLR = 0;

DataIn = 0;

CLR = 1; #10;

CLR = 0; #10;

regWE = 1;

DataIn = 32'hAB; #10;

regWE = 0;

DataIn = 32'hFF; #10;

end

endmodule

Το compilation report

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

**Μέρος V με Chat GPT**

**Άσκηση 1 και 2**

Ο κώδικας SregFile.sv είναι

module SregFile #(parameter N = 32) (

input logic clk,

input logic clr,

input logic regWE,

input logic [2:0] addr,

input logic [N-1:0] data\_in,

output logic [N-1:0] data\_out

);

logic [7:0] write\_enable;

logic [N-1:0] reg\_data [7:0];

// Instantiate Write Decoder

WRDecode wr\_decoder (

.addr(addr),

.regWE(regWE),

.enable(write\_enable)

);

// Instantiate 8 Registers

genvar i;

generate

for (i = 0; i < 8; i++) begin : register\_block

regN #(N) register\_inst (

.clk(clk),

.clr(clr),

.data\_in(data\_in),

.data\_out(reg\_data[i])

);

end

endgenerate

// Instantiate Multiplexer

mux3281 #(N) mux\_inst (

.in(reg\_data),

.sel(addr),

.out(data\_out)

);

Endmodule

Ο κώδικας RegN.sv

module regN #(parameter N = 32) (

input logic clk,

input logic clr,

input logic [N-1:0] data\_in,

output logic [N-1:0] data\_out

);

always\_ff @(posedge clk) begin

if (clr)

data\_out <= '0; // Clear the register

else

data\_out <= data\_in; // Store input data

end

endmodule

O κώδικας WRDecode.sv

module WRDecode (

input logic [2:0] addr,

input logic regWE,

output logic [7:0] enable

);

always\_comb begin

enable = 8'b0; // Default all disabled

if (regWE)

enable[addr] = 1'b1; // Enable specific register

end

endmodule

Ο κώδικας mux3281.sv

module mux3281 #(parameter N = 32) (

input logic [N-1:0] in [7:0], // 8 input buses, each N-bit wide

input logic [2:0] sel,

output logic [N-1:0] out

);

always\_comb begin

out = in[sel]; // Select the desired register output

end

endmodule

Και ο κώδικας testbenchp5.sv είναι

module testbenchp5;

parameter N = 32;

logic clk, clr, regWE;

logic [2:0] addr;

logic [N-1:0] data\_in;

logic [N-1:0] data\_out;

// Instantiate DUT

SregFile #(N) dut (

.clk(clk),

.clr(clr),

.regWE(regWE),

.addr(addr),

.data\_in(data\_in),

.data\_out(data\_out)

);

// Clock generation

initial clk = 0;

always #5 clk = ~clk;

// Test sequence

initial begin

clr = 1;

regWE = 0;

addr = 3'b000;

data\_in = 32'hDEADBEEF;

#10 clr = 0;

#10 regWE = 1; addr = 3'b000; data\_in = 32'h12345678;

#10 regWE = 1; addr = 3'b001; data\_in = 32'h87654321;

#10 regWE = 0; addr = 3'b000;

#10 addr = 3'b001;

#10 $stop;

end

endmodule

Η προσομοίωση

Εικόνα που περιέχει κείμενο, λογισμικό, λογισμικό πολυμέσων, λογισμικό γραφικών

Περιγραφή που δημιουργήθηκε αυτόματα

To compilation report

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα

**Άσκηση 3**

Ο κώδικας regFileBeh.sv είναι

module regFileBeh #(parameter N = 32, NUM\_REGS = 8) (

input logic clk, // Ρολόι

input logic clr, // Καθαρισμός (clear)

input logic regWE, // Ενεργοποίηση εγγραφής

input logic [2:0] addr, // Διεύθυνση για εγγραφή/ανάγνωση (3-bit)

input logic [N-1:0] data\_in, // Δεδομένα εισόδου

output logic [N-1:0] data\_out // Δεδομένα εξόδου

);

// Καταχωρητές (8 καταχωρητές των 32-bit)

logic [N-1:0] registers [NUM\_REGS-1:0];

// Εγγραφή στον καταχωρητή (σύγχρονη λειτουργία με ρολόι)

always\_ff @(posedge clk) begin

if (clr) begin

// Καθαρισμός όλων των καταχωρητών

for (int i = 0; i < NUM\_REGS; i++) begin

registers[i] <= '0;

end

end else if (regWE) begin

// Εγγραφή δεδομένων στον επιλεγμένο καταχωρητή

registers[addr] <= data\_in;

end

end

// Ανάγνωση από τον καταχωρητή (ασύγχρονη λειτουργία)

assign data\_out = registers[addr];

endmodule

Ο κώδικας του testbench regFileBeh\_tb.sv είναι

module regFileBeh\_tb;

parameter N = 32;

parameter NUM\_REGS = 8;

// Σήματα Testbench

logic clk, clr, regWE;

logic [2:0] addr;

logic [N-1:0] data\_in;

logic [N-1:0] data\_out;

// Instantiate DUT

regFileBeh #(N, NUM\_REGS) dut (

.clk(clk),

.clr(clr),

.regWE(regWE),

.addr(addr),

.data\_in(data\_in),

.data\_out(data\_out)

);

// Clock generation

initial clk = 0;

always #5 clk = ~clk;

// Test sequence

initial begin

// Αρχικοποίηση

clr = 1;

regWE = 0;

addr = 3'b000;

data\_in = 32'h00000000;

// Καθαρισμός καταχωρητών

#10 clr = 0;

// Εγγραφή στον καταχωρητή 0

#10 regWE = 1; addr = 3'b000; data\_in = 32'h12345678;

#10 regWE = 1; addr = 3'b001; data\_in = 32'h87654321;

// Ανάγνωση από καταχωρητή 0

#10 regWE = 0; addr = 3'b000;

// Ανάγνωση από καταχωρητή 1

#10 addr = 3'b001;

// Τέλος προσομοίωσης

#10 $stop;

end

endmodule

Η προσομοίωση

Εικόνα που περιέχει κείμενο, λογισμικό, λογισμικό πολυμέσων, λογισμικό γραφικών

Περιγραφή που δημιουργήθηκε αυτόματα

Το compilation report

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα