**Laborator 5**

**1)** Sa se scrie un modul avand **o intrare pe 8 biti(notata in)** si **4 iesiri pe 1 bit(notate a, b, c, d).** Iesirile vor fi egale cu: a = in[7] AND in[6]; b = in[5] OR in[4]; c = in[3] XOR in[2]; d = NOT in[1] AND NOT in[0].  
  
 **module** lab5\_1(

**input** [7:0] in,

**output** a,b,c,d

);

**assign** a = in[7] & in[6];

**assign** b = in[5] | in[4];

**assign** c = in[3] ^ in[2];

**assign** d = (~in[1]) & (~in[0]);

**endmodule**

**module** lab5\_1\_tb();

reg [7:0] in\_tb;

wire a\_tb,b\_tb,c\_tb,d\_tb;

*// monitor*

**always** @(in\_tb) **begin**

$monitor("time=%d , out : a=%b b=%b c=%b d=%b", $time,a\_tb,b\_tb,c\_tb,d\_tb);

**end**

*// generare stimuli*

**initial** **begin**

in\_tb = 0'b 10010011;

#1 in\_tb = 0'b 10110011;

#1 in\_tb = 0'b 11000101;

#1 in\_tb = 0'b 00100011;

**end**

*// instantiere DUT*

lab5\_1 aRandomName (

.in(in\_tb),

.a(a\_tb),

.b(b\_tb),

.c(c\_tb),

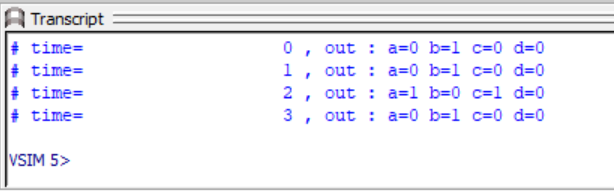
.d(d\_tb)

);

**endmodule**

**Graphical user interface

Description automatically generated**

****

**2)** Sa se scrie un modul avand **o intrare pe 8 biti** si **o iesire pe 1 bit.** Iesirea va fi egala cu functia **AND** aplicata tuturor bitilor intrarii.  
  
**module** lab5\_2 (

**input** [7:0] in,

**output** out

);

**assign** out = in[0] & in[1] & in[2] & in[3] & in[4] & in[5] & in[6] & in[7];

**endmodule**

**module** lab5\_2\_tb();

reg [7:0] in\_tb;

wire out\_tb;

*// monitor*

**always** @(in\_tb) **begin**

$monitor("Time=%d out=%b", $time, out\_tb);

**end**

*// generare stimuli*

**initial** **begin**

in\_tb = 0'b 11111111;

#1 in\_tb = 0'b 10100111;

#1 in\_tb = 0'b 10111110;

#1 in\_tb = 0'b 11110111;

**end**

*// instantiere DUT*

lab5\_2 aRandomName (

.in(in\_tb),

.out(out\_tb)

);

**endmodule**

**Graphical user interface, text, application

Description automatically generated with medium confidenceGraphical user interface

Description automatically generated**

**3)** Sa se scrie un modul cu **4 intrari pe 1 bit(notate a, b, c, d)** si **o** **iesire pe 1 bit. Iesirea va fi egala cu a AND (b OR c) OR NOT D.**

**module** lab5\_3(

**input** a,b,c,d,

**output** out

);

**assign** out = a & (b | c) | (~d);

**endmodule**

**module** lab5\_3\_tb();

reg a\_tb,b\_tb,c\_tb,d\_tb;

wire out\_tb;

*// monitor*

**always** @(a\_tb,b\_tb,c\_tb,d\_tb) **begin**

$monitor("Time=%d a=%b b=%b c=%b d=%b / out=%b", $time,a\_tb,b\_tb,c\_tb,d\_tb,out\_tb);

**end**

*// generare stimuli*

**initial** **begin**

a\_tb=0; b\_tb=1; c\_tb=1; d\_tb=0;

#1;

a\_tb=1; b\_tb=1; c\_tb=1; d\_tb=0;

#1;

a\_tb=1; b\_tb=0; c\_tb=0; d\_tb=1;

**end**

*// instantiere DUT*

lab5\_3 aRandomName (

.a(a\_tb),

.b(b\_tb),

.c(c\_tb),

.d(d\_tb),

.out(out\_tb)

);

**endmodule**

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

**4)** Sa se scrie un modul cu **2 intrari pe 1 bit (notate a, b) si o iesire pe 3 biti(notat out)**. Primul bit de la iesire va fi egal cu a AND b, al doilea bit de la iesire va fi egal cu a OR b, iar al3lea bit va fi egal cu a XOR b. **!Ordinea bitilor nu conteaza.**

**module** lab5\_4 (

**input** a,b,

**output** [2:0] out

);

**assign** out[0] = a & b;

**assign** out[1] = a | b;

**assign** out[2] = a ^ b;

**endmodule**

**module** lab5\_4\_tb();

reg a\_tb,b\_tb;

wire [2:0] out\_tb;

*// monitor*

**always** @(a\_tb,b\_tb) **begin**

$monitor("Time=%d a=%b b=%b / out : out[0]=%b out[1]=%b out[2]=%b",

$time,a\_tb,b\_tb,out\_tb[0],out\_tb[1],out\_tb[2]);

**end**

*// generare stimuli*

**initial** **begin**

a\_tb=0; b\_tb=1;

#1;

a\_tb=0; b\_tb=0;

#1;

a\_tb=1; b\_tb=0;

#1

a\_tb=1; b\_tb=1;

**end**

*// instantiere DUT*

lab5\_4 aRandomName (

.a(a\_tb),

.b(b\_tb),

.out(out\_tb)

);

**endmodule**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, chart

Description automatically generated

**5)** Sa se scrie un modul cu **2 intrari pe 8 biti si o iesire pe 8 biti.** Iesirea va fi egala cu functia AND aplicata intre bitii celor 2 intrari.

**module** lab5\_5(

**input** [7:0] in1, [7:0] in2,

**output** [7:0] out

);

**assign** out[0] = in1[0] & in2[0];

**assign** out[1] = in1[1] & in2[1];

**assign** out[2] = in1[2] & in2[2];

**assign** out[3] = in1[3] & in2[3];

**assign** out[4] = in1[4] & in2[4];

**assign** out[5] = in1[5] & in2[5];

**assign** out[6] = in1[6] & in2[6];

**assign** out[7] = in1[7] & in2[7];

**endmodule**

**module** lab5\_5\_tb();

reg [7:0] in1\_tb;

reg [7:0] in2\_tb;

wire [7:0] out\_tb;

*// monitor*

**always** @(in1\_tb,in2\_tb) **begin**

$monitor("Time=%d out[0]=%b out[1]=%b out[2]=%b out[3]=%b out[4]=%b out[5]=%b out[6]=%b out[7]=%b",

$time, out\_tb[0],out\_tb[1],out\_tb[2],out\_tb[3],out\_tb[4],out\_tb[5],out\_tb[6],out\_tb[7]);

**end**

*// generare stimuli*

**initial** **begin**

in1\_tb = 0'b 10110101;

in2\_tb = 0'b 01010111;

#1;

in1\_tb = 0'b 11111010;

in2\_tb = 0'b 11100010;

#1;

in1\_tb = 0'b 11111110;

in2\_tb = 0'b 00001010;

#1;

in1\_tb = 0'b 100010111;

in2\_tb = 0'b 010000110;

**end**

*// instantiere DUT*

lab5\_5 aRandomName (

.in1(in1\_tb),

.in2(in2\_tb),

.out(out\_tb)

);

**endmodule**

Graphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generated