**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 Problema2(

Input[7:0] in,

output reg a

);

always @(in) begin

a = in[7] & in[6] & in[5] & in[4] & in[3] & in[2] & in[1] & in[0];

end

endmodule

module Problema2\_tb;

wire a\_tb;

reg[7:0] in\_tb;

reg[7:0] val;

initial begin

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

end

Problema2 aRandomName( .a(a\_tb),

.in(in\_tb)

);

initial begin

for(val = 0; val < 255; val = val + 1)

begin

#1;

in\_tb = val;

end

end

endmodule

**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 Problema3(

input a,b,c,d,

output reg rezultat

);

always @(\*) begin

rezultat = a & (b | c) | ~d;

end

endmodule

module Problema3\_tb;

wire rezultat\_tb;

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

reg val;

initial begin

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

end

Problema3 aRandomName( .rezultat(rezultat\_tb),

.a(a\_tb),

.b(b\_tb),

.c(c\_tb),

.d(d\_tb)

);

initial begin

for(val = 0; val < 255; val = val + 1)

begin

#1;

rezultat\_tb = val;

end

end

endmodule

**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 Problema4(

input a,b,

output reg[2:0] out

);

always @(a,b) begin

out[0] = a & b;

out[1] = a | b;

out[2] = a ^ b;

end

endmodule

module Problema4\_tb;

wire[2:0] out\_tb;

reg a\_tb,b\_tb;

reg val;

initial begin

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

end

Problema4 aRandomName( .out(out\_tb),

.a(a\_tb),

.b(b\_tb)

);

initial begin

for(val = 0; val < 255; val = val + 1)

begin

#1;

out\_tb = val;

end

end

Endmodule

**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 Problema5(

Input[7:0] a,b,

output reg[7:0] out

);

always @(a,b) begin

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

out[1]=a[1] & b[1];

out[2]=a[2] & b[2];

out[3]=a[3] & b[3];

out[4]=a[4] & b[4];

out[5]=a[5] & b[5];

out[6]=a[6] & b[6];

out[7]=a[7] & b[7];

end

endmodule

module Problema5\_tb;

wire[7:0] out\_tb;

reg[7:0] a\_tb,b\_tb;

reg val;

initial begin

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

end

Problema5 aRandomName( .out(out\_tb),

.a(a\_tb),

.b(b\_tb)

);

initial begin

for(val = 0; val < 255; val = val + 1)

begin

#1;

out\_tb = val;

end

end

endmodule