1. **KANONİK İFADE**

F = B'D' + A'BD + AD' + A'BC'

1. **GRUPLANDIRMALAR**

Gruplar: (m0, m4, m12, m8), (m0, m8, m2, m10), (m5, m7), (m12, m8, m14, m10), (m4, m5).

1. **MAXTERMLER**

F = PRODUCT(M1, M3, M6, M9, M11, M13, M15)

1. **POS GÖSTERİMİ**

F = (A+B+C+D')(A+B+C'+D')(A+B'+C'+D)(A'+B'+C+D')(A'+B'+C'+D')(A'+B+C+D')(A'+B+C'+D')

1. **MINTERMLER**

f= SUM (m0, m2, m4, m5, m7, m8, m10, m12, m14)

1. **SOP GÖSTERİMİ**

F = A'B'C'D' + A'BC'D' + ABC'D'+ AB'C'D'+ A'BC'D + A'BCD + A'B'CD' + ABCD' + AB'CD'

1. **MODÜL KODLARI**
   1. **Expression Modülü**

module exp (x\_e,a,b,c,d);

input a,b,c,d;

output x\_e;

wire not\_a,not\_b,not\_c,not\_d;

wire w\_1,w\_2,w\_3,w\_4;

not(not\_a,a);

not(not\_b,b);

not(not\_c,c);

not(not\_d,d);

and(w\_1,not\_b,not\_d);

and(w\_2,not\_a,b,d);

and(w\_3,a,not\_d);

and(w\_4,not\_a,b,not\_c);

or(x\_e,w\_1,w\_2,w\_3,w\_4);

endmodule

* 1. **SOP Modülü**

module sop(x\_out,a,b,c,d);

input a,b,c,d;

output x\_out;

wire w\_1,w\_2,w\_3,w\_4,w\_5,w\_6,w\_7,w\_8,w\_9;

wire not\_a,not\_b,not\_c,not\_d;

not (not\_a, a);

not (not\_b, b);

not (not\_c, c);

not (not\_d, d);

and(w\_1, not\_a,not\_b,not\_c,not\_d);

and(w\_2, not\_a,b,not\_c,not\_d);

and(w\_3, a,b,not\_c,not\_d);

and(w\_4, a,not\_b,not\_c,not\_d);

and(w\_5, not\_a,b,not\_c,d);

and(w\_6, not\_a,b,c,d);

and(w\_7, not\_a,not\_b,c,not\_d);

and(w\_8, a,b,c,not\_d);

and(w\_9, a,not\_b,c,not\_d);

or (x\_out,w\_1,w\_2,w\_3,w\_4,w\_5,w\_6,w\_7,w\_8,w\_9);

endmodule

* 1. **POS Modülü**

module pos(x\_out,a,b,c,d);

input a,b,c,d;

output x\_out;

wire w\_1,w\_2,w\_3,w\_4,w\_5,w\_6,w\_7;

wire not\_a,not\_b,not\_c,not\_d;

not (not\_a, a);

not (not\_b, b);

not (not\_c, c);

not (not\_d, d);

or(w\_1,a,b,c,not\_d);

or(w\_2,a,b,not\_c,not\_d);

or(w\_3,a,not\_b,not\_c,d);

or(w\_4,not\_a,not\_b,c,not\_d);

or(w\_5,not\_a,not\_b,not\_c,not\_d);

or(w\_6,not\_a,b,c,not\_d);

or(w\_7,not\_a,b,not\_c,not\_d);

and(x\_out,w\_1,w\_2,w\_3,w\_4,w\_5,w\_6,w\_7);

endmodule

* 1. **Testbench**

module tb();

reg a,b,c,d;

wire x\_e, x\_s,x\_p;

exp exp1(x\_e, a, b, c, d);

sop sop1(x\_s, a, b, c, d);

pos pos1(x\_p, a, b, c, d);

initial

begin

#100

a=1'b0;

b=1'b0;

c=1'b0;

d=1'b0;

#100

a=1'b0;

b=1'b0;

c=1'b0;

d=1'b1;

#100

a=1'b0;

b=1'b0;

c=1'b1;

d=1'b0;

#100

a=1'b0;

b=1'b0;

c=1'b1;

d=1'b1;

#100

a=1'b0;

b=1'b1;

c=1'b0;

d=1'b0;

#100

a=1'b0;

b=1'b1;

c=1'b0;

d=1'b1;

#100

a=1'b0;

b=1'b1;

c=1'b1;

d=1'b0;

#100

a=1'b0;

b=1'b1;

c=1'b1;

d=1'b1;

#100

a=1'b1;

b=1'b0;

c=1'b0;

d=1'b0;

#100

a=1'b1;

b=1'b0;

c=1'b0;

d=1'b1;

#100

a=1'b1;

b=1'b0;

c=1'b1;

d=1'b0;

#100

a=1'b1;

b=1'b0;

c=1'b1;

d=1'b1;

#100

a=1'b1;

b=1'b1;

c=1'b0;

d=1'b0;

#100

a=1'b1;

b=1'b1;

c=1'b0;

d=1'b1;

#100

a=1'b1;

b=1'b1;

c=1'b1;

d=1'b0;

#100

a=1'b1;

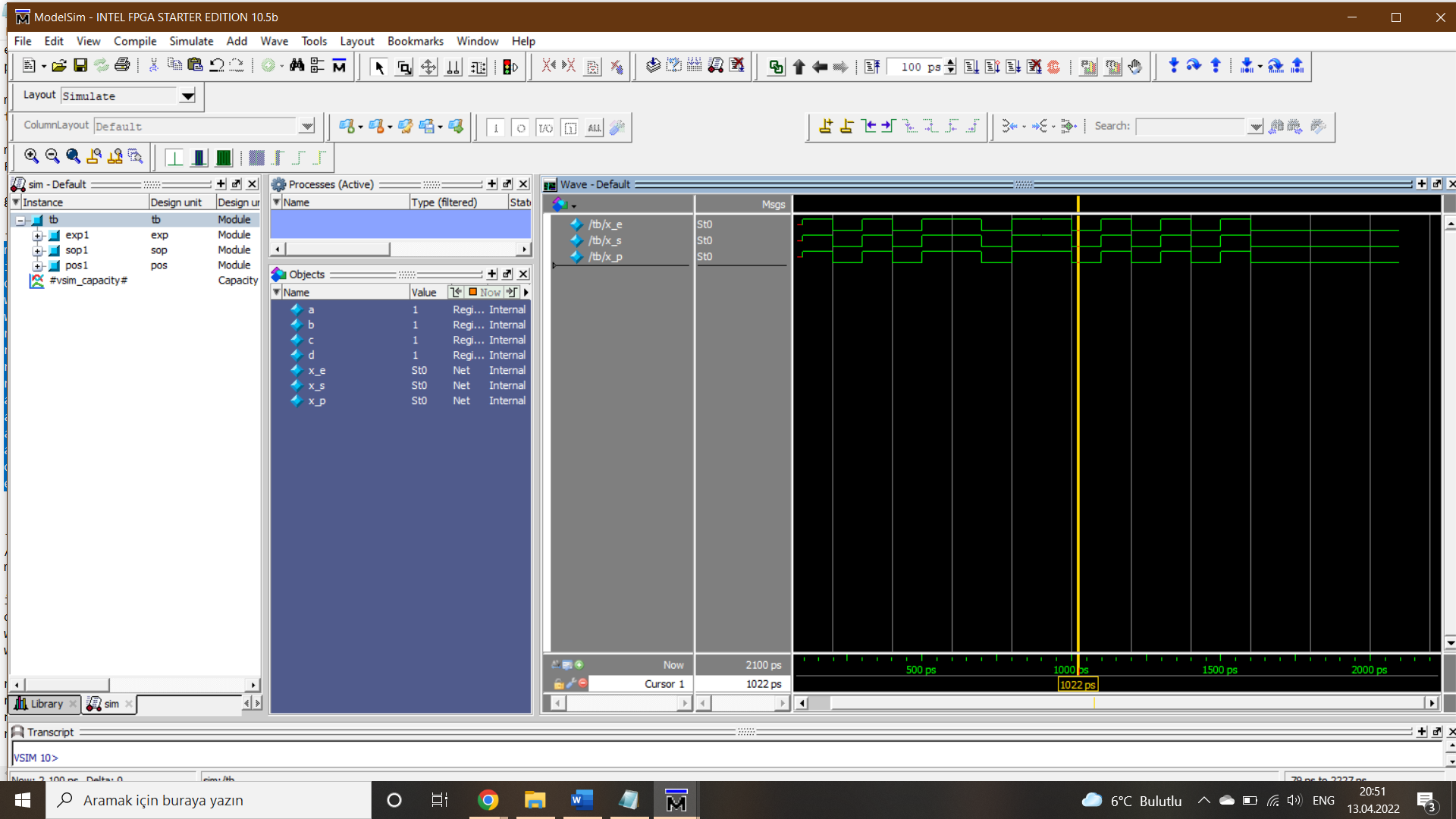
b=1'b1;

c=1'b1;

d=1'b1;

end

endmodule

1. **MODELSİM SİMÜLASYONU**

Farklı giriş kombinasyonları için çıkışlar aynı sinyalleri vermektedir.