library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity Comparator\_comportamental is

port(

a : in BIT;

b : in BIT;

c : in BIT;

d : in BIT;

f1 : out BIT;

f2 : out BIT;

f3 : out BIT

);

end Comparator\_comportamental;

--}} End of automatically maintained section

architecture Arhitectura of Comparator\_comportamental is

signal notac,notbd,nota,notb,notc,notd:bit;

signal s1,s2,s3,p1,p2,p3:bit;

begin

-- enter your statements here --

NA: process(a)

begin

nota <= not(a);

end process NA;

NB: process(b)

begin

notb <= not(b);

end process NB;

NC: process(c)

begin

notc <= not(c);

end process NC;

ND: process(d)

begin

notd <= not(d);

end process ND;

XOR\_AC:process(a,c)

begin

notac <= a xnor c;

end process XOR\_AC;

XOR\_BD:process(b,d)

begin

notbd <= b xnor d;

end process XOR\_BD;

P11:process(nota,notb,d)

begin

p1 <= nota and notb and d;

end process P11;

P12:process(notb,c,d)

begin

p2 <= notb and c and d;

end process P12;

P13:process(nota,c)

begin

p3 <= nota and c;

end process P13;

S11:process(a,notc)

begin

s1 <= a and notc;

end process S11;

S12:process(a,b,notd)

begin

s2 <= a and b and notd;

end process S12;

S13:process(b,notc,notd)

begin

s3 <= b and notc and notd;

end process S13;

FINAL1:process(notac,notbd)

begin

f1 <= notac and notbd;

end process FINAL1;

FINAL2:process(p1,p2,p3)

begin

f2 <= p1 or p2 or p3;

end process FINAL2;

FINAL3:process(s1,s2,s3)

begin

f3 <= s1 or s2 or s3;

end process FINAL3;

end Arhitectura;