library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

ENTITY mux21 IS PORT (

E: IN Std\_Logic;

S: IN Std\_Logic;

D0: IN Std\_Logic;

D1: IN Std\_Logic;

Y: OUT Std\_Logic);

end mux21;

ARCHITECTURE arch OF mux21 IS

BEGIN

PROCESS (E, S, D0, D1)

BEGIN

IF E='1' THEN

CASE S IS

WHEN '0' => Y <= D0 AFTER 10 ns;

WHEN '1' => Y <= D1 AFTER 10 ns;

WHEN OTHERS => Y <= '0' AFTER 10 ns;

END CASE;

ELSE

Y <= '0' AFTER 10 ns;

END IF;

END PROCESS;

END arch;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

ENTITY mux161 IS PORT (

D: IN STD\_LOGIC\_VECTOR(0 TO 15);

S: IN STD\_LOGIC\_VECTOR(0 TO 3);

E: IN STD\_LOGIC;

f: OUT STD\_LOGIC );

END mux161;

ARCHITECTURE multipleksor OF mux161 IS

component mux21 port (E,S,D0,D1: IN std\_logic; Y: OUT std\_logic);

end component;

signal i: std\_logic\_vector (0 TO 13);

BEGIN

c1: mux21 port map (D0=>D(0), D1=>D(1), Y=>i(0), S=>S(0), E=>E);

c2: mux21 port map (D0=>D(2), D1=>D(3), Y=>i(1), S=>S(0), E=>E);

c3: mux21 port map (D0=>D(4), D1=>D(5), Y=>i(2), S=>S(0), E=>E);

c4: mux21 port map (D0=>D(6), D1=>D(7), Y=>i(3), S=>S(0), E=>E);

c5: mux21 port map (D0=>D(8), D1=>D(9), Y=>i(4), S=>S(0), E=>E);

c6: mux21 port map (D0=>D(10), D1=>D(11), Y=>i(5), S=>S(0), E=>E);

c7: mux21 port map (D0=>D(12), D1=>D(13), Y=>i(6), S=>S(0), E=>E);

c8: mux21 port map (D0=>D(14), D1=>D(15), Y=>i(7), S=>S(0), E=>E);

c9: mux21 port map (D0=>i(0), D1=>i(1), Y=>i(8), S=>S(1), E=>E);

c10:mux21 port map (D0=>i(2), D1=>i(3), Y=>i(9), S=>S(1), E=>E);

c11:mux21 port map (D0=>i(4), D1=>i(5), Y=>i(10), S=>S(1), E=>E);

c12:mux21 port map (D0=>i(6), D1=>i(7), Y=>i(11), S=>S(1), E=>E);

c13:mux21 port map (D0=>i(8), D1=>i(9), Y=>i(12), S=>S(2), E=>E);

c14:mux21 port map (D0=>i(10), D1=>i(11), Y=>i(13), S=>S(2), E=>E);

c15:mux21 port map (D0=>i(12), D1=>i(13), Y=>f, S=>S(3), E=>E);

END multipleksor;

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

ENTITY funkcija IS PORT (

A: IN STD\_LOGIC;

B: IN STD\_LOGIC;

C: IN STD\_LOGIC;

D: IN STD\_LOGIC;

f: OUT STD\_LOGIC );

END funkcija;

ARCHITECTURE sklop OF Funkcija IS

SIGNAL ulaz:std\_logic\_vector(0 to 3);

BEGIN

ulaz<=(D,C,B,A);

sklopka: entity work.mux161 port map ( ( '0','0','1','1','1','1','0','1','1','1','1','0','0','1','1','0'),ulaz,'1',f);

END sklop;