**ΕΡΓΑΣΙΑ/ΔΙΑΓΩΝΙΣΜΑ 48ΩΡΟ**

ΚΑΡΑΒΙΑ ΜΑΡΙΑ ΣΠΥΡΙΔΟΥΛΑ (ΑΜ:2026202100049)

* *ΘΕΜΑ 1*

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας και θυρων

entity function\_proti is

    Port ( clk : in STD\_LOGIC;

           rst : in STD\_LOGIC;

           X : in STD\_LOGIC;

           Y : in STD\_LOGIC;

           Z: in STD\_LOGIC;

           F : out STD\_LOGIC);

end function\_proti;

--δηλωση αρχιτεκτονικης

architecture Behavioral of function\_proti is

begin

process (clk,rst)  --εναρξη διεργασιας

    begin

         if rst = '1' then

            F<='0' ;

          elsif rising\_edge(clk) then –ανοδικη ακμη ρολογιου

            F<=( not X or Y) and (X or not Y) and (X or not Z);

           end if ;

end process ;

end Behavioral;

*--TESTBENCH*   
   
library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity tb\_function\_proti is

end tb\_function\_proti;

--δηλωση αρχιτεκτονικης

architecture Behavioral of tb\_function\_proti is

    signal clk, rst, X, Y, Z, F: STD\_LOGIC; --λογικα σηματα

-- component δηλωση για την οντοτητα function\_proti

    component function\_proti

        Port ( clk : in STD\_LOGIC;

               rst : in STD\_LOGIC;

               X : in STD\_LOGIC;

               Y : in STD\_LOGIC;

               Z: in STD\_LOGIC;

               F : out STD\_LOGIC);

    end component;

--εναρξη τεστ

begin

    DUT: function\_proti

port map (clk, rst, X, Y, Z, F);

  --διεργασια ρολογιου

    process

    begin

        clk <= '0';

        wait for 10 ns;

        clk <= '1';

        wait for 10 ns;

    end process;

    process

    begin

        -- αρχικοποιηση εισοδων

        rst <= '1';

        X <= '0';

        Y <= '0';

        Z <= '0';

        wait for 20 ns;

        rst <= '0';

        wait for 10 ns;

        X <= '0';

        Y <= '0';

        Z <= '0';

        wait for 20 ns;

        X <= '1';

        Y <= '0';

        Z <= '1';

        wait for 20 ns;

        wait;

    end process;

end Behavioral;

* *ΘΕΜΑ 2*   
  library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity function\_deyteri is

    Port( X : in STD\_LOGIC;

          Y :in STD\_LOGIC;

          Z : in STD\_LOGIC;

          F : in STD\_LOGIC);

end function\_deyteri;

--δηλωση αρχιτεκτονικης

architecture Structural of function\_deyteri is

--component δηλωσεις για τις πυλες

component AND\_gate

        Port ( a, b , d: in STD\_LOGIC;

               c : out STD\_LOGIC);

    end component;

    component OR\_gate

        Port ( a, b : in STD\_LOGIC;

               c : out STD\_LOGIC);

    end component;

    component NOT\_gate

        Port ( a : in STD\_LOGIC;

               c : out STD\_LOGIC);

    end component;

    signal Sig01, Sig02, Sig03, Sig04: STD\_LOGIC; --λογικα σηματα

--δημιουργια πυλων και συνδεση των εισοδων με τα σηματα εξοδου

begin

    UT1 : NOT\_gate port map ( X , Sig01);

    UT2 : OR\_gate port map (Sig01, Y,Sig02);

    UT3 : NOT\_gate port map (Y , Sig01);

    UT4 : OR\_gate port map ( X, Sig01, Sig03);

    UT5: NOT\_gate port map( Z ,Sig01);

    UT6: OR\_gate port map ( X, Sig01 , Sig04);

    UT7: AND\_gate port map (Sig02, Sig03, Sig04, F); --εξοδος της συναρτης

end Structural;

*--TESTBENCH*    
   
library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

 entity tb\_function\_deyteri is

end tb\_function\_deyteri;

--δηλωση αρχιτεκτονικης

architecture Structural of tb\_function\_deyteri is

--δηλωση σηματων

    signal X, Y, Z, F: STD\_LOGIC;

    --component δηλωση για την οντοτητα function\_deyteri

    component function\_deyteri

        Port( X : in STD\_LOGIC;

              Y : in STD\_LOGIC;

              Z : in STD\_LOGIC;

              F : in STD\_LOGIC);

    end component;

--εναρξη ελεγχου αποτελεσματων

begin

    DUT: function\_deyteri port map(X, Y, Z, F);

     process

    begin

        wait for 20 ns;

--αναθεση τιμων στις εισοδους

        X <= '0'; Y <= '0'; Z <= '0';

wait for 15 ns;

        X <= '1'; Y <= '0'; Z <= '0';

wait for 15 ns;

        X <= '0'; Y <= '1'; Z <= '0';

wait for 15 ns;

        X <= '1'; Y <= '1'; Z <= '0';

wait for 15 ns;

        X <= '0'; Y <= '0'; Z <= '1';

wait for 15 ns;

        X <= '1'; Y <= '0'; Z <= '1';

wait for 15 ns;

        X <= '0'; Y <= '1'; Z <= '1';

wait for 15 ns;

        X <= '1'; Y <= '1'; Z <= '1';

wait;

    end process;

    process

    begin

        wait for 20 ns;

        --ελεγχος αποτελεσματων και αναφορα σφαλματων

        assert F = '0' report "Test Case 1 Failed" severity error;

        wait for 15 ns;

        assert F = '1' report "Test Case 2 Failed" severity error;

        wait for 15 ns;

        assert F = '1' report "Test Case 3 Failed" severity error;

        wait for 15 ns;

        assert F = '0' report "Test Case 4 Failed" severity error;

        wait for 15 ns;

        assert F = '1' report "Test Case 5 Failed" severity error;

        wait for 15 ns;

        assert F = '1' report "Test Case 6 Failed" severity error;

        wait for 15 ns;

        assert F = '1' report "Test Case 7 Failed" severity error;

        wait for 15 ns;

        assert F = '0' report "Test Case 8 Failed" severity error;

        wait;

    end process;

end Structural ;   
   
 

* *ΘΕΜΑ 3*  
     
  --adder,behavioral

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity adder is

Port (a : in STD\_LOGIC;

       b : in STD\_LOGIC;

       in\_c: in STD\_LOGIC;

       out\_c: out STD\_LOGIC;

       sum : out STD\_LOGIC);

end adder;

--δηλωση αρχιτεκτονικης

architecture Behavioral of adder is

begin

out\_c <= ( in\_c AND ( a XOR b)) OR (a AND b);

sum <= (a XOR b) XOR in\_c ;

end Behavioral; 

*--TESTBENCH*   
   
library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

  --δηλωση οντοτητας

entity tb\_adder is

end tb\_adder;

  --δηλωση αρχιτεκτονικης

architecture Behavioral of tb\_adder is

--δηλωση component την οντοτητα adder

component adder is

Port (a : in STD\_LOGIC;

       b : in STD\_LOGIC;

       in\_c: in STD\_LOGIC;

       out\_c: out STD\_LOGIC;

       sum : out STD\_LOGIC);

end component;

--δηλωση σηματων

signal i\_a : std\_logic;

signal i\_b : std\_logic;

signal o\_s : std\_logic;

signal i\_c : std\_logic;

signal o\_c : std\_logic;

--εναρξη τεστ

begin

DUT: adder port map (i\_a, i\_b, i\_c , o\_c, o\_s);

process

begin

--περιπτωση 1

    i\_a <='0';

    i\_b <= '0';

    i\_c <='0';

    wait for 15 ns;

--περιπτωση 2

    i\_a <= '0';

    i\_b <= '0';

    i\_c <='1';

    wait for 15 ns;

--περιπτωση 3

    i\_a <='0';

i\_b <='1';

    i\_c <='1';

    wait for 15 ns;

--περιπτωση 4

    i\_a <= '1';

    i\_b <='0';

    i\_c <='0';

    wait for 15 ns;

--περιπτωση 5

    i\_a <= '1';

    i\_b <='0';

    i\_c <='1';

    wait for 15 ns;

--περιπτωση 6

    i\_a <= '1';

    i\_b <='1';

    i\_c <='0';

    wait for 15 ns;

--περιπτωση 7

    i\_a <='1';

    i\_b <='1';

    i\_c <= '1';

    wait for 20 ns;

    wait;

end process;

end Behavioral;

--adder,structural   
   
library ieee;

use ieee.std\_logic\_1164.all;

--δηλωση οντοτητας

entity adder\_str is

Port (a : in STD\_LOGIC;

       b : in STD\_LOGIC;

       in\_c: in STD\_LOGIC;

       out\_c: out STD\_LOGIC;

       sum : out STD\_LOGIC);

  );

end adder\_str;  

--δηλωση αρχιτεκτονικης

architecture structural of adder\_str is 

signal Sig01, Sig02, Sig03: std\_logic;  -- δηλωση σηματων

begin

--λογικες πραξεις μεταξυ των εισοδων που αποθηκευονται στα σηματα εξοδου

   Sig01<= a xor b;

   Sig02 <= a and b;

   Sig03 <= Sig01 and in\_c;

   out\_c <= Sig02 or Sig03;

   sum <= Sig01 xor in\_c;

end structural;

*--TESTBENCH*   
   
USE IEEE.Std\_logic\_1164.all;

 --δηλωση οντοτητας

entity tb\_adder­\_str is

end tb\_adder­\_str;

--δηλωση αρχιτεκτονικης

architecture  structural of tb\_adder­\_str is

--δηλωση component για την οντοτητα adder\_str

component adder\_str

   Port (a : in STD\_LOGIC;

       b : in STD\_LOGIC;

       in\_c: in STD\_LOGIC;

       out\_c: out STD\_LOGIC;

       sum : out STD\_LOGIC);

  );

end component;

signal Sa,Sb,S\_in\_c: std\_logic:='0'; --σηματα

signal S\_sum,S\_out\_c: std\_logic;

--εναρξη τεστ

begin

stradder: adder\_str

port map ( a => Sa,

    b => Sb,

     in\_c=> S\_in\_c,

    sum => S\_sum,

    out\_c => S\_out\_c

   );

  process

  begin

--περιπτωση 1

   Sa <= '0';

   Sb <= '0';

   S\_in\_c <='0';

    wait for 50 ns;

  --περιπτωση 2

    Sa <= '0';

   Sb <= '0';

    S\_in\_c <='1';

   wait for 50 ns;

     --περιπτωση 3

   Sa<= '0';

    Sb<= '1';

    S\_in\_c<='1';

   wait for 50 ns;

--περιπτωση 4

   Sa <= '0';

    Sb<='1';

    S\_in\_c<='0';

   wait for 50 ns;

--περιπτωση 5

   Sa <= '1';

   Sb<= '0';

   S\_in\_c <= '0';

    wait for 50 ns;

  --περιπτωση 6

    Sa <= '1';

   Sb <= '0';

   S\_in\_c <= '1';

    wait for 50 ns;

--περιπτωση 7

   Sa <= '1';

   Sb<= '1';

   S\_in\_c<= '0';

    wait for 50 ns;

--περιπτωση 8

    Sa <= '1';

    Sb<= '1';

   S\_in\_c <= '1';

   wait for 100 ns;

  end process;

end structural;   
 

* *ΘΕΜΑ 4*   
     
  --SUBTRACTOR

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

--δηλωση οντοτητας

entity subtractor\_4bit is

port (

a, b: in std\_logic\_vector(3 downto 0);

result: out std\_logic\_vector(3 downto 0);

borrowout: out std\_logic

);

end entity subtractor\_4bit;

--δηλωση αρχιτεκτονικης

architecture behavioral of subtractor\_4bit is

begin

process(a, b) -- διεργασια με λιστα ευαισθησιας

begin

--ελεγχος συνθηκης για το αποτελεσμα του borrowout

if unsigned(a) < unsigned(b) then

result <= (others => '0');

borrowout <= '1';

else

result <= std\_logic\_vector(unsigned(a) - unsigned(b));

borrowout <= '0';

end if;

end process;

end behavioral;

*--testbench*

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

--δηλωση οντοτητας

entity tb\_subtractor\_4bit is

end entity tb\_subtractor\_4bit;

--δηλωση αρχιτεκτονικης

architecture behavioral of tb\_subtractor\_4bit is

--δηλωσεις σηματων

signal Sa, Sb: std\_logic\_vector(3 downto 0);

signal S\_result: std\_logic\_vector(3 downto 0);

signal S\_borrowout: std\_logic;

-- δηλωση component για την οντοτητα subtractor\_4bit

Component subtractor\_4bit

port (

a, b: in std\_logic\_vector(3 downto 0);

result: out std\_logic\_vector(3 downto 0);

borrowout: out std\_logic

);

end component subtractor\_4bit;

begin

--εναρξη τεστ

process

begin

-- περιπτωση 1

Sa <= "1010";

Sb <= "0101";

wait for 10 ns;

-- περιπτωση 2

Sa<= "1100";

Sb <= "0101";

wait for 10 ns;

wait;

end process ;

-- εναρξη τεστ

uut: subtractor\_4bit

port map (

a => Sa,

b => Sb,

result => S\_result,

borrowout => S\_borrowout

);

process

begin

wait for 20 ns;

--επαναληψη και ελεγχος σφαλματων τιμων των σηματων

while now < 100 ns loop

report "a = " & to\_string(Sa) &

", b = " & to\_string(Sb) &

", result = " & to\_string(S\_result) &

", borrowout = " & to\_string(S\_borrowout);

wait for 10 ns;

end loop;

wait;

end process;

end behavioral;

* *ΘΕΜΑ 6:*

ΕΡΩΤΗΜΑ Α:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωσης οντοτητας

entity dff\_a is

Port ( clk : in STD\_LOGIC;--ρολοι

d : in STD\_LOGIC;--δεδομενα

rst : in STD\_LOGIC;--συγχρονο reset

q : out STD\_LOGIC—έξοδος

);

end dff\_a;

--δηλωση αρχιτεκτονικης

architecture Behavioral of dff\_a is

begin

process(clk)--εναρξη διεργασιας για συγχρονο κυκλωμα

begin

if clk'event and clk='0' then –εναλλαγη ρολογιου, καθοδικη ακμη

if (rst='1') then

q<='0';

else

q<=d;

end if;

end if;

end process;

end Behavioral;

testbench

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity tb\_dff\_a is

end tb\_dff\_a;

--αρχιτεκτονικη testbench

architecture tb\_behavioral of tb\_dff\_a is

signal clk, rst, d, q : STD\_LOGIC := '0';

--δηλωση component για την οντοτητα dff\_a

component dff\_a

Port ( clk : in STD\_LOGIC;

d : in STD\_LOGIC;

rst : in STD\_LOGIC;

q : out STD\_LOGIC);

end component;

--εναρξη δοκιμης

begin

uut: dff\_a

port map (clk, d, rst, q);

-- διεργασια για το ρολόι

process

begin

clk <= '0';

wait for 5 ns;

clk <= '1';

wait for 5 ns;

end process;

-- διεργασια ελεγχου

process

begin

-- κανει reset

rst <= '1';

wait for 15 ns;

rst <= '0';

wait for 15 ns;

-- δινει τις τιμες των εισοδων

d <= '1';

wait for 10 ns;

d <= '0';

wait for 10 ns;

-- ανοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- καθοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- κανει ξανα reset

rst <= '1';

wait for 15 ns;

rst <= '0';

wait for 15 ns;

-- ανοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- καθοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- κανει ξανα reset

rst <= '1';

wait for 15 ns;

rst <= '0';

wait for 15 ns;

-- ανοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- καθοδικη ακμη ρολογιου

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

wait;

end process;

end tb\_behavioral;

ΕΡΩΤΗΜΑ Β:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity dff\_ce is

Port ( d : in STD\_LOGIC;

clk : in STD\_LOGIC;

rst : in STD\_LOGIC;--ασυγχρονο reset

ce : in STD\_LOGIC;--επιτρεψη

q : out STD\_LOGIC);

end dff\_ce;

--δηλωση αρχιτεκτονικης

architecture Behavioral of dff\_ce is

begin

process (clk, rst) begin—διεργασια για ασυγχρονο κυκλωμα

if (rst = '0') then

q <= '0';

elsif (clk'event and clk = '1') then—εναλλαγη ρολογιου, ανοδικη ακμη

if (ce = '1') then

q <= d;

end if;

end if;

end process;

end Behavioral;

*testbench*

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοτητας

entity tb\_dff\_ce is

end tb\_dff\_ce;

--δηλωση αρχιτεκτονικης

architecture Behavioral of tb\_dff\_ce is

signal clk, rst, ce, d, q : STD\_LOGIC := '0';--ενδιαμεσα λογικα σηματα, (αρχικοποιηση)

--δηλωση component για την οντοτητα dff\_ce

component dff\_ce is

Port ( d : in STD\_LOGIC;

clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

ce : in STD\_LOGIC;

q : out STD\_LOGIC);

end component;

--εναρξη δοκιμης flip-flop με επιτρεψη

begin

uut: dff\_ce

port map (d, clk, rst, ce, q);

-- διεργασια ρολογιου

process

begin

clk <= '0';

wait for 5 ns;

clk <= '1';

wait for 5 ns;

end process;

-- διεργασια ελεγχου

process

begin

-- κανει reset

rst <= '1';

wait for 15 ns;

rst <= '0';

wait for 15 ns;

-- επιτρεψη

ce <= '1';

-- δινει τις τιμες των εισοδων

d <= '1';

wait for 10 ns;

d <= '0';

wait for 10 ns;

-- τεστ ανοδικης ακμης , όταν η επιτρεψη είναι ενεργoποιημενη

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- τεστ ανοδικης ακμης, όταν η επιτρεψη είναι απενεργοποιημενη

ce <= '0';

wait for 10 ns;

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- ενεργοποιει το ρολοι ξανα

ce <= '1';

-- τεστ καθοδικης ακμης

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- reset ξανα

rst <= '1';

wait for 10 ns;

rst <= '0';

wait for 10 ns;

-- τεστ ανοδικης ακμης , όταν η επιτρεψη είναι ενεργoποιημενη

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- τεστ ανοδικης ακμης, όταν η επιτρεψη είναι απενεργοποιημενη

ce <= '0';

wait for 10 ns;

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

-- ενεργοποιει το ρολοι ξανα

ce <= '1';

-- τεστ καθοδικης ακμης

d <= '1';

wait for 5 ns;

d <= '0';

wait for 5 ns;

wait;

end process;

end Behavioral;

ΕΡΩΤΗΜΑ Γ:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

use IEEE.NUMERIC\_STD.ALL;

--δηλωση οντοτητας

entity counter\_4bit is

port (clk, rst : in std\_logic;

count : out unsigned(3 downto 0)—εξοδος 4 bit

);

end entity;

--δηλωση αρχιτεκτονικης

architecture behavioral of counter\_4bit is

signal counter: unsigned(3 downto 0);--σημα δυαδικου μετρητη 4 bit

begin

count <= counter;--εκχωρηση τρεχοντος τιμης του μετρητη στη θυρα εξοδου

process (clk, rst)—διεργασια ασυγχρονου κυκλωματος

begin

if rst = '1' then—ασυγχρονο reset

counter <= (others => '0');--μηδενισμος μετρητη

elsif clk'event and clk = '1' then—εναλλαγη ρολογιου,ανοδικης ακμης

if counter = 15 then—μεγιστη τιμη μετρητη :15

counter <= (others => '0');--μηδενισμος μετρητη

else

counter <= counter + 1;--αυξηση μετρητη

end if;

end if;

end process;

end behavioral;

testbench

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

use IEEE.NUMERIC\_STD.ALL;

--δηλωση οντοτητας

entity tb\_counter\_4bit is

end tb\_counter\_4bit;

--δηλωση αρχιτεκτονικης

architecture behavioral of tb\_counter\_4bit is

signal clk, rst : std\_logic := '0';--αρχικοποιηση σηματων

signal count : usigned(3 downto 0);--σημα εξοδου μετρητη 4 bit

--δηλωση component για την οντοτητα counter\_8bit

component counter\_4bit

port (

clk, rst : in std\_logic;

count : out unsigned(3 downto 0)

);

end component;

begin

-- δοκιμη μετρητη

uut: counter\_4bit

port map (clk, rst, count);

-- διεργασια ρολογιου

process

begin

clk <= '0';

wait for 5 ns;

clk <= '1';

wait for 5 ns;

end process;

-- διεργασια ελεγχου

process

begin

-- κανει reset

rst <= '1';

wait for 10 ns;

rst <= '0';

wait for 10 ns;

-- ελεγχος τιμης μετρητη μετα από μερικους κυκλους ρολογιου

wait for 20 ns;

wait for 20 ns;

wait;

end process;

end behavioral;

ΕΡΩΤΗΜΑ Δ:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

--δηλωση οντοητας

entity shift\_reg\_8bit is

port (clk, rst: in bit;

shift : in bit;

d\_out : out bit\_vector(7 downto 0)

);

end entity;

--δηλωση αρχιτεκτονικης

architecture behavioral of shift\_reg\_8bit is

signal reg8\_shift : bit\_vector(7 downto 0);--σημα 8 bit

begin

d\_out <= reg8\_shift;--εκχωρηση περιεχομενου του καταχωρητη στη θυρα του σηματος εξοδου

process (clk, rst) --διεργασια ασυγχρονου κυκλωματος

begin

if (rst='1') then --ασυγχρονο reset

reg8\_shift<= (others=>'0'); --μηδενισμος καταχωρητη

elsif (clk'event and clk='1') then—εναλλαγη ρολογιου, ανοδικη ακμη

reg8\_shift(7 downto 1) <= reg8\_shift(6 downto 0);--μετακινησηολων των bits αριστερα

reg8\_shift(0) <= shift;--εισαγωγη σηματος “shift” στο λιγοτερο σημαντικο bit

end if;

end process;

end behavioral;

testbench

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

--δηλωση οντοτητας

entity tb\_shift\_reg\_8bit is

end tb\_shift\_reg\_8bit;

--δηλωση αρχιτεκτονικης

architecture behavioral of tb\_shift\_reg\_8bit is

signal clk, rst, shift : bit := '0';-- αρχικοποιηση σηματων

signal d\_out : bit\_vector(7 downto 0) := (others => '0');--σημα εξοδου, αρχικοποιηση

signal stop\_sim: Boolean := false;-- δηλωση για την διακοπη της προσομοιωσης

-- δηλωση component για την οντοτητα shift\_reg\_8bit

component shift\_reg\_8bit

port (clk, rst : in bit;

shift : in bit;

d\_out : out bit\_vector(7 downto 0));

end component;

begin

-- δοκιμη καταχωρητη

shift\_reg\_inst : shift\_reg\_8bit

port map (

clk => clk,

rst => rst,

shift => shift,

d\_out => d\_out

);

-- διεργασια ρολογιου

process

begin

while not stop\_sim loop

clk<= not clk;--ενεργοποιηση σηματος ρολογιου

wait for 10 ns;

end loop;

wait;

end process;

-- διεργασια ελεγχου

process

begin

-- reset

rst <= '1';

wait for 10 ns;

rst <= '0';

-- τιμες εισοδου

shift <= '0';

wait for 25 ns;

shift<= '1';

wait for 25 ns;

shift<= '0';

wait for 20 ns;

-- σταματαει η προσομοιωση

stop\_sim <= true;

wait;

end process;

end behavioral;

* *ΘΕΜΑ 5*

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

use ieee.NUMERIC\_STD.all;

  --δηλωση οντοτητας

entity ar\_lo\_u is

     port(a,b: in std\_logic\_vector(7 downto 0);

            clk,rst,ce,d: std\_logic;

            in\_ ar\_lo\_u: in std\_logic\_vector(3 downto 0);

            out\_ ar\_lo\_u: out std\_logic\_vector(7 downto 0);

            carry\_out,q : out std\_logic

          );

end ar\_lo\_u;

--δηλωση αρχιτεκτονικης

architecture Behavioral of ar\_lo\_u is

    signal ar\_lo\_u \_res: std\_logic\_vector (7 downto 0);

    signal temp: std\_logic\_vector (8 downto 0);

begin

process (clk, rst) begin--διεργασια για ασυγχρονο κυκλωμα

    if (rst = '0') then

        q <= '0';

    elsif (clk'event and clk = '1') then--εναλλαγη ρολογιου, ανοδικη ακμη

        if (ce = '1') then

            q <= d;

        end if;

    end if;

end process;

--διεργασια ελεγχου

process(a,b, in\_ar\_lo\_u)

begin

    case(in\_ar\_lo\_u) is

          when "0000" => -- προσθεση

             out­\_ar\_lo\_u <= x + y ;

          when "0001" => -- αφαιρεση

             out­\_ar\_lo\_u <= x - y ;

          when "0010" => -- λογικο and

             out­\_ar\_lo\_u <= x and y;

          when "0011" => -- λογικο or

             out­\_ar\_lo\_u <= x or y;

          when "0100" => --λογικο not

             out­\_ar\_lo\_u <= not x;

          when "0101" => --λογικο not

             out­\_ar\_lo\_u <= not y;

          when others=>

                out­\_ar\_lo\_u <= "00000000";

      end case;

     end process;

out­\_ar\_lo\_u <= ar\_lo\_u\_res; --εξοδος του ALU

    temp <= ('0' & a) + ('0' & b);

    carry\_out <= temp(8); -- κρατουμενο

end Behavioral;

   
*--TESTBENCH*

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

  --δηλωση οντοτητας

entity tb\_ar\_lo\_u is

end tb\_ar\_lo\_u;

--δηλωση αρχιτεκτονικης

architecture Behavioral of tb\_ar\_lo\_u is

component ar\_lo\_u

        port(x,y: in std\_logic\_vector(7 downto 0);

            clk,rst,ce,d: std\_logic;

            in\_ar\_lo\_u: in std\_logic\_vector(3 downto 0);

           out­\_ar\_lo\_u: out std\_logic\_vector(7 downto 0);

            carry\_out,q : out std\_logic

          );

end component;

  --σηματα

   signal x : std\_logic\_vector(7 downto 0) := (others => '0');

   signal y : std\_logic\_vector(7 downto 0) := (others => '0');

   signal in\_ar\_lo\_u: std\_logic\_vector(3 downto 0) := (others => '0');

    signal out­\_ar\_lo\_u : std\_logic\_vector(7 downto 0);

    signal carry\_out : std\_logic;

    signal clk, rst, ce, d, q : STD\_LOGIC := '0';

    signal i:integer;

begin

-- test

   uut: alu PORT MAP (

          x=> x,

          y => y,

          in­­\_ar\_lo\_u => in\_ar\_lo\_u,

          clk=>clk,

          rst=>rst,

          d=>d,

          ce=>ce,

          q=>q,

          out­\_ar\_lo\_u => out­\_ar\_lo\_u,

          carry\_out => carry\_out

        );

--διεργασια ρολογιου

  process

  begin

        clk <= '0';

        wait for 10 ns;

        clk <= '1';

        wait for 10 ns;

    end process

--reset

process

    begin

         rst <= '1';

        wait for 20 ns;

         rst <= '0';

        wait for 20 ns;

end process;

   process

   begin

--επαναληψη από 0 εως 15

        for i in 0 to 15 loop

            in­\_ar\_lo\_u <= in\_ar\_lo\_u + 1;

         wait for 100 ns;

        end loop;

        end process;

end Behavioral; 