Σχεδίαση Ψηφιακών Συστημάτων

ΕΞΑΜΗΝΙΑΙΑ ΑΣΚΗΣΗ ΘΕΩΡΙΑΣ

ΜΕΡΟΣ 05

Τμήμα Μηχανικών Πληροφορικής

& Υπολογιστών

(Καθηγητής: Ιωάννης Βογιατζής)

Μάρκος Παντελιδάκης - 18390228 - ice18390228@uniwa.gr

1. Καταχωρητής 4-bits

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3   ENTITY reg4 IS PORT (
4       D: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
5       Resetn, Clock: IN STD_LOGIC;
6       Q: OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
7   END reg4;
8   ARCHITECTURE behavioral OF reg4 IS
9   BEGIN
10      PROCESS (Resetn, Clock)BEGIN
11          IF Resetn = '0' THEN
12              Q <= "0000" ;
13          ELSIF rising_edge(Clock) THEN
14              Q <= D;
15          END IF;
16      END PROCESS;
17  END behavioral;
```

Εικ5.1: Κώδικας αρχείου VHDL κυκλώματος καταχωρητή 4-bits.

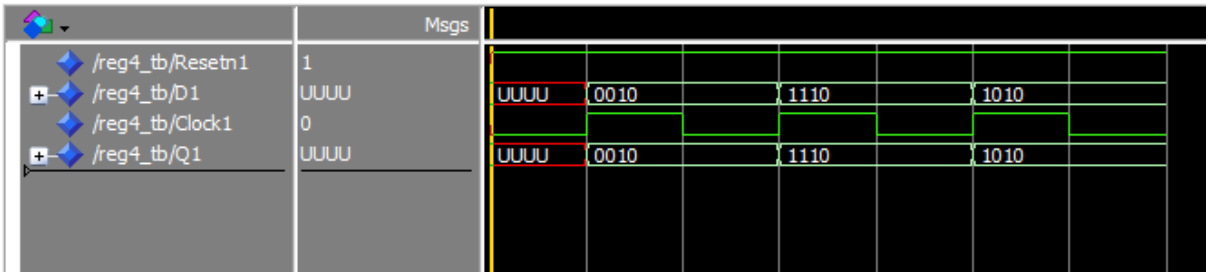
```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity reg4_tb is
5  end reg4_tb;
6
7  architecture test_b of reg4_tb is
8  signal D1 : STD_LOGIC_VECTOR(3 DOWNTO 0);
9  signal Resetn1, Clock1 : STD_LOGIC;
10 signal Q1 : STD_LOGIC_VECTOR(3 DOWNTO 0);
11
12 component reg4 PORT(
13         D:                  IN STD_LOGIC_VECTOR(3 DOWNTO 0);
14         Resetn, Clock: IN STD_LOGIC;
15         Q:                  OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
16 end component;
17
18 BEGIN
19 M1 : reg4 PORT MAP(D => D1, Resetn => Resetn1, Clock => Clock1, Q => Q1);
20 process
21 begin
22         Resetn1 <= '1';
23         Clock1 <= '0';
24         wait for 50 ps;
25
26         D1 <= "0010";
27         Clock1 <= '1';
28         wait for 50 ps;
29         Clock1 <= '0';
30         wait for 50 ps;
31
32         D1 <= "1110";
33         Clock1 <= '1';
34         wait for 50 ps;
35         Clock1 <= '0';
36         wait for 50 ps;
37
38         D1 <= "1010";
39         Clock1 <= '1';
40         wait for 50 ps;
41         Clock1 <= '0';
42         wait for 50 ps;
43 end process;
44 end test_b;
```

Εικ5.1.2: Κώδικας αρχείου VHDL testbench κυκλώματος καταχωρητή 4-bits.

Πίν5.1.1: Τιμές εισόδου-εξόδου προγράμματος.

D	Q
0010	0010
1110	1110
1010	1010

Για την έξοδο Q οι τιμές της είναι ίδιες με της εισόδου και αυτό επαληθεύεται από την κυμματομορφή της εικόνας 5.1.3.



Εικ5.1.3: Αναπαράσταση λειτουργίας σε κυματομορφή κυκλώματος καταχωρητή 4-bits.

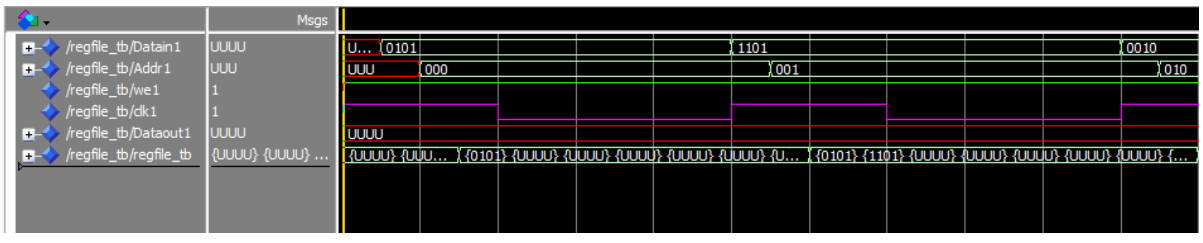
2. Αρχείο 8 καταχωρητών 4-bits

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  USE ieee.std_logic_unsigned.all;
4  USE ieee.numeric_std.ALL;
5
6  entity regfile is       port (
7      Datain  : in  std_logic_vector(3 downto 0);
8      Addr    : in  std_logic_vector(2 downto 0);
9      we      : in  std_logic;
10     clk     : in  std_logic;
11     Dataout : out std_logic_vector(3 downto 0));
12 end regfile;
13
14 architecture behavioural of regfile is
15 type regArray is array(0 to 7) of std_logic_vector(3 downto 0);
16 signal regfile: regArray;
17
18 BEGIN
19 process(clk)
20 begin
21 if(clk'event and clk='1') then
22         if we='1' then
23                 regfile(to_integer(unsigned(Addr))) <= Datain;
24         else
25                 Dataout <= regfile(to_integer(unsigned(Addr)));
26         end if;
27 end if;
28 end process;
29 end behavioural;
```

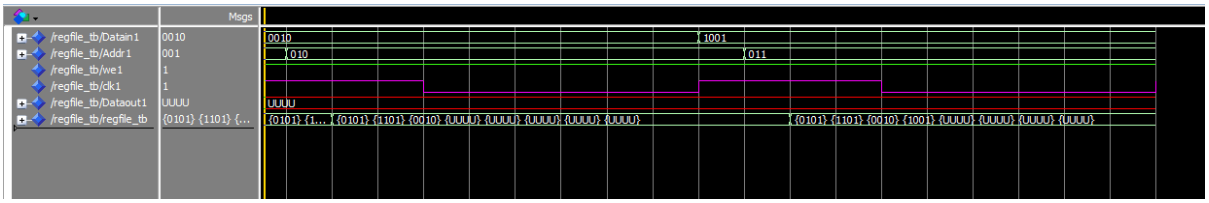
Εικ5.2.1: Κώδικας αρχείου VHDL κυκλώματος Register file.

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  USE ieee.std_logic_unsigned.all;
4  USE ieee.numeric_std.ALL;
5
6  entity regfile_tb is
7  end regfile_tb;
8
9  architecture test_b of regfile_tb is
10 component regfile port(
11         Datain  : in  std_logic_vector(3 downto 0);
12     Addr    : in  std_logic_vector(2 downto 0);
13     we      : in  std_logic;
14     clk     : in  std_logic;
15     Dataout : out std_logic_vector(3 downto 0));
16 end component;
17
18 signal Datain1 : std_logic_vector(3 downto 0);
19 signal Addr1    : std_logic_vector(2 downto 0);
20 signal we1      : std_logic;
21 signal clk1     : std_logic;
22 signal Dataout1 : std_logic_vector(3 downto 0);
23 constant ClockPeriod: time:= 10 ps;
24 type regArray is array(0 to 7) of std_logic_vector(3 downto 0);
25 signal regfile_tb : regArray;
26
27 BEGIN
28 M1 : regfile port map(Datain => Datain1, Addr => Addr1, we => we1, clk => clk1, Dataout => Dataout1);
29 CLK1<= not CLK1 after ClockPeriod/2;
30 process
31 begin
32         we1 <= '1'; wait for 10 ps;
33         Datain1 <= "0101"; wait for 10 ps;
34         Addr1 <= "000"; wait for 10 ps;
35         regfile_tb(to_integer(unsigned(Addr1))) <= Datain1;
36         wait for 70 ps;
37
38         Datain1 <= "1101"; wait for 10 ps;
39         Addr1 <= "001"; wait for 10 ps;
40         regfile_tb(to_integer(unsigned(Addr1))) <= Datain1;
41         wait for 80 ps;
42
43         Datain1 <= "0010"; wait for 10 ps;
44         Addr1 <= "010"; wait for 10 ps;
45         regfile_tb(to_integer(unsigned(Addr1))) <= Datain1;
46         wait for 80 ps;
47
48         Datain1 <= "1001"; wait for 10 ps;
49         Addr1 <= "011"; wait for 10 ps;
50         regfile_tb(to_integer(unsigned(Addr1))) <= Datain1;
51         wait for 80 ps;
52 end process;
53 end test_b;
```

Εικ5.2.2: Κώδικας αρχείου VHDL testbench κυκλώματος Register file.



Εικ5.2.3: Αναπαράσταση εγγραφής δεδομένων σε κυματομορφή σε Register file.



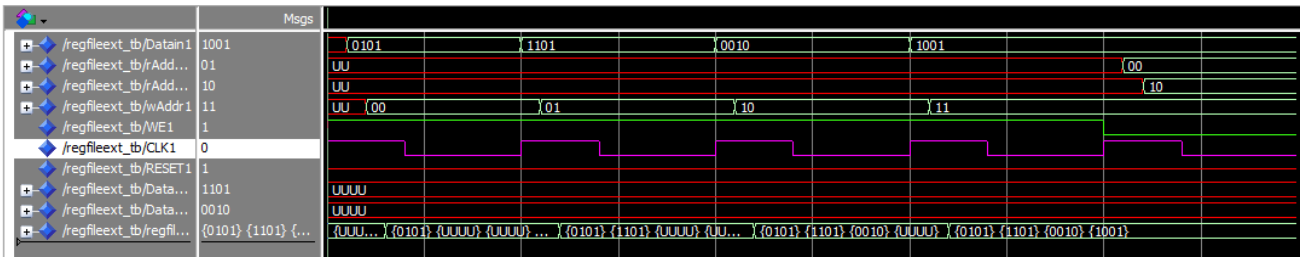
Εικ5.2.4: Αναπαράσταση ανάγνωσης δεδομένων σε κυματομορφή από Register file.

```
1   -- Ασκηση 5.4 Register File Extended
2   library ieee;
3   use ieee.std_logic_1164.all;
4   USE ieee.std_logic_unsigned.all;
5   USE ieee.numeric_std.ALL;
6
7   entity regfileEXT is
8   generic (dw : natural:= 4; size : natural:= 4; adrw : natural:= 2);
9   port(
10          Datain : in  std_logic_vector(dw-1 downto 0);
11          rAddr1 : in  std_logic_vector(adrw-1 downto 0);
12          rAddr2 : in  std_logic_vector(adrw-1 downto 0);
13          wAddr  : in  std_logic_vector(adrw-1 downto 0);
14          we     : in  std_logic;
15          clk    : in  std_logic;
16          reset  : in  std_logic;
17          Dataout1 : out std_logic_vector(dw-1 downto 0);
18          Dataout2 : out std_logic_vector(dw-1 downto 0));
19  end regfileEXT;
20
21  ARCHITECTURE behavioural OF regfileEXT IS
22  type regArray is array(0 to size-1) of std_logic_vector(dw-1 downto 0);
23  signal regfile : regArray;
24
25  BEGIN
26  process(clk)
27  begin
28  if (clk'event and clk='1') then
29          if reset ='1' then
30                  for i IN 0 TO dw-1 loop
31                          Dataout1(i)<='0';
32                          Dataout2(i)<='0';
33                  end loop;
34          else
35                  if we='1' then
36                          regfile(to_integer(unsigned(wAddr))) <= Datain;
37                  else
38                          Dataout1 <= regfile(to_integer(unsigned(rAddr1)));
39                          Dataout2 <= regfile(to_integer(unsigned(rAddr2)));
40                  end if;
41          end if;
42  end if;
43  end process;
44  end behavioural;
```

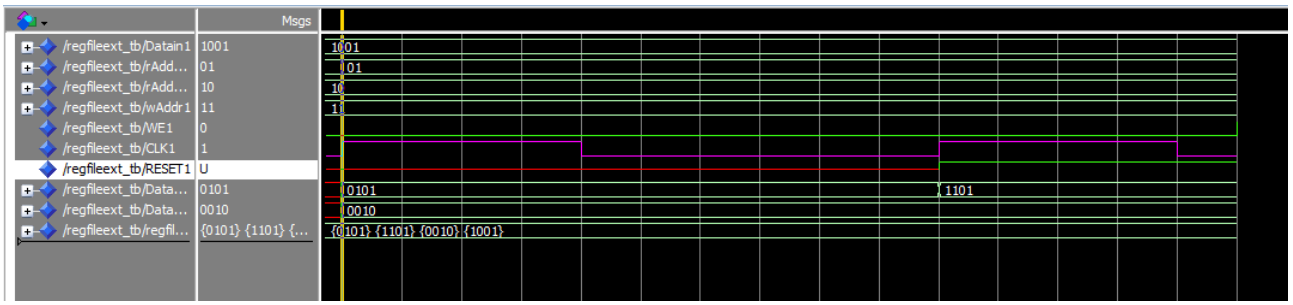
Εικ5.3.1: Κώδικας αρχείου VHDL κυκλώματος Register file με 2 θύρες ανάγνωσης και 1 εγγραφής.

```
-- Ασκηση 5.5 Register File Extended Testbench
library ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

entity regfileEXT_tb IS
generic ( dw : natural := 4; size : natural := 4; adrw : natural := 2);
end regfileEXT_tb;

ARCHITECTURE test_b OF regfileEXT_tb IS

component regfileEXT PORT(
        Datain : in std_logic_vector(dw-1 downto 0);
        rAddr1: in std_logic_vector(adrw-1 downto 0);
        rAddr2: in std_logic_vector(adrw-1 downto 0);
        wAddr : in std_logic_vector(adrw-1 downto 0);
        we : in std_logic;
        clk : in std_logic;
        reset : in std_logic;
        Dataout1 : out std_logic_vector(dw-1 downto 0);
        Dataout2 : out std_logic_vector(dw-1 downto 0));
end component;

signal Datain1: std_logic_vector(dw-1 downto 0);
signal rAddr11: std_logic_vector(adrw-1 downto 0);
signal rAddr21: std_logic_vector(adrw-1 downto 0);
signal wAddr1: std_logic_vector(adrw-1 downto 0);
signal WE1: std_logic;
signal CLK1: std_logic;
signal RESET1: std_logic;
signal Dataout11: std_logic_vector(dw-1 downto 0);
signal Dataout21: std_logic_vector(dw-1 downto 0);
constant ClockPeriod: time:= 50 ps;
type regArray is array(0 to size-1) of std_logic_vector(dw-1 downto 0);
signal regfileEXT_tb : regArray;

BEGIN
M: regfileEXT PORT MAP(Dataout2=>Dataout21, Dataout1=>Dataout11, Datain=>Datain1, rAddr1=>rAddr11, rAddr2 => rAddr21,
wAddr=>wAddr1, we=>WE1, clk=>CLK1, reset => RESET1);
CLK1<= not CLK1 after ClockPeriod/2;
process
begin
        WE1<='1'; wait for 10 ps;
        Datain1<="0101"; wait for 10 ps;
        wAddr1<="00"; wait for 10 ps;
        regfileEXT_tb(to_integer(unsigned(wAddr1))) <= Datain1;
        wait for 70 ps;

        Datain1<="1101"; wait for 10 ps;
        wAddr1<="01"; wait for 10 ps;
        regfileEXT_tb(to_integer(unsigned(wAddr1))) <= Datain1;
        wait for 80 ps;

        Datain1<="0010"; wait for 10 ps;
        wAddr1<="10"; wait for 10 ps;
        regfileEXT_tb(to_integer(unsigned(wAddr1))) <= Datain1;
        wait for 80 ps;

        Datain1<="1001"; wait for 10 ps;
        wAddr1<="11"; wait for 10 ps;
        regfileEXT_tb(to_integer(unsigned(wAddr1))) <= Datain1;
        wait for 80 ps;

        WE1<='0'; wait for 10 ps;
        rAddr11<="00"; wait for 10 ps;
        rAddr21<="10"; wait for 10 ps;
        wait for 70 ps;

        rAddr11<="01"; wait for 10 ps;
        rAddr21<="10"; wait for 10 ps;
        wait for 80 ps;

        RESET1<='1';
        wait for 50 ps;
end process;
end test_b;
```

Εικ5.3.2: Κώδικας αρχείου VHDL testbench κυκλώματος Register file με 2 θύρες ανάγνωσης και 1 εγγραφής.



Εικ5.3.3: Αναπαράσταση εγγραφής δεδομένων σε κυματομορφή σε Register file με 2 θύρες ανάγνωσης και 1 εγγραφής.



Εικ5.2.4: Αναπαράσταση ανάγνωσης δεδομένων σε κυματομορφή από Register file με 2 θύρες ανάγνωσης και 1 εγγραφής.