Αναφορά 1ης Εργαστηριακής Άσκησης

Ομάδα Β 10

Νικόλαος Τζιρής 0313425 Σταυρογιαννης Νεκταριος-Χρηστος 03113555 Κατσαρος Περικλης 03113302

A.2

Υλοποίηση με αρχιτεκτονική dataflow:

```
1 library IEEE;
 2 use IEEE.STD LOGIC 1164.ALL;
 4 entity dec8 df is
      Port ( enc : in STD LOGIC VECTOR (2 downto 0);
               dec : out STD LOGIC VECTOR (7 downto 0));
 7 end dec8 df;
 9 architecture Dataflow of dec8 df is
11 signal not enc : STD LOGIC VECTOR (2 downto 0);
12
13 begin
14
15 not_enc(0) <= not enc(0);
16 not enc(1) <= not enc(1);
17 not_enc(2) <= not enc(2);
18
19 dec(0) <= not_enc(2) and not_enc(1) and not_enc(0);
20 dec(1) <= not enc(2) and not enc(1) and enc(0);
21 dec(2) <= not_enc(2) and enc(1) and not_enc(0);
22 dec(3) <= not_enc(2) and enc(1) and enc(0);
23 dec(4) <= enc(2) and not enc(1) and not enc(0);
24 \operatorname{dec}(5) \le \operatorname{enc}(2) and not \operatorname{enc}(1) and \operatorname{enc}(0);
25 dec(6) <= enc(2) and enc(1) and not enc(0);
26 dec(7) <= enc(2) and enc(1) and enc(0);
27
28 end Dataflow;
```

Υλοποίηση με αρχιτεκτονική behavioral:

```
1 library IEEE;
 2 use IEEE.STD LOGIC 1164.ALL;
4 entity dec8 be is
     Port ( enc : in STD LOGIC VECTOR (2 downto 0);
             dec : out STD LOGIC VECTOR (7 downto 0));
7 end dec8 be;
9 architecture Behavioral of dec8 be is
10
11 begin
12
   process (enc) is
13
    begin
14
        case enc is
15
            when "000" => dec <= "00000001";
16
             when "001" => dec <= "00000010";
17
             when "010" => dec <= "00000100";
18
             when "011" => dec <= "00001000";
19
             when "100" => dec <= "00010000";
20
             when "101" => dec <= "00100000";
21
             when "110" => dec <= "01000000";
             when "111" => dec <= "10000000";
22
23
             when others => dec <= "000000000";
24
        end case;
25 end process;
26
27 end Behavioral;
```

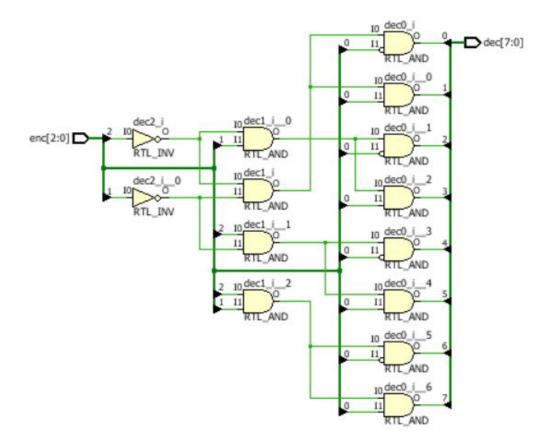
Υλοποίηση testbench για την αρχιτεκτονική dataflow:

```
1 library IEEE;
 2 use IEEE.STD LOGIC 1164.ALL;
 3 use ieee.numeric std.all;
 5 entity dec8 df tb is
 6 end dec8_df_tb;
8 architecture Behavioral of dec8_df_tb is
10 component dec8_df
11
         Port ( enc : in STD LOGIC VECTOR (2 downto 0);
                dec : out STD LOGIC VECTOR (7 downto 0));
12
13
   end component;
14
15
     signal enc: STD LOGIC VECTOR (2 downto 0);
16 signal dec: STD_LOGIC_VECTOR (7 downto 0);
17
     constant CLOCK_PERIOD : time := 10 ns;
18
19 begin
20
    uut: dec8_df port map ( enc => enc,
21
                              dec => dec );
22
23
     stimulus: process
24
     begin
25
          for i in 0 to 7 loop
26
              enc <= std logic vector(to unsigned(i, 3));
27
              wait for CLOCK PERIOD;
28
          end loop;
29
          wait;
30 end process;
31 end Behavioral;
```

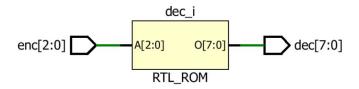
Υλοποίηση testbench για **behavioral**:

```
1 library IEEE;
 2 use IEEE.STD LOGIC 1164.ALL;
 3 use ieee.numeric std.all;
 5 entity dec8_be_tb is
 6 end dec8 be tb;
 8 architecture Behavioral of dec8_be_tb is
10
     component dec8_be
11
       Port ( enc : in STD LOGIC VECTOR (2 downto 0);
12
               dec : out STD LOGIC VECTOR (7 downto 0));
13
     end component;
14
15
     signal enc: STD LOGIC VECTOR (2 downto 0);
     signal dec: STD LOGIC VECTOR (7 downto 0);
16
17
18
     constant CLOCK_PERIOD : time := 10 ns;
19
20 begin
21
22
    uut: dec8_be port map ( enc => enc,
23
                             dec => dec );
24
25
     stimulus: process
26
    begin
27
         for i in 0 to 7 loop
28
              enc <= std_logic_vector(to_unsigned(i, 3));
29
             wait for CLOCK PERIOD;
30
         end loop;
31
         wait;
    end process;
33 end Behavioral;
```

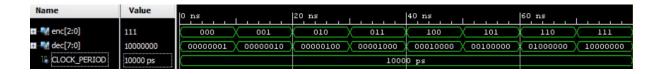
RTL σχήμα για την αρχιτεκτονική dataflow:



RTL σχήμα για την αρχιτεκτονική behavioral:



Η προσομοίωση και για τις δύο αρχιτεκτονικές δίνει τα ίδια αποτελέσματα. Αυτό είναι αναμενόμενο γιατί και οι δύο περιγράφουν ένα δυαδικό αποκωδικοποιητή 3 σε 8. Παρακάτω το αποτέλεσμα της προσομοίωσης:

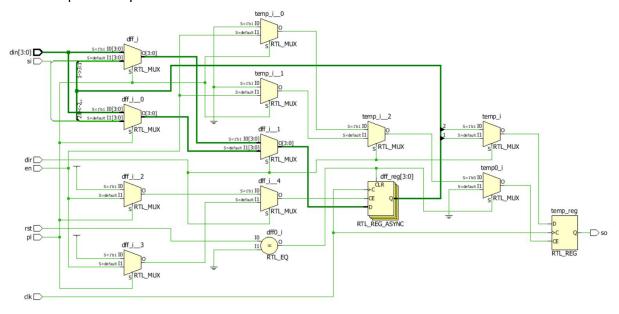


B2

Υλοποίηση:

```
22 library IEEE;
23 use IEEE.STD LOGIC 1164.ALL;
24 -- Uncomment the following library declaration if using
25 -- arithmetic functions with Signed or Unsigned values
26 -- use IEEE.NUMERIC STD.ALL;
27
28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx leaf cells in this code.
30 -- library UNISIM;
31 -- use UNISIM. VComponents.all;
32 entity shift reg3 is
33
   port (
   clk, rst, si, en, pl: in std logic;
34
35
   dir: in std logic;
    din: in std logic vector(3 downto 0);
   so: out std logic);
38 end shift_reg3;
39
40 architecture Behavioral of shift_reg3 is
   signal dff: std logic vector(3 downto 0);
     signal temp: std logic := '0';
42
43 begin
44
    edge: process (clk,rst)
45
    begin
46
      if rst='0' then
47
           dff <= (others=>'0');
      elsif clk'event and clk='1' then
48
49
          if dir='1' then
50
              if pl='1' then
51
                 dff <= din;
              elsif en='1' then
52
53
                 temp <= dff(3);
54
                 dff <= dff(2 downto 0)&si;
55
              end if;
56
        else
57
             if pl='1' then
58
                dff <= din;
59
            elsif en='l' then
              temp <= dff(0);
60
              dff <= si&dff(3 downto 1);
61
62
             end if:
63
           end if;
64
        end if;
65
     end process;
      --so <= dff(3) when dir = '1' else dff(0);
66
      so <= temp;
68 end Behavioral;
```

Το κύκλωμα που προκύπτει είναι:



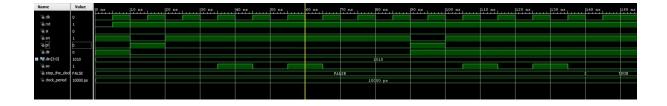
οι διαφορές με το σχήμα 2.8 της εκφώνησης είναι ότι έχουμε περισσότερους multiplexers. Αυτό είναι αναμενόμενο καθώς έχουμε υλοποιήσει και την δεξιά ολίσθηση.

Υλοποίηση testbench:

```
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using
26 -- arithmetic functions with Signed or Unsigned values
27 -- use IEEE.NUMERIC STD.ALL;
28
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 -- library UNISIM;
32 -- use UNISIM. VComponents.all;
33
34 entity shift_reg3_tb is
35 end shift reg3 tb;
37 architecture bench of shift reg3 tb is
38
39 component shift_reg3
40
     port (
41
      clk, rst, si, en, pl: in std logic;
42
      dir: in std logic;
      din: in std logic vector (3 downto 0);
43
44
      so: out std logic);
45 end component;
46
47 signal clk, rst, si, en, pl: std logic;
48 signal dir: std logic;
49 signal din: std logic vector(3 downto 0);
50 signal so: std logic;
51
52 constant clock_period: time := 10 ns;
53 signal stop_the_clock: boolean;
54
55 begin
56
    uut: shift reg3 port map ( clk => clk,
58
                               rst => rst,
59
                               si => si,
60
                               en => en,
                               pl => pl,
61
62
                               dir => dir,
63
                               din => din,
64
                               so => so );
65
```

```
66 stimulus: process
 67 begin
 68
 69
      -- Put initialisation code here
 70
    si <= '0';
     din <= "1010";
 71
 72
    dir <= '1';
 73 en <= '1';
 74
      pl <= '0';
 75
     rst <= '0';
 76
 77
     wait for 5 ns;
 78
     rst <= '1';
    wait for 5 ns;
 79
 80
 81
      -- Put test bench stimulus code here
      -- check for dir='0' and we expect right shift
 83
     dir <= '0';
 84
      pl <= '1';
 85
 86
     en <= '0';
 87
     wait for clock period;
 88
    pl <= '0';
 89
      en <= '1';
     wait for 7 * clock_period;
 90
 91
 92
     -- check for dir='1' and we expect left shift
 93
     dir <= '1';
 94
    pl <= '1';
     en <= '0';
 95
 96
      wait for clock period;
 97
     pl <= '0';
    en <= '1';
 98
99
      wait for 4 * clock period;
100
101 stop_the_clock <= true;</pre>
102
     wait;
103 end process;
104
105 clocking: process
106 begin
     while not stop_the_clock loop
107
       clk <= '0', '1' after clock_period / 2;
108
109
       wait for clock period;
110
    end loop;
111
     wait;
112 end process;
113 end bench;
```

Προσομοίωση:



B3

Υλοποίηση:

```
l library IEEE;
 2 use IEEE.std logic 1164.all;
 3 use IEEE.std logic unsigned.all;
4 use IEEE.numeric_std.all;
 6 entity count3 is
     port( clk,
            up.
9
            resetn,
           count en : in std logic;
11
           modulo : in std_logic_vector(2 downto 0);
12
            sum : out std logic vector(2 downto 0);
13
           cout : out std logic);
14 end;
15
16 architecture rtl_limit of count3 is
17
    signal count : std logic vector(2 downto 0);
     --constant LIMIT : integer := 5;
18
19 begin
20
     process(clk, resetn)
21
22
         if resetn='0' then
23
          -- Ασύγχρονος μηδενισμός
24
             count <= (others=>'0');
25
         elsif clk'event and clk='l' then
26
              if count_en = 'l' then
27
                  case up is
28
                      when '1' =>
                         if count/=modulo then
29
30
                              count <= count + 1;
31
32
                              count <= (others => '0');
33
                          end if;
                      when '0' =>
34
35
                         if count/=0 then
36
                              count <= count - 1;</pre>
37
38
                             count <= modulo:
39
                          end if;
40
                      when others =>
41
                          count <= (others => '0');
42
43
               end if;
44
         end if;
45 end process;
46 sum <= count;
47 cout <= '1' when (count=modulo and count_en='1' and up='1') or (count=0 and count_en='1' and up='0') else '0';
48 end;
```

Υλοποίηση testbench:

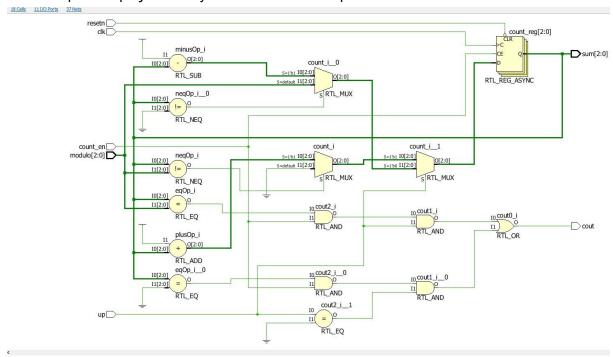
```
21 library IEEE;
22 use IEEE.Std logic 1164.all;
23 use IEEE.Numeric Std.all;
25 entity count3 tb is
26 end;
27
28 architecture bench of count3 tb is
29 component count3
30
      port ( clk,
31
            up,
32
            resetn,
33
            count_en : in std logic;
34
            modulo : in std logic vector(2 downto 0);
35
            sum : out std logic vector(2 downto 0);
36
             cout : out std logic);
37 end component;
38
39 signal clk, up, resetn, count_en: std logic;
40 signal modulo: std logic vector(2 downto 0);
41 signal sum: std logic vector(2 downto 0);
42 signal cout: std logic;
43
44 constant clock period: time := 10 ns;
45 signal stop_the_clock: boolean;
46
47 begin
48
49 uut: count3 port map ( clk => clk,
                         up => up,
50
                         resetn => resetn,
51
52
                         count_en => count_en,
53
                         modulo => modulo,
                                => sum,
54
                         sum
55
                         cout => cout );
56
57 stimulus: process
58 begin
   -- Put initialisation code here
59
60 up <= '0';
61
    count en <= '0';
    resetn <= '0';
62
63 wait for 5 ns;
64 resetn <= '1';
65 wait for 5 ns;
```

```
67
    -- Put test bench stimulus code here
68
69
    modulo <= "101";
70
71 -- up = 1
72 up <= '1';
73
     count_en <= '1';
74
     wait for 10 * clock_period;
75
76
    -- test count en
77
   count en <= '0';
78 wait for 3 * clock_period;
79
    -- up = 0
80
    up <= '0';
81
82
    count_en <= '1';
83 wait for 10 * clock_period;
84
85
86 stop_the_clock <= true;</pre>
87
   wait;
88 end process;
89
90 clocking: process
91 begin
    while not stop_the_clock loop
92
93 clk <= '0', '1' after clock_period / 2;
      wait for clock_period;
95
   end loop;
96
     wait;
97 end process;
99 end bench;
```

Προσομοίωση:

Name	Value	0 ns 2	0 ns .	40 ns .	60 ns	80 ns .	100 ns	120 ns	140 ns	160 ns	180 ns	200 ns	220 ns	240 ns	260 ns
ી∳ dk															****
Un up															
Ve resetn															
1 count_en															
	101	עטט							—						
	011		(010	011 (100	X 101 X 000 X 0	01 / 010 / 0	1 ×	100	V 011 V 0	0 (001) 0	00 × 101 × 10	0 / 011 / 0	0 / 001 / 0	00 X	101
	0					- (333)(3									
In stop_the_dock	FALSE						FA	LSE							TRUE
16 dock_period									10000 p						
															_

Το κύκλωμα που μας δίνει ο synthesizer είναι το παρακάτω:



Παρατηρούμε οτι θα μπορούσε να σχεδιαστεί απλούστερα με το χέρι με τη χρήση 3 flip flops και ίσως μερικών mux για τη λογική up/down.