

1^η Εργαστηριακή Άσκηση
ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΗ ΓΛΩΣΣΑ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ VHDL ΚΑΙ ΤΗΝ
ΙΕΡΑΡΧΙΚΗ ΣΧΕΔΙΑΣΗ (STRUCTURAL VHDL)
3/3/2017

Ομάδα LAB20332061

ΛΑΜΠΡΙΑΝΗ ΤΣΑΠΑΝΟΥ 2014030015

ΑΝΑΣΤΑΣΙΟΣ ΜΠΟΚΑΛΙΔΗΣ 2014030069

Σκοπός εργαστηριακής άσκησης

Σκοπός της 1^{ης} εργαστηριακής άσκησης ήταν η πρώτη επαφή και εξοικείωση με την γλώσσα VHDL, όπως επίσης και με το εργαλείο Xilinx.

Προεργασία – Περιγραφή

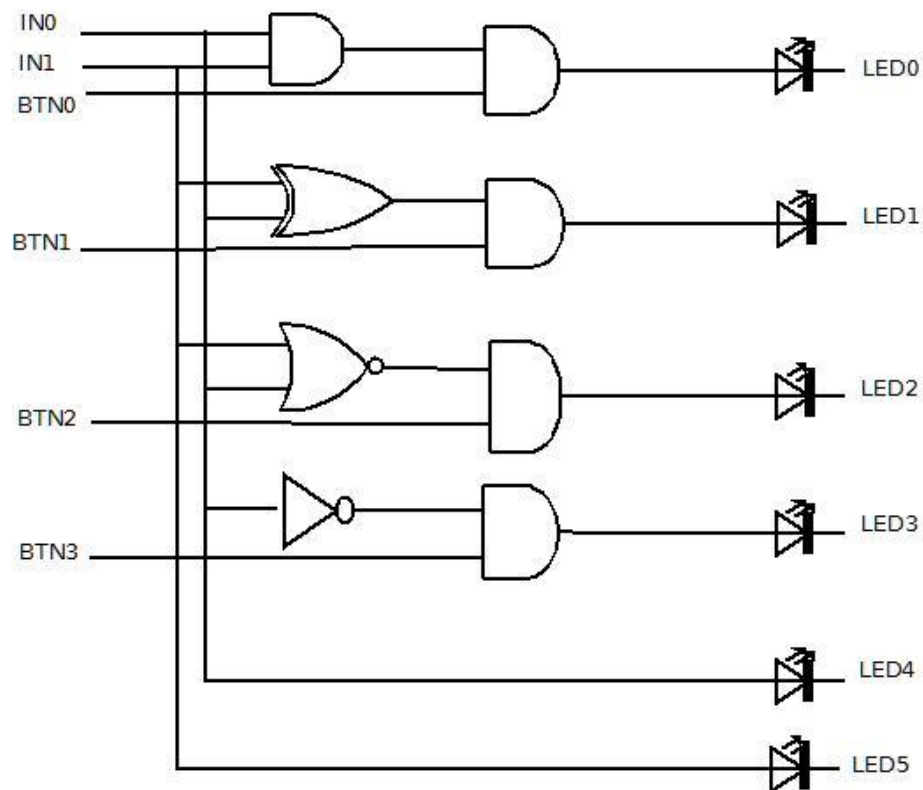
1^ο Κύκλωμα

Το 1^ο κύκλωμα που μας ζητήθηκε να υλοποιήσουμε έχει 6 εισόδους (4 από τις οποίες είναι τύπου BTN) και 6 εξόδους LED. Οι τιμές που παίρνουν αυτές οι μεταβλητές φαίνονται παρακάτω:

- LED0 = IN0 **AND** IN1 ανBTN0 = 1
LED0 = 0 ανBTN0 = 0
- LED1 = IN0 **XOR** IN1 ανBTN1 = 1
LED1 = 0 ανBTN1 = 0
- LED2 = IN0 **NOR** IN1 ανBTN2 = 1
LED2 = 0 ανBTN2 = 0
- LED3 = **NOT** IN0 ανBTN3 = 1
LED3 = 0 ανBTN3 = 0
- LED4 = IN0
- LED5 = IN1

Έτσι οδηγούμαστε στο παρακάτω λογικό διάγραμμα του κυκλώματος.

Διάγραμμα



2^ο Κύκλωμα

Το 2^ο κύκλωμα που μας ζητήθηκε να υλοποιήσουμε έναν ημιαθροιστή (Half Adder) με την χρήση λογικών πυλών.

Half Adder

Πίνακας Αληθείας

A	B	SUM	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πίνακες Karnaugh

Sum :

A/B	0	1
0	0	1
1	1	0

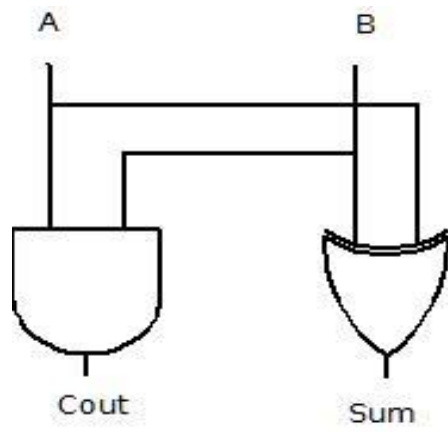
Sum = $AB' + A'B = A \text{ XOR } B$ (Λογική Πύλη XOR)

Cout :

A/B	0	1
0	0	0
1	0	1

Cout = $A * B$ (Λογική Πύλη AND)

Διάγραμμα Half Adder



Έπειτα υλοποιήσαμε έναν πλήρη αθροιστή (Full Adder) χρησιμοποιώντας τον ημιαθροιστή ως υποκύκλωμα.

Full Adder

Πίνακας Αληθείας

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Πίνακες Karnaugh

Sum :

CIN/AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$\text{Sum} = \text{Cin } A' B' + \text{Cin}' A' B + \text{Cin } A B + \text{Cin}' A B'$$

$$= \text{Cin}(A' B' + A B) + \text{Cin}' (A' B + A B')$$

$$= \text{Cin}(A' B + A B')' + \text{Cin}' (A' B + A B')$$

$$= (A \text{ XOR } B) \text{ XOR } \text{Cin}$$

Cout :

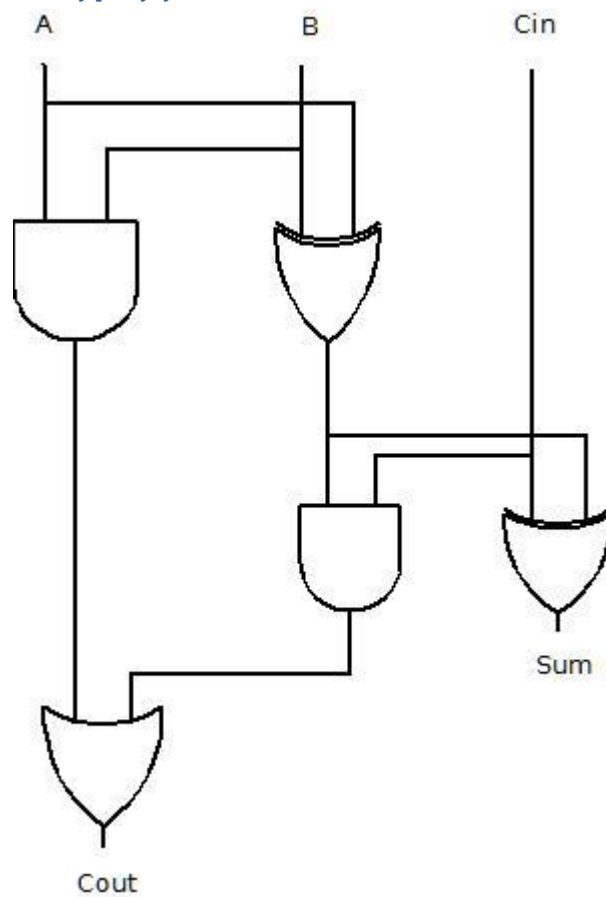
CIN/AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$\text{Cout} = \text{Cin } A' B + A B + \text{Cin } A B'$$

$$= \text{Cin } (A' B + A B') + A B$$

$$= AB + \text{Cin } (A \text{ XOR } B)$$

Διάγραμμα Full Adder

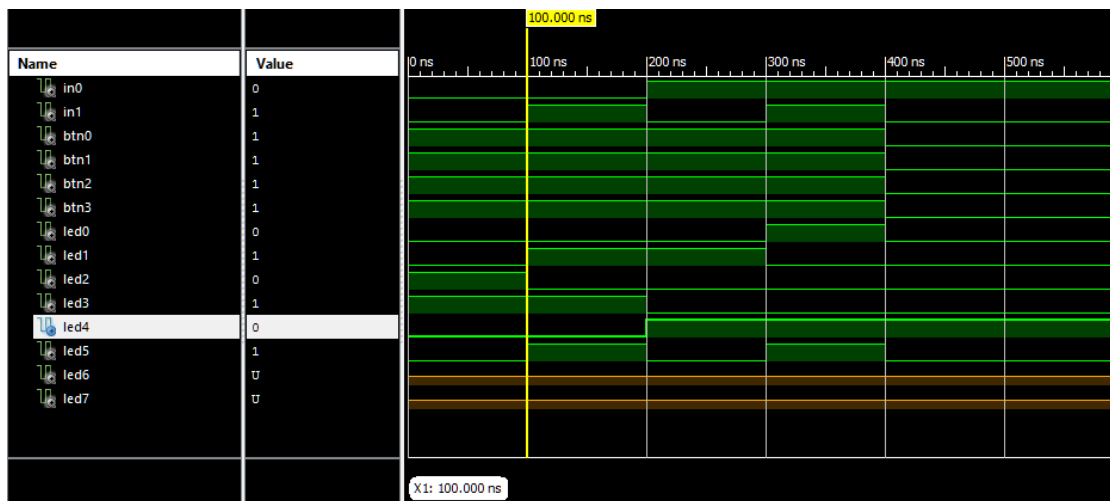


Κυματομορφές-Προσομοίωση

1^ο Κύκλωμα

Στο Testbench για το 1^ο κύκλωμα εφαρμόσαμε 5 παραδείγματα:

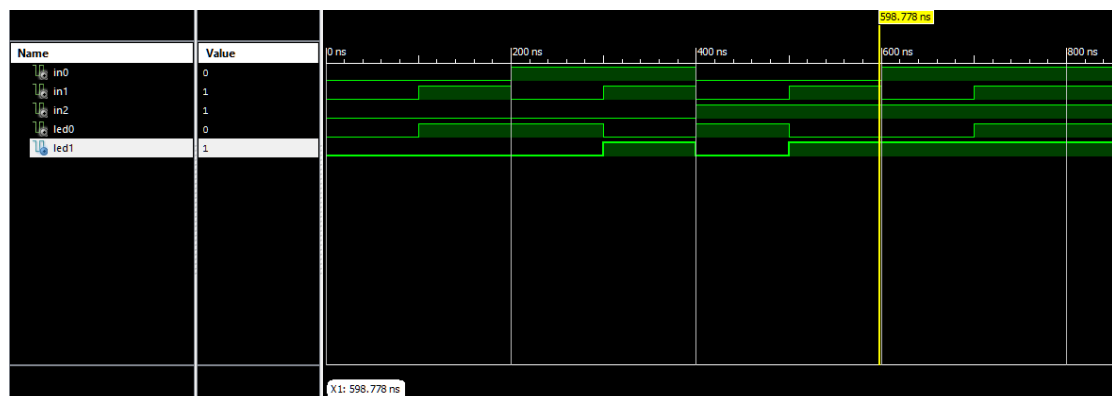
- IN0=0, IN1=0, BTN0=1, BTN1=1, BTN2=1, BTN3=1.
- IN0=0, IN1=1, BTN0=1, BTN1=1, BTN2=1, BTN3=1.
- IN0=1, IN1=0, BTN0=1, BTN1=1, BTN2=1, BTN3=1.
- IN0=1, IN1=1, BTN0=1, BTN1=1, BTN2=1, BTN3=1.
- IN0=1, IN1=0, BTN0=0, BTN1=0, BTN2=0, BTN3=0.



2^ο Κύκλωμα

Στο Testbench για το 2^ο κύκλωμα εφαρμόσαμε 8 παραδείγματα:

- IN0=0, IN1=0, IN2=0.
- IN0=0, IN1=1, IN2=0.
- IN0=1, IN1=0, IN2=0.
- IN0=1, IN1=1, IN2=0.
- IN0=0, IN1=0, IN2=1.
- IN0=0, IN1=1, IN2=1.
- IN0=1, IN1=0, IN2=1.
- IN0=1, IN1=1, IN2=1.



Συμπεράσματα

Υλοποιώντας αυτά τα απλά συνδυαστικά κυκλώματα καταφέραμε να εξοικειωθούμε με την γλώσσα VHDL, με την σχεδιαστική ροή απλών ψηφιακών συστημάτων, καθώς επίσης και με την υλοποίηση κυκλωμάτων μέσα από την ένωση components (πχ Full Adder). Μάθαμε την χρήση του εργαλείου Xilinx καθώς και τις λειτουργίες: προσομοίωση (Simulation), μετάφραση (Translation), αποτύπωση (Map), τοποθέτηση και διασύνδεση (Place and Route).

Κώδικας

1ο Κύκλωμα

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity lab1_a is
33     Port ( IN0 : in  STD_LOGIC;
34           IN1 : in  STD_LOGIC;
35           BTN0 : in  STD_LOGIC;
36           BTN1 : in  STD_LOGIC;
37           BTN2 : in  STD_LOGIC;
38           BTN3 : in  STD_LOGIC;
39           LED0 : out STD_LOGIC;
40           LED1 : out STD_LOGIC;
41           LED2 : out STD_LOGIC;
42           LED3 : out STD_LOGIC;
43           LED4 : out STD_LOGIC;
44           LED5 : out STD_LOGIC;
45           LED6 : out STD_LOGIC;
46           LED7 : out STD_LOGIC);
47 end lab1_a;
48
49 architecture Behavioral of lab1_a is
50
51 begin
52
53 LED0 <= (IN0 AND IN1) AND BTN0;
54 LED1 <= (IN0 XOR IN1) AND BTN1;
55 LED2 <= (IN0 NOR IN1) AND BTN2;
56 LED3 <= (NOT IN0) AND BTN3;
57 LED4 <= IN0;
58 LED5 <= IN1;
59
60 end Behavioral;
61
```

2ο Κύκλωμα

Half Adder

```
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity half_adder is
33     Port ( IN0 : in  STD_LOGIC; --A
34           IN1 : in  STD_LOGIC; --B
35           LED0 : out STD_LOGIC; --SUM
36           LED1 : out STD_LOGIC); --CARRY
37 end half_adder;
38
39 architecture Behavioral of half_adder is
40
41 begin
42
43 LED0 <= IN0 XOR IN1;
44 LED1 <= IN0 AND IN1;
45
46 end Behavioral;
47
```


Full Adder

```
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity full_adder is
33     Port ( IN0 : in  STD_LOGIC;
34           IN1 : in  STD_LOGIC;
35           IN2 : in  STD_LOGIC;
36           LED0 : out STD_LOGIC;
37           LED1 : out STD_LOGIC);
38 end full_adder;
39
40 architecture Behavioral of full_adder is
41
42     component half_adder is
43         Port ( IN0 : in  STD_LOGIC; -- IN0=A
44               IN1 : in  STD_LOGIC; -- IN1=B
45               LED0 : out STD_LOGIC; --LED0=Sum
46               LED1 : out STD_LOGIC); --LED1=Cout
47     end component ;
48
49     signal hsum, hcarry, tcarry: std_logic;
50
51     begin
52
53     HA1 : half_adder
54         port map ( IN0=>IN0,
55                   IN1=>IN1,
56                   LED0=>hsum,
57                   LED1=>hcarry);
58     HA2 : half_adder
59         port map ( IN0=>hsum,
60                   IN1=>IN2, --Cin
61                   LED0=>LED0,
62                   LED1=>tcarry);
63
64         LED1 <= tcarry or hcarry;
65
66     end Behavioral;
```