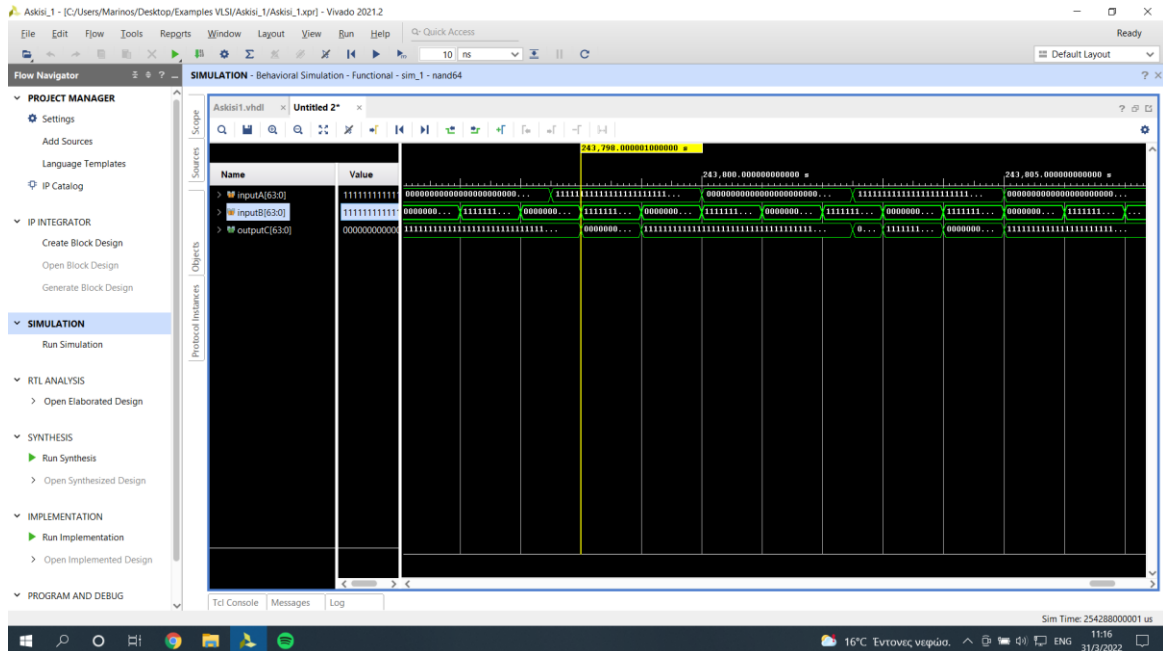


Σχεδιασμός Συστημάτων VLSI 2022

Εργασία 1

1)



2) Από την εξομοίωση παρατηρούμε ότι οι συνδυασμοί εισόδων με τα αποτελέσματα των εξόδων, ταυτίζονται με την λογική πύλη NAND. Συνεπώς, ο κώδικας που εξομοιώσαμε είναι πύλη NAND 2 εισόδων από 64 bits η κάθε μία, με μία έξοδο 64 bits.

3) Εναλλακτικός κώδικας:

```
D:\Σχολή Μαρίνος\Σχεδιασμός Συστημάτων VLSI\ΕΑΣΚΗΣΗ-1\ΓΚΙΖΑΣ-MAPINOS-AM-10543
Αρχείο Επεξεργασία Εύρεση Προβολή Κωδικοποίηση Γλώσσα Ρυθμίσεις Tools Μα
DB connection.php counter4.v syn_clr_testcounter8.v nand1.vhdl
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.all;
3
4  ENTITY nand1 IS
5  PORT (inputA :IN std_logic_vector(63 downto 0);
6        inputB :IN std_logic_vector(63 downto 0);
7        outputC :OUT std_logic_vector(63 downto 0));
8  END nand1;
9
10 ARCHITECTURE structural OF nand1 IS
11 BEGIN
12     outputC <= inputA NAND inputB;
13 END structural;
```

Τρόπος σκέψης

Με αυτόν τον εναλλακτικό τρόπο υλοποίησης του κώδικα, δηλαδή αφαιρώντας την for loop, εκμεταλλευόμαστε την παραλληλία που προσφέρουν οι γλώσσες περιγραφής υλικού (στην περίπτωση μας η VHDL), δίνοντάς μας τα παρακάτω πλεονεκτήματα.

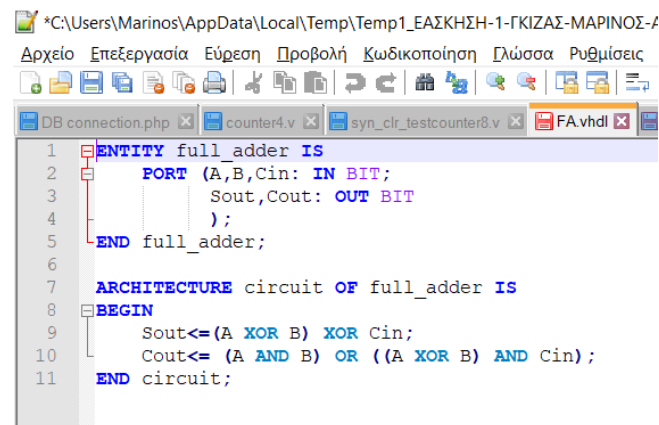
(+) Πλεονεκτήματα:

- Μικρότερο χρόνο εκτέλεσης απότι με την for loop που θα εκτελούσε σειριακά την εντολή.
- Λιγότερες γραμμές κώδικα, άρα και πιο απλός.

(-) Μειονεκτήματα:

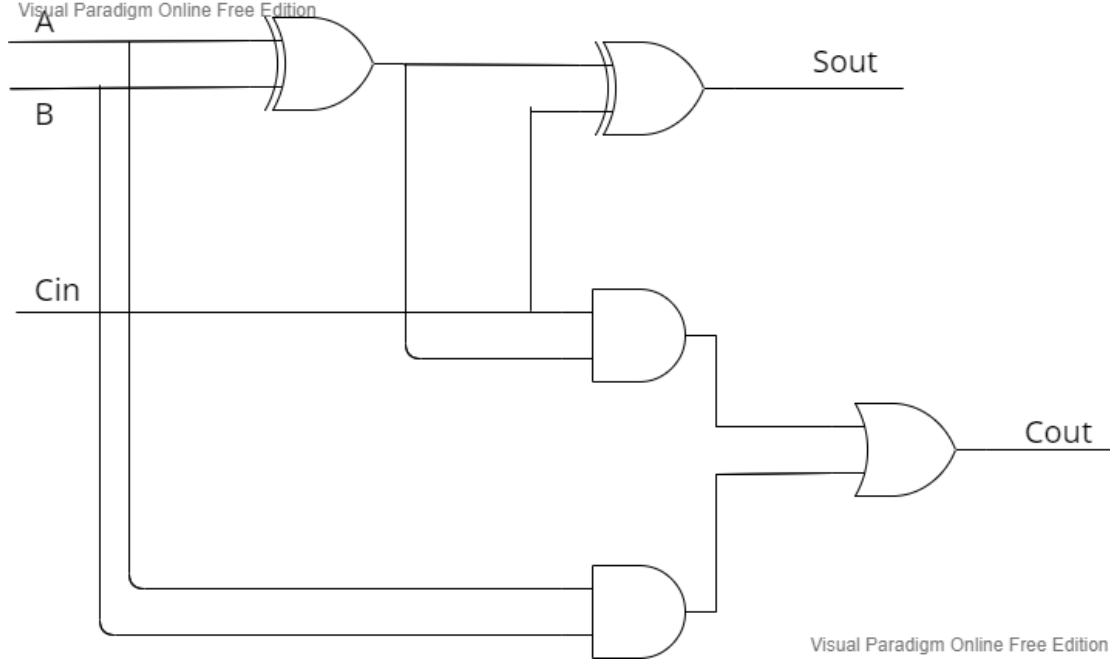
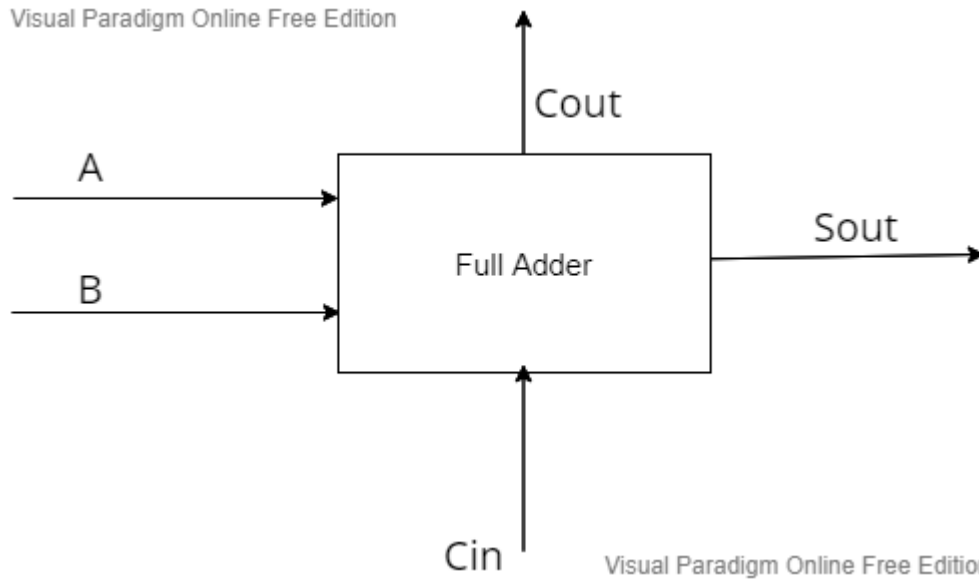
Γ)

1) Full-Adder

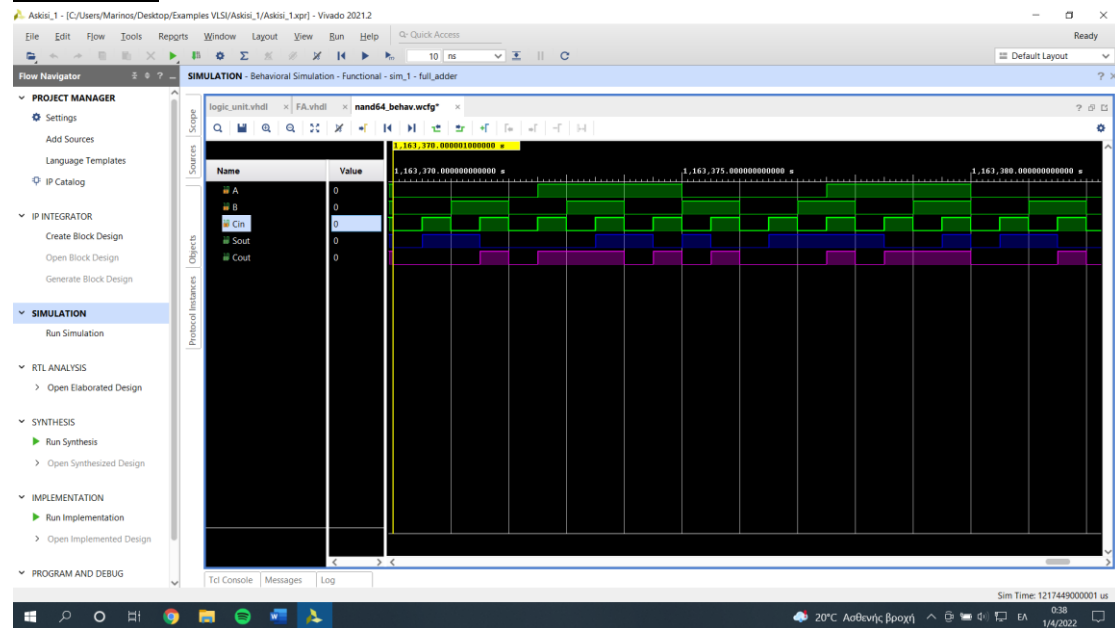


```
1 ENTITY full_adder IS
2   PORT (A,B,Cin: IN BIT;
3         Sout,Cout: OUT BIT
4        );
5 END full_adder;
6
7 ARCHITECTURE circuit OF full_adder IS
8 BEGIN
9   Sout<= (A XOR B) XOR Cin;
10  Cout<= (A AND B) OR ((A XOR B) AND Cin);
11 END circuit;
```

Architecture: Για την σχεδίαση του πλήρη αθροιστή θα χρησιμοποιήσουμε 2 πύλες XOR, 1 OR και 2 AND.



Waveforms:

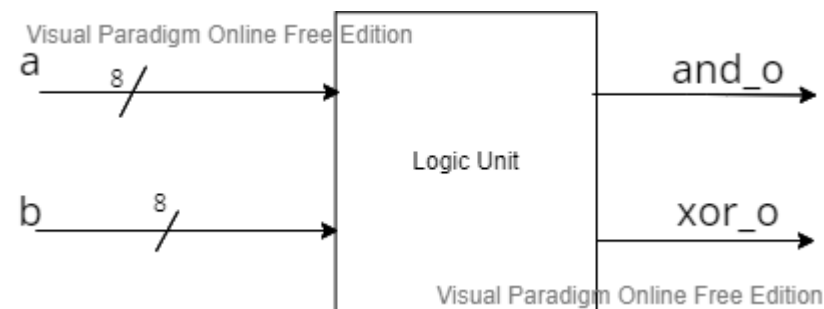


Παρατηρούμε ότι από τους συνδυασμούς των τιμών εισόδου-εξόδου, επαληθεύεται ο πίνακας αληθείας του full-adder.

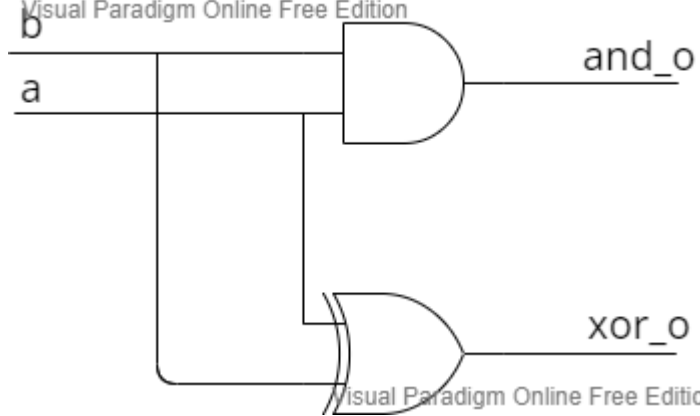
2) Logic-Unit

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.all;
3
4  ENTITY logic_unit IS
5      PORT (a :IN std_logic_vector(7 downto 0);
6            b :IN std_logic_vector(7 downto 0);
7            xor_o :OUT std_logic_vector(7 downto 0);
8            and_o :OUT std_logic_vector(7 downto 0));
9  END logic_unit;
10
11 ARCHITECTURE circuit OF logic_unit IS
12     BEGIN
13         xor_o <= a XOR b;
14         and_o <= a AND b;
15     END circuit;
16
```

Architecture:

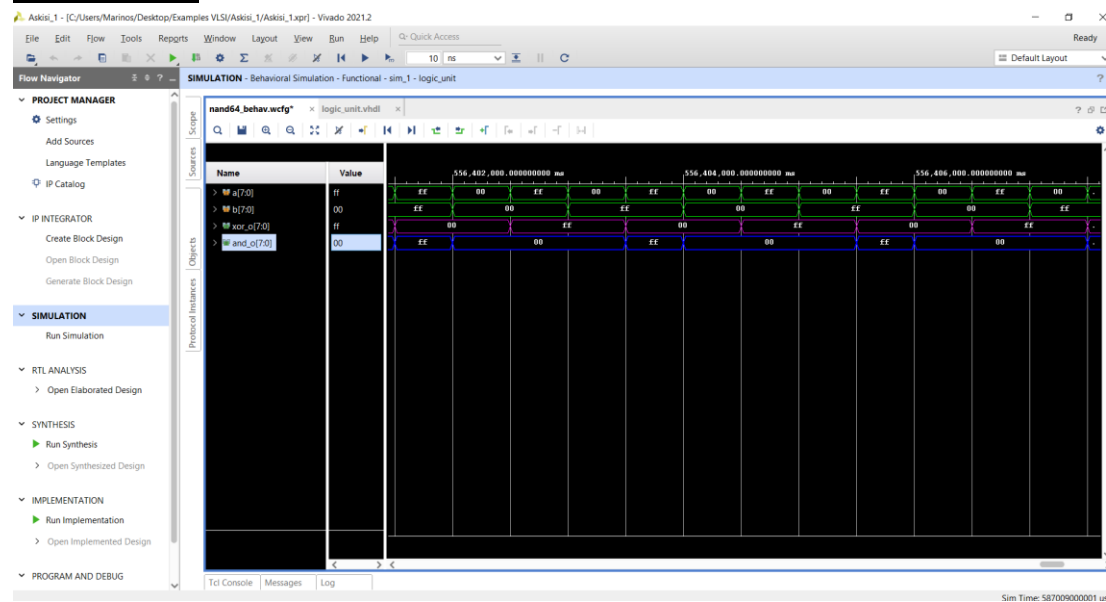


Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Waveforms:



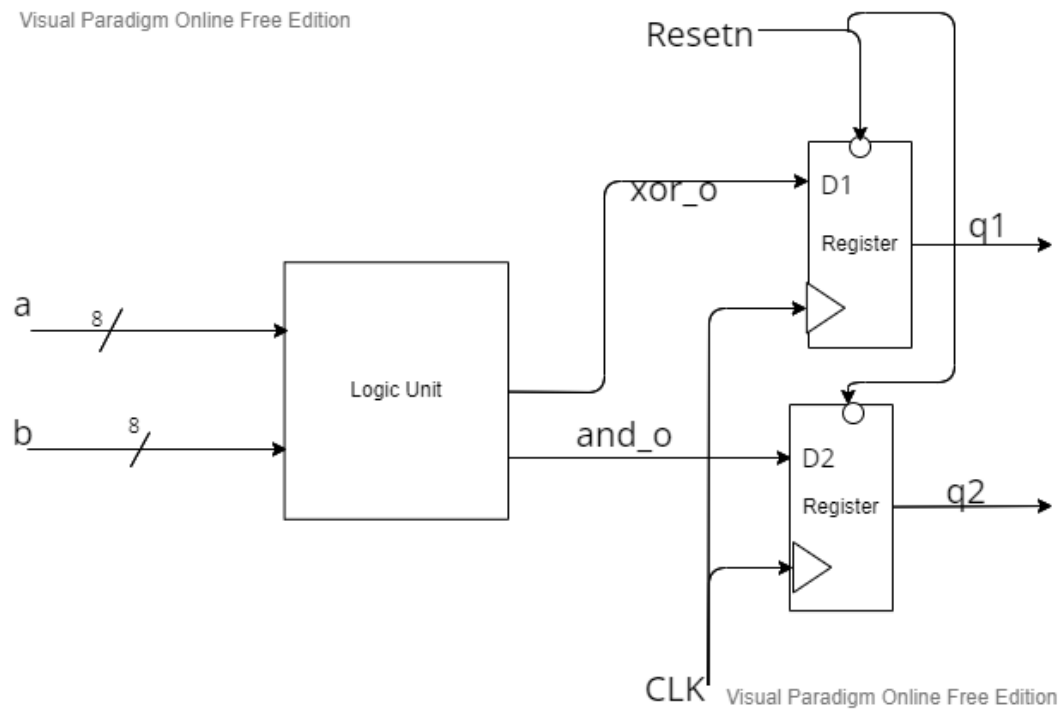
Γ3) Logic-Unit with Registers

```
*C:\Users\Marinos\AppData\Local\Temp\Temp1_EA5KXZH-1-ΓΚΙΖΑΣ-MAPINOΣ-AM-1054348-ΣΧΕΔΙΑΣΜΟΣ-ΣΥΣΤΗΜΑΤΟΣ\
Αρχείο Επεξεργασία Εύρεση Προβολή Κωδικοποίηση Γλώσσα Ρυθμίσεις Tools Μακροεντολή Εκτέλεση
DB connection.php counter4.v syn_clk_testcounter8.v FA.vhdl logic_unit.vhdl lu.vhdl
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.all;
3
4  ENTITY lu IS
5      PORT (a :IN std_logic_vector(7 downto 0);
6            b :IN std_logic_vector(7 downto 0);
7            Resetn, Clock : IN std_logic ;
8            q1 :OUT std_logic_vector(7 downto 0); -- XOR
9            q2 :OUT std_logic_vector(7 downto 0)); --AND
10  END lu;
11
12  ARCHITECTURE circuit OF lu IS
13      SIGNAL xor_o, and_o : std_logic_vector(7 downto 0);
14      BEGIN
15          xor_o <= a XOR b;
16          and_o <= a AND b;
17          PROCESS ( Resetn, Clock)
18              BEGIN
19                  IF (Resetn = '0') THEN
20                      q1 <= "00000000" ;
21                  ELSIF rising_edge(Clock) THEN
22                      q1 <= xor_o;
23                  END IF ;
24              END PROCESS ;
25          PROCESS ( Resetn, Clock)
26              BEGIN
27                  IF (Resetn = '0') THEN
28                      q2 <= "00000000" ;
29                  ELSIF rising_edge(Clock) THEN
30                      q2 <= and_o;
31                  END IF ;
32              END PROCESS ;
33      END circuit;
```

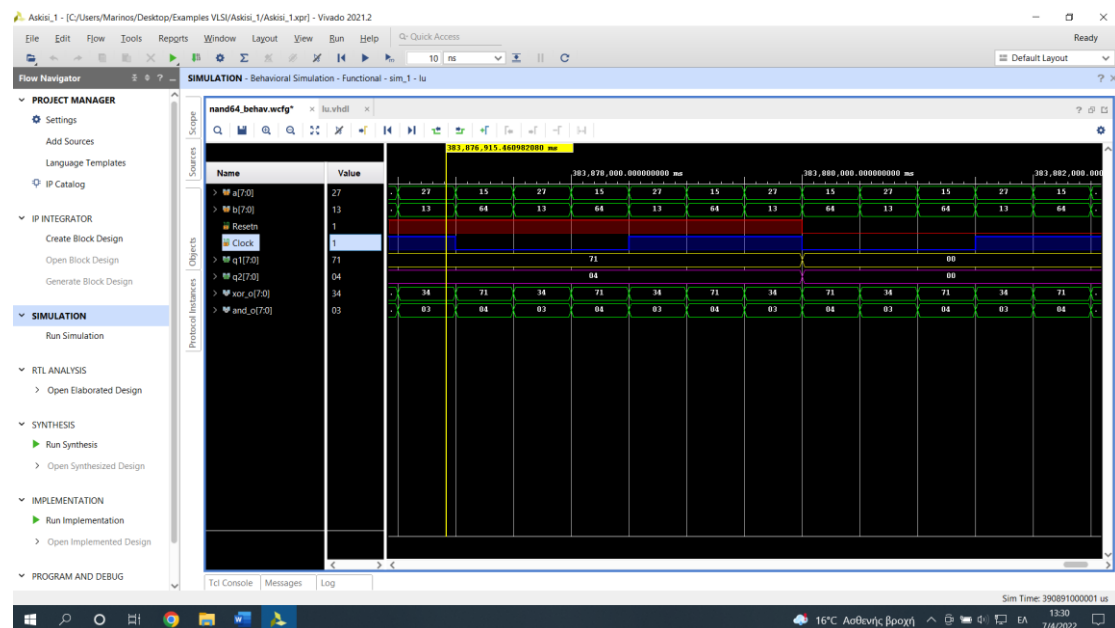
Οι registers έχουν υλοποιηθεί με ένα D-FF των 8 bits.

Architecture:

Visual Paradigm Online Free Edition



Waveforms:



Όταν έχουμε $\text{Resetn} = 0$ τότε παίρνουμε ως έξοδο από τους δύο registers την τιμή 0, ενώ τα σήματα xor_o , and_o έχουν αποτελέσματα.