5/3/2017

1ή Εργαστηριακή Άσκηση

Εξοικείωση με την γλώσσα περιγραφής υλικού και την ιεραρχική σχεδίαση

(Structural Vhdl)

03/03/2017

Ομάδα LAB20332005

ΧΡΗΣΤΟΣ ΖΗΣΚΑΣ 2014030191
ΠΑΝΑΓΙΩΤΗΣ ΣΑΒΒΑΙΔΗΣ 2013030180

Σκοπός εργαστηριακής άσκησης

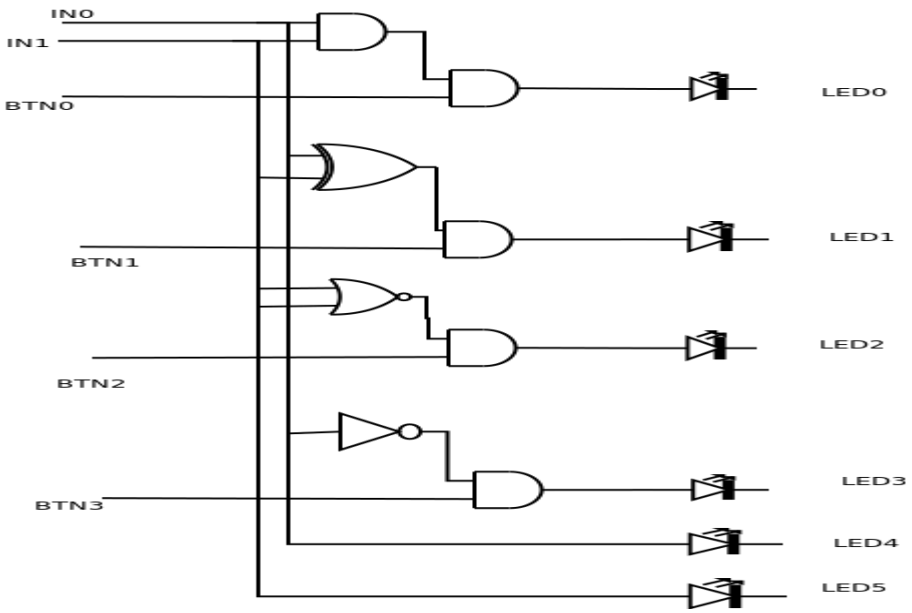
Είναι η εξοικείωση με τη γλώσσα περιγραφής υλικού VHDL(VHSIC Hardware Description Language) για απλά συνδυαστικά κυκλώματα, καθώς επίσης και με την πλήρη σχεδιαστική ροή απλών ψηφιακών συστημάτων. Επιπλέον, γίνεται μια πρώτη προσέγγιση πάνω στο σχεδιαστικό πρόγραμμα Xilinx ISE δημιουργώντας την λειτουργικότητα του για υλοποίηση σχεδίασης συνδυαστικών κυκλωμάτων ψηφιακά και την σύνδεση με την αναδιατασσόμενη συσκευή (FPGA) για επαλήθευση της ορθής λειτουργίας του κυκλώματος.

Προεργασία

Παρουσιάζουμε τις κυμματομορφές για το κύκλωμα 1 που αφορά τις συσχετίσεις μεταξύ απλών πυλών αλλά και του κυκλώματος 2 που αφορά την υλοποίηση του δυαδικού αθροιστή (FA-Full Adder- Παρουσιάζουμε και τον ημιαθροιστή) αλλά και τις συναρτήσεις που αφορούν τις εξόδους των πυλών.

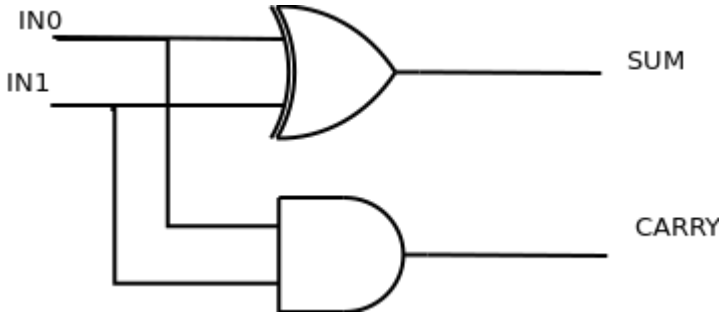
Κύκλωμα 1

Συσχετίσεις Πυλών με κουμπία και LEDs



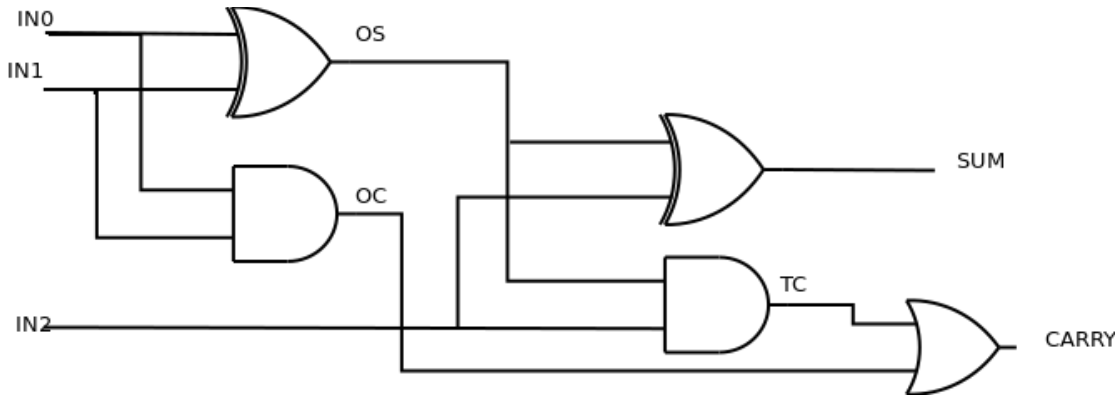
- ***LED0 = IN0*IN1*BTN0***
- ***LED1= (((IN0)'*IN1)+(IN0*(IN1)'))*BTN1***
- ***LED2= (IN0+IN1)'BTN2***
- ***LED3=IN0' * BTN3***
- ***LED4=IN0***
- ***LED5=IN1***

Κύκλωμα 2
HA-Half Adder



- ***SUM= (((IN0)'*IN1)+(IN0*(IN1)'))***
- ***CARRY = IN0*IN1***

FA- Full Adder



- ***OC= IN0 * IN1***
- ***OS=(((IN0)'*IN1)+(IN0*(IN1)'))***
- ***SUM=((OS)'*IN2) + (OS*(IN2)'))***
- ***TC=OS*IN2***
- ***CARRY=TC + OC***

Περιγραφή

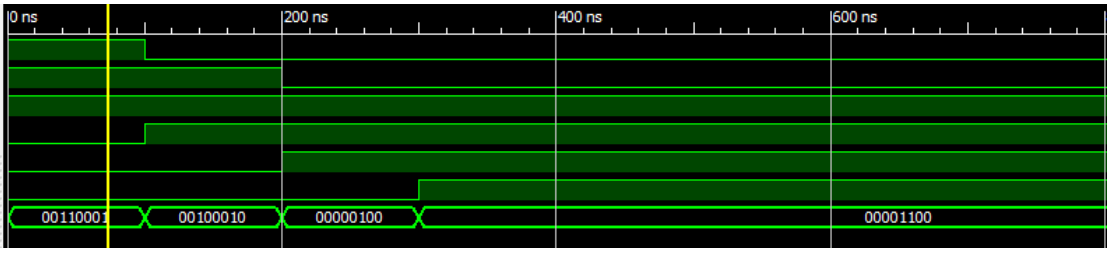
Για το κύκλωμα 1 , ζητείται η σχεδίαση πυλών AND,XOR,NOR,NOT για εισόδους IN1,IN0 και εξόδους που οδηγούν σε LED ενώ οι έξοδοι δίνουν αποτέλεσμα για ενεργά κουμπιά (BTN0-BTN3) . Για αυτό συνδέουμε τις εξόδους των διαφόρων πυλών ως εισόδους στις διάφορες AND. Η 2η είσοδος σε κάθε AND είναι κάποιο κουμπί . Όταν είναι ενεργά τα κουμπιά οι πύλες AND αφήνουν να περάσει στην έξοδος το αποτέλεσμα των αντίστοιχων πυλών στα διάφορα LED . Στα LED4-LED5 έχουμε τις εισόδους IN0,IN1. Για την υλοποίηση του κυκλώματος 2 , χρειάζεται να κατασκευαστεί ένας ημιαθροιστής. Χρησιμοποιούνται μια πύλη AND για αναπαρασταθεί το κρατούμενο CARRY, ενώ το άθροισμα αναπαρίσταται από μια πύλη XOR. Ο πλήρης αθροιστής 2-bit κατασκευάζεται από 2 ημιαθροιστές και μια πύλη OR. Στο SUM του πλήρη αθροιστή έχουμε ως εισόδους την έξοδο της πύλης XOR και το άλλο bit, ενώ στο CARRY πηγαίνει το κρατούμενο που έχει προέρθει από το άθροισμα του ημιαθροιστή και εφόσον υπάρχει εμφανίζεται ως έξοδος της OR.

Κυματομορφές-Προσομοίωση

Παρουσιάζουμε τις κυμματομορφές των 2 κυκλωμάτων

Κύκλωμα 1

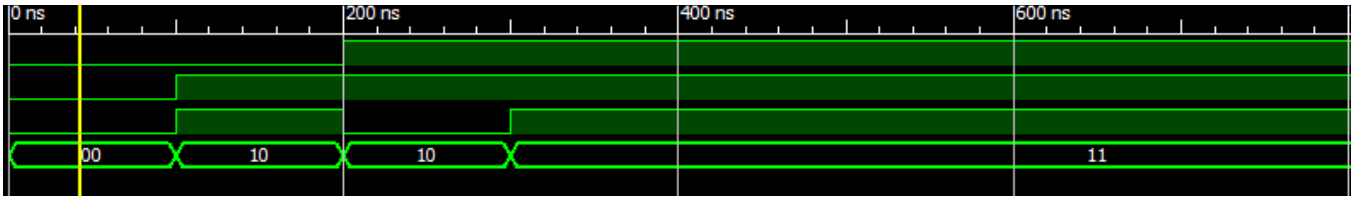
Συσχετίσεις Πυλών με κουμπία και LEDs



Τα LED6,LED7 τα έχουμε αρχικοποιήσει με 0. Το LSB αφορά την πύλη AND και συνεχίζει με την XOR,NOR,NOT,IN0,IN1. Παρατηρούμε ότι δίνοντας εισόδους IN0=1, IN1=1 και ενεργό κουμπί BTN0, ότι η πύλη AND μας δίνει έξοδο καθώς και τα LED που έπαιρναν ως είσοδο κατευθείαν τα σήματα εισόδου , αποτέλεσμα που αναμέναμε αφού αυτά δεν συνδέονται

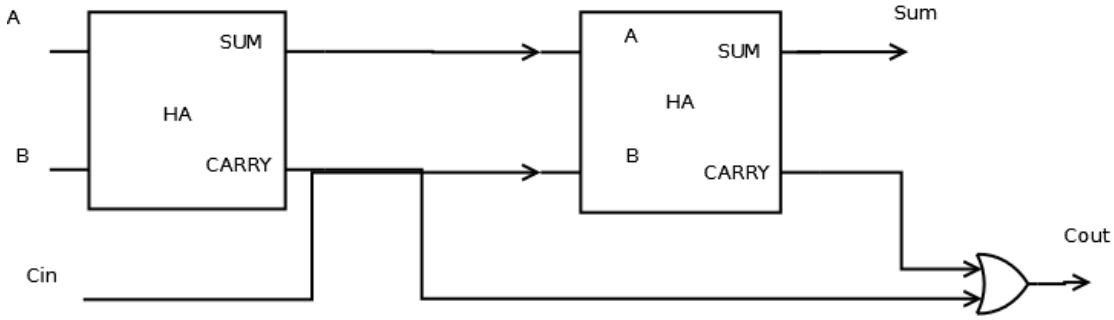
με κουμπιά(τα υπόλοιπα κουμπιά είναι ανενεργά οπότε οι πύλες AND δεν βγάζουν έξοδο). Εξετάζοντας περαιτέρω , για εισόδους IN1=1 , IN0=0 , βλέπουμε ότι μόνο η XOR εμφανίζει έξοδο ΄όταν είναι ενεργό το κουμπί BTN1(το κουμπί BTN0 παραμένει ενεργό). Αντίστοιχα αποτελέσματα έχουμε και για τις υπόλοιπες περιπτώσεις εισόδων (θεωρούμε για κάθε περίπτωση ενεργά τα αντίστοιχα κουμπιά καθώς και αυτά που έχουν ήδη ενεργοποιηθεί).

Κύκλωμα 2
FA- Full Adder



Σύμφωνα με τα αποτελέσματα των κυμματομορφών αλλά και τα αποτελέσματα της σχεδίαση διαπιστώνουμε ότι το διάγραμμα απεικονίζει πλήρως την λειτουργικότητα ενός πλήρη αθροιστή. Θεωρώντας όλα τα bit εισόδου 0, ο αθροιστής δεν πραγματοποιεί καμία ενέργεια άθροισης. Για εισόδους IN1=1,IN2=1,IN0=0, η πρόσθεση του IN1 με το IN0 μας δίνει στο SUM 1 χωρίς CARRY , ενώ προσθέτοντας και το IN2 (η πύλη XOR μας δίνει έξοδο 0, ενώ η AND δημιουργεί CARRY από το IN2 και από την XOR του IN0,IN1) εμφανίζεται κρατούμενο που απεικονίζεται στο δεύτερο bit (στο σχήμα το MSB είναι αριστερά και το LSB δεξιά). Στη συνέχεια εναλλάσσοντας τα IN0,IN2 εμφανίζονται ισοδύναμα συμπεράσματα . Τελικά για όλες τις εισόδους ενεργές και το SUM και το CARRY δίνουν έξοδο 1(Η XOR IN1,IN0 δίνει 0 και σε συνδυασμό με την XOR του IN2 προκύπτει ως αποτέλεσμα του SUM το 1 , ενώ το CARRY της πρώτης άθροισης παρέχει κρατούμενο από την AND του IN0,IN1 το οποίο προορίζεται και στην τελική AND που θα βγάλει το τελικό CARRY.

Παρακάτω παρουσιάζουμε και τη σχεδίαση του πλήρη αθροιστή σε block diagram



Συμπεράσματα

Με την περάτωση αυτής της εργαστηριακής άσκησης γίνονται αντιληπτά ορισμένα αξιοσημείωτα συμπεράσματα όπως, η εντριβή και η εξάσκηση με τις εντολές για προγραμματισμό υλικού σε ιεραρχική σχεδίαση, η σύνδεση τους με αναδιατασσόμενες συσκευές καθώς και η εισαγωγή σε σχεδίαση πολύπλοκων κυκλωμάτων

Κώδικας

CIRCUIT 1

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;


entity circuit1 is

   Port ( IN0 : in  STD_LOGIC;

          IN1 : in  STD_LOGIC;

          BTN0 : in  STD_LOGIC;

          BTN1 : in  STD_LOGIC;

          BTN2 : in  STD_LOGIC;

          BTN3 : in  STD_LOGIC;

          LED : out  STD_LOGIC_VECTOR (7 downto 0));

end circuit1;


architecture Behavioral of circuit1 is


begin

LED(0)<=IN0 AND IN1 AND BTN0;

LED(1)<=(IN0 XOR IN1) AND BTN1;

LED(2)<=(IN0 NOR IN1) AND BTN2;

LED(3)<=NOT IN0 AND BTN3;

LED(4)<=IN0;
```

```
LED(5)<=IN1;
LED(6)<='0';
LED(7)<='0';
end Behavioral;
```

FULL ADDER

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Full_Adder is
   Port ( IN0 : in  STD_LOGIC;
          IN1 : in  STD_LOGIC;
          IN2 : in  STD_LOGIC;
          LED : out  STD_LOGIC_VECTOR (1 downto 0));
end Full_Adder;
architecture Behavioral of Full_Adder is
SIGNAL OS,OC,TC: STD_LOGIC;

component Half_Adder
port (IN0,IN1: in  STD_LOGIC;
                     SUM,CARRY : out  STD_LOGIC);
end component;

begin
Half_Adder1 : Half_Adder
port map ( IN0=>IN0,IN1=>IN1,SUM=>OS,CARRY=>OC);
```

```
Half_Adder2 : Half_Adder
port map ( IN0=>OS ,IN1=>IN2,SUM=>LED(0),CARRY=>TC);


LED(1)<= TC OR OC;


end Behavioral;
```

HALF ADDER

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity Half_Adder is
   Port ( IN0 : in  STD_LOGIC;
          IN1 : in  STD_LOGIC;
          SUM : out  STD_LOGIC;
CARRY : out STD_LOGIC);
end Half_Adder;


architecture Behavioral of Half_Adder is
begin
SUM<=IN0 XOR IN1;
CARRY<=IN0 AND IN1;


end Behavioral;
```