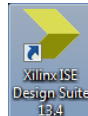


Σχεδιαστική Ροή με το εργαλείο ISE της Xilinx

Οι παρακάτω οδηγίες αναφέρονται στην έκδοση 13.4 Xilinx ISE Webpack. Στο εργαστήριο μπορεί να χρησιμοποιηθεί διαφορετική έκδοση του λογισμικού οι διαδικασίες είναι εντελώς ίδιες και το περιβάλλον μοιάζει σε αρκετά μεγάλο βαθμό.

Την συγκεκριμένη έκδοση των εργαλείων της Xilinx ISE Web Pack μπορείτε να την κατεβάσετε από το site της εταιρίας www.xilinx.com. Να σημειωθεί ότι όταν πάτε να κατεβάσετε την έκδοση **13.4 Xilinx ISE WebPack**, θα πρέπει πρώτα να δημιουργήσετε account σύμφωνα με τις οδηγίες που δίνονται από το site.

1. Για να ξεκινήσετε το ISE κάνετε διπλό κλικ στο εικονίδιο (Εικόνα 1) στην επιφάνεια εργασίας:



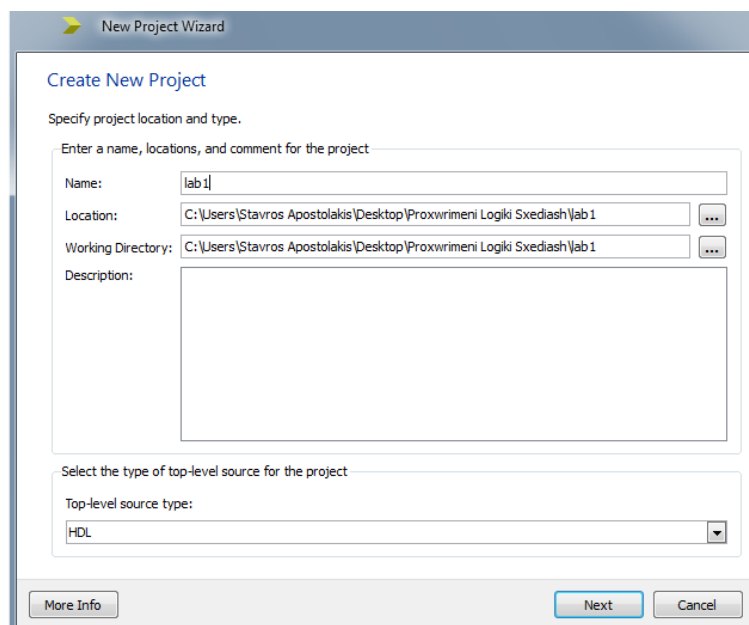
Εικόνα 1: Εικονίδιο Xilinx ISE 13.4

Ή εναλλακτικά ξεκινήστε το ISE επιλέγοντας **Start → All Programs → Xilinx ISE Design Suite 13.4 → ISE Design Tools → Project Navigator → Click (^.^)**

2. New Project

Δημιουργείται ένα **Νέο Project**. Αναλυτικά σε βήματα:

α) Επιλέγετε **File→New Project** και εμφανίζεται ο New Project Wizard όπως φαίνεται στην εικόνα 2:



Εικόνα 2.

b) Στο πεδίο **Name** δώστε το όνομα lab1 και ως **Location** καθορίστε το path του project: **C:\Lab_User_Temp\HRY203\lab1** όπως φαίνεται στη Εικόνα 2.

c) Βεβαιωθείτε ότι ως Top-Level Source Type είναι επιλεγμένη η **HDL**.

d) Επιλέξτε **Next** για να συνεχίσετε.

3. New Project – Project Settings

Συμπληρώστε τον Πίνακα που ακολουθεί ως εξής:

- Family: **Spartan3E**
- Device: **XC3S100E**
- Package: **CP132**
- Speed: **-5**
- Top-Level Module Type: **HDL**
- Synthesis Tool: **XST (VHDL/Verilog)**
- Simulator: **ISim**
- Preferred Language: **VHDL**

Όταν τελειώσετε ο πίνακας πρέπει να είναι όπως στην Εικόνα 3:

New Project Wizard

Project Settings

Specify device and project properties.
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3E
Device	XC3S100E
Package	CP132
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info Next Cancel

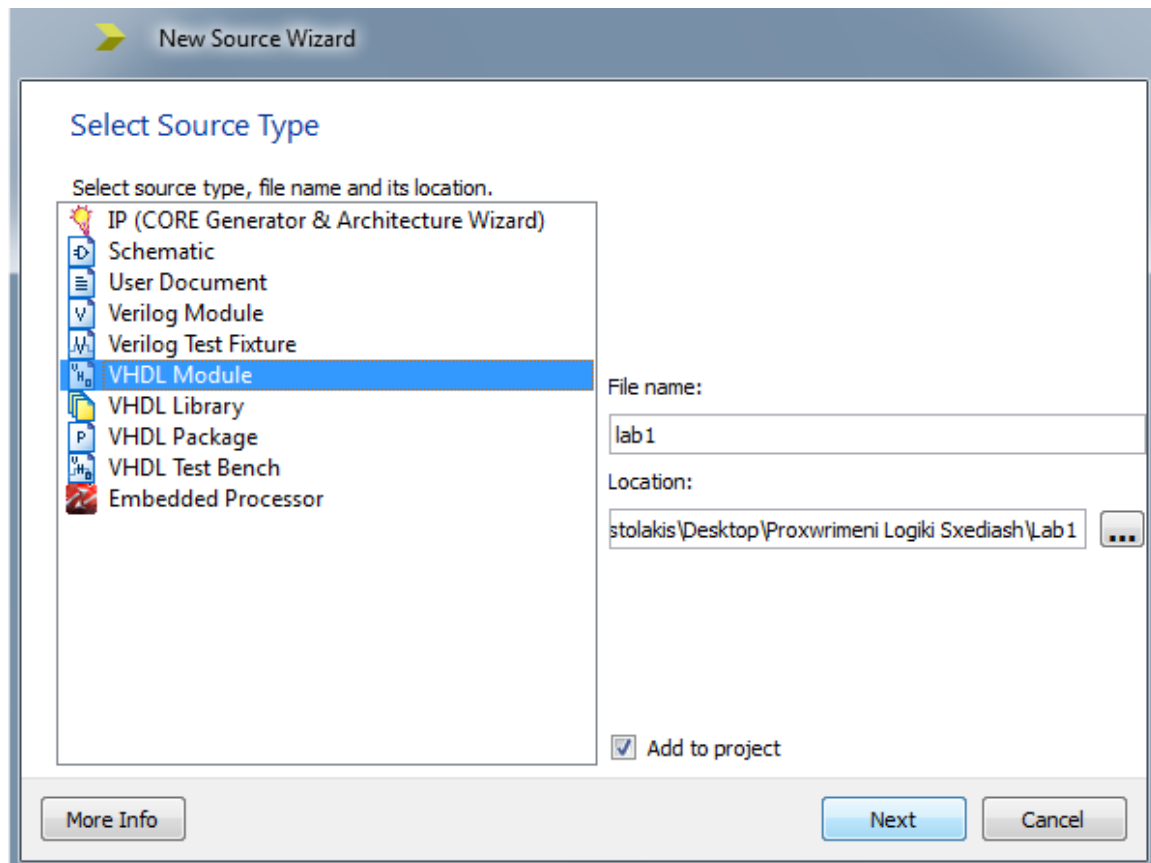
Εικόνα 3

Επιλέξετε **Next** για να συνεχίσετε και Finish στο Project Summary.

4. Εισαγωγή νέου Αρχείου (New Source) .

Έχουμε φτάσει σε σημείο όπου έχει δηλωθεί η συσκευή με την οποία θα δουλέψουμε στο project μας. Τώρα πρέπει να εισάγουμε στο project μας τα δικά μας αρχεία σχεδίασης.

Επιλέγουμε **Project** → **New Source...** οπότε εμφανίζεται η εικόνα:



Εικόνα 4: New Source Wizard

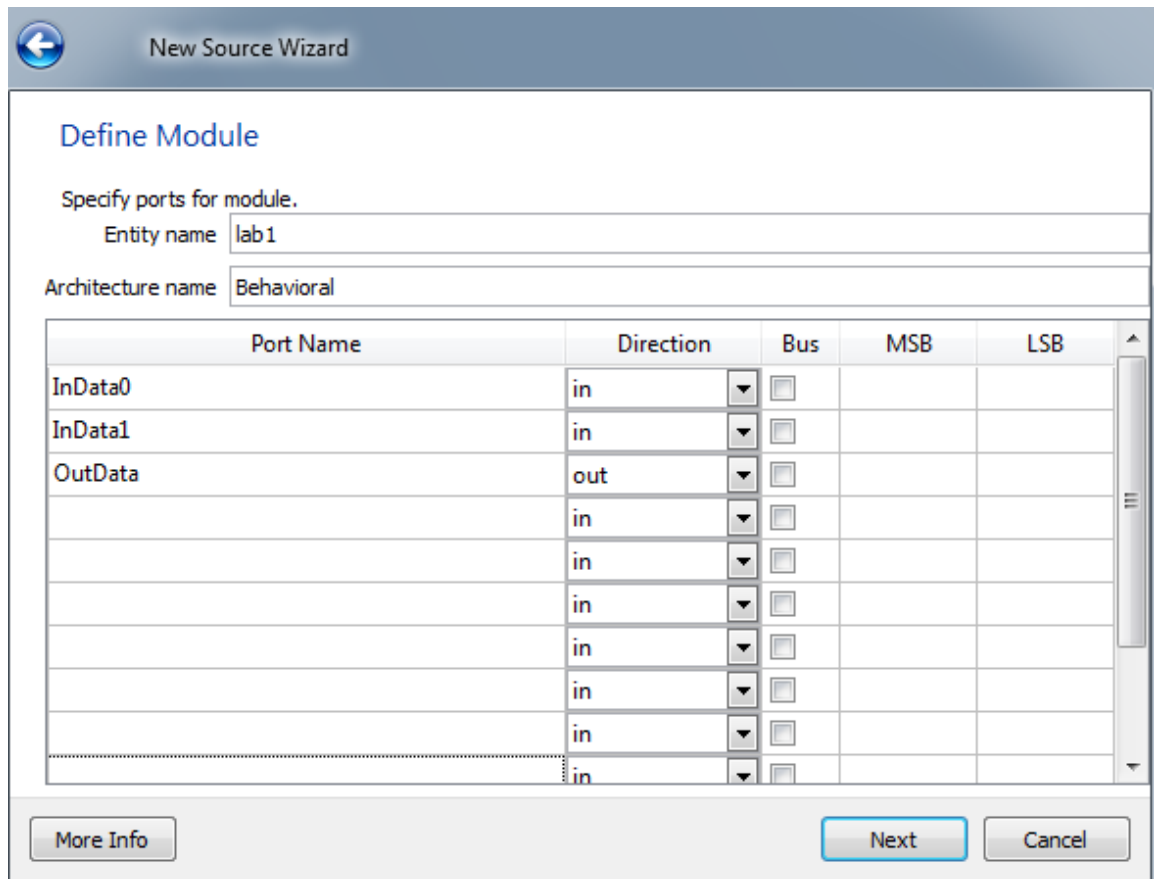
Επιλέγουμε **VHDL Module** και στο πεδίο File Name βάζουμε ένα όνομα πχ lab1

5. Ορισμός εισόδων

Στο συγκεκριμένο παράδειγμα θα φτιάξουμε μία πύλη AND δύο εισόδων (InData1, InData0) θα έχει ως έξοδο την OutData.

Γεμίζουμε τα αντίστοιχα πεδία με τα ονόματα των εισόδων/εξόδων και επιλέγουμε αν είναι είσοδος ή έξοδος. Σε περίπτωση που είχαμε σήμα πλάτους μεγαλύτερου από 1 θα σημειώναμε την επιλογή BUS.

Επιλέγετε **Next** για να συνεχίσετε...



The image shows a software window titled "New Source Wizard" with a "Define Module" section. It includes input fields for "Entity name" (lab1) and "Architecture name" (Behavioral). Below these is a table for defining module ports. The table has columns for Port Name, Direction, Bus, MSB, and LSB. Pre-defined ports include InData0 (input), InData1 (input), and OutData (output). There are also several empty rows for additional ports. At the bottom, there are buttons for "More Info", "Next", and "Cancel".

Port Name	Direction	Bus	MSB	LSB
InData0	in	<input type="checkbox"/>		
InData1	in	<input type="checkbox"/>		
OutData	out	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

Εικόνα 5: New Source Wizard - Define Module

Μετά που θα επιλέξετε Summary οπότε σας παρουσιάζονται όλα τα σήματα εισόδου και εξόδου και επιλέγεται **Finish**.

6. Εισαγωγή Κώδικα

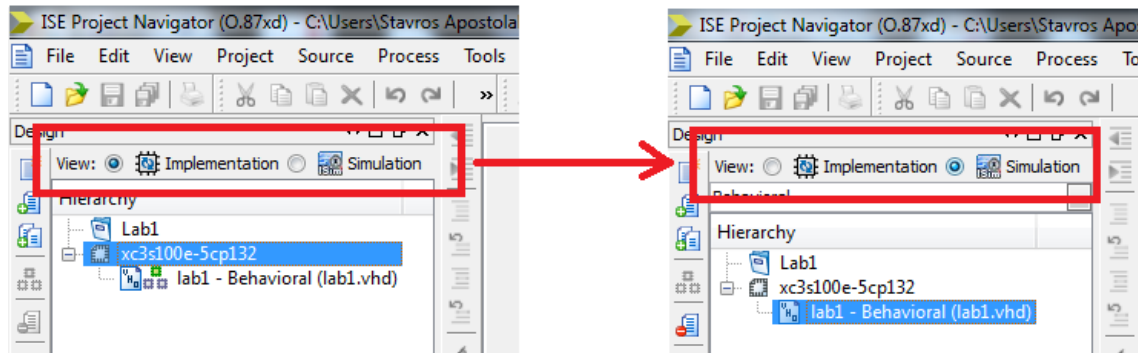
Εμφανίζεται το template του κώδικα VHDL όπως φαίνεται στην εικόνα 10 όπου έχει προστεθεί η γραμμή 42 που υλοποιεί την πύλη AND.

```
1 -----
2 -- Company:
3 -- Engineer:
4 --
5 -- Create Date:    00:00:00 00/00/0000 :P
6 -- Design Name:
7 -- Module Name:    lab1 - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
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 is
33     Port ( InData0 : in  STD_LOGIC;
34           InData1 : in  STD_LOGIC;
35           OutData  : out STD_LOGIC);
36 end lab1;
37
38 architecture Behavioral of lab1 is
39
40 begin
41
42 OutData <= InData0 and InData1;
43
44 end Behavioral;
```

Εικόνα 7: VHDL template

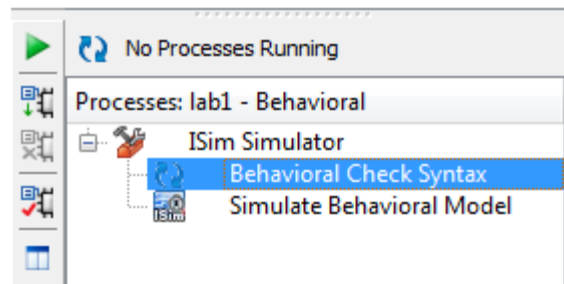
7. Behavioral Check Syntax.

Αφού κάνουμε τις εισαγωγές και τις μορφοποιήσεις στο αρχικό template θα πρέπει να ελέγξουμε εάν η σύνταξη της γλώσσας VHDL που έχουμε γράψει είναι σωστή. Για την διαδικασία αυτή αλλάζουμε το view της σχεδίασης μας από το Implementation στο Simulation όπως φαίνεται στη εικόνα 8.



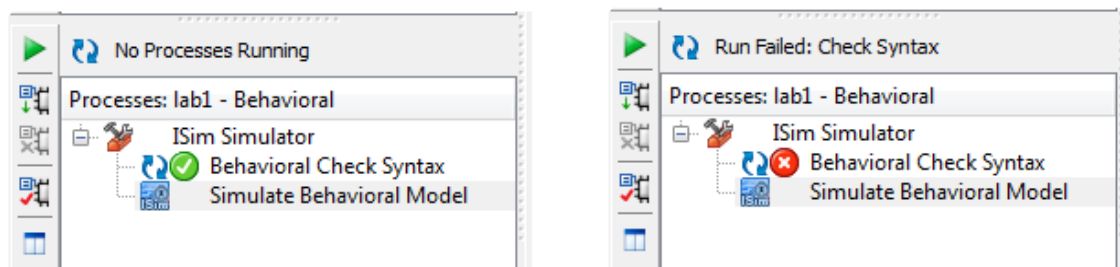
Εικόνα 8.

Έπειτα επιλέγουμε το .vhd αρχείο όπως βλέπουμε στο δεξιό μέρος της εικόνας και πατάμε διπλό click στο Behavioral Check Syntax.



Εικόνα 9.

Εάν η σύνταξη είναι σωστή το αποτέλεσμα θα είναι αυτό της αριστερής εικόνας αλλιώς αυτό της δεξιάς.



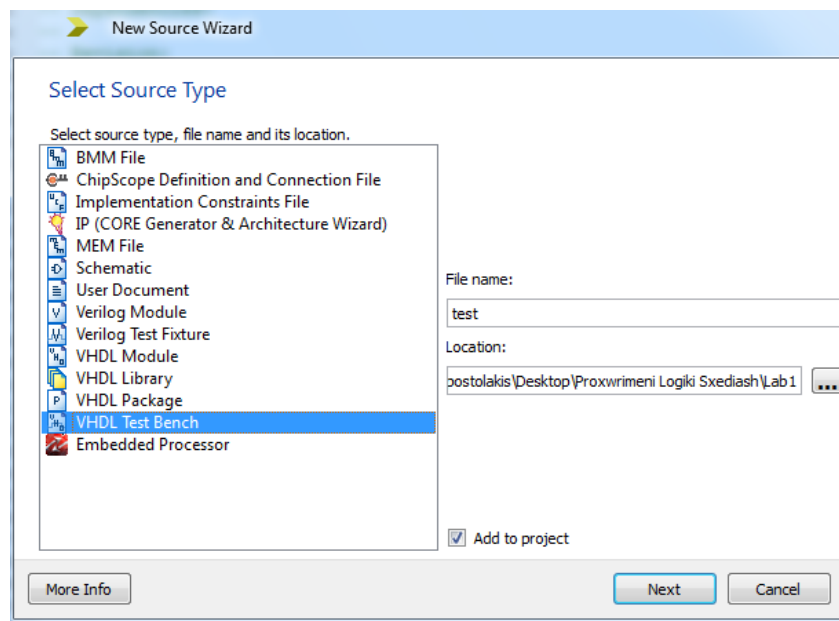
Εικόνα 10.

8. Προσομοίωση

Προσομοίωση είναι η διαδικασία κατά την οποία ελέγχουμε την ορθή λειτουργία ενός κυκλώματος με εργαλεία ECAD (Electronic Computer Aided Design). Πιο συγκεκριμένα δίνουμε τιμές στις εισόδους και εξακριβώνουμε ότι οι έξοδοι έχουν την αναμενόμενη τιμή σε συγκεκριμένο χρόνο. Στα εργαλεία της Xilinx υποστηρίζονται δύο τύποι προσομοίωσης η συμπεριφερική (behavioral simulation) και η μετά από τοποθέτηση και διασύνδεση (post place and route simulation). Στην πρώτη δε λαμβάνονται υπόψη οι χρόνοι απόκρισης του κυκλώματος ούτε των επιμέρους στοιχείων του ενώ στη δεύτερη το μοντέλο που προσομοιώνεται είναι πιο πλήρες αφού λαμβάνονται και οι χρόνοι υπόψη.

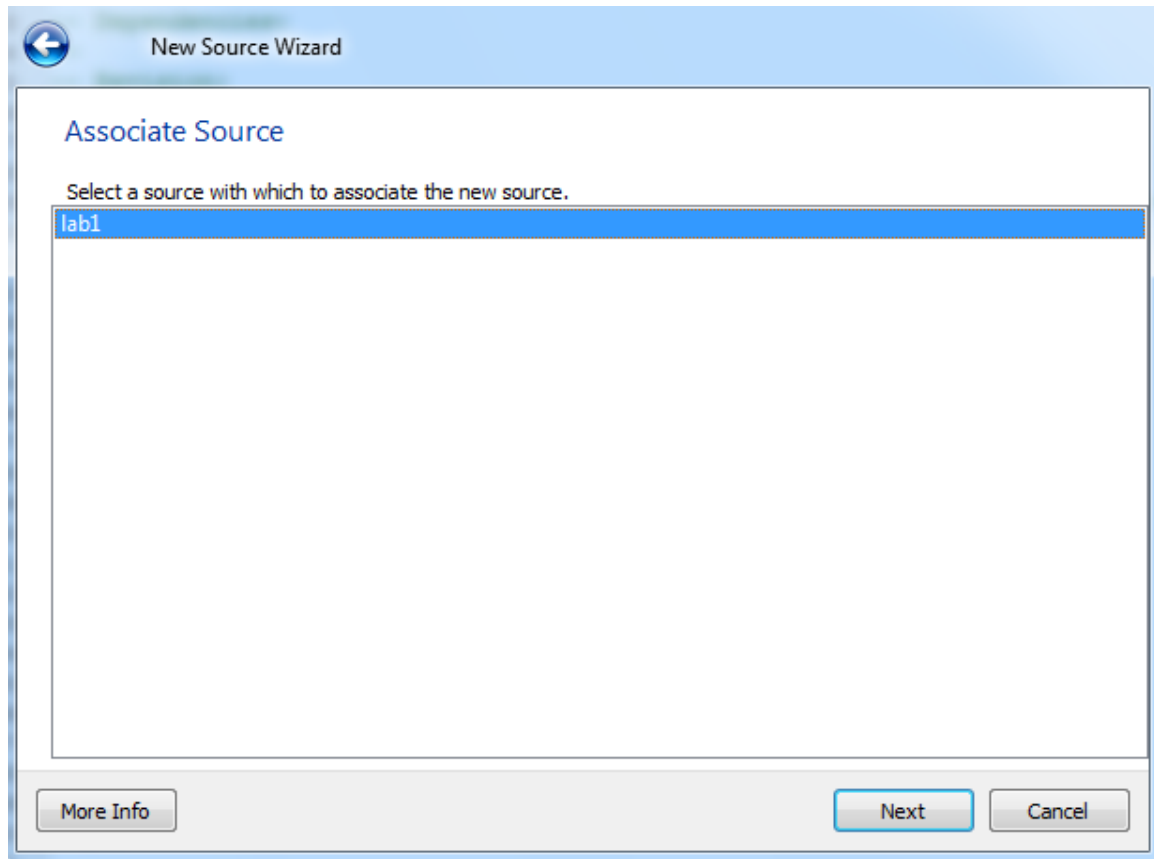
Για τον έλεγχο της ορθής λειτουργίας του κυκλώματος πρέπει να εκτελεστεί η προσομοίωση του. Για το λόγο αυτό γράφονται ειδικά VHDL αρχεία (Test Bench) με τα οποία την εκτελούμε με αυτοματοποιημένο τρόπο.

Δημιουργούμε ένα νέο αρχείο με την διαδικασία που αναφέραμε παραπάνω, το αρχείο test.vhd, μόνο που αυτή τη φορά δεν θα επιλέξουμε την επιλογή VHDL Module αλλά την επιλογή **Test Bench** όπως φαίνεται στην εικόνα 11.



Εικόνα 11: New Source Wizard

Στη συνέχεια μας ρωτάει με ποιο αχρείο συνδέεται το Test Bench και επιλέγουμε το lab1 (**Test Bench** εικόνα 12)



Εικόνα 12: New Source Wizard - Associate Source

Πατάμε **NEXT** και στη συνέχεια **Finish**

Σύνταξη Test Bench

Αρχικά βάζουμε σε σχόλια τις εντολές που βρίσκονται στα κόκκινα πλαίσια στην εικόνα 13, καθώς αποτελούν ρυθμίσεις για τα ακολουθιακά κυκλώματα που θα δούμε σε επόμενα εργαστήρια.

```
50
51  --Inputs
52  signal InData0 : std_logic := '0';
53  signal InData1 : std_logic := '0';
54
55  --Outputs
56  signal OutData : std_logic;
57  -- No clocks detected in port list. Replace <cloc
58  -- appropriate port name
59
60  --constant <clock>_period : time := 10 ns;
61
62  BEGIN
63
64  -- Instantiate the Unit Under Test (UUT)
65  uut: lab1 PORT MAP (
66      InData0 => InData0,
67      InData1 => InData1,
68      OutData => OutData
69  );
70
71  -- Clock process definitions
72  -- <clock>_process :process
73  -- begin
74  --   <clock> <= '0';
75  --   wait for <clock>_period/2;
76  --   <clock> <= '1';
77  --   wait for <clock>_period/2;
78  -- end process;
79
80
81  -- Stimulus process
82  stim_proc: process
83  begin
84      -- hold reset state for 100 ns.
```

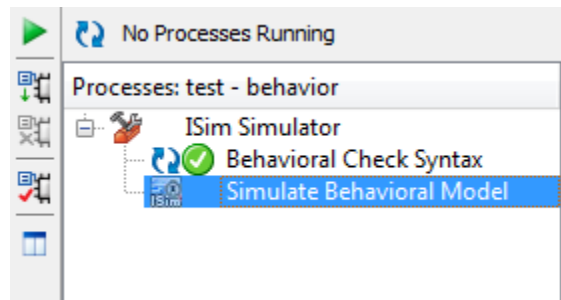
Εικόνα 13.

Έπειτα εισάγουμε των κώδικα που βρίσκεται στο μπλέ πλαίσιο όπως φαίνεται στην εικόνα 14.

```
80
81  -- Stimulus process
82  stim_proc: process
83  begin
84      -- hold reset state for 100 ns.
85      InData0 <= '0';
86      InData1 <= '0';
87      wait for 100 ns;
88
89      InData0 <= '0';
90      InData1 <= '1';
91      wait for 100 ns;
92
93      InData0 <= '1';
94      InData1 <= '0';
95      wait for 100 ns;
96
97      InData0 <= '1';
98      InData1 <= '1';
99      wait for 100 ns;
100
101      -- insert stimulus here
102      wait;
103  end process;
104
105  END;
```

Εικόνα 14.

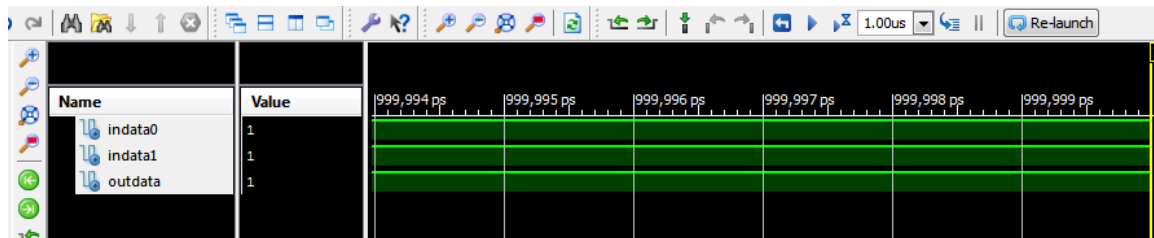
Έχοντας εισάγει τις αλλαγές μας στο Testbench πατάμε διπλό κλικ στην επιλογή Simulate Behavioral Model στο όπως φαίνεται στην εικόνα 15 για να εμφανιστεί η κυματομορφή.



Εικόνα 15.

Ανάγνωση κυματομορφής

Αφού πατήσουμε την επιλογή Simulate Behavioral Model μας εμφανίζεται η οθόνη του simulation.



Εικόνα 16: Κυματομορφή

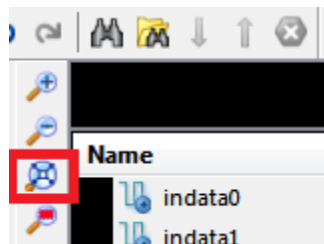
Εκεί φαίνονται οι είσοδοι και έξοδοι του κυκλώματος καθώς και οι τιμές τους τη δεδομένη χρονική στιγμή (εικόνα 17)

The screenshot shows a close-up of the data table in the ISim simulation interface. The table has the following data:

Name	Value
indata0	1
indata1	1
outdata	1

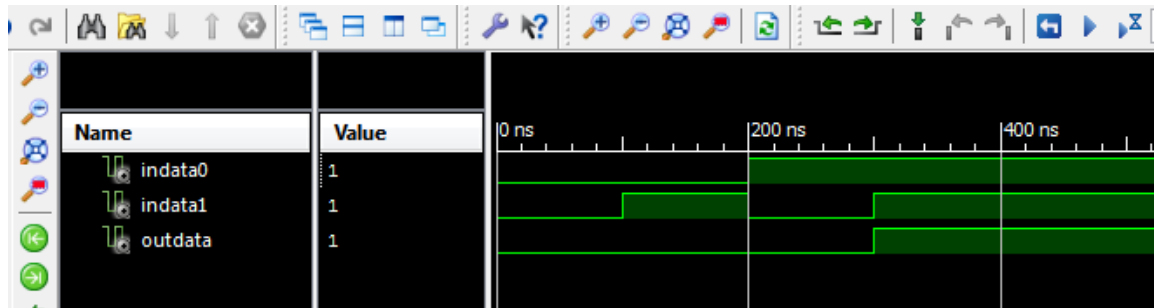
Εικόνα 17: Είσοδοι έξοδοι του κυκλώματος και τιμές τη δεδομένη χρονική στιγμή (μπλε γραμμή)

Επειδή το πρόγραμμα ISim τρέχει τις υλοποιήσεις για default χρόνο τα 1.00us μετά που θα εμφανιστεί η κυματομορφή στην μορφή της εικόνας 16 πατάμε το πλήκτρο zoom to full view που είναι σημειωμένο στην εικόνα 18 ώστε να εμφανιστεί η προσωμοίωση μας στους σωστούς χρόνους.



Εικόνα 18: Τα πλήκτρα zoom-in zoom-out και run zoom to full view

Τώρα μπορεί να φανεί το σύνολο της προσομοίωσης. (Εικόνα 19)



Εικόνα 19: Το σύνολο της προσομοίωσης

Παρατηρούμε τις εισόδους και στο ίδιο χρονικό σημείο την έξοδο. Εφόσον το αποτέλεσμα είναι το αναμενόμενο τότε έχουμε τελειώσει με την προσομοίωση επιτυχώς!!!

8. Δημιουργία αρχείου .UCF (User Constrain File)

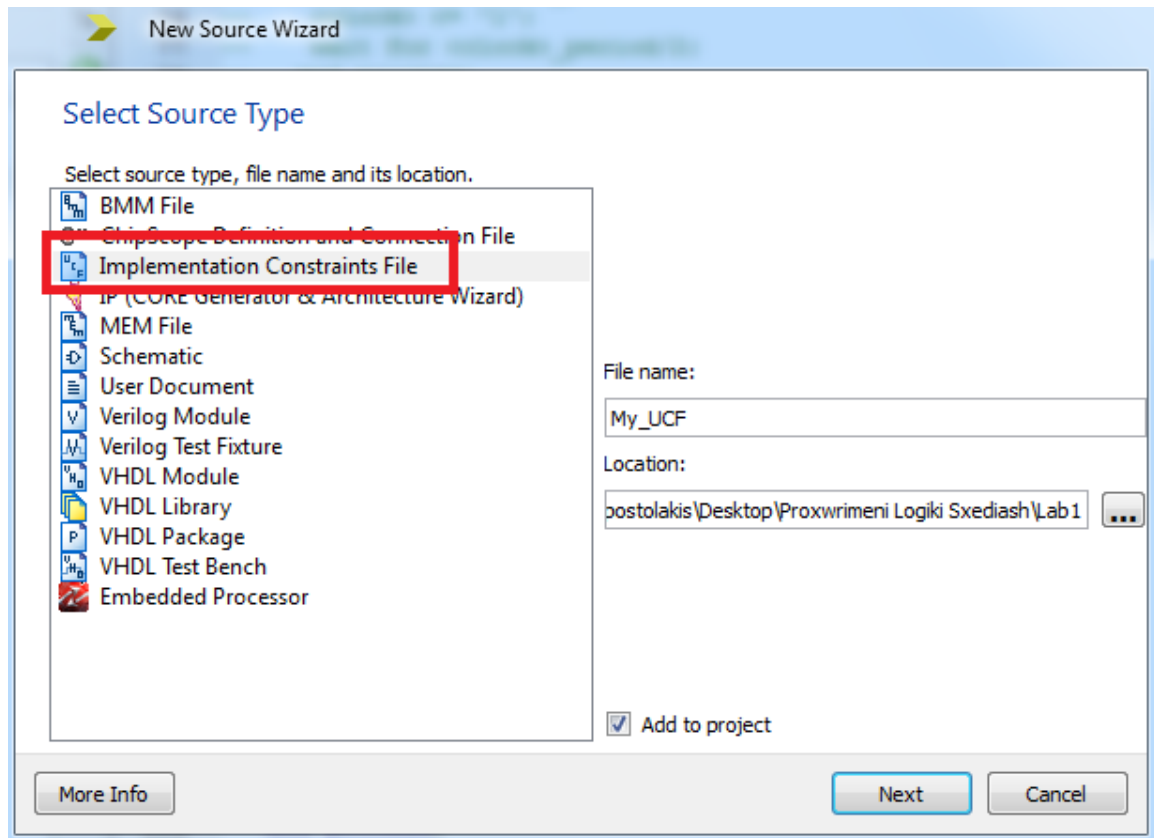
Τα αρχεία *.ucf* αντιστοιχούν συγκεκριμένα κομμάτια της σχεδίασης με συγκεκριμένους πόρους του ολοκληρωμένου κυκλώματος. Για παράδειγμα αντιστοιχούν ένα συγκεκριμένο pin του ολοκληρωμένου κυκλώματος με μία είσοδο ή έξοδο, ή ένα κομμάτι της λογικής της σχεδίασης το αντιστοιχούν με μια συγκεκριμένη περιοχή. Επίσης χρησιμοποιούνται για να δίνουν περιορισμούς που πρέπει να επιτύχει το εργαλείο που χρησιμοποιούμε πχ ταχύτητα ρολογιού. Στο εργαστήριο θα συντάξουμε *.ucf* αρχεία με σκοπό μόνο την αντιστοίχιση pins του ολοκληρωμένου κυκλώματος με τις εισόδους και τις εξόδους της σχεδίασης μας.

Για την συγκεκριμένη σχεδίαση το αρχείο *.UCF* αποτελείτε από τις τρεις παρακάτω γραμμές αποκλειστικά

```
NET "InData0" LOC = "P11" ;  
NET "InData1" LOC = "L3" ;  
NET "OutData" LOC = "M5" ;
```

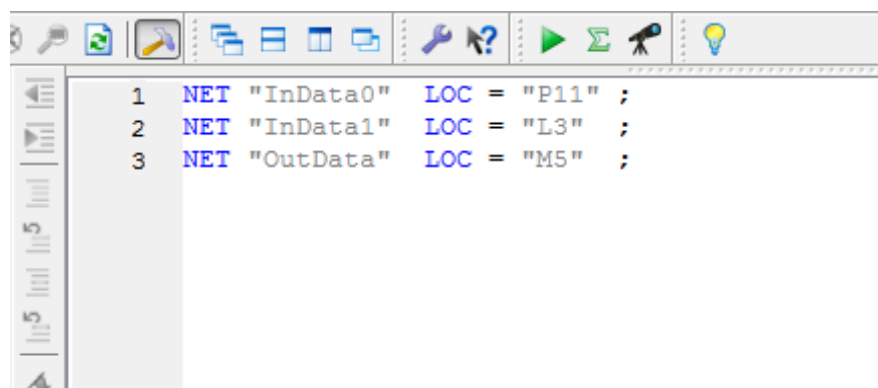
Οι οποίες αντιστοιχούν τα σήματα εισόδου και εξόδου με συγκεκριμένα pin του ολοκληρωμένου κυκλώματος.

Για την παραγωγή του .ucf αρχείου πάμε πάλι στο Project -> New Source και μας εμφανίζει το παράθυρο του New Source Wizard(εικόνα 20). Επιλέγουμε την επιλογή Implementation Constraints File και τις δίνουμε το όνομα π.χ My_UCF. Πατάμε Next και στο Summary πατάμε Finish.



Εικόνα 20.

Στη κενή σελίδα που δημιουργείτε με τίτλο My_UCF.ucf κάνουμε copy-paste της τρείς γραμμές κώδικα(εικόνα 21) που αναφέραμε πιο πάνω και πατάμε Save.



Εικόνα 21.

9. Σύνθεση (Synthesize)

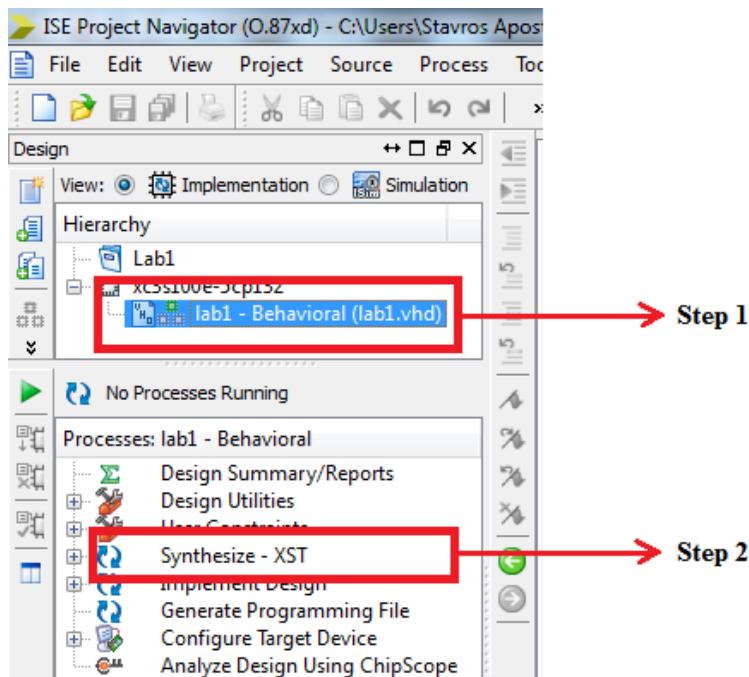
Σύνθεση είναι η διαδικασία κατά την οποία ο κώδικας (VHDL στη συγκεκριμένη περίπτωση) μετατρέπεται σε αρχείο ή αρχεία ενδιάμεσης μορφής (netlist). Το ενδιάμεσο αυτό αρχείο (ή αρχεία) απεικονίζει το κύκλωμα σαν ένα σύνολο πυλών συνδεδεμένα μεταξύ τους και εξαρτάται από την τεχνολογία στην οποία θα απεικονιστεί τελικά το κύκλωμα.

Το ζητούμενο είναι να γίνει Σύνθεση του αρχείου VHDL που έχει τον κώδικα (lab1.vhd) και όχι του αρχείου VHDL που έχει το test bench (test.vhd).

Σημείωση : Η VHDL χωρίζεται στην συνθέσιμη (Synthesizable) και στην μη συνθέσιμη (non Synthesizable). Τα πραγματικά συστήματα σχεδιάζονται με τη χρήση συνθέσιμης (Synthesizable) VHDL. Η μη συνθέσιμη (non Synthesizable) χρησιμοποιείται ως βοηθητική (πχ για προσομοίωση). Η εντολή wait είναι μια εντολή μη συνθέσιμη αφού δε μπορεί να αντιστοιχηθεί σε πραγματικό κύκλωμα.

Για να γίνει η σύνθεση κάνουμε μετάβαση από το Simulation view στο Implementation view με τον τρόπο που δείξαμε και στην εικόνα 8 προηγουμένως.

Στη συνέχεια επιλέγουμε το module lab1(αρχείο lab1.vhd) Step 1 και πατάμε διπλό click στο Synthesize - XST Step2. (εικόνα 22)



Εικόνα 22.

Όταν η διαδικασία τελειώσει με επιτυχία τότε εμφανίζεται ένα πράσινο σημάδι δίπλα στο **synthesize** και στην **console** εμφανίζεται το μήνυμα: «Process "Synthesis" completed successfully»

10. Implement Design

Η διαδικασία αυτή χωρίζεται σε τρία στάδια

A. Translate

Η διαδικασία της μετάφρασης (translate) δημιουργεί ένα ενιαίο αρχείο .NGD (native generic database) το οποίο περιλαμβάνει όλα τα αρχεία netlist και όλους τους περιορισμούς. Το αρχείο .NGD μπορεί να απεικονιστεί στο ολοκληρωμένο κύκλωμα που έχουμε επιλέξει.

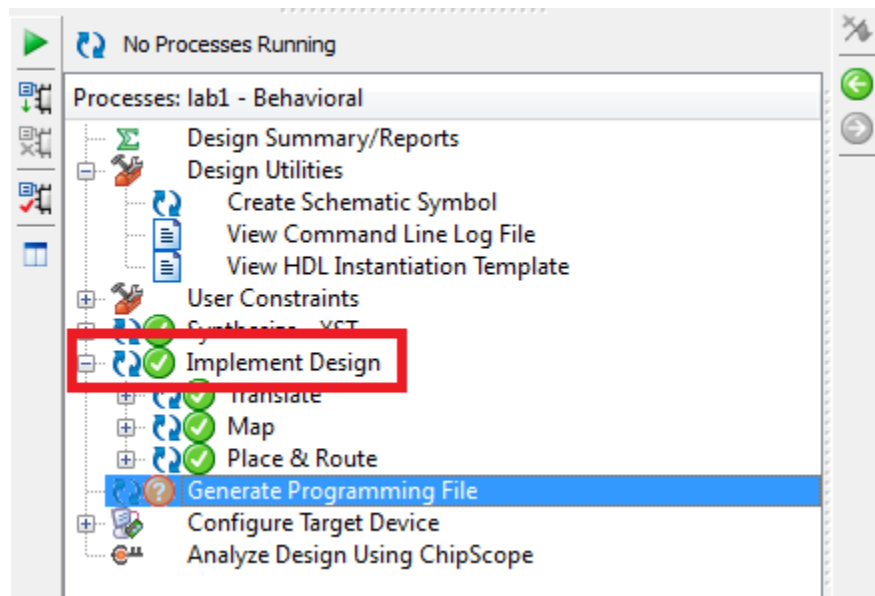
B. Map

Η διαδικασία απεικόνισης (Map) δέχεται ως είσοδο το αρχείο .NGD και ελέγχει αν είναι συμβατή η σχεδίαση με τους κανόνες σχεδίασης (DRC design rule check) και παράγει το αρχείο .NCD (Native Circuit Design)

C. Place and Route

Η διαδικασία τοποθέτησης και διασύνδεσης (Place and Route PAR) δέχεται ως είσοδο το αρχείο .NCD. Τοποθετεί με βέλτιστο τρόπο τη λογική στους αντίστοιχους πόρους λογικής και στη συνέχεια διασύνδεει τη λογική χρησιμοποιώντας τους αντίστοιχους πόρους διασύνδεσης.

Για την υλοποίηση και των τριών βημάτων αρκεί ένα διπλό click πάνω στο **Implement Design** όπως φαίνεται στην εικόνα 23. Εάν έχουν πάει όλα καλά τότε θα έχουμε επιτυχημένο Translate, Map και Place & Route.



Εικόνα 23.

11. Download

Η FPGA λειτουργεί σαν μία μνήμη η οποία φορτώνεται κατάλληλα συνεπώς το πρώτο βήμα για το download είναι η δημιουργία του αρχείου φόρτωσης.

Πατάμε διπλό click **Generate Programming File** και δημιουργείται το αρχείο .bit.

Για να γίνει η φόρτωση του ολοκληρωμένου πρέπει να συνδεθεί κατάλληλα το αναπτυξιακό με τον υπολογιστή μας και απλά να πατήσουμε το **Configure Target Device**. Λόγω της σχετικής ευαισθησίας η διαδικασία αυτή θα γίνεται με την βοήθεια του εργαστηριακού προσωπικού για την αποφυγή «ατυχημάτων».