

---

---

**Εργαστηριακές Σημειώσεις:  
Σύντομος Οδηγός Εκμάθησης  
του Λογισμικού Xilinx ISE**

---

---

**Μιχάλης Ψαράκης**

---

---

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ - ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΠΜΣ «ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»



## Πίνακας Περιεχομένων

1	Εισαγωγή.....	5
2	Επισκόπηση του παραθύρου Project Navigator .....	5
2.1	Παράθυρο Design .....	6
2.2	Παράθυρο Processes.....	6
2.3	Παράθυρο Console .....	7
3	Εισαγωγή σχεδίασης (design entry) .....	7
3.1	Δημιουργία νέου έργου (new project) .....	7
3.2	Σχεδίαση βασισμένη στη γλώσσα περιγραφής υλικού VHDL .....	10
3.3	Σχεδίαση βασισμένη σε σχηματική αναπαράσταση .....	13
4	Λειτουργική προσομοίωση (functional simulation) .....	17
5	Σύνθεση της σχεδίασης (design synthesis).....	26
6	Υλοποίηση της σχεδίασης (design implementation) .....	31
7	Χρονική προσομοίωση (timing simulation) .....	35
8	Προγραμματισμός του FPGA (FPGA programming) .....	36
9	Προγραμματισμός της PROM (PROM programming) .....	42

## Κατάλογος Εικόνων

Εικόνα 1: Ένα τυπικό παράθυρο του Project Navigator.....	6
Εικόνα 2: New Project Dialog (1 από 3) .....	8
Εικόνα 3: New Project Dialog (2 από 3) .....	9
Εικόνα 4: New Project Dialog (3 από 3) .....	9
Εικόνα 5: New Source Dialog (1 από 3).....	10
Εικόνα 6: New Source Dialog (2 από 3).....	11
Εικόνα 7: New Source Dialog (3 από 3).....	11
Εικόνα 8: Ολοκληρωμένη σχεδίαση.....	13
Εικόνα 9: New Source Dialog (1 από 2).....	13
Εικόνα 10: New Source Dialog (2 από 2).....	14
Εικόνα 11: Symbol Browser.....	15
Εικόνα 12: Τοποθέτηση συμβόλων στο σχηματικό διάγραμμα.....	16
Εικόνα 13: Καλωδίωση και τοποθέτηση ονομάτων στο σχηματικό διάγραμμα.....	17
Εικόνα 14: Τοποθέτηση buffer και I/O marker στο σχηματικό διάγραμμα.....	17
Εικόνα 15: Ρυθμίσεις του λογισμικού Xilinx ISE για τον προσομοιωτή Modelsim .....	18
Εικόνα 16: New Source Dialog (1 από 3).....	18
Εικόνα 17: New Source Dialog (2 από 3).....	19
Εικόνα 18: New Source Dialog (3 από 3).....	19
Εικόνα 19: Ολοκληρωμένο Test Bench.....	22
Εικόνα 20: Compile HDL Simulation Libraries .....	23
Εικόνα 21: Simulation Process Properties .....	23
Εικόνα 22: ModelSim Transcript Window .....	24
Εικόνα 23: Sim Window.....	24
Εικόνα 24: Objects Window .....	25
Εικόνα 25: Wave Window.....	25
Εικόνα 26: RTL and Technology Schematic View .....	31
Εικόνα 27: Look Up Table Dialog Box .....	31
Εικόνα 28: New Source Wizard (1 από 2).....	32
Εικόνα 29: New Source Wizard (2 από 2).....	33
Εικόνα 30: PlanAhead Software .....	33
Εικόνα 31 (α) Routed Design, (β) Component ep_out .....	35
Εικόνα 32: Simulation Process Properties .....	36
Εικόνα 33: Programming File Properties (Configuration Options) .....	37
Εικόνα 34: Programming File Properties (Startup Options) .....	37
Εικόνα 35: Τοποθέτηση jumper της αναπτυξιακής πλακέτας .....	38
Εικόνα 36: Μήνυμα σχετικά με το iMPACT αρχείο .....	38
Εικόνα 37: Εργαλείο προγραμματισμού iMPACT .....	39
Εικόνα 38: Επιλογή του αρχείου προγραμματισμού του FPGA.....	39
Εικόνα 39: Παράκαμψη του αρχείου προγραμματισμού της PROM .....	40
Εικόνα 40: Τελικές ρυθμίσεις προγραμματισμού του FPGA .....	40
Εικόνα 41: Εναλλακτική μέθοδος ανάθεσης των αρχείων προγραμματισμού .....	41
Εικόνα 42: Program device.....	41
Εικόνα 43: Βήματα καθορισμού αρχείου προγραμματισμού της PROM.....	42
Εικόνα 44: Τελικό αποτέλεσμα δημιουργίας αρχείου για την PROM.....	43
Εικόνα 45: Ανάθεση νέου αρχείου προγραμματισμού στην PROM .....	43
Εικόνα 46: Ορισμός τύπου της PROM.....	44
Εικόνα 47: Ορισμός παραμέτρων προγραμματισμού της PROM.....	44
Εικόνα 48: Παράμετροι προγραμματισμού της PROM.....	45

## 1 Εισαγωγή

Ο σύντομος αυτός οδηγός θα σας βοηθήσει στην εκμάθηση του λογισμικού ISE της εταιρείας Xilinx για τη σχεδίαση και ανάπτυξη ψηφιακών κυκλωμάτων με τη χρήση FPGAs. Ο βασικός στόχος του οδηγού εκμάθησης είναι να εξοικειωθείτε με τα εργαλεία που θα χρησιμοποιήσετε για την υλοποίηση των εργασιών του μαθήματος «Προηγμένη Ψηφιακή Σχεδίαση».

Ο οδηγός εκμάθησης περιγράφει διαφορετικά στάδια της ροής σχεδίασης ψηφιακών κυκλωμάτων με τη χρήση του λογισμικού Xilinx ISE:

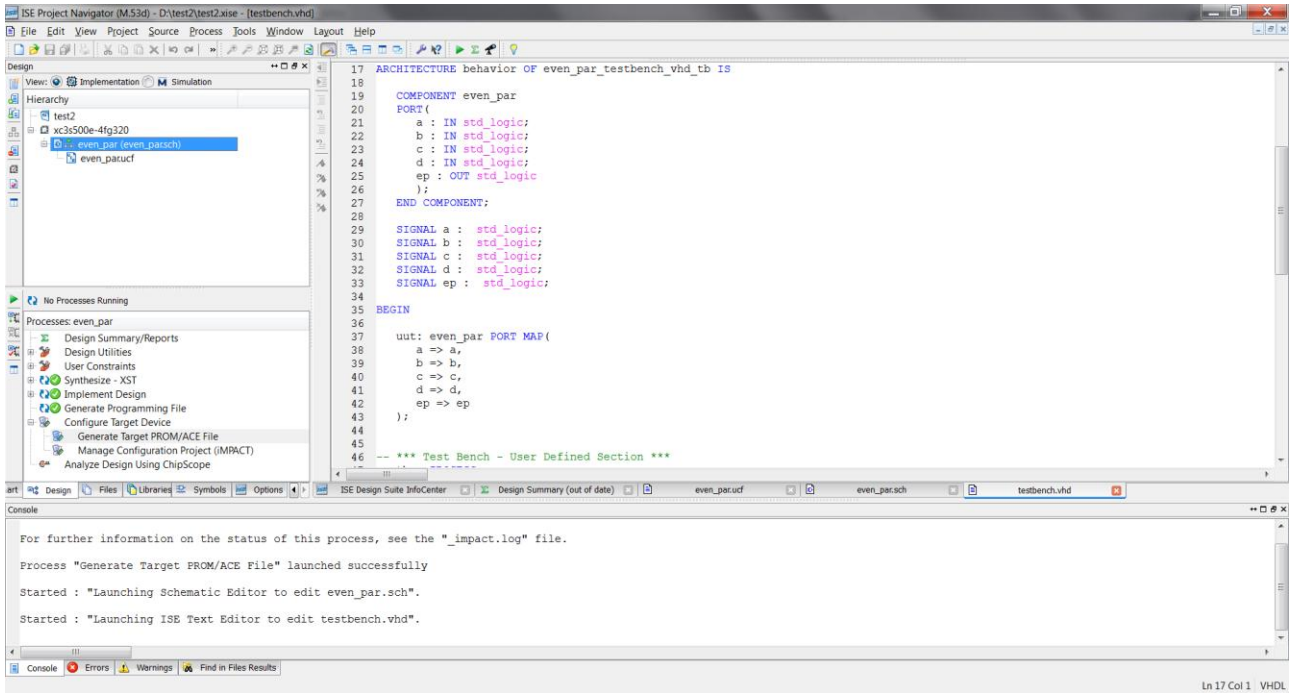
- Πως θα χρησιμοποιήσετε τα εργαλεία εισαγωγής της σχεδίασης (design entry tools) που βασίζονται είτε στη χρήση μιας γλώσσας περιγραφής υλικού (π.χ. VHDL) είτε στη χρήση σχηματικού διαγράμματος
- Πως θα εκτελέσετε λειτουργική προσομοίωση (functional simulation) ή χρονική προσομοίωση (timing simulation)
- Πως θα συνθέσετε και θα υλοποιήσετε μια σχεδίαση
- Πως θα προγραμματίσετε τη συσκευή FPGA

Σημείωση: Τα παραδείγματα που χρησιμοποιούνται στον οδηγό εκμάθησης αφορούν την αναπτυξιακή πλατφόρμα Spartan-3E Starter Kit (είναι η πλατφόρμα που θα χρησιμοποιήσετε για την υλοποίηση των εργασιών του μαθήματος).

Σημείωση: Ο οδηγός εκμάθησης αφορά την έκδοση του εργαλείου ISE 12.x.

## 2 Επισκόπηση του παραθύρου Project Navigator

Το παράθυρο Project Navigator διαιρείται σε τέσσερα κύρια υπο-παράθυρα, όπως φαίνεται στην Εικόνα 1. Πάνω αριστερά είναι το παράθυρο Design που απεικονίζει ιεραρχικά τα στοιχεία που περιλαμβάνονται στο έργο και έχει τις επιλογές εμφάνισης Implementation και Simulation. Κάτω από το παράθυρο Design είναι το παράθυρο Processes που απεικονίζει τις διαθέσιμες διεργασίες για την τρέχουσα επιλεγμένη πηγή. Το τρίτο παράθυρο στο κάτω μέρος του Project Navigator είναι το παράθυρο Console που απεικονίζει μηνύματα κατάστασης (status messages), λάθη (errors), και προειδοποιήσεις (warnings), και που ενημερώνεται κατά τη διάρκεια όλων των ενεργειών του έργου. Το τέταρτο παράθυρο στα δεξιά είναι για την επεξεργασία και την ανάγνωση αρχείων (σχεδίασης ή αναφοράς). Κάθε παράθυρο μπορεί να αλλάξει μέγεθος, να μετακινηθεί εκτός του Project Navigator ή σε μια νέα θέση μέσα στο κύριο παράθυρο.



Εικόνα 1: Ένα τυπικό παράθυρο του Project Navigator

## 2.1 Παράθυρο Design

Το παράθυρο Design παρέχει πρόσβαση στην Απεικόνιση (View), στην Ιεραρχία (Hierarchy) και στις Διεργασίες (Processes) του ISE. Κάθε ετικέτα αναλύεται με περισσότερες λεπτομέρειες:

- *View Panel*: τα κουμπιά του view panel επιτρέπουν την εμφάνιση των πηγαίων αρχείων στο Hierarchy Panel, τα οποία συσχετίζονται με την υλοποίηση (Implementation) ή την προσομοίωση (Simulation) της εκάστοτε σχεδίασης. Αν επιλέξετε την προσομοίωση, πρέπει να ορίσετε και τον τύπο της από το αντίστοιχο μενού.
- *Hierarchy Panel*: απεικονίζει το όνομα του έργου, τα έγγραφα του χρήστη, τον καθορισμένο τύπο της συσκευής FPGA, και τα πηγαία αρχεία της σχεδίασης. Κάθε αρχείο στο Module View συνδέεται με ένα εικονίδιο, το οποίο υποδηλώνει τον τύπο του αρχείου (π.χ. αρχείο VHDL ή αρχείο κειμένου). Για μια πλήρη λίστα των πιθανών τύπων αρχείου και των σχετικών εικονιδίων, δείτε την απευθείας βοήθεια του Project Navigator (επιλέξτε Help→Help Topics). Εάν ένα αρχείο περιέχει χαμηλότερα επίπεδα ιεραρχίας, το εικονίδιο έχει ένα + στα αριστερά του ονόματος. Τα αρχεία VHDL έχουν το + για να υποδηλώσουν ότι περιέχουν άλλες λειτουργικές μονάδες μέσα στο αρχείο. Μπορείτε να ξεδιπλώσετε την ιεραρχία πατώντας το +. Μπορείτε να ανοίξετε ένα αρχείο για τροποποίηση κάνοντας διπλό κλικ στο όνομα του αρχείου.
- *Processes Panel*: το παράθυρο αυτό αναλύεται στην επόμενη ενότητα.

## 2.2 Παράθυρο Processes

Τα περιεχόμενα του παραθύρου Processes εξαρτώνται από την επιλογή Implementation ή Simulation αλλά αλλάζουν κάθε φορά και ανάλογα με τον τύπο του πηγαίου αρχείου που έχει επιλεγεί στο παράθυρο Design. Από το παράθυρο Processes, μπορείτε να εκτελέσετε τις απαραίτητες λειτουργίες ώστε να καθορίσετε τις παραμέτρους, να εκτελέσετε (προσομοίωση ή σύνθεση) και να δείτε τη σχεδίασή σας. Το παράθυρο Processes παρέχει πρόσβαση στις ακόλουθες λειτουργίες:

- *Design Utilities* (εργαλεία εισαγωγής της σχεδίασης)  
Παρέχει πρόσβαση στην παραγωγή συμβόλων (Create Schematic Symbol), στα Log αρχεία του Command Line, στο Check Design Rules, κ.α.
- *User Constraints* (περιορισμοί του χρήστη)

Παρέχει πρόσβαση στον καθορισμό περιορισμών του χρήστη, π.χ. περιορισμοί θέσης και χρονισμού.

- *Synthesize – XST (σύνθεση)*

Παρέχει πρόσβαση στην απεικόνιση του σχηματικού διαγράμματος σε επίπεδο RT (View RTL Schematic) ή με λεπτομέρειες της τεχνολογίας (View Technology Schematic) και στην παραγωγή μοντέλου προσομοίωσης μετά-τη-σύνθεση (Post-Synthesis Simulation Model). Αυτό ποικίλλει ανάλογα με το εργαλείο σύνθεσης που χρησιμοποιείτε.

- *Implement Design (υλοποίηση της σχεδίασης)*

Παρέχει πρόσβαση στα εργαλεία υλοποίησης (Translate, Map και Place & Root) και στα αντίστοιχα αρχεία αναφορών.

- *Generate Programming File (παραγωγή του αρχείου προγραμματισμού)*

Παρέχει πρόσβαση στα εργαλεία διαμόρφωσης (configuration tools) και παραγωγής του αρχείου προγραμματισμού (bitstream generation).

Το παράθυρο Processes ενσωματώνει την τεχνολογία automake. Αυτό επιτρέπει στο χρήστη να επιλέξει οποιαδήποτε διεργασία στη ροή και το λογισμικό εκτελεί αυτόματα όλες τις απαραίτητες διεργασίες ώστε να φτάσει η ροή στο επιθυμητό βήμα. Για παράδειγμα, εάν εκτελέσετε τη διεργασία Implementation, το εργαλείο εκτελεί επίσης τη διεργασία Synthesis, εάν είναι απαραίτητο, επειδή η διεργασία υλοποίησης εξαρτάται από τα ενημερωμένα αποτελέσματα της διεργασίας σύνθεσης.

## 2.3 Παράθυρο Console

Το παράθυρο Console απεικονίζει λάθη, προειδοποιήσεις, και ενημερωτικά μηνύματα. Τα λάθη υποδηλώνονται από ένα κόκκινο X δίπλα στο μήνυμα, ενώ οι προειδοποιήσεις έχουν ένα κίτρινο (!). Τα μηνύματα προειδοποίησης και λάθους μπορούν επίσης να αντιμετωπισθούν χωριστά από τα άλλα μηνύματα κειμένου της κονσόλας με την επιλογή είτε της ετικέτας Warnings είτε της ετικέτας Errors στο κάτω μέρος του παραθύρου Console.

Μπορείτε να πλοηγηθείτε από ένα λάθος σύνθεσης ή ένα μήνυμα προειδοποίησης στο παράθυρο Console στη θέση του λάθους στο πηγαίο αρχείο πηγής HDL. Για να το επιτύχετε, επιλέξτε το μήνυμα λάθους ή προειδοποίησης, κάντε δεξί-κλικ στο ποντίκι, και από το μενού επιλέξτε **Goto Source**. Το πηγαίο αρχείο HDL ανοίγει και ο δρομέας μετακινείται στη γραμμή με το λάθος.

Μπορείτε επίσης να πλοηγηθείτε από ένα μήνυμα λάθους ή προειδοποίησης στο παράθυρο Console στα σχετικά έγγραφα λύσεων στις ιστοσελίδες υποστήριξης της Xilinx. Αυτοί οι τύποι λαθών ή προειδοποιήσεων μπορούν να προσδιοριστούν από το εικονίδιο στα αριστερά του λάθους. Για να πλοηγηθείτε στο έγγραφο λύσης, επιλέξτε το μήνυμα λάθους ή προειδοποίησης, κάντε δεξί-κλικ στο ποντίκι, και από το μενού επιλέξτε Go to Answer Record. Ο φυλλομετρητής Ιστού (web browser) ανοίγει και απεικονίζει όλα τα έγγραφα λύσεων που εφαρμόζονται σε αυτό το μήνυμα.

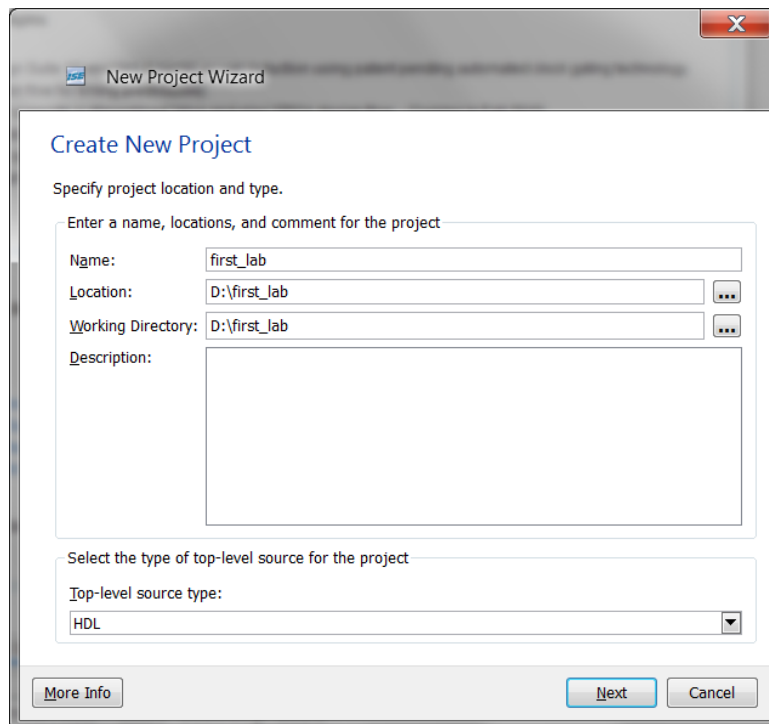
Στο τέταρτο παράθυρο, μπορείτε να έχετε πρόσβαση στον επεξεργαστή κειμένων (ISE Text Editor), στα πρότυπα γλώσσας (ISE Language Templates), και στο επεξεργαστή κειμένων προσομοίωσης (HDL Bench Text Editor). Ο επεξεργαστής κειμένων σας επιτρέπει να επεξεργαστείτε πηγαία αρχεία και να προσπελάσετε γλωσσικά πρότυπα (π.χ. HDL template files, User Constraint Files). Μπορείτε να χρησιμοποιήσετε και να τροποποιήσετε αυτά τα πρότυπα για τη σχεδίασή σας.

## 3 Εισαγωγή σχεδίασης (design entry)

Η σχεδίαση που χρησιμοποιείται σε αυτόν τον οδηγό εκμάθησης είναι ένα απλό κύκλωμα υπολογισμού της άρτιας ισοτιμίας ενός μηνύματος των 4-bit. Το κύκλωμα υλοποιείται με ένα δέντρο 3 πυλών XOR των 2-εισόδων. Η σχεδίαση θα περιγραφεί αρχικά με VHDL και στη συνέχεια με σχηματικό διάγραμμα.

### 3.1 Δημιουργία νέου έργου (new project)

Κάντε διπλό-κλικ στο εικονίδιο Project Navigator στην επιφάνεια εργασίας ή επιλέξτε Start→Programs→Xilinx ISE→Project Navigator. Από το Project Navigator, επιλέξτε File→New Project. Θα εμφανιστεί το πρώτο πλαίσιο διαλόγου του New Project, όπως φαίνεται στην Εικόνα 2.



**Εικόνα 2: New Project Dialog (1 από 3)**

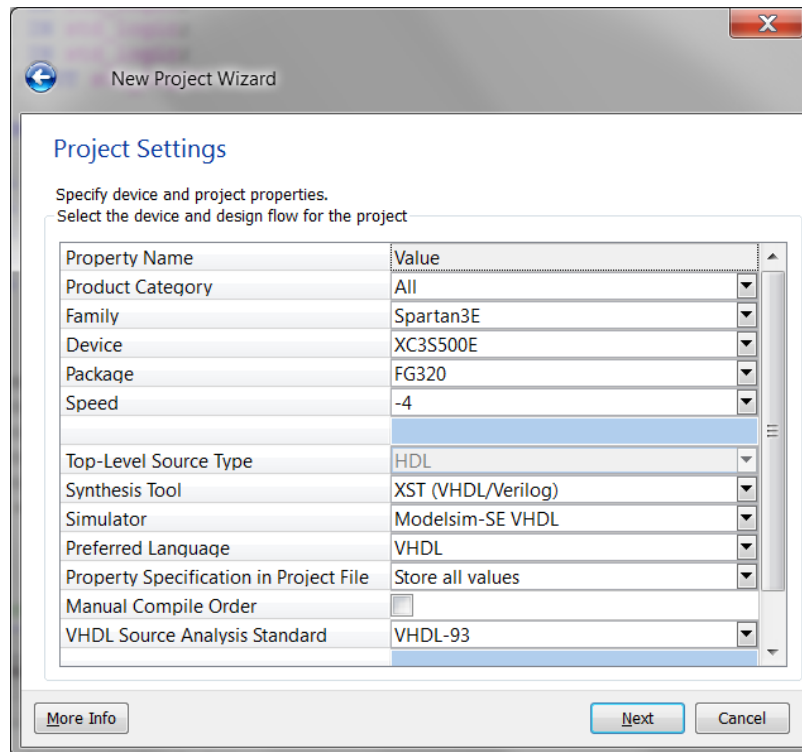
Το πλαίσιο διαλόγου σας προτρέπει να εισάγετε το όνομα του έργου, τη θέση του έργου, και τον τύπο της μονάδας του πιο υψηλού επιπέδου της σχεδίασης, όπως φαίνεται στην Εικόνα 2. Αφού συμπληρώσετε τα στοιχεία, πατήστε Next.

Σημείωση: Μην χρησιμοποιείτε ονόματα αρχείων ή φακέλων που περιέχουν διαστήματα.

Σημείωση: Δεν είναι απαραίτητο οι μονάδες στο υψηλότερο επίπεδο και στα χαμηλότερα επίπεδα να είναι του ίδιου τύπου. Για παράδειγμα, θα μπορούσατε να επιλέξετε τύπο Schematic για το top-level module και τύπο HDL για τα lower-level modules ή και το αντίστροφο. Στο συγκεκριμένο παράδειγμα επιλέγουμε τύπο HDL για όλες τις μονάδες.

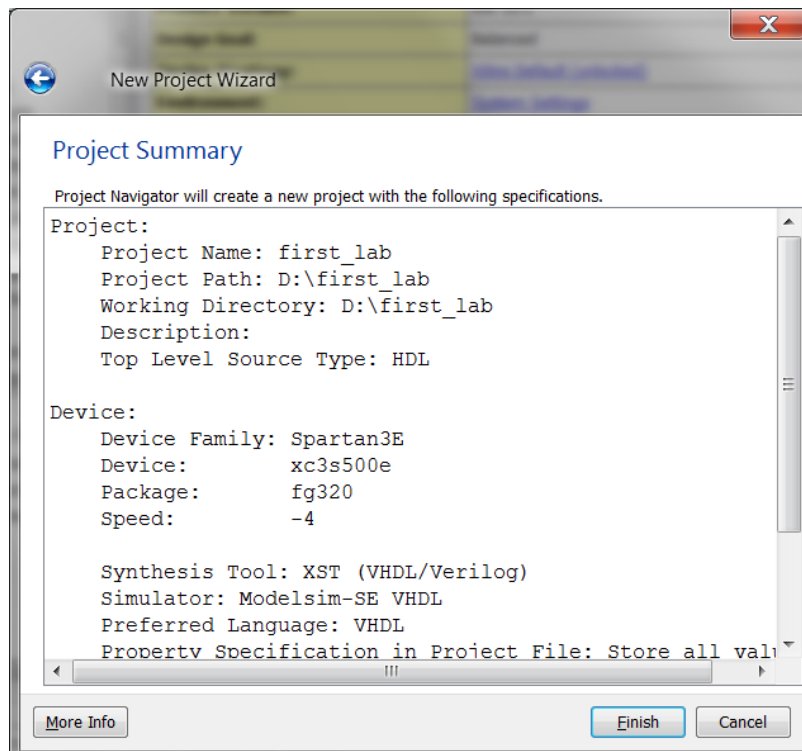
Το επόμενο πλαίσιο διαλόγου σας επιτρέπει να θέσετε τις πρόσθετες επιλογές του έργου. Η πρώτη ομάδα ρυθμίσεων καθορίζει τον τύπο της συσκευής FPGA που θα χρησιμοποιήσετε. Οι ρυθμίσεις που παρουσιάζονται στην Εικόνα 3 αντιπροσωπεύουν τη συσκευή FPGA που βρίσκεται στην αναπτυξιακή πλατφόρμα Spartan-3E Starter Kit. Η δεύτερη ομάδα ρυθμίσεων αντιπροσωπεύει τον τρόπο εισαγωγής της σχεδίασης, το εργαλείο σύνθεσης, και τον προσομοιωτή που θα χρησιμοποιήσετε. Αφού θέσετε τις ρυθμίσεις όπως φαίνεται στην Εικόνα 3, πατήστε Next.





Εικόνα 3: New Project Dialog (2 από 3)

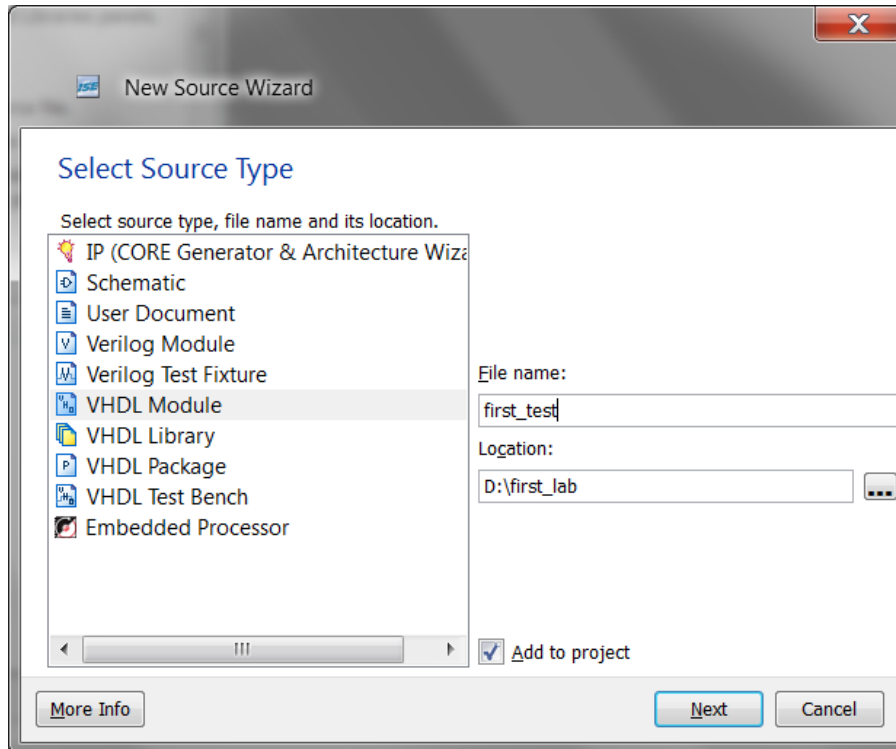
Το τελικό πλαίσιο διαλόγου στη διαδικασία δημιουργίας νέου έργου, που φαίνεται στην Εικόνα 4, παρέχει μια περίληψη του έργου που το Project Navigator θα δημιουργήσει βασισμένο στις ρυθμίσεις σας. Ελέγξτε την περίληψη για να σιγουρευτείτε ότι ταιριάζει με ότι φαίνεται στην Εικόνα 4. Εάν όχι, πατήστε Back για να διορθώσετε οποιοδήποτε λάθος. Διαφορετικά, πατήστε Finish για να ολοκληρώσετε τη διαδικασία.



Εικόνα 4: New Project Dialog (3 από 3)

### 3.2 Σχεδίαση βασισμένη στη γλώσσα περιγραφής υλικού VHDL

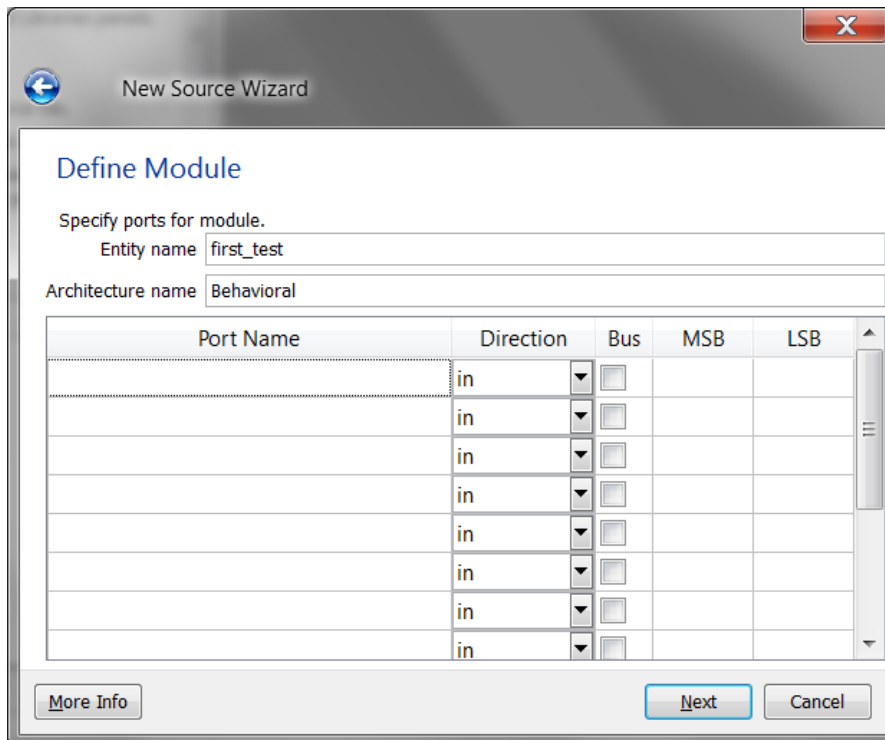
Σε αυτό το σημείο, το έργο που έχει δημιουργηθεί δεν περιέχει κανένα πηγαίο αρχείο. Δημιουργήστε ένα νέο πηγαίο αρχείο για τη σχεδίαση του κυκλώματος υπολογισμού άρτιας ισοτιμίας. Επιλέξτε Project→New Source από το κυρίως μενού. Το πρώτο από τα νέα κουτιά διαλόγου θα εμφανιστεί, όπως φαίνεται στην Εικόνα 5.



Εικόνα 5: New Source Dialog (1 από 3)

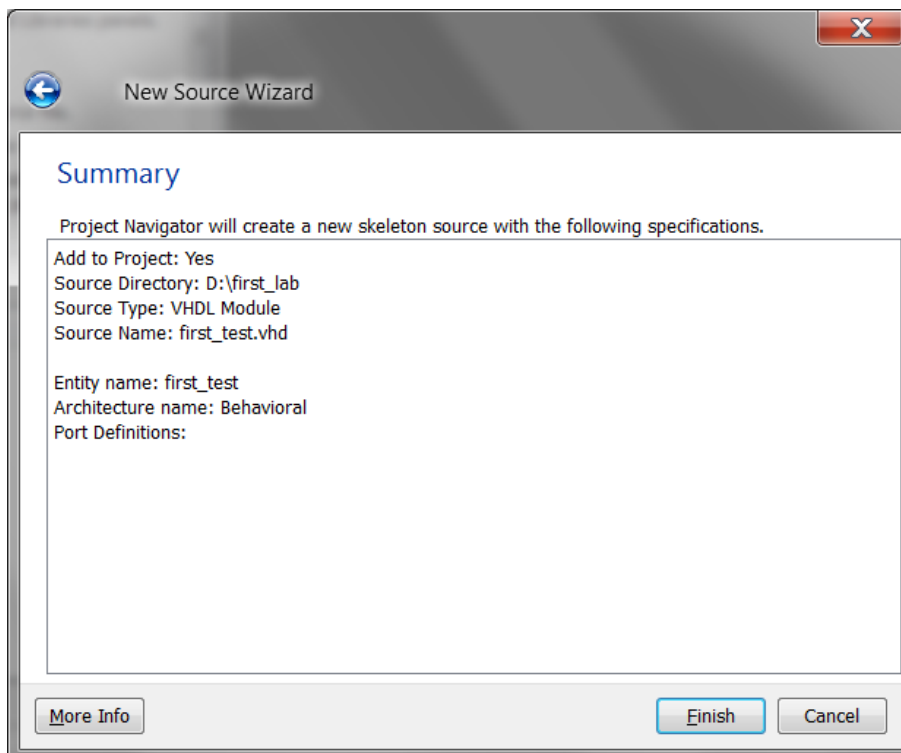
Επιλέξτε VHDL Module για να δηλώσετε ότι δημιουργείτε μια μονάδα σχεδίασης βασισμένη στη γλώσσα VHDL. Κατόπιν, δώστε ένα όνομα αρχείου όπως φαίνεται στην Εικόνα 5. Δεν χρειάζεται να αλλάξετε τη θέση του αρχείου, η οποία θα είναι μέσα στον κατάλογο του έργου που δημιουργήσατε προηγουμένως. Πατήστε Next.

Το επόμενο πλαίσιο διαλόγου σας επιτρέπει (προαιρετικά) να καθορίσετε τις θύρες (ports) της μονάδας. Αυτό μπορεί επίσης να γίνει στον επεξεργαστή κειμένου, κατά την επεξεργασία της μονάδας, έτσι αγνοήστε το σε αυτή τη φάση. Απλά επιβεβαιώστε ότι οι ρυθμίσεις ταιριάζουν με εκείνες που φαίνονται στην Εικόνα 6 και πατήστε Next.



Εικόνα 6: New Source Dialog (2 από 3)

Το τελικό πλαίσιο διαλόγου (Εικόνα 7) παρέχει μια περίληψη του πηγαίου αρχείου που θα δημιουργήσει το Project Navigator βασισμένο στις ρυθμίσεις σας. Ελέγξτε την περίληψη για να σιγουρευτείτε ότι ταιριάζει με την Εικόνα 7. Εάν όχι, πατήστε Back για να διορθώσετε οποιοδήποτε λάθος. Διαφορετικά, πατήστε Finish για να ολοκληρώσετε τη διαδικασία. Το νέο πηγαίο αρχείο θα ανοίξει αυτόματα στον επεξεργαστή κειμένου.



Εικόνα 7: New Source Dialog (3 από 3)

Στον επεξεργαστή κειμένου, μερικές από τις βασικές δομές του VHDL αρχείου είναι ήδη γραμμένες. Οι λέξεις κλειδιά (keywords) της γλώσσας απεικονίζονται με μπλε χρώμα, οι τύποι δεδομένων με κόκκινο, τα σχόλια με πράσινο, και οι τιμές με μαύρο. Αυτή η κωδικοποίηση ανάλογα με το χρώμα ενισχύει την αναγνωσιμότητα του VHDL αρχείου και την εύρεση τυπογραφικών λαθών. Τώρα, εισάγετε την περιγραφή του κυκλώματος υπολογισμού άρτιας ισοτιμίας. Αντιγράψτε τα περιεχόμενα του αρχείου, όπως δίνονται παρακάτω:

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

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;

entity even_par is
    port( a,b,c,d :    in std_logic;
          ep :    out std_logic);
end even_par;

architecture Behavioral of even_par is

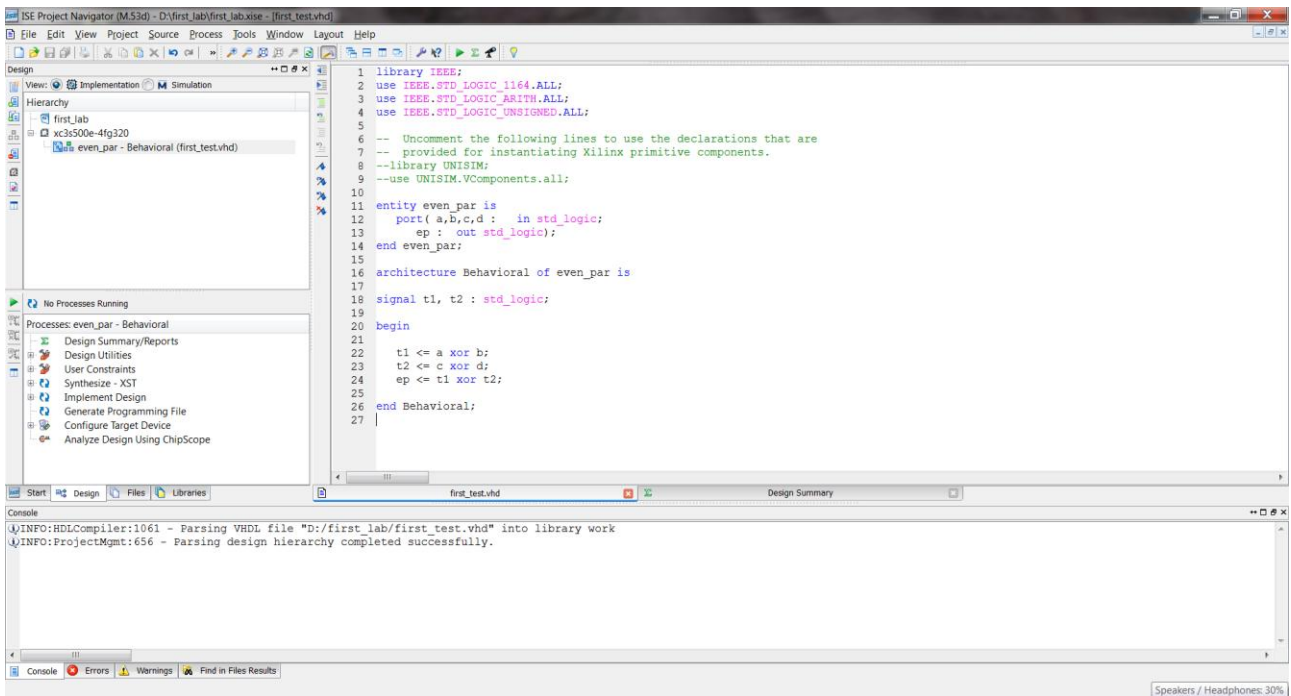
    signal t1, t2 : std_logic;

begin

    t1 <= a xor b;
    t2 <= c xor d;
    ep <= t1 xor t2;

end Behavioral;
```

Σε αυτό το σημείο, πρέπει να καταλήξετε με ένα παράθυρο που μοιάζει με αυτό που φαίνεται στην Εικόνα 8. Μόλις τελειώσετε, αποθηκεύστε το αρχείο και κλείστε το παράθυρο. Υπάρχουν επιλογές στο κυρίως μενού για να σώσετε είτε μεμονωμένα αρχεία είτε όλο το έργο.



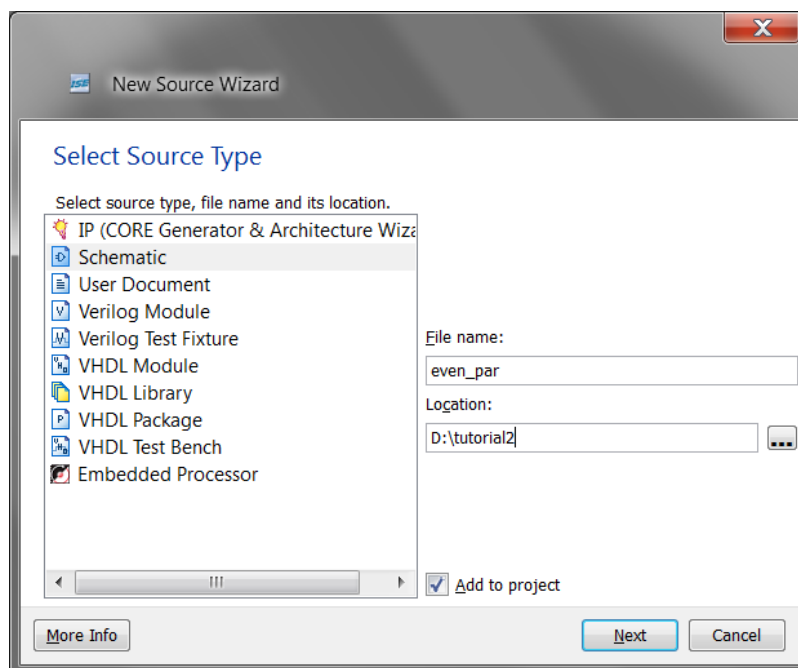
Εικόνα 8: Ολοκληρωμένη σχεδίαση

### 3.3 Σχεδίαση βασισμένη σε σχηματική αναπαράσταση

Τα πηγαία αρχεία του έργου μπορεί να είναι είτε HDL αρχεία είτε σχηματικά διαγράμματα (schematic diagrams). Ιεραρχικές σχεδιάσεις μπορεί να ακολουθούν μεικτή σχεδίαση περιέχοντας και HDL αρχεία και σχηματικά διαγράμματα. Σε αυτήν την ενότητα θα δημιουργήσετε ένα νέο σχηματικό διάγραμμα για τη σχεδίαση του κυκλώματος υπολογισμού άρτιας ισοτιμίας ενός μηνύματος 4-bit.

**Σημείωση:** Δημιουργήστε ένα νέο έργο (με όνομα tutorial2) και επιλέξτε τύπο Schematic για το top-level module.

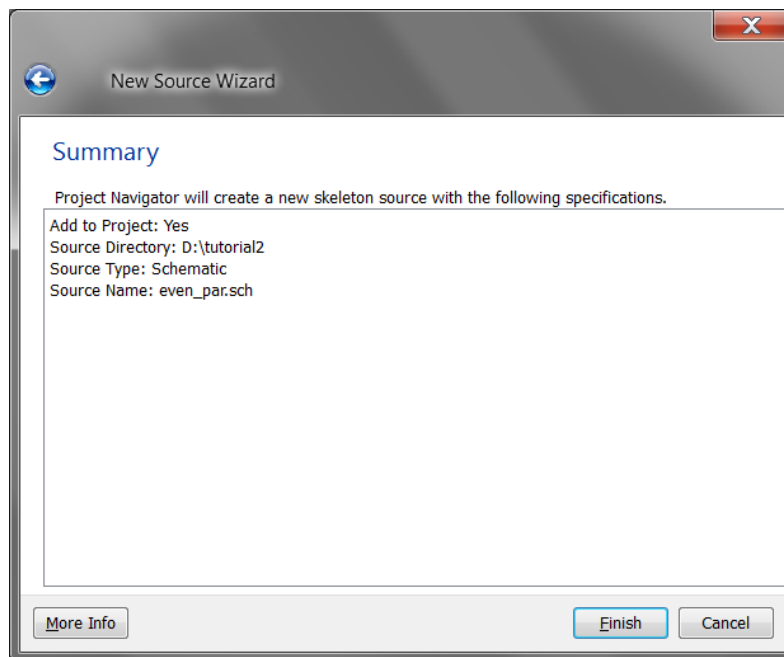
Επιλέξτε Project→New Source από το κυρίως μενού. Το πρώτο από τα νέα κουτιά διαλόγου θα εμφανιστεί, όπως φαίνεται στην Εικόνα 9.



Εικόνα 9: New Source Dialog (1 από 2)

Επιλέξτε Schematic για να δηλώσετε ότι δημιουργείτε μια μονάδα σχεδίασης βασισμένη σε σχηματική αναπαράσταση. Κατόπιν, δώστε ένα όνομα αρχείου όπως φαίνεται στην Εικόνα 9. Δεν χρειάζεται να αλλάξετε τη θέση του αρχείου, η οποία θα είναι μέσα στον κατάλογο του έργου που δημιουργήσατε προηγουμένως. Πατήστε Next.

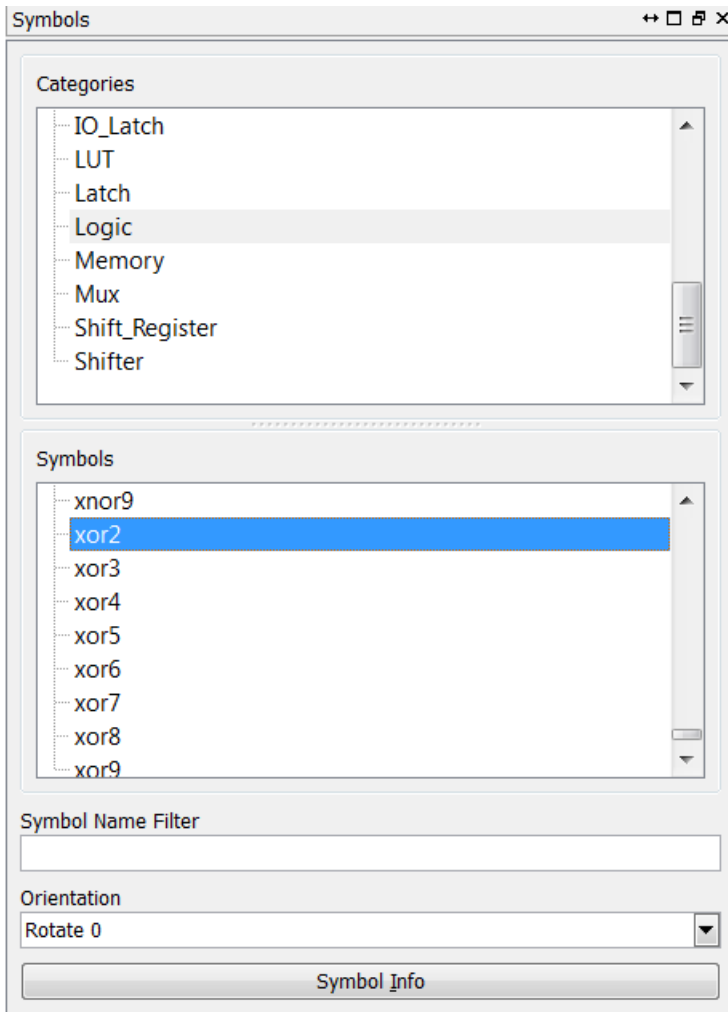
Το δεύτερο πλαίσιο διαλόγου (Εικόνα 10) παρέχει μια περίληψη του πηγαίου αρχείου που θα δημιουργήσει το Project Navigator βασισμένος στις ρυθμίσεις σας. Πατήστε Finish για να ολοκληρώσετε τη διαδικασία. Σε αυτό το σημείο θα φορτωθεί το πρόγραμμα ECS (Engineering Capture System) το οποίο είναι ένα εργαλείο σχεδίασης κυκλωμάτων. Αυτόματα, το πρόγραμμα ECS θα ανοίξει το νέο σχηματικό διάγραμμα even\_par.sch.



**Εικόνα 10: New Source Dialog (2 από 2)**

Μπορείτε να προσθέσετε στο σχηματικό διάγραμμα διάφορα συστατικά (components), για την ακρίβεια σύμβολα των συστατικών. Για κάθε έργο, διαθέσιμα συστατικά υπάρχουν στην εργαλειοθήκη Symbol Libraries. Τα διαθέσιμα συστατικά της εργαλειοθήκης είναι διατεταγμένα αλφαβητικά σε κάθε βιβλιοθήκη. Από το κυρίως μενού του ECS, επιλέξτε Add→Symbol ή κάντε κλικ στο εικονίδιο Add Symbol στη μπάρα εργαλείων.

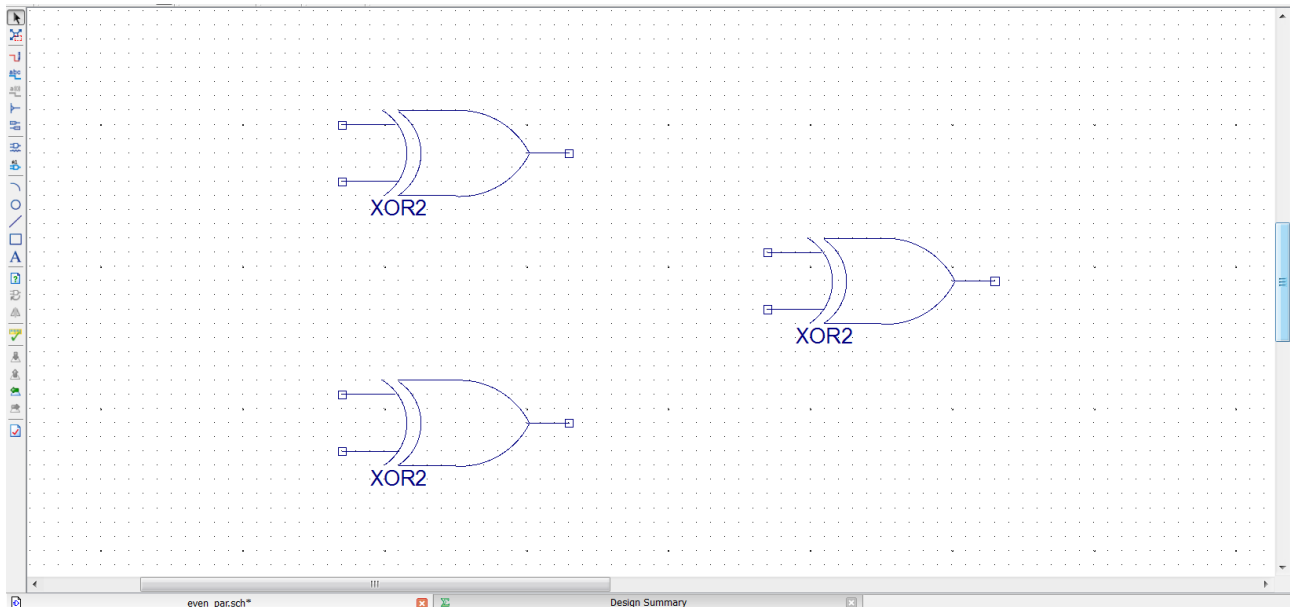
Αυτό ανοίγει το πλαίσιο διαλόγου Symbol Browser στα αριστερά του επεξεργαστή σχηματικού διαγράμματος (schematic editor), όπως φαίνεται στην Εικόνα 11, και το οποίο απεικονίζει όλες τις βιβλιοθήκες και τα συστατικά τους.



**Εικόνα 11: Symbol Browser**

Το πρώτο συστατικό που πρέπει να τοποθετήσετε στο σχηματικό διάγραμμα είναι μια πύλη XOR των 2-εισόδων. Για να επιλέξετε το αντίστοιχο σύμβολο, επιλέξτε την βιβλιοθήκη (category) Logic και έπειτα επιλέξτε το συστατικό xor2 από τη λίστα συμβόλων, όπως φαίνεται στην Εικόνα 11. Μετακινήστε το ποντίκι πίσω στο παράθυρο του διαγράμματος, στη θέση που επιθυμείτε να τοποθετήσετε το σύμβολο και κάντε κλικ για να τοποθετήσετε το αντικείμενο. Θα παρατηρήσετε ότι ο δρομέας έχει αλλάξει για να αναπαραστήσει το σύμβολο της xor2.

Στη συνέχεια, τοποθετήστε ακόμα 2 πύλες xor2 στο διάγραμμα, όπως φαίνεται στην Εικόνα 12.



**Εικόνα 12: Τοποθέτηση συμβόλων στο σχηματικό διάγραμμα**

Εάν έχετε κάνει κάποιο λάθος στην τοποθέτηση των συστατικών, μπορείτε εύκολα να διαγράψετε ή να μετακινήσετε τα συστατικά. Για να μετακινήσετε ένα συστατικό, επιλέξτε το και έπειτα σύρτε το ποντίκι στην επιθυμητή θέση. Για να διαγράψετε ένα συστατικό, επιλέξτε και έπειτα πατήστε το πλήκτρο Delete, ή κάντε δεξί κλικ και επιλέξτε Delete.

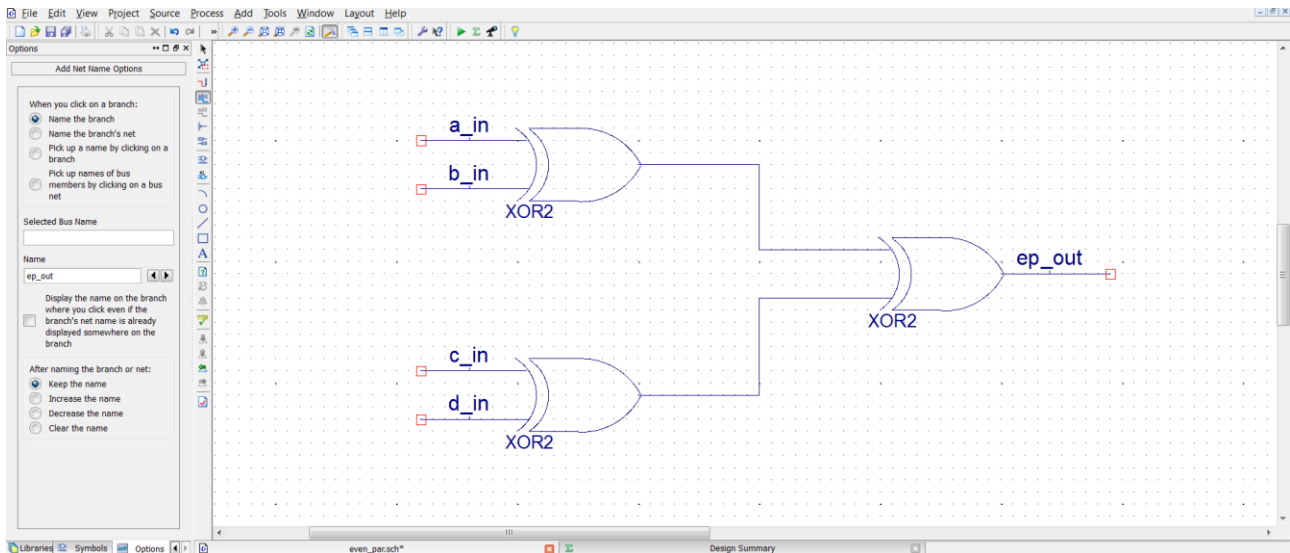
Το επόμενο βήμα είναι να υλοποιήσετε τις συνδέσεις του κυκλώματος. Κάποιες συνδέσεις καλωδίων συνδέουν τα συστατικά του διαγράμματος και κάποιες άλλες επεκτείνονται και αφήνονται ξεκρέμαστες (hanging wires). Για να ενεργοποιήσετε το εργαλείο διασύνδεσης (wiring tool), επιλέξτε από το κυρίως μενού Add→Wire ή κάντε κλικ στο εικονίδιο Add Wire στη μπάρα εργαλείων.

Για να συνδέσετε τους ακροδέκτες δύο πυλών xor2 (την έξοδο μιας πύλης xor με την είσοδο μιας άλλης πύλης xor), κάντε κλικ στο έναν ακροδέκτη (στο τετράγωνο άκρο του), σύρτε το ποντίκι και κάντε κλικ στο δεύτερο ακροδέκτη.

Για να επεκτείνετε το καλώδιο ενός ακροδέκτη (είσοδου ή εξόδου), κάντε κλικ στον ακροδέκτη (στο τετράγωνο άκρο του), σύρτε το ποντίκι και κάντε διπλό κλικ στο σημείο που επιθυμείτε να τερματιστεί το καλώδιο. Όταν τελειώσετε με την καλωδίωση, πατήστε το πλήκτρο Esc για να απενεργοποιήσετε το εργαλείο Add Wire.

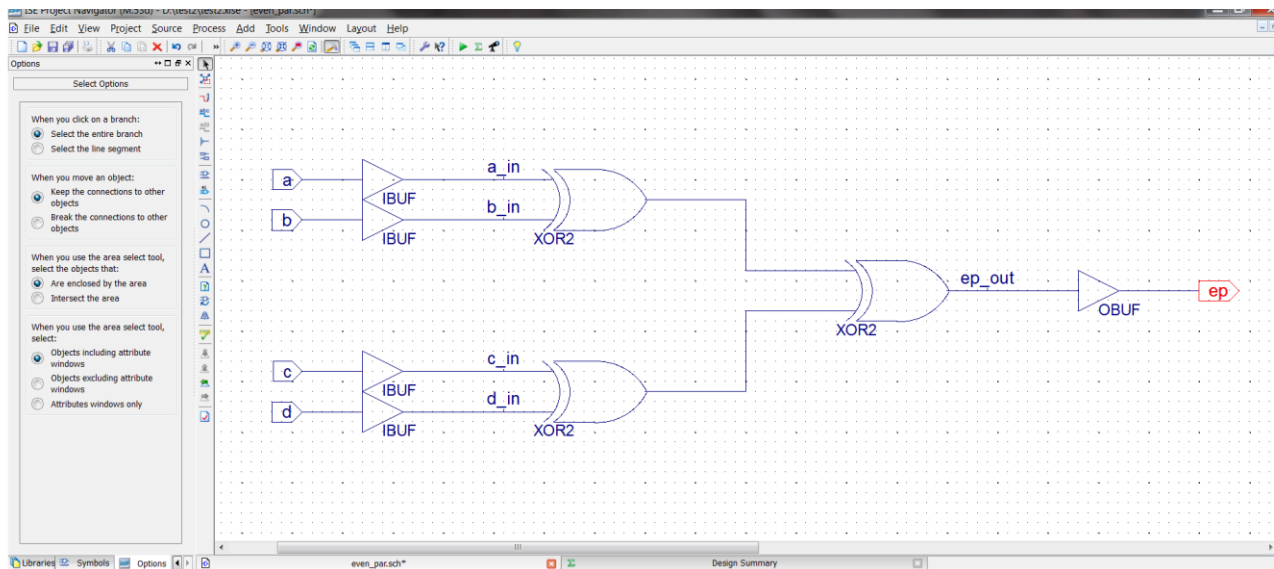
Το επόμενο βήμα είναι να προσθέσετε ονόματα στις συνδέσεις του κυκλώματος. Επιλέξτε από το κυρίως μενού Add→Net Name ή κάντε κλικ στο εικονίδιο Add Net Name στη μπάρα εργαλείων. Έπειτα, δώστε κατάλληλα ονόματα στις συνδέσεις του κυκλώματος. Για να προσθέσετε ένα όνομα σε μια σύνδεση, πληκτρολογήστε το όνομα στο πεδίο Name του παραθύρου Add Net Name Options, τοποθετήστε το δρομέα - που τώρα απεικονίζει το όνομα του καλωδίου - στο καλώδιο και κάντε κλικ. Το διάγραμμα με τα ονόματα των καλωδίων φαίνεται στην Εικόνα 13.





**Εικόνα 13: Καλωδίωση και τοποθέτηση ονομάτων στο σχηματικό διάγραμμα**

Τοποθετήστε input buffers στις εισόδους του κυκλώματος επιλέγοντας την κατηγορία IO και το σύμβολο ibuf. Έπειτα τοποθετήστε output buffer στην έξοδο του κυκλώματος επιλέγοντας την κατηγορία IO και το σύμβολο obuf. Κατόπιν, τοποθετήστε I/O marker στις εισόδους και τις εξόδους του κυκλώματος. Θα παρατηρήσατε ότι το πρόγραμμα ECS ονομάζει τυχαία τα I/O marker. Κάντε δεξί κλικ σε κάθε I/O marker και επιλέξτε Rename Port για να αλλάξετε το όνομα. Το τελικό διάγραμμα του κυκλώματος υπολογισμού άρτιας ισοτιμίας ενός μηνύματος 4-bit φαίνεται στην Εικόνα 14.

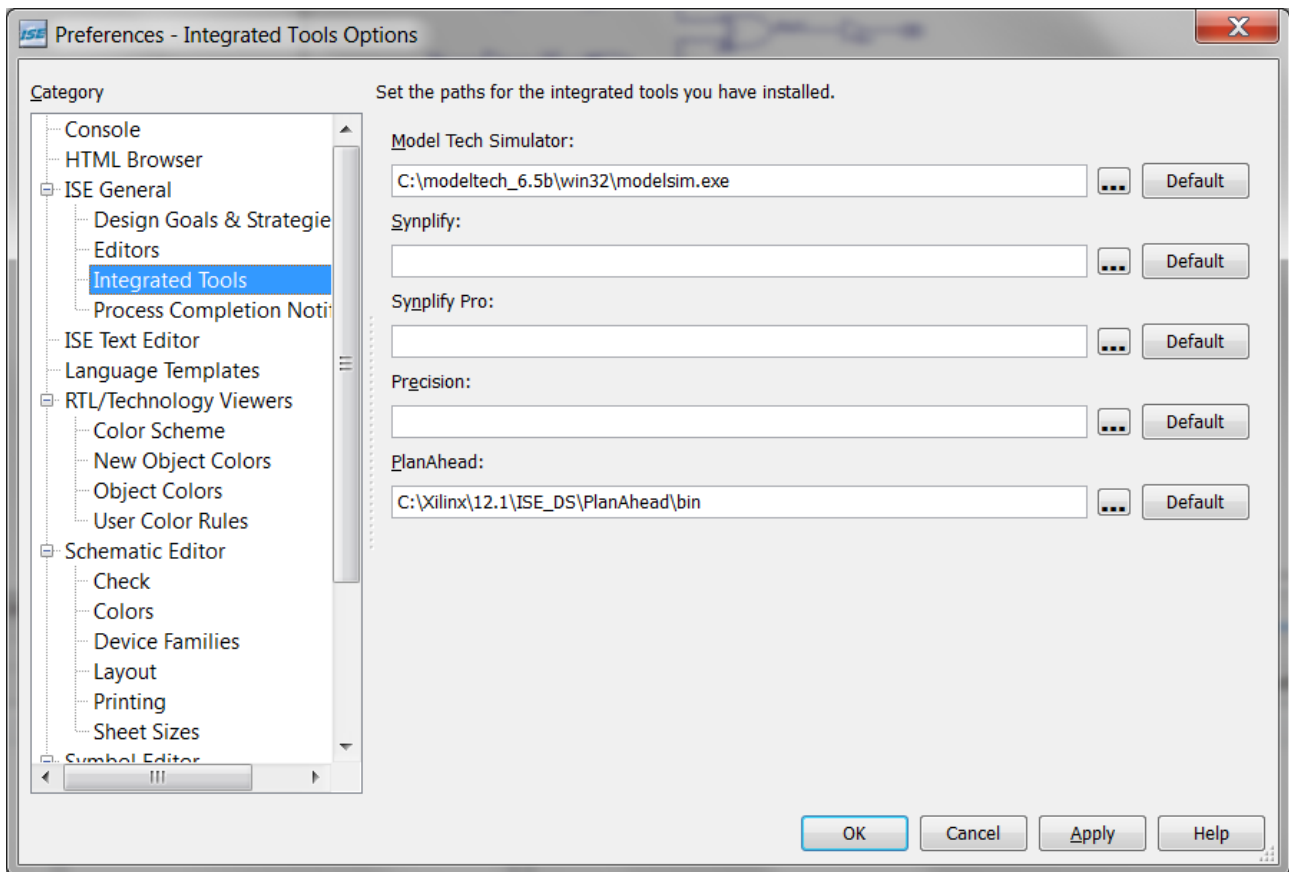


**Εικόνα 14: Τοποθέτηση buffer και I/O marker στο σχηματικό διάγραμμα**

Μόλις τελειώσετε, αποθηκεύστε και κλείστε το πρόγραμμα ECS.

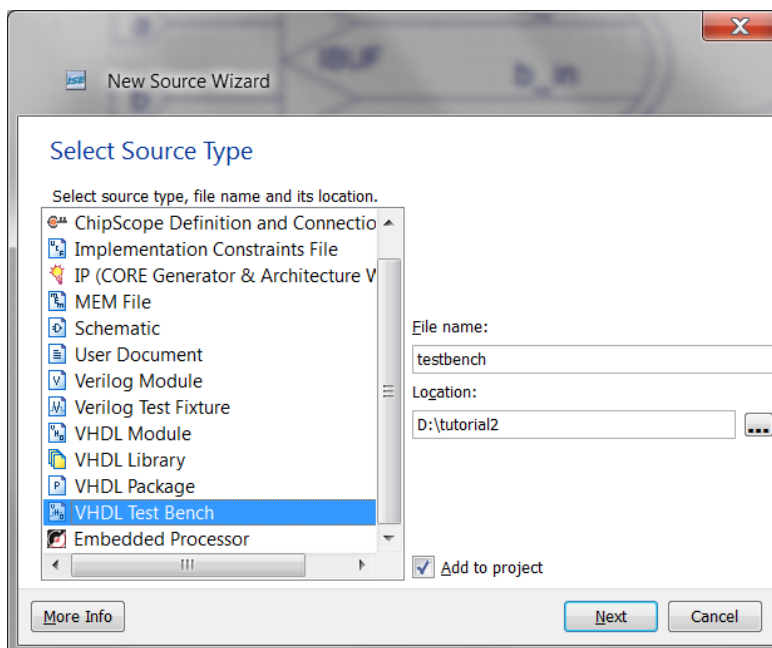
## 4 Λειτουργική προσομοίωση (functional simulation)

Η λειτουργική προσομοίωση εκτελείται προτού να γίνει σύνθεση της σχεδίασης για να επαληθεύσει ότι η λογική του κυκλώματος είναι σωστή. Αυτό επιτρέπει σε ένα σχεδιαστή να βρει και να διορθώσει τυχόν λάθη στη σχεδίαση προτού ξοδέψει χρόνο στα επόμενα βήματα. Το Project Navigator ενσωματώνει τον προσομοιωτή Modelsim και μας επιτρέπει να εκτελέσουμε τις προσομοιώσεις από το Project Navigator. Πριν προχωρήσετε στα παρακάτω βήματα για την εκτέλεση της λειτουργικής προσομοίωσης επιβεβαιώστε ότι έχει θέσει σωστά την διαδρομή του προσομοιωτή Modelsim στις ρυθμίσεις του ISE (Edit → Preferences) σύμφωνα με την Εικόνα 15.



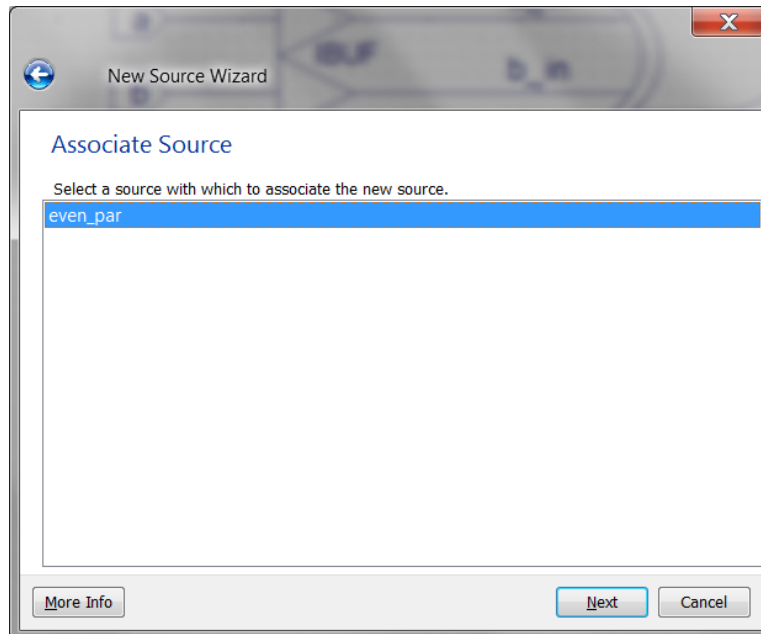
**Εικόνα 15: Ρυθμίσεις του λογισμικού Xilinx ISE για τον προσομοιωτή Modelsim**

Προκειμένου να προσομοιώσουμε τη σχεδίαση, απαιτείται ένα πρόγραμμα δοκιμής (test bench) για να παράγει τα απαραίτητα ερεθίσματα εισόδου (input stimulus) στη σχεδίαση. Δημιουργήστε ένα νέο πηγαίο αρχείο για το testbench. Είτε επιλέξτε Project→New Source από το κυρίως μενού είτε χρησιμοποιήστε την ισοδύναμη διαδικασία στο παράθυρο Processes for Current Source. Το πρώτο από τα κουτιά διαλόγου New Source θα εμφανιστεί, όπως φαίνεται στην Εικόνα 16.



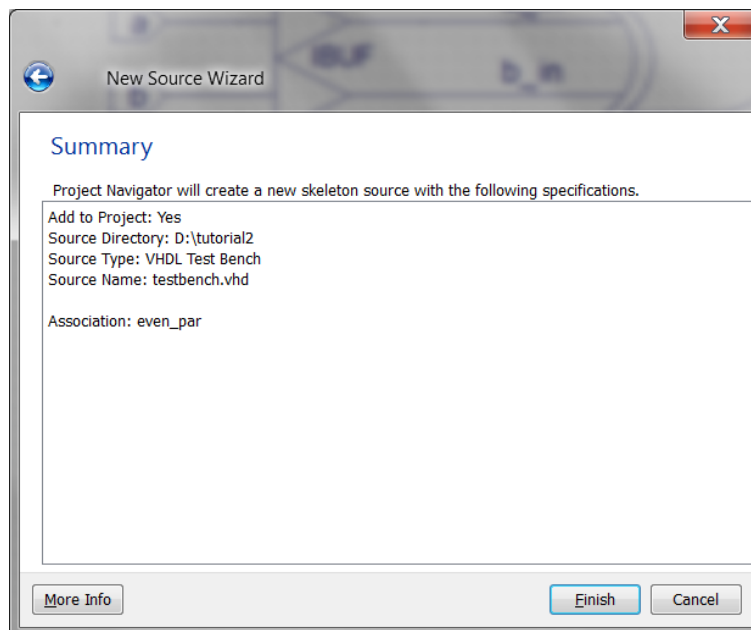
**Εικόνα 16: New Source Dialog (1 από 3)**

Επιλέξτε VHDL Test Bench για να δηλώσετε ότι δημιουργείτε μια μονάδα δοκιμής (testbench) βασισμένη στη γλώσσα VHDL. Κατόπιν, δώστε ένα όνομα αρχείου όπως φαίνεται στην Εικόνα 16. Δεν χρειάζεται να αλλάξετε τη θέση του αρχείου, η οποία θα είναι μέσα στον κατάλογο του έργου που δημιουργήσατε προηγουμένως. Πατήστε Next.



Εικόνα 17: New Source Dialog (2 από 3)

Το δεύτερο πλαίσιο διαλόγου, που φαίνεται στην Εικόνα 17, σας ζητά να προσδιορίσετε μια μονάδα σχεδίασης με την οποία πρέπει να συσχετιστεί το test bench. Επιλέξτε τη μονάδα even\_par όπως φαίνεται στην Εικόνα 17 και πατήστε Next.



Εικόνα 18: New Source Dialog (3 από 3)

Το τελικό πλαίσιο διαλόγου (Εικόνα 18) παρέχει μια περίληψη του πηγαίου αρχείου που θα δημιουργήσει το Project Navigator βασισμένο στις ρυθμίσεις σας. Ελέγξτε την περίληψη για να σιγουρευτείτε ότι ταιριάζει με την Εικόνα 18. Εάν όχι, πατήστε Back για να διορθώσετε οποιοδήποτε λάθος. Διαφορετικά, πατήστε Finish για να ολοκληρώσετε τη διαδικασία. Το νέο πηγαίο αρχείο θα ανοίξει αυτόματα στον επεξεργαστή κειμένου.

Στον επεξεργαστή κειμένου, μερικές από τη βασικές δομές του αρχείου testbench είναι ήδη γραμμένες. Τώρα, εισάγετε την περιγραφή του testbench. Αντιγράψτε τα περιεχόμενα του αρχείου, όπως δίνονται παρακάτω:

```
-- VHDL Test Bench Created from source file even_par.vhd -- 18:15:04 04/03/2007
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY even_par_testbench_vhd_tb IS
END even_par_testbench_vhd_tb;

ARCHITECTURE behavior OF even_par_testbench_vhd_tb IS

    COMPONENT even_par
    PORT (
        a : IN std_logic;
        b : IN std_logic;
        c : IN std_logic;
        d : IN std_logic;
        ep : OUT std_logic
    );
    END COMPONENT;

    SIGNAL a : std_logic;
    SIGNAL b : std_logic;
    SIGNAL c : std_logic;
    SIGNAL d : std_logic;
    SIGNAL ep : std_logic;

BEGIN

    uut: even_par PORT MAP (
        a => a,
        b => b,
        c => c,
        d => d,
        ep => ep
    );

-- *** Test Bench - User Defined Section ***
    tb : PROCESS
    BEGIN
-- All possible input combinations
```

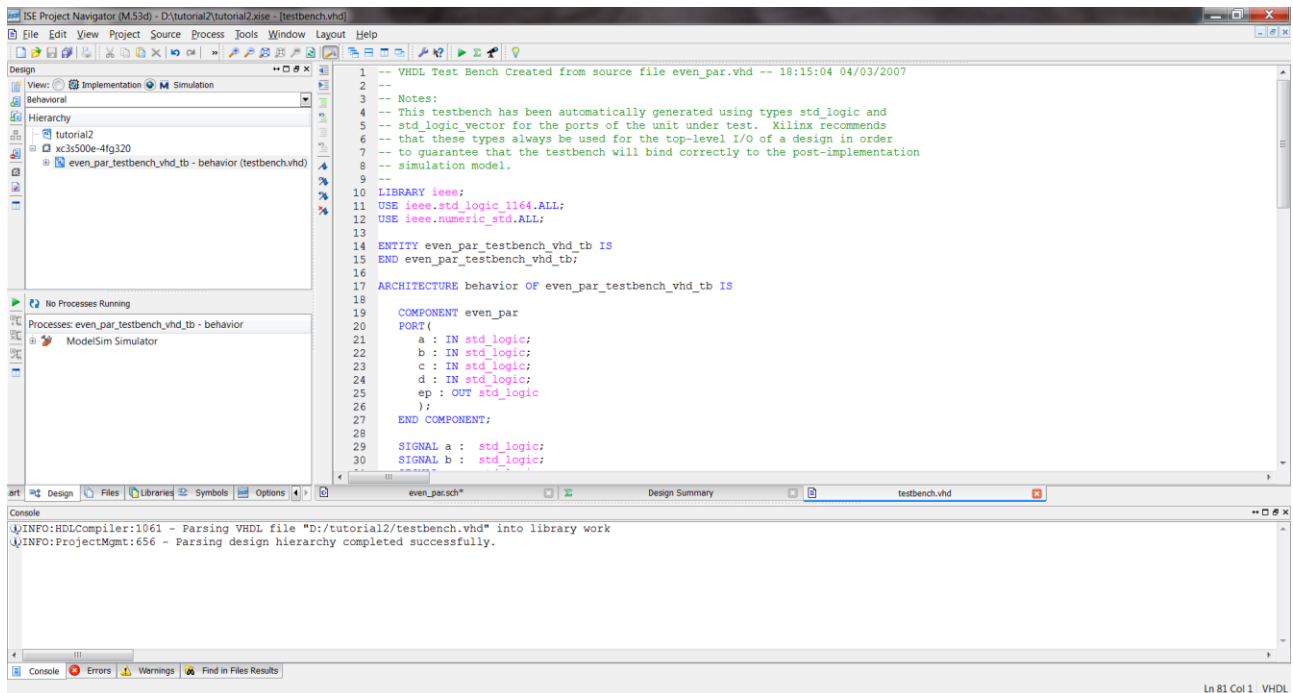
```

a <= '0'; b <= '0'; c <= '0'; d <= '0';
wait for 20 ns;
a <= '0'; b <= '0'; c <= '0'; d <= '1';
wait for 20 ns;
a <= '0'; b <= '0'; c <= '1'; d <= '0';
wait for 20 ns;
a <= '0'; b <= '0'; c <= '1'; d <= '1';
wait for 20 ns;
a <= '0'; b <= '1'; c <= '0'; d <= '0';
wait for 20 ns;
a <= '0'; b <= '1'; c <= '0'; d <= '1';
wait for 20 ns;
a <= '0'; b <= '1'; c <= '1'; d <= '0';
wait for 20 ns;
a <= '0'; b <= '1'; c <= '1'; d <= '1';
wait for 20 ns;
a <= '1'; b <= '0'; c <= '0'; d <= '0';
wait for 20 ns;
a <= '1'; b <= '0'; c <= '0'; d <= '1';
wait for 20 ns;
a <= '1'; b <= '0'; c <= '1'; d <= '0';
wait for 20 ns;
a <= '1'; b <= '0'; c <= '1'; d <= '1';
wait for 20 ns;
a <= '1'; b <= '1'; c <= '0'; d <= '0';
wait for 20 ns;
a <= '1'; b <= '1'; c <= '0'; d <= '1';
wait for 20 ns;
a <= '1'; b <= '1'; c <= '1'; d <= '0';
wait for 20 ns;
a <= '1'; b <= '1'; c <= '1'; d <= '1';

wait; -- will wait forever
END PROCESS;
-- *** End Test Bench - User Defined Section ***

END;
```

Σε αυτό το σημείο, πρέπει να καταλήξετε με ένα παράθυρο που μοιάζει με αυτό που φαίνεται στην Εικόνα 19. Μόλις τελειώσετε, αποθηκεύστε το αρχείο και κλείστε το παράθυρο.

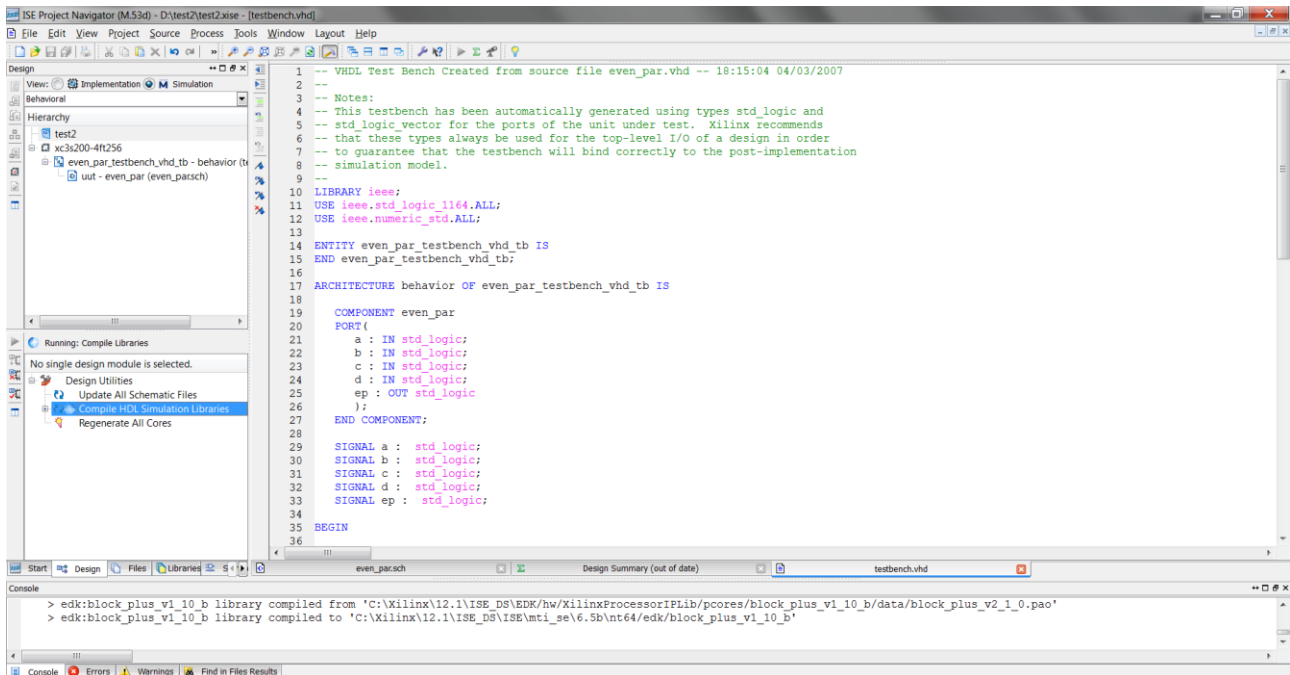


**Εικόνα 19: Ολοκληρωμένο Test Bench**

Τώρα που έχετε ένα testbench στο έργο σας, μπορείτε να εκτελέσετε λειτουργική προσομοίωση στη σχεδίασή σας. Οι διεργασίες προσομοίωσης σας επιτρέπουν να τρέξετε προσομοίωση στη σχεδίαση χρησιμοποιώντας τον προσομοιωτή Modelsim. Για να εντοπίσετε τις διεργασίες του προσομοιωτή Modelsim, επιλέξτε το Simulation στο παράθυρο Design. Κατόπιν, επιλέξτε το είδος της προσομοίωσης επεκτείνοντας το μενού ακριβώς κάτω από την επιλογή Simulation. Οι παρακάτω διεργασίες προσομοίωσης είναι διαθέσιμες:

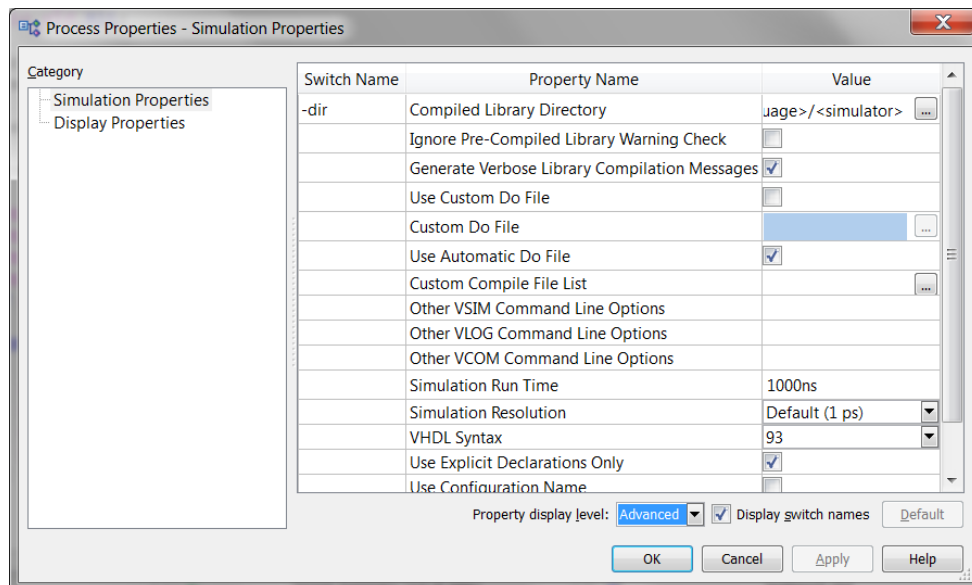
- Simulate Behavioral Model. Προσομοιώνει το μοντέλο συμπεριφοράς του κυκλώματος; έτσι αρχίζει η προσομοίωση της σχεδίασης.
- Simulate Post-Translate VHDL Model. Προσομοιώνει το μοντέλο του κυκλώματος (netlist) που προκύπτει μετά από το στάδιο της υλοποίησης NGDBuild.
- Simulate Post-Map VHDL Model. Προσομοιώνει το μοντέλο του κυκλώματος (netlist) που προκύπτει μετά από το στάδιο της αντιστοίχισης (mapping).
- Simulate Post-Place & Route VHDL Model. Προσομοιώνει το μοντέλο του κυκλώματος (netlist) που προκύπτει μετά από το στάδιο της τοποθέτησης (placement) και της δρομολόγησης (routing).

**Προσοχή:** Την πρώτη φορά που θα εκτελέσετε προσομοίωση πρέπει να εκτελέσετε την εντολή Compile HDL Simulation Libraries. Για να το κάνετε αυτό πρέπει να επιλέξετε το top σημείο στο Simulation (test στην Εικόνα 20). Κατόπιν κάντε click στο + δίπλα από το Design Utilities και εκτελέστε την εντολή Compile HDL Simulation Libraries όπως φαίνεται στην Εικόνα 20.



**Εικόνα 20: Compile HDL Simulation Libraries**

Σε αυτό το σημείο, θα εκτελέσετε μια λειτουργική προσομοίωση χρησιμοποιώντας το Simulate Behavioral Model αλλά πρέπει πρώτα να καθορίσετε τις ιδιότητες της διεργασίας προσομοίωσης. Κάντε δεξί κλικ στο Simulate Behavioral Model, και επιλέξτε Process Properties. Το πλαίσιο διαλόγου Process Properties εμφανίζεται, όπως φαίνεται στην Εικόνα 21.

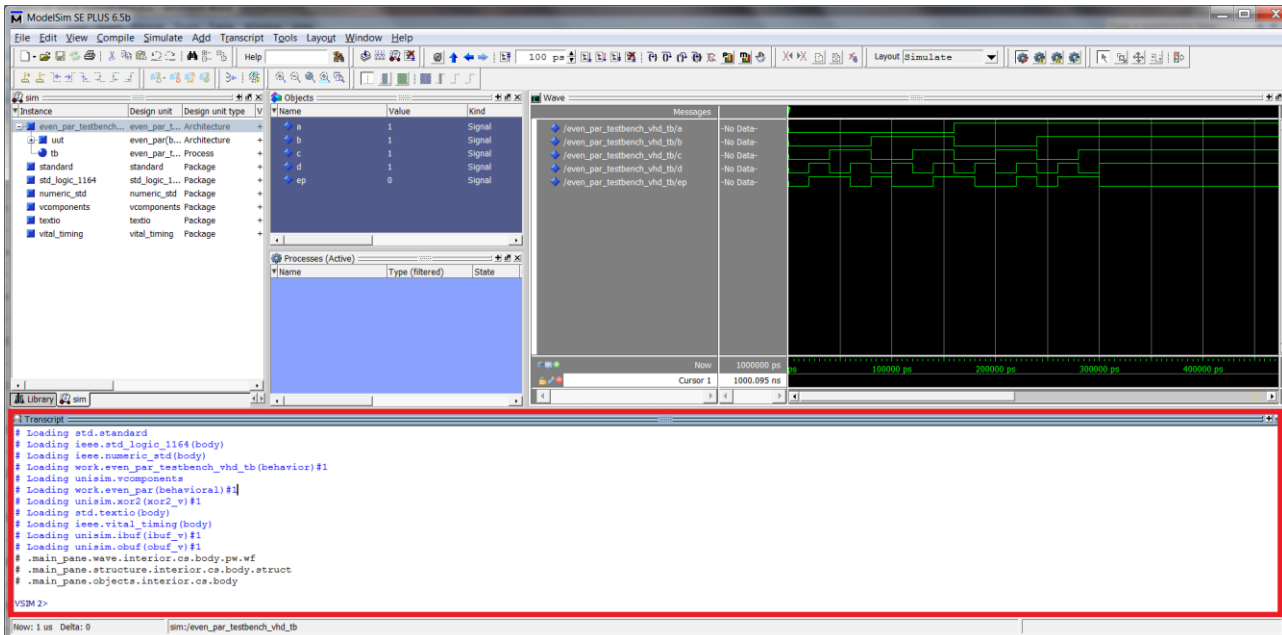


**Εικόνα 21: Simulation Process Properties**

Εάν δεν εμφανιστούν όλες οι ιδιότητες που φαίνονται στην Εικόνα 21, μπορείτε να τις καταστήσετε ορατές ως εξής: Στο μενού Property display level, αλλάξτε την τιμή από Standard σε Advanced. Σηγουρευτείτε ότι οι ιδιότητες έχουν τις τιμές που φαίνονται στην Εικόνα 21. Μια από τις παραμέτρους που μπορεί να σας ενδιαφέρει σε αυτό το στάδιο είναι ο χρόνος εκτέλεσης προσομοίωσης (1000 ns); είναι περισσότερο από ότι χρειάζεστε για να τρέξει το testbench που δημιουργήσατε. Πατήστε OK.

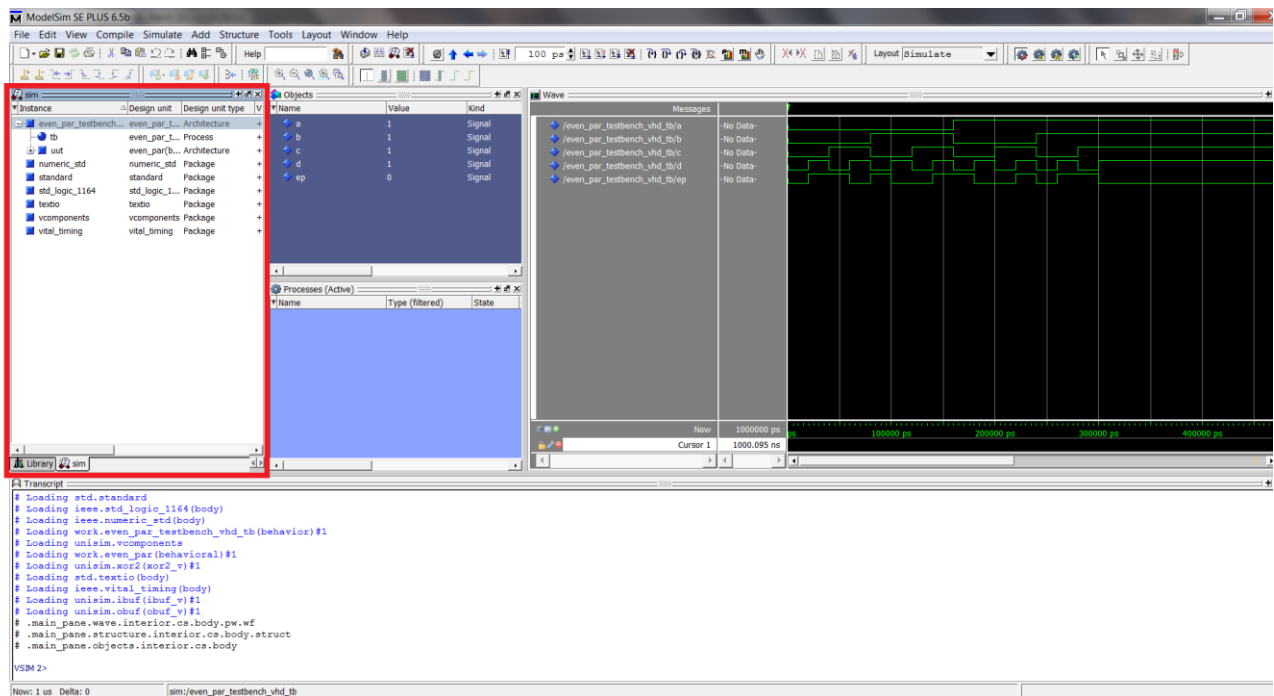
Για να αρχίσετε την προσομοίωση, κάντε διπλό-κλικ στο Simulate Behavioral Model. Το Modelsim δημιουργεί έναν κατάλογο εργασίας, μεταγλωττίζει τα πηγαία αρχεία, φορτώνει τη σχεδίαση, και εκτελεί την προσομοίωση για τον καθορισμένο χρόνο. Τέσσερα παράθυρα του Modelsim θα εμφανιστούν. Το πρώτο, και πιο σημαντικό, είναι η κύρια κονσόλα (transcript) του Modelsim, που φαίνεται στην Εικόνα 22.

Αυτό το παράθυρο εμφανίζει τα μηνύματα από τον προσομοιωτή. Αυτά τα μηνύματα περιλαμβάνουν σημειώσεις, προειδοποιήσεις, και λάθη, συν οποιοδήποτε μήνυμα εξόδου παραχθεί από τη σχεδίαση που προσομοιώνετε.



Εικόνα 22: ModelSim Transcript Window

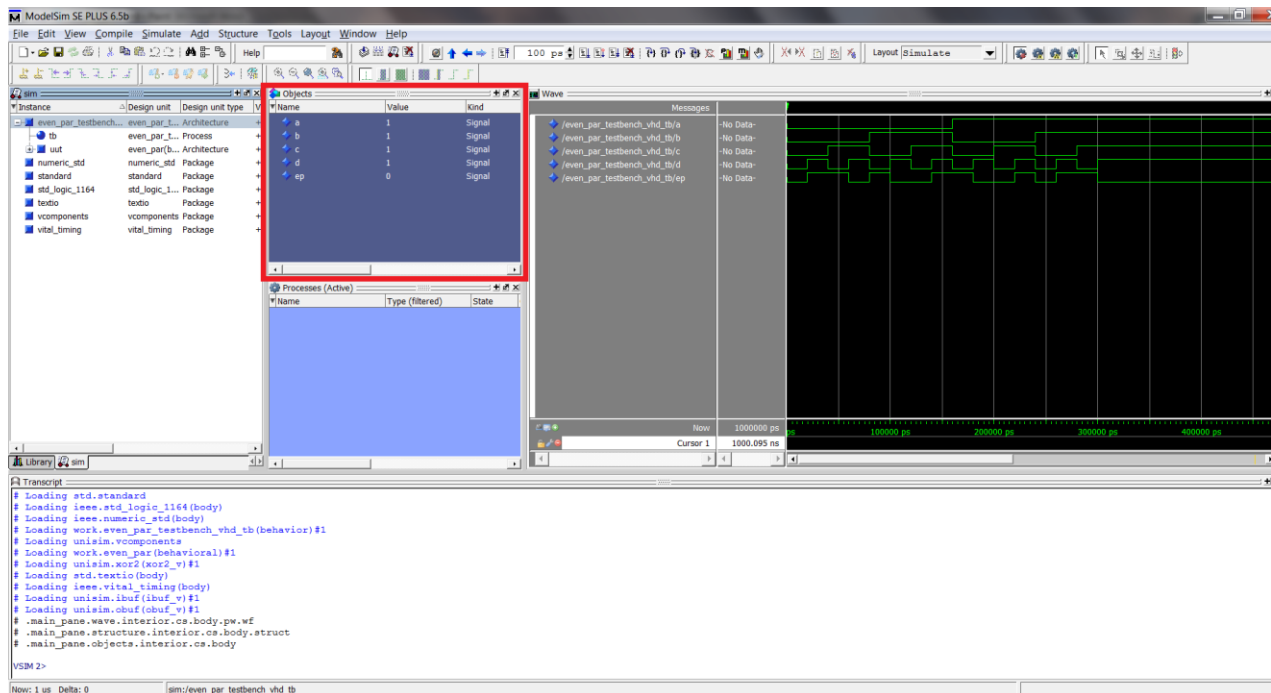
Το δεύτερο παράθυρο είναι το παράθυρο sim, που φαίνεται στην Εικόνα 23. Αυτό το παράθυρο σας επιτρέπει να αναζητήσετε την ιεραρχία του testbench και της σχεδίασης υπό δοκιμή. Στις μεγάλες ιεραρχικές σχεδιάσεις, αυτό είναι πολύ χρήσιμο.



Εικόνα 23: Sim Window

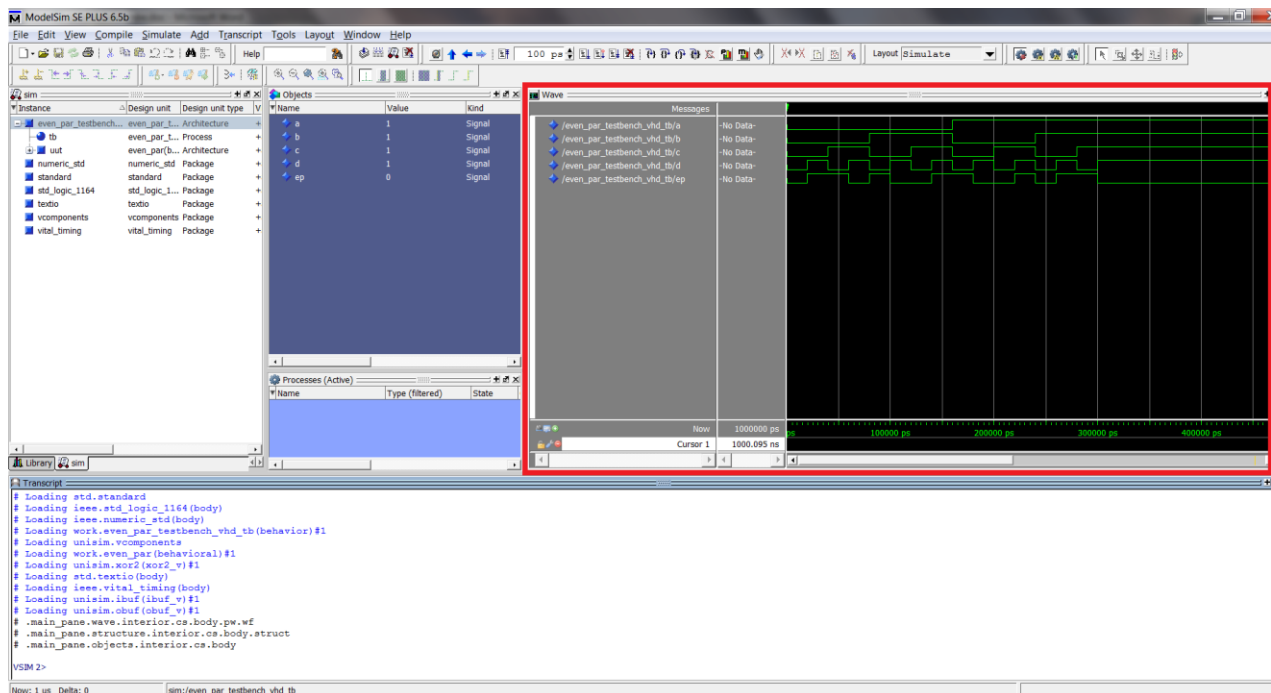
Το τρίτο παράθυρο είναι το παράθυρο objects, που φαίνεται στην Εικόνα 24. Αυτό το παράθυρο δείχνει τα σήματα του μέρους της σχεδίασης που έχετε επιλέξει στο παράθυρο structure.





Εικόνα 24: Objects Window

Το τέταρτο και τελευταίο παράθυρο είναι το παράθυρο wave, το οποίο χρησιμοποιείται για να απεικονίσει τις κυματομορφές που προσομοιώνονται. Το Project Navigator προσθέτει αυτόματα όλα τα σήματα του υψηλού επιπέδου σχεδίασης στο παράθυρο wave, όπως φαίνεται στην Εικόνα 25. Πρόσθετα σήματα απεικονίζονται στο παράθυρο signals ανάλογα με την δομή που θα επιλέξετε στο παράθυρο structure.



Εικόνα 25: Wave Window

Υπάρχουν δύο βασικές μέθοδοι για τα προσθέσετε σήματα στο παράθυρο wave. Μπορείτε να τα μεταφέρετε από το παράθυρο signals (με drag and drop), ή να τα επιλέξετε στο παράθυρο signals και έπειτα να επιλέξετε Add→to Wave→Selected Signals. Εάν χρησιμοποιήσετε αυτήν τη δεύτερη τεχνική, θα δείτε ότι υπάρχουν επιπλέον διαθέσιμες επιλογές.

Όταν προσθέτετε τα νέα σήματα στο παράθυρο wave, θα παρατηρήσετε ότι οι κυματομορφές δεν εμφανίζονται αυτόματα. Αυτό είναι επειδή το Modelsim δεν κατέγραψε τα δεδομένα προσομοίωσης για αυτά τα σήματα. Εξ ορισμού, το Modelsim καταγράφει δεδομένα μόνο για τα σήματα που έχουν προστεθεί στο παράθυρο wave πριν ή κατά τη διάρκεια της προσομοίωσης. Επομένως, όταν προστίθενται νέα σήματα στο παράθυρο wave, η προσομοίωση πρέπει να ξαναξεκινήσει και να επαναληφθεί για το επιθυμητό χρονικό διάστημα. Για να ξαναξεκινήσει η προσομοίωση, κάντε κλικ στο Restart Simulation. Τότε εμφανίζεται το πλαίσιο διαλόγου Restart; απλά πατήστε Restart.

Ο προσομοιωτής Modelsim παρέχει την ικανότητα της αποθήκευσης του καταλόγου σημάτων στο παράθυρο wave. Αυτό μπορεί να είναι σημαντικό όταν προστίθενται νέα σήματα ή ερεθίσματα, και η προσομοίωση ξαναξεκινά. Στο παράθυρο wave, επιλέξτε αρχείο File→Save Format. Αφού ξαναξεκινήσετε μια προσομοίωση, μπορείτε να επιλέξετε File→Load Format στο παράθυρο wave για να επαναφέρετε τη λίστα σημάτων.

## 5 Σύνθεση της σχεδίασης (design synthesis)

Με μια λειτουργικά σωστή περιγραφή της σχεδίασης σε VHDL, το επόμενο βήμα είναι να χρησιμοποιήσετε ένα εργαλείο σύνθεσης (synthesis tool) για να μετασχηματίσετε την περιγραφή σας σε μια λίστα συνδέσεων (netlist). Μια λίστα συνδέσεων (netlist) είναι μια σχηματική αναπαράσταση που μπορεί να διαβαστεί από αυτόματα εργαλεία. Σε αυτήν την κατηγορία, θα χρησιμοποιήσουμε ένα εργαλείο που ονομάζεται XST, το οποίο είναι ενσωματωμένο στον Project Navigator και μπορεί να στοχεύσει μόνο FPGA συσκευές Xilinx.

Επιλέξτε το even\_par στο tab Implementation του παραθύρου Design. Κατόπιν, κάντε διπλό-κλικ στη διαδικασία Synthesize - XST στο παράθυρο Processes: even\_par. Το Project Navigator θα συνθέσει τη σχεδίαση και θα εμφανίσει πληροφορίες στο παράθυρο Console. Σαν ενημερωτική σημείωση, είναι δυνατό να αλλάξετε τις επιλογές σύνθεσης προτού να συνθέσετε τη σχεδίαση κάνοντας δεξί κλικ στο Synthesize—XST και έπειτα επιλέγοντας Process Properties. Για αυτόν τον οδηγό εκμάθησης, αφήστε τις προεπιλεγμένες επιλογές (default settings).

Δεν πρέπει να δείτε κανένα λάθος στο παράθυρο Console. Εντούτοις, πρέπει πάντα να διαβάζετε το αρχείο log, το οποίο είναι διαθέσιμο εάν κάνετε δεξί click στο Synthesize – XST και επιλέξετε την εντολή View Text Report. Εάν δεν καταλαβαίνετε ένα συγκεκριμένο μήνυμα, μπορείτε να ψάξετε στην ιστοσελίδα υποστήριξης της Xilinx. Για λόγους σύγκρισης, ακολουθεί ένα δείγμα αρχείου log.

```
Release 12.1 - xst M.53d (nt64)
Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.
--> Parameter TMPDIR set to xst/projnav.tmp
Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.10 secs

--> Parameter xsthdpdir set to xst
Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.10 secs

--> Reading design: even_par.prj
TABLE OF CONTENTS
  1) Synthesis Options Summary
  2) HDL Compilation
  3) Design Hierarchy Analysis
  4) HDL Analysis
  5) HDL Synthesis
    5.1) HDL Synthesis Report
  6) Advanced HDL Synthesis
    6.1) Advanced HDL Synthesis Report
  7) Low Level Synthesis
  8) Partition Report
  9) Final Report
    9.1) Device utilization summary
    9.2) Partition Resource Summary
```

## 9.3) TIMING REPORT

```

=====
*                               Synthesis Options Summary                               *
=====
---- Source Parameters
Input File Name                  : "even_par.prj"
Input Format                      : mixed
Ignore Synthesis Constraint File : NO
---- Target Parameters
Output File Name                 : "even_par"
Output Format                     : NGC
Target Device                    : xc3s500e-4-fg320
---- Source Options
Top Module Name                  : even_par
Automatic FSM Extraction         : YES
FSM Encoding Algorithm           : Auto
Safe Implementation              : No
FSM Style                        : lut
RAM Extraction                   : Yes
RAM Style                        : Auto
ROM Extraction                   : Yes
Mux Style                        : Auto
Decoder Extraction               : YES
Priority Encoder Extraction       : YES
Shift Register Extraction        : YES
Logical Shifter Extraction       : YES
XOR Collapsing                  : YES
ROM Style                        : Auto
Mux Extraction                   : YES
Resource Sharing                 : YES
Asynchronous To Synchronous     : NO
Multiplier Style                : auto
Automatic Register Balancing     : No
---- Target Options
Add IO Buffers                   : YES
Global Maximum Fanout           : 500
Add Generic Clock Buffer (BUFG)  : 24
Register Duplication             : YES
Slice Packing                    : YES
Optimize Instantiated Primitives : NO
Use Clock Enable                 : Yes
Use Synchronous Set              : Yes
Use Synchronous Reset            : Yes
Pack IO Registers into IOBs      : auto
Equivalent register Removal      : YES
---- General Options
Optimization Goal                : Speed
Optimization Effort              : 1
Library Search Order             : even_par.lso
Keep Hierarchy                   : NO
Netlist Hierarchy                : as_optimized
RTL Output                       : Yes
Global Optimization              : AllClockNets
Read Cores                       : YES
Write Timing Constraints          : NO
Cross Clock Analysis             : NO
Hierarchy Separator              : /
Bus Delimiter                    : <>
Case Specifier                   : maintain
Slice Utilization Ratio          : 100
BRAM Utilization Ratio           : 100

```

```
Verilog 2001                : YES
Auto BRAM Packing           : NO
Slice Utilization Ratio Delta : 5
=====
*                           HDL Compilation                            *
=====
Compiling vhdl file "D:/tutorial2/even_par.vhf" in Library work.
Entity <even_par> compiled.
Entity <even_par> (Architecture <BEHAVIORAL>) compiled.
=====
*                           Design Hierarchy Analysis                *
=====
Analyzing hierarchy for entity <even_par> in library <work> (architecture
<BEHAVIORAL>).
=====
*                           HDL Analysis                            *
=====
Analyzing Entity <even_par> in library <work> (Architecture <BEHAVIORAL>).
  Set user-defined property "IBUF_DELAY_VALUE = 0" for instance <XLXI_7> in
unit <even_par>.
  Set user-defined property "IFD_DELAY_VALUE = AUTO" for instance <XLXI_7> in
unit <even_par>.
  Set user-defined property "IOSTANDARD = DEFAULT" for instance <XLXI_7> in
unit <even_par>.
  Set user-defined property "IBUF_DELAY_VALUE = 0" for instance <XLXI_9> in
unit <even_par>.
  Set user-defined property "IFD_DELAY_VALUE = AUTO" for instance <XLXI_9> in
unit <even_par>.
  Set user-defined property "IOSTANDARD = DEFAULT" for instance <XLXI_9> in
unit <even_par>.
  Set user-defined property "IBUF_DELAY_VALUE = 0" for instance <XLXI_10> in
unit <even_par>.
  Set user-defined property "IFD_DELAY_VALUE = AUTO" for instance <XLXI_10>
in unit <even_par>.
  Set user-defined property "IOSTANDARD = DEFAULT" for instance <XLXI_10> in
unit <even_par>.
  Set user-defined property "DRIVE = 12" for instance <XLXI_11> in unit
<even_par>.
  Set user-defined property "IOSTANDARD = DEFAULT" for instance <XLXI_11> in
unit <even_par>.
  Set user-defined property "SLEW = SLOW" for instance <XLXI_11> in unit
<even_par>.
  Set user-defined property "IBUF_DELAY_VALUE = 0" for instance <XLXI_12> in
unit <even_par>.
  Set user-defined property "IFD_DELAY_VALUE = AUTO" for instance <XLXI_12>
in unit <even_par>.
  Set user-defined property "IOSTANDARD = DEFAULT" for instance <XLXI_12> in
unit <even_par>.
Entity <even_par> analyzed. Unit <even_par> generated.
=====
*                           HDL Synthesis                            *
=====
Performing bidirectional port resolution...
Synthesizing Unit <even_par>.
  Related source file is "D:/tutorial2/even_par.vhf".
Unit <even_par> synthesized.
=====
HDL Synthesis Report
Found no macro
=====
```

```

=====
*                               Advanced HDL Synthesis                               *
=====
Advanced HDL Synthesis Report
Found no macro
=====
*                               Low Level Synthesis                               *
=====
Optimizing unit <even_par> ...
Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block even_par, actual ratio is 0.
Final Macro Processing ...
=====
Final Register Report
Found no macro
=====
*                               Partition Report                               *
=====
Partition Implementation Status
-----
  No Partitions were found in this design.
-----
=====
*                               Final Report                               *
=====
Final Results
RTL Top Level Output File Name      : even_par.ngr
Top Level Output File Name          : even_par
Output Format                        : NGC
Optimization Goal                   : Speed
Keep Hierarchy                      : NO
Design Statistics
# IOs                                : 5
Cell Usage :
# BELS                               : 3
# XOR2                               : 3
# IO Buffers                         : 5
# IBUF                               : 4
# OBUF                               : 1
=====
Device utilization summary:
-----
Selected Device : 3s500efg320-4
  Number of Slices:                0 out of 4656    0%
  Number of IOs:                   5
  Number of bonded IOBs:           5 out of 232    2%
-----
Partition Resource Summary:
-----
  No Partitions were found in this design.
-----
=====
TIMING REPORT
NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.
Clock Information:

```

```

-----
No clock signals found in this design
Asynchronous Control Signals Information:
-----
No asynchronous control signals found in this design
Timing Summary:
-----
Speed Grade: -4
  Minimum period: No path found
  Minimum input arrival time before clock: No path found
  Maximum output required time after clock: No path found
  Maximum combinational path delay: 7.158ns
Timing Detail:
-----
All values displayed in nanoseconds (ns)
=====
Timing constraint: Default path analysis
  Total number of paths / destination ports: 4 / 1
-----
Delay:                7.158ns (Levels of Logic = 4)
  Source:              d (PAD)
  Destination:        ep (PAD)
  Data Path: d to ep

      Cell:in->out      fanout      Gate      Net
                        Delay          Delay      Logical Name (Net Name)
-----
      IBUF:I->O          1      1.218     0.420     XLXI_10 (d_in)
      XOR2:I0->O         1      0.704     0.420     XLXI_2 (XLXN_2)
      XOR2:I0->O         1      0.704     0.420     XLXI_3 (ep_out)
      OBUF:I->O          3.272
                        XLXI_11 (ep)
-----
      Total              7.158ns (5.898ns logic, 1.260ns route)
                        (82.4% logic, 17.6% route)
=====
Total REAL time to Xst completion: 5.00 secs
Total CPU time to Xst completion: 5.19 secs

-->
Total memory usage is 270368 kilobytes
Number of errors      :    0 (    0 filtered)
Number of warnings    :    0 (    0 filtered)
Number of infos       :    0 (    0 filtered)

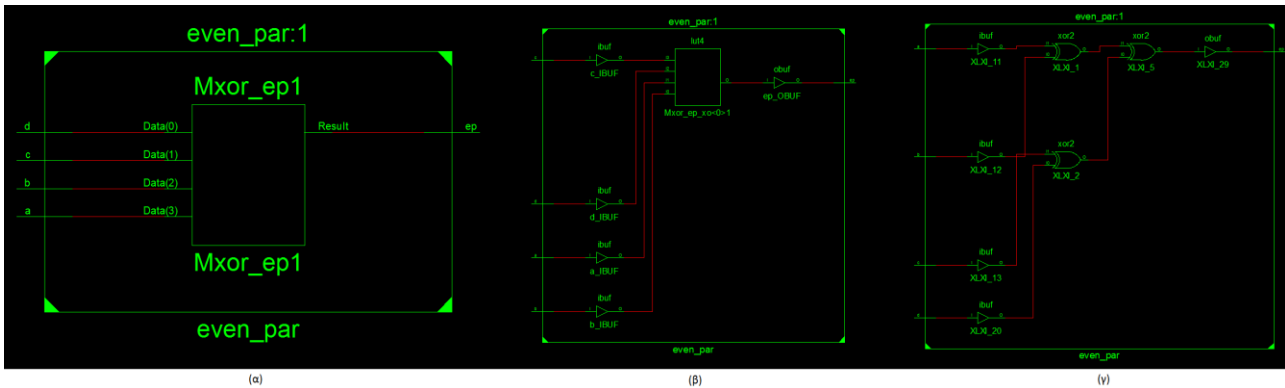
```

Διαβάζοντας την αναφορά είναι ένας καλός τρόπος να ανακαλύψετε ποιους τύπους (και πόσους) πόρους της συσκευής χρησιμοποίησε το εργαλείο σύνθεσης. Μπορείτε επίσης να ανακαλύψετε άλλα προβλήματα με αυτόν τον τρόπο. Για παράδειγμα, εάν διαπιστώσετε ότι η συγκεκριμένη περιγραφή της σχεδίασης οδήγησε στη χρήση ενός flip-flop, εκτός από τους πίνακες αναζήτησης (LUTs) και τους I/O buffers, παρόλο που η συνάρτηση του κυκλώματος είναι συνδυαστική, πρέπει να επιστρέψετε στη σχεδίαση και να αναζητήσετε τι πήγε στραβά. Για αυτό το λόγο, πρέπει να έχετε μια εικόνα του υλικού που προσπαθείτε να δημιουργήσετε όταν γράφετε την περιγραφή του κυκλώματός σας. Σε αυτό το σημείο, πρέπει να έχετε ένα πράσινο σημάδι (✓) δίπλα στη διαδικασία Synthesize—XST.

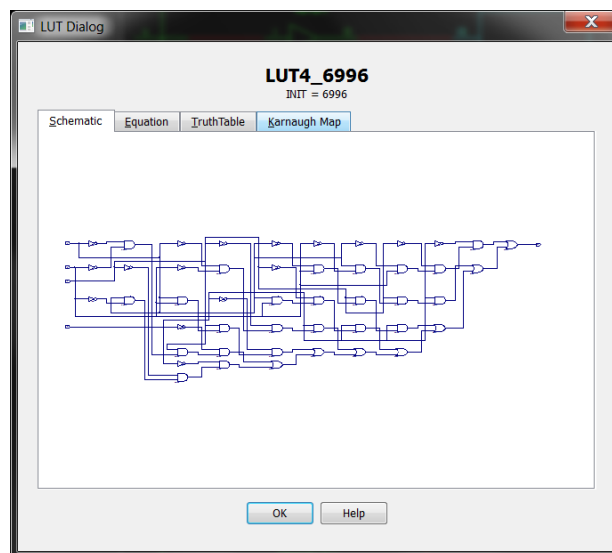
**Προσοχή: Το επόμενο βήμα πρέπει να πραγματοποιηθεί στο πρώτο παράδειγμα του tutorial και αφού έχετε κάνει σύνθεση το top design.**

Επεκτείνεται το μενού Synthesize – XST και εκτελέστε την εντολή View RTL Schematic. Στο παράθυρο που θα σας εμφανιστεί επιλέξτε Start with a schematic of top level block. Το ISE θα σας εμφανίσει το RTL σχηματικό για την σχεδίαση σας. Κατόπιν επαναλάβετε τα ίδια βήματα για να εμφανίσετε το Technology Schematic (View Technology Schematic). Τα αποτελέσματα των δύο εντολών παρουσιάζεται στην Εικόνα 26 όπου στην πρώτη περίπτωση (α) παρουσιάζεται το RTL Schematic με τις εισόδους και τις εξόδους του

κυκλώματος, ενώ στην δεύτερη περίπτωση (β) παρουσιάζεται το Technology Schematic με την αντιστοίχιση της συνάρτησης σε ένα Look Up Table (LUT) του FPGA. Κάνοντας διπλό click πάνω στο Lut4 ανοίγει ένα νέο πλαίσιο διαλόγου (Εικόνα 27) από το οποίο μπορούμε να δούμε το σχηματικό (τις πύλες του κυκλώματος), την συνάρτηση, τον πίνακα αληθείας και το χάρτη Karnaugh για το συγκεκριμένο LUT. Τέλος στην Εικόνα 26 (γ) παρουσιάζεται το αποτέλεσμα και των δύο μορφών σχηματικού, εάν το top design της σχεδίασης μας είναι σχηματικής αναπαράστασης, όπως ακριβώς συμβαίνει και με το παράδειγμα του tutorial 2.



Εικόνα 26: (α) RTL Schematic View, (β) Technology Schematic View, (γ) Αποτέλεσμα και των δύο μορφών εάν το top design είναι σχηματικής μορφής.



Εικόνα 27: Look Up Table Dialog Box

## 6 Υλοποίηση της σχεδίασης (design implementation)

Η υλοποίηση της σχεδίασης είναι η ακολουθία γεγονότων που μεταφράζει τη λίστα συνδέσεων της σχεδίασης που έχετε ήδη συνθέσει (synthesized design netlist) σε ένα αρχείο προγραμματισμού για τη συσκευή FPGA. Η περιγραφή του κυκλώματός σας, που έχετε συνθέσει τώρα, έχει έναν αριθμό θυρών (ports) στο υψηλότερο επίπεδο (top level). Τα εργαλεία υλοποίησης (implementation tools) πρέπει να γνωρίζουν πώς θα αναθέσουν τις θύρες στο υψηλότερο επίπεδο της σχεδίασής σας στους φυσικούς ακροδέκτες (pins) του FPGA, οι οποίοι συνδέονται με διάφορους πόρους της πλακέτας Spartan-3E Starter Kit. Εάν δεν ορίσετε ρητές αναθέσεις, τα εργαλεία θα ορίσουν τυχαία τους ακροδέκτες για σας. Προφανώς, αυτό είναι μια κακή ιδέα αφού οι τυχαίες αναθέσεις θα είναι λανθασμένες.

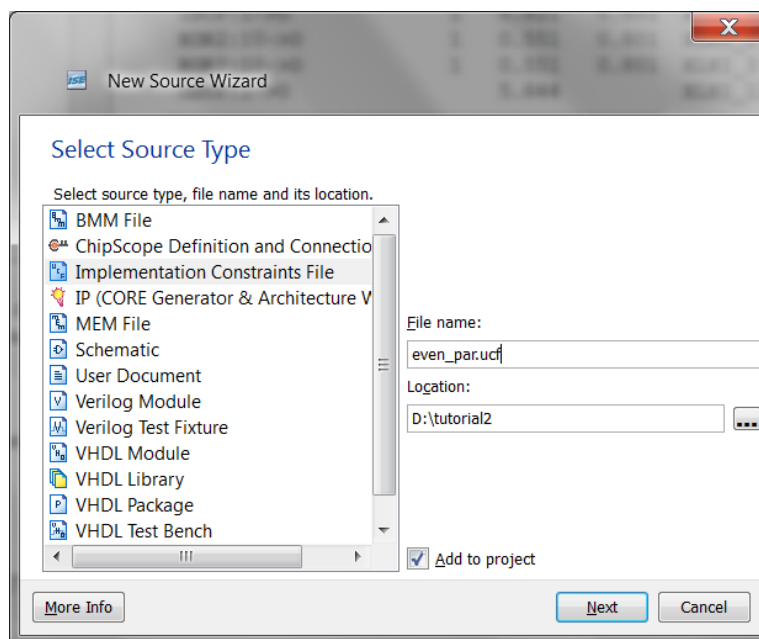
Το υψηλότερο επίπεδο της σχεδίασης του παραδείγματος έχει 4 θύρες εισόδου (a, b, c, d), και μια θύρα εξόδου (ep). Άρα, θέλουμε να έχουμε 4 διακόπτες, SW0, SW1, SW2 και SW3, που συνδέονται με τις εισόδους. Επιπλέον, θέλουμε την έξοδο να συνδέεται με μια ενδεικτική λυχνία (LED) έτσι ώστε να μπορούμε να την παρατηρήσουμε – το LD0 είναι κατάλληλο για αυτόν τον σκοπό.

Εάν επιθεωρήσετε την πάνω πλευρά της πλακέτας Spartan-3E Starter Kit, θα παρατηρήσετε ότι σχεδόν κάθε πόρος έχει σχολιαστεί με κάποιο κείμενο που προσδιορίζει με ποιους ακροδέκτες του FPGA συνδέεται. Αυτές οι πληροφορίες είναι επίσης διαθέσιμες στον οδηγό χρήσης της πλακέτας (Spartan-3E Starter Kit User Guide). Προσπαθήστε να προσδιορίσετε ποιοι ακροδέκτες του FPGA χρησιμοποιούνται για τα SW0, SW1, SW2, SW3 και LD0, και ελέγξτε έπειτα τα αποτελέσματά σας με αυτά που παρουσιάζονται παρακάτω:

- SW0 → FPGA Pin N17
- SW1 → FPGA Pin H18
- SW2 → FPGA Pin L14
- SW3 → FPGA Pin L13
- LD0 → FPGA Pin C11

Έχετε τώρα αρκετές πληροφορίες για να δημιουργήσετε αυτό που ονομάζεται αρχείο περιορισμών του χρήστη (user constraint file), ή UCF. Αυτό το αρχείο περιέχει τους περιορισμούς της σχεδίασης που δεν καθορίσατε στην περιγραφή VHDL, όπως οι περιορισμοί θέσης των ακροδεκτών (pin location) και απόδοσης της σχεδίασης (design performance). Είναι βολικό να παρασχεθούν σε ένα UCF παρά στην VHDL περιγραφή. Για παράδειγμα, εάν κάνετε ένα λάθος στις αναθέσεις των ακροδεκτών, δεν χρειάζεται να επιστρέψετε και να επανασυνθέσετε το κύκλωμά σας.

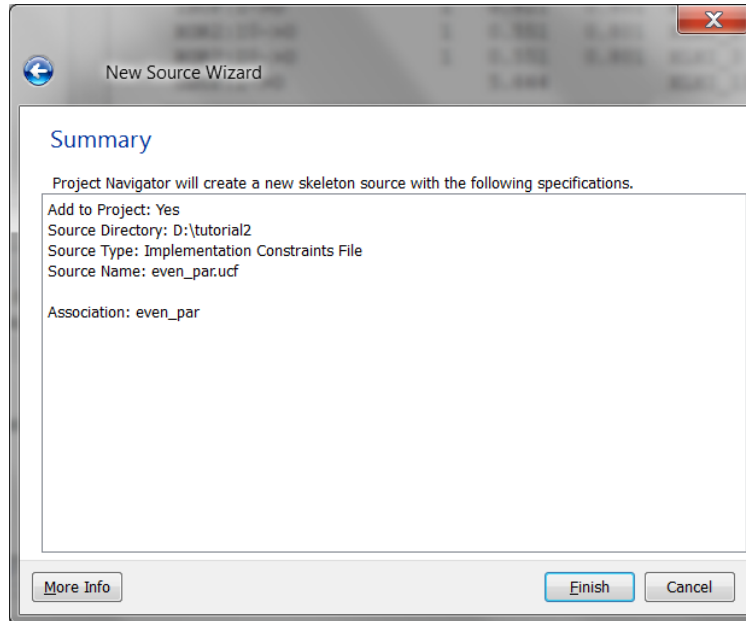
Μπορείτε να προσθέσετε ένα UCF στο έργο χρησιμοποιώντας την ίδια διαδικασία που χρησιμοποιήσατε για την προσθήκη της σχεδίασης και του testbench. Δημιουργήστε ένα νέο πηγαίο αρχείο: επιλέξτε Project→New Source από το κυρίως μενού. Το πρώτο από τα κουτιά διαλόγου New Source θα εμφανιστεί, όπως φαίνεται στην Εικόνα 28.



**Εικόνα 28: New Source Wizard (1 από 2)**

Επιλέξτε Implementation Constraints File για να δηλώσετε ότι θέλετε να δημιουργήσετε ένα αρχείο περιορισμών. Κατόπιν, δώστε ένα όνομα αρχείου όπως φαίνεται στην Εικόνα 28. Δεν χρειάζεται να αλλάξετε την καθορισμένη θέση, η οποία είναι μέσα στον κατάλογο του έργου που έχετε δημιουργήσει. Πατήστε Next.

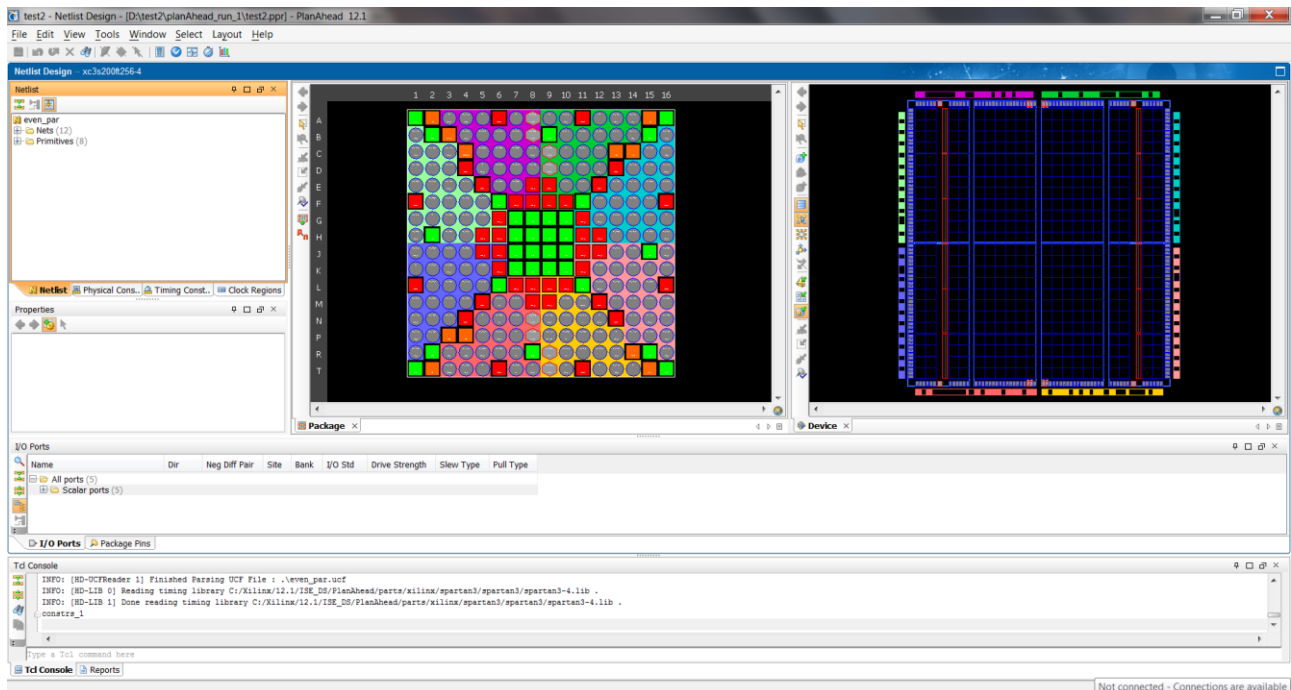




**Εικόνα 29: New Source Wizard (2 από 2)**

Το τελικό πλαίσιο διαλόγου (Εικόνα 29) παρέχει μια περίληψη του αρχείου που το Project Navigator θα δημιουργήσει βασισμένο στις ρυθμίσεις σας. Διαβάστε την περίληψη για να σιγουρευτείτε ότι ταιριάζει με την Εικόνα 29. Εάν όχι, πατήστε Back και διορθώστε οποιαδήποτε λάθη. Διαφορετικά, πατήστε Finish για να ολοκληρώσετε την διαδικασία.

Το User Constraints File (UCF) είναι ένα αρχείο κειμένου στο οποίο μπορεί άμεσα να γράψει ο χρήστης μέσω του ενσωματωμένου κειμενογράφου του ISE. Επίσης, ένας άλλος εύκολος τρόπος για την επεξεργασία του αρχείου UCF είναι με την βοήθεια δύο γραφικών εργαλείων, του Constraint Editor και του PlanAhead™. Χρησιμοποιώντας το PlanAhead είναι δυνατόν να κάνουμε τις αντιστοιχίες των I/O Pins του FPGA με τα σήματα της σχεδίασης μας. Το PlanAhead είναι ένα αρκετά ισχυρό εργαλείο με πολλές δυνατότητες αλλά θα χρησιμοποιήσουμε μόνο μια μικρή μερίδα των δυνατοτήτων του σε αυτόν τον οδηγό εκμάθησης.



**Εικόνα 30: PlanAhead Software**

Για να ξεκινήσουμε το PlanAhead επιλέγουμε το `even_par` (`even_par.sch`) από το tab Implementation του μενού Design. Επεκτείνουμε το μενού User Constraints πατώντας το + δίπλα του και επιλέγουμε την εντολή I/O Pin Planning (PlanAhead) – Post-Synthesis. Όπως παρατηρήσατε υπάρχει και η επιλογή I/O Pin Planning (PlanAhead) – Pre-Synthesis. Είναι προτιμότερο να χρησιμοποιείτε την επιλογή Post-Synthesis, δεδομένου ότι μετά την σύνθεση παρέχεται περισσότερη πληροφορία για τα I/O του κάθε design που έχετε δημιουργήσει. Η Εικόνα 30 παρουσιάζει το πρόγραμμα PlanAhead.

Στο μενού I/O Ports επεκτείνετε την επιλογή Scalar Ports κάτω από το All Ports. Σε αυτό το σημείο θα πρέπει να βλέπετε τις τέσσερις εισόδους (a, b, c, d) και τη μία έξοδο (ep) της σχεδίασής σας. Επιλέξτε μία προς μία τις εισόδους και τις εξόδους της σχεδίασής σας και πραγματοποιήστε drag and drop στα αντίστοιχα pin του Package Tool.

- a → FPGA Pin N17
- b → FPGA Pin H18
- c → FPGA Pin L14
- d → FPGA Pin L13
- ep → FPGA Pin C11

Κλείνοντας αποθηκεύστε τις αλλαγές σας από το μενού File→Save Project και κλείστε το εργαλείο PlanAhead. Επιστρέψτε στο Project Navigator, επιλέξτε το αρχείο `even_par.ucf` και εκτελέστε την εντολή Edit Constraints (Text) ώστε να επαληθεύσετε τις αντιστοιχίες των ακροδεκτών που μόλις κάνατε. Σε αυτό το σημείο πρέπει να δείτε τα παρακάτω:

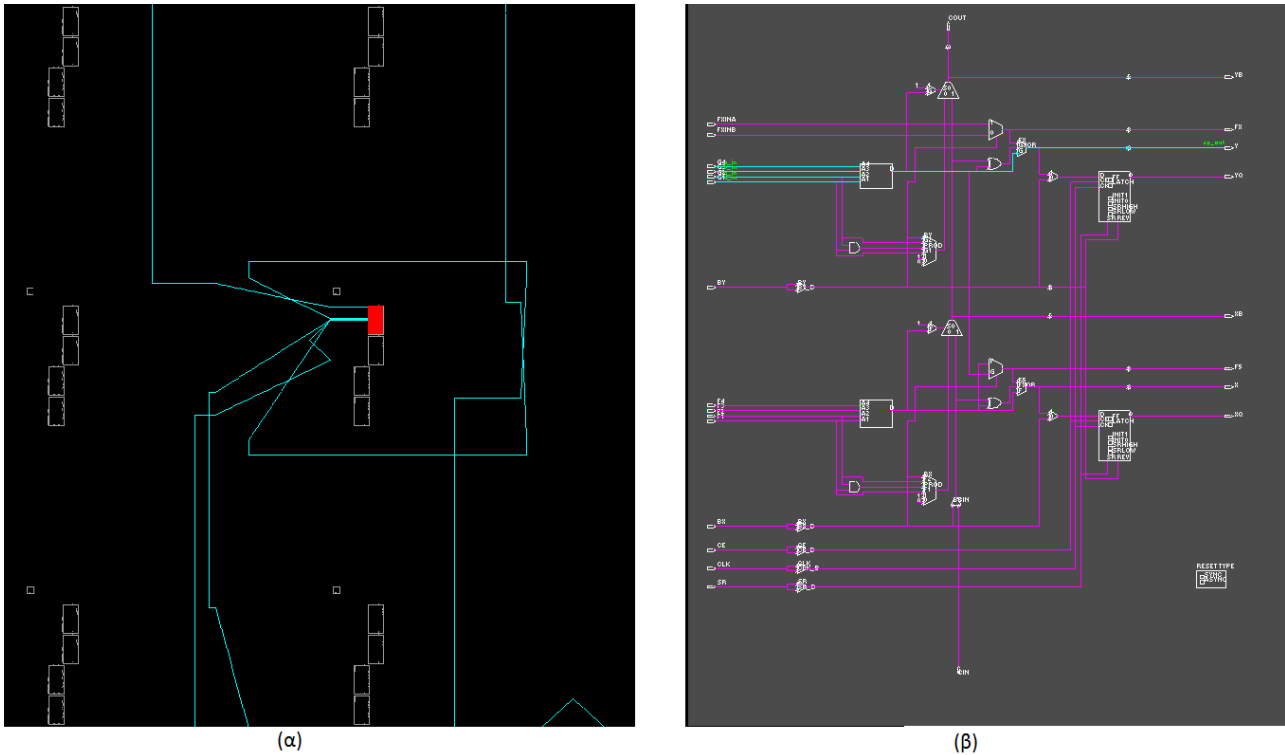
```
# PlanAhead Generated physical constraints
```

```
NET "a" LOC = N17;
NET "b" LOC = H18;
NET "c" LOC = L14;
NET "d" LOC = L13;
NET "ep" LOC = C11;
```

Τώρα που έχετε ένα αρχείο περιορισμών στο έργο σας, μπορείτε να υλοποιήσετε τη σχεδίαση. Επιλέξτε το `even_par` στο παράθυρο Design. Κατόπιν, κάντε διπλό-κλικ στη διεργασία Implement Design στο παράθυρο Processes: `even_par`. Το Project Navigator θα υλοποιήσει τη σχεδίαση και θα εμφανίσει πληροφορίες στο παράθυρο Console. Σαν ενημερωτική σημείωση, είναι δυνατό να αλλάξετε τις επιλογές υλοποίησης κάνοντας δεξί κλικ στο Implement Design και έπειτα επιλέγοντας Process Properties. Για αυτόν τον οδηγό εκμάθησης, εντούτοις, αφήστε τις επιλογές στις προεπιλεγμένες ρυθμίσεις τους.

Δεν πρέπει να δείτε κανένα λάθος στο παράθυρο Console. Εντούτοις, πρέπει πάντα να διαβάζετε τα τρία αρχεία log, τα οποία είναι διαθέσιμα κάνοντας δεξί click πάνω στα Translate, Map, και Place and Route και εκτελώντας την εντολή View Text Report. Εάν δεν καταλαβαίνετε ένα συγκεκριμένο μήνυμα, ψάξτε στην ιστοσελίδα υποστήριξης της Xilinx. Σε αυτό το σημείο, πρέπει να έχετε ένα πράσινο σημάδι (✓) δίπλα στη διεργασία Implement Design.

Επεκτείνεται την διεργασία Implement Design και κατόπιν την διεργασία Place and Route. Εκτελέστε την διεργασία View/Edit Routed Design (FPGA Editor). Θα σας ανοίξει ο FPGA Editor μέσα από τον οποίο μπορείτε να δείτε την υλοποίηση σας στο FPGA. Χρωματίστε τα a, b, c, d, ep, ep\_out από την επιλογή Hilited και κάντε zoom στο κόκκινο σημείο το οποίο και αποτελεί ένα slice του FPGA Εικόνα 31(α). Κάνοντας διπλό click στο slice βλέπετε την υλοποίηση του component ep\_out στο FPGA σύμφωνα με την Εικόνα 31(β).

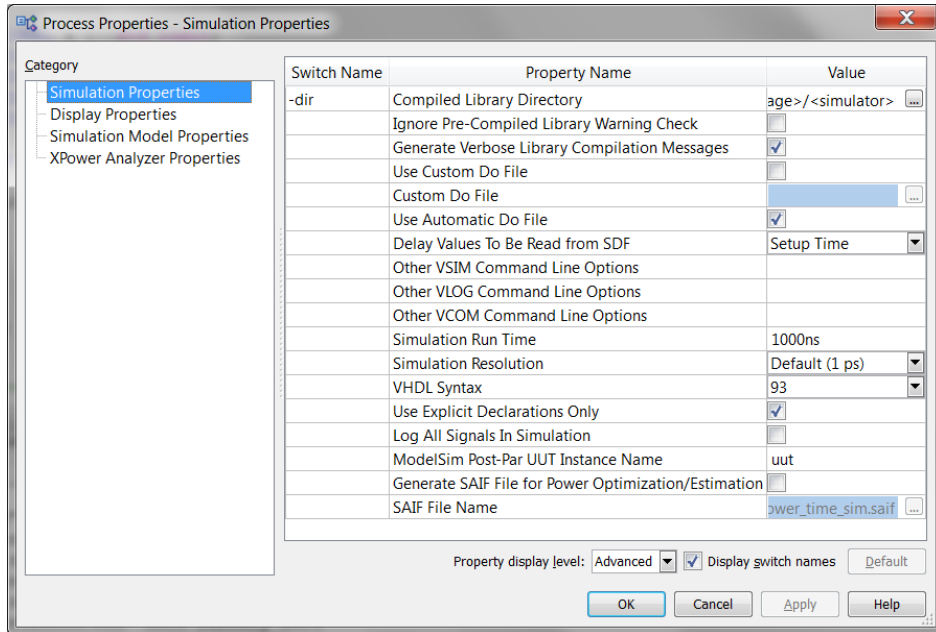


Εικόνα 31 (α) Routed Design, (β) Component ep\_out

## 7 Χρονική προσομοίωση (timing simulation)

Αφού ολοκληρώσετε τα βήματα της υλοποίησης, μπορείτε να προσομοιώσετε πάλι της σχεδιάσή σας – αυτή τη φορά, χρησιμοποιώντας μια αναπαράσταση δομής της σχεδίασής σας που έχει γίνει σύνθεση, έχει τοποθετηθεί, και δρομολογηθεί (synthesized, placed and routed design) και επίσης περιέχει πληροφορίες καθυστέρησης της χειρότερης περίπτωσης (worst-case delay information). Η ιδέα είναι να προσομοιώσετε τη σχεδιάσή σας, όπως αυτή υλοποιείται φυσικά στη συσκευή FPGA.

Οι διαδικασίες προσομοίωσης σας επιτρέπουν να εκτελέσετε την προσομοίωση της σχεδιάσής σας χρησιμοποιώντας το Modelsim. Για να εντοπίσετε τις διαδικασίες του προσομοιωτή Modelsim, από το παράθυρο Design επιλέξτε το μενού Simulation και ενεργοποιήστε την επιλογή Post-Route ακριβώς κάτω από το Simulation. Κατόπιν, κάντε κλικ στο + δίπλα στο στοιχείο Modelsim Simulator στο παράθυρο Processes: even\_par\_testbench\_vhd\_tb – behavior (testbench.vhd) για να το επεκτείνετε. Θα εκτελέσετε μια χρονική προσομοίωση χρησιμοποιώντας το Simulate Post-Place & Route VHDL Model αλλά πρέπει πρώτα να καθορίσετε τις ιδιότητες της διαδικασίας της προσομοίωσης, όπως ακριβώς κάνατε για τη λειτουργική προσομοίωση. Δεξί κλικ στο Simulate Post-Place & Route VHDL Model, και επιλογή του Process Properties. Το πλαίσιο διαλόγου Process Properties εμφανίζεται, όπως φαίνεται στην Εικόνα 32. Σιγουρευτείτε ότι οι ιδιότητες έχουν τις τιμές που φαίνονται στην Εικόνα 32. Πατήστε OK.



**Εικόνα 32: Simulation Process Properties**

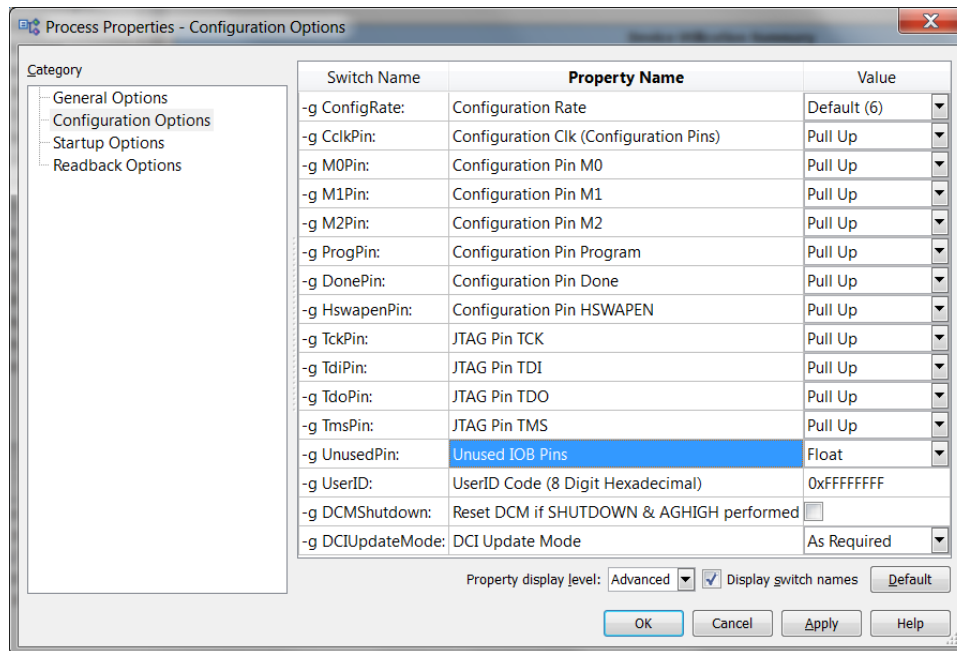
Για να αρχίσετε την προσομοίωση, κάντε διπλό κλικ στο Simulate Post-Place & Route VHDL Model. Το Modelsim δημιουργεί έναν κατάλογο εργασίας, μεταγλωττίζει τα πηγαία αρχεία, φορτώνει τη σχεδίαση, και εκτελεί την προσομοίωση για τον καθορισμένο χρόνο. Ο προσομοιωτής θα τρέξει, και θα δείτε τα αποτελέσματα όπως πριν. Θα δείτε ότι τα αποτελέσματα διαφέρουν από αυτά της λειτουργικής προσομοίωσης. Εάν εξετάσετε το παράθυρο Wave, θα δείτε ότι υπάρχει μια χρονική καθυστέρηση στην διάδοση των τιμών από τις εισόδους στην έξοδο του κυκλώματος. Η ακριβής καθυστέρηση εξαρτάται από τη συσκευή που χρησιμοποιείτε (FPGA device), την τοποθέτηση (placement), τη δρομολόγηση (routing), και τη σχεδίασή σας (design).

Σε αυτό το σημείο, είστε έτοιμοι να προγραμματίσετε το FPGA με τη σχεδίασή σας. Η πλακέτα Spartan-3E Starter Kit μπορεί να προγραμματιστεί με δύο διαφορετικές μεθόδους. Η πρώτη είναι να προγραμματιστεί το FPGA με ένα usb καλώδιο (download cable). Η άλλη είναι να προγραμματιστεί η συσκευή PROM με το usb καλώδιο (download cable), και έπειτα να έχουμε την PROM να προγραμματίζει το FPGA. Και οι δύο μέθοδοι καλύπτονται στην επόμενη ενότητα.

## 8 Προγραμματισμός του FPGA (FPGA programming)

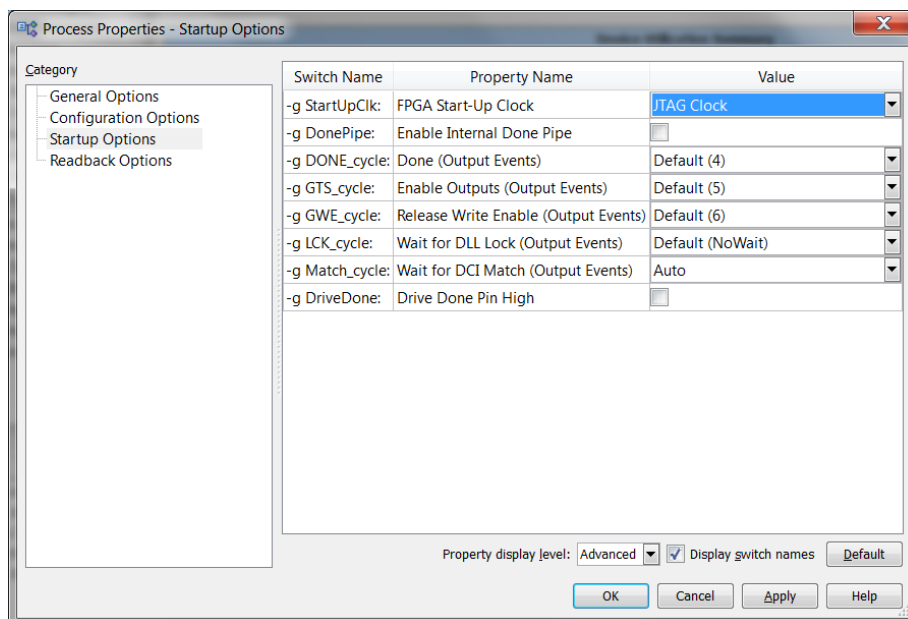
Προγραμματίζοντας το FPGA κατευθείαν με το καλώδιο είναι ένας βολικός τρόπος για να δοκιμαστεί μια σχεδίαση. Αυτή η μέθοδος είναι χρήσιμη όταν θέλετε να δοκιμάσετε γρήγορα κάτι, ή αν δεν είστε σίγουροι ότι η σχεδίασή σας είναι τελική. Για παράδειγμα, σε αυτό το σημείο είστε αρκετά βέβαιοι ότι το κύκλωμά σας είναι σωστό. Εντούτοις, πρέπει να κατανοήσετε ότι τα σύνθετα κυκλώματα σπάνια δουλεύουν με την πρώτη δοκιμή. Ένα από τα μεγάλα πλεονεκτήματα των FPGAs έναντι των ASICs είναι ότι το κόστος που πληρώνετε εάν έχετε κάνει λάθος στην πρώτη δοκιμή είναι ελάχιστο.

Το πρώτο βήμα είναι να δημιουργήσετε ένα αρχείο προγραμματισμού για το FPGA. Επιλέξτε το even\_par στο παράθυρο Design με ενεργοποιημένο το Implementation. Στο παράθυρο Processes: even\_par, κάντε δεξί-κλικ στο Generate Programming File έπειτα επιλέξτε Process Properties. Το πλαίσιο διαλόγου Process Properties εμφανίζεται. Επιλέξτε την ετικέτα Configuration Options, όπως φαίνεται στην Εικόνα 33.



**Εικόνα 33: Programming File Properties (Configuration Options)**

Αλλάξτε την επιλογή Unused IOB Pins σε Float. Οι άλλες ρυθμίσεις πρέπει ήδη να είναι σωστές, αλλά σιγουρευτείτε ότι ταιριάζουν με την Εικόνα 33. Έπειτα, επιλέξτε την ετικέτα Startup Options, όπως φαίνεται στην Εικόνα 34.

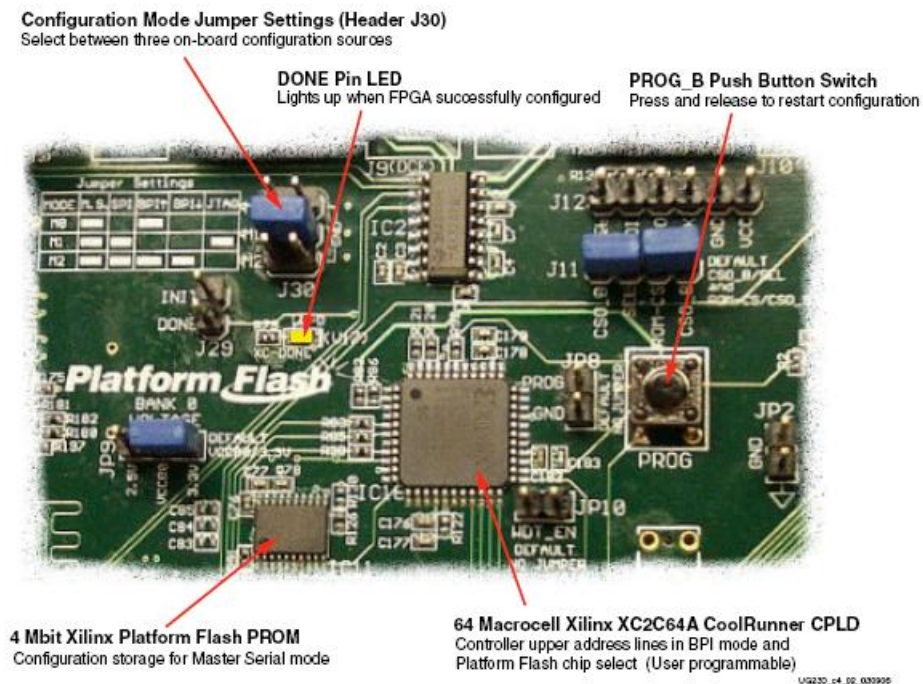


**Εικόνα 34: Programming File Properties (Startup Options)**

Αλλάξτε την επιλογή FPGA Start-Up Clock σε JTAG Clock. Οι άλλες ρυθμίσεις πρέπει ήδη να είναι σωστές, αλλά σιγουρευτείτε ότι ταιριάζουν με την Εικόνα 34. Πατήστε OK για να αποθηκεύσετε τις ρυθμίσεις.

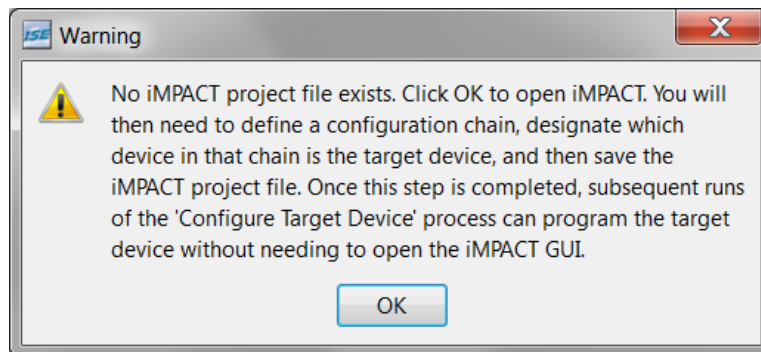
Επιβεβαιώστε ότι το even\_par είναι επιλεγμένο. Έπειτα, κάντε διπλό-κλικ στην διαδικασία Generate Programming File στο παράθυρο Processes: even\_par. Το Project Navigator θα παράγει ένα αρχείο προγραμματισμού και θα εμφανίσει πληροφορίες στο παράθυρο Console.

Πριν συνεχίσετε, πρέπει να έχετε διαθέσιμα την πλακέτα Spartan-3E Starter Kit, την τροφοδοσία, και το καλώδιο προγραμματισμού. Συνδέστε το καλώδιο προγραμματισμού στην usb θύρα του PC που χρησιμοποιείτε. Ελέγξτε αν οι διακλαδωτήρες (jumpers) J30 της πλακέτας Spartan-3E είναι τοποθετημένοι σωστά σύμφωνα με την Εικόνα 35 ώστε ο προγραμματισμός να πραγματοποιηθεί μέσω του JTAG.



Εικόνα 35: Τοποθέτηση jumper της αναπτυξιακής πλακέτας

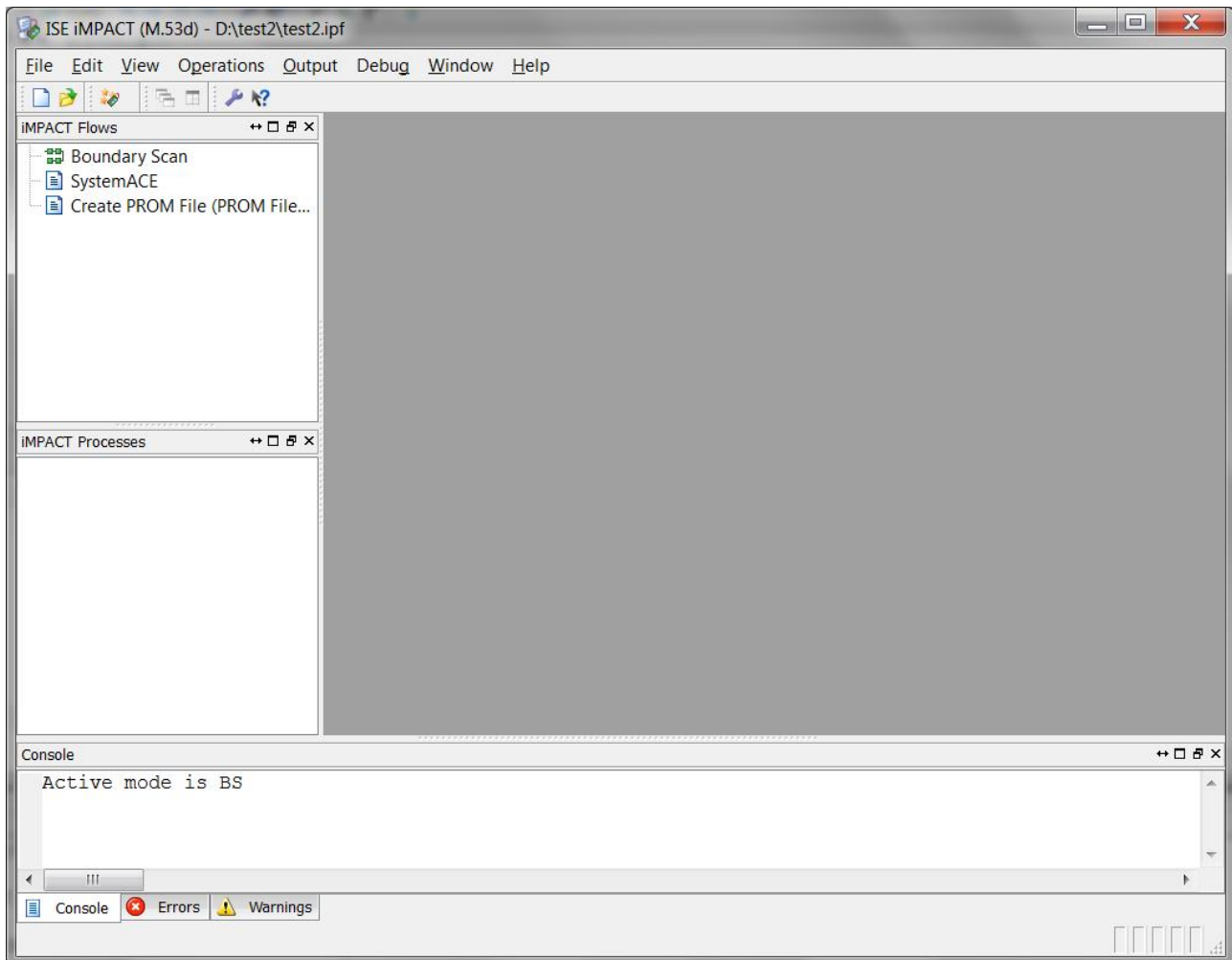
Για να κατεβάσετε (προγραμματίσετε) το αρχείο προγραμματισμού στη συσκευή FPGA, κάντε διπλό click στο Configure Target Device. Αυτό θα φορτώσει το πρόγραμμα ISE iMPACT σε άλλο παράθυρο. Την πρώτη φορά που θα εκτελέσετε αυτή την εντολή θα λάβετε ένα μήνυμα όπως αυτό που φαίνεται στην Εικόνα 36 σχετικά με τον καθορισμό του iMPACT αρχείου.



Εικόνα 36: Μήνυμα σχετικά με το iMPACT αρχείο

Κατόπιν θα εμφανιστεί στην οθόνη σας ένα νέο εργαλείο, το iMPACT όπως φαίνεται στην Εικόνα 37 το οποίο σας επιτρέπει να προγραμματίσετε το FPGA.

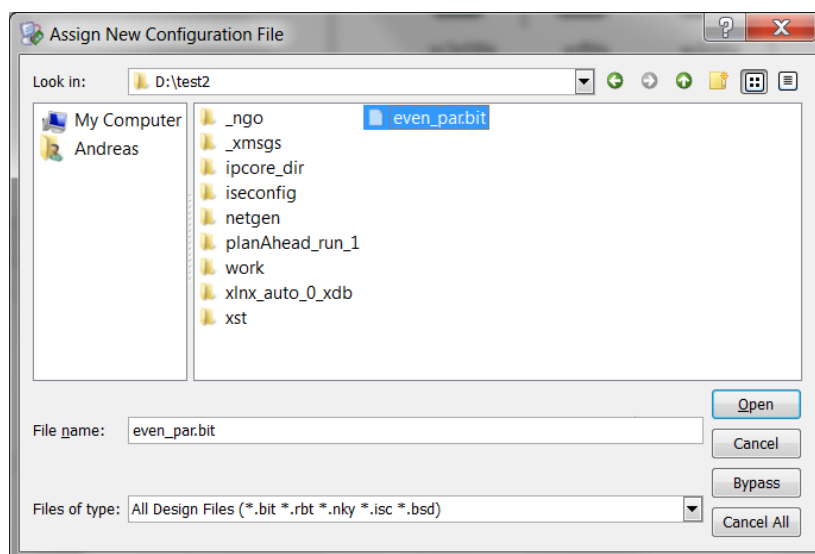




**Εικόνα 37: Εργαλείο προγραμματισμού iMPACT**

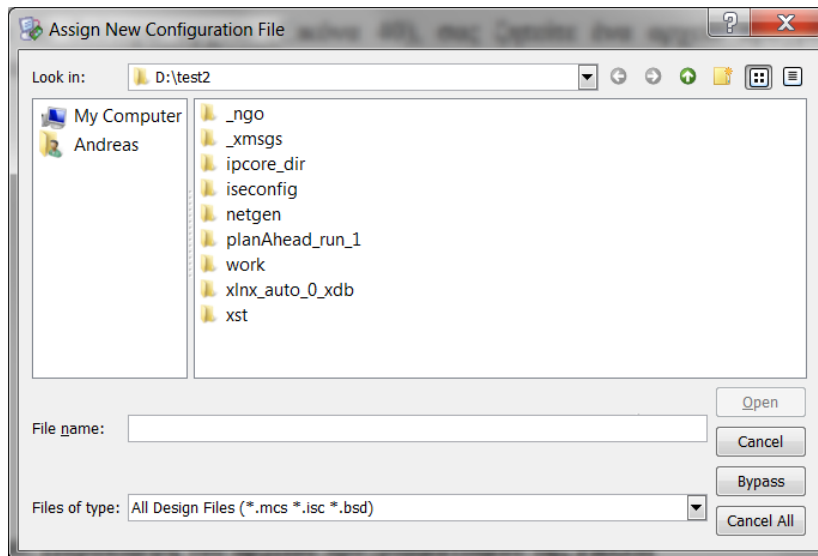
Κάντε διπλό click στο boundary Scan και έπειτα δεξί click στο Right Click to add Device or Initialize JTAG chain και δώστε την εντολή Initialize Chain.

Έπειτα, πρέπει να σας ζητηθούν τρία αρχεία. Για το πρώτο αρχείο (Εικόνα 38), επιλέξτε το even\_par.bit που δημιουργήσατε με τη διαδικασία υλοποίησης. Αυτό είναι το αρχείο προγραμματισμού του FPGA.



**Εικόνα 38: Επιλογή του αρχείου προγραμματισμού του FPGA**

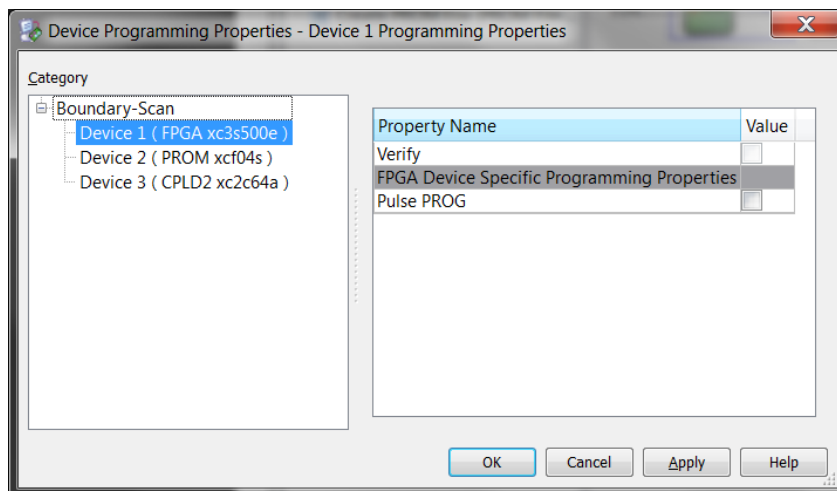
Για το δεύτερο αρχείο (Εικόνα 39), σας ζητείτε ένα αρχείο προγραμματισμού της PROM. Δεν προγραμματίζουμε την PROM αυτήν τη στιγμή, επομένως επιλέξτε Bypass όπως ακριβώς και στο τρίτο πλαίσιο διαλόγου.



**Εικόνα 39: Παράκαμψη του αρχείου προγραμματισμού της PROM**

Σημείωση: Εάν δεν σας ζητηθούν σε αυτό το σημείο τα δύο αρχεία, αγνοήστε το και συνεχίστε με τα παρακάτω βήματα. Μάλλον, πρόκειται για bug του προγράμματος iMPACT.

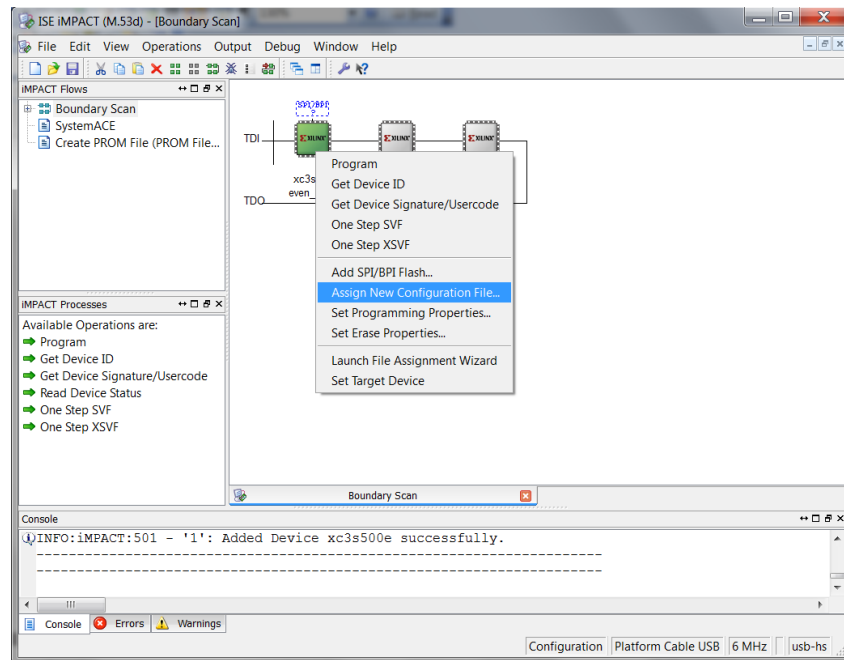
Στο τέλος της διαδικασίας λαμβάνεται ένα πλαίσιο διαλόγου όπως αυτό που φαίνεται στην Εικόνα 40.



**Εικόνα 40: Τελικές ρυθμίσεις προγραμματισμού του FPGA**

Σε αυτό το σημείο, είστε έτοιμοι να προγραμματίσετε το FPGA. Εάν έχετε κάνει κάποιο λάθος, μπορείτε να διορθώσετε τις αναθέσεις σας με τη χρησιμοποίηση της τεχνικής που απεικονίζεται στην Εικόνα 41.

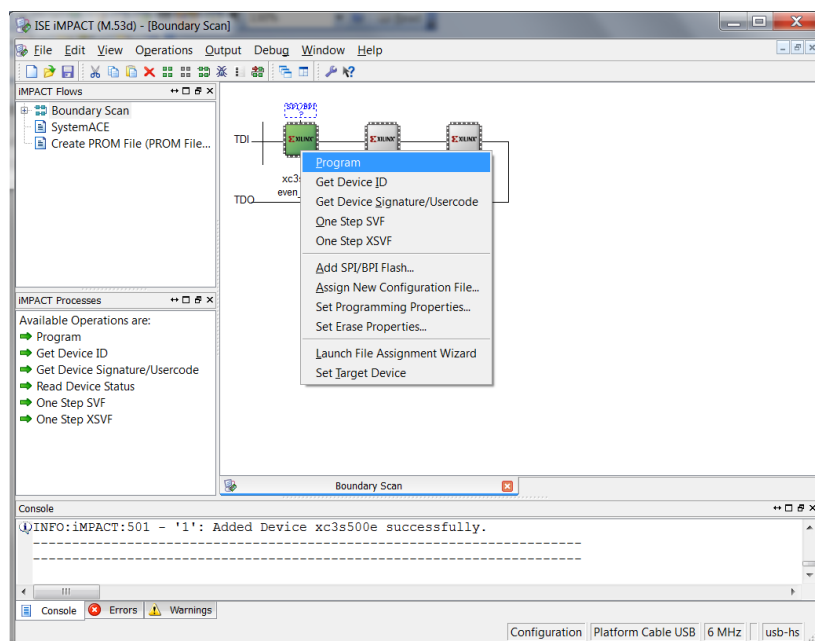




**Εικόνα 41: Εναλλακτική μέθοδος ανάθεσης των αρχείων προγραμματισμού**

Για να διορθώσετε μια ανάθεση αρχείου, ή να κάνετε τις αρχικές αναθέσεις εάν το iMPACT δεν σας προτρέπει αυτόματα για τα αρχεία, επιλέξτε το εικονίδιο του FPGA στο παράθυρο iMPACT. Κάντε δεξί κλικ και επιλέξτε Assign New Configuration File. Θα σας ζητηθεί ένα αρχείο όπως φαίνεται στην Εικόνα 38. Μπορείτε να επαναλάβετε αυτήν την διαδικασία με το εικονίδιο της PROM, και θα σας ζητηθεί ένα άλλο αρχείο όπως φαίνεται στην Εικόνα 39.

Τέλος, θα φθάσετε στο σημείο που φαίνεται στην Εικόνα 42. Το iMPACT είναι έτοιμο να προγραμματίσει το FPGA. Επιλέξτε το εικονίδιο του FPGA στο παράθυρο και έπειτα χρησιμοποιήστε το δεξί κουμπί του ποντικιού για να ενεργοποιήσετε το μενού και επιλέξτε την επιλογή Program.



**Εικόνα 42: Program device**

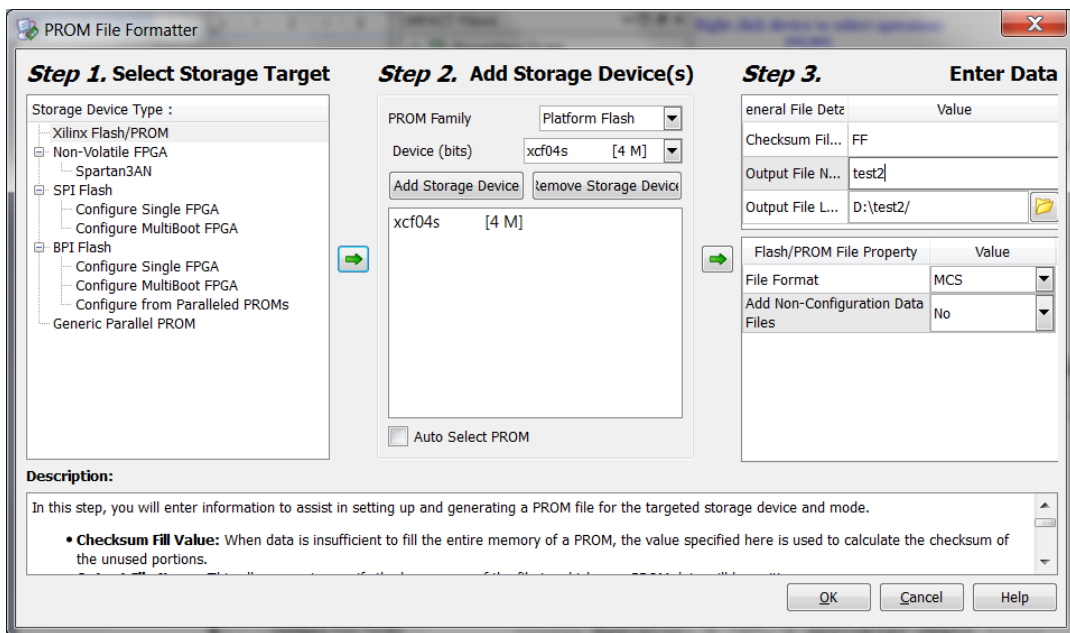
Ένας δείκτης προόδου θα εμφανιστεί. Μόλις ο προγραμματισμός ολοκληρωθεί, το πρόγραμμα θα σας ενημερώσει εάν ο προγραμματισμός πέτυχε ή απέτυχε. Εάν ο προγραμματισμός έχει αποτύχει, επανελέγξτε τις συνδέσεις των καλωδίων σας, τη σύνδεση τροφοδοσίας και προσπαθήστε ξανά. Εάν συνεχίσει να αποτυγχάνει, ζητήστε τη βοήθεια του καθηγητή.

Τώρα, μπορείτε να δοκιμάσετε τη σχεδιάσή σας στο υλικό. Εντοπίστε τους διακόπτες SW0-SW3 στην πλακέτα, και εξετάστε το κύκλωμά σας δοκιμάζοντας τους 16 πιθανούς συνδυασμούς των τιμών των διακοπών και παρατηρώντας το LD0. Το κύκλωμα σας συμπεριφέρεται όπως αναμένετε; Εάν όχι, ζητήστε τη βοήθεια του καθηγητή. Εάν λειτουργεί σωστά, είστε έτοιμοι να δοκιμάσετε την άλλη μέθοδο προγραμματισμού. Βγείτε από το πρόγραμμα iMPACT (δεν χρειάζεται να σώσετε). Μην αποσυνδέσετε το καλώδιο τροφοδοσίας και προγραμματισμού.

## 9 Προγραμματισμός της PROM (PROM programming)

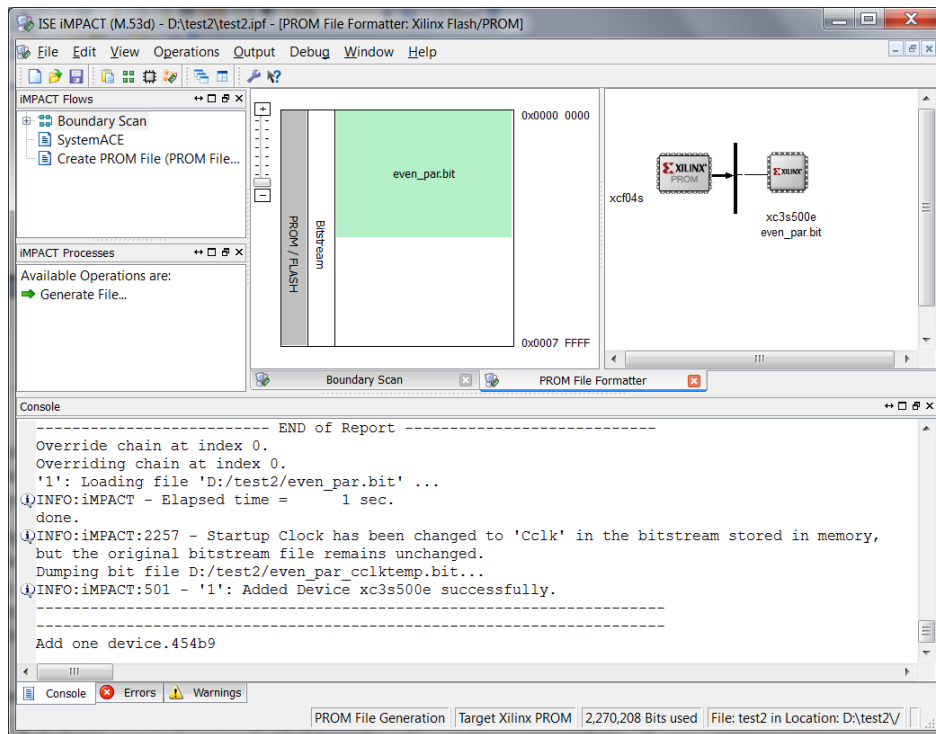
Η άλλη μέθοδος είναι να προγραμματίσετε την PROM με το καλώδιο, και να έχετε την PROM να προγραμματίζει το FPGA. Τυπικά, προγραμματίζουμε την PROM όταν θεωρούμε ότι το κύκλωμά μας είναι εντελώς σωστό. Αφότου προγραμματίσουμε την PROM, κάθε φορά που ανάβουμε την τροφοδοσία, το FPGA θα φορτώνει αυτόματα το αρχείο προγραμματισμού από την PROM. Αφότου προγραμματιστεί η PROM, δεν απαιτείται καλώδιο προγραμματισμού.

Επεκτείνετε τη διαδικασία Configure Target Device κάνοντας κλικ στο + δίπλα σε αυτήν, και έπειτα κάντε διπλό-κλικ στη διαδικασία Generate Target PROM/ACE File. Αυτό θα φορτώσει πάλι το πρόγραμμα iMPACT. Από το μενού iMPACT Flows εκτελέστε την εντολή Create PROM File. Θα εμφανιστεί η Εικόνα 43 όπου στο πρώτο βήμα επιλέγετε το Xilinx Flash/PROM. Στο δεύτερο βήμα επιλέγετε τη μνήμη Flash της πλατφόρμας, όπου στην περίπτωση της πλακέτας Spartan-3E είναι η xcf04s. Στο τρίτο και τελευταίο βήμα δίνετε ένα όνομα για το αρχείο προγραμματισμού της rom (test2 στη περίπτωση μας) και ορίζετε τον τύπο του αρχείου (mcs στην περίπτωση μας) και ολοκληρώνετε την διαδικασία.



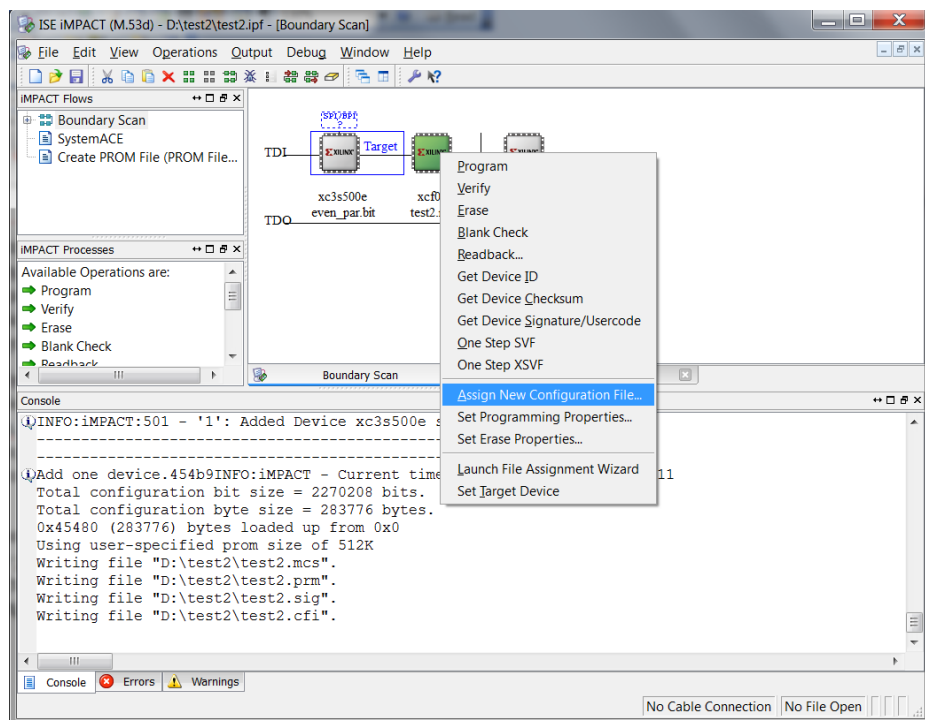
Εικόνα 43: Βήματα καθορισμού αρχείου προγραμματισμού της PROM

Όταν σας ζητηθεί το νέο αρχείο, επιλέξτε το αρχείο even\_par.bit, το οποίο είναι το ίδιο που χρησιμοποιήσατε πριν. Αφού προσθέσετε το αρχείο even\_par.bit, το iMPACT θα σας ρωτήσει εάν θέλετε να προσθέσετε και άλλο αρχείο σχεδίασης στα δεδομένα της PROM. Πατήστε NO. Θυμηθείτε ότι προηγουμένως, κατά τη δημιουργία του αρχείου προγραμματισμού, θέσατε την επιλογή ρολογιών Startup Clock σε JTAG Clock. Αυτή η ρύθμιση απαιτείται όταν προγραμματίζουμε το FPGA κατευθείαν από το καλώδιο, αλλά για τον προγραμματισμό της PROM πρέπει να χρησιμοποιηθεί η ρύθμιση CCLK. Το iMPACT κάνει αυτήν την αλλαγή για σας χωρίς να χρειάζεται να ξανατρέξετε τη διαδικασία Generate Programming File. Το τελικό αποτέλεσμα παρουσιάζεται στην Εικόνα 44. Τέλος εκτελέστε την εντολή Generate File από το μενού iMPACT Processes. Αν όλα έχουν πάει καλά θα λάβετε το μήνυμα “Generate Succeeded”.

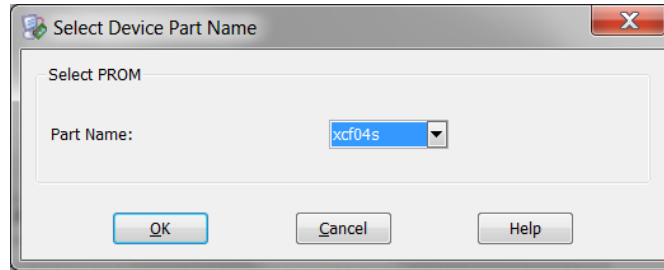


**Εικόνα 44: Τελικό αποτέλεσμα δημιουργίας αρχείου για την PROM**

Επιστρέψτε στην επιλογή Boundary Scan κάντε δεξί click στο xc3s500e και εκτελέστε την εντολή Assign New Configuration File όπως φαίνεται στη Εικόνα 45. Αμέσως μετά θα σας ζητηθεί να ορίσετε την RPOM. Ορίστε την xc3s500e όπως φαίνεται στην Εικόνα 46.

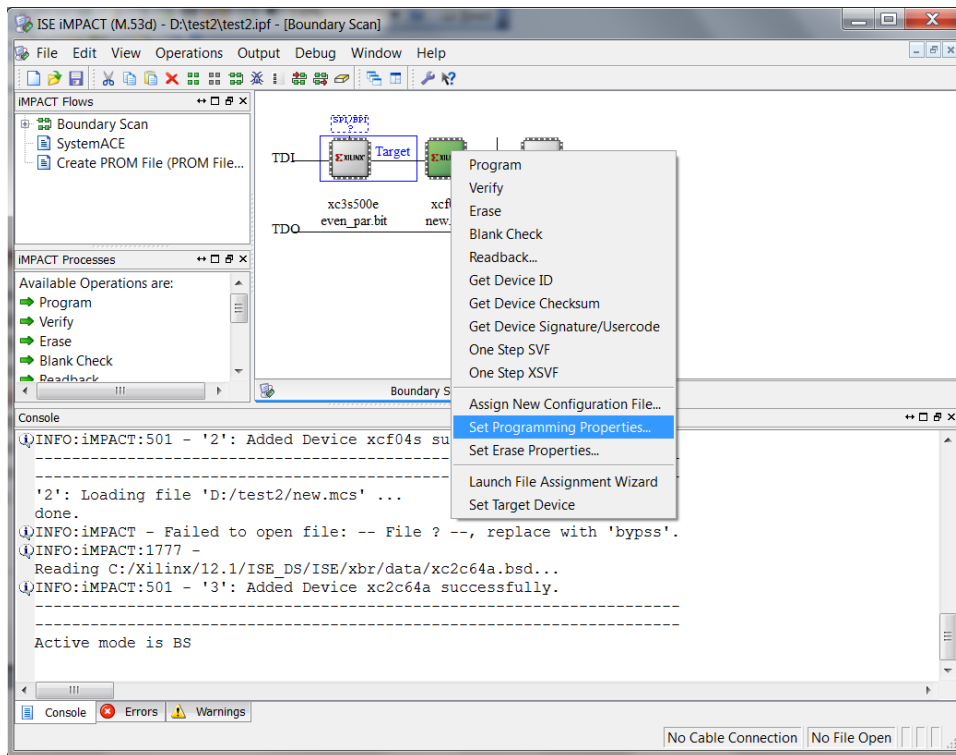


**Εικόνα 45: Ανάθεση νέου αρχείου προγραμματισμού στην PROM**



**Εικόνα 46: Ορισμός τύπου της PROM**

Πριν εκτελέσετε την εντολή Program πρέπει να αλλάξετε την τοποθέτηση των βραχυκυκλωτήρων (jumper) J30 της πλακέτας αφού βέβαια την αποσυνδέσετε από την τροφοδοσία. Ανατρέξτε στην Εικόνα 35 για να βρείτε τα jumper J30 της πλακέτας, όπου πρέπει να προσθέσετε και τα υπόλοιπα δύο jumper ώστε να πραγματοποιήσετε τον προγραμματισμό του FPGA μέσω της PROM. Τροφοδοτήστε εκ' νέου την αναπτυξιακή πλακέτα. Από το iMPACT πατήστε δεξί click πάνω στην PROM (xcf04s) και εκτελέστε την εντολή Set Programming Properties όπως φαίνεται στην Εικόνα 47. Ένα νέο παράθυρο θα ανοίξει το οποίο θα σας ζητήσει να καθορίσετε τις ρυθμίσεις προγραμματισμού της PROM. Ορίστε τις επιλογές του όπως φαίνονται στην Εικόνα 48. Η τελευταία επιλογή ορίζει την δυνατότητα προγραμματισμού του FPGA μέσω της PROM και είναι και ο λόγος για τον οποίο αλλάξαμε τα jumper της πλακέτας μας.

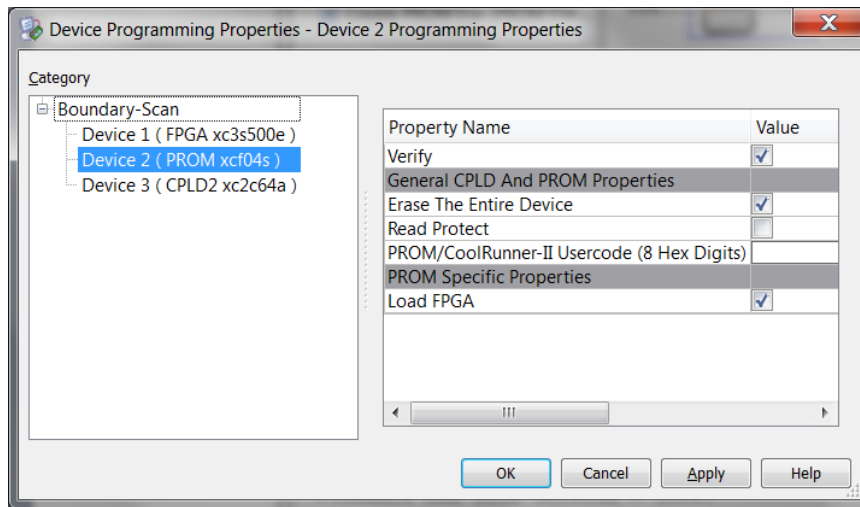


**Εικόνα 47: Ορισμός παραμέτρων προγραμματισμού της PROM**

Όπως ακριβώς και πριν κάντε δεξί click πάνω στο device της prom και εκτελέστε την εντολή Program. Ένας δείκτης προόδου θα εμφανιστεί. Μόλις ο προγραμματισμός ολοκληρωθεί, το πρόγραμμα θα σας ενημερώσει εάν ο προγραμματισμός πέτυχε ή απέτυχε. Εάν ο προγραμματισμός έχει αποτύχει, επανελέγξτε τις συνδέσεις των καλωδίων σας, τη σύνδεση της τροφοδοσίας, και τους βραχυκυκλωτήρες – και προσπαθήστε ξανά. Εάν αποτύχει ξανά, ζητήστε τη βοήθεια του καθηγητή.

Τώρα, μπορείτε πάλι να δοκιμάσετε το κύκλωμά σας. Βγείτε από το πρόγραμμα iMPACT (δεν χρειάζεται να σώσετε). Αποσυνδέστε το καλώδιο προγραμματισμού από την πλακέτα. Αποσυνδέστε την τροφοδοσία, περιμένετε μερικά δευτερόλεπτα, και έπειτα συνδέστε πάλι την τροφοδοσία. Το FPGA πρέπει να φορτώσει αυτόματα τη σχεδίαση από την PROM. Για να ελέγξετε ότι λειτουργεί σωστά, εφαρμόστε τους 16 πιθανούς συνδυασμούς των τιμών των διακοπών SW0-SW3 παρατηρώντας το LD0. Το κύκλωμά σας

συμπεριφέρεται όπως αναμένετε; Εάν όχι, ζητήστε τη βοήθεια του καθηγητή. Εάν λειτουργεί σωστά, έχετε ολοκληρώσει με επιτυχία τον οδηγό εκμάθησης.



**Εικόνα 48: Παράμετροι προγραμματισμού της PROM**