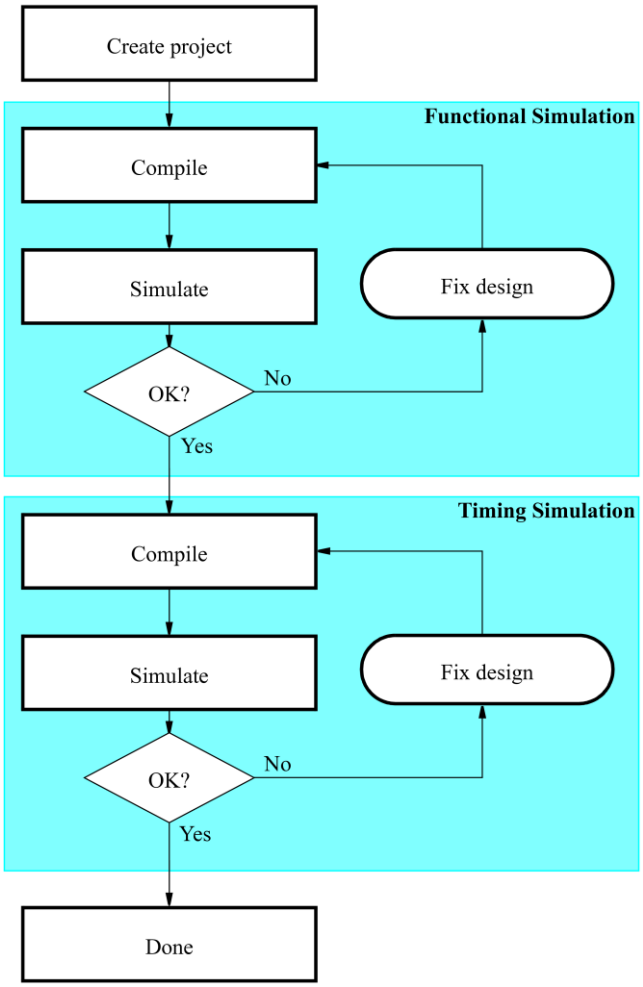
Προσομοίωση Ψηφιακών κυκλωμάτων με τον προσομοιωτή Modelsim

1 Προσομοίωση Ψηφιακών Κυκλωμάτων – Βασικές έννοιες

Οι σχεδιαστές ψηφιακών συστημάτων αναπόφευκτα αντιμετωπίζουν το έργο της επιβεβαίωσης της ορθής λειτουργίας των κυκλωμάτων τα οποία έχουν περιγράψει σε μία γλώσσα περιγραφής υλικού (Hardware Description Language, HDL), όπως η Verilog. Στη γενική περίπτωση, ένα ψηφιακό κύκλωμα μπορεί να αποτελείται από πολλές υπομονάδες (υπό-κυκλώματα), κάθε μία από τις οποίες πρέπει να ελέγχεται μεμονωμένα και στη συνέχεια να ενσωματώνεται στο συνολικό κύκλωμα.

Η επιβεβαίωση της σωστής λειτουργίας ενός κυκλώματος γίνεται μέσω της προσομοίωσης κατά την οποία εφαρμόζουμε τιμές στις εισόδους του κυκλώματος και παρατηρούμε τις τιμές των εξόδων του. Συγκεκριμένα, το αποτέλεσμα της προσομοίωσης είναι ένα σύνολο κυματομορφών που απεικονίζουν τη συμπεριφορά ή αλλιώς λειτουργία ενός κυκλώματος με βάση μία δεδομένη ακολουθία εισόδων.

Η γενική ροή της προσομοίωσης ενός κυκλώματος απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 1. Ροή προσομοίωσης

Όπως δείχνεται στο σχήμα, αρχικά αναπτύσσουμε ένα μοντέλο περιγραφής της λειτουργίας του κυκλώματος χρησιμοποιώντας μία γλώσσα HDL όπως η Verilog ή η VHDL. Στην

περίπτωση πολύπλοκων συστημάτων, η σχεδίαση καλείται σχεδιαστικό έργο ή απλούστερα έργο (project). Στη συνέχεια λαμβάνουν χώρα δύο είδη προσομοίωσης τα οποία είναι η λειτουργική προσομοίωση (functional simulation) και η προσομοίωση χρονισμού (timing simulation).

Η λειτουργική προσομοίωση ελέγχει τη λογική λειτουργία ενός κυκλώματος χωρίς να λαμβάνονται υπόψη οι καθυστερήσεις των πυλών και των διασυνδέσεων τους. Η προσομοίωση αυτή είναι αφενός γρήγορη και αφετέρου εξαιρετικά χρήσιμη αφού μας επιτρέπει να ελέγξουμε ότι πράγματι το HDL μοντέλο του κυκλώματος που έχουμε αναπτύξει υλοποιεί την επιθυμητή λογική λειτουργία. Σε αυτού του είδους την προσομοίωση, το κύριο χαρακτηριστικό είναι ότι η κυματομορφή της εξόδου μίας πύλης δεν παρουσιάζει καθυστέρηση ως προς τις κυματομορφές των εισόδων της. Με άλλα λόγια, οι μεταβολές των εξόδων δεν παρουσιάζουν καθυστέρηση ως προς τις μεταβολές των εισόδων της πύλης.

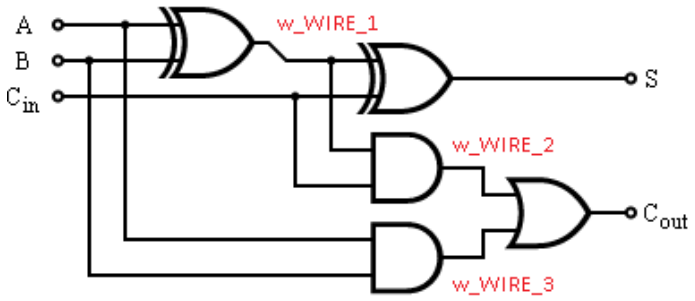
Το δεύτερο βήμα είναι η προσομοίωση χρονισμού. Πρόκειται για πιο περίπλοκο τύπο προσομοίωσης όπου το λογισμικό προσομοίωσης ή προσομοιωτής (simulator) λαμβάνει υπόψη του τις καθυστερήσεις των πυλών και διασυνδέσεων (καλωδίων) του κυκλώματος. Για το λόγο αυτό, το κύριο χαρακτηριστικό αυτού του είδους προσομοίωσης είναι ότι η κυματομορφή της εξόδου μίας πύλης παρουσιάζει καθυστέρηση ως προς τις κυματομορφές των εισόδων της. Είναι προφανές ότι αυτός ο τύπος προσομοίωσης είναι πιο ρεαλιστικός από τη λειτουργική προσομοίωση. Ωστόσο, απαιτείται περισσότερος χρόνος για την εκτέλεση της προσομοίωσης.

Συνοψίζοντας, αρχικά εκτελούμε λειτουργική προσομοίωση ώστε να επιβεβαιώσουμε ότι την ορθότητα του σχεδιασμού μας σε επίπεδο λογικής λειτουργίας και κατόπιν εκτελούμε προσομοίωση χρονισμού ώστε να επιβεβαιώσουμε ότι η καθυστέρηση του κυκλώματος είναι η επιθυμητή.

Στη συνέχεια ακολουθεί ένα παράδειγμα όπου παρουσιάζεται αναλυτικά ο τρόπος διεξαγωγής της λειτουργικής προσομοίωσης για το κύκλωμα του πλήρη αθροιστή του1 bit.

2 Κύκλωμα του πλήρη αθροιστή 1 bit και Verilog μοντέλο

Όπως γνωρίζεται ο πλήρης αθροιστή του 1 bit είναι ένα ψηφιακό κύκλωμα που έχει τρεις εισόδους A, B και Cin (κρατούμενο εισόδου) και δύο εξόδους τις S (άθροισμα) και Cout (κρατούμενο εξόδου). Η δε κυκλωματική του υλοποίηση σε επίπεδο πυλών είναι επίσης γνωστή και απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 2. Κύκλωμα πλήρη αθροιστή του 1 bit

Ένας κώδικας σε Verilog που περιγράφει τη λειτουργία του πλήρη αθροιστή του του 1 bit είναι ο ακόλουθος.

```
module full_adder (A, B, Cin, S, Cout);
input     A, B, Cin;
output    S, Cout;
wire      w_WIRE_1, w_WIRE_2, w_WIRE_3;

assign w_WIRE_1 = A ^ B;
assign w_WIRE_2 = w_WIRE_1 & Cin;
assign w_WIRE_3 = A & B;
assign S = w_WIRE_1 ^ Cin;
assign Cout = w_WIRE_2 | w_WIRE_3;
endmodule
```

Σχήμα 3. Κώδικας σε Verilog

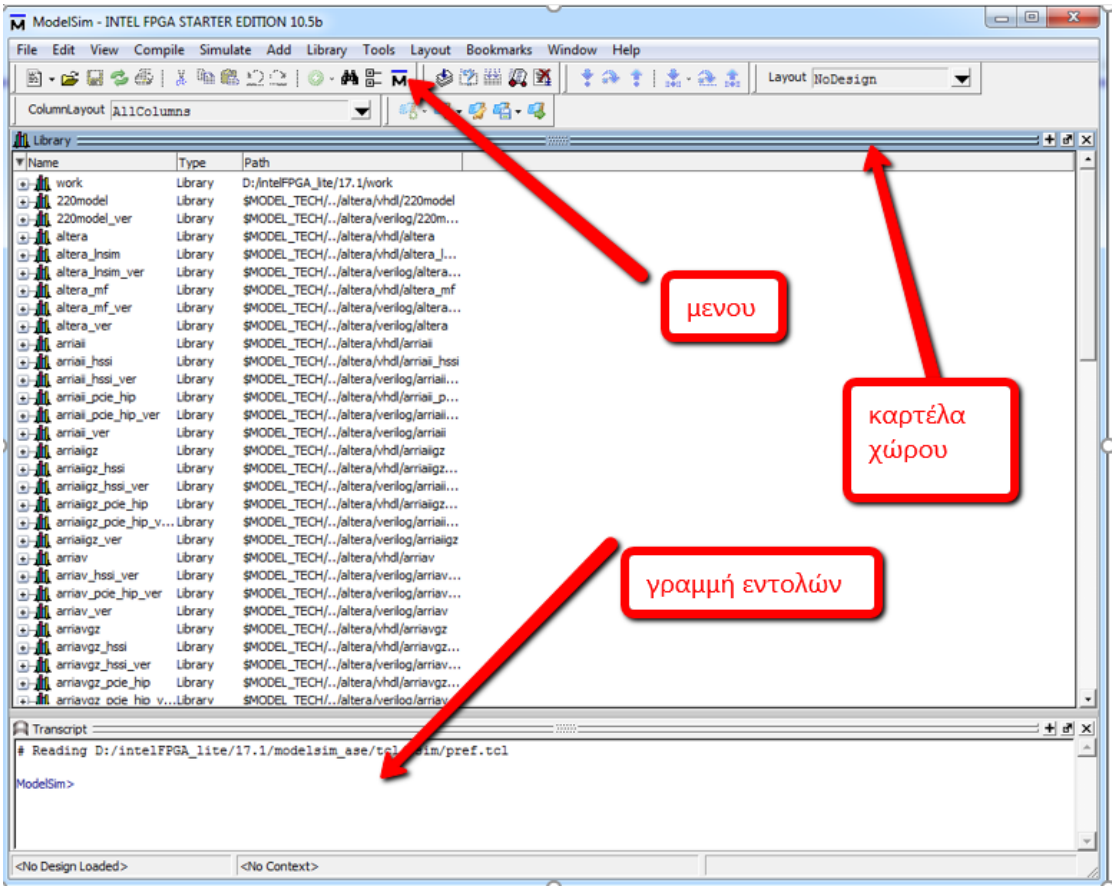
3 Λειτουργική προσομοίωση με το ModelSim

Ξεκινάμε δείχνοντας πώς να εκτελέσετε μια λειτουργική προσομοίωση για το παραπάνω κύκλωμα με τον προσομοιωτή ModelSim.

Το πρώτο βήμα είναι το άνοιγμα του προγράμματος ModelSim. Το παράθυρο προγράμματος ModelSim, που φαίνεται στο Σχήμα 4, αποτελείται από τρία τμήματα: **το κύριο μενού στην κορυφή**, ένα **σύνολο από τις καρτέλες χώρου εργασίας** και **μια γραμμή εντολών στο κάτω μέρος**.

Το μενού χρησιμοποιείται για πρόσβαση σε λειτουργίες που είναι διαθέσιμες στο ModelSim. Ο χώρος εργασίας περιέχει μια λίστα με βιβλιοθήκες που είναι διαθέσιμες σε εσάς καθώς και λεπτομέρειες του έργου (project) που εργάζεστε. Μια νέα περιοχή εργασίας θα εμφανιστεί στα δεξιά των βιβλιοθηκών, όταν χρειάζεται για εμφάνιση των κυματομορφών ή / και αρχεία κειμένου. Τέλος, η γραμμή εντολών στο κάτω μέρος δείχνει την ανατροφοδότηση από το εργαλείο προσομοίωσης και επιτρέπει στους χρήστες να εισάγουν εντολές.

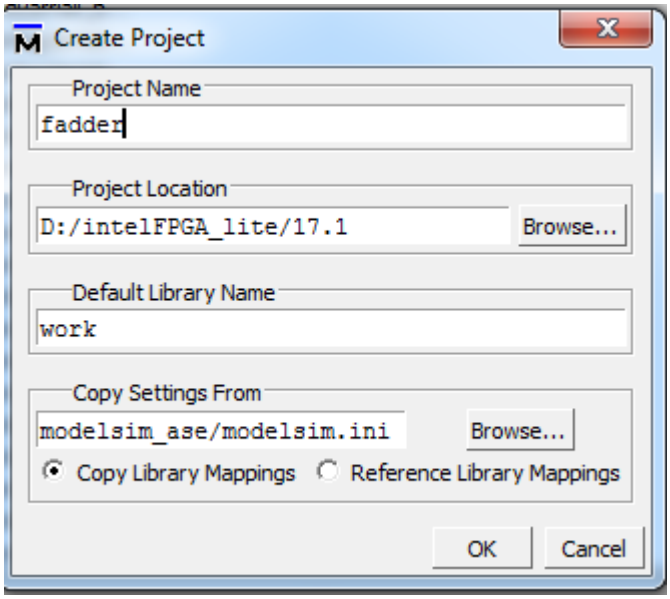
Για να εκτελέσουμε μία λειτουργική προσομοίωση με το ModelSim θα ακολουθήσουμε τη βασική ροή που φαίνεται στο Σχήμα 1. Ξεκινάμε δημιουργώντας ένα έργο (project) στο οποίο θα περιλαμβάνονται όλα τα αρχεία (HDL μοντέλα) του σχεδιασμού τα οποία πρόκειται να προσομοιωθούν. Στη συνέχεια ορίζουμε τα αρχεία HDL και μετά εκτελούμε την προσομοίωση. Με βάση τα αποτελέσματα της προσομοίωσης, ο σχεδιασμός μπορεί να τροποποιηθεί μέχρι να ικανοποιήσει τις επιθυμητές λειτουργικές προδιαγραφές.



Σχήμα 4. ModelSim window

3.1 Δημιουργία Project

Για να δημιουργήσετε ένα project στο ModelSim, επιλέξτε **File > New > Project**. Θα εμφανιστεί ένα παράθυρο δημιουργία έργου (Create Project) όπως φαίνεται στο Σχήμα 5.



Σχήμα 5 Δημιουργία νέου project

Το παράθυρο δημιουργίας έργου αποτελείται από τα ακόλουθα πεδία: **project name**, **project location**, **default library name**, και **copy settings field**.

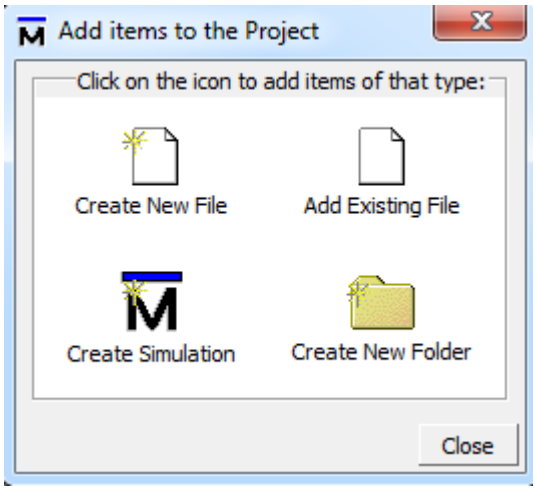
Το όνομα του έργου (project name) επιλέγεται από το χρήστη και η θέση του στο δίσκο (project location) είναι ο κατάλογος όπου βρίσκονται τα αρχεία προέλευσης. Για το παράδειγμά μας, επιλέγουμε το όνομα του έργου να είναι fadder, ώστε να ταιριάζει με το όνομα της μονάδας μας.

Το προκαθορισμένο πεδίο όνομα βιβλιοθήκης (default library name) καθορίζει ένα όνομα στο οποίο αποθηκεύινται τα αρχεία που δημιουργούνται κατά τη λειτουργία του προσομοιωτή ModelSim. Για το παράδειγμά μας δίνουμε το όνομα work.

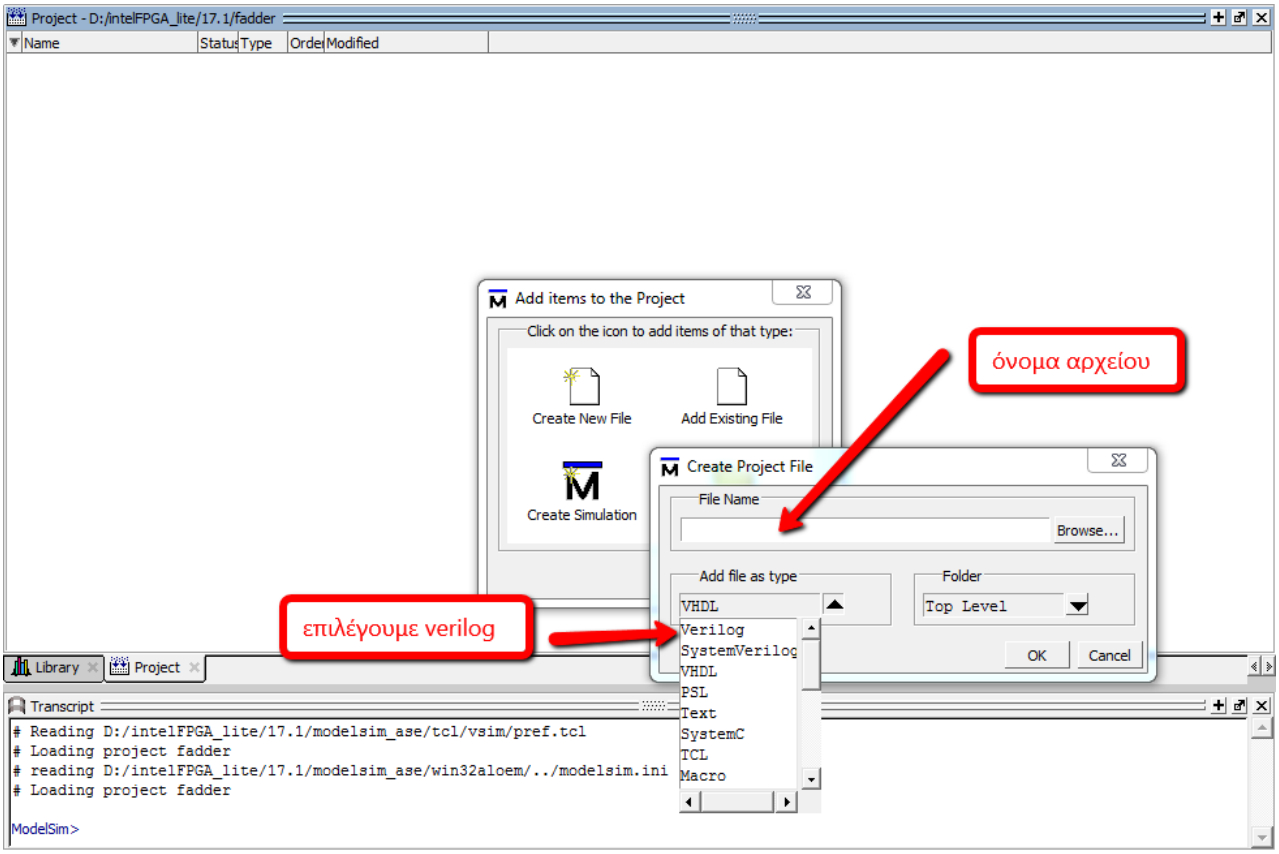
3.2 Εισαγωγή Αρχείων

Το παράθυρο στο Σχήμα 6 δίνει τις δυνατές επιλογές για την προσθήκη αρχείων στο έργο, συμπεριλαμβανομένης της δημιουργίας νέων αρχείων και καταλόγων, ή της ενσωμάτωσης υπαρχόντων αρχείων.

Εάν έχετε δημιουργήσει το αρχείο στο οποίο περιγράφεται σε HDL τη λειτουργία του κυκλώματος, κάντε κλικ στην επιλογή **Add Existing File** και επιλέξτε το συγκεκριμένο αρχείο. Αν όχι επιλέξετε **Create New File** και θα εμφανιστεί η καρτέλα όπως φαίνεται στο Σχήμα 7.

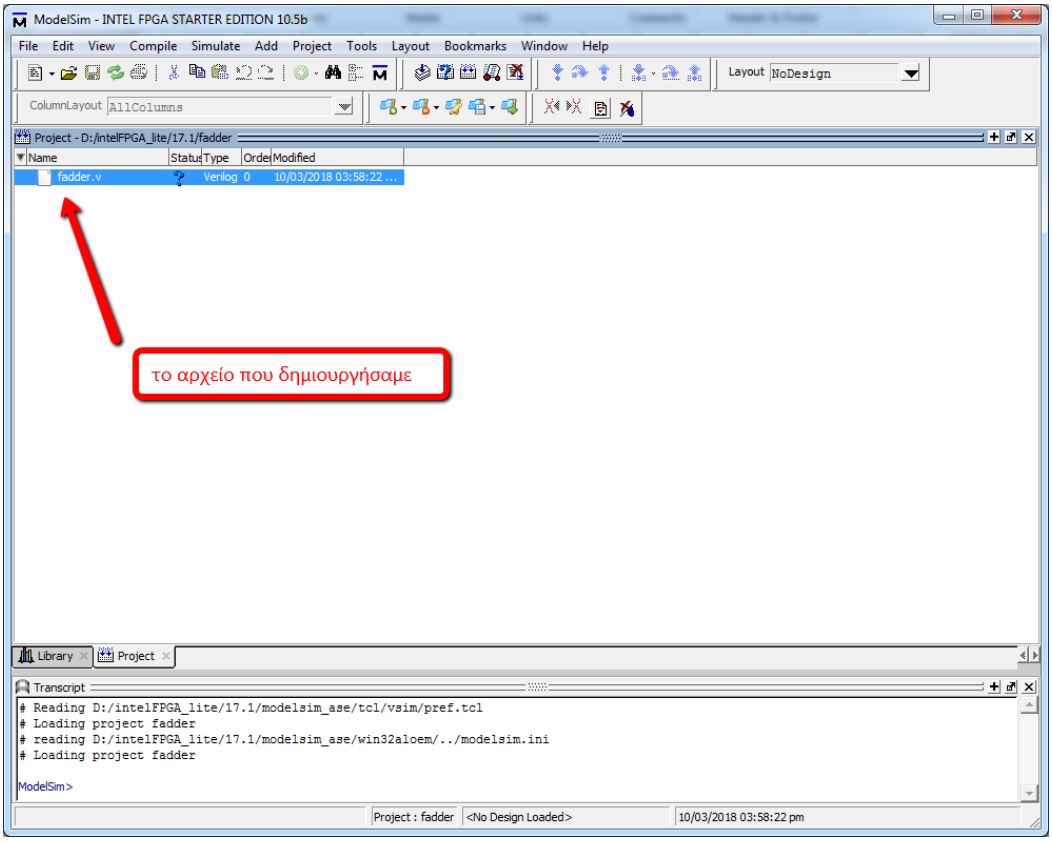


Σχήμα 6 Προσθήκη αρχείου



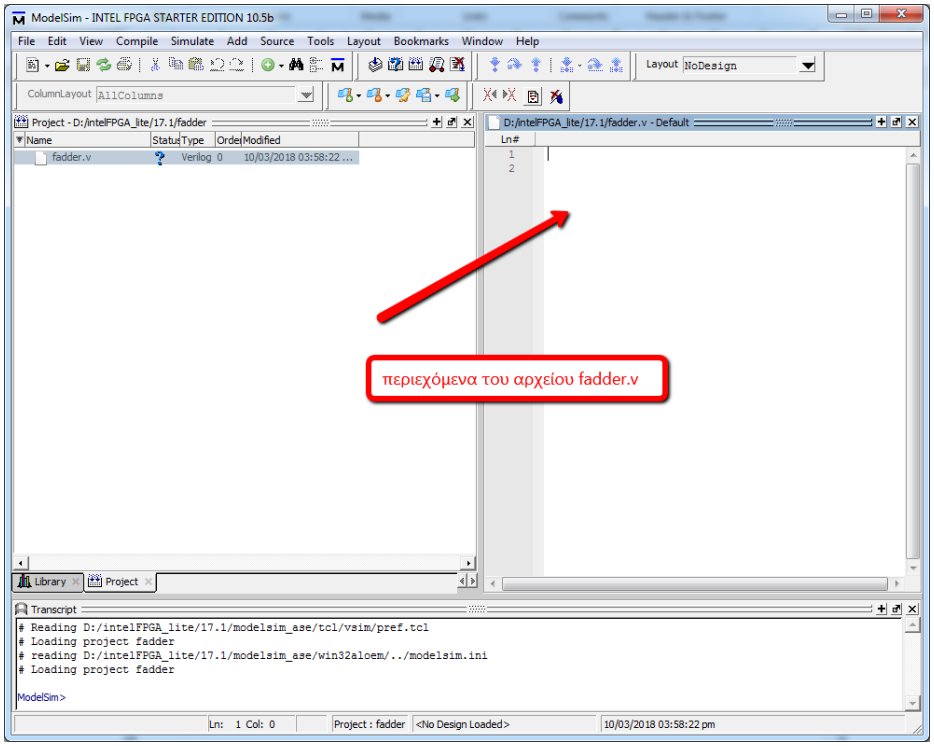
Σχήμα 7 Δημιουργία νέου αρχείου

Στη συνέχεια αφού δημιουργήσουμε το αρχείο μας fadder.v έχουμε την εικόνα που φαίνεται στο παρακάτω Σχήμα 8. Όλα τα αρχεία Verilog έχουν κατάληξη .v



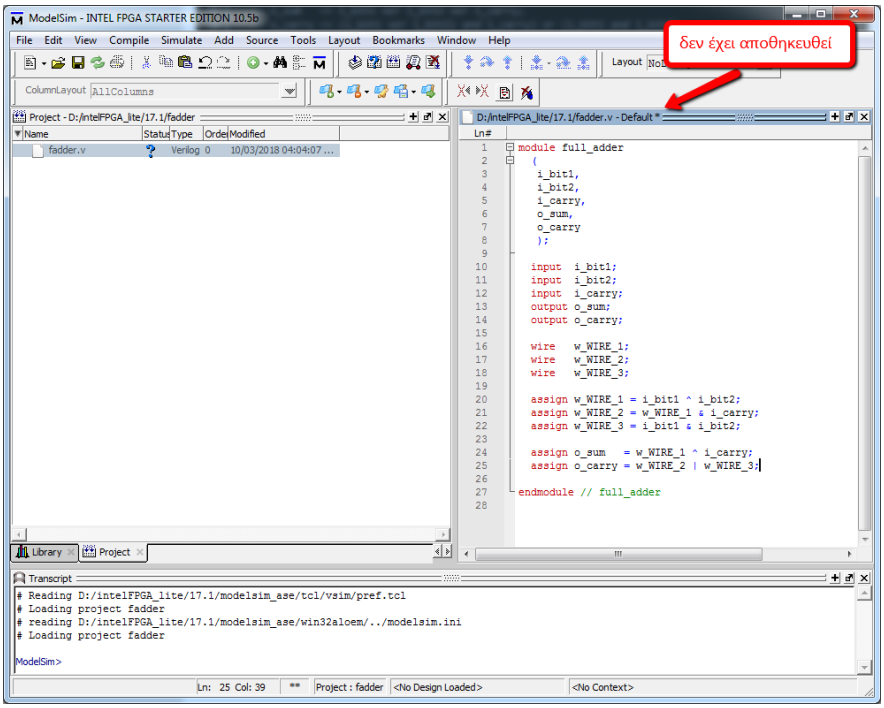
Σχήμα 8. Αρχείο fadder.v

Κάνουμε διπλό κλικ πάνω στο αρχείο fadder.v και εμφανίζεται η παρακάτω καρτέλα με το περιεχόμενο του αρχείου, όπως φαίνεται στο Σχήμα 9, το οποίο είναι προφανώς κενό αφού δεν έχουμε γράψει ακόμη το αντίστοιχο πρόγραμμα σε Verilog.



Σχήμα 9 Περιεχόμενο αρχείου

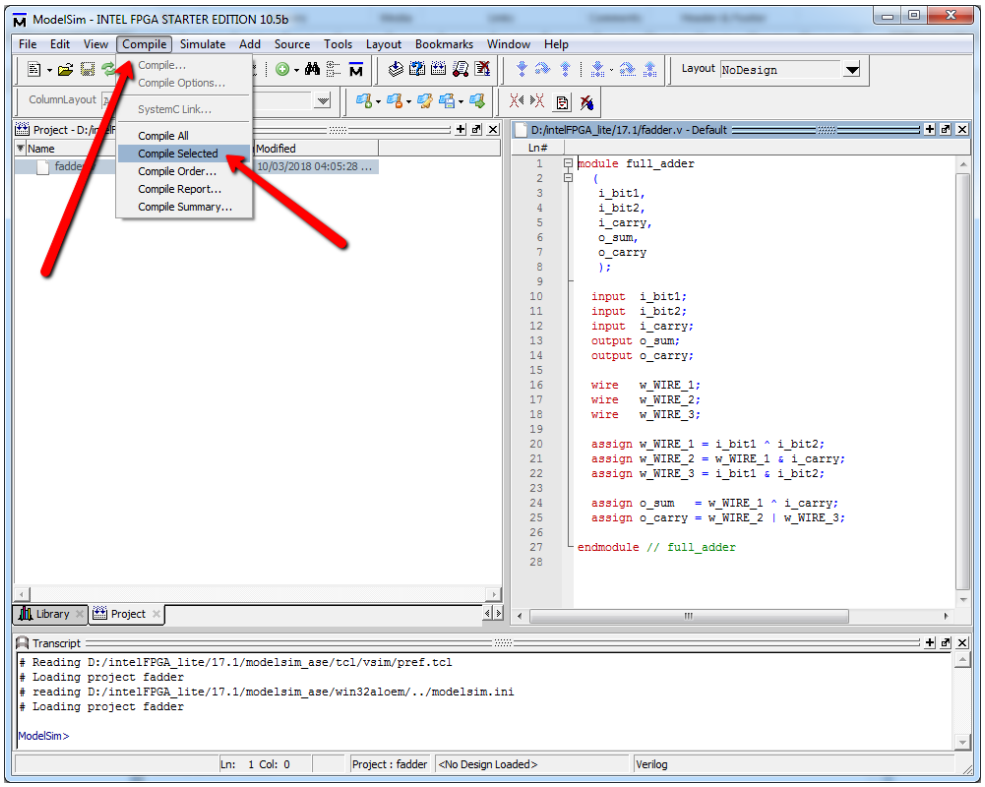
Κάνουμε επικόλληση τον κώδικα για τον πλήρη αθροιστή και αποθηκεύουμε. Εάν δεν αποθηκεύσουμε το αρχείο έχει πάνω στο όνομά του ένα αστερίσκο που δείχνει ότι δεν έχει γίνει αποθήκευση (βλ. Σχήμα 10)



Σχήμα 10. Αστερίσκος στο αρχείο

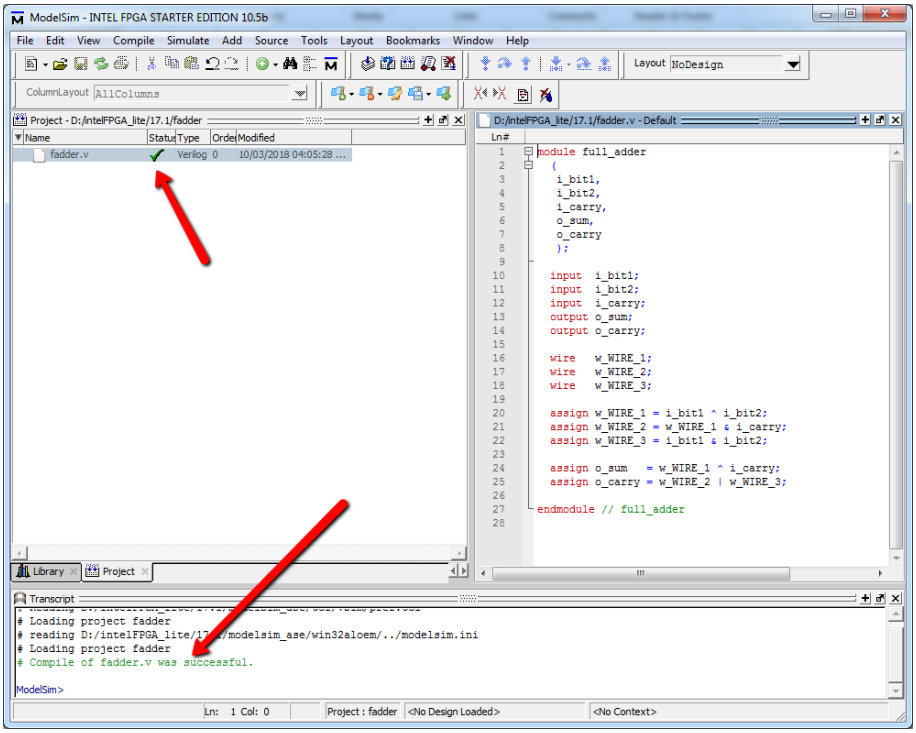
3.3 Μεταγλώττιση Project

Για να κάνουμε μεταγλώττιση (compile) στο αρχείο μας, επιλέγουμε από το μενού **Compile -> Compile selected** (βλ. Σχήμα 11)



Σχήμα 11. Μεταγλώττιση (Compile)

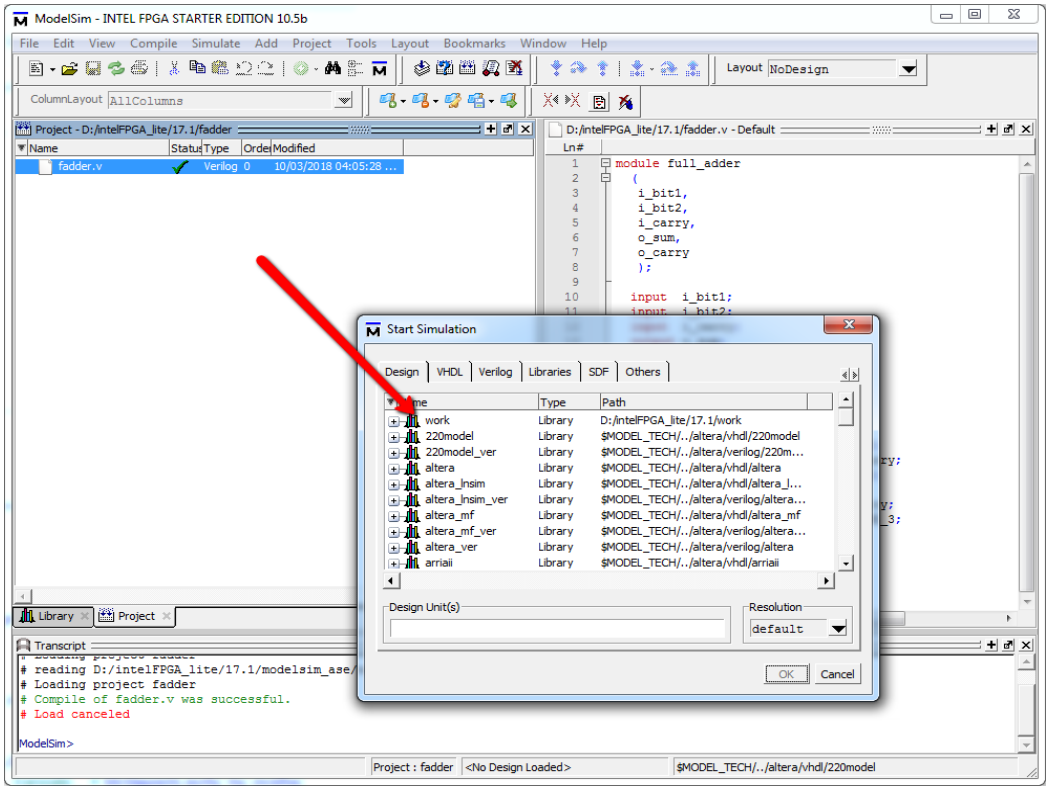
Αν όλα πάνε καλά και ο κώδικάς μας δεν έχει λάθη τότε θα έχουμε την εικόνα όπως φαίνεται παρακάτω Σχήμα 12.



Σχήμα 12 Αποτέλεσμα compile

3.4 Προσομοίωση

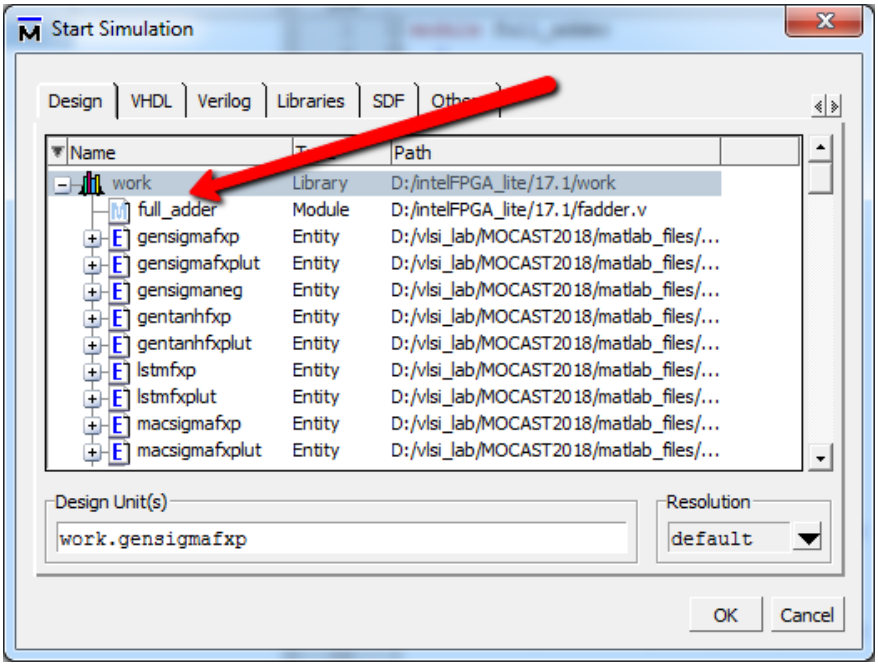
Για να ξεκινήσει μια προσομοίωση του σχεδιασμού, το λογισμικό πρέπει να τεθεί σε λειτουργία προσομοίωσης. Για να το κάνετε αυτό, επιλέξτε **Simulate > Start Simulation**. Θα εμφανιστεί το παράθυρο στο Σχήμα 13.



Σχήμα 13 Επιλογή simulation

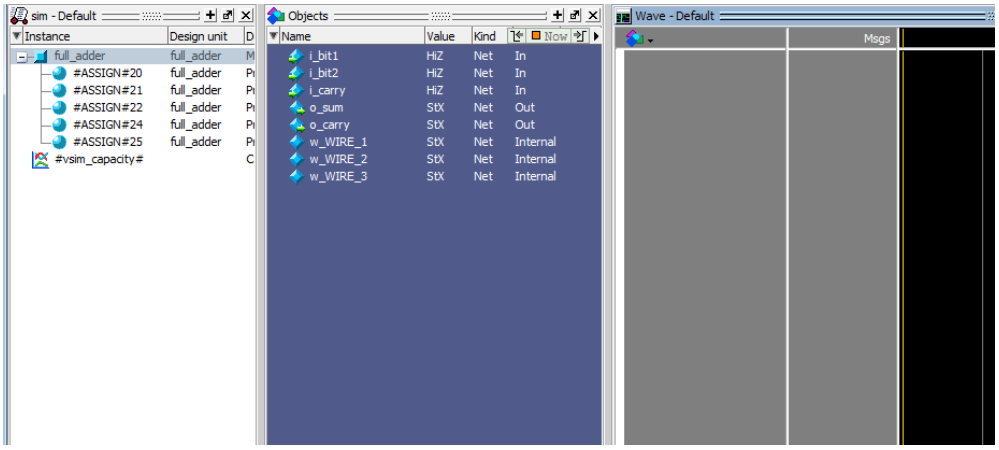
Το παράθυρο για την εκκίνηση προσομοίωσης αποτελείται από πολλές καρτέλες. Αυτές περιλαμβάνουν μια καρτέλα **Design** που παραθέτει τα σχέδια που είναι διαθέσιμα για προσομοίωσης, **VHDL** και **Verilog** για να καθορίσετε τις συγκεκριμένες επιλογές γλώσσας, μια καρτέλα **Libraries** για να συμπεριλάβετε τυχόν επιπλέον βιβλιοθήκες, χρονισμό και άλλες επιλογές στις υπόλοιπες δύο καρτέλες. Για τους σκοπούς της λειτουργικής προσομοίωσης, εμείς πρέπει μόνο να δούμε την καρτέλα **Design**.

Στην καρτέλα **Design** θα δείτε μια λίστα βιβλιοθηκών και ενοτήτων που μπορείτε να προσομοιώσετε. Θέλουμε να προσομοιώσουμε ένα κύκλωμα που ονομάζεται full_adder, το οποίο περιγράφεται στο αρχείο fadder.v. Για να επιλέξετε αυτήν την ενότητα, **εντοπίστε τη βιβλιοθήκη work και κάντε κλικ στο σύμβολο συν (+)**. Στη συνέχεια **επιλέξετε full_adder** και **κάντε κλικ στο κουμπί OK** για να ξεκινήσετε την προσομοίωση. (Σχήμα 14)



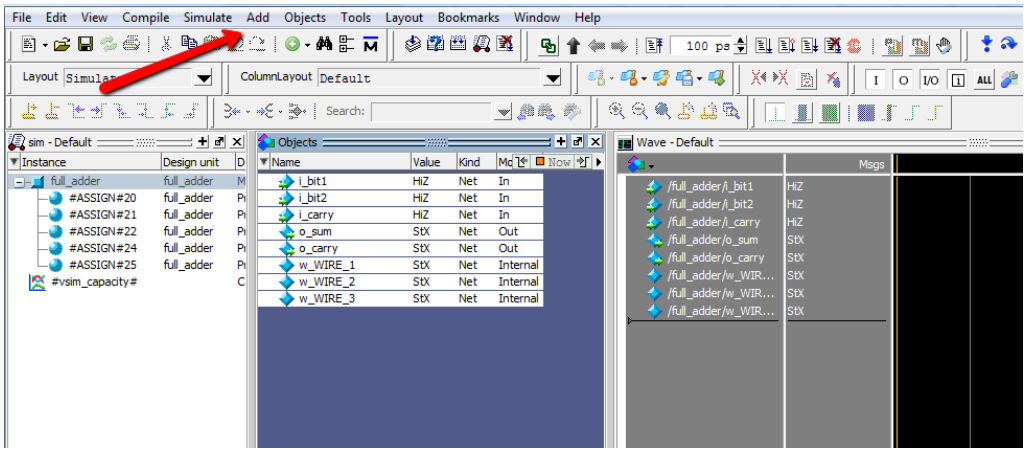
Σχήμα 14. Επιλογή του design

Στη συνέχεια εμφανίζεται το παρακάτω παράθυρο (Σχήμα 15).



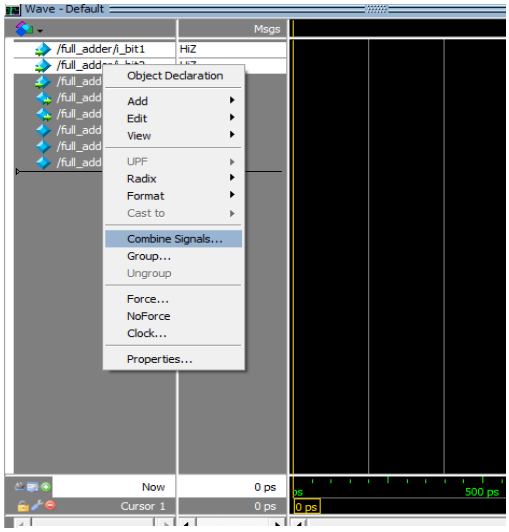
Σχήμα 15Παράθυρο προσομοίωσης

Στη συνέχεια από το παράθυρο Objects μπορούμε να επιλέξουμε ποια αντικείμενα (σήματα) θέλουμε να προσθέσουμε στο wave και μετά τα τοποθετούμε στο wave απλά με drag and drop. Εναλλακτικά μπορούμε να πάμε στο μενού **add > to wave > selected signals**.



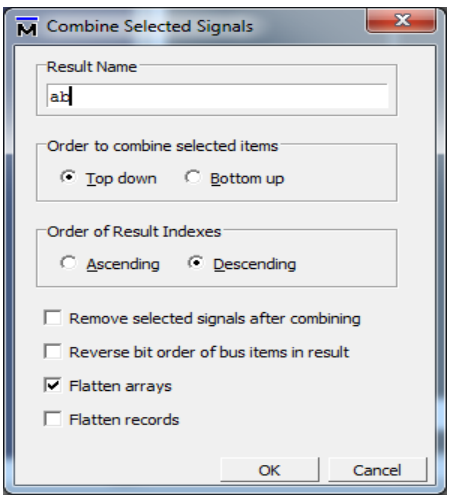
Σχήμα 16 Προσθήκη σημάτων στο παράθυρο wave

Μπορούμε να ομαδοποιήσουμε σήματα επιλέγοντάς τα και με δεξί κλικ επιλέγουμε Combine Sigmals (Σχήμα 17, Σχήμα 18,

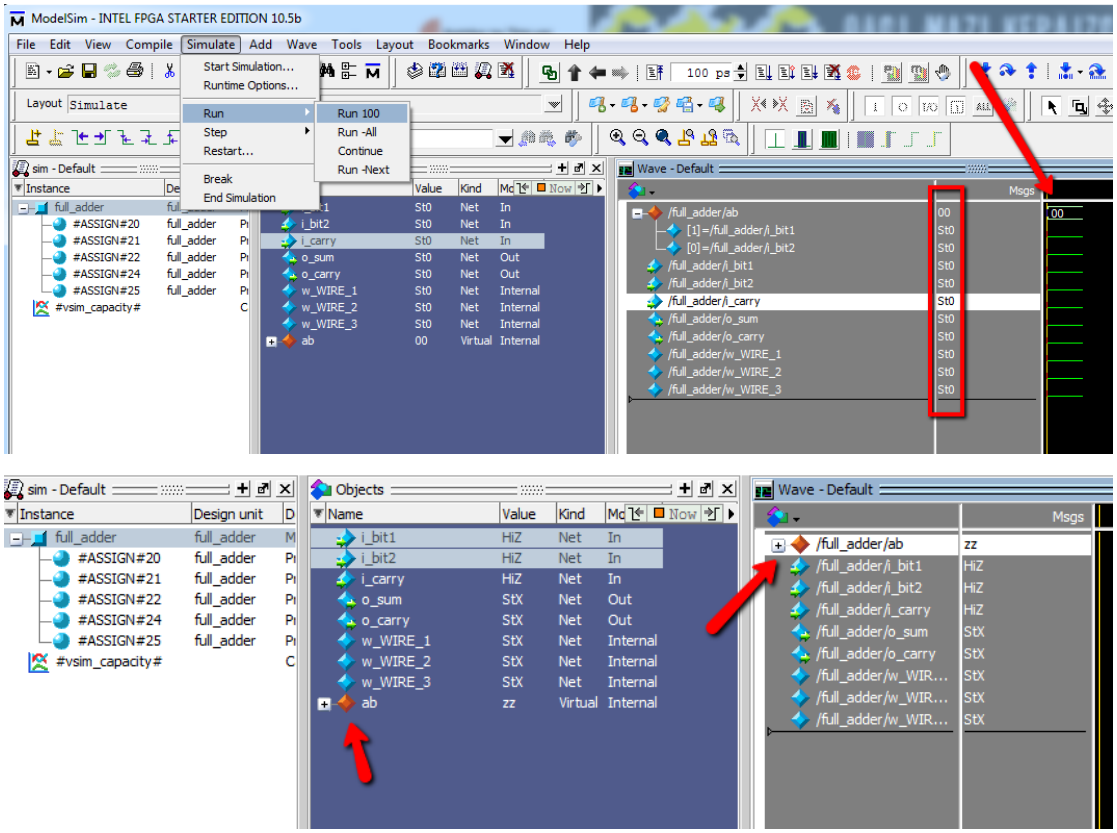


)

Σχήμα 17 Ομαδοποίηση σημάτων

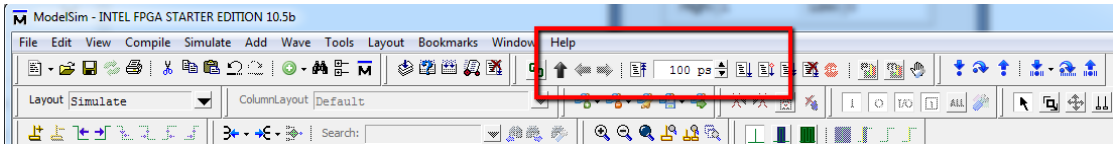


Σχήμα 18 Ομαδοποίηση σημάτων επιλογές



Σχήμα 19 Ομαδοποίηση σημάτων

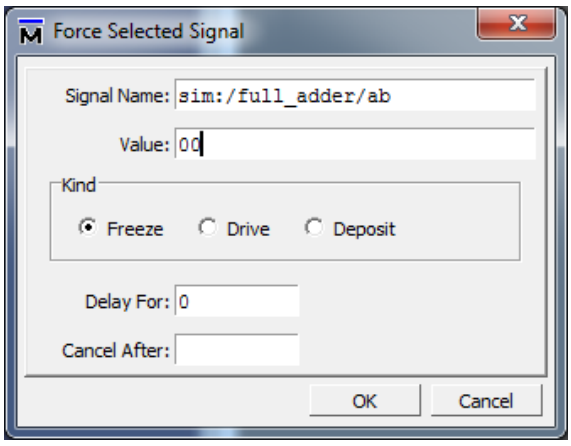
Στη συνέχεια για να ξεκινήσουμε την προσομοίωση επιλέγουμε από τα εργαλεία τη διάρκεια της προσομοίωσης (100ps)



Σχήμα 20.

Σχήμα 20. Διάρκεια προσομοίωσης

Κατόπιν κάνουμε δεξί κλικ πάνω στο σήμα στο οποίο θέλουμε να δώσουμε τιμές και εμφανίζεται το παρακάτω παράθυρο Σχήμα 21. Εδώ επιλέξαμε την ομάδα ab, οπότε θα πρέπει να δώσουμε 2-bit τιμή.

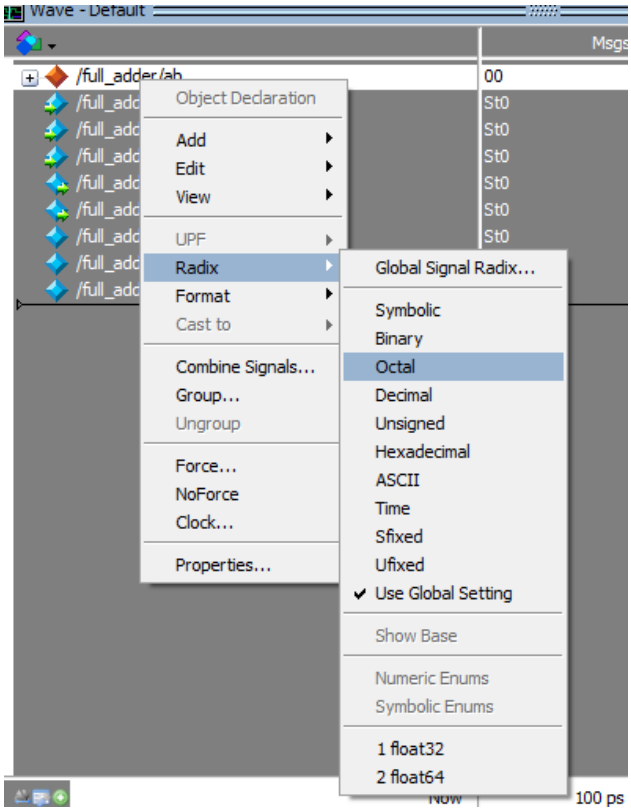


Σχήμα 21 Ανάθεση τιμών

Από το μενού Simulate > Run > Run 100 εκτελούμε προσομοίωση για 100ps (Σχήμα 22). Στο παράθυρο waves παρατηρούμε την εναλλαγή των σημάτων. Σημαντική παρατήρηση είναι ότι εδώ δεν έχουμε καθυστερήσεις στις πύλες.

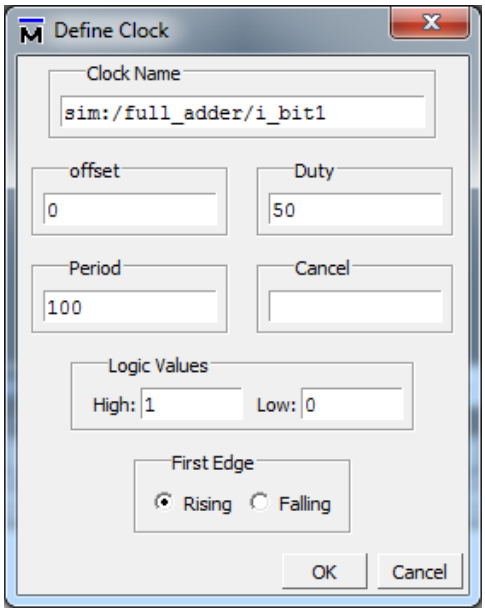
Σχήμα 22 Εκτέλεση προσομοίωσης

Αν θέλουμε τα σήματα να εμφανίζονται με διαφορετική μορφή από δυαδική, μπορούμε με δεξί κλικ να επιλέξουμε τη μορφή που θέλουμε. (Σχήμα 23)



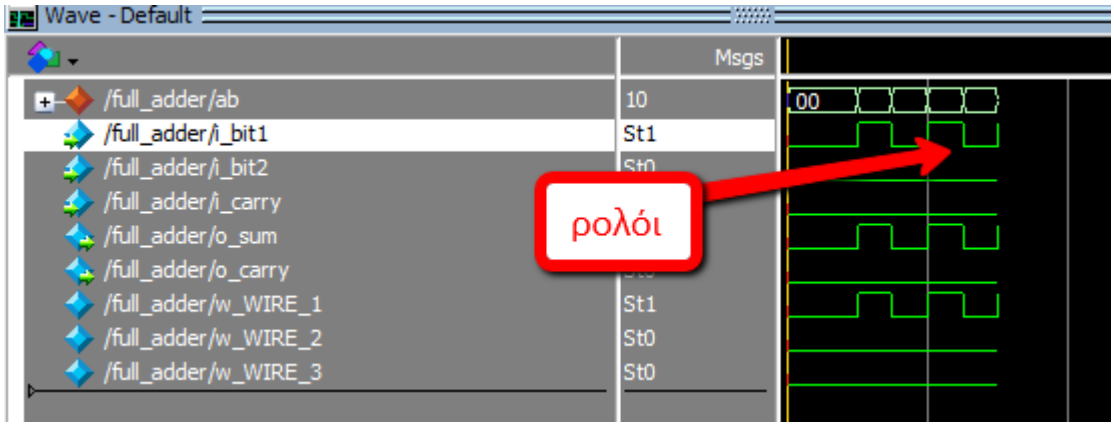
Σχήμα 23 Επιλογή μορφής αριθμών

Σε περίπτωση που το κύκλωμά μας περιέχει ρολόι τότε κάνουμε δεξί κλικ στο σήμα το οποίο θέλουμε να ορίσουμε ως ρολόι και στη συνέχεια επιλέγουμε clock κι εμφανίζεται το παρακάτω παράθυρο στο οποίο μπορούμε να ρυθμίσουμε το ρολόι μας.



Σχήμα 24 Ορισμός ρολογιού

Εκτελώντας ξανά προσομοίωση με την επιλογή από το μενού simulate > run > run 100 έχουμε για τα επόμενα 100ps το αποτέλεσμα στο παράθυρο wave.



Σχήμα 25 Εμφάνιση στο παράθυρο wave