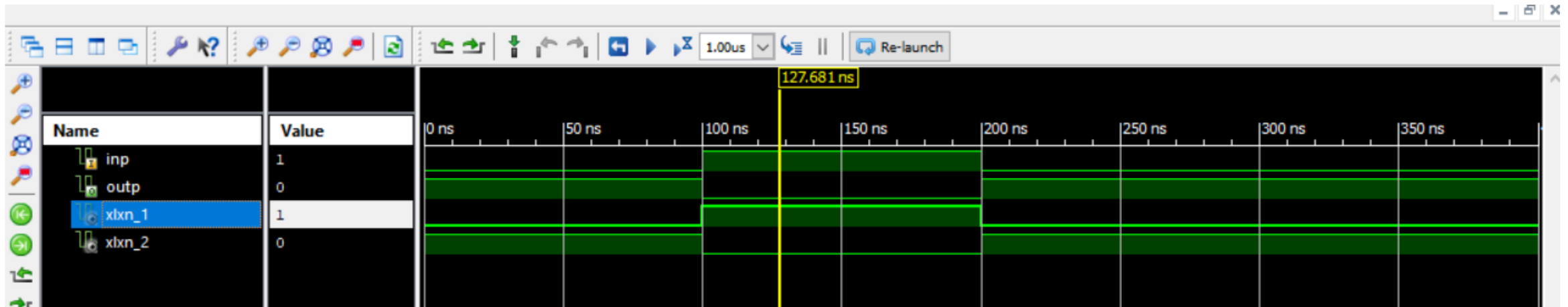


Στο προηγούμενο είχαμε δημιουργήσει ένα project (inverter.xise) στο ISE για την κάρτα του εργαστηρίου (Spartan 3 XC3S400 FT256 -4) σε ένα φάκελο (C:\Ergasies\Syndyastika\inverter). Προσθήσαμε το σχηματικό ενός αναστροφέα (inverter.sch) και επιθεωρήσαμε το functional model (inverter.vhf). Προχωρήσαμε σε λειτουργική προσομοίωση του κυκλώματος και αποθηκεύσαμε το αποτέλεσμα (inverter.wcfg) όπως και συνολικά το project. Όλα τα παραπάνω αρχεία βρίσκονται στο φάκελο του project και μπορούμε να τα ανοίξουμε με το κατάλληλο εκτελέσιμο Xilinx πρόγραμμα (που βρίσκονται στην τοποθεσία της εγκατάστασης, πχ στην C:\Xilinx\14.7\ISE_DS\ISE\bin\nt64). Για παράδειγμα τρέχοντας το isimgui.exe μπορούμε να ανοίξουμε το inverter.wcfg...



inp, outp αντιστοιχούν στα IO markers και xlxn_1, xlxn_2 στα IO buffers που εισήγαμε και είναι απαραίτητα για μετέπειτα διαδικασίες (πχ synthesis)

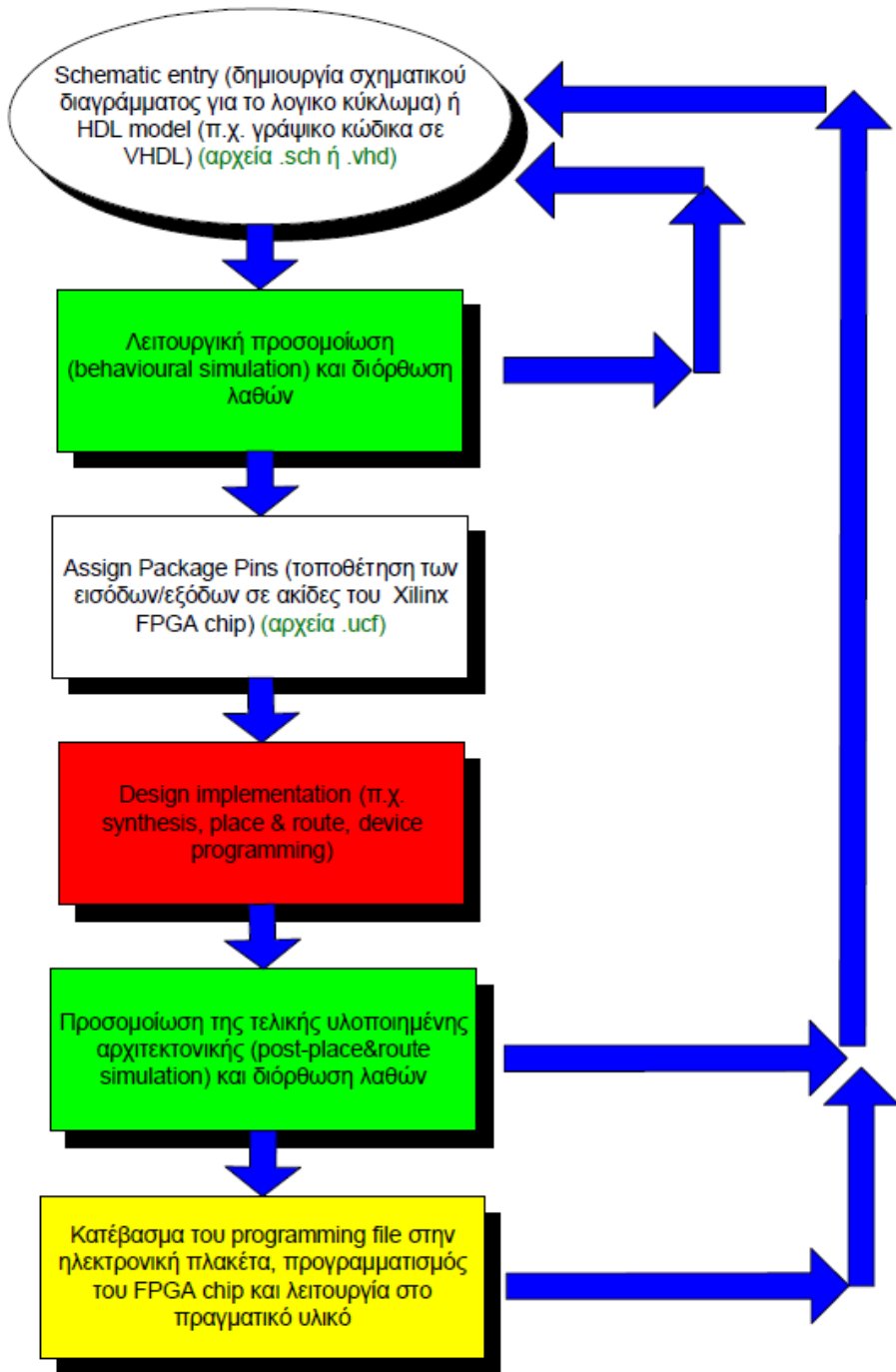
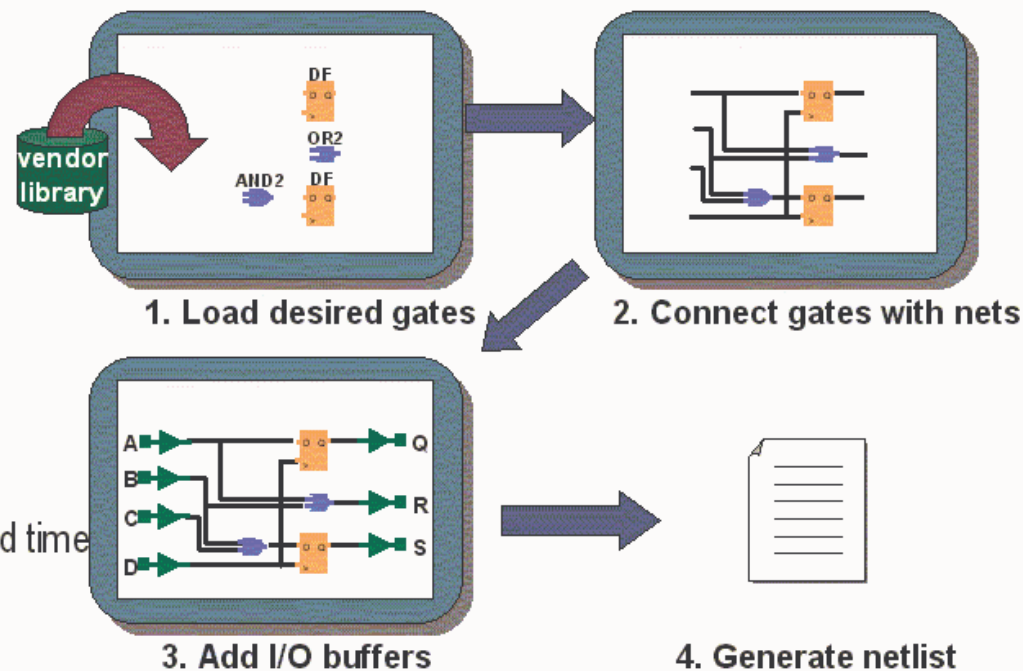
Με διπλό κλικ στο inverter.xise εκκινούμε το αποθηκευμένο project...

PLD Design flow - Schematic Capture

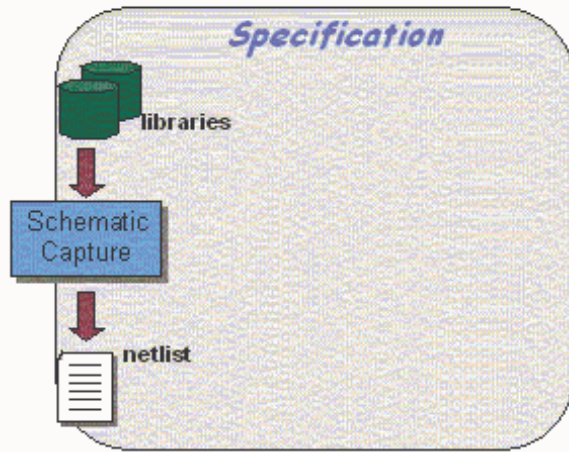
Defn: A software program that allows designers to graphically describe a circuit.

- Design flow is identical to standard logic design except I/O buffers are defined - consider the design within the PLD as a mini PCB!

- However this PCB can be changed time and time again quickly and easily

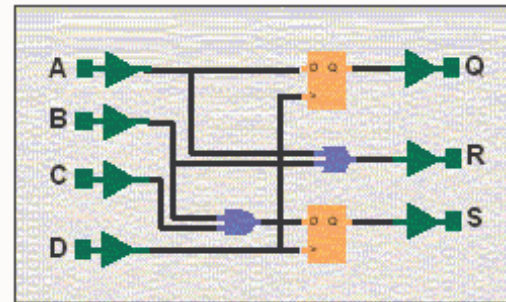


Design Flow



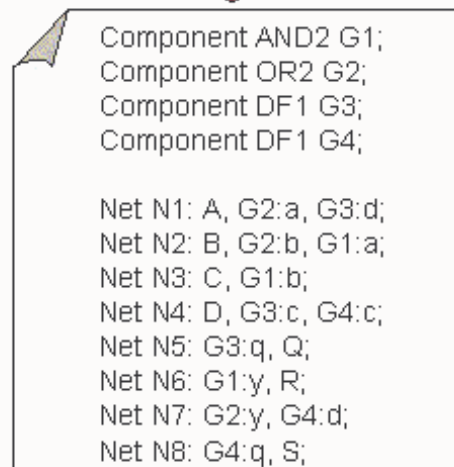
Design Specification Netlist

Defn: A gate-level text file representation of a circuit.



design schematic

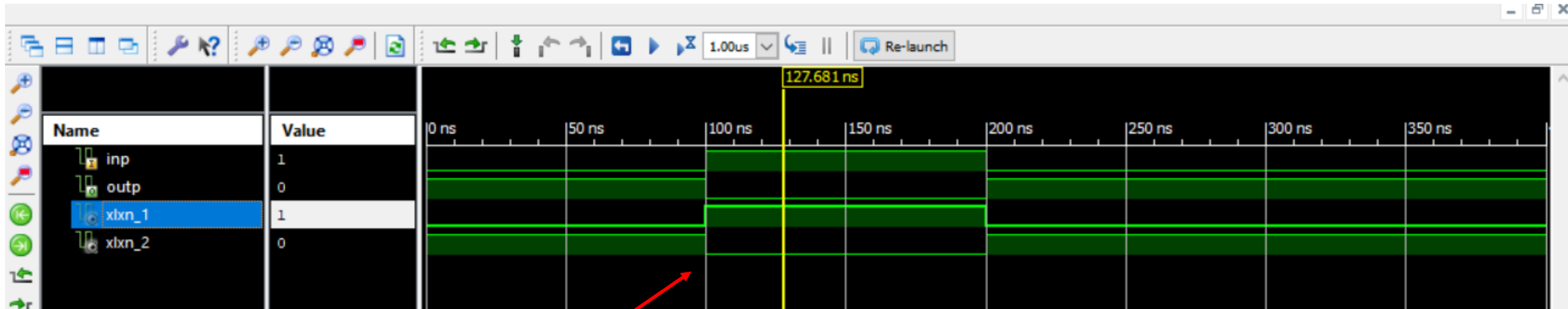
- Graphical description
- Created by humans
- For reading by humans



design netlist

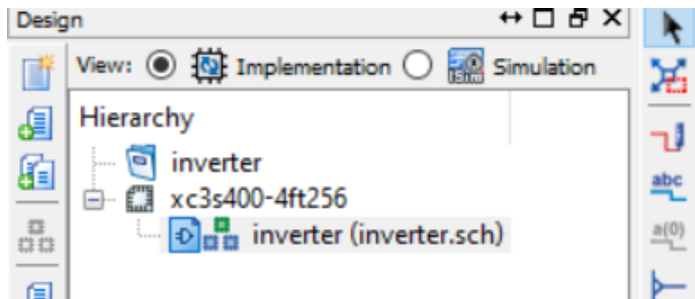
- Text file
- Output of schematic capture
- PC friendly
- EDIF is industry standard
(Electronic Digital Interchange Format)
- XNF is Xilinx Netlist Format

12. Θα παρατηρήσετε ότι στην περίπτωση της λειτουργικής προσομοίωσης οι έξοδες των συνδυαστικών κυκλωμάτων αλλάζουν τιμή (αν αυτό απαιτείται από την λειτουργία τους) **στιγμιαία**, χωρίς καθυστέρηση και ακριβώς στον ίδιο χρόνο που αλλάζουν τιμές και οι είσοδες του κυκλώματος. Αυτό συμβαίνει διότι στην λειτουργική προσομοίωση εξετάζουμε μόνον την σωστή λειτουργία του κυκλώματος και όχι τις καθυστερήσεις που παρουσιάζονται στα πραγματικά κυκλώματα. Αυτή η στιγμιαία απόκριση των εξόδων δεν συμβαίνει όμως στην post-route (μετά-την-υλοποίηση) προσομοίωση, διότι η τελευταία περιέχει πληροφορία και για την πραγματική καθυστέρηση των υλικών του κυκλώματος, την οποία θα εξετάσετε σε επόμενα βήματα των ασκήσεων.

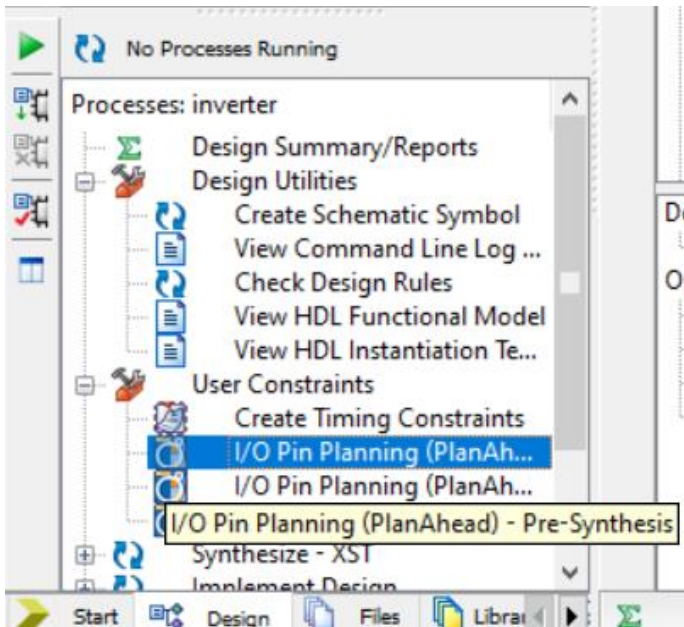


V. Υλοποίηση του σχεδίου (σχηματικό ή VHDL) με το εργαλείο Xilinx ISE

1. Αφού τελειώσετε και ελέγξετε το σχέδιο (schematic design) σας, είτε είναι σχηματικό, είτε είναι σε γλώσσα VHDL, επιλέξτε (αριστερό κλικ του ποντικιού) το παράθυρο Design, αμέσως μετά κάντε απλό κλικ (επιλέξτε) το σχέδιό σας <όνομα σχεδίου>.vhd ή <όνομα σχεδίου>.sch. Μετά επιλέξτε στο View: την επιλογή **Implementation**. Τότε θα δείτε ακριβώς από κάτω και μέσα στο παράθυρο Processes να εμφανίζονται τα διάφορα εργαλεία για την υλοποίηση.

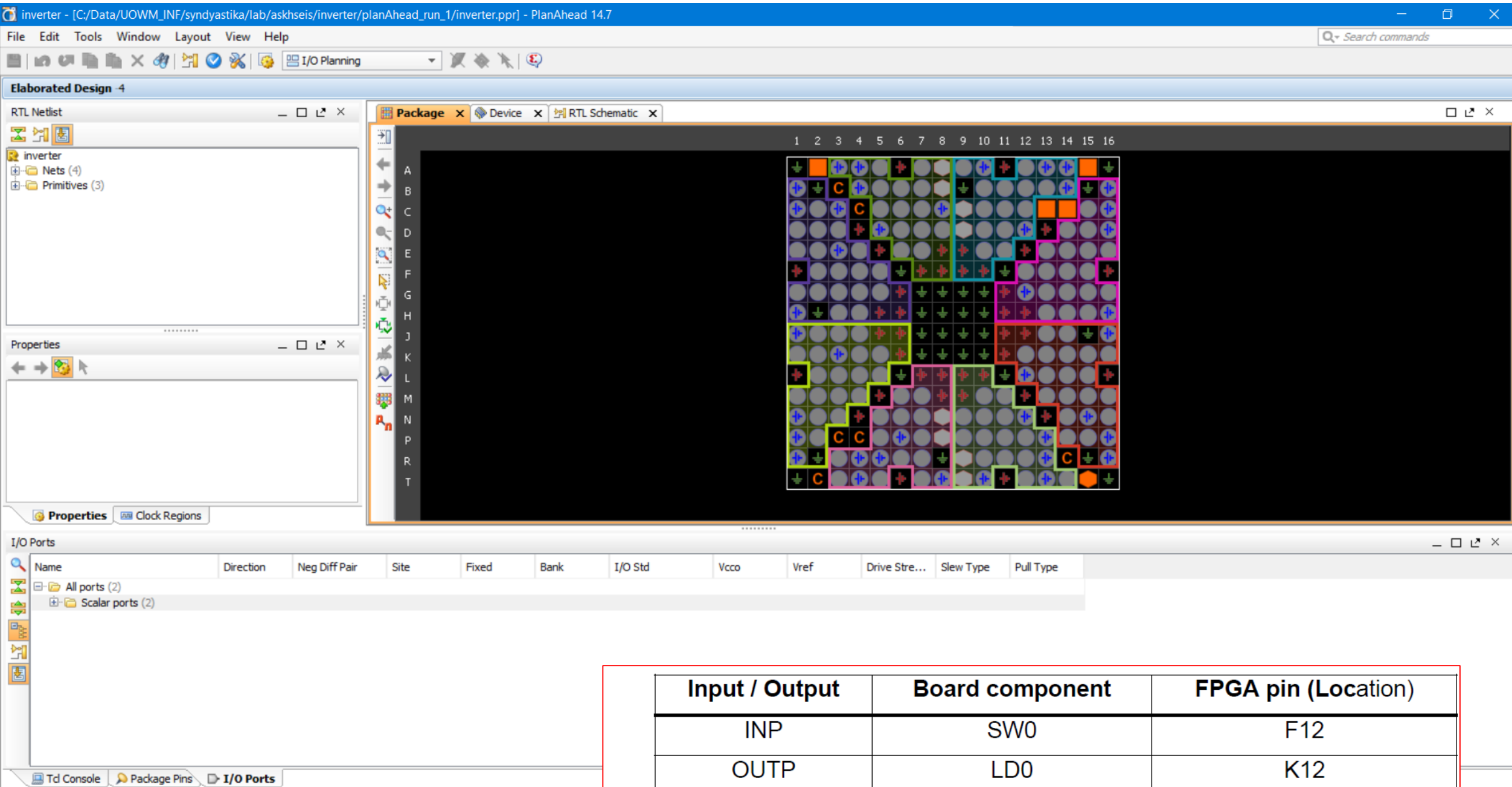


2. Από την λίστα των προγραμμάτων, ανοίξτε την ομάδα **User Constraints** και από τα υπο-εργαλεία της κάνετε διπλό κλικ (εκτελέστε) στο εργαλείο **I/O Pin Planning**. Αμέσως μετά το εργαλείο θα σας ρωτήσει αν θέλετε να προσθέσει ένα αρχείο περιορισμών .UCF στο έργο (project) σας. Απαντήστε με κλικ στην επιλογή Yes. Η τελευταία ενέργειά σας θα έχει σαν αποτέλεσμα να ανοίξει ένα καινούργιο παράθυρο που ονομάζεται **PlanAhead**. Εκεί κάντε κλικ στο OK ή close αν σας ρωτήσει. Επιλέξτε το υπο-παράθυρο Package στα δεξιά. Πηγαίνετε τώρα στο κείμενο της άσκησης παρακάτω, και διαβάστε τα locations των εισόδων και εξόδων. Στην στήλη **Loc** βρείτε τις τοποθεσίες (locations) των FPGA pins για τις E/E του κυκλώματός σας, όπως σας ζητείται στην κάθε εργαστηριακή άσκηση. Πηγαίνετε στο κάτω υπο-παράθυρο του εργαλείου PlanAhead, και επιλέξτε το tab **Package Pins**. Τώρα πάτε πίσω στο παράθυρο Package και με το ποντίκι επιλέξτε ένα από τα locations που έχετε στην άσκηση, και μετά πηγαίνετε στο Package Pins και γράψτε στην στήλη Port το όνομα της εξόδου ή της εισόδου. Επαναλάβετε αυτή την διαδικασία έως ότου τοποθετήσετε όλες τις Εισόδους/Εξόδους. Αν στο υπο-παράθυρο αυτό σας εμφανίσει άσχετα ονόματα E/E τα οποία δεν υπάρχουν στο σχέδιό σας, ή δεν υπάρχει η λίστα με τα locations τότε απλά βάλτε όσα locations έχετε ή όσα χωράνε στην στήλη Loc και αφού τελειώσετε και βγείτε από το εργαλείο, μπορείτε να ανοίξετε το αρχείο <όνομα_σχεδίου>.ucf με κάποιον κειμενογράφο των Windows και να διορθώσετε εκεί με το χέρι τις τιμές ή/και τα ονόματα που είναι διαφορετικά ή που λείπουν από τα FPGA locations. Αφού τελειώσετε με τα locations, σώστε τις αλλαγές σας με το εικονίδιο Save και κλείστε το παράθυρο PlanAhead. Πριν κλείσει αυτό το εργαλείο σας



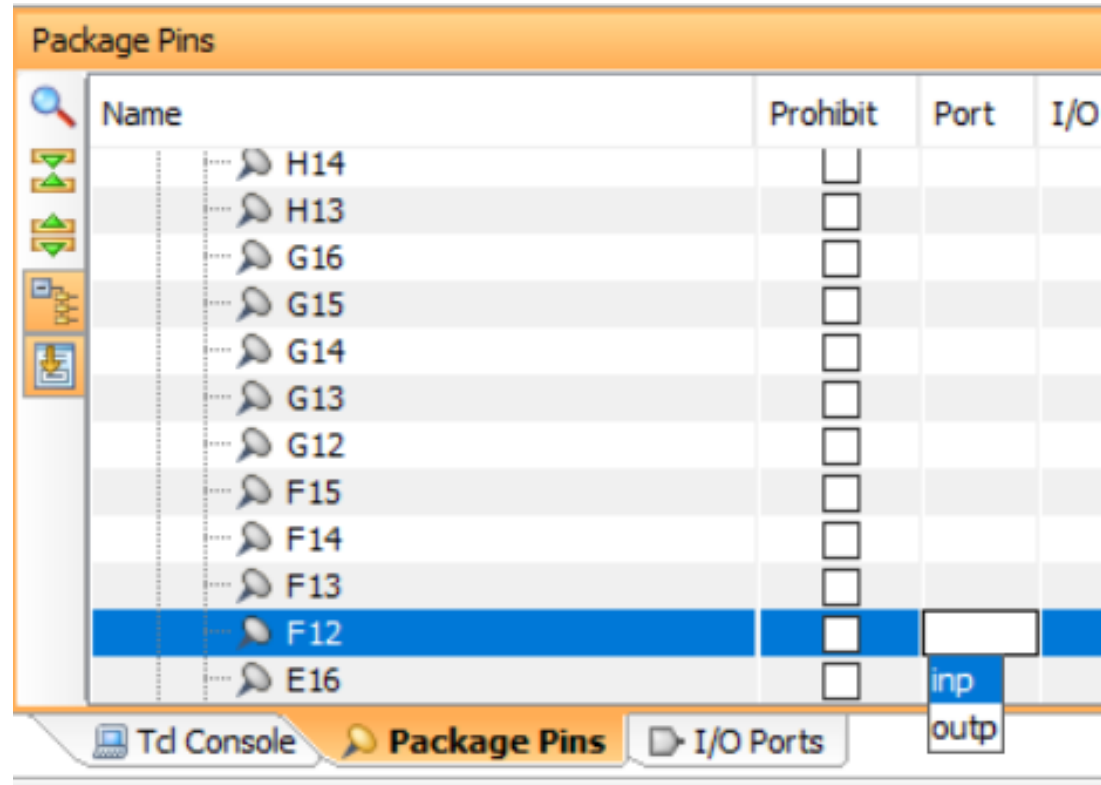
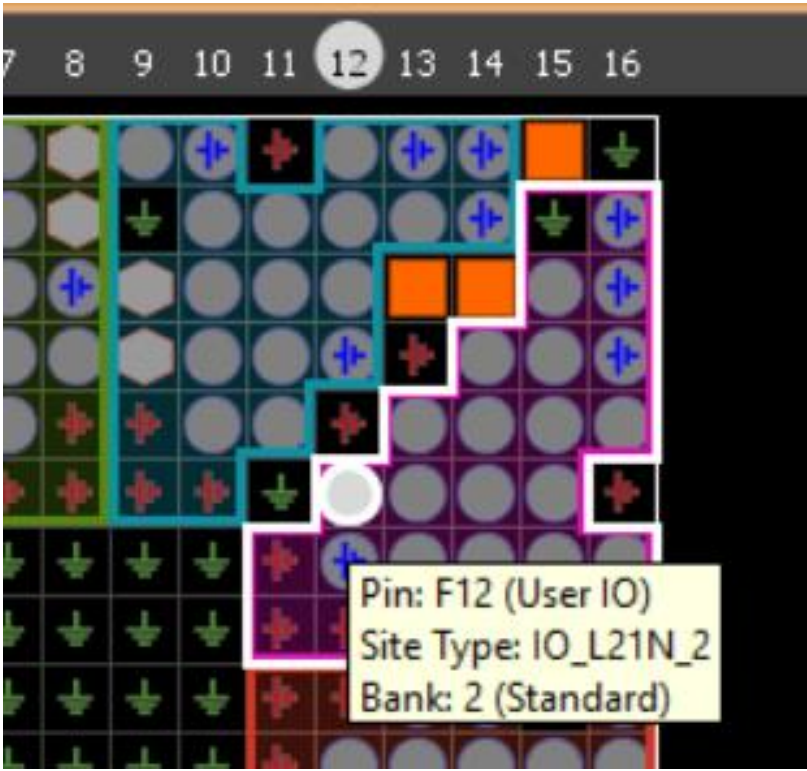
Input / Output	Board component	FPGA pin (Location)
INP	SW0	F12
OUTP	LD0	K12

Πίνακας 1: Τοποθετήσεις (locations) των E/E του κυκλώματος στο FPGA chip

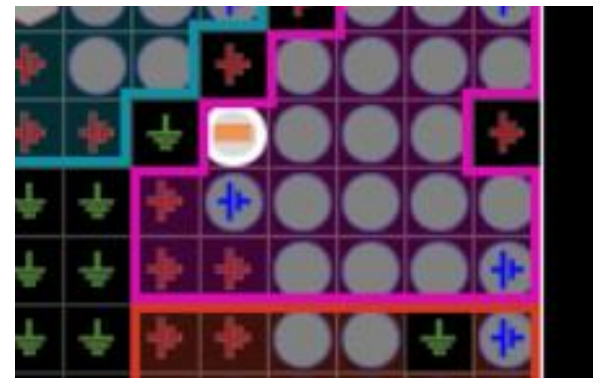
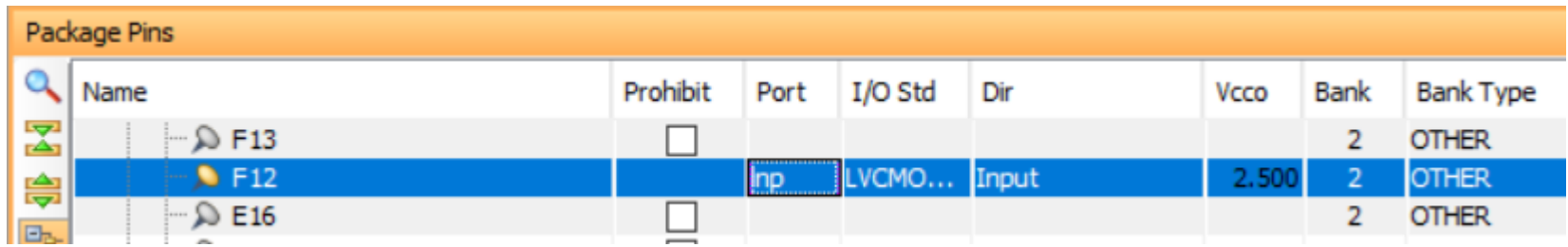


Input / Output	Board component	FPGA pin (Location)
INP	SW0	F12
OUTP	LD0	K12

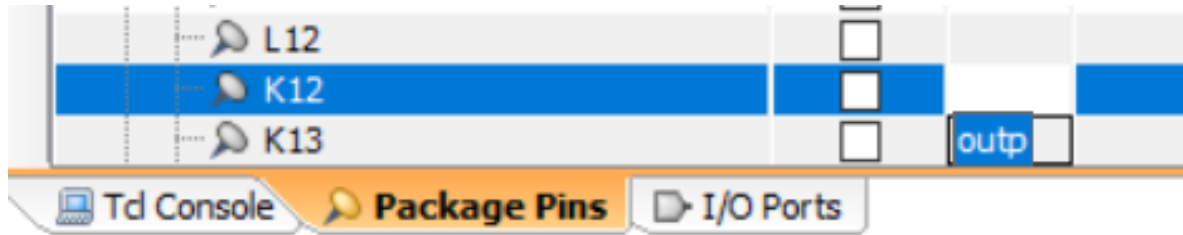
Πίνακας 1: Τοποθετήσεις (locations) των Ε/Ε του κυκλώματος στο FPGA chip



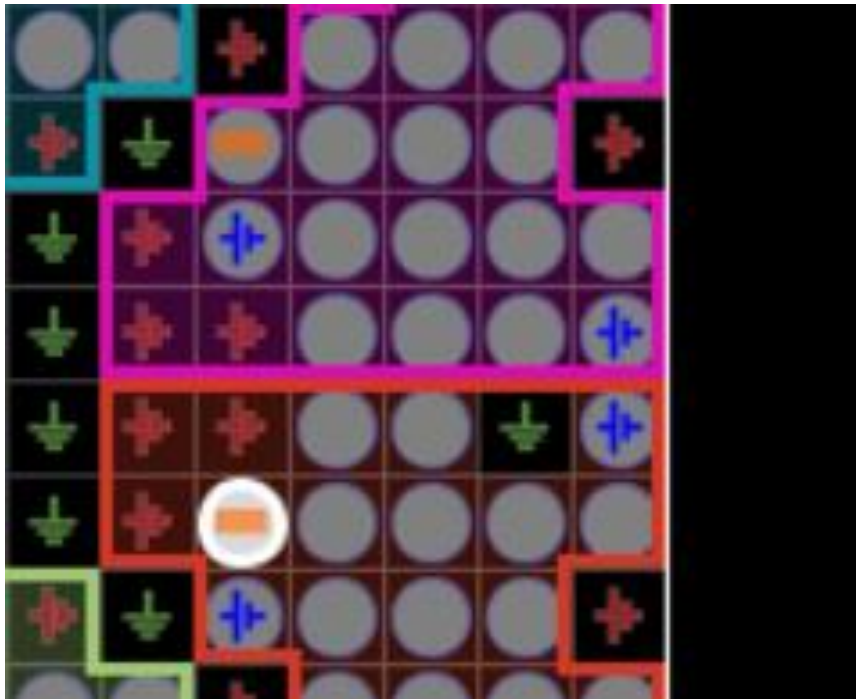
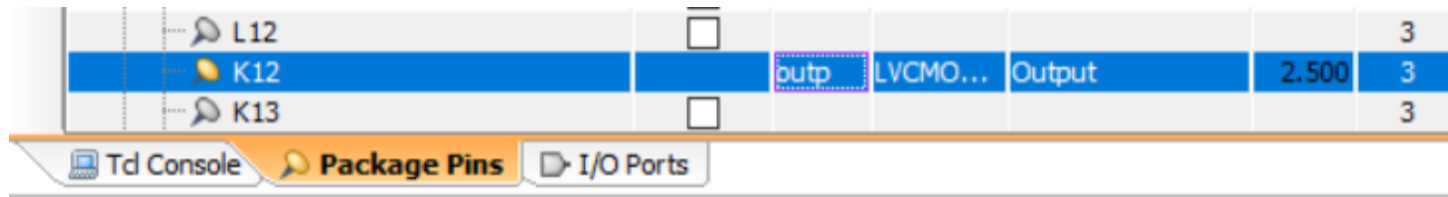
Single Click στο F12: επιλέγουμε inp και πατάμε enter. Η γραμμή στο Package Pins αλλάζει με τιμές (πχ Voltage κλπ) και το pin «δεσμεύεται» στο παράθυρο Package. Το pin αντιστοιχεί στο SW0 της συγκεκριμένης κάρτας (Spartan-3 XC3S400).



Ακολουθούμε την ίδια διαδικασία για όλα τα υπόλοιπα pins (εδώ έχουμε άλλο ένα, το outp → K12)

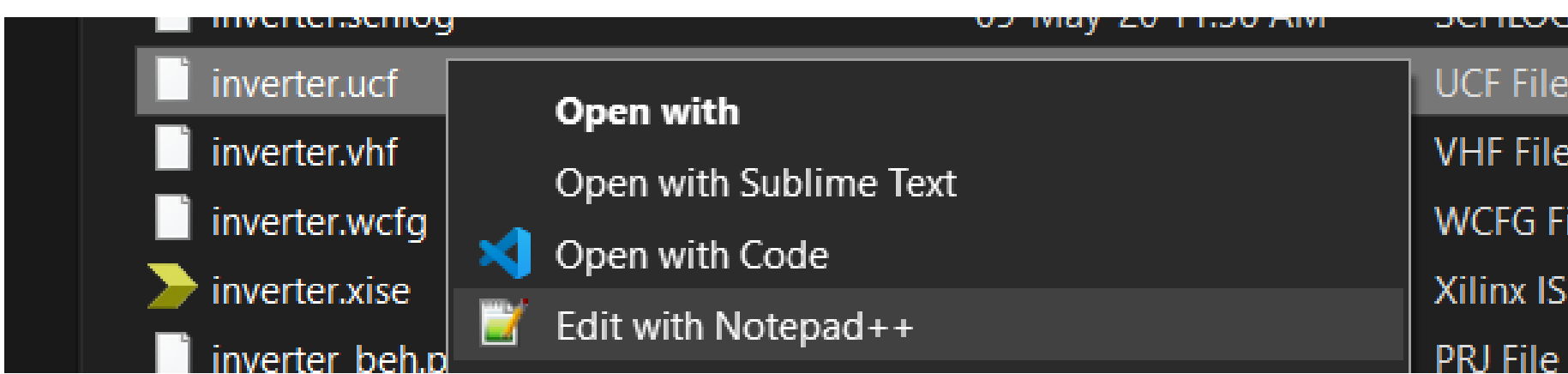


το inr δεν υπάρχει πλέον σαν επιλογή... επιλέγουμε το outp και πατάμε enter



...«δεσμεύεται» και το τελευταίο pin K12 (LDO) με το outp

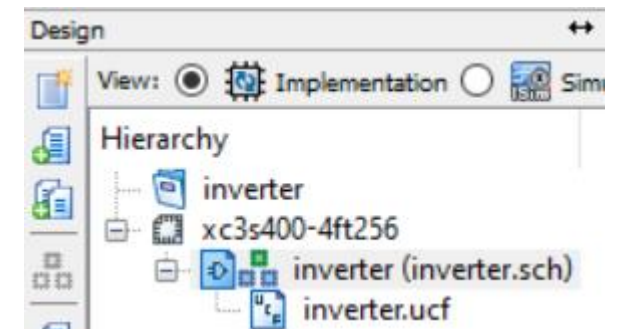
Αποθηκεύουμε με Save και κλείνουμε το παράθυρο PlanAhead!
Έχει δημιουργηθεί αρχείο inverter.ucf (user constraints file) το οποίο μπορούμε να επιθεωρήσουμε (ή και να διορθώσουμε αν θέλουμε!)



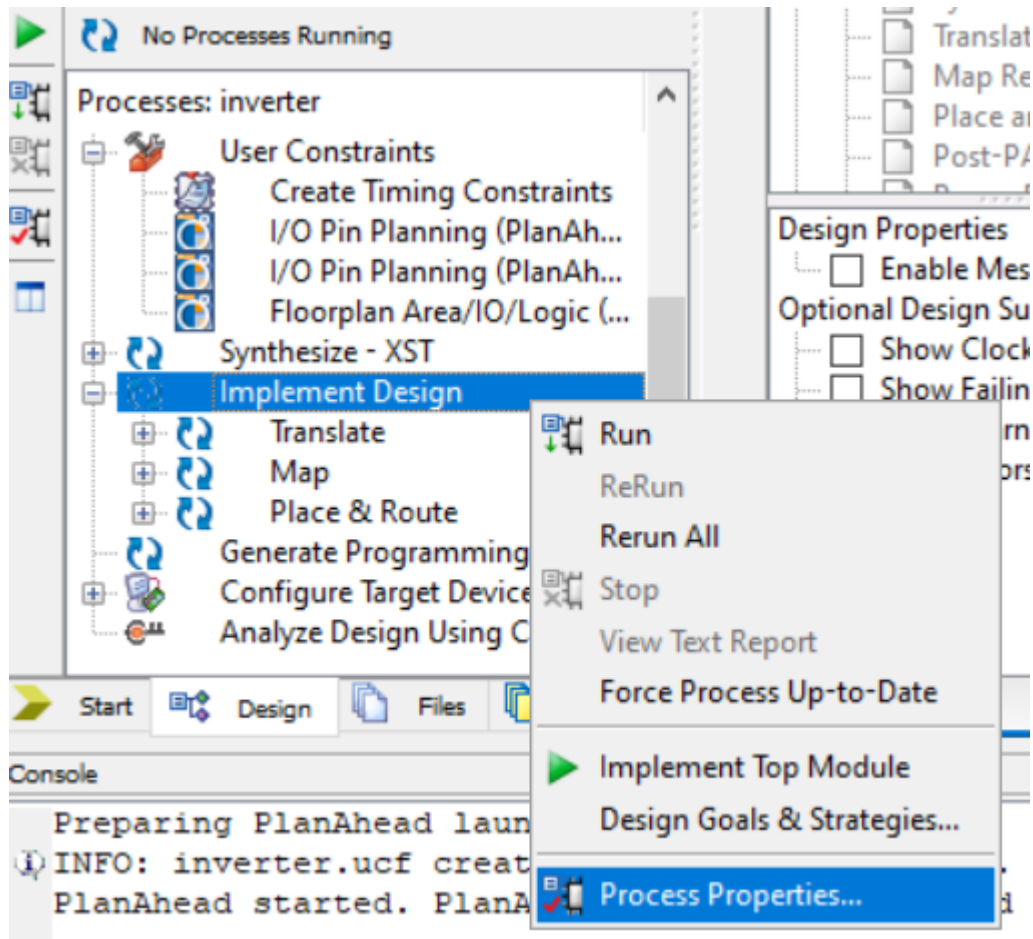
```
C:\Data\UOWM_INF\syndyastika\lab\askhseis\inverter\inverter.ucf - N
File Edit Search View Encoding Language Settings Tools Mac
inverter.ucf x
1
2 # PlanAhead Generated physical constraints
3
4 NET "inp" LOC = F12;
5 NET "outp" LOC = K12;
6
```

Input / Output	Board component	FPGA pin (Location)
INP	SW0	F12
OUTP	LD0	K12

Πίνακας 1: Τοποθετήσεις (locations) των Ε/Ε του κυκλώματος στο FPGA chip



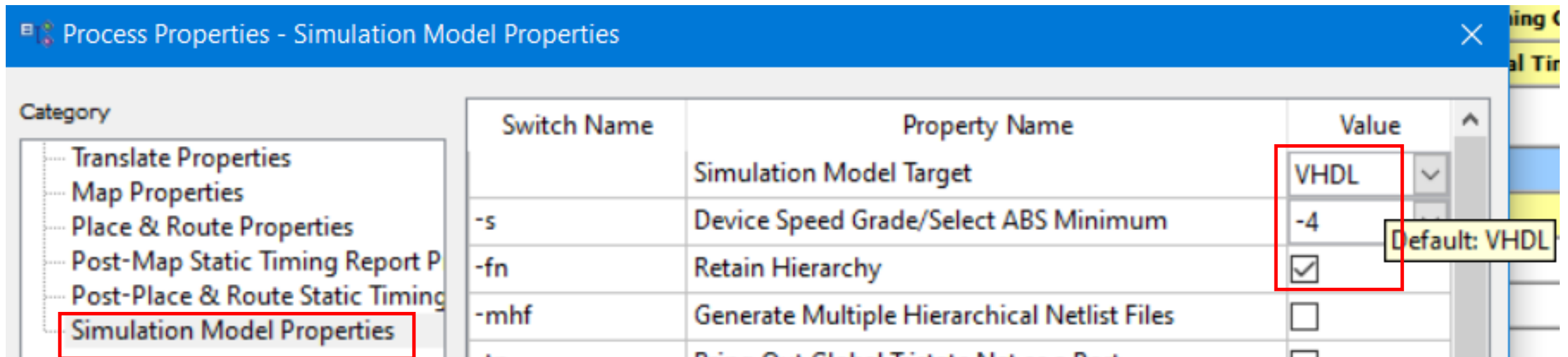
4. Μέσα στο παράθυρο Processes κάντε δεξί κλικ στο Implement Design και επιλέξτε Properties -> Process Properties και ενεργοποιείτε (βάζοντας tick ✓) τις επιλογές **Map Properties->Generate Detailed MAP Report**, και από τις επιλογές του Place&Route Properties τα: **Generate Asynchronous Delay Report, Generate Post-Place & Route Simulation Model** και Generate Post-Place&Route Power Report. Μετά κάντε κλικ στο **Apply**.



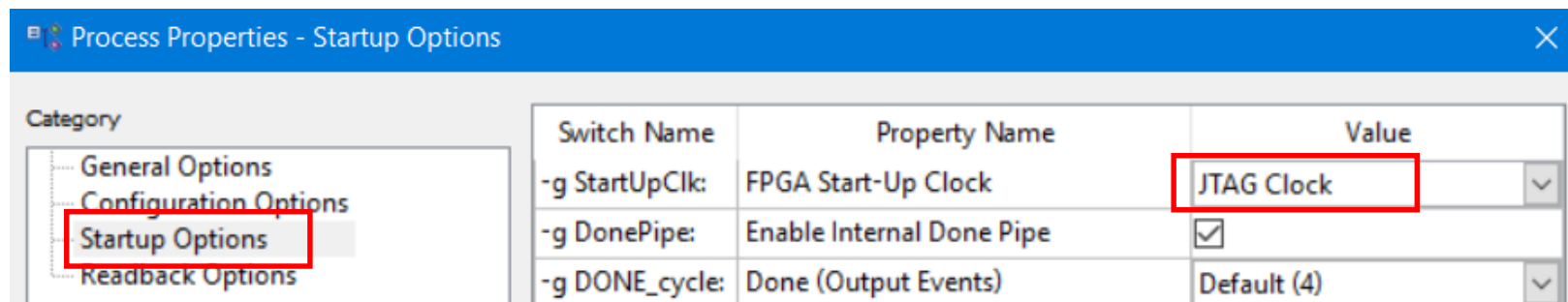
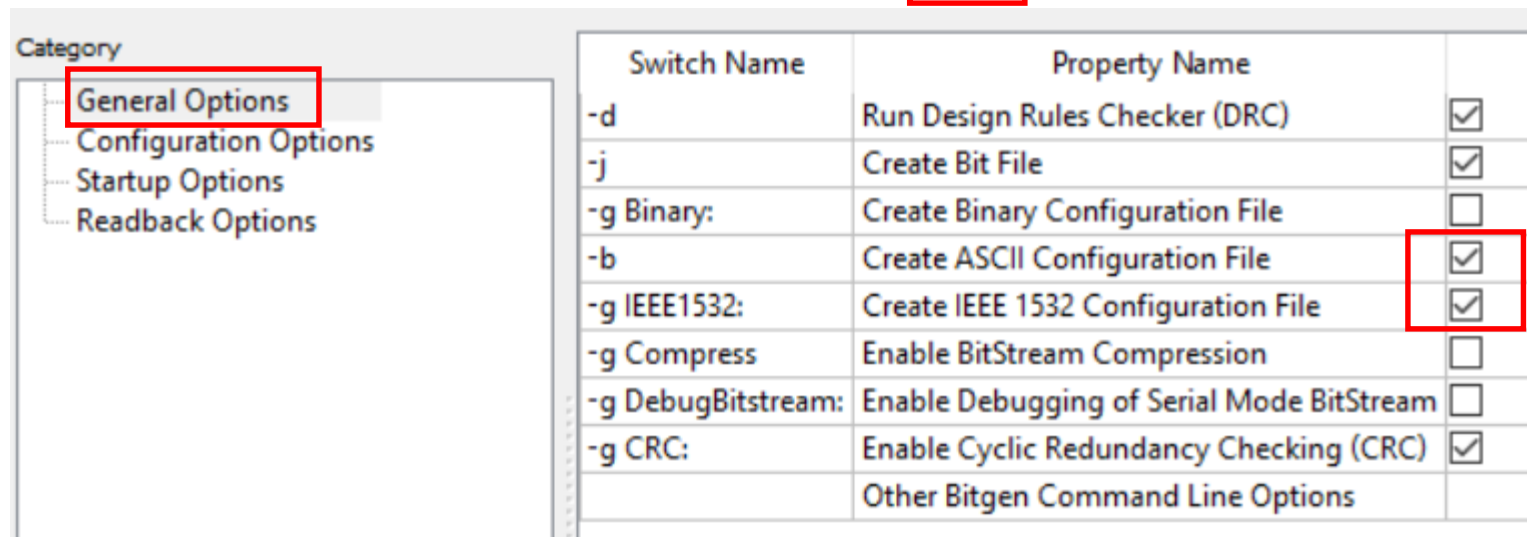
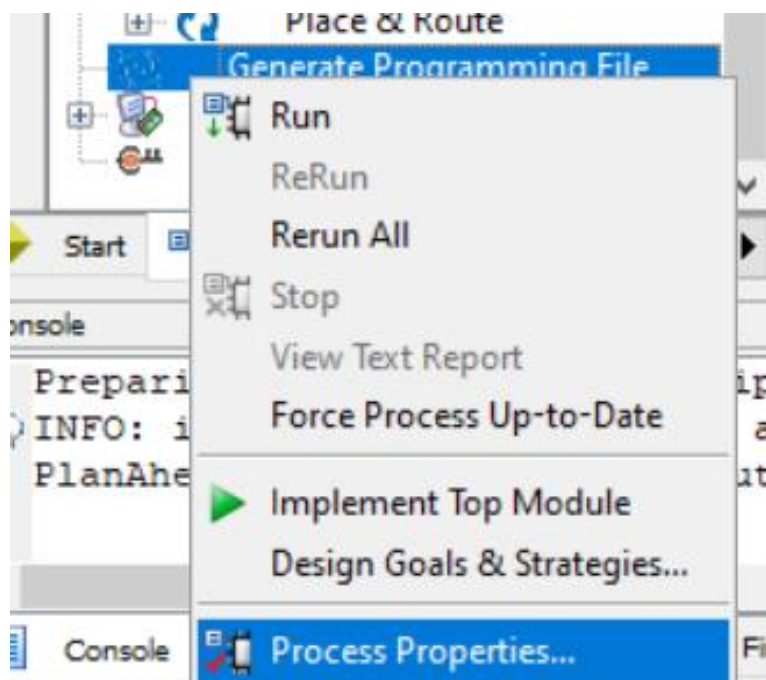
Category	Switch Name	Property Name	V
Map Properties	-xe	Extra Effort	Non
	-t	Starting Placer Cost Table (1-100)	1
	-logic_opt	Combinatorial Logic Optimization	<input type="checkbox"/>
	-register_duplication	Register Duplication	Off
	-x	Ignore User Timing Constraints	<input type="checkbox"/>
	-ntd	Timing Mode	Non
	-u	Trim Unconnected Signals	<input checked="" type="checkbox"/>
	-ignore_keep_hierarchy	Allow Logic Optimization Across Hierarchy	<input type="checkbox"/>
	-cm	Optimization Strategy (Cover Mode)	Area
	-detail	Generate Detailed MAP Report	<input checked="" type="checkbox"/>

Category	Switch Name	Property Name	
Place & Route Properties	-r, -p	Place And Route Mode	Norm
	-ol	Place & Route Effort Level (Overall)	High
	-pl	Placer Effort Level (Overrides Overall Level)	None
	-rl	Router Effort Level (Overrides Overall Level)	None
	-xe	Extra Effort (Highest PAR level only)	None
	-t	Starting Placer Cost Table (1-100)	1
	-x	Ignore User Timing Constraints	<input type="checkbox"/>
	-ntd	Timing Mode	Perfo
		Generate Asynchronous Delay Report	<input checked="" type="checkbox"/>
		Generate Clock Region Report	<input type="checkbox"/>
		Generate Post-Place & Route Simulation Model	<input checked="" type="checkbox"/>
		Generate Post-Place & Route Power Report	<input type="checkbox"/>

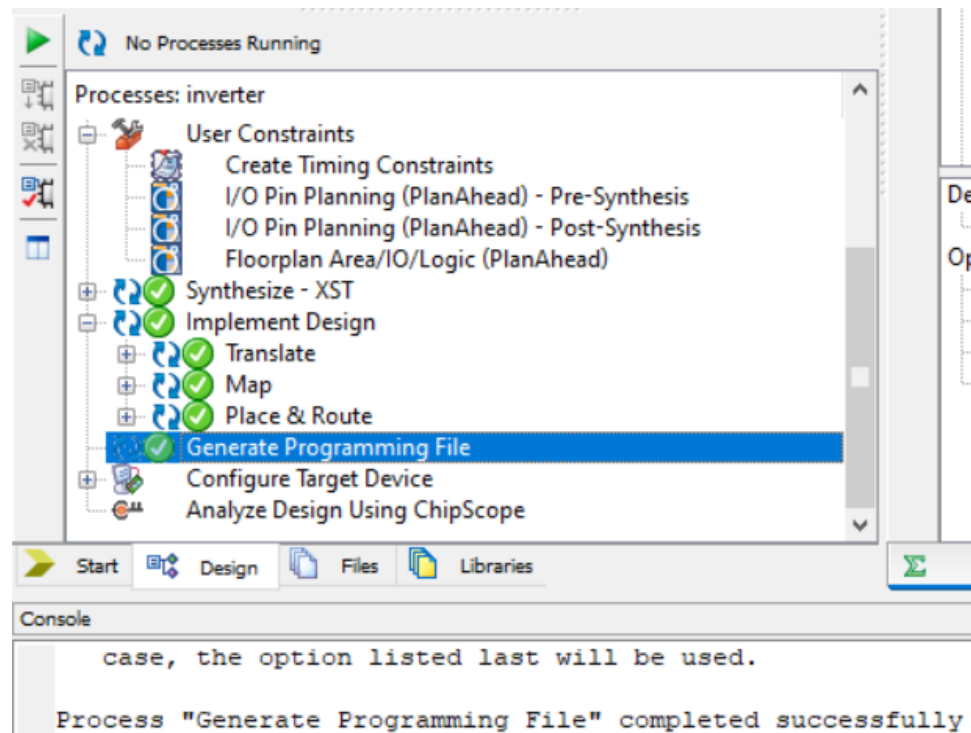
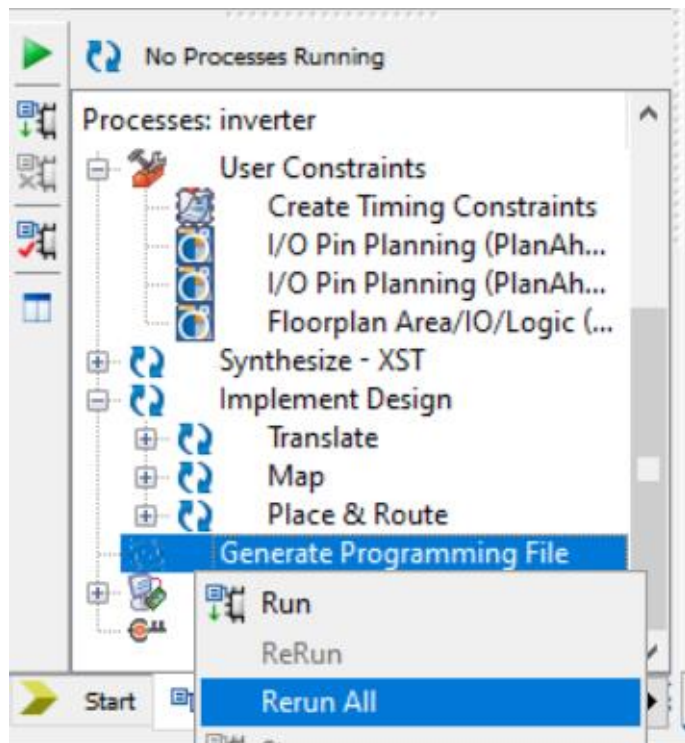
5. Μέσα στο παράθυρο Processes Properties κάντε δεξί κλικ στο Implement Design και επιλέξτε Properties -> Simulation Model Properties και ενεργοποιείτε (βάζοντας tick ✓) την επιλογή **Retain Hierarchy** και διαλέξτε την επιλογή **Simulation Model Target -> VHDL**. Μετά κάντε κλικ στο **OK**.



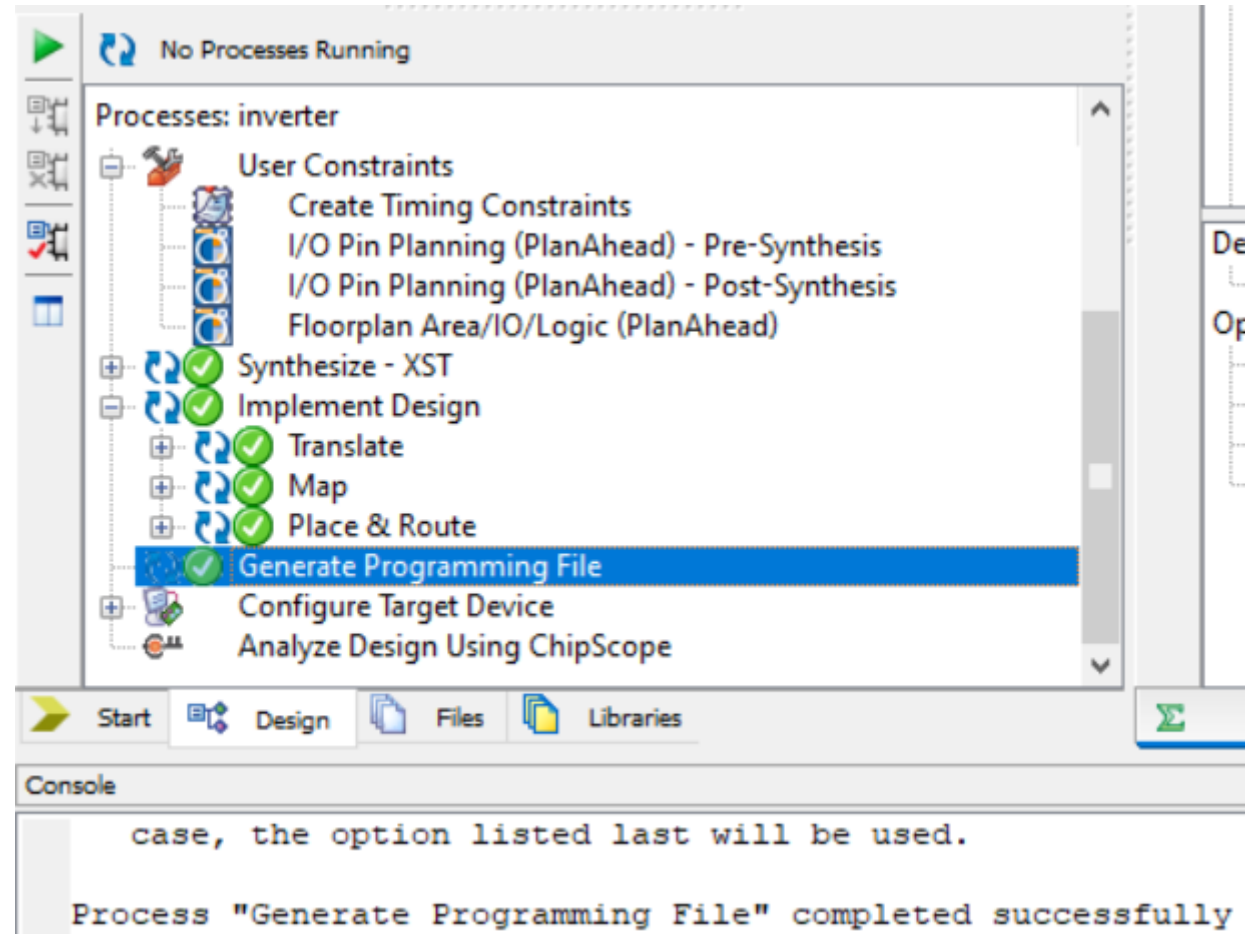
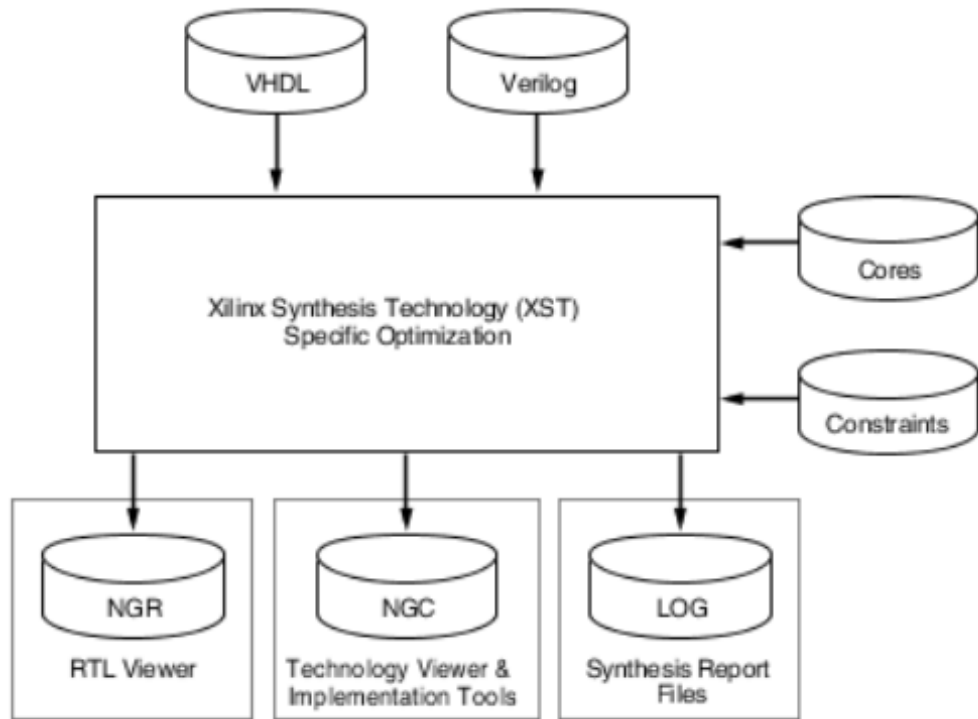
6. Μέσα στο παράθυρο Processes κάντε δεξί κλικ στο Generate Programming File και επιλέξτε Process Properties -> General Options και ενεργοποιείτε (βάζοντας tick ✓) τις επιλογές **Create ASCII Configuration File** και **Create IEEE 1532 Configuration File**. Μετά πηγαίνετε στην επιλογή Generate Programming File -> δεξί κλικ -> Process Properties -> Startup Options και στην λίστα **FPGA Start-up Clock** επιλέγετε την επιλογή **JTAG Clock**. Μετά κάντε κλικ στο **OK** για να βγείτε από αυτό το menu.



7. Μέσα στο παράθυρο Processes κάντε δεξί κλικ στο **Generate Programming File** και επιλέξτε **Rerun All**. Αυτή η ενέργεια θα προκαλέσει την εκτέλεση με συγκεκριμένη σειρά όλων των εργαλείων του περιβάλλοντος ISE που απαιτούνται για την υλοποίηση σε FPGA configuration (programming) bitstream του κυκλώματος που σχεδιάσατε. Δώστε ιδιαίτερη προσοχή στα reports και τις πληροφορίες που παρουσιάζονται κατά την εκτέλεση των διάφορων εργαλείων καθώς θα σας ζητηθεί να δώσετε απαντήσεις σε αυτό το κείμενο, όσον αφορά συγκεκριμένες λεπτομέριες από τις πληροφορίες αυτής της εκτέλεσης. Αν αφού εκτελεστούν όλα τα εργαλεία υλοποίησης του ISE, σας εμφανιστεί ένα παράθυρο με τίτλο **Xilinx Web Talk Dialog**, τότε απλά αγνοήστε το και κλείστε το με το κόκκινο X.



8. Αν όλα τα εργαλεία εκτελεστούν με επιτυχία και χωρίς λάθη ή άλλα προβλήματα τότε θα παρατηρήσετε πράσινα ticks σε όλα τα εργαλεία που εκτελέστηκαν, όπως επίσης και το μήνυμα: **Process "Generate Programming File" completed successfully** να εμφανίζεται στο κάτω μέρος του παραθύρου Console του ISE.



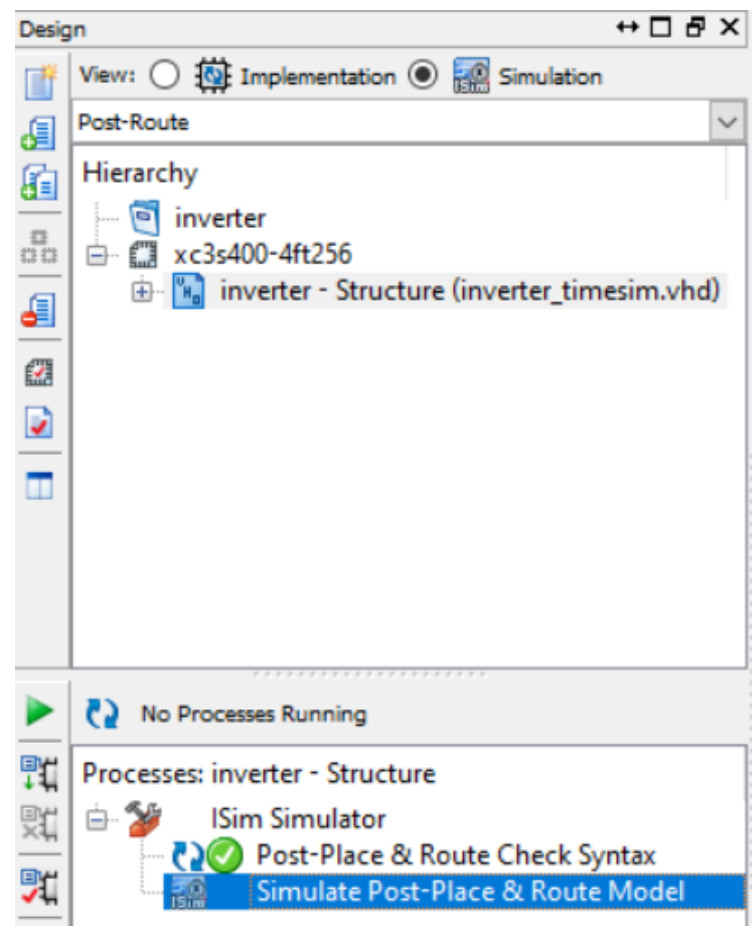
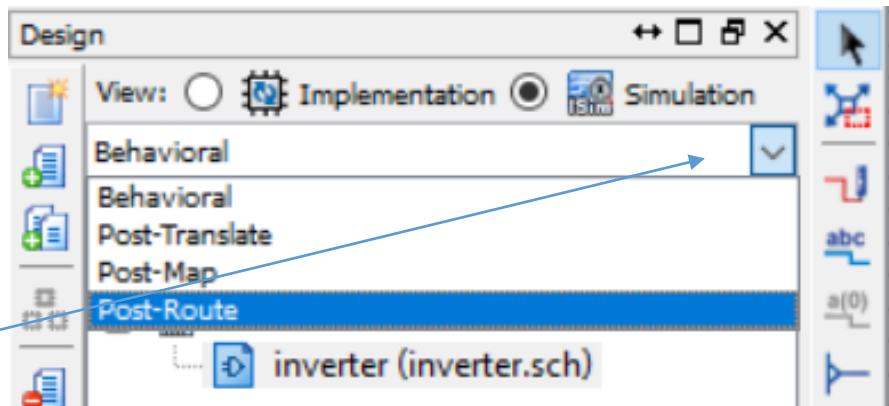
Programming File (Bitstream) → inverter.bit

[Xilinx ISE Synthesis](#)
[Xilinx ISE Implementation](#)

VI. Προσομοίωση υλοποιημένου δικτύου (post-route simulation)

Σημειώστε ότι πρωταρχικός σκοπός της προσομοίωσης είναι να ελέγξουμε την σωστή λειτουργία του κυκλώματός μας (αν με τις δεδομένες τιμές στις εισόδους, παίρνουμε τις αναμενόμενες τιμές στις εξόδους του κυκλώματος).

1. Στο παράθυρο **Design**, επιλέξτε στην λίστα του **View** :, την επιλογή **Post-Route Simulation**, αν δεν είναι ήδη επιλεγμένη.
2. Στο παράθυρο **Design**, επιλέξτε το σχηματικό σας σχέδιο, π.χ. **simple_inverter1(simple_inverter1.sch)**
3. Αν κάνετε επιτυχώς τα παραπάνω βήματα θα δείτε στο παράθυρο **Processes** το εργαλείο **Xilinx ISE Simulator**. Μπορείτε να ανοίξετε περισσότερο αυτή την ομάδα εργαλείων αν κάνετε κλικ στο κουτάκι με το +, οπότε θα εμφανιστεί η επιλογή **Simulate Post-Place & Route Model**. Με διπλό κλικ πάνω στο **Simulate Post-Place & Route Model** θα σας εμφανίσει τα παράθυρα για την προσομοίωση του κυκλώματός σας. Αν δεν σας ανοίξει το παράθυρο **Simulation** με τις κυματομορφές, τότε στο κάτω τμήμα του δεξιού μέρους του ISE κάντε κλικ στο tab **Simulation**.

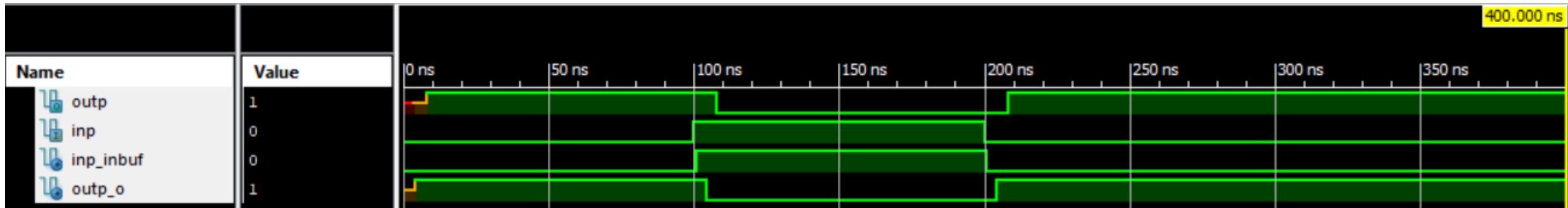


4. Όλα σχεδόν τα υπόλοιπα βήματα και οι εντολές για την εκτέλεση αυτής της προσομοίωσης είναι ίδια με εκείνα της λειτουργικής προσομοίωσης που εξηγήθηκε πιο πριν. Αφιερώστε χρόνο για να εξοικειωθείτε με την προσομοίωση υλοποιημένου δικτύου (post-route simulation) και να μετρήσετε τις καθυστερήσεις στο χρόνο της απόκρισης των εξόδων (βλέποντας τις αποστάσεις στο χρόνο από τις αλλαγές στις τιμές των εισόδων, έως να αλλάξουν οι τιμές των εξόδων του κυκλώματος) χρησιμοποιώντας τους markers στο παράθυρο **Simulation**, όπως εξηγήθηκε πριν στην λειτουργική προσομοίωση. Όταν τελειώσετε την προσομοίωση και πιστεύετε ότι το κύκλωμά σας λειτουργεί σωστά κλείστε το παράθυρο της κυματομορφής της εξομοίωσης.

```
ISim> restart
```

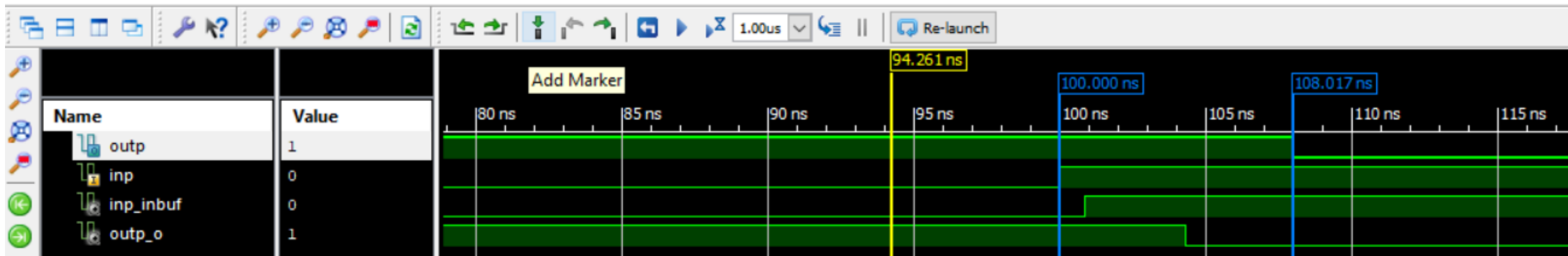
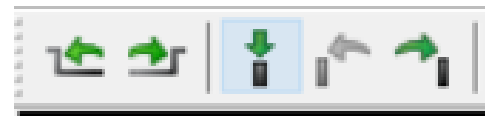
```
ISim> isim force add inp 0 -time 0 ns -value 1 -time 100 ns -value 0 -time 200 ns
```

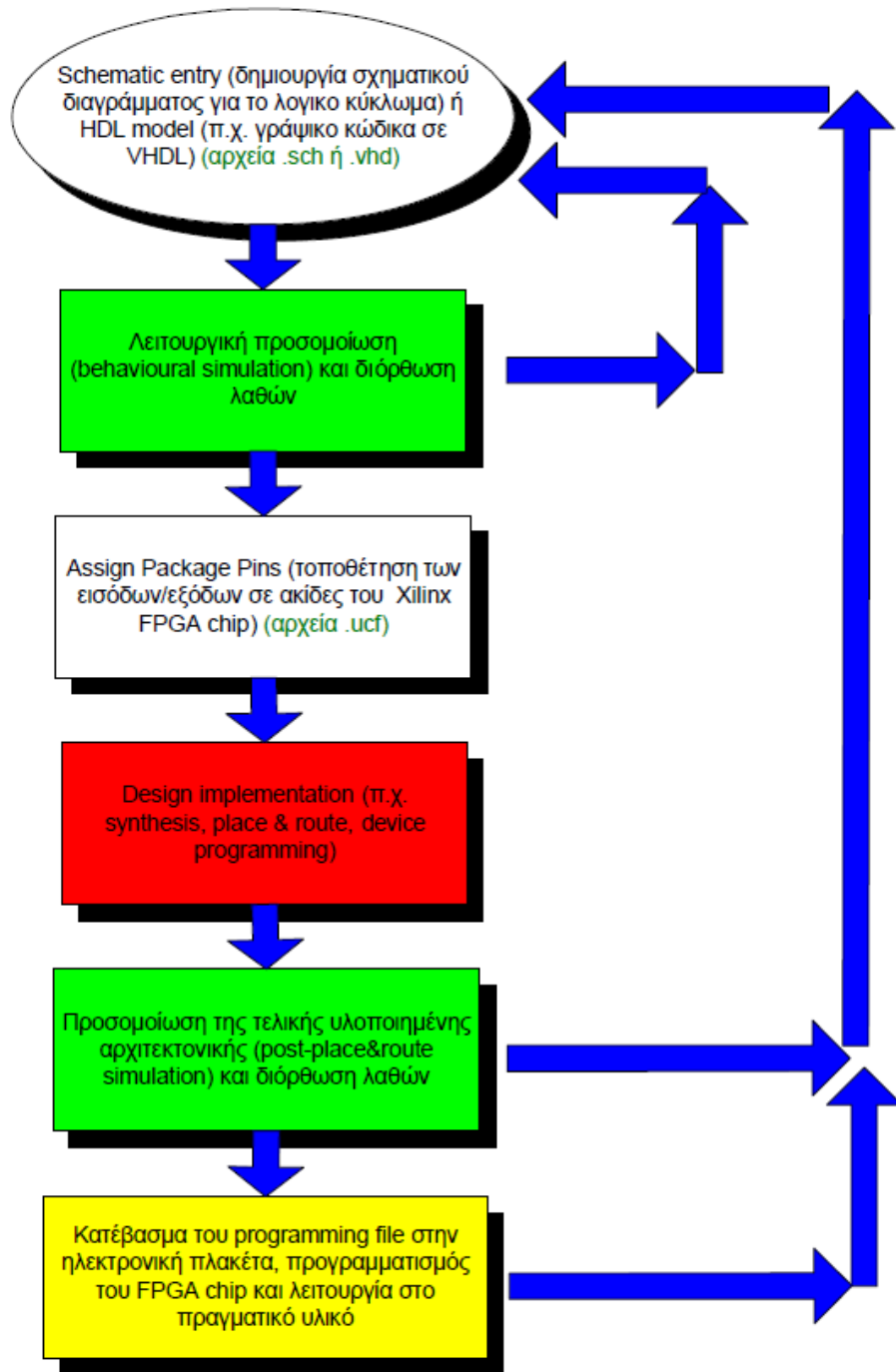
```
ISim> run 400 ns
```



11. Μπορείτε να προσθέτετε έναν ή περισσότερους **markers** (κάθετες χρωματιστές γραμμές μέτρησης) στο παράθυρο Simulation (κυματομορφής) με το πράσινο εικονίδιο **Single Marker** που βρίσκεται δίπλα στο παράθυρο κυματομορφών. Παρατηρείστε ότι το πάνω μέρος του κάθε marker υπάρχει ο χρόνος μέσα στην προσομοίωση στον οποίο αυτός ο marker δείχνει. Έτσι μπορείτε να μετράτε αποστάσεις στον χρόνο από μία αλλαγή τιμής ενός σήματος έως μία άλλη αλλαγή τιμής. Αυτό θα σας είναι χρήσιμο για να μετράτε την καθυστέρηση απόκρισης του κυκλώματός σας από τις εισόδους προς τις εξόδους του. Επίσης χρησιμοποιώντας κατάλληλους συνδυασμούς των εικονιδίων των markers και των εικονιδίων των zoom, μπορείτε με ακρίβεια να τοποθετείτε τους makers εκεί που θέλετε και να μετράτε με ακρίβεια την απόσταση στον χρόνο μεταξύ τους. Όταν κλειδώνει σε μία αλλαγή της τιμής του κάθε σήματος που επιλέγετε ο marker τότε εμφανίζει κοντά στην αλλαγή έναν μικρό δίσκο, ώστε να το αφήσετε εκεί.

Αποθηκεύστε το νέο simulation: inverter_post.wcfg





Για αυτή την εβδομάδα...

Κατεβάστε και εγκαταστήστε το Xilinx ISE 14.7 στον υπολογιστή σας με WEBPACK License αν δεν το έχετε ήδη κάνει.

Συνεχίστε το project inverter (προσθήκη constraints, set implementation properties, δημιουργία programming file, προσομοίωση δικτύου) ακολουθώντας και τις οδηγίες από το φυλλάδιο 2.

Μου αποστέλλετε συμπιεσμένο το φάκελο του project ως επισύναψη ή μέσω Google Drive.