

Αρχιτεκτονική Η/Υ ΙΙ

**Ασκήσεις Πράξης:** Ενότητα 2

Πετρέλλης Νικόλαος

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.

**Χρηματοδότηση**

* Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Τ.Ε.Ι. Θεσσαλίας**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



**Σκοποί ενότητας**

Σκοπός αυτού του εργαστηρίου είναι να γνωρίσουμε και να **κατανοήσουμε το περιβάλλον** VHDL προγραμματισμού (πχ Max+Plus II, Leonardo-Spectrum, και άλλα), **με παραδείγματα προσομοίωσης** (discrete-event simulation) βασικών λειτουργικών μονάδων ενός υπολογιστικού συστήματος, έτσι ώστε να εξοικειωθούμε τόσο με τη γλώσσα VHDL, όσο και με το περιβάλλον ανάπτυξης.

**Περιεχόμενα**

[1. Επίπεδα (αφαιρετικά) σχεδίασης στη VHDL (*abstraction levels*) 4](#_Toc411444292)

[1.1. Επίπεδο λειτουργίας (*Functional level*) 4](#_Toc411444293)

[1.2. Επίπεδο συμπεριφοράς (*Behavioral level*) 4](#_Toc411444294)

[1.3. Επίπεδο RTL (*Register Transfer Level*) 4](#_Toc411444295)

[1.4. Επίπεδο λογικής ή πυλών (*Logic/gate level*) 5](#_Toc411444296)

[2. Περιγραφή της δομής (*structure*) ενός κυκλώματος 5](#_Toc411444297)

[3. Περιγραφή συμπεριφοράς/λειτουργίας (*Behavior/Function*) 5](#_Toc411444298)

[4. Διαδικασία σχεδίασης 6](#_Toc411444299)

[5. Το περιβάλλον σχεδίασης και προγραμματισμού Max+PlusII 6](#_Toc411444300)

[6. Βήματα εισαγωγής σχεδίου κυκλώματος στο περιβάλλον Max+PlusII 7](#_Toc411444301)

[6.1. Με πρόγραμμα (κώδικα) σε VHDL 7](#_Toc411444302)

[6.2. Με σχηματικό διάγραμμα 8](#_Toc411444303)

[6.3. Με τον πίνακα αληθείας 9](#_Toc411444304)

[7. Ασκήσεις 9](#_Toc411444305)

[7.1. Άσκηση 1 10](#_Toc411444306)

[Σημειώματα 13](#_Toc411444307)

**Πίνακες**

[*Πίνακας 1* 10](#_Toc411444308)

**Εικόνες**

[*Εικόνα 1* 4](#_Toc411444309)

[*Εικόνα 2* 5](#_Toc411444310)

[*Εικόνα 3* 6](#_Toc411444311)

[*Εικόνα 4* 7](#_Toc411444312)

[*Εικόνα 5* 10](#_Toc411444313)

[*Εικόνα 6* 10](#_Toc411444314)

[*Εικόνα 7* 12](#_Toc411444315)

# Επίπεδα (αφαιρετικά) σχεδίασης στη VHDL (abstraction levels)



Εικόνα 1

## Επίπεδο λειτουργίας (Functional level)

Οι αλγόριθμοι μπορούν να περιγραφούν και να γίνει η προσομοίωσή τους (functional simulation) στο επίπεδο αυτό, καθώς δεν χρειάζονται κάποιες χρονικές πληροφορίες (timing information).

## Επίπεδο συμπεριφοράς (Behavioral level)

Σε αυτό το επίπεδο δεν είναι απαραίτητη η γνώση της εσωτερικής αρχιτεκτονικής. Ένα μοντέλο (behavioral model) μπορεί να περιγραφεί ως λειτουργικές μονάδες (functional modules), και η διασύνδεσή τους (interface). Π.χ. η προσομοίωση της λειτουργίας ενός εξειδικευμένου μοντέλου το οποίο θα συνδεθεί σε ένα επεξεργαστή, όπου σε αυτή την περίπτωση η εσωτερική λειτουργία του επεξεργαστή είναι σύνθετη, και δεν έχει καμία σημασία (black box) και είναι σημαντική μόνο η εξωτερική του συμπεριφορά.

## Επίπεδο RTL (Register Transfer Level)

Το μοντέλο εδώ, αποτελείται από κώδικα σε κάποια γλώσσα, η οποία περιγράφει την ασύγχρονη και σύγχρονη κατάσταση των μηχανών.

## Επίπεδο λογικής ή πυλών (Logic/gate level)

Αποτελεί περιγραφή με λογικές εκφράσεις στην άλγεβρα του Boole, ή με δίκτυο πυλών.

# Περιγραφή της δομής (structure) ενός κυκλώματος

Κάθε ψηφιακό ηλεκτρονικό σύστημα μπορεί να περιγραφεί ως μία μονάδα (module)

με εισόδους – εξόδους (inputs/outputs). Π.χ.:



Εικόνα 2

# Περιγραφή συμπεριφοράς/λειτουργίας (Behavior/Function)

Πολλές φορές η σχεδίαση μιας μονάδας δομικά δεν είναι η κατάλληλη μέθοδος, ιδίως όταν χρησιμοποιούνται ολοκληρωμένα κυκλώματα (ICs). Σε αυτές τις περιπτώσεις υλοποιούμε την περιγραφή της λειτουργίας ή συμπεριφοράς της μονάδας.

Για παράδειγμα, εάν η συνάρτηση λειτουργίας της οντότητας F στο παραπάνω σχήμα (a), είναι η e**X**clusive-**OR**, τότε η περιγραφή της συμπεριφοράς της οντότητας F, δίνεται από την παρακάτω Boolean έκφραση:

$$Y= \overbar{A}∙\overbar{B}+A∙\overbar{B}$$

Αυτή η περιγραφή μπορεί να γίνει και με προγραμματισμό, όπως γίνεται µε τη γλώσσα VHDL.

Όταν η δομή και λειτουργία μιας μονάδας καθορισθεί, μπορούμε να προσομοιώσουμε τη μονάδα µε την εκτέλεση της περιγραφής της.

# Διαδικασία σχεδίασης

Μία απλή και τυπική διαδικασία σχεδίασης που θα μπορούσαμε να ακολουθήσουμε, δίνεται με το παρακάτω σχήμα.



Εικόνα 3

# Το περιβάλλον σχεδίασης και προγραμματισμού Max+PlusII

Μία άποψη του περιβάλλοντος σχεδίασης και προγραμματισμού σε VHDL ψηφιακών κυκλωμάτων, Max+plusII της Altera, που θα μπορούσε να χρησιμοποιηθεί για την υλοποίηση των ασκήσεων, δίνεται στην εικόνα που ακολουθεί.

Υπάρχουν βέβαια και αρκετές άλλες ολοκληρωμένες εφαρμογές που θα μπορούσαν να χρησιμοποιηθούν για τον ίδιο σκοπό (πχ SLIDE, OODE, BORIS, Synopsis VSS, View Logic, Leonardo-Spectrum, GM VHDL compiler/simulator, Electric, και άλλα). Έτσι, ο αναγνώστης θα μπορούσε να χρησιμοποιήσει και οποιαδήποτε άλλη εφαρμογή (πχ ένα VHDL compiler & simulator), που θα επιτρέπει την εισαγωγή (VHDL code editing), ή ακόμα και τη σχηματική εισαγωγή του κυκλώματος υπό μελέτη, τη μεταγλώττιση (VHDL compilation), και εκτέλεση (VHDL simulation) προγραμμάτων σε VHDL, για την υλοποίηση των παραδειγμάτων και των ζητούμενων ασκήσεων.



Εικόνα 4

Θα ξεκινήσουμε αναφέροντας ότι στο περιβάλλον Max+plusII (αλλά και σε άλλες εφαρμογές), είναι δυνατόν η εισαγωγή σχεδίου ενός κυκλώματος να γίνει µε τρεις τρόπους:

* Με **σχηματικό διάγραμμα** (με τη βοήθεια του επεξεργαστή γραφικών παραστάσεων – Graphics editor).
* Με τη **γλώσσα VHDL** (με τη βοήθεια του επεξεργαστή κειμένων – Text editor).
* Με τον **πίνακα αληθείας** (με τη βοήθεια του επεξεργαστή κυματομορφών – Waveform editor).

Βασικά, για την εισαγωγή κώδικα σε VHDL, θα μπορούσε να χρησιμοποιηθεί οποιοσδήποτε επεξεργαστής κειμένου μπορεί να δημιουργήσει ένα απλό αρχείο ASCII κειμένου.

# Βήματα εισαγωγής σχεδίου κυκλώματος στο περιβάλλον Max+PlusII

## Με πρόγραμμα (κώδικα) σε VHDL

Περιγραφή της διαδικασίας σχεδίασης και μελέτης ενός λογικού κυκλώματος με κώδικα σε VHDL.

**Α. Εισαγωγή προγράμματος**

**Α.1.** Δίνουμε όνομα στο έργο μας (File⇒Project⇒Name), πχ example1.

**Α.2.** Ανοίγουμε τον επεξεργαστή κειμένου (MAX+plusII ⇒ **Text Editor**).

**Α.3.** Δίνουμε όνομα στο αρχείο με τον κώδικα (FILE ⇒ SAVE AS), πχ example1.**vhd**, και πληκτρολογούμε τον κώδικα που περιγράφει το παραπάνω κύκλωμα.

**Β. Μεταγλώττισης του προγράμματος**

**Β.1.** Ανοίγουμε το παράθυρο του μεταγλωττιστή (MAX+plusII ⇒ **Compiler**).

**Β.2.** Για να δημιουργήσει ο μεταγλωττιστής την πληροφορία που χρειάζεται για την εκτέλεση της προσομοίωσης στη συνέχεια (προσομοίωση χρονισμού), κάνουμε την επιλογή Processing ⇒ **Functional** **SNF** **Extractor** (θα μπορούσαμε να κάνουμε την επιλογή Interfaces Netlist writer, έτσι ώστε να κάνουμε structural VHDL).

**Β.3.** Πατάμε το πλήκτρο **Start** του μεταγλωττιστή.

**Γ. Προσομοίωσης λειτουργίας**

**Γ.1.** Ανοίγουμε τον επεξεργαστή κυματομορφών (MAX+plusII ⇒ **Waveform Editor**) για να δημιουργήσουμε τις επιθυμητές κυματομορφές (διανύσματα δοκιμής) για να παραστήσουμε τα σήματα εισόδου.

**Γ.2.** Δίνουμε όνομα στο αρχείο με τα διανύσματα δοκιμών (FILE ⇒ SAVE AS), πχ example1.**scf**, το οποίο και θα αποθηκεύσει τις κυματομορφές που θα χρησιμοποιηθούν στην προσομοίωση.

**Γ.3.** Εισάγουμε τους κόμβους του έργου με την επιλογή Node ⇒ **Enter Nodes from SNF** (List).

**Γ.4.** Καθορίζουμε το συνολικό χρονικό διάστημα προσομοίωσης του κυκλώματος (File ⇒ **End Time**), και το πλάτος των κυματομορφών (Options ⇒**Grid Size**) στο πλέγμα.

**Γ.5.** Σχεδιάζουμε τις κυματομορφές εισόδου χρησιμοποιώντας το ***waveform editing tool***, πιέζοντας το εικονίδιό του στο αριστερό άκρο του παραθύρου.

**Γ.6.** Ανοίγουμε το παράθυρο της προσομοίωσης (MAX+plusII ⇒ **Simulator**), και πατάμε το πλήκτρο **Start** για να ξεκινήσει η εκτέλεση της προσομοίωσης της λειτουργίας, τα αποτελέσματα της οποίας μπορούμε να παρατηρήσουμε στο παράθυρο του επεξεργαστή κυματομορφών (Open SCF).

## Με σχηματικό διάγραμμα

Περιγραφή της διαδικασίας σχεδίασης και μελέτης ενός λογικού κυκλώματος με σχηματικό διάγραμμα.

**Α. Εισαγωγή του σχηματικού διαγράμματος**

**Α.1.** Δίνουμε όνομα στο έργο μας (File ⇒ Project ⇒ **Name**), πχ example1.

**Α.2.** Ανοίγουμε τον επεξεργαστή γραφικών παραστάσεων (MAX+plusII ⇒ **Graphics** **Editor**).

**Α.3.** Δίνουμε όνομα στο αρχείο με το σχηματικό διάγραμμα της λογικής συνάρτησης του κυκλώματος (FILE ⇒ SAVE AS), πχ example1.**gdf**.

**Α.4.** Σχεδιάζουμε το κύκλωμα εισάγοντας πρώτα τα συμβολικά στοιχεία του (πχ πύλες) από την κατάλληλη βιβλιοθήκη (πχ Primitives) επιλέγοντας Symbol ⇒ **Enter** **Symbol**.

**Α.5.** Στη συνέχεια συνδέουμε μεταξύ τους τα σύμβολα του σχηματικού διαγράμματος με τη βοήθεια του ποντικιού, και χρησιμοποιώντας το εργαλείο επιλογής **selection tool** το οποίο βρίσκεται ως εικονίδιο στο αριστερό άνω άκρο του παραθύρου.

**Β. Μεταγλώττισης**

Παρομοίως με την προηγούμενη περίπτωση εισαγωγής σχεδίου κυκλώματος.

**Γ. Προσομοίωση λειτουργίας**

Παρομοίως με την προηγούμενη περίπτωση εισαγωγής σχεδίου κυκλώματος.

## Με τον πίνακα αληθείας

Περιγραφή της διαδικασίας σχεδίασης και μελέτης ενός λογικού κυκλώματος με τη βοήθεια του πίνακα αληθείας.

**Α. Εισαγωγής του πίνακα αληθείας μέσω διαγράμματος χρονισμού**

**Α.1.** Δίνουμε όνομα στο έργο μας (File ⇒ Project ⇒ **Name**), πχ example1.

**Α.2.** Ανοίγουμε τον επεξεργαστή κυματομορφών (MAX+plusII ⇒ **Waveform Editor**) για να δημιουργήσουμε ένα αρχείο σχεδίου κυματομορφής.

**Α.3.** Δίνουμε όνομα στο αρχείο (FILE ⇒ SAVE AS), πχ example1.**wdf**.

**Α.4.** Δημιουργούμε το διάγραμμα χρονισμού ορίζοντας τα σήματα εισόδου και εξόδου του κυκλώματος με την επιλογή Node ⇒ **Insert Node**.

**Α.5.** Καθορίζουμε το συνολικό χρονικό διάστημα προσομοίωσης του κυκλώματος (File ⇒ **End Time**), και το πλάτος των κυματομορφών (Options ⇒ **Grid Size**) στο πλέγμα.

**Α.6.** Σχεδιάζουμε τις κυματομορφές εισόδου **και** εξόδο, χρησιμοποιώντας το **waveform editing tool**, πιέζοντας το εικονίδιό του στο αριστερό άκρο του παραθύρου.

**Β. Μεταγλώττισης**

Παρομοίως με την προηγούμενη περίπτωση εισαγωγής σχεδίου κυκλώματος.

**Γ. Προσομοίωση λειτουργίας**

Ανοίγουμε το παράθυρο της προσομοίωσης (MAX+plusII ⇒ **Simulator**) και πατάμε το πλήκτρο **Start** για να ξεκινήσει η εκτέλεση της προσομοίωσης της λειτουργίας.

# Ασκήσεις

## Άσκηση 1

Για το παρακάτω κύκλωμα $ f=x\_{1}x\_{2}+\overbar{x\_{2}}x\_{3}$ να γραφεί το πρόγραμμα σε VHDL το οποίο να περιγράφει.

Πίνακας 1

| **X1** | **X2** | **X3** | **f** |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



Εικόνα 5

Ας δούμε την υλοποίηση της παραπάνω άσκησης στο περιβάλλον της εφαρμογής **Max**+**plusII**.



Εικόνα 6

Τα βήματα υλοποίησης της παραπάνω άσκησης έχουν ως εξής:

**Εισαγωγή του προγράμματος**

1. Δίνουμε όνομα στο έργο μας (File ⇒ Project ⇒ **Name**), πχ example1.
2. Ανοίγουμε τον επεξεργαστή κειμένου (MAX+plusII ⇒ **Text Editor**).
3. Δίνουμε όνομα στο αρχείο με τον κώδικα (FILE ⇒ SAVE AS), πχ example1.**vhd**, και πληκτρολογούμε τον κώδικα που περιγράφει το παραπάνω κύκλωμα.

**Μεταγλώττιση του προγράμματος**

1. Ανοίγουμε το παράθυρο του μεταγλωττιστή (MAX+plusII ⇒ **Compiler**).
2. Για να δημιουργήσει ο μεταγλωττιστής την πληροφορία που χρειάζεται για την εκτέλεση της προσομοίωσης στη συνέχεια, κάνουμε την επιλογή Processing ⇒ **Functional SNF Extractor** (θα μπορούσαμε να κάνουμε την επιλογή Interfaces Netlist writer, έτσι ώστε να κάνουμε structural VHDL).
3. Πατάμε το πλήκτρο **Start** του μεταγλωττιστή.

**Προσομοίωση της λειτουργίας**

1. Ανοίγουμε τον επεξεργαστή κυματομορφών (MAX+plusII ⇒ **Waveform Editor**) για να δημιουργήσουμε τις επιθυμητές κυματομορφές (διανύσματα δοκιμής) για να παραστήσουμε τα σήματα εισόδου.
2. Δίνουμε όνομα στο αρχείο με τα διανύσματα δοκιμών (FILE ⇒S AVE AS), πχ example1.**scf**, το οποίο και θα αποθηκεύσει τις κυματομορφές που θα χρησιμοποιηθούν στην προσομοίωση.
3. Εισάγουμε τους κόμβους του έργου με την επιλογή Node ⇒ **Enter Nodes from SNF** (List), **x1**, **x2**, **x3**, **f**.
4. Καθορίζουμε το συνολικό χρονικό διάστημα προσομοίωσης του κυκλώματος (File ⇒ **End Time**) σε **160ns** και το πλάτος των κυματομορφών (Options ⇒ **Grid Size**) στο πλέγμα σε **20ns**.
5. Σχεδιάζουμε τις κυματομορφές εισόδου χρησιμοποιώντας το ***waveform editing tool***, πιέζοντας το εικονίδιό του στο αριστερό άκρο του παραθύρου.
6. Ανοίγουμε το παράθυρο της προσομοίωσης (MAX+plusII ⇒ **Simulator**) και πατάμε το πλήκτρο **Start** για να ξεκινήσει η εκτέλεση της προσομοίωσης της λειτουργίας, τα αποτελέσματα της οποίας μπορούμε να παρατηρήσουμε στο παράθυρο του επεξεργαστή κυματομορφών (Open SCF).

Μία άλλη άποψη του παραπάνω παραδείγματος του προγράμματος VHDL εισαγωγής και μεταγλώττισης στο περιβάλλον της εφαρμογής **Leonardo-Spectrum** της Altera,

με τα αποτελέσματα της μεταγλώττισης και την αναφορά, έχει ως εξής:



Εικόνα 7

**Τέλος Ενότητας**

Επεξεργασία: Σοφιανίδου Γεωργία

 

#

# Σημειώματα

**Σημείωμα Ιστορικού Εκδόσεων Έργου**

Το παρόν έργο αποτελεί την έκδοση 1.01.

**Σημείωμα Αναφοράς**

Copyright Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλίας, Πετρέλλης Χ. Νικόλαος, 2015. Πετρέλλης Χ. Νικόλαος. «Αρχιτεκτονική Η/Υ ΙΙ». Έκδοση: 1.0. Λάρισα 01/03/2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://cdev.teilar.gr/courses/TMA112/index.php>.

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λπ., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] [http://creativecommons.org/licenses/by-nc-sa/4.0/](http://creativecommons.org/licenses/by-nc-sa/4.0/deed.el)

Ως **Μη Εμπορική** ορίζεται η χρήση:

που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο,

που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο,

που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο.

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Διατήρηση Σημειωμάτων**

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

* το Σημείωμα Αναφοράς,
* το Σημείωμα Αδειοδότησης,
* τη Δήλωση Διατήρησης Σημειωμάτων,
* το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει).

μαζί με τους συνοδευόμενους υπερσυνδέσμους.