



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:
ΗΡΥ 203 - ΠΡΟΧΩΡΗΜΕΝΗ ΛΟΓΙΚΗ ΣΧΕΔΙΑΣΗ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018

Εργαστήριο 1

ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΗ ΓΛΩΣΣΑ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ VHDL ΚΑΙ
ΤΗΝ ΙΕΡΑΡΧΙΚΗ ΣΧΕΔΙΑΣΗ

ΕΚΠΟΝΗΣΗ: Καθ. Α. Δόλλας, Δρ. Κ. Παπαδημητρίου,
Δρ. Ε. Σωτηριάδης, Μ. Κιμιωνής

ΒΟΗΘΟΙ: (θα ανακοινωθούν ονόματα)

ΕΚΔΟΣΗ : 9.1

Χανιά

Σκοπός

Είναι η εξοικείωση με τη γλώσσα περιγραφής υλικού VHDL (VHSIC Hardware Description Language) για απλά συνδυαστικά κυκλώματα, και με την πλήρη σχεδιαστική ροή απλών ψηφιακών συστημάτων. Με τη χρήση του εργαλείου Xilinx ISE θα δημιουργήσετε ένα project στο οποίο θα δημιουργήσετε αρχεία, θα κάνετε σύνθεση του κώδικα σας, και στη συνέχεια προσομοίωση (simulation). Το επόμενο βήμα είναι η υλοποίηση της σχεδίασης μέσα από τα στάδια της μετάφρασης (translation), της αποτύπωσης (map) και της τοποθέτησης και διασύνδεσης (place and route). Τελευταίο στάδιο είναι η δημιουργία του αρχείου προγραμματισμού (.bit file) της αναδιατάσσόμενης συσκευής (FPGA) και ο προγραμματισμός της FPGA. Επαληθεύστε ότι το κύκλωμα λειτουργεί σύμφωνα με ότι παρατηρήσατε στην προσομοίωση. Για την προσομοίωση δημιουργήστε δικό σας testbench για να επαληθεύσετε όλες τις περιπτώσεις λειτουργίας της σχεδίασης.

Προεργασία (20%)

Πριν την προσέλευση σας στο εργαστήριο να έχετε
α) υπολογίσει τις συναρτήσεις, β) σχεδιάσει το σχετικό
κύκλωμα (block diagrams), και γ) υλοποιήσει σε γλώσσα VHDL
τα παρακάτω κυκλώματα.

Κύκλωμα 1

Ζητούμενα

Να σχεδιάσετε και να υλοποιήσετε κύκλωμα που έχει εισόδους και εξόδους όπως στον Πίνακα 1.

Όνομα	in/out	Πλάτος σε bit	Αντιστοίχιση στο Board
IN0	in	1	SW0
IN1	in	1	SW1
BTN0	in	1	BTN0
BTN1	in	1	BTN1
BTN2	in	1	BTN2
BTN3	in	1	BTN3
LED	out	6	LD[5:0]

Πίνακας 1: Είσοδοι – έξοδοι κυκλώματος

Το κύκλωμα λειτουργεί ως εξής:

1) Το LED[0] είναι το αποτέλεσμα της IN0 **NAND** IN1, αν και μόνο αν το BTN0 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

2) Το LED[1] είναι το αποτέλεσμα της IN0 **XOR** IN1 αν και μόνο αν το BTN1 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

3) Το LED[2] είναι το αποτέλεσμα της IN0 **OR** IN1 αν και μόνο αν το BTN2 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

4) Το LED[3] είναι το αποτέλεσμα της **NOT** IN0 αν και μόνο αν το BTN3 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

5) Οι έξοδοι των LED[4] και LED[5] συνδέονται απευθείας με τις εισόδους IN0 και IN1 αντίστοιχα.

Κύκλωμα 2

Ζητούμενα

Να σχεδιάσετε και να υλοποιήσετε κύκλωμα ημιαθροιστή (Half Adder) με τη χρήση λογικών πυλών. Έπειτα με χρήση του ημιαθροιστή ως υποκύκλωμα, υλοποιήστε πλήρη αθροιστή (Full Adder). Κάθε είσοδος στον πλήρη αθροιστή είναι 1 bit (Full Adder). Κάνετε χρήση των εντολών component - port map. Οι εισοδοί/έξοδοι φαίνονται στον Πίνακα 2.

Όνομα	in/out	Πλάτος σε bit	Αντιστοίχιση στο board
IN0	in	1	SW0
IN1	in	1	SW1
IN2	in	1	SW2
LED	out	2	LD[7:6]

Πίνακας 2: Είσοδοι - έξοδοι κυκλώματος

Παρατηρήσεις/Σημειώσεις

(1) Τα κυκλώματα της άσκησης αυτής είναι συνδυαστικά.

(2) Επαληθεύστε τη λειτουργία κάθε υποκυκλώματος αφού το σχεδιάσετε ξεχωριστά, με χρήση ξεχωριστού testbench, δηλ. για κάθε πύλη φτιάξτε ένα testbench και προσομοιώστε την. Το ίδιο κάνετε για τον ημιαθροιστή, και για τον πλήρη αθροιστή.

(3) Η σύνδεση των modules γίνεται ιεραρχικά, δηλ. έχουμε modules που συνδέονται μεταξύ τους μέσα σε άλλο module που βρίσκεται σε 1 παραπάνω επίπεδο. Αυτό με τη σειρά του μπορεί να συνδέεται με άλλα modules, σε module που βρίσκεται σε ακόμη παραπάνω επίπεδο κ.ο.κ. Σκεφτείτε πως θα συνδέσετε half-adders για τη δημιουργία του full-adder! Ακόμα και αν δεν συνδέονται μεταξύ τους τα modules, χρειάζεται να φτιάξετε 1 module σε 1 παραπάνω επίπεδο το οποίο περιέχει τα χαμηλότερα σε ιεραρχία modules, ή, τα instances τους (αυτό συμβαίνει στις περισσότερες

περιπτώσεις).

(4) Δημιουργήστε `topLevel` στο οποίο θα συνδέσετε όλα τα σήματα που θέλετε να «δείτε» στην προσομοίωση. Φτιάξτε το τελικό `testbench`, και συνδέστε το με το `topLevel`.

(5) Σε κάθε κύκλωμα που θα σχεδιάζετε, για να επικοινωνήσει με τον εξωτερικό κόσμο, δηλ. I/O π.χ. `buttons`, `switches`, `LEDs`, απαιτείται η δημιουργία του αρχείου `ucf` (`user constraint file`). Εκεί «συνδέουμε» όσα σήματα της σχεδίασής μας χρειάζονται. Θα σας δώσουμε ένα υπόδειγμα `ucf`, όμως εσείς πρέπει να φτιάξετε το δικό σας.

(6) Λεπτομέρειες για το αναπτυξιακό `Basys2` υπάρχουν στο link:

https://reference.digilentinc.com/media/basys2:basys2_rm.pdf

Παραδοτέα:

Πηγαίος κώδικας VHDL, κυματομορφές προσομοίωσης, παρουσίαση κυκλώματος, επίσης να δείξετε τη διαδικασία λύσης, τι χρησιμοποιήσατε από τη θεωρία και πως, π.χ. πίνακας καταστάσεων.

Βαθμολογία: Τα κυκλώματα είναι ισοδύναμα βαθμολογικά

Διεξαγωγή εργαστηρίου	Σύνολο: 70%
	Προετοιμασία: 20%
	Προσομοίωση: 30%
	Ορθή λειτουργία κυκλώματος στο Board: 20%
Αναφορά	30%

ΠΡΟΣΟΧΗ!

- 1) Η προεργασία να είναι σε ηλεκτρονική μορφή, σε αρχείο pdf.
- 2) Η έλλειψη προετοιμασίας οδηγεί σε απόρριψη.
- 3) Η διαπίστωση αντιγραφής σε οποιοδήποτε σκέλος της άσκησης οδηγεί στην απόρριψη από το σύνολο των εργαστηριακών ασκήσεων. Αυτό γίνεται οποιαδήποτε στιγμή στη διάρκεια του εξαμήνου.
- 4) Ο βαθμός της αναφοράς μετράει στον τελικό βαθμό εργαστηρίου μόνο αν ο βαθμός διεξαγωγής είναι (35/70)%.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ! ☺