



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:
ΗΡΥ 203 - ΠΡΟΧΩΡΗΜΕΝΗ ΛΟΓΙΚΗ ΣΧΕΔΙΑΣΗ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2021

Εργαστήριο 1

ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΗ ΓΛΩΣΣΑ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ VHDL ΚΑΙ
ΤΗΝ ΙΕΡΑΡΧΙΚΗ ΣΧΕΔΙΑΣΗ

ΕΚΠΟΝΗΣΗ: Καθ. Α. Δόλλας, Δρ. Ε. Σωτηριάδης

ΣΥΝΕΡΓΑΤΕΣ: Δρ. Ευριπίδης Σωτηριάδης, Μ. Κιμιωνής,

ΕΚΔΟΣΗ : 12.0

Χανιά

Σκοπός - Βήματα

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). Επαληθεύσετε ότι το κύκλωμα λειτουργεί σύμφωνα με ότι παρατηρήσατε στην προσομοίωση. Για την προσομοίωση δημιουργήσετε δικό σας αρχείο testbench για να επαληθεύσετε όλες τις περιπτώσεις λειτουργίας της σχεδίασης. Τα βήματα της σχεδιαστικής ροής, δηλαδή της (1) μετάφρασης (**translation**), (2) αποτύπωσης (**map**), (3) της τοποθέτησης και διασύνδεσης (**place and route**) και (4) της δημιουργίας του αρχείου προγραμματισμού (.bitfile) της αναδιατασσόμενης συσκευής (FPGA), δεν θα τα κάνετε στο Εργαστήριο.

Μέρος Α - Κύκλωμα 1

Ζητούμενα

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

Όνομα Σήματος	in/out	Πλάτος σε bit
A	in	1
B	in	1
C0	in	1
C1	in	1
C2	in	1
C3	in	1
C4	in	1
C5	in	1
Result	out	6

Πίνακας 1: Είσοδοι - Εξοδοι κυκλώματος

Το κύκλωμα λειτουργεί ως εξής:

1) Το RESULT[0] είναι το αποτέλεσμα της A **NAND** B, αν το C0 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

2) Το RESULT[1] είναι το αποτέλεσμα της A **NOR** B αν το C1 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

3) Το RESULT[2] είναι το αποτέλεσμα της A **AND** B αν το C2 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

4) Το RESULT[3] είναι το αποτέλεσμα της A **XOR** B αν το C3 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

5) Το RESULT[4] είναι το αποτέλεσμα της (A **AND** B) **OR** (A' **AND** B') αν το C4 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

6) Το RESULT[5] είναι το αποτέλεσμα της (A' **AND** B) **XOR** (A' **OR** B) αν το C5 είναι πατημένο (δηλαδή '1'), διαφορετικά είναι '0'.

Για την επίλυση του προβλήματος απαιτείται να κάνετε τα παρακάτω βήματα στη διαδικασία σχεδίασης :

- α) να υπολογίσετε τις συναρτήσεις στο χαρτί
- β) σχεδιάσει το σχετικό κύκλωμα ή block diagram,
- γ) να υλοποιήσετε σε γλώσσα VHDL τα παραπάνω κυκλώματα, με σωστή τεκμηρίωση (σχόλια) στα Αγγλικά
- δ) να δημιουργήσετε εξαντλητικό testbench για κάθε κύκλωμα και να έχετε επαληθεύσει την λειτουργία των κυκλωμάτων σας.

Μέρος Β - Κύκλωμα 2

Ζητούμενα

Να υλοποιήσετε το κύκλωμα ημιαθροιστή (HalfAdder) με τη χρήση λογικών πυλών. Έπειτα με χρήση του ημιαθροιστή ως υποκύκλωμα, υλοποιήσετε τον πλήρη αθροιστή (FullAdder) 1 bit, δηλαδή με εισόδους A, B, Cin και εξόδους S, Cout. Στη συνέχεια με τη χρήση του πλήρους αθροιστή 1 bit να σχεδιάσετε μόνοι σας έναν αθροιστή 2 bit. Κάνετε χρήση των εντολών component - portmap. Οι είσοδοι/έξοδοι του τελικού κυκλώματος που αθροίζει τελεστέους 2 Bit φαίνονται στον Πίνακα 2.

Όνομα	in/out	Πλάτος σε bit
A	In	2
B	in	2
Cin	in	1
RESULT	out	2
Cout	out	1

Πίνακας 2: Είσοδοι - Έξοδοι κυκλώματος

Η διαδικασία σχεδίασης είναι όπως στο Μέρος Α

Παρατηρήσεις/Σημειώσεις

- (1) Τα κυκλώματα της άσκησης αυτής είναι συνδυαστικά.
- (2) Επαληθεύσετε 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) Δημιουργήσετε topLevel στο οποίο θα συνδέσετε όλα τα σήματα που θέλετε να «δείτε» στην προσομοίωση. Φτιάξτε το τελικό testbench, και συνδέστε το με το topLevel.

Παραδοτέα:

Ένα αρχείο .zip (ή ανάλογο) με τα παρακάτω αρχεία

- 1) Κυκλώματα και block diagrams σε ηλεκτρονική μορφή (όχι φωτογραφίες χειρόγραφων) σε μορφή αρχείου .pdf
- 2) Κώδικες VHDL (όχι ολόκληρο το project της Xilinx) σε μορφή αρχείου .vhd
- 3) Screenshots από τα simulations και σχόλια που εξηγούν γιατί το simulation είναι εξαντλητικό.

Βαθμολογία: Τα δύο κυκλώματα είναι ισοδύναμα βαθμολογικά

Κυκλώματα και Block diagrams :20%
Κώδικες :50%
Προσομοίωση : 30%

ΠΡΟΣΟΧΗ!

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. Ως αντιγραφή νοείται και μέρος της αναφοράς, π.χ. σχήματα.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ! ☺