



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 202 ΨΗΦΙΑΚΑ ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2023-2024

Εργαστήριο 1

**ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΗ ΓΛΩΣΣΑ ΠΕΡΙΓΡΑΦΗΣ ΥΛΙΚΟΥ VHDL
ΣΥΜΠΕΡΙΦΕΡΙΚΗ ΣΧΕΔΙΑΣΗ ΣΕ ΑΚΟΛΟΥΘΙΑΚΑ
ΚΥΚΛΩΜΑΤΑ
(BEHAVIORAL DESIGN OF SEQUENTIAL CIRCUITS)**

ΕΚΠΟΝΗΣΗ: Δρ. Ε.Σωτηριάδης, Ι.Πολογιώργη, Αν. Καθ. Σ.Ιωαννίδης

**ΕΚΔΟΣΗ : 2.0
Χανιά**

Σκοπός

Είναι η εξοικείωση με τη γλώσσα VHDL και πιο συγκεκριμένα με την υλοποίηση ακολουθιακών κυκλωμάτων με behavioral (συμπεριφερικό) τρόπο σχεδίασης. Θα σχεδιάσετε έναν «παράξενο» μετρητή ο οποίος θα μετρά ανάλογα με την είσοδο ελέγχου που θα δέχεται κατά -5,-2,0,1,2,5,6,12 σε σχέση με την παρούσα του κατάσταση.

ΠΡΟΣΟΧΗ: το κύκλωμα που θα σχεδιάσετε, εκτός από ότι μετράει διαφορετικά «βήματα» σε σχέση με ένα απλό μετρητή που όταν μετράει πηγαίνει +1 ή -1 από την προηγούμενη κατάσταση, έχει και άλλες ιδιότητες που δεν τις έχουν οι κανονικοί μετρητές, όπως την έννοια του Overflow, το σταμάτημα (πάγωμα) του μετρήματος, κλπ. Μην μπερδέψετε αυτές τις «ειδικές» ιδιότητες με το πως είναι οι συμβατικοί μετρητές (που «ρολάρουν» και προς τα πάνω και προς τα κάτω, και δεν σταματούν το μέτρημα).

Μέρος του σκοπού του Εργαστηρίου 1 είναι να δείτε πόσο εύκολο είναι με τα κατάλληλα εργαλεία και χρησιμοποιώντας συμπεριφερική (behavioral) VHDL να σχεδιάσουμε κυκλώματα που με όρους Λογικής Σχεδίασης (διάγραμμα καταστάσεων, πίνακες Karnaugh, κλπ.) θα ήταν απλά αδύνατο να σχεδιάσουμε. Κάποιες από τις ιδιότητες του παράξενου αυτού μετρητή είναι όντως χρήσιμες κατά περίπτωση, δηλαδή το να παγώνει όταν **πρόκειται** να υπερχειλίσει θα μας ήταν απαραίτητο σε ένα μετρητή που δείχνει το κορυφαίο στοιχείο μίας στοίβας, όταν η στοίβα γεμίσει. Άρα, κατά περίπτωση κάποιες από τις ασκήσεις ή μεμονωμένα στοιχεία τους, εδώ, αντιστοιχούν σε πραγματικά, χρήσιμα κυκλώματα. Κάτι αντίστοιχο επίσης θα μπορούσαμε να σκεφτούμε και για τον μετρητή προγράμματος (PC), που άλλες φορές ανεβαίνει κατά 4, άλλες του προσθέτουμε μια προσημασμένη τιμή (branches), και άλλες τον θέτουμε σε συγκεκριμένη τιμή (jumps). Για να είναι πιο απλό το εργαστήριο, εδώ επιτρέπουμε να προκύψει η υπερχειλίση πριν παγώσουμε την λειτουργία με ένδειξη Overflow, Underflow, καθώς αυτό δημιουργεί μία και μόνο, απλή συνθήκη για υλοποίηση.

Προετοιμασία

Πριν την προσέλευση σας στο εργαστήριο **να έχετε υλοποιήσει σε γλώσσα VHDL το ζητούμενο κύκλωμα.**

Ζητούμενα

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

Όνομα	in/out	Πλάτος σε bit
Clk	είσοδος	1
RST	είσοδος	1
Control	είσοδος	3
Count	έξοδος	8
Overflow	έξοδος	1
Underflow	έξοδος	1
Valid	έξοδος	1

Πίνακας 1: Είσοδοι - έξοδοι κυκλώματος

Το κύκλωμα λειτουργεί ως εξής:

Ο παράξενος μετρητής μετράει από το 0 έως το 255, δεν παίρνει αρνητικές τιμές ούτε τιμές μεγαλύτερες του 255. Εάν πάρει τιμές εκτός της περιοχής [0,255] τότε ανάβει το αντίστοιχο σήμα εξόδου (underflow, overflow) και παγώνει η λειτουργία του μέχρι να έρθει RST, και η έξοδος Valid γίνεται 0.

Το Clk είναι απαραίτητο σε κάθε ακολουθιακό κύκλωμα και είναι αυτόνομο - δε το χρησιμοποιούμε ποτέ σε συνδυασμό με κανένα σήμα.

Το RST δίνει στο μετρητή την τιμή 0, είναι σύγχρονο (δηλαδή δουλεύει με σχέση με το ρολόι) και επίσης δε το χρησιμοποιούμε ποτέ σε συνδυασμό με κανένα σήμα. Είναι κατά συνθήκη ενεργό στο «1». (Υπάρχει και ασύγχρονο reset, που επίσης δουλεύει μια χαρά, αλλά δε θέλουμε αυτό τώρα.)

Το Control είναι είσοδος 3 bit που μας δείχνει τις 8 διαφορετικές περιπτώσεις που θα μετρήσει ο μετρητής ως εξής:

Τιμή Εισόδου Control	Βήμα Μέτρησης
0	-5
1	-2
2	0
3	1
4	2
5	5
6	6
7	12

Πίνακας 2: Αντιστοίχιση εισόδου με βήμα μέτρησης

Το Count είναι η έξοδος του μετρητή, και η οποία ισχύει (είναι σωστή και θα έπρεπε να λαμβάνεται υπόψη, π.χ. από ένα άλλο κύκλωμα) όσο το Valid είναι 1.

Οδηγίες για τη δημιουργία testbench: Για τη συγκεκριμένη άσκηση θα ακολουθήσετε τα εξής

- 1) Η προσομοίωση θα ξεκινάει με το σήμα RST ενεργό, το οποίο και θα παραμένει ενεργό για τουλάχιστον 10 κύκλους.
- 2) Θα πρέπει να δείξετε ότι όταν γίνεται overflow παγώνει το σύστημα και μετά από ενεργοποίηση του RST επανέρχεται στο 0 και ξεκινάει ξανά.
- 3) Θα πρέπει να δείξετε ότι όταν γίνεται underflow παγώνει το σύστημα και μετά από ενεργοποίηση του RST επανέρχεται στο 0 και ξεκινάει ξανά.
- 4) Θα πρέπει να δείξετε ότι όλα τα βήματα μέτρησης είναι λειτουργικά και μπορούμε να αλλάζουμε βήμα μέτρησης από κύκλο σε κύκλο.

Παρατηρήσεις/Σημειώσεις

(1) Επαληθεύσετε τη λειτουργία του κυκλώματος ώστε να είσαστε σίγουροι ότι λειτουργεί, με χρήση testbench. Γιατί δεν είναι εφικτό στα πλαίσια εργαστηριακής άσκησης να γίνει εξαντλητικά;

(2) Είναι απαραίτητη η χρήση βιβλιοθηκών αριθμητικής για την εύκολη υλοποίηση του κυκλώματος. Πρέπει να ψάξετε να βρείτε τις κατάλληλες βιβλιοθήκες.

(3) Στην άσκηση αυτή υπάρχει μόνο top level σχεδίαση στην οποία πρέπει να γίνει προσεκτική χρήση της εντολής if. Η χρήση των if πρέπει να είναι πάντοτε σε φωλιασμένα statements.

Παραδοτέα:

Πηγαίος κώδικας VHDL, κυματομορφές προσομοίωσης θα επιδειχθούν στο εργαστήριο, παρουσίαση κυκλώματος, καθώς επίσης και να δείξετε τη διαδικασία λύσης, τι χρησιμοποίησατε από τη θεωρία και πως, π.χ. πίνακας καταστάσεων.

Βαθμολογία:

Διεξαγωγή εργαστηρίου	Προετοιμασία:30%
	Προσομοίωση: 70%

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.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!