



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

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

ΕΑΡΙΝΟΕΞΑΜΗΝΟ 2021

Εργαστήριο 5

ΣΧΕΔΙΑΣΗ ΕΝΟΣ ΑΠΛΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΛΕΓΧΟΥ ΜΝΗΜΗΣ – ΜΙΑ
ΑΠΛΗ ΣΤΟΙΒΑ

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

ΣΥΝΕΡΓΑΤΕΣ: Δρ. Ευριπίδης Σωτηριάδης, Μ. Κιμιωνή

ΕΚΔΟΣΗ : 12.0
Χανιά

Σκοπός

Είναι η μοντελοποίηση και υλοποίηση μιας στοίβας με δυνατότητα εντολών push και pop 4-bit αριθμών.

Ζητούμενα

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

Όνομα	in/out	Πλάτος σε bit	Αντιστοίχιση στο Board
CLK	In	1	Clk
RST	In	1	Btn0
Push	In	1	Btn1
Pop	In	1	Btn2
NumberIN	In	4	SW[3:0]
NumberOUT	Out	4	LD[3:0]
Empty	Out	1	LD[4]
Full	Out	1	LD[5]
AlmostEmpty	Out	1	LD[6]
AlmostFull	Out	1	LD[7]

Πίνακας 1: Είσοδοι - έξοδοι του κυκλώματος

Προδιαγραφές

Το κύκλωμα θα έχει μία μνήμη πλάτους 4-bit και βάθους 12 θέσεων μνήμης. Η στοίβα θα λειτουργεί με τον κατάλληλο έλεγχο ως στοίβα 10 στοιχείων. Κάθε αριθμός που είναι στην είσοδο θα αποθηκεύεται στην στοίβα εφόσον πατηθεί το πλήκτρο Push. Κάθε φορά που θα πατηθεί το πλήκτρο Pop θα διαγράφεται ο αριθμός στην κεφαλή της στοίβας (στην ουσία δεν θα έχουμε διαγραφή αλλά μετακίνηση του Top of Stack - TOS). Η έξοδος της στοίβας NumberOUT θα είναι η κεφαλή της στοίβας (TOS) εκτός αν είναι άδεια οπότε θα είναι η θέση 0 της μνήμης. Όταν η στοίβα είναι άδεια θα ανάβει η έξοδος Empty και κάθε φορά που είναι γεμάτη, δηλαδή έχει 10 στοιχεία, θα ανάβει η έξοδος Full. Όταν η στοίβα είναι άδεια δεν θα μπορεί να αφαιρέσει κανείς στοιχεία (Pop) και όταν είναι γεμάτη (Full) δε θα μπορεί να προσθέσει στοιχεία (Push), αντίστοιχα. Προσοχή: Στις στοίβες με n στοιχεία μνήμης μπορούμε να αποθηκεύσουμε (n-1) δεδομένα γιατί διαφορετικά μας «ξεφεύγουν» οι pointers. Στο παρόν εργαστήριο λύνουμε το πρόβλημα με το να φτιάξουμε μία μνήμη βάθους 12 θέσεων αλλά χρησιμοποιώντας μόνο 11 από αυτές (για 10 χρήσιμα στοιχεία: θέση 0 για άδεια στοίβα, θέσεις 1-10 για αποθήκευση, θέση 11 αχρησιμοποίητη). Επί πλέον από τα παραπάνω σήματα έχουμε και τις εξόδους AlmostEmpty και AlmostFull, και οι οποίες γίνονται 1 εφόσον έχουμε αποθηκευμένα (αντίστοιχα) 1-3 ή 8-10 δεδομένα.

Μας συμφέρει η στοίβα να είναι pre-increment/post-decrement γιατί ο TOS pointer ανά πάσα στιγμή είναι η διεύθυνση της μνήμης που δείχνουμε στην οθόνη (σε post-increment/pre-decrement θέλουμε δύο pointers, ένα για το TOS και ένα για την θέση που θα δείχνει η οθόνη). Δείτε τις σχετικές σημειώσεις του μαθήματος. Για ευκολία δεν σας ζητάμε να υλοποιήσετε το κύκλωμα λειτουργίας Onfl/Unfl σε περίπτωση που προσπαθήσουμε να κάνουμε Push σε γεμάτη στοίβα ή Pop από άδεια στοίβα.

Παρατηρήσεις/Σημειώσεις

- (1) Η μνήμη θα δημιουργηθεί αυτόματα από τα εργαλεία της Xilinx
- (2) Ο έλεγχος της μνήμης θα γίνει από μία FSM.
- (3) Η FSM μαζί με τη μνήμη θα ενωθούν με ένα toplevel αρχείο.

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 επαρκές.

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.