



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

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017

## Εργαστήριο 4

### ΑΝΑΓΝΩΡΙΣΗ ΠΡΑΞΕΩΝ ΓΙΑ ΣΧΕΔΙΑΣΗ ΜΙΑΣ ΑΡΙΘΜΟΜΗΧΑΝΗΣ

ΕΚΠΟΝΗΣΗ : Καθ. Α. Δόλλας

ΕΔΙΠ: Μ. Κιμιωνής  
Δρ. Κ. Παπαδημητρίου  
ΒΟΗΘΟΙ: Π. Μαλακωνάκης  
Ι. Κοϊδής  
Κ. Μαλαβάζος  
Π. Γιακουμάκης  
Μ. Πισσαδάκης

ΕΚΔΟΣΗ : 1.0 (Εαρινό εξάμηνο 2017)

Χανιά 2017

## **Σκοπός των Εργαστηρίων 4 και 5**

Στα εργαστήρια αυτά θα δημιουργήσουμε τις βασικές δομές μίας αριθμομηχανής (calculator) που λειτουργεί με την μέθοδο Reverse Polish Notation – RPN, δηλαδή την μέθοδο που χρησιμοποιήθηκε ευρύτατα σε αριθμομηχανές της εταιρίας Hewlett-Packard κατά τις δεκαετίες 1970 – 1990. Οι αριθμομηχανές αυτές δεν έχουν πλήκτρο “=” αλλά κάνουν πράξη στα δύο κορυφαία στοιχεία της στοίβας. Ενδεικτικά, η πράξη  $5+3$  γίνεται ως Push 5, Push 3, +. Δεν θα υλοποιήσουμε πολλές πράξεις, και όλες οι πράξεις θα έχουν τελεστέους 8-Bit και αποτέλεσμα επίσης 8-Bit (αλλιώς έχουμε υπερχειλίση). Θα επεκτείνουμε την σχεδίαση των Εργαστηρίων 1-3 (α) με δυνατότητα αναγνώρισης πράξεων (Εργ. 4) και (β) με υλοποίηση των πράξεων αυτών (Εργ. 5). Είναι σημαντικό οι σχεδιάσεις να είναι καλά δομημένες και ιεραρχικές ώστε να αποσφαλματώνονται και να επεκτείνονται εύκολα.

## **Σκοπός του Εργαστηρίου 4**

Είναι η εξέλιξη της στοίβας προς την κατεύθυνση του να γίνει μία αριθμομηχανή. Οι αριθμητικές και λογικές πράξεις που θα υποστηρίζονται τελικά είναι επτά (περιλαμβανομένων και των Push/Pop):

1. Push
2. Pop
3. Πρόσθεση αριθμών 2's complement ( $TOS + "TOS-1"$ )
4. Αφαίρεση αριθμών 2's complement ( $TOS - "TOS-1"$ )
5. Μοναδιαία αφαίρεση 2's complement (unary subtraction,  $-TOS$ )
6. Εναλλαγή TOS με TOS-1 (γνωστή και σαν  $X<>Y$ )

## **Προετοιμασία**

Κατά την προσέλευση σας στο εργαστήριο θα πρέπει:

α) Να έχετε σχεδιάσει το block diagram της σχεδίασης, το σχήμα της μνήμης, και το αναλυτικό διάγραμμα της FSM για αναγνώριση πλήκτρων. Προφανώς η FSM μπορεί να είναι ιεραρχική, αν αυτό εξυπηρετεί την σχεδίαση.

β) Θα πρέπει να έχετε υλοποιήσει σε κώδικα VHDL το παρακάτω κύκλωμα, μαζί με τα κατάλληλα testbench.

## Ζητούμενα

Να σχεδιάσετε και να υλοποιήσετε ένα κύκλωμα που έχει εισόδους και εξόδους όπως στον Πίνακα 1, επεκτείνοντας την λειτουργικότητα του Εργαστηρίου 3.

Όνομα	Είσοδος/Εξοδος	Πλάτος σε bit	Αντιστοίχιση στο Board
Push	Είσοδος	1	BTN0
Pop	Είσοδος	1	BTN1
Mode	Είσοδος	1	BTN2
Reset	Είσοδος	1	BTN3
Clock	Είσοδος	1	MCLK
Num In	Είσοδος	8	SW7-SW0
Num Out	Εξοδος	8	LED7-LED0
SSD En	Εξοδος	4	AN3-AN0
Empty	Εξοδος	1 -> 7	SEG6-SEG0 (E)
Full	Εξοδος	1 -> 7	SEG6-SEG0 (F)
Stack Ovf	Εξοδος	1 -> 7	SEG6-SEG0 (OVF)

Πίνακας 1: Είσοδοι - έξοδοι του κυκλώματος

## Περιγραφή κυκλώματος

Στο Εργαστήριο 4 θα κάνουμε μόνο την αναγνώριση της επιθυμητής πράξης. Το προφανές πρόβλημα που θα λύσουμε είναι το πως θα αναγνωρίσουμε την επιθυμητή πράξη όταν έχουμε μόνο 4 Πλήκτρα (BTN0-BTN3) και επτά πράξεις - τα δεδομένα θα μπαίνουν όπως στο Εργαστήριο 3. Αυτό θα γίνει με την χρήση του BTN2 σαν MODE, και με **διαδοχικά** πατήματα αυτού να αναγνωρίσουμε την επιθυμητή πράξη. Η απεικόνιση πράξεων σε αλληλουχία πλήκτρων είναι η εξής:

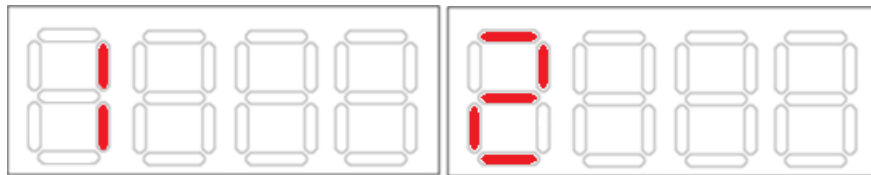
Λειτουργία	1° Πάτημα Πλήκτρου	2° Πάτημα Πλήκτρου	3° Πάτημα Πλήκτρου
	Mode 0	Mode 1	Mode 2
Push	BTN0		
Pop	BTN1		
Add		BTN0	
Sub		BTN1	
Unary Sub			BTN0
X <> Y			BTN1
Μετάβαση στο Mode 1	BTN2		
Μετάβαση στο Mode 2		BTN2	
Επαναφορά στο Mode0			BTN2
Reset	BTN3	BTN3	BTN3

Παρατηρούμε ότι αν πατήσουμε στο Mode 0 το BTN2 κατά λάθος και μας βάλει στο Mode 1, αρκεί να το πατήσουμε ακόμη **δύο φορές** ώστε να επανέλθουμε στο Mode 0. Το Reset λειτουργεί πάντα, ακόμη και όταν είμαστε σε Mode0 - Mode2, κλπ.

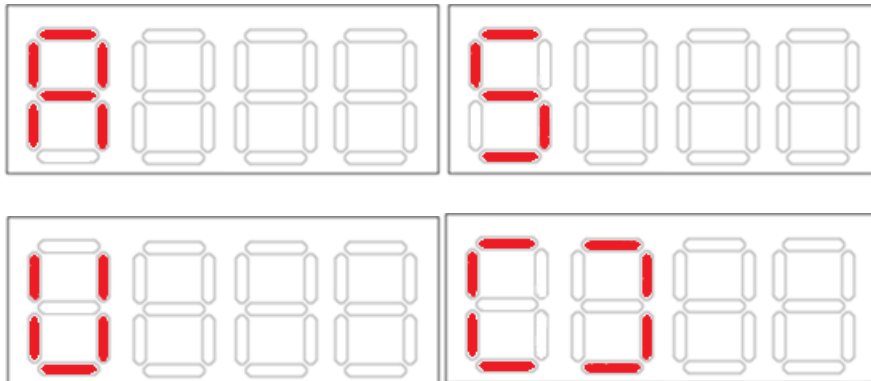
Για την υλοποίηση του Εργ. 4 θα γίνουν δύο μέρη, ένα που αφορά την ίδια την αναγνώριση και ένα που αφορά την παρουσίαση στην οθόνη 7-segment LED του τι αναγνωρίστηκε. Θα χρησιμοποιήσετε το κύκλωμα debounce των διακοπών. Στο Mode 0 δεν αλλάζει τίποτα από το Εργαστήριο 3. Στα Mode 1 - 2 θα δείξετε στο αριστερότερο 7-segment LED τους αριθμούς 1 και 2 αντίστοιχα.

### Σημάνσεις σχετικά με το Seven Segment Display

1) Για τα Mode 1 και Mode 2 έχουμε την εξής οθόνη:



2) Για τις πράξεις Add, Sub, Unary Sub, και  $X <> Y$  έχουμε αντίστοιχα τις οθόνες:



3) Όλες οι οθόνες που είχαμε κάνει στο Εργαστήριο 3 και η αντίστοιχη λειτουργικότητα, LED, κλπ. ισχύουν επίσης.

### **Παρατηρήσεις / Σημειώσεις**

- Για τα κουμπιά BTN0-BTN3 θα πρέπει να χρησιμοποιήσετε τα module debounce και test\_button τα οποία θα βρείτε

στο "Υλικό του Εργαστηρίου".

- Η σχεδίαση θα πρέπει να είναι καλά δομημένη. Κάθε μία από τις πράξεις που αναγνωρίζετε στο επόμενο εργαστήριο θα υλοποιηθεί με διαφορετική μηχανή πεπερασμένων καταστάσεων, όσο πιο καθαρή είναι η σχεδίασή σας στο παρόν εργαστήριο τόσο πιο εύκολη θα είναι η υλοποίηση του επόμενου.

**Παραδοτέα:** Πηγαίος κώδικας VHDL, κυματομορφές προσομοίωσης, παρουσίαση κυκλώματος.

**Βαθμολογία:**

Διεξαγωγή εργαστηρίου	<b>70%</b>
	Προετοιμασία 20%
	Προσομοίωση 30%
	Σωστή λειτουργία του κυκλώματος στο Board 20%
Αναφορές	<b>30%</b>

**ΠΡΟΣΟΧΗ!**

1) Η έλλειψη προετοιμασίας οδηγεί στην απόρριψη στη συγκεκριμένη εργαστηριακή άσκηση.

2) Η διαπίστωση αντιγραφής σε οποιοδήποτε σκέλος της άσκησης ή της αναφοράς οδηγεί στην άμεση απόρριψη από το σύνολο των εργαστηριακών ασκήσεων (άρα και από το μάθημα).

3) Μην ξεχάσετε το testbench!!!

4) Ο βαθμός της αναφοράς μετράει στον τελικό βαθμό του εργαστηρίου μόνο αν ο βαθμός της διεξαγωγής του εργαστηρίου είναι (35/70)%.

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ! ☺**