



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

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017

Εργαστήριο 3

ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ POST-INCREMENT, PRE-DECREMENT ΣΤΟΙΒΑΣ ΣΕ VHDL

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

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

ΕΚΔΟΣΗ : 10.0 (Εαρινό εξάμηνο 2017)

Χανιά 2017

Σκοπός του Εργαστηρίου

Είναι η περαιτέρω εξοικείωση με την γλώσσα VHDL σε behavioral και structural μορφή. Σε αυτό το εργαστήριο θα σχεδιαστεί κύκλωμα το οποίο θα υλοποιεί την βασική λειτουργικότητα μιας **post-increment, pre-decrement** στοίβας. Το κύκλωμα θα εκτελεί μόνο τις απλές λειτουργίες push/pop, ενώ σε επόμενα εργαστήρια θα κάνουμε αριθμητικές πράξεις στα δύο κορυφαία στοιχεία της στοίβας που είναι στο TOS (Top of stack) και του προηγούμενου στοιχείου (TOS-1). Τέλος στα LED του αναπτυξιακού Basys 2 θα απεικονίζεται σε δυαδική μορφή το στοιχείο του TOS (δεν παίζει ρόλο για το παρόν εργαστήριο αλλά θα θεωρήσουμε ότι είναι 2's complement) ενώ στα 7segment display θα απεικονίζονται χρήσιμες πληροφορίες (π.χ. Empty, Full, OVf).

Προετοιμασία

Κατά την προσέλευση σας στο εργαστήριο θα πρέπει:

α) Να έχετε σχεδιάσει το block diagram της σχεδίασης, το σχήμα της μνήμης, αναλυτικό διάγραμμα της FSM στο οποίο θα φαίνονται αναλυτικά οι είσοδοι/έξοδοι, οι διάφορες μεταβάσεις και να μπορείτε να μας περιγράψετε πως η FSM αλληλεπιδρά με την μνήμη. Προφανώς η FSM μπορεί να είναι ιεραρχική, αν αυτό εξυπηρετεί την σχεδίαση.

β) Θα πρέπει να έχετε υλοποιήσει σε κώδικα VHDL το παρακάτω κύκλωμα.

Ζητούμενα

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

Όνομα	Είσοδος/Έξοδος	Πλάτος σε bit	Αντιστοίχιση στο Board
Push	Είσοδος	1	BTN0
Pop	Είσοδος	1	BTN1
(Not used yet)	Είσοδος	1	BTN2
Reset	Είσοδος	1	BTN3
Clock	Είσοδος	1	MCLK
Num In	Είσοδος	8	SW7-SW0
Num Out	Έξοδος	8	LED7-LED0
SSD_En	Έξοδος	4	AN3-AN0
Empty	Έξοδος	1 -> 7	SEG6-SEG0 (E)
Full	Έξοδος	1 -> 7	SEG6-SEG0 (F)
Stack OvF	Έξοδος	1 -> 7	SEG6-SEG0 (OVF)

Πίνακας 1: Είσοδοι - έξοδοι του κυκλώματος

Περιγραφή κυκλώματος

Το κύκλωμα αποτελείται από μία στοίβα των 32 στοιχείων (προσοχή: με 32 στοιχεία στην στοίβα μπορούμε να εισαγάγουμε 31 αριθμούς, όπως εξηγήσαμε στο μάθημα) πλάτους 8-bits η οποία μπορεί να εκτελεί τις εξής λειτουργίες:

α) Εισαγωγή ενός αριθμού των 8 bits στην στοίβα. **(Push)**

β) Αφαίρεση ενός αριθμού των 8 bits από την στοίβα (προσοχή: αφαίρεση από την στοίβα – όχι αφαίρεση σαν πράξη) . **(Pop)**

γ) Όταν η στοίβα υπερβεί το μέγιστο αριθμό στοιχείων μετά από διαδοχικά Push το σύστημα σταματάει να ανταποκρίνεται σε όλα τα σήματα εισόδου εκτός του σήματος αρχικοποίησης **Reset**. **(ένδειξη OVF)**

δ) όταν πατηθεί το **Reset** η στοίβα αρχικοποιείται, με το TOS να είναι στην θέση 0 και τα περιεχόμενα στην θέση αυτή να παίρνουν τιμή 00000000 (προσοχή: η στοίβα είναι άδεια, το 00000000 είναι αρχικοποίηση. Επειδή είναι Post-increment η θέση θα πανωγραφεί όταν κάνουμε push). ΔΕΝ χρειάζεται να αρχικοποιηθούν όλες οι θέσεις τις μνήμης γιατί θα πανωγράφονται, αλλά είναι καλή ιδέα να το δοκιμάσετε για άσκηση αφού κάνετε όλα τα άλλα μέρη.

Επιπλέον το κύκλωμα υποστηρίζει σημάνσεις που ενημερώνουν για την κατάσταση στην οποία βρίσκεται:

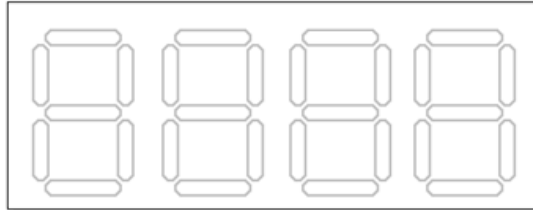
Σημάνσεις σχετικά με την στοίβα

Ο δυαδικός αριθμός που υπάρχει στους διακόπτες, εφόσον γίνει Push θα γραφτεί στο TOS. Προσοχή: επειδή είναι post-increment η στοίβα χρειαζόμαστε δύο μετρητές, ένα που να δείχνει που θα γραφτεί ο επόμενος αριθμός και ένα που να δείχνει το τελευταίο στοιχείο που γράψαμε! Οι δύο μετρητές διαφέρουν πάντα κατά μία θέση, αλλά δεν χρειάζεται να κάνουμε ένα μετρητή και ένα αθροιστή.

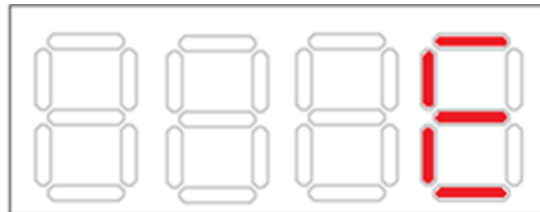
Βοήθημα: οι μνήμες μας έχουν δύο θύρες ανάγνωσης και μία θύρα εγγραφής. Πρακτικά αυτό σημαίνει ότι και τους δύο μετρητές, ένα για το τελευταίο στοιχείο που γράψαμε στο TOS και ένα για τον δείκτη (pointer) της θέσης όπου θα γίνει το επόμενο push μπορούμε να τους βάλουμε σε ξεχωριστή θύρα (για ανάγνωση και εγγραφή αντίστοιχα), και μας περισσεύει και μία ακόμη θύρα ανάγνωσης, που θα είναι χρήσιμη σε επόμενα εργαστήρια.

Σημάνσεις σχετικά με το Seven Segment Display

1) Όταν η στοίβα λειτουργεί αλλά δεν είναι άδεια, γεμάτη, ή σε υπερχείλιση το SSD είναι σβηστό:



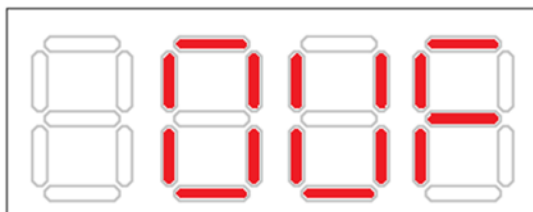
2) Όταν η στοίβα είναι άδεια (**Empty**) ανάβουν τα **SEG6-SEG0** που αντιστοιχούν στο **AN0** (δείχνουν **E**).



3) Όταν η στοίβα είναι γεμάτη (**Full**) ανάβουν τα **SEG6-SEG0** που αντιστοιχούν στο **AN0** (δείχνουν **F**).



4) Όταν η στοίβα υπερβεί το μέγιστο αριθμό στοιχείων (**Stack_Ovf**) ανάβουν τα **SEG6-SEG0** που αντιστοιχούν στα **AN2-AN0** (δείχνουν **OVF**).



Παρατηρήσεις / Σημειώσεις

- Για τα κουμπιά Push, Pop και Reset θα πρέπει να

χρησιμοποιήσετε τα module debounce και test_button τα οποία θα βρείτε στο "Υλικό του Εργαστηρίου".

- ΠΡΟΣΟΧΗ: Σύμφωνα με το block diagram που έχετε σχεδιάσει πρέπει να έχετε ένα αρχείο για κάθε διαφορετικό υποσύστημα.
- Η υλοποίηση της στοίβας θα γίνει αποκλειστικά με την χρήση μίας μνήμης BlockRam των τριάντα δύο (32) θέσεων και πλάτους των 8 bits.
- Η μνήμη θα υλοποιηθεί **ΥΠΟΧΡΕΩΤΙΚΑ** με το εργαλείο Core Generator.
- Τα E,F,OVF είναι 1-bit έκαστο αλλά αποκωδικοποιούνται στα 7-segment LED display.

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

Βαθμολογία:

Διεξαγωγή εργαστηρίου	70%
	Προετοιμασία 20%
	Προσομοίωση 30%
	Σωστή λειτουργία του κυκλώματος στο Board 20%
Αναφορές	30%

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! ☺