



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

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017

## Εργαστήριο 5

### ΣΧΕΔΙΑΣΗ ΜΙΑΣ ΑΠΛΗΣ 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 του Εργαστηρίου

Είναι η εξέλιξη των προηγούμενων εργαστηρίων ώστε να γίνει μία απλή αριθμομηχανή. Οι αριθμητικές και λογικές πράξεις που θα υποστηρίζονται τελικά είναι έξι (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, το TOS γίνεται  $-TOS$ )
6. Εναλλαγή TOS με TOS-1 (γνωστή και σαν  $X <> Y$ )

## Προετοιμασία

Κατά την προσέλευση σας στο εργαστήριο θα πρέπει:

α) Να έχετε σχεδιάσει **ιεραρχικά** το block diagram της σχεδίασης, το σχήμα της μνήμης, και το αναλυτικό διάγραμμα της FSM για αναγνώριση πλήκτρων και των δομών που κάνουν τις πράξεις. 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

Οι πράξεις που κάνει η αριθμομηχανή, αναγνώριση λειτουργίας, κλπ. είναι όπως στο Εργαστήριο 4. Στο παρόν εργαστήριο θα υλοποιήσουμε δύο απλές δομές datapath (προσθαφαιρέτης, προσωρινοί καταχωρητές), τις μηχανές πεπερασμένων καταστάσεων για τις αντίστοιχες πράξεις, και ένα απλό κύκλωμα εντοπισμού υπερχείλισης (Overflow) που οδηγεί στην ένδειξη OVF στα 7 segment LED που ήδη έχετε από το Εργ. 3.

## Περιγραφή κυκλώματος

Πιο συγκεκριμένα, όσον αφορά το **datapath** χρειάζεται να κάνετε:

1. Ένα κύκλωμα προσθαφαιρέτη, σε **structural** VHDL, υλοποιημένο με όποιο αλγόριθμο πρόσθεσης θέλετε. Η σχεδίαση είναι συνδυαστική (όχι ακολουθιακή) και μπορεί να βασίζεται στο Εργαστήριο 2 – στην πιο απλή μορφή παίρνετε δύο αντίγραφα του CLA που έχετε ήδη, ενώνετε το Cout του ενός στο Cin του δεύτερου και έτσι

δημιουργείτε ένα Group Ripple Carry Adder των 8 bit. Ακόμη και μία σχεδίαση αθροιστή με διάδοση κρατουμένου (RCA) είναι OK, αν την προτιμάτε. Μην χάσετε χρόνο σε μεγάλες βελτιστοποιήσεις (δεν βαθμολογείστε για αυτό), αλλά η σχεδίαση πρέπει να είναι πολύ καθαρή, οι κώδικες VHDL καλά δομημένοι σε structural VHDL και ιεραρχικοί όπου χρειάζεται, και κάποιο testbench πρέπει να δοκιμάζει κατάλληλα διαλεγμένες περιπτώσεις (βαθμολογείστε για αυτό).

2. Συμπληρωματικό κύκλωμα που μετατρέπει τον αθροιστή σε προσθαφαιρέτη (δεν είναι τίποτα άλλο από το να περνάει κάθε Bit του τελεστήου B από μία XOR, της οποίας η άλλη είσοδος συνδέεται με το σήμα ελέγχου ADD'/SUB, δηλ. «1» όταν πρόκειται για αφαίρεση και «0» όταν πρόκειται για πρόσθεση).
3. Συμπληρωματικό κύκλωμα που εντοπίζει υπερχείλιση στην πρόσθεση/αφαίρεση. Βοήθημα για την υπερχείλιση στην αφαίρεση: ουσιαστικά ισχύει ο κανόνας που ισχύει και στην πρόσθεση αλλά επί **ετερόσημων** αριθμών (άρα εκεί που θα είχαμε XNOR για την πρόσθεση έχουμε XOR για την αφαίρεση). Ολόκληρο το κύκλωμα για την υπερχείλιση μπορεί να εκφραστεί σε μία απλή λογική συνάρτηση άλγεβρας Boole και οδηγεί σε ένδειξη OVF.
4. Ένα απλό κύκλωμα όπου όταν κάνουμε Pop κρατάει σε εσωτερικούς καταχωρητές τα δύο τελευταία Pop.

Για το **control path** πρέπει να κάνετε μικρές μηχανές πεπερασμένων καταστάσεων για κάθε μία από τις πράξεις. Κάθε μία από τις πράξεις είναι μία **απλή** FSM, **λίγων** κύκλων. Μπορείτε να υλοποιήσετε τις FSM σε behavioral VHDL και να συνδέσετε την καθεμία με την αντίστοιχη αποκωδικοποίηση που έχετε από το Εργαστήριο 4. Πιο συγκεκριμένα:

1. Push/pop/Reset – είναι όπως στο Εργ. 3.
2. Add, Sub: Κάνουμε Pop το TOS και το TOS-1 στους προσωρινούς καταχωρητές που datapath (βλ. νούμερο 4, παραπάνω), και οι οποίοι είναι απ' ευθείας συνδεδεμένοι με τον προσθαφαιρέτη. Ένα κύκλο αργότερα κάνουμε Push στο TOS το αποτέλεσμα.
3. Unary Sub: Κάνουμε Pop μία φορά, στον επομενο κύκλο αντιγράφουμε τον προσωρινό καταχωρητή που είχε το TOS στον προσωρινό καταχωρητή που σε Add/Sub κρατάει το TOS-1, αρχικοποιώντας τον προσωρινο καταχωρητή που κρατάει το TOS σε 0. Το υπόλοιπο είναι όπως η αφαίρεση. Εναλλακτικά, σε ένα κύκλο Pop πάμε το TOS κατ' ευθείαν στον καταχωρητή που χρειάζεται και αρχικοποιούμε τον άλλο (με πολυπλέκτες και στις δύο περιπτώσεις).

## Παρατηρήσεις / Σημειώσεις

- Όλες οι FSM μπορούν να γίνουν με behavioral VHDL. Επίσης οι καταχωρητές ολίσθισης μπορούν να γίνουν σε behavioral VHDL.
- Η σχεδίαση θα πρέπει να είναι καλά δομημένη. Κάθε μία από τις πράξεις που αναγνωρίζετε πρέπει να οδηγεί σε μία FSM που την υλοποιεί. Μπορεί βέβαια μία FSM κατάλληλα σχεδιασμένη να υλοποιεί δύο πράξεις (ενδεικτικά, πρόσθεση και αφαίρεση).
- Το Reset λειτουργεί όπως στο Εργ. 3, και το OVF επίσης οδηγεί σε «κλείδωμα» του συστήματος, όπως στο Εργ. 3 (κατόπιν χρειάζεται Reset).

**Παραδοτέα:** Πηγαίος κώδικας VHDL με testbench, 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! ☺