

**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**  
**Αρχιτεκτονική Υπολογιστών**

**Εργαστηριακή Άσκηση 1 (25% εργαστηριακού βαθμού)**

**Ομαδική εργασία σε ομάδες τριών (3) μελών**

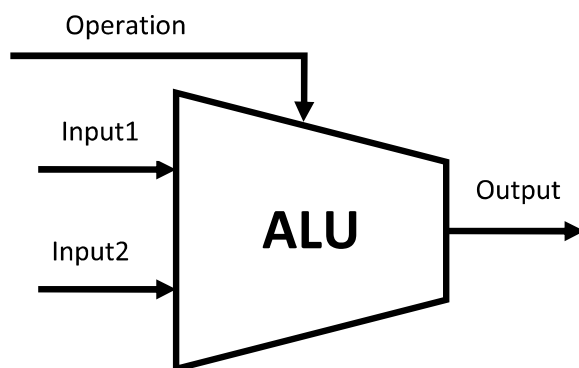
**Ημερ/νία παράδοσης: 10/4/2020, 12.00 μεσημέρι (μέσω e-class)**

- Για όλα τα προβλήματα να πραγματοποιήσετε functional και timing simulation για το πέμπτον διάστημα ώστε να επαληθεύσετε τη σωστή λειτουργία του κυκλώματος. Για κάθε κύκλωμα θα πρέπει να παραδίδετε το αρχείο VHDL, το RTL διάγραμμα του κυκλώματος (σε μορφή εικόνας) καθώς και το functional και timing simulation (σε μορφή εικόνας)
- Χρησιμοποιήστε τα ονόματα όταν σας δίνονται για input/output του κυκλώματος.
- Να έχετε (λίγα αλλά περιεκτικά) σχόλια στη VHDL όταν χρειάζεται να εξηγήσετε «δύσκολα» κομμάτια κώδικα, ρουτίνες, κλπ.
- Γράψτε (σε σχόλια στην αρχή του προγράμματος) **τα ονόματά σας και τους ΑΜ.**

## 1) ALU

α) Θα κατασκευάσετε με VHDL σε **structural κώδικα** (όχι σε behavioral, πρέπει να σχεδιάσετε πύλη-προς-πύλη το εργαστήριο αυτό χωρίς να χρησιμοποιήσετε δομικά στοιχεία του Quartus αλλά μόνο αυτά που θα ορίσετε εσείς) μια ALU για κάποιον επεξεργαστή.

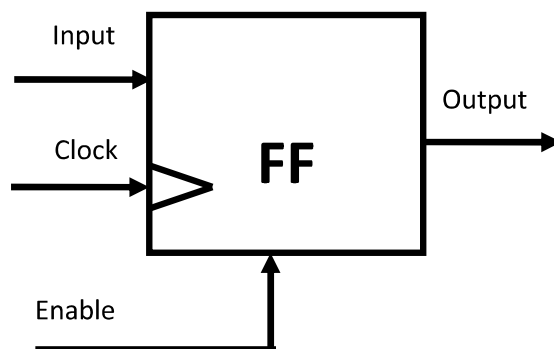
Το κύκλωμα θα παίρνει σαν είσοδο δύο 16-bit αριθμούς και ένα 3-bit σήμα το οποίο δηλώνει την πράξη που πρέπει να εκτελεστεί, και θα βγάζει το 16-bit αποτέλεσμα.



Εντολή	Περιγραφή	Πράξη	Operation
ADD	Πρόσθεση	$O = I1 + I2$	000
SUB	Αφαίρεση	$O = I1 - I2$	001
AND	Λογικό AND (bit προς bit)	$O = I1 \text{ AND } I2$	010
OR	Λογικό OR (bit προς bit)	$O = I1 \text{ OR } I2$	011
GEQ	Μεγαλύτερο-Ίσο (Greater Equal)	$O = (I1 \geq 0)$	100
NOT	Not	$O = (I1 == 0)$	101

- Οι αριθμοί θα είναι σε μορφή συμπλήρωμα ως προς 2, συνεπώς οι πράξεις θα πρέπει να γίνονται όπως ορίζονται στο πακέτο *ieee.std\_logic\_signed.all*
- Η εντολή GEQ βάζει στο Output τον αριθμό 1 εάν το Input1 είναι μη αρνητικός ( $\geq 0$ ) ή τον αριθμό 0 εάν είναι αρνητικός ( $< 0$ ). Γι' αυτό μας βοηθάει το πιο σημαντικό ψηφίο του.
- Η εντολή NOT βάζει στο Output τον αριθμό 1 (15 bit 0 και το LSB 1) εάν το Input1 είναι 0 ή τον αριθμό 0 εάν είναι διάφορο του 0. **ΔΕΝ** αλλάζει τον αριθμό bit προς bit.

β) Κατασκευάστε ένα 16-bit Flip Flop σε structural κώδικα (πύλη προς πύλη) με όνομα reg σε VHDL. Το κύκλωμα θα παίρνει είσοδο το ρολόι και ένα Enable σήμα. Η εγγραφή θα γίνεται μόνο όταν το Enable είναι 1, τη στιγμή που «ανεβαίνει» το ρολόι.



γ) Χρησιμοποιώντας τα κομμάτια που φτιάξατε παραπάνω, να κατασκευάσετε το παρακάτω κύκλωμα σε VHDL.

