



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Σχεδιασμός Ολοκληρωμένων Κυκλωμάτων VLSI II

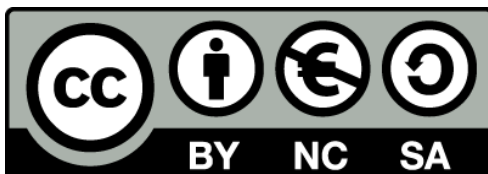
3^η Εργαστηριακή Άσκηση

**Σχεδίαση και Υλοποίηση μίας
ALU δύο εισόδων**

ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα **ΠΠ**

Άδειες Χρήσης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη Δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Ανάπτυξη

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών.



Άσκηση: Σχεδίαση και Υλοποίηση μίας ALU

Στόχος της άσκησης είναι αρχικά να σχεδιαστεί μια πλήρως λειτουργική μονάδα αριθμητικών και λογικών πράξεων **σε δομική σχεδιαστική μορφή**. Η μονάδα θα πρέπει να έχει τις εξής εισόδους και εξόδους:

A: Είσοδος (8-bit) – Πρώτος τελεστής σε συμπλήρωμα ως προς 2

B: Είσοδος (8-bit) – Δεύτερος τελεστής σε συμπλήρωμα ως προς 2

Op: Είσοδος (3-bit) – Κωδικός πράξης

Out: Έξοδος (8-bit) – Αποτέλεσμα σε συμπλήρωμα ως προς 2.

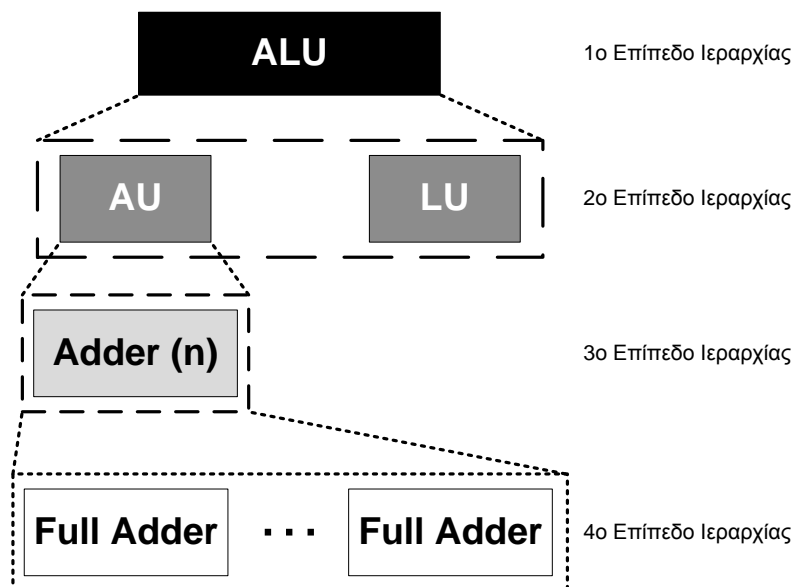
Zero: Έξοδος (1-bit) – Ενεργοποιημένη αν το αποτέλεσμα είναι μηδέν

Cout: Έξοδος (1-bit) – Ενεργοποιημένη αν υπήρξε κρατούμενο (Carry)

Η «συμπεριφορά» της ALU είναι η εξής:

Κωδικός	Πράξη	Αποτέλεσμα
Op = 000	Πρόσθεση	Out = A + B
Op = 001	Αφαίρεση	Out = A - B
Op = 100	Λογικό «ΚΑΙ»	Out = A & B
Op = 101	Αντιστροφή του A	Out = ! A
Op = 110	Λογικό «Η»	Out = A B
Op = 111	Λογικό «XOR»	Out = A ⊕ B

Ιεραρχία της σχεδίασης:

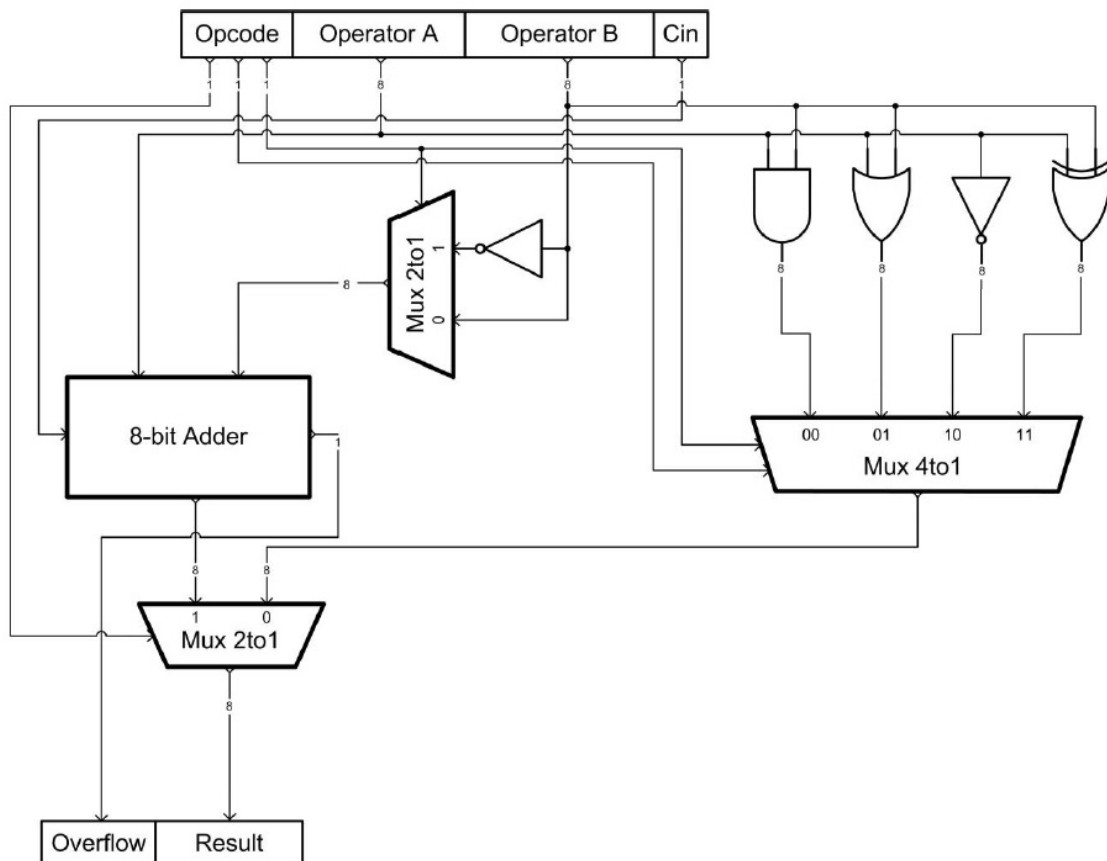


Στην σχεδίασή σας πρέπει να χρησιμοποιήσετε τουλάχιστον τέσσερα επίπεδα ιεραρχίας. Στο υψηλότερο επίπεδο βρίσκεται η μονάδα ALU, η οποία αποτελείται από δύο εξαρτήματα (components) ένα για αριθμητικές πράξεις (AU) και ένα για λογικές πράξεις (LU), και ό,τι λογική είναι απαραίτητη για την διασύνδεσή τους (2^ο επίπεδο).

Το εξάρτημα Αριθμητικών Πράξεων (AU) πρέπει οπωσδήποτε να υλοποιηθεί σε δομική μορφή (π.χ. με χρήση της λειτουργικότητας GENERATE) και χρησιμοποιώντας ένα εξάρτημα πλήρους αθροιστή (σε δομική μορφή επίσης – 3^ο επίπεδο). Το εξάρτημα Λογικών πράξεων θα πρέπει να υλοποιηθεί επίσης με δομική μορφή (κάθε 32-bit πύλη ξεχωριστά).

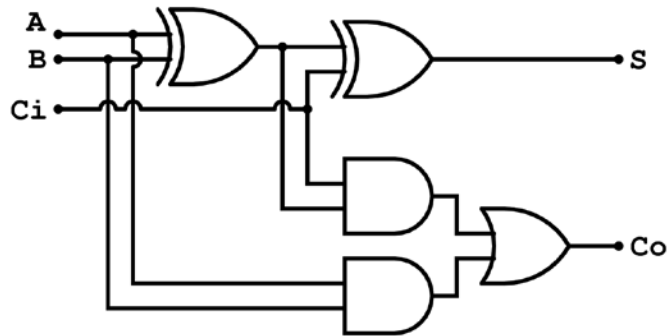
Ενδεικτική Αρχιτεκτονική

Παρακάτω παρατίθεται μια ενδεικτική αρχιτεκτονική σε δομικό διάγραμμα, την οποία μπορείτε να συμβουλευτείτε. Στη συγκεκριμένη αρχιτεκτονική, η αφαίρεση έχει γίνει με τη βοήθεια συμπληρώματος ως προς 2 (2's complement), το οποίο πραγματοποιείται με την αντιστροφή του τελεστή που αφορά τον αφαιρέτη και πρόσθεση του 1 μέσω της εισόδου Cin. Γίνεται χρήση 3 πολυπλεκτών, οι οποίοι μπορούν εύκολα να υλοποιηθούν με τη βοήθεια κυκλωμάτων προηγούμενων εργαστηρίων, ενός αθροιστή 8-bit και απλών λογικών πυλών. Σε κάθε περίπτωση, και οι δύο εισοδοι εισέρχονται στην ALU και εκτελούνται όλες οι πράξεις (αριθμητικές και λογικές). Στο τέλος, μέσω ενός μεγάλου πολυπλέκτη, επιλέγεται η επιθυμητή έξοδος ανάλογα με τη ζητούμενη πράξη (Opcode εντολής εισόδου). Το εύρος σε bits κάθε σήματος αναγράφεται πάνω στο σήμα. Η ενδεικτική αρχιτεκτονική είναι η ακόλουθη:



Παρατηρήσεις:

Δώστε προσοχή στις εξόδους Zero και Cout. Ποιές είναι οι συνθήκες που ορίζουν αυτές τις εξόδους? Πως πετυχαίνω την αφαίρεση χρησιμοποιώντας έναν αθροιστή όταν η κωδικοποίηση των τελεστών είναι σε συμπλήρωμα ως προς δύο; Ένας πλήρης αθροιστής υλοποιείται δομικά με χρήση πυλών XOR, AND και OR:



Η άσκηση να υλοποιηθεί ΑΥΣΤΗΡΑ με ΔΟΜΙΚΗ μορφή κώδικα (structural VHDL)

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Βασίλης Παλιουράς, Γεώργιος Θεοδωρίδης,
«Σχεδιασμός Ολοκληρωμένων Κυκλωμάτων (VLSI) II»

Έκδοση: 1.0 Πάτρα 2015

Διαθέσιμο στη διαδικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/EE892/>

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

