

Σχεδίαση Ψηφιακών Συστημάτων Εργασία 2

Τμήμα Πληροφορικής

Εαρινό Εξάμηνο 2021-2022

Αριθμός ομάδας: DSD030

Νικόλαος Λιονής (3180098)
email: nicklioniss@gmail.com

Αθανάσιος Τριφώνης (3200298)
email: p3200298@aueb.gr

ΑΝΑΦΟΡΑ

ΜΕΡΟΣ 1^ο

Στο πρώτο μέρος της εργασίας υλοποιήσαμε μία ALU για πράξεις με 1 bit. Κατασκευάσαμε τα packages: [basic_components](#), [inverter_mux2to1](#), [my_mux4to1](#), [fullAdd_component](#).

[basic_components](#)-> Εδώ περιλαμβάνονται 3 βασικές πύλες(AND, OR, XOR) οι οποίες δέχονται 2 εισόδους η κάθε μία και παράγουν 1 έξοδο.

[inverter_mux2to1](#)-> Αυτό το πακέτο είναι ένας πολυπλέκτης 2 σε 1, δέχεται ως εισόδους: in1, in2, s και παράγει μία έξοδο inverted. Ουσιαστικά διαβάζει την είσοδο s και βάσει αυτής επιλέγει μία από τις in1, in2 οι οποίες θα είναι μία τιμή και το συμπλήρωμά της. Για s = 0 επιστρέφει την αρχική τιμή, διαφορετικά επιστρέφει το συμπλήρωμά της. (Το κομμάτι στο οποίο βγαίνει το συμπλήρωμα το υλοποιήσαμε σε υψηλότερο επίπεδο, σε αυτό της alu.)

[my_mux4to1](#)-> Πολυπλέκτης 4 σε 1. Χρησιμοποιείται για να επιλέγουμε την επιθυμητή πράξη που θα εκτελεστεί σε επίπεδο alu.

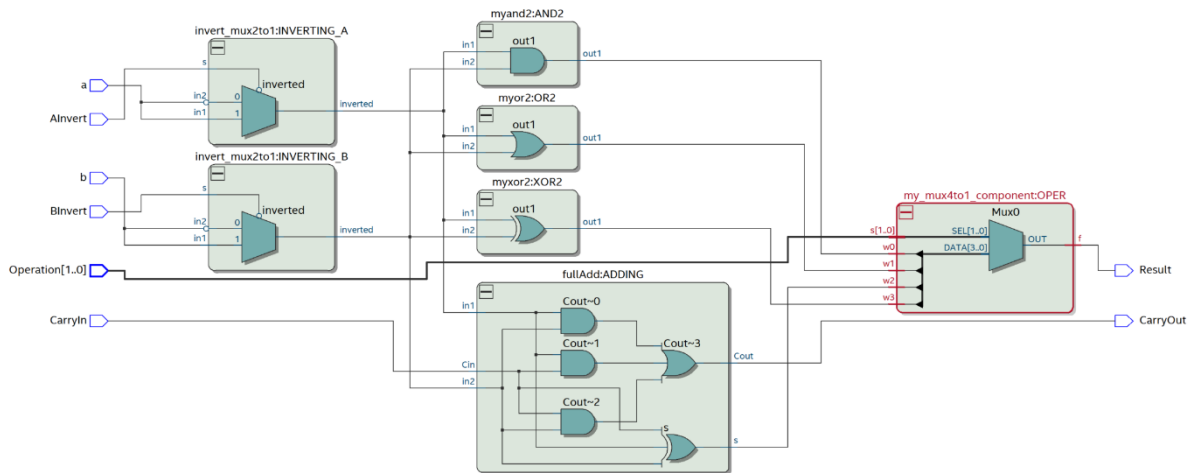
[fullAdd_component](#)-> Πλήρης αθροιστής, δέχεται 3 εισόδους, είσοδο κρατουμένου και 2 τιμές του 1 bit και παράγει το sum που είναι το άθροισμά τους, καθώς και το CarryOut που είναι το κρατούμενο.

ALU 1-BIT

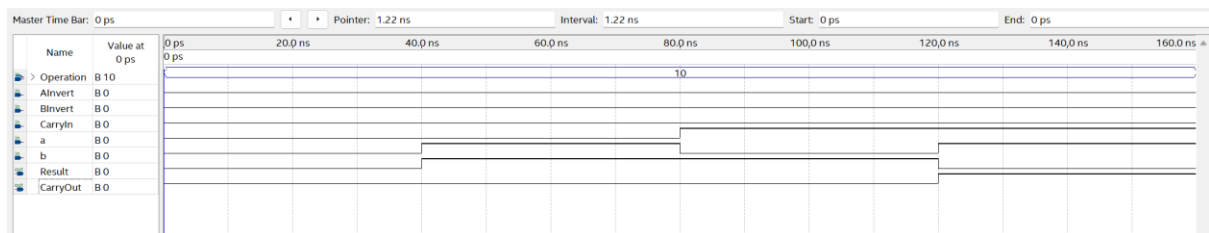
Η ALU δέχεται 2 αριθμούς του 1 bit, δέχεται για κάθε έναν από αυτούς μία είσοδο 0 ή 1 για να τους κάνει invert ή όχι, δέχεται είσοδο κρατούμενου και δέχεται και μία τιμή μήκους 2 bits για να γίνει η επιλογή πράξης. Παράγει το αποτέλεσμα της πράξης (Result) και ένα κρατούμενο εξόδου.

Παρακάτω επισυνάπτονται οι κυματομορφές για κάθε πράξη στην alu καθώς και το RTL διάγραμμα της.

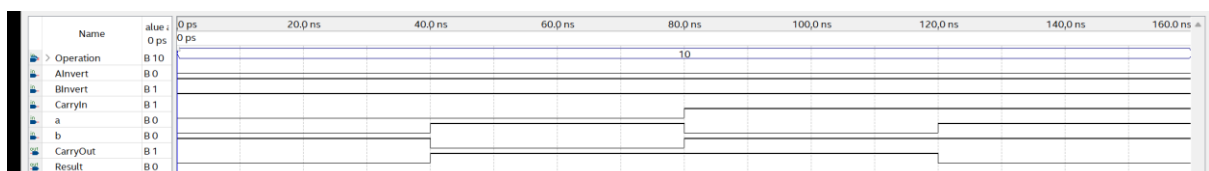
RTL diagram



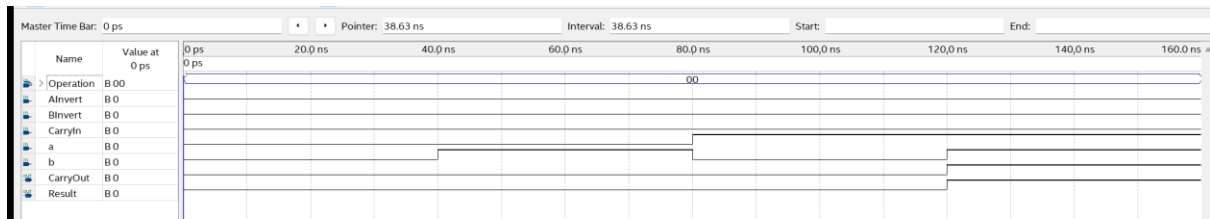
Πρόσθεση: $\alpha + \beta$



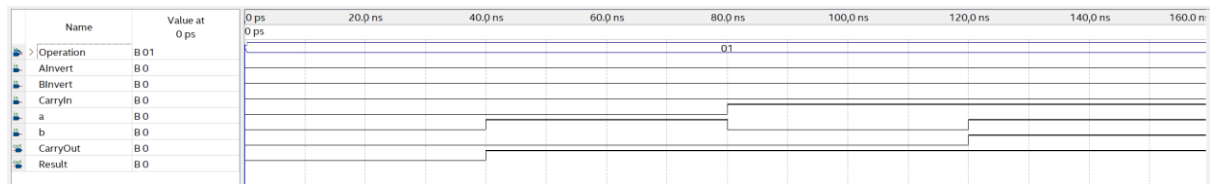
Αφαίρεση: $\alpha - \beta$



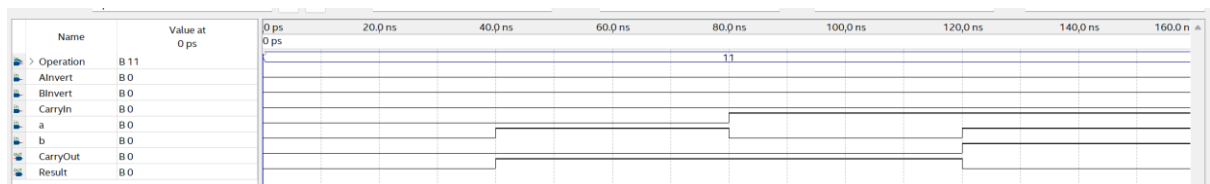
Λογική πράξη σύζευξης: a AND b



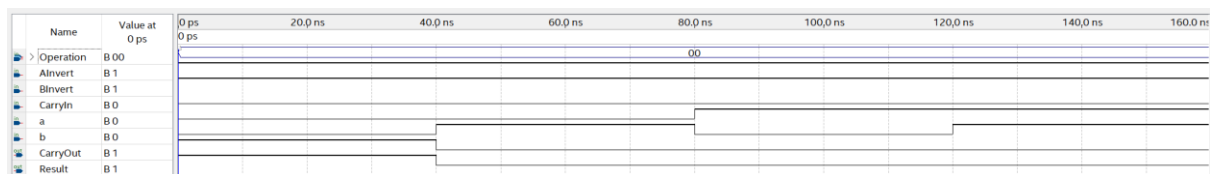
Λογική πράξη διάζευξης: a OR b



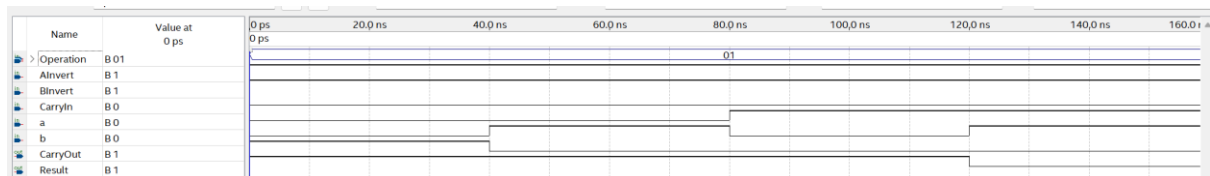
Λογική πράξη αποκλειστικής διάζευξης: a XOR b



Λογική πράξη NOR: a NOR b



Λογική πράξη NAND: a NAND b



ΜΕΡΟΣ 2°

Στο δεύτερο μέρος της εργασίας υλοποιούμε μία ALU 16-bits βασιζόμενη στη διαδοχική σύνδεση 16 slices της 1-bit ALU. Στην 16-bit το κρατούμενο εξόδου κάθε slice θα γίνεται κρατούμενο εισόδου για το επόμενο slice για να μπορεί να διαδοθεί το κρατούμενο πρακτικά.

Σε αυτό το project αντιγράψαμε τα αρχεία του πρώτου μέρους με την διαφορά πως το αρχείο που αντιστοιχεί στην 1-bit ALU εδώ το μετατρέψαμε σε πακέτο. Επιπλέον των προηγούμενων πακέτων δημιουργήσαμε το πακέτο [control_circuit](#).

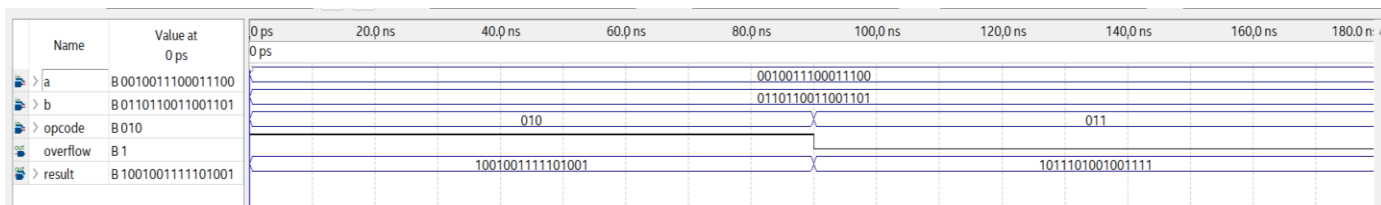
[control_circuit](#)-> Το πακέτο αυτό χρησιμοποιείται για να μπορέσουμε να διαλέξουμε την επιθυμητή πράξη κάθε φορά αναλόγως το operation code.

ALU 16-BIT

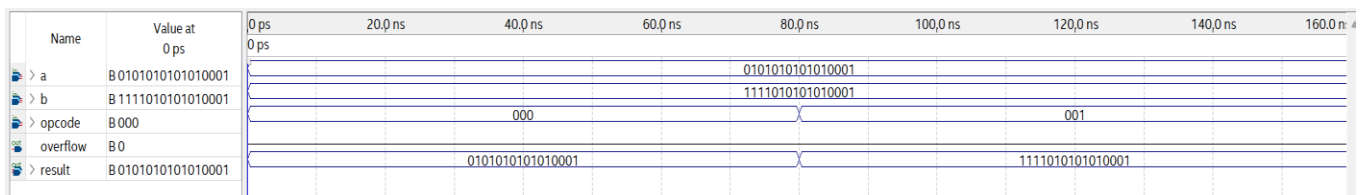
Η ALU αυτή δέχεται εισόδους δύο αριθμούς των 16bit και ένα opcode των 3bit για να γίνει η αντίστοιχη επιλογή της επιθυμητής πράξης. Παράγει σαν αποτέλεσμα το result της πράξης και μία τιμή overflow η οποία είναι 0 όταν δεν έχουμε υπερχείλιση επειδή δεν επιλέχθηκε η πράξη της αφαίρεσης ή της πρόσθεσης ή αν επιλέχθηκε η πρόσθεση/αφαίρεση δεν υπήρξε υπερχείλιση. Σε περίπτωση που βρεθεί υπερχείλιση(διαπιστώνεται όταν τα δυο τελευταία κρατούμενα διαφέρουν μεταξύ τους) τότε το overflow παίρνει τιμή 1.

Παρακάτω παραθέτονται οι κυματομορφές κάθε πράξης(σημειώνονται και οι αντίστοιχοι opcode) καθώς και το διάγραμμα RTL της 16bit ALU.

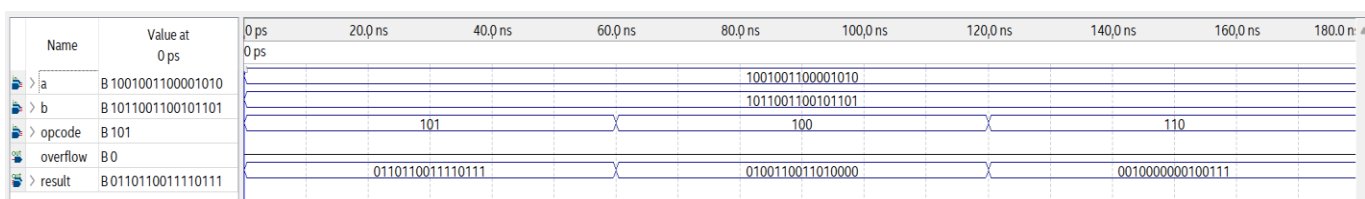
Πρόσθεση(010): $a + b$, Αφαίρεση(011): $a - b$



Λογικές πράξεις AND(000) και OR(001)



Λογικές πράξεις NAND(101), NOR(100) και XOR(110)



RTL diagram

