ΣΧΕΔΙΑΣΗ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Εργασία 2

ΜΑΡΙΑ ΣΧΟΙΝΑΚΗ 3210191 p3210191@aueb.gr

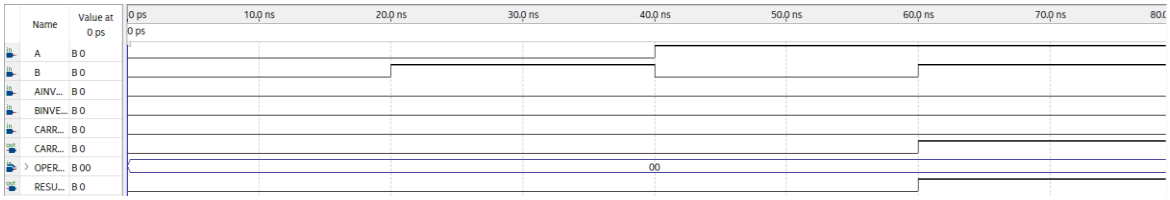
ΧΡΗΣΤΟΣ ΣΤΑΜΟΥΛΟΣ 3210188 p3210188@aueb.gr

Μέρος 1ο

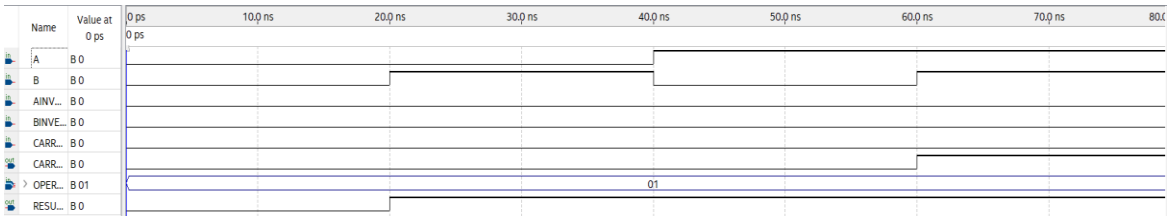
Ξεκινήσαμε χτίζοντας Entities για τις λογικές πράξεις OR και AND αντίστοιχα. Στην συνέχεια φτιάξαμε έναν αθροιστή στον οποίο βάλαμε 2 εισόδους 1-bit για τους τελεστέους και μια έξοδο 1-bit για το άθροισμα. Προσθέσαμε επίσης μια δεύτερη έξοδο αυτή του κρατουμένου CarryOut και μία τρίτη είσοδο αυτή της εισόδου κρατουμένου CarryIn. Έως τώρα έχουμε υλοποιήσει τις πράξεις AND, OR και πρόσθεση. Η υλοποίηση της αφαίρεσης δεν είναι τίποτα περισσότερο από την άθροιση του αφαιρέτη με το συμπλήρωμα ως προς 2 του αφαιρετέου. Οπότε υλοποιήσαμε έναν Multiplexer ο οποίος θα δέχεται σαν είσοδο ένα σήμα που θα προκαθορίζει αν θέλει ή όχι(0 για πρόσθεση και 1 για αφαίρεση) να μετατρέψει τον αριθμό σε συμπλήρωμα ως προς 1. Αν το σήμα είναι 0 ο αριθμός θα παραμείνει ως έχει, αλλιώς θα μετατραπεί σε συμπλήρωμα ως προς 1 και θέτοντας το CarryIn = 1, στον αθροιστή θα εκτελεστεί η πράξη της αφαίρεσης καθώς στον αριθμό που είναι σε συμπλήρωμα ως προς 1 προσθέτουμε τον αριθμό 1, άρα γίνεται συμπλήρωμα ως προς 2. Για την πράξη του NOR χρειαζόμαστε απλά και για τους 2 αριθμούς-εισόδους τον Multiplexer 2 to 1 που είναι δομημένος το ίδιο οπότε δεν χρειάστηκε να κάνουμε καινούριο. Όταν θα θέλουμε την πράξη NOR θα δίνουμε σήμα 1 στον Multiplexer 2 φορές με κάθε αριθμό σαν είσοδο κάθε φορά και μετά τις εξόδους του Multiplexer 2 to 1 θα τις δίνουμε στην AND. Ακριβώς ανάλογα και για την πράξη NAND δίνουμε σήμα 1 στον Multiplexer 2 φορές με κάθε αριθμό σαν είσοδο κάθε φορά και μετά τις εξόδους του Multiplexer 2 to 1 θα τις δίνουμε στην OR. Τέλος, προσθέσαμε ένα Entity XOR και προσθέσαμε τον 4 to 1 Multiplexer ο

οποίος σύμφωνα με το σήμα που θα δέχεται θα αποφασίζει ποια πράξη θα εκτελέσει και θα παράγει το τελικό αποτέλεσμα(Result).

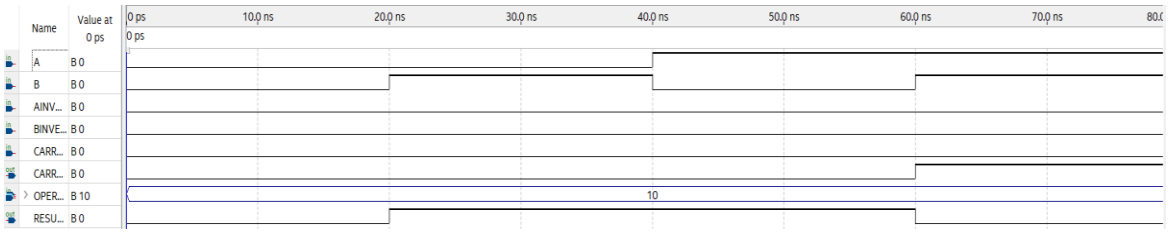
ΠΡΑΞΗ AND (OPERATION = 00)



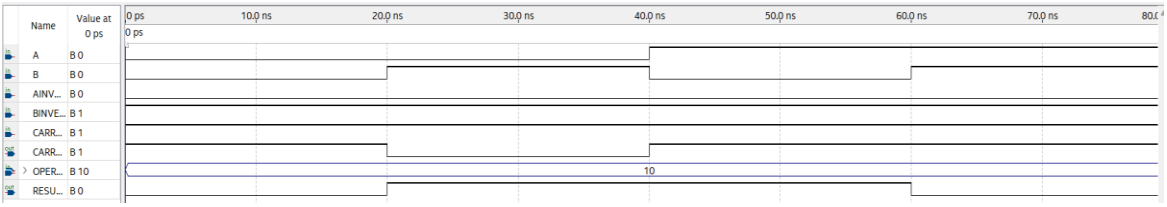
ΠΡΑΞΗ OR (OPERATION = 01)



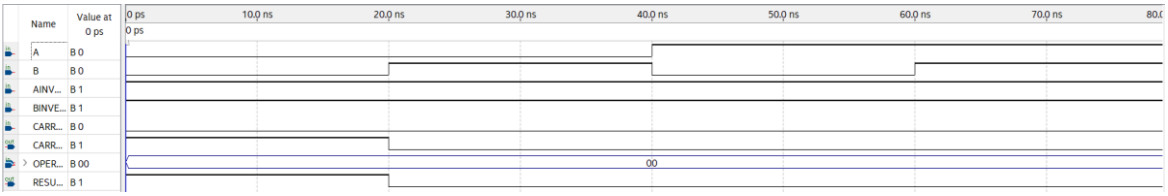
ΠΡΑΞΗ ADD (OPERATION = 10)



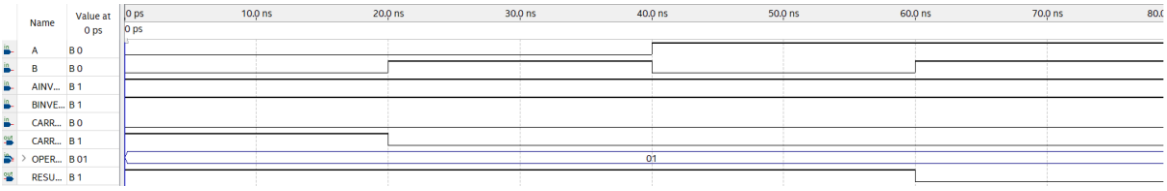
ΠΡΑΞΗ SUB (OPERATION = 10, BINVERT=1, CARRYIN=1)



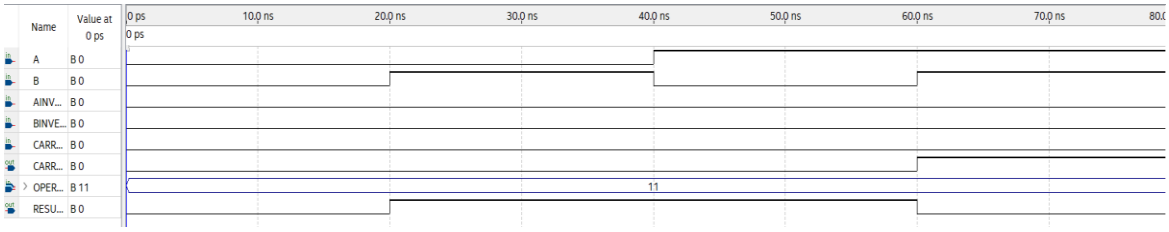
ΠΡΑΞΗ NOR (OPERATION = 00, AINVERT=1, BINVERT=1)



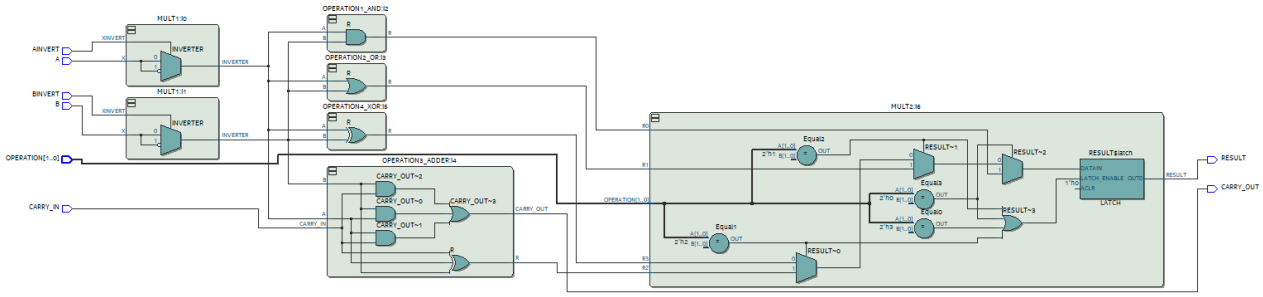
ΠΡΑΞΗ NAND (OPERATION = 01, AINVERT=1, BINVERT=1)

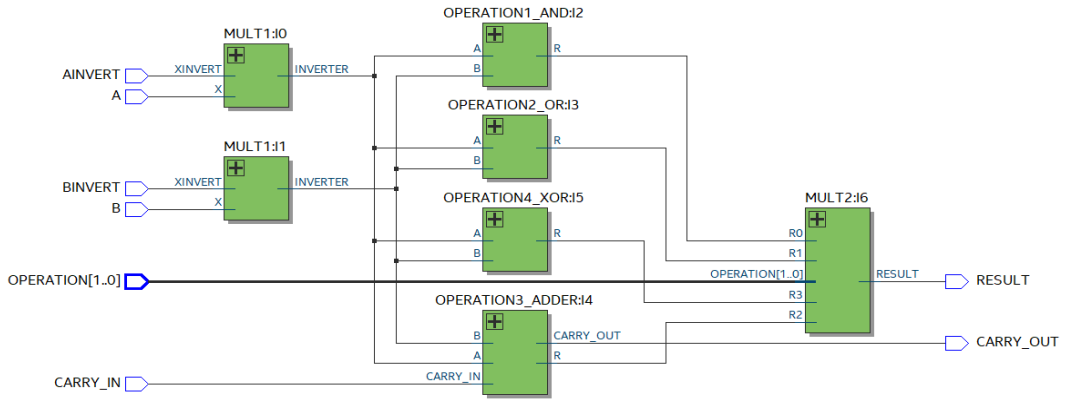


ΠΡΑΞΗ XOR (OPERATION = 11)



RTL ΔΙΑΓΡΑΜΜΑ





Μέρος 2ο

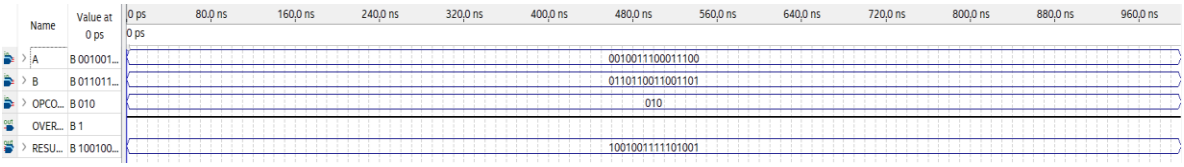
Για την 16-bit ALU, αρχικά συνδέσαμε την 1-bit ALU στο 2ο αρχείο, βάζοντας την στον ίδιο φάκελο και επιλέγοντάς την ύστερα από τις

επιλογές του new project wizard. Κατασκευάσαμε 1 Multiplexer ο οποίος ανάλογα με το σήμα του opcode που λαμβάνει θέτει τα CarryIn, Operation, Ainvert, Binvert αντίστοιχα σύμφωνα με τον πίνακα της εκφώνησης. Προσθέσαμε και ένα Entity για το Overflow ώστε να ενημερώνεται το ενδεχόμενο της υπερχείλισης. Μετά κατασκευάσαμε το Top-level Entity το οποίο δέχτηκε τον Multiplexer ως components , 1 component Overflow και την 1-bit ALU του 1^{ου} αρχείου επίσης ως component. Υλοποιήσαμε όλα τα σήματα εισόδου-εξόδου και components σε port maps και παρήχθησαν τα τελικά αποτελέσματα για κάθε slice 1-bit, ώστε να ενωθούν μετά όλα τα αποτελέσματα των slices σε ένα ενιαίο «Result» και να έχουμε το επιθυμητό αποτέλεσμα.

i) ΠΡΑΞΗ ADD (OPCODE=010)

A = 0010011100011100
B = 0110110011001101

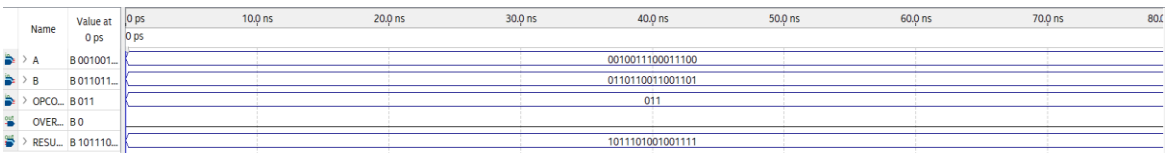
A + B= 1001001111101001



ΠΡΑΞΗ SUB (OPCODE=011)

A = 0010011100011100
B = 0110110011001101

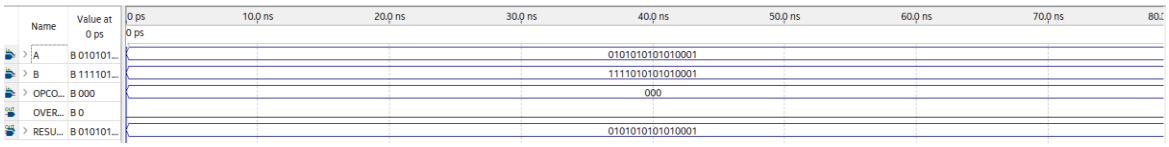
A - B= **1011101001001111**



ii) ΠΡΑΞΗ AND (OPCODE=000)

A = 0101010101010001
B = 1111010101010001

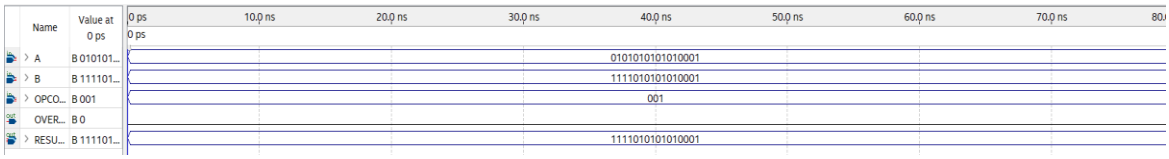
A AND B = **0101010101010001**



ΠΡΑΞΗ OR(OPCODE=001)

A = 0101010101010001
B = 1111010101010001

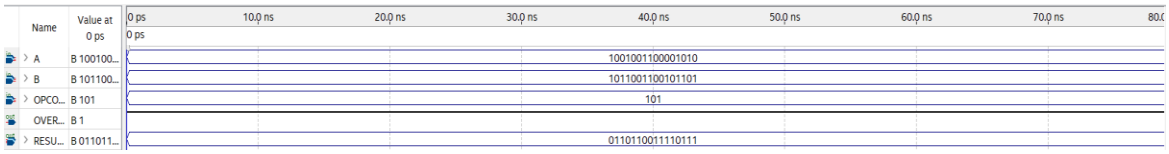
A OR B = **1111010101010001**



iii) ΠΡΑΞΗ NAND(OPCODE=101)

A = 1001001100001010
B = 1011001100101101

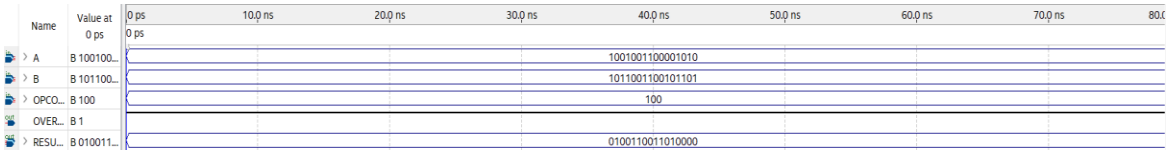
A NAND B = **0110110011110111**



ΠΡΑΞΗ NOR(OPCODE=100)

A = 1001001100001010
B = 1011001100101101

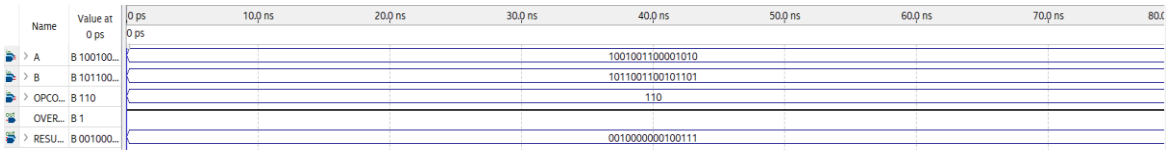
A NOR B = **010011001101000**



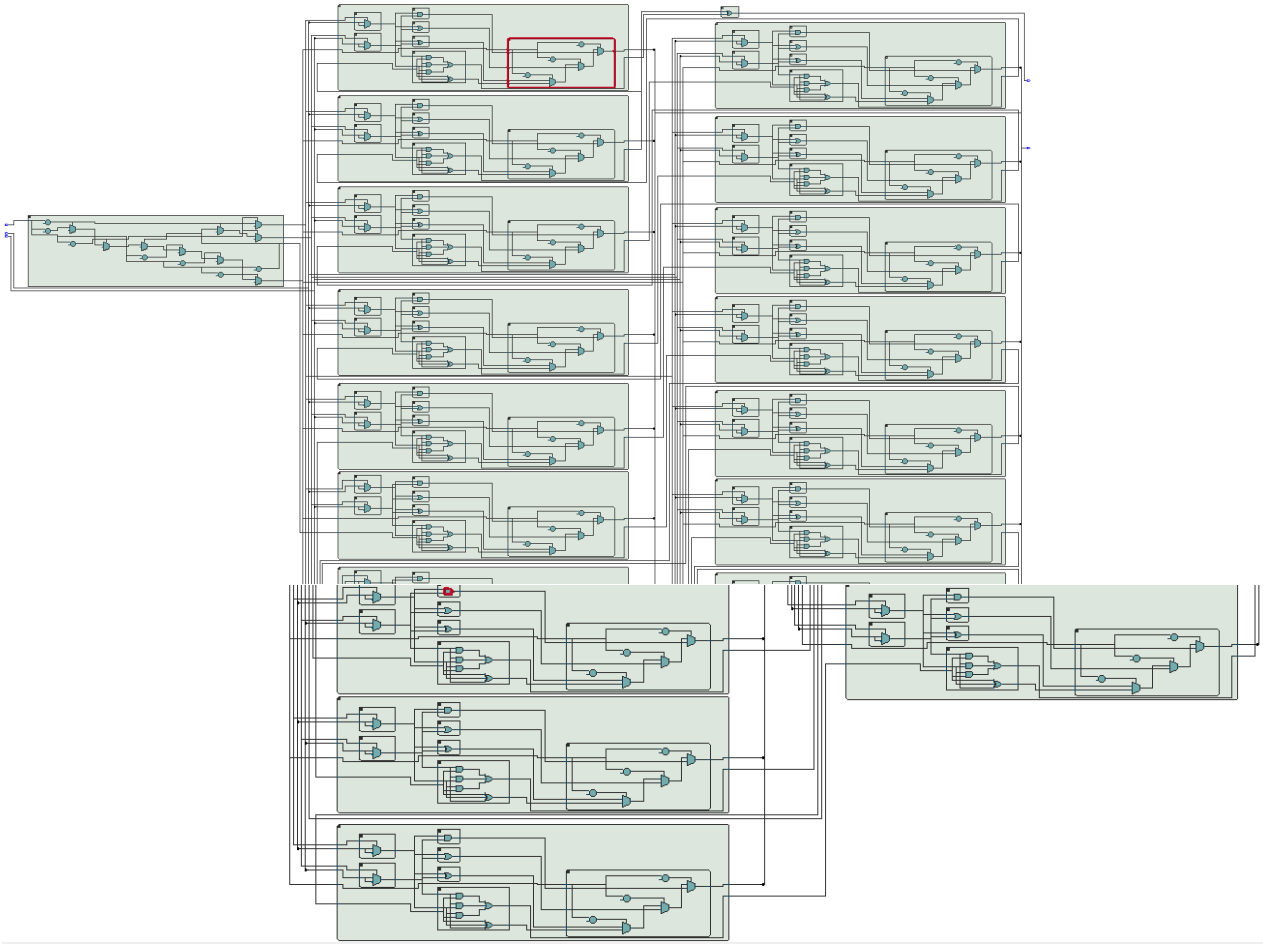
ΠΡΑΞΗ XOR(OPCODE=110)

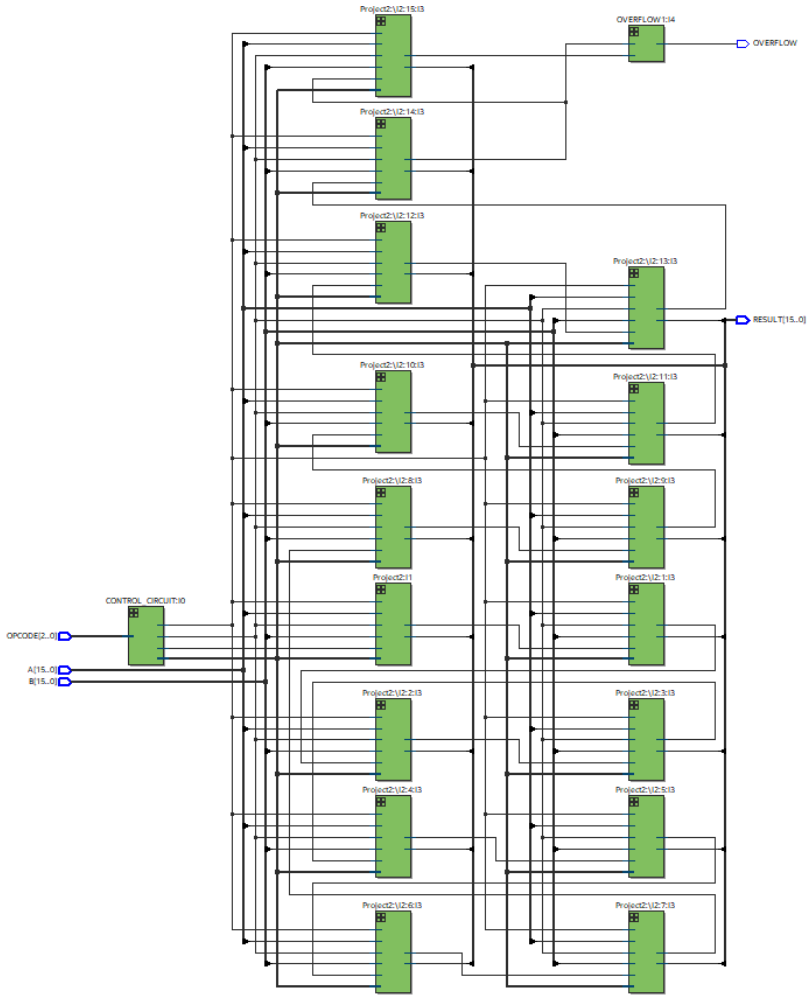
A = 1001001100001010
B = 1011001100101101

A XOR B = **0010000000100111**



RTL ΔΙΑΓΡΑΜΜΑ




Project2:\I2:15:I3
OVERFLOW1:I4
OVERFLOW
Project2:\I2:14:I3
Project2:\I2:12:I3
Project2:\I2:13:I3
RESULT[15..0]
Project2:\I2:10:I3
Project2:\I2:11:I3
Project2:\I2:8:I3
Project2:\I2:9:I3
Project2:I1
Project2:\I2:1:I3
CONTROL_CIRCUIT:I0
OPCODE[2..0]
A[15..0]
B[15..0]
Project2:\I2:2:I3
Project2:\I2:3:I3
Project2:\I2:4:I3
Project2:\I2:5:I3
Project2:\I2:6:I3
Project2:\I2:7:I3