



Σχεδίαση Ψηφιακών Συστημάτων Εργασία 2 Τμήμα Πληροφορικής

Εαρινό Εξάμηνο 2021-2022 Αριθμός ομάδας: DSD030

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

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

АМАФОРА

MEPOΣ 1°

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 πολυπλέχτης 2 σε 1, δέχεται ως εισόδους: in1, in2, s και παράγει μία έξοδο inverted. Ουσιαστικά διαβάζει την είδοδο s και βάσει αυτής επιλέγει μία από τις in1, in2 οι οποίες θα είναι μία τιμή και το συμπλήρωμα 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-> Πολυπλέχτης 4 σε 1. Χρησιμοποιήται για να επιλέγουμε την επιθυμητή πράξη που θα εκτελεστεί σε επίπεδο alu.

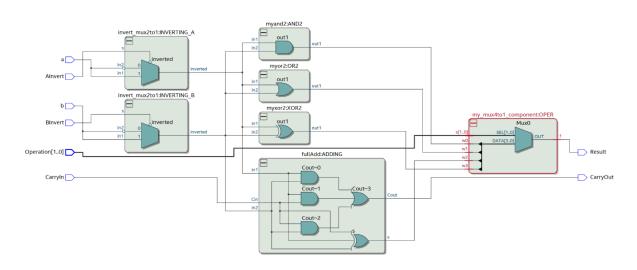
fullAdd_component-> Πλήρης αθροιστής, δέχεται 3 εισόδους, είσοδο κρατουμένου και 2 τιμές του 1 bit και παράγει το sum που είναι το άθροισμα τους, καθώς και το CarryOut που είναι το κρατούμενο.

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 κρατουμένου 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 διάγραμμά της.

RTL diagram



Πρόσθεση: α + β



Αφαίρεση: α - β

Name	alue a		20.0 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120,0 ns	140,0 ns	160.0 ns
> Operation	B 10					10				
Alnvert	BO									
Binvert	B 1									
Carryln	B 1									
. a	BO	-								
b	BO									
CarryOut	B1									
Result	BO									

Λογική πράξη σύζευξης: a AND b

		Value at	0 ps	20.0 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120,0 ns	140,0 ns	160.0 ns	
	Name	e Ops		0 ps	2017110							
>	Operation	B 00					00					
	Alnvert	во										
	BInvert	B 0										
	CarryIn	BO										
	a	BO										
	b	BO										
1	CarryOut	BO										
1	Result	B 0										

Λογική πράξη διάζευξης: a OR b

		Value at	0 ps	20.0 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120,0 ns	140,0 ns	160.0 n:
	Name	0 ps	0 ps								
-	Operation	B 0 1	E				01				
in.	Alnvert	B 0									
<u>in</u>	BInvert	B 0									
in.	Carryln	B 0									
<u>in</u> _	a	B 0									
in.	b	B 0									-
955	CarryOut	B 0									
945	Result	BO									

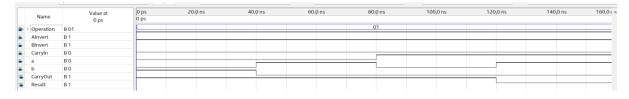
Λογική πράξη αποκλειστικής διάζευξης: a XOR b

	Name	Value at 0 ps	0 ps 0 ps	20.0 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120 _, 0 ns	140,0 ns	160.0 n
	> Operation	B 11					11				
in.	Alnvert	B 0									
in_	Blnvert	B 0									
in.	CarryIn	B 0									
in_	a	B 0									
in_	b	B 0									
out	CarryOut	B 0									
Out	Result	BO									

Λογική πράξη NOR: a NOR b

		Value at	0 ps	20.0 ns	40.0 ns	60.0 n	s 80.0	ns	100,0 ns	120,0 ns	140,0 ns	160.0 ns
	Name	0 ps	0 ps									
-	Operation	B 00					0					
in_	Alnvert	B 1										
in.	BInvert	B 1										
in_	CarryIn	BO										
in.	a	B 0										
in.	b	BO										
915	CarryOut	B 1										
out	Result	B 1										

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.

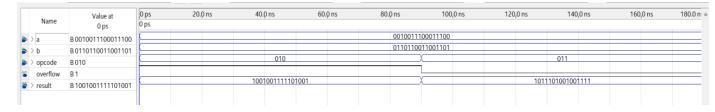
control_circuit-> Το πακέτο αυτό χρησιμοποιείται για να μπορέσουμε να διαλέξουμε την επιθυμητή πράξη κάθε φορά αναλόγως το operation code.

ALU 16-BIT

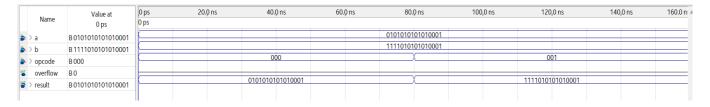
Η ALU αυτή δέχεται εισόδους δύο αριθμούς των 16bit και ένα opcode των 3bit για να γίνει η αντίστοιχη επιλογή της επιθυμητής πράξης. Παράγει σαν αποτέλεσμα το result της πράξης και μία τιμή overflow η οποία είναι 0 όταν δεν έχουμε υπερχείληση επειδή δεν επιλέχθηκε η πράξη της αφαίρεσης ή της πρόσθεσης ή αν επιλέχθηκε η πρόσθεση/αφαίρεση δεν υπήρξε υπερχείληση. Σε περίπτωση που βρεθεί υπερχείληση(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 ΑΝD(000) και OR(001)



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

	Name	Value at	0 ps	20.0 ns	40.0 ns	60.0 ns	80.0 ns	100 _r 0 ns	120 _r 0 ns	140 ₁ 0 ns	160 ₁ 0 ns	180.0 n: 🔺
	Ivallie	0 ps	0 ps									
5	a	B 1001001100001010					10010011	00001010				
-	> b	B 1011001100101101					10110011	00101101				
5	opcode	B 101		10	1	X	10	00	X	110)	
out	overflow	В0										
*	> result B 0110110011110111			011011001	1110111	X	01001100	11010000	X	001000000	0100111	+
			1									

RTL diagram

