



ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2018-19

Εργαστήριο 3

Ολοκλήρωση και προσομοίωση datapath επεξεργαστή ενός κύκλου (χωρίς το κομμάτι ελέγχου)

Σκοπός του Εργαστηρίου

Η ολοκλήρωση και ο έλεγχος του datapath του εργαστηρίου #2. Θα πρέπει να συνδέσετε τις βαθμίδες που υλοποιήσατε στο προηγούμενο εργαστήριο και μία (μόνο) μνήμη συνδεδεμένη ταυτόχρονα με τις βαθμίδες IF και MEM. Παρατηρήστε το datapath που κατασκευάσατε από το προηγούμενο ερώτημα. Συλλέξτε όλα τα απαραίτητα σημεία ελέγχου και σχεδιάστε την προσέγγιση ελέγχου του ενιαίου datapath.

Εκτέλεση

1. Γράψτε τον κώδικα VHDL που υλοποιεί το datapath του επεξεργαστή ενώνοντας τις 4 βαθμίδες που υλοποιήσατε στο 2^ο Εργαστήριο. Προσθέστε την είσοδο ImmExt στην βαθμίδα Dec. Διορθώστε τον κώδικα μνήμης με τον πιο πρόσφατο.
2. Επιβεβαιώστε την ορθή λειτουργία του datapath με τα προγράμματα αναφοράς και άλλα δικά σας. Για το πρώτο θα σας δίδεται το αρχείο αρχικοποίησης της μνήμης και το testbench, Συμπληρώστε τα σήματα ελέγχου για το δεύτερο, και φτιάξτε το αντίστοιχο testbench. Για να ελέγξετε το κύκλωμά σας απλά δώστε στο test bench Reset='1' για μερικούς κύκλους και μετά Reset='0' έτσι ώστε να ξεκινήσει η εκτέλεση από την θέση 0.

Παραδοτέα

1. Αρχεία κώδικα VHDL (πηγαίος).
2. Σχηματικό διάγραμμα του ολοκληρωμένου datapath δείχνοντας τις μεταξύ των βαθμίδων συνδέσεις.
3. Ο πίνακας σημάτων ελέγχου για το δεύτερο πρόγραμμα και το testbench που τον υλοποιεί.
4. Σύντομη αναφορά στη διαδικασία σχεδίασης και υλοποίησης (μαζί με σχόλια και ενδεχόμενα προβλήματα που αντιμετωπίσατε).

Προαιρετικά (+30%) στον βαθμό του εργαστηρίου

- Υλοποιήστε την αποκωδικοποίηση εντολών ώστε τα σήματα ελέγχου να παράγονται αυτόματα.
- Προσομοιώστε τα προγράμματα αναφοράς και άλλα δικά σας.

Πρόγραμμα αναφοράς #1

```

00: addi r5, r0, 8
04: ori  r3, r0, 0xABCD
08: sw   r3, 4(r0) // γράφει στην διεύθυνση 0x4 => 0x404 την τιμή 0x0000ABCD
0C: lw   r10, -4(r5) // διαβάζει από την διεύθυνση 0x4 => 0x404 την τιμή 0x0000ABCD
10: lb   r16, 4(r0) // διαβάζει byte από την διεύθυνση 0x4 => 0x404 την τιμή 0xFFFFFCD
14: nand r4, r10, r16

```

Εντολή	Opcode	rs	rd	rt	func	Immed	Binary	Hex
00: addi r5,r0,8	110000	00000	00101	-	-	00000000000001000	1100 00 00 000 0 0101 00000000000001000	C005 0008
04: ori r3,r0,ABCD	110011	00000	00011	-	-	10101011111001101	1100 11 00 000 0 0011 10101011111001101	CC03 ABCD
08: sw r3,4(r0)	011111	00000	00011	-	-	00000000000000100	0111 11 00 000 0 0011 00000000000000100	7C03 0004
0C: lw r10,-4(r5)	001111	00101	01010	-	-	1111111111111100	0011 11 00 101 0 1010 1111111111111100	3CAA FFFC
10: lb r16,4(r0)	000011	00000	10000	-	-	00000000000000100	0000 11 00 000 1 0000 00000000000000100	0C10 0004
14: nand r4,r10,r16	100000	01010	00100	10000	110010	-	1000 00 01 010 0 0100 1000 0 000 00 11 0010	8144 8032

Εντολή	Σήματα ελέγχου								
	PC_Sel	PC_LdEn	RF_B_sel	RF_WrData_sel	ImmExt	ALU_Bin_sel	RF_Wr_en	ALU_func	MEM_Wr_En
00: addi r5,r0,8	0	1	D	ALU_out	1 (sign)	Immed	1	add	0
04: ori r3,r0,ABCD	0	1	D	ALU_out	0 (zero)	Immed	1	or	0
08: sw r3,4(r0)	0	1	D	D	1 (sign)	Immed	0	add	1
0C: lw r10,-4(r5)	0	1	D	MEM_out	1 (Sign)	Immed	1	add	0
10: lb r16,4(r0)	0	1	D	ALU_out	1 (Sign)	Immed	1	add	0
14: nand r4,r10,r16	0	1	rt	ALU_out	-	RF_B	1	nand	0

Πρόγραμμα αναφοράς #2

// *** Δεύτερο πρόγραμμα, μόνο διακλαδώσεις, εκτελεί δύο εντολές με αποτυχημένη διακλάδωση και ξανά

```
00: bne r5, r5, 8    // αποτυχημένη διακλάδωση
04: b -2            // branch (PC=04 + 4 -2*4 = 00) infinite loop!
08: addi r0, r0, 0  // δεν θα εκτελεστεί
```

Εντολή	Opcode	rs	rd	rt	func	Immed	Binary	Hex
00: bne r5,r5,8	000001	00101	00101	-	-	00000000000001000	0000 01 00 101 0 0101 00000000000001000	04A5 0008
04: b -2	111111	00000	00000	-	-	1111111111111110	1111 11 00 000 0 0000 1111111111111110	FC00 FFFE
08: add r1,r2,r3	100000	00010	00001	00011	110000	-	1000 00 00 010 0 0001 0001 1 000 00 11 0000	8041 1830

	Σήματα ελέγχου								
Εντολή	PC_Sel	PC_LdEn	RF_B_sel	RF_WrData_sel	ImmExt	ALU_Bin_sel	RF_Wr_en	ALU_func	MEM_Wr_En
00: bne r5,r5,8									
04: b -2									
08: add r1,r2,r3									

Φτιάξτε και άλλα προγράμματα (η επεκτείνετε αυτά) ώστε να ελέγξετε και άλλες περιπτώσεις.