



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΗΡΥ 302: ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021
Παρεμβάσεις: Κυπριανός Παπαδημητρίου

Εργασία #2: Σχεδίαση επεξεργαστή πολλαπλών κύκλων και μετατροπή του σε pipeline

Χωρίζεται σε 2 φάσεις

4^η φάση: χρόνος ολοκλήρωσης 8 ημέρες
«Σχεδίαση επεξεργαστή πολλαπλών κύκλων»

- Μελετήστε πρώτα καλά όλη την εκφώνηση -

Σκοπός της 4^{ης} φάσης

Μετατροπή του επεξεργαστή ενός κύκλου που σχεδιάσατε στην Εργασία#1 σε επεξεργαστή πολλαπλών κύκλων.

Προαπαιτούμενα

Λειτουργική σχεδίαση επεξεργαστή ενός κύκλου

Διεξαγωγή

A) Μετατροπή του datapath του επεξεργαστή ενός κύκλου σε datapath πολλαπλών κύκλων

Μελετήστε το datapath που κατασκευάσατε στην Εργασία#1 και προσθέστε καταχωρητές μεταξύ των βαθμίδων όπου χρειάζεται, ώστε να κρατάτε τις τιμές για κάποια σήματα που παράγονται από μια βαθμίδα και πρέπει να χρησιμοποιηθούν σε επόμενη βαθμίδα. Μπορείτε να κάνετε όσες επιπλέον αλλαγές κρίνετε απαραίτητες, είτε εσωτερικά σε επίπεδο σχεδίασης του datapath είτε σε επίπεδο διεπαφής. Ονομάστε το αρχείο σας **DATAPATH_MC.vhd**.

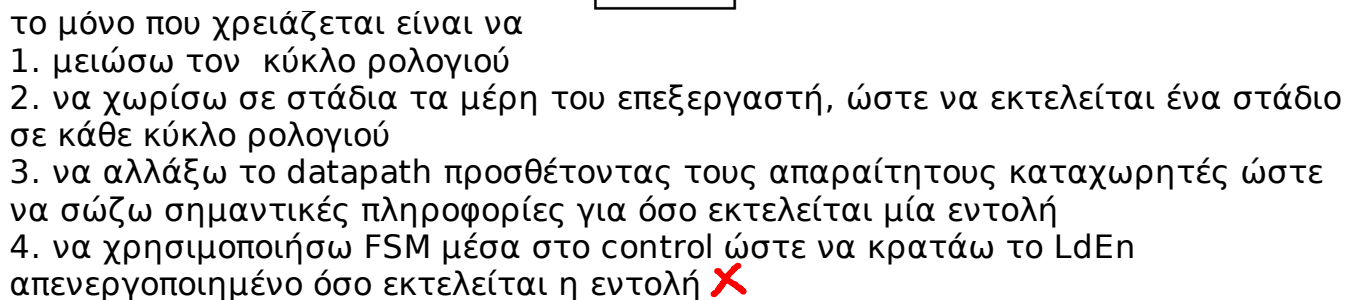
B) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάστε τη μονάδα ελέγχου, η οποία θα είναι μια Μηχανή Πεπερασμένων Καταστάσεων (FSM) που θα ελέγχει τη ροή εκτέλεσης της κάθε εντολής γεννώντας τα απαιτούμενα σήματα ελέγχου σε κάθε κύκλο ρολογιού. Η FSM έχει σαν είσοδο την εντολή (opcode), πιθανώς flags όπως το Zero, κ.λ.π., και σαν εξόδους όλα τα σήματα ελέγχου του Datapath. Η FSM μπορεί να είναι τύπου Moore. Ονομάστε το αρχείο σας **CONTROL_MC.vhd**.

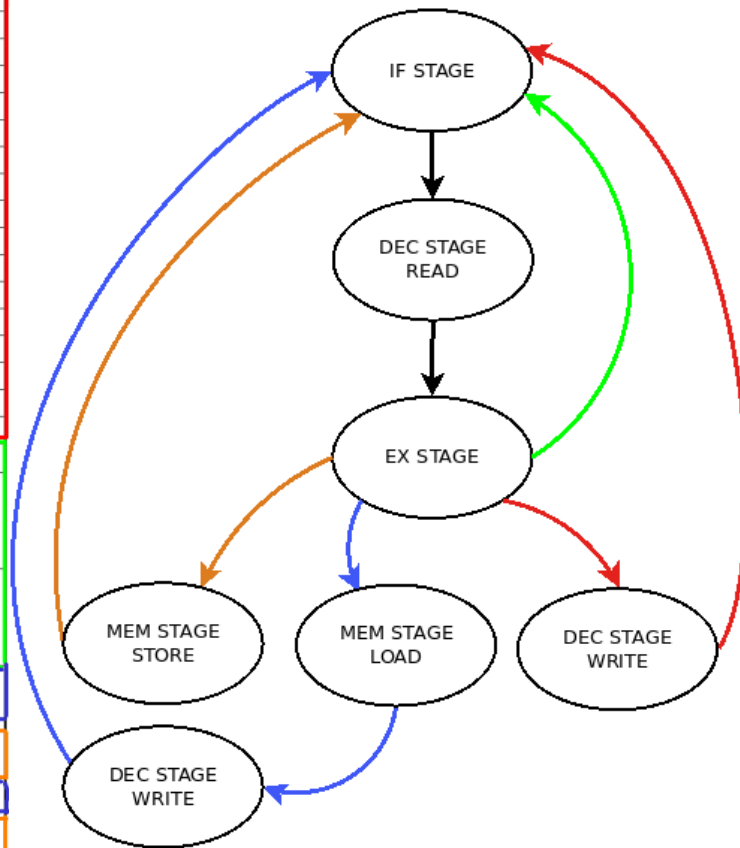
Γ) Ολοκλήρωση και έλεγχος ορθότητας

Συνδέστε το datapath με το control για να υλοποιήσετε την πλήρη λειτουργία ενός επεξεργαστή πολλαπλών κύκλων. Ονομάστε το αρχείο σας **PROCESSOR_MC.vhd**. Η κύρια μνήμη θα πρέπει να

Δείξτε την ορθότητα της σχεδίασης σας με τα προγράμματα αναφοράς της Εργασίας#1 και επίσης δημιουργήστε τουλάχιστον ένα ακόμη δικό σας πρόγραμμα αναφοράς το οποίο θα περιλαμβάνει όλες τις εντολές του ISA τουλάχιστον από μία φορά.



Opcode	FUNC	ΕΝΤΟΛΗ	ΠΡΑΞΗ
100000	110000	add	$RF[rd] \leftarrow RF[rs] + RF[rt]$
100000	110001	sub	$RF[rd] \leftarrow RF[rs] - RF[rt]$
100000	110010	and	$RF[rd] \leftarrow RF[rs] \text{ AND } RF[rt]$
100000	110011	or	$RF[rd] \leftarrow RF[rs] \text{ } RF[rt]$
100000	110100	not	$RF[rd] \leftarrow \neg RF[rs]$
100000	110101	nand	$RF[rd] \leftarrow RF[rs] \text{ NAND } RF[rt]$
100000	110110	nor	$RF[rd] \leftarrow RF[rs] \text{ NOR } RF[rt]$
100000	111000	sra	$RF[rd] \leftarrow RF[rs] \gg 1$
100000	111001	srl	$RF[rd] \leftarrow RF[rs] \gg 1$ (Logical, zero fill MSB)
100000	111010	sll	$RF[rd] \leftarrow RF[rs] \ll 1$ (Logical, zero fill LSB)
100000	111100	rol	$RF[rd] \leftarrow \text{Rotate left}(RF[rs])$
100000	111101	ror	$RF[rd] \leftarrow \text{Rotate right}(RF[rs])$
111000	-	li	$RF[rd] \leftarrow \text{SignExtend}(Imm)$
111001	-	lui	$RF[rd] \leftarrow Imm \ll 16$ (zero-fill)
110000	-	addi	$RF[rd] \leftarrow RF[rs] + \text{SignExtend}(Imm)$
110010	-	nandi	$RF[rd] \leftarrow RF[rs] \text{ NAND } \text{ZeroFill}(Imm)$
110011	-	ori	$RF[rd] \leftarrow RF[rs] \text{ } \text{ZeroFill}(Imm)$
111111	-	b	$PC \leftarrow PC + 4 + (\text{SignExtend}(Imm) \ll 2)$
000000	-	beq	if $(RF[rs] == RF[rd])$ $PC \leftarrow PC + 4 + (\text{SignExtend}(Imm) \ll 2)$ else $PC \leftarrow PC + 4$
000001	-	bne	if $(RF[rs] != RF[rd])$ $PC \leftarrow PC + 4 + (\text{SignExtend}(Imm) \ll 2)$ else $PC \leftarrow PC + 4$
000011	-	lb	$RF[rd] \leftarrow \text{ZeroFill}(31 \text{ downto } 8) \text{ \& } MEM[RF[rs] + \text{SignExtend}(Imm)](7 \text{ downto } 0)$
000111	-	sb	$MEM[RF[rs] + \text{SignExtend}(Imm)] \leftarrow \text{ZeroFill}(31 \text{ downto } 8) \text{ \& } RF[rd](7 \text{ downto } 0)$
001111	-	lw	$RF[rd] \leftarrow MEM[RF[rs] + \text{SignExtend}(Imm)]$
011111	-	sw	$MEM[RF[rs] + \text{SignExtend}(Imm)] \leftarrow RF[rd]$



Σήματα που πρέπει να ελέγχονται από τα states!

PC_LdEn
RF_WrEn
MEM_WrEn
 instr_reg_we
 immedi_reg_we
 A_reg_we
 B_reg_we
 ALU_reg_we
 MEM_data_reg_we

ΕΝΕΡΓΑ
ΣΗΜΑΤΑ

IF STAGE

PC_LdEn
instr_addr_reg_we

DEC STAGE WRITE

immedi_reg_we
RF_WrEn

EX STAGE

ALU_reg_we

MEM STAGE STORE

MEM_WrEn

DEC STAGE READ

A_reg_we
B_reg_we
immedi_reg_we

MEM STAGE READ

MEM_data_reg_we

5^η φάση: χρόνος ολοκλήρωσης 12 ημέρες

«Σχεδίαση επεξεργαστή pipeline»

- Μελετήστε πρώτα καλά όλη την εκφώνηση -

Σκοπός της 5^{ης} φάσης

Μετατροπή του επεξεργαστή πολλαπλών κύκλων σε pipeline επεξεργαστή.

Προαπαιτούμενα

Λειτουργική σχεδίαση επεξεργαστή πολλαπλών κύκλων

Διεξαγωγή

A) Αλλαγές στο datapath του επεξεργαστή πολλαπλών κύκλων

Προσθέστε του κατάλληλους καταχωρητές pipeline όπου κρίνετε ότι χρειάζεται ώστε για όλες τις βαθμίδες του datapath που υλοποιήσατε στην 4^η φάση η έξοδος της κάθε βαθμίδας να γίνεται είσοδος στην επόμενη. Λάβετε επιπρόσθετα υπόψη ότι χρειάζεται προώθηση (forwarding) ή/και stalls για την αντιμετώπιση των κινδύνων δεδομένων (data hazards). Βάσει αυτού υλοποιήστε τις απαιτούμενες επεκτάσεις στη σχεδίαση σας. Ονομάστε το αρχείο σας **DATAPATH_PIPELINE.vhd**.

B) Σχεδιασμός και υλοποίηση της μονάδας ελέγχου (control)

Σχεδιάστε τη μονάδα ελέγχου ώστε να δημιουργεί τα απαιτούμενα σήματα ελέγχου για κάθε εντολή. Μπορείτε να χρησιμοποιήσετε είτε τη μονάδα ελέγχου του επεξεργαστή ενός κύκλου (Εργασία#1) είτε τη μονάδα ελέγχου του επεξεργαστή πολλαπλών κύκλων (4^η φάση) ως σημείο εκκίνησης. Ανάλογα με τη σχεδίαση της μονάδας ελέγχου που θα χρησιμοποιήσετε ως σημείο εκκίνησης, θα χρειαστεί να κάνετε και τις απαιτούμενες αλλαγές. Θα πρέπει να ακολουθήσετε τη θεωρία και να σχεδιάσετε το control για τον pipeline επεξεργαστή ανάλογα με τα παραδείγματα που παρουσιάστηκαν στο μάθημα. Λάβετε υπόψη ότι υπάρχουν και data hazards μεταξύ εντολών που μπορούν να αντιμετωπιστούν με forwarding ή/και με stalls, και πιθανό να χρειαστούν επιπλέον επεκτάσεις στο control (ανάλογα με τη συνολική σχεδίαση του επεξεργαστή σας). Αγνοήστε τα control hazards. Ονομάστε το αρχείο σας **CONTROL_PIPELINE.vhd**.

Προτείνεται να βασιστείτε στην μονάδα ελέγχου του επεξεργαστή ενός κύκλου ως σημείο εκκίνησης.

Γ) Ολοκλήρωση και έλεγχος ορθότητας

Συνδέστε το datapath με το control ώστε να υλοποιήσετε την πλήρη λειτουργία ενός pipeline επεξεργαστή. Ονομάστε το αρχείο σας **PROCESSOR_PIPELINE.vhd**.

Δείξτε την ορθότητα της σχεδίασης σας με ένα δικό σας πρόγραμμα αναφοράς τα οποίο θα περιλαμβάνει μόνο τις εντολές **li**, **lw**, **sw**, **add** πολλές φορές ώστε να φανεί η λειτουργία του pipeline και η αντιμετώπιση των κινδύνων δεδομένων.

Αναλυτική περιγραφή παραδοτέων Εργασίας #2

- Ακολουθήστε τις οδηγίες όπως είναι ακριβώς -

Ο τελικός φάκελος που θα υποβάλετε θα φέρει το επίθετο και τον αριθμό μητρώου σας π.χ. [christodoulou_2019123456](#). **Κάνετε zip τον φάκελο** (συγκεκριμένα zip, και όχι rar ή 7z ή οτιδήποτε άλλο). Προσοχή: το επίθετο σας να είναι σε greeklish και επίσης παρεμβάλλετε το σύμβολο “_” μεταξύ του επιθέτου και του ΑΜ. Εσωτερικά δημιουργείτε 3 υποφακέλους: REPORT, SOURCES, WAVEFORMS, και οι 3 με κεφαλαία γράμματα. Προσέξτε την ονομασία του φακέλου να είναι σωστή και πλήρης. Αν υπάρχει λάθος ο κώδικάς σας δεν θα βαθμολογηθεί.

1. Αναφορά (μέχρι 6 σελίδες, PDF)

Βάλτε την στον υποφάκελο REPORT. . Η αναφορά δεν θα περιλαμβάνει κώδικα, εκτός αν είναι code snippet μέχρι 10 γραμμές και ακολουθείται από τον αντίστοιχο σχολιασμό. Η αναφορά πρέπει να περιλαμβάνει block diagrams που να αποτυπώνουν καλά και περιεκτικά τη δουλειά σας, π.χ. κάποια σήματα και συνδέσεις που θεωρείτε σημαντικά να δείξετε, ή, νέα modules που φτιάξατε που δεν αναφέρονται ρητά στην εκφώνηση.

2. Κώδικας VHDL

Όλα τα αρχεία vhd που έχετε δημιουργήσει στον φάκελο SOURCES, οργανωμένα σε υποφακέλους όπως περιγράφεται παρακάτω.

Περιεχόμενα φακέλου SOURCES

1. Φάκελος **MC**: όλα τα αρχεία vhd για τον επεξεργαστή πολλαπλών κύκλων
2. Φάκελος **PIPELINE**: όλα τα αρχεία vhd για τον pipeline επεξεργαστή

- Κανένας από τους παραπάνω φακέλους δεν πρέπει να περιέχει υποφακέλους.
- Κάθε φάκελος από τους παραπάνω θα περιέχει ένα (1) μόνο testbench : το τελικό με το οποίο ελέγξατε την αντίστοιχη σχεδίαση: τον MC και τον PIPELINE
- Δεν χρειάζεται να υποβάλλετε τα .vhd αρχεία που υποβάλλατε στην Εργασία#1 εκτός αν τα έχετε τροποποιήσει. Σε αυτή την περίπτωση δημιουργήστε ένα επιπλέον φάκελο μέσα στον φάκελο SOURCES με όνομα **SUBMODULES** και βάλτε τα εκεί. Κάθε .vhd αρχείο στο φάκελο SUBMODULES θα πρέπει να περιέχει σε σχόλια μια σύντομη περιγραφή και αιτιολόγηση των αλλαγών που έγιναν, και επίσης σε ένα αρχείο .txt.

3. Κυματομορφές Προσομοίωσης και Προγράμματα Αναφοράς

Στον φάκελο WAVEFORMS βάλτε τις κυματομορφές προσομοίωσης. Θα σας βοηθήσει η χρήση του Waveform Configuration File (.wcfg) που προσφέρει η Xilinx, για ν' αποθηκεύετε τα σήματα σε αρχείο. Στον ίδιο φάκελο θα βρίσκονται και τα προγράμματα αναφοράς που δημιουργήσατε, συνοδευόμενο το καθένα από ένα text αρχείο που θα περιλαμβάνει τις assembly εντολές του προγράμματος αναφοράς. Στον φάκελο WAVEFORMS δεν θα υπάρχουν υποφακέλοι. Σημείωση: το αρχείο .wcfg το κάνετε “Load” μετά από compile, για να εμφανίζονται αυτόματα τα σήματα που θέλετε να παρακολουθείτε στην προσομοίωση.