ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Αρχιτεκτονική Υπολογιστών

Εργαστηριακή Άσκηση 3 (50% εργαστηριακού βαθμού)

Ημερομηνία παράδοσης: 20/6/2021, 23.55

1. Εισαγωγή

Σε αυτήν την Εργαστηριακή Άσκηση θα κατασκευάσετε έναν RISC pipelined CPU. Ήδη έχετε κατασκευάσει αρκετά από τα μέρη τα οποία θα χρησιμοποιεί το CPU σας σε προηγούμενες ασκήσεις. Σε αυτήν την άσκηση θα κατασκευάσετε τα υπόλοιπα και κάνοντας χρήση αυτών και των προηγούμενων, θα συνθέσετε τον ζητούμενο επεξεργαστή.

Είναι καλό να ακολουθήσετε επακριβώς τις συμβουλές της εκφώνησης, αλλά δεν είναι υποχρεωτικό. Τα σημεία όμως που πρέπει να ακολουθήσετε πλήρως είναι αυτά που αφορούν στην λειτουργικότητα της CPU.

2. Επιμέρους κομμάτια

2.1 Στην 1η Εργαστηριακή Άσκηση κατασκευάσετε την ALU του επεξεργαστή. Στην 2η αρκετά από τα λειτουργικά κομμάτια.

2.2 Τώρα πρέπει να φτιάξετε τα υπόλοιπα.

2.2.1 Forwarder (forwarder)

Ελέγχει τις διευθύνσεις (του Reg File) που πρέπει να γραφτούν, αλλά βρίσκονται ακόμα στα στάδια Memory ή Write Back και πρέπει και να διαβαστούν. Εάν δηλαδή ο καταχωρητής που διαβάσαμε είναι ίδιος με το καταχωρητή ενός από αυτά τα στάδια, θα πρέπει να κάνουμε το forwarding. Αλλιώς θα διαβάσουμε τα παλιά δεδομένα. Προφανώς όταν μιλάμε για τον καταχωρητή 0, η εγγραφή δεν πραγματοποιείται, συνεπώς δεν πρέπει να κάνουμε το forwarding.

2.2.2 Επιλογέας (Select)

Ανάλογα με την απόφαση του Forwarder δίνει τη κατάλληλη τιμή (από regFile, από Memory, ή από WriteBack) στην ALU για πρώτη παράμετρο. Χρησιμοποιείται και για τις δύο εισόδους της ALU.

2.2.3 Control (Control)

Ελέγχει την διαδικασία εκτέλεσης των εντολών. Παράγει όλα τα σήματα ελέγχου.
Αποφασίζει για τα παρακάτω:

- Εάν πρόκειται για την εντολή MFPC (move from PC)
- Εάν πρόκειται για εντολή Jump
- Εάν πρόκειται για την εντολή διαβάσματος ψηφίου
- Εάν πρόκειται για την εντολή εκτύπωσης στην οθόνη
- Τον καταχωρητή που θα αποθηκευτεί η πληροφορία-αποτέλεσμα
- Εάν πρόκειται για εντολή τύπου R
- Εάν πρόκειται για την εντολή save word
- Εάν πρόκειται για την εντολή load word
- Εάν πρόκειται για εντολή branch
- Εάν πρόκειται για την εντολή Jump Register

Εάν η εντολή πρέπει να γίνει flush, τότε όλες οι παραπάνω τιμές μηδενίζονται
Σε όλα αυτά θα δίνει απάντηση 0 εάν πρέπει να γίνει flush η εντολή.

2.2.4 Hazard Unit (HazardUnit)

Το Hazard Unit αποφασίζει για το εάν θα πρέπει να γίνει flush η εντολή. Flush γίνεται μια εντολή όταν έχει προηγηθεί ένα branch ή ένα jump, οπότε η τρέχουσα εντολή πρέπει να διαγραφεί. Επίσης αποφασίζει για το εάν η επόμενη εντολή θα πρέπει να γίνει flush. Τέλος αποφασίζει για το ποιο σήμα θα πρέπει να φορτωθεί στον program counter.

2.2.5 Trap Unit (TrapUnit)
Παρακολουθεί το opcode κάθε εντολής και μόλις δεχθεί την εντολή End Of Running αναγκάζει το σύστημα σε ένα διαρκές flush εντολών αλλά και σε «πάγωμα» του PC. Με αυτό τον τρόπο αποφεύγεται η εκτέλεση άκυρων εντολών (dummy instructions).

2.2.6 Register IF_ID
Λαμβάνει ως είσοδο το PC (από τον PC Register) και το Instruction (από το Input) καθώς και τα clock και mode (flush/enable) και επιστρέφει το PC (i) στην ALU (που το χρησιμοποιεί μέσω πολυπλέκτη 2σε1 αν η εντολή είναι MFPC) καθώς και (ii) ξανά πίσω στον PC Register. Αν το enable είναι 1 τότε το PC πάει στην επόμενη εντολή (+2 --16-bit) ενώ αν είναι flush τότε μηδενίζονται τα περιεχόμενα του PC και του outInstruction.

2.2.7 Register ID_EX
Λαμβάνει είσοδο από τον Controller (τον τύπο της εντολής), το Register File και τον SignExtender και δρομολογεί το output τόσο στην ALU, αν π.χ. πρόκειται για R-Type εντολή η στον EX_MEM Register, αν π.χ. πρόκειται για LW/SW ή PrintDigit – έξοδος οθόνης.

2.2.8 Register EX_MEM
Λαμβάνει ως είσοδο τα PrintDigit, ReadDigit, WriteEnable, isLW, από τον ID_EX Register, το clock από την είσοδο, το Result από την ALU και τα RegAD και R2Reg από τον ID_EX Register. Εκχωρεί τις εισόδους στην έξοδο στο rising edge. Το RegAD_EXMEM πάει στον Forwarder και στο Register File και το Result τόσο στην έξοδο όσο και πίσω την είσοδο.

2.2.9 Register MEM_WB
Λαμβάνει ως είσοδο το result (είτε από το keyboard –keyData- αν είναι isReadDigit του controller true) είτε από τη μνήμη –fromData- αν το isLW του controller είναι 1, και τη διεύθυνση του καταχωρητή (RegAD) και επιστρέφει τις ίδιες τιμές στο rising edge. Εκχωρεί τα writeData και writeAD που πάνε στο RegisterFile στα αντίστοιχα write ports.

2.2.10 JRSelector
Πρόκειται για επιλογέα με βάση το JRopcode που λαμβάνει από το HazardUnit και επιλέγει τη διεύθυνση (jump) που θα σταλεί στον PC : Αν 00 τότε περνάει το PCP2ADD όπως έχει υπολογιστεί από τον IF_ID Register (+2 – επόμενη διεύθυνση), αν 01 περνάει το JumpAD, αν 10 περνάει το BranchAD, σε κάθε άλλη περίπτωση περνάει το PCP2AD.

2.2.11 PC Register
Στέλνει τη διεύθυνση εκτέλεσης της τρέχουσας εντολής ή jump ή branch στον IF_ID register (που με τη σειρά του υπολογίζει την επόμενη εντολή (+2) και τη στέλνει στην ALU) καθώς και στην έξοδο (instructionAD).

3. Ένωση

Θα πρέπει χρησιμοποιώντας τα κυκλώματα που έχετε φτιάξει να συνθέσετε το CPU και να κάνετε το simulation στο Quartus.

4. Γενικές Παρατηρήσεις

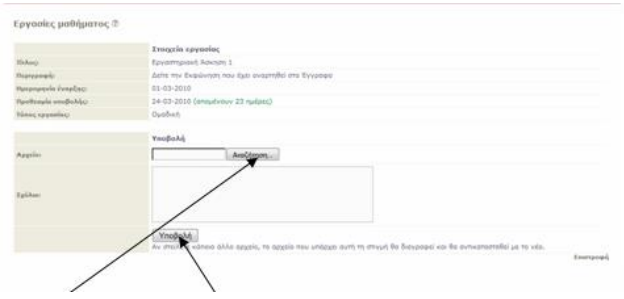
4.1 Ονοματοδοσία

Για να μην υπάρξει σύγχυση με την ονοματοδοσία, παρατίθενται παρακάτω κάποιοι κανόνες, τους οποίους είναι καλό να ακολουθήσετε όσο περισσότερο μπορείτε.

	Ερμηνεία	**Παράδειγμα**
`is*`	Τα σήματα που έχουν boolean ερμηνεία	**`isR`**: Σήμα που φέρει την πληροφορία του εάν η εντολή είναι τύπου R
`*AD`	Η διεύθυνση του αντίστοιχου καταχωρητή	**`reg2AD`**: Η διεύθυνση του 2ου καταχωρητή
`IF,ID,EX,ME,WB`	Τα ονόματα των αντίστοιχων σταδίων του pipelining	**`regADME`**: Η διεύθυνση του καταχωρητή που έρχεται από το στάδιο Memory

5. Γενικές Οδηγίες

- Χρησιμοποιήστε ίδια ονόματα για σήματα, modules κλπ με αυτά που σας δίνονται.
- Μπορείτε να χρησιμοποιήσετε είτε behavioral είτε structural κώδικα, αλλά ενδείκνυται ο behavioral όπου αυτό είναι δυνατόν.
- Για όλα τα προβλήματα πραγματοποιήστε το ανάλογο functional και timing simulation για το πρέπον διάστημα για να επαληθεύσετε την σωστή λειτουργία του κυκλώματός σας.
- Να έχετε σύντομα και περιεκτικά σχόλια στην VHDL όταν χρειάζεται να εξηγήσετε «δύσκολα» κομμάτια κώδικα, ρουτίνες, κλπ.
- Γράψτε (σε σχόλια) στην αρχή κάθε αρχείου τα **ονόματά σας και ΑΜ**.
- Αφού παραδώσετε τον κώδικα, τα RTL διαγράμματα και τα simulations, θα ακολουθήσει προφορική εξέταση.
- Η παράδοση της άσκησης θα πρέπει να γίνει μέσω του e-class από την επιλογή Εργασίες Φοιτητών σύμφωνα με το παρακάτω υπόδειγμα ως ergasia3.zip.



A) Επισύναψη | Υποβολή