ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

HRY 415 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2018-9

Εργαστήριο 1 : Υλοποίηση αλγορίθμου Tomasulo σε VHDL

ΕΚΔΟΣΗ 1.2, Δ. Πνευματικάτος

(Προ) Απαιτούμενα:
Καλή κατανόηση της VHDL, οργάνωσης υπολογιστών και του αλγορίθμου Tomasulo.

Περιγραφή:
Στα επόμενα εργαστήρια θα υλοποιήσετε τμηματικά τον πυρήνα ενός επεξεργαστή με δυναμική ομοχειρία (pipeline) βασισμένη στον αλγόριθμο του Tomasulo. Τα βασικά στοιχεία της είναι: (α) Reservation Stations, Functional Units, Register File (Register Result Status), Common Data Bus (CDB), και ο έλεγχος των παραπάνω. Η σχεδίαση θα πρέπει να γίνει δομημένα ώστε να επιτρέπονται αλλαγές αργότερα (και συγκεκριμένα η εισαγωγή ενός Reorder Buffer και μιας μονάδας πρόσβασης μνήμης).

Οι εντολές εισάγονται στο σύστημα εξωτερικά, μία-μία, και μόνο αν γίνουν δεκτές από τον έλεγχο του συστήματος.

Ο έλεγχος και η πρόσβαση στο Common Data Bus είναι pipelined σε δύο κύκλους. Στον πρώτο –ένα κύκλο πριν την ολοκλήρωση της πράξης– θα γίνεται έλεγχος για διαιτησία πρόσβασης (access arbitration) και στον επόμενο (δεύτερο) κύκλο θα γίνεται και κοινοποίηση (broadcast) του tag (Q) από την επιλεγμένη λειτουργική μονάδα στο υπόλοιπο σύστημα, καθώς και η κοινοποίηση του αποτελέσματος (V).

Ζητούμενο Πρώτου Εργαστηρίου: Σχεδίαση βασικών κομματιών του συστήματος
1) Reservation Stations + Functional Unit + τοπικός έλεγχος
2) Register File
3) Common Data Bus (CDB) + έλεγχος
4) Έκδοση (Issue) εντολών στο σύστημα

1) RS + FU + Έλεγχος

Reservation Station
Σχεδιάστε ένα reservation station (RS) το οποίο έχει όλα τα κατάλληλα πεδία. Ορίστε πώς «γεμίζει» το RS κατά την έκδοση εντολών στο σύστημα. Προσθέστε λογική η οποία «ακούει» το πεδίο Q του CDB (CDB.Q), και αποφασίζει αν πρέπει να συγκρατήσει την τιμή που υπάρχει στο CDB.V. Προσθέστε μια έξοδο η οποία δείχνει κατά πόσο το RS είναι έτοιμο για εκτέλεση. Προσθέστε επίσης μια έξοδο η οποία δίνει τα περιεχόμενα του RS προς την λειτουργική μονάδα (Functional Unit –FU) εάν επιλεγεί το συγκεκριμένο RS για εκτέλεση. Θεωρήστε ότι τα πεδία Q έχουν πλάτος 5 bit.

Functional Unit
Υπάρχουν δύο τύποι FU στο σύστημα, μια για λογικές πράξεις OR/AND/NOT με καθυστέρηση 2 κύκλων, και μια για αριθμητικές πράξεις (+/-)με καθυστέρηση 3 κύκλων. Οι λειτουργικές μονάδες είναι πλήρως ομόχειρες (pipelined) και εφόσον δεν είναι απασχολημένες δέχονται τα ορίσματα (πεδία V), τον κωδικό πράξης και το tag που θα χρησιμοποιήσουν για την κοινοποίηση του αποτελέσματος στο CDB, και ξεκινούν την πράξη.
Ένα κύκλο πριν την ολοκλήρωση της πράξης η (κάθε) λειτουργική μονάδα ζητάει από τον έλεγχο του CDB άδεια πρόσβασης. Εφόσον η αίτηση δίνει δεκτή μέσω ενός σήματος Grant.i (i = FU) από τον έλεγχο του CDB, στον επόμενο κύκλο κοινοποιεί το tag στο CDB.Q, και το αποτέλεσμα της πράξης στο CDB.V.

Τοπικός έλεγχος RS + FU
Ο τοπικός έλεγχος RS και FU αναλαμβάνει να πάρει όλες τις «τοπικές» αποφάσεις: γνωρίζει την κατάσταση των RS και δίνει σήμα Available/Full προς την λογική έκδοσης εντολών (φτάνει ένα bit ναι/όχι για την έκδοση εντολών;). Εφόσον υπάρχει έτοιμη εντολή, ελέγχει αν η FU μπορεί να δεχτεί νέα πράξη και εάν ναι, διαβάζει (ενεργοποιεί την ανάγνωση) το RS και ξεκινά την εκτέλεση της εντολής.

Ολοκλήρωση επιμέρους κομματιών
Χρησιμοποιώντας τα παραπάνω συστατικά, φτιάξτε δύο πλήρεις λειτουργικές μονάδες, μια για λογικές πράξεις με 2 RS και μια για αριθμητικές πράξεις με 3 RS. Το πεδίο Q που γράφεται στο CDB έχει πλάτος 5 bits και αποτελείται από 2 περισσότερο σημαντικά bits που δείχνουν τον κωδικό της λειτουργικής μονάδας (πως γνωρίζει η κάθε λειτουργική μονάδα τον κωδικό της;) και 3 bits που δίνουν το συγκεκριμένο RS εντός της λειτουργικής μονάδας.

2) Register File (+ Register Result Status)
Υλοποιήστε το αρχείο καταχωρητών, το οποίο διαβάζεται χρησιμοποιώντας την διεύθυνση του καταχωρητή (register specifier), αλλά γράφεται «διαλέγοντας» τιμές από το CDB (Q και V). Η οργάνωση είναι παρόμοια με αυτή των RS με λιγότερα πεδία. Η register file έχει δύο πόρτες ανάγνωσης για να διαβάζει τους δύο καταχωρητές κατά την διάρκεια της έκδοσης εντολών. Κατά την έκδοση εντολής, το αρχείο καταχωρητών γράφει στο πεδίο Q του καταχωρητή προορισμού της εντολής που εκδίδεται τον κωδικό του RS στο οποίο εκδίδεται η εντολή. Σχεδιάστε ένα καταχωρητή όπως και στην περίπτωση των RS, και μετά δημιουργήστε το αρχείο με αντίγραφα (στιγμιότυπα) αυτού του καταχωρητή και την όποια απαιτούμενη λογική ελέγχου. (Αν και «αρχείο καταχωρητών», η δομή αυτή δεν χρησιμοποιεί μνήμη αλλά ανεξάρτητους καταχωρητές και λογική).
Η RF πρέπει να υλοποιεί την λειτουργία «fall-through» δηλαδή αν στον ίδιο κύκλο έχουμε εγγραφή σε ένα καταχωρητή (δηλαδή match με το CDB.Q) και ανάγνωση του ιδίου καταχωρητή διαβάζονται τα νέα δεδομένα (CDB.V) και το tag διαβάζεται μηδέν.

3) Common Data Bus (CDB) + έλεγχος
Υλοποιήστε το CDB, δηλαδή το datapath και τον έλεγχο του.
Το CDB έχει μια πόρτα σύνδεσης για κάθε λειτουργική μονάδα και μια για την Register File (μόνο εξόδου). Θεωρήστε ότι υπάρχουν 3 λειτουργικές μονάδες, και αφήστε την μία αχρησιμοποίητη.
Το datapath του CDB ενσωματώνει τους πολυπλέκτες που διαλέγουν ποιος μεταδίδει πληροφορία στα καλώδια του CDB.
Ο έλεγχος του CDB δέχεται αιτήσεις από κάθε μονάδα που θέλει να χρησιμοποιήσει το CDB και επιλέγει μία από αυτές για πρόσβαση. Εφόσον υπάρχουν περισσότερες της μιας αιτήσεις, η εξυπηρέτηση θα γίνεται κυκλικά μεταξύ των αιτούντων (τεχνική round-robin).

4) Έκδοση (Issue) εντολών στο σύστημα

Υλοποιήστε την λογική έκδοσης εντολών. Η είσοδος είναι μια αποκωδικοποιημένη εντολή και μια αίτηση Issue. Η λογική έκδοσης ελέγχει δομικούς κινδύνους και εφόσον η εντολή μπορεί να γίνει δεκτή προχωράει σε:

- Αποδοχή της εντολής για έκδοση (απάντηση Accepted προς την μονάδα IF που βρίσκεται στον έξω κόσμο)
- Ανάγνωση των καταχωρητών από την RF (συνδυαστικό κύκλωμα, απάντηση εντός του κύκλου)
- Επιλογή, δέσμευση και «γέμισμα» του κατάλληλου RS
- Ενημέρωση του αρχείου καταχωρητών για τον καταχωρητή προορισμού

Διεπαφή με την εξωτερική μονάδα IF (την οποία δεν υλοποιείτε)

Σήμα	Πλάτος	Είδος	Περιγραφή
Issue	1bit	Είσοδος	Υπάρχει εντολή για έκδοση
FU-type	2 bit	Είσοδος	Τύπος εντολής, 00 = Λογική πράξη, 01 = Αριθμητική πράξη, 1x δεσμευμένοι κωδικοί
Fop	2 bit	Είσοδος	Κωδικός Πράξης: Λογική πράξη: 00, 01, 10 = OR, AND, NOT Αριθμητική πράξη: 00 = «+», 01 = «-», 01 = « << » Υπόλοιποι κωδικοί δεσμευμένοι
Ri	5 bit	Είσοδος	Αριθμός καταχωρητή προορισμού
Rj	5 bit	Είσοδος	Αριθμός καταχωρητή πηγής #1
Rk	5 bit	Είσοδος	Αριθμός καταχωρητή πηγής #2
Accepted	1 bit	Έξοδος	Η εντολή έγινε δεκτή (εκδόθηκε)

Παράδειγμα

Θεωρήστε για παράδειγμα ότι η εντολή υπό αποκωδικοποίηση είναι η add r1, r2, r3 (issue = 1, FU-type = 01, Fop = 00, Ri = 1, Rj = 2, Rk = 3)

ΕΚΤΕΛΕΣΗ ΕΡΓΑΣΤΗΡΙΟΥ ΚΑΙ ΠΑΡΑΔΟΤΕΑ:

1. Ορίστε τις διεπαφές των μονάδων
2. Δώστε ένα σχηματικό διάγραμμα του όλου συστήματος και των επιμέρους μονάδων.
3. Δώστε ένα διάγραμμα χρονισμού του όλου συστήματος και των επιμέρους μονάδων για την εκτέλεση μιας εντολής.
4. Υλοποιήστε τις ανεξάρτητες μονάδες, και ελέγξτε τις ανεξάρτητα.
5. Η αναφορά πρέπει να περιλαμβάνει: (α) τα (1) - (3) παραπάνω, όπου τα σχηματικά διαγράμματα θα έχουν το σωστό επίπεδο λεπτομέρειας για την γενική κατανόηση λειτουργικότητας και χρονισμού αλλά χωρίς να «πνίγονται» στην λεπτομέρεια, (β) σύντομη περιγραφή σε κατανοητή μορφή (δηλ. όχι «χύμα» διάγραμμα χρονισμού από τον προσομοιωτή) της προσέγγισης ελέγχου της κάθε μονάδας, τι εισόδους εισάγετε και τι περιμένετε να δείτε στην έξοδο, (γ) τεκμηρίωση της λειτουργικότητας της κάθε ανεξάρτητης μονάδας.