ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ
ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ: ΗΡΥ 312
ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017

Εργαστήριο 6

Προσθήκη εξαιρέσεων και κρυφής μνήμης

Σκοπός του Εργαστηρίου

1) Δημιουργία ενός pipelined επεξεργαστή που μπορεί να αντιμετωπίζει εξαιρέσεις (exceptions).
2) Προσθήκη κρυφής μνήμης δεδομένων (data cache).

Υλοποίηση Εξαιρέσεων (50%)

Οι εξαιρέσεις αφορούν απρόσμενα συμβάντα κατά την λειτουργία ενός επεξεργαστή, πχ. η εκτέλεση άγνωστης εντολής. Για να διαχειριστεί το λειτουργικό σύστημα τέτοιου είδους γεγονότα, μόλις συμβεί κάποιο exception, ο program counter μεταβαίνει σε μία προκαθορισμένη διεύθυνση και εκτελεί τον exception handler του λειτουργικού συστήματος.

Στη γενική περίπτωση, χρησιμοποιούνται δύο επιπλέον καταχωρητές, ο EPC και ο Cause Register. Ο EPC αφορά τη διεύθυνση της εντολής στην οποία πραγματοποιήθηκε το exception, ενώ ο Cause Register αφορά το λόγο της εξαίρεσης. Τα exceptions που καλείστε να υλοποιήσετε, οι αντίστοιχες διευθύνσεις στις οποίες θα μεταβαίνει η εκτέλεση του προγράμματος όταν αυτά συμβούν καθώς και οι τιμές του cause register παρουσιάζονται στον ακόλουθο πίνακα:

Ονομα	Αιτιολογία	Διεύθυνση handler	Τιμή Cause Register
IBUS	Άγνωστη εντολή (Άγνωστο Opcode/Func)	0x100	0x00001111
ADDRL	Λανθασμένη διεύθυνση ανάγνωσης (διεύθυνση > 10 bits)	0x200	0x11110000

Για το εργαστήριο αυτό θα πρέπει να υλοποιήσετε έναν pipelined επεξεργαστή που θα εκτελεί μόνο τις εντολές **LW, ADD**, **LI και JUMP_EPC**. Η λειτουργία καθώς και το format των εντολών θα είναι αυτό που περιγράφτηκε στα προηγούμενα εργαστήρια.

Η κωδικοποίηση της εντολής **JUMP_EPC** γίνεται σύμφωνα με τον ακόλουθο πίνακα:

Opcode	FUNC	ΕΝΤΟΛΗ	ΠΡΑΞΗ
101010	-	jump_epc	PC ← EPC + 4

Στην περίπτωση που προκληθεί κάποια εξαίρεση, θα πρέπει να αποθηκεύεται στον EPC η διεύθυνση της εντολής που προκάλεσε την εξαίρεση και στη συνέχεια να φορτώνεται στον PC η αντίστοιχη τιμή του exception handler. Επίσης, στον Cause register θα πρέπει να αποθηκεύεται η τιμή που προκάλεσε το exception. Έτσι, η εκτέλεση της εντολής που προκάλεσε το exception θα πρέπει να διακόπτεται και το pipeline θα πρέπει να “αδειάζει” από όλες τις επόμενες εντολές.

Ο EPC και ο Cause Register στην περίπτωσή μας θα είναι δύο registers 32 bits που θα βρίσκονται στη βαθμίδα Decode Stage. Στην περίπτωση που συμβεί κάποιο exception, θα πρέπει

η τιμή του PC της εντολής που προκάλεσε το exception να γραφτεί στον καταχωρητή EPC, ο λόγος που προκάλεσε το exception να γραφτεί στον Cause Register, να ακυρωθούν όλες οι εντολές που μπήκαν στο pipeline μετά την εντολή που προκάλεσε το exception και το PC να πάρει την κατάλληλη διεύθυνση του handler σύμφωνα με τον παραπάνω πίνακα.

Για να είναι εφικτή η επιστροφή από τον exception handler και η συνέχιση του προγράμματος θα χρησιμοποιείται η εντολή jump_epc.

Υλοποίηση Κρυφών Μνημών (50%)

Σε αυτό το σημείο θα πρέπει να αντικαταστήσετε την Data Memory του παραπάνω επεξεργαστή από ένα επίπεδο κρυφής μνήμης δεδομένων (cache). Η cache που καλείστε να υλοποιήσετε είναι **Direct- mapped associative** και write through (αλλά αυτό δεν θα πρέπει να σας απασχολεί από την στιγμή που δεν υλοποιείτε την εντολή store). Η cache θα έχει **32 lines (sets)** και το κάθε set θα αποτελείται συνολικά από 1 block. Κάθε Block της cache θα περιέχει **16 bytes**.

Η παραγόμενη διεύθυνση που θα κάνει πρόσβαση στην cache χωρίζεται στα πεδία: tag, set-index, kai block-offset (το οποίο το σπάμε εσωτερικά σε Word offset και byte offset).

ΖΗΤΟΥΜΕΝΟ: Χωρίστε τα 12 Least Significant Bits (LSB) της διεύθυνσης μνήμης στα αντίστοιχα πεδία.

Αντικαταστήστε την Data Mem της 4ης βαθμίδας του επεξεργαστή σας από μία cache με τα παραπάνω χαρακτηριστικά. Όταν φτάνει στην βαθμίδα MEM μία νέα διεύθυνση ανάγνωσης, ο επεξεργαστής θα πρέπει (1ος κύκλος) να ελέγχει εάν το δεδομένο βρίσκεται στην cache και στην περίπτωση που βρίσκεται (2ος κύκλος) να προκύπτουν τα νέα δεδομένα στην έξοδο της βαθμίδας MEM. Αν πάλι τα δεδομένα δεν υπάρχουν τότε απλά στον 2ο κύκλο να “σηκώνεται” ένα σήμα miss (έξοδος από τον επεξεργαστή σας) και η εκτέλεση των εντολών να συνεχίζεται κανονικά χωρίς να εκτελείται η εγγραφή οποιασδήποτε τιμής στην RF για την συγκεκριμένη εντολή. Αν αντιθέτως υπάρχει η ζητούμενη τιμή στην cache τότε θα γίνεται η ανάγνωση της και η εκτέλεση της εντολής θα προχωράει κανονικά. Η μορφή της cache παρουσιάζεται παρακάτω και θα πρέπει να προσθέσετε ότι παραπάνω λογική χρειάζεται για τη σωστή λειτουργία της, πχ. πολυπλέκτες, AND και OR πύλες. Η cache όπως παρουσιάζεται παρακάτω αποτελείται από 3 βασικά modules (2 modules αποτελούν κάθε block της cache και 1 module για το LRU). Τα Blocks της cache είναι Single Port RAM με τα παρακάτω χαρακτηριστικά.

Lines	1 bit	**;;;** bits	128 0bits
0	**Valid**	**Tag**	**Data**
…			
31			

Valid Bit: Γίνεται 1 εάν η γραμμή αυτή περιέχει έγκυρα δεδομένα (αρχικοποίηση σε 0). Αυτό θα σας δίνεται διότι το datapath σας δεν θα υλοποιεί την store.

Tag: Τα **;;;** MS bits της διεύθυνσης της μνήμης δεδομένων. (Πρέπει να βρείτε εσείς ποσα bit χρειάζονται για tag με βάση την θεωρία που έχει διδαχθεί.

Data: Τα δεδομένα του κάθε set της μνήμης δεδομένων .

ΠΡΟΣΟΧΗ!!!! Σύμφωνα με την παραπάνω διαδικασία ο επεξεργαστής σας θα υλοποιεί 6 stages pipeline. Επίσης, υλοποιήστε την cache έτσι όπως περιγράφεται παραπάνω γιατί θα σας δοθούν συγκεκριμένα data αρχεία.

Καλή επιτυχία