

## Αναφορά Εργαστηρίου 6

Κωδικός Ομάδας **LAB31231465**

Διαλεκτάκης Γιώργος
Βαϊλάκης Απόστολος Νικόλαος

### Περιγραφή Άσκησης

Σκοπός της 6ης εργαστηριακής άσκησης ήταν η σχεδίαση ενός pipelined επεξεργαστή που αντιμετωπίζει εξαιρέσεις και πιο συγκεκριμένα όταν υπάρχει άγνωστη εντολή ή λανθασμένη διεύθυνση και η υλοποίηση κρυφής μνήμης (Cache).

#### **Εξαιρέσεις:**

Τις εξαιρέσεις χειρίζεται η μονάδα ελέγχου η οποία βρίσκεται στο στάδιο ID του pipeline επεξεργαστή μας.

Το Control αντιλαμβάνεται αν υπάρχει Exception και αν ναι τότε αν οφείλεται σε λανθασμένη εντολή (δηλαδή opcode ή func) ή σε λανθασμένη διεύθυνση(>12 bits) όταν εκτελείται πράξη που περιλαμβάνει τη μνήμη. Η μονάδα ελέγχου βγάζει ως έξοδο ένα σήμα EX το οποίο αν είναι '1' τότε υπάρχει εξαίρεση και ένα σήμα ExSel το οποίο δείχνει τι τύπου Exception έχουμε. Πιο συγκεκριμένα το Control επικοινωνεί με την IF Stage και ελέγχει έναν πολυπλέκτη της με το σήμα ExSel για να πει στον Program Counter ποια διεύθυνση να πάρει,

την 0x100 ή 0x200 που αντιστοιχούν στις διευθύνσεις handler για λανθασμένη εντολή ή διεύθυνση αντίστοιχα. Επίσης, η μονάδα ελέγχου αποθηκεύει σε έναν καταχωρητή και συγκεκριμένα τον EPC την διεύθυνση της εντολής που υπήρχε εξαίρεση με σκοπό να επανέρθει ο επεξεργαστής και ο PC στην διεύθυνση αυτή αφότου εκτελέσει τον Exception Handler. Αυτό το καταφέρνει μέσω της εντολής JUMP\_EPC ( $PC \leftarrow EPC + 4$ ).

Τέλος, το σήμα EX δίνεται από τη μονάδα ελέγχου και στον pipeline καταχωρητή IF/ID έτσι ώστε να μετατρέψει την εντολή που προκάλεσε εξαίρεση σε μία NOP και να αδείασει το pipeline από όλες τις επόμενες εντολές.

#### **Cache:**

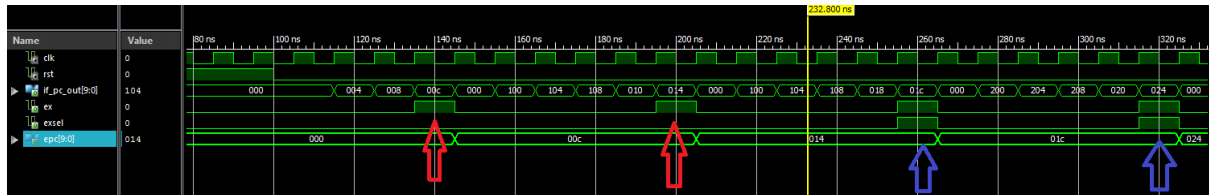
Για την υλοποίηση της κρυφής μνήμης προστέθηκε ένας pipeline register ανάμεσα στον EX/MEM και MEM/WB. Επίσης δημιουργήθηκε ένα νέο module με όνομα Cache Stage το οποίο δέχεται την διεύθυνση(12 bits) η οποία μέχρι τώρα πήγαινε στην Memory(η οποία αντικαταστήθηκε από την Cache) και την σπάει σε 3 bits TAG, 5 bits Set Index, 2 bits word

Offset και 2 bits Byte Offset. Τα word offset και byte offset χρησιμοποιούνται ούτως η άλλως για την ανάκτηση των κατάλληλων δεδομένων από την cache (πράγμα το οποίο και δεν μας κοστίζει περισσότερους κύκλους). Τα δεδομένα αυτά ανακτούνται από το block της κρυφής μνήμης (χρησιμοποιώντας το Set Index) το οποίο και αποτελείται από τέσσερα words, ένα tag των 3-bit και ένα valid flag του ενός bit. Ενώ λοιπόν η κρυφή μνήμη ανακτά τα δεδομένα που της ζητήθηκαν, χρησιμοποιεί δύο ελέγχους για τον έλεγχο της εγκυρότητας τους. Πρώτος έλεγχος είναι το valid bit του block δεδομένων το οποίο και πρέπει να είναι '1'. Δεύτερος έλεγχος είναι η σύγκριση του tag που προσκομίσθηκε από το Memory Address με αυτό που βρίσκεται στο block που ανακτήθηκε από την cache, τα οποία και πρέπει να είναι ίδια. Στην περίπτωση που δεν πληρούνται όλες οι παραπάνω προϋποθέσεις το Cache module βγάζει '1' στο σήμα miss της εξόδου του.

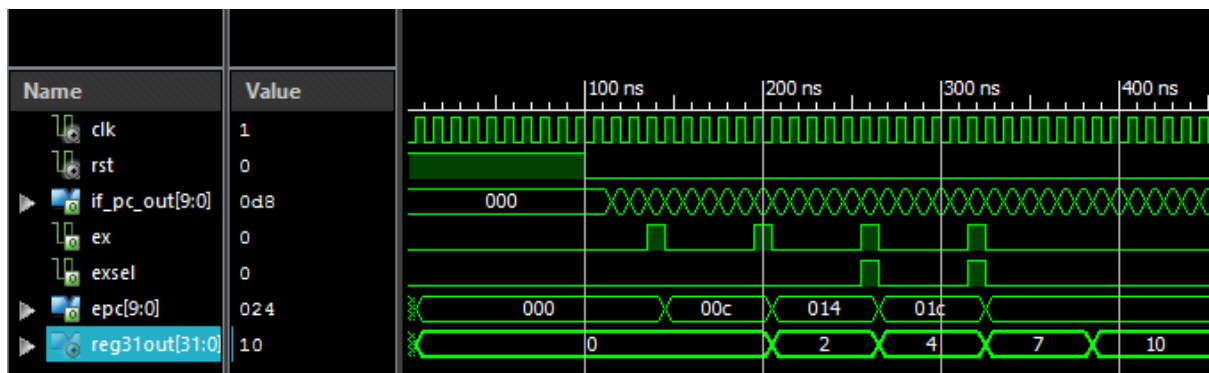
Αυτό το σήμα έπειτα χρησιμοποιείται στο επόμενο pipeline stage. Αν το σήμα miss είναι '1' και ταυτόχρονα κάνουμε πράξη lw τότε το σήμα rf\_wren μηδενίζεται έτσι ώστε να μην καταγραφούν λάθος δεδομένα στην register file.

## Κυματομορφές :

### A) Exceptions :

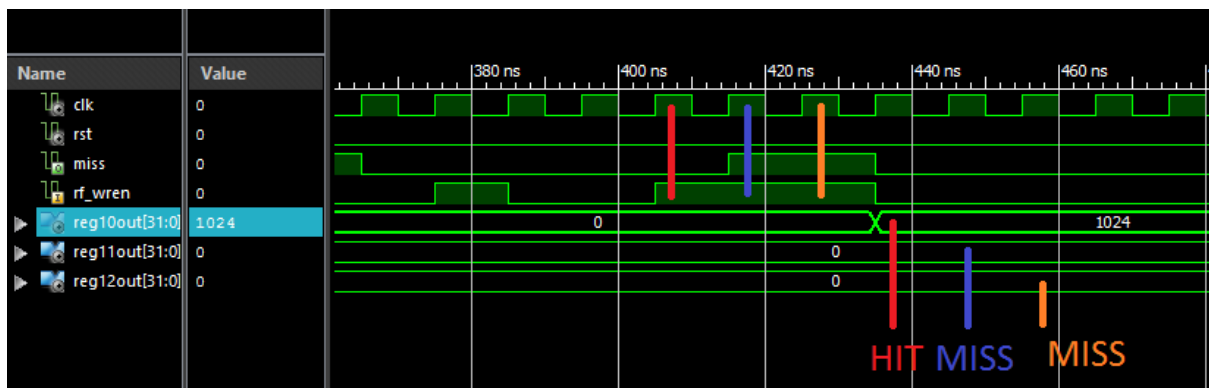


Στην παραπάνω κυματομορφή βλέπουμε διαδοχικά δύο wrong instruction (με κόκκινο) και δύο wrong address exceptions (με μπλε). Ακόμη μπορούμε να δούμε ότι η εντολή που έρχεται απο τον IF/ID register μηδενίζεται και επίσης οι επόμενες εντολές μετά από κάθε exception αντιστοιχούν στην θέση του κάθε exception handler. Τέλος μπορούμε να δούμε το σήμα EPC το οποίο μετά από κάθε εξαίρεση αντιστοιχεί στην θέση του PC που την προκάλεσε.



Συνεχίζοντας με τις εξαιρέσεις μπορούμε να δούμε και τις μεταβολές που υπέστη ο register31 μετά από τα exception (δυσ προσθέσεις +2 και έπειτα δυο προσθέσεις +3).

### B) Cache :



Παραπάνω μπορούμε να δούμε τρεις διαδοχικές lw, η πρώτη εκ των οποίων είναι HIT, ενώ οι δύο επόμενες MISS επίσης μπορούμε να δούμε ότι ενώ το σήμα rf\_wren είναι και στις 3 διαδοχικές lw '1', το οποίο κανονικά θα οδηγούσε τους registers 11 και 12 να πάρουν την τιμή 1024, οι registers αυτοί δεν αλλάζουν τιμή.

## Συμπεράσματα

Έτσι λοιπόν φτάσαμε στην ολοκλήρωση ενός pipeline επεξεργαστή ο οποίος εκτελεί 6 εντολές παράλληλα και υποστηρίζει την αντιμετώπιση εξαιρέσεων αλλά και την χρήση κρυφής μνήμης. Αυτή η υλοποίηση επεξεργαστή αντιπροσωπεύει καλύτερα τις τεχνικές και αρχιτεκτονικές που χρησιμοποιούνται στην σχεδίαση των σημερινών επεξεργαστών (αν και ακόμη μίλια μακριά από μία χρήσιμη υλοποίηση).