ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
Οργάνωση Υπολογιστών

Ομάδα: LAB31235453
Μπαλαμπάνης Ηλίας 2014030127
Μποκαλίδης Αναστάσιος 2014030069

Αναφορά Εργαστηρίου 6

Περιγραφή Άσκησης

Στο 6ο και τελευταίο εργαστήριο μας ζητήθηκε να υλοποιήσουμε 2 ακόμη πράγματα. Πρώτον, ένα κύκλωμα που θα ελέγχει πιθανές εξαιρέσεις (exceptions) και μία μνήμη Cache η οποία μας έδινε διευθύνσεις και λέξεις, απαραίτητες για το πρόγραμμα.

1) Exceptions

Υπάρχουν τρεις πιθανές εξαιρέσεις. Η πρώτη περίπτωση είναι για λάθος instruction, δηλαδή μια λάθος εντολή από την μνήμη. Λάθος σημαίνει ότι δεν μπορεί να την εκτελέσει ο επεξεργαστής μας, του είναι άγνωστη. Σε αυτή την περίπτωση το πρόγραμμα μας πηγαίνει στην διεύθυνση 0x100 και γράφεται στον cause register η τιμή 0x00000111. Αυτό γίνεται με την βοήθεια της νέας εντολής JUMP_EPC. Αντίστοιχα, αν είχαμε μια λάθος διεύθυνση ανάγνωσης, δηλαδή μια διεύθυνση μεγαλύτερη του 2048. Τότε, η JUMP_EPC έστελνε το πρόγραμμα μας στην 0x200 και γράφεται στον cause register η τιμή 0x00111000. Τέλος, αν τύχαινε μια υπερχείληση σε μία πρόσθεση, τότε η JUMP_EPC έστελνε το πρόγραμμα μας στην 0x300 και γράφεται στον cause register η τιμή 0x11000000.

2) Μνήμη Cache

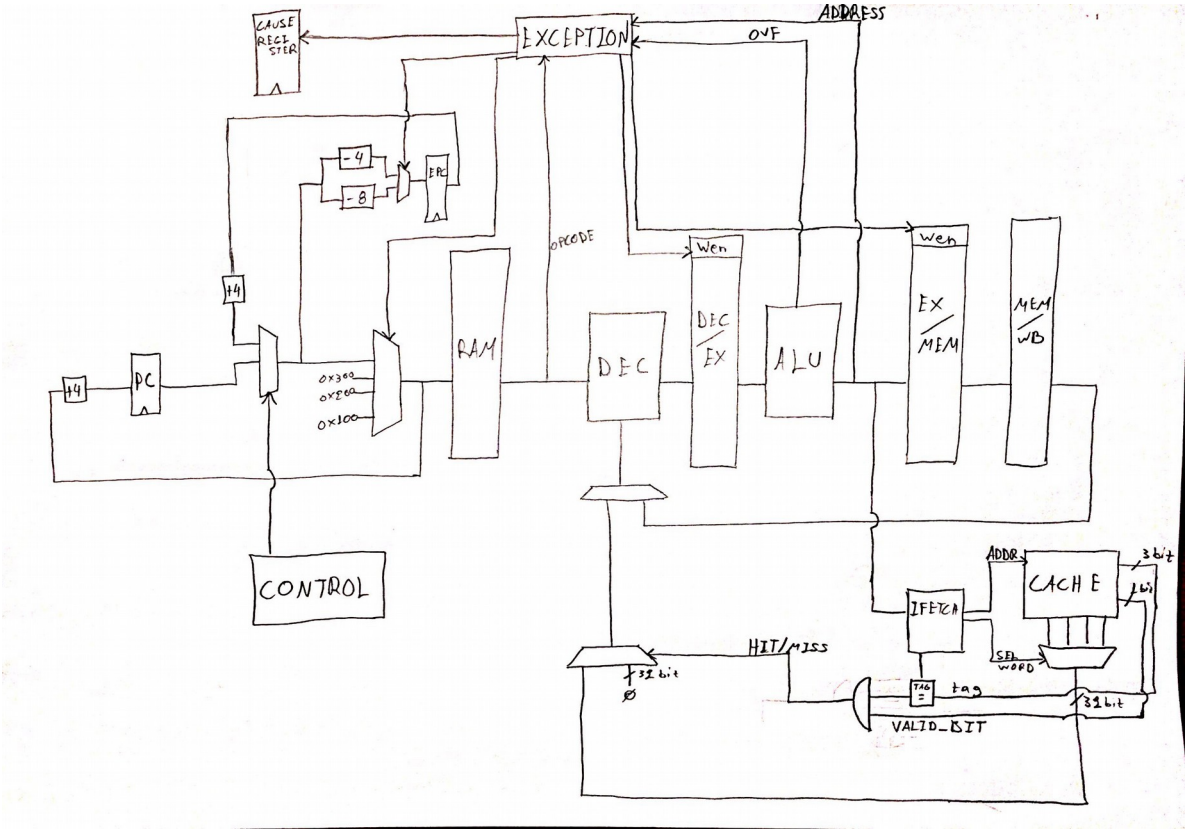
Η κρυφή μνήμη είναι ένας ειδικός τύπος μνήμης που έχει σκοπό να αυξάνει τις επιδόσεις μια αρχιτεκτονικής ενός επεξεργαστή. Για την υλοποίησή της στα πλαίσια του εργαστηρίου, εργαστήκαμε ως εξής:

Αρχικά υλοποιήσαμε μία νέα κρυφή μνήμη δεδομένων (cache), η οποία έχει 32 lines (sets) και το κάθε set αποτελείται συνολικά από 1 block μεγέθους 16 bytes. Η παραγόμενη διεύθυνση που θα κάνει πρόσβαση στην cache, χωρίζεται στα εξής πεδία:

- Tag

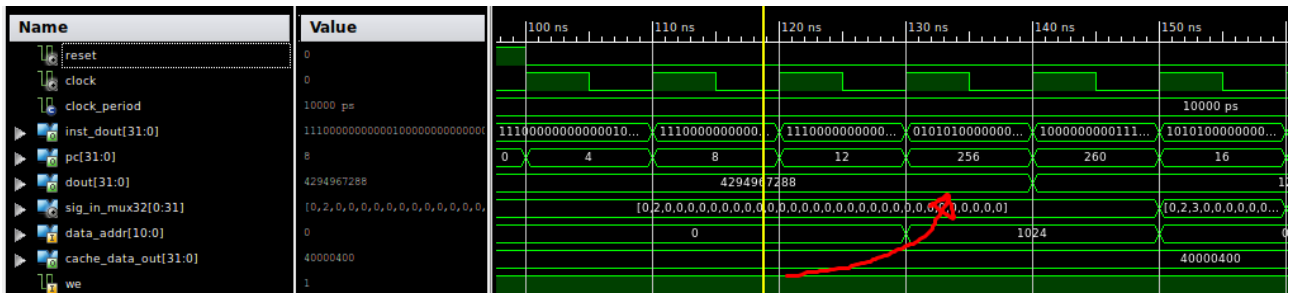
- Set index
- Word offset
- Byte offset

Πρόκειται να χωρίσουμε τα 12 LSB της διεύθυνσης μνήμης στα αντίστοιχα πεδία, όπως ζητήθηκε. Ουσιαστικά, το σήμα valid δεσμεύει 1 bit, άρα έχουμε στη διάθεσή μας 11 ακόμη bits. Γνωρίζουμε, αρχικά, πως το μέγεθος λέξης είναι 4 bytes, επομένως μπορούμε να βρούμε τον αριθμό word/block : block size/word size = 16 byte / 4byte = 4bytes. Άρα, για να επιλέξουμε ποια λέξη θέλουμε στην έξοδο χρειαζόμαστε log 2 (4) =2 bits, τα οποία αντιστοιχούν στο πεδίο word offset. Αντίστοιχα, για το πεδίο byte offset, βρίσκουμε το byte/word = 4bytes, άρα χρειαζόμαστε 2bits. Για τη διευθυνσιοδότηση στις 32 lines της κρυφής μνήμης χρειαζόμαστε log 2 (32) = 5 bits, τα οποία αφορούν το πεδίο set index. Γνωρίζουμε πως πρόκειται για word-addressable μνήμη, επομένως αγνοούμε το κομμάτι που αφορά το byte offset. Τέλος, στο πεδίο tag θα αντιστοιχίσουμε τα υπολειπόμενα bits, έχοντας λάβει υπόψιν τους προηγούμενους υπολογισμούς και θεωρήσεις, επομένως θα έχει μήκος 3 bits. Παρακάτω βλέπουμε τον τελικό διαχωρισμό όπως αξιοποιήθηκε στην υλοποίησή μας:

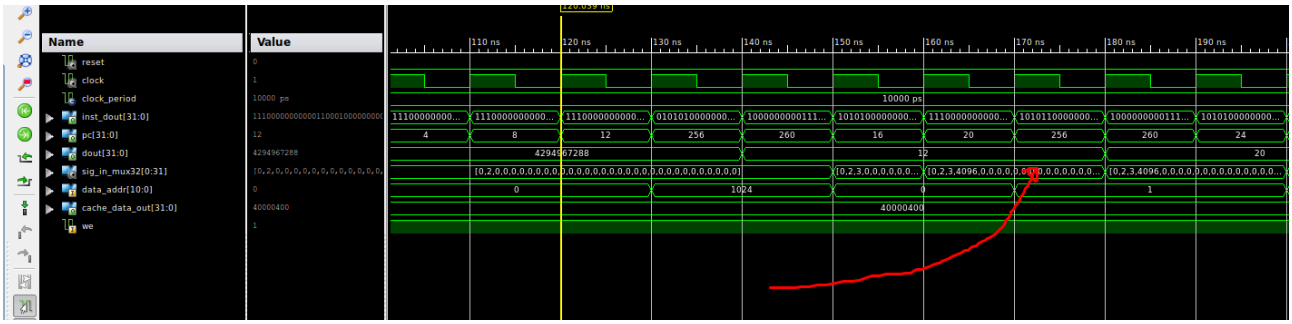


Κυμματομορφές

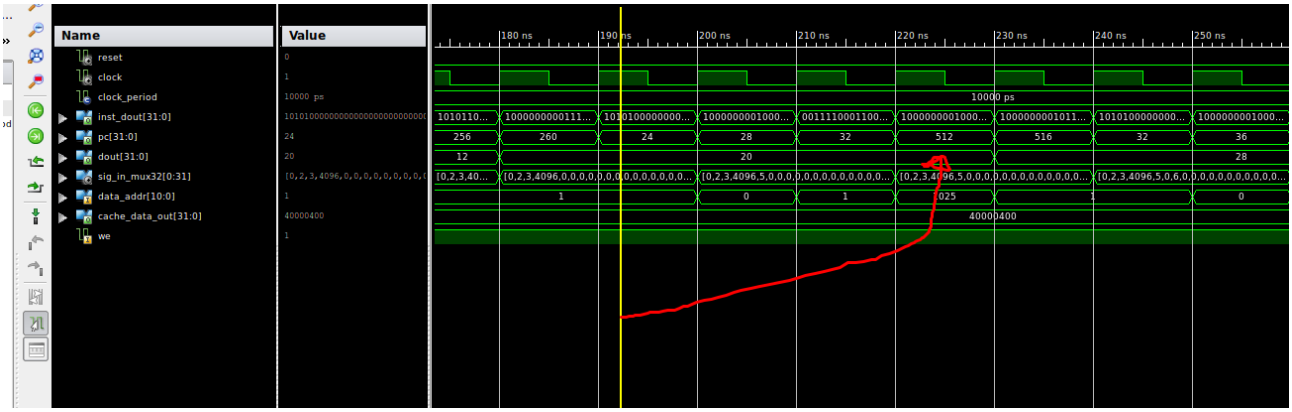
--WRONG INSTRUCTION (Exception $31 = 2)



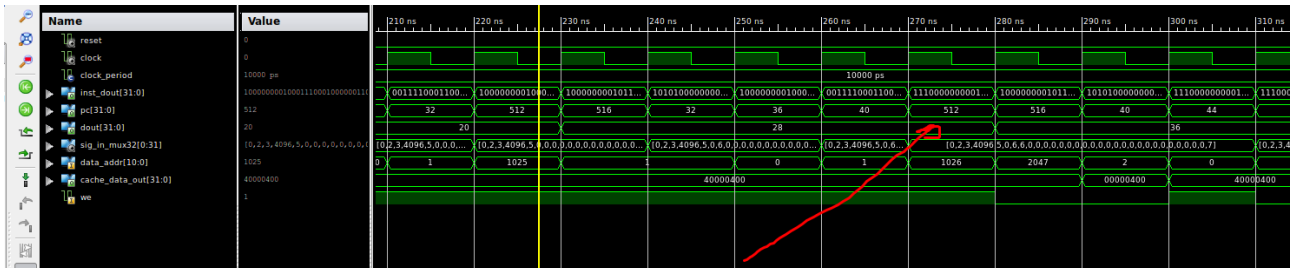
--WRONG INSTRUCTION (Exception $31 = 4)

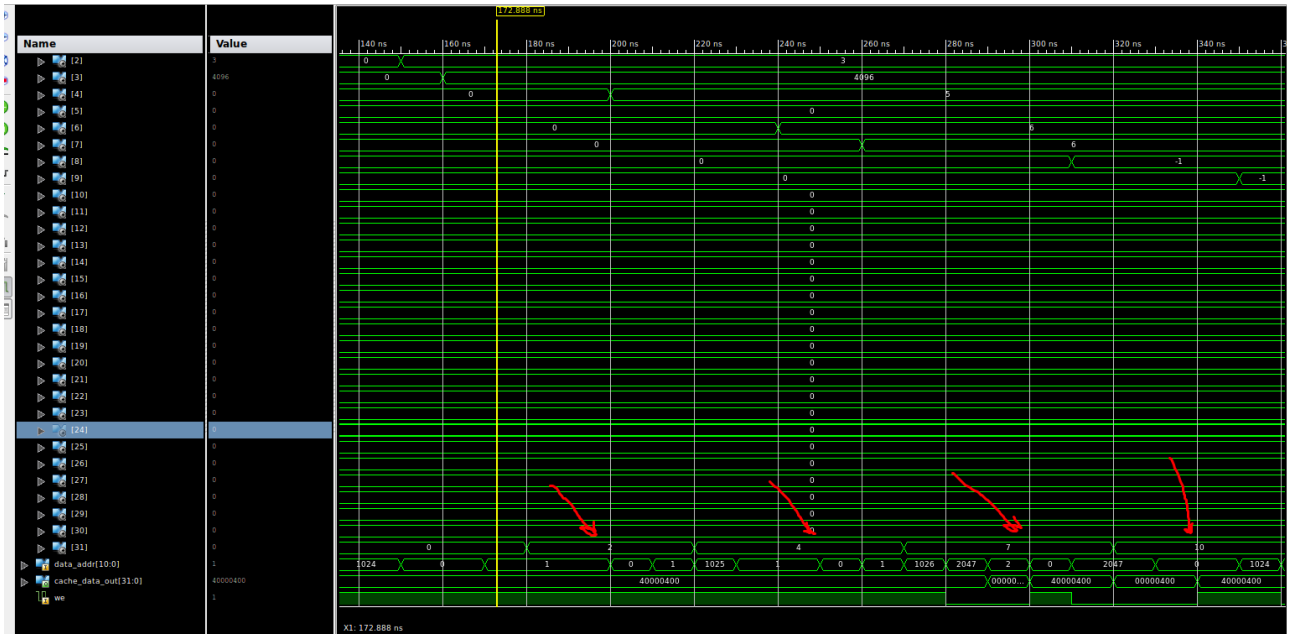


lw $4, 4($3) -- WRONG ADDRESS (4100) (Exception $31 = 7)

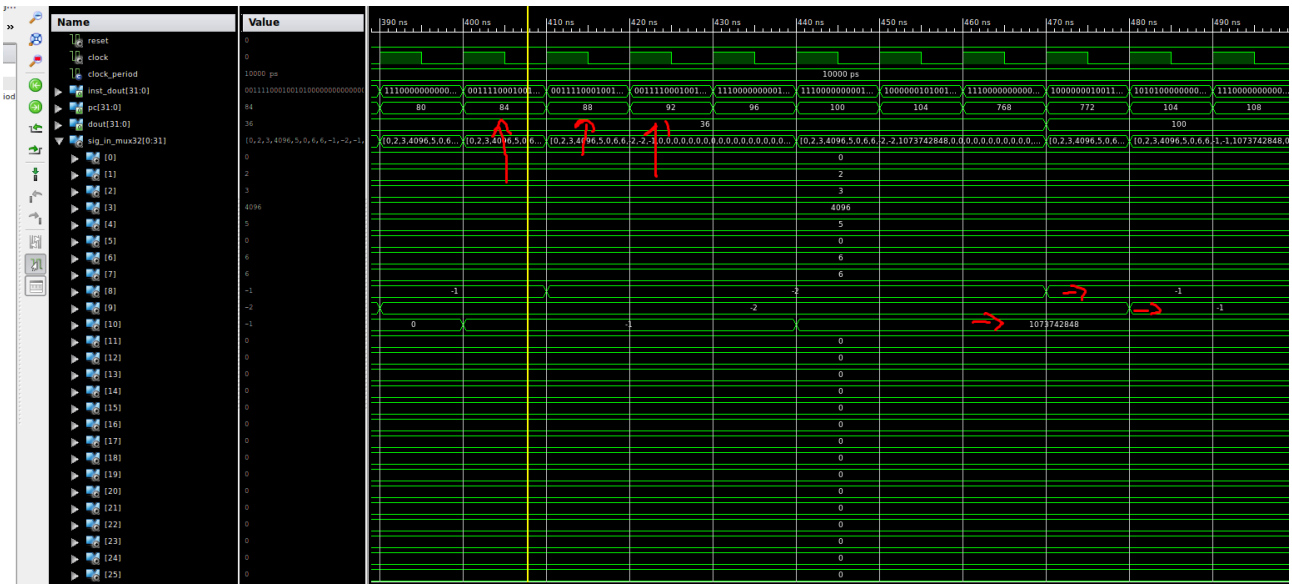


lw $6, 10($3) -- WRONG ADDRESS (4106) (Exception $31 = 10)





lw $10, 0($2) -- HIT ($10 = 0x40000400)
lw $11, 1($2) -- MISS (NOT THE SAME TAG) ($11 = No change)
lw $12, 5($2) -- MISS (NO VALID BIT) ($12 = No change)
li $8, xFFFF-- $8 = -1
li $9, xFFFF-- $9 = -1



add $8, $10, $10 -- (OVERFLOW $31 = 15)