

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 9

(1)

Βαθμίδες pipeline:

A) Instruction Fetch (IF) : Αρχικά στη βαθμίδα αυτή, έχουμε τον program counter (PC) που μέσω ενός αθροιστή δίνει συνεχώς τιμές που αναλογούν στον αριθμό της επομένης εντολής. Επομένη εντολή μπορεί να είναι η «αμέσως» επομένη (+4), ή αν μιλάμε για branch σε κάποια άλλη (πριν ή μετά).

B) Instruction Demory & register file read (ID) : Στη συνέχεια στη βαθμίδα αυτή διαβάζεται η εντολή και αποκωδικοποιείται σε : opcode , rs – rs –rd/Imm, ή/και funct. Καταλαβαίνουμε δηλαδή για ποια εντολή μιλάμε(add κτλ), με ποιες παραμετρους-καταχωρητες ή σταθερες τιμες και που πρέπει να αποθηκευτει αν χρειαζεται κατι ή αν πρέπει να προσπαλασει και να διαβασει ή να γραψει στην κυριως μνημη. Κανει επισης sign extension της 16bitης ποσοτητας σε 32 bit. Register File (RF) : Αργότερα οι καταχωρητες που χρησιμοποιουνται κατευθυνονται στο RF για να τους διαβασουμε από εκει ή να γραψουμε αργότερα σε αυτους

Γ) Execution (EX): Πιο μετά, αναλογα με τον opcode αρα και το ποια εντολή θελουμε να εκτελεσουμε η ALU είναι η υπευθυνη για να κανει πραξεις μεταξυ καταχωρητων ή καταχωρητων και σταθερων(Imm-sign extended)

Δ) Data Memory Access (MEM) : Η κυριως μνημη, από οπου ο επεξεργαστης αντλει και αποθηκευει τιμες και δεδομενα γενικοτερα

E) Write Back (WB) : Αναγνωση δεδομενων απο μνημη και εγγραφη στο αρχαιο των καταχωρητων

(2)

Απαραιτητες βαθμίδες:

A) IF : Κάθε εντολή χρειαζεται τον pc ώστε να ξερουμε ότι προκειται για αυτην, και πως προκυπτει (+4 ή branch)

B) ID : Για να αποκωδικοποιησουμε κάθε εντολή και να δουμε ποια είναι, χρειαζομαστε τον opcode και στις R-format εντολες το funct. Οποτε χρειαζεται η IM για να δουμε ποια εντολή θα εκτελεστει ενώ στη συνεχεια και με ποιες παραμετρους-καταχωρητες.

Οι υπολοιπες βαθμίδες δεν είναι παρουσες σε κάθε εκτελεση εντολης αναλογα με την εντολή, αν μιλάμε δηλαδή για load & store ή add,sub.. κτλ.

(3)

Σε συνεχεια λοιπον των προηγουμενων,

A) Οι εντολες διακλαδωσεις δεν χρησιμοποιουν τις βαθμίδες:

- EX
- MEM
- WB

Β) Για τις αριθμητικές πράξεις μέσω ALU δηλαδή δεν χρησιμοποιείται η βαθμίδα MEM αφού οι εντολές αυτές δεν διαβάζουν από την μνήμη αλλά από καταχωρητές

Γ) Οι εντολές store & load δεν χρειάζονται την βαθμίδα ALU καθώς δεν κάνουν κάποια πράξη με καταχωρητές ή αριθμούς γενικότερα

(4)

ADDI R2 R1 4

ADD R3 R1 R2

SUB R3 R2 R1

SW R3 16(R0)

LW R5 16(R1)

NOP

BREAK

NOP

(5)

Αν και οι εντολές μοιάζουν να είναι οι ίδιες, στο δεύτερο σετ εντολών, η ADD R4 R1 R6 χρησιμοποιεί τον R1 στον οποίο θα αποθηκευτεί το αποτέλεσμα της αφαίρεσης R2-R3 από το προηγούμενο SUB. Αυτό σημαίνει ότι R1 δεν θα είναι έτοιμο και αποθηκευμένο για να χρησιμοποιηθεί από επόμενη εντολή. Αρα θα χρειαστεί να περιμένει έως ότου αποθηκευτεί η τιμή στον R1. Αρα η καθυστέρηση θα οφείλεται στους παραπάνω κυκλούς ρολογιού που θα χρειαστεί μέχρι να αποθηκευτεί στον καταχωρητή η τιμή.

(6)

Αν στο datapath ήταν δυνατή η χρήση forwarding η τιμή με το που υπολογιζόταν θα πήγαινε κατευθείαν σε επόμενη εντολή, αν βεβαίως αυτή την ζητούσε. Αυτό σημαίνει ότι θα γίνονται οι εντολές λίγο πιο γρήγορα από τον συμβατικό τρόπο, αλλά όχι σε επίπεδο pipeline. Στο κυκλώμα δεν υπάρχει forwarding αφού και καθυστέρηση έχουμε και ξεχωριστό καταχωρητή ζητάει.