

ΑΛΕΞΑΝΔΡΟΣ ΜΑΡΑΝΤΟΣ

A.M. 3329

ΑΣΚΗΣΗ 8

(1)

Αρχικά, οι αρχικοποιήσεις θα μπορούσαν να παραληφθούν αλλά τότε οι καταχωρητές θα επαιρναν την τιμή 0 ως αρχική.

Επειτα, θα μπορούσαμε να τις αντικαταστήσουμε με κανονικές εντολές αρχικοποίησης (είτε add είτε addi)

π.χ. .register \$s4 15 ---> addi \$s4, \$0, 15
 .register \$s5 16 ---> addi \$s5, \$0, 16

(2)

Παραμένουν ίδια τα σήματα ελέγχου των Control, ALUcontrol για τις παρακάτω εντολές:

- RegDst
- Jump
- Branch
- MemtoReg
- MemWrite
- MemRead
- ALUSrc
- ALUop
- RegWrite
- OpCode

Αντιθέτα, αλλάζει το:

- FuncField

Αυτό ισχύει καθώς οι εντολές με R-Format σε αντίθεση με τις άλλες εντολές χρησιμοποιούν εκτός από το opcode και το funct, οπότε θα πρέπει το FuncField να αλλάζει

(3)	sw	lw
RegDst	0	0
Jump	0	0
Branch	0	0
MemtoReg	0	1
MemtoWrite	1	0
MemRead	0	1
ALUSrc	1	1
ALUOp1	0	0
ALUOp2	0	0
RegWrite	0	1
funct	000000	000000

Και για την sw και για την lw στην ALU εκτελείται η εντολή 010 (δηλαδή η add). Αυτό σημαίνει ότι στην ALU εκτελείται η ίδια εντολή, καθώς προσθεται το Imm (γι' αυτό add).

π.χ. sw \$s0, Imm(\$t0) = (\$t0 + Imm)

(4)	addi	andi	ori
RegDst	0	0	0
Jump	0	0	0
Branch	0	0	0
MemtoReg	0	0	0
MemWrite	0	0	0
MemRead	0	0	0
ALUSrc	1	1	1
ALUOp1	1	1	1
ALUOp2	1	1	1
RegWrite	1	1	1

- Συνεπώς και τα 3 σημάτια ελέγχου είναι ίδια για τις συγκεκριμένες εντολές (addi, andi, ori)
- Τα σημάτια ALUop1 & ALUop2 είναι ίδια για addi, andi & ori και add, and & or αντιστοίχα, αρα και οι πράξεις που εκτελούνται
- Η διαφορά στις εισόδους της ALU βρίσκεται στα πεδία funct, opcode & ALUop για τις εντολές addi, andi & ori / add, and & or αντιστοίχα
- Οι τελεστές διαβάζονται ως 16bitοι από το πεδίο Addr ενώ στη συνέχεια μετατρέπονται σε 32bitους μέσω sign extension (δηλαδή άλλες 16 φορές αριστερά το τελευταίο bit που διαβάστηκε) και ο πολυπλεκτής που το καθορίζει είναι ο mux4 που ελέγχεται από το ALUSrc
- Η διαφορά μεταξύ I-format και R-format εντολών είναι ότι στις I στέλνεται το Imm ώστε να γίνει το sign extension και μετά κατευθύνεται στον mux4 για να διαβαστεί. Αντιθέτως στις R-format εντολές ο mux4 βρίσκει και διαβάζει τα δεδομένα από το ReadData2 με τον ALUSrc αυτή τη φορά να είναι 0

(5)

Εστω λοιπόν ότι η beq είναι true, τότε ο PC θα πάρει την τιμή της εντολής label. Αν είναι false όμως ο PC θα πάρει την τιμή της εντολής που ακολουθεί την beq. Εστω π.χ. ότι έχουμε beq \$t0, \$t1, label, αν ισχύει ότι \$t0=\$t1 τότε ο PC θα πάει στη διεύθυνση του label.

Τέλος ο πολυπλεκτής που είναι υπεύθυνος γι' αυτό είναι ο mux5