

DATAPATH & CONTROL

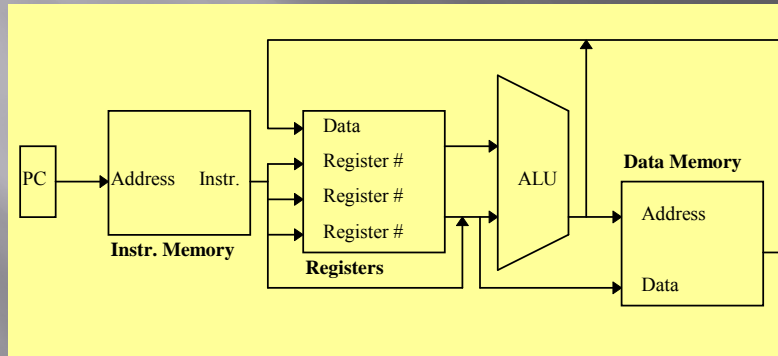
Αρχιτεκτονική Υπολογιστών

DATAPATH & CONTROL

- ▣ Για κάθε εντολή υπάρχουν δυο βήματα που πρέπει να γίνουν:
 - Προσκόμιση της εντολής (fetch) από τη θέση που δείχνει ο PC
 - Ανάγνωση των περιεχομένων ενός ή δύο καταχωρητών
- ▣ Τέτοιες λειτουργίες γίνονται διαμέσου του Datapath
- ▣ Ο έλεγχος του Datapath γίνεται διαμέσου της μονάδας ελέγχου (Control)

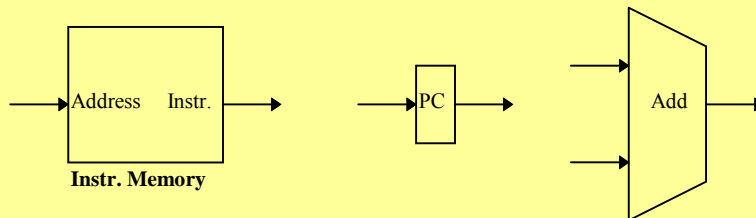
Αρχιτεκτονική Υπολογιστών

ΥΛΟΠΟΙΗΣΗ MIPS



Αρχιτεκτονική Υπολογιστών

ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΥΛΟΠΟΙΗΣΗΣ



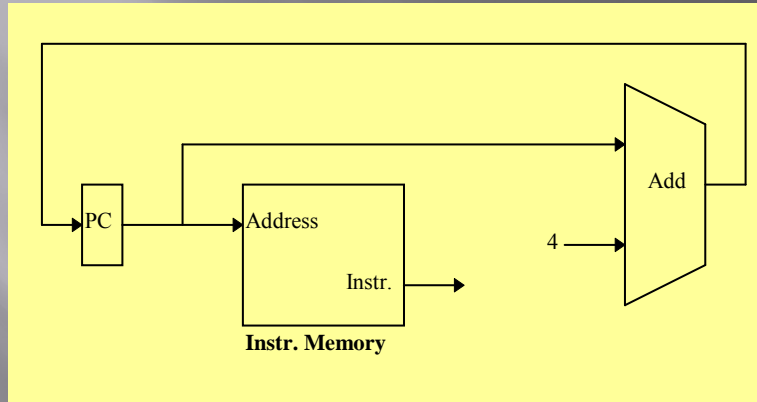
a. Instruction memory

b. Program counter

c. Adder

Αρχιτεκτονική Υπολογιστών

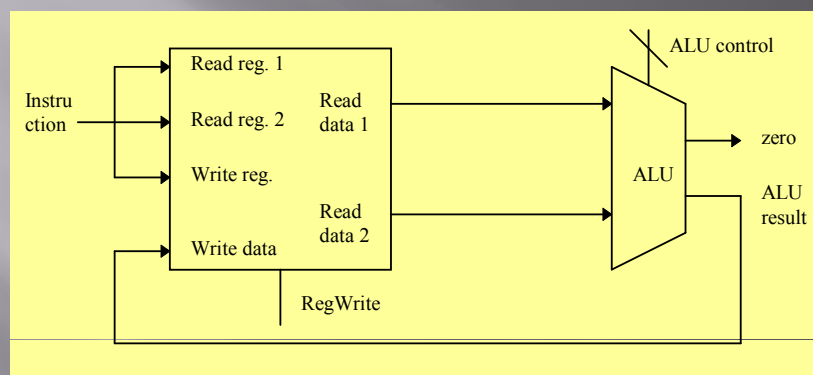
INSTRUCTION FETCH



Instruction Fetch

Αρχιτεκτονική Υπολογιστών

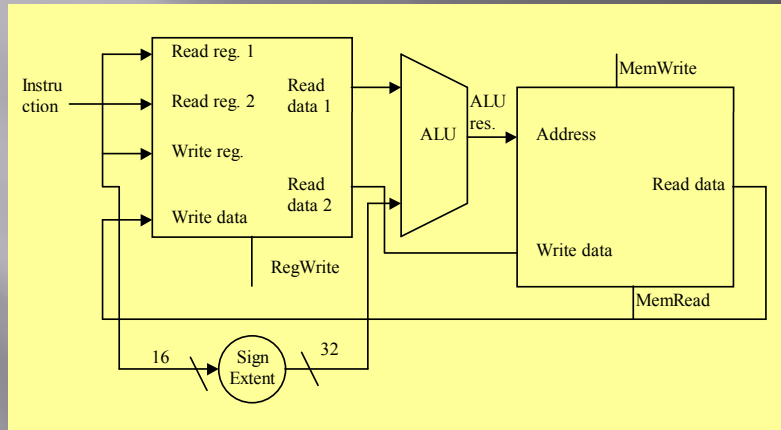
DATAPATH



R-Type Instructions

Αρχιτεκτονική Υπολογιστών

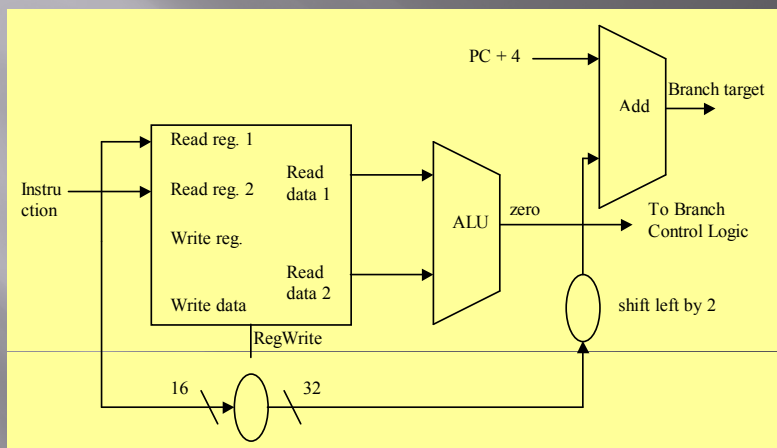
DATAPATH



Load/Store Instructions

Αρχιτεκτονική Υπολογιστών

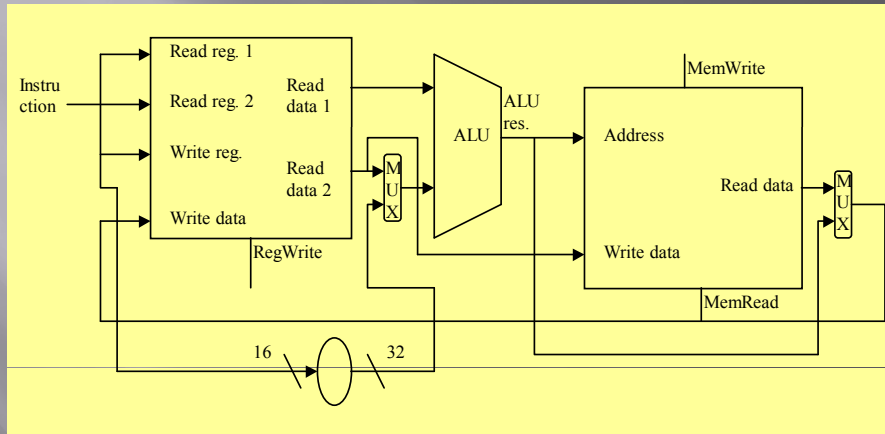
DATAPATH



Branch Instructions

Αρχιτεκτονική Υπολογιστών

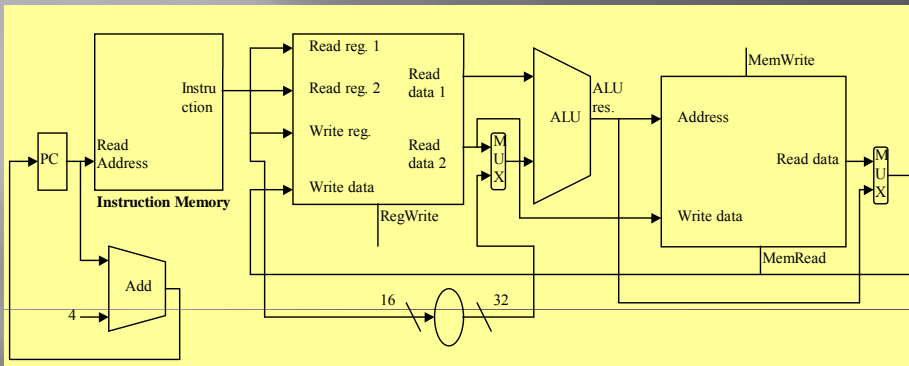
DATAPATH



R type & Load/Store Instructions

Αρχιτεκτονική Υπολογιστών

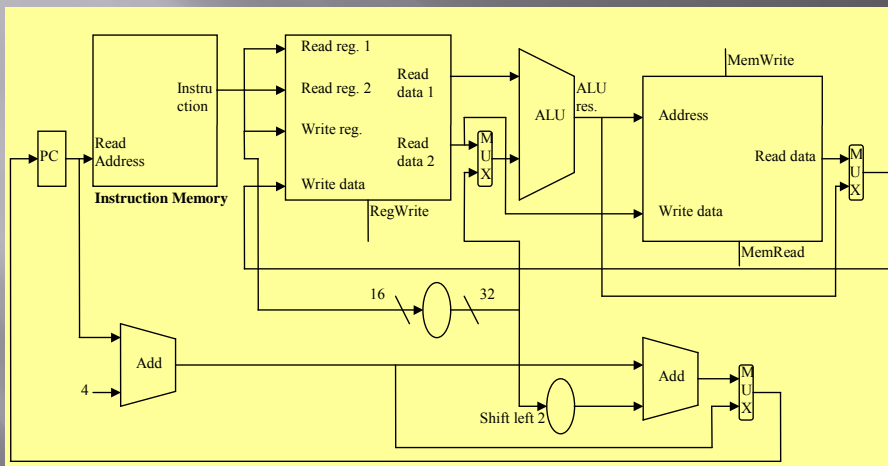
DATAPATH



R type, Load/Store and Instruction Fetch

Αρχιτεκτονική Υπολογιστών

DATA PATH



R type, Load/Store, Branch and Instruction Fetch

Αρχιτεκτονική Υπολογιστών

ΕΛΕΓΧΟΣ ALU

ALU control input	Function
000	And
001	Or
010	Add
110	Subtract
111	Set-on-less-than

Αρχιτεκτονική Υπολογιστών

ΔΗΜΙΟΥΡΓΙΑ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ALU

Instruction opcode	ALUOp	Instruction operation	Function code	Desired ALU action	ALU control Input
LW	00	load word	xxxxxx	add	010
SW	00	store word	xxxxxx	add	010
Branch equal	01	branch equal	xxxxxx	subtract	110
R-type	10	add	100000	add	010
R-type	10	subtract	100010	subtract	110
R-type	10	AND	100100	and	000
R-type	10	OR	100101	or	001
R-type	10	set-on-less-than	101010	set-on-less-than	111

Αρχιτεκτονική Υπολογιστών

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Operation
0	0	x	x	x	x	x	x	010
x	1	x	x	x	x	x	x	110
1	x	x	x	0	0	0	0	010
1	x	x	x	0	0	1	0	110
1	x	x	x	0	1	0	0	000
1	x	x	x	0	1	0	1	001
1	x	x	x	1	0	1	0	111

X: Don't Care

Αρχιτεκτονική Υπολογιστών

ΣΥΓΚΡΙΣΗ ΕΝΤΟΛΩΝ

R-type π.χ. add rd,rs,rt

0	rs	rt	rd	shumplt	funct
31-26	25-21	20-16	15-11	10-6	5-0

Load/Store π.χ. lw rt, immediate(rs)

35/43	rs	rt	immediate
31-26	25-21	20-16	15-0

Branch π.χ. beq rs,rt,immediate

4	rs	rt	immediate
31-26	25-21	20-16	15-0

Αρχιτεκτονική Υπολογιστών

ΣΗΜΑΤΑ ΕΛΕΓΧΟΥ

Signal name	Effect when deasserted	Effect when asserted
MemRead	None	Data memory contents at the read address are put on read data output.
MemWrite	None	Data memory contents at address given by write address is replaced by value on write data input.
ALUSrc	The second ALU operand comes from the second register file output.	The second ALU operand is the sign-extended lower 16-bits of the instruction.
RegDst	The register destination number for the Write register comes from the rt field.	The register destination number for the Write register comes from the rd field.
RegWrite	None	The register on the Write register input is written into with the value on the write data input.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC+4.	The PC is replaced by the output of the adder that computes the branch target.
MemtoReg	The value fed to the register write data input comes from the ALU.	The value fed to the register write data input comes from the data memory.

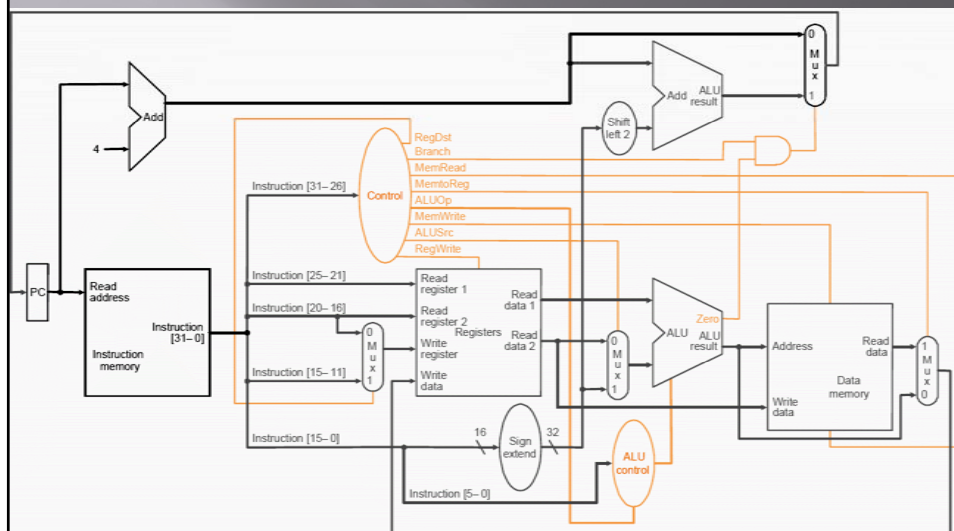
Αρχιτεκτονική Υπολογιστών

ΤΙΜΕΣ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ

Instruction	Reg Dst	ALU Src	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALU Op1	ALU Op0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	x	1	x	0	0	1	0	0	0
beq	x	0	x	0	0	0	1	0	1

Αρχιτεκτονική Υπολογιστών

DATAPATH & CONTROL



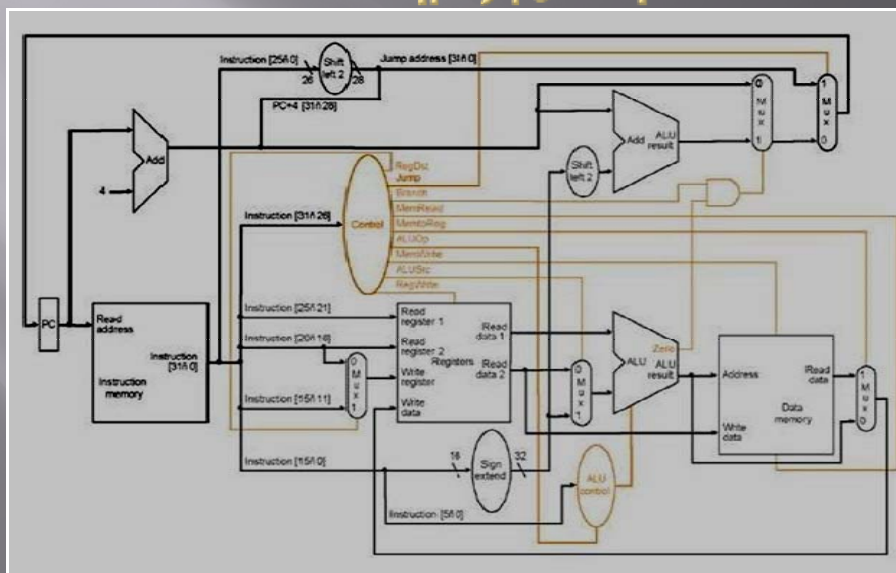
Αρχιτεκτονική Υπολογιστών

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΜΟΝΑΔΑΣ ΕΛΕΓΧΟΥ

		R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	x	x
	ALUSrc	0	1	1	0
	MemtoReg	0	1	x	x
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
ALUOp0	0	0	0	1	

Αρχιτεκτονική Υπολογιστών

DATAPATH & CONTROL Υποστήριξη Jump



CRITICAL PATH

Instruction Type	Functional units used by the Instruction type				
R-format	Instruction fetch	Register Access	ALU	Register access	
Load word	Instruction fetch	Register Access	ALU	Memory access	Register access
Store word	Instruction fetch	Register Access	ALU	Memory access	
Branch	Instruction fetch	Register Access	ALU		
Jump	Instruction fetch				

CPU execution time = Instruction count x CPI x Clock cycle time

Αρχιτεκτονική Υπολογιστών

EXECUTION TIME

Instruction type	Instruction memory	Register read	ALU operation	Data memory	Register write	Total	Frequency of Use
R-format	10	5	10	0	5	30ns	49%
Load word	10	5	10	10	5	40ns	22%
Store word	10	5	10	10		35ns	11%
Branch	10	5	10	0		25ns	16%
Jump	10					10ns	2%

Σταθερός κύκλος

CPU clock cycle=40ns

Μεταβλητός κύκλος

CPU clock cycle = $40 \times 22\% + 35 \times 11\% + 30 \times 49\% + 25 \times 16\% + 10 \times 2\% = 31.6$ ns.

Βελτίωση $40/31.6=1.27$

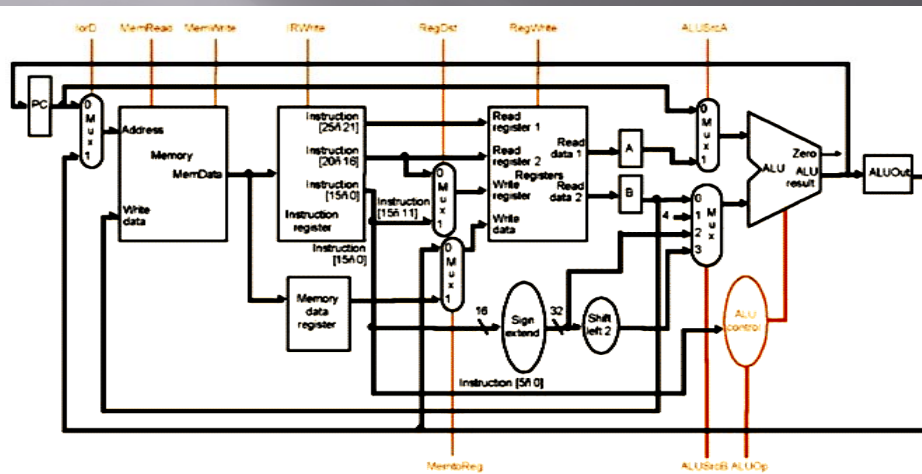
Αρχιτεκτονική Υπολογιστών

ΥΛΟΠΟΙΗΣΗ ΠΟΛΛΑΠΛΩΝ ΚΥΚΛΩΝ

- Κάθε εντολή εκτελείται σε περισσότερους από ένα κύκλους ρολογιού
- Γίνεται επαναχρησιμοποίηση μονάδων
- Απαιτούνται περισσότεροι πολυπλέκτες και ενδιάμεσοι καταχωρητές

Αρχιτεκτονική Υπολογιστών

ΥΛΟΠΟΙΗΣΗ ΠΟΛΛΑΠΛΩΝ ΚΥΚΛΩΝ



Αρχιτεκτονική Υπολογιστών

ΚΥΚΛΟΙ ΕΚΤΕΛΕΣΗΣ ΕΝΤΟΛΩΝ

- ΚΥΚΛΟΣ 1 (ίδιος για όλες τις εντολές)
 - $IR \leftarrow \text{Memory}[PC]$
 - $PC \leftarrow PC+4$
- ΚΥΚΛΟΣ 2 (ίδιος για όλες τις εντολές)
 - Ανάγνωση καταχωρητών: $\text{read RF}[rs], \text{RF}[rt]$
 - Υπολογισμός της διεύθυνσης Branch: $\text{Target} \leftarrow [PC+4]+4*\text{Imm}16$
 - Αποκωδικοποίηση του Opcode, 'αναγνώριση της εντολής'
- ΚΥΚΛΟΣ 3 (διαφέρει ανάλογα με την εντολή)
 - Αν η εντολή είναι ALU/LW/SW: πράξη ALU
 - Αν η εντολή είναι jump: $PC \leftarrow \text{Imm}26$
 - Αν η εντολή είναι Branch: ALU: συγκρίνει τα περιεχόμενα των δύο καταχωρητών αφαιρώντας τα. Αν το αποτέλεσμα είναι 0 (zero flag) τότε: $PC \leftarrow \text{Target}$

Αρχιτεκτονική Υπολογιστών

ΚΥΚΛΟΙ ΕΚΤΕΛΕΣΗΣ ΕΝΤΟΛΩΝ

- ΚΥΚΛΟΣ 4 (διαφέρει ανάλογα με την εντολή)
 - Αν η εντολή είναι ALU: Το αποτέλεσμα της ALU πηγαίνει σε κάποιο καταχωρητή:
 - $\text{RF}[rd/rt] \leftarrow \text{ALU}$
 - Αν η εντολή είναι η lw: Το αποτέλεσμα της ALU είναι μία θέση μνήμης, από την οποία πρέπει να διαβάσουμε:
 - Memory Read: $\text{read M}[\text{ALU}]$
 - Αν η εντολή είναι η sw: Το αποτέλεσμα της ALU είναι μία θέση μνήμης, στην οποία πρέπει να γράψουμε το περιεχόμενο κάποιου καταχωρητή:
 - Memory Write: $\text{M}[\text{ALU}] \leftarrow \text{RF}[rt]$
- ΚΥΚΛΟΣ 5 (μόνο για την lw)
 - Το περιεχόμενο της μνήμης που διαβάσαμε στον 4ο κύκλο οδηγείται σε κάποιο καταχωρητή:
 - Memory Read: $\text{RF}[rt] \leftarrow \text{M}[\text{ALU}]$

Αρχιτεκτονική Υπολογιστών

