

BİLGİSAYAR MİMARİSİ

Yılıçi Sınavı - ÖRGÜN Eğitim

- 1) Aşağıdaki tabloda bulunan pseudo-komutlar için gerçek MIPS makine dili komutları karşılıklarını bulunuz? [20p]

\$at register, geçici değer atamaları için kullanılabilir...

Pseudo-instruction	Solution
addi \$t5, \$t3, imm32	
beq \$t5, imm32, Label	
ble \$t5, \$t3, Label	
bgt \$t5, \$t3, Label	

- 2) Aşağıda verilen MIPS yazılımının yaptığı fonksiyonun C dilindeki ifadesini yazınız? [20p]

\$a0 : a dizisinin başlangıç adresini tutan register

\$to : i değişkenini tutan register

\$a1 : b dizisinin başlangıç adresini tutan register

\$So: c sabit değerini tutan register

```
addu $t0, $zero, $zero
addu $t1, $a0, $zero
addu $t2, $a1, $zero
addiu $t3, $zero, 101
loop: lw $t4, 0($t2)
      addu $t5, $t4, $s0
      sw $t5, 0($t1)
      addiu $t0, $t0, 1
      addiu $t1, $t1, 4
      addiu $t2, $t2, 4
      bne $t0, $t3, loop
```


- 4) Tek Çevrimli (single cyle) CPU ile Çoklu Çevrimli (multi cyle) CPU performansları karşılaştırılmak isteniyor. [20p]

Çeşitli işlem aşamalarında harcanan süreler aşağıda verilmektedir:

Instruction memory access time = 190 ps,	Data memory access time = 190 ps
Register file read access time = 150 ps,	Register file write access = 150 ps
ALU delay for basic instructions = 190 ps,	ALU delay for multiply or divide = 550 ps

Komut Seti içinde ALU komutları %30, Multiply & Divide komutları %15, load & store komutları %30, Branch %15, Jump %10 oranında bulunduğunu düşünün.

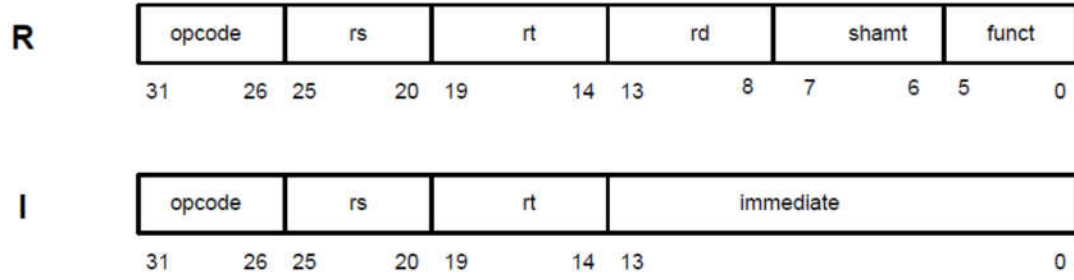
Not: Mux, Sign-Ext , Kontrol Unitesi gecikmeleri ihmal edilecektir.

- a) Tek Çevrimli CPU için clock cyle time değerini bulunuz?

- b) Çoklu Çevrimli CPU için clock cyle time değerini bulunuz?

- c) Performans iyileştirme oranını hesaplayınız (Speedup) ?

5) Aşağıda 64 Register içeren MIPS 32 işlemcinin değiştirilmiş komut formatları görülmektedir. [20p]



Buna göre aşağıdaki komutların bu formatlara göre tanımlı olup olmadıklarını tartışınız..
Tanımlı değilse nedenini belirtiniz...

Tanımlı olmayan komutların yerine bu formata göre tanımlı MIPS komutları kullanarak aynı fonksiyonu gerçekleştiriniz?

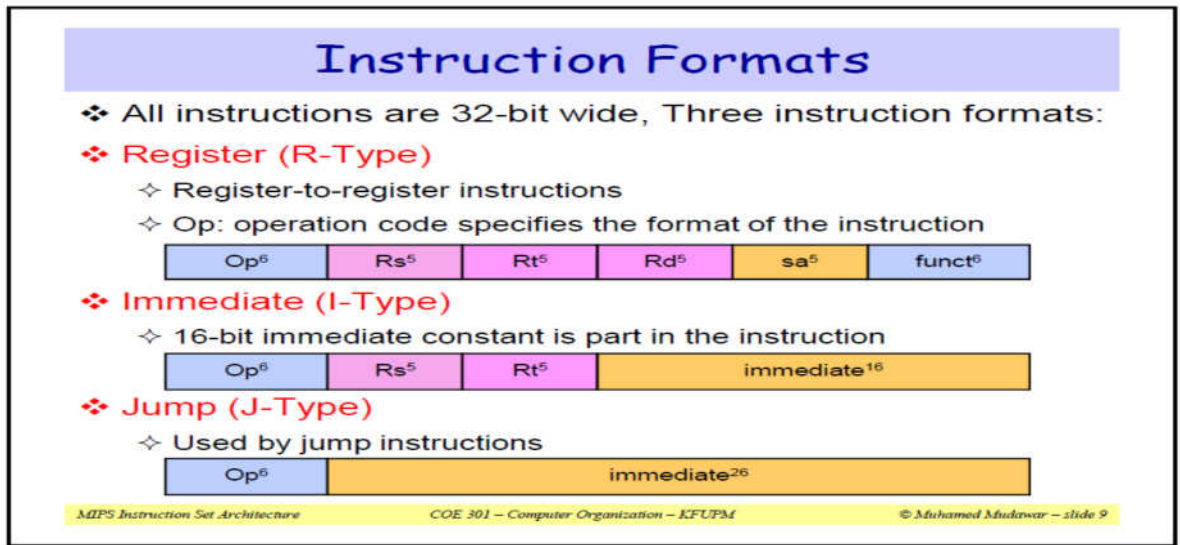
a)

`addi $56, $9, 4`

b)

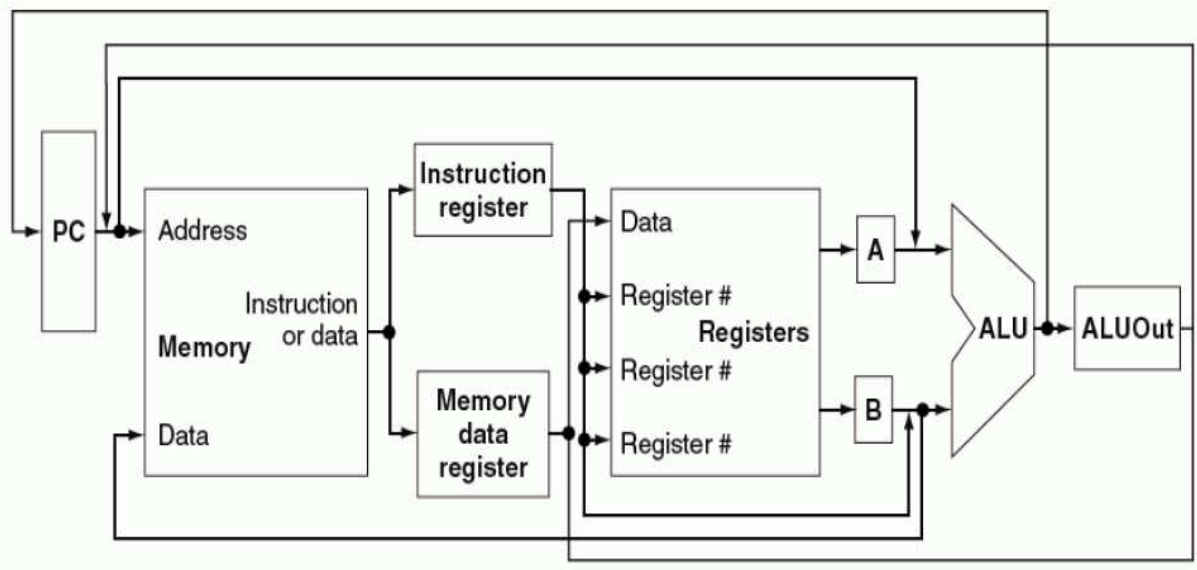
`sll $56, $9, 4`

EK MIP KOMUT FORMATLARI



Çok Çevrimli MIPS Komut İşleme Aşamaları

Step name	Action for R-type instructions	Action for memory-reference instructions	Action for branches	Action for jumps
Instruction fetch	IR ← Memory[PC] PC ← PC + 4			
Instruction decode/register fetch	A ← Reg [IR[25:21]] B ← Reg [IR[20:16]] ALUOut ← PC + (sign-extend (IR[15:0]) << 2)			
Execution, address computation, branch/jump completion	ALUOut ← A op B	ALUOut ← A + sign-extend (IR[15:0])	if (A == B) PC ← ALUOut	PC ← (PC [31:28], (IR[25:0]), 2'b00)
Memory access or R-type completion	Reg [IR[15:11]] ← ALUOut	Load: MDR ← Memory[ALUOut] or Store: Memory [ALUOut] ← B		
Memory read completion		Load: Reg[IR[20:16]] ← MDR		



Category	Instr	Op Code	Example	Meaning
Arithmetic (R & I format)	add	0 and 32	add \$s1, \$s2, \$s3	\$s1 = \$s2 + \$s3
	subtract	0 and 34	sub \$s1, \$s2, \$s3	\$s1 = \$s2 - \$s3
	add immediate	8	addi \$s1, \$s2, 6	\$s1 = \$s2 + 6
	or immediate	13	ori \$s1, \$s2, 6	\$s1 = \$s2 v 6
Data Transfer (I format)	load word	35	lw \$s1, 24(\$s2)	\$s1 = Memory(\$s2+24)
	store word	43	sw \$s1, 24(\$s2)	Memory(\$s2+24) = \$s1
	load byte	32	lb \$s1, 25(\$s2)	\$s1 = Memory(\$s2+25)
	store byte	40	sb \$s1, 25(\$s2)	Memory(\$s2+25) = \$s1
	load upper imm	15	lui \$s1, 6	\$s1 = 6 * 2 ¹⁶
Cond. Branch (I & R format)	br on equal	4	beq \$s1, \$s2, L	if (\$s1==\$s2) go to L
	br on not equal	5	bne \$s1, \$s2, L	if (\$s1!=\$s2) go to L
	set on less than	0 and 42	slt \$s1, \$s2, \$s3	if (\$s2<\$s3) \$s1=1 else \$s1=0
	set on less than immediate	10	slti \$s1, \$s2, 6	if (\$s2<6) \$s1=1 else \$s1=0
Uncond. Jump (J & R format)	jump	2	j 2500	go to 10000
	jump register	0 and 8	jr \$t1	go to \$t1
	jump and link	3	jal 2500	go to 10000; \$ra=PC+4

0 \$zero constant 0 (Hdware)	16 \$s0 callee saves
1 \$at reserved for assembler	... (caller can clobber)
2 \$v0 expression evaluation &	23 \$s7
3 \$v1 function results	24 \$t8 temporary (cont'd)
4 \$a0 arguments	25 \$t9
5 \$a1	26 \$k0 reserved for OS kernel
6 \$a2	27 \$k1
7 \$a3	28 \$gp pointer to global area
8 \$t0 temporary: caller saves	29 \$sp stack pointer
... (callee can clobber)	30 \$fp frame pointer
15 \$t7	31 \$ra return address (Hdware)

YANITLAR

ÖRÜN ÖĞRETİM

BİLGİSAYAR MİMARISI

ÇÖZÜMLER

①
a)

addi \$t5, \$t3, imm32

6p

lui \$at, upper 16
ori \$at, \$at, lower 16
add \$t5, \$t3, \$at

b)

beq \$t5, imm32, Label

lui \$at, upper 16
ori \$at, \$at, lower 16
beq \$t5, \$at, Label

6p

c)

ble \$t5, \$t3, Label

slt \$at, \$t3, \$t5
beq \$at, \$zero, Label

4p

d)

bgt \$t5, \$t3, Label

slt \$at, \$t5, \$t3
bne \$at, \$zero, Label

4p

2) for ($i=0, i \leq 100, i=i+1$)
 $\{a[i] = b[i] + c\}$
(20P)

5) a) addi \$56, \$9, 4

Tanımlı bir komuttur

(4P) \$56 ve \$9 \Rightarrow 6 bit ile kodlanabilir.
($56 < 64$)

(4P) imm. data 4 \Rightarrow 16 bit ile kodlanabilir.

b) sll \$56, \$9, 4

Tanımlı değildir. (Bu format için)

(4P) Shift amount = 2 bit olduğunda
4 değeri 2 bit ile kodlanamaz.

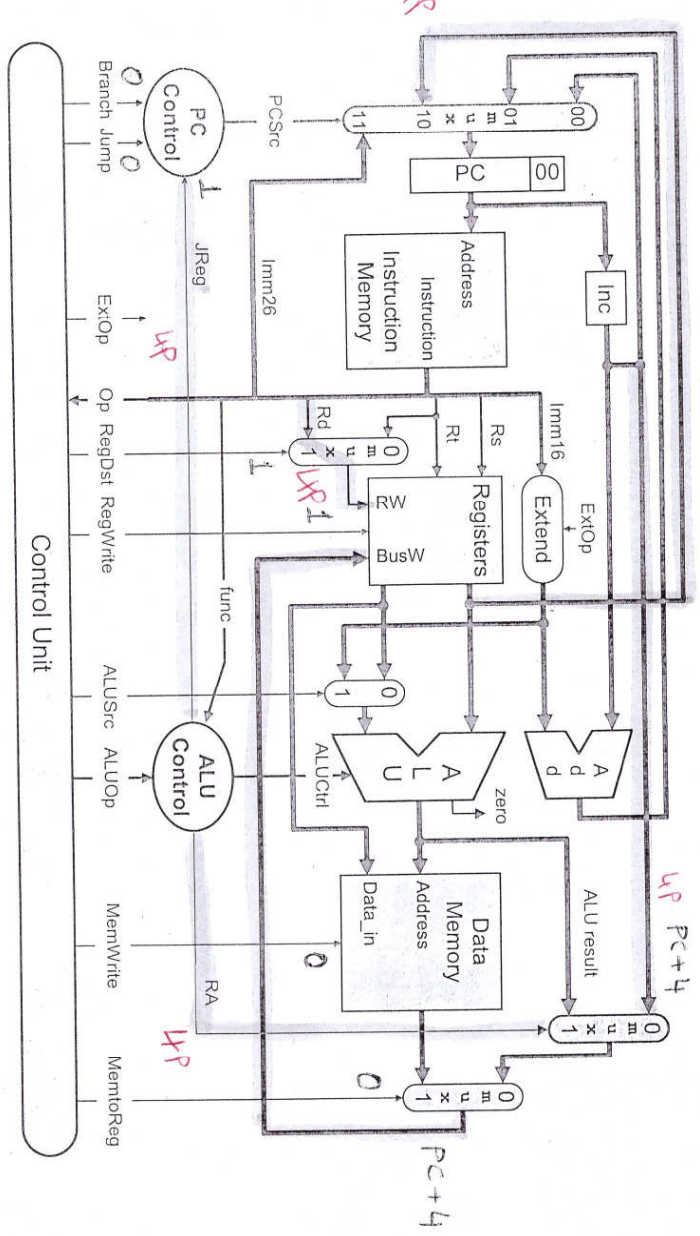
(8P) sll \$56, \$9, 3
sll \$56, \$56, 1

veya

sll \$56, \$9, 2

sll \$56, \$56, 2

3)



$PC \leftarrow \langle RS \rangle$ jump address
 $Rd \leftarrow PC + 4$ (next program)

4. (4 pts) We want to compare the performance of a **single-cycle CPU design** with a **multi-cycle CPU**. Suppose we add the multiply and divide instructions. The operation times are as follows:

Instruction memory access time = 190 ps, Data memory access time = 190 ps
 Register file read access time = 150 ps, Register file write access = 150 ps
 ALU delay for basic instructions = 190 ps, ALU delay for multiply or divide = 550 ps

Ignore the other delays in the multiplexers, control unit, sign-extension, etc.

Assume the following instruction mix: 30% ALU, 15% multiply & divide, 30% load & store, 15% branch, and 10% jump.

- a) What is the total delay for each instruction class and the clock cycle for the single-cycle CPU design.

Instruction Class	Instruction Memory	Register Read	ALU	Data Memory	Register Write	Total Delay
Basic ALU	190 ps	150 ps	190 ps		150 ps	680 ps
Mul & Div	190 ps	150 ps	550 ps		150 ps	1040 ps
Load	190 ps	150 ps	190 ps	190 ps	150 ps	870 ps
Store	190 ps	150 ps	190 ps	190 ps		720 ps
Branch	190 ps	150 ps	190 ps			530 ps
Jump	190 ps	150 ps				340 ps

Clock cycle = max delay = 1040 ps.

- b) Assume we fix the clock cycle to 200 ps for a multi-cycle CPU, what is the CPI for each instruction class and the speedup over a fixed-length clock cycle?

CPI for Basic ALU = 4 cycles

CPI for Multiply & Divide = 6 cycles (ALU takes 3 cycles)

CPI for Load = 5 cycles

CPI for Store = 4 cycles

CPI for Branch = 3 cycles

CPI for Jump = 2 cycles

I am going to assume that 30% for load and store is divided equally as 15% and 15%, because the problem does not separate their percentages.

Average CPI = $0.3 * 4 + 0.15 * 6 + 0.15 * 5 + 0.15 * 4 + 0.15 * 3 + 0.1 * 2 = 3.8$

- c) Speedup of multi-cycle over single-cycle = $(1040 * 1) / (200 * 3.8) = 1.37$

Handwritten calculations:

$$1040 / (200 * 3.8) = 1.37$$

(Note: The handwritten calculation shows a correction from 1.27 to 1.37, with a circled 4 and a circled 3.8.)