

1. MIPS procesor izvršava instrukciju **beq \$4, \$5, 1021** i sadržaji registara \$4 i \$5 su različiti. Šta će tačno biti upisano u PC i koji signali omogućuju ovaj upis?

*Odgovor:*

Stanje 0: **PCWrite=1, PCSource=00** →  $PC=PC+4$ .

Stanje 1: isto

Stanje 8: **PCWriteCond=1, Zero=0, PCWrite=0** → isto

Dakle, u registar PC biće upisano  $PC+4$ .

2. Čemu služi **Shift left 2** jedinica na posljednjem ulazu multipleksora (drugi ulaz ALU) i u toku izvršavanja kojih instrukcija ona dolazi do izražaja?

*Odgovor:*

**Shift left 2** (množenje sa 4) se koristi prilikom računanja adrese uslovnog skoka kod instrukcija uslovnog skoka.

3. U toku izvršavanja kojih instrukcija se vrši upis u **Target** registar? Šta se dešava sa njegovim sadržajem u toku izvršavanja instrukcije **jal 2000**?

*Odgovor:*

Upis u **Target** registar se vrši u toku izvršavanja svih instrukcija (stanje 1 je zajedničko). Sadržaj ovog registra u toku izvršavanja instrukcije **jal 2000** je  $PC+4+4*2000$ .

4. MIPS procesor redom izvršava instrukcije **slti \$8, \$0, 1** i **bne \$8, \$0, 111**. Šta će biti tačno upisano u PC registar nakon njihovog izvršenja? Pojasniti!

*Odgovor:*

**slti \$8,\$0,1**:  $PC=PC+4$ , u \$8 je 1

**bne \$8,\$0,111**:  $PC=PC+4$ , uslov je ispunjen →  $PC=PC+4*111$

Dakle, u registar PC biće upisano  $PC+8+4*111$ .

5. U toku je izvršavanje instrukcije **lw \$20, 20(\$20)**. Objasniti proceduru formiranja adrese sa koje se preuzima podatak za upis u odgovarajući registar.

*Odgovor:*

Računanje adrese sa koje se preuzima podatak za upis u odgovarajući registar obavlja se u stanju 2: **ALU result**= $IR[25-21]+signextended(IR[15-0])$ . Signal **ALU result** se vodi na ulaz 1 MUX-a na **Read address** ulazu **Memory** jedinice, kao adresa sa koje je potrebno pročitati podatak za upis u registar \$20.

6. MIPS procesor izvršava instrukciju **bne \$13, \$0, 130**, a prije nje je izvršio instrukciju **addi \$13, \$0, 0**. Šta će biti upisano u PC u posljednjem taktu izvršavanja pomenute bne instrukcije? Objasniti!

*Odgovor:*

Nakon izvršenja instrukcije **addi \$13, \$0, 0**, u \$13 je upisana nula i  $PC=PC+4$ . Obzirom da, u toku izvršenja instrukcije **bne \$13, \$0, 130**, neće doći do skoka (sadržaj \$0 i \$13 nije različit), sadržaj PC registra biće uvećan za 4.

Dakle, u registar PC biće upisano  $PC+8$ .

7. MIPS procesor izvršava instrukciju **beq \$3, \$7, 1021** i sadržaji registara \$3 i \$7 su isti. Šta će biti upisano u PC i koji signali omogućuju ovaj upis?

Odgovor:

Stanje 0:  $PC=PC+4$  (**PCWrite=1**, **PCSource=00**)

Stanje 1:  $PC=PC+4*1021$  (**PCWriteCond=1**, **Zero=1**, **PCSource=01**)

Dakle, u registar PC biće upisano  $PC+4+4*1021$ .

8. MIPS procesor redom izvršava instrukcije **slti \$8, \$0, 1021** i **bne \$8, \$0, 1021**. Šta će biti tačno upisano u PC registar nakon njihovog izvršenja? Pojasniti!

Odgovor:

**slti \$8, \$0, 1021**:  $PC=PC+4$ , u \$8 je 1

**bne \$8, \$0, 1021**:  $PC=PC+4$ , uslov je ispunjen  $\rightarrow PC=PC+4*1021$

Dakle, u registar PC biće upisano  $PC+8+4*1021$ .

9. Čemu služi **Zero** signal na izlazu ALU? Pojasniti njegovu ulogu prilikom izvršavanja instrukcije **lw \$3, 1021(\$10)**.

Odgovor:

**Zero** signal se dobija kao rezultat NOR operacije nad svim bitima rezultata ALU-e. Ovaj signal nema nikakvu ulogu prilikom izvršavanja instrukcije **lw \$3, 1021(\$10)**.

10. Od MIPS instrukcija koje su implementirane datom šemom, koja se instrukcija izvršava najkraće, a koja najduže, i koliko traje njihovo izvršavanje izraženo u broju clock-ova (ciklusa)?

Odgovor:

Najduže traje **load** - pet taktova, a po tri takta traju **jump** i **branch**.

11. Zbog čega je potrebno uvesti **Target** registar u multicycle implementacionu šemu?

Odgovor:

Za realizaciju pojedinih instrukcija, potrebno je ALU-u koristiti više puta (u više taktova). U stanju 1 (koje je zajedničko) ALU se koristi za izračunavanje adrese na koju se usmjerava tok programa za slučaj da je u pitanju instrukcija uslovnog skoka. Izračunata adresa se čuva u **Target** registru. Nakon toga, ALU se može koristiti za potpunu implementaciju date instrukcije.

12. Čemu služi multiplexer na **Write data** ulazu **Registers** jedinice i u toku izvršavanja kojih instrukcija on dolazi do izražaja?

Odgovor:

Multiplexer na **Write data** ulazu **Registers** jedinice služi za odlučivanje da li će se u određeni registar **Registers** jedinice upisati rezultat neke aritmetičko-logičke operacije doveden sa **ALU result** izlaza ALU-e (instrukcije R-tipa), ili, podatak pročitani iz memorije (instrukcija **lw**).

13. U toku je izvršavanje instrukcije **sw \$21, 120(\$19)**. U kom taktu izvršenja ove instrukcije se formira memorijska adresa u koju se upisuje sadržaj predmetnog registra? Objasniti proceduru formiranja ove adrese!

Odgovor:

Računanje adrese u koju se upisuje sadržaj registra obavlja se u stanju 2: **ALU result**= $IR[25-21]+signextended(IR[15-0])$ . Signal **ALU result** se vodi na **Write address** ulaz **Memory** jedinice, kao adresa na koju je potrebno upisati sadržaj registra \$21.

14. MIPS procesor izvršava instrukciju 1010 1110 1100 1100 0000 0100 0000 0010, koja se nalazi na adresi 100. Šta će tačno biti upisano u **Target** registar u toku izvršenja ove instrukcije i u kojem taktu? Objasniti! Koristi li se sadržaj ovog registra tokom izvršavanja ove instrukcije?

*Odgovor:*

**OP**=101011<sub>(2)</sub>=43<sub>(10)</sub>=0x2B → instrukcija **sw**

**address**=0000010000000010<sub>(2)</sub>=1026<sub>(10)</sub>

Stanje 0: **PC**=PC+4=100+4=104

Stanje 1: **Target**=PC+4\*1026=104+4\*1026

Sadržaj registra **Target** se ne koristi tokom izvršavanja instrukcije.

15. Čemu služi **Shift left 2** jedinica na posljednjem ulazu multipleksora koji vodi u **PC** i u toku izvršavanja kojih instrukcija ona dolazi do izražaja?

*Odgovor:*

**Shift left 2** služi za množenje sa 4, prilikom računanja adrese bezuslovnog skoka. Koristi se kod naredbi **J**-tipa.

16. U toku je izvršavanje instrukcije **beq \$21, \$19, 160**. Koliko iznosi adresa na koju se "skače" i od čega zavisi da li će doći do skoka?

*Odgovor:*

**Target**=PC+4+4\*160

Da li će doći do skoka zavisi od sadržaja registara \$21 i \$19, odnosno od vrijednosti **Zero** signala. Naime, ako su sadržaji registara \$21 i \$19 jednaki, na **ALU result** izlazu ALU-e će biti nula (stanje 8). **Zero** signal u tom slučaju ima vrijednost jedan. Obzirom da je **PCWriteCond**=1 i **PCSource**=01, u **PC** registar se upisuje vrijednost iz **Target** registra. Kao posljedica, tok izvršavanja programa usmjerava se na instrukciju na adresi PC+4+4\*160 (došlo je do "skoka"). Ukoliko su sadržaji registara \$21 i \$19 različiti, na **ALU result** izlazu ALU-e će biti vrijednost različita od nule. **Zero** signal u tom slučaju ima vrijednost nula. Dakle, u **PC** registru bi ostala vrijednost PC+4 (nije došlo do "skoka").

17. U toku izvršavanja kojih instrukcija se vrši upis u **Target** registar? Šta se dešava sa njegovim sadržajem u toku izvršavanja instrukcije **beq \$13, \$17, 221**?

*Odgovor:*

Upis u **Target** registar se vrši u toku izvršavanja svih instrukcija (stanje 1 je zajedničko).  
**beq \$13, \$17, 221: Target**=PC+4+4\*221