

ZADACI

RELATIVNA PERFORMANSA:

1. (15 bodova) Razmotrimo dvije različite implementacije iste instrukcijske arhitekture I_1 i I_2 , za koje su podatci prikazani tablicom:

	f	CPI_A	CPI_B	CPI_C	CPI_D
I_1	1,5 GHz	1	2	3	4
I_2	2 GHz	2	2	2	2

U tablici f označava radnu frekvenciju, a CPI_X prosječni broj perioda za izvršavanje instrukcija iz razreda X .

- (a) Neka je zadan program s 10^6 instrukcija distribuiranih po razredima kako slijedi: razred A: 10%; razred B: 20%; razred C: 50%; razred D: 20%.
Koja implementacija će brže izvesti zadani program? Odrediti ukupan broj taktova, kao i globalni CPI u oba slučaja.
- (b) Pretpostavimo da možemo odabrati jedno od sljedeća dva poboljšanja za I_1 : i) $CPI_D=2$, te ii) $CPI_C=2$. Koje bi od ta dva poboljšanja bilo opravdanije?
2. (10 bodova) Razmatramo poboljšanje računala na način da dodamo vektorsku procesnu jedinicu koja obavlja operacije nad vektorima od $n_v=20$ elemenata jednakom brzinom kao i konvencionalna procesna jedinica nad skalarima.

Definirajmo postotak vektorizacije p_v kao udio vremena proveden u operacijama koje bi potpuno iskoristavale resurse vektorske procesne jedinice.

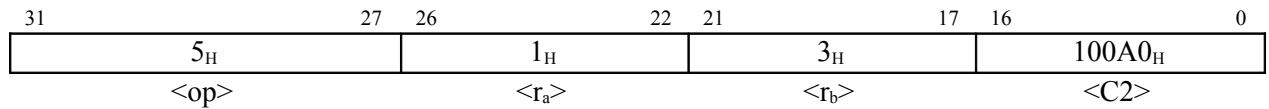
Koliki bi morao biti p_v ako želimo ostvariti:

- (a) dvostruko ubrzanje u odnosu na skalarnu izvedbu
(b) polovinu od maksimalnog ubrzanja u odnosu na skalarnu izvedbu
(c) dvostruko ubrzanje u odnosu na skalarnu izvedbu, ali uz $n_v=100$

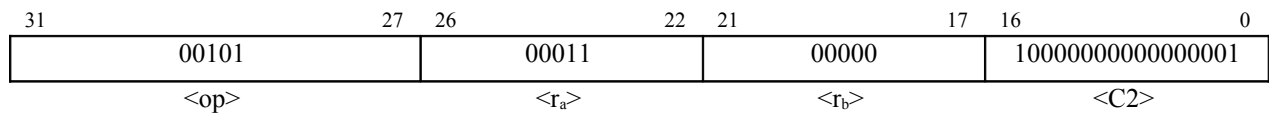
RISC ARHITEKTURA:

3. (10 bodova) Prikazati binarni kod instrukcije zbrajanja koja registar $r23$ umanjuje za 9.
Neka je poznato da operacijski kôd $\$de$ zauzima 8 najznačajnijih bitova instrukcije. Kôdovi izvornog i odredišnog registra zauzimaju po 6 bitova, dok je za usputnu konstantu predviđeno 12 najmanje značajnih bitova instrukcije.
4. (5 bodova) Predloži instrukcijski format tipa RISC kojim bi se instrukcije iz osnovna tri razreda kodirale pomoću samo 16 bitova.
Usporedi prednosti i nedostatke takve instrukcijske arhitekture u odnosu na arhitekturu SRISC koja je prikazana na predavanjima.

5. Instrukcija procesora SRISC **la** (load address) prikazana je na slici. Označene su vrijednosti i značenja pojedinih polja u instrukcijskoj riječi. Uz pretpostavku da je sadržaj registra R3 jednak 05000005_H, odredite posljedice izvođenja ove instrukcije. (4 boda)

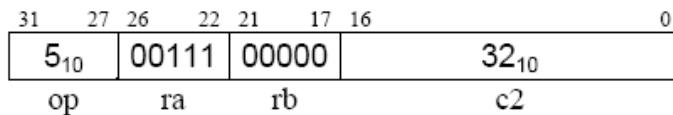


6. Instrukcija procesora SRISC **la** (load address) koja računa adresu operanda, ali umjesto dohvata operanda pohranjuje izračunatu vrijednost u R[r_a] ima za naš slučaj ovakav format:



Odredite sadržaje registara na koje ova instrukcija utječe. (4 boda)

7. Instrukcija **la r7,32** (la – load address) ima format prikazan na slici.

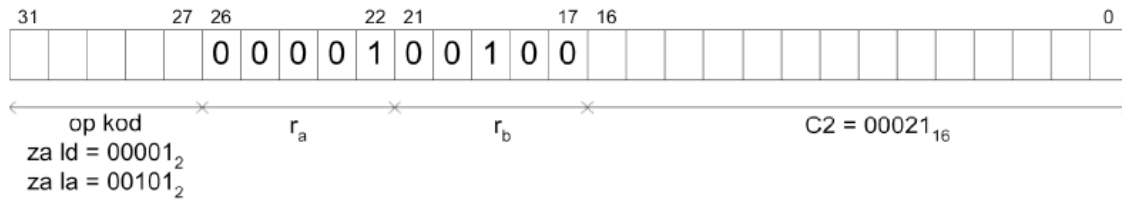


Odredite promijenjene sadržaje registara u programskom modelu SRISC-a. (4 boda)

Isto to napravite za instrukciju **la r7,32(r5)**.

Napomena: Početni sadržaj registra R[r5] neka je 510.

8. (3 boda) Za procesor SRISC i instrukcije **ld** ra, C2 (rb) i **la** ra, C2 (rb) čiji je format prikazan na slici:



pri čemu je operacijski kod za **ld** instrukciju 00001 a za **la** 00101, odredite:

a) efektivnu adresu (**izrazite je heksadekadno**) koju koristi instrukcija ld te označite registar ili registre opće namjene koji mijenjaju sadržaje tijekom izvođenja instrukcije;

b) promjene sadržaja registra (ili registara) opće namjene koje izaziva instrukcija la. Nove vrijednosti sadržaja registra **označite heksadekadno**. Sadržaj registara programskog modela SRISC je (neposredno prije izvođenja jedne ili druge instrukcije):

(r0) = 00 00 AA AA (heksadekadno)

(r1) = 00 00 00 01

(r2) = 00 00 00 02

(r3) = 00 00 00 03

(r4) = 00 00 00 04

(r5) = 00 00 00 06

(r6) = 00 00 00 07

(r7) = 00 00 00 08

(r8) = 00 00 00 09

(r9) = 00 00 00 0A

...

(r31) = 00 00 00 F0

PROTOČNA ARHITEKTURA:

9. Protočna upravljačka jedinica procesora neka se sastoji od 5 tipičnih protočnih segmenata. Navedite te tipične protočne segmente i ukratko opišite njihovu funkciju. Uz pretpostavku sljedećih vremena obrade u segmentima: $t_{S1}=15\text{ns}$, $t_{S2}=10\text{ns}$, $t_{S3}=17\text{ns}$, $t_{S4}=t_{S5}=15\text{ns}$, izračunajte efektivno vrijeme obrade za $N \gg M$. (3 boda)

10. Slika prikazuje Ganttov dijagram za protočni model procesora koji ima 5 protočnih segmenata:

Protočni segmenti							
WB				1	2	3	
ME			1	2	3	4	
EX			1	2	3	4	5
ID		1	2	3	4	5	6
IF	1	2	3	4	5	6	7
	ΔT	ΔT	ΔT	ΔT	ΔT	ΔT	ΔT vrijeme

Nacrtajte jednostavan model protočnosti, izračunajte vrijeme obrade i odnos između vremena potrebnog za efektivno izvođenje jedne instrukcije u neprotočnoj strukturi i u protočnoj strukturi. Neka je $t_1 = t_2 = t_3 = t_4 = t_5 = t_S = 10\text{ ns}$ za protočnu organizaciju, a vrijeme potrebno za izvođenje instrukcije u neprotočnoj izvedbi neka je $t_{NE} = t_1 + t_2 + t_3 + t_4 + t_5 = 5 t_S$. Pretpostavite da je broj instrukcija koje se izvode $N=10^7$. (3 boda)

11. Za model protočne strukture dubine 7 sa sljedećim karakteristikama:

$$tS1 = tS2 = tS3 = 40\text{ ns}$$
$$tS4 = tS5 = tS6 = tS7 = 45\text{ ns}$$

za $N=10000$ identičnih operacija odredite faktor ubrzanja obrade $SP=T/TP$, gdje je T vrijeme potrebno za obradu u neprotočnoj strukturi jednakih značajki, a TP vrijeme potrebno za obradu u protočnoj strukturi. (4 boda)

12. Napišite programski odsječak za SRISC procesor koji zamjenjuje CISC instrukciju $INC\ M$; gdje je M adresa memorijske lokacije na kojoj se nalazi operand. Uz pretpostavku da SRISC ima četiri instrukcijska protočna segmenta ocijenite broj perioda potreban za izvođenje programskog odsječka (POZOR! Uzmite u obzir mogućnost postojanja hazarda). (4 boda)

13. Protočna instrukcijska struktura procesora sastoji se od sljedećih pet protočnih segmenata:

1. IF – vrijeme dohvaćanja instrukcije 16 ns
2. ID – vrijeme dekodiranja 10 ns
3. EX – vrijeme izvršavanja aritmetičke ili logičke operacije / računanja efektivne adrese 16 ns
4. ME – vrijeme pristupa memoriji 18 ns
5. WB – vrijeme upisivanja rezultata 16 ns

Pretpostavite da se izvršava program od 12.000 instrukcija i da nema resursnih, upravljačkih i podatkovnih hazarda.

a) Izračunajte efektivno vrijeme potrebno za izvođenje jedne instrukcije

b) Ocijenite faktor ubrzanja u odnosu na neprotočnu jedinicu koja zahtijeva 76 ns za instrukcijski ciklus.

(4 boda)

14. Skicirati dio puta podataka protočne arhitekture MIPS koji je aktivan tijekom izvođenja instrukcija bezuvjetnog relativnog grananja, te obrazložiti takvu organizaciju. Naveći sve aktivnosti na putu podataka grupirane prema protočnim segmentima tijekom kojih se događaju. Bonus. U skicu ucrtati sklop u kojem se računa odredišna adresa grananja, te detaljno prikazati kako se on spaja s ostalim komponentama na putu podataka. (10 bodova)

15. (10 bodova) Zadan je 5-segmentni put podataka sa sljedećim trajanjem obrade po segmentima: IF: 150 ps; ID: 100 ps; EX: 125 ps; MEM: 150 ps; WB: 100 ps.

Odrediti latencije instrukcija ld, st, sub, te addi za slučaj neprotočne odnosno protočne izvedbe procesora.

16. (10 bodova) U zadanom programskom odsječku identificirati sve podatkovne hazarde (RAW, WAR, WAW).

```
lw r1, 40(r6)
add r6, r2, r2
sw r6, 50(r1)
lw r5, -16(r5)
sw r5, -16(r5)
add r5, r5, r5
```

17. (3 boda) Pretpostavite da je zbrajalo za brojeve s pomičnim zarezom ostvareno kao protočna jedinica sa sljedećim protočnim segmentima: *Oduzimanje eksponenta* (vrijeme obrade: 40 ns), *Poravnavanje mantise* (vrijeme obrade: 70 ns), *Zbrajanje* (vrijeme obrade: 50 ns) i *Normalizacija* (vrijeme obrade: 80 ns). Pretpostavite da takva protočna jedinica izvodi operaciju za $N = 10000$ parova operanada.

a) Ocijenite efektivno vrijeme obrade za jedan par operanada.

b) Ocijenite faktor ubrzanja $S_P = \frac{T}{T_P}$ gdje je T vrijeme obrade za neprotočnu strukturu (zbroj svih vremena sklopovlja za operaciju zbrajanja), a T_P vrijeme obrade za protočnu izvedbu jedinice. *Opaska:* U ovoj ocjeni pretpostavite da $N \rightarrow \infty$.

18. (4 boda) Protočna **sinkrona dinamička** instrukcijska struktura sastoji se od 8 protočnih segmenata. Vremena obrade u pojedinim segmentima su:

$$\begin{aligned}t_1 &= t_2 = t_3 = t_4 = 8 \text{ ns} \\t_5 &= 10 \text{ ns} \\t_6 &= t_7 = t_8 = 9,5 \text{ ns.}\end{aligned}$$

Izvodi se $N_1 = 10\,000$ instrukcija jednog tipa instrukcija. Nakon toga dinamička protočna struktura se rekonfigurira za potrebe izvođenja drugog tipa instrukcija kojih ima $N_2 = 20\,000$. Vrijeme potrebno za rekonfiguriranje protočne strukture je $T_r = 60 \text{ ns}$.

a) Odredite efektivno vrijeme izvođenja jedne instrukcije u gornjoj mješavini instrukcija $T_{\text{ef-prot}}$ (Opaska: Konačni rezultat možete ostaviti u obliku razlomka)

b) Odredite faktor ubrzanja $S = T_{\text{neprot}} / T_{\text{ef-prot}}$, gdje je T_{neprot} vrijeme obrade za neprotočnu strukturu nezavisno od tipa instrukcije i ono iznosi 64 ns .

(Opaska: Konačni rezultat možete ostaviti u obliku razlomka)

19. Pokaži kako bismo sljedeće instrukcije uklopili u temeljnu protočnu organizaciju arhitekture MIPS:

- `lwinc Rd, offset, Rs # Rd <- Mem[Rs+offset]; Rs=Rs+4`
- `addmem Rd, offset, Rs # Rd <- Rd + Mem[Rs+offset]`

Koje bi dodatno sklopovlje trebalo uvrstiti u put podataka za svaku od navedene dvije instrukcije, te kako bi se njihovo uvrštanje odrazilo na svojstva temeljne protočne strukture arhitekture?

20.

Zadana je sljedeća programska petlja u strojnom jeziku za računala arhitekture MIPS (`$s1`, `$s3` i `$t0` predstavljaju registre opće namjene):

```
Loop:
    lw    $t0, 0($s1)    # load word
    addi  $t0, $t0, 25    # add immediate
    sw    $t0, 0($s1)    # store word
    addi  $s1, $s1, 4     # add immediate
    bne   $s1, $s3, Loop # branch if not equal
```

- Skicirati kod u C-u koji bi na klasičnom SISD procesoru mogao biti preveden u zadani strojni kod.
- Za petlju iz prethodnog pitanja prikazati kakav bi kod bio generiran za procesor s dvostrukim statičkim izdavanjem koji istovremeno može izvoditi:
 - jednu instrukciju tipa ALU/branch,
 - jednu instrukciju tipa load/store.

Pretpostaviti da se učitani registar ne može koristiti u instrukciji neposredno nakon instrukcije čitanja memorije. Odrediti CPI koji se postiže u tijelu petlje ako zane-marimo instrukcije `nop`.

- Pokušaj ponuditi raspored u kojem jedan prolaz kroz petlju troši samo četiri ciklusa.
- Pokušaj ponuditi raspored u kojem četiri prolaza kroz petlju troši osam ciklusa (BONUS).