7.		1
8.	1 - St. J	
9.		
10.		

17.	
18.	
19.	
20.	

UKUPNO

38

Potpis nastavnika:





Roglić Danijel

## Pismeni ispit iz Arhitekture računala 2

	A	В	C	D			A	В	С	D
B 1		0	0	0	0	9.+	0	0	•	0
15 2	•	0	0	0	15	10	0	0	0	•
15 3.t	0	•	0	0	C	11.4	0	0	•	0
13 4.+.	0	•	0	0	3	12.+	0	•	0	0
C 5.+	0	0	•	0	h	13	•	0	0	0
C 6.+	0	0	•	0	6	14.4	0	0	•	0
₾ 7.—	0	0	0	•	A	15.—	0	0	•	0
0 8.+	0	0	0	•	3	16	•	0	0	0

- 1. Ako zanemarimo promašaje stranice, vrijeme pristupa memorijskom podatku na modernom računalu je:
  - (a) uvijek jednako
  - (b) varira puno zbog mogućih promašaja u PM i TLB
  - (c) varira malo (+-50%)
  - (d) varira puno zbog mogućih promašaja u PM
- 2. Broj instrukcija arhitekture x86 je:
  - (a) do 2000. je rastao ali je sada konstantan
  - (b) u stalnom porastu
  - (c) do 2004. je rastao ali je sada konstantan
  - (d) do 2004. je rastao ali sada opada

- 3. Koja od sljedećih tehnika ne vodi povećanju propusnosti modula DRAM?
  - (a) protočnost
  - (b) asinkroni upravljački protokol
  - (c) širenje podatkovne sabirnice
  - (d) brzi pristup retku uz grupni prijenos podataka
- 4. Koji glavni učinak na izvođenje programa možemo očekivati nakon vektoriziranja algoritma?
  - (a) ne možemo očekivati nikakav učinak
  - (b) smanjivanje broja izvedenih instrukcija
  - (c) povećavanje broja izvedenih instrukcija
  - (d) povećavanje memorijskog prometa

3+/

-7.

## Ig = C5. (\$\phi\_{18} + \$\phi\_{10}) \gamma MDR - A: Resultat &itamin

- 5. Za vektorske instrukcije okupljanja vrijedi da:
  - (a) spremaju vektorski operand u memoriju
  - (b) spremaju više skalara na uzastopne memorijske lokacije
  - (c) učitavaju vektorski operand iz memorijskih lokacija koje ne moraju biti uzastopne
  - (d) iz memorije učitavaju jedan skalarni operand
- Moderni superskalarni procesori tipično postižu:
  - (a) CPI > 100
  - (b) CPI ∈ [2, 10]
  - (c) CPI < 1
  - (d) CPI > 3 GHz
- 7. Koliko bitova nam je potrebno za kodiranje memorijskog operanda instrukcija pojednostavljenog modela procesora?
  - (a) 16 ili 32
  - (b) 4
  - (c) 8 ili 16
  - (d) 8
- Performansa računala u kontekstu izvođenja slijednih programa u posljednjih nekoliko godina:
  - (a) prati Mooreov zakon
  - (b) opada
  - (c) raste uz ubrzanje porasta
  - (d) raste uz usporenje porasta
- Podatak kojeg je osaminstrukcijski procesor pročitao iz memorije može u svom cjelovitom obliku doći i do:
  - (a) registra PC
  - (b) registra MAR
  - (c) registra AC
  - (d) niti do jednog registra
- 10. Koja komponenta modernih računala nije bila prisutna u originalnoj Von Neumannovoj arhitekturi?
  - (a) ožičena upravljačka jedinica
  - (b) skup registara
  - (c) radna memorija
  - (d) interna sabirnica

- 11. Kakvi podatci se tipično stavljaju na upravljački stog računala?
  - (a) operacijski kodovi instrukcija
  - (b) ovisi o tome postoji li numerički koproce-
  - (c) parametri potprograma i lokalne varijable
  - (d) s pomičnim zarezom, ali ne i cjelobrojni
- 12. Memorijski adresni registar je:
  - (a) ponor podatka na podatkovnoj sabirnici
  - (b) izvor podatka na adresnoj sabirnici
  - (c) spojen i na adresnu i na podatkovnu sabirnicu
  - (d) ponor podatka na adresnoj sabirnici
- Širina mikroprogramskog adresnog registra H određena je:
  - (a) kapacitetom sekundarne memorije
  - (b) kapacitetom mikroprogramske memorije
  - (c) kapacitetom radne memorije
  - (d) kapacitetom priručne programske memorije
- Za arhitekture tipa VLIW je specifično da se usporedno izvođenje slijednog programa pospješuje prvenstveno:
  - (a) adresnim preslikavanjem
  - (b) dinamičkim raspoređivanjem instrukcija u sklopovlju procesora
  - (c) statičkim raspoređivanjem instrukcija tijekom prevođenja
  - (d) dubokom protočnom strukturom
- 15. Kako na tipičnoj arhitekturi tipa RISC postižemo izravno adresiranje memorijskog podatka na proizvoljnoj adresi?
  - (a) u dva koraka: i) dohvat adrese u registar,ii) registarsko indirektno adresiranje podatka
  - (b) instrukcijom st
  - (c) instrukcijom ld
  - (d) u dva koraka: i) transfer podatka na stog, ii) adresiranje podatka
- Kod straničenja, adresno preslikavanje se izvodi pomoću sljedeće operacije:
  - (a) zbrajanje
  - (b) prozivanje tablice
  - (c) dijeljenje
  - (d) ostatak pri cjelobrojnom dijeljenju

UKUPNO 38

Potpis nastavnika:

8. rujna 2015.

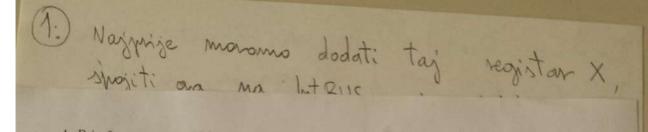
Zavod za elektroniku, mikrelektroniku, računalne i inteligentne sustave

## Arhitektura računala 2 pismeni ispit

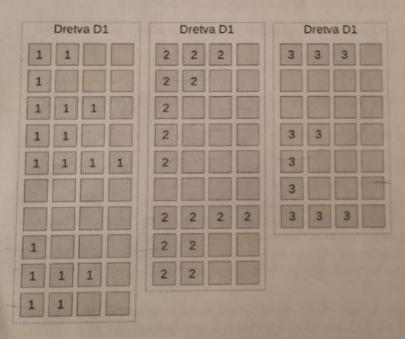
- 1. Model 8-instrukcijskog procesora potrebno je modificirati tako da podržava indeksirano adresiranje instrukcijom LDAIND. Instrukcija LDAIND ADDR učitava podatak koji se nalazi na adresi X + ADDR, gdje je X sadržaj indeksnog registra X. Detaljno prikažite što bi sve trebalo promijeniti na putu podataka i u upravljačkoj jedinici kako bi procesor implementirao traženu instrukciju. Predložite skup dodatnih instrukcija koji bi omogućio korištenje instrukcije LDAIND u praksi.
- 2. Napišite program za pojednostavnjeni model 8-bitnog procesora koji određuje apsolutnu vrijednost akumulatora. Pretpostavite da su na raspolaganju instrukcije:
  - BMI offset: relativno grananje ako je rezultat posljednje operacije ALU negativan.
  - XOR const: A <- A xor const
  - ADD const: A <- A + const
- 3. Razmotrite sljedeći odsječak u programskom jeziku C gdje je a znakovni niz (char \*).

```
int count=0;
for (int i=0; i!=n; ++i){
  count += a[i];
}
```

- (a) Prevedite prikazani odsječak u strojni kod za arhitekturu MIPS pod pretpostavkom savršenog predviđanja grananja (instrukcije grananja ne trebaju priključak za kašnjenje) te pod pretpostavkom da se zakašnjelo učitavanje ne koristi. Varijable count, i, n, a i query predstavite registrima t1-t5, a uvjet izlaska iz petlje izrazite usporedbom indeksa. Koristite instrukcije 1b (load byte), add, beq (branch if equal) i bne (branch if not equal).
- (b) Pokažite podatkovne hazarde u vašem strojnom kodu i naznačite njihovu vrstu. Navedite koji bi se od tih hazarda mogli riješiti i) prosljeđivanjem, ii) promjenom redoslijeda instrukcija ili iii) protočnim mjehurićem (odnosno instrukcijom nop).
- (c) Pretpostavite da se vaš strojni kôd izvodi na procesoru s dvostrukim statičkim izdavanjem koji ima savršeno predviđanje grananja, jednu protočnu strukturu za instrukcije ALU/branch i jednu protočnu strukturu za instrukcije load/store. Prikažite redoslijed izdavanja instrukcija tijekom jednog prolaska kroz petlju (dakle, zanemarivši inicijalizaciju na početku petlje), kao i instrukcije nop koje procesor treba dodati zbog podatkovnih hazarda.



- 4. Priručna memorija s 32-bitnim adresama koristi izravno preslikavanje i ima sljedeću strukturu adrese:
  - · bitovi 31-14: oznaka
  - · bitovi 13-4: indeks
  - · bitovi 3-0: pomak
  - (a) Odredite veličinu bloka, kapacitet priručne memorije, te ukupni broj bitova potrebnog za realizaciju takve priručne memorije.
  - (b) Neposredno nakon uključivanja računala zabilježen je sljedeći niz pristupa adresama: 0, 4, 16, 132, 232, 160, 1024, 30, 140, 3100, 180, 2180. Odredite koliko blokova će biti zamijenjeno i koji je omjer pogotka.
- 5. Za tri dretve (vidi sliku) prikažite izvođenje
  - (a) u superskalarnom simultano višedretvenom procesoru SMT (izdaje do četiri instrukcije; protočne strukture nisu specijalizirane);
  - (b) u višedretvenom superskalarnom procesoru (izdaje do četitri instrukcije) koji koristi finozrnatu dretvenost.



Slika uz zadatak 5.