

## Arhitektura računala 2

Pismeni ispit, problemski dio (60% bodova)

1. (12 bodova) Na sabirnicu računala temeljenog na pojednostavljenom modelu CISC procesora spojen je ulazno-izlazni uređaj s dva interna registra (podatkovnim i statusnim), tako da se registri nalazi na dvjema najvećim mogućim adresama (adrese su 16-bitne).

- (a) Nacrtajte shemu spajanja ulazno-izlaznog uređaja na sabirnicu računala, uz pretpostavku da uređaj ima jedan ulaz za omogućavanje  $E^*$  (radi u negativnoj logici), ulaz RS (register select;  $RS=0$  odabire statusni, a  $RS=1$  podatkovni registar), ulaz  $R/W^*$ , te 8 podatkovnih priključaka D0-D7.
- (b) Napišite programski odsječak koji će provjeriti je li sadržaj statusnog registra ulazno-izlaznog uređaja različit od nule, te ako jest, pročitati podatak iz podatkovnog registra i pohraniti ga u memoriju na adresu \$7FFF. Programski odsječak neka započinje na adresi \$3000.

Na raspolaganju su sljedeće instrukcije, s pripadnim operacijskim kodovima i opisima (x predstavlja jednu heksadekadsku znamenku):

LDA A, \$xxxx (op. kod \$B6) - čita memorijski operand s apsolutne adrese \$xxxx u akumulator A.

STA A, \$xxxx (op. kod \$B7) - upisuje podatak iz akumulatora A u memoriju na apsolutnu adresu \$xxxx.

TST A (op. kod \$4D) - postavlja zastavice Z (zero) i N (negative) ovisno o trenutnom sadržaju akumulatora A.

BEQ \$xx (op. kod \$27) - uvjetno grananje ako je zastavica Z=1 na relativno zadanu adresu kodiranu jednim bajtom (ako je Z=1:  $PC \leftarrow PC + \$xx$ ).

BNE \$xx (op. kod \$26) - uvjetno grananje ako je zastavica Z=0 na relativno zadanu adresu kodiranu jednim bajtom (ako je Z=0:  $PC \leftarrow PC + \$xx$ ).

- (c) Nacrtajte stanje na vanjskim sabirnicama procesora tijekom izvođenja tog programskog odsječka.
2. (12 bodova) Zadan je sljedeći potprogram u programskom jeziku C:

```
void sxpy(int *x, int* x_end, int *y) {
    while (x!=x_end) {
        (*x) += (*y);
        ++y; ++x;
    }
}
```

- (a) Izraziti funkcionalnost potprograma prirodnim jezikom.
- (b) Izraziti tijelo funkcije ekvivalentnim neoptimiranim strojnim kodom za arhitekturu MIPS pod pretpostavkom da registri \$s1, \$s2 i 2\$s3 odgovaraju varijablama x, y, i x\_end.
- (c) Predložiti ekvivalentni optimirani strojni kod za računalo s jednostrukim izdavanjem (Uputa: pokušajte popuniti priključke za kašnjenje te pripazite da u prolazu kroz petlju imate samo jedno grananje).

BONUS:

- (d) Predložiti ekvivalentni optimirani kod za MIPS sa statičkim dvostrukim izdavanjem koji ima jednu protočnu strukturu ALU/branch i jednu protočnu strukturu load/store, pod pretpostavkom da je računalo opremljeno sklopovljem za predviđanje grananja te razrješavanje hazarda nakon instrukcija učitavanja automatskim umetanjem instrukcija nop.
- (e) Procijeniti broj taktova potrebnih za izvođenje jednog prolaza kroz petlju na računalu pod (d) korištenjem programa pod (b), (c) i (d).



3. (12 bodova) Razmatramo dvostruko asocijativnu priručnu memoriju kapaciteta 512B s linijama od 16 bajta te algoritmom zamjene LRU. Zadan je sljedeći redoslijed 32-bitnih memorijskih referenci bajtnih podataka: 0x0001 0x0086 0x00d4 0x0001 0x0187 0x00d5 0x01d2 0x0281 0x0002 0x008c 0x0189 0x00dd
- (a) Odrediti prosječno vrijeme pristupa memoriji ako je poznato da je priručna memorija na početku izvođenja bila prazna, da vrijeme pristupa priručnoj memoriji iznosi 1T te da pristup radnoj memoriji traje 100T.
- (b) Prikazati promjenu prosječnog vremena pristupa kad bi na raspolaganju bila i početno prazna osmerostruko asocijativna priručna memorija L2 kapaciteta 1MB s linijama od 16 bajta, algoritmom zamjene LRU i latencijom 10 T.
4. (12 bodova) Neki računalni sustav koristi jednorazinsko straničenje sa stranicama veličine 4KB i potpuno asocijativnim translacijskim spremnikom (TLB) s četiri elementa. Procesor generira sljedeći niz adresa: 0x24ec, 0x78f4, 0x4ac0, 0xb5a6, 0x94de, 0x410d, 0xbd60. Translacijski spremnik (TLB) koristi FIFO algoritam zamjene (First In First Out, tj. u slučaju zamjene izbacuje se element koji je vremenski prvi upisan u TLB), a njegov početni sadržaj zadan je donjom tablicom (TLB je tijekom ranijeg rada računala popunjen redoslijedom odozgo prema dolje, odnosno element u gornjem retku tablice prvi je upisan u TLB).

Početni sadržaj relevantnog dijela stranične tablice:

Početni sadržaj TLB-a:

Valid	Oznaka	Fizička stranica
1	11	12
1	7	4
1	3	6
0	0	0

Valid	Fizička stranica
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

Za svaki pristup memoriji odredite je li riječ o pogotku TLB-a, pogotku stranične tablice, ili straničnom promašaju. Odredite i konačno stanje TLB-a i relevantnog dijela stranične tablice. U slučaju da je neku stranicu potrebno dohvatiti sa diska, pretpostavite da se njen indeks fizičke stranice dobiva inkrementiranjem najvećeg do tada korištenog indeksa fizičke stranice, te da je kapacitet RAM-a dovoljno veliki da nema potrebe za izbacivanjem stranica iz RAM-a za navedene pristupe memoriji.

5. (12 bodova) Razmatramo sljedeći izraz:

$$c = s \cdot (a \bullet b)$$

pri čemu je  $s$  skalar, a  $a$  i  $b$  su 64-komponentni vektori (polja) brojeva s pomičnim zarezom jednostruke preciznosti. Na početku izvođenja programa vektori su pohranjeni u memoriju računala pri čemu je početna adresa vektora  $a$  \$a\$, vektora  $b$  \$b, skalara  $s$  \$s te skalara  $c$  \$c. Napišite programski odsječak za vektorski procesor koji računa navedeni izraz. Vektorski procesor ima vektorske registre  $v_0 - v_7$  gdje je svaki registar veličine 256 bita. Za svaku od instrukcija u programskom odsječku označite tip.

Napomena: na raspolaganju imamo instrukciju koja računa sumu svih brojeva u vektorskom registru i ima prototip  $\text{sumv } v, r$  gdje  $v$  predstavlja vektorski operand, a  $r$  skalarni operand u koji se sprema rezultat. Skalarni produkt računa se prema izrazu:

$$a \bullet b = \sum_{i=0}^{n-1} a_i \cdot b_i = a_0 \cdot b_0 + a_1 \cdot b_1 + \dots + a_{n-1} \cdot b_{n-1}$$

BONUS: predložite implementaciju gdje se instrukcija  $\text{sumv}$  poziva samo jednom.