

Zavod za elektroniku, mikroelektroniku,  
računalne i inteligentne sustave

## Arhitektura računala 2

### Završni ispit

- Na vanjsku sabirnicu računala s 16-bitnom adresnom i 8-bitnom podatkovnom sabirnicom spojena su tri memorijska modula. Na početku adresnog prostora smješten je memorijski ROM modul kapaciteta 16KiB. Kontinuirani adresni prostor neposredno nakon ROM-a zauzima memorijski RAM modul s priključcima  $A_0-A_{13}$ ,  $D_0-D_7$ ,  $E$ ,  $\overline{E}$  i  $R/\overline{W}$ . Ostatak adresnog prostora zauzeo je preostali RAM modul koji ima nepoznati broj adresnih priključaka te priključke  $D_0-D_7$ ,  $E$ ,  $\overline{E}$  i  $R/\overline{W}$ . Zadaci:
  - Odredite minimalan broj adresnih priključaka posljednjeg RAM modula, tako da se maksimalno iskoristi adresni prostor računala.
  - Odredite kapacitet za oba RAM modula, te njihove raspone adresa.
  - Prikažite shemu spajanja svih modula na vanjske sabirnice računala.
- Predložite prijevod priloženog potprograma za arhitekturu MIPS pod pretpostavkom da se argumenti  $p$ ,  $p_2$  i  $q$  prenose preko registara  $\$4$ ,  $\$5$  i  $\$6$ .

```
void memcpy(int *p, int *p2, int *q, int *r){
    do{
        *q=*p;
        ++p;
        ++q;
    } while (p!=p2)
}
```

Prikažite dvije verzije potprograma: jednu neoptimiranu te jednu optimiranu za procesor s jednostrukim izdavanjem.

- Razmatramo procesor s dvorazinskom priručnim memorijom s izravnim preslikavanjem. Veličina linije za obje razine memorije je 16B. Veličina L1 memorije je 32B, dok je veličina L2 memorije 10KiB. Razmatramo određivanje ukupnog zbroja elemenata matrice  $8 \times 8$  koja je deklarirana kao `char m[8][8]`, a inicijalno je smještena u priručnoj memoriji L2. Zadaci:
  - Izračunajte omjer pogodaka i promašaja memorije L1 za slučaj kada iteriranje izvodimo prvo po stupcima a onda po retcima.
  - Izračunajte omjer pogodaka i promašaja memorije L1 za slučaj kada iteriranje izvodimo prvo po retcima a onda po stupcima.

4. Računalo s virtualnim memorijskim sustavom sa straničenjem posjeduje potpuno asocijativni translacijski spremnik (TLB) veličine 4 elementa koji koristi algoritam zamjene LRU. Svaki element TLB-a može se opisati trojkom (zastavica valjanosti, broj virtualne stranice, broj fizičke stranice). Stranice su veličine 4KiB, a elementi stranične tablice mogu se prikazati kao parovi (zastavica valjanosti, broj fizičke stranice). Razmatramo situaciju neposredno po uključenju računala, pri čemu su TLB i stranična tablica „prazni“ (tj. svi elementi imaju bit valjanosti = 0). Računalo pristupa sljedećim virtualnim adresama (0x označava heksadekadsku vrijednost): 0x24EC, 0x78F4, 0x4AC0, 0xB5A6, 0x94DE, 0x410D, 0xBD60

Za svaki od navedenih pristupa odredite radi li se o (i) pogotku TLB; (ii) promašaju TLB i pogotku u straničnoj tablici, ili (iii) promašaju TLB i promašaju u straničnoj tablici. Odredite i prikažite konačni sadržaj TLB-a i relevantnog (korištenog) dijela stranične tablice. Pri tome pretpostavite da se, prilikom pokušaja pristupa virtualnoj stranici koja još nije u fizičkoj memoriji, za odgovarajuću fizičku stranicu alocira prvi slobodni blok (najniže adrese), počevši od 0.

5. Prikažite izvođenje dretvi s priložene slike na superskalarnim procesorima koji izdaju do dvije instrukcije po taktu. Pretpostavite da su protočne strukture specijalizirane te da se izmjena dretvi provodi prema kružnom prioritetu. Zadatak riješite pod sljedećim pretpostavkama:

- (a) simultano višedretveni (SMT) procesor,
- (b) procesor s finoizrnatom višedretvenosti,
- (c) procesor s gruboizrnatom višedretvenosti.

1	2 2	3	4
	2		
1 1	2	3	4
1			4 4
1	2 2	3 3	
	2	3	4

Slika uz zadatak 5.

5) a) gmbozmata

1  
2 2  
2  
2  
3  
4  
1 1  
1  
1  
2 2  
2  
2 3  
4  
4 4  
3 3  
3  
4

b) finozmata

1  
2 2  
3  
4  
1 1  
2  
3  
4  
1  
2  
3 3  
4 4  
1  
2 2  
3  
4  
2

c) SMT

1 2  
3  
4 1  
2 3  
1 4  
3 2  
4 4  
1 2  
3  
2 4  
3  
1  
2  
2

② - neoptimirano:

```
lw lw $t0, 0($4) // t0 = p
lw $t1, 0($5) // t1 = p2
lw $t2, 0($6) // t2 = q

Loop: lw $t3, 0($t0) // t3 = *p
      nop
      sw $t3, 0($t2) // *q = $t3
      addi $t0, $t0, 4 // ++p
      addi $t2, $t2, 4 // ++q
      bne $t0, $t1, Loop // while (p != p2)
```

optimirano:

```
lw $t0, 0($4)
lw $t1, 0($5)
lw $t2, 0($6)
```

```
Loop: lw $t3, 0($t0)
      addi $t0, $t0, 4
      sw $t3, 0($t2)
      addi $t2, $t2, 4
      bne $t0, $t1, Loop
```

5) a) gmbozmata

1  
2 2  
2  
2  
3  
4  
1 1  
1  
1  
2 2  
2  
2 3  
4  
4 4  
3 3  
3  
4

b) finozmata

1  
2 2  
3  
4  
1 1  
2  
3  
4  
1  
2  
3 3  
4 4  
1  
2 2  
3  
4  
2

c) SMT

1 2  
3  
4 1  
2 3  
1 4  
3 2  
4 4  
1 2  
3  
2 4  
3  
1  
2  
2

① ROM - 16 KiB }  $\Rightarrow$  ROM =  $2^{14}$  B  
 Data - 1 B

RAM<sub>1</sub> - A<sub>0</sub>-A<sub>13</sub>, D<sub>0</sub>-D<sub>7</sub>,  $\bar{E}$ , E, R/W  $\Rightarrow$  16 KiB =  $2^{14}$  B

RAM<sub>2</sub> - D<sub>0</sub>-D<sub>7</sub>,  $\bar{E}$ , E, R/W

16 BITOVA za adres  $\Rightarrow$  memorijski prostor =  $2^{16}$  B

RAM<sub>2</sub> =  $2^{16}$  B -  $2 \cdot 2^{14}$  B =  $2^{16} - 2^{15}$  B =  $2^{15}$  B

a) RAM<sub>2</sub> - 15 adresnih priključaka

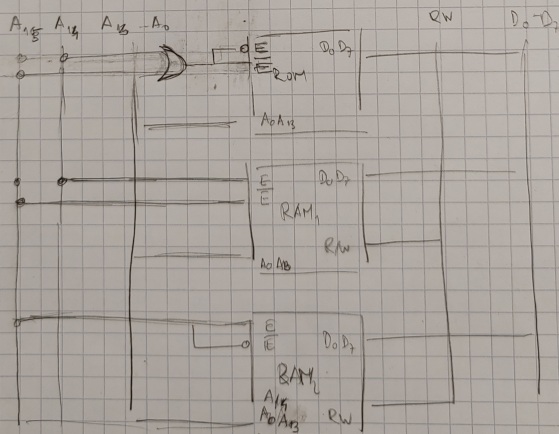
b) RAM<sub>1</sub> =  $2^{14}$  B = 16 KiB

RAM<sub>2</sub> =  $2^{15}$  B = 32 KiB

c) Rom 00

RAM<sub>1</sub> 01

RAM<sub>2</sub> 1X



4) 4 KiB  $\Rightarrow 2^{12}$  B

TLB - 4 elementa potpuno asocijativno

1) TLB

0	-	-
0	-	-
0	-	-
0	-	-



6) 0x3408

TLB miss  
PT miss

0	3	5
1	7	2
1	4	3
1	B	4

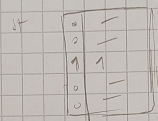
(4)	1	1
(7)	1	3
(7)	1	2
(3)	1	5
(B)	1	4

2) 0x24EC

TLB miss, PT miss

1	2	1
0	-	-
0	-	-
0	-	-

write  
add  
0



7) 0x410D TLB hit, PT hit

3) 0x76F4

TLB miss, PT miss

1	2	1
1	7	2
0	-	-
0	-	-

0	0	0
2	1	1
7	1	2

8) 0xB060 TLB hit, PT hit

4) 0x4AC0

TLB miss, PT miss

1	2	1
1	7	2
1	4	3
0	-	-

(2)	1	1
(4)	1	3
(7)	1	2

5) 0xB5A6

TLB miss, PT miss

1	2	1
1	7	2
1	4	3
1	B	4

(2)	1	1
(4)	1	3
(7)	1	2
(B)	1	4