

Računarske arhitekture - 19413

Zadaća 1.

Zadatak 1.

Za prikazanu funkciju napisati i objasniti MIPS asemblerski kod.

```
int f(int array[], int size)
{
    int cnt = 0;
    for (int i = 0; i < size - 1; ++i)
    {
        if (array[i] > array[i + 1]) ++cnt;
    }
    return cnt;
}
```

f:

Prolog funkcije - spremanje važnih registara na stek

addiu \$sp, \$sp, -24

sw \$ra, 20(\$sp) # Spremanje povratne adrese

sw \$fp, 16(\$sp) # Spremanje frame pointer-a

move \$fp, \$sp # Postavljanje frame pointera

Inicijalizacija varijabli

sw \$4, 12(\$fp) # Spremanje adrese niza

sw \$5, 8(\$fp) # Spremanje veličine niza

sw \$zero, 4(\$fp) # Inicijalizacija cnt na 0

sw \$zero, 0(\$fp) # Inicijalizacija varijable petlje i na 0

j loop_condition # Skok na uslov petlje

loop_condition:

lw \$t1, 0(\$fp) # Učitavanje i

lw \$t2, 8(\$fp) # Učitavanje veličine niza

addiu \$t2, \$t2, -1 # Veličina niza - 1

slt \$t1, \$t1, \$t2 # Usporedba i < veličina - 1

beqz \$t1, loop_end # Izlazak iz petlje ako je i >= veličina niza - 1

```

# Tijelo petlje
lw $t1, 0($fp)      # Učitavanje i
sll $t1, $t1, 2      # i * 4 (jer je svaki int veličine 4 bajta)
lw $t2, 12($fp)      # Učitavanje adrese niza
$t1, $t1, $t2        # Računanje adrese od array[i]
lw $t2, 0($t1)       # Učitavanje array[i]
lw $t1, 4($t1)        # Učitavanje array[i+1]
slt $t1, $t1, $t2     # Upoređivanje array[i+1] < array[i]
beqz $t1, increment_i # Skok na inkrementiranje varijable i ako uslov nije ispunjen

# Inkrementiranje cnt varijable
lw $t1, 4($fp)        # Učitavanje cnt
addiu $t1, $t1, 1     # Inkrementiranje cnt
sw $t1, 4($fp)        # Spremanje nove vrijednosti cnt

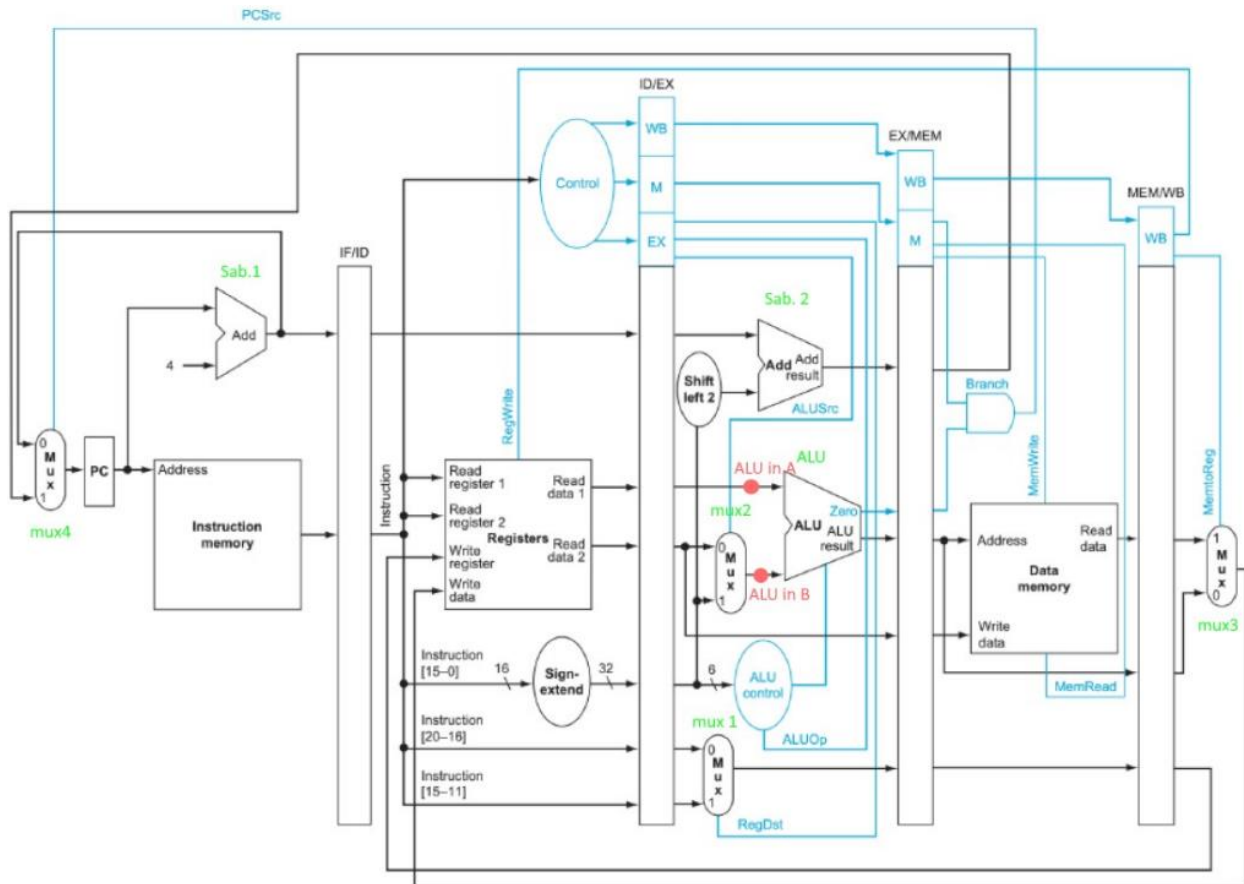
increment_i:
lw $t1, 0($fp)        # Učitavanje i
addiu $t1, $t1, 1     # Inkrementiranje i
sw $t1, 0($fp)        # Spremanje nove vrijednosti i
j loop_condition      # Skok nazad na uslov petlje

loop_end:
# Epilog funkcije - vraćanje steka na staro i povratak
lw $ra, 20($sp)        # Vraćanje povratne adrese
lw $fp, 16($sp)        # Vraćanje frame pointer-a
addiu $sp, $sp, 24     # Oslobađanje prostora na steku
jr $ra                # Povratak iz funkcije

```

Zadatak 2.

Neka se na izlaznom portu memorije instrukcija nalazi instrukcija: `addi r2, r4, 6`. Pretpostaviti da su u memoriji podataka na svim adresama upisane vrijednosti 0, a vrijednost registra `ri` = `i`.



a) Za multipleksere na slici (mux1, mux2, mux3, mux4), sabirač (sab2) i ALU prikazati ulazne i izlazne vrijednosti, kao i vrijednosti kontrolnih signala za multipleksere (mux1, mux2, mux3, mux4) kada je ova instrukcija u odgovarajućim fazama.

mux1: Ulazi su redni broj registra r2 i proširena konstanta 6, a izlaz je redni broj registra r2 jer je instrukcija tipa I, pa je signal RegDst postavljen na 0.

mux2: Ulazi su vrijednost iz registra r2 i proširena konstanta 6, a izlaz je proširena konstanta 6 jer je signal AluSrc postavljen na 1.

mux3: Ulazi su adresa izračunata u ALU-u ($6 + 4 = 10$) i rezultat iz ALU-a (10), a izlaz je rezultat iz ALU-a jer signal MemToReg je postavljen na 0.

mux4: U prvom ciklusu izvršenja date instrukcije, mux4 prima adresu te instrukcije na gornjem ulazu, čija točna vrijednost nije navedena u postavci, dok na donjem ulazu prima vrijednost izračunanu kroz sab2. Budući da prethodna instrukcija koja se izvršavala nije poznata, niti je poznata vrijednost Branch signala koji upravlja ovim mux-om, ne možemo odrediti koja će vrijednost biti proslijeđena na izlaz ovog multiplexera u prvoj fazi. U petoj fazi izvršenja ove instrukcije, situacija je slična prvoj fazi. Na gornjem ulazu se i dalje proslijeđuje adresa te instrukcije sabrana sa 4, dok na donjem ulazu dolazi adresa grananja izračunata kroz sab2. Međutim, pošto nije riječ o instrukciji grananja, Branch signal se postavlja na 0. Izlaz iz ovog multiplexera će stoga biti adresa trenutne instrukcije sabrana sa 4, što ukazuje da je sljedeća instrukcija koja će se izvršiti ona koja se nalazi iza addi r2, r4, 6.

ALU: Na ulazu su vrijednosti iz registra r4 (4) i proširena konstanta 6, a na izlazu je rezultat operacije (10).

sab2: Na gornjem ulazu, sab2 prima adresu trenutne instrukcije iz memorije instrukcija, koja je sabrana sa 4. Na donjem ulazu, sab2 prima proširenu konstantu 6 koja je shiftana ulijevo dva puta. Na izlazu, sab2

daje rezultat koji je zbir adrese trenutne instrukcije sabrane sa 4 i proširene konstante 6. Ova vrijednost predstavlja odredišnu adresu koja se koristi u procesu izvršavanja instrukcija, iako u ovom slučaju nije povezana s grananjem.

b) Koje vrijednosti će se nalaziti na ulaznom portu u skupu registara kada je ova instrukcija u odgovarajućim fazama.

Faza 2: Ulazi su redni broj registra r4 (4) i redni broj registra r2 (2). Na taj način se osigurava da se iz registara čitaju odgovarajuće vrijednosti za operaciju.

Faza 5: Ulazi su rezultat izračunat u ALU-u (10) i redni broj registra r2 (2). Ovdje se vrijednost rezultata iz ALU-a proslijeđuje na ulaz skupa registara kako bi se upisala u odgovarajući registar.

c) Pretpostaviti da je došlo do greške u kontrolnoj jedinici MIPS procesora i da su vrijednosti svih kontrolnih signala 0; koji će se problemi desiti u svakoj od faza izvršenja ove instrukcije?

U fazi 2: Ako bi svi kontrolni signali bili postavljeni na 0, ne bi se koristila konstanta 6 iz instrukcije već bi se umjesto nje koristila vrijednost iz registra r2. To bi dovelo do pogrešnog rezultata operacije.

U fazi 5: Ako bi kontrolni signal RegWrite bio postavljen na 0, ne bi se mogla izvršiti operacija upisa rezultata izračunatog u ALU-u u registar. Umjesto toga, koristila bi se vrijednost iz registra r2, što bi rezultiralo pogrešnim rezultatom. Ostali signali, poput čitanja registara ili adresiranja memorije, ne bi direktno utjecali na ovu fazu, ali bi mogli uzrokovati neispravno ponašanje ako bi bili postavljeni na nulu.

Zadatak 3.

Sljedeća sekvenca instrukcija se izvršava u oglednoj protočnoj strukturi sa proslijeđivanjem:

```
add r1, r2, r3
sw r1, 6(r10)
andi r4, r1, 5
lw r2, 6(r10)
sub r3, r2, r1
and r5, r4, r3
```

Pretpostaviti da su u memoriji podataka na svim adresama upisane vrijednosti 0, a vrijednost registra ri = i.

a) Napisati raspored faza izvršavanja po ciklusima. Napisati gdje i kada je proslijeđivanje potrebno.

Između linija **lw r2, 6(r10)** i **sub r3, r2, r1** ćemo dodati jedan zastoje jer se vrijednost registra r2 proslijeđuje nakon MEM faze

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
<i>add r1, r2, r3</i>	IF	ID	EX	MEM	WB						
<i>sw r1, 6(r10)</i>		IF	ID	EX	MEM	WB					
<i>andi r4, r1, 5</i>			IF	ID	EX	MEM	WB				
<i>lw r2, 6(r10)</i>				IF	ID	EX	MEM	WB			
<i>nop</i>					IF	ID	EX	MEM	WB		

<i>sub r3, r2, r1</i>						IF	ID	EX	MEM	WB	
<i>and r5, r4, r3</i>							IF	ID	EX	MEM	WB

Prvo prosljeđivanje je nakon EX faze prve instrukcije iz EX/MEM registra (vrijednost registra r1).

Drugo prosljeđivanje je nakon MEM faze druge instrukcije iz MEM/WB registra (vrijednost registra r1).

Treće prosljeđivanje je nakon MEM faze četvrte instrukcije iz MEM/WB registra (vrijednost registra r2).

Četvrto prosljeđivanje je nakon EX faze šeste instrukcije iz EX/MEM registra (vrijednost registra r3).

b) Napisati konačne vrijednosti svih registara kojima je promijenjena vrijednost u toku izvršenja sekvence, kao i instrukciju koja je postavila konačnu vrijednost registra.

Registar r1 sadrži konačnu vrijednost 5 (add r1,r2,r3)

Registar r2 sadrži konačnu vrijednost 5 (lw r2, 6(r10))

Registar r3 sadrži konačnu vrijednost 0 (sub r3, r2, r1)

Registar r4 sadrži konačnu vrijednost 5 (andi r4, r1, 5)

Registar r5 sadrži konačnu vrijednost 0 (and r5, r4, r3)

c) Ukoliko se desi kvar na jedinici za prosljeđivanje podataka takav da jedinica nikada ne detektuje da je potrebno prosljeđivanje podataka, da li će se promijeniti raspored faza izvršavanja po ciklusima ili konačne vrijednosti registara iz podzadataka a) i b)? Ako da, napisati šta je promijenjeno.

Neće doći do promjene rasporeda faza izvršavanja po ciklusima ali nećemo imati očekivane vrijednosti.

d) Ukoliko se na kraju 6. ciklusa desi kvar na jedinici za prosljeđivanje podataka tako da uvijek detektuje da je na ALU ulaz A potrebno proslijediti podatak iz EX/MEM pregradnog registra, koji registri će imati pogrešne vrijednosti nakon izvršavanja sekvence?

U tom slučaju će registar r5 imati pogrešnu vrijednost.

Zadatak 4.

Neka se sljedeća sekvenca instrukcija izvršava u oglednoj protočnoj strukturi sa prosljeđivanjem i detekcijom hazarda:

```

add r1, r2, r3
bne r1, r5, lbl
addi r3, r4, -1
lbl2: sw r2, 0(r0)
and r4, r3, r1
lbl: sub r2, r3, r4
addi r3, r3, 4
beq r3, r7, lbl2
sw r2, 0(r0)

```

Neka je vrijednost registra $r_i=i$, i neka se kontrolni hazardi rješavaju zastoјima. Odluka o grananju se donosi u ID fazi izvršenja instrukcije (pretpostaviti da postoje mehanizmi prosljeđivanja podataka u komparator u ID fazi).

Napisati raspored faza izvršavanja po ciklusima. Napisati gdje i kada je prosljeđivanje potrebno.

add r1, r2, r3 - l1

```

bne r1, r5, lbl - I2
addi r3, r4, -1 - I3
lbl2: sw r2, 0(r0) - I4
and r4, r3, r1 - I5
lbl: sub r2, r3, r4 - I6
addi r3, r3, 4 - I7
beq r3, r7, lbl2 - I8
lbl2: sw2, 0(r0) - I9
and r4, r3, r1 - I10
lbl: sub r2, r3, r4 - I11
addi r3, r3, 4 - I12
beq r3, r7, lbl2 - I13
sw r2, 0(r0) - I14

```

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.
I1	IF	ID	EX	MEM	WB													
I2		IF	ID	EX	MEM	WB												
I3		-	IF	ID	EX	MEM	WB											
I4				IF	ID	EX	MEM	WB										
I5					IF	ID	EX	MEM	WB									
I6						IF	ID	EX	MEM	WB								
I7							IF	ID	EX	MEM	WB							
I8							-	IF	ID	EX	MEM	WB						
I9									IF	ID	EX	MEM	WB					
I10										IF	ID	EX	MEM	WB				
I11											IF	ID	EX	MEM	WB			
I12												IF	ID	EX	MEM	WB		
I13													IF	ID	EX	MEM	WB	
I14													-	IF	ID	EX	MEM	WB

Prvo prosljeđivanje je nakon EX faze prve instrukcije iz EX/MEM registra (vrijednost registra r1).

Drugo prosljeđivanje je nakon MEM faze treće instrukcije iz MEM/WB registra (vrijednost registra r3).

Treće prosljeđivanje je nakon EX faze pete instrukcije iz EX/MEM registra (vrijednost registra r4).

Četvrto prosljeđivanje je nakon EX faze sedme instrukcije iz EX/MEM registra (vrijednost registra r3).

Peto prosljeđivanje je nakon EX faze desete instrukcije iz EX/MEM registra (vrijednost registra r4).

Šesto prosljeđivanje je nakon EX faze dvanaeste instrukcije iz EX/MEM registra (vrijednost registra r3).