

Fakultet elektrotehnike i računarstva
Zavod za automatiku i računalno inženjerstvo

Arhitektura računala 1

Zadaci za 1. ciklus laboratorijskih vježbi

1. Organizacija vježbe

Zadatke iz nastavka treba **izraditi prije dolaska na termin vježbe**. Na samom terminu vježbe treba demonstrirati rješenja zadataka.

Na termin vježbe treba doći s rješenjima zadataka u elektroničkom obliku, spremnima za pokretanje. Potpuno je svejedno da li rješenja demonstrirate na fakultetskom ili vlastitom računalu.

Nakon uspješne demonstracije rješavaju se zadaci koji nose od 0 do 4 boda. Uspješna **demonstracija rješenja** ne nosi bodove, no **nužan je uvjet za pristup rješavanju zadataka** za bodove. Nakon demonstracije rezultata možete i odustati od rješavanja zadataka za bodove (u tom slučaju iz vježbe dobivate 0 bodova).

Neuvjerljiva demonstracija rezultata (ako asistent/ica ima razloga vjerovati da niste samostalno izradili vježbu, bilo zbog vašeg nesnalaženja s alatima i/ili manjka suvislih odgovora na pitanja iz vježbe) **donosi 2 negativna boda**. Neuvjerljiva demonstracija rezultata također automatski znači nemogućnost rješavanja zadataka za bodove.

2. Prvi zadatak: pretvorba formata zapisa brojeva

U memoriji od adrese 500_{16} nalazi se blok podataka sa "zaglavljem". **Broj podataka u bloku** zapisan je u "zaglavlju", tj. u prvom 32-bitnom podatku na adresi 500_{16} . Iza toga slijedi niz podataka zapisanih u **16-bitnom formatu s bitom za predznak**.

Napišite program za FRISC koji će podatke iz opisanog bloka čitati i pretvarati u **8-bitni format dvojnog komplementa**, a **pretvorene brojeve će zapisivati u novi blok od adrese 1000_{16}** . Ako se neki podatak **ne može pretvoriti** zbog prevelikog opsega, u novom bloku treba upisati **8-bitnu vrijednost 80_{16}** . Podaci u početnom bloku trebaju biti: -1_{10} , $+2_{10}$, $+291_{10}$, $+37_{10}$, -700_{10} , -33_{10} .

Program treba spremi u direktorij atlas/vjezba1, asemblirati i simulirati alatima *xconas* i *xcompas*. Provjerite je li rezultatni blok (u tablici ispod) jednak onom koji ste dobili. Pri ispravljanju eventualnih logičkih grešaka u programu, primijenite točke praćenja i prekidne točke.

Početni blok (od adrese 500_{16}) prikazano po 16-bitnim riječima (osim zaglavlja)	Rezultatni blok (od adrese 1000_{16}) Prikazano po bajtovima
0000 0006, (broj podataka u bloku 32 bita) 8001, 0002, 0123, 0025, 82BC, 8021	FF, 02, 80, 25, 80, DF

Vodite računa da FRISC sve podatke zapisuje u *little-endian* poretku. Na primjer, podatak FC13 ispravno je po oktetima zapisan ovako:

adresa	podatak
2000	13
2001	FC

Neka od pitanja koja bi vam asistenti mogli postaviti:

- Što je SR, ALU, PC?
- Ako pretpostavimo da su brojevi 99999999 i 77777777 u 32 bitnom NBC formatu, koji je veći? A ako je to zapis u formatu 2'k, koji je broj veći?
- Kakvo značenje poprima zastavica C kod operacije oduzimanja i kada će biti postavljena - u kojem odnosu moraju biti brojevi koje sudjeluju u oduzimanju?
- Zbrojite dva broja te odredite stanja zastavica.
- Pretvoriti brojeve -6 i 3 u format 2'k.
- Da li naredba LOAD mijenja zastavice?
- Koje naredbe mijenjaju zastavice, a koje ne?
- Kako znamo kada je došlo do pogreške prilikom računanja sa brojevima u 2'k formatu a kada s brojevima u NBC formatu (koje zastavice treba ispitivati?)?
- Pokazati gdje se u memoriji nalazi izvorišni niz podataka?
- Pokazati gdje se u memoriji nalazi odredišni niz podataka?
- Nabrojite sve memorijske lokacije koje zauzima 2. član izvorišnog niza podataka?

3. Drugi zadatak: računanje pariteta

Za procesor FRISC treba napisati potprogram VISE_1 koji prima dva parametra preko stoga. Svaki parametar je **16-bitni broj u formatu 2'k**. Potprogram treba odrediti u kojem 16-bitnom parametru se nalazi više jedinica. Broj koji ima više jedinica treba vratiti preko registra R0, ali kao **32-bitni 2'k broj**. Ako brojevi imaju jednaki broj jedinica, treba vratiti broj 80000000_{16} .

Napisati glavni program koji će obraditi blok 16-bitnih podataka iz memorije pomoću potprograma VISE_1 na taj način da učitava po dva podatka iz bloka, šalje ih potprogramu, te na njihovo mjesto upiše 32-bitni podatak koji je vratio potprogram VISE_1.

Blok 16-bitnih podataka nalazi se neposredno iza naredbe HALT. U njemu mora biti paran broj podataka, a zaključen je sa dva 16-bitna podatka 8000_{16} i 8000_{16} koje se ne smije mijenjati. Ispitni podatci u bloku trebaju biti sljedeći (heksadekadski): 1234, FF, FF, -FF, 1, -1, 2A, 20A, 456, 123.

Izgled podataka u memoriji trebao bi biti kao u tablici. Lijevi dio tablice prikazuje početni izgled bloka s parovima 16-bitnih brojeva, a u zagradama je prikazan broj jedinica u broju. Desni dio tablice prikazuje blok nakon izvođenja programa s 32-bitnim brojevima:

Blok prije izvođenja		Blok nakon izvođenja
1234 (5)	00FF (16)	000000FF
00FF (16)	FF01 (17)	FFFFFFF01
0001 (1)	FFFF (16)	FFFFFFFFF
002A (3)	020A (3)	80000000
0456 (5)	0123 (4)	00000456
8000	8000	80008000

Program treba spremi u direktorij atlas/vjezba2. Simulirajte rad programa. Provjerite jesu li rezultati dobiveni programom ispravni.

Neka od pitanja koja bi vam asistenti mogli postaviti:

- Kojim naredbama se ostvaruje pomak ulijevo i udesno? Koja je razlika između aritmetičkog i logičkog pomaka?
- Koja je razlika između pomaka (*shift*) i rotacije?
- Pokazati gdje se u memoriji nalazi niz podataka koji se obrađuje?
- Kako se može ispitati stanje bita u registru?
- Pokazati sadržaj registara procesora?
- Pokrenite program s uključenom točkom praćenja 2 (*tracepoint*).
- Pokrenite program s postavljenom prekidnom točkom na zadanoj naredbi (*breakpoint*).
- Koji registar u vašem programu koristite kao brojač za petlju?
- Pokažite naredbu kojom čitate podatak iz bloka?
- Kako i gdje se spremaju registri u potprogramu?
- Kako se prenose parametri u potprogram?