

Fakultet elektrotehnike i računarstva
Zavod za automatiku i računalno inženjerstvo

Arhitektura računala 1

Zadaci za 2. ciklus laboratorijskih vježbi

1. PRIPREMA

Proučiti:

- Programske primjere iz 6. dijela predavanja - *Povezivanja računala s okolinom: ulazno izlazni prijenos podataka (FRISC-IO)*.

2. 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 računalu ili vlastitom laptopu.

3. VJEŽBA 3 – UVJETNE, BEZUVJETNE I PREKIDNE VANJSKE JEDINICE

U računalnom sustavu nalaze se FRISC i tri vanjske jedinice: *vj1*, *vj2* i *vj3* (redom na adresama FFFF1000₁₆, FFFF2000₁₆ i FFFF3000₁₆). *Vj1* radi u bezuvjetnom načinu, *vj2* u uvjetnom a *vj3* u prekidnom načinu.

FRISC u glavnom programu u beskonačnoj petlji ispituje spremnost *vj2*. Kad *vj2* postane spremna, na nju šalje podatak s lokacije BROJ (inicijalno postaviti u 0). Za vrijeme rada glavnog programa, *vj3* koja radi u prekidnom načinu (spojena je na *INT0*) u određenim trenucima generira prekid. U prekidnom potprogramu čita se 2'k podatak s bezuvjetne ulazne *vj1*, obrađuje se pomoću potprograma OBRADI, te se rezultat zapisuje na lokaciju BROJ. Također se na *vj3* šalje ukupan broj dosad poslanih podataka na *vj2*.

Potprogram OBRADI preko stoga prima 2'k podatak pročitani s *vj1*. Potprogram ulaznim podatkom sa *vj1* potencira broj 2 (primaju se mali pozitivni brojevi, nema prekoračenja opsega) - rezultat=2^{podatak s *vj1*} i taj broj kao rezultat obrade vraća preko R0. Potprogram mora čuvati vrijednosti registara, a pozivatelj uklanja parametar sa stoga.

Kada se sa *vj1* primi prvi negativan podatak, treba zabraniti daljnje prekide i zaustaviti izvođenje programa.

Vanjske jedinice imaju općenitu građu prikazanu na predavanjima (osim *vj3* - vidi napomenu):

vj1 FFFF1000 - prijenos podatka

vj2 FFFF2000 - prijenos podatka
FFFF2004 - ispitivanje i brisanje bistabila stanja

vj3 FFFF3000 - prijenos podatka
FFFF3004 - dojava prihvata prekida i ispitivanje stanja
FFFF3008 - dojava kraja posluživanja
FFFF300C - omogućavanje/zabranjivanje postavljanja prekida
-upis 0 zabranjuje, a upis 1 dozvoljava postavljanje zahtjeva za prekid
-**NAPOMENA:** za razliku od predavanja, *vj3* ima inicijalno zabranjeno postavljanje prekida

Program spremite u direktorij (kojeg već imate, jer ste ga kreirali skriptom *arh1* u pripremi za prvu vježbu) *atlas/vjezba3* (u suprotnom program se neće ispravno simulirati), te iz istog direktorija treba pokretati alate *xconas* i *xcompas*. Odsimulirajte program te provjerite njegovu ispravnost. Za provjeru ispravnosti potrebno je vanjskim jedinicama uključiti točku praćenja *trace 5* (*trace 5* ispisuje poslanih i primljene podatke u dekadskoj bazi).

vj3 generira prekid svakih 10000 taktova clock-a						
vj1 šalje	9	7	5	3	1	-1
vj2 prima	512	128	32	8	2	KRAJ

Neka od pitanja koja bi vam asistenti/ca mogli postaviti:

- Koja je razlika između naredaba RET/RETI/RETN?
- Što se događa prilikom prihvata prekida?
- Kako zabraniti prihvaćanje prekida?
- Gdje se nalazi i čemu služi bit GIE?
- Pokažite gdje ispituje stanje uvjetne vanjske jedinice?
- Koji registri ulaze u kontekst prekidnog potprograma?
- Gdje spremate kontekst u vašem prekidnom potprogramu?
- Na kojoj adresi se nalazi prekidni vektor, a na kojoj prekidni potprogram?
- Što sve morate u prekidnom potprogramu napraviti s prekidnom VJ?

4. VJEŽBA 4 – PRIJENOS PODATAKA POMOĆU JEDINICE DMA

U računalnom sustavu nalaze se FRISC, CT (adresa FFFF0000₁₆), DMA-jedinica (adresa FFFF1000₁₆) i bezuvjetna vanjska jedinica BVJ (adresa FFFFFFFC₁₆).

Korištenjem sklopa CT svakih 100 μ s treba pomoću jedinice DMA prenijeti blok od 10₁₀ podataka iz bezuvjetne vanjske jedinice u memoriju od adrese 1F0₁₆.

Sklop CT treba podesiti da postaje spreman svakih 100 μ s. Spremnost sklopa CT treba programski ispitivati, te kad on postane spreman treba **inicijalizirati DMA jedinicu i pokrenuti prijenos bloka podataka**. Na ulaz CNT sklopa CT spojen je signal takta frekvencije 10 MHz.

Svakih 100 μ s jedinica DMA treba prenijeti 10₁₀ podataka iz vanjske jedinice BVJ u memoriju od adrese 1F0₁₆ (prvih 10₁₀ podataka od adrese 1F0₁₆, sljedećih 10₁₀ od adrese 218₁₆, itd.).

Jedinica DMA prenosi podatke **krađom ciklusa**. **Kraj prijenosa** treba ispitivati programski.

Kada se prenesu 10₁₀ blokova od 10₁₀ podataka potrebno je zaustaviti procesor.

Program spremite u direktorij (kojeg već imate, jer ste ga kreirali skriptom *arh1* u pripremi za prvu vježbu) *atlas/vjezba4* (u suprotnom program se neće ispravno simulirati), te iz istog direktorija treba pokretati alate *xconas* i *xcompas*. Odsimulirajte program te provjerite njegovu ispravnost. U jedinici DMA možete točkama praćenja pratiti prijenose pojedinih podataka: točkom praćenja *trace* 1 možete dobiti ispis o prijenosu pojedinih podataka (ispisi su u heksadekadskoj bazi), a točka praćenja *trace* 3 čeka na vašu potvrdu da nastavi s radom. Kod jedinice CT uključivanjem točke praćenja 5 dobijate ispis svih akcija provedenih nad CT-om (upis u registar, kraj ciklusa, itd).

vj1 šalje redom brojeve: 1, 2, 3, ... 100₁₀

Neka od pitanja koja bi vam asistenti/ca mogli postaviti:

- Koje vrste DMA prijenosa podržava jedinica FRISC-DMA?
- Kako se zadaje da li jedinica FRISC-DMA generira prekid?
- Kada jedinica FRISC-DMA generira prekid/postaje spremna?
- Opišite kontrolnu riječ za sklop FRISC-CT!
- Kada FRISC-CT generira prekid/postaje spreman?
- Što FRISC-CT radi kad mu brojač postane jednak ničtici?
- Pokažite gdje se u vašem programu zadaje koliko podataka DMA treba prenijeti?