

Fakultet elektrotehnike i računarstva
Zavod za automatiku i računalno inženjerstvo

Arhitektura računala 1

Zadaci za 3.ciklus laboratorijskih vježbi

1. Priprema

Proučiti:

- Sva predavanja trećeg ciklusa predavanja – procesor ARM.

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 ili vlastitom računalu.

3. Vježba 5 – osnove programiranja procesora ARM

Za procesor ARM **napišite potprogram za kubiranje (KUB) i potprogram za cjelobrojno dijeljenje uzastopnim oduzimanjem (DIV)** 32-bitnih brojeva u 2^k formatu.

Parametre treba prenositi stogom, rezultat vraćati registrom. Potprogrami moraju čuvati kontekst (osim registara kojima vraćaju rezultat).

Napišite **glavni program koji** korištenjem potprograma KUB i DIV **izračunava** vrijednost funkcije $y = (x^3 - 1) / (2 * x)$.

Ulazne vrijednosti pohranjene su od adrese 100₁₆, rezultate treba spremati od adrese 400₁₆. Ulazni niz zaključen je podatkom 80000000₁₆ i sadrži sljedeće podatke: 0, 3, 6, -1, -6.

Simulirajte rad programa. 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.

VAŽNO: Da bi mogli ispravno asemblirati i simulirati vaš program, u terminalu se morate pozicionirati u direktorij pripadne vježbe. Npr. za izradu pete vježbe trebate se pozicionirati u direktorij *atlas/vjezba5* pomoću naredbe '**cd atlas/vjezba5**'. Sve dalje vezano za pojedinu vježbu obavljate u njenom direktoriju (pokretanje programa ATLAS-a, spremanje vaših programa, itd.). Za podsjetu ponovno pročitajte pripremu za 1. laboratorijsku vježbu.

Početni blok (od adrese 100 ₁₆)	Rezultantni blok (od adrese 400 ₁₆)
0000 0000,	0000 0000,
0000 0003,	0000 0004,
0000 0006,	0000 0011,
FFFF FFFF,	0000 0001,
FFFF FFFA,	0000 0012
8000 0000	

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

- Koje naredbe postavljaju zastavice?
- Kako se koristi stog kod procesora ARM? Koje su razlike u odnosu na FRISC?
- Kako se sprema kontekst?
- Kako se prenose parametri stogom, kako lokacijama iza programa?
- Nabrojite nekoliko načina adresiranja procesora ARM?
- Koji način adresiranja koristite pri dohvat i spremanju podataka u memoriju?
- Koji način adresiranja koristite pri dohvat i spremanju podataka na stog?
- Objasnite mehanizam poziva potprograma kod procesora ARM?

4. Vježba 6 – ARM vanjske jedinice

Pomoću ARM-a, sklopova GPIO i RTC, te LCDa ostvariti blesimeter za natjecanje u slovkanju.

Na lokaciji 400₁₆ nalazi se niz ASCII znakova koji predstavlja riječ koju natjecatelj mora slovhati. Niz je zaključen nul-znakom ('\0', ASCII kod 0). **Svake sekunde program treba ispisati sljedeće slovo iz niza na LCD-u.** Nakon što se ispiše cijeli niz, ispis se ponavlja ispočetka.

Na sklop RTC spojen je signal frekvencije 256 Hz. Adresa sklopa RTC je FFFFFFFE00₁₆. **Sklop RTC treba podesiti da generira prekid nakon što istekne 1s, te u prekidnoj rutini treba obaviti ispis znakova na LCD.** Za ispis jednog znaka na LCD koristiti potprogram *LCDWR* (poglavlje „Vanjske jedinice za procesor ARM“ iz predavanja, primjer: „Ispis teksta na LCD“). Prekidna rutina mora čuvati kontekst, a poziciju unutar niza znakova treba čuvati na memorijskoj lokaciji *INDEX*. Glavni program izvodi beskonačnu petlju.

LCD (opisan u predavanju) spojen je na vrata B vanjske jedinice GPIO (GPIO-a je na adresi FFFFFFF00₁₆). LCD radi tako da primljeni ASCII-znak uvijek postavlja na krajnje desno mjesto i istisne krajnje lijevi znak. LCD se briše slanjem ASCII-znaka za *CR* (*carriage return*) koji ima vrijednost D₁₆. Prikaz trenutnog sadržaja na LCD-u vrši se slanjem ASCII-znaka za *LF* (*line feed*) koji ima vrijednost A₁₆.

Ispis na LCDu mora izgledati ovako: ">>>X<<<" (X označava trenutno slovo, ASCII kod znaka '>' je 3E₁₆, a znaka '<' 3C₁₆).

Simulirajte program i **isprobajte njegov rad pomoću niza znakova navedenog u tablici**. Točke praćenja *trace 1* (ispis registara) i *trace 2* (komunikacija procesora i sklopa GPIO) u komponenti *gpio* omogućuju detaljnije praćenje rada komponente GPIO. Točka praćenja *trace 7* u komponenti *lcd* omogućava detaljnije praćenje stanja LCD-a.

VAŽNO: Da bi mogli ispravno asemblirati i simulirati vaš program, u terminalu se morate pozicionirati u direktorij pripadne vježbe. Npr. za izradu šeste vježbe trebate se pozicionirati u direktorij *atlas/vjezba6* pomoću naredbe '**cd atlas/vjezba6**'. Sve dalje vezano za pojedinu vježbu obavljate u njenom direktoriju (pokretanje programa ATLAS-a, spremanje vaših programa, itd.). Za podsjetu ponovno pročitajte pripremu za 1. laboratorijsku vježbu.

riječ 'Internationalisation'	DB 49,6e,74,65,72,6e,61,74,69,6f,6e,61,6c,69,73,61,74,69,6f,6e,0
------------------------------	--

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

- Opišite mehanizam obrade iznimke kod procesora ARM? Nabrojite vrste iznimaka.
- Koja je razlika između prekidnog i običnog potprograma?
- Što omogućuje bržu obradu kod *FIQ* prekida, u odnosu na obične prekide?
- Koje sabirnice se koriste kod ARM-a i koja je njihova osnovna namjena?
- Na koju sabirnicu je spojen sklop GPIO i zašto?
- Opišite način komunikacije između jedinice GPIO i LCDa?