



JMBAG



IME & PREZIME



BROJ BODOVA

GRADA RAČUNALA

Drugi kolokvij, 03. veljače 2020.

Obavezno potpišite i predajte sve papire sa zadacima. Na kolokvijju je dozvoljeno imati samo službeni podsjetnik s web stranice kolegija.

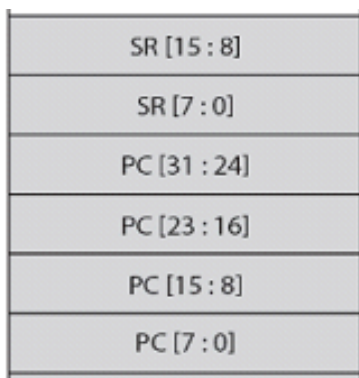
Zadatak 1. (20 bodova)

Slika 1. prikazuje stanje stogova neposredno **PRIJE** posljednje instrukcije RTE (*Return from Exception*) **prekidnog** programa. Nevješti programer je umjesto instrukcije RTE odlučio upotrijebiti programski odsječak za koji je bio uvjeren da obavlja identičnu funkciju povratka iz prekida:

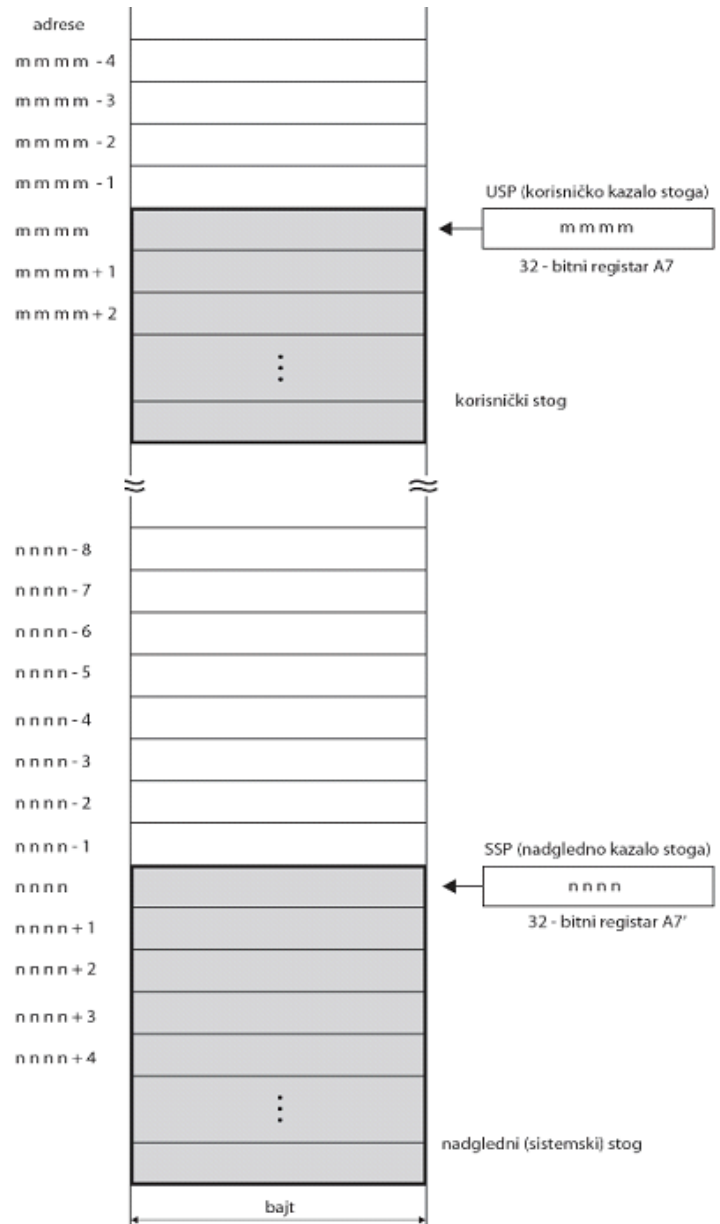
```
MOVE.W (A7)+, SR
RTS
```

Uz pretpostavku da se procesor u trenutku kada se dogodio prekid nalazio u **KORISNIČKOM** načinu rada, te da se tijekom izvođenja gornjeg odsječka nije dogodio novi prekid, **ODREDITE** promijenjene sadržaje registara procesora MC 68000 te promjene koje su nastupile u stogu (ili stogovima).

Kratko komentirajte posljedice izvođenja gornjeg programskog odsječka! U objašnjenju vam može pomoći slika 2. koja vam govori o minimalnom kontekstu.



Slika 2.



Slika 1.

JMBAG

IME & PREZIME

BROJ BODOVA

GRAĐA RAČUNALA

Drugi kolokvij, 03. veljače 2020.

Zadatak 2. (15 bodova)

Glavna memorija ima vrijeme pristupa 260 ns. Vrijeme pristupa priručnoj memoriji je 25 ns. Za omjer pogotka $H = 0.98$ ocijenite srednje vrijeme pristupa memorijskom sustavu t_A .

JMBAG

IME & PREZIME

BROJ BODOVA

GRAĐA RAČUNALA

Drugi kolokvij, 03. veljače 2020.

Zadatak 3. (15 bodova)

Opisati instrukciju RISC procesora `brl` (*Branch and link*) i njenu ulogu u prijenosu upravljanja s jednog programskog modula na drugi.

JMBAG

IME & PREZIME

BROJ BODOVA

GRAĐA RAČUNALA

Drugi kolokvij, 03. veljače 2020.

Zadatak 4. (20 bodova)

Flynnova klasifikacija arhitekture računala. Objasniti, nacrtati te odgovoriti koja se kategorija arhitekture, strogo teorijski gledano, ne može realizirati. Podrobnije grafički predočiti SIMD kategoriju.

JMBAG

IME & PREZIME

BROJ BODOVA

GRAĐA RAČUNALA

Drugi kolokvij, 03. veljače 2020.

Zadatak 5. (30 bodova)

Napišite funkciju analognu C-ovskoj funkciji `int strlen(char* str)`, koja računa duljinu danog stringa. Kraj stringa označen je nul-znakom (nul-znak ne ulazi u duljinu). Funkcija parametre prima i vraća preko registara: u registar `D0` zapisuje duljinu stringa (32-bitnu) čiju adresu prima preko registra `A0`. Svi registri osim `D0` trebaju ostati nepromijenjeni.

Napišite funkciju analognu `void strcat(char *s1, char *s2, int size)`. Funkcija treba zalijepiti string `s2` na kraj stringa `s1`, pod uvjetom da je zbroj duljina stringova manji ili jednak vrijednosti `size`. Ukoliko nije, funkcija odmah izlazi. Parametri se primaju preko stoga, tako da je `size` stavljen prvi a `s1` zadnji. Funkcija nema povratnih parametara. Potrebno je počistiti stog i vratiti originalne vrijednosti registara. Za računanje duljine stringa obavezno zovite funkciju iz prvog dijela zadatka (čak i ako je niste napisali).

Koristeći prethodne funkcije, napišite program koji spaja 2 stringa, čije se adrese nalaze na lokaciji `$6000` (`s1`) i `$6004` (`s2`), a na adresi `$6008` se nalazi 32-bitni broj koji označava maksimalnu duljinu stringova (`size`).