

Zavod za elektroniku, mikroelektroniku,
računalne i inteligentne sustave

Arhitektura računala II

Međuispit, problemski dio (60% bodova)

1. (12 bodova) Razmotrimo izvedbu memorijskih modula i njihovo priključivanje u računalni sustav.

a) Prikažite izvedbu memorijskog modula kapaciteta 4 4-bitne riječi pomoću binarnih ćelija s 3 stanja. (BC3S). Napomena: Sa strane skicirajte koji ulazi/izlazi ćelije BC3S predstavljaju signale *data*, *select* i *R/W*.

b) Prikažite shemu spajanja triju takvih modula u računalni sustav sa 16-bitnom adresnom i 4-bitnom podatkovnom sabirnicom, uz pretpostavku da se oni javljaju u kontinuiranom adresnom prostoru s početnom adresom \$1F70.

2. (12 bodova) Pojednostavljeni model CISC procesora kojemu je pridodano 16-bitno kazalo stoga (SP), program smješten u memoriji na početnoj adresi \$300. Program je u memoriji predstavljen sljedećim nizom bajtova (vrijednosti koje slijede su u heksadekadskom obliku): B6, 03, 01, 87, 20, 00.

Poznati su operacijski kodovi instrukcija:

00 → NOP

01 → INC A (inkrement akumulatora)

02 → ADD *const* (Uvećanje akumulatora neposrednom 8-bitnom konstantom)

20 → JMP *offset* (Bezuvjetni skok na relativno zadanu adresu)

87 → CALL *addr* (Poziv potprograma na apsolutno zadanoj adresi, povratna adr. sprema se na stog)

B6 → LDA *addr* (Punjenje akumulatora podatkom na apsolutno zadanoj adresi)

Prikažite program u mnemoničkom obliku. Uz pretpostavku da je početni sadržaj kazala SP = \$1F00 nacrtajte stanje na sabirnici tijekom izvođenja ovog programa te označite fazu „PRIBAVI“ i „IZVRŠI“ pojedinih instrukcija. Odredite konačni sadržaj registara i izmjenjenih memorijskih lokacija nakon izvođenja programa (Nepoznate vrijednosti označite s odgovarajućim brojem X-eva, pri čemu X označava jednu heksadekadsku znamenku).

3. (12 bodova) Za model 8-instrukcijskog procesora (slika na poledini), napišite logičke jednažbe upravljačkih signala za fazu PRIBAVI i IZVRŠI instrukcije AND A.

4. (12 bodova) Promatramo program za sljedeću razdiobu instrukcija: memorijske instrukcije – 45%, aritmetičko-logičke instrukcije – 35%, instrukcije grananja – 20%. Poznato da CPI-jevi memorijskih, aritmetičkih te instrukcija grananja iznose 2,1 i 3. Pretpostavite da želimo izvršiti povećanje performansa programa za 10%.

a) Razmotrite mogućnost postizanja cilja sljedećim pristupima:

- izmjenom CPI-ja memorijskih instrukcija za faktor x_a ($CPI_{MEM}' = CPI_{MEM} * x_a$)
- izmjenom CPI-ja aritmetičko logičkih instrukcija za faktor x_b ($CPI_{ALU}' = CPI_{ALU} * x_b$)
- izmjenom radnog takta za x_c ($f' = x_c * f$)

U situacijama gdje je moguće cilj ostvariti, odredite odgovarajuće faktore x_a , x_b , x_c .

b) Razmotrite mogućnost ostvarivanja cilja poboljšanim algoritmom koji bi broj ukupan memorijskih instrukcija algoritma izmjenio za faktor x_d . Ako je cilj moguće ostvariti, odredite x_d .

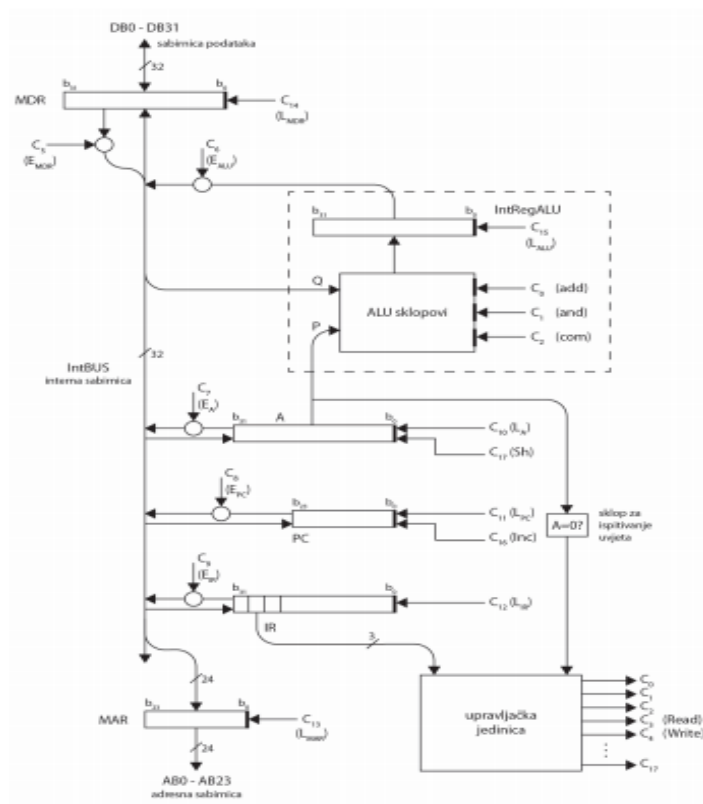
5. (12 bodova) Razmotrimo računalo s 32-bitnim adresama. Računalo izvodi glavni program koji poziva potprogram prema sljedećem kodu u programskom jeziku C.

```
int proc (int a){
    int c[2];
    c[0] = a - 1;
    c[1] = 2 * c[0];
    return c[0] + c[1];
}
```

```
int main(){
    proc(3);
}
```

Za prijenos parametara i smještaj lokalnih varijabli koristi se stog. Neka je početna vrijednost kazala stoga: SP = \$3FFE88, te neka se instrukcija za pozivanje potprograma nalazi na adresi \$385F20 (instrukcija zauzima 4B). Pretpostavite da varijable tipa int zauzimaju 32 bita. Skicirajte stanje stoga i odredite vrijednost kazala stoga:

- Prije pozivanja potprograma
- Unutar potprograma, neposredno prije povratka
- Nakon povratka iz potprograma
- Utvrđite što bi se dogodilo kad bi se u potprogramu, neposredno prije instrukcije *return*, dodala instrukcija $c[2] = c[2] - 4$.



Slika uz zadatak 3: organizacija osaminstrukcijskog procesora