

Priprema za drugu laboratorijsku vežbu

- Arhitektura i organizacija računara 2 -

1. Potrebno je dodati nov način adresiranja *shortimm*, koji predstavlja kratko 4-bitno neposredno adresiranje. Potrebno je *imm* izmestiti tako da izgled načina adresiranja bude kao u sledećoj tabeli.

adresiranje	IR ₂₃	IR ₂₂	IR ₂₁	IR ₂₀	IR ₁₉	IR ₁₈	IR ₁₇	IR ₁₆
<i>imm</i>	1	1	1	0	/			
<i>shortimm</i>	1	1	1	1	disp			

Instrukcije koje mogu da imaju ovakvo adresiranje su: XOR, OR, AND, SUB, ADD, LDB, LDW.

2. Potrebno je realizovati da instrukcije INTE i INTD ne reaguju na prekide.
3. Potrebno je da se pri ulasku u prekidnu rutinu na stek čuvaju akumulatori AB i AW (pored PC i PSW).
4. Potrebno je dodati novu instrukciju:

SWAPB – instrukcija koja treba da zameni argument i akumulator AB. U slučaju da se adresira neposredna vrednost, prijaviti grešku.

MIN – instrukcija koja pronalzi minimalni element u nizu. Bitovima IR_{23..21} je striktno definisano memorijsko adresiranje. Bitovima IR_{21..16} definiše se registar GPR u kome se nalazi dužina niza. U narednih IR_{15..0} definiše se početak niza u memoriji. Elementi su dužine 8 bita.

Za vežbu:

1. Dodati nov način adresiranja:
 - I. *regdirpostinc* – vrednost se nalazi u registru indeksiran vrednostima IR₂₀IR₁₉IR₁₈IR₁₇IR₁₆, nakon čega se inkrementira vrednost registra
 - II. *regdirpreinc* – registar indeksiran vrednostima IR₂₀IR₁₉IR₁₈IR₁₇IR₁₆ se inkrementira, nakon čega se vrednost čita iz tog registra
2. Potrebno je realizovati da instrukcije RTI, TRPD, TRPE ne reaguju na prekide.

3. Potrebno je realizovati da svi unutrašnji prekid (greška u kodu, greška u adresiranju, softverski zahtev za prekid, zahtev za prekid posle svake instrukcije) na stek stavljaju PC instrukcije u kojoj je došlo do prekida (a ne PC sledeće instrukcije).
4. Potrebno je dodati nove instrukcije:
 - I. MAX - instrukcija koja pronalzi maksimalni element u nizu. Bitovima $IR_{23..21}$ je striktno definisano memorijsko adresiranje. Bitovima $IR_{21..16}$ definiše se registar GPR u kome se nalazi dužina niza. U narednih $IR_{15..0}$ definiše se početak niza u memoriji. Elementi su dužine 8 bita.
 - II. STRLEN – instrukcija koja ima jedan argument koji predstavlja početak string-a u memoriji. Rezultat instrukcije treba da u akumulator AB upiše dužinu string-a. Karakteri string-a su 8-bitne veličine, a kraj string-a se definiše sa karakterom vrednosti 0.
 - III. CMPSTR - instrukcija koja upoređuje dva stringa. Bitovima $IR_{23..21}$ je striktno definisano memorijsko adresiranje. Bitovima $IR_{21..16}$ definiše se registar GPR u kome se nalazi adresa prvog stringa. U narednih $IR_{15..0}$ definiše se adresa drugog stringa. Rezultat ove instrukcije se upisuje u akumulator (-1 – prvi string je veći od drugog; 0 – stringovi su isti, 1 – drugi string je veći od prvog). Karakteri string-a su 8-bitne veličine, a kraj string-a se definiše sa karakterom vrednosti 0.