

MIKRORAČUNALA

VJEŽBA BR. 1 - MCU 8051 - Asembler

Cilj vježbe: Naučiti i uvježbati osnovne postupke u radu sa simulatorom mikrokontrolera. Uvježbati korištenje naredbi i pisanje jednostavnih programskih sekvenci koje izvršavaju aritmetičke operacije, logičke operacije i operacije za rad s bitovima -.

Proučite i riješite sljedeće zadatke. Pratite pri tome promjene u registrima procesora te na pristupima p1 i p3!

Uvod:

Otvoriti novi projekt izbornikom **Project – New Project...**
upisati ime projekta (npr. prvi zadatak) i pohraniti ga

Zadatak 1. Napisati program koji će zbrojiti hex. vrijednosti 2D i 33 uz pomoć registara A i B. Rezultat pohraniti u registar A.

```
mov    a, #2Dh      Opaska: Znak # uz hex. broj u naredbi označava da se radi o konstanti!!!
mov    b, #33h
add    a, b
end
```

Obratite pozornost da su heksadecimalne vrijednosti naznačene slovom h.

- **mov** je instrukcija za postavljanje određenih vrijednosti na neku adresu ili u neki registar.

- **add** je instrukcija za zbrajanje.

Zadatak 2. Napisati program koji će zbrojiti hex. vrijednosti 57 i 4B uz pomoć registara A i B. Rezultat pohraniti u registar A.

Zadatak 3. Napisati program koji će oduzeti (subb) hex. vrijednosti 7F i 3C uz pomoć registara A i B. Rezultat pohraniti u registar A.

4) Napisati program koji će pomnožiti (mul) hex. vrijednosti 11 i 3 uz pomoć registara A i B. Provjeriti dobiveni rezultat!

ZBRAJANJE I ODUZIMANJE

1) Zbrojite hex. vrijednosti 4Ah i C3h koristeći se registrima A i B!

```
mov    a, #4Ah
mov    b, #0C3h
add    a, b
end
```

Objasni dobiveni rezultat i stanje zastavica!

*Opaske:

- **ispred vrijednosti konstante pišemo nulu - 0, ako ona počinje slovom** kao u ovom primjeru - C3
- nakon izvršenja naredbe ADD postavljena je **zastavica prijenosa (CY)** jer je rezultat veći od maksimalne vrijednosti koju akumulator može spremati – FFh ili 255 decimalno
- postoje 3 vrste zastavica na koje naredba ADD utječe: **CY** (carry flag – zastavica prijenosa), **AC** (auxiliary flag – zastavica poluprijenosa) i **OV** (overflow flag – zastavica preliva)
- **CY** se postavlja ako rezultat premašuje FFh, u suprotnom se briše
- **AC** se postavlja ako postoji “carry” iz nižeg “polubajta” u viši “polubajt”, tj. iz trećeg bita u četvrti
- **OV** služi kad radimo s brojevima s predznakom (kada je rezultat izvan opsega +127 i –128)!

2) Oduzmite hex. vrijednost 64h od 80h, koristeći se registrima A i B!

```
mov a, #80h
mov b, #64h
subb a, b
end
```

Objasni dobiveni rezultat i stanje zastavica!

3) Zbrojite dekadске vrijednosti 49 i 24 koristeći se registrima A i B!!

Objasni dobiveni rezultat!

4) Oduzmite dekadsku vrijednost 10 od 55 koristeći se registrima A i B!!

Objasni dobiveni rezultat!

MNOŽENJE I DIJELJENJE

1) Pomnožite hex. vrijednosti 15 i F !

* **Opaske:**

- naredba **mul** može pomnožiti **samo sadržaje registara A i B**
- ako rezultat premašuje 255, postavlja se overflow flag; low byte rezultata je pohranjen u A, a High byte u B; s obzirom na to da rezultat ne može premašiti FFFF, carry flag se nikad ne postavlja.

Objasni dobiveni rezultat tj. sadržaje registara a i b!

2) Podijeliti dekadске vrijednosti 99 i 5!

```
mov a, #99
mov b, #05
div ab
end
```

LOGIČKE OPERACIJE

1) U akumulator spremi vrijednost 2Bh, nakon toga izvršiti operaciju OR (uz pomoć instrukcije ORL) nad akumulatorom i operandom 1Ah. Objasni dobiveni rezultat!

2) U akumulator spremi vrijednost C5h, nakon toga izvršiti operaciju AND (uz pomoć instrukcije ANL) nad akumulatorom i operandom F0h. Objasni dobiveni rezultat!

3)

OPERACIJE ZA RAD S BITOVIMA, OPERACIJE ZA BRISANJE SADRŽAJA I SL.

- 1) Spremiti vrijednosti C4h u akumulator, te akumulator obrisati instrukcijom CLR A!
- 2) Obriši 5. bit u registru P3 (P3.4). Prati promjenu na odgovarajućem pristupu!
- 3) Postavi 4. bit u registru P1 u logičku "1" (P1.3), zatim ga komplementiraj. Prati promjenu na odgovarajućem pristupu!
- 4) Postavi bit C u "1" te bit P1.5 u "0". Obavi logičku operaciju I između bita C i bita P1.5. Prati promjenu na odgovarajućem pristupu!
- 5) Postavi bit C u "0" te bit P3.2 u "1". Obavi logičku operaciju ILI između bita C i bita P3.2!

Napomena: Zadatke 4. i 5. riješiti tako da rezultat bude u:

- a) bitu C
- b) bitu pristupa (P1.5 tj. P3.2)!

DODATAK

- 1) Zbrojiti dekadске brojeve 58 i 19 uz pomoć registara A i B! Provjeri rezultat i u dekadskom i u heksadecimalnom sustavu!
- 2) Koristeći akumulator i registar b, pomnoži dekadске brojeve 15 i 4. Dobiveni rezultat podijeli sa 5. Provjeri dobivene rezultate!