

MIKRORAČUNALA

VJEŽBA BR.2: Operacije premještanja podataka i operacije skoka u programu

Cilj vježbe: Naučiti i uvježbati osnovne postupke u radu sa simulatorom mikrokontrolera. Uvježbati korištenje naredbi i pisanje jednostavnih programskih sekvenci koje izvršavaju operacije premještanja podataka i operacije skoka u programu.

Upute

* za sve zadatke napiši program zadatka, provjeru rezultata

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

Uvodni zadatak

1. U akumulator spremi vrijednost 7h, a na lokaciju 13h upisati konstantu 1h. Uvećati sadržaj lokacije 13h za 1, a sadržaj akumulatora smanjiti za 1. Instrukcijom skoka (skok ako je akumulator 0) priredi na programsku lokaciju sa oznakom **primjer!**

```
mov a, #07h
mov 13h, #01h
```

primjer:

```
inc 13h
dec a
jnz primjer
end
```

- **jnz** : instrukcija skoka u programu na određenu oznaku, ako sadržaj akumulatora nije 0 (jump if not zero)

Pitanje: Koliko će se puta izvršiti ovaj programski odsječak te koliki će biti sadržaj lokacije 13h po završetku istoga?

INSTRUKCIJE ZA PRIJENOS PODATAKA

- prenošenje podataka vrši se uz pomoć instrukcije **mov**
- opći oblik instrukcije mov: **mov <x>, <y>**
x ; registar ili memorijska lokacija na koju se sprema podatak
y ; podatak ili memorijska lokacija s koje se podatak sprema u <x>

1. Izvedi sljedeće:

```
mov r5, #0C3h      ; r5 = ?
mov a, r5          ; a = ?
clr c              ; c = ?
mov p1.4, c        ; p1.4 = ?
mov p1, #2Fh       ; p1 = ?
mov a, p1          ; a = ?
mov a, #25h        ; a = ?
mov p3, a          ; p3 = ?
setb p3.5          ; p3.5 = ?
mov c, p3.5        ; c = ?
end
```

2. Izvedi sljedeće:

```
clr p3.2           ; p3.2 = ?
setb p3.4          ; p3.4 = ?
mov c, P3.4        ; c = ?
anl c, P3.2        ; c = ?
mov P3.7, c        ; p3.7 = ?
end
```

INSTRUKCIJE ZA SKOKOVE U PROGRAMU

1. Koristeći akumulator, lokaciju 2Ch i lokaciju 2Dh zbrojiti brojeve od 1 do 6!

```
mov a, #00h
mov 2Dh, #01h
mov 2Ch, #06h
```

opet:

```
add a, 2Dh
inc 2Dh
djnz 2Ch, opet
end
```

Pitanja: a) Koliko se puta izvršava ovaj programski odsječak?
b) Objasni djelovanje instrukcije: djnz 2Ch, opet !

2. Izvedi sljedeći programski odsječak. Prati promjene na pristupu p1.3!

```
clr p1.3
mov r2, #5
delay1:
djnz r2, delay1
setb p1.3
mov r2, #5
delay2:
djnz r2, delay2
end
```

Pitanja: a) Čemu služi konstanta koju upisujemo u registar r2?
b) Koliko se puta izvršava ovaj programski odsječak?

DODATAK

1. Izvedi sljedeći programski odsječak. Objasni promjene na pristupu p3.2!

```
clr p3.2
mov r5, #7
petlja:
cpl p3.2
djnz r5, petlja
end
```

Pitanja: a) Čemu služi konstanta koju upisujemo u registar r5?
b) Koliko se puta izvršava ovaj programski odsječak?