

### **MOVE (#),B –op. Kod #03**

Ova makronaredba puni registar B sadržajem navedene memorijske lokacije u okruglim zagradama.

- jedan bajt nakon operacijskog koda zapisana je adresa memorijske lokacije
- sadržaj programskog brojila se sprema u MAR, programsko brojilo se uvećava za 1, obavlja se čitanje sadržaja memorijske lokacije i preko registra MAR prenosi u akumulator
- vrijednost se potom iz akumulatora pohranjuje u registar B

030	JCR 1	HCZ, FF1	LMI/1	R9/I	čitanje	PC->MAR, (MAR)->M, PC+1->PC
031	JCR 2	HCZ, FF0	LMM/1	AC/II	čitanje	M->MAR, M->AC
032	JZR 0	HCZ, FF1	SDR/2	R1/I	Nista	AC->B

### **ADD #,A**

-Pribrajanje konstante registru A. Postaviti zastavice u skladu s rezultatom (C,Z)

-Operacijski kod: #04

-pročitaj konstantu I spremi ju u AC

-zbroji AC + A I spremi u A

-postavi zastavice

040	JCR 1	HCZ/FF1	LMI/1	R9/I	Čitanje	PC-> MAR; MAR->M; PC++
041	JCR 2	HCZ/FF0	ACM/0	AC/II	Nista	M->AC
042	JCR 3	STC/FF0	ADR/3	R0/I	Nista	AC+A->A
043	JCR 4	STZ/FF0	TZR/5	R0/I	Nista	Zero test
044	JCR 5	HCZ/FFZ	CSR/2	T/I	Nista	Komplement zero
045	JZR	STZ/FF0	TZR/5	T/I	Nista	Zero test

ASR A –Aritmeticki posmak sadržaja registra A udesno za 1(bit predznaka ostaje očuvan)

-op. Kod #05

-prvo zbrajamo A sa samim sobom radi prijenosa(viši bit)

**-Nakon toga shiftamo udesno te spremamo nazad u registar A**

050	JCR 1	HCZ, FF0	ILR/0	R0/I	nista	<b>A→AC</b>
051	JCR 2	STC/FF0	ADR/3	R0/I	nista	<b>AC+A→AC</b>
052	JCR 3	HCZ, FF0	ILR/0	R0/I	nista	<b>A→AC</b>
053	JCR 4	HCZ, FFC	SRA/0	AC/III	nista	<b>ATL→A,LI→ATH ATH→ATL</b>
054	JZR 0	HCZ, FF1	SDR/2	R0/I	Nista	<b>AC-&gt;A</b>

### **PUSH A-op kod #06**

Makronaredba pohranjuje sadržaj registra A na stog.

- adresa stoga pohranjuje se u MAR i vrijednost SP-a se povećava za jedan
- zatim se vrijednost registra B smješta u akumulator i obavlja se spremanje vrijednosti u memoriju na adresu memorijske lokacije koja je zapisana u MAR-u

060	JCR 1	HCZ, FF1	LMI/1	R8/I	Nista	SP->MAR, SP+1->SP
061	JZR 0	HCZ, FF0	ILR/0	R0/I	pisanje	A+0->A, AC AC-> (MAR)

**Primjer programa:**

**MOVE (#30), B**

**PUSH B**

**MOVE (#40), A**

**ASR A**

**PUSH A**