

## 8.Hafta

26 Kasım 2020 Perşembe 19:07

### Örnek 1

P1 giriş portundan okunan değer 0 olduğunda duran ve bunu P0 portuna yazdıran program:

- JZ komutu ile Akümülot'deki değerin 0 olup olmadığını kontrol ediyoruz.
- \$ işareti satır başı anlamına geliyor.
- CJNE komutu istenilen değeri karşılamıyorsa dallanır, karşılıyorsa alt satıra iner ve bu yüzden dallanacak kısmın bu alt satırda olmaması gerekir.
- Eğer okunacak değer farklıysa 8.satırdaki Hexi değiştirebiliriz.

```
1 $MOD51
2
3 cseg
4     org 0h
5
6     MOV P1,0FFH
7 OKU:  MOV A,P1
8       CJNE A,#0H,YAZ
9       JMP DUR
10 YAZ:  MOV P0,A
11       JMP OKU
12 DUR:  JMP $ ;JMP DUR
13       END
14
```

- Pin değerini okumak için ACC.3 kullanırız. Akümülotör'deki 3 nolu bit anlamındadır.
- JNC komutundaki NC'nin açılımı "Not Carry" ve anlamı carry yoksa yani sıfırsa'dır.

```
MOV C,ACC.3
JC DUR
```

### Örnek 2

- ORG 0H, Ephrom'a bu adresten itibaren kaydet anlamındadır.
- @ işareti ile dolaylı adresleme yaparız. Koddaki anlamı R0=0'dı bu yüzden 0 adresindeki bilgiyi P0'a gönder
- INC bir attırır.

```
1 $MOD51
2
3 cseg
4     org 0h ;0H TEN 7FH E KADAR OLAN BÖLGE (RAM) P0 PORTUNDAN DISARI ATILACAK
5
6     MOV R0,#0H ;DATALARIN OKUNACAGI BASLANGIÇ ADRESİ
7 GONDER: MOV P0,@R0
8         INC R0
9         CJNE R0,#80H,GONDER
10        JMP $
11        END
12
```

- 7FH'den 0'a kadar olan bölge için Başlagıç adresini değiştiriyoruz ve geri geri gideceğinden azaltma işleme için DEC kullanabiliriz ama DJNZ kullanarak karşılaştırma işleminide yaparız.

```
MOV R0,#7FH
GONDER: MOV P0,@R0
        DJNZ R0,GONDER
```

### Örnek 3

```
1 $MOD51
2
3 cseg
4     org 0h ;40H TEN ITIBAREN 10 BYTE BILGI OKUNUP 80H TEN ITIBAREN KOPYALANSIN
5
6     MOV R0,#40H ;DATALARIN OKUNACAGI KAYNAK ADRESIN BASLANGICI
7     MOV R1,#80H ;DATALARIN YAZILACAGI HEDEF ADRESIN BASLANGICI
8
9 DEVAM: MOV A,@R0
10        MOV @R1,A
11
12        INC R0
13        INC R1
14
15        CJNE R0,#4AH,DEVAM
16        JMP $
17        END
```

### Örnek 3

- MOVC; Ephrom üzerinden okuma yapmak için kullanılır.
- @A+DPTR aslında A ile DPTR'yi önce toplar, bu sayıyı adreslicecek. Nedeni DPTR'yi yalnız şekilde dolaylı adresleme yapamıyoruz. Bu sebeple Akümülotörü CLR ediyoruz.
- DB açılımı "Define Byte" yani Byte, 8bit olarak tanımla. Kodda 1001'de K, 1002'de A harfi olarak tek

tırnak işareti içerisindekileri sırayla yazar. Virgül ile ayırıp 0 koyduğumu 1007 oluyor.

```
1 $MOD51
2
3 cseg
4     org 0h ;EPROM UN 1000H ADRESİNDEKİ BİLGİNİN DAHİLİ RAM DE 50H ADRESİNDEN İTİBAREN KOPYALANMASI
5
6     MOV     DPTR,#MYDATA ;KAYNAK BAŞLANGIÇ ADRESİ
7     ;HEDEF BAŞLANGIÇ ADRESİ
8 L1:    CLR A
9         MOVC A,@A+DPTR
10        JZ L2
11        MOV P0,A
12
13        INC DPTR
14        INC R0
15        JMP L1
16
17 L2:    JMP $
18
19     ORG 1000H
20     MYDATA: DB 'KARABÜK',0
21     END
```

### Örnek 3

```
1 $MOD51
2
3 cseg
4     org 0h ;P1 PORTUNDAN OKUNAN SAYININ KARESİNİ P2 PORTUNDAN GÖNDEREN PROGRAM (OKUNAN SAYILAR 0.1.2.3.4.5.6.7.8.9)
5
6     MOV     DPTR,#MYDATA ;KAYNAK BAŞLANGIÇ ADRESİ
7
8     MOV P1,#0FFH
9
10 L1:    MOV A,P1
11        MOV B,A
12        MUL AB
13
14        MOV P2,A
15
16        JMP L1
17
18 L2:    JMP $
19
20     ORG 1000H
21     MYDATA: DB 0,1,4,9,16,25,36,49,64,81
22     END
```