

MSC-1 Ailesi

Assembler Programlama Dili ve Program Geliřtirme

Kasım 2008
Yrd. Doç. Dr. Mustafa Engin

Program Geliştirme

- ◆ Problem Tanımlama
- ◆ Bağlantı Şekli
- ◆ Algoritma
- ◆ Akış Diyagramı
- ◆ Kaynak Program Yazma
- ◆ Assembly
- ◆ Simulatör
- ◆ Mikroişlemciye Yükleme ve Çalıştırma

Problem Tanımlama

- ◆ Problem kullanılacak denetleyiciye veya işlemciye göre teknik bir dille yeniden tanımlanır.
- ◆ Derste verilen sorular genellikle tanımlanmış problemler olduğu için yeniden tanımlanmaz.

Bağlantı Şekli

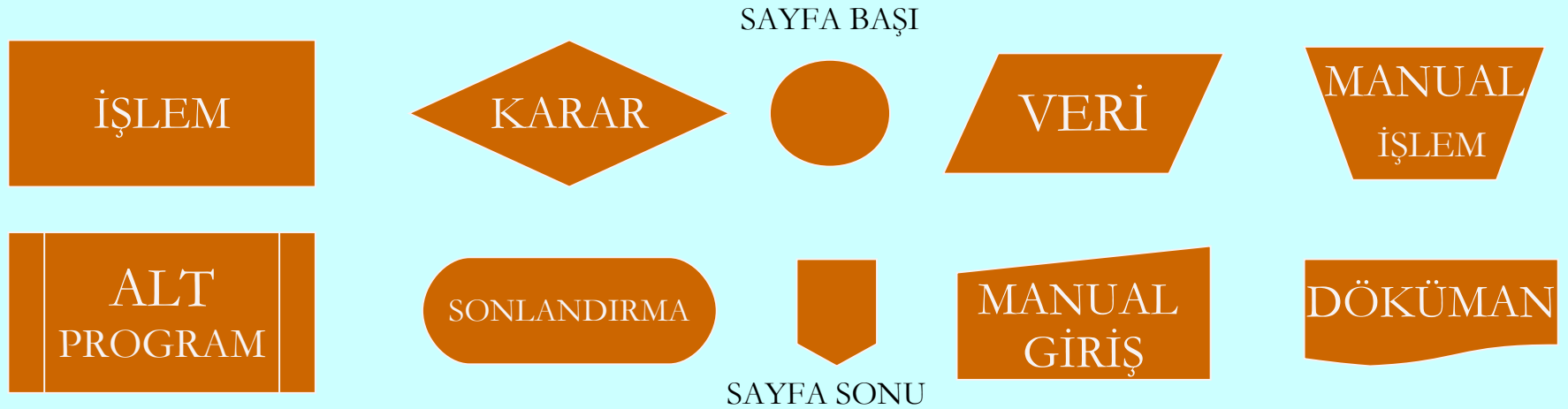
- ◆ Eğer tanımlanan problem yeni bir bağlantı gerekiyorsa açık veya blok şema çizilir.
- ◆ Özellikle bağlantı yapılacak port isimleri mutlaka yazılır.
- ◆ Bağlanacak elemanların hatlarının isimleri yazılmalıdır.

Algoritma

- ◆ Çözüm yönteminin konuşma dili ile maddeler halinde tanımlanmasıdır.
- ◆ Algoritma problemin çözümünü temel oluşturur.
- ◆ Bağlantı şeklini desteklemelidir.
- ◆ Yalın bir dil kullanılarak yazılmalıdır.
- ◆ Bir problemin birden fazla çözüm algoritması olabilir.

Akış Diyagramı

- ◆ Algoritmanın şekilsel olarak gösterimidir.
- ◆ Her yapılan işlem bir simge ile gösterilir.
- ◆ Programın işleyiş sırasını anlamayı kolaylaştırır.



Kaynak Program

;Programın başlığı, yazar kişi, tarih

;program hakkında açıklama

Org 0

;

isim:

Mov A,#25

;A=25

.....

;.....

TKR:

.....

;.....

Sjmp \$

;son

;Programın başlığı, yazar kişi, tarih

;program hakkında açıklama

Simulator

- ◆ Yazılan program assebmly edilir.
- ◆ Simulatöre yüklenerek programın çalışması adım modunda denetlenir.
- ◆ Hata var ise algoritmaya geri dönülür.
- ◆ Hatalar tamamen yok edilinceye kadar devam edilir

Mikroişlemciye Yükleme ve Çalıştırma

- ◆ Hataları giderilen program programlayıcı kullanılarak mikroişlemciye yüklenir.
- ◆ Gerekli bağlantılar kart üzerinde ve çevresinde yapılır.
- ◆ Devreye güç verilerek çalıştırılır ve sonuçlar gözlemlenerek doğruluğu belirlenir.
- ◆ Hata var ise algoritmaya geri dönülür.
- ◆ Hata yok ise tasarlanan cihaz üretime hazırdır.

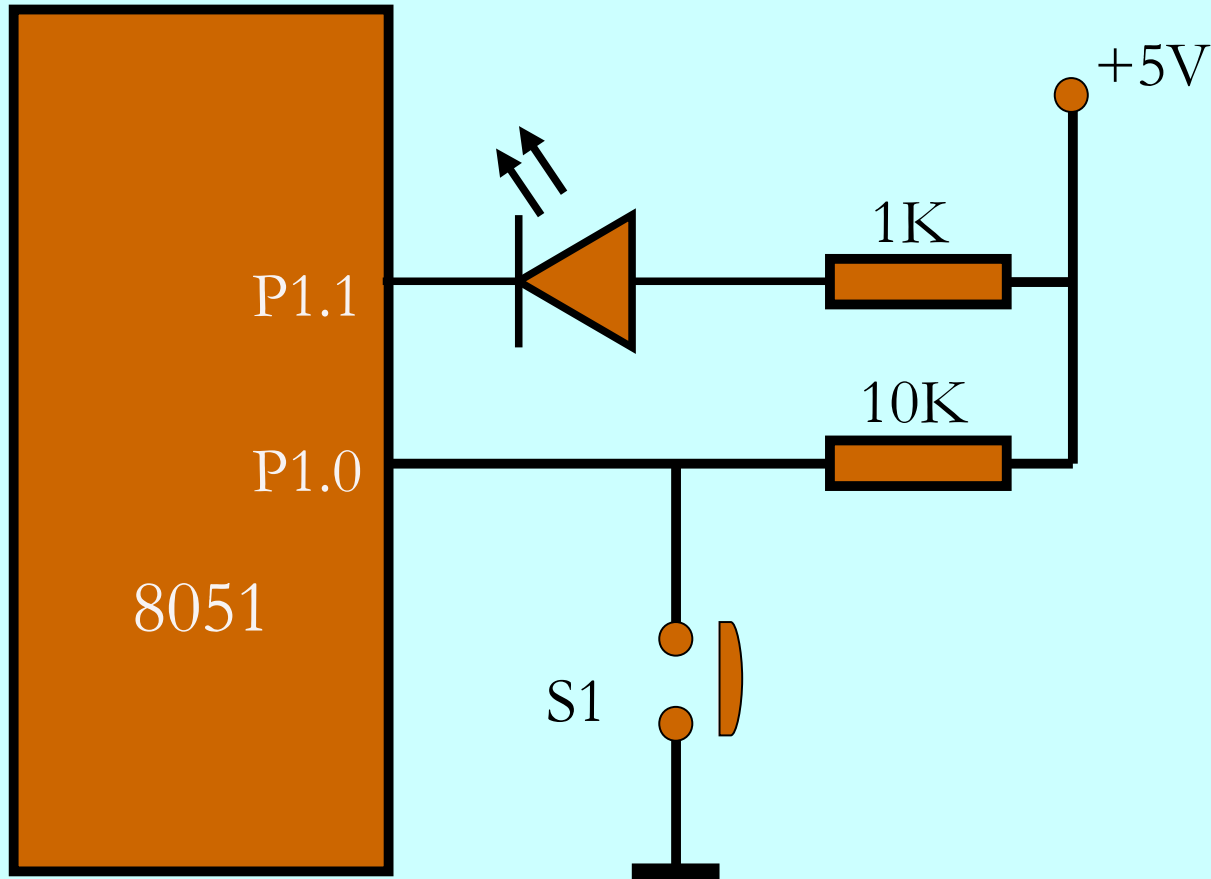
Örnek

- ◆ Bir butona basıldığında LED'i yakan ikinci basmada ise söndüren programı yazın, sistemi kurup çalıştırın.

1. Problemi Yeniden Tanımlama:

8051'in port hattına bir buton ve bir diğer hattına LED bağlıdır. Butona her basıldığında LED'in durumunu tersleyen programı yazın.

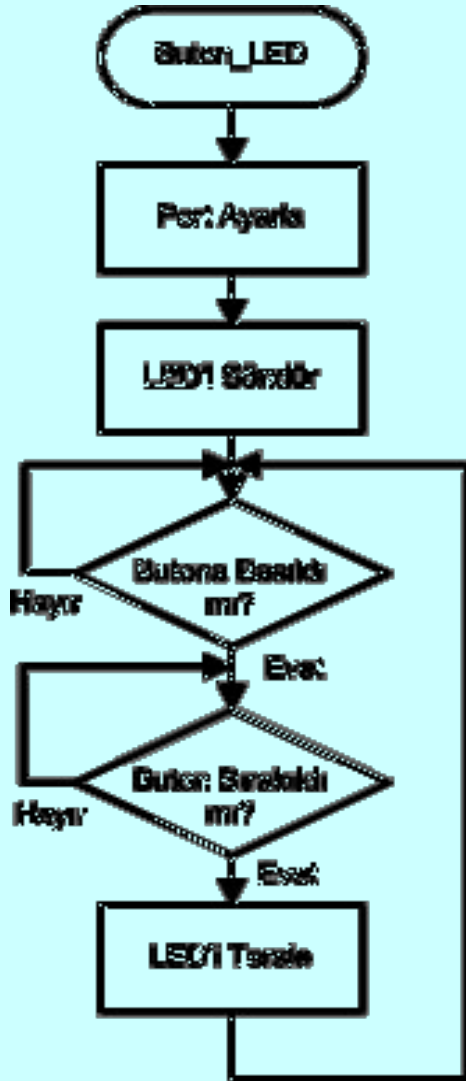
2. Bağlantı Şekli



3. Algoritma

1. Buton için port hattını giriş yap.
2. LED için port hattını çıkış yap
3. LED'i başlangıç için söndür.
4. Butona basılana kadar bekle.
5. Buton bırakılana kadar bekle.
6. LED'i tersle.
7. Adım 4'e dön.

4. Akış Diyagramı



5. Kaynak Program

BUT_LED:

```
Mov P1,#0FFh ;P1.0 giriş, P1.1,  
;çıkış, LED sönük
```

BUT: Jb P1.0,\$;Butona basılana kadar bekle

Jnb P1.0,\$;Buton bırakılana kadar bekle

Cpl P1.1 ;LED'i tersle

Sjmp BUT ;Sürekli yap

Örnek 1 16 Bit Değişkene 8 Bit Değişken Toplama.

Bu alt program A'yı DPTR'ye toplar, taşma var ise elde bayrağı kurulur. İşletilmeden önce DPTR'ye ve A'ya bir değer girilmelidir.

İşlem sonunda; $DPTR = DPTR + A$

Etkilenen yazaçlar; PSW.CY, DPTR

topla16b:

Push Acc ; A'yı yedekle.

Add A,DPL ; A'ya DPL topla.

Mov DPL,A ; Sonucu DPL'ye yaz.

Mov A,DPH ; DPH'yi al.

Addc A,#00h ; Elde var ise yüksek kısım bir artırılır.

Mov DPH,A ; DPH yaz.

Pop Acc ; A'yı yedekten al.

Ret ; Ana programa geri dön.

ÖRNEK 2 16 Bit Değişkenden 8 Bit Değişken Çıkarma.

;Bu alt program DPTR yazacından Acc'yi çıkarır. Sonuç negatif ise elde
;bayrağını kurar. İşletilmeden önce DTPR'ye ve Acc'ye bir değer girilmelidir.
;İşlem sonunda DPTR=DPTR-Acc
;Etkilenen yazaçlar; PSW.CY, DPTR

cikar16b:

Push Acc ; A'yı yedekle.
Clr c ; Eldeyi temizle
Xch a,DPL ; A ile DPL'yi yer değiştir.
Subb a,DPL ; Çıkar.
Mov DPL,a ; DPL'yi yaz
Mov a,DPH ; DPH'yi al.
Subb a,#00h ;Borç var ise YD kısmı bir azalt.
Mov DPH,a ; DPH'yi yaz.
Pop Acc ; A'yı yedekten al
Ret ; Ana programa geri dön

ÖRNEK 3 16 Bitlik Çıkarma

;Bu program 16 bit sayıdan 16 bit sayıyı çıkarır.

;R7R6-R5R4=R3R2

Cıkar16bit:

<i>MOV A,R6</i>	<i>;1. deęişkenin düşük kısmını A'ya al</i>
<i>CLR C</i>	<i>;Eldeyi temizle</i>
<i>SUBB A,R4</i>	<i>;2. deęişkenin düşük kısmını A'dan çıkar</i>
<i>MOV R2,A</i>	<i>;Sonucun düşük kısmını yaz</i>
<i>MOV A,R7</i>	<i>;1. deęişkenin yüksek kısmını A'ya al</i>
<i>SUBB A,R5</i>	<i>;2. deęişkenin yüksek kısmını A'dan çıkar</i>
<i>MOV R3,A</i>	<i>;Sonucun yüksek kısmını yaz</i>
<i>RET</i>	<i>;Sonucu R2 ve R3'te ana programa götür.</i>

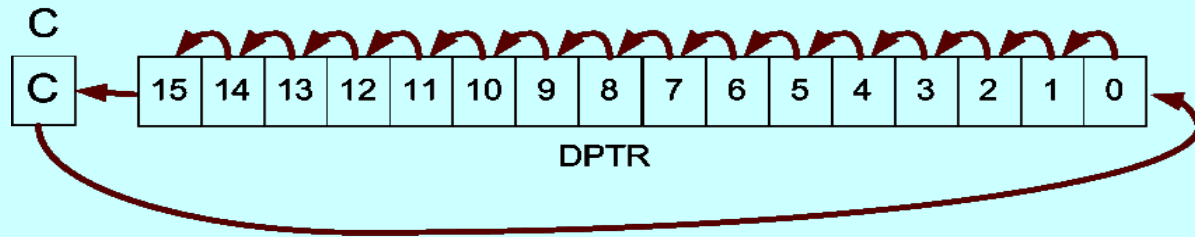
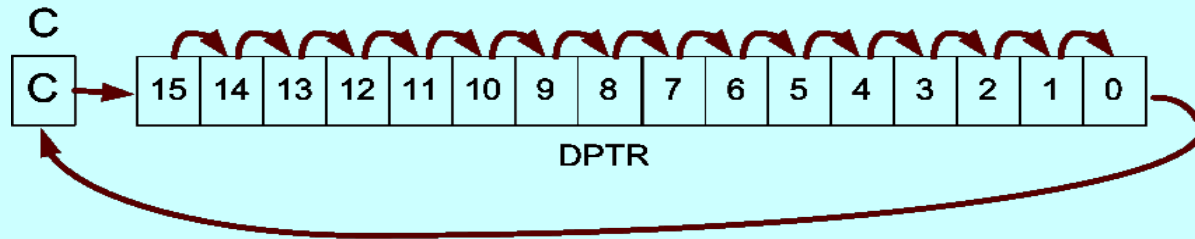
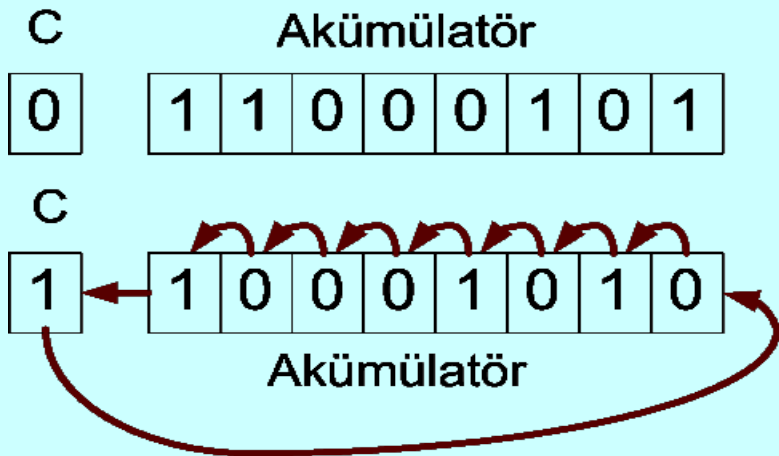
ÖRNEK 4 DPTR Yazacını Bir Bit Sağa Öteleme.

;Bu alt program DPTR yazacının içeriğini aritmetik bir bit sağa öteler. İşletilmeden önce DPTR içerisine bir değer girilmeli.

Dptrsag:

```
Push Acc           ; A'yı yedekle.  
Clr C              ; Eldeyi temizle.  
Mov A,DPH         ; DPTR'nin yüksek kısmını al.  
Rrc A             ; Sağa ötele.  
Mov DPH,A         ; DPH'ye yaz.  
Mov A,DPL         ; DPTR'in düşük kısmını al.  
Rrc A             ; Sağa ötele.  
Mov DPL,A         ; DPL'yi yaz.  
Pop Acc           ; Acc'yi tedekten al.  
Ret              ; Ana programa dön.
```

RLC A, RRC A



Örnek 5 DPTR Yazacını Bir Bit Sola Öteleme.

;Bu alt program DPTR yazacının içeriğini bir bit sola öteler.

;DPTR * 2 = DPTR

Dptrsol:

<i>Push Acc</i>	<i>; Acc'yi yedekle.</i>
<i>Clr C</i>	<i>; Eldeyi temizle.</i>
<i>Mov A,DPL</i>	<i>; DPTR'nin düşük kısmını al.</i>
<i>Rlc A</i>	<i>; Sola ötele.</i>
<i>Mov DPL,A</i>	<i>; DPL'ye yaz.</i>
<i>Mov A,DPH</i>	<i>; DPTR'nin yüksek kısmını al.</i>
<i>Rlc A</i>	<i>; Sola ötele.</i>
<i>Mov DPH,A</i>	<i>; DPH'ye yaz.</i>
<i>Pop Acc</i>	<i>;Yedekten al</i>
<i>Ret</i>	<i>; Ana programa dön.</i>

Örnek 6 DPTR'nin 10 Katını Alma.

;Bu alt program DPTR'nin 10 katını alır. DPTR'ye işlem öncesi bir 0-6553 arası bir değer yazılmalıdır. İşlem sonunda DPTR=DPTR * 10, etkilenen yazaçlar PSW,DPTR

DPTRX10:

<i>Push B</i>	<i>; B'yi yedekle.</i>
<i>Push Acc</i>	<i>; A'yı yedekle</i>
<i>Mov A,DPL</i>	<i>; Düşük kısmı al</i>
<i>Mov B,#10</i>	<i>; Çarpanı B'ye yaz</i>
<i>Mul AB</i>	<i>; BA = DPL * 10</i>
<i>Mov DPL,A</i>	<i>;Sonucun düşük kısmını DPL yaz</i>
<i>Push B</i>	<i>;Sonucun yüksek kısmını yedekle</i>
<i>Mov A,DPH</i>	<i>;Yüksek kısmını al</i>
<i>Mov B,#10</i>	<i>;Çarpanı B'ye yaz</i>
<i>Mul AB</i>	<i>;BA = DPH * 10</i>
<i>Pop B</i>	<i>;B yedekten al</i>
<i>Add A,B</i>	<i>;Düşük kısımdan oluşan ile topla</i>
<i>Mov DPH,A</i>	<i>;Sonucun yüksek kısmını DPH yaz</i>
<i>Pop Acc</i>	<i>;A'yı yedekten al</i>
<i>Pop b</i>	<i>;B'yi yedekten al</i>
<i>ret</i>	<i>;Ana programa dön</i>

Örnek 7 DPTR'yi 4 Bit Sağa Ötele

;Bu Alt Program Dptr'yi 4 Bit Sağa öter. Etkilenen Yazacılar Psw,Dptr, DPTR'yi 16'ya böler.

Dptr4sag:

```
    Push Acc          ; A'yı Yedekle
    Push 0            ; R0'i Yedekle
    Mov R0,#4         ; Öteleme Sayacını Ayarla
L1:  Mov A,DPH        ; Yüksek Kısmı Al
     Clr C
     Rrc A            ; Sağa ötele
     Mov DPH,A       ; DPH'ye Yaz
     Mov A,DPL        ; Dptr'nin düşük Kısmını Al
     Rrc A            ; Sağa ötele
     Mov DPL,A
     Djnz R0,L1       ; Sayaç=0 olana kadar ötele
     Pop 0            ; Yedekten Al
     Pop Acc          ; Yedekten Al
     Ret              ; Ana Programa Geri Dön
```

Örnek 8 Dptr 4 Bit Sola Öteleme

;Bu Alt Program Dptr Yazacını 4 Bit Sola Öteler, Düşük Değerli Bitlere Sıfır Yerleştirir.

;Etkilenen Yazacılar; Psw.Cy, Dptr

Dptr4sol:

```
    Push Acc ; A'yı Yedekle
    Push 0      ; R0'ı Yedekle
    Mov  R0,#4   ; Öteleme Sayacını Kur
L2: Clr  C       ; Öteleme İçin Sıfırla
    Mov  A,Dpl   ; Dptr'nin Düşük Kısmını Al
    Rlc  A       ; Sola Ötele
    Mov  Dpl,A   ; Dpl'yi Yaz
    Mov  A,Dph   ; Dptr'nin Yüksek Kısmını Al
    Rlc  A       ; Sola Ötele
    Mov  Dph,A   ; Dph'yi Yaz
    Djnz R0,L2  ; Sayaç Sıfır Olana Kadar Devam Et
    Pop  0       ; Yedekten Al
    Pop  Acc     ; Yedekten Al
    Ret                ; Ana Programa Geri Dön
```

Örnek 9 AND Maskeleye ve Tablo Okuma

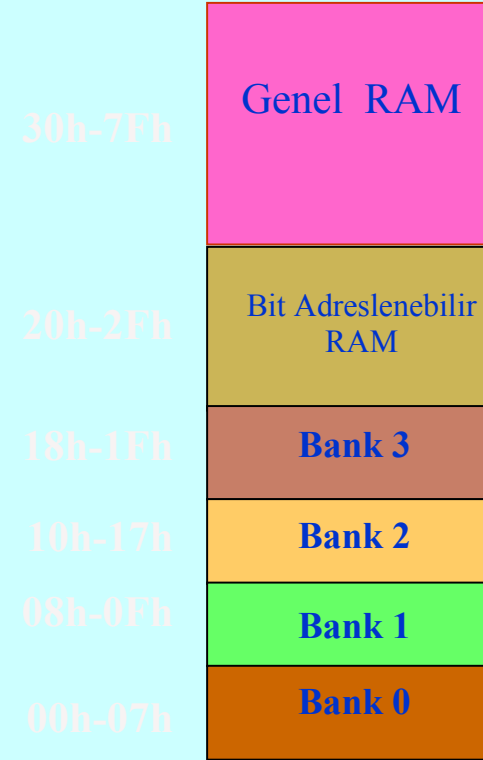
P1'den okunan 0 ile F arası sayıların karesini başvuru tablosu kullanarak alan ve P2'ye yazan programı yazın

Kare_al:

```
Mov DPTR, #TABLO ; Tablonun başlangıç adresi
Mov P1, #0FFH ; P1'i giriş yap
TKR: Mov A, P1 ; Sayıyı P1'den oku
Anl A, #0Fh ; Yüksek değerli 4 biti sıfırla
Movc A, @A+DPTR ; Sayının karesini tablodan al
Mov P2, A ; Sonucu P2'ye yaz
Sjmp TKR ; Sürekli yap
ORG 300H ; Tablo 0300H'dan başlar
TABLO: DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225
```

8051'de Yığın Kullanımı

- ◆ Yığını **SP** (stack pointer) yazacı ile erişebiliriz.
- ◆ Yığın işaretleyici SP 8051'de 8 bit genişliğindedir, bunun anlamı bu yazaç içeriği 00'dan FFH kadar değerleri içerebilir.
- ◆ 8051 çalıştırıldığında, SP yazacının içeriği 07'dir.



Yığın reset sonrası buradan başlar