

# Mikroişlemcili Sistemler ve Laboratuvarı

## Assembly ve Adresleme Türleri

Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü  
Yrd.Doç.Dr. Murat İSKEFİYELİ

### Amaçlar

- Assembly dilinin genel özelliklerini tanımak
  - 8051 mikrodenetleyicisinin adresleme türlerini kavramak
  - 8051 komut setinin yapısını kavramak
  - Assembly dilinde örnekler yapabilmek
- Bu sunumdaki şekiller “C ile 8051 Mikrodenetleyici Uygulamaları, A.T.Özcerit, M.Çakıroğlu, C.Bayılmış, Papatya Yayınları” kitabından alınmıştır.



## Assembly Dili ve Özellikleri

- Her bir komut, gerçekleştirdiği işleve karşılık gelen İngilizce kelimenin ya da cümlelerin kısaltması ile ifade edilir.
- (MOV =MOVE ve DJNZ=Decrement Jump Not Zero)
- Her bir kısaltma, makine dilindeki farklı bit dizisine karşılık gelmektedir.
- MOV R1, #61h ; R1 kaydedicisine 61h değerini yükle
- Assembly dilini kullanarak uygulama geliştirmek, makine diline (ikili sayı sistemine) göre daha kolaydır. C, Basic, Pascal vb. gibi yüksek seviyeli dillerde program yazmaya nazaran daha zordur.



## Assembly Dili ve Özellikleri

### Asembly Dilinin Avantajları

- Donanım hakkında daha fazla detay bilmeyi gerektirir. Bu dezavantaj gibi görölse de aslında kullanıcıya önemli bilgi birikimi sağlamaktadır.
- Özel donanım ihtiyaçları üzerinde daha fazla kontrol sağlar.
- Yüksek seviyeli dillere göre daha küçük, daha az yer kaplayan ve daha hızlı icra edilebilir kodlar üretilebilir.



# Program Formatı

Etiket	İşlem Kodu (Komut-Opkod)	İşlenen (Operand)	Açıklama
Tekrar:	MOV	R1, #61h	;R1 kaydecisine 61h değerini ata

## Etiket alanı

- Komut satırının ilk bilgisidir ve sembolik isimlerden oluşur.
- Program içerisindeki belirli işlevlerin gerçekleştiği bölümlerin başlangıcını göstermek amacı ile kullanılır.
- Program içerisinde istenilen kısma kolaylıkla dallanılmasını sağlar.
- Etiket ismi olarak mikroişlemci komut setinde tanımlı olan bir komut ismi verilemez.
- Etiket bir harf ile başlamak zorundadır.

# Program Formatı

## Komut

- Kısaltma(mnemonik) olarak da adlandırılan, komut seti içerisinde mikroişlemcinin belirli bir işi yapmasını sağlayan tanımlanmış sembollerdir.
- Komut alanına etiketten sonra 1 boşluk ya da sekme (tab) ile girilir.

## Operand

- Bu alan, işlemciye işlenecek veriyi ya da verinin nerede olduğunu gösterir.
- Tek başına bir anlam ifade etmez.
- Genelde komutun etki edeceği **hedef ve kaynak** bilgisini içerir. Hedef ve kaynak bilgisi birbirinden virgül (,) ile ayrılır.



# Program Formatı

## Assembly dilinde sayı sistemlerinin kullanımı

Ön Takı	Son Takı	Anlamı	Örnek
(Boşluk)	D	Onlu sayı (decimal)*	55 – 55D
%	B	İkili sayı (binary)	%01010101 – 01010101B
@	O	Sekizli sayı (octal)	@33 – 33O
\$	H	Onaltılık sayı (hexadecimal)	\$FB – FBH

## Açıklama Satırı

- ☐ Assembly dili (;) ile başlayan satırları açıklama satırı olarak kabul eder.
- ☐ Bu satırları yorumlamaz ve makine kodu üretmez.
- ☐ Yazılan uygulamanın anlaşılabilirliğini artırır

# Yönergeler

## ORG

- Kod bellek içerisinde programın başlangıç adresini belirtmek için kullanılan adres konumlandırma talimatıdır.
- ORG 'Adres' şeklinde kullanılır.
- Bir program içerisinde birden fazla ORG komutu kullanılabilir.

Talimat	Açıklama
ORG 0000h ;program 0000h adresinden başlasın PC=0000	
ORG 0030h ;program 0030h adresinden başlasın PC=0030	

# Yönergeler

## END

- Programın bitiğini gösteren talimattır.

## DB (Define Bayt)

- Kod bellek içerisinde sayı ve kelime (string) dizilerinin tanımlanmasını sağlar.

İsim	DB	ifadeler	Açıklama
Max_sayi	DB	255	;tek bir değişken tanımlanması
Tablo	DB	0, 5, 4, 3, -10	;dizi olarak tanımlama
Yaz	DB	'8051 ogreniyorum'	;string olarak tanımlama

# Yönergeler

## EQU

- EQU (Equal = eşittir) bir sayısal değerin istenilen sembol adına atanması işlemini gerçekleştirir
- Bu tanımlama program içerisinde bir ifadenin ya da değerin çok fazla tekrar edildiğinde programın anlaşılabilirliğini arttırmak için kullanılır.

İsim	Talimat	Değer	Açıklama
Pi	EQU	3.14	;sabit değer tanımlama
Bilgi	EQU	55h	;55h adresindeki veriyi bilgi değişkenine ata

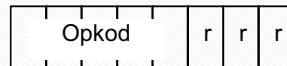
# Adresleme Yöntemleri

- Adresleme modu, bir bellek konumuna ya da bir veriye erişimin nasıl olacağını belirtir.
- Doğrudan kullanılan komut uzunluğunu etkiler.
- Kullanılan komutlara bağlı olarak bilginin farklı yollarla hedefe gitmesine olanak sağlar.
- 8051 mikrodenetleyicisinde kullanılan 8 farklı adresleme yöntemi şunlardır.
  - Kaydedici adresleme
  - Doğrudan adresleme
  - Dolaylı adresleme
  - İvedi adresleme
  - Bağlı (Koşullu) adresleme
  - Mutlak adresleme
  - Uzun adresleme
  - İndisli adresleme yöntemi

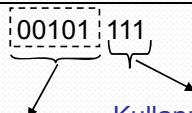


## Kaydedici Adresleme

- 8051 mikrodenetleyicisinde R0'dan R7'ye kadar 8 tane genel amaçlı kaydedici vardır.
- Kaydedici adreslemede
  - Komutu oluşturan en yüksek değerlikli 5 bit yapılacak işlevi
  - En düşük değerlikli 3 bit ise R0 ile R7 arasındaki hangi kaydedicinin kullanılacağını gösterir.



Assembly	Açıklama
ADD A, R7	;R7 kaydedicisinin içeriğini Akümülatöre ekle



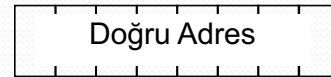
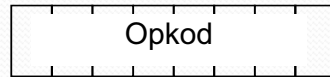
komutun makine dilindeki karşılığı 00101111B'dir

Kullanılan kaydedici R7

ADD işlemini gösteren Opkod

# Doğrudan Adresleme

- Doğrudan adresleme yöntemi, dahili alt RAM (lower RAM) ve SFR alanına erişmek için kullanılır.
- Doğrudan adresleme yönteminde komutlar 2 bayt uzunluğundadır.
  - İlk bayt opkod'u (gerçekleştirilecek işlemi),
  - ikinci bayt adres bilgisini gösterir.
- Doğrudan adresleme yöntemi adresleri örtüşen üst RAM ile SFR bölgeleri birbirinden ayrılmasını sağlar.
- Bu iki alandan SFR bölgesine doğrudan adresleme yöntemi kullanılarak erişilebilir.



# Doğrudan Adresleme

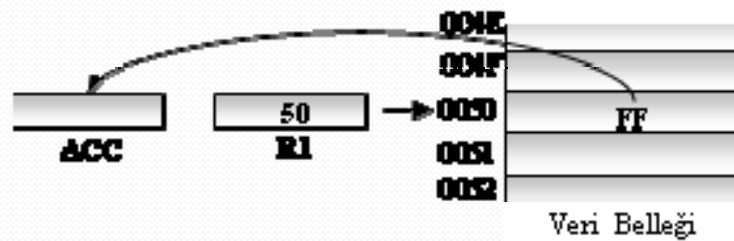
Assembly	Açıklama
MOV P1,A	;Aküyü Port 1'e kopyala
MOV A,70h	;70h adresinin içeriğini Aküye kopyala
MOV A,80h	;SFR bölgesine erişilir, 80h Port 0'ın adresidir. ;P0'daki bilgi Aküye kopyala

# Dolaylı Adresleme

- Tanımlanan bir değişkenin adresinin değiştirilmesi, hesaplanması ya da tekrar değiştirilmesi işlemlerinde dolaylı adresleme yöntemi kullanılmaktadır.
- Adresleme yöntemlerinin en güçlüsüdür.
- Bu adreslemede kaynak veya hedefin adresi komutun içerisinde açık olarak verilmez.
- Verinin gerçek adresini tutmak için R0 ve R1 kaydedicileri “işaretçi” olarak kullanılır.
- Bu kaydediciler bilginin RAM’de yazılacağı veya okunacağı adresi içermektedirler.



# Dolaylı Adresleme



Assembly	Açıklama
MOV A,@R1	;Alt RAM'deki 50h adresinin içeriği (FFh) Aküye aktar





## İvedi Adresleme

- DPTR'nin kullanıldığı istisnai durum dışında 2 bayt uzunluğundaki komutlardan oluşur.
- Bilginin geçici olarak komut içerisine yüklenmesi yüksek komut hızı sağlar.



Assembly	Açıklama
MOV A, #12	;Akü'ye 12 değerinin atılması
MOV R0, #10h	;10h bilgisini R0 saklayıcısına yükle
MOV DPTR, #2000H	;2000h bilgisini DPTR'ye yükle, 3 bayt'lık komut

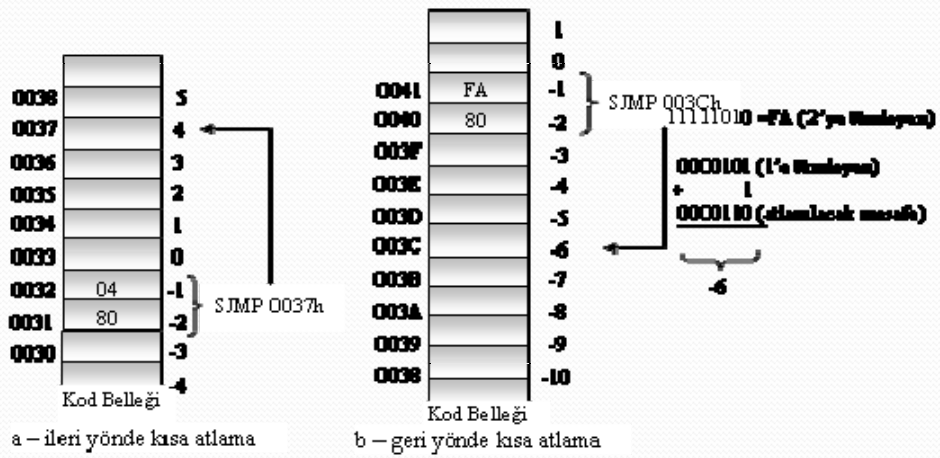


## Bağıl Adresleme

- Sadece atlama komutları ile birlikte kullanılır.
- Komutlar 1 bayt opkod ve 1 bayt adres bilgisi olmak üzere toplam 2 bayt uzunluğundadır.
- Adres bilgisi 8-bit ile ifade edildiği için maksimum +127 (ileri yön) ve -128 (geri yön) aralığında bir atlama işlemi gerçekleştirilir.



## Bağıl Adresleme



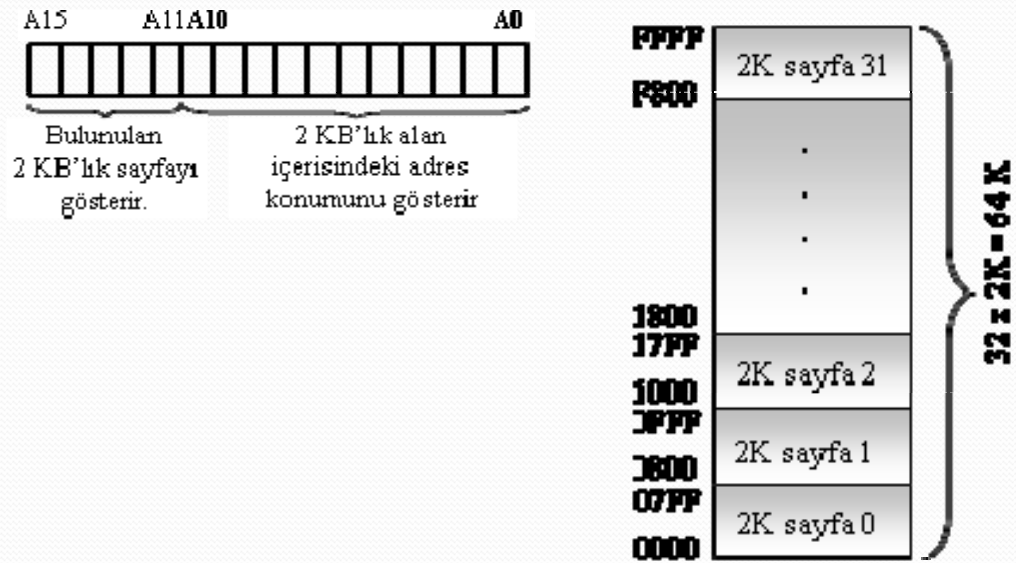
## Mutlak Adresleme

- Sadece ACALL ve AJMP komutları ile kullanılır.
- 2 bayt uzunluğundadır ve kod bellek içerisinde 2 KBaytlık bir alanı adresleyebilirler.
- Maksimum 64 KBayt olan kod bellek 2 KBaytlık 32 bölmeye ayrılabilir
- Hangi bölmenin seçileceğini program sayacı (PC) belirlemektedir.

A10-A8 Opkod

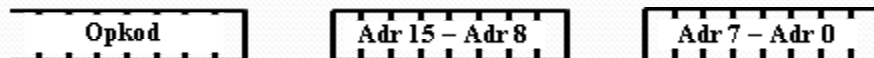
A7-A0

## Mutlak Adresleme



## Uzun Adresleme

- Yalnızca 3 bayt'lık LCALL ve LJMPL komutları kullanılır.
- 16-bit hedef adres bulunabilir.
- $2^{16} = 64$  KBaytlık adres aralığında atlama işlemi gerçekleştirilebilir.



## Sıralı Adresleme

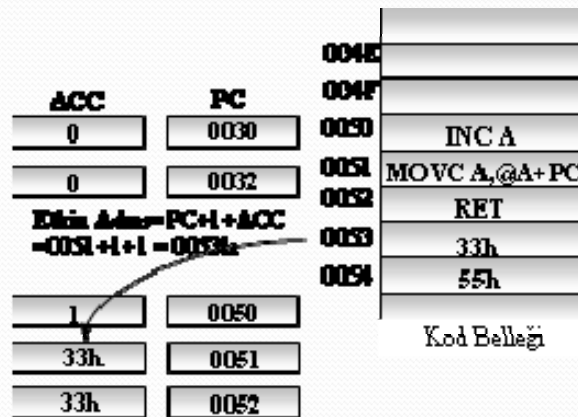
- Bellekte bulunan sıralı bilgilere erişmek için en elverişli adresleme yöntemidir.
- Çok sayıda veriye az sayıda komut kullanarak erişmek mümkündür.
- JMP ve MOVC komutları kullanılır.
- PC veya DPTR ile akümülatörün toplamı, atlanılacak olan etkin adres bilgisini belirler.

$$\boxed{\text{Opkod}} + \boxed{\text{ACC}} = \boxed{\text{Etkin Adres}}$$



## Sıralı Adresleme

Adres	Kodlar
0030h	MOV A, #0
0032h	CALL BASLA
	-----
	-----
0050h	BASLA: INC A
0051h	MOVC A, @A+PC
0052h	RET
0053h	TABLO: DB 33h
0054h	DB 55h
	-----



Sorular:

?



Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü  
Yrd.Doç.Dr. Murat İSKEFİYELİ