

## Assembly Kodlarına Giriş

⇒ Assembly kodlarında her bir komut, gerçekleştirdiği işlere karşılık gelen İngilizce kelimenin kısaltılmasıyla ifade edilir. Bu ifade sistemine "mnemonic - anımsatıcı" denmektedir.

⇒ Her bir kısaltma, makine dilindeki farklı bit dizisine karşılık gelmektedir.

# mov a, #55 ; akümülatöre 55 değerini yükle.

mov a, @55 ; 55 değerinin gösterdiği adresteki değeri akümülatöre yükle

mov a, #55 ; 55 sayısının yanında herhangi bir belirtme olmadığında, bu değeri varsayılan olarak 10'luk tabanda kabul ederiz.

mov a, #55h ; 55 sayısının yanındaki "h" sayının bize On altılı sayı sistemi (Hexadecimal) olduğunu belirtir.

## Assembly dilinin bazı avantajları

⇒ Assembly dilini daha aktif kullanabilmek için donanım hakkında fazladan bilgi sahibi olmak gerekir. Bu durum yazılımcı için bir dezavantaj gibi görülsede donanım hakkında tasarımcıya önemli bir bilgi birikimi sağlar.

⇒ Yüksek seviyeli dillere (C) göre daha küçük, daha az yer kaplayan ve daha hızlı işlenebilen kodlar üretilebilir.

## Program Formatı

**Etiket**   **Komut (işlem)**   **Operand (işlenen)**   **Acıklama**

Örnek:   **mov**   **a, #77h**   ; Akümülatöre (77)<sub>16</sub> değerini yükle

## Etiket alanı

Komut satırının ilk bilgisidir ve sembolik isimlerden oluşur.

Program içerisinde belirli işlemlerin gerçekleştirildiği bölümlerin başlangıcını göstermek amacıyla kullanılır.

Program içerisinde istenilen kısma kolaylıkla daldırılmasını sağlar.

Etiket ismi olarak mikrosistemci komut setinde tanımlı olan bir komut ismi verilemez.

Etiket bir harf ile başlamak zorundadır.



## Komut

Mnemonic tabanlı komut seti içerisinde mikroislemcinin belirli bir işi yapmasını sağlayan tanımlanmış kısaltmalardır.

Komut alanına etiketten sonra bir boşluk ya da sekme (tab) ile girilir.

## İstencen (Operand)

Bu alan, istencije istenecek 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.

## Acıklama satırı

Assembly dili noktalı virgül (;) ile başlayan satırları acıklama satırı olarak kabul eder.

Bu satırları yorumlatmaz ve makine kodu üretmez.

Yazılan uygulamanın okunabilirliğini artırır.

## Assembly dilinde sayı sistemlerinin kullanımı

Ön tiki	Son tiki	Anlamı	Örnek
(Boşluk)	D	Onlu sayı (decimal)	55 - 55D
%	B	İkili sayı (binary)	%01010 - 010110B
@	O	Sekizli sayı (octal)	@33 - 33O
\$	H	Onaltılı sayı (hexadecimal)	\$FB - FBH

## Yönergeler

### ORG

Kod bellek içerisinde programın başlangıç adresini belirlemek için kullanılan adres konumlandırma talimatıdır.

ORG 'Adres' şeklinde kullanılır.

Bir program içerisinde birden fazla ORG komutu olabilir.



## Talimat

## Açıklama

ORG 0000h ; program 0000h adresinden başlasın PC=0000

ORG 0030h ; program 0030h adresinden başlasın PC=0030

END

Programın bittiğini gösteren talimattır.

## DB (Define Byte)

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

## İsim

## DB ifadeler

## Açıklama

Max\_sayi DB 255 ; tek bir değişken tanımlama

Tablo DB 0, 5, 4, 3, -10 ; dizi olarak tanımlama

Yaz DB '8051 öğreniyorum' ; string olarak tanımlama.

## EQU

EQU (Equal = eşittir) bir sayısal değer için istenilen sembol adına ataması işlemini gerçekleştirir.

Bu tanımlama program içerisinde bir ifadenin yada değer için çok fazla tekrar edildiğinde programın anlaşılabilirliğini arttırmak için kullanılır.

## İsim

## Talimat

## Değer

## Açıklama

D1 EQU 3.14 ; sabit değer tanımlama

Bilgi EQU 55h ; 55h adresindeki veriyi bilgi değişkenine ata.