

8086 Yazmaçları ve DEBUG Komutları;

8086 YAZMAÇLARI: Merkezi işlem birimi yani cpu bilgisayarın beyni olarak tanımlanabilir. Nasıl insan vücudunda tüm vücut aktivitelerini beyin yönlendiriyorsa bilgisayarda da bu işi cpu yapar. Mikro işlemci milyonlarca transistör ve bunlardan oluşan mantık devrelerinden meydana gelen entegre devre yada çip şeklindedir. Cpu bilgisayarın işlem adımlarının tümünü yönetir. Ana kart üzerindeki tüm bileşenlerle bağlantısı vardır. İşlemci görevlerini yerine getirebilmesi için adres, bellek ve kontrol yolları ile donatılmıştır. Bilgisayarın genel performansı kullanılan işlemci ile doğru orantılı olarak değişir. Intel, 16 bitlik 8086 işlemcisini 1978 yılında piyasaya sürdü. Yüksek seviyeli programlama dillerine ve daha etkin işletim sistemlerine sahip ilk işlemci olan 8086, IBM uyumlu sistemlerin temelini oluşturdu. Arkasından çıkan 8088 işlemci ile IBM ilk kişisel bilgisayarı (PC) piyasaya sürdü. Bu ilk PC'nin 16K hafızası, grafik özelliği olmayan ekranı ve bir teyp bandı sürücüsü vardı. Bu ilk işlemci dış veri yolu olarak 8 biti destekliyordu ve 4.77 MHz saat hızında çalışmaktaydı.

Pin Bağlantıları

AD15-AD7 (8086) Adres/Data Bus

Bu hatlar 8086 üzerindeki adres/data bus hattının multiplex edilmiş upperını (yüksek değerlikli 8-biti) içerir. Bu hatlar ALE ucu lojik 1 iken A15-A8 adres bitleri ve lojik 0 iken D15-D8 data bus bitleri ni içerir. Bu pinler mikroişlemci hold durumunda iken yüksek empedans gösterir. 8086/8088 mikroişlemcileri için minimum mod işlemcileri için MN/MX direkt +5V'a bağlanarak yapılır. Pin +5V'a bir pullup direnciyle bağlanamaz.

1. **IO/M(8088), M/IO(8086) Bellek Giriş/Çıkış;** Hem bellek hem de I/O adreslerinin bulunduğu Mikroişlemci adres bus hattındaki bellek yada I/O'yu gösteren bir pindir. Mikroişlemci hold durumunda iken yüksek empedans gösterir.
2. **WR-Write** ;8086/8088 'nın bellek yada I/O birimleri için data çıkışını gösteren bir strobdur. Bu pin Mikroişlemci hold durumunda iken yüksek empedans durumuna geçer.
3. **INTA-Interrupt kabul edildi** ;INTR giriş pininden sorumludur. INTA pini genelde data üzerindeki interrupt isteklerinden sorumlu olan interrupt vektöre bir kapı olarak kullanılır.
4. **ALE-Adres Latch Enable** ;8086/8088 adres data bus hattının adres bilgileri içerdiğini gösterir. Bu adres bellek yada I/O port numarası olabilir. Mikroişlemci hold durumunda iken bu pinin sinyal vermediğine dikkat edilmelidir.
5. **DT/R Data Transmit(gönderme)/Receive(alma)** ;Mikroişlemcinin veriyi aldığını veya verdiğini gösterir. Bu sinyal harici data bus bufferlarını etkin hale getirmek için kullanılır.
6. **DEN-Data Bus Enable** ;Harici data bus bufferlarını etkin hale getirmek için kullanılır.
7. **HOLD-Hold** ;Direkt belleğe erişme isteğidir. HOLD lojik 1 ise mikroişlemci yazılım yürütme işlemini durdurur ve adres, data ve kontrol bus hatları yüksek empedans durumuna geçerler. Eğer HOLD lojik 0 seviyesinde ise mikroişlemci yazılımını normal olarak yürütür.

8. **HLDA-Hold Acknowledge(Hold isteği kabul edildi)** ;8086/8088'in HOLD durumuna geçtiğini gösterir.
9. **SS0 (8088)** ; Mikroişlemci max modda çalışırken kullanılan S0'a eşdeğerdir. Bu sinyal IO/M ve DT/R ile birleştirilerek o andaki bus çevriminin yaptığı işlevi decode etmek için kullanılır.

DEBUG KOMUTLARI:

Assembly komutlarının nasıl çalıştıklarını deneyerek görmek ve yazılan programların amacına uygun çalışıp çalışmadığını anlamak varsa hatalarını düzeltmek için DOS'un **DEBUG.COM** isimli programı kullanılmaktadır. Her ne kadar daha gelişmiş, kullanıcıya hitap eden debugger'lar olsa da kolay bulunulabilirliği ve basit uygulamalarda sağladığı rahatlık sayesinde tercih nedenidir. DEBUG programını kullanabilmek için komutlarını bilmek gereklidir.

C:\>DEBUG;

-

Çizgi işareti artık debug programının komut kabul etmeye hazır olduğunu belirtmektedir.

- R (Register) : Bu komut o andaki register değerlerini ve bayrakların durumunu belirlemekte kullanılır. Bu komut yardımı ile register'lara istenilen değerler (hexadecimal olmak şartıyla) verilebilir.

Flag İsmi	Set(1)	Clear (0)
Overflow - OF	OV (overflow)	NV (not overflow)
Direction - DF	DN (decrement - down)	UP (increment - up)
Interrupt - IF	EI (enable)	DI (disable)
Sign - SF	NG (negative)	PL (plus)
Zero - ZF	ZR (zero)	NZ (not zero)
Auxiliary Carry- AF	AC (auxiliary carry)	NA (not auxiliary)
Parity - PF	PE (even parity)	PO (odd parity)
Carry - CF	CY (carry yes)	NC (not carry)

Tablo 18-1 Debugger'da kullanılan Flag kısaltmaları

- D (Dump / Display): Bu komut ile belleğin istediğimiz segment'inin istediğimiz offset'inden itibaren bilgileri görebiliriz. İstersek görmek istediğimiz alanı sınırlama imkanına da sahibiz. D komutunun yanına gösterilmesi istenen aralığın adresi verilecek olursa sadece belirlenen adres aralığındaki değerler ekranda görüntülenecektir.

A (Assembly) : Bu komut Debug yardımıyla basit programlar yazmayı sağlar. Debug içinde yazılan programlar COM uzantılı programlarda olduğu gibi 100H offset adresinden başlarlar. Bu neden ile program çalıştırılacağı zaman mutlaka IP register'ının değerinin **R** komutu ile 100H e getirilmesi gerekmektedir.

- T (Trace) : Yazılmış olan assembly komutları teker teker çalıştırılarak işlem sonunda register ve bayrakların aldıkları değerleri ekranda gösterir. T komutu çalışmaya IP nin gösterdiği noktadan itibaren başlar ve her komut işlendikten sonra durur. Eğer kullanıcı ister ise T komutunun yanına bir sayı da verebilir. Bu durumda verilen sayı kadar komut satırı arka arkaya çalıştırılır, her seferinde register ve flag değerleri ekrana yazılır.

- U (Unassembly) : CS'de yazılı olan makine kodunu kullanıcının anlayacağı formdaki mnemonic'lere çevirmek için kullanılan bir komuttur. Her çağırılışında 32 byte uzunluğunda makine kodunun mnemonic'lerini kullanıcıya gösterir.

- **E (Edit)** : Bellekte herhangi bir yerdeki deęerleri deęiřtirmek amacıyla kullanılan bir komuttur.

- **F (Fill)** : Bellek bařlangıç ve bitiş adresleri belirlenen bir alanı belirlenen bir sayı ile doldurmak amacıyla kullanılır.

C (Compare) : Belirlenen aralıktaki bilgileri karşılařtırmak amacıyla kullanılan bir komuttur. farklı segment'ler üzerindeki bilgiler de karşılaştırılabilir.

-**G (Go)**: Programın verilen bir adrese kadar ya da bitene kadar çalışmasını saęlamak amacıyla kullanılır.

-G

Program terminated normally

Program eęer düzgün çalıştı ise yukarıdaki mesaj ekrana gelecektir. Programın dönüş adreslerinde bir hata varsa ya da programda bir mantık hatası yapılmış ise program hiçbir zaman bitmez. Bu durumda makinenin reset edilmesi gerekecektir.

- **H (Hex)** : Verilen iki sayının hexadecimal toplamını ve farkını hesaplamak için kullanılır.

- **L (Load)** : Belleęe programı tekrar yüklemek için kullanılır.

- **M (Move)** : Bir yerdeki bir blok bilgiyi başka bir yere taşımak için kullanılır

- M DS:100,110,CS:500

Bu komut ile DS:100H ile DS:110H arasında kalan bilgi CS:500H adresinden itibaren yerleřtirilecektir.

- **P (Proceed)** : Bu komut T komutu ile benzer özellikler göstermektedir. Komutları çalıştırdıktan sonra register ve flag deęerlerini ekranda gösterir. Ancak T komutundan en büyük farklılığı özellikle LOOP, CALL, INT gibi komutların işlenmesindedir. Bu komutlardan herhangi biri P komutu ile çalıştırılacak olursa o işlemin detaylarına girmeden sonucu üretir. (Eęer bir LOOP ise çevrim sonuçlanana kadar, bir CALL ise işlemler yerine getirilip geri dönülene kadar ya da bir INT ise interrupt handler'in yapması gereken tüm işlemler yapıldıktan sonra register ve Bayrakların durumlarını gösterir).

- **Q (Quit)**: Debugger dan çıkmak için kullanılan bir komuttur.

-Q

C:\>

KAYNAKLAR:

- Assembly kodları ,Mühendislik Mimarlık Fakültesi Bilgisayar Mühendislięi Bölümü

- Mikroişlemciler ,yıldız üniversitesi yayınları

- www.gencbilim.com

