

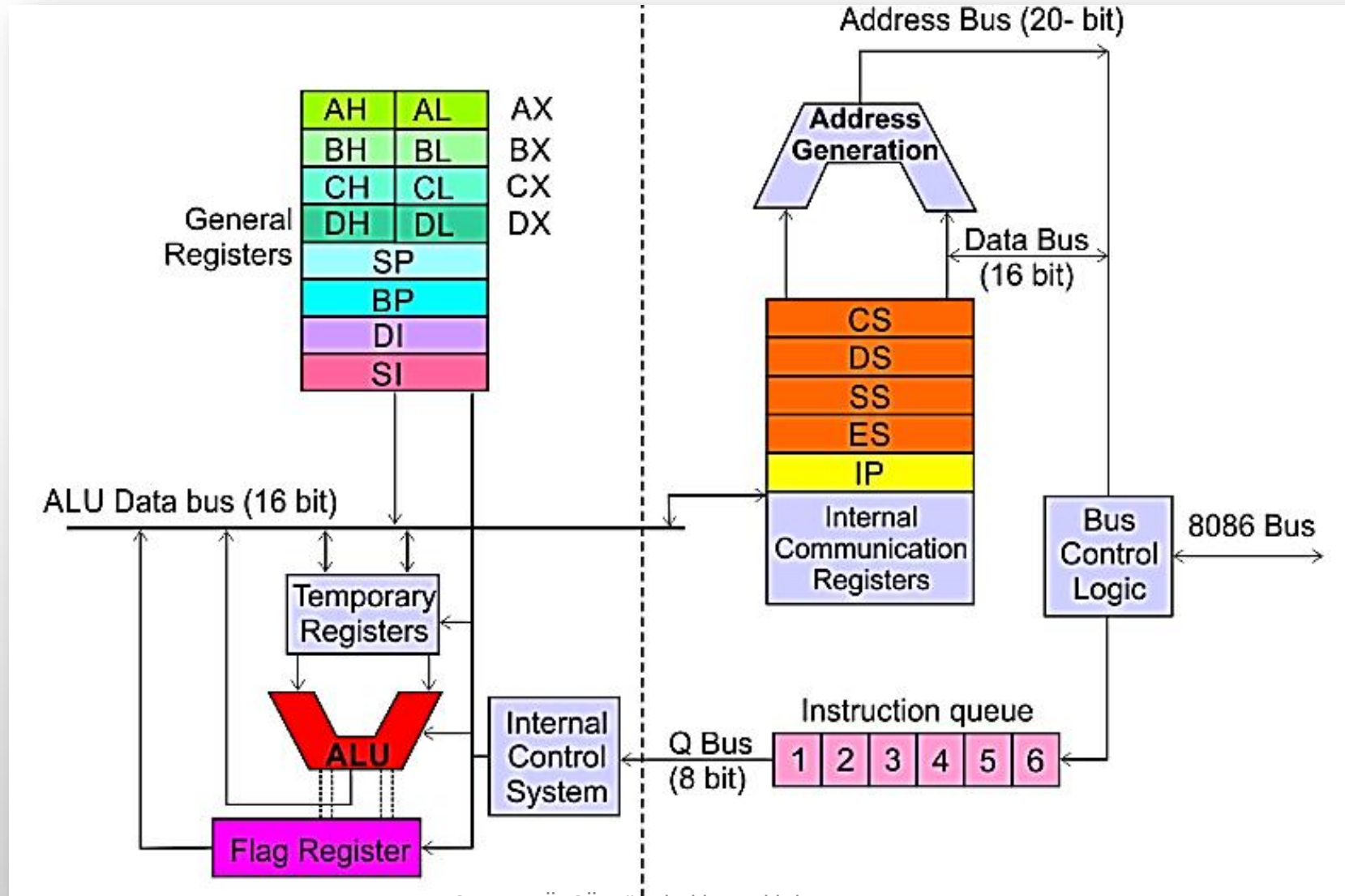


# **Bölüm 1: 8086 Mimarisi**

## **Mikroişlemciler**



# 8086 Mimarisi





# 8086 Mimarisi

- 8086, Intel tarafından tasarlanmış 8-bit/16-bit mikroişlemci.
- x86 mikroişlemci ailesinin ilk üyesidir.
- Kişisel bilgisayarlarda yaygın olarak kullanılır.
- Karmaşık komut kümesi bilgisayar (CISC) mimarisine dayanır.
- 20-bitlik adres veri yolu bulunur.
- 1 MB belleği adresleyebilir. ( $2^{20}$ )
- 16-bitlik veri yolu,
  - mikroişlemci, bellek, I/O aygıtları arasında veri transferi sağlar.



# Kesimli Bellek Mimarisi

- *Segmented Memory Architecture*
- Bellek,
  - kesim yazmacı (*segment register*) ve
  - bağıl konum (*offset*) kullanılarak adreslenir.
- Kesim yazmacı, kesimin başlangıcına işaret eder.
- Bağıl konum, belirli bir *byte*'ın kesim içindeki konumunu belirtir.
- Bu yöntemle *20-bitlik* adresleme yapılmakta.



# 8086 Mimarisi

- 8086 mikroişlemcisi, iki temel birime sahip.
  - Yürütme Birimi (*Execution Unit - EU*).
    - BIU'dan alınan komutları işleyerek yürütür.
    - Aritmetik ve mantıksal işlemleri yapar.
  - Veri Yolu Arayüz Birimi (*Bus Interface Unit - BIU*).
    - Komutları bellekten alır ve çözer.
    - Mikroişlemci, bellek ve G/Ç aygıtları arasında veri transferini yönetir.
    - Mikroişlemcinin işlemesi gereken verilere erişimini sağlar.



# Yazmaç Kümesi

- Genel amaçlı yazmaçlar (*general-purpose registers*)
  - Veri saklar ve aritmetik/mantıksal işlemleri gerçekleştirir.
- Kesim yazmaçları (*segment registers*)
  - Bellek kesimlerini adresler.
- Özel yazmaçlar (*special registers*)
  - Bayrak yazmaçları (*flags register*)
    - Önceki işlemin sonucuyla ilgili bilgileri içerir.
  - Komut işaretçisi (*instruction pointer*)
    - Bir sonraki komutun konumunu belirtir.



# Bellek Yapısı

- Mikroişlemci, *CPU*'nun tüm işlevlerini içeren bir entegre devredir.
- Mikroişlemci tek başına kullanılamaz;
  - Mikrodenetleyici gibi kendi belleği ve çevre birimleri yoktur.
- 8086, içerisinde *RAM* veya *ROM* barındırmaz.
- Geçici ve nihai sonuçları saklamak için yazmaçlara sahiptir.
- Bellek erişimi, sistem yolu (*system bus*) üzerinden gerçekleşir.

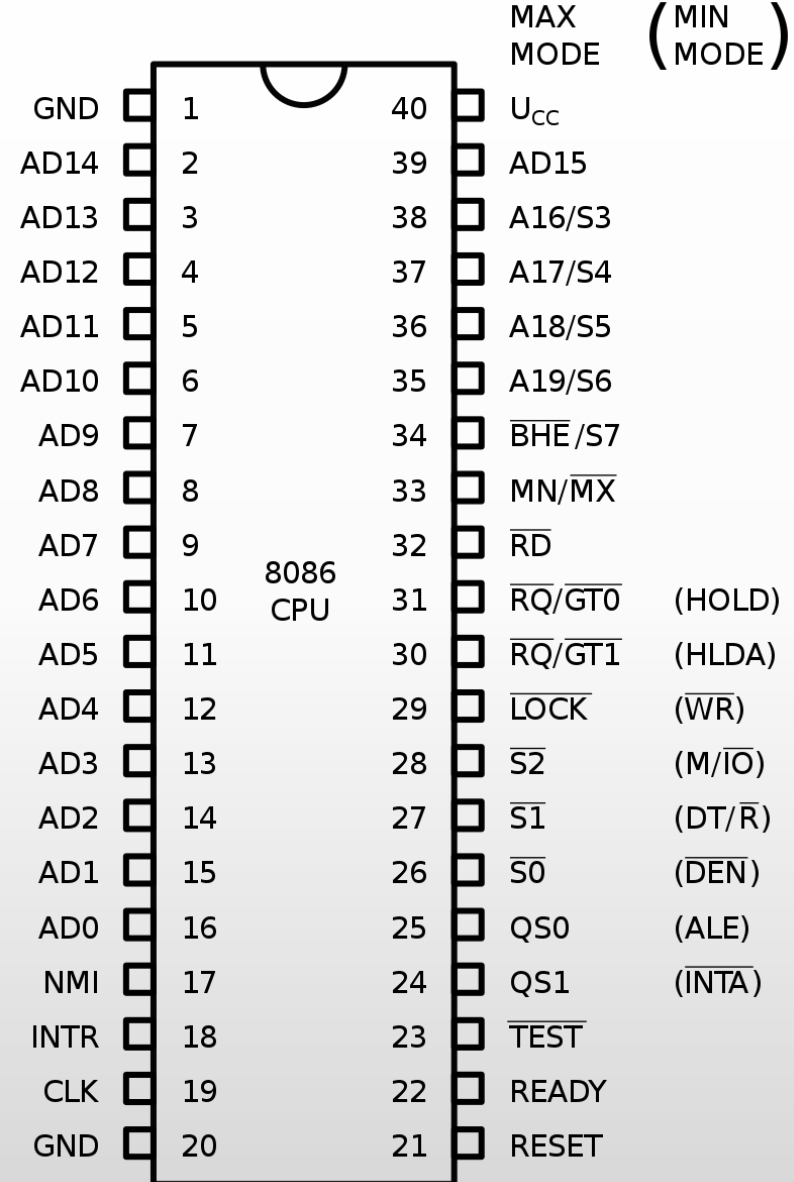
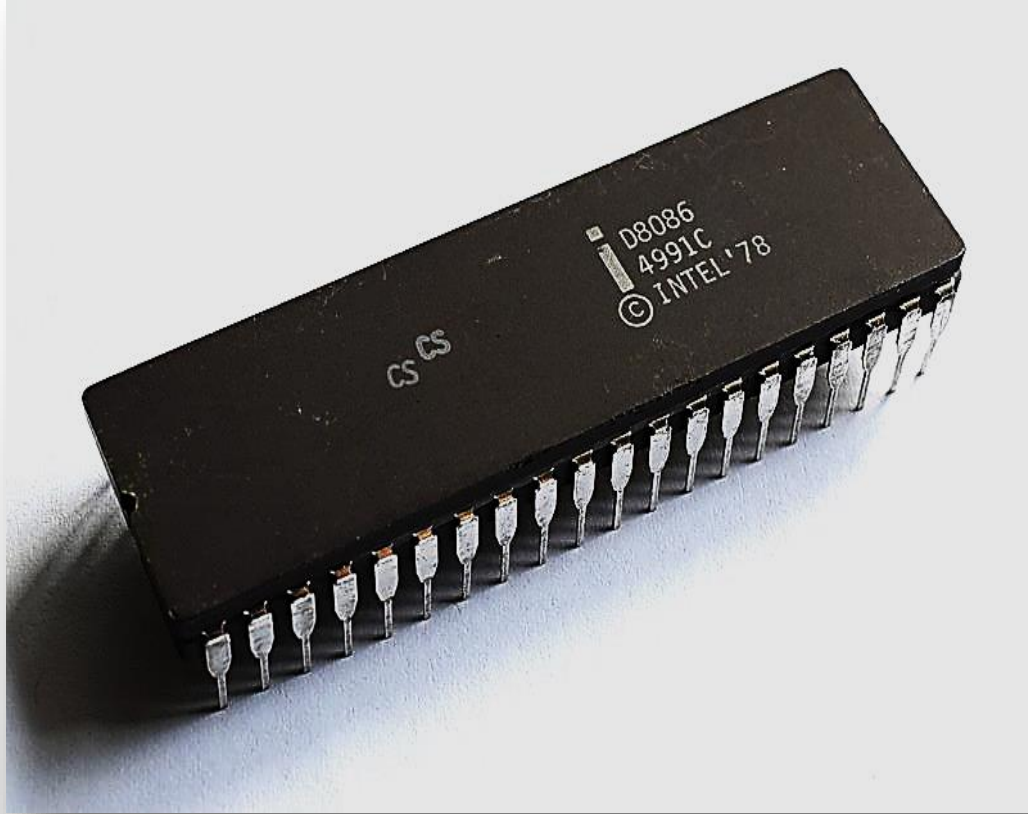


# Dahili Yazmaçlar

- 8086,
  - *16-bitlik* bir tamsayı (vektörel olmayan) işlemci,
  - 40 pinli, çift sıralı paketli (*dual inline packaged*) olarak tasarlanmıştır.
- Yonga içinde bulunan yazmaçların boyutu,
  - işlemcinin aynı anda üzerinde çalışabileceği veri miktarını gösterir.
  - dolayısıyla *16-bitlik* yazmaçlar.
- Programcıya *14 adet 16-bitlik* dahili yazmaç sunar.
- Yazmaçlar, işlemci içinde veri taşıma ve işleme yeteneğini temsil eder.



# 8086 Pin Diyagram





# Bellek Kesimleme

- Yürütme ve veri getirme hızını artırmak amacıyla bellek kesimlere ayrılır.
- 8086'nın *20-bit* adres yolu, *1 MB* belleği adresleyebilir.
- Ancak, belleği *16 adet 64 KB'lık* kesimlere böler.
- 8086, *1 MB* bellek içinde sadece *4 adet 64 KB'lık* kesimle çalışır.



# Veri Yolu Arayüz Birimi (BIU)

- Harici bellek ve  $G/\mathcal{C}$  aygıtlarına sistem yolu üzerinden arayüz sağlar.
- *20-bit* fiziksel bellek erişimi için adres üretir.
- Bellekten komutları getirir.
- Bellek ve  $G/\mathcal{C}$  arasında veri transferi yapar.
- 6-byte önden getir (*prefetch*) komut kuyruğunu (*instruction queue*) yönetir.
  - boru hattını (*pipelining*) destekler.



# Komut İşaretçisi

- *Instruction Pointer (IP)*
- 16-bitlik bir yazmaç.
- Kod kesiminde (*code segment*) bir sonraki komutun bağlı konumunu tutar.
- Her komut (*instruction byte*) getirildikten sonra *IP* değeri artar.
- *IP*, her dallanma (*branch*) komutundan sonra yeni bir değer alır.
- *CS*, *10H* ile çarpılarak kod kesiminin *20-bit* fiziksel adresini verir.
- Bir sonraki komutun adresi,  $CS \times 10H + IP$  formülü ile hesaplanır.
- Örnek:  $CS = 4321H$ ,  $IP = 1000H$ , olsun.
  - Sonraki komut adresi =  $4321 \times 10H + 1000H = 44210H$ .



# Kesim Yazmaçları

- Kesimler bellekte, kesim yazmaçları mikroişlemci içerisinde bulunur.
- Kesim yazmaçları, bellekteki her kesimin başlangıç adresini saklar.
- *Code Segment (CS)*:
  - Programın saklandığı ve IP ile erişilen kesimin adresini tutar.
- *Data Segment (DS)*:
  - Veri kesiminin temel adresini (*base address*) tutar.
- *Stack Segment (SS)*:
  - Yığın kesiminin temel adresini tutar.
- *Extra Segment (ES)*:
  - Ek kesimin temel adresini tutar.



## 6 Bytelik Önden Getir Komut Kuyruğu

- *Prefetch Instruction Queue*
- 6 byte'lık bir kuyruk yapısı (*FIFO - İlk Giren, İlk Çıkar*).
- Bir komut yürütülürken, sonraki komutun getirilmesi *pipelining*.
- Kuyruk, sıralı bir şekilde çalışır.
- Bir dallanma komutu yürütüldüğünde temizlenir.
- Bir komutun boyu (işlenen ve veri dahil) maksimum 6 byte'dır.
- 2 byte'lık bir boşluk olduğunda sonraki komut getirilmeye başlanır.



# Önden Getirme Birimi

- *Prefetch Unit*
- Komutları bellekten getirir ve bir kuyrukta saklar.
- Paralel olarak birden fazla komut getirilerek performansı artırır.
- Bellek erişim süresini azaltır, birim sürede daha fazla komut yürütülebilir.
- Bir tampon (*buffer*) ve bir program sayacından oluşur.
  - Tampon, getirilen komutları saklar.
  - Program sayacı, sonraki getirilecek komutun bellek konumunu tutar.
- Komutlar bellek yerine tampondan yürütülebilir.



# Yürütme Birimi

- *Execution Unit*
- Başlıca bileşenleri,
  - Genel amaçlı yazmaçlar (*general purpose registers*)
  - Aritmetik mantık birimi (*arithmetic and logic unit*)
  - Özel amaçlı yazmaçlar (*special purpose registers*)
  - Komut yazmacı (*instruction register*)
  - Komut çözücü (*instruction decoder*)
  - Bayrak/Durum yazmaçları (*flags status registers*).





# Genel Amaçlı Yazmaçlar

- *AX, BX, CX* ve *DX* olmak üzere 4 adet 16-bitlik yazmaçlar.
- Her biri yazmaç 8-bitlik 2 bölüme (*high, low*) ayrılır.
- *AX (AH ve AL)*:
  - İşlenen ve sonuçları saklar. Birikeç (*accumulator*) olarak da kullanılır.
- *BX (BL ve BH)*:
  - Dolaylı adresleme modunda bellek adresini (*offset*) saklar.
- *CX (CL ve CH)*:
  - Döngü, döndürme, kaydırma işlemlerinde sayaç olarak kullanılır.
- *DX (DL ve DH)*:
  - *AX* ile birlikte 32-bitlik değerleri tutmak için kullanılır.



# Aritmetik Mantık Birimi

- *Arithmetic Logic Unit*
- *8-bit* ve *16-bit* aritmetik ve mantık işlemlerini yürütür.
- Aritmetik işlemler, toplama, çıkarma, çarpma ve bölme gibi işlemler.
- Mantıksal işlemler, *AND*, *OR*, *XOR* gibi işlemler.



# Özel Amaçlı Yazmaçlar

- *Special purpose registers*
- Belirli bellek konumlarına işaret eden bağıl konum (*offset*) yazmaçlarıdır.
- Kesim (*segment*) ve bağıl konum (*offset*) bellek yönetiminde temel bileşen.
- Yığın işaretçisi (*stack pointer*),
  - Yığınin üstünü (*top*) gösterir, *push*, *pop*, *call*, *ret* komutlarında kullanılır.
- Taban işaretçisi (*base pointer*),
  - Yığınin rastgele konumlarına erişebilmek için kullanılır.
- Kaynak indis (*source index*) ve hedef indis (*destination index*),
  - Veri (*data*) ve Ekstra kesimlerinde konum (*offset*) adreslerini tutar.



# Komut Yazmacı ve Komut Çözücü

- Komut yazmacı (*instruction register*),
  - Yürütme birimi, kuyruktan işletilecek komutu alır.
  - Komutun kodu (*opcode*) geçici olarak bu yazmaçta saklanır.
- Komut çözücü (*instruction decoder*),
  - Komut yazmacındaki işlem kodunu çözer.
  - Gerekli bilgiyi yürütme için kontrol devresine gönderir.



# Durum Yazmaçları

- Durum bayrakları, her aritmetik ve mantıksal işlemde sonra güncellenir.
  - Taşıma Bayrağı (*carry flag*)
  - Eşlik Bayrağı (*parity flag*)
  - Yardımcı Bayrağı (*auxiliary flag*)
  - Sıfır Bayrağı (*zero flag*)
  - İşaret Bayrağı (*sign flag*)
  - Taşma Bayrağı (*overflow flag*)



# Kontrol Yazmaçları

- Kontrol bayrakları, işlemleri kontrol etmek ve yönlendirmek için kullanılır.
- *CLC, STC, CLD, STD, CLI, STI* komutları ile değer atanabilir.
  - Tuzak Bayrağı (TF)
  - Kesme Bayrağı (IF)
  - Yön Bayrağı (DF)



# Çözümleme Birimi

- *Decode Unit*
- Bellekten alınan makine komutlarını, mikro işlemlere çevirir.
- Önden getir (*prefetch*) birimi ile paralel olarak çalışır.
- Koşullu atlama (*conditional jumps*), çağrı (*call*) ve dönüş (*return*) gibi karmaşık komutları çözümler.
- Yazmaçtan yazmaca ve bellekten yazmaca veri transferlerini gerçekleştirir.
- Performansı arttırmaya yardımcı olur.



# Kontrol Birimi

- *Control Unit*
- Komut ve verilerin mikroişlemci içindeki akışını yönetir.
- Bellekten komutları almak (*fetch*), çözmek (*decode*), yürütmek (*execute*),
  - Mikroişlemcinin durumunu (*state*) güncellemekten sorumludur.
- Kesme isteklerini (*interrupt request*) yönetir.
- Güç yönetimi (*power management*),
  - Hata ele alma (*error handling*) görevlerini gerçekleştirir.
- Diğer bileşenlerin faaliyetlerini koordine eder.





# Veri Yolları

- Adres Yolu (*Address Bus*):
  - Okunacak/yazılacak komut/verinin bellek adresi iletilir.
  - 16 bit genişliğinde, bellekte 64 KB'a kadar adreslemeye olanak tanır.
- Veri Yolu (*Data Bus*):
  - Mikroişlemci ile bellek arasında veri transferi için kullanılır.
  - 16 bit veriler tek seferde transfer edilebilir.
- Kontrol Yolu (*Control Bus*):
  - Mikroişlemci ile diğer bileşenler arasında kontrol sinyalleri iletilir.
  - Okuma, yazma, kesme istekleri, durum bilgisi gibi sinyaller iletilir.



# 8086 Mimarisinin İşleyişi

1. Getirilecek komutun fiziksel adresi hesaplanır.
  1. Veri yolu arayüz birimi, kesim yazmaçlarını kullanarak hesaplar.
  2. Örneğin;
    1. kod kesimi (CS) bir kesim adresi (*segment address*) içeriyor,
    2. komut işaretçisi (IP) bir bağıl konum (*offset*) değeri içeriyor
    3. fiziksel adres hesaplayıcı getirecek komutun konumunu bulur.
2. Getirecek komut veri yoluna konur.
  1. Komut önden getir (*prefetch*) kuyruğuna gelir.
  2. MOV AX,BX 1 byte'lık komut
  3. MOV BX,4050H 3 byte'lık komut



## 8086 Mimarisinin İşleyişi

3. FIFO mantığına göre komut çalıştırılmaya hazır olduğunda,
  1. Yürütme birimindeki kontrol sistemine gelir.
  2. Çözme sistemi bir kod (*opcode*) üretir.
  3. Kontrol sistemi gerekli genel ve özel yazmaçlarla ilgili sinyal üretir.
4. Genel yazmaçlardan alınan veri ALU'ya gelir.
  1. Gerekli hesaplama yapılır.
5. Bayrak yazmaçları güncellenir.



SON