

# Mikroişlemci Sistemleri

Dr. Öğr. Üyesi Hamza Osman İLHAN

2019/1-Ders 2

# Mikroişlemci

## Kaydediciler

İşlemci içerisinde ham bilgi girdisinin hızlı biçimde işlenerek kullanılabilir çıktıya dönüştürülmesi için sistemde verileri geçici olarak üzerinde tutacak bir grup veri saklayıcıya ihtiyaç duyulmaktadır. Kaydediciler verinin manevrasında ve geçici olarak tutulmasında görevlidirler.

## Aritmetik ve Mantık Birimi (ALU)

Aritmetik işlemler denilince başta toplama, çıkarma, çarpma, bölme, mantık işlemleri denilince AND, OR, EXOR ve NOT gibi işlemler akla gelir. Komutlarla birlikte bu işlemleri mantık kapılarının oluşturduğu toplayıcılar, çıkarıcılar ve kaydıran kaydediciler gerçekleştirirler. ALU'da gerçekleşen bütün bu işlemler kontrol sinyalleri vasıtasıyla Zamanlama ve Kontrol Biriminin gözetiminde eş zamanlı olarak yapılır.

# Mikroişlemci

## Zamanlama ve Kontrol Birimi

Zamanlama ve kontrol birimi, bellekte program bölümünde bulunan komut kodunun alınıp getirilmesi, kodunun çözülmesi, ALU tarafından işlenilmesi ve sonucunun alınıp belleğe geri konulması için gerekli olan kontrol sinyalleri üretir. Bilgisayar sisteminde bulunan dahili ve harici bütün elemanlar bu kontrol sinyalleri ile denetlenir.

Basit bir mikroişlemcide bu bölüm 3 değişik işlevi yerine getirir:

1. Zamanlama Kontrolü: İşlemci, harici saat sinyali üreten bir birimden giriş alan iç-saat devresine sahiptir. Bir sinyal alınarak talebe göre zaman sinyallerine çevrilerek sisteme dağıtılır.
2. Komut kod çözücüsü: Bu devre komut kaydedicisinde tutulan komutları yorumlar ve ALU'ya kaydedicilerle çalışması için uygun sinyaller gönderilir.
3. Kesme ve Mantık Birimi: Bu birim diğer kontrol elemanlarına benzer. Gerekli durumlarda kesme sinyallerini alarak işlemciyi uyarır.

# 8086 Mimarisi

## 16 Bit İşlemci

İşlemci içerisindeki bölümler fonksiyonel açıdan iki mantıksal ana bölüm halinde daha ayrıntılı olarak açıklanmaktadır.

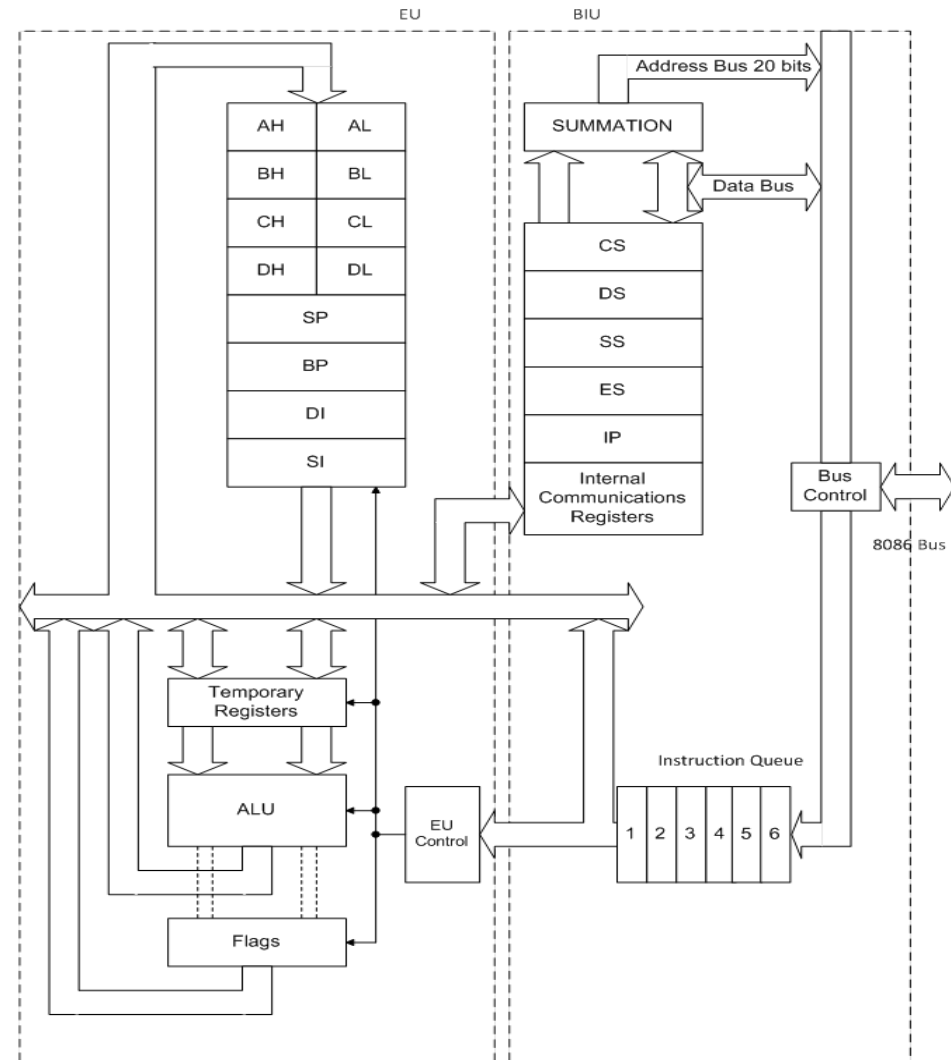
16 bitlik x86 tabanlı işlemciler

- Veri yolu bağdaştırma birimi (BIU)
- İcra birimi (EU)

olmak üzere iki ana bölümde toplanabilir.

BIU birimi, EU birimini veriyle beslemekten sorumluyken, icra birimi komut kodlarının çalıştırılmasından sorumludur.

BIU bölümüne segment kaydedicileriyle birlikte IP ve komut kuyrukları ve veri alıp getirme birimleri dahilken, EU bölümüne genel amaçlı kaydediciler, kontrol birimi, aritmetik ve mantıksal komutların işlendiği birim dahildir. Bu birimler çip üzerinde birbirine fiziksel olarak bağlıdır.



# 8086 Yazmaçları

- AX
  - AL
  - AH
- CX
  - CL
  - CH
- BX
  - BL
  - BH
- DX
  - DL
  - DH
- SI
- DI
- SP
- BP

Genel amaçlı yazmaçlar

Segment yazmaçları

- SS
- CS
- DS
- ES
- IP

- FLAGS
  - CF
  - PF
  - AF
  - ZF
  - SF
  - TF
  - IF
  - DF

Özel amaçlı yazmaç

Bayrak yazmacı

# 8086 Yazmaçları – AX, AL, AH

- AX : 16 bitlik akümülatör yazmaç
- AH, AL : 8 bitlik akümülatör yazmaçlar
- Aritmetik, lojik ve veri transferi işlemlerinde kullanılabilir
- Çarpma ve bölme işlemlerinde gizli operand olark kullanılır
- Giriş çıkış komutlarında kullanılır

# 8086 Yazmaçları – BX, BL, BH

- BX : 16 bitlik genel amaçlı yazmaç, (base register)
- BL, BH : 8 bitlik genel amaçlı yazmaçlar
- Dizi şeklindeki veri erişiminde kullanılır

# 8086 Yazmaçları – CX, CL, CH

- CX : 16 bitlik genel amaçlı yazmaç
- CL, CH : 8 bitlik genel amaçlı yazmaçlar
- Tekrarlı işlemlerde tekrar sayısını saklar (CX)
- Öteleme ve kaydırma işlemlerinde tekrar sayısını saklar (CL)



# 8086 Yazmaçları – DX, DL, DH

- DX : 16 bitlik genel amaçlı yazmaç
- DL, DH : 8 bitlik genel amaçlı yazmaçlar
- Çarpma ve bölme komutlarında bölünen sayıyı oluşturmak için kullanılır
- Giriş çıkış işlemlerinde port numarasını saklar

# 8086 Yazmaçları – SP

- SP : yığın yazmacı (stack pointer)
- Yığının en üst adresini işaretlemek için kullanılır
- SS ile birlikte kullanılır
- Her zaman çift bir değer gösterir
- WORD tipinde veriyi gösterir

# 8086 Yazmaçları – BP

- BP : Base pointer
- Fonksiyona parametre aktarılırken kullanılır
- SS ile birlikte kullanılır

# 8086 Yazmaçları – SI

- SI : kaynak indisi yazmacı (source index)
- Dizi komutlarında kaynak indisini tutar
  - DS ile birlikte kullanılır

# 8086 Yazmaçları – DI

- DI : hedef indisi yazmacı (destination index)
- Dizi komutlarında hedef indisini tutar
  - ES ile birlikte kullanılır

# 8086 Kesim (Segment) Yazmaçları

- CS : Kod segment, IP ile kullanılır
  - DS : Data segment, BX, SI, DI ile kullanılır
  - ES : Extra segment, DS gibi
  - SS : Stack segment, BP ve SP ile kullanılır
- 
- DS=1230H, SI=0045H ikilisi ile erişilen fiziki adres  
 $12300H + 0045H = 12345H$

# 8086 Yazmaçları – IP

- IP : Instruction pointer
  - Sıradaki işlenecek komutu gösterir
  - CS ile birlikte kullanılır
- 
- Efektif program adresi :
  - $CS \times 10H + IP$

# 8086 Bayrak Yazmacı

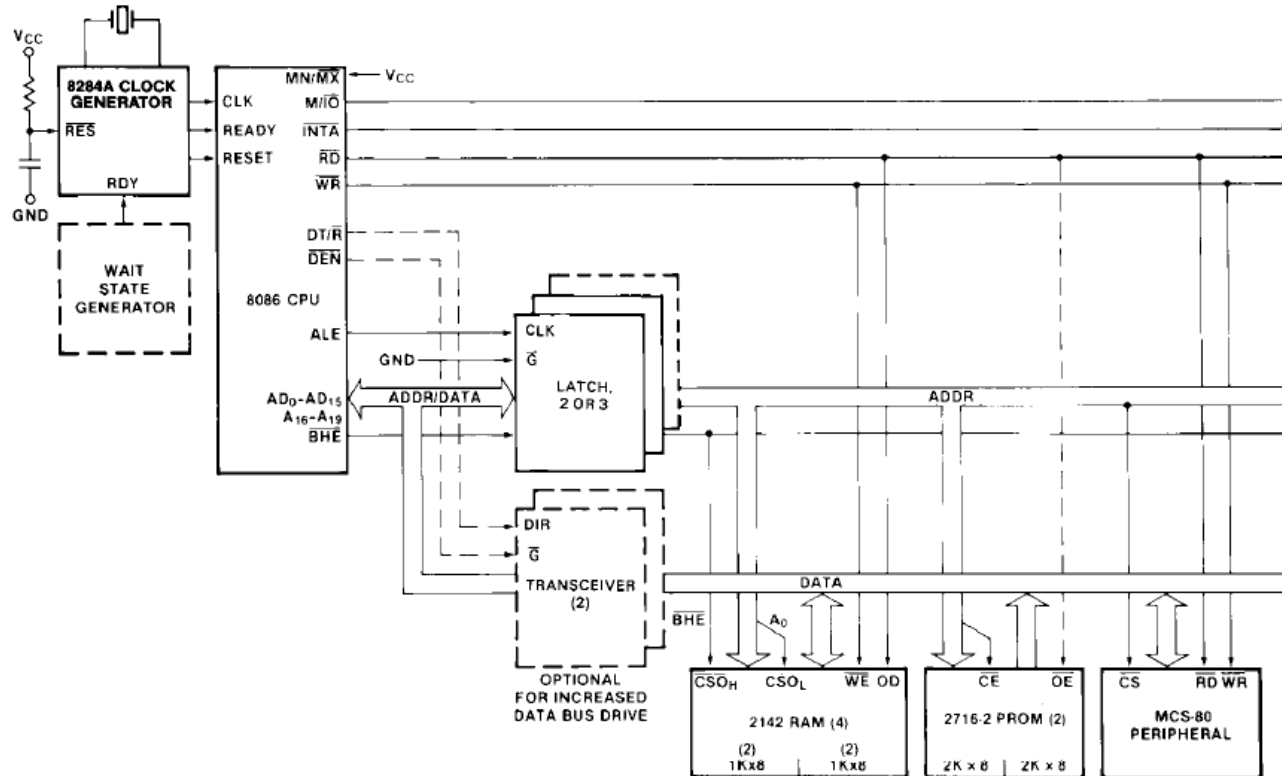
- **Carry Flag (CF)** : İşaretsiz işlemlerde taşma olursa 1 değerini alır
- **Parity Flag (PF)** : İşlem sonucunda 1 olan bitlerin sayısı tek ise 0, çift ise 1 değerini alır
- **Auxiliary Flag (AF)** : 4 bitlik kısımların toplama-çıkarma sonucu elde değerini tutar
- **Zero Flag (ZF)** : İşlem sonucu 0 ise ZF=1 olur
- **Sign Flag (SF)** : İşlem sonucu negatif ise SF=1 olur



# 8086 Bayrak Yazmacı

- **Trap Flag (TF)** : Her komuttan sonra kesme oluşmasını sağlar
- **Interrupt enable Flag (IF)** : Kesme kaynaklarının kesme oluşturmalarına izin verir
- **Direction Flag (DF)** : Dizi işlemlerinde başlangıç adresinden itibaren arttırarak/azaltarak sıradaki göze erişimi belirler
- **Overflow Flag (OF)** : İşaretli işlemlerde taşma durumunda 1 değerini alır

# 8086 $\mu$ P Sistem Blok Yapısı



# Aritmetik Komutlar

- ADD
- ADC
- SUB
- SBB
- INC
- DEC

- NEG
- CMP
- MUL
- IMUL
- DIV
- IDIV

# Ders-2 Konular

- Çevre Birimleri
  - LED Arayüzü
- G/Ç Assembly Komutları
- G/Ç Haritalama Yöntemleri
- Basit Arayüz Devreleri
  - Düğme Arayüzü
- Basit Çevre Birimleri ve Adres Çözümleme
  - Basit Çıkış Birimi
  - Basit Giriş Birimi
- Çevre Birimleri Kontrolü

# Çevre Birimleri (Peripheral devices)

- Giriş ve/veya çıkış cihazları
- Hafıza birimleri gibi belirli bir adres bölgesine yerleştirilir
- Hafıza birimine göre adres genişliği dardır (1-4 byte)

# Çevre Birimleri

- Çıkış Birimi : Hafıza birimine benzer şekilde çıkış birimlerine veri yazılabilir ➔ OUT
- Giriş Birimi : Hafıza birimiyle benzer şekilde giriş birimlerinde veri okunabilir ➔ IN

# G/Ç Assembly Komutları

	IN Komutu	OUT Komutu
Fixed address	IN AL, p8	OUT p8, AL
	IN AX, p8	OUT p8, AX
Variable address	IN AL, DX	OUT DX, AL
	IN AX, DX	OUT DX, AX

# G/Ç Assembly Komutları

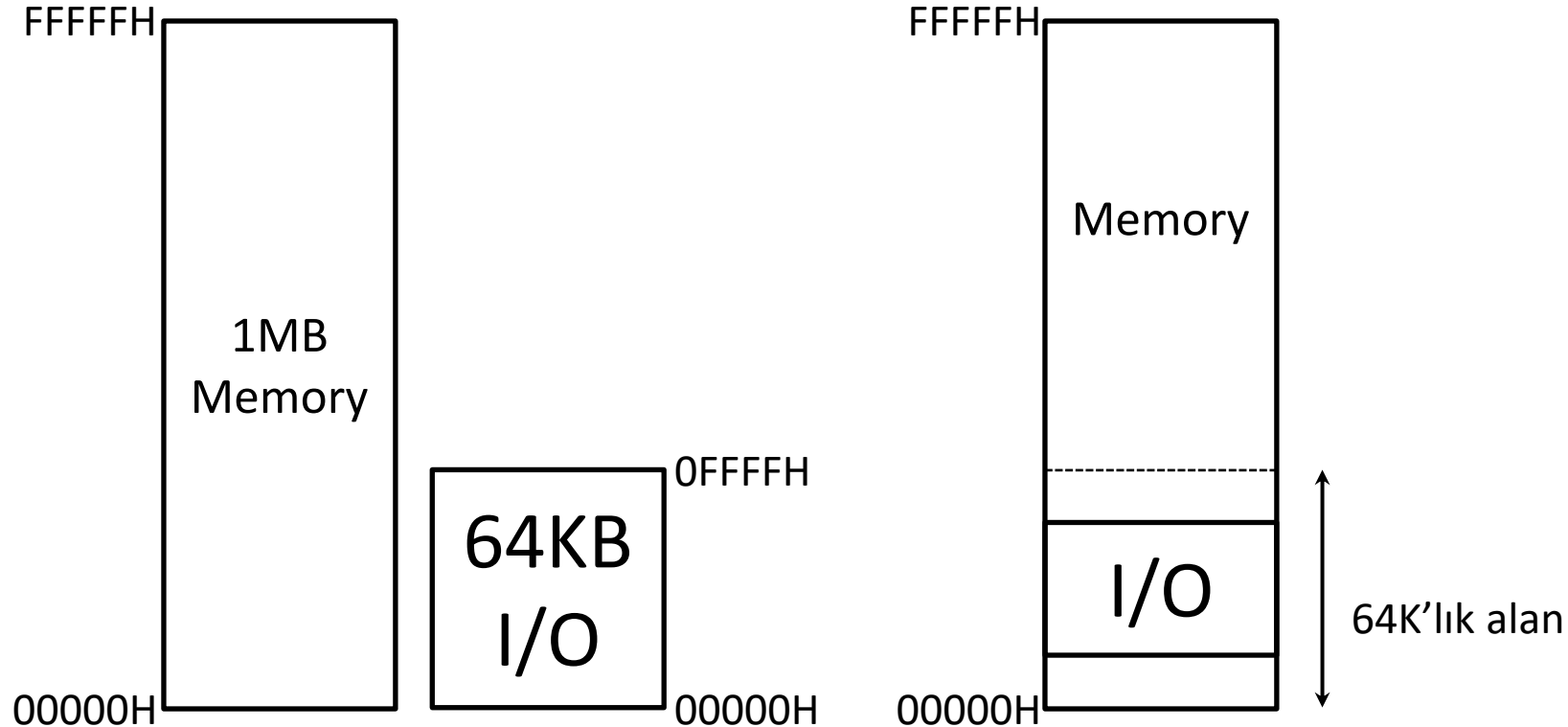
- 0-0FFH arası G/Ç (fixed address) işlemlerinde adres değeri komut içinde saklanır
- 0100H-FFFFH arası adresler DX yazmacı ile dolaylı olarak oluşturulur
- Herhangi bir G/Ç adresinden 8 bitlik veya 16 bitlik veri işlemi yapılabilir



# G/Ç Haritalama Yöntemleri

- Isolated I/O Mapping  $\rightarrow M/\overline{IO}$  ucu adres  
çözümleme için kullanılır
- Memory Mapped I/O  $\rightarrow M/\overline{IO}$  ucu adres  
çözümleme için kullanılmaz

# Isolated I/O – Memory Mapped I/O



# Isolated I/O

- Hafıza uzayının tamamı hafıza birimleri için kullanılabilir
  - Hafıza uzayında G/Ç için yer ayrılmamıştır
- G/Ç için daha hızlı olan özel komutları kullanılabilir
- G/Ç için daha basit adres çözümleme devreleri kullanılabilir

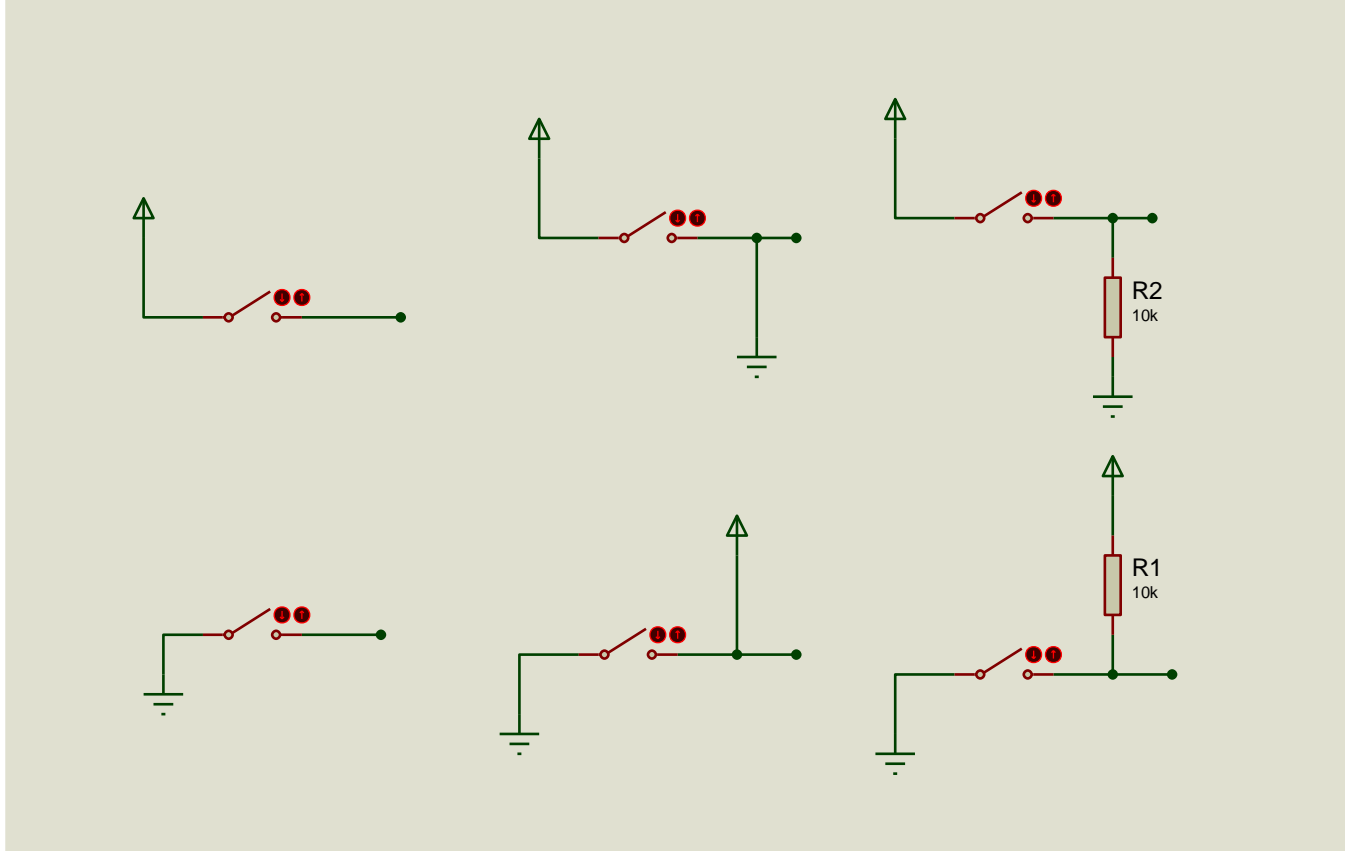
# Memory Mapped I/O

- Bazı işlemciler sadece 1 adres uzayına izin verir, Bunlarda:
  - Hafıza komutları ile G/Ç birimlerine erişilebilir
  - G/Ç birimleri için ayrı komutlara gerek yoktur
  - G/Ç – Hafıza işlemlerini ayıran fiziki uçlara gerek yoktur

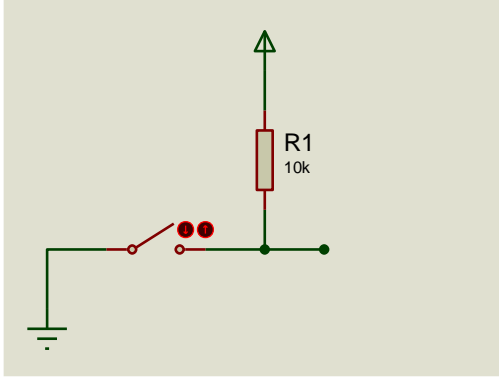
# Düğme Arayüzü (Basit Arayüz Devreleri)

- Basit giriş arayüzü olarak kullanılabilir
- Basılmadığı durumda geçerli lojik bir seviye üretmelidir
- Kontakt gürültüne karşı yazılımsal veya donanımsal önlem gereklidir

# Düğme Arayüzü



# Düğme Arayüzü – Kontakt Gürültüsü

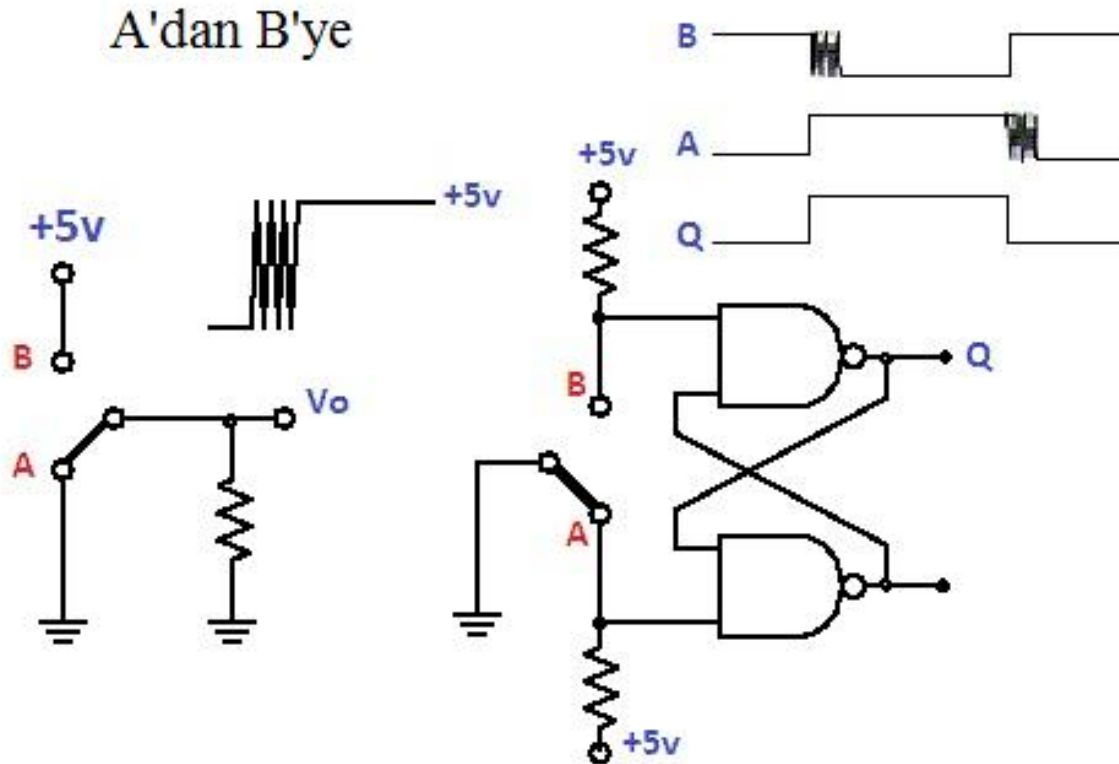


Düğme t anında kapanırsa



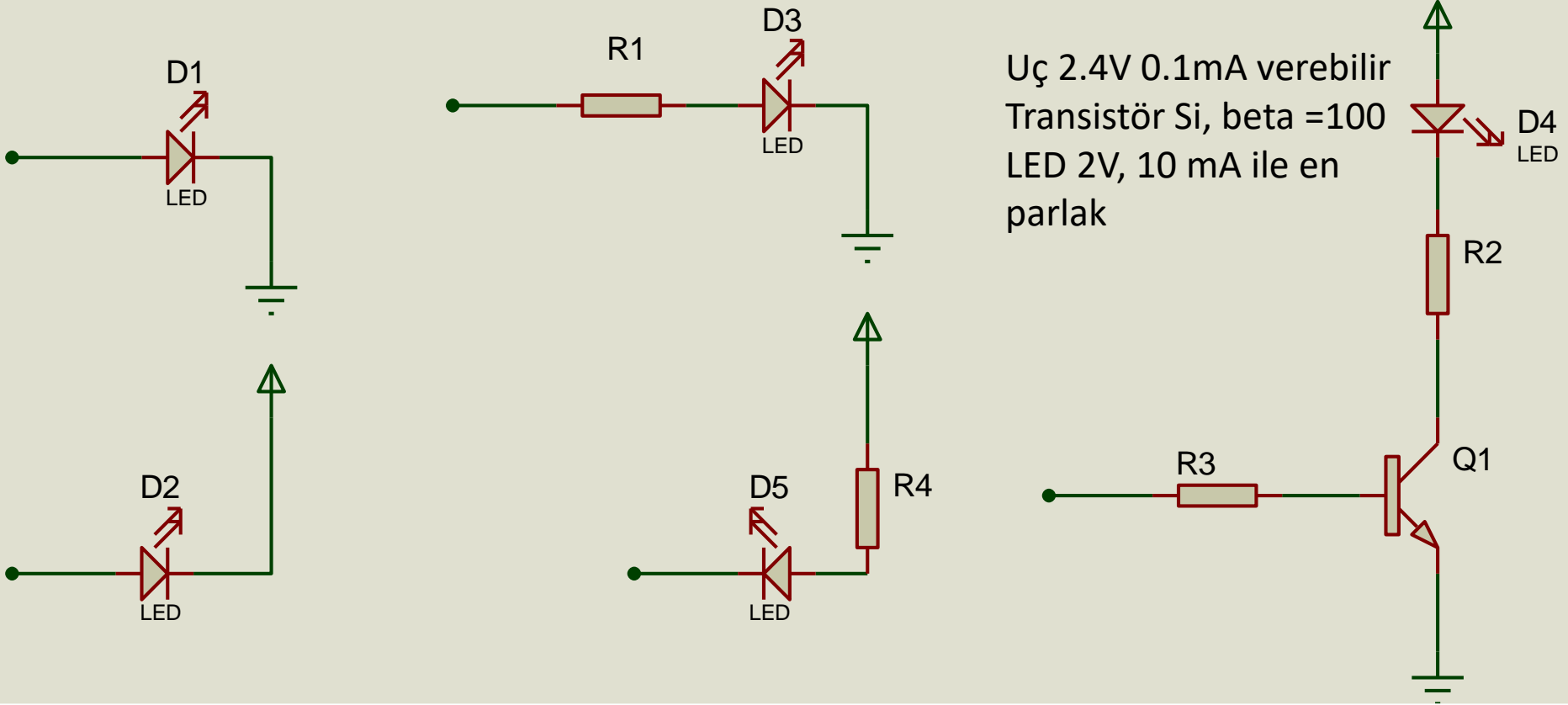
- Kontakt gürültüsünü gidermek için yazılımsal veya donanımsal çözümler uygulanmalıdır
- Yazılımsal olarak kontakt gürültüsü giderme : ilk değişim yakalandıktan sonra belirli süre aktif bekleme yapıp uç tekrar kontrol edilir

# Hardware Debounce Switch





# LED Arayüzü (Basit Arayüz Devreleri)



# Basit Çevre Birimleri için Adres Çözümleme

- Memory :  $A_{19}-A_{(i+1)}$
- Variable I/O :  $A_{15}-A_{(i+1)}$
- Fixed I/O :  $A_7-A_{(i+1)}$

Adres çözümleme devresine gitmeli

# Basit Çıkış Birimi

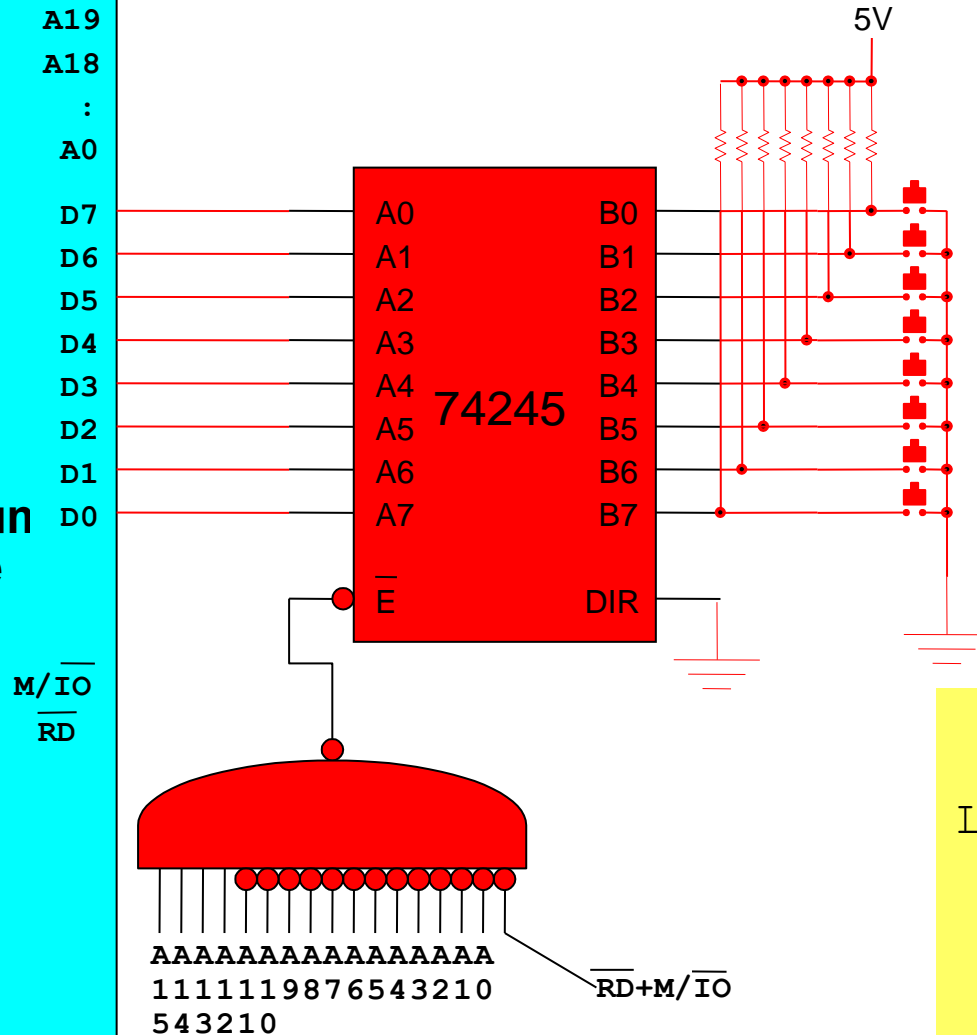
- 8 LED kullanarak 0F000H adresine yerleştirilmiş basit bir çıkış biriminin tasarlanması → donanım + adres çözümleme
- LED'lerde (on,off,on,off...) şeklinde patern oluşturma → I/O programlama



# Basit Giriş Birimi

- 8 düğme kullanarak 0F000H adresine yerleştirilmiş basit bir giriş biriminin tasarlanması → donanım + adres çözümleme
- Düğmelerin okunması → I/O programlama

# 8086 Minimun Mode



```

:
mov dx, 0F000h
in  al, dx
:

```

```

mov dx, 0F000h

L1:  in  al, dx
      cmp al, 0FFh
      je  L1
:

```

# Basit Giriş/Çıkış Birimi

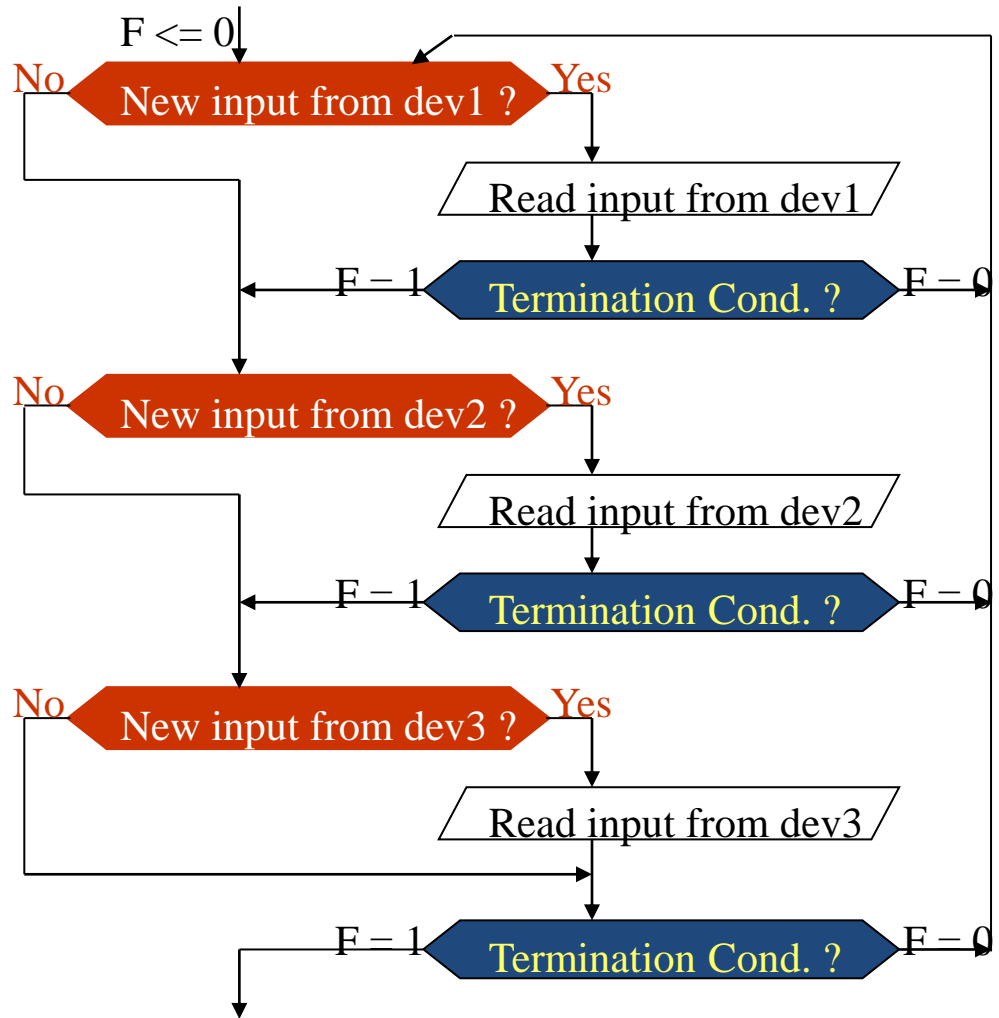
- Aynı adreste giriş ve çıkış birimi yerleştirildi → bu durum problem oluşturur mu?
- Basit giriş birimi / çıkış birimi 0F001H adresine yerleştirmek için ne yapılmalıdır?
- 0F000H adresinden itibaren 16 bitlik bir basit çıkış birimi nasıl tasarlanmalıdır?

# Çevre Birimleri Kontrolü

- Çevre birimlerinin kontrolünde
  - Polling : Çevre birim çevrim içinde sürekli control edilir, aranan durum oluşmuşsa işlem yapılır
    - Priority Polling
    - Round Robin Polling
  - Interrupt : Çevre birim ayarlanır, işlemci sürekli kontrol etmez, aranan durum oluştuğunda çevre birim işlemciyi uyarır



# Priority Polling



# Round Robin Polling

