

# MİKROİŞLEMÇİLER

ilk mikroişlemci  $\longrightarrow$  Intel 4004

- 4 bitlik mikroişlemci
- Adresleme kapasitesi  $4096 \times 4$
- 45 komut
- Sanıye 50K işlem

12 adreslenecek ucu

$$2^{12}$$

$\hookrightarrow$  Sonrasında daha yüksek frekanslı 4040 mikroişlemci geliştirildi.

Intel 8008  $\longrightarrow$  1971

- 8 bitlik mikroişlemci
- 16 kB adresleme kapasitesi
- 48 farklı komut

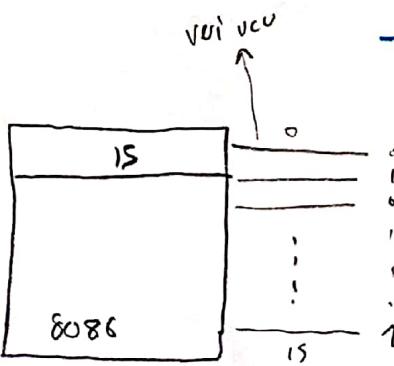
Intel 8080  $\longrightarrow$  1973

- 8 bitlik mikroişlemci
- 8080'e göre 10 kat daha hızlı
- ilk modern 8 bitlik mikroişlemci

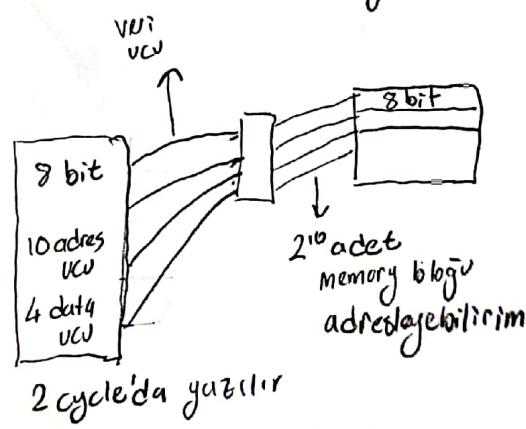
Intel 8085  $\longrightarrow$  1977

Intel 8086/8088  $\longrightarrow$  1978/1979

- 16 bitlik mikroişlemci
- Sanıye 2,5 milyon işlem
- Adreslene Kapasitesi 1MB
- 4 veya 6 bytelik komut kuyruğu
- 8086/8088 CISC (complex instruction set computer) mimarisinde
- 8086/8088: 20 adet adres ucuna sahip  $\rightarrow 2^{20} = 1M \rightarrow 2^{20} \times 8 \text{ bit}$   
1 byte
- 8086: 16 veri ucu
- 8088: 8 veri ucu



16 veri ucu  
16 bitlik mikroişlemci  
Veri tek cyclede yazılır.  
(8 olsaydı 2 cycle)



2 cycle'da yazılır

Intel 80386 → 1986

- 32 bit adres yolu
- 32 bit veri yolu  $2^{32} = 4.6 \times 10^9$
- 4 gb adreslene kapasitesi

## 8086 Yazmaları

$AX, AH, AL \rightarrow$  - AX: 16 bitlik akumülatör yazmaq

- AL, AH: 8 bitlik akumülatör yazmaq.

- Aritmetik, lojik ve veri transferi işlemlerinde kullanılır.

- Grupma ve bölme işlemlerinde gittiği operatör olarak kullanılır.

- Giriş çıkış komutlarında kullanılır.

$BX, BL, BH \rightarrow$  - BX: 16 bitlik genel anahtarı yazmaq

- BL, BH: 8 bitlik genel anahtarı yazmaq.

- Dizi şeklindeki veri erişiminde kullanılır.

$CX, CL, CH \rightarrow$  - CX: 16 bitlik genel anahtarı yazmaq.

- CL, CH: 8 bitlik genel anahtarı yazmaq.

- Tekrarlı işlemlerde tekrar sayısını sahalar. (CX)

- Öteleme ve kaydırma işlemlerinde tekrar sayısını sahalar. (CL)

$DX, DL, DH \rightarrow$  - DX: 16 bitlik genel anahtarı yazmaq.

- DL, DH: 8 bitlik genel anahtarı.

- Grupma ve bölme komutlarında böleni sayıyı oluşturmak için kullanılır.

- Giriş ve çıkış işlemlerinde port numarası sahalar.

- 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.

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

- SI** →
- SI: source index
  - Dizi komutlarında kaynak indisini tutar.
- DI** →
- DI: destination index
  - Dizi komutlarında hedef indisini tutar.

## Segment Yazıcıları

- CS → Kod segment; IP ile kullanılır. CS:IP
- DS → Data segment; BX, SI, DI ile kullanılır. DS:SI
- ES → Extra segment; DS gibi
- SS → Stack segment; BP, SP ile kullanılır.

$$\sim DS = 1280H, SI = 0045H \text{ ikilisi ile erişilen fiziki adres} ] DS:SI$$

$$1280H + 0045H = 12845H$$

IP → - instruction pointer

- Sıradaki işlenenek komutu gösterir

- CS ile birlikte kullanılır

-  $CS \times 10H + IP$

## 8086 Bayrak Yuzmacı

Carry Flag (CF) → işaretsız işlemlerde taşıma olursa 1 değerini alır.

Parity Flag (PF) → işlem sonucunda 1 olu bitlerin sayısı tek ise 0, çift ise 1 değerini alır.

Auxiliary Flag (AF) → 4 bitlik kısımların toplamaikhâma sonucu elde değerini tutur.

Zero Flag (ZF) → işlem sonucu 0 ise ZF=1 olur.

Sign Flag (SF) → işlem sonucu negatif ise SF=1 olur.

Trap Flag (TF) → Her komuttan sonra kesme oluşmasını sağlar.

Interrupt Enable Flag (IF) → Kesme kaynaklarının kesme oluşturmasına izin verir.

Direction Flag (DF) → Dizi işlemlerinde buslukta adresinde itibaren artırarak/azaltarak sıradaki gize erişimi belirler.

Overflow Flag (OF) → işaretli işlemlerde taşıma durumunda 1 değerini alır.

## Real Mode Hafıza işaretleme

- Real modda sadece 1 MB alan adreslerdir

- 8086 hafıza uzayı 1 MB (20 adres ucu → 1MB)

- Tüm bilgisayarlar ağıldığında real modda açılır.

- Real modda segment, adres ve ofset adres değerlerinin birleşimiyle hafızayı istenilen alana erişilir.

- Bir segment değeri 64 KB'lık alan gösterir.

- Ofset değeri 64 KB'lık alan içinde bir yerini gösterir.

- CS:IP, SS:SP, SS:BP, DS:BX, DS:DI, DS:SI, ES:DI

## Adresleme Modları

### Hemci Adresleme

- sabit değer atama

MOV AL, 0F2H

MOV CX, 100

MOV AH, 'A'

### Dogrular Adresleme

- Erisilecek hafıza gözünden doğrudan gösterildiği durum.

MOV BX, [1234H]

$\hookrightarrow BX \leftarrow DS : 1234H$

### Yazma Dolaylı Adresleme

- Hafıza offset değerini bir yazmaya saklar

MOV AX, [BX]

$\hookrightarrow AX \leftarrow DS : BX$

### Base+Index Adresleme

MOV DX, [BX+DI]

### Yazma GÖRELİ Adresleme

MOV AX, [BX+100H]

## Yığın Adresleme Modları

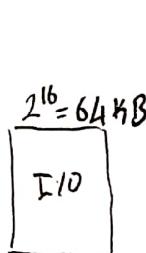
- SP yazmacı programının taşımadığı yığının genişliğini göstererek şekilde ilk değer alır.
- Her PUSH işleminde SP-1 ve SP-2 adreslerine 2 byte veri yazılır ve SP değeri 2 azaltılır.
- Her POP işleminde SP+1 ve SP+2 adreslerinden 2 byte veri okunur ve SP değeri 2 arttırılır.

# 8086 Üç Tanimlari

adres ucu, data ucu, sinyaller

bunu

halkacagiz



1MB  
Memory

Pin Number		Maximum mode	Minimum mode
GND	1	Vcc	
AD14		AD15	
AD13		A16, S3	
AD12		A17, S4	
AD11		A18, S5	
AD10		A19, S6	
AD9		/BHE, S7 → Bus high enable	
AD8		MN, /MX → minimum mode, maximum mode select	
AD7		/RD → Data read	
AD6		/RQ, /GTO	HOLD
AD5		/RD1, /GT1	/HLDA → Hold acknowledge
AD4		/LOCK	/WR → Data write
AD3		/S2	M, /IO → Memory or IO?
AD2		/SI	DT/R → Data transmit, /Receive
AD1		/SO	/DEN → Data enable
AD0		QSO	ALE → Address Latch Enable
		S81	/INTA → interrupt acknowledge
		/TEST	
		READY → Stabil clock gelene kadar 0	gevre birimi istegini deguledecegim
	20	RESET → mikroişlemci reset	

System

non-maskable interrupt  
 $A0 == /BLE$   
interrupt request  
gevre birimi istegi

CLK

GND

ALE → AD uclarinda adres degeri mevcut iken ALE=1

Adres ucu sadece gitig yonlu

/DEN → AD uclarinda data degeri mevcut iken /DEN=0

data ucu hem gitig

hem giriş

M, /IO → MOV AX, [100H] ; M/IO=1

DT/R

IN AX, DX ; M/IO=0

+ AD<sub>15</sub> - AD<sub>0</sub>: (I/O-3)

Address - data ugları. ALE ucu 1 ise adres bitlerini tutuyor.  
/DEN ucu sıfır ise data bitlerini tutuyor.

+ A<sub>19</sub>/S6 - A<sub>16</sub>/S<sub>3</sub>: (0-3)

Adres - status ugları.

+ /RD : (0-3)

/RD ucu 0 olduğunda, data ugları memory'de veya I/O cihazlarından veri alıcı konumda olur.

+ READY : (I)

Bu giriş, mikro işlemecinin żemanturasına beklenme durumunu eklemek için kontrol edilir.

- READY=0 → bekler ve boşta kalır
- READY=1 → mikro işlemecinin çalışmasına karışmaz

+ NMI : (I)

Non-maskable interrupt ucu. NMI interrupt flag 1 mi diye kontrol edilir.

Önemli, kritik kaynaklardan gelen interruptlar için kullanılır.

+ RESET : (I)

Mikro işlemeyi sıfırlıyor.

CS = FFFF0H , IP = 0000H , IF = 0

+ MN/MX : (I)

Minimum-Maximum modu pin seçimi.

+ /BHE, S<sub>7</sub> : (0-3)

Veri yolunun yüksek katmanlı 8 bitinde katlandığı ve olduğunu gösteriyor.

## + M/IOS (0-3)

Yapılan işlemin memory işlemi mi I/O işlemi mi olduğunu gösteriyor.  
Bizim adıştırdığımız komuta bakanlık yapıyor.

## + WR : (0-3)

Memory'ye veya I/O cihazına veri aktığı, veya 0 olur.

## + INTA : (0-3)

Interrupt acknowledge. INTR ucuna gelen donet gibi aktarır.  
Gelen kesme isteklerine göre örtülmeyeceği bilgisi veriyor.

## + ALE : (0)

AD ucunun adres bilgisi tuttuğunu gösteriyor.  
Memory adresi olabilir veya I/O port numarası olabilir.

## + DT, IR : (0-3)

Data bus veri alıyor mu veriyor mu bilgisini gösterir.  
Veri ayırtırma yardımcı derrelerine bağlı.

## + DEN : (0-3)

Data bus enable. AD ucunda data bilgisi olduğunu gösterir.

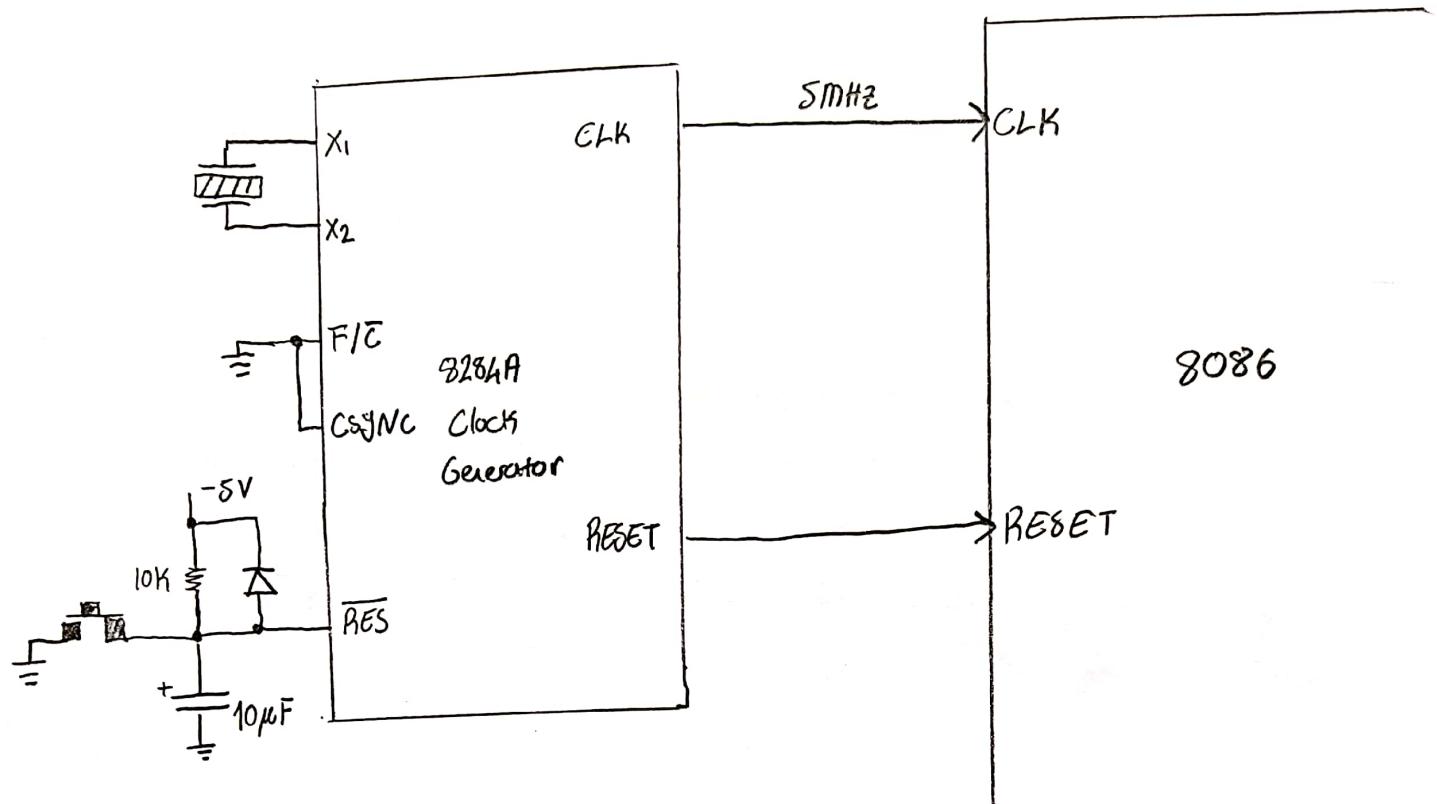
## + HOLD : (I)

Hold inputu DMA (direct memory access) ister. Eğer HOLD=1 ise mikroişlemci  
durur ve kendisi adresini, verisini bir yere yelektirir.

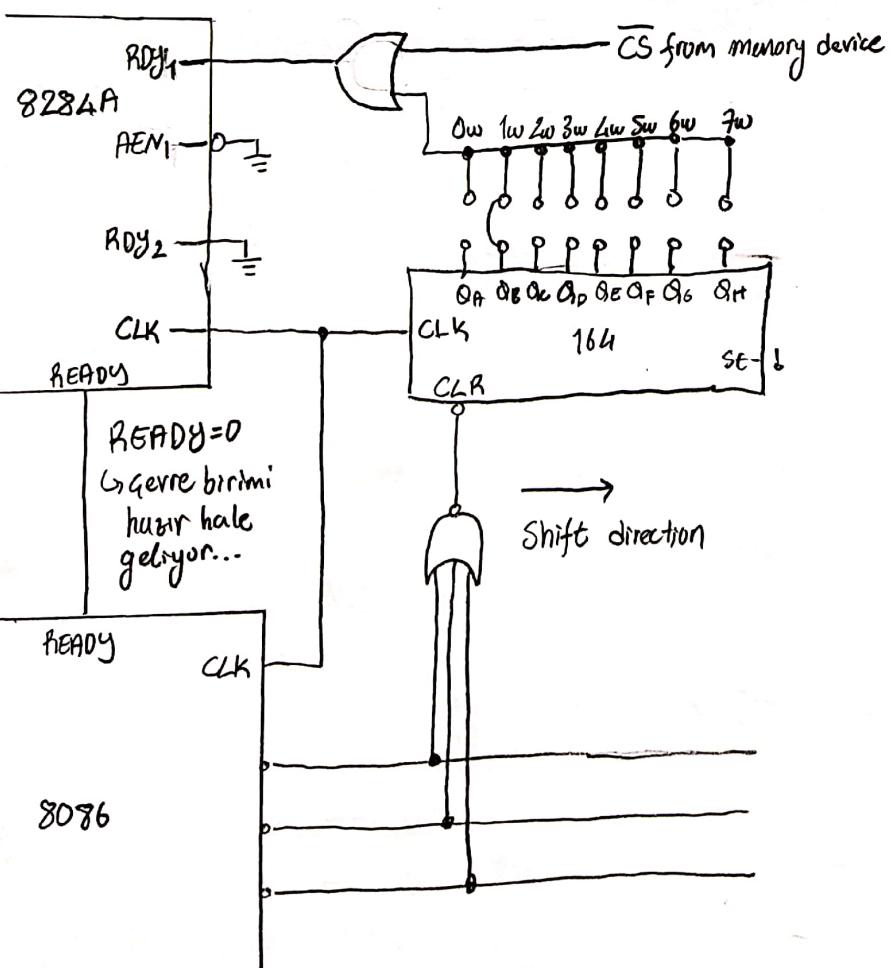
## + HLDA : (0)

8086 mikroişlemcisinin hold durumunda olmadığını gösterir.

# Clock Generator (8284A)



→ mikroişlemeyi hızlandırmak için



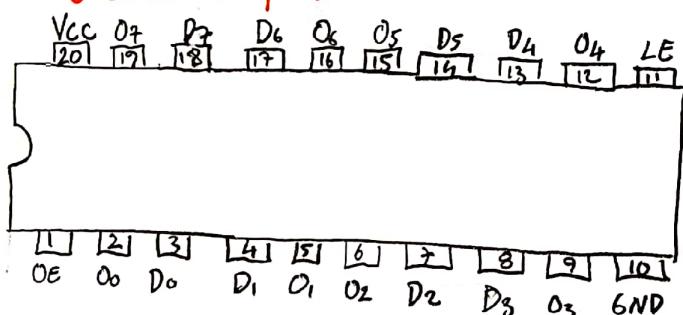
→ yavaş çevre birimlerinde  
ve okuma yazma olmak üzere  
mikroişlecinin bekletilmesinde  
kullaniliyor.

# Bus Buffering & Latching

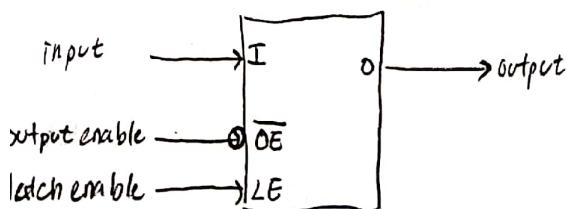
- Örneğin write veya read mikroişlemcilerde aktıyor ve button genel birimleme bağlıyor. Dolayısıyla bu ucların uygun elektriksel seviyelerde kalması gerekiyor. Buffer devreleriyle data ucan yollar boyunca stabil kalmasını sağlıyoruz.
- Latching ise, zaman gerekliliği olarak (AD) adres ucları t1 anda adres değeri varken t2 anda data değeri oluyor. t1 anda adres değerleri bir latching devresine geçirilerek tutuluyor, button komutun aktaması surende sisteme bu adres değerleri sağlanıyor. Aynı t2 anda data ıain de gerekli. AD ucları ne kadar değişirse değişim bunlar tutulup sisteme sağlanır (komut yürütülmesi bitene kadar.).

## 74LS373 Octal Transparent Latch

With 3-state Outputs



8 bitlik tutucu.



LS373			
D <sub>n</sub>	LE	OE	O <sub>n</sub>
H	H	L	H
L	H	L	L
X	L	L	O <sub>o</sub>
X	X	H	Z*

OE=0, LE=1  
OE=0, LE=1  
OE=1

H = High voltage

L = Low voltage

X = Immaterial

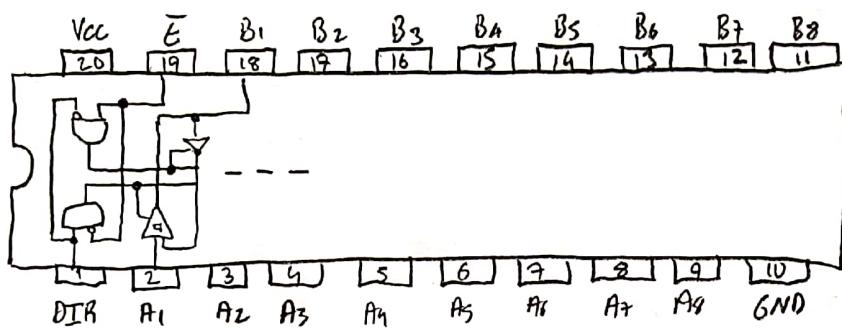
Z = High impedance

OE=1 olursa entegre disable  $\rightarrow$  input ile output aktarımı engellenir. İfadesi.

OE=0, LE=1 olursa (giriş neyese çıkışa kopyala, LE=0 oluna kadar)

Örnek bağlantı
AD — Input
ALE — LE
OE — O
Output == adres değeri

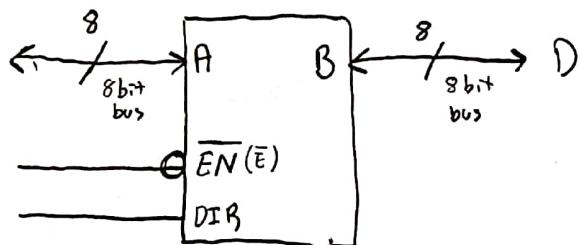
## 74LS245 Octal Bus Tranceiver



INPUTS		OUTPUT
E	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	isolation

H = High Voltage level  
L = Low Voltage level  
X = immaterial

Transmit/Receive 8 bitlik entegre



EN=1 iken A ve B ukları yüksek impedanslı birbirinden koparılmış gibi davranışlıyor.

EN=0, DIR=0 iken B'yi A'ya kopyalıyor.

EN=0, DIR=1 iken A'yi B'ye kopyalıyor.

→ Vur aymak için kullanılır.

Örnek bağlantı

AD—A

Githiq tarafında  
↓ saat döşenme

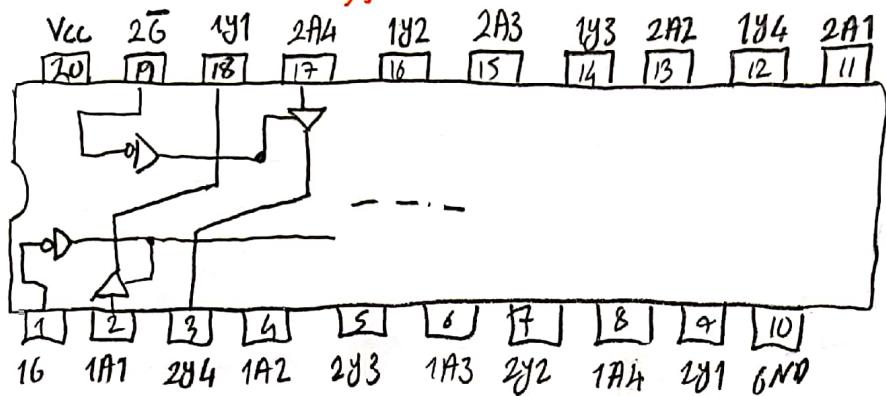
D—B

DT<sub>i/R</sub>—DIR

DT=0 veri gönderme  
R=0 vur okuma

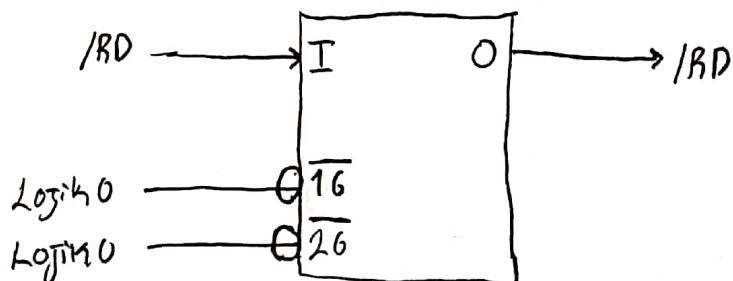
/DEN—/E

## 74LS244 Octal Buffer



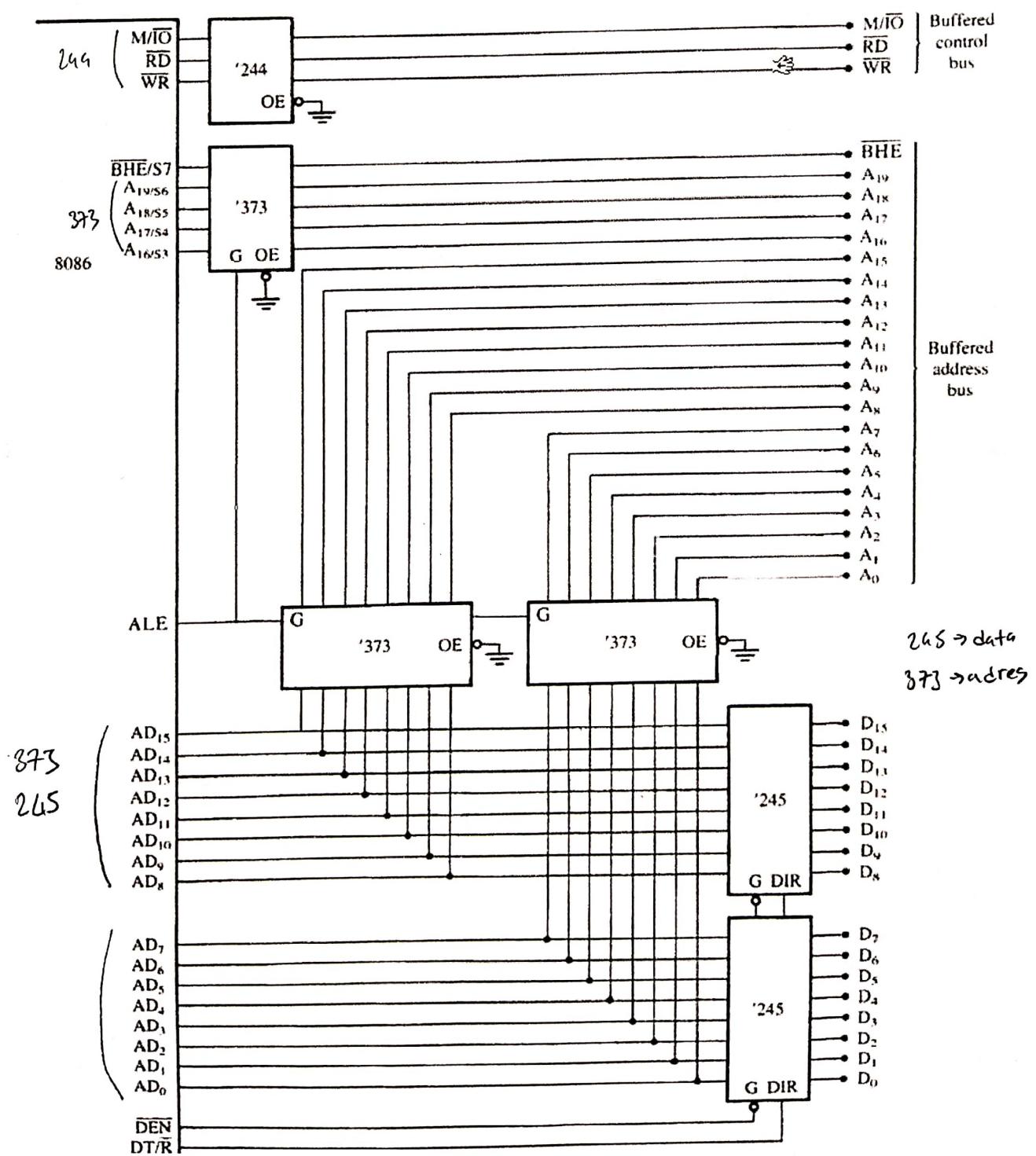
INPUTS	OUTPUT
16,26	D
L	L
L	H
H	X
	(?)

Veri tekrarlayıcı (Kuvvetlendirme)

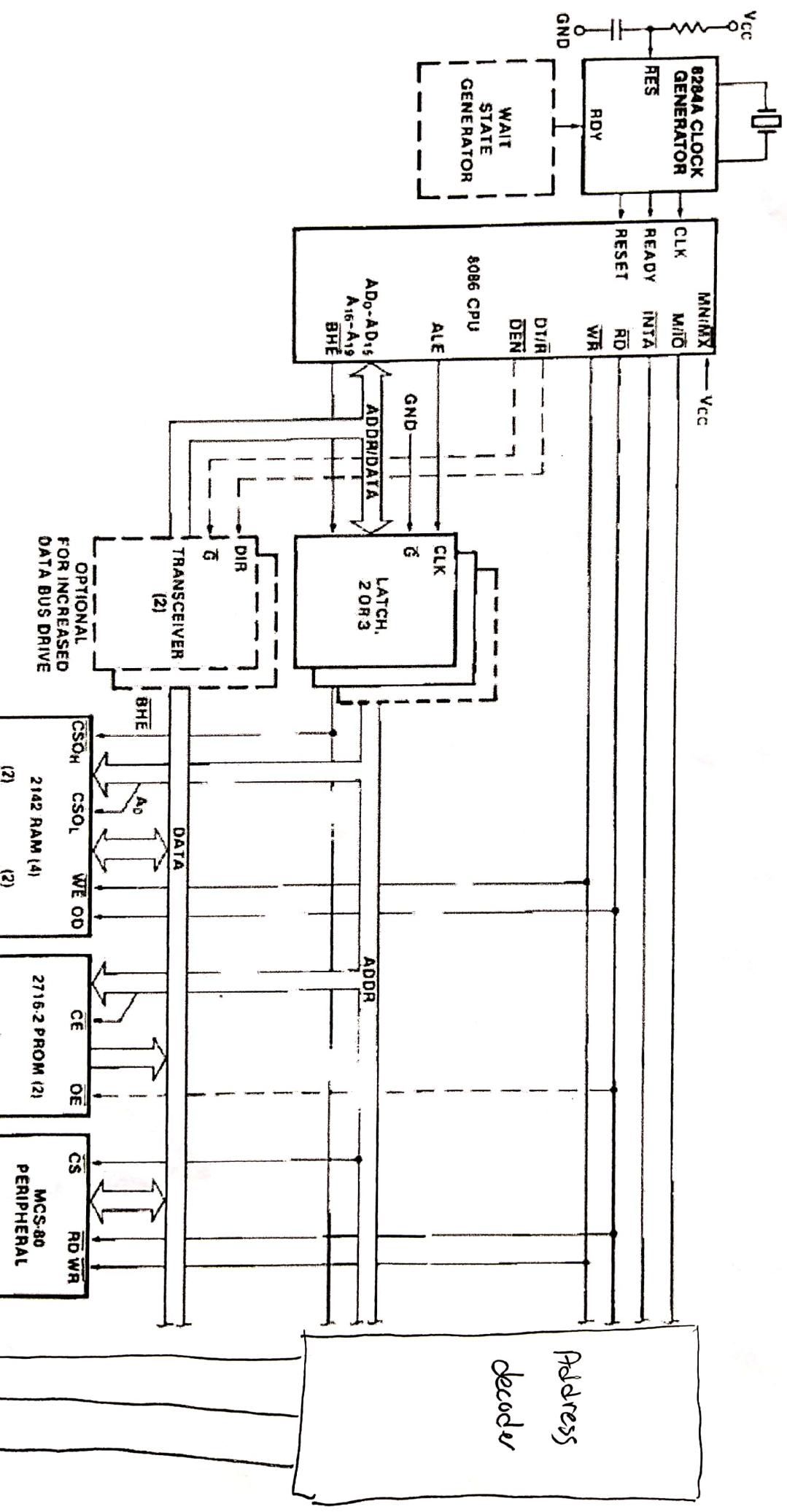


Bu yardımcı derre fiziki uglaan  
akislarinda 8086'ya gore daha  
yuksek miktarlarda akim ile bu aglari  
surebilirsin. orn 8086 = 300mA  
2A4 = 500mA

# Bus Buffering & Latching



# BUS Buffering & Latching



## Gevre Birimleri (Peripheral Devices)

Makine kodu: 00100000 00000000 00000000 00000000

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

Hafıza birimine bazer şekilde çıkış birimlerine  
vei yazılabilir → OUT

- Giriş Birimi:

Hafıza birimiyle bazer şekilde giriş birimlerinden  
vei okunabilir → IN

## Giriş/Çıkış Assembly Komutları

p8 → port adresi

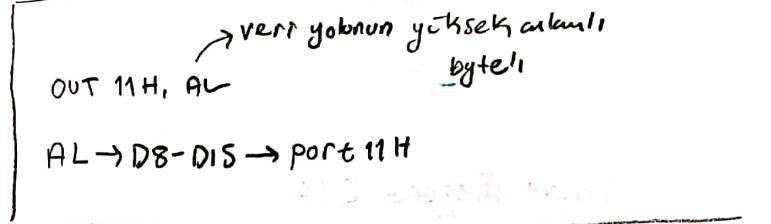
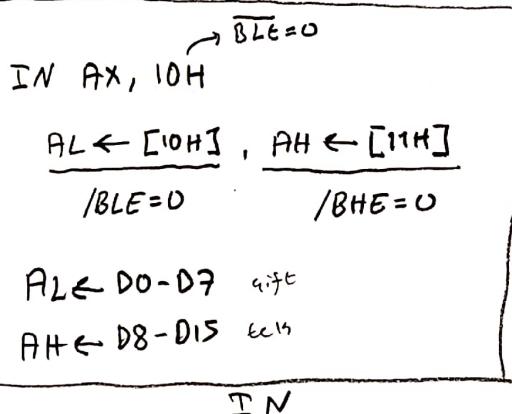
	IN Komutu	OUT Komutu
Fixed Address	IN AH, P8	OUT P8, AL
	IN AX, P8	OUT P8, AX
Variable Address	IN AL, DX	OUT DX, AL
	IN AX, DX	OUT DX, AX

port adresi 8 bite olduğuyorsa  
DX üzerinden port tanımlanır.

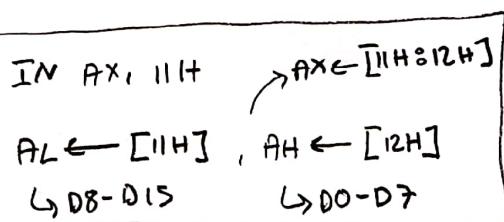
Data nerede? → AL veya AX

port çiftice düşük  
okunur

port tekise yüksek  
okunur



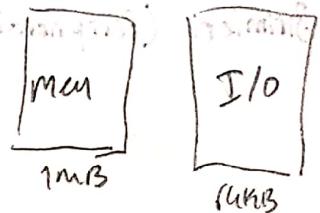
IN AL, 10H  
port 10H → D0-D7



## Giriş / çıkış Haritalama Yöntemleri

### - Isolated I/O Mapping:

M. / IO von adres gösterimleme için kullanılır.



Hafıza uzayıyla  
I/O uzayı ayıri

Mem  
1MB

I/O  
1KB

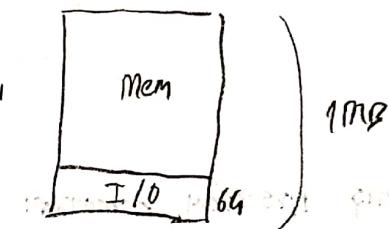
### - Memory Mapped I/O:

M. / IO von adres gösterimleme için kullanılmaz.

MOV [IOH], AL  
OUT IOH, AL

MOV komutlarıyla da  
I/O işlemleri yapabilmiz.

Hafıza uzayı  
ile I/O uzayı ayıri  
değil



### Isolated I/O:

- Hafıza uzayının tamamı hafıza birimleri için kullanılabilir.

    → Hafıza uzayında I/O için yer ayrılmamıştır.

- I/O için daha hızlı olur özel komutları kullanılabilir.

- I/O için daha basit adres gösterimleme devreleri kullanılabilir.

### Memory Mapped I/O:

- Sadece 1 adres uzayına izin verilir.

- Hafıza komutları ile I/O birimlerine erişilebilir.

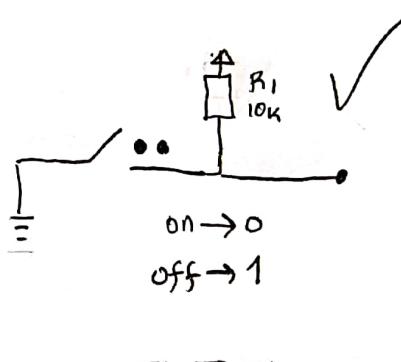
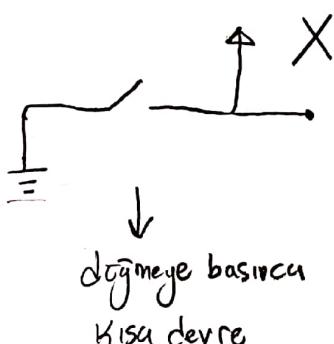
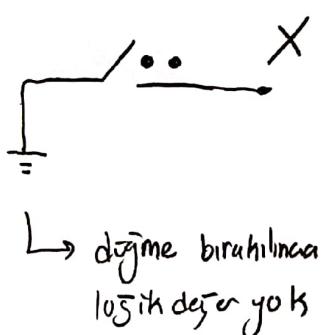
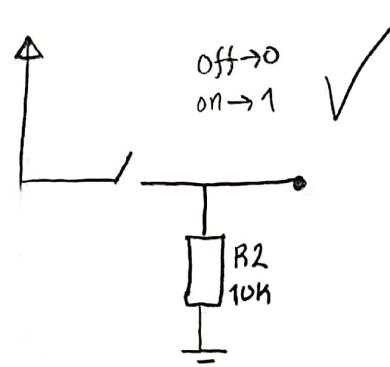
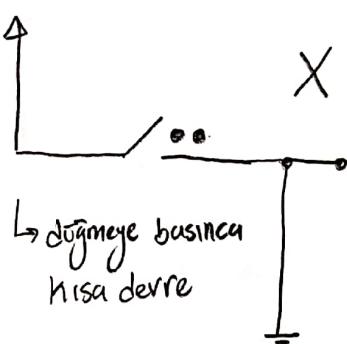
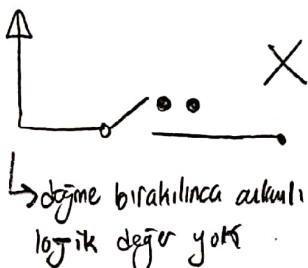
- I/O birimleri için ayrı komutlara gerek yoktur.

- I/O - Hafıza işlemini ayıran fiziki valarca gerek yoktur.

## Düğme Arayüzü (Basit Arayüz Devresi)

SGM74HC164

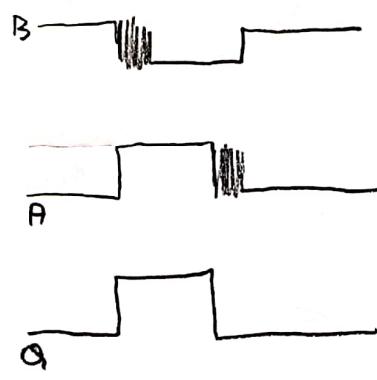
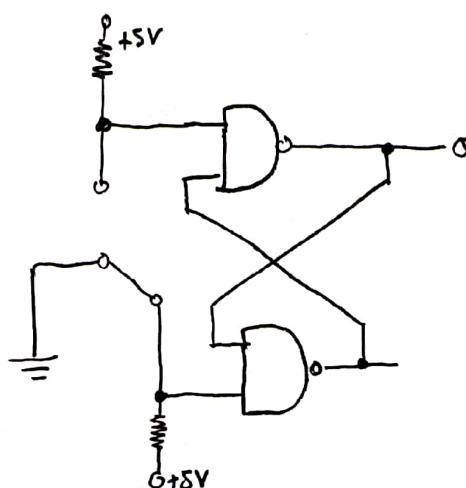
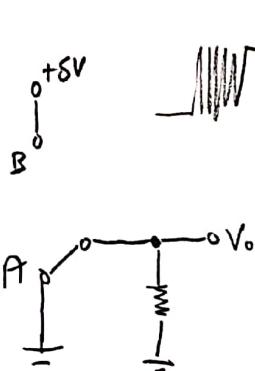
- Basit giriş arayüzü olarak kullanılabilir.
- Basılmadığı durumda geçerli lojik bir seviye iretmelidir.
- Kontakt görünübüne karşı yazılımsal veya donanımsal önleme gereklidir.
- Düğmeye basınca / bırakınca besleneyenle ground kısa devre olmamalı.



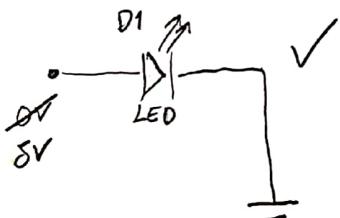
- Yazılımsal olarak kontakt görünübü gidermes ilk değişim yakalandıktan sonra belirli süre aktif bekleme yapıp ve tekrar kontrol edilir.

düğme t anında kapanırsa

### Hardware Debounce Switch



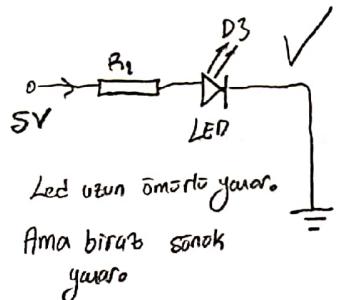
## LED Arayüzü



İşik verir ama omzunu  
kısa olur.

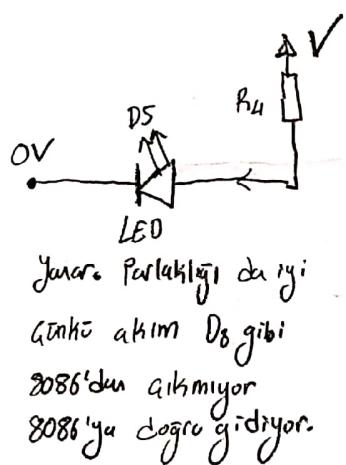
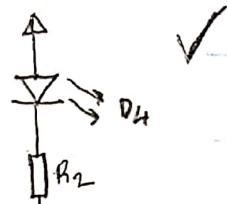


Yanmaz



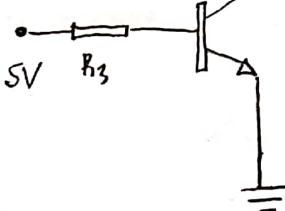
Led ızın omzunu yanar.  
Ama biraz sönük  
yanar.

$U_a = 2.4V$   $I_a = 10mA$   
Yazabilir. Si transistör  
 $\beta = 160$  LED 2V,  
10mA ile en parlak



Yanar. Parlaklığını da iyisi  
gördü akım D8 gibi  
2086'dan akışmıyor  
2086'ya doğru gitmektedir.

Dogrudan 8086'dan sürerken akımı.  
Yanar. Uzun omzunu parlak.



## Busit Çevre Birimleri için Adres Gözümlene

Memory  $\rightarrow$  A19 - A(0+1)

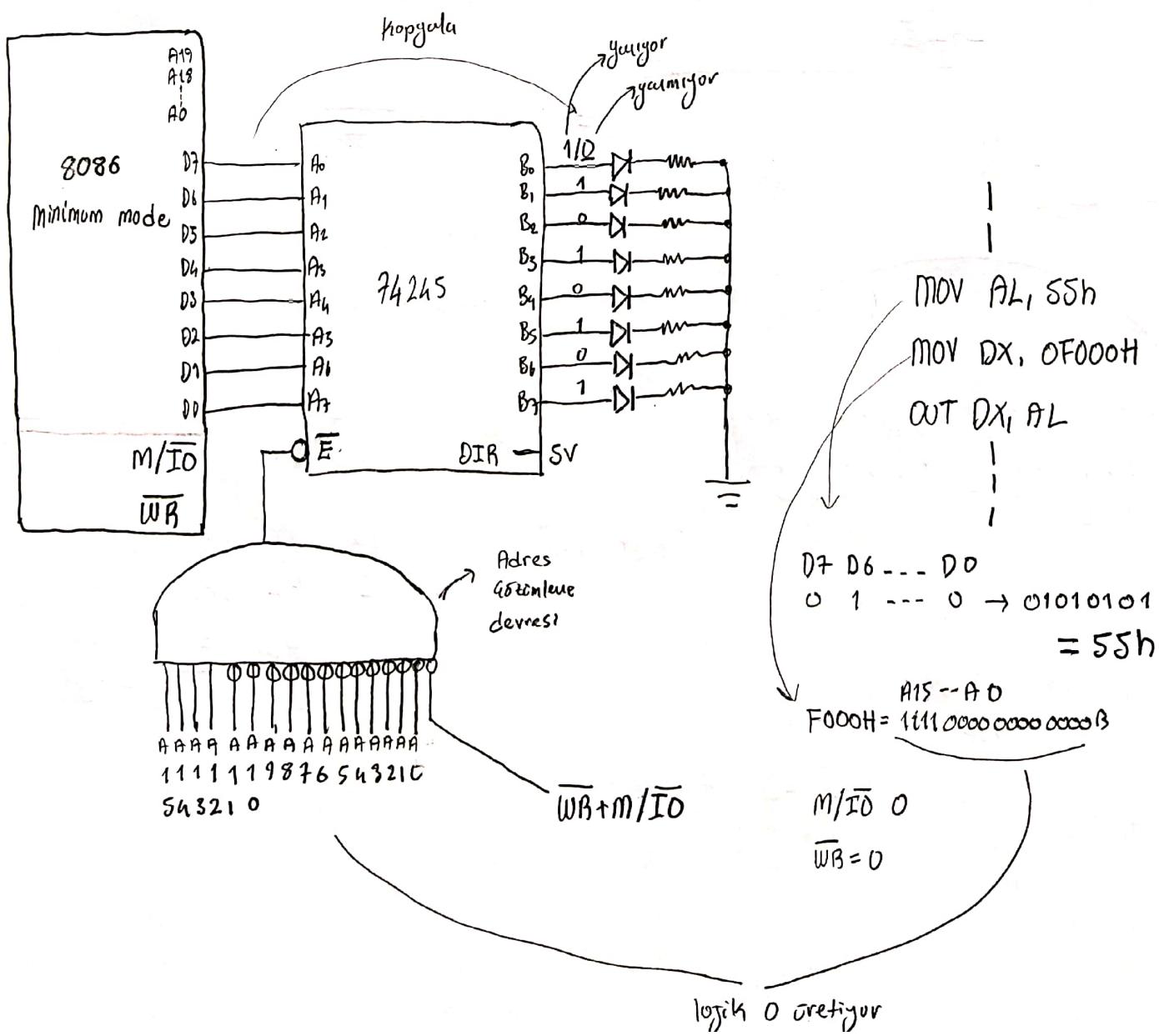
Variable I/O  $\rightarrow$  A15 - A(0+1)

Fixed I/O  $\rightarrow$  A7 - A(0+1)

adres çözümlene devresine gitmeli.

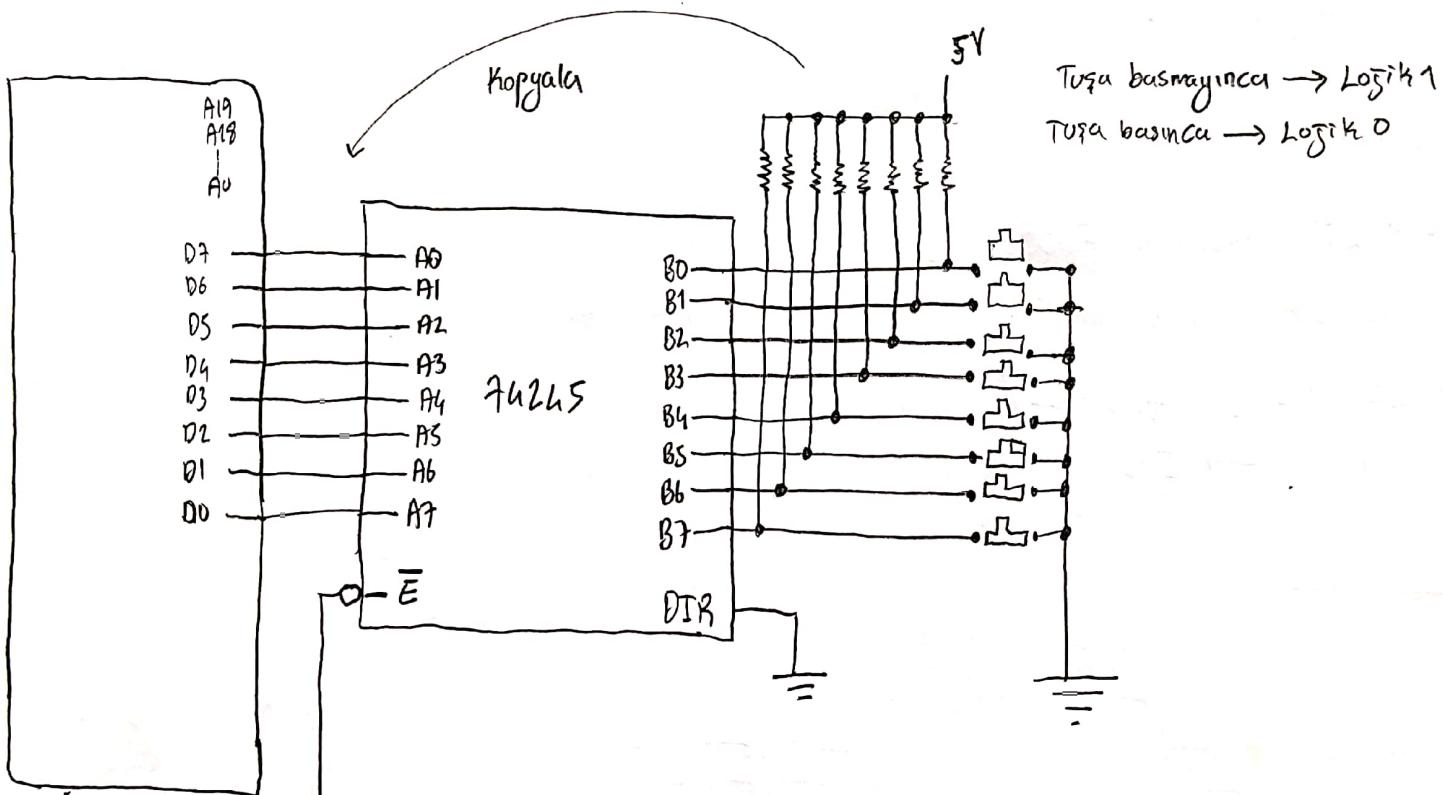
## Busit Giriş Birimi

- 8 LED Kullanarak F000H adresine yerleştirilmiş busit bir giriş biriminin tasarılaması  $\rightarrow$  donanım + adres çözümlme
- LED'lerde (on, off, on, off) şeklinde pattern oluşturma  $\rightarrow$  I/O programlama



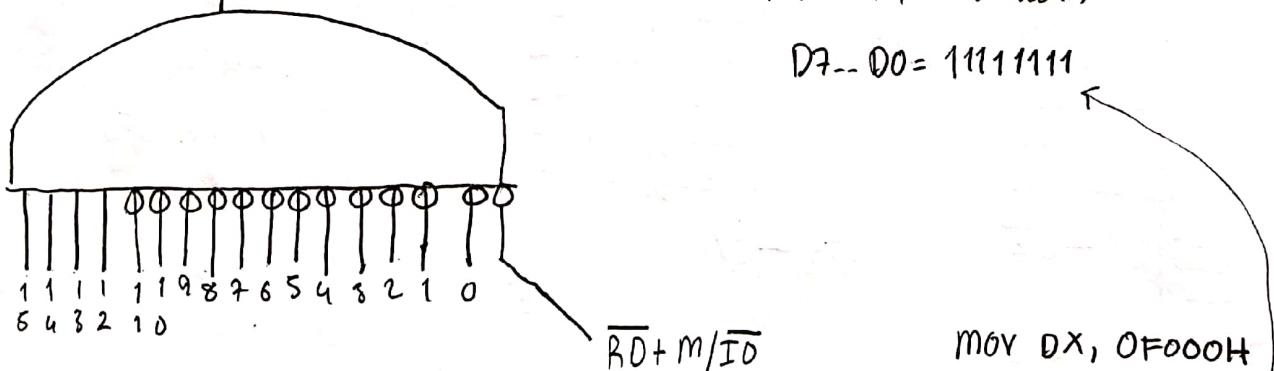
## Basit Giriş Birimi

- 8 doğme kullanarak OF000H adresine yerleştirilmiş  
basit bir giriş biriminin fonsiyonları → donanım adres gottenme
- Doğmelerin okunması → I/O programlama



Habir tuşa basmazsa

$$D7 \dots D0 = 11111111$$



MOV DX, 0F000H

L1: IN AL, DX,

CMP AL, OFFH

JNE L1

Tuşa basılmadığında  
zaman boş döngüde

dönüyor

$$\begin{array}{ll}
 A15=1 & A10=0 \\
 A14=1 & A9=0 \\
 | & | \\
 A11=1 & A0=0
 \end{array}
 \quad \overline{RD} = 0 \quad M/\overline{IO} = 0$$

## Basit Giriş Çıkış Birimi

Hem OF000H, hem OF001H

↳ 2'nci adres gözetleyici

- Aynı adreste giriş ve çıkış birimi yerleştirildi.  
→ bu durum problem oluşturur mu? Hayır.
- Basit I/O birimi OF001H adresine yerlestirmek için ne yapılmalıdır? (Data değerleri D8-D15)
- OF000H adresinde ifade edilen 16 bitlik bir basit çıkış birimi nasıl tasarlanmalıdır?

F001H

↳ 1) D8-D15

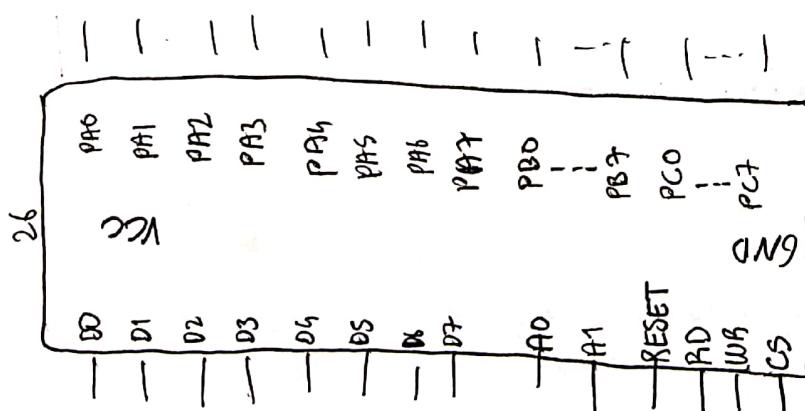
2) A0=1 ile adres eşit

veya

/BHE=0 ile adres eşit

## 8255 PPI

- PPI: programmable peripheral interface
- TTL uyumluluğu herhangi I/O cihazı μP ile bağlanabilirliğinin kullanılabilirliği.
- 3 adet genel anahtarı portu vardır
- 8-bit veri yolu genişliğine sahiptir.



- CS: 8255 enable (adres gözetimine devresinin aktifisi — CS)
- RD: μP'in 8255'ten okuma yapmasını sağlar (μP RD — RD)
- WR: μP'in 8255'e yazma yapmasını sağlar (μP WR — WR)
- A0 ve A1: Port seçim vaları

$A_1$	$A_0$	$\overline{RD}$	$\overline{WR}$	$CS$	İşlem	Okuma	Yazma
0	0	0	1	0	Port A $\rightarrow$ Data Bus	1	0
0	1	0	1	0	Port B $\rightarrow$ Data bus	1	0
1	0	0	1	0	Port C $\rightarrow$ Data bus	1	0
1	1	0	1	0	Control Word $\rightarrow$ Data bus	1	0
0	0	1	0	0	Data bus $\rightarrow$ Port A	0	1
0	1	1	0	0	Data bus $\rightarrow$ Port B	0	1
1	0	1	0	0	Data bus $\rightarrow$ Port C	0	1
1	1	1	0	0	Data bus $\rightarrow$ Control word	0	1

Port A, B, C

**Port A:** 8 bit output latch/buffer ve 8 bit input latch.  
8 bit giriş veya 8 bit çıkış olursa kullanılır.

**Port B:** 8 bit output latch/buffer ve 8 bit input latch.  
8 bit giriş veya 8 bit çıkış olursa kullanılır.

**Port C:** 8 bit output latch/buffer ve 8 bit input buffer.  
PCL ve PCH'ler 4'er bit I/O için programlanabilir.  
Mod1 ve Mod2'de kontrol ve durum belirler.

## 16 Bitlik Veriyolunda 8-bitlik Geniş Bittişmeleri

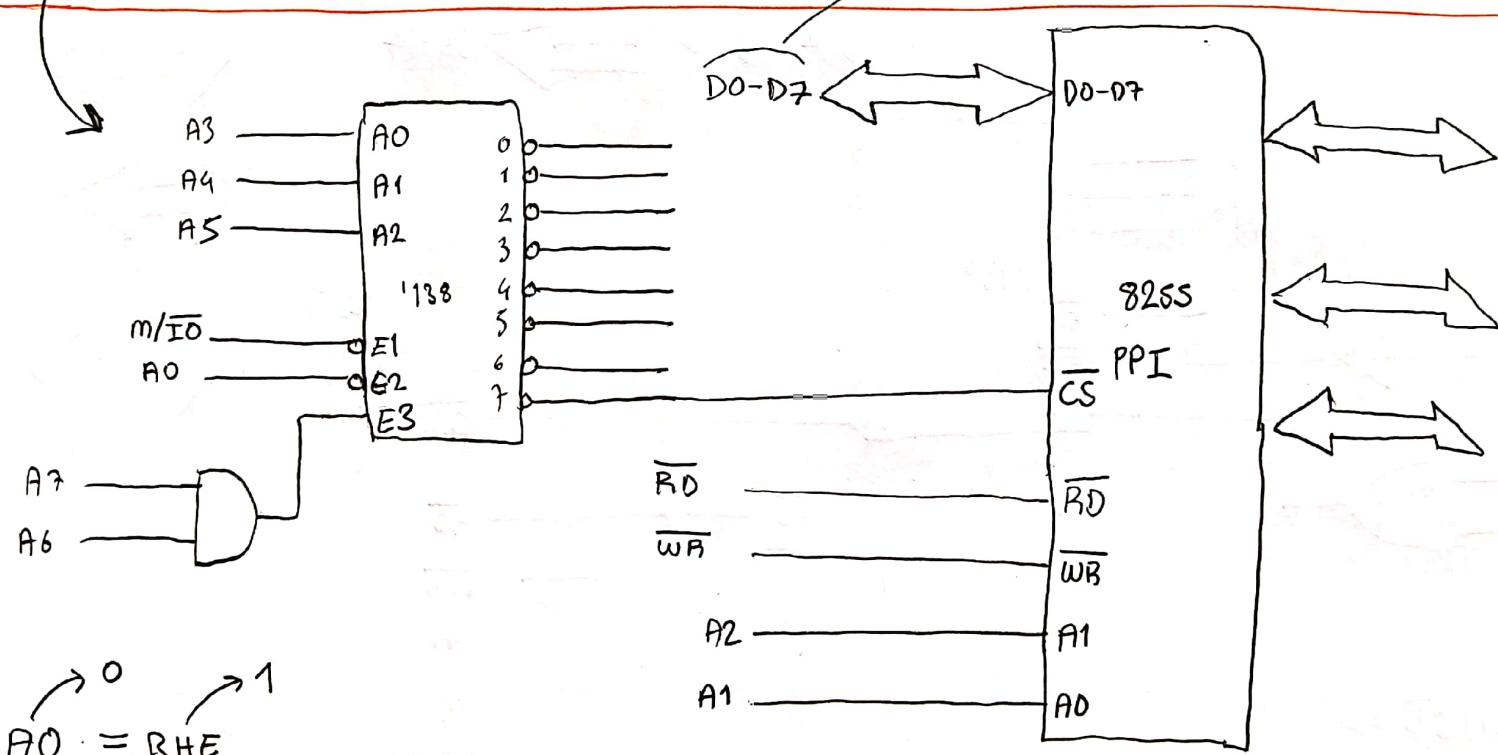
$B+E$	$A_0$	Transfer
0	0	Anlamlı değil.
0	1	Tek adresleki byte D8-D15
1	0	Cift adresleki byte D0-D7
1	1	Anlamlı değil.

## Example

8255 paralel interface'ini var. Bu cihazı OF8H adresinde, itibaren  
adreslik çift adreslere yuleşirmek istiyoruz. Adres aitonlamine devreniz  
nasıl olmalıdır?

	A7	A6	A5	A4	A3	A2	A1	A0
OF8H	1	1	1	1	1	0	0	0
OF9H	1	1	1	1	1	0	1	0
OFCH	1	1	1	1	1	1	0	0
OFEH	1	1	1	1	1	1	1	0

→ Döşkoulamlı byte: çift adres



OUT OF8H, AL / IN AL, OF8H

OF8H → PORT A

OF9H → PORT B

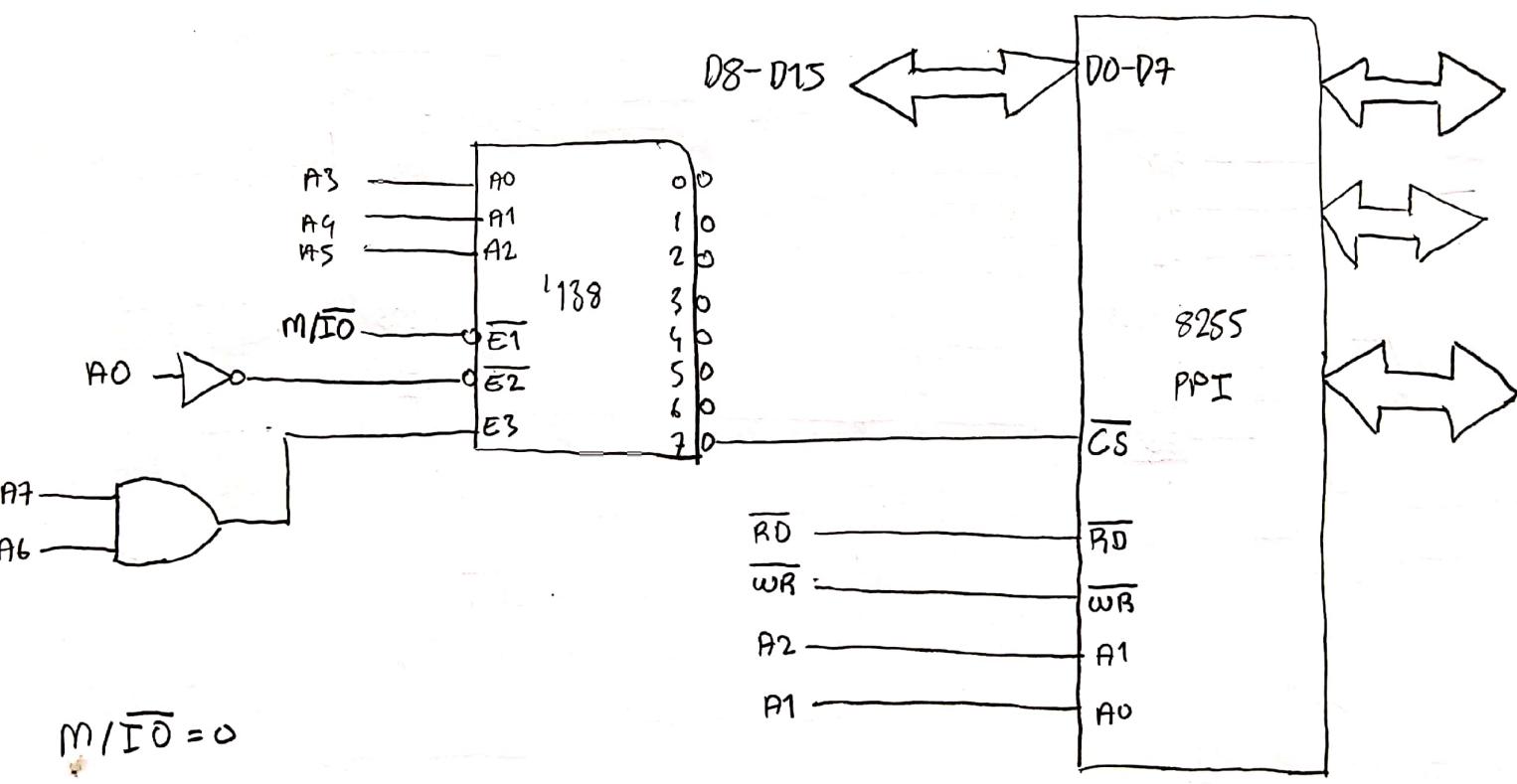
OFCH → PORT C

OFEH → CONTROL

## Örnek

8255 parallel interface limitiz var. Bu cihazı OF9H adresinden itibaren ardışık tek adreslere yerleştirmek istiyoruz. Adres gözetleme devreniz nasıl olmalıdır?

	A7	A6	A5	A4	A3	A2	A1	A0
OF9H	1	1	1	1	1	0	0	1
OFB <sub>H</sub>	1	1	1	1	1	0	1	1
OFDH	1	1	1	1	1	1	0	1
OFFH	1	1	1	1	1	1	1	1



$$M/I\bar{O} = 0$$

$$\overline{BHE} = 0$$

OUT OF9H, AL / IN AL, OFBH

OF9H → PORT A

OFBH → PORT B

OFDH → PORT C

OFFH → CONTROL

## 8255 Mod Seçimi

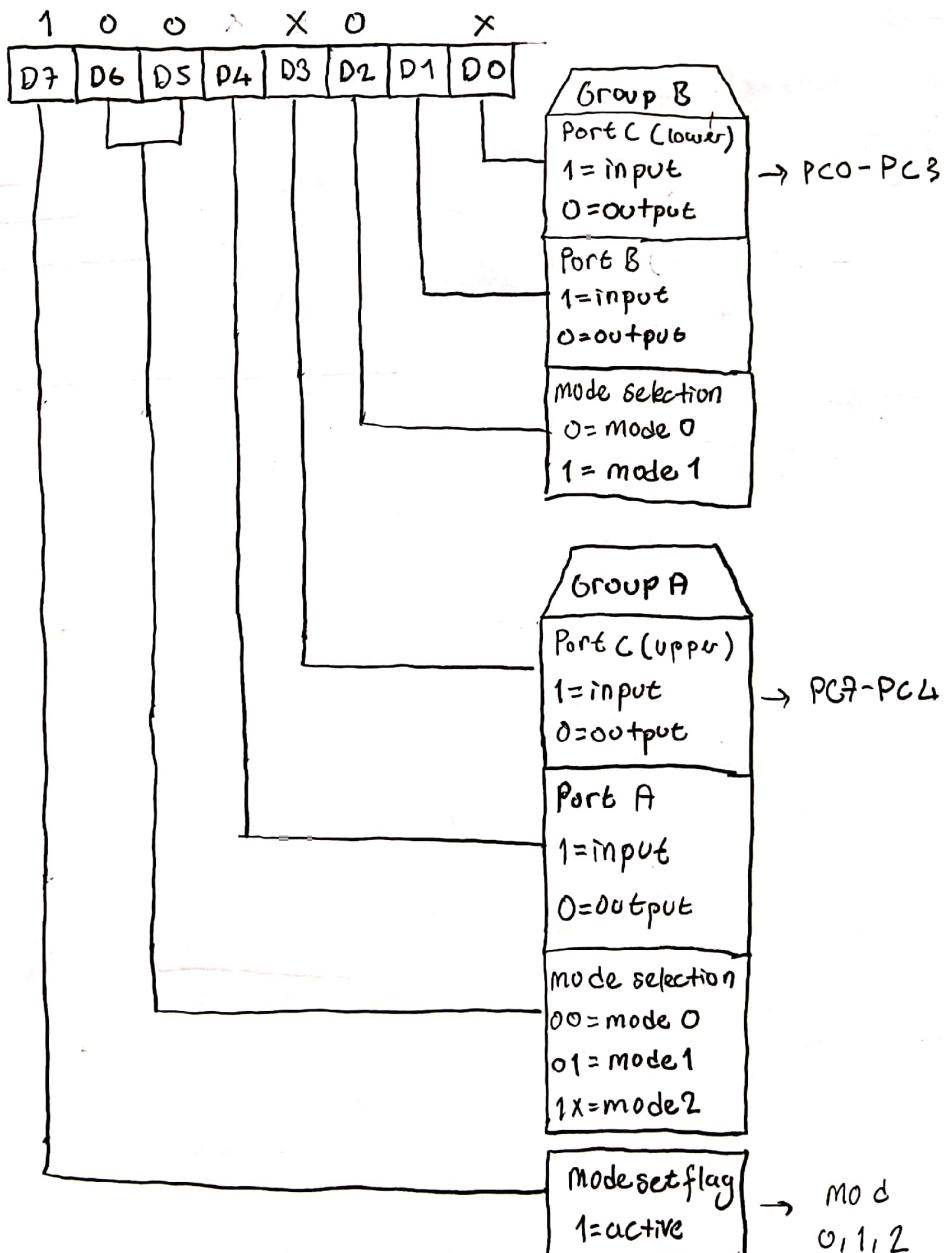
- 8255, 4 farklı mod ile çalışabilir:

— Mod 0: Basic I/O → input, output

— Bit Set Reset Mod

— Mod 1: Strobed I/O → handshaking

— Mod 2: Bi-directional Bus



8255

Kontrol Yatmacı →  
Mod 0

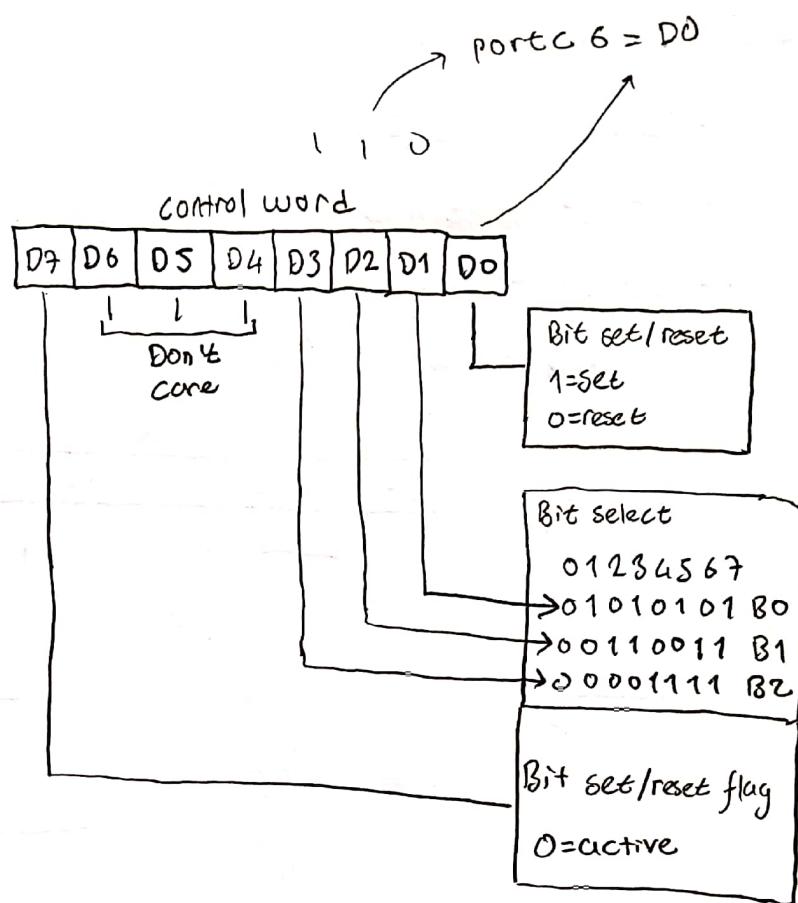
8255 }  
A1=1 } → control word  
A0=1 }

A → LED, output=0

A → button, input=1

PA, PB, PCL, PCH

Mod 0 için 2<sup>4</sup> farklı konfigürasyonda



8255

Kontrol Yazmacı  $\Rightarrow$

BSR

- $\rightarrow$  PC her bir pinini ayrı ayrı  
out 1,0
- $\rightarrow$  mod1, mod2 handshaking  
config bit