

# MIKROİŞLEMÇİLER

ilk mikroislemci  $\longrightarrow$  Intel 4004

- 4 bitlik mikroislemci
- Adresleme kapasitesi  $4096 \times 4$
- 45 komut
- Santride 50k işlem

12 adreslenecek ucu

$$2^{12}$$

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

Intel 8008  $\longrightarrow$  1971

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

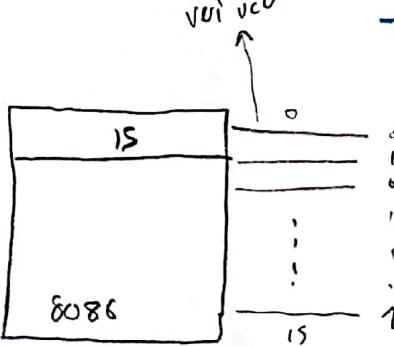
Intel 8080  $\longrightarrow$  1973

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

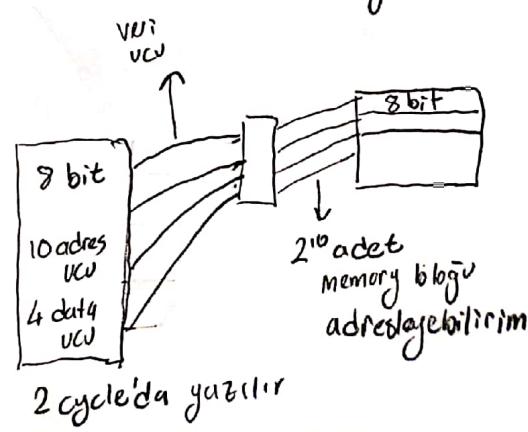
Intel 8085  $\longrightarrow$  1977

Intel 8086/8088  $\longrightarrow$  1978/1979

- 16 bitlik mikroislemci
- Santride 2,5 milyon işlem
- Adreslene Kapasitesi 1MB
- 4 veya 6 byte lik 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 mikroislemci  
Veri tek cyclede yazılır.  
(8 olsaydı 2 cycle)



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

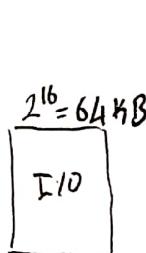
- SP yazmacı programının taşımadığı yığının genişliğini göstererek şekilde ilk değer alır.
- Her PUSH işleminde SP-1 ve SP-2 adreslerine 2 byte veri yazılır ve SP değeri 2 azaltılır.
- Her POP işleminde SP+1 ve SP+2 adreslerinden 2 byte veri okunur ve SP değeri 2 arttırılır.

# 8086 Üç Tanimlari

adres ucu, data ucu, sinyaller

bunu

halkacagiz



1MB  
Memory

Pin 1 - 40		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 deguledeiyorum
	20	RESET → mikroişlemci reset	

System

non-maskable interrupt  
AO == /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

IN AX, DX ; M/IO=0

DT/R

+ 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 etmesi.  
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 örtülmüyor 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.

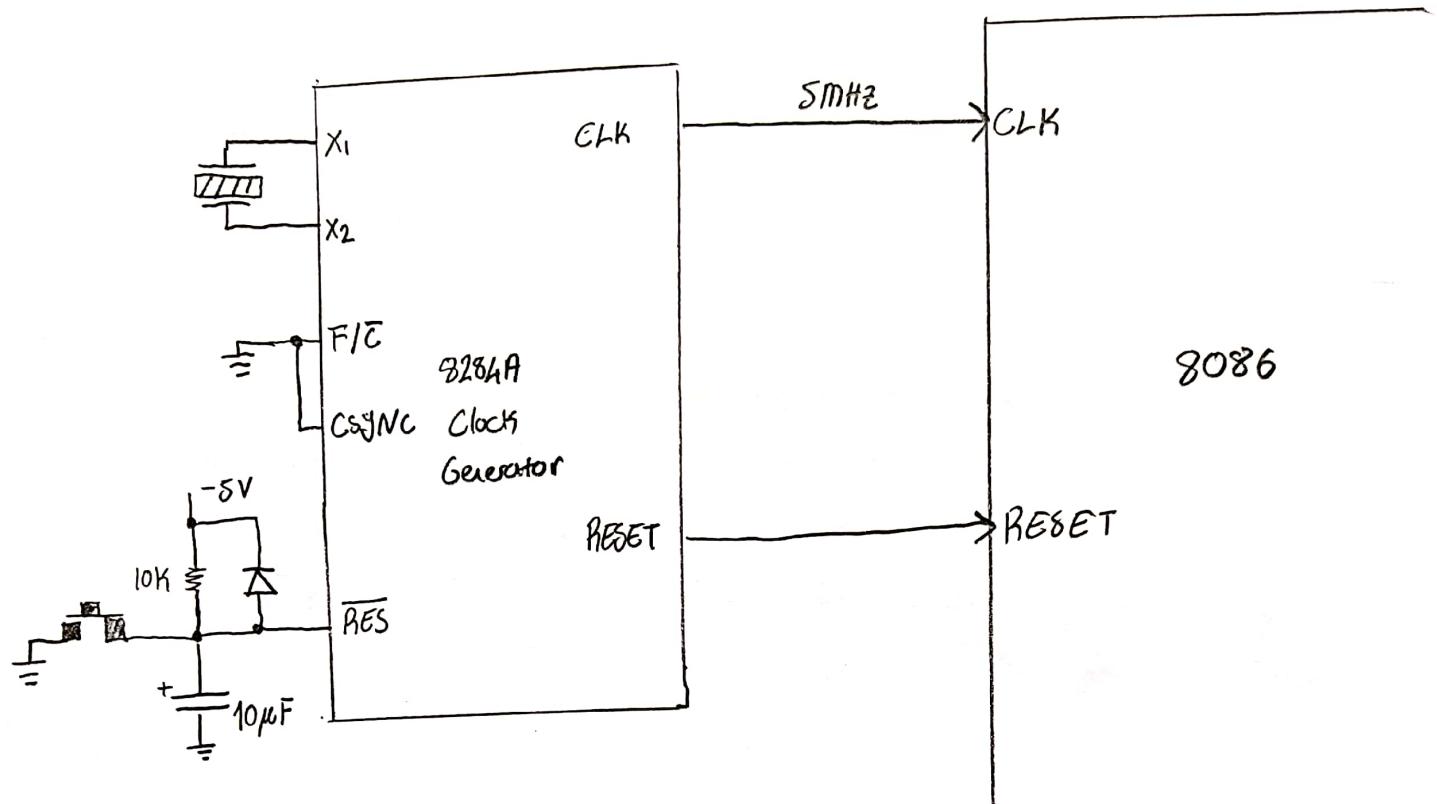
## + HOLD : (I)

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

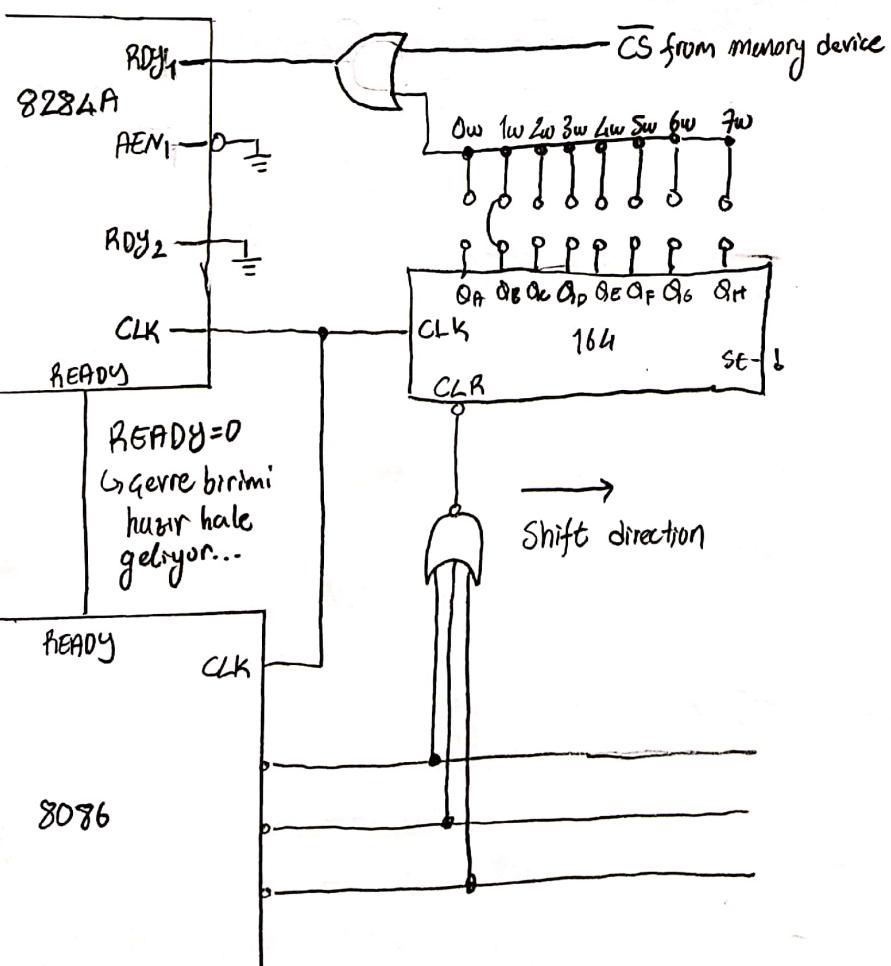
## + HLDA : (0)

8086 mikroişlemcisinin hold durumunda olmadığını gösterir.

# Clock Generator (8284A)



→ mikroişlemeyi hızlandırmak için



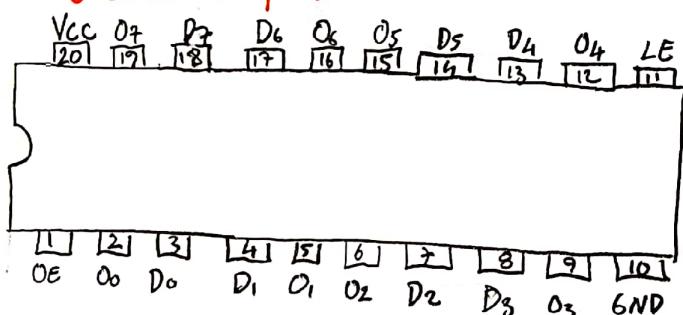
→ yavaş çevre birimlerinde  
ve okuma yazma olmak üzere  
mikroişleçin bekletilmesinde  
kullanılıyor.

# Bus Buffering & Latching

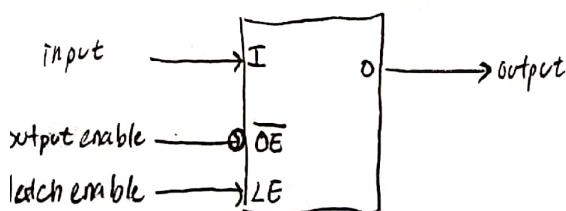
- Örneğin write veya read mikroişlemcilerde aktıyor ve button genel birimleme bağlıyor. Dolayısıyla bu ucların uygun elektriksel seviyelerde kalması gerekiyor. Buffer devreleriyle data ucan yollar boyunca stabil kalmasını sağlıyoruz.
- Latching ise, zaman gerekliliği olarak (AD) adres ucları t1 anda 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 değişse 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

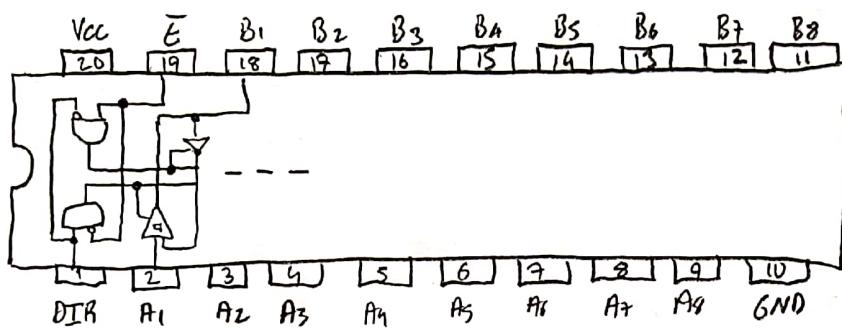
Z = High impedance

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

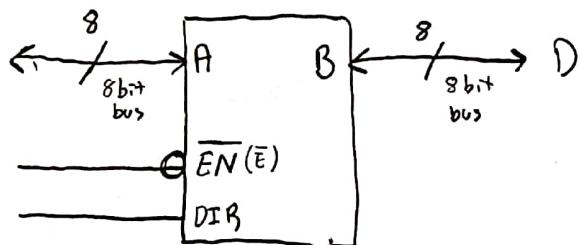
## 74LS245 Octal Bus Tranceiver



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  
↓ sıfır dataları

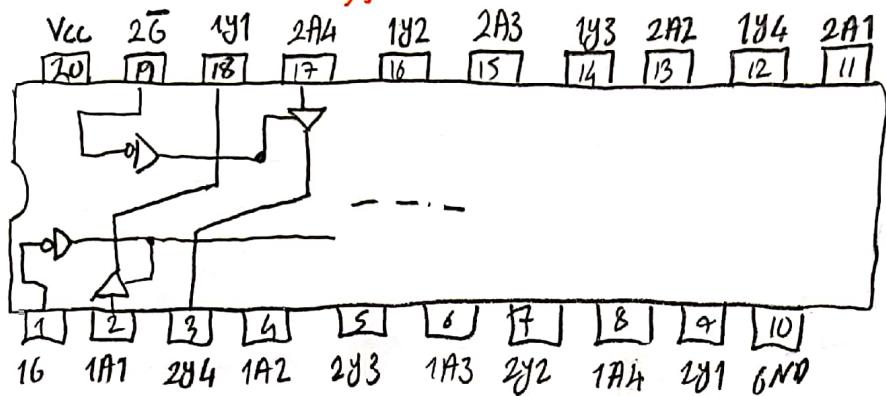
D—B

DT<sub>i/R</sub>—DIR

DT=0 veri gönderme.  
R=0 vur okuma

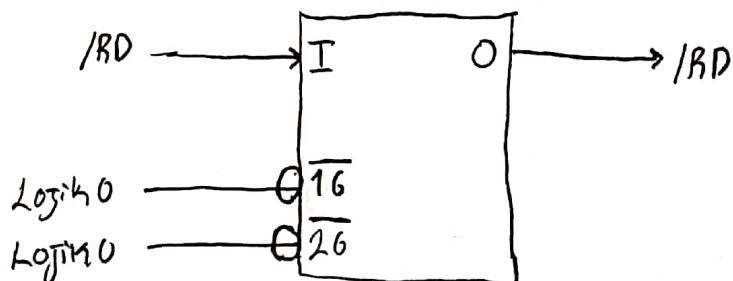
/DEN—/E

## 74LS244 Octal Buffer



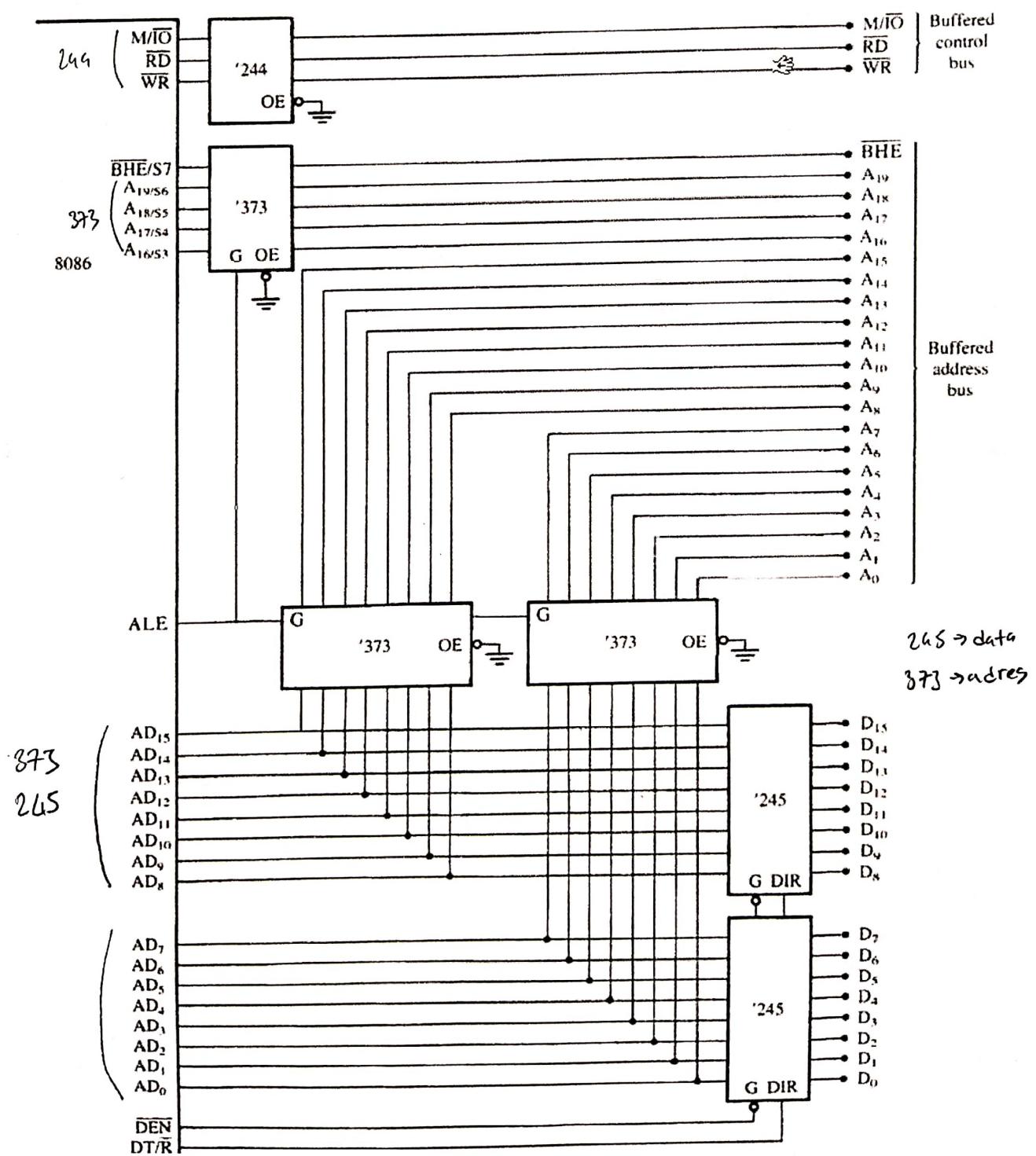
INPUTS	OUTPUT
16,26	D
L	L
L	H
H	X
	(?)

Veri tekrarlayıcı (Kuvvetlendirme)

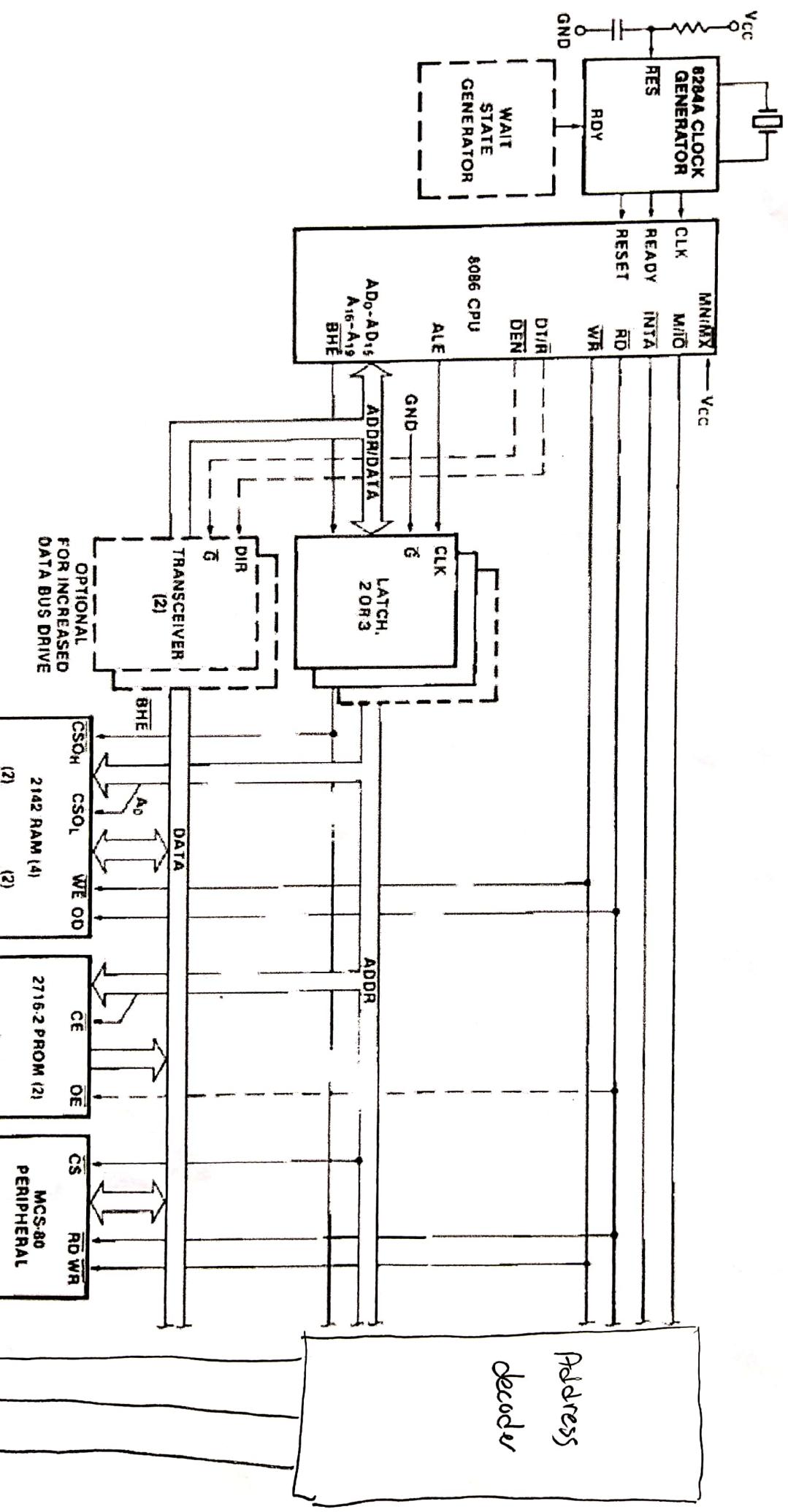


Bu yardımcı derre fiziki uglaan  
akıslarında 8086'ya göre daha  
yüksek miktarlarda akım ile bu oqları  
sürebiliyor  
örn 8086 = 300mA  
244 = 500mA

# Bus Buffering & Latching



# BUS Buffering & Latching



# Gevre Birimleri (Peripheral Devices)

Makine kodu: 00100000 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

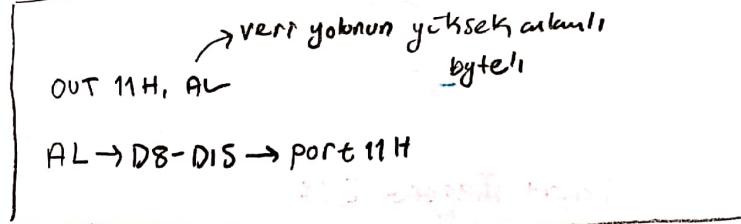
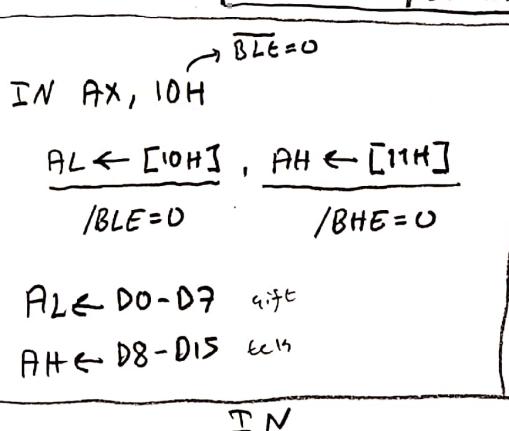
	IN Komutu	OUT Komutu
Fixed Address	IN AH, P8	OUT P8, AL
	IN AX, P8	OUT P8, AX
Variable Address	IN AL, DX	OUT DX, AL
	IN AX, DX	OUT DX, AX

port adresi 8 bite olduğuyorsa  
DX üzerinde port tanımlanır.

Data nerede? → AL veya AX

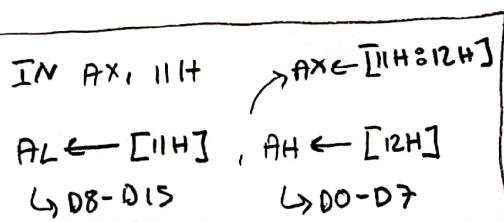
port çiftice düşük  
okunur

port tekise yüksek  
okunur



IN AL, 10H

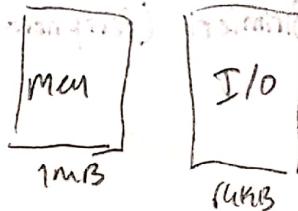
port 10H → D0-D7



## 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  
64KB

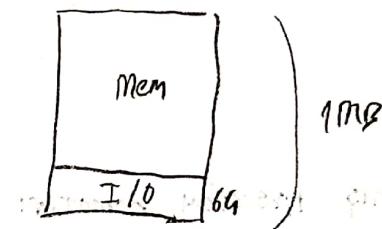
### - Memory Mapped I/O:

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.

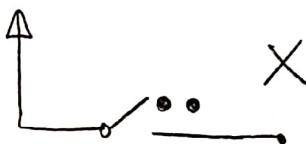
- I/O birimleri için ayrı komutlara gerek yoktur.

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

## Düğme Arayüzü (Basit Arayüz Devresi)

SGM74HC164

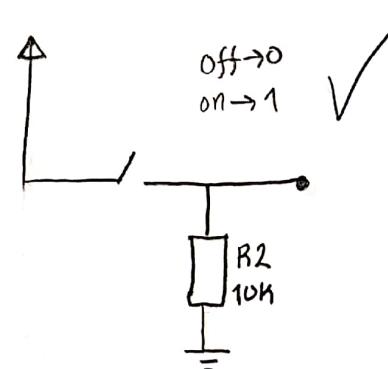
- Basit giriş arayüzü olarak kullanılabilir.
- Basılmadığı durumda geçerli lojik bir seviye iretmelidir.
- Kontakt görünübüne karşı yazılımsal veya donanımsal önleme gereklidir.
- Düğmeye basınca / bırakınca besleneyenle ground kısa devre olmamalı.



↳ Düğme bırakılınca aksılı  
lojik değer yok



↳ Düğmeye basınca  
kısa devre



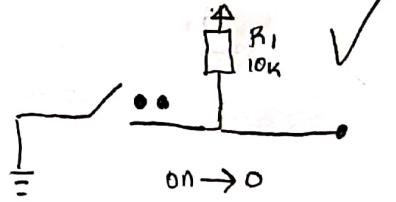
off → 0  
on → 1



↳ Düğme bırakılınca  
lojik değer yok



↓  
Düğmeye basınca  
kısa devre



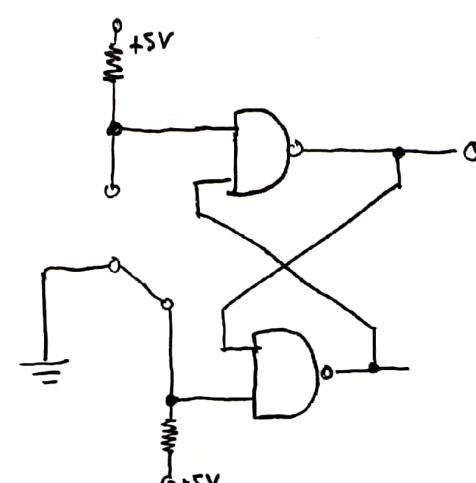
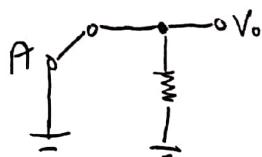
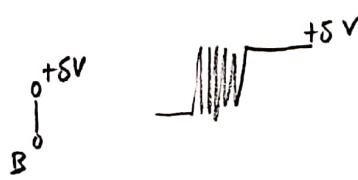
on → 0  
off → 1



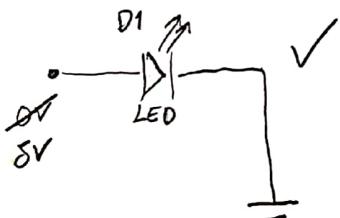
düğme t anında Kapsulursa

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

### Hardware Debounce Switch



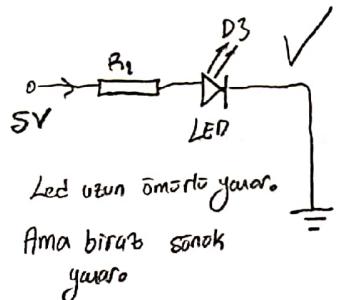
## LED Arayüzü



Işık verir ama omzunu  
kısıca olur.

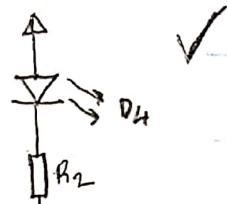


Yanmaz

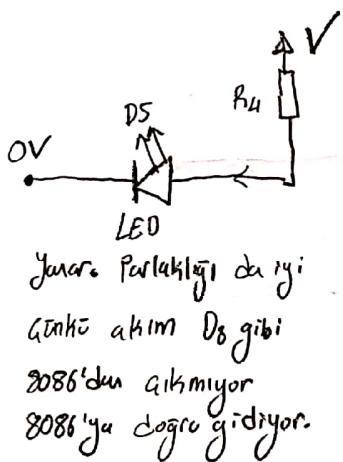


Led ızdırır omzunu yanar.  
Ama biraz sönük  
yanar.

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



Dogrudan 8086'dan sorerken ahim.  
Yanar. Uzun omzunu parlak.



Yanar. Parlaklığını da iyisi  
gördü akmış D8 gibi  
8086'dan aksımyor  
8086'ya doğru gitdiyor.

## Busit Çevre Birimleri için Adres Gözümlene

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

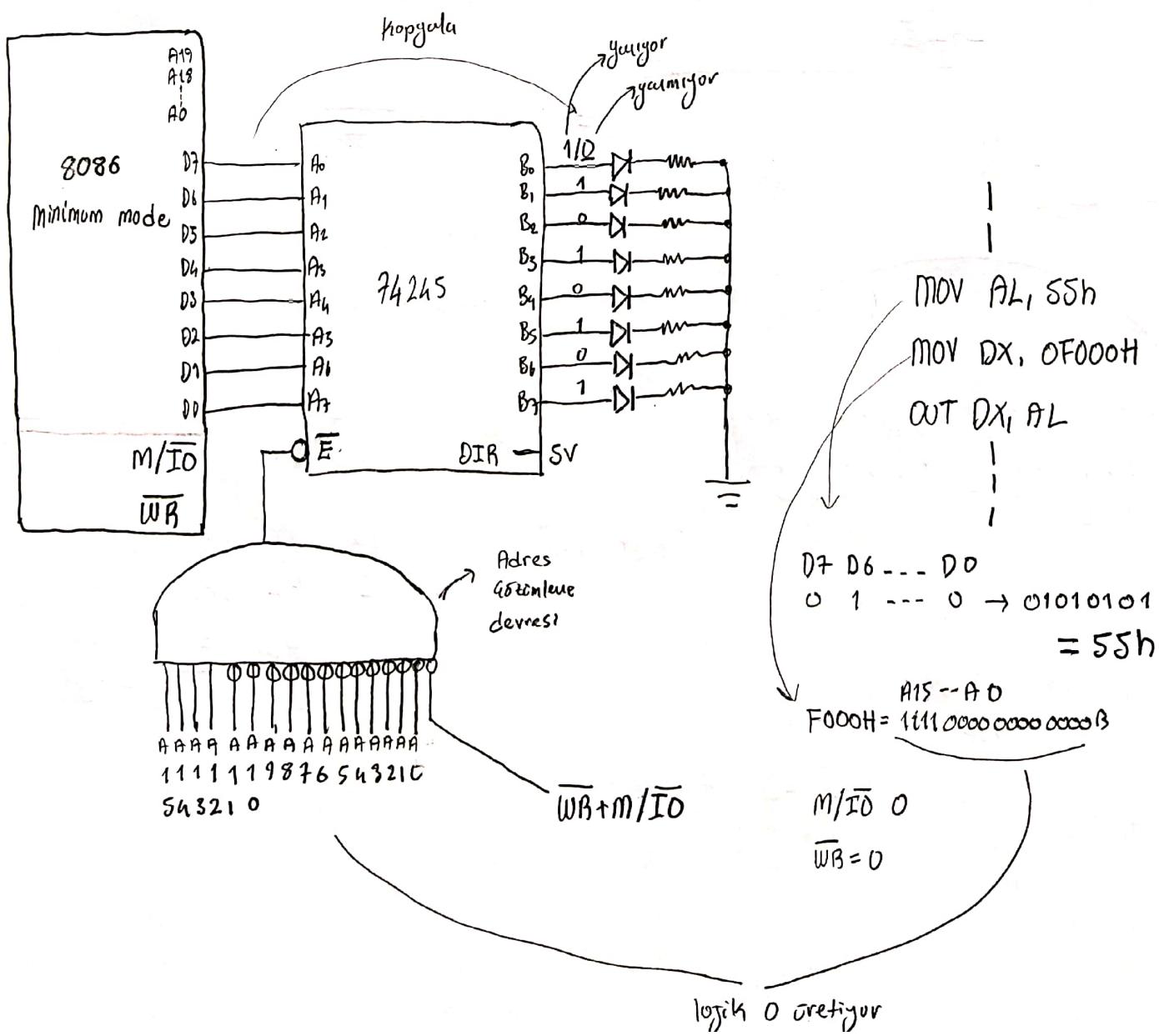
Variable I/O  $\rightarrow$  A15 - A(0+1)

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

adres çözümlene devresine gitmeli.

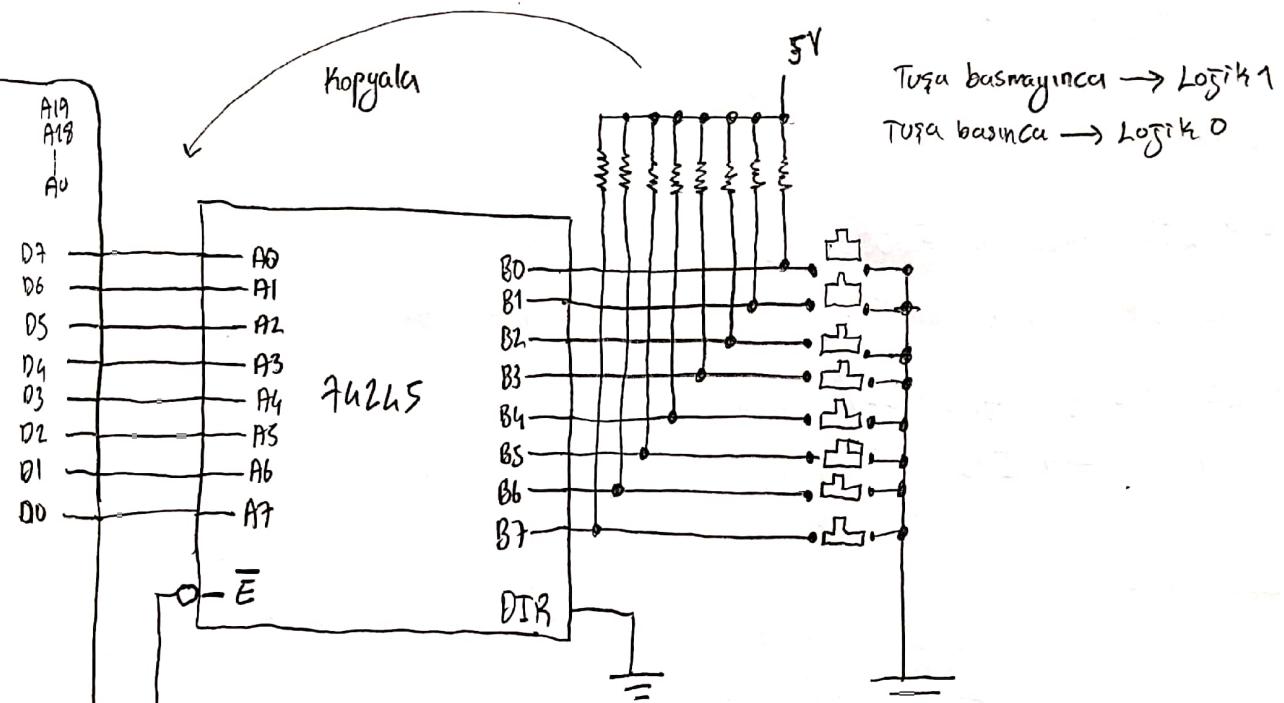
## Busit Giriş Birimi

- 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

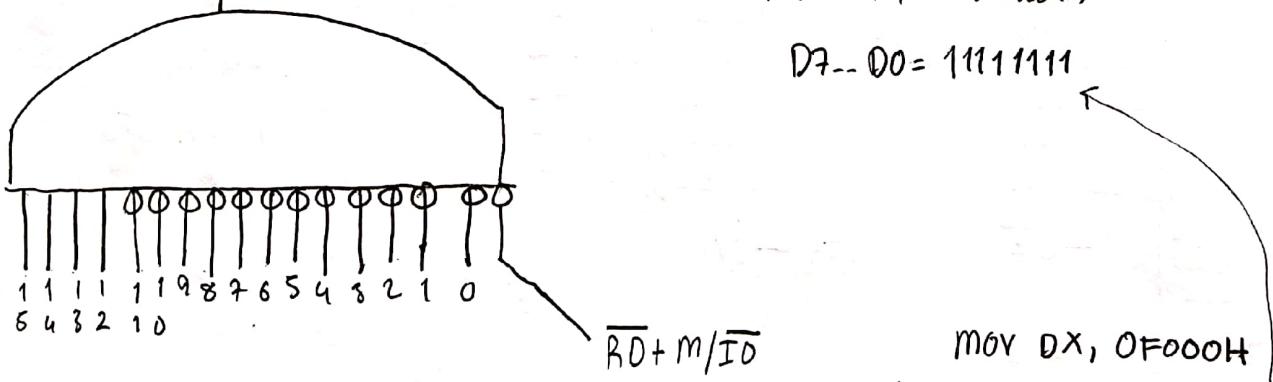
- 8 doğme kullanarak OF000H adresine yerleştirilmiş  
basit bir giriş biriminin fonsiyonları → donanım adres gottenme
- Doğmelerin okunması → I/O programlama



Tuşa basmadığında → Logik 1  
Tuşa basınca → Logik 0

Habir tuşa basmazsa

$$D7 \dots D0 = 11111111$$



MOV DX, 0F000H

L1: IN AL, DX,

CMP AL, OFFH

JNE L1

Tuşa basıldığında  
zaman bos dengide

dönüyor

$$\begin{array}{ll} A18=1 & A10=0 \\ A16=1 & A9=0 \\ | & | \\ A11=1 & A0=0 \end{array} \quad \begin{array}{l} \bar{RD} = 0 \\ M/I\bar{O} = 0 \end{array}$$

## Basit Giriş Çıkış Birimi

Hem OF000H, hem OF001H

- 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?

↳ 2-karadere adres çözümleyici

F001H

↳ 1) D8-D15

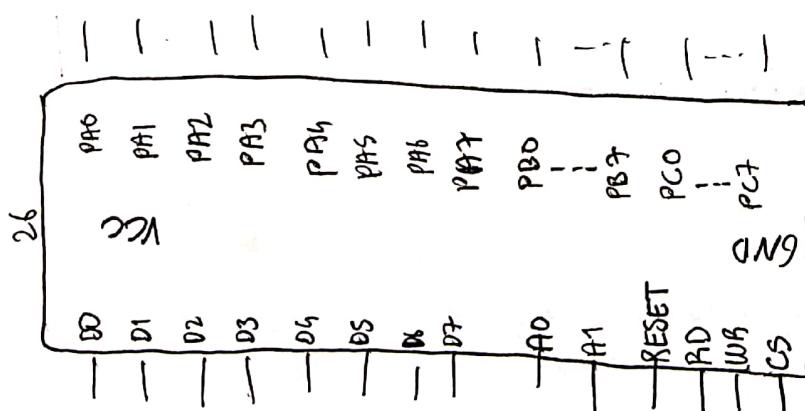
2) A0=1 ile adres çöze

veya

/BHE=0 ile adres çöze

## 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 çözümleme devresinin aktifisi — CS)
- RD: μP'in 8255'ten okuma yapmasını sağlar (μP RD — RD)
- WR: μP'in 8255'ye 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

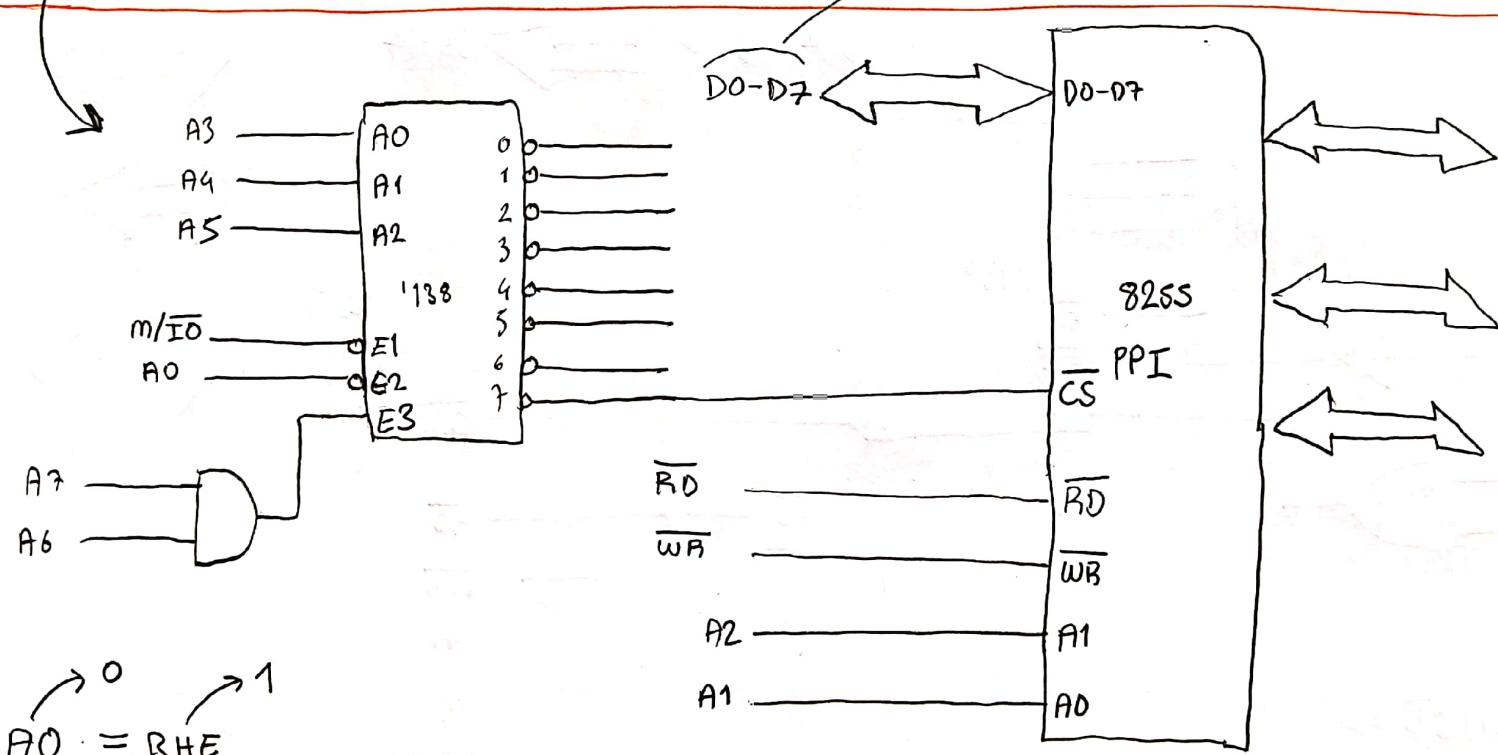
$B+E$	$A_0$	Transfer
0	0	Anlamlı değil.
0	1	Tek adresleki byte D8-D15
1	0	Cift adresleki byte D0-D7
1	1	Anlamlı değil.

## 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

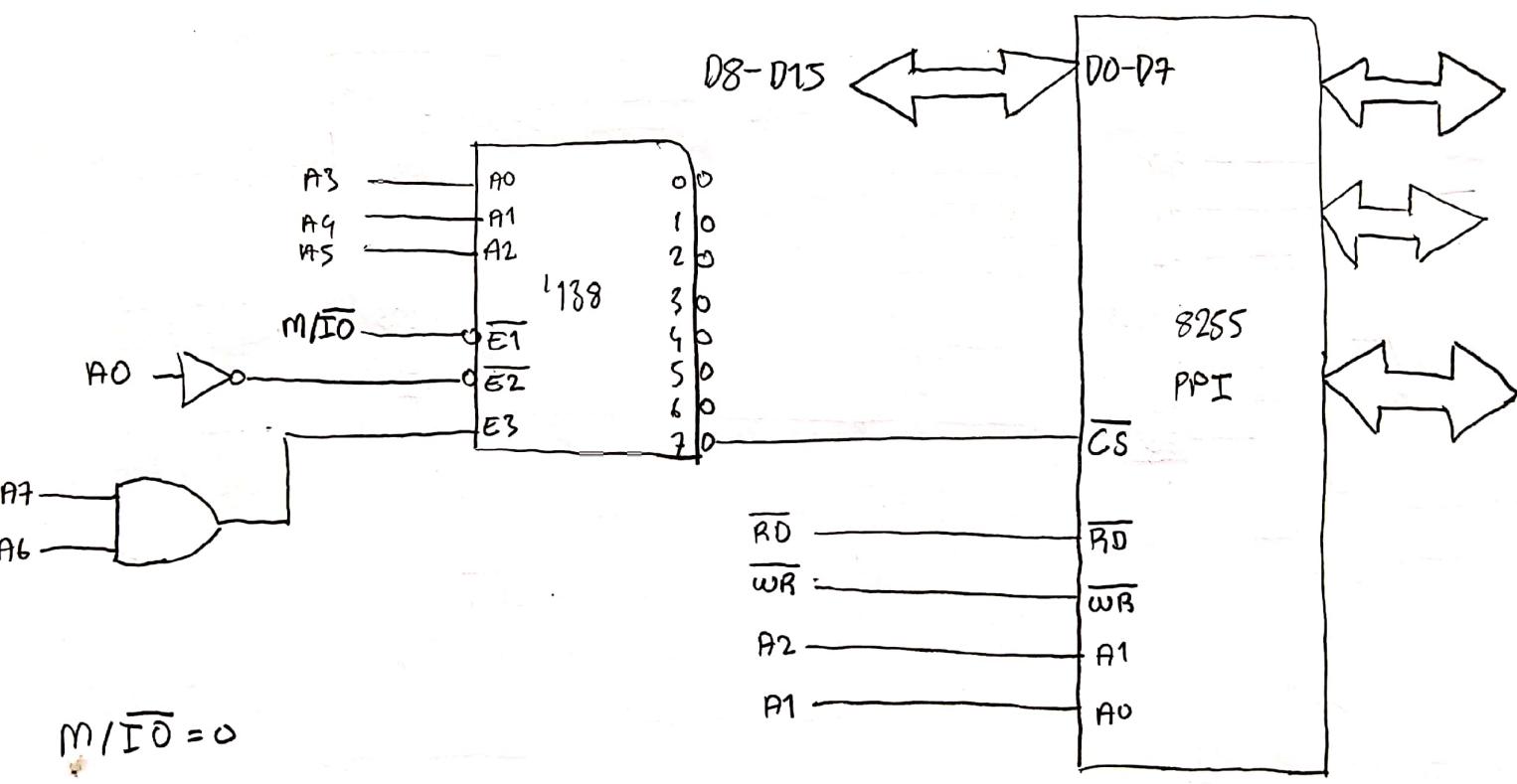
OFCH → PORT C

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



OF9H → PORT A

OFBH → PORT B

OFDH → PORT C

OFFH → CONTROL

## 8255 Mod Seçimi

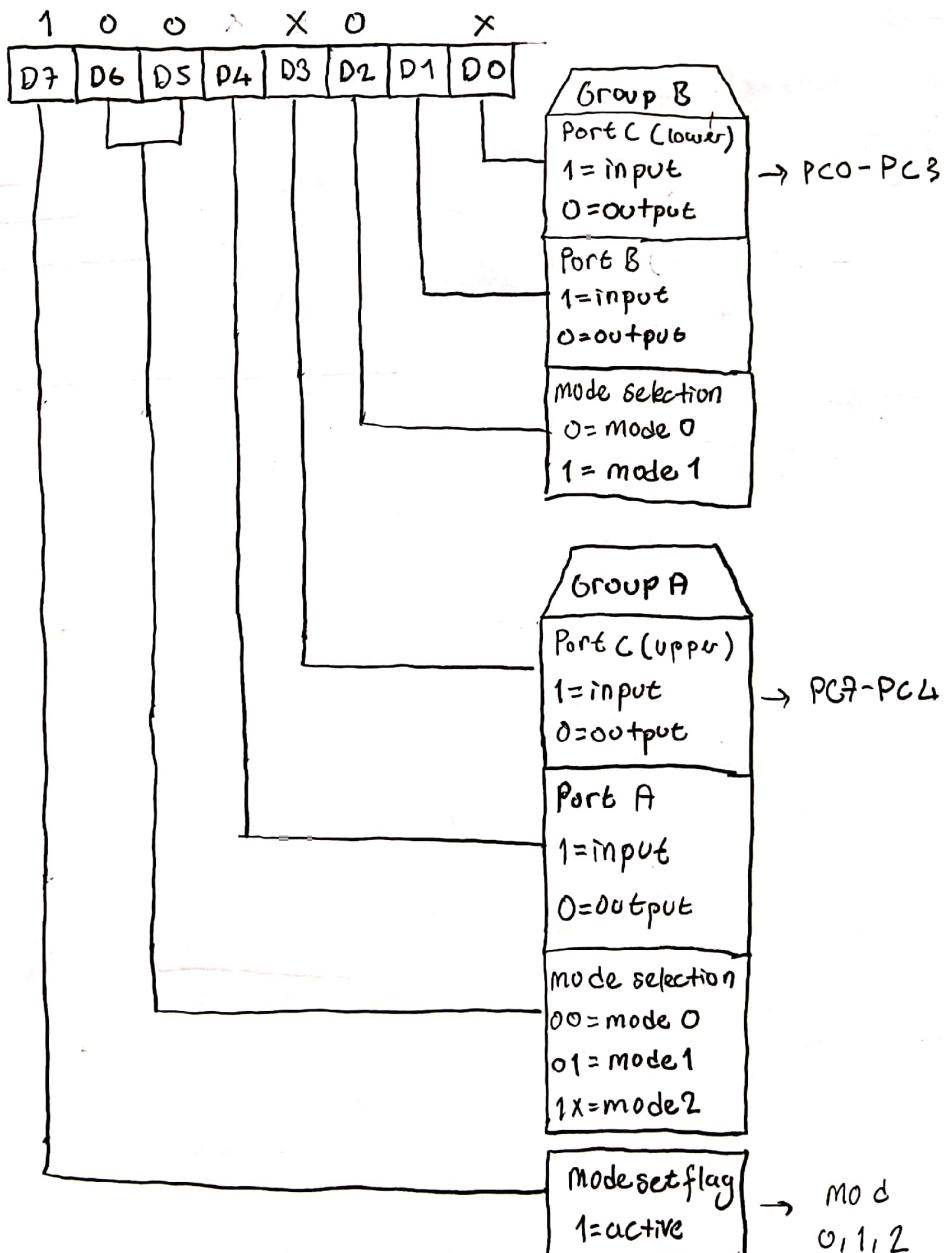
- 8255, 4 farklı mod ile çalışabilir:

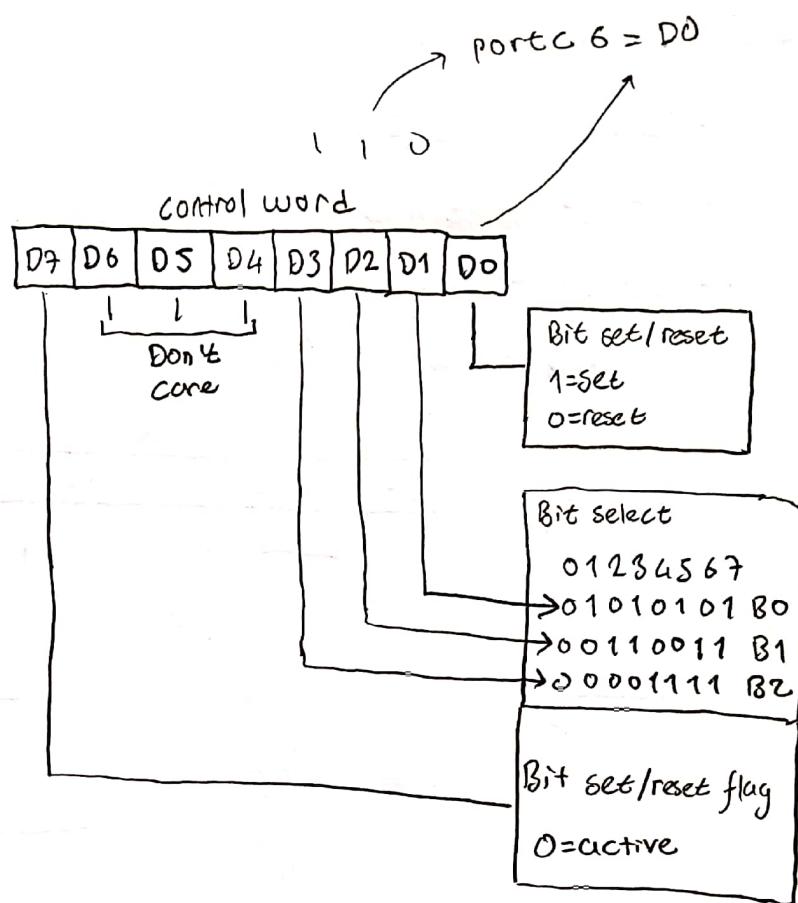
— Mod 0: Basic I/O → input, output

— Bit Set Reset Mod

— Mod 1: Strobed I/O → handshaking

— Mod 2: Bi-directional Bus





8255

Kontrol Yazmacı  $\Rightarrow$

BSR

PC her bir pinini ayrı ayrı  
out 1,0  
mod1, mod2 handshaking  
config bit

# I/O Mode

## Mode 0

Simple I/O for all three ports A, B and C.

PA  $\rightarrow$  I: 8 bit  
O: 8 bit

PB  $\rightarrow$  I: 8 bit  
O: 8 bit

PCL  $\rightarrow$  I: 4 bit  
D: 4 bit

PCH  $\rightarrow$  I: 4 bit  
O: 4 bit

- No handshaking

## Control word

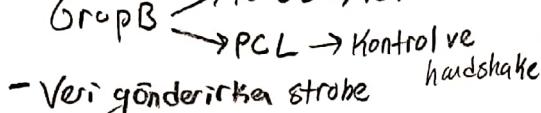
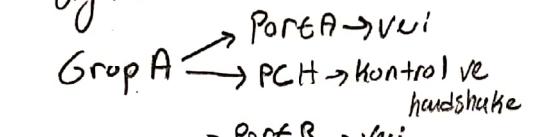
D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	1/0	X	X	X

PC<sub>6,7</sub>  
1= input  
0= output

## Mode 1

Handshaking I/O for ports A and B.

Port C bits are used for handshake signals.



- Veri göndermekten önce strobe işaretini.
- Veri okundugunda acknowledge.

Veri yönü tako.

Grup A ve grup B ayrı modlarda çalıştırılabilir.

## Mode 2

Bidirectional databus only for Port A. Port B

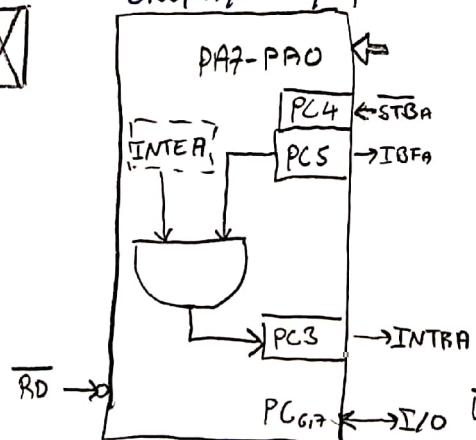
can be used in either mode 0 or mode 1. Handshake signals derived from bits of port C.

STB  $\rightarrow$  Strobe

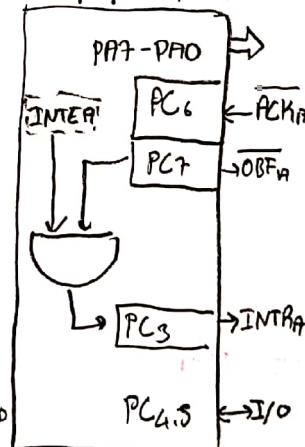
strobe = 0  $\rightarrow$  veri uclarinda veri hazır, okuyabilirsin

IBF  $\rightarrow$  input buffer full

## Group A / Mod1 / input



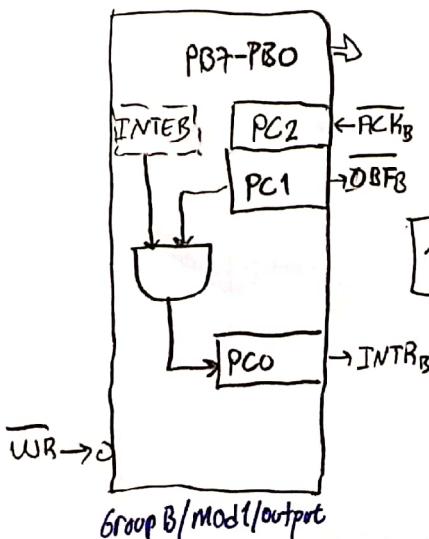
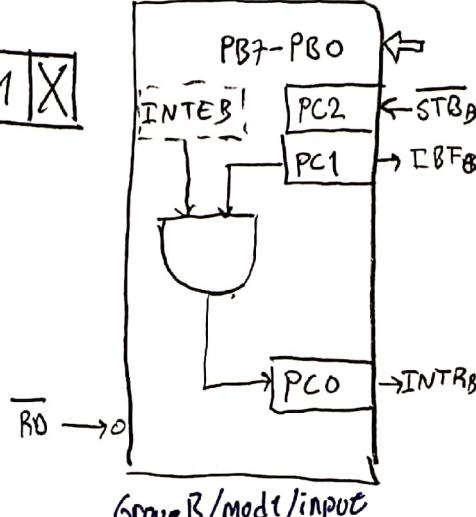
## Group A / Mod1 / output



1	0	1	0	1/0	X	X	X
---	---	---	---	-----	---	---	---

PC<sub>4,5</sub>  
1= input  
0= output

1	X	X	X	X	1	1	X
---	---	---	---	---	---	---	---



1	X	X	X	X	1	0	X
---	---	---	---	---	---	---	---

## 8255 Mod 1 - Input

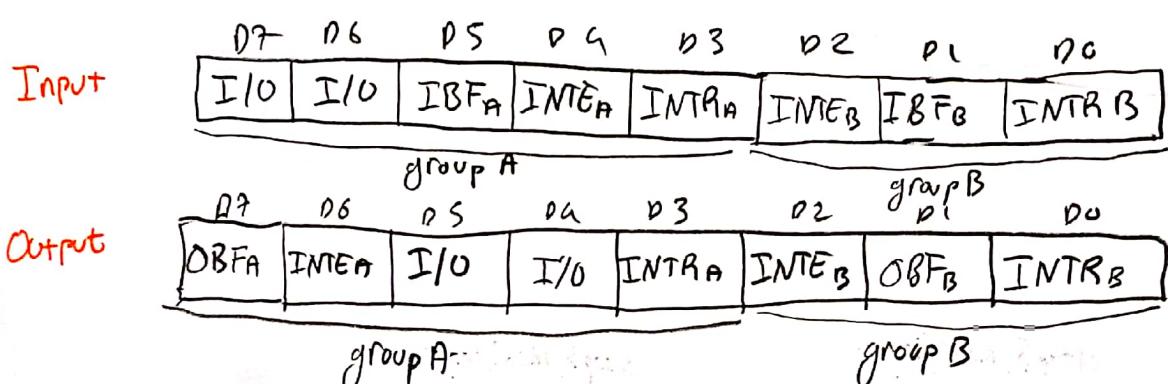
- STB: Logik sıfır değeri verinin port uşalarından içerdeki port yutmacına okunmasını sağlayan kontrol işaretü
- IBF: Logik 1 çıktıfo içerdeki veri kopyalanma işlemi bittiği zaman IBF 1 olur. 8086 buffer'ı okursa IBF 0 olur (RD ile Odlur).
- INTR: Seviye tetiklemeli kesme ucu. STB, IBF ve INTE 1 ise INTR 1 olur. RD ile INTR Odlur.
- INTE A: Bit <sup>Set</sup> reset modunda PC4 set edilir
- INTE B: Bit <sup>Set</sup> reset modunda PC2 set edilir.

## 8255 Mod 1 - Output

- OBF: Output buffer full.  $\mu$ P 8255'in registerlerinden birine veri gönderince logik 0. WR iken OBF 0.
- Ack: Acknowledge. Kursatı tarafı veriyi değerlendirdi.
- INTR: Ack, OBF ve INTE 1 iken INTR 1.
- INTE A: PC6
- INTE B: PC2

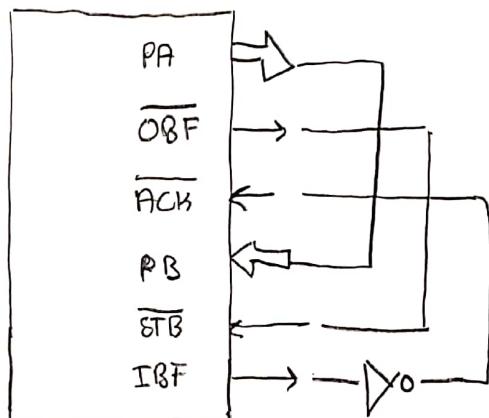
## 8255 Mod 1 - Status Word

- 8255 Mod 1'in ayarlanmışsa PORT C'de yapılan okunmalar STATUS WORD'dür.
- OBF, IBF, INTR değerleri ile I/O'ların kullanılan PORTC uşuları okunabilir.



## Örnek:

8285 Group A Mod 1 output, 8255 Group B mod 1 input yonlu ayarlamak, bunlar arasinda handshaking tabanlı I/O islemleri nasıl yapilabilir?



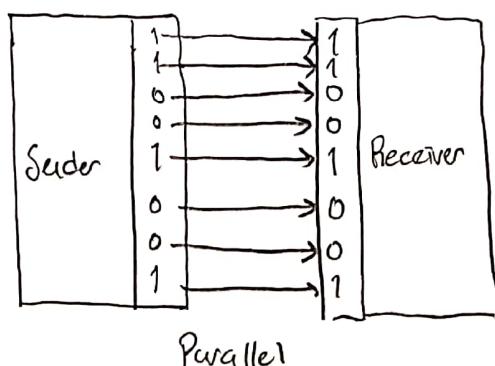
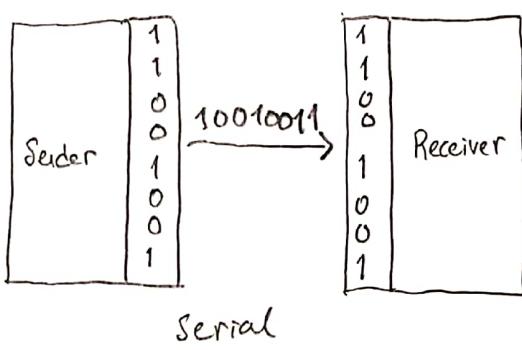
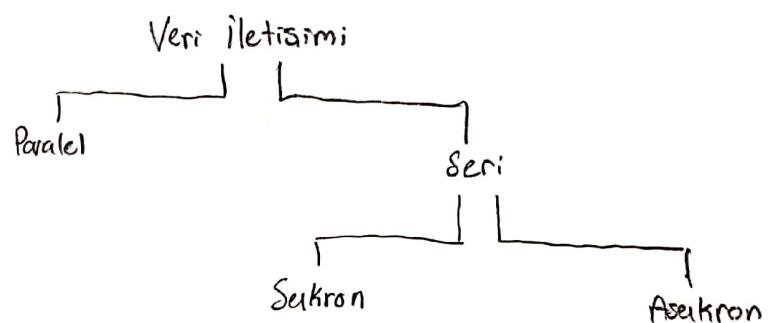
## Veri iletisimi: Seri - Paralel

### - Seri

- Daha az malzeme
- Daha yavaş

### - Paralel

- Daha hızlı
- Kısa mesafeler için



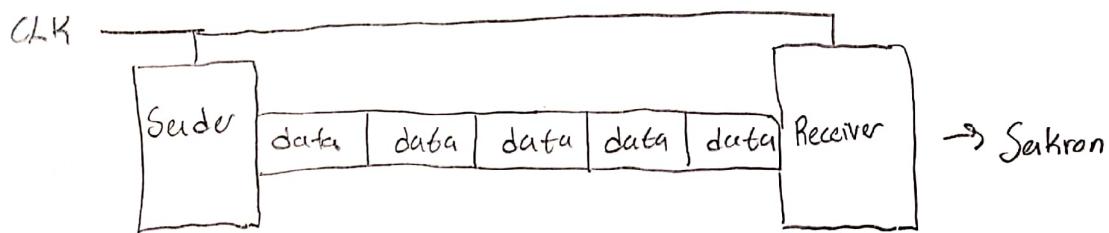
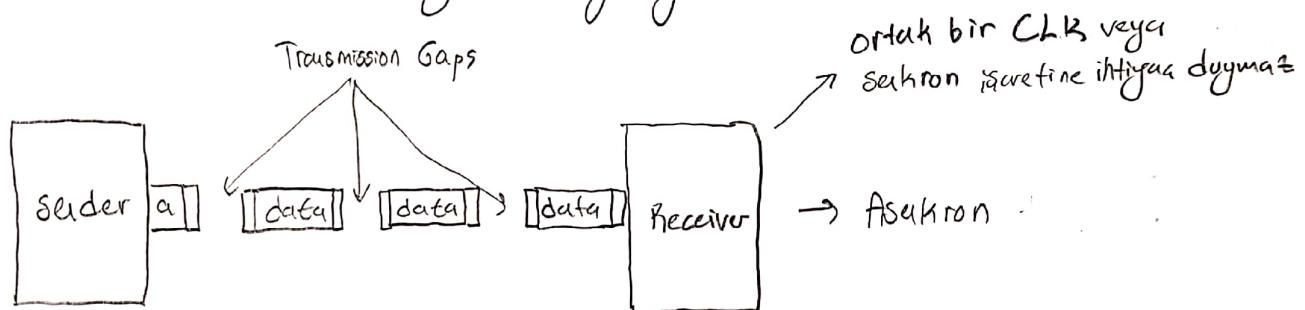
## Seri İletişim

### Sektron

- Sender ve receiver sektron olmalı.
- o Done in hardware using phase locked loops (PLLs)
- Vuru bloğu gönderilebilir.
- Daha verimli asenkron iletişimde daha ek yük.
- Maliyetli

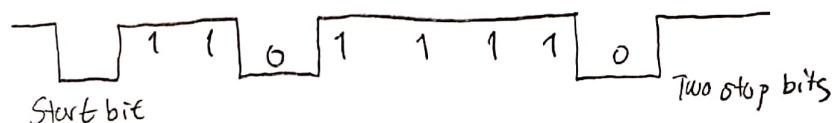
### Asenkron

- Her byte iletişim için kodlanmalıdır.
  - o Start ve stop bitler
- Sender ve receiver sektronluk yönüne ihtiyaç yoktur.



### Asenkron

- Vuiler bitler halinde gönderilir.
- Alıcı taraf iletisimin başladığını/bittiğini, start/stop bitleri ile ölçer.
- Hat başta iki logik 1 değerindedir.



## Error Checks

### - Parity check

- o Even parity: When odd numbers of 1 make  $D_7 = 1$

- Send even number of 1

- o Odd parity: When even number of 1 make  $D_7 = 1$

- Send odd number of 1

### - Check sum

- o Used for block of data

- o Sum of all bytes without carry and 2's complement

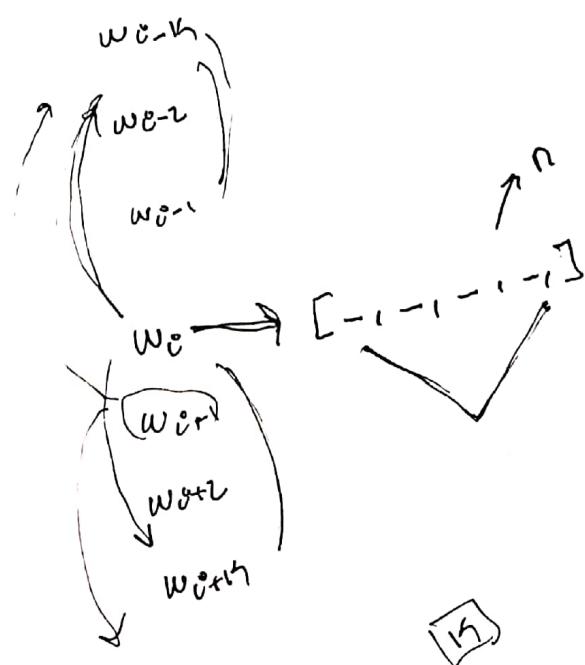
- o Total sum Result should be zero

### - Cyclic Redundancy Code (CRC)

- o Synchronous Communication

- o Stream of data can be represented by Cyclic polynomial that divided by a constant polynomial

- o Reminder to set Bits and send out as check for error.



# Simplex/Duplex

## - Simplex

- o Data tek bir yönde iletilir
- o Örnek: CPU → printer

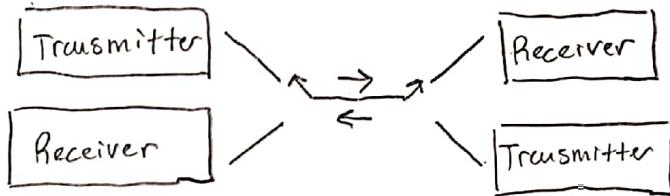
## - Duplex

- o Her iki yönde veri aktığı
- o Half Duplex (iletim bir anda devam ediyor)
- o Full Duplex (Both ways simultaneously)

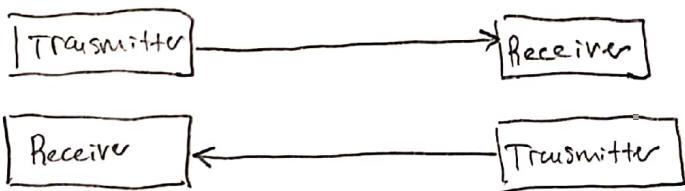
Simplex



Half duplex



Full Duplex



## Transmission Rate

- Rate at which bits are transmitted (BAUD)
- 1200Baud means Bit stay for  $1/1200 = 0.83\text{ms}$

- What is the data rate in bits/sec and character rate if the bit time is 8.33 ms (1 start, 8 data, 2 stop)

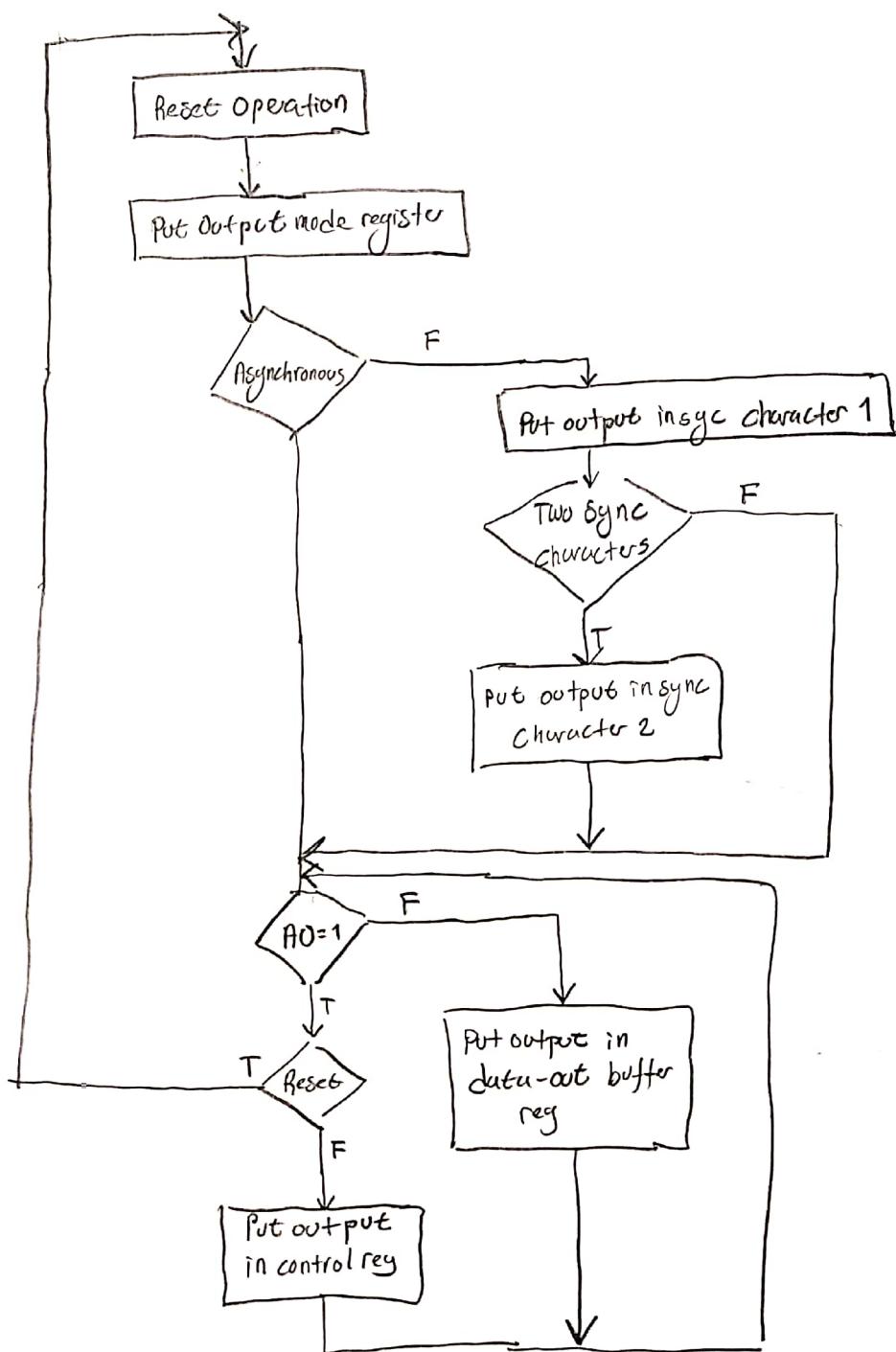
$$\text{Bit rate} = 1/8.33 = 300 \text{ bits/sec}$$

11 × 8.33 ms = 86.63 ms required to transmit a character so character

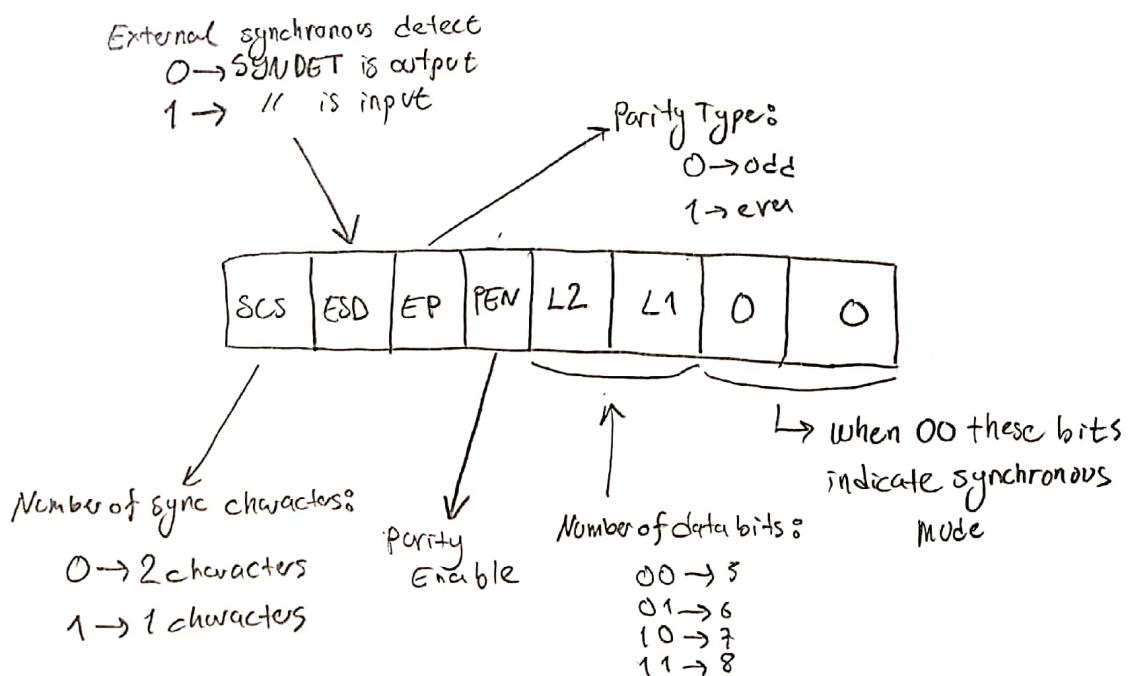
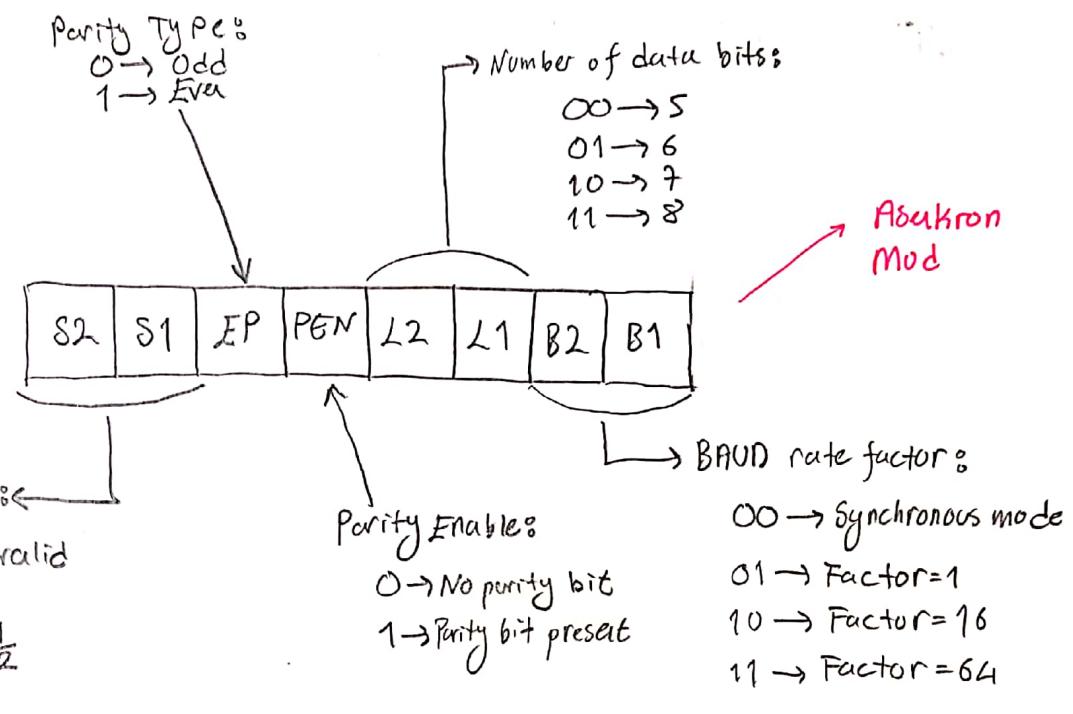
$$\text{rate} = 1/86.63 = 27.3 \text{ char/sec}$$

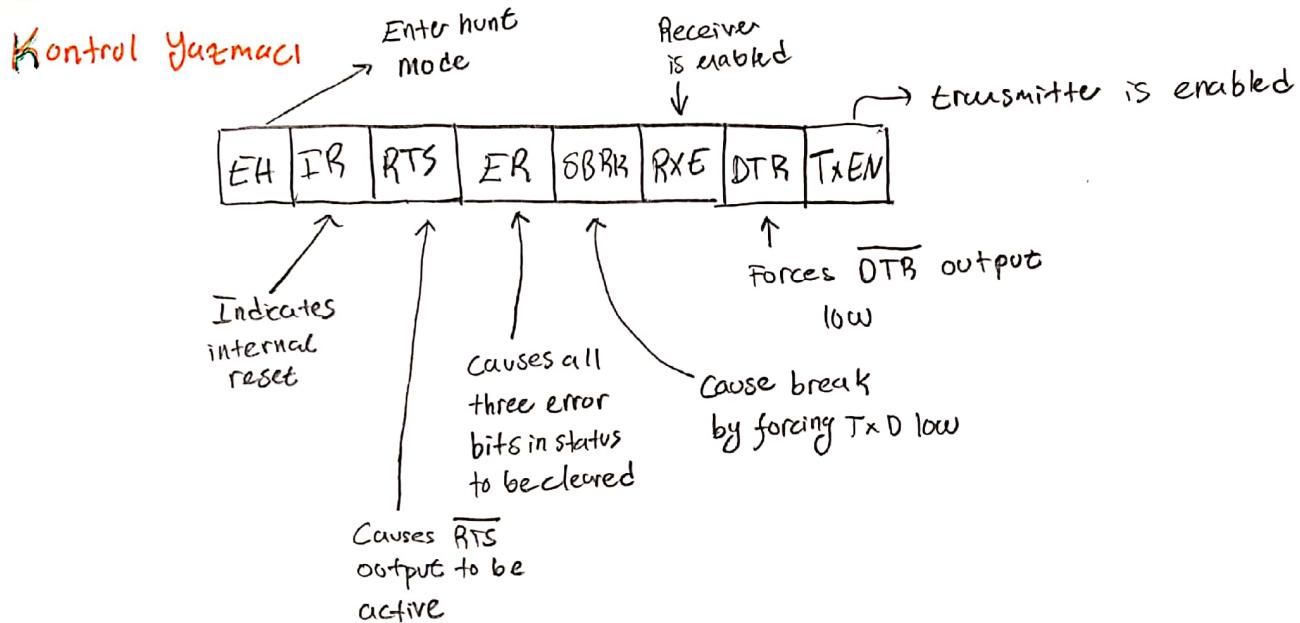
# 8251 USART

$\overline{CS}$	C/D	$\overline{RD}$	$\overline{WR}$	Ankan
1	X	X	X	Data-bus tristate
0	X	1	1	Data-bus tristate
0	1	0	1	Status $\rightarrow$ CPU
0	1	1	0	Mode Control, Sync $\leftarrow$ CPU
0	0	0	1	Data $\rightarrow$ CPU
0	0	1	0	Data $\leftarrow$ CPU

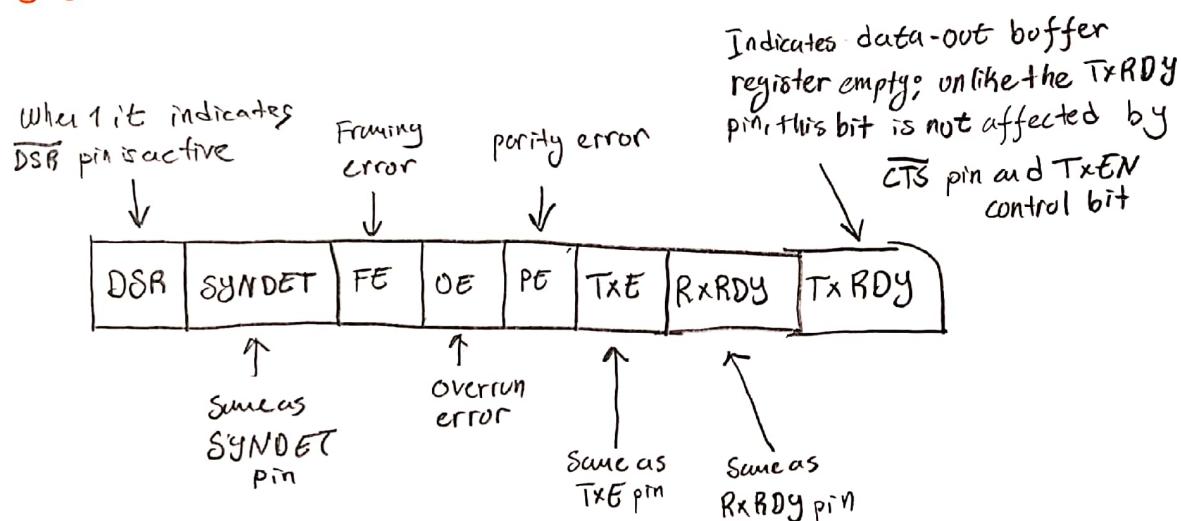


## Mod Yazzmacı



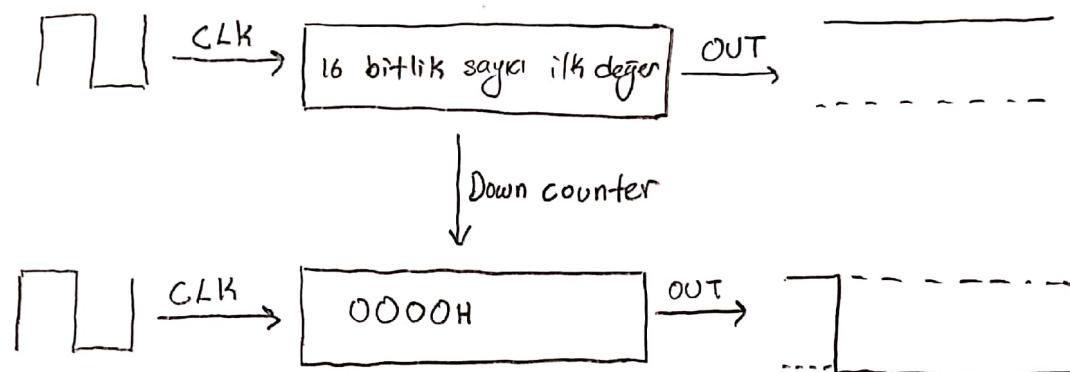


## Status Yazmaci

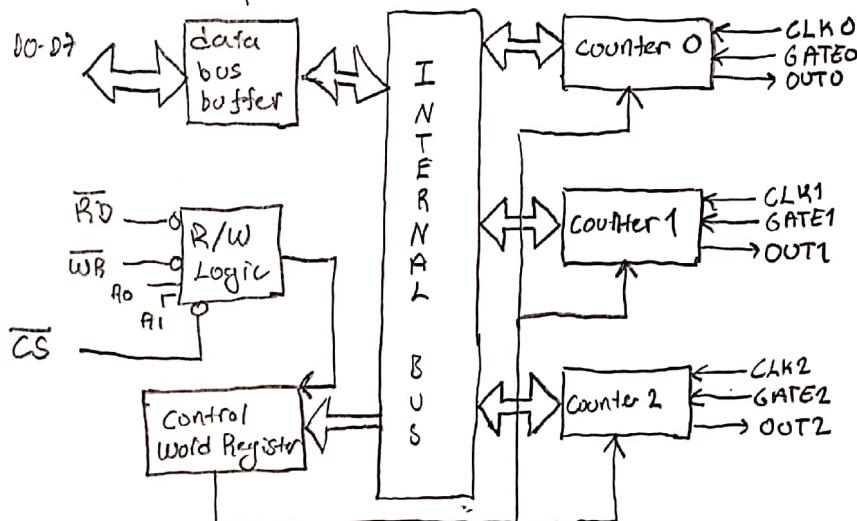


# 8254 Programmable Interval Timer (PIT)

- 8 bağımsız, 16 bitlik programlanabilir sayıcıları mevcuttur.
- Her bir sayıçın 6 farklı modda birinde programlanabilir.
- 8 MHz frekansına kadar saat girişinde çalışabilir.
- Sayıcı 1 → saat frekansını böler



8254 ia yapısı → 8 bitlik r/wi yolu



sayma değeri 0'a düşüğünde  
OUT akışında uygun değeri  
üretiyor olacak.  
status word üzerinde

CLK → sayıcının saat derbesi  
beslemesi için kullanıcamız  
ve.

GATE → Temelde dışarıdan sayma  
işlemlerini duraklatabile-  
ceğiniz veya yerden baş-  
latabileceğiniz giriş ucu

A0 A1  
2 adres ucu → 00 → counter 0  
4 byte  
01 → counter 1  
10 → counter 2  
11 → Kontrol yazmacına erisim

# 8284 Control Word

Format	$A_1, A_0 = 11$	Control Word %							
	$\overline{CS} = 0$	D7	D6	D5	D4	D3	D2	D1	DO
	$\overline{RD} = 1$	SC1	SC0	RW1	RW0	M2	M1	M0	BCD
	$\overline{WR} = 0$								

SC1

SC0

- 0 → Select counter 0
- 0 → Select counter 1
- 1 → Select counter 2
- 1 → Read-Back Command

BCD

- 0 → binary counter 16-bits
- 1 → binary coded decimal (BCD) counter (4 decades)

M2      M1      M0

- 0      0      0 → mode 0
- 0      0      1 → mode 1
- x      1      0 → mode 2
- x      1      1 → mode 3
- 1      0      0 → mode 4
- 1      0      1 → mode 5

RW1      RW0

- 0      0 → Counter latch command
- 0      1 → Read/write LSB only. (byte)
- 1      0 → R/W MSB only. (byte)
- 1      1 → R/W LSB first, then MSB

## 8254 Ayarları

- 98H adresindeki iki byte'lik çift adreslere yerleştirilmiş 8254'nin asağıdaki komutlar çalıştırılırsa:

MOV AL, 00110110B  
OUT 9EH, AL

→ counter 0  
binary sayma  
LSB-MSB  
mod 3

- 98H adresindeki iki byte'lik çift adreslere yerleştirilmiş 8254'nin
  - CNTR0: binary sayma, mod 3, CLK0'ı 4282 (BCD) ye bölecek şekilde ayarlayın.
  - CNTR2: binary sayma, mod 3, CLK2'yi C26A hex'e bölecek şekilde ayarlayın
  - CLK0 = 102 MHz, CLK2 = 108 MHz ise OUT1 ve OUT2 frekansı nedir?

a) MOV AL, 00110111  
OUT 9EH, AL

first  
LSB  
then  
MSB

MOV AX, 4282H  
OUT 98H, AL  
MOV AL, AH  
OUT 9CH, AL

b) MOV AL, 10110110  
OUT 9EH, AL

MOV AX, 0C26AH } first LSB  
OUT 9CH, AL } then MSB  
MOV AL, AH  
OUT 9CH, AL

$$c) f_{OUT0} = 102 \text{ MHz} / 4282 = 280 \text{ Hz}$$

$$f_{OUT2} = 108 \text{ MHz} / 49770 = 36 \text{ Hz}$$

BCD modda yazılabilen en büyük değer?  
Binary modda yazılabilen en büyük değer?

640000

FFFFFH

# 8254 Modları

MODE 0: interrupt on terminal count

MODE 1: hardware retriggerable one shot

MODE 2: rate generator

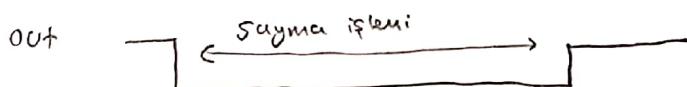
MODE 3: square wave generator

MODE 4: software triggered strobe

MODE 5: Hardware triggered strobe

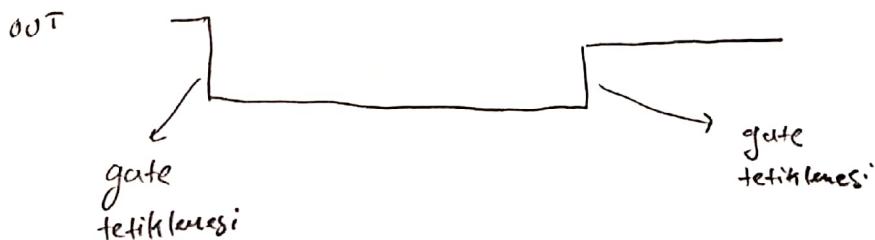
## Mod 0

- Kontrol yazıldıktan sonra  $OUT \rightarrow 0$
- Her CLK dösu keleviden sayıma değeri azaltılır.
- $GATE=1$  ise sayı sayar,  $GATE=0$  ise sayıma durur.
- Sayıma değeri 0 olduğunda  $OUT 0 \rightarrow 1$ , kalır.
- Yeni kontrol ve sayıma değeri yazılırsa tekrar sayar.



## Mod 1

- Ayar ve sayıma değeri yazılır.
- OUT başta 1
- $GATE 0 \rightarrow 1$  yegiğinde  $OUT 1 \rightarrow 0$
- Sayıma bittiğinde  $OUT 0 \rightarrow 1$
- Sayıma bittiğinden sonra  $GATE 0 \rightarrow 1$  iflari tekrarlar.
- Sayıma bitmeden  $GATE 0 \rightarrow 1$  olursa OUT 0 → 1 olmadan sayıma兀ar.



Mod 0'ın  
gate ucu ile  
geride tefiklerebilir  
versiyonu

## MOD 2

- Tekrarlı olarak belirli aralıklarla pulse üretir.
- OUT basılı 1.
- Sayma değeri 1 olduğunda OUT  $1 \rightarrow 0$
- ICLK sonrası OUT  $0 \rightarrow 1$
- Periyodik tekrarlıdır.
- GATE=0 olursa sayma durakları.
- Yarı sayma değeri devam eder cycle bitince ethi eder.



1 clock cycle'lik pulse'lar.

Örnek → CNTR0'ı  $\overset{\text{sañye}}{50\mu s}$ 'de 1 pulse üretecek şekilde ayarlayın.  $CLK_0 = 2MHz$

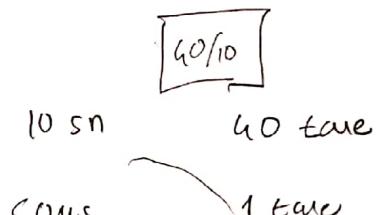
Control Word : 00 01 010 0B = 14H

CNTR0 seq, sadece L8B, binary sayma

$$\text{Sayma değeri} = \frac{2 \times 10^6}{\frac{1}{50 \times 10^{-6}}} = 100 = 64H$$

frekans

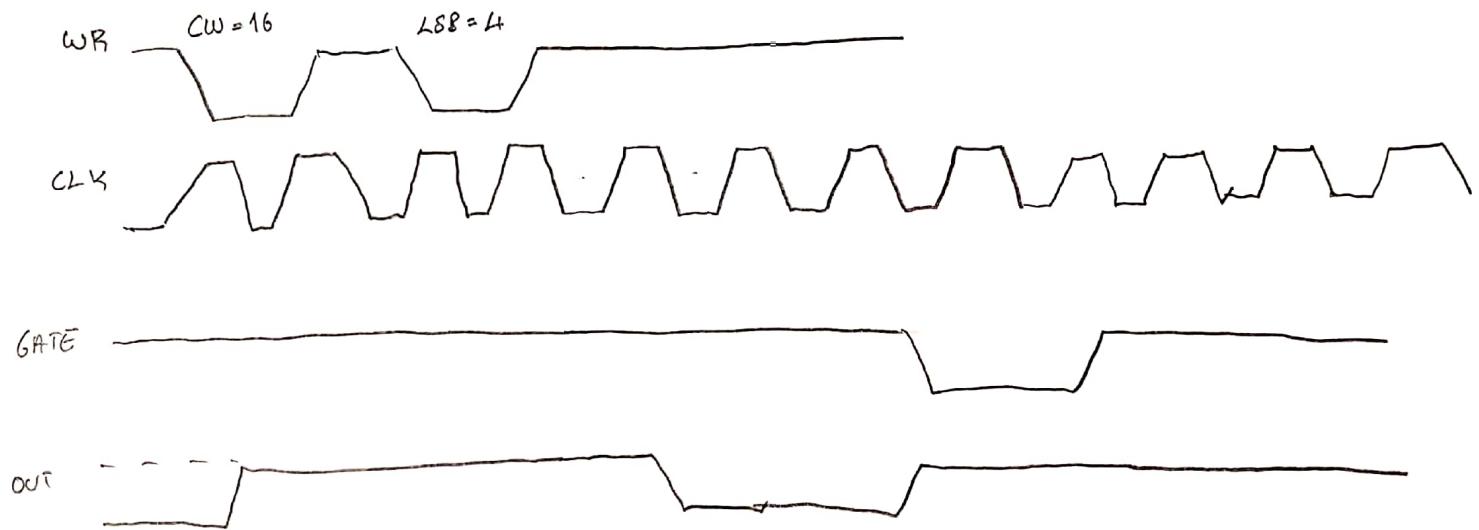
MOV AL, 14H  
OUT CONTROL-ADDRESS, AL  
MOV AL, 64H  
OUT CNTR0-ADDRESS, AL



$$\frac{1}{50}$$

## MOD 3

- Kare dalga üretici.
- OUT sayma değerinin yarısında 1, diğer yarısında 0 olur.
- Periyodik olarak tekrarlar.
- MOD 2'den duty değerinin %50 olması ile ayırtır. %50 duty cycle  
kare dalga üretiri.
- GATE=0 ise sayma durakları.



Örnek → CNTR1'i 1KHz frekansında kare dalga üretmek için şeitilde ayarlayın.

$$CLK_1 = 2 \text{ MHz}$$

$$\text{Sayma değeri} = \frac{2 \times 10^6}{1 \times 10^3} = 2000$$

Control word → 0111 0111B = 77H

CNTR1 seq, önce LSS sonra MSB, BCD sayma

```

MOV AL, 77H
OUT CNTRL-word-Addr, AL
MOV AX, 2000H
OUT CNTR1, AL
MOV AL, AH
OUT CNTR1, AL

```

≡

```

MOV AL, 76H
OUT CNTRL-WORD-ADDR, AL
MOV AX, 2000
OUT CNTR1, AL
MOV AL, AH
OUT CNTR1, AL

```

→ MOV AL, 76H  
 → MOV AX, 7000H

## MOD 4

- OUT başta 1
- Sayma değeri 1 olunca OUT  $1 \rightarrow 0$
- 1 CLK sonra OUT  $0 \rightarrow 1$  olur ve kalır
- Yarı sayma değeri yazılımadıkça tekrarlanmaz
- GATE=0 ise sayma duraklar.
- Mod 2'nin tek pulse'lı hali.

## MOD 5

- OUT başta 1
- GATE  $0 \rightarrow 1$  geçişinde saymeye başla
- Sayma değeri 1 olunca OUT  $1 \rightarrow 0$
- 1 CLK sonra OUT  $0 \rightarrow 1$  olur ve kalır
- GATE'ye yarı  $0 \rightarrow 1$  geçisi olmadıkça tekrarlanmaz.

## Counter Latch Command

- Sayıcıının (counter element, CE) o anki değerini yarıya yaramadı (OL) kopyalar.
- Latch'lenmeden, sayma devam ederken CE okunmaya çalışmakta hatalı.
- OL'ye kopyalanın değer CPU tarafından okunana kadar saklanır.

## Read Back Command

- Latch'lenmiş sayma değerini veya sayıcı durumunu (STATUS) okunmak için kullanılır.
- Birde fazla sayıçı için sayma değer/durumu bir seferde okunmak için kullanılabilir.
- Bir sayıciya ilişkin hem sayma hem de durum okunmak için komut verilirse
  - \* Sayıcı adresinden ilk okuma durumu
  - \* İkinci (ve öncü) okuma latches'lenmiş sayma değerleri
  - \* Sonraki okunalar latches'lenmiş sayıcı değerleri veir.

A0,A1=11 CS=0

RD=1 WR=0

07	06	05	04	03	02	01	00
1	1	COUNT	STATUS	CNT2	CNT1	CNT0	0

05 → Latch count of selected counters      04 → Latch status of selected counters

00 → reserved for future expansion; must be 0

## ADC

TERAKKI:

00H - FFFH  
0V SV → SV/256 ] 8 bit  
MOV DX, 200H  
MOV AL, 00H  
OUT DX, AL → CS  
ADC INTR UCİN  
400H adresine sahip I/O cihazı  
tanımlanmıştır.  
→ TSB

00H - FFFH  
0V SV → SV/1024 ] 10 bit  
INTR-KONTROL  
IN AL, DX → AD7  
TEST AL, 80H → en yüksek  
dallı bit  
JNZ INTR-KONTROL  
döşer olana kadar

Parallel ADC  
Ramp ADC → 0V-2.5V, shift register  
ADC6804 → I/O 1 byte  
CALL DELAY  
JMP TEKrar  
+ 1 byte Three  
state  
Buffer

ADC6804 → I/O 1 byte  
CALL DELAY  
JMP TEKrar  
+ 1 byte Three  
state  
Buffer

## DAC

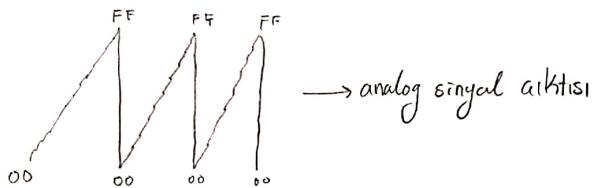
```

MOV DX, 1000H
MOV AL, 00H
TEKRAR:
    OUT DX, AL
    CALL DELAY
    INC AL
    JMP TEKRAR

```

$$\begin{array}{l}
 \text{FF} \rightarrow \delta V \\
 \text{00} \rightarrow 0V
 \end{array}
 \left. \begin{array}{l}
 \text{linear} \rightarrow V \\
 \downarrow \\
 \text{notinspace}
 \end{array} \right]
 \begin{array}{l}
 \text{AL} \leftarrow x \\
 \text{AL} \leftarrow x+1
 \end{array}
 \left. \begin{array}{l}
 V_{x+1} - V_x = 5V/256
 \end{array} \right.$$

8 bitlik cihaz



```

MOV DX, 1000H
MOV AL, 00H
XOR BX, BX
TEKRAR:
    OUT DX, AL
    CALL DELAY
    CMP AL, FFH
    JNE incre
    MOV BX, 1
    JMP cont
incre: CMP AL, 00H
    JNE cont
    XOR BX, BX
cont: CMP BX, 0
    JNE decre
    INC AL
    JMP end
decre: DEC AL
end: JMP TEKRAR

```



# Interrupts

Interrupt Kaynakları →

- Yazılımsal olarak : INT, INTO, INT3
- Donanımsal olarak : INTR, NMI
  - Donanım interrupt ile acknowledge : INTA → handshaking
- Interrupt ile ilgili bayraklar : IF, TF
- Interrupt ile ilgili dönüş komutu : IRET

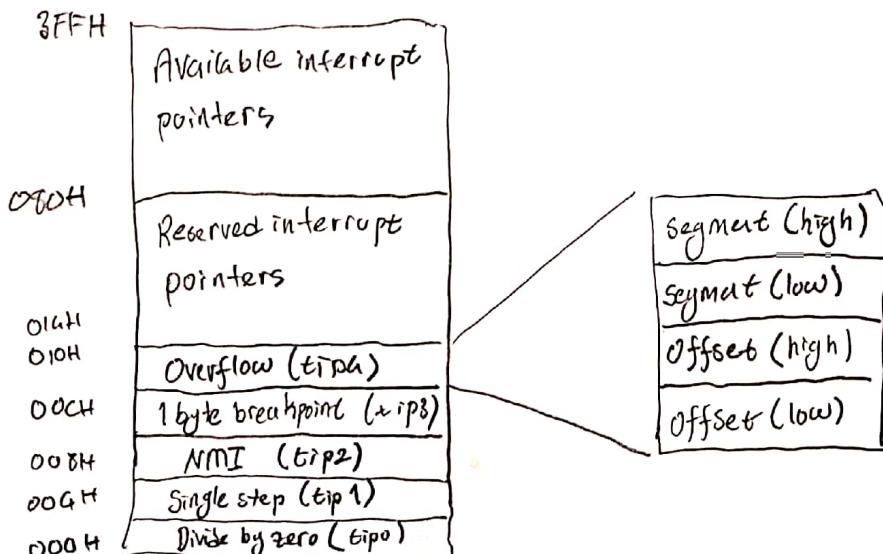
tip doğrudan arka

tip 2 interrupt

handshaking

## Interrupt Vektör Tablosu

- Interrupt oluşturulduğunda mevcut CS:IP ile gösterilen kodun yürütülmesi bırakılarak, kesmeye ilişkin fonksiyonun yer aldığı kesim ve offset değerindeki kod işlenmelidir.
  - 256 interrupt
  - software
  - hardware
  - DOS
  - User
- Her kesmeye ilişkin fonksiyonların hedef hafıza adresinde yer aldığı Interrupt Vektor Tablosu ile tutulur.
- IVT hafıza üzerinde 00000H-003FFH adres aralığında 1024 byte'lık olmalıdır.
- 256 farklı interrupt'in ilgili fonksiyonların offset ve interrupt değerlerini saklar.



## Interrupt İşlemi

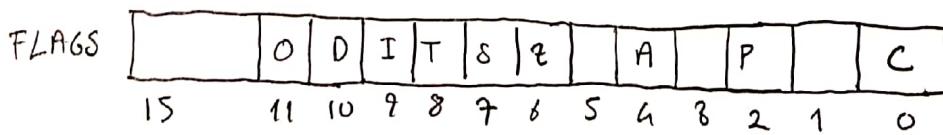
Sıradaki işlem bittiğinde, işlemci' s

Kritik istekler

- Komut işleme ile oluşur, single-step, NMI, INTR, INT sırasıyla bir kesme olup olmadığını kontrol eder.
- Kesme varsa
- PUSHF
- IF $\leftarrow$  0, TF $\leftarrow$  0 (INTR, single-step interruptı engellerir)
- PUSH CS
- PUSH IP
- IP $\leftarrow$  interrupt-vektör-tablosu, CS $\leftarrow$  interrupt-vektör-tablosu
- Interrupt alt program yürütür.
- Interrupt alt programının sonunda IRET ile dönür.
- POP IP
- POPCS
- POPF

## Interrupt Flags

- INTR ucu sadece IF=1 ise kesme oluşturabilir.
- TF=1 ise her komut işledirken sonra single-step kesmesi oluşur (Debug modu)
- STI% set interrupt flag, CLI% clear interrupt flag, TF'in özel komut yok.



PUSHF yapın  
bit manipülasyonu  
yapılabilir.

## Kesme Vektör Tablosunu Değiştirme

Örnek: DIV0 Kesme alt programının adresini değiştirmek.

```
STACKSG SEGMENT PARA STACK 'STACK'
```

```
    DW 12 DUP(?)
```

```
STACKSG ENDS
```

```
DATA$G SEGMENT PARA 'DATA'
```

```
DIZI1      DW 1,2,3,4,5,6,7
```

```
DIZI2      DW 1,0,3,4,5,6,7
```

```
ELEMAN     DW?
```

```
OKOOF      DWO
```

```
OLSEG      DWO
```

```
DATA$G ENDS
```

```
CODE$G SEGMENT PARA 'CODE'
```

```
ASSUME CS:CODE$G, DS:DATA$G, SS:STACK$S
```

```
NEWINT PROC FAR
```

```
    PUSH BP
```

```
    MOV BP, SP
```

```
    MOV AX, [BP+2]
```

```
    ADD AX, 2
```

```
    MOV [BP+2], AX
```

```
    MUV AX, 0FFFFH
```

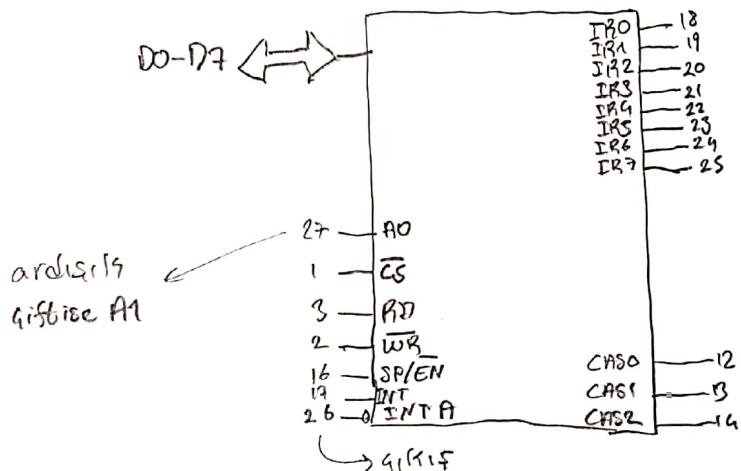
```
    POP BP
```

```
    IRET
```

```
NEWINT ENDP
```

# 8259A Programmable Interrupt Controller

- Programlabilir Kesme Kontrol devresi.
- Kesme isteklerine öncelik atanabilir.
- Tek basına 8 farklı Kesme vektörü sağlar.
- Master (1) / Slave (8) formunda 64 farklı Kesme vektörü sağlar.



- D0-D7 : Data pinleri
- IRO-IR7 : Kesme istek pinleri
- WR, A0, CS
- INT : µp INTREQ ucunca
- INTA : µp INTA ucundan
- A1 : adres vcu
- CAS2-CAS0 : Kaskat segim vgları
- SP/EN : Slave program/enable buffer
- 8259A basic µp sistemlerde tels olurak kullanıllırsa SP/EN vcu 1 olurak ayarlanır. (Master)
- 8259A, initialization command words (ICWs) ve operation command words (OCWs) kullanıllı olarak ayarlanır.

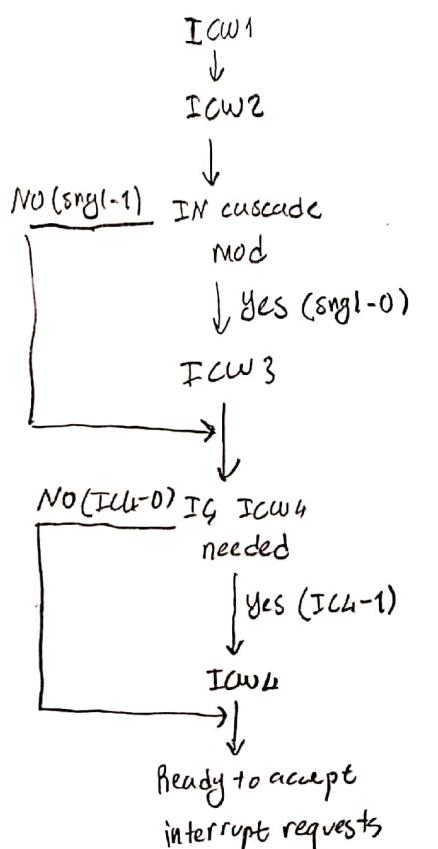
## Kesme işlemler sıralaması

- IR 7-0 ugalarında biri veya birden fazlası 1'e aitken
  - 8259 istekleri değerlendirip, CPU'ya INTR gönderir
  - CPU, INTA ile karşılık verir
  - INTA 8259A tarafına erişince, en yükseliş öncelikli Kesmeye ilişkin ISR biti 1, IRR biti 0 yapılır.
  - CPU ikinci INTA darbesini gönderir, 8259A karşılık olarak Kesme vektör numarasını veriyoluna kayar.
  - AEOI Modunda ISR biti 0 yapılır. (EOI modunda ise ISR bitini 0 yapmak için uygun bir komut bekler)
- end of interrupt*

Automatic  
end of intr

## 8259A - Ayarlama — ICW

CS	AO	initialization
0	0	ICW1
0	1	ICW2, ICW3, ICW4
1	X	Not Address



## ICW1 & ICW2

ADD	D7	D6	D5	D4	D3	D2	D1	DO
0	0	0	0	1	LTIM	0	SN61	IC4
	0 for x86				1 for level trigger 0 for edge trigger		1=single 0=cascade	1=IC4 needed 0=no IC4 needed

ICW1 Select

NMI Kurs  
tetikleme  
↓  
intr vcu  
seviye tetikleme

ADD	D7	D6	D5	D4	D3	D2	D1	DO
1	T7	T6	TS	T4	T3	T2	T1	TO
	T7=TO is the assign to IRO, Vector address for ISR							

→ 8269'un gerektiği kesme tiplerine karşılık gelir.

Örnek: 100000 → IRO=000 → ISR ile kesme isteği gelirse  
 ISR=111 → Mpu'ye 10000100 aktarılır. → 84H

## ICW3 slave & ICW3 master & ICW4

ADD	D7	D6	D5	D4	D3	D2	D1	DO
1	S7	S6	S5	S4	S3	S2	S1	SO
	$S_C = 1$ ise IR input has a slave $S_C = 0$ ise IR input doesn't have a slave							

$S_{NL} = 0$  ise kullanılır.

ADD	D7	D6	D5	D4	D3	D2	D1	DO
1	0	0	0	0	0	ID2	ID1	ID0
	$(ID2\ ID1\ ID0)$ & slave ID							

$S_{NL} = 0$  ise kullanılır.

ADD	D7	D6	D5	D4	D3	D2	D1	DO
1	0	0	0	SFNW	BUF	MS	AEOI	PM
	PM: 1 for 8086							

→ 03H

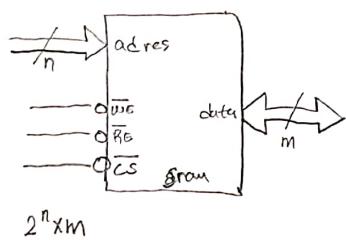
# RAM

SRAM

DRAM

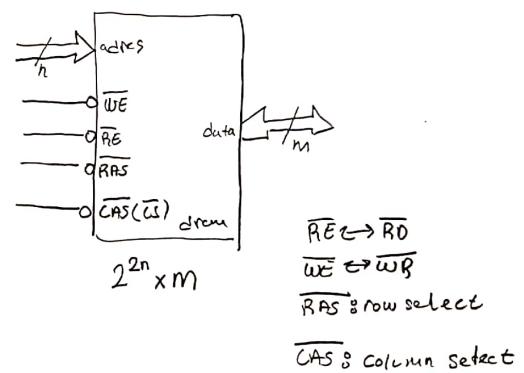
SRAM yazma  $\rightarrow \overline{BL} = 1$  and  $\overline{BL} = 0$   
 $WL = 1$

SRAM



DRAM

- ↳ Kapasitor + transistor
- ↳ lojik devreler günceller
- ↳ SAAM'e göre A katduha uz yur kaplar
- ↳ 0 degrit kayipst saklansır  
1 "guncellmesse kaybedilir"



ROM

8086 reset vektoründe bir

Rum yarışılıktı.

→ CS:IP → FFFFH:0000H

masked Rom

PROM

## EPRROM

EEPRAM

## Flash Memory

