

Mikroişlemci Sistemleri

Dr. Öğr. Üyesi Erkan Uslu

1

YTÜ-CE

Ders Sayfası

- <https://avesis.yildiz.edu.tr/euslu>

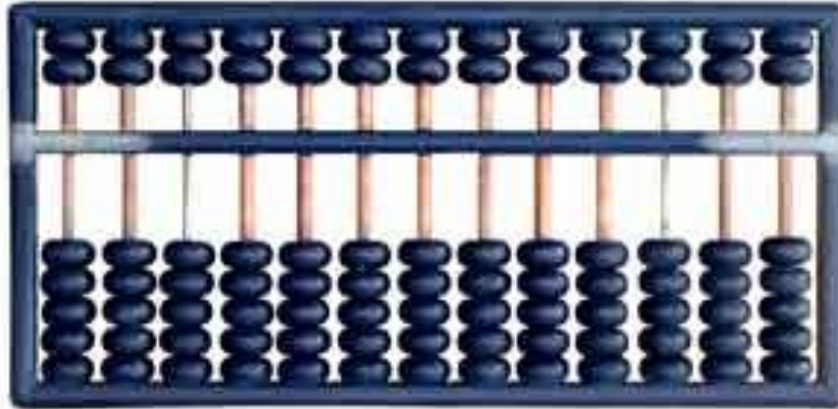
Dersin Konusu

- Genel tanımlar ve karşılaştırmalar
- 8086 μ P minimum mod kullanımında:
 - İç yapı
 - Uç tanımları
 - Yardımcı devreler
- Hafıza erişimi
- Arayüzler
 - 8255
 - 8251
 - 8254
 - DAC, ADC
 - 8259

MİKROİŞLEMCİLERİN TARİHSEL GELİŞİMİ

Mekanik Çağ

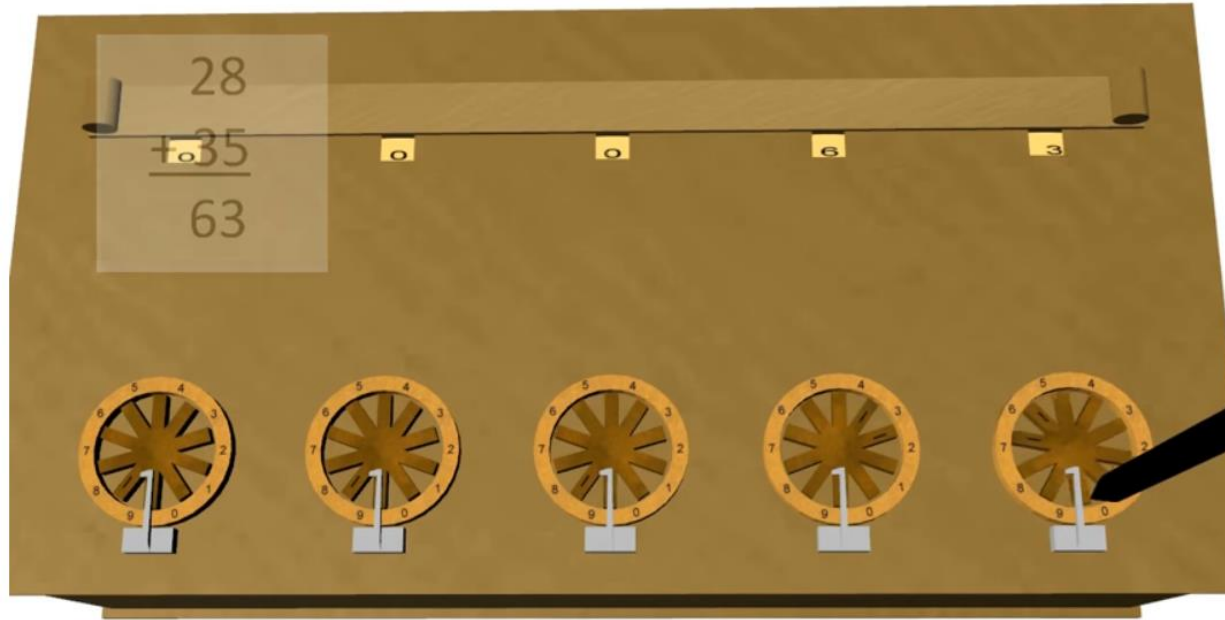
- Hesaplama için araç kullanımı MÖ 500'lü yıllara kadar uzanır
- Babilliler abaküs'ü kullandılar
 - İlk mekanik hesap
 - Boncuklar ile
- Çin abaküsü



Mekanik Çağ

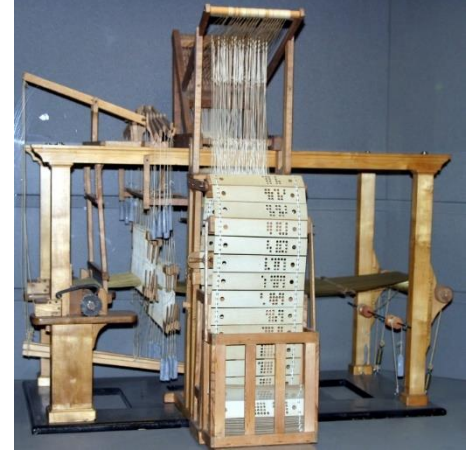
- 1642 Blaise Pascal çark ve dişlilerden oluşan bir hesap makinesi geliştirdi (Pascaline)
 - Her dişli 10 diş içeriyordu
- İlk dişli bir tam tur attığında ikinci dişli 1 adım ilerliyor
- <https://www.youtube.com/watch?v=3h71HAJWnVU>

Mekanik Çağ



Mekanik Çağ

- 1800'lü yıllarda dişli mekanik makineler ile hesaplama ve veri işleme için uygulamalar gerçekleştirildi
- 1801'de Joseph Jacquard dokuma makinesinde desen oluşturmak için delikli kartları kullandı



Mekanik Çağ

- 1823'te Charles Babbage Kraliyet Donanması için seyir tablolarını oluşturmak için programlanabilir hesaplama makinesini geliştirmesi için görevlendirildi (Analytical Engine)
- Buhar gücüyle çalışan, yeniden programlanabilir, delikli kartlar ile giriş yapılan bir konsept
- Tamamlanamadı ama fikri olarak çığır açtı

Elektriksel Çağ

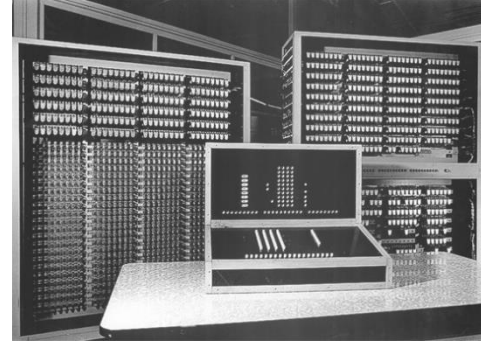
- 1800'lü yıllarda elektrik motor Michael Farady tarafından tasarlandı
- <https://www.youtube.com/watch?v=fcb-zjbsmBk>
- Pascaline'in elektrik motorlu versiyonları geliştirildi
- 1889 yılında Herman Hollerith elektrik motorlar ile sürülen veri işleyen bir makine geliştirdi
- Bu makine Amerika 1890 nüfus sayımı sonuçlarının istatistik çıkarımı için kullanıldı
- Veri girişi delikli kartlar kullanılmıştı

Elektriksel Çağ

- 1896'da Hollerith nüfus sayımlarında kullanılacak makineleri üretmek amacıyla Tabulating Machine firmasını kurdu
- Bir takım firma birleşmeleri sonucunda firma International Business Machines Corporation olarak isim aldı (IBM Inc.)

Elektriksel Çağ

- Elektrik-mekanik makineler 1941 yılına kadar yaygın olarak kullanıldı
- Uçak ve füze tasarımı için hesaplamaları yapmak amacıyla, Konrad Zuse röle temelli ilk elektromekanik bilgisayarı geliştirdi (Z3)
- Z3 5.33Hz frekansında çalışıyordu

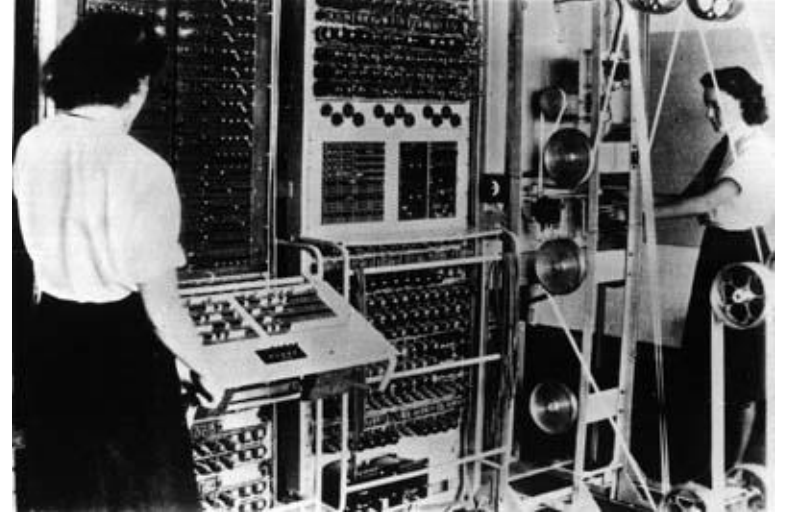


Elektriksel Çağ

- Konrad Zuse Z3'ü geliştirmeden önce sisteminin mekanik (Z1) ve öncül elektro-mekanik versiyonlarını da geliştirmişti

Elektronik Çağ

- İlk elektronik bilgisayar olarak gösterilen tasarım Alan Turing tarafından geliştirilmiştir (Colossus)
- Elektronik komponent olarak vakum tüpleri kullanılmıştır

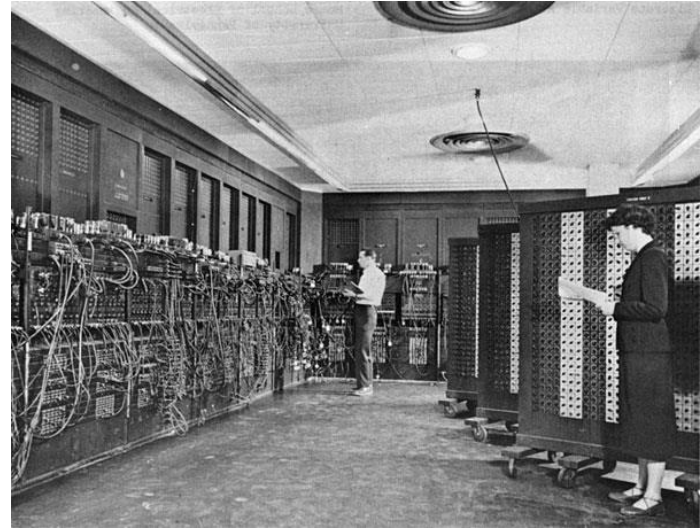


Elektronik Çağ

- Colossus Alman şifreleme sistemi Enigma'nın çözülmesi için geliştirilmişti
- Başka problemlere uyarlanamıyordu
- Programlanabilir değildi
- Bu yapıyla bir sabit programlı bilgisayar yapısındaydı (special purpose computer)

Elektronik Çağ

- Genel amaçlı programlanabilir elektronik bilgisayar 1946 yılında Pensilvanya Üniversitesinde geliştirildi
- Electronic Numerical Integrator and Calculator (ENIAC)
 - Büyük bir makine
 - 17000 vakum tüpü
 - 800 km kablo
 - 30 ton ağırlık
 - Saniyede 100000 işlem



Elektronik Çağ

- ENIAC'ın programlanması kablo bağlantılarının değiştirilmesi ile yapılıyordu
 - Programlanması pek çok işçinin birkaç günlük çabasını gerektiriyordu
- Vakum tüplerinin kısa ömürlerini dolayısıyla sık bakım gerektiriyordu

Elektronik Çağ

- 1947 yılında Bell laboratuvarlarında John Bardeen, William Shockley, Walter Brattain tarafından transistör geliştirildi
- Sonrasında 1958 yılında entegre devreler geliştirildi
- 1960'larda sayısal entegre devreler geliştirildi
- 1971 yılında ise Intel firması tarafından ilk mikroişlemci geliştirildi
- Federico Faggin, Ted Hoff, Stan Mazor 4004 mikroişlemcisini geliştirdi

Mikroişlemci Çağı

- İlk mikroişlemci Intel firmasının geliştirdiği 4004'tür
 - 4 bitlik mikroişlemci
 - Adresleme kapasitesi: 4096 x 4 bit
 - Komut seti 45 komuttan oluşuyor
 - 30 gram ağırlığında
 - Saniyede 50000 işlem (30 ton ENIAC saniyede 100000 işlem)
 - Oyun ve küçük kontrol sistemlerinde kullanıldı
 - RTL (direnç –transistör lojisi ile tasarlanmış)
- Sonrasında daha yüksek frekanslı 4040 mikroişlemci geliştirildi

Mikroişlemci Çağı

- 1971'de Intel 8008 mikroişlemciyi tanıttı
 - 8-bitlik bir mikroişlemci
 - 16KB adresleme kapasitesi
 - Toplamda 48 farklı komut yürütebiliyordu
- Mikroişlemcilerin daha karmaşık sistemlerde kullanımı mümkün oldu

Mikroişlemci Çağı

- 1973 yılında Intel 8080 mikroişlemciyi tanıttı
- İlk modern 8 bitlik mikroişlemci olarak kabul edilir
- 8080
 - 64KB adresleme kapasitesi
 - 8008'e göre yaklaşık 10 kat daha hızlı
 - TTL (transistör- transistör lojiği ile tasarlamış)

Mikroişlemci Çağı

- 8080'in sunumundan 6 ay sonra Motorola MC6800 mikroişlemciyi sundu
- Diğer firmalar tarafından da 8 bitlik mikroişlemciler piyasaya sunuldu
- Fairchild – F8, MOS tech – 6502, National Semiconductors – IMP8, Zilog – Z8
- 1974'te MITS Altair 8800 sunuldu
 - 1975'te Bill Gates ve Paul Allen Altair 8800 için BASIC dilini geliştirdi

Mikroişlemci Çağı

- 1977 yılında Intel 8085 mikroişlemciyi sundu
- Intel'in son 8 bitlik mikroişlemcisi
- Saniyede 769230 işlem
- Dahili saat üretici kullanımı
- Entegre komponent sayısında artış

Modern Mikroişlemciler

- 1978 yılında 8086 ve bir yıl sonra 8088 mikroişlemciler tanıtıldı
- 16 bitlik mikroişlemciler
- Komut yürütme süresi 400 ns (saniyede 2,5 milyon işlem)
- Adresleme kapasitesi 1MB
- 4 veya 6 byte'lık komut kuyruğu mevcut (sıradaki birkaç komutun birlikte okunması)
- Çarpma bölme gibi komutların sunulması
- Varyasyonları ile 20000'i bulan komut sayısı

Modern Mikroişlemciler

- 8086/8088 CISC (complex instruction set computers) mimarisindedir
- Yazmaç sayısında artış söz konusu
- 8086 ve 8088: 20 adet adres ucuna sahip
- 8086: 16 veri ucuna sahip
- 8088: 8 veri ucuna sahip

Modern Mikroişlemciler

- 1983 yılında 80286 tanıtıldı
- 16MB adresleme kapasitesine sahip
- Komutlar 8086'ya benzer şekilde olmakla birlikte 16MB hafıza için komutlarda güncelleme var
- Saat frekansı 8MHz → saniyede 4 milyon işlem

Modern Mikroişlemciler

- 1986 yılında 80386 sunuldu
- 32 bit adres yolu, 32 bit veri yolu
- 4GB adresleme kapasitesi

Genel Tanımlar ve Karşılaştırmalar

- ALU
- Register
- CPU
- μ P
- μ C
- SoC
- Harvard arc.
- Von Neumann arc.
- CISC
- RISC
- EPIC
- Little endian
- Big endian
- Data bus
- Address bus
- Control bus
- Accumulator

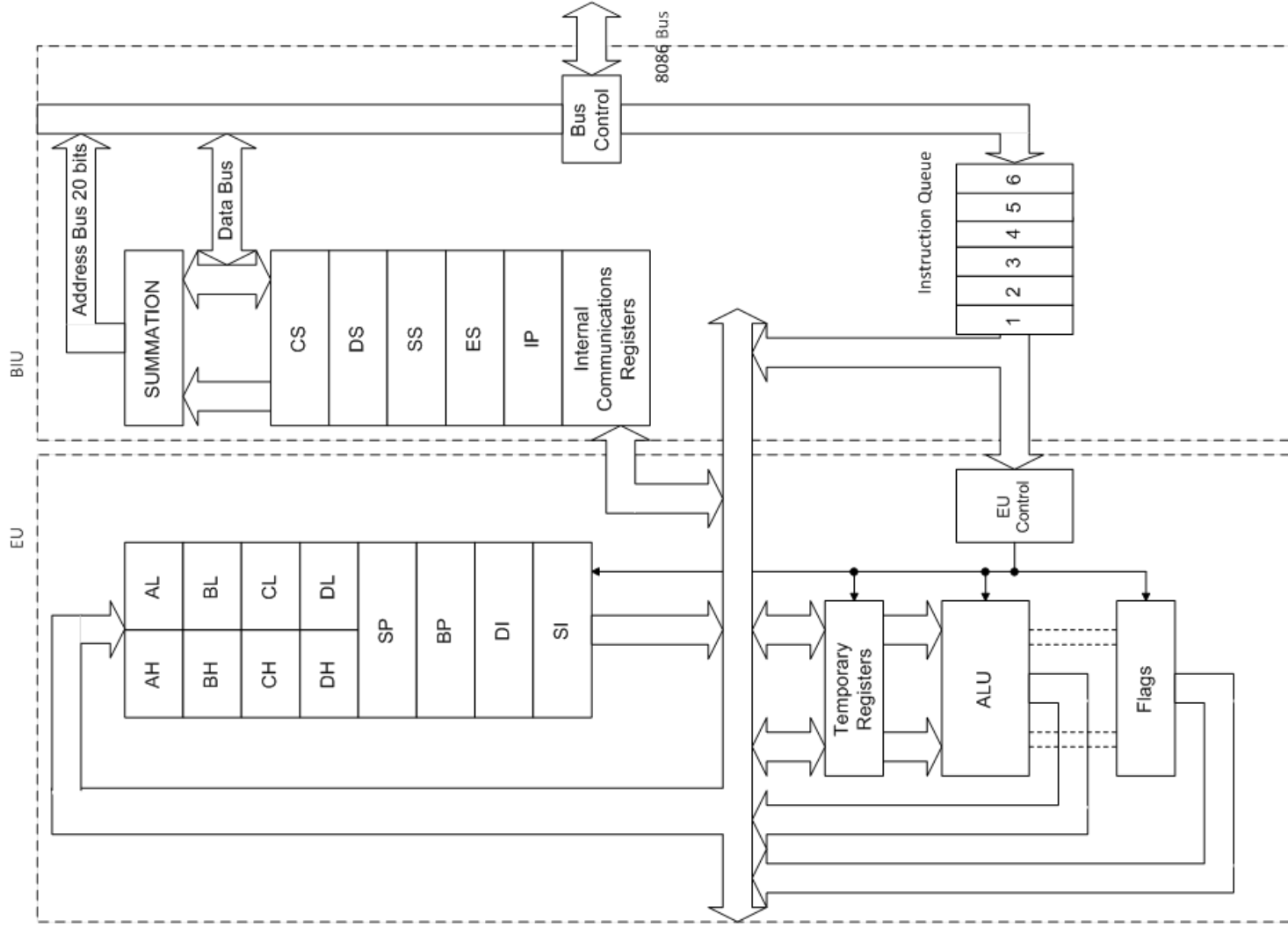
Genel Tanımlar ve Karşılaştırmalar

- 8086 (1978)
 - 16 bit veri yolu, 16 bit register, 20 bit adres yolu
- 8088 (1979)
 - 8 bit veri yolu, 16 bit register, 20 bit adres yolu
- 80286 (1982)
 - 16 bit veri yolu, 16 bit register, 24 bit adres yolu
- 80386 (1985)
 - 32 bit veri yolu, 32 bit register, 32 bit adres yolu

Genel Tanımlar ve Karşılaştırmalar

- Real mode
- Protected mode
- Virtual mode
- Prefetch queue
- Pipeline
- Offset memory model
- Time multiplexing
- Coprocessor
- Cache

8086 İç Yapısı



Yazmaçlar

- AX
 - AL
 - AH
- BX
 - BL
 - BH
- CX
 - CL
 - CH
- DX
 - DL
 - DH
- SP
- BP
- SI
- DI
- SS
- CS
- DS
- IP
- ES

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

REAL – PROTECTED MOD

Real Mod Hafıza Adresleme

- 8086 real modda hafıza adresleme yapar
- Real modda sadece 1MB alan adreslenebilir
- 8086 hafıza uzayı 1MB (20 adres ucu → 1MB)
- Tüm bilgisayarlar açıldığında real modda açılır

Real Mod Hafıza Adresleme

- Real modda segment adres ve ofset adres değerlerinin birleşimiyle hafızada istenen alana erişilir
- Bir segment değeri 64KB'lık alanı gösterir (NEDEN?)
- Ofset değeri 64KB'lık alan içinde bir yeri gösterir
- Örnek: Segment değeri 1000H, ofset değeri 2000H ise mikroişlemcide erişilen fiziki adres ne olur?
- $1000H \times 10H + 2000H = 12000H$ (1000H:2000H)

Real Mod Hafıza Adresleme

- Varsayılan Segment ve Ofset yazmaçları
- Program hafızasına erişimde CS:IP birlikte kullanılır
- Yığın (stack) erişiminde SS:SP veya SS:BP kullanılır
- Veri erişiminde DS:BX, DS:DI, DS:SI kullanılır
- String işlemlerinde ES:DI ile kullanılır

Protected Mod Hafıza Adresleme

- 1MB'tan daha geniş hafızayı adreslemede kullanılır
- Segment adres değeri bir tablonun (descriptor table) bir satırını gösterir
- Bu tablonun herbir gözünde kullanım için ayrılmış segmentin boyu, yeri, erişim izinleri yazılıdır
- Tabloların global ve yerel versiyonları vardır
- Global descriptor tablosunda tüm programlar için kullanılabilecek segment bilgisi yer alır
- Yerel descriptor tablosunda ise uygulamaya özel segment bilgisi yer alır

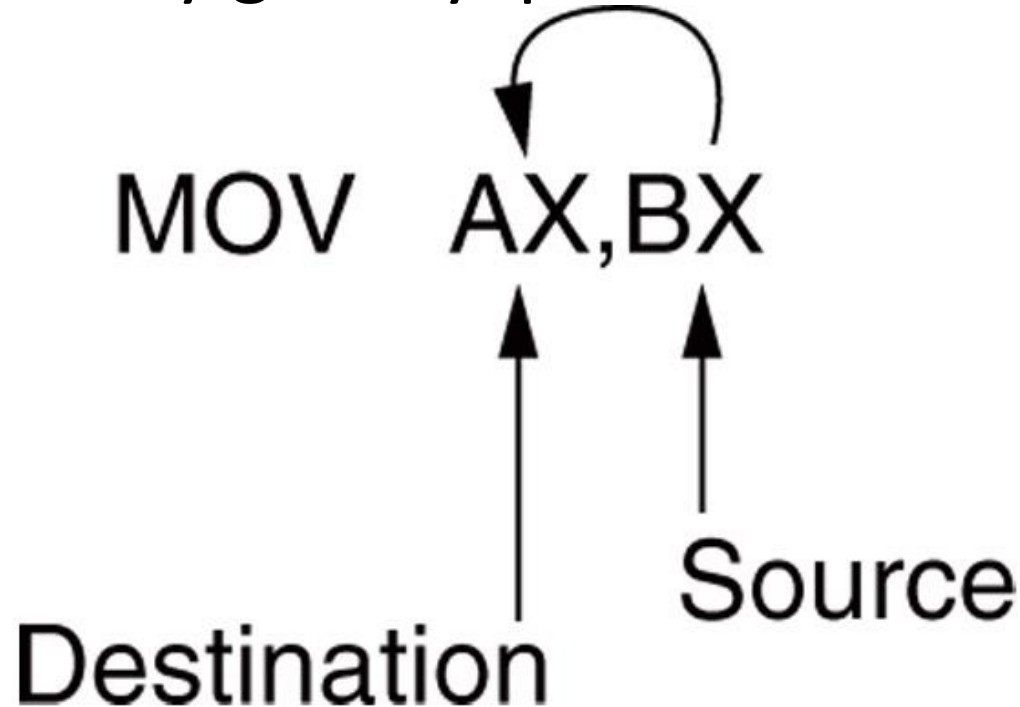
ADRESLEME MODLARI

Adresleme Modları

- Verimli assembly programları geliştirebilmek için komutlar ile birlikte kullanılan adresleme modlarının bilinmesi gerekmektedir.
- Veri adresleme modları
- Program hafızası adresleme modları

Veri Adresleme Modları

- 8086 assembly genel yapısı



Veri Adresleme Modları

- 8086 assembly genel yapısı
 - $AX \leftarrow 1234H$
 - $BX \leftarrow ABCDH$
 - $MOV\ AX, BX$
 - $AX \leftarrow ABCDH$
 - $BX \leftarrow ABCDH$
- $MOV\ DST, SRC$
 - $DST \leftarrow SRC$

Yazmaç Adresleme (Register Addressing)

- Yazmaç Adresleme (Register Addressing)
- 8 bitlik AL, AH, BL, BH, CL, CH, DL ,DH yazmaçları kullanılabilir
- 16 bitlik AX, BX, CX, DX, SP, BP, SI, DI yazmaçları kullanılabilir
- Yazmaç adreslemede kullanılan yazmaç genişlikleri uyumlu olmalıdır

MOV BX, CX

Hemen Adresleme (Immediate Addressing)

- Sabit değer atamayı ifade eder
- 16 bit veya 8 bit sabit değer atama söz konusu olabilir

MOV AL, 0F2H

MOV CX, 100

MOV BL, 01010101B

MOV AH, 'A' ;ASCII A karakteri AH yazmacına atanır

Doğrudan Adresleme (Direct Addressing)

- Erişilecek hafıza gözünün doğrudan gösterildiği durumdur

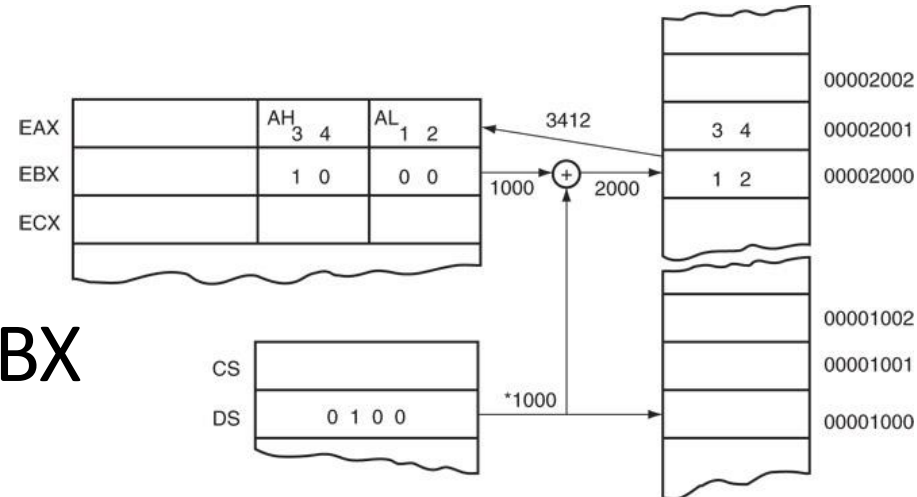
MOV AL, DATA ; DATA bir etiket olup assembler bunu karşılık gelen adres değeri ile değiştirir

MOV BX, [1234H] ; $BX \leftarrow DS:1234H$

Yazmaç Dolaylı Adresleme (Register Indirect Addressing)

- BP (SS ile), BX, DI ve SI (DS ile) yazmaçları ile kullanılabilir
- Hafıza ofset değeri bir yazmaçta saklanır

MOV AX, [BX] ; $AX \leftarrow DS:BX$



Dolaylı Adresleme (Indirect Addressing)

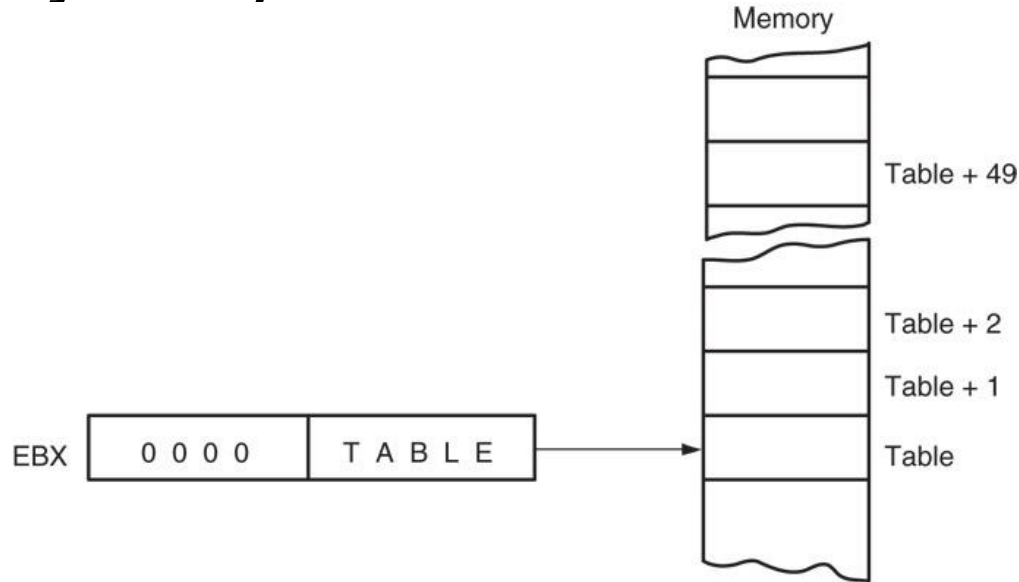
- Dolaylı adreslemelerin bir kısmında BYTE PTR, WORD PTR gibi özel tanımlayıcılar vermek gerekir
- Arttırma komutu olan INC, dolaylı adresleme ile hafızada bir Word mü yoksa Byte değeri mi arttıracığını bilemez

INC WORD PTR [BX]

- Dolaylı adreslemede operandlardan birisi yazmaç ise BYTE PTR, WORD PTR'ye gerek yoktur (NEDEN?)

Yazmaç Dolaylı Adresleme (Register Indirect Addressing)

- Dizi olarak tutulan veriye sıralı erişimde yazmaç dolaylı adresleme kullanımı uygundur



Base+Index Addressing

- Temelde bir dolaylı adresleme modudur
- Base yazmacı (BX veya BP) işlem yapılacak hafıza konumunun başlangıcını göstermek için kullanılır
- Index yazmaçları (DI veya SI) verinin bu başlangıç adresine görece yerini tutmak için kullanılır

MOV DX, [BX+DI]

Yazmaç Görelî Adresleme (Register Relative Addressing)

- Base (BP veya BX) veya Index (DI, SI) yazmaçlarının bir sabit ofset değeri ile kullanılmasını ifade eder

MOV AX, [BX+100H]

Base Relative + Index Addressing

- İki boyutlu veri adresleme için uygundur

MOV AX,[BX + SI + 100H]

Program Hafızası Adresleme Modları

- Program akışı sırasında fonksiyon çağırma, koşullu ve koşulsuz dallanma komutları ile farklı program hafızası adresleme modları kullanılır
- Doğrudan (direct)
- Göreli (relative)
- Dolaylı (indirect)

Doğrudan Program Hafızası Adresleme

- Doğrudan bir program adresine ulaşmak için kullanılır
- Mevcut kod segmentinden farklı bir kod segmentine geçiş sağlayacağı için segmentler-arası bir işlemdir
- Hem CS hem de IP değeri uygun şekilde değiştirilir

JMP 200H:300H ; CS $\leftarrow$ 200H, IP $\leftarrow$ 300H

CALL 200H:300H

Görelî Program Hafızası Adresleme

- Mevcut IP yazmacı değerine göre hangi program hafızasının adresleneceğini ifade eder
- JMP komutu 1 byte veya 2 byte işaretli sabit değerli operand kabul eder

JMP 100

JMP 0FFH ; IP değeri 1 azalır (NEDEN?)

JMP 1000H

Dolaylı Program Hafızası Adresleme

- CALL ve JMP komutları ile kullanılır

JMP BX

CALL [BX]

Yığın Adresleme Modları

- Tüm yazmaçlardan yığına veri basılabilir
- CS hariç tüm yazmaçlara yığından veri çekilebilir

PUSH CS ; çalışır

POP CS ; assembler hatası verir

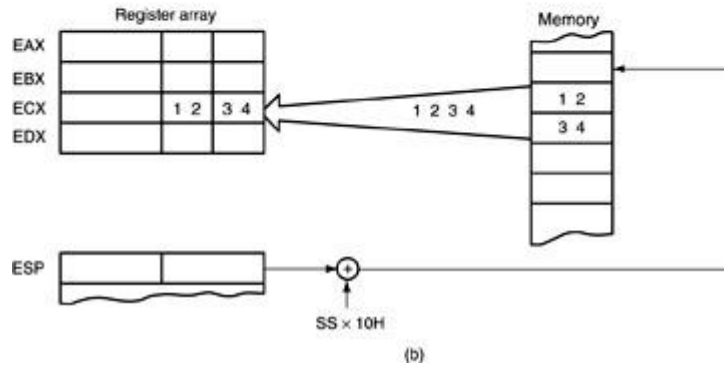
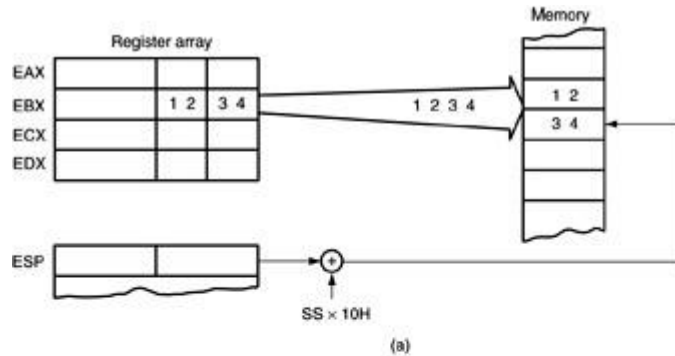
Yığın Adresleme Modları

- 8086'da yığın geçici veri saklamak için ve fonksiyonlardan dönüşlerde dönüş adreslerini saklamak için kullanılır
- Yığın LIFO mantığında çalışır (Last In First Out)
- Yığın ile ilgili PUSH ve POP komutları kullanılır
- PUSH yığına WORD basar, POP yığından WORD çeker
PUSH BL ; (DOĞRU MU?)
- Yığın adresleme için SS:SP ikilisi kullanılır

Yığın Adresleme Modları

- SP yazmacı programcının tanımladığı yığının genişliğini gösterecek 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

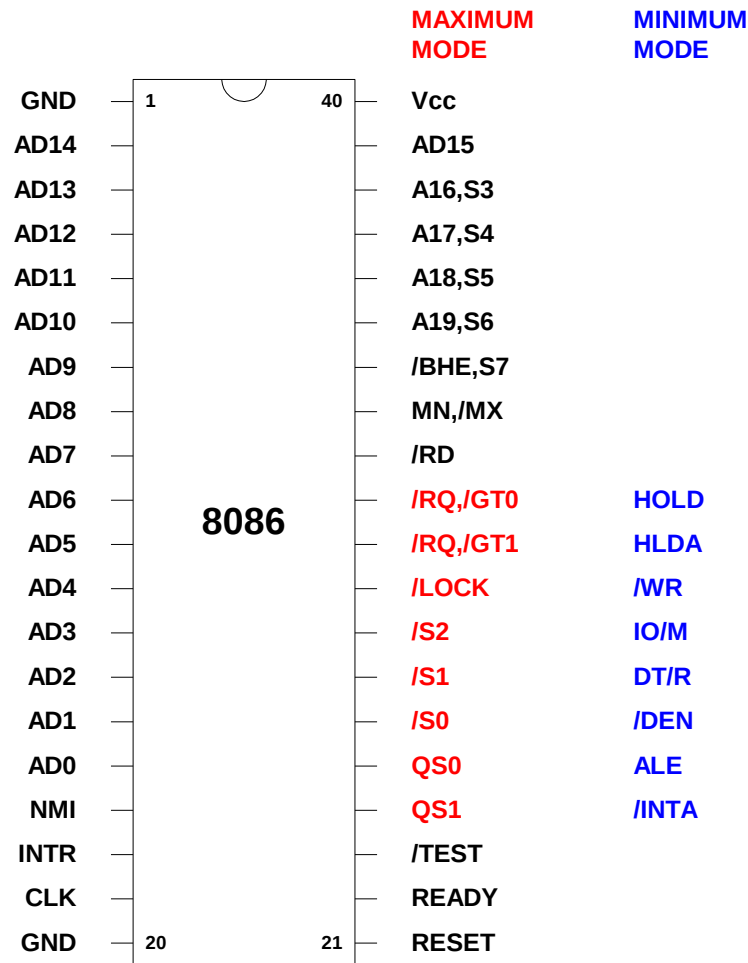
Yığın Adresleme Modları



8086

Uç

Tanımları



8086 Uç Tanımları

- **$AD_{15}-AD_0$:** (I/O-3)

The 8086 address/data bus lines compose the upper multiplexed address/data bus on 8086. These lines contains address bits whenever ALE is logic 1. These pins enter a high-impedance state whenever a hold acknowledge occurs.

8086 Uç Tanımları

- $A_{19}/S_6-A_{16}/S_3$: (0-3)

The address/status bus bits are multiplexed to provide address signals $A_{19}-A_{16}$ and also status bits S_6-S_3 . The pins also attain a high-impedance state during the hold acknowledge. S_4 and S_3 show which segment is accessed during the current bus cycle.

8086 Uç Tanımları

- $\overline{RD}$: (0-3)

Whenever the read signal is logic 0, the data bus is receptive to data from the memory or I/O devices connected to system.

- READY: (1)

This input is controlled to insert wait states into the timing of the microprocessor.

- READY=0: μP enters into wait states and remain idle
- READY=1: It has no effect on operation of μP

8086 Uç Tanımları

- $\overline{TEST}$: (I)

The test pin is an input that is tested by the WAIT instruction

- NMI: (I)

The non-maskable interrupt input is similar to INTR except that the NMI does not check to see if IF flag bit is a logic 1. This interrupt input uses interrupt vector 2.

8086 Uç Tanımları

- RESET: (I)

The reset input causes the μ P to reset itself if this pin is held high for a minimum four clocking periods. It begins executing instructions at memory location FFFF0H and disables future interrupts by clearing the IF flag bit.

8086 Uç Tanımları

- $MN / \overline{MX}$: (1)

Minimum/maximum mode pin select.

- $\overline{BHE} / S_7$: (0-3)

BHE pin is used to enable the most sig. data bus bits (D_{15} - D_8) during a read or write operation.

8086 Uç Tanımları

- $M / \overline{IO}$: (O-3)

The pin selects memory or I/O. This pin indicates that the microprocessor address bus contains either a memory address or an I/O port address.

- $\overline{WR}$: (O-3)

This line indicates that 8086 is outputting data to a memory or I/O device.

8086 Uç Tanımları

- $\overline{INTA}$: (0-3)

The interrupt acknowledge signal is a response to the INTR input pin. This pin is normally used to gate the interrupt vector number onto the data bus in response to an interrupt request.

- ALE : (0)

Address latch enable shows that the 8086 address/data bus contains address information. This address can be a memory address or an I/O port number.

8086 Uç Tanımları

- $DT / \overline{R}$: (O-3)

The data transmit/receive signal shows that the microprocessor data bus is transmitting or receiving data.

- $\overline{DEN}$: (O-3)

Data bus enable activates external data bus buffers.

8086 Uç Tanımları

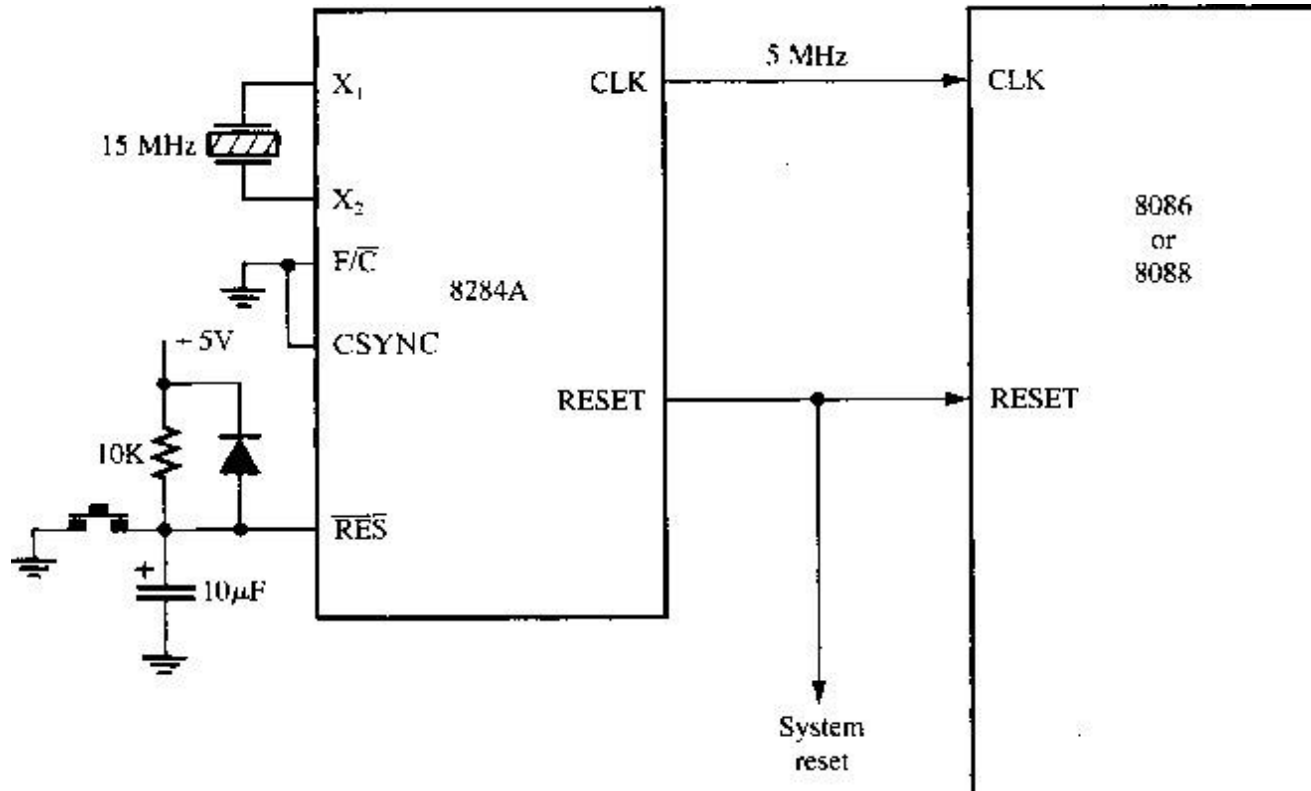
- HOLD : (I)

The hold input requests a direct memory access (DMA). If the HOLD signal is logic 1, the microprocessor stops executing software and places its address, data and control bus at the high-impedance state.

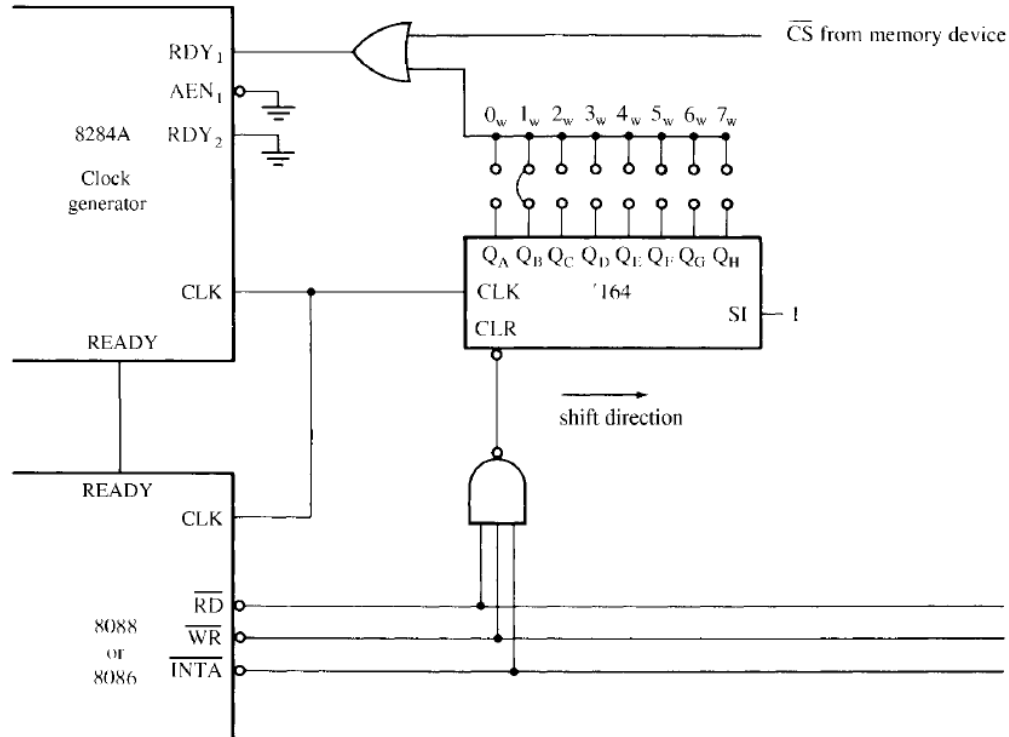
- HLDA : (O)

Hold acknowledge indicates that the 8086 microprocessor entered the hold state.

Clock Generator (8284A)



Clock Generator (8284A)

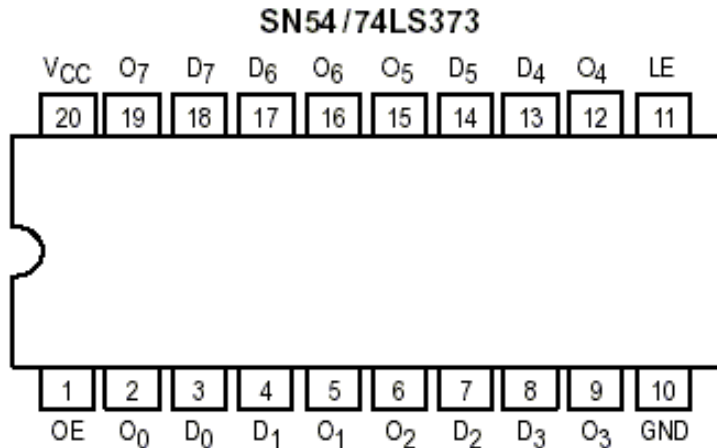


Bus Buffering & Latching

- The address/data bus of the 8086/8088 is multiplexed (shared) to reduce the number of pins required for the integrated circuit.
- Memory & I/O require the address remain valid and stable throughout a read/write cycle.
- If buses are multiplexed, the address changes at the memory and I/O, causing them to read or write data in the wrong locations

Bus Buffering & Latching

- 74LS373 Octal Transparent Latch with 3-state Outputs



LS373

D _n	LE	OE	O _n
H	H	L	H
L	H	L	L
X	L	L	Q ₀
X	X	H	Z*

H = HIGH Voltage Level

L = LOW Voltage Level

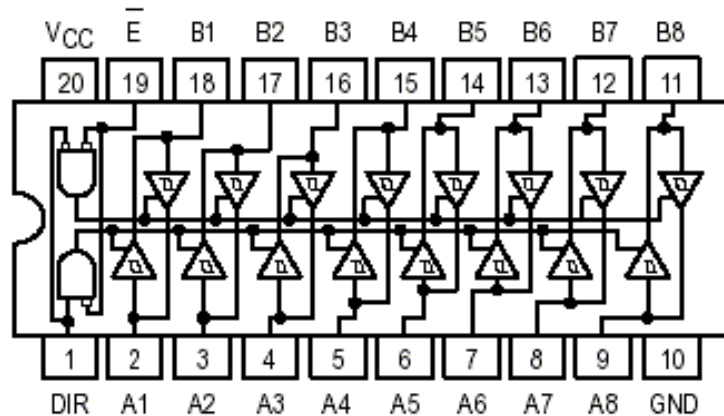
X = Immaterial

Z = High Impedance

Bus Buffering & Latching

- 74LS245 Octal Bus Transceiver

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)



TRUTH TABLE

INPUTS		OUTPUT
$\overline{E}$	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

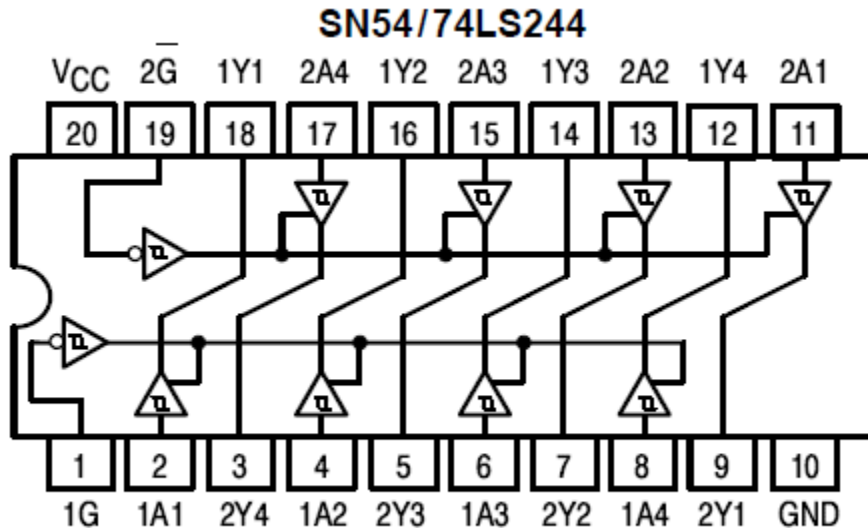
H = HIGH Voltage Level

L = LOW Voltage Level

X = Immaterial

Bus Buffering & Latching

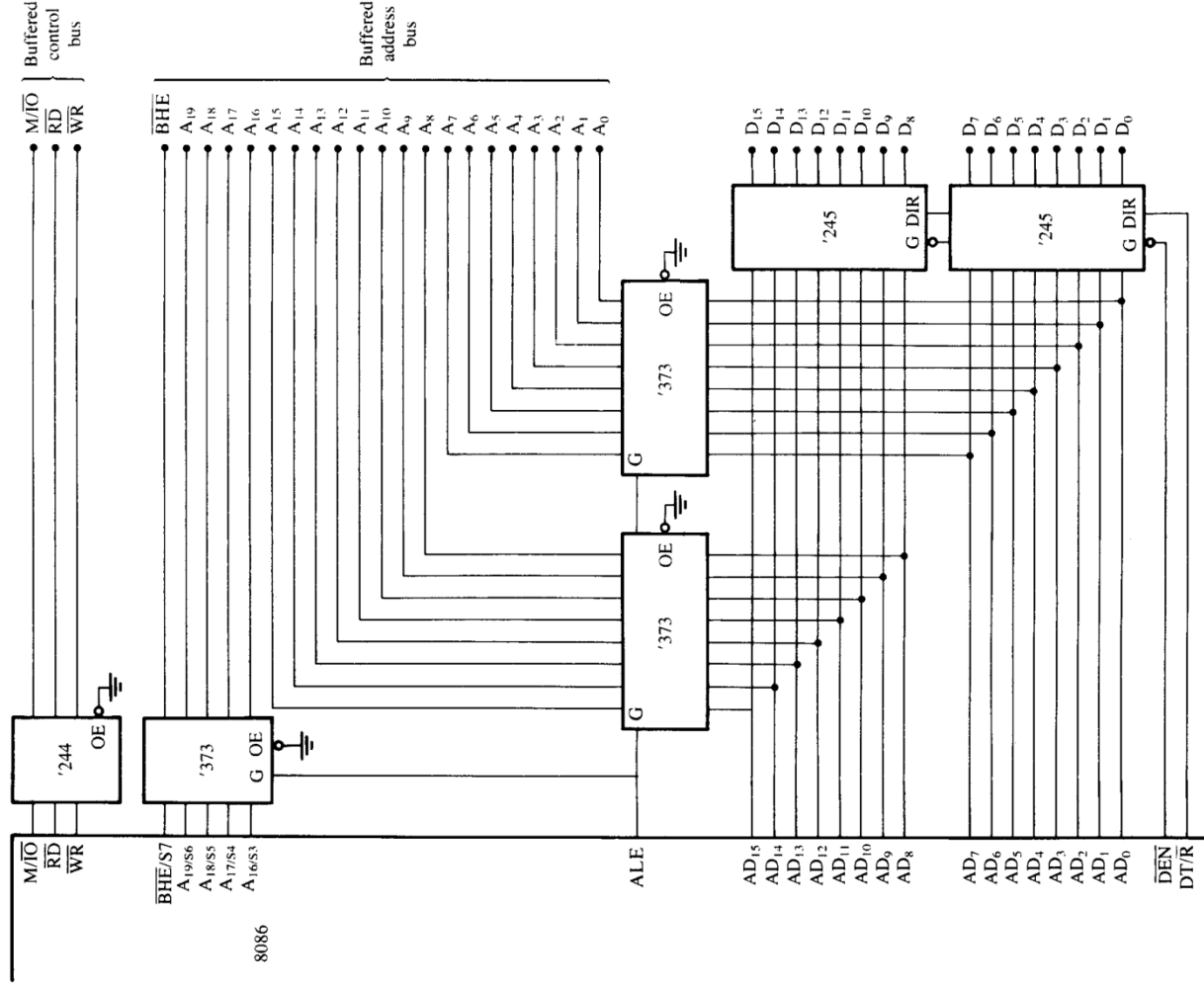
- 74LS244 Octal Buffer



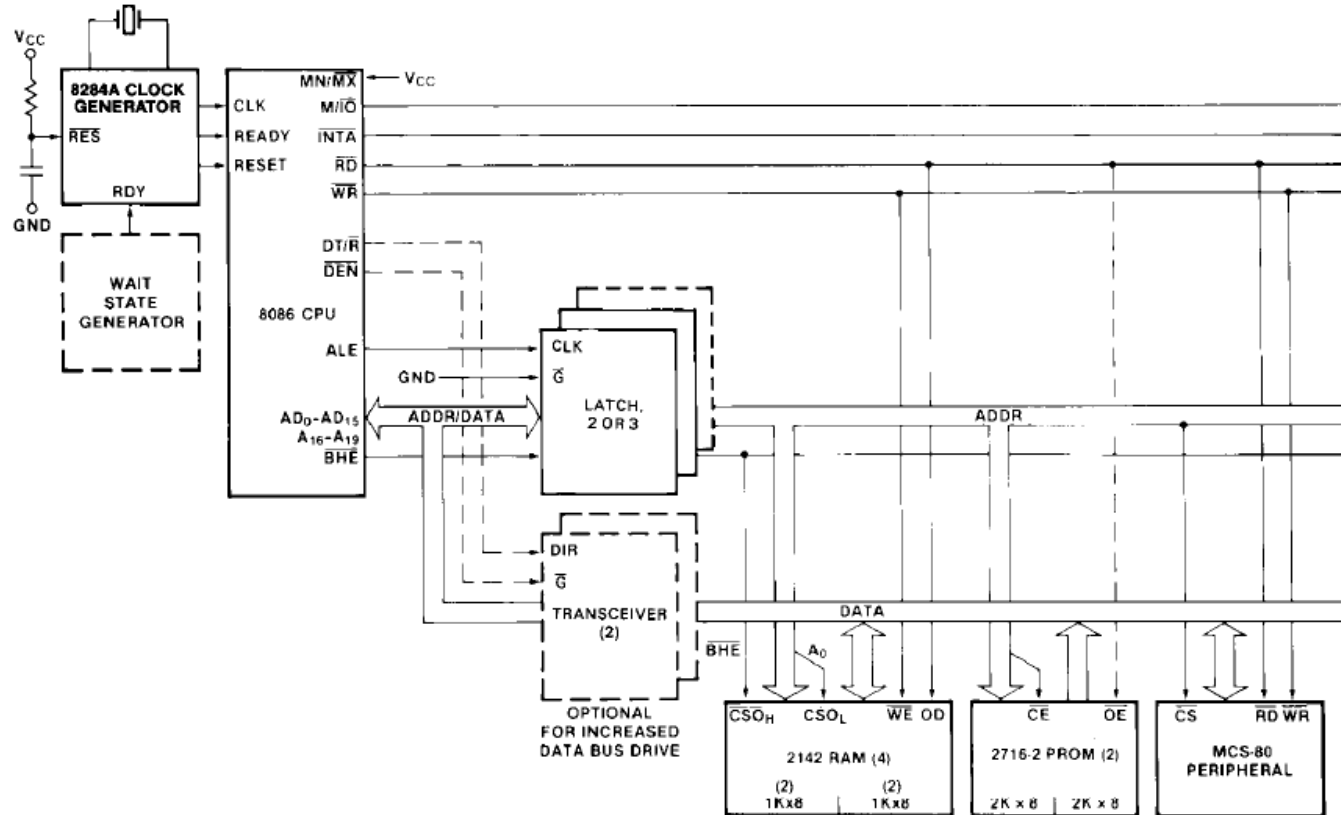
SN54/74LS244

INPUTS		OUTPUT
1G, 2G	D	
L	L	L
L	H	H
H	X	(Z)

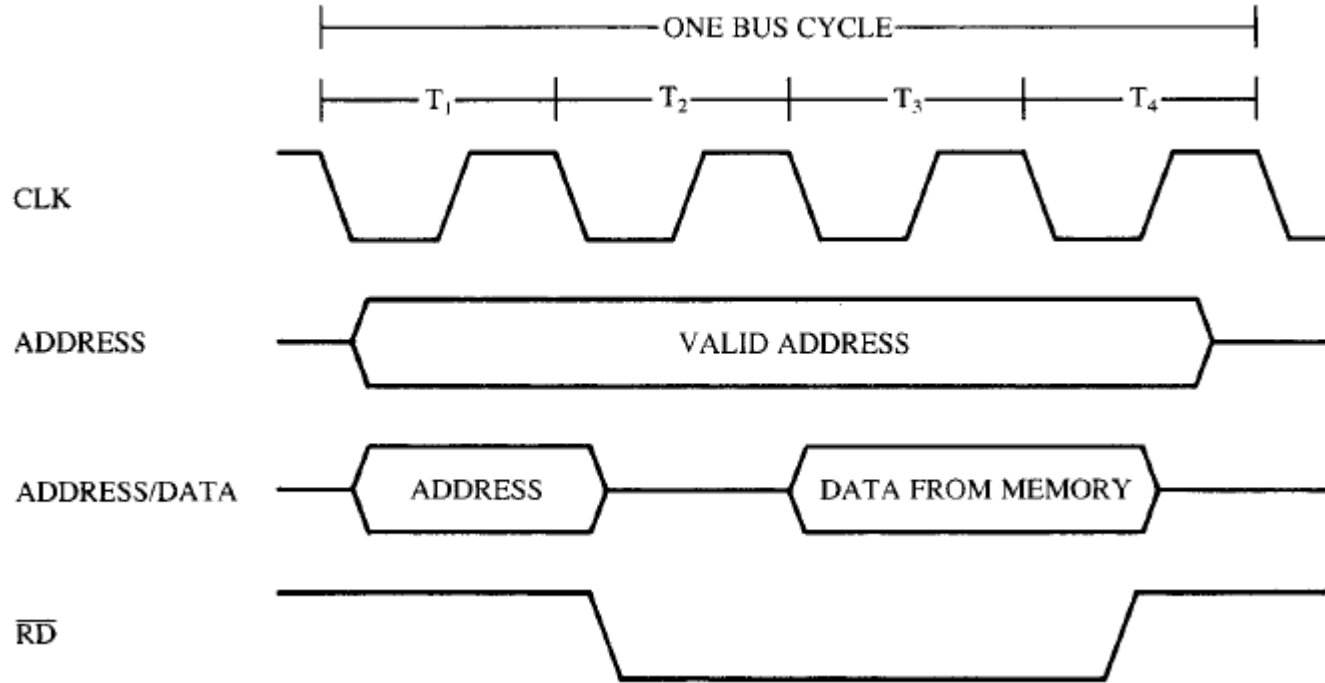
Bus Buffering & Latching



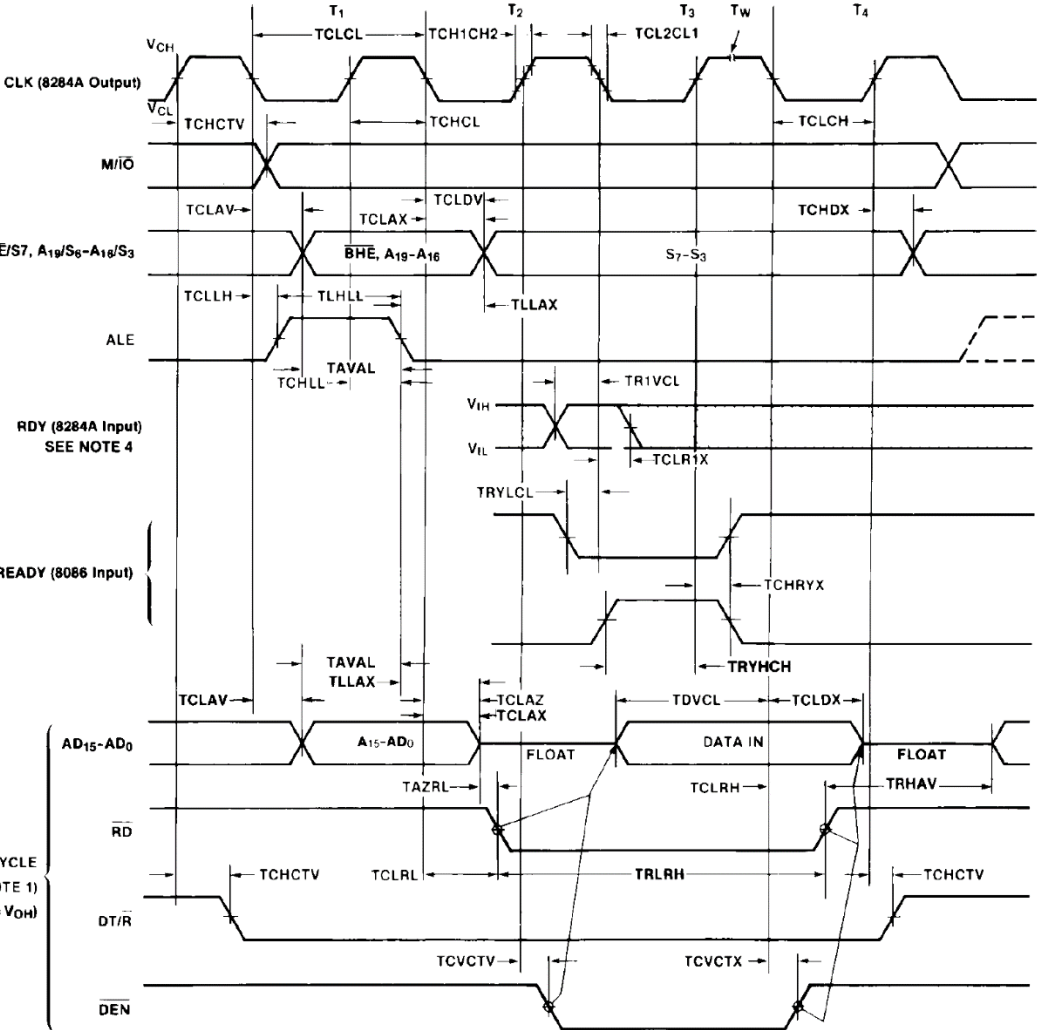
Bus Buffering & Latching



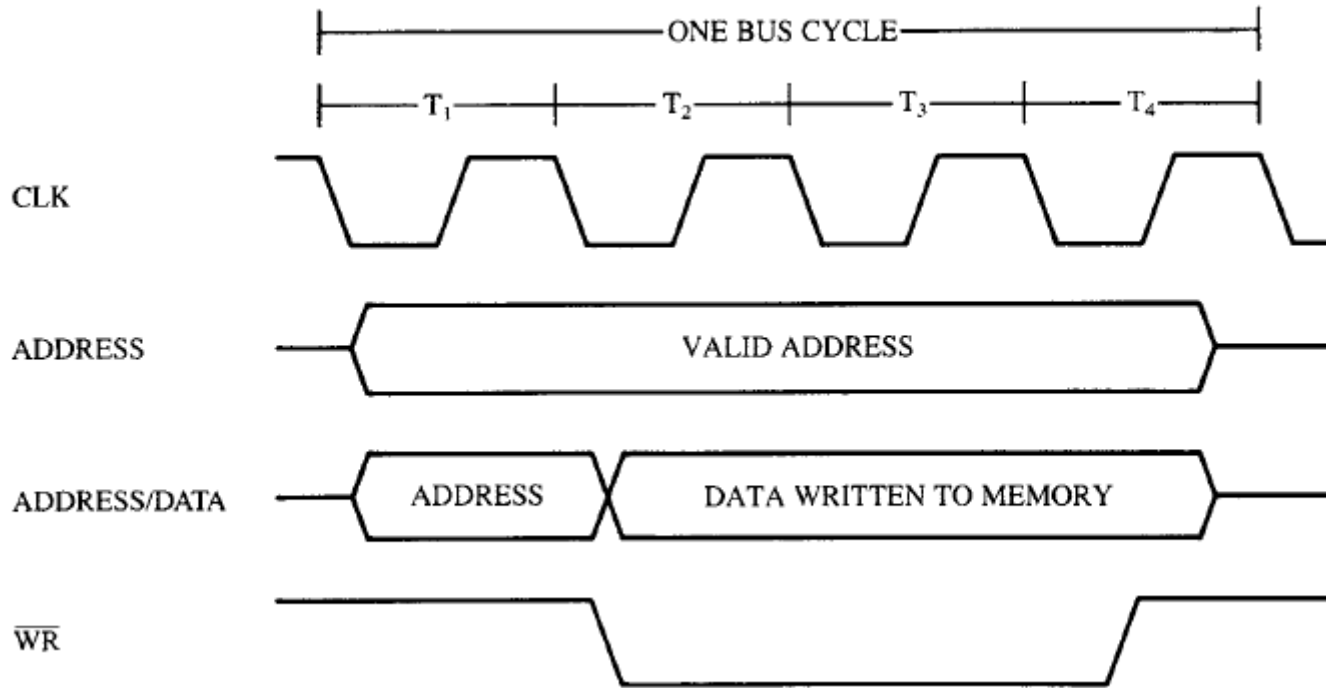
Bus Timing – Simplified Read



Read



Bus Timing – Simplified Write



Bus Timing Write

