Lecture 21: 80x86 Memory Interfacing

Seyed-Hosein Attarzadeh-Niaki

Based on the slides by Hongzi Zhu and Barry Brey

Review

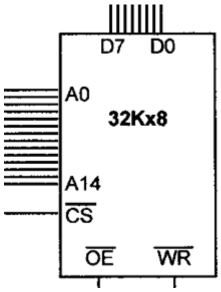
- Assembly programming
  - Addition and subtraction
  - Multiplication and division (unsigned)
  - BCD arithmetic
  - Rotate instructions

Outline

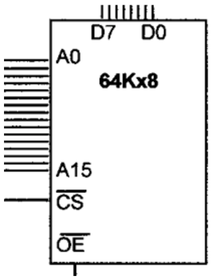
- Memory and memory interfacing
  - Memory chips
  - Address decoding
  - Data integrity
  - 8086 memory interfacing

Review on Memory Chips

- Key concepts
  - Capacity
  - organization (the number of locations X the size of addressable unit)
  - access time
- Packaging



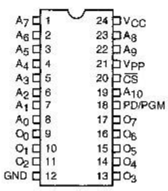
RAM chip



ROM chip

2716: 2K × 8 EPROM

PIN CONFIGURATION



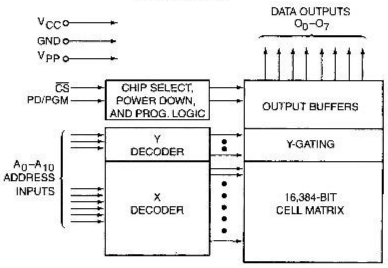
PIN NAMES

A_0–A_{10}	ADDRESSES
PD/PGM	POWER DOWN/PROGRAM
$\overline{CS}$	CHIP SELECT
O_0–O_7	OUTPUTS

MODE SELECTION

MODE \ PINS	PD/PGM (18)	$\overline{CS}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Deselect	Don't care	V_{IH}	+5	+5	High Z
Power Down	V_{IH}	Don't care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

BLOCK DIAGRAM



2716: 2K × 8 EPROM

A.C. Characteristics

T_A - 0°C to 70°C, V_{CC}[1] = +5V ±5%, V_{PP}[2] = V_{CC} +0.6V[3]

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.[4]	Max.		
t_{ACC1}	Address to Output Delay		250	450	ns	PD/PGM = $\overline{CS}$ = V_{IL}
t_{ACC2}	PD/PGM to Output Delay		250	450	ns	$\overline{CS}$ = V_{IL}
t_{CO}	Chip Select to Output Delay			120	ns	PD/PGM = V_{IL}
t_{PF}	PD/PGM to Output Float	0		100	ns	$\overline{CS}$ = V_{IL}
t_{DF}	Chip Deselect to Output Float	0		100	ns	PD/PGM = V_{IL}
t_{OH}	Address to Output Hold	0			ns	PD/PGM = $\overline{CS}$ = V_{IL}

Capacitance[5] T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

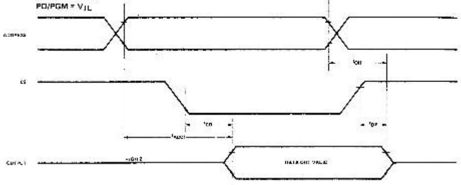
A.C. Test Conditions:

Output Load: 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times: ≤20 ns
Input Pulse Levels: 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs 1V and 2V
Outputs 0.8V and 2V

WAVEFORMS

A. Read Mode

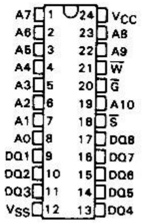
PD/PGM = V_{IL}



- The basic speed of this EPROM is 450 ns.
  - 8086/8088 operates with a 5 MHz clock
- This type of component requires wait states to operate properly with 8086/8088 because of its rather long access time.

4016: 2K × 8 read/write SRAM

TMS4016 . . . NL PACKAGE
(TOP VIEW)



On the slowest 4016, access time $t_{a(A)}$ is 250 ns, fast enough to connect directly to an 8088/8086 at 5 MHz

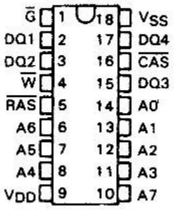
switching characteristics over recommended voltage range, T_A = 0°C to 70°C

PARAMETER		TMS4016-12		TMS4016-15		TMS4016-20		TMS4016-25		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
$t_{a(A)}$	Access time from address		120		150		200		250	ns
$t_{a(S)}$	Access time from chip select low		60		75		100		120	ns
$t_{a(G)}$	Access time from output enable low		50		60		80		100	ns
$t_{v(A)}$	Output data valid after address change	10		15		15		15		ns
$t_{dis(S)}$	Output disable time after chip select high		40		50		60		80	ns
$t_{dis(G)}$	Output disable time after output enable high		40		50		60		80	ns
$t_{dis(W)}$	Output disable time after write enable low		50		60		60		80	ns
$t_{en(S)}$	Output enable time after chip select low	5		5		10		10		ns
$t_{en(G)}$	Output enable time after output enable low	5		5		10		10		ns
$t_{en(W)}$	Outoul enable time after write enable high	5		5		10		10		ns

NOTES: 3. C_L = 100pF for all measurements except $t_{dis(W)}$ and $t_{en(W)}$.
C_L = 5 pF for $t_{dis(W)}$ and $t_{en(W)}$.
4. t_{dis} and t_{en} parameters are sampled and not 100% tested.

TMS4464: 64K × 4 dynamic RAM (DRAM)

TMS4464 . . . JL OR NL PACKAGE
(TOP VIEW)



(a)

PIN NOMENCLATURE	
A0-A7	Address Inputs
CAS	Column Address Strobe
DQ1-DQ4	Data-In/Data-Out
Ḡ	Output Enable
RAS	Row Address Strobe
V_{DD}	+ 5-V Supply
V_{SS}	Ground
W̄	Write Enable

(b)

- After 2 or 4 ms, the contents of the DRAM must be completely rewritten (*refreshed*).
- DRAM requires so many address pins that manufacturers multiplexed address inputs.

DRAM Addressing

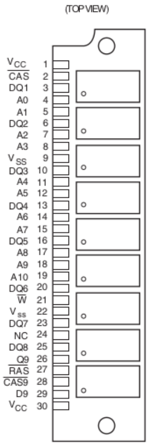
- First, A_0–A_7 are placed on the address pins and strobed into an internal row latch by **row address strobe** ($\overline{RAS}$) as the row address.
- Next, address bits A_8-A_{15} are placed on the same eight address inputs and strobed into an internal column latch by the **column address strobe** ($\overline{CAS}$) as the column address

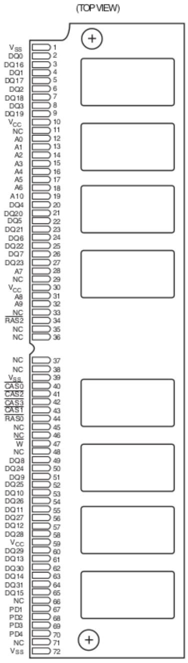
Single In-Line Memory Modules

DRAM is often placed on small boards called SIMMs

The 30-pin SIMM is organized most often as 1M × 8 or 1M × 9, and 4M × 8 or 4M × 9.

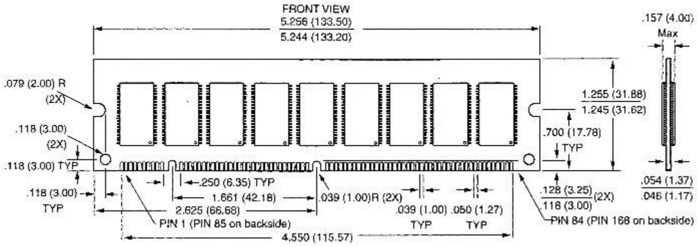
72-pin SIMMs are often organized as 1M × 32 or 1M × 36 (with parity).





Dual In-line Memory Modules

- Pentium 4 microprocessors have a 64-bit wide data bus
- The memory on DIMMs is organized as 64 bits wide.
- Available in DRAM, EDO, SDRAM, and DDR (double-data rate) forms
- An EPROM provides information to the system on the size and the speed of the memory device for plug-and-play applications.



Reflection

- How does the 8086 CPU execute an instruction like **MOV BX, [1000h]**?
- What is the difference between executing **MOV BX, [1000h]** and executing **MOV [1001h], BX** ?
- What is the difference between accessing memory and accessing I/O devices?

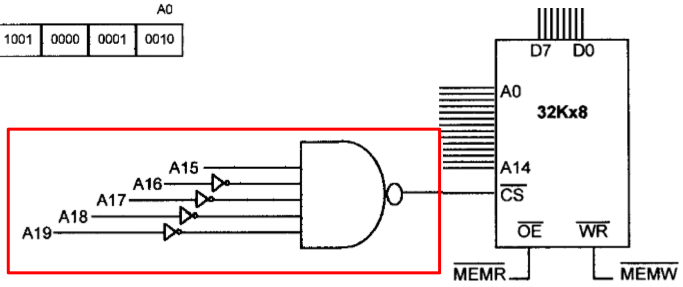
Memory Address Decoding

- According to **your instructions** that access memory,
  - E.g., `MOV AX, [0012H]`
- **CPU** calculates the physical address and put corresponding signals on the address bus
  - E.g., if DS=0900H, *what's the PA?*
- **Memory address decoding circuitry** locates the specific memory chip that stores the desired data
  - Examine address decoding using logic gates and 74LS138 decoder chips

Memory Address Decoding

PA of 0900:0012 = 09012H

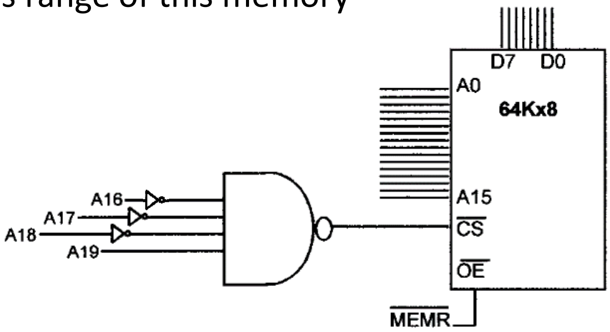
A19				A0
0000	1001	0000	0001	0010



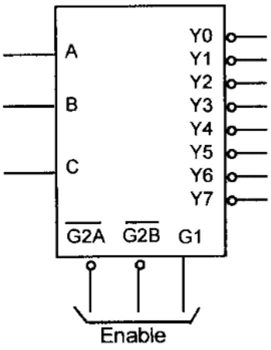
A19				A0	
0000	1000	0000	0000	0000	=08000H
0000	1111	1111	1111	1111	= 0FFFFH

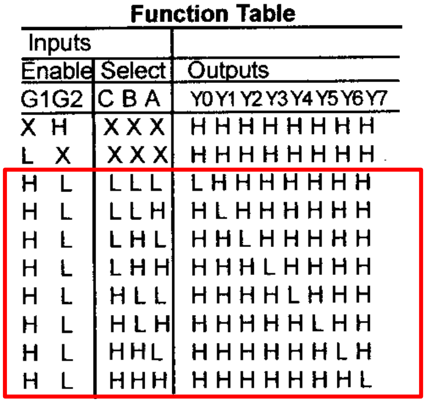
Question

What’s the type and the address range of this memory chip?



The 74LS138 Decoder Chip



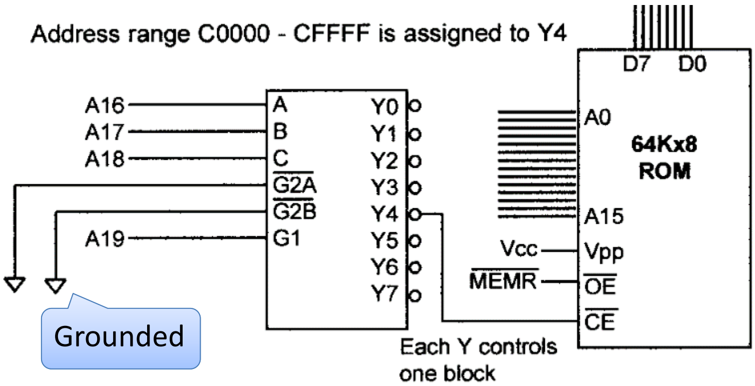


Function Table

Inputs												
Enable		Select			Outputs							
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

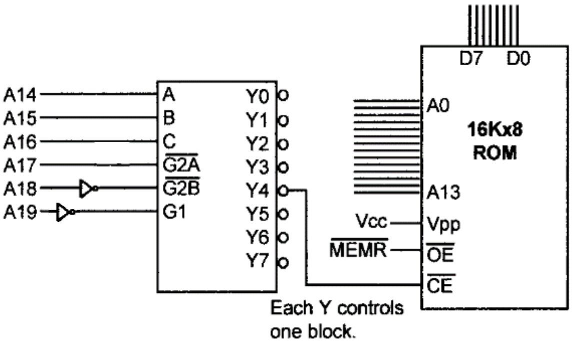
Using 74LS138 to Decode

Address range C0000 - CFFFF is assigned to Y4



Question 2

What's the address range for Y4 and Y7?



More on Address Decoding

- Absolute address decoding
  - All address lines are decoded
- Linear select decoding
  - Only selected lines are decoded
  - Cheap
  - But with aliases*:* the same memory unit (I/O port) with multiple addresses
    - *Why this happens?*

Example

In a particular computer,

- the address range from **0000h** to **3FFFh** is used for ROM,
- the range from **4000h** to **5FFFh** is reserved for future use,
- and the range from **6000h** to **0FFFFh** is used for RAM.

Assume that the control signals for RAM are CS~ and WE~, and the CPU has 16 address pins (i.e., A15~A0), 8 data pins (i.e., D7~D0), and R/W~ and MREQ~ control signals. Achieve the following requests:

1. draw the address decoding solution using a 74LS138 chip
2. if both ROM and RAM are built with 8K×1 memory chips, try to draw the connection between the CPU and the memory.
3. if ROM is built with 8K×8 memory chips and RAM is built with 4K×8 chips, try to draw the connection between the CPU and the memory.
4. what if ROM is built with 16K×8 memory chips and RAM is built with 8K×8 memory chips?

(1) draw the address decoding solution using a 74LS138 chip

Solution.

The logic expression of each output of the 74LS138 chip:

romsel0 = $\overline{A15} * \overline{A14} * \overline{A13} * \overline{MREQ\#}$

romsel1 = $\overline{A15} * \overline{A14} * A13 * \overline{MREQ\#}$

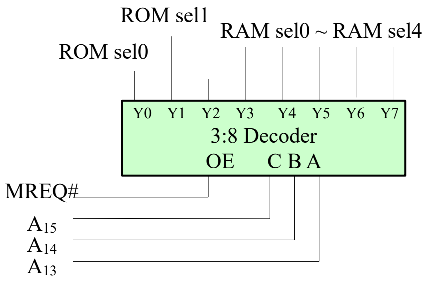
ramsel0 = $\overline{A15} * A14 * A13 * \overline{MREQ\#}$

ramsel1 = $A15 * \overline{A14} * \overline{A13} * \overline{MREQ\#}$

ramsel2 = $A15 * \overline{A14} * A13 * \overline{MREQ\#}$

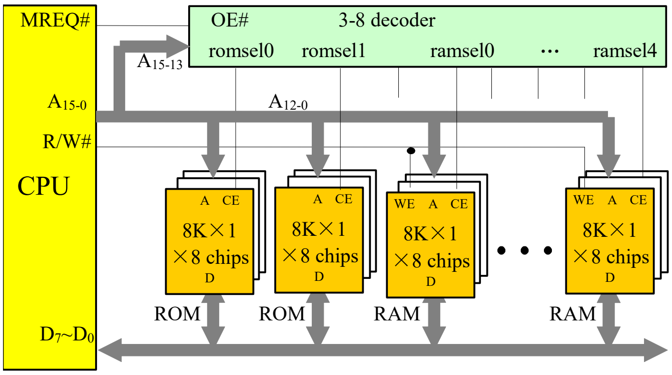
ramsel3 = $A15 * A14 * \overline{A13} * \overline{MREQ\#}$

ramsel4 = $A15 * A14 * A13 * \overline{MREQ\#}$



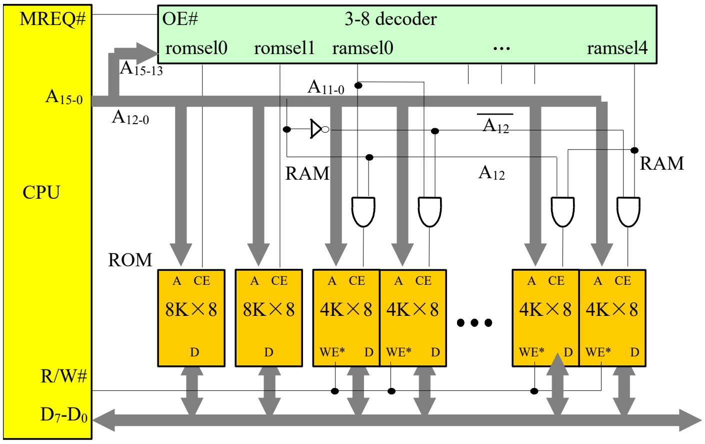
(2) if both ROM and RAM are built with 8K×1 memory chips, try to draw the connection between the CPU and the memory.

Solution.



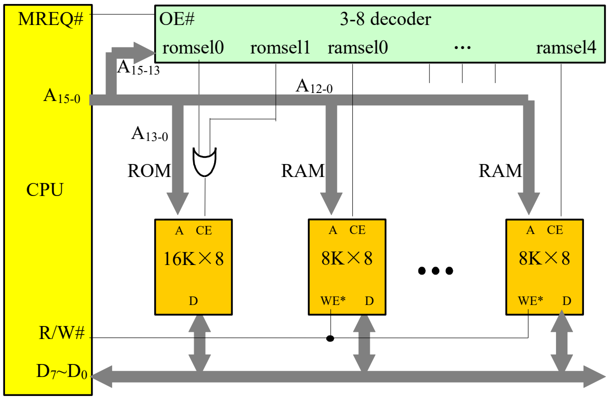
(3) if ROM is built with 8K×8 memory chips and RAM is built with 4K×8 chips, try to draw the connection between the CPU and the memory.

Solution.



(4) what if ROM is built with 16K×8 memory chips and RAM is built with 8K×8 memory chips?

Solution.



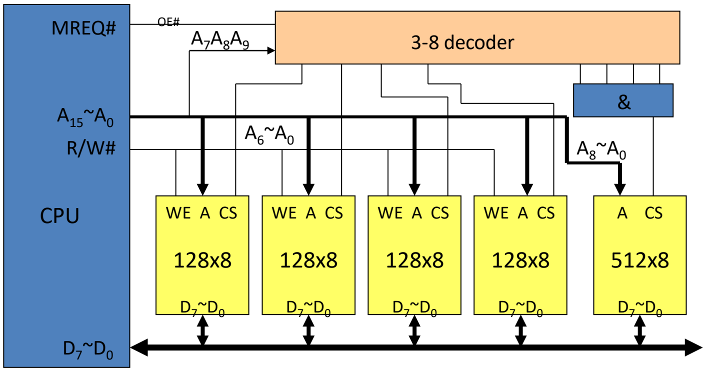
Example 2. Assume one computer system needs 512 byte RAM and 512 byte ROM. If RAM is built with 128×8 memory chips and ROM is built with 512×8 memory chips, please specify the address range of each memory chip. Given that RAM chips need CS~ and WE~ control signals, ROM chips need only CS~ control signal, and the CPU has 16 address pins (A15~A0), 8 data pins (D7~D0) and R/W~ and MREQ~ control signals, draw the connection between the CPU and the memory.

Solution. The address range of each memory chip：

Memory Chip	Address range (hex)	binary
RAM1	0000~007F	0 0 0 x x x x x x x
RAM2	0080~00FF	0 0 1 x x x x x x x
RAM3	0100~017F	0 1 0 x x x x x x x
RAM4	0180~01FF	0 1 1 x x x x x x x
ROM	0200~03FF	1 x x x x x x x x x

Example 2. Assume one computer system needs 512 byte RAM and 512 byte ROM. If RAM is built with 128×8 memory chips and ROM is built with 512×8 memory chips, please specify the address range of each memory chip. Given that RAM chips need CS~ and WE~ control signals, ROM chips need only CS~ control signal, and the CPU has 16 address pins (A15~A0), 8 data pins (D7~D0) and R/W~ and MREQ~ control signals, draw the connection between the CPU and the memory.

As the total volume of the memory is 1K, we need 10 address lines. RAM chips need 7 address lines and ROM chips need 9 address lines.



Data Integrity

Checksum byte for ROM

Check the integrity of a series of bytes

- Calculation
  - Add all bytes together and drop all carries
  - Take the 2's complement of the sum
- Store the checksum byte together with data
- check the integrity by adding data and the checksum together
- *Then how to prove the integrity of the data?*

E.g., 38H, 23H, 33H, 07H, what is the checksum byte? 6BH

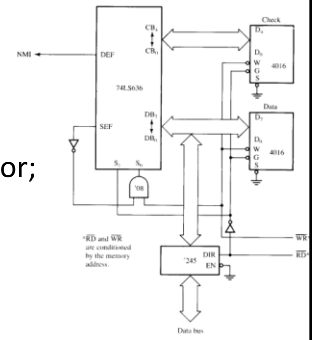
Parity bit for DRAM

Check the integrity of a series of bits (a byte)

- even parity: if the number of 1s in the series of bits is odd, then the parity bit is set to 1; otherwise, set to 0, making the total number of 1s even (Data + the parity bit)
- odd parity: if the number of 1s in the series of bits is odd, then the parity bit is set to 0; otherwise, set to 1
- In 8086, odd parity is used, i.e., if data have even number of 1s, the parity bit is set

CRC for disks and the Internet

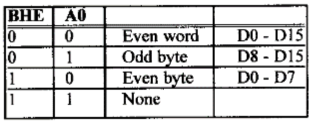
Error-Correcting Circuits (ECC)



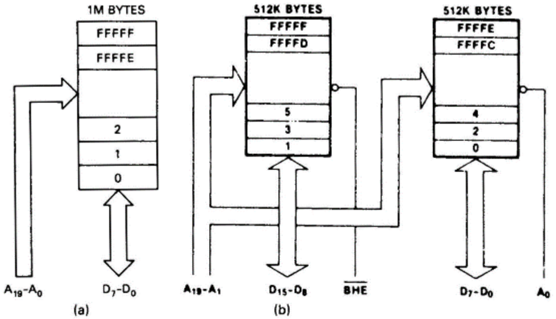
- 74LS636 is an 8-bit error correction and detection circuit that
  - corrects any single-bit memory read error;
  - flags any 2-bit error.
  - Called SECDED (**single error correction /double error detection**).
  - Stores five parity bits
- Modern DDR error-correction memory (ECC) does not actually have logic circuitry on board that detects and corrects errors.
- On 64-bit Pentiums, ECC memory is 72-bits wide using the additional 8 bits to store the ECC code.

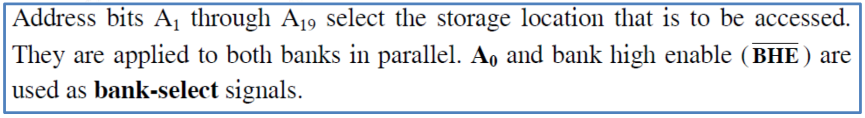
Memory Organization in 8086

- Even and odd banks



BHE	A0		
0	0	Even word	D0 - D15
0	1	Odd byte	D8 - D15
1	0	Even byte	D0 - D7
1	1	None	

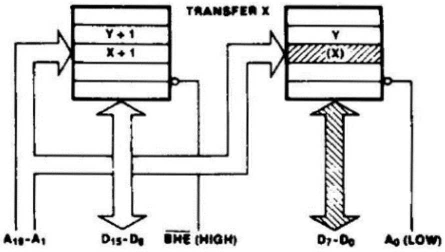




Address bits A_1 through A_{19} select the storage location that is to be accessed. They are applied to both banks in parallel. $\mathbf{A_0}$ and bank high enable ($\overline{\mathbf{BHE}}$) are used as **bank-select** signals.

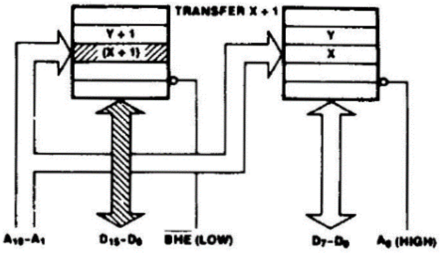
Byte-Memory Operations

- Byte-memory operation at even address X



```
MOV AL, [100h]
```

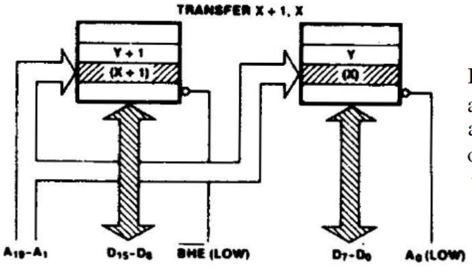
- Byte-memory operation at odd address X+1



```
MOV AL, [101h]
```

Aligned Word-Memory Operations

- Accessing an aligned word at even address X



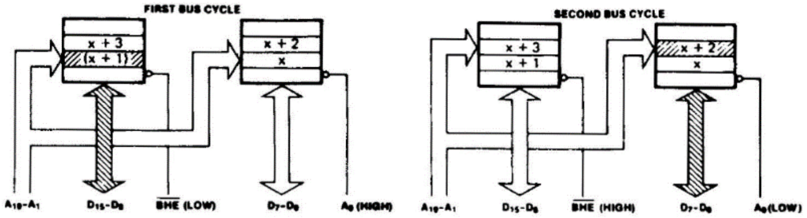
Both the high and low **banks** are accessed at the **same time**. Both $\mathbf{A_0}$ and $\overline{\mathbf{BHE}}$ are set to 0. This 16-bit word is transferred over the complete data bus $\mathbf{D_0}$ through $\mathbf{D_{15}}$ in just **one bus cycle**.

```
MOV AX, [100h]
```

Misaligned Word-Memory Operations

- Accessing an misaligned word at odd address X+1

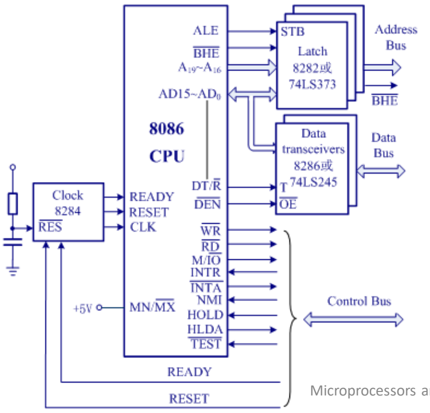
```
MOV AX, [101h]
```

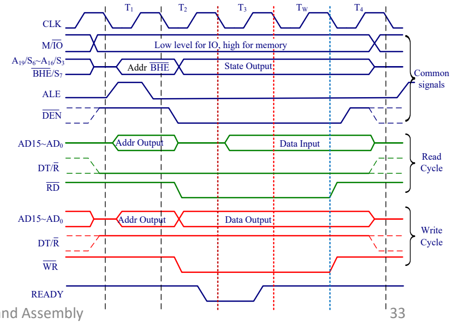


Two bus cycles are needed. During the first bus cycle, the byte of the word located at address X + 1 in the high bank is accessed over $\mathbf{D_8}$ through $\mathbf{D_{15}}$. Even though the data transfer uses data lines $\mathbf{D_8}$ through $\mathbf{D_{15}}$, to the processor it is the **low byte** of the addressed data word. In the **second memory bus cycle**, the even byte located at X + 2 in the low bank is accessed over bus lines $\mathbf{D_0}$ through $\mathbf{D_7}$.

Review

What is the complete procedure of the 8086 CPU executing an instruction like **MOV BX, [1000h]**? What about **MOV [1001h], BX**?


ALE
BHE
A19~A16
AD15~AD0
8086
CPU
DT/R
DEN
STB
Latch
8282或
74LS373
Address
Bus
BHE
Data
transceivers
8286或
74LS245
T
OE
Data
Bus
Clock
8284
RES
READY
RESET
CLK
WR
RD
M/IO
INTR
INTA
NMI
HOLD
HLDA
TEST
+5V
MN/MX
Control Bus
READY
RESET


T1
T2
T3
TW
T4
CLK
M/IO
Low level for IO, high for memory
A19/S6~A16/S3
BHE/S7
Addr BHE
State Output
ALE
DEN
Common signals
AD15~AD0
Addr Output
Data Input
DT/R
RD
Read Cycle
AD15~AD0
Addr Output
Data Output
DT/R
WR
Write Cycle
READY