You're sharing your screen

Mansoura University
Faculty of Computers and Information
Department of Computer Science

[CS214P] Assembly Language: An Introduction

Grade: Third Year (Computer Science)

Sara El-Metwally, Ph.D.

Faculty of Computers and Information,

Mansoura University,

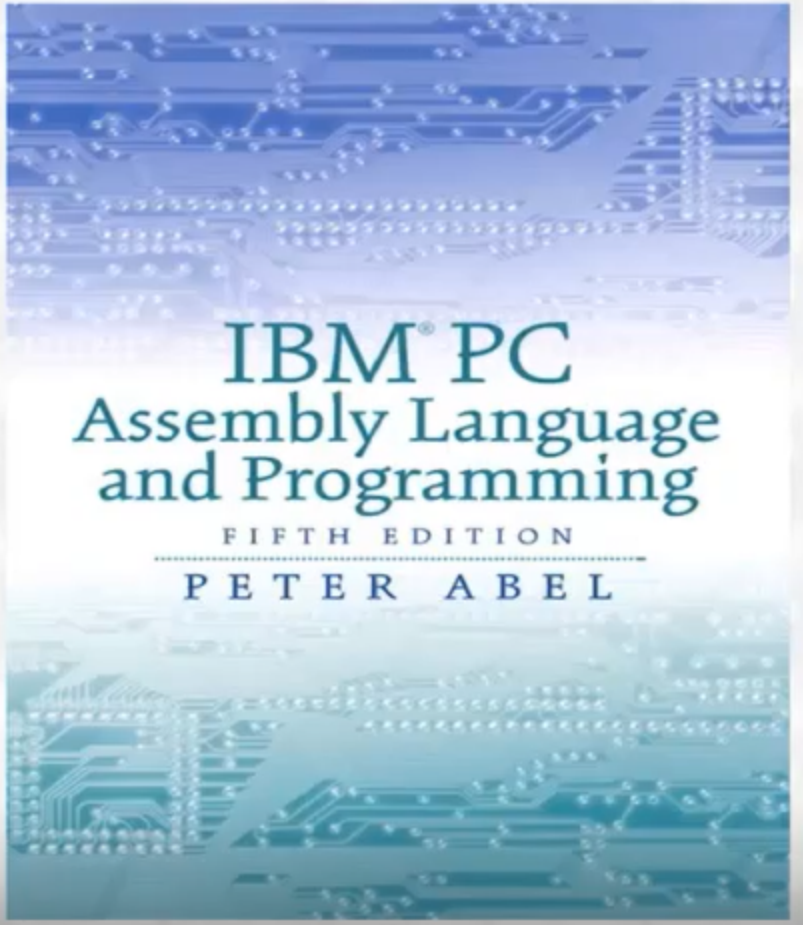
Egypt.

Assembly-Language-Introduction-Lecture 1- Part 1(Fall 2021)
021

You're sharing your screen

Course outlines

- **Course Text Book:**



You're sharing your screen

Course outlines

- **Course Labs " materials ":**
  - ➢ Assignments and solving some exercises.
  - ➢ DosBox.
  - ➢ Debug Program.
  - ➢ Emulator 8086.
  - ➢ AE-TASM.
- **Course TA's:**
  - ➢ Eng. Ghada Shafiq

You're sharing your screen

Course Objectives

- Understand the HW of the personal computer.
- Understand machine-language code and hexa-decimal format.
- Understand the steps involved in assembling, linking, and executing a program.
- Write programs in assembly language to handle the keyboard and screen, perform arithmetic, convert between ASCII and binary formats, perform table searches and sorts, and handle disk I/O.

You're sharing your screen

Course Objectives

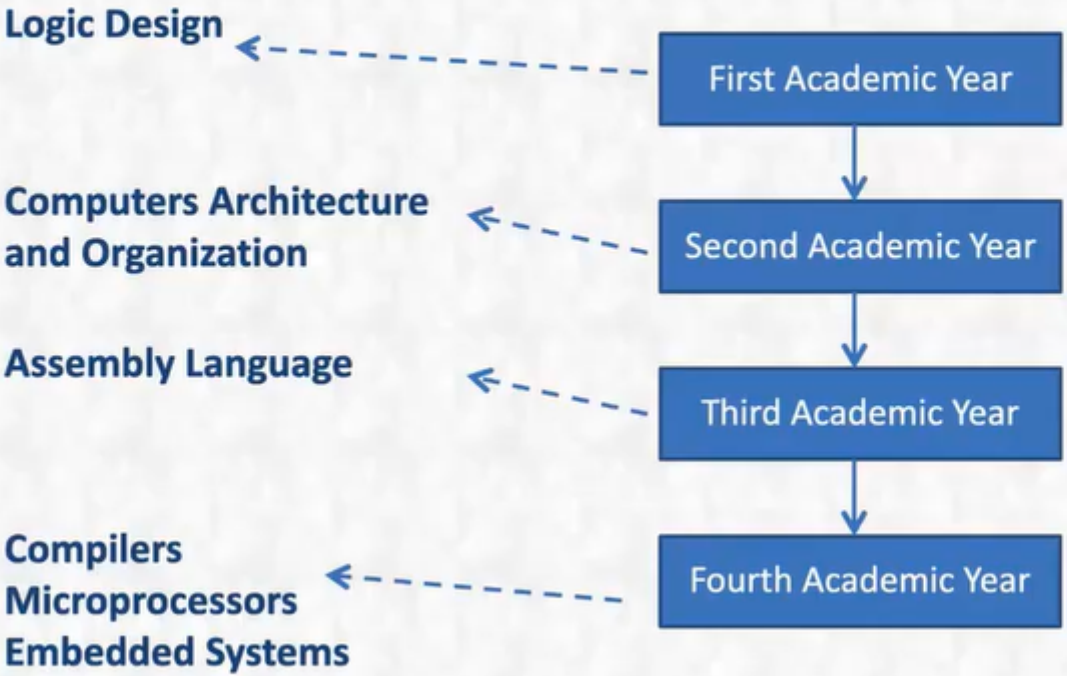
- Trace machine execution as an aid in program debugging.
- Write your own macro instructions to facilitate faster coding.
- Linking separately assembled programs into one executable program.

You're sharing your screen

Course Requirements

You're sharing your screen

Course Requirements



You're sharing your screen

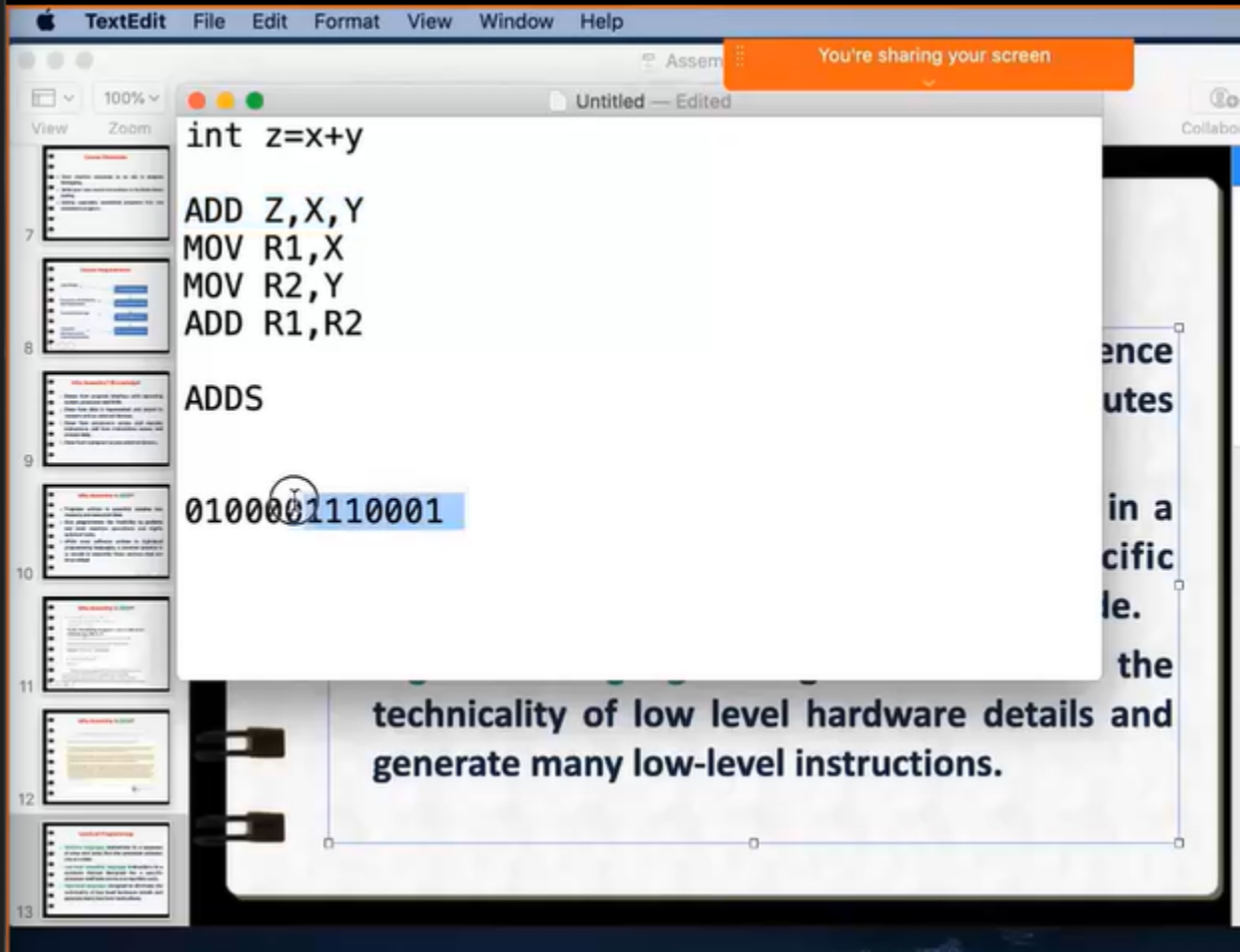
Why Assembly? (Knowledge)

- Shows how program interface with operating system, processor and BIOS.
- Show how data is represented and stored in memory and on external devices.
- Show how processors access and execute instructions and how instructions access and process data.
- Show how a program access external devices.

You're sharing your screen

Levels of Programming

- **Machine language**: instructions in a sequence of ones and zeros that the processor executes one at a time.
- **Low-level assembly language**: instructions in a symbolic format designed for a specific processor and have one to one machine code.
- **High-level language**: designed to eliminate the technicality of low level hardware details and generate many low-level instructions.


TextEdit File Edit Format View Window Help
You're sharing your screen
Untitled — Edited
int z=x+y
ADD Z,X,Y
MOV R1,X
MOV R2,Y
ADD R1,R2
ADDS
0100001110001
technicality of low level hardware details and
generate many low-level instructions.

You're sharing your screen

Levels of Programming

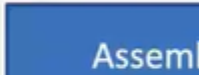
```
unsigned int  bit_seq_i;
unsigned char *bit_seq_buff,
bit_seq_i=16;
```

High-level language

Compiler

```
mov ax, data
mov ds, ax
mov es, ax
mov cx, 10
mov dx, 0
```

Low-level assembly language



Assembler

```
0001100001000010001
0010000010000111100
1001001001100001001
```

Machine language

You're sharing your screen

What is Assembly Language?

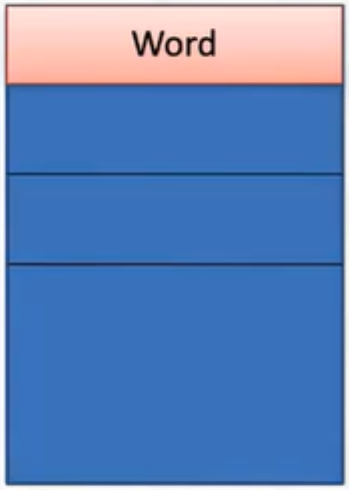
- **Assembly language** is a low-level programming language for a computer or other programmable device specific to a particular computer architecture.
- Each personal computer has a microprocessor that manages the computer's arithmetical, logical, and control activities.
- Each family of processors has its own set of instructions for handling various operations such as getting input from keyboard, displaying information on screen etc.

You're sharing your screen

To Start:

You need to know your Computer ?

0101 0000 0000 500H



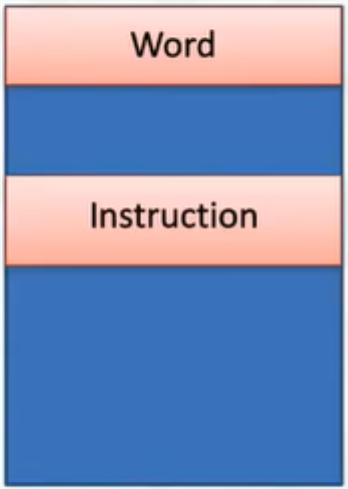
Memory

You're sharing your screen

To Start:

You need to know your Computer ?

0101 0000 0000 500H



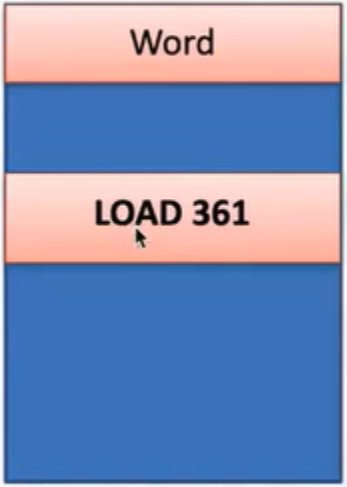
Memory

You're sharing your screen

To Start:

You need to know your Computer ?

0101 0000 0000 500H



Memory

You're sharing your screen

To Start:

You need to know your Computer ?

0101 0000 0000 500H

Memory

TextEdit File Edit Format View Window Help

You're sharing your screen

Untitled — Edited

```
ins = 16 bit
inst= opcode + address
16= ?+12
load 361
load 0110

361H=12bit


load 0110
add 1110
sub 0101
```

Memory

You're sharing your screen

To Start:

You need to know your Computer ?

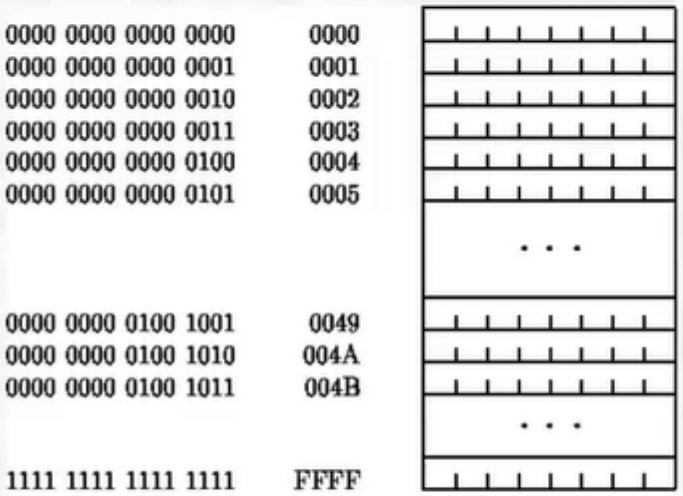
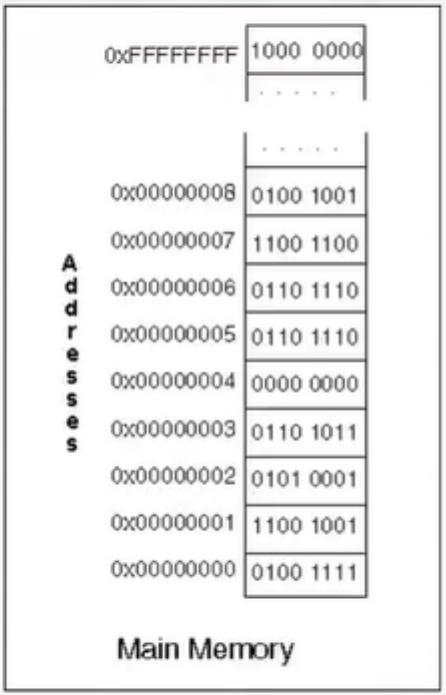


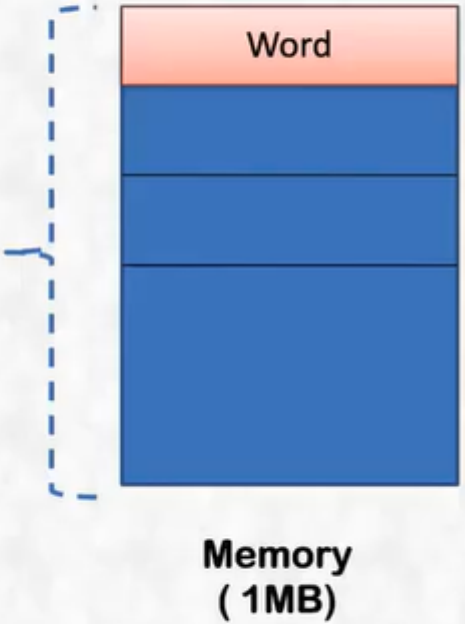
Figure 1.2: Memory and Addresses



You're sharing your screen

To Start:

You need to know your Computer ?

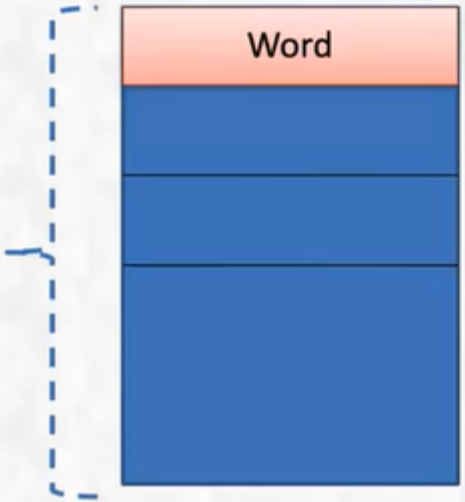


$$memory\ size\ in\ words = \frac{memory\ size\ in\ bits}{word\ size}$$

You're sharing your screen

To Start:

You need to know your Computer ?



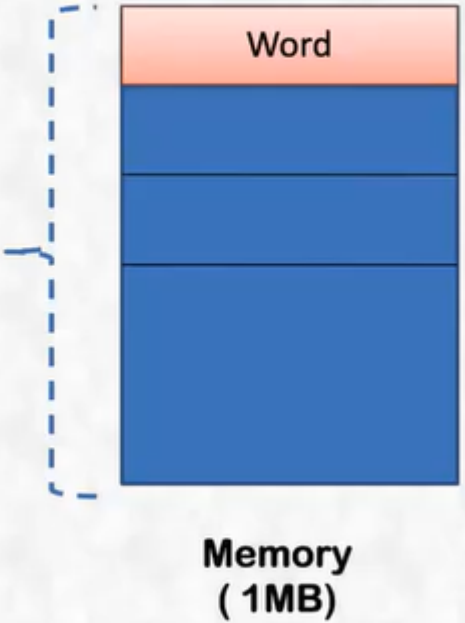
Memory
(1MB)

$$memory\ size\ in\ words = \frac{memory\ size\ in\ bits}{word\ size}$$

$$memory\ size\ in\ words = \frac{2^{10} \times 2^{10} \times 2^{3}}{2^{3}} = 2^{20}\ words$$

You're sharing your screen

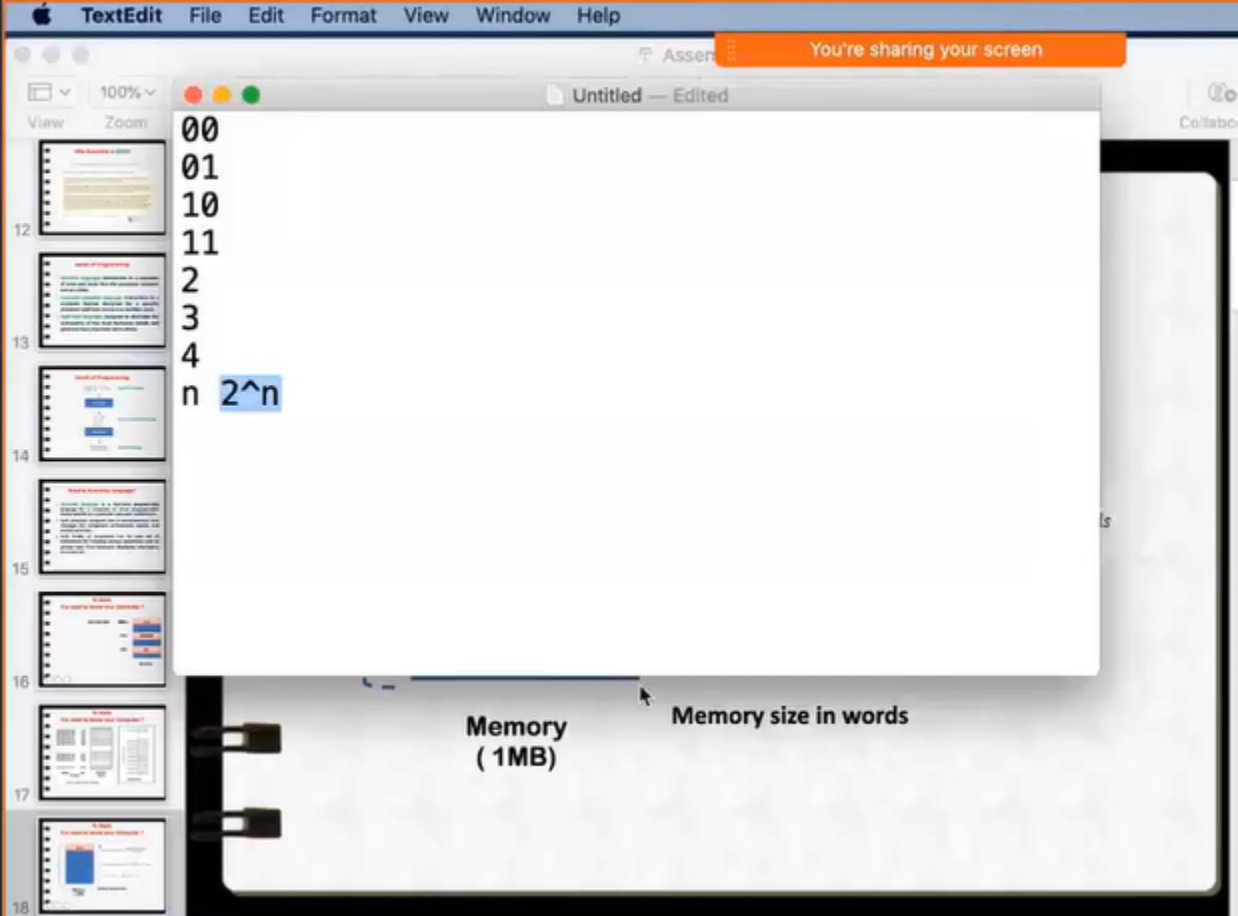
To Start:
You need to know your Computer ?



$$memory\ size\ in\ words = \frac{memory\ size\ in\ bits}{word\ size}$$

$$memory\ size\ in\ words = \frac{2^{10} \times 2^{10} \times 2^{3}}{2^{3}} = 2^{20}\ words$$

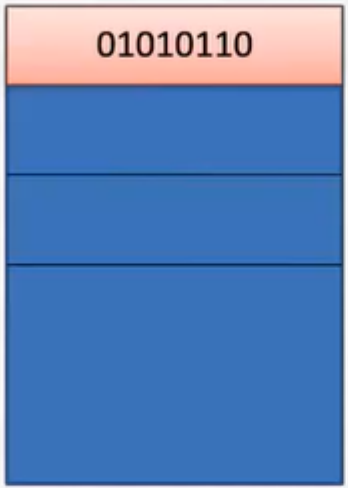
$$\#\ bits\ for\ address = 2^{20}\ words = 20\ bits$$


TextEdit File Edit Format View Window Help
You're sharing your screen
Untitled — Edited
00
01
10
11
2
3
4
n 2^n
Memory
(1MB)
Memory size in words

You're sharing your screen

To Start:

You need to know your Computer ?



Memory
(1MB)

0001 0001 0010 0010 1111

0x1122F

$$memory\ size\ in\ words = \frac{memory\ size\ in\ bits}{word\ size}$$

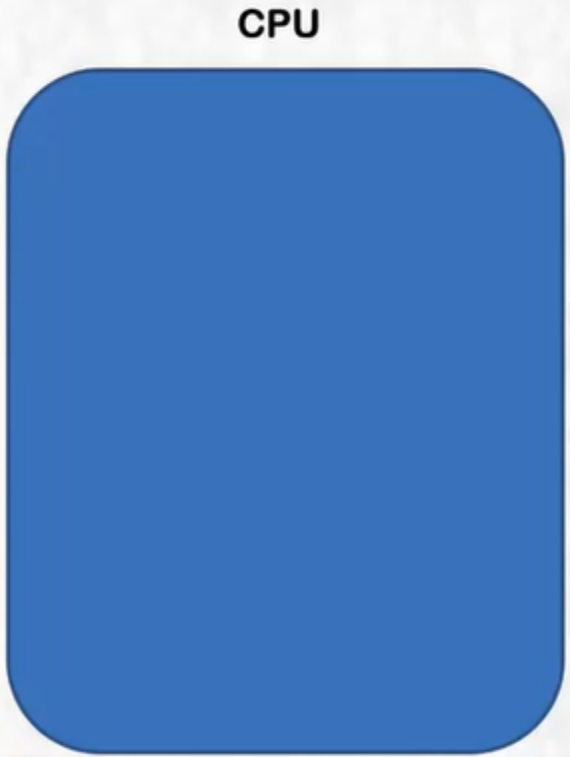
$$memory\ size\ in\ words = \frac{2^{10} \times 2^{10} \times 2^{3}}{2^{3}} = 2^{20}\ words$$

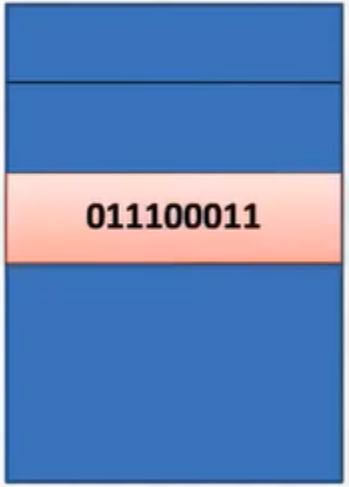
$$\#\ bits\ for\ address = 2^{20}\ words = 20\ bits$$

You're sharing your screen

To Start:

You need to know your Computer ?

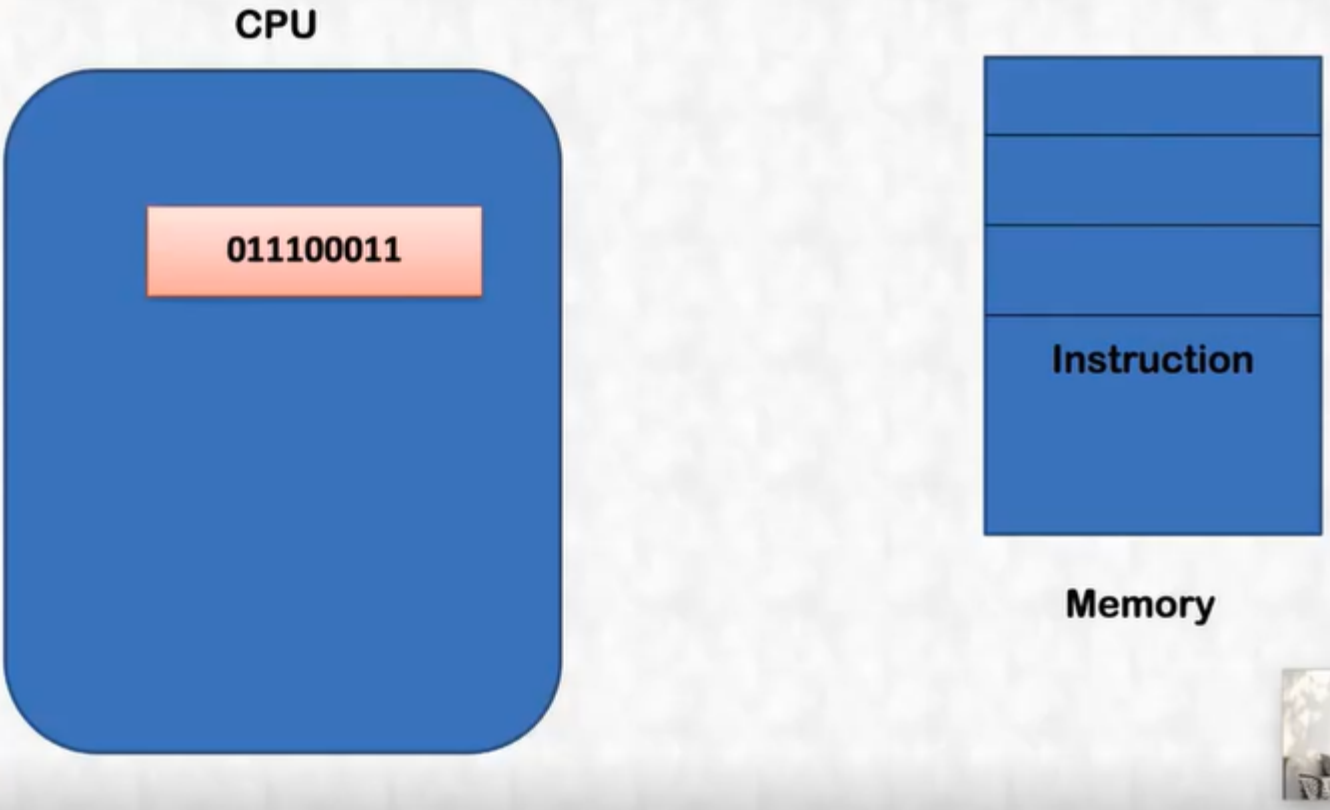




You're sharing your screen

To Start:

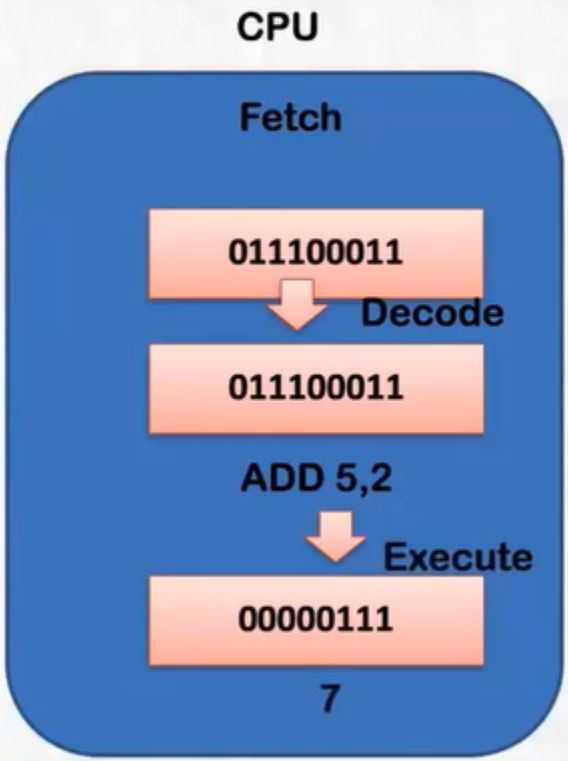
You need to know your Computer ?

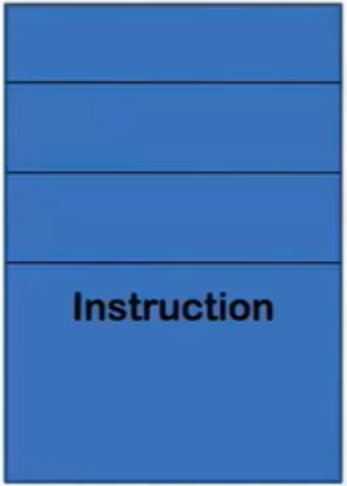


You're sharing your screen

To Start:

You need to know your Computer ?

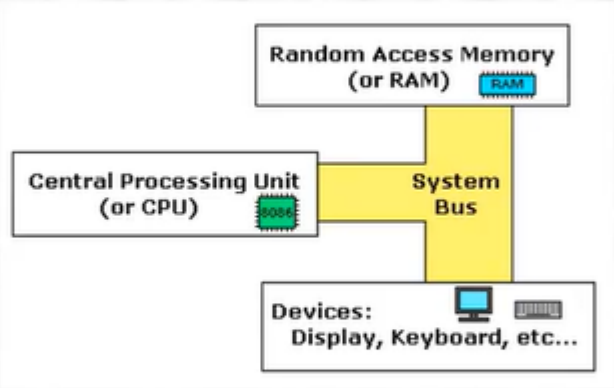




You're sharing your screen

To Start:

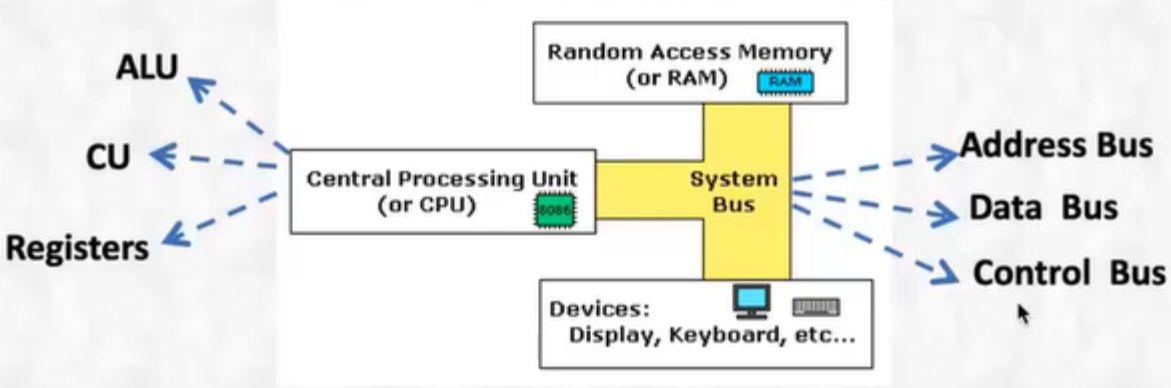
You need to know your Computer ?

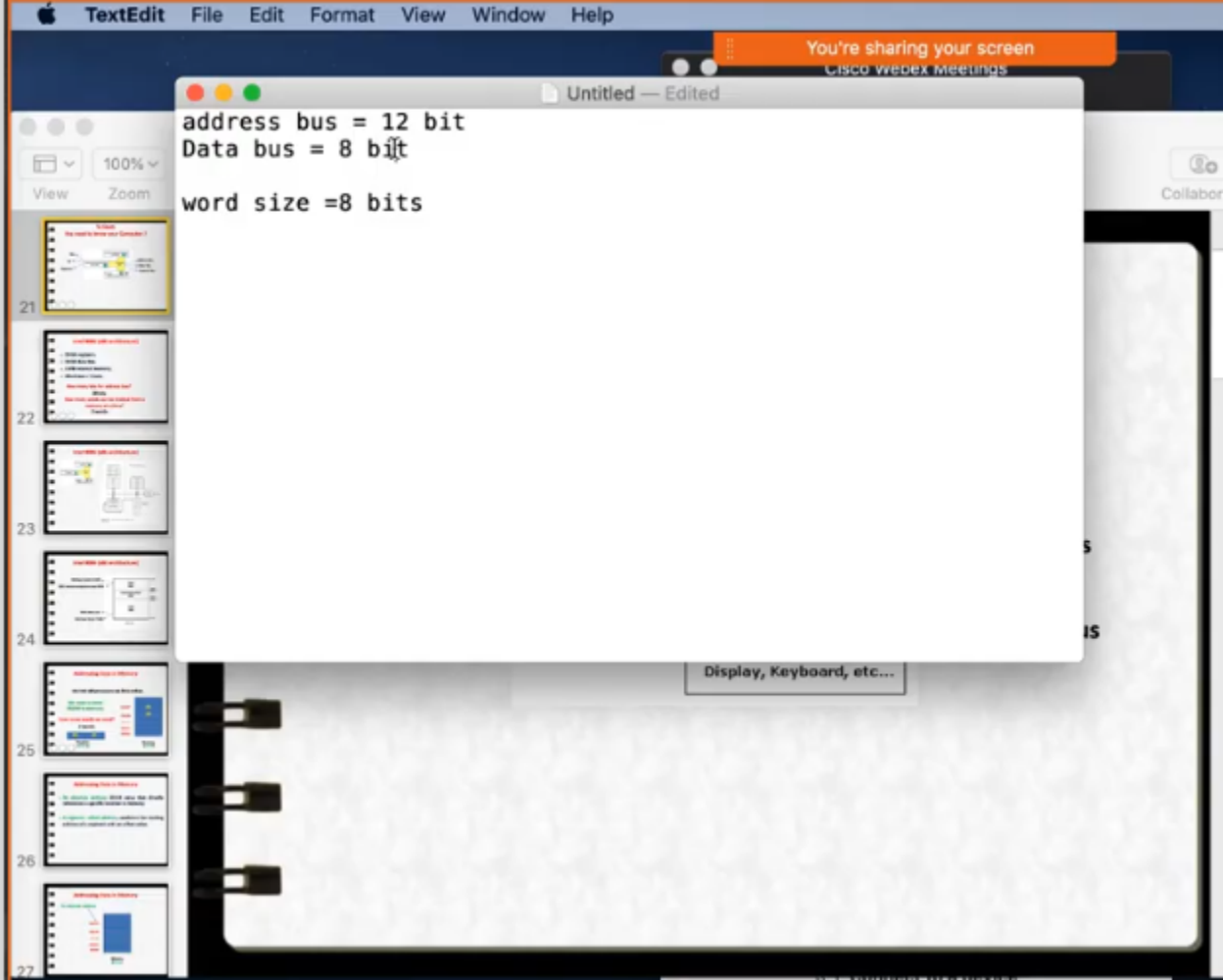


You're sharing your screen

To Start:

You need to know your Computer ?

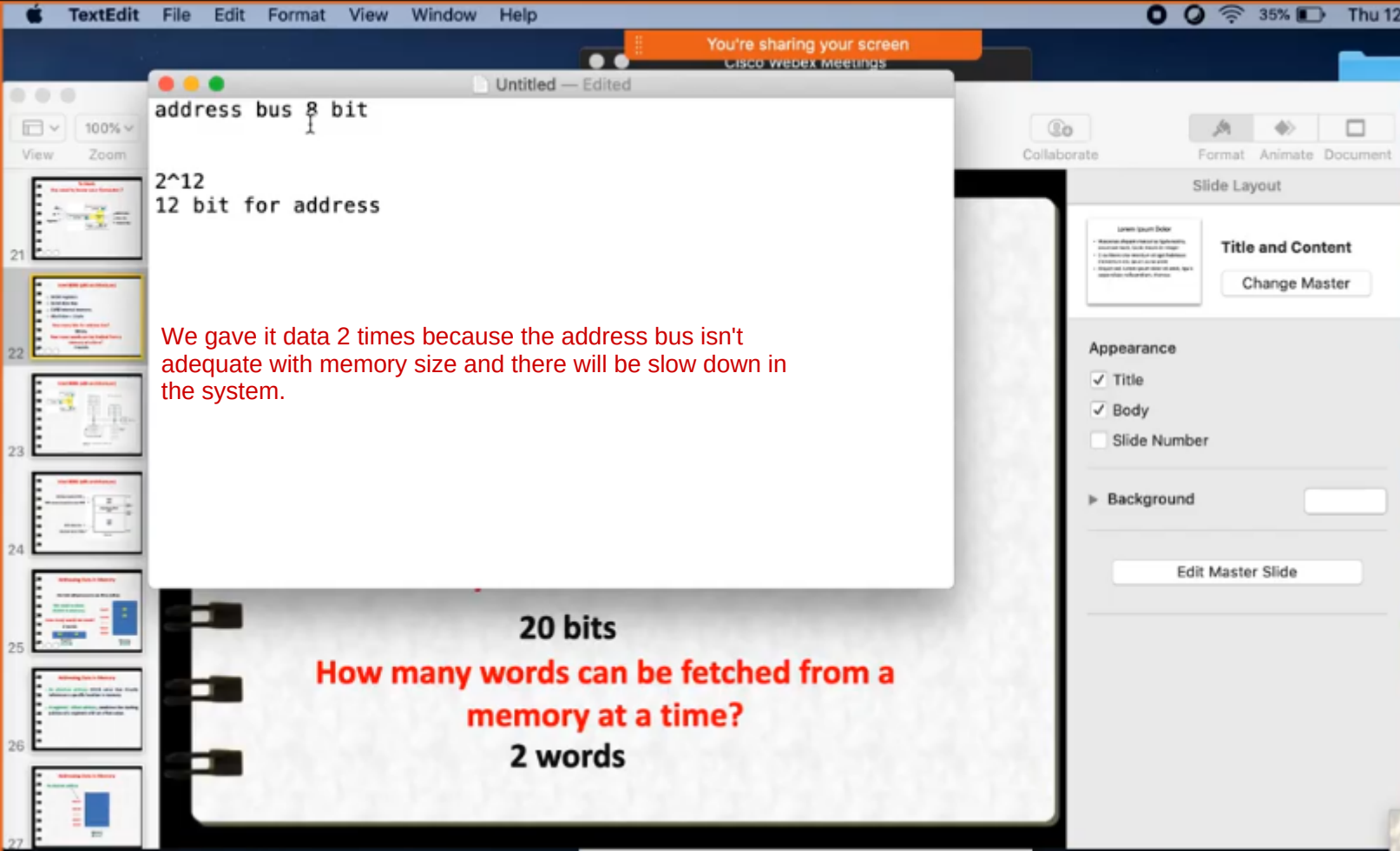



TextEdit File Edit Format View Window Help
You're sharing your screen
Cisco Webex Meetings
Untitled — Edited
address bus = 12 bit
Data bus = 8 bit
word size =8 bits
Display, Keyboard, etc...

You're sharing your screen

Intel 8086 (x86 architecture)

- 16-bit registers.
- 16-bit data bus.
- 1 MB internal memory.
- Word size = 1 byte.


TextEdit File Edit Format View Window Help
35% Thu 12
You're sharing your screen
Cisco Webex Meetings
Untitled — Edited
address bus 8 bit
2^12
12 bit for address
We gave it data 2 times because the address bus isn't adequate with memory size and there will be slow down in the system.
20 bits
How many words can be fetched from a memory at a time?
2 words
View
Zoom
Collaborate
Format
Animate
Document
Slide Layout
Title and Content
Change Master
Appearance
Title
Body
Slide Number
Background
Edit Master Slide

You're sharing your screen

Intel 8086 (x86 architecture)

- 16-bit registers.
- 16-bit data bus.
- 1 MB internal memory.
- Word size = 1 byte.

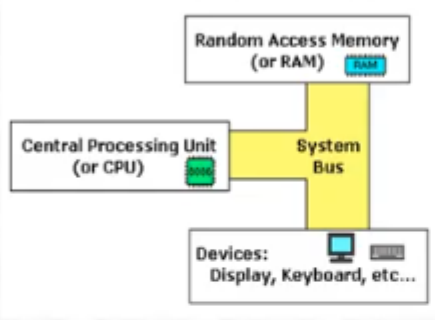
How many bits for address bus?

20 bits

How many words can be fetched from a memory at a time?

You're sharing your screen

Intel 8086 (x86 architecture)



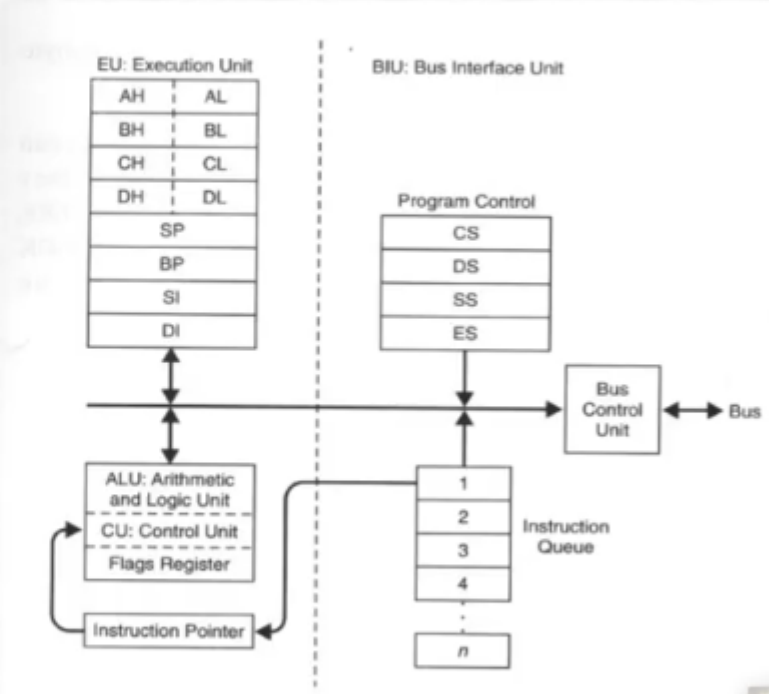
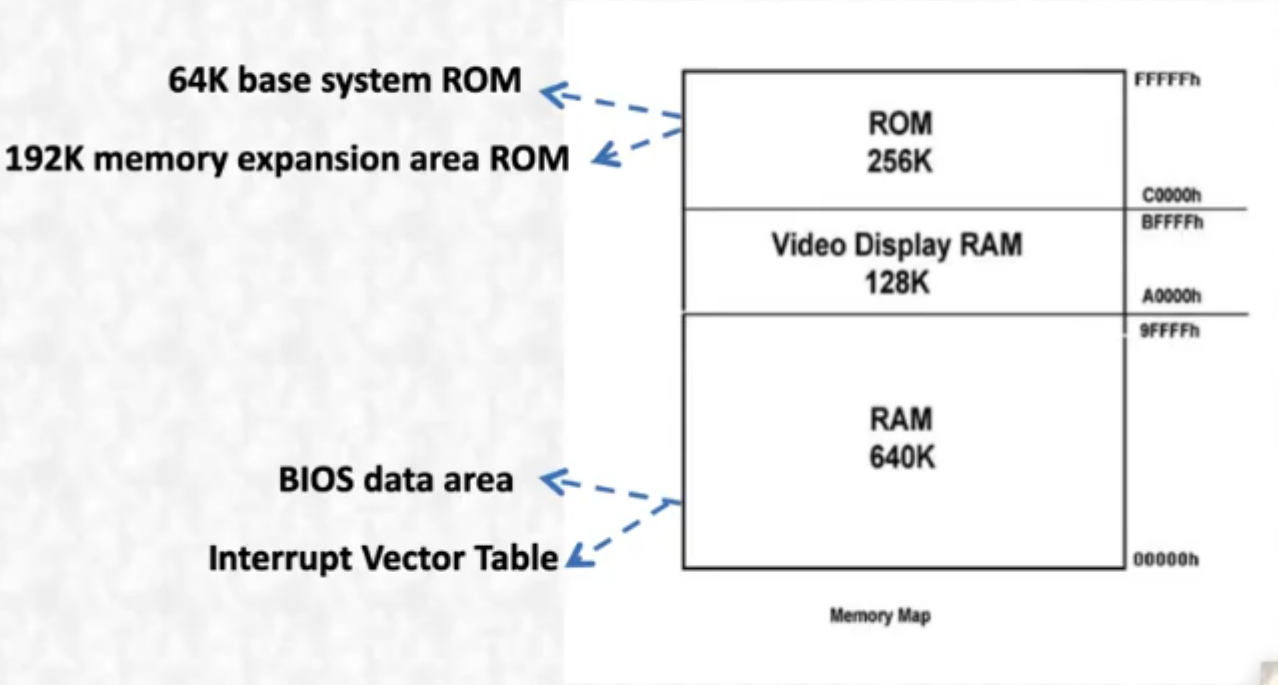


Figure 1-2 Execution Unit and Bus Interface Unit

You're sharing your screen

Intel 8086 (x86 architecture)



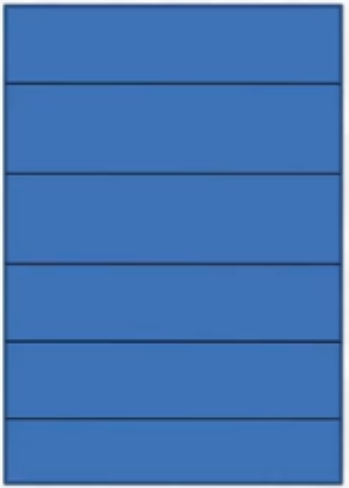
You're sharing your screen

Addressing Data in Memory

the Intel x86 processors use little-endian.

We need to store
0529H in memory

Register
(Queue)



Memory
(Stack)

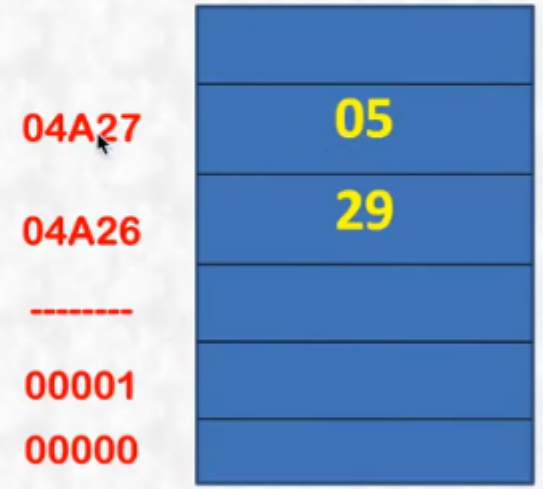
You're sharing your screen

Addressing Data in Memory

the Intel x86 processors use little-endian.

We need to store
0529H in memory

Register
(Queue)



Memory
(Stack)

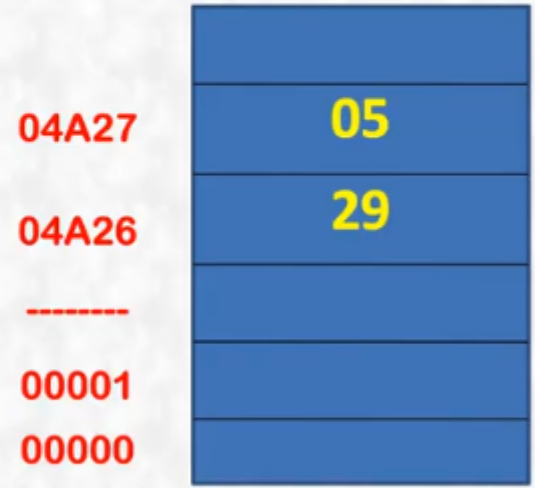
You're sharing your screen

Addressing Data in Memory

the Intel x86 processors use little-endian.

We need to store 0529H in memory

Register
(Queue)



Memory
(Stack)