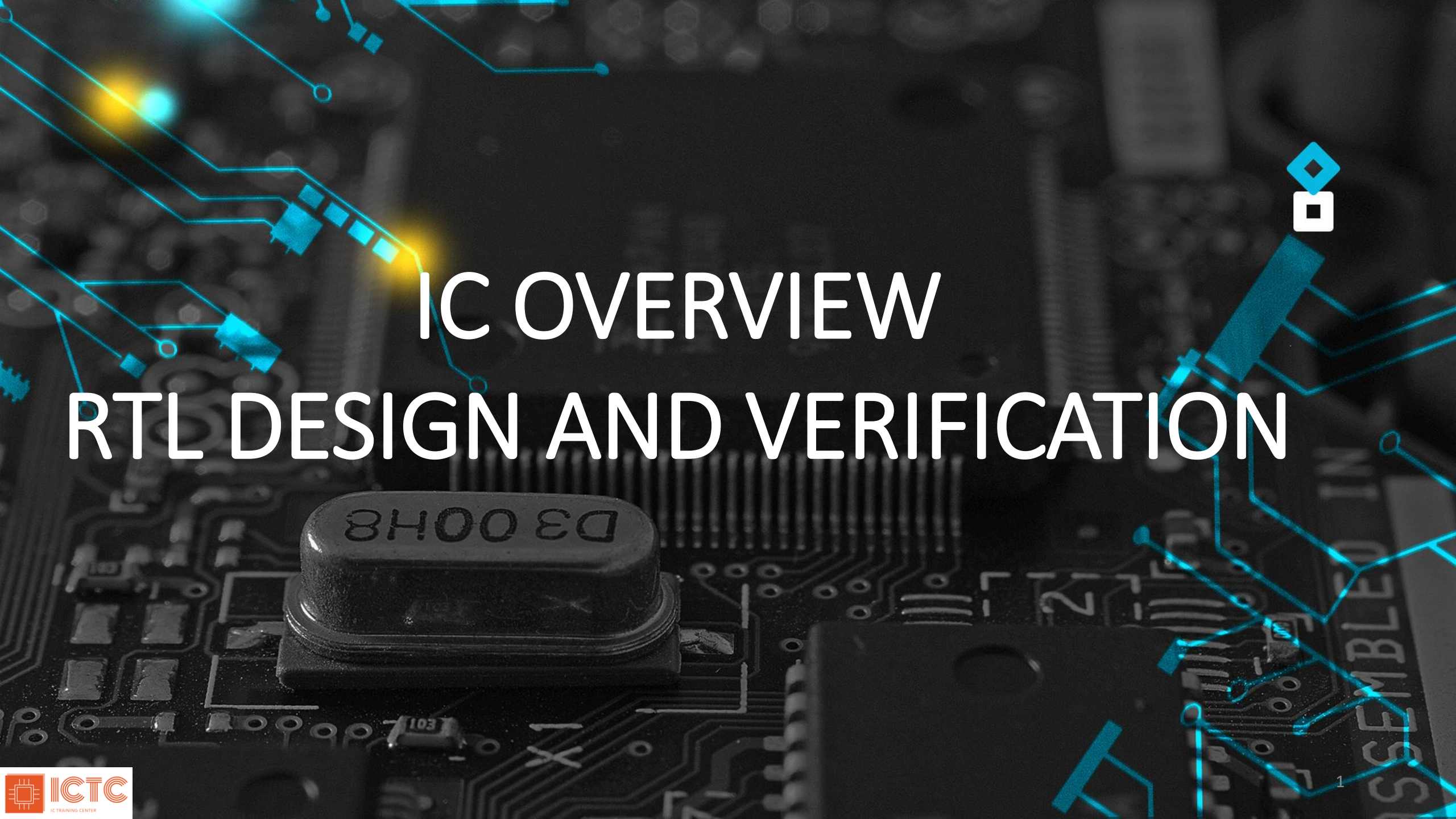
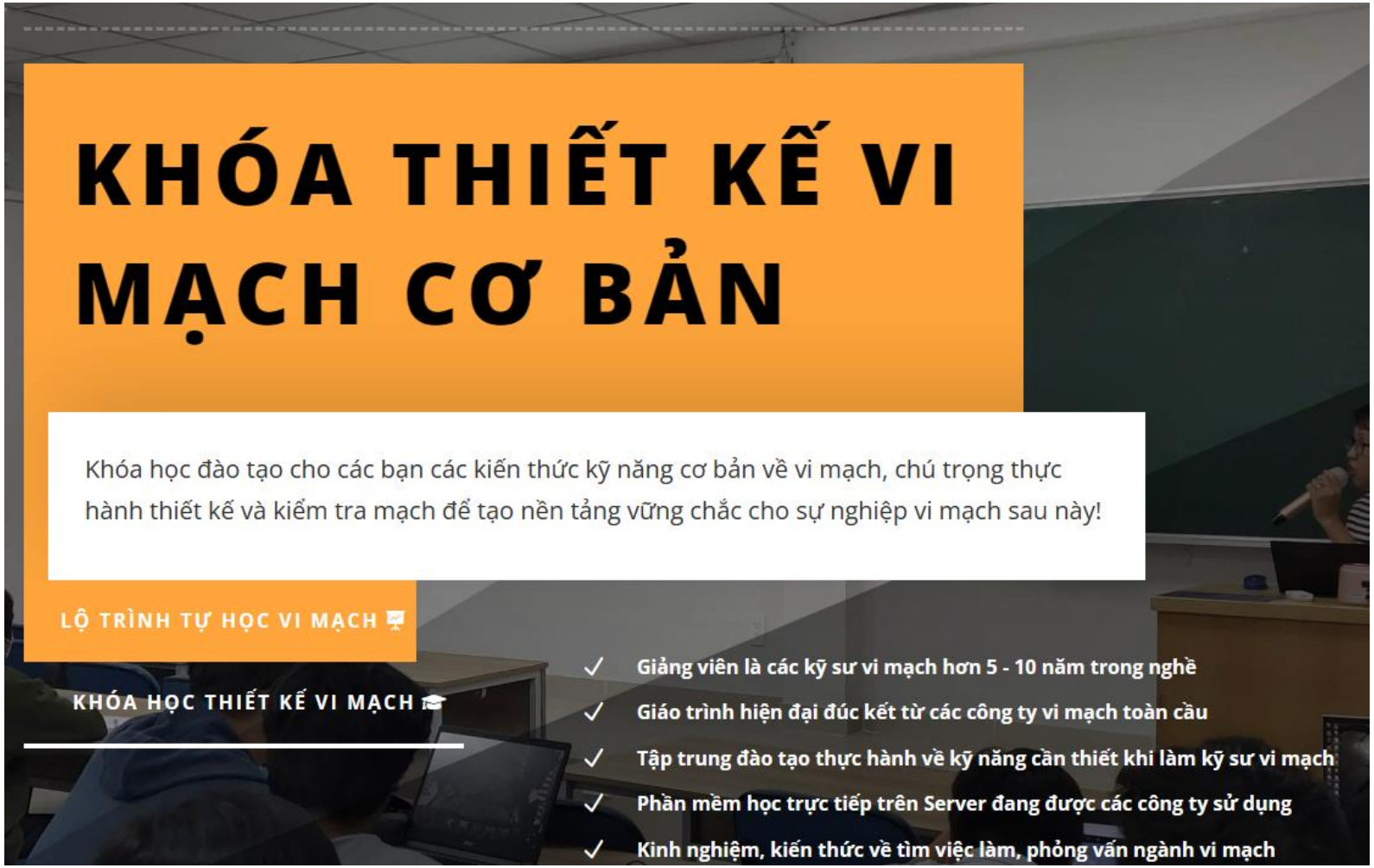


IC OVERVIEW
RTL DESIGN AND VERIFICATION

COURSE INTRODUCTION

[Khóa Học Thiết Kế Vi Mạch Cơ Bản - Trung Tâm Đào Tạo Thiết Kế Vi Mạch ICTC]



COURSE INTRODUCTION

SUMMARY

HOMEWORK

QUESTION

SELF-LEARNING

Session 1: Number Systems and Computer Fundamentals

1. Number System
2. Basic Computer
3. Address space

Number systems

Fundamental elements

There are 2 main elements of a number system:

- **Digit**: represent numerical values in the system. For example, in the decimal system, digits are 0 through 9.
- **Base**: the number of unique digits (including zero) used in that system. It determines the counting or positional value of each digit. Common bases include 10 (decimal), 2 (binary), 8 (octal), and 16 (hexadecimal).

Number systems

Decimal Systems

- Digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Base: 10
- Example of how to analyze a decimal number:

$$1534\,D = 1.10^3 + 5.10^2 + 3.10^1 + 4.10^0$$
$$3570\,D = 3.10^3 + 5.10^2 + 7.10^1 + 0.10^0$$
$$0.315\,D = 3.10^{-1} + 1.10^{-2} + 5.10^{-3}$$

Number systems

Binary Systems

- Binary system is used for computer.
- Use 0 and 1 to represent for low voltage level and high voltage level respectively.
- Each digit is called a “bit” (binary digit).
- A string of 0 and 1 can be used to encode all kind of data so that computer can process.
- Digit: 0, 1
- Base: 2

$10101\ B = 1.2^4 + 0.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = 21\ D$

$11.01\ B = 1.2^1 + 1.2^0 + 0.2^{-1} + 1.2^{-2} = 3.25\ D$

- Value range:

Binary	Decimal
0000	0
0001	1
…	…
0111	7
…	…
1111	15
…	…
11…1 (n-bit)	2^n-1

Number systems

Octal Systems

- Why need to know this system ?
- Using binary system is very long to express a data. Then, we need a number system that can be shorten the expressing and conversing process.
- Digit: 0, 1, 2, 3, 4, 5, 6, 7
- Base: 8

$505\ O = 5.8^2 + 0.8^1 + 5.8^0 = 325\ D$

$123.45\ O = 1.8^2 + 2.8^1 + 3.8^0 + 4.8^{-1} + 5.8^{-2}$

Octal digit	Binary	Decimal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Number systems

Octal Systems

- Convert from binary to octal

1 001 101 011 100 B = 11534 O
1 1 5 3 4

- Convert from octal to binary

567 O = 101 110 111 B = 101110111 B
5 6 7

Octal digit	Binary	Decimal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Number systems

Hexadecimal Systems

- Octal system is still long. Then hexadecimal number system is created.
- Digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Base: 16

$5F\ H = 5.16^1 + 15.16^0 = 95\ D$

$EA0\ H = 14.16^2 + 10.16^1 + 0.16^0 = 3744\ D$

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Number systems

Hexadecimal Systems

- Convert from binary to hex

101 1101 1010 B = 5DA H

5 D A

- Convert from hex to binary

25 H = 0010 0101 B

2 5

F05 H = 1111 0000 0101 B

F 0 5

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Number systems

Hexadecimal Systems

Practice:

- ❑ Convert 57 H to Binary
- ❑ Convert 01 1011 0110 to Hex

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Number systems

Binary addition and substraction

Addition Rule:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0 (carry = 1)
- Example

```
 1     1 1   1   (carry)
   1 0 0 1 1 0 1
+
   1 1 0 1 1 0 1
 ----------------
 1 0 1 1 1 0 1 0
```

Substraction Rule:

- 0 - 0 = 0
- 0 - 1 = 1 (borrow 1)
- 1 - 0 = 1
- 1 - 1 = 0
- Example

```
 1   1 1 1   (borrow)
   1 0 1 0 1 0 1
-
   0 1 0 1 1 1 0
 ----------------
   0 1 0 0 1 1 1
```

Number systems

Binary addition and substraction

Addition rule:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0 (carry = 1)
- Practice:

```
  0 1 0 1 1 1 1
+
  1 1 0 1 1 0 1
 ----------------
```

Substraction Rule:

- 0 - 0 = 0
- 0 - 1 = 1 (borrow 1)
- 1 - 0 = 1
- 1 - 1 = 0
- Practice

```
  1 1 1 0 0 1 1 0
-
  0 0 1 1 1 1 1 1
 ----------------
```

Number systems

1’s and 2’s Complement

- 1’s complement: 1's complement is a method of representing signed binary numbers by inverting all the bits of its positive counterpart.
- 2’s complement: 2's complement is a method of representing signed binary numbers where the negative of a number is obtained by inverting all the bits of its positive counterpart and adding 1 (1’s complement + 1)

- Example

```
  1 1 1 0 0 1 1 0
  ----------------
  0 0 0 1 1 0 0 1   (1’s complement)
+
                1
  ----------------
  0 0 0 1 1 0 1 0   (2’s complement)
```

Number systems

Another way for substraction

Actually, a substraction can be transformed to an addition with a 2's complement form of the second operand.

- Practice: Let's do this practice again using the above method and compare the result

```
  1 1 1 0 0 1 1 0
-
  0 0 1 1 1 1 1 1
 ------------------
```

Number systems
SUMMARY

SUMMARY:

- There are 2 mains elements of the number system: base and digit.
- Each number system has different ways of base and digit expression.
- There are 4 main number system: binary, decimal, octal, hexadecimal.
- Each number system can convert to another systems.
- Need to remember the rule to perform binary addition/substraction

COMPUTER SYSTEM

INFORMATION UNIT

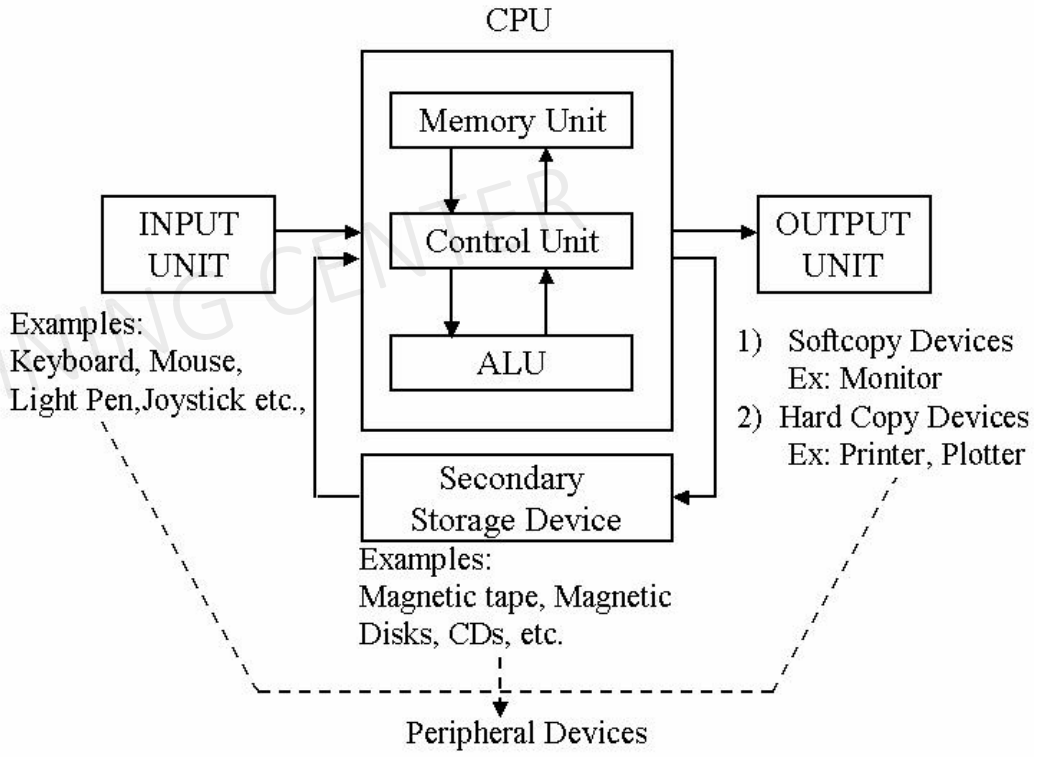
- Byte (B) is the smallest information unit.
- Each byte has 8 bits, therefore it can represent 256 different values.
- We can use KB, MB, GB, TB, PB … to represent bigger data value.
- 1 KB (Kilo Byte) = 1024 B
- 1 MB (Mega Byte) = 1024 KB
- 1 GB (Giga Byte) = 1024 MB
- 1 TB (Tetra Byte) = 1024 GB
- 1 PB (Peta Byte) = 1024 TB

…

COMPUTER SYSTEM

COMPUTER BLOCK DIAGRAM

- **CPU** (Central Processing Unit) : brain of a computer, performs most of the processing tasks that enable a computer to function.
- **BUS**: a communication system that allows components can "talk" (transfer data) to each other.
- **I/O** (Input/Output): Input devices provide a means for users to interact with the computer (keyboard, mouse, microphone, webcams …), while output devices display or convey the results of processing (monitor, printer, speaker …).
- **Memory**: electronic storage space used by a computer to store data temporarily or permanently (RAM, ROM, Flash….)
- **Peripheral**: external device connected to a computer that expands its capabilities or serves as an additional input/output component (hard drive, SSD, USB …)



Computer block diagram

COMPUTER SYSTEM

MEMORY ADDRESS CALCULATION

- Conceptually, memory is just a big array of bytes
- The memory addresses indexes a particular value in memory we want to access

⋮	⋮
5	0x22
4	0x8A
3	0xFF
2	0x08
1	0x12
0	0xAA

COMPUTER SYSTEM

MEMORY ADDRESS CALCULATION

- Memory space can be grouped into multi-byte chunks called "*word*".
- Two possibilities for addressing:
  - Word addressing: words are selected using word indices
  - Byte addressing: words are selected using the smallest byte index in the word.

Word Address	Byte Address				
⋮	⋮		⋮		
3	12	0x3F	0x15	0x28	0x8B
2	8	0x08	0x71	0xC6	0xD5
1	4	0x12	0xA6	0x78	0x95
0	0	0xAA	0xBC	0xF5	0x69

COMPUTER SYSTEM

MEMORY ADDRESS CALCULATION

- Address space in chip often uses byte addressing.
- Each IP is given a address range for register allocation.

Start address	End address	Size	Peripheral Register
0x4000_0000	0x4000_03FF	1KB	CAN
0x4000_0400	0x4000_13FF	4KB	USB
0x4000_1400	0x4000_1BFF	2KB	UART

COMPUTER SYSTEM

MEMORY ADDRESS CALCULATION

0x3FC	1020	0x3F	0x15	0x28	0x8B
0x3F8	1016	0x08	0x71	0xC6	0xD5
0x3F4	1012	0x12	0xA6	0x78	0x95
0x3F0	1008	0xAA	0xBC	0xF5	0x69
	⋮		⋮		
0x00C	12	0x3F	0x15	0x28	0x8B
0x008	8	0x08	0x71	0xC6	0xD5
0x004	4	0x12	0xA6	0x78	0x95
0x000	0	0xAA	0xBC	0xF5	0x69

1024 bytes (1KB)

- Example of byte addressing for a 1KB address space.
- Need 10-bit address for a 1KB address space

COMPUTER SYSTEM

ADDRESS SPACE CALCULATION

Practice:

- ❑ Having a 4KB address space starting from address 0x0, calculate the end address.
- ❑ Having a 8KB address space starting from 0x5000_1000, calculate the end address.
- ❑ Having an address space from 0x0010_0000 – 0x002F_FFFF. Calculate the size of that address space.

COMPUTER SYSTEM

ADDRESS SPACE CALCULATION

Homework:

- ❑ Having a 16KB address space end at from 0x6020_FFFF, calculate the start address.
- ❑ Having an address space from 0x5010_0000 – 0x5012_FFFF. Calculate the size of that address space.
- ❑ *Calculate the address bit-width of 16KB, 256KB, 1MB address space.

SUMMARY:

- Byte is the smallest information unit.
- Basic elements of a computer: CPU, BUS, I/O, Memory, Peripherals.
- Address size = number of bytes in an address range in hexadecimal number.