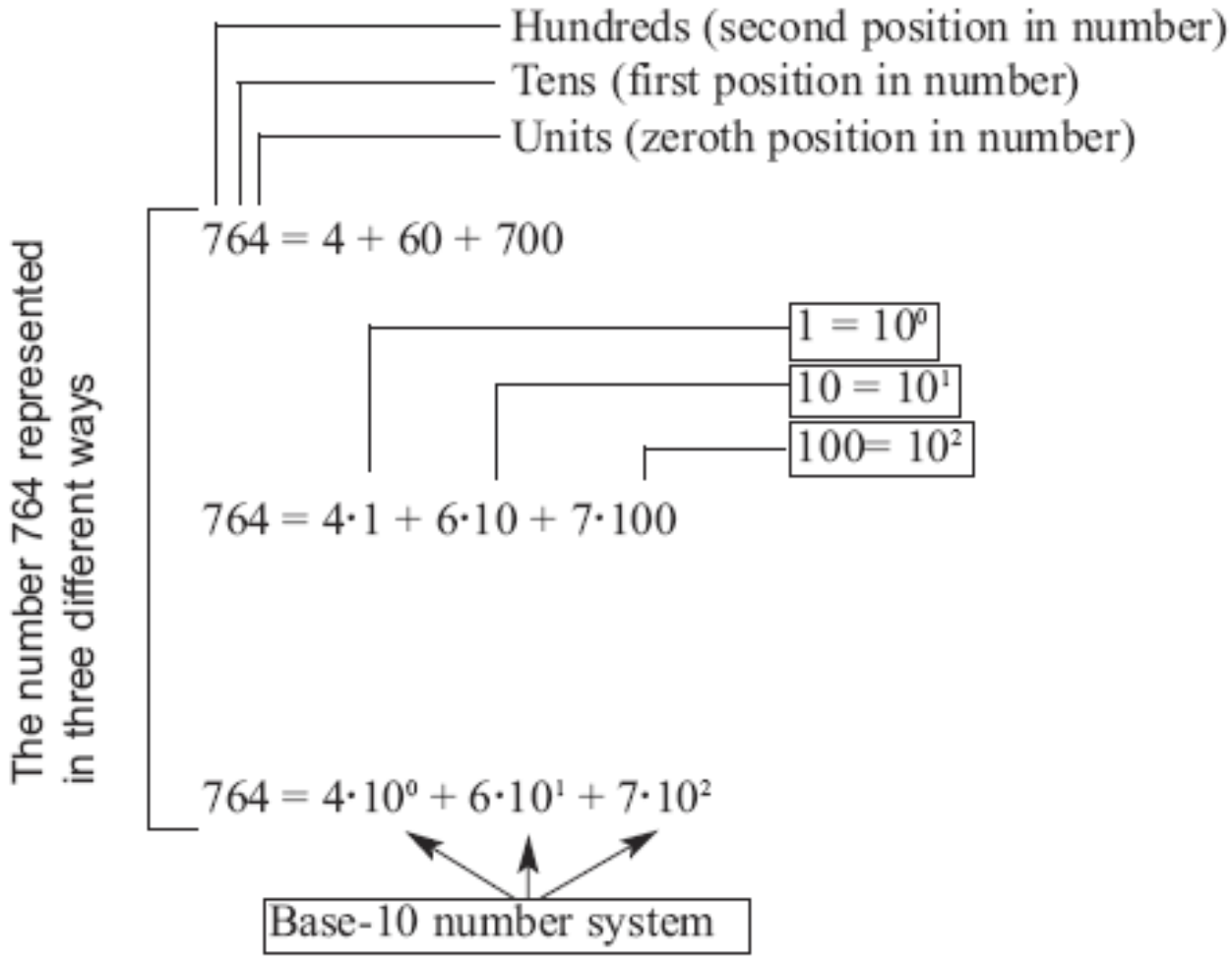
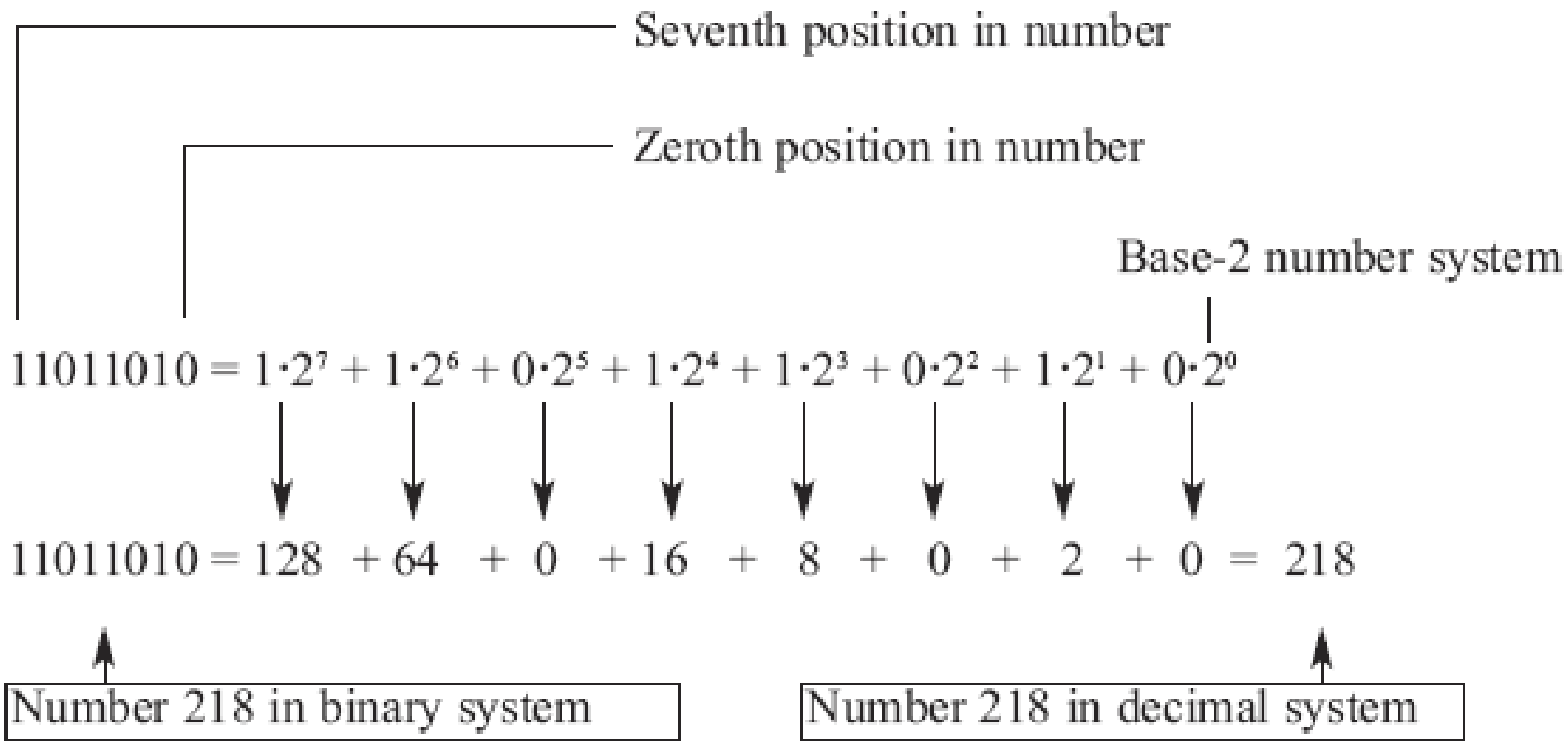
CÁC KIẾN THỨC CƠ BẢN

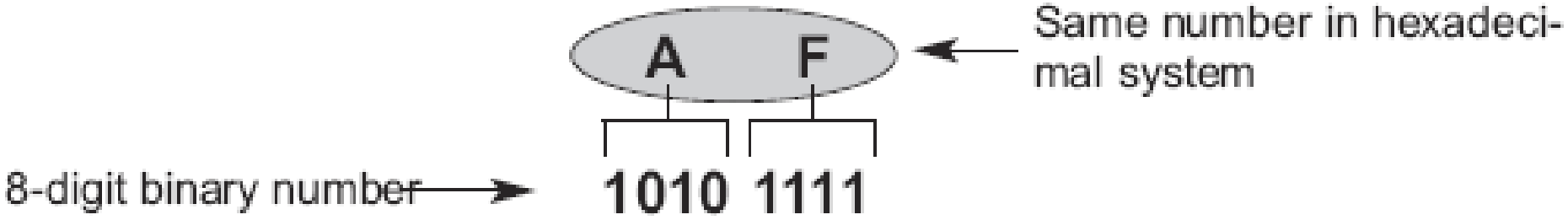
Hệ cơ số 10



Hệ cơ số nhị phân (BIN)



Hệ cơ số 16 (HEX)



= 10 x 16 + 15 = 175

Mã hóa BCD

INPUT		OUTPUT	
Dec	4-bit BCD	Dec(x5)	8-bit BCD (x5)
0	0000	0	0000 0000
1	0001	5	0000 0101
2	0010	10	0001 0000
3	0011	15	0001 0101
4	0100	20	0010 0000
5	0101	25	0010 0101
6	0110	30	0011 0000
7	0111	35	0011 0101
8	1000	40	0100 0000
9	1001	45	0100 0101

Chuyển các cơ số sau sang hệ thập phân

10101_2 100011_2

$$10101_2 = 10101B = 1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0 = 16+4+1= 21$$

$$100011_2 = 1\times2^5+0\times2^4+0\times2^3+0\times2^2+1\times2^1+1\times2^0$$
$$=32+2+1= 35$$

Chuyển các cơ số sau sang hệ thập phân

$$28_{16} \qquad 2F_{16} \qquad BC12_{16}$$

$$28_{16} = 28H = 2\times16^1+8\times16^0 = 40$$

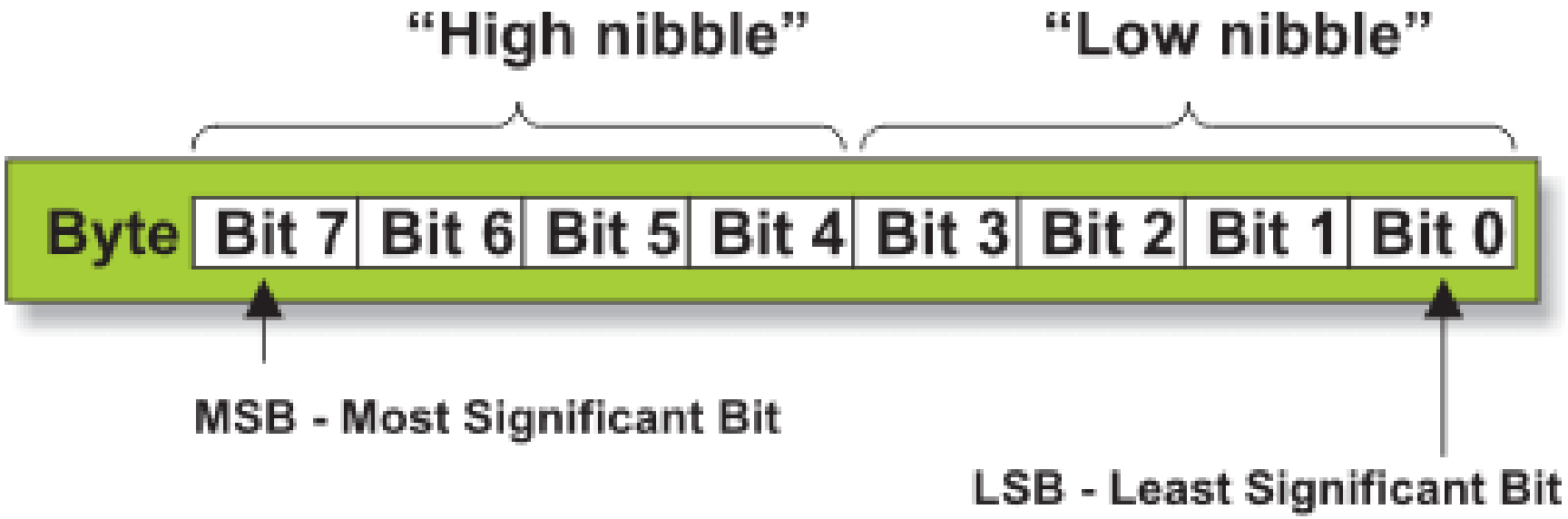
$$2F_{16} = 2FH = 2\times16^1+15\times16^0 = 47$$

$$BC12_{16} = BC12H = 11\times16^3+12\times16^2+1\times16^1+2\times16^0 = 48146$$

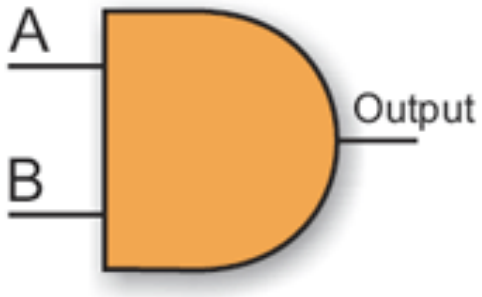
Chuyển hệ thập phân 67 và 4660 : sang hệ BIN, HEX, BCD

6	7	Decimal	4	6	6	0	Decimal
0110	0111	BCD	0100	0110	0110	0000	BCD
0100	0011	Binary	0001	0010	0011	0100	Binary
4	3	Hex	1	2	3	4	Hex

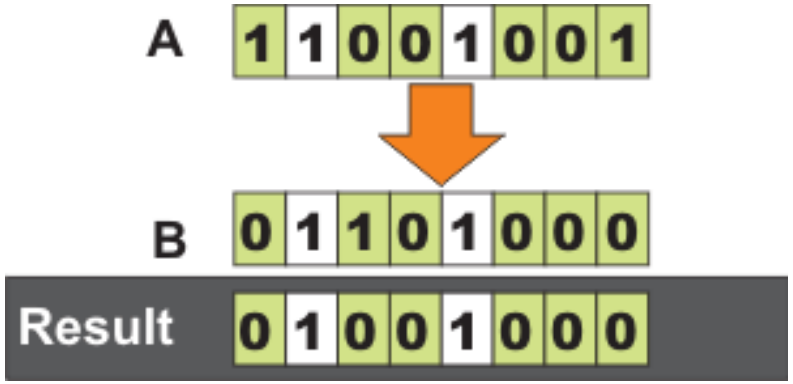
BIT và BYTE


“High nibble”
“Low nibble”
Byte
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0
MSB - Most Significant Bit
LSB - Least Significant Bit

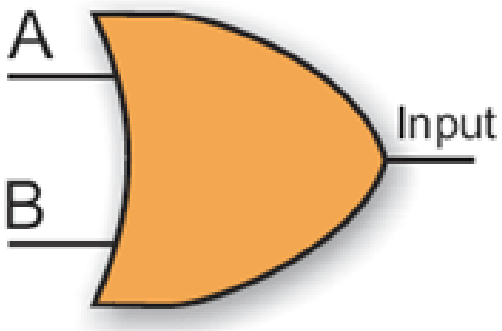
TOÁN TỬ LOGIC : AND



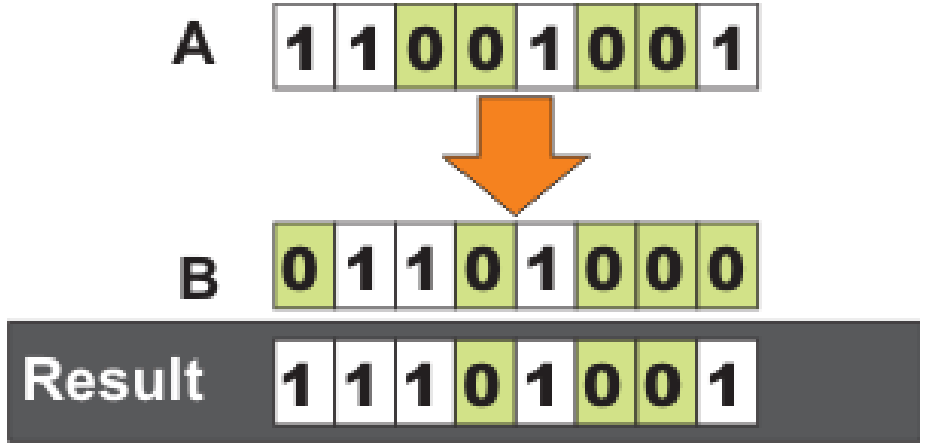
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



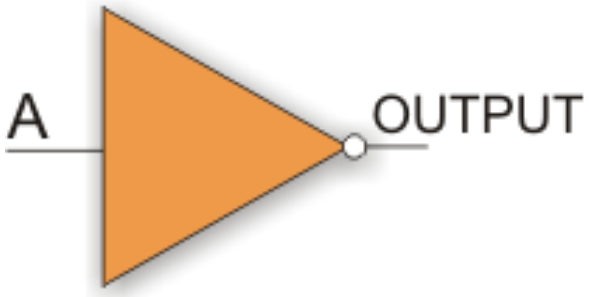
TOÁN TỬ LOGIC : OR



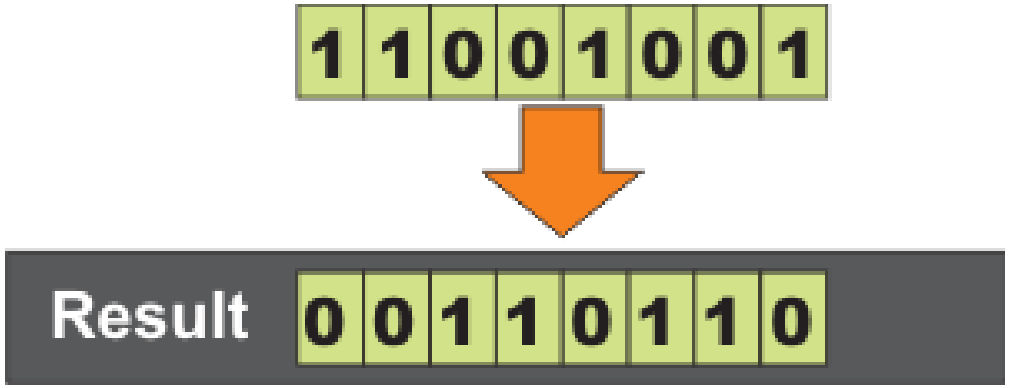
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



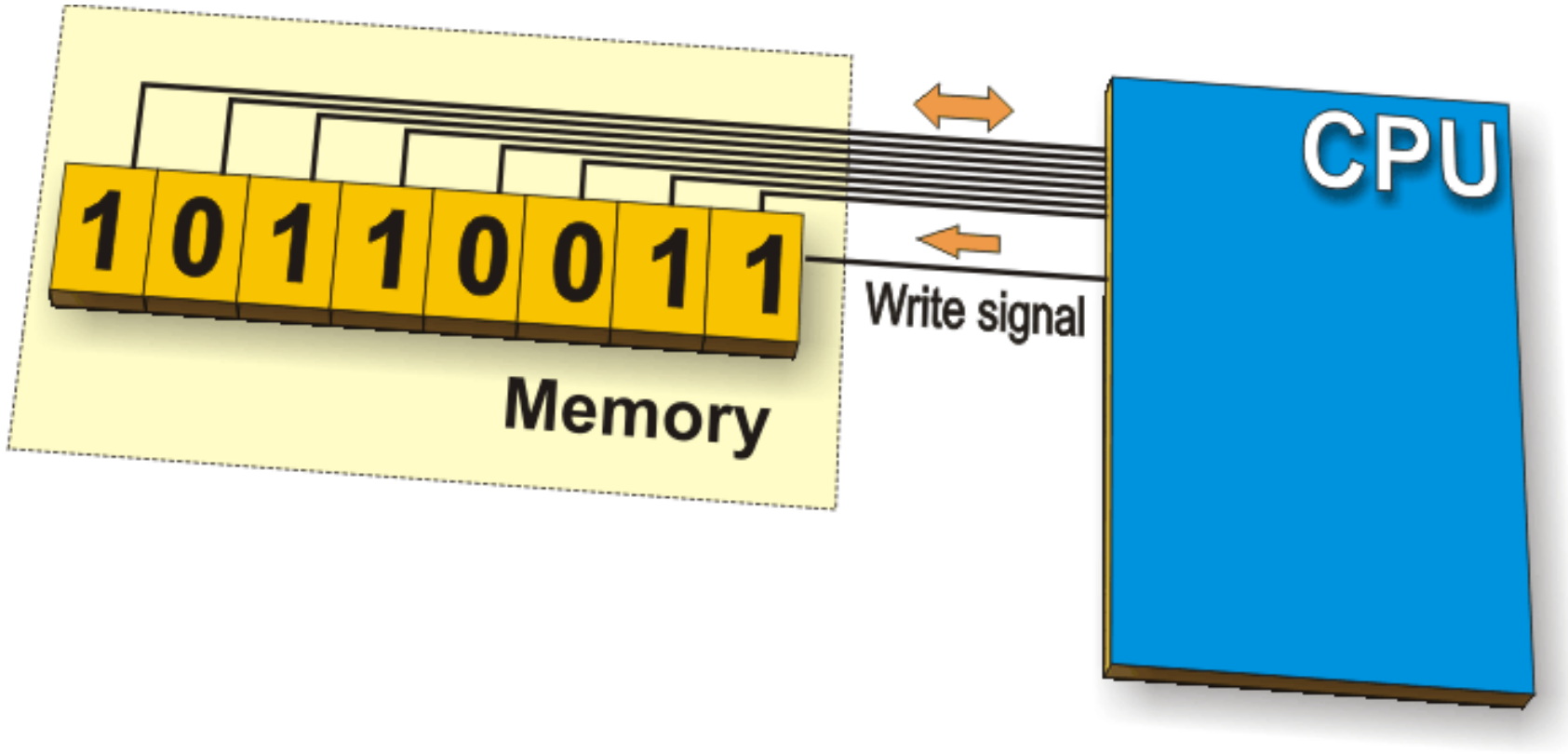
TOÁN TỬ LOGIC : NOT



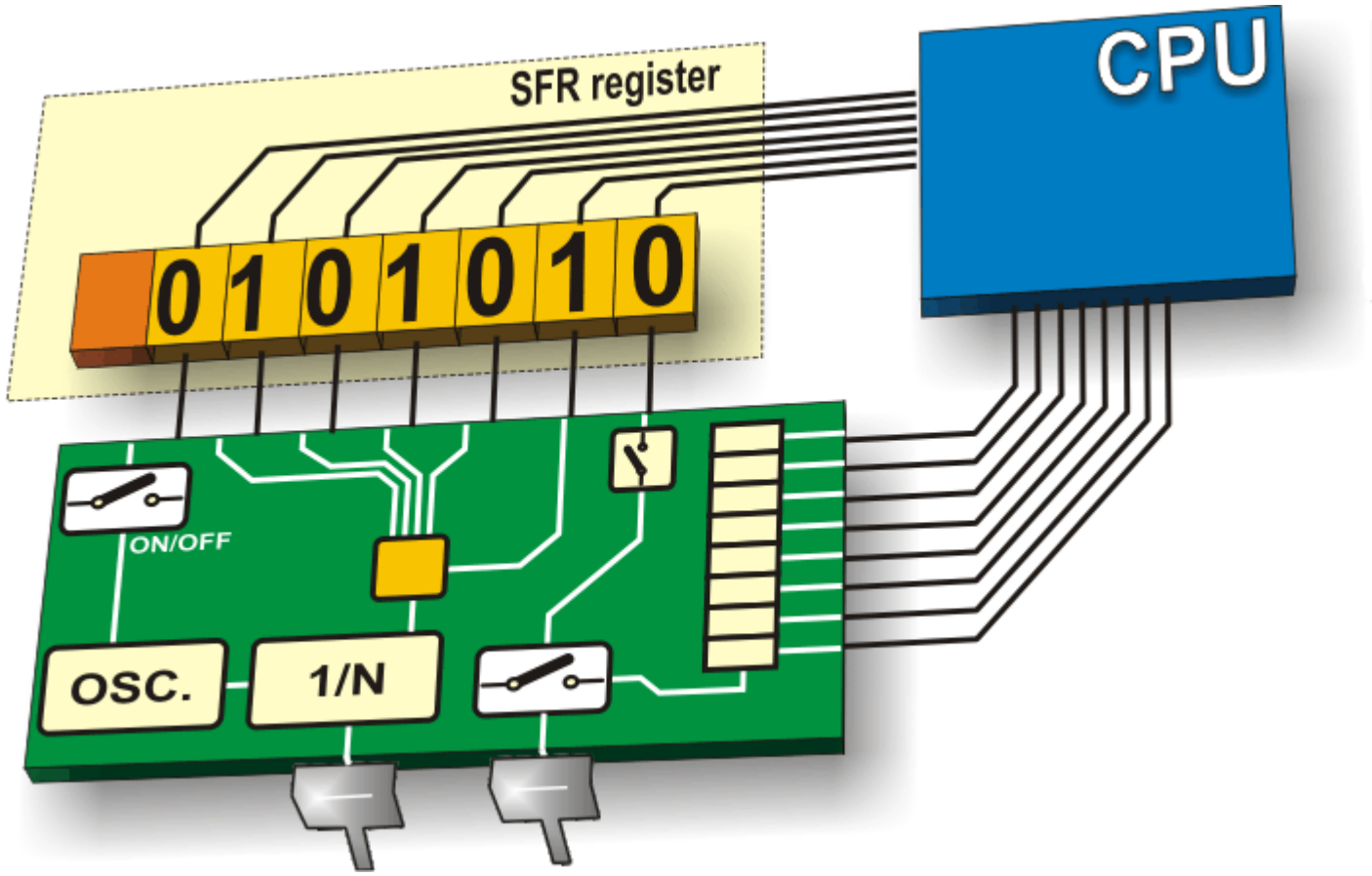
A	Output
0	1
1	0



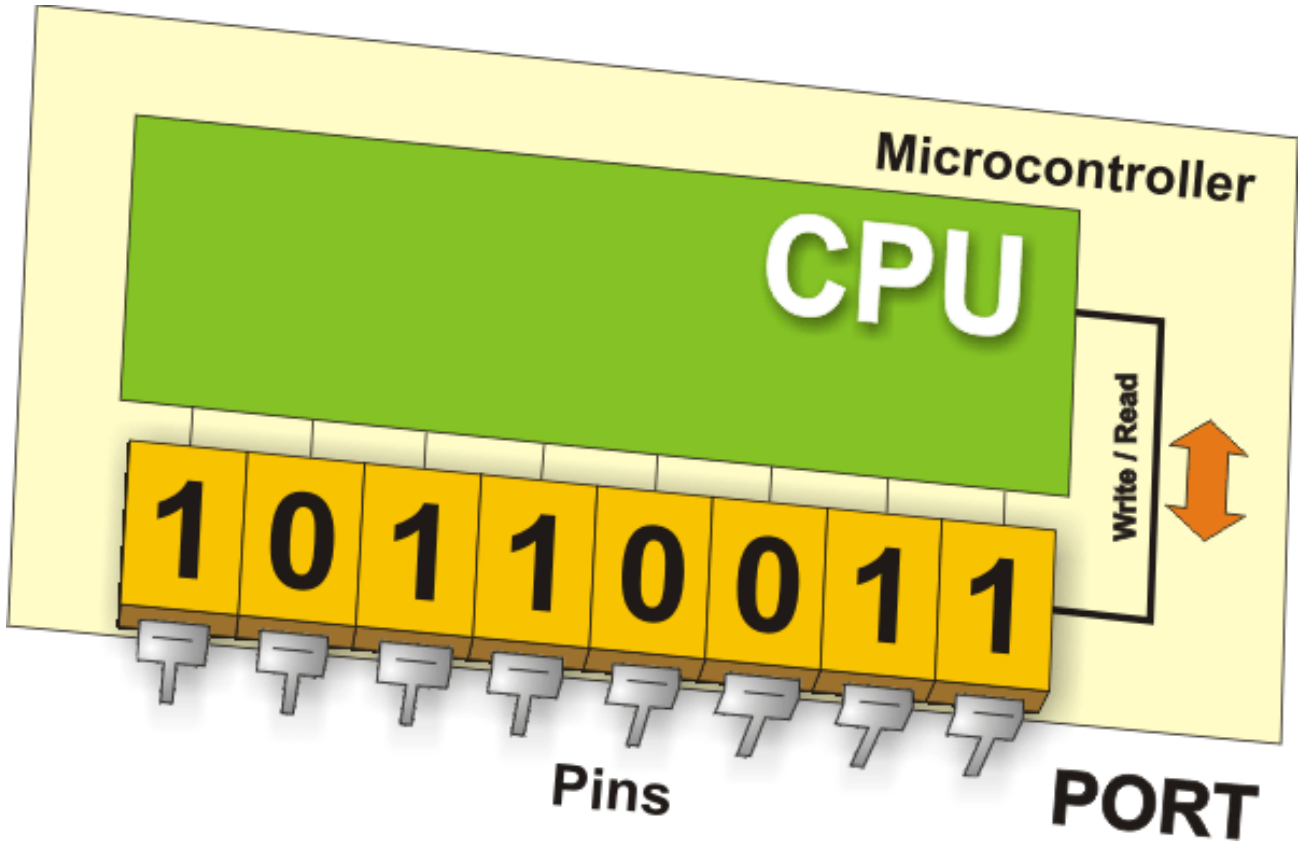
Register



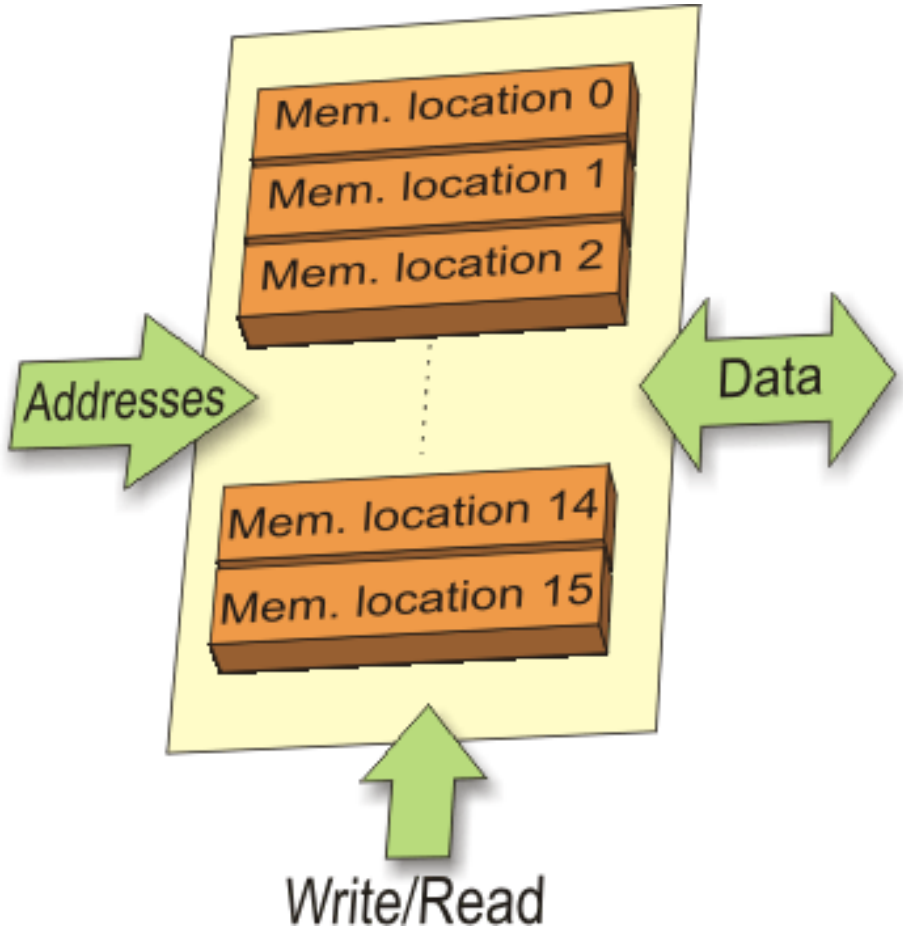
Special Function Register



Input / Output Ports



Memory Unit


Mem. location 0
Mem. location 1
Mem. location 2
Mem. location 14
Mem. location 15
Addresses
Data
Write/Read

Read Only Memory (ROM)

ROM (Read Only Memory) is used to permanently save the program being executed. The size of a program that can be written depends on the size of this memory. Today's microcontrollers commonly use 16-bit addressing, which means that they are able to address up to 64 Kb of memory, i.e. 65535 locations. As a novice, your program will rarely exceed the limit of several hundred instructions. There are several types of ROM.

Electrically Erasable Programmable ROM

The contents of the EEPROM may be changed during operation (similar to RAM), but remains permanently saved even upon the power supply goes off (similar to ROM). Accordingly, an EEPROM is often used to store values, created during operation, which must be permanently saved. For example, if you design an electronic lock or an alarm, it would be great to enable the user to create and enter a password, but useless if it is lost every time the power supply goes off. The ideal solution is the microcontroller with an embedded EEPROM.

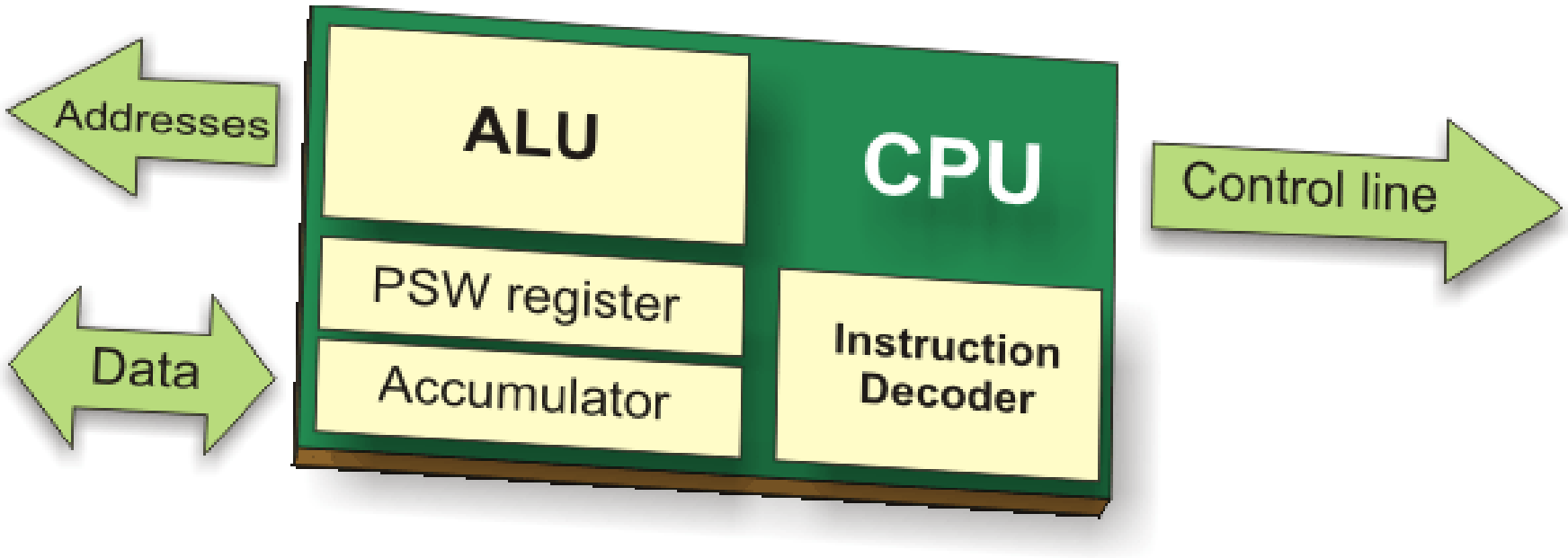
Flash memory

This type of memory was invented in the 80s in the laboratories of INTEL and were represented as the successor to the UV EPROM. Since the contents of this memory can be written and cleared practically an unlimited number of times, the microcontrollers with Flash ROM are ideal for learning, experimentation and small-scale manufacture. Because of its popularity, the most microcontrollers are manufactured in flash versions today. So, if you are going to buy a microcontroller, the type to look for is definitely Flash!

Random Access Memory (RAM)

Once the power supply is off the contents of RAM (Random Access Memory) is cleared. It is used for temporary storing data and intermediate results created and used during the operation of the microcontroller. For example, if the program performs an addition (of whatever), it is necessary to have a register representing what in everyday life is called the “sum”. For that purpose, one of the registers in RAM is called the “sum” and used for storing results of addition.

Central Processor Unit (CPU)

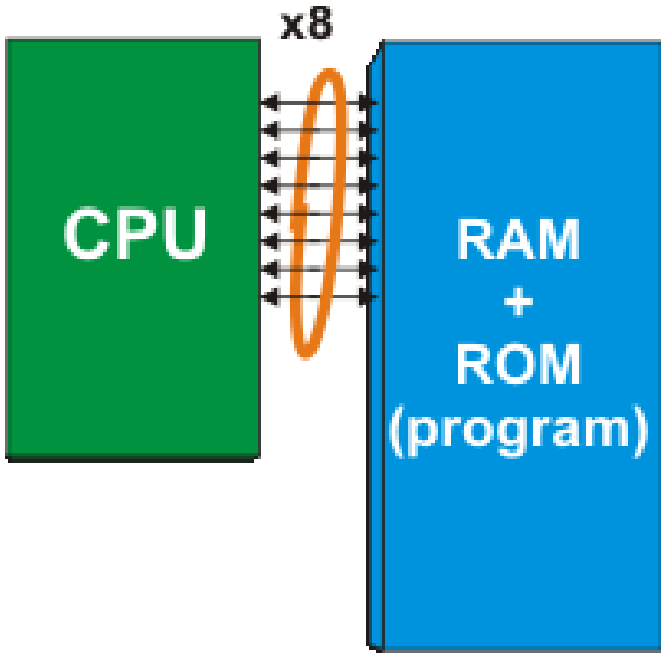


Bus

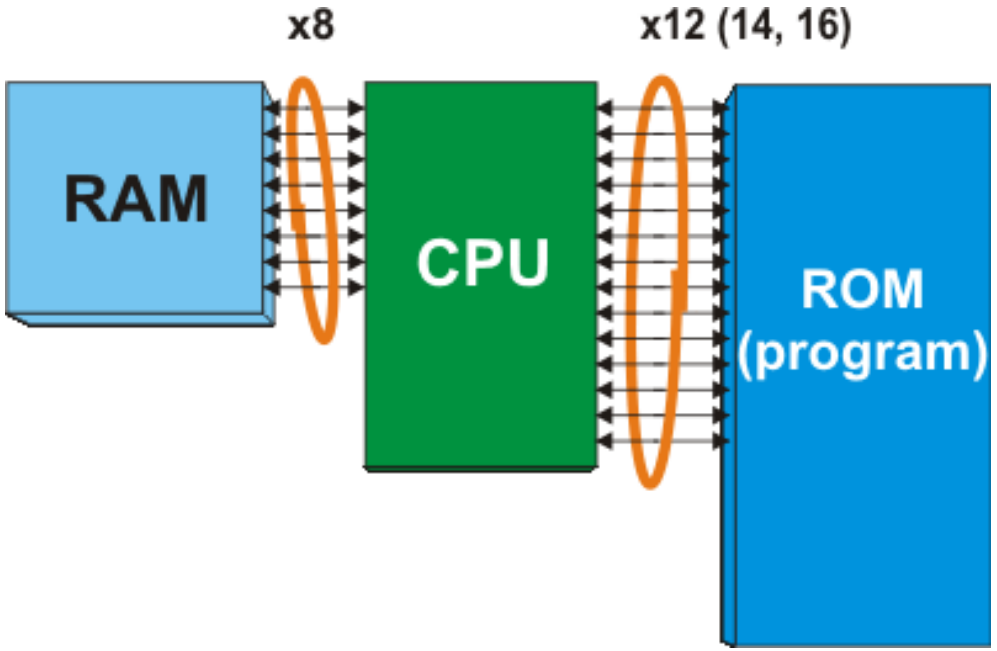
Physically, the bus consists of 8, 16 or more wires. There are two types of buses: the address bus and the data bus. The address bus consists of as many lines as necessary for memory addressing. It is used to transmit the address from the CPU to the memory. The data bus is as wide as the data, in our case it is 8 bits or wires wide. It is used to connect all circuits inside the microcontroller.

Kiến trúc vi điều khiển

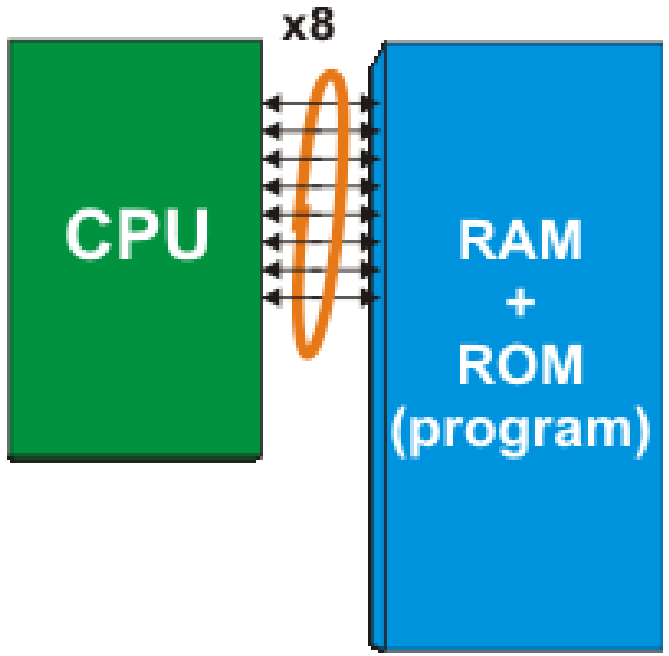
Von Neumann Structure



Havard Structure



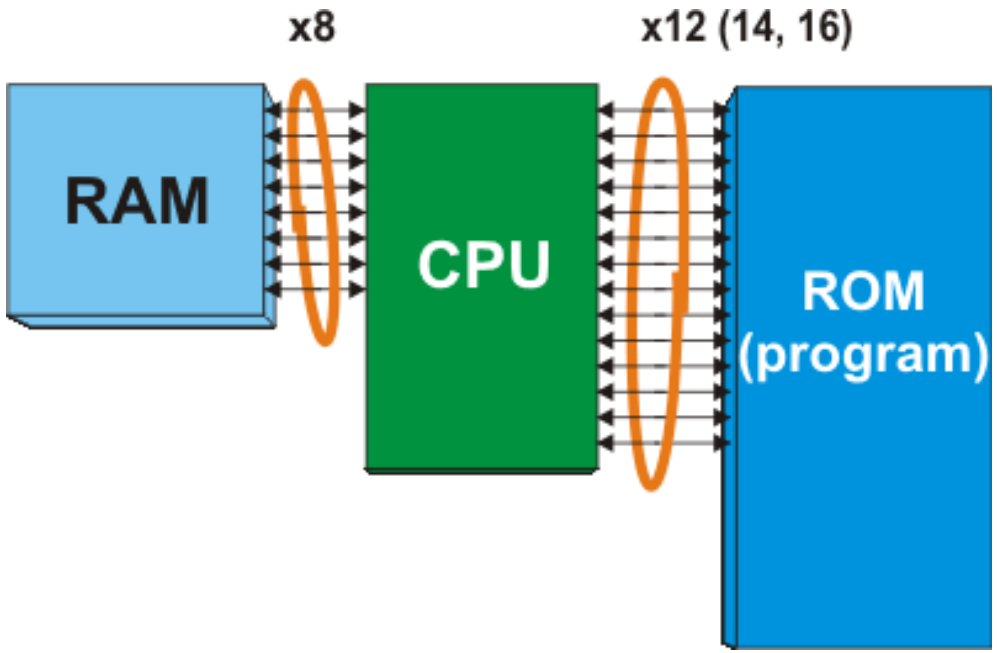
Von Neumann Structure



- Trao đổi dữ liệu giữa CPU và bộ nhớ là data bus 8-bit
- Dữ liệu trao đổi chậm và kém hiệu quả
- The CPU có thể đọc và viết vào bộ nhớ
- Không thể đọc và ghi cùng thời điểm trên đường bus

=> thực hiện các nhiệm vụ tuần tự. Hầu hết các máy tính ngày nay (và ARM7) đều sử dụng thiết kế Von Neumann. Sử dụng kiến trúc khác khi đòi hỏi xử lý rất nhanh, và có thể chi trả với giá cao.

Havard Structure



- Có 2 đường giao tiếp dữ liệu khác nhau
  - CPU – RAM : bus 8 bit
  - CPU – ROM : bus 12,14 hoặc 16 bit
- The CPU có thể đọc và xử lý lệnh đồng thời