

Bài 04

# HỆ THỐNG ĐẾM TRONG MÁY TÍNH

Giảng viên: ...

Email: ...



Khoa Công Nghệ Thông Tin  
Trường Đại Học Khoa Học Tự Nhiên

# Nội Dung

---

- Giới thiệu về hệ thống đếm
- Hệ thống đếm thập phân
- Biểu diễn dữ liệu bằng hệ nhị phân
- Thao tác toán học trên hệ nhị phân
- Cách phần cứng lưu trữ dữ liệu
- Thao tác ALU trên hệ nhị phân

# Giới thiệu về hệ thống đếm

- Hệ đếm là tập hợp các kí hiệu và qui tắc sử dụng tập kí hiệu đó để biểu diễn và xác định giá trị các số
- Mỗi hệ đếm có một số ký tự hữu hạn
- Tổng số ký tự của mỗi hệ đếm được gọi là *cơ số* (base hay radix), ký hiệu là  $b$
- Có 2 loại hệ đếm:
  - Hệ đếm không phụ thuộc vị trí: hệ đếm La Mã
  - Hệ đếm phụ thuộc vào vị trí: hệ nhị phân, hệ thập phân,...

# Hệ đếm La Mã

- Bảng kí hiệu

Kí hiệu	I	V	X	L	C	D	M
Giá trị	1	5	10	50	100	500	1000

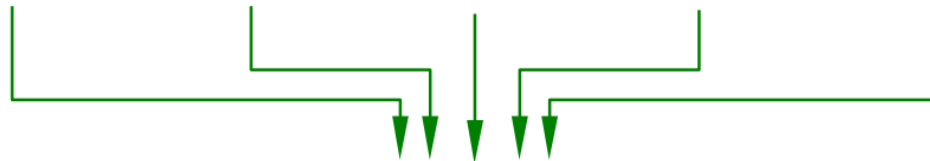
- Quy tắc

- Số La Mã được biểu diễn bằng cách viết các kí hiệu cạnh nhau
- Chữ số cơ bản được lặp lại không quá 3 lần
  - VD: I = 1      II = 2      III = 3
- Giá trị được tính theo qui tắc “*Phải cộng, trái trừ*”
  - VD:  
$$MLVI = 1000 + 50 + 5 + 1 = 1056$$
$$MLIV = 1000 + 50 + 5 - 1 = 1054$$

# Hệ đếm thập phân

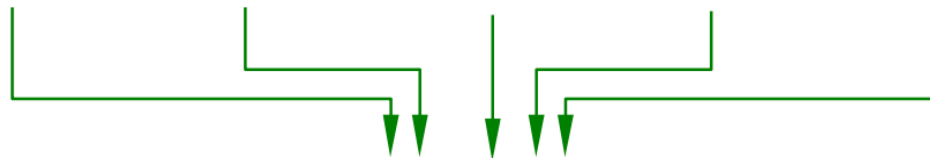
- Ký hiệu:  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Sử dụng 10 ký số này để biểu diễn 1 số
- Quy tắc: mỗi đơn vị ở 1 hàng bất kì có giá trị bằng 10 đơn vị của hàng kế cận bên phải

$$3 \times 10^4 + 5 \times 10^3 + 4 \times 10^2 + 6 \times 10^1 + 2 \times 10^0$$



35462

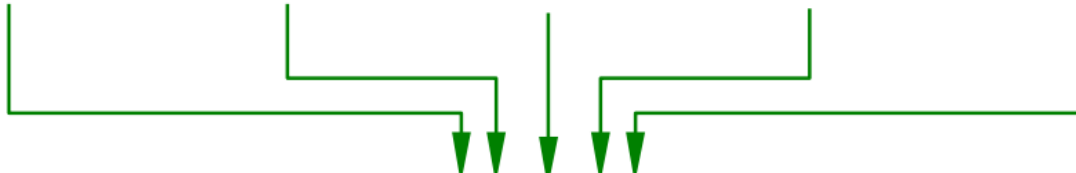
$$3 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 6 \times 10^{-2} + 2 \times 10^{-3}$$

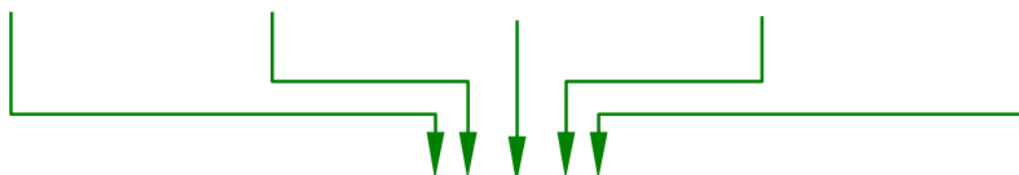


35.462

# Hệ đếm nhị phân

- Ký hiệu:  $\{0, 1\}$
- Quy tắc: vị trí 1 ở một hàng bất kì có giá trị bằng 2 đơn vị của hàng kế cận bên phải

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$10110_b = 22_d$$

$$1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$$

$$10.110_b = 2.75_d$$

# Hệ đếm nhị phân – Phép cộng

- Cộng có nhớ các cặp số cùng vị trí từ phải sang trái

**Bảng cộng**

+	0	1
0	0	1
1	1	10

**Ví dụ**

$$\begin{array}{rcccc} & 1 & 1 & 1 & 0 \\ + & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

# Hệ đếm nhị phân – Phép trừ

- Số bù 1
  - Đảo tất cả các ký số của 1 số nhị phân ta sẽ có được số bù 1 của nó
- Số bù 2
  - Lấy số bù 1 cộng 1 ta được số bù 2 của số nhị phân ban đầu
- Ví dụ:  $x = 1010$ 
  - Số bù 1 của  $x$  : 0101
  - Số bù 2 của  $x$  : 0110



# Hệ đếm nhị phân – Phép trừ

- Cho hai số nhị phân  $x$  và  $y$

$$x - y = x + \text{số bù 2 của } y$$

- Ví dụ

- $x = 1010, y = 0101$
- Số bù 1 của  $y$  :  $1010$
- Số bù 2 của  $y$  :  $1011$  ( $y_2$ )
- $x - y = x + y_2 = 1010 + 1011 = 0101$

# Hệ đếm nhị phân – Phép nhân

- Nhân từ phải qua trái theo cách nhân tay thông thường

**Bảng nhân**

x	<b>0</b>	<b>1</b>
<b>0</b>	0	0
<b>1</b>	0	1

**Ví dụ**

$$\begin{array}{r} \phantom{x} \phantom{0000} 1 \phantom{00} 0 \phantom{000} 1 \phantom{0000} 1 \\ x \phantom{00000000} \phantom{00000} 1 \phantom{000000} 0 \\ \hline \phantom{00000000} 0 \phantom{0000000} 0 \phantom{00000000} 0 \phantom{00000000} 0 \\ \\ 1 \phantom{00000000} 0 \phantom{00000000} 1 \phantom{00000000} 1 \\ \hline 1 \phantom{00000000} 0 \phantom{00000000} 1 \phantom{00000000} 1 \phantom{00000000} 0 \end{array}$$

# Hệ đếm nhị phân – Phép chia

- Thực hiện tương tự như phép chia trong hệ cơ số 10

- Ví dụ

$$\begin{array}{r} \text{—} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \bigg| \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \\ \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \quad \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \\ \hline \quad \mathbf{0} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \quad \quad \\ \quad \text{—} \quad \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \quad \quad \\ \quad \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1} \quad \quad \quad \\ \quad \quad \text{—} \quad \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \quad \quad \\ \quad \quad \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \leftarrow \text{Số dư} \end{array}$$

# Hệ đếm thập lục phân

- Ký hiệu:  $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E\}$
- Qui ước:  $A = 10, B=11, C=12, D=13, E=14, F=15$ 
  - VD:  
 $6654733_d = 0110-0101-1000-1011-0000-1101_b = 658B0D_h$
- Dùng tiếp đầu ngữ **0x** để đánh dấu sự khác biệt giữa hệ thập phân và thập lục phân
  - VD:  $0x658B0D = 6654733_d, 0x123 = 291_d$

# Biểu diễn cơ số bất kỳ

- Có  $b$  ký tự để thể hiện giá trị của số
- Ký tự nhỏ nhất là  $0$ , lớn nhất là  $b-1$
- Số  $N_{(b)}$  trong hệ đếm cơ số  $b$  được biểu diễn như sau

$$N_{(b)} = a_n a_{n-1} \dots a_0 a_{-1} \dots a_{-m}$$

và có giá trị

$$N_{(b)} = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 \\ + a_{-1} b^{-1} + \dots + a_{-m} b^{-m}$$

# Biểu diễn cơ số bất kỳ

- Trong đó
  - $b$  là cơ sở của biểu diễn,  $b \in N, b \geq 2$
  - $a_i$  là các ký số và  $a_i \in N, 0 \leq i \leq n, 0 \leq a_i < b$
  - Cách viết trên được gọi là biểu diễn cơ sở  $b$  của  $a$
  - Chiều dài của biểu diễn bằng  $n+1$
  - Nếu có số lẻ thì vị trí đầu tiên sau dấu phẩy là  $-1$ , các vị trí tiếp theo là  $-2, -3, \dots, -m$

# Nhân xét

---

- Máy tính sử dụng số được mã hóa trong *hệ cơ số 2, 8, 16*
- Con người quen thuộc với số được mã hóa trong *hệ cơ số 10*
- Chuyển đổi qua lại giữa các hệ cơ số
  - Từ hệ cơ số 10 sang hệ cơ số 2
  - Từ hệ cơ số 16 sang hệ cơ số 10

# Chuyển base-b $\rightarrow$ base-10

- Khai triển biểu diễn và tính giá trị biểu thức

- Ví dụ

$$\begin{aligned} 1011.01_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 0 + 2 + 1 + 0 + 0.25 \\ &= 11.25_{10} \end{aligned}$$



# Chuyển base-10 $\rightarrow$ base-b

- **Đổi phần nguyên**

- Chia phần nguyên cho  $b$  và tiếp tục lấy phần nguyên của kết quả chia cho  $b$ . Thực hiện cho đến khi thương của phép chia là 0.
- Dãy các số dư ở mỗi lần chia là  $a_0, a_1, \dots, a_n$
- Phần nguyên của số hệ cơ sở  $b$  là  $(a_n \dots a_1 a_0)$

- **Đổi phần lẻ**

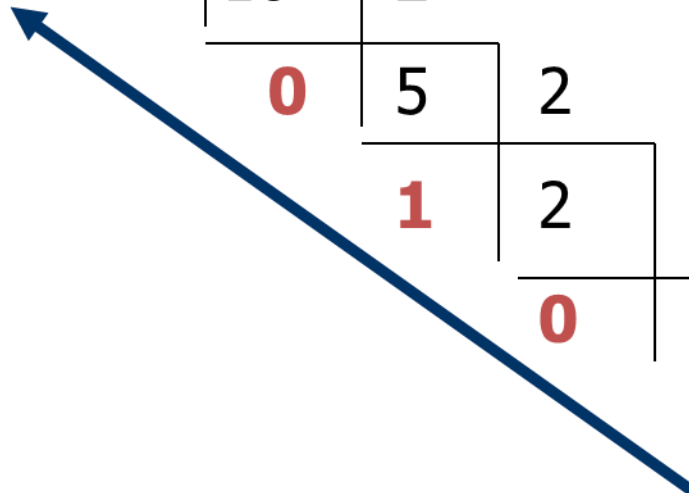
- Nhân phần lẻ cho  $b$  và tiếp tục lấy phần lẻ của kết quả nhân cho  $b$ . Tiếp tục cho tới khi nào phần lẻ của tích là 0.
- Dãy các số nguyên ở mỗi lần nhân là  $a_{-1}, a_{-2}, \dots, a_{-m}$  tạo thành phần lẻ ở hệ cơ sở  $b$

# Chuyển base-10 $\rightarrow$ base-b

- Ví dụ chuyển  $21.125_{10}$  sang hệ nhị phân

**Đổi phần nguyên**

21	2				
<b>1</b>	10	2			
	<b>0</b>	5	2		
		<b>1</b>	2	2	
			<b>0</b>	1	2
				<b>1</b>	<u>0</u>



**Đổi phần lẻ**

$$0.125 \times 2 = \mathbf{0}.25$$

$$0.25 \times 2 = \mathbf{0}.5$$

$$0.5 \times 2 = \mathbf{1}.\mathbf{\underline{0}}$$

**Kết quả:**  $21.125 = 10101.001$

# Chuyển base-2 $\rightarrow$ base-b

- Từ base-2 sang base-16
  - Nhóm từng bộ 4 ký số trong biểu diễn nhị phân rồi chuyển sang ký số tương ứng trong hệ 16

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

# Chuyển base-2 $\rightarrow$ base-b

- Từ base-2 sang base-8
  - Nhóm từng bộ 3 ký số trong biểu diễn nhị phân rồi chuyển sang ký số tương ứng trong hệ bát phân

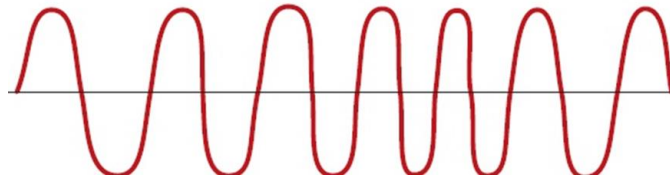
8	2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

# Biểu diễn thông tin

- Có 2 cách lưu trữ và quản lí thông tin

- **Analog**

- Liên tục, có nhiều cường độ khác nhau và tương đồng với dữ liệu trình bày
- VD: âm thanh, nhiệt độ, cường độ, ánh sáng,...



- **Digital**

- Rời rạc, chỉ gồm 2 mức cao (1) và thấp (0)
- Thông tin được chia nhỏ thành từng phần và lưu trữ bằng hệ đếm nhị phân
- VD: compact disc



# Digital information

---

- Mọi thông tin trên máy tính được lưu trữ dạng digital
  - Số
  - Chuỗi
  - Hình ảnh
  - Âm thanh
  - Video
  - Chỉ thị lệnh
  - ...
- Thông tin được số hóa bằng cách chia nhỏ và biểu diễn dạng số

# Digital information

- Mỗi ký tự nhị phân (0,1) được gọi là 1 bit (**B**inary **D**igit)
- Mỗi bit có thể nhận 1 trong 2 trạng thái 0 hoặc 1
- Tổ hợp các bit  $\Rightarrow$  thông tin

## 1 bit

0

1

## 2 bits

00

01

10

11

## 3 bits

000

001

010

011

100

101

110

111

## 4 bits

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

1110

1111

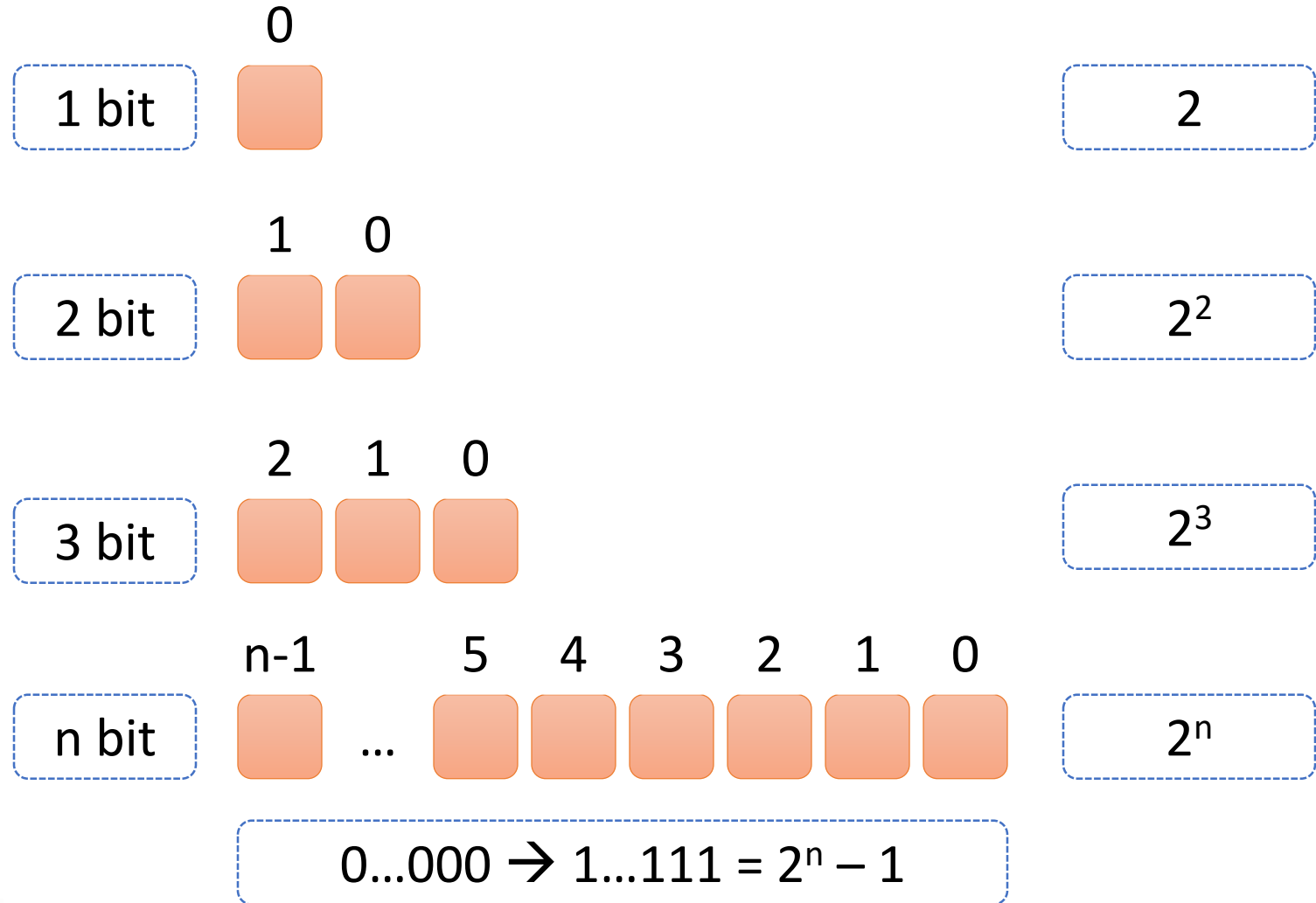
Thêm một bit  $\Rightarrow$  số trạng thái  $\times 2$

# Digital information

1 bit ?	$2^1 = 2$	}	N bits $\Rightarrow 2^N$
2 bits ?	$2^2 = 4$		
3 bits ?	$2^3 = 8$		
4 bits ?	$2^4 = 16$		
5 bits ?	$2^5 = 32$		



# Đơn vị lưu trữ thông tin



# Biểu diễn văn bản

- Mỗi ký tự được mã hóa bởi một số.
- Ký tự gồm: số, chữ, khoảng trắng, dấu câu
- ASCII
  - Dùng mã hóa văn bản không dấu
  - Được biểu diễn bằng 7 bit (0→127)
  - Bảng mã ASCII mở rộng chứa 256 mô tả cặp ký tự số.
    - 128 ký tự đầu giống ASCII ban đầu.
    - 128 ký tự sau bao gồm 1 số ký hiệu tiếng Hy Lạp ('α', 'β', 'π', ...), các biểu diễn tiền tệ ('£', '¥', ...), ...
  - Bảng mã ASCII không thể biểu diễn các ký tự của các ngôn ngữ khác như tiếng Việt (có dấu), Nga, Nhật, ...

# Biểu diễn văn bản

0	NUL	16	DLE	32	SPC	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

“Hello” =      **72**            **101**            **108**            **108**            **111**  
                   **1001000** **1100101** **1101100** **1101100** **1101111**

# Biểu diễn văn bản

## ▪ Bảng mã Unicode (UTF-8):

- Dùng mã hóa chuỗi có dấu
- Gồm 1,114,112 kí tự và được biểu diễn theo 4 mẫu như hình

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Character		Code point	UTF-8
		Binary	Binary
\$	U+0024	010 0100	00100100
¢	U+00A2	000 1010 0010	11000010 10100010
₹	U+0939	0000 1001 0011 1001	11100000 10100100 10111001
🌀	U+10348	0 0001 0000 0011 0100 1000	11110000 10010000 10001101 10001000



# Biểu diễn văn bản

- Sự khác biệt giữa ASCII và Unicode

ASCII	Unicode
7 bit biểu diễn 1 kí tự	16, 24, 32 bit biểu diễn 1 kí tự
Biểu diễn được 128 kí tự khác nhau	Biểu diễn được 1,114,112 kí tự khác nhau
Chữ cái tiếng Anh, số và kí hiệu	Hỗ trợ tất cả kí tự thuộc ngôn ngữ khác nhau
Yêu cầu ít không gian	Yêu cầu nhiều không gian

# Biểu diễn dữ liệu số

- **Biểu diễn số nguyên không dấu**

- Biểu diễn các đại lượng luôn dương. Ví dụ: chiều cao, cân nặng, mã ASCII...

- Tất cả bit được sử dụng để biểu diễn giá trị.

- Số nguyên không dấu 1 byte lớn nhất là

$$1111\ 1111_2 = 2^8 - 1 = 255_{10}.$$

- Số nguyên không dấu 1 word lớn nhất là

$$1111\ 1111\ 1111\ 1111_2 = 2^{16} - 1 = 65535_{10}.$$

- Tùy nhu cầu có thể sử dụng số 2, 3... word.

# Biểu diễn dữ liệu số

- **Biểu diễn số nguyên có dấu**

- Lưu các số dương hoặc âm.
- Bit cao nhất dùng để biểu diễn dấu
- Ví dụ đối với số nguyên sử dụng 8 bit lưu trữ (gọi tắt là số nguyên 8 bits)
  - 0 biểu diễn số dương. VD: 0101 0011
  - 1 biểu diễn số âm. VD: 1101 0011
- Số âm trong máy được biểu diễn ở dạng số bù 2.

# Biểu diễn thông tin

- Số bù 1 và số bù 2

Số 5 (1 byte)

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

Số bù 1 của 5

1	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---

+

1
---

Số bù 2 của 5

1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---

+ Số 5

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

Kết quả

1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---



# Biểu diễn dữ liệu số

- **Biểu diễn số nguyên có dấu**

- Số bù 2 của  $x$  cộng với  $x$  là một dãy toàn bit 0 (**không tính bit 1 cao nhất do vượt quá phạm vi lưu trữ**). Do đó số bù 2 của  $x$  chính là giá trị âm của  $x$  hay  $-x$ .
- Đổi số thập phân **âm**  $-5$  sang nhị phân?  
 $\Rightarrow$  Đổi 5 sang nhị phân rồi lấy số bù 2 của nó.
- Thực hiện phép toán  $a - b$ ?  
 $a - b = a + (-b) \Rightarrow$  Cộng với số bù 2 của  $b$

# Biểu diễn dữ liệu số

- **Tính giá trị không dấu và có dấu của 1 số?**
  - Ví dụ số word (16 bit): **1100 1100 1111 0000**
  - Số nguyên không dấu ?
    - Tất cả 16 bit lưu giá trị.
    - $\Rightarrow$  giá trị là **52464**.
  - Số nguyên có dấu ?
    - Bit cao nhất = **1** do đó số này là **số âm**.
    - $\Rightarrow$  độ lớn là giá trị của số bù 2.
    - Số bù 2 = **0011 0011 0001 0000** = **13072**.
    - $\Rightarrow$  giá trị là **-13072**.

# Biểu diễn dữ liệu số

- **Biểu diễn số thực**

- Sử dụng dấu chấm động (floating-point).
- Ví dụ:
  - Số thực hệ 10:  $-123.4d = -12.34 \times 10^1 = -1.234 \times 10^2 = -0.1234 \times 10^3$
- Biểu diễn khoa học:  $-1.234 \times 10^2$
- Chia làm 3 phần:
  - 1 bit để biểu diễn dấu.
  - Một chuỗi bit để biểu diễn số mũ.
  - Một chuỗi bit để biểu diễn phần định trị

# Biểu diễn dữ liệu số

- **Biểu diễn số thực**

- Với ví dụ trên:  $-1.234 \times 10^2$ 
  - Bit biểu diễn dấu là 1 (ứng với giá trị âm)
  - Biểu diễn số mũ là 2
  - Biểu diễn phần trị là 1234
    - Quy ước bên trái dấu chấm là 1 ký số khác không

# Biểu diễn hình ảnh

- **Biểu diễn hình ảnh**

- Hình ảnh được mã hóa thành dãy nhị phân từ trái qua phải, từ trên xuống dưới

0	0	0	0	0	0	000000
0	1	0	0	1	0	010010
0	1	0	0	1	0	010010
0	0	0	0	0	0	000000
0	1	1	1	1	0	011110
0	0	0	0	0	0	000000

- Hình màu (4 màu) cần 2 bit cho mỗi điểm ảnh (pixel).

00 – màu trắng

01 – màu đen

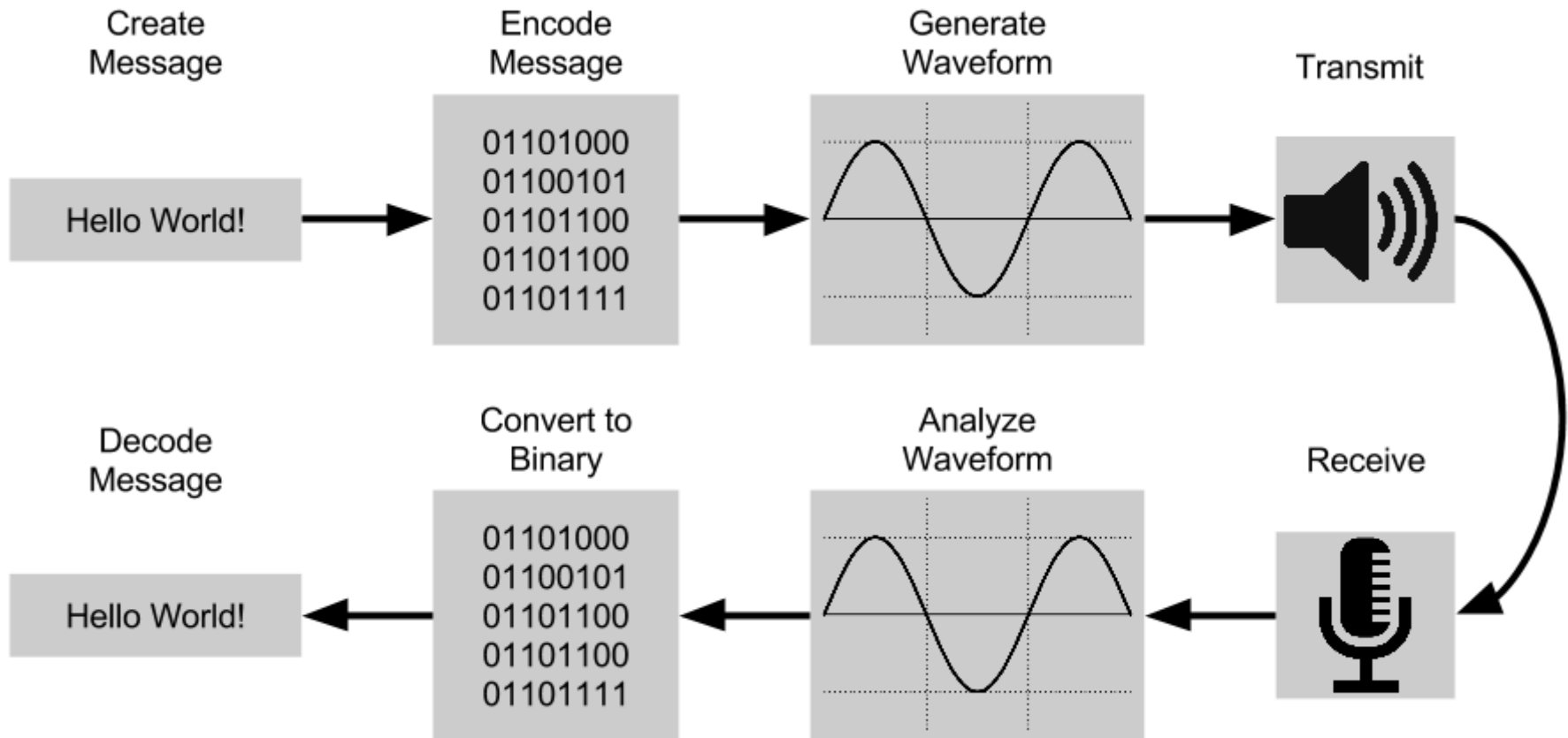
10 – màu vàng

11 – màu xanh

10	10	10	10	10	10	101010101010
10	00	10	10	00	10	100010100010
10	11	10	10	11	10	101110101110
10	10	10	10	10	10	101010101010
10	01	01	01	01	10	100101010110
10	10	10	10	10	10	101010101010

# Biểu diễn âm thanh

- Biểu diễn âm thanh



# Lưu trữ thông tin

---

- Cấu hình máy tính:
  - 2.8 GHz Pentium 4 Processor
  - 512 MB RAM
  - 80 GB Hard Disk
  - 48x CD-RW / DVD-ROM Combo Drive
  - 17” Video Display with 1280 x 1024 resolution
  - 56 Kb/s Modem

**Các thông số này có ý nghĩa gì?**

# Bộ nhớ



A diagram illustrating memory cells. On the left, a vertical list of addresses from 9278 to 9286 is shown in blue text. To the right of these addresses is a vertical column of eight empty rectangular boxes, each representing a memory cell. Above and below this column are three blue dots, indicating that the sequence of cells continues in both directions.

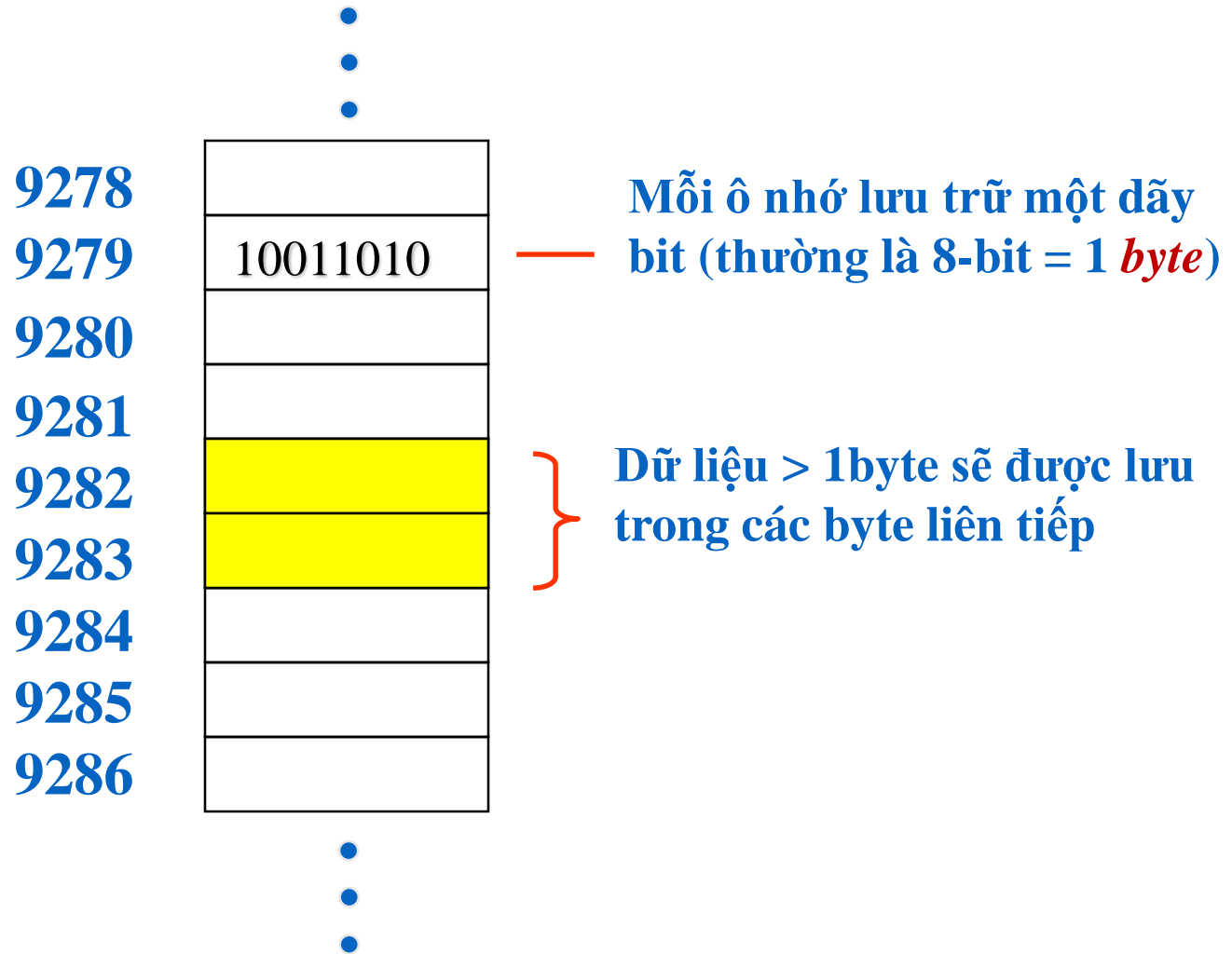
9278	
9279	
9280	
9281	
9282	
9283	
9284	
9285	
9286	

Bộ nhớ chính được chia thành nhiều phần gọi là ô nhớ (*cells*)

Mỗi ô nhớ có một địa chỉ (*address*) duy nhất



# Bộ nhớ



# Khả năng lưu trữ

- Mỗi thiết bị có một khả năng lưu trữ cho biết số bytes có thể lưu
- Khả năng lưu trữ được tính bằng đơn vị:

<u>Đơn vị</u>	<u>Ký hiệu</u>	<u>Số byte</u>
kilobyte	KB	$2^{10} = 1,024$
megabyte	MB	$2^{20} (1,048,576)$
gigabyte	GB	$2^{30} (1,073,741,824)$
terabyte	TB	$2^{40} (1,099,511,627,776)$
petabyte	PB	$2^{50} (1,125,899,906,842,624)$
exabyte	EB	$2^{60} (1,125,921,504,606,846,976)$
zettabyte	ZB	$2^{70} (1,180,591,620,717,411,303,424)$
yottabyte	YB	$2^{80} (1,208,925,819,614,629,174,706,176)$

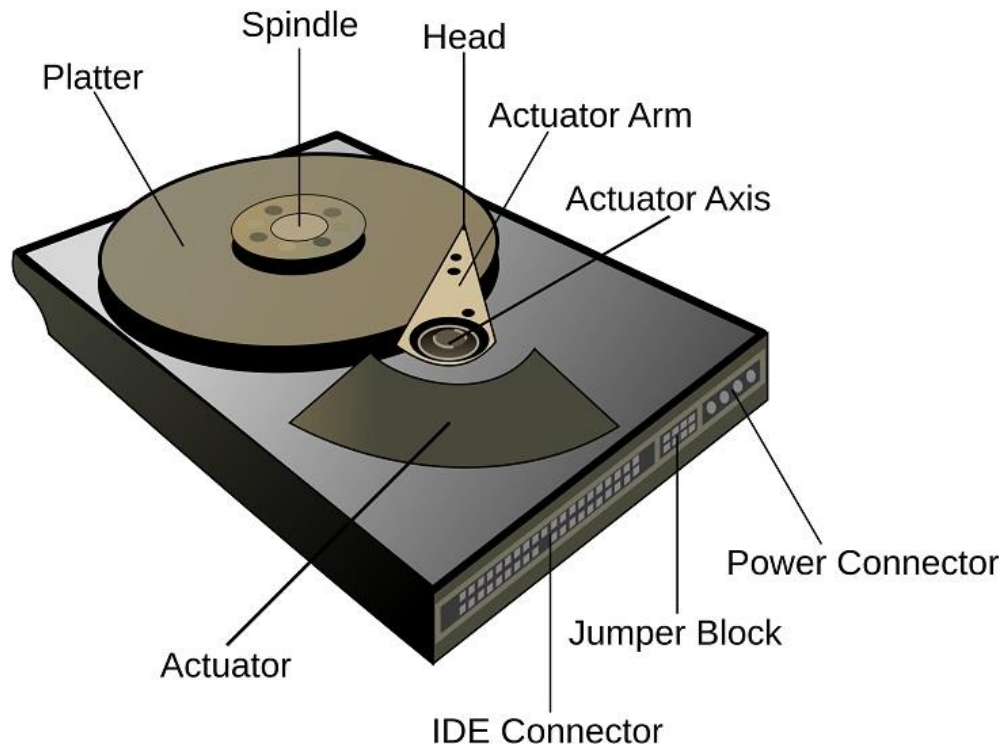
# Hard disk drive (HDD)

---

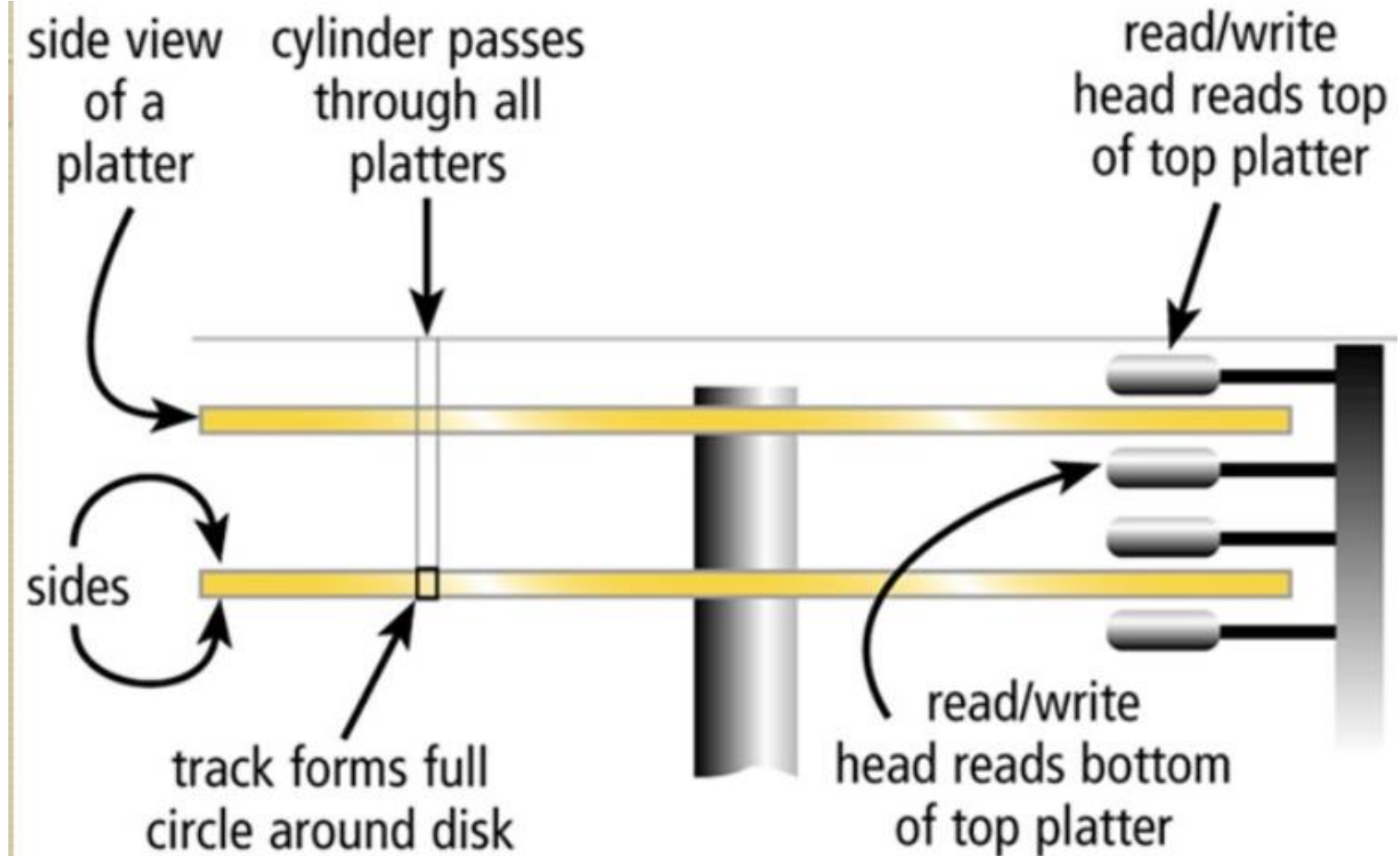


# Hard disk drive (HDD)

- Gồm nhiều đĩa tròn bằng nhôm được phủ từ tính
- Giữa ổ đĩa có động cơ quay để đọc/ghi dữ liệu, được điều khiển bởi bo mạch điện tử. Tốc độ quay 5,400 → 15,000 (vòng/phút)
- Khả năng lưu trữ 160GB → 1.5TB



# Hard disk drive (HDD)



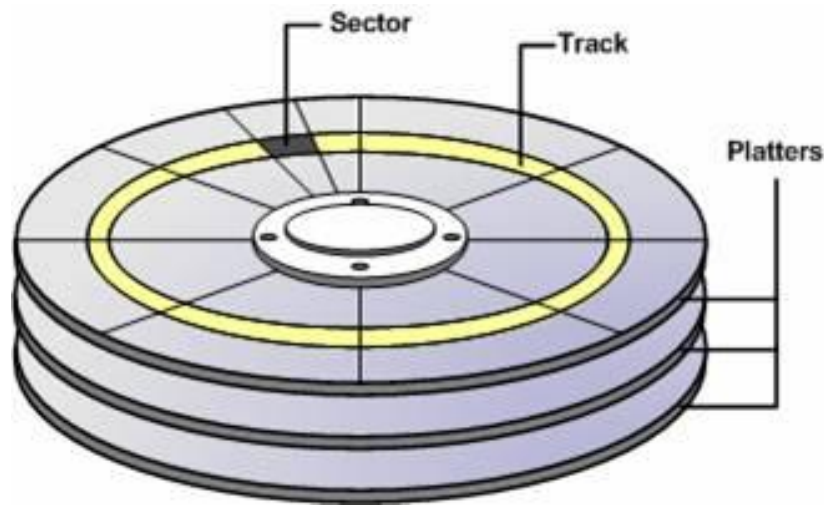
# Hard disk drive (HDD)

- **Thành phần**

- Platter đường kính ~ 3.5 inch, dùng ghi dữ liệu
- Track là rãnh hẹp chạy hết 1 vòng đĩa
- Sector là 1 phần track
- Cluster là đơn vị nhỏ nhất của đơn vị lưu trữ thông tin

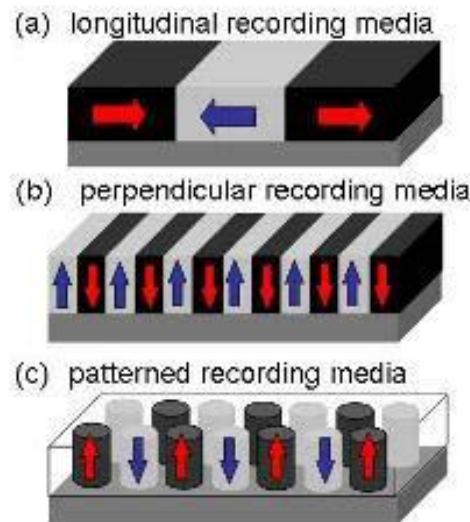
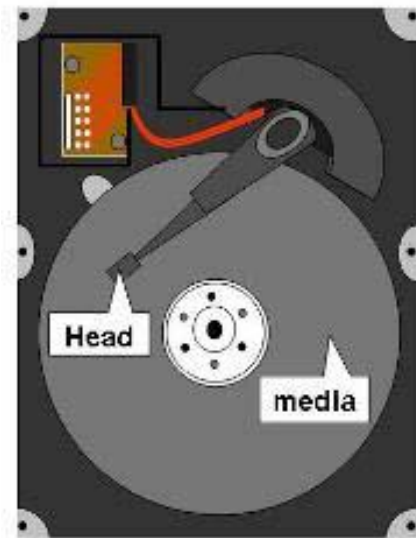
- **Định dạng**

- Là quá trình chia đĩa thành các track và sector để OS có thể lưu trữ và định vị dữ liệu và thông tin trên đĩa

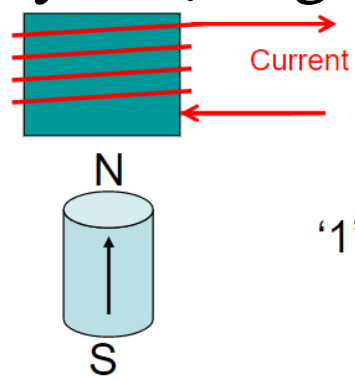
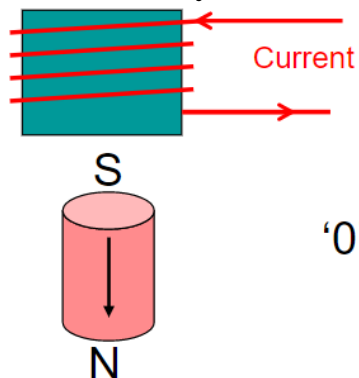


# Hard disk drive (HDD)

- Có 3 cách bố trí dữ liệu: chiều dọc, chiều ngang và mẫu

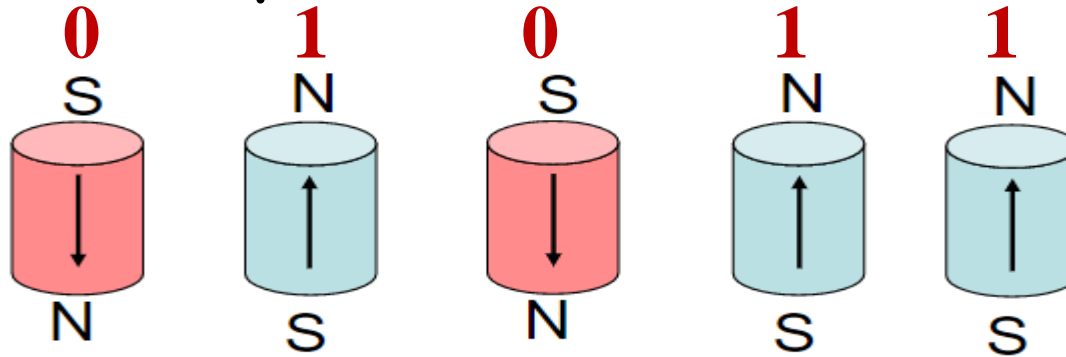


- Chiều của hạt từ tính quyết định giá trị của nó

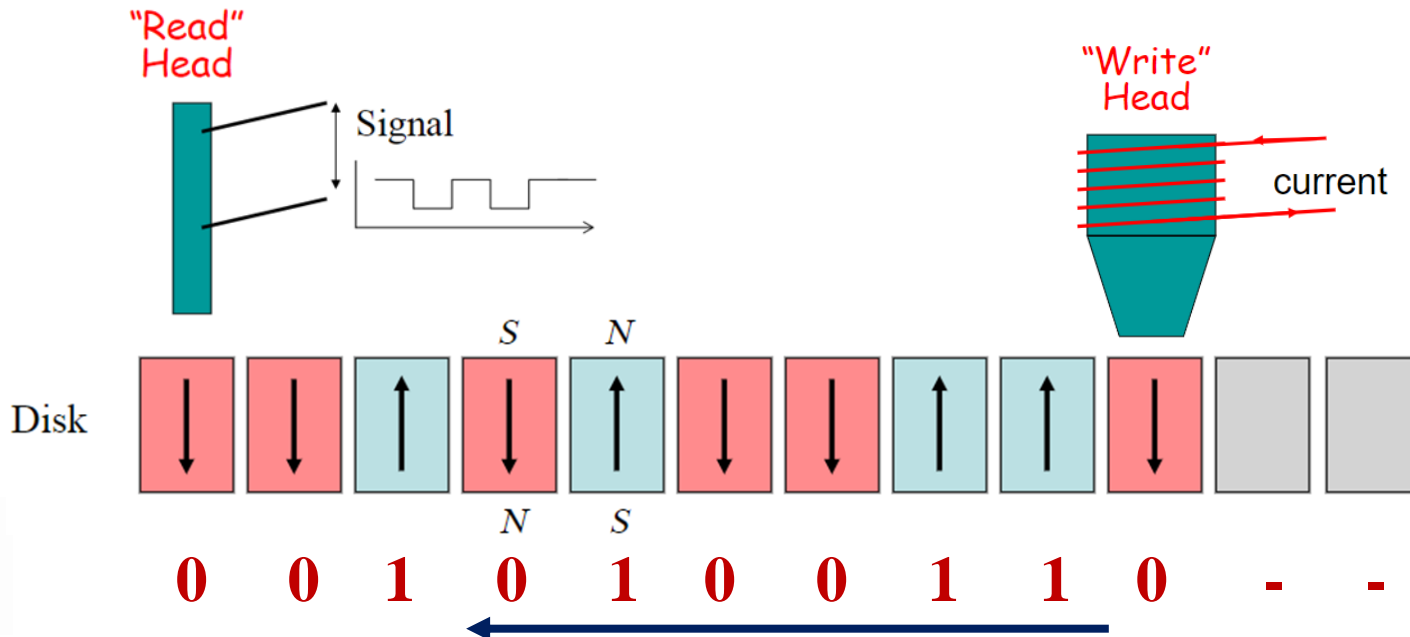


# Hard disk drive (HDD)

- VD: lưu trữ kí tự **k = 01011**



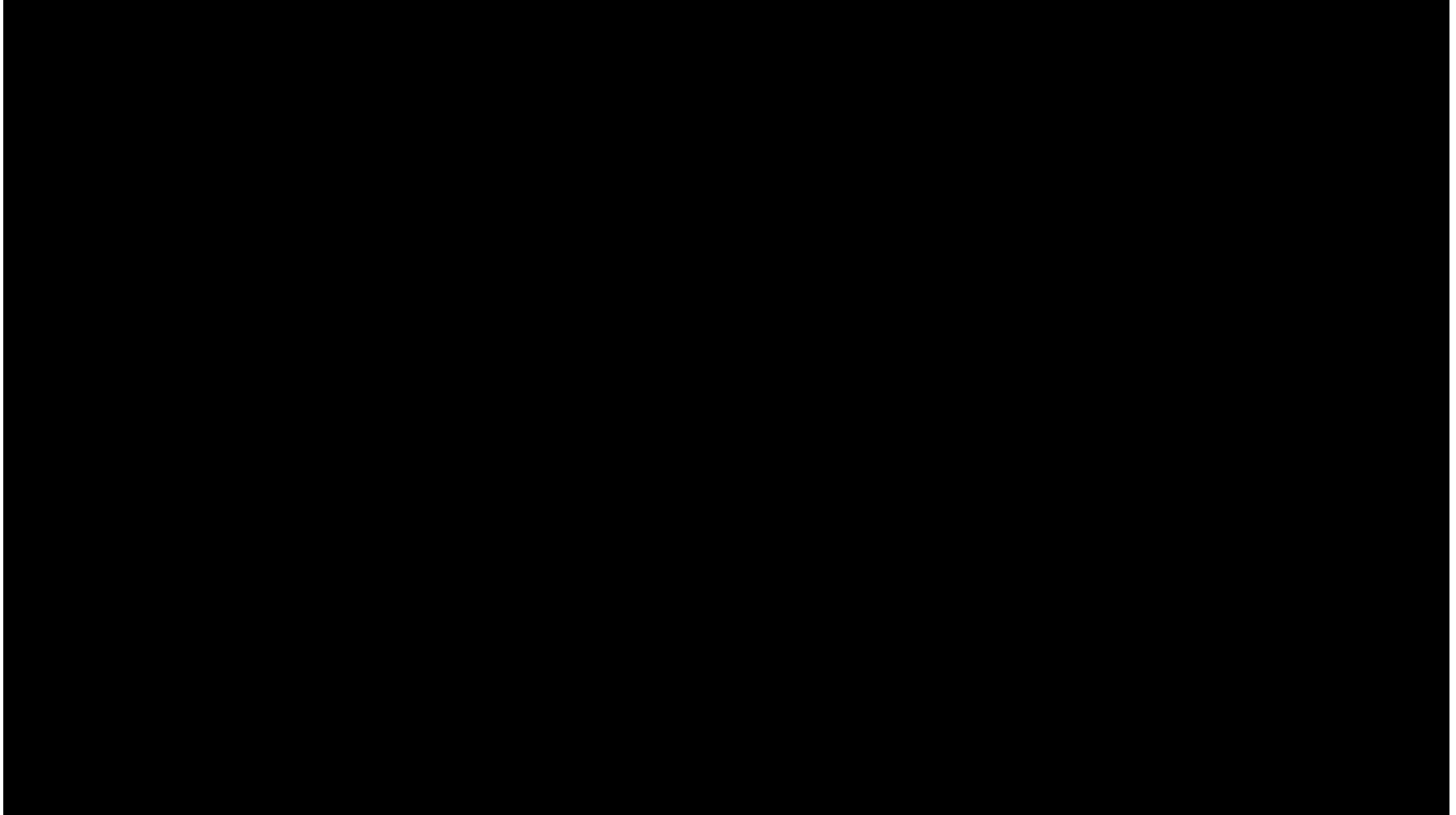
- Đọc/ ghi dữ liệu





# Flash memory

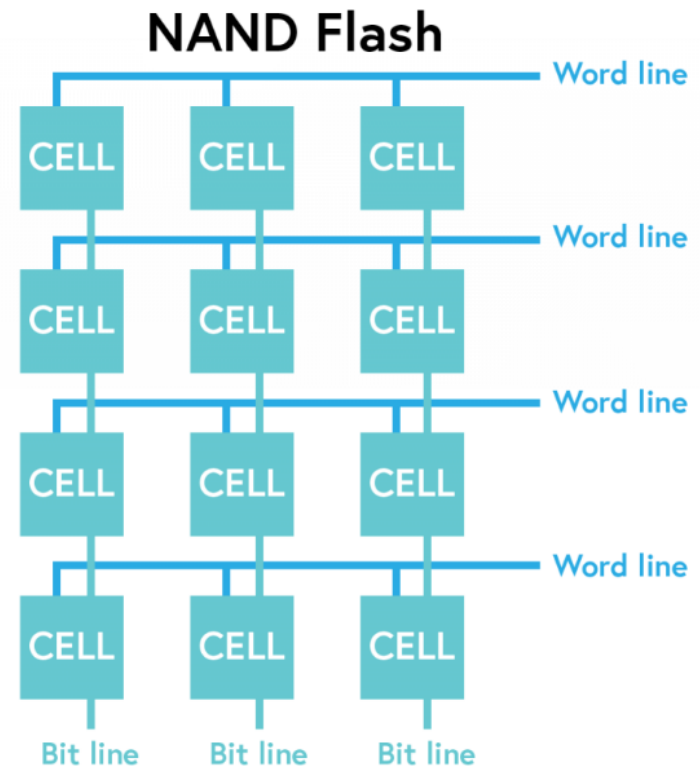
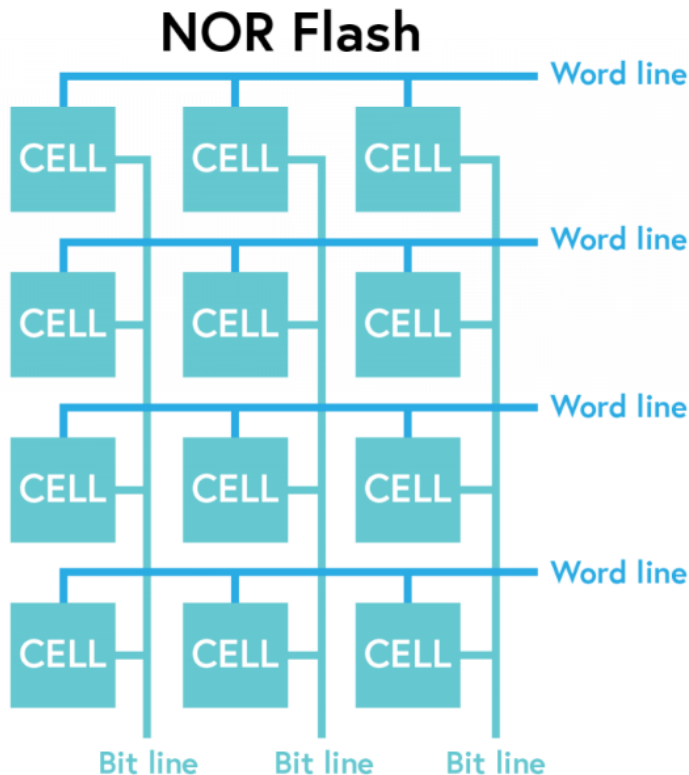
---



# NAND Flash Memory

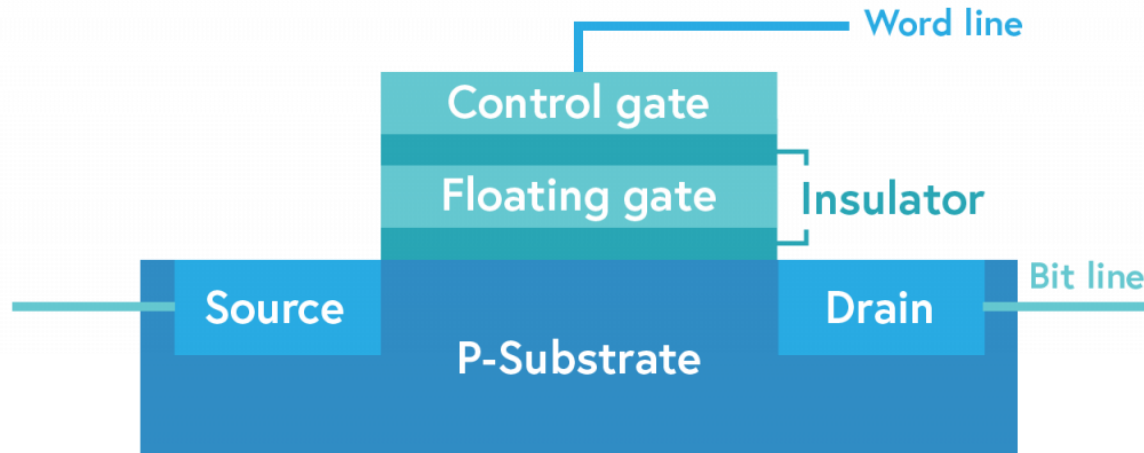
- **Flash memory**

- Là phương thức lưu trữ trạng thái rắn (Solid-state storage)
- Lưu trữ và truy xuất thông tin bằng mạch điện tử
- Thông tin được lưu trong một mảng các Memory Cell



# Floating gate transistor

## Floating Gate Transistor

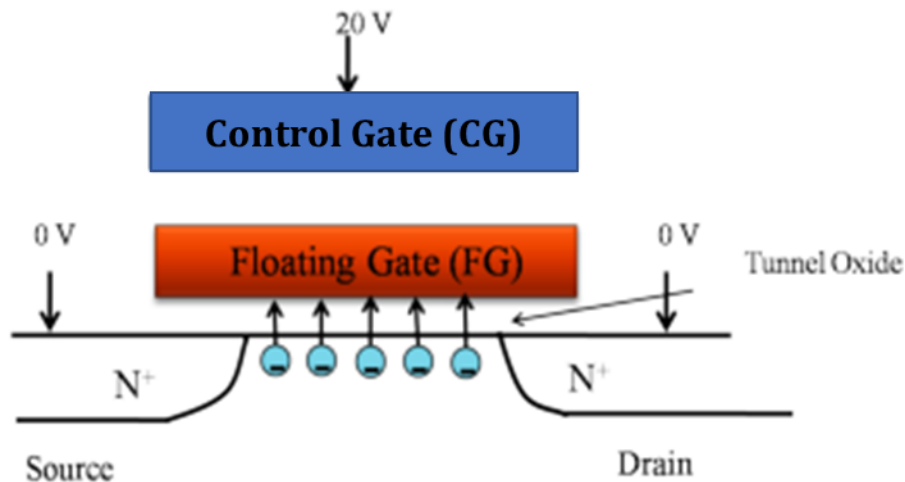


- Gồm 5 thành phần
  - Control gate – tắt mở kênh truyền
  - Floating gate – lưu trữ điện tích
  - Insulator (Oxide layer)
  - P-Substrate (Poly-Silicon)
  - Source và Drain

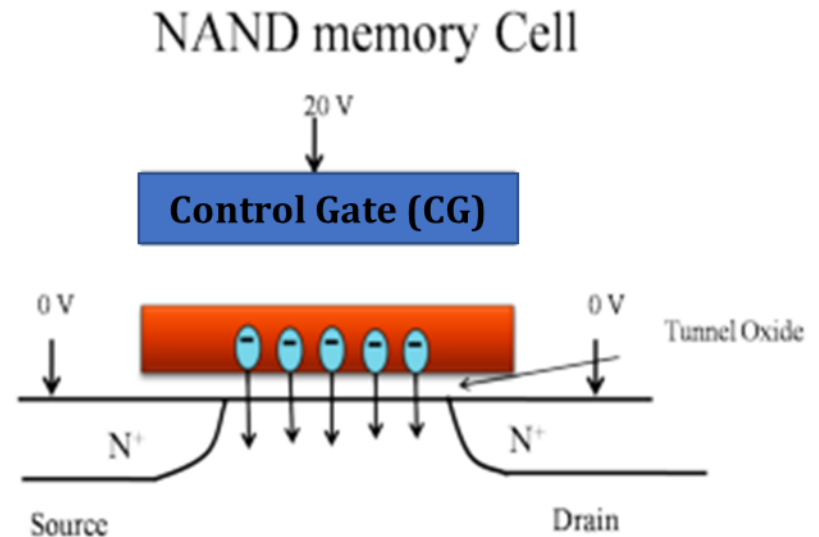
# Cơ chế hoạt động

- Control gate mở và đóng bằng cách tăng giảm điện áp.
  - Nếu cell tích điện = FG chứa electron  $\rightarrow 0$
  - Ngược lại = FG không chứa electron  $\rightarrow 1$

NAND memory Cell

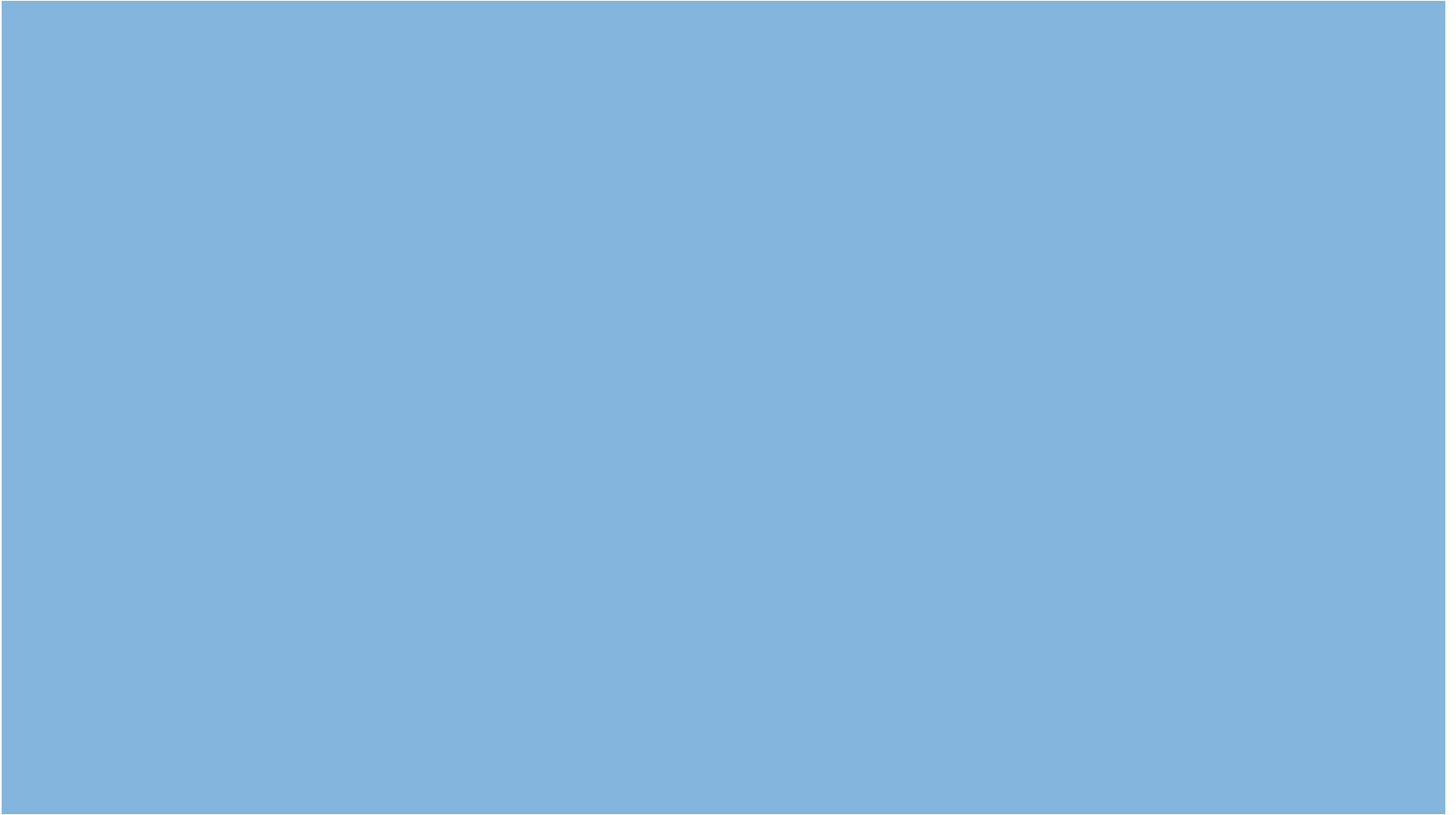


Programming



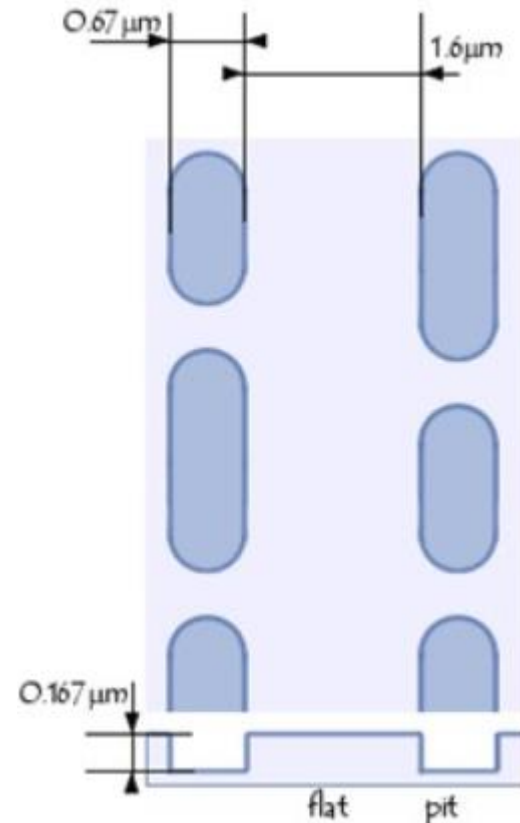
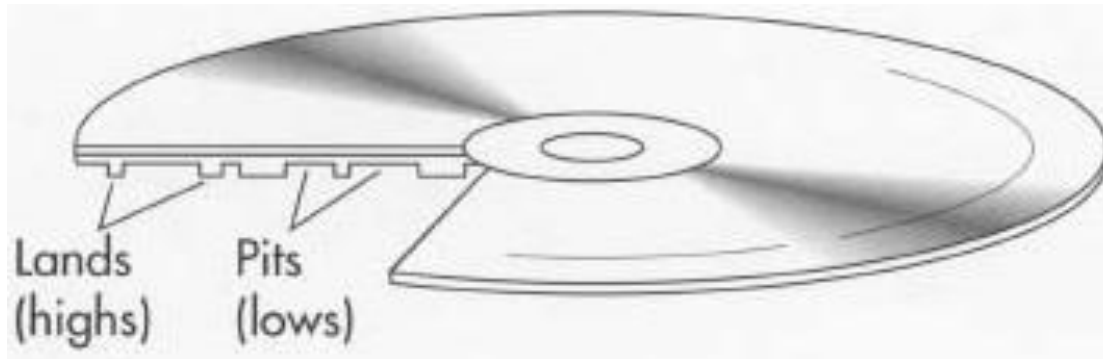
Erasing

# Optical disc



# Compact disc

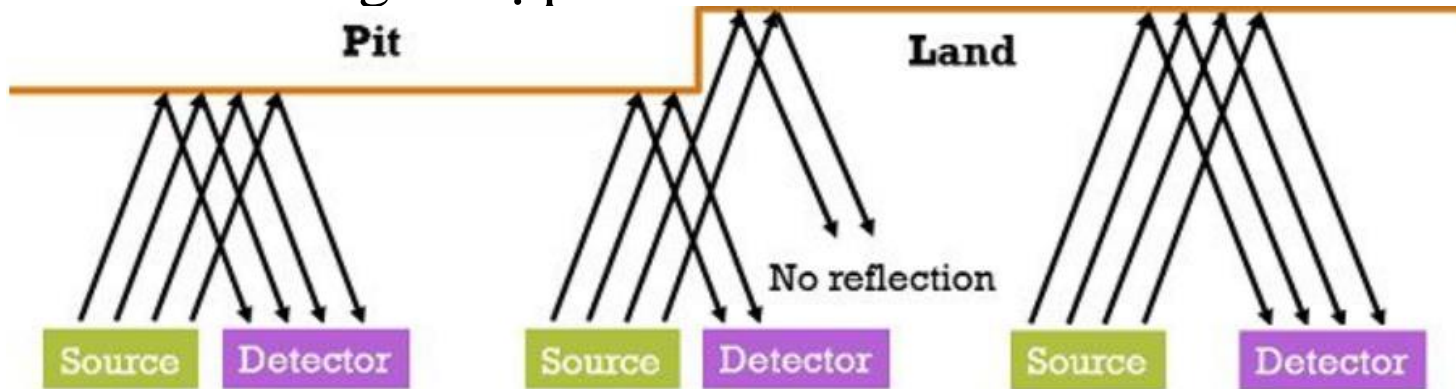
- CD là đĩa quang kích thước 12 cm, dày 1.2 mm, lưu được 650MB
- Bề mặt CD có các rãnh (**pit** = 1) và mặt phẳng (**land** = 0)



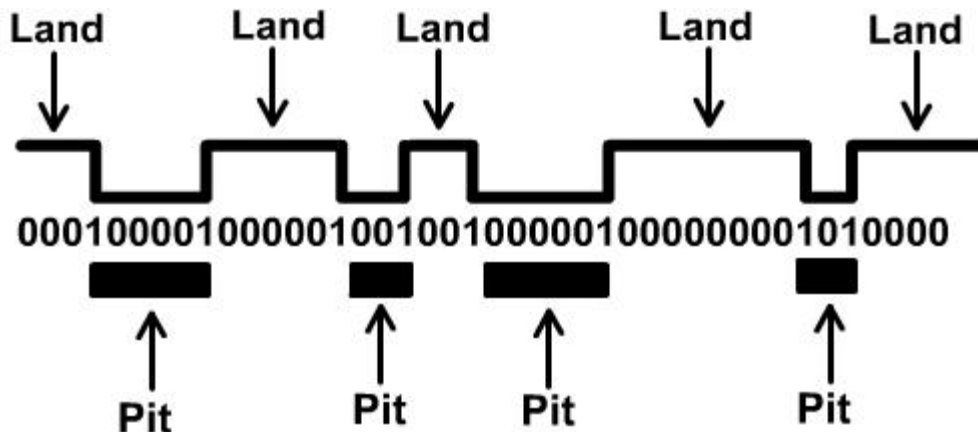
# Compact disc

- Dữ liệu được đọc bằng cách chiếu tia laser có cường độ thấp khi gặp:

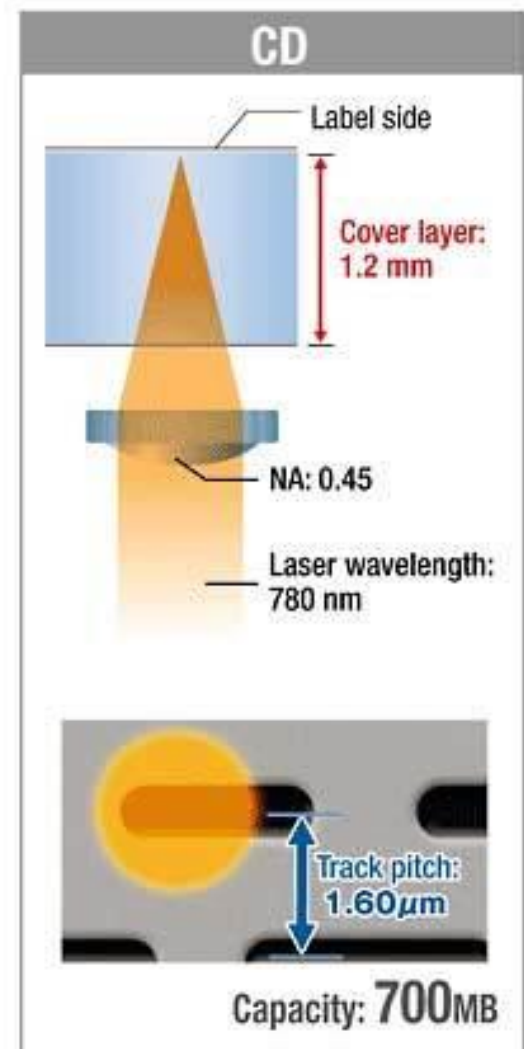
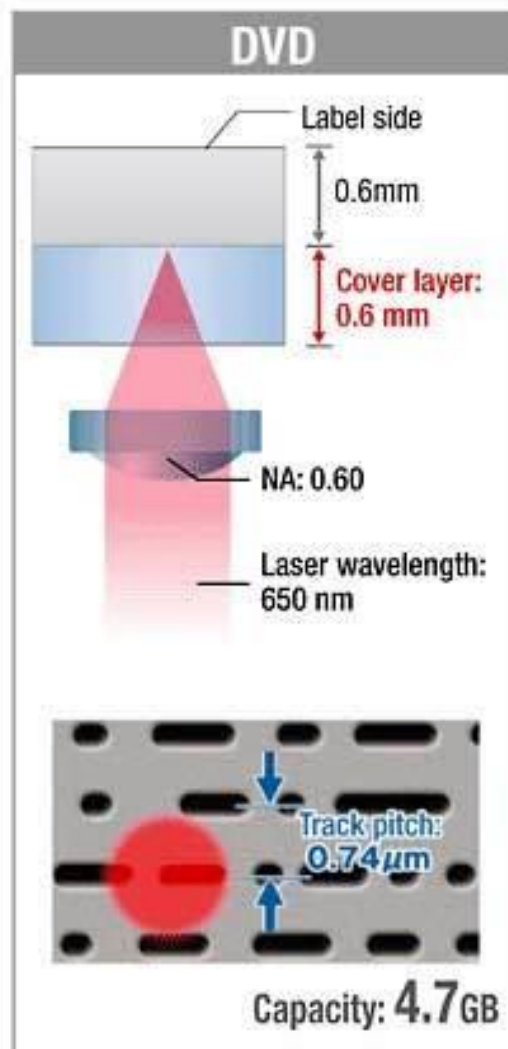
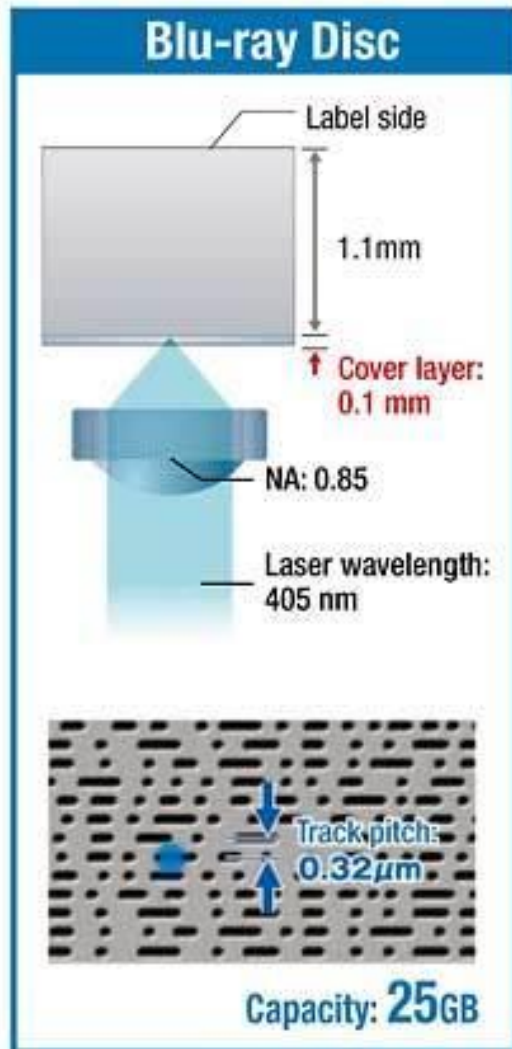
- **Land** ánh sáng sẽ bị phản xạ
- **Pit** ánh sáng sẽ bị phân tán



- VD:

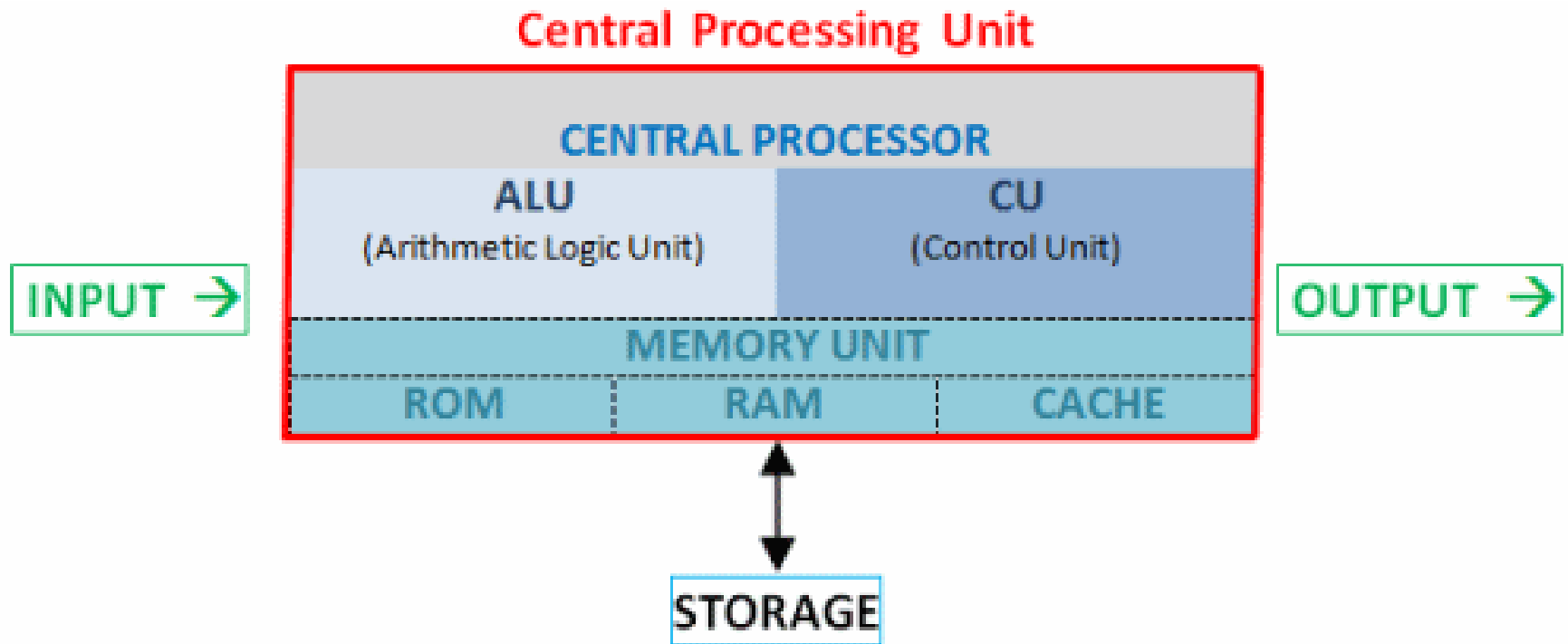


# Disc



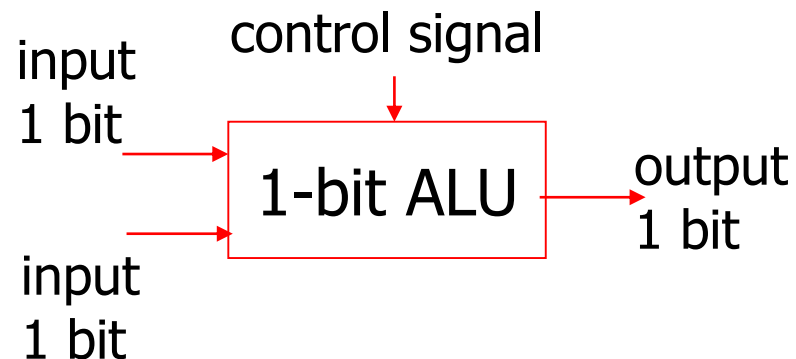


# Kiến trúc máy tính



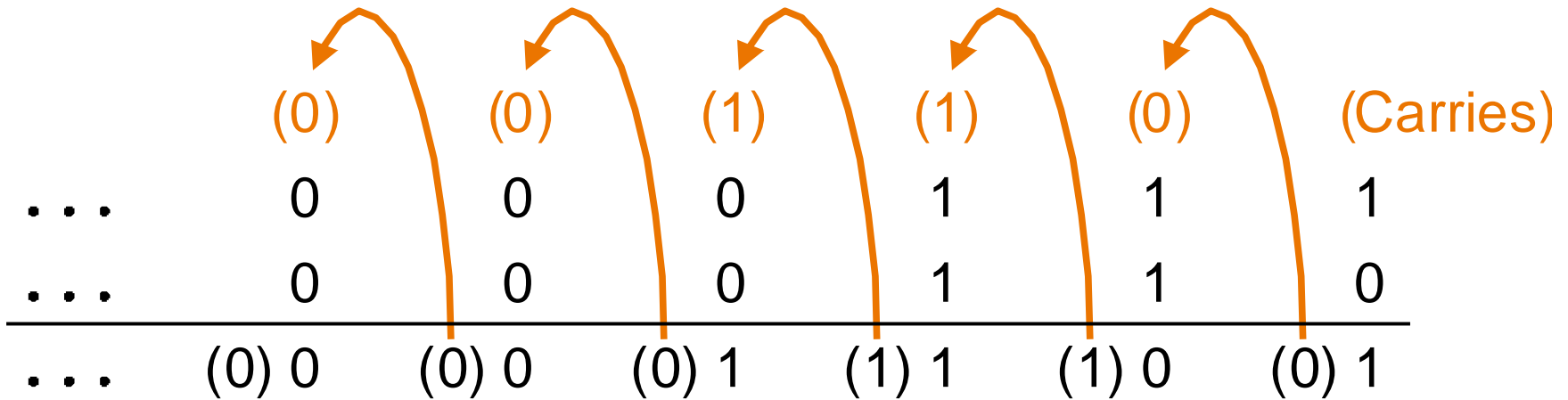
# ALU (**A**rithmetic **L**ogic **U**nit)

- ALU là sức mạnh của máy tính
- **Hàm**
  - Số học: add, sub
  - Logic: AND, OR, NOT, MUX
  - Xoay (rotate) và dịch (shift)
    - Circular shift (rotation)
    - Logical shift
    - Arithmetic shift
- **MIPS dùng 32-bit, cần ALU 32-bit**
  - 1-bit ALU
  - 32-bit ALU

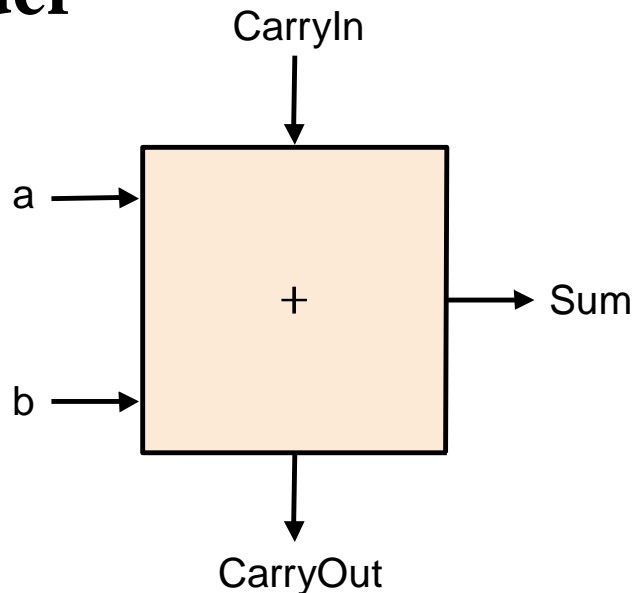


# Hàm số học

- 1-bit Addition



- Full adder



# Hàm số học

- Bảng chân trị

Input			Output		comment
a	b	CarryIn	CarryOut	Sum	
0	0	0	0	0	$0 + 0 + 0 = 00$
0	0	1	0	1	$0 + 0 + 1 = 01$
0	1	0	0	1	$0 + 1 + 0 = 01$
0	1	1	1	0	$0 + 1 + 1 = 10$
1	0	0	0	1	$1 + 0 + 0 = 01$
1	0	1	1	0	$1 + 0 + 1 = 10$
1	1	0	1	0	$1 + 1 + 0 = 10$
1	1	1	1	1	$1 + 1 + 1 = 11$

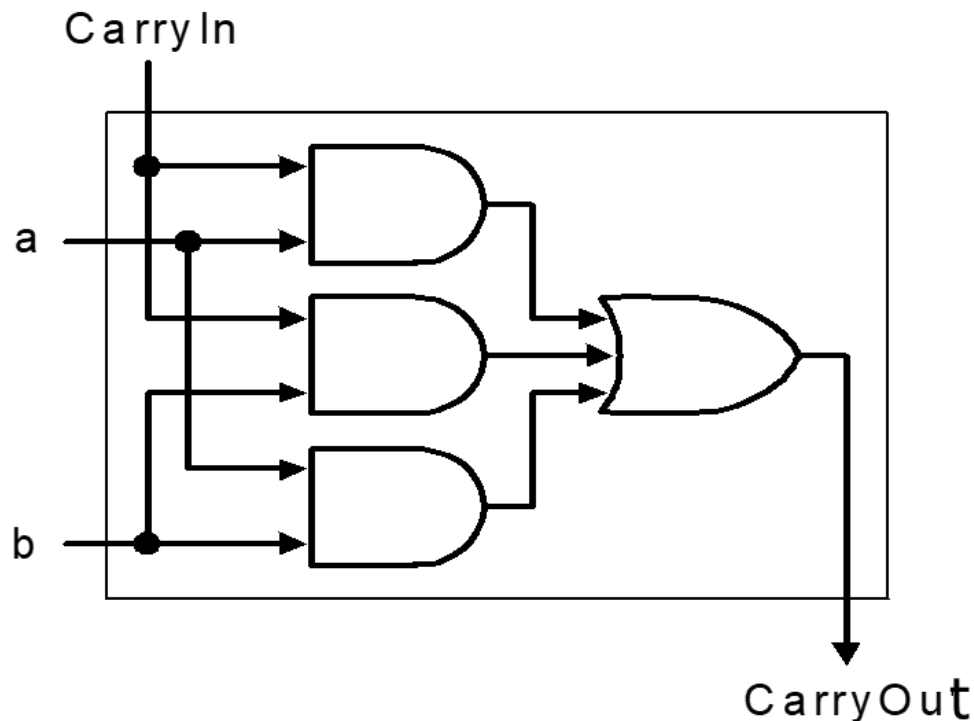
3 inputs → 2 outputs

# Hàm số học

- **Bản chân trị  $\rightarrow$  hàm logic**

$$\text{CarryOut} = (\underset{\substack{\uparrow \\ \text{AND}}}{b \bullet \text{CarryIn}}) + (\underset{\substack{\uparrow \\ \text{OR}}}{a \bullet \text{CarryIn}}) + (a \bullet b) + (\cancel{a \bullet b \bullet \text{CarryIn}})$$

- Hàm logic  $\rightarrow$  cổng logic

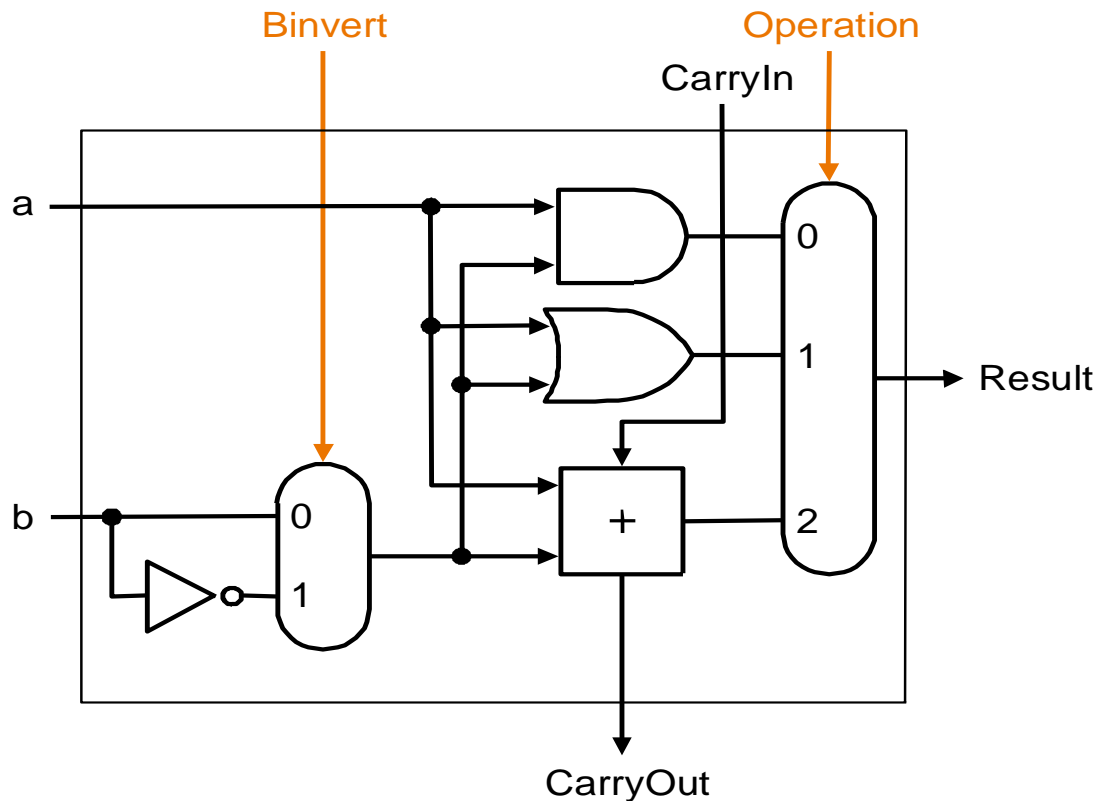


# Hàm số học

- 1-bit Sub

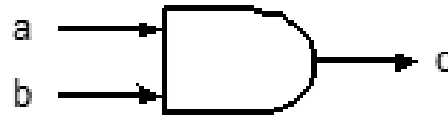
$$a - b = a + (-b) = a + \bar{b} + 1$$

complement: invert each bit of b



# Hàm logic

- AND ( $c = a \cdot b$ )



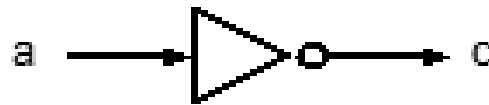
a	b	$c = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

- OR ( $c = a + b$ )



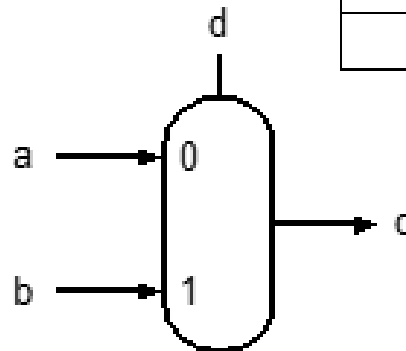
a	b	$c = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

- NOT ( $c = \bar{a}$ )



a	$c = \bar{a}$
0	1
1	0

- Multiplexor (MUX)
  - (if  $d=0$ ,  $c = a$ ; else  $c = b$ )



d	c
0	a
1	b

# Circular shift

0 1 1 0 0 1 0 1

The original bit pattern

— 0 1 1 0 0 1 0

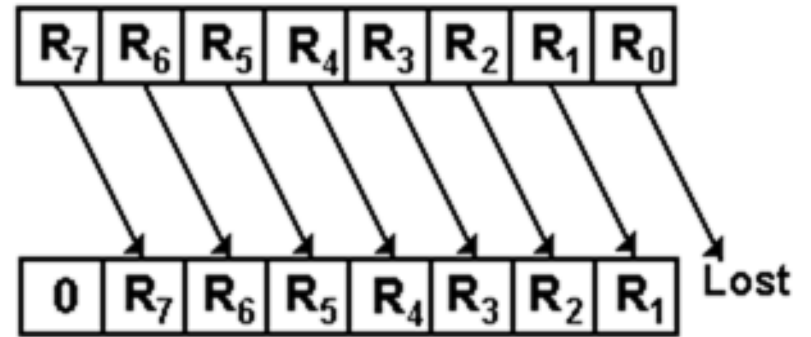
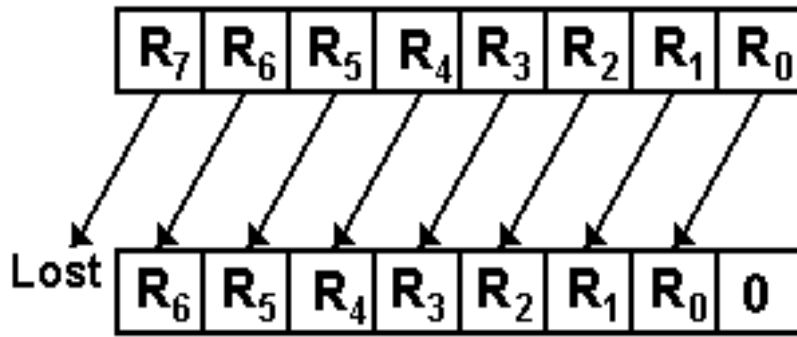
The bits move one position to the right. The rightmost bit “falls off” the end and is placed in the hole at the other end.

1 0 1 1 0 0 1 0

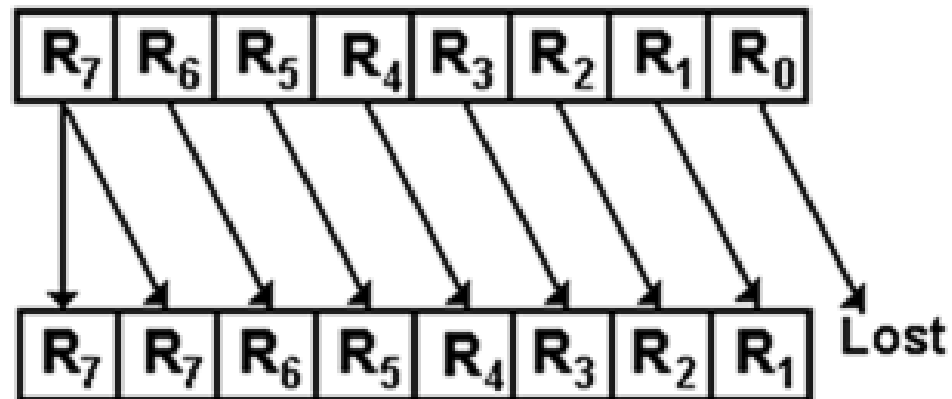
The final bit pattern



# Logical shift



# Arithmetic shift



# Bài tập

1. Một đĩa CD có dung lượng 650MB lưu trữ được 2000 trang văn bản. Vậy nếu dùng một ổ đĩa cứng có dung lượng 10GB thì lưu giữ được bao nhiêu trang văn bản?
2. Chuyển chuỗi ký tự sau thành mã nhị phân: **Khoa Công Nghệ Thông Tin**
3. Dãy bit 01001110 01101000 01100001 01110000 00100000 01001101 01101111 01101110 00100000 01000011 01001110 01010100 01010100 tương ứng là mã ASCII của dãy ký tự nào.
4. Viết các số thực sau dưới dạng dấu phẩy động: 11005 ; 25,879 ; 0,000984
5. Đổi các số sau sang hệ nhị phân và hệ cơ số 16: 7; 15; 22; 127; 97; 123.75
6. Đổi các số sau sang hệ cơ số 10: 5D1616 ; 7D71616; 11111122; 1011010122
7. Đổi từ hệ hexa sang hệ nhị phân 5E; 2A; 4B; 6C .
8. Đổi từ hệ nhị phân sang hệ hexa 1101011; 10001001; 1101001; 10110

Hết!