



Môn học: KIẾN TRÚC MÁY TÍNH

- Chương 1 • Tổng quan về máy tính
- Chương 2 • **Biểu diễn số học trong máy tính**
- Chương 3 • Hệ thống máy tính
- Chương 4 • CPU (Central Processing Unit)
- Chương 5 • Bộ nhớ máy tính (Memory)
- Chương 6 • Thiết bị giao tiếp – Thiết bị ngoại vi
- Chương 7 • Cài đặt máy tính
- Chương 8 • Sao lưu và phục hồi

Chương 2 – Biểu diễn số học máy tính

Kỹ thuật số trong máy tính

Cơ bản về các hệ đếm

Lưu trữ và truyền dữ liệu số

Tính toán số học trong máy tính

Biểu diễn số âm bằng số bù 2

Biểu diễn số dạng BCD

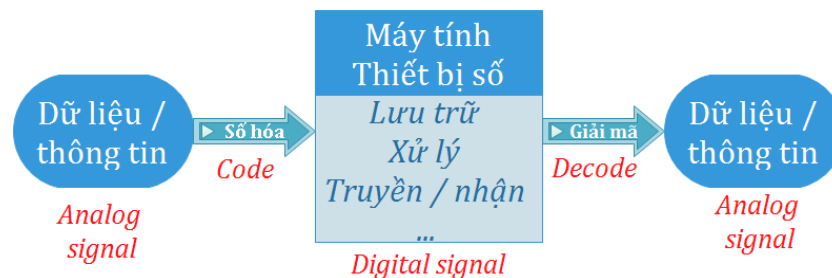
FACULTY OF INFORMATION TECHNOLOGY

- 3 -

Kỹ thuật số trong máy tính

■ Khái niệm “kỹ thuật số” (*Digital*)

- ▶ Là kỹ thuật lưu trữ / xử lý / truyền, nhận / ... dữ liệu dưới dạng chuỗi số 2 trạng thái (gọi là *bit*) **0** và **1**.
- ▶ Thông tin (dữ liệu) dạng *Analog* sẽ được “số hóa” thành bit (tín hiệu *Digital*) khi đưa vào máy tính.
- ▶ Tín hiệu *Digital* được giải mã thành *Analog* khi ra khỏi máy tính.



FACULTY OF INFORMATION TECHNOLOGY

- 4 -

Kỹ thuật số trong máy tính

■ Số hóa dữ liệu *Text*

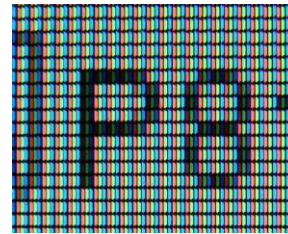
- ▶ Chuẩn **ASCII**: 1 chữ cái (*character*) → **8 bit** (~ 1 Byte)
- ▶ Chuẩn **Unicode**: 1 chữ cái (*character*) → **16 bit** (~ 2 Bytes)
- ▶ Ví dụ:
 - chữ "**A**" → 0100 0001 chữ "**B**" → 0100 0010
 - chữ "**a**" → 0110 0001 chữ "**b**" → 0110 0010
- ▶ Tham khảo bảng mã ASCII:

Decimal	Hex	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII
32	20		64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	\$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g

Kỹ thuật số trong máy tính

■ Số hóa dữ liệu *Picture*

- ▶ Ảnh được chia thành ma trận 2 chiều (**A x B**) của các điểm ảnh (gọi là **Pixel** - *picture element*).
- ▶ Mỗi *pixel* được mã hóa màu sắc bởi **n** bit.
- ▶ Ví dụ:
 Ảnh **1,200 x 1,800 pixel** (ảnh 2 *Megapixel* – 2 triệu điểm ảnh), mã hóa **24 bit** màu (tương đương 16 triệu màu)
 Ảnh này được số hóa bởi: **1,200 x 1,800 x 24 bit = 51,840,000 bits**
 (≈ **6,480,000 Bytes** ≈ **6,4 MegaBytes**)

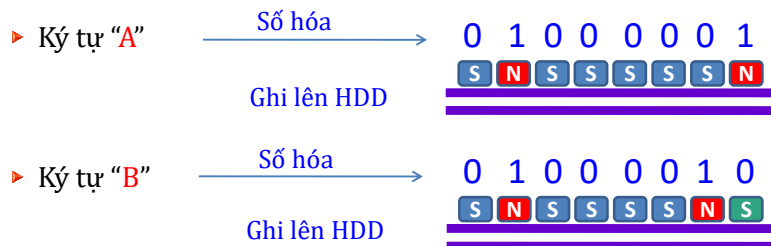


Lưu trữ và truyền dữ liệu số

■ Lưu trữ dữ liệu số:

- Phương thức ghi bit (0 và 1) trên các thiết bị lưu trữ:

Mã hóa	HDD	CD-ROM	RAM
Bit "0"	Cực S của hạt từ	Lỗ thủng	0.3V - 0.7V
Bit "1"	Cực N của hạt từ	Phản quang	3.5V - 5.0V

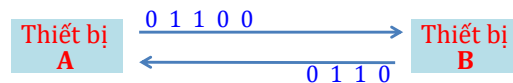


FACULTY OF INFORMATION TECHNOLOGY

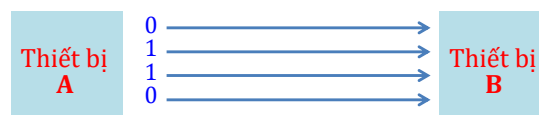
Lưu trữ và truyền dữ liệu số

■ Truyền dữ liệu số:

- Kiểu truyền nối tiếp (đường truyền *Serial bus*):
- Các bit 0, 1 nối tiếp nhau truyền trên dây dẫn từ A sang B.
 - Các bit truyền từ B sang A trên dây dẫn khác



- Truyền song song (đường truyền *Parallel bus*)
- Nhiều bit cần truyền được đặt cùng lúc trên nhiều đường dây nối.
 - Các bit sẽ truyền đồng loạt từ A sang B hoặc ngược lại.
 - Đường bus **n** bit là đường truyền có **n** dây nối (**n** = 4/8/16/32/64...)



FACULTY OF INFORMATION TECHNOLOGY

Lưu trữ và truyền dữ liệu số

■ Tần số truyền:

- ▶ Tần số (*Frequency*) là số lần truyền dữ liệu trong 1 giây (*second*)
- ▶ Đơn vị tính: **Hz** (*Hertz*) hoặc **KHz**, **MHz**, **GHz** ...

■ Băng thông đường truyền:

- ▶ Băng thông (*Bandwidth*) là chỉ số cho biết số lượng *bits* (hoặc *bytes*) truyền được trong thời gian 1 giây (*second*)
- ▶ Đơn vị tính:
bps (*Bits per second*) hoặc **Kbps**, **Mbps**, **Gbps** ...
 hoặc **B/s** (*Bytes/second*) hoặc **KB/s**, **MB/s**, **GB/s** ...
- ▶ Công thức tính BW của *Parallel bus*:

$$\text{Băng thông} = \text{độ rộng bus} \times \text{tần số truyền} / 8 \text{ (Bytes/s)}$$

- ▶ Công thức tính BW của *Serial bus*:

$$\text{Băng thông} = \text{tần số truyền} / 10 \text{ (Bytes/s)}$$

FACULTY OF INFORMATION TECHNOLOGY

Cơ bản về các hệ đếm

■ Hệ đếm:

- ▶ Là phương pháp dùng các ký hiệu để biểu diễn dãy số tự nhiên.
- ▶ Tên hệ đếm cũng là số lượng ký hiệu sử dụng.

■ Các hệ đếm thông dụng:

Decimal	Binary	Hexa
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7

Decimal	Binary	Hexa
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	10

FACULTY OF INFORMATION TECHNOLOGY

- 10 -

Cơ bản về các hệ đếm

■ Các hệ đếm thông dụng:

► Tính chất:

	Decimal	Binary	Hexa
Cơ số (b)	10	2	16
Khi dùng 2 ký số để biểu diễn một số nguyên, giá trị <i>min - max</i> là:	00 - 99 (0 → 99)	00 - 11 (0 → 3)	00 - FF (0 → 255)
Khi dùng (n) ký số để biểu diễn một số nguyên, giá trị <i>min - max</i> là:	0 - bⁿ 0 - 10ⁿ	0 - bⁿ 0 - 2ⁿ	0 - bⁿ 0 - 16ⁿ
Biểu diễn hệ đếm của một số: hoặc:	101 _D 101 ₁₀	101 _B 101 ₂	101 _H 101 ₁₆

Cơ bản về các hệ đếm

■ Tính giá trị cho một số:

► Một số **A** có **n** ký số (a), biểu diễn theo cơ số **b**:

$$a_{n-1}a_{n-2}a_{n-3}\dots a_1a_0$$

► Ví dụ: **1028**₁₀ (số **8** ở vị trí **i=0**, số **2** vị trí **i=1**,...)

► Giá trị **A** của số được tính theo công thức:

$$A = \sum_{i=0}^{n-1} a_i \cdot b^i$$

$$\text{► } A = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0$$

■ Ví dụ: tính giá trị của 1 số biểu diễn theo hệ đếm:

$$\begin{aligned}
 1208_{10} &= 1 \cdot 10^3 + 2 \cdot 10^2 + 0 \cdot 10^1 + 8 \cdot 10^0 \\
 &= 1 \cdot 1000 + 2 \cdot 100 + 0 \cdot 10 + 8 \cdot 1 \\
 &= 1000 + 200 + 0 + 8 = 1208_{10}
 \end{aligned}$$

Cơ bản về các hệ đếm

■ Tính giá trị cho một số:

► Ví dụ: tính giá trị của 1 số biểu diễn theo hệ đếm:

$$\begin{aligned} 10110_2 &= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 \\ &= 16 + 0 + 4 + 2 + 0 = 22_{10} \end{aligned}$$

$$\begin{aligned} 4B8_{16} &= 4 \cdot 16^2 + B \cdot 16^1 + 8 \cdot 16^0 \\ &= 4 \cdot 256 + 11 \cdot 16 + 8 \cdot 1 \\ &= 1024 + 176 + 8 = 1208_{10} \end{aligned}$$

$$\begin{aligned} 1100\ 1001_2 &= 2^7 + 2^6 + 2^3 + 2^0 \\ &= 128 + 64 + 8 + 1 = 201_{10} \end{aligned}$$

Cơ bản về các hệ đếm

■ Chuyển đổi hệ Thập phân → hệ Nhị phân:

► Ví dụ: Chuyển số 118_{10} sang hệ Nhị phân:

• Dùng phương pháp: *chia 2 - lấy dư số*.

TT	Phép tính	Số dư	TT	Phép tính	Số dư
1	$118 \div 2 = 59$	0	5	$7 \div 2 = 3$	1
2	$59 \div 2 = 29$	1	6	$3 \div 2 = 1$	1
3	$29 \div 2 = 14$	1	7	$1 \div 2 = 0$	1
4	$14 \div 2 = 7$	0	8	$0 \div 2 = 0$	0

► Lược trình các con số dư theo thứ tự từ dưới lên, ta có số nhị phân $0111\ 0110_2$.

Cơ bản về các hệ đếm

■ Chuyển đổi hệ Thập phân → hệ Nhị phân:

▶ Một cách chuyển đổi khác:

- Phân chia số hệ 10 thành tổng của các số 2^n :
- Số $118_{10} = 64 + 32 + 16 + 4 + 2$ (tương đương: $2^6 + 2^5 + 2^4 + 2^2 + 2^1$)
- Viết thành số Nhị phân: với vị trí thứ i , nếu có thì = 1, nếu không có thì = 0)

$$118_{10} = 0111\ 0110_2$$

- ▶ Ví dụ: số $131_{10} = 128 + 2 + 1$ (tương đương: $2^7 + 2^1 + 2^0$)
 $= 1000\ 0011_2$

- ▶ Ví dụ: số $215_{10} = 128 + 64 + 16 + 4 + 2 + 1$
 (tương đương: $2^7 + 2^6 + 2^4 + 2^2 + 2^1 + 2^0$)
 $215_{10} = 1101\ 0111_2$

Tính toán số học trong máy tính

■ Tính toán số học trong máy tính.

- ▶ Máy tính thực hiện tính toán trên số nhị phân (bit).
- ▶ Tốc độ tính toán phụ thuộc vào tốc độ của CPU (bộ xử lý) và chiều dài của thanh ghi (Register)

■ Cơ bản về thanh ghi (Register) trong CPU:

- ▶ Vai trò của thanh ghi:
 - CPU nạp thông tin từ Bộ nhớ vào các Thanh ghi.
 - CPU tiến hành xử lý thông tin trên các thanh ghi.
 - Ghi kết quả xử lý vào thanh ghi khác.
 - CPU xuất kết quả từ thanh ghi vào Bộ nhớ.
- ▶ Độ dài của thanh ghi:
 - Tùy công nghệ chế tạo CPU => thanh ghi 8 bit / 16 bit / 32 bit / 64 bit.
 - Thanh ghi tiêu chuẩn: 8 bit.

Tính toán số học trong máy tính

■ Biểu diễn số học trên thanh ghi tiêu chuẩn 8 bit.

- ▶ Nếu chỉ lưu trữ số dương: **0000 0000** → **1111 1111** (từ **0** đến **255**)
- ▶ Nếu dùng cho cả số âm và dương:
 - Bit cực trái là **"0"**: 7 bit còn lại biểu diễn giá trị **dương**
 - Bit cực trái là **"1"**: 7 bit còn lại biểu diễn giá trị **âm**
- ▶ Giá trị số biểu diễn bởi thanh ghi 8 bit: từ **-127** đến **+127** (chỉ còn 7 bit biểu diễn số)
- ▶ Ví dụ

$0001\ 0011_2 \rightarrow +19_{10}$
 $1001\ 0001_2 \rightarrow -17_{10}$

Tính toán số học trong máy tính

■ Phép cộng nhị phân:

▶ Nguyên tắc:

$$\begin{aligned}
 0 + 0 &= 0 \\
 0 + 1 &= 1 \\
 1 + 0 &= 1 \\
 1 + 1 &= 0 \text{ (nhớ 1 lên hàng thứ 2)}
 \end{aligned}$$

Ví dụ:

$$\begin{array}{r}
 \begin{array}{cccccc}
 & 1 & 1 & 1 & 1 & 1 & \text{(nhớ)} \\
 & 0 & 1 & 1 & 0 & 1 & \\
 + & 1 & 0 & 1 & 1 & 1 & \\
 \hline
 = & 1 & 0 & 0 & 1 & 0 & 0
 \end{array}
 \end{array}$$

■ Phép trừ nhị phân:

▶ Nguyên tắc:

$$\begin{aligned}
 0 - 0 &= 0 \\
 0 - 1 &= 1 \text{ (mượn 1 ở bit tiếp theo)} \\
 1 - 0 &= 1 \\
 1 - 1 &= 0
 \end{aligned}$$

Ví dụ:

$$\begin{array}{r}
 \begin{array}{ccccccc}
 & & & * & * & * & * \\
 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
 - & & & 1 & 0 & 1 & 1 & 1 \\
 \hline
 = & 1 & 0 & 1 & 0 & 1 & 1 & 1
 \end{array}
 \end{array}$$

Tính toán số học trong máy tính

■ Các phép dịch bit trong thanh ghi 8 bits:

- ▶ Phép dịch trái - *Bits shift left*
 - Ký hiệu: **<<**
 - Mã lệnh: **SHL**
- ▶ Phép dịch phải - *Bits shift right*
 - Ký hiệu: **>>**
 - Mã lệnh: **SHR**

(chỉ số của bit)	7	6	5	4	3	2	1	0	
A	0	0	0	1	1	0	0	0	= 24
A >> 1	0	0	0	0	1	1	0	0	= 12
A >> 2	0	0	0	0	0	1	1	0	= 6
A << 3	1	1	0	0	0	0	0	0	= 192
A >> 4	0	0	0	0	0	0	0	1	= 1

Tính toán số học trong máy tính

■ Các phép toán logic:

- ▶ Phép **AND** (Ký hiệu: **&**)
 - **A&B** chỉ bằng 1 khi cả A và B cùng là 1
- ▶ Phép **OR** (Ký hiệu: **|**)
 - **A|B** chỉ bằng 0 khi cả A và B cùng là 0
- ▶ Phép **XOR** (Ký hiệu: **^**)
 - Giống nhau thì = 0
 - Khác nhau thì = 1

A	0	0	1	1
B	0	1	0	1
A&B	0	0	0	1

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

A	0	0	1	1
B	0	1	0	1
A^B	0	1	1	0

- ▶ Phép **NOT** (Ký hiệu: **!** Hoặc: **~**): là phép đảo bit.

Biểu diễn số âm bằng số bù 2

■ Lỗi tính toán khi có phần tử là số âm.

- ▶ Xét ví dụ: $+19_{10} - 17_{10}$
 $\approx (0001\ 0011_2 + 1001\ 0001_2)$
- ▶ Tính:

0001 0011 ₂	+19 ₁₀
1001 0001	-17
1010 0100	-36

Biểu diễn số âm bằng số bù 2

■ Quy tắc tính bù 2 cho số nhị phân:

- ▶ Bước 1: đảo tất cả các bit của số nhị phân (gọi là phép tính bù 1)
- ▶ Bước 2: cộng thêm 1 vào kết quả của bước 1.
- ▶ Ví dụ: bù 2 của số **0001 0010₂**

Số nhị phân:	0001 0010 ₂
Bước 1 (bù 1):	1110 1101
Bước 2 (bù 2):	1110 1110

■ Cách tính khác:

- ▶ Nhìn từ bên phải sang: giữ nguyên các bit cho đến khi gặp bit **1** đầu tiên.
- ▶ Đảo tất cả các bit còn lại phía sau bit 1 đó.
- ▶ Ví dụ: bù 2 của số **0001 0010₂**
 - Nhìn từ bên phải sang, giữ nguyên 2 bit màu đỏ: **0001 0010**
 - Đảo tất cả các bit còn lại, ta có: **1110 1110**

Biểu diễn số âm bằng số bù 2

■ Số bù 2 sửa lỗi tính toán khi có phần tử là số âm.

- ▶ Xét ví dụ: $+19_{10} - 17_{10}$
(tương đương $0001\ 0011_2 + 1001\ 0001_2$)
- ▶ Số -17_{10} lưu trữ dạng bù 2 là: **1110 1111**)
- ▶ Tính:

$$\begin{array}{r|l}
 0001\ 0011 & +19_{10} \\
 1110\ 1111 & -17 \\
 \hline
 1\ 0000\ 0010 & +2
 \end{array}$$

(bit 1 ở bìa trái bị tràn ra khỏi thanh ghi 8 bit => bỏ bit này)

Biểu diễn số dạng BCD

■ Số BCD.

- ▶ Biểu diễn các số hệ *Decimal* dưới dạng *Binary* (BCD – *Binary Coded Decimal*)
- ▶ Mỗi số BCD dùng 4 bit để biểu diễn 10 ký số (từ 0 đến 9) của hệ Dec.
- ▶ Ví dụ:
 - Biểu diễn số **83** dạng BCD là: **1000 0011**
(số $1000_2 = 8_{10}$, số $0011_2 = 3_{10}$)
 - Biểu diễn số **567** dạng BCD là: **0101 0110 0111**

Thảo Luận

