Kiến trúc máy tính
(Computer Architecture)

Chương 1
Đại cương

CANTHO UNIVERSITY

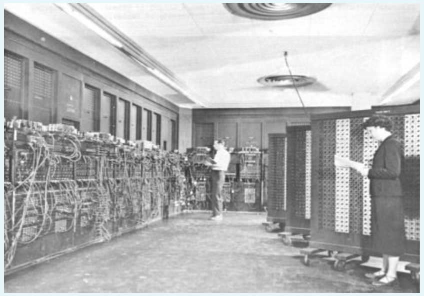
Mục đích, yêu cầu

- **Mục đích:** Giới thiệu lịch sử phát triển, phân loại, thành quả của máy tính điện tử, xu hướng phát triển, các khái niệm cơ bản về thông tin, các phương pháp mã hoá thông tin trong máy tính điện tử.
- **Yêu cầu:** Sinh viên nắm được cách phân chia thế hệ và xu hướng phát triển của máy tính điện tử; Phương pháp phân loại và đánh giá thành quả phát triển của máy tính điện tử; Các khái niệm cơ bản liên quan đến thông tin và phương pháp biến đổi giữa các hệ thống số được dùng trong máy tính điện tử.

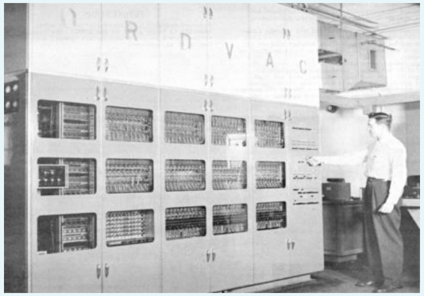
Nội dung

1. Các thế hệ máy tính
2. Phân loại máy tính
3. Thành quả của máy tính
4. Thông tin và sự mã hóa thông tin

1. Các thế hệ máy tính



Thế hệ thứ I
(1946 - 1957)



Thế hệ thứ II
(1958 - 1964)

Thế hệ thứ III
(1965 - 1971)

Thế hệ thứ IV
(1972 - ...)

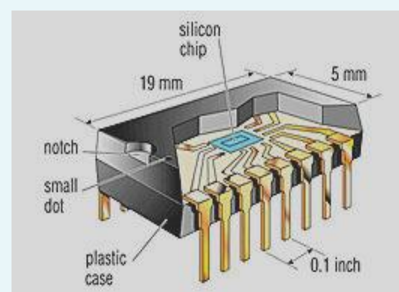
Các thế hệ máy tính

Tiêu chuẩn phân chia các thế hệ máy tính

Sự tiến bộ của công nghệ chế tạo các linh kiện cơ bản của máy tính (bộ xử lý trung tâm và bộ nhớ trong).

Thế hệ thứ I
Đèn điện tử
(1946 - 1957)

Thế hệ thứ II
Transistor
(1958 - 1964)

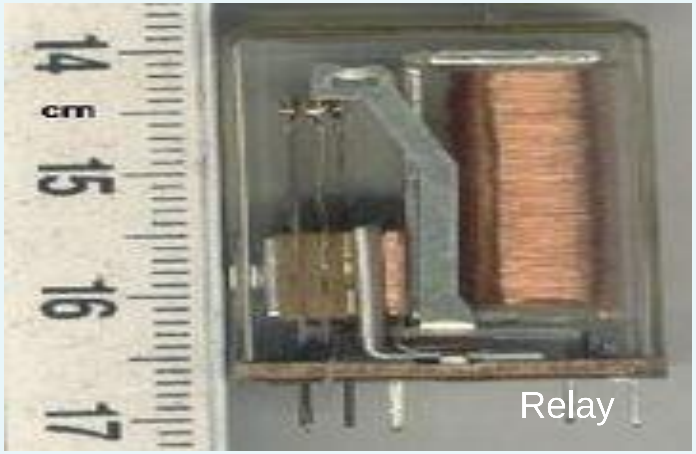


Thế hệ thứ III
IC: Integrated circuit
(1965 - 1971)

Thế hệ thứ IV
IC: Integrated circuit
(1972 - …)

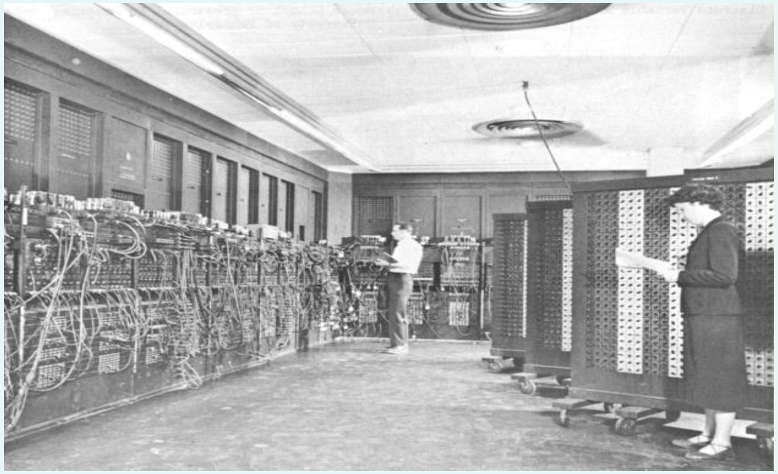
Thế hệ đầu tiên (1946-1957)

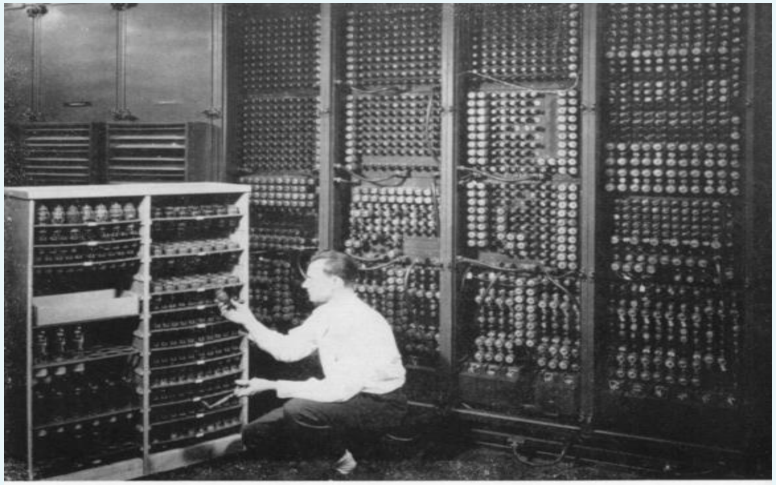
- **Công nghệ chế tạo:**
  + Đèn điện tử
  + Bộ nhớ dùng Rờ le
- **Phần mềm:**
  + Lập trình bằng tay



Thế hệ đầu tiên (1946-1957)

- ENIAC (Electronic Numerical Integrator and Computer) là máy tính điện tử số đầu tiên do Giáo sư John Mauchly (Đại học Pennsylvania) thiết kế 1943 và hoàn thành 1946.
- ENIAC bao gồm: 18.000 đèn điện tử, 1.500 rờ le, cân nặng 30 tấn, và tiêu thụ 140KW giờ.
- Kích thước: dài 20 mét, cao 2,8 mét và rộng vài mét.
- Có 20 thanh ghi 10 bit (tính toán trên số thập phân).
- Có khả năng thực hiện 5.000 phép toán cộng trong một giây.





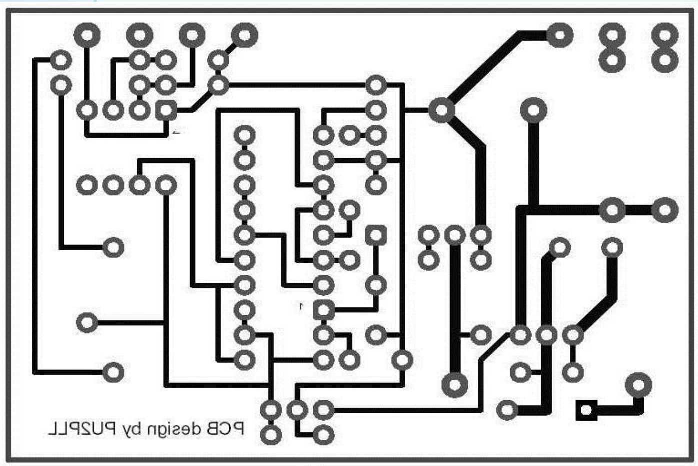
Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

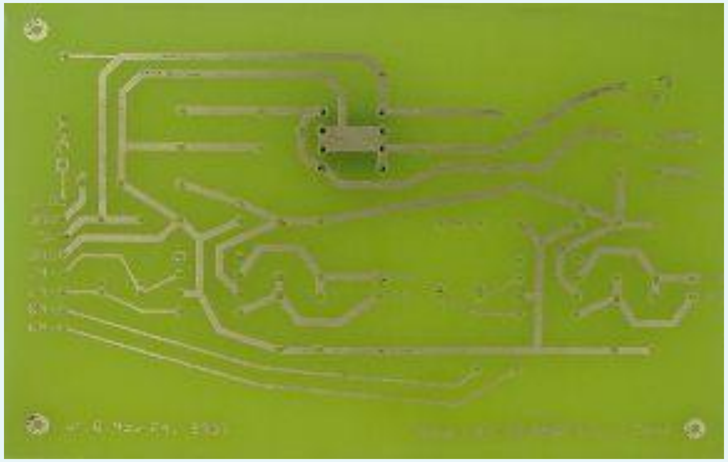
Thế hệ thứ hai (1958-1964)

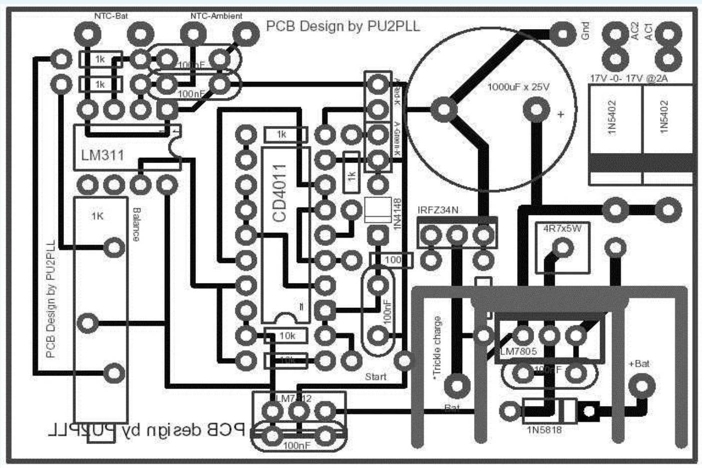
- **Công nghệ chế tạo:**
  - Transistor lưỡng cực (Bipolar transistor).
  - Mạch in (PCB: Printed Circuit Board)
  - Bộ nhớ xuyến từ.
- **Phần mềm:**
  - Ngôn ngữ cấp cao
    + FORTRAN (1956)
    + COBOL (1959)
    + ALGOL (1960)
  - Hệ điều hành kiểu tuần tự (Batch Processing).

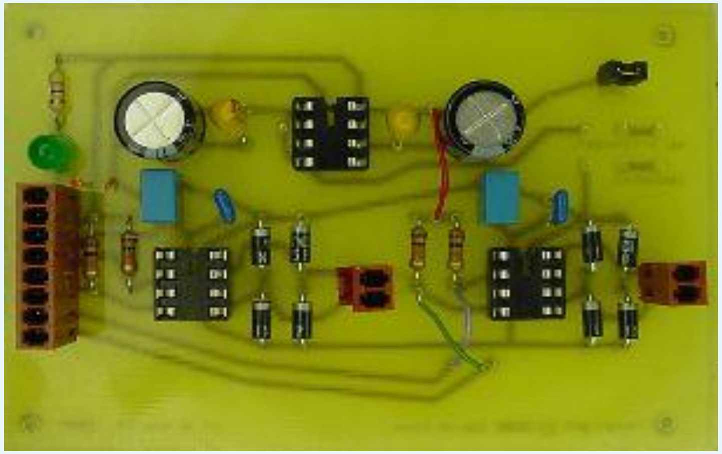
1947: Công ty Bell phát minh Transistor.

Mạch in (PCB: Printed Circuit Board)

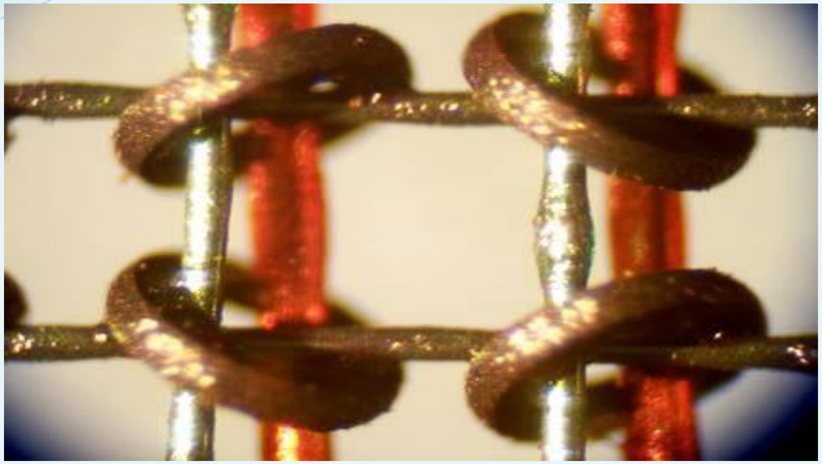


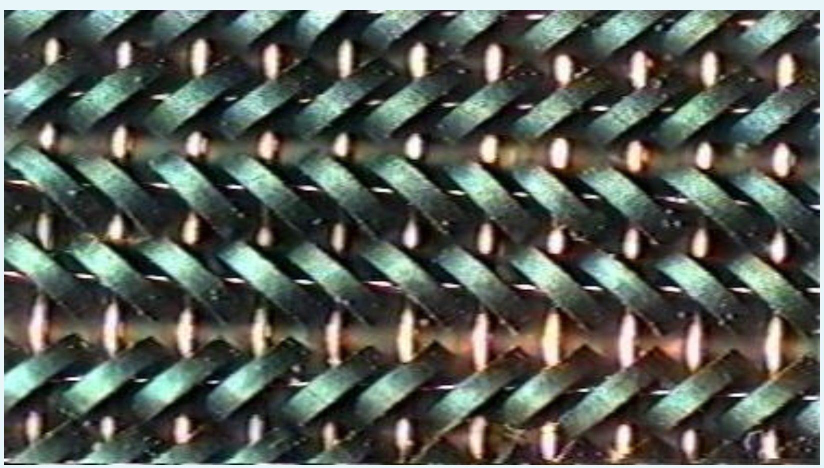




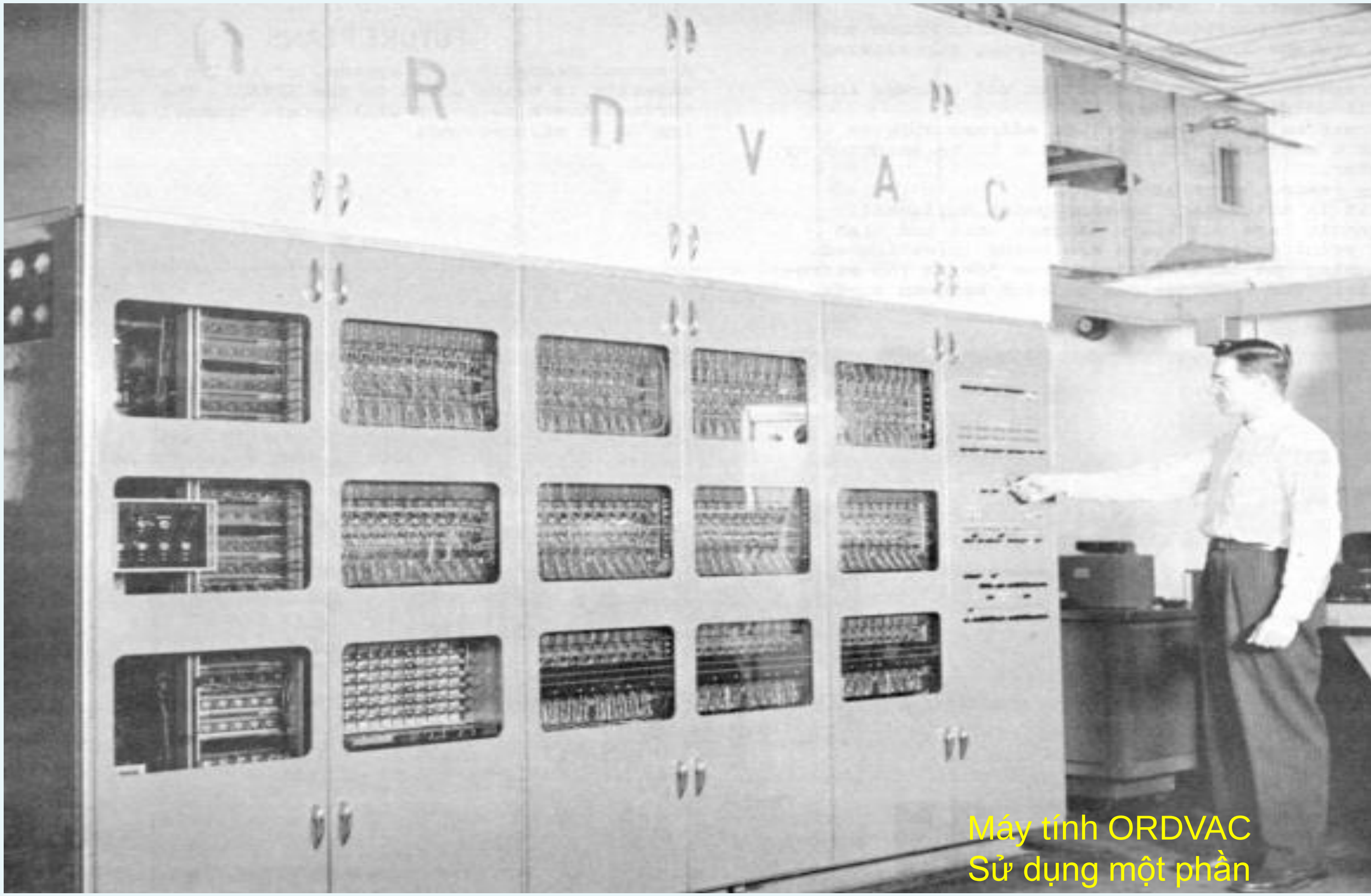


Bộ nhớ xuyến từ





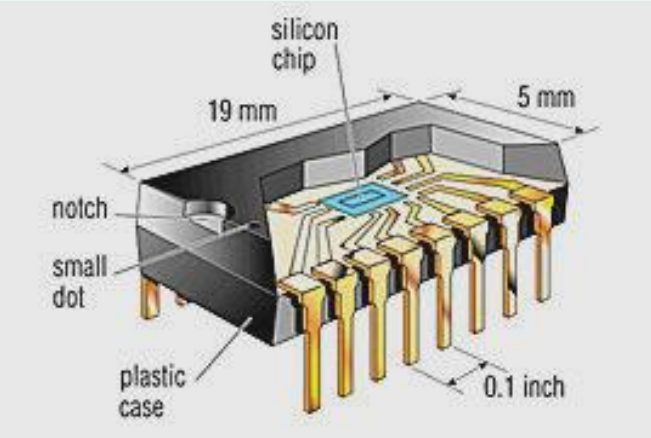
Thế hệ thứ hai (1958-1964)

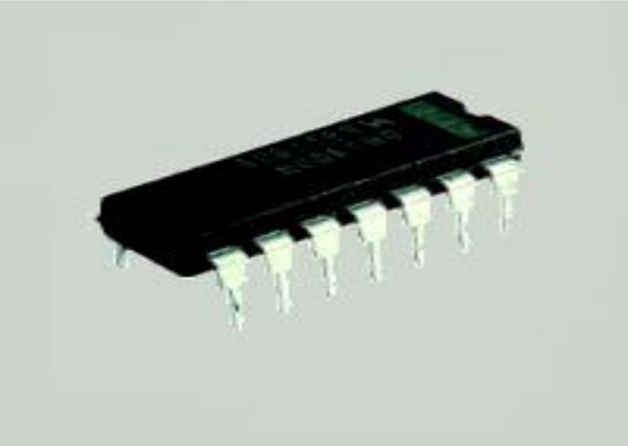


Máy tính ORDVAC
Sử dụng một phần
transistor lưỡng cực

Thế hệ thứ ba (1965-1971):

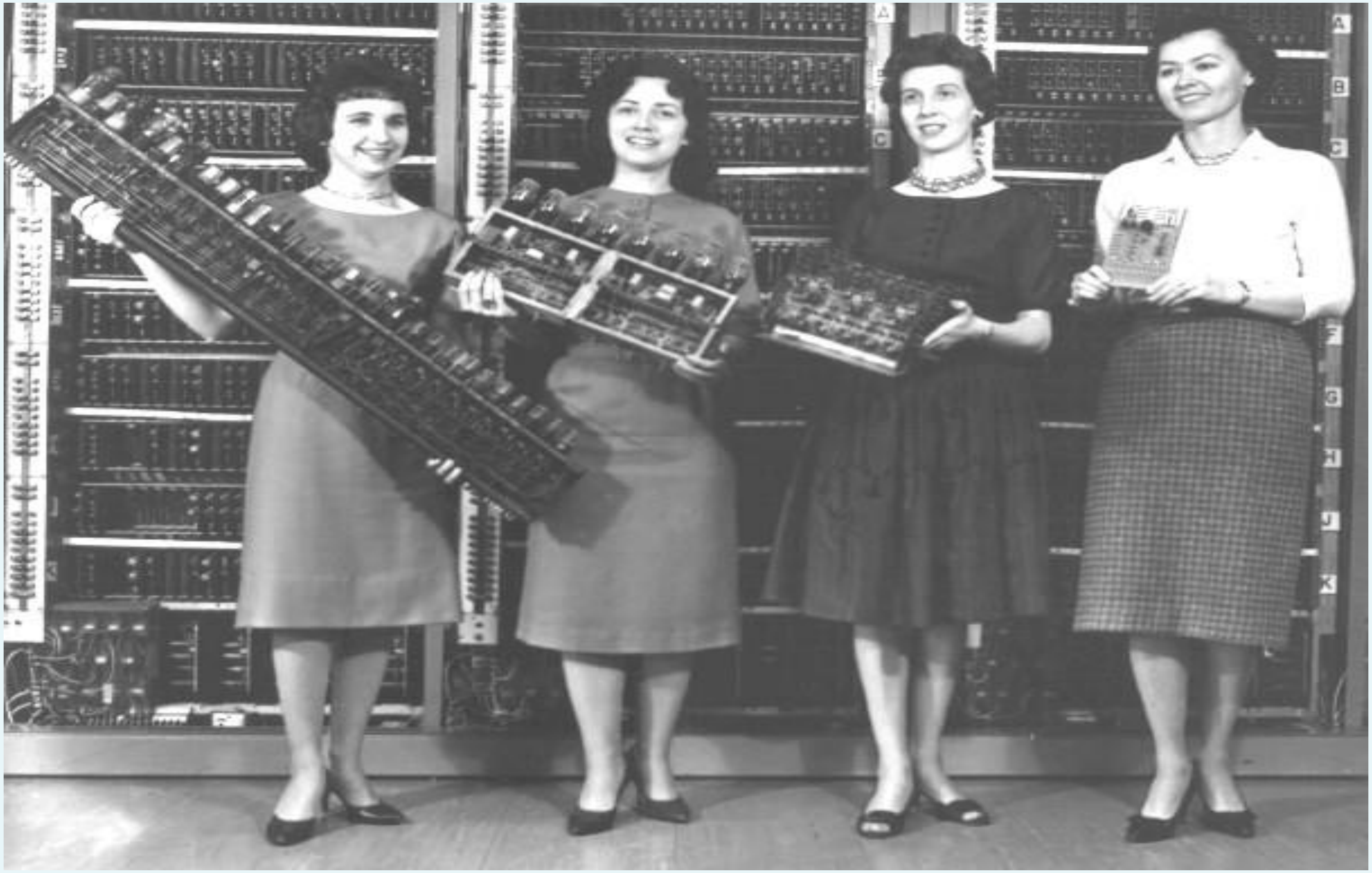
- **Công nghệ chế tạo**
  - Mạch tích hợp (IC: Integrated Circuit).
    - + SSI: Small Scale Integration
    - + MSI: Medium Scale Integration
  - Mạch in nhiều lớp.
  - Bộ nhớ bán dẫn.
- **Phần mềm**
  - Máy tính đa chương trình.
  - Hệ điều hành chia thời gian.





Thế hệ thứ ba (1965-1971):

Thế hệ thứ ba (1965-1971):



Thế hệ thứ tư (1972-…..)

☞ **Công nghệ chế tạo:**
- Mạch tích hợp (IC)
  - LSI: Large scale integration
  - VLSI: very large scale integration
- Bộ vi xử lý (microprocessor)
  - PC (Personal Computer)
  - NC (Network Computer)
- Bộ nhớ bán dẫn, bộ nhớ cache và bộ nhớ ảo.
- Kỹ thuật ống dẫn (pipeline), máy tính song song.

☞ **Phần mềm**
- Các giải thuật song song
- Hệ điều hành phân tán

Thế hệ thứ tư (1972-.....)

Máy tính SUN3

Khuynh hướng hiện tại

Việc chuyển từ thế hệ thứ tư sang thế hệ thứ năm còn chưa rõ ràng.

☞ **Thế hệ của những máy tính thông minh**: Chương trình nghiên cứu của Nhật

* Dựa trên các ngôn ngữ trí tuệ nhân tạo như LISP và PROLOG
* Giao diện người và máy thông minh.

☞ **Thế hệ của máy tính song song**:

Tiến bộ về mật độ tích hợp trong VLSI ⇨ các mạch vi xử lý mạnh

* Các bộ xử lý RISC (1986)
* Các bộ xử lý siêu vô hướng (1990).

Chính các bộ xử lý này giúp thực hiện các máy tính song song với từ vài bộ xử lý đến vài ngàn bộ xử lý.

☞ **Nhận xét**: Ý kiến này cần được bàn cải vì việc ngày có nhiều linh kiện điện tử tích hợp trong một VLSI chưa hẳn là một thay đổi công nghệ cơ bản như ta đã thấy trong sự chuyển đổi giữa các thế hệ máy tính trước đây (chất bán dẫn thay bóng chân không, bộ nhớ bán dẫn thay bộ nhớ điện từ, mạch tích hợp thay transistor rời).

2. Phân loại máy tính

Supercomputer
(Cray-2)

Mainframe
(Honeywell-Bull DPS 7)

MInicomputer
(PDP-8)

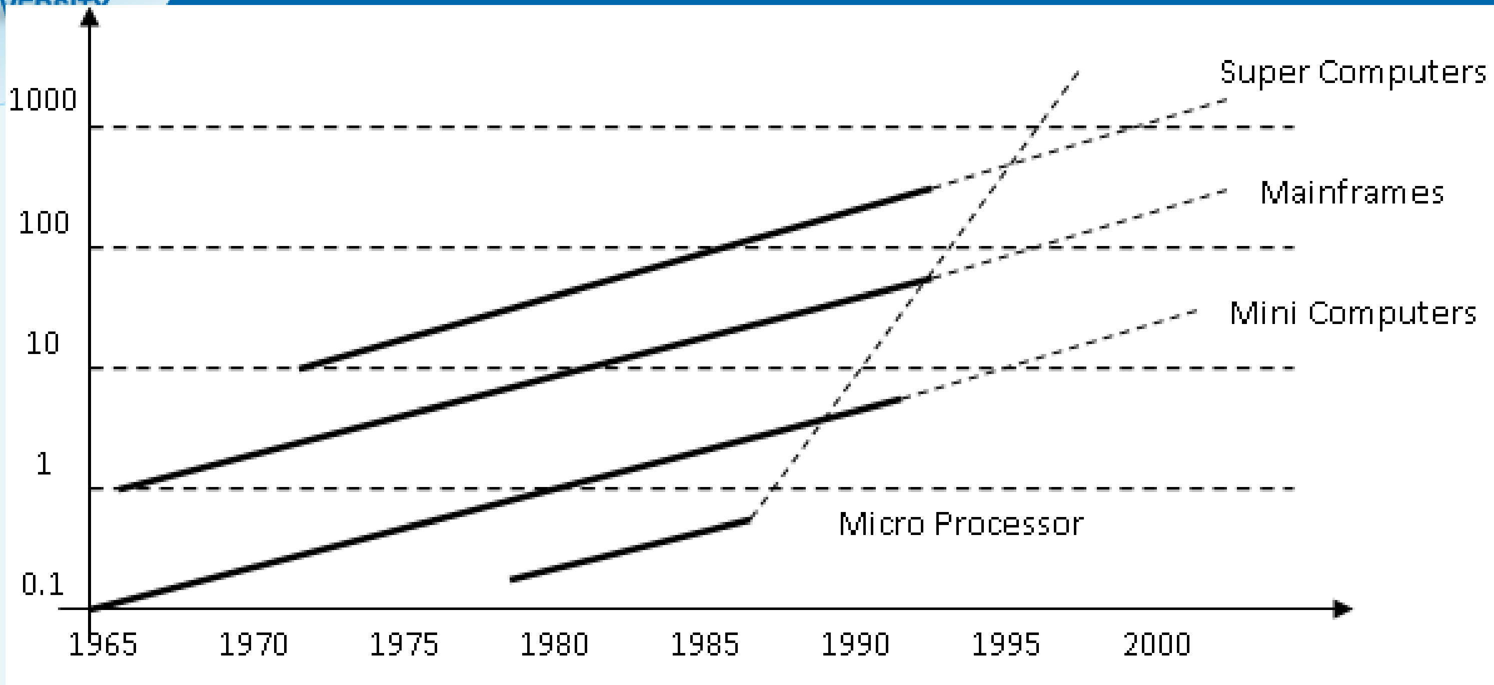
MIcrocomputer
(PC: Personal Computer)

Phân loại máy tính

Việc phân loại máy tính dưa vào tính năng kỹ thuật và giá tiền.

- **Siêu máy tính (Supercomputer):**
  - + Tính băng kỹ thuật rất cao với nhiều bộ xử lý song song
  - + Giá vài triệu USD.
  - + Sử dụng cho tính toán khoa học
- **Máy tính lớn (Mainframe):**
  - + Máy tính đa dụng, với hệ thống vào ra mạnh
  - + Vài trăm ngàn USD.
  - + Sử dụng cho tính toán khoa học và quản lý.
- **Máy tính nhỏ (Minicomputer):**
  - + Đảm nhận một phần công viêc của máy tính lớn.
  - + Giá vài chục ngàn USD.
- **Máy vi tính (Microcomputer):**
  - + Máy tính cá nhân (PC/NC), dùng trong các hệ thống nhỏ.
  - + Vài trăm đến vài ngàn USD.

3. Thành quả máy tính



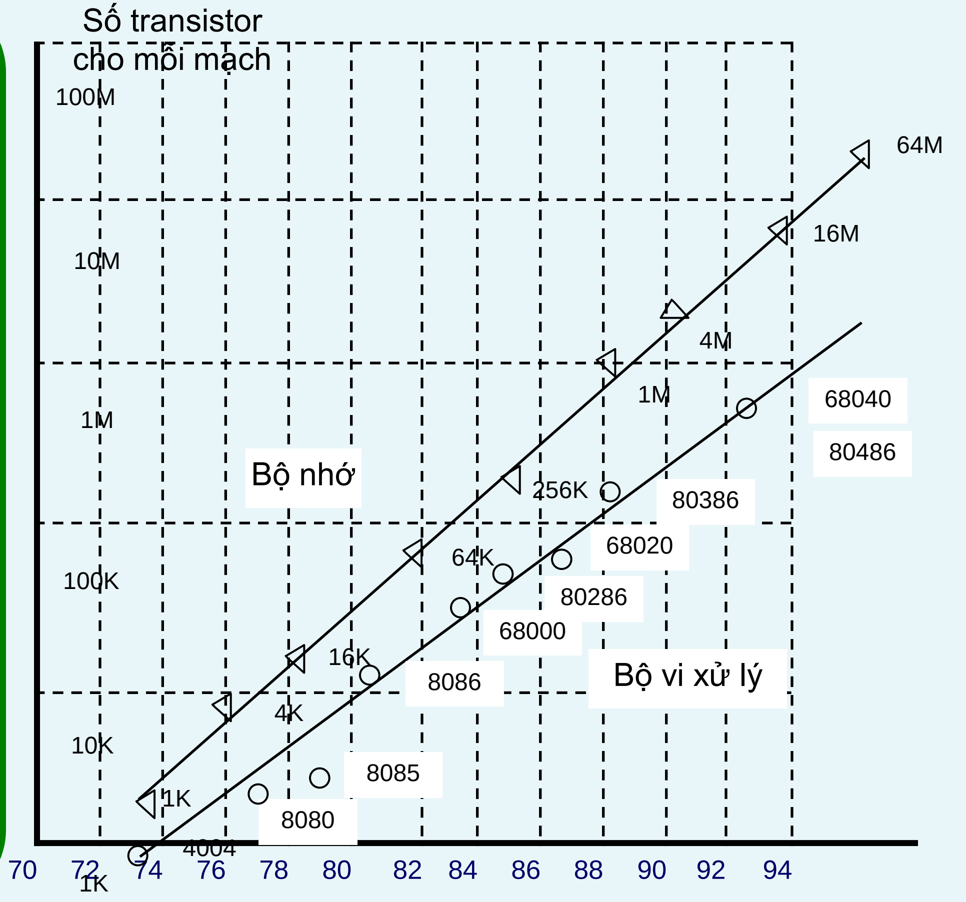
Đánh giá thành quả của các loại máy tính

– Thành quả tối đa của máy tính tăng theo hàm mủ.
– Máy vi tính tăng 35% mỗi năm
– Các loại khác tăng 20% mỗi năm

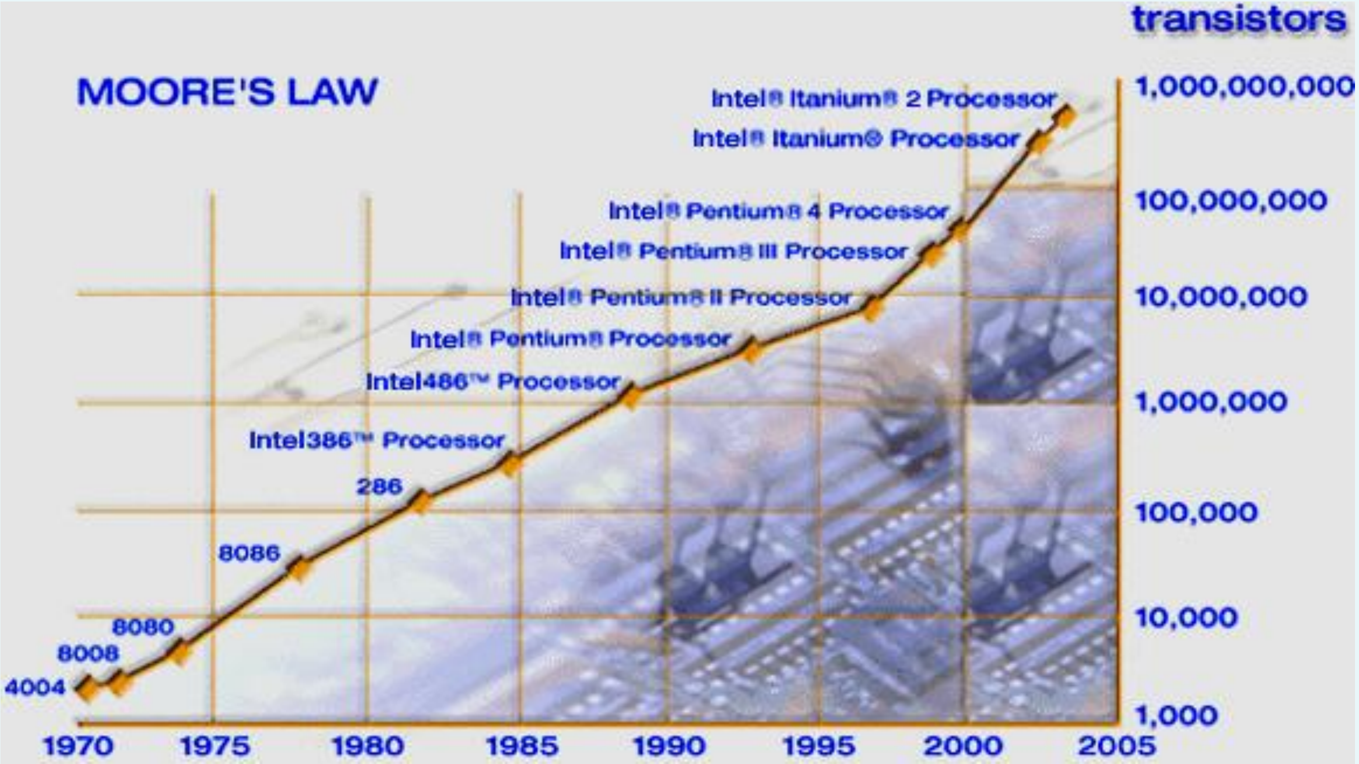
Thành quả máy tính

Đánh giá thành quả về mật độ tích hợp
(Số chức năng)

- Mật độ tich hợp tăng theo hàm mủ
- Mật độ tich hợp tăng 50 % mỗi năm đối với bộ nhớ.
- Mật độ tich hợp tăng 35 % mỗi năm đối với bộ xử lý.



Thành quả máy tính



Mật độ tích hợp tăng theo hàm mũ

Thành quả máy tính

Đánh giá thành quả về tần số xung nhịp

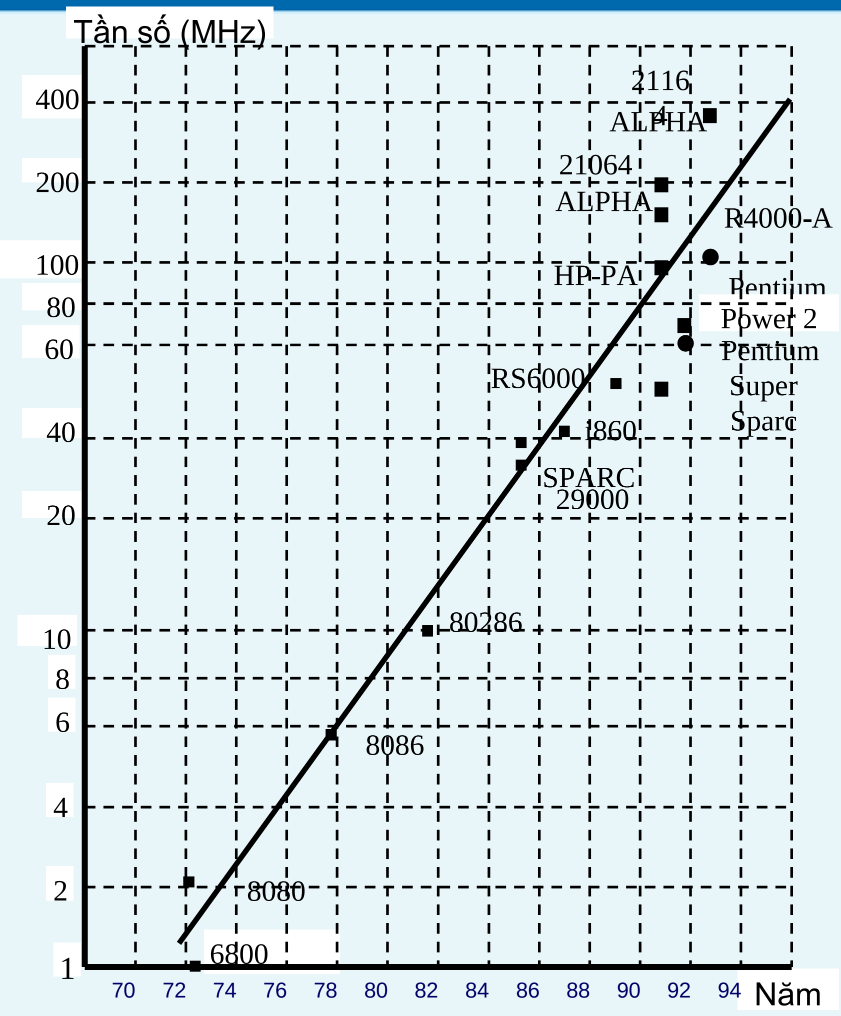
– Tăng theo hàm mủ.
– Tỷ lệ tăng 24% / năm

Công suất tính toán P=S*T

S: Số mạch chức năng
T: Tần số thực hiện nhiệm vụ
⇨ P tăng theo hàm mủ

Công suất tiêu thụ điện

+ nMOS - 12V;
+ pMOS - 5V;
+ CMOS - 3.3V



Thành quả máy tính

Quy luật Moore

- Khả năng của máy tính tăng lên gấp đôi sau 18 tháng với giá thành là như nhau.
- Kết quả của quy luật Moore là:
  - ✓ Chi phí cho máy tính sẽ giảm.
  - ✓ Tốc độ hệ thống sẽ tăng lên.
  - ✓ Tiết kiệm năng lượng cung cấp.
  - ✓ Các IC thay thế cho các linh kiện rời.
  - ✓ Giảm kích thước các linh kiện
  - ⇨ Máy tính sẽ giảm kích thước.

4. Thông tin và mã hoá thông tin

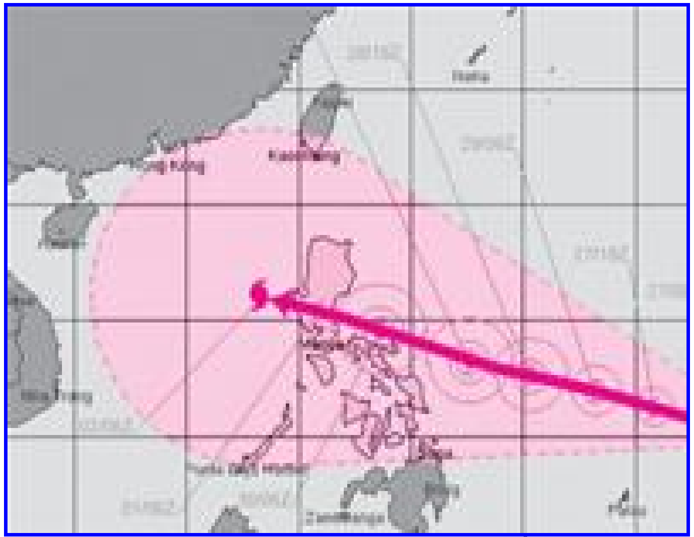
Thông tin (Information) ?
Tín hiệu (Signal)?
Dữ liệu (Data)?

- **Thông tin:** là sự kiện, trạng thái, khái niệm, định luật, định lý,…
- **Tín hiệu:**
  - -là một đại lượng vật lý chứa thông tin
  - -Là một hàm của nhiều biến độc lập
  - -Khi xử lý thì phải biến thành tín hiệu điện (điện áp, dòng điện), nó là hàm 1 biến của thời gian.
- **Dữ liệu:** Là phương tiện, hình thức lưu trữ và truyền tải thông tin

Thông tin gắn liền với sự hiểu biết

TT (TP.HCM) - Đêm qua 27-11, bão Durian (tiếng Thái Lan nghĩa là trái sầu riêng) có tâm ở vào khoảng 10,7 độ vĩ bắc, 137,7 độ kinh đông, cách vùng Visayaz (Philippines) khoảng 1.300km.

Sức gió mạnh nhất gần tâm bão khoảng 93km/giờ (cấp 10), giật 120km/giờ (cấp 12) và đang di chuyển theo hướng tây với tốc độ 20km/giờ.



Dự báo của hải quân Mỹ cho thấy trong ngày 1-12 bão Durian sẽ vào biển Đông

Thông tin và sự mã hoá thông tin

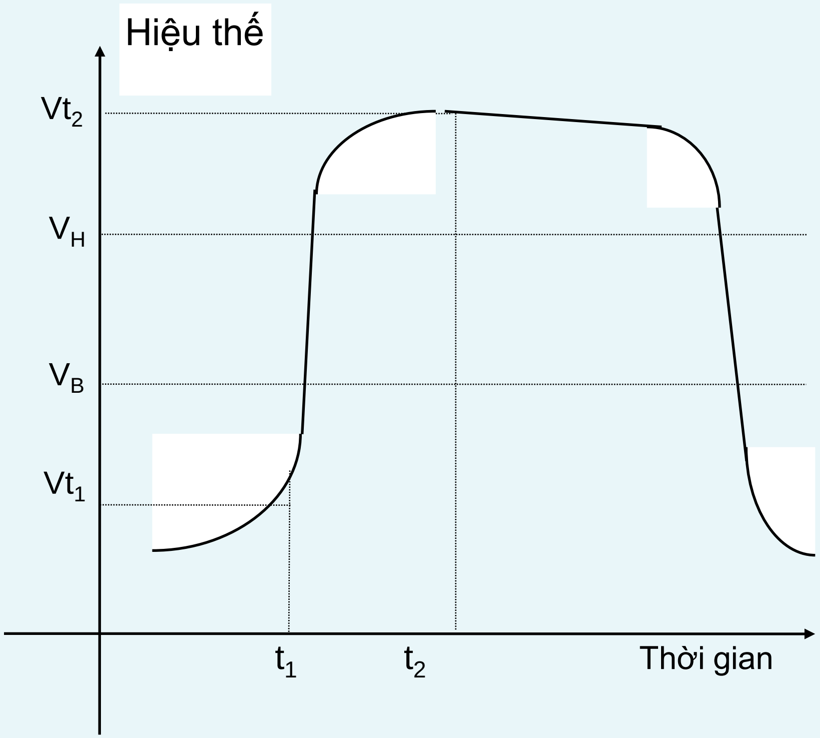
Khái niệm thông tin trong lĩnh vực máy tính

Thông tin là sự xác định một trạng thái trong nhiều trạng thái có thể có vào một thời điểm cho trước.

Quy ước:

- Trạng thái thấp: $V_t \leq V_B$
- Trạng thái cao: $V_t \geq V_H$.

Lưu ý: Thông tin gắn liền với thời gian, thông tin được xác định tại 1 thời điểm.



Ví dụ: Tại thời điểm t_1 thì tín hiệu ở trạng thái thấp (mức 0).
Tại thời điểm t_2 thì tín hiệu ở trạng thái cao (mức 1).

Thông tin và sự mã hoá thông tin

Ví dụ: Tại 1 thời điểm, ta xét điện thế của 1 hoặc nhiều điểm trong mạch điện:

1 điểm A sẽ có thể có 2^1= 2 trạng thái
A
0
1

2 điểm A và B sẽ có thể có 2^2= 4 trạng thái	
A	B
0	0
0	1
1	0
1	1

3 điểm A, B và C sẽ có thể có 2^3= 8 trạng thái		
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Thông tin và sự mã hoá thông tin

Lượng thông tin và sự mã hóa thông tin:

- Thông tin được đo lường bằng đơn vị thông tin mà ta gọi là bít.
- Lượng thông tin được định nghĩa bởi công thức **$I = Log_2(N)$**
  I là lượng thông tin tính bằng bít; **N** là số trạng thái có thể có.

<u>Ví dụ</u>: Sự hiểu biết của một trạng thái trong 8 trạng thái có thể ứng với một lượng thông tin là: $I = Log_2(8) = 3$ bít

Như vậy: *Lượng thông tin là số bit nhị phân cần thiết để biểu diễn số trạng thái có thể có. Một từ n bít tương ứng với một lượng thông tin n bít.*

Trạng thái	X_2	X_1	X_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Thông tin và sự mã hoá thông tin

Biểu diễn các số

– Một số được biểu diễn bằng các ký hiệu (gọi là chữ số, digit) đặt kề nhau. Ví dụ: hệ thống thập phân có 10 quen thuộc: {0,1,2,3,4,5,6,7,8,9}. Giá trị của các chữ số được gọi là trọng số và tùy thuộc vị trí của nó. Chẳng hạn 1998, số thứ 3 từ sau đếm tới có trọng số là 900, số 9 thứ 2 có trọng số là 90.

– **Tổng quát:** một hệ thống số b được định nghĩa gồm các yếu tố sau:

(1) Tập ký hiệu $S_b = \{S_0, S_1, S_2, \ldots, S_{b-1}\}$ (có b ký hiệu)

(2) Cơ số là b

(3) Một số N được viết: $N=(a_{n-1}a_{n-2}\ldots a_0 , a_{-1}a_{-2}\ldots a_{-m})$, với a_i là một phần tử của S_b

N sẽ có giá trị thập phân được tính như sau:

$N=a_{n-1}b^{n-1}+a_{n-2}b^{n-2}+\ldots a_0b^0+ a_{-1}b^{-1}+a_{-2}b^{-2}+\ldots a_{-m}b^{-m}$

Thông tin và sự mã hoá thông tin

Hệ thống số

- Cơ sở của một hệ thống số là tập chữ sô của nó, cở sở của một hệ thống số xác định phạm vi các giá trị có thể có của một chữ số.

  Ví dụ: - Hệ thập phân: Một chữ số có giá trị từ 0 → 9

  - Hệ nhị phân: Một chữ số (1 bit) chỉ có giá trị là 0 hoặc 1

- Công thức tính giá trị thập phân của 1 số trong 1 hệ thống số bất kỳ:

$$N_k = \sum_{i=-m}^{n-1} S_i . b^i$$

Trong đó: N_k: Số cần biểu diễn

m: Số con số của phần lẻ (có chỉ số từ -1 → -m)

n: Số con số của phần nguyên (có chỉ số từ 0→n-1)

S_i: Giá trị của số thứ I

b: Cơ số (hệ số)

<u>Ví dụ:</u> $541.25_{10} = 5x10^{-2} + 2x10^{-1} + 1x10^{0} + 4x10^{1} + 5x10^{2}$

$111.101_{2} = 1x2^{-3} + 0x2^{-2} + 1x2^{-1} + 1x2^{0} + 1x2^{1} + 1x2^{2}$

Thông tin và sự mã hoá thông tin

Đổi từ thập phân sang nhị phân

- Cách biến đổi số thập phân sang nhị phân:

  Ví dụ: Đổi số 23.375_{10} sang số nhị phân

  - Phần nguyên: 23_{10}

23/2	= 11	⇨	1	← d_0
11/2	= 5	⇨	1	← d_1
5/2	= 2	⇨	1	← d_2
2/2	= 1	⇨	0	← d_3
1/2	= 0	⇨	1	← d_4

  $(10111)_2$

  - Phần lẻ: .375

.375x2	= 0.75	⇨	0	← d_{-1}
.75x2	= 1.5	⇨	1	← d_{-2}
.5x2	= 1.0	⇨	1	← d_{-3}

  $(011)_2$

Kết quả:$(\underline{10111}.\underline{011})_2$

Thông tin và sự mã hoá thông tin

Câu hỏi:

(1) Khi chuyển đổi phần lẻ thập phân sang nhị phân, có trường hợp nhân mãi không thể kết thúc chẵn 1 (không có điểm dừng). Khi đó ta sẽ dừng lại khi nào?

- Trong các ngôn ngữ lập trình, mỗi kiểu dữ liệu được mô tả bởi 1 số bit nhị phân xác định (VD: Trong Java, kiểu short có 16 bit, kiểu float có 32 bit,…), ta sẽ nhân đến khi đủ số bit và chấp nhận sai số.
- Ví dụ: 23.31_{10} , giả sử đây là số float 32 bit. Trước tiên ta chuyển phần nguyên sang nhị phân, ta được 10111, có 5 bit. Như vậy, phần lẻ sẽ có số bit còn lại là 27, ta tìm đủ 27 bit cho phần lẻ thì dừng lại.

(1) Trên cơ sở nào người ta đã tìm ra cách chuyển đổi này? (Xem tài liệu tham khảo trên trang web elcit)

Thông tin và sự mã hoá thông tin

Giải thích thêm

Một máy tính được chủ yếu cấu tạo bằng các linh kiện điện tử có hai trạng thái. Vì vậy rất tiện lợi khi dùng các số nhị phân để biểu diễn số trạng thái của các mạch điện hoặc để mã hóa các ký tự, các số cần thiết cho vận hành của máy tính.

Để đơn giản hóa việc viết các số nhị phân người ta đã dùng một tổ hợp 4 bit nhị phân để mô tả một ký hiệu thập phân hoặc thập lục phân (mã BCD, sẽ được đề cập trong phần sau).

Số nhị phân	Số thập lục phân	Số nhị phân	Số thập lục phân
0 0 0 0	0	1 0 0 0	8
0 0 0 1	1	1 0 0 1	9
0 0 1 0	2	1 0 1 0	A
0 0 1 1	3	1 0 1 1	B
0 1 0 0	4	1 1 0 0	C
0 1 0 1	5	1 1 0 1	D
0 1 1 0	6	1 1 1 0	E
0 1 1 1	7	1 1 1 1	F

Thông tin và sự mã hoá thông tin

Biểu diễn các số trong hệ nhị phân

- **Số nguyên dương:** Một từ n bít có thể biểu diễn tất cả các số dương từ 0 tới 2^n-1. Nếu d_i là một số nhị phân thứ i, một từ n bít tương ứng với một số nguyên thập phân N.

d_{n-1}	d_{n-2}	d_{n-3}	. . .	. . .	d_2	d_1	d_0

$$N = \sum_{i=0}^{n-1} d_i 2^i$$

VD: *Một byte (gồm 8 bít) có thể biểu diễn các số từ 0 tới 255 và một từ 32 bít cho phép biểu diễn các số từ 0 tới 4,294,967,295.*

Ví dụ: Cho một số nhị phân 8 bits

d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0	1	1	0	1	0	1	1

$$N = \sum_{i=0}^{7} d_i 2^i = 1x2^0+1x2^1+0x2^2+1x2^3+0x2^4+1x2^5+1x2^6+0x2^7 = 107_{10}$$

Thông tin và sự mã hoá thông tin

Số nguyên có dấu

Có nhiều cách để biểu diễn một số n bít có dấu. Trong hầu hết mọi cách thì bít cao nhất luôn tượng trưng cho dấu.

- Bit dấu = 0 thì số nguyên dương
- Bit dấu = 1 thì số nguyên âm.

d_{n-1}	d_{n-2}	d_{n-3}	. . .	. . .	d_2	d_1	d_0

Bit dấu

- Số nguyên dương có bit $d_{n-1} = 0$ và có giá trị tuyệt đối xác định bởi các bit từ d_0 tới d_{n-2}.
- Số nguyên âm có bit $d_{n-1} = 1$ và có trị số phụ thuộc vào phương pháp biểu diễn số và các bit từ d_0 tới d_{n-2}

Thông tin và sự mã hoá thông tin

Số nguyên có dấu

➢ **Cách biểu diễn bằng trị tuyệt đối và dấu:**

- Bít d_{n-1} là bít dấu và các bít từ d_0 … d_{n-2} cho giá trị tuyệt đối.
- Một từ n bít tương ứng với số nguyên thập phân có dấu là:

$$N = (-1)^{d_{n-1}} \sum_{i=0}^{n-2} d_i 2^i$$

- Một byte có thể biểu diễn các số có dấu: **-127** tới **+127**

$(1111\ 1111)_2 – (0111\ 1111)_2$

- Có hai cách biểu diễn số 0 là 0000 0000 (+0) và 1000 0000 (-0).
- Một từ 32 bít cho phép biểu diễn các số có dấu:-$(2^{31}$-1)→+$(2^{31}$-1) với hai cách biểu diễn số không.

Ví dụ: Ta có số + 65_{10} = 01000001_2

\- 65_{10} = 11000001_2

Thông tin và sự mã hoá thông tin

- **Chuyển từ THẬP PHÂN sang NHỊ PHÂN bit dấu**
  – Đổi giá trị tuyệt đối sang nhị phân (số dương)
  – Thêm bit 0 vào phía trước cho đủ bit (nếu chưa đủ bit qui định).
  – Đổi bit có trọng số lớn nhất thành 1, nếu là số âm
- VD: Chuyển đổi các số thập phân sau sang nhị phân bit đấu 16 bit: 1987, -256

Chuyển từ NHỊ PHÂN bit dấu sang THẬP PHÂN

- Áp dụng công thức:

$$N = (-1)^{d_{n-1}} \sum_{i=-m}^{n-2} d_i 2^i$$

VD: Chuyển đổi các số nhị phân bit dấu sau sang thập phân:

01100101

11001110

Thông tin và sự mã hoá thông tin

Số nguyên có dấu

➢ Cách biểu diễn bằng số bù 1

- Trong cách biểu diễn này, số âm -N được có bằng cách thay các số nhị phân d_i của số dương N bằng số bù của nó (nghĩa là nếu d_i = 0 thì người ta đổi nó thành 1 và ngược lại).
- Một byte sẽ biểu diễn tất cả các số có dấu:

Từ -127 đến +127

$(10000000)_2 \rightarrow (01111111)_2$

- Với hai cách biểu diễn cho 0 là 0000 0000 (+0) và 11111111 (-0).

<u>Ví dụ</u>: Cho số âm -56 đổi ra số nhị phân 8 bit.

Ta có số dương $56_{10} = 00111000_2$

Lấy số bù 1 ta được số âm $-56_{10} = 11000111_2$

Thông tin và sự mã hoá thông tin

CANTHO UNIVERSITY

Đổi từ THẬP PHÂN sang NHỊ PHÂN KIỂU BÙ 1

1) Đổi giá trị tuyệt đối sang nhị phân (số dương)
2) Thêm bit 0 vào phía trước cho đủ bit (nếu chưa đủ bit qui định).
3) Lấy bù 1, nếu là số âm

VD: Đổi từ thập phân sang nhị phân 16 bit kiểu bù 1 các số sau: 367, -145

Đổi từ NHỊ PHÂN KIỂU BÙ 1 sang THẬP PHÂN

Nếu bit trọng số lớn nhất là 0 (số dương) áp dụng công thức: $N = \sum_{i=-m}^{n-2} d_i 2^i$

Nếu bit trọng số lớn nhất là 1 (số âm): $N = (-1)^{d_{n-1}} \sum_{i=-m}^{n-2} \bar{d}_i 2^i$

Với $\bar{d}_i$ là bù của d_i

VD: Đổi từ số nhị phân bù 1 sang thập phân các số sau:

01100101 và 10010101

Thông tin và sự mã hoá thông tin

Số nguyên có dấu

➤ **Cách biểu diễn bằng số bù 2**

- Để có số bù 2 của một số nào đó, người ta lấy số bù 1 rồi cộng thêm 1. Vậy một từ n bít (d_{n-1} .... d_0) có trị thập phân.

$$N = -d_{n-1}2^{n-1} + \sum_{i=0}^{n-2} d_i 2^i$$

- Một từ n bít có thể biểu diễn các số có dấu từ - 2^{n-1} đến 2^{n-1}-1.
- Chỉ có một cách duy nhất để biểu diễn cho số không là tất cả các bít của số đó đều bằng không.

Ví dụ: Cho số âm -56 đổi ra số nhị phân 8 bit.

Ta có số dương	56_{10}	$= 00111000_2$
Lấy số bù 1		$= 11000111_2$
Sau đó cộng thêm 1		$+ 00000001_2$
Ta dược số âm	-56_{10}	$= 11001000_2$

Thông tin và sự mã hoá thông tin

Đổi từ THẬP PHÂN → NHỊ PHÂN KIỂU BÙ 2

1)Đổi giá trị tuyệt đối sang nhị phân (số dương)

2)Thêm bit 0 vào phía trước cho đủ bit (nếu chưa đủ bit qui định).

3)Lấy bù 2, nếu là số âm

VD: Đổi từ thập phân sang nhị phân 16 bit kiểu bù 2 các số sau: 367, -145

Đổi từ NHỊ PHÂN KIỂU BÙ 2 → THẬP PHÂN

1)Nếu bit trọng số lớn nhất là 0 (số dương) áp dụng CT: $N = \sum_{i=-m}^{n-2} d_i . 2^i$

2)Nếu bit trọng số lớn nhất là 1 (số âm),

Cách 1: lấy bù 2 để có số dương, áp dụng công thức trên để tính trị tuyệt đối. Sau đó thêm vào dấu -

Cách 2: Áp dụng công thức: $N = -d_{n-1} 2^{n-1} + \sum_{i=0}^{n-2} d_i 2^i$

VD: Đổi từ số nhị phân bù 1 sang thập phân các số sau:

01100101 và 10010101

Thông tin và sự mã hoá thông tin

Số nguyên có dấu

➢ **Cách biểu diễn bằng số thừa K**

Trong cách này, số âm nhỏ nhất có thể biểu diễn được sẽ được qui ước lại là 0 và số 0 được dời lên đến K.

Ví dụ: Số âm nhỏ nhất có thể biểu diễn được là -127 và ta chọn số thừa K là K = 127 thì các số nguyên có dấu được biểu diễn như sau:

Biểu diễn thông thường	-127	-126	…	-1	0	1	…	125	126
Biểu diễn bằng số thừa K	0	1	…	126	127	128	…	252	253

Vậy một số nguyên dương N sẽ bđược biểu diễn bằng số thừa K là N_K = K + N. Một số nguyên âm –N sẽ được biểu diễn bằng số thừa K là N_K = K – N (với K ≥ N)

Thông tin và sự mã hoá thông tin

- **Cách biểu diễn bằng số thừa K**
  Tổng quát, N là số đại số thì giá trị thừa K của nó là $N_K = K + N$.
  Số thừa K được chọn sao cho tổng của K và một số âm bất kỳ luôn luôn dương.
  Ví dụ: Đổi số thập sau thành số thừa K=128
  $+25_{10}$ = $128_{10} + 25_{10} = 1000\ 0000_2 + 0001\ 1001_2 = 1001\ 1001_2$
  -25_{10} = $0110\ 0110_2$ (lấy số bù 1 của +25)
  $0000\ 0001_2$ (cộng 1)

  $0110\ 0111_2$ (103 = 128-25)
  Nhận xét: - Một số 8 bit, giá trị lớn nhất: +127, giá trị nhỏ nhất: -128
  - Chỉ có 1 giá trị 0: +0 ($1000\ 0000_2$), -0 ($1000\ 0000_2$)

Thông tin và sự mã hoá thông tin

Đổi từ số thập phân sang nhị phân thừa K

Áp dụng công thức $N_K = K + N$

Cách 1: Thực hiện trên số nhị phân: đổi K và N ra nhị phân thực hiện phép cộng nhị phân (Lưu ý N là số nhị phân bù 2) theo công thức trên.

Cách 2: Thực hiện phép cộng thập phân (N có thể dương hoặc âm) sau đó chuyển sang nhị phân

Bài tập: Đổi các số thập phân sau đây sang nhị phân thừa K, K = 128:

105, -78

Đổi từ số nhị phân thừa K sang thập phân

Áp dụng công thức $N = N_K - K$

Cách 1: Thực hiện trên số nhị phân: đổi K ra nhị phân thực hiện phép trừ nhị phân (Lưu ý: Phép trừ số nhị phân tương đương với phép cộng sau đó lấy bù 2) theo công thức trên.

Cách 2: Đổi N_K sang thập phân sau đó thực hiện phép trừ thập phân

Bài tập: Đổi các số thập phân sau đây sang nhị phân thừa K, K = 128:

10010011, 01110111

CANTHO UNIVERSITY

Thông tin và sự mã hoá thông tin

Decimal	Unsigned	Sign–Mag	1's Comp.	2's Comp.	Excess 4
7	111	-	-	-	-
6	110	-	-	-	-
5	101	-	-	-	-
4	100	-	-	-	-
3	011	011	011	011	111
2	010	010	010	010	110
1	001	001	001	001	101
+0	000	000	000	000	100
-0	-	100	111	000	100
-1	-	101	110	111	011
-2	-	110	101	110	010
-3	-	111	100	101	001
-4	-	-	-	100	000

3-bit Interger Representations

Thông tin và sự mã hoá thông tin

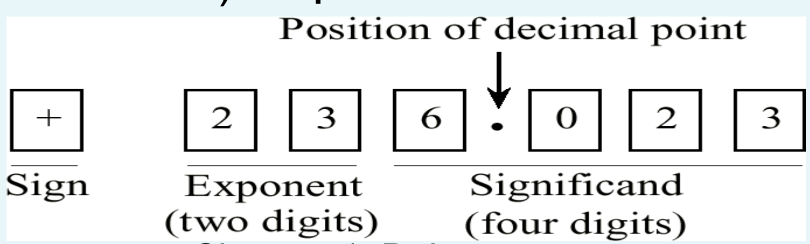
Số nguyên có dấu

- **Nhận xét:**
  - Cách biểu diễn số nguyên có dấu bằng số bù 2 được dùng cho các phép tính số nguyên. Nó không cần thuật toán đặc biệt cho các phép tính cộng và trừ, và dễ phát hiện các trường hợp bị tràn.
  - Các cách biểu diễn "dấu, trị tuyệt đối" hoặc "số bù 1" phải dùng các thuật toán phức tạp và có hai cách biểu diễn của số không. Cách biểu diễn "dấu, trị tuyệt đối" dùng cho phép nhân của số có dấu chấm động.
  - Cách biểu diễn bằng số thừa K dùng cho số mũ của số có dấu chấm động. Cách này làm cho việc so sánh các số mũ có dấu khác nhau trở thành việc so sánh các số nguyên dương.

Thông tin và sự mã hoá thông tin

Cách biểu diễn số với dấu chấm động

- Trong hệ thập phân, số 254 có thể biểu diễn dưới các dạng sau:
  254×10^0, 25.4×10^1, 2.54×10^2, 0.254×10^3, 0.0254×10^4, …
  Các cách biểu diễn khác nhau sẽ gây khó khăn khi so sánh các số.
- Trong hệ nhị phân, số chuẩn hoá được biểu diễn như sau.
  <u>Ví dụ:</u> đổi $(9.375 \times 10^{-2})_{10}$ sang số nhị phân dạng chuẩn hoá
  Ta có $(9.375 \times 10^{-2}) = (0.09375)_{10} = (0.00011)_2 = 1.1 \times 2^{-4}$
- Số chấm động được *chuẩn hoá*, cho phép biểu diễn gần đúng các số thập phân rất lớn hay rất nhỏ. Thành phần của số chấm động bao gồm: *phần dấu*, *phần mũ* và *phần thập phân*. Như vậy, tất cả các số đều có cùng cách biểu diễn.
  Ví dụ: số $(+6.023 \times 10^{23})$ được biểu diễn như sau:



Thông tin và sự mã hoá thông tin

Cách biểu diễn số với dấu chấm động

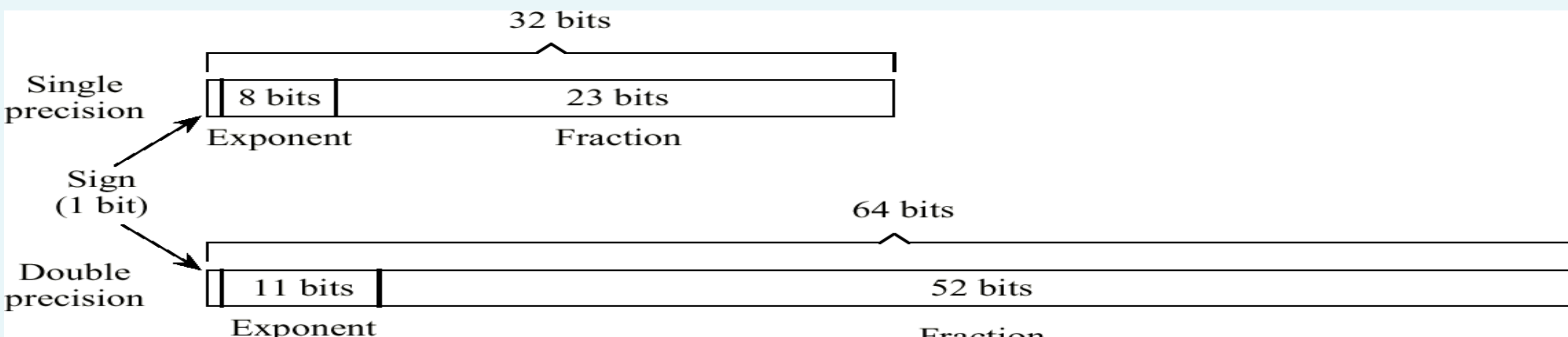
- Cách này cho phép biểu diễn gần đúng các số thực. Một số sẽ được biểu diễn bằng dấu của nó, phần định trị và số mũ của nó.
- Theo chuẩn IEEE 754 được dùng rộng rãi hiện nay, phần định trị có dạng 1,f với số 1 ẩn tàng và f là phần số lẽ. Tuỳ theo độ chính xác ta có:

  Số thực chính xác kép (64 bit) $(-1)^S \times (1,f_1\ f_2\ .....\ f_{52}) \times 2^{(E-1023)}$.

  Số thực chính xác đơn (32 bit) $(-1)^S \times (1,f_1\ f_2\ .....\ f_{23}) \times 2^{(E-127)}$.
- Chuẩn IEEE 754:

  - Các số chuẩn hóa (các bít của E không cùng lúc bằng 0 hoặc 1)

  - Trị số 0 (các bít của E không cùng lúc bằng 0 và phần lẽ = 0),



Thông tin và sự mã hoá thông tin

Cách biểu diễn số với dấu chấm động

➢ **<u>Nhận xét:</u>**

\+ Số mũ được biểu diễn bằng số thừa K= 1023

Số mũ thay đổi từ -1022 đến 1023.

Số chuẩn hóa dương - nhỏ nhứt = 2^{-1022}

- lớn nhứt $\approx 2 \times 2^{1023} \approx 2^{1024}$

➢ **<u>Ví dụ:</u>** Cho số thực 438.25 được biểu diễn như sau:

$438.25 = 110110110.01 = 1.1011011001_2 \times 2^8$

$= (-1)^S \times (1.f_1\ f_2\ .....\ f_{23}) \times 2^{(E - 127)}$.

⇨ $S = 0_{10} = 0_2$ (1bit)

$E = 127+8 = 135_{10} = 10000111_2$ (8bit)

$f = 1011011001_2 = 10110110010000000000000_2$ (23bit)

S	E								f																						
d_{31}	d_{31}	d_{29}	d_{28}	d_{27}	d_{26}	d_{25}	d_{24}	D_{23}	d_{22}	d_{21}	d_{20}	d_{19}	d_{18}	d_{17}	d_{16}	d_{15}	d_{11}	d_{13}	d_{12}	d_{11}	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0	1	0	0	0	0	1	1	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Thông tin và sự mã hoá thông tin

- **Đổi từ thập phân sang dấu chấm động**

1) Đổi số thập phân sang số nhị phân.
2) Viết lại dạng chuẩn hoá.
3) Chuyển số mũ thành số thừa K
4) Điền vào các trường của thanh ghi (phần số lẻ nếu thiếu bit thì thêm bit 0 vào phía sau cho đủ bit)

Bài tập: Chuyển các số thập phân sau đây sang số nhị phân dấu chấm động chính xác đơn:

253.52; - 317.31

- **Đổi từ số nhị phân dấu chấm động sang thập phân**

1) Chuyển số mũ từ thưa K sang số thập phân.
2) Viết lại dạng chuẩn hoá và dời dấu chấm thập phân về dạng thường (số mũ bắng 0).
3) Đổi phần nguyên và phần lẻ thành số thập phân (giá trị tuyệt đối).
4) Thêm dấu trừ nều là số âm.

Bài tập: Tìm giá trị thập phâng của các số đầu chấm động sau:

0 10000000 11000000000000000000000

1 01111011 01010000000000000000000

Thông tin và sự mã hoá thông tin

Biểu diễn các số thập phân/Hex bằng mã BCD

+ Một vài ứng dụng, đặc biệt ứng dụng quản lý, bắt buộc các phép tính thập phân phải chính xác, không làm tròn số.
+ Với một số bít cố định, ta không thể đổi một cách chính xác số nhị phân thành số thập phân và ngược lại.
+ Vì vậy, khi cần phải dùng số thập phân, ta dùng cách biểu diễn số thập phân mã bằng nhị phân (BCD: Binary Coded Decimal) theo đó mỗi số thập phân / Hex được mã với 4 số nhị phân (bảng I.6).

Số thập phân	Mã BCD				Số thập phân	Mã BCD			
	d3	d2	d1	d0		d3	d2	d1	d0
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	A	1	0	1	0
3	0	0	1	1	B	1	0	1	1
4	0	1	0	0	C	1	1	0	0
5	0	1	0	1	D	1	1	0	1
6	0	1	1	0	E	1	1	1	0
7	0	1	1	1	F	1	1	1	1

Thông tin và sự mã hoá thông tin

- **Biểu diễn các số thập phân bằng mã BCD**
- Số dương: qui ước dấu cộng là tổ hợp 4 bit 0 đứng trước (0000)
- Số âm: là bù 10 của số dương
- Bù 10 là bù 9 cộng 1.
- Bù 9 của 1 số N có n chữsố là một số (ký hiệu là N_{b9}) là số có n chữ số thoả điều kiện: $N + N_{b9}$ là một số có n chữ số đều là 9.

Ví dụ 1: Số thập phân 1357_{10} được biểu diễn bằng mã BCD như sau

0000 0001 0011 0101 0111

Ví dụ 2: Số thập phân -1357_{10} được biểu diễn bằng mã BCD như sau

- Lấy bù 9 của 1357, ta được: 1001 1000 0110 0100 0011
- Lấy bù 10 ta được -1357_{10} = **1001** 1000 0110 0100 0100

Nhận xét:

- 1 số dương có tổ hợp 4 bit đầu là 0000 (tương ứng với dấu +)
- 1 số âm có tổ hợp 4 bit đầu là 1001 (tương ứng với dấu -)

Thông tin và sự mã hoá thông tin

Bài tập:

1. Biểu diễn các số thập phân sau bằng mã BCD:

2010; -2010

2. Cho biết giá trị thập phân các mã BCD sau đây:

0000 0101 1000 0110 0111 0001 0010

10101 0100 0011 1001 0011 0101 0100

- **Biểu diễn các số HEX bằng mã BCD**
- Tương tự như số thập phân, mỗi chữ số của 1 số hex được biểu bằng tổ hợp 4 bit nhị phân.

Ví dụ: Ta xét trường hợp số Hex không có dấu là 3A6E, mã BCD của nó là:

0011 1010 0110 1110

Thông tin và sự mã hoá thông tin

- **Chuyễn đổi giữa số thập phân và số HEX**
- **Đổi từ thập phân sang thập lục phân:** ta cũng dùng cách lặp lại phép chia cho 16 và lấy số dư như trước.

Ví dụ: Đổi số 765_{10} thành số thập lục phân. Thực hiện phép chia như sau:

$$\frac{765}{16} = 47 + \text{dư } 13 \leftarrow \text{LSD}$$

$$\frac{47}{16} = 2 + \text{dư } 15$$

$$\frac{2}{16} = 0 + \text{dư } 2 \leftarrow \text{MSD}$$

$$\Rightarrow 765_{10} = 2FD_{16}$$

- **Đổi từ thập lục phân sang thập phân: N =** $\sum_{i=0}^{n-1} d_i 16^i$

CANTHO UNIVERSITY

Thông tin và sự mã hoá thông tin

Chuyễn đổi giữa số nhị phân và số HEX

- **Đổi từ HEX sang nhị phân: mỗi ký hiệu thập lục phân được đổi sang giá trị nhị phân bằng mã BCD. Ví dụ:** Đổi số 8D2H (H là biểu thị cho số Hex**)** thành nhị phân

$8D2_{16}$	=	8	D	2
	=	⇓	⇓	⇓
	=	1000	1101	0010
	=	100011010010_2		

- **Đổi từ nhị phân sang HEX: bắt đầu từ bit có trọng số nhỏ nhất, nhóm 4 bit thành 1 ký hiệu số Hex (mã BCD), đền các bit có trọng số lớn nhất nếu không đủ 4 bit thì thêm bit 0 vào. Ví dụ: Đổi số 110011011012 thành số HEX**

11001101101_2	=	0110	0110	1101
	=	⇓	⇓	⇓
	=	6	6	D
	=	$66D_{16}$		

Thông tin và sự mã hoá thông tin

Nhận xét

- Mã BCD của số Hex tương ứng với 1 số thập phân chính là dạng nhị phân của số thập phận đó.

Ví dụ: $67_{10} = 43_{Hex} = 0100\ 0011_2$

→ đổi từ thập phân sang nhị phân ta có thể thực hiện theo cách sau: đổi thập phân sang số hex sau đó viết mã BCD.

- Ta có thể đổi từ thập phân sang nhị phân bằng cách đi ngược lại quá trình đổi nhị phân sang thập phân, đó là : số thập phân được trình bày dưới dạng tổng các lũy thừa của 2, sau đó ghi các kí số 0 và 1 vào vị trí bit tương ứng.

Ví dụ: $135_{10} = 128 + 4 + 2 + 1 = 2^7 + 2^2 + 2^1 + 2^0 = 10000111_2$

Cách này có thể nhanh hơn cách chia lặp cho 2 như đã trình bày trước đây.

CANTHO UNIVERSITY

Thông tin và sự mã hoá thông tin

Biểu diễn các ký tự

+ Người ta thường dùng mã ASCII (American Standard Codes for Information Interchange) để biểu diễn các chữ, số và một số dấu thường dùng mà ta gọi chung là ký tự.
+ Mỗi ký tự được biểu diễn bằng một byte. Ví dụ, trong mã ASCII, chữ "A" được biểu diễn bằng mã 01000001_2 và số "9" được biểu diễn bởi 00111001_2.

Ký tự	Ascii Code			Ký tự	Ascii Code			Ký tự	Ascii Code		
	DEC	HEX	BIN		DEC	HEX	BIN		DEC	HEX	BIN
Space	32	20	00100000	**G**	71	47	01000111	**Q**	81	51	01010001
0	48	30	00110000	**H**	72	48	01001000	**R**	82	52	01010010
…	-	-	-	**I**	73	49	01001001	**S**	83	53	01010011
9	57	39	00111001	**J**	74	4A	01001010	**T**	84	54	01010100
A	65	41	01000001	**K**	75	4B	01001011	**U**	85	55	01010101
B	66	42	01000010	**L**	76	4C	01001100	**V**	86	56	01010110
C	67	43	01000011	**M**	77	4D	01001101	**W**	87	57	01010111
D	68	44	01000100	**N**	78	4E	01001110	**X**	88	58	01011000
E	69	45	01000101	**O**	79	4F	01001111	**Y**	89	59	01011001
F	70	46	01000110	**P**	80	50	01010000	**Z**	90	5A	01011010

CANTHO UNIVERSITY

Câu hỏi ôn tập

1. **Tiêu chuẩn phân chia máy tính thành các thế hệ ?**
2. **Đặc trưng cơ bản của các máy tính thế hệ thứ nhất ?**
3. **Đặc trưng cơ bản của các máy tính thế hệ thứ hai ?**
4. **Đặc trưng cơ bản của các máy tính thế hệ thứ ba ?**
5. **Đặc trưng cơ bản của các máy tính thế hệ thứ tư ?**
6. **Khuynh hướng phát triển của máy tính điện tử ngày nay ?**
7. **Tiêu chuẩn phân loại máy tính điện tử ?**
8. **Đánh giá thành quả chung của máy tính điện tử ?**
9. **Đánh giá thành quả của mật độ tích hợp ?**
10. **Đánh giá thành quả của tần số xung nhịp ?**
11. **Khái niệm thông tin trong máy tính được hiểu như thế nào?**
12. **Lượng thông tin là gì ?**
13. **Điểm chung nhất trong các cách biểu diễn một số nguyên n bit có dấu là gì?**
14. **Biểu diễn số nguyên có dấu n bit bằng dấu và trị tuyệt đối ?**
15. **Biểu diễn số nguyên có dấu n bit bằng số bù 1 ?**
16. **Biểu diễn số nguyên có dấu n bit bằng số bù 2 ?**
17. **Biểu diễn số nguyên có dấu n bit bằng số thừa K ?**
18. **Biểu diễn số thực dấu chấm động ?**
19. **Biểu diễn số thập phân BCD ?**
20. **Biểu diễn các ký tự ?**

Bài tập

1. **Số nhị phân 8 bit (11001100) tương ứng với số nguyên thập phân có dấu là bao nhiêu trong các phép biểu diễn số có dấu sau đây:**
   - **Dấu và trị tuyệt đối.**
   - **Số bù 1.**
   - **Số bù 2.**
   - **Số thừa K=128**
2. **Cho số nguyên -120, biểu diễn số nguyên dưới dạng nhị phân 8 bit trong các phép biểu diễn sau:**
   - **Dấu và trị tuyệt đối.**
   - **Số bù 1.**
   - **Số bù 2.**
   - **Số thừa K=128**
3. **Đổi các số sau đây:**
   - **011011 → số thập phân.**
   - **1500 → Số nhị phân.**
   - **55.875 → số nhị phân.**
   - **-2005 → số nhị phân 16 bits.**
4. **Biểu diễn các số thực dưới đây bằng số có dấu chấm động chính xác đơn 32 bit.**
   - **31.75**
   - **-371.675**
   - **1250.6875**
   - **-1457.125**

(Chú ý: Sinh viên nộp bài bằng file MS-Word, tên file: MSSV.doc)

Đổi số nhị phân có dấu sang thập phân

d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
1	0	0	1	0	0	1	0
128	64	32	16	8	4	2	1

- **Dấu và trị tuyệt đối:** công thức $$N = (-1)^{d_{n-1}} \sum_{i=0}^{n-2} d_i 2^i$$

N = $(-1)^1$x $(0x2^0+1x2^1+0x2^2+0x2^3+1x2^4+0x2^5+0x2^6)$

= (-1) x (2 +16) = - 18

- **Số bù 1:**

	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
	1	**0**	**0**	**1**	**0**	**0**	**1**	**0**
Bù 1	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**
	N = - 109							

- **Số bù 2:** công thức $$N = -d_{n-1}2^{n-1} + \sum_{i=0}^{n-2} \mathrm{d_i} 2^i$$

N = - $1x2^7$ + $(0x2^0+1x2^1+0x2^2+0x2^3+1x2^4+0x2^5+0x2^6)$

N = - 128 + 18 = - 110

Đổi số thập phân ra số nhị phân

- Phần nguyên:

12/2 = 6	0
6/2 = 3	0
3/2 = 1	1
1/2 = 0	1

12 = 1100

- Phần lẻ:

0.125 x 2 = 0.25	0
0.250 x 2 = 0.5	0
0.5 x 2 = 1.0	1

0.125 = 001

12.125 = 1100.001

CANTHO UNIVERSITY

Đổi số nguyên có dấu ra số nhị phân

- **Dấu và trị tuyệt đối:**

125/2 = 62   1
62/2 = 31   0
31/2 = 15   1
15/2 = 7   1
7/2 = 3   1
3/2 = 1   1
1/2 =0   1

125 = 01111101

-125 = 11111101

- **Số bù 1:**

125 ⇔ 01111101

- 125 ⇔ 10000010

- **Số bù 2:**

125 ⇔	01111101
Bù 1	10000010
+	00000001
-125 =	10000011

Biểu diễn số thập phân theo kiểu số thực dấu chấm động 32 bit: -378.125

1. Đổi trị tuyệt đối ra số nhị phân

Phần nguyên:

378/2 = 189	0	d_0
189/2 = 94	1	.
94/2 = 47	0	.
47/2 = 23	1	.
23/2 = 11	1	.
11/2 = 5	1	.
5/2 = 2	1	.
2/2 = 1	0	.
1/2 = 0	1	d_{0-1}

378 ⇔ 101111010

Phần lẻ:

0.125x2 = 0.25	0	d_{-1}
0.250x2 = 0.5	0	.
0.500x2 = 1.0	1	d_{-m}

0.125 ⇔ 001

2. **Chuẩn hoá:**

-378.125_{10}= -101111010.001_2

-101111010.001= $-1.01111010001 \times 2^8$

$(-1)^s \times 1,f \times 2^{E-127} = -1.01111010001 \times 2^8$

⇨ **S = 1**

⇨ **F = 01111010001**

⇨ **E =127+8=135 = 10000111**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
S	**E=10000111**								**F=01111010001**																						