

# TIN HỌC CƠ SỞ

(45 tiết)

Tài liệu tham khảo chính:

- 1.- Tin học cơ sở - Hoàng Chí Thành. NXB ĐHQG HN. Sinh viên phải có!, viết về WinXP
- 2.- Tin học cơ sở - Nguyễn Kim Tuấn (chủ biên) & Phương Lan (hiệu đính), viết về Win 7, Office 7

## Phần 1. ĐẠI CƯƠNG VỀ TIN HỌC

### Chương 1. Thông tin và xử lý thông tin

#### 1.1. Khái niệm về thông tin (xem sách).

- **Thông tin** (Information) là các điều mà người ta có thể hiểu biết được. Thông tin có  **nhiều dạng**: văn bản, hình ảnh, âm thanh, mùi vị, ...
- Thông tin thường được  **lưu trên** giấy, trên phim, trên màn hình, trên băng từ đĩa từ, thẻ nhớ,...
- **Tin học** là ngành khoa học nghiên cứu về việc  **xử lý** (process) thông tin bằng  **máy tính điện tử** (MTĐT) hay  **PC** (personal computer).
- Thông tin để đưa vào máy tính gọi là  **đầu vào** (input) để xử lý, rồi đưa tới  **đầu ra** (output).
- Tuy nhiên, vì máy tính chỉ  **sử dụng điện, quang hay từ** , chỉ có 2 trạng thái ( **bit** ): 0 là không có, 1 là có (điện, quang hay từ), nên cần phải  **mã hóa**  thành dạng máy tính tiếp thu và xử lý được ở dạng các bit 0 hay 1 đó. Thông tin được mã hóa gọi là  **dữ liệu** (data). Việc làm đó gọi là  **mã hóa** (encoding) thông tin. Sau khi xử lý các dữ liệu dạng bit đó được dữ liệu mới vẫn ở dạng bit, rồi  **giải mã** (decoding) ra cho thông tin.
- **Ví dụ** : Cộng 2 với 3, người xử lý cho kết quả là 5. Nhưng với máy, khi ta gõ số 2 từ bàn phím thì vào máy phải được mã hóa thành 01, 3 thành 11. máy xử lý 01+11=101, dữ liệu thu được là 101, giải mã ra màn hình là 5.

#### 1.2. Mã hóa và giải mã

- Dữ liệu dùng trong máy là dạng mã hóa của thông tin. Máy tính không hiểu trực tiếp được thông tin.
- **Đơn vị thông tin nhỏ nhất**  là các bit 0 và 1.
- Để đưa thông tin vào máy sau khi đã mã hóa. Ngày xưa thường  **dùng bìa hay băng giấy đục lỗ**  và các tế bào quang điện. Ngày nay, cũng  **dùng tia laser** , tế bào quang điện và các mạch điện tử  **hấp thụ từ** ,...
- Nếu dùng nhóm 2 bit, ta có thể mã hóa được 4 trạng thái thông tin: 00,01,10,11.
- Nếu dùng nhóm 3 bit, ta có thể mã hóa được 8 trạng thái thông tin: 000,001,010,011,100,101,110,111...
- **Nếu dùng nhóm n bit, ta có thể mã hóa được  $2^n$  trạng thái.**
- Khi thiết kế máy tính, mới đầu, ta dùng bộ  **kí tự** (characters) tiêu chuẩn  **ASCII** (American Standard Codes for Information Interchange) gồm  **256 kí tự** (=  $2^8$ ), nên người ta dùng các nhóm 8 bit để có thể mã hóa đủ 256 kí tự. Bộ kí tự này gồm các kí hiệu cơ bản, chữ cái, chữ số, các dấu và  **kí tự điều khiển** (control character, chỉ làm nhiệm vụ điều khiển, không hiện hình, như các phím mũi tên, Esc, Enter,.... ). Số thứ tự các kí tự gọi là  **mã thập phân, từ 0 đến 255. Ví dụ**  chữ 'A' có mã thập phân là 65
- **Mọi thông tin được mã hóa thành dãy bit có số lượng chia hết cho 8.**
- Từ đó, người ta quy định: Nhóm 8 bit gọi tắt là 1  **Byte** , tương ứng 1-1 với một kí tự nếu dùng ASCII.
- Ngày nay có  **bộ kí tự Quốc tế** (Unicode)  **gồm 65536 kí tự** (=  $2^{16}$ ). Unicodes phải dùng 16 bit, tức 2 Bytes để mã hóa 1 kí tự.
- Các đơn vị dẫn suất khác để chỉ độ lớn của thông tin:

$ 2^{10} $ Byte =	1024 Byte gọi là	1 KB, <b> ki lô Byte </b> ,	$ \approx $ 1 trang sách.
$ 2^{10} $ KB =	1024 KB gọi là	1 MB, <b> mega byte </b> ,	$ \approx $ 1 ngàn trang sách.
$ 2^{10} $ MB =	1024 MB gọi là	1 GB, <b> giga byte </b> ,	$ \approx $ 1 triệu trang sách.
$ 2^{10} $ GB =	1024 GB gọi là	1 TB, <b> têta byte </b> ,	$ \approx $ 1 tỷ trang sách...
- **Chú ý: K = 1024, k=1000**  lần.  **Ví dụ** :
- **Modem dial-up**  có tốc độ truy nhập mạng tối đa là 56kbps (56000 bits per second). Mã hóa thông tin thành dạng bit giống như thành dạng tách-tè để đánh manip trong thời kỳ kháng chiến chống Pháp. Một cô điện báo viên đánh giới lăm được hơn 300 bit sau 1 phút...
- **Modem ADSL**  có tốc độ tối đa là 100Mbps. Trong thực tế Việt Nam hiện nay mới đạt được con số  $\leq 10$ Mbps. Sắp tới thay thành đường cáp quang triển vọng nhanh hơn nhiều
- Thuật ngữ Modem là ghép nối giữ  **Modulation** (Biến tín hiệu từ đường truyền điện thoại thành tín hiệu ký thuật số để máy sử dụng được) với  **DeModulation** (Biến tín hiệu từ máy tính thành tín hiệu truyền được trên đường dây điện thoại).

## Chương 2. Đại cương về máy tính điện tử

### 2.1. Kiến trúc MTĐT

#### 2.1.1 Chức năng của MTĐT

- MTĐT phải đảm nhiệm các chức năng sau:
  - **Nhập** dữ liệu, **Lưu trữ** dữ liệu, **Xử lý** dữ liệu, **Xuất** dữ liệu, và **Điều khiển các thiết bị vào/ra**.

#### 2.1.2 Cấu trúc chung

- MTĐT bao gồm các bộ phận chính ghép nối với nhau như sơ đồ (SGK): Sau đây ta nghiên cứu sơ lược các nhóm thiết bị sau:
  - **Bộ xử lý trung tâm** (CPU),
  - **Bộ nhớ** (Memory) gồm **trong** (Main mory) và **ngoài** (Auxiliary Storage),
  - **Bộ điều khiển** (Keyboard, Mouse),
  - **Các thiết bị vào/ra** (Input/Output devices).

#### 1) Bộ nhớ (Memory)

Là **dãy các ô nhớ** cỡ 1 Byte gọi là một **từ máy**. Các ô nhớ này **được đánh số từ 0 đến hết...** gọi là **địa chỉ** của ô nhớ. **Hiệu suất** làm việc của bộ nhớ **phụ thuộc vào dung lượng** (do bằng Byte) và **tốc độ** (do bằng **hetz** (Hz)). *Ví dụ:* tốc độ 800 MHz, tức là trong 1 giây, khoảng 800 triệu đợt ghi vào hay lấy.

- **Bộ nhớ trong** dung lượng nhỏ nhưng tốc độ truy cập nhanh, nó gồm 2 phần:
  - **ROM** (Read-Only Memory) là bộ nhớ **chỉ đọc ra** được, mà không ai sửa chữa hay xóa được các thông tin trong đó, trừ nhà sản xuất chế tác lại, nó ví như bia đá, hay cuốn sách. Khi mất điện dữ liệu của nó vẫn còn!
  - **RAM** (Random Access Memory) là bộ nhớ **truy cập ngẫu nhiên**, có thể đưa dữ liệu vào hay lấy dữ liệu ra từ bất cứ địa chỉ nào mà không cần phải tua lần lượt. Việc này cũng ví như gửi đồ ở các siêu thị, muốn gửi vào ô nào cũng được... **Bảng từ là một ví dụ Bộ nhớ truy cập tuần tự** (succesif hay sequential memory device), trái ngược với ngẫu nhiên (random).
  - **Dữ liệu trên RAM chỉ là tạm thời, sẽ mất đi khi mất điện**.
- **Bộ nhớ ngoài** gồm các thiết bị lưu trữ như các đĩa, thẻ nhớ, băng từ. Nó có dung lượng lớn nhưng tốc độ truy cập chậm.
  - **Đĩa cứng** HD (hard disk), nằm cố định trong một ổ chắc chắn trong thân máy. Dung lượng của đĩa cứng có thể tới hàng trăm GB. Hiện nay người ta còn sản xuất được các ổ đĩa cứng cắm ngoài, dung lượng cũng rất lớn, có loại tới 1 TB. Trên đĩa cứng cổ điển có các hạt từ rất nhỏ, sắp xếp thành các vòng tròn đồng tâm, và cũng được chia thành từng cung, kiểu như ở sân vận động! Nó phải quay tít liên tục để đầu đọc quét các dữ liệu. Đĩa hiện đại dùng công nghệ đặc biệt, không tốn nguồn điện, không cần quay mà vẫn truy cập được dữ liệu. Tốc độ truy cập nhanh nhưng vẫn kém hơn so với bộ nhớ trong. Một đĩa cứng nên chia nhỏ thành nhiều **phân vùng** (partition), có bộ phận đọc/ghi riêng tạo thành một **ổ đĩa**, HDD (hard disk drive).
  - **Đĩa mềm** FD (floppy disk), dung lượng thông thường là 1.44 MB, quá nhỏ, nên hiện nay rất ít người dùng đến nữa. Nơi đọc ghi đĩa mềm gọi là **ổ đĩa mềm**, FDD (floppy disk drive).
  - **Đĩa quang** CD (compact disk), còn gọi là CD, trên mặt đĩa có các hạt phản quang hay không phản quang tương ứng với các bit 1 hay 0. Khi tia laser chiếu vào nó phản xạ sang tế bào quang điện để nhận lấy kết quả và đưa vào bộ nhớ. Đĩa loại này có dung lượng **740 MB**. Có **nhiều loại CD** như: **VCD**, **CD-R** dùng để ghi vào được hay **CD-RW** cũng để ghi vào – xóa đi – ghi lại.
  - **Đĩa quang video** DVD (digital video disk) kỹ thuật cao hơn, dung lượng lên tới **4.7 GB** (hơn 6 lần đĩa CD). Tương tự như CD, cũng Có **nhiều loại DVD** như: **DVD-R** dùng để ghi vào, **DVD-RW** cũng để ghi vào – xóa đi – ghi lại.
  - **Thẻ nhớ** (mestick) rất hay dùng cho máy ảnh hay quay phim, các **USB-Flash** hay các băng từ... có các dung lượng khác nhau...
  - *Chú ý: đĩa ≠ ổ đĩa* (như chim và tổ chim)! **Drive ≠ Driver**

#### 2) Bộ xử lý trung tâm (CPU = Central Processing Unit)

- Là **thiết bị chính của MTĐT**, nơi thực hiện các lệnh do chương trình từ bộ nhớ đưa đến. Giá trị của CPU **phụ thuộc vào tốc độ** (speed) xử lý. Tốc độ đó đo bằng Hz, là số các **thao tác** (operation) cơ bản (cũng gọi là phép toán cơ bản) thực hiện được trong một giây. Một lệnh có thể bao gồm nhiều thao tác! *Ví dụ:* Pentium IV có tốc độ 2.4GHz, tức là trong 1 giây có thể thực hiện được  $2,4 \times 10^9$  phép toán, tức là gần 2,4 tỷ thao tác cơ bản. Hiện nay **Core i7** đang được đánh giá là nhanh nhất.
- Trong CPU có 2 thành phần chính: **Đơn vị điều khiển**, CU (Control Unit) và **Đơn vị Số học và Logic**, ALU (Arithmetic-Logic Unit) thực hiện các phép toán số học và logic.
- Ngoài ra còn có **Bộ nhớ ẩn** (Cache), **các Thanh ghi** (Registers) làm việc trực tiếp ở trong CPU phục vụ quá trình thực hiện lệnh.



- Ví dụ: Gõ chữ 'A' có mã 8 bit là 01000001 chạy trên 8 đường dây song song rất nhỏ vào máy. Nếu bàn phím có đầu cắm USB 8 luồng bit kia đã được tổ chức chuyển sang nối tiếp để vào máy! Qua cổng rồi, vào máy chúng chuyển thành 8 bit chạy song song trên các mạch tích rất nhỏ...
- **Con chuột** (mouse) cũng là một thiết bị điều khiển. Có thể có chuột bi, hay chuột quang, có loại có dây hay loại không dây. Tuy nhiên, chức năng của nó là vị trí xác định trên màn hình bằng hình ảnh của mũi tên chuột theo các thao tác (nhấp đơn, đúp, nhấp phải hay kéo thả...) để máy xử lý.
- Chuột: bình thường có **2 phím cơ bản**: là **phím trái** và **phím phải** với nguyên tắc sau:
 

<b>Nhấp trái</b> (left-click)	để <b>chọn</b> đối tượng mà nó đang chỉ vào.
<b>Nhấp phải</b> (right-click)	để <b>hiện thực đơn</b> (menu, bảng chọn) áp dụng cho đối tượng.
<b>Nhấp đúp</b> (double-click)	để <b>kích hoạt</b> (mở thư mục/chương trình).
<b>Kéo thả</b> (drag-and-drop)	để <b>di chuyển đối tượng</b> (tệp/thư mục/nền/đường biên,...).

#### 4) Các thiết bị vào/ra (Input devices, Output devices)

- Là các **thiết bị dùng để đưa dữ liệu từ ngoài vào máy hoặc/và đưa dữ liệu từ trong máy ra ngoài**.
- Thiết bị vào/ra thường dùng hiện nay là: Bàn phím (đưa kí tự từ ngoài vào máy), các ổ đĩa cứng, ổ đĩa quang (có thể chỉ đưa vào), đầu đọc thẻ nhớ, máy quét ảnh, màn hình, máy in, máy chiếu, webcam, modem, bộ loa...  
*Chú ý: Bàn phím vừa là thiết bị điều khiển vừa là thiết bị vào.*
- **Ổ đĩa cứng HDD** (hard disk drive) là hộp chứa đĩa cứng (HD) và tiến hành việc đọc/ghi dữ liệu. Bình thường thì HDD để phía trong **vỏ máy** (case), nhưng còn có ổ đĩa cứng **lưu động** (removable) cắm ngoài vào cổng USB rất tiện hơn, dung lượng của nó có thể tới 500GB hay hơn nữa.
- **Một đĩa cứng HD** (hard disk) mua về nên phân chia thành nhiều phân vùng nhỏ (partition) để MTĐT dễ bề quản lý: C: D: E:, F:,... mỗi vùng coi như một ổ đĩa cứng có tên C:, D:,...!
- **Ổ đọc đĩa quang CD** (rom) chỉ đọc các loại đĩa CD, VCD, mà không ghi được.
- **Ổ đọc/ghi đĩa quang CD-R** (recordable) ghi được một lượt thôi. Trên đĩa quang, các điểm đều phản quang cả, đầu ghi sẽ làm **mất tính phản quang** (còn gọi là đốt, bum) của các **điểm nào sẽ tương ứng với bit 0**, còn **điểm nào sẽ ứng với các bit 1 thì để nguyên cho nó phản quang**. Khi ghi ta thường chú ý 2 chế độ ghi nhiều lần (multi-section) cho đến khi đầy hoặc ghi một lần (one-section) mặc dù chưa đầy. **Ổ loại này cũng dùng để đọc được các loại đĩa CD, VCD.**
- **Ổ ghi đĩa quang CD-RW** (rewritable), ghi rồi lại xóa trắng để ghi tiếp. Đặc biệt **chú ý nên chọn ghi với tốc độ chậm thì ghi được sâu hơn và lưu trữ được lâu dài hơn**. Tuy nhiên, tuổi thọ của dữ liệu ghi trên đĩa quang cũng chỉ được 2-3 năm thôi. Cho nên ta cũng **không nên kỳ vọng vào chúng để lưu trữ dữ liệu lâu dài!**
- Tương tự ta cũng có **các đầu đọc/ghi đĩa DVD-ROM, DVD-R, DVD-RW....**
- **Ổ đĩa DVD-RW tuy đắt nhưng đọc được các loại đĩa CD và DVD khác, ghi được các loại đĩa CD-R, CD-RW, DVD-R, DVD-RW.** Đối với đĩa **USB** ta chỉ nên dùng để **lưu trữ tạm thời và vận chuyển dữ liệu**. Tuy nhiên, khi **cắm USB vào máy phải diệt virus trước**. **Không cắm USB trước khi khởi động máy** vì virus truyền vào máy chiếm quyền điều hành máy... trước khi bị phát hiện!
- **Thiết bị ra chuẩn là màn hình** (screen), mà trên đó có các **điểm ảnh** (pixel) sẽ hiện các màu do CPU đưa đến. Nó có **độ phân giải** (resolution) bằng số cột x số hàng của bảng các điểm ảnh. Độ phân giải càng lớn cao thì hình càng mịn. Mỗi điểm ảnh mang **thông tin về màu sắc** của nó, tùy thuộc **chất lượng màu** (color quality). **Hệ điều hành DOS chỉ có 16 màu**. Hệ điều hành **Windows tạo ra được chất lượng 8 bit, 16 bit, 24 bit hay 32 bit**. Ví dụ: Với chất lượng màu 16 bit, mỗi điểm ảnh chiếm mất 2B của bộ nhớ màn hình. **Màn hình chất lượng tốt nếu nó có độ phân giải cao, chất lượng màu lớn và bộ nhớ màn hình lớn**.
- *Chú ý:* Nếu đem một ảnh chất lượng tốt xuất ra trên một màn hình chất lượng màu kém thì ảnh thường bị rạn, rỗ vì có những điểm mà màu không thể hiện ra được. Ta cần phân biệt: **Monitor là cái màn hình**, còn **Screen là màn hiển hình**. **Monitor là phần cứng**. **Screen là thuộc phần mềm**. Ví như cái **máy vô tuyến truyền hình** (televisor) và **chương trình tivi** (television).
- **Máy in có 3 loại chính: Máy in kim** cho ra sản phẩm là các hình nét không liền mà lấm chấm, thường dùng để in hoa đơn các loại cho đỡ tốn mực. **Máy in mực**, thường xuyên hết mực phải đổ mực, nét liền hơn in kim, loại này rất thông dụng vì rẻ, nhưng mực lại đắt. **Máy in laser** rất cao cấp và đắt tiền hơn, nhất là laser màu, nên ít thông dụng cho lớp người dùng bình dân.

#### Muốn có một máy tính tốt cần tham khảo ý kiến các chuyên gia. Bình thường nên có:

- Bộ xử lý trung tâm CPU nhanh (đo bằng Hz), thường là 2.8 – 3.6 GHz hay cao hơn,
- Bộ nhớ RAM lớn (khoảng 1GB hay lớn hơn) với tốc độ truy cập nhanh khoảng 1 GHz hoặc nhanh hơn.
- Đĩa cứng HD lớn chứa được nhiều dữ liệu (đo bằng Byte), thường 250 GB hay lớn hơn, Về phải phân ra làm nhiều ổ nhỏ gọi là phân vùng: C:, D:, E:, F:,...
- Màn hình độ phân giải cao cùng chất lượng màu lớn (tối thiểu 16 bit), nên mua màn hình tinh thể lỏng (LCD) đỡ hại mắt và chiếm ít diện tích và cũng là đỡ tốn điện hơn!
- Chọn thương hiệu có uy tín và mọi thứ nên đồng bộ để phát huy hết khả năng của các thiết bị!
- Nên mua máy có ổ ghi DVD-RW

## 2.2. Nguyên lý Von Neumann

- **John Von Neumann** (28/12/1903 – 8/2/1957) là một nhà toán học người Mỹ gốc Hungary và là một nhà bác học thông thạo nhiều lĩnh vực đã đóng góp vào vật lý lượng tử, giải tích hàm, lý thuyết tập hợp, kinh tế, khoa học máy tính, giải tích số, động lực học chất lỏng, thống kê và nhiều lĩnh vực toán học khác.
- **Kiến trúc máy tính nêu ở phần trên** lần đầu tiên được John Von Neumann mô tả, nên cũng được gọi là **kiến trúc Von Neumann** (về mặt vật lý). Ông cũng đưa ra nguyên lý hoạt động cũng gọi là **nguyên lý Von Neumann** như sau:
- MTĐT phải được **điều khiển bằng chương trình lưu trữ sẵn trong bộ nhớ** của nó, theo một kịch bản mà người ta đã chuẩn bị sẵn cho nó
- MTĐT **truy cập tới dữ liệu thông qua địa chỉ**. Dữ liệu ở đây gồm dữ liệu vào, trung gian hay dữ liệu ra, cùng với chương trình. Các mã lệnh của chương trình cũng được đưa vào bộ nhớ trong.
- Nguyên lý này **đảm bảo tính mềm dẻo tổng quát trong việc xử lý thông tin**, người lập trình viết yêu cầu một cách tổng quát đến các dữ liệu nằm ở đâu mà không cần biết giá trị cụ thể của chúng.

### Chú ý:

Ở giai đoạn thử nghiệm theo các cấu trúc và nguyên lý mới, máy tính lượng tử, máy tính sinh học đã cho một số kết quả khá quan!

## 2.3. Cơ sở số học và logic của MTĐT

### 2.3.1. Hệ đếm trong MTĐT

#### 1) Định nghĩa hệ đếm (xem sách)

#### 2) Các hệ đếm thường dùng trong MTĐT

**Hệ thập phân (DECimal) gồm 10 chữ số 0, 1, ... và 9. Đếm bình thường.**

- Dạng khai triển một **số tự nhiên D** có **n chữ số** là một **Đa thức  $P_n(10)$** , bậc n với **các hệ số lần lượt là các số từ 0 đến 9**.
- Các phép tính số học là bình thường:  $1 + 1 = 2$ , không có gì xáo trộn cả.

**Hệ nhị phân (BINary) gồm 2 chữ số 0 và 1.**

- Đếm 0, 1, 10, 11, 100, 101, 110, 111, 1000, ...
- Để đổi ra số hệ DEC ta khai triển **một số tự nhiên D** có **n chữ số** thành **Đa thức  $P_n(2)$** , bậc n với **các hệ số lần lượt là 0 và 1**.
- Để ám chỉ đây là số nhị phân ta viết thêm chỉ số 2.
- Ví dụ:  $1101_2 = 1.2^3 + 1.2^2 + 0.2 + 1 = 13$  (ở hệ DEC).
- **Phép nhân bình thường, nhưng phép cộng có khác:  $1_2 + 1_2 = 10_2$ .**

**Hệ bát phân (OCTal) gồm 8 chữ số 0, 1, ... và 7. (Xem sách)**

**Hệ thập lục phân (HEXadecimal) gồm 16 chữ số 0, 1, ..., 9, A, B, C, D, E và F.**

- Mặc dù có thể dùng chữ in thường nhưng ta nên dùng chữ IN HOA cho chuẩn.
- Đếm 0, 1, ..., 9, A, ..., F, 10, 11, ..., FF, 100, 101, ..., FFF, 1000, ...
- Để ám chỉ đây là số hệ HEX ta viết thêm chữ \$ hoặc # phía trước chữ số đầu tiên hoặc chỉ số 16 hay chữ h phía sau. Ví dụ: \$1F, #1F, 1F16, 1Fh, #1f, \$1f là đều như nhau.
- Để đổi ra số hệ DEC ta khai triển **một số tự nhiên D** có **n chữ số** thành **Đa thức  $P_n(16)$** , bậc n và **các hệ số lần lượt là các chữ số từ 0 đến F**.
- Ví dụ:  
 $\$1F = \$1.\$10 + \$F = 1.16 + 15 = 31$ ,  
 $\$FF = \$F.\$10 + \$F = 15.16 + 15 = 255$ .
- Các phép tính cũng khác cũng có nhiều cái đặc biệt: ... chẳng hạn:  
 $\$1 + \$9 = \$A$ ,  
 $\$1 + \$F = \$10, \dots$

**Chuyển đổi số tự nhiên từ hệ đếm khác ra hệ DEC:**

- Phương pháp dễ nhớ để đổi một số các hệ BIN, OCT hay HEX ra DEC là dùng **lược đồ Horner** để tính giá trị của đa thức  $P(x)$  theo triển khai của Horner.
- Ví dụ: Để tính:
- $P(x) = ax^3 + bx^2 + cx + d$ , Horner không tính từng đơn thức rồi cộng lại mà như sau:

- $P(x) = ((ax + b)x + c)x + d$ , theo lược đồ cũng mang tên Horner:

	a	b	c	d
Đối số = x	$P = a$	$P = xP + b$	$P = xP + c$	$P = xP + d$

- Ví dụ:
- $1101_2 = ?$

	1	1	0	1
Đối số = 2	1	3	6	13 (kết quả)

- Ví dụ:
- $\$2A3C = ?$

	2	10	3	12
Đối số = 16	2	42	675	10812 (kết quả)

### Chuyển đổi số tự nhiên từ hệ DEC ra các hệ đếm khác:

- Phương pháp dễ nhớ để đổi một số từ hệ DEC ra các hệ khác vẫn từ khai triển của Horner: Bài toán ví dụ:
- Biết số P sẽ có dạng khai triển  $ax^3 + bx^2 + cx + d$ , hãy tính các hệ số a, b, c và d (theo P và x).
- Ta có:
  - $P = ((ax + b)x + c)x + d$ .
  - P chia cho x được  $((ax + b)x + c)$  dư d. Đặt P mới là  $(ax + b)x + c$ .
  - P chia cho x được  $(ax + b)$  dư c. Đặt P mới là  $ax + b$ .
  - P chia cho x được a dư b. Đặt P mới là a.
  - P chia cho x được 0 dư a. (Đề ý rằng a, b, c, d đều phải bé hơn x).
  - Như vậy, ta lần lượt thu được d, c, b và a.
- Ví dụ:  $13 = \overline{abcd}_2$  (hệ đếm 2). Tức là tính a, b, c và d?
- Ta có: (Xem Bìa đọc thêm ở Sách giáo khoa Đại số lớp 10, nâng cao)
  - 13 chia cho 2 được 6 dư 1.
  - 6 chia cho 2 được 3 dư 0.
  - 3 chia cho 2 được 1 dư 1.
  - 1 chia cho 2 được 0 dư 1.
  - Kết quả là  $1101_2$ . (Viết từ dưới lên!).
- (Xem bài đọc thêm sách giáo khoa đại số 10).
- **Chú ý:**  
Đặc biệt chuyển đổi giữa BIN và HEX còn có thể làm như Sách: Coi mỗi chữ số của hệ HEX như dãy 4 bit, thì việc chuyển đổi này dễ dàng thực hiện được. Yêu cầu nhớ cách đổi đó (Xem sách, trang 11).

### 2.3.2. Biểu diễn thông tin trong MTĐT (tức là mã hóa thông tin và giải mã dữ liệu thành thông tin)

#### 1) Biểu diễn dữ liệu số:

- **Số tự nhiên nhỏ** (gọi là số kiểu Byte) thành dạng 8 bit từ 00000000 đến 11111111 tức là giá trị là 0 đến 255. Trong khi làm toán vượt quá 8 bit thì bit thứ 9 bị bỏ đi.
- Ví dụ:  $\boxed{255+1=0}$ . Muốn chính xác phải để kết quả ở kiểu lớn hơn.
- **Số tự nhiên lớn** (gọi là số kiểu Word) thành dạng 16 bit (2 Bytes) từ 0000000000000000 đến 1111111111111111, tức là giá trị là 0 đến 65535. Trong khi làm toán vượt quá 16 bit thì bit thứ 17 bị bỏ đi.
- Ví dụ:  $\boxed{65535+1=0}$ . Muốn chính xác phải để kết quả ở kiểu lớn hơn.
- **Số nguyên ngắn** (gọi là số kiểu ShortInt) thành dạng 8 bit (1 Byte): Bit đầu là bit dấu (0 là +, 1 là -), 7 bit sau là các bit số. Giá trị số kiểu này từ 10000000 đến 01111111 tức là giá trị là -128 đến 127. Trong khi làm toán số vượt quá 127 thì quay về -128.
- Ví dụ:  $01111111_2 + 1_2 = 10000000_2$  tức  $\boxed{127+1=-128}$ .
- Muốn chính xác phải để kết quả ở kiểu lớn hơn.
- **Số nguyên** (gọi là số kiểu Integer) thành dạng 16 bit (2 Bytes): Bit đầu là bit dấu (0 là +, 1 là -), 15 bit sau là các bit số. Giá trị từ số kiểu này là từ -32768 đến 32767. Trong khi làm toán số vượt quá 32767 thì quay về -32768.

- Ví dụ:  $32767+1=-32768$  . Muốn chính xác phải để kết quả ở kiểu lớn hơn.
- **Số nguyên dài** (gọi là số kiểu LongInt) thành dạng 32 bit (4 Bytes): Bit đầu là bit dấu (0 là +, 1 là -), 31 bit sau là các bit số. Giá trị từ -2147483648 đến 2147483647. Trong khi làm toán số vượt quá 2147483647 thì quay về -2147483648.
- Ví dụ:  $2147483647+1=-2147483648$  . Muốn chính xác phải để kết quả ở kiểu lớn hơn.
- **Số thực** (gọi là kiểu Real) viết thành số thập phân dấu phẩy động (dạng khoa học, ví dụ:  $3,17 \times 10^3$  hoặc  $4,3 \times 10^{-3}$  ) hay dạng dấu phẩy tĩnh (dạng thông thường, ví dụ: 345,7645). Chú ý: Trong tin học, ùng dấu chấm (.) là dấu phẩy thập phân, và dấu sao (\*) làm dấu nhân.
- Cách mã hóa số thực là viết dưới dạng khoa học chuẩn là gồm:
  - $\pm$ , 0. <số tự nhiên>\*10<số nguyên>.
  - Ví dụ: 3141592 viết thành  $0.3141592*10^8$ .
  - Do vậy muốn mã hóa chỉ cần mã hóa các số tự nhiên và số nguyên đó.

Nhận xét:

- **Mỗi dạng thông tin nào biểu diễn được qua các số thì mã hóa được**, rồi sau lại **giải mã được**.
- Ví dụ:*
  - Số phức  $z = a + bi$ , a và b là số thực, vậy cũng sẽ mã hóa được.
  - Các nốt nhạc biểu diễn được qua tần số (số tự nhiên) và thời gian (đo bằng mili giây, cũng là số tự nhiên), nên không khó khăn gì để mã hóa chúng!

## 2) Ký tự và xâu ký tự

- Bảng mã ASCII gồm 256 ký tự: Mỗi ký tự có số thứ tự từ 0 đến 255 (gọi là mã thập phân) nên dùng 8 bit (1 Byte) để mã hóa ký tự (gọi là mã nhị phân) của nó.
- Một số mã đặc biệt:
  - Ký tự 'A' = ký tự số 65 và do vậy mã nhị phân là 01000001. Từ đó tính tiếp.
  - Ký tự 'a' = ký tự số 97 và do vậy mã nhị phân là 01100001. Từ đó tính tiếp.
  - Ký tự '0' = ký tự số 48 và do vậy mã nhị phân là 00110000. Từ đó tính tiếp.
  - Ký tự **Esc** = ký tự số 27 và do vậy mã nhị phân là 00011011.
  - Ký tự **đưa con trỏ xuống dòng dưới** là = ký tự số 10, mã nhị phân là 00001010.
  - Ký tự **đưa con trỏ về đầu dòng** hiện thời là = ký tự số 13, mã nhị phân là 00001101.
- **Ghép ký tự số 10 và 13 lại thành phím Enter.**
- **Dấu cách** = ký tự số 32, mã nhị phân là 00100000.
- **BackSpace** = ký tự số 8, mã nhị phân là 00001000.
- **Tab** = ký tự số 9, mã nhị phân là 00001001.
- **Xâu ký tự là dãy các ký tự** nên cũng mã hóa thập phân được thành dãy các số tự nhiên, rồi thành dãy các bit. Rồi khi giải mã thì theo chiều ngược lại!
- *Ví dụ:* Với ASCII thì:
  - 'TIN'  $\Leftrightarrow$  84,73,78  $\Leftrightarrow$  01010100-01001001-01001110
  - '2010'  $\Leftrightarrow$  50,48,49,48  $\Leftrightarrow$  00110010-00110000-00110001-00110000.
  - Dấu - ở đây chỉ viết để dễ nhìn...
- Mỗi sinh viên phải tập mã hóa nhiều cho quen, chẳng hạn tên mình, ngày tháng năm sinh của mình.

## Bảng mã Unicode gồm 65536 ký tự

- Mỗi ký tự có số thứ tự từ 0 đến 65535 (gọi là mã thập phân) nên dùng 16 bit (2 Byte) để mã hóa ký tự (gọi là mã nhị phân) của nó.

## 3) Màu sắc

- Ban đầu máy tính có thể **cho 16 màu cơ bản**, do đó ta nên dùng hệ đếm HEX để mã hóa hay DEC tương ứng. Tất nhiên là từ đó mã hóa ngay sang 4 bit của hệ BIN được ngon lành.
- Nhớ ở hệ HEX dễ dàng hơn:

\$0 = 0	đen (black)	\$5 = 5	tím tối (magenta)
\$1 = 1	xanh tối (blue)	\$6 = 6	nâu (brown)
\$2 = 2	lá tối (green)	\$7 = 7	ghi (light-gray)
\$3 = 3	trời tối (cyan)	\$8 = 8	xám (dark-gray)
\$4 = 4	đỏ tối (red)	\$9 = 9	xanh sáng light-blue)

\$A = 10	lá sáng (light-green)	\$D = 13	tím sáng (light-magenta)
\$B = 11	trời sáng (light-cyan)	\$E = 14	vàng (yellow)
\$C = 12	đỏ sáng (light-red)	\$F = 15	trắng (white)

- Trong hệ **điều hành phiên bản thấp DOS**, màn hình là màn hình văn bản, thường là 80 cột x 25 dòng, giao của mỗi cột và mỗi dòng là một kí tự, mà màu nền của kí tự và màu của kí tự hợp lại với nhau để mô tả **thuộc tính (attribut) của kí tự** theo quy định như sau:

- Thuộc tính = \$XY, (\$X = màu nền, \$Y = màu chữ).**

Ở đây X và Y là các chữ số của hệ HEX

Quy ước: **Nếu \$X>7 thì màu nền=\$X-8 và chữ thì nhấp nháy.**

Số \$XY chạy từ \$00 đến \$FF, chiếm 1 Byte.

Ví dụ:

Chữ 'A' nền xanh chữ trắng có mã=65, thuộc tính=\$1F.

Chữ 'b' với thuộc tính \$4E có màu nền đỏ, chữ vàng.

Kí tự có mã 49, thuộc tính \$CF là chữ số '1' màu nền là đỏ, màu chữ trắng và nhấp nháy.

- Sau này ở hệ **điều hành phiên bản cao hơn** như Windows, giao của cột và dòng là một **điểm ảnh (pixel)**, thông tin về màu của nó. Số màu trong Windows cũng phong phú hơn nhiều

- Ví dụ:

Hiện nay ít nhất cũng phải là 65536 màu. Màu có mã thập phân từ 0 đến 65535, tức là mã nhị phân từ 0000000000000000<sub>2</sub> đến 1111111111111111<sub>2</sub>. Ta nói rằng chất lượng màu này là loại 16 bit. Cao hơn nữa là 24 bit và 32 bit. Cùng với độ phân giải lớn thì hình ảnh xem rất đẹp mắt, nhưng cũng sẽ tốn bộ nhớ hơn!

- Trong Internet, người ta dùng loại 24 bit cho đỡ tốn bộ nhớ và truyền tải hình ảnh đi được nhanh. Người ta quy định dùng mẫu mã hóa màu là #RRGGBB, ở đây R, G, B là các chữ số của hệ HEX. #RR là độ đậm của màu đỏ (Red), #GG của màu lá (Green) và #BB của màu xanh (Blue).

- Ví dụ:

#102A1F làm mã hóa của màu trộn của độ đỏ = #10, nghĩa là #10/#FF = 16/255 = 6.27%.

#FF0000 là màu đỏ 100%, #0000FF là blue 100%, #FF00FF là trộn màu đỏ với màu xanh cho màu tím 100%, #000000 là màu đen, #FFFFFF là màu trắng,...

Mã nguồn sau đây của trang web đơn giản:

```
<html>
<body>
<font face=arial size=7 color=#FF00FF> Chao cac ban! </font>
</body>
</html>
```

- Viết ra màn hình dòng chữ "Chao cac ban" có cỡ là 7, font kiểu Arial và màu là tím.
- Bạn hãy thay mã màu trên bằng một số khác, lưu mã nguồn lại (Save As) dưới một tên nào đó có đuôi là .htm, chẳng hạn Test.htm và Open nó bằng một trình duyệt quen biết như Internet Explorer (viết tắt là IE) hay FireFox sẽ thấy màu đó như thế nào!

### 2.3.3. Các phép toán logic và mạch điện tử

- Mệnh đề toán học là mệnh đề chỉ có thể nói là đúng hoặc sai, không có cái trung gian hay cũng không đơn thuần là mệnh đề văn học.

- Ví dụ:

- Mệnh đề "Ai đi đâu đấy, hỏi ai?", "Ôi, Hà Nội!", "anh Minh ơi" chỉ là những mệnh đề văn học.

- Các mệnh đề như Mặt trăng sáng hoen mặt trời, 3>2,... là những mệnh đề toán học.

- Mệnh đề x>3 tuy chưa biết đúng sai nhưng khi thay x cụ thể vào sẽ biết được kết quả.

- Cũng như trong đại số, mệnh đề phụ thuộc vào một đối số thay đổi chưa cụ thể gọi là một hàm mệnh đề f(x).

- Giá trị của một mệnh đề toán học được mã hóa là 1 (đúng) và 0 (sai).

- Cho 2 mệnh đề toán học a và b ta có thể tạo ra các mệnh đề toán học mới bằng cách nối chúng lại, chẳng hạn: a AND b, a OR b, a XOR b, IF a THEN b, NOT(A). Cụ thể:

a AND b chỉ đúng khi a và b đều đúng. Kí hiệu  $a \wedge b$ ,  $a.b$ ,  $ab$  cũng được.

a OR b chỉ sai khi a và b đều sai. Kí hiệu  $a \vee b$ , hay  $a+b$  cũng được.

a XOR b chỉ đúng khi a và b không cùng giá trị. Kí hiệu  $a \underline{\vee} b$ .

IF a THEN b chỉ sai khi a đúng và b sai. Kí hiệu  $a \rightarrow b$ ,  $a \Rightarrow b$ ,  $a \Rightarrow b$  đều được.

Not(a) lấy giá trị ngược lại với a, là phủ định của mệnh đề a. Kí hiệu  $\neg a$ , hay  $\overline{a}$

- Từ những phép tính trên ta có thể tạo ra những biểu thức phức tạp hơn.

- Tuy nhiên mọi biểu thức logic đều có thể biểu diễn qua 3 phép toán là AND, OR và NOT.

- Ví dụ:

$$a \vee b = a \overline{b} + \overline{a} b$$

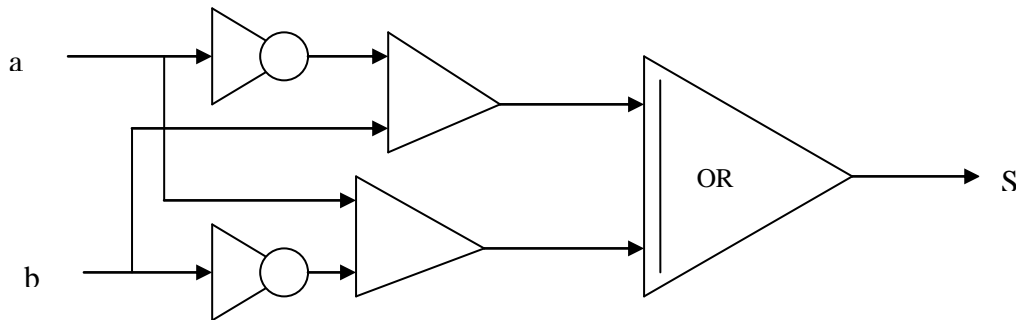
$$a \rightarrow b = \overline{a} + b.$$

- Với phát hiện đó, người ta chế tạo ra các mạch điện tử dùng 3 chip cơ bản AND, OR và NOT, từ đó chế tác ra đủ thứ khác...
- Phương trình hàm mệnh đề. Biết bảng giá trị của một biểu thức mệnh đề, ta có thể tìm ra biểu thức mệnh đề cụ thể đó. Như vậy gọi là giải phương trình hàm mệnh đề.

• Ví dụ

Công tắc a	Công tắc b	Đèn S
0	0	0
0	1	1
1	0	1
1	1	0

- Đèn sáng khi chỉ một trong 2 công tắc được bật. Đó là loại công tắc cầu thang! Theo định nghĩa trên của phép  $\vee$  thì rõ ràng  $S = a \vee b = a \overline{b} + \overline{a} b$ . Như vậy mạch điện tử sẽ là:



- Không chỉ MTĐT mà các thiết bị dân dụng khác như điện thoại di động cũng đều được thiết kế qua các chip điện tử AND, OR và NOT.
- Khi có một con chip không biết công thức của nó, thì bằng phép thử thay đổi giá trị ở tất cả đầu vào để đo kết quả ở mỗi đầu ra. Từ bảng giá trị của biểu thức ở mỗi đầu ra ta biết được công thức của từng đầu ra...
- Tin học kết hợp với điện tử thì có thể làm ra được rất nhiều thiết bị điện tử có giá trị.

[lightsmok@gmail.com](mailto:lightsmok@gmail.com)