

## MỤC LỤC

MỤC LỤC.....	1
Chương 1 KHÁI NIỆM VỀ HỆ THỐNG MÁY TÍNH.....	7
1.1. Lịch sử phát triển của máy tính điện tử.....	7
1.2. Khái niệm hệ thống máy tính .....	7
1.3. Cấu trúc cơ bản của hệ thống máy tính.....	9
1.4. Biểu diễn thông tin trong máy tính điện tử.....	10
1.4.1. Các hệ đếm.....	10
1.4.2. Chuyển đổi từ hệ thập phân sang các hệ khác .....	10
1.4.3. Chuyển từ hệ nhị phân sang các hệ khác .....	12
1.4.4. Chuyển từ hệ bát phân và hexa sang hệ thập phân và nhị phân.....	12
1.5. Các phép tính số học trên hệ nhị phân .....	13
1.5.1. Phép cộng.....	13
1.5.2. Phép trừ.....	14
1.5.3. Phép nhân.....	15
1.5.4. Phép chia.....	15
1.6. Khái niệm thuật toán .....	16
Chương 2: GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH C .....	17
2.1. Giới thiệu.....	17
2.1.1. Lịch sử ngôn ngữ C .....	17
2.1.2. Đặc điểm của ngôn ngữ C.....	17
2.1.3. Ứng dụng của ngôn ngữ C .....	17
2.2. Tập ký tự.....	17
2.3. Định danh .....	18
2.3.1. Từ khóa.....	18
2.3.2. Quy tắc đặt tên .....	18
2.4. Các kiểu dữ liệu .....	19
2.4.1. Kiểu dữ liệu cơ bản .....	19
2.4.2. Kiểu mở rộng .....	20
2.4.3. Định nghĩa kiểu dữ liệu bằng typedef .....	20
2.5. Biến .....	21
2.5.1. Khái niệm biến .....	21
2.5.2. Khai báo biến .....	21
2.5.3. Vị trí của khai báo biến.....	21
2.5.4. Lấy địa chỉ của biến.....	22
2.6. Hằng .....	22
2.6.1. Hằng kiểu số nguyên .....	23

2.6.2. Hằng kiểu long.....	23
2.6.3. Hằng dấu phẩy động.....	23
2.6.4. Hằng ký tự .....	23
2.6.5. Hằng chuỗi.....	24
2.6.6. Khai báo hằng .....	24
2.7. Các phép toán .....	24
2.7.1. Phép toán số học.....	24
2.7.2. Phép toán so sánh .....	26
2.7.3. Phép toán logic.....	26
2.7.4. Phép toán xử lý bit.....	27
2.7.5. Một số phép toán khác.....	28
2.8. Phép toán xử lý bit .....	29
2.8.1. Phép toán AND .....	29
2.8.2. Phép toán OR .....	29
2.8.3. Phép toán XOR .....	29
2.8.4. Phép toán NOT.....	30
2.8.5. Phép toán dịch trái/phải .....	30
2.9. Biểu thức .....	30
2.9.1. Khái niệm biểu thức .....	30
2.9.2. Phép gán.....	31
2.9.3. Chuyển đổi kiểu dữ liệu.....	31
2.10. Cấu trúc cơ bản của chương trình C.....	32
2.10.1. Cấu trúc chương trình.....	33
2.10.2. Một số chú ý khi viết chương trình .....	34
2.11. Làm quen với môi trường soạn thảo Turbo C .....	35
2.11.1. Mở Turbo C .....	35
2.11.2. Soạn thảo chương trình mới.....	35
2.11.3. Lưu chương trình vào bộ nhớ.....	36
2.11.4. Mở chương trình từ bộ nhớ.....	36
2.11.5. Dịch và thực hiện chương trình.....	36
2.11.6. Thoát khỏi môi trường C .....	36
Chương 3 CÁC LỆNH XUẤT NHẬP DỮ LIỆU .....	37
3.1. Giới thiệu chung.....	37
3.2. Các lệnh xuất dữ liệu cơ bản .....	37
3.2.1. Chuỗi điều khiển .....	37
3.2.2 Số nguyên hệ 10 có dấu và không dấu .....	39
3.2.3. Các lệnh xuất dữ liệu lên màn hình.....	41

3.2.4	Xuất dữ liệu ra máy in.....	44
3.3.	Các lệnh nhập dữ liệu.....	44
3.3.1.	Chuỗi điều khiển.....	44
3.3.2.	Dòng vào chuẩn (Stdin).....	46
3.4.	Một số hàm xuất/nhập bổ sung.....	50
3.4.1.	Các hàm xuất dữ liệu bổ sung.....	50
3.4.2.	Các hàm nhập dữ liệu bổ sung.....	52
3.4.4.	Xóa màn hình và di chuyển con trỏ.....	53
Chương 4	CÁC LỆNH ĐIỀU KHIỂN.....	55
4.1.	Giới thiệu chung.....	55
4.2.	Câu lệnh if.....	55
4.2.1.	Cú pháp câu lệnh if.....	55
4.2.2.	Lưu đồ.....	55
4.2.3.	Giải thích lưu đồ.....	56
4.3.	Câu lệnh if.. else.....	59
4.3.1.	Cú pháp câu lệnh if ... else (dạng đầy đủ).....	59
4.3.2.	Lưu đồ.....	59
4.3.3.	Giải thích lưu đồ.....	59
4.4.	Biểu thức điều kiện:.....	60
4.6.	Câu lệnh switch.....	62
4.6.1.	Cú pháp câu lệnh switch.....	62
4.6.2.	Lưu đồ thuật toán câu lệnh switch.....	62
4.6.3.	Giải thích lưu đồ.....	63
4.6.4.	Một số chú ý.....	63
4.6.5.	Một số ví dụ.....	63
4.7.	Câu lệnh for.....	65
4.7.1.	Cú pháp câu lệnh for.....	65
4.7.2.	Lưu đồ thuật toán câu lệnh for.....	65
4.7.3.	Giải thích lưu đồ.....	66
4.7.4.	Một số chú ý.....	66
4.7.5.	Một số ví dụ.....	66
4.8.	Câu lệnh while.....	68
4.8.1.	Cú pháp câu lệnh while.....	68
4.8.2.	Lưu đồ thuật toán.....	68
4.8.3.	Giải thích lưu đồ.....	69
4.8.4.	Một số chú ý.....	69
4.8.5.	Một số ví dụ.....	69

4.9. Câu lệnh do ... while.....	70
4.9.1. Cú pháp câu lệnh do ... while.....	70
4.9.2. Lưu đồ thuật toán .....	70
4.9.3. Giải thích lưu đồ.....	70
4.9.4. Ví dụ .....	71
4.10. Nhóm các lệnh điều khiển khác .....	71
4.10.1. Câu lệnh break.....	71
4.10.2. Câu lệnh continue.....	72
4.10.3. Câu lệnh return.....	72
4.10.4. Câu lệnh nhảy không điều kiện goto .....	73
Chương 5        HÀM .....	75
5.1. Giới thiệu chung.....	75
5.2. Định nghĩa và lời gọi hàm .....	75
5.2.1. Định nghĩa.....	76
5.2.2. Lời gọi hàm.....	76
5.2.3. Thí dụ về cách gọi hàm.....	77
5.3. Nguyên tắc hoạt động của hàm.....	78
5.4. Cách sử dụng các tham số trong hàm:.....	78
5.5. Chuyển giao tham số của hàm .....	79
5.5.1. Truyền bằng giá trị .....	79
5.5.2. Truyền bằng tham chiếu .....	79
5.6. Sự trả về từ hàm.....	80
5.7. Hàm đệ quy.....	80
5.8. Một số thí dụ.....	82
Chương 6        CON TRỞ .....	88
6.1. Giới thiệu.....	88
6.2. Biến con trỏ .....	88
6.2.1. Khái niệm con trỏ.....	88
6.2.2. Khai báo biến con trỏ .....	88
6.2.3. Gán giá trị cho biến con trỏ .....	89
6.3. Cấp phát bộ nhớ .....	91
6.3.1. Cấp phát bộ nhớ để lưu dữ liệu .....	91
6.3.3 Giải phóng vùng nhớ động .....	92
6.4. Các phép toán trên biến con trỏ .....	92
6.4.1. Phép cộng/trừ biến con trỏ với số nguyên .....	92
6.4.2 Phép gán và phép so sánh .....	93
6.4.3. Sự chuyển kiểu.....	94

6.4.4 Con trỏ hằng và con trỏ đến đối tượng hằng .....	94
6.4.5. Con trỏ NULL.....	95
6.5. Con trỏ và hàm.....	95
6.5.1. Khái niệm và cách khai báo con trỏ hàm.....	95
6.5.2. Đối con trỏ hàm.....	96
6.5.3. Thí dụ.....	96
Chương 7    MẢNG VÀ CHUỖI.....	99
7.1. Giới thiệu chung.....	99
7.2. Phần tử mảng và các chỉ số mảng.....	99
7.3. Cách khai báo một mảng.....	99
7.3.1. Khái niệm.....	100
7.3.2. Khai báo mảng một chiều.....	100
7.3.3. Mảng nhiều chiều.....	102
7.4. Sử dụng mảng một chiều và mảng hai chiều.....	105
7.4.1. Sử dụng mảng một chiều.....	105
7.4.2. Sử dụng mảng hai chiều để cộng ma trận.....	108
7.5. Con trỏ và mảng.....	112
7.5.1. Mối quan hệ giữa con trỏ và mảng.....	112
7.5.2. Con trỏ và mảng một chiều.....	113
7.5.3. Con trỏ và mảng nhiều chiều.....	115
7.5.4. Một số thí dụ.....	116
7.6. Các biến và hằng kiểu chuỗi.....	119
7.7. Các thao tác nhập/xuất chuỗi.....	119
7.7.1. Thao tác nhập/xuất chuỗi đơn giản.....	119
7.7.2. Thao tác nhập/xuất chuỗi có định dạng.....	120
7.8. Sử dụng các hàm về chuỗi.....	121
7.8.1. Hàm strcat().....	121
7.8.2. Hàm strcmp().....	122
7.8.3. Hàm strchr().....	122
7.8.4. Hàm strcpy().....	122
7.8.5. Hàm strlen().....	123
7.8.6. Sắp xếp chuỗi sử dụng các hàm trong thư viện.....	123
7.8.7. Sử dụng hàm để chuyển một mảng ký tự về chữ hoa.....	125
Chương 8    CẤU TRÚC VÀ DANH SÁCH LIÊN KẾT.....	127
8.1. Giới thiệu chung.....	127
8.2. Kiểu cấu trúc.....	127
8.2.1. Khái niệm cấu trúc.....	127

8.2.2. Định nghĩa kiểu cấu trúc.....	128
8.2.3. Khai báo theo kiểu cấu trúc đã định nghĩa .....	129
8.2.4. Truy xuất đến các thành phần của cấu trúc.....	130
8.2.5. Cấu trúc lồng nhau .....	131
8.3. Mảng cấu trúc .....	132
8.3.1. Khai báo biến mảng cấu trúc .....	132
8.3.2. Truy xuất các phần tử mảng cấu trúc .....	132
8.3.3. Khởi tạo giá trị cho các phần tử của mảng cấu trúc .....	132
8.4. Con trỏ cấu trúc.....	132
8.4.1. Khai báo con trỏ cấu trúc.....	132
8.4.2. Sử dụng con trỏ cấu trúc.....	133
8.4.3. Con trỏ và mảng cấu trúc.....	134
8.5. Danh sách liên kết.....	135
8.5.1. Khái niệm cấu trúc tự trỏ .....	135
8.5.2. Khái niệm danh sách liên kết.....	135
8.5.3. Các phép toán trên danh sách liên kết.....	136
8.6. Một số kiểu dữ liệu tự tạo khác .....	140
8.6.1. Kiểu liệt kê (enum).....	140
8.6.2. Kiểu hợp (union).....	140
Chương 9 TẬP TIN.....	142
9.1. Giới thiệu chung.....	142
9.2. Một số khái niệm về tệp tin .....	142
9.3. Thao tác trên tệp tin văn bản .....	143
9.3.1. Hàm đóng/mở file .....	143
9.3.2. Một số hàm có chức năng điều khiển.....	145
9.3.3. Các hàm ghi dữ liệu.....	146
9.3.4. Các hàm đọc dữ liệu từ file.....	147
9.4. Thao tác trên tệp tin nhị phân .....	149
9.4.1. Các hàm ghi dữ liệu.....	149
9.4.2. Các hàm đọc dữ liệu .....	149
9.4.3. Di chuyển con trỏ tệp tin - Hàm fseek() .....	150

## **Chương 1 KHÁI NIỆM VỀ HỆ THỐNG MÁY TÍNH**

### **1.1. Lịch sử phát triển của máy tính điện tử**

Lịch sử máy tính điện tử gắn liền với chặng đường phát triển của IBM-PC. Do đó ta có thể tóm tắt quá trình phát triển của máy tính điện tử như sau:

**1979-1980:** IBM cho ra đời máy Datamaster dùng vi xử lý 16 bit 8086 của Intel

**1980:** Đưa ra khái niệm: Personal Computer (PC). Chiếc IBM-PC đầu tiên dùng vi xử lý 8bit 8085 của Intel.

**1981-1982:** Dù Intel có vi xử lý 16bit nhưng giá thành còn cao, Để đáp ứng thị trường máy rẻ tiền, Intel đưa ra vi xử lý 8 bit 8088 mà trong nó là vi mạch 16bit 8086.

**1984:** Khi vi xử lý 16bit đã quen thuộc thị trường, Intel đưa ra vi xử lý 80286, là vi xử lý 16bit hoàn thiện, có thêm 4bit bus địa chỉ, quản lý 16MB bộ nhớ.

**1987:** Thế hệ PC mới ra đời với vi xử lý 80386. Bắt đầu từ đây IBM công khai cấu tạo máy và nội dung chương trình hệ điều hành vào ra cơ sở (BIOS), điều này giúp các hãng khác có thể sản xuất các máy tính tương thích và các bản mạch cắm tương thích khiến cấu trúc IBM-PC trở thành một cấu trúc chuẩn công nghiệp.

**1990:** 80486 ra đời với nhiều chức năng hơn, cụ thể là 8 Kbyte bộ nhớ đệm mã lệnh (code cache) và một bộ đồng xử lý toán học. Tần số làm việc đặc trưng của máy vi tính trong thời kỳ này là 66MHz.

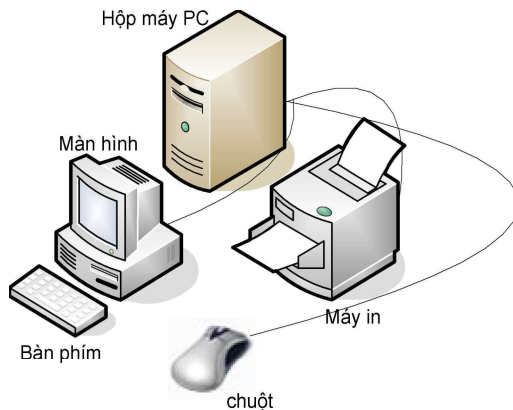
**1993:** Vi xử lý Pentium đầu tiên ra đời mở ra một kỷ nguyên mới với 64bit bus dữ liệu, 32bit bus địa chỉ, 8KB bộ đệm dữ liệu, 8KB bộ đệm mã lệnh. Bộ đồng xử lý toán học của Pentium làm việc nhanh gấp 10 lần so với 80486. Khi này các nhà sản xuất phần cứng lớn thoả thuận một chuẩn khe cắm mới PCI-bus (Peripheral Components Interconnect), và do đó bản mạch chính máy vi tính cá nhân chỉ còn lại vài vi mạch, tất cả các vi mạch ngoại vi của cấu trúc IBM-PC cũng như vi mạch điều khiển PCI được tích hợp vào một vi mạch duy nhất, có tên là PCI-chipset.

**1995:** Khả năng đa môi trường (multimedia) của máy vi tính cá nhân càng ngày càng hoàn thiện khi Pentium MMX , Pentium Pro, Pentium II lần lượt ra đời. Tần số đồng hồ cao nhất 300 MHz. Một chuẩn giao diện ngoại vi mới ra đời từ sự thoả thuận từ nhiều hãng lớn là bus tuần tự đa dạng USB (Universal Serial Bus).

Từ năm **2000:** Một cấu trúc vi xử lý 64 bit ra đời. Intel cho ra đời nhiều vi mạch tổng hợp thích hợp với vi xử lý của chính hãng. Chipset đảm nhiệm hầu hết các chức năng điều khiển trên máy và có bộ điều khiển hiển thị cấy ở bên trong. Thị trường **máy tính cá nhân** cũng như thị trường vi xử lý và vi mạch tổng hợp được chia thành nhiều phân đáp ứng nhu cầu đa dạng trong xã hội.

### **1.2. Khái niệm hệ thống máy tính**

**Máy tính** (computer) là thiết bị điện tử thực hiện công việc nhận thông tin vào, xử lý thông tin theo chương trình nhớ sẵn bên trong bộ nhớ máy tính sau đó cho kết quả thông tin đầu ra.



**Hình 1.1 Hệ thống máy tính điển hình**

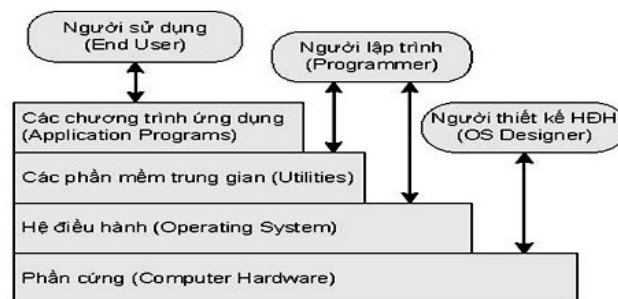
**Chương trình** (program) là dãy các lệnh nằm trong bộ nhớ để yêu cầu máy tính thực hiện công việc cụ thể. Như vậy, một máy tính bất kỳ đều làm việc theo chương trình.

**Phần cứng** (hardware) bao gồm các đối tượng hữu hình như các vi mạch (IC), các bảng mạch in, cáp nối, nguồn điện, bộ nhớ, máy đọc đĩa...

**Phần mềm** (software) bao gồm các thuật toán và các biểu diễn cho máy tính của chúng, đó chính là các chương trình (program). Cái cơ bản nhất của phần mềm là tập các chỉ thị tạo nên chương trình chứ không phải là môi trường vật lý được sử dụng để ghi chương trình. Ví dụ hệ điều hành DOS, Window, Unix ...

**Phần sụn** là trung gian giữa phần cứng và phần mềm, hay nói cách khác, phần sụn chính là phần mềm được nhúng vào các phần cứng trong quá trình chế tạo các phần cứng này. Phần sụn được sử dụng đối với các chương trình hiếm khi hoặc không bao giờ cần thay đổi hay trong trường hợp các chương trình không được phép bị mất khi tắt điện.

Hình sau mô tả mô hình phân lớp của máy tính sẽ giúp bạn đọc hiểu hơn về các khái niệm cơ bản nêu trên.

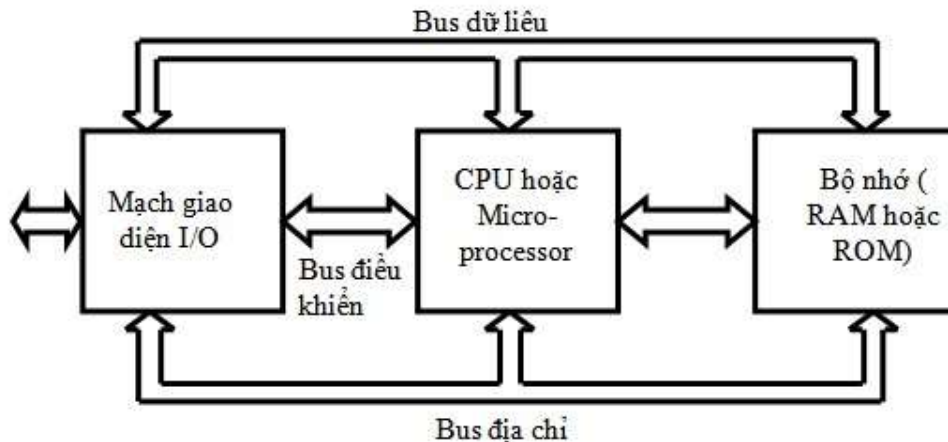


**Hình 1.2 Mô hình phân lớp của máy tính**



### 1.3. Cấu trúc cơ bản của hệ thống máy tính

Các phần chính của máy tính cơ bản là một đơn vị xử lý trung tâm (hoặc thường gọi là bộ vi xử lý), bộ nhớ và các mạch vào ra. Các phần tử đó được nối với nhau qua đường bus chính: bus địa chỉ, bus điều khiển và bus dữ liệu. Hình 1.3 mô tả một hệ thống cơ bản. Các các thiết bị ngoại vi như bàn phím, màn hình, ổ đĩa, có thể nối trực tiếp với các bus: dữ liệu, địa chỉ và điều khiển, hoặc ghép với các mạch vào ra.



**Hình 1.3** Sơ đồ cấu trúc một hệ thống máy tính đơn giản

**Khối xử lý trung tâm CPU** (Central Processing Unit): dùng để thu thập và cho chạy các lệnh. Bên trong CPU gồm các mạch điều khiển logic, mạch tính số học và logic,.. Nếu CPU được xây dựng trên một hoặc vài vi mạch, thường được đóng trong một chip, thì nó được gọi là bộ vi xử lý  $\mu P$  (microprocessor). Một máy tính dùng  $\mu P$  làm bộ xử lý thì được gọi là máy vi tính (microcomputer) hay máy tính cá nhân PC (Personal Computer).

**Bộ nhớ:** dùng để lưu trữ các lệnh và dữ liệu cho bộ xử lý. Nó bao gồm hai loại: bộ nhớ trong (được tạo bởi các vi mạch nhớ bán dẫn) và bộ nhớ ngoài (được tạo bởi các môi trường nhớ khác như đĩa từ, đĩa quang). Bộ nhớ thường được chia thành từng ô nhớ nhỏ như từ hay byte (1 byte=8 bit, 1 từ=2 byte). Mỗi ô nhớ đó cũng như một thiết bị vào/ra được gán cho một địa chỉ (address) để CPU có thể định vị khi cần đọc hay viết dữ liệu lên nó.

**Các thiết bị ngoại vi:** gồm các thiết bị vào/ra (I/O: input/output) dùng để nhập hoặc xuất các dữ liệu. Bàn phím, chuột, máy quét,...thuộc loại thiết bị vào. Màn hình, máy in,...thuộc loại thiết bị ra. Các ổ đĩa ở bộ nhớ ngoài có thể được coi vừa là thiết bị vào vừa là thiết bị ra. Các thiết bị ngoại vi này liên hệ với CPU qua các mạch ghép nối vào/ra (I/O interface). Mạch này cho phép nối hai bộ phận độc lập nhằm làm cho chúng có thể tương hợp và thông tin được với nhau.

**Bus hệ thống:** là một tập hợp các đường dây mà qua đó CPU có thể liên kết với các bộ phận khác.

## 1.4. Biểu diễn thông tin trong máy tính điện tử

### 1.4.1. Các hệ đếm

Trong cuộc sống hàng ngày chúng ta thường dùng các hệ cơ số 10 để biểu diễn các giá trị số. Tuy nhiên đối với máy tính lại khác, do máy tính được cấu tạo từ các mạch điện tử và các mạch này chỉ có hai trạng thái có điện và không có điện. Do đó để biểu diễn một giá trị số trong máy tính người ta sử dụng hệ đếm cơ số hai hay hệ đếm nhị phân (Binary). Trong hệ đếm này chỉ tồn tại hai chữ số 1 và 0 tương ứng với hai trạng thái có điện và không có điện của các mạch điện tử.

Nếu dùng hệ cơ số hai để biểu diễn các số có giá trị lớn sẽ gặp bất tiện là số hệ hai thu được quá dài, thí dụ:  $55 = 110111$  hay  $253 = 11111101$ .

Do đó, để viết kết quả biểu diễn các số cho gọn lại người ta sử dụng các hệ đếm khác như hệ cơ số 16 (thập lục phân hay hexa) và hệ cơ số 8 (bát phân). Bảng sau đây trình bày một số hệ đếm cơ bản:

**Bảng 1.1 Các hệ đếm cơ bản**

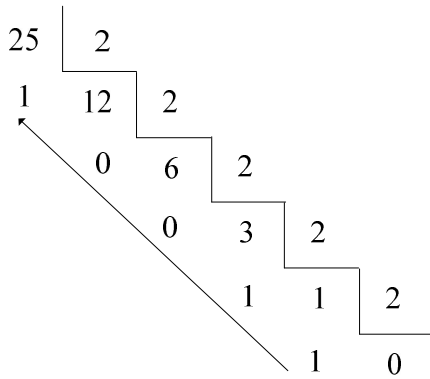
Các hệ đếm	Cơ số	Số ký số và ký tự	Dạng ký số và ký tự
Nhị phân (Binary)	2	2	0,1
Bát phân (Octal)	8	8	0,1,2,3,4,5,6,7
Thập phân (Decimal)	10	10	0,1,2,3,4,5,6,7,8,9
Thập lục phân (Hexadecimal)	16	16	0,1,2,3,4,5,6,7,8,9 A,B,C,D,E,F

### 1.4.2. Chuyển đổi từ hệ thập phân sang các hệ khác

#### 1.4.2.1. Chuyển đổi từ hệ thập phân sang hệ nhị phân

**Quy tắc:** Để thực hiện việc đổi từ hệ thập phân sang nhị phân, ta áp dụng phương pháp chia lặp như sau: Lấy số thập phân chia cho cơ số để thu được một thương số và số dư. Số dư được ghi lại để làm một thành tố của số nhị phân. Sau đó, số thương lại được chia cho cơ số một lần nữa để có thương số thứ 2 và số dư thứ 2. Số dư thứ hai là con số nhị phân thứ hai. Quá trình tiếp diễn cho đến khi số thương bằng 0. Sau đó viết các số dư theo chiều từ phép chia cuối cùng đến phép chia đầu tiên.

**Thí dụ:** Chuyển số 25 trong hệ 10 sang hệ nhị phân:



Kết quả là:  $(25)_{10} = (11001)_2$

Ngoài ra, cần xét đến quy tắc chuyển đổi từ một số thập phân sang số nhị phân như sau: Lấy số cần đổi nhân với 2, tích gồm phần nguyên và phần lẻ. Lấy phần lẻ nhân tiếp với 2 cho đến khi nào tích thu được bằng 1 thì dừng lại. Chọn riêng phần nguyên của các tích thu được và viết theo thứ tự từ phép nhân đầu tiên đến phép nhân cuối cùng. Thí dụ: chuyển số thập phân 0.125 sang hệ nhị phân

$$0.125 \times 2 = 0.25$$

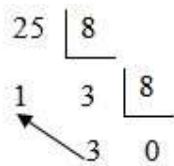
$$0.25 \times 2 = 0.50 \quad \Rightarrow (0.125)_{10} = (0.001)_2$$

$$0.50 \times 2 = 1.0$$

#### 1.4.2.2. Chuyển đổi từ hệ thập phân sang hệ bát phân

**Quy tắc:** Quy tắc chuyển đổi từ hệ thập phân sang hệ bát phân cũng thực hiện tương tự như với hệ nhị phân. Ta cũng thực hiện phép chia số thập phân cho cơ số (cơ số 8) được thương và số dư. Số dư sẽ là thành phần của số bát phân. Tiếp tục lấy thương chia cho cơ số và thực hiện cho tới khi nào thương bằng 0. Số bát phân sẽ là tập dãy các số dư được viết theo chiều từ phép chia cuối đến phép chia đầu.

**Thí dụ:** chuyển số 25 sang hệ bát phân:



Kết quả:  $(25)_{10} = (31)_8$

#### 1.4.2.3. Chuyển đổi từ hệ thập phân sang hệ thập lục phân

**Quy tắc:** Việc chuyển từ hệ thập phân sang hệ hexa (16) cũng được thực hiện hoàn toàn tương tự như quy tắc chuyển từ hệ thập phân sang hệ bát phân và nhị phân. Tức là ta cũng thực hiện các phép chia liên tiếp số thập phân đó và thương của nó cho cơ số 16 cho đến khi thương bằng 0. Và kết quả hệ 16 là dãy số dư được viết ngược từ phép chia cuối đến phép chia đầu.

Tuy nhiên, cần lưu ý rằng trong phép chia cho 16, các số dư là từ 0 đến 15. Để thuận tiện biểu diễn, từ số 10 đến 15 sẽ được thay bằng các ký tự chữ cái từ a đến f (hoặc A đến F).

Thí dụ: Chuyển số 45 sang hệ Hexa:

$$\begin{array}{r}
 45 \quad | \quad 16 \\
 \hline
 D \leq 13 \quad 2 \quad | \quad 16 \\
 \hline
 \quad \quad \quad 2 \quad 0
 \end{array}$$

Kết quả:  $(45)_{10} = (2D)_{16}$

### 1.4.3. Chuyển từ hệ nhị phân sang các hệ khác

#### 1.4.3.1. Chuyển từ hệ nhị phân sang hệ thập phân

Để chuyển từ hệ nhị phân sang thập phân ta tính tổng các tích số giữa các hệ số với các trọng số  $2^i$  tại từng vị trí thứ  $i$ .

Thí dụ:

$$\begin{aligned}
 (1101)_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
 &= 8 + 4 + 0 + 1 = (13)_{10} \\
 (10110.11)_2 &= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} \\
 &= 16 + 0 + 4 + 2 + 0 + 0,5 + 0,25 = (22,75)_{10}
 \end{aligned}$$

#### 1.4.3.2. Chuyển từ hệ nhị phân sang hệ bát phân

Để chuyển từ hệ nhị phân sang hệ bát phân, ta nhóm từng nhóm 3 bit, bắt đầu từ phải sang trái. Sau đó tính giá trị của từng nhóm đó (cho giá trị từ 0 đến 7), ghi các giá trị liền nhau theo trình tự cũ, ta sẽ có giá trị bát phân tương ứng.

Thí dụ:

$$\begin{aligned}
 \text{Hệ nhị phân:} & \quad 11 \ 001 \ 101 \ 011 \\
 \text{Hệ bát phân:} & \quad 3 \ 1 \ 5 \ 3 \\
 \Rightarrow & \quad (11001101011)_2 = (3153)_8
 \end{aligned}$$

#### 1.4.3.3. Chuyển từ hệ nhị phân sang hệ hexa

Để chuyển từ hệ nhị phân sang hệ hexa, ta cũng thực hiện nhóm các bit thành các nhóm 4 bit, bắt đầu từ phải sang trái. Tính giá trị của từng nhóm bit đó (cho giá trị từ 0 đến 15), ghi các giá trị lần lượt theo thứ tự cũ ta được số hexa tương ứng.

Thí dụ:

$$\begin{aligned}
 \text{Hệ nhị phân:} & \quad 110 \ 1001 \ 1101 \ 0000 \\
 \text{Hệ Hexa:} & \quad 6 \ 9 \ D \ 0 \\
 \Rightarrow & \quad (110100111010000)_2 = (69D0)_{16}
 \end{aligned}$$

### 1.4.4. Chuyển từ hệ bát phân và hexa sang hệ thập phân và nhị phân

#### 1.4.4.1. Chuyển từ hệ bát phân và hệ hexa sang hệ thập phân

Quy tắc để chuyển từ hệ bát phân và hệ hexa sang hệ thập phân là tương tự nhau và tương tự với quy tắc chuyển từ hệ nhị phân sang hệ thập phân. Giá trị thập phân sẽ là tổng của các tích số giữa các hệ số với trọng số  $8^i$  (với hệ bát phân) hoặc  $16^i$  (với hệ hexa) ứng với từng vị trí thứ  $i$ .

Thí dụ:

$$\begin{aligned}(407)_8 &= 4 \cdot 8^2 + 0 \cdot 8^1 + 7 \cdot 8^0 \\ &= 256 + 0 + 7 \\ &= (263)_{10} \\ (4CA0)_{16} &= 4 \cdot 16^3 + 12 \cdot 16^2 + 10 \cdot 16^1 + 0 \cdot 16^0 \\ &= 16384 + 3072 + 160 + 0 \\ &= 19616\end{aligned}$$

#### 1.4.4.2. Chuyển từ hệ bát phân và hexa sang hệ nhị phân

Quy tắc chuyển từ hệ bát phân và hệ hexa sang hệ nhị phân đơn giản là cách tính ngược lại so với cách chuyển từ hệ nhị phân sang hai hệ này. Tức là lấy từng chữ số trong số bát phân (hoặc số hexa) chuyển sang giá trị nhị phân tương ứng là một nhóm 3 bit (hoặc nhóm 4 bit). Giữ nguyên thứ tự các chữ số, ta được giá trị số nhị phân tương ứng.

Thí dụ:

$$\begin{aligned}(407)_8 &= (100\ 000\ 111)_2 \\ &= (100000111)_2 \\ (20A)_{16} &= (0010\ 0000\ 1010)_2 \\ &= (1000001010)_2\end{aligned}$$

### 1.5. Các phép tính số học trên hệ nhị phân

Phép tính số học dùng trong hệ nhị phân cũng tương tự như các phép tính được áp dụng trong các hệ khác. Các phép cộng, trừ, nhân và chia cũng được áp dụng với các giá trị số nhị phân.

#### 1.5.1. Phép cộng

Phép tính đơn giản nhất trong hệ nhị phân là tính cộng. Cộng hai đơn vị trong hệ nhị phân được làm như sau:

$$\begin{aligned}0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 10 \text{ (nhớ 1 lên hàng thứ 2)}\end{aligned}$$

Cộng hai số "1" với nhau tạo nên giá trị "10", tương đương với giá trị 2 trong hệ thập phân. Số "1" sẽ được nhớ vào phép cộng của các bit liền kề bên trái.

Thí dụ: thực hiện phép cộng:

$$11001011$$

$$\begin{array}{r}
 + \quad 1011101 \\
 \hline
 = \quad 100101000
 \end{array}$$

Tương tự, nếu ta cộng 3 bit “1” với nhau ta sẽ được giá trị là “11” và số một có trọng số cao hơn (phía bên trái) sẽ được nhớ vào phép cộng của các bit tiếp theo.

### 1.5.2. Phép trừ

Phép tính trừ theo quy tắc tương tự như phép cộng:

$$0 - 0 = 0$$

$$0 - 1 = -1 \text{ (mượn)}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Phép trừ thực chất cũng là phép cộng với một số âm của số đó.

Thí dụ:

$$\begin{array}{r}
 11011010 \\
 - \quad 1001110 \\
 \hline
 = 10001100
 \end{array}$$

Phép toán trên tương đương với phép toán trong hệ thập phân

$$218 - 78 = 140$$

Ta có thể hiểu phép toán trên là phép cộng:

$$218 + (-78) = 140$$

Việc biểu diễn một số âm trong máy tính được thực hiện bởi phương pháp bù 2. Theo phương pháp này, bit cực trái (là bit nằm bên trái cùng của byte) được sử dụng làm bit dấu (*sign bit* - là bit tượng trưng cho dấu của số) với quy ước: nếu bit dấu là 0 thì số là số dương, còn nếu nó là 1 thì số là số âm. Ngoài bit dấu này ra, các bit còn lại được dùng để điều khiển độ lớn của số.

*Quy tắc biểu diễn số nguyên âm theo phương pháp bù 2 như sau:*

- Biểu diễn phần nguyên dương dưới dạng nhị phân.
- Đảo tất cả các bit nhận được trong bước 1
- Cộng thêm 1 vào kết quả của bước 2
- Bit ngoài cùng bên trái là bit 1 (bit dấu của số âm)

Thí dụ: Biểu diễn số -7 theo phương pháp bù 2

Bước 1: Biểu diễn số 7 dưới dạng nhị phân: 0000 0111

Bước 2: Đảo tất cả các bit trong dãy bit trên: 1111 1000

Bước 3: Cộng thêm 1 vào kết quả của bước 2: 1111 1001.

Như vậy kết quả biểu diễn số -7 trong máy tính là: 1111 1001.

Như vậy, trong thí dụ về phép trừ 218 cho 78, ta có thể biểu diễn số -78 theo phương pháp bù 2, sau đó thực hiện phép cộng 218 (dạng nhị phân) với -78 (dạng nhị phân).

Tuy nhiên, cần lưu ý rằng khi thực hiện phép tính cộng với số âm biểu diễn theo phương pháp bù 2, ta thực hiện như phép cộng nhị phân bình thường nhưng trong trường hợp khi đã thực hiện phép cộng đến bit cực trái mà vẫn phát sinh bit nhớ thì ta bỏ bit nhớ này đi.

Quay lại thí dụ phép trừ trên, ta biểu diễn số -78 theo phương pháp bù 2:

Bước 1:  $(78)_{10} = (0100\ 1110)_2$

Bước 2: đảo bit: 1011 0001

Bước 3: Cộng thêm 1: 1011 0010

Như vậy số -78 có dạng nhị phân là: 1011 0010. Giờ ta sẽ thực hiện phép cộng nhị phân của số 218 và -78:

$$\begin{array}{r}
 11011010 \\
 - 10110010 \\
 \hline
 = 10001100 \quad (\text{Đã bỏ bit nhớ ngoài cùng bên trái})
 \end{array}$$

### 1.5.3. Phép nhân

Phép tính nhân trong hệ nhị phân cũng tương tự như phương pháp làm trong hệ thập phân.

Thí dụ:

$$\begin{array}{r}
 10011 \quad = (19)_{10} \\
 \times \quad 101 \quad = (5)_{10} \\
 \hline
 10011 \\
 + 00000 \\
 + 10011 \\
 \hline
 = 1011111 \quad = (95)_{10}
 \end{array}$$

### 1.5.4. Phép chia

Phép chia nhị phân cũng tương tự như phép chia thập phân:

Thí dụ:

$$\begin{array}{r}
 110010 \overline{)101} \\
 - 101 \quad 1010 \\
 \hline
 0010
 \end{array}$$