

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

TaiLieu.vn

LỜI GIỚI THIỆU

Để thực hiện biên soạn giáo trình đào tạo nghề Điện tử công nghiệp ở trình độ Cao Đẳng và Trung Cấp, giáo trình **lập trình vi xử lý** là một trong những giáo trình mô đun đào tạo chuyên ngành được biên soạn theo nội dung chương trình chi tiết mô đun **lập trình vi xử lý**. Nội dung biên soạn ngắn gọn, dễ hiểu, tích hợp kiến thức và kỹ năng chặt chẽ với nhau, logic.

Khi biên soạn, nhóm biên soạn đã cố gắng cập nhật những kiến thức mới có liên quan đến nội dung chương trình đào tạo và phù hợp với mục tiêu đào tạo, nội dung lý thuyết và thực hành được biên soạn gắn với nhu cầu thực tế trong sản xuất đồng thời có tính thực tiễn cao.

Nội dung giáo trình được biên soạn với dung lượng thời gian đào tạo 75 giờ gồm có:

Bài 1: Tổng quan về các hệ vi xử lý

Bài 2: Đơn vị vi xử lý trung tâm

Bài 3: Bộ nhớ trong của hệ vi xử lý

Bài 4: Thiết bị vào ra của hệ vi xử lý

Trong quá trình sử dụng giáo trình, tùy theo yêu cầu cũng như khoa học và công nghệ phát triển có thể điều chỉnh thời gian và bổ sung những kiến thức mới cho phù hợp. Trong giáo trình có đề ra nội dung thực tập của từng bài để người học cũng cố và áp dụng kiến thức phù hợp với kỹ năng. Tuy nhiên, tùy theo điều kiện cơ sở vật chất và trang thiết bị của trường có thể sử dụng cho phù hợp.

Rất mong nhận được đóng góp ý kiến của quý thầy, cô giáo, bạn đọc để nhóm biên soạn sẽ hiệu chỉnh hoàn thiện hơn. Các ý kiến đóng góp xin gửi về Khoa Điện tử điện lạnh - Trường Cao đẳng nghề Kỹ thuật Công nghệ.

Hà Nội, ngày tháng năm 2019

BAN CHỦ NHIỆM BIÊN SOẠN GIÁO TRÌNH

NGHỀ: ĐIỆN TỬ CÔNG NGHIỆP

TRƯỜNG CAO ĐẲNG NGHỀ KỸ THUẬT CÔNG NGHỆ

MỤC LỤC

	TRANG
LỜI GIỚI THIỆU	2
BÀI 1: TỔNG QUAN VỀ CÁC HỆ VI XỬ LÝ.....	9
1. Các hệ đếm	9
1.1 Hệ thập phân (Decimal Number System)	9
1.2. Hệ nhị phân (Binary Number System).....	10
1.3. Hệ thập lục phân (Hexadecimal Number System).....	12
1.4. Mã BCD (Binary Coded Decimal).....	13
1.5. Mã hiển thị Led 7 thanh (7-segment display)	13
2. Chuyển đổi lẫn nhau giữa các hệ đếm.....	14
2.1. Hệ nhị phân và hệ thập phân	14
2.2. Hệ nhị phân và hệ hexa	15
3. Biểu diễn thông tin trong các hệ vi xử lý	15
3.1. Mã hóa các thông tin không số	15
3.1.1. Mã hóa chữ và dữ liệu kiểu văn bản.....	15
3.1.2. Các dữ liệu logic.....	16
3.1.4. Âm thanh	17
3.2. Mã hóa các thông tin số	18
3.2.1. Mã ASCII	19
3.2.2. Quan hệ giữa mã ASCII và số BCD	21
3.3. Biểu diễn dữ liệu số trong máy tính.....	21
3.4. Bản chất vật lý của thông tin trong các hệ vi xử lý.....	22
4. Các phép tính trong hệ đếm nhị phân.....	22
4.1. Phép cộng và phép trừ.....	22
4.1.1. Phép cộng	22
4.1.2. Phép trừ	23
4.2. Phép nhân và phép chia.....	24
4.2.1. Phép nhân	24
4.2.2. Phép chia	25

4.3. Sơ lược các phép toán logic	25
4.3.1 Sơ lược các cổng đệm (buffer) và các cổng logic (logic gate).....	25
4.3.2. Thiết bị logic lập trình được.....	26
BÀI 2: ĐƠN VỊ VI XỬ LÝ TRUNG TÂM.....	32
1. Trung tâm vi xử lý họ 80x86.....	32
1.1. Mô tả chân của μP 8086 và các tín hiệu	32
1.2. Cấu trúc trung tâm vi xử lý họ 80x86	35
1.3. Hệ thống thanh ghi trong các μP 8086	37
1.3.1. Các thanh ghi đa năng.....	37
1.3.2. Các thanh ghi đoạn.....	37
1.3.3. Thanh ghi cờ.....	39
1.3.4. Thanh ghi con trỏ lệnh IP.....	41
1.3.5. Các thanh ghi dữ liệu	41
1.3.6. Các thanh ghi con trỏ và chỉ số	41
1.4. Các chế độ làm việc MIN/MAX.....	42
1.5. Phương thức quản lý bộ nhớ, các mode địa chỉ.....	42
1.5.1. Phương thức quản lý bộ nhớ	42
1.5.2. Các mode đánh địa chỉ	44
1.6. Phương thức đánh địa chỉ thiết bị ngoại vi	47
1.7. Chương trình và tập lệnh của μP 8086	47
1.7.1. Sơ lược về lập trình hợp ngữ.....	47
1.7.2. Các bước khi lập trình.....	48
1.7.3 Cấu trúc chung của chương trình hợp ngữ.....	50
1.7.4 Các cấu trúc điều khiển cơ bản	61
1.7.5. Ngắt trong Assembly	65
2. Cấu trúc và tính năng của một số chip vi xử lý hiện đại.....	67
2.1. Cấu trúc chip vi xử lý Pentium	67
2.2. Quản lý bộ nhớ.....	70
2.3. Bộ nhớ cache.....	71

BÀI 3: BỘ NHỚ TRONG CỦA HỆ VI XỬ LÝ	74
1. Bộ nhớ trong hệ vi xử lý	74
1.1. Phần tử nhớ, vi mạch nhớ, từ nhớ và dung lượng bộ nhớ	74
1.2. Vài nét về bộ nhớ trong của hệ vi xử lý và máy tính.....	76
1.3. Phân loại các chip nhớ ROM, RAM.....	79
2. Tổ chức bộ nhớ cho hệ vi xử lý.....	81
2.1. Tổ chức bộ nhớ vật lý	81
2.2. Thiết kế vi nhớ cho hệ vi xử lý	82
BÀI 4: THIẾT BỊ VÀO RA CỦA HỆ VI XỬ LÝ.....	86
1. Bàn phím HEX (keyboard)	86
2. Ghép nối bàn phím với hệ vi xử lý.....	92
2.1. Hệ thống bàn phím của máy vi tính	92
2.2. Quá trình truyền dữ liệu từ bàn phím cho CPU	93
3. Bài tập thực hành vi xử lý 8086	96
3.1. Thực hành với led đơn	96
3.1.1. Yêu cầu công nghệ.....	96
3.1.2.Lưu đồ giải thuật	97
3.1.3.Chương trình điều khiển	98
3.2. Thực hành với led 7 thanh.....	100
3.2.1. Cấu trúc mã hiển thị dữ liệu trên LED 7 thanh.....	100
3.2.2. Yêu cầu công nghệ.....	101
3.2.3.Lưu đồ giải thuật	102
3.2.4.Chương trình điều khiển	103
3.3. Thực hành với LED ma trận	105
3.3.1.Cấu tạo led ma trận 8x8	105
3.3.2. Phương pháp quét led ma trận 8x8	105
3.3.3. Yêu cầu công nghệ.....	106
3.3.4. Chương trình điều khiển	106
3.4. Thực hành với LCD	108
3.4.1. Mô tả các chân của LCD.....	108

3.4.2. Các mã lệnh LCD.....	109
3.4.3. Yêu cầu công nghệ.....	110
3.4.4. Chương trình điều khiển	110
TÀI LIỆU THAM KHẢO.....	115

TaiLieu.vn

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: VI XỬ LÝ

Mã môn học/mô đun: MĐ ĐTCN 20

Vị trí, tính chất, ý nghĩa và vai trò của môn học/mô đun:

- Vị trí:

Mô đun được bố trí dạy cuối chương trình sau khi học xong các môn học cơ bản như linh kiện điện tử, đo lường điện tử, kỹ thuật xung - số, điện tử công suất....

- Tính chất:

Là mô đun chuyên môn nghề.

- Ý nghĩa và vai trò của môn học/mô đun:

+ Ý nghĩa của mô đun: sau khi học xong mô đun Vi xử lý, người học phải biết lập trình một số ứng dụng từ đơn giản đến phức tạp và xử lý được các kết nối máy tính với thiết bị ngoại vi.

+ Vai trò của mô đun: Giáo trình mô đun “Vi xử lý” nhằm cung cấp cho người học những kiến thức về lập trình và giao tiếp với máy tính bằng vi xử lý.

- **Mục tiêu của môn học/mô đun:**

- Về kiến thức:

+ Trình bày được về hệ đếm và mã hóa trong máy tính, tương tác giữa máy tính và vi xử lý, các bộ vi xử lý intel đang được ứng dụng

+ Giải thích được nguyên lý làm việc các hệ điều khiển ứng dụng vi xử lý

- Về kỹ năng:

+ Lập trình hợp ngữ một số bài tập cơ bản một cách thành thạo

+ Xử lý được một số dạng kết nối máy tính với vi xử lý và các thiết bị ngoại vi

- Về thái độ:

+ Rèn luyện tính tỷ mỉ, chính xác và an toàn vệ sinh công nghiệp

Nội dung của mô đun:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Thi/Kiểm tra
1	Bài 1: Tổng quan về các hệ vi xử lý	12	4	8	0
2	Bài 2: Đơn vị vi xử lý trung tâm	16	5	10	1
3	Bài 3: Bộ nhớ trong của hệ vi xử lý	12	5	6	1
4	Bài 4: Thiết bị vào ra của hệ vi xử lý	33	11	21	1
5	Thi kết thúc mô đun	2			2
	Cộng	75	25	45	5

BÀI 1: TỔNG QUAN VỀ CÁC HỆ VI XỬ LÝ

Mã bài: MĐ ĐTCN 20 - 01

Giới thiệu:

Khoa học kỹ thuật đang ngày càng phát triển rất mạnh mẽ, các công nghệ mới thuộc các lĩnh vực khác nhau nhờ đó đã ra đời nhằm đáp ứng nhu cầu của xã hội và kỹ thuật **Vi xử lý** cũng nằm trong số đó. Hiện nay kỹ thuật Vi xử lý đã được giảng dạy rộng rãi ở các trường Đại học và Cao đẳng trong cả nước, tuy nhiên lĩnh vực mới này (**Vi xử lý**) vẫn đang còn rất mới mẻ, và những ứng dụng của nó vẫn chưa được khai thác triệt để trong các hệ thống điều khiển, đo lường và điều chỉnh của các dây chuyền công nghiệp.

Mục tiêu:

- Trình bày được một số hệ đếm, và các mã thường dùng trong hệ vi xử lý
- Tính toán, chuyển đổi được các phép toán nhị phân
- Trình bày được cách biểu diễn thông tin trong các hệ vi xử lý
- Rèn luyện tính tư duy, tác phong trong công nghiệp

Nội dung chính:

1. Các hệ đếm

1.1 Hệ thập phân (Decimal Number System)

Đây là hệ đếm mà chúng ta sử dụng hàng ngày. Hệ đếm này sử dụng mười chữ số ‘0’..’9’ để biểu diễn các số hệ 10. Chữ số ‘0’ biểu diễn số 0, chữ số ‘1’ biểu diễn số 1,... Khi đếm đến 10 thì hết số nên phải thêm 1 sang hàng bên trái để thành ‘10’ (mười) rồi lại tiếp tục đếm ‘11’, ‘12’, ‘13’,... Một số hệ 10 có giá trị bằng tổng giá trị của từng chữ số nhân với trọng số tương ứng của chữ số. Trọng số của chữ số thứ i là 10^i .

Ví dụ: $12345,67 = 1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2}$.

Khi làm việc với nhiều hệ đếm các số hệ 10 có thêm chữ D ở cuối để chỉ ra rằng đó là số hệ 10.

Ví dụ: 12,25_D.

Tổng quát hóa cho hệ đếm cơ số a bất kỳ ($a \geq 2, a \in \mathbb{N}$):

Sử dụng a chữ số để biểu diễn các số hệ a . Chữ số có giá trị nhỏ nhất là ‘0’, chữ số có giá trị lớn nhất tùy thuộc vào từng hệ đếm nhưng có giá trị là $a-1$.

Giá trị (lượng) của một số hệ a bằng tổng giá trị của từng chữ số nhân với trọng số tương ứng của chữ số. Trọng số của chữ số thứ i là a_i .

Ví dụ: Số 5346,72 biểu diễn như sau:

$$5346,72 = 5 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10 + 6 + 7 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

Tuy nhiên, trong các mạch điện tử, việc lưu trữ và phân biệt 10 mức điện áp khác nhau rất khó khăn nhưng việc phân biệt hai mức điện áp thì lại dễ dàng. Do đó, người ta sử dụng hệ nhị phân để biểu diễn các giá trị trong hệ thống số. Trong số thập phân thì:

Số tận cùng bên trái là số có giá trị lớn nhất MSD (Most Significant Digit).

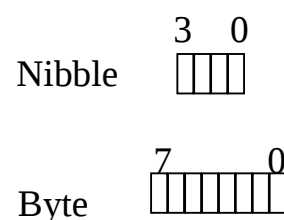
Số tận cùng bên phải là số có giá trị nhỏ nhất LSD (Least Significant Digit).

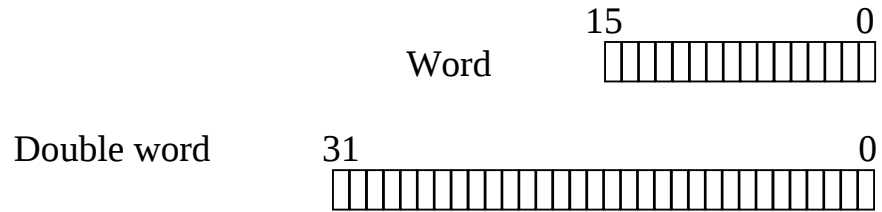
1.2. Hệ nhị phân (Binary Number System)

Hệ nhị phân gồm có 2 chữ số: 0 và 1, ký số nhị phân gọi là bit (binary digit). Cơ sở hệ nhị phân hay gọi là cơ sở 2. Một số nhị phân (binary digit) thường được gọi là **bit**. Một chuỗi gồm 4 bit nhị phân gọi là **nibble**, chuỗi 8 bit gọi là **byte**, chuỗi 16 bit gọi là **word** và chuỗi 32 bit gọi là **double word**. Bit tận cùng bên trái là bit có giá trị lớn nhất **MSB (Most Significant bit)**, bit tận cùng bên phải là bit có giá trị nhỏ nhất **LSB (Least Significant bit)**. Một số trong hệ nhị phân được biểu diễn theo số mũ của 2. Ta thường dùng chữ **B** cuối chuỗi bit để xác định đó là số nhị phân.

Các linh kiện điện tử cấu tạo nên máy tính chỉ có hai trạng thái: có điện và không có điện. Hai trạng thái này có thể được biểu diễn bằng 1 và 0. Chính vì lý do này mà hệ đếm cơ sở 2 là hệ đếm duy nhất được dùng trong máy tính.

Trong kỹ thuật máy tính, mỗi chữ số nhị phân được gọi là một bit (viết tắt của từ tiếng anh binary digit). Một cụm 4 bit tạo thành một nibble, cụm 8 bit tạo thành 1 byte, cụm 16 bit tạo thành 1 từ (word), cụm 32 bit tạo thành một từ kép (double word). Bit đầu tiên bên trái trong các số nhị phân gọi là bit có trọng số lớn nhất (Most Significant Bit, MSB), còn bit tận cùng bên phải gọi là bit có trọng số nhỏ nhất (Least Significant Bit, LSB). Vị trí của các bit trong nibble, byte, word, double word như sau:

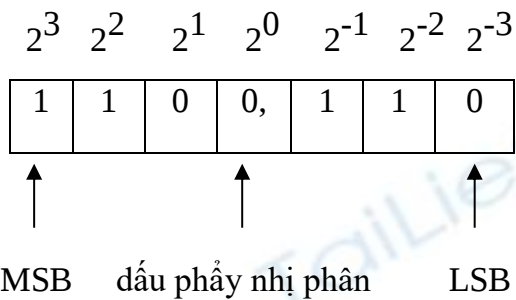




Ví dụ 1: Số 101110.01b biểu diễn giá trị số:

$$101110.01b = 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 0 + 0x2^{-1} + 1x2^{-2}$$

Ví dụ 2: cho 1 số nhị phân 1100,1102 được minh hoạ như hình vẽ:



$$1100,1102 = 1x2^3 + 1x2^2 + 0x2^1 + 0x2^0 + 1x2^{-1} + 1x2^{-2} + 0x2^{-3}$$

Cách đếm số nhị phân: ta dùng một số nhị phân 4 bit trình bày như bảng sau:

2^3	2^2	2^1	2^0	Thập phân tương ứng
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12

1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Với số nhị phân 4 bit như trên có $2^4 = 16$ trạng thái khác nhau và số thập phân tương ứng lớn nhất là: $2^4 - 1 = 15$. Vậy với một số nhị phân n bit thì:

+ Số thập phân tương ứng lớn nhất $2^n - 1$.

+ Số trạng thái 2^n .

1.3. Hệ thập lục phân (Hexadecimal Number System)

Nếu dùng hệ nhị phân thì sẽ cần một số lượng lớn các bit để biểu diễn. Trong thực tế để viết kết quả biểu diễn các số cho gọn lại người ta tìm cách nhóm 4 số hệ hai (1 nibble) thành một số hệ mười sáu. Khác với hệ BCD hệ 16 dùng hết các tổ hợp có thể của 4 bit để biểu diễn các giá trị số. Để làm được điều này người ta sử dụng các chữ số sẵn có của hệ mười (0 .. 9) để biểu diễn các giá trị số ứng với 0 .. 9 và dùng thêm các chữ cái A .. F để biểu diễn các giá trị còn lại ứng với 10 .. 15. Để phân biệt một số hệ mười sáu với các số hệ khác ta kèm thêm chữ H ở cuối. Ta cũng dễ nhận thấy rằng số mười chỉ là một bộ phận của hệ mười sáu.

- A biểu diễn cho 10
- B biểu diễn cho 11
- C biểu diễn cho 12
- D biểu diễn cho 13
- E biểu diễn cho 14
- F biểu diễn cho 15

Khi đếm đến 16 vì không còn chữ số nên phải thêm 1 sang bên trái để tạo thành '10',... Giá trị của một số hệ 16 bằng tổng các tích giữa giá trị của từng chữ số nhân với trọng số của nó. Trọng số của chữ số thứ i trong một số hệ 16 là 16^i :

$$S_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

Ví dụ 1: Để biểu diễn số 255 ta cần đến 8 bit viết như sau:

$$255 = 11111111_2$$

$$\textbf{Ví dụ 2: } 2AE4H = 2 \times 16^3 + A \times 16^2 + E \times 16^1 + 4 \times 16^0 = 10980D$$

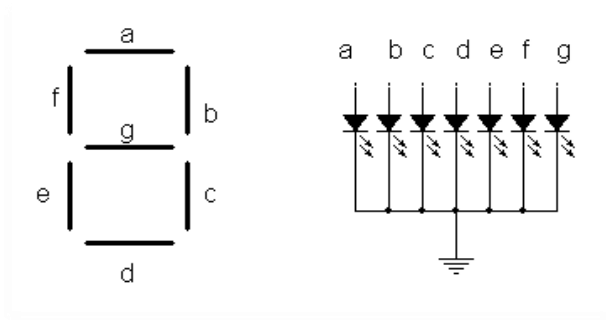
1.4. Mã BCD (Binary Coded Decimal)

Trong thực tế, đối với một số ứng dụng như đếm tần, đo điện áp, ... ngõ ra ở dạng số thập phân, ta dùng mã BCD. Mã BCD dùng 4 bit nhị phân để mã hóa cho một số thập phân 0..9. Như vậy, *các số hex A..F không tồn tại trong mã BCD*. Vì tầm quan trọng của các số BCD nên các bộ vi xử lý thường có các lệnh thao tác với chúng.

Ví dụ: Số thập phân 5 2 9
 Số BCD 0101 0010 1001

1.5. Mã hiển thị Led 7 thanh (7-segment display)

Đối với các ứng dụng dùng hiển thị số liệu ra Led 7 đoạn, ta dùng mã hiển thị Led 7 đoạn (bảng 1.1).



Bảng 1.1: Bảng mã led 7 đoạn

Số thập phân	Số thập lục phân	Số nhị phân	Mã Led 7 đoạn	Hiển thị
			a b c d e f g	
0	0	0000	1 1 1 1 1 1 0	0
1	1	0001	0 1 1 0 0 0 0	1
2	2	0010	1 1 0 1 1 0 1	2
3	3	0011	1 1 1 1 0 1 1	3
4	4	0100	0 1 1 0 0 1 1	4
5	5	0101	1 0 1 1 0 1 1	5
6	6	0110	1 0 1 1 1 1 1	6
7	7	0111	1 1 1 0 0 0 0	7
8	8	1000	1 1 1 1 1 1 1	8
9	9	1001	1 1 1 0 0 1 1	9
10	A	1010	1 1 1 1 1 0 1	A
11	B	1011	0 0 1 1 1 1 1	B
12	C	1100	0 0 0 1 1 0 1	C
13	D	1101	0 1 1 1 1 0 1	D
14	E	1110	1 1 0 1 1 1 1	E
15	F	1111	1 0 0 0 1 1 1	F

2. Chuyển đổi lẫn nhau giữa các hệ đếm

2.1. Hệ nhị phân và hệ thập phân

❖ Chuyển số nhị phân thành số thập phân:

Để chuyển một số nhị phân thành một số thập phân, ta chỉ cần nhân các chữ số của số nhị phân với giá trị thập phân của nó và cộng tất cả các giá trị lại.

Ví dụ: $1011.11 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 11.75$

❖ Chuyển số thập phân thành số nhị phân:

Để chuyển một số thập phân thành số nhị phân, ta dùng 2 phương pháp sau:

Phương pháp 1: Ta lấy số thập phân cần chuyển trừ đi 2^i trong đó 2^i là số lớn nhất nhỏ hơn hay bằng số thập phân cần chuyển. Sau đó, ta lại lấy kết quả này và thực hiện tương tự cho đến 2^0 thì dừng. Trong quá trình thực hiện, ta sẽ ghi lại các giá trị 0 hay 1 cho các bit tùy theo trường hợp số thập phân nhỏ hơn 2^i (0) hay lớn hơn 2^i (1).

Ví dụ: Xét số 21 thì số 2^i lớn nhất là 2^4

$$\begin{array}{ccccc} 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 16 & 8 & 4 & 2 & 1 \\ 21 = 2^4 + 0 \times 2^3 + 2^2 + 0 \times 2^1 + 2^0 = 16 + 0 + 4 + 1 \end{array}$$

Phương pháp 2: Lấy số cần chuyển chia cho 2, ta nhớ lại số dư và lấy tiếp thương của kết quả trên chia cho 2 và thực hiện tương tự cho đến khi thương cuối cùng bằng 0. Kết quả chuyển đổi sẽ là chuỗi các bit là các số dư lấy theo thứ tự ngược lại.

Ví dụ: Chuyển 227 ra số nhị phân

Số bị chia	Thương	Số dư
227	113	1 (LSB)
113	56	1
56	28	0
28	14	0
14	7	0
7	3	1
3	1	1
1	0	1 (MSB) (227 = 11100011b)

Để thực hiện chuyển các số thập phân nhỏ hơn 1 sang các số nhị phân, ta làm như sau: lấy số cần chuyển nhân với 2, giữ lại phần nguyên và lại lấy phần lẻ nhân với 2. Quá trình tiếp tục cho đến khi phần lẻ bằng 0 thì dừng. Kết quả chuyển đổi là chuỗi các bit là giá trị các phần nguyên.

Ví dụ: Chuyển 0.625 thành số nhị phân

$$0.625 \times 2 = 1.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

$$(0.625 = 0.101b)$$

2.2. Hệ nhị phân và hệ hexa

Cách chuyển nhị phân sang hex

Phương pháp: các bit nhị phân được nhóm 4 bit từ LSB, mỗi nhóm 4 bit được chuyển sang số hex tương ứng (bảng 1.1). Nếu số bit không đủ thì cộng thêm bit 0 vào MSB.

$$\text{Ví Dụ: } 11101001102 = \underbrace{0011}_3 \underbrace{1010}_A \underbrace{0110}_6 = 3A6_{16}$$

3. Biểu diễn thông tin trong các hệ vi xử lý

3.1. Mã hóa các thông tin không số

3.1.1. Mã hóa chữ và dữ liệu kiểu văn bản

Đơn vị cơ sở của dữ liệu văn bản là chữ. Chữ ở đây được hiểu theo nghĩa rộng, không chỉ là các chữ cái Latinh mà kể cả chữ số, các dấu chính tả, các dấu toán học, các kí hiệu để trình bày. Mặt khác không phải dân tộc nào cũng dùng chữ Latinh nên đối với một số dân tộc có thể có những chữ riêng. Chúng ta dùng thuật ngữ *ký tự* (character) với ý nghĩa là một ký hiệu dùng trong văn bản.

Nếu dùng một vùng nhớ k bit để mã hóa một chữ thì chỉ có thể biểu diễn tối đa được 2^k ký tự vì chỉ có thể tạo được đúng 2^k các mã nhị phân khác nhau. Điều này giải thích tại sao người Mỹ chỉ cần 7 bit để mã hóa cho các chữ của họ; để có thêm các mặt chữ Châu Âu, chữ Hy Lạp hay người Nhật phải dùng các mã 16 bit.

Các văn bản được hình dung như một chuỗi ký tự. Nội dung một cuốn sách, một bài thơ được đưa vào máy tính là những ví dụ cụ thể về thông tin văn bản. Hầu hết các máy tính và môi trường lập trình hiện nay đều sử dụng một byte để mã hóa một chữ.

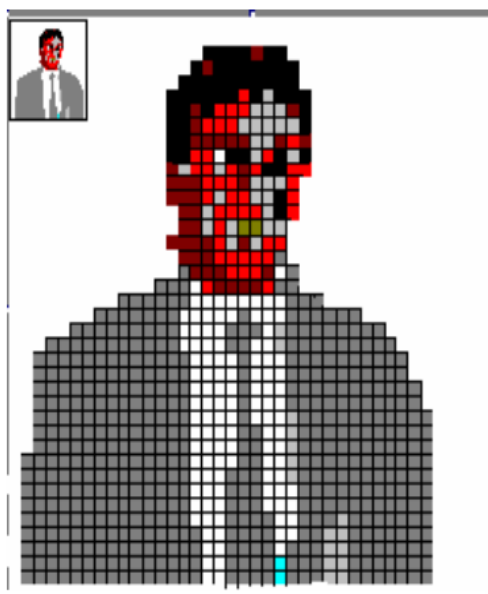
Về nguyên tắc có thể mã hóa giá trị sai hay không bởi bit0, giá trị đúng hay không bởi bit1. Tuy nhiên ít khi người ta sử dụng tới mức bit vì cơ chế địa chỉ hóa thường ít nhất ở mức byte. Khi đó người ta vẫn dùng một byte để mã hóa các giá trị logic.

3.1.2. Các dữ liệu logic

Dữ liệu loại logic chỉ có thể hiện một trong hai trạng thái đối lập là đúng/sai, hoặc có/không. Điều này ta thường thấy trong rất nhiều loại hồ sơ. Ví dụ: Trong lý lịch cá nhân: họ tên, quê quán là dữ liệu kiểu văn bản, ngày tháng năm sinh, lương có thể hiện bằng số, còn các thông tin như là đoàn viên thì không, có gia đình hay không là các thông tin có kiểu logic. Các thông tin kiểu logic chịu tác động của các phép toán so sánh, các phép toán nhân logic “và”, cộng logic hoặc hay phủ định logic “không”.

Về nguyên tắc có thể mã hóa giá trị sai hay không bởi bit0, giá trị đúng hay có bởi bit1. Tuy nhiên ít khi người ta sử dụng tới mức bit vì cơ chế địa chỉ hóa thường ít nhất ở mức byte. Khi đó người ta vẫn dùng một byte để mã hóa các giá trị logic.

3.1.3. Hình ảnh



Hình 1.1. Ảnh bitmap

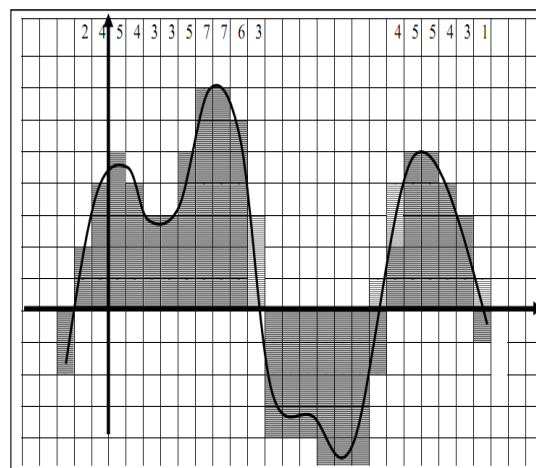
Hình ảnh cũng có thể xử lý bằng máy tính. Khác với hình ảnh thông thường, hình ảnh trong máy tính được mã hóa dưới dạng nhị phân. Có rất nhiều kiểu mã hóa ảnh trong đó hai kiểu thông dụng nhất là Ảnh bitmap (nghĩa là bản đồ các bit) thể hiện ảnh như một lưới điểm. Như vậy mỗi điểm sẽ phải nằm trong một hàng và một cột nào đó trong lưới, ngoài ra màu của điểm cũng được mã hóa.

Các ảnh khí tượng do các vệ tinh chụp gửi về, ảnh phong cảnh, chân dung đều có thể thể hiện theo kiểu này. Ta cũng có thể đưa một ảnh bất kỳ vào máy dưới dạng bitmap bằng máy quét ảnh (scanner), máy quay video số (digital video camera) hay máy chụp ảnh số (digital camera)... Nói chung dữ liệu ảnh này là dữ liệu lớn. Vì vậy, người ta thường sử dụng các kỹ thuật nén ảnh trước khi đưa vào máy lưu trữ và khôi phục ảnh khi trình bày. Có rất nhiều chuẩn ảnh khác nhau, chủ yếu khác nhau ở cách tổ chức để nén được ảnh mà vẫn giữ được chất lượng và thể hiện được các hiệu ứng ảnh. Còn lúc hiển thị để xem thì ảnh sẽ được khôi phục dưới dạng bitmap. Ảnh thể hiện theo từng điểm còn gọi là ảnh raster.

Kiểu thứ 2 thể hiện ảnh theo cách vẽ. Kiểu này chỉ phù hợp với các ảnh có thành phần là các điểm rời rạc, các đường hoặc hình thể hiện bằng các đường biên như bản vẽ kiến trúc các bản vẽ kỹ thuật, bản đồ. Cách lưu trữ là lưu thông tin về các thành phần của ảnh. Đối với một đoạn thẳng thì chỉ lưu tọa độ các đầu mút, đối với một hình tròn thì chỉ lưu tọa độ tâm và bán kính... Vì thế các ảnh này thường gọn gàng và dễ phóng to thu nhỏ (vì chỉ dùng các phép biến đổi tọa độ). Các ảnh kiểu này gọi là ảnh vector.

3.1.4. Âm thanh

Âm thanh cũng có thể được xử lý bằng máy tính. Cũng có nhiều phương pháp mã hóa âm thanh. Cách đơn giản nhất là mã hóa bằng cách xấp xỉ dao động sóng âm bằng một chuỗi các byte thể hiện biên độ dao động tương ứng theo từng khoảng thời gian bằng nhau. Dĩ nhiên các đơn vị thời gian này cần phải đủ nhỏ để không làm nghèo âm thanh. Đơn vị thời gian này gọi là chu kỳ lấy mẫu. Hình vẽ dưới đây minh họa cách lưu trữ xấp xỉ sóng âm, theo đó sẽ lưu lại dãy các giá trị sau (xem hình 1.2).



Hình 1.2. Số hóa âm thanh

Khi phát, một mạch điện sẽ khôi phục lại sóng âm với một sai lệch chấp nhận được.

Một cách khác là phân tích dao động âm thanh thành tổng các dao động điều hòa (các dao động hình sin với tần số và biên độ khác nhau) và chỉ lưu lại các đặc trưng về tần số và biên độ.

Còn có nhiều cách mã hóa âm thanh dựa theo những nguyên lý nén dữ liệu rất hiệu quả. Việc số hóa âm thanh cũng được thực hiện nhờ các thiết bị chuyên dụng. Xử lý âm thanh trên máy tính gồm những việc sau:

Thu và mã hóa âm thanh.

Biên tập (sửa chữa, ghép, cắt).

Phân tích (tìm các đặc trưng để nhận dạng tiếng nói). Một số máy tính đã có thể nghe được các lệnh đơn giản. Các máy điện thoại di động hiện nay đã có khả năng nhận dạng tiếng nói.

Tổng hợp tiếng nói. Ở mức độ đơn giản máy tính có thể đọc văn bản thành lời.

3.2. Mã hóa các thông tin số

Nếu dùng 1 byte (8 bit) để biểu diễn các số nguyên không dấu (số tự nhiên) thì ta có thể biểu diễn được $2^8 = 256$ số từ 0 đến 255.

Mã nhị phân	Số
0000 0000	0
0000 0001	1
0000 0010	2
.....	
1111 1111	255

Nếu dùng 1 byte để biểu diễn các số nguyên có dấu thì có thể biểu diễn được $2^8 = 256$ số từ -128 đến 127. Do đó khoảng số nguyên biểu diễn được là từ -128 đến 127 là vì phải dùng mất 1 bit để biểu diễn dấu. Bit biểu diễn dấu là bit MSB. **MSB = 1 cho số âm, MSB = 0 cho số dương.** Khi MSB = 0 thì ta có số dương lớn nhất là 0111 1111 = 127. Khi MSB = 1 ta có số âm, số âm có nhiều cách biểu diễn nhưng hay dùng nhất là kiểu số bù hai. Với kiểu số bù hai số âm nhỏ nhất là 1000 0000 = -128 (ta sẽ nói cách tìm số -128 sau).

❖ Cách tìm số bù hai biểu diễn một số âm như sau:

- Đổi trị tuyệt đối của số âm ra mã nhị phân (với độ dài từ mã cố định, ví dụ 8, 16, 32).
- Tìm số bù một của số nhị phân bằng cách đảo các bit: 0 thành 1, 1 thành 0.
- Cộng số bù một với 1 ta sẽ được số bù hai.

Ví dụ: Biểu diễn số -128 trong dạng số bù hai 8 bit

$$128_{10} = 1000\ 0000_2$$

$$\text{Bù một} = 0111\ 1111$$

$$+ \quad \quad \quad 1$$

$$\text{Bù hai} = 1000\ 0000$$

Ngược lại, từ một số nhị phân hãy tìm số nguyên mà nó biểu diễn (giả sử số nhị phân biểu diễn số nguyên có dấu). Khi đó ta làm như sau:

- Xét bit MSB. Nếu MSB = 0 thì chỉ cần đổi số nhị phân sang hệ mười. Nếu MSB = 1 thì chuyển sang bước tiếp theo.
- Tìm số bù một
- Lấy số bù một cộng với 1 để được số bù hai.
- Đổi số nhị phân bù hai sang hệ mười.

Ví dụ: Cho số nhị phân 1101 1001, số này biểu diễn một số nguyên có dấu, hãy tìm số nguyên đó.

Bit MSB = 1 => Số biểu diễn là số âm.

$$\text{Số bù một} = 0010\ 0110$$

$$\text{Số bù hai} = 0010\ 0111$$

$$\text{Số nguyên cần tìm là } -39$$

3.2.1. Mã ASCII

Thông tin, dữ liệu xung quanh chúng ta có rất nhiều loại như văn bản, số liệu, âm thanh, hình ảnh,... Muốn đưa các loại thông tin này vào máy tính chúng ta phải dùng mã nhị phân để biểu diễn. Trong thực tế thông tin được truyền đi, được lưu giữ trong các bộ nhớ máy tính hoặc để hiển thị trên màn hình đều ở dưới dạng ký tự và tuân theo một loại mã được dùng rộng rãi trên thế giới gọi là mã ASCII (American Standard Code for Information Interchange, mã chuẩn của Mỹ dùng để trao đổi thông tin). Việc dùng các ký tự để mã hóa thông tin theo bảng mã ASCII cho phép các máy tính và các bộ phận của một máy tính có thể trao đổi thông tin với nhau.

Các từ mã trong bảng mã ASCII có độ dài 1 byte (8 bit). Bảng mã ASCII được chia thành 2 nửa: nửa đầu có mã từ 0 đến 127 gọi là bảng mã ASCII tiêu chuẩn, nửa sau có mã từ 128 đến 255 gọi là bảng mã ASCII mở rộng. Hầu hết các nước trên thế giới có bảng mã ASCII tiêu chuẩn giống nhau. Bảng mã ASCII mở rộng thường khác nhau và dùng để chứa bộ ký tự của riêng từng nước.

Trong bảng mã ASCII tiêu chuẩn người ta chỉ dùng hết 7 bit (từ bit 0 đến bit 6) để mã hóa các ký tự, còn 1 bit MSB có thể cho liên tục bằng 0, hoặc bằng 1, hoặc có thể dùng để chứa bit parity phục vụ việc phát hiện lỗi khi truyền.

Khi xem xét bảng mã ASCII tiêu chuẩn ta có thể rút ra mấy nhận xét sau:

- + Hai cột đầu (cột 0 và cột 1) của bảng mã dùng cho các ký tự điều khiển.
- + Các ký tự số nằm ở cột 3. Mã của ký tự '0' là 30H, của ký tự '9' là 39H. Giữa giá trị số và mã ASCII của số đó có khoảng cách là 30H.
- + Các chữ cái hoa nằm ở cột 4 và 5, các chữ cái thường nằm ở cột 6 và 7. Khoảng cách giữa chữ thường và chữ hoa cùng tên là 20H.

Bảng 1.2. BẢNG MÃ ASCII TIÊU CHUẨN

Hexa 1 Hexa 2	0	1	2	3	4	5	6	7
0	<NULL> 0	<DLE> 16	<SP> 32	0 48	@ 64	P 80	` 96	p 112
1	<SOH> 1	<DC1> 17	! 33	1 49	A 65	Q 81	a 97	q 113
2	<STX> 2	<DC2> 18	" 34	2 50	B 66	R 82	b 98	r 114
3	<ETX> 3	<DC3> 19	# 35	3 51	C 67	S 83	c 99	s 115
4	<EOT> 4	<DC4> 20	\$ 36	4 52	D 68	T 84	d 100	t 116
5	<ENQ> 5	<NAK> 21	% 37	5 53	E 69	U 85	e 101	u 117
6	<ACK> 6	<SYN> 22	& 38	6 54	F 70	V 86	f 102	v 118
7	<BEL> 7	<ETB> 23	' 39	7 55	G 71	W 87	g 103	w 119
8	<BS> 8	<CAN> 24	(40	8 56	H 72	X 88	h 104	x 120