



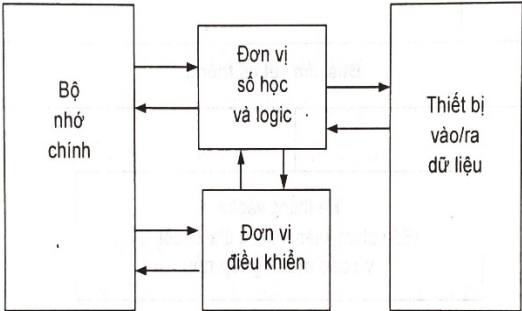
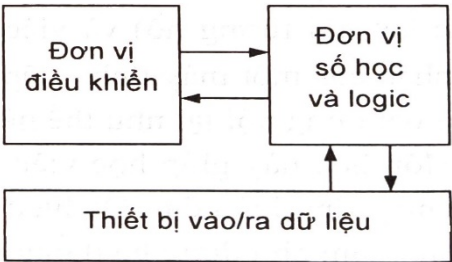
Ôn tập KTMT C1,2 - ktmt

Tiếng anh CNTT (Trường Đại học Công nghiệp Hà Nội)

CHƯƠNG 1: TỔNG QUAN VỀ KIẾN TRÚC MÁY TÍNH

Câu 1.1:

Phân biệt kiến trúc máy tính Von Neumann và Non Von Neumann.

Kiến trúc máy tính Von Neumann	Kiến trúc máy tính Non Von Neumann
Cuối năm 1945, Von Neumann đưa ra ý tưởng thiết kế máy tính có chương trình được lưu trữ bên trong: Có một bộ nhớ chính: để lưu trữ dữ liệu và chương trình đang thực thi.	Máy tính không có bộ nhớ chính lưu trữ chương trình bên trong
Sơ đồ khối kiến trúc	
	

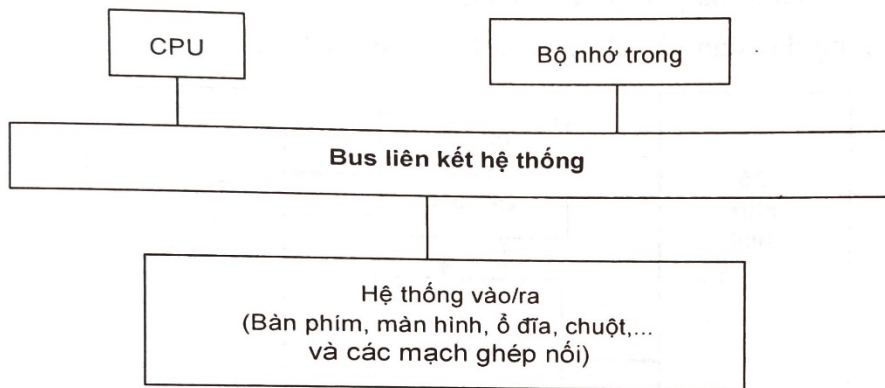
Câu 1.2:

Vẽ sơ đồ khối cấu trúc chung của máy tính IBM-PC và trình bày tóm tắt chức năng các khối.

Gợi ý:

Cấu trúc chung của hệ thống máy tính IBM-PC hoặc tương thích IBM-PC bao gồm 4 thành phần cơ bản (hình 1.3): Đơn vị xử lý trung tâm (Central Processing Unit – CPU), bộ nhớ trong (Internal Memory), hệ thống vào ra (Input/Output System - I/O System), bus liên kết hệ thống (System Bus).

Bài Làm



Câu 1.3:

Trình bày chức năng, đặc điểm của bus địa chỉ.

-Chức năng của bus địa chỉ: vận chuyển tín hiệu địa chỉ từ CPU đến bộ nhớ hay cổng vào/ra để xác định ngăn nhớ hay cổng vào/ra nào cần trao đổi thông tin.

-Đặc điểm của bus địa chỉ: có độ rộng và chỉ vận chuyển tín hiệu điện theo một chiều, đó là từ CPU bên ngoài. Trong quá trình hoạt động, CPU sẽ điều khiển bus địa chỉ để truyền dữ liệu giữa các khu vực.

Câu 1.4:

Cho biết ý nghĩa khi nói Bus địa chỉ có độ rộng 24 bit.

- Bus gồm 24 đường dây dẫn
- 24 bit này được dùng để đánh địa chỉ
- Có khả năng quản lý tối đa giá trị địa chỉ ngăn nhớ hay byte nhớ (vì bộ nhớ chính quản lý theo byte): quản lý tối đa byte = $2^{24} = 16\text{MB}$.
- Bus địa chỉ: vận chuyển tín hiệu địa chỉ từ CPU đến bộ nhớ hay cổng vào/ra để xác định ngăn nhớ hay cổng vào/ra nào cần trao đổi thông tin.
- Bus địa chỉ: có độ rộng và chỉ vận chuyển tín hiệu điện theo một chiều, đó là từ CPU bên ngoài. Trong quá trình hoạt động, CPU sẽ điều khiển bus địa chỉ để truyền dữ liệu giữa các khu vực.

Câu 1.5:

- Chức năng của bus dữ liệu:
- + Vận chuyển lệnh từ bộ nhớ đến CPU.

- + Vận chuyển dữ liệu giữa các thành phần khác nhau trong phạm vi máy tính như CPU, bộ nhớ, hệ thống vào/ra.
- Đặc điểm của bus dữ liệu:
 - + Có độ rộng và vận chuyển tín hiệu điện theo cả hai chiều.
 - + Ký hiệu: \div thì độ rộng bus là m bit (m thường có giá trị: 8, 16, 32, 64).

Câu 1.6:

- Ý nghĩa khi nói Bus dữ liệu có độ rộng 32 bit:
 - + Bus gồm 32 đường dây dẫn, CPU có khả năng xử lý toán hạng 32 bit trong 1 chu kỳ lệnh.
 - + Tức là vận chuyển một lúc 4 byte.
- Bus dữ liệu có chức năng:
 - + Vận chuyển lệnh từ bộ nhớ đến CPU.
 - + Vận chuyển dữ liệu giữa các thành phần khác nhau trong phạm vi máy tính như CPU, bộ nhớ, hệ thống vào/ra.
- Đặc điểm của bus dữ liệu:
 - + Có độ rộng và vận chuyển tín hiệu điện theo cả hai chiều.
 - + Ký hiệu: \div thì độ rộng bus là m bit (m thường có giá trị: 8, 16, 32, 64).

Câu 1.7:

- Chức năng của bus điều khiển: vận chuyển các tín hiệu điều khiển
- Đặc điểm của bus điều khiển:
 - + Là tập hợp các tín hiệu điều khiển riêng lẻ, hoặc là phát ra từ CPU để điều khiển đọc/ghi bộ nhớ hay hệ thống vào/ra, hoặc là phát ra từ bộ nhớ hay hệ thống vào/ra đến yêu cầu CPU.

Câu 1.8 và 1.9:

Trình bày hiểu biết của em về tín hiệu điều khiển đọc/ghi bộ nhớ hay thiết bị vào/ra.

- Có 4 tín hiệu điều khiển cơ bản:
 - + Memory Read (MEMR): Là tín hiệu điều khiển đọc dữ liệu từ một ngăn nhớ có địa chỉ xác định đưa lên bus dữ liệu. Khi bus địa chỉ xác định ngăn nhớ cần đọc thì MEMR sẽ điều khiển mở ngăn nhớ để đưa dữ liệu lên Bus dữ liệu.

+ Memory Write (MEMW): Điều khiển ghi dữ liệu có sẵn trên bus dữ liệu đến ngăn nhớ có địa chỉ xác định. Khi bus địa chỉ xác định ngăn nhớ cần ghi và dữ liệu trên bus dữ liệu đã ổn định thì MEMW sẽ điều khiển mở ngăn nhớ để đưa dữ liệu từ bus dữ liệu vào ngăn nhớ.

+ Input Output Read (IOR): Là tín hiệu điều khiển đọc dữ liệu từ cổng.

+ Input Output Write (IOW): Là tín hiệu điều khiển ghi dữ liệu ra cổng.

Câu 1.10:

Trình bày 3 tín hiệu điều khiển ngắt cứng điển hình (NMI, INTR, INTA).

- Non Maskable Interrupt (NMI) – tín hiệu ngắt không che được: Nó là tín hiệu từ mạch bên ngoài gửi đến chân NMI của CPU và CPU phải ngắt ngay sau khi hoàn thành lệnh đang thực hiện.
- Interrupt Request (INTR) – tín hiệu ngắt che được: Là tín hiệu được gửi từ mạch điều khiển ưu tiên ngắt gửi đến chân INTR của CPU yêu cầu CPU ngắt, CPU có thể ngắt, có thể không.
- Interrupt Acknowledge (INTA): Là tín hiệu phát ra từ chân INTA của CPU, báo cho mạch bên ngoài biết CPU có chấp nhận ngắt hay không.

CHƯƠNG 2: KIẾN TRÚC TẬP LỆNH

Câu 2.2: Đổi số 12035.125 về hệ 2 :

- Đổi số 12035 về hệ 16 :

12035	16				
3	752	16			
	0	47	16		
		15	2	16	
			2	0	

→ 12035 = 2F03 H

- Đổi số 2F03 H về hệ 2 ta được : 0010111100000011 B
- Đổi 0.125 về hệ 2 : $0.125 \times 2 = 0.25$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

→ 0.125 = 001 B

→ 12035.125 = 0010111100000011.001 B

Câu 2.3: Đổi số 7899.8 về hệ 2

- Đổi số 7899 về hệ 16 :

7899	16				
11	493	16			
	13	30	16		
		14	1	16	
			1	0	

→ 7899 = 1EDB H

- Đổi số 1EDB H về hệ 2 ta được : 0001111011011011 B
- Đổi 0.8 về hệ 2 : $0.8 \times 2 = 1.6$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

.....

→ 0.8 = 11001100.... B

→ 7899.8 = 0001111011011011.11001100... B

Câu 2.4:

Đổi số 10110010.11B về hệ 10.

Gợi ý:

Đổi số Q hệ cơ số a về số P hệ cơ số 10: ta thực hiện theo công thức sau.

$$Q = m_{k-1} m_k \dots m_1 m_0 m_{-1} m_{-n+k}$$

$$P = a^{k-1} \times m_{k-1} + a^{k-2} \times m_{k-2} + \dots + a^1 \times m_1 + m_0 + a^{-1} \times m_{-1} + \dots + a^{-n+k} \times m_{-n+k} \quad (2.1)$$

Ta có thể thực hiện theo 2 cách:

Cách 1: Đổi số đã cho qua số hệ 16, rồi đổi tiếp từ số hệ 16 về hệ 2

- Đổi số 10110010.11B về hệ 16, ta có:

$$10110010.11B = \underline{1011}0010.\underline{1100}B = 0B2.BH$$

- Đổi số 0B2.BH về hệ 10 theo công thức (2.1), ta có:

$$11 * 16^1 + 2 * 16^0 + 11 * 16^{-1} = 176 + 2 + 0.75 = 178.75$$

Cách 2: Đổi số đã cho thẳng sang số hệ 10 (theo công thức 2.1)

$$10110010.11B = 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 2^{-1} + 2^{-2}$$

	3		
11		7	16
	7		0

Vậy $31634 = 7B92H$

Câu 2.9:

Đổi số nguyên thập phân sau ra số hex 16 bit: -16.

Gợi ý

- Số bù 2 chính là số nguyên có dấu, dấu của số được xác định bởi giá trị của MSB (xem hình 2.1- GT KTMT) – đây chính là một yếu tố để kiểm tra lại đáp án:

+ Nếu MSB = 0: là số dương, cách xác định giá trị theo công thức 2.1.

+ Nếu MSB = 1: là số âm.

- Số -16 là bù 2 của +16 và ngược lại, vì vậy ta cần biểu diễn số +16 ở dạng nhị phân đủ 16 bit, sau đó xác định số bù 2 của +16, giá trị nhận được chính là số -16 ở dạng hệ 2 đủ 16 bit, cuối cùng ta đổi giá trị này ra hệ 16

- Số hệ hex 16 bit phải là số có đủ 4 chữ số.

Bài làm

Số -16 là số bù 2 của +16 và ngược lại, vậy ta tìm số bù 2 của +16.

$$+16 = 0000000000010000B$$

$$\text{Số bù 1 (của +16)} = 111111111101111B$$

$$+ \underline{\hspace{10em} 1}$$

$$\text{Số bù 2 (của +16)} = 11111111110000B$$

$$\text{Vậy } -16 = 11111111110000B = FFF0H$$

Câu 2.10: Đổi số nguyên thập phân sau ra số hex 16 bit: -116.

Số -116 là số bù 2 của +116 và ngược lại, vậy ta tìm số bù 2 của +116.

$$+116 = 0000000001110100B$$

$$\text{Số bù 1 (của +116)} = 111111110001011B$$

$$+ \underline{\hspace{10em} 1}$$

$$\text{Số bù 2 (của +116)} = 111111110001100B$$

$$\text{Vậy } -116 = 111111110001100B = FF8CH$$

Câu 2.11: Biểu diễn số -67 ở dạng nhị phân 16 bit.

Số -67 là số bù 2 của +67 và ngược lại, vậy ta tìm số bù 2 của +67.

$$+67 = 0000000001000011B$$

$$\text{Số bù 1 (của +67)} = 111111110111100B$$

$$\begin{array}{r} + \quad \underline{\hspace{1cm}} 1 \\ \text{Số bù 2 (của +67)} = 111111110111101 \text{ B} \end{array}$$

(Nếu đề bài là Hex 16bit thì đáp án là -67 = 111111110111101 B = FFBD H)

Câu 2.12 :

Đổi -128 về dạng nhị phân.

Gợi ý

Giải quyết bài này, ta cần chú ý tới giới hạn biểu diễn dành cho số bù 2:

- Số bù 2 độ dài 8 bit: biểu diễn các số từ -128 ÷ +127.
- Số bù 2 độ dài 16 bit: biểu diễn các số từ -32768 ÷ +32767.

Và số bit để biểu số nguyên trong máy tính là (8×2^i) bits: do vậy biểu diễn số nguyên phải đủ 8 bit hoặc 16 bit hoặc 32 bit,...

Nhận xét:

Số bù 2 của -128 là +128 và ngược lại, số -128 thuộc miền giới hạn cho số 8 bit, xong nếu ta biểu diễn số +128 lại không thuộc miền giới hạn cho số 8 bit mà thuộc miền biểu diễn của số 16 bit. Vì vậy, ta phải biểu diễn đủ bằng 16 bit.

Bài làm

Cách làm sai:

Số bù 2 của -128 là +128 và ngược lại, vậy ta có:

$$128 = 10000000\text{B} \text{ (biểu diễn chỉ 8 bit)}$$

$$\text{Số bù 1 (của +128)} = 01111111\text{B}$$

$$+ \quad \underline{\hspace{1cm}} 1$$

$$\text{Số bù 2 (của +128)} = 10000000\text{B}$$

$$\text{Vậy } -128 = 10000000\text{B} = +128 \text{ (sai)}$$

Nguyên nhân:

1. Cách đặt vấn đề là cho $128 = 10000000\text{B}$ (chỉ đúng cho số nguyên không dấu), với số nguyên có dấu, thì ta thấy MSB = 1, đây là số âm.
2. Số 128 là số nguyên có dấu nên cần biểu diễn đủ 16 bit vì 8 bit không đủ để biểu diễn (nằm ngoài giới hạn biểu diễn dành cho số 8 bit).

Cách làm đúng:

Số -128 là số bù 2 của +128 và ngược lại, vậy ta tìm số bù 2 của +128.

$$+128 = 0000000010000000\text{B}$$

$$\text{Số bù 1 (của +128)} = 1111111101111111\text{B}$$

$$\begin{array}{r} + \quad \underline{\hspace{2cm}} 1 \\ \hline \end{array}$$

Số bù 2 (của +128) = 1111111110000000B

Vậy -128 = 1111111110000000B = 10000000B (biểu diễn 8 bit, MSB = 1)

Câu 2.13: Đổi -132 về dạng nhị phân.

Số -132 là số bù 2 của +132 và ngược lại, vậy ta tìm số bù 2 của +132.

$$+132 = 0000000010000100B$$

$$\text{Số bù 1 (của +132)} = 111111110111011B$$

$$+ \quad \underline{\hspace{2cm}} 1$$

$$\text{Số bù 2 (của +132)} = 111111110111100B$$

$$\text{Vậy -132} = 111111110111100B$$

Câu 2.14: Đổi -92 về dạng nhị phân.

Số -92 là số bù 2 của +92 và ngược lại, vậy ta tìm số bù 2 của +92.

$$+92 = 0000000001011100B$$

$$\text{Số bù 1 (của +92)} = 111111110100011B$$

$$+ \quad \underline{\hspace{2cm}} 1$$

$$\text{Số bù 2 (của +92)} = 111111110100100B$$

$$\text{Vậy -92} = 111111110100100B = 10100100B \text{ (biểu diễn 8 bit, MSB = 1)}$$

Câu 2.15:

Sau đây là tên và nội dung (dạng hex) của các thanh ghi 16 bit trong vi xử lý:

$$AX = 1234, BX = 900A, CX = FFFA, DX = 7FFF$$

Hãy tìm giá trị và dấu của giá trị trong chúng ở dạng hệ 10.

Gợi ý:

Thực chất đây là bài toán ngược, đổi các số nhị phân có dấu về hệ 10

Đầu tiên ta đổi giá trị trong thanh ghi ra dạng nhị phân, xác định dấu của số:

- Nếu MSB = 0, là số dương, đổi ra giá trị hệ 10 bình thường theo công thức 2.1.
- Nếu MSB = 1, là số âm, ta cần xác định trị tuyệt đối của nó ở dạng nhị phân (chính là số bù 2 của nó), đổi giá trị này về hệ 10 rồi thêm dấu âm trước số.

Bài làm

$$\begin{aligned} \text{a. Với } AX &= 1234H \\ &= 0001001000110100B \text{ (MSB = 0 là số dương)} \\ &= 2^{12} + 2^9 + 2^5 + 2^4 + 2^2 \end{aligned}$$

$$= 4096 + 512 + 32 + 16 + 4 = 4660$$

b. Với BX = 900AH

$$= 1001000000001010B$$

Gọi B là giá trị chứa trong BX, ta thấy MSB = 1, nên giá trị B trong BX là số âm, ta xác định số bù 2 của B

$$B = 1001000000001010B$$

$$\text{Số bù 1(B)} = 0110111111110101B$$

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

$$\text{Số bù 2(B)} = \underline{0110111111110101}B = 6FF6H$$

$$= 6 * 16^3 + 15 * 16^2 + 15 * 16 + 6$$

$$= 28662 = |B|$$

$$\text{Vậy giá trị trong BX} = B = 28662$$

c. Với CX = FFFAH

$$= 111111111111010B$$

Gọi B là giá trị chứa trong CX, ta thấy MSB = 1, nên giá trị B trong CX là số âm, ta xác định số bù 2 của B

$$B = 111111111111010B$$

$$\text{Số bù 1(B)} = 000000000000101B$$

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

$$\text{Số bù 2(B)} = 000000000000110B = 0006H$$

$$= 6 * 16^0 = 6 = |B|$$

$$\text{Vậy giá trị trong CX} = B = 6$$

d. Với DX = 7FFFH = 0111111111111111 B (MSB = 0 là số dương)

$$= \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times \times = 32767.$$

Câu 2.16:

Biểu diễn số +1022,9 ở dạng số dấu chấm động trong máy tính theo chuẩn IEEE 32 bit.

Gợi ý

Mục tiêu của bài toán: nắm được thuật toán mã hóa số thực hệ 10 để biểu diễn theo cấu trúc dữ liệu của số thực dấu chấm động lưu trữ trong máy tính, có thể sử dụng thuật toán này để viết chương trình dịch cho các ngôn ngữ lập trình để đưa số thực nhập từ bàn phím vào lưu trữ trong máy tính.

Dạng tổng quát của số dấu chấm động hệ 2 theo nguyên lý che số 1 như sau:

$$R = (-1)^S * 1.M * 2^{E - \text{Bias}} \quad (2.2)$$

Số thực dấu chấm (phẩy) động được biểu diễn bằng 2 phần chính và một bit dấu, theo khuôn dạng sau :

S	E	M
---	---	---

Hình 2.2. mô tả sắp xếp của một số dấu chấm (phẩy) động trong máy tính

Với số dấu chấm động 32 bit thì:

- S là bit dấu, chiếm 1 bit dấu.
- E là Exponent (phần mũ) luôn lớn hơn hoặc bằng 0, chiếm 8 bit.
- M là Mantissa (phần định trị) - chỉ lưu phần lẻ sau dấu chấm nhị phân, chiếm 23 bit.
- Vì phần mũ ($E - \text{Bias}$) có thể âm, hoặc dương, xong trong cấu trúc dữ liệu lưu trữ số dấu chấm động, chỉ được phép có một bit dấu của số (không được phép thêm bit dấu của số mũ) nên E phải trong khoảng $[0 \div 255]$. Vì vậy để khử giá trị âm cho phần mũ, ta sử dụng số Bias, $E < \text{Bias}$ thì phần mũ âm, $E > \text{Bias}$ thì phần mũ dương. Giá trị của Bias được xác định để cân đối cả 2 phần âm và dương. Vậy với E chiếm 8 bit thì $\text{Bias} = 2^8/2 - 1 = 127$
- Độ chính xác dữ liệu là 2^{-127} : vì $E \geq 0$, nên $E - \text{Bias} \geq -127$, do vậy giá trị dữ liệu biểu diễn nhỏ nhất với kiểu short real là 2^{-127} .

Để giải quyết bài toán này ta cần thực hiện theo 3 bước :

1. Đổi số thực hệ 10 về dạng nhị phân (không cần quan tâm đến dấu của số).
2. Chuẩn hóa về khuôn dạng như hình 2.2.
3. Ghép các giá trị tìm được (S, E, M) vào khuôn dạng và đổi về hệ 16 cho dễ xem.

Bài làm

Bước 1. Đổi số 1022.9 về hệ 2.

- a. Đổi phần nguyên hệ 10 về hệ 2 (sử dụng cách nhẩm nhanh)

$$1022 = (1024 - 1) - 1 = (2^{10} - 1) - 1 = 1111111111\text{B} - 1 = 1111111110\text{B}$$

- b. Đổi phần lẻ thập phân về phần lẻ nhị phân

$$\begin{array}{l} 0.9 \times 2 = 1.8 \\ 0.8 \times 2 = 1.6 \\ 0.6 \times 2 = 1.2 \\ 0.2 \times 2 = 0.4 \\ 0.4 \times 2 = 0.8 \end{array} \left. \vphantom{\begin{array}{l} 0.9 \times 2 = 1.8 \\ 0.8 \times 2 = 1.6 \\ 0.6 \times 2 = 1.2 \\ 0.2 \times 2 = 0.4 \\ 0.4 \times 2 = 0.8 \end{array}} \right\} \begin{array}{l} \text{Cụm số 1100 lặp} \\ \text{vô hạn tuần hoàn} \end{array}$$

$$\text{Vậy } 0.9 = 0.1\underline{1110}1110\underline{1110}1110\dots\text{B}$$

(Từng cụm 4 bit bắt đầu từ cụm gạch chân đầu tiên, thể hiện sự lặp vô hạn tuần hoàn, số bit ta lấy là bao nhiêu kể từ dấu chấm nhị phân phụ thuộc vào độ chính xác của giá trị cần lấy.)

c. Ghép 2 phần

$$1022.9 = 111111110.11100110011001100\dots B$$

Bước 2. Chuẩn hóa theo định dạng số dấu chấm động 32 bit.

$$1022.9 = 111111110.11100110011001100\dots B$$

$$= (-1)^0 \times 1.1111111011100110011001B \times 2^9$$

(Sau dấu chấm nhị phân lấy đủ 23 bit)

$$E - \text{Bias} = 9 \implies E = 9 + \text{Bias} = 9 + 127 = 128 + 8$$

$$E = 2^7 + 8 = 10000000B + 1000B = 10001000B$$

Vậy ta có :

$$S = 0.$$

$$E = 10001000B$$

$$M = 1111111011100110011001B$$

Bước 3. Ghép vào khuôn dạng.

$$1022.9 = 0100010001111111011100110011001B = 447FB999H$$

Câu 2.17: Biểu diễn số -1012,8 ở dạng số dấu chấm động trong máy tính theo chuẩn IEEE 32 bit.

- Đổi phần nguyên hệ 10 về hệ 2 (sử dụng cách nhẩm nhanh)

$$1012 = 1111110100 B$$

- Đổi phần lẻ thập phân về phần lẻ nhị phân

$$0.8 \times 2 = 1.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$0.8 \times 2 = 1.6$$

.....

$$\rightarrow 0.8 = 11001100\dots B$$

- Ghép 2 phần: $1012.8 = 1111110100.11001100\dots B$
- Chuẩn hóa theo định dạng số dấu chấm động 32 bit :

$$1012.8 = 1111110100.11001100\dots B$$

$$= \times 1.1111101001100110011001B \times$$

$$E - \text{Bias} = 9 \implies E = 9 + \text{Bias} = 9 + 127 = 128 + 8$$

$$E = 2^7 + 8 = 10000000B + 1000B = 10001000B$$

- Khi đó ta có : $S = 1$ (vì -1012,8 là số âm)

$$E = 10001000B$$

$$M = 11111010011001100110011$$

$$\rightarrow \text{Vậy } 1012.8 = 11000100011111010011001100110011B = C47D3333 H$$

Câu 2.18: Biểu diễn số +912,0625 ở dạng số dấu chấm động trong máy tính theo chuẩn IEEE 32 bit.

- Đổi phần nguyên hệ 10 về hệ 2 (sử dụng cách nhẩm nhanh)

$$912 = 1110010000 B$$

- Đổi phần lẻ thập phân về phần lẻ nhị phân

$$0.625 \times 2 = 1.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

$$\rightarrow 0.625 = 101 B$$

- Ghép 2 phần: $912.625 = 1110010000.101B$
- Chuẩn hóa theo định dạng số dấu chấm động 32 bit :

$$912.625 = 1110010000.101B$$

$$= \times 1.110010000101000000000000 B \times$$

$$E - \text{Bias} = 9 \Rightarrow E = 9 + \text{Bias} = 9 + 127 = 128 + 8$$

$$E = 2^7 + 8 = 10000000B + 1000B = 10001000B$$

- Khi đó ta có : $S = 0$ (vì +912,625 là số dương)

$$E = 10001000B$$

$$M = 110010000101000000000000$$

$$\rightarrow \text{Vậy } 912.625 = 01000100011001000010100000000000B = 44642800 H$$

Câu 2.19:

Cho biết số dấu chấm động có giá trị 447FB999H bằng bao nhiêu ở hệ 10.

Gợi ý

Mục tiêu của bài toán: nắm được thuật toán giải mã số dấu chấm động lưu trữ trong bộ nhớ ra số hệ 10, có thể sử dụng thuật toán này để viết chương trình dịch cho các ngôn ngữ lập trình để đọc số thực từ bộ nhớ rồi đưa ra màn hình ở dạng hệ 10.

Đây là bài toán ngược với bài toán xác định biểu diễn số thập phân ở dạng số dấu chấm động 32 bit. Để giải bài toán này ta cũng thực hiện theo 3 bước :

1. Đổi số đã cho ra số nhị phân, xác định các giá trị S, E-Bias, M
2. Đưa ra số nhị phân dấu chấm động ở dạng chuẩn hóa theo công thức 2.2. và chuyển về số nhị phân dấu chấm tĩnh.
3. Chuyển giá trị nhị phân dấu chấm tĩnh về hệ 10.

Bài làm

Bước 1. Đổi số đã cho ra số nhị phân, xác định các giá trị S, E-Bias, M

447FB999H = 0100010001111111011100110011001B.

Theo khuôn dạng lưu trữ, ta tách được:

$$S = 0.$$

$$E = 10001000B = 2^8 + 2^3 = 128 + 8 = 136$$

$$E - \text{Bias} = 136 - 127 = 9$$

$$M = 1111111011100110011001B$$

Bước 2. Đưa ra số nhị phân dấu chấm động ở dạng chuẩn hóa theo công thức 2.2. và chuyển về số nhị phân dấu chấm tĩnh.

$$\begin{aligned} \text{Ta có: } R &= (-1)^S \times 1.M \times 2^{E-\text{Bias}} \\ &= (-1)^0 \times 1.1111111011100110011001B \times 2^9 \\ &= 111111110.11100110011001B \end{aligned}$$

Bước 3. Chuyển giá trị nhị phân dấu chấm tĩnh về hệ 10.

a. Đổi phần nguyên nhị phân về phần nguyên thập phân.

$$111111110B = (2^{10} - 1) - 1 = 1024 - 1 - 1 = 1022$$

$$(\text{phương pháp nhẩm nhanh: } 111111111B = 2^{10} - 1)$$

b. Đổi phần lẻ nhị phân về phần lẻ thập phân

$$0.11100110011001B = 0.1110011001100100B$$

$$= 0.E664H$$

$$= 14 * 16^{-1} + 6 * 16^{-2} + 6 * 16^{-3} + 4 * 16^{-4}$$

$$= 0.875 + 0.0234375 + 0.001465 + 0.0000610$$

$$= 0.8999634$$

c. Ghép 2 phần nguyên và phần lẻ thập phân

$$\text{Vậy } 447FB999H = 1022.8999634 \approx 1022.9$$

Câu 2.20 :

Cho biết số dấu chấm động có giá trị 409CCCCCH bằng bao nhiêu ở hệ 10.

Bài làm

Bước 1. Đổi số đã cho ra số nhị phân, xác định các giá trị S , E -Bias, M
409CCCCCH = 01000000100111001100110011001100B.

Theo khuôn dạng lưu trữ, ta tách được:

$$S = 0.$$

$$E = 10000001B = 129$$

$$E - \text{Bias} = 129 - 127 = 2$$

$$M = 000000100111001100110011001100B$$

Bước 2. Đưa ra số nhị phân dấu chấm động ở dạng chuẩn hóa theo công thức 2.2. và chuyển về số nhị phân dấu chấm tĩnh.

$$\begin{aligned}\text{Ta có: } R &= (-1)^S \times 1.M \times 2^{E-\text{Bias}} \\ &= (-1)^0 \times 1.000000100111001100110011001100B \times 2^2 \\ &= 100.0000100111001100110011001100B\end{aligned}$$

Bước 3. Chuyển giá trị nhị phân dấu chấm tĩnh về hệ 10.

a. Đổi phần nguyên nhị phân về phần nguyên thập phân.

$$100B = 4$$

(phương pháp nhẩm nhanh: $1111111111B = 2^{10} - 1$)

b. Đổi phần lẻ nhị phân về phần lẻ thập phân

$$0.0000100111001100110011001100B$$

$$= 0.09CCCCCH$$

$$= 0 * 16^{-1} + 9 * 16^{-2} + 12 * 16^{-3} + 12 * 16^{-4} + 12 * 16^{-5} + 12 * 16^{-6} + 12 * 16^{-7}$$

$$= 0.03828124702$$

Ghép 2 phần nguyên và phần lẻ thập phân

$$\text{Vậy } 409CCCCCH = 4.03828124702$$

Câu 2.21 :

Cho biết số dấu chấm động có giá trị BF400000H bằng bao nhiêu ở hệ 10.

Bài làm

Bước 1. Đổi số đã cho ra số nhị phân, xác định các giá trị S , E -Bias, M

$$BF400000H = 10111111010000000000000000000000B.$$

Theo khuôn dạng lưu trữ, ta tách được:

$$S = 1.$$

$$E = 01111110B = 129$$

$$E - \text{Bias} = 129 - 127 = 2$$

M = 11111101000000000000000000000000B

Bước 2. Đưa ra số nhị phân dấu chấm động ở dạng chuẩn hóa theo công thức 2.2. và chuyển về số nhị phân dấu chấm tĩnh.

Ta có: $R = (-1)^s \times 1.M \times 2^{E-Bias}$
 $= (-1)^1 \times 1.11111101000000000000000000000000B \times 2^2$
 $= 111.111101000000000000000000000000B$

Bước 3. Chuyển giá trị nhị phân dấu chấm tĩnh về hệ 10.

a. Đổi phần nguyên nhị phân về phần nguyên thập phân.

$111B = 7$

(phương pháp nhẩm nhanh: $1111111111B = 2^{10} - 1$)

b. Đổi phần lẻ nhị phân về phần lẻ thập phân

$0.111101000000000000000000000000B$

$= 0.F400000H$

$= 15 * 16^{-1} + 4 * 16^{-2} + 0 * 16^{-3} + 0 * 16^{-4} + 0 * 16^{-5} + 0 * 16^{-6} + 0 * 16^{-7}$

$= 0.953125$

Ghép 2 phần nguyên và phần lẻ thập phân

Vậy $BF400000H = 7.953125$

Câu 2.22:

Giả sử chuỗi “S12.75” đang lưu trong bộ nhớ bắt đầu tại địa chỉ 0, cho biết nội dung của các byte từ 0 đến 5 dưới dạng số hex.

Gợi ý

Các ô nhớ có địa chỉ từ 0 đến 5 đang lưu trữ các ký tự, nội dung của các ô nhớ đó ở dạng số nhị phân mã hóa các ký tự, các số nhị phân này sẽ tương đương với các số hệ hex trong bảng mã ASCII. Do vậy tra bảng mã ASCII tại trang 42 GT KTMT.

Bài làm.

Theo bảng mã ASCII ta có :

$S12.75 = 53\ 31\ 32\ 2E\ 37\ 35\ (H)$

Câu 2.23:

Hãy dịch thông điệp đã mã hóa dưới dạng mã ASCII (hệ hex) sau đây:

$41\ 74\ 74\ 61\ 63\ 6B\ 20\ 61\ 77\ 6E$

Gợi ý

Tra bảng mã ASCII, ta thấy giá trị 41H, mã hóa ký tự “A”, tương tự tra cho các giá trị khác.

Bài làm

Theo bảng mã ASCII ta có :

41 74 74 61 63 6B 20 61 77 6E
= A t t a c k () a w n

Câu 2.24:

Giả sử một byte có mã ASCII của một chữ hoa, hỏi phải cộng thêm một số hex là bao nhiêu để đổi nó thành chữ thường.

Gợi ý

Tra bảng mã ASCII cho các ký tự hoa, ký tự thường, so sánh rồi cho nhận xét.

Bài làm

Cộng thêm 20H (32 hệ thập phân)

Câu 2.25:

Giả sử một byte có nội dung là mã ASCII mã hóa một số thập phân từ 0 đến 9, hỏi phải trừ đi một số hex là bao nhiêu để đổi nó thành chính số đó.

Gợi ý

Tra bảng mã ASCII cho các ký tự số, so sánh với giá trị số tương ứng rồi cho nhận xét.

Bài làm

Trừ 30H.

Câu 2.26 :

Thực hiện trừ 2 số nhị phân sau: 00001110B – 01111111B

Gợi ý

Trong máy tính không có bộ trừ, chỉ có bộ cộng, vì vậy phép toán

$$A = B - C$$

Được thực hiện ở dạng: $A = B + (-C)$, trong đó $-C$ là số bù 2 của C .

Bài làm

Đặt $B = 00001110B$, $C = 01111111B$

Ta có : $A = B - C = B + (-C)$

Ta có: $+C = 01111111B$

Số bù 1(của $+C$) = $10000000B$

+ 1

Số bù 2(của $+C$) = $10000001B$

Vậy $-C = 10000001B$

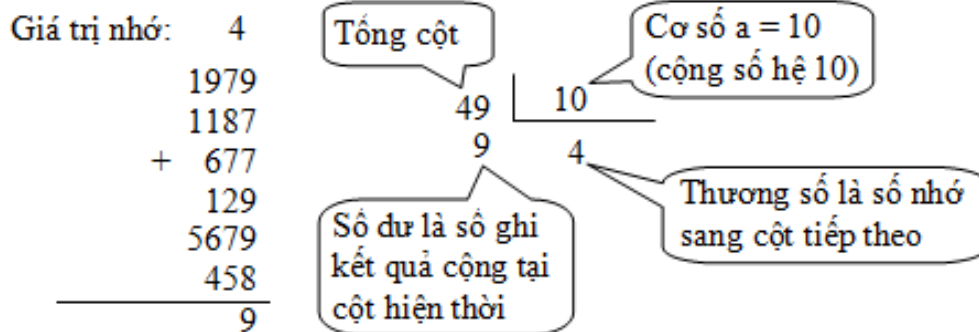
Vậy $A = B + (-C) = 00001110B + 10000001B = 10001111B$

Câu 2.27 :

Cộng các số nhị phân sau: $01111011B$, $11001111B$, $10111111B$, $00111101B$, $01111111B$, $01111110B$

Gợi ý

- Ta có thể cộng quán chiều từng cặp 2 số nhị phân một, xong mất thời gian, dễ nhầm lẫn
 - Ta thực hiện theo phương pháp tổng quát cộng số ở hệ cơ số a như sau:
 - + Cộng tất cả các giá trị số theo từng cột, bắt đầu từ cột có trọng số nhỏ nhất, thao tác cộng bình thường như cộng số hệ 10 ta thường làm.
 - + Ta lấy tổng cột có trọng số nhỏ nhất, chia cho a (a là cơ số - ví dụ cộng hệ 10 thì chia cho 10, cộng ở hệ 2 thì chia cho 2,...), kết quả: thương sẽ là số nhớ để cộng sang cột có trọng số cao hơn tiếp theo, số dư sẽ là kết quả ghi cho cột.
 - + Cứ như vậy ta thực hiện cộng cho cột có trọng số cao hơn tiếp theo...
- Ví dụ sau thể hiện với số hệ 10 tại cột có trọng số thấp nhất



Và ở số hệ 2, ta giải quyết bài toán này như sau: đầu tiên cộng các số nhị phân ở cột có trọng số 0, ta được 5, lấy 5 chia 2 (2 là cơ số) được 2 (là số nhớ của cột tiếp theo), dư 1 (ghi cho kết quả cộng cột). Tương tự ta cộng các cột tiếp theo như sau:

Giá trị nhớ: 134555432

```
01111011b
11001111b
+ 10111111b
00111101b
01111111b
01111110b
-----
1101000011b
```

Tổng cột

5

2

1

2

Cơ số a = 2
(cộng số hệ 2)

Số dư là số ghi
kết quả cộng tại
cột hiện thời

Thương số là số nhớ
sang cột tiếp theo

Câu 2.28 :

Cộng các số nhị phân sau: 11101101B, 11011110B, 10111111B, 10101101B, 01111101B

Bài làm

Giá trị nhớ: 133334322

```
11101101B
11011110B
+ 10111111B
10101101B
01111101B
= 1110110100B
```

Câu 2.29 :

Thực hiện nhân 2 số nhị phân 8 bit sau: 11101111B x 11111101B

Gợi ý

- Nhân 2 số đã cho như nhân với số hệ 10.
- Thực hiện cộng các bit trong các cột tương ứng như bài 2.26.

Lưu ý :

- Nhân 2 số 8 bit cho kết quả là số 16 bit.
- Nhân 2 số 16 bit cho kết quả là số 32 bit

Bài làm

11101111B

×

11111101B

$$\begin{array}{r}
 \hline
 11101111B \\
 00000000B \\
 11101111B \\
 11101111B \\
 + \quad 11101111B \\
 11101111B \\
 11101111B \\
 11101111B \\
 \hline
 = \quad \underline{1110110000110011B}
 \end{array}$$

Giá trị nhớ: 123345554322100

Câu 2.30 :

Nêu tên và công dụng của các thanh ghi trong hệ thống KTMT nói chung.

Bài làm

- Có 3 loại thanh ghi:
 - + Thanh ghi công dụng chung
 - + Thanh ghi địa chỉ
 - + Thanh ghi trạng thái
 - Công dụng:
 - + Thanh ghi công dụng chung: Có thể dùng để chứa toán tử cho các phép toán số học và logic và cũng có thể dùng để chứa địa chỉ của ô nhớ trong một số chế độ địa chỉ. Ví dụ: Thanh ghi BX, SI, DI
 - + Thanh ghi địa chỉ:
 - Thanh ghi đoạn
 - Thanh ghi chỉ số
 - Thanh ghi con trỏ
 - + Thanh ghi trạng thái:
 - Program counter (PC) : Chứa địa chỉ của lệnh sẽ được nạp vào thanh ghi lệnh
 - Instruction register (IR): Chứa lệnh vừa được nạp

- Memory address register (MAR): Chia địa chỉ của ô nhớ sẽ được đọc hoặc ghi
- Memory buffer register (MBR): Chứa dữ liệu sẽ được ghi ra bộ nhớ hoặc dữ liệu vừa được đọc vào từ bộ nhớ.

Câu 2.31 :

Nêu tên và tóm tắt chức năng các cờ trong thanh ghi cờ của CPU 8086.

Bài làm

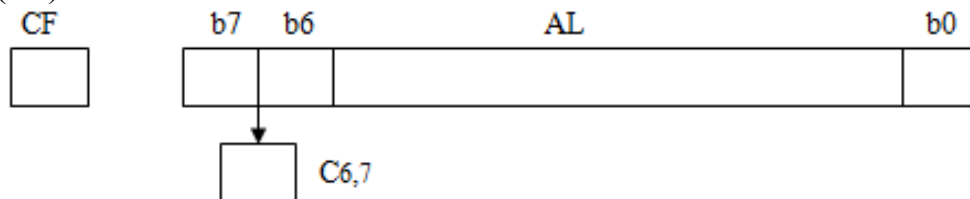
- Cờ Zero (cờ rỗng) : được thiết lập lên 1 khi kết quả của phép toán bằng 0
- Cờ Sign (cờ dấu) : được thiết lập lên 1 khi kết quả của phép toán nhỏ hơn 0
- Cờ Carry (cờ nhớ) : được thiết lập lên 1 nếu phép toán có nhớ ra ngoài bit cao nhất -> cờ báo tràn với số không dấu
- Cờ Overflow (cờ tràn) : được thiết lập lên 1 nếu cộng hai số nguyên cùng dấu mà kết quả có dấu ngược lại -> cờ báo tràn với số có dấu

Câu 2.32 :

Xác định giá trị của cờ tràn cho phép toán sau: $01110001B + 00100011B$.

Gợi ý

- Giả thiết ta làm việc với số bù 2 dài 8 Bit, kết quả để ở AL. Gọi $C_{6,7}$ là cờ nhớ từ Bit 6 lên Bit 7 (b7), trong đó b7 là MSB và cũng chính là Bit dấu (SF) của AL



Quan hệ giữa cờ OF với cờ $C_{6,7}$ tuân theo phương trình sau:

$$OF = CF \oplus C_{6,7} \quad (\oplus - \text{phép XOR})$$

Nghĩa là: khi thực hiện các phép toán với số bù 2 (số có dấu), hiện tượng tràn sẽ xảy ra và thể hiện qua cờ $OF = 1$: nếu có nhớ từ MSB (tức là SF) sang CF và không có nhớ từ $C_{6,7}$ vào chính nó (SF) hoặc ngược lại.

- Tương tự với số 16 bit: $OF = CF \oplus C_{14,15}$

Bài làm

$$\begin{array}{r} 01110001B \\ + 00100011B \\ \hline 10010100B \end{array}$$

Ta thấy: $CF = 0$ (Không có nhớ từ MSB ra ngoài).
 $C_{6,7} = 1$ (có nhớ từ bit 6 sang bit 7)

$$\text{Vậy } OF = CF \oplus C_{6,7} = 0 \oplus 1 = 1$$

Lưu ý: Khi $OF = 1$, hiện tượng tràn số đã xảy ra, cụ thể ta thấy kết quả phép toán trên là sai cho số có dấu (cộng 2 số dương – $MSB = 0$, cho kết quả là số âm – $MSB = 1$)

Câu 2.33 :

Xác định giá trị của cờ tràn cho phép toán sau: $11110001B + 01000010B$.

Bài làm

$$\begin{array}{r} 11110001B \\ + \\ 01000010B \\ \hline = 100110011B \end{array}$$

Ta thấy : $CF = 1$ (có nhớ từ MSB ra ngoài)

$C_{6,7} = 0$ (không có nhớ từ bit 6 sang bit 7)

$$\text{Vậy } OF = CF \oplus C_{6,7} = 1 \oplus 0 = 1$$

Câu 2.34 :

Xác định giá trị của cờ tràn cho phép toán sau: $10110001B + 11000010B$.

Bài làm

$$\begin{array}{r} 10110001B \\ + \\ 11000010B \\ \hline = 101110011B \end{array}$$

Ta thấy: $CF = 1$ (có nhớ từ MSB ra ngoài)

$C_{6,7} = 0$ (Không có nhớ từ bit 6 sang bit 7)

$$\text{Vậy } OF = CF \oplus C_{6,7} = 1 \oplus 0 = 1$$

