

Chương 1.

Tổng quan về kiến trúc máy tính

Câu 1.1.

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

Von Neumann	Non Von Neumann
<ul style="list-style-type: none">-Được xây dựng theo ý tưởng “chương trình được lưu trữ”-Đặc điểm:<ul style="list-style-type: none">+Dữ liệu vào chương trình được chứa trong bộ lọc ghi+Bộ nhớ được đánh địa chỉ cho các ngăn nhớ không phụ thuộc vào nội dung của chúng+Các lệnh của máy tính được thực hiện một cách tuần tự-Gồm 5 thành phần chính:<ul style="list-style-type: none">+Bộ xử lý trung tâm(CPU)+Bộ nhớ làm việc(RAM)+Bộ nhớ vĩnh viễn(ROM)+Thiết bị vào(Input)+Thiết bị ra(Output)	<ul style="list-style-type: none">-Thuật ngữ Non Von Neumann thường được dành riêng cho các máy đại diện cho sự ra đi triệt để khỏi mô hình Von Neumann và do đó thường không được áp dụng cho đa bộ xử lý hoặc đa máy tính.-Một máy không Von Neumann có thể không có khái niệm về luồng điều khiển tuần tự(Nghĩa là không có bất kỳ thanh ghi nào tương ứng với 1 chương trình đối lập và không chỉ ra điểm hiện tại đã đạt được khi thực hiện chương trình)-Không có khái niệm về 1 biến(tức là không có các vị trí lưu trữ có tên và sau đó được tham chiếu hoặc thay đổi)

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

Đơn vị xử lý trung tâm
(Central Processing Unit –
CPU)

Bộ nhớ trong (Internal
Memory)

Bus liên kết hệ thống

Hệ thống vào ra (Input/Output
System - I/O System)

Câu 1.3.

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

- Chức năng: vận chuyển địa chỉ để xác định ngăn nhớ hay cổng vào/ra

- Đặc điểm:

+Độ rộng bus địa chỉ: cho biết số lượng ngăn nhớ tối đa được đánh địa chỉ.

*, N bit: $A_{N-1}, A_{N-2}, \dots, A_2, A_1, A_0 \rightarrow$ có thể đánh địa chỉ tối đa cho 2^N ngăn nhớ (không gian địa chỉ bộ nhớ)

Câu 1.4.

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

- Bus gồm 24 đường dây dẫn, CPU có khả năng quản lý không gian nhớ là $2^{24} = 16\text{MB}$

Câu 1.5.

Trình bày chức năng, đặc điểm của bus dữ liệu.

- Chức năng:

+ vận chuyển lệnh từ bộ nhớ đến CPU

+ vận chuyển dữ liệu giữa CPU, module nhớ, module vào/ra với nhau

- Đặc điểm:

+Độ rộng bus dữ liệu: Xác định số bit dữ liệu có thể được trao đổi đồng thời.

* M bit: $D_{M-1}, D_{M-2}, \dots, D_2, D_1, D_0$

* M thường là 8, 16, 32, 64, 128 bit.

Câu 1.6.

Cho biết ý nghĩa khi nói Bus dữ liệu có độ rộng 32 bit.

-Bus gồm 32 đường dây, CPU có khả năng xử lý toán hạng 32 bit trong 1 chu kỳ lệnh

Câu 1.7.

Trình bày chức năng, đặc điểm của bus điều khiển.

Chức năng: vận chuyển các tín hiệu điều khiển

- Đặc điểm:

+ Các loại tín hiệu điều khiển bao gồm:

* Các tín hiệu điều khiển đọc/ghi

* Các tín hiệu điều khiển ngắt

* Các tín hiệu điều khiển bus

Câu 1.8.

Trình bày hiểu biết của em về tín hiệu điều khiển đọc/ghi bộ nhớ.

- Các tín hiệu (phát ra từ CPU) điều khiển đọc-ghi bộ nhớ :

+ Memory Read (MEMR): điều khiển đọc dữ liệu từ một ngăn nhớ có địa chỉ xác định lên bus dữ liệu.

+ Memory Write (MEMW): điều khiển ghi dữ liệu có sẵn trên bus dữ liệu đến một ngăn nhớ có địa chỉ xác định.

+ I/O Read (IOR): điều khiển đọc dữ liệu từ một cổng vào-ra có địa chỉ xác định lên bus dữ liệu.

+ I/O Write (IOW): điều khiển ghi dữ liệu có sẵn trên bus dữ liệu ra một cổng có địa chỉ xác định.

Câu 1.9.

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

- Tín hiệu điều khiển đọc/ghi thiết bị vào/ra.

+ Bus Request (BRQ) hay là Hold: Tín hiệu từ module điều khiển vào/ra gửi đến yêu cầu CPU chuyển nhượng quyền sử dụng bus.

- + Bus Grant (BGT) hay là Hold Acknowledge (HLDA): Tín hiệu phát ra từ CPU chấp nhận chuyển nhượng quyền sử dụng bus.
- + Lock/ Unlock: Tín hiệu cấm/cho phép xin chuyển nhượng bus

Câu 1.10.

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

- 3 tín hiệu điều khiển ngắt cứng:

- + Interrupt Request (INTR): Tín hiệu từ bộ điều khiển vào-ra gửi đến yêu cầu ngắt CPU để trao đổi vào-ra. Tín hiệu INTR có thể bị che.
- + Interrupt Acknowledge (INTA): Tín hiệu phát ra từ CPU báo cho bộ điều khiển vào-ra biết CPU chấp nhận ngắt để trao đổi vào-ra.
- + Non Maskable Interrupt (NMI): tín hiệu ngắt không che được gửi đến ngắt CPU.

Chương 2.

Kiến trúc tập lệnh

1.1. Tóm tắt lý thuyết cơ bản.

1.1.1. Thông tin - biểu diễn và xử lý thông tin

1.1.1.1. Thông tin

Thông tin là sự hiểu biết, nhận thức của con người về thế giới khách quan, hay nói cách khác, thông tin là một đại lượng phi vật chất, mà con người chỉ có thể cảm nhận thông qua thế giới hiện thực khách quan. Muốn lưu trữ và truyền tải thông tin thì chúng ta phải vật chất hóa thông tin thông qua một đại lượng vật lý nào đó.

1.1.1.2. Dữ liệu

Dữ liệu chính là thông tin đã được vật chất hóa thông qua một đại lượng vật lý nào đó, hay nói cách khác dữ liệu là một đại lượng mang tin. Muốn có thông tin thì chúng ta phải tập hợp và xử lý dữ liệu.

1.1.1.3. Biểu diễn thông tin và xử lý dữ liệu trong máy tính

Trong máy tính, thông tin được vật chất hóa thông qua tín hiệu điện hay nói cách khác, dữ liệu trong máy tính được biểu diễn, lưu trữ và truyền tải thông qua tín hiệu điện.

Trong máy tính có rất nhiều bóng đèn, mỗi bóng đèn ở một trong 2 trạng thái là sáng hay tắt.

+ Trạng thái tắt (không có điện hay mức điện áp thấp) □ cho ta tín hiệu 0.

+ Trạng thái sáng (có điện hay mức điện áp cao) □ cho ta tín hiệu 1.

Tập các tín hiệu 0/1 cho ta một số hệ cơ số 2 (số nhị phân - binary) biểu diễn một giá trị dữ liệu nào đó. Ví dụ, ta có 8 bóng đèn đánh số từ 7 - 0, mỗi bóng ở một trạng thái tương ứng như sau: tắt, sáng, tắt, tắt, tắt, tắt, tắt, sáng; tức ta có tập các tín hiệu 0/1 tương ứng là 0,1,0,0,0,0,0,1 và biểu diễn một số nhị phân là 01000001b; nếu dữ liệu lưu trữ là số thì tập tín hiệu này biểu diễn số 65, nếu dữ liệu lưu trữ là ký tự thì tập các tín hiệu này biểu diễn chữ "A".

Quá trình xử lý dữ liệu trong máy tính có thể tóm tắt như sau:

- Nhận dữ liệu đầu vào đã được số hóa.
- Xử lý dữ liệu thông qua dãy các lệnh.
- Đưa ra kết quả xử lý và lưu trữ dữ liệu đầu ra.

1.1.1.4. Đơn vị thông tin

Mỗi một bit cho ta biết được trạng thái của một tín hiệu điện trên một đường dây tại một thời điểm: điện áp ở mức cao (có điện) là 1, điện áp ở mức thấp (không có điện) 0.

Trong máy tính để biểu diễn một giá trị số, chúng ta dùng *hệ cơ số 2* hoặc nói ngắn gọn hơn là *hệ hai* (Binary number system, viết tắt là hệ B). Trong đó mỗi chữ số chỉ tồn tại 2 giá trị là 0 hoặc 1 (ứng với 2 trạng thái: không có điện và có điện). Hay mỗi chữ số trong số nhị phân ta gọi là 1 bit.

Bit là đơn vị cơ sở để xác định dung lượng của bộ nhớ, bộ nhớ được tổ chức theo byte.

1 byte = 8 bits.

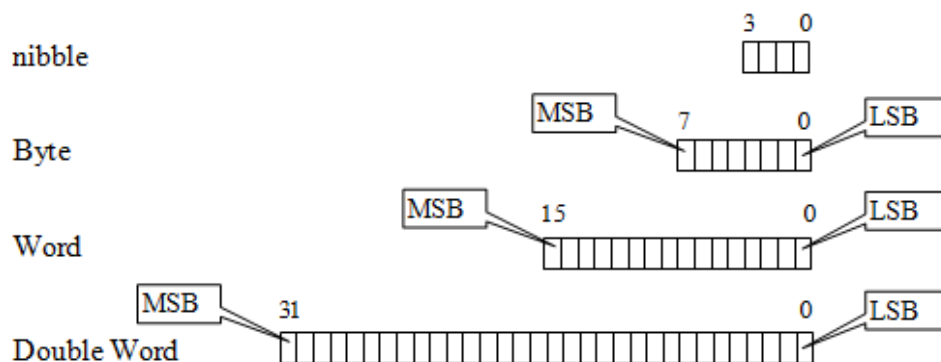
1 KB = 2^{10} bytes. (KB: Kilobyte)

1 MB = 2^{10} KB = 2^{20} bytes. (MB: Megabyte)

1 GB = 2^{10} MB = 2^{20} KB = 2^{30} bytes. (GB: Gigabyte)

1 TB = 2^{10} GB = 2^{20} MB = 2^{30} KB = 2^{40} bytes. (TB: Terabyte)

1.1.1.5. Một số đơn vị đo độ dài của số hệ 2 dẫn xuất từ bit.



Hình 2.1. Các đơn vị đo độ dài của số hệ 2 dẫn xuất từ bit

1.1.2. Hệ đếm và các hệ cơ số

● Hệ đếm cơ số a (tổng quát)

Một số biểu diễn ở hệ đếm cơ số a (tổng quát) như sau:

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

Trong đó:

- i là trọng số của chữ số tương ứng [$i = -(n-k) \div (k-1)$].
- $m_i = [0 \div (a - 1)]$

● Hệ đếm thập phân (Decimal - hệ cơ số 10): $a = 10$, $m_i = 0 \div 9$.

Biểu diễn: Viết bình thường, ví dụ 1029.375

● Hệ đếm nhị phân (Binary - hệ cơ số 2): $a = 2$, $m_i = 0/1$.

Biểu diễn: cuối số phải có chữ b hoặc B hoặc (2).

Ví dụ: 11000100.11b hoặc 11000100.11B, hoặc 11000100.11(2)

● Hệ đếm thập lục phân (Hex - hệ cơ số 16): $a = 16$, $m_i = 0 \div 15$

Biểu diễn:

- Các chữ số có giá trị từ 0 đến 9, viết bình thường.
- Các chữ số từ 10 đến 15 biểu diễn bằng các ký tự tương ứng như sau:

A	10
B	11
C	12
D	13
E	14
F	15

- Chữ số ở đầu không được phép là ký tự (bắt buộc phải là chữ số)
- Cuối số phải có chữ h hoặc H hoặc (16)

Ví dụ biểu diễn đúng:

0A70E.36FDh hoặc 0A70E.36FDH hoặc
0A70E.36FD(16).

Ví dụ biểu diễn sai:

A70E.36FDh hoặc A70E.36FDH hoặc
A70E.36FD(16).

Ví dụ: 11000100.11b hoặc 11000100.11B, hoặc
11000100.11(2)

1.2. Câu hỏi và bài tập chương 2

Câu 2.1.

Đổi số 1097.75 về hệ 2

Gợi ý:

Ta cần thực hiện 3 bước

Bước 1. Đổi phần nguyên về số hệ 2

Để **Chuyển đổi số nguyên giữa các hệ cơ số bất kỳ** được nhanh chóng, tránh thực hiện quá nhiều phép toán, ta thực hiện theo thứ tự như chiều mũi tên:

Số hệ 10 \longleftrightarrow số hệ 16 \longleftrightarrow số hệ 2 \longleftrightarrow số hệ 8 (hệ bát phân)

Cụ thể:

- Nếu từ số hệ 10, cần đổi về hệ 2, trước hết ta đổi số hệ 10 về hệ 16 (chia số hệ 10 cho 16), rồi đổi từng chữ số hệ 16 về hệ 2 (giảm được số phép chia và đỡ nhầm lẫn).
- Nếu từ số hệ 2 cần đổi về hệ 10, trước hết ta đổi số hệ 2 về hệ 16 (nhóm đủ 4 chữ số hệ 2 tính từ dấu chấm ngăn cách giữa phần nguyên và phần lẻ nhị phân, phần nguyên nhóm sang trái, phần lẻ nhị phân nhóm sang phải). Cuối cùng, đổi số hệ 16 về số hệ 10 theo công thức (2.1). (Giảm số phép nhân)

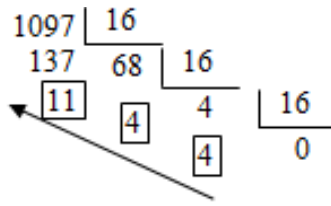
Bước 2. Đổi phần lẻ thập phân về phần lẻ nhị phân (sử dụng thuật toán nhân 2)

Bước 3. Ghép kết quả 2 phần (phần nguyên nhị phân và phần lẻ nhị phân)

Bài làm:

Đổi số 1097 về hệ 2.

- Đổi số 1097 về hệ 16:



Lấy kết quả theo chiều mũi tên. Ta được:

$$1097 = 44BH$$

- Đổi số 44BH về hệ 2 ta được: 010001001011B.
- Vậy $1097 = 10001001011B$.

Đổi 0.75 về hệ 2.

$$0.75 \times 2 = 1.5$$

$$0.5 \times 2 = 1.0$$

- Vậy $0.75 = 0.11B$ (kết quả: lấy phần nguyên của kết quả nhân từ theo thứ tự từ trên xuống dưới)

Ghép 2 phần, ta có:

$$1097.75 = 10001001011.11B$$

Câu 2.2.

Đổi số 12035.125 về hệ 2

- Đổi 12035 về hệ 2

$$12035:2=6017 \text{ dư } 1$$

$$6017:2= 3008 \text{ dư } 1$$

$$3008:2= 1504 \text{ dư } 0$$

$$1504:2= 752 \text{ dư } 0$$

$$752:2= 376 \text{ dư } 0$$

$$376:2= 188 \text{ dư } 0$$

$$188:2= 94 \text{ dư } 0$$

$$94:2= 47 \text{ dư } 0$$

$$47:2= 23 \text{ dư } 1$$

$$23:2= 11 \text{ dư } 1$$

$$11:2= 5 \text{ dư } 1$$

$$5:2= 2 \text{ dư } 1$$

$$2:2= 1 \text{ dư } 0$$

$$1:2= 0 \text{ dư } 1$$

$$\text{Vậy } 12035 = 10111100000011B$$

- Đổi 0.125 về hệ 2

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

$$\text{Vậy } 0.125 = 0.001\text{B}$$

$$\text{Vậy } 12035.25 = 10111100000011.001\text{B}$$

Câu 2.3.

Đổi số 7899.8 về hệ 2

- Đổi 7899 về hệ 2

$$7899:2 = 3949 \text{ dư } 1$$

$$3949:2 = 1974 \text{ dư } 1$$

$$1974:2 = 987 \text{ dư } 0$$

$$987:2 = 493 \text{ dư } 1$$

$$493:2 = 246 \text{ dư } 1$$

$$246:2 = 123 \text{ dư } 0$$

$$123:2 = 61 \text{ dư } 1$$

$$61:2 = 30 \text{ dư } 1$$

$$30:2 = 15 \text{ dư } 0$$

$$15:2 = 7 \text{ dư } 1$$

$$7:2 = 3 \text{ dư } 1$$

$$3:2 = 1 \text{ dư } 1$$

$$1:2 = 0 \text{ dư } 1$$

$$\text{Vậy } 7899 = 1111011011011\text{B}$$

- Đổi 0.8 về hệ 2

$$0.8 \times 2 = 1.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

$$\text{Vậy } 0.8 = 0.1100110011001100\dots\text{B}$$

$$\text{Vậy } 7899.8 = 1111011011011.110011001100\dots\text{B}$$

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:

$$10110010.11B = \underline{1011}0010.\underline{11}00B = 0B2.BH$$

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

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

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

$$\begin{aligned} 10110010.11B &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 \\ &\quad + 2^{-1} + 2^{-2} \\ &= 128 + 32 + 16 + 2 + 0.5 + 0.25 = 178.75 \end{aligned}$$

Câu 2.5.**Đổi số 1111011110110010.110001B về hệ 10.****(Thực hiện theo 2 cách như câu 2.4).**

Cách 1:

- Đổi 1111011110110010.110001B về hệ 16, ta có:

$$\text{Nhóm 1: } 1111(2) = 15(10) = F(16)$$

$$\text{Nhóm 2: } 0111(2) = 7(10) = 7(16)$$

$$\text{Nhóm 3: } 1011(2) = 11(10) = B(16)$$

$$\text{Nhóm 4: } 0010(2) = 2(10) = 2(16)$$

$$\text{Nhóm 5: } 1100(2) = 12(10) = C(16)$$

$$\text{Nhóm 6: } 0100(2) = 4(10) = 4(16)$$

$$\text{Vậy } 1111011110110010.110001B = F7B2.C4H$$

- Đổi F7B2.C4 về hệ 10, ta có:

$$15 \times 16^3 + 7 \times 16^2 + 11 \times 16^1 + 2 \times 16^0 + 12 \times 16^{-1} + 4 \times$$

$$16^{-2} = 63410.76563$$

Cách 2:

1111011110110010.110001B=

$$1 \cdot 2^{15} + 1 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^{12} + 0 \cdot 2^{11} + 1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} = 63410.76563$$

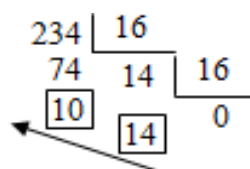
Câu 2.6.

Đổi số nguyên thập phân sau ra số hex 16 bit: 234.

Gợi ý

Số hệ hex 16 bit phải là số có đủ 4 chữ số hệ 16, 234 là số nguyên dương nên đổi về hệ 16 bình thường theo thuật toán chia 16.

Bài làm



Vậy 234 = 00EAH

Câu 2.7.

Đổi số nguyên thập phân sau ra số hex 16 bit: 7899.

Bài làm

$$7899:16 = 493 \text{ dư } 11$$

$$493:16 = 30 \text{ dư } 13$$

$$30:16 = 1 \text{ dư } 14$$

$$1:16 = 0 \text{ dư } 1$$

$$\text{Vậy } 7899 = 1EDBH$$

Câu 2.8.

Đổi số nguyên thập phân sau ra số hex 16 bit: 31634.

Bài làm

$$31634:16 = 1977 \text{ dư } 2$$

$$1977:16 = 123 \text{ dư } 9$$

$$123:16 = 7 \text{ dư } 11$$

$$7:16 = 0 \text{ dư } 7$$

Bài làm

Ta có : + 67= 0000000001000011B

Số bù 1(của +67) = 1111111110111100B

+ 1

Số bù 2 (của+67) = 1111111110111101B

vậy -67= 1111111110111101B

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:

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

Số bù 1 (của +128) = 01111111B

+ 1

Số bù 2 (của +128) = 10000000B

Vậy -128 = 10000000B = +128 (sai)

Nguyên nhân:

1. Cách đặt vấn đề là cho $128 = 10000000B$ (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:

- 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

a. Với AX = 1234H

$$= 0001001000110100B \text{ (MSB = 0 là số dương)}$$

$$= 2^{12} + 2^9 + 2^5 + 2^4 + 2^2$$

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

b. Với BX = 900AH

$$= \underline{1001000000001010}B$$

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

$$B = \underline{1001000000001010}B$$

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

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

$$\text{Số bù 2(B)} = \underline{0110111111110110}B = 6FF6H$$

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

$$= 28662 = |B|$$

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

c. Với CX = FFFAH

$$= \underline{1111111111111010}B$$

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

$$C = \underline{1111111111111010}B$$

$$\text{Số bù 1(của C)} = \underline{0000000000001010}B$$

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

$$\text{Số bù 2(của C)} = \underline{0000000000001100}B = 0006H = 6 = |C|$$

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

d. Với DX = 7FFH

$$= \underline{0111111111111111}B \text{ (MSB = 0 là số dương)}$$

$$= 32767$$

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)

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

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

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

$$\text{Vậy } 0.9 = 0.11110111011101110\dots B$$

(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

$$1022.9 = 1111111110.11110111011101110\dots B$$

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

$$1022.9 = 1111111110.11110111011101110\dots B$$

$$= (-1)^0 \times 1.11111111011100110011001B \times 2^9$$

(Sau dấu chấm nhị phân lấy đủ 23 bit)

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

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

$$10001000B$$

Vậy ta có :

$$S = 0.$$

$$E = 10001000B$$

$$M = 11111111011100110011001B$$

Bước 3. Ghép vào khuôn dạng.

$$1022.9 = 01000100011111111011100110011001B = 447FB999H$$

Câu 2.17.

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

Bài làm

Ta có CT: $R = (-1)^S \cdot 1.M \cdot 2^{(E - \text{Bias})}$

$$\text{Bias} = 2^{\text{số bit của } E/2 - 1}$$

Khuôn dạng: SEM

- Đổi - 1012.8 về hệ 2

- Đổi - 1012 về hệ 2:

Vậy -1012= 1111110100B

- Đổi 0.8 về hệ 2

$$0.8 \times 2 = 1.6$$

$$0.6 \times 2 = 1.2$$

$$0.2 \times 2 = 0.4$$

$$0.4 \times 2 = 0.8$$

Vậy 0.8= 0.110011001100....B

Vậy -1012.8= 1111110100110011001100...B

$$= (-1)^1 \times 1.11111010011001100110011 B \times 2^9$$

E- Bias= 9 suy ra E= 9+ Bias= 9+127=128+8= 136= 10000000B+1000B= 10001000B

Vậy ta có:

$$S= 1$$

$$E= 10001000B$$

$$M= 11111010011001100110011B$$

Vậy -1012.8= 11000100011111010011001100110011B=C47D333H

Câu 2.18.

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

Bài làm

Ta có CT: $R = (-1)^S \cdot 1.M \cdot 2^{(E-Bias)}$

$$Bias = 2^{\text{số bit của } E/2 - 1}$$

Khuôn dạng: SEM

Đổi 912.0625 về hệ 2:

-Đổi 912 về hệ 2:

$$912 = 1110010000B$$

-Đổi 0.0625 về hệ 2:

$$0.0625 \times 2 = 0.125$$

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

$$\text{Vậy } 0.0625 = 0.0001\text{B}$$

$$\text{Vậy } 912.0625 = 1110010000.0001\text{B}$$

$$= (-1)^0 \times 1.110010000000100000000000 \times 2^9$$

$$\text{E-Bias} = 9 \Rightarrow \text{E} = 9 + \text{Bias} = 9 + 127 = 128 + 8 = 10000000\text{B} + 1000\text{B} = 10001000\text{B}$$

Vậy ta có:

$$\text{S} = 0$$

$$\text{E} = 10001000\text{B}$$

$$\text{M} = 110010000000100000000000\text{B}$$

$$\text{Vậy } 912.0625 = 01000100011001000000010000000000\text{B} = 4464040\text{H}$$

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

$$447\text{FB999H} = 01000100011111111011100110011001\text{B}.$$

Theo khuôn dạng lưu trữ, ta tách được:

$$\text{S} = 0.$$

$$\text{E} = 10001000\text{B} = 2^8 + 2^3 = 128 + 8 = 136$$

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

$$\text{M} = 11111111011100110011001\text{B}$$

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.

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

$$(\text{phương pháp nhanh: } 11111111B = 2^{10} - 1)$$

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

$$\begin{aligned}0.11100110011001B &= 0.1110011001100100B \\ &= 0.E664H \\ &= 14 * 16^{-1} + 6 * 16^{-2} + 6 * 16^{-3} + 4 * 16^{-4} \\ &= 0.875 + 0.0234375 + 0.001465 + \\ &0.0000610 \\ &= 0.8999634\end{aligned}$$

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

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

$$\text{Bias} = 2^{\text{số bit của E}/2 - 1}$$

Khuôn dạng: SEM

$$409CCCCCH = 01000000100111001100110011001100B$$

Theo khuôn dạng lưu trữ, ta tách được:

$$S = 0$$

$$E = 10000001 = 2^7 + 2^0 = 128 + 1 = 129$$

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

$$M = 00111001100110011001100B$$

Ta có :

$$\begin{aligned}R &= (-1)^0 \times 1.00111001100110011001100B \times 2^2 \\ &= 100.111001100110011001100B\end{aligned}$$

Có $100B=4$

$0.111001100110011001100000B = E66660H = 14 \times 16^{(-1)} + 6 \times 16^{(-2)} + 6 \times 16^{(-3)} + 6 \times 16^{(-4)} + 6 \times 16^{(-5)} = 0.8999996185$

Vậy $409CCCCCH = 4.8999996185 \approx 4,9$

Câu 2.21.

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

Bài làm

Ta có CT: $R = (-1)^S \times 1.M \times 2^{(E-Bias)}$

$Bias = 2^{\text{số bit của } E/2 - 1}$

Khuôn dạng: SEM

$BF400000H = 10111111010000000000000000000000B$

Theo khuôn dạng lưu trữ, ta tách được:

$S=1$

$E = 01111110B = 2^7 - 2^1 = 128 - 2 = 126$

$E-Bias = 126 - 127 = -1$

$M = 10000000000000000000000000B$

Ta có:

$R = (-1)^1 \times 1.10000000000000000000000000B \times 2^{(-1)}$

$= 0.11000000000000000000000000B$

Có:

$0000B = 0$

$0.11000000000000000000000000B = 0.C00000H$

Vậy $BF400000H = 12 \times 16^{(-1)} = 0.75$

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

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 32

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 36

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 quần chiếu 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

Giá trị nhớ: 4

1979	
1187	
+ 677	
129	
5679	
458	
9	

Tổng cột

49

10

9

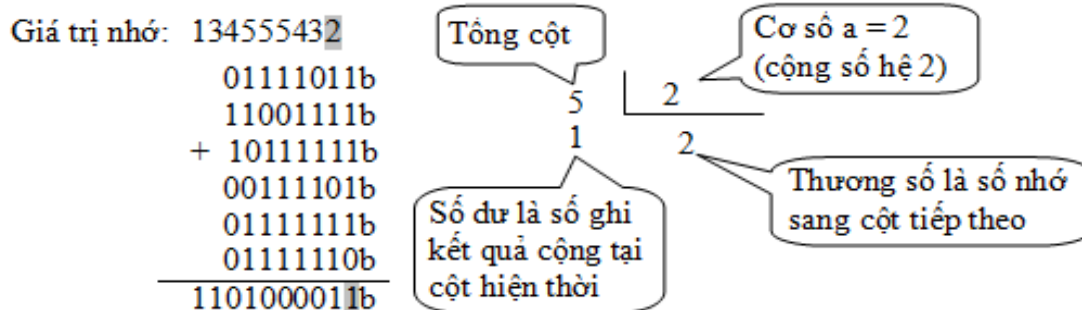
4

Cơ số $a = 10$
(cộng số hệ 10)

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

Thương số là số nhớ sang cột tiếp theo

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:



Câu 2.28.

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

Bài làm

Giá trị nhớ: 133334322

$$\begin{array}{r}
 11101101B \\
 11011110B \\
 + 10111111B \\
 10101101B \\
 01111101B \\
 = 1110110100B
 \end{array}$$

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

$$\begin{array}{r}
 11101111B \\
 \times \\
 11111101B \\
 \hline
 11101111B \\
 00000000B \\
 11101111B \\
 11101111B \\
 + 11101111B \\
 11101111B \\
 11101111B \\
 11101111B \\
 \hline
 = \underline{1110110000110011B}
 \end{array}$$

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): Chứa 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

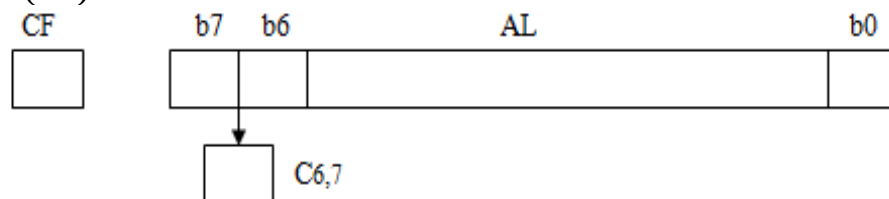
- Cờ Zero (cờ rỗng) : được thiết lập lên 1 khi kết quả của phép toán bằng 0
- Cờ Sign (cờ dấu) : được thiết lập lên 1 khi kết quả của phép toán nhỏ hơn 0
- Cờ Carry (cờ nhớ) : được thiết lập lên 1 nếu phép toán có nhớ ra ngoài bit cao nhất -> cờ báo tràn với số không dấu
- Cờ Overflow (cờ tràn) : được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

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)

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)

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

Chương 3. CPU, Đường truyền và hệ thống vào/ra

Câu 3.1.

Cho biết ý nghĩa khi nói Bus địa chỉ có độ rộng 32 bit.

Bài làm

- Bus gồm 32 đường dây, CPU có khả năng xử lý toán hạng 32 bit trong 1 chu kỳ lệnh

Câu 3.2.

Cho biết ý nghĩa khi nói Bus dữ liệu có độ rộng 64 bit.

Bài làm

- Bus gồm 64 đường dây dẫn, CPU có khả năng quản lý không gian nhớ là 2^{64} byte

Câu 3.3.

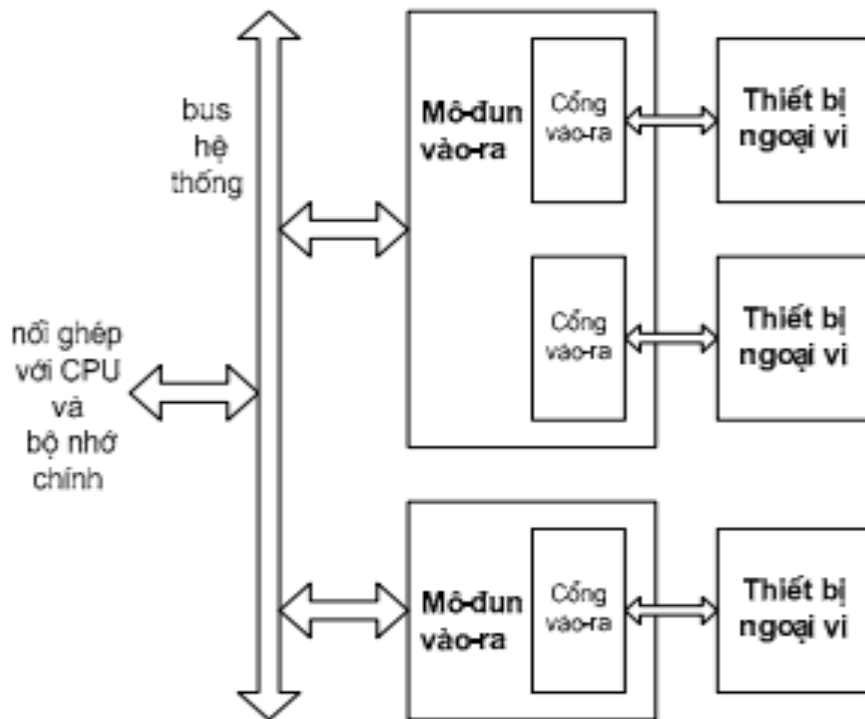
Vẽ sơ đồ khối chung của hệ thống xử lý vào/ra trong máy tính và trình bày tóm tắt chức năng của từng thành phần trong sơ đồ.

Gợi ý

Hệ thống xử lý vào/ra dữ liệu gồm 4 thành phần: CPU, bộ nhớ chính, hệ thống vào/ra dữ liệu và bus

Bài làm

Sơ đồ khối chung của hệ thống xử lý vào/ra trong máy tính:



Chức năng:

- CPU: Điều khiển mọi hoạt động bên trong máy tính và thực hiện các phép tính
- Bộ nhớ chính: Lưu trữ dữ liệu và chương trình trong máy tính
- Hệ thống vào/ra dữ liệu(thiết bị ngoại vi): giao tiếp giữa máy tính với thế giới bên ngoài (con người)
- Bus hệ thống: Liên kết & truyền tín hiệu giữa các thành phần trong máy tính

Câu 3.4.

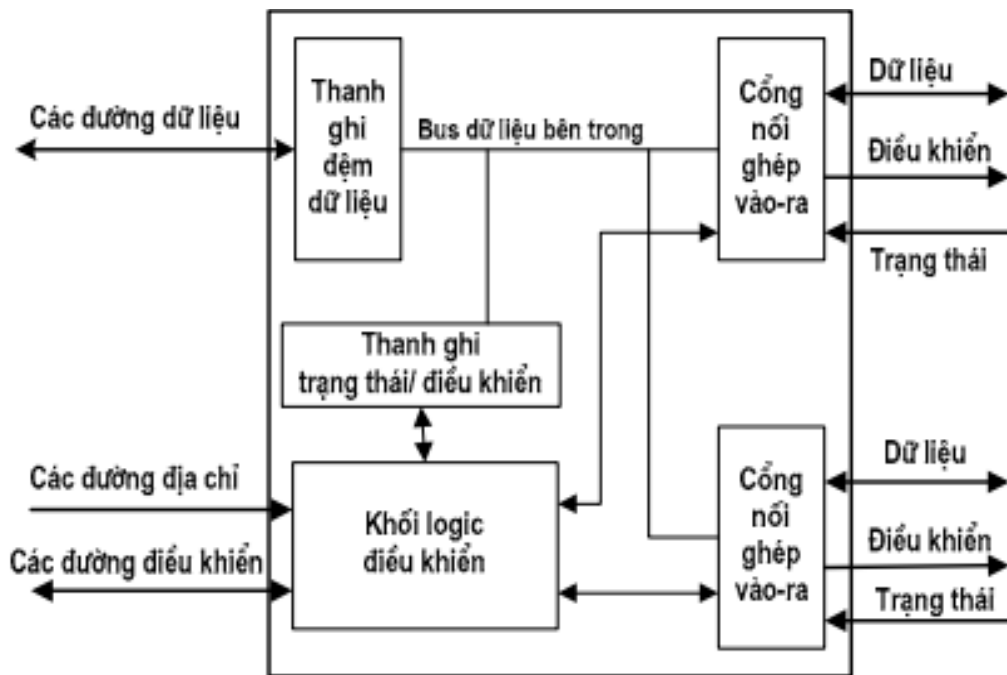
Vẽ sơ đồ khối chung của hệ thống vào/ra dữ liệu và trình bày tóm tắt chức năng của từng thành phần trong sơ đồ.

Gợi ý

Hệ thống vào/ra dữ liệu gồm 2 thành phần: Module vào/ra (khối ghép nối/ các điều khiển) và thiết bị vào/ra

Bài làm

Sơ đồ khối chung của hệ thống vào/ra dữ liệu:



Chức năng:

- Module vào/ra: nối ghép các thiết bị ngoại vi với máy tính
- Mỗi module vào/ra có một hoặc một vài cổng vào/ra (I/O Port).
- Mỗi cổng vào/ra được đánh một địa chỉ xác định.
- Các thiết bị ngoại vi được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào/ra.

Câu 3.5.

Trình bày cấu trúc chung của modul vào/ra dữ liệu.

Bài làm

Cấu trúc chung của modul vào/ra dữ liệu gồm:

- Thanh ghi đệm dữ liệu: đệm dữ liệu trong quá trình trao đổi
- Các cổng vào/ra (I/O Port): kết nối với các thiết bị ngoại vi, mỗi cổng có một địa chỉ xác định.
- Thanh ghi trạng thái/điều khiển: lưu trữ thông tin trạng thái/điều khiển cho các cổng vào/ra
- Khối logic điều khiển: điều khiển mô-đun ra/vào

Câu 3.6.

Trình bày 2 phương pháp vào ra dữ liệu do CPU chủ động.

Gợi ý

Đó là phương pháp vào/ra theo định trình và vào/ra kiểu thăm dò.

Bài làm

Phương pháp vào/ra theo định trình:

- Thực hiện tức thời vào ra bằng cách sử dụng câu lệnh vào/ra (IN/OUT).
- CPU không cần kiểm tra trạng thái sẵn sàng của thiết bị cũng như mô-đun vào/ra.
- Nhược điểm : Độ tin cậy trong truyền, nhận dữ liệu không cao, dữ liệu truyền nhận dễ bị chồng lên gây mất dữ liệu nếu xử lý không kịp.
- Phương pháp này thích hợp với quá trình vào/ra có chu kỳ cố định và có thể xác định trước

Phương pháp vào/ra kiểu thăm dò:

- CPU yêu cầu thao tác vào/ra
- Mô-đun vào/ra thực hiện thao tác
- Mô-đun vào/ra thiết lập các bit trạng thái
- CPU kiểm tra các bit trạng thái:
 - + Nếu chưa sẵn sàng thì quay lại kiểm tra
 - + Nếu sẵn sàng thì chuyển sang trao đổi dữ liệu với mô-đun vào/ra

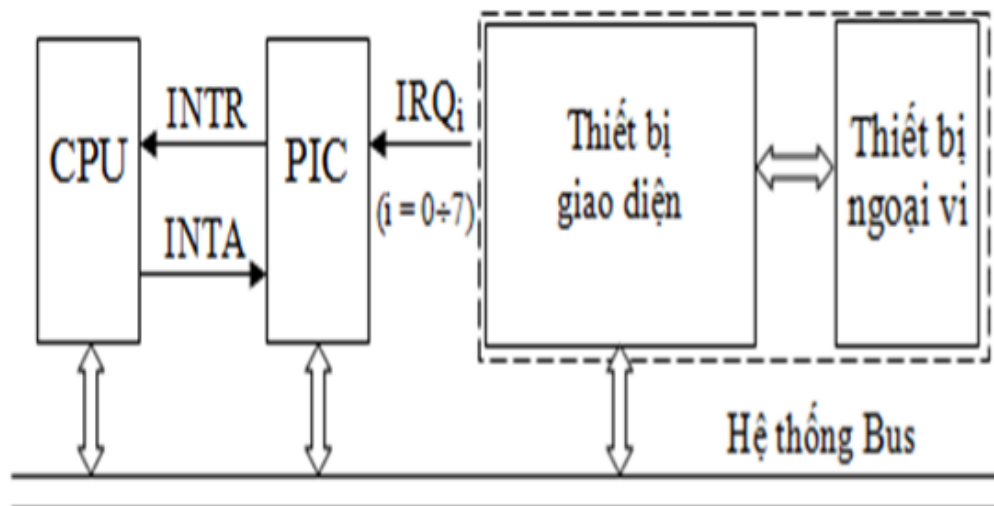
Câu 3.7.

Trình bày cấu trúc của hệ thống vào/ra theo ngắt cứng (vẽ hình).

Gợi ý

Hệ thống vào/ra theo ngắt cứng gồm 4 thành phần: CPU, PIC (Priority Interrupt Controller), hệ thống vào/ra dữ liệu và bus.

Bài làm



- Hệ thống vào/ra theo ngắt cứng gồm 4 thành phần: CPU, PIC (Priority Interrupt Controller), hệ thống vào/ra dữ liệu và bus.
- PIC nhận được yêu cầu ngắt IRQ_i ($i = 0 \div 7$), xử lý ưu tiên ngắt và cung cấp số liệu ngắt có ưu tiên cao nhất cho CPU qua BUS dữ liệu. CPU căn cứ vào số hiệu này thực hiện quá trình vào ra dữ liệu với chương trình được chọn.

Câu 3.8.

Trình bày quá trình vào/ra dữ liệu theo phương pháp vào/ra dữ liệu theo ngắt cứng.

Bài làm

Quá trình vào/ra dữ liệu theo phương pháp ngắt cứng:

- Các thiết bị vào/ra có yêu cầu phục vụ, phát ra tín hiệu IRQ_i ($i = 0 \div 7$) tới PIC. Mỗi thiết bị vào/ra đã được ấn định sẵn một số hiệu ngắt nhất định.
- Thiết bị PIC lựa chọn số hiệu ngắt có mức ưu tiên cao nhất, phát tín hiệu INTR (Interrupt Request) tới CPU yêu cầu CPU phục vụ.
- Nếu CPU chấp nhận ngắt, CPU hoàn thành nốt lệnh đang thực hiện, tiến hành lưu trạng thái của tiến trình đang thực hiện và trạng thái hiện thời của CPU.
- CPU phát ra tín hiệu INTA (Interrupt Acknowledge) trả lời PIC, báo sẵn sàng phục vụ yêu cầu ngắt.
- PIC phát ra số hiệu ngắt được chọn tới CPU qua BUS dữ liệu.

- Dựa vào số liệu ngắt này, CPU xác định được địa chỉ của chương trình con phục vụ ngắt, kích hoạt và thực hiện chương trình con phục vụ ngắt để thực hiện vào/ra dữ liệu với thiết bị được chọn.
- Khi chương trình con phục vụ ngắt kết thúc, CPU lấy trạng thái cũ của CPU và của tiến trình vừa bị ngắt để tiếp tục thi hành.

Câu 3.9.

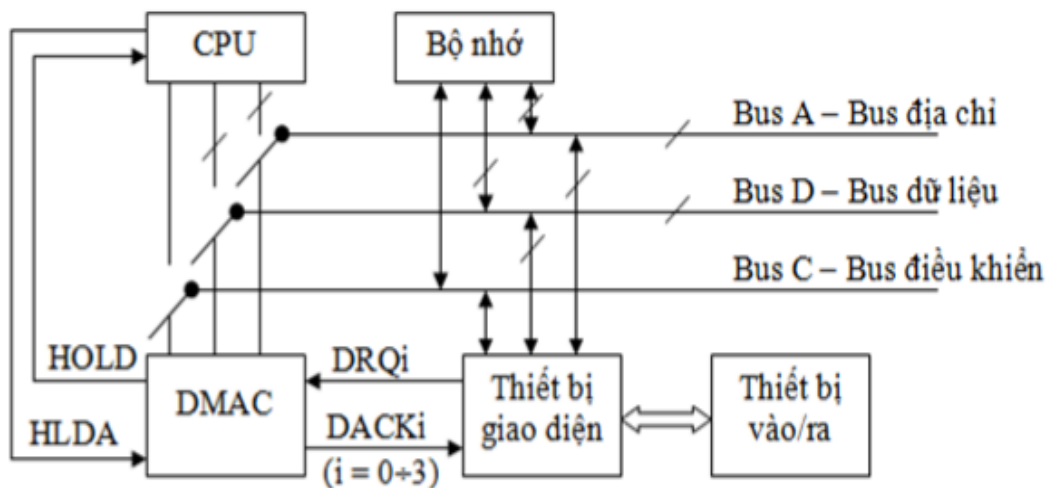
Trình bày cấu trúc của hệ thống vào/ra dữ liệu theo kiểu DMA.

Gợi ý

Hệ thống vào/ra dữ liệu theo kiểu DMA gồm 5 thành phần: CPU, DMAC (Direct Memory Access Controller), bộ nhớ, hệ thống vào/ra và bus

Bài làm

Cấu trúc của hệ thống vào/ra dữ liệu theo kiểu DAM:



Câu 3.10.

Trình bày khái niệm quá trình DMA và quá trình vào/ra dữ liệu kiểu DMA (quá trình DMA).

Bài làm

Khái niệm quá trình DMA:

- DMA- Direct Memry Access(Truy cập bộ nhớ trực tiếp)
- Quá trình DMA là quá trình vào/ra dữ liệu giữa bộ nhớ và thuật bị ngoại vi mà không thông qua CPU.

Quá trình DMA:

- Thiết bị có nhu cầu trao đổi DMA sẽ gửi yêu cầu trao đổi đến chân DRQ_i cho DMAC, DMAC chọn thiết bị có mức ưu tiên cao nhất.

- DMAC phát tín hiệu BRQ (Bus Request)/HOLD = 1 cho CPU, yêu cầu CPU chuyển nhượng Bus.
- Nếu CPU đồng ý thì thực hiện nốt chu kỳ bus, CPU sẽ tiến hành khởi tạo cho DMAC
- DMAC điều khiển bus. Phát tín hiệu DACK_i để báo cho thiết bị biết là chấp nhận trao đổi DMA
- DMAC điều khiển quá trình trao đổi dữ liệu giữa thiết bị vào/ra và bộ nhớ
- Sau khi truyền được một dữ liệu thì:
 - + Nội dung thanh ghi địa chỉ tăng
 - + Nội dung bộ đếm dữ liệu giảm
- Khi bộ đếm dữ liệu = 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA

Câu 3.11.

Trình bày các kiểu trao đổi dữ liệu theo phương pháp vào/ra dữ liệu kiểu DMA.

Gợi ý

Có 3 kiểu trao đổi dữ liệu theo phương pháp vào/ra dữ liệu kiểu DMA: Treo CPU một khoảng thời gian để trao đổi cả mảng dữ liệu, Treo CPU để trao đổi từng byte, Tận dụng thời gian CPU không dùng Bus để trao đổi dữ liệu (dùng lên bus).

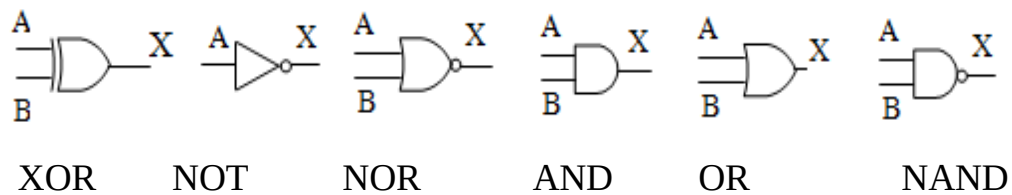
Bài làm

Có 3 kiểu trao đổi dữ liệu DMA:

- Treo CPU một khoảng thời gian để trao đổi cả mảng dữ liệu (Block-transfer): DMA sử dụng bus để truyền xong cả khối dữ liệu.
- Treo CPU để trao đổi từng byte (Cycle Stealing DMA): DMAC cưỡng bức CPU treo tạm thời từng chu kỳ bus, DMAC chiếm bus thực hiện truyền một từ dữ liệu
- Tận dụng thời gian CPU không dùng Bus để trao đổi dữ liệu (Transparent DMA): DMAC nhận biết những chu kỳ nào CPU không sử dụng bus để trao đổi một từ dữ liệu

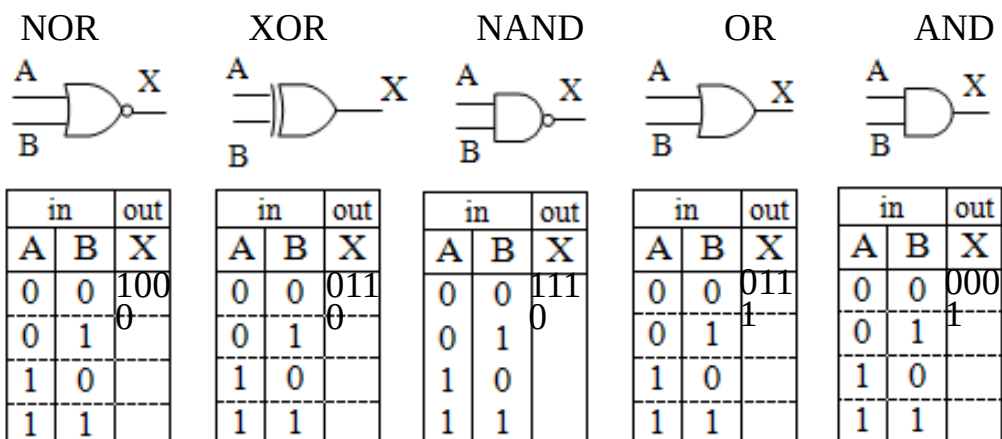
Câu 3.12.

Nhận biết các mạch cổng sau (ghi rõ tên mạch cổng ở bên dưới hình)



Câu 3.13.

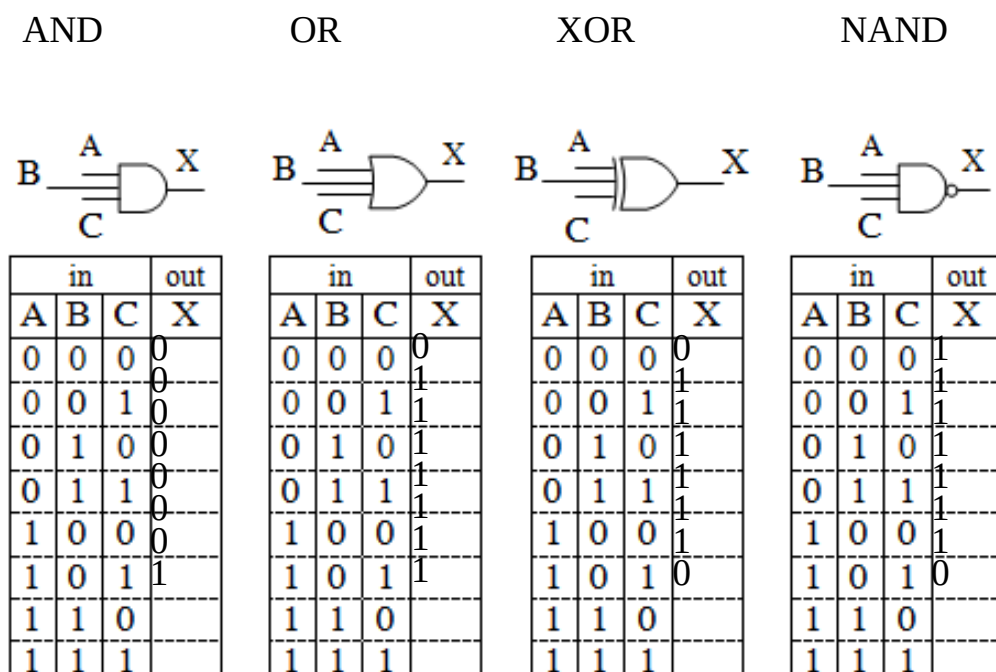
Cho các mạch cổng 2 đầu vào, ghi tên cổng, hoàn thiện các bảng chân lý tương ứng DẠCK_i để Tên cổng:



Câu 3.14.

Cho các mạch cổng 3 đầu vào, ghi tên cổng, hoàn thiện các bảng chân lý tương ứng

Tên cổng:



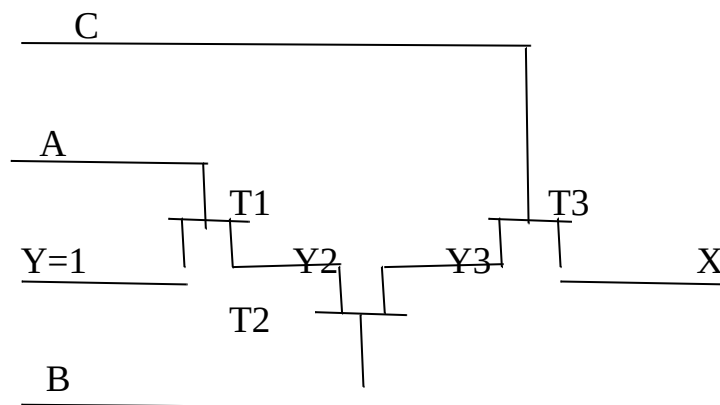
Câu 3.15.

Vẽ sơ đồ mạch điện của cổng AND 3 đầu vào từ các transistor

Gợi ý

Xem sơ đồ mạch của cổng AND 2 đầu vào từ các transistor (hình 3.11. tài liệu học tập - giáo trình kiến trúc máy tính)

Bài làm

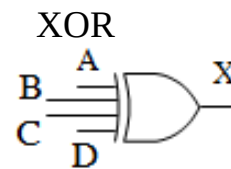
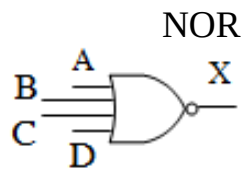
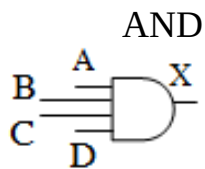


Sơ đồ mạch điện của cổng AND 3 đầu vào từ các transistor

Câu 3.16.

Cho các mạch cổng 4 đầu vào sau, ghi tên cổng, hoàn thiện các bảng chân lý tương ứng

Tên cổng:



in				out
A	B	C	D	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

in				out
A	B	C	D	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

in				out
A	B	C	D	X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

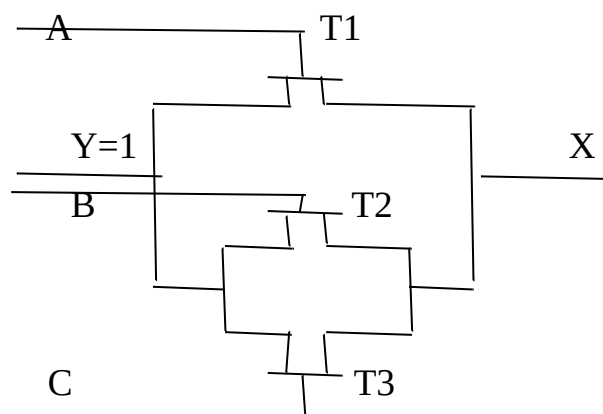
Câu 3.17.

Vẽ sơ đồ mạch điện của cổng OR 3 đầu vào từ các transistor

Gợi ý

Xem sơ đồ mạch của cổng OR 2 đầu vào từ các transistor (hình 3.11. tài liệu học tập - giáo trình kiến trúc máy tính)

Bài làm



Sơ đồ mạch điện của cổng OR 3 đầu vào từ các transistor

Câu 3.18.

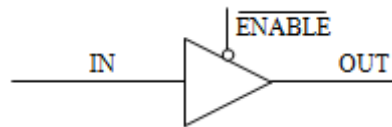
Vẽ mạch điện có chức năng tương ứng mạch NOT

Bài làm



Câu 3.19.

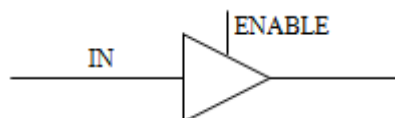
Cho biết tên và nguyên lý làm việc của mạch logic sau:



- Đây là mạch thiết bị 3 trạng thái
- Nguyên lý làm việc :
 - Khi chân có mức logic thấp (0) thì đầu ra (OUT) bằng đầu vào (IN).
 - Khi chân có mức logic cao (1) thì đầu ra và đầu vào cách ly (đầu ra có mức trở kháng cao).

Câu 3.20.

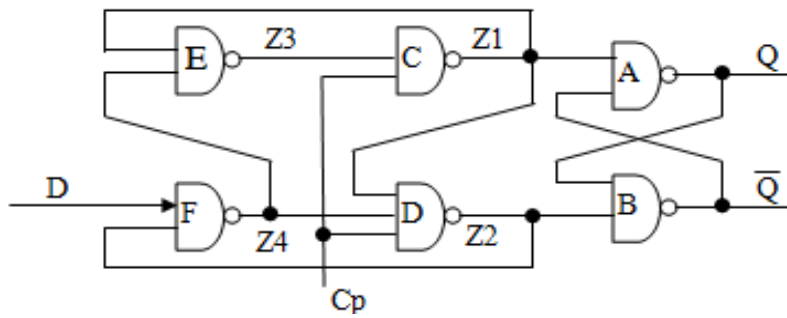
Cho biết tên và nguyên lý làm việc của mạch logic sau:



- Đây là mạch thiết bị 3 trạng thái ENABLE
- Nguyên lý làm việc :
 - Khi chân ENABLE có mức logic cao (1) thì đầu ra (OUT) bằng đầu vào (IN).
 - Khi chân ENABLE có mức logic thấp (0) thì đầu ra và đầu vào cách ly (đầu ra có mức trở kháng cao).

Câu 3.21.

Cho sơ đồ mạch Flip-Flop sau, chứng minh rằng khi $C_p = 0$, trạng thái đầu ra Q không đổi, khi $C_p = 1$ (chuyển từ 0 - 1) thì $Q = D$



Gợi ý

- Khi $C_p = 0$, tính $Z1$, $Z2$ và giả sử ban đầu $Q = 1$, tính , sau đó tính lại Q , tương tự với giả thiết ban đầu $Q = 0$.
- Khi $C_p = 1$ (tức C_p chuyển từ 0 lên 1): dữ liệu đầu vào là $z1 = z2 = 1$ (có khi $C_p = 0$)

Bài làm

+Cho $D=0$ CM $Q=0$

$$Z4 = \text{NAND}(D, Z2) = \text{NAND}(0, 1) = 1$$

$$Z2 = \text{NAND}(C_p, Z1, Z4) = \text{NAND}(1, 1, 1) = 0$$

Ta có: $Z2=0$

$$\square = \text{NAND}(Z2, Q) = \text{NAND}(0, Q) = 1 \square \text{Vây } Q=0 \text{ (1)}$$

+Cho $D=1$ CM $Q=1$

$$Z4 = \text{NAND}(D, Z2) = \text{NAND}(1, 1) = 0$$

$$Z3 = \text{NAND}(Z4, Z1) = \text{NAND}(0, 1) = 1$$

$$Z1 = \text{NAND}(Z3, C_p) = \text{NAND}(1, 1) = 0$$

Ta có: $Z1=0$

$$\square Q = \text{NAND}(Z1,) = \text{NAND}(0,) = 1(2)$$

Từ (1) và (2) \square dpcm

Câu 3.22.

Thiết kế mạch logic để phát hiện lỗi trong mã BCD. Lỗi vào là mã BCD, lỗi ra ở trạng thái 1 khi có lỗi.

Gợi ý

Ta cần nắm một số khái niệm sau:

- Số BCD là số nhị phân 4 bit biểu diễn một số từ thập phân có một chữ số.
- Số BCD không gói là số nhị phân 8 bit biểu diễn một số từ thập phân có một chữ số.

- Số BCD gói là số nhị phân 8 bit biểu diễn một số từ thập phân có hai chữ số. Trong đó số BCD cao (4 bits cao) biểu diễn số hàng chục, số BCD thấp (4 bits thấp) biểu diễn số hàng đơn vị.

Số BCD lỗi là khi giá trị của số nhị phân 4 bit lớn hơn 9 (từ 10 đến 15)

Bài làm

Giả sử ta có số nhị phân 4 bit sau: B3B2B1B0

Số BCD lỗi là khi giá trị của số nhị phân 4 bit lớn hơn 9

$$\square \text{OR}(\text{AND}(\text{B3}, \text{B1}), \text{AND}(\text{B3}, \text{B2})) = \text{B1} \cdot \text{B3} + \text{B2} \cdot \text{B3} = \text{B3}(\text{B1} + \text{B2})$$

$$\begin{array}{r} \text{B3} \\ \hline \text{B2} \\ \hline \text{B1} \\ \hline \end{array}$$

Câu 3.23.

Hãy xây dựng cổng XOR 2 đầu vào từ các cổng NOT, AND, OR.

Gợi ý

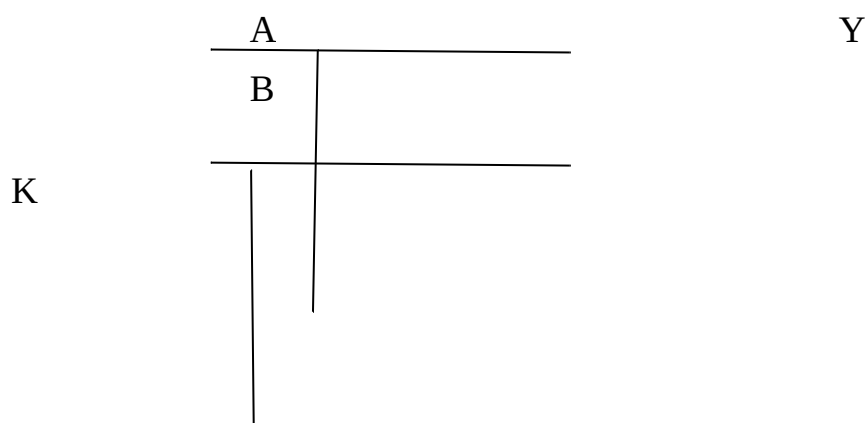
Để xây dựng 1 cổng XOR có 2 đầu vào từ các cổng NOT, AND, OR, ta có bảng chân lý sau.

A	B	$X = A \text{ AND } B$	$Y = A \text{ OR } B$	$Z = \text{NOT } X$	$K = Y \text{ AND } Z = (A \text{ XOR } B)$
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	1	0	0

Từ bảng chân lý này ta vẽ được sơ đồ mạch logic số.

Sinh viên thực hiện vẽ sơ đồ mạch:

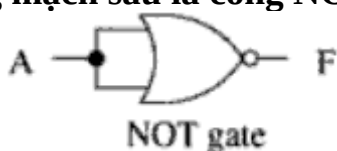
Ta có: $K = \text{AND}(Y, Z)$
 $\text{AND}(\text{OR}(A, B), \text{NOT}(X)) = \text{AND}(\text{OR}(A, B), \text{NOT}(\text{AND}(A, B)))$
 $= (A + B) \cdot ()$



_____ X Z

Câu 3.24.

Chứng minh rằng mạch sau là cổng NOT



Bài làm

C1: Ta có: $F = \text{NOR}(A, A) = \neg(A \vee A) = \neg A$ (dpcm)

C2:

Ta có bảng chân lý của mạch NOT

A	F
0	1
1	0

Ta thấy:

$F = \text{NOR}(A, A)$

A	F
0	1
1	0

□Đầu ra của mạch trùng với bảng chân lý của mạch NOT□ Mạch trên là cổng NOT

Câu 3.25.

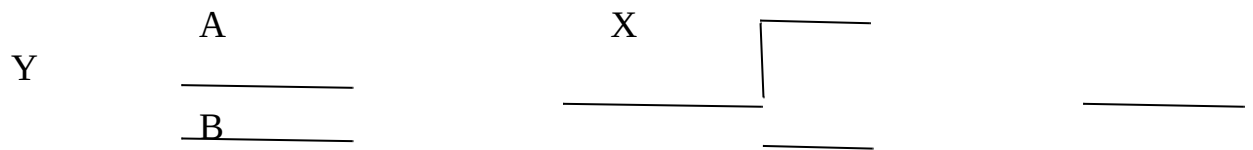
Hãy xây dựng cổng AND 2 đầu vào từ các cổng NAND 2 đầu vào.

Gợi ý

Để xây dựng 1 cổng AND có 2 đầu vào từ các cổng NAND 2 đầu vào, ta có bảng chân lý sau.

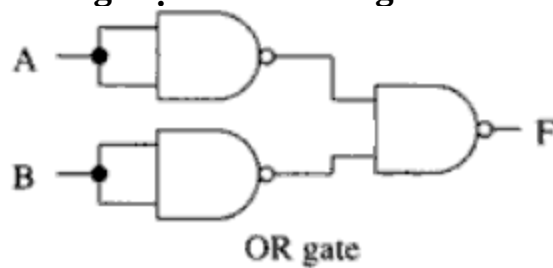
A	B	$X = A \text{ NAND } B$	$Y = X \text{ NAND } X (= A \text{ AND } B)$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Sinh viên thực hiện vẽ sơ đồ mạch:



Câu 3.26.

Chứng minh rằng mạch sau là cổng OR 2 đầu vào



Bài làm

C1: Ta có : $F = \text{NAND}(\text{NAND}(A,A), \text{NAND}(B,B))$

Mà $\text{NAND}(A,A) =$

$\text{NAND}(B,B) =$

□

$F = \text{NAND}(\text{NAND}(A,A), \text{NAND}(B,B)) = \text{NAND}(,) = = A + B (\text{dpcm})$

C2:

Ta có bảng chân lý của mạch OR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Ta thấy:

$F = \text{NAND}(\text{NAND}(A,A), \text{NAND}(B,B))$

Đặt $X = \text{NAND}(A,A)$

$Y = \text{NAND}(B,B)$

□ $F = \text{NAND}(X,Y)$

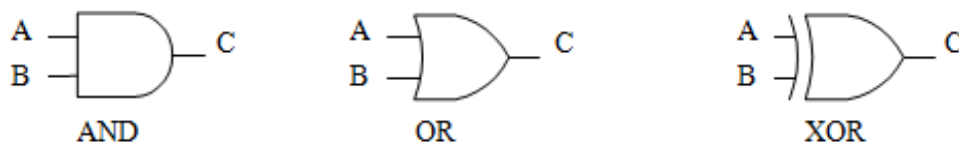
A	B	X	Y	F
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

□ Đầu ra của mạch trùng với bảng chân lý của mạch OR □ Mạch trên là cổng OR

Câu 3.27.

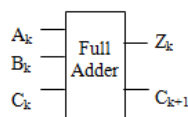
Hãy đọc những mô tả dưới đây về các phép toán logic và bộ cộng đầy đủ, sau đó trả lời các ý từ 1 tới 3.

a. Các kí hiệu mạch logic cho các phép toán logic chính như sau.



Trong đó A, B đầu vào, C đầu ra

b. Sau đây là hình vẽ bộ cộng đầy đủ, thực hiện việc cộng các số nhị phân theo từng chữ số có tính tới việc nhớ. Bảng cho dưới đây là bảng chân lý cho bộ cộng đầy đủ đó.



Trong đó:
 A_k, B_k : đầu vào
 Z_k : đầu ra
 C_k : nhớ từ số thứ $k-1$ sang
 C_{k+1} : nhớ tới số thứ $k+1$.

Bộ cộng đầy đủ (full adder)

Bảng chân lý của bộ cộng đầy đủ:

Đầu vào			Đầu ra	
C_k	A_k	B_k	Z_k	C_{k+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1		
1	1	0	0	1
1	1	1	1	1

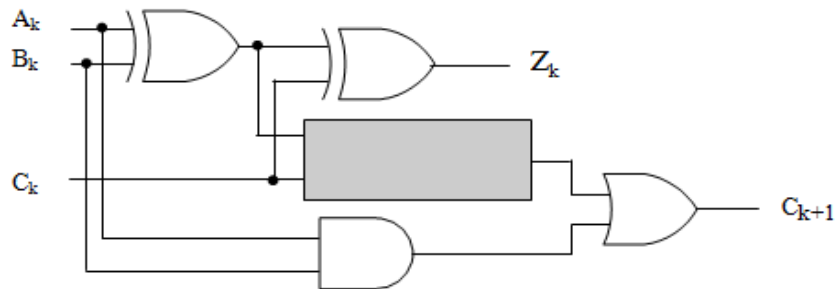
Ý 1 : Từ nhóm câu trả lời dưới đây, hãy chọn câu trả lời đúng để điền vào hộp

trống 0 | 1 trong bảng chân lí của bộ cộng đầy đủ.

Nhóm câu trả lời:

a)	0	0	b)	0	1
c)	1	0	d)	1	1

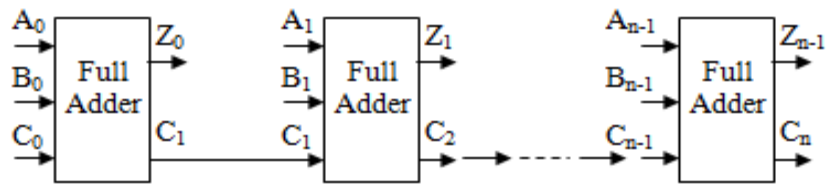
Ý 2: Từ nhóm câu trả lời dưới đây, hãy chọn câu trả lời đúng để điền vào hộp trống a trong mạch logic của bộ cộng đầy đủ.



Nhóm câu trả lời:



Ý 3: Khi một mạch logic được cấu tạo bằng các bộ cộng đầy đủ để cộng các số nhị phân n-chữ số được biểu diễn như phần bù hai, việc cộng các chữ số có ý nghĩa nhất (A_n , B_n và C_n) gây ra sự tràn (phần tô đậm của bảng chân lí của bộ cộng đầy đủ). Mạch logic để phát hiện việc này có thể được cấu tạo bằng một mạch XOR. Hãy chọn từ nhóm câu trả lời dưới đây tổ hợp đúng của các đầu vào X và Y cho mạch logic này.



Ghi chú: $C_0 = 0$



Nhóm câu trả lời:

- | | | |
|-----------------------|-----------------------|-----------------------|
| a) A_{n-1}, B_{n-1} | b) A_{n-1}, Z_{n-1} | c) B_{n-1}, Z_{n-1} |
| d) C_{n-1}, C_n | e) C_{n-1}, Z_{n-1} | f) C_n, Z_{n-1} |

Gợi ý cho ý 3

- Xem ví dụ về cách xác định giá trị cờ tràn OF ở mục 2.3.2 (tài liệu học tập – Giáo trình kiến trúc máy tính).
- Mỗi mạch cộng đầy đủ thực hiện công 2 bit cùng trọng số i [$i = 0 \div (n-1)$] trong 2 số nhị phân. Kết quả đầu ra là Z_i và số nhớ C_{i+1} đưa sang cộng với 2 bit cùng trọng số cao hơn tiếp theo (đưa vào mạch cộng tiếp theo).

Lời giải ý 3

Câu 3.28.

Làm thế nào để xây dựng 1 cổng AND có 2 lối vào từ những cổng NOR 2 lối vào.

Gợi ý

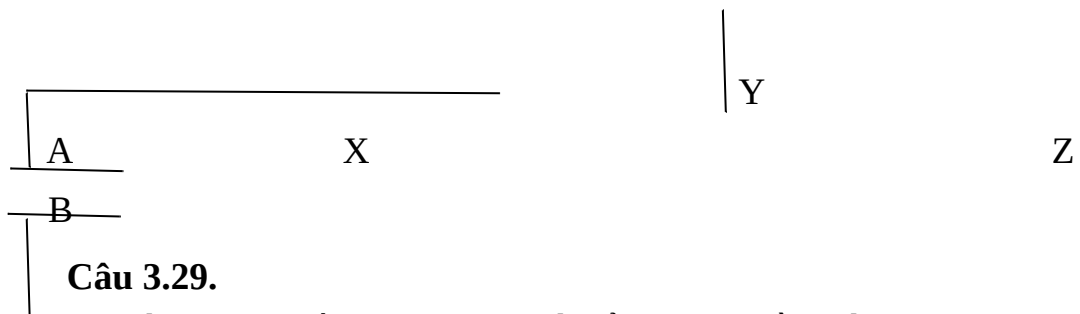
Để xây dựng 1 cổng AND có 2 lối vào từ những cổng NOR 2 lối vào, ta có bảng chân lý sau.

A	B	$X = A \text{ NOR } B$	$Y = A \text{ NOR } X$	$Z = B \text{ NOR } Y$	$K = Y \text{ NOR } Z = (A \text{ AND } B)$
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

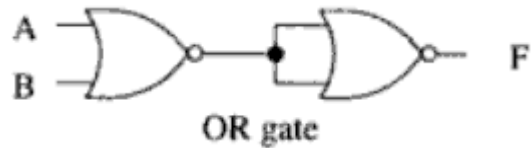
Từ bảng chân lý này ta vẽ được sơ đồ mạch.

Sinh viên thực hiện vẽ sơ đồ mạch

K



Chứng minh rằng mạch sau là cổng OR 2 đầu vào



Bài làm

C1: Ta có: $F = \text{NOR}(\text{NOR}(A,B), \text{NOR}(A,B)) = A + B$ (dpcm)

C2:

Ta có bảng chân lý của mạch OR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Ta thấy:

$F = \text{NOR}(\text{NOR}(A,B), \text{NOR}(A,B))$

Đặt $X = \text{NOR}(A,B)$

$\square F = \text{NOR}(X,X)$

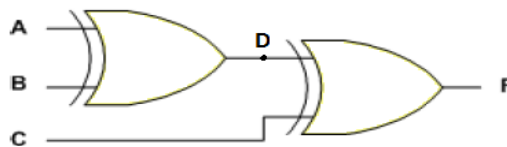
A	B	X	F
0	0	1	0
0	1	0	1

1	0	0	1
1	1	0	1

Đầu ra của mạch trùng với bảng chân lý của mạch OR □ Mạch trên là cổng OR

Câu 3.30.

Chứng minh rằng mạch sau là mạch lẻ 3 đầu vào (số bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0), với đầu vào là A, B, C ; đầu ra là F



Bài làm

Ta có: bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0

$$F = \text{XOR}(\text{XOR}(A, B), C)$$

A	B	C	Kết quả cần	Đầu ra F
0	0	0	0	0
0	1	0	1	1
0	0	1	1	1
0	1	1	0	0
1	1	0	0	0
1	0	1	0	0
1	0	0	1	1
1	1	1	1	1

Câu 3.31.

Chứng minh rằng mạch sau là cổng NOT



Bài làm

C1:Ta có: $F = \text{NOR}(A,A) = \neg A = \neg A$ (dpcm)

C2:

Ta có bảng chân lý của mạch NOT

A	F
0	1
1	0

Ta thấy:

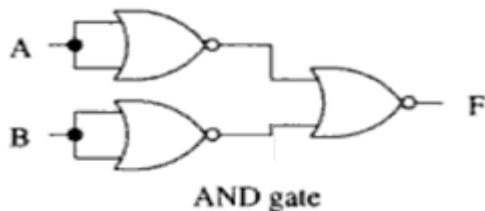
$F = \text{NOR}(A,A)$

A	F
0	1
1	0

□ Đầu ra của mạch trùng với bảng chân lý của mạch NOT □ Mạch trên là cổng NOT

Câu 3.32.

Chứng minh rằng mạch sau là cổng AND 2 đầu vào



Bài làm

C1:Ta có: $F = \text{NOR}(\text{NOR}(A,A), \text{NOR}(B,B))$

Mà $\text{NOR}(A,A) = \neg A$

$\text{NOR}(B,B) = \neg B$

□ $F = \text{NOR}(\neg A, \neg B) = A.B$ (dpcm)

C2:Ta có bảng chân lý của mạch AND

A	B	F
0	0	0
0	1	0

1	0	0
1	1	1

Ta thấy:

$$F = \text{NOR}(\text{NOR}(A, A), \text{NOR}(B, B))$$

$$\text{Đặt } X = \text{NOR}(A, A)$$

$$Y = \text{NOR}(B, B)$$

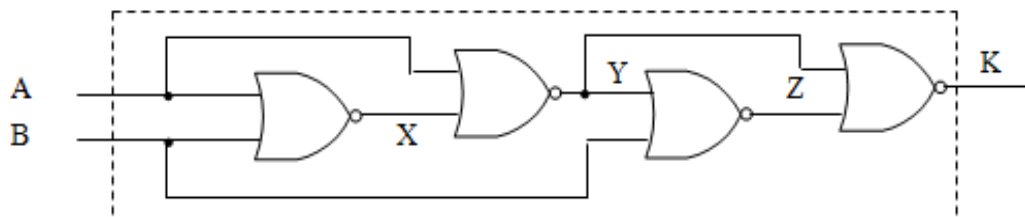
$$F = \text{NOR}(X, Y)$$

A	B	X	Y	F
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Đầu ra của mạch trùng với bảng chân lý của mạch AND. Mạch trên là cổng AND.

Câu 3.33.

Chứng minh rằng đây là mạch AND 2 đầu vào A, B, kết quả ra K.



Bài làm

Ta có bảng chân lý của mạch AND

A	B	K
0	0	0
0	1	0
1	0	0
1	1	1

Ta thấy :

$$X = \text{NOR}(A, B)$$

$$Y = \text{NOR}(X, A)$$

$$Z = \text{NOR}(B, Y)$$

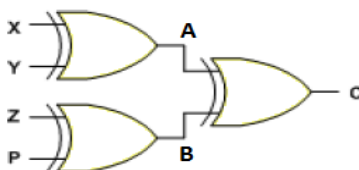
$$K = \text{NOR}(Y, Z)$$

A	B	X	Y	Z	K
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

□ Đầu ra của mạch trùng với bảng chân lý của mạch AND □ Mạch trên là mạch AND 2 đầu vào A và B, đầu ra K

Câu 3.34.

Chứng minh rằng mạch sau là mạch lẻ 4 đầu vào (số bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0), với đầu vào là X, Y, Z, P ; đầu ra là C



Bài làm

Ta có : số bit 1 là lẻ thì đầu ra là 1

$$C = \text{XOR}(\text{XOR}(X, Y), \text{XOR}(Z, P))$$

X	Y	Z	P	Kết quả cần	Đầu ra C
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	0	0	0	0
1	0	1	0	0	0
1	0	0	1	0	0
0	1	1	0	0	0
0	1	0	1	0	0

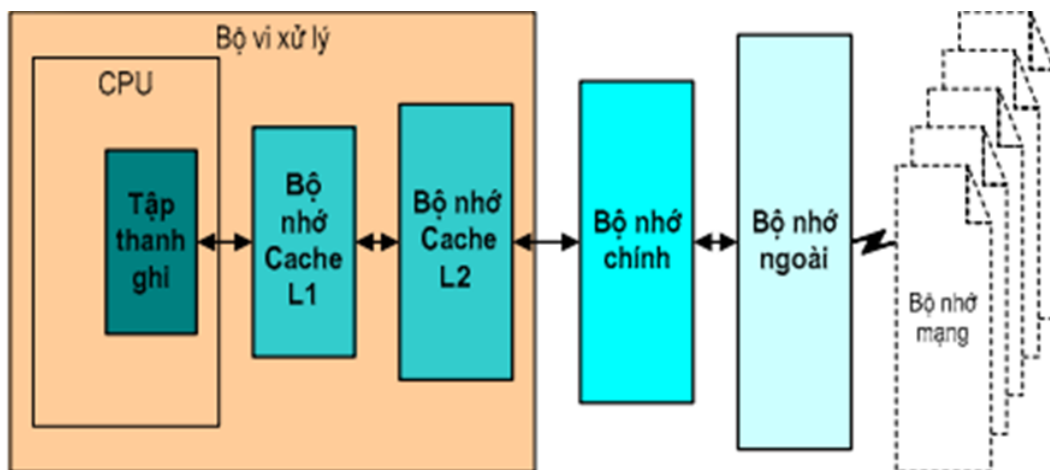
0	0	1	1	0	0
1	1	1	0	1	1
1	1	0	1	1	1
1	0	1	1	1	1
0	1	1	1	1	1
1	1	1	1	0	0

Chương 4. Kiến trúc hệ thống nhớ

Câu 4.1.

Vẽ và nêu ý nghĩa của mô hình phân cấp hệ thống nhớ.

Bài làm



- Ý nghĩa: Từ trái sang phải
 - +Dung lượng tăng dần
 - +Tốc độ giảm dần
 - +Giá thành trên 1 bit giảm dần

Câu 4.2.

Việc xây dựng bộ nhớ cache dựa trên 2 nguyên lý nào?

Bài làm

Việc xây dựng bộ nhớ cache dựa trên 2 nguyên lý:

- Thời gian: Một lệnh hoặc dữ liệu vừa được truy nhập thì thường sẽ được truy nhập ngay sau đó.

- Không gian: Một lệnh hoặc một dữ liệu vừa được truy nhập thì thường những lệnh hoặc dữ liệu lân cận sẽ được truy nhập ngay sau đó.

Câu 4.3.

Trình bày nguyên tắc hoạt động của bộ nhớ cache (trao đổi dữ liệu với bộ nhớ và với CPU như thế nào?).

Bài làm

Nguyên tắc hoạt động của bộ nhớ cache:

- Nguyên lý cục bộ tham chiếu bộ nhớ:
 - + Thời gian: Một lệnh hoặc dữ liệu vừa được truy nhập thì thường sẽ được truy nhập ngay sau đó.
 - + Không gian: Một lệnh hoặc một dữ liệu vừa được truy nhập thì thường những lệnh hoặc dữ liệu lân cận sẽ được truy nhập ngay sau đó.
- Cache có tốc độ nhanh hơn bộ nhớ chính.
- Cache được đặt giữa CPU và bộ nhớ chính nhằm tăng tốc độ CPU truy cập bộ nhớ chính
- Cache có thể được đặt trên chip CPU.

Câu 4.4.

Trình bày phương pháp chung để tổ chức bộ nhớ cache.

Bài làm

Các phương pháp chung để tổ chức bộ nhớ cache :

- Phương pháp ánh xạ trực tiếp
- Phương pháp ánh xạ liên kết toàn phần
- Phương pháp ánh xạ liên kết tập

Câu 4.5.

Trình bày và vẽ sơ đồ thao tác đọc bộ nhớ cache với mỗi khối chứa một byte dữ liệu.

Bài làm

Câu 4.6.

Trình bày thao tác ghi bộ nhớ với mỗi khối chứa một byte dữ liệu.

Bài làm

Câu 4.7.

Trình bày phương pháp ánh xạ trực tiếp trong tổ chức bộ nhớ cache.

Gợi ý

Trình bày phương pháp ánh xạ trực tiếp cùng sơ đồ tổ chức cache

Bài làm

Phương pháp ánh xạ trực tiếp:

- Mỗi Block của bộ nhớ chính chỉ có thể được nạp vào một Line của cache:

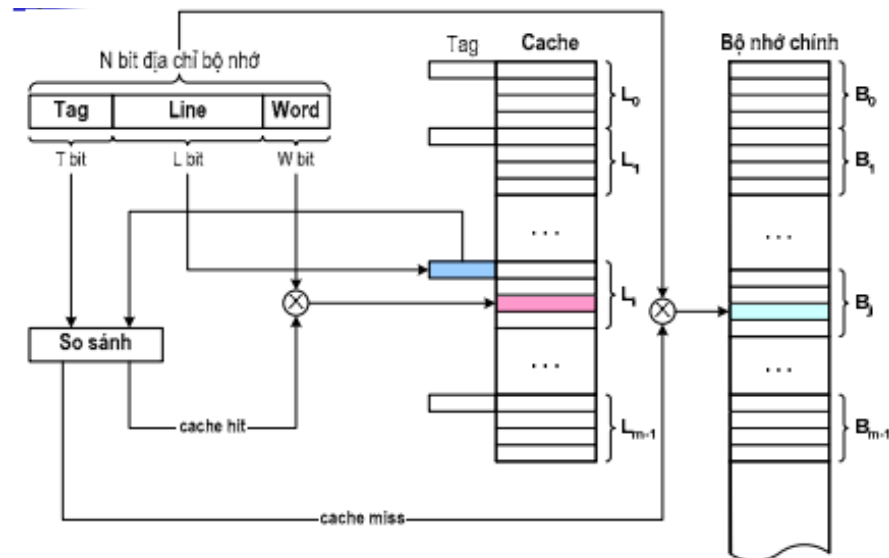
$$+ B_0 \rightarrow L_0$$

$$+ B_1 \rightarrow L_1$$

$$+$$

$$+ B_{m-1} \rightarrow L_{m-1}$$

- + $B_m \rightarrow L_0$
- + $B_{m+1} \rightarrow L_1$
- +
- Tổng quát
 - + B_j chỉ có thể nạp vào $L_{j \bmod m}$
 - + m là số *Line* của *cache*.



Sơ đồ tổ chức cache

Đặc điểm:

- Mỗi 1 địa chỉ N bit của bộ nhớ chính gồm có 3 trường:
 - + Trường Word gồm W bit xác định 1 từ nhớ trong trong Block hay Line :
 - $2^W =$ kích thước của Block hay Line
 - + Trường Line gồm L bit xác định một trong số các Line trong cache:
 - $2^L =$ số Line trong cache = $m \rightarrow$ dung lượng cache = 2^{L+W}
 - + Trường Tag gồm T bit xác định block nhớ cần truy cập
 - $T = N - (W+L)$
- Bộ so sánh đơn giản
- Xác suất cache hit thấp

Câu 4.8.

Trình bày thao tác đọc cache theo phương pháp ánh xạ trực tiếp trong tổ chức bộ nhớ cache.

Bài làm

Thao tác đọc cache theo phương pháp ánh xạ trực tiếp trong tổ chức bộ nhớ:

- B1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / n_{\text{block}} = 2^W (\text{bytes})$$

- B2: Xác định số line của cache

$$n_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 2^L (\text{lines})$$

- B3: Xác định số bits để đánh số hiệu block trong bộ nhớ

$$\text{Số block} = 2^T$$

- B4: Xác định số bits địa chỉ hợp lệ

$$N = W + L + T$$

- B5: Đọc bộ nhớ địa chỉ

- + Xác định tính hợp lệ của địa chỉ

- + Xác định số line trong cache cần truy cập

- + Xác định block của bộ nhớ cần truy cập

- + Xác định block có nạp đúng line của cache không

$$\text{Số hiệu line} = \text{số hiệu block} \bmod \text{số line của cache}$$

- B6. Bộ điều khiển cache sẽ truy cập vào line L của cache và đọc trường Tag của line này, sau đó đem so sánh với số hiệu block cần truy cập. Có 2 khả năng xảy ra.

- + Khả năng 1: Miss cache → **Tag** ≠ số hiệu block

- CPU phải tiến hành nạp cache: trước khi nạp cache, nếu line L có $F = 1$, tức là nội dung có sự thay đổi, CPU phải ghi nội dung của nó ra block nhớ có số hiệu bằng giá trị trên **tag** của line, sửa $F = 0$ (còn trường hợp $F = 0$ thì thao tác này không cần thiết).

- CPU ra bộ nhớ đọc block nhớ, nạp vào line L, sửa tag của line L thành số hiệu block nhớ. Cuối cùng CPU tiến hành đọc byte trong line.

- + Khả năng 2: Hit cache → **Tag** = số hiệu block → CPU đọc byte trong line L có địa chỉ tương đối là T bits thấp của địa chỉ W.

Câu 4.9.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là B7281Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ trực tiếp và địa chỉ ô nhớ cần truy cập trong block cho trường hợp phát ra địa chỉ trên.

Gợi ý

- Hoàn thành và nắm chắc nội dung câu 4.7 và câu 4.8.
- Xem ví dụ 1 trong nội dung *phương pháp ánh xạ trực tiếp* thuộc mục 4.1.3.3. – tài liệu giáo trình kiến trúc máy tính, Đại học CNHN.

Bài làm

B1:Tính dung lượng 1 block

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9 \Rightarrow W = 9 \text{ bits}$$

B2:Xác định số line của cache

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines}) \Rightarrow L = 4 \text{ bits}$$

B3:Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \Rightarrow T = 9 \text{ bits}$$

B4:Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = W + L + T = 9 + 4 + 9 = 22 \text{ bits}$$

B5:Phân tích địa chỉ:

B7281AH = 101101110010100000011010B \Rightarrow Chiều dài địa chỉ là 24 bits \Rightarrow Không hợp lệ về chiều dài

\Rightarrow Địa chỉ không hợp lệ

Câu 4.10.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 2F1025h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ trực tiếp cho trường hợp phát ra địa chỉ trên và địa chỉ ô nhớ cần truy cập trong block theo từng trường hợp.

Bài làm

B1:Tính dung lượng 1 block

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

□ $W = 9$ bits

B2: Xác định số line của cache

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines}) \quad \square L = 4 \text{ bits}$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \quad \square T = 9 \text{ bits}$$

B4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = W + L + T = 9 + 4 + 9 = 22 \text{ bits}$$

B5: Phân tích địa chỉ:

2F1025H = 1011110001000000100101B □ Chiều dài địa chỉ là 22 bits □ Hợp lệ về chiều dài

$$L = 1000B = 8$$

$$BI = 101111000B = 376$$

Block 376 phải nạp vào line: $376 \bmod 16 = 8$ □ Block nạp đúng line □ Địa chỉ hợp lệ

□ CPU phát địa chỉ của ô nhớ cần truy cập ra bus địa chỉ.

B6: Bộ điều khiển cache sẽ truy cập vào line 8 của cache và đọc trường Tag của line này, sau đó đem so sánh với số hiệu block cần truy cập 376. Có 2 khả năng xảy ra.

- Khả năng 1: Miss cache → **Tag** ≠ 376.

+ CPU phải tiến hành nạp cache: trước khi nạp cache, nếu line 8 có $F = 1$, tức là nội dung có sự thay đổi, CPU phải ghi nội dung của nó ra block nhớ có số hiệu bằng giá trị trên **tag** của line, sửa $F = 0$ (còn trường hợp $F = 0$ thì thao tác này không cần thiết).

+ CPU ra bộ nhớ đọc block nhớ 376, nạp vào line 8, sửa tag của line 8 thành 376. Cuối cùng CPU tiến hành đọc byte trong line.

- Khả năng 2: Hit cache → **Tag** = 376 → CPU đọc byte trong line 8 có địa chỉ tương đối là 9 bits thấp của địa chỉ $W = 000100101 = 37$

+ Địa chỉ vật lý của ô nhớ trong bộ nhớ chính được tính bằng cách ghép địa chỉ của block và địa chỉ của từ nhớ lại.

$$\rightarrow \text{Địa chỉ vật lý} = TW$$

$$\rightarrow \text{Địa chỉ vật lý} = 101111000000100101$$

$$= 2F025H$$

Câu 4.11.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 2F0825h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ trực tiếp cho trường hợp phát ra địa chỉ trên và địa chỉ ô nhớ cần truy cập trong block theo từng trường hợp.

Bài làm

B1: Tính dung lượng 1 block

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

□ W = 9 bits

B2: Xác định số line của cache

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines}) \quad \square L = 4 \text{ bits}$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \quad \square T = 9 \text{ bits}$$

B4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = W + L + T = 9 + 4 + 9 = 22 \text{ bits}$$

B5: Phân tích địa chỉ:

2F0825H = 1011110000100000100101B □ Chiều dài địa chỉ là 22 bits □ Hợp lệ về chiều dài

$$L = 0100B = 4 \text{ bits}$$

$$BI = 101111000B = 376$$

Block 376 phải nạp vào line: $376 \bmod 16 = 8$ □ Block nạp không đúng line □ Địa chỉ không hợp lệ.

Câu 4.12.

Trình bày phương pháp ánh xạ liên kết hoàn toàn trong tổ chức bộ nhớ cache.

Gợi ý

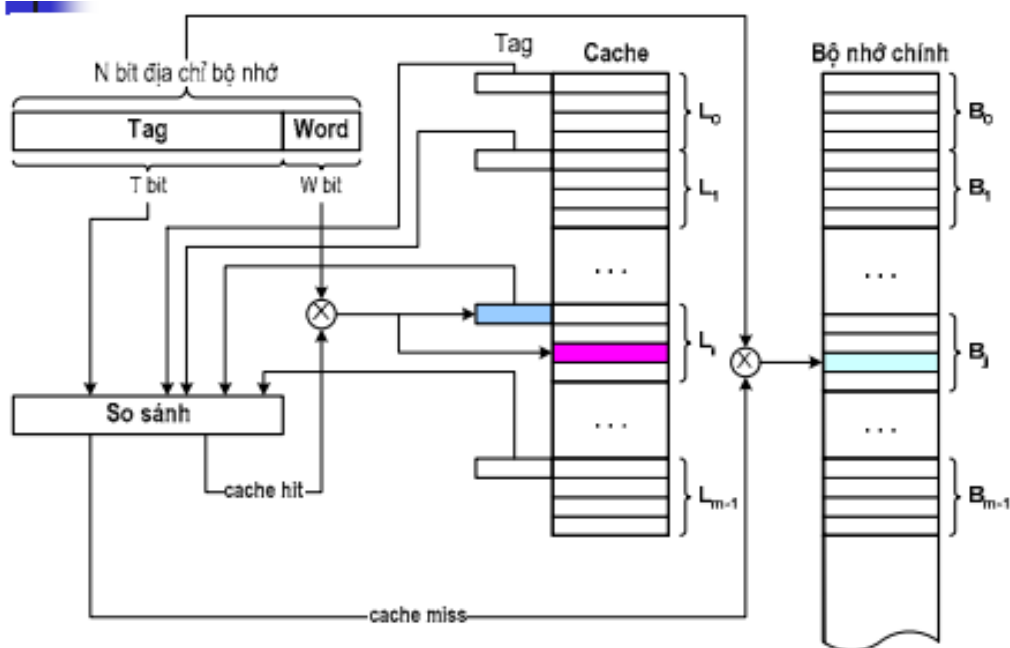
Trình bày phương pháp tổ chức cache theo kỹ thuật ánh xạ liên kết hoàn toàn cùng sơ đồ tổ chức cache.

Bài làm

Phương pháp ánh xạ liên kết hoàn toàn:

- Mỗi Block có thể được nạp vào line bất kỳ của cache.
- Địa chỉ của bộ nhớ chính bao gồm hai trường:

- + Trường **Word** giống như trường hợp ở trên.
- + Trường **Tag** dùng để xác định *Block* của bộ nhớ chính.
- Tag xác định Block nào đang nằm ở Line đó.



Sơ đồ tổ chức cache kỹ thuật ánh xạ liên kết hoàn toàn

- Đặc điểm:
 - + Phải so sánh đồng thời với tất cả các Tag → Tốn thời gian so sánh.
 - + Bộ so sánh phức tạp.
 - + Tỷ lệ cache hit cao hơn phương pháp ánh xạ trực tiếp.

Câu 4.13.

Trình bày thao tác đọc cache theo phương pháp ánh xạ liên kết hoàn toàn trong tổ chức bộ nhớ cache.

Bài làm

- B1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / n_{\text{block}} = 2^W (\text{bytes})$$

- B2: Xác định số bit cần thiết để đánh số hiệu block trong bộ nhớ.

$$\text{Số block} = 2^T \rightarrow \text{Cần } T \text{ bits để đánh số hiệu block.}$$

- B3: Xác định số bit địa chỉ hợp lệ cho bộ nhớ là: $T + W$ (bits)
- B4: Đọc cache địa chỉ :
 - + Chuyển địa chỉ từ hệ hex sang hệ nhị phân:

- + Kiểm tra xem địa chỉ cần truy cập có hợp lệ không: Hợp lệ nếu số bit của địa chỉ là (T+W) bằng số bit địa chỉ hợp lệ → CPU phát địa chỉ này lên bus địa chỉ.
- + Bộ điều khiển cache xác định giá trị của trường Tag là T bit cao của địa chỉ trên để xác định số hiệu block bộ nhớ cần truy cập
- + Bộ điều khiển cache sẽ so sánh trường Tag của tất cả các line.

Có 2 khả năng xảy ra:

- + Trường hợp 1: Không có Tag của line nào có giá trị số hiệu block A → Block cần truy cập chưa có trong cache (miss cache). CPU phải nạp block có số hiệu A từ bộ nhớ chính vào cache. CPU tìm line nào trong cache sử dụng kém hiệu quả nhất để thay bằng block mới. Nếu bit F của line bị thay thế = 1 thì nội dung của line đó sẽ được ghi trả lại bộ nhớ vào đúng block có giá trị bằng trường Tag của line đó. Sau đó xóa F = 0 rồi nạp block nhớ có số hiệu A vào line này và cập nhật trường Tag của nó bằng A. Sau đó CPU sẽ đọc byte cần trong line này.
- + Trường hợp 2: Có một Tag của line nào đó = A (hit cache). CPU sẽ đọc byte có số hiệu là 9 bit thấp của địa chỉ: W → đọc byte có số hiệu của line đó.

Địa chỉ vật lý của ô nhớ cần truy cập chính là: địa chỉ truy cập bộ nhớ

Câu 4.14.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 3280Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết hoàn toàn và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

Gợi ý

- Hoàn thành và nắm chắc nội dung câu 4.13 và câu 4.14.
- Xem ví dụ 2 trong nội dung *phương pháp ánh xạ liên kết hoàn toàn* thuộc mục 4.1.3.3. – tài liệu giáo trình kiến trúc máy tính, Đại học CNHN.

Bài làm

B1: Tính dung lượng 1 block

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

□ W=9 bits

B2: Xác định số line của cache

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines}) \quad L = 4 \text{ bits}$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \quad T = 9 \text{ bits}$$

B4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = W + T = 9 + 9 = 18 \text{ bits}$$

B5: Phân tích địa chỉ:

$$3280\text{AH} = 110010100000001010\text{B} \quad \text{Chiều dài địa chỉ là 18 bits}$$

Hợp lệ về chiều dài

□ Địa chỉ hợp lệ

□ CPU phát địa chỉ này lên bus địa chỉ.

+ Bộ điều khiển cache xác định giá trị của trường Tag là 9 bit cao của địa chỉ trên để xác định số hiệu block bộ nhớ cần truy cập: $110010100 = 404$

+ Bộ điều khiển cache sẽ so sánh trường Tag của tất cả các line với 404.

Có 2 khả năng xảy ra.

+ Trường hợp 1: Không có Tag của line nào có giá trị 404 → Block cần truy cập chưa có trong cache (miss cache). CPU phải nạp block có số hiệu 404 từ bộ nhớ chính vào cache. CPU tìm line nào trong cache sử dụng kém hiệu quả nhất để thay bằng block mới. Nếu bit F của line bị thay thế = 1 thì nội dung của line đó sẽ được ghi trả lại bộ nhớ vào đúng block có giá trị bằng trường Tag của line đó. Sau đó xóa F = 0 rồi nạp block nhớ có số hiệu 404 vào line này và cập nhật trường Tag của nó bằng 404. Sau đó CPU sẽ đọc byte cần trong line này.

+ Trường hợp 2: Có một Tag của line nào đó = 404 (hit cache). CPU sẽ đọc byte có số hiệu là 9 bit thấp của địa chỉ: $000001010 = 10$ → đọc byte có số hiệu 10 của line đó.

□ Địa chỉ vật lý của ô nhớ cần truy cập chính là: 3280AH

Câu 4.15.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 7280Ah. Hãy trình bày chi tiết phương

pháp đọc cache theo kỹ thuật ánh xạ liên kết hoàn toàn và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

Bài làm

B1: Tính dung lượng 1 block

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

□ $W = 9$ bits

B2: Xác định số line của cache

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines}) \quad \square L = 4 \text{ bits}$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \quad \square T = 9 \text{ bits}$$

B4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = W + T = 9 + 9 = 18 \text{ bits}$$

B5: Phân tích địa chỉ:

7280AH = 111001010001010B □ Chiều dài địa chỉ là 19 bits □
Không hợp lệ về chiều dài
□ Địa chỉ không hợp lệ

Câu 4.16.

Trình bày phương pháp ánh xạ liên kết tập hợp trong tổ chức bộ nhớ cache.

Gợi ý

Trình bày phương pháp xây dựng cache theo kỹ thuật ánh xạ liên kết tập hợp cùng sơ đồ tổ chức cache.

Bài làm

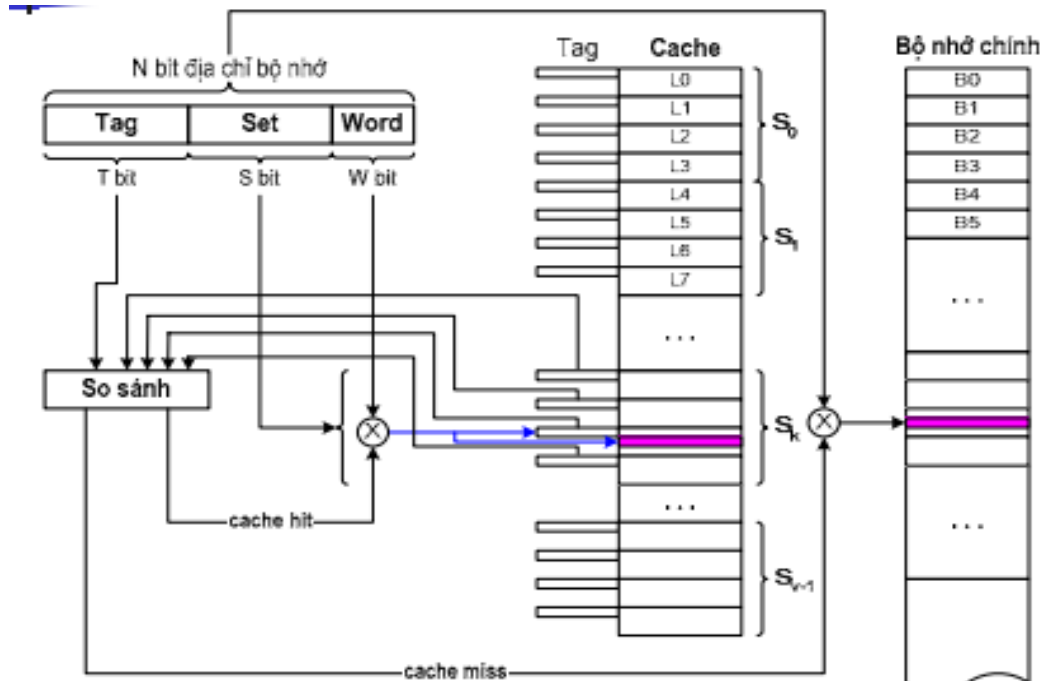
Phương pháp xây dựng cache theo kỹ thuật ánh xạ liên kết tập hợp:

- Cache được chia thành các Tập (Set)
- Mỗi một Set chứa một số Line
- Ví dụ:
 - + 4 Line/Set → 4-way associative mapping
- Ánh xạ theo nguyên tắc sau:
 - + $B_0 \rightarrow S_0$
 - + $B_1 \rightarrow S_1$

$$+ B_2 \rightarrow S_2$$

+ ...

$$+ B_m \rightarrow S_{m \bmod q}$$



Sơ đồ tổ chức cache theo kỹ thuật ánh xạ liên kết tập hợp

- Đặc điểm của ánh xạ liên kết tập hợp

N bits địa chỉ hợp lệ được chia làm 3 phần:

- + W bits thấp nhất dùng để xác định số hiệu của từ nhớ trong line (block) cần truy cập \rightarrow kích thước của một block = 2^W
- + S bits kế tiếp dùng để xác định một trong 2^S set cần truy cập.
- + T bits trọng số cao nhất dùng để xác định block cần truy cập.

$$\rightarrow N = T + S + W$$

Câu 4.17.

Trình bày thao tác đọc cache theo phương pháp ánh xạ liên kết tập hợp trong tổ chức bộ nhớ cache.

Bài làm

- SI (Set Index)– là số hiệu Set cần truy nhập.
- BI (Block Index)- là số hiệu Block nhớ cần truy nhập.
- + Bước 1: CPU kiểm tra tính hợp lệ của địa chỉ truy nhập.

- Trường hợp 1: Nếu số bit có nghĩa biểu diễn địa chỉ truy nhập lớn hơn số bit hợp lệ (lớn hơn n), thì địa chỉ này không hợp lệ, tiến trình phải dừng.

- Trường hợp 2: Nếu SI # (BI mod q), thì địa chỉ này không hợp lệ, tiến trình phải dừng (Số lượng Set trong cache là q).
- Ngược lại, CPU thực hiện phát ra địa chỉ truy nhập lên Bus A, sang bước 2.

+ Bước 2: Đọc cache

Câu 4.18.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB chia làm 4 set. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 22A07h. Hãy trình bày chi tiết phương pháp đọc cache và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

Gợi ý

- Hoàn thành và nắm chắc nội dung câu 4.13 và câu 4.14.
- Xem ví dụ 3 trong nội dung *phương pháp ánh xạ liên kết tập hợp* thuộc mục 4.1.3.3. – tài liệu giáo trình kiến trúc máy tính, Đại học CNHN.

Bài làm

B1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

□ W=9 bits

B2: Xác định số bit để đánh số hiệu cho tập(set) của cache

$$n_{\text{set}} = 4 = 2^2 \square S = 2 \text{ bits}$$

+ Tính số lines của cache:

$$n_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines})$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}} / n_{\text{set}} = 16 / 4 = 4 (\text{line/1set})$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \square T = 9 \text{ bits}$$

B4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = T + S + W = 9 + 2 + 9 = 20 \text{ bits}$$

B5: Xác định địa chỉ CPU phát ra có hợp lệ không:

22A07H = 100010101000000111B □ Chiều dài địa chỉ là 18 bits
chiều dài tối đa địa chỉ hợp lệ □ Địa chỉ hợp lệ về chiều dài(1)

$$SI=01B=1$$

$$BI=1000101B=69$$

Theo ánh xạ liên kết tập hợp thì block 69 phải nạp vào set sau:

$69 \bmod 4 = 1$ □ Block nạp đúng set (2) □ Từ (1) và (2) □ Địa chỉ CPU phát là hợp lệ

B6: Truy xuất cache

Từ địa chỉ CPU phát ra thì đơn vị điều khiển cache sẽ đi kiểm tra 4 line của set 2 xem trường tag có bằng với 69 không?

Có 2 khả năng xảy ra:

- Không có trường tag nào bằng 69 □ miss cache, Khi đó CPU phải nạp block nhớ có số hiệu 69 vào một line của Set 2 → như trường hợp 2.
- Có 1 tag của line nào đó trong set 2 bằng 69 □ hit cache □ CPU đọc ô nhớ có số hiệu(địa chỉ lệch) $W=000000111B=7$

Địa chỉ vật lý: $TW=001000101000000111B=08A07H$

Câu 4.19.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 2280Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết tập hợp và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

Bài làm

B1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256KB / 512 = 256 \times 1024 / 512 = 0.5KB = 512 = 2^9$$

□ $W=9$ bits

B2: Xác định số bit để đánh số hiệu cho tập(set) của cache

$$n_{\text{set}} = 4 = 2^2 \quad S=2 \text{ bits}$$

+ Tính số lines của cache:

$$n_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8KB / 0.5KB = 16 = 2^4 (\text{lines})$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}} / n_{\text{set}} = 16 / 4 = 4 (\text{line/1set})$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \quad T=9 \text{ bits}$$

B4:Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N=T+S+W=9+2+9= 20 \text{ bits}$$

B5: Xác định địa chỉ CPU phát ra có hợp lệ không:

2280AH=100010100000001010B □ Chiều dài địa chỉ là 18 bits
chiều dài tối đa địa chỉ hợp lệ □ Địa chỉ hợp lệ về chiều dài(1)

$$SI=00B=0$$

$$BI=000001010B=10$$

Theo ánh xạ liên kết tập hợp thì block 10 phải nạp vào set sau:

$10 \bmod 4=2$ □ Block nạp không đúng set (2) □ Từ (1) và (2) □ Địa chỉ CPU phát là không hợp lệ

Câu 4.20.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 120812h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết tập hợp và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

Bài làm

B1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}}=C_{\text{memory}}/N_{\text{block}}=256\text{KB}/512=256\times 1024/512=0.5\text{KB}=512=2^9$$

□ $W=9 \text{ bits}$

B2: Xác định số bit để đánh số hiệu cho tập(set) của cache

$$n_{\text{set}}=4=2^2 \square S=2 \text{ bits}$$

+Tính số lines của cache:

$$n_{\text{line}}=C_{\text{cache}}/C_{\text{block}}=8\text{KB}/0.5\text{KB}=16=2^4(\text{lines})$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}}/n_{\text{set}}=16/4=4(\text{line}/1\text{set})$$

B3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}}=512=2^9 \square T=9 \text{ bits}$$

B4:Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N=T+S+W=9+2+9= 20 \text{ bits}$$

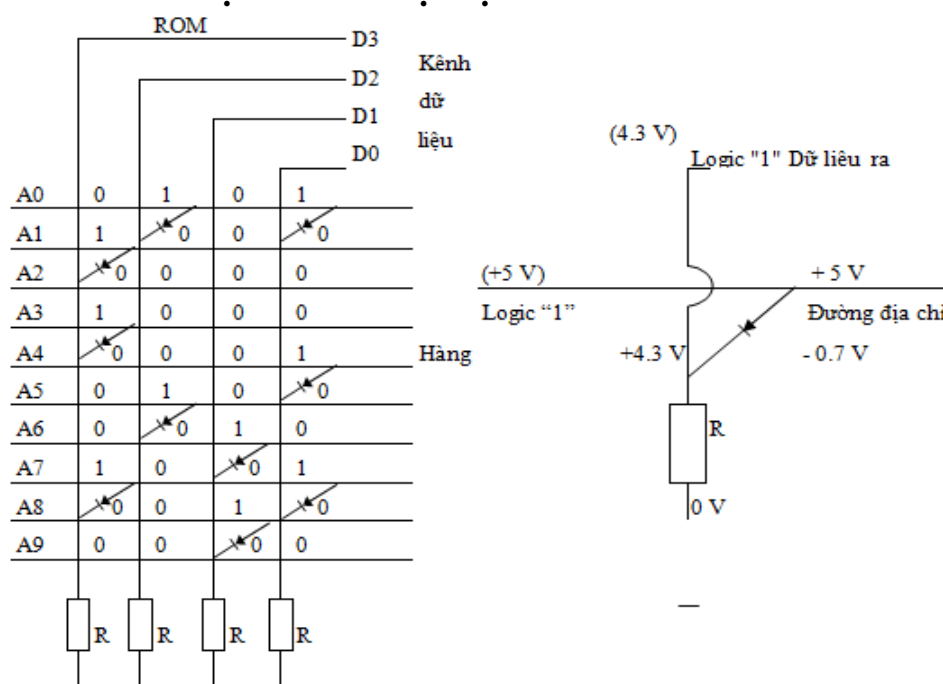
B5: Xác định địa chỉ CPU phát ra có hợp lệ không:

120812H=100100000100000010010B □ Chiều dài địa chỉ là 21 bits
chiều dài tối đa địa chỉ hợp lệ □ Địa chỉ không hợp lệ về chiều dài

□Địa chỉ CPU phát là không hợp lệ

Câu 4.21.

Cho sơ đồ mạch ROM mặt nạ sau:

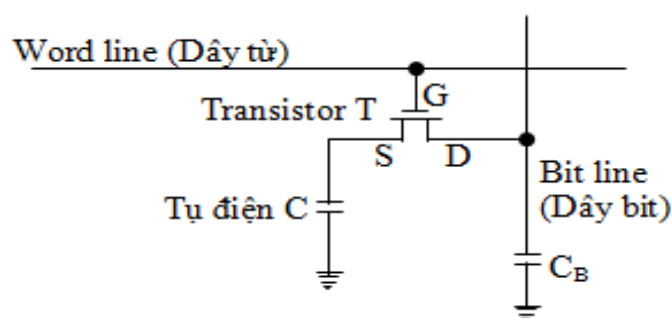


Hãy xác định giá trị dữ liệu tại các địa chỉ tương ứng sau:

Địa chỉ	Giá trị dữ liệu
000000011b	
000000101b	
000000111b	

Câu 4.22.

Trình bày nguyên lý ghi, đọc của phần tử nhớ động 1 bóng sau:



Bài làm

- Ghi:

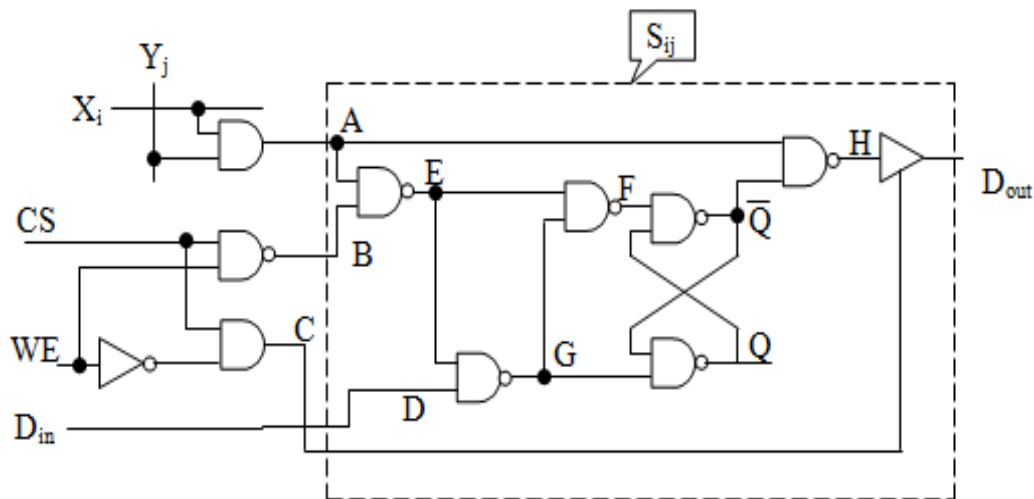
+ Address line =1 □ T thông □ Dòng điện từ đường Bit line qua T thông nạp đầy cho tụ C (lưu bit 1)

- Đọc;

+ Address line = 1 □ T thông □ Điện tích tụ C phóng qua T thông ra đường dây Bit line □ Căn cứ vào I ở đường Bit line biết là 0 hay 1

Câu 4.23.

Cho phần tử nhớ sau:



a. Chứng minh với $Y_j = 1$, $X_i = 1$, $WE = 1$ và $CS = 1$ thì $Q = D_{in}$ đồng thời H và D_{out} ngắt mạch:

Ta có: $Y_j = 1$, $X_i = 1$, $WE = 1$, $CS = 1$ CM $Q = D_{in}$ đồng thời H và D_{out} ngắt mạch

$WE = 1 \Rightarrow \text{inverted} = 0$

$C = \text{AND}(, CS) = \text{AND}(0, 1) = 0$ thiết bị 3 trạng thái K ở trạng thái trở kháng cao □ H và D_{out} ngắt mạch

$B = \text{NAND}(WE, CS) = \text{NAND}(1, 1) = 0$

$A = \text{AND}(X_i, Y_j) = \text{AND}(1, 1) = 1$

$E = \text{NAND}(B, A) = \text{NAND}(0, 1) = 1$

Giả sử $D_{in} = 0$ CM $Q = 0$

$G = \text{NAND}(D_{in}, E) = \text{NAND}(0, 1) = 1$

$F = \text{NAND}(G, E) = \text{NAND}(1, 1) = 0$

$= \text{NAND}(F, Q) = \text{NAND}(0, Q) = 1$

□ $Q = \text{NAND}(G,) = \text{NAND}(1, 1) = 0(1)$

Giả sử $D_{in} = 1$ CM $Q = 1$

$G = \text{NAND}(D_{in}, E) = \text{NAND}(1, 1) = 0$

$$F = \text{NAND}(G, E) = \text{NAND}(0, 1) = 1$$

$$Q = \text{NAND}(G,) = \text{NAND}(0,) = 1(2)$$

Từ (1) và (2) $\square Q = D_{in}$

b. $Y_j = 1, X_i = 1, WE = 0$ và $CS = 1$ thì Q không phụ thuộc D_{in} và $D_{out} = Q$:

Ta có:

$$WE = 0 \square = 1$$

$$C = \text{AND}(, CS) = \text{AND}(1, 1) = 1$$

$$B = \text{NAND}(WE, CS) = \text{NAND}(0, 1) = 1$$

$$A = \text{AND}(X_i, Y_j) = \text{AND}(1, 1) = 1$$

$$E = \text{NAND}(B, A) = \text{NAND}(1, 1) = 0$$

$$G = \text{NAND}(D_{in}, E) = \text{NAND}(D_{in}, 0) = 1$$

$$F = \text{NAND}(G, E) = \text{NAND}(1, 0) = 1$$

Xét $Q(t-1) = 0$

$$(t) = \text{NAND}(F, Q(t-1)) = \text{NAND}(1, 0) = 1 \square Q(t) = 0(1)$$

Xét $Q(t-1) = 1$

$$(t) = \text{NAND}(F, Q(t-1)) = \text{NAND}(1, 1) = 0 \square Q(t) = 1(2)$$

Từ (1) và (2) $\square Q(t) = Q(t-1) \square Q$ không đổi $\square Q$ không phụ thuộc vào D_{in}

$$D_{out} = \text{NAND}(A,) = \text{NAND}(1,) = Q$$

c. $Y_j = 1, X_i = 1, WE = 0/1$ và $CS = 0$ thì Q không phụ thuộc D_{in} , đồng thời H và $Dout$ ở trạng thái trở kháng cao (ngắt mạch).

$$WE = 0 \square = 1$$

$C = \text{AND}(CS,) = \text{AND}(0, 1) = 0 \square$ thiết bị 3 trạng thái K ở trạng thái trở kháng cao $\square H$ và $Dout$ ngắt mạch

$$B = \text{NAND}(WE, CS) = \text{NAND}(0, 1) = 1$$

$$A = \text{AND}(X_i, Y_j) = \text{AND}(1, 1) = 1$$

$$E = \text{NAND}(B, A) = \text{NAND}(1, 1) = 0$$

$$G = \text{NAND}(D_{in}, E) = \text{NAND}(D_{in}, 0) = 1$$

$$F = \text{NAND}(G, E) = \text{NAND}(1, 0) = 1$$

Xét $Q(t-1) = 0$

$$(t) = \text{NAND}(F, Q(t-1)) = \text{NAND}(1, 0) = 1 \square Q(t) = 0(1)$$

Xét $Q(t-1) = 1$

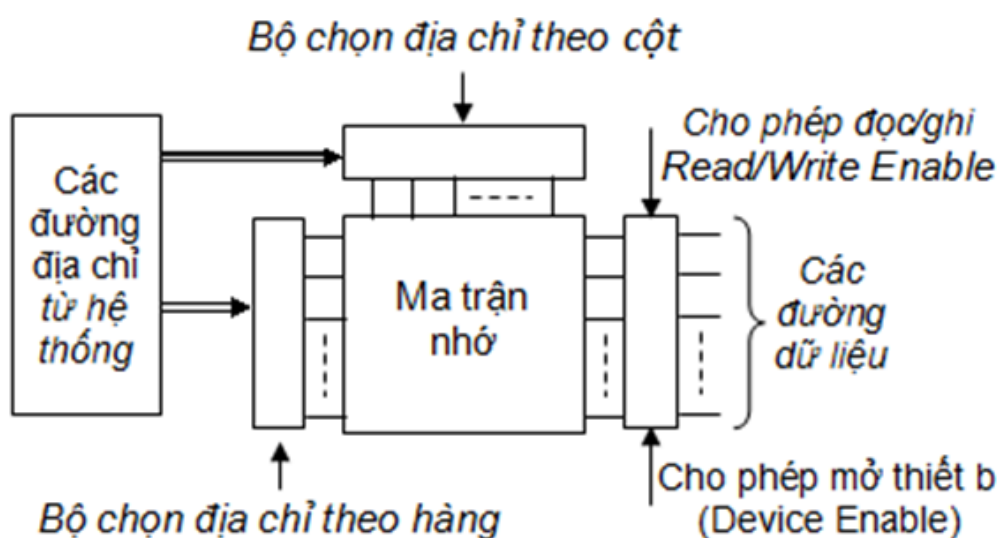
$$(t) = \text{NAND}(F, Q(t-1)) = \text{NAND}(1, 1) = 0 \square Q(t) = 1(2)$$

Từ (1) và (2) $Q(t) = Q(t-1) \oplus Q$ không đổi \square Q không phụ thuộc vào D_{in}

Câu 4.24.

Vẽ sơ đồ khối ghép nối giữa CPU và bộ nhớ chính, trình bày chức năng của các đường dây ghép nối.

Bài làm



Sơ đồ khối ghép nối giữa CPU và bộ nhớ chính

Chức năng các đường dây ghép nối:

- Mạch này gồm hai phần: mạch chọn địa chỉ theo hàng và mạch chọn địa chỉ theo cột. Các đường dây địa chỉ sẽ chọn địa chỉ hàng và cột.
- Đường dây enable dùng để mở các mạch điện lối ra bộ nhớ theo ba trạng thái.
- Còn đường dây Read/write quyết định dạng thao tác sẽ thực hiện.

Câu 4.25.

Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý tương ứng với các địa chỉ logic phát ra là $\langle 2, 43h \rangle$, $\langle 3, 4FFh \rangle$, $\langle 4, 600h \rangle$, $\langle 5, 100h \rangle$, cho biết bảng quản lý phân đoạn như sau:

P	A	L
1	100h	200h
0	-	400h
1	300h	700h
0	-	500h
1	A00h	600h

Bài làm

a. Với các địa chỉ logic phát ra là $\langle 2, 43h \rangle$

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.

- Số hiệu đoạn cần truy nhập là $S = 2 < 5$, địa chỉ đoạn cần truy nhập là hợp lệ.

- Phần tử 2 trong bảng SCB, cho biết đoạn 2 có độ dài là $L = 700h$ byte, so sánh : $L = 700h > d = 43h$, nên địa chỉ lệch cần truy nhập là hợp lệ

- Phần tử 2 trong bảng SCB có $P = 1$, cho biết đoạn 2 đã được nạp vào vùng nhớ có địa chỉ đầu $A = 300h$.

- Địa chỉ vật lý cần truy nhập $= A + d = 300h + 43h = 343h$

b. Với các địa chỉ logic phát ra là $\langle 4, 4FFh \rangle$

$S=4 < 5$ □ Thỏa mãn □ Lấy thông tin mục vào 4 (dòng 4, dòng đánh số từ 0)

- Có $P=1$ □ Module 4 đã nạp vào RAM

- Có $A=A00h$ □ Module 4 nạp vào vị trí bắt đầu là A00h

- Có $L=600h$ □ Kích thước module 4 là 600h

□ $d=A00h < L=4FFh$ □ Thỏa mãn

□ Địa chỉ vật lý $= A + d = A00h + 4FFh = EFFh$

c. Với các địa chỉ logic phát ra là $\langle 3, 4FFh \rangle$

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.

- Số hiệu đoạn cần truy nhập là $S = 3 < 5$, địa chỉ đoạn cần truy nhập là hợp lệ.

- Phần tử 3 trong bảng SCB, cho biết đoạn 3 có độ dài là $L = 500h$ byte, so sánh : $L = 500h > d = 4FFh$, nên địa chỉ lệch cần truy nhập là hợp lệ

- Phần tử 3 trong bảng SCB có $P = 0$, cho biết đoạn 3 đã chưa được nạp bộ nhớ chính. Do vậy hệ thống cần thực hiện nạp đoạn hoặc đổi đoạn để tiến hành tiếp tục thực thi.

d. Với các địa chỉ logic phát ra là <1, 306h>

$S=1 < 5$ □ Thỏa mãn □ Lấy thông tin mục vào 1 (dòng 1, dòng đánh số từ 0)

- Có $L=400h$ □ Kích thước module là 400h

□ $d=306h < L=400h$ □ Thỏa mãn

- Có $P=0$ □ Module 1 không nạp vào RAM

□ Hệ thống cần thực hiện nạp đoạn hoặc đổi đoạn để tiến trình tiếp tục thực thi

e. Với các địa chỉ logic phát ra là <4, 600h>

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.

- Số hiệu đoạn cần truy nhập là $S = 4 < 5$, địa chỉ đoạn cần truy nhập là hợp lệ.

- Phần tử 4 trong bảng SCB, cho biết đoạn 4 có độ dài là $L = 500h$ byte, so sánh : $L = 600h \leq d = 600h$, nên địa chỉ lệch cần truy nhập không hợp lệ, tiến trình dừng.

f. Với các địa chỉ logic phát ra là <2, 701h>

$S=2 < 5$ □ Thỏa mãn □ Lấy thông tin mục vào 2 (dòng 2, dòng đánh số từ 0)

+ Có $P=1$ □ Module 2 đã nạp vào RAM

+ Có $A=300h$ □ Module 2 nạp vào vị trí bắt đầu là 300h

+ Có $L=700h$ □ Kích thước module 2 là 700h

□ $d=701h > L=700h$ □ Không thỏa mãn □ Địa chỉ lệch truy nhập không hợp lệ

g. Với các địa chỉ logic phát ra là <5, 33h>

$S=5 < 5$ □ Không thỏa mãn

□ Địa chỉ truy nhập không hợp lệ

Câu 4.26.

Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 18 bytes có nội dung '0123456789ABCDEFGH'. Và bảng quản lý trang (PCB) có nội dung sau:

P	A
1	7
0	-
1	3
1	5
0	-

- Xác định địa chỉ vật lý truy nhập tương ứng với các địa chỉ logic sau: $\langle 0,2 \rangle$; $\langle 2,3 \rangle$; $\langle 1,3 \rangle$; $\langle 4,1 \rangle$; $\langle 2,4 \rangle$; $\langle 3,5 \rangle$; $\langle 5,1 \rangle$
- Cho biết nội dung các ô nhớ tương ứng với các địa chỉ trên.

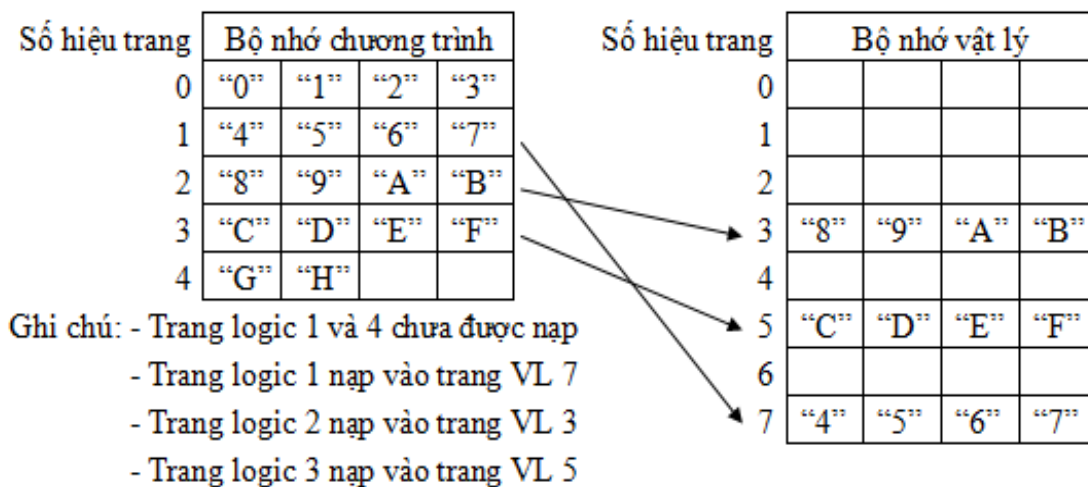
Bài làm

* Phần tính toán chung

- Kích thước một trang vật lý là: $L = \text{dung lượng bộ nhớ vật lý} / \text{số trang vật lý}$.

$$= 32/8 = 4 \text{ bytes} = 2^2.$$

- Vậy số bit để đánh độ lệch trong trang là: $n_1 = 2$.
- Số bit để đánh số hiệu trang vật lý là: $8 = 2^3$, vậy $n_2 = 3$.
- Số bit để đánh địa chỉ vật lý là: $32 = 2^5$, vậy $n = 5$ ($n = n_2 + n_1$).
- Số trang logic của chương trình = dung lượng chương trình / $L = 18/4 = 4,5$. Vậy số trang logic của chương trình phải là 5. (bảng PCB trên là phù hợp).
- Căn cứ vào bảng PCB, dung lượng bộ nhớ vật lý và kích thước của một trang, ta có sơ đồ nạp trang như sau:



a. Xác định địa chỉ vật lý (ĐCVL) truy nhập tương ứng với địa chỉ logic $\langle 0,2 \rangle$

- Số hiệu trang logic cần truy nhập là: $N_p = 0 < 5$ (số trang logic của chương trình). nên số hiệu trang cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là: $d = 2 < 4$ (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là hợp lệ.

- Phần tử 0 trong bảng PCB có $P = 1$, cho biết trang logic 1 đã được nạp vào trang vật lý $A = 7$. $A = 7 < 8$ (số trang vật lý), nên giá trị này là hợp lệ.

- ĐCVL cần truy nhập là:

$$\begin{aligned}\text{ĐCVL} &= A \text{ SHL } n_1 \text{ OR } d = 7 \text{ SHL } 2 \text{ OR } 2 = 0111\text{b} \text{ SHL } 2 \text{ OR } 10\text{b} \\ &= 11100\text{b} \text{ OR } 10\text{b} = 11110\text{b} = 30.\end{aligned}$$

- Nhìn vào sơ đồ ta thấy nội dung ô nhớ có địa chỉ 30 là “6”.

b. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <2,3>

$N_p = 2 < 5$ □ Thỏa mãn. Từ dòng 2 của PCB có:

- $P = 1$ □ Trang 2 đã nạp vào RAM

- $A = 3$ □ Trang 2 đã nạp vào trong địa chỉ vật lý

$d = 3 < 4$ □ Thỏa mãn

□ Địa chỉ vật lý cần truy nhập là:

$$PA = A * L + d = 3 * 4 + 3 = 15$$

Hoặc

Dịch trái $A = 3$ đi $n_1 = 2$ bit OR $d = 3$

$$\begin{array}{r} 001100\text{b} \\ + \quad 0011\text{b} \\ \hline 001111\text{b} = 15 \end{array}$$

8
9
A
B

12

13 □ Nội dung: “B”

14

15

c. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <1,3>

- Số hiệu trang logic cần truy nhập là: $N_p = 1 < 5$ (số trang logic của chương trình) . nên số hiệu trang cần truy nhập là hợp lệ.

- Địa chỉ lệch cần truy nhập trong trang là: $d = 3 < 4$ (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là hợp lệ.

- Phần tử 1 trong bảng PCB có $P = 0$, cho biết trang logic 0 chưa được nạp vào bộ nhớ vật lý, nên hệ thống cần nạp trang hoặc đổi trang.

d. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <4,1>

$N_p=4 < 5$ □ Thỏa mãn. Từ dòng 4 của PCB có:

- $d=1 < 4$ □ Thỏa mãn □ Địa chỉ logic hợp lệ

- $P=0$ □ Trang 4 không nạp vào RAM

□ Hệ thống cần nạp trang hoặc đổi trang

e. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <3,5>

- Số hiệu trang logic cần truy nhập là: $N_p = 3 < 5$ (số trang logic của chương trình) . nên số hiệu trang cần truy nhập là hợp lệ.

- Địa chỉ lệch cần truy nhập trong trang là: $d = 5 \geq 4$ (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là không hợp lệ. Do vậy tiến trình phải dừng.

f. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <4,4>

$N_p=4 < 5$ □ Thỏa mãn. Từ dòng 4 của PCB có:

- $d=4 < 4$ □ Không thỏa mãn □ Địa chỉ logic không hợp lệ

g. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <5,3>

$N_p=5 < 5$ □ Không thỏa mãn □ Số hiệu trang truy nhập không hợp lệ

Câu 4.27.

Giả sử bộ nhớ vật lý (BNVL) có dung lượng 64MB, chương trình gồm 4 modul, xác định địa chỉ truy cập trong chiến lược phân trang – phân đoạn, với:

SCB			PCB ₀		PCB ₂		PCB ₃	
Ps	Ap	Ls	P	A	P	A	P	A
1	400400h	4	0		1	407h	1	40Ah
0	-	4	1	405h	0	-	1	40Fh
1	400C00h	3	1	406h	1	408h	0	-
1	401000h	4	0	-			1	40Bh

Biết kích thước 1 trang là 4KB, địa chỉ đầu của bảng phân đoạn Rs = 400000h.

Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic sau:

<0, 2, 20Bh>; <0, 1, 7Ah>; <1, 0, 46h>; <1, 1, 1001h>;
<2, 1, 1000h>; <2, 3, 200h>; <4, 1, 57h>

* Phần tính toán chung

- Kích thước một trang vật lý là:

$$L = 4 \text{ KB} = 2^2 * 2^{10} \text{ bytes} = 2^{12} \text{ bytes}$$

$$= 1000000000000B \text{ bytes} = 1000H \text{ bytes}$$

- Vậy số bit để đánh độ lệch trong trang là: $n_1 = 12$.

- Số trang vật lý = dung lượng BNVL / kích thước 1 trang

$$= 64MB/4KB = 64 * 2^{10} \text{ KB} / 4 \text{ KB} = 16 * 2^{10} = 2^4 * 2^{10} = 2^{14}$$

$$= 100000000000000B = 4000H$$

- Số bit để đánh số hiệu trang vật lý là: $n_2 = 14$.

- Số bit để đánh địa chỉ vật lý là: $64 \text{ MB} = 2^{26}$, vậy $n = 26$ ($n = n_2 + n_1$).

a. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <0, 2, 20Bh>

- Số hiệu đoạn cần truy nhập là: $S = 0 < 4$ (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.

- Số hiệu trang logic cần truy nhập là: $N_p = 2 < L_0 = 4$ (độ dài đoạn 0 – xem phần tử 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.

- Địa chỉ lệch cần truy nhập trong trang là: $d = 20BH < L = 1000H$, địa chỉ lệch cần truy nhập trong trang là hợp lệ.

- Phần tử 0 trong SCB có $P_s = 1$, cho biết đoạn 0 đã được nạp và $A_p = 400400h$ là địa chỉ đầu của bảng PCB₀ trong bộ nhớ vật lý.

- Truy cập bảng PCB₀, ta thấy phần tử 2 (quản lý trang $N_p = 2$) cho biết trang $N_p = 2$ đã được nạp ($P = 1$) vào trang vật lý có số hiệu $A = 406H$.

- Vậy địa chỉ vật lý cần truy nhập là:

$$\text{ĐCVL} = A \text{ SHL } n_1 \text{ OR } d = 406H \text{ SHL } 12 \text{ OR } 20BH$$

$$= 406000H \text{ OR } 20BH = 40620BH.$$

b. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <0, 1, 7Ah>

$S=0 < 4$ □ Thỏa mãn. Từ mục 0 của bảng PCB có:

- $P_s=1$ □ Đoạn 0 đã nạp
 - $A_p=400400h$ □ Bảng PCB_0 đã bắt đầu tại địa chỉ $400400h$
 - $L_p=4$ □ Đoạn 0 có kích thước 4 trang
 $N_p=1 < 4$ □ Thỏa mãn. Từ mục vào 1 của bảng PCB_0 có:
 - $P=1$ □ Trang 1 của của module 1 đã nạp
 - $A=405h$ □ Trang 1 nạp vào trang vật lý $405h < 4000h$ □ Thỏa mãn
 $d=7Ah < L=1000h$ □ Thỏa mãn
 □ Địa chỉ vật lý cần truy nhập là:
 $ĐCVL=A*n_1 + d=405h*12+7Ah=405000h + 7Ah=40507Ah$

c. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <1, 0, 46h>

- Số hiệu đoạn cần truy nhập là: $S = 1 < 4$ (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
- Số hiệu trang logic cần truy nhập là: $N_p = 0 < L_1 = 4$ (độ dài đoạn 0 – xem phần từ 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là: $d = 46H < L = 1000H$, địa chỉ lệch cần truy nhập trong trang là hợp lệ.
- Phần từ 1 trong SCB có $P_s = 0$, cho biết đoạn 1 chưa được nạp vào bộ nhớ vật lý. Hệ thống cần nạp đoạn hoặc đổi đoạn.

d. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <1, 1, 1001h>

$S=1 < 4$ □ Thỏa mãn. Từ mục 1 của bảng PCB có:

- $P_s=0$ □ Đoạn 1 không được nạp □ Hệ thống cần nạp đoạn hoặc đổi đoạn

e. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <2, 1, 1000h>

- Số hiệu đoạn cần truy nhập là: $S = 2 < 4$ (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
- Số hiệu trang logic cần truy nhập là: $N_p = 1 < L_2 = 4$ (độ dài đoạn 0 – xem phần từ 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là: $d = 1000H \geq L = 1000H$, địa chỉ lệch cần truy nhập trong trang là không hợp lệ, do vậy tiến trình phải dừng.

f. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <2,3, 200h>

$S=2<4$ □ Thỏa mãn. Từ mục 2 của bảng PCB có:

- $P_s=1$ □ Đoạn 2 đã được nạp

- $A_p=400C00h$ □ Bảng PCB_0 bắt đầu tại địa chỉ $400C00h$

- $L_s=3$ □ Đoạn 2 có kích thước 3 trang

$N_p=3<4$ □ Thỏa mãn. Từ mục vào 3 của bảng PCB_0 có:

- $P=0$ □ Trang 3 của module 3 đã không được nạp

□ Hệ thống cần nạp trang hoặc đổi trang

g. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic $<4, 1, 57h>$

$S=4<4$ □ Không thỏa mãn

□ Địa chỉ đoạn truy nhập không hợp lệ

Câu 4.28.

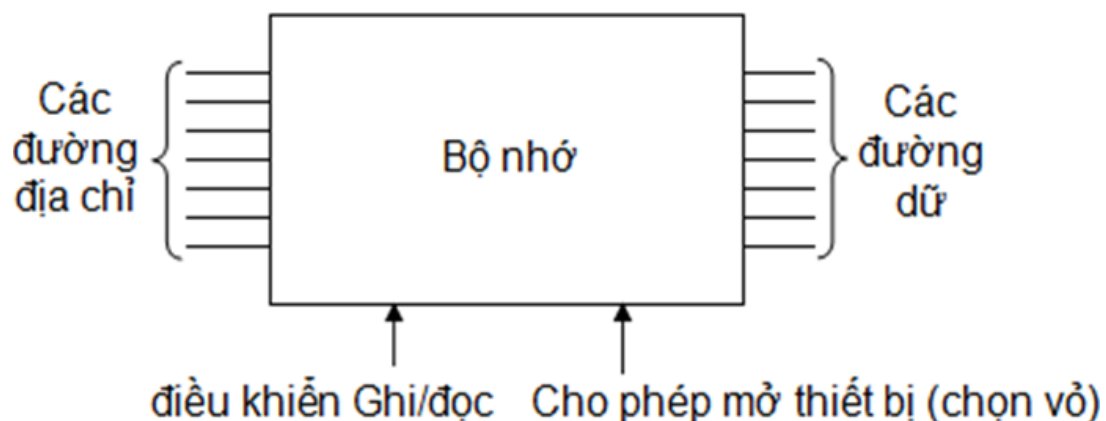
Vẽ và trình bày sơ đồ khối chung của vi mạch nhớ.

Gợi ý

Vi mạch nhớ nói chung, có 4 nhóm tín hiệu như sau:

- Nhóm tín hiệu vào: tín hiệu địa chỉ, tín hiệu chọn vở, tín hiệu điều khiển đọc/ghi.
- Nhóm tín hiệu vừa vào, vừa ra: tín hiệu dữ liệu.

Bài làm



Sơ đồ khối chung của vi mạch nhớ

Số đường dây địa chỉ là $n \rightarrow$ dung lượng bộ nhớ sẽ là 2^n

Câu 4.29.

Vẽ sơ đồ khối vi mạch nhớ 2K x 8 bits

Bài làm

Câu 4.30.

Vẽ và trình bày sơ đồ khối chung của vi mạch giải mã địa chỉ (GMĐC).

Gợi ý

Vi mạch GMĐC nói chung, có 3 nhóm tín hiệu như sau:

- **Nhóm tín hiệu vào: tín hiệu địa chỉ và tín hiệu chọn vỏ cho GMĐC.**
- **Nhóm tín hiệu ra: tín hiệu chọn vỏ cho vi mạch nhớ.**

Bài làm

Câu 4.31.

Vẽ sơ đồ khối và đưa ra bảng chân lý cho vi mạch GMĐC vào 3 ra 8.

Gợi ý

- Xem ví dụ sơ đồ khối mạch giải mã địa chỉ bộ nhớ vào 2 ra 4, mục 4.2.4., b) - Tài liệu Giáo trình kiến trúc máy tính, trường Đại học CNHN.

Bài làm

Câu 4.32.

Cho các vi mạch nhớ 1K x 4 (bit), hãy xây dựng vi mạch nhớ 1Kx8 bit.

Gợi ý

- **Đọc lại lý luận chung và ví dụ tại mục 4.2.4., c) – Tài liệu Giáo trình Kiến trúc máy tính, trường Đại học CNHN.**

Bài làm

Câu 4.33.

Trình bày phương pháp và các bước xây dựng module nhớ có số lượng gần nhớ mong muốn.

Bài làm

Câu 4.34.

Cho vi mạch nhớ 2Kx8 bit, xây dựng vi mạch nhớ 4Kx8 bit và phối ghép với CPU 8086, giải mã địa chỉ tự chọn.

Gợi ý

- Bài toán ngầm định yêu cầu phối ghép bộ nhớ với CPU bắt đầu tại địa chỉ 0.
- Xem ví dụ 1, mục 4.2.4., d) - Tài liệu Giáo trình Kiến trúc máy tính, Đại học CNHN.

Bài làm

Câu 4.35.

Thiết kế mạch ghép nối giữa 8086 và chip nhớ RAM 1K \times 8 bit để thành chip nhớ 6K \times 8 bit biết địa chỉ bắt đầu là ABC00h. Giải mã địa chỉ tự chọn.

Gợi ý

- Xem ví dụ 2, mục 4.2.4. d) - Tài liệu Giáo trình Kiến trúc máy tính, Đại học CNHN.

Bài làm

Câu 4.36.

Cho vi mạch nhớ $8K \times 8$ (bit), xây dựng phối ghép bộ nhớ $64K \times 8$ (bit) với CPU 8086 và tổ chức thành 4 bank nhớ.

Gợi ý

- Bài toán ngầm định yêu cầu phối ghép bộ nhớ với CPU bắt đầu tại địa chỉ 0.
- Xem ví dụ, mục 4.2.4., e) - Tài liệu Giáo trình Kiến trúc máy tính, Đại học CNHN.

Bài làm

Chương 5. Kỹ thuật đường ống và RISC

Câu 5.1.

Trình bày kỹ thuật đường ống (pipeline) để đưa ra công thức tính thời gian thực hiện n lệnh, với ví dụ một đường ống gồm 6 công đoạn: Nhận lệnh, giải mã lệnh, tính địa chỉ toán hạng, nhận toán hạng, thi hành lệnh, ghi toán hạng.

Bài làm

- Kỹ thuật đường ống gồm 6 công đoạn:
 - + Nhận lệnh (IF: Instruction Fetch).
 - + Giải mã lệnh (ID: Instruction decode).
 - + Tính địa chỉ toán hạng (Operand calculation - OC).
 - + Nhận toán hạng (Operands Fetch – OF).
 - + Thực hiện lệnh (Instruction execution – IE).
 - + Ghi kết quả (WO – Write operation).
- Công thức tính thời gian thực hiện lệnh n:

$$T_s = n * m * T_c$$

$$T_p = m * T_c + (n - 1) * T_c$$

Trong đó: m là số công đoạn

n là số câu lệnh

T_c là thời gian thực hiện 1 công đoạn

T_p : thời gian thực hiện n câu lệnh

Câu 5.2.

Trình bày các khó khăn và hướng khắc phục với kỹ thuật đường ống đơn vị lệnh

Bài làm

Câu 5.3.

Giả sử một lệnh được chia làm 5 công đoạn: nhận lệnh, giải mã lệnh, nhận toán hạng, xử lý, cất kết quả. Mỗi công đoạn thực hiện trong $5 \cdot 10^{-9}$ giây. Một lệnh nếu thực hiện tuần tự hết $22 \cdot 10^{-9}$ giây. So sánh thời gian thực hiện 40 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.

Gợi ý

- Áp dụng công thức $T_p = m \cdot T_c + (n-1) \cdot T_c$, trong đó:

T_p : thời gian thực hiện n lệnh theo kỹ thuật pipeline.

T_c : thời gian thực hiện một công đoạn lớn nhất trong m công đoạn theo kỹ thuật pipeline.

n : số lệnh.

M: số công đoạn trong một lệnh.

- Và công thức $T_t = n * T$, trong đó:

T: thời gian thực hiện 1 lệnh khi thực hiện tuần tự.

Bài làm

Ta có: $T_t = n * T = 40 * 22 * 10^{-9}$ (giây)

$$\begin{aligned} T_p &= m * T_c + (n-1) * T_c = 5 * (5 * 10^{-9}) + (40 - 1) * (5 * 10^{-9}) \\ &= 44 * (5 * 10^{-9}) \text{ (giây)} \end{aligned}$$

Vậy thực hiện 40 lệnh với kỹ thuật pipeline nhanh hơn 40 lệnh với kỹ thuật tuần tự là : $T_t/T_p = 40 * 22 * 10^{-9} / (44 * 5 * 10^{-9}) = 8 * 22 / 44 = 4$ lần.

Câu 5.4.

Giả sử một lệnh được chia làm 6 công đoạn: nhận lệnh, giải mã lệnh, tính địa chỉ toán hạng, nhận toán hạng, xử lý, cất kết quả. Mỗi công đoạn thực hiện trong thời gian $5 * 10^{-9}$ giây. Một lệnh nếu thực hiện tuần tự hết $22 * 10^{-9}$ giây. So sánh thời gian thực hiện 50 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.

Bài làm

Ta có: $T_t = n * T = 50 * 22 * 10^{-9}$ (giây)

$$\begin{aligned} T_p &= m * T_c + (n-1) * T_c = 6 * (5 * 10^{-9}) + (50 - 1) * (5 * 10^{-9}) \\ &= 55 * (5 * 10^{-9}) \text{ (giây)} \end{aligned}$$

Vậy thực hiện 50 lệnh với kỹ thuật pipeline nhanh hơn 40 lệnh với kỹ thuật tuần tự là : $T_t/T_p = 50 * 22 * 10^{-9} / (55 * 5 * 10^{-9}) = 4$ lần.

Câu 5.5.

Trình bày tóm tắt các đặc điểm trong kỹ thuật cài đặt RISC.

Bài làm

Đặc điểm trong kỹ thuật cài đặt RISC

- Kích thước lệnh cố định 4 bytes.
- Sử dụng lệnh load/store.
- Nhiều thanh ghi: 32 thanh ghi 32 bit cho số nguyên, 32 thanh ghi 64 bit dành cho số thực.
- Đánh địa chỉ đơn giản.
- Tập lệnh nhỏ.
- Hỗ trợ ít kiểu dữ liệu.
- Các lệnh thực hiện nhanh chỉ cần 1 xung clock.

- Có bus riêng cho dữ liệu và mã lệnh

Câu 5.6.

Cho biết ý nghĩa của cụm từ RISC (Reduced Instruction Set Computer)

Bài làm

RISC là viết tắt của Bộ hướng dẫn giảm. Nó được thiết kế để giảm thời gian thực hiện bằng cách đơn giản hóa tập lệnh. Nó sử dụng các hướng dẫn tối ưu hóa cao. Bộ xử lý dựa trên RISC thường được sử dụng cho các thiết bị di động như điện thoại di động và máy tính bảng vì chúng hiệu quả hơn.

5. Trình bày ngắn gọn sự khác biệt giữa kiến trúc CISC và kiến trúc RISC.

Bài làm

	RISC	CISC
Viết tắt	Reduced Instruction Set Computer	Complex Instruction Set Computer
Định nghĩa	Bộ xử lý RISC có một bộ hướng dẫn nhỏ hơn với một vài nút địa chỉ	Bộ xử lý CISC có một bộ hướng dẫn lớn hơn với nhiều nút địa chỉ.
Đơn vị bộ nhớ	Nó không có đơn vị bộ nhớ và sử dụng một phần cứng riêng để thực hiện các hướng dẫn.	Nó có một đơn vị bộ nhớ để thực hiện các hướng dẫn phức tạp.
Chương trình	Nó có một đơn vị lập trình cứng.	Nó có một đơn vị lập trình vi mô.
Thiết kế	Nó là một thiết kế phức tạp	Nó là một thiết kế dễ dàng hơn
Tính toán	Các tính toán nhanh hơn và chính xác.	Các tính toán là chậm và chính xác
Giải mã	Giải mã hướng dẫn rất đơn giản.	Giải mã các hướng dẫn là phức tạp.
Thời gian	Thời gian thực hiện là rất ít.	Thời gian thực hiện rất cao
Bộ nhớ ngoài	Nó không yêu cầu bộ nhớ ngoài để tính toán.	Nó đòi hỏi bộ nhớ ngoài để tính toán

Đường ống	Pipelining không hoạt động chính xác.	Đường ống không hoạt động chính xác.
Ngừng	Sự đình trệ chủ yếu là giảm trong bộ xử lý.	Các bộ xử lý thường bị đình trệ.
Mở rộng mã	Mở rộng mã có thể là một vấn đề.	Mở rộng mã không phải là một vấn đề.
Không gian đĩa	Không gian được lưu.	Không gian bị lãng phí
Các ứng dụng	Được sử dụng trong các ứng dụng cao cấp như xử lý video, viễn thông và xử lý hình ảnh.	Được sử dụng trong các ứng dụng cấp thấp như hệ thống bảo mật, tự động hóa gia đình...

Mục lục

Chương 1. Tổng quan về kiến trúc máy tính

Chương 2. Kiến trúc tập lệnh

1.1. Tóm tắt lý thuyết cơ bản.

1.1.1. Thông tin - biểu diễn và xử lý thông tin

1.1.1.1. Thông tin

1.1.1.2. Dữ liệu

1.1.1.3. Biểu diễn thông tin và xử lý dữ liệu trong máy tính

1.1.1.4. Đơn vị thông tin

1.1.1.5. Một số đơn vị đo độ dài của số hệ 2 dẫn xuất từ bit.

1.1.2. Hệ đếm và các hệ cơ số

1.2. Câu hỏi và bài tập chương 2

Chương 3. CPU, Đường truyền và hệ thống vào/ra

Chương 4. Kiến trúc hệ thống nhớ

Chương 5. Kỹ thuật đường ống và RISC