**1.Đổi các số nguyên thập phân sau ra số hex 16 bit: - 234 (biểu diễn số -234 ở dạng 16 bit rồi chuyển 16 bit này sang hệ hex)**

Ta có: +234 = 0000000011101010B

Số bù 1( của +234) = 1111111100010101B

+ 1

Số bù 2( của +234) = 1111111100010110B

Vậy -234 = 1111111100010110B =FF16H

**2.Đổi các số nguyên thập phân sau ra số hex 16 bit: –16**

Ta có : +16= 0000000000010000B

Số bù 1( của +16) =1111111111101111B

+ 1

Số bù 2( của +16) = 1111111111110000B

Vậy -16 = 1111111111110000B =FFF0H

**3.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit: –0.125**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

* Đổi số 0.125 về hệ 2:

0.125 x 2 = 0.25

0.25 x 2 = 0.5

0.5 x 2 = 1

=>0.125 = 0.001B

Chuẩn hóa: 0.125 =0.001B

= (-1)1 x 1.00000000000000000000000 B x 2-3

E – Bias = -3 ==> E = -3 + Bias = -3 + 127 = 124 = 01111100B

Vậy:

S = 1

E = 01111100B

M = 00000000000000000000000B

🡺-0.125 = 10111110000000000000000000000000 B = BE000000H

**4.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit: –0.75**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 0.75 về hệ 2:

0.75x2=1.5

0.5x2=1.0

🡺0. 75= 0.11B+++

Chuẩn hóa: -0.75=0.11B

= (-1)^1x1.10000000000000000000000Bx2-1

E-Bias= -1🡺E= -1+Bias=126=26+25+24+23+22+21=01111110B

Vậy:

S=1

E=01111110B

M=10000000000000000000000B

🡺-0.75=10111111010000000000000000000000B=BF400000H

**5.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit:**

**–154.25**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 154.25 về hệ 2:

+Đổi 154 về hệ 2: 154=2^7+2^4+2^3+2^1=10011010B

+Đổi 0.25 về hệ 2:

0.25x2=0.5

0.5x2=1

🡺 0.25=0.01B

🡺 154.25= 10011010.01B

Chuẩn hóa: -154.25= 10011010.01B

=(-1)^1x1.00110100100000000000000Bx2^7

E-Bias=7🡺E=7+127=134=2^7+2^2+2^1=10000110B

Vậy:

S=1

E=10000110B

M=00110100100000000000000B

🡺-154.25=11000011000110100100000000000000B=C31A4000H

**6.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit: +76.75**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 76.75 về hệ 2:

+Đổi 76 về hệ 2: 76=2^6+2^3+2^2=01001100B

+ Đổi 0.75 về hệ 2:

0.75x2=1.5

0.5x2=1.0

🡺0. 75= 0.11B

🡺 76.75= 01001100.11B

Chuẩn hóa: 76.75= 01001100.11B

=(-1)^0x1.00110011000000000000000Bx2^6

E-Bias=6🡺E=6+127=133=2^7+2^2+2^0=10000101B

Vậy:

S=0

E=10000101B

M=00110011000000000000000B

🡺+76.75=01000010100110011000000000000000B=42998000H

**7.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit:**

**+1022.0625**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 1022.0625 về hệ 2:

+Đổi 1022 về hệ 2: 1022=2^9+2^8+2^7+2^6+2^5+2^4+2^3+2^2+2^1

=1111111110B

+ Đổi 0.0625 về hệ 2:

0.0625x2=0.125

0.125x2=0.25

0.25x2=0.5

0.5x2=1.0

🡺0.x0625= 0.0001B

🡺 1022.0625=1111111110.0001B

Chuẩn hóa: 1022.0625=1111111110.0001B

=(-1)^0x1.11111111000010000000000B x2^9

E-Bias=9🡺E=9+127=136=2^7+2^3=10001000B

Vậy:

S=0

E=10001000B

M=11111111000010000000000B

🡺+1022.0625=01000100011111111000010000000000B=447F8400H

**8.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit: 1032.0625**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 1032.0625 về hệ 2:

+Đổi 1032 về hệ 2: 1032=2^10+2^3=10000001000B

+ Đổi 0.625 về hệ 2:

0.625x2=0.125

0.125x2=0.25

0.25x2=0.5

0.5x2=1.0

🡺0. 625= 0.0001B

🡺 1032.0625=10000001000.0001B

Chuẩn hóa: 1032.0625=10000001000.0001B

=(-1)^0x1.00000010000001000000000Bx2^10

E-Bias=10🡺E=10+127=137=2^7+2^3+2^0=10001001B

Vậy:

S=0

E=10001001B

M=00000010000001000000000B

🡺+1032.0625=01000100100000010000001000000000B=44810200H

**9.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit:**

**–0.03125**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 0.03125 về hệ 2:

0.03125x2=0.0625

0.0625x2=0.125

0.125x2=0.25

0.25x2=0.5

0.5x2=1.0

🡺0.03125=0.00001B

Chuẩn hóa : 0.03125=0.00001B=(-1)^1x1.00000000000000000000000Bx2^-5

E-Bias=-5🡺E=-5+127=122=2^6+2^5+2^4+2^3+2^1=01111010B

Vậy:

S=1

E=01111010B

M=00000000000000000000000B

🡺-0.03125=10111101000000000000000000000000B=BD000000H

**10.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit: +129.9**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 129.9 về hệ 2:

+Đổi 129 về hệ 2: 129=2^7+2^0=10000001B

+ Đổi 0.9 về hệ 2:

0.9x2=1.8

0.8x2=1.6

0.6x2=1.2

0.2x2=0.4

0.4x2=0.8

🡺0. 9= 0.11100110011001100…….B

🡺 129.9=10000001.11100110011001100…….B

Chuẩn hóa: 129.9=10000001.11100110011001100…….B

=(-1)^0x1.00000011110011001100110Bx2^7

E-Bias=7🡺E=7+124=134=2^7+2^2+2^1=10000110B

Vậy:

S=0

E=10000110B

M=00000011110011001100110B

🡺+129.9=01000011000000011110011001100110B=4301E666H

**11.Biểu diễn số sau ở dạng dấu chấm động trong máy tính theo chuẩn IEEE 32 bit:**

**–129.8**

Ta có CT: R=(-1)^S\*1.M\*2^(E-Bias)

Bias= 2^số bit của E/2-1

Khuôn dạng: SEM

-Đổi 129.8 về hệ 2:

+Đổi 129 về hệ 2: 129=2^7+2^0=10000001B

+ Đổi 0.8 về hệ 2:

0.8x2=1.6

0.6x2=1.2

0.2x2=0.4

0.4x2=0.8

0.8x2=1.6

🡺0. 8= 0.1100110011001100……..B

🡺 129.8=10000001.1100110011001100……..B

Chuẩn hóa: 129.8=10000001.1100110011001100……..B

=(-1)^1x1.00000011100110011001100Bx2^7

E-Bias=7🡺E=7+124=134=2^7+2^2+2^1=10000110B

Vậy:

S=1

E=10000110B

M=00000011100110011001100B

🡺-129.8=11000011000000011100110011001100B=C301CCCCH

**12.a) Cho biết ý nghĩa khi nói Bus địa chỉ có độ rộng 24 bit.**

**b) Trình bày sơ đồ khối chung của hệ thống vào/ra trong máy tính.**

Giải:

a, Bus gồm 24 đường dây dẫn, CPU có khả năng quản lý không gian nhớ là 2^24 = 16MB

b, CPU thực hiện trao đổi thông tin với các thiết bị ngoại vi thông qua các thiết bị giao diện (các khối ghép nối) như sau:

Thiết bị giao diện là loại thiết bị khả trình (các mạch điều khiển, các mạch có thể lập trình) mỗi loại thiết bị giao diện đều có 3 loại thanh ghi, mỗi thanh ghi thực hiện 1 chức năng khác nhau, đó là thanh ghi điều khiển CONTROL nhận và chứa các từ đk xác lập chế độ làm việc, thanh ghi trạng thái STATUS chứa thông tin trạng thái làm việc, thanh ghi dữ liệu DATA thực hiện chức năng bộ đệm nơi chung chuyển dữ liệu vào ra. Mỗi thanh ghi đều được gán 1 địa chỉ nhất định, địa chỉ thanh ghi dữ liệu được gọi là địa chỉ cơ sơ của công hay gọi tắt là đia chỉ cổng.

**13. a,Trình bày cấu trúc chung của modul vào/ra dữ liệu.**

**b, Cho biết ý nghĩa khi nói Bus dữ liệu có độ rộng 32 bit**

Giải:

a, 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 dữ thông tin trạng thái/điều khiển cho các cổng vào/ra
* Khối logic điều khiển: điều khiển mô-đun ra/vào

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

**14.Trình bày phương pháp vào ra dữ liệu theo định trình**

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

**15.Trình bày phương pháp vào ra dữ liệu kiểu thăm dò**

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 sãn sàng thì quay lại kiểm tra

+ Nếu sãn sàng thì chuyển sang trao đổi dữ liệu với mô-đun vào/ra

* Ưu điểm : quá trình trao đổi dữ liệu có độ tin cậy rất cao vì việc truyền nhận dữ liệu chỉ xảy ra khi hai bên truyền và nhận đều sẵn sàng.
* Nhược điểm : chiếm dụng nhiều thời gian CPU cho việc thăm dò nên hiệu quả hoạt động của hệ thống không cao

**16.Trình bày phương pháp vào/ra dữ liệu theo ngắt cứng.**

- KN ngắt cứng: là CPU phải tạm dừng tiến trình đang thực hiện để chuyển sang chế thực hiện tiến trình phục vụ ngắt khi có yêu cầu ngắt từ phần mạch bên ngoài gửi đến CPU.

- Phương pháp vào/ra theo ngắt cứng là phương pháp mà thiết bị vào/ra chủ động khởi tạo quá trình vào/ra dữ liệu nhờ hệ thông ngắt.

-Ưu điểm : Quá trình vào/ra dữ liệu được thực hiện ngay khi có yêu cầu, độ tin cậy cao. Hiệu suất CPU cao vì không mất thời gian thăm dò trạng thái làm việc của thiết bị vào/ra.

-Nhược điểm : phương pháp này vẫn phải thông qua CPU nên hiệu năng hoạt động toàn hệ thống chưa phải là cao nhất.

**17.Trình bày cấu trúc của hệ thống vào/ra theo ngắt cứng.**

Cấu trúc hệ thống vào/ra theo kiểu ngắt cứng:

* 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 IRQi ( i= 0÷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

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

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

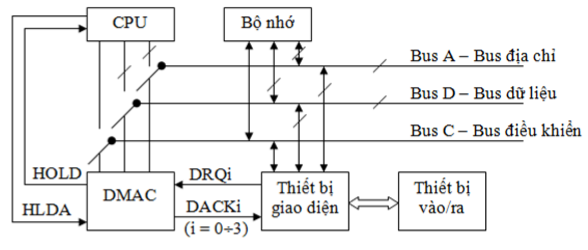
* Các thiết bị vào/ra cóyêu cầu phục vụ, phát ra tín hiệu IRQi ( i= 0÷7) tới PIC. Mỗi thiết bị vào/ra đã được ấn định sẵn một số hiệu ngắt nhất định.
* Thiết bị PIC lựu chọn số hiệu ngắt có mức ưu tiên cao nhất, phát tín hiệu INTR ( Interrupt Request) tới CPU yêu cầu CPU phục vụ.
* Nếu CPU chấp nhận 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 đẻ tiếp tục thi hành.

**19.Trình bày khái niệm quá trình DMA, cấu trúc của hệ thống vào/ra theo kiểu DMA**

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 thuết bị ngoại vi mà không thông qua CPU.

Cấu trúc hệ thống vào/ra theo kiểu DMA:



**20.Trình bày quá trình vào/ra dữ liệu kiểu DMA (quá trình DMA).**

Quá trình DMA:

* Thiết bị có nhu cầu trao đổi DMA sẽ gửi yêu cầu trao đổi đến chân DRQi 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 DACKi 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

**21.Cho mạch sau:**

A0

A11

A12

A21

A22

A31

A32

A41

A42

A51

A52

A61

A62

A7

D0

D1

D2

D3

D4

D5

D6

D7

O1

O2

O3

O4

O5

O6

C

S0

S1

S2

S3

S4

S5

S6

S7

Trong đó: D7 ÷ D0 : dữ liệu đầu vào C: bit điều khiển hướng dịch

S7 ÷ S0: dữ liệu đầu ra

Chứng minh khi bit C = 0, dữ liệu đầu ra là kết quả của dữ liệu đầu vào dịch phải tất cả các bit đi 1 bit

Ta phải CM:

S7=0

S6=D7

S5=D6

S4=D5

S3=D4

S2=D3

S1=D2

S0=D1

* Vào : D7D6D5D4D3D2D1D0🡺 ra 0D6D5D4D3D2D1D0

Giải: Ta có: C=0

S0=AND( ,D1)=AND(1,D1)=D1

S1=OR(A0,A21)=OR( AND(C,D0),AND( ,D2))= OR( AND(0,D0),AND( ,D2))=OR(0,D2)=D2

S2= OR(A12,A31)=OR( AND(C,D1),AND( ,D3))= OR( AND(0,D1),AND( ,D3))=OR(0,D3)=D3

S3= OR(A22,A41)=OR( AND(C,D2),AND( ,D4))= OR( AND(0,D2),AND( ,D4))=OR(0,D4)=D4

S4= OR(A32,A51)=OR( AND(C,D3),AND( ,D5))= OR( AND(0,D3),AND( ,D5))=OR(0,D5)=D5

S5= OR(A42,A61)=OR( AND(C,D4),AND( ,D6))= OR( AND(0,D4),AND( ,D6))=OR(0,D6)=D6

S6= OR(A52,A7)=OR( AND(C,D5),AND( ,D7))= OR( AND(0,D5),AND( ,D7))=OR(0,D7)=D7

S7=A62­=AND(C,D6)=AND(0,D6)=0

Vậy C=0 là dịch phải

**22.Cho mạch sau:**

A0

A11

A12

A21

A22

A31

A32

A41

A42

A51

A52

A61

A62

A7

D0

D1

D2

D3

D4

D5

D6

D7

O1

O2

O3

O4

O5

O6

C

S0

S1

S2

S3

S4

S5

S6

S7

Trong đó: D7 ÷ D0 : dữ liệu đầu vào C: bit điều khiển hướng dịch

S7 ÷ S0: dữ liệu đầu ra

Chứng minh khi bit C = 1, dữ liệu đầu ra là kết quả của dữ liệu đầu vào dịch trái tất cả các bit đi 1 bit

Giải:

Ta phải CM:

S7=D6

S6=D5

S5=D4

S4=D3

S3=D2

S2=D1

S1=D0

S0=0

* Vào : D7D6D5D4D3D2D1D0🡺 ra D6D5D4D3D2D1D00

Giải: Ta có: C=1

S0=AND( ,D1)=AND(0,D1)=0

S1=OR(A0,A21)=OR( AND(C,D0),AND( ,D2))= OR( AND(1,D0),AND( ,D2))=OR(D0,0)=D0

S2= OR(A12,A31)=OR( AND(C,D1),AND( ,D3))= OR( AND(1,D1),AND( ,D3))=OR(D1,0)=D1

S3= OR(A22,A41)=OR( AND(C,D2),AND( ,D4))= OR( AND(1,D2),AND( 0,D4))=OR(D2,0)=D2

S4= OR(A32,A51)=OR( AND(C,D3),AND( ,D5))= OR( AND(1,D3),AND( 0,D5))=OR(D3,0)=D3

S5= OR(A42,A61)=OR( AND(C,D4),AND( ,D6))= OR( AND(1,D4),AND( 0,D6))=OR(D4,0)=D4

S6= OR(A52,A7)=OR( AND(C,D5),AND( ,D7))= OR( AND(1,D5),AND(0,D7))=OR(D5,0)=D5

S7=A62­=AND(C,D6)=AND(1,D6)=D6

Vậy C=1 là dịch trái

**23.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 0, Bk = 0, Ck = 0 thì Zk = 0 và Ck+1 = 0**

Ta có: Ak = 0, Bk = 0, Ck = 0

+CM: Zk = 0

Zk = XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(0,0)=0

🡺 Zk = XOR(A,Ck)=XOR(0,0)=0(dpcm)

+CM: Ck+1 = 0

Ck+1 = OR(B,C)

mà B=AND(A,Ck)=AND(0,0)=0

mà C=AND(Ak,Bk)=AND(0,0)=0

🡺 Ck+1 = OR(B,C)=OR(0,0)=0(dpcm)

**24.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 0, Bk = 1, Ck = 0 thì Zk = 1 và Ck+1 = 0**

Ta có: Ak = 0, Bk = 1, Ck = 0

+CM:Zk=1

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(0,1)=1

🡺 Zk=XOR(A,Ck)=XOR(1,0)=1(dpcm)

+CM: Ck+1 = 0

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(0,1)=0

C=AND(Ak,Bk)=AND(0,1)=0

🡺 Ck+1 = OR(B,C)=OR(0,0)=0(dpcm)

**25.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 1, Bk = 0, Ck = 0 thì Zk = 1 và Ck+1 = 0**

Ta có: Ak = 1, Bk = 0, Ck = 0

+CM: Zk = 1

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(1,0)=1

🡺 Zk=XOR(A,Ck)=XOR(1,0)=1(dpcm)

+ Ck+1 = 0

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(0,1)=0

C=AND(Ak,Bk)=AND(1,0)=0

🡺 Ck+1 = OR(B,C)=OR(0,0)=0(dpcm)

**26.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 1, Bk = 1, Ck = 0 thì Zk = 0 và Ck+1 = 1**

Ta có : Ak = 1, Bk = 1, Ck = 0

+CM: Zk = 0

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(1,1)=0

🡺 Zk=XOR(A,Ck)=XOR(0,0)=0(dpcm)

+ Ck+1 = 1

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(0,0)=0

C=AND(Ak,Bk)=AND(1,1)=1

🡺 Ck+1 = OR(B,C)=OR(0,1)=1(dpcm)

**27.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 0, Bk = 0, Ck = 1 thì Zk = 1 và Ck+1 = 0**

Ta có : Ak = 0, Bk = 0, Ck = 1

+CM: Zk = 1

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(0,0)=0

🡺 Zk=XOR(A,Ck)=XOR(0,1)=1(dpcm)

+ Ck+1 = 1

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(1,0)=0

C=AND(Ak,Bk)=AND(0,0)=0

🡺 Ck+1 = OR(B,C)=OR(0,0)=0(dpcm)

**28.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 1, Bk = 0, Ck = 1 thì Zk = 0 và Ck+1 = 1**

Ta có : Ak = 1, Bk = 0, Ck = 1

+CM: Zk = 0

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(1,0)=1

🡺 Zk=XOR(A,Ck)=XOR(1,1)=0(dpcm)

+ Ck+1 = 1

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(1,1)=1

C=AND(Ak,Bk)=AND(1,0)=0

🡺 Ck+1 = OR(B,C)=OR(1,0)=1(dpcm)

**29.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 0, Bk = 1, Ck = 1 thì Zk = 0 và Ck+1 = 1**

Ta có : Ak = 0, Bk = 1, Ck = 1

+CM: Zk = 0

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(0,1)=1

🡺 Zk=XOR(A,Ck)=XOR(1,1)=0(dpcm)

+ Ck+1 = 1

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(1,1)=1

C=AND(Ak,Bk)=AND(0,1)=0

🡺 Ck+1 = OR(B,C)=OR(1,0)=1(dpcm)

**30.Cho mạch sau:**

A

Ck

Ak

Bk

Zk

Ck+1

B

C

Trong đó:

Ak, Bk: đầu vào

Zk: đầu ra

Ck: nhớ từ số thứ k–1 sang

Ck+1: nhớ tới số thứ k+1.

**Chứng minh : nếu Ak = 1, Bk = 1, Ck = 1 thì Zk = 1 và Ck+1 = 1**

Ta có : Ak = 1, Bk = 1, Ck = 1

+CM: Zk = 1

Zk=XOR(A,Ck)

mà A=XOR(Ak,Bk)=XOR(1,1)=0

🡺 Zk=XOR(A,Ck)=XOR(0,1)=1(dpcm)

+ Ck+1 = 1

Ck+1 = OR(B,C)

mà B=AND(Ck,A)=AND(1,0)=0

C=AND(Ak,Bk)=AND(1,1)=1

🡺 Ck+1 = OR(B,C)=OR(0,1)=1(dpcm)

**31.Cho sơ đồ mạch Flip-Flop sau, chứng minh rằng khi Cp = 0, trạng thái đầu ra Q không đổi**

Cp

D

Z1

Z3

Z4

Z2

Q

Q

E

F

D

B

A

C

Giải:

Xét: Q(t-1)=0 cần CM Q(t)=0

Z1(t)= NAND(Z3(t),Cp)=NAND(Z3(t),0)=1

Q(t-1)=0 thì  (t-1)=1

* Q(t)=NAND(Z1(t),(t-1))=NAND(1,1)=0 (1)

Xét: Q(t-1)=1 cần CM Q(t)=1

Q(t-1)=1 thì  (t-1)=0

* Q(t)=NAND(Z1(t),(t-1))=NAND(1,0)=1 (2)

🡺Từ (1) và (2)🡺 Q(t)=Q(t-1)==> Q không đổi.

**32.Cho sơ đồ mạch Flip-Flop sau, chứng minh rằng khi Cp = 1 (chuyển từ 0 → 1)**

**thì Q = D, biết khi Cp = 0 thì Z1 = 1 và Z2 = 1**

Cp

D

Z1

Z3

Z4

Z2

Q

Q

E

F

D

B

A

C

~~Cp = 0 thì Z1 = 1 và Z2 = 1🡺 Cp = 1 thì Z1 = 0 và Z2 = 0~~

+Cho D=0 CM Q=0:

Z4=NAND(D,Z2)=NAND(0,1)=1

Z2=NAND(Cp,Z1,Z4)=NAND(1,1,1)=0

Ta có: Z2=0

🡺 =NAND(Z2,Q)=NAND(0,Q)=1🡺Vậy Q=0 (1)

+Cho D=1 CM Q=1:

Z4=NAND(D,Z2)=NAND(1,1)=0

Z3=NAND(Z4,Z1)=NAND(0,1)=1

Z1=NAND(Cp,Z3)=NAND(1,1)=0

Ta có: Z1=0

🡺Q=NAND(Z1, )=NAND(0, )=1 (2)

Từ (1) và (2)🡺 Khi Cp=1 thì Q=D

**33.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 0, B2 = 0, B1 = 0, B0 = 0 hoặc B0 thì T = 0 (mã BCD không lỗi)**

(Thế nào là mã BCD?-Mã BCD sủ dụng 4 bit nhị phân để mã chứ số thập phân từ 0 đén 9.Vì dùng 4 bít dẫn đến ngoài các chữ số từ 0 đén 9 thì còn các giá trị A(1010),B(1011),C(1100),D(1101),E(1110),F(111).Các giá trị từ Ah đén Fh hay từ 10 đến 15 là các mã BCD lỗi.)

CM: T=0

Ta có B3=0,B2=0,B1=0,B0=0 hoặc B0=1 :

T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(0,OR(0,0))=AND(0,0)=0(dpcm)

**34.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 0, B2 = 0, B1 = 1, B0 = 0 hoặc B0 = 1 thì T = 0 (mã BCD không lỗi)**

Giải:

CM: T=0

Ta có B3 = 0, B2 = 0, B1 = 1, B0 = 0 hoặc B0 = 1

T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(0,OR(1,0))=AND(0,1)=0(dpcm)

**35.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 0, B2 = 1, B1 = 0, B0 = 0 hoặc B0 = 1 thì T = 0 (mã BCD không lỗi)**

Giải:

CM :T=0

Ta có: B3 = 0, B2 = 1, B1 = 0, B0 = 0 hoặc B0 = 1

T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(0,OR(0,1))=AND(0,1)=0(dpcm)

**36.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 1, B2 = 0, B1 = 0, B0 = 0 hoặc B0 = 1 thì T = 0 (mã BCD không lỗi)**

Giải:

CM :T=0

Ta có: B3 = 1, B2 = 0, B1 = 0, B0 = 0 hoặc B0 = 1

T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(1,OR(0,0))=AND(1,0)=0(dpcm)

**37.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 1, B2 = 0, B1 = 1, B0 = 0 hoặc B0 = 1, thì T = 1 (mã BCD lỗi)**

Giải:

CM :T=1

Ta có: B3 = 1, B2 = 0, B1 = 1, B0 = 0 hoặc B0 =1 T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(1,OR(1,0))=AND(1,1)=1(dpcm)

**38.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 1, B2 = 1, B1 = 0, B0 = 0 hoặc B0 = 1, thì T=1 (mã BCD lỗi)**

Giải:

CM :T=0

Ta có: B3 = 1, B2 = 1, B1 = 0, B0 = 0 hoặc B0=1 T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(1,OR(0,1))=AND(1,1)=1(dpcm)

**39.Mạch logic được thiết kế để phát hiện lỗi trong mã BCD. Lối vào là 3 bit cao của mã BCD, lối ra ở trạng thái 1 khi có lỗi.**

B2

B1

C

B3

T

**Chứng minh rằng B3 = 1, B2 = 1, B1 = 1, B0 = 0 hoặc B0 = 1 thì T = 1 (mã BCD lỗi)**

Giải:

CM :T=1

Ta có: B3 = 1, B2 = 1, B1 = 1, B0 = 0 hoặc B0 = 1

T=AND(B3,C)=AND(B3,OR(B1,B2))=AND(1,OR(1,1))=AND(1,1)=1(dpcm)

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

X

A

B

Y

Z

K

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=NOR(A,B)

Y=NOR(X,A)

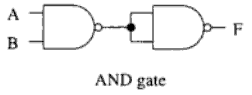
Z=NOR(B,Y)

K=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

**41. Chứng minh rằng mạch sau là cổng AND**



C

C1: Ta có: F=NAND(C,C)=NAND(NAND(A,B),NAND(A,B))=    = =A.B(dpcm)

C2:

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:

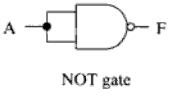
C=NAND(A,B)

F=NAND(C,C)

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | K |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

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

**42.Chứng minh rằng mạch sau là cổng NOT**



C1: Ta có: F=NAND(A,A)= = (dpcm)

C2:

Ta có bảng chân lý của mạch NOT

|  |  |
| --- | --- |
| A | F |
| 0 | 1 |
| 1 | 0 |

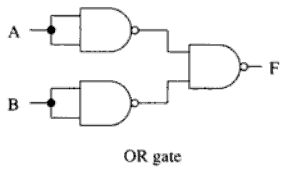
Ta thấy:

F=NAND(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

**43.Chứng minh rằng mạch sau là cổng OR**

****

C1:Ta có : F=NAND(NAND(A,A),NAND(B,B))

Mà NAND(A,A)=  =

NAND(B,B)= =

🡺 F=NAND(NAND(A,A),NAND(B,B))=NAND( , )= = =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=NAND(NAND(A,A),NAND(B,B))

Đặt X=NAND(A,A)

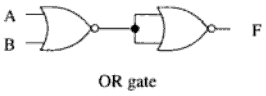
Y=NAND(B,B)

🡺F=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

**44.Chứng minh rằng mạch sau là cổng OR**

****

C1: Ta có: F= NOR(NOR(A,B),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= NOR(NOR(A,B),NOR(A,B))

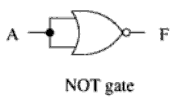
Đặt X=NOR(A,B)

🡺F=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

**45.Chứng minh rằng mạch sau là cổng NOT**

****

C1:Ta có: F=NOR(A,A)= = . =  (dpcm)

C2:

Ta có bảng chân lý của mạch NOT

|  |  |
| --- | --- |
| A | F |
| 0 | 1 |
| 1 | 0 |

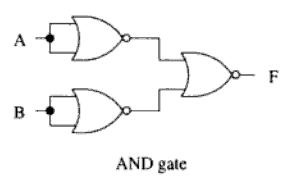
Ta thấy:

F=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

**46.Chứng minh rằng mạch sau là cổng AND**

****

C1:Ta có: F=NOR(NOR(A,A),NOR(B,B))

Mà NOR(A,A)= =. =

NOR(B,B)= =. =

🡺 F=NOR(NOR(A,A),NOR(B,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=NOR(NOR(A,A),NOR(B,B))

Đặt X=NOR(A,A)

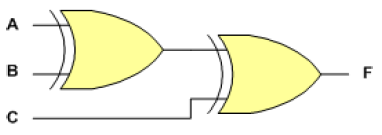
Y=NOR(B,B)

🡺F=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

**47.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**

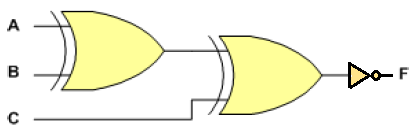


Ta có: bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0

F=XOR(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 |

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



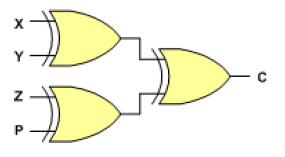
Ta có: bit 1 là lẻ thì đầu ra là 1, ngược lại đầu ra là 0

Gọi X= XOR(A,B); Y=XOR(X,C)🡺F=NOT(Y)

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

**49.Chứng minh rằng mạch sau là mạch lẻ 4 đầu vào (số bit 1 là lẻ thì đầu ra là 1),**

**với đầu vào là X, Y, Z, P ; đầu ra là C**

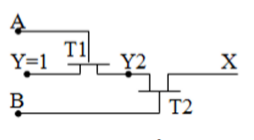


Ta có : số bit 1 là lẻ thì đầu ra là 1

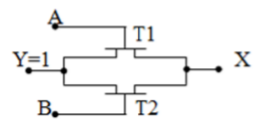
C=XOR(XOR(X,Y),XOR(Z,P))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 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 |

**50.Hãy vẽ sơ đồ mạch AND 2 đầu vào từ các Transistors**



**51.Hãy vẽ sơ đồ mạch OR 2 đầu vào từ các Transistors**



**52**.**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 cho trường hợp phát ra địa chỉ trên và địa chỉ ô nhớ cần truy cập trong block theo từng trường hợp.**

B1:Xác định dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+L+T=9+4+9= 22 bits

B5:Phân tích địa chỉ:

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

🡺Địa chỉ không hợp lệ

**53.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 BF0825h. 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.**

B1:Tính dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+L+T=9+4+9= 22 bits

B5:Phân tích địa chỉ:

BF0825H=101111110000100000100101B🡺Chiều dài địa chỉ là 24 bits 🡺 Không hợp lệ về chiều dài

🡺Địa chỉ không hợp lệ

**54.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.**

B1:Tính dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+L+T=9+4+9= 22 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 mod 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.

→ Địa chỉ vật lý = TW

→ Địa chỉ vật lý = 101111000000100101

= 2F025H

**55.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.**

B1:Tính dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+L+T=9+4+9= 22 bits

B5:Phân tích địa chỉ:

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

L=0100B=4

BI=101111000B=376

Block 376 phải nạp vào line: 376 mod 16 =8 🡺Block nạp không đúng line🡺 Địa chỉ không hợp lệ.

**56.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 cho trường hợp phát ra địa chỉ trên và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo từng trường hợp.**

B1:Tính dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+T=9+9= 18 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

**57.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 F0812h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết hoàn toàn cho trường hợp phát ra địa chỉ trên và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo từng trường hợp.**

B1:Tính dung lượng 1 block

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

B2:Xác định số line của cache

Nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)🡺L=4 bits

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

Nblock=512=29🡺T=9 bits

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

N=W+T=9+9= 18 bits

B5:Phân tích địa chỉ:

F0812H= 11110000100000010010B🡺Chiều dài địa chỉ là 20 bits 🡺 Không hợp lệ về chiều dài

🡺Địa chỉ không hợp lệ

**58.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 95418h, 132415h, 72426h. Hãy trình bày chi tiết phương pháp đọc cache cho trường hợp phát ra địa chỉ trên và địa chỉ vật lý của ô nhớ cần truy nhập tương ứng.**

**\*95418H**

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

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

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

nset =4=22🡺S=2 bits

+Tính số lines của cache:

nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)

+ Tính số line trên 1 set là:

nline/set = nline/nset=16/4=4(line/1set)

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

Nblock=512=29🡺T=9 bits

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

N=T+S+W=9+2+9= 20 bits

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

95418H= 10010101010000011000B🡺Chiều dài địa chỉ là 20 bits chiều dài tối đa địa chỉ hợp lệ🡺Địa chỉ hợp lệ về chiều dài(1)

SI=10B=2

BI=100101010B=298

Theo ánh xạ liên kết tập hợp thì block 298 phải nạp vào set sau:

298 mod 4=2🡺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 298 không?

Có 2 khả năng xảy ra:

* Không có trường tag nào bằng 298🡺 miss cache, Khi đó CPU phải nạp block nhớ có số hiệu 298 vào một line của Set 2 → như trường hợp 2.
* Có 1 tag của line nào đó trong set 2 bằng 298🡺 hit cache🡺CPU đọc ô nhớ có số hiệu( địa chỉ lệch) W=000011000B=24

Địa chỉ vật lý: TW=100101010000011000B =25418H

**\*132415H**

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

132415H= 100110010010000010101B🡺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

**\*72426H**

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

72426H= 1110010010000100110B🡺Chiều dài địa chỉ là 19 bits chiều dài tối đa địa chỉ hợp lệ🡺Địa chỉ hợp lệ về chiều dài

SI=10B=2

BI=11100100B=228

Theo ánh xạ liên kết tập hợp thì block 228 phải nạp vào set sau:

228 mod 4=0🡺Block không nạp đúng set🡺Địa chỉ CPU phát là không hợp lệ

**59.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 72426h. Hãy trình bày chi tiết phương pháp đọc cache cho trường hợp phát ra địa chỉ trên và địa chỉ vật lý của ô nhớ cần truy nhập tương ứng.**

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

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

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

nset =4=22🡺S=2 bits

+Tính số lines của cache:

nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)

+ Tính số line trên 1 set là:

nline/set = nline/nset=16/4=4(line/1set)

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

Nblock=512=29🡺T=9 bits

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

N=T+S+W=9+2+9= 20 bits

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

72426H=1110010010000100110B🡺Chiều dài địa chỉ là 19 bits chiều dài tối đa địa chỉ hợp lệ🡺Địa chỉ hợp lệ về chiều dài

SI=10B=2

BI=11100100B=228

Theo ánh xạ liên kết tập hợp thì block 228 phải nạp vào set sau:

228 mod 4=0🡺Block không nạp đúng set🡺Địa chỉ CPU phát là không hợp lệ

**60.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 132415h. Hãy trình bày chi tiết phương pháp đọc cache cho trường hợp phát ra địa chỉ trên và địa chỉ vật lý của ô nhớ cần truy nhập tương ứng.**

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

Cblock=Cmemory/Nblock=256KB/512=256x1024/512=0.5KB=512=29 🡺W=9 bits

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

nset =4=22🡺S=2 bits

+Tính số lines của cache:

nline=Ccache/Cblock=8KB/0.5KB=16=24(lines)

+ Tính số line trên 1 set là:

nline/set = nline/nset=16/4=4(line/1set)

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

Nblock=512=29🡺T=9 bits

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

N=T+S+W=9+2+9= 20 bits

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

132415H= 100110010010000010101B🡺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

**61.Cho phần tử nhớ sau, chứng minh khi:**

**- Yj = 1, Xi = 1, WE = 1 và CS = 1 thì Q = Din đồng thời H và Dout ngắt mạch.**

T

H

Q

Q

F

G

D

E

C

B

A

WE

Din

CS

Sij

Xi

Dout

Yj

**Giải:**

Ta có: Yj = 1, Xi = 1, WE = 1 và CS = 1🡺 CM Q=Din, H và Dout ngắt mạch

+WE=1🡺 =0

+C=AND(CS,)=AND(1,0)=0🡺H và Dout ngắt mạch

+B=NAND(CS,WE)=NAND(1,1)=0

+A=AND(Xi,Yj)=AND(1,1)=1

+E=NAND(A,B)=NAND(1,0)=1

Xét Din=1 CM Q=1:

+G=NAND(Din,E)=NAND(1,1)=0

+F=NAND(E,G)=NAND(1,0)=1

+Q=NAND( ,G)=NAND(,0)=1 (1)

Xét Din=0 CM Q=0:

+G=NAND(Din,E)=NAND(0,1)=1

+F=NAND(E,G)=NAND(1,1)=0

+ =NAND(F,Q)=NAND(0,Q)=1🡺Q=0 (2)

Từ (1) và (2) 🡺 Q=Din

**62.Cho phần tử nhớ sau, chứng minh khi:**

**- Yj = 1, Xi = 1, WE = 0 và CS = 1 thì Q không phụ thuộc Din và Dout = Q**

T

H

Q

Q

F

G

D

E

C

B

A

WE

Din

CS

Sij

Xi

Dout

Yj

Giải:

Ta có: Yj = 1, Xi = 1, WE = 0 và CS = 1 CM Q không phụ thuộc Din(Q không đổi), Dout = Q

+WE=0🡺 =1

+C=AND(CS,)=AND(1,1)=1

🡺Thiết bị 3 trạng thái K ở trạng thái thông, Vậy Dout =H

+B=NAND(CS,WE)=NAND(1,0)=1

+A=AND(Xi,Yj)=AND(1,1)=1

+E=NAND(A,B)=NAND(1,1)=0

+G=NAND(Din,E)=NAND(Din,0)=1

+F=NAND(E,G)=NAND(0,1)=1

Xét Q(t-1)=0

(t)=NAND(F,Q(t-1))=NAND(1,0)=1🡺Q(t)=0(1)

Xét Q(t-1)=1

(t)=NAND(F,Q(t-1))=NAND(1,1)=0🡺Q(t)=1(2)

Từ (1) và (2)🡺Q(t)=Q(t-1)🡺Q không đổi🡺 Q không phụ thuộc vào Din

Dout =H=NAND(A,)=NAND(1,)=Q

**63.Cho phần tử nhớ sau, chứng minh khi:**

**- Yj = 1, Xi = 1, WE = 0/1 và CS = 0 thì Q không phụ thuộc Din, đồng thời H và Dout ở trạng thái trở kháng cao (ngắt mạch).**

T

H

Q

Q

F

G

D

E

C

B

A

WE

Din

CS

Sij

Xi

Dout

Yj

Giải:

Ta có: Yj = 1, Xi = 1, WE = 0/1 và CS = 0 CM Q không phụ thuộc Din, H và Dout ở trạng thái trở kháng cao (ngắt mạch).

+WE=1🡺 =0

+C=AND(CS,)=AND(0,0)=0🡺H và Dout ở trạng thái trở kháng cao( ngắt mạch)

+B=NAND(CS,WE)=NAND(0,1)=1

+A=AND(Xi,Yj)=AND(1,1)=1

+E=NAND(A,B)=NAND(1,1)=0

+G=NAND(Din,E)=NAND(Din,0)=1

+F=NAND(E,G)=NAND(0,1)=1

Xét Q(t-1)=0

(t)=NAND(F,Q(t-1))=NAND(1,0)=1🡺Q(t)=0(1)

Xét Q(t-1)=1

(t)=NAND(F,Q(t-1))=NAND(1,1)=0🡺Q(t)=1(2)

Từ (1) và (2)🡺Q(t)=Q(t-1)🡺Q không đổi🡺 Q không phụ thuộc vào Din

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  | **PCB2** | |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |
| **1ư** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  | **1** | **40Ah** |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  | **1** | **40Bh** |  | **1** | **40Dh** |
| **1** | **400C00h** | **4** |  | **1** | **406h** |  | **0** | **-** |  | **0** | **-** |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  | **0** | **-** |  | **0** | **-** |

**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, 0, 2Bh>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=0<4🡪Thỏa mãn.Từ mục vào 0 của SCB có:

-P=1🡪Đoạn 0 đã được nạp

-A=400400h🡪Bảng PCB0 bắt đầu tại địa chỉ 400400h

-L=4🡪Đoạn 0 kích thước 4 trang

Np=0<L0=4🡪Thỏa mãn.Từ mục vào 0 của PCB0 có:

-Pp=1🡪Trang 0 đã được nạp

-Ap=405h🡪Trang 0 nạp vào trang vật lý 405h<4000h🡪Thỏa mãn

d=2Bh<L=1000h🡪Thỏa mãn

🡺Địa chỉ vật lý cần truy nhập là:

ĐCVL=Ap\*L + d=405h\*1000h+2Bh=405000h + 2Bh=40502Bh

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  | **PCB2** | |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  | **1** | **40Ah** |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  | **1** | **40Bh** |  | **1** | **40Dh** |
| **1** | **400C00h** | **4** |  | **1** | **406h** |  | **0** | **-** |  | **0** | **-** |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  | **0** | **-** |  | **0** | **-** |

**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, 1, 7Ah>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=0<4🡪Thỏa mãn.Từ mục vào 0 của SCB có:

-P=1🡪Đoạn 0 đã được nạp

-A=400400h🡪Bảng PCB0 bắt đầu tại địa chỉ 400400h

-L=4🡪Đoạn 0 kích thước 4 trang

Np=1<L0=4🡪Thỏa mãn.Từ mục vào 1 của PCB0 có:

-Pp=0🡪Trang 1 đã không được nạp

🡺Hệ thống cần nạp trang hoặc đổi trang

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  |  | **Pp** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  |  | **1** | **40Dh** |
| **0** | **-** | **4** |  | **1** | **406h** |  | **0** | **-** |  |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  |  | **0** | **-** |

**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: <1, 0, 46h>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=1<4🡪Thỏa mãn.Từ mục vào 1 của SCB có:

-P=1🡪Đoạn 1 đã được nạp

-A=400800h🡪Bảng PCB1 bắt đầu tại địa chỉ 400800h

-L=4🡪Đoạn 1 kích thước 4 trang

Np=0<L0=4🡪Thỏa mãn.Từ mục vào 0 của PCB1 có:

-Pp=1🡪Trang 0 đã được nạp

-Ap=407h🡪Trang 0 nạp vào trang vật lý 407h<4000h🡪Thỏa mãn

d=46h<L=1000h🡪Thỏa mãn

🡺Địa chỉ vật lý cần truy nhập là:

ĐCVL=Ap\*L + d=407h\*1000h+46h=407000h + 46h=407046h

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  |  | **Pp** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  |  | **1** | **40Dh** |
| **0** | **-** | **4** |  | **1** | **406h** |  | **0** | **-** |  |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  |  | **0** | **-** |

**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: <1, 1, 1001h>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=1<4🡪Thỏa mãn.Từ mục vào 1 của SCB có:

-P=1🡪Đoạn 1 đã được nạp

-A=400800h🡪Bảng PCB1 bắt đầu tại địa chỉ 400800h

-L=4🡪Đoạn 1 kích thước 4 trang

Np=1<L1=4🡪Thỏa mãn.Từ mục vào 1 của PCB1 có:

-Pp=1🡪Trang 1 đã được nạp

-Ap=408h🡪Trang 0 nạp vào trang vật lý 408h<4000h🡪Thỏa mãn

d=1001h<L=1000h🡪Không thỏa mãn

🡺Địa chỉ vật lý cần truy nhập không hợp lệ

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  | **1** | **40Dh** |
| **0** | **-** | **4** |  | **1** | **406h** |  | **0** | **-** |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  | **0** | **-** |

**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: <2, 1, 101h>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=2<4🡪Thỏa mãn.Từ mục vào 2 của SCB có:

-P=0🡪Đoạn 2 đã không được nạp

🡺Hệ thống cần nạp đoạn hoặc đổi đoạn

**Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic sau: <1, 0, 46h>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=1<4🡪Thỏa mãn.Từ mục vào 1 của SCB có:

-P=1🡪Đoạn 1 đã được nạp

-A=400800h🡪Bảng PCB1 bắt đầu tại địa chỉ 400800h

-L=4🡪Đoạn 1 kích thước 4 trang

Np=0<L1=4🡪Thỏa mãn.Từ mục vào 0 của PCB1 có:

-Pp=1🡪Trang 0 đã được nạp

-Ap=407h🡪Trang 0 nạp vào trang vật lý 407h<4000h🡪Thỏa mãn

d=46h<L=1000h🡪Thỏa mãn

🡺Địa chỉ vật lý cần truy nhập là:

ĐCVL=Ap\*L + d=407h\*1000h+46h=407000h + 46h=407046h

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **P** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  | **1** | **40Dh** |
| **0** | **-** | **4** |  | **1** | **406h** |  | **0** | **-** |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  | **0** | **-** |

**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: <3, 1, 2ABh>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=3<4🡪Thỏa mãn.Từ mục vào 3 của SCB có:

-P=1🡪Đoạn 3 đã được nạp

-A=401000h🡪Bảng PCB3 bắt đầu tại địa chỉ 401000h

-L=4🡪Đoạn 3 kích thước 4 trang

Np=1<L3=4🡪Thỏa mãn.Từ mục vào 1 của PCB3 có:

- Pp=1🡪Trang 1 đã được nạp

-Ap=40Dh🡪Trang 1 nạp vào trang vật lý 40Dh<4000h🡪Thỏa mãn

d=2ABh<L=1000h🡪Thỏa mãn

🡺Địa chỉ vật lý cần truy nhập là:

ĐCVL=Ap\*L + d=40Dh\*1000h+2ABh=40D000h + 2ABh=40DABh

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

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCB** | | |  | **PCB0** | |  | **PCB1** | |  | **PCB3** | |
| **P** | **A** | **L** |  | **Pp** | **Ap** |  | **Pp** | **Ap** |  | **P** | **Ap** |
| **1** | **400400h** | **4** |  | **1** | **405h** |  | **1** | **407h** |  | **1** | **40Ch** |
| **1** | **400800h** | **4** |  | **0** | **-** |  | **1** | **408h** |  | **1** | **40Dh** |
| **0** | **-** | **4** |  | **1** | **406h** |  | **0** | **-** |  | **1** | **40Fh** |
| **1** | **401000h** | **4** |  | **0** | **-** |  | **1** | **409h** |  | **0** | **-** |

**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, 1, 1001h>**

Kích thước một trang vật lý là:

L = 4 KB = 22\*210 bytes = 212 bytes

= 1000000000000B bytes = 1000H bytes

- Vậy số bit để đánh độ lệch trong trang là: n1 = 12.

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

= 64MB/4KB = 64\*210 KB/4 KB = 16 \* 210 = 24 \* 210 = 214

= 100000000000000B = 4000H

- Số bit để đánh số hiệu trang vật lý là: n2 = 14.

- Số bit để đánh địa chỉ vật lý là: 64 MB = 226, vậy n = 26 (n = n2 + n1).

S=0<4🡪Thỏa mãn.Từ mục vào 0 của SCB có:

-P=1🡪Đoạn 0 đã được nạp

-A=400400h🡪Bảng PCB0 bắt đầu tại địa chỉ 400400h

-L=4🡪Đoạn 0 kích thước 4 trang

Np=1<4🡪Thỏa mãn.Từ mục vào 1 của PCB0 có:

-Pp=0🡪Trang 1 đã không được nạp

🡺Hệ thống cần nạp trang hoặc đổi trang

**71.Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 16 bytes có nội dung ‘0123456789ABCDEF’. Và bảng quản lý trang (PCB) có nội dung sau:**

|  |  |  |
| --- | --- | --- |
| **P** | **A** | **- Xác định địa chỉ vật lý truy nhập tương ứng với các địa chỉ** |
| **1** | **7** | **logic sau: <2, 4>** |
| **0** | **-** | **- Cho biết nội dung các ô nhớ tương ứng với địa chỉ trên.** |
| **1** | **3** |  |
| **1** | **5** |  |

- Kích thước một trang vật lý là: L = dung lượng bộ nhớ vật lý / số trang vật lý.

= 32/8 = 4 bytes = 22.

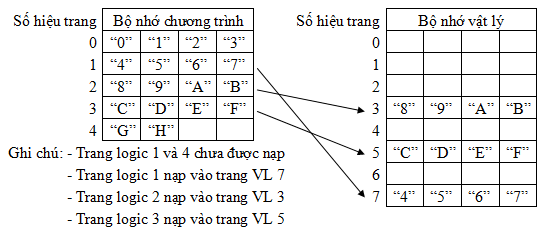
- Vậy số bit để đánh độ lệch trong trang là: n1 = 2.

- Số bit để đánh số hiệu trang vật lý là: 8 = 23, vậy n2 = 3.

- Số bit để đánh địa chỉ vật lý là: 32 = 25, vậy n = 5 (n = n2 + n1).

- Số trang logic của chương trình = dung lượng chương trình/ L = 16/4 = 4. Vậy số trang logic của chương trình phải là 4. (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:



Np=2<4🡪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=4<4🡪Không thỏa mãn

🡺Địa chỉ logic không hợp lệ

**72.Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 16 bytes có nội dung ‘0123456789ABCDEF’. Và bảng quản lý trang (PCB) có nội dung sau:**

|  |  |  |
| --- | --- | --- |
| **P** | **A** | **- Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ** |
| **1** | **7** | **logic sau: <2,3>** |
| **0** | **-** | **- Cho biết nội dung các ô nhớ tương ứng với địa chỉ trên.** |
| **1** | **3** |  |
| **1** | **5** |  |

- Kích thước một trang vật lý là: L = dung lượng bộ nhớ vật lý / số trang vật lý.

= 32/8 = 4 bytes = 22.

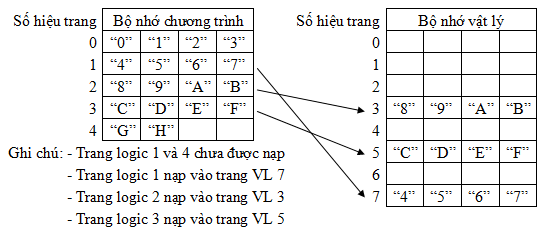
- Vậy số bit để đánh độ lệch trong trang là: n1 = 2.

- Số bit để đánh số hiệu trang vật lý là: 8 = 23, vậy n2 = 3.

- Số bit để đánh địa chỉ vật lý là: 32 = 25, vậy n = 5 (n = n2 + n1).

- Số trang logic của chương trình = dung lượng chương trình/ L = 16/4 = 4,. Vậy số trang logic của chương trình phải là 4. (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:



Np=2<4🡪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 n1=2 bít OR d=3

001100b

+ 0011b

--------------

001111b=15

|  |
| --- |
| 8 |
| 9 |
| A |
| B |

12

13 🡺Nội dung: “B”

14

15

**73.Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 16 bytes có nội dung ‘0123456789ABCDEF’. Và bảng quản lý trang (PCB) có nội dung sau:**

|  |  |  |
| --- | --- | --- |
| **P** | **A** | **- Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ** |
| **1** | **7** | **logic sau: <1, 2>** |
| **0** | **-** | **- Cho biết nội dung các ô nhớ tương ứng với địa chỉ trên.** |
| **1** | **3** |  |
| **1** | **5** |  |

- Kích thước một trang vật lý là: L = dung lượng bộ nhớ vật lý / số trang vật lý.

= 32/8 = 4 bytes = 22.

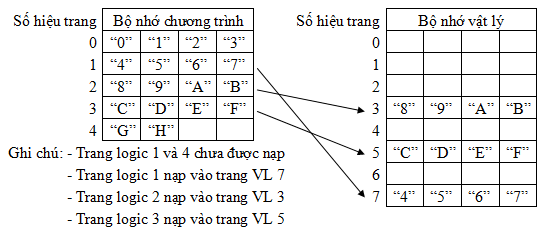
- Vậy số bit để đánh độ lệch trong trang là: n1 = 2.

- Số bit để đánh số hiệu trang vật lý là: 8 = 23, vậy n2 = 3.

- Số bit để đánh địa chỉ vật lý là: 32 = 25, vậy n = 5 (n = n2 + n1).

- Số trang logic của chương trình = dung lượng chương trình/ L = 16/4 = 4. Vậy số trang logic của chương trình phải là 4. (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:



Np=1<4 🡪Thỏa mãn.Từ dòng 1 của PCB có:

-P=0🡪Trang 1 đã không nạp vào RAM

🡺Địa chỉ logic không hợp lệ

**74.Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 16 bytes có nội dung ‘0123456789ABCDEF’. Và bảng quản lý trang (PCB) có nội dung sau:**

|  |  |  |
| --- | --- | --- |
| **P** | **A** | **- Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ** |
| **1** | **7** | **logic sau: <0, 3>** |
| **0** | **-** | **- Cho biết nội dung các ô nhớ tương ứng với địa chỉ trên.** |
| **1** | **3** |  |
| **1** | **5** |  |

- Kích thước một trang vật lý là: L = dung lượng bộ nhớ vật lý / số trang vật lý.

= 32/8 = 4 bytes = 22.

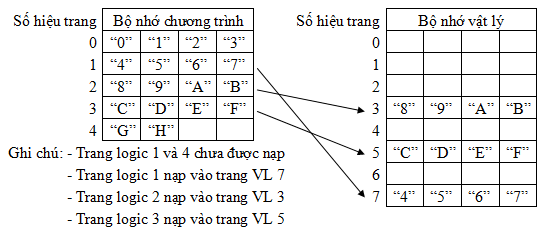
- Vậy số bit để đánh độ lệch trong trang là: n1 = 2.

- Số bit để đánh số hiệu trang vật lý là: 8 = 23, vậy n2 = 3.

- Số bit để đánh địa chỉ vật lý là: 32 = 25, vậy n = 5 (n = n2 + n1).

- Số trang logic của chương trình = dung lượng chương trình/ L = 16/4 = 4. Vậy số trang logic của chương trình phải là 4. (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:



Np=0<4 🡪Thỏa mãn.Từ dòng 0 của PCB có:

-P=1🡪Trang 0 đã nạp vào RAM

-A=7🡪Trang 0 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=7\*4+3=31

Hoặc

Dịch trái A=7 đi n1=2 bít OR d=3

011100b

+ 0011b

--------------

011111b=31

|  |
| --- |
| 0 |
| 1 |
| 2 |
| 3 |

28

29 🡺Nội dung: “3”

30

31

**75.Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý phát ra tương ứng với địa chỉ logic sau <2, 43h>, 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** |

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=43h<L=700h🡺Thỏa mãn

🡺Địa chỉ vật lý=A+d=300h+43h=343h

**76.Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý phát ra tương ứng với địa chỉ logic sau <3, 4FFh>, 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** |

S=3<5🡪Thỏa mãn🡪Lấy thông tin mục vào 3(dòng 3, dòng đánh số từ 0)

-Có P=0🡪Module 3 đã không được nạp vào RAM

🡪Hệ thống cần tiến hành nạp đoạn hoặc đổi đoạn

**77.Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý phát ra tương ứng với địa chỉ logic sau <4, 600h>, 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** |

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=600h<L=600h🡺Không thỏa mãn

🡺Địa chỉ không hợp lệ

**78.Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý phát ra tương ứng với địa chỉ logic sau <5, 100h>, 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** |

S=5<5🡪Không thỏa mãn

🡪Địa chỉ không hợp lệ

**79.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\*10-9 giây. Một lệnh nếu thực hiện tuần tự hết 22\*10-9 giây. So sánh thời gian thực hiện 30 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.**

Ta có: Tt = n \* T = 30 \* 22 \*10-9 (giây)

Tp = m\*Tc + (n-1)\*Tc = 5 \* (5\*10-9) + (30 -1)\* (5\*10-9)

= 34 \* (5\*10-9) (giây)

Vậy thực hiện 30 lệnh với kỹ thuật pipeline nhanh hơn 30 lệnh với kỹ thuật tuần tự là : Tt/Tp = 30\*22\*10-9 / (34\*5\*10-9) = 66/17 lần.

**80.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\*10-9 giây. Một lệnh nếu thực hiện tuần tự hết 22\*10-9 giây.So sánh thời gian thực hiện 40 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.**

Ta có: Tt = n \* T = 40 \* 22 \*10-9 (giây)

Tp = m\*Tc + (n-1)\*Tc = 5 \* (5\*10-9) + (40 -1)\* (5\*10-9)

= 44 \* (5\*10-9) (giây)

Vậy thực hiện 40 lệnh với kỹ thuật pipeline nhanh hơn 40 lệnh với kỹ thuật tuần tự là : Tt/Tp = 40\*22\*10-9 / (44\*5\*10-9) = 8\*22/44 = 4 lần.

**81.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\*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.**

Ta có: Tt = n \* T = 50 \* 22 \*10-9 (giây)

Tp = m\*Tc + (n-1)\*Tc = 5 \* (5\*10-9) + (50 -1)\* (5\*10-9)

= 54 \* (5\*10-9) (giây)

Vậy thực hiện 50 lệnh với kỹ thuật pipeline nhanh hơn 50 lệnh với kỹ thuật tuần tự là : Tt/Tp = 50\*22\*10-9 / (54\*5\*10-9) = 110/27 lần.

**82.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\*10-9 giây. Một lệnh nếu thực hiện tuần tự hết 22\*10-9 giây. So sánh thời gian thực hiện 20 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.**

Ta có: Tt = n \* T = 20 \* 22 \*10-9 (giây)

Tp = m\*Tc + (n-1)\*Tc = 5 \* (5\*10-9) + (20 -1)\* (5\*10-9)

= 24 \* (5\*10-9) (giây)

Vậy thực hiện 20 lệnh với kỹ thuật pipeline nhanh hơn 20 lệnh với kỹ thuật tuần tự là : Tt/Tp = 20\*22\*10-9 / (24\*5\*10-9) =11/3 lần.

**83.Nêu các đặc điểm trong kỹ thuật cài đặt RISC.**

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

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

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

**85.Trình bày kỹ thuật đường ống đơn vị lệnh (pipeline lệnh).**

* Chia quá trình thực hiện câu lệnh thành nhiều công đoạn và cho các công đoạn thực hiện gối nhau.
* Các loại sung đột:
  + Cấu trúc: Một số công đoạn của các lệnh khác nhau cùng dùng chung một tài nguyên phần cứng.
  + Dữ liệu
  + Điều khiển

Lý thuyết:

1. 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 = 16MB

2. Trình bày sơ đồ khối chung của hệ thống vào ra trong máy tính

- CPU thực hiện trao đổi thông tin với các thiết bị ngoại vi thông qua các thiết bị giao diện (các khối ghép nối) như sau:

Thiết bị giao diện là thiết bị khả trình (có thể lập trình), có 3 loại thanh ghi

+ Thanh ghi điều khiển (control): nhận và chứa các từ điều khiển xác lập chế độ làm việc

+ Thanh ghi trạng thái (status): chứa thông tin trạng thái làm việc

+ Thanh ghi dữ liệu (data): làm bộ đệm nơi trung chuyển dữ liệu vào ra

--> mỗi thanh ghi đều được gán một địa chỉ nhất định, địa chỉ thanh ghi dữ liệu được gọi là địa chỉ cơ sở của công hay gọi tắt là địa chỉ cổng

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

- Thiết bị giao diện được chế tạo theo module, gọi là module vào/ra hoặc gọi là control card. Mỗi module có 3 loại thanh ghi:

+ các thanh ghi điều khiển nhận và chứa các từ điều khiển xác lập chế độ làm việc của thiết bị

+ các thanh ghi trạng thái chứa thông tin phản ánh trạng thái làm việc của thiết bị ngoại vi

+ các thanh ghi dữ liệu thực hiện chức năng bộ đệm, trung chuyển dữ liệu ra vào

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

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

5. Trình bày phương pháp vào ra dữ liệu theo định trình

- Đây là phương pháp mà quá trình vào ra được thưc hiện tức thời nhờ các lệnh vào/ra và CPU không cần quan tâm đến trạng thái của thiết bị vào ra (bao gồm giao diện và thiết bị ngoại vi)

+ Nhược điểm: độ tin cậy trong truyền, nhận dữ liệu không cao, dữ liệu truyền nhận dễ bị chồng lên nhau gây mất dữ liệu nếu bên nhận xử lý không kịp

- phương pháp này thích hợp với những quá trình vào/ra có chu kỳ cố định và có thể xác định trước.

6. Trình bày phương pháp vào ra theo kiểu thăm dò:

- CPU kiểm tra trạng thái sẵn sàng làm việc của thiết bị trước khi việc vào/ra dữ liệu được thực hiện. Khi thiết bị chưa sẵn sàng làm việc thì CPU lại phải tiếp tục thăm dò, việc thăm dò được

lặp đi lặp lại cho đến khi thiết bị sẵn sàng trao đổi dữ liệu với CPU thì quá trình nhận dữ liệu được tiến hành

Ưu điểm: quá trình nhận dữ liệu có độ tin cậy cao vì việc truyền nhận dữ liệu chỉ xảy ra khi hai bên truyền nhận đều sẵn sàng

Nhược điểm: chiếm dụng nhiều thời gian của CPU cho việc thăm dò nên hiệu quả hệ thống không cao.

7. Trình bày phương pháp dữ liệu vào ra theo kiểu ngắt cứng.

- Khái niệm ngắt cứng: CPU phải tạm dừng tiến trình đang thực hiện để chuyển sang thực hiện tiến trình phục vụ ngắt khi có yêu cầu ngắt từ phần mạch bên ngoài gửi đến CPU

- Phương pháp vào ra theo kiểu ngắt cứng là phương pháo mà thiết bị vào ra chủ động khởi tạo quá trình vào/ra dữ liệu nhờ hệ thống ngắt

Ưu điểm: quá trình vào ra dữ liệu được thực hiện ngay khi có yêu cầu, độ tin cậy cao. Hiệu suất CPU cao vì không mấy thời gian thăm dò trạng thái của thiết bị vào ra

Tuy nhiên, pp này vẫn phải thông qua CPU nên hiệu năng hoạt động toàn hệ thống chưa phải là cao nhất

8.Trình bày cấu trúc của hệ thống vào ra theo kiểu ngắt cứng

- Hệ thống ngắt cứng gồm thiết bị điều khiển ngắt PIC, BUS dữ liệu và các thiết bị vào ra

- PIC nhận các yêu cầu ngắt IRQi ( i = 0 -> 7), xử lý ưu tiên ngắt và cung cấp số hiệu 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

với chương trình được chọn

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

Quá trình xảy ra cụ thể như sau:

- CPU đang thực hiện tiến trình A

- Các thiết bị vào/ra có yêu cầu phục vụ, phát ra tín hiệu IRQi (i = 0 -> 7) tới PIC. Mỗi thiết bị vào ra 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 yêu cầu 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 tín hiệu sẵn sàng phục vụ yêu cầu ngắt tới PIC

- PIC phát số tín hiệu ngắt được chọn tới CPU qua bus dữ liệu

- Dựa vào số liệu này CPU xác định địa chỉ chương trình con phục vụ ngắt, kích hoạt và thực hiện vào ra dữ liệu

- Kết thúc, CPU lấy lại trạng thái cũ của CPU và tiến trình A vừa bị ngắt để tiếp tục thi hành

Ưu điểm: quá trình trao đổi dữ liệu có độ tin cậy rất cao vì việc truyền nhận chỉ xảy ra khi 2 bên truyền và nhận đều sẵn sàng

Nhược: chiếm dụng nhiều thời gian của CPU cho việc tham dò nên hiệu quả của hệ thống không cao

10. Trình bày khái niệm quá trình DMA, cấu trúc của hệ thống vào ra theo kiểu DMA

Khái niệm:

- DMA: direct memory access (truy cập bộ nhớ trực tiếp)

- Quá trình DMA là quá trình vào/ra dữ liệu giữa bộ nhớ và thiết bị ngoại vi mà không thông qua CPU

- Cấu trúc hệ thống: thêm mạch DMAC ghép nối với bus hệ thống để điều khiển trao đổi dữ liệu giữa module vào ra với bộ nhớ chính

11. Trình bày quá trình vào/ra dữ liệu theo kiểu DMA

- CPU đang hoạt động bình thường (CPU quản lý bus hệ thống gồm bus A - bus địa chỉ, bus D - bus dữ liệu, bus C - bus điều khiển)

- DMAC được xác lập chế độ làm việc, nhận thông tin về địa chỉ đầu khối nhớ chứa dữ liệu và kích thước khối dữ liệu cần truyền

- Các thiết bị vào/ra phát tín hiệu DRQi choa DMAC để chọn tín hiệu có mức ưu tiên cao nhất

- DMAC phát tín hiệu BRQ/HOLD = 1 cho CPU, yêu cầu CPU chuyển nhượng bus

- Nếu CPU chấp nhận, CPU thực hiện nốt chu kỳ máy, phát tín hiệu BGT/HLDA chấp nhận chuyển nhượng bus

- CPU tự tách ra khỏi hệ thống bus, quyền điều khiển bus thuộc về DMAC

- DMAC phất tín hiệu DACKi báo cho thiết bị được chọn vào/ra dữ liệu. DMAC phất địa chỉ ô nhớ đầu tiên của khối dữ liệu cần truyền lên bus A,

quá trình truyền dữ liệu bắt đầu. Trong quá trình truyền, DMAC giảm bộ đếm và tăng nội dung con trỏ chứa địa chỉ đến khi nội dung bộ đếm bằng 0

thì khối dữ liệu đã truyền xong.

- DMAC kết thúc quá trình DMA, phát tín hiệu BRQ/HOLD = 0 cho CPU, trả lại quyền điều khiển bus cho CPU, CPU tiếp tục làm việc bình thường

12. Các đặc điểm của 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

13. Cho biết ý nghĩa của cụm từ RISC (reduce instruction set computer)

- RISC (reduce instruction set computer): máy tính với tập lệnh đơn giản hóa, là một phương pháp thiết kế các bộ vi xử lý

theo hướng đơn giản hóa tập lệnh, trong đó thời gian thực thi tất cả các lệnh đều như nhau. Hiện nay các bộ vi xử lý RISC phổ biến

là ARM, SuperH, MIPS, SPARC, DEC, Alpha, PA-RISC, PIC và PowerPC của IBM

14. Trình bày kỹ thuật đường ống đơn vị lệnh

- Chia một lệnh ra làm nhiều công đoạn và cho xử lý các công đoạn gối lên nhau.