



## BT KTMT - Bài tập

Kiến trúc máy tính (Trường Đại học Cần Thơ)



Scan to open on Studocu

## BÀI 1

1) Hãy mô tả các thành phần cơ bản của kiến trúc máy tính theo mô hình John von Neumann. Giải thích ngắn gọn về vận hành của máy tính.

- Các thành phần cơ bản của máy tính bao gồm:

\*Ngõ nhập

\*Ngõ xuất

\*Bộ nhớ

\*Bộ xử lí

- Ngõ nhập ghi dữ liệu vào bộ nhớ, sau đó bộ xử lí nhận lệnh và dữ liệu từ bộ nhớ để xử lí, xử lí xong dữ liệu được truyền cho bộ nhớ, sau đó ngõ xuất đọc dữ liệu ra từ bộ nhớ.

2) Hãy mô tả và giải thích ngắn gọn về các lớp phân cấp từ phần cứng đến phần mềm.

- Được phân thành 3 cấp: ứng dụng – phần mềm hệ thống – phần cứng.

Trong đó: phần mềm hệ thống có nhiệm vụ làm cầu nối, giao tiếp trực tiếp với phần cứng nhằm hỗ trợ cho các ứng dụng. Phần mềm hệ thống gồm hệ điều hành ‘điều hành chương trình, quản lý các nguồn tài nguyên của máy tính nhằm hỗ trợ các chương trình chạy trên máy tính đó’, và trình biên dịch ‘dịch các ngôn ngữ cấp cao sang hợp ngữ - Assembly’

3) Cho một màn hình màu sử dụng 8 bit để hiển thị một màu cơ bản (đỏ - Red, xanh lá cây - Green, xanh dương - Blue) trong mỗi pixel với độ phân giải 1280x800 pixel.

Hãy cho biết dung lượng tối thiểu cần có của bộ đệm màn hình để có thể chứa một khung ảnh?

- Mỗi pixel cần  $3 \times 8 = 24$  bit

→ toàn bộ màn hình cần  $1280 \times 800 \times 24 = 24\,576\,000$  bit = 3 072 000 byte.

## BÀI 2:

1) Cho 3 bộ xử lý P1, P2 và P3: cùng thực thi một tập lệnh với các tần số xung clock và CPI được cho như bảng bên dưới:

Bộ xử lý Clock Rate CPI

Bộ xử lý	Clock Rate	CPI
P1	2 Ghz	1.5
P2	1.5 Ghz	1.0
P3	3 Ghz	2.5

a) Bộ xử lý nào có hiệu suất cao nhất dựa theo tiêu chí số lệnh thực thi trong 1 giây (IPS) và số triệu lệnh thực thi trong một giây (MIPS).

Tiêu chí số lệnh thực thi:

$$P1: 1s = (\#instructions1 * CPI1) / \text{clock rate } 1$$

$$\Rightarrow \#instructions1 = \text{clock rate } 1 / CPI1 = 2e9 / 1.5 = 1.333e9$$

$$P2: 1s = (\#instructions2 * CPI2) / \text{clock rate } 2$$

$$\Rightarrow \#instructions2 = \text{clock rate } 2 / CPI2 = 1.5e9 / 1 = 1.5e9$$

$$P3: 1s = (\#instructions3 * CPI3) / \text{clock rate } 3$$

$$\Rightarrow \#instructions3 = \text{clock rate } 3 / CPI3 = 3e9 / 2.5 = 1.2e9$$

Tương tự với tiêu chí số triệu lệnh thực thi.

$\Rightarrow$  P2 có hiệu suất cao nhất vì cùng một thời gian mà thực thi nhiều lệnh nhất.

b) Nếu các bộ xử lý thực thi 1 chương trình nào đó hết 10 giây, tính tổng số chu kì và tổng số lượng lệnh tương ứng.

Theo câu a ta có:

Tổng số lệnh P1 thực hiện trong 10 giây:  $10 * 1.333e9 = 13.33e9$

$\Rightarrow$  Tổng số chu kì = tổng số lệnh \* CPI =  $13.33e9 * 1.5 = 19.95e9$

Tổng số lệnh P1 thực hiện trong 10 giây:  $10 * 1.5e9 = 15e9$

→ Tổng số chu kì = tổng số lệnh \*CPI=15e9

Tổng số lệnh P1 thực hiện trong 10 giây:  $10 * 1.2e9 = 12e9$

→ Tổng số chu kì = tổng số lệnh \*CPI= 30e9

c) Nếu chúng ta mong muốn giảm 30% thời gian thực thi sẽ dẫn tới việc tăng 20% CPI thì tần số xung clock mới của từng bộ xử lý tương ứng phải là bao nhiêu?

Dựa vào câu a ta có:

$$1s = (\#instructions * CPI) / clockrate$$

$$\Leftrightarrow 1 - 1 * 30\% = (1.333e9 * (1.5 + 1.5 * 20\%)) / clockrate \rightarrow clockrate = 3.428Ghz$$

Tương tự ta có:

$$Clockrate_2 = 2.571Ghz$$

$$Clockrate_3 = 5.142Ghz$$

2) Xét 2 cách thiết kế khác nhau của cùng kiến trúc bộ lệnh cho hai bộ xử lý P1 và P2. Có 4 lớp lệnh: A, B, C và D. Tần số xung và CPI của mỗi cách thiết kế được cho như bảng bên dưới:

Bộ xử lý	Clock rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
P1	1.5 Ghz	1	2	3	4
P2	2 Ghz	2	2	2	3

a) Cho một chương trình với  $10^6$  lệnh được chia thành các lớp sau: 10% lớp A, 20% lớp B, 50% lớp C và 20% lớp D. Cách thiết kế cho bộ xử lý nào thực thi chương trình này nhanh hơn.

$$T1 = (10^6 * (10\% * 1 + 20\% * 2 + 50\% * 3 + 20\% * 4)) / (1.5 * 10^9) = 0.00186s$$

$$T2 = (10^6 * (10\% * 2 + 20\% * 2 + 50\% * 2 + 20\% * 3)) / (2 * 10^9) = 0.0011s$$

=> P2 nhanh hơn vì thời gian chậm hơn.

b) Tính CPI trung bình của mỗi bộ xử lý với chương trình trên

CPI đoạn code 1 = tổng số chu kì/ tổng số lệnh  
=( $10^6 * (10\% * 1 + 20\% * 2 + 50\% * 3 + 20\% * 4)$ )/ $10^6$ =2.8

CPI đoạn code 2 = tổng số chu kì/ tổng số lệnh  
=( $10^6 * (10\% * 2 + 20\% * 2 + 50\% * 2 + 20\% * 3)$ )/ $10^6$ =2.2

c) Tính tổng số chu kì xung clock của chương trình trên P1 và P2

Tổng số chu kì xung clock P1  
=( $10^6 * (10\% * 2 + 20\% * 2 + 50\% * 2 + 20\% * 3)$ )= $2.8 * 10^6$

Tổng số chu kì xung clock P2  
=( $10^6 * (10\% * 2 + 20\% * 2 + 50\% * 2 + 20\% * 3)$ )= $2.2 * 10^6$

### **BÀI 3**

**1) Hãy tìm mã hợp ngữ MIPS tương đương với chương trình C như sau:**

a)

**x = 5**

giả sử x->\$s0

addi \$s0,\$zero,5

**y = x - 2**

giả sử y->\$s0 x->\$s1

addi \$s0,\$s1,-2

**a = x \* 4**

giả sử a->\$s0 x->\$s1

sll \$s2,\$s1,2

addi \$s0,\$s2,\$zero

**b = y \* 2**

giả sử  $b \rightarrow \$s0$   $y \rightarrow \$s1$

`sll $s2,$s1,1`

`addi $s0,$s2,$zero`

$$z = (x + a) - (y + b)$$

giả sử  $z \rightarrow \$s0$   $x \rightarrow \$s1$   $y \rightarrow \$s2$   $a \rightarrow \$s3$   $b \rightarrow \$s4$

`add $s5,$s1,$s3`

`add $s6,$s2,$s4`

`sub $s0,$s5,$s6`

**Biết rằng các biến x, y, a, b là các số nguyên 32 bit**

**b)**

$$x = y + a[5]$$

giả sử  $x \rightarrow \$s0$   $y \rightarrow \$s1$   $a \rightarrow \$s2$

`lw $s3,20($s2)`

`add $s0,$s1,$s3`

Giả sử mỗi phần tử của mảng a là một word/tù nhớ 4 bytes.

**2) Hãy phân biệt sự khác nhau giữa**

**Big End (Big Endian) và**

**Little End (Little Endian)**

Little end là khi byte có giá trị nhỏ nhất được lưu trước byte có giá trị lớn nhất, còn big end là khi byte có giá trị lớn nhất được lưu trước byte có giá trị nhỏ nhất.

**MIPS sử dụng Big End hay Little End?**

MIPS sử dụng Big End.

**3) Cho 2 số x, y viết theo nhu sau:**

x	y
4	-6
12	-9
19	-12
60	88

Hãy biểu diễn x, y theo nhị phân 8 bit có dấu bù 2 và thực hiện tính ( $x + y$ ) theo hệ nhị phân

X		Y		X+Y
4	00000100	-6	11111010	11111110
12	00001100	-9	11110111	00000011
19	00010011	-12	11110100	00000111
60	00111100	88	01011000	10010100(-108)->tràn

#### 4) Liệt kê các định dạng lệnh MIPS

R-type hoặc R-format (cho các lệnh chỉ làm việc với thanh ghi)

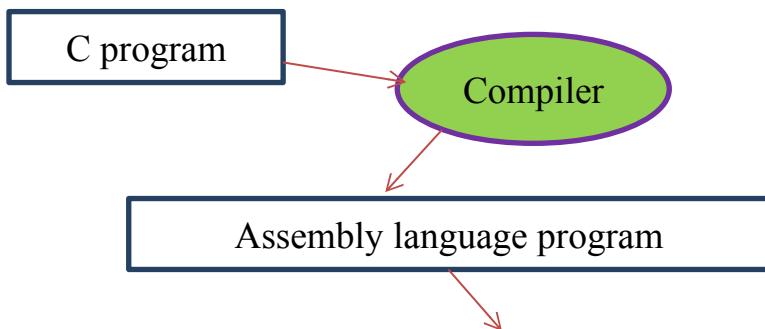
I-type hoặc I-format (cho các lệnh liên quan đến số tức thời và truyền dữ liệu)

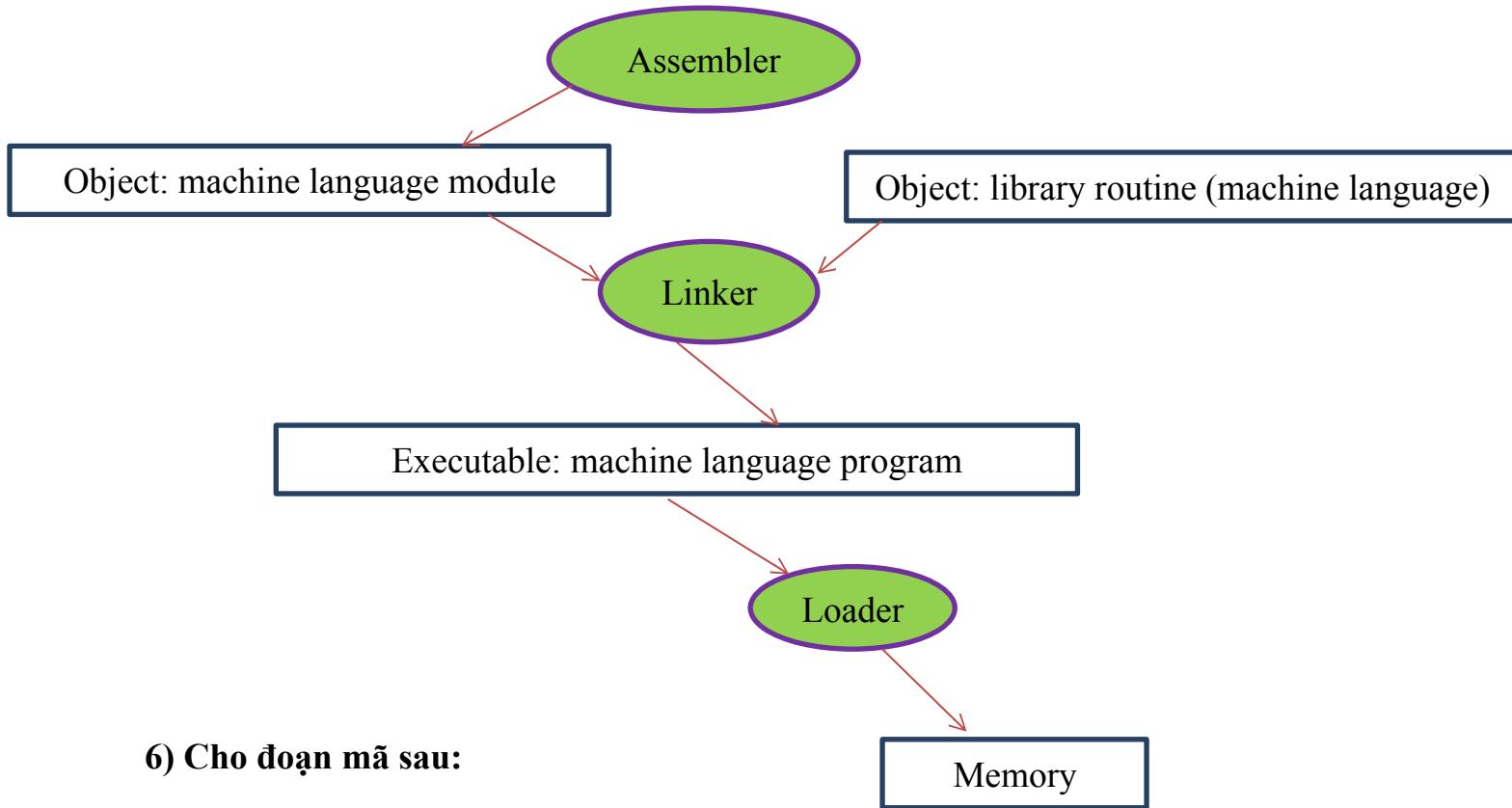
J-type hoặc J-format (lệnh nhảy lệnh ra quyết định)

Hãy biểu diễn nhị phân lệnh: add \$t0,\$s1,\$s2

op	rs	rt	rd	shamt	funct
000000	10001	10010	01000	00000	100000

5) Mô tả các bước chuyển 1 chương trình C thành mã máy thực thi trên máy tính





**6) Cho đoạn mã sau:**

**if (x < 0)**

**h = 0;**

**else**

**h = x;**

**với x, h được lưu trong \$s0, \$s1**

**Hãy dịch sang assembly MIPS đoạn mã trên**

**Slt \$s2,\$s0,\$zero**

**Beq \$s2,\$zero,else**

**Addi \$s1,\$zero,0**

**J exit**

**Else add \$s1,\$s0,\$zero**

**Exit:**

## **7) Các bước thực thi thủ tục**

1. Đặt các tham số ở một nơi mà thủ tục có thể truy xuất được.
2. Chuyển quyền điều khiển cho thủ tục.
3. Yêu cầu tài nguyên lưu trữ cần thiết cho thủ tục đó.
4. Thực hiện công việc (task).
5. Lưu kết quả ở một nơi mà chương trình có thể truy xuất được.
6. Trả điều khiển về vị trí mà thủ tục được gọi.

## **8) Cấu trúc thủ tục**

Phần đầu:

- Khai báo kích thước cho stack.
- Lưu các thanh ghi cần thiết.

Phần thân:

- Thực hiện chức năng yêu cầu (có thể gọi các thủ tục khác...).

Phần cuối:

- Phục hồi các thanh ghi cần thiết.
- Xóa stack.

## **BÀI 4**

1) Dùng 8 bit nhị phân để biểu diễn số nguyên hệ thập phân. Thực hiện các phép tính:

a)  $7 + 8$

$$7 = 0000\ 0111$$

$$8 = 0000\ 1000$$

$$\begin{array}{r} 0000\ 0111 \\ + 0000\ 1000 \\ \hline \end{array}$$

$$+ \quad 0000\ 1000$$

$$= \quad 0000\ 1111(15)$$

b)  $7 - 8$

$$7 = 0000\ 0111$$

$$8 = 0000\ 1000 \quad (-8 = 11111000)$$

$$0000\ 0111$$

$$- \quad 0000\ 1000$$

$$= \quad 1111\ 1111(-1)$$

c)  $7 \times 8$

Step	Action	Multiplicand	Product/multiplier
0	Khởi tạo	0000 0111	0000 0000 0000 1000
1	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0000 0000 0100
2	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0000 0000 0010
3	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0000 0000 0001
4	Bit cuối Product/multiplier = 1 Product/multiplier = nửa cao Product/multiplier + multiplicand Shift right Product/multiplier	0000 0111	0000 0111 0000 0001  0000 0011 1000 0000
5	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0001 1100 0000
6	Bit cuối Product/multiplier =	0000 0111	0000 0000 1110

	0 Shift right Product/multiplier		0000
7	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0000 0111 0000
8	Bit cuối Product/multiplier = 0 Shift right Product/multiplier	0000 0111	0000 0000 0011 1000

d)  $13 : 4$

Step	Action	Quotient	divisor	remainder
0	Khởi tạo	00000	00100 00000	00000 01101
1	R=R-D $R < 0 \Rightarrow R=R+D$ , dịch trái Q 1 bit Dịch phải D 1 bit	00000	00010 00000	00000 01101
2	R=R-D $R < 0 \Rightarrow R=R+D$ , dịch trái Q 1 bit Dịch phải D 1 bit	00000	00001 00000	00000 01101
3	R=R-D $R < 0 \Rightarrow R=R+D$ , dịch trái Q 1 bit Dịch phải D 1 bit	00000	00000 10000	00000 01101
4	R=R-D $R < 0 \Rightarrow R=R+D$ , dịch trái Q 1 bit Dịch phải D 1 bit	00000	00000 01000	00000 01101
5	R=R-D $R > 0 \Rightarrow$ , dịch trái Q 1 bit, Q[0]=1	00001	00000 00100	00000 00101

	Dịch phải D 1 bit			
6	R=R-D R>0 =>, dịch trái Q 1 bit, Q0=1 Dịch phải D 1 bit	00011	00000 00010	00000 00001

Chú ý, trình bày các bước thực thi của giải thuật.

2) Thực hiện các phép tính trên số thực chấm động:

a)  $2.125 + 1.25$

$$2.125 = 17/8 = 1.0001 * 2^{-3} = 1,0001 * 2^1$$

$$1.25 = 5/4 = 101 * 2^{-2} = 1.01 * 2^0$$

$$1,0001 * 2^1 + 1.01 * 2^0 = 1.0001 * 2^1 + 0.101 * 2^1$$

$$1.0001 + 0.101$$

$$= 1.1011$$

$$\Rightarrow 1.1011 * 2^1 = 11011 * 2^{-3} = 27/8 = 3.375$$

b)  $1.25 \times 2.5$

$$1.25 = 5/4 = 101 * 2^{-2} = 1.01 * 2^0$$

$$2.5 = 5/2 = 101 * 2^{-1} = 1.01 * 2^1$$

$$1.01 * 2^0 * 1.01 * 2^1 = (1.01 * 1.01) * 2^1$$

$$\begin{array}{r} 101 \\ *101 \\ \hline 101 \\ 000 \\ 101 \end{array}$$

$$= 11001$$

$$\Rightarrow 1.1001 * 2^1 = 11001 * 2^{-3} = 25/8 = 3.125$$

c)  $2.5 : 0.25$

$$2.5 = 5/2 = 101 * 2^1 - 1 = 1.01 * 2^1$$

$$0.25 = 1/4 = 1 * 2^{-2}$$

$$1.01 * 2^1 : 1 * 2^{-2} = (1.01:1) * 2^3$$

$$= 1.01 * 2^3 = 1010 * 2^0 = 10$$