

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH



KIẾN TRÚC MÁY TÍNH

CT173

Nhóm 09

Sinh viên thực hiện: Huỳnh Thanh Phong

MSSV: B2207555

Mã lớp: D22Z6A1

Giáo viên giảng dạy: Th.S Lâm Chí Nguyễn

Năm học 2023

CHƯƠNG 1. MÁY TÍNH CÁC KHÁI NIỆM VÀ CÔNG NGHỆ

Câu 1. Lợi ích của máy tính cho xã hội và cuộc sống hàng ngày?

- Nâng cao năng suất (robotics,...)
- Hợp hóa giao tiếp (Chuyển đổi ngôn ngữ,...).
- Tốc độ tính toán nhanh, lưu trữ dữ liệu lớn và truy xuất dữ liệu nhanh (Cơ sở dữ liệu về di truyền học, về quỹ đạo phóng vệ tinh...).
- Tích hợp các tiện ích phục vụ đời sống con người (Ứng dụng,...).

Câu 2. Tác hại của máy tính cho xã hội và cuộc sống hàng ngày?

- Tác hại cho sức khỏe (Lạm dụng mạng xã hội,...)
- Nguy cơ thất nghiệp ở một số ngành nghề (Trí tuệ nhân tạo dần thay thế con người,...)
- Vấn đề an toàn thông tin cá nhân (Lừa đảo trên không gian mạng, ...)

Câu 3. Hãy kể các thiết bị nhập - xuất?

Thiết bị nhập	Thiết bị xuất
<ul style="list-style-type: none">- Bàn phím- Chuột- Micro- Máy quét (scan)- ...	<ul style="list-style-type: none">- Màn hình- Loa- Máy in, máy chiếu-

Câu 4. Giả sử màn hình màu sử dụng 8 bit cho mỗi màu cơ bản (đỏ, lục, lam) trên mỗi pixel và kích thước khung hình là 1280×1024 .

a. Kích thước tối thiểu tính bằng byte của bộ đệm khung để lưu trữ khung là bao nhiêu?

b. Tối thiểu sẽ mất bao lâu để khung được gửi trên 100 Mbit/s?

a. Mỗi pixel cần tới 3 bytes

Kích thước của bộ đệm khung để lưu trữ khung là: $3 \times 1280 \times 1024 = 3\,932\,160$ bytes

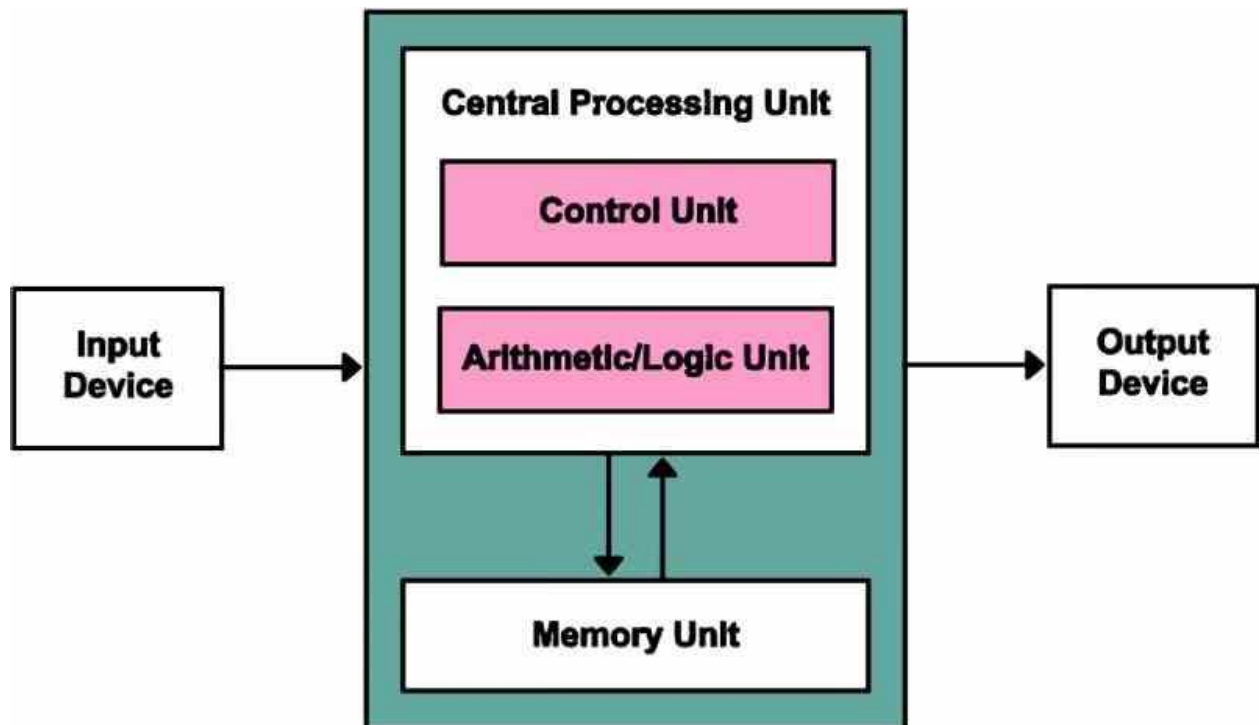
b. Ta có 1 Megabit = 1 000 000 bits

1 byte = 8 bit

$3\,932\,160 \text{ bytes} = 31\,457\,280 \text{ bits}$

Tốc độ = $31\,457\,280 / 1\,000\,000 = 31.46 \text{ s}$.

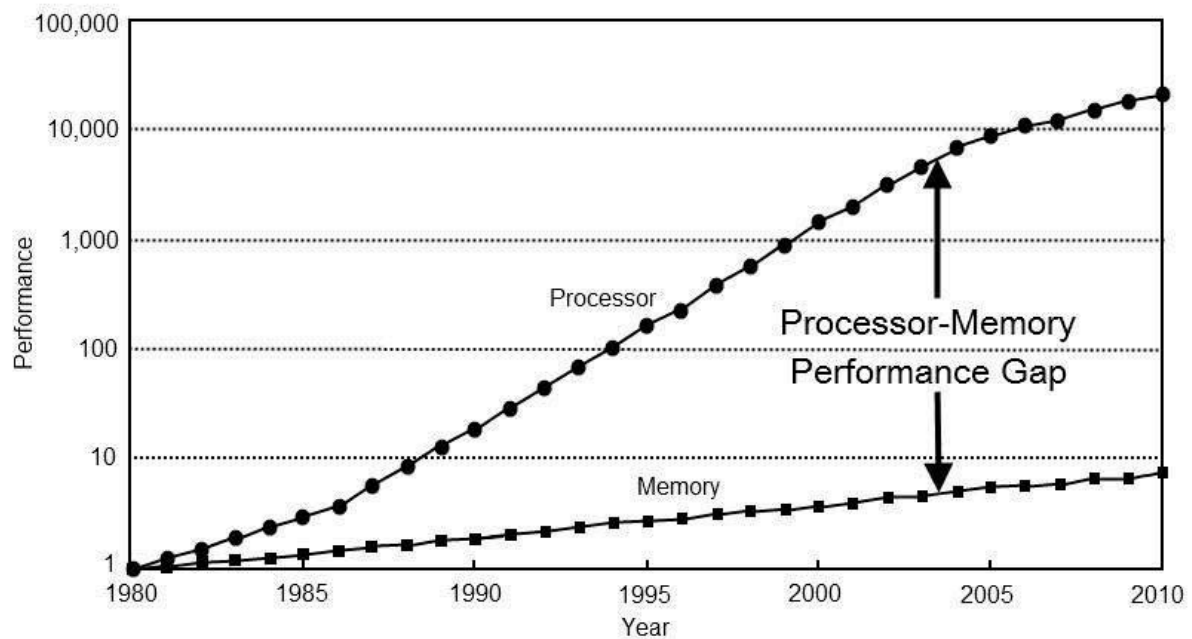
Câu 5. Trình bày mô hình kiến trúc máy tính John von Neumann, 1945 cùng với những hạn chế?



Kiến trúc Von Neumann dựa trên tất cả các bộ xử lý PC, vì tất cả chúng đều được tổ chức với một loạt các thành phần chung, như sau:

- **Đơn vị điều khiển (Control Unit):** Trong phụ trách các giai đoạn thu nhận và giải mã của chu trình lệnh.
- **Đơn vị logic-số học hoặc ALU (Arithmetic/Logic Unit):** Trong phụ trách thực hiện các phép toán và logic theo yêu cầu của chương trình.
- **Bộ nhớ (Memory Unit):** Bộ nhớ trong đó chương trình được lưu trữ, chúng ta gọi là bộ nhớ RAM
- **Thiết bị đầu vào (Input Device):** Từ đó chúng tôi giao tiếp với máy tính.
- **Thiết bị đầu ra (Output Device):** Từ đó máy tính giao tiếp với chúng ta.

Những hạn chế: Nhược điểm chính là bộ nhớ RAM, nơi chứa các lệnh và dữ liệu được xử lý, được thống nhất và chia sẻ thông qua cùng một bus dữ liệu và địa chỉ chung. Vì vậy, các hướng dẫn và dữ liệu phải được ghi tuần tự từ bộ nhớ. Nút cổ chai này được gọi là nút cổ chai Von Neumann. Đó là lý do tại sao các bộ vi xử lý khác nhau có bộ nhớ đệm gần nhất với bộ xử lý được chia thành hai loại, một cho dữ liệu và một cho lệnh.



Câu 6. Trình bày xu thế phát triển của máy tính trong tương lai dưới sự tác động của cuộc cách mạng 4.0?

Trong tương lai, có sự xuất hiện của:

- Máy tính thông minh: Một số lĩnh vực mà máy tính thông minh có thể đảm nhiệm trong tương lai gần là các chương trình thay thế chuyên gia thật: Chẩn đoán bệnh, tư vấn luật - giáo dục, đào tạo - đọc hình ảnh X - quang/ MIR - phân tích địa lý - phân tích lỗi, .v.v Máy dịch sử dụng trí tuệ nhân tạo (A.I translation) cũng đã được thử nghiệm và tiếp tục hoàn thiện. Trên thực tế, các chương trình phiên dịch như Babel Fish hay Google translator đang được hàng triệu người sử dụng.
- Máy tính mang lại trải nghiệm: Ứng dụng hấp dẫn nhất của máy tính mang lại trải nghiệm là thực tại ảo, cho phép người dùng tương tác với một môi trường do máy tính mô phỏng. Với đồ họa 3 D, các giao diện tương tác, người dùng có cảm giác đang ở trong thế giới thực.
- Máy tính sống: Máy tính sẽ dần thu nhỏ và tiện ích, gắn liền với mọi hoạt động trong cuộc sống của con người, nó dần dần sẽ là 1 phần tất yếu của cuộc sống hiện đại.

Các xu thế phát triển của máy tính trong tương lai:

1. Artificial Intelligence (AI) và Machine Learning
2. Internet of Things (IoT)
3. Virtual Reality (VR) và Augmented Reality (AR)
4. Blockchain
5. Computing Power
6. Smarter Devices (Thiết bị thông minh hơn)
7. Quantum Computing (QC)
8. 3D Printing
9. New Energy Solutions
10. Bioengineering

CHƯƠNG 2. MÁY TÍNH NHÚNG ỨNG DỤNG

Câu 1: Hãy kể tên các loại máy tính nhúng mà bạn biết?

- Máy tính Arduino.
- Máy tính Raspberry.
- Máy tính Microbit:BBC.
- Máy tính Jetson Nano.
- ...

Câu 2: Hãy cho biết một số ứng dụng máy tính nhúng hiện nay mà bạn biết?

- Internet of Things (IoT).
- Ngôi nhà thông minh.
- Điều khiển bằng giọng nói.
- Điều khiển Robot.
- Xe tự lái.
- ...

Câu 3. Trình bày những hiểu biết về ứng dụng của máy tính nhúng trong ngôi nhà thông minh?

Nhà thông minh (Home Automation) là kiểu nhà được lắp đặt các thiết bị điện, điện tử có thể được điều khiển hoặc tự động hoá hoặc bán tự động, thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử này giao tiếp với người dùng thông qua bảng điện tử đặt trong nhà, ứng dụng trên điện thoại di động, máy tính bảng hoặc một giao diện web.

Các chức năng:

- Chức năng nhận diện giọng nói: Loa thông minh, hệ thống robot phục vụ,..
- Chức năng cảm biến nhiệt độ và cường độ ánh sáng: Hệ thống làm mát tự động, hệ thống đèn tự chiếu sáng khi có người,...

- Chức năng nhận diện hình ảnh: Hệ thống cửa thông minh tự nhận diện khi có người vào, hệ thống camera giám sát,...
- Chức năng an ninh thông minh: Hệ thống cảnh báo, bảo mật,...
- Chức năng tự hành: Robot làm vườn, hệ thống vườn kín,...

Các tiện ích:

- Cảm biến không dây, kết nối bằng sóng vô tuyến...
- Kiểm soát bằng các ứng dụng thông minh như điện thoại thông minh,...
- Tiết kiệm thời gian và công sức cho người sử dụng.
- Tự động hóa tất các (hoặc một số công đoạn).

CHƯƠNG 3. HIỆU SUẤT MÁY TÍNH

LÝ THUYẾT

1. Thời gian đáp ứng (Response time): Cũng gọi là thời gian thực thi (Execution time) là tổng thời gian để máy tính hoàn thành một tác vụ nào đó, bao gồm thao tác truy cập ổ đĩa, truy cập bộ nhớ, hoạt động I/O, thời gian thực thi của hệ điều hành,...
2. Thông năng/ Hiệu suất (Throughput/Performance): Cũng gọi là bandwidth, là số lượng tác vụ hoàn thành trong một đơn vị thời gian:

Hiệu suất = $1/\text{Thời gian thực thi}$.

- So sánh hiệu suất:

Hiệu suất máy tính X > Hiệu suất máy tính

$1/\text{Thời gian thực thi của máy tính X} > 1/\text{Thời gian thực thi của máy tính Y}$.

Thời gian thực thi của máy tính X > Thời gian thực thi của máy tính Y.

⇒ Để so sánh hiệu suất giữa các máy tính với nhau thật chất là so sánh thời gian của các máy tính.

3. Xung đồng hồ máy tính (Xung clock): Hầu hết các máy tính đều cần một “đồng hồ” để xác định khi nào một sự kiện/thao tác được thực hiện trong phần cứng. Khối tạo ra các khoảng thời gian định thời cho máy tính làm việc.

- Chu kỳ xung đồng hồ/xung clock (Clock cycle time/Clock cycle/Cycle time)
- **Tần số xung đồng hồ/xung clock (Clock rate):** là số lần của một hiện tượng lặp lại trên một đơn về thời gian.
Tần số xung Clock (Clock rate) = $1/\text{Chu kỳ xung Clock (Clock cycle)}$.

4. Tính hiệu suất dựa trên chu kỳ và tần số xung đồng hồ:

- Dựa trên chu kỳ xung đồng hồ (Clock cycle/ Clock cycle time/ Cycle time):
Thời gian thực thi của một chương trình
= Tổng số chu kỳ xung clock chương trình cần X Chu kỳ xung clock.
- Dựa trên tần số xung đồng hồ (Clock rate):
Thời gian thực thi của một chương trình
= Tổng số chu kỳ xung clock chương trình cần/Tần số xung clock.
- CPI (Clock cycle per instruction): Số chu kỳ xung clock cần để thực thi một lệnh.
Tổng số chu kỳ xung clock chương trình cần
= Tổng số lệnh chương trình X CPI.
Thời gian thực thi
= Tổng số chu kỳ clock chương trình cần X Chu kỳ xung clock.
Thời gian thực thi
= Tổng số chu kỳ clock chương trình cần /Tần số xung clock.

⇒ Thời gian thực thi = Tổng số lệnh chương trình X CPI X Chu kỳ xung clock.

⇒ Thời gian thực thi = (Tổng số lệnh chương trình X CPI)/Tần số xung clock.

5. MIPS (Million instruction per second): Một cách đo tốc độ thực thi của chương trình dựa trên số lượng triệu lệnh trên giây. MIPS được tính bằng số lượng lệnh chia cho tích của thời gian thực thi và giá trị 10^6 .

$\text{MIPS} = \text{Tổng số lệnh chương trình}/(\text{Thời gian thực thi} \times 10^6)$.

$\text{MIPS} = \text{Tổng số lệnh chương trình}/[(\text{Tổng số lệnh chương trình} \times \text{CPI})/\text{Tần số xung Clock}] \times (1/10^6)$.

⇒ $\text{MIPS} = \text{Tần số xung clock}/(\text{CPI} \times 10^6)$.

6. IPS (Instruction per second): Số lệnh trên giây.

$$IPS = MIPS \times 10^6.$$

7. Số chu kỳ xung Clock trung bình:

$$CPU \text{ Cycles} = \sum_{i=1}^n (CPI_i \times C_i).$$

$$\Rightarrow CPI \text{ (trung bình)} = \frac{CPU \text{ Cycles}}{\sum_{i=1}^n C_i} = \frac{\sum_{i=1}^n (CPI_i \times C_i)}{\sum_{i=1}^n C_i}$$

BÀI TẬP

Câu 1. Hãy xem xét ba bộ xử lý khác nhau P1, P2 và P3 thực hiện cùng một tập lệnh. P1 có tốc độ xung nhịp 3 GHz và CPI là 1,5. P2 có tốc độ xung nhịp 2,5 GHz và CPI là 1,0. P3 có tốc độ xung nhịp 4,0 GHz và có CPI là 2,2.

a. Bộ xử lý nào có hiệu suất cao nhất được thể hiện bằng hướng dẫn mỗi giây?

b. Nếu mỗi bộ xử lý thực hiện một chương trình trong 10 giây, hãy tìm số chu kỳ và số lượng lệnh.

c. Chúng tôi đang cố gắng giảm 30% thời gian thực hiện nhưng điều này dẫn đến chỉ số CPI tăng 20%. Chúng ta nên có tốc độ xung nhịp bao nhiêu để giảm thời gian này?

	P1	P2	P3
Tần số xung nhịp	3 GHz	2.5 GHz	4.0 GHz
CPI	1.5	1.0	2.2

a.

Do cả 3 cùng thực hiện chung một tập lệnh

Ta có:

Thời gian thực hiện = CPI/Tần số xung

$$T1 = 1.5/3 = 0.5(s)$$

$$T2 = 1.0/2.5 = 0.4(s)$$

$$T3 = 2.2/4.0 = 0.55(s)$$

Hiệu suất = 1/Thời gian thực hiện.

$$H1 = 1/0.5 = 2$$

$$H2 = 1/0.4 = 2.5$$

$$H3 = 1/0.55 = 1.82$$

b.

$$t = 10(s).$$

Thời gian thực hiện = Tổng số chu kỳ / Tần số xung

Thời gian thực hiện = (Tổng số lệnh X CPI) / Tần số

	P1	P2	P3
Tổng số chu kỳ	30	25	40
Tổng số lệnh	20	25	18

c.

Ta có:

$$T2 = \frac{NXCPI2}{F2}$$

$$\Rightarrow 0,7T1 = \frac{NXCPI2}{F2}$$

$$\Rightarrow 0,7 \times \frac{NXCPI1}{F1} = \frac{NXCPI2}{F2}$$

$$\Rightarrow \frac{0,7}{F1} = \frac{1,2}{F2}$$

$$\Rightarrow F2 = \frac{1,2}{0,7} F1$$

\Rightarrow Tăng Tần số xung lên 171,4283%

	P	P1	P2	P3
T1	T1	T1		
T2	0.7T1	0.7T1		
CPI1	CPI1	1.5	1.0	2.2
CPI2	1.2CPI1	1.8	1.2	2.64
F1	F1	3 GHz	2.5 GHz	4.0 GHz
F2	$\frac{1,2}{0,7} F1$	$\frac{36}{7} GHz$	$\frac{30}{7} GHz$	$\frac{48}{7} GHz$

Câu 2. Hãy xem xét hai cách triển khai khác nhau của cùng một kiến trúc tập lệnh. Các hướng dẫn có thể được chia thành bốn lớp theo CPI của chúng (lớp A, B, C và D). P1 với tốc độ xung nhịp 2,5 GHz và CPI là 1, 2, 3 và 3 và P2 với tốc độ xung nhịp là 3 GHz và CPI là 2, 2, 2 và 2. Cho một chương trình có số lệnh động là Hướng dẫn 1.0E6 được chia thành các lớp như sau: 10% lớp A, 20% lớp B, 50% lớp C và 20% lớp D, việc triển khai nào nhanh hơn?

a. CPI toàn cầu cho mỗi lần thực hiện là bao nhiêu?

b. Tìm chu kỳ đồng hồ cần thiết trong cả hai trường hợp.

	P1				P2			
CPI	A(10%)	B(20%)	C(50%)	D(20%)	A(10%)	B(20%)	C(50%)	D(20%)
	1	2	3	3	2	2	2	2
F	2.5 GHz				3 GHz			
N	1.0E6 = 10 ⁶							
	1.10 ⁵	2. 10 ⁵	5. 10 ⁵	2. 10 ⁵	1. 10 ⁵	2. 10 ⁵	5. 10 ⁵	2. 10 ⁵

$$T=1/F$$

$$\text{Tổng số chu kỳ thực thi P1} = \sum_{i=1}^n (C_i \times CPI_i) = (1 \times 1 + 2 \times 2 + 3 \times 5 + 3 \times 2) \cdot 10^5 = 26 \cdot 10^5$$

$$\text{Tổng số chu kỳ thực thi P2} = \sum_{i=1}^n (C_i \times CPI_i) = (2 \times 1 + 2 \times 2 + 2 \times 5 + 2 \times 2) \cdot 10^5 = 20 \cdot 10^5$$

$$\text{CPI trung bình P1 } CPI_{tb} = \frac{\text{Tổng số chu kỳ thực thi P1}}{\text{Tổng số lệnh P1}} = \frac{26 \cdot 10^5}{10^6} = 2.6$$

$$\text{CPI trung bình P2 } CPI_{tb} = \frac{\text{Tổng số chu kỳ thực thi P2}}{\text{Tổng số lệnh P2}} = \frac{20 \cdot 10^5}{10^6} = 2.0$$

Thời gian thực thi

$$t1 = \frac{N \times CPI_{tb1}}{F} = \frac{10^6 \times 2.6}{2.5 \times 10^6} = 1.04$$

$$t2 = \frac{N \times CPI_{tb2}}{F} = \frac{10^6 \times 2.0}{3.0 \times 10^6} = 0.67$$

	P1	P2
CPI_{tb}	2.6	2.0
F	2.5 GHz	3 GHz
T	0.4 s	0.33 s
t	1.04 s	0.67 s

⇒ P2 có thời gian thực hiện nhanh hơn.

Câu 3. Trình biên dịch có thể có tác động sâu sắc đến hiệu suất của ứng dụng. Giả sử rằng đối với một chương trình, trình biên dịch A tạo ra số lệnh động là 1,0E9 và có thời gian thực hiện là 1,1 giây, trong khi trình biên dịch B tạo ra số lệnh động là 1,2E9 và thời gian thực hiện là 1,5 giây.

a. Tìm CPI trung bình cho mỗi chương trình biết rằng bộ xử lý có thời gian chu kỳ xung nhịp là 1 ns.

b. Giả sử các chương trình đã biên dịch chạy trên hai bộ xử lý khác nhau. Nếu thời gian thực hiện trên hai bộ xử lý là như nhau thì đồng hồ của bộ xử lý chạy mã của trình biên dịch A nhanh hơn bao nhiêu so với đồng hồ của bộ xử lý chạy mã của trình biên dịch B?

c. Một trình biên dịch mới được phát triển chỉ sử dụng các lệnh 6.0E8 và có CPI trung bình là 1,1. Tốc độ sử dụng trình biên dịch mới này so với việc sử dụng trình biên dịch A hoặc B trên bộ xử lý gốc là bao nhiêu?

a.

Ta có:

$t = \text{Tổng số chu kỳ} \times T$

$$\Rightarrow \text{Tổng số chu kỳ} = \frac{t}{T}$$

$$\text{Tổng số chu kỳ của trình biên dịch A} = \frac{1.1}{10^{-9}} = 1.1 \cdot 10^9$$

$$\text{Tổng số chu kỳ của trình biên dịch B} = \frac{1.5}{10^{-9}} = 1.5 \cdot 10^9$$

$$CPI_{tb} = \frac{\text{Tổng số chu kỳ thực thi}}{\text{Tổng số lệnh}}$$

$$CPI_{tb} \text{ của trình biên dịch A} = \frac{1.1 \cdot 10^9}{1 \cdot 10^9} = 1.1$$

$$CPI_{tb} \text{ của trình biên dịch B} = \frac{1.5 \cdot 10^9}{1.2 \cdot 10^9} = 1.25$$

	A	B
t	1.1 s	1.5 s
T	10^{-9} s	
F	1.0 GHz	
CPI	1.1	1.5
N	$1 \cdot 10^9$	$1.2 \cdot 10^9$
CPI_{tb}	1.1	1.125

b.

Ta có:

$$t = N \times CPI1 \times T1$$

$$t = N \times CPI2 \times T2$$

$$\frac{t}{t} = \frac{N \times CPI1 \times T1}{N \times CPI2 \times T2} = \frac{CPI1 \times T1}{CPI2 \times T2}$$

$$\Rightarrow CPI2 \times T2 = CPI1 \times T1$$

$$\frac{T1}{T2} = \frac{CPI2}{CPI1} = \frac{1.5}{1.1} = 1.364$$

➔ Vậy đồ hồ của bộ xử lý chạy trình biên dịch A nhanh 1.364 hơn trình biên dịch B

c.

$$N' = 6.0E8 = 6.10^8$$

$$CPI' = 1.1$$

Thời gian thực thi:

$$t' = N' \times CPI' \times T = 6.10^8 \times 1.1 \times 10^{-9} = 0.66$$

⇒ Trình biên dịch mới có thời gian thực thi nhanh hơn trình biên dịch A,B.

Câu 4. Nếu máy tính A chạy một chương trình trong 10 giây và máy tính B chạy cùng một chương trình trong 15 giây thì A nhanh hơn B bao nhiêu?

- Ta có:

$$\begin{aligned} \frac{\text{Hiệu suất máy tính A}}{\text{Hiệu suất máy tính B}} &= \frac{\frac{1}{\text{Thời gian thực thi máy A}}}{\frac{1}{\text{Thời gian thực thi máy B}}} \\ &= \frac{\text{Thời gian thực thi máy B}}{\text{Thời gian thực thi máy A}} = \frac{15}{10} = 1.5 \end{aligned}$$

⇒ Vậy máy tính A nhanh hơn gấp 1.5 lần so với máy tính B

Câu 5. Chương trình yêu thích của chúng tôi chạy trong 10 giây trên máy tính A, có xung nhịp 2 GHz. Chúng tôi đang cố gắng giúp một nhà thiết kế máy tính xây dựng một chiếc máy tính B, sẽ chạy chương trình này trong 6 giây. Người thiết kế đã xác định rằng có thể tăng đáng kể tốc độ xung nhịp, nhưng mức tăng này sẽ ảnh hưởng đến phần còn lại của thiết kế CPU, khiến máy tính B yêu cầu số chu kỳ xung nhịp gấp 1,2 lần máy tính A cho chương trình này. Chúng ta nên yêu cầu nhà thiết kế nhắm tới tốc độ xung nhịp nào?

	Máy tính A	Máy tính B
Thời gian thực thi	10 s	6 s
Tần số xung nhịp	2 GHz	X GHz
Tổng số chu kỳ xung nhịp	$T = 20 \times 10^9$	$1.2T = 24 \times 10^9$

- Ta có:

$$\text{Thời gian thực thi máy tính A} = \frac{\text{Tổng số chu kỳ xung nhịp máy tính A}}{\text{Tần số xung nhịp máy tính A}}$$

$$\Rightarrow 10 = \frac{\text{Tổng số chu kỳ xung nhịp máy tính A}}{2 \times 10^9}$$

$$\Rightarrow \text{Tổng số chu kỳ xung nhịp máy tính A} = 20 \times 10^9$$

- Ta có:

$$\text{Thời gian thực thi máy tính B} = \frac{\text{Tổng số chu kỳ xung nhịp máy tính B}}{\text{Tần số xung nhịp máy tính B}}$$

$$\Rightarrow 6 = \frac{24 \times 10^9}{\text{Tần số xung nhịp máy tính B}}$$

$$\Rightarrow \text{Tần số xung nhịp máy tính B} = \frac{24 \times 10^9}{6} = 4 \text{ GHz.}$$

Câu 6. Giả sử chúng ta có hai cách triển khai cùng một kiến trúc tập lệnh. Máy tính A có thời gian chu kỳ xung nhịp là 250 ps và CPI là 2,0 cho một số chương trình và máy tính B có thời gian chu kỳ xung nhịp là 500 ps và CPI là 1,2 cho cùng một chương trình. Máy tính nào nhanh hơn cho chương trình này và nhanh hơn bao nhiêu?

	Máy tính A	Máy tính B
Thời gian chu kỳ xung nhịp	250 ps	500 ps
CPI	2,0	1,2
Thời gian thực thi		

Gọi kiến trúc tập lệnh là N:

Ta có:

- Thời gian thực thi máy tính A = N X CPI X Thời gian chu kỳ xung nhịp máy tính A
 \Rightarrow Thời gian thực thi máy tính A = N X 2 X 250 ps
 \Rightarrow Thời gian thực thi máy tính A = 500N ps
- Thời gian thực thi máy tính B = N X CPI X Thời gian chu kỳ xung nhịp máy tính B
 \Rightarrow Thời gian thực thi máy tính B = N X 1.2 X 500 ps
 \Rightarrow Thời gian thực thi máy tính B = 600N ps

Vậy: Máy tính A nhanh hơn máy tính B

- Ta có:

$$\frac{\text{Thời gian thực thi máy tính B}}{\text{Thời gian thực thi máy tính A}} = \frac{600 \text{ ps}}{500 \text{ ps}} = 1.2$$

Vậy: Máy tính A nhanh hơn 1.2 lần máy tính B.

Câu 7. Một nhà thiết kế trình biên dịch đang cố gắng quyết định giữa hai chuỗi mã cho một máy tính cụ thể. Các nhà thiết kế phần cứng đã cung cấp những thông tin sau:

	CPI cho mỗi lớp hướng dẫn		
	A	B	C
CPI	1	2	3

Một nhà biên dịch thiết kế đang cố gắng quyết định giữa hai mã chuỗi cho một máy tính. Phần cứng thiết kế của nhà cung cấp các thông tin sau:

	Số lượng lệnh cho mỗi lớp hướng dẫn		
	A	B	C
1	2	1	2
2	4	1	1

Chuỗi mã nào thực thi nhiều lệnh nhất? Cái nào sẽ nhanh hơn? CPI cho mỗi chuỗi là bao nhiêu?

Tổng số lệnh thực thi:

Nhóm 1: 5

Nhóm 2: 6

⇒ Chuỗi mã nhóm 1 thực thi nhiều lệnh hơn

Tổng số chu kỳ: $\sum_{i=0}^n (C_i \times \text{CPI}_i)$

Nhóm 1: $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$

Nhóm 2: $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$

⇒ Chuỗi mã nhóm 2 thực thi nhanh hơn chuỗi mã nhóm 1

CPI trung bình: $\text{CPI}_{\text{tb}} = \frac{\sum \text{Chu kỳ}}{\sum \text{Lệnh}}$

CPI trung bình nhóm 1: $\text{CPI}_{\text{tb1}} = \frac{10}{5} = 2$

CPI trung bình nhóm 2: $\text{CPI}_{\text{tb2}} = \frac{9}{6} = 1.5$

Câu 8. Một ứng dụng nhất định được viết bằng Java chạy 15 giây trên bộ xử lý máy tính để bàn. Một trình biên dịch Java mới được phát hành chỉ yêu cầu số lượng lệnh bằng 0,6 lệnh so với trình biên dịch cũ. Thật không may, nó làm tăng CPI lên 1,1. Tính thời gian của trình biên dịch mới?

Cùng máy tính → tần số và chu kỳ như nhau

	Java 1	Java 2
Thời gian thực thi	15 s	X s
CPI	CPI	1.1 CPI
Số lệnh	N	0.6 N
Tần số xung nhịp	A Hz	A Hz
Chu kỳ xung nhịp	T s	T s

- Ta có:

$T1 = N \times \text{CPI} \times T$

$T2 = 0.6N \times 1.1\text{CPI} \times T$

$$\Rightarrow \frac{T2}{T1} = \frac{0.6N \times 1.1\text{CPI} \times T}{N \times \text{CPI} \times T} = 0.66$$

$$\Rightarrow T2 = 0.66T1 = 9.9 \text{ s}$$

Câu 8. Mục 1.10 coi việc sử dụng một tập hợp con của phương trình hiệu suất làm thước đo hiệu suất là một phạm bẫy. Để minh họa điều này, hãy xem xét hai bộ xử lý sau. P1 có tốc độ xung nhịp 4 GHz, CPI trung bình là 0,9 và yêu cầu thực hiện các lệnh 5.0E9. P2 có tốc độ xung nhịp 3 GHz, CPI trung bình là 0,75 và yêu cầu thực hiện các lệnh 1.0E9.

	P1	P2
CPI	0.9	0.75
F	4 GHz	3 GHz
T	0.25 ns	0.33 ns
N	$5.0E9 = 5,0 \cdot 10^9$	$1.0E9 = 1,0 \cdot 10^9$

a. Một sai lầm thông thường là coi máy tính có tốc độ xung nhịp lớn nhất là có hiệu suất lớn nhất. Kiểm tra xem điều này có đúng với P1 và P2 không.

b. Một sai lầm khác là cho rằng bộ xử lý thực thi số lượng lệnh lớn nhất sẽ cần thời gian CPU lớn hơn. Xét rằng bộ xử lý P1 đang thực hiện chuỗi lệnh 1.0E9 và CPI của bộ xử lý P1 và P2 không thay đổi, hãy xác định số lượng lệnh mà P2 có thể thực thi cùng lúc với P1 cần thực hiện lệnh 1.0E9.

a.

Ta có

$$t = \frac{N \times CPI}{F}$$

$$- \text{Thời gian thực thi của bộ xử lý P1: } t(P1) = \frac{5,0 \cdot 10^9 \times 0,9}{4 \cdot 10^9} = 1.125$$

$$- \text{Thời gian thực thi của bộ xử lý P2: } t(P2) = \frac{1,0 \cdot 10^9 \times 0,75}{3 \cdot 10^9} = 0.25$$

Ta có:

$$H = \frac{1}{t}$$

$$- \text{Hiệu suất bộ xử lý P1: } H(P1) = \frac{1}{1.125} = 0.89$$

$$- \text{Hiệu suất bộ xử lý P2: } H(P2) = \frac{1}{0.25} = 4$$

b.

Thời gian thực thi

$$t' = \frac{N(P1) \times CPI(P1)}{F(P1)} = \frac{1,0 \cdot 10^9 \times 0.9}{4 \cdot 10^9} = 0.225$$

Ta có:

$$t' = \frac{N(P2) \times CPI(P2)}{F(P2)}$$

$$\Rightarrow N(P2) = \frac{t'_{XF(P2)}}{CPI(P2)} = \frac{0.225 \times 3.10^9}{0.75} = 0.9.10^9 = 0.9E9$$

Câu 9. Giả sử chúng tôi phát triển một bộ xử lý mới, đơn giản hơn có tải điện dung bằng 85% so với bộ xử lý cũ phức tạp hơn. Hơn nữa, giả sử rằng nó có điện áp có thể điều chỉnh để có thể giảm điện áp 15% so với bộ xử lý B, dẫn đến tần số giảm 15%. Tác động đến sức mạnh năng động là gì?

Ta có : Sức mạnh động năng (W_d) = $\frac{1}{2} \times \text{Điện dung (F)} \times [\text{Điện áp (V)}]^2 \times \text{Chuyển đổi tần số (T)}$

W_d mới	W_d cũ
$= \frac{1}{2} \times (0,85.F) \times (0,85.V)^2 \times (0,85.T)$	$= \frac{1}{2} \times F \times V^2 \times T$

$$\Rightarrow \frac{W_d \text{ mới}}{W_d \text{ cũ}} = \frac{\frac{1}{2} \times (0,85.F) \times (0,85.V)^2 \times (0,85.T)}{\frac{1}{2} \times F \times V^2 \times T}$$

$$\Rightarrow \frac{W_d \text{ mới}}{W_d \text{ cũ}} = 0,85^4 = 0,522$$

Câu 10. CPI điện tử thay đổi theo hệ số 5 đối với SPEC CPU2006 trên máy tính Intel Core i7 trong Hình 1.18, MIPS cũng vậy. Cuối cùng và quan trọng nhất, nếu một chương trình mới thực thi nhiều lệnh hơn nhưng mỗi lệnh nhanh hơn thì MIPS có thể thay đổi độc lập với hiệu suất!

Hãy xem xét các phép đo hiệu suất sau đây cho một chương trình:

	Máy A	Máy B
Tổng lệnh	10 tỷ	8 tỷ
Tần số xung clock	4Ghz	4GHz
CPI	1.0	1.1

a. Máy tính nào có xếp hạng MIPS cao hơn?

$$MIPS = \frac{\text{Tổng lệnh}}{\text{Thời gian thực hiện} \times 10^6} = \frac{\text{Tổng lệnh}}{\frac{\text{Tổng lệnh} \times CPI}{\text{Tần số xung clock}} \times 10^6} = \frac{\text{Tần số xung clock}}{CPI \times 10^6}$$

	Máy A	Máy B
MIPS	4.10^3	$\frac{40}{11}.10^3$

\Rightarrow Xếp hạng MIPS máy A cao hơn máy B

b. Máy tính nào nhanh hơn?

$$\text{Thời gian thực hiện 1 chương trình} = \frac{\text{Tổng lệnh} \times \text{CPI}}{\text{Tần số xung clock}}$$

	Máy A	Máy B
Thời gian thực hiện 1 chương trình	25s	22s

⇒ Máy B có thời gian thực hiện 1 chương trình nhanh hơn máy A

CHƯƠNG 4. KIẾN TRÚC BỘ LỆNH

LÝ THUYẾT → <https://www.cit.ctu.edu.vn/~dtngghi/cod/nasm.pdf>

BÀI TẬP

Câu 1. Sinh viên viết chương trình nhập vào hai số nguyên và hiện thị kết quả các phép toán +, -, *, /

Bài làm:

.data

myString1 : .ascii "BAI TAP VE NHA"

myString21 : .ascii "NHAP GIA TRI CUA A"

myString22 : .ascii "GIA TRI CUA A: "

myString31 : .ascii "NHAP GIA TRI CUA B"

myString32 : .ascii "GIA TRI CUA B: "

myString4: .ascii "GIA TRI A-B: "

myString5: .ascii "GIA TRI A+B: "

myString6: .ascii "GIA TRI A*B: "

myString7: .ascii "GIA TRI A%B: "

myString8: .ascii "\n"

text.

Main:

1. li \$v0, 4 la \$a0, myString1 syscall li \$v0, 4	2. syscall li \$v0, 4 la \$a0, myString31 syscall
---	---

<pre> la \$a0, myString8 syscall li \$v0, 4 la \$a0, myString21 syscall li \$v0, 5 syscall move \$s1, \$v0 li \$v0, 4 la \$a0, myString22 syscall li \$v0, 1 move \$a0, \$s1 syscall li \$v0, 4 la \$a0, myString8 </pre>	<pre> li \$v0, 5 syscall move \$s2, \$v0 li \$v0, 4 la \$a0, myString32 syscall li \$v0, 1 move \$a0, \$s2 syscall li \$v0, 4 la \$a0, myString8 </pre>
<pre> 3. syscall sub \$s3, \$s1, \$s2 li \$v0, 4 la \$a0, myString4 syscall li \$v0, 1 move \$a0, \$s3 syscall li \$v0, 4 la \$a0, myString8 syscall add \$s4, \$s1, \$s2 li \$v0, 4 la \$a0, myString5 syscall li \$v0, 1 move \$a0, \$s4 syscall li \$v0, 4 la \$a0, myString8 </pre>	<pre> 4. syscall mult \$s1, \$s2 mflo \$s5 li \$v0, 4 la \$a0, myString6 syscall li \$v0, 1 move \$a0, \$s5 syscall li \$v0, 4 la \$a0, myString8 syscall div \$s1, \$s2 mflo \$s6 li \$v0, 4 la \$a0, myString7 syscall li \$v0, 1 move \$a0, \$s6 syscall li \$v0, 4 la \$a0, myString8 syscall </pre>

Exit:

Kết quả:

```
BAI TAP VE NHA
NHAP GIA TRI CUA A 18
GIA TRI CUA A: 18
NHAP GIA TRI CUA B 9
GIA TRI CUA B: 9
GIA TRI A-B: 9
GIA TRI A+B: 27
GIA TRI A*B: 162
GIA TRI A%B: 2
```

-- program is finished running (dropped off bottom) --

Câu 2. Đối với câu lệnh C sau đây, mã hợp ngữ MIPS tương ứng là gì? Giả sử rằng các biến f, g, h và i đã cho và có thể được coi là số nguyên 32 bit như được khai báo trong chương trình C. Sử dụng số lượng tối thiểu các hướng dẫn lắp ráp MIPS.

f = g + (h - 5);

Bài làm

addi \$S1, \$S2, -5; //\$S1=F, \$S2=H, F=H-5

add \$S1, \$S3, \$S1; //\$S3 = G, F=G+H-5

Câu 3. Đối với các hướng dẫn lắp ráp MIPS sau đây ở trên, một câu lệnh C tương ứng?

add f, g, h (1)

add f, i, f (2)

Bài làm

Lệnh (1)	f = g + h
Lệnh (2)	f = i + f
Lệnh (1)(2)	f = i + g + h

Câu 4. Đối với câu lệnh C sau đây, câu lệnh tương ứng là gì? Mã lắp ráp MIPS? Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi \$s0, \$s1, \$s2, \$s3 và \$s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi \$s6 và \$s7.

B[8] = A[i-j];

Bài làm

Lệnh	Ghi chú
------	---------

Sub \$t0, \$s3, \$s4	\$t0 = i - j
Sll \$t0, \$t0, 2	Dịch \$t0 sang trái 4 byte
Add \$t0, \$t0, &s6	Lấy địa chỉ thực tế bằng cách lấy địa chỉ mảng cộng cho địa chỉ hiện tại &A[i-j]
Lw \$t1, 0(\$t0)	Lấy giá của A[i-j] lưu vào \$t1
Sw \$t1, 8*4(\$s7)	Gán giá trị \$t1 vào B[8]

Câu 5. Đối với các hướng dẫn lắp ráp MIPS bên dưới, câu lệnh C tương ứng là gì? Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi \$s0, \$s1, \$s2, \$s3 và \$s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi \$s6 và \$s7.

Bài làm

Lệnh	Giải thích	Kết quả
<i>sll \$t0, \$s0, 2</i>	Dịch trái f 4 byte lưu vào \$t0	$\$t0 = f \ll 4 \text{ byte}$
<i>add \$t0, \$s6, \$t0</i>	Lấy địa chỉ mảng A cộng cho địa chỉ \$t0 ra địa chỉ cần tìm lưu vào \$t0	$\$t0 = \&A[f] + \&t0$
<i>sll \$t1, \$s1, 2</i>	Dịch trái g 4 byte lưu vào \$t1	$\$t1 = g \ll 4 \text{ byte}$
<i>add \$t1, \$s7, \$t1</i>	Lấy địa chỉ mảng B cộng cho địa chỉ \$t1 ra địa chỉ cần tìm lưu vào \$t1	$\$t1 = \&B[g] + \$t1$
<i>lw \$s0, 0(\$t0)</i>	Lưu giá trị của f vào A[f]	$\$s0 = A[f]$
<i>addi \$t2, \$t0, 4</i>	Dịch phải 4 byte A[f]	$\$t2 = A[f+1]$
<i>lw \$t0, 0(\$t2)</i>	Lưu giá trị \$t0 cho \$t2	$\$t0 = A[f+1]$
<i>add \$t0, \$t0, \$s0</i>	Cộng \$t0 với \$s0 rồi lưu kết quả vào \$t0	$A[f+1] = A[f+1] + A[f]$
<i>sw \$t0, 0(\$t1)</i>	Gán kết quả \$t0 cho \$t1	$B[g] = A[f+1] + A[f]$

Câu 6. Hiện thị cách sắp xếp giá trị 0xabcdef12 trong bộ nhớ của máy little-endian và big-endian. Giả sử dữ liệu được lưu trữ bắt đầu từ địa chỉ 0.

0Xabcdef12

= 1010 1011 1100 1101 1110 1111 0001 0010

- little-endian: Max nằm bên trái:

1111 1110 1101 1100 1011 1010 0010 0001

- big-endian: Max nằm bên phải:

0001 0010 1010 1011 1100 1101 1110 1111

Câu 7. Dịch mã C sau sang MIPS. Giả sử rằng các biến f, g, h, i và j được gán lần lượt vào các thanh ghi \$s0, \$s1, \$s2, \$s3 và \$s4. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi \$s6 và \$s7. Giả sử rằng các phần tử của mảng A và B là các từ 4 byte:

B[8] = A[i] + A[j];

Bài làm

Lệnh	Ghi chú
Sll \$s0, \$s3, 2	Dịch trái i 4 byte rồi lưu vào f, $f = i \ll 4$
Add \$s0, \$s0, \$s6	Lấy địa chỉ cần tìm &A[i] lưu vào f, $f = \&A[i]$
Lw \$s1, 0(\$s0)	Lưu giá trị của mảng tại vị trí i vào g, $g = A[i]$
Sll \$s0, \$s4, 2	Dịch trái j 4 byte rồi lưu vào f, $f = j \ll 4$
Add \$s0, \$s0, \$s6	Lấy địa chỉ cần tìm &A[j] lưu vào f, $f = \&A[j]$
Lw \$s2, 0(\$s0)	Lưu giá trị của mảng tại vị trí j vào h, $h = A[j]$
Add \$s0, \$s1, \$s2	$f = A[i] + A[j]$
Sw \$s0, 8*4(\$s7)	Gán giá trị f cho mảng B tại vị trí 8 $B[8] = A[i] + A[j]$

Câu 8. Dịch mã MIPS sau sang C. Giả sử rằng các biến f, g, h, i và j được gán cho các thanh ghi \$s0, \$s1, \$s2, \$s3 và \$s4, tương ứng. Giả sử rằng địa chỉ cơ sở của mảng A và B lần lượt nằm trong các thanh ghi \$s6 và \$s7.

Lệnh	Giải thích	Kết quả
addi \$t0, \$s6, 4	Dịch 4 byte của mảng A[0] lưu vào \$t0	$\$t0 = \&A[1]$
add \$t1, \$s6, \$0	Lấy chỉ mảng A[0] lưu vào \$t1	$\$t1 = \&A[0]$
sw \$t1, 0(\$t0)	Giá trị \$t0 cho \$t1	$\$t1 = \&A[0] = A[1]$
lw \$t0, 0(\$t0)	Lưu giá trị \$t0 vào \$t0	$\$t0 = A[1]$
add \$s0, \$t1, \$t0	Lấy \$t1 cộng \$t0 lưu vào f	$f = A[0] + A[1]$

Câu 9. Giả sử rằng các thanh ghi \$s0 và \$s1 giữ các giá trị tương ứng là 0x80000000 và 0xD0000000.

a. Giá trị của \$t0 cho mã hợp ngữ sau đây là bao nhiêu?

add \$t0, \$s0, \$s1

$\$t0 = \$s0 + \$s1$

$= 0x80000000 + 0xD0000000$

$$= 0xF00000000$$

- b. Đối với nội dung của các thanh ghi \$s0 và \$s1 như được chỉ định ở trên, giá trị của \$t0 cho mã hợp ngữ sau đây là bao nhiêu?**

sub \$t0, \$s0, \$s1

$$\begin{aligned} \$t0 &= \$s0 - \$s1 \\ &= 0x80000000 - 0xD0000000 \\ &= 0xFB0000000 \end{aligned}$$

- c. Đối với nội dung của các thanh ghi \$s0 và \$s1 như được chỉ định ở trên, giá trị của \$t0 cho mã hợp ngữ sau đây là bao nhiêu?**

add \$t0, \$s0, \$s1

add \$t0, \$t0, \$s0

$$\begin{aligned} \$t0 &= \$s0 + \$s1 \\ &= 0x80000000 + 0xD0000000 \\ &= 0xF00000000 \\ \$t0 &= \$t0 + \$s0 \\ &= 0xF00000000 + 0x80000000 \\ &= 0xF80000000 \end{aligned}$$

Câu 10. Cung cấp hướng dẫn về loại và hợp ngữ cho giá trị nhị phân sau:
0000 0010 0001 0000 1000 0000 0010 0000two

000000 10000 10000 10000 00000 100000

Op : 0

Function : 20

⇒ Thanh ghi R lệnh add

Rs : 16

Rt : 16

Rd : 16

Add \$s0, \$s0, \$s0

Câu 11. Cung cấp loại và biểu diễn thập lục phân của lệnh sau: sw \$t1, 32(\$t2)

Lệnh thanh ghi I

Op(2b) : 101011
 Rt(\$t2/10) : 01010
 Rs(\$t1/9) : 01001
 Constant of address(32) : 0000 0000 0010 0000
 1010 1101 0100 1001 0000 0000 0010 0000
 0XAD490020

Câu 12. Cung cấp loại, hướng dẫn hợp ngữ và biểu diễn nhị phân của hướng dẫn được mô tả bởi các trường MIPS sau:

op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

op (0) : 000000

funct(22hec): 100010

➔ Loại thanh ghi R lệnh sub

Rs(3) : 00011/\$v1

Rt(2) : 00010/\$v0

Rd(3) : 00011/\$v1

0000 0000 0110 0010 0001 1000 0010 0010

0X00621822

Sub \$v1, \$v1,\$v0

Câu 13. Cung cấp loại, hướng dẫn hợp ngữ và biểu diễn nhị phân của hướng dẫn được mô tả bằng các trường MIPS sau:

op=0x23, rs=1, rt=2, const=0x4

⇒ Loại thanh ghi I lệnh Lw

Op(23hec) : 100011

Const : 0000 0000 0000 0010

Rs(1) : 00001

Rt(2) : 00010

1000 1100 0010 0010 0000 0000 0000 0010

0X8C220004

Lw \$v0, 4(\$at)

Câu 14. 2.19 Giả sử nội dung thanh ghi sau:

\$t0 = 0xAAAAAAAA, \$t1 = 0x12345678

- a. Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của \$t2 cho chuỗi lệnh sau là bao nhiêu?

0XAAAAAAAA

= 1010 1010 1010 1010 1010 1010 1010 1010

0X12345678

= 0001 0010 0011 0100 0101 0110 0111 1000

sll \$t2, \$t0, 44

Dịch sang trái

\$t2 = 0000 0000 0001 0101 0101 0101 0101 0101

or \$t2, \$t2, \$t1

\$t2 = 0000 0000 0001 0101 0101 0101 0101 0101

Hoặc

\$t2 = 0001 0010 0011 0100 0101 0110 0111 1000

- b. Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của \$t2 cho chuỗi lệnh sau là bao nhiêu?

sll \$t2, \$t0, 4

addi \$t2, \$t2, -1

- c. Đối với các giá trị thanh ghi được hiển thị ở trên, giá trị của \$t2 cho chuỗi lệnh sau là bao nhiêu?

srl \$t2, \$t0, 3

addi \$t2, \$t2, 0xFFEF

- d. Giả sử \$t0 giữ giá trị 0x00101000. Giá trị của \$t2 sau các hướng dẫn sau là bao nhiêu?

\$t0 = 0000 0000 0001 0000 0001 0000 0000 0000

\$t0 = 1 052 672

slt \$t2, \$0, \$t0

Dịch 0 sang trái $2^{(1\ 052\ 672)}$ byte và lưu vào \$t2

\$t2 = 0000 0000 0000 0000 0000 0000 0000 0000

bne \$t2, \$0, ELSE

\$t2 != 0 nếu:

- Đúng: nhảy tới else
- Sai tiếp tục câu lệnh tiếp dưới.

\$t2=0 → đúng

j DONE

ELSE: addi \$t2, \$t2, 2

$\$t2 = \$t2 + 2 = 2$

DONE:

Câu 15. Chuyển đổi đoạn mã MIPS sang ngôn ngữ C?

MIPS:

Bne \$s3, \$s3, Else

Add \$s0, \$s1, \$s2

J Exit:

Else: Sub \$s0, \$s1, \$s2

Exit:

Biết f,g,h,i và j là năm biến. Năm biến f đến j lần lượt tương ứng với 5 thanh ghi từ \$s0 đến \$s4.

Bne \$s3, \$s4, Else If (i!=j) đúng Add \$s0, \$s1, \$s2 $f = g + h$ J Exit: Thoát vòng if Ngược lại Else: Sub \$s0, \$s1, \$s2 $f = g - h$ Thoát Exit:	Ngôn ngữ C If (i!=j) { $f = g + h$; } Else $f = g - h$;
---	--

Câu 16. Dịch đoạn Code C sau sang mã MIPS

Code C:

While (save[i] == k)

{ i+=1; }

Biết biến i và k tương ứng với thanh ghi \$s3 và \$s5, địa chỉ cơ sở của mảng save lưu trong thanh ghi \$s6.

Acssembly MIPS	Giải thích
Loop: Sll \$t1,\$s3,2	Dịch i sang trái 4 byte để lấy địa chỉ của i trong mảng save
Add \$t0, \$t0, \$s6	Cộng địa chỉ của i với địa chỉ cơ sở của mảng save để có được địa chỉ cần tìm của Save[i]
Lw \$t0, 0(\$t1)	Load dữ liệu tại địa chỉ Save[i] vào thanh ghi \$t0
Bne \$t0, \$s5, Exit	So sánh xem \$t0 != k không, nếu bằng tới exit, ngược lại tới câu lệnh tiếp theo
Addi \$s3, \$s3, 1 J Loop:	Tính $i = i + 1$. Lặp
Exit	Thoát

Câu 17. Đổi mã Asscemly MIPS sau sang mã máy

MIPS: Add \$s1,\$s2,\$s3

Op : 00 0000
Rs : 1 0010
Rt : 1 0011
Rd : 1 0001
Shamt : 0 0000
Func : 10 0000

Mã máy: 0000 0010 0101 0011 1000 1000 0010 0000

⇒ 0x02538820

MIPS: Beq \$s1, \$s2, Address (Biết Beq có địa chỉ 0x00400010 và địa chỉ của Address 0x00400000)

$\text{if}(R[\text{rs}] == R[\text{rt}]) \text{PC} = \text{PC} + 4 + \text{BranchAddr}$

BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }

Op : 00 0100

Rs : 1 0001

Rt : 1 0010

Constant :

Ox00400000 = Ox00400010 + 4 + { 14{immediate[15]}, immediate, 2'b0 }

⇒ { 14{immediate[15]}, immediate, 2'b0 } = 0000 0000 0100 0000 0000 0000
0000 0000

– 0000 0000 0100 0000 0000 0000 0001 0000 - 0000 0000 0000 0000 0000 0000
0000 0100

⇒ { 14{immediate[15]}, immediate, 2'b0 } = 1111 1111 1111 1111 1111 1111
1110 1100

⇒ **Immediate** = 1111 1111 1111 1111 1011

Mã máy: 0001 0010 0011 0010 1111 1111 1111 1011

⇒ **Ox1232FFFB**

MIPS: Slti \$s1, \$s2, -19

R[rt] = (R[rs] < SignExtImm)? 1 : 0

Op : 00 1010

Rs : 1 0001

Rt : 1 0010

Constant : 1111 1111 1110 1101

Mã máy: 0010 1010 0011 0010 1111 1111 1110 1101

⇒ **Ox2A32FFED**

Jal Label (Biết Label có địa Ox00400000 và Jal có địa chỉ Ox00400014)
--

PC=JumpAddr

JumpAddr = { PC+4[31:28], address, 2'b0 }

Op : 00 0011

Address :

Ox00400000 = { PC+4[31:28], address, 2'b0 }

⇒ Ox00400000 = { Ox00400014 + 4[31:28], address, 2'b0 }

⇒ Ox00400000 = { Ox00400018[31:28], address, 2'b0 }

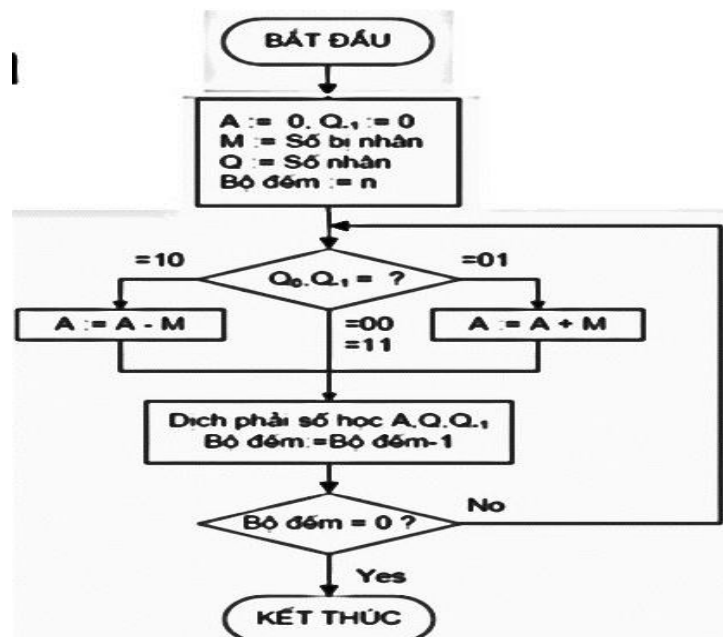
⇒ $\text{Ox00400000} = \{4'b0, \text{address}, 2'b0\}$

⇒ $0000\ 0000\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000 = \{4'b0, \text{address}, 2'b0\}$

⇒ Address = $0000\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000\ 00$

Mã máy: $0000\ 1100\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000$

⇒ Ox0C100000



Step	Thực hiện	Q1	Q0	A
0	Khởi tạo	0		0000 0000
...				

Lưu ý: $A-M = A+(-M)$

D. BIỂU DIỄN DẠNG THỪA K

Ta cần biểu diễn số -B dạng thừa K.

- Bước 1: Chọn giá trị $K = \text{const}$ ($-128 \leq K < 127$).
- Bước 2: Tính giá trị $C = K + (-B)$.
- Bước 3: Biểu diễn số C dưới dạng chuỗi nhị phân.

II. PHÉP CỘNG

Quy tắc	$0 + 0 = 0$
	$0 + 1 = 1$
	$1 + 0 = 1$
	$1 + 1 = 1 \text{ (nhớ 1)}$

Lưu ý:

- Thực hiện phép cộng từ phải sang trái (hàng thứ 0 cho đến hàng n).
- Số nhớ ở hàng cộng thứ i sẽ được cộng vào hàng cộng thứ i+1.

III. PHÉP TRỪ

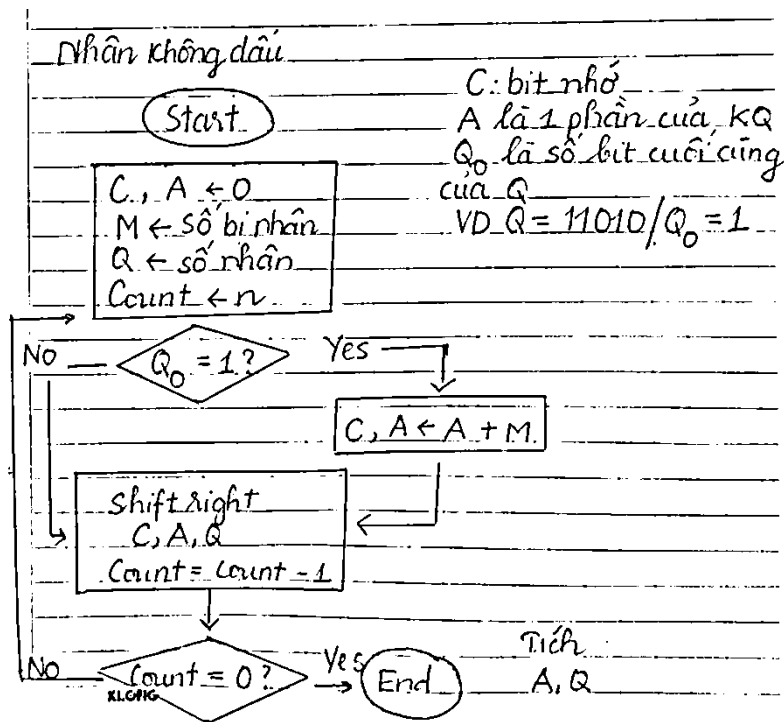
Quy tắc	$0 - 0 = 0$
	$1 - 1 = 0$
	$1 - 0 = 1$
	$0 - 1 = 1 \text{ (mượn 1)}$

Lưu ý:

- Thực hiện phép trừ từ phải sang trái (hàng thứ 0 cho đến hàng thứ n).
- Số mượn ở hàng thứ i sẽ được cộng vào cho số trừ ở hàng thứ i+1.

IV. PHÉP NHÂN

Nhân 2 số 32 bit:



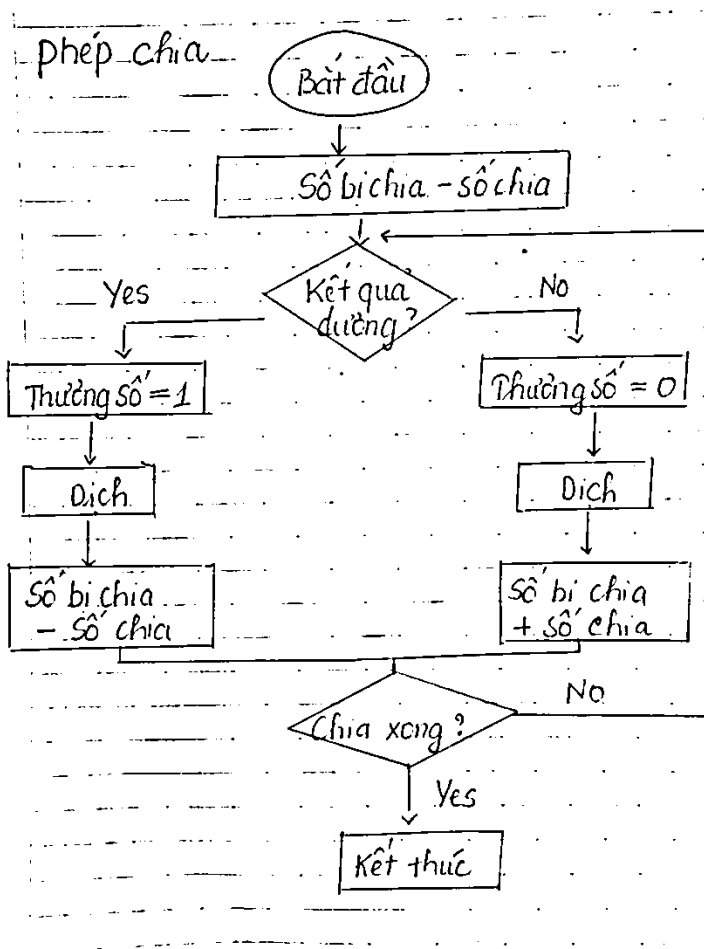
Chú ý:

Khi thực hiện phép nhân cho giải thuật theo sơ đồ, ta thấy có 3 bước và 3 bước này được lặp lại 32 lần. Nếu mỗi bước được thực hiện bởi 1 chu kỳ xung clock cho phép toán nhân hai số 32 bit.

Step	Bước thực hiện	Số bị nhân	Số nhân	Tích
0	Khởi tạo			
1	1. a=0			
	2. Dịch trái số nhân			
	3. Dịch phải số bị nhân			
2	1. a=1, Tích = Tích + Số nhân			
	2. Dịch trái số nhân			
	3. Dịch phải số bị nhân			
...				

I. PHÉP CHIA

Giải thuật của phép chia:



Chú ý:

Nếu phép chia có dấu

- Bước 1: Bỏ qua dấu, thực hiện phép chia thông thường.
- Bước 2: Xét dấu

+ Dấu của thương sẽ trái với dấu hiện tại nếu dấu của số chia và số bị chia ngược nhau.

+ Dấu của số dư:

Số bị chia = Thương X Số chia + Số dư

⇒ Số dư = Số bị chia – (Thương X số chia)

Phép chia tròn MIPS: Sau khi thực hiện phép chia xong

- Thanh ghi Hi chứa phần dư.
- Thanh ghi Lo chứa thương số.

Step	Thực hiện	Thương	Số chia	Số dư
------	-----------	--------	---------	-------

		(Q)		
0	Khởi tạo			
1	1.Số dư = số dư – số chia			
	2.Số dư < 0, Khôi phục số dư, Q0=0			
	3. Dịch phải số chia			
2	1.Số chia=số chia-số bị chia			
	2.Số dư > 0, Sll Q, Q0=1			
	3.Dịch phải số chia			
...				

II. SỐ THỰC DẤU CHẤM ĐỘNG

Định nghĩa: Trong máy tính, các số nhị phân phải được đưa về dạng chuẩn như sau:

$$1. xxxxxxxx_{10} \times 2^{yyyy}$$

Biểu diễn số thực dấu chấm động (với Bias = 127)

$$(-1)^S \times (1 + Fraction) \times 2^{(Exponent - Bias)}$$

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
S	Exponent										Fraction																				

Trong đó:

- S biểu diễn dấu của số thực dấu chấm động (1 nghĩa là âm, 0 nghĩa là dương).
- Phần mũ (Exponent) có kích thước 8 bit.
- Phần lẻ (Fraction) dùng 23 bit để biểu diễn cho .xxxxxxxx

Chú ý:

- Tràn trên (Overflow): Trường hợp này xảy ra khi kích thước của số mũ lớn hơn kích thước giới hạn trên (số mũ dương).
- Tràn dưới (Underflow): Trường hợp này xảy ra khi kích thước của số mũ nhỏ hơn kích thước giới hạn dưới (số mũ âm).
- Cách biểu diễn IEEE 754:
 - + Độ chính xác đơn (Single precision): Một số thực dấu chấm động được biểu diễn ở dạng 32 bit (bias of 127).

- + Độ chính xác kép (Double precision): Một số thực dấu chấm động được biểu diễn ở dạng 64 bit (bias of 1023).

BÀI TẬP

Câu 1. Tính 5ED4 - 07A4 là gì khi các giá trị này biểu thị số thập lục phân 16 bit không dấu? Kết quả phải được viết bằng hệ thập lục phân. Chỉ ra công việc của bạn.

5ED4=0101 1110 1101 0100

07A4=0000 0111 1010 0100

0101 1110 1101 0100

0000 0111 1010 0100 -

0101 0111 0011 0000

⇒ Vậy: 5ED4 – 07A4 = 5730.

Câu 2. Tính 5ED4 - 07A4 là gì khi các giá trị này biểu thị các số thập lục phân 16 bit có dấu được lưu trữ ở định dạng cường độ dấu? Kết quả phải được viết dưới dạng thập lục phân. Chỉ ra công việc của bạn.

5ED4=0101 1110 1101 0100 (bit dấu =0 → số dương)

07A4=0000 0111 1010 0100 (bit dấu =0 → số dương)

0101 1110 1101 0100

- 0000 0111 1010 0100

0101 0111 0011 0000 (bit dấu =0 → số dương)

⇒ Vậy: 5ED4 – 07A4 = 5730.

Câu 4. Chuyển 5ED4 thành số nhị phân. Điều gì làm cho cơ số 16 (thập lục phân) trở thành một hệ thống đánh số hấp dẫn để biểu diễn các giá trị trong máy tính?

- Số nhị phân có thể đổi được sang hệ thập lục phân đôi chút dễ dàng hơn. Sự dễ dàng này là do gốc của hệ thập lục phân (16) là số mũ của gốc hệ nhị phân (2). Cụ thể hơn $16 = 2^4$. Vậy chúng ta phải cần 4 ký tự số trong hệ nhị phân để có thể biểu đạt được một ký tự số trong hệ thập lục phân.

CÁCH 1: Chuyển đổi trực tiếp.

5(hex) = 0101(bin).

E(hex) = 14 = 1110(bin).

D(hex) = 13 = 1101(bin).

4(hex) = 0100(bin).

⇒ Vậy 5ED4 = 0101 1110 1101 0100.

CÁCH 2: Chuyển đổi gián tiếp.

$$5ED4(\text{hex}) = [(5 \times 16^3) + (14 \times 16^2) + (13 \times 16^1) + (4 \times 16^0)](\text{dec}) = 24\,276(\text{dec}).$$

$$24\,276(\text{dec}) = 0101\,1110\,1101\,0100.$$

$$\Rightarrow \text{Vậy } 5ED4 = 0101\,1110\,1101\,0100.$$

Câu 5. 4365 - 3412 là gì khi các giá trị này biểu thị số bát phân 12 bit không dấu? Kết quả phải được viết bằng bát phân. Chỉ ra công việc của bạn.

$$4365(\text{Oct}) = 0000\,1000\,1111\,0101(\text{Bin}).$$

$$3412(\text{Oct}) = 0000\,0111\,0000\,1010(\text{Bin}).$$

$$0000\,1000\,1111\,0101$$

$$0000\,0111\,0000\,1010 \quad -$$

$$0000\,0001\,1110\,1011$$

$$\text{Vậy } 4365 - 3412 = 0753$$

Câu 6. 4365 - 3412 là gì khi các giá trị này biểu thị số bát phân 12 bit có dấu được lưu trữ ở định dạng cường độ dấu? Kết quả phải được viết dưới dạng bát phân. Chỉ ra công việc của bạn.

$$4365(\text{Oct}) = 0000\,1000\,1111\,0101(\text{Bin}) \text{ (bit dấu bằng 0 } \rightarrow \text{ Số dương).}$$

$$3412(\text{Oct}) = 0000\,0111\,0000\,1010(\text{Bin}) \text{ (bit dấu bằng 0 } \rightarrow \text{ Số dương).}$$

$$0000\,1000\,1111\,0101$$

$$- 0000\,0111\,0000\,1010$$

$$0000\,0001\,1110\,1011 \text{ (bit dấu bằng 0 } \rightarrow \text{ Số dương).}$$

$$\Rightarrow \text{Vậy } 4365 - 3412 = 0753$$

Câu 7. Giả sử 185 và 122 là số nguyên thập phân 8 bit không dấu. Tính 185 - 122. Có tràn, thiếu hay không?

$$185(\text{Dec}) = 1011\,1001(\text{Bin}).$$

$$122(\text{Dec}) = 0111\,1010(\text{Bin})$$

$$1011\,1001$$

$$0111\,1010 \quad -$$

$$0011\,1111$$

$$\Rightarrow \text{Vậy: } 185 - 122 = 63 \text{ (không tràn hoặc thiếu).}$$

Câu 8. Giả sử 185 và 122 là số nguyên thập phân 8 bit có dấu được lưu trữ ở định dạng cường độ dấu. Tính 185+122. Có tràn, tràn dưới hay không?

$$185(\text{Dec}) = 1011\,1001(\text{Bin}).$$

$$122(\text{Dec}) = 0111\,1010(\text{Bin})$$

$$1011\,1001$$

+

0111 1010

$$\begin{array}{r} 1\ 0011\ 0011 \\ \hline \end{array}$$

⇒ Vậy: $185 + 122 = 307$ (Do Max-8-bit = 255 → Tràn).

Câu 9. Giả sử 185 và 122 là số nguyên thập phân 8 bit có dấu được lưu trữ ở định dạng cường độ dấu. Tính $185 - 122$. Có tràn, thiếu hay không?

$185(\text{Dec}) = 1011\ 1001(\text{Bin})$.

$122(\text{Dec}) = 0111\ 1010(\text{Bin})$.

⇒ $-122(\text{Dec}) = 1\ 0111\ 1010(\text{Bin bit dấu})$.

⇒ Vậy $185 - 122 = 185 + (-122)$.

1011 1001

$$\begin{array}{r} +\quad 1\ 0111\ 1010 \\ \hline \end{array}$$

10 0011 0011 (bit dấu bằng 0 → số dương)

⇒ Vậy: $185 - 122 = 63$ (Bị thiếu).

Câu 10. Giả sử 151 và 214 là số nguyên thập phân 8 bit có dấu được lưu ở định dạng phần bù hai. Tính $151 + 214$ bằng số học bão hòa. Kết quả phải được viết dưới dạng thập phân. Chỉ ra công việc của bạn.

$151(\text{Dec}) = 1001\ 0111(\text{Bin-Bù } 2)$

⇒ Bin-Bù 1 = 1001 0110

⇒ Bin = 0110 1001

$214(\text{Dec}) = 1101\ 0110(\text{Bin-Bù } 2)$

⇒ Bin-Bù 1 = 1101 0101

⇒ Bin = 0010 1010

0110 1001

$$\begin{array}{r} +\quad 0010\ 1010 \\ \hline \end{array}$$

1001 0011

⇒ Bin - Bù 1 = 0110 1100

⇒ Bin - Bù 2 = 0110 1101

⇒ $151(\text{Dec}) + 214(\text{Dec}) = 365(\text{Dec})$

Câu 11. Giả sử 151 và 214 là số nguyên thập phân 8 bit có dấu được lưu ở định dạng phần bù hai. Tính $151 - 214$ bằng số học bão hòa. Kết quả phải được viết dưới dạng thập phân. Chỉ ra công việc của bạn.

$151(\text{Dec}) = 1001\ 0111(\text{Bin-Bù } 2)$

⇒ Bin-Bù 1 = 1001 0110

⇒ Bin = 0110 1001

$$214(\text{Dec}) = 1101\ 0110(\text{Bin-Bù } 2)$$

$$\Rightarrow \text{Bin-Bù } 1 = 1101\ 0101$$

$$\Rightarrow \text{Bin} = 0010\ 1010$$

$$\begin{array}{r} 0110\ 1001 \\ - 0010\ 1010 \\ \hline 0011\ 1111 \end{array}$$

$$\Rightarrow \text{Bin-Bù } 1 = 1100\ 0000$$

$$\Rightarrow \text{Bin-Bù } 2 = 1100\ 0001$$

$$\Rightarrow \text{Vậy } 151(\text{Dec}) - 214(\text{Dec}) = -63(\text{Dec})$$

Câu 12. Giả sử 151 và 214 là số nguyên 8 bit không dấu. Tính 151+ 214 bằng số học bão hòa. Kết quả phải được viết dưới dạng thập phân. Chỉ ra công việc của bạn.

Lấy $k = -150$;

$$151(\text{Dec}) = 0000\ 0001(\text{Bin-Thừa } k = -150)$$

$$214(\text{Dec}) = 0100\ 0000(\text{Bin-Thừa } k = -150)$$

$$\begin{array}{r} 0000\ 0001 \\ + 0100\ 0000 \\ \hline 0100\ 0001 \end{array}$$

$$\Rightarrow 0100\ 0001(\text{Bin-Thừa } k' = -300) \text{ biểu diễn cho số } 365(\text{Dec}).$$

$$\Rightarrow \text{Vậy } 151(\text{Dec}) + 214(\text{Dec}) = 365(\text{Dec}).$$

Câu 13. Sử dụng bảng tương tự như trong Hình 3.6, tính tích của các số nguyên 6 bit không dấu bát phân 62 và 12 bằng phần cứng được mô tả trong Hình 3.3. Bạn nên hiển thị nội dung của mỗi thanh ghi trên mỗi bước.

$$62_{\text{oct}} = 110\ 010_{\text{bin}} (\text{Số bị nhân}).$$

$$12_{\text{oct}} = 001\ 010_{\text{bin}} (\text{Số nhân}).$$

$$\begin{array}{r} 110\ 010 \\ 001\ 010 \\ \hline 000\ 000 \\ 1\ 100\ 100 \\ 00\ 000\ 000 \\ 110\ 010\ 000 \\ 0\ 000\ 100\ 000 \\ 00\ 000\ 000\ 000 \\ \hline 00\ 111\ 110\ 100 \end{array}$$

Step	Bước thực hiện	Số bị nhân	Số nhân	Tích
------	----------------	------------	---------	------

0	Khởi tạo	110 010	000 000 001 010	000 000 000 000
1	1. a=0	110 010	000 000 001 010	000 000 000 000
	2. Dịch trái số nhân	110 010	000 000 010 100	000 000 000 000
	3. Dịch phải số bị nhân	011 001	000 000 010 100	000 000 000 000
2	1. a=1, Tích = Tích +Số nhân	011 001	000 000 010 100	000 000 010 100
	2. Dịch trái số nhân	011 001	000 000 101 000	000 000 010 100
	3. Dịch phải số bị nhân	001 100	000 000 101 000	000 000 010 100
3	1. a=0	001 100	000 000 101 000	000 000 010 100
	2. Dịch trái số nhân	001 100	000 001 010 000	000 000 010 100
	3. Dịch phải số bị nhân	000 110	000 001 010 000	000 000 010 100
4	1. a=0	000 110	000 001 010 000	000 000 010 100
	2. Dịch trái số nhân	000 110	000 010 100 000	000 000 010 100
	3. Dịch phải số bị nhân	000 011	000 010 100 000	000 000 010 100
5	1. a=1, Tích = Tích +Số nhân	000 011	000 010 100 000	000 010 110 100
	2. Dịch trái số nhân	000 011	000 101 000 000	000 010 110 100
	3. Dịch phải số bị nhân	000 001	000 101 000 000	000 010 110 100
6	1. a=1, Tích = Tích +Số nhân	000 001	000 101 000 000	000 111 110 100
	2. Dịch trái số nhân	000 001	001 010 000 000	000 111 110 100
	3. Dịch phải số bị nhân	000 000	001 010 000 000	000 111 110 100

Câu 14. Sử dụng bảng tương tự như trong Hình 3.6, tính tích của các số nguyên 8 bit không dấu thập lục phân 62 và 12 bằng cách sử dụng phần cứng được mô tả trong Hình 3.5. Bạn nên hiển thị nội dung của mỗi thanh ghi trên mỗi bước.

62dec = 0011 1110bin (số nhân)

12dec = 0000 1100bin (số bị nhân)

```

          0011 1110
          0000 1100
          -----
          0000 0000
        0 0000 0000
        00 1111 1000
        001 1111 0000
        0000 0000 0000
        0 0000 0000 0000
        00 0000 0000 0000
        000 0000 0000 0000
        -----
        000 0010 1110 1000

```

Step	Bước thực hiện	Số bị nhân	Số nhân	Tích
0	Khởi tạo	0000 1100	0000 0000 0011 1110	0000 0000 0000 0000
1	1. a=0	0000 1100	0000 0000 0011 1110	0000 0000 0000 0000
	2. Dịch trái số nhân	0000 1100	0000 0000 0111 1100	0000 0000 0000 0000
	3. Dịch phải số bị nhân	0000 0110	0000 0000 0111 1100	0000 0000 0000 0000
2	1. a=0	0000 1100	0000 0000 0011 1110	0000 0000 0000 0000

	2. Dịch trái số nhân	0000 1100	0000 0000 0111 1100	0000 0000 0000 0000
	3. Dịch phải số bị nhân	0000 0011	0000 0000 1111 1000	0000 0000 0000 0000
3	1.a=1, Tích = Tích + Số nhân	0000 0011	0000 0000 1111 1000	0000 0000 1111 1000
	2. Dịch trái số nhân	0000 0011	0000 0001 1111 0000	0000 0000 1111 1000
	3. Dịch phải số bị nhân	0000 0001	0000 0001 1111 0000	0000 0000 1111 1000
4	1.a=1, Tích = Tích + Số nhân	0000 0001	0000 0001 1111 0000	0000 0010 1110 1000
	2. Dịch trái số nhân	0000 0001	0000 0011 1110 0000	0000 0010 1110 1000
	3. Dịch phải số bị nhân	0000 0000	0000 0011 1110 0000	0000 0010 1110 1000
5	1.a=0	0000 0000	0000 0011 1110 0000	0000 0010 1110 1000
	2. Dịch trái số nhân	0000 0000	0000 0111 1100 0000	0000 0010 1110 1000
	3. Dịch phải số bị nhân	0000 0000	0000 0111 1100 0000	0000 0010 1110 1000
6	1.a=0	0000 0000	0000 0111 1100 0000	0000 0010 1110 1000
	2. Dịch trái số nhân	0000 0000	0000 1111 1000 0000	0000 0010 1110 1000
	3. Dịch phải số bị nhân	0000 0000	0000 1111 1000 0000	0000 0010 1110 1000
7	1.a=0	0000 0000	0000 1111 1000 0000	0000 0010 1110 1000
	2. Dịch trái số nhân	0000 0000	0001 1110 0000 0000	0000 0010 1110 1000
	3. Dịch phải số bị nhân	0000 0000	0001 1110 0000 0000	0000 0010 1110 1000
8	1.a=0	0000 0000	0001 1110 0000 0000	0000 0010 1110 1000
	2. Dịch trái số nhân	0000 0000	0011 1100 0000 0000	0000 0010 1110 1000
	3. Dịch phải số bị nhân	0000 0000	0011 1100 0000 0000	0000 0010 1110 1000

Câu 15. Biểu diễn giá trị 6.015625 thành số nhị phân bằng dấu chấm động?

1. $xxxxxxxx_{10} \times 2^{yyyy}$

Biểu diễn số thực dấu chấm động (với Bias = 127)

$$(-1)^S \times (1 + Fraction) \times 2^{(Exponent - Bias)}$$

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0										
S	Exponent 8 bits								Fraction 23 bits																						

Trong đó:

- S biểu diễn dấu của số thực dấu chấm động (1 nghĩa là âm, 0 nghĩa là dương).
- Phần mũ (Exponent) có kích thước 8 bit.
- Phần lẻ (Fraction) dùng 23 bit để biểu diễn cho .xxxxxxxx

- Ta có:

6dec = 0110

0.015625dec = 000001

$$0.015625 \times 2 = 0.03125$$

$$0.03125 \times 2 = 0.0625$$

$$0.0625 \times 2 = 0.125$$

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

➔ $6.015625_{\text{dec}} = 0110.000001_{\text{bin}}$

Đưa về dạng chuẩn: 1.10000001×2^2

- Tìm E:

$$E - \text{bias} = 2$$

$$\Rightarrow E = 2 + 127 = 129$$

$$129_{\text{dec}} = 1000\ 0001$$

$$\mathbf{6.015625} = 0\ 1000\ 0001\ 1000\ 0001\ 0000\ 0000\ 0000\ 000$$

Câu 16. Thực hiện phép chia cho 2 số 4 bit sau:

$$7_{10} : 2_{10}$$

Ta có

$$7_{10} = 0111$$

$$2_{10} = 0010$$

Step	Thực hiện	Thương	Số chia	Số dư
0	Khởi tạo	0000	0010 0000	0000 0111
1	1.Số dư = số dư – số chia	0000	0010 0000	1110 0111
	2.Số dư < 0, Khôi phục số dư, Q0=0	0000	0010 0000	0000 0111
	3. Dịch phải số chia	0000	0001 0000	0000 0111
2	1.Số dư = số dư – số chia	0000	0001 0000	1111 0111
	2. Số dư < 0, Khôi phục số dư, Q0=0	0000	0001 0000	0000 0111
	3.Dịch phải số chia	0000	0000 1000	0000 0111
3	1.Số dư = số dư – số chia	0000	0000 1000	1111 1111
	2.Số dư < 0, Khôi phục số dư, Q0=0	0000	0000 1000	0000 0111
	3. Dịch phải số chia	0000	0000 0100	0000 0111
4	1.Số chia=số chia-số bị chia	0000	0000 0100	0000 0011

	2.Số dư >0, Q0=1	0001	0000 0100	0000 0011
	3.Dịch phải số chia	0001	0000 0010	0000 0011
	1.Số dư=Số dư-Số chia	0001	0000 0010	0000 0001
5	2.Số dư >0, Q0=0	0011	0000 0010	0000 0001
	3.Dịch phải số chia	0011	0000 0001	0000 0001

Vậy: $0111 : 0010 = 0011$ (dư 0001)

Câu 16. Thực hiện phép nhân hai số thực dấu chấm động sau:

$$(1.110_{ten} \times 10^{10}) \times (92.00_{ten} \times 10^{-6})$$

Đưa về dạng chuẩn: $(1.110_{ten} \times 10^{10}) \times (9.200_{ten} \times 10^{-5})$

Step	Content
1	Tính số mũ mới: $= 10 + (-5) = 5$
2	$ \begin{array}{r} 1.110 \\ 9.200 \\ \hline 0000 \\ 00000 \\ 222000 \\ 9990000 \\ \hline 10212000 \end{array} $
Kết quả	$ \begin{aligned} &(1.110_{ten} \times 10^{10}) \times (9.200_{ten} \times 10^{-5}) \\ &= 10.212000 \times 10^5 \end{aligned} $

Câu 17. Thực hiện phép tính $(-19_{10} \times 20_{10})$

Ta có:

$$M = -19 = 101101$$

$$-M = 19 = 010011$$

$$Q = 20 = 010100$$

Thực hiện	A	Q (Q0)	Q1
Khởi tạo	000000	010100	0
N=6, Q0Q1=00 Dịch phải	000000	001010	0
N=5, Q0Q1=00 Dịch phải	000000	000101	0
N=4, Q0Q1=10 A=A+(-M) Dịch phải	010011 001001	000010	1
N=3, Q0Q1=01 A=A+M Dịch phải	110110 111011	100001	0
N=2, Q0Q1=10 A=A+(-M) Dịch phải	1 001110 000111	001000	1
N=1, Q0Q1=01 A=A+M Dịch phải	110100 111010	000100	0
N=0, Stop	111010	000100	0

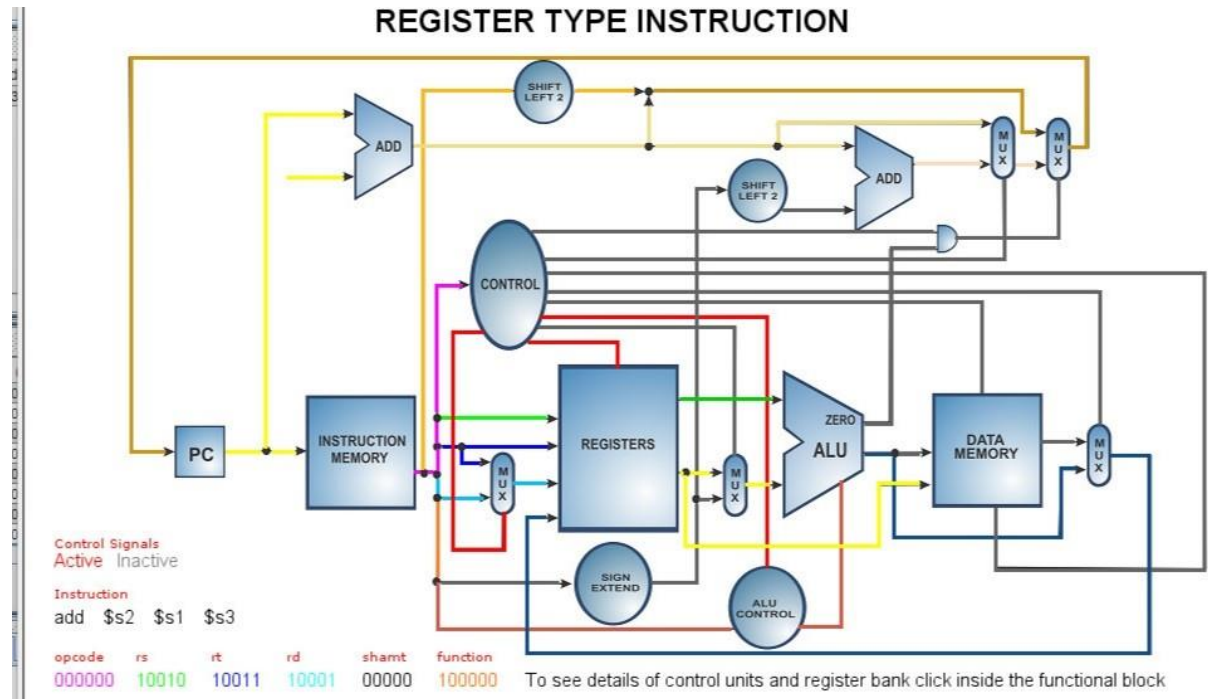
Do âm X dương ➔ bits trái nhất là 1

Kết quả: 1 11010 000100

CHƯƠNG 6. BỘ XỬ LÝ VÀ KỸ THUẬT ỚNG DẪN

Câu 1. Vẽ sơ đồ datapath của lệnh Add rd, rt, rs

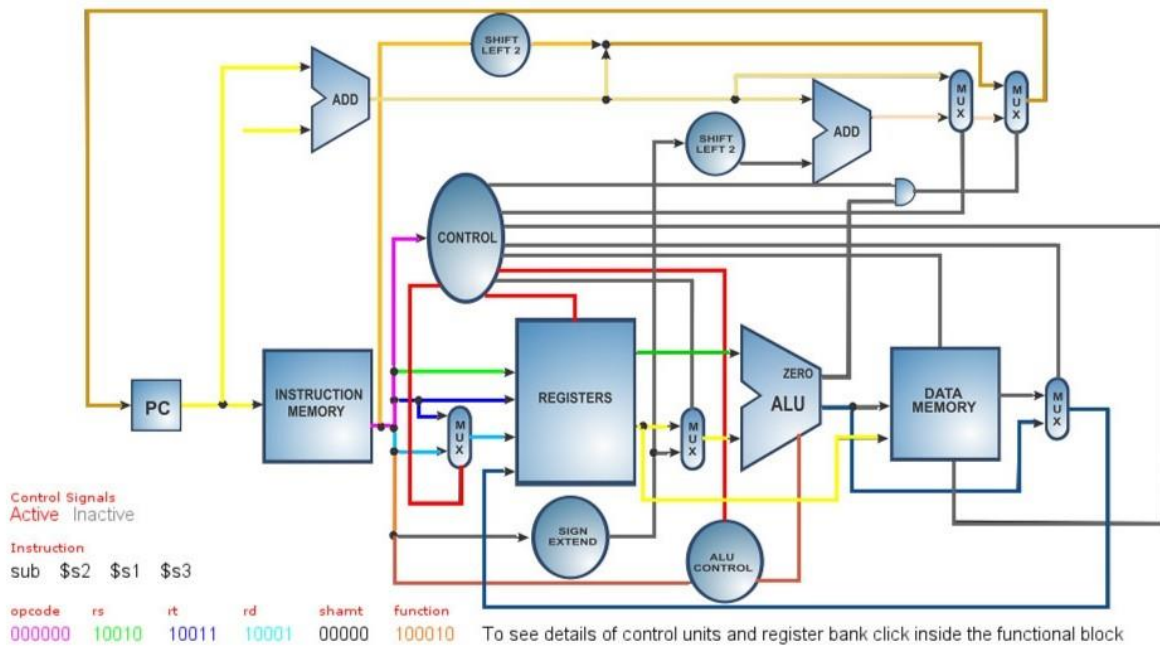
Add \$s1, \$s2, \$s3 ($s1 = s2 + s3$)



Câu 2. Vẽ sơ đồ datapath của lệnh Sub rd, rt, rs

Sub \$s1, \$s2, \$s3 (\$s1 = \$s2 - \$s3)

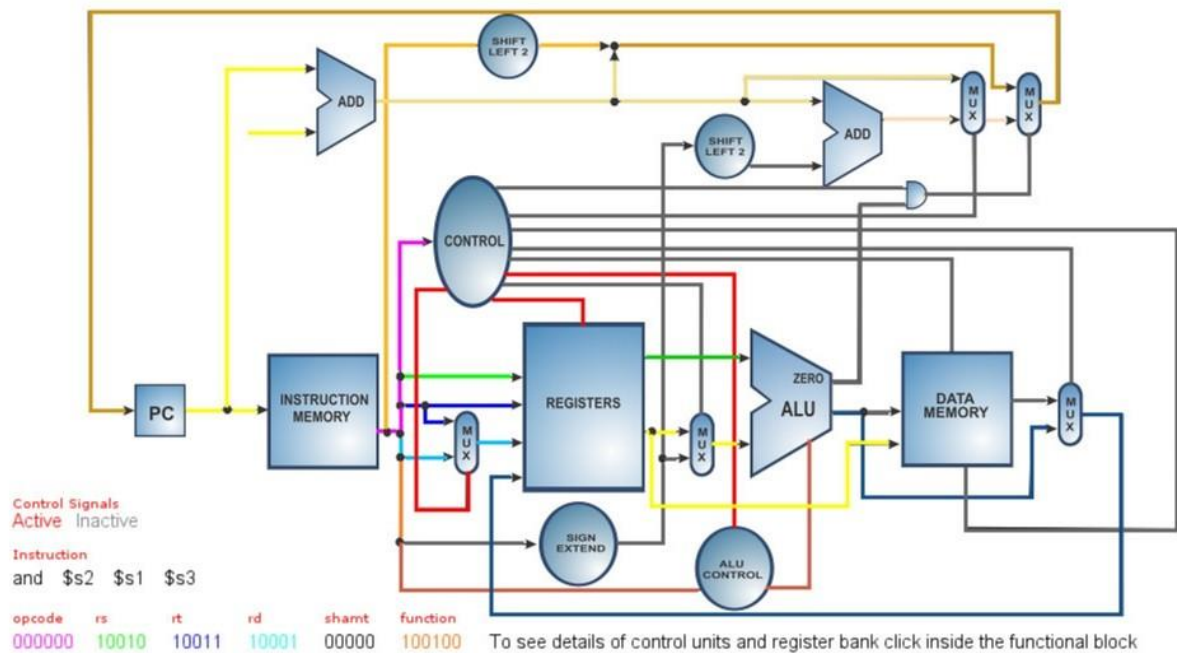
REGISTER TYPE INSTRUCTION



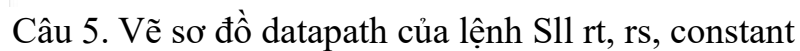
Câu 3. Vẽ sơ đồ datapath của lệnh And rd, rt, rs

And \$s1, \$s2, \$s3 (\$s1 = \$s2 and \$s3)

REGISTER TYPE INSTRUCTION



Or s_1, s_2, s_3 ($s_1 = s_2 \parallel s_3$)

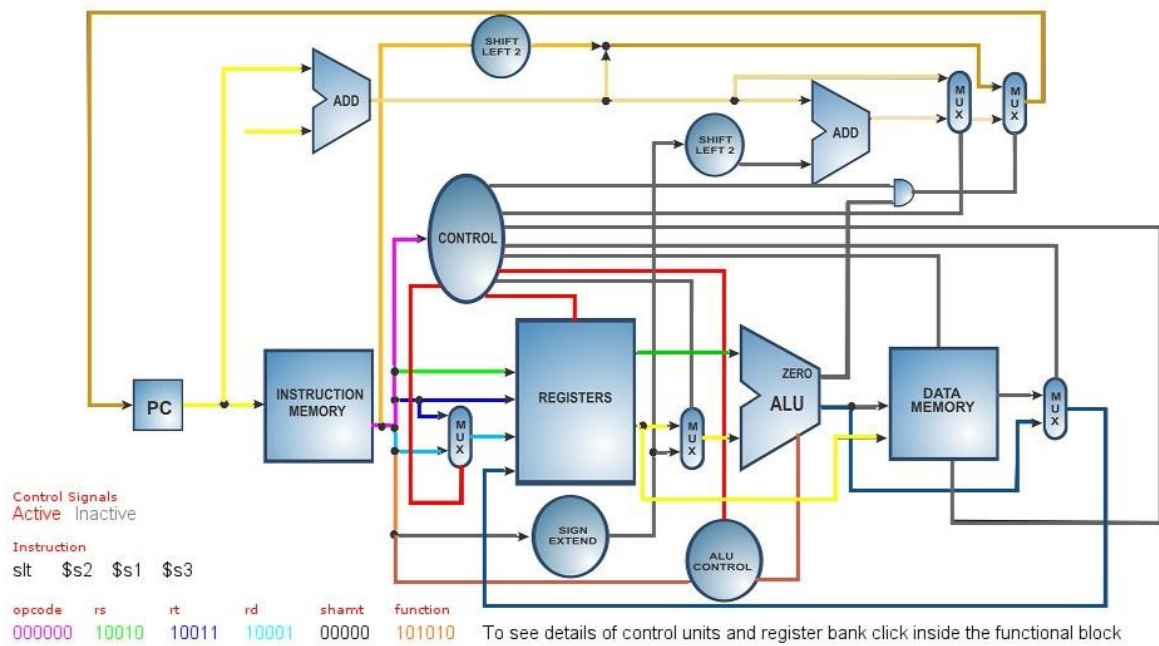


And $s_1, s_2, 10$ ($s_1 = s_2 \ll 10$)



Slt \$s1, \$s2, \$s3 (\$s2 < \$s3) ? \$s10:1

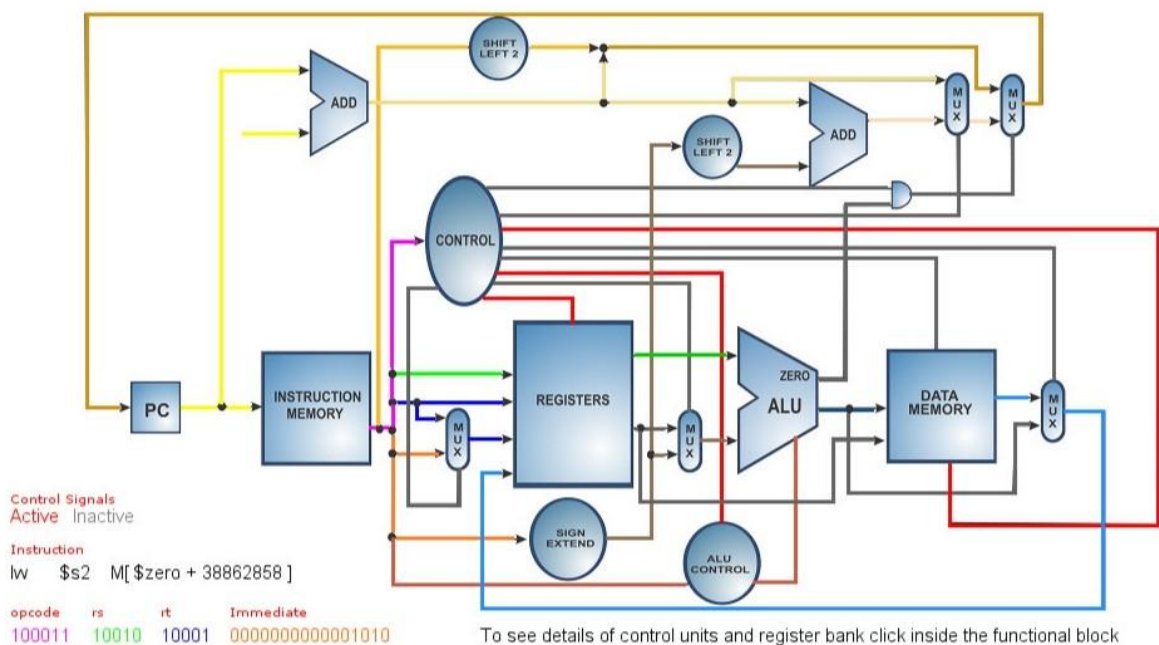
REGISTER TYPE INSTRUCTION



Câu 6. Vẽ sơ đồ datapath của lệnh Lw rt, constant(rd)

Lw \$s1, 10(\$s2)

LOAD TYPE INSTRUCTION



Câu 7. Vẽ sơ đồ datapath của lệnh Sw rt, constant(rd)

Sw \$s1, 10(\$s2)

STORE TYPE INSTRUCTION

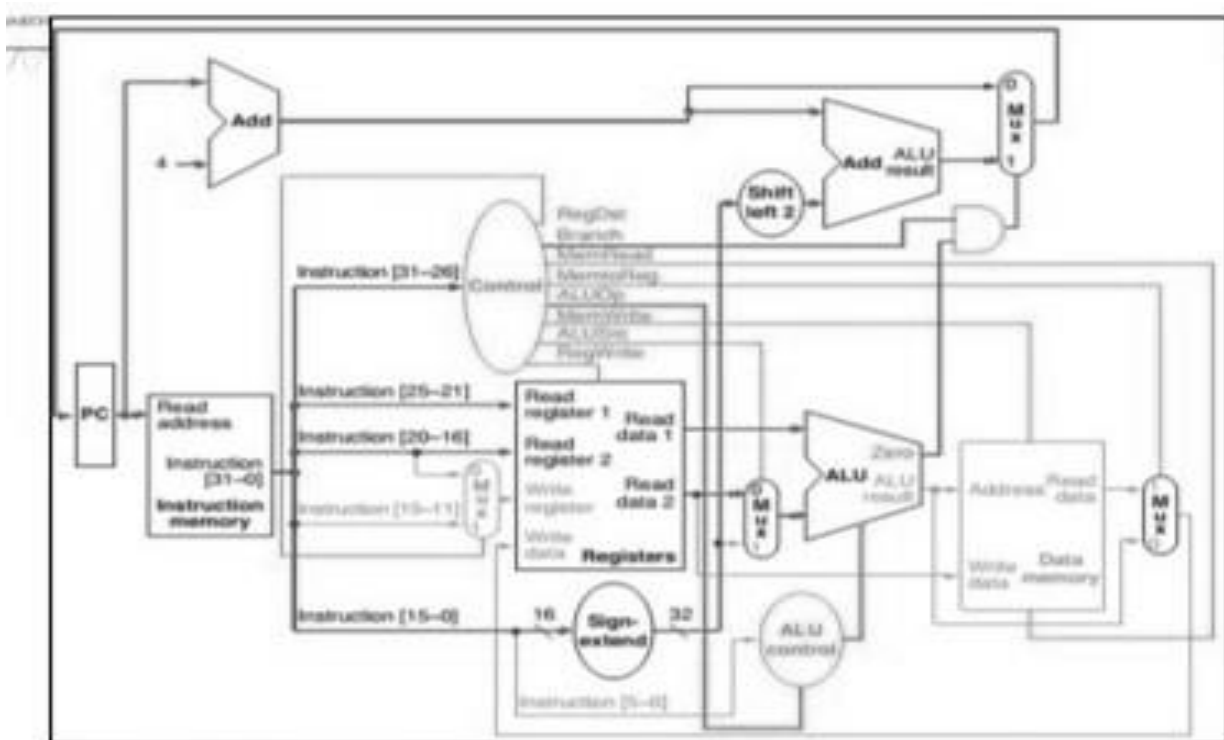
Control Signals
Active Inactive

Instruction
sw \$s2 M[\$zero + 38862858]

opcode	rs	rt	Immediate
101011	10010	10001	0000000000001010

To see details of control units and register bank click inside the functional block

Câu 8. Vẽ sơ đồ datapath của lệnh Beq rs, rt, constant và Bne rs, rt, constant



Các đường đậm nét (đỏ) là các đường hoạt động khi lệnh **beg** thực thi

Câu 10. Khi các nhà thiết kế bộ xử lý xem xét khả năng cải tiến đường dẫn dữ liệu của bộ xử lý, quyết định thường phụ thuộc vào sự cân bằng giữa chi phí/hiệu suất. Trong ba vấn đề sau, giả sử rằng chúng ta đang bắt đầu với một đường dẫn dữ liệu từ Hình 4.2, trong đó các khối I-Mem, Add, Mux, ALU, Regs, D-Mem và Control có độ trễ là 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps và 100 ps tương ứng và chi phí lần lượt là 1000, 30, 10, 100, 200, 2000 và 500.

Hãy xem xét việc bổ sung một số nhân vào ALU. Việc bổ sung này sẽ tăng thêm 300 ps vào độ trễ của ALU và sẽ tăng thêm chi phí 600 cho ALU. Kết quả là số lệnh được thực hiện sẽ ít hơn 5% vì chúng ta sẽ không cần mô phỏng lệnh MUL nữa.

	I mem	Add	Mux	ALU	Rg	D mem	Control
Đồ trễ	400ps	100ps	30ps	120ps	200ps	350ps	100ps
Chi phí	1000	30	10	100	200	2000	500
Tổng số lệnh	I						
SAO KHI BỔ SUNG NHÂN CHO ALU +300ps độ trễ và +600 chi phí							
Đồ trễ	400ps	100ps	30ps	420ps	200ps	350ps	100ps
Chi phí	1000	30	10	700	200	2000	500
Tổng số lệnh	0.95I						

a. Thời gian chu kỳ đồng hồ có và không có cải tiến này là bao nhiêu?

	Không cải tiến	Có cải tiến
Thời gian chu kỳ đồng hồ	400ps	420ps

b. Sự tăng tốc đạt được bằng cách thêm cải tiến này là gì?

Ta có:

Thời gian thực thi = Tổng số lệnh X CPI X Thời gian chu kỳ đồng hồ

	Không cải tiến	Có cải tiến
Thời gian thực thi	$I \times CPI \times 400 = 400I.CPI$	$0.95IXCPIX420=399I.CPI$

⇒ Thời gian thực thi của máy sau khi cải tiến nhanh hơn thời gian thực thi của máy không cải tiến

c. So sánh tỷ lệ chi phí/hiệu suất có và không có điều này sự cải tiến.

Ta có:

$$\text{Hiệu suất} = \frac{1}{\text{Thời gian thực hiện}}.$$

	Không cải tiến (1)	Có cải tiến (2)
Hiệu suất	$\frac{1}{400I.CPI}$	$\frac{1}{399I.CPI}$
Chi phí	3840	4440
$\frac{Chi\ phí}{Hiệu\ suất}$ (x)	3840.400.I.CPI	4440.399.I.CPI
$\frac{x_2}{x_1}$	$\frac{4440.399.I.CPI}{3840.400.I.CPI} = 1.1534$	

Câu 10. Trong bài tập này, chúng ta kiểm tra cách đường ống ảnh hưởng đến thời gian chu kỳ xung nhịp của bộ xử lý. Các bài toán trong bài tập này giả định rằng các giai đoạn riêng lẻ của đường dữ liệu có độ trễ sau:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Ngoài ra, giả sử rằng các lệnh được bộ xử lý thực thi được chia nhỏ như sau:

ALU	BEQ	LW	SW
45%	20%	20%	15%

a. Thời gian chu kỳ đồng hồ trong bộ xử lý có đường ống và không có đường ống là bao nhiêu?

- Không có ống dẫn:

[illegible]

BE Q						IF	ID	EX	ME M	W B										
LW											IF	ID	EX	ME M	W B					
SW																IF	ID	EX	ME M	W B

Time	ALU	BEQ	LW	SW	Tổng
	3150ps	1400ps	1400ps	1050ps	7000ps

- Có ống dẫn:

Time	350	350	350	350	350	350	350	350
ALU	IF	ID	EX	MEM	WB			
BEQ		IF	ID	EX	MEM	WB		
LW			IF	ID	EX	MEM	WB	
SW				IF	ID	EX	MEM	WB

Time	ALU	BEQ	LW	SW	Tổng
	1260ps	560ps	560ps	420ps	2800ps

⇒ Thời gian chu kỳ: 350ps

b. Tổng độ trễ của lệnh LW trong đường ống là bao nhiêu và bộ xử lý không có đường ống?

Tổng độ trễ của lệnh LW:

- Có ống dẫn : 700ps
- Không có ống dẫn : 3500ps

Câu 11. Trong bài tập này, chúng tôi xem xét sự phụ thuộc dữ liệu ảnh hưởng như thế nào đến việc thực thi trong quy trình 5 giai đoạn cơ bản được mô tả trong Phần 4.5. Các vấn đề trong bài tập này liên quan đến trình tự hướng dẫn sau:

or r1,r2,r3

or r2,r1,r4

or r1,r1,r2

Or			IF	ID	EX	MEM	WB		
NOP									
Or					IF	ID	EX	MEM	WB

- e. Thêm hướng dẫn nop vào mã này để loại bỏ các mối nguy hiểm nếu chỉ có chuyển tiếp ALU-ALU (không chuyển tiếp từ MEM sang giai đoạn EX).

- f. Tổng thời gian thực hiện của chuỗi lệnh này là bao nhiêu chỉ với chuyển tiếp ALU-ALU? Tăng tốc qua đường ống không chuyển tiếp là gì?

Câu 12. Trong bài tập này, chúng tôi xem xét các mối nguy hiểm về tài nguyên, các mối nguy hiểm về kiểm soát và thiết kế Kiến trúc Tập lệnh (ISA) có thể ảnh hưởng đến việc thực thi theo đường ống như thế nào. Các vấn đề trong bài tập này liên quan đến đoạn mã MIPS sau:

Giả sử rằng các giai đoạn quy trình riêng lẻ có độ trễ sau:

Time	200	200	200	R6	200	200	200	200	200	200	200	200
Sw	IF	ID	EX	MEM	WB							
NOP						R16						
Lw			IF	ID	EX	MEM	WB					
NOp												
Bea					IF	ID	EX	MEM	WB			

Add						IF	ID	EX	MEM	WB		
Slr							IF	ID	EX	MEM	WB	

- Tổng thời gian 2200ps.
- Không thể dung lệnh nop để giải quyết mỗi nguy cơ cấu trúc.

Câu 13. Trong bài tập này, chúng tôi kiểm tra chi tiết cách thực hiện một lệnh trong đường dẫn dữ liệu một chu kỳ. Các vấn đề trong bài tập này đề cập đến một chu kỳ xung nhịp trong đó bộ xử lý tìm nạp từ lệnh sau:

101011000110001000000000000010100.

Giả sử rằng bộ nhớ dữ liệu hoàn toàn bằng 0 và các thanh ghi của bộ xử lý có các giá trị sau vào đầu chu kỳ trong đó từ lệnh trên được tìm nạp:

R0	R1	R2	R3	R4	R5	R6	R8	R12	R31
0	-1	2	-3	-4	10	6	8	2	-16

Ta có: 101011 000110001000000000000010100

101011 = 2B ➔ Lệnh sw

101011 00011 00010 00000000000010100

Op : 2B

Rs : 3

Rt : 2

Constant : 20

Lệnh: sw \$v0, 20(\$v1)

- Đầu ra của đơn vị mở rộng dấu hiệu và bước nhảy “Shift left 2” (gần đầu Hình 4.24) cho từ lệnh này là gì?

Constant (16 bit) = 00000000000010100

Mở rộng dấu Constant (32 bit) = 0000 0000 0000 0000 0000 0000 0001 0100

- Giá trị đầu vào của bộ điều khiển ALU cho lệnh này là gì?

Đầu vào ALU:

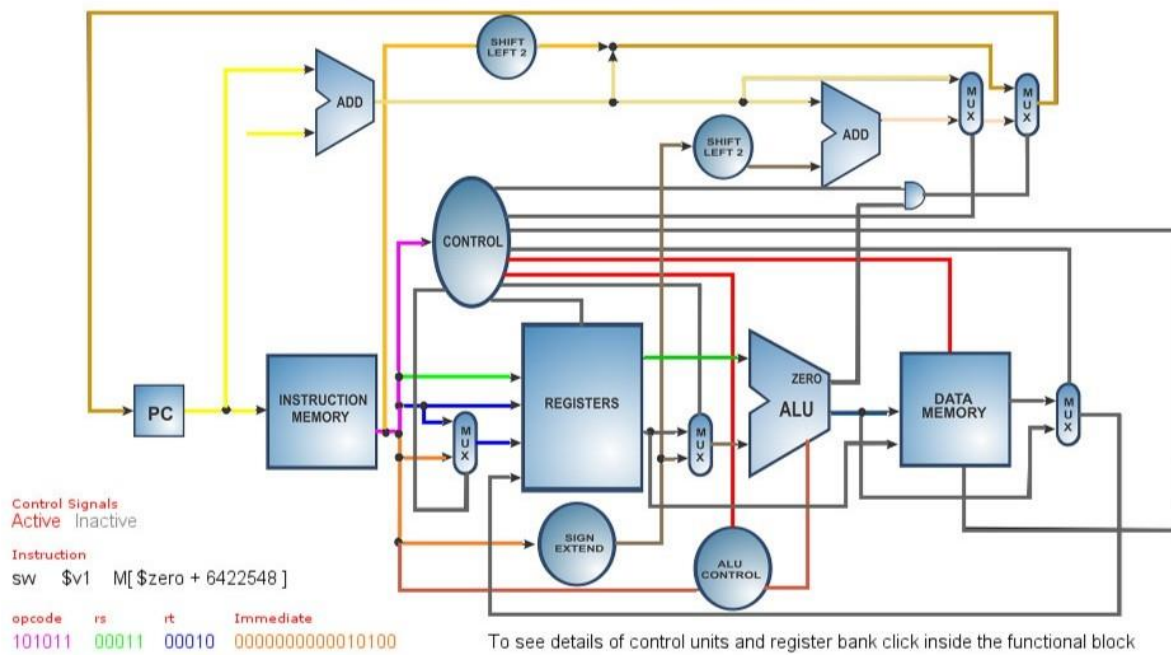
Rs : -3

Constant : 20

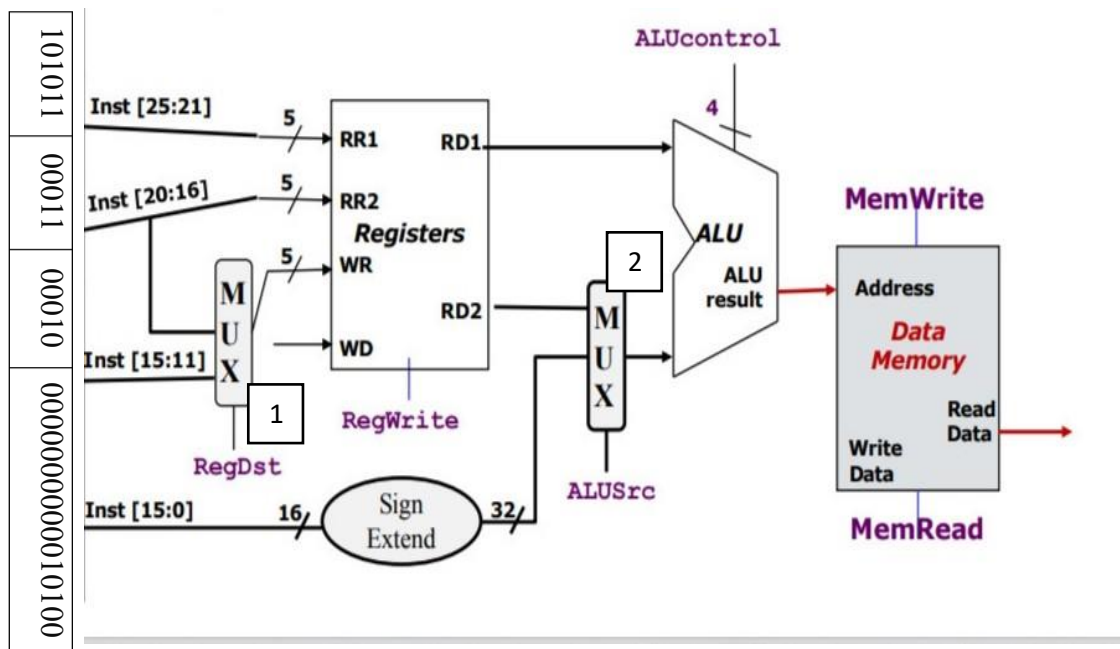
- Địa chỉ PC mới sau khi lệnh này được thực thi là gì? Đánh dấu đường dẫn mà giá trị này được xác định.

Địa chỉ PC=PC+4

STORE TYPE INSTRUCTION



- d. Đối với mỗi Mux, hiển thị các giá trị dữ liệu đầu ra của nó trong quá trình thực hiện lệnh này và các giá trị thanh ghi này.



MUX 01 : rt(00010) = 2

MUX 02 : Constant(00000000000010100) = 20

- e. Đối với ALU và hai đơn vị cộng, giá trị đầu vào dữ liệu của chúng là bao nhiêu?

ALU :

Rs (00011) = -3

Constant(00000000000010100) = 20

- f. Giá trị của tất cả các đầu vào cho đơn vị "Registers" là gì?

R1 = Rs(00011) = -3

R2 trống = 0

Câu 14. Vẽ sơ đồ ống dẫn cho chuỗi lệnh sau:

T1: lw \$t1, 8(\$t0)

T2: addi \$t2, \$t1, 2 → Xung đột dữ liệu \$t1

T3: Or \$t3, \$t1, \$t2 → Xung đột dữ liệu \$t2

T4: Sw \$t4, -4(\$t3) → Xung đột dữ liệu \$t3

T5: sub \$t5, \$t3, \$t4 → Xung đột dữ liệu \$t4

Chưa cải tiến:

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
T1	IF	ID	EX	MEM	WB																				
T2						IF	ID	EX	MEM	WB															
T3											IF	ID	EX	MEM	WB										
T4																IF	ID	EX	MEM	WB					
T5																					IF	ID	EX	MEM	WB

Lệnh Lw, Sw tới MEM mới có kết quả → Cải tiến MEM – ALU (MEM-EX)

Lệnh Add, Sub, Or, And tới EX có kết quả → Cải tiến ALU -ALU (EX-EX)

Time	1	2	3	4	5	6	7	8	9	10	11
T1	IF	ID	EX	MEM	WB						
NOP											
T2			IF	ID	EX	MEM	WB				
T3				IF	ID	EX	MEM	WB			
T4					IF	ID	EX	MEM	WB		
NOP											
T5							IF	ID	EX	MEM	WB