



kiến trúc máy tính đề cương ôn tập

Đề cương kiến trúc máy tính cuối kì (Trường Đại học Bách khoa Hà Nội)

Chương 1:

Câu 1: Trình bày khái niệm về máy tính điện tử?

- Máy tính điện tử là thiết bị có khả năng xử lý dữ liệu theo một tập hợp của các mạng máy tính hay câu lệnh đã được chứa sẵn bên trong một cách một cách tạm thời hay vĩnh viễn .

Câu 2: Máy tính điện tử đầu tiên ENIAC sử dụng linh kiện đèn điện tử hay transistor? Số lượng linh kiện tích hợp trên chip thể hiện điều gì?

- Máy tính điện tử đầu tiên ENIAC sử dụng linh kiện đèn điện tử.
- Là loại linh kiện điện tử sử dụng sự phát xạ điện tử do nung nóng điện cực nằm trong môi trường chân không cao để thực hiện điều khiển dòng điện tích trong các khuếch đại.

Câu 3: Trình bày chức năng cơ bản của máy tính?

- Chức năng cơ bản của máy tính lưu trữ dữ liệu, xử lý dữ liệu, trao đổi dữ liệu, điều khiển.

Câu 4: Trình bày các thành phần cơ bản của máy tính, nêu chức năng của từng thành phần đó?

Các thành phần cơ bản của máy tính:

- Hộp máy tính :là nơi để gắn các thành phần của máy tính thành khối như nguồn, Mainboard, Card v.v... có tác dụng bảo vệ máy tính
- Bộ nguồn: là nơi cung cấp hầu hết hệ thống điện cho các thiết bị bên trong máy tính
- CPU : bộ xử lý trung tâm của máy tính, là thành phần quan trọng nhất của máy tính
- Bộ nhớ trong (RAM,ROM): là nơi lưu trữ chương trình phục vụ trực tiếp cho việc xử lý của CPU, nó giao tiếp trực tiếp với CPU
- Bộ nhớ ngoài: là nơi lưu trữ dữ liệu và chương trình, gián tiếp phục vụ cho CPU thông qua 1 thiết bị trung gian (thường là RAM)
- Màn hình: Là thiết bị đưa thông tin ra giao diện trực tiếp với người dùng. Đây là thiết bị xuất chuẩn của máy vi tính.
- Bàn phím: Thiết bị nhập tin vào giao diện trực tiếp với người dùng. Đây là thiết bị nhập chuẩn của máy vi tính.
- Chuột: Thiết bị điều khiển trỏ giao diện trực tiếp với người sử dụng.
- Máy in : Thiết bị xuất thông tin ra giấy thông dụng nhất.

Câu 5: Hiểu thế nào là phần dẻo (firmware) trong máy tính? Chúng có nhiệm vụ chức

- Firmware là 1 chương trình cơ sở có dữ liệu được lưu trữ trên máy tính hoặc ROM của thiết bị phần cứng khác (bộ nhớ chỉ đọc) cung cấp hướng dẫn về cách thiết bị đó hoạt động.
- Firmware có vai trò cho phép phần cứng thực hiện những gì được yêu cầu, một điểm quan trọng là firmware được lưu trữ trên chính thiết bị phần cứng.

Câu 6: Đơn vị xử lý trung tâm (CPU MIPS) gồm những bộ phận nào? Trình bày chức năng của từng bộ phận đó?

- Khối điều khiển (CU - Control Unit): có nhiệm vụ thông dịch các lệnh của chương trình và điều khiển hoạt động xử lý, được điều tiết chính xác bởi xung nhịp đồng hồ hệ thống.
- Khối tính toán ALU (Arithmetic Logic Unit): Chức năng thực hiện các phép toán số học và logic sau đó trả lại kết quả cho các thanh ghi hoặc bộ nhớ.
- Các thanh ghi (Registers): Là các bộ nhớ có dung lượng nhỏ nhưng tốc độ truy cập rất cao, nằm ngay trong CPU, dùng để lưu trữ tạm thời các toán hạng, kết quả tính toán, địa chỉ các ô nhớ hoặc thông tin điều khiển.

Câu 7: Nói việc trao đổi dữ liệu giữa máy tính và thiết bị ngoại vi được thực hiện qua 1 cổng (port) có đúng không? Ví dụ?

- Việc trao đổi dữ liệu giữa máy tính và thiết bị ngoại vi được thực hiện qua 1 cổng là đúng
- Vd : chuột kết nối với bó mạch chủ qua: COM, [PS/2](#), [USB](#) và [kết nối không dây](#).

Câu 8: . Bộ nhớ chính trong máy tính gồm những thành phần nào? Tại sao gọi RAM là bộ nhớ truy cập ngẫu nhiên?

- Bộ nhớ chính trong hệ thống nhớ của máy tính bao gồm 2 thành phần chính là ROM và RAM.
- [RAM](#) được gọi là bộ nhớ truy cập ngẫu nhiên vì nó có đặc tính: thời gian thực hiện thao tác đọc hoặc ghi đối với mỗi ô nhớ là như nhau, cho dù đang ở bất kỳ vị trí nào trong bộ nhớ. Mỗi ô nhớ của RAM đều có một địa chỉ. Thông thường, mỗi ô nhớ là một byte (8 bit); tuy nhiên hệ thống lại có thể đọc ra hay ghi vào nhiều byte (2, 4, 8 byte).

Câu 9: Dựa vào những tiêu chí gì để đánh giá sự phát triển của máy tính điện tử số qua các giai đoạn khác nhau?

- Dựa vào số lượng transistors trên chip, dự đoán số transistors trên chip sẽ tăng lên theo cấp số nhân(tăng gấp đôi sau 18 tháng).

Câu 10: Trình bày nguyên lý Von Neumann?

- Nguyên lý điều khiển bằng chương trình: máy tính thực hiện một công việc theo chương trình được đưa vào bộ nhớ. Nguyên lý này đảm bảo khả năng thực hiện tự động để giải quyết một bài toán của máy tính điện tử
- Nguyên lý truy cập qua địa chỉ: dữ liệu trong chương trình không chỉ định bằng giá trị mà thông qua địa chỉ trong bộ nhớ. Nguyên lý đảm bảo tính mềm dẻo của chương trình, có thể thể hiện thuật toán không phụ thuộc vào các giá trị phát sinh trong chương trình

Câu 11: Thông tin được lưu trữ và truyền bên trong máy tính dưới dạng nào?

- Thông tin được lưu trữ và truyền bên trong máy dưới dạng dãy nhị phân dựa trên các số 0 và 1

Câu 12: .Trình bày trình tự xử lý thông tin của máy tính điện tử/

- Nhận thông tin : Thu thập thông tin từ thế giới bên ngoài vào máy tính .Thực chất đây là quá trình chuyển đổi thông tin ở thế giới thực sang dạng biểu diễn thông tin trong máy tính thông qua các thiết bị vào.
- Xử lý thông tin: Biến đổi, phân tích, tổng hợp, tra cứu... những thông tin ban đầu để có được những thông tin mong muốn .
- Xuất thông tin: Đưa ra các thông tin kết quả đã được xử lý ra trở lại thế giới bên ngoài thông qua các thiết bị đầu ra.
- Lưu trữ thông tin: Ghi nhớ lại các thông tin đã được ghi nhận để có thể đem ra sử dụng trong những lần xử lý về sau .

Câu 13: Thế nào là bus hệ thống? Chức năng của chúng?

- Bus hệ thống là một bus máy tính đơn lẻ kết nối các thành phần chính của một hệ thống máy tính, kết hợp các chức năng của một bus dữ liệu để mang thông tin, một bus địa chỉ để xác định nơi thông tin sẽ được gửi và bus điều khiển để xác định lệnh của nó.

Câu 14: Biểu đồ luật Moore thể hiện điều gì?

- Biểu đồ Moore thể hiện Số lượng transistor trên mỗi đơn vị inch vuông sẽ tăng lên gấp đôi sau mỗi 18 tháng.

Câu 15: .Hiểu thế nào là kiến trúc máy tính?

- Kiến trúc máy tính là thiết kế khái niệm và cấu trúc hoạt động căn bản của một hệ thống [máy tính](#), đáp ứng được các mục đích về tính năng, hiệu suất và giá cả.

Câu 16 : RISC, CISC được hiểu như thế nào? Vi xử lý MIPS thuộc kiến trúc nào?

- RISC là một kiến trúc tập lệnh được thiết kế để thực hiện một số lượng nhỏ hơn các lệnh máy tính để nó có thể hoạt động ở tốc độ cao hơn. CISC là một bộ đầy đủ các hướng dẫn máy tính có ý định cung cấp các khả năng cần thiết một cách hiệu quả.
- MIPS là kiến trúc bộ tập lệnh RISC

Câu 17: .Chức năng nhiệm vụ của CPU?

- Bộ xử lý trung tâm CPU được xem là não bộ của máy tính với nhiệm vụ chính là xử lý các chương trình vi tính, dữ kiện đầu vào của máy tính và xử lý tất cả các lệnh mà CPU nhận được từ phần cứng và phần mềm chạy trên laptop hay máy tính.

Chương 3:

Câu 1: Đối với số nguyên không dấu, 8 bit, giá trị biểu diễn số 261 là 1000 0111 có đúng không? Tại sao?

- Biểu diễn số 261 là 1000 0111 là sai tại vì dải biểu diễn của số nguyên không dấu 8 bit chạy từ 0 -> $2^8 - 1$.

Câu 2: Đối với số nguyên có dấu, 8 bit, giá trị biểu diễn số 124 là 0111 1100 có đúng không? Tại sao?

- Biểu diễn số 124 là 0111 1100 là đúng vì ta chuyển đổi số nguyên sang nhị phân như bình thường thôi.

Câu 3: Một số nguyên không dấu n bit, khả năng biểu diễn của nó bằng bao nhiêu? Dải biểu diễn của nó?

- Đối với số nguyên không dấu n bit , khả năng biểu diễn của nó = 2^n (số)
- Dải biểu diễn = 0 -> $2^n - 1$.

Ví dụ: n = 8 bit. dải biểu diễn = 0 -> $2^8 - 1$ (0 -> 255)

n = 16 bit. dải biểu diễn = 0 -> $2^{16} - 1$ (0 -> 65525)

Câu 4: Một số nguyên có dấu n bit, khả năng biểu diễn của nó bằng bao nhiêu? Dải biểu diễn của nó?

- Đối với số nguyên có dấu n bit, khả năng biểu diễn của nó = 2^n (số)
 - Dải biểu diễn = $-2^{(n-1)} \rightarrow 2^{(n-1)} - 1$
- Ví dụ: n=8 bit. dải biểu diễn = $-2^7 \rightarrow 2^7 - 1$ (-128 \rightarrow 127)
 n=16 bit. dải biểu diễn = $-2^{15} \rightarrow 2^{15} - 1$ (-32768 \rightarrow 32767)

Câu 5: Trình bày khái niệm về mã bù 2? Trong máy tính dùng mã bù 2 để làm gì?

- Bù 2 là một số trong hệ nhị phân là bù đúng của một số khác . Một số bù 2 có được do đảo tất cả các bit trong số nhị phân (đổi 1 thành 0 và ngược lại) rồi thêm 1 vào kết quả vừa đạt được. Thực chất bù 2 là số biểu diễn ở bù 1 rồi cộng thêm 1.
- Mã bù 2 thường được sử dụng để biểu diễn số âm trong máy tính. Với bit đầu là 0 thì số dương , là 1 thì số âm. Ngoài bit đầu ra các bit còn lại biểu diễn độ lớn.

Câu 6: Đối với số nguyên có dấu 8 bit, dùng phương pháp “Mã bù 2”, giá trị biểu diễn số 101 là?

- Bằng phương pháp bù 2 giá trị biểu diễn 101 = -5.

Câu 7: Đối với số nguyên có dấu, 8 bit, dùng phương pháp “Mã bù 2”, giá trị biểu diễn số - 29 là?

- Bằng phương pháp bù 2 giá trị biểu diễn số -29 = 11100011.

Câu 8: Đối với các số 8 bit, không dấu. Hãy cho biết kết quả khi thực hiện phép cộng: 0100 0111 + 0101 1111?

- Nguyên tắc cộng nhị phân:

$$0+0=0$$

$$1+0=1$$

$$0+1=1$$

$$1+1=0 \text{ (nhớ 1)}$$

- Tổng 2 số 8 bit , không dấu: 0100 0111+ 0101 1111= 1010 0110.

Câu 9: . Đối với số có dấu, 8 bit, xét phép cộng: (-39) + (-42). Phát biểu nào sau đây là đúng: (A. Không cho kết quả, vì tràn số; B. Không cho kết quả, vì có nhớ ra khỏi bit cao nhất; C. Tổng là -81; D. Tổng là 81)? Tại sao?

- Phát biểu “C.Tổng là -81” là đúng vì đối với số nguyên có dấu 8 bit thì dải biểu diễn từ -2^7 đến 2^7-1 , tổng của 2 số trên là -81 thuộc vào khoảng đó \rightarrow biểu diễn được dạng nhị phân

Câu 10: .Đối với số có dấu, 8 bit, xét phép cộng: $91 + 63$. Phát biểu nào sau đây là đúng: (A. Không cho kết quả, vì tràn số; B. Kết quả sai, vì có nhớ ra khỏi bit cao nhất; C. Tổng là 154; D. Tổng là -102)? Tại sao?

- Phát biểu “ D. tổng là -102” đúng vì đối với số nguyên có dấu 8 bit thì dải biểu diễn từ -128 đến 127 phép cộng trên ra 154 lớn hơn khoảng trên nhưng giá trị đủ lớn hay bé thì vòng tròn trực số vẫn quay, ta chiếu 154 lên trục số được số liền sau là -102

Câu 11: Phép cộng 2 số hệ 16: $1C37286A + 9395E84B$ có kết quả bằng bao nhiêu?

- Tổng 2 số hệ 16: $1C37286A + 9395E84B = AFCD10B5$.

Câu 12: .Phép trừ 2 số hệ 16: $B14FC675 - 839EA247$ có kết quả bằng bao nhiêu?

- Hiệu 2 số hệ 16: $B14FC675 - 839EA247 = 2DB1242E$.

Câu 13: Số hex $9DB1242E$ là số dương hay âm?

- Số hex $9DB1242E$ là số âm vì số đầu tiên từ 8-15 là số âm , từ 0-7 là số dương

Câu 14: Giá trị lớn nhất của số nguyên không dấu 8 bit bằng bao nhiêu?

- Giá trị lớn nhất của số nguyên không dấu 8 bit là 255

Câu 15: Giá trị lớn nhất của số nguyên có dấu 8 bit bằng bao nhiêu?

- Giá trị lớn nhất của số nguyên có dấu 8 bit là 127

Câu 16: .1KB = ?byte

- 1KB = 1000 byte

Câu 17: .1Kb = ?byte

- 1Kb = 12.5 byte

**Câu 18: .Cho số hệ thập lục phân $8A1F$, giá trị hệ thập phân đối với số có dấu và không dấu bằng bao nhiêu?

- Hệ 16 : $8A1F$ sang hệ 10 đối với số không dấu là 105037.
- Hệ 16 : $8A1F$ sang hệ 10 đối với số có dấu là 105037.

Câu 19: .Khái niệm về Carry và Overflow?

- Carry dùng để :
 - +) cộng hoặc trừ các số nguyên không dấu
 - +) Cho biết rằng tổng không dấu nằm ngoài phạm vi
 - +) Nhỏ hơn 0 hoặc lớn giá trị n bit không dấu lớn nhất

- Overflow dùng để:
 - +) Cộng trừ các số nguyên có dấu
 - +) Cho biết tổng các số nguyên có dấu nằm ngoài phạm vi
- Overflow xảy ra khi:
 - +) Cộng hai số dương và tổng là số âm
 - +) Cộng hai số âm và tổng là số dương
 - +) Có thể xảy ra khi tổng số bit cố định

Câu 20: Thực hiện phép cộng 2 số nhị phân $11010010 + 10111101$. Hãy cho biết giá trị 2 bit carry (C) và overflow (O)?

11010010

+

10111101

10001111

- Carry=1

Câu 21: Thực hiện phép cộng 2 số hexa $A1CF + B2D3$. Hãy cho biết giá trị 2 bit carry (C) và overflow (O)?

- phép cộng 2 số hexa $A1CF + B2D3$

1010 0001 1100 1111

+

1011 0010 1101 0011

0101 0100 1010 0010(54A2)

Câu 22: Số nhị phân (1111 1111 1111 1111 1111 1111 1111 1101)₂ được biểu diễn trong hệ thập phân là số nào sau đây: A. (-1)₁₀; B. (-2)₁₀; C. (-3)₁₀; D. (-4)₁₀

- Số nhị phân (1111 1111 1111 1111 1111 1111 1111 1101) được biểu diễn trong hệ thập phân là ý (C) -3.

Câu 23: .Chuyển đổi số thập phân (-2047) sang số nhị phân 32 bit có kết quả bằng bao nhiêu?

- [illegible]

Câu 24: Số thập phân $(-1)_{10}$ biểu diễn trong số nhị phân 32 bit bằng bao nhiêu?

- Chuyển đổi số thập phân (-1) sang số nhị phân 32 bit là

Mặt khác bộ nhớ chính hay đc gọi là bộ nhớ truy cập ngẫu nhiên (RAM), nó chứa dữ liệu và chương trình đang chạy . bộ nhớ chính được đặt bên ngoài CPU

****Câu 4: Chức năng của thanh ghi PC, HI, LO?**

- Chức năng của thanh ghi PC: lưu địa chỉ của lệnh sẽ được nạp
- Chức năng của thanh ghi HI:
- Chức năng của thanh ghi LO:

Câu 5: Khối ALU trong CPU có chức năng nhiệm vụ gì?

- Khối ALU trong CPU Có chức năng thực hiện các lệnh của đơn vị điều khiển và xử lý tín hiệu.

Câu 6: Khối Integer mul/div trong CPU có chức năng nhiệm vụ gì?

- Lệnh Mul (Multiply): Thực hiện phép nhân trên số không dấu. Nếu [Toán hạng nguồn] là toán hạng 8 bit thì lệnh sẽ nhân nội dung của [Toán hạng nguồn] với giá trị thanh ghi AL, kết quả 16 bit chứa ở thanh ghi Ax.
- Lệnh Div (Divide): Thực hiện phép chia trên số không dấu. Nếu [Toán hạng nguồn] là toán hạng 8 bit thì lệnh sẽ lấy giá trị của thanh ghi Ax (số bị chia) chia cho [Toán hạng nguồn] (số chia), kết quả thương số chứa trong thanh ghi AI, số dư chứa trong thanh ghi Ah.

Câu 7: EIU thực hiện chức năng gì?

- EIU có chức năng: chịu trách nhiệm thực hiện tất cả các lệnh kiểu dữ liệu số nguyên, cụ thể là tính toán số học, tính toán logic

Câu 8: FPU thực hiện chức năng gì?

- FPU có là mạch dùng để thực hiện các phép toán dấu phẩy động

Câu 9: MIPS có bao nhiêu thanh ghi đa năng/ dấu phẩy động? Các thanh ghi đó có chức năng gì?

- MIPS có 4 thanh ghi đa năng và 32 thanh ghi dấu phẩy động
- Chức năng:

+ thanh ghi đa năng : * AX (Accumulator): thanh ghi chứa, thường dùng để chứa kết quả của các thao tác chẳng hạn như lệnh nhân, chia và một số lệnh hiệu chỉnh dữ liệu.

* BX (Base): thanh ghi cơ sở, thường chứa địa chỉ lệch của ô nhớ trong đoạn DS.

* CX (Count): thanh ghi đếm, CX thường chứa số lần lặp trong các lệnh lặp, CL thường chứa số lần dịch hoặc quay trong lệnh dịch và quay thanh ghi.

- * DX (Data): thanh ghi dữ liệu, DX cùng với AX tham gia vào phép nhân hoặc chia các số 16 bit. DX còn dùng để chứa địa chỉ của các cổng vào/ra trong các lệnh vào ra dữ liệu

trục
tiếp.

+ thanh ghi dấu phẩy động : biểu diễn độ chính xác
đơn của số thực

Câu 10: .Khối FP Arith có chức năng gì?

- Một điểm chức năng (FP) là một thành phần của phát triển phần mềm giúp để xấp xỉ chi phí phát triển ban đầu trong quá trình này. Nó là một quá trình trong đó xác định các chức năng cần thiết và tính phức tạp của họ trong một phần của phần mềm để ước tính kích thước và phạm vi của phần mềm sau khi hoàn thành.

Câu 11: Thanh ghi trong máy tính có thể tháo rời để nâng cấp được

- Thanh ghi không thể tháo rời để nâng cấp

Câu 12: RAM có thể tháo rời để nâng cấp được không?

- Ram có thể tháo rời để nâng cấp

Câu 13: Trong các thanh ghi đã năng của MIPS, có thanh ghi không thể lập trình được không?

- Có thanh ghi không thể lập trình được

Câu 14: Kiến trúc tập lệnh MIPS có bao nhiêu định dạng lệnh, trình bày từng định dạng?

- Kiến trúc tập lệnh MIPS có 3 định dạng lệnh , gồm:
 - + R-type : _ các lệnh sử dụng 2 toán hạng là giá trị 2 thanh ghi và lưu kết quả vào 1 thanh ghi
 - _ Op: mã phép toán quy định lệnh
 - _ rs,rt: 2 thanh ghi thực hiện tính toán
 - _ rd: thanh ghi lưu kết quả của lệnh
 - _ sa: số bit cần dịch trong lệnh dịch trái và dịch phải.
 - _ funct: Vì các lệnh R-format đều có chung opcode bằng 0 nên ta thêm trường này để cho máy biết cần thực hiện lệnh nào.

Op(6)	Rs(5)	Rt (5)	Rd (5)	sa(5)	Funct(6)
-------	-------	--------	--------	-------	----------

+ I- type: dùng cho thao tác giữa thanh ghi và một hằng số được lưu sẵn trong lệnh

- _rs, rt: thanh ghi nguồn, thanh ghi đích
- _ immediate: Một giá trị hằng số mà lệnh sử dụng.

Op (6)	Rs (5)	Rt(5)	Immediate(16)
--------	--------	--------	---------------

+ J- type: dùng cho các lệnh nhảy

Op (6)	Immediate(26)
--------	---------------

Câu 15: Định dạng R-type có mấy trường bit? Ý nghĩa của từng trường bit?

- Định dạng R- type có 6 trường bit
- Ý nghĩa từng trường bit: + opcode: mã thao tác , cho biết loại lệnh gì
 - + funct: dùng kết hợp với opcode để xác định lệnh làm gì (trường hợp các lệnh có cùng mã thao tác opcode
 - + shamt: số bit cần dịch trong lệnh dịch trái và dịch phải.
 - + rs: chứa địa chỉ thanh ghi nguồn 1
 - + rt: chứa địa chỉ thanh ghi nguồn 2
 - + rd: thanh ghi lưu kết quả của lệnh

Câu 16: Định dạng I-type có mấy trường bit? Ý nghĩa của từng trường bit?

- Định dạng I-type có 4 trường bit
- Ý nghĩa từng trường bit: + opcode, cho máy biết đây là lệnh gì. Vì I-format không có trường funct nên các lệnh I-format không dùng chung opcode như các lệnh R-format.
 - + rs, rt: thanh ghi nguồn, thanh ghi đích
 - + immediate: Một giá trị hằng số mà lệnh sử dụng.

Câu 17: Định dạng J-type có mấy trường bit? Ý nghĩa của từng trường bit?

- Định dạng J- type có 2 trường bit
- Ý nghĩa từng trường là : + opcode: cho máy biết đây là lệnh gì.
 - + immediate: Địa chỉ rút gọn của lệnh cần nhảy đến

Câu 18: Trong định dạng I-type, muốn khởi tạo hằng số 32 bit ta phải làm như thế nào

- Muốn khởi tạo hằng số 32 bit ta làm như sau:

Câu 19: Giả sử f, g, h, i và j được ánh xạ vào các thanh ghi \$s0, ..., \$s4. Cho đoạn code sau: addu \$t0, \$s1, \$s2

addu \$t1, \$s3, \$s4

subu \$s0, \$t0, \$t1 Xác định \$s0?

- Addu \$t0, \$s1, \$s2: cộng không dấu $t0 = s1 + s2 = gh$
- Addu \$t1, \$s3, \$s4: cộng không dấu $t1 = s3 + s4 = ij$
- Subu \$s0, \$t0, \$t1: trừ không dấu $s0 = t0 - t1 = s1 + s2 - (s3 + s4) = gh - ij$

Câu 20: Cho câu lệnh addu \$t0, \$s1, \$s2. Xác định mã máy?

- Trường op/ funct tra trong bảng "MIPS reference data" tương ứng với 0/21(hex)

- thanh ghi \$s0 đến \$s7 có chỉ số tương ứng từ 16 đến 23, và thanh ghi \$t0 đến \$t7 có chỉ số tương ứng từ 8 đến 15

op	rs=\$s1	rt= \$s2	rd= \$t0	sa	funct
000000	10001	10010	01000	00000	100100001

Câu 21. Cho câu lệnh hợp ngữ addu \$s1, \$s2, \$s3. Xác định mã máy?

op	rs=\$s1	rt= \$s2	rd= \$t0	sa	funct
000000	10010	10011	10001	00000	100100001

Câu 22. Cho câu lệnh hợp ngữ add \$t1, \$s2, \$s3. Xác định mã máy?

- Mã máy add \$t1, \$s2, \$s3 = 000000 10010 10011 01001 00000 100000

Câu 23. Cho câu lệnh hợp ngữ subu \$t2, \$s2, \$s3. Xác định mã máy?

- Mã máy subu \$t2, \$s2, \$s3 = 000000 10010 10011 01010 00000 100011

Câu 24. Cho câu lệnh hợp ngữ and \$t0, \$s1, \$s2. Xác định mã máy?

- Mã máy and \$t0, \$s1, \$s2 = 000000 10001 10010 01000 00000 100100

Câu 25. Cho câu lệnh hợp ngữ or \$t3, \$s4, \$s5. Xác định mã máy?

- Mã máy or \$t3, \$s4, \$s5 = 000000 10100 10101 01011 00000 100101

Câu 26. Giả sử \$s1 = 0xabcd1234 và \$s2 = 0xffff0000. Cho lệnh hợp ngữ and \$s0, \$s1, \$s2, hãy xác định kết quả trên thanh ghi \$s0?

- \$s1 = 10101011110011010001001000110100
- \$s2 = 11111111111111110000000000000000

⇒ \$s0 = 10101011110011010000000000000000

Câu 27. Giả sử \$s1 = 0xabcd1234 và \$s2 = 0xffff0000. Cho lệnh hợp ngữ or \$s0, \$s1, \$s2, hãy xác định kết quả trên thanh ghi \$s0?

- \$s1 = 10101011110011010001001000110100
- \$s2 = 11111111111111110000000000000000

⇒ \$s0 = 11111111111111110001001000110100

Câu 28. Giả sử \$s1 = 0xabcd1234 và \$s2 = 0xffff0000. Cho lệnh hợp ngữ xor \$s0, \$s1, \$s2, hãy xác định kết quả trên thanh ghi \$s0?

- \$s1 = 10101011110011010001001000110100
- \$s2 = 11111111111111110000000000000000

⇒ \$s0 = 01010100001100100001001000110100

Câu 29. Giả sử \$s1 = 0x11223344 và \$s2 = 0xffff0000. Cho lệnh hợp ngữ xor \$s0, \$s1, \$s2, hãy xác định kết quả trên thanh ghi \$s0?

- \$s1 = 00010001001000100011001101000100
- \$s2 = 11111111111111110000000000000000

⇒ \$s0 = 11101110110111010011001101000100

Câu 30. Cho câu lệnh hợp ngữ sll \$s1, \$s2, 10, với \$s2 = 0xaabbccdd, xác định \$s1?

⇒ \$s1 = 0xEF337400

Câu 31. Cho câu lệnh hợp ngữ sll \$s1, \$s2, 4, với \$s2 = 0x1122ccdd, xác định \$s1?

⇒ \$s1 = 0x122CCDD0

Câu 32. Cho câu lệnh hợp ngữ srl \$s1, \$s2, 10, với \$s2 = 0xaabbccdd, xác định \$s1?

⇒ \$s1 = 0x2AAEF3

Câu 33. Cho câu lệnh hợp ngữ srl \$s1, \$s2, 4, với \$s2 = 0xccdd1122, xác định \$s1?

⇒ \$s1 = 0xCCDD112

Câu 34. Cho câu lệnh hợp ngữ srlv \$s1, \$s2, \$s3, với \$s2 = 0xaabbccdd, \$s3=8, xác định \$s1?

⇒ \$s1 = 0xAABBCC

Câu 35. Cho câu lệnh hợp ngữ sra \$s1, \$s2, 10, với \$s2 = 0xaabbccdd, xác định \$s1?

⇒ \$s1 = 0xFFEAAEF3

Câu 36. Cho câu lệnh hợp ngữ sra \$s1, \$s2, 10, với \$s2 = 0x1234aabb, xác định \$s1?

⇒ \$s1 = 0xFFC48D2A

Câu 37. Cho dãy bit 0000000000100101000101010000000, hãy tìm câu lệnh hợp ngữ tương ứng với dãy bit đó?

- op = 000000(2) = 0(hex)
- funct = 000000(2) = 0(hex) ⇒ Lệnh sll
- dạng R-type
 - + rs = 00000(2) = 0(10): vậy địa chỉ nền được lưu trong thanh ghi thứ 0, chính là thanh ghi \$zero
 - + rt = 10010(2) = 18(10): thanh ghi thứ 18, chính là thanh ghi \$s2.
 - + rd = 10001(2) = 17(10): thanh ghi thứ 17, chính là thanh ghi \$s1.

shamt = 01010(2) = 10

⇒ sll \$s1, \$s2, 10.

Câu 38. Cho số hexa 0x02728806. Hãy dịch sang lệnh hợp ngữ?

- op = 001001(2) = 9(hex) lệnh addiu

- dạng I-type
- + rs = 11001(2) = 25(10) vậy địa chỉ được lưu trên thanh ghi thứ 25, là thanh ghi \$t9
- + rt = 01000(2) = 8(10): thanh ghi thứ 8, chính là thanh ghi \$t0.

Immediate(16 bit) = 0x8600
 \Rightarrow addiu \$t0, \$t9, 0x8600

Câu 39. Dùng lệnh dịch thực hiện phép tính sau $\$s1 * 32$? 40. Đoạn code hợp ngữ sau đây thể hiện phép nhân nào?

sll \$t0, \$s1, 2

sll \$t1, \$s1, 5

addu \$s2, \$t0, \$t1

- $\$s1 * 32$ sử dụng lệnh dịch \Rightarrow sll \$t0, \$s1, 5

Câu 40. Đoạn code hợp ngữ sau đây thể hiện phép nhân nào? sll \$t0, \$s1, 2 sll \$t1, \$s1, 5 addu \$s2, \$t0, \$t1

- sll \$t0, \$s1, 2 \$t0 = \$s1 * 4
- sll \$t1, \$s1, 5 \Rightarrow \$t1 = \$s1 * 32
- addu \$s2, \$t0, \$t1 \$s2 = \$s1 * 36

Câu 41. Trong kiến trúc tập lệnh MIPS, kết quả thực hiện phép nhân được lưu trong thanh ghi nào?

- Phép nhân được lưu trên thanh ghi Hi và Lo

Câu 42. Trong kiến trúc tập lệnh MIPS, kết quả thực hiện phép chia được lưu trong thanh ghi nào?

- Phép chia được lưu trên thanh ghi Hi và Lo: Hi lưu số dư, Lo lưu thương

Câu 43. Tập lệnh MIPS có lệnh subi và subiu không? Tại sao?

- không có lệnh subi (tức trừ một thanh ghi với một số tức thời) vì đã có lệnh addi và số tức thời trong addi có thể âm hoặc dương, nên subi không cần thiết.

Câu 44. Hằng số 16 bit trong lệnh andi là số có dấu hay không dấu? ví dụ?

- Giả sử giá trị đang chứa trong thanh ghi \$t1 là 0x12345678.
- Kết quả: sau lệnh trên, giá trị thanh ghi \$t0 = 0x00005678
- Quy trình lệnh thực hiện:
Số tức thời: 0xffff = 1111 1111 1111 1111 (2)
ZeroExtImm(0xffff) = 0000 0000 0000 0000 1111 1111 1111 1111 (2)
\$t0 = \$t1 & ZeroExtImm(0xffff) = 0x00005678

Câu 45. Xác định mã máy câu lệnh hợp ngữ sau addiu \$s2, \$s1, -1?

- addiu \$s2, \$s1, -1
op = 9(16) = 001001 dạng I-type

$rt = \$s2 = 18(10) = 10010$
 $rs = \$s1 = 17(10) = 10001$
 $immediate = -1(10) = 1111111111111111$
 \Rightarrow mã máy : 001001 10010 10001 11111111111111

Câu 46. Cho đoạn code: `addiu $s1, $0, 0xAC51` `sll $s1, $s1, 16` `ori $s1, $s1, 0x65D9` Xác định \$s1?

`addiu $s1, $0, 0xAC51` | $\$s1 = 0xAC51 = 000000\ 00000\ 00000$
 1010110001010001
`sll $s1, $s1, 16` | $\$s1 = 1010\ 1100\ 0101\ 0001\ 0000\ 0000$
 $0000\ 0000$
`ori $s1, $s1, 0x65D9` | $0x65D9 = 0000\ 0000\ 0000\ 0000\ 0110\ 0101$
 $1101\ 1001$
 $\Rightarrow \$s1 = 1010\ 1100\ 0101\ 0001\ 0110\ 0101$
 $1101\ 1001$

Câu 47. Cho đoạn code: `lui $s1, 0xAC51` `ori $s1, $s1, 0x65D9` Xác định \$s1?

`lui $s1, 0xAC51` | $0xAC51 = 000000\ 00000\ 00000\ 1010110001010001$
 $\$s1 =$
`ori $s1, $s1, 0x65D9` | $0x65D9 = 0000\ 0000\ 0000\ 0000\ 0110\ 0101\ 1101$
 1001
 $\Rightarrow \$s1 = 1010\ 1100\ 0101\ 0001\ 0110\ 0101$
 $1101\ 1001$

*****Câu 48.** Trong kiến trúc tập lệnh MIPS, định dạng lệnh J-Type trường hằng số có độ dài bao nhiêu? Giá trị thanh ghi PC thay đổi thế nào sau khi thực hiện lệnh J?

- cấu trúc J-Type, có thể nhảy trong khoảng 2^{26} từ nhớ ($\sim 2^{28}$ byte)

Câu 49. Giải thích câu lệnh sau `beq Rs, Rt, label`. Giá trị PC bằng bao nhiêu sau khi thực hiện lệnh trên?

- `Beq`: câu lệnh rẽ nhánh có điều kiện.
 - `Label`: câu lệnh rẽ nhánh không điều kiện.
- > Các lệnh rẽ nhánh có thể nhảy tới các địa chỉ có khoảng cách $\pm 2^{15}$ từ nhớ từ PC ($\pm 2^{17}$ bytes).

Câu 50. Giả sử $\$s0 = 1$ và $\$s1 = -1 = 0xffffffff$. Lệnh `slt $t0, $s0, $s1` cho kết quả \$t0 bằng bao nhiêu?

- Giả sử $\$s0 = 1$ và $\$s1 = -1 = 0xffffffff$. Lệnh `slt $t0, $s0, $s1` vì $\$s0 > \$s1 \Rightarrow \$t0 = 0$;

Câu 51. Thực hiện chuyển phát biểu IF `if (($s1 > 0) && ($s2 < 0)), {$s3++};` sang hợp ngữ MIPS?


```
IF if (($s1>0) && ($s2<0)) {$s3++;} sang hợp ngữ MIPS
blez $s1, next
bgez $s2, next
addiu $s3,$s3,1
```

next:

Câu 52.Vi xử lý mips truy cập bộ nhớ bằng câu lệnh nào? Ví dụ?

- Truy cập bộ nhớ bằng câu lệnh lw và sw

ví dụ: lw \$t0, 12 (\$s0) . sw \$t0, 12 (\$s0)

Câu 53.Giả sử có lệnh lw 8(\$t0). Vùng nhớ có địa chỉ bằng?

- lw 8(\$t0). Vùng nhớ có địa chỉ bằng \$t0 + 8

Câu 54.Giải thích câu lệnh sw \$t0, 12(\$s0)?

- câu lệnh sw \$t0, 12(\$s0) Lệnh này lưu giá trị trong thanh ghi \$t0 vào ô nhớ có địa chỉ (\$s0 + 12)

Câu 55.Giả sử A là mảng các từ nhớ, \$s3 là địa chỉ bắt đầu của A. Lệnh lw \$t0, 20(\$s3) cho ra kết quả \$t0?

- A là mảng các từ nhớ, \$s3 là địa chỉ bắt đầu của A. Lệnh lw \$t0, 20(\$s3) cho ra kết quả \$t0 = A[5]

Câu 56.Giả sử A là mảng các từ nhớ, g, h được ánh xạ lần lượt là \$s1, \$s2. \$s3 là địa chỉ bắt đầu của A. Câu lệnh g = h + A[5] được biên dịch thành lệnh hợp ngữ MIPS như thế nào?

```
lw $t0, 20($s3) #$t0 gets A[5]
add $s1, $s2, $t0 #$s1 = h+A[5]
```

Câu 57.Giải thích câu lệnh lb \$s0, 3(\$s1)?

- câu lệnh lb \$s0, 3(\$s1) – Lệnh này nạp giá trị byte nhớ có địa chỉ (\$s1 + 3) vào byte thấp của thanh ghi \$s0.

Câu 58.Viết đoạn code hợp ngữ mips thể hiện biểu thức $A[300] = h + A[300]$?

- thể hiện biểu thức $A[300] = h + A[300]$, h3= \$s1
- ```
lw $t0, 1200($s3) #$t0=A[300]
add $t0, $s1, $t0 #$t0 = $s1 + $t0
```

**Câu 59.**Xác định 1 lệnh hợp ngữ mips thể hiện đoạn code C if (\$s2 < \$s3) \$s1 = 1; else \$s1 = 0;?

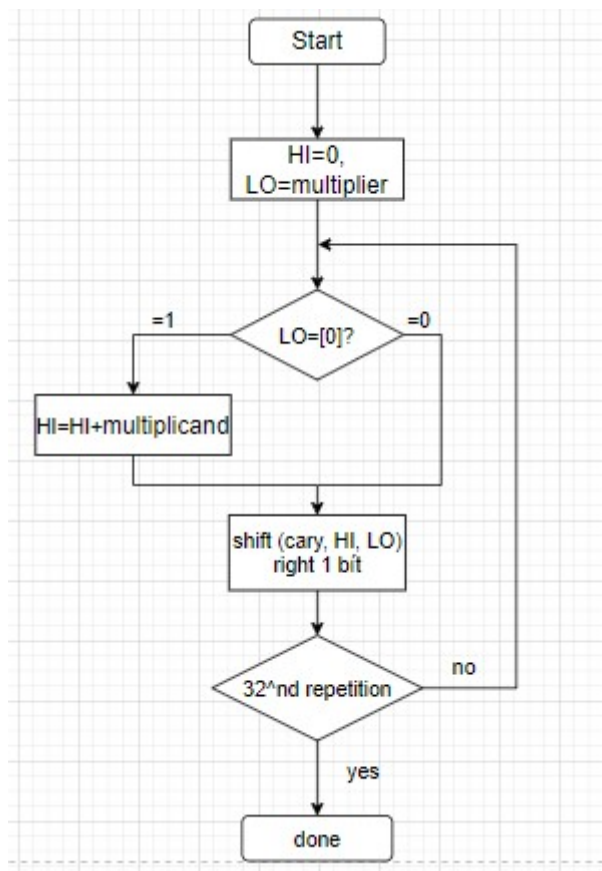
```
slt $t1, $s2, $s3
beq $t1, 0, lenh
lenh:
add $s1,$zero, 1
j next
add $s1, $zero, 0
next:
```

Câu 60. Vi xử lý MIPS hỗ trợ lưu trữ dữ liệu theo dạng big endians hay little endians?

- MIPS lưu trữ dữ liệu theo dạng Big Endian, tức là byte cao sẽ được lưu ở địa chỉ thấp

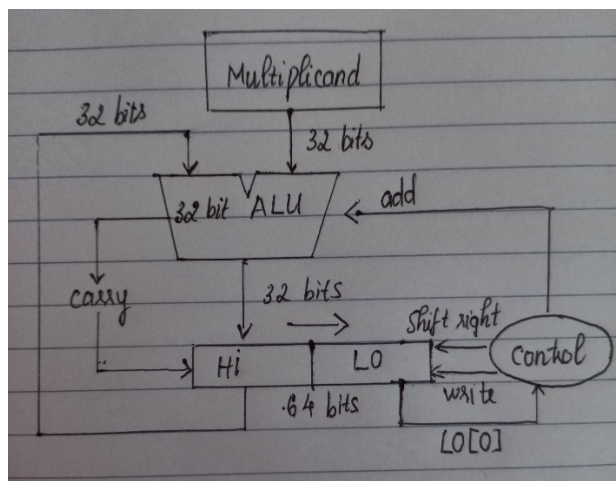
## Chương 6:

Câu 1. Vẽ lưu đồ thuật toán thực hiện phép nhân với số nguyên không dấu? Giải thích.



- khởi tạo HI=0, LO=multiplier (hệ số nhân)
- Sau khi khởi tạo xong. Mỗi vòng lặp (iteration) sẽ gồm 2 bước:
  - B1. Kiểm tra bit 0 của multiplier xem có bằng 1 hay không; nếu bằng 1 thì  $product = product + multiplicand$ ; nếu bằng 0, không làm gì cả
  - B2. Dịch phải Multiplier 1 bit
- Số vòng lặp cho giải thuật này đúng bằng số bit dùng biểu diễn (ví dụ 1 yêu cầu dùng số 4 bit, thì có 4 vòng lặp)
- Sau khi kết thúc số vòng lặp, giá trị trong thanh ghi product chính là kết quả phép nhân

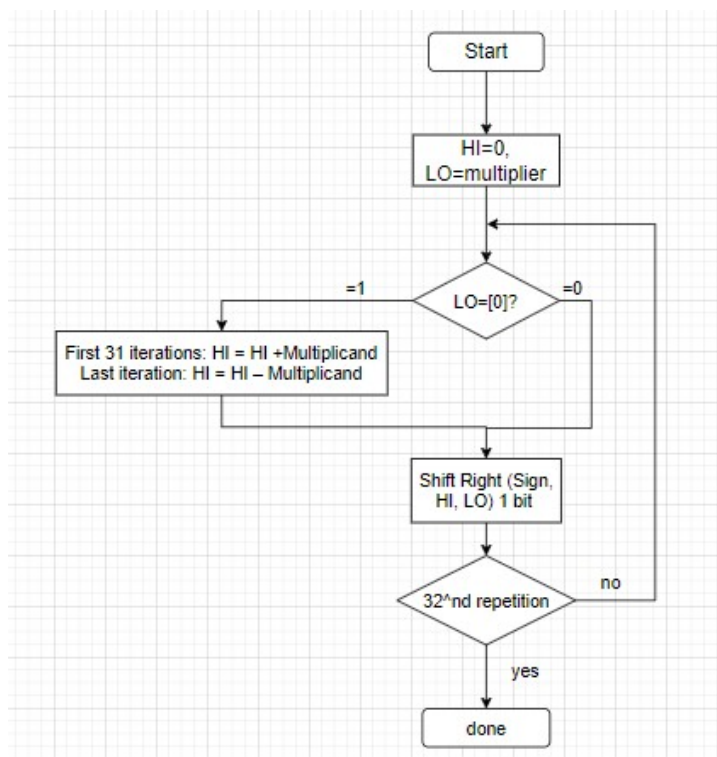
Câu 2. Vẽ datapath thực hiện phép nhân không dấu? Giải thích các tín hiệu và các thành phần datapath?



Câu 3. Cho phép tính  $1110 \times 1010$ . Triển khai các bước vi xử lý thực hiện phép nhân? Xác định kết quả? (giả sử 2 số hạng trên có độ rộng 4 bit)

| iteration |                                                                 | Multiplicand | Carry | Product = HI, LO       |
|-----------|-----------------------------------------------------------------|--------------|-------|------------------------|
| 0         | Initialize (HI = 0, LO = Multiplier)                            | 1110         |       | 0000 1010              |
| 1         | LO[0] = 0 => Do Nothing<br>Shift Right (Carry, HI, LO) by 1 bit | 1110         |       | 0000 0101              |
| 2         | LO[0] = 1 => add<br>Shift Right (Carry, HI, LO) by 1 bit        | 1110         | + 0   | 1110 0101<br>0111 0010 |
| 3         | LO[0] = 0 => Do Nothing<br>Shift Right (Carry, HI, LO) by 1 bit | 1110         |       | 0011 1001              |
| 4         | LO[0] = 1 => ADD<br>Shift Right (Carry, HI, LO) by 1 bit        | 1110         | + 1   | 0001 1001<br>1000 1100 |

Câu 4. Vẽ lưu đồ thuật toán thực hiện phép nhân với số nguyên có dấu?  
Giải thích.



**Câu 5. Vẽ datapath thực hiện phép nhân có dấu? Giải thích các tín hiệu và các thành phần datapath?**

**\*\*\*Câu 6. Cho phép tính  $1100 (-4) \times 1010 (-3)$ . Triển khai các bước vi xử lý thực hiện phép nhân? Xác định kết quả? (giả sử 2 số hạng trên có độ rộng 4 bit)**

| iteration |                                             | Multiplicand | sign | Product = HI, LO |
|-----------|---------------------------------------------|--------------|------|------------------|
| 0         | Initialize (HI = 0, LO = Multiplier)        | 1100         |      | 0000 1010        |
| 1         | $LO[0] = 0 \Rightarrow$ Do Nothing          |              | +    |                  |
|           | Shift Right (sign, HI, LO) by 1 bit         | 1100         |      | 0000 0101        |
| 2         | $LO[0] = 1 \Rightarrow$ add                 |              | + 0  | 1100 0101        |
|           | Shift Right (sign, HI, LO) by 1 bit         | 1100         |      | 0110 0010        |
| 3         | $LO[0] = 0 \Rightarrow$ Do Nothing          |              |      |                  |
|           | Shift Right (sign, HI, LO) by 1 bit         | 1100         |      | 0011 0001        |
| 4         | $LO[0] = 1 \Rightarrow$ SUB (ADD 2's compl) | 0100         | + 0  | 0111 0001        |
|           | Shift Right (sign, HI, LO) by 1 bit         |              |      | 0011 1000        |

## Chương 7

Câu 1. Khái niệm về dấu phẩy động? Các số  $5,341 \times 10^3$  ;  $55,41 \times 10^2$  ;  $3,1415$ ;  $314,15 \times 10^{-2}$  có phải là số dấu phẩy động hay không?

- Dấu phẩy động được dùng để chỉ một hệ thống biểu diễn số mà trong đó sử dụng một chuỗi chữ số (hay bit) để biểu diễn một số hữu tỉ.
- các số trên đều là số dấu phẩy động trong hệ cơ số 10

Câu 2. Khái niệm về số dấu phẩy động được chuẩn hóa? Các số  $3,1415$ ;  $314,15 \times 10^{-2}$  ,  $1,1101 \times 2^5$  ;  $111,01 \times 2^3$  ;  $0,11101 \times 2^6$  có phải là số dấu phẩy động được chuẩn hóa không? Giải thích?

- Số dấu phẩy động được chuẩn hóa là số mà trước dấu phẩy phải xuất hiện một chữ số khác không. ở dạng số thập phân, chữ số này có thể từ 1 đến 9, trong số nhị phân chữ số này phải là 1
- Các số  $3,1415$ ;  $314,15 \times 10^{-2}$  là số dấu phẩy động được chuẩn hóa hệ 10 vì trước dấu phẩy là số khác 0
- Các số  $1,1101 \times 2^5$  ;  $111,01 \times 2^3$  là số dấu phẩy động được chuẩn hóa hệ nhị phân vì trước dấu phẩy là số 1
- Số  $0,11101 \times 2^6$  không là số dấu phẩy động được chuẩn hóa vì trước dấu phẩy là số 0

Câu 3. Người ta nói rằng một số dấu phẩy động được biểu diễn bằng ba trường bit S (sign), E (exponent), F (fraction) có đúng không? Giải thích ý nghĩa từng trường bit đó?

\*Đúng

\*Ý nghĩa của từng trường :

- S là bit dấu ( 0 là dương , 1 là âm), biểu diễn được gọi là dấu và độ lớn
- E là trường số mũ (có dấu):
  - + các số rất lớn có số mũ dương lớn
  - + các số rất nhỏ gần bằng 0 có số mũ âm nhỏ
  - + nhiều bit hơn trong trường số mũ tăng phạm vi giá trị
- F là trường phân số (phần số sau dấu nhị phân)
  - + nhiều bit hơn trong trường phân số sẽ cải thiện độ

chính xác của số FP

Câu 4. Theo chuẩn IEEE 754, số dấu phẩy động chính xác đơn được biểu diễn như thế nào? Giải thích ý nghĩa các trường bit có trong cách biểu diễn đó?

- Số dấu phẩy động chính xác đơn được biểu diễn như sau:  
Dấu 1 bit + số mũ 8 bit + phần số 23 bit

Câu 5. Theo chuẩn IEEE 754, số dấu phẩy động chính xác kép được biểu diễn như thế nào? Giải thích ý nghĩa các trường bit có trong cách biểu diễn đó?

- Số dấu phẩy động chính xác kép được biểu diễn như sau:  
Dấu 1 bit + số mũ 11 bit + phần số 52 bit
- 

**Câu 6. Giá trị của số dấu phẩy động được tính theo công thức nào? Giải thích ý nghĩa?**

- Giá trị của số dấu phẩy động được tính theo công thức:  $(-1)^S \times \text{val}(F) \times 2^{(\text{val}(E))} = (-1)^S \times 1.F \times 2^{(e-127)}$   
S là bit dấu (số dương S = 0)  
e là mã excess của phần mũ E (e = E+127 hay E = e-127, số 127 ở đây là độ lệch bias)  
F là phần lẻ của phần định trị F (F = 1.f)

**Câu 7. Khái niệm bias trong biểu diễn dấu phẩy động được hiểu như thế nào?**

- bias trong biểu diễn dấu phẩy động để biểu diễn một số mũ có dấu
- Giá trị của số mũ =  $\text{val}(E) = E - \text{Bias}$  (Bias là một hằng số)

**Câu 8. Chuẩn IEEE 754 biểu diễn dấu phẩy động chính xác đơn có độ rộng trường E = 8 bit, điều này có ý nghĩa như thế nào?**

Ý nghĩa của chuẩn IEEE 754 biểu diễn dấu phẩy động chính xác đơn có độ rộng trường E = 8 bit

- E có thể nằm trong khoảng 0 đến 255
- E = 0 và E = 255 được dành riêng cho mục đích sử dụng đặc biệt
- E = 1 đến 254 được sử dụng cho các số dấu phẩy động chuẩn hóa

**Câu 9. Đối với vi xử lý mips, các số dấu phẩy động thường là bội số của một word, bit, hay byte?**

- Đối với vi xử lý mips, các số dấu phẩy động thường là bội số của một bit.

**Câu 10. Bao nhiêu bit được sử dụng để quyết định độ lệch (bias) trong biểu diễn dấu phẩy động chính xác kép?**

- Có 11 bit được sử dụng để quyết định độ lệch trong biểu diễn dấu phẩy động chính xác kép

**Câu 11. Sự không rõ ràng trong dấu phẩy động được biểu diễn dưới dạng nào?**

**Câu 12. Tìm biểu diễn dấu phẩy động chính xác đơn của số thập phân -0,15625?**

- $0.15625 \times 2 = 0.3125$
- $0.3125 \times 2 = 0.625$

- $0.625 \times 2 = 1.25$
- $0.25 \times 2 = 0.5$
- $0.5 \times 2 = 1.0$

→  $0.15625$  hệ 10 =  $0.00101$  hệ 2

Fraction:  $(0.00101)$  hệ 2 =  $(1.01)$  hệ 2  $\times 2^{-3}$

Exponent:  $-3 + 127$  (bias) =  $124$  hệ 10 =  $01111100$  hệ 2

Là số âm nên bit dấu = 1

→ biểu diễn dấu phẩy động chính xác đơn của số thập phân

$-0,15625$  là  $101111100010$  00000000000000000000

**Câu 13. Giá trị thập phân của dãy bit biểu diễn dấu phẩy động chính xác đơn  $0100\ 0001\ 0010\ 0110\ 0000\ 0000\ 0000\ 0000$  bằng bao nhiêu**

- Dấu của số : bit dấu  $S=0$  → là số dương
- Phần mũ  $E$ : mã lệch  $e = 10000010(2) = 130(10)$   
→  $E = e - 127 = 130 - 127 = 3$
- Phần định trị  $1.f = 1.011(2) = 1.375(10)$
- Ta có :  $1.f \times 2^{(e - 127)} = 1.375 \times 2^3 = 11$

**Câu 14. Thực hiện phép tính cộng 2 số dấu phẩy động đã chuẩn hóa với độ chính xác 4 bit sau:  $(1,111)$  hệ 2  $\times 2^{-1}$  +  $(1,011)$  hệ 2  $\times 2^{-3}$**

- $1.011 \times 2^{-3} = 0.01011 \times 2^{-1}$  → hiệu giữa 2 số mũ =  $-1 - (-3) = 2$

Vì vậy dịch sang phải 2 bit

1.111

+0.01011

-----

10.00111

Ta được  $1.111 \times 2^{-1} + 1.011 \times 2^{-3} = 10.00111 \times 2^{-1}$

- Chuẩn hóa kết quả:  $10.00111 \times 2^{-1} = 1.000111 \times 2^0$

**Câu 15. Thực hiện phép tính trừ 2 số dấu phẩy động đã chuẩn hóa với độ chính xác 4 bit sau:  $(1,000)$  hệ 2  $\times 2^{-3}$  –  $(1,000)$  hệ 2  $\times 2^2$**

- Hiệu giữa 2 số mũ =  $2 - (-3) = 5$
- Dịch sang phải 5 bit:  $1.000 \times 2^{-3} = 0.00001000 \times 2^2$
- Chuyển phép trừ thành phép cộng thành phần bù của bù 2

+  $0.00001 \times 2^2$

–  $1.00000 \times 2^2$

-----

0  $0.00001 \times 2^2$

1  $1.00000 \times 2^2$

-----

1  $1.00001 \times 2^2$  → bù 2 :  $- 0.11111 \times 2^2$

Ta được :  $(1.000)$  hệ 2  $\times 2^{-3}$  –  $(1.000)$  hệ 2  $\times 2^2 = - 0.11111 \times 2^2$

Chuẩn hóa kết quả :  $- 0.11111 \times 2^2 = -1.1111 \times 2^1$

**Câu 16.**Thực hiện phép tính nhân 2 số dấu phẩy động đã chuẩn hóa với độ chính xác 4 bit sau: (1,010 hệ  $2 \times 2^{(-1)}$ )  $\times$  (-1,110 hệ  $2 \times 2^{(-2)}$ )

- Giá trị lũy thừa kết quả :  $(-1) + (-2) = -3$
- Ta có :  $E_z = E_x + E_y - \text{bias}$   
 $E_x = (-1) + 127 = 126$  (Bias = 127 for SP)  
 $E_y = (-2) + 127 = 125$   
 $E_z = 126 + 125 - 127 = 124$  ( giá trị  $= -3$ )

```

1.010
+1.110

0000
1010
1010
1010

10001100

```

→  $1,010 \times 1,110 = 10,001100$

- Ta có dấu của  $S_z = 1$  (âm)  
→ Vì vậy  $1,010 \times 2^{(-1)} \times (-1,110) \times 2^{(-2)} = (-10,001100) \times 2^{(-3)}$
- Chuẩn hóa  $-10,001100) \times 2^{(-3)} = -1,0001100 \times 2^{(-2)}$

**Câu 17.**Câu lệnh hợp ngữ MIPS lwc1 \$f2, 40(\$t0) có nghĩa như thế nào?

- Nạp từ vào bộ đồng xử lý 1 ( FPU) : đặt giá trị \$f2 thành 32 bit từ địa chỉ \$t0+40

**Câu 18.**Câu lệnh hợp ngữ MIPS swc1 \$f2, 40(\$t0) có nghĩa như thế nào?

- Lưu trữ giá trị 32 bit trong thanh ghi \$f2 đến địa chỉ \$t0+40

**Câu 19.**Muốn đọc (nạp) số dấu phẩy động 32 bit vào thanh ghi \$f4 thực hiện như thế nào?

- Lwc1 \$f4, 40(\$t0)

## Chương 8

**Câu 1.** Nêu khái niệm về Response time, throughput?

- Response time ( thời gian đáp ứng): 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 (operating system over)

$$\text{Response Time} = \text{CPU Time} + \text{Waiting Time}$$



- Throughput: số lượng tác vụ máy có thể chạy trong một khoảng thời gian nhất định

**Câu 2. Để giảm thông số response time có thể thực hiện bằng cách nào?**

- Để giảm thông số response time thực hiện bằng cách: sử dụng ít thời gian hơn để chạy 1 tác vụ → có thể thực hiện được nhiều nhiệm vụ hơn

**Câu 3. Khái niệm về CPI, CPU Execution time, clock cycle, clock rate?**

- CPI: Số chu kỳ xung clock cần để thực thi một lệnh
- CPU Execution time; thời gian đã sử dụng trong khi thực hiện các hướng dẫn chương trình
- Clock cycle: chu kỳ xung đồng hồ
- Clock rate: số clock cycle thực hiện trong 1 giây (Hz)

**Câu 4. Công thức sau có ý nghĩa như thế nào?**

$$\frac{\text{Performance}_x}{\text{Performance}_y} = \frac{\text{Execution time}_y}{\text{Execution time}_x} = n$$

- X nhanh gấp n lần Y

**Câu 5. Bộ vi xử lý A có mức tiêu thụ năng lượng cao hơn 20% so với bộ vi xử lý B, nhưng nếu bộ vi xử lý A thực hiện nhiệm vụ chỉ trong 70% thời gian trung bình mà B cần, thì mức tiêu thụ năng lượng của nó sẽ là bao nhiêu?**

**Câu 6. Một vi xử lý chạy ở mức sử dụng 100% là 100W, trong đó 20% công suất được quy cho rò rỉ. Tổng công suất tiêu tán khi đang chạy với mức sử dụng 50% là bao nhiêu?**

**Câu 7. Nếu máy tính A chạy một chương trình mất 10 giây và máy tính B cũng chạy chương trình đó mất 15 giây, vậy máy tính A chạy nhanh hơn máy tính B bao nhiêu lần?**

- $\text{Execution time}_B / \text{Execution time}_A = 1.5;$

⇒ A nhanh gấp 1.5 lần Y

**Câu 8. Người ta nói 1 chu kỳ lệnh bao gồm chu kỳ lấy lệnh, chu kỳ thực thi lệnh, chu kỳ giải mã lệnh có đúng không? Giải thích từng chu kỳ?**

1 chu kỳ lệnh bao gồm chu kỳ lấy lệnh, chu kỳ thực thi lệnh, chu kỳ giải mã lệnh

- Chu kỳ lấy lệnh: Trong đó lệnh được bắt từ RAM và được sao chép vào bên trong bộ xử lý.
- Chu kỳ giải mã lệnh: Trong đó lệnh được bắt trước đó được giải mã và gửi đến các đơn vị thực thi

- Chu kỳ thực thi lệnh: Nơi lệnh được giải quyết và kết quả được ghi trong thanh ghi bên trong của bộ xử lý hoặc trong địa chỉ bộ nhớ của RAM

**Câu 9.** Giả sử có lệnh li (i nhận giá trị từ 1 đến 7). Trong đó l1, l4, l5 mỗi lệnh chiếm 1 chu kỳ; l2 chiếm 4 chu kỳ; l3, l6 mỗi lệnh chiếm 2 chu kỳ, l7 chiếm 3 chu kỳ. CPI (Clock Cycles per Instruction) bằng bao nhiêu?

Ta có 7 lệnh và 14 chu kỳ  
 $\Rightarrow \text{CPI} = 14/7=2$ ;

## Chương 9

**Câu 1.** Nếu các bước thực hiện thiết kế 1 bộ vi xử lý đơn chu kỳ?

Các bước thực hiện :

- Bước 1: phân tích tập lệnh  $\rightarrow$  xác định thành phần của datapath
- Bước 2: thiết kế, lựa chọn các thành phần của datapath và phương pháp cấp xung nhịp
- Bước 3: gắn các thành phần của datapath đáp ứng yêu cầu công việc của từng lệnh
  - + Xác định các giá trị của các tín hiệu điều khiển cho việc điều khiển dòng lưu chuyển của dữ liệu
- Bước 4: thiết kế và thêm vào bộ điều khiển

**Câu 2.** Nêu các tập lệnh con MIPS trong quá trình thiết kế vi xử lý đơn chu kỳ?

| Hướng dẫn                     | Ý nghĩa           | Sự sắp xếp |        |        |                  |   |      |
|-------------------------------|-------------------|------------|--------|--------|------------------|---|------|
| add rd, rs, rt                | Phép cộng         | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x20 |
| Sub rd, rs, rt                | Phép trừ          | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x22 |
| And rd, rs, rt                | Bitwise và        | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x24 |
| Or rd, rs, rt                 | Bitwise hoặc      | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x25 |
| Xor rd, rs, rt                | Loại trừ hoặc     | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x26 |
| Slt rd, rs, rt                | Đặt trên ít hơn   | op[6] = 0  | rs [5] | rt [5] | rd[5]            | 0 | 0x2a |
| Addi rt, rs, im <sup>16</sup> | Thêm ngay lập tức | 0x08       | rs [5] | rt [5] | im <sup>16</sup> |   |      |
| Slti rt, rs, im <sup>16</sup> | Slt ngay          | 0x0a       | rs [5] | rt [5] | im <sup>16</sup> |   |      |
| Andi rt, rs, im <sup>16</sup> | Và ngay lập tức   | 0x0c       | rs [5] | rt [5] | im <sup>16</sup> |   |      |
| Ori rt, rs, im <sup>16</sup>  | Hoặc ngay         | 0x0d       | rs [5] | rt [5] | im <sup>16</sup> |   |      |

|                              |                         |      |                  |        |                  |
|------------------------------|-------------------------|------|------------------|--------|------------------|
|                              | lập tức                 |      |                  |        |                  |
| Xori rt, im <sup>16</sup>    | Loại trừ ngay lập tức   | 0x0e | rs [5]           | rt [5] | Im <sup>16</sup> |
| Lw rt im <sup>16</sup> (rs)  | Nạp từ                  | 0x23 | rs [5]           | rt [5] | Im <sup>16</sup> |
| Sw rt im <sup>16</sup> (rs)  | Lưu trữ từ              | 0x2b | rs [5]           | rt [5] | Im <sup>16</sup> |
| beq rs, rt, im <sup>16</sup> | Rẽ nhánh nếu bằng nhau  | 0x04 | rs [5]           | rt [5] | Im <sup>16</sup> |
| bne rs, rt, im <sup>16</sup> | Rẽ nhánh nếu không bằng | 0x05 | rs [5]           | rt [5] | Im <sup>16</sup> |
| j im <sup>26</sup>           | Nhảy                    | 0x02 | Im <sup>16</sup> |        |                  |

**Câu 3. Thanh ghi PC trong vi xử lý có ý nghĩa như thế nào? Tại sao 2 bit cuối (ít ý nghĩa nhất) của thanh ghi PC luôn là '00'? Khi thực hiện lệnh add Rd, Rs, Rt thì thanh ghi PC thay đổi như thế nào?**

**Câu 4. Trình bày các bước thực thi lệnh loại R, I? Ví dụ về lệnh Lw?**

Các bước thực hiện lệnh loại R, I:

- Loại R:
  - + hướng dẫn tìm nạp: Instruction  $\leftarrow$  MEM[PC]
  - + tìm nạp toán hạng : data1  $\leftarrow$  Reg(Rs), data2  $\leftarrow$  Reg(Rt)
  - + thực hiện hoạt động : ALU\_result  $\leftarrow$  func(data1, data2)
  - + ghi tên kết quả ALU: Reg(Rd)  $\leftarrow$  ALU\_result
  - + địa chỉ PC tiếp theo : PC  $\leftarrow$  PC + 4

- Loại I:
  - + hướng dẫn tìm nạp: Instruction  $\leftarrow$  MEM[PC]
  - + tìm nạp toán hạng : data1  $\leftarrow$  Reg(Rs), data2  $\leftarrow$

Extend(imm16)

- + thực hiện thao tác : ALU\_result  $\leftarrow$  op(data1, data2)
- + viết kết quả ALU: Reg(Rt)  $\leftarrow$  ALU\_result
- + địa chỉ PC tiếp theo: PC  $\leftarrow$  PC + 4

**Câu 5. Trình bày các thành phần yêu cầu từ tập lệnh?**

Các thành phần yêu cầu từ tập lệnh

- Bộ nhớ:
  - + bộ nhớ tập lệnh, nơi lưu trữ tập lệnh
  - + dữ liệu nơi dữ liệu được lưu trữ
- Thanh ghi:
  - + 31x32-bit thanh ghi mục đích chung, R0 luôn bằng 0
  - + đọc thanh ghi nguồn rs
  - + đọc thanh ghi nguồn rt
  - + ghi thanh ghi đích rt hoặc rd
- Thanh ghi PC bộ đếm chương trình và adder để tăng PC
- Bộ mở rộng sign và zero cho hằng số ngay lập tức
- ALU để thực hiện các hướng dẫn

**Câu 6. Datapath gồm những thành phần nào? Nêu chức năng của từng thành phần đó?**

Các thành phần của datapath:

- Các thành phần mạch tổ hợp : + ALU, adder  
+ bộ xử lý ngay lập tức  
+ bộ ghép lệnh
- Các thành phần lưu trữ : + bộ nhớ hướng dẫn  
+ bộ nhớ dữ liệu  
+ thanh ghi PC  
+ thanh ghi tệp
- Xung nhịp : + thời gian ghi

### Câu 7. Mô tả hoạt động truy xuất dữ liệu của thanh ghi?

Hoạt động truy xuất dữ liệu của thanh ghi:

- n bit vào và ra
- Write Enable (WE):  
+ cho phép/cấm ghi vào thanh ghi  
+ Cấm(0): Data\_out không đổi  
+ Cho phép (1): Data\_out = Data\_in sau cạnh lên xung nhịp

Xung nhịp kích cạnh lên (0->1)

+ giá trị out\_put được thay đổi tại cạnh lên của xung nhịp

### Câu 8. Trình bày ý nghĩa của tín hiệu vào/ra của bộ thanh ghi (tệp) register file?

Tín hiệu vào :

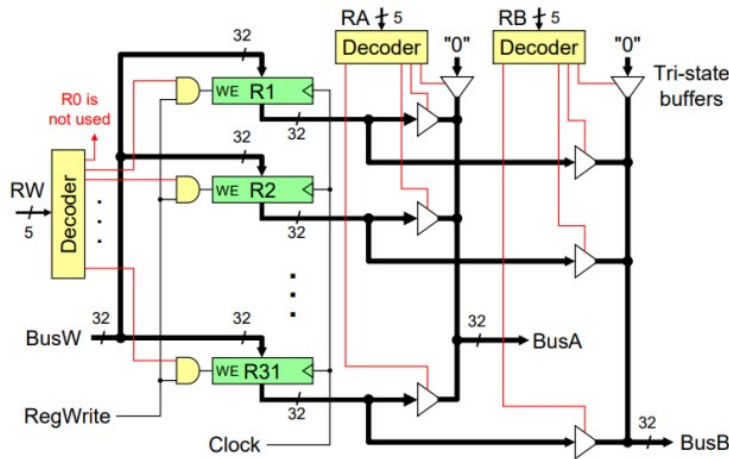
- RA lựa chọn thanh ghi đọc cho giá trị ở BusA
- RB lựa chọn thanh ghi đọc cho giá trị ở BusB
- RW lựa chọn thanh ghi được ghi vào

Tín hiệu ra :

- BusA và BusB : 32 bit ngõ ra cho 2 toán hạng nguồn
- BusW : 32 bit ngõ vào để ghi giá trị vào thanh ghi khi RegWrite=1

### Câu 9. Cho sơ đồ chi tiết bộ thanh ghi sau. Trình bày quá trình truy xuất dữ liệu thanh ghi? (đọc dữ liệu từ thanh ghi và ghi dữ liệu vào thanh ghi)?

## Details of the Register File



**Caau 10.Trình bày ý nghĩa của tín hiệu vào/ra của bộ nhớ lệnh và bộ nhớ dữ liệu (instruction memory và data memory)?**

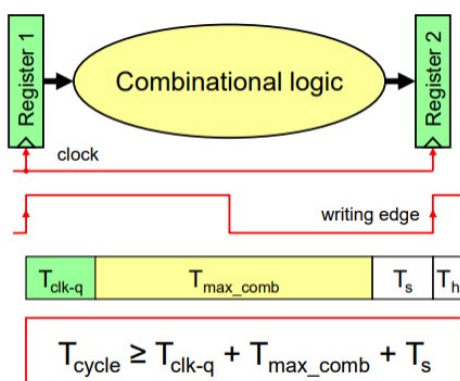
Instruction memory cung cấp quyền truy cập đọc:

- Vì đường dẫn dữ liệu không viết hướng dẫn
- Hành vi như phép toán Logic dùng để đọc
- Địa chỉ chọn chỉ dẫn sau thời gian truy cập

## Data memory dùng để tải và lưu trữ

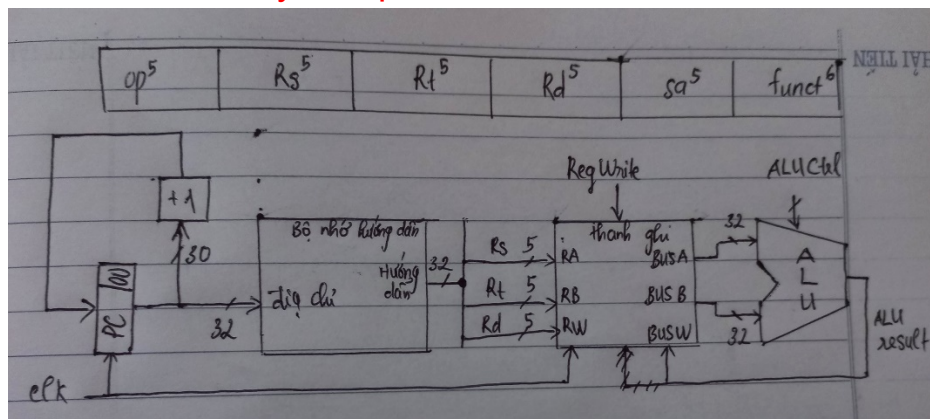
- MemRead: cho phép xuất trên Data-out  
Địa chỉ chọn từ để đưa vào Data-out
- MemWrite: cho phép ghi Data\_in  
Địa chỉ chọn từ bộ nhớ được ghi  
Đồng hồ đồng bộ hóa thao tác ghi

Câu 11. Có những phương pháp cấp xung nhịp nào? Cho hình minh họa phương pháp kích cạnh lên sau, giải thích ý nghĩa của  $T_{clk-q}$ ,  $T_{max\_comb}$ ,  $T_s$ ,  $T_h$ , và  $T_{cycle}$  ?



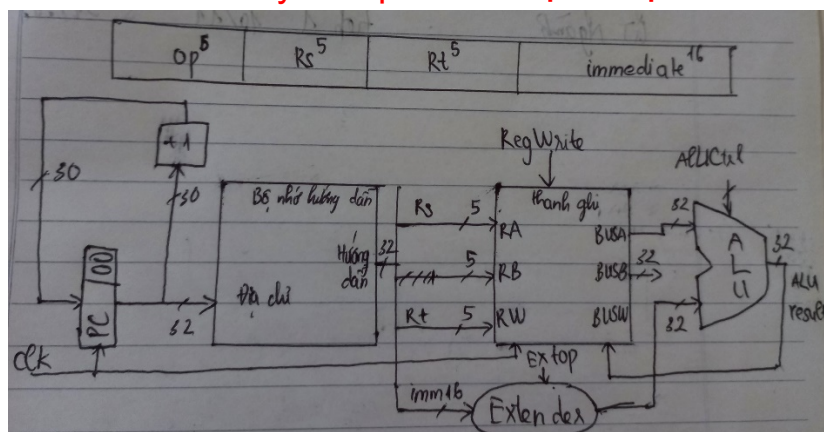
- Tclk-q: độ trễ đồng hồ đầu ra thông qua đăng ký
- Tmax\_comb: độ trễ lâu nhất thông qua logic tổ hợp
- Ts: thời gian thiết lập đầu vào cho một đăng ký phải ổn định trước khi đến cạnh đồng hồ
- Th: giữ thời gian mà đầu vào cho một đăng ký phải giữ sau sự xuất hiện của cạnh đồng hồ
- $T_{cycle} \geq T_{clk-q} + T_{max\_comb} + T_s$

## Câu 12. Trình bày datapath cho lệnh loại R?



- Trường Rs và Rt chọn 2 thanh ghi để đọc. trường Rd chọn thanh ghi để ghi
- Bus A và Bus B cung cấp đầu vào dữ liệu cho ALU. Kết quả ALU được kết nối với Bus W
- Cập nhật cùng một đồng hồ cho PC và thanh ghi Rd  
Tín hiệu điều khiển
- ALU Ctrl có nguồn gốc từ trường funct vì  $op = 0$  cho kiểu R  
→ Regwrite được sử dụng để cho phép ghi kết quả ALU

## Câu 13. Trình bày datapath cho lệnh loại I?



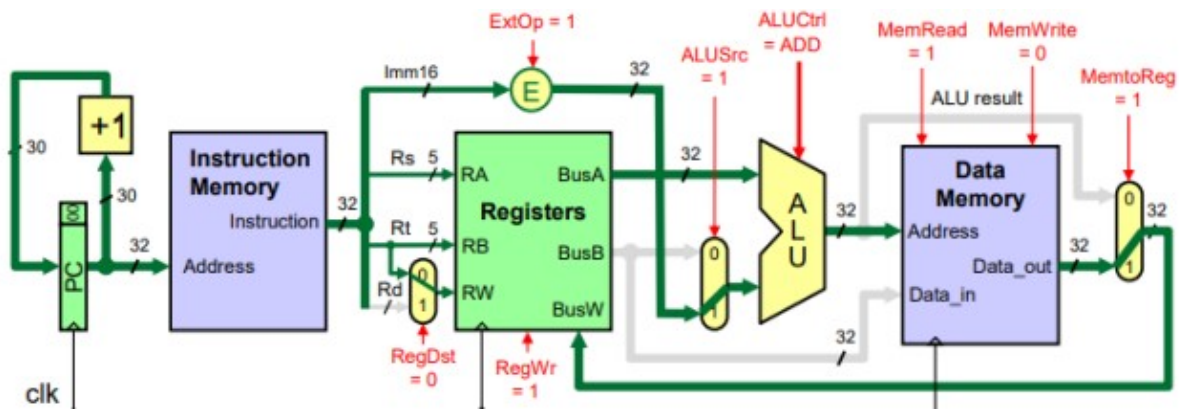
- Rt thanh ghi để viết, không phải Rd
- Đồng hồ giống nhau cập nhật cạnh PC và Rt
- Đầu vào ALU thứ hai đến từ phần mở rộng ngay tức khắc. RB và Bus B không được sử dụng

Tín hiệu điều khiển:

- ALU Ctrl có nguồn gốc từ trường op  
→ Regwrite được sử dụng để cho phép ghi kết quả ALU
- ExtOp được sử dụng để kiểm soát phần mở rộng của 16-bit ngay lập tức

## Câu 14. Cho datapath của lệnh load sau. Trình bày ý nghĩa các tín hiệu RegDst, RegWrite, ExtOp, ALUSrc, ALU Ctrl, MemRead, MemtoReg và clk?





- RegDst = '0' chọn Rt như đăng ký đích
- RegWrite = '1' để kích hoạt viết của tập tin đăng ký
- ExtOp = 1 đến ký mở rộng ngay lập tức từ 16 đến 32 bit
- ALUSrc = '1' chọn mở rộng ngay lập tức làm đầu vào ALU thứ hai
- ALU Ctrl = 'ADD' để tính toán bộ nhớ dữ liệu địa chỉ dưới dạng Reg (Rs) + mở rộng (Imm16)
- MemRead = '1' để đọc bộ nhớ dữ liệu
- MemtoReg = '1' đặt dữ liệu đọc từ bộ nhớ trên Bus
- Cập nhật đồng hồ PC và đăng ký Rt

**Câu 15.** Cho datapath của lệnh store sau. Trình bày ý nghĩa các tín hiệu RegDst, RegWrite, ExtOp, ALUSrc, ALU Ctrl, MemWrite, MemtoReg và clk?

- RegDst = 'X' vì không có thanh ghi nào được ghi
- = '0' để tắt ghi tập đăng ký
- ExtOp = 1 đến ký mở rộng ngay lập tức 16 đến 32 bit
- ALUSrc = '1' chọn mở rộng ngay lập tức như đầu vào ALU thứ hai
- ALU Ctrl = 'ADD' để tính toán bộ nhớ dữ liệu địa chỉ dưới dạng Reg (Rs) + mở rộng (Imm16)
- MemWrite = '1' thành ghi dữ liệu bộ nhớ
- MemtoReg = 'X' vì không quan tâm dữ liệu nào được đưa vào BusW
- Cập nhật cạnh đồng hồ PC và bộ nhớ dữ liệu

**Câu 16.** Trình bày hạn chế của thiết kế vi xử lý đơn chu kỳ?

- Hạn chế của thiết kế vi xử lý đơn chu kỳ là: Thời gian chu kỳ dài