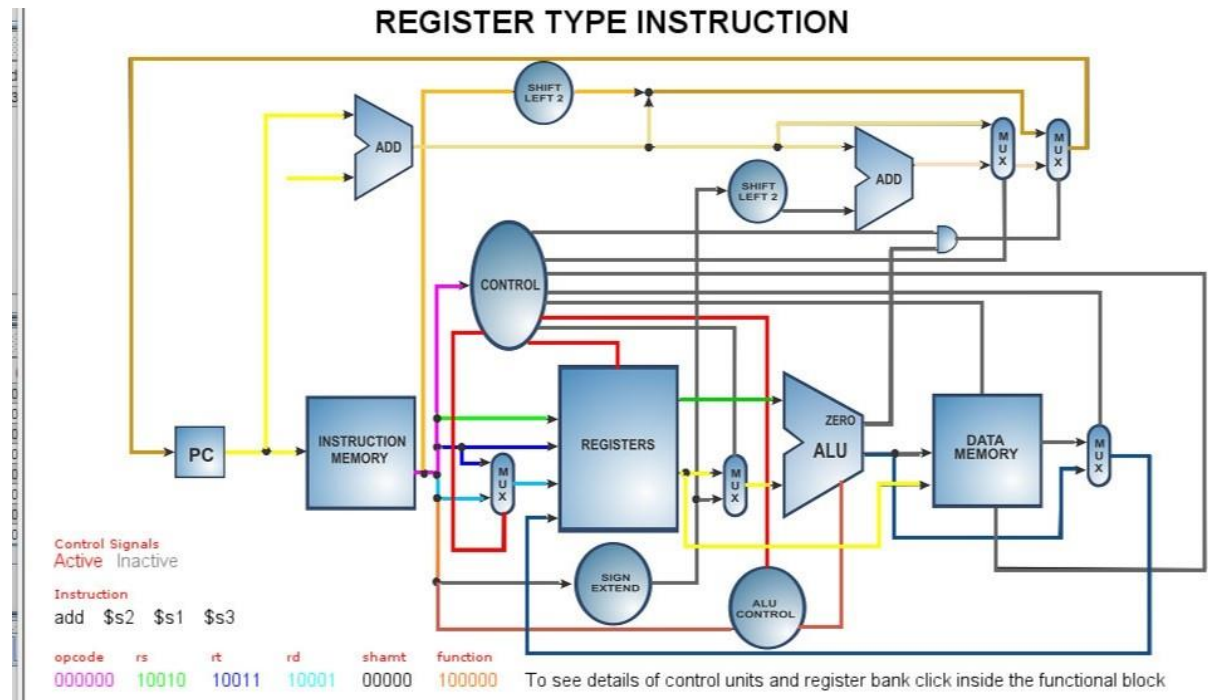


CHƯƠNG 4. The Processor

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

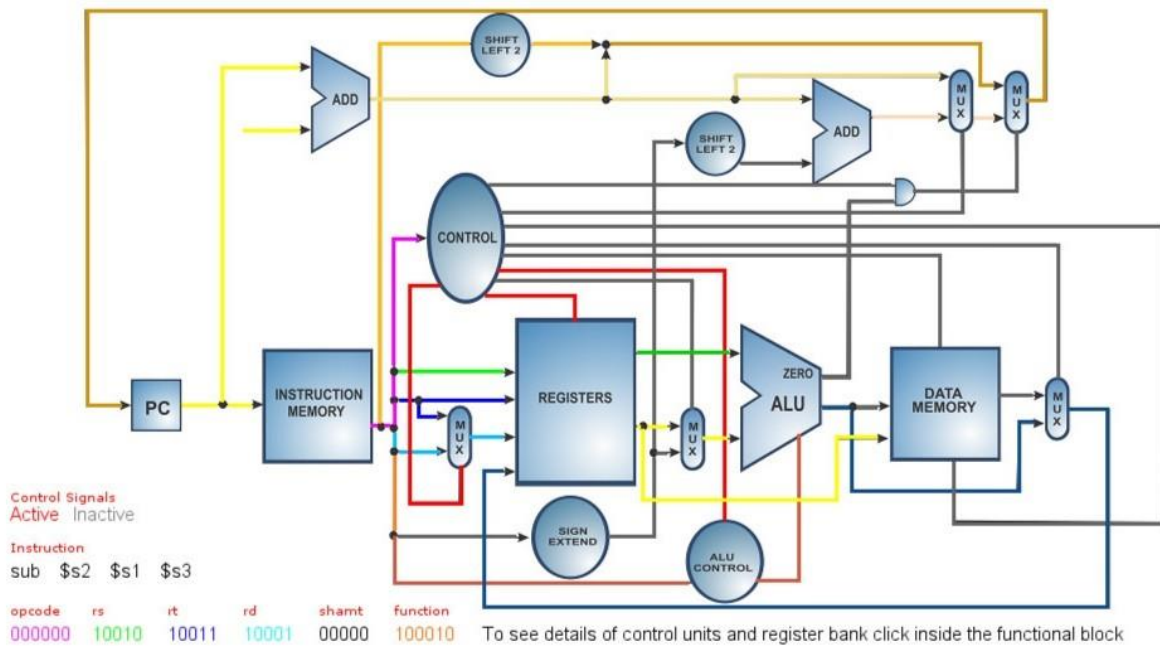
Add \$s1, \$s2, \$s3 (\$s1 = \$s2 + \$s3)



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

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

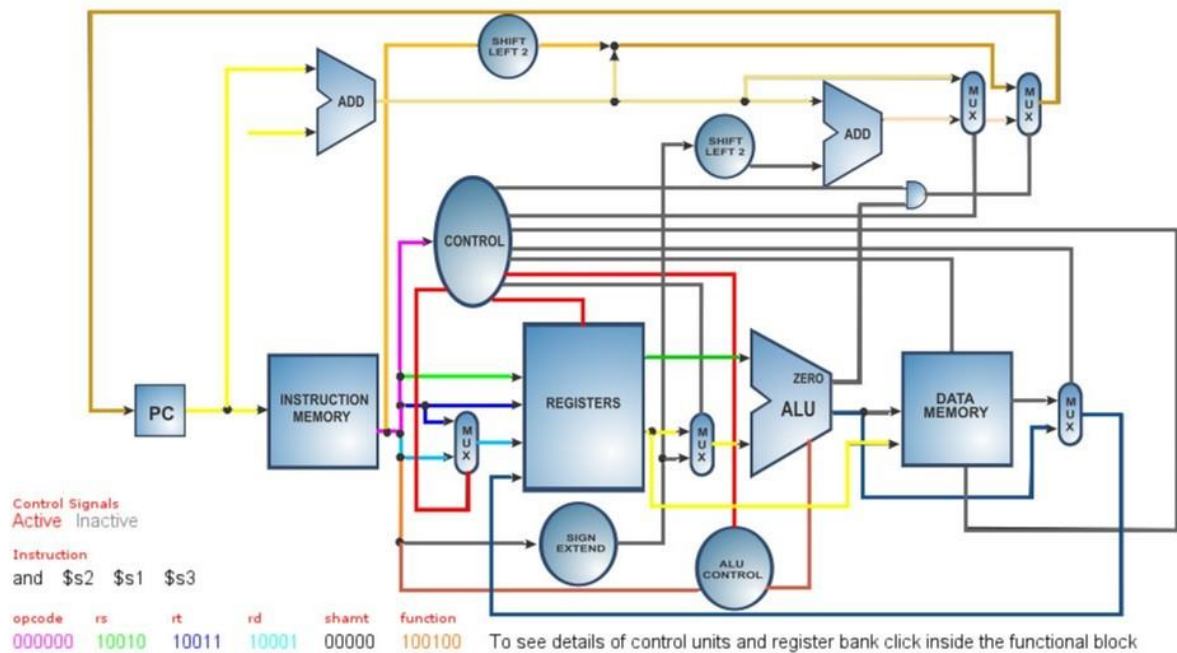
REGISTER TYPE INSTRUCTION



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

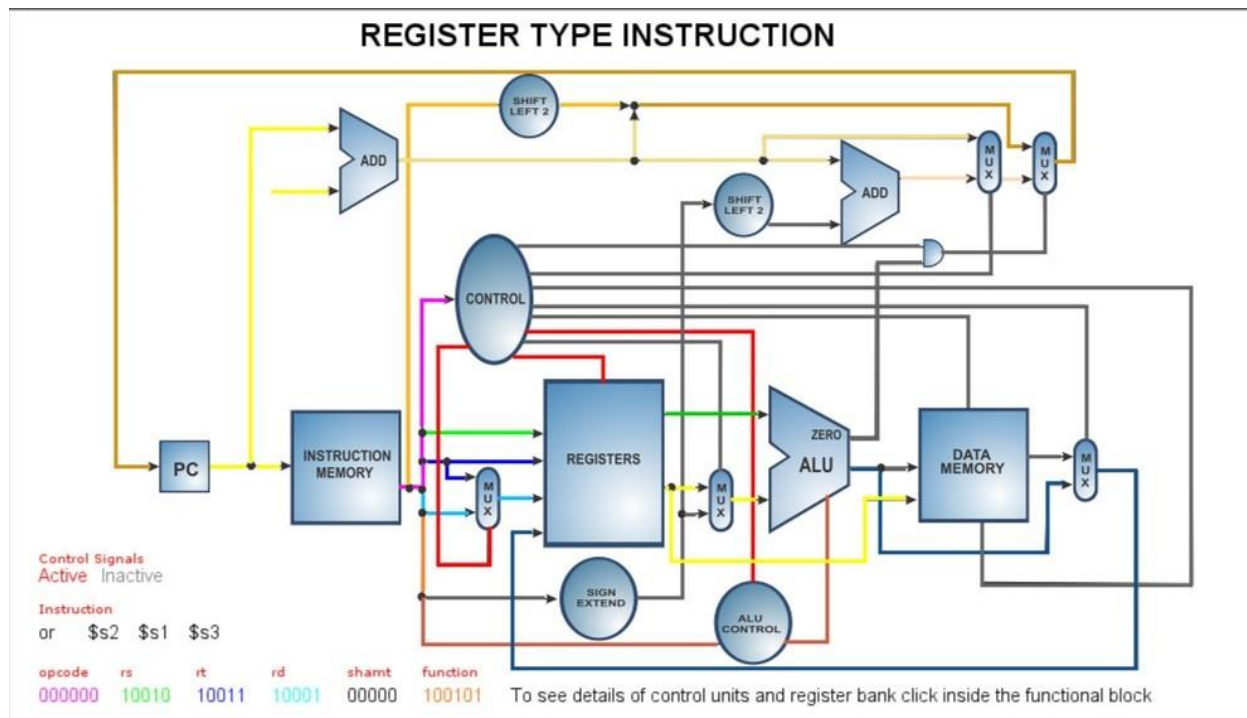
And \$s1, \$s2, \$s3 (\$s1 = \$s2 and \$s3)

REGISTER TYPE INSTRUCTION



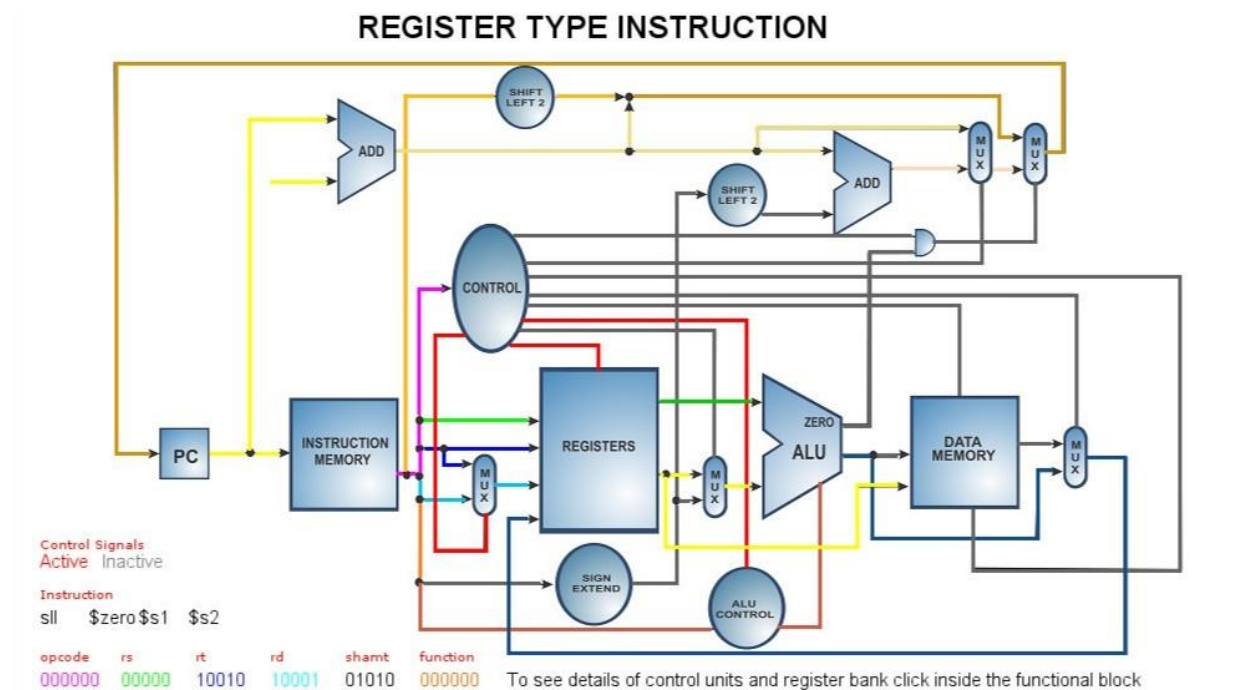
Câu 4. Vẽ sơ đồ datapath của lệnh Or rd, rt, rs

Or \$s1, \$s2, \$s3 (\$s1 = \$s2 || \$s3)



Câu 5. Vẽ sơ đồ datapath của lệnh Sll rt, rs, constant

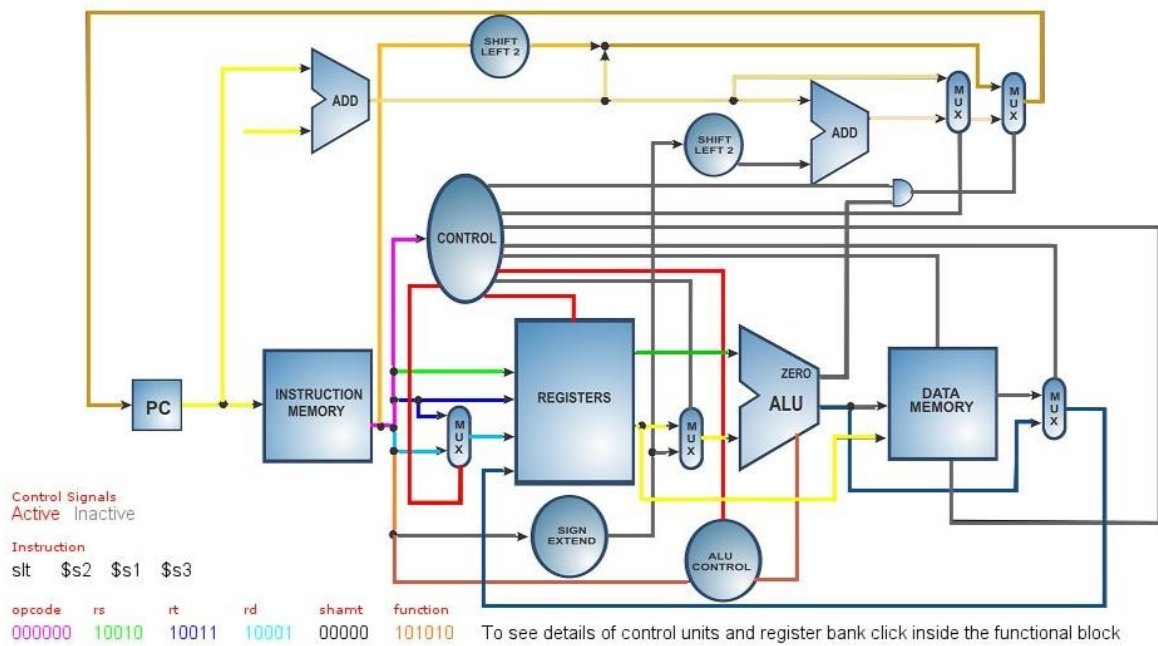
And \$s1, \$s2, 10 (\$s1 = \$s2 <<10)



Câu 5. Vẽ sơ đồ datapath của lệnh Slt rd, rs, rt

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

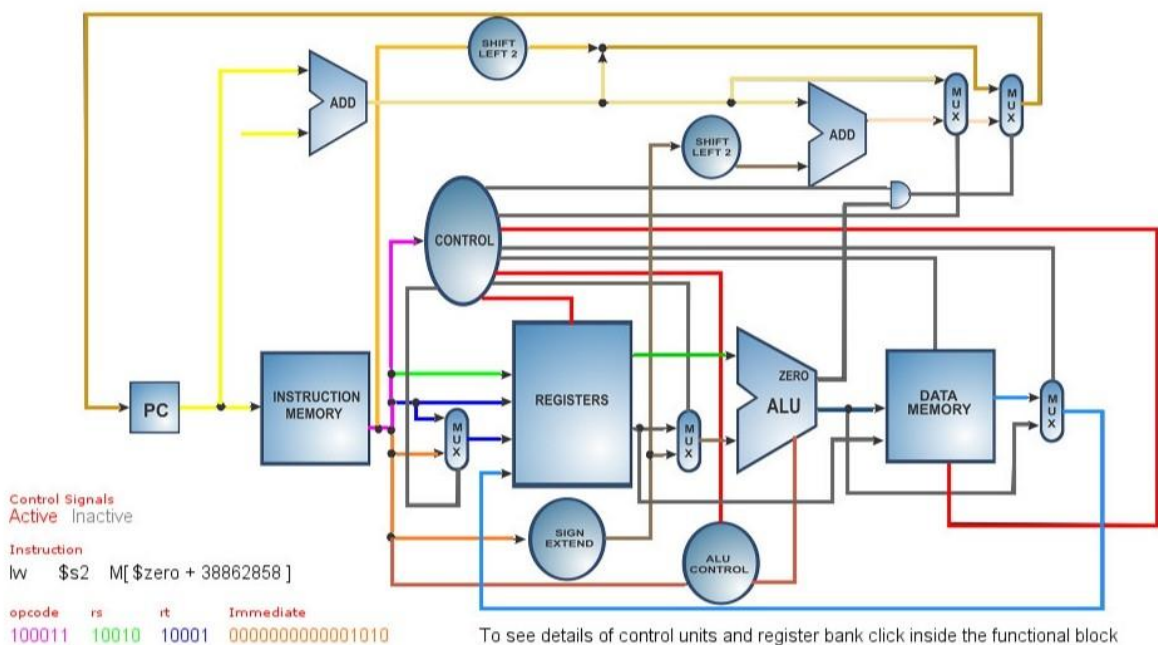
REGISTER TYPE INSTRUCTION



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

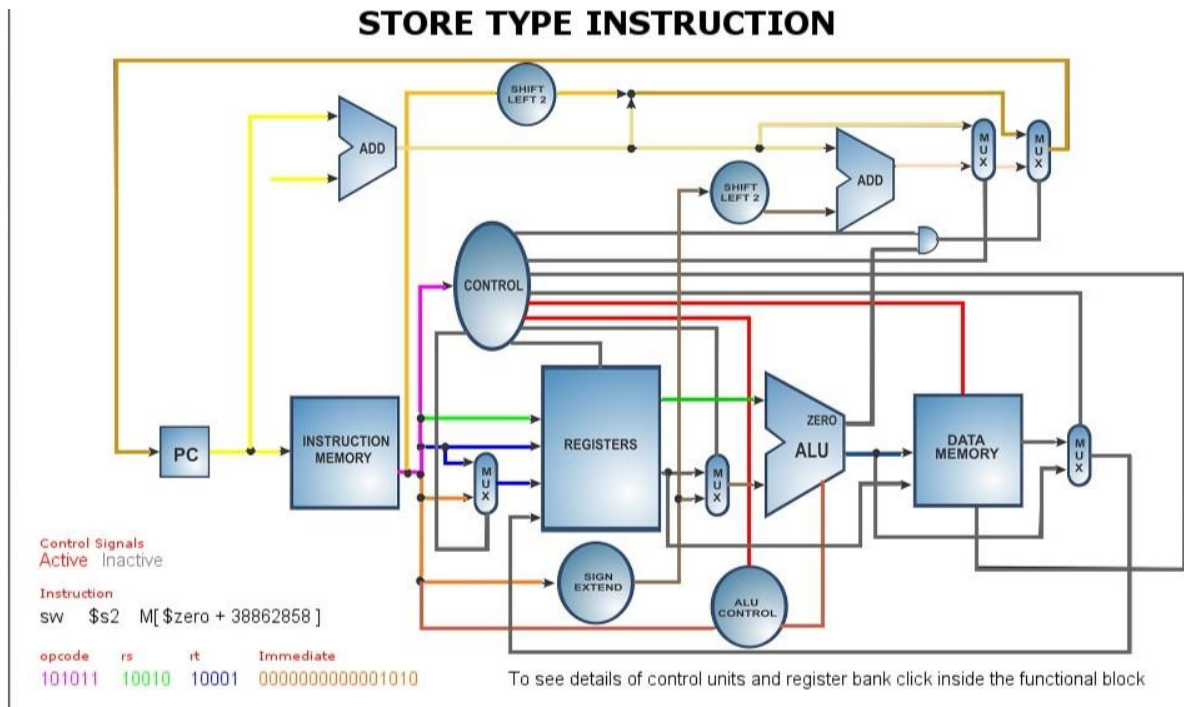
Lw \$s1, 10(\$s2)

LOAD TYPE INSTRUCTION

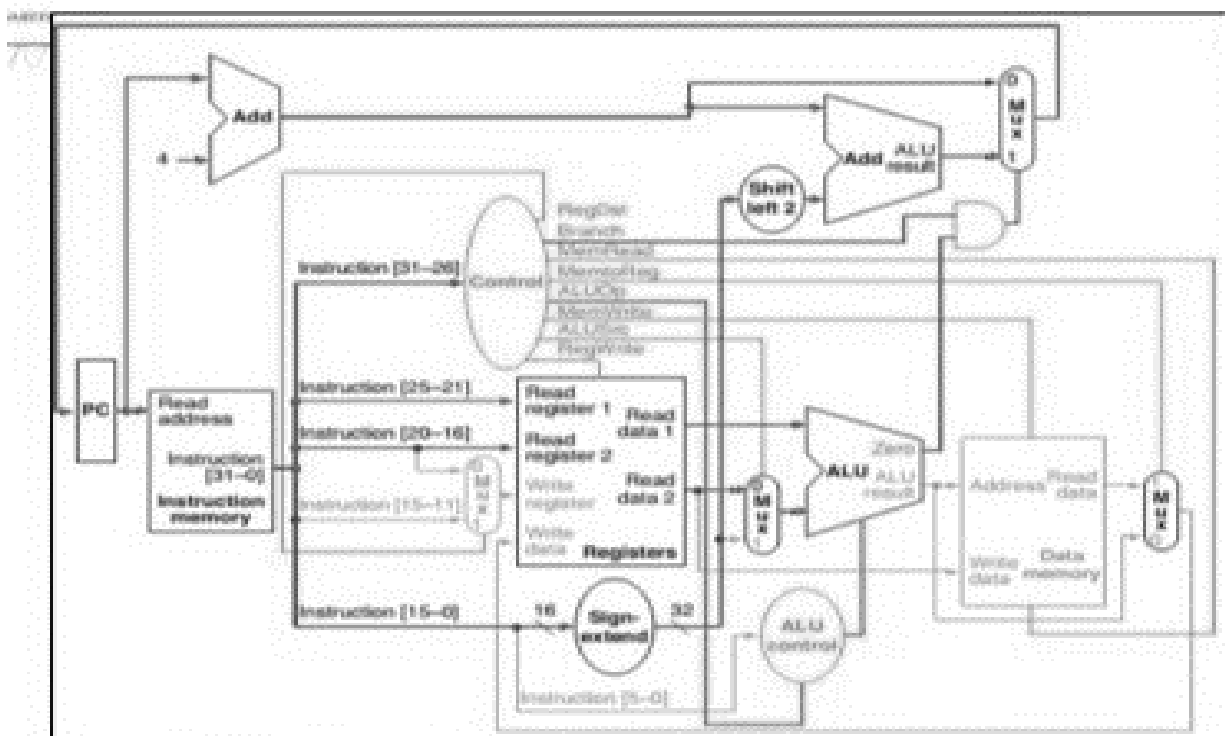


Câu 7. Vẽ sơ đồ datapath của lệnh Sw rt, constant(rd)

Sw \$s1, 10(\$s2)



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 beq 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	$\frac{1}{400L.CPI}$	$\frac{1}{399L.CPI}$
Chi phí	3840	4440
$\frac{\text{Chi phí}}{\text{Hiệu suất}}$ (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]

BEQ						IF	ID	EX	MEM	WB										
LW											IF	ID	EX	MEM	WB					
SW																IF	ID	EX	MEM	WB

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

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:

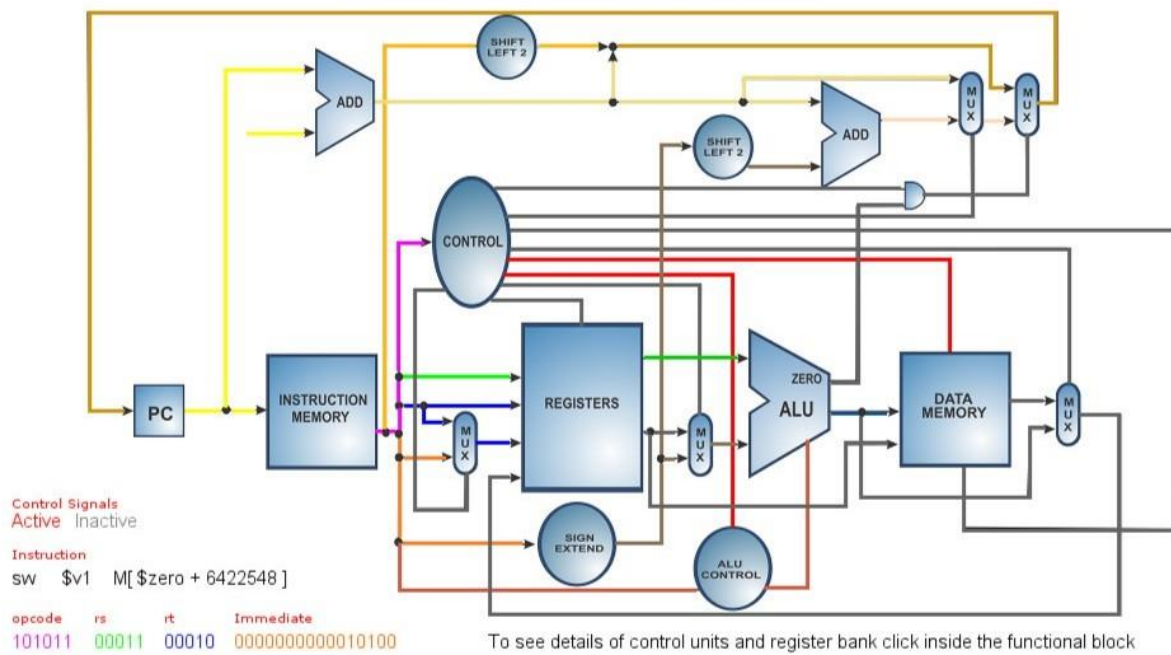
Rs : -3

Constant : 20

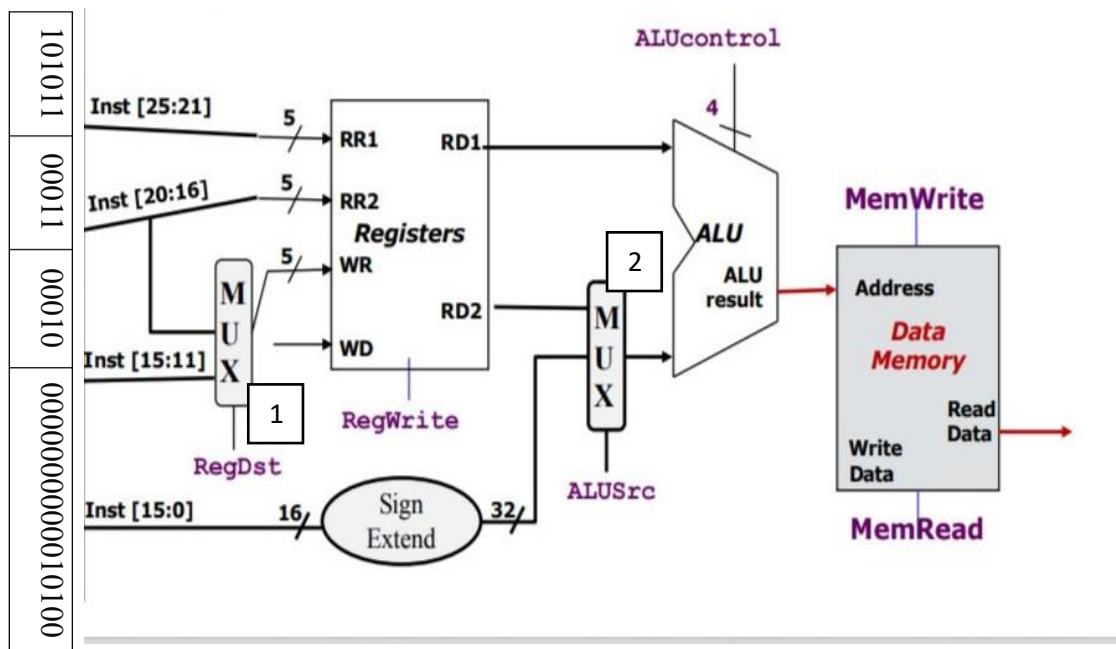
- Địa chỉ PC mới sau khi lệnh này được thực thi là gì? Đánh dấu đường dẫn mà giá trị này được xác định.

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