



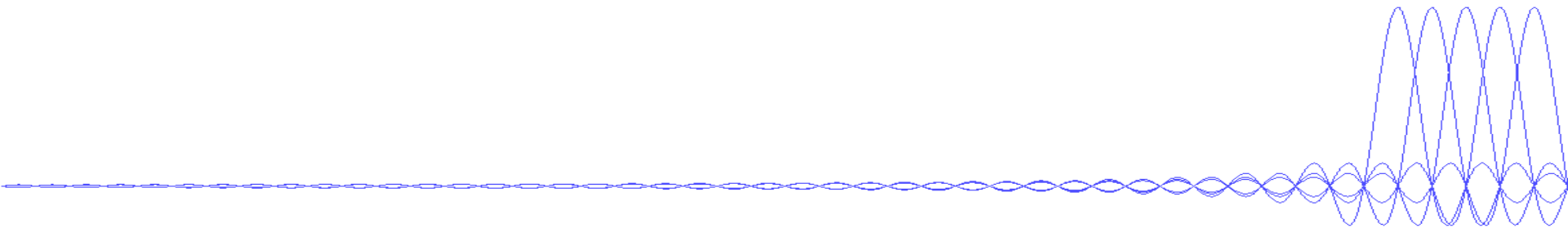
COMPUTER ENGINEERING



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

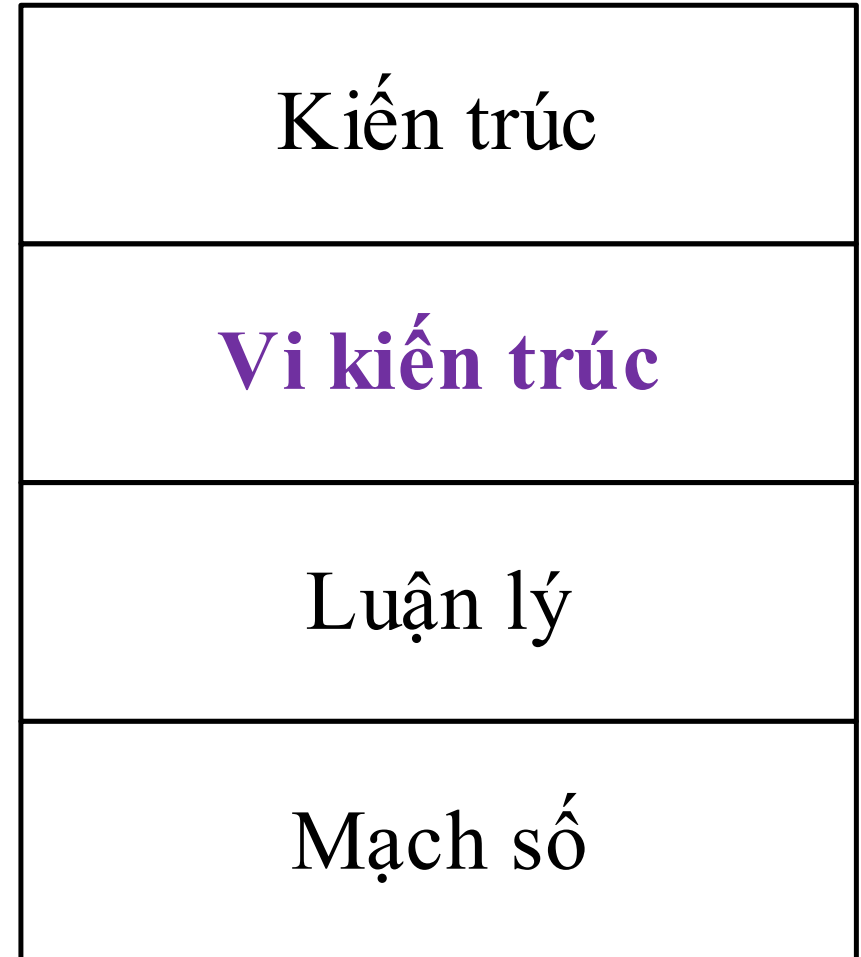
TỔ CHỨC VÀ CẤU TRÚC MÁY TÍNH II

Chương 8 Bộ xử lý





- Vi kiến trúc
- Datapath
- Thực thi lệnh
- Bài tập





Vi kiến trúc (1/2)

■ Kiến trúc Máy tính bao gồm 3 thành phần chính:

□ Kiến trúc tập lệnh (ISA): Quy định máy tính có thể làm những việc gì?

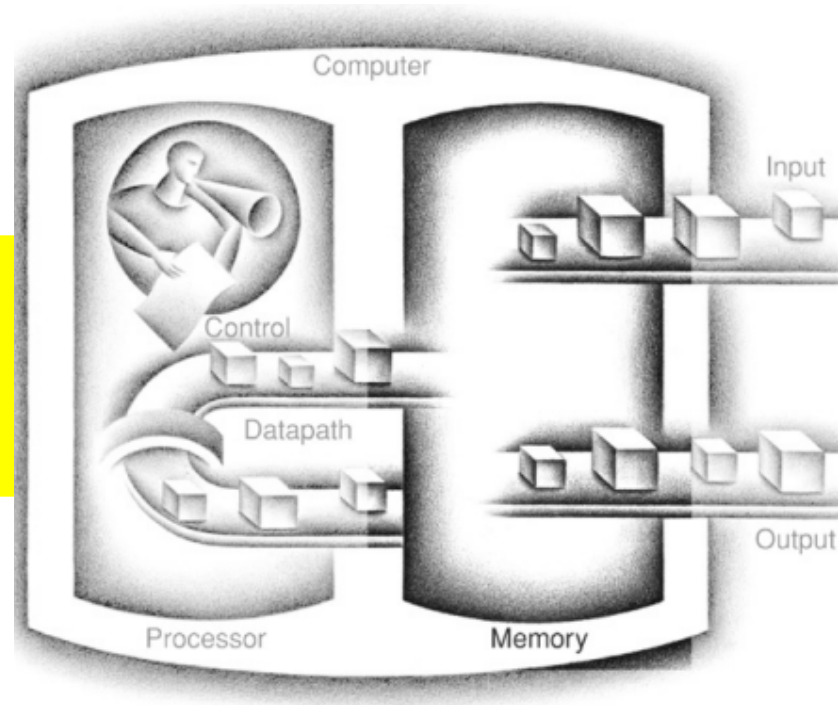
■ Lệnh

□ Vi kiến trúc (Tổ chức Phần cứng Máy tính): Quy định máy tính làm việc như thế nào?

■ Hiện thực ISA

□ Hệ thống Máy tính: Quy định các thành phần của máy tính phối hợp trong một hệ thống điện toán như thế nào?

■ Ảo hóa, Quản lý Bộ nhớ, Xử lý Đồ họa...



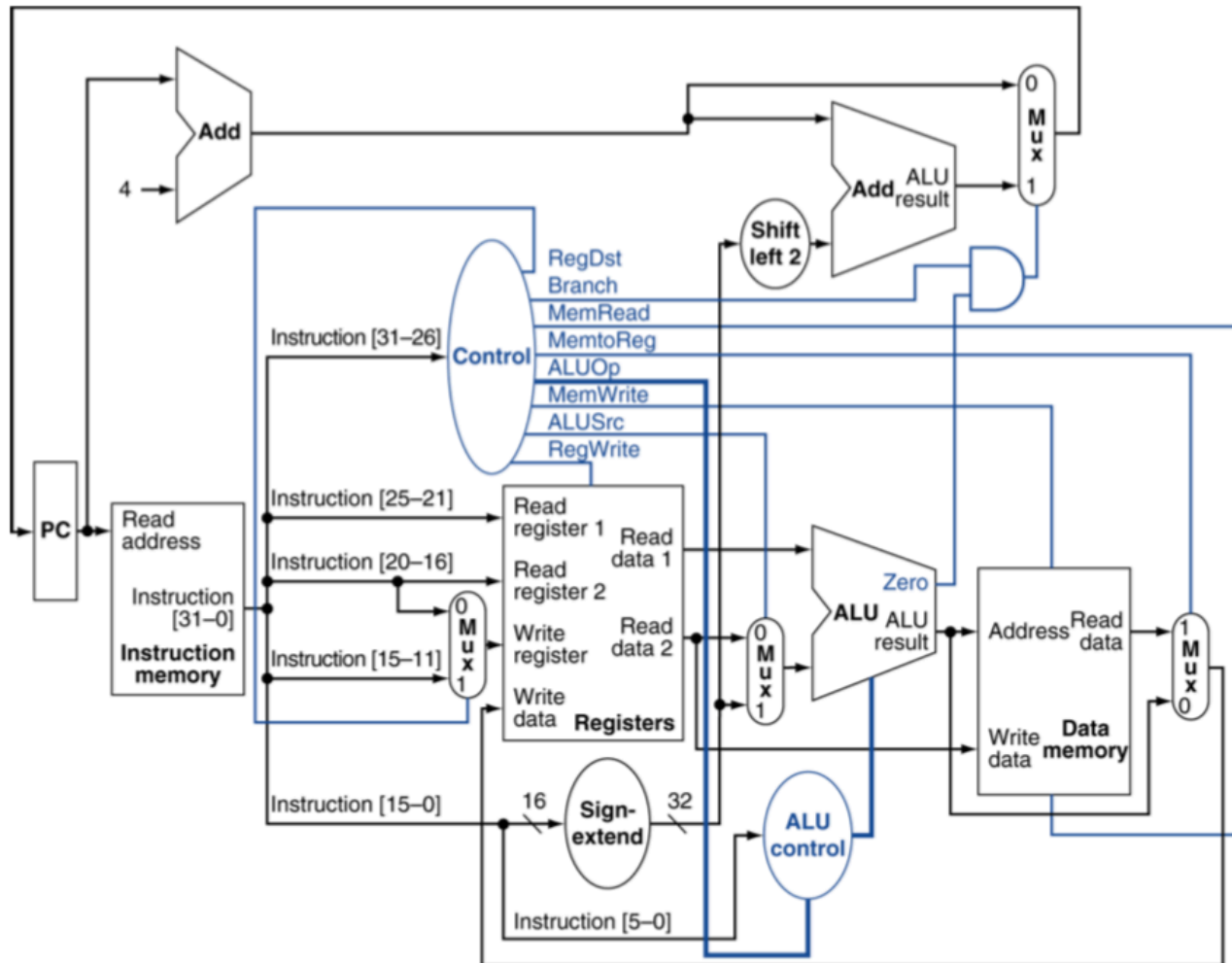


Vi kiến trúc (2/2)

- Về chức năng, Vi kiến trúc là một tổ chức phần cứng dùng để hiện thực tập lệnh của một máy tính.
- Về cấu tạo, Vi kiến trúc được chia thành 2 khối:
 - Khối đường dữ liệu (datapath): Thực thi lệnh
 - Lưu trữ: Bộ nhớ lệnh, Bộ nhớ dữ liệu, Tập thanh ghi, ...
 - Truyền/nhận: Các đường tín hiệu dữ liệu, địa chỉ, điều khiển
 - Xử lý: ALU, Bộ so sánh, Mux, Bộ mở rộng dấu, Bộ dịch, ...
 - Khối điều khiển (control unit): Điều khiển datapath hoạt động
 - Dựa trên opcode của lệnh và trạng thái của datapath



CPU MIPS: Datapath and Controller

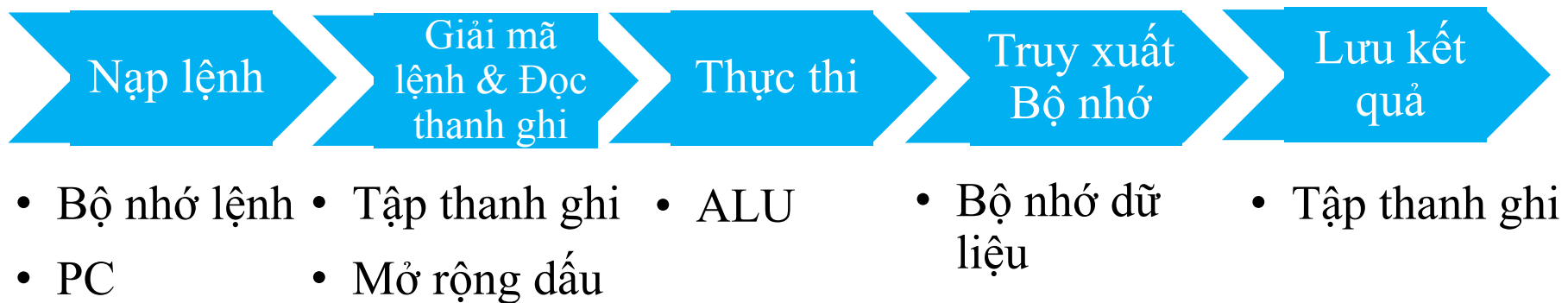




Datapath (1/9) – Chu kỳ thực thi lệnh

■ Datapath dùng để thực thi lệnh! Một lệnh thực thi như thế nào?

□ Chu kỳ thực thi lệnh!



Lệnh R (add, sub, and, or, slt): 1, 2, 3, 5

Lw: 1, 2, 3, 4, 5

Sw: 1, 2, 3, 4

Beq: 1, 2, 3



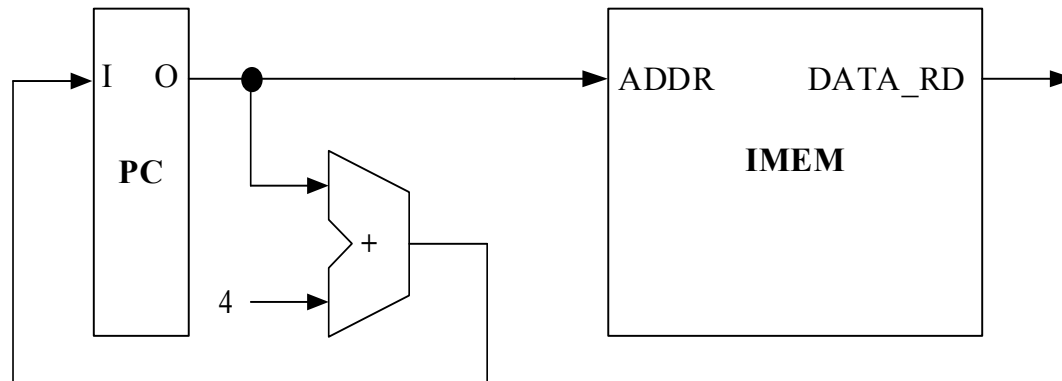
COMPUTER ENGINEERING





Datapath (1/5) – Nạp lệnh

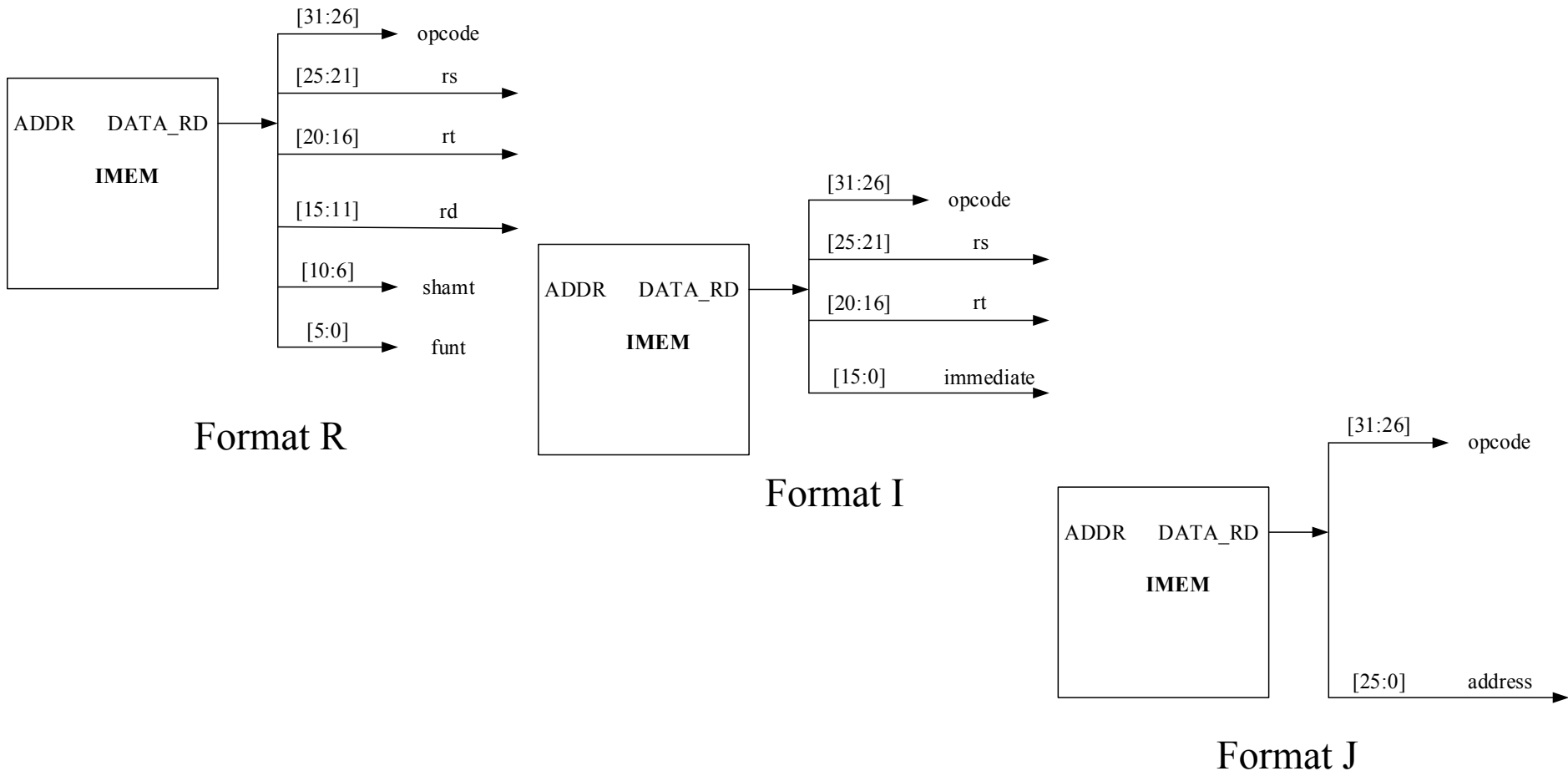
- Lệnh cần nạp lưu trong Bộ nhớ lệnh
- Địa chỉ của lệnh cần nạp lưu trong thanh ghi PC
 - Tăng PC lên 4 để chuẩn bị nạp lệnh tiếp theo -> cần thêm bộ cộng





Datapath (2/5) – Giải mã lệnh – Định dạng lệnh

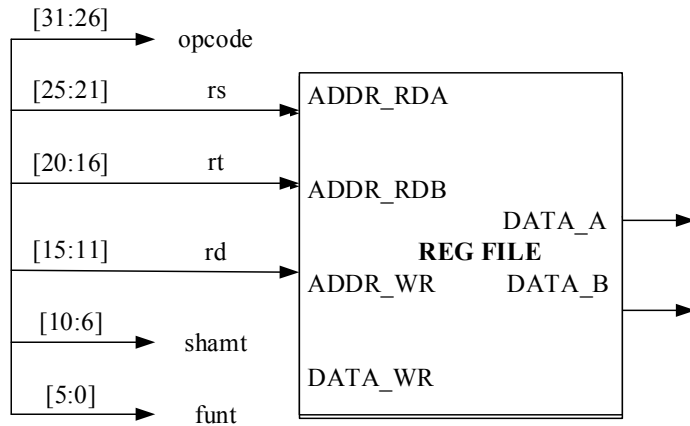
■ Dựa vào opcode để xác định định dạng lệnh



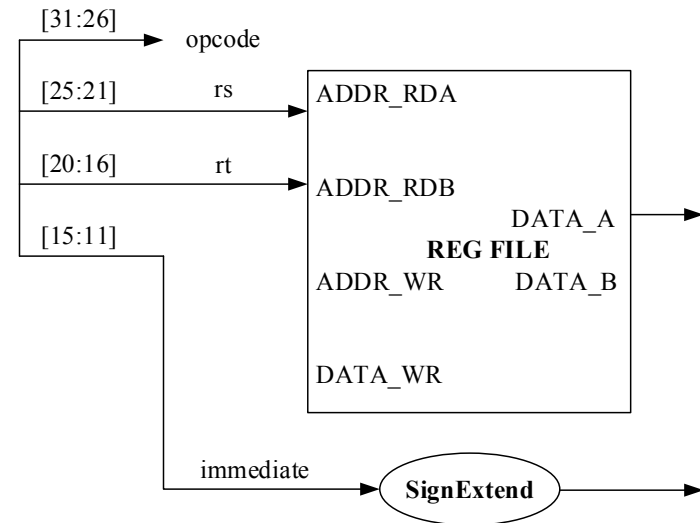


Datapath (2/5) – Giải mã lệnh - Nạp toán hạng

- Dựa vào định dạng lệnh mà nạp toán hạng tương ứng



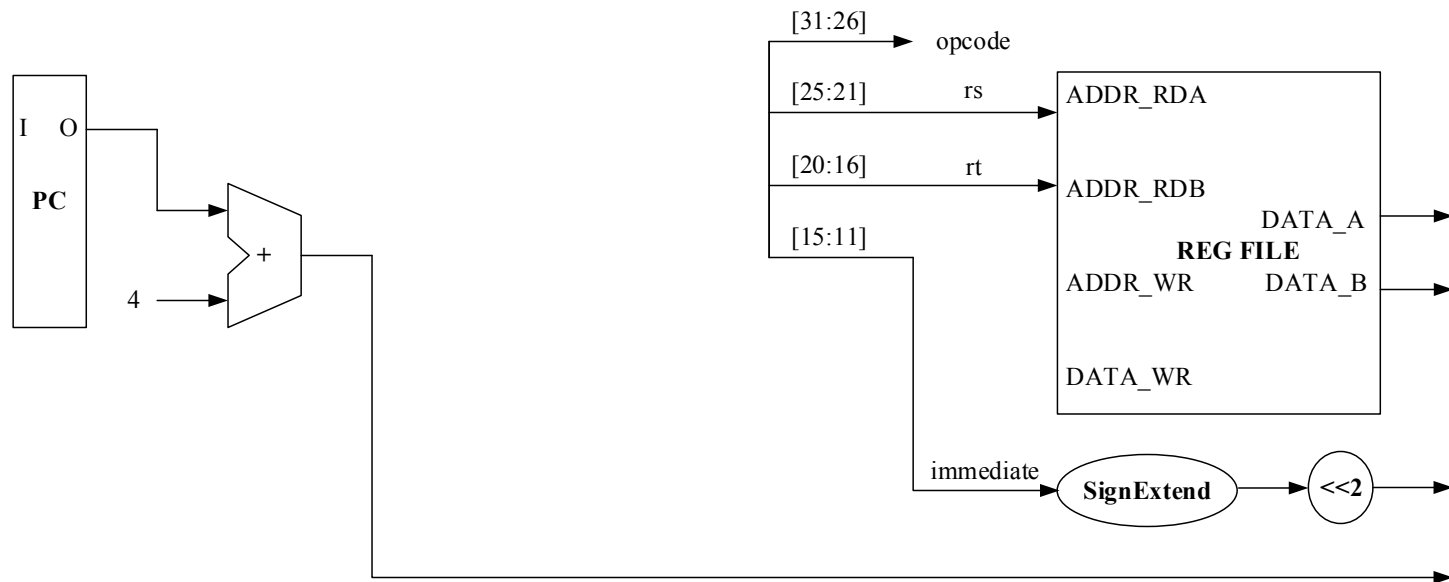
Lệnh R



Lệnh lw, sw



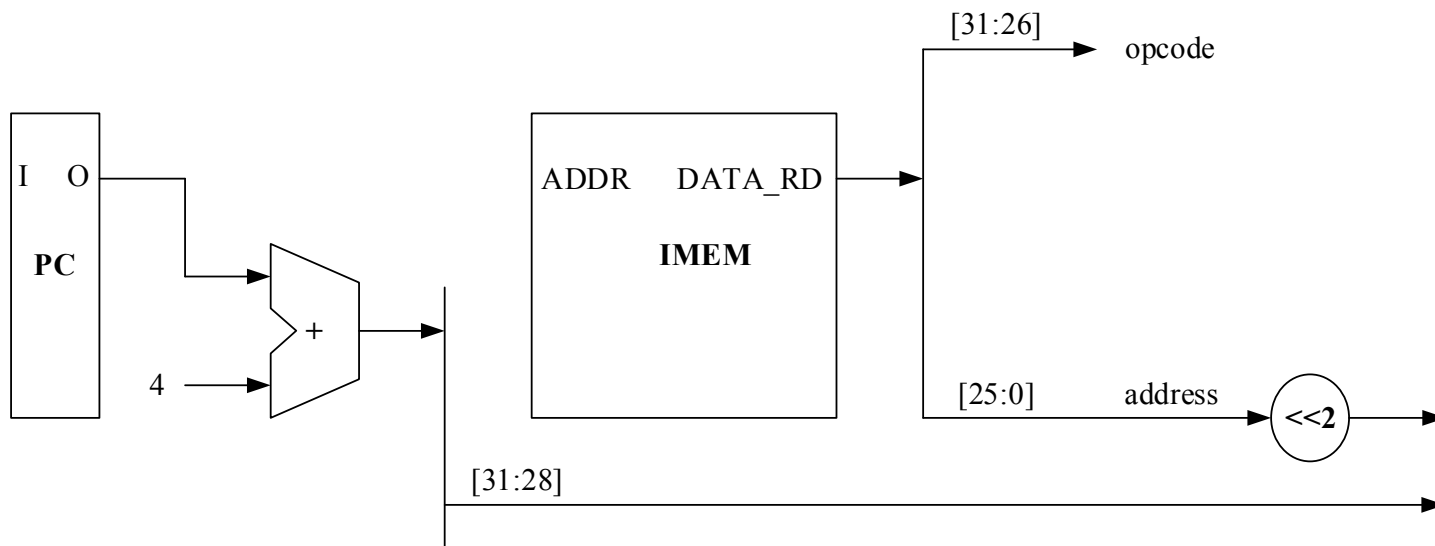
Datapath (2/5) – Giải mã lệnh - Nạp toán hạng



Lệnh `beq/bne`



Datapath (2/5) – Giải mã lệnh - Nạp toán hạng

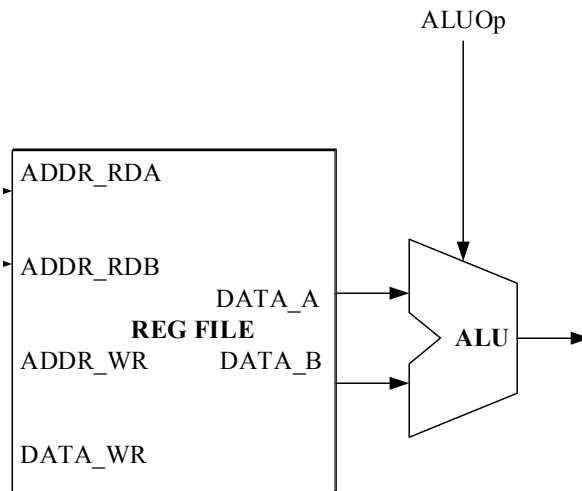


Lệnh Jump

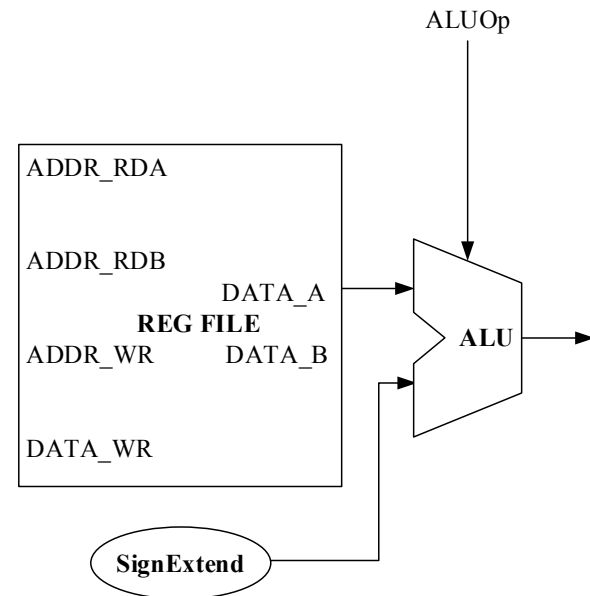


Datapath (3/5) – Thực thi

- Dựa vào opcode và funct để quyết định thao tác gì sẽ được thực thi



Lệnh R, beq/bne

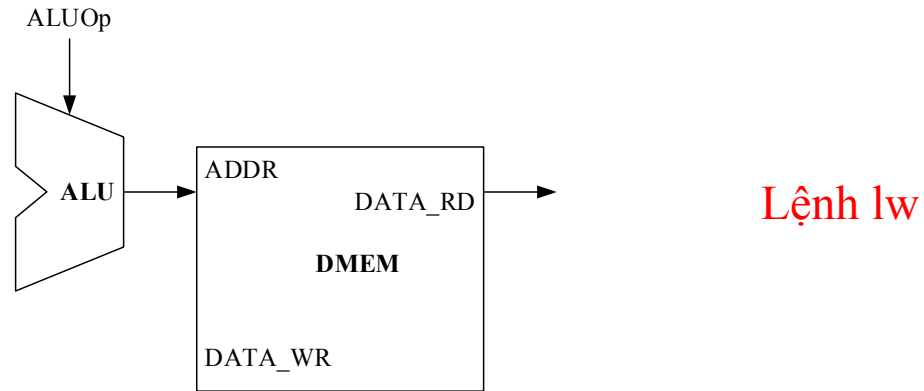


Lệnh lw, sw

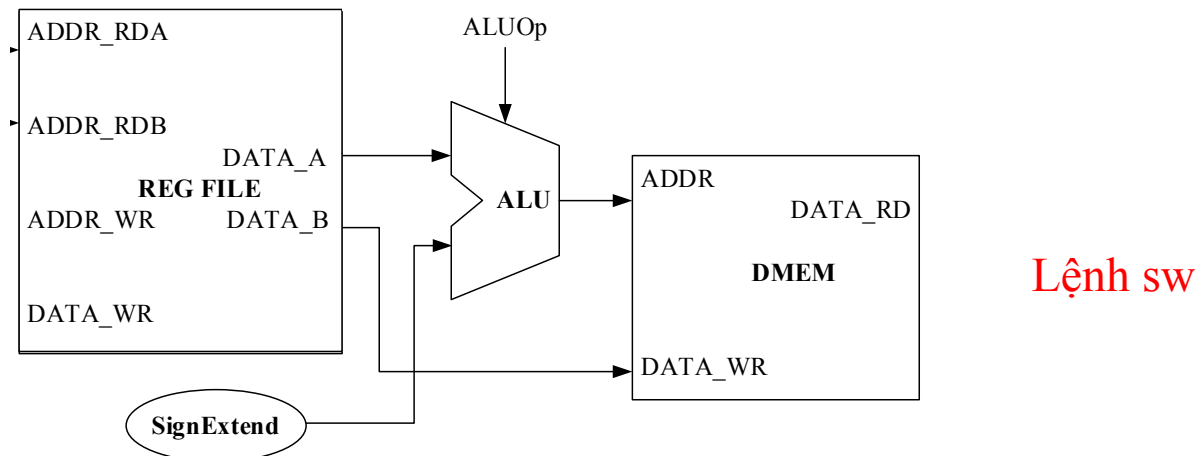


Datapath (4/5) – Truy cập bộ nhớ

- Đọc dữ liệu từ bộ nhớ dữ liệu đối với các lệnh nạp



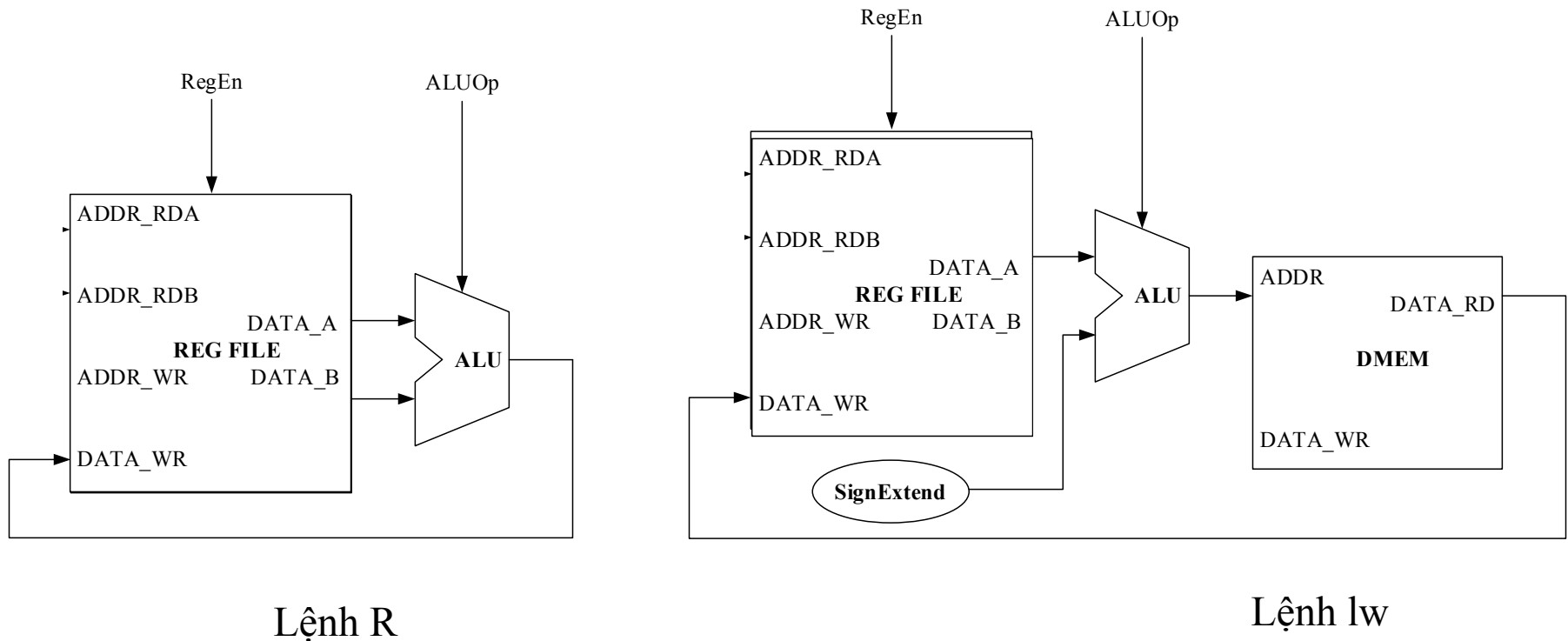
- Ghi dữ liệu tới bộ nhớ dữ liệu đối với các lệnh lưu





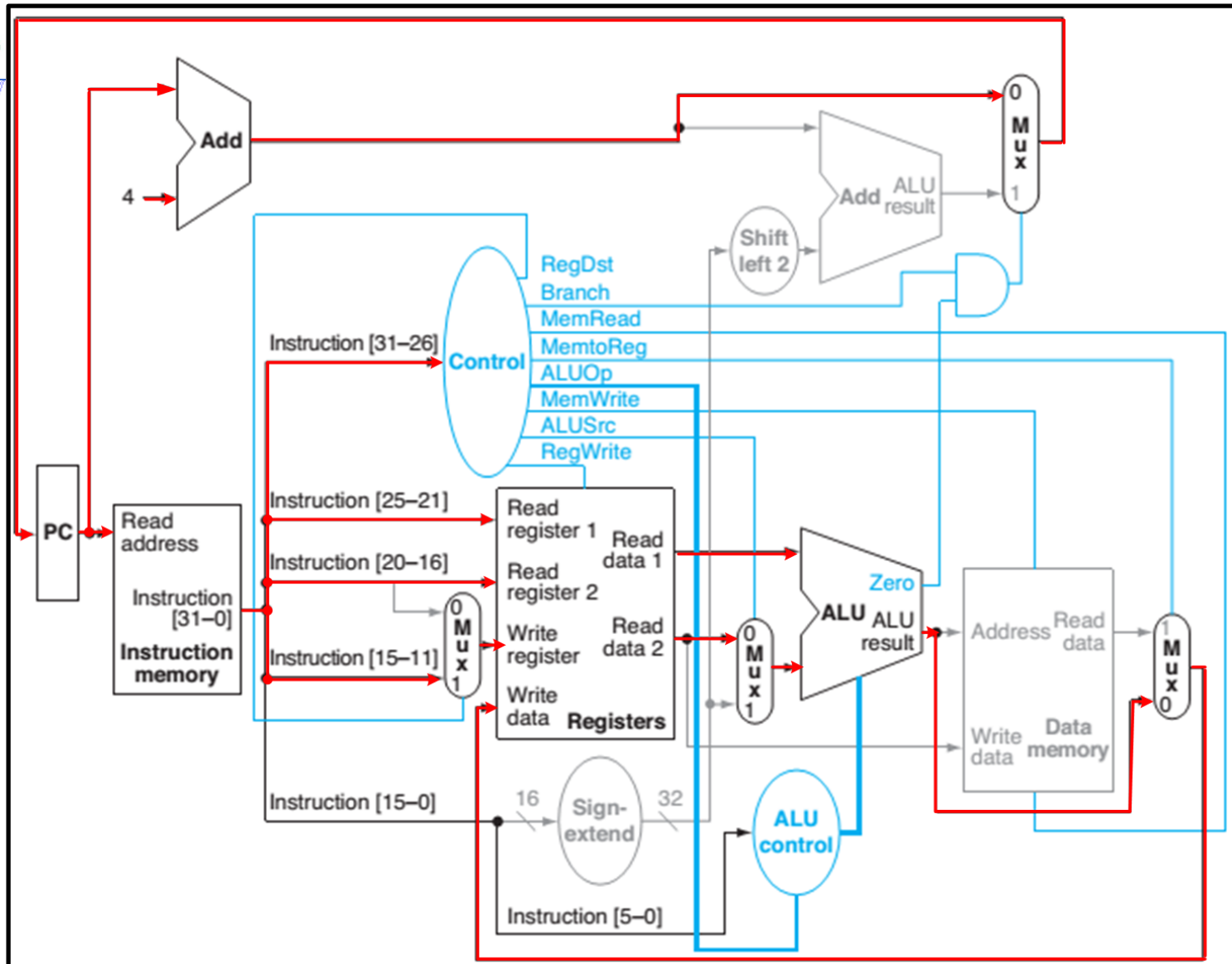
Datapath (5/5) – Lưu kết quả

- Có thể ghi dữ liệu về lại Tập thanh ghi





Xem lại Datapath với từng nhóm lệnh

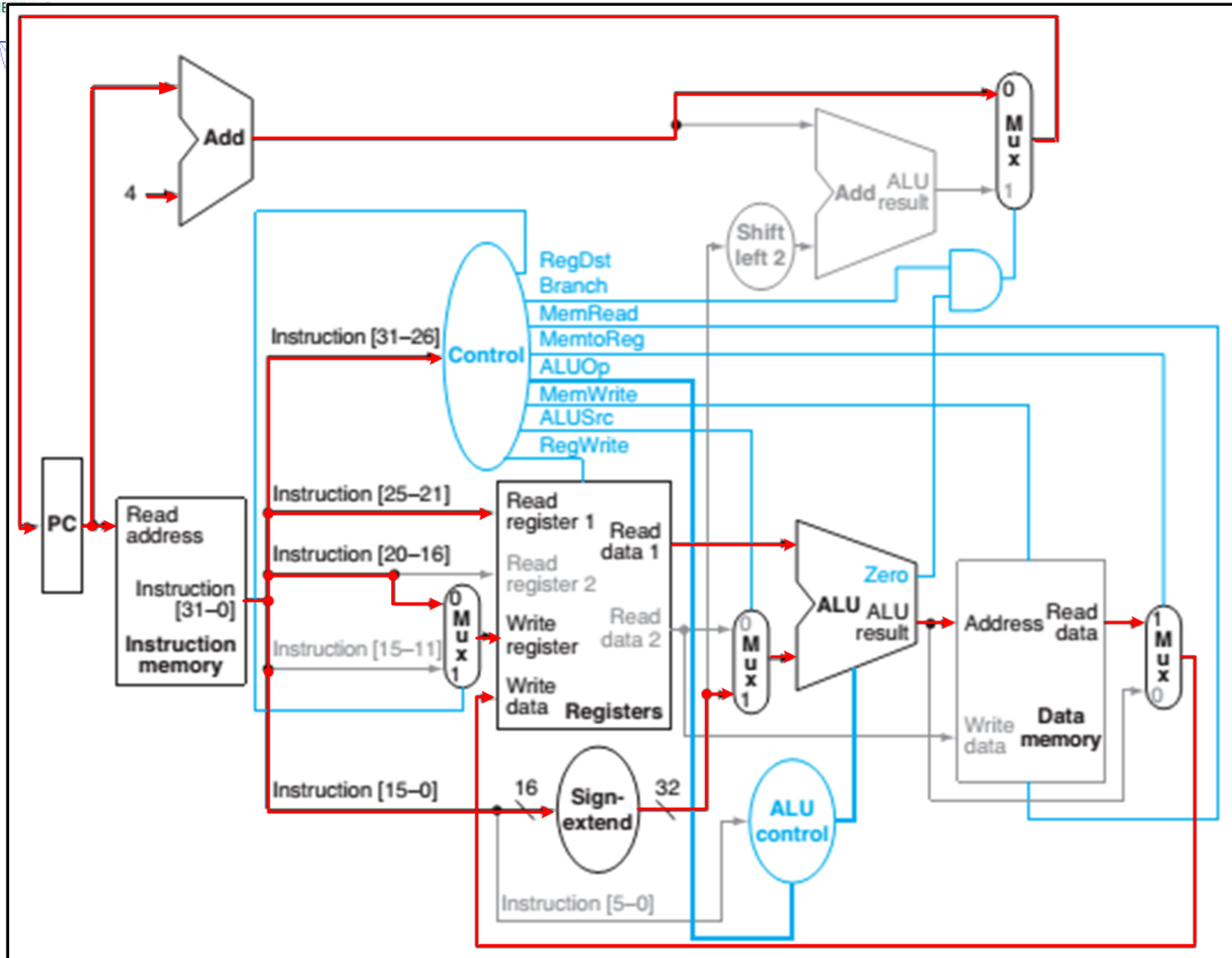


Các đường đỏ là các đường hoạt động khi lệnh thuộc **nhóm logic và số học R** (add, sub, and, or, slt) thực thi



Xem lại Datapath với từng nhóm lệnh

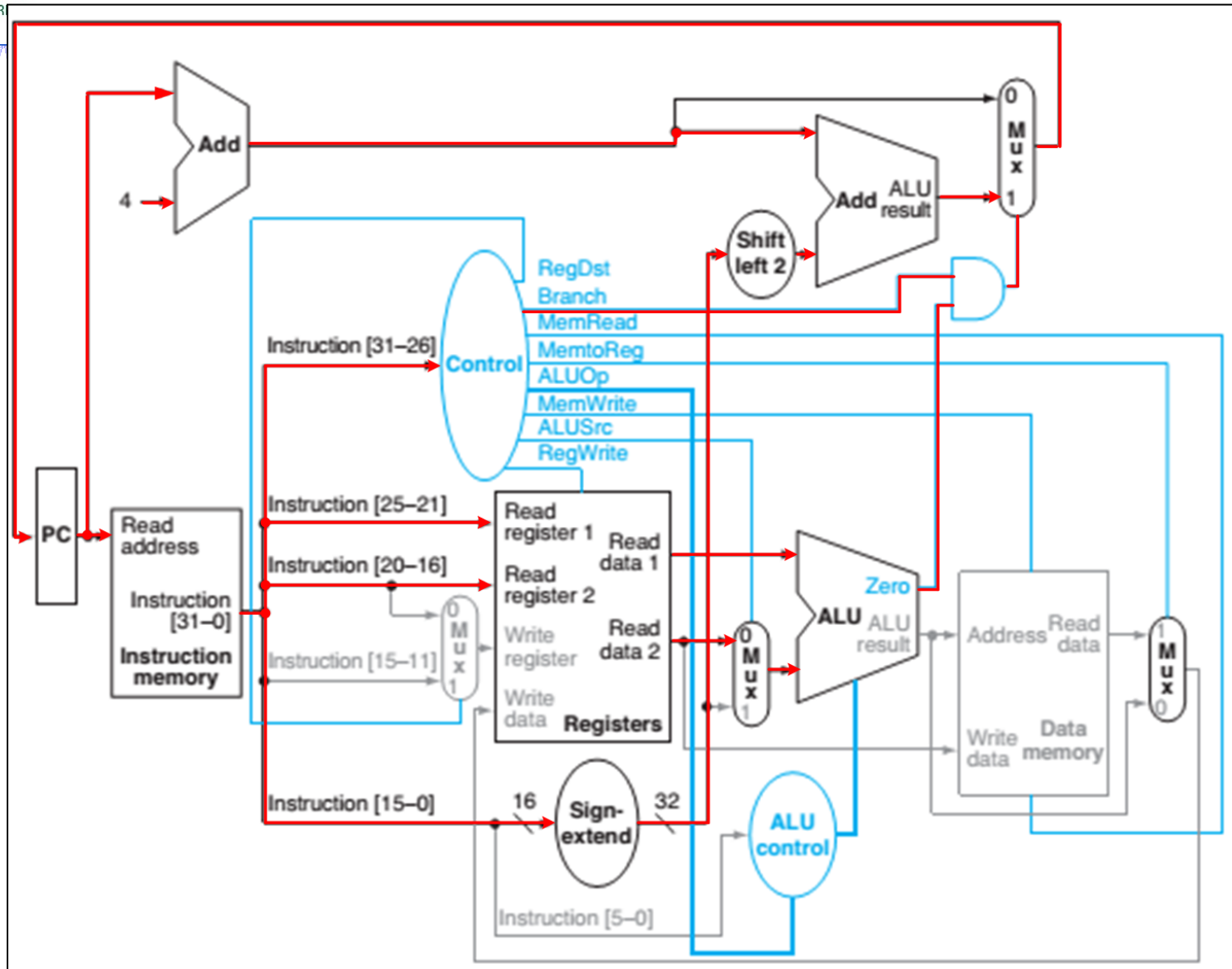
COMPUTER ENGINE





Xem lại Datapath với từng nhóm lệnh

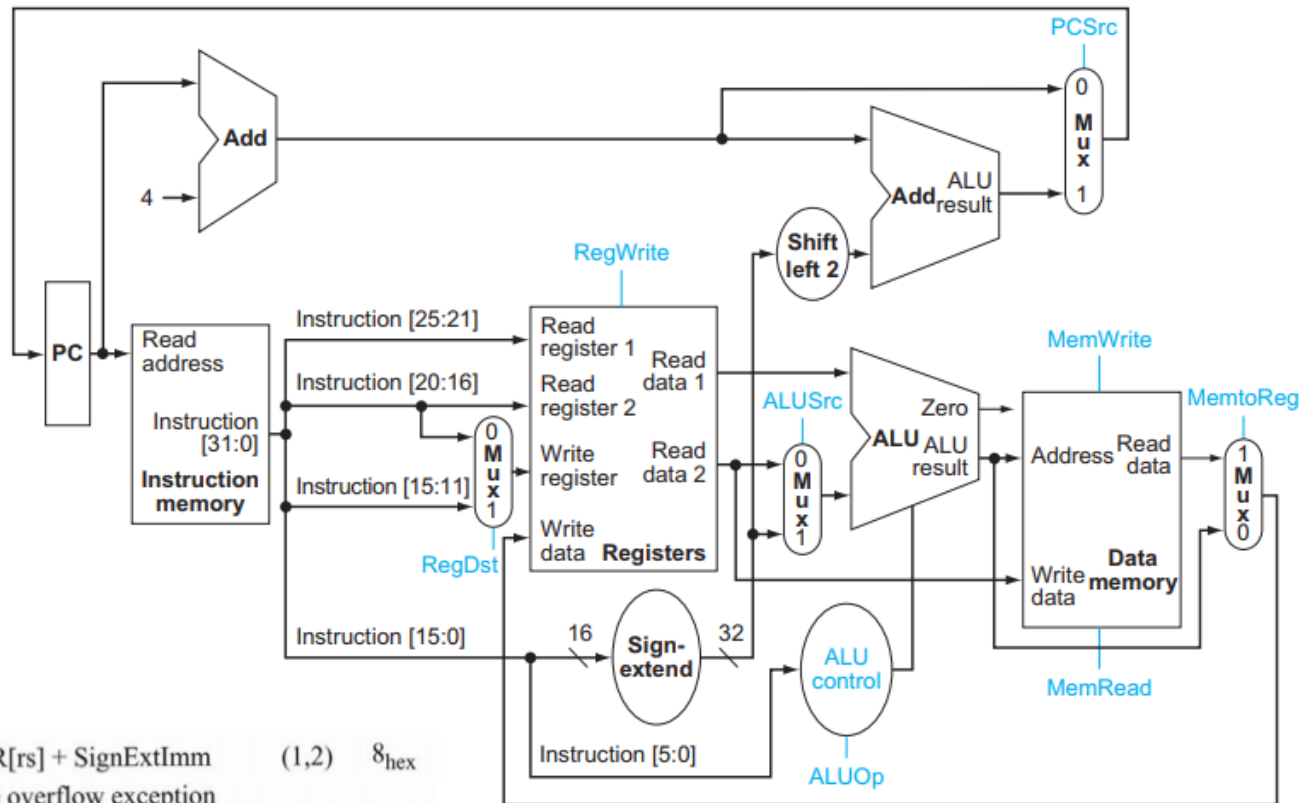
COMPUTER ENGINEER





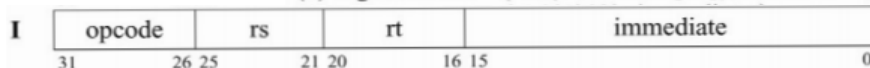
Bài tập (1/5)

Trình bày các khối chức năng được sử dụng khi thực thi lệnh addi?



Add Immediate `addi` I $R[rt] = R[rs] + \text{SignExtImm}$ (1,2) 8_{hex}

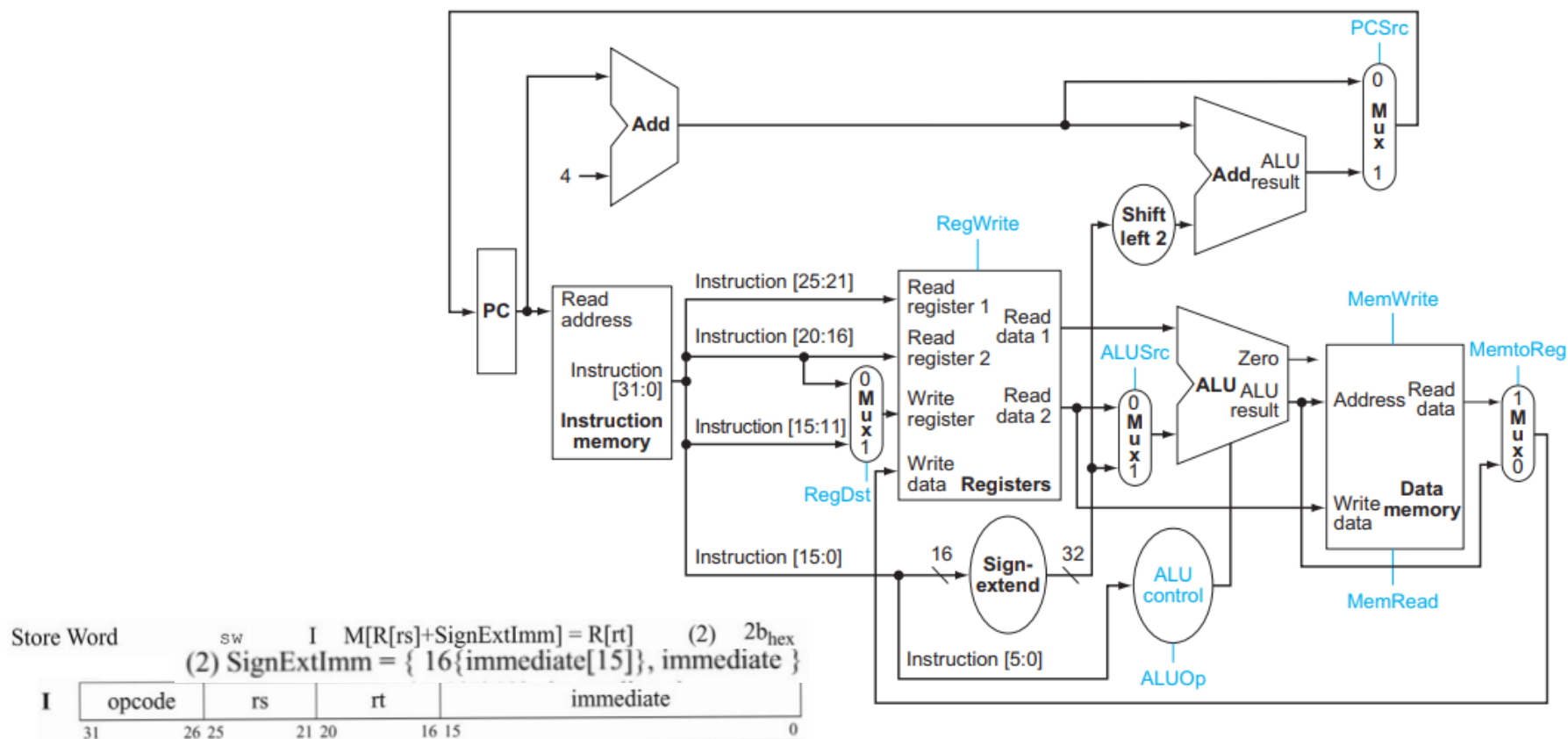
(1) May cause overflow exception
(2) $\text{SignExtImm} = \{ 16\{\text{immediate}[15]\}, \text{immediate} \}$





Bài tập (2/5)

Trình bày các khối chức năng được sử dụng khi thực thi lệnh sw?





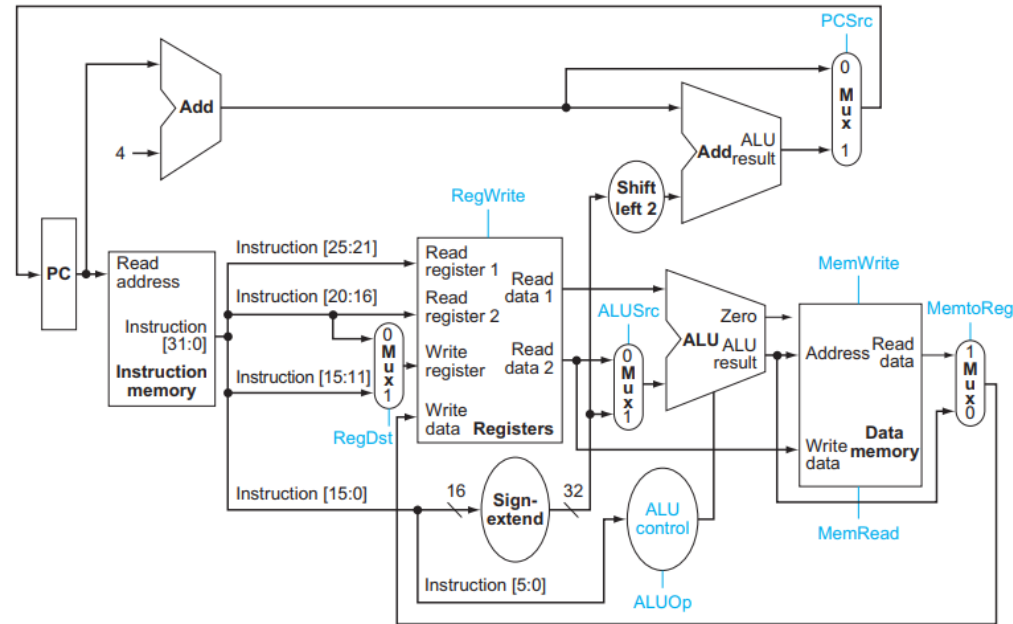
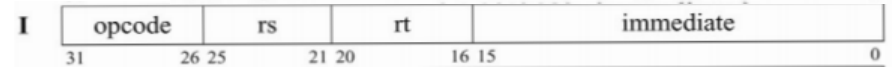
Bài tập (3/5)

Tìm Critical path của CPU khi thực thi lệnh lw?

Load Word

$1w \quad I \quad R[rt] = M[R[rs] + \text{SignExtImm}] \quad (2) \quad 23_{\text{hex}}$

(2) $\text{SignExtImm} = \{ 16\{\text{immediate}[15]\}, \text{immediate} \}$

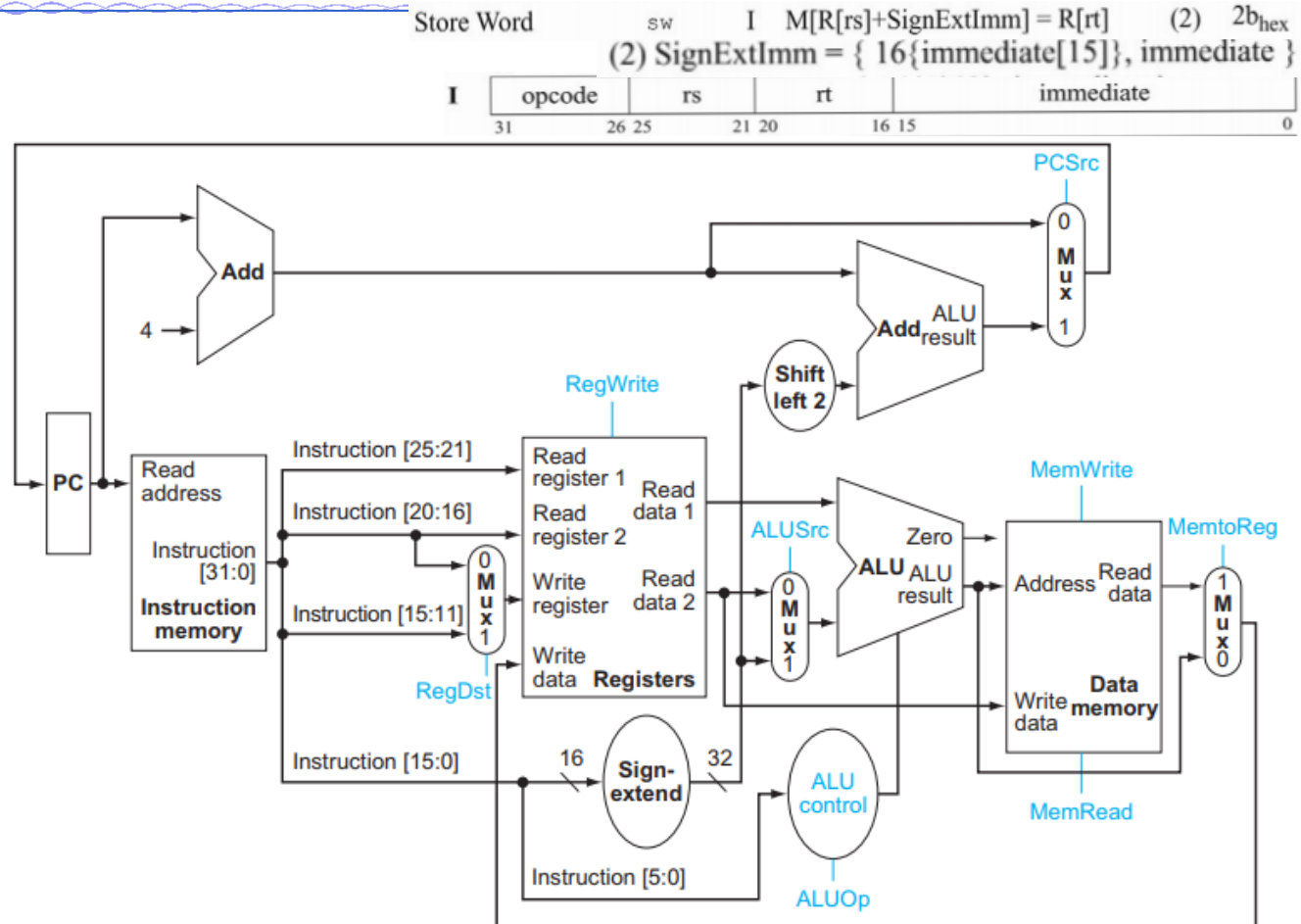


I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps



Bài tập (4/5)

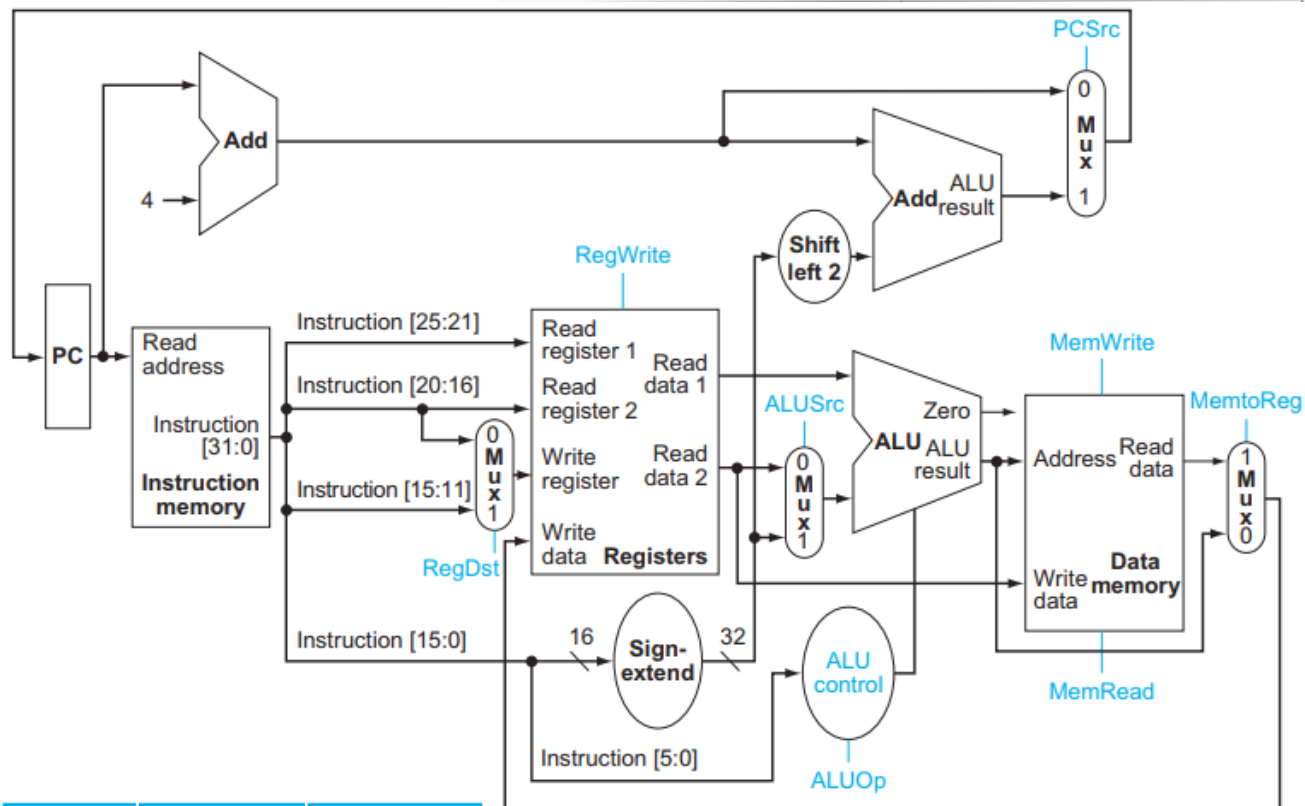
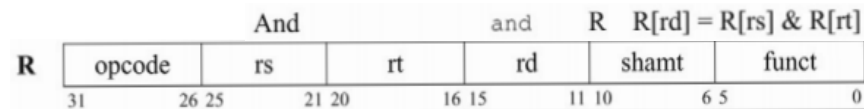
Tìm Critical path của CPU khi thực thi lệnh R?





Bài tập (5/5)

Tìm thời gian cần thiết tối thiểu để thực thi lệnh **and**?



I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps



■ Khối điều khiển

Kiến trúc

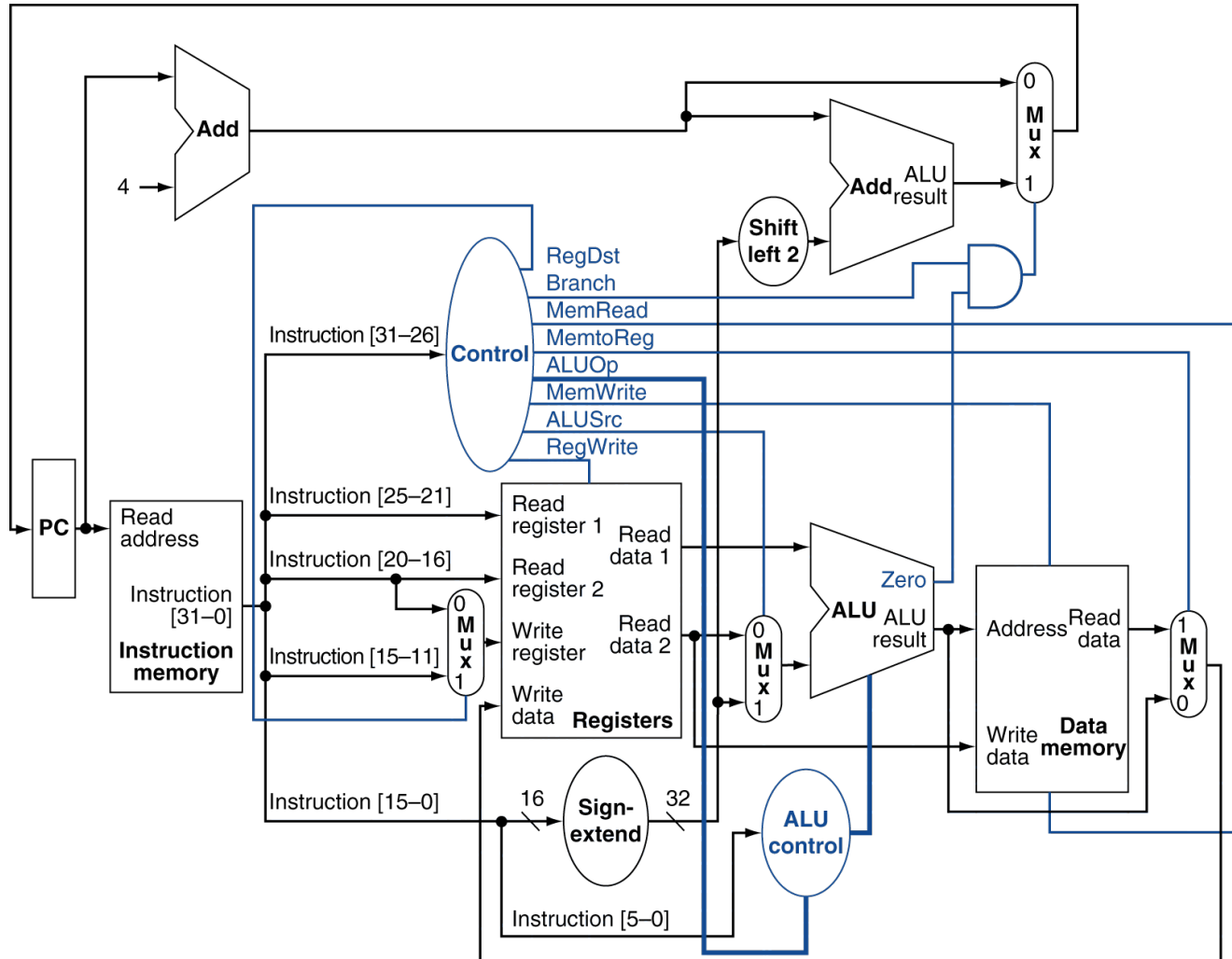
Vi kiến trúc

Luận lý

Mạch số

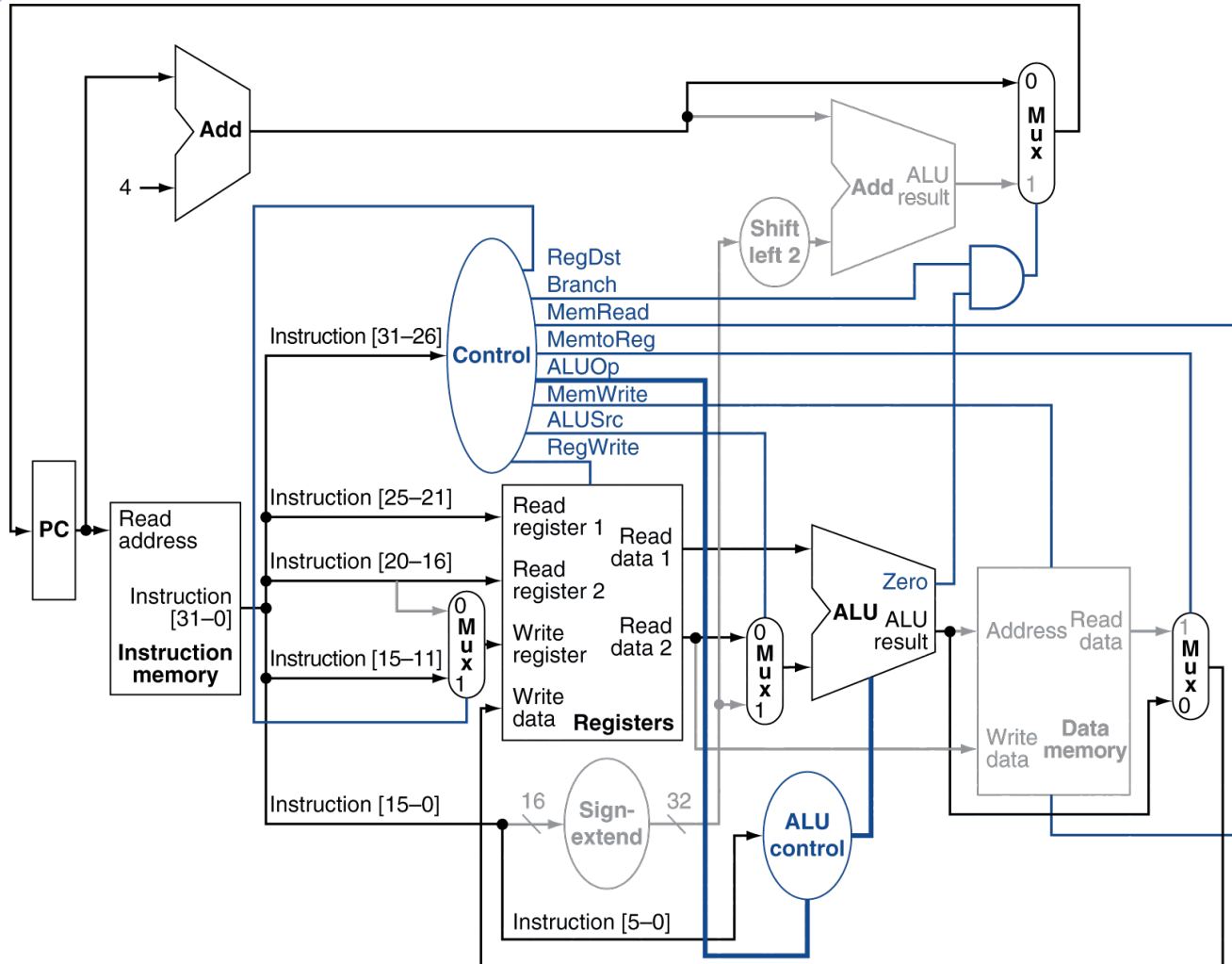


Khởi điều khiển (1/6)



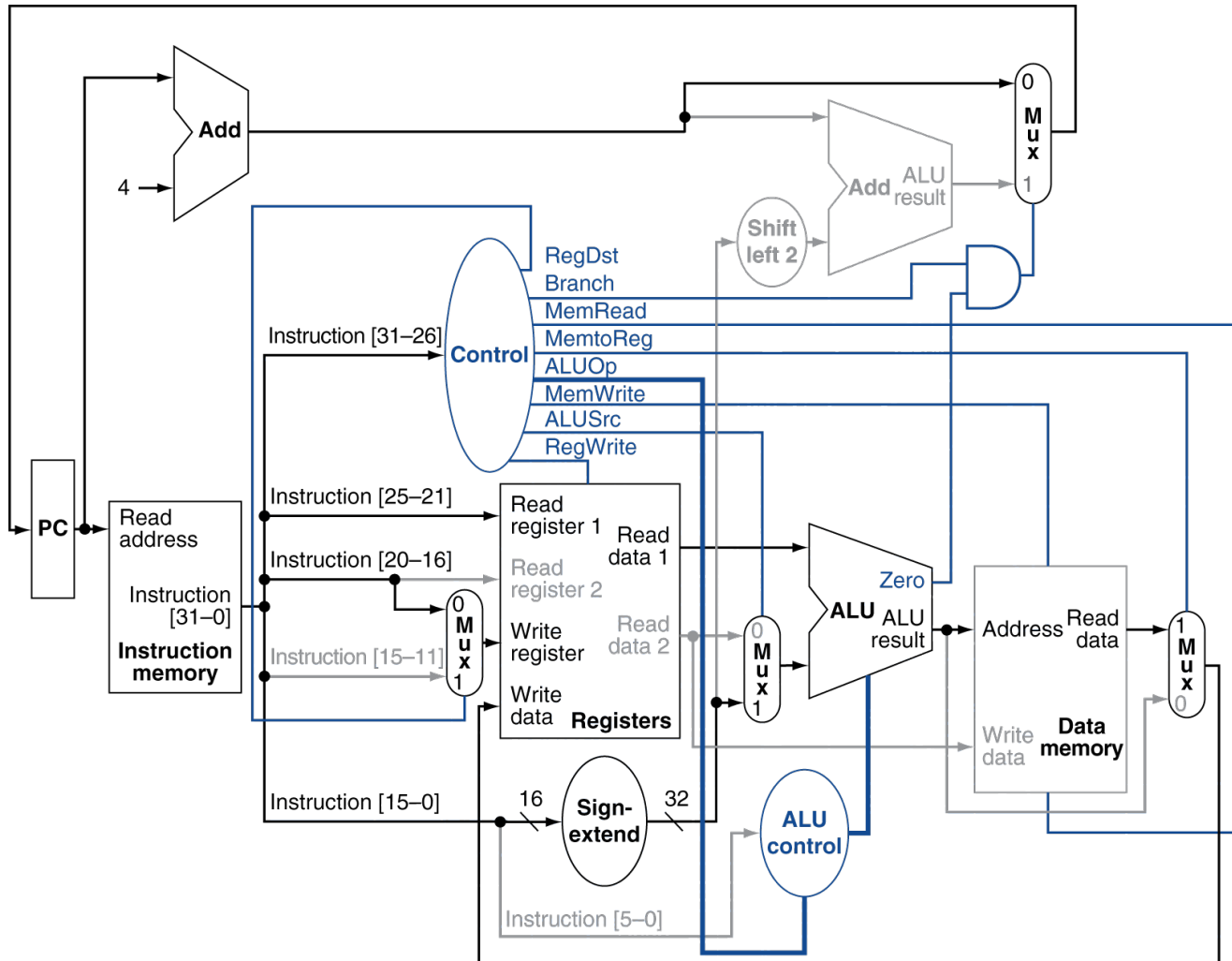


Khối điều khiển (2/6) – Loại R





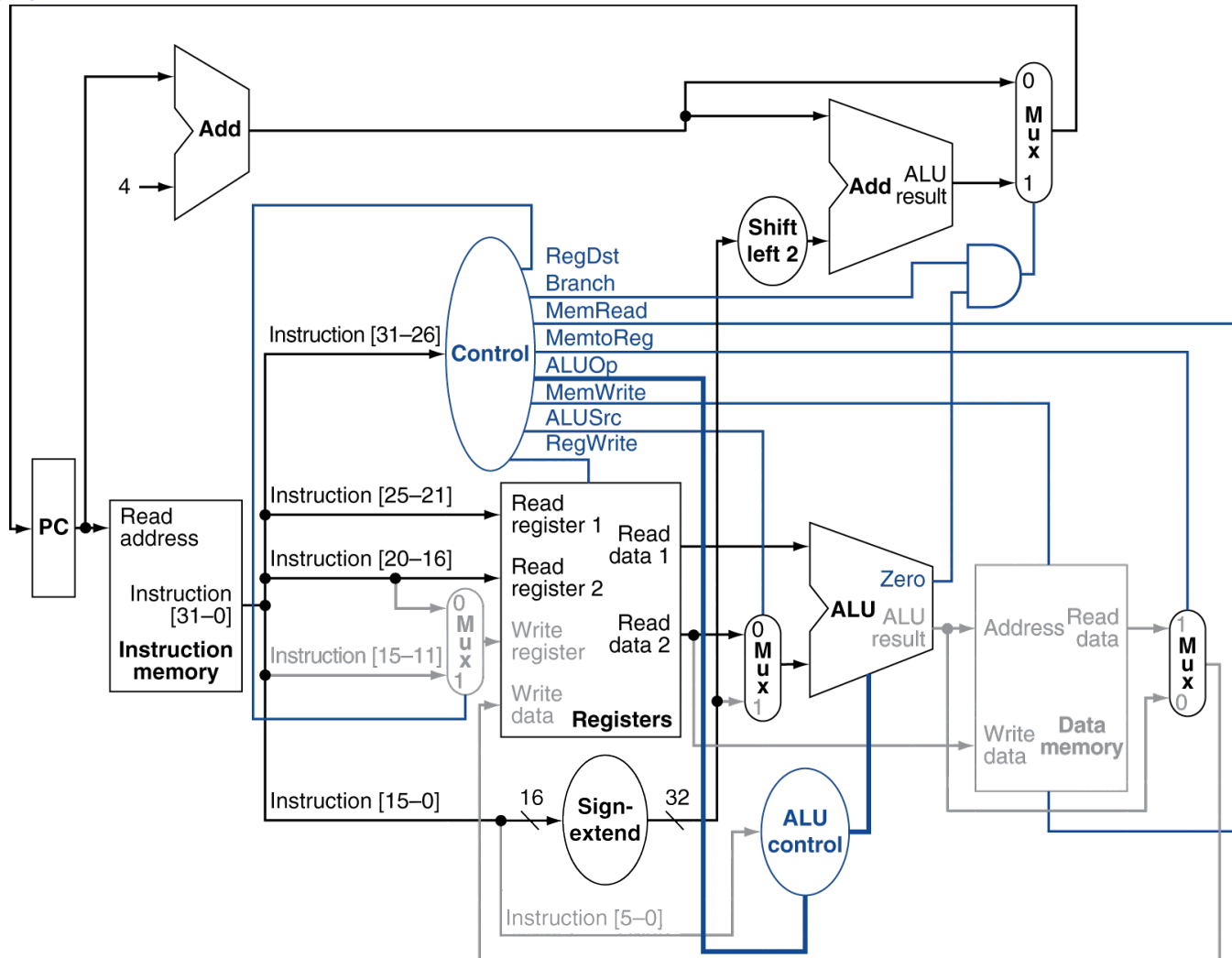
Khối điều khiển (3/6) – lw





Khối điều khiển (4/6) – beq

COMPUTER ENGINEERING





Khởi điều khiển (5/6) – ALU Control

- ALU chỉ cần thực hiện các phép toán: Cộng, trừ, AND, OR, Thiết lập nếu nhỏ hơn
- ALU Control là một mạch tổ hợp điều khiển ALU có bảng chân trị sau:

opcode	ALUOp	Lệnh	funct	Phép toán ALU	ALU control
lw	00	Nạp word	XXXXXX	Cộng	0010
sw	00	Lưu word	XXXXXX	Cộng	0010
beq	01	Nhảy nếu bằng	XXXXXX	Trừ	0110
Loại R	10	Cộng	100000	Cộng	0010
		Trừ	100010	Trừ	0110
		AND	100100	AND	0000
		OR	100101	OR	0001
		Thiết lập nếu nhỏ hơn	101010	Thiết lập nếu nhỏ hơn	0111



Khối điều khiển (6/6) - Control

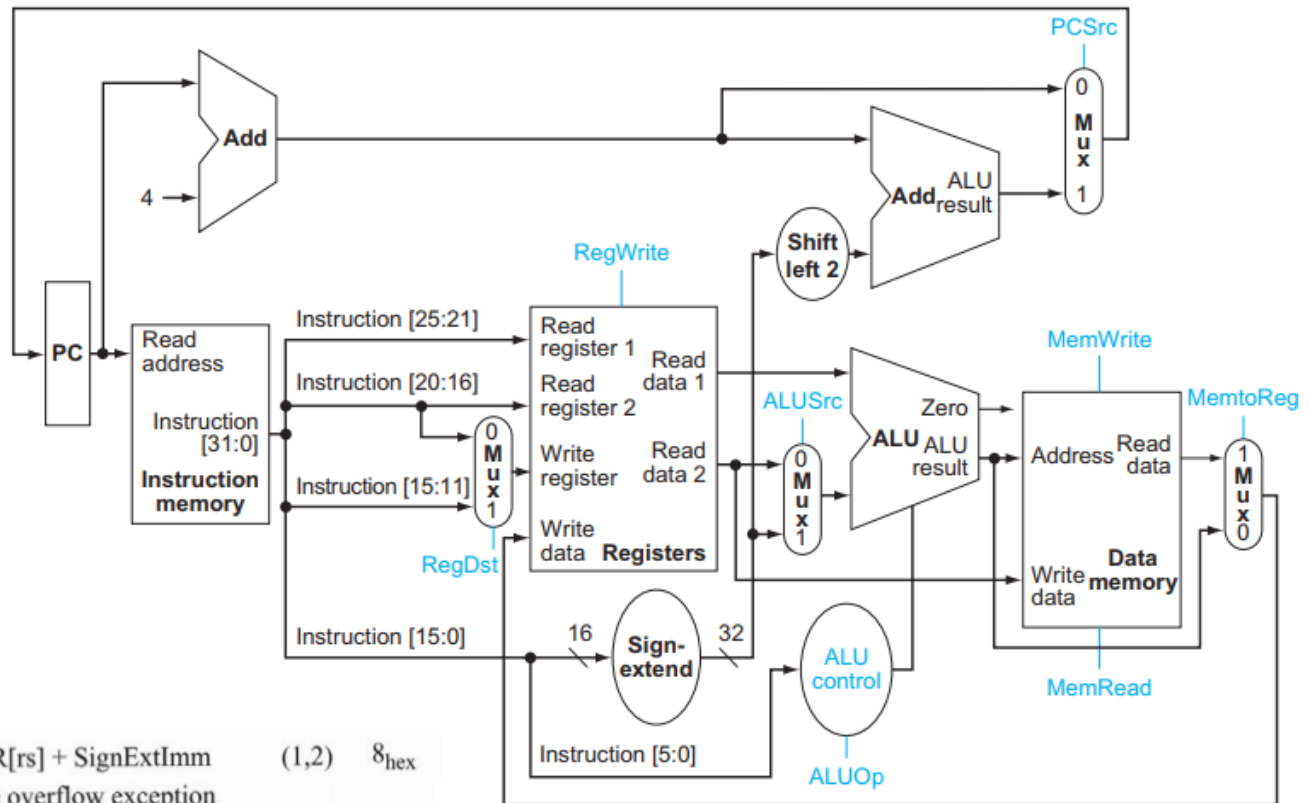
- Khối điều khiển (Control) một mạch tổ hợp điều khiển datapath hoạt động có bảng chân trị sau:

opcode	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemtoReg	Branch
lw	1	1	00	0	1	1	0
sw	0	1	00	1	0	X	0
beq	0	0	01	0	0	X	1
Loại R	1	0	10	0	0	0	0



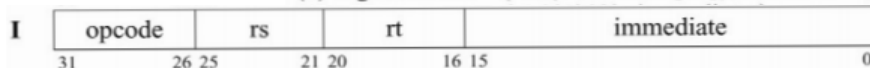
Bài tập (1/2)

Trình bày các tín hiệu điều khiển được sử dụng khi thực thi lệnh addi?



Add Immediate `addi` I $R[rt] = R[rs] + \text{SignExtImm}$ (1,2) 8_{hex}

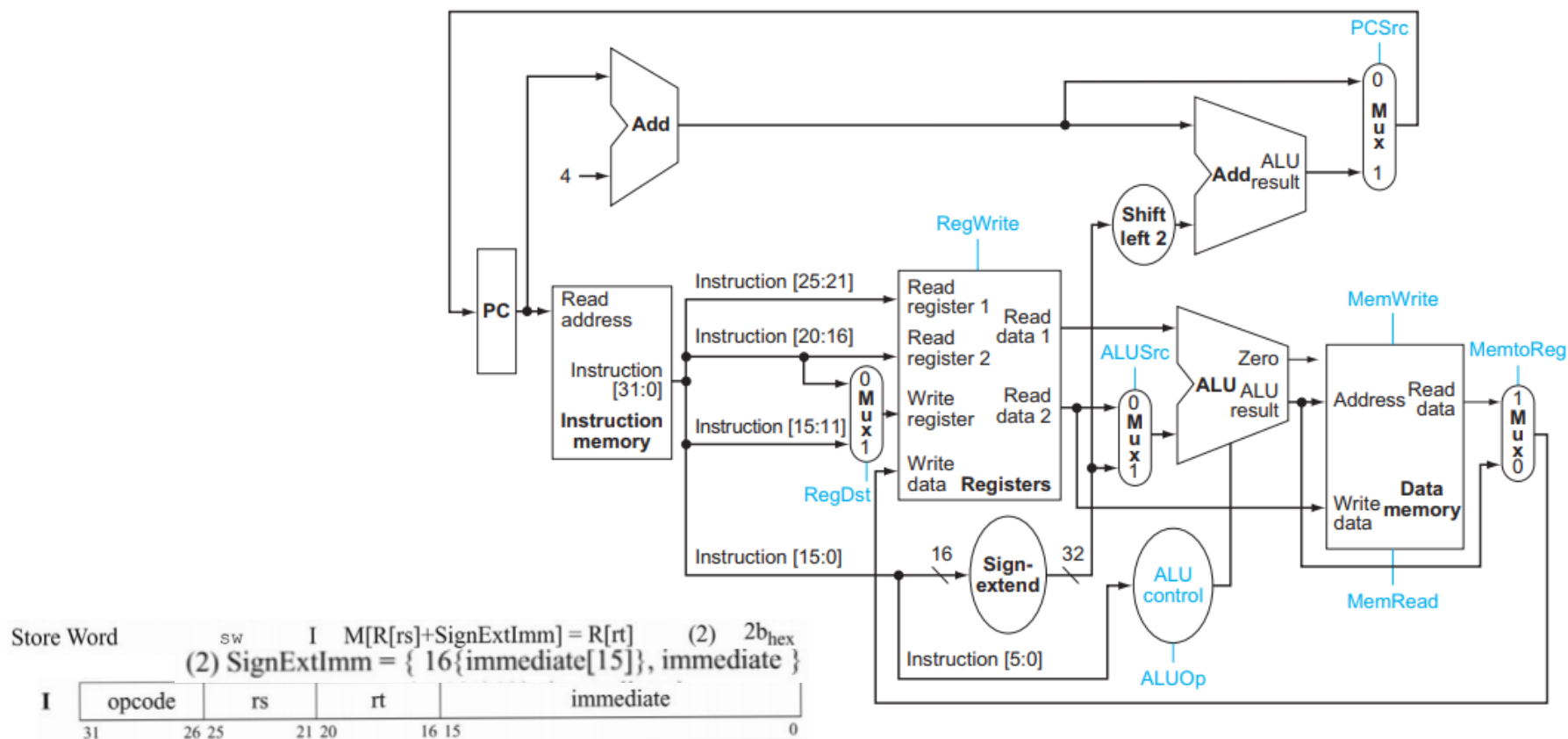
(1) May cause overflow exception
(2) $\text{SignExtImm} = \{ 16\{\text{immediate}[15]\}, \text{immediate} \}$





Bài tập (2/2)

Trình bày các tín hiệu điều khiển được sử dụng khi thực thi lệnh sw?





COMPUTER ENGINEERING



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

THẢO LUẬN

