



# KIẾN TRÚC MÁY TÍNH

## ÔN TẬP CUỐI KỲ

Ôn tập các lý thuyết đã trình bày trong 8 tuần học: các phép toán số học trong máy tính, bộ xử lý và pipeline

PHAN ĐÌNH DUY



# NỘI DUNG

1. Cấu trúc đề thi
2. Nội dung ôn thi
3. Ôn tập lý thuyết
4. Ôn tập bài tập
5. Giải đề thi mẫu
6. Hỏi – đáp – thảo luận



# 1. Cấu trúc đề thi

- Trắc nghiệm: 30 câu – 9.0 điểm (0.3 điểm/câu)
- Tự luận: 1 câu – 1 điểm
- Thời gian làm bài: 80 phút
- Thi theo lịch của phòng ĐT (11/1/2024 ca 2)



## 2. Nội dung ôn thi

- Tuần 1: Máy tính – Các khái niệm & Công nghệ
- Tuần 2: Hiệu suất máy tính
- Tuần 3: Kiến trúc tập lệnh
- **Tuần 4: Kiến trúc tập lệnh (ASM  $\Leftrightarrow$  Mã máy)**
- Tuần 5: Kiến trúc tập lệnh (tiếp theo)
- Tuần 6: Kiến trúc tập lệnh (tiếp theo)
- Tuần 7: Ôn thi giữa kỳ



## 2. Nội dung ôn thi (tt)

- Tuần 8-9: Các phép toán số học trong máy tính
- Tuần 10: Datapath trong bộ xử lý
- Tuần 11: Datapath trong bộ xử lý (tiếp theo)
- Tuần 12: Bài tập cho Datapath của Bộ xử lý
- Tuần 13: Kỹ thuật ống dẫn
- Tuần 14: Bài tập cho kỹ thuật ống dẫn
- Tuần 15: Ôn tập cuối kỳ



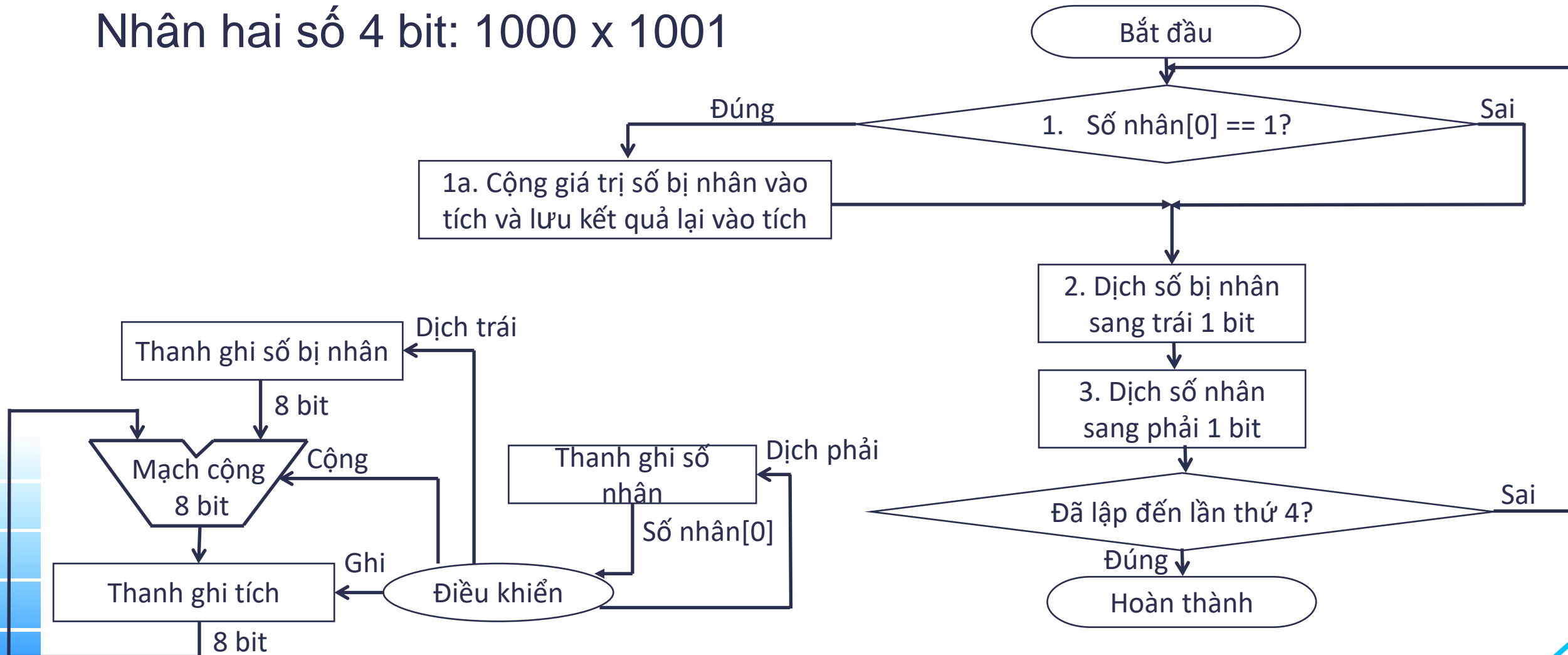
### 3. Ôn tập lý thuyết – chương 3

- Phép cộng
- Phép trừ
- Tràn số học
- Phép nhân
- Phép chia



### 3. Phép nhân

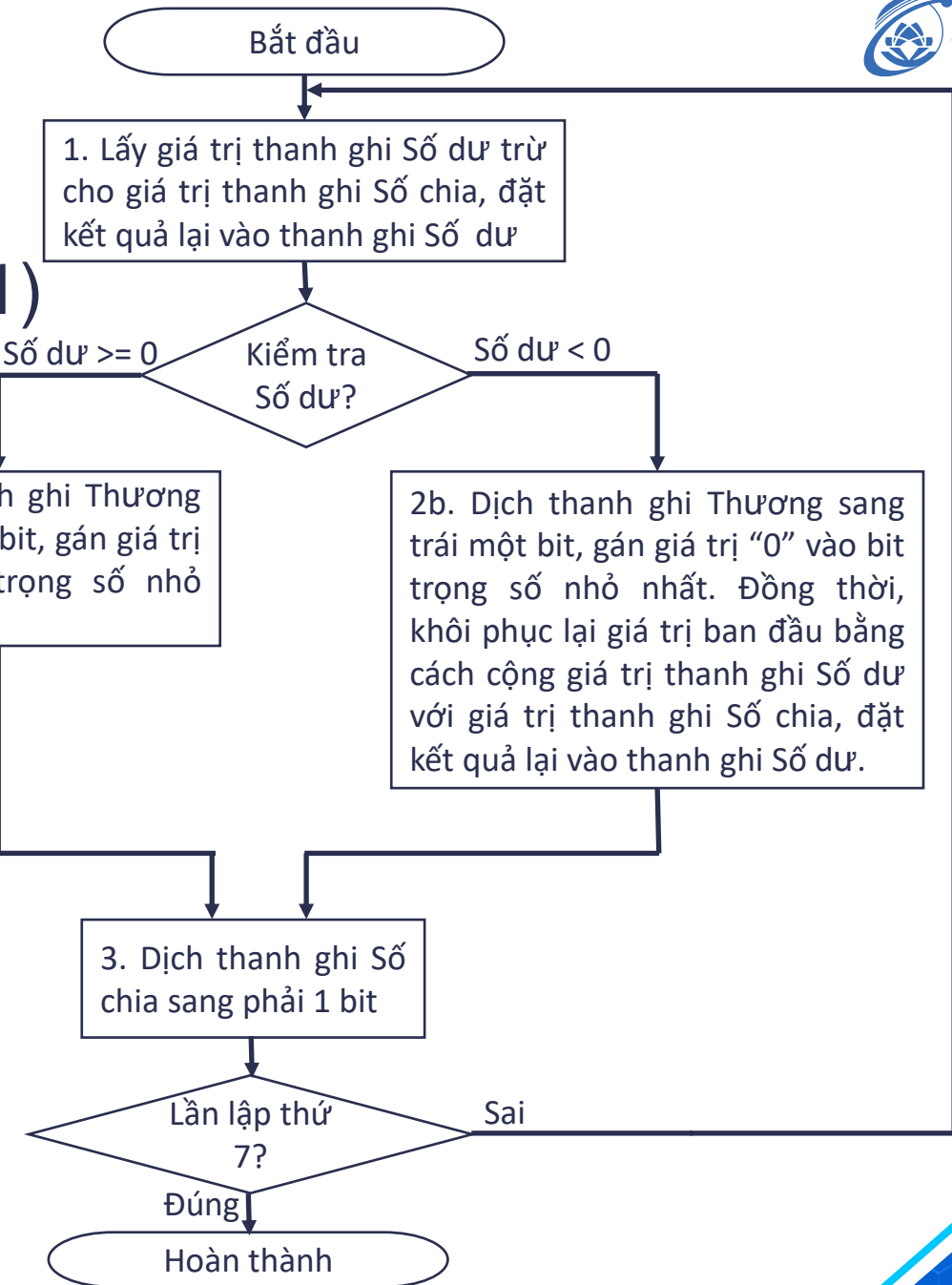
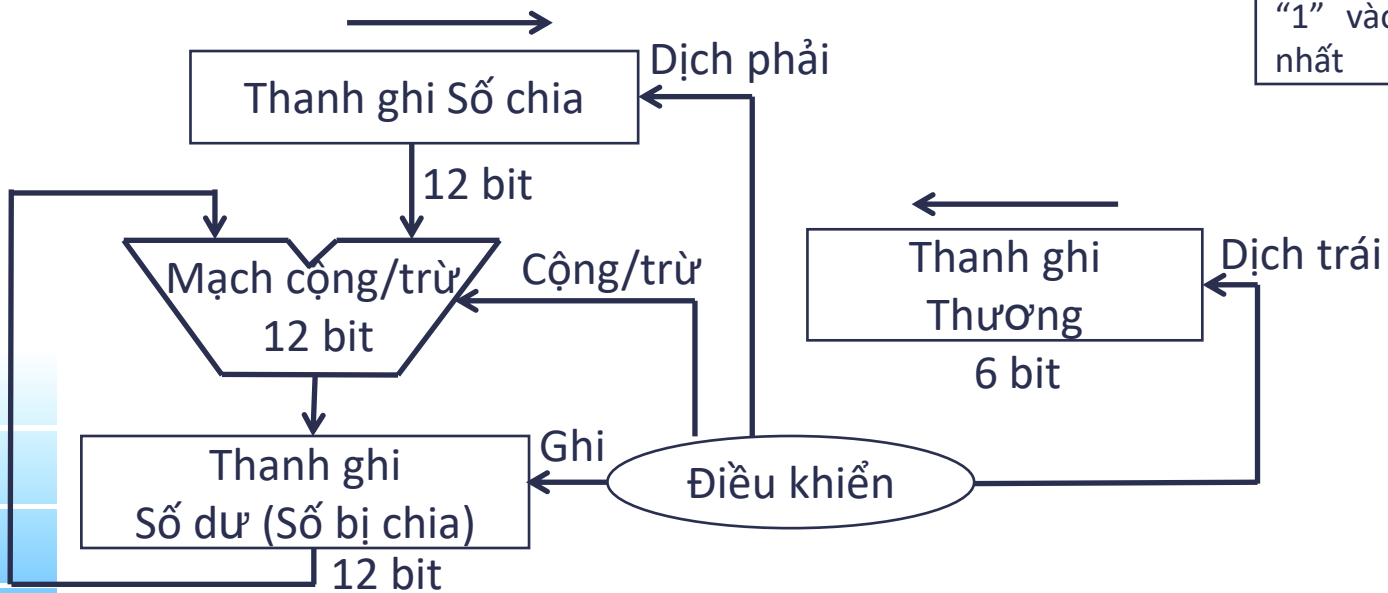
Nhân hai số 4 bit: 1000 x 1001





## 4. Phép chia

Chia hai số 6 bit: 40:19 (101000 : 010011)



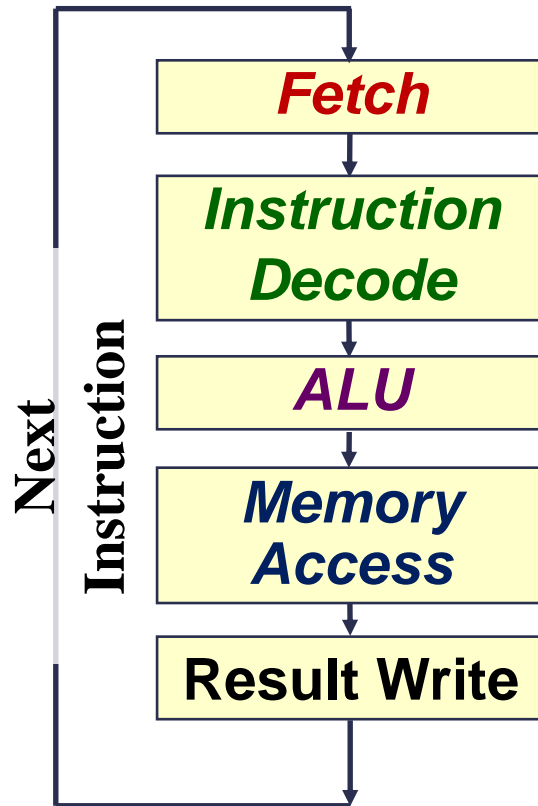




### 3. Ôn tập lý thuyết – chương 4

- Giới thiệu
- Nhắc lại các quy ước thiết kế logic
- Xây dựng đường dữ liệu (datapath) đơn giản
- Hiện thực datapath đơn chu kỳ

# Quy trình thực thi lệnh của MIPS (5 công đoạn)



■ **Instruction Fetch** (Nạp lệnh)

■ **Instruction Decode & Operand Fetch**  
(Giải mã và lấy các toán hạng cần thiết, Gọi tắt là “Instruction Decode”)

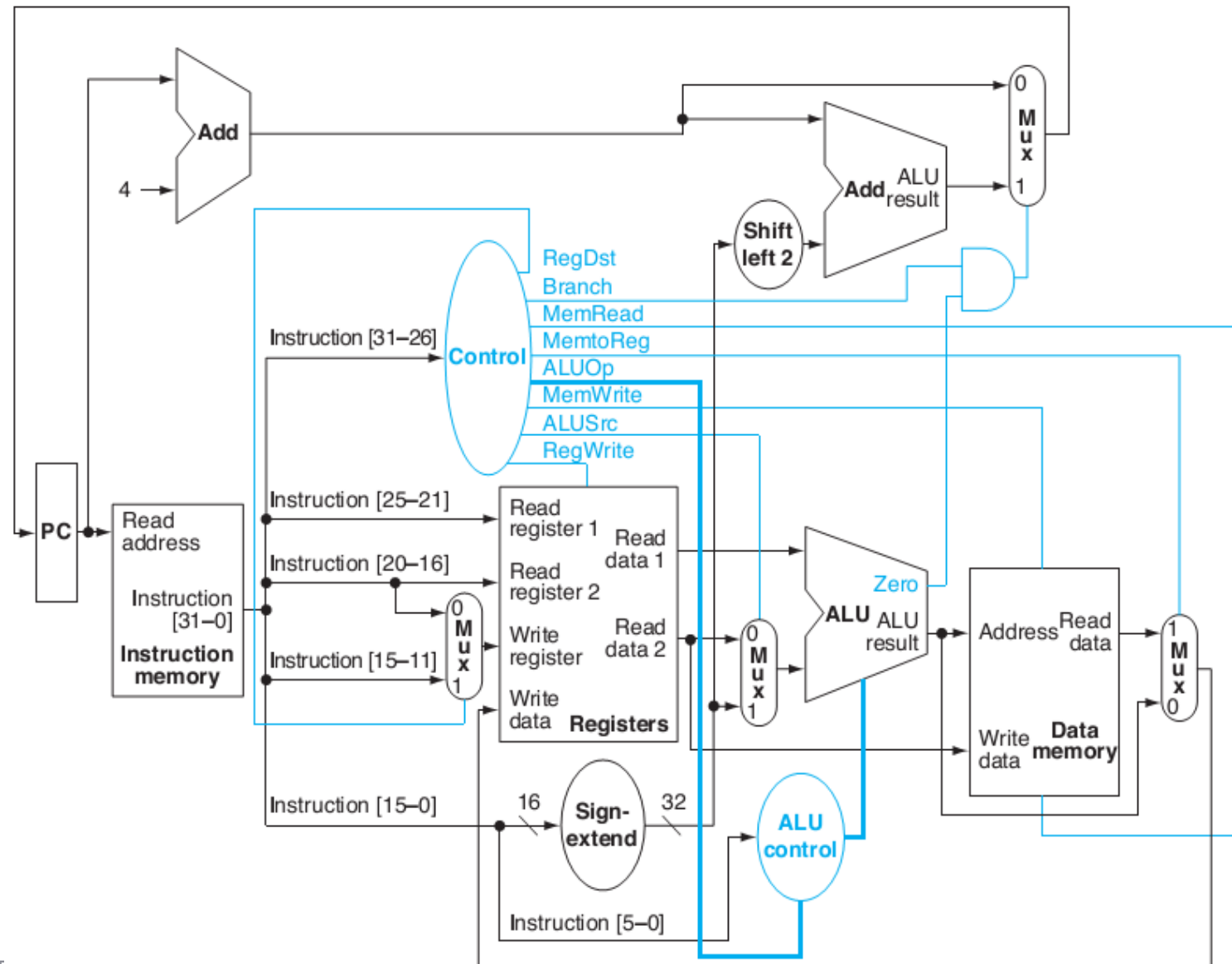
■ **ALU** (Giai đoạn sử dụng ALU hay giai đoạn thực thi)

■ **Memory Access** (Giai đoạn truy xuất vùng nhớ)

■ **Result Write** (Giai đoạn ghi lại kết quả/lưu trữ)



# 3. Ôn tập lý thuyết – chương 4





### 3. Ôn tập lý thuyết – chương 4

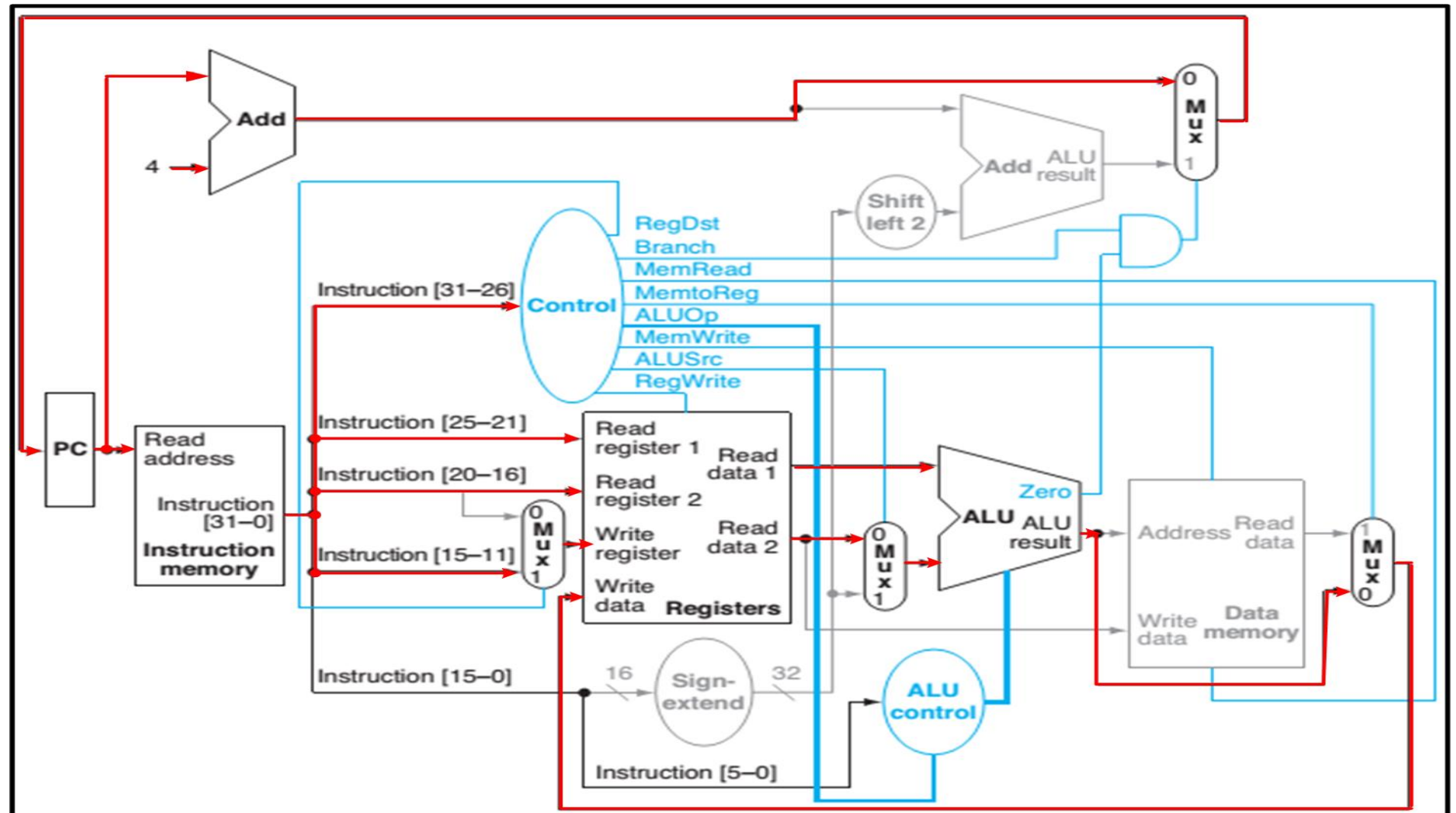
- Quá trình thực thi lệnh
- Các khối được sử dụng trong quá trình thực hiện lệnh  $\Rightarrow$  critical path của một lệnh  $\Rightarrow$  thời gian cần để thực thi lệnh
- Tín hiệu đầu vào và đầu ra của các khối
- Chu kỳ clock cho processor đơn chu kỳ: Thời gian thực thi của lệnh dài nhất



# Các đường hoạt động khi lệnh thuộc nhóm logic và số học thực thi

## I-Mem, Mux, Regs, Mux, ALU, Mux, Regs

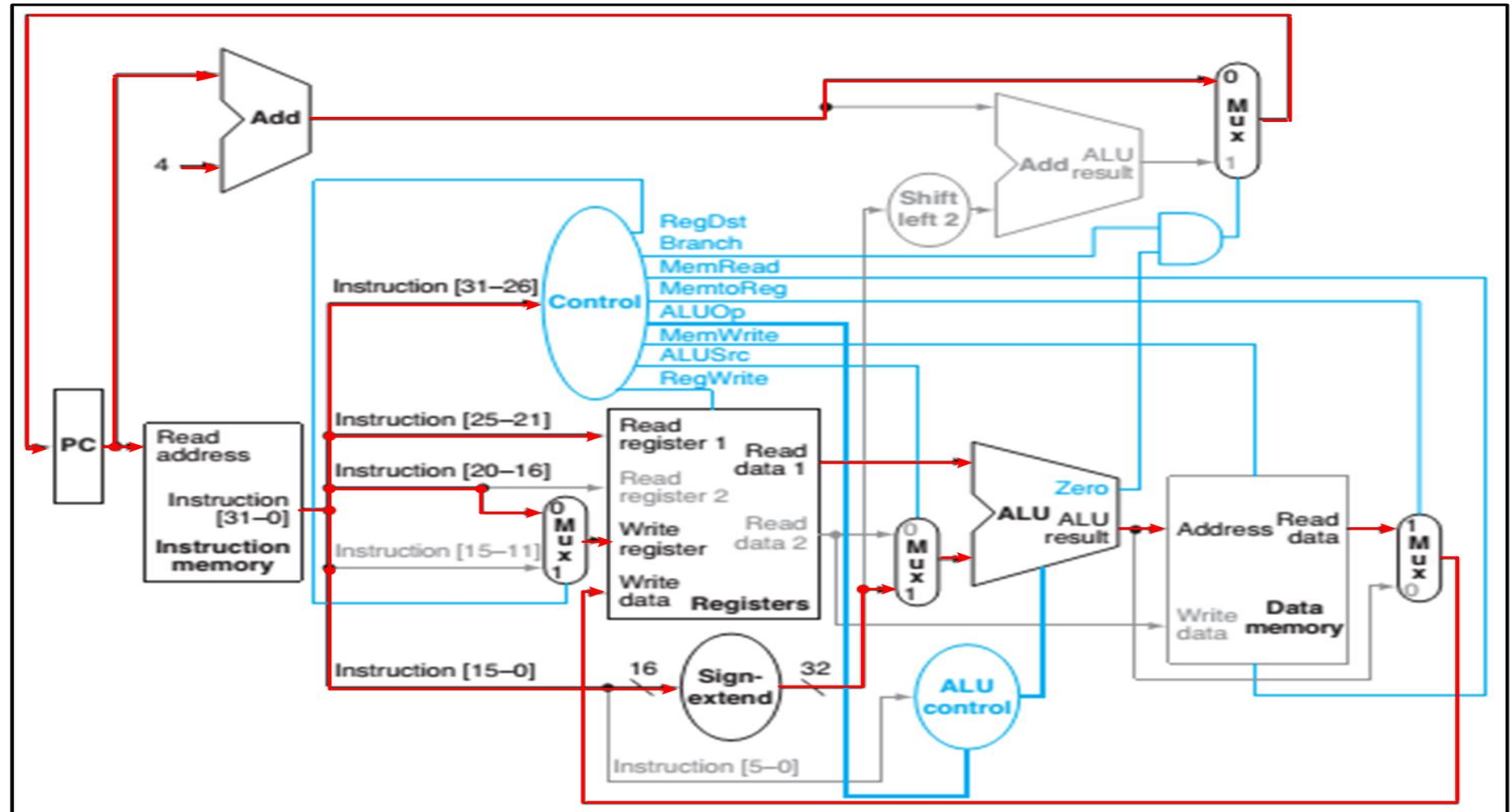
Trình bày  
các khối  
chức năng  
được sử  
dụng khi  
thực thi lệnh  
add?



# Các đường hoạt động khi lệnh lw thực thi

I-Mem, Mux, Regs, ALU, D-mem, Mux, Regs

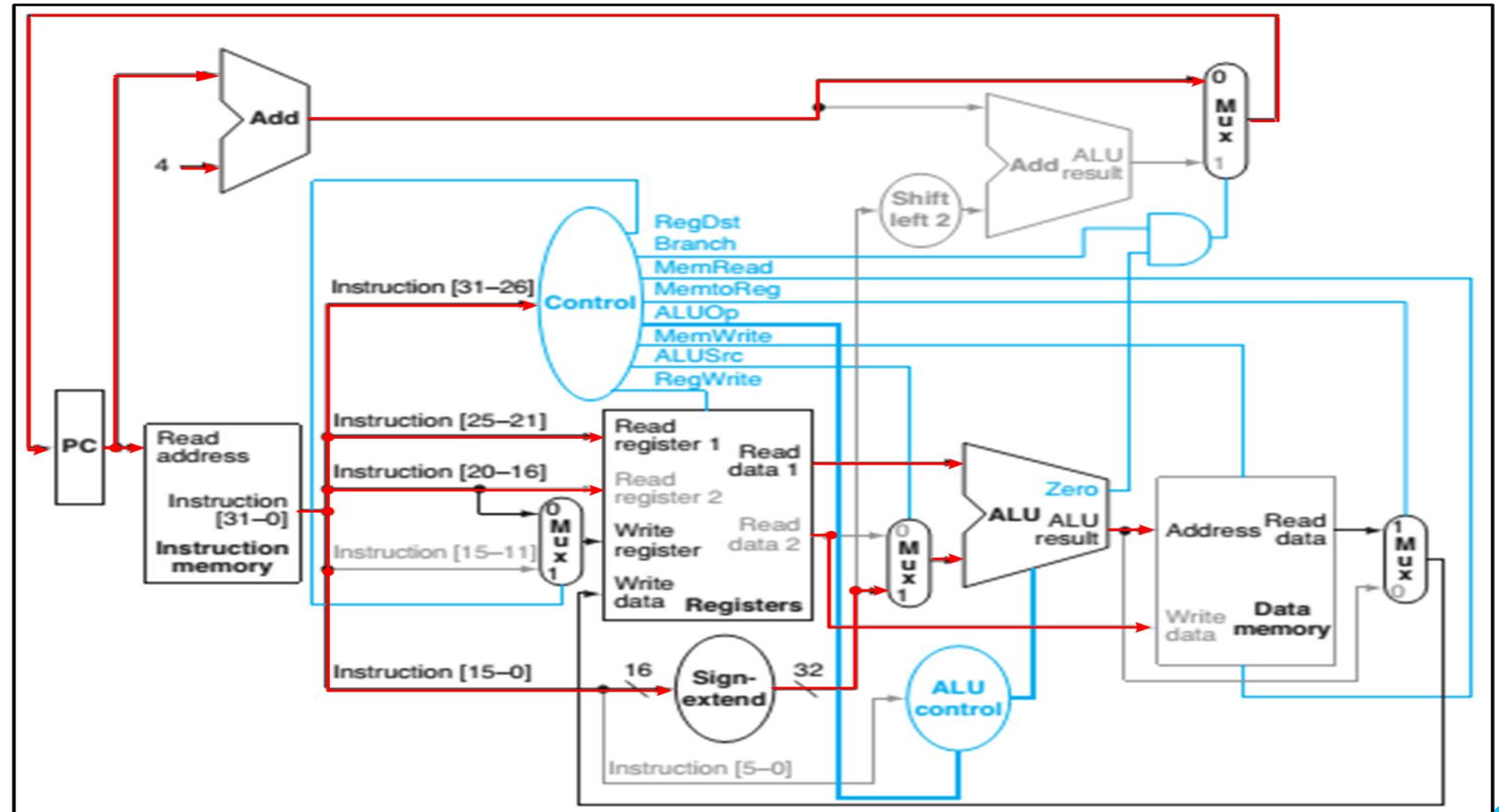
Trình bày  
các khối  
chức năng  
được sử  
dụng khi  
thực thi lệnh  
lw?



# Các đường hoạt động khi lệnh sw thực thi

## I-Mem, Regs, ALU, D-mem

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

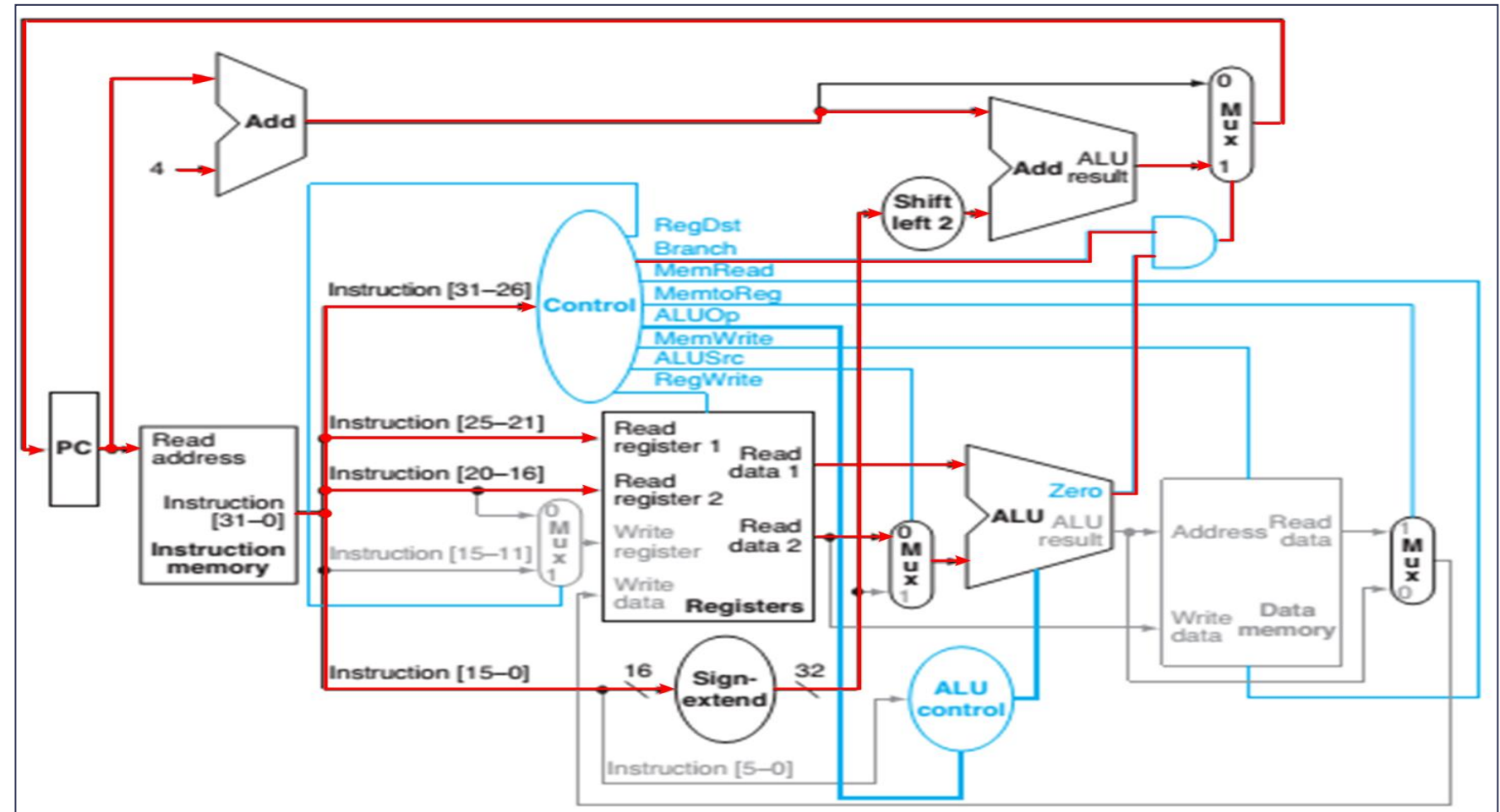




# Các đường hoạt động khi lệnh beq thực thi

I-Mem, Regs, Mux, ALU, Mux

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







# Control

**Giá trị các tín hiệu điều khiển tương ứng với mỗi lệnh như sau:**

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

*Khối “Control” trong datapath nhận input là 6 bits từ trường “opcode” của mã máy, dựa vào đó các tín hiệu điều khiển được sinh ra tương ứng như bảng.*

# ALU Control

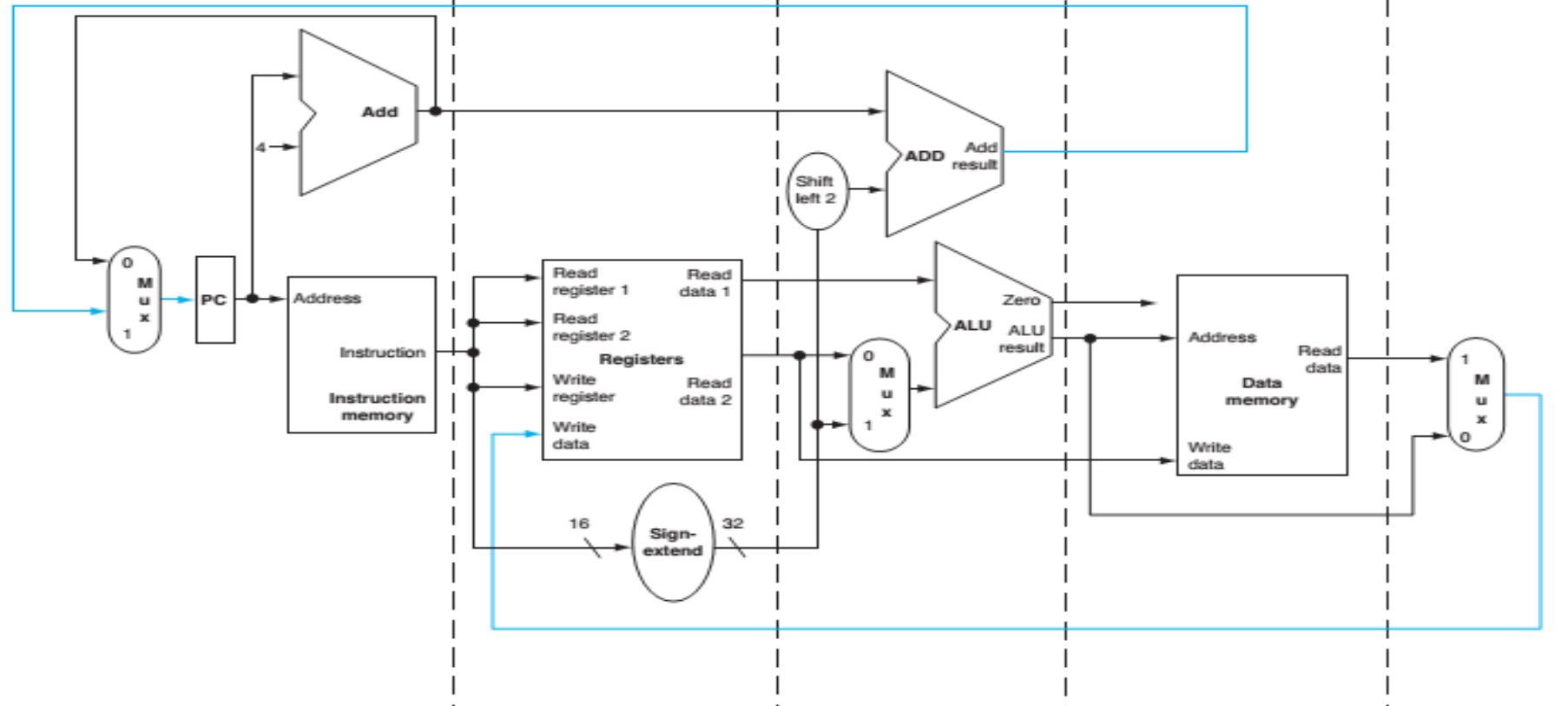
- “ALU Control” nhận input là 6 bits từ trường funct của mã máy, đồng thời dựa vào 2 bits “ALUOp” được sinh ra từ khối “Control” để sinh ra output là 4 bits điều khiển ALU, theo quy tắc như bảng sau:

Instruction opcode	ALUOp	Instruction operation	Funct field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	AND	0000
R-type	10	OR	100101	OR	0001
R-type	10	set on less than	101010	set on less than	0111



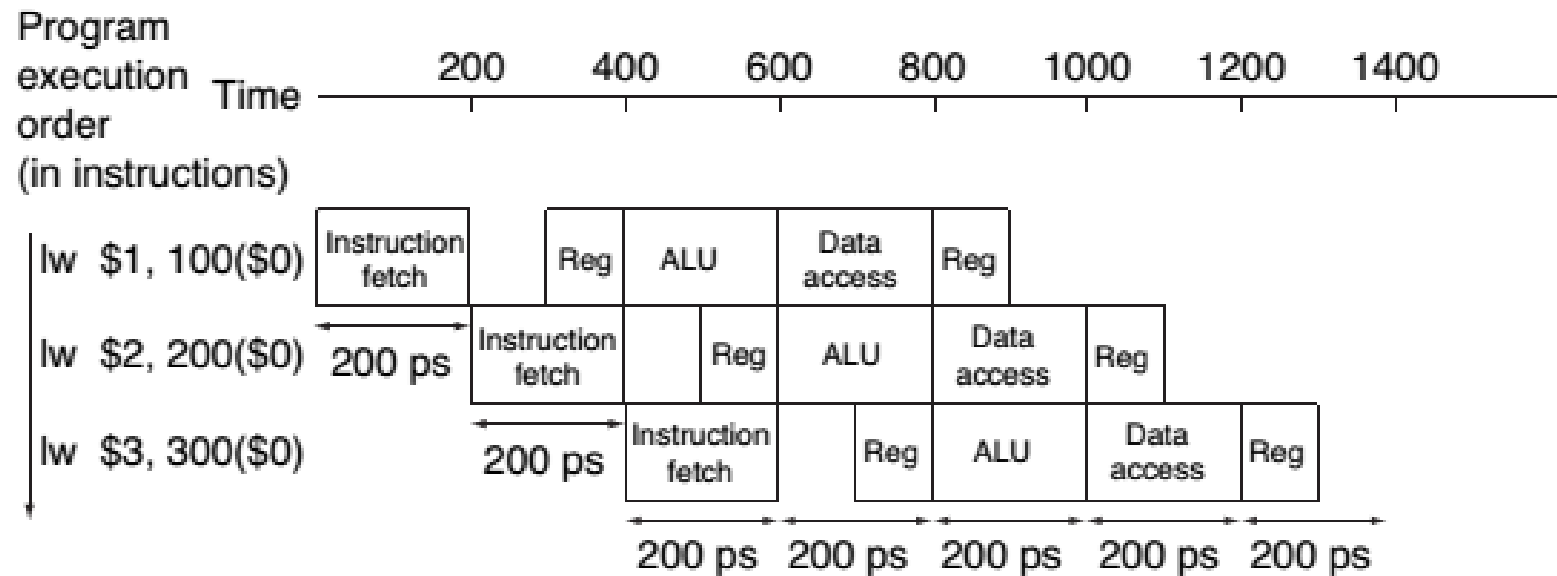
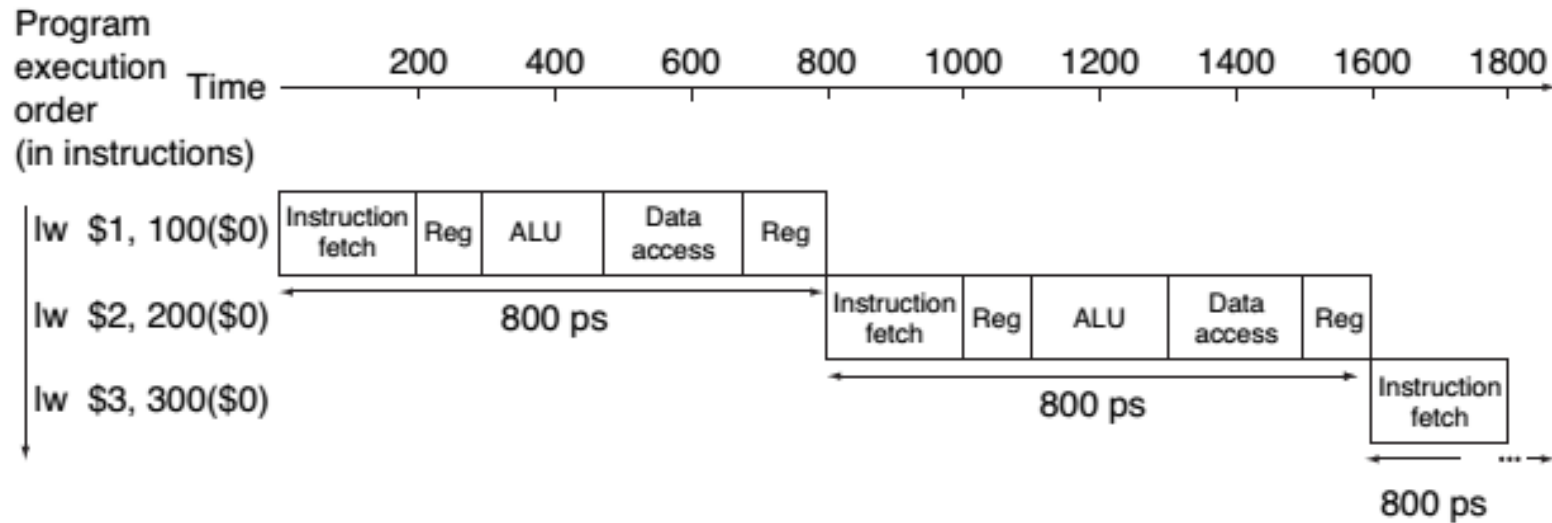
### 3. Ôn tập lý thuyết – chương 5

- Giới thiệu kỹ thuật pipeline
- Sự tăng tốc của kỹ thuật pipeline so với đơn chu kỳ, đa chu kỳ
- Các xung đột xảy ra khi thực hiện pipeline
  - Xung đột cấu trúc
  - Xung đột dữ liệu
  - Xung đột điều khiển

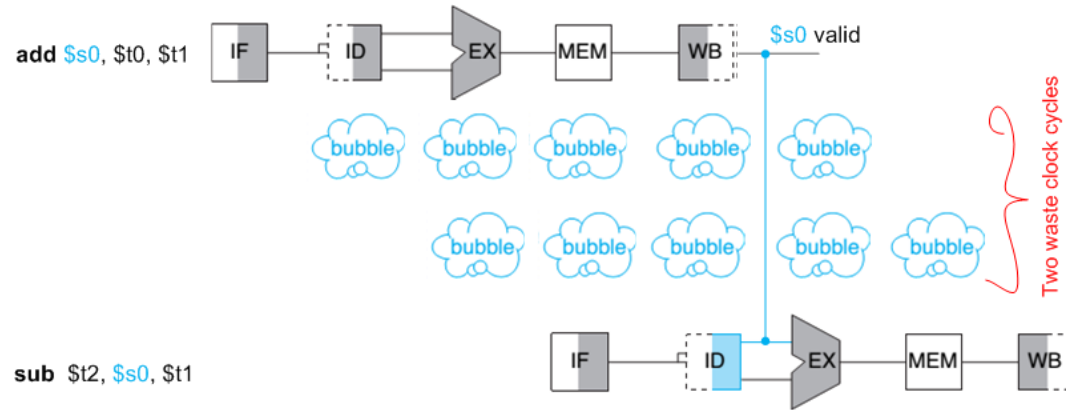


1. Nạp lệnh từ bộ nhớ – **IF**
2. Giải mã lệnh và đọc các thanh ghi – **ID**
3. Thực thi – **EX**
4. Truy xuất bộ nhớ – **MEM**
5. Ghi kết quả vào thanh ghi – **WB**

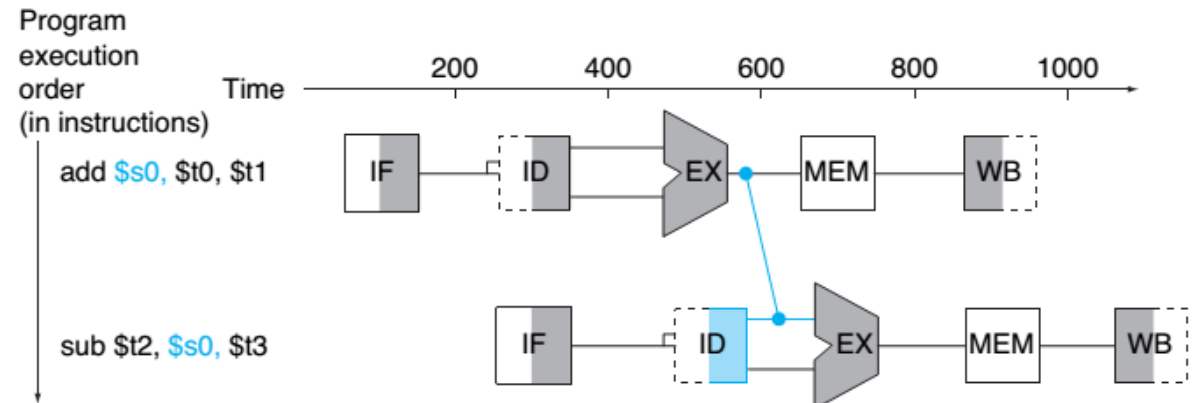
# 3. Ôn tập lý thuyết – chương 5



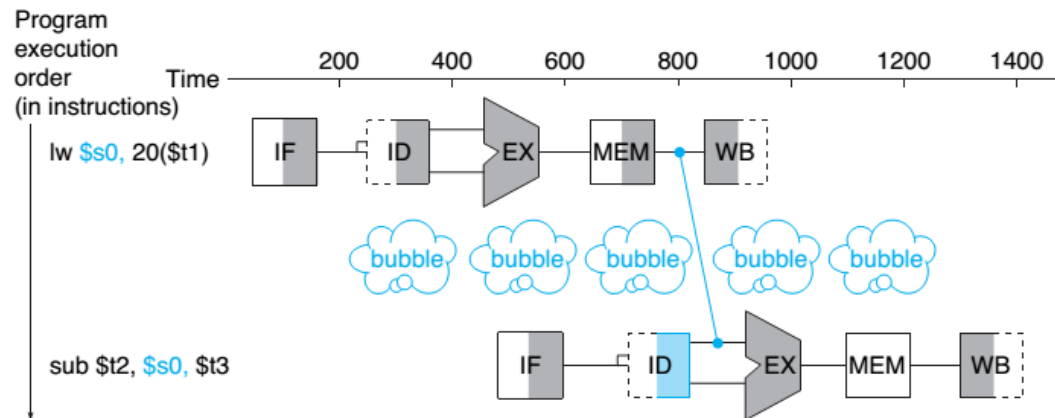
# 3. Ôn tập lý thuyết – chương 5



Non-forwarding



Full-forwarding





### 3. Ôn tập lý thuyết – chương 5

- Chu kỳ của đơn chu kỳ, đa chu kỳ pipeline, đa chu kỳ không pipeline
- Xử lý xung đột dữ liệu (full-forwarding hoặc non-forwarding)



# THẢO LUẬN

