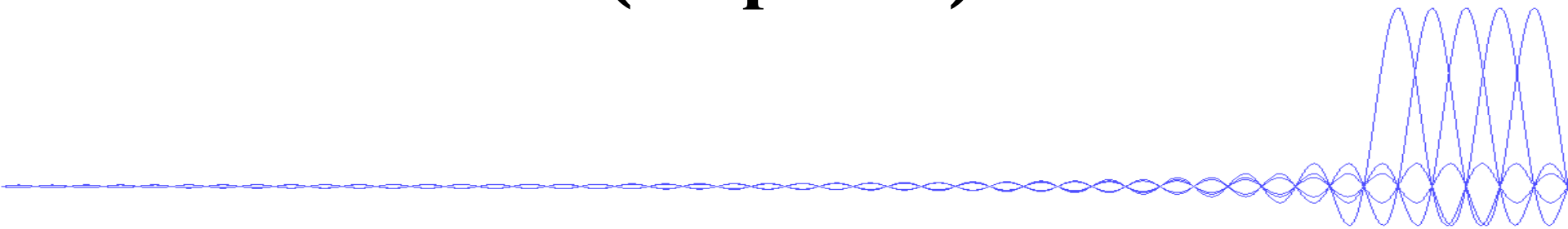
COMPUTER ENGINEERING

KIẾN TRÚC MÁY TÍNH

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Tuần 11

BỘ XỬ LÝ
PROCESSOR
(Tiếp theo)



Mục đích:

- ✓ Hiểu cơ chế thực thi lệnh và các quy ước về thiết kế logic
- ✓ Thiết kế Datapath với 8 lệnh cơ bản cho một bộ xử lý và cách hiện thực thiết kế này.

Slide tham khảo từ:

1. ***Computer Organization and Design: The Hardware/Software Interface***, Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, Revised Fourth Edition, 2011.
2. ***NUS***, Singapore

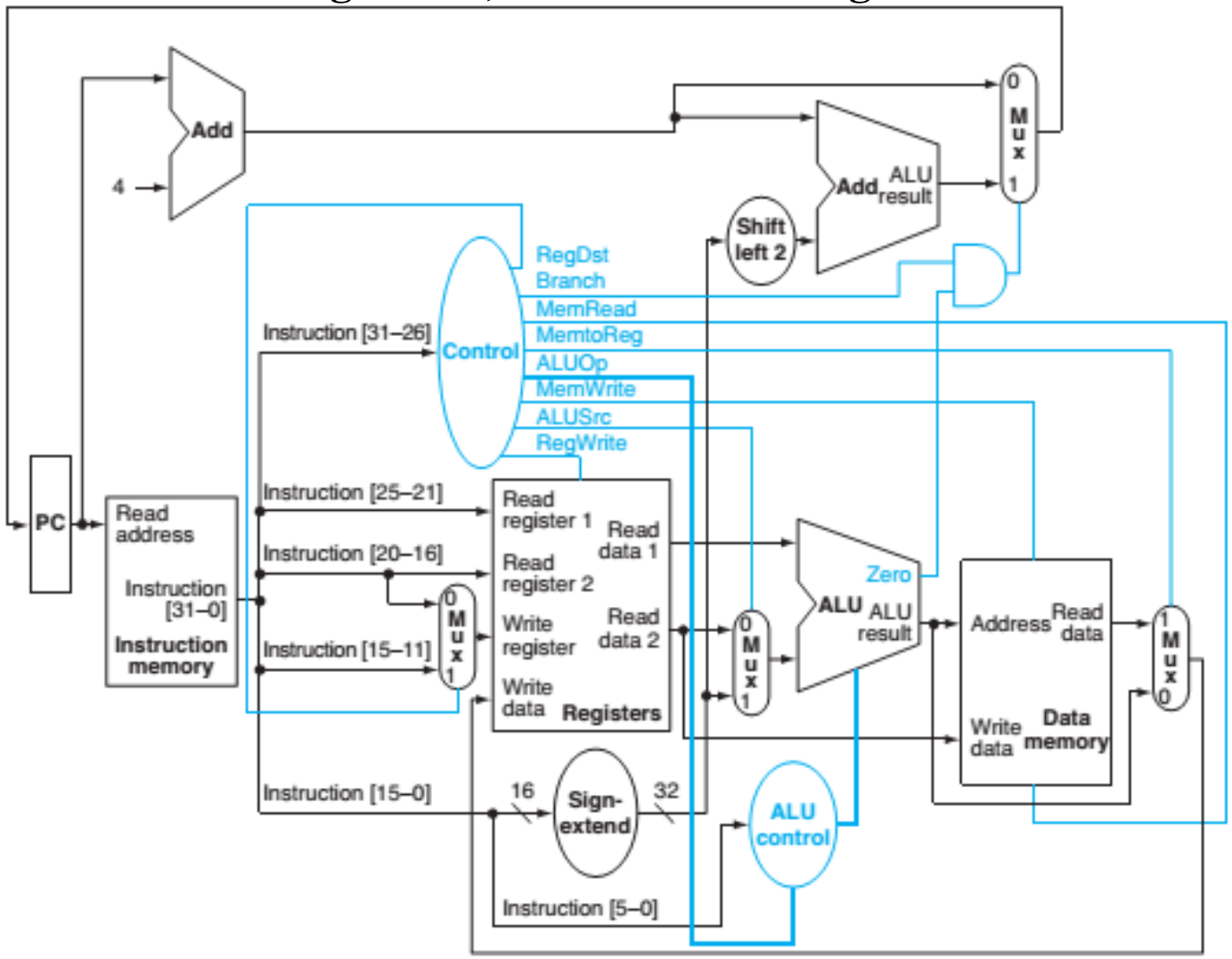
BỘ XỬ LÝ

1. Giới thiệu
2. Nhắc lại các quy ước thiết kế logic
3. Xây dựng đường dữ liệu (datapath) đơn giản
4. **Hiện thực datapath đơn chu kỳ**

COMPUTER ENGINEERING

Hiện thực datapath

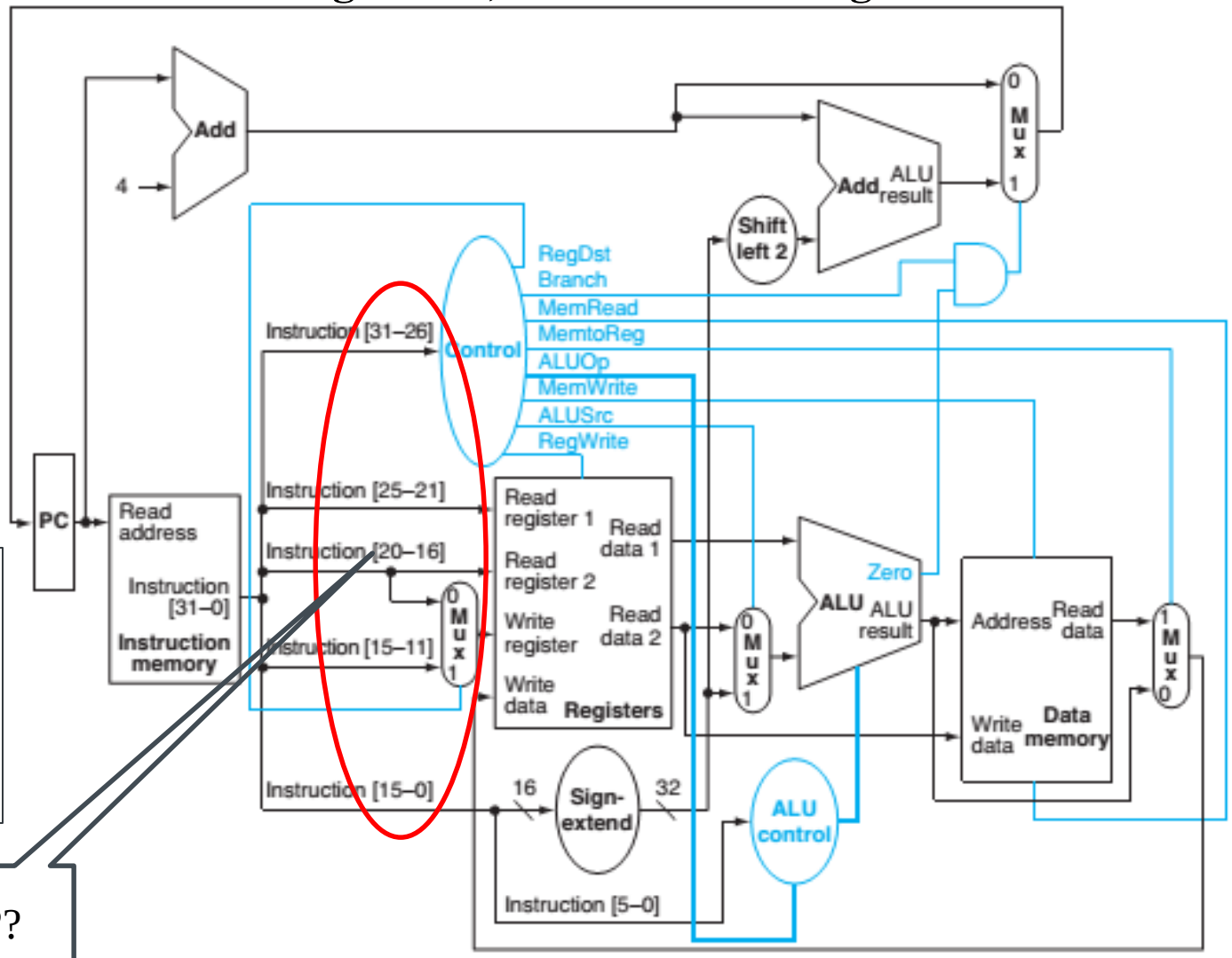
1. Inputs của khối "Registers", "Control" và "Sign-extend"



Datapath với đầy đủ dữ liệu input cho từng khối

Hiện thực datapath

1. Inputs của khối "Registers", "Control" và "Sign-extend"



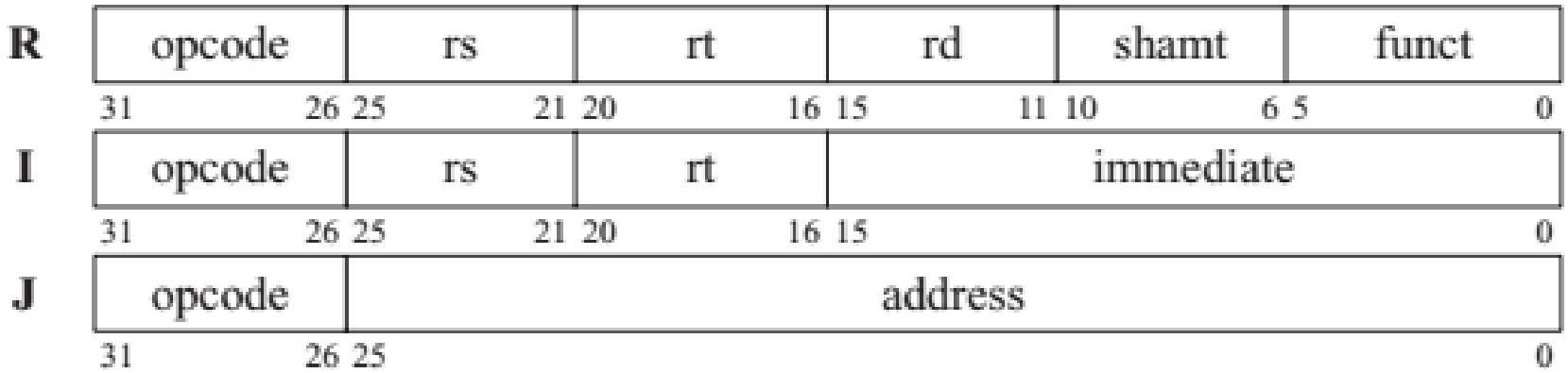
MUX có thêm 0 và 1 ở các ngõ vào ???

???

Datapath với đầy đủ dữ liệu input cho từng khối

COMPUTER ENGINEERING

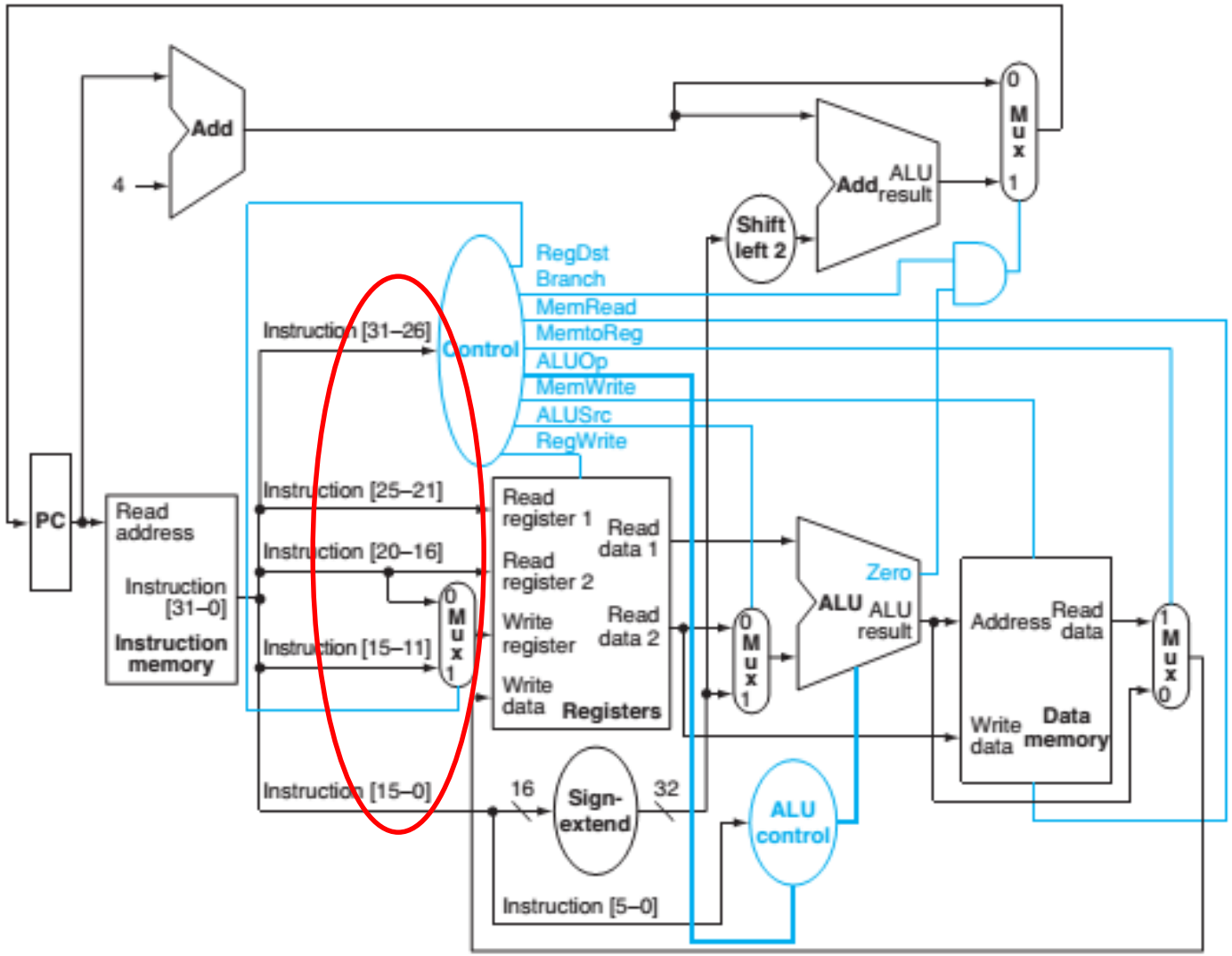
Hiện thực datapath



- Trường op (hay opcode) luôn chứa bits từ 31:26.
- Hai thanh ghi dùng để đọc trong tất cả các lệnh luôn luôn là rs và rt, tại vị trí bits từ 25:21 và 20:16.
- Thanh ghi nền cho lệnh load và store luôn là *rs* và tại vị trí bits 25:21.
- 16 bits offset cho *beq, lw* và *sw* thì luôn tại vị trí 15:0.
- Các thanh ghi đích dùng để ghi kết quả vào ở hai vị trí: Với *lw*, thanh ghi đích tại vị trí bits từ **20:16 (*rt*),** trong khi với nhóm lệnh logic và số học, thanh ghi đích ở vị trí **15:11 (*rd*).** Vì vậy, một multiplexer cần sử dụng ở đây để lựa chọn thanh ghi nào sẽ được ghi.

COMPUTER ENGINEERING

Hiện thực datapath

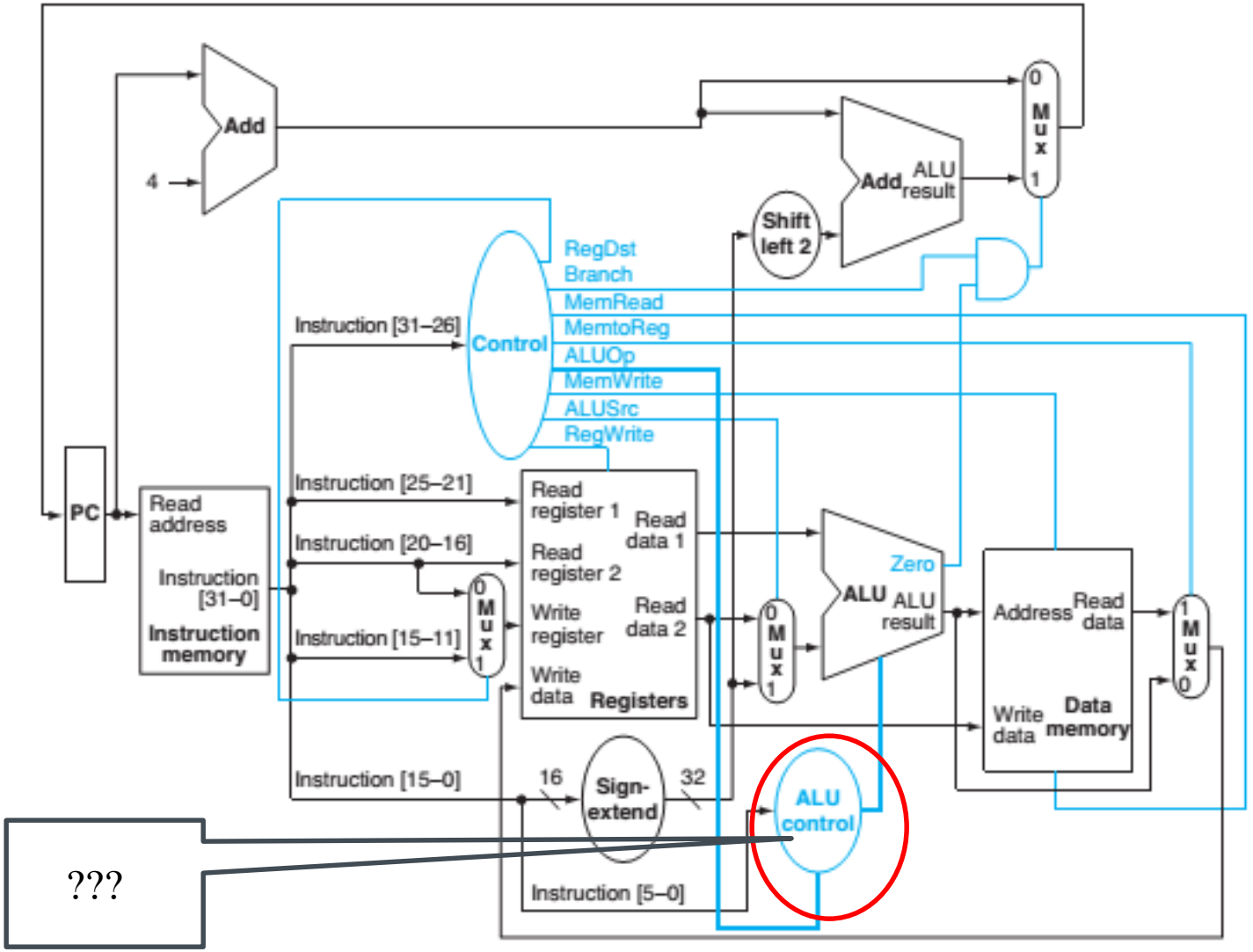


Datapath với đầy đủ dữ liệu input cho từng khối

COMPUTER ENGINEERING

Hiện thực datapath

2. Khối "ALU Control"



COMPUTER ENGINEERING

Hiện thực datapath

Bộ ALU của MIPS gồm 6 chức năng tính toán dựa trên 4 bits điều khiển đầu vào:

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

Tùy thuộc vào từng nhóm lệnh mà ALU sẽ thực hiện 1 trong 5 chức năng đầu (NOR sẽ được dùng cho các phần khác)

- Với các lệnh **load word** và **store word**, ALU sử dụng chức năng **'add'** để tính toán địa chỉ của bộ nhớ
- Với các lệnh thuộc **nhóm logic và số học**, ALU thực hiện 1 trong 5 chức năng (***AND, OR, subtract, add,*** *và* ***set on less than***), tùy thuộc vào giá trị của trường funct (6 bits) trong mã máy lệnh.
- Với lệnh **nhảy nếu bằng**, ALU thực hiện chức năng **'subtract'** để xem điều khiện bằng có đúng không.

COMPUTER ENGINEERING

Hiện thực datapath

Như vậy, để sinh ra 4 bits điều khiển ALU, một trong số các cách hiện thực có thể là sử dụng thêm một khối điều khiển "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

Một gợi ý để sinh ra 4 bits điều khiển ALU dựa vào trường "opcode" và trường "funct" của mã máy.

COMPUTER ENGINEERING

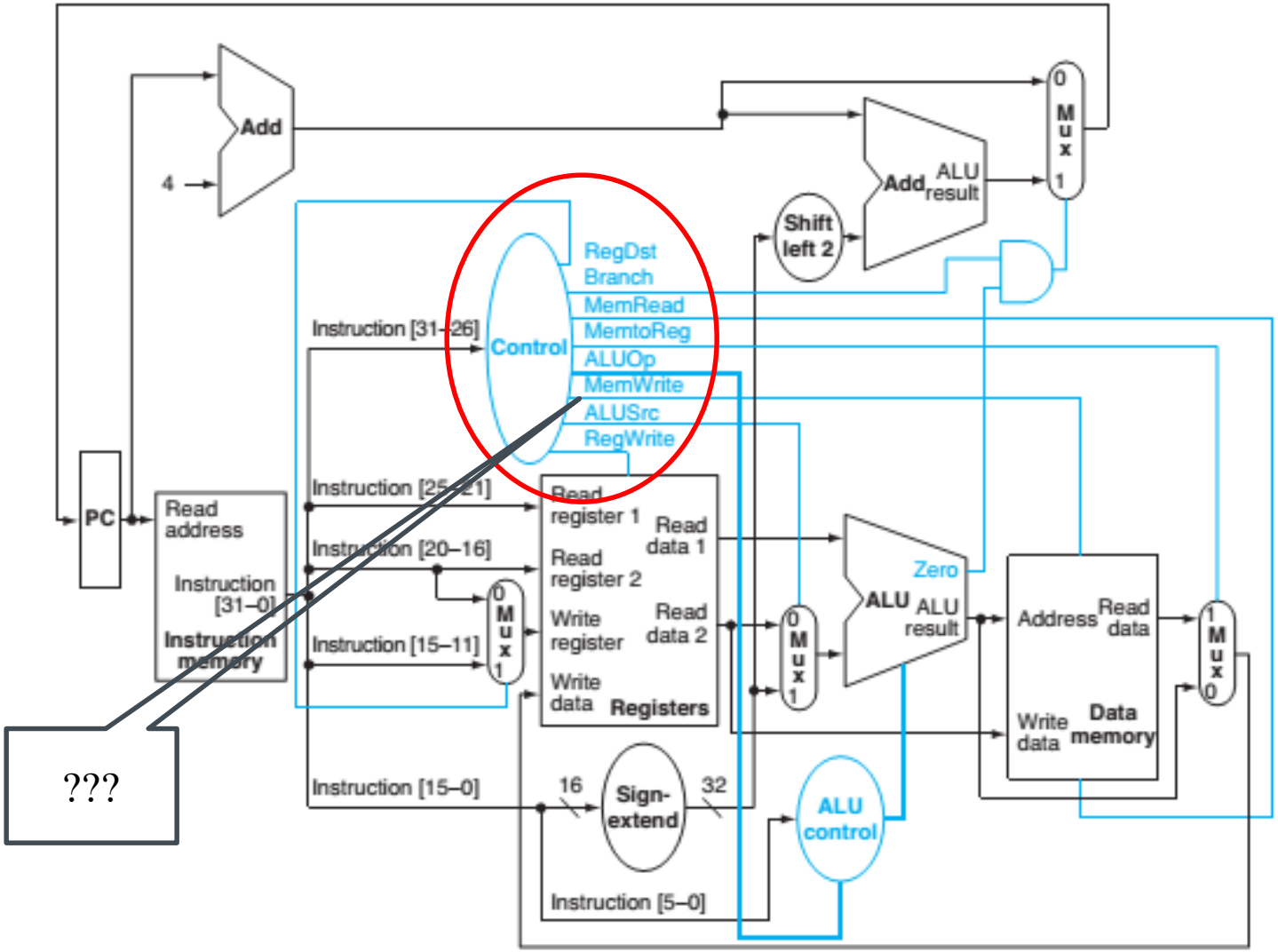
Hiện thực datapath

Bảng sự thật: Từ quy tắc hoạt động, bảng sự thật gợi ý cho khối "ALU Control" như sau

ALUOp		Funct field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
0	1	X	X	X	X	X	X	0110
1	0	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	0	X	X	0	1	0	0	0000
1	0	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

Hiện thực datapath

3. Khối điều khiển chính "Control"



COMPUTER ENGINEERING

Hiện thực datapath

Các tín hiệu điều khiển	Tác động khi ở mức thấp	Tác động khi ở mức cao
RegDst	Thanh ghi đích cho thao tác ghi sẽ từ thanh ghi *rt* (bits từ 20:16)	Thanh ghi đích cho thao tác ghi sẽ từ thanh ghi *rd* (bits từ 15:11)
RegWrite	Khối "Registers" chỉ thực hiện mỗi chức năng đọc thanh ghi	Ngoài chức năng đọc, khối "Register" sẽ thực hiện thêm chức năng ghi. Thanh ghi được ghi là thanh ghi có chỉ số được đưa vào từ ngõ "Write register" và dữ liệu dùng ghi vào thanh ghi này được lấy từ ngõ "Write data"
ALUSrc	Input thứ hai cho ALU đến từ "Read data 2" của khối "Registers"	Input thứ hai cho ALU đến từ output của khối "Sign-extend"
Branch	Cho biết lệnh nạp vào không phải "beq". Thanh ghi PC nhận giá trị là PC + 4	Lệnh nạp vào là lệnh "beq", kết hợp với điều kiện bằng thông qua cổng AND nhằm xác định xem lệnh tiếp theo có nhảy đến địa chỉ mới hay không. Nếu điệu kiện bằng đúng, PC nhận giá trị mới từ kết quả của bộ cộng "Add"
MemRead	(Không)	Khối "Data memory" thực hiện chức năng đọc dữ liệu. Địa chỉ dữ liệu cần đọc được đưa vào từ ngõ "Address" và nội dung đọc được xuất ra ngõ "Read data"
MemWrite	(Không)	Khối "Data memory" thực hiện chức năng ghi dữ liệu. Địa chỉ dữ liệu cần ghi được đưa vào từ ngõ "Address" và nội dung ghi vào lấy từ ngõ "Write data"
MemtoReg	Giá trị đưa vào ngõ "Write data" đến từ ALU	Giá trị đưa vào ngõ "Write data" đến từ khối "Data memory"

Tác động của các tín hiệu điều khiển

COMPUTER ENGINEERING

Hiện thực datapath

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.

Hiện thực datapath

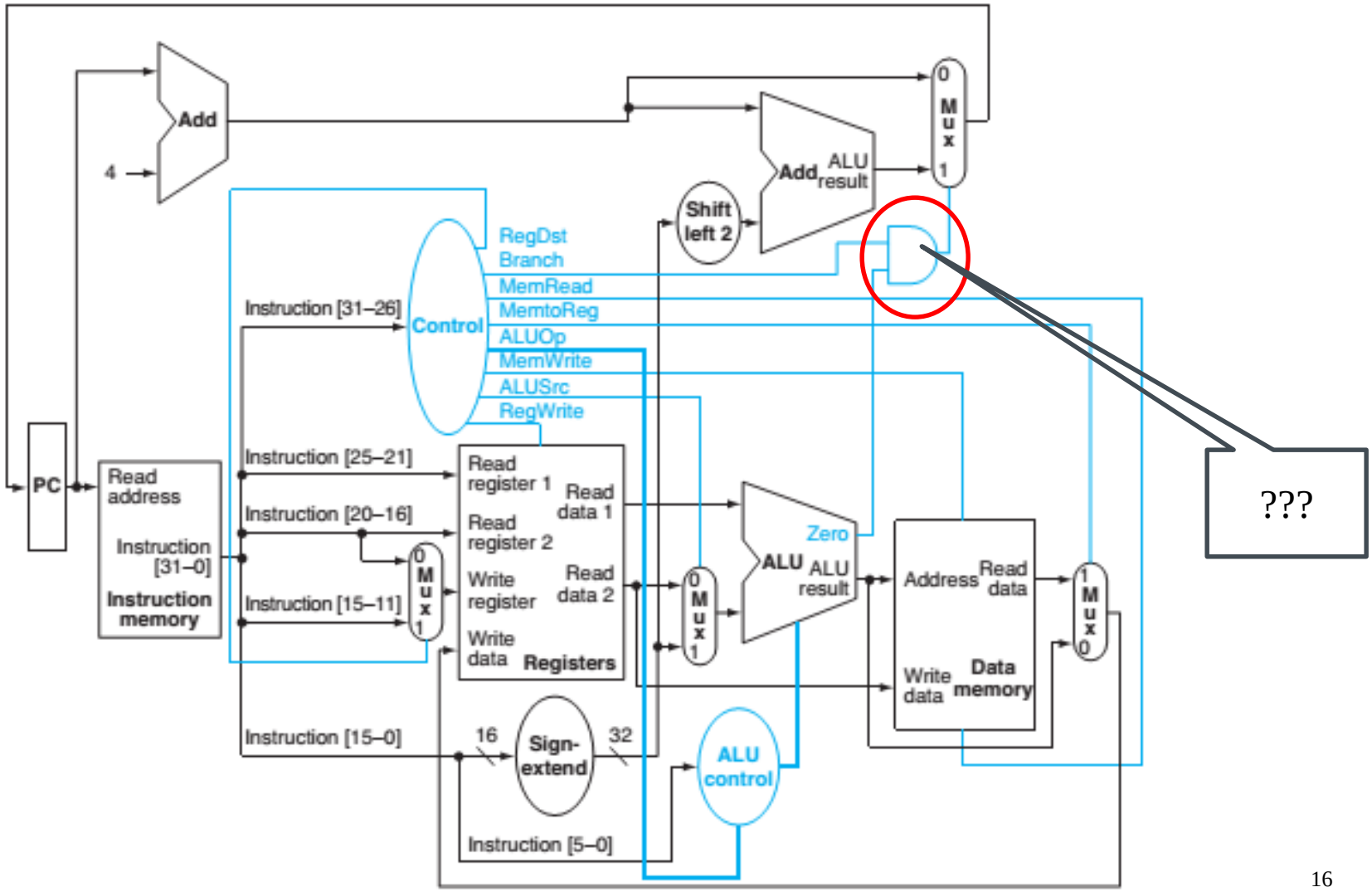
Bảng sự thật khối "Control":

Input or output	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Bảng sự thật khối "Control"

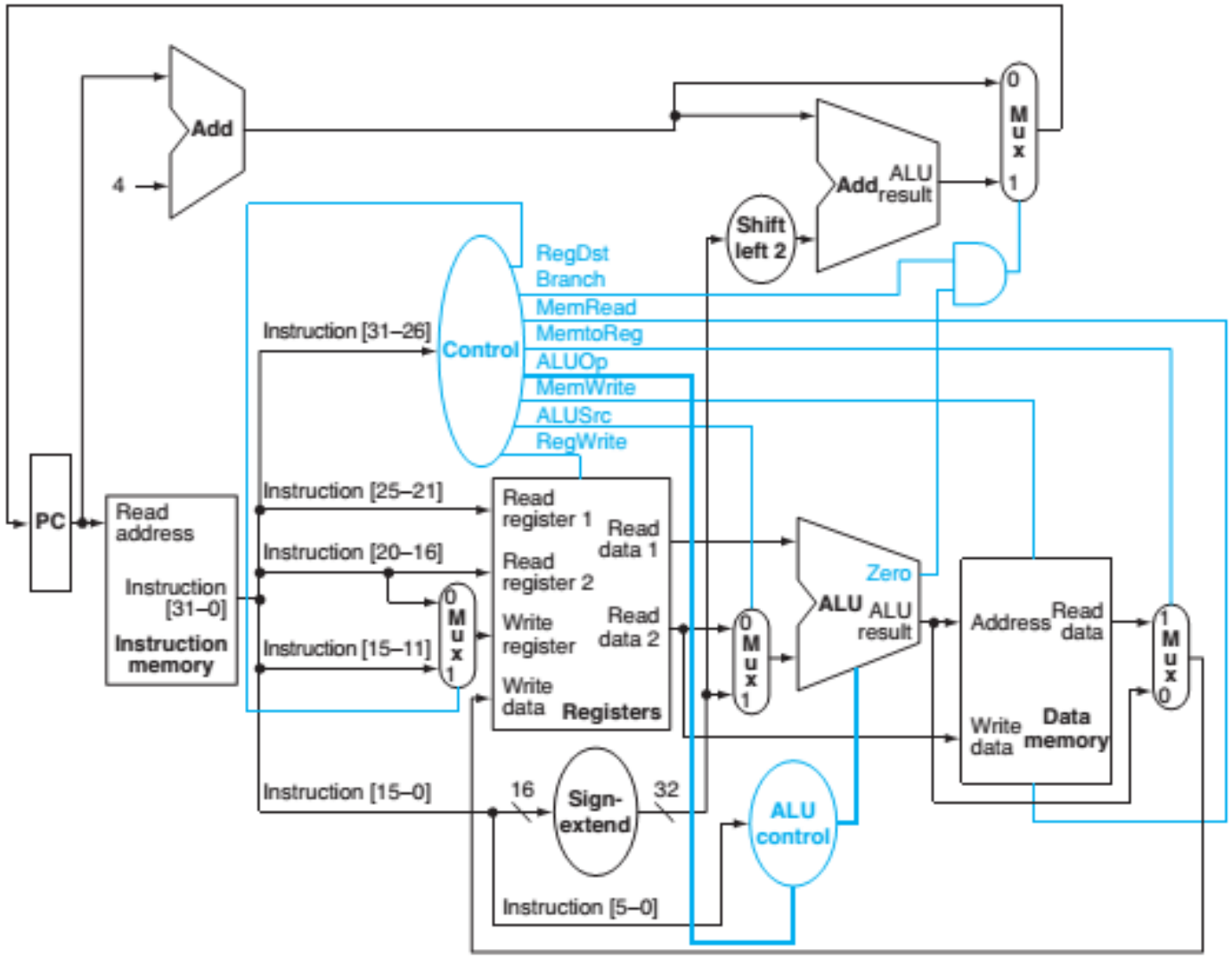
Hiện thực datapath

3. Khối điều khiển chính "Control"



Hiện thực datapath

3. Khối điều khiển chính "Control"



COMPUTER ENGINEERING

Hiện thực datapath

- **Hiện thực bộ xử lý đơn chu kỳ** (Single-cycle implementation hay single clock cycle implementation): là cách hiện thực sao cho bộ xử lý đáp ứng thực thi mỗi câu lệnh chỉ trong 1 chu kỳ xung clock ➔ đòi hỏi chu kỳ xung clock phải bằng thời gian của lệnh dài nhất.

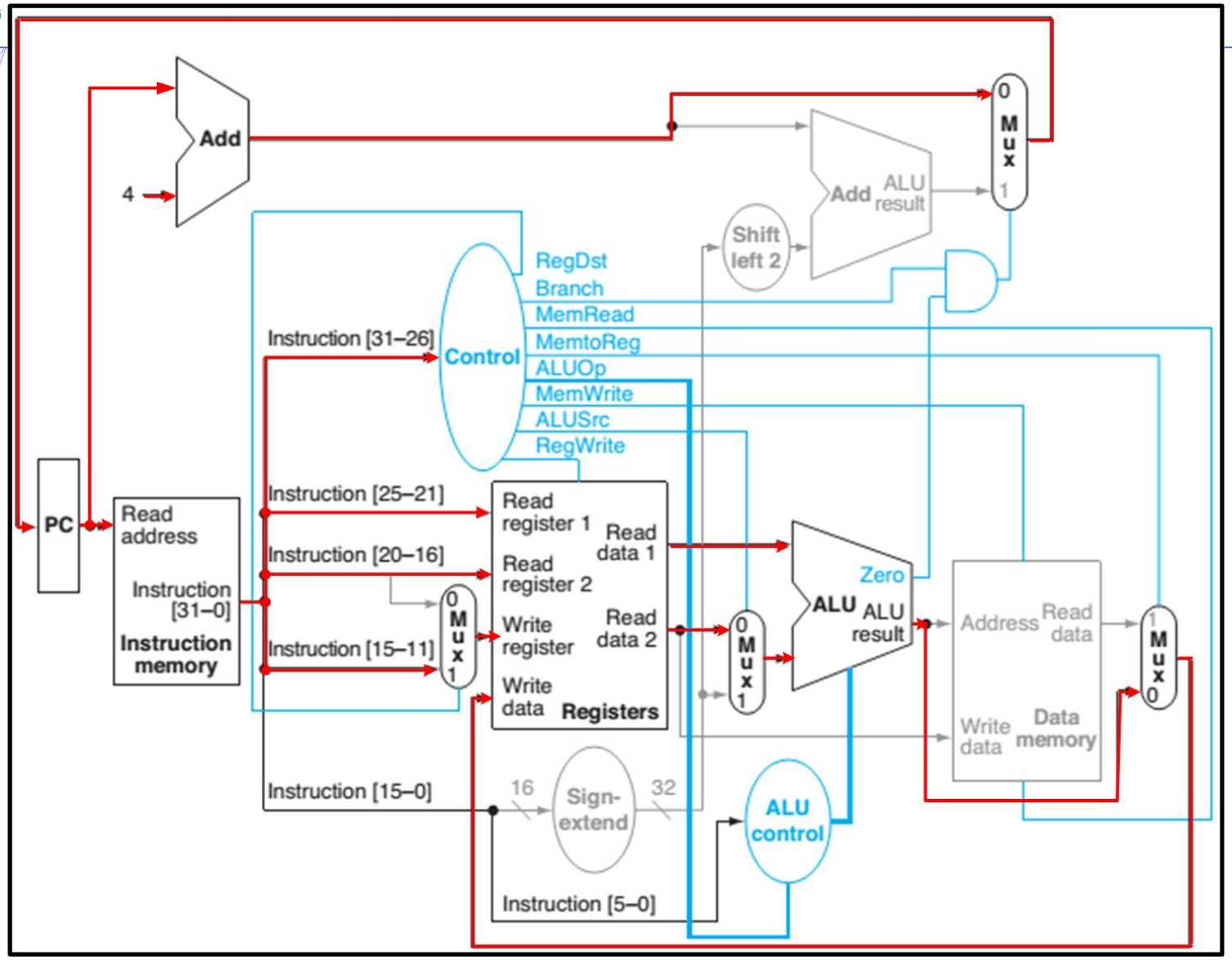
- Cách hiện thực bộ xử lý như đã trình bày trên là cách hiện thực **đơn chu kỳ**:

  Lệnh dài nhất là *lw*, gồm truy xuất vào "Instruction memory", "Registers", "ALU", "Data memory" và quay trở lại "Registers", trong khi các lệnh khác không đòi hỏi tất cả các công đoạn trên → chu kỳ xung clock thiết kế phải bằng thời gian thực thi lệnh *lw*.

- Mặc dù hiện thực bộ xử lý đơn chu kỳ có CPI = 1 nhưng hiệu suất rất kém, vì một chu kỳ xung clock quá dài, các lệnh ngắn đều phải thực thi cùng thời gian với lệnh dài nhất.
  Vì vậy, **Hiện thực đơn chu kỳ hiện tại không còn được sử dụng (hoặc chỉ có thể chấp nhận cho các tập lệnh nhỏ)**

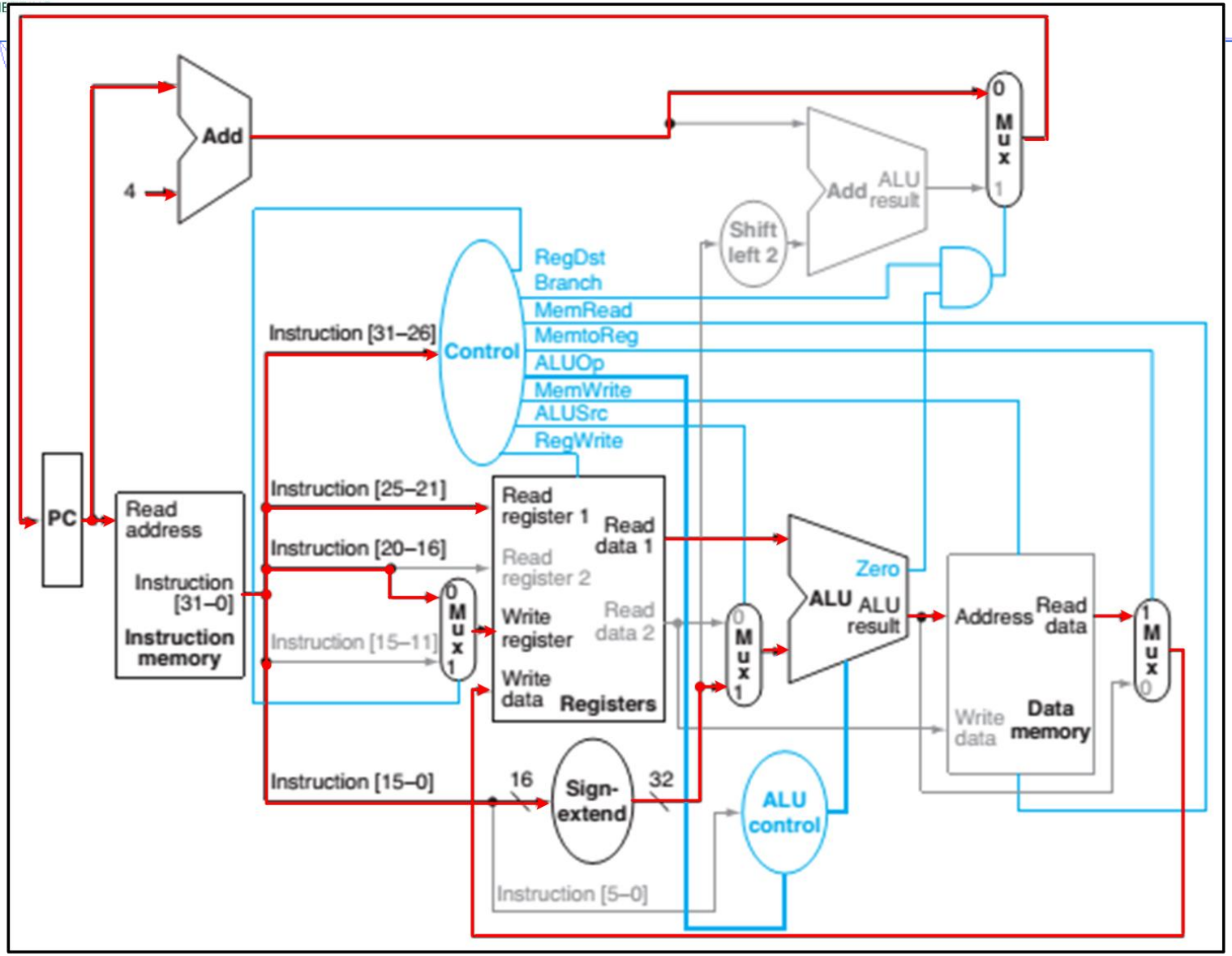
COMPUTER ENGINEERING

Xem lại Datapath với từng nhóm lệnh



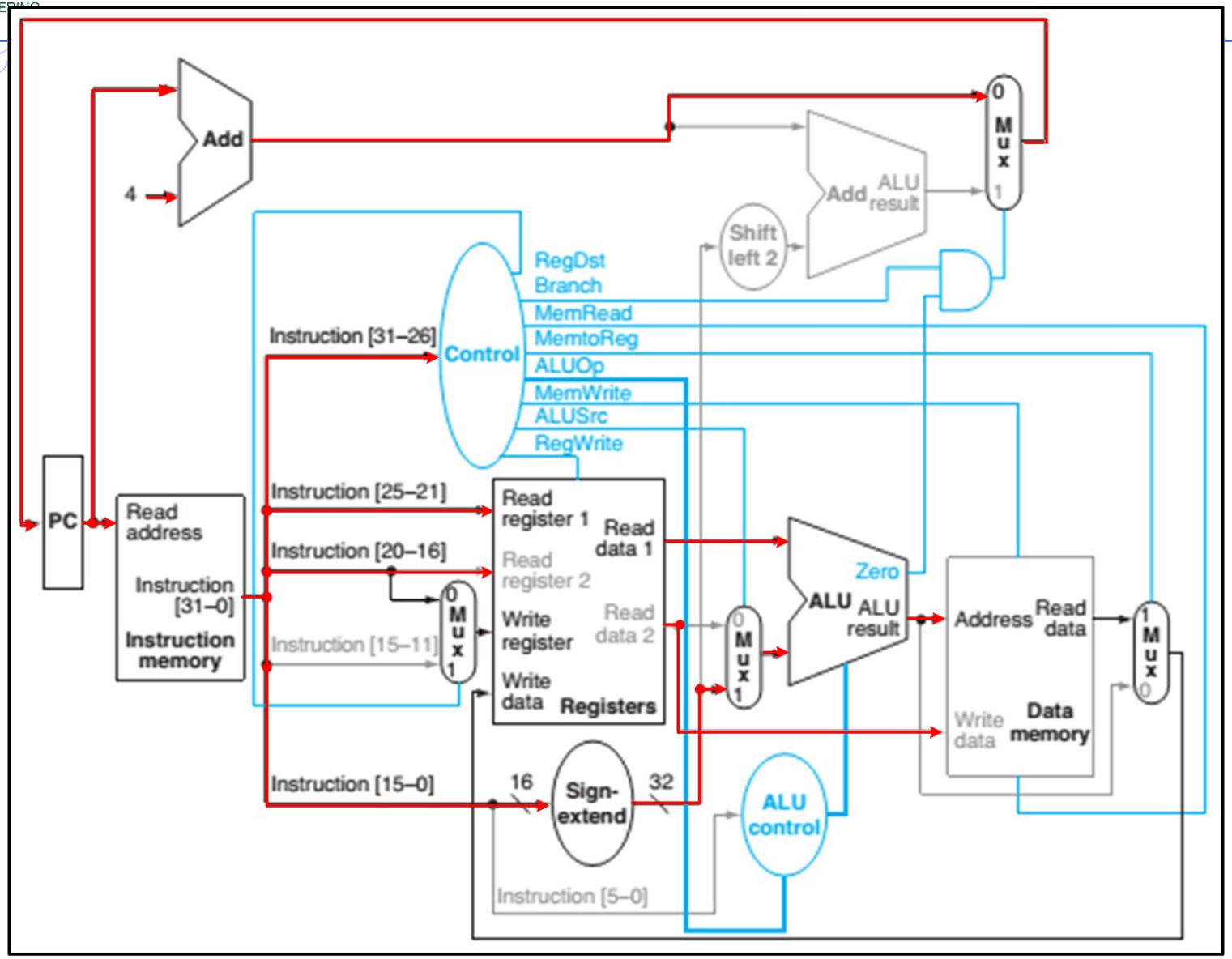
Các đường đỏ là các đường hoạt động khi lệnh thuộc ***nhóm logic và số học*** *thực thi*

Xem lại Datapath với từng nhóm lệnh



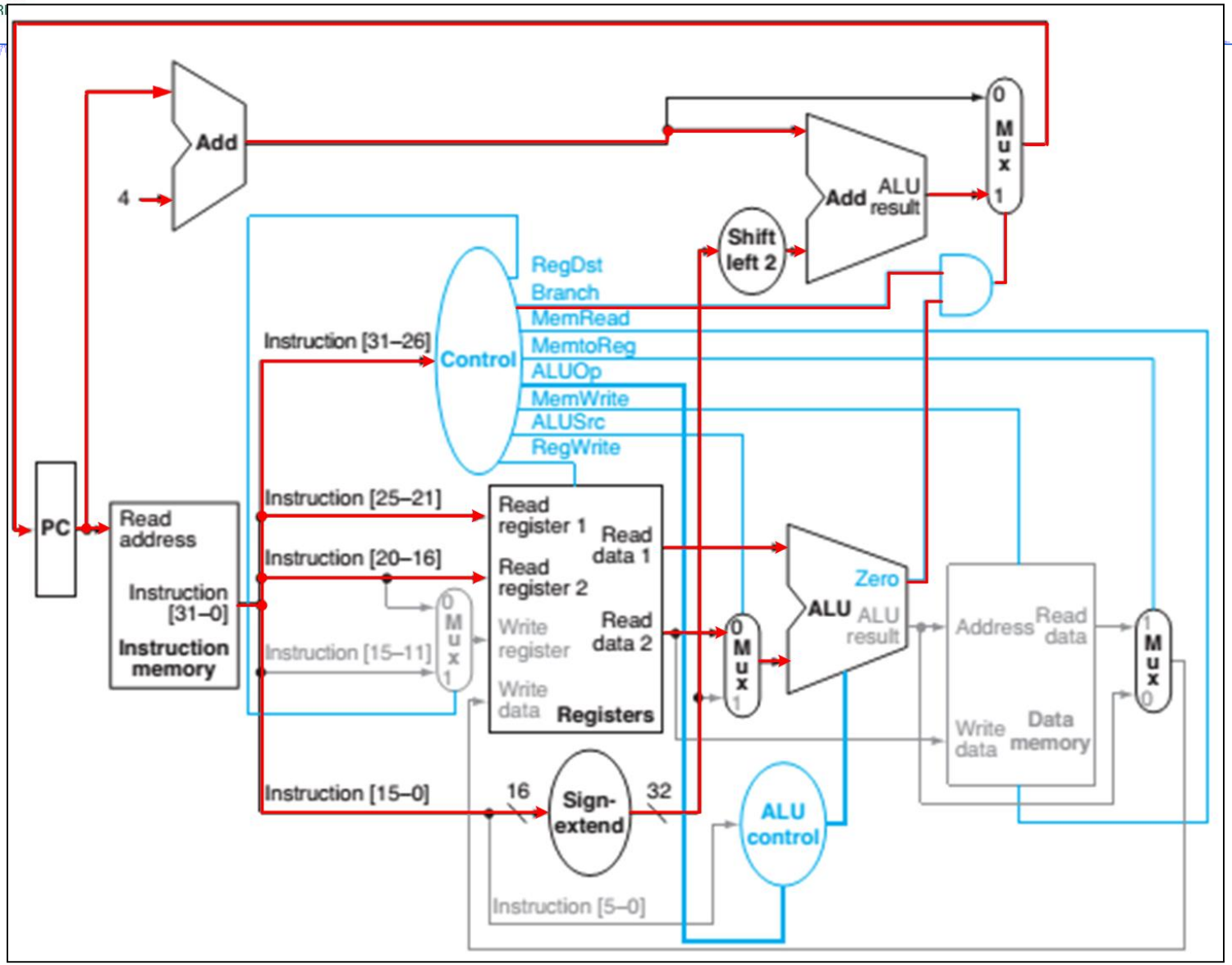
*Các đường đỏ là các đường hoạt động khi lệnh **lw** thực thi*

Xem lại Datapath với từng nhóm lệnh



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

Xem lại Datapath với từng nhóm lệnh



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

Tổng kết:

Hoàn chỉnh datapath cho 8 lệnh cơ bản:

- add, sub, and, or, slt
- lw, sw
- beq

Hiểu cách hiện thực các khối chức năng cơ bản trong datapath (khối Control, khối ALU Control)

BỘ XỬ LÝ

- **Lý thuyết: Đọc sách tham khảo**
  - Mục: 4.4
  - Sách: *Computer Organization and Design: The Hardware/Software Interface*, Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, Revised Fourth Edition, 2011.

- **Bài tập:** file đính kèm