



Khoa
CÔNG NGHỆ THÔNG TIN
ĐH Khoa học Tự nhiên TP HCM

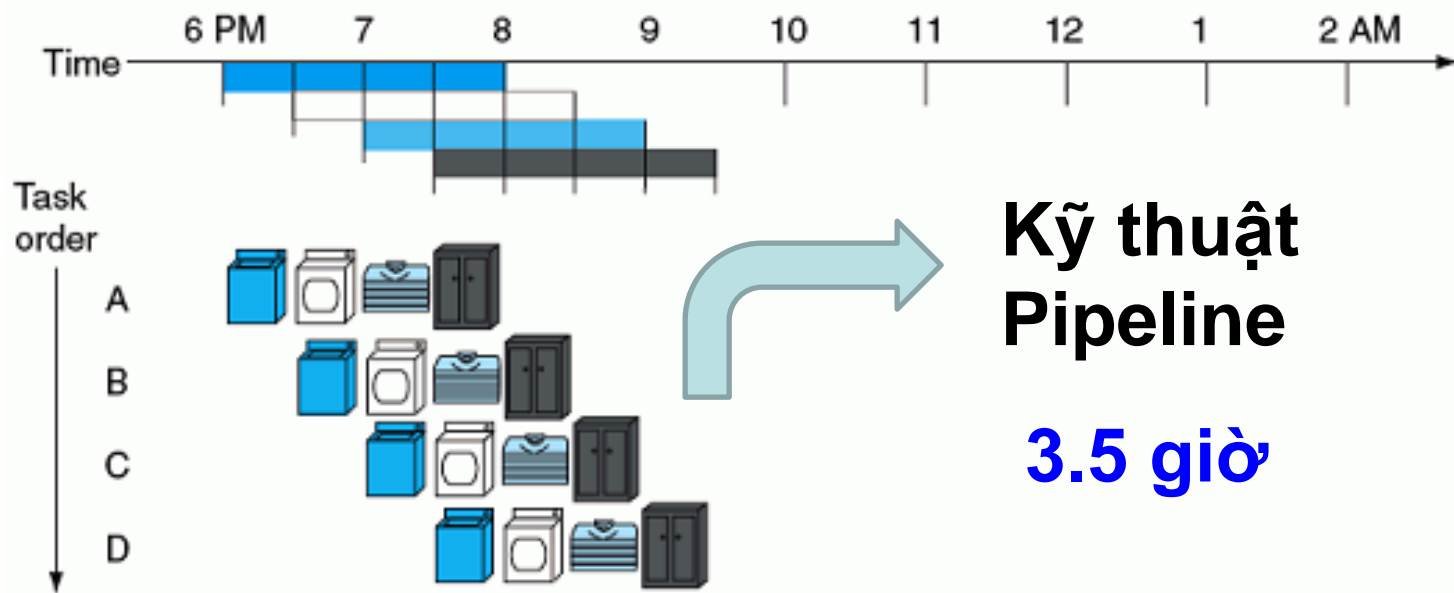
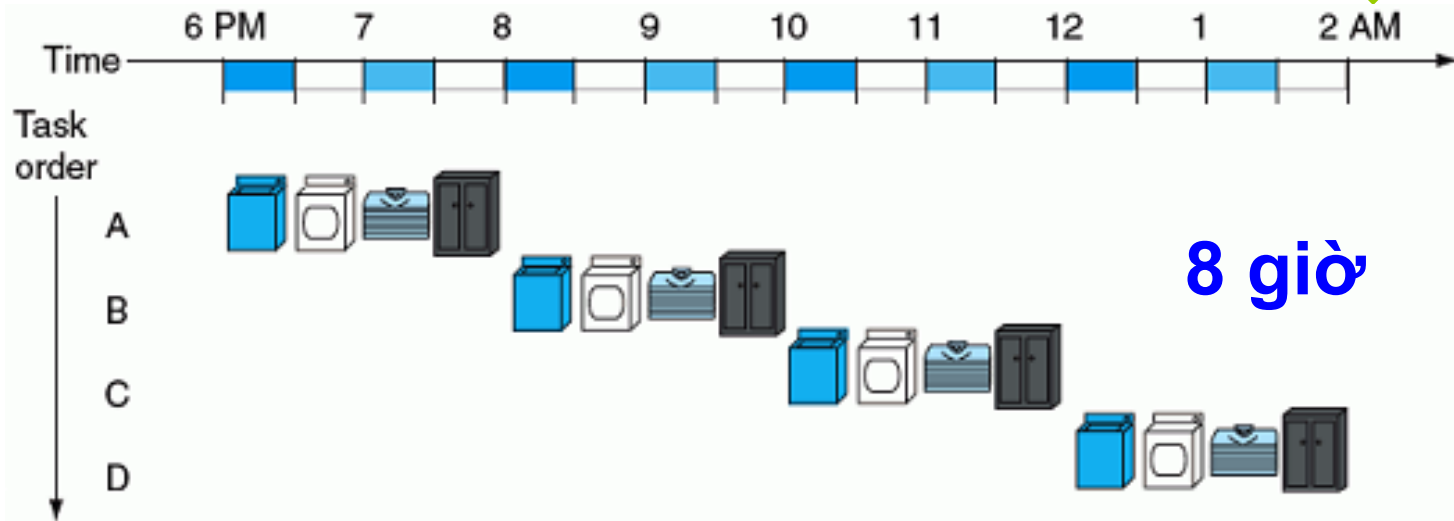
Bài 09: Kỹ thuật pipeline

Phạm Tuấn Sơn

ptson@fit.hcmus.edu.vn

Ý tưởng

1. Giặt
2. Sấy
3. Xếp
4. Cất tủ



Kỹ thuật pipeline

Thời gian

IFtch Dcd Exec Mem WB

IFtch Dcd Exec Mem WB

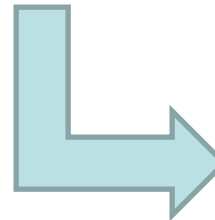
IFtch Dcd Exec Mem WB

IFtch Dcd Exec Mem WB

IFtch Dcd Exec Mem WB

IFtch Dcd Exec Mem WB

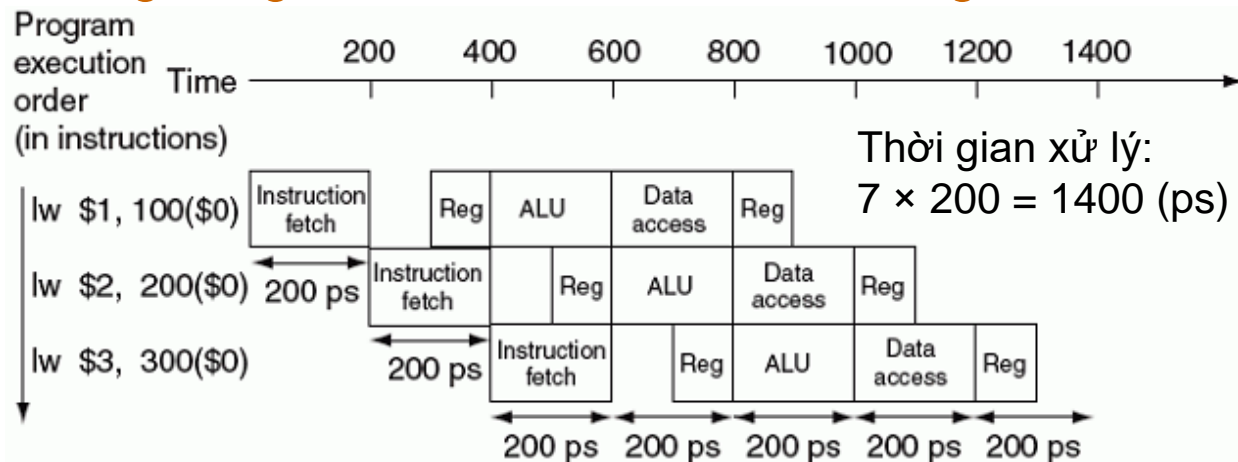
- 1) IFtch: Instruction Fetch, Increment PC
- 2) Dcd: Instruction Decode, Read Registers
- 3) Exec:
Mem-ref: Calculate Address
Arith-log: Perform Operation
- 4) Mem:
Load: Read Data from Memory
Store: Write Data to Memory
- 5) WB: Write Data Back to Register



**5-stage
Pipeline**

Hiệu năng của kỹ thuật pipeline

- Tăng tốc độ xử lý bằng cách tăng throughput của tất cả công việc chứ không giảm latency của một công việc
 - Latency: thời gian (độ trễ) hoàn thành một công việc
 - Throughput: khối lượng công việc có thể hoàn thành trong một khoảng thời gian



- Với chương trình có IC (instruction count) lệnh thì sẽ xử lý
 - Non-pipeline: $(IC \times 5) \times 200$ (ps)
 - Pipeline: $(4 + IC) \times 200$ (ps)
- Trường hợp IC lớn thì trong điều kiện lý tưởng sử dụng kỹ thuật pipeline sẽ nhanh gấp N lần so với không sử dụng kỹ thuật pipeline, với N là số công đoạn (stage)

Vấn đề của kỹ thuật Pipeline

- **Structural hazard**

- Khi các công đoạn của các lệnh được nạp chồng, trong một chu kỳ có thể sẽ xảy ra đụng độ việc sử dụng các thành phần mạch như: ALU, Mem, Register

- **Data hazard**

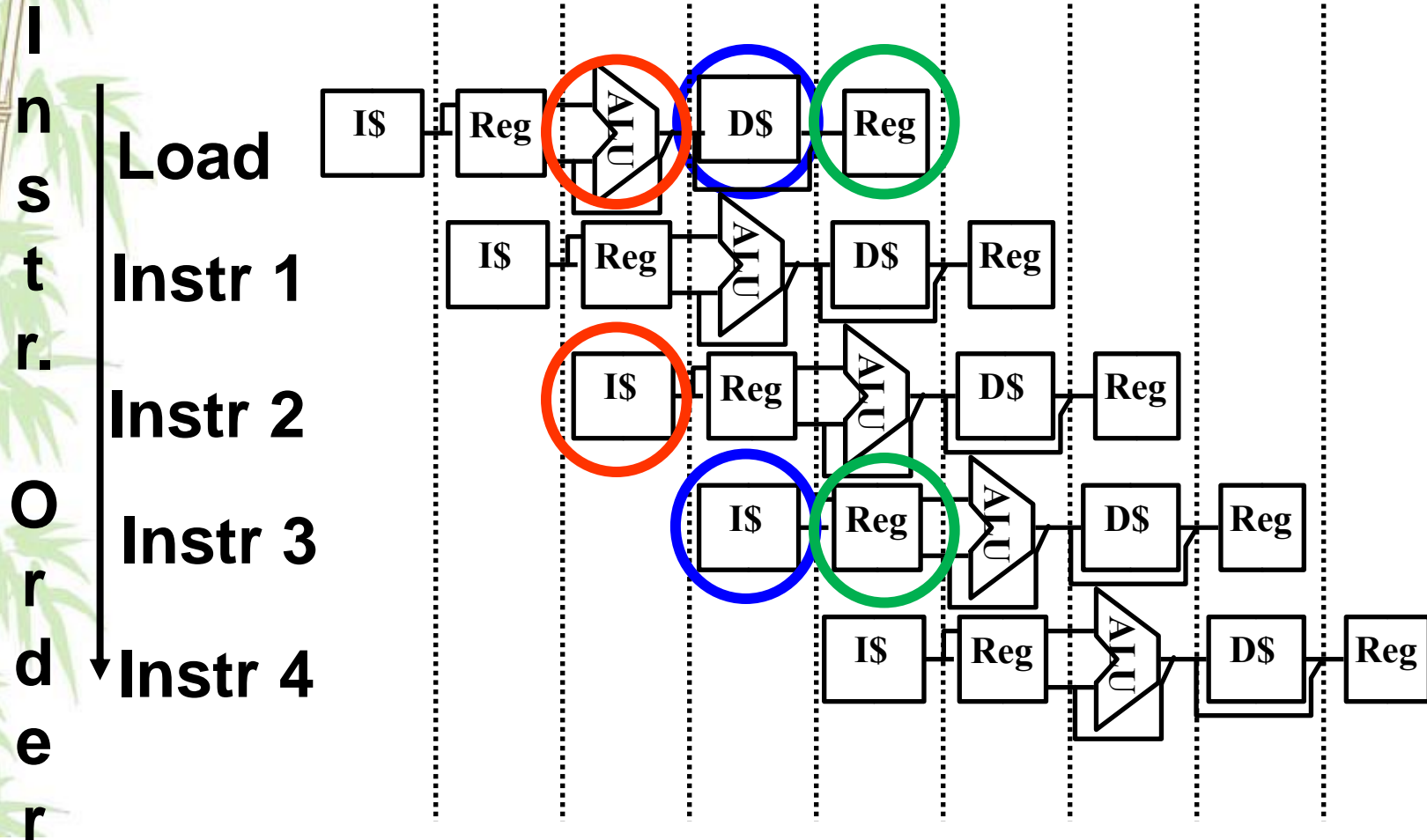
- Lệnh sau có thể phụ thuộc dữ liệu lệnh trước. Việc nạp chồng thực hiện có thể xảy ra trường hợp dữ liệu kết quả của lệnh trước chưa có mà lệnh sau đã cần đọc dữ liệu này

- **Control hazard**

- Khi đang thực hiện lệnh rẽ nhánh thì không thể biết lệnh tiếp theo sẽ được thực hiện là lệnh nào. Việc nạp chồng có thể sẽ không hiệu quả

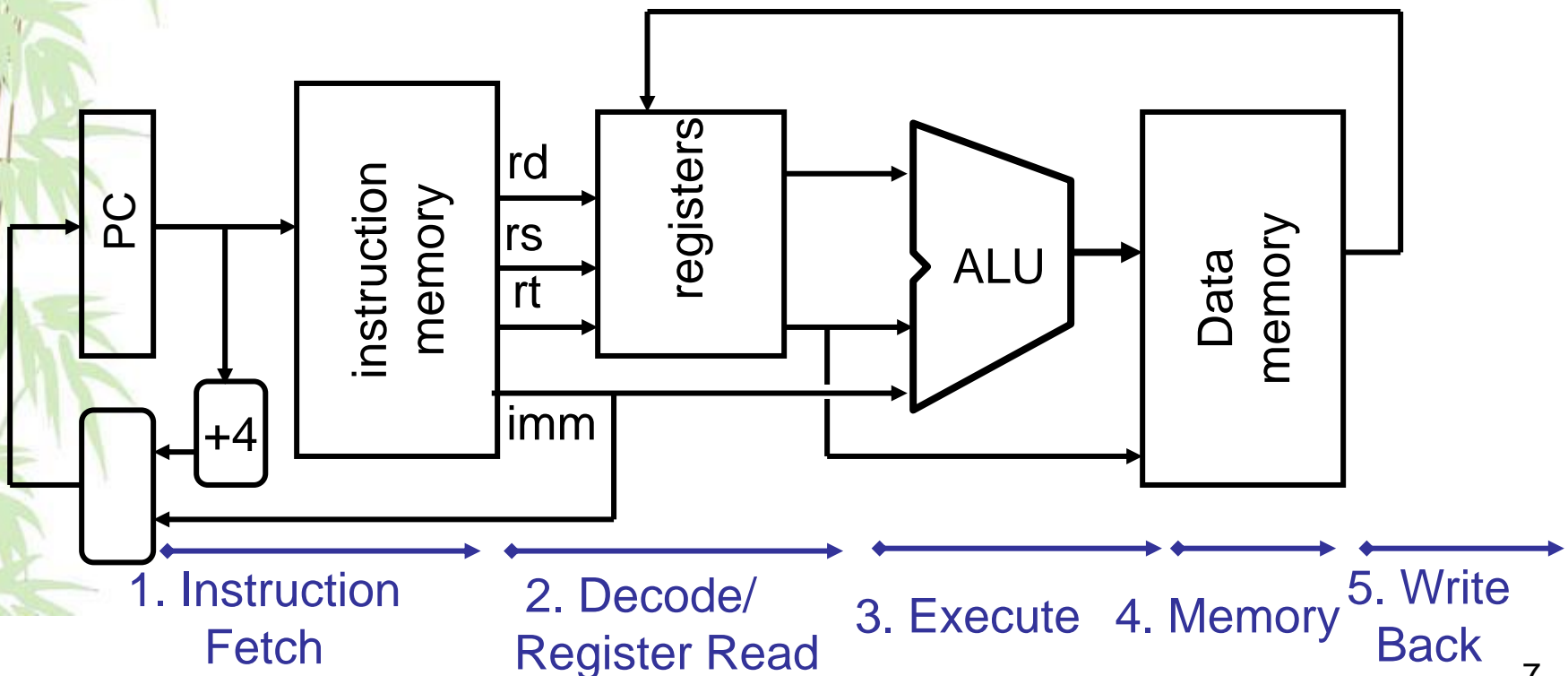
Structural Hazard

Time (clock cycles)



Structural Hazard – Giải pháp

- Ghi vào thanh ghi nửa chu kỳ đầu; đọc thanh ghi nửa chu kỳ cuối
- Tách ALU và Adder
- Tách IMem và DMem (Cache)



Data Hazard (1/2)

- Xem xét luồng xử lý các lệnh sau

add \$1, \$2, \$3

sub \$5, \$1, \$4

and \$7, \$1, \$6

or \$9, \$t1, \$t8

lw \$10, 0(\$1)

xor \$12, \$10, \$11

Data Hazard (2/2)

Sơ đồ xử lý

Time (clock cycles)

add \$1, \$2, \$3

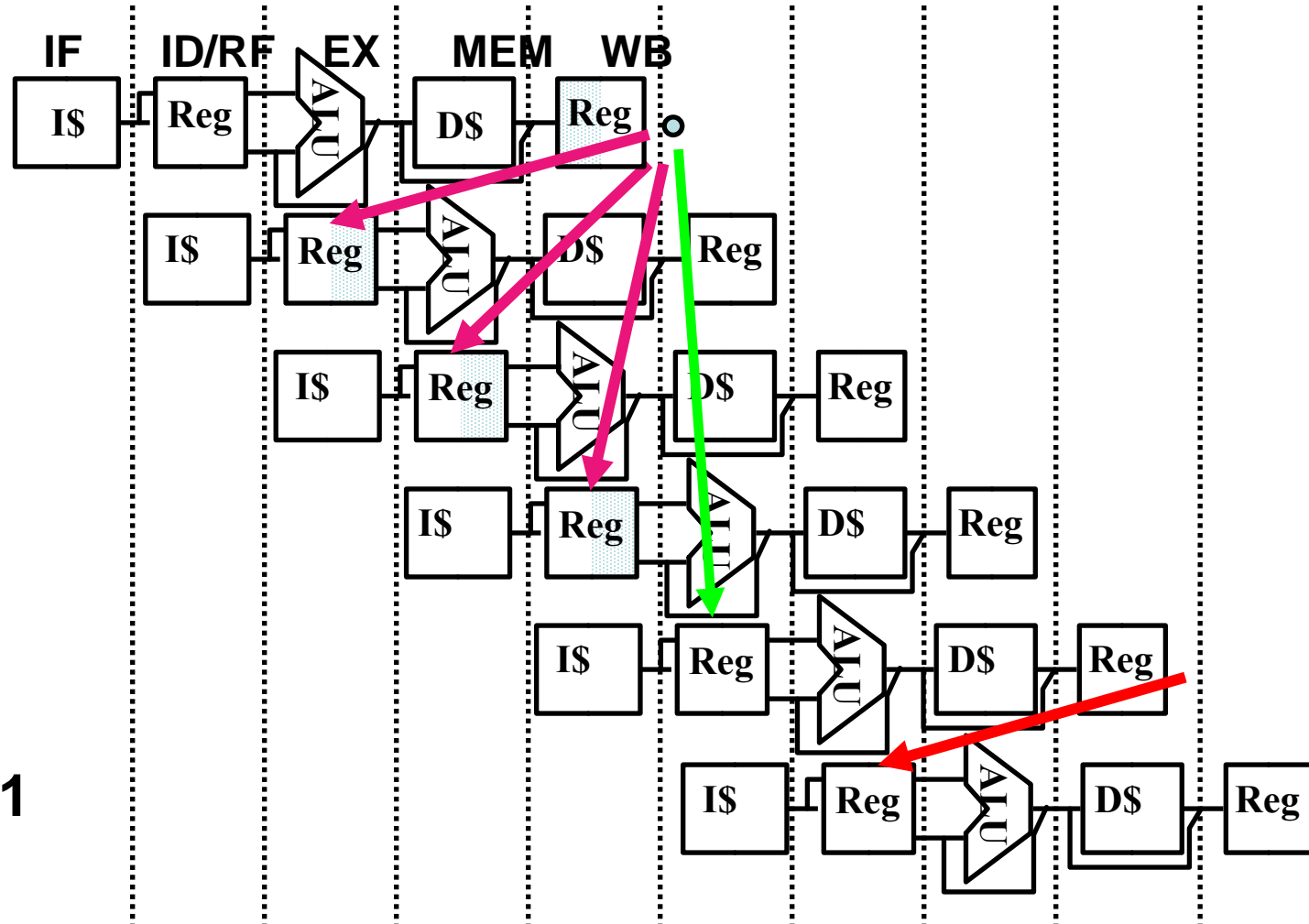
sub \$5, \$1, \$4

and \$7, \$1, \$6

or \$9, \$1, \$8

lw \$10, 0(\$1)

xor \$12, \$10, \$11



Data Hazard – Giải pháp (1/3)

• Forwarding

add \$1, \$2, \$3

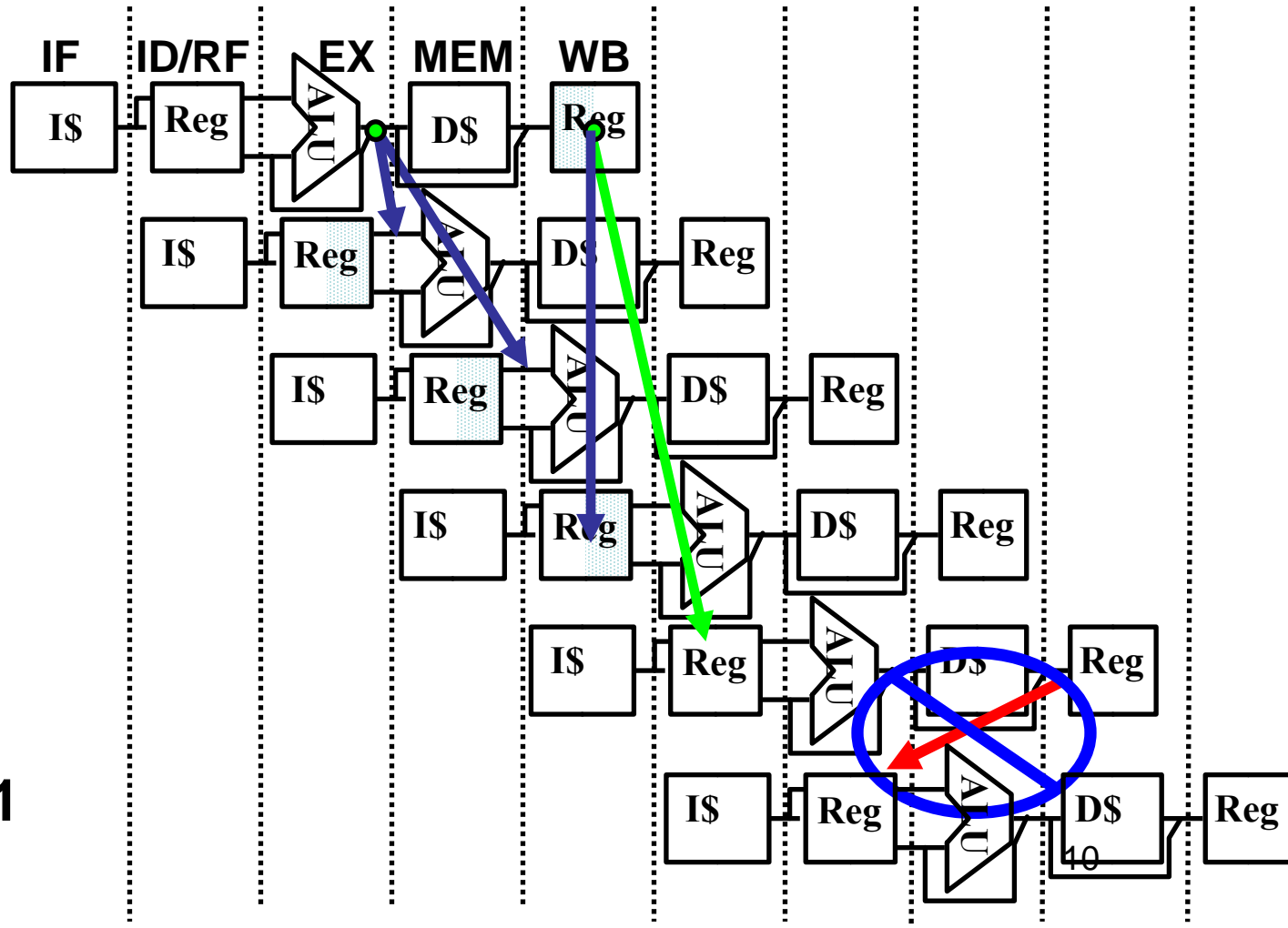
sub \$5, \$1, \$4

and \$7, \$1, \$6

or \$9, \$1, \$8

lw \$10, 0(\$1)

xor \$12, \$10, \$11

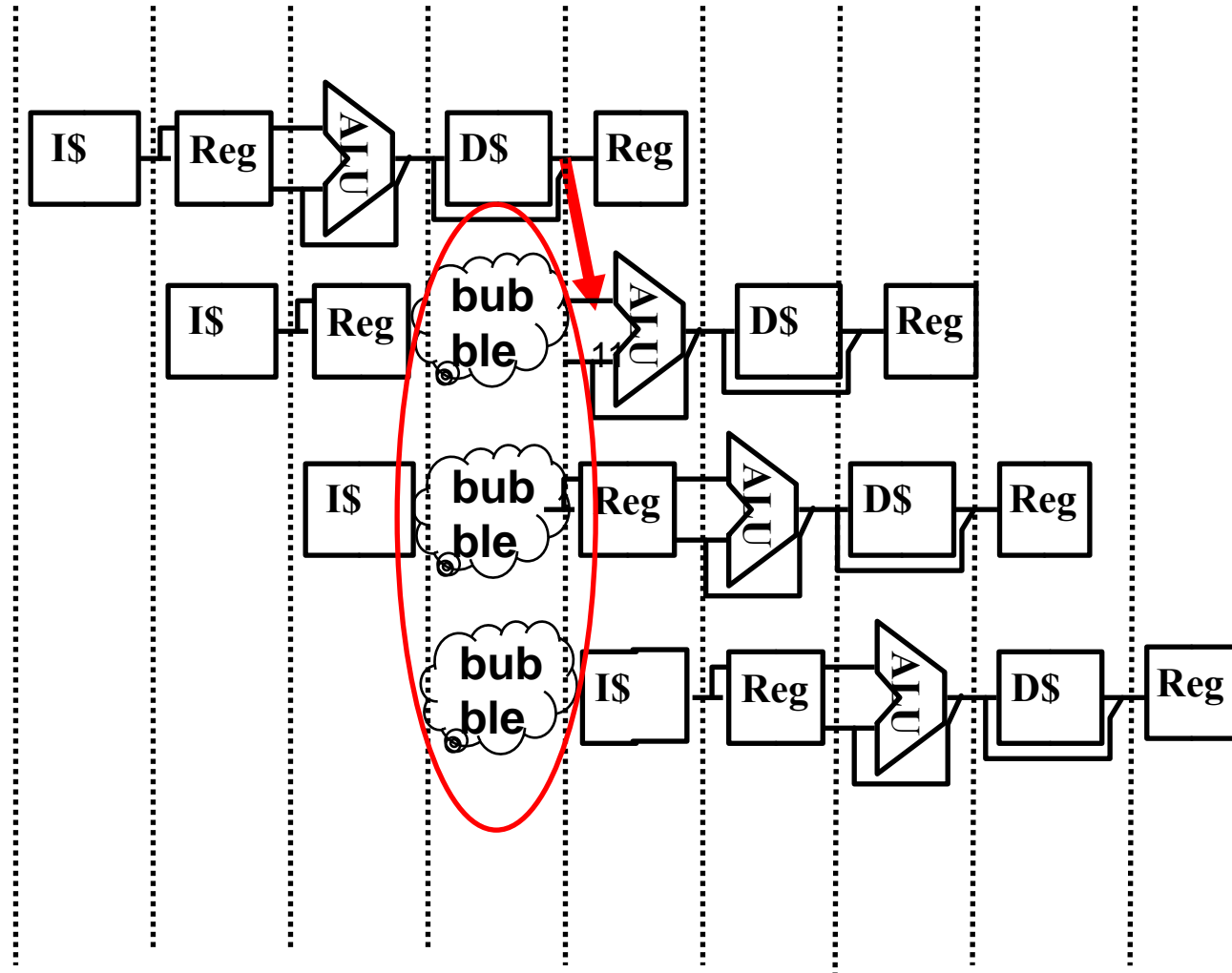


Data Hazard – Giải pháp (2/3)

• Stall

lw \$10,0(\$1)

xor \$12,\$10,\$11



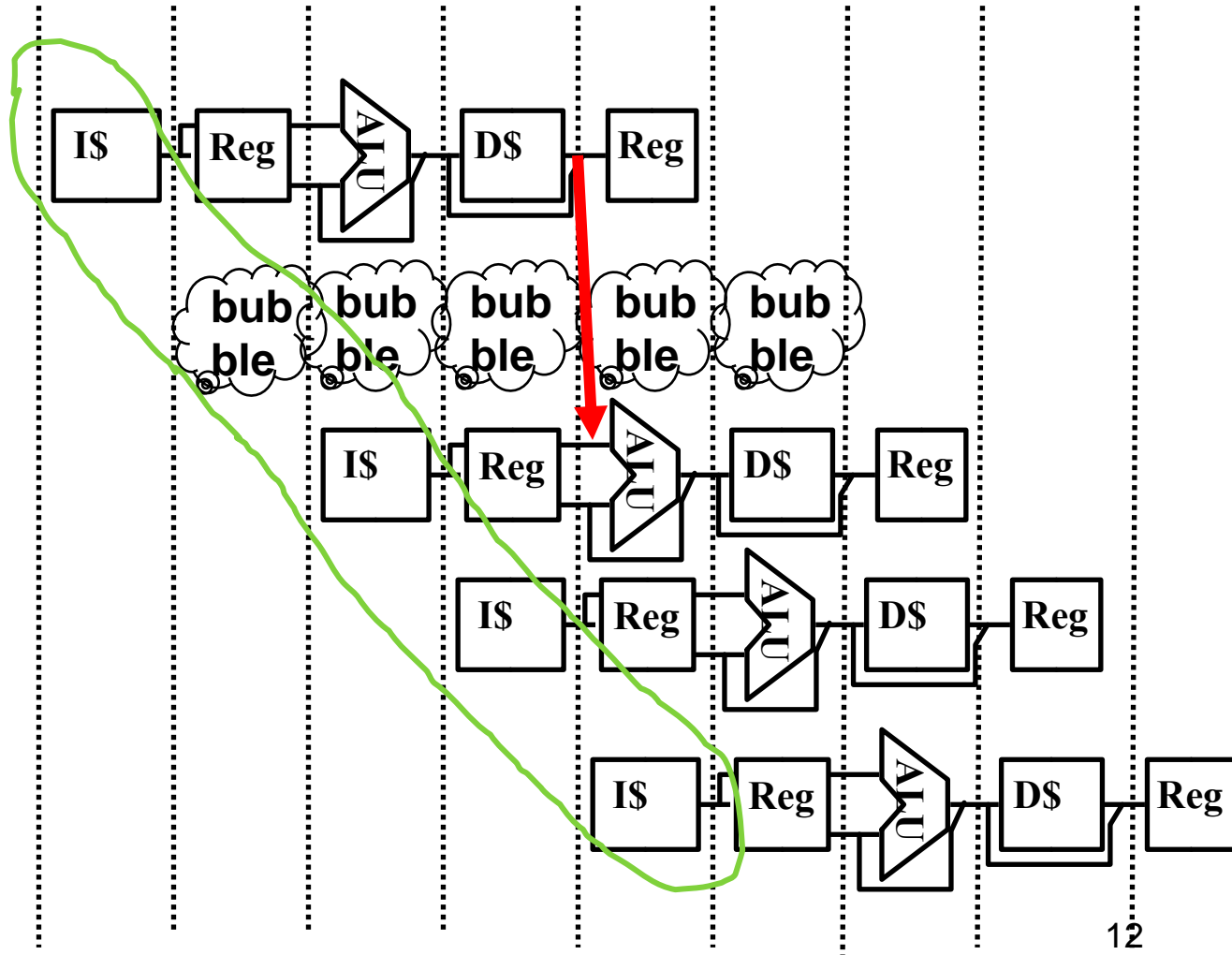
Data Hazard – Giải pháp (3/3)

- Nop (~ stall)

lw \$10,0(\$1)

nop

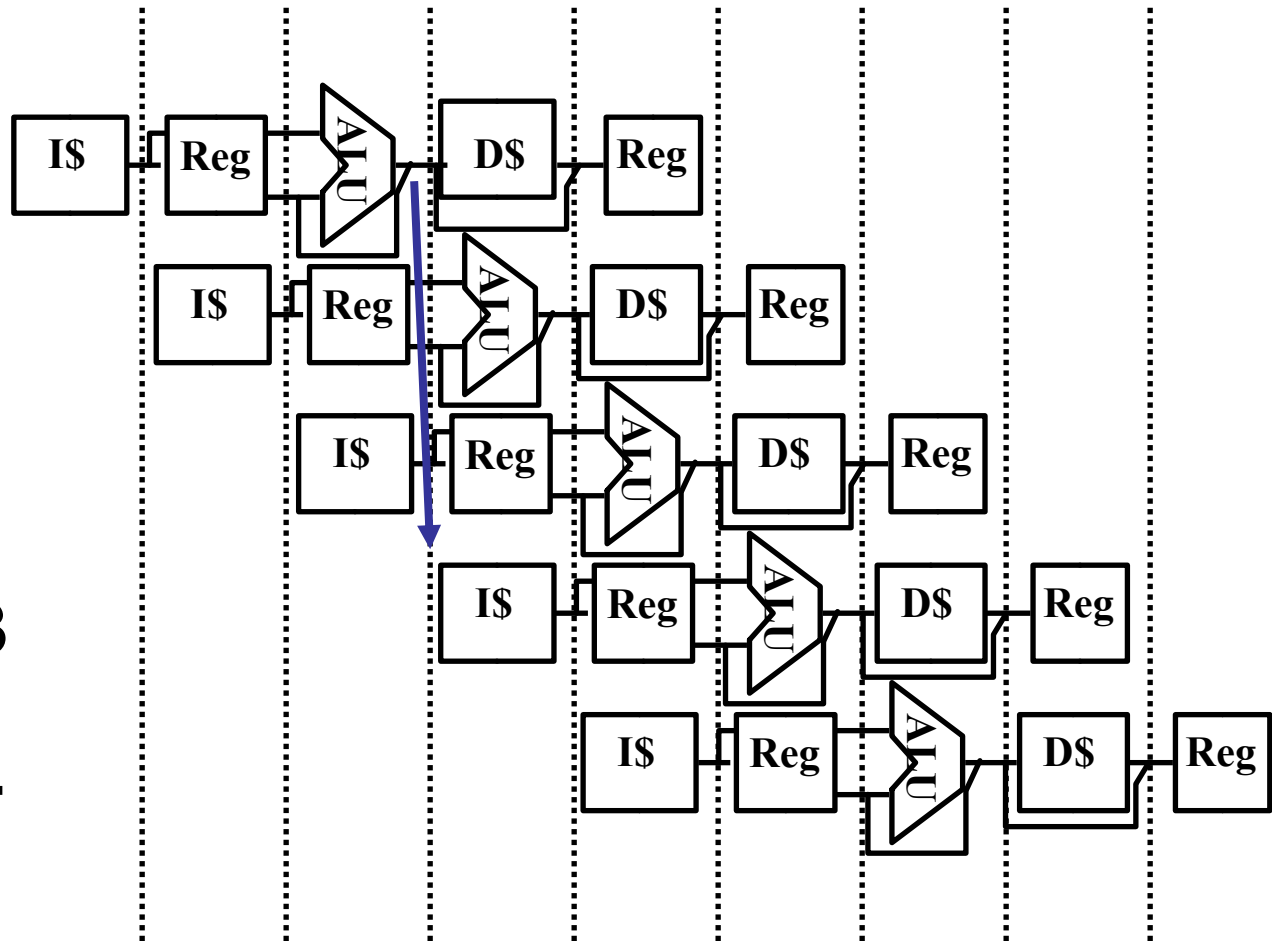
xor \$12,\$10,\$11



Control Hazard

Time (clock cycles)

Instruction Order
↓
beq
Instr 1
Instr 2
Instr 3
Instr 4



Nạp chồng lệnh nào (Instr1 hay Instr3) ?

Nếu nạp sai phải hủy tất cả các lệnh đang được xử lý !!!

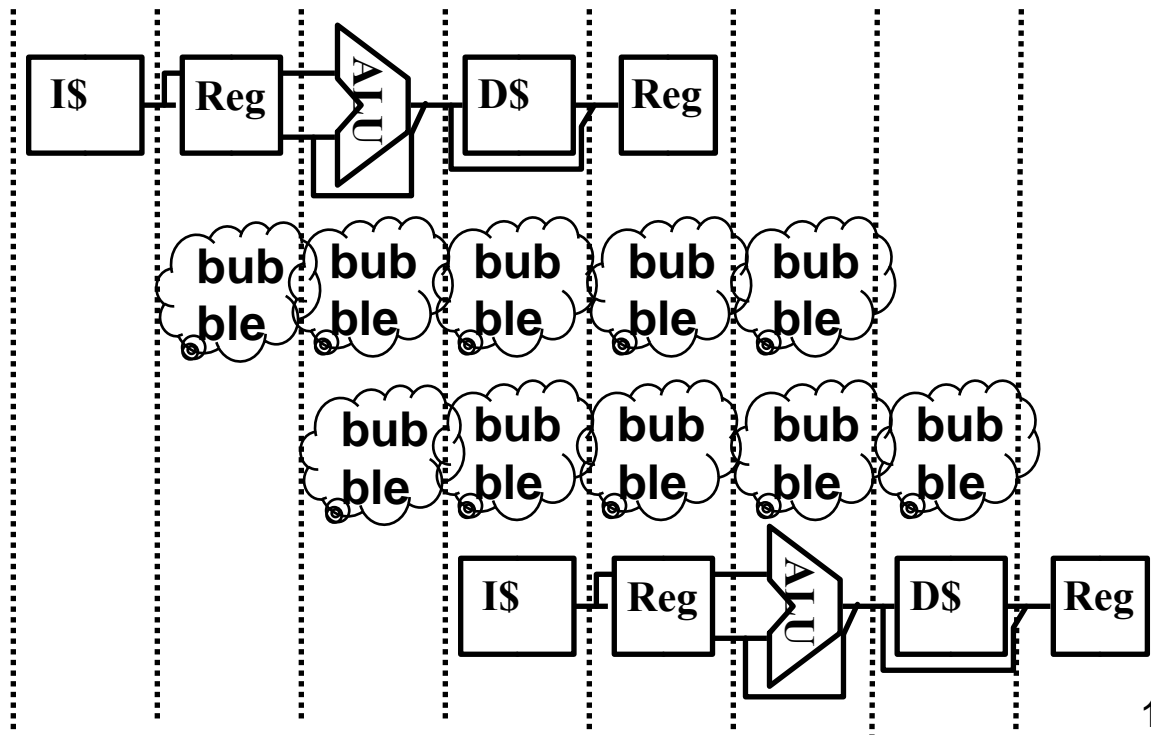
Control Hazard – Giải pháp (1/4)

- Stall: không nạp chồng cho đến khi xác định được lệnh nào sẽ được xử lý kế tiếp (chu kỳ 3 – công đoạn ALU của lệnh rẽ nhánh) → phải stall 2 chu kỳ
- Nhược điểm: không tận dụng được hiệu quả nạp chồng của kỹ thuật pipeline

beq \$1,\$2,NHAN

nop

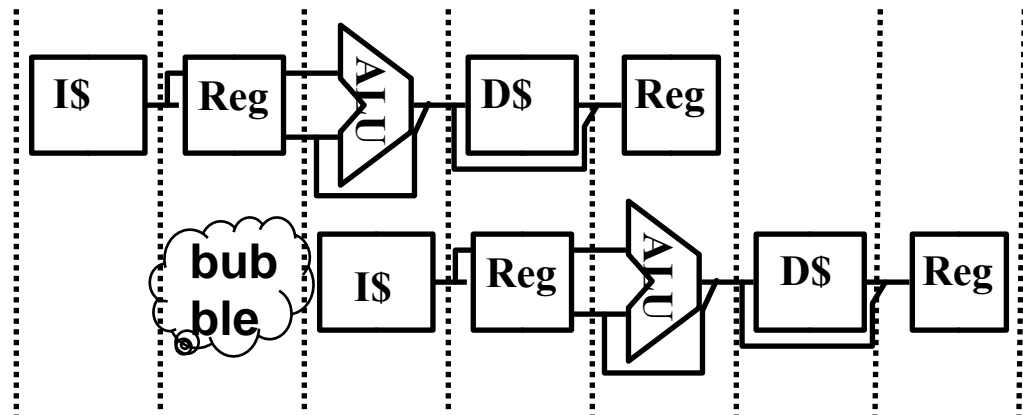
nop



Control Hazard – Giải pháp (2/4)

- Chuyển thao tác so sánh lên công đoạn giải mã lệnh → chỉ cần stall 1 chu kỳ

beq \$1,\$2,NHAN



Control Hazard – Giải pháp (2/4)

- Delayed Branch

– Sắp xếp các lệnh sao cho có thể trì hoãn việc rẽ nhánh

Nondelayed Branch

or \$8, \$9, \$10

add \$1, \$2, \$3

sub \$4, \$5, \$6

beq \$1, \$4, Exit

xor \$10, \$1, \$11

Exit:

Delayed Branch

add \$1, \$2, \$3

sub \$4, \$5, \$6

beq \$1, \$4, Exit

or \$8, \$9, \$10

xor \$10, \$1, \$11

Exit:

Control Hazard – Giải pháp (2/4)

- Branch Prediction: đoán trước là sẽ rẽ nhánh hay không
 - Tĩnh
 - Branch Not Taken: luôn đoán trước là không rẽ nhánh (MIPS 47%)
 - Branch Taken: luôn đoán trước là không rẽ nhánh (MIPS 53%)
 - Động: lưu lại các thông tin rẽ nhánh trong quá khứ để quyết định
 - Branch History Table
 - Correlated Branch Prediction
 - Tournament Predictors
 - Branch Target Buffer

- Thiết kế chi tiết datapath và control của bộ xử lý theo kỹ thuật pipeline
→ môn KTMT nâng cao
- Các kỹ thuật hiện đại nâng cao hiệu năng CPU
 - Instruction-Level Parallelism
 - Thread-Level Parallelism
 - Multi-processor
→ môn KTMT nâng cao

Tham khảo

- Phần 6.1, P&H

