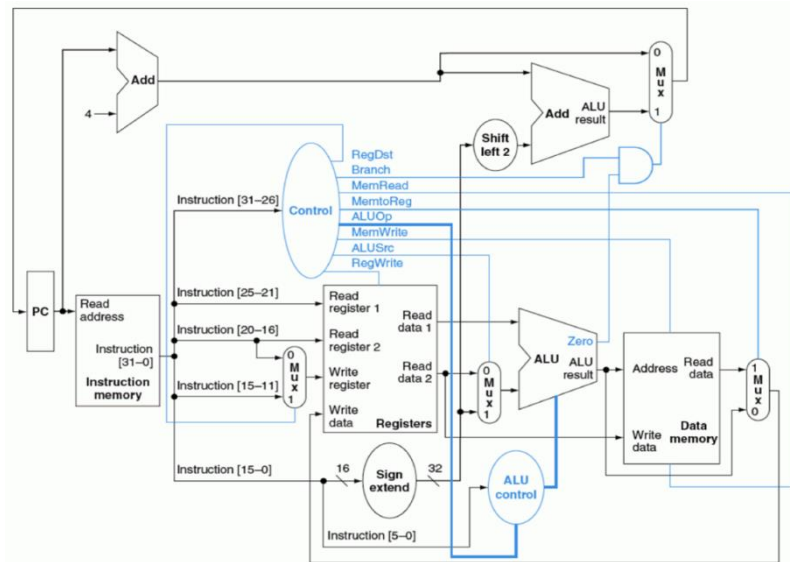


## BÀI TẬP CHƯƠNG 4: BỘ XỬ LÝ PROCESSOR

### DẠNG 1: HIỂU VỀ ĐƯỜNG ĐI CỦA DATAPATH



Hình 4.1

**Bài tập 1 [4.1/357 - Computer Organization and Design]:** Xem xét lệnh sau đây:

**Lệnh:** AND Rd, Rs, Rt

**Diễn giải:**  $\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rs}] \text{ AND } \text{Reg}[\text{Rt}]$

**1.1** Giá trị của các tín hiệu điều khiển được tạo ra bởi bộ điều khiển trong Hình 4.1 cho lệnh trên là gì?

**1.2** Những tài nguyên (khối) nào thực hiện chức năng hữu ích cho lệnh này?

**1.3** Những tài nguyên (khối) nào tạo ra đầu ra nhưng đầu ra của chúng không được sử dụng cho lệnh này? Những tài nguyên nào không tạo ra đầu ra cho lệnh này?

### Bài tập 2

**Instruction:** SUB Rd, Rs, Rt

**Interpretation:**  $\text{Reg}[\text{Rd}] = \text{Reg}[\text{Rs}] - \text{Reg}[\text{Rt}]$

**2.1** Giá trị của các tín hiệu điều khiển được tạo ra bởi phần điều khiển trong Hình 4.1 đối với lệnh trên là gì?

**2.2** Những tài nguyên (blocks) nào thực hiện chức năng hữu ích cho lệnh này?

**2.3** Những tài nguyên nào tạo ra đầu ra nhưng đầu ra của chúng không được sử dụng cho lệnh này? Những tài nguyên nào không tạo ra đầu ra cho lệnh này?

### Bài tập 3

**Instruction:** `LW Rt, offset(Rs)`

**Interpretation:** `Reg[Rt] = Memory[Reg[Rs] + offset]`

**3.1** Giá trị của các tín hiệu điều khiển được tạo bởi phần điều khiển trong Hình 4.1 đối với lệnh trên là gì?

**3.2** Xác định tất cả các khối datapath được sử dụng và mô tả cách dữ liệu di chuyển qua chúng để thực hiện lệnh này.

**3.3** Những tài nguyên nào tạo ra đầu ra nhưng đầu ra của chúng không được sử dụng cho lệnh này? Những tài nguyên nào không tạo ra đầu ra cho lệnh này?

### Bài tập 4

**Instruction:** `BEQ Rs, Rt, offset`

**Interpretation:** `if (Reg[Rs] == Reg[Rt]) PC = PC + 4 + (offset << 2)`

**4.1** Các tín hiệu điều khiển được kích hoạt để thực hiện lệnh này là gì?

**4.2** Mô tả cách xác định địa chỉ tiếp theo của PC trong lệnh này.

**4.3** Những khối nào trong datapath được sử dụng nhưng đầu ra của chúng không được dùng để thay đổi PC?

### Bài tập 5

**Instruction:** `SW Rt, offset(Rs)`

**Interpretation:** `Memory[Reg[Rs] + offset] = Reg[Rt]`

**5.1** Giá trị của các tín hiệu điều khiển cần thiết cho lệnh này là gì?

**5.2** Đường đi của dữ liệu từ thanh ghi Rs và Rt đến bộ nhớ là gì?

**5.3** Giá trị đưa vào ALU chính là gì?

**5.4** Những khối datapath nào không tham gia hoặc không cần thiết trong lệnh này?

**Bài tập 6: Cho bảng giá trị của các đường của control như sau:**

| Ins | RegDest | ALUSrc | MemToReg | RegWrite | MemRead | MemWrite | Branch | ALUOp0 | ALUOp1 |
|-----|---------|--------|----------|----------|---------|----------|--------|--------|--------|
| R   | 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      |

Trong hệ thống đơn chu kỳ, giả sử có 1 đường tín hiệu bị lỗi (luôn ở mức 0) thì những lệnh gì sẽ hoạt động sai? Giải thích. Giả sử mỗi câu sau là độc lập.

- a) RegWrite = 0                      b) ALUOp0 = 0                      c) ALUOp1 = 0  
d) Branch = 0                      e) MemRead = 0                      f) MemWrite = 0

**DẠNG 2: BÀI TOÁN KẾT HỢP (HIỂU VÀ TÍNH CYCLE TIME)**

**Bài tập 1 [4.3/358 - Computer Organization and Design]:** Khi các nhà thiết kế vi xử lý cân nhắc một cải tiến tiềm năng đối với đường đi dữ liệu (datapath) của vi xử lý, quyết định thường phụ thuộc vào sự đánh đổi giữa chi phí và hiệu năng. Trong ba vấn đề sau, giả sử chúng ta bắt đầu với một đường đi dữ liệu từ Hình 4.1, nơi mà các khối I-Mem, Add, Mux, ALU, Regs, D-Mem, và Control có độ trễ lần lượt là 400 ps, 100 ps, 30 ps, 120 ps, 200 ps, 350 ps, và 100 ps, và chi phí lần lượt là 1000, 30, 10, 100, 200, 2000, và 500.

Xem xét việc thêm một khối nhân (multiplier) vào ALU. Sự bổ sung này sẽ làm tăng độ trễ của ALU lên 300 ps và tăng chi phí của ALU lên 600. Kết quả là sẽ giảm 5% số lệnh được thực thi, vì chúng ta không còn cần phải mô phỏng lệnh MUL.

**1.1** Thời gian của chu kỳ xung nhịp (clock cycle time) trước và sau khi áp dụng cải tiến này là bao nhiêu?

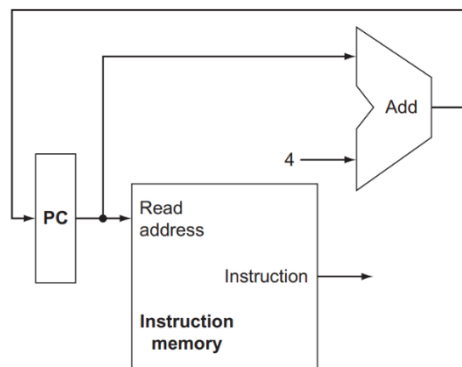
**1.2** Tốc độ cải thiện (speedup) đạt được nhờ cải tiến này là bao nhiêu?

**1.3** So sánh tỷ lệ chi phí/hiệu năng (cost/performance ratio) trước và sau khi áp dụng cải tiến này.

**Bài tập 2 [4.4/358 - Computer Organization and Design]:** Các bài tập trong phần này giả định rằng các khối logic cần thiết để triển khai đường đi dữ liệu (datapath) của bộ xử lý có độ trễ như sau:

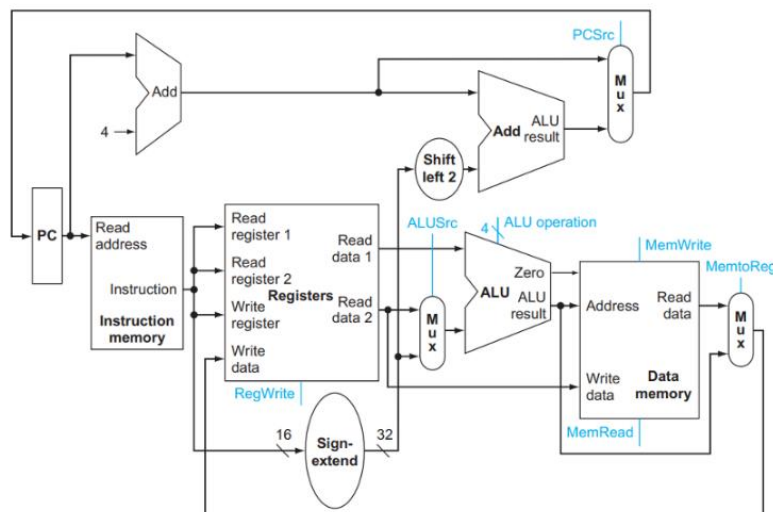
| I-Mem  | Add   | Mux   | ALU   | Regs  | D-Mem  | Sign-Extend | Shift-Left-2 |
|--------|-------|-------|-------|-------|--------|-------------|--------------|
| 200 ps | 70 ps | 20 ps | 90 ps | 90 ps | 250 ps | 15 ps       | 10 ps        |

**2.1** Nếu công việc duy nhất mà bộ xử lý cần thực hiện là nạp (fetch) các lệnh liên tiếp (Hình 4.2), thì thời gian chu kỳ xung nhịp (cycle time) là bao nhiêu?



*Hình 4.2*

**2.2** Xem xét một đường đi dữ liệu tương tự như Hình 4.3, nhưng dành cho một bộ xử lý chỉ có một loại lệnh: nhánh không điều kiện (unconditional PC-relative branch). Thời gian chu kỳ xung nhịp cho đường đi dữ liệu này là bao nhiêu?



*Hình 4.3*

### 2.3 Lặp lại câu hỏi 2.2, nhưng lần này bộ xử lý chỉ hỗ trợ các lệnh nhánh có điều kiện (conditional PC-relative branch).

Ba câu hỏi tiếp theo liên quan đến thành phần "Shift-Left-2" trên đường đi dữ liệu:

## 2.4 Những loại lệnh nào cần sử dụng thành phần "Shift-Left-2"?

**2.5** Với những loại lệnh nào (nếu có), thành phần này nằm trên đường dẫn quan trọng (critical path)?

**2.6** Giả sử bộ xử lý chỉ hỗ trợ các lệnh beq và add, hãy thảo luận về cách các thay đổi trong độ trễ của thành phần này ảnh hưởng đến thời gian chu kỳ xung nhịp của bộ xử lý. Giả sử độ trễ của các thành phần khác không thay đổi.

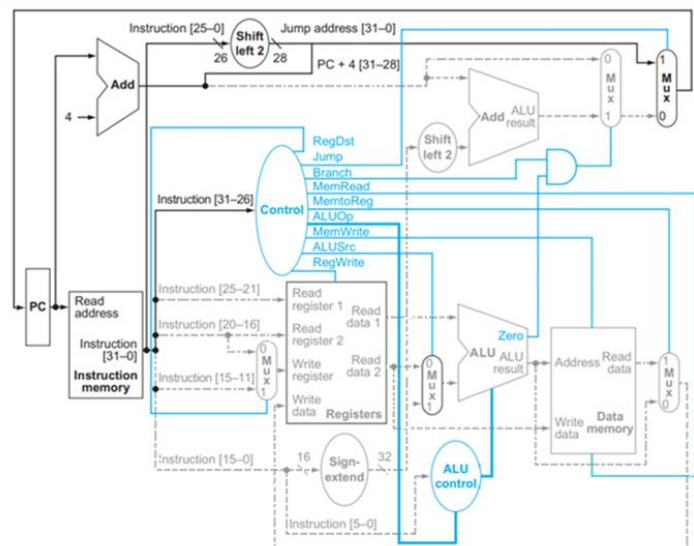
**Bài tập 3 [4.7/360 - Computer Organization and Design]:** Trong bài tập này, chúng ta sẽ xem xét chi tiết cách một lệnh được thực thi trong một đường đi dữ liệu (datapath) đơn chu kỳ. Các vấn đề trong bài tập này đề cập đến một chu kỳ xung nhịp trong đó bộ xử lý nạp (fetch) từ bộ nhớ lệnh với từ lệnh sau đây:

101011000110001000000000000010100.

Giả sử rằng bộ nhớ dữ liệu (data memory) chứa toàn số 0 và các thanh ghi của bộ xử lý có giá trị như sau tại thời điểm bắt đầu của chu kỳ khi từ lệnh trên được nạp:

| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r8 | r12 | r31 |
|----|----|----|----|----|----|----|----|-----|-----|
| 0  | -1 | 2  | -3 | -4 | 10 | 6  | 8  | 2   | -16 |

**3.1** Đầu ra của các thành phần "Sign-Extend" và "Shift left 2" (ở gần đầu của Hình 4.4) là gì đối với từ lệnh này?



Hình 4.4

**3.2** Các giá trị của các ngõ vào của đơn vị điều khiển ALU (ALU control unit) cho lệnh này là gì?

**3.3** Địa chỉ của bộ đếm chương trình (PC) mới sau khi lệnh này được thực thi là gì? Hãy làm nổi bật đường dẫn dữ liệu (datapath) thông qua đó giá trị này được xác định.

**3.4** Đối với mỗi Mux, hãy hiển thị giá trị đầu ra dữ liệu của nó trong quá trình thực thi lệnh này và các giá trị thanh ghi đã cho.

**3.5** Đối với ALU và hai bộ cộng (add units), các giá trị đầu vào dữ liệu của chúng là gì?

**3.6** Các giá trị của tất cả các đầu vào cho đơn vị "Registers" là gì?

Donate:  
TRAN MINH PHU

