

BÀI TẬP CHƯƠNG 5: KỸ THUẬT ỚNG DẪN (PINELINE)

Bài tập 1 [4.8/361 - Computer Orgranization and Design]: Trong bài tập này, chúng ta xem xét cách mà pipelining ảnh hưởng đến thời gian chu kỳ xung nhịp của bộ xử lý. Các bài tập giả định rằng các giai đoạn riêng lẻ của datapath có độ trễ (latency) như sau:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Ngoài ra, giả sử các lệnh được thực thi bởi bộ xử lý được chia như sau:

ALU	BEQ	LW	SW
45%	20%	20%	15%

- Thời gian chu kỳ xung nhịp của bộ xử lý trong trường hợp có pipelining và không có pipelining là bao nhiêu?
- Độ trễ tổng cộng của một lệnh **LW** trong trường hợp có pipelining và không có pipelining là bao nhiêu?
- Nếu chúng ta có thể chia một giai đoạn của datapath trong pipelining thành hai giai đoạn mới, mỗi giai đoạn có độ trễ bằng một nửa độ trễ của giai đoạn ban đầu, bạn sẽ chọn chia giai đoạn nào và thời gian chu kỳ xung nhịp mới của bộ xử lý sẽ là bao nhiêu?
- Giả sử không có hiện tượng **stall** hoặc **hazard**, mức độ sử dụng của bộ nhớ dữ liệu (data memory) là bao nhiêu?
- Giả sử không có hiện tượng **stall** hoặc **hazard**, mức độ sử dụng của cổng ghi (write-register port) trong đơn vị “Registers” là bao nhiêu?
- Thay vì tổ chức theo kiểu một chu kỳ (single-cycle), chúng ta có thể sử dụng tổ chức đa chu kỳ (multi-cycle), trong đó mỗi lệnh mất nhiều chu kỳ nhưng một lệnh hoàn thành trước khi lệnh khác được lấy (fetched). Trong tổ chức này, một lệnh chỉ đi qua các giai đoạn mà nó thực sự cần (ví dụ: **ST** chỉ mất 4 chu kỳ vì không cần giai đoạn **WB**). So sánh thời gian chu kỳ xung nhịp và thời gian thực thi với tổ chức kiểu single-cycle, multi-cycle, và pipelined.

Bài tập 2 [4.13/364 - Computer Organization and Design] Bài tập này nhằm giúp bạn hiểu mối quan hệ giữa chuyển tiếp dữ liệu (**forwarding**), phát hiện hazard (**hazard detection**), và thiết kế ISA. Các vấn đề trong bài tập này dựa trên dãy lệnh sau, giả sử rằng nó được thực thi trên một datapath pipeline 5 giai đoạn:

```
add r5, r2, r1
lw  r3, 4(r5)
lw  r2, 0(r2)
or  r3, r5, r3
sw  r3, 0(r5)
```

- a) Nếu không có chuyển tiếp dữ liệu (**forwarding**) hoặc phát hiện hazard (**hazard detection**), hãy chèn các lệnh **nop** để đảm bảo thực thi đúng.
- b) Lặp lại bài 4.13.1 nhưng lần này chỉ chèn lệnh **nop** khi không thể tránh được hazard bằng cách thay đổi hoặc sắp xếp lại các lệnh. Giả sử thanh ghi **R7** có thể được sử dụng để lưu giá trị tạm thời trong mã đã chỉnh sửa.
- c) Nếu bộ xử lý có cơ chế chuyển tiếp dữ liệu (**forwarding**) nhưng chúng ta quên triển khai đơn vị phát hiện hazard (**hazard detection unit**), điều gì sẽ xảy ra khi đoạn mã này được thực thi?

Bài tập 3: Xác định các xung đột trong chương trình sau, biết đã có dùng kỹ thuật ghi nửa chu kỳ đầu và đọc cuối chu kỳ. Thêm **nop** để chương trình chạy không xung đột.

```
add $s1, $s2, $s3
sub $t0, $s2, $s3
sub $t1, $s1, $s2
lw  $t0, 20($s3)
addu $t1, $t0, $t1
```

Bài tập 4: Cho thời gian trễ của mỗi khối như sau

Bộ nhớ lệnh	ALU	Register	Bộ nhớ dữ liệu
180ps	150ps	100ps	200ps

a) Cho 3 lệnh:

```
lw $s1, 40($s2)      add $s1, $s2, $s3      beq $t1, $zero, L1
```

Yêu cầu: ước lượng thời gian xử lý của từng câu lệnh trên

b) Bỏ qua độ trễ của khối ADD, MUX, Control, tính thời gian chu kỳ của hệ thống đơn chu kỳ và pipeline?

c) Tính speed-up khi thực thi một chương trình 100 dòng lệnh (không có xung đột) của hệ thống đơn chu kỳ và pipeline.

Donate:
TRAN MINH PHU

