

## Chương 5. Kỹ thuật ống dẫn

**Bài tập 1 [4.8/361 - Computer Organization 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%

- a) Thời gian chu kỳ xung nhịp của bộ xử lý trong trường hợp có pipelining và không có pipelining là bao nhiêu?

Có pipeline :  $T = 350 \text{ ps}$

Không pipeline :  $T = 250 + 350 + 150 + 300 + 200 = 1250 \text{ ps}$

- b) Độ trễ tổng cộng của một lệnh LW trong trường hợp có pipelining và không có pipelining là bao nhiêu?

Độ trễ khi có pipeline =  $350 \times 5 = 1750 \text{ ps}$

Độ trễ khi không có pipeline =  $1250 \text{ ps}$

↳ Số giai đoạn  $\times$  chu kỳ xung nhịp

- c) 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?

Giai đoạn có độ trễ lớn nhất là ID. Nếu ta chia giai đoạn ID thành 2 giai đoạn mới thì độ trễ của mỗi giai đoạn mới là  $350 : 2 = 175 \text{ ps}$

Sau khi chia, giai đoạn có độ trễ lớn nhất bây giờ là MEM

$\Rightarrow T' = 300 \text{ ps}$

- d) Giả sử không có hiện tượng **stall** hoặc **hazard**, mức độ sử dụng của bộ nhớ dữ liệu (data memory) là bao nhiêu?

⊗ Mức độ sử dụng: tỷ lệ % các chu kỳ mà bộ nhớ dữ liệu được truy cập trong pipeline

Data Memory được sử dụng trong giai đoạn MEM của pipeline  $\rightarrow$  mức độ sử dụng của bộ nhớ dữ liệu phụ thuộc vào các lệnh có sử dụng MEM  $\rightarrow$  lw, sw

$$\Rightarrow \text{Mức độ sử dụng} = 20\% + 15\% = 35\%$$

e) 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?

$\rightarrow$  WB } ALU  
lw

$$\text{Mức độ sử dụng} = 45\% + 20\% = 65\%$$

f) 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.

⊗ single - cycle

$$\text{Thời gian chu kỳ xung nhịp} = 1250 \text{ ps}$$

$$\text{Thời gian thực thi} = 1250 \text{ ps} = \text{thời gian thực thi trung bình}$$

⊗ Trong tổ chức multi-cycle, mỗi lệnh chỉ đi qua các giai đoạn mà nó thực sự cần, và 1 lệnh hoàn thành trước khi lệnh tiếp theo được lấy

$$\text{Thời gian chu kỳ xung nhịp} = 350 \text{ ps}$$

Thời gian thực thi:

- ALU:  $4 \times 350 = 1400 \text{ ps}$

- Beq:  $3 \times 350 = 1050 \text{ ps}$

- lw:  $5 \times 350 = 1750 \text{ ps}$

- sw:  $4 \times 350 = 1400 \text{ ps}$

$$\Rightarrow \text{Thời gian thực thi trung bình} = 1400 \cdot 45\% + 1050 \cdot 20\% + 1750 \cdot 20\% + 1400 \cdot 15\% = 1400 \text{ ps}$$

⊗ pipeline

Thời gian chu kỳ xung nhịp = 350ps

Thời gian thực thi:

- ALU :  $4 \times 350 = 1400 \text{ ps}$
- Beq :  $3 \times 350 = 1050 \text{ ps}$
- lw :  $5 \times 350 = 1750 \text{ ps}$
- sw :  $4 \times 350 = 1400 \text{ ps}$

$$\Rightarrow \text{Thời gian thực thi trung bình} = 1400 \cdot 45\% + 1050 \cdot 20\% + 1750 \cdot 20\% + 1400 \cdot 15\% \\ = 1400 \text{ ps}$$

**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.

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

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?

```
add r5, r2, r1
```

```

lw r3, 4(r5)
lw r2, 0(r2)
nop
or r3, r5, r3
sw r3, 0(r5)

```

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

```

add $s1, $s2, $s3
sub $t0, $s2, $s3  nop
sub $t1, $s1, $s2
lw $t0, 20($s3)  nop  nop
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:

$lw \$s1, 40(\$s2) = 730$      
 $add \$s1, \$s2, \$s3 = 530$      
 $beq \$t1, \$zero, L1 = 530$

**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 kì của hệ thống đơn chu kì và pipeline?

$$\tau_{single} = 730ps ; \tau_{pipeline} = 200ps$$

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

$$\begin{aligned}
 \text{Speed up} &= \frac{100 \cdot 730}{(5 + 100 - 1) \cdot 200} \\
 &= 3,51
 \end{aligned}$$