

# Bài Tập (Pipeline)

---oOo---

Các bài tập chương này được trích dẫn và dịch lại từ:

*Computer Organization and Design: The Hardware/Software Interface*, Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, **Third Edition**, 2011.

-----

## Bài 1 (4.12 - sách tham khảo chính)

Trong bài tập này, chúng ta khảo sát pipeline ảnh hưởng như thế nào tới chu kỳ xung clock (clock cycle time) của processor. Giả sử rằng mỗi công đoạn (stage) trong pipeline có thời gian thực hiện

	IF	ID	EX	MEM	WB
a.	300ps	400ps	350ps	500ps	100ps
b.	200ps	150ps	120ps	190ps	140ps

- Chu kỳ xung clock cần cho processor là bao nhiêu nếu processor thiết kế có pipeline và không pipeline đơn chu kỳ.

**Trả lời:**

- Processor có pipeline: 500ps (trong 5 công đoạn, công đoạn nào chiếm thời gian dài nhất, chọn công đoạn đó)

Processor không pipeline:  $300 + 400 + 350 + 500 + 100 = 1650\text{ps}$

- Processor có pipeline: 200ps

Processor không pipeline:  $200 + 150 + 120 + 190 + 140 = 800\text{ps}$

- Thời gian cần thiết để thực hiện lệnh lw cho trường hợp processor có pipeline và không pipeline đơn chu kỳ có thể là bao nhiêu

**Trả lời:**

- Thời gian lw cần để thực hiện trong trường hợp processor có pipeline:

lw cần trải qua các công đoạn: IF + ID + EX + MEM + WB, 5 công đoạn. Các công đoạn tuy có thời gian hoàn thành khác nhau, nhưng vì có pipeline đưa vào nên các công đoạn được thiết kế sao cho thời gian thực hiện như nhau và bằng thời gian của công đoạn lâu nhất; vì vậy, thời gian lw cần để thực hiện trong trường hợp có pipeline:  $5 * 500 = 2500\text{ps}$

Thời gian lw cần để thực hiện trong trường hợp processor đơn chu kỳ không có pipeline bằng một chu kỳ xung clock của processor đơn chu kỳ không pipeline = 1650ps ( $300 + 400 + 350 + 500 + 100 = 1650\text{ps}$ )

- b. Thời gian cần cho lw khi processor có pipeline:  $5 * 200 = 1000\text{ ps}$   
 Thời gian cần cho lw khi processor không pipeline: 800ps

**Hỏi thêm:** Thời gian cần thiết để thực hiện lệnh **add** cho trường hợp processor có pipeline và không pipeline đơn chu kỳ có thể là bao nhiêu.

**Trả lời:**

*add* chỉ cần trải qua các công đoạn: IF + ID + EX + WB, không có MEM. Nhưng khi đưa vào thiết kế pipeline này, lệnh nào cũng phải trải qua 5 công đoạn. Công đoạn không có tác vụ gì (ví dụ MEM của *add*) thì vẫn phải trải qua nhưng thời gian đó lệnh sẽ không làm gì.

*Chú ý: các thiết kế pipeline cải tiến khác có thể khác. Nhưng trong phạm vi môn học này, theo sách tham khảo chính, các bài tập sẽ tuân theo thiết kế pipeline này.*

Vậy lệnh *add* theo thiết kế pipeline này vẫn phải trải qua 5 công đoạn; mỗi công đoạn tương ứng với 1 chu kỳ xung clock là 500ps. Toàn bộ thời gian *add* cần để thực hiện trong trường hợp có pipeline:  $5 * 500 = 2500\text{ps}$

Với processor đơn chu kỳ không có pipeline, dù *add* thực sự chỉ chạy qua các công đoạn IF + ID + EX + WB, không có MEM ( $300+400+350+100 = 1150\text{ps}$ ), tức sau 1150ps là đã hoàn thành. Nhưng với thiết kế processor đơn chu kỳ, không pipeline, lệnh nào cũng phải chờ đúng một chu kỳ xung clock của processor mới được chuyển sang lệnh khác.

Vì vậy, thời gian để thực hiện lệnh *add* của processor đơn chu kỳ không pipeline bằng đúng một chu kỳ xung clock = 1650ps ( $300 + 400 + 350 + 500 + 100 = 1650\text{ps}$ )

Giả sử rằng các lệnh được thực thi trong processor được phân rã như sau (áp dụng cho câu 3 và 4)

	ALU	beq	lw	sw
--	-----	-----	----	----

a.	50%	25%	15%	10%
b.	30%	25%	30%	15%

3. Giả sử rằng không có khoảng thời gian rỗi (stalls) hoặc xung đột (hazards), phần truy xuất bộ nhớ (MEM) và phần truy xuất ghi trên tập thanh ghi (WB) sử dụng bao nhiêu % chu kỳ của toàn chương trình

**Trả lời:**

**phần truy xuất bộ nhớ dữ liệu (MEM)**

thao tác truy xuất bộ nhớ dữ liệu thật sự chỉ thực hiện trên 2 dạng lệnh: lw và sw. Vì vậy % chu kỳ MEM có thao tác thật sự:

- a.  $15\% + 10\% = 25\%$   
b.  $30\% + 15\% = 45\%$

**phần truy xuất ghi trên tập thanh ghi (WB)**

thao tác truy xuất ghi trên khối thanh ghi thật sự chỉ xuất hiện ở 2 loại lệnh: ALU và lw. Vì vậy % chu kỳ WB có thao tác thật sự:

- a.  $50\% + 15\% = 65\%$   
b.  $30\% + 30\% = 60\%$

Chú ý: Bảng sau liệt kê chi tiết lệnh nào thật sự cần và không cần công đoạn nào

	IF	ID	EX	MEM	WB
lw	x	x	x	x	x
sw	x	x	x	x	
ALU (add, sub, AND, OR, slt)	x	x	x		x
Branch (beq)	x	x	x		

(Lệnh branch thực hiện 3 stage hoặc 4 stages tùy vào một số thay đổi, nếu có cho vấn đề này, bài tập sẽ nói rõ)

4. Giả sử có thiết kế mới như sau: mỗi lệnh chỉ sử dụng đúng các giai đoạn cần có của nó, có thể lấy nhiều chu kỳ để hoàn thành, nhưng một lệnh phải hoàn thành xong thì những lệnh khác mới được nạp vào. Thiết kế này tạm gọi là thiết kế đa chu kỳ.

Theo kiểu này, mỗi lệnh chỉ đi qua những công đoạn mà nó thực sự cần (ví dụ, sw chỉ sử dụng 4 công đoạn, không có công đoạn WB). Tính chu kỳ xung clock, so sánh thời gian thực thi của thiết kế đa chu kỳ này với thiết kế đơn chu kỳ (single cycle design) và pipeline. (Chú ý: lw: sử dụng 5 stage; sw: 4 stage (không WB); ALU: 4 stage (không MEM), beq 4 stage (không WB))

### Trả lời:

- Chu kỳ xung clock
  - Chu kỳ xung clock của thiết kế pipeline
    - a. 500 ps
    - b. 200 ps
  - Chu kỳ xung clock của thiết kế đơn chu kỳ
    - a. 1650
    - b. 800
  - Chu kỳ xung clock của thiết kế đa chu kỳ: giống pipeline

- Execution time

a.

$$\frac{\text{Multi-cycle execution time}}{\text{pipeline execution time}} = 0.15 \times 5 + 0.85 \times 4 = 4.15$$

$$\frac{\text{single-cycle execution time}}{\text{pipeline execution time}} = \frac{1650}{500} = 3.3$$

b.  $\frac{\text{Multi-cycle execution time}}{\text{pipeline execution time}} = 0.3 \times 5 + 0.7 \times 4 = 4.30$

$$\frac{\text{single-cycle execution time}}{\text{pipeline execution time}} = \frac{800}{200} = 4$$

(Gợi ý:

- Thời gian thực thi n lệnh của pipeline =  $5T + (n-1).T$  với T là 1 chu kỳ pipeline
- Thời gian thực thi n lệnh của kiểu thiết kế đơn chu kỳ =  $nT'$  (với T' là thời gian 1 chu kỳ xung clock của đơn chu kỳ)
- Thời gian thực thi n lệnh của kiểu thiết kế đa chu kỳ =  $(0.15 \times 5 + 0.85 \times 4) nT''$  (với T'' là thời gian 1 chu kỳ xung clock của đa chu kỳ, cũng bằng T của pipeline), do chỉ có 15% lệnh lw là chiếm 5 chu kỳ, 85% các lệnh khác đều 4 chu kỳ (cho câu a)

$$\Rightarrow \frac{\text{Thời gian thực thi của đa chu kỳ}}{\text{Thời gian thực thi của pipeline}} = \frac{(0.15 \times 5 + 0.85 \times 4) nT}{5T + (n-1).T}$$

Khi n là rất lớn,  $\frac{\text{Thời gian thực thi của đa chu kỳ}}{\text{Thời gian thực thi của pipeline}} = 0.15 \times 5 + 0.85 \times 4$

$$\frac{\text{Thời gian thực thi của đơn chu kỳ}}{\text{Thời gian thực thi của pipeline}} = \frac{nT'}{5T + (n-1).T}$$

Khi  $n$  là rất lớn,  $\frac{\text{single-cycle execution time}}{\text{pipeline execution time}} = \frac{T'}{T}$

### Tóm tắt:

**- Pipeline:** Chu kỳ xung clock = Thời gian của công đoạn dài nhất

Thời gian thực thi 1 lệnh bất kỳ = 5 \* chu kỳ xung clock

**- Không pipeline:**

+ Đơn chu kỳ:

Chu kỳ xung clock = Tổng thời gian các công đoạn của lệnh dài nhất (lw)

Thời gian thực thi 1 lệnh bất kỳ = chu kỳ xung clock

+ Đa chu kỳ:

Chu kỳ xung clock = Thời gian của công đoạn dài nhất

Thời gian thực thi 1 lệnh bất kỳ = số công đoạn \* chu kỳ xung clock

Với tất cả các bài tập theo sau, bộ nhớ dữ liệu và bộ nhớ lệnh là riêng lẻ nên mặc định không có xung đột cấu trúc xảy ra.

### **Bài 2 (4.13 – sách tham khảo chính)**

Cho chuỗi lệnh như sau :

a.

lw \$1, 40(\$6)

add \$6, \$2, \$2

sw \$6, 50(\$1)

b.

lw \$5, -16(\$5)

sw \$5, -16(\$5)

add \$5, \$5, \$5

1. Trong trường hợp pipeline 5 tầng và không dùng kỹ thuật nhìn trước (no forwarding), sử dụng lệnh 'nop' để giải quyết xung đột xảy ra (nếu có) trong chuỗi lệnh trên.
2. Trong trường hợp pipeline 5 tầng và có kỹ thuật nhìn trước (forwarding), sử dụng lệnh 'nop' để giải quyết xung đột xảy ra (nếu có) trong chuỗi lệnh trên.

*Chú ý: Vẽ rõ ràng hình ảnh các chu kỳ pipeline khi đoạn lệnh trên thực thi*

Cho bảng thể hiện chu kỳ xung clock như sau

	Không forwarding	Có forwarding đầy đủ (full-forwarding)	Chỉ có ALU-ALU forwarding, không có MEM-ALU forwarding
a.	300ps	400ps	325ps
b.	200ps	250ps	220ps

*Chú ý : ALU-ALU forwarding cũng chính là EX-EX forwarding. MEM-ALU forwarding cũng chính là MEM-EX forwarding.*

3. Tính thời gian thực thi của chuỗi lệnh trên trong trường hợp không forwarding và có full-forwarding? Sự tăng tốc đạt được bởi việc đưa kỹ thuật full-forwarding vào pipeline so với không forwarding là bao nhiêu?
4. Giả sử processor chỉ có kỹ thuật ALU-ALU forwarding (không có MEM-ALU forwarding), sử dụng lệnh ‘nop’ để giải quyết xung đột dữ liệu
5. Tính thời gian thực thi của chuỗi lệnh trên khi áp dụng ALU-ALU forwarding? Sự tăng tốc đạt được của việc dùng ALU-ALU forwarding so với không forwarding là bao nhiêu?

#### Trả lời:

1. Thêm lệnh nop để giải quyết xung đột dữ liệu trong trường hợp không forwarding

a.

```
lw $1, 40($6)
add $6, $2, $2
nop
nop
sw $6, 50($1)
```

#### Lưu ý: Cho đoạn lệnh sau

```
add $6, $2, $2
sw $6, 50($1)
```

Với tất cả các lệnh, các thanh ghi cần lấy giá trị đều được lấy trong giai đoạn giải mã; riêng với lệnh sw, thanh ghi \$6 như ví dụ trên có thể xảy ra hai trường hợp:

- Giá trị của \$6 sẽ được lấy trong công đoạn ID như tất cả các lệnh khác  
Giải quyết xung đột trong trường hợp này sw phải chờ sao cho ID của sw trùng với WB của add (thêm 2 nop ở giữa)
- Hoặc là giá trị \$6 chỉ cần sẵn sàng trước khi lệnh sw vào MEM

**Giải quyết xung đột trong trường hợp này sw phải chờ sao cho EX của sw trùng với WB của add (thêm 1 nop ở giữa)**

**Các bạn hiện tại có thể dùng 2 cách.**

**Phần bài giải phía sau giải theo trường hợp 1.**

Hình ảnh chu kỳ pipeline

*lw \$1, 40(\$6)*  
*add \$6, \$2, \$2*  
***nop***  
***nop***  
*sw \$6, 50(\$1)*

IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		o	o	o	o	o		
			o	o	o	o	o	
				IF	ID	EX	MEM	WB

b.

*lw \$5, -16(\$5)*  
***nop***  
***nop***  
*sw \$5, -16(\$5)*  
*add \$5, \$5, \$5*

Hình ảnh chu kỳ pipeline

*lw \$5, -16(\$5)*  
***nop***  
***nop***  
*sw \$5, -16(\$5)*  
*add \$5, \$5, \$5*

IF	ID	EX	MEM	WB				
	o	o	o	o	o			
		o	o	o	o	o		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

2. Thêm lệnh nop để giải quyết xung đột dữ liệu trong trường hợp có forwarding

a.

*lw \$1, 40(\$6)*  
*add \$6, \$2, \$2*  
*sw \$6, 50(\$1)*

Hình ảnh chu kỳ pipeline

lw \$1, 40(\$6)  
add \$6, \$2, \$2  
sw \$6, 50(\$1)

IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		

b.

lw \$5, -16(\$5)  
***nop***  
sw \$5, -16(\$5)  
add \$5, \$5, \$5

Hình ảnh chu kỳ pipeline

lw \$5, -16(\$5)  
***nop***  
sw \$5, -16(\$5)  
add \$5, \$5, \$5

IF	ID	EX	MEM	WB				
	o	o	o	o	o			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	

3.

a.

• **Chuỗi lệnh trong trường hợp không forwarding**

lw \$1, 40(\$6)  
add \$6, \$2, \$2  
***nop***  
***nop***  
sw \$6, 50(\$1)

⇒ Đoạn lệnh này mất 9 chu kỳ xung clock để thực hiện nên thời gian thực thi =  $9 \times 300 = 2700$  ps.

• **Chuỗi lệnh trong trường hợp full-forwarding**

lw \$1, 40(\$6)  
add \$6, \$2, \$2  
sw \$6, 50(\$1)

⇒ Đoạn lệnh này mất 7 chu kỳ xung clock để thực hiện nên thời gian thực thi =  $7 \times 400 = 2800$  ps.

Tăng tốc đạt được:  $2700/2800 = 0.96$

➔ Trường hợp này là thật sự là một sự giảm tốc, không phải tăng

b.

Với số liệu câu b.

Thời gian thực thi không forwarding:  $9 \times 200 = 1800$ ps



Thời gian thực thi có full-forwarding:  $7 \times 250 = 1750\text{ps}$

Sự tăng tốc đạt được:  $1800/1750 = 1.0285$

→ Trường hợp này mới thật là một sự tăng tốc

4.

a.

*lw \$1, 40(\$6)*  
*add \$6, \$2, \$2*  
*nop\**  
*sw \$6, 50(\$1)*

Chú ý: *nop\** đứng trước hay sau lệnh *add* đều được.

b.

*lw \$5, -16(\$5)*  
*nop*  
*nop*  
*sw \$5, -16(\$5)*  
*add \$5, \$5, \$5*

5.

a.

Thời gian thực thi với ALU-ALU forwarding:  $8 \times 325 = 2600\text{ps}$

Sự tăng tốc đạt được =  $2700/2600 = 1.038$

b.

Thời gian thực thi với ALU-ALU forwarding:  $9 \times 220 = 1980\text{ps}$

Sự tăng tốc đạt được =  $1800/1980 = 0.91$

### Tóm tắt:

- **Không sử dụng forwarding:** Nếu lệnh phía sau có sử dụng thanh ghi là kết quả của lệnh phía trước thì cần thêm lệnh *nop* sao cho lệnh phía sau cách lệnh phía trước **2 câu lệnh**
- **Chỉ sử dụng ALU-ALU forwarding:** Nếu lệnh phía sau có sử dụng thanh ghi là kết quả của lệnh phía trước, thì nếu:
  - + Lệnh phía trước là lệnh *lw*: thì cần thêm lệnh *nop* sao cho lệnh phía sau cách lệnh phía trước **2 câu lệnh**
  - + Lệnh phía trước là các lệnh khác: thì **không cần thêm lệnh *nop***

- **Sử dụng full forwarding:** Nếu lệnh phía sau có sử dụng thanh ghi là kết quả của lệnh phía trước, thì nếu:

+ Lệnh phía trước là lệnh lw: thì cần thêm lệnh nop sao cho lệnh phía sau      cách

lệnh phía trước **1 câu lệnh**

+ Lệnh phía trước là các lệnh khác: thì **không cần thêm lệnh nop**

### Bài 3

Cho đoạn lệnh sau :

a.

```
lw $1, 40($2)
add $2, $3, $3
add $1, $1, $2
sw $1, 20($2)
```

b.

```
add $1, $2, $3
sw $2, 0($1)
lw $1, 4($2)
add $2, $2, $1
```

c.

```
lw $1, 40($6)
add $2, $3, $1
add $1, $6, $4
sw $2, 20($4)
add $1, $1, $4
```

d.

```
add $1, $5, $3
sw $1, 0($2)
lw $1, 4($2)
add $5, $5, $1
sw $1, 0($2)
```

1. Trong trường hợp pipeline 5 tầng, không nhìn trước (no forwarding), sử dụng lệnh nop để giải quyết nếu có xung đột xảy ra trong chuỗi lệnh trên.
2. Trong trường hợp pipeline 5 tầng, có nhìn trước (forwarding), sử dụng lệnh nop để giải quyết nếu có xung đột xảy ra trong chuỗi lệnh trên.

*Chú ý: Vẽ rõ ràng hình ảnh các chu kỳ pipeline khi đoạn lệnh trên thực thi*

Trả lời :

	Không forwarding	Có forwarding
a.	<i>lw \$1, 40(\$2)</i> <i>add \$2, \$3, \$3</i> <i>nop</i> <i>nop</i> <i>add \$1, \$1, \$2</i> <i>nop</i> <i>nop</i> <i>sw \$1, 20(\$2)</i>	<i>lw \$1, 40(\$2)</i> <i>add \$2, \$3, \$3</i> <i>add \$1, \$1, \$2</i> <i>sw \$1, 20(\$2)</i>
b.	<i>add \$1, \$2, \$3</i> <i>nop</i> <i>nop</i> <i>sw \$2, 0(\$1)</i> <i>lw \$1, 4(\$2)</i> <i>nop</i> <i>nop</i> <i>add \$2, \$2, \$1</i>	<i>add \$1, \$2, \$3</i> <i>sw \$2, 0(\$1)</i> <i>lw \$1, 4(\$2)</i> <i>nop</i> <i>add \$2, \$2, \$1</i>
c.	<i>lw \$1, 40(\$6)</i> <i>nop</i> <i>nop</i> <i>add \$2, \$3, \$1</i> <i>add \$1, \$6, \$4</i> <b><i>nop*</i></b> <i>sw \$2, 20(\$4)</i> <i>add \$1, \$1, \$4</i>	<i>lw \$1, 40(\$6)</i> <i>nop</i> <i>add \$2, \$3, \$1</i> <i>add \$1, \$6, \$4</i> <i>sw \$2, 20(\$4)</i> <i>add \$1, \$1, \$4</i>
d.	<i>add \$1, \$5, \$3</i> <i>nop</i> <i>nop</i> <i>sw \$1, 0(\$2)</i> <i>lw \$1, 4(\$2)</i>	<i>add \$1, \$5, \$3</i> <i>sw \$1, 0(\$2)</i> <i>lw \$1, 4(\$2)</i> <i>nop</i> <i>add \$5, \$5, \$1</i>

	<i>nop</i> <i>nop</i> <i>add \$5, \$5, \$1</i> <i>sw \$1, 0(\$2)</i>	<i>sw \$1, 0(\$2)</i>
--	---	-----------------------