

BÀI TẬP CHƯƠNG 5 – PIPELINE

Bài 1: (4.12 - sách tham khảo chính)

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). (Chú ý: lw: sử dụng 5 stages; sw: 4 stages (không WB); ALU: 4 stages (không MEM), beq 3 stages (không MEM và không WB))

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		

Trong bài tập này, chúng ta khảo sát pipeline ảnh hưởng như thế nào tới chu kỳ xung clock (clock cycle time) và thời gian thực thi của 1 lệnh (hoặc đoạn chương trình nhiều lệnh) 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, không pipeline đa chu kỳ và không pipeline đơn chu kỳ.**
 - Processor có pipeline: 500 ps
Processor không có pipeline: $300 + 400 + 350 + 500 + 100 = 1650$ ps
 - Processor có pipeline: 200 ps
Processor không có pipeline: $200 + 150 + 120 + 190 + 140 = 800$ ps
- Thời gian cần thiết để thực hiện lệnh “lw” cho trường hợp processor thiết kế có pipeline, không pipeline đa chu kỳ và không pipeline đơn chu kỳ.**
 - Thời gian thực hiện lệnh “lw” trong trường hợp processor có pipeline: $5 \times 500 = 2500$ ps
Thời gian thực hiện lệnh “lw” trong trường hợp processor không có pipeline: $300 + 400 + 350 + 500 + 100 = 1650$ ps
 - Thời gian thực hiện lệnh “lw” trong trường hợp processor có pipeline: $5 \times 200 = 1000$ ps
Thời gian thực hiện lệnh “lw” trong trường hợp processor không có pipeline: $200 + 150 + 120 + 190 + 140 = 800$ ps
- Thời gian cần thiết để thực hiện lệnh “add” cho trường hợp processor thiết kế có pipeline, không pipeline đa chu kỳ và không pipeline đơn chu kỳ.**
 - Thời gian thực hiện lệnh “add” trong trường hợp processor có pipeline: $5 \times 500 = 2500$ ps
Thời gian thực hiện lệnh “add” trong trường hợp processor không có pipeline: $300 + 400 + 350 + 500 + 100 = 1650$ ps
 - Thời gian thực hiện lệnh “add” trong trường hợp processor có pipeline: $5 \times 200 = 1000$ ps

Thời gian thực hiện lệnh “add” trong trường hợp processor không có pipeline: $200 + 150 + 120 + 190 + 140 = 800$ ps

4. Thời gian cần thiết để thực hiện một chương trình có 10 lệnh bao gồm 4 lệnh “lw” và 6 lệnh “add” cho trường hợp processor thiết kế có pipeline, không pipeline đa chu kỳ và không pipeline đơn chu kỳ.

Giả sử rằng các lệnh được thực thi trong processor được phân rã như sau (áp dụng cho câu 5, 6)

	ALU	beq	lw	sw
a.	50%	25%	15%	10%
b.	30%	25%	30%	15%

5. 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.

	IF	ID	EX	MEM	WB
lw	x	x	x	x	x
sw	x	x	x	x	
ALU	x	x	x		x
beq	x	x	x		

❖ Phần truy xuất bộ nhớ dữ liệu (MEM):

a. $15\% + 10\% = 25\%$

b. $30\% + 15\% = 45\%$

❖ Phần truy xuất ghi trên tập thanh ghi (WB):

a. $50\% + 15\% = 65\%$

b. $30\% + 30\% = 60\%$

6. Thời gian cần thiết để thực hiện một chương trình có 100 lệnh được phân bố như bảng trên cho trường hợp processor thiết kế có pipeline, không pipeline đa chu kỳ và không pipeline đơn chu kỳ.

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 ps

b. 800 ps

Chu kỳ xung clock của thiết kế đa chu kỳ : giồng pipeline

$$\begin{aligned} \text{a) Thời gian thực thi 100 lệnh của thiết kế có pipeline} &= 5T + (n - 1).T \\ &= 5.500 + (100 - 1).500 \\ &= 52000 \text{ ps} \end{aligned}$$

$$\text{Thời gian thực thi 100 lệnh của thiết kế đơn chu kỳ} = n.T = 100.1650 = 165000 \text{ ps}$$

$$\begin{aligned} \text{Thời gian thực thi 100 lệnh của thiết kế đa chu kỳ} &= (0.15 \times 5 + 0.6 \times 4 + 0.25 \times 3) nT \\ &= 3.9 \times 100 \times 500 = 195000 \text{ ps} \end{aligned}$$

$$\begin{aligned} \text{b) Thời gian thực thi 100 lệnh của thiết kế có pipeline} &= 5T + (n - 1).T = 5.200 + (100 - 1).200 \\ &= 20800 \text{ ps} \end{aligned}$$

Bài 2: Cho chuỗi lệnh như sau :

a.

```
lw $s1, 40($s6)
add $s6, $s2, $s2
sw $s6, 50($s1)
```

b.

```
lw $s5, -16($s5)
sw $s5, -16($s5)
add $s5, $s5, $s5
```

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.

a. lw \$s1, 40(\$s6)

add \$s6, \$s2, \$s2

nop

nop

sw \$s6, 50(\$s1)

b. lw \$s5, -16(\$s5)

nop

nop

sw \$s5, -16(\$s5)

add \$s5, \$s5, \$s5

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.

a. lw \$s1, 40(\$s6)

add \$s6, \$s2, \$s2

sw \$s6, 50(\$s1)

b. lw \$s5, -16(\$s5)

nop

sw \$s5, -16(\$s5)

add \$s5, \$s5, \$s5

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	330ps	315ps
b.	200ps	220ps	210ps

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?

a. Chuỗi lệnh trong trường hợp không forwarding mất 9 chu kỳ xung clock để thực thi nên có thời gian: $9 \times 300 = 2700 \text{ ps}$

Chuỗi lệnh trong trường hợp full-forwarding mất 7 chu kỳ xung clock để thực thi nên có thời gian : $7 \times 400 = 2800$ ps

→ $2700/2800 = 0.96$, trường hợp này là giảm tốc.

b. Tương tự, thời gian thực thi không có forwarding : $9 \times 200 = 1800$ ps

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

→ $1800/1750 = 1.0285$, tăng tốc.

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.

a. lw \$s1, 40(\$s6)

nop*

add \$s6, \$s2, \$s2

sw \$s6, 50(\$s1)

b. lw \$s5, -16(\$s5)

nop

nop

sw \$s5, -16(\$s5)

add \$s5, \$s5, \$s5

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?

a. Thời gian thực thi với ALU-ALU forwarding: $8 \times 325 = 2600$ ps

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

b. Thời gian thực thi với ALU-ALU forwarding: $9 \times 220 = 1980$ ps

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

Bài 3:

a.

lw \$s1, 40(\$s2)

add \$s2, \$s3, \$s3

add \$s1, \$s1, \$s2

sw \$s1, 20(\$s2)

b.

add \$s1, \$s2, \$s3

sw \$s2, 0(\$s1)

lw \$s1, 4(\$s2)

add \$s2, \$s2, \$s1

c.

lw \$s1, 40(\$s6)

add \$s2, \$s3, \$s1

add \$s1, \$s6, \$s4

sw \$s2, 20(\$s4)

add \$s1, \$s1, \$s4

d.

add \$s1, \$s5, \$s3

sw \$s1, 0(\$s2)

```
lw $s1,4($s2)
add $s5, $s5, $s1
sw $s1, 0($s2)
```

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. Tính thời gian thực thi đoạn lệnh khi chu kỳ là 200ps.

a. lw \$s1, 40(\$s2)
add \$s2, \$s3, \$s3
nop
nop
add \$s1, \$s1, \$s2
nop
nop
sw \$s1, 20(\$s2)

b. add \$s1, \$s2, \$s3
nop
nop
sw \$s2, 0(\$s1)
lw \$s1, 4(\$s2)
nop
nop
add \$s2, \$s2, \$s1

c. lw \$s1, 40(\$s6)
nop
nop
add \$s2, \$s3, \$s1
add \$s1, \$s6, \$s4
nop*
sw \$s2, 20(\$s4)
add \$s1, \$s1, \$s4

d. add \$s1, \$s5, \$s3
nop
nop
sw \$s1, 0(\$s2)
lw \$s1,4(\$s2)
nop
nop
add \$s5, \$s5, \$s1
sw \$s1, 0(\$s2)

2. Trong trường hợp pipeline 5 tầng, có nhìn trước (full-forwarding), sử dụng lệnh nop để giải quyết nếu có xung đột xảy ra trong chuỗi lệnh trên. Tính thời gian thực thi đoạn lệnh khi chu kỳ là 220ps.

a. lw \$s1, 40(\$s2)
add \$s2, \$s3, \$s3
add \$s1, \$s1, \$s2

```
    sw $s1, 20($s2)
b.  add $s1, $s2, $s3
    sw $s2, 0($s1)
    lw $s1, 4($s2)
    nop
    add $s2, $s2, $s1
c.  lw $s1, 40($s6)
    nop
    add $s2, $s3, $s1
    add $s1, $s6, $s4
    sw $s2, 20($s4)
    add $s1, $s1, $s4
d.  add $s1, $s5, $s3
    sw $s1, 0($s2)
    lw $s1, 4($s2)
    nop
    add $s5, $s5, $s1
    sw $s1, 0($s2)
```