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

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

2. 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

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.

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

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))

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)

if id ex mem wb
if id ex mem wb
if id ex mem wb

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?

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