



CO2008 - KIẾN TRÚC MÁY TÍNH

Khoa Khoa học và kỹ thuật máy tính
Đại học Bách Khoa - ĐHQG Tp.HCM

02/2022

Bài thực hành 7

CHƯƠNG 4 BỘ XỬ LÝ: Kiến trúc Pipeline.

Mục tiêu

- Xác định thời gian chu kỳ của hệ thống single clock, multi clock và pipeline.
- Tính hiệu suất của hệ thống pipeline so với hệ thống single clock và multi clock.
- Hiểu chức năng của cơ chế pipeline và cách khắc phục các hiện tượng Hazard do quá trình pipeline gây ra.

Yêu cầu

- Xem lại kiến trúc Pipeline.

Hình ảnh so sánh hệ thống single cycle, multi cycle và pipeline

Single Clock cycle

Load					Add					Jump					Store					Branch				
IF	ID	EXE	MEM	WR	IF	ID	EXE	MEM	WR	IF	ID	EXE	MEM	WR	IF	ID	EXE	MEM	WR	IF	ID	EXE	MEM	WR

Multi cycle

Load					Add					Jump					Store					Branch				
IF	ID	EXE	MEM	WR	IF	ID	EXE	WR		IF	ID	IF	ID	EXE	MEM	IF	ID	EXE		IF	ID	EXE		

Pipeline

IF	ID	EXE	MEM	WR																				
	IF	ID	EXE	MEM	WR																			
		IF	ID	EXE	MEM	WR																		
			IF	ID	EXE	MEM	WR																	
				IF	ID	EXE	MEM	WR																

(Load)

(Add)

(Jump)

(Store)

(Branch)

- **Single Clock Cycle:** Một lệnh thực thi trong 1 chu kỳ. Ví dụ lệnh load thực thi trong 1 chu kỳ màu xanh, lệnh store thực thi trong 1 chu kỳ màu vàng. Thời gian của màu xanh và vàng là bằng nhau.
- **Multi Clock Cycle:** Một lệnh thực thi trong nhiều chu kỳ. Ví dụ lệnh load thực thi trong 5 chu kỳ màu xanh (5 chu kỳ nhỏ này tương ứng với 1 chu kỳ lớn bên single clock cycle), lệnh store thực thi trong 4 chu kỳ màu vàng.
- **Pipeline:** Lệnh đầu tiên thực thi trong 5 chu kỳ, các lệnh còn lại hoàn thành chỉ mất 1 chu kỳ.

Các bước hiện thực lệnh MIPS

Bộ xử lý Pipeline chia quá trình thực thi lệnh thành 5 bước, mỗi bước thực thi trong một chu kỳ.

1. **IF:** Lấy lệnh (liên quan khối Instruction memory), 32bits lệnh chứa các thông tin của 1 lệnh được lấy ra từ Instruction memory.
2. **ID:** Giải mã lệnh (liên quan khối Registers và Control), xác định toán tử, các tín hiệu điều khiển, nội dung các thanh ghi, giá trị Immediate.
3. **EXE:** Thực thi tác vụ lệnh (liên quan khối ALU).
4. **MEM:** Truy xuất vùng nhớ (liên quan khối Data memory) - chỉ dùng cho lệnh load/store.
5. **WB:** Ghi kết quả vào thanh ghi (liên quan khối Registers).

Bài tập và thực hành

Bài 1. Xác định clock cycle.

Cho thời gian trễ của các khối như bảng 1.

Resources	Delay
Instruction memory	150ns
Registers	100ns
ALU	100ns
Data memory	150ns
Các khối khác	0ns

Bảng 1: thời gian trễ qua các khối.

Đoạn chương trình sẽ chạy như sau:

```
    addi $t1,$zero,100
    addi $t2,$zero,0

loop:
    beq $t1,$t2,exit
    addi $t1,$t1,-1
    addi $t2,$t2,1
    j loop

exit:
```

- Xác định clock cycle của hệ thống single clock, multi clock và pipeline clock.
- Xác định thời gian thực thi của chương trình trên khi chạy với hệ thống single cycle, multi cycle và pipeline (không kể stall).
- Tính speed up của hệ thống pipeline so với các hệ thống còn lại.
- Khi delay ALU thay đổi từ 100ns sang 150ns. Hãy tính lại kết quả.

Bài 2. Xử lý Hazard.

Dùng lại đoạn code của bài 1.

- Xác định các sự phụ thuộc dữ liệu trong đoạn chương trình.
- Giải quyết data hazard bằng chèn stall (giải quyết bằng phần mềm), khi thực thi đoạn code trên với hệ thống pipeline thì cần chèn vào bao nhiêu stall?
- Dùng cơ chế forward để giải quyết hazard (giải quyết bằng phần cứng), khi đó còn bao nhiêu stall? Vẽ hình minh họa.
- Ngoài 2 cách trên, ta có thể giảm stall bằng cách sắp xếp lại thứ tự lệnh (giải pháp dùng cho trình biên dịch). Áp dụng giải pháp này vào đoạn code trên sẽ còn bao nhiêu stall?

Bài 3. Xử lý Hazard (lệnh load)

Làm lại bài 2 với đoạn code sau:

```
    addi $t1,$zero,100
    addi $t2,$zero,100
    add $t3,$t1,$t2
    lw $t4,0($a0)
    lw $t5,4($a0)
    and $t6,$t4,$t5
    sw $t6,8($a0)
```

Bài tập làm thêm

- Bài tập 4.13 trong textbook.
- Bài tập 4.16 trong textbook.