CO2008 - KIẾN TRÚC MÁY TÍNH

Khoa Khoa học và Kỹ thuật Máy tính
Đại học Bách Khoa – Tp.HCM

9/2017

Bài thực hành số 7
KIẾN TRÚC MIPS: PIPELINE

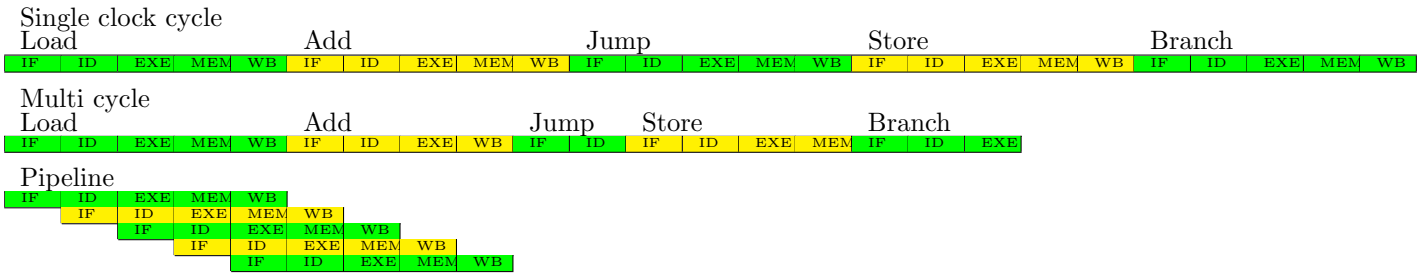
Mục tiêu

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

Yêu cầu

- Xem slide về pipeline.

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



Các bước hiện thực lệnh MIPS

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

- **IF**: Lấy lệnh, 32bits lệnh chứa các thông tin của 1 lệnh được lấy ra từ instruction memory.
- **ID**: Giải mã lệnh, xác định loại lệnh, toán tử, các tín hiệu điều khiển, nội dung các thanh ghi, giá trị immediate ...
- **EXE**: Thực thi tác vụ lệnh.
- **MEM**: Truy xuất vùng nhớ - chỉ dùng cho lệnh **load/store**.
- **WB**: Ghi kết quả vào thanh ghi.

Bài tập và Thực hành

Bài 1: Xác định clock cycle

Cho thời gian delay của các khối như Bảng 1

Bảng. 1: delay của các khối phần cứng

Phần cứng	Delay (ns)
Instruction memory	150
Register	100
ALU	100
Data memory	150
Các bộ phần cứng khác	0

Xét đoạn chương trình như sau:

```
        addi $t1, $0, 10
        addi $t2, $0, 0
loop:
        beq  $t1, $t2, exit
        addi $t1, $t1, -1
        addo $t2, $t2, 1
        j loop
```

(a) Xác định clock cycle của hệ thống single clock, multi clock và pipeline clock.

(b) Xác định thời gian thực thi của chương trình trên khi chạy với hệ thống single cycle, multi cycle và pipeline cycle(không xét stall).

(c) Tính speed up của hệ thống pipeline với hệ thống multi cycle và với single cycle.

(d) Khi delay ALU thay đổi từ 100 –> 150. Tính lại kết quả câu a,b,c

Bài 2: Xử lý Hazard.

Dùng lại đoạn code của **Bài 1**:

(a) Xác định sự phụ thuộc dữ liệu trong đoạn chương trình trên.

(b) 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 (khựng lại) ?

(c) Dùng cơ chế forward để giải quyết hazard (giải quyết bằng phần cứng), khi đó có bao nhiêu stall? Vẽ hình minh họa.

(d) Ngoài 2 cơ chế ở trên, ta có thể giảm stall bằng cách sắp xếp lại thứ tự code (giải quyết bằng trình biên dịch compiler). Hãy sắp xếp lại code sao cho ít stall nhất.

Bài 3: Xử lý Hazard (lệnh load)

Cho 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)
```

Trả lời câu hỏi trong **Bài 2**:

Bài tập TextBook

4.13, 4.16