



Bài tập/Thực hành 7
CHƯƠNG 4 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 Hazard do quá trình pipeline gây ra.

Yêu cầu

- Xem slide về pipeline.
- Nộp các file code hợp ngữ đặt tên theo format «lab7.[txt,pdf]» (ví dụ lab7_1.[txt,pdf], lab7_2.[txt,pdf]) và chứa trong folder lab7_MSSV.

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



- **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 giữa màu xanh và vàng là bằng nhau.
- **Multi Clock Cycle:** 1 lệnh thực thi trong nhiều chu kỳ. Ví dụ lệnh Load thực thi trong 5 chu kỳ (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 5 chu kỳ, các lệnh còn lại sau mỗi chu kỳ hoàn thành xong một lệnh.

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 (khối Instruction Memory), 32bits lệnh chứa các thông tin của 1 lệnh được lấy ra từ instruction memory.
2. **ID:** Giải mã lệnh (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 (khối ALU).
4. **MEM:** Truy xuất vùng nhớ (khối Data Memory) - chỉ dùng cho lệnh `load/store`.
5. **WB:** Ghi kết quả vào thanh ghi (khối Registers).

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:

```
1      addi $t1, $zero, 100
2      addi $t2, $zero, 0
3  loop:
4      beq  $t1, $t2, exit
5      addi $t1, $t1, -1
6      addi $t2, $t2, 1
7      j    loop
```

- Xác định thời gian một clock 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 cycle (không xét stall).
- Tính speed up của hệ thống pipeline với hệ thống multi cycle và với single cycle.
- Khi delay ALU thay đổi từ 100 \rightarrow 150. Tính lại kết quả câu a,b,c

Bài 2: Xử lý Hazard.

Dùng lại đoạn code của [Bài 1](#):

- Xác định sự phụ thuộc dữ liệu trong đoạn chương trình trên.
- 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) ?
- Dùng cơ chế forward để giải quyết data hazard (giải quyết bằng phần cứng), khi đó có bao nhiêu stall? Vẽ hình minh họa.
- Dùng cơ chế forward, stall, để giải quyết hazard (control và data), khi đó có bao nhiêu stall?
- 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:

```
1 addi $t1, $zero, 100
2 addi $t2, $zero, 100
3 add  $t3, $t1,  $t2
4 lw  $t4, 0($a0)
5 lw  $t5, 4($a0)
6 and $t6, $t4, $t5
7 sw  $t6, 8($a0)
```

Trả lời câu hỏi trong [Bài 2](#):

Bài tập TextBook

4.13, 4.16