Computer Architecture

Fall, 2024
Week 10
2024.11.01

組別：_____ **簽名：**_____

[group 6]

1. 下列何者為 pipeline 的特性(T/F):
   1) Pipeline 同時會對加速單一任務的 latency，也會增加整體的 throughout
   2) Pipeline rate 會受限於最慢的階段的執行時間
   3) 若要實作 load 的指令，其步驟可以依序拆解成 Instruction fetch -> instruction decode -> address computation -> memory access
   4) 有什麼因素會讓我們無法達到 potential speedup = number pipe stages

Ans:

1) False, 不會加速單一任務的 latency
2) True
3) False，最後要把從 memory 取出來的值寫回去 register，因此 memory access 的下一步為 memory read completion
4) unbalanced stage length, time to fill and drain the pipeline

[group 3]

1. 依照課本設計的 single-cycle processor 和 pipelined processor，假設 cycle time 分別是 880ms 和 200ms ，各自的 CPI 是多少？一個指令要多久完成？ pipelined 的效能是原本的幾倍？

Ans:

CPI: single-cycle = 1, pipelined = 1

指令延遲：single-cycle = 880ms, pipelined = 1000ms

效能為：880/200 = 4.4

[group 4]

2. Give me some reasons why the pipeline is relatively simple in the MIPS architecture.

Ans:

All instructions are of the same length

Just a few instruction formats

Memory operands only in loads and stores

[group 5]

3. 說明 Drawback of Single-Cycle Design

Ans:

Long cycle time for the worst case (**load**):

- Cycle time must be long enough for the load instruction:
  - PC's Clock-to-Q +

- Instruction Memory Access Time +
- Register File Access Time +
- ALU Delay (address calculation) +
- Data Memory Access Time +
- Register File Setup Time +
- Clock Skew

Cycle time for **load** is much longer than needed for all other instructions

[group 9]

4. Pipeline 設計中，R type 指令其實只需要 4 個 cycle 就能完成，但為何實際上是用 5 個 cycle 來執行?

Ans:

為了配合 lw、sw 指令，同一個元件每個 cycle 只能有一個指令使用，因此會加入一個 NOP stage 讓指令各 stage 對齊 lw、sw 避免衝突發生。

[group 13]

5. “If we use pipeline instead of sequential tasks, will the actual speedup equal the number of pipeline stages? If not, what factors will affect the speedup?”

Ans:

No, the actual speedup will be reduced due to varying stage lengths and the time to fill or drain the pipeline.”

[group 14]

6. How many clock cycles does it take to execute a program with N instruction count on S - stage pipeline machine?

Ans:

The total clock cycle = (S –1) + N

[group 8]

7. Answer these questions:

a) What is the concept of a pipeline?

b) Why is pipelining easy to implement in MIPS?

Ans:

a) Pipelining in computer architecture is a technique to increase the efficiency of instruction processing. It involves dividing the execution of an instruction into several stages, with each stage handled by different parts of the processor. This allows multiple instructions to be processed in overlapping phases, improving overall performance.

b) Pipelining is easy to implement in MIPS because:

- All instructions are the same length.
- Just a few instruction formats.
- Memory operands only in loads and stores

[group 11]

8. T/F Question

Please answer with True/False and give brief explanation for your answer

1. Pipelining decreases the time it takes to complete a single task

2. Pipelining improves overall throughput when processing multiple tasks
3. In a non-pipelined system, all stages of the process are working concurrently
4. The speed-up from pipelining is proportional to the number of stages in the pipeline

Ans:

1. False: in pipelining, the time to complete one task remains the same because each task still passes through every stages
2. True: pipelining allows multiple tasks to be worked simultaneously at different stages, leading to faster overall completion of many tasks, which improves throughput
3. False: in a non-pipelined system, only one task is being worked at a time. Each stage waits of the previous stage to finish before starting (tasks are completed sequentially)
4. True: if all stages take roughly the same amount of time and the pipeline is full, the speed-up is proportional to the number of stages. For example, a pipeline with five stages can complete five tasks in the time it would take a non-pipelined system to complete one

[group 12]

9. Give some functional units used in the pipeline datapath at each stage.

Ans:

IF stage: instruction memory, PC adder, etc

ID stage: register file (read ports), main controller, sign extender, etc

Exec stage: ALU, etc

Mem stage: data memory, etc

WB stage: register file's write port, etc

[group 1]

10. Assuming that Instruction Fetch, Instruction Decode, Execute, Memory Access, writing each takes 40ns, 30ns, 50ns, 20ns, 100ns respectively.

Calculate the total time and the number of cycles needed to process N load-word instructions using:

a) A single cycle processor

b) A processor that uses pipeline (assume that each clock cycle length is the length of the longest instruction delay)

Ans:

a) A single cycle processor would process only one instruction at each cycle, so we have IF+ID+EX+MEM+WB = 40 + 30 + 50 +

20 + 100 = 240 ns for each instruction, thus it takes 240*N ns and N cycles to process N load word instructions.

b) For executing a single instruction, it would take 5 cycles to complete.

For executing two instructions it would take 6 cycles to complete.

For executing three instructions it would take 7 cycles to complete.

Thus, N instructions would take (4+N) * 100 ns and 4+N cycles to complete.