Lecture Notes of Computer Architecture

Zhou Fan
ACM Class, Shanghai Jiao Tong University

Contents

1 Pipelining

1.1 Introduction of Pipelining

Pipelining is an implementation technique whereby multiple instructions are overlapped in execution; it takes advantage of parallelism that exists among the actions needed to execute an instruction.[1]

1.1.1 Laundry Example

Suppose we have many loads of clothes to wash, dry and fold.

- Each step (washing, drying and folding) is called a *pipe stage* or a *pipe segment*.
- *Latency* is the total time spent on single task, which is not improved by pipelining.Unbalanced lengths of pipe stages reduces speedup.
- *Throughput* is defined as the number of loads of clothes per minute. It shows how often a load of clothes exits the pipeline.
- The time required between moving an instruction one step down the pipeline is a *processor cycle*. In a computer, this processor cycle is usually 1 clock cycle.

1.1.2 Speedup from Pipelining

- Unbalanced lengths of pipe stages reduces speedup.
- Handover time between pipe stages reduces speedup.

To improve the efficiency of a pipeline, one should balance the length of each pipeline stage. If the stages are perfectly balanced, then the time per instruction on the pipeline processor is equal to (under ideal conditions)

$$\frac{\text{Time per instruction on unpipelined machine}}{\text{Number of pipe stages}}$$

and the throughput of the pipeline is equal to

$$\text{Number of pipe stages} \times \text{Throughput on unpipelined machine}$$

References

[1] John L. Hennessy, David A. Patterson, et al. *Computer Architecture: A Quantitative Approach*, Fifth Edition, 2012.