TÜRK HAVA KURUMU ÜNİVERSİTESİ
HAVACILIK ve UZAY BİLİMLERİ

Computer Architecture: Pipelining

Furkan YILMAZ

Department of Computer Engineering, University of Turkish Aeronautical Association, Ankara.

Email: furkann.yilmaz@icloud.com

How to cite this paper:
Computer Architecture: Pipelining.
Introduction to Computer Engineering.

Received: May 16, 2020

Furkan Yılmaz

Teacher:
Mert ÖZARAR, Ph.D.

Abstract

Pipelining is an implementation technique whereby multile instructions are overlapped in execution; it takes the advantage of parallelism that exists among the actions needed to pexecute an instruction.

Keywords

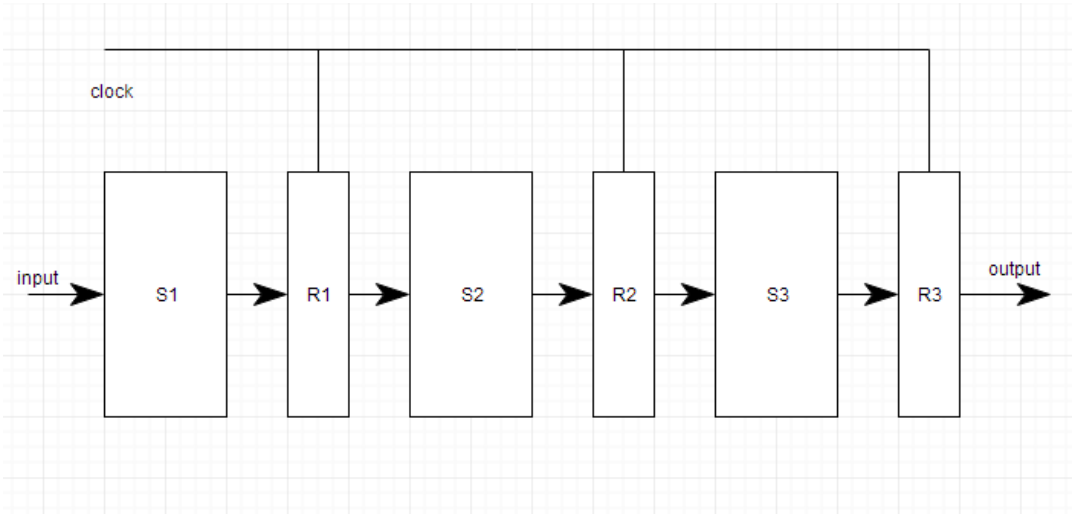
Pipeline, Pipeline Types, Advantages and Disadvantages.

1.Introduction

Pipelining is an implementation technique where multiple instructions are overlapped in execution. The computer pipeline is divided in stages. Each stage completes a part of an instruction in parallel. The stages are connected one to the next to form a pipe, instructions enter at one end, progress through the stages, and exit at the other end.

Pipelining does not decrease the time for individual instruction execution. Instead, it increases instruction throughput. The throughput of the instruction pipeline is determined by how often an instruction exits the pipeline.

- Pipelining is widely used in modern processors.
- Pipelining improves system performance in terms of throughput.
- Pipelined organization requires sophisticated compilation techniques.



2.Types of Pipeline

It is divided into 2 categories

1.Arithmetic Pipeline

2.Instruction Pipeline

Arithmetic Pipeline:

- Pipeline arithmetic units are usually found in every high-speed computers.
- Floating-point operations, multiplication of fixed-point numbers, and similar computations in scientific problem.

Instruction Pipeline:

- Pipeline processing can occur also in the instruction stream. An instruction pipeline reads consecutive instructions from memory while previous instructions are being executed in other segments.

3.Pipeline Conflicts

There are some factors that cause the pipeline to deviate its normal performance. Some of these factors are given below:

- **1.Timing Variations**

All stages cannot take same amount of time. This problem generally occurs in instruction processing where different instructions have different operand requirements and thus different processing time.

- **2. Data Hazards**

When several instructions are in partial execution, and if they reference same data then the problem arises. We must ensure that next instruction does not attempt to access data before the current instruction, because this will lead to incorrect results.

- **3. Branching**

In order to fetch and execute the next instruction, we must know what that instruction is. If the present instruction is a conditional branch, and its result will lead us to the next instruction, then the next instruction may not be known until the current one is processed.

- **4. Interrupts**

Interrupts set unwanted instruction into the instruction stream. Interrupts effect the execution of instruction.

- **5. Data Dependency**

It arises when an instruction depends upon the result of a previous instruction but this result is not yet available.

4. Advantages and Disadvantages of Pipelining

Advantages:

- Instruction throughput increases.
- Increase in the number of pipeline stages increases the number of instructions executed simultaneously.
- Pipelined CPU's works at higher clock frequencies than the RAM.
- Pipelining increases the overall performance of the CPU.

Disadvantages:

- Designing of the pipelined processor is complex.
- Instruction latency increases in pipelined processors.
- The throughput of a pipelined processor is difficult to predict.
- The longer the pipeline, worse the problem of hazard for branch instructions.

Conclusion

Basically the pipelining is used to improve the performance of the system by improving its efficiency. Today, pipelining is the key implementation technique used to make fast CPUs.

References

- https://binaryterms.com/pipelining-in-computer-architecture.html
- https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/pipelining/index.html
- https://en.wikipedia.org/wiki/Pipeline_(computing)