Topics

• Introduction
• Concept of Pipelining
• Pipeline Characteristics
• Software Pipeline Performance And Overhead
• Hardware Pipeline
• Instruction and Data Pipeline
• Software Pipelining

Hardware Pipeline and Performance Improvement
Topics

• When Pipelining Can Be Used

• Pipeline Architectures

• Pipeline vs Parallel

• Pipeline Setup, Stall, and Flush Times

• Superscalar Architecture
Recall

- Earlier parts: on processors, memory, I/O as basic functional units
- Last chapter: parallelism to increase performance
- Earlier parts: on processors, memory, I/O as basic functional units
- How pipelining increases performance?
- Ways of using pipelining
- Motivation

Chapter covers

Introduction
Concept of Pipelining

- A na rchitecture in which digital information flows through a series of processing components.
- Pipelining is a concept used in several situations.
- Illustration of the pipeline concept. The example has four stages, and information flows through each stage.

Software pipelining exists.

Pipelining is a concept used in several situations.

Series of processing components.

An architecture in which digital information flows through a series of processing components.

Concept of Pipelining
Pipeline Characteristics

• Hardware or software implementation
  - Pipelining can be used with hardware or software
    - e.g. of software pipeline: Unix pipe command

• Large or small scale
  - Stages can be simple or powerful
    - e.g. of software pipeline: Linux pipe command
      - Pipelining can be used with hardware or software
        - Hardware or software implementation

Pipeline Characteristics
Pipeline Characteristics

• Synchronous or asynchronous communication
  - Synchronous is like an assembly, all stations simultaneously move data forward;
  - Asynchronous allows forwarding at any time; downside is like an assembly, all stations simultaneously move data forward.

• Buffered or unbuffered pipelines
  - Buffered or unbuffered pipelines

Stalling:

 Pipeline buffering between stages, helpful with asynchronous communication.
Pipeline Characteristics

- **Heterogeneous**: Use hardware suitable for each stage.
- **Homogeneous or Heterogeneous Stages**
  - Simultaneously in parallel
  - Between stages data can be moved in single serial path or
- **Serial or Parallel Path**
  - C.F. separate mechanism to move information
  - Automatic data feed or manual data feed
- ** Finite chunks e.g. Packets**
- Finite chunks or continuous bit streams

Dr. Rajesh Subramanayan, 2005
Software Pipelining

- Allows large complex task to be broken to smaller pieces
- Use Unix pipe `|` command to feed output of one command as input to another command
- Software pipelining does not need underlying hardware
- Does software pipelining perform better or worse than a single program?

Certain cases it can improve performance
- depends

- Software pipelining does not need underlying hardware
- as input to another command
- Unix `|` pipe command to feed output of one command
- Allows large complex task to be broken to smaller pieces
Hardware pipelining offers higher performance.

- Maybe able to reuse pieces of other designs
- Manageable pieces
- Can reduce complexity, break complex tasks to smaller

**Hardware Pipelining**
Instruction and Data Pipeline

- Instruction pipeline
  - Machine instructions are executed using a pipeline, stage to stage.

- Data pipeline
  - Data is passed in pipeline, stage to stage.

- Instruction pipeline
  - Each stage is responsible for a part of the overall fetch-decode-execute cycle most modern processors have
  - Machine instructions are executed using a pipeline
A pipeline used in place of a single processor in an Internet router.

Even if a data pipeline uses the same speed processors as a non-pipeline architecture, a data pipeline has higher overall throughput (i.e., number of data items processed per second).

A data pipeline passes through a series of stages that each examine or modify the data. If it uses the same speed processors as a non-pipeline architecture, a data pipeline will not improve the overall time needed to process a given data item.
When Pipelining Can Be Used?

- It must be possible to partition processing into independent stages.
- Overhead to move data between stages must be insignificant.
- Processing performed at each stage must take approximately the same time as the processing performed at other stages.
- Throughput of pipeline is limited by the slowest stage.

CS250 -- Part V

Dr. Rajesh Subramanyan, 2005
Pipeline vs Parallel

Sequence are divided among stages of the pipeline.

(a) Processing on a conventional processor, and (b) equivalent processing in a data pipeline. The functions performed in (a) processing on a conventional processor, and (b) equivalent processing in a data pipeline.
Pipeline vs Parallel

Pipeline divides a series of sequential operations into separate stages.

Stage S

However, unlike a parallel architecture, data has to pass all stages.

The pipeline allows each stage to operate in parallel.

Parallel divides a series of sequential operations into separate groups that are each handled by a separate stage.
Pipeline Architectures

• The term pipeline architecture is used when it is the central paradigm around which the system is built.

• Pipelined systems are dedicated to special purpose functions, e.g. network systems where high data rates are used.

• General purpose computers restrict pipeline hardware to instruction pipeline in the processor or special purpose pipeline in I/O device.

• Not many applications can be decomposed into independent operations that can be applied sequentially.

Pipeline Architectures
Pipeline Setup, Stall, and Flush Times

- Setup time
  - a mount of time to start a pipeline after an idle period

- Stalls
  - stage delays because it cannot complete processing

- Flush time
  - amount of time between input being unavailable and the pipeline finishes current processing

- Setup time
  - amount of time to start a pipeline after an idle period
Superpipeline Architecture

- Given pipeline stage is subdivided into multiple partial stages
- Example instruction pipeline has
  - Instruction fetch
  - Instruction decode
  - ALU operation
  - Operand fetch
  - Store
Superpipeline Architecture

- Superpipeline increases throughput
- Fetch indirect operand values
- Fetch value from memory
- Fetch immediate value or register value
- Decode operand

The third stage, i.e., operand fetch can be subdivided into...
Summary

- Pipelining is a fundamental concept that is used with both hardware and software.
- Data pipeline does not decrease overall time to process a single data item, it increases overall throughput (items processed/second).
- The pipeline stage requiring the most time to process an item limits the throughput of the pipeline.
- Data pipeline does not decrease overall time to process a single data item, it increases overall throughput.
- Hardware and software.
- Pipelining is a fundamental concept that is used with both hardware and software.