Computer Organization

Douglas Comer

Computer Science Department
Purdue University
250 N. University Street
West Lafayette, IN 47907-2066

http://www.cs.purdue.edu/people/comer

IX

Memory And Storage

Key Aspects Of Memory

- Technology
  - Type of underlying hardware
  - Many technologies available
  - Differ in cost, persistence, performance
- Organization
  - How underlying hardware used to build memory system
  - Directly visible to programmer

Memory Characteristics

- Volatile or nonvolatile
- Random or sequential access
- Read-write or read-only
- Primary or secondary

Memory Volatility

- Volatile memory
  - Contents disappear when power is removed
  - Least expensive
- Nonvolatile memory
  - Contents remain without power
  - More expensive than volatile memory
  - May have slower access times
  - Variants ‘‘cheat’’ by using battery to keep contents

Memory Access Paradigm

- Random access
  - Typical
- Sequential access
  - Special purpose memory
  - Hardware known as *FIFO* (*First-In-First-Out*)

Permanence Of Values

- ROM (Read Only Memory)
  - Values can be read, but not changed
  - Useful for firmware
- PROM (Programmable Read Only Memory)
  - Contents can be altered, but doing so is time-consuming
  - Change may involve removal from a circuit and exposure to ultraviolet light

Permanence Of Values
(continued)

- EEPROM
  - Form of PROM that can be changed while installed
  - Variants such as *Flash ROM* used in digital cameras

Primary And Secondary Memory

- Broad classification of memory technologies
- Terminology is qualitative

Traditional Terminology

- Primary memory
  - Highest speed
  - Most expensive, therefore smallest
  - Typically solid state technology
- Secondary memory
  - Lower speed
  - Less expensive, therefore can be larger
  - Typically magnetic media and electromechanical drive mechanism

In Practice

- Distinction between primary and secondary storage blurred
- Solid state technology replacing electromechanical technology
- Example: *microdrive* that uses flash memory

Memory Hierarchy

- Key concept to memory designer
- Related to definitions of *primary / secondary* memory
- Arise as tradeoff
  - Highest performance memory costs the most
  - Architect chooses set of memories to satisfy both performance and cost constraints

Memory Hierarchy
(continued)

- Small amount of memory has highest performance
- Slightly larger amount of memory has somewhat lower performance
- Large amount of memory has lowest performance
- Example hierarchy
  - Dozens of general-purpose registers
  - Megabytes of main memory
  - Gigabytes of secondary storage

General Principle

To optimize memory performance for a given cost, a set of technologies are arranged in a hierarchy that contains a relatively small amount of fast memory and larger amounts of less expensive, but slower memory.

How Should Memory Be Designed?

- Separate memories, one for programs and another for data
- One memory that holds both programs and data

Instruction Store And Data Store

- Early computers had separate memories known as
  - *Instruction store*
  - *Data store*
- Most modern computers
  - One memory for both instructions and data
- Note single memory design is known as a *Von Neumann architecture*

A Note About Memory Types

- Some special-purpose computers use separate memories
- Motivation: choose memory technology that is optimized for pattern of use
  - Instruction store: typically sequential
  - Data store: typically random

The Fetch-Store Paradigm

- Access paradigm used by memory
- Two operations
  - *Fetch* the value from a specified location
  - *Store* a value into a specified location
- Two operations also called
  - *Read*
  - *Write*
- We will discuss implementation of fetch / store later

Summary

- Two key aspects of memory are
  - Technology
  - Organization
- Memory can be characterized as
  - Volatile or nonvolatile
  - Random or sequential access
  - Permanent or nonpermanent
  - Primary or secondary

Summary
(continued)

- Memory systems use fetch / store paradigm
- Two operations are known as
  - *Fetch* (*Read*)
  - *Store* (*Write*)


Questions?