Memory Hierarchy in Embedded System

Neeraj Goel
IIT Delhi
Introduction

- **What is memory**
  - Any element which can retain a state with time
  - Electronics, magnetic, optical
  - Most simple example is a capacitor

- **Why it is required in embedded system**
  - Instruction memory
  - Data memory
  - Control data : Bitfiles

- **Size of memory**
Type of memory

- **Optical**
  - CDs, DVDs
  - Cheep, large access time, even large write time

- **Magnetic memory**
  - Used in Hard-disks and floppy
  - Cheep, large access (read and write) time

- **DRAM**
  - Capacitor based memory
  - Regular refresh is required (high power consuming)
  - Cheep, large access time
Type of memory (contd.)

- **SRAM**
  - Flip-flop based
  - High cost, low access time

![Diagram with access time and cost per byte axes, showing SRAM and DRAM positions]

- Optical
- Magnetic
- DRAM
- SRAM

Cost per byte

Access time
How to select a memory

- **Parameters**
  - Size, cost, access time, power consumption

- **How to achieve large memory with less access time**
  - Use memory hierarchy
  - Keep most urgently required in costly but fast access memory, keep size of fast access memory less

- **Cache access principle**
  - Spatial locality
  - Temporal locality
Typical Memory Hierarchy in a Computer

- Register File (64x32bits)
- L1 cache (16k)
- L2 cache (128k)
- L3 cache (2MB)
- RAM (1GB)
- Hard-disk (160GB)
Memory Hierarchy in Embedded System

- Application specific requirement
  - Size and performance

- Various example cases
  - Time critical real time system
    - No cache
    - Difficult to estimate performance with cache
  - Graphics application
    - One frame is access at one time
    - Frame memory store one frame
  - Applications that work on specific data
    - Scratch pad memory
    - Compiler controlled
Memory Hierarchy in Embedded System

- **Methodology**
  - No standard method
  - Ad-hoc methods are used

- **Select a memory hierarchy**
  - Analyze the application and its memory accesses
  - Explore to find best size at various level
    - Various open-source simulators are available
    - Simplescalar is widely used one
  - Find best fit with board constraint and performance constraints
Recap

- Various type of memories available (cost and access time)
- Different application requirements
- Different possible memory hierarchy
- Map application to memory hierarchy with given constraints