# CS305
# Computer Architecture

## The Memory System: A Hierarchy of Caches

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

# Memory Systems: Why Important?

- Memory: the second crucial part of a computer
- Today: memory systems dictate performance
  - Processor performance well above memory performance
  - Cannot throw more gates to get faster memory
- Some numbers:
  - Memory access latency: 20+ ns
  - Compare: processor cycle < 1 ns

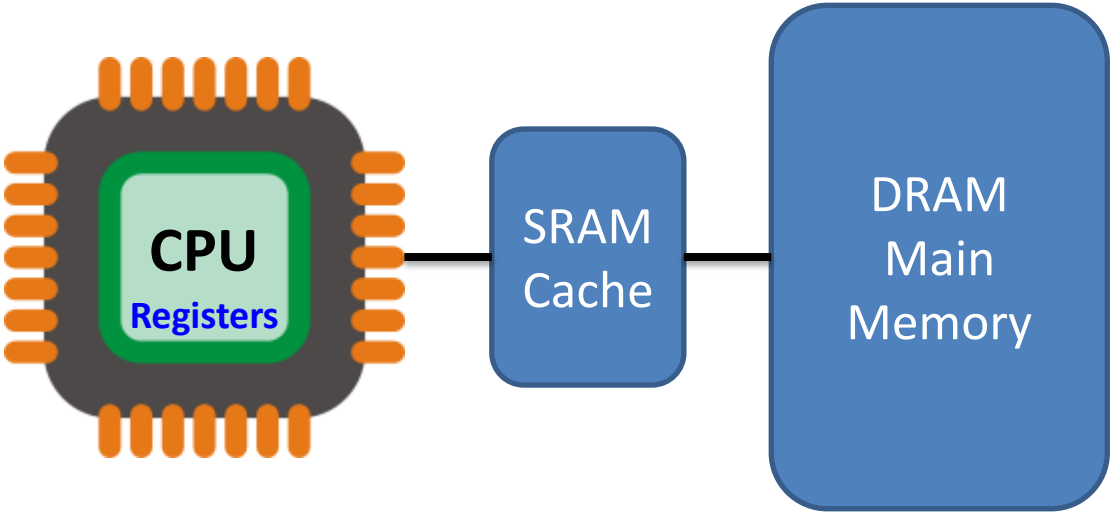
# What Programmer Wants vs Reality

- What programmer wants: large memory, fast, cheap
- Reality: large X fast
  - Large memory ➔ slow
  - Large memory ➔ cost per byte is smaller
- Memory system: create **illusion** of large & fast memory
  - Cache memory, main memory, virtual memory, secondary memory (I/O)

# What is a Cache?

- **Cache (English):** a safe place to store something
- **Cache (CS):** a temporary place for a copy (usually) of something, for fast, easy, efficient access
- Examples of caching you are aware of ?

# Cache in a Computer System



# DRAM versus SRAM

## Dynamic RAM (DRAM)

- Uses less transistors 1
- Needs to refresh periodically
- More power consumption
- Slower: access latency 20+ ns
- Cycle time can be > access time
  - DRAM needs time to refresh
- Cheaper
- Used for main memory

## Static RAM (SRAM)

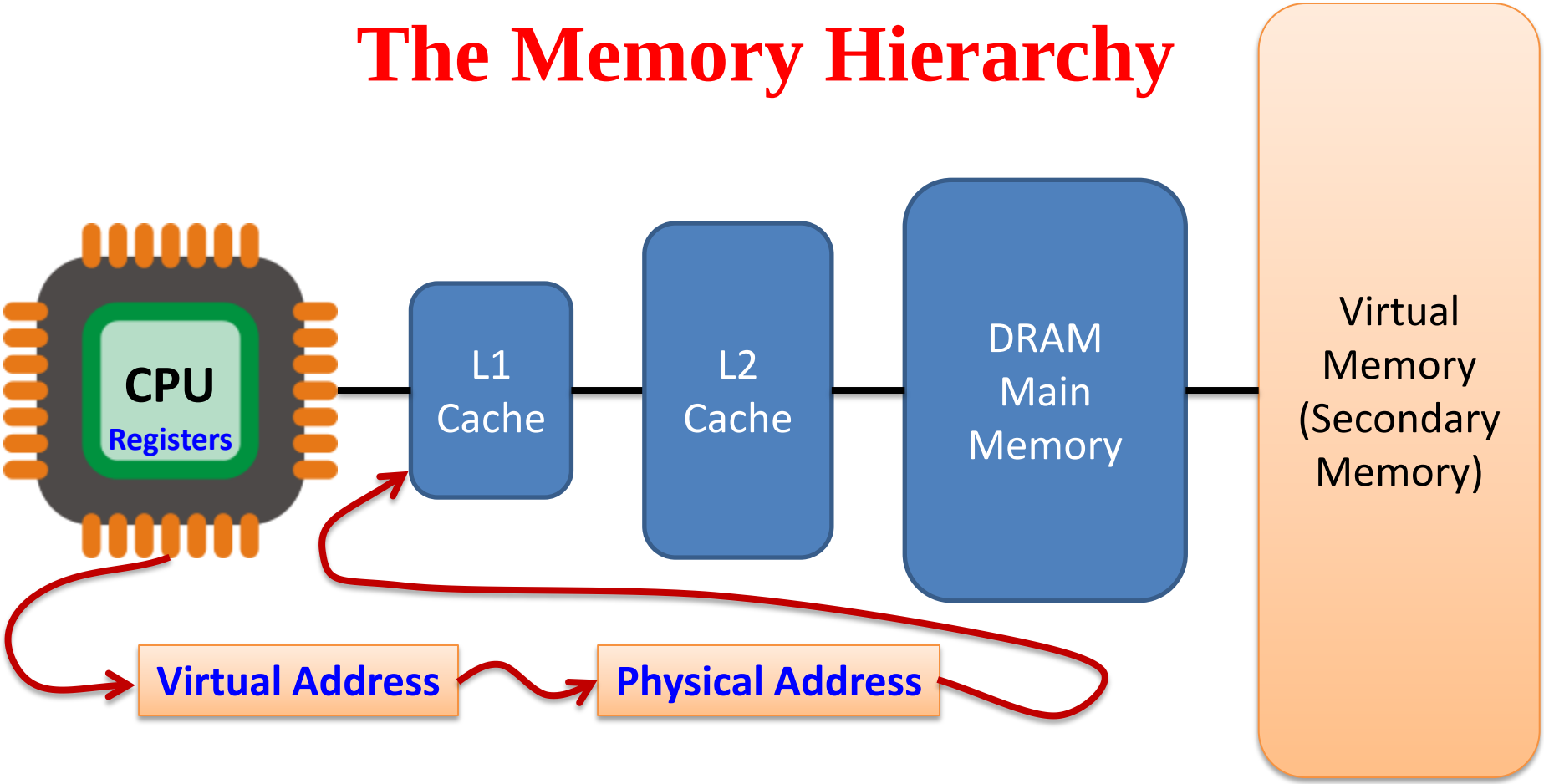
- Uses more transistors 6
- No need to refresh
- Less power consumption
- Faster: access latency ~2 ns
- Cycle time = access time
  - Access one locn. after another
- More expensive
- Used for cache memory

Three reasons why cache is faster: SRAM, smaller, closer to CPU

# Why Caches Work: The Principle of Locality

- **Temporal locality:** if X is accessed now, it will likely be accessed again in the near future
- **Spatial locality:** if X is accessed now, locations $X \pm \delta$ will likely be accessed in the near future
- For instructions: sequential execution, loops
- For data: arrays, structures, variables in a function

# The Memory Hierarchy



# Summary

- Memory system: a hierarchy of caches
- Caching: principle of locality
- Next: cache design