Lecture 2 – Memory Hierarchy Design

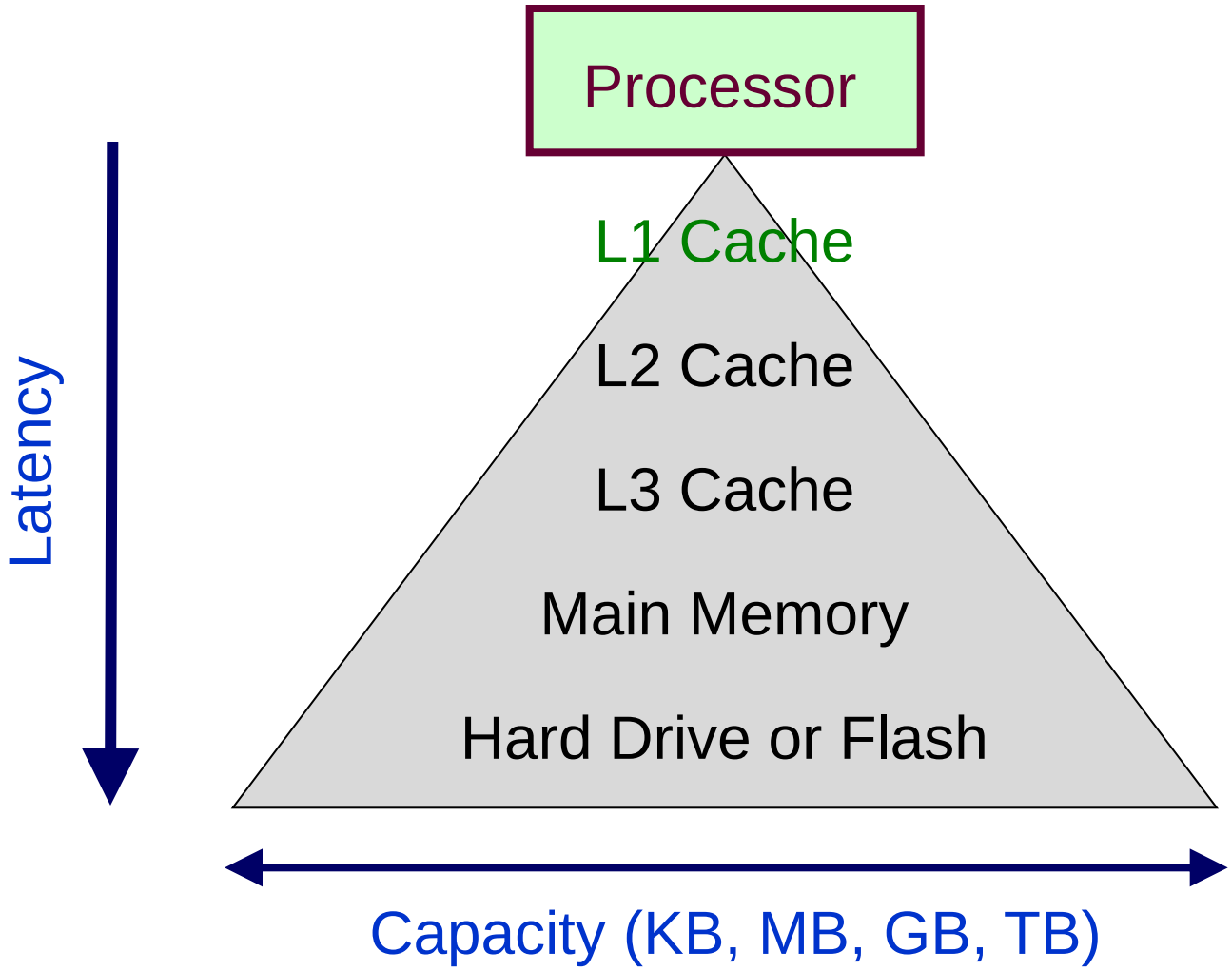
by

Sameer Akram

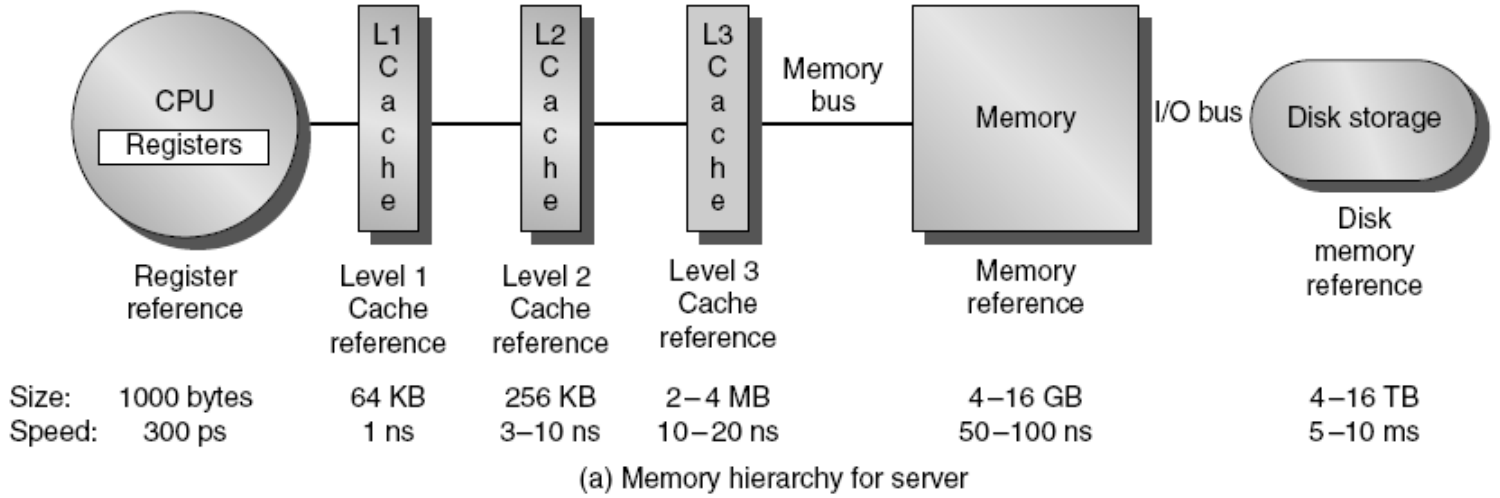
Introduction

- Programmers want very large memory with low latency
- Fast memory technology is more expensive per bit than slower memory
- Solution: organize memory system into a hierarchy
  - Entire addressable memory space available in largest, slowest memory
  - Incrementally smaller and faster memories, **each containing a subset of the memory below it**, proceed in steps up toward the processor
- Temporal and spatial locality insures that nearly all references can be found in smaller memories
  - Gives the allusion of a large, fast memory being presented to the processor

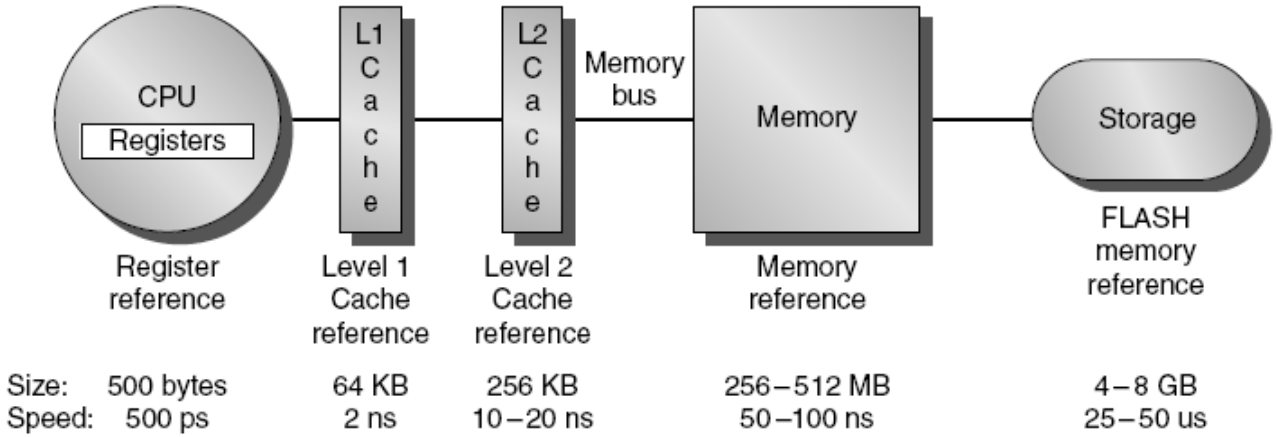
Memory Hierarchy



Memory Hierarchy (contd.)



(a) Memory hierarchy for server



(b) Memory hierarchy for a personal mobile device

Memory Hierarchy (contd.)

Moving from Disk to Registers

- Memory becomes smaller
- Memory becomes expensive
- Memory becomes faster (Memory access time decreases)
- Volatility

Functions of different memories

- **Hard Disk:** For permanent storage because it is non-volatile.
- **RAM:** It is executable memory. Any software which is going to be executed, must take some space in this memory. CPU only generates the address of RAM. It is also called Primary Memory or Main Memory.
- **Cache Memory:** Cache memory speeds up the processing speed, as most recent / frequent instructions and data are stored in it and is used for subsequent use.
- **Register(s):** These are very fast small memories inside the CPU. These are of two types
  - 1) Dedicated Registers: For the exclusive use of CU and cannot accessed by the user program.
  - 2) General Purpose Registers: These are used by users for storing data, addresses and transitional & final results.