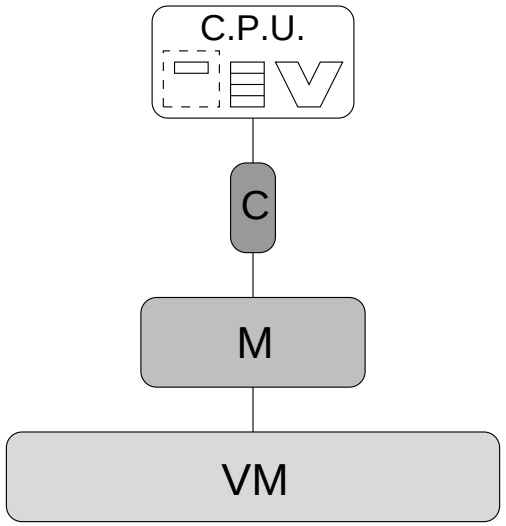
Memory Hierarchy

Ideally we would like large amounts of fast memory. This is neither economically viable nor technically possible.



Type of Storage		Typical Access
Registers		Several Accesses per Cycle
Cache	SRAM	One Access per Cycle
Main Memory	DRAM	Several Cycles per Access
Virtual Memory	Hard Disk	Many Cycles per Access

The solution is a hierarchy of memory containing some large components and some fast components.

Our task is to make it appear that we have large amounts of very fast memory.

Cache Performance

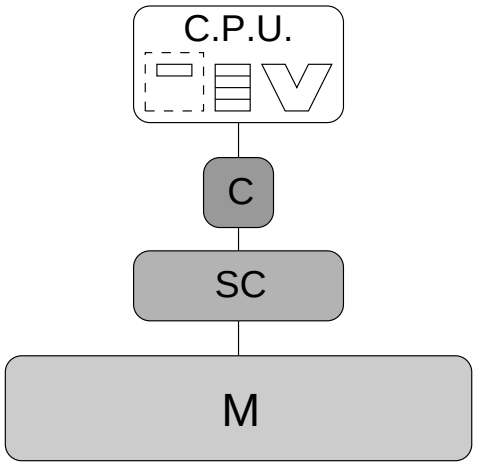
$$Average\ memory\ access\ time = Hit\ time + Miss\ rate \times Miss\ penalty$$

- Hit Time
  Time for a cache hit - may be different for read and write.
- Miss Penalty
  Additional time for a cache miss.
- Miss Rate
  In order to analyse miss rate consider the reasons for a miss:
  - Compulsory Miss
    The required block has never been accessed before.
  - Capacity Miss
    The required block was displaced when the cache was full.
  - Collision Miss
    The required block was displaced by another although the cache was not full.

Summary of Cache Features

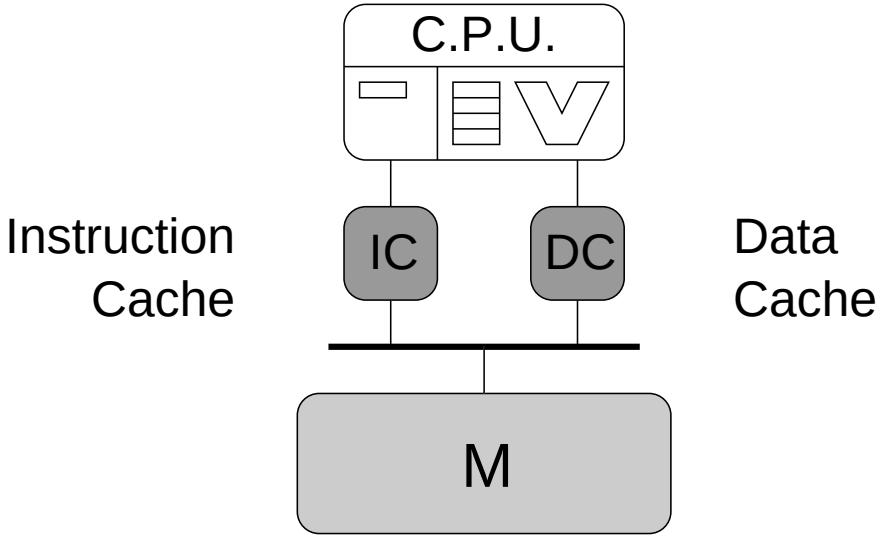
- Capacity
- Block/Line Size
- Storage Strategy
  - Fully Associative – requires associative memory
  - Direct Mapped
  - Set Associative
- Discard Strategy
  - Least Recently Used
  - FIFO
  - Random
  - None – for direct mapped cache
- Write Strategy
  - Write Through
  - Write Back
  - Write Around
  - Fetch on Write

Multi-Level Cache



- First Level Cache
  - Simple cache
  - Optimized for hit time
  - Operates at speed of CPU
- Second Level Cache
  - Large cache
  - Optimized for miss rate

Multiple Caches



Pseudo-Harvard Architecture

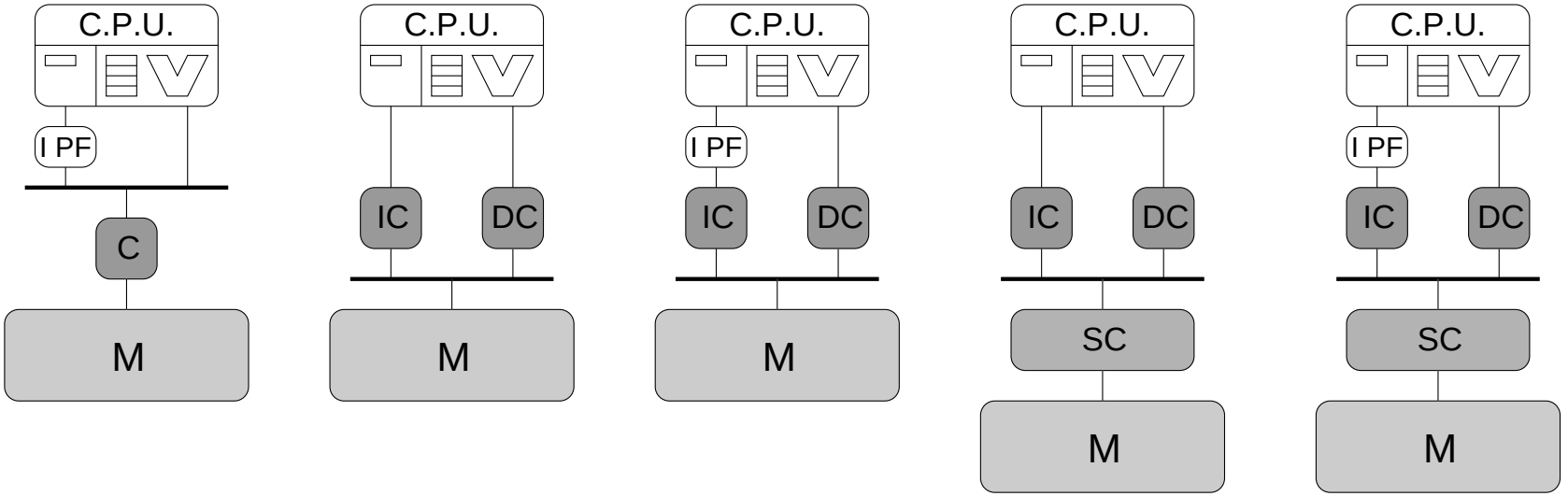
- With multiple caches a machine with an external Princeton architecture may achieve performance approaching that of a Harvard machine.
  - Double CPU to cache bandwidth
  - Separately optimized caches
    Different capacities, block sizes and associativities.
    Instruction caches are read-only with respect to the CPU.

Multiple Caches

- Instruction Pre-Fetch

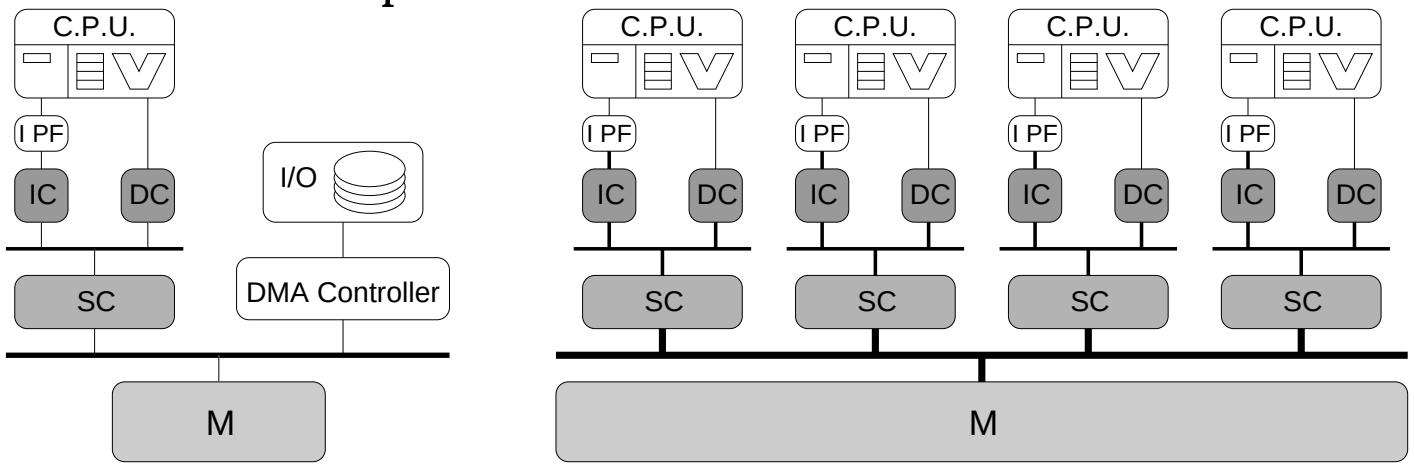
  The pseudo Harvard machine may also make use of an instruction pre-fetch buffer containing one or more blocks of instructions. This buffer reduces the probability of a compulsory miss when the CPU reads a new instruction.

- Cache configurations to support instruction fetch.



Cache Coherence

- Where we have multiple copies of data we must ensure that all copies are the same. This becomes a problem when there are multiple bus masters, e.g. DMA controllers and multiprocessors.



- Bus snooping by cache controllers
  - Discard cache copy?
  - Update cache copy?
- Is it safe to assume that Program cache does not suffer from these problems?