Ch 6.1 - 6.3

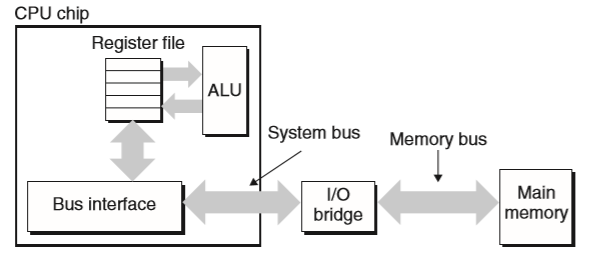
Intro

* memory system
  + heiarchy of storage devices with different data
* cache memory
  + near cpu, act as staging area for data and instructions stored in memory
* learn how to access data faster if needed

6.1 Storage Technologies

6.1.1 Random Access Memory (RAM)

* static and dynamic
* static (SRAM)
  + faster and significantly more expensive
  + cache memory
* dynamic (DRAM)
  + main memory
  + frame buffer of graphics system
* static RAM
  + stores bits in bistable memory cell, transistor
  + stable cells, either one side or the other
  + not sensitive to disturbance
* dynamic RAM
  + stores as charge on capacitor
  + sensitive to disturbance
* conventional DRAM
  + d supercells
  + w DRAM cells
  + d x w DRAM stores dw bits
  + memory controller
    - transfer w bits at a time to and from each chip
    - row i and column j
    - i is called RAS (row access strobe)
    - j is called CAS (column access strobe)
    - share same access pin
    - internal row buffer, where row is copied, then smaller item is extracted
* memory modules
  + DRAM packed in memory module
  + plug into expansion slots on motherboard
  + has 8 DRAM chips, which all get one byte and return 64-bit word
* enhanced DRAMs
  + fast page mode DRAM (FPM DRAM)
    - conventional DRAM copies rows to internal buffer row
    - FPM DRAM allows consecutive accesses to same row
  + extended data out DRAM (EDO DRAM)
    - FPM that allows individual CAS signals to be spaced closer together
  + synchronous DRAM (SDRAM)
    - outputs data at faster rate
  + double data-rate synchronous DRAM (DDR SDRAM)
    - doubles speed by using both clock edges
    - DDR (2 bits), DDR2 (4 bits), and DDR3 (8 bits)
  + video RAM (VRAM)
    - similar to FPM DRAM
    - VRAM output produced by shifting entire contents of internet buffer sequence
    - concurrent read and write to memory
* nonvolatile memory
  + DRAMs and SRAMs are volatile
    - lose info if supply of voltage is turned off
  + read only memories (ROM)
    - programmable ROM (PROM)
      * programmed once
      * blow fuse once written
    - erasable programmable ROM (EPROM)
      * can be reprogrammed with UV light shined on quartz window
      * also electronically EPROM
  + flash memory
    - based on EEPROMs
  + programs on ROM are called firmware
    - computer runs firmware on startup
* accessing main memory



* + data flows between processor and DRAM over buses
  + bus transaction
    - read transaction transfers data from main memory to CPU
    - write transaction transfers data from CPU to main memory
  + bus
    - parallel wires that carry address, data, control signals
  + I/O bridge translates electrical signals of system bus to electrical signals of memory bus
    - also connects system bus and memory bus to I/O bus
  + movq A, %rax
  + reading
    - 1. CPU places address A on system bus
    - 2. main memory senses address signal on memory bus, reads address from memory bus, fetches data from DRAM, and writes data to memory bus
    - 3. I/O bridge translates memory bus signal into system bus signal and passes it along to system bus
    - 4. CPU senses data on system bus, reads data from bus, and copies to %rax
  + movq %rax, A
  + contents of %rax written to address A
  + write
    - 1. CPU places address on system bus
    - 2. memory reads address from bus and waits for data to arrive
    - 3. CPU copies data in %rax to system bus
    - 4. main memory reads data from memory bus and stores bits in DRAM

6.2 Locality

Intro

* reference data near other recently referenced data items
* temporal locality
  + memory location that is referenced once is likely to be referenced again multiple times in the near future
* spatial locality
  + memory location referenced once, then program is likely to reference nearby memory location in near future
* exploit locality
  + cache memories in hardware

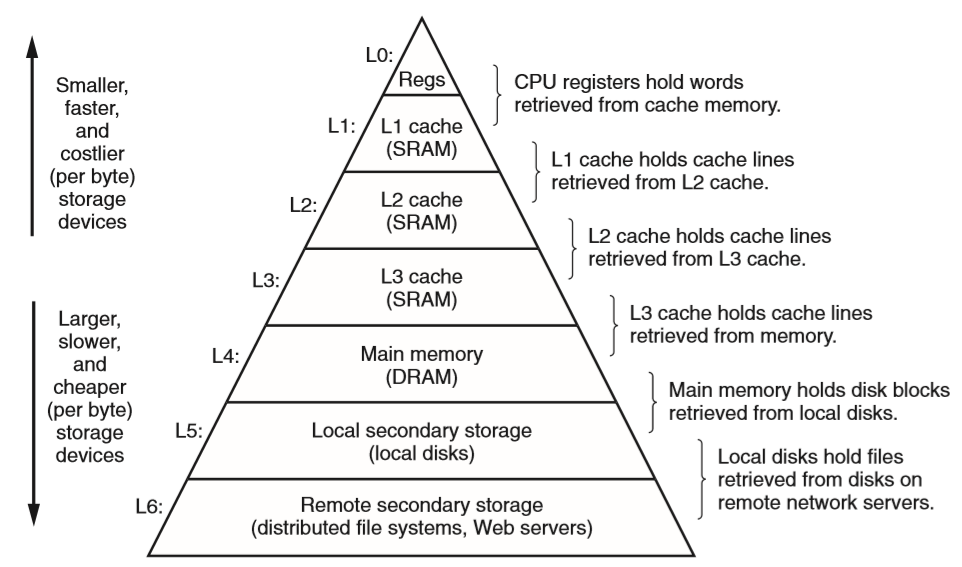
6.2.1 Locality of References to Program Data

* summing values in array
  + good spatial locality, bad temporal locality
  + stride-k pattern
    - visiting kth element of contiguous vector
    - as stride increase, spatial locality decreases

6.2.2 Locality of Instruction Fetches

* loops are good
  + good temporal and spatial locality

6.3 The Memory Hierarchy



6.3.1 Caching in the Memory Hierarchy

* cache
  + small fast storage device for staging data objects
* cache hits
  + getting data from cache
* cache misses
  + not at level k
  + possibly overwrite block at k with info at k + 1