* Dr. Sudhakar Ganti
* Download free textbook
* Set up MS Teams for course communication
* [sganti@uvic.ca](mailto:sganti@uvic.ca) (avoid using this)
* Office hours
  + Monday Thursday 3:30 - 5:00pm
  + Zoom
* Labs: week of May 15th
* Harddrive, processors, cd drives, cpu, etc are all connected through “buses”
* Instruction sets
  + Making the hardware work
* RISC and CISC
  + Reduced instruction set computer
  + Complex instruction set computer
* To save costs, computers have a Memory Heirarchy because memory is expensive
  + 1. Cache
    - CPU
  + 2. Cache 2
  + 3. Main Memory
* Everything is tied to performance
* Compiler
* Assembly language directly uses the Instruction Set
* Microsoft Teams
  + Summer 2023 CSC 230 A01x A02x
* Categories of computers
  + Personal or microcomputers
    - Desktops
    - Notebooks
    - Laptops
  + Handheld
    - Smartphones
    - PDAs
  + Mainframes
  + Supercomputers
  + Cloud computing environments
    - Amazon, google, IBM
* MIPS: millions of instructions per second
* FLOPS: floating point operations
  + Gigaflops
  + Teraflops
* Applications, browser, etc
  + Operating system
    - Device drivers
      * Hardware
* CPU: Central Processing Unit
* Memory: two kinds
  + RAM (random access memory)
  + ROM (read only memory)
* Storage devices and media
  + Hard disk
  + SSD, Flash storage
  + DVD, DVD-ROM, DVD-RW, etc
* Hardware
  + BUS: common communication path
    - Allows multiple devices to communicate
  + CPU
  + Memory
  + Video
  + Keyboard Controller
  + Touchpad Controller
  + SSD Controller
* Chipset defines the major functionality of the CPU system
  + Video, memory, etc
* Embedded systems don’t need an operating system. For general purpose computing, you do
  + Every general purpose processor uses the **fetch-decode-execute** cycle
    - Cpu will always fetch the instruction from the memory
    - This cycle is driven by **clock pulses** called hertz
      * Giga is 10^9
      * Gigahertz = 10^9 cycles
  + Inside the CPU (not just one piece: 3)
    - Step 1: RAM
      * fetch
    - Step 2: Control unit
      * Fetches instructions from memory, decoding
    - Step 3: Arithmetic / logic unit ALU
    - Step 4: RAM (again)
      * Execute
* Timing (clock)
  + Nearly all computers depend upon some sort of timing signal or clock
  + 1 cycle:
    - Fetch
    - Decode
    - execute
  + **Pipeline** staggers the cycles to increase speed