* + Topic: Techniques that processor designers use to to boost performance of computer operations
  + Instruction sets
    - Expensive operations like divide, graphics, operations, encryption, etc. are implemented into the the ALU hardware which does make design harder, but makes the system more capable
    - Newer processors tends to keep adding more and more to the instruction set
  + Cache
    - Small piece of RAM on the CPU
    - Requests a memory location from RAM, the RAM can transmit not just one single value, but a whole block of data.
    - Instead of going back and forth between RAM, we have a small chunk of saved data that is much easier to access
      * Cache hit if request data is already stored in cache, cache miss otherwise
    - Also can be used as scratch space for intermediary values when doing large computation
  + Dirty Bit
    - Data is saved onto cache copy since it is faster than going to RAM immediately
      * Ends up being mismatch with RAM data, so we have a dirty bit to determine when we need to update the RAM data when we free the cache
  + Instruction Pipeline
    - Parallelization across pipeline in order to improve throughput
    - Fully CPU is being utilized
    - Dependencies of instruction could cause issues
      * pipelined processors have to look ahead for data dependencies, and if necessary, stall their pipelines to avoid problems
      * Out-of-order execution - dynamically reorder instructions with dependencies in order to minimize stalls and keep the pipeline moving
    - Conditional Jump operations
      * Speculative execution - guess which way they are going to go, and start filling their pipeline with instructions based off that guess
        + If wrong guess, flush the pipeline
        + Branch prediction
  + Multicore and redundancy
    - Taking advantage of parts of CPU not in use and also duplicating popular components for better performance