1 Course Overview

- Even though Moore's Law[1] is still valid, heat and power are of primary concerns.
  - These challenges can be overcome with smaller and more efficient processors or simply more processors
  - To make better use of the added computation power, parallelism is used.
- Parallel vs. Concurrent: In both cases, one of the difficulties is to actually determine which processes can overlap and which can't:
  - Concurrent: Focus on which activities may b executed at the same time (= overlapping execution)
  - Parallel: Overlapping execution on a real system with constraints imposed by the execution platform.
- Parallel/Concurrent vs. distributed: In addition to parallelism/concurrency, systems can actually be physically distributed (e.g. BOINC)
- Concerns in PP:
  - Expressing Parallelism
  - Managing state (data)
  - Controlling/coordinating parallel tasks and data

2 Parallel Architectures

- Turing machine:
  - Infinite tape
  - Head that reads/writes symbols on tape
  - State registers
  - Program is expressed as rules: (reg)(head) $\rightarrow$ (reg)(head)(movement)
- Today's computers:
  - Consist of CPU, memory and I/O
  - Stored Program: program instructions are stored in memory
  - Von Neumann Architecture: Program data and program instruction in the same memory
- Since accessing memory became slower than accessing CPU registers, CPUs now have caches which are closer (faster and smaller) to the CPU. Caches are:
  - Faster then memory
  - Smaller than memory

[1] "The number of transistors on integrated circuits doubles approximately every two years"

    - Organized in multi-level hierarchies (e.g. L1,L2,L3)

- To improve sequential processor performance, you can use the following parallelism techniques:
    - Vectorization
      For example, when adding vectors (load → operation(s) → store)
        - Normal: 1-at-a-time
        - Vectors: N-at-a-time (bigger registers)
    - Pipelining[2]

      maybe add diagram from slides?
        - Multiple stages (CPU Functional Units)
            - Instruction Fetch
            - Instruction Decode
            - Execution
            - Data access
            - Writeback
        - Each instruction takes 5 time units (cycles)
        - 1 instruction per cycle (not always possible though)
    - Instruction Level Parallelism (ILP)
        - Superscalar CPUs
            - Multiple instructions per cycle
            - multiple functional units
        - Out-of-Order (OoO) Execution
            - Potentially change execution order of instructions
            - As long as the programmer observes the sequential program order
        - Speculative execution
            - Predict results to continue execution
- Moore's Law
    - *"The number of transistors on integrated circuits doubles approximately every two years"* - Gordon E.Moore, 1965
    - Actually an observation
    - For a long time, CPU Architects improved sequential execution by exploiting Moore's Law and ILP
    - More transistors → more performance

[2] Think laundry: you can either wash, dry, fold and repeat, or while the n load is drying, the $n+1$ load can start washing