Lecture 5.1
Flynn’s Taxonomy

EN 600.320/420/620

Instructor: Randal Burns

12 February 2018

Department of Computer Science, *Johns Hopkins University*

Why do I care about architecture?

- What’s my machine?
  - What do I need to know about the processors and memory architecture?
- How can I program it?
  - Different classes of machines mandate different tools

- The interaction of architecture and programming environment places many constraints on how best to solve a parallel computing problem

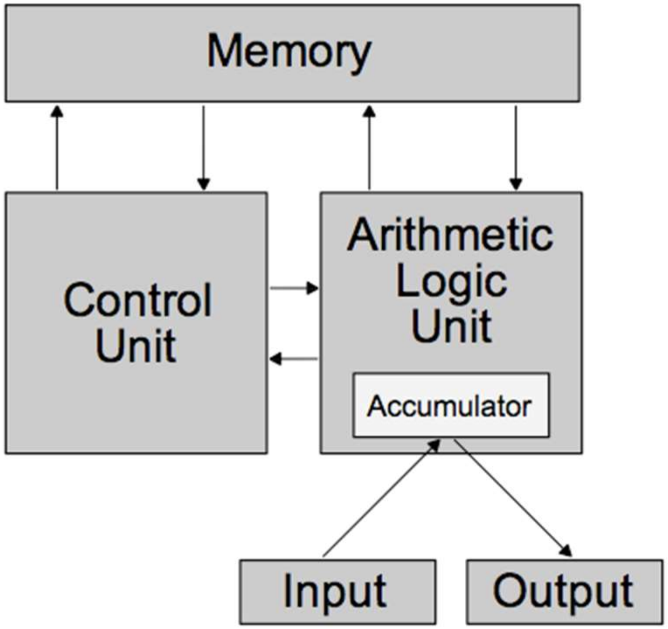
Flynn’s Taxonomy

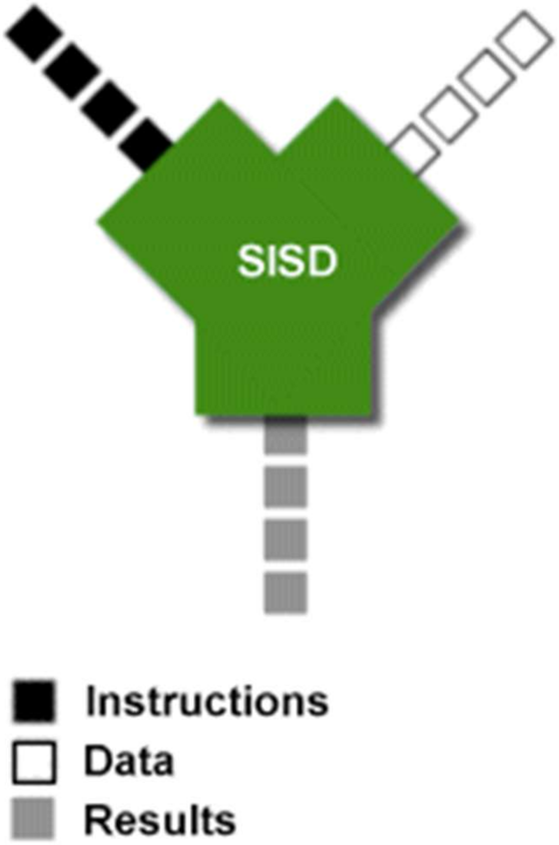
- Characterize machines by number of instruction streams and data streams
  - Defined in 1972. Still common practice.
  - A little too restrictive, but a starting place

- SISD: single instruction, single data
- SIMD: single instruction, multiple data
- MISD: multiple instruction, single data
  - Irrelevant. No such machines.
- MIMD: multiple instruction, multiple data

SISD

- Single instruction, single data
- The von Neumann architecture
  - Implements a universal Turing machine
  - Conforms to serial algorithmic analysis

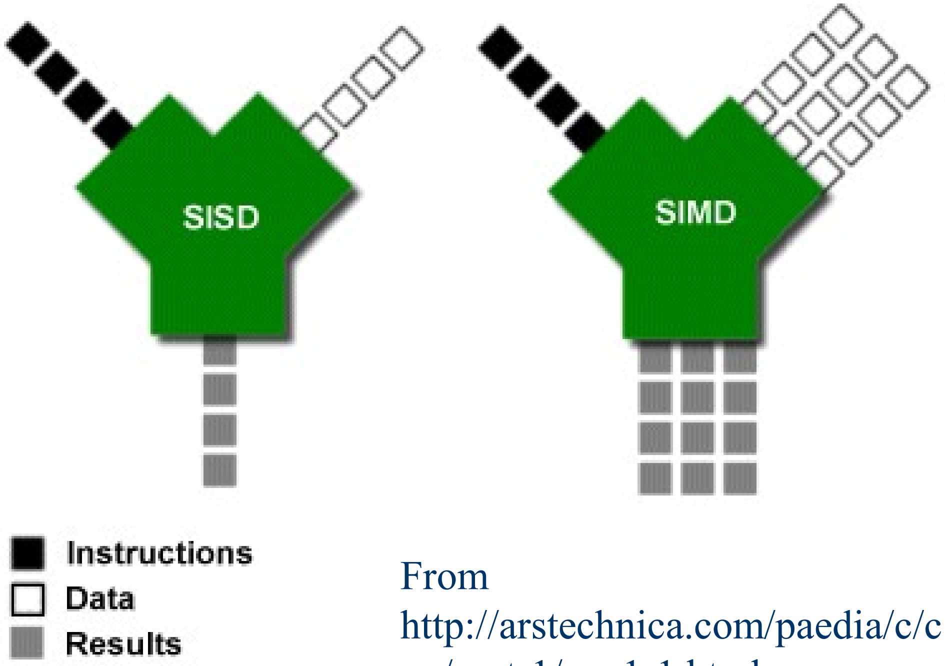




From http://arstechnica.com/paedia/c/cpu/part-1/cpu1-1.html

SIMD: Single Instruction, Multiple Data

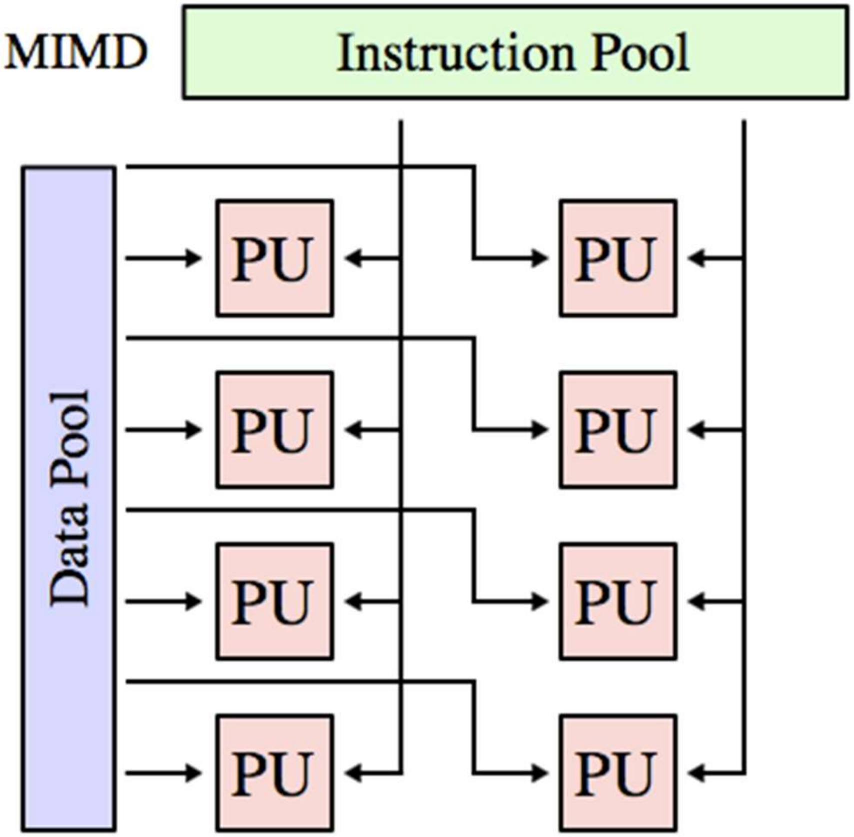
- Single control stream
  - All processors operating in lock step
  - Fine-grained parallelism without inter-process communication
- Examples
  - Intel vector processors
  - GPU stream processor
    - Not the whole card



From http://arstechnica.com/paedia/c/cpu/part-1/cpu1-1.html

MIMD: Multiple Instructions, Multiple Data

- Most the machines we are interested in
  - Multi-core, SMP, Clusters, ccNUMA, etc.
- Flynn's taxonomy not so useful
  - Must further divide the world
  - By architectural features and programming model


MIMD
Instruction Pool
Data Pool
PU
PU
PU
PU
PU
PU
PU
PU