Eight Great Ideas in Computer Architecture

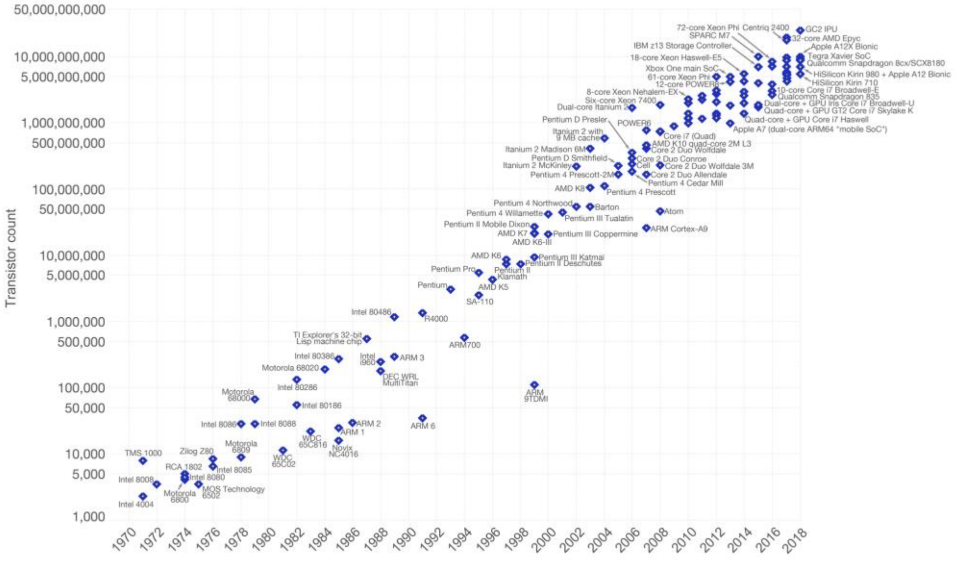
Gordon Moore

- **Design for Moore's Law**
  - The one constant for computer designers is rapid change, which is driven largely by **Moore's Law**. It states that integrated circuit resources double every 18–24 months.
  - Moore's Law resulted from a 1965 prediction of such growth in IC capacity made by Gordon Moore, one of the founders of Intel.
  - As computer designs can take years, the resources available per chip can easily double or quadruple between the start and finish of the project. Computer architects must anticipate where the technology will be when the design finishes rather than design for where it starts.

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Our World in Data

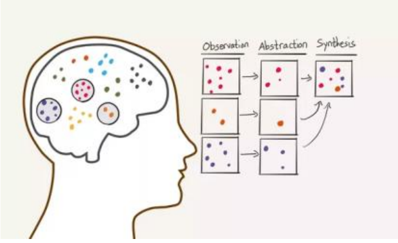
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

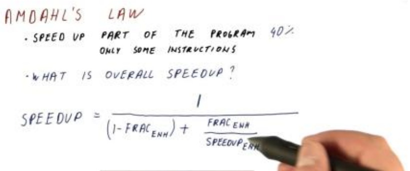
Eight Great Ideas in Computer Architecture

- **Use Abstraction to Simplify Design**
  - Both computer architects and programmers had to invent techniques to make themselves more productive, for otherwise design time would lengthen as dramatically as resources grew by Moore's Law.
  - A major productivity technique for hardware and software is to use **abstractions** to represent the design at different levels of representation; lower-level details are hidden to offer a simpler model at higher levels.



Eight Great Ideas in Computer Architecture

- **Make the common case fast**
  - Amdahl's Law
  - This will tend to enhance performance better than optimizing the rare case.
  - Ironically, the common case is often simpler than the rare case and hence is often easier to enhance. This common sense advice implies that you know what the common case is, which is only possible with careful experimentation and measurement.


AMDAHL'S LAW
• SPEED UP PART OF THE PROGRAM 40%
ONLY SOME INSTRUCTIONS
• WHAT IS OVERALL SPEEDUP ?
$SPEEDUP = \frac{1}{(1-FRAC_{ENH}) + \frac{FRAC_{ENH}}{SPEEDUP_{ENH}}}$

Gene Amdahl

Eight Great Ideas in Computer Architecture

- **Performance via Parallelism**
  - Since the dawn of computing, computer architects have offered designs that get more performance by performing operations in parallel.

Eight Great Ideas in Computer Architecture

- **Performance via Pipelining**
  - A particular pattern of parallelism is so prevalent in computer architecture that it merits its own name:
  - **pipelining**.
    - For example, before fire engines, a "bucket brigade" would respond to a fire. A human chain was used to carry a water source to fire, as they could much more quickly move buckets up the chain instead of individuals running back and forth.

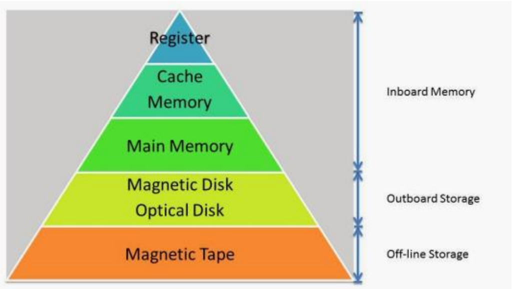
Eight Great Ideas in Computer Architecture

- **Performance via Prediction**
  - Sometimes it can be better to ask for forgiveness than to ask for permission.
  - In some cases it can be faster on average to guess and start working rather than wait until you know for sure, assuming that the mechanism to recover from a misprediction is not too expensive and your prediction is relatively accurate.

41

Eight Great Ideas in Computer Architecture

- **Hierarchy of Memories**
  - Programmers want memory to be fast, large, and cheap, as memory speed often shapes performance, capacity limits the size of problems that can be solved, and the cost of memory today is ofen the majority of computer cost.
  - Architects have found that they can address these conflicting demands with a **hierarchy of memories**, with the fastest, smallest, and most expensive memory per bit at the top of the hierarchy and the slowest, largest, and cheapest per bit at the bottom.



42

Eight Great Ideas in Computer Architecture

- **Dependability via Redundancy**
  - Computers not only need to be fast; they need to be dependable.
  - Since any physical device can fail, we make systems **dependable** by including redundant components that can take over when a failure occurs *and* to help detect failures.

43

Computer Organization

- A digital computer consists of an interconnected system of processors, memories, and input/output devices.
- We will begin by introducing to these three components and to their interconnection, as background for the examination of the specific levels in our abstract machine hierarchy
- Processors, memories, and input/output are key concepts that are present at every level, so we will look at all three in turn

44