Course Introduction and Overview

Aziz Qaroush

Course Info

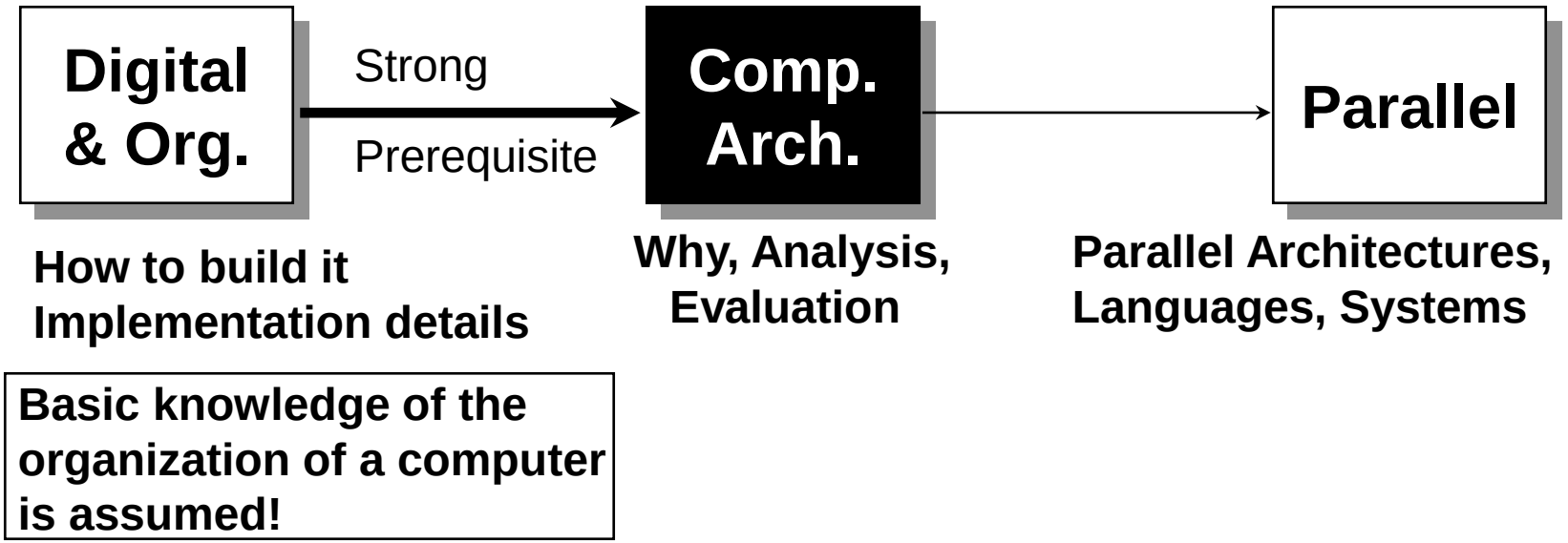
- **Main Textbook (required)**
  - Computer Organization and Design: The Hardware/Software Interface, John L. Hennessy and David A. Patterson, Morgan Kaufmann, 6th **Edition (2021)**
- **Supplement Text**
  - Computer Architecture a Quantitative Approach**,** John L. Hennessy, David A. Patterson, Morgan Kaufmann 6th **Edition, 2019**.
  - Modern Processor Design, John Paul Shen and Mikko H. Lipasti, McGraw-Hill, 2005.

- **Grading**
  - 15% Quizzes
  - 20% Midterm Exam
  - 40% Final Exam
  - 25% Projects

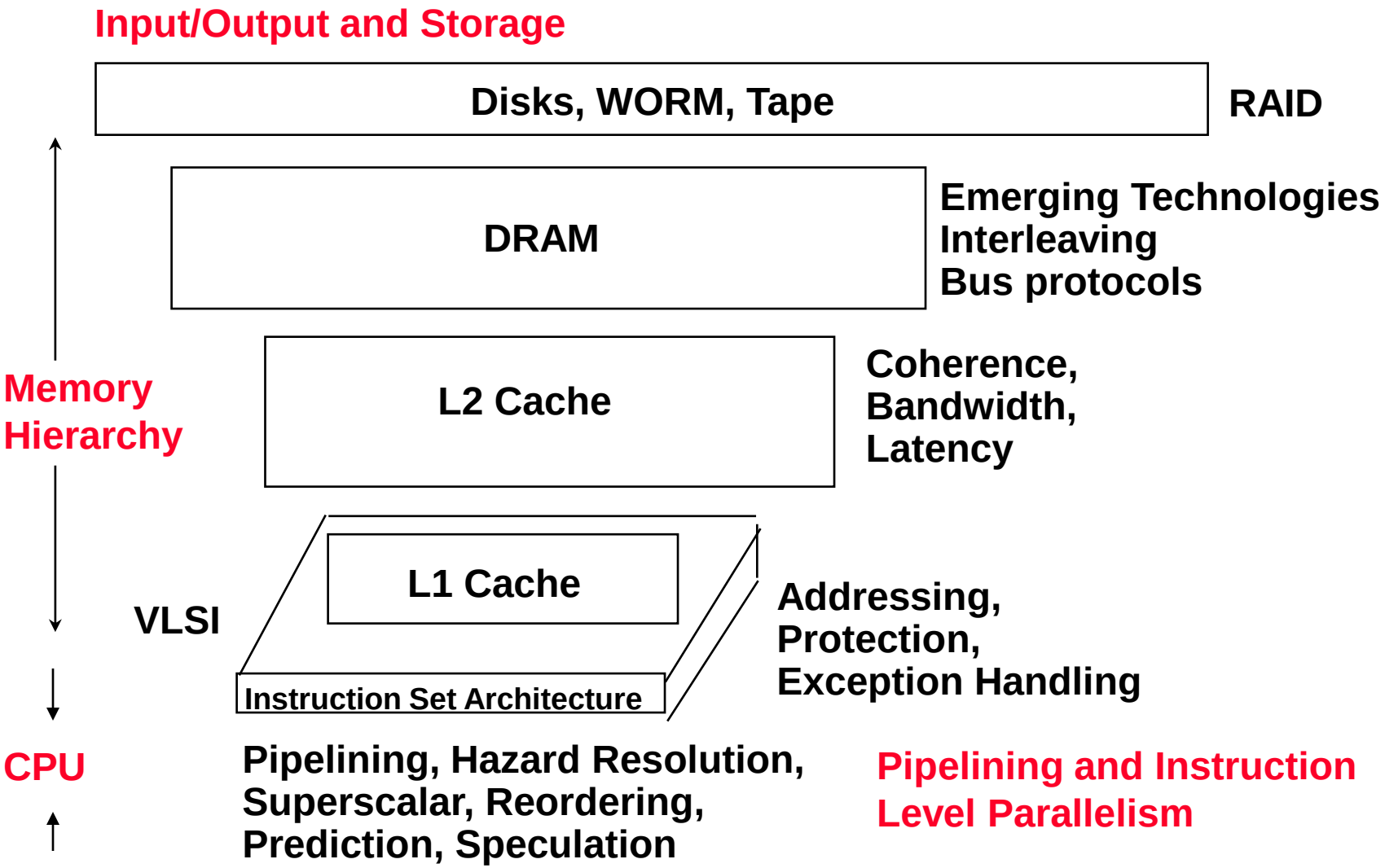
Course Schedule

Topics	Expected Time	Materials
Introduction	2 lectures	T.B Ch. #1
Instructions Set Architecture	3 Lectures	T.B Ch. #2
Computer Performance and Cost • **Benchmarks & simulations** • **Analysis (CPU execution time)** • **Amdal's Law**	5 Lectures (2 + 2 + 1)	T.B Ch. #4 + R.B Ch. #1
CPU Organization & Design • **Building Datapath and Control unit for** **1. Single cycle scheme** **2. Multi-Cycle scheme** **3. Pipelining scheme**	12 Lectures (5+2+5)	T.B Ch. #5 + T.B Ch. #4
Instruction-Level Parallelism & Superscalar Processors • **Dynamic Branch Prediction** • **Dynamic scheduling** • **Hardware-Based Speculation** • **Superscalar Processors**	9 Lectures (3 + 2 + 2 + 2)	R.B Ch. #2
Memory Hierarchy & Cache Design • **Basics of memory hierarchy and caches** • **Measuring and improving cache performance** • **Parallelisms and memory hierarchy** • **Virtual memory**	10 Lectures (2 + 3 + 2 + 2)	T.B Ch. #5 + R.B Ch. #5
Storage & I/O Systems • **Disk structure** • **Interfacing** • **RAID Systems** • **Measuring I/O Performance**	4 Lectures (1 + 1 + 1 + 1)	T.B Ch. #6 R.B Ch #6
Multiprocessor Systems	If time permits	T.B Ch. #7

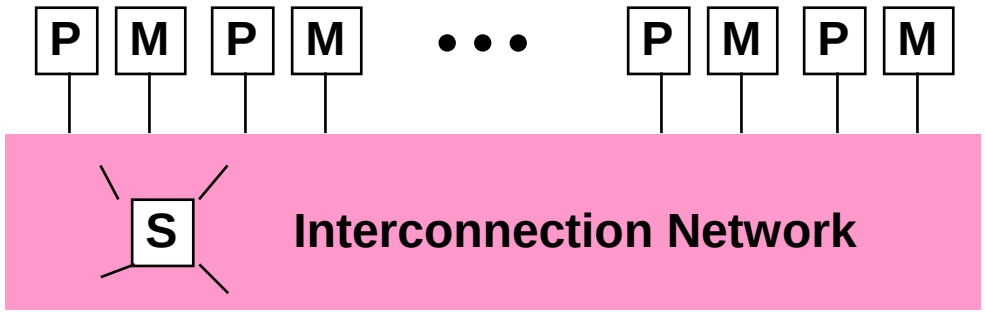
Related Courses



Computer Architecture Topics



Computer Architecture Topics



Processor-Memory-Switch

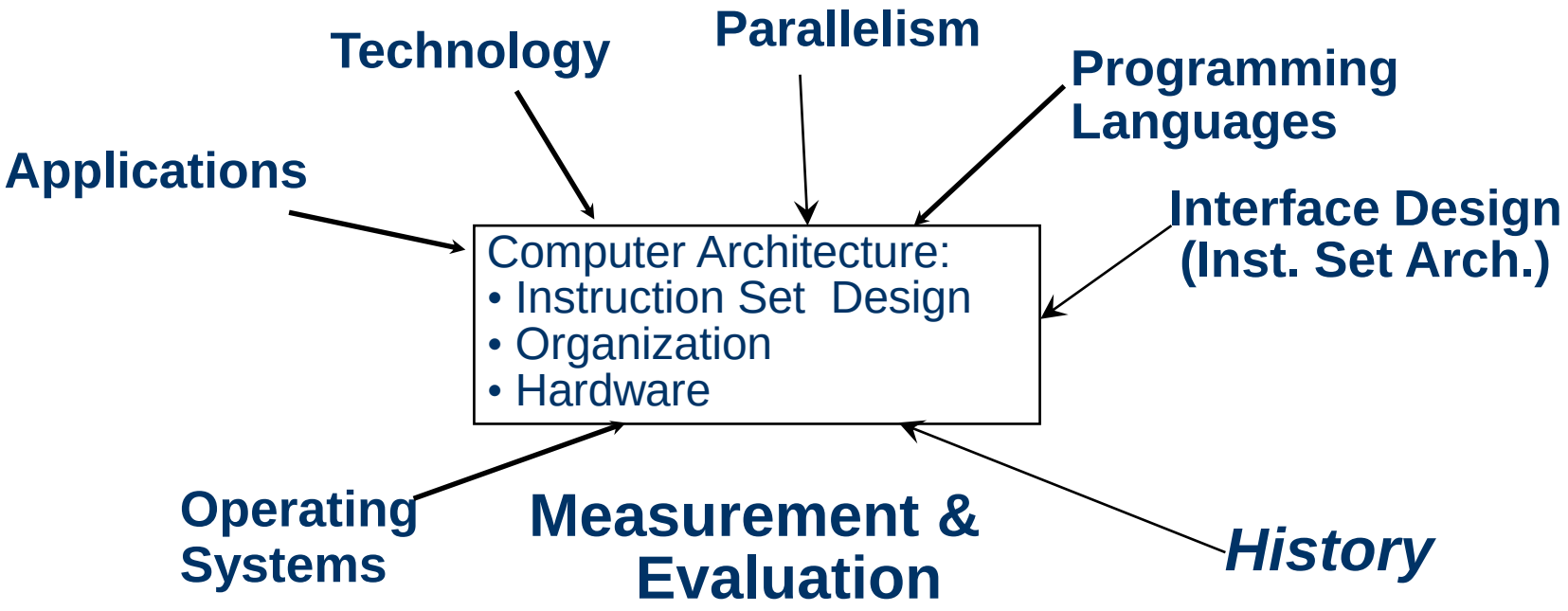
Multiprocessors
Networks and Interconnections

Topologies,
Routing,
Bandwidth,
Latency,
Reliability

Course Focus

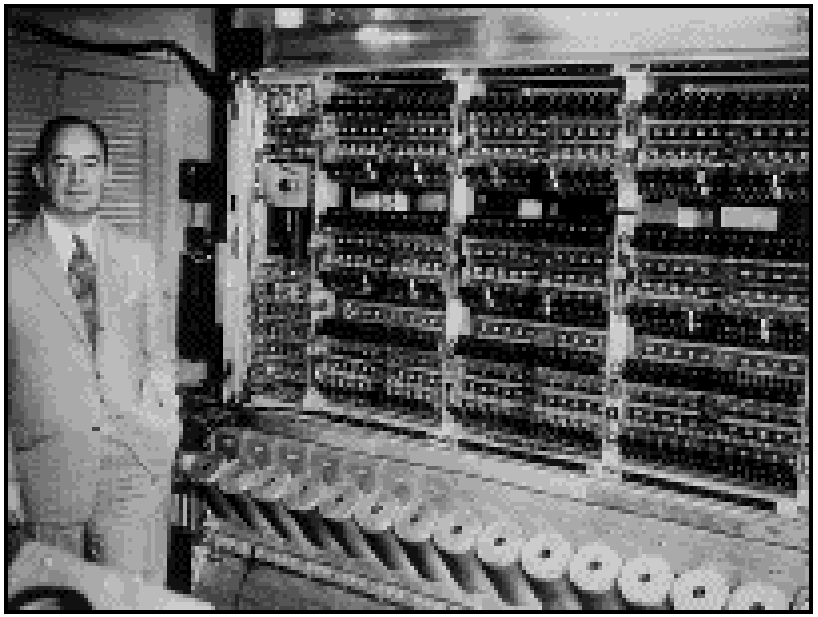
- **To Understand the design techniques, machine structures, technology factors, evaluation methods that will determine the form of computers in 21st Century**

- **Role of a computer architect:**
  - **To design and engineer the various levels of a computer system to maximize *performance* and *programmability* within limits of *technology* and *cost***



History

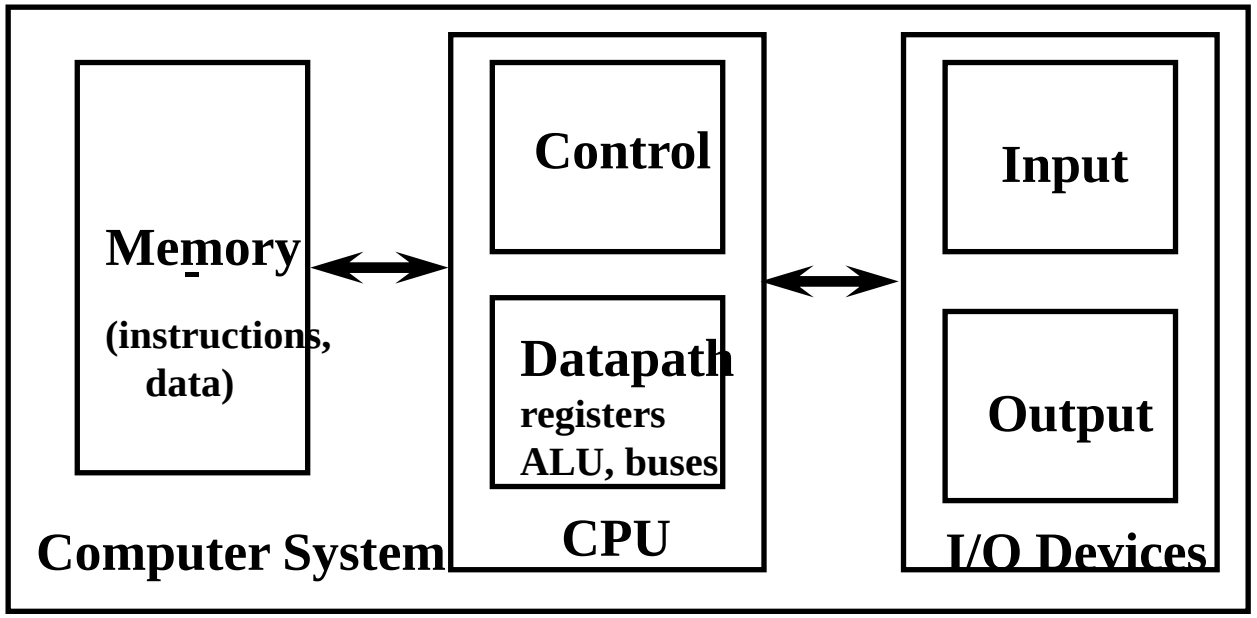
Von Newmann:
Invented EDSAC (1949).
First Stored Program
Computer. Uses Memory.



Importance: We are still using
The same basic design.

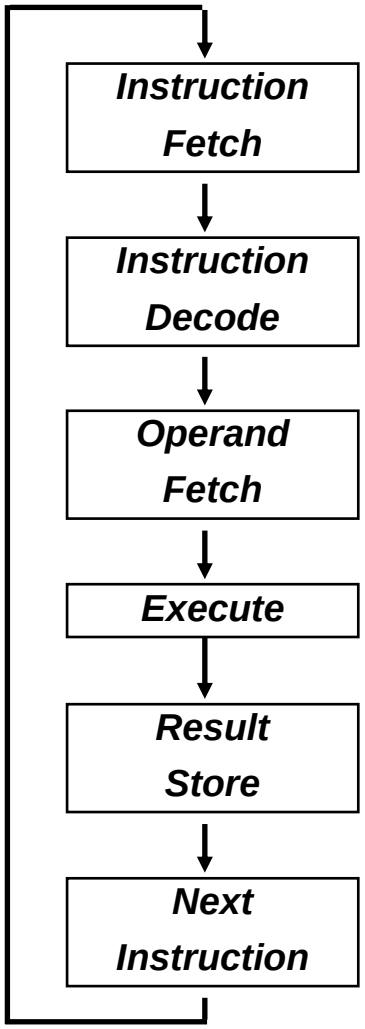
The Von Neumann Computer Model

- **Partitioning of the computing engine into components:**
  - Central Processing Unit (CPU): **Control Unit (instruction decode , sequencing of operations), Datapath (registers, arithmetic and logic unit, buses).**
  - Memory: **Instruction and operand storage.**
  - Input/Output (I/O) sub-system: **I/O bus, interfaces, devices.**
  - The stored program concept: **Instructions from an instruction set are fetched from a common memory and executed one at a time**



Major CPU Performance Limitation: The Von Neumann computing model implies sequential execution one instruction at a time

Generic CPU Machine Instruction Execution Steps



(Implied by The Von Neumann Computer Model)

Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

Compute result value or status

Deposit results in storage for later use

Determine successor or next instruction

What is “Computer Architecture” ?

- **Computer Architecture =**

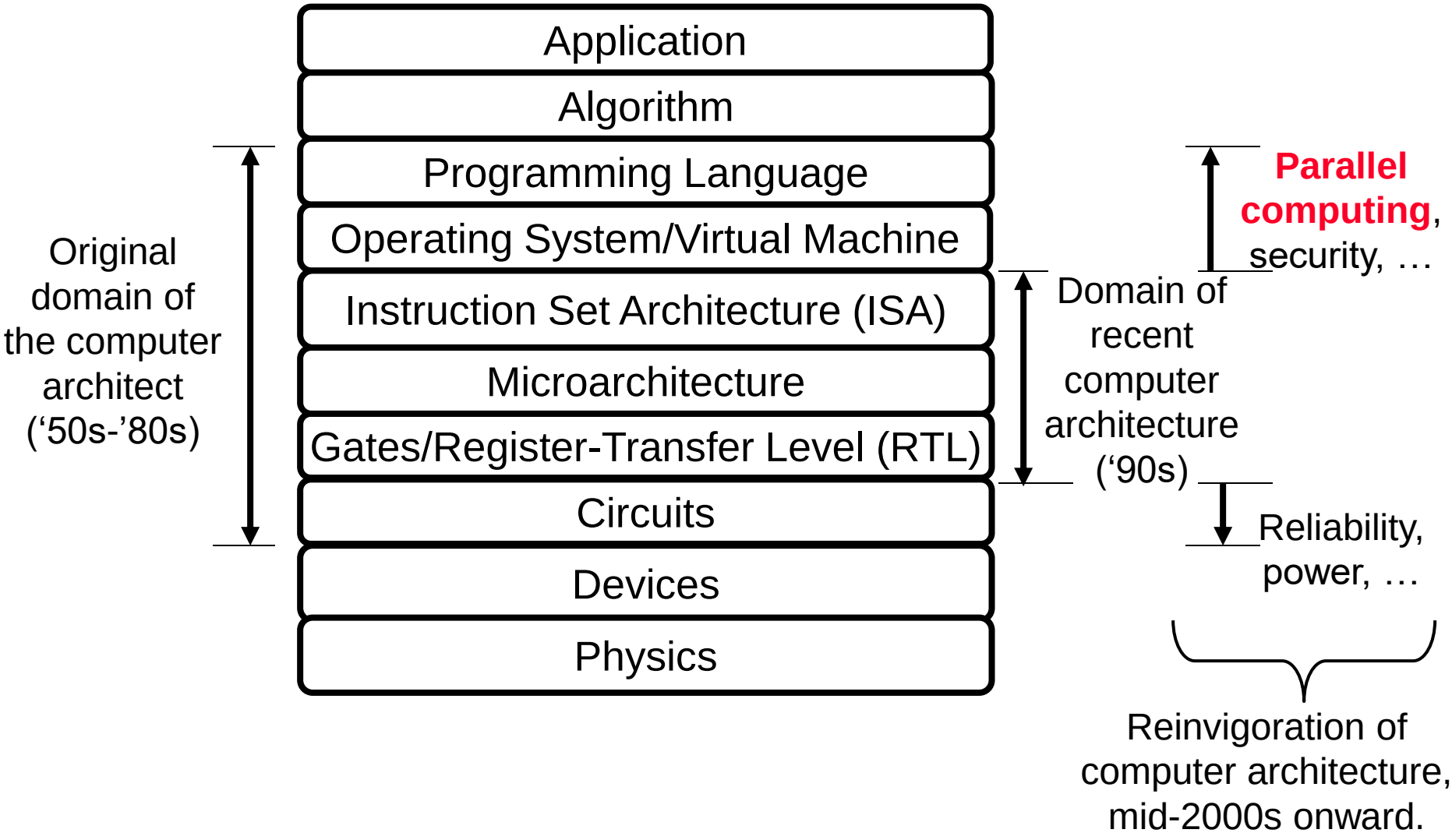
  Instruction Set Architecture +
  Computer Organization

- **Instruction Set Architecture (ISA)**
  - **WHAT the computer does (logical view)**
- **Computer Organization**
  - **HOW the ISA is implemented (physical view)**
- **We will study both in this course**

Computer Organization

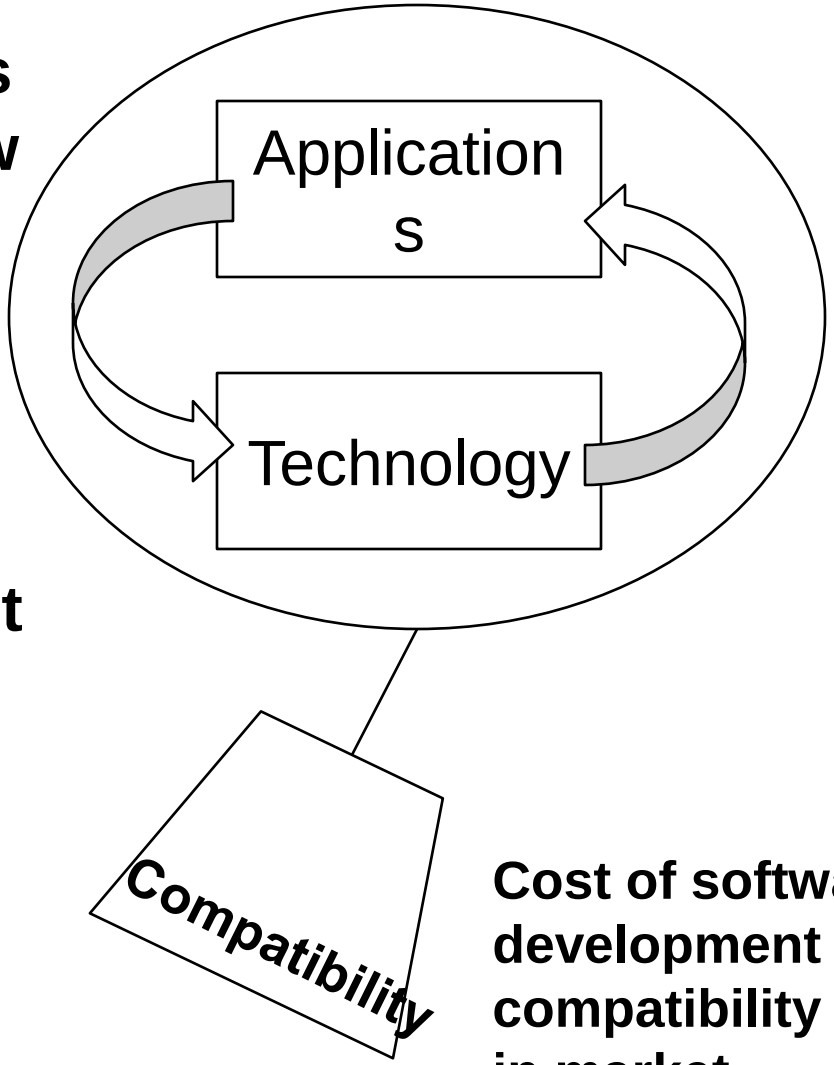
- **Realization of the Instruction Set Architecture**
- **Characteristics of principal components**
  - **Registers, ALUs, FPUs, Caches, ...**
- **Ways in which these components are interconnected**
- **Information flow between components**
- **Means by which such information flow is controlled**
- **Register Transfer Level (RTL) description**

Abstraction Layers in Modern Systems



Architecture continually changing

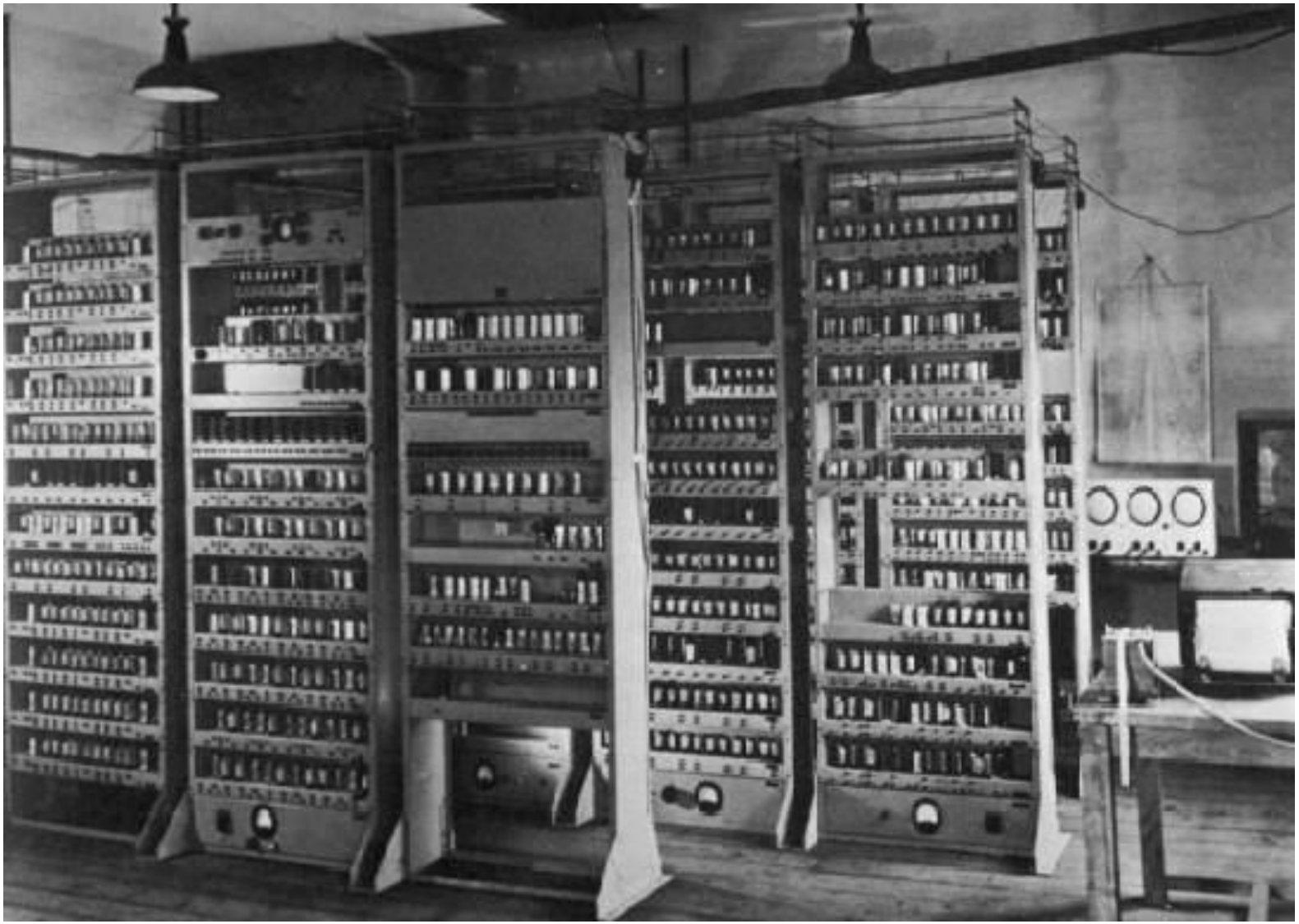
Applications suggest how to improve technology, provide revenue to fund development



Improved technologies make new applications possible

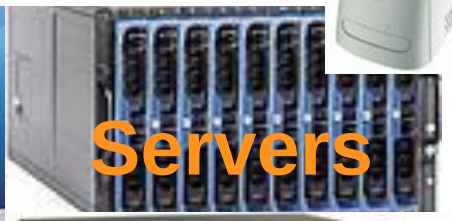
Cost of software development makes compatibility a major force in market

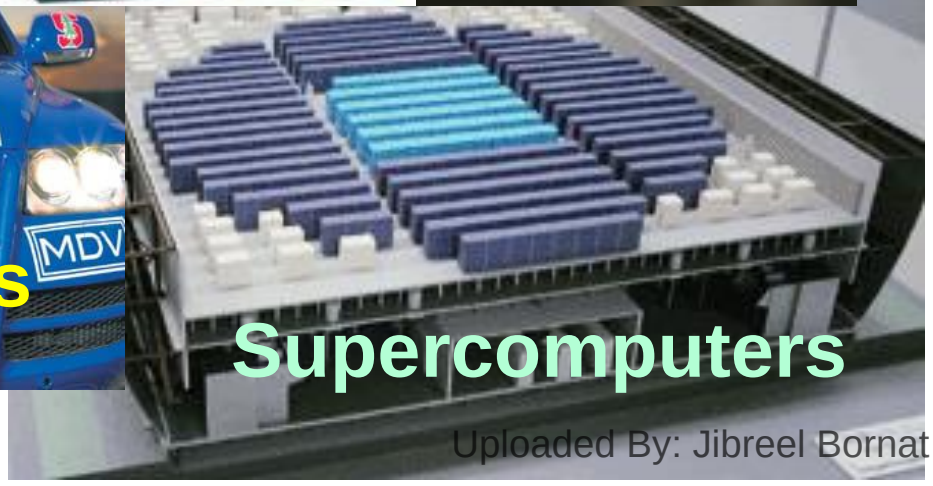
Computing Devices Then…



EDSAC, University of Cambridge, UK, 1949

Computing Devices Now





Crossroads: Conventional Wisdom in Comp. Arch

- Old Conventional Wisdom: Power is free, Transistors expensive
- New Conventional Wisdom: "Power wall" Power expensive, Xtors free
  (Can put more on chip than can afford to turn on)
- Old CW: Sufficiently increasing Instruction Level Parallelism via compilers, innovation (Out-of-order, speculation, VLIW, …)
- New CW: "ILP wall" law of diminishing returns on more HW for ILP
- Old CW: Multiplies are slow, Memory access is fast
- New CW: "Memory wall" Memory slow, multiplies fast
  (200 clock cycles to DRAM memory, 4 clocks for multiply)
- Old CW: Uniprocessor performance 2X / 1.5 yrs
- New CW: Power Wall + ILP Wall + Memory Wall = Brick Wall
  - **Uniprocessor performance now 2X / 5(?) yrs**

⇒ **Sea change in chip design: multiple "cores"**
(2X processors per chip / ~ 2 years)

» **More simpler processors are more power efficient**

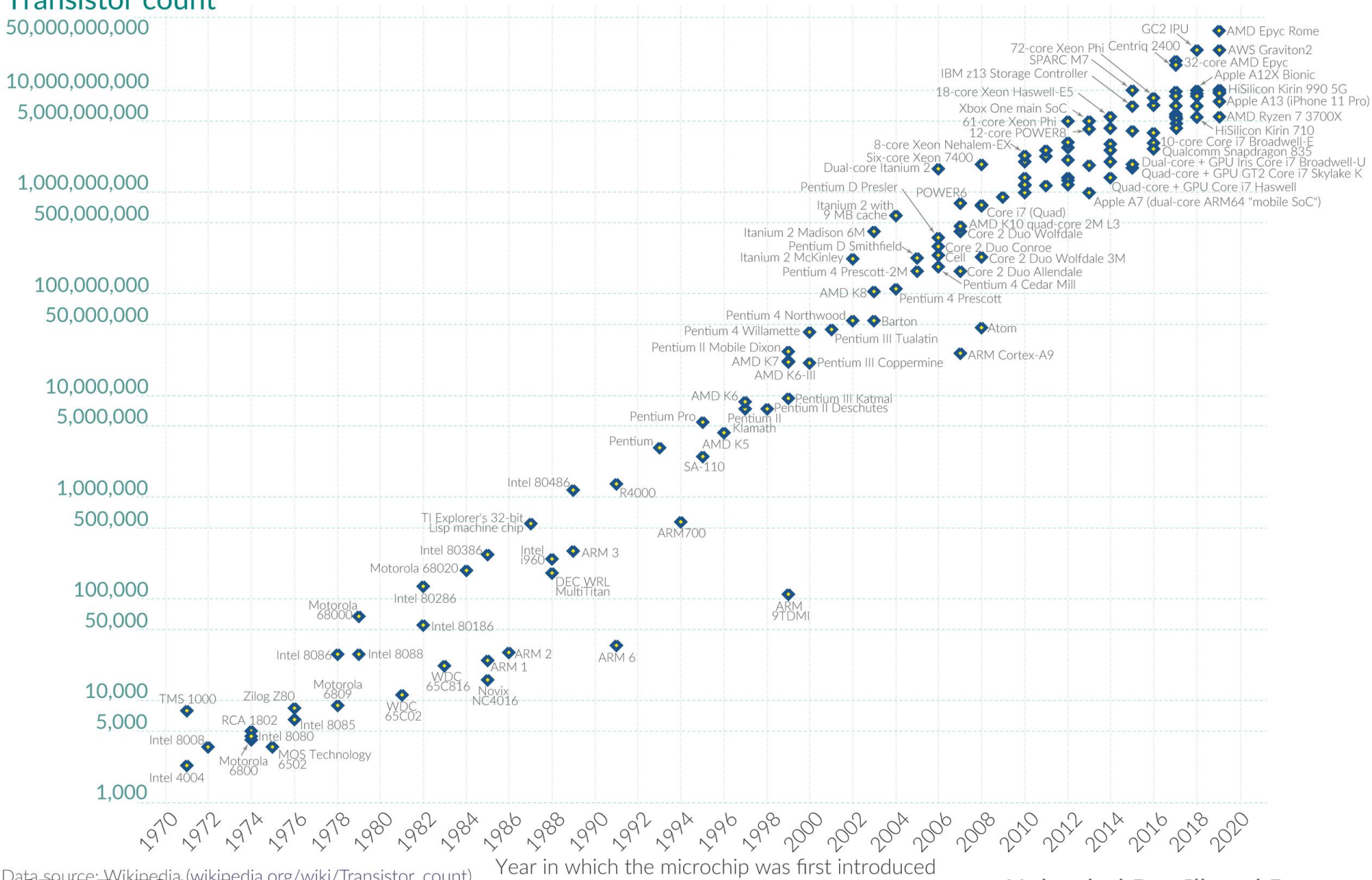
Technology Change

- **Technology changes rapidly**
  - **HW**
    - » **Vacuum tubes: Electron emitting devices**
    - » **Transistors: On-off switches controlled by electricity**
    - » **Integrated Circuits(IC/ Chips): Combines thousands of transistors**
    - » **Very Large-Scale Integration(VLSI): Combines millions of transistors**
    - » **What next?**
  - **SW**
    - » **Machine language: Zeros and ones**
    - » **Assembly language: Mnemonics**
    - » **High-Level Languages: English-like**
    - » **Artificial Intelligence languages: Functions & logic predicates**
    - » **Object-Oriented Programming: Objects & operations on objects**

Moore's Law: The number of transistors on microchips doubles every two years

Our World in Data

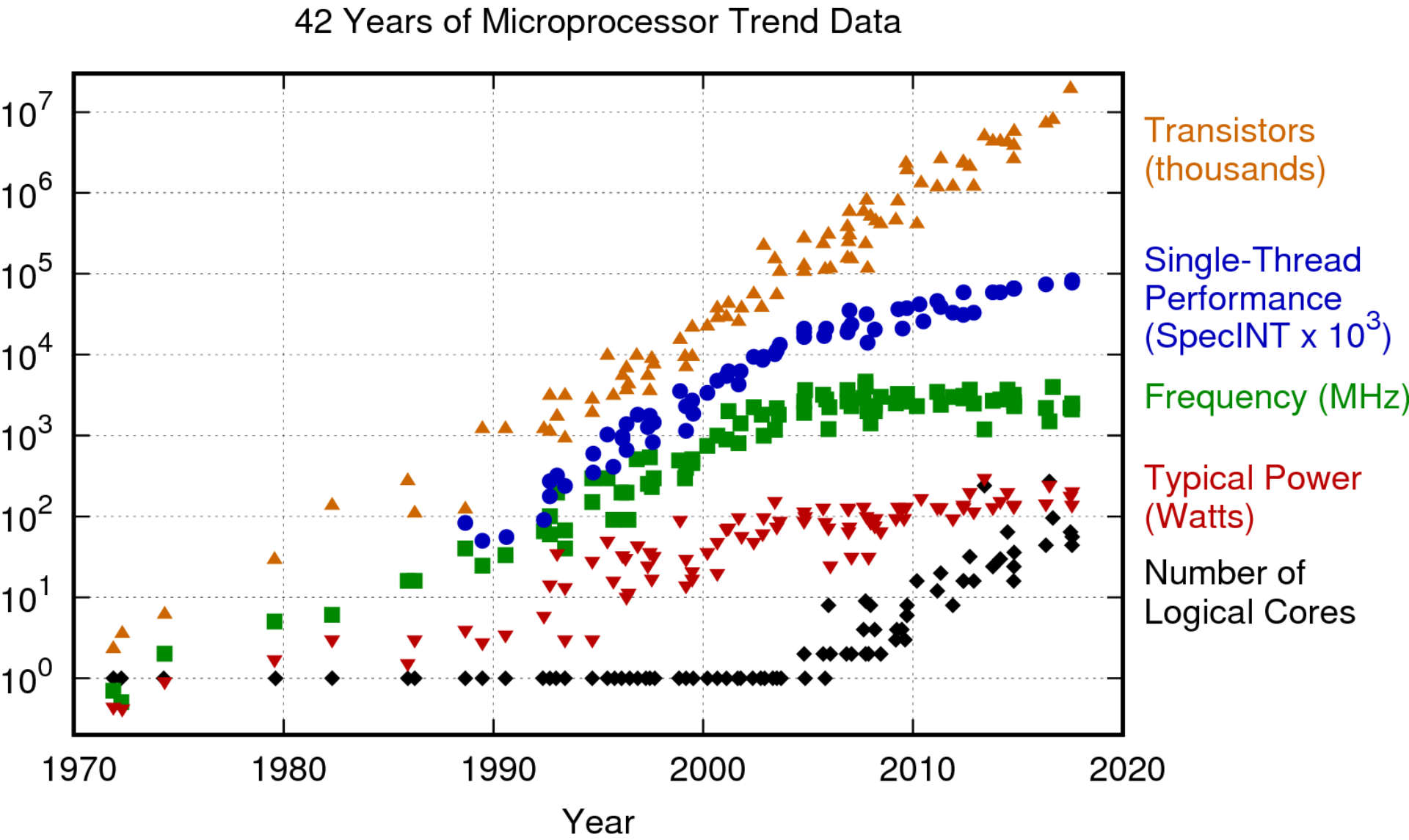
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

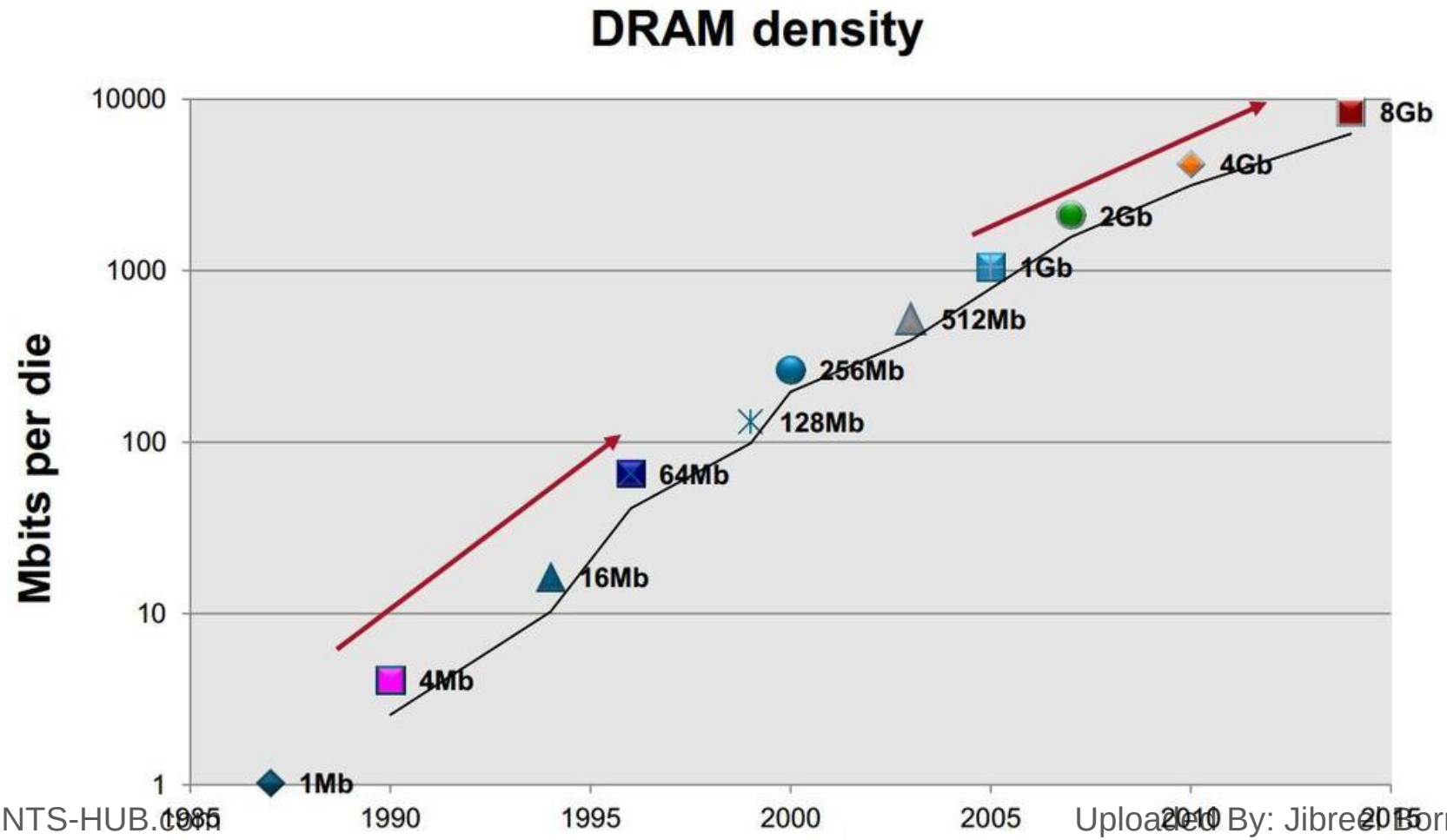
OurWorldInData.org – Research and data to make progress against the world's largest problems.

Processor Performance (1970-2020)

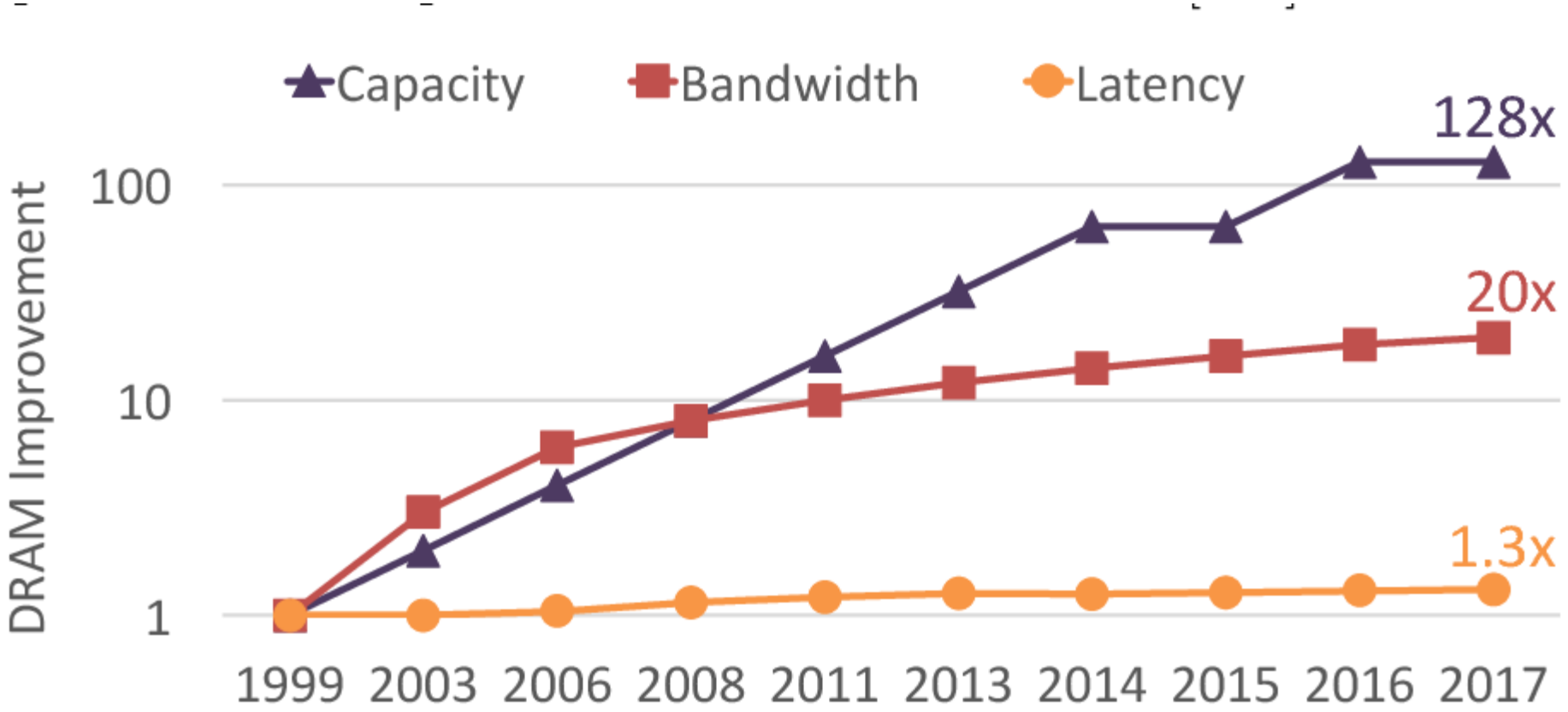


Growth of Capacity per DRAM Chip

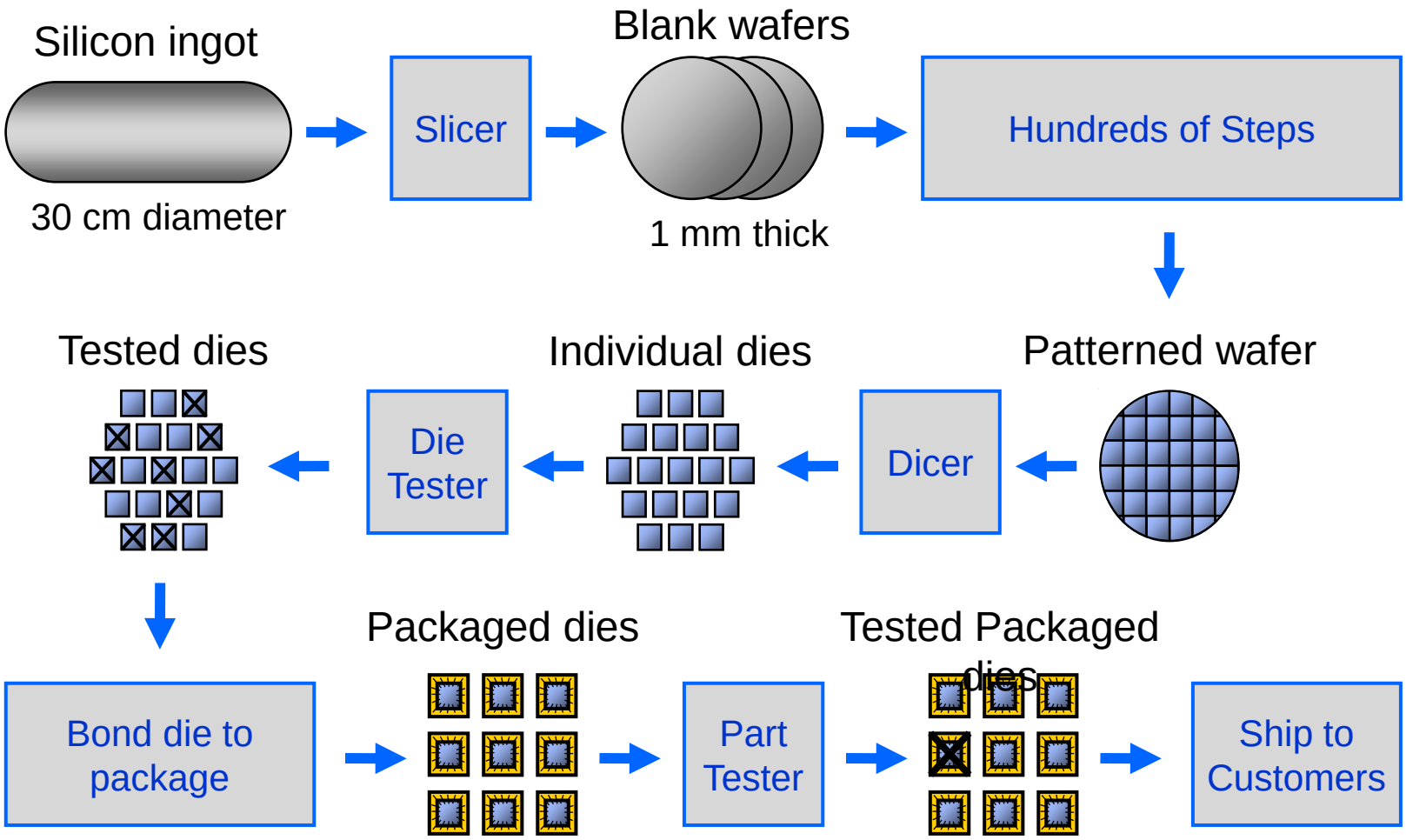
- DRAM capacity quadrupled almost every 3 years
  - 60% increase per year, for 20 years



Memory Improvements



Chip Manufacturing Process



Course Roadmap

- **Instruction set architecture**
- **Performance issues**
- **Constructing a processor**
- **Pipelining to improve performance**
- **Memory: caches and virtual memory**
- **Introduction to Parallel Architectures**