Chapter 1

Computer Abstractions and Technology

The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore's Law
    - vacuum tube → transistor → IC → VLSI
    - ~ doubling every 18 months:

      memory capacity

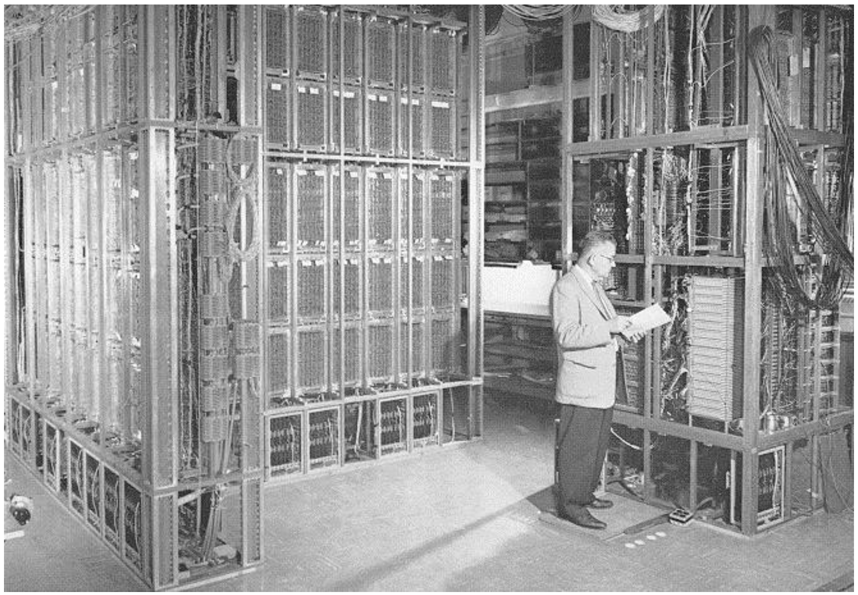
      processor speed

      » (Due to advances in technology and organization/architecture)

Old Mainframes

ILLIAC II (1962)

8192 words of memory



IBM 7094 (1970's vintage)

32K words of memory

Newer, pervasive computers

- Tablets and phones

More powerful than the mainframes on previous slide.

The Computer Revolution

- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasive!

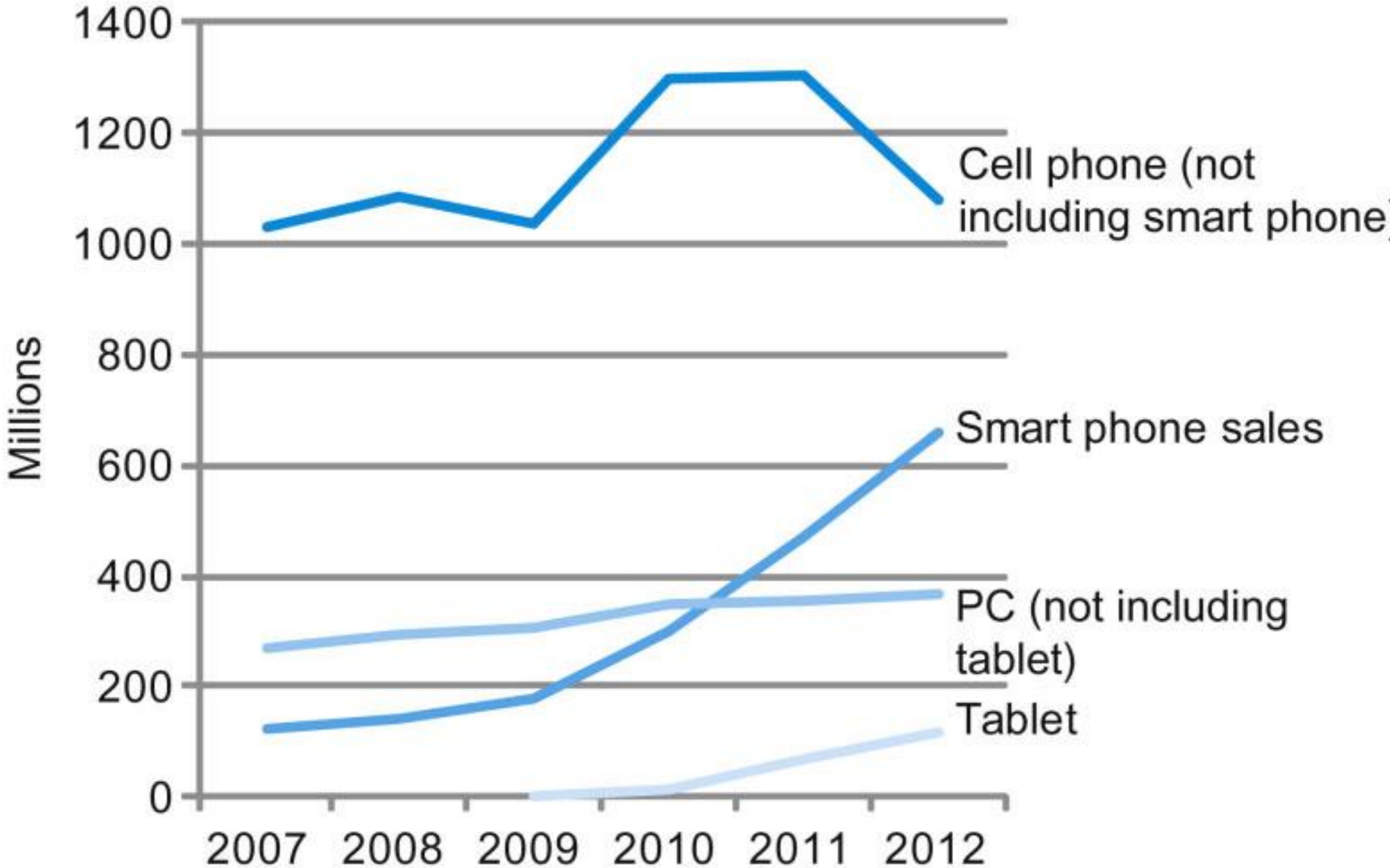
Classes of Computers

- Desktop computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized

Classes of Computers

- Supercomputers
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

The PostPC Era



The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- Cloud computing
  - Warehouse Scale Computers (WSC)
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion run in the Cloud
  - Amazon and Google

What You Will Learn

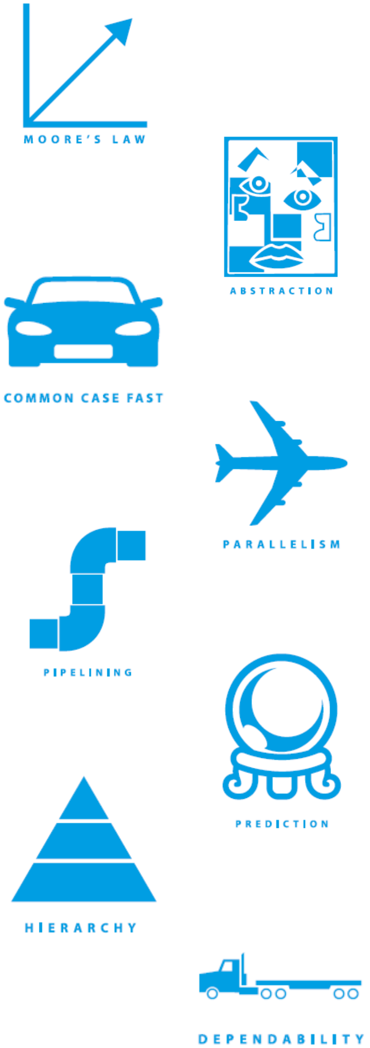
- How programs are translated into the machine language
  – And how the hardware executes them
- The hardware/software interface
- What determines program performance
  – And how it can be improved
- How hardware designers improve performance
- What is parallel processing

Understanding Performance

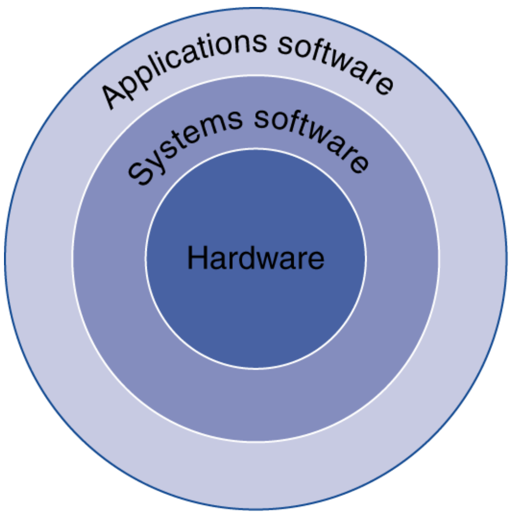
- Algorithm
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation
- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

Eight Great Ideas

- Design for ***Moore's Law***
- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** *via* redundancy



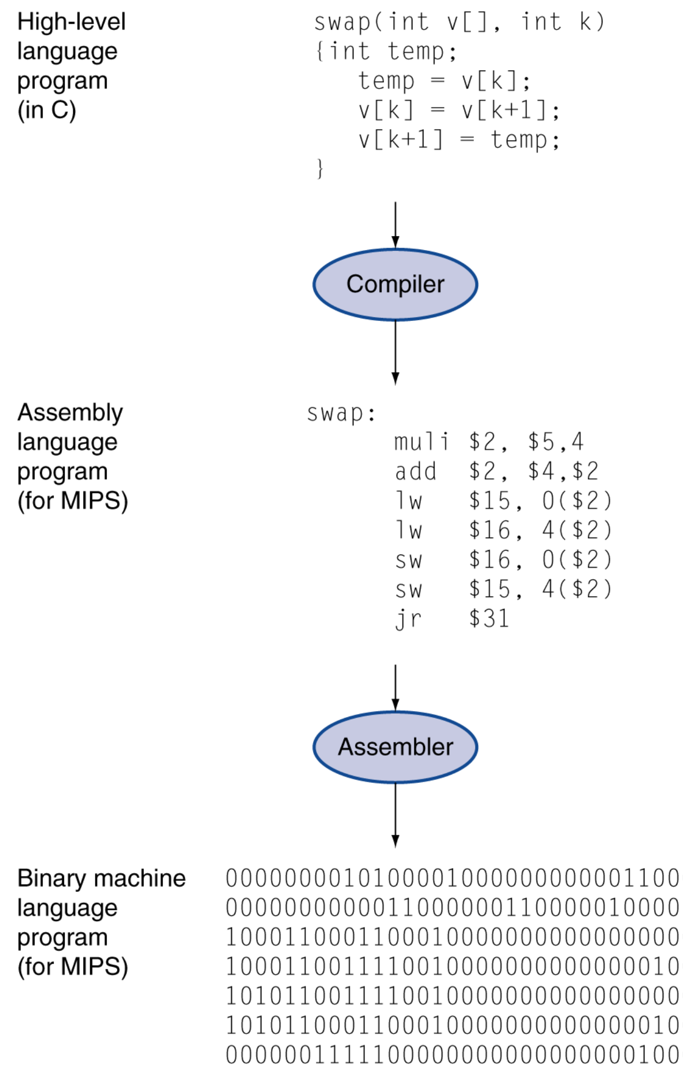
Below Your Program



- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



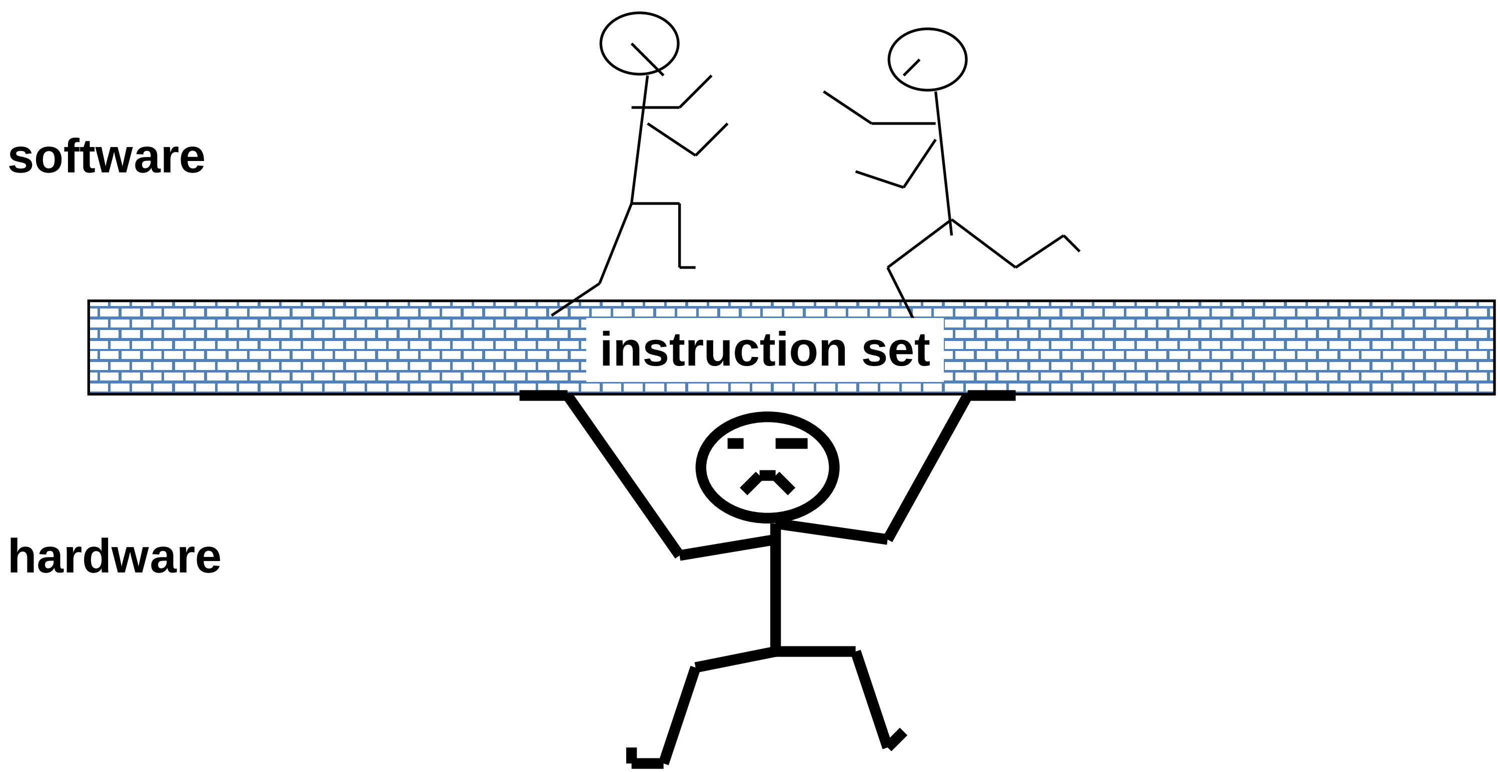
What is “Computer Architecture”

Computer Architecture =

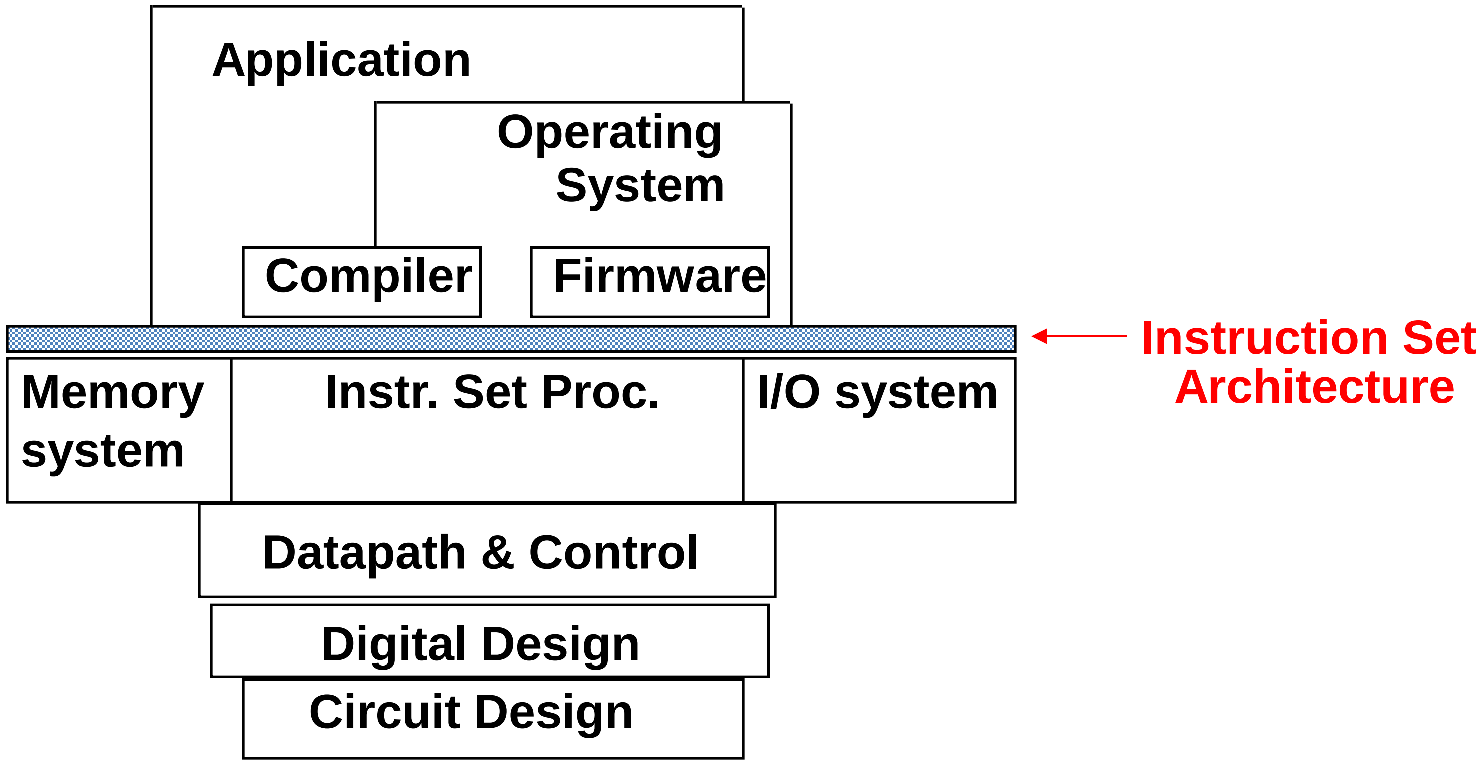
Instruction Set Architecture (sw) +

Machine Organization (hw)

The Instruction Set: a Critical Interface



How Do the Pieces Fit Together?



- Coordination of many *levels of abstraction*
- Under a rapidly *changing set of forces*
- Design, measurement, *and* evaluation

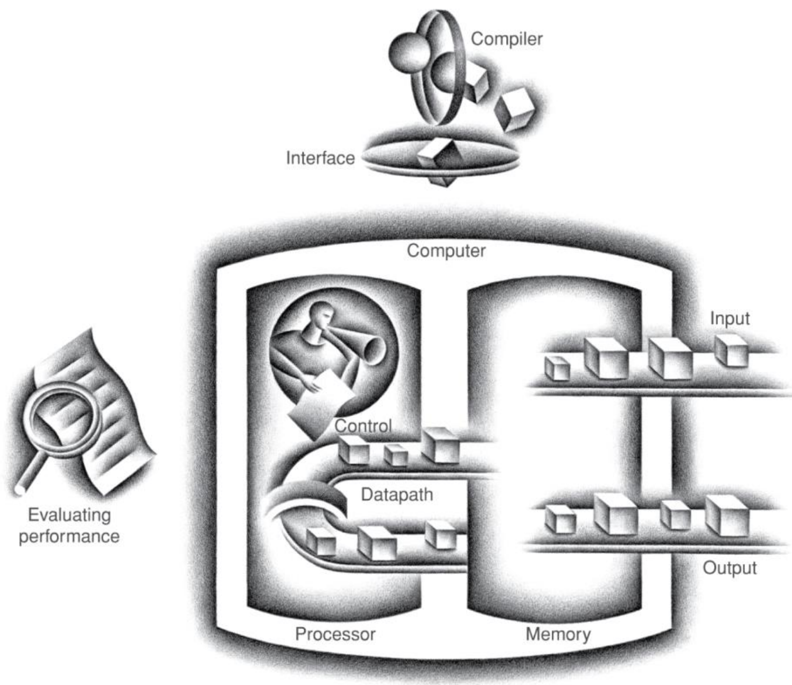
Abstractions

The BIG Picture

- Abstraction helps us deal with complexity
  - Hide lower-level detail
- Instruction set architecture (ISA)
  - The hardware/software interface
- Application binary interface
  - The ISA plus system software interface
- Implementation
  - The details underlying and interface

Components of a Computer

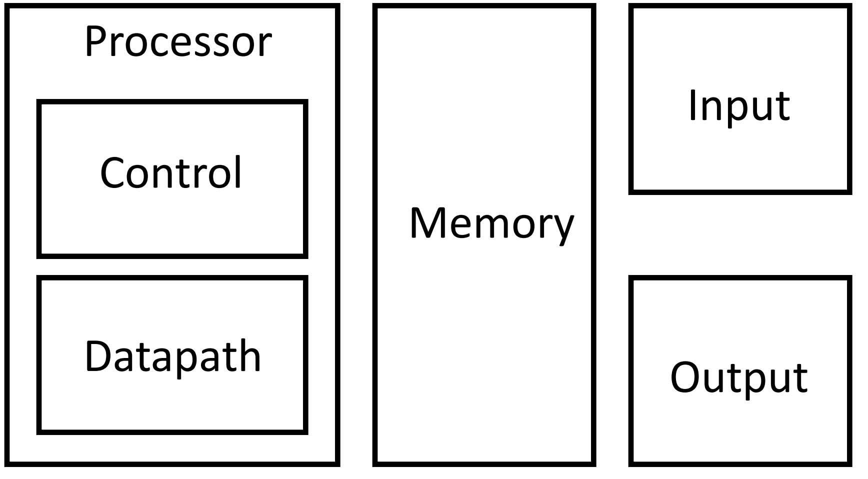
The BIG Picture



- Same components for all kinds of computer
  - Desktop, server, embedded
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

The Big Picture

- The Five Classic Components of a Computer



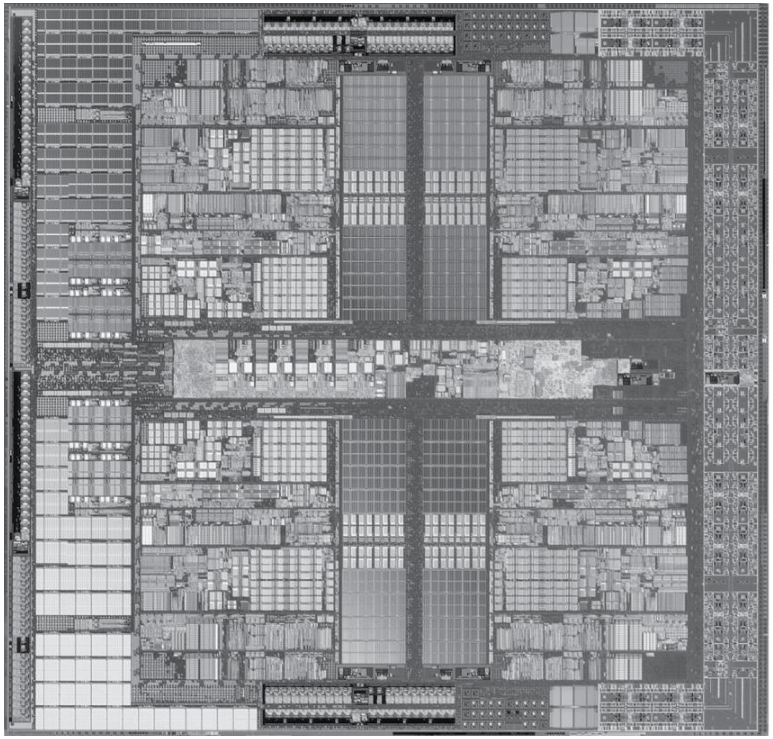
Inside the Processor (CPU)

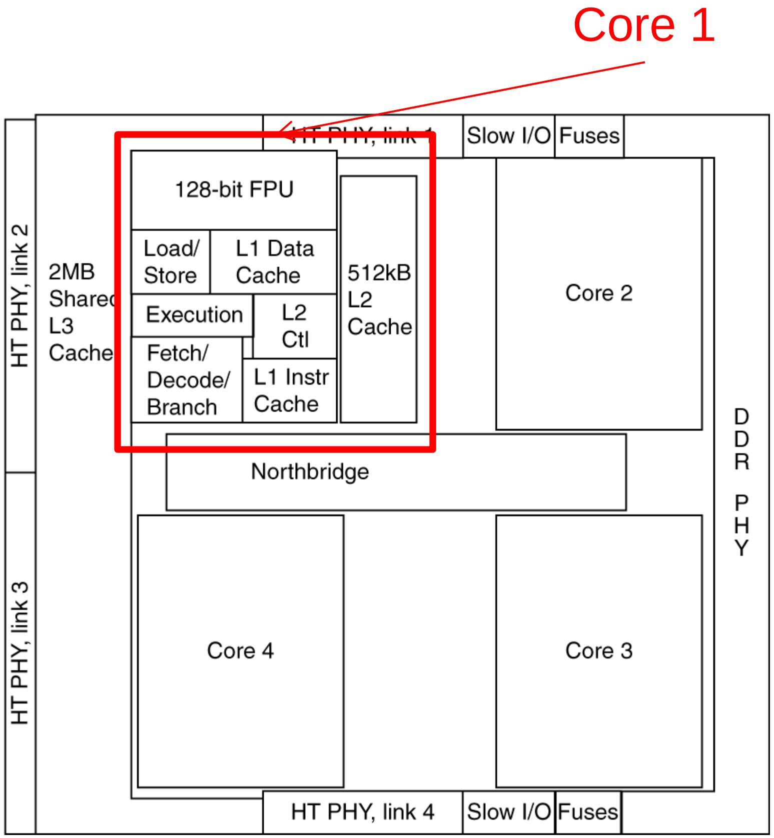
- Datapath: performs operations on data
- Control: sequences datapath, memory, ...
- Cache memory
  – Small fast SRAM memory for immediate access to data

Chapter 4: We' ll spend a lot of our time in this course on these bullets!

Inside the Processor

- AMD Barcelona: 4 processor cores





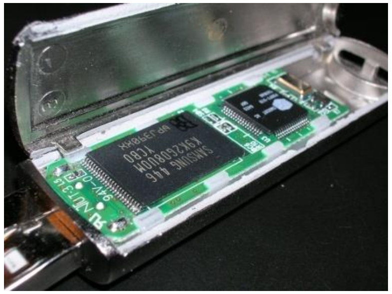
Inside the Processor

- Apple A5: Dual core ARM (iphone, ipad, apple TV)

2 cores

A Safe Place for Data

- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Solid state drives (SSD): becoming more common
  - Optical disk (CDROM, DVD)

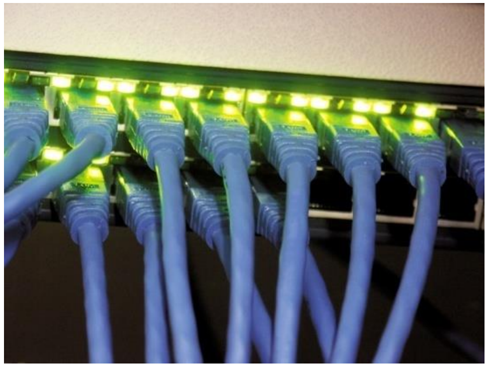


SSD drive

Mag disk

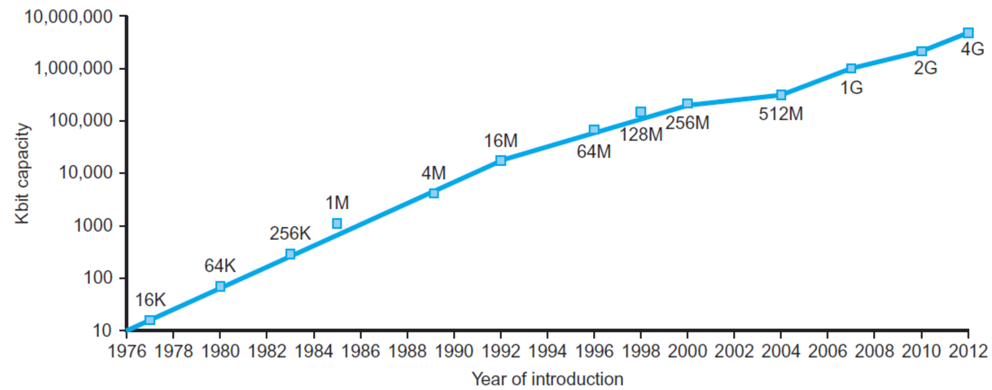
Networks

- Communication and resource sharing
- Local area network (LAN): Ethernet
  - Within a building
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth



Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



DRAM capacity

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Where we are headed

- Read Chapter 1 (introduction) on your own.
- Performance issues (Chapter 1) *vocabulary and motivation*, *basic concepts.*
- Instruction Set Architectures in general and a specific instruction set architecture (MIPS) (Chapter 2)
- Number representations, computer arithmetic, and how to build a simple integer ALU (Chapter 3)
- Constructing a MIPS processor (Chapter 4)
  - Build a processor to execute a subset of MIPS instruction set (datapath+control)
  - Pipelining to improve performance (datapath+control), hazards, etc.
- Memory architecture: caches and virtual memory (Chapter 5)
- Parallel Architectures/multiprocessors (Chapter 6)