# CS161 – Design and Architecture of Computer Systems

## Introduction

*Khaled N. Khasawneh, PhD Student*
*Department of Computer Science and Engineering*
*kkhas001@ucr.edu*

UNIVERSITY OF CALIFORNIA, RIVERSIDE

# Welcome!

# About me

- Born and raised in Jordan
- Jordan University of Science & Technology, Jordan
  - BS Computer Engineering ’12
- Binghamton University, Binghamton, NY
  - MS Computer Science ’14
- UCR, Riverside, CA
  - PhD Computer Science (In Progress)
- Research Interest
  - Hardware support for security, Malware detection, Side channels

# CS161 Goal

- Introduction to Computer Architecture
  - Familiarity with processor components (pipeline, caches, registers, etc. )
  - Provide foundation for further comp arch courses
    - CS162 – Computer Architecture
    - CS203 – Advanced Computer Architecture

# So, I Hope You Are Here for This

- How does an assembly program end up executing as digital logic?
- What happens in-between?
- How is a computer designed using logic gates and wires to satisfy specific goals?

"C" as a model of computation

Programmer's view of how a computer system works

*Architect/microarchitect's view: How to design a computer that meets system design goals.*

*Choices critically affect both the SW programmer and the HW designer*

HW designer's view of how a computer system works

Digital logic as a model of computation

# Topics Covered

- Prerequisite: CS/EE 120A
- Background
  - Quantifying Performance, Technology Trends, …
- Instruction Set Architecture
- CPU Design
  - Single cycle, Multi cycle
- Processor Pipelining
  - 5-stage pipeline
- Memory hierarchy
  - Memory, Cache, Virtual Memory
- Reliability
  - RAID

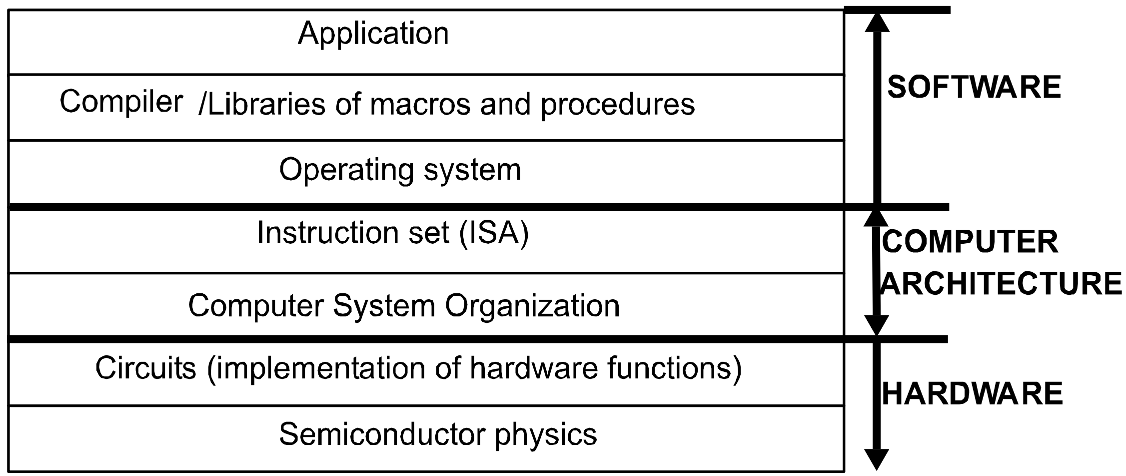
# Why learn Comp Arch?

- Computer Architecture is the glue that binds software and hardware
  - Inter-disciplinary in nature
  - Devices, Circuits, OS, Runtime, PL, Compilers
- Advancement of computer architecture is vital to all other areas of computing
  - IoT, Embedded
  - Mobile
  - Data centers, HPC

# What is Computer Architecture?

- Hardware organization of computers
  - how to build computers

- Layered view of computer systems

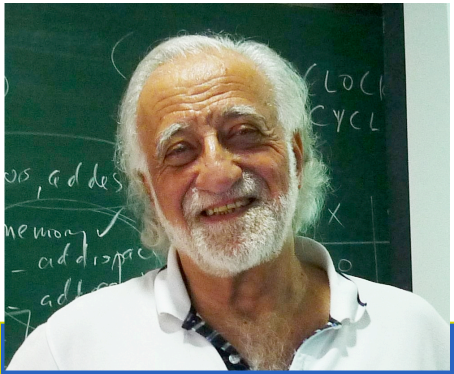



- Role of the computer architect:
  - To make design trade-offs across the hw/sw interface to meet functional, performance and cost requirements

# Role of the (Computer) Architect

## *Role of the Architect*

- *-- Look Backward (Examine old code)*
- *-- Look forward (Listen to the dreamers)*
- *-- Look Up (Nature of the problems)*
- *-- Look Down (Predict the future of technology)*



from Yale Patt's lecture notes

# Logistics

- Course Website
  - www.cs.ucr.edu/~kkhas001/cs161-f16.html
  - Check often for announcements
- Assignments/Projects
  - iLearn (iLearn.ucr.edu)
- Discussion/Help
  - Piazza (piazza.com/ucr/fall2016/cs161/home)

# Textbook

- Computer Organization and Design, 5th Edition
  By Patterson and Hennessy

- Not required, but I encourage you get the book

# Attendance/Grading

- Attendance
  - You are expected to attend **all** lectures.
  - Some slides only make sense in lecture. ☺
  - Come to class on time
  - Start early - do not procrastinate
- Grade Breakdown
  - Homework: 30%
  - Midterm: 30%
  - Final: 35%
  - Participation, Quizzes or Reading: 5%

# Assignment Policies

- Do them to truly understand the material not for the grade
- 10% penalty per late day
- If it's one minute late, it's still late
- No extensions will be given
- A total of one late submission (2 days) per quarter allowed
- Assignments should be uploaded to iLearn

# Contact

- Instructor: Khaled N. Khasawneh
  - Email: kkhas001@ucr.edu
  - Homepage: http://www.cs.ucr.edu/~kkhas001
  - Office: WCH 110
  - Office Hours: Tuesday & Thursday 2:30pm-3:30pm

- TA: Joshua Potter
  - Email: jpott002@ucr.edu
  - Office: WCH 464
  - Office Hours: Wednesday 4:30pm – 5:30pm

# CS161 – Design and Architecture of Computer Systems

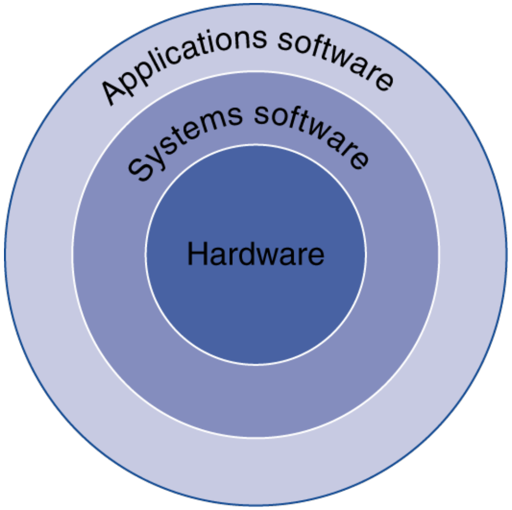
## Technology Trends

UNIVERSITY OF CALIFORNIA, RIVERSIDE

# 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

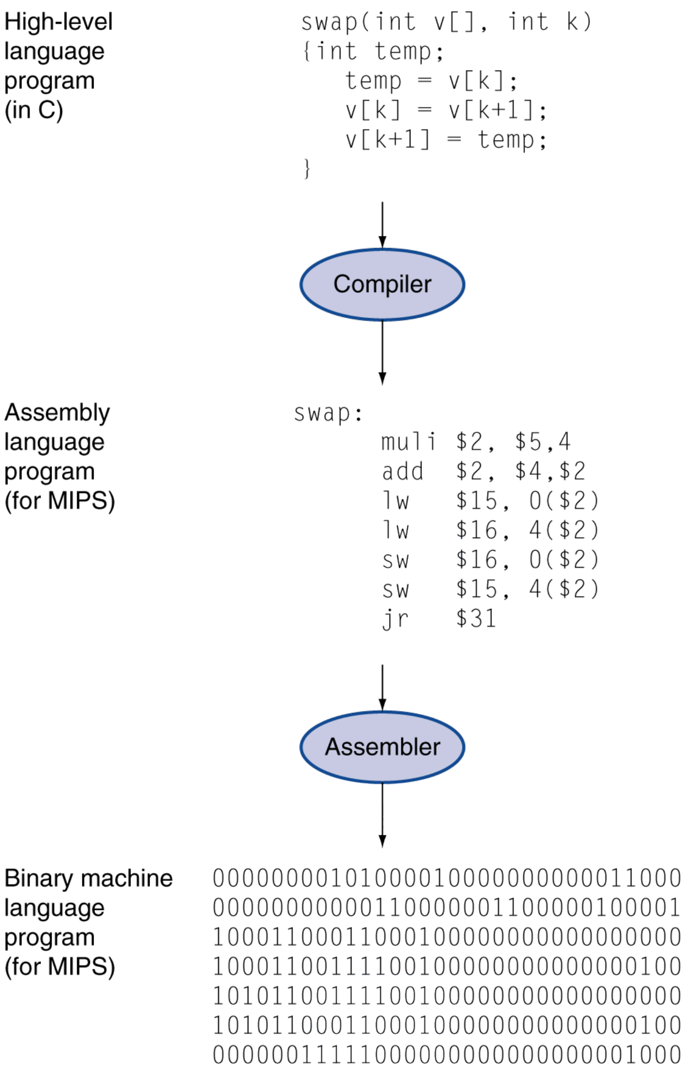
# Below Your Program




- 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

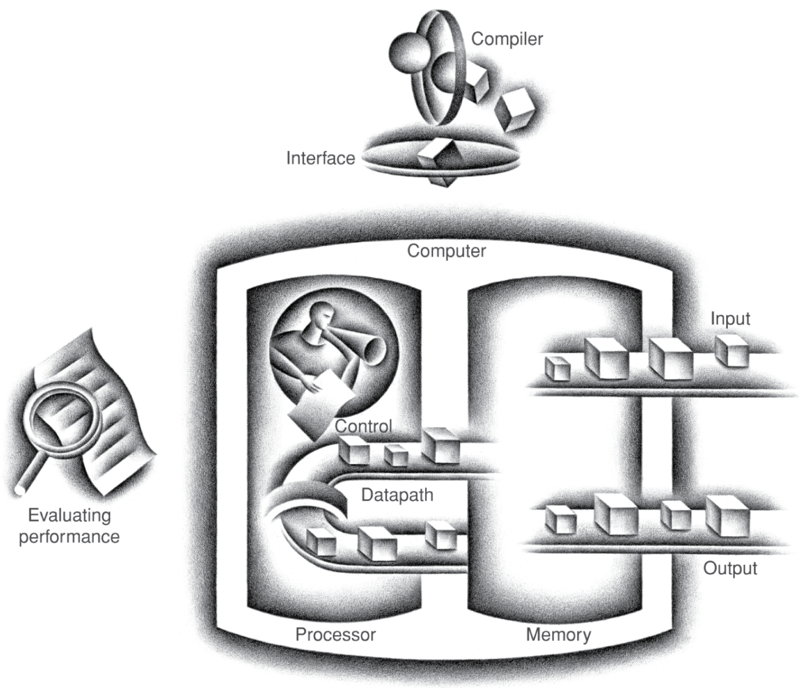
# Levels of Program Code

- 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



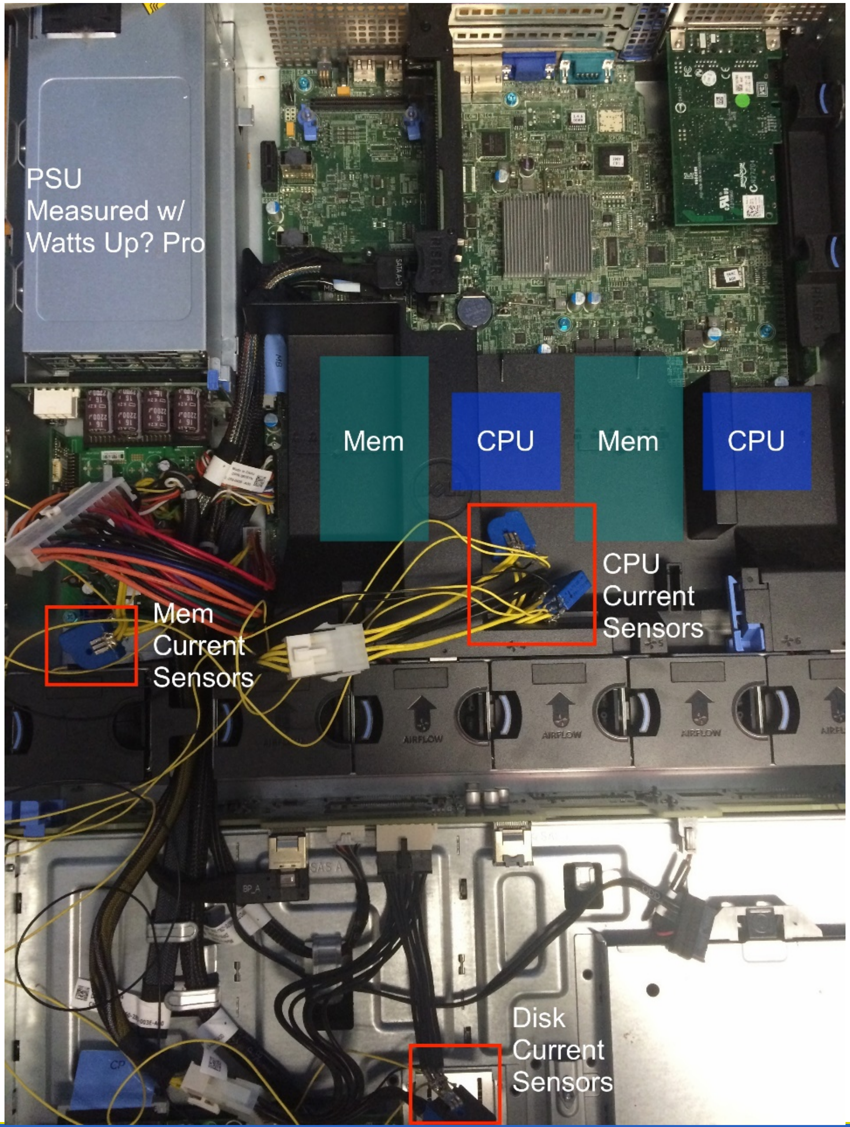
# Components of a Computer

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

# Anatomy of a Computer

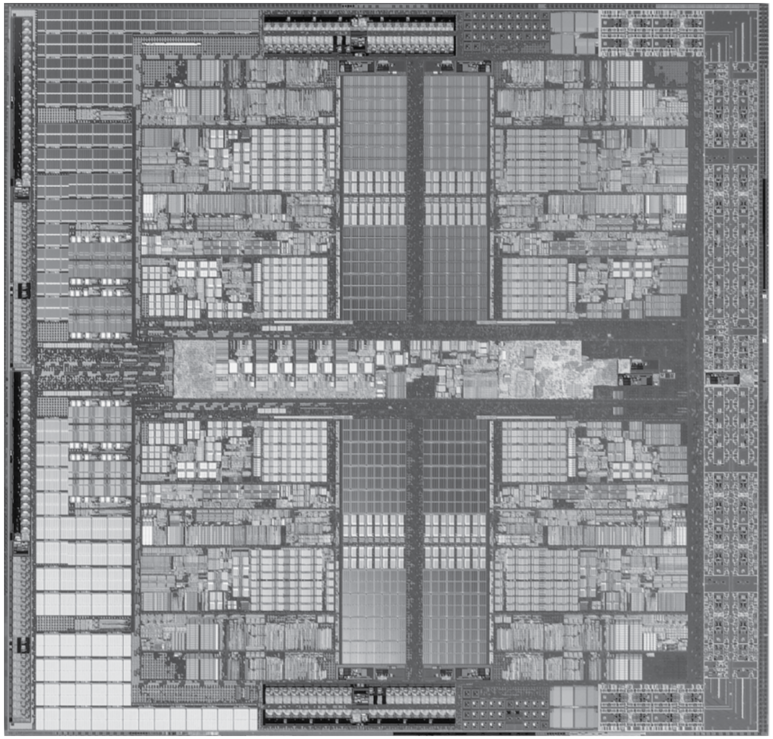


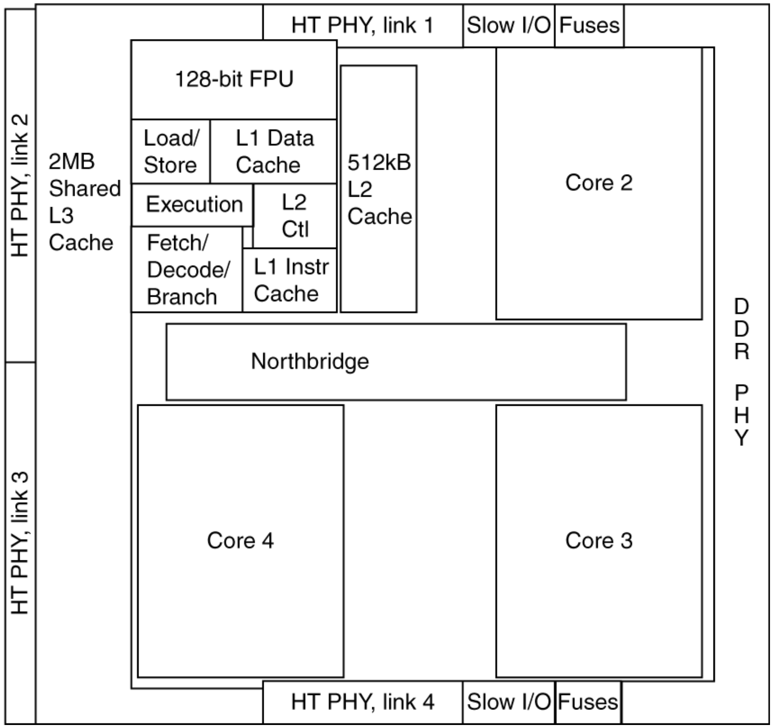
# Inside the Processor (CPU)

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

# Inside the Processor

- AMD Barcelona: 4 processor cores





# iPad 2 logic board


FIGURE 1.8 The logic board of Apple iPad 2 in Figure 1.7. The photo highlights five integrated circuits. The large integrated circuit in the middle is the Apple A5 chip, which contains a dual ARM processor cores that run at 1 GHz as well as 512 MB of main memory inside the package. Figure 1.9 shows a photograph of the processor chip inside the A5 package. The similar sized chip to the left is the 32 GB flash memory chip for non-volatile storage. There is an empty space between the two chips where a second flash chip can be installed to double storage capacity of the iPad. The chips to the right of the A5 include power controller and I/O controller chips. (Courtesy iFixit, www.ifixit.com)

# Apple A5 processor

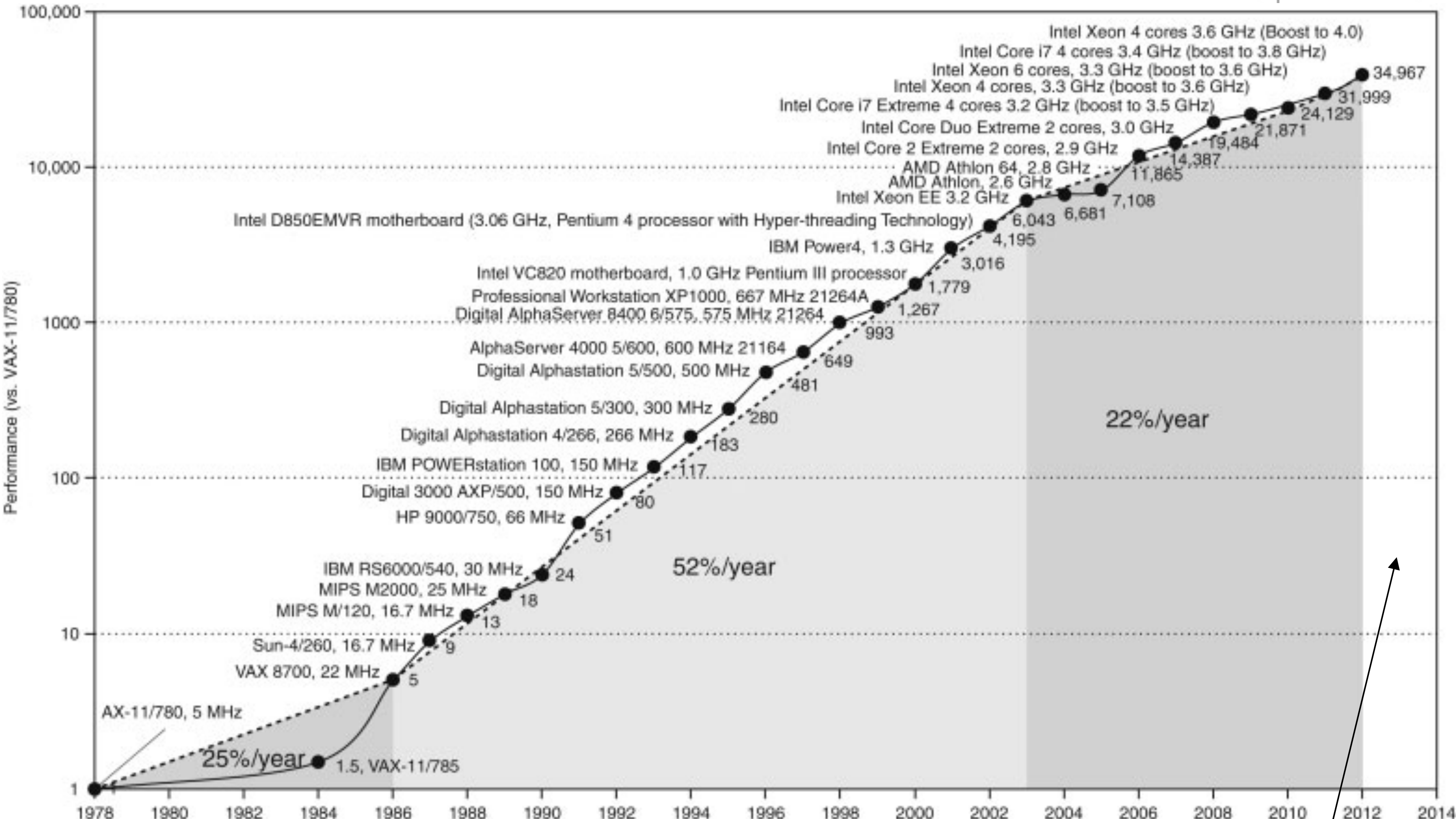
FIGURE 1.9 The processor integrated circuit inside the A5 package. The size of chip is 12.1 by 10.1 mm, and it was manufactured originally in a 45-nm process (see Section 1.5). It has two identical ARM processors or cores in the middle left of the chip and a PowerVR graphical processor unit (GPU) with four datapaths in the upper left quadrant. To the left and bottom side of the ARM cores are interfaces to main memory (DRAM). (Courtesy Chipworks, www.chipworks.com)

# How fast is it getting faster?

| Year | Technology used in computers | Relative performance/unit cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit | 900 |
| 1995 | Very large-scale integrated circuit | 2,400,000 |
| 2013 | Ultra large-scale integrated circuit | 6,200,000,000 |

$6.2x10^9$ in 62 years, the growth rate is = **36.37%**

# Uniprocessor Performance




Constrained by power, instruction-level parallelism, memory latency

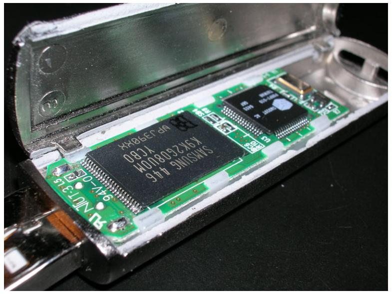
# Abstractions

**The BIG Picture**

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

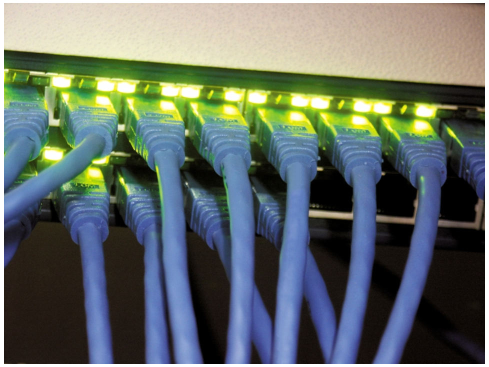
# A Safe Place for Data

- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)



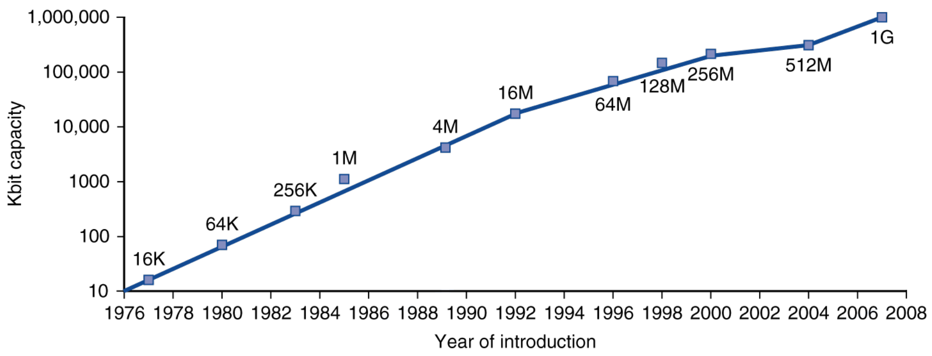
# Networks

- 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 |
| 2005 | Ultra large scale IC | 6,200,000,000 |