CSCI 330: Computer Architecture

Dr. Paul E. West

Department of Computer Science
Charleston Southern University

January 13, 2020

Complex Yet Simple

- Computers are some of the more complex devices humans have created.
- Yet, are simple in many ways. EX:
  - A CPU (where a lot of the “work” occurs) is basically a bunch of transistors etched on silicon.
  - by ‘a bunch’ I mean billions

From C to Execution

- How does the code that you type in go from text to executing on the computer?
- How does a piece of silicon turn you “English-like” C code into a dynamic computation?
- We will explore these in more depth.

Logic

- The computer executes your program (set of instructions) through electronic signals.
- These signals (in general) are executed in steps (clock cycle).
- Our job here is to see how this is done and give you an idea of how a computer can be built to do computations.

The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore’s Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasive

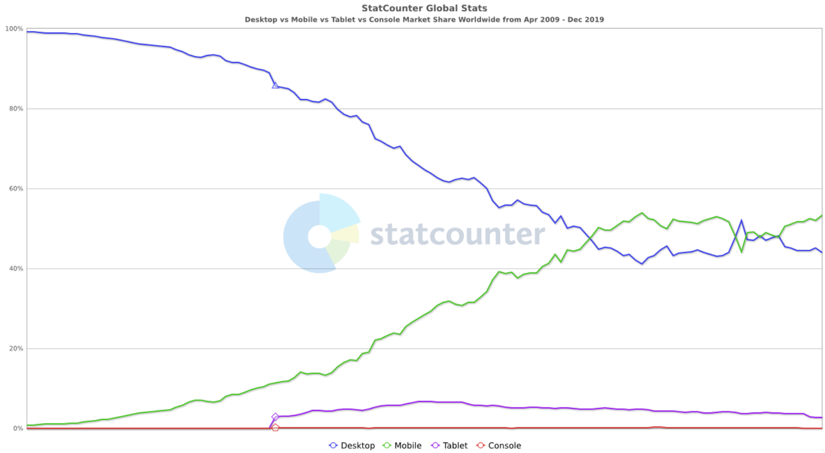
Classes of Computers

- Personal computers
  - General purpose, variety of software
  - Subject to unit cost vs 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
  - `http://www.top500.org` maintains a list
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

PC Mobile Tablet Console Market Share



Mobile Devices and Cloud

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses, smart watches
- 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

Patterns to Keep in Mind

- The three Ps for improving performance:
  - Parallelism
  - Pipelining
  - Prediction
- Small is fast, Big is slow
- You must think in parallel!
- Verilog
  - A different language used to describe processors

About the Professor

- PhD from Florida State University in Computer Science
- Faculty Experience:
    - Charleston Southern: Assistant 2015-Present
    - College of Charleston: Adjunct 2013-2014
- Work Experience:
    - Naval Research Lab: 2018 - Present
    - Google (2014): Android Bluetooth/Wi-Fi/Telephony
    - SPAWAR (2009-2014, 2015): Communication systems
    - DenimGroup (2004-2005): Start-up; web design and network security

Github

- The programming parts of your assignments will be submitted through Github.
- Please create an account on Github and email me your username.
- More instruction to follow...

ScrumDo

- A board to organize our workflow for this course.
- We will use if it will allow us to create enough users.
- Lets find out...

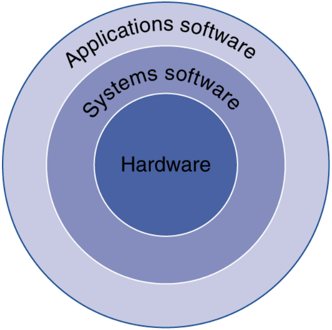
Syllabus

Lets go over the syllabus...

Discussion

- Any questions about the syllabus or the course?
- Have you seen anything in the news?

Below Your Program

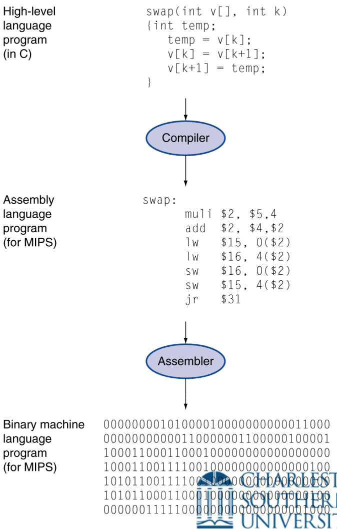


- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks and sharing resources
- Hardware
  - Processor, memory,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

CHARLESTON SOUTHERN UNIVERSITY

Levels of 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



High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Assembly language program (for MIPS)

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

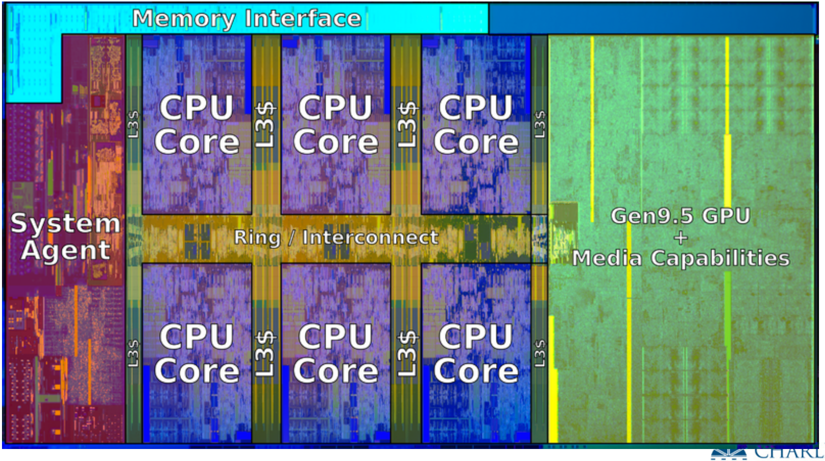
Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

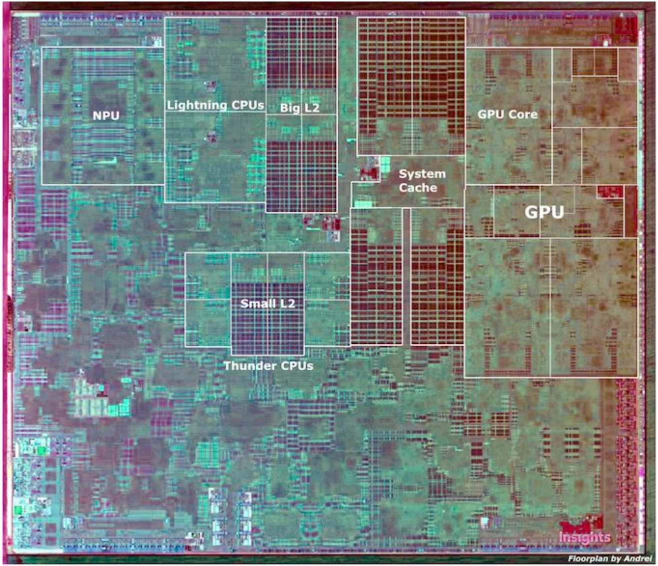
Inside the Processor (CPU)

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

Intel Haswell



Apple A13


NPU
Lightning CPUs
Big L2
GPU Core
System Cache
GPU
Small L2
Thunder CPUs
TechInsights
Floorplan by Andrei

Safe Place for Data

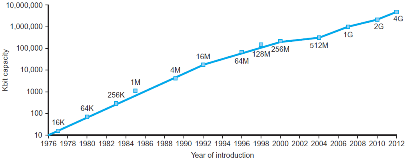
- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)

Networks

- Communication, resource sharing, nonlocal access
- Local area network (LAN): Ethernet
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth

Technology Trends

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

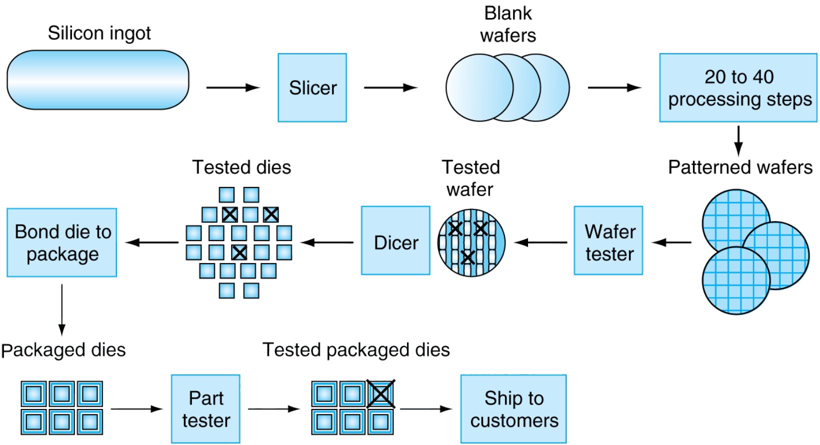


Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Semiconductor

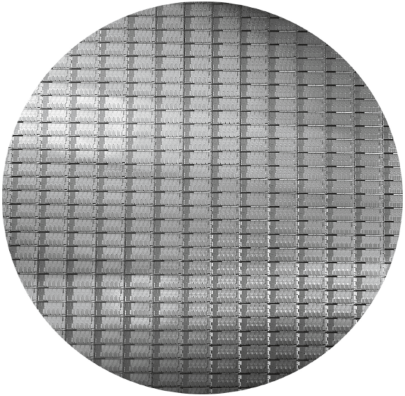
- Silicon: semiconductor
- Add materials to transform properties:
  - Conductors
  - Insulators
  - Switch

Manufacturing ICs



- Yield: proportion of working dies per wafer

Intel Core i7 Wafer



- 300mm wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm

- We will be using Verilog throughout the course
- I ***highly*** recommend Virtualizing Ubuntu and running everything from there.
- **Homework:** Please download iVerilog (Icarus Verilog) and be ready to start coding next class
  - Icarus Verilog: Open source/free Verilog tool chain
  - Linux: use your package manager or `http://iverilog.icarus.com/`
  - Windows: `http://bleyer.org/icarus/`
- Verilog is a industry language used to design processors
- We will be building a simplified x86 processor
  - x86 = 32-bit processor, which is the root of virtually all PCs today.
  - modern day CPUs use x86_64, a 64-bit version of x86.