Introduction

EE685, Fall 2025

Hank Dietz

`http://aggregate.org/hankd/`

University of Kentucky

Course Overview

- Sort-of like CPE380/480, but broader & deeper

- **There will be *Verilog*, and you'll design stuff**

- A lot more of the *advanced* stuff
  - Fancy things inside processors
  - Lots of memory & parallel architecture

- Higher-level models and/or simulation

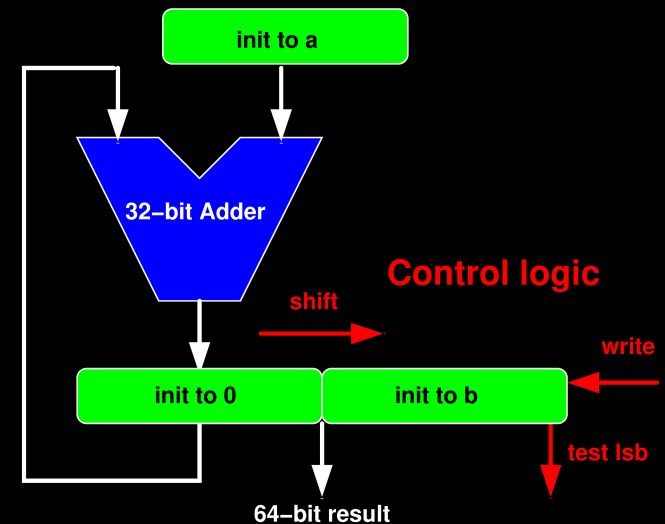# Verilog 32-bit Multiplier

```verilog
module mul(ready, c, a, b, reset, clk);

parameter BITS = 32;
input [BITS-1:0] a, b;
input reset, clk;
output reg [BITS*2-1:0] c;
output reg ready;
reg [BITS-1:0] d;
reg [BITS-1:0] state;
reg [BITS:0] sum;

always @(posedge clk or posedge reset) begin
  if (reset) begin
    ready <= 0;
    state <= 1;
    d <= a;
    c <= {{BITS{1'b0}}, b};
  end else begin
    if (state) begin
      sum = c[BITS*2-1:BITS] + d;
      c <= (c[0] ? {sum, c[BITS-1:1]} :
            (c >> 1));
      state <= {state[BITS-2:0], 1'b0};
    end else begin
      ready <= 1;
    end
  end
end
endmodule
```
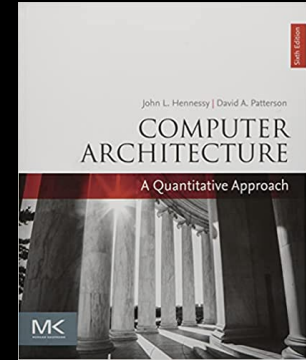
Textbook

- The text is:
  *Computer Architecture, 6th Edition:
  A Quantitative Approach,*
  by Patterson & Hennessy

- Same text as was used in CPE480…
  and *optionally* referenced here

- Lots of additional materials…

Grading & Such

- Four assignments, ~10% each
- Midterm exam, ~20%
- Final exam, ~40%

- Material cited from the text, from lectures, from canvas, or from the course URL: **http://aggregate.org/EE685/**

- You are expected to regularly attend class
- I try not to curve much; always in your favor

Course Content

- Precise content depends on you:
  - How many of you took CPE380? When?
  - How many of you took CPE480? When?

This course is sort-of **++CPE480**...

- The syllabus is posted, but it is a little vague because things will vary significantly based on the backgrounds of students in the course
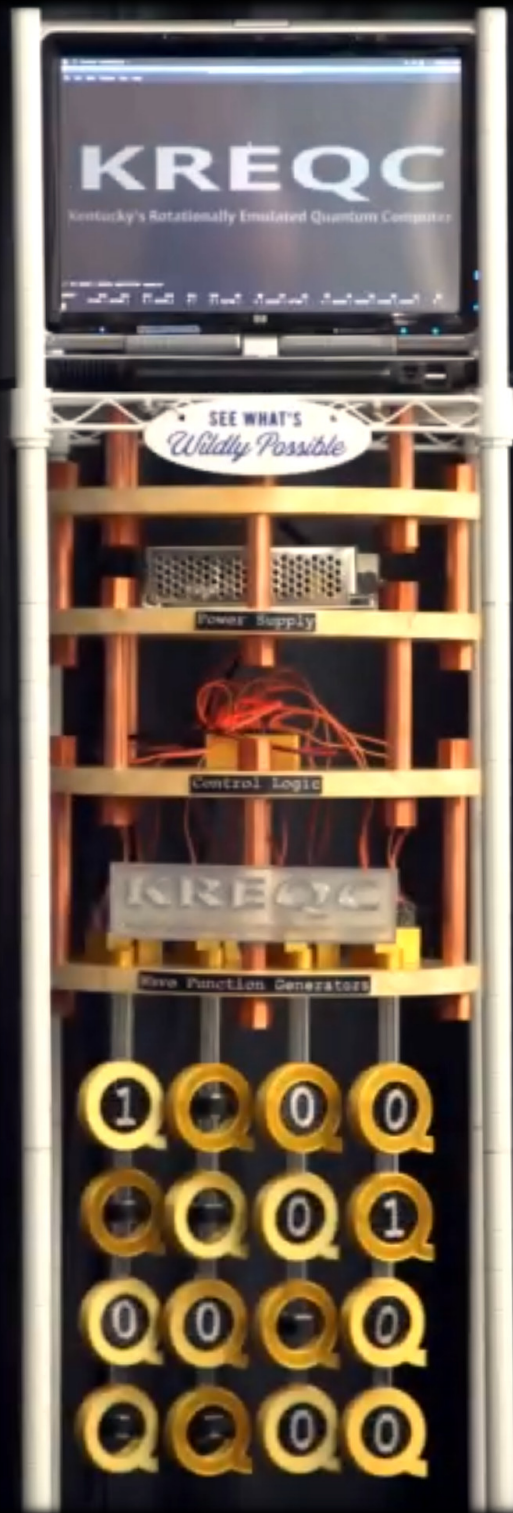
Course Content

Lectures	Topic
1	Introduction
3	Verilog (project)
6	Pipelined RISC machine (project)
3	Instruction-level parallelism
1	Review for midterm
1	In-class midterm exam
3	Advanced processor internals (assignment)
5	Memory hierarchy and protection (assignment)
4	Scalable parallel processing and networking
1	Review for final exam

Me (and why I'm biased)

- Hank Dietz, ECE Professor and
  James F. Hardymon Chair in Networking
- Office: 203 Marksbury
- Research in parallel compilers & architectures:
  - Built 1st Linux PC cluster supercomputer
  - Antlr, AFNs, SWAR, FNNs, MOG, …
  - Various awards & world records for best
    price/performance in supercomputing
- Lab: 108/108A Marksbury – I have **TOYS**!

KREQC
Kentucky's Rotationally Emulated Quantum Computer

SEE WHAT'S
Wildly Possible

Power Supply

Control Logic

KREQC

Wave Function Generators
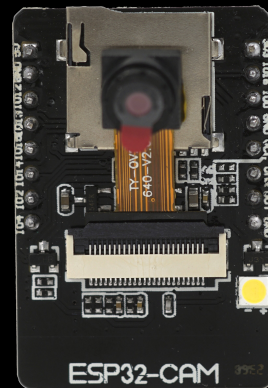
Electrical & Computer Engineering

UK
ECE

What is Advanced stuff?

El Capitan supercomputer:
11,039,616 cores, 2.746 Exaflop/s
Cost approx. $600M, 29.6 MW power
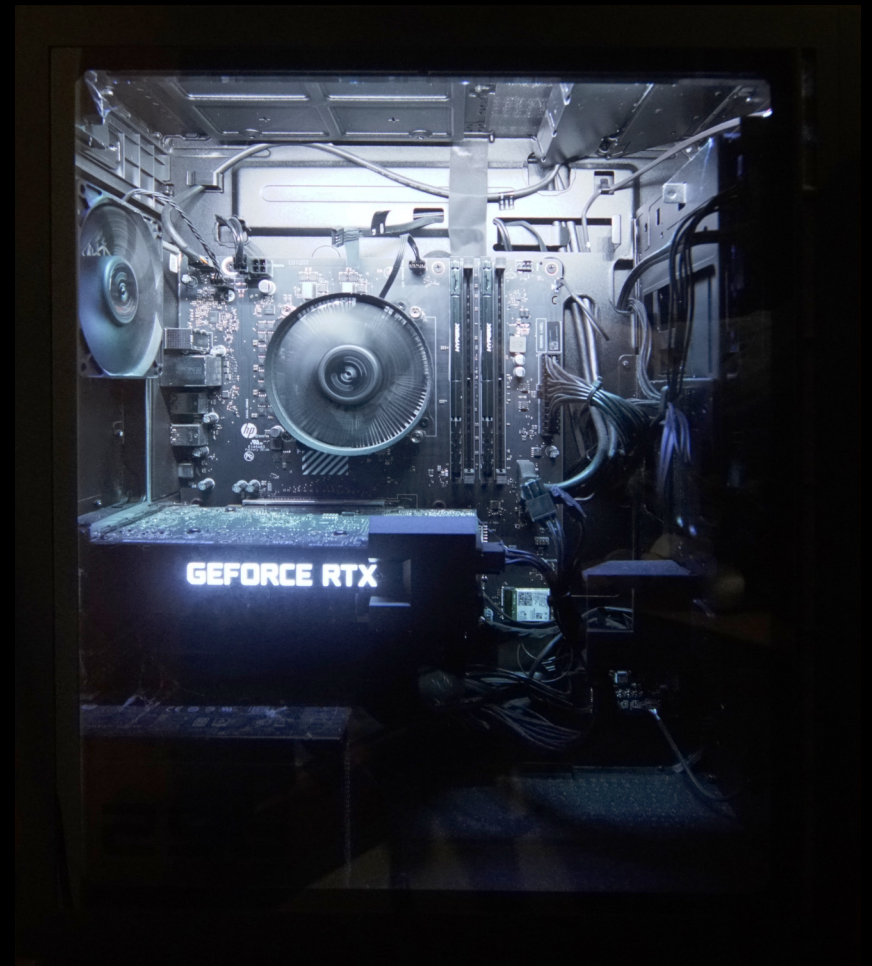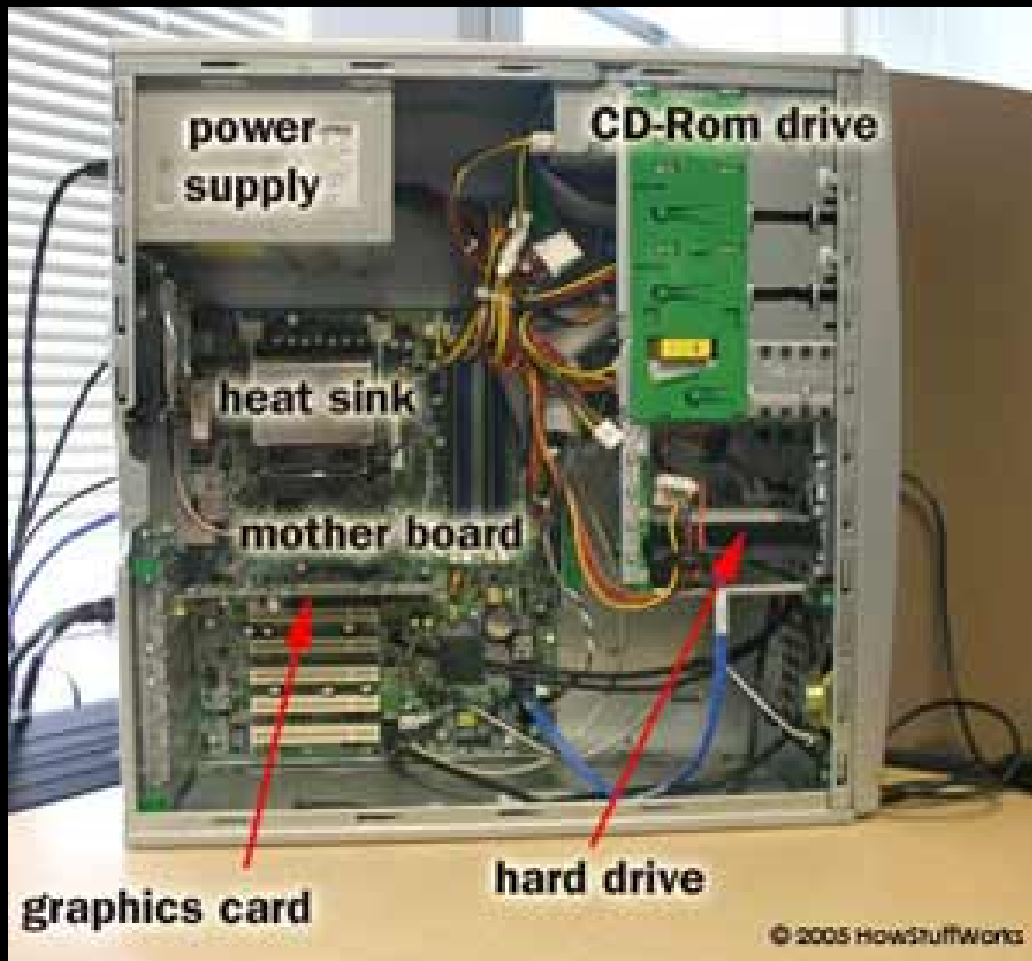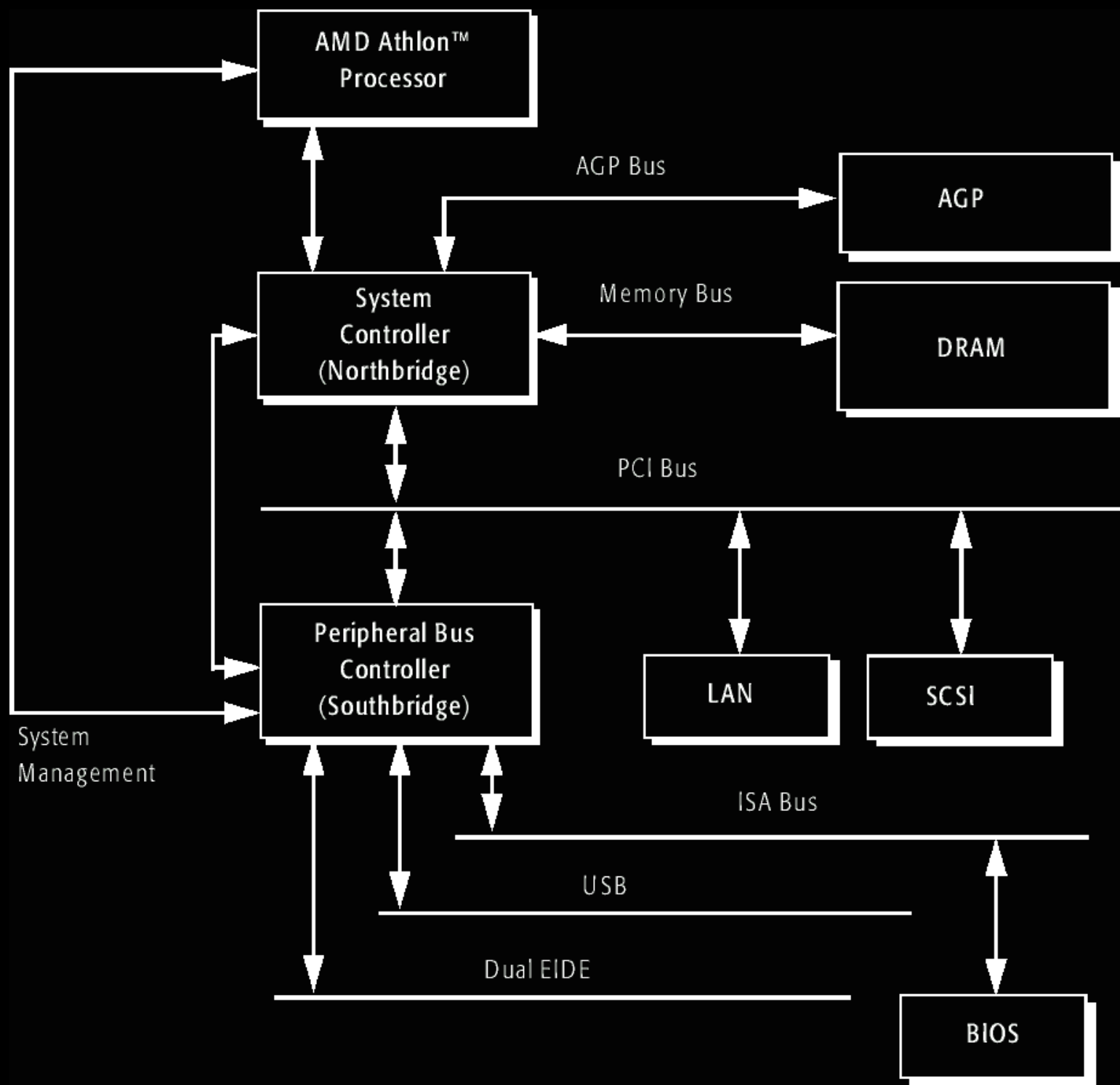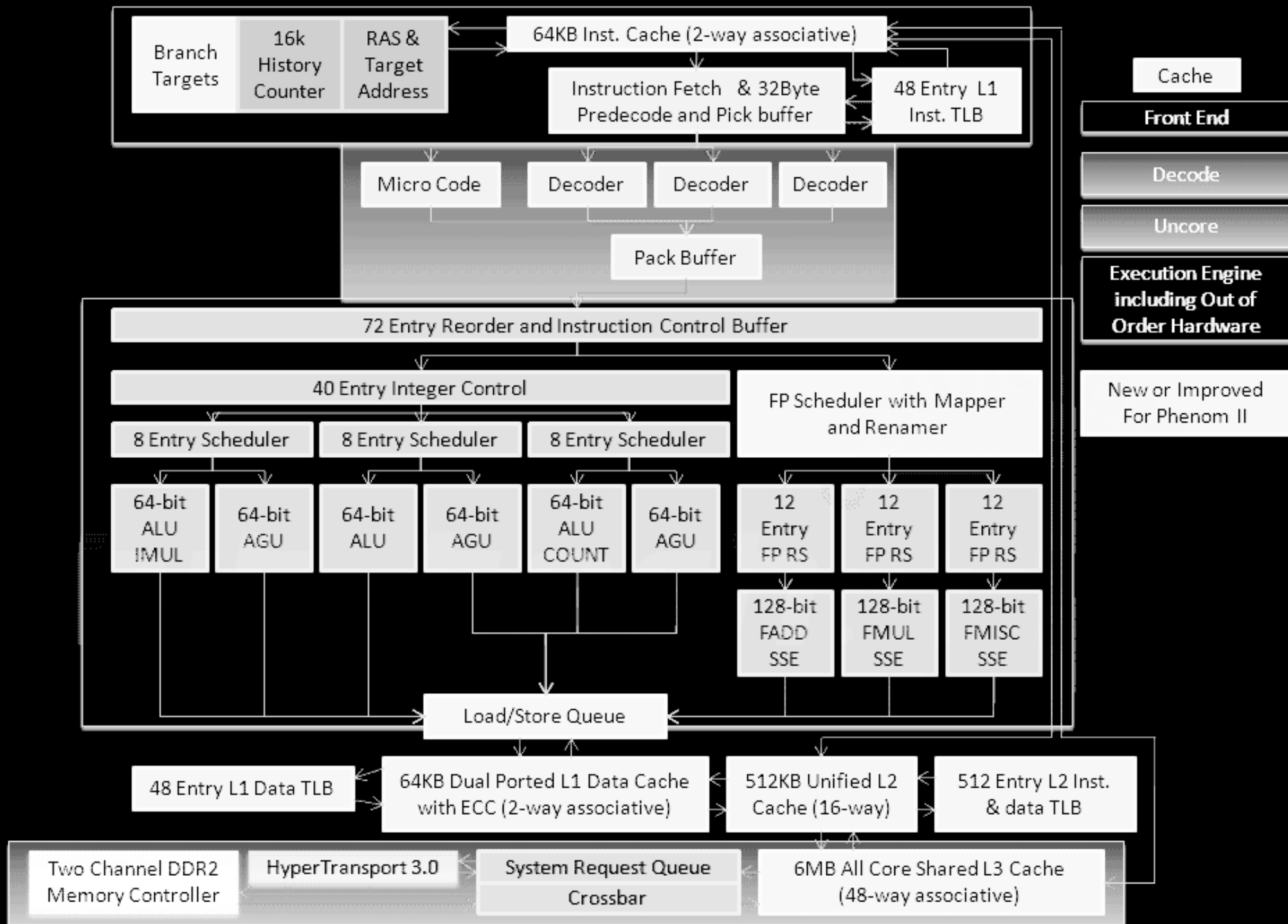
What is advanced stuff?

ESP32-CAM microcontroller:
2+ULP cores, 802.11+Bluetooth, TF, 2MP camera
Cost approx. $7, 5μA sleep current

What's inside a PC?

power supply — heat sink — mother board — graphics card — CD-Rom drive — hard drive

© 2005 HowStuffWorks

GEFORCE RTX

Decode Buffer

Data Registers

Write Buffer

Inst. L1 TLB

Inst. L2 TLB

Inst. L1 Cache
8–32KB, 2–3 cycles

Data L1 TLB

Data L2 TLB

Data L1 Cache
8–32KB, 2–3 cycles

Shared Inst./Data L2 Cache

Multi–Core Shared L3 Cache

Main Memory (DRAM): 2–64GB, ~1K cycles/reference

SSD, Disk, or Disk with SSD cache: 32GB–5TB, ~10K–10M cycles/reference

Cloud or other mass storage systems
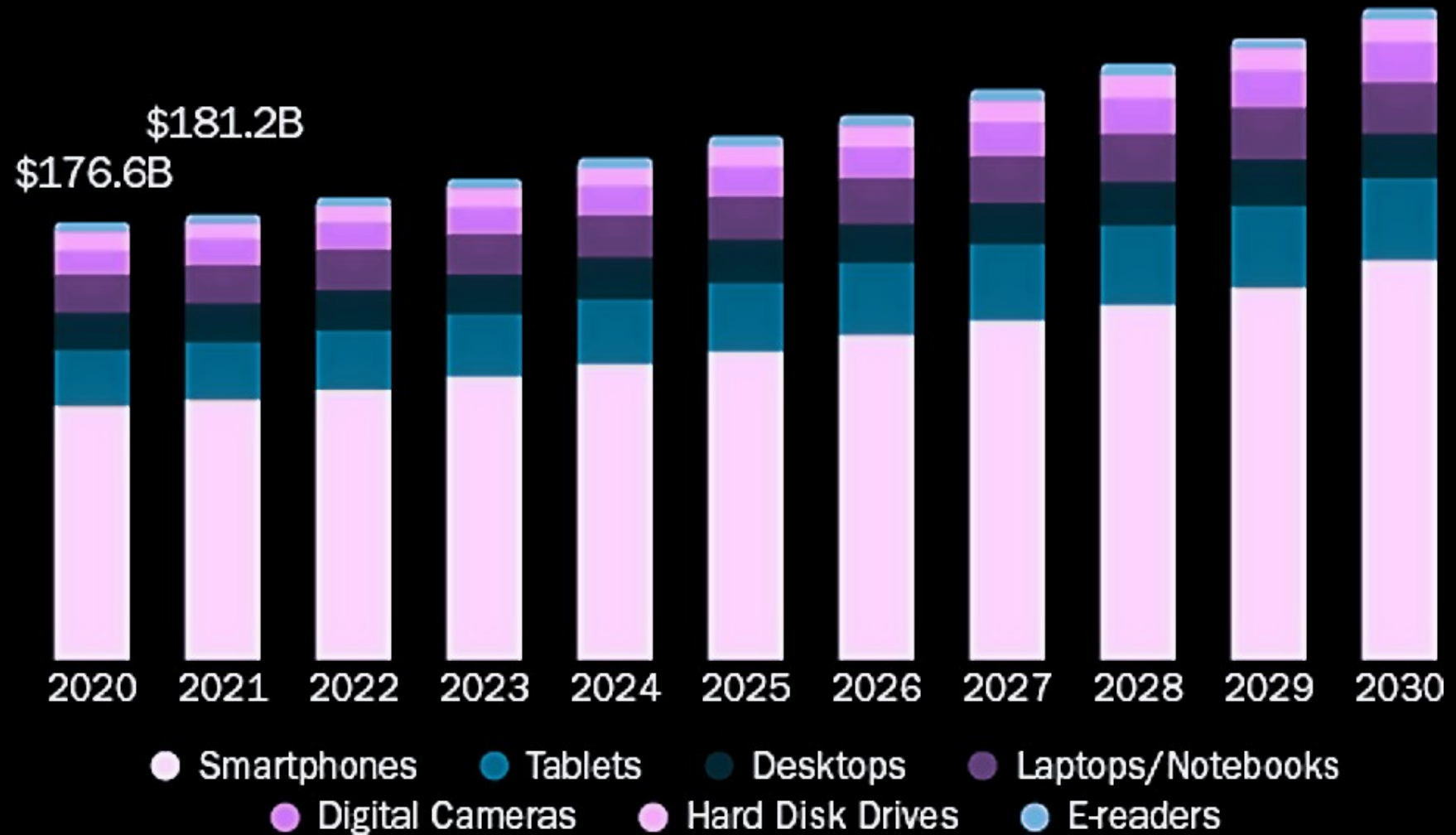
U.S. Consumer Electronics Market

Size, by Product, 2020 - 2030 (USD Billion)

$181.2B

$176.6B

2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030

● Smartphones ● Tablets ● Desktops ● Laptops/Notebooks
● Digital Cameras ● Hard Disk Drives ● E-readers

LCD

mother
board

processor

heat sink

graphics
chip

©2005 HowStuffWorks

Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

Computer board

FlashAir

W-02

Wireless LAN
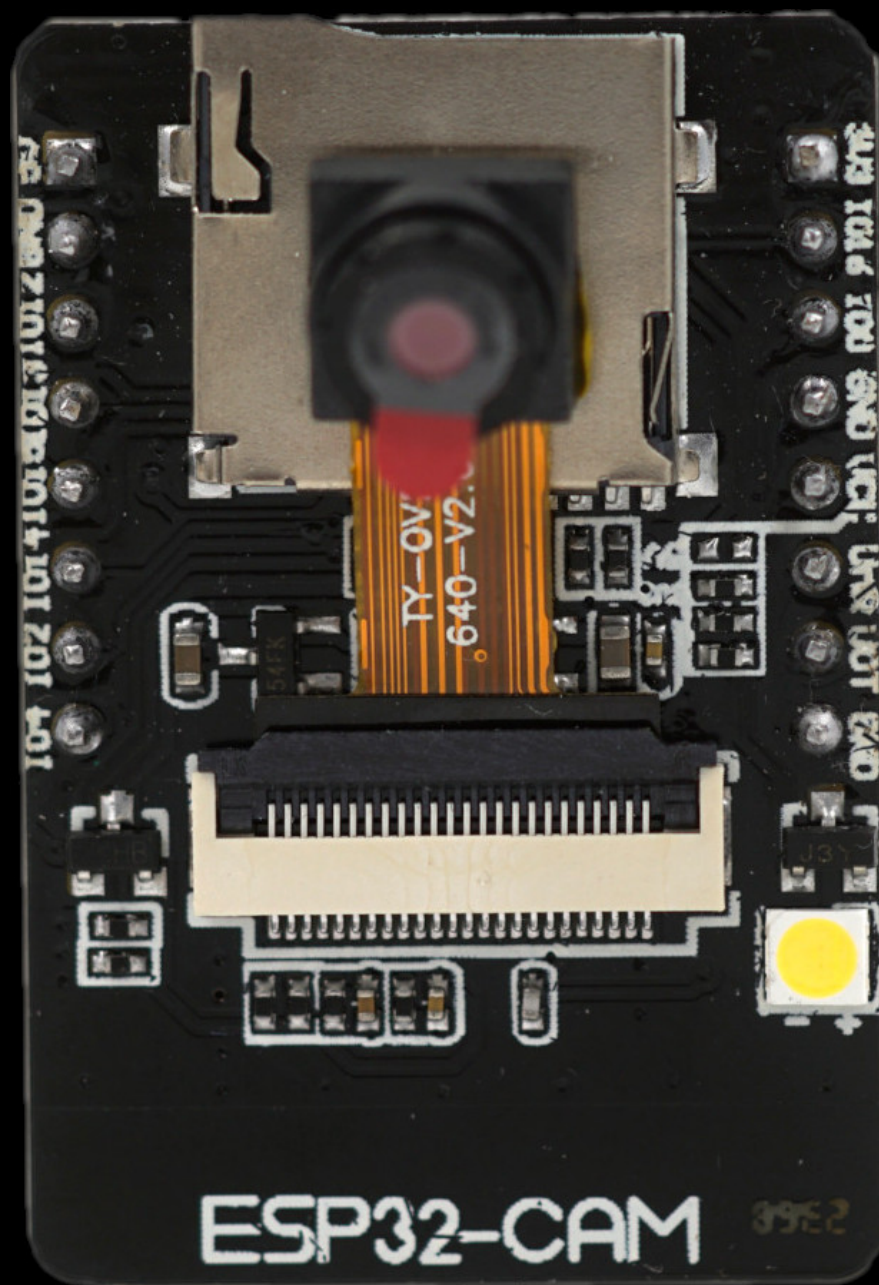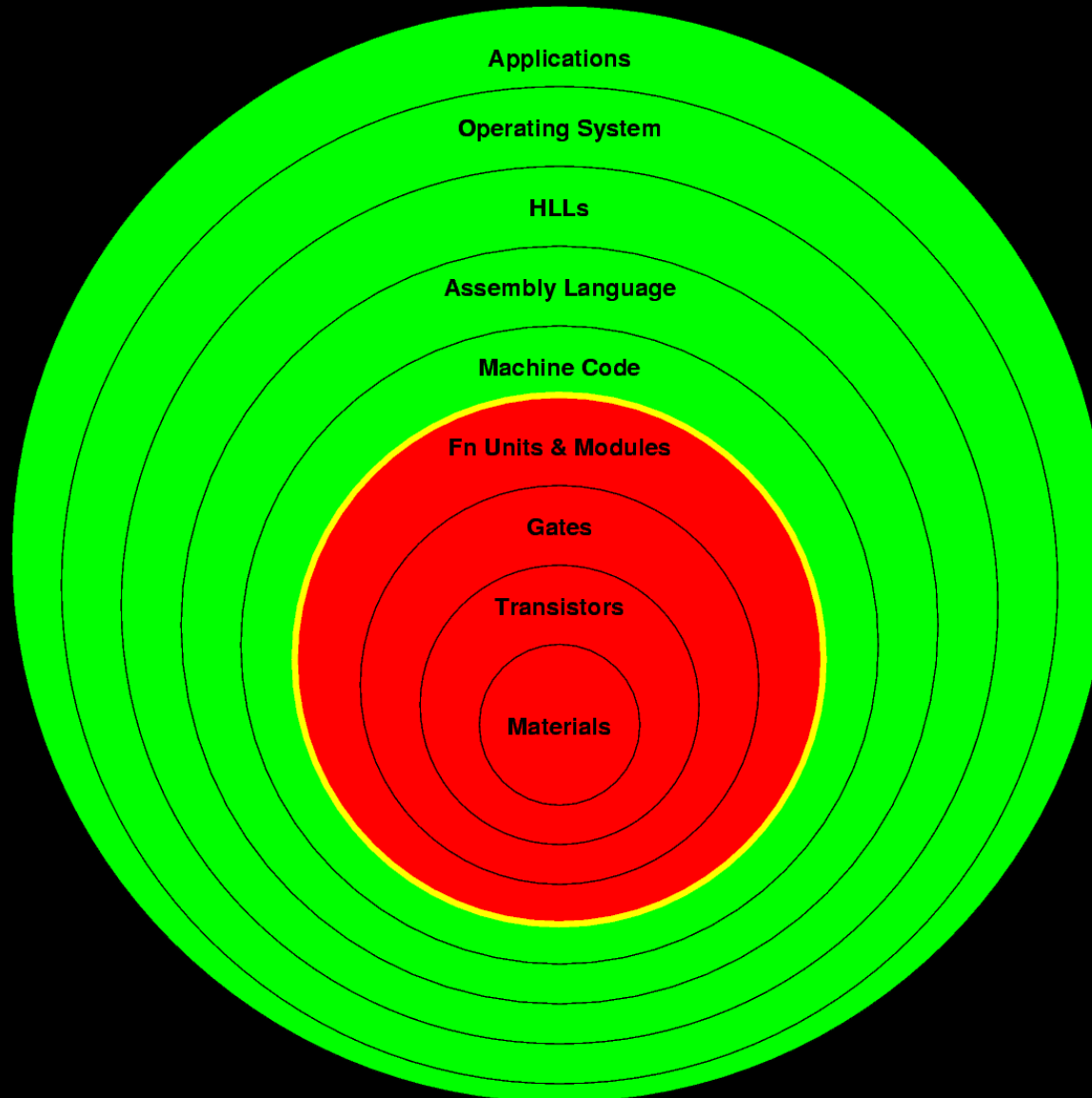
10 SDHC 32 GB

TOSHIBA

ESP32-CAM

Complexity is Increasing!

- Lots of things you use every day have **BILLIONS** of components!

- You don't live long enough to know it all

Abstraction "Onion"

Applications

Operating System

HLLs

Assembly Language

Machine Code

Fn Units & Modules

Gates

Transistors

Materials

Software Layers

- Applications...
- Operating Systems (OS)...
- High-Level Languages (HLLs)
  Aka, High Order Languages (HOLs)
  - Designed for humans to write & read
  - Modularity
  - Abstract data types, type checking
  - Assignment statements
  - Control constructs
  - I/O statements

Instruction Set Architecture

- ISA defines HW/SW interface
- Assembly Language
  - Operations match hardware abilities
  - Relatively simple & limited operations
  - Mnemonic (human readable?)
- Machine Language
  - Bit patterns – 0s and 1s
  - Actually executed by the hardware

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;
}
```

Compiler

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
```

Assembler

Binary machine language program (for MIPS)

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

Hardware Layers

- Function-block organization (CPE380 stuff)
- Gates & Digital Logic (CPE282 stuff)
- Transistors
    - Used as bi-level (saturated) devices
    - Amplifiers, not just on/off switches
- Materials & Integrated Circuits
    - Implementation of transistors, etc.
    - Analog properties

Who Does What?

- Instruction Set Design, by *Architect*
  - Machine & Assembly Languages
  - "**Computer Architecture**"
  - **Instruction Set Architecture** / Processor
- Computer Hardware Design, by *Engineer*
  - Logic Design & Machine Implementation
  - "**Processor Architecture**"
  - "Computer Organization"

How does EE685 see this?

- A lot like CPE380/CPE480…

- Like CPE380, Verilog runs through it

- Focus is on the hardware/software interface
  - Models that guide design and use
  - Modules that enable or boost performance

SI Terminology Of Scale

1000^1	kilo	k	1000^-1	milli	m
1000^2	mega	M	1000^-2	micro	u
1000^3	giga	G	1000^-3	nano	n
1000^4	tera	T	1000^-4	pico	p
1000^5	peta	P	1000^-5	femto	f
1000^6	exa	E			

- 1000^x vs. 1024^x
- 1 Byte (B) is 8-10 bits (b), 4 bits in a Nybble
- Hertz (Hz) is frequency (vs. period)

Main topics for EE685

- Start with Verilog
- Review pipelined RISC (where CPE380 ends)
- Add instruction-level parallelism (SWAR, VLIW, SuperScalar...)
- Add advanced processor internals (floating point, interrupts)
- Add memory heirarchy (caches, virtualization)
- Add scalable parallelism (SIMD/GPU, MIMD, networking)