Introduction

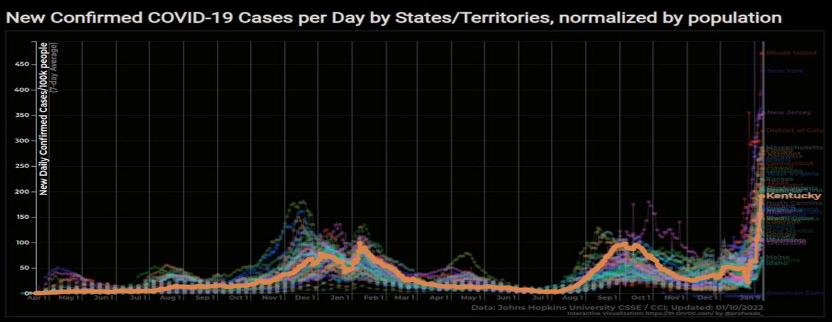
CPE380/CS380, Spring 2022

Hank Dietz

http://aggregate.org/hankd/

University of Kentucky

Class Meetings



- We are scheduled to meet in person, and that is the plan for the semester… *as I write this*
- **Masks must be worn for in-person meetings**
- Omicron is ***much more easily transmitted*** than Alpha or Delta strains
  - If you might have COVID19, get tested
  - If you are under quarantine, stay home
  - Quarantine is an excused absence, and we will help you keep up with recordings, etc.

Course Overview

- You know how to write a simple program...
  from CS courses
- You know how to build simple combinatorial
  and sequential logic circuits...
  from EE courses (especially CPE282)
- This course fills the gap between the two:
  - So you can better **specify & use** that stuff
  - So you can **create** the stuff in between

Changes, as of Fall 2021

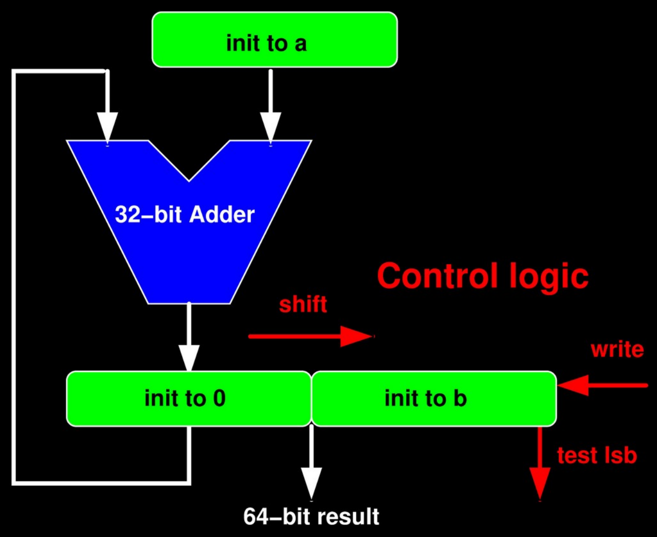
- There's now Freshman Engineering
- EE383 used to come after EE380…
  now, EE383 is EE287 and comes before it
- CS270/CPE282 overlaps some EE380 material;
  e.g., remember CPE282's uKY processor?
- EE380 used to be required for CS/EE/CPE...
  now, it's required only for Computer Engineers
- In sum, CPE380 will slowly change focus from **specify & use** to **create using Verilog**

Verilog 32-bit Multiplier

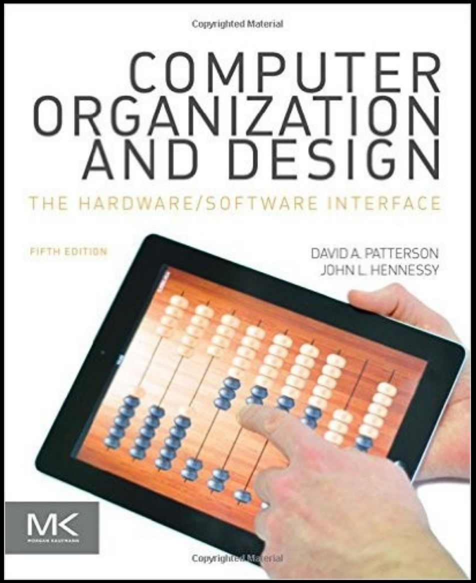
```
module mul(ready, c, a, b, reset, clk);

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 Organization & Design, 5^{th} Edition: The Hardware/Software Interface* by Patterson & Hennessy
- You can use any edition from 2^{nd} – 6^{th}, but we'll reference sections from the 5^{th}
- We will not assign problems from the text
- Lots of additional materials at the course URL and presented in class… text is reference only

Grading & Such

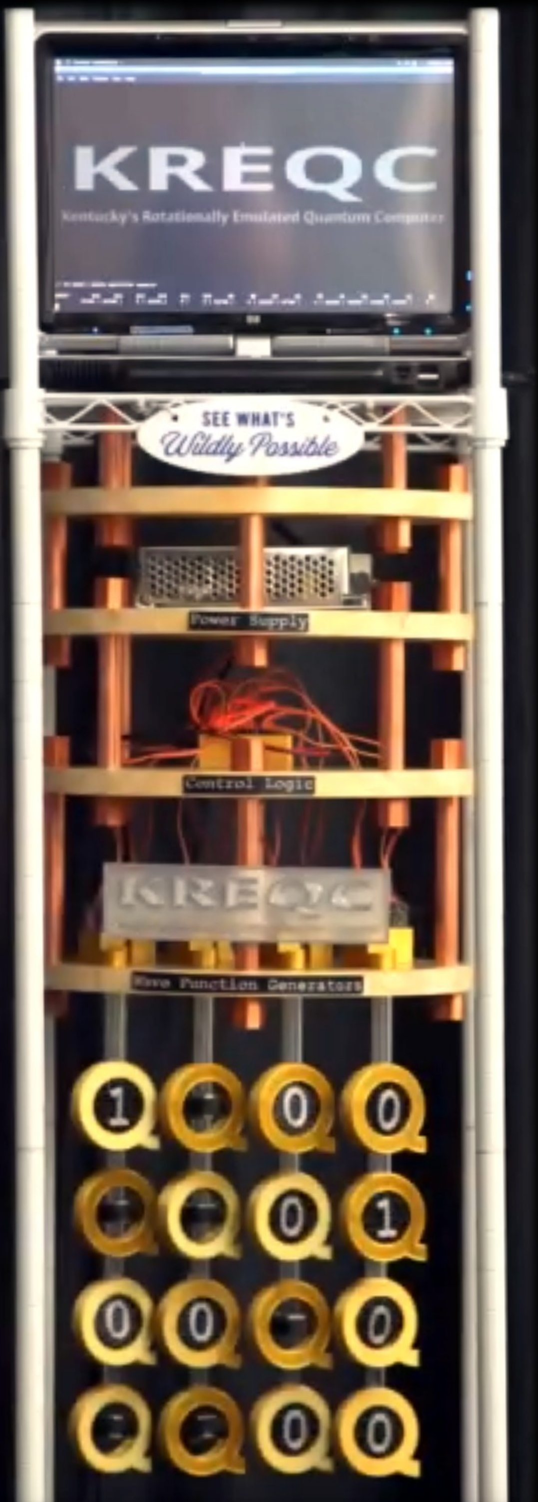
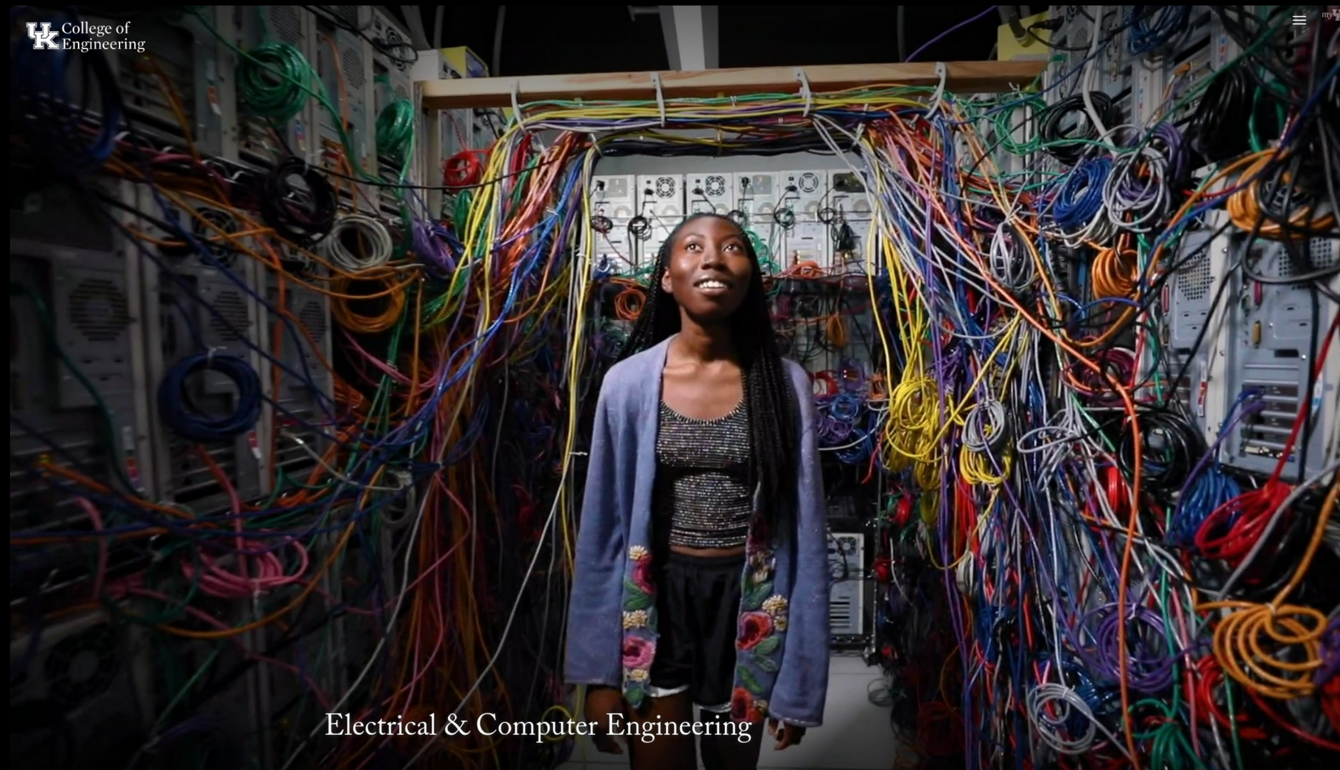
- Two online exams, ~15% each
- One final exam, ~30%
- Material cited from the text, from lectures, from canvas, or from the course URL:
  `http://aggregate.org/CPE380/`
- Homework via WWW forms, ~40%
  (there might be some 2% in-class quizzes)
- I try not to curve much; always in your favor

Course Content

Topic	Lectures	Exam
Introduction	3	0
A simple multi-cycle machine	4	0
Performance evaluation	2	0
Review for Exam 0		
Machine and assembly language (generic, SIMD/MIMD, MIPS)	4	1
Arithmetic… integer and floating point, with Verilog implementations	4	1
Single-cycle machine design	2	1
Review for Exam 1		
Pipelined machine design	4	2
Memory hierarchy and I/O	3	2
Parallel processing, advanced architecture, including lab tour	2	2
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**!

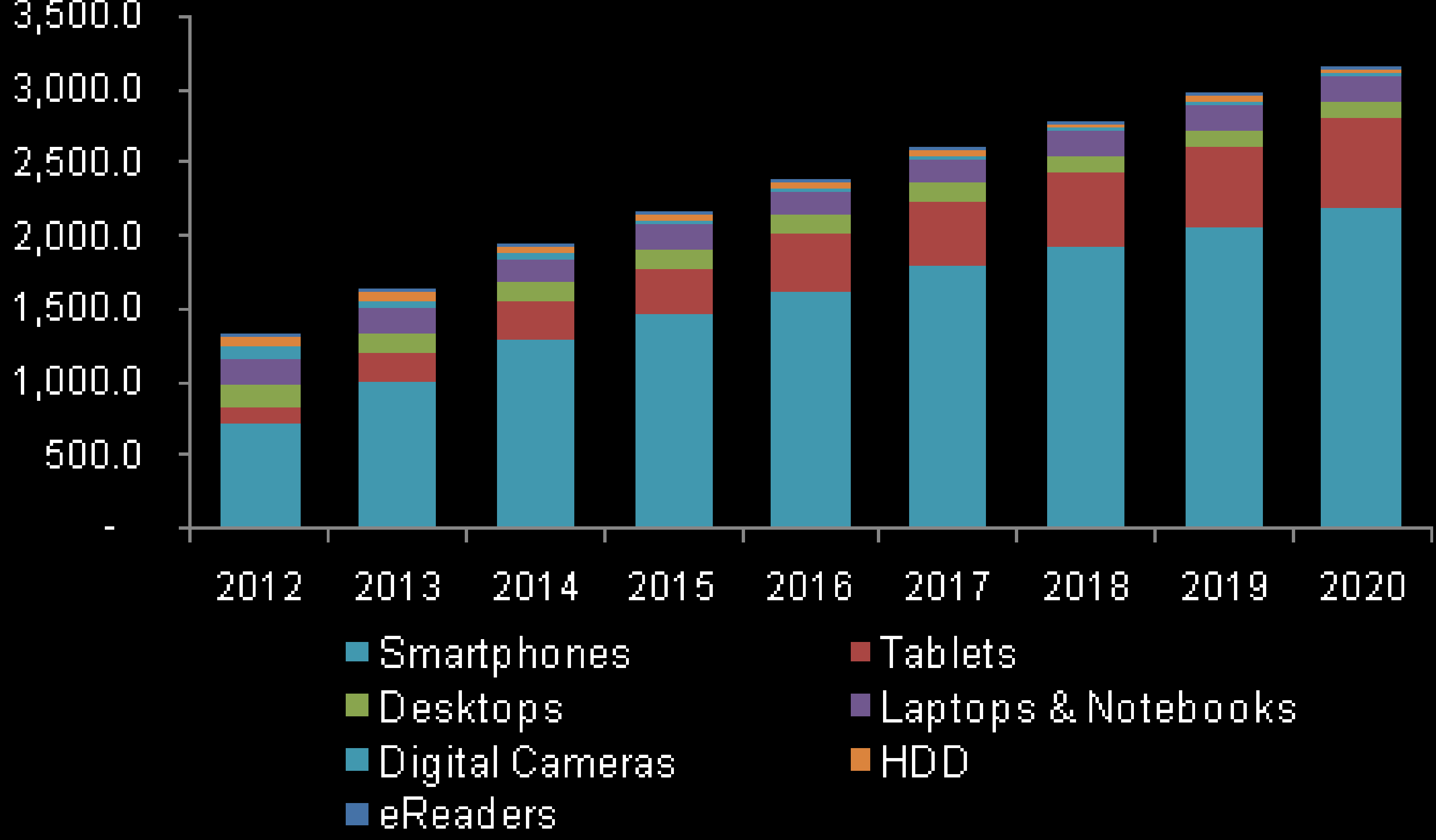

KREQC
Kentucky's Rotationally Emulated Quantum Computer
SEE WHAT'S
Wildly Possible
Power Supply
Control Logic
KREQC
Wave Function Generators

College of Engineering
Electrical & Computer Engineering

UK
ECE

Let's Talk About Computers

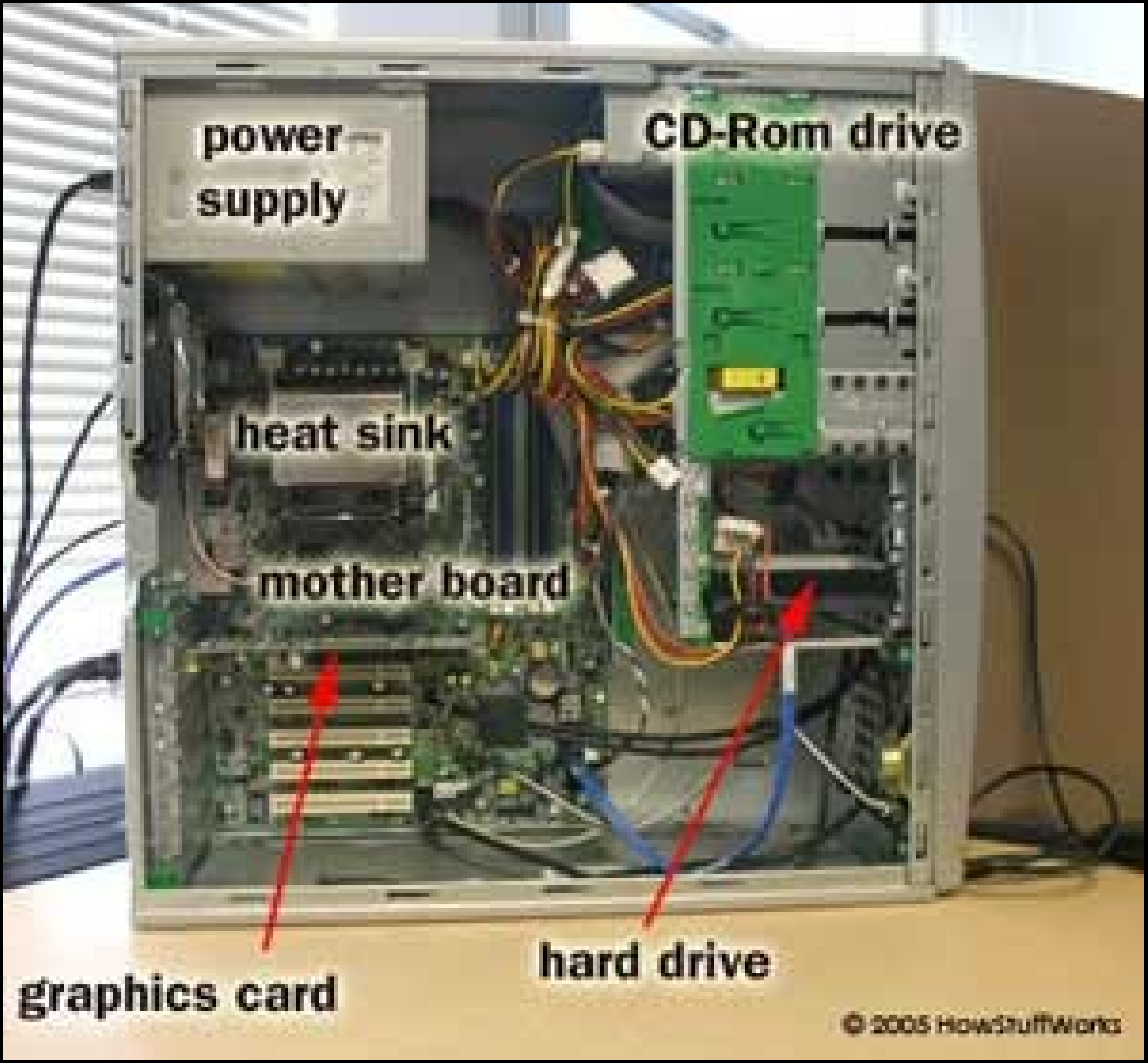
- Embedded computers, IoT (Internet of Things)
- Personal Mobile Devices (PMDs)...
  usually "smart phones" and tablets
- Personal Computers (PCs)
- Servers
- Supercomputers
- Clusters, Farms, Grids, and Clouds
  (Warehouse Scale Computers – WSC,
  Software as a Service – SaaS)

M-Unit Sales, Global Personal Electronics

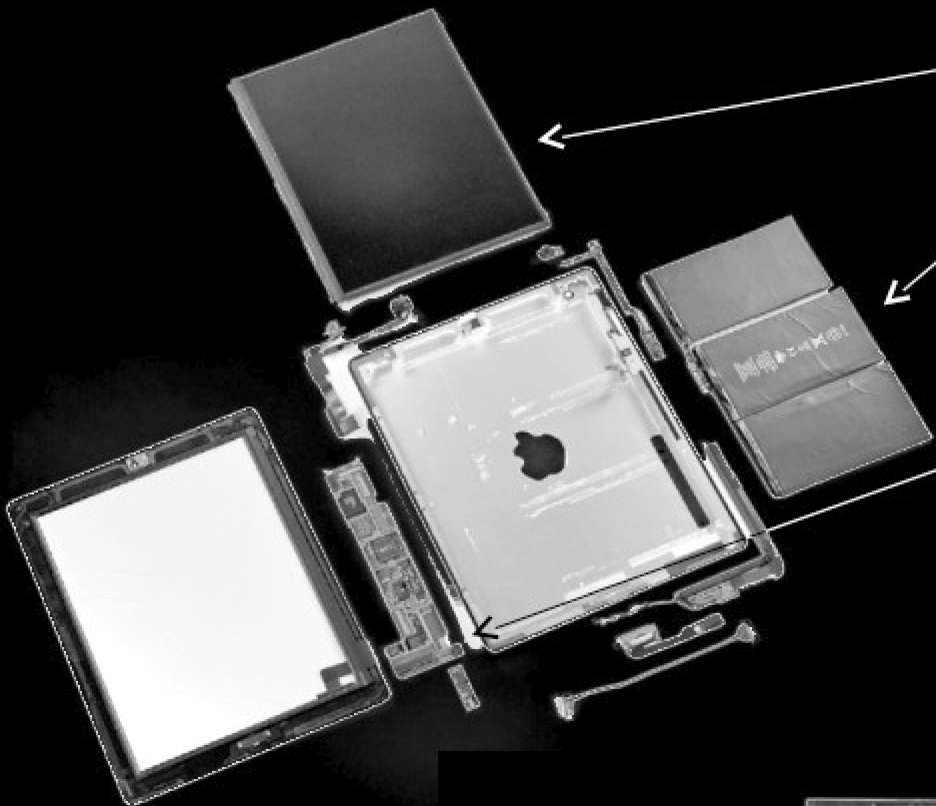


https://www.grandviewresearch.com/industry-analysis/personal-consumer-electronics-market

What's Inside?


power supply
CD-Rom drive
heat sink
mother board
graphics card
hard drive
© 2005 HowStuffWorks

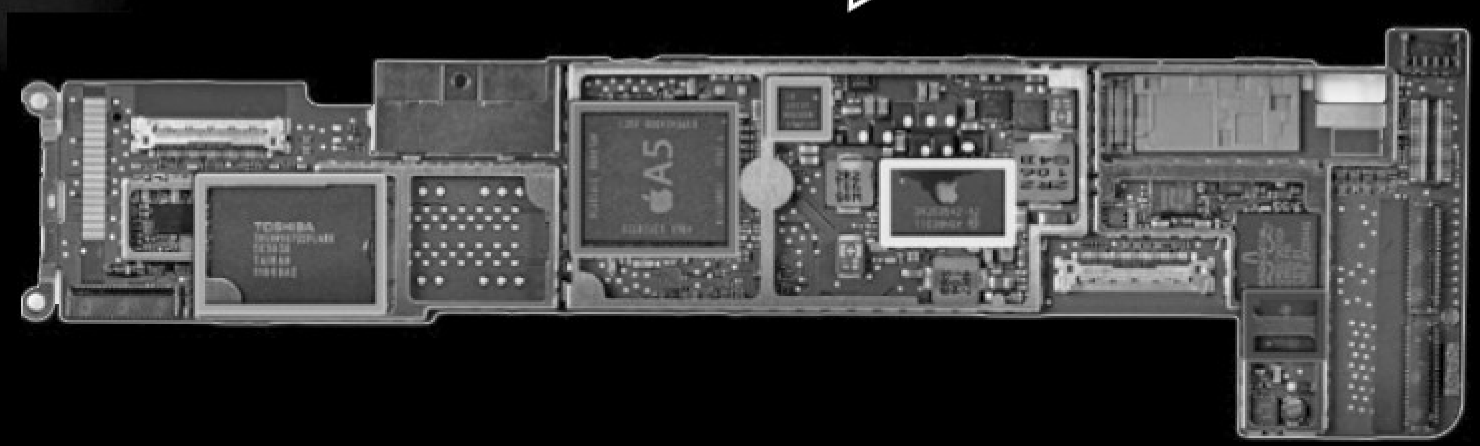

LCD
heat sink
mother
board
processor
graphics
chip
©2005 HowStuffWorks



Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

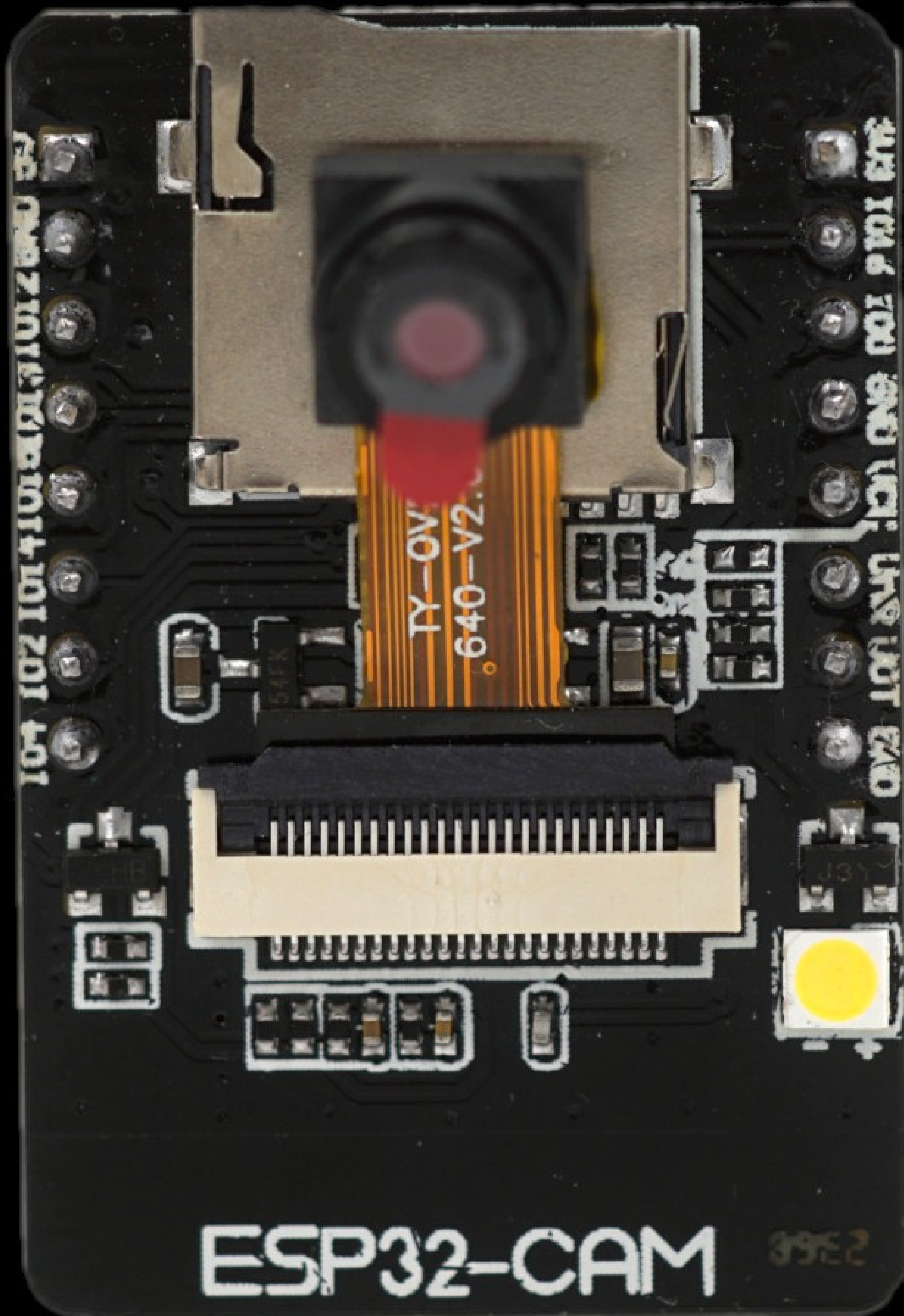
Computer board

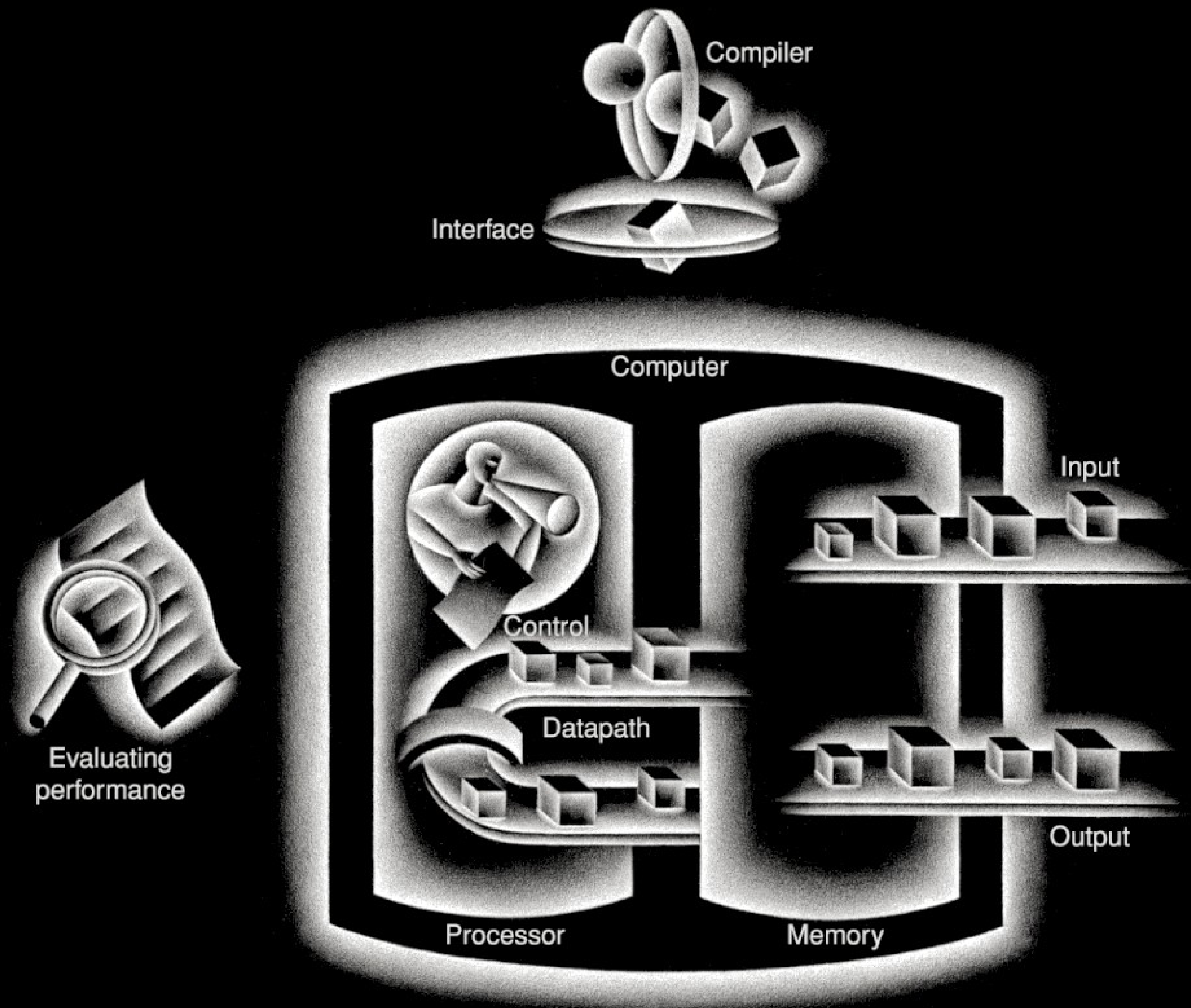


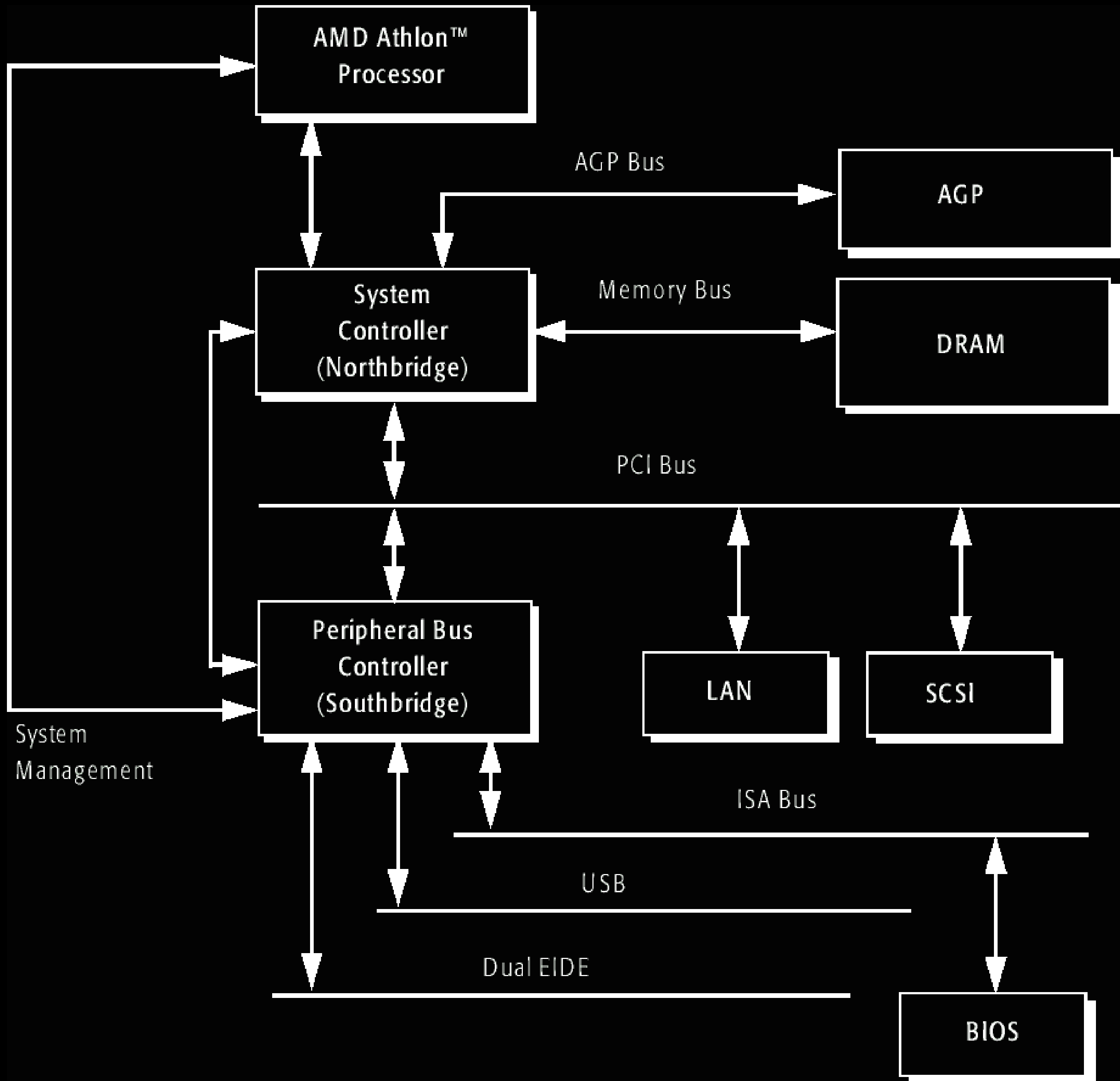

DiGiC4


Canon
8x
Canon


FlashAir
W-02
Wireless LAN
10
SDHC
32 GB
TOSHIBA


ESP32-CAM


Compiler
Interface
Computer
Input
Evaluating
performance
Control
Datapath
Output
Processor
Memory


AMD Athlon™ Processor
AGP Bus
AGP
System Controller (Northbridge)
Memory Bus
DRAM
PCI Bus
Peripheral Bus Controller (Southbridge)
LAN
SCSI
System Management
ISA Bus
USB
Dual EIDE
BIOS

Memory Terminology

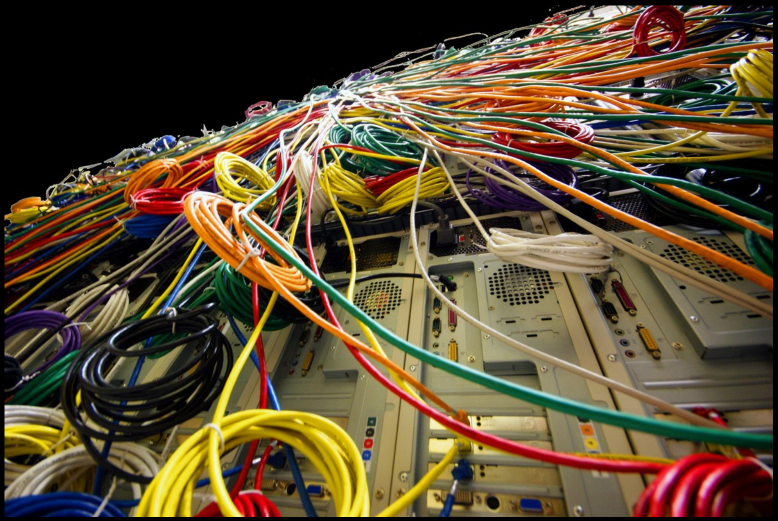
- Volatile – power off, data fades away
- ROM – non-volatile Read Only Memory
- PROM, EPROM, OTP, EEROM, Flash, 3DXPoint – types of non-volatile programmable memory
- RAM – volatile Random Access Memory
  - SRAM – Static RAM, fast but big cells
  - DRAM – Dynamic RAM, slow but small cells
  - EDO, SDRAM, DDR, RamBus – DRAM types
- Core – non-volatile magnetic RAM technology
- Registers, Cache – fast working memories

More Memory Terminology

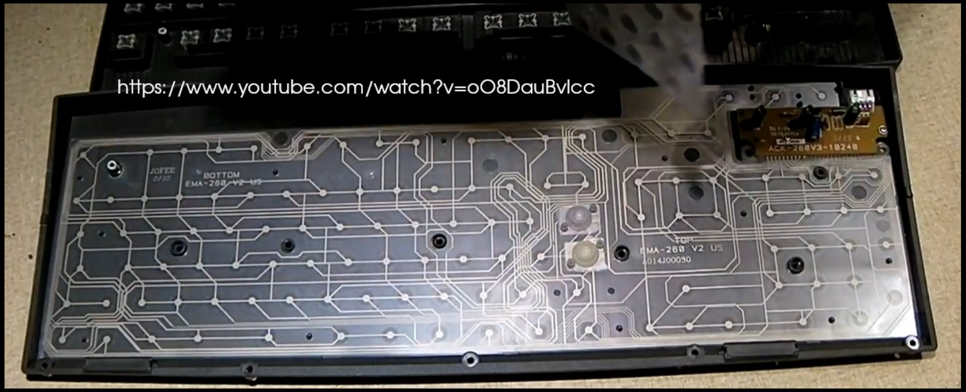
- Punched cards
- Punched paper tape
- Tape, Magtape
- Drum
- Disks:
  Floppy, Hard, Magneto-optical, CD (-R, -RW), DVD (+/-R, +/-RW, -RAM), Blu-ray
- Solid State Disk, Optane

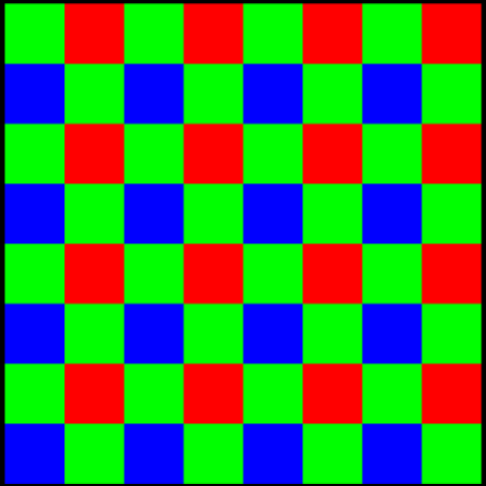
Network Terminology

- SAN, LAN, MAN, WAN – Area Network; System/Storage, Local, Metropolitan, Wide
- Ethernet, DSL (Digital Subscriber Line)
- USB, FireWire
- Hub, Switch, Router
- WiFi, Bluetooth, NFC
- Bandwidth, Latency



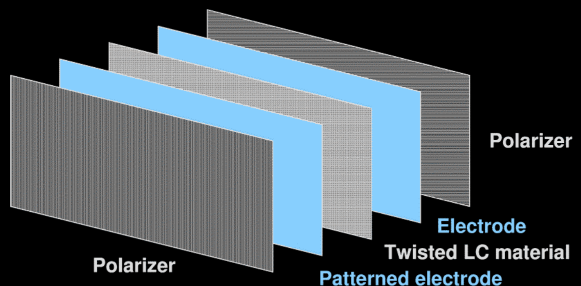
Other I/O Terminology

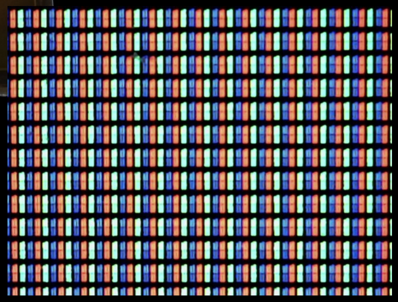


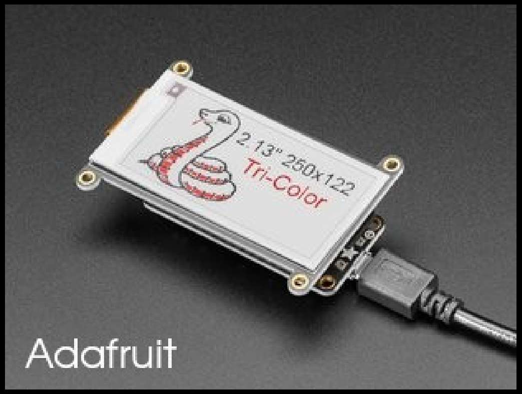


- Keyboard
- Mouse, Trackball, Touchscreen, Lightpen, Touchpad, etc.
- Pixel – Picture Element
- Camera: Charge-Coupled Device, CMOS

Other I/O Terminology





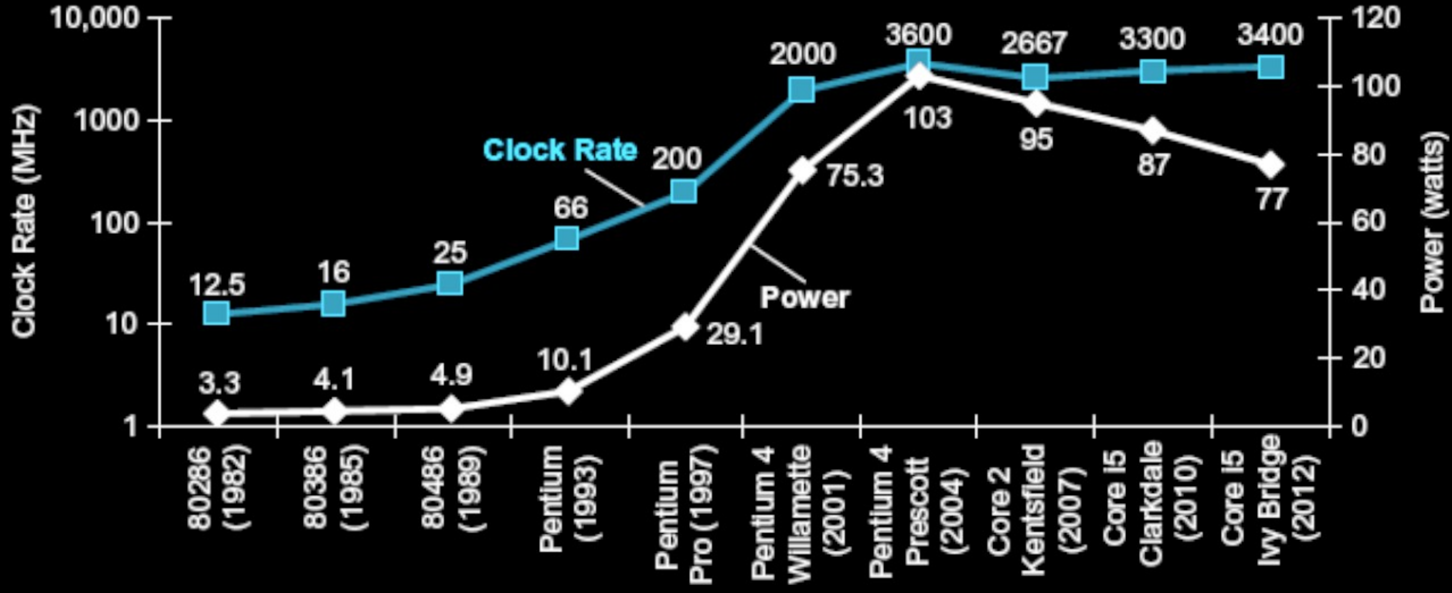


- Display: Cathode Ray Tube, Plasma, Liquid Crystal Display, Digital Micromirror Device aka Digital Light Processor, Organic Light Emitting Diode, eInk

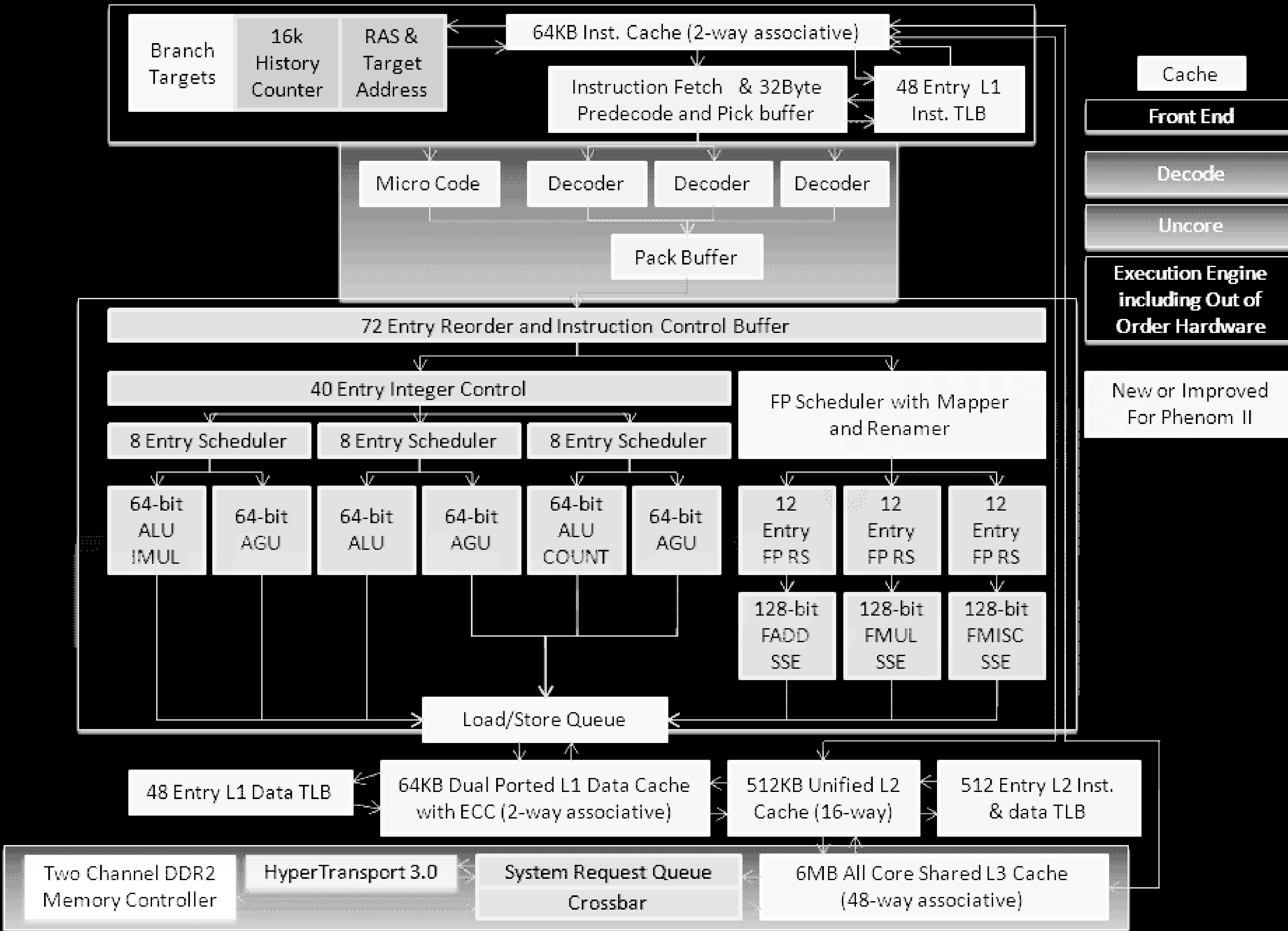
Processor Terminology

- CPU – Central Processing Unit
- PE, Core – Processing Element
- Processor – CPU or chip containing PEs
- "Computer Family" – same ISA
- IA32, x64 – ISAs based on the Intel 386
- MIPS, ARM, SPARC – other common ISAs
- DSP – Digital Signal Processor
- GPU – Graphics Processing Unit

Why Multi-Core?



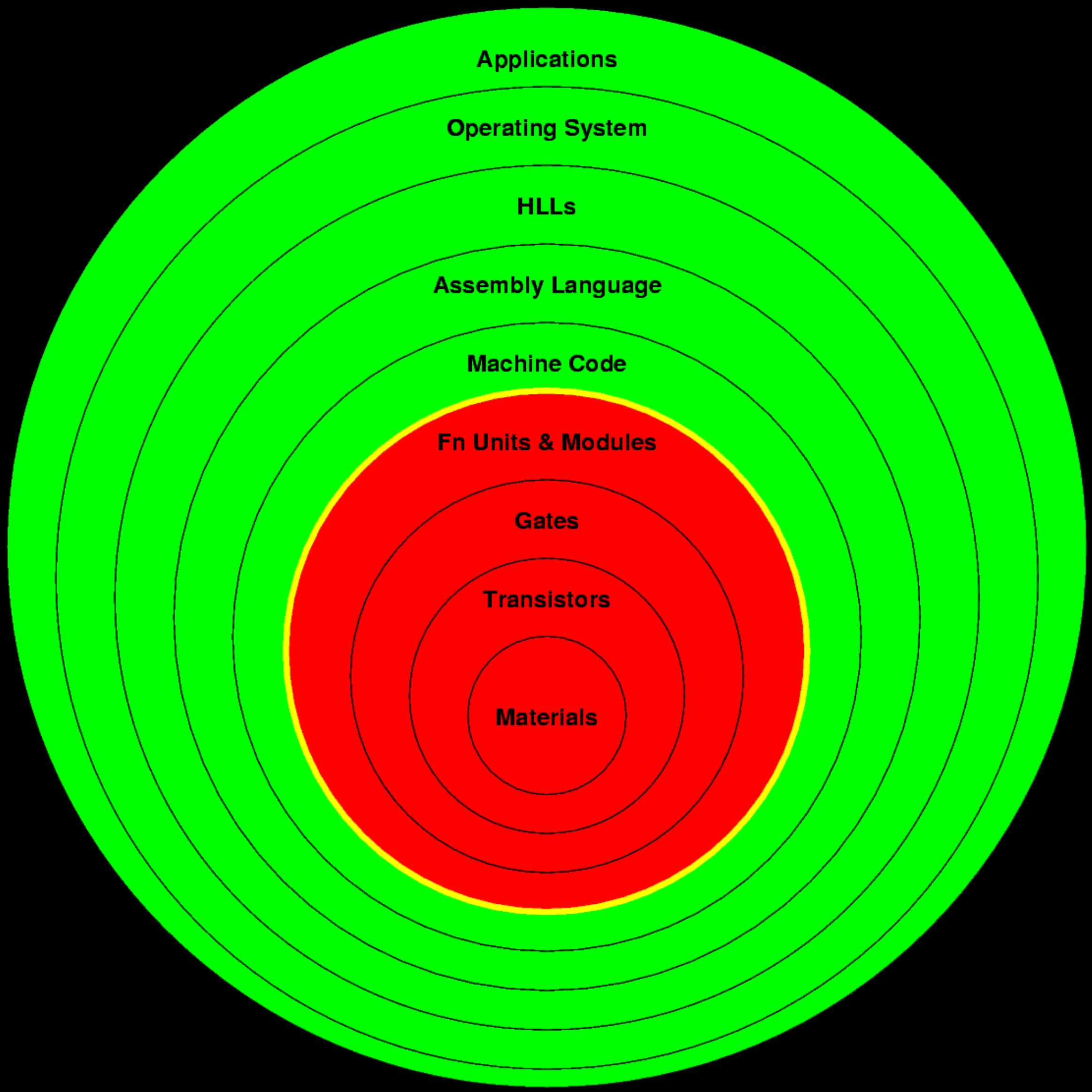
- Hit the "power wall"
- Lower voltage & slower clock reduce power more than performance
- Software companies changed license fees


Branch Targets
16k History Counter
RAS & Target Address
64KB Inst. Cache (2-way associative)
Instruction Fetch & 32Byte Predecode and Pick buffer
48 Entry L1 Inst. TLB
Micro Code
Decoder
Decoder
Decoder
Pack Buffer
72 Entry Reorder and Instruction Control Buffer
40 Entry Integer Control
FP Scheduler with Mapper and Renamer
8 Entry Scheduler
8 Entry Scheduler
8 Entry Scheduler
64-bit ALU IMUL
64-bit AGU
64-bit ALU
64-bit AGU
64-bit ALU COUNT
64-bit AGU
12 Entry FP RS
12 Entry FP RS
12 Entry FP RS
128-bit FADD SSE
128-bit FMUL SSE
128-bit FMISC SSE
Load/Store Queue
48 Entry L1 Data TLB
64KB Dual Ported L1 Data Cache with ECC (2-way associative)
512KB Unified L2 Cache (16-way)
512 Entry L2 Inst. & data TLB
Two Channel DDR2 Memory Controller
HyperTransport 3.0
System Request Queue
Crossbar
6MB All Core Shared L3 Cache (48-way associative)
Cache
Front End
Decode
Uncore
Execution Engine including Out of Order Hardware
New or Improved For Phenom II

Complexity

- Things are much more complex now
- Lots of things you use every day have **BILLIONS** of components!
- You don't live long enough to know it all

Abstraction "Onion"



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
- 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.
  - SSI, MSI, LSI, VLSI, ... WSI?

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 To Use Layers

- Things are too complex to “know everything”
- Need to know only layers adjacent
  - Makes design complexity reasonable
  - Makes things reusable
- Can tunnel to lower layers
  - For efficiency
  - For special capabilities

8 Great Ideas

- Design for Moore's Law
- Abstraction
- Make the common case fast
- Pipelining
- Parallelism
- Prediction
- Hierarchy of memories
- Dependability via redundancy

Computer Architecture Is Quickly Evolving

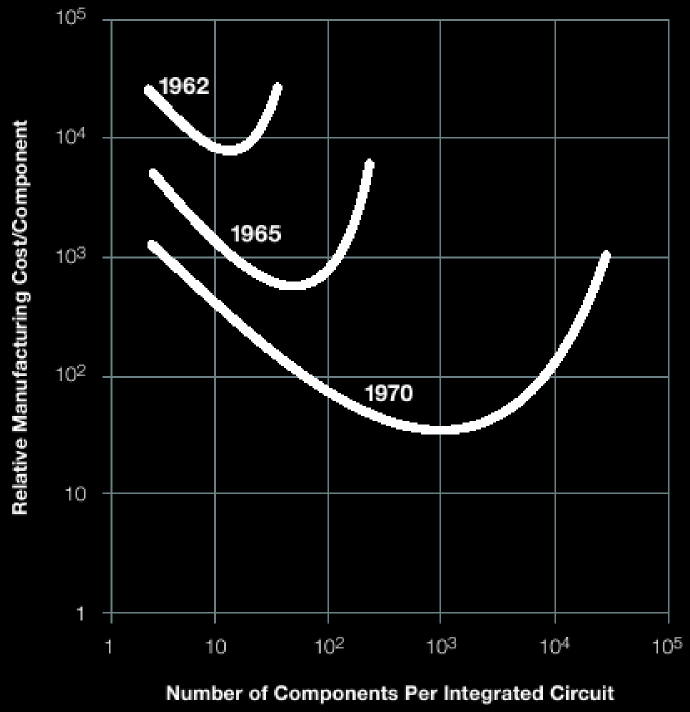
- Applications
  e.g., DVDs -> MMX, Doom -> 3DNow! & SSE;
  e.g., embedded systems, cell phones, etc.
- Programming Languages
  e.g., C -> call stack, flat memory addresses
- OperatingSystems
  e.g., Windows -> execute permission
- Technology
  e.g., Power density -> power management

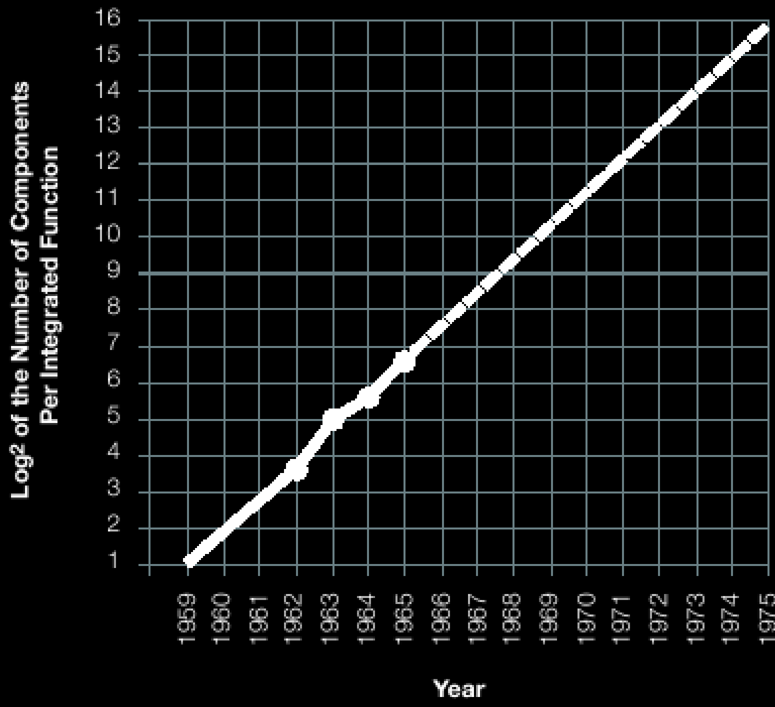
Chip Terminology

- Silicon Ingot – sausage-like single crystal
- Wafer – slice from above
- Die – one chip's area on a wafer
- Chip – a mounted die
- Yield – fraction that are good
- SSI, MSI, LSI, VLSI, WSI – Scale Integration; Small, Medium, Large, Very Large, Wafer

Moore's Law

"Cramming more components onto integrated circuits," Electronics, Vol. 38, No. 8, April 19, 1965.

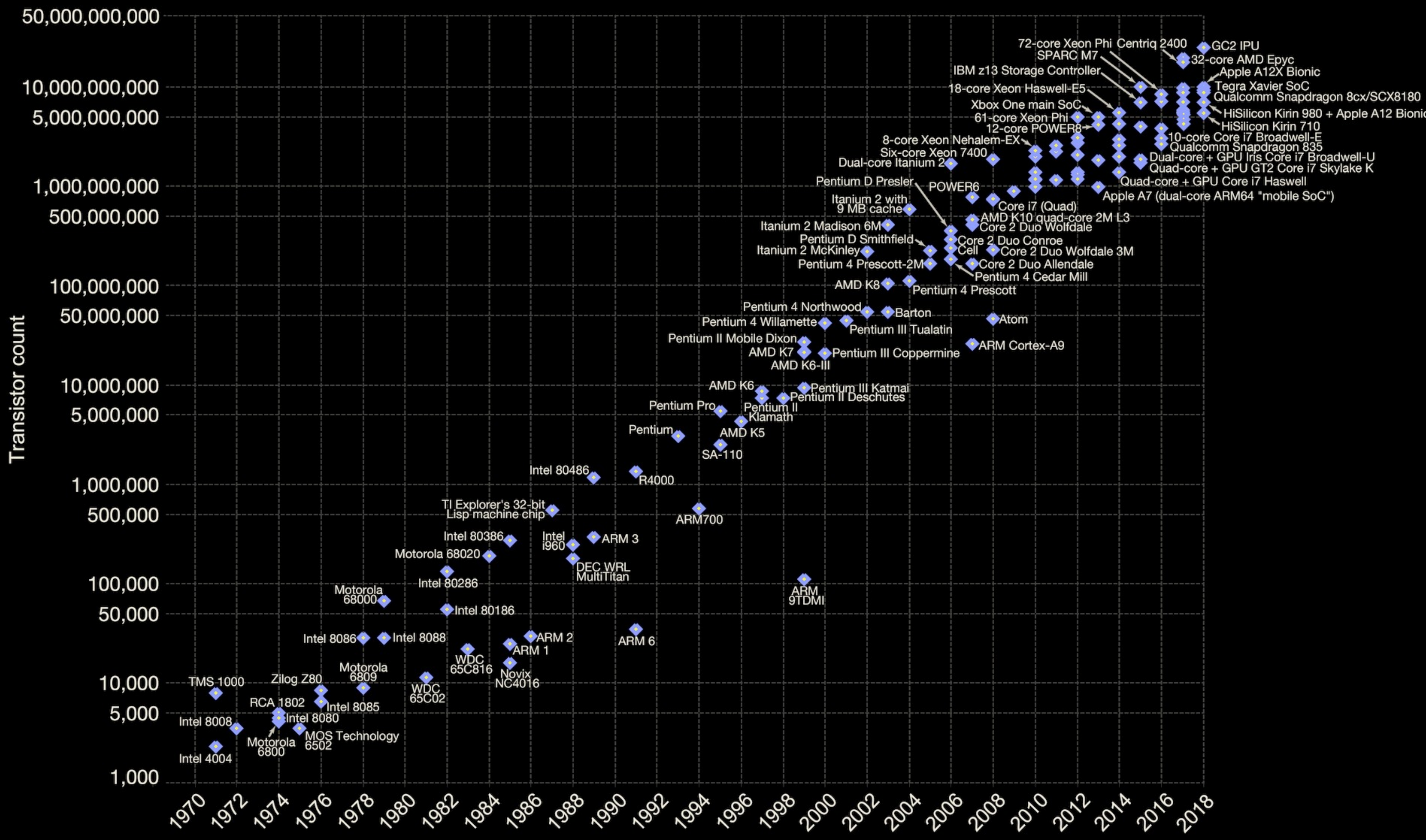




Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

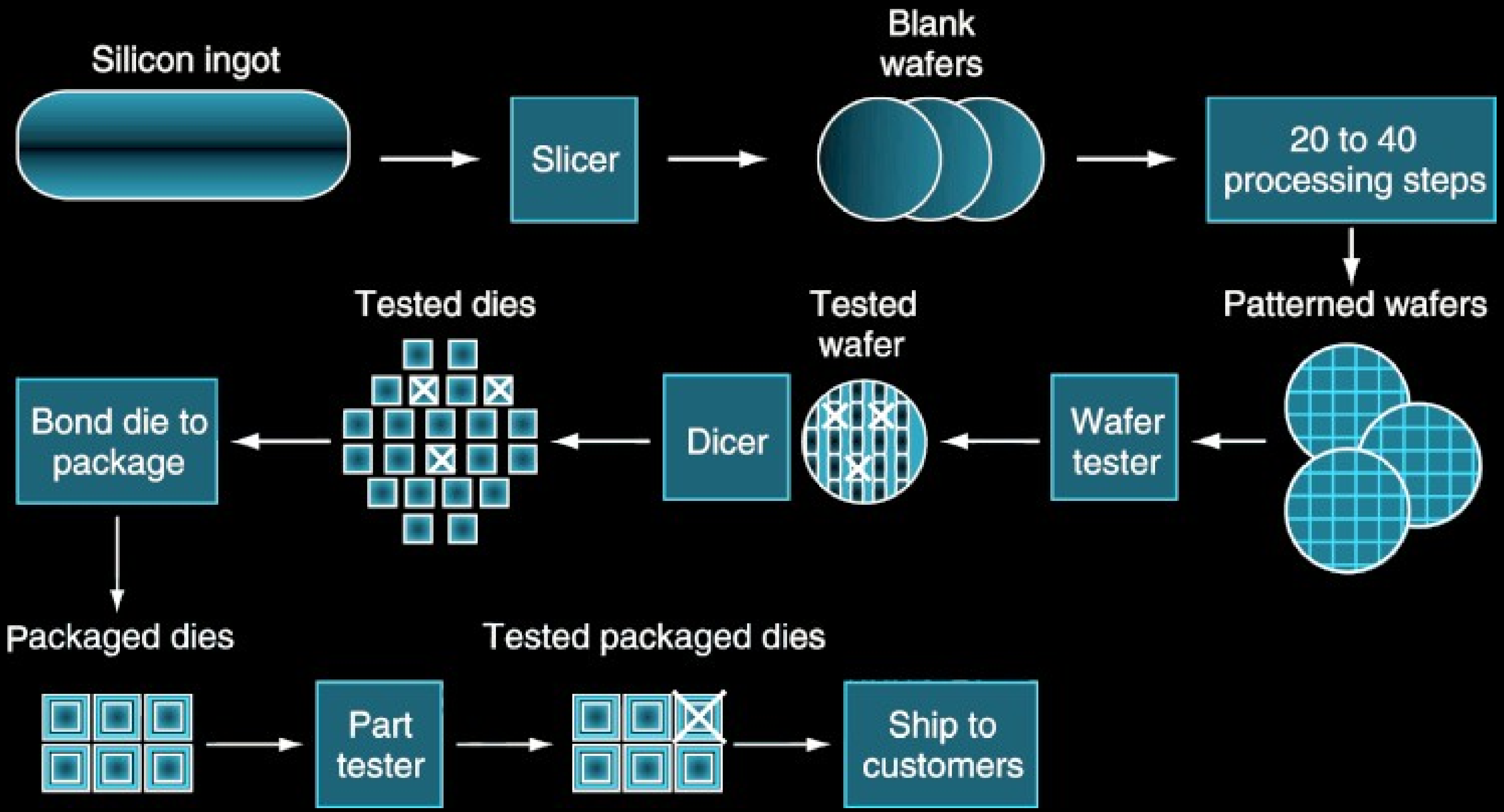
Our World in Data

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



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Chip Fabrication



Chip Fabrication

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area} / \text{Die area}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

- Moore's Law is primarily about density, not speed
- Fab cost ~ cube of the die area

IC Costs: Dies To Chips

$$\text{IC cost} = \frac{\text{Die cost} + \text{Testing cost} + \text{Packaging cost}}{\text{Final test yield}}$$

Packaging Cost: depends on pins, heat dissipation

Chip	*Die cost*	*Package pins*	*Package type*	*Package cost*	*Test & Assembly*	*Total*
386DX	$4	132	QFP	$1	$4	$9
486DX2	$12	168	PGA	$11	$12	$35
PowerPC 601	$53	304	QFP	$3	$21	$77
HP PA 7100	$73	504	PGA	$35	$16	$124
DEC Alpha	$149	431	PGA	$30	$23	$202
SuperSPARC	$272	293	PGA	$20	$34	$326
Pentium	$417	273	PGA	$19	$37	$473

Technology Trends

	Capacity	Speed
Logic	2X in 2 years	2X in 3 years
DRAM	4X in 3 years	1.4X in 10 years
Disk	4X in 3 years	1.4X in 10 years

Different rates mean relationships change;
e.g., memory used to be faster than Add logic,
now it's ~2000X slower!

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)

Conclusion

- LOTS of stuff to know about...
  this course just does the basic stuff around the ISA and its implementation
- New technologies & applications mean new architectures & architectural concepts
- Look at the history references on the WWW:
  not to memorize who, what, when, & where,
  but to ***see trends***...