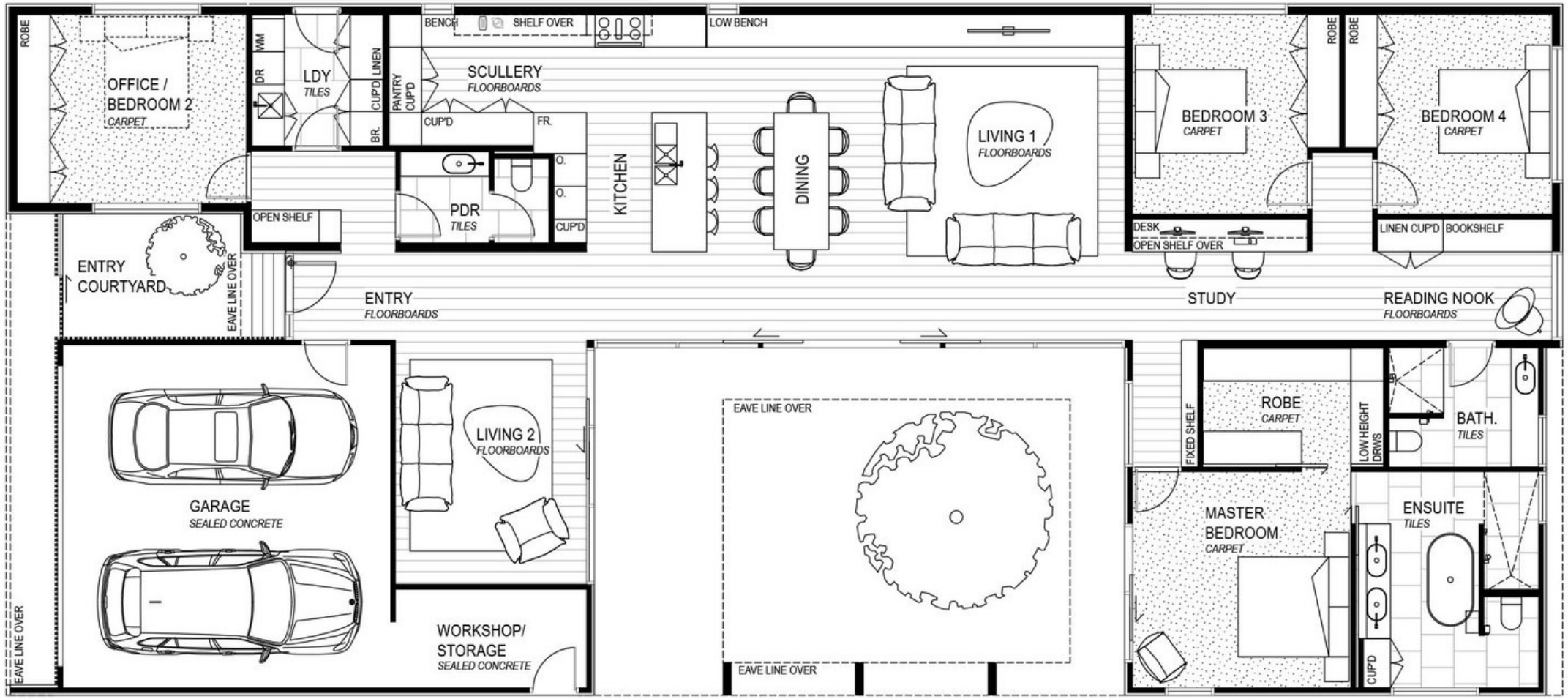
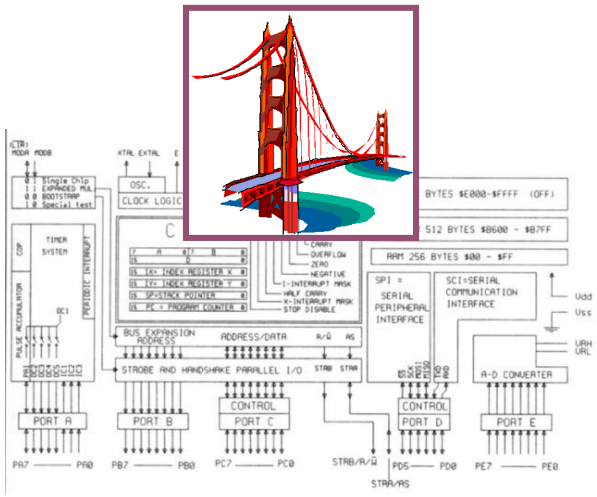
Introduction

ICT 2203 Computer Architecture

What is Computer Architecture?



What's Computer Architecture?



- Architecture (in general) =
  - Design of a functional structure
- Computer Architecture (CA) =
  - Design of the logical structure and functional organization of a computer system.
  - Especially its CPU and associated components
- Computer Architecture does not traditionally include other aspects of computer system design…
  - Enclosures, styling, packaging, applications, power supplies, cooling systems, peripheral devices.
  - But these are all important in designing real-world products!

Hardware

- This is a course about what's inside the box.
- How does the hardware run the software?
- How do hardware design features impact software execution?
- How does the software interface with the hardware?

In this course

C Programming

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?

Computer Hardware

"C" as a model of computation

Programmer's view of 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 a computer system works

Digital logic as a model of computation

Architecture and Organization

- Architecture is the design of the system visible to the assembly level programmer.
  - What instructions
  - How many registers
  - Memory addressing scheme
- Organization is how the architecture is implemented.
  - How much cache memory
  - Microcode or direct hardware
  - Implementation technology

Same Architecture Different Organization

- Almost every program that can run on an original Pentium (or 8086) can run on a Pentium 4.
- All computers in the Intel Pentium series have the same architecture.
- Each version of the Pentium has a different organization or implementation.
- The IBM 360 computer was released in several different models.
- All had the same architecture. A program compiled on one IBM 360 would run on all models.
- The different models had different implementations, speed and price.

Why Study Computer Architecture?

- Enable better systems
  - make computers faster, cheaper, smaller, more reliable
  - By exploiting advances and changes in underlying technology/circuits
- Enable new applications
  - Life-like 3D visualization 20 years ago?
  - Virtual reality?
  - Personal genomics?
- Enable better solutions to problems
  - Software innovation is built into trends and changes in computer architecture
  - > 50% performance improvement per year has enabled this innovation
- Understand why computers work the way they do

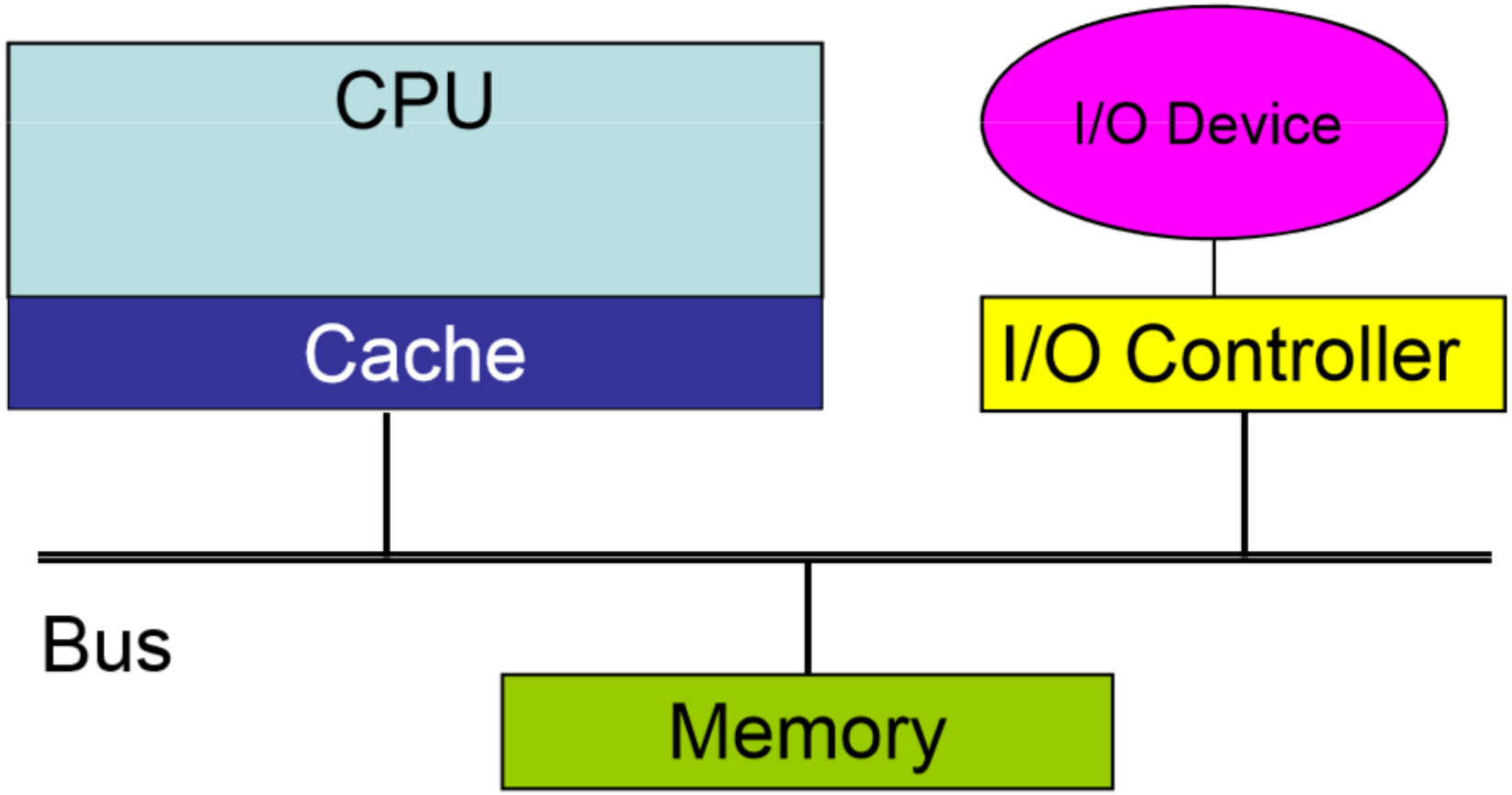
Von Neumann Computers

- CPU
- Memory (disk drives, DRAM, SRAM, CD)
- Input (mouse, keyboard)
- Output (display, printer)
- Network
- Software

Levels of Abstraction

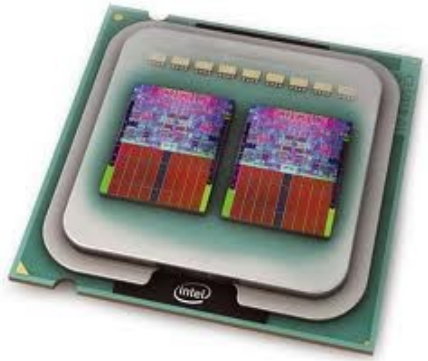
- Software:
  - Application
  - Operating system
  - Firmware
- Instruction set architecture:
  - Data type and structures: encodings and machine representation
  - Instruction set
  - Instruction formats
  - Addressing modes and accessing data and instructions
- Hardware:
  - Instruction set processing
  - I/O System
  - Digital design
  - Circuit design
  - Layout

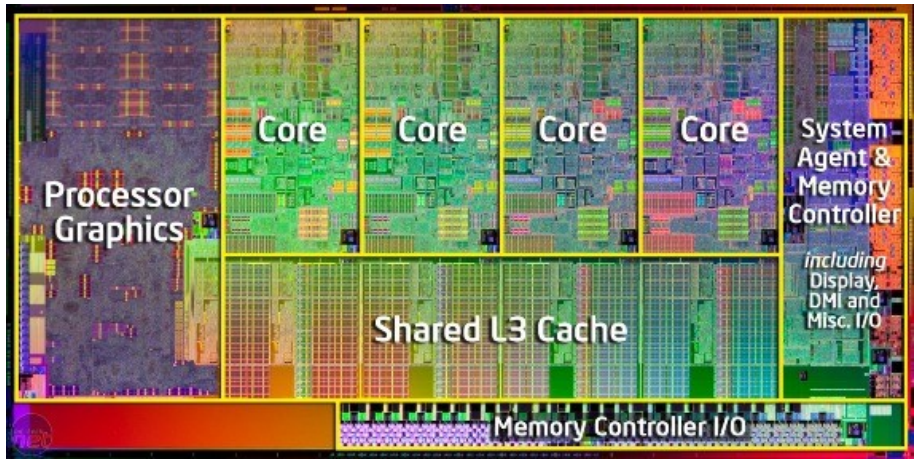
Basic Computer Components



Central Processing Unit

- Contains the control logic that initiates most activities in the computer.
- The Arithmetic Logic Units perform the math and logic calculations.
- Registers contain temporary data values.
- Program Counter contains the address of the next instruction to execute.





Registers

- The CPU has registers to temporarily hold data being acted upon.
- Different architectures have different number of registers.
- Some registers are available for the user programs to use directly.
- Some registers are used indirectly (such as the program counter).
- Some registers are used only by the operating system (i.e. program status reg)

Bus

- The bus is a set of parallel wires that connect the CPU, memory and I/O controllers.
- It has logic (the chipset) to determine who can use the bus at any given instant.
- The width of the bus determines the maximum memory configuration

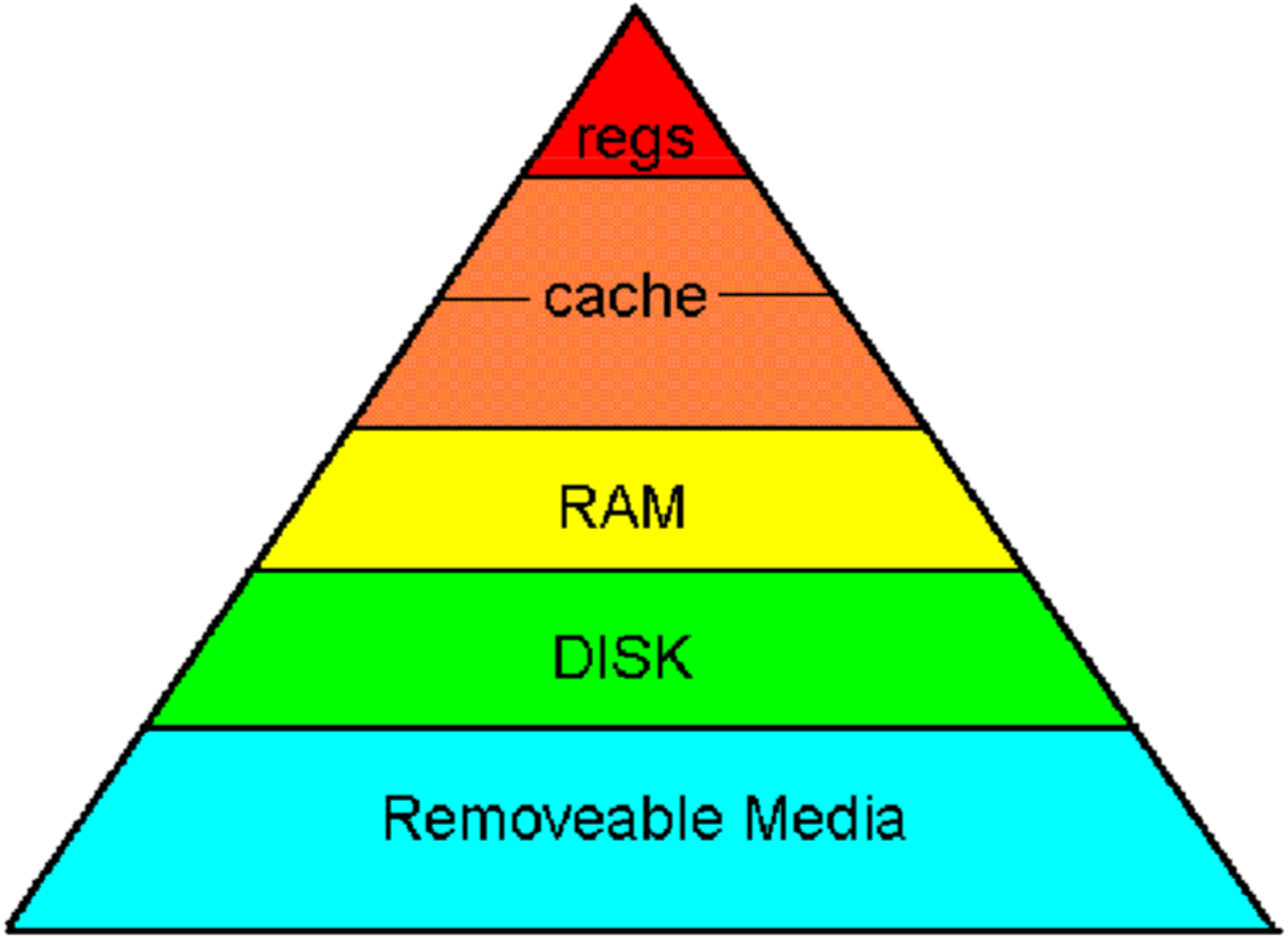
I/O Controllers

- Direct the flow of data to and from I/O devices.
- CPU sends a request to the I/O controller to initiate I/O.
- I/O controllers run independently and in parallel with the CPU
- I/O controllers may interrupt the CPU upon completion of request or error.

Memory

- The internal memory is Random Access Memory (RAM).
- Both data and program instructions are kept in RAM.
- Instructions must be in RAM to be executed.

Memory Hierarchy



Instruction Cycle

- Fetch the instruction from the memory address in the Program Counter register
- Increment the Program Counter
- Decode the type of instruction
- Fetch the operands
- Execute the instruction
- Store the results

Simple Model of Execution

- Instruction sequence is determined by a simple conceptual control point.
- Each instruction is completed before the next instruction starts.
- One instruction is executed at a time.

Layers

- You can consider computer operation at many different levels:
  - Applications
  - Middleware
  - High level languages
  - Machine Language
  - Microcode
  - Logic circuits
  - Gates
  - Transistors
  - Silicon structures

History of Computers

Historical Progression

- People have worked to build “thinking” devices for a long time.
- Improvements usually build on earlier work
- Before the 1940’s “Computer” was a job title, not a machine.

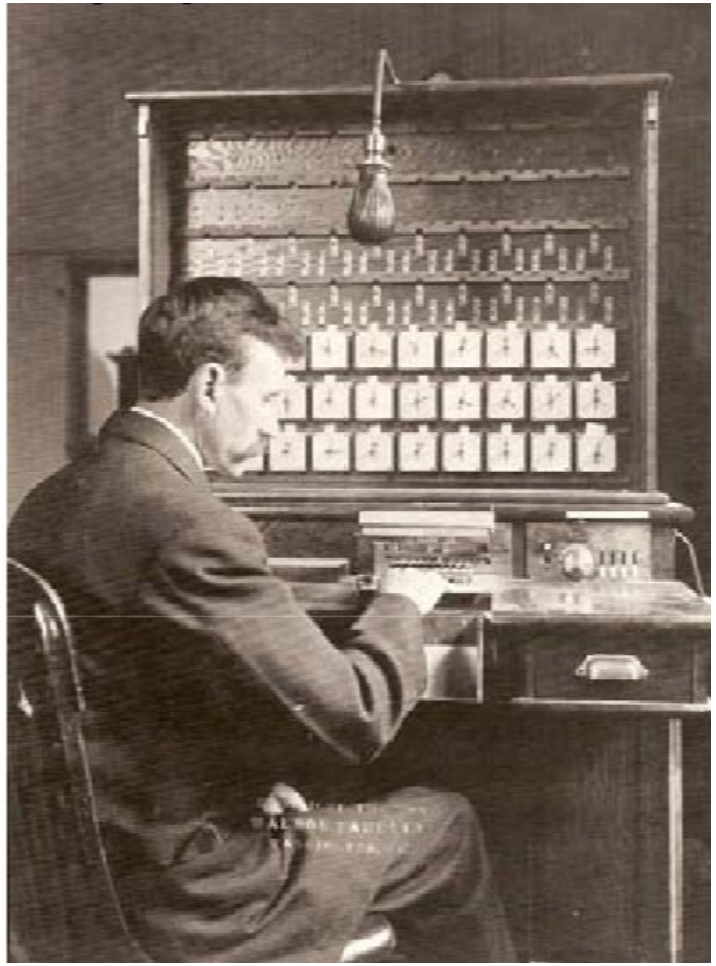
Date	Who	What
~1000 BC	?	Abacus
1621	William Oughtred	Slide Rule
1642	Blaise Pascal	Adding machine

Punch Cards

- In 1804-05 Joseph-Marie Jacquard invented a loom that used punch cards to specify the pattern pattern.

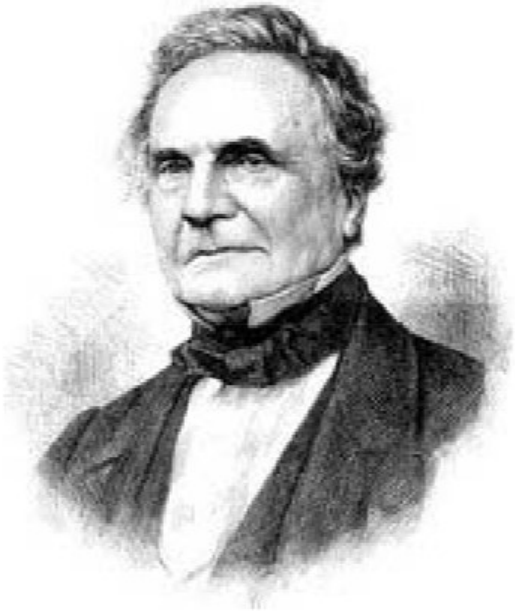
Tabulating Equipment

- In 1882 Herman Hollerith created a punch card tabulating machine.
- It was used to calculate the1890 census.
- Punched cards were used through the late 1970s.



Charles Babbage

- Charles Babbage built a mechanical computer starting in 1822.
- He never completed the machine.



Ada Lovelace

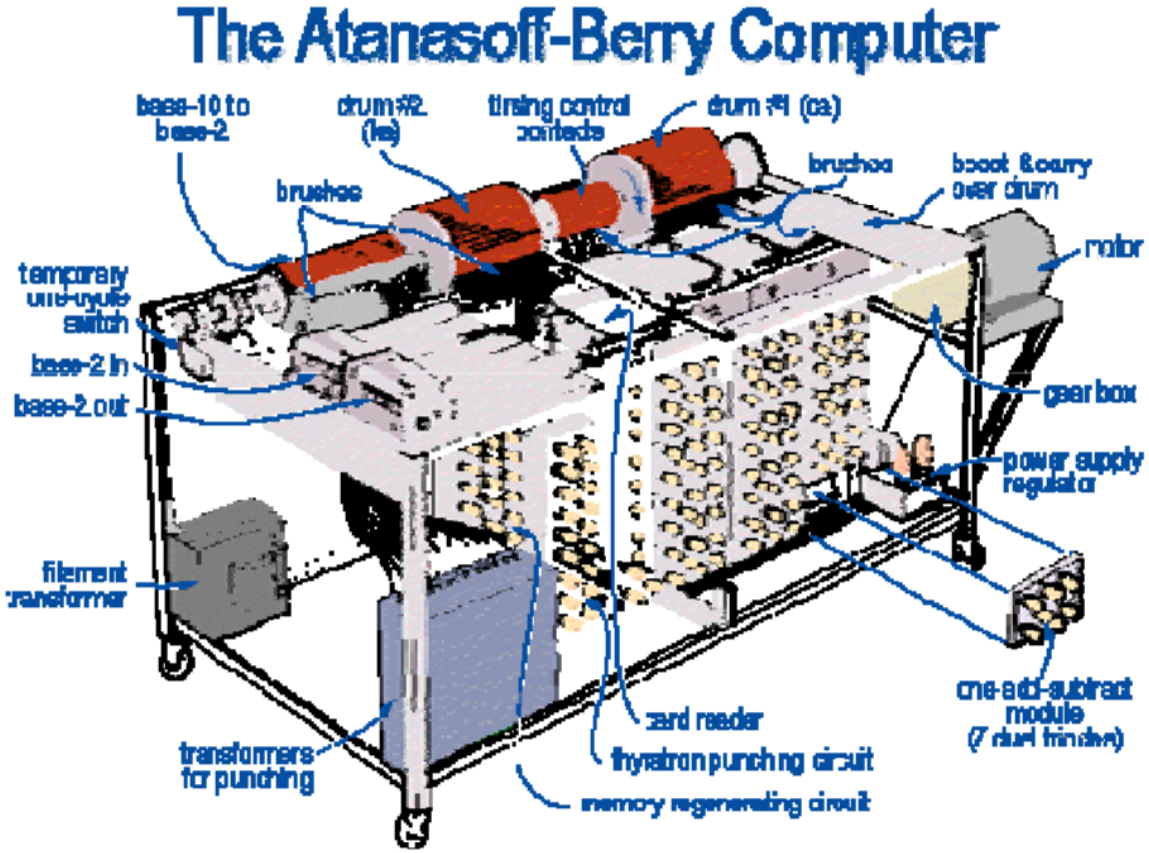
- Augusta Ada, Countess of Lovelace, was the daughter of Lord Byron and friend of Charles Babbage.
- She is considered the first computer programmer.

Alan Turing

- In 1936 Alan Turing invented the theoretical Turing Machine.
- With Alonzo Church developed the Turing-Church thesis.
- “Every function which would naturally be regarded as computable, can be computed by a Turing machine”
- He broke the code of the German Enigma machine in WWII.

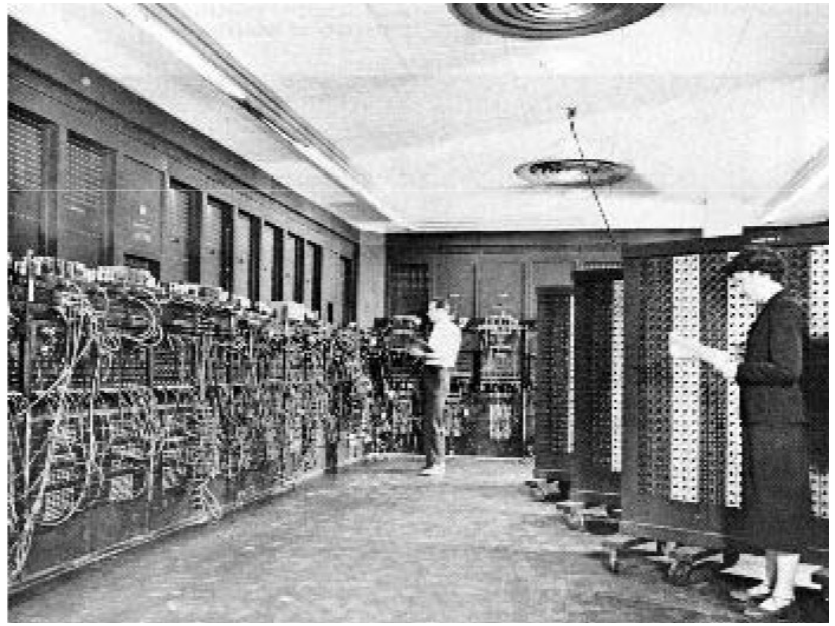
ABC machine

- John Atanasoff and Clifford Berry built the Atanasoff-Berry computer (ABC) in 1939.



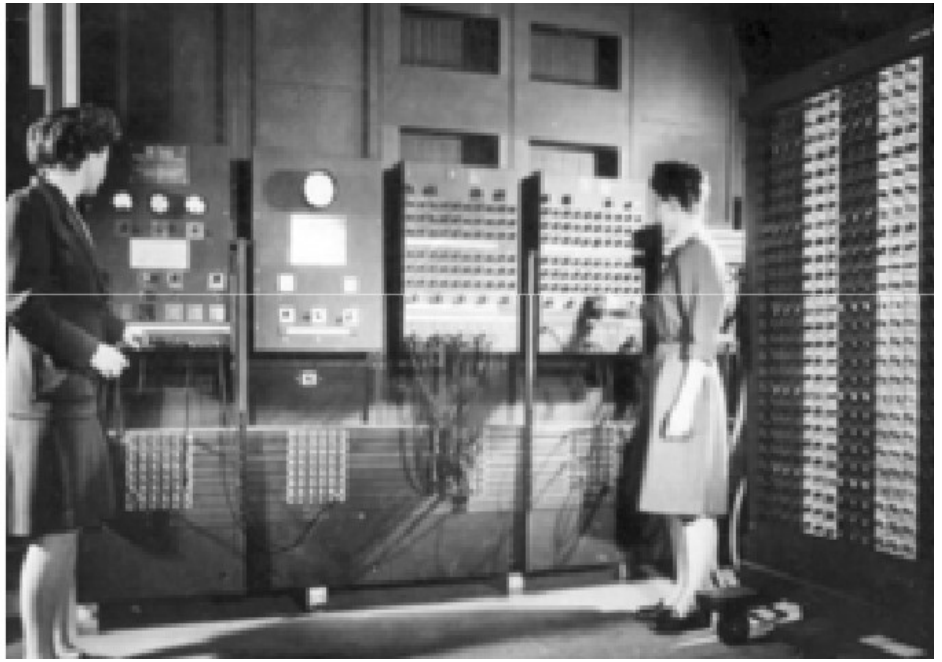
ENIAC

- Electronic Numerical Integrator and Computer
- John Eckert and J. Presper Mauchly
- University of Pennsylvania
- Trajectory tables for weapons
- Started 1943
- Finished1946
- Finished 1946
  - Too late for war effort
- Used until 1955



ENIAC

- Decimal (not binary)
- 20 accumulators of 10 digits
- Programmed manually by switches
- 18,000 vacuum tubes
- 30 tons
- 15 ,000 square feet
- 140 kW power
- 5,000 additions/sec

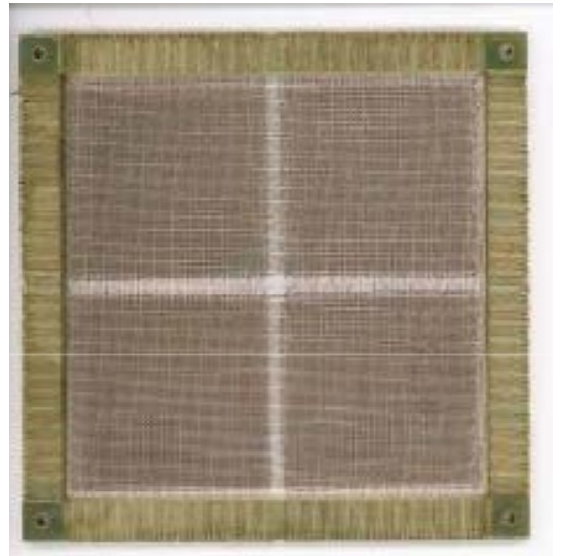


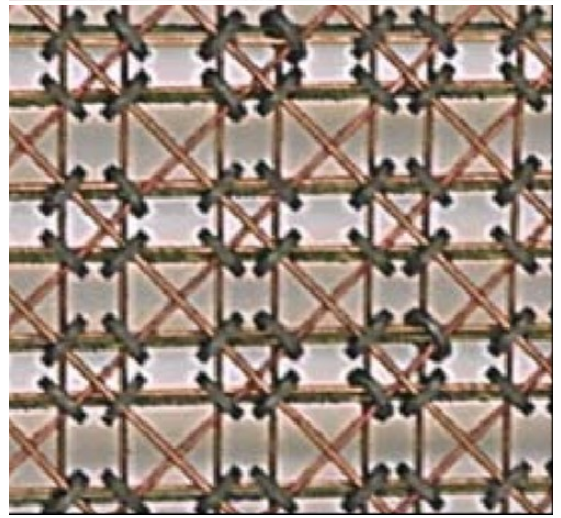
von Neumann Architecture

- Stored Program concept
- Main memory storing programs and data
- ALU operating on binary data
- Control unit interpreting instructions from memory and executing
- Input and output equipment operated by control unit
- Completed 1952

Core Memory

- Invented by An Wang and Way-Dong Woo in 1949
- A bit is stored by magnetizing a ring of iron.
- Cycle times of about 6µs
- Non-volatile storage





Transistors

- Replaced vacuum tubes
- Smaller
- Cheaper
- Less heat dissipation
- Solid State device
- Made from silicon (sand)
- Invented 1947 at Bell Labs
- William Shockley et al.

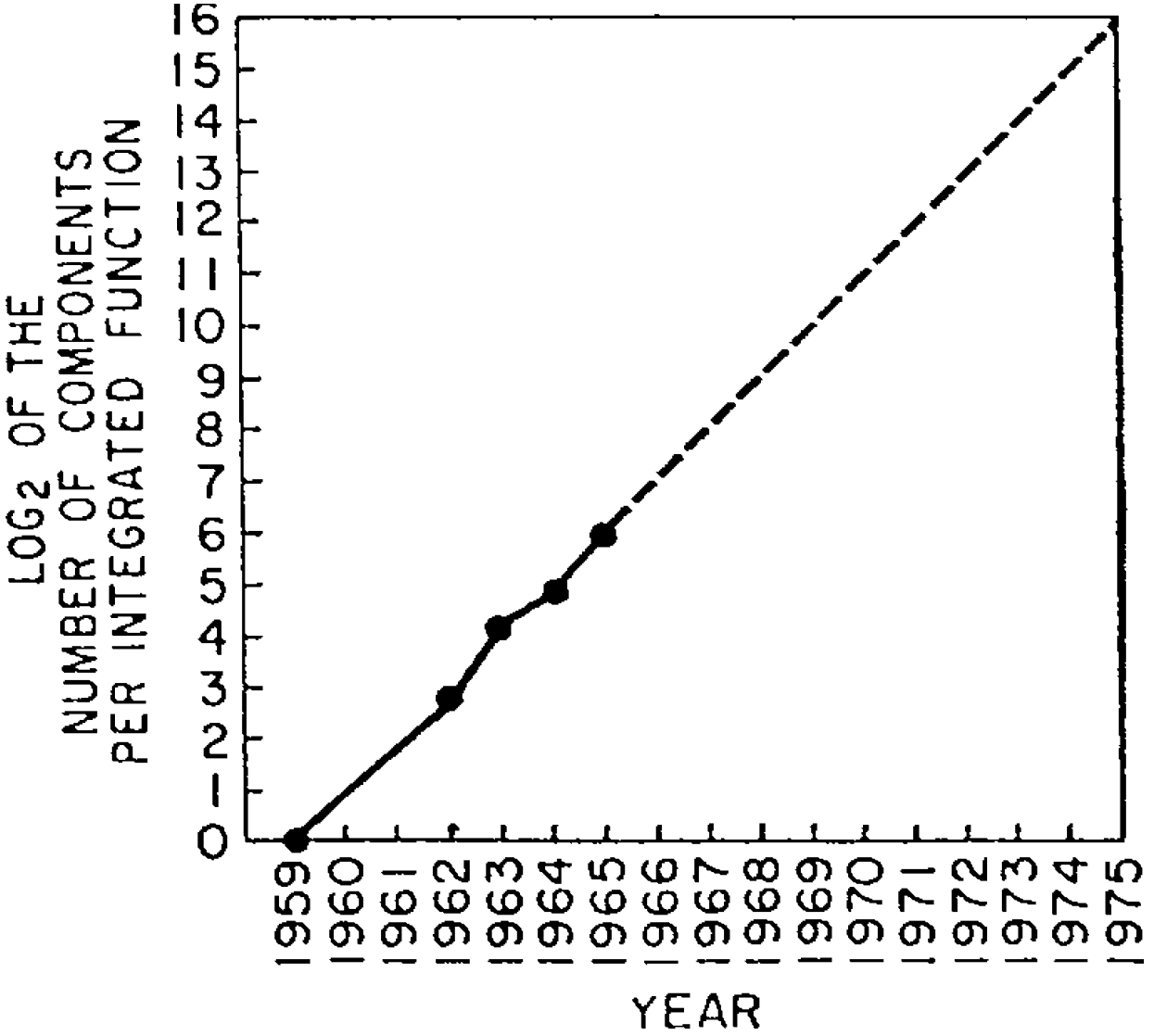
Semiconductor Memory

- Created in1970 at Fairchild corporation
- Size of a single core
  - i.e. 1 bit of magnetic core storage
- Non-destructive read
- Much faster than core
- Capacity approximately doubles each year

Moore's Law

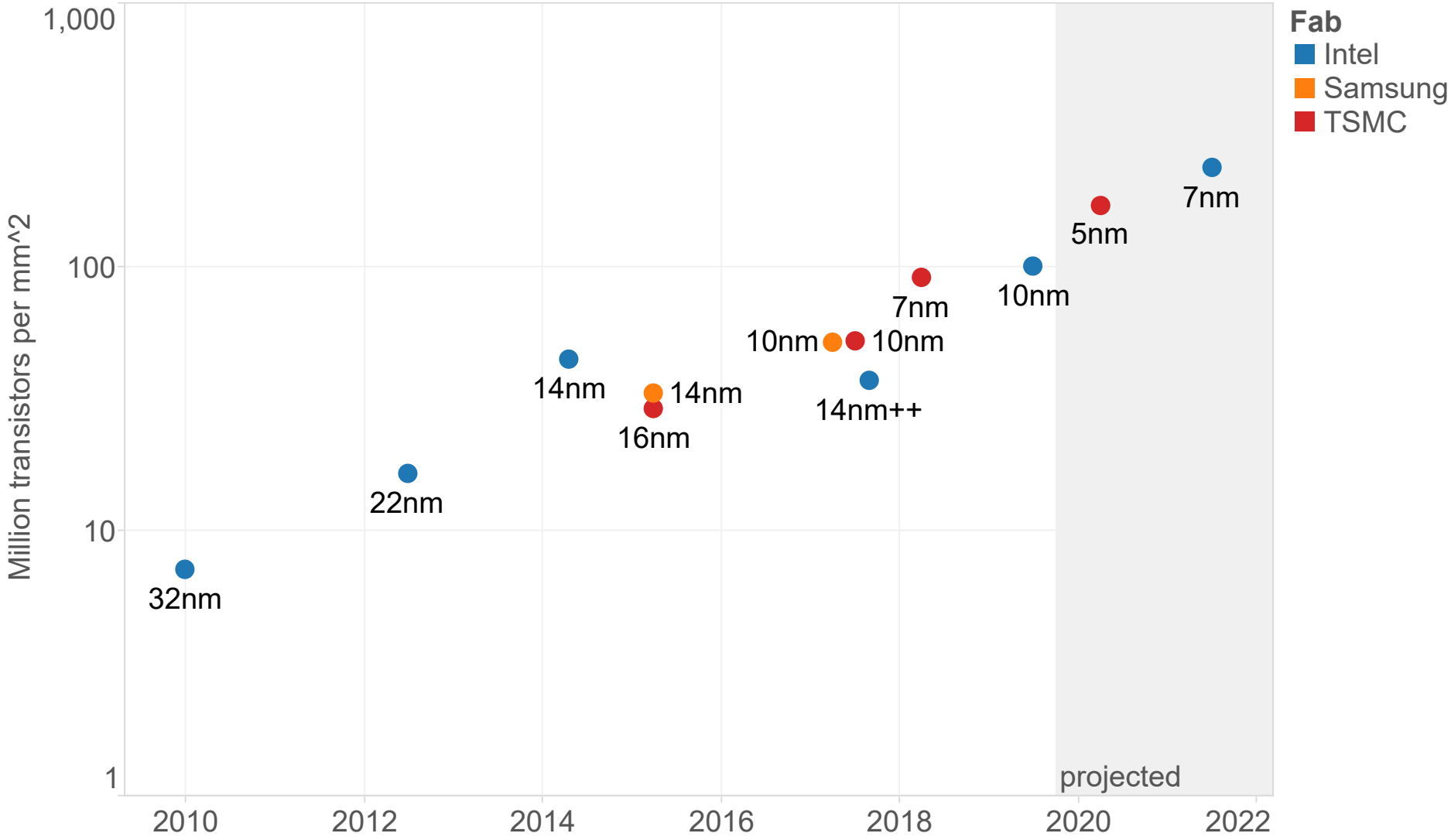
- Increased density of components on chip
- Gordon Moore –co-founder of Intel
- Number of transistors on a chip will double every year
- Since 1970's development has slowed a little
- **Number of transistors on a chip doubles every 18 months**
- Cost of a chip has remained almost unchanged
- Higher packing density means shorter electrical paths, giving higher performance
- Reduced power and cooling requirements
- Fewer interconnections increases reliability

Moore's Law - 1965

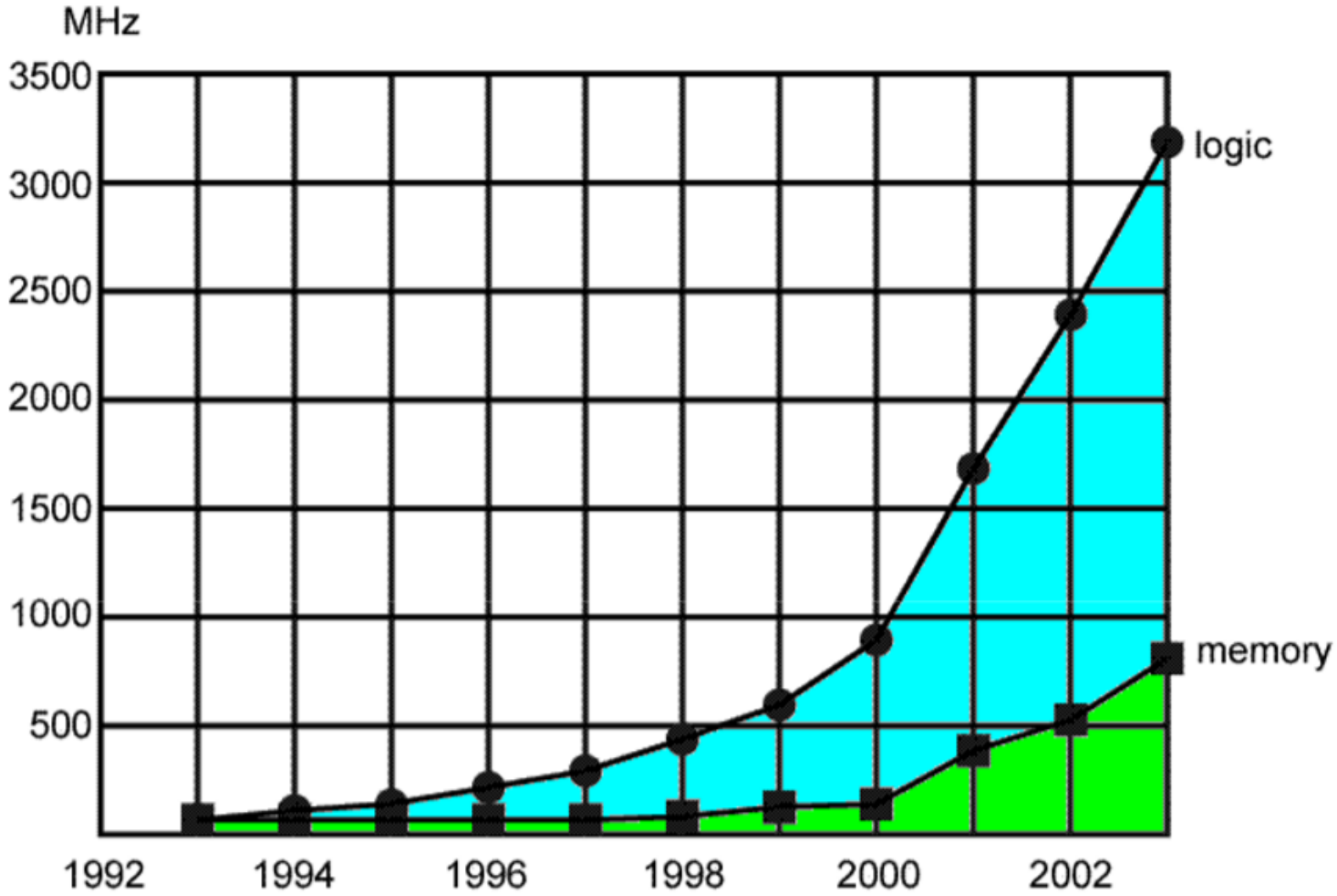


data c/o WikiChip

Moore's Law today



Logic and Memory Performance Gap

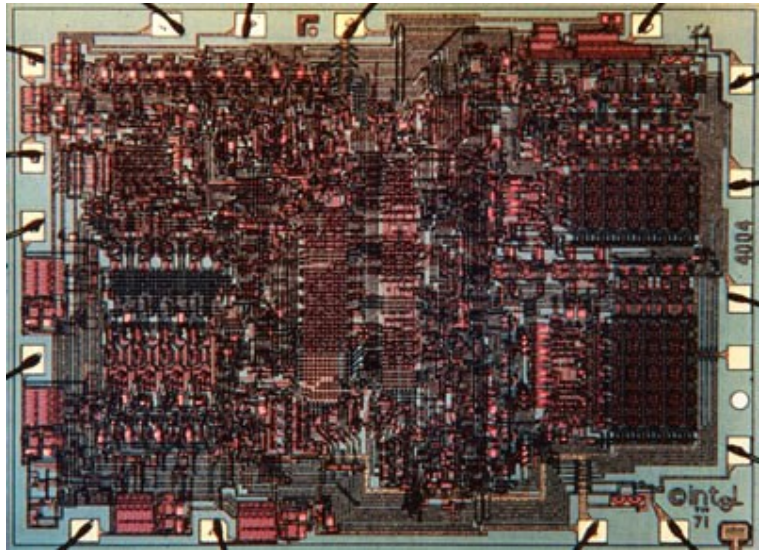


Revolution I: The Microprocessor

- Microprocessor revolution
  - One significant technology threshold was crossed in 1970s
  - Enough transistors (~25K) to put a 16-bit processor on one chip
  - Huge performance advantages: fewer slow chip-crossings
  - Even bigger cost advantages: one “stamped-out” component
- Microprocessors have allowed new market segments
  - Desktops, CD/DVD players, laptops, game consoles, set-top boxes, mobile phones, digital camera, mp3 players, GPS, automotive
- And replaced incumbents in existing segments
  - Microprocessor-based system replaced supercomputers, “mainframes”, “minicomputers”, “desktops”, etc.

First Microprocessor

- Intel 4004 (1971)
  - Application: calculators
  - Technology: 10,000 nm
  - 2300 transistors
  - 13 mm2
  - 108 KHz
  - 12 Volts
  - 4-bit data
  - Single-cycle datapath



Revolution II: Implicit Parallelism

- Then to extract implicit instruction-level parallelism
  - Hardware provides parallel resources, figures out how to use them
  - Software is oblivious
- Initially using pipelining …
  - Which also enabled increased clock frequency
- … Caches
  - Which became necessary as processor clock frequency increased
- …and integrated floating-point.
- Then deeper pipelines and branch speculation
- Then multiple instructions per cycle (superscalar)
- Then dynamic scheduling (out-of-order execution)

Pinnacle of Single-Core Microprocessors

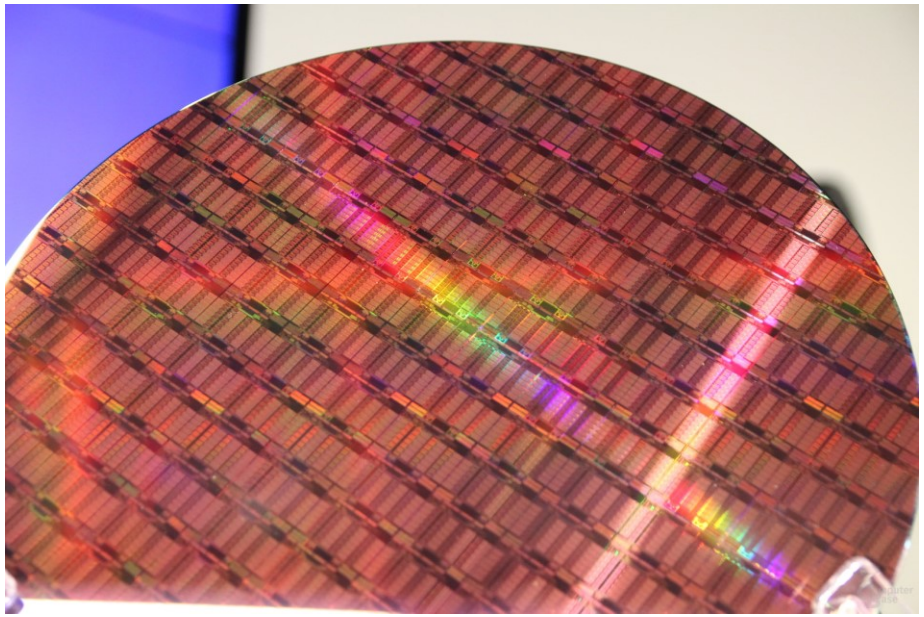
- Intel Pentium4 (2003)
  - Application: desktop/server
  - Technology: 90nm (1/100x)
  - 55M transistors (20,000x)
  - 101 mm2 (10x)
  - 3.4 GHz (10,000x)
  - 1.2 Volts (1/10x)
  - 32/64-bit data (16x)
  - 22-stage pipelined datapath
  - 3 instructions per cycle (superscalar)
  - Two levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading

Revolution III: Explicit Parallelism

- Then to support explicit data & thread level parallelism
  - Hardware provides parallel resources, software specifies usage
  - Why? diminishing returns on instruction-level-parallelism
- First using (subword) vector instructions…, Intel's SSE
  - One instruction does four parallel multiplies
- … and general support for multi-threaded programs
  - Coherent caches, hardware synchronization primitives
- Then using support for multiple concurrent threads on chip
  - First with single-core multi-threading, now with multi-core
- Graphics processing units (GPUs) are highly parallel
  - Converging with general-purpose processors (CPUs)?

Modern Multicore Processor

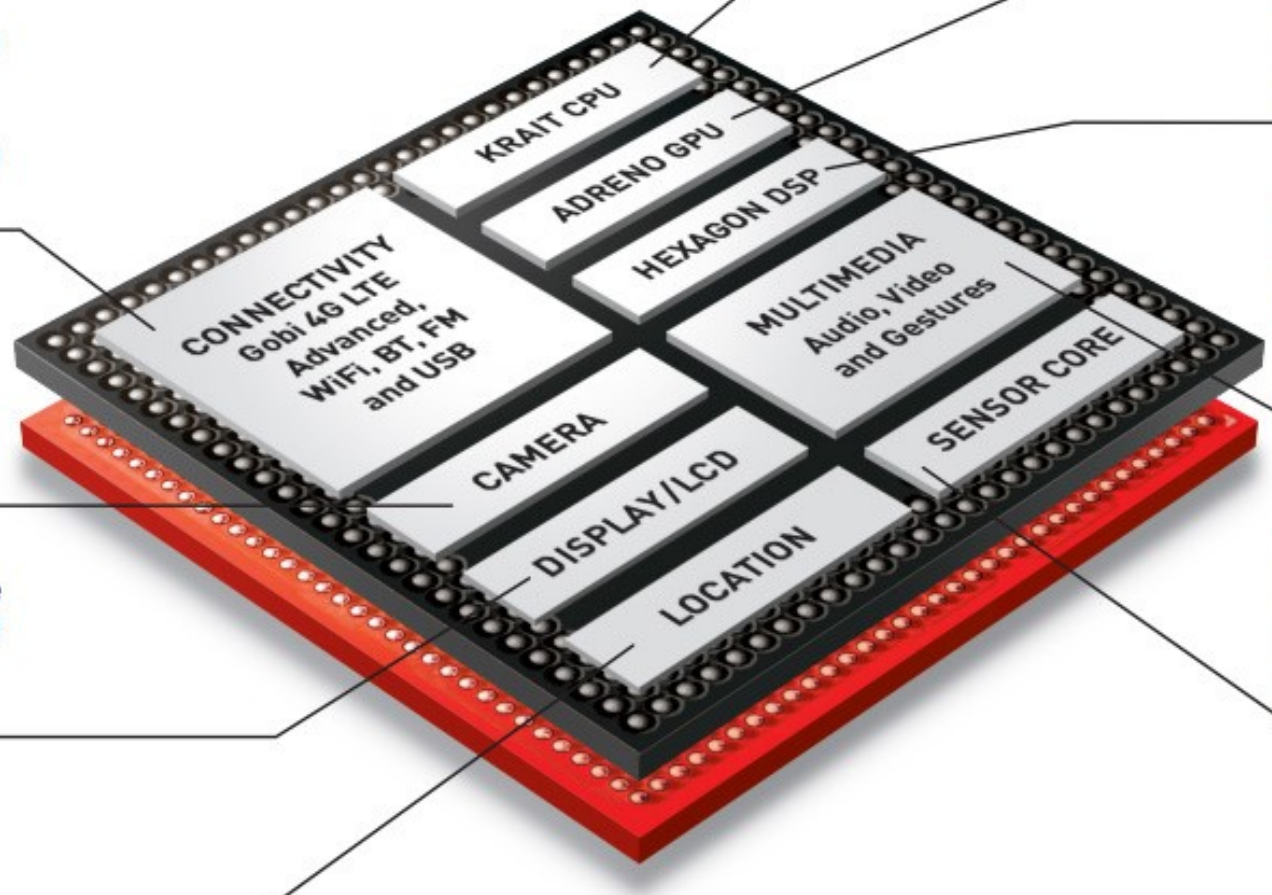
- Intel Xeon E5-2699 V4 (2016)
  - Application: server
  - Technology: 14nm (16% of P4)
  - 7.2B transistors (130x)
  - 456 mm2 (4.5x)
  - 2.4 to 3.6 Ghz (~1x)
  - 1.8 Volts (~1x)
  - 256-bit data (2x)
  - 14-stage pipelined datapath (0.5x)
  - 4 instructions per cycle (1x)
  - Three levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading
  - 22-core multicore (22x)



Revolution IV: Heterogeneous Processing

- Combining multiple kinds of compute engines in one die
  - not just homogenous collection of cores
  - System-on-Chip (SoC) is one common example in mobile space
- Lots of stuff on the chip beyond just CPUs
  - Graphics Processing Units (GPUs)
    - throughput-oriented specialized multicore processors
    - good for gaming, machine learning, computer vision, …
  - Special-purpose logic
    - media codecs, radios, encryption, compression, machine learning
- Excellent energy efficiency and performance
  - extremely complicated to program!

SNAPDRAGON 805 PROCESSOR



Faster performance and more multitasking with Krait 450 CPU at up to 2.7 GHz

Console quality gaming with new generation Adreno 420 GPU

More power-efficient apps and system processing with the Hexagon™ QDSP6

Capture and play back Ultra HD video and enjoy 7.1 surround sound on the go or at home with advanced video and audio engines

Get more use and greater accuracy from sensor-intensive apps with the dedicated Snapdragon Sensor Engine

Stay connected and stream large files fast with industry leading connectivity, including the world's most advanced 4G LTE and VIVE™ 2-stream 802.11ac Wi-Fi

Capture sharper photos, even in low light, with the mobile industry's first dual ISP

Enjoy Ultra HD resolution content on Ultra HD-capable mobile devices and Ultra HD TVs with the Snapdragon Display Processor

Find your way outdoors and indoors with IZat GNSS with support for GPS, Glonass and BeiDou constellations

Cerebras: Wafer-Scale Engine

- Giant 8.5” square chip!
- Full of deep learning accelerators
- 18GB on-chip memory
- 9 PB/sec on-chip memory bandwidth
- TSMC 16nm transistors

Technology Disruptions

- Classic examples:
  - The transistor
  - Microprocessor
- More recent examples:
  - Flash-based solid-state storage
  - Accelerators
- Nascent disruptive technologies:
  - non-volatile memory (“disks” as fast as DRAM)
  - Chip stacking (also called 3D die stacking)
- Disruptive “end-of-scaling”:
  - “If something can’t go on forever, it must stop eventually”
  - Transistor speed/energy efficiency not improving like before

Themes

- Parallelism
  - How do we enhance system performance by doing multiple things at once.
  - This happens at multiple levels: Instruction Level Parallelism, multicore, GPU
- How do we accelerate access to data
  - Exploiting locality of reference, building storage hierarchies and caches
  - Try to provide the illusion of a single large, fast memory

Next:

CPU Functions

Processor modes

Architectural performance