CS 3410: Computer System Organization and Programming

Anne Bracy

Computer Science

Cornell University

The slides are the product of many rounds of teaching CS 3410 by Professors Weatherspoon, Bala, Bracy, and Sirer.

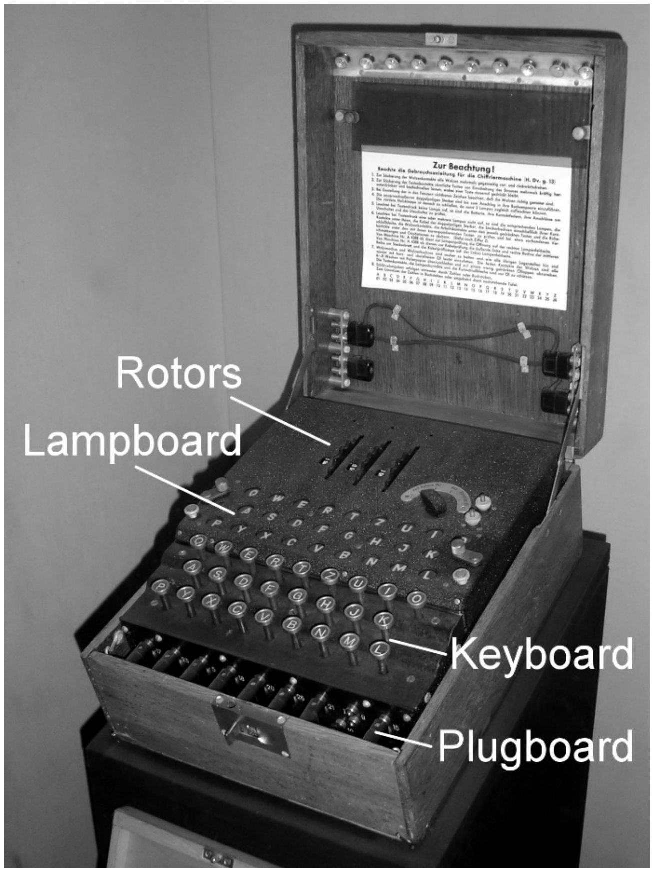
“Sometimes it is the people
that no one imagines anything of
who do the things
that no one can imagine”

--quote from the movie The Imitation Game

“Can machines think?”

-- Alan Turing, 1950

Computing Machinery and Intelligence



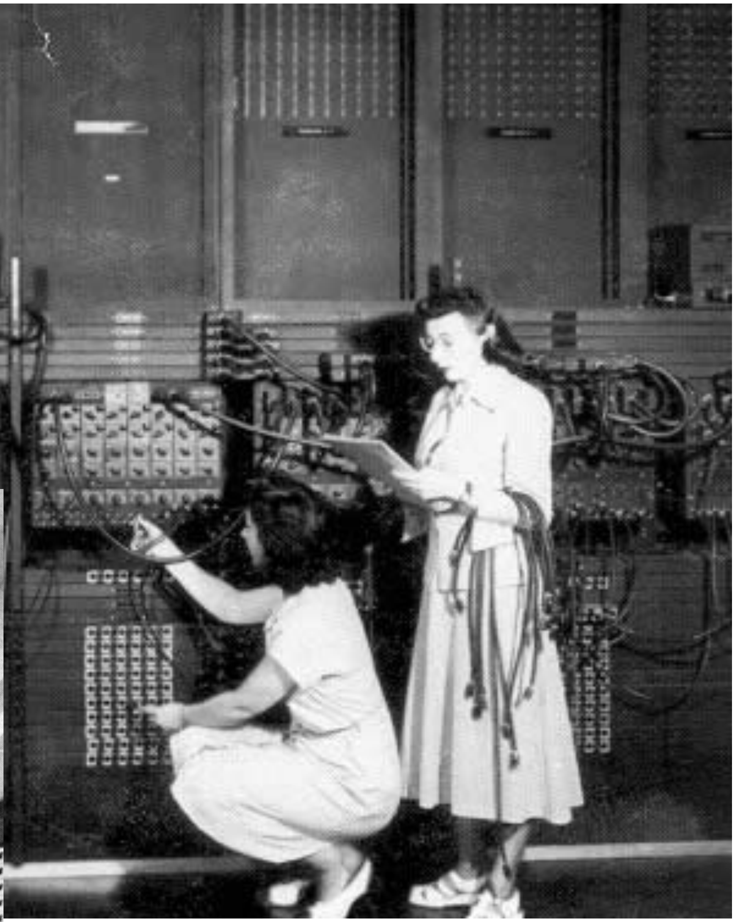
Enigma machine

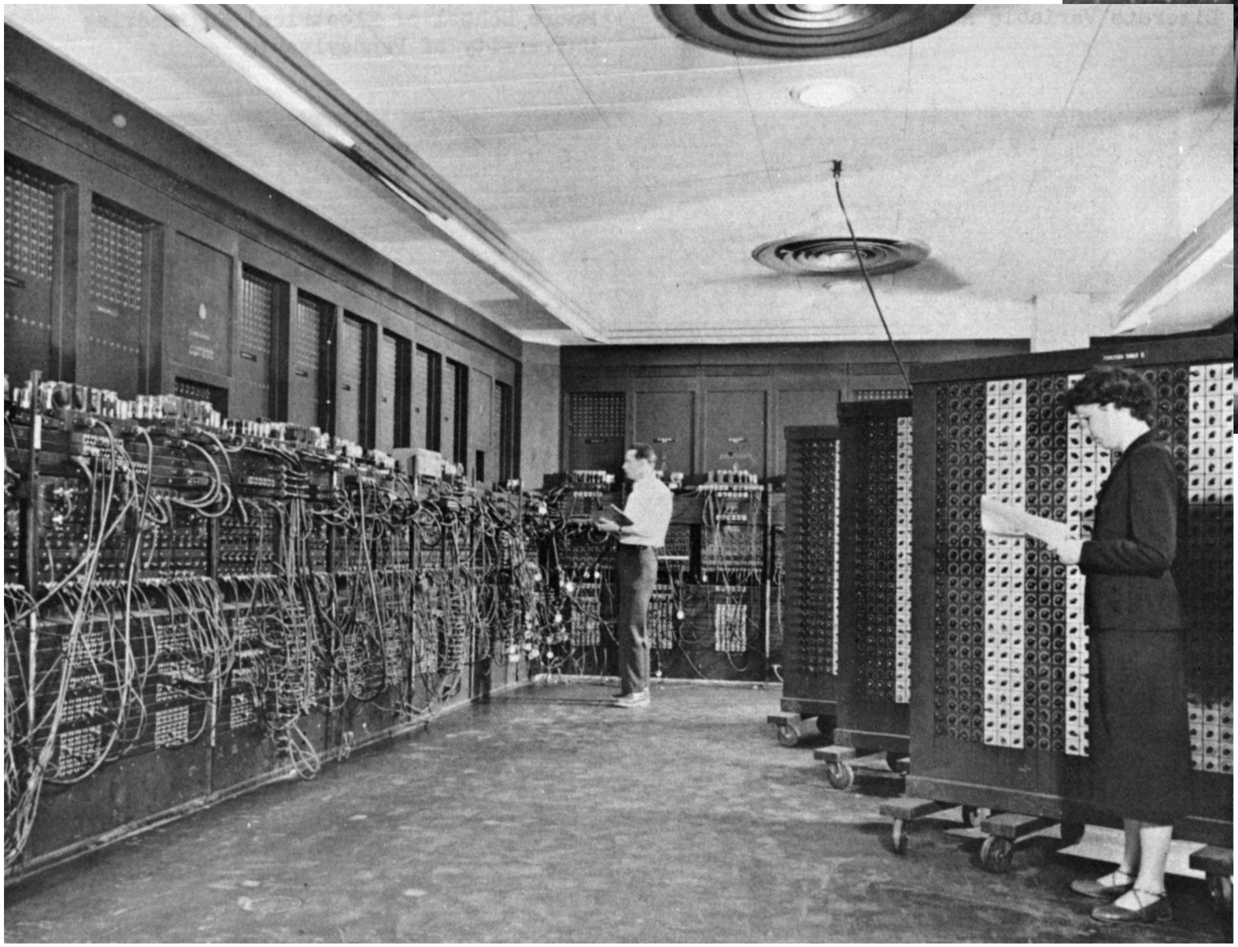
Used by the Germans during World War II to encrypt and exchange secret messages

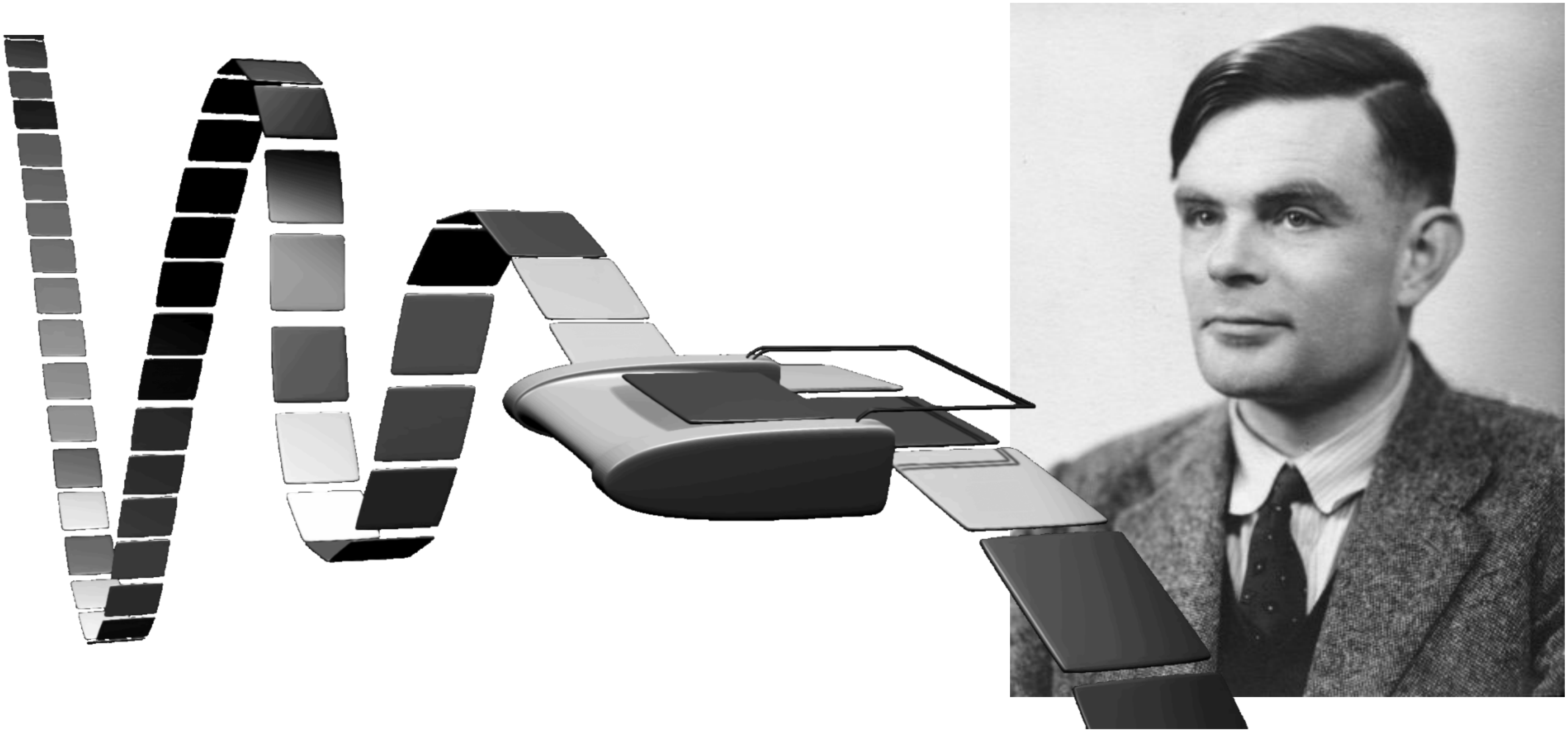
The Bombe

used by the Allies to break the German Enigma machine during World War II

ENIAC







Turing Machine
1936

Alan Turing

= abstract model for CPU that can simulate any algorithm

Who are you?

Demographics

Introduce yourself to the people next to you

“Sometimes it is the people that no one imagines anything of who do the things that no one can imagine.”

Turing Award Winners?

Course Objective

Understand the HW / SW interface software

- How a processor works
- How a computer is organized

Establish a foundation for building applications

- How to write a good program
  - Good = correct, fast, and secure
- How to understand where the world is going

Understand technology (past, present, future)

What is this?

```
#include <stdio.h>

int main() {
  printf("Hello world!\n");
  return 0;
}
```

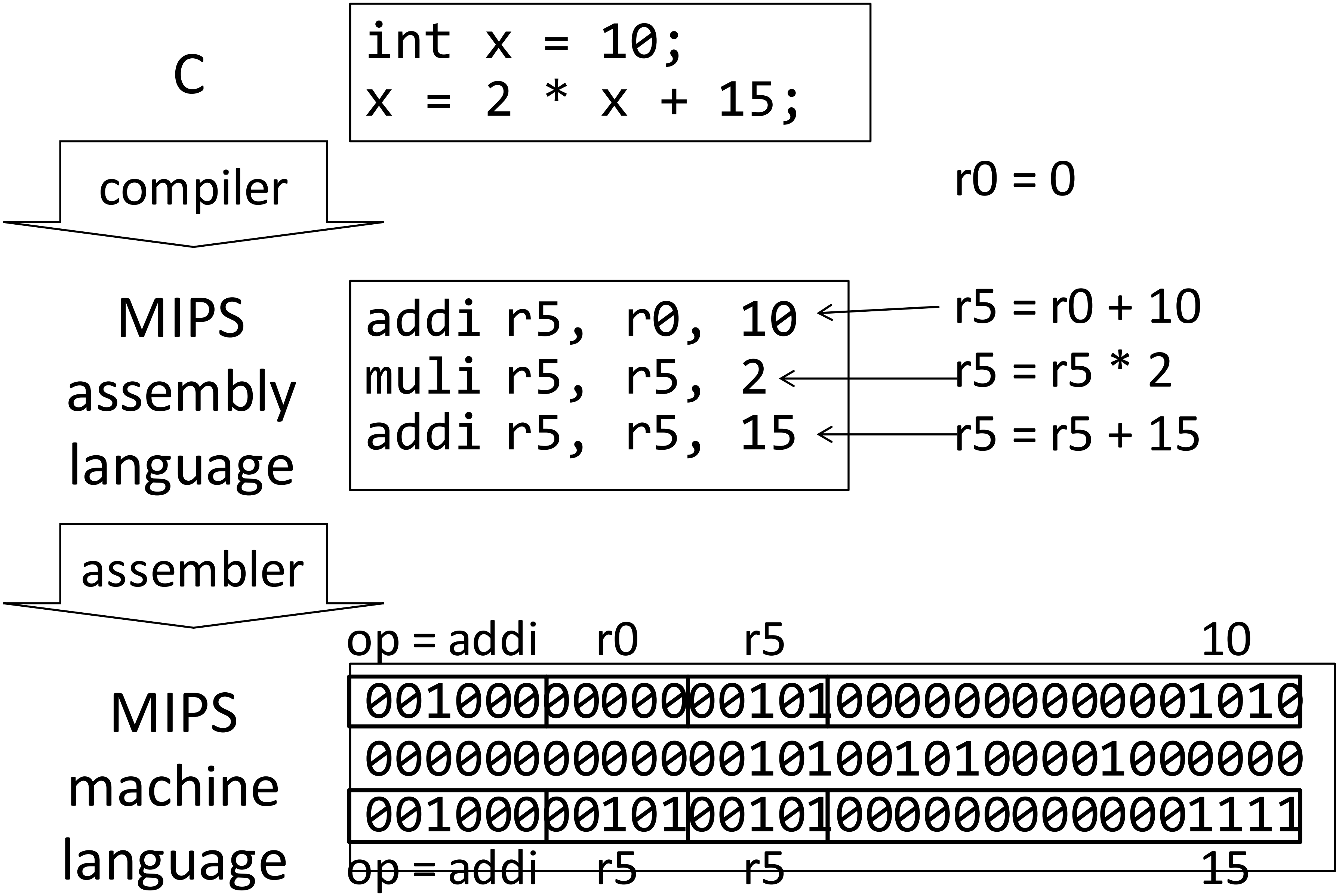
How does it work?

I’m glad you asked…

15 weeks later and you’ll know!

“I know Kung Fu.”

Compilers & Assemblers



C

```
int x = 10;
x = 2 * x + 15;
```

compiler

r0 = 0

MIPS assembly language

```
addi r5, r0, 10
muli r5, r5, 2
addi r5, r5, 15
```

r5 = r0 + 10

r5 = r5 * 2

r5 = r5 + 15

assembler

MIPS machine language

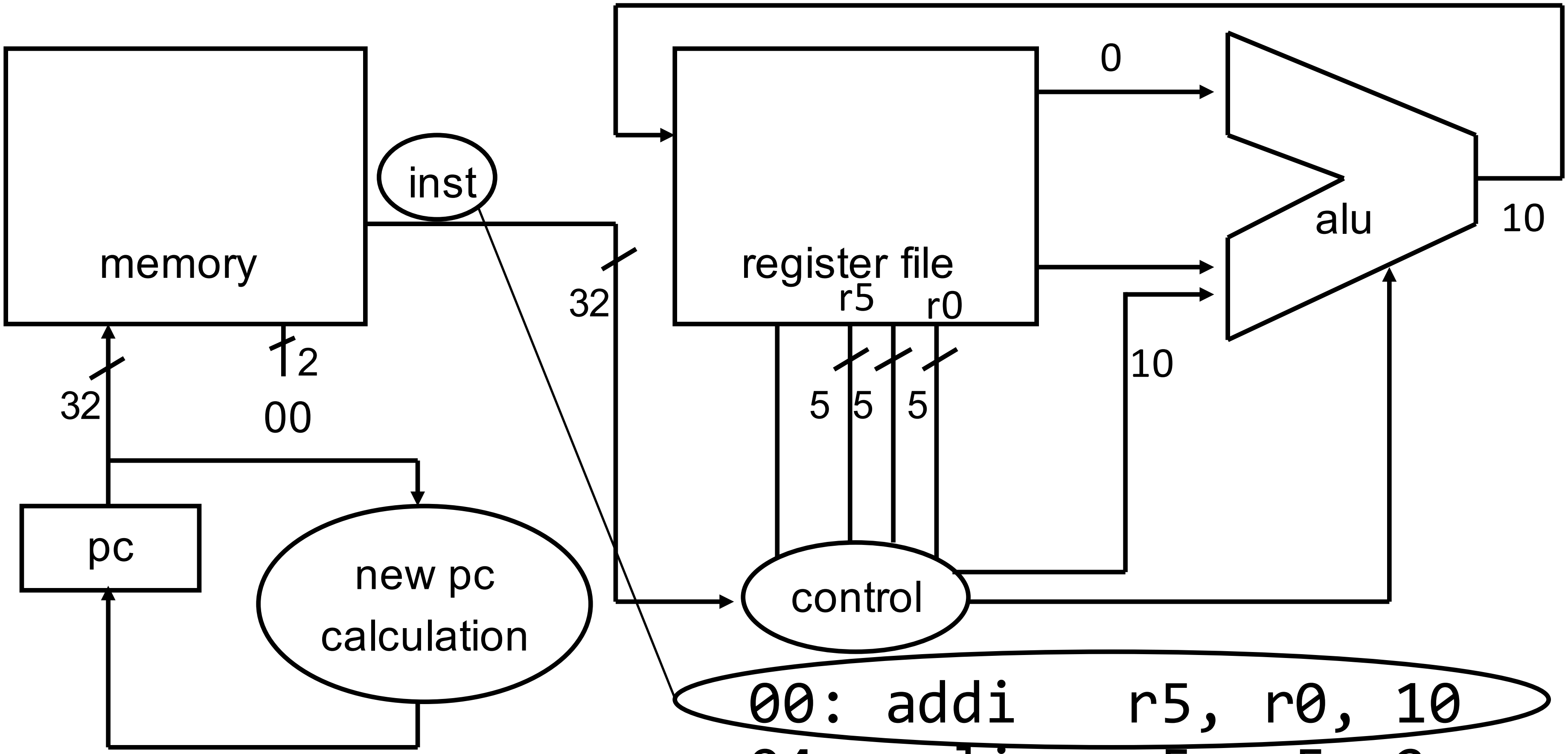
op = addi r0 r5 10

```
001000 00000 00101 0000000000001010
00000000000001010010100001000000
001000 00101 00101 0000000000001111
```

op = addi r5 r5 15

Everything is a number!

How to Design a Simple Processor



```
00: addi   r5, r0, 10
04: muli   r5, r5, 2
08: addi   r5, r5, 15
```

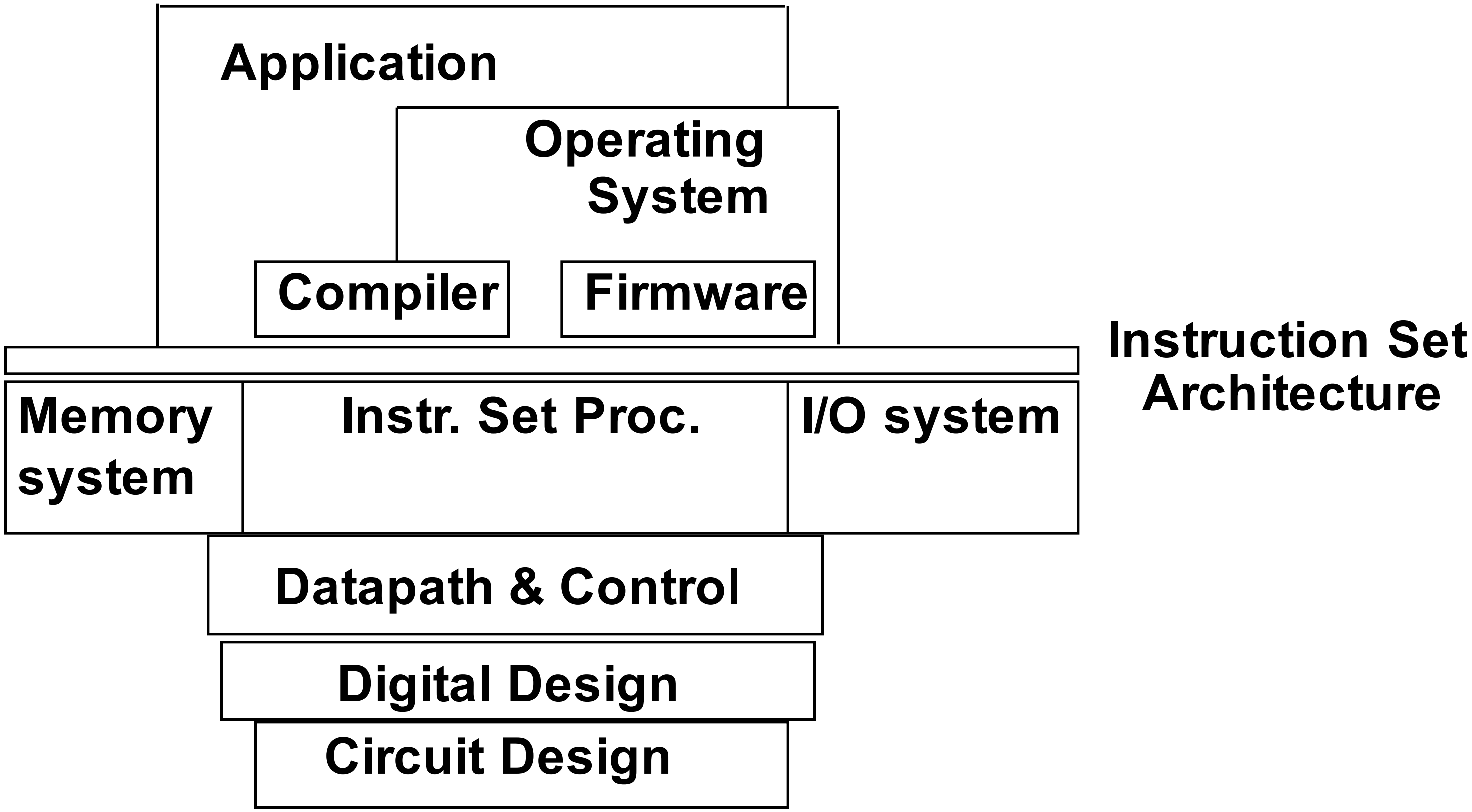
Instruction Set Architecture

ISA

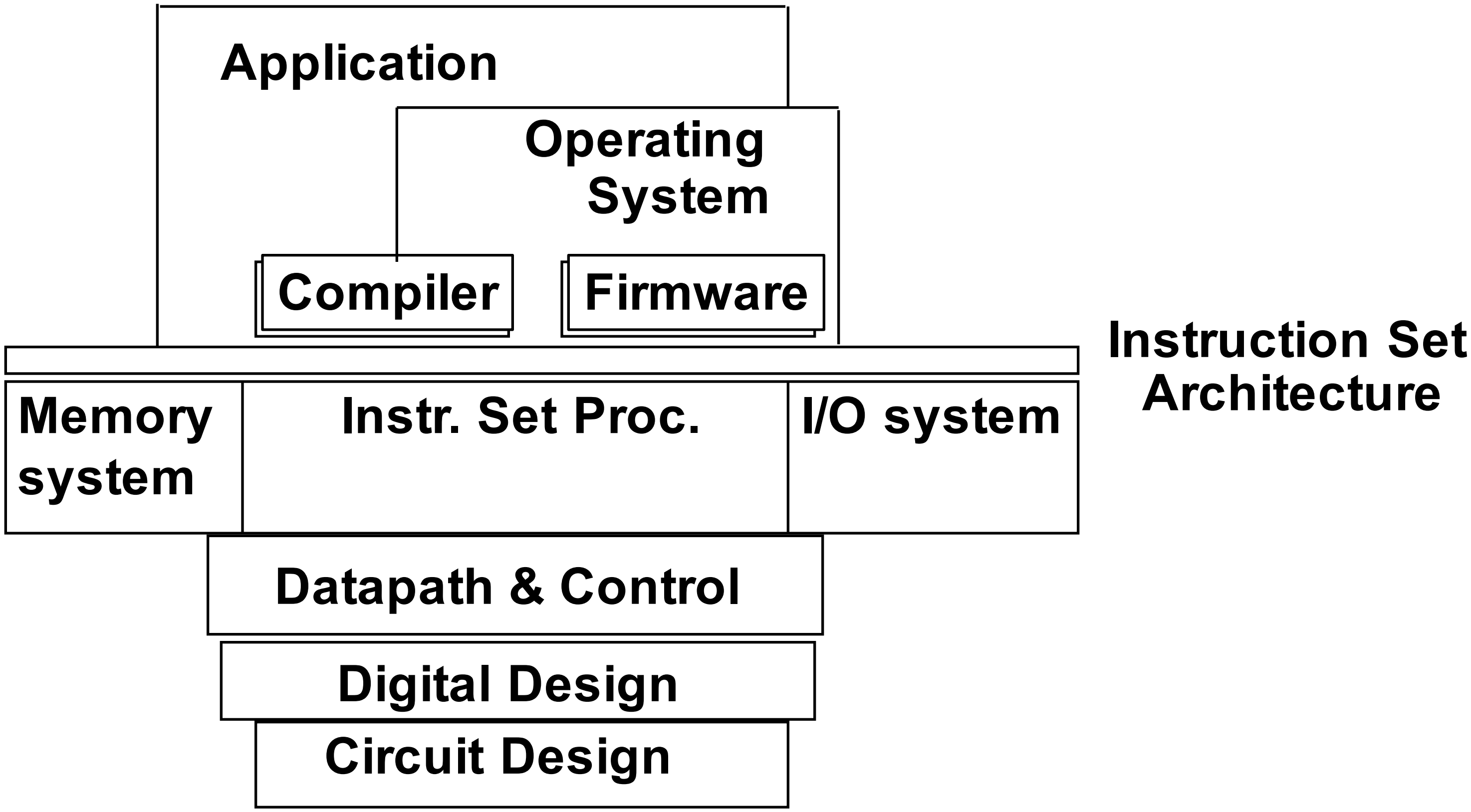
- abstract interface between hardware and the lowest level software

- user portion of the instruction set plus the operating system interfaces used by application programmers

Overview


Application
Operating System
Compiler
Firmware
Instruction Set Architecture
Memory system
Instr. Set Proc.
I/O system
Datapath & Control
Digital Design
Circuit Design

Covered in this course

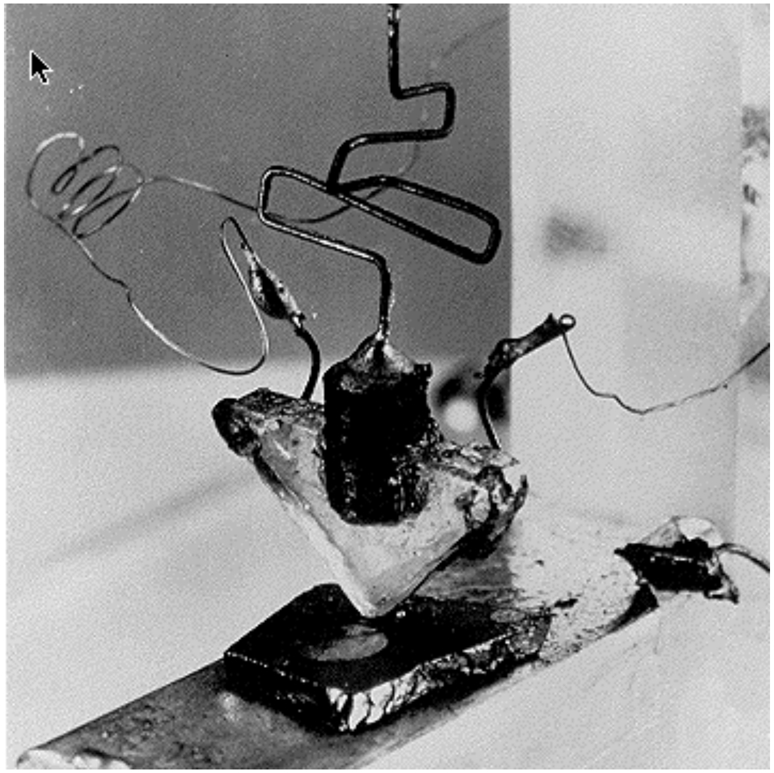


Where did it begin?

Electrical Switch

- On/Off
- Binary

Transistor



The first transistor on a workbench at AT&T Bell Labs in 1947

Transistors

Moore's Law

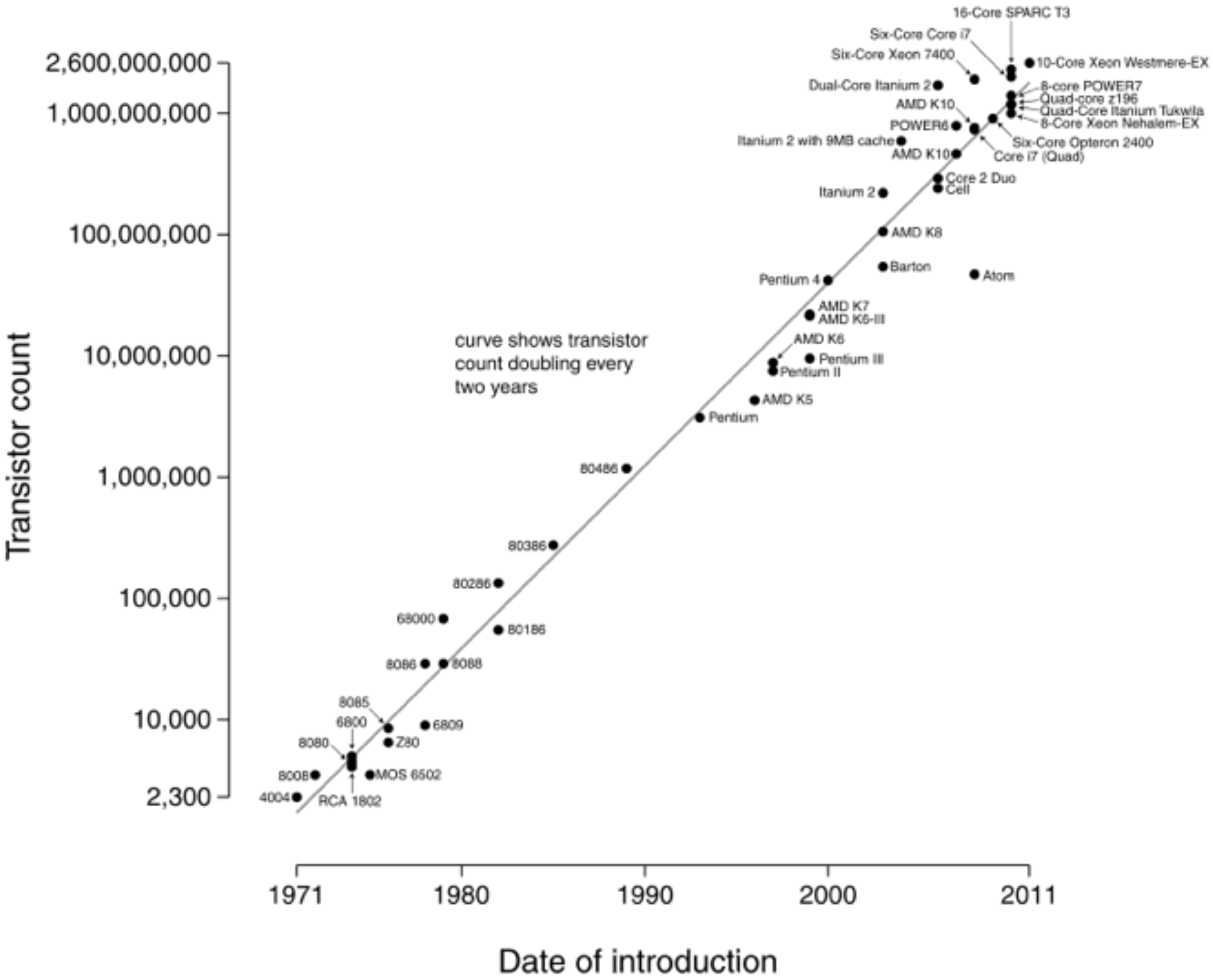
1965

- # of transistors integrated on a die doubles every 18-24 months (*i.e.,* grows exponentially with time)

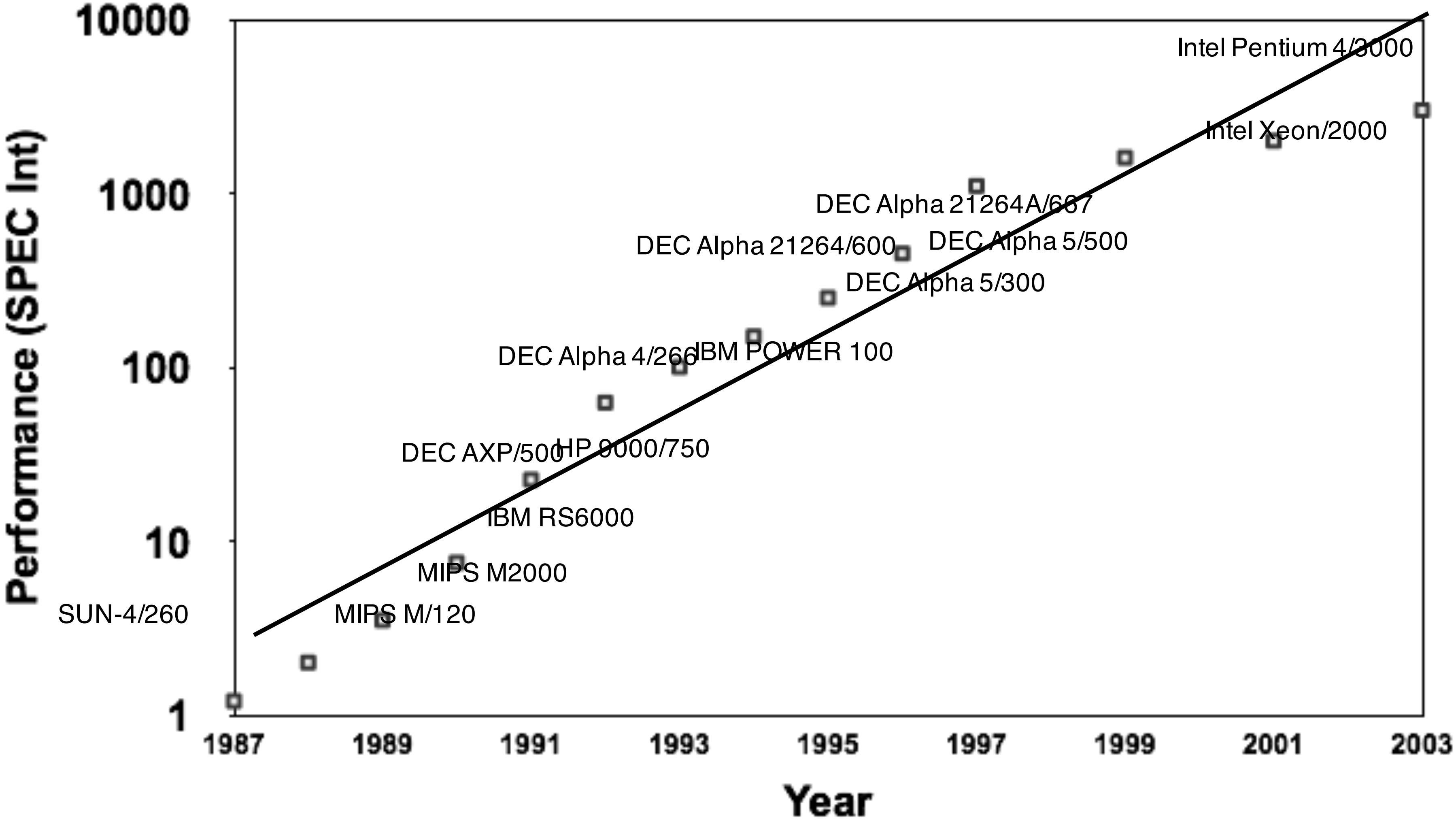
Amazingly visionary

- 2300 transistors, 1 MHz clock (Intel 4004) - 1971
- 16 Million transistors (Ultra Sparc III)
- 42 Million transistors, 2 GHz clock (Intel Xeon) – 2001
- 55 Million transistors, 3 GHz, 130nm technology, 250mm^2 die (Intel Pentium 4) – 2004
- 290+ Million transistors, 3 GHz (Intel Core 2 Duo) – 2007
- 721 Million transistors, 2 GHz (Nehalem) - 2009
- 1.4 Billion transistors, 3.4 GHz Intel Haswell (Quad core) – 2013

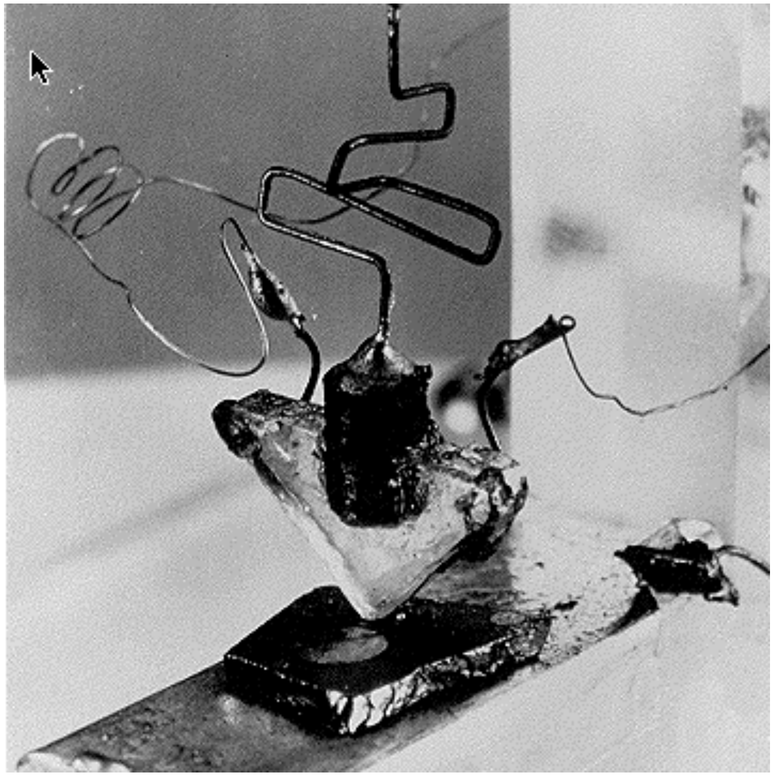
Microprocessor Transistor Counts 1971-2011 & Moore's Law

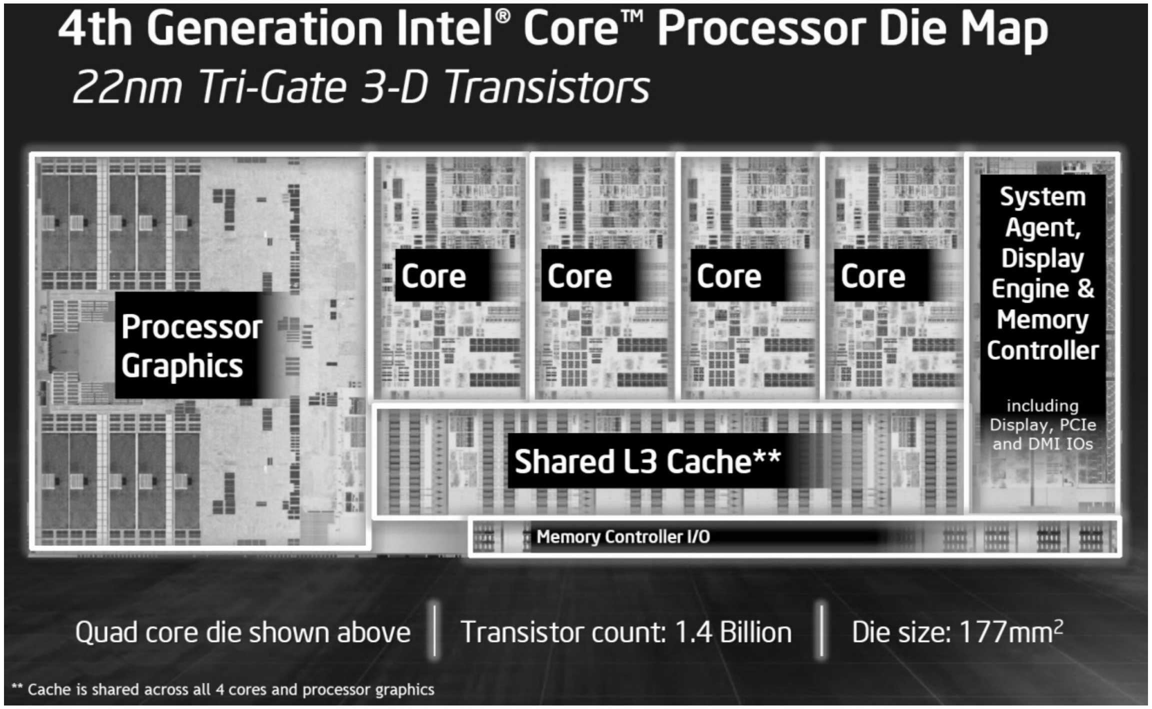


Processor Performance Increase



Then and Now




4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors
Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O
Quad core die shown above
Transistor count: 1.4 Billion
Die size: 177mm²

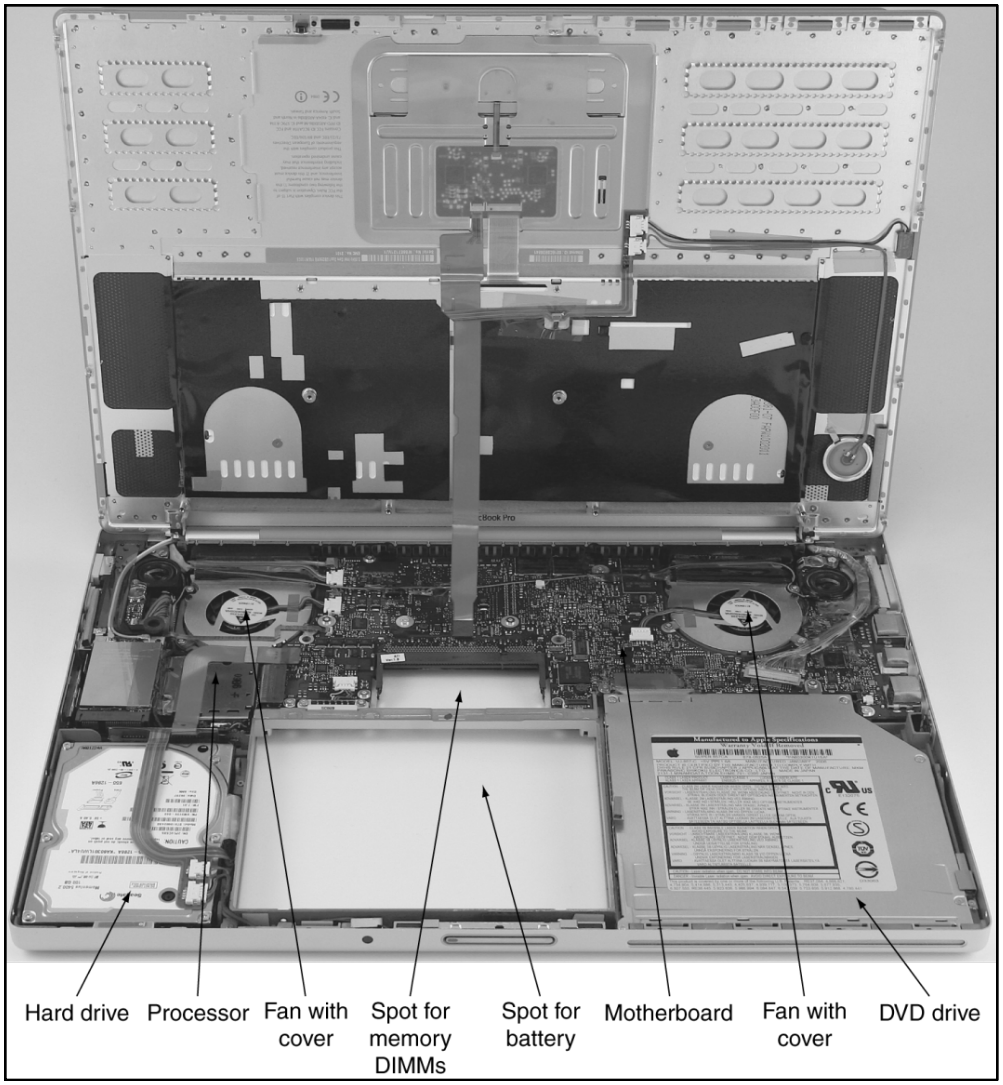
** Cache is shared across all 4 cores and processor graphics

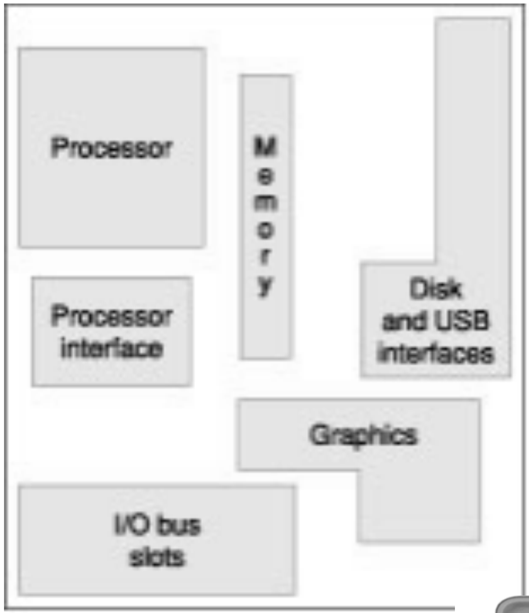
http://techguru3d.com/4th-gen-intel-haswell-processors-architecture-and-lineup/

- The first transistor
  - One workbench at AT&T Bell Labs
  - 1947
  - Bardeen, Brattain, and Shockley
- An Intel Haswell
  - 1.4 billion transistors
  - 177 square millimeters
  - Four processing cores

What are we doing with all these transistors?

Computer System Organization





Reflect

Why take this course?

- Basic knowledge needed for *all* other areas of CS:

  operating systems, compilers, ...
- Levels are not independent

  hardware design ↔ software design ↔ performance
- Crossing boundaries is hard but important

  device drivers
- Good design techniques

  abstraction, layering, pipelining, parallel vs. serial, ...
- Understand where the world is going

  The Mysteries of Computing will be revealed!