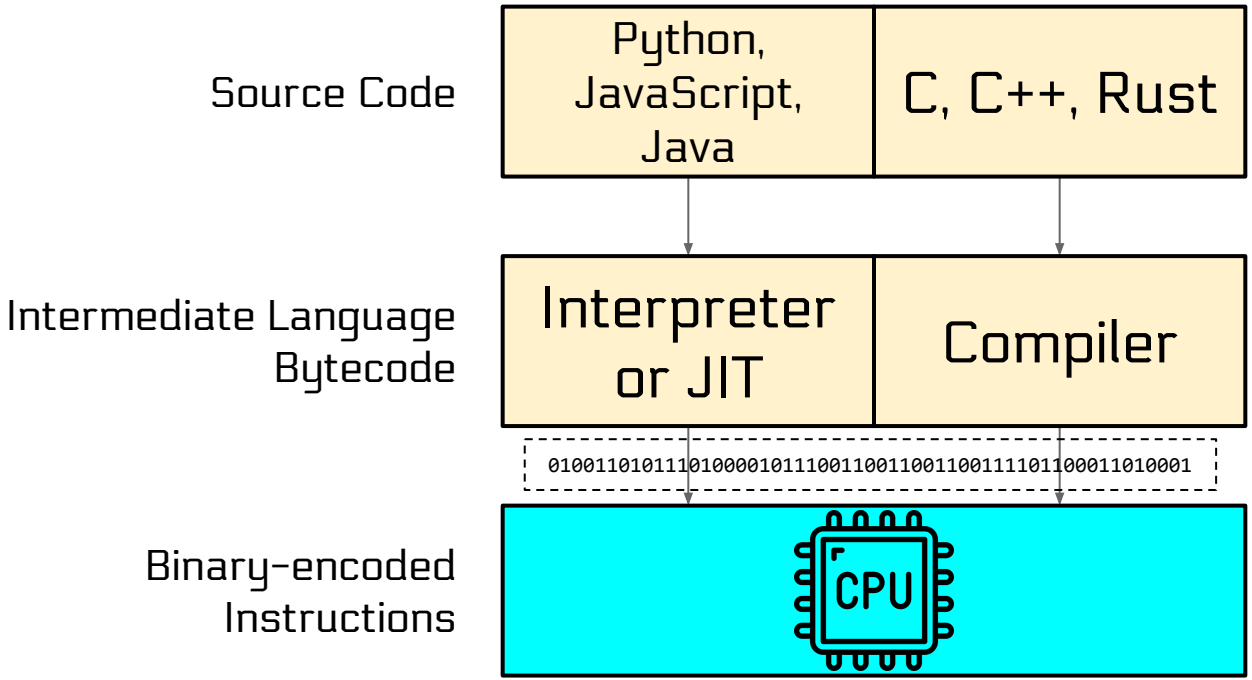
Assembly Crash Course

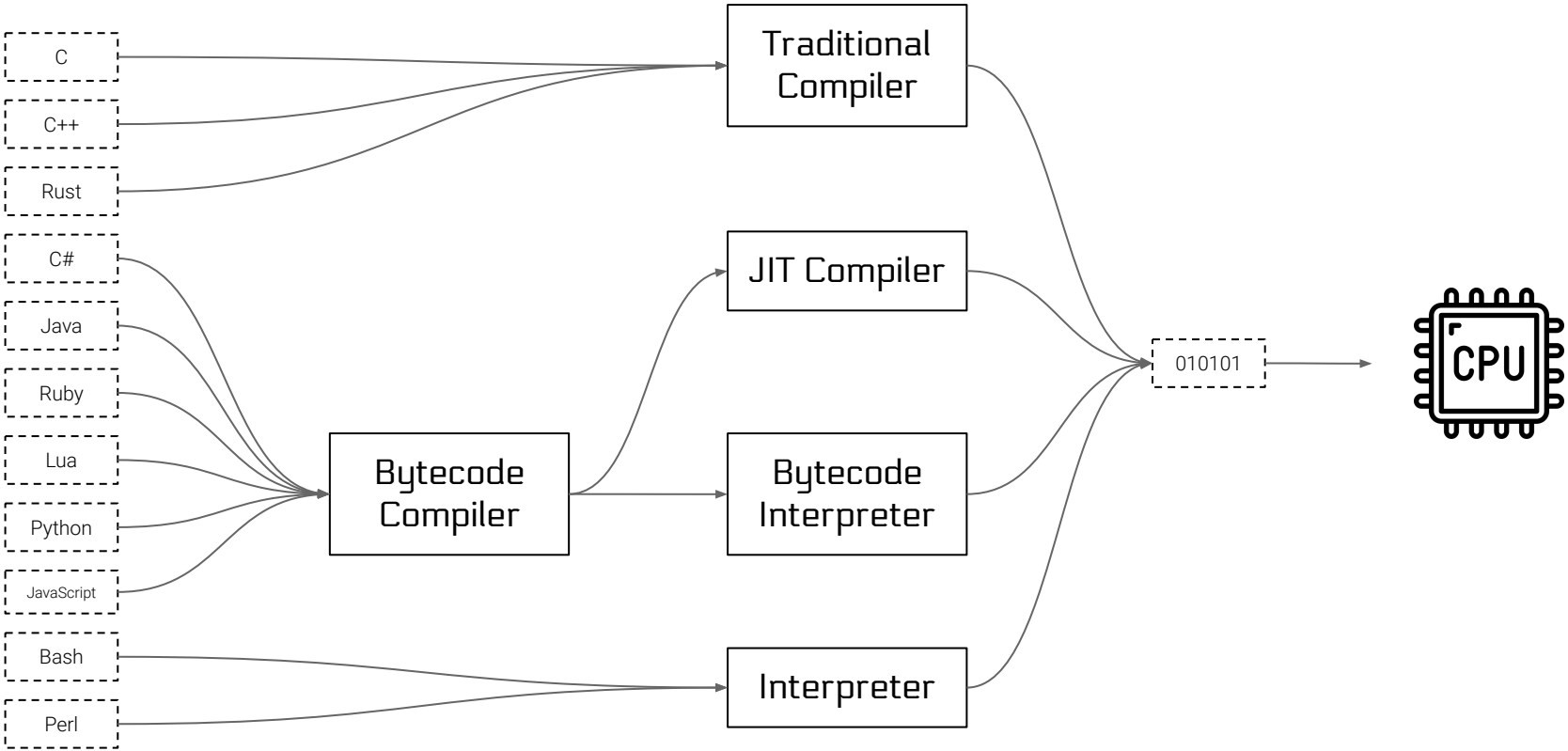
Assembly

Yan Shoshitaishvili
Arizona State University

All roads lead to the CPU



All Roads Lead to the CPU



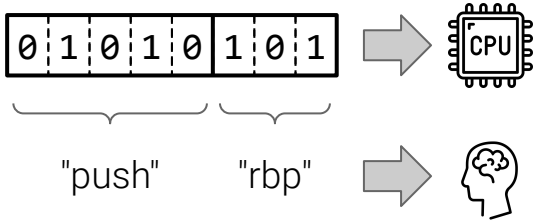
Speaking Binary

Humans have a hard time with binary code...

So we created a text *representation* of the binary...

This representation is called **Assembly**.

The binary and the assembly code is *equivalent**.

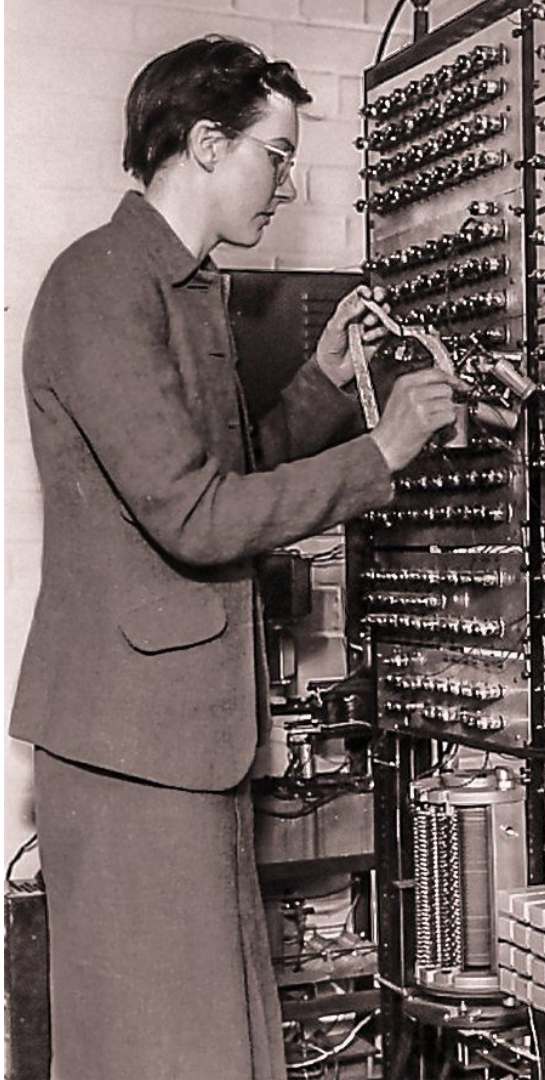


"Assembly"?

Assembly is named "Assembly" because it is *assembled* (*not* compiled) into binary code.

Invention:
Kathleen Booth,
Late 1940s/early 1950s,
For the APE(X)C (All-purpose Electronic (Rayon) Computer).

Adoption:
The second "stored-program computer" had an assembler,
Written by David Wheeler in 1948.



Assembly tells the CPU *what to do*

How do we tell people what to do? Sentences.

Let's look at an assembly "sentence" in terms of English grammar:

Sentence: we'll call this an "instruction" in assembly.
Verb: what do you want the instruction to do? We'll call this an "**operation**".
Noun: what do you want the instruction to do it *to*? We'll call this an "**operand**".

... that's it?
Simple!

Simplicity

Assembly is the **simplest** programming language.

It'd have to be, CPUs need to understand it!

You can master assembly in a week!

Nouns / Operands

What types of nouns might we deal with? Data!

For the most part, the CPU is concerned with three types of data:

data we directly give it as part of the instruction

data that is close at hand

data in storage

Verbs / Operations

What might you want to tell the computer to do with data?

Some ideas:

add some data together
subtract some data
multiply some data
divide some data
move some data into or out of storage
co**mp**are two pieces of data with each other
test some other properties of data

Now you (almost) know some Assembly!

Assembly Dialects

Assembly is a direct translation of binary code ingested by the CPU...
... so it's very CPU architecture dependent.

Every architecture has its own variant:

x86 assembly
arm assembly
ppc assembly
mips assembly
risc-v assembly
pdp-11 assembly

The list goes on! Regardless of dialect, an assembly instruction looks like one of:

```
OPERATION
OPERATION OPERAND
OPERATION OPERAND OPERAND
OPERATION OPERAND OPERAND OPERAND
```

Dialects of Assembly Dialects

In the beginning (of x86), Intel created:
... the Intel 8085 CPU
... then the Intel 8086 CPU
... then the Intel 80186 CPU
... then the Intel 80286 CPU
... then the Intel 80386 CPU, which became modern x86
... and gave us a great Assembly dialect for all of them!

AT&T came along and created a (subjectively) TERRIBLE Assembly syntax for x86.
Why? No one knows.

tl;dr: there are two competing Assembly syntaxes for x86:
the right one (Intel) and the VERY WRONG one (AT&T).

Use Intel x86 syntax. They literally made the architecture.