An ARM backend for the stage2 zig compiler

Joachim Schmidt

10th October 2020

@This()

- Joachim Schmidt

@This()

name.zig:1:1 : error: cannot pronounce

- Joachim Schmidt

@This()

- Joachim Schmidt
- Technische Universität Darmstadt

Overview

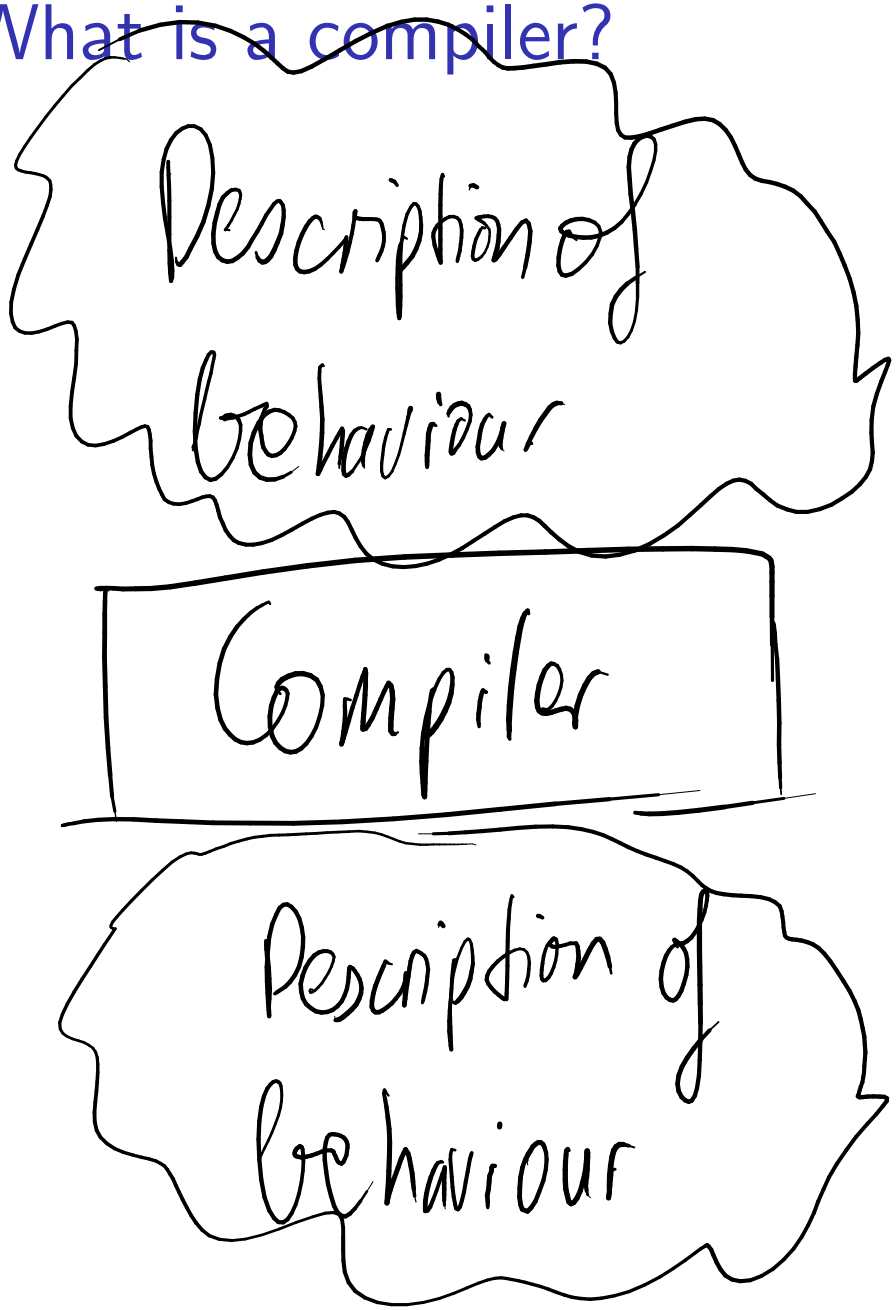
Compiler backends and stage2

ARM architecture

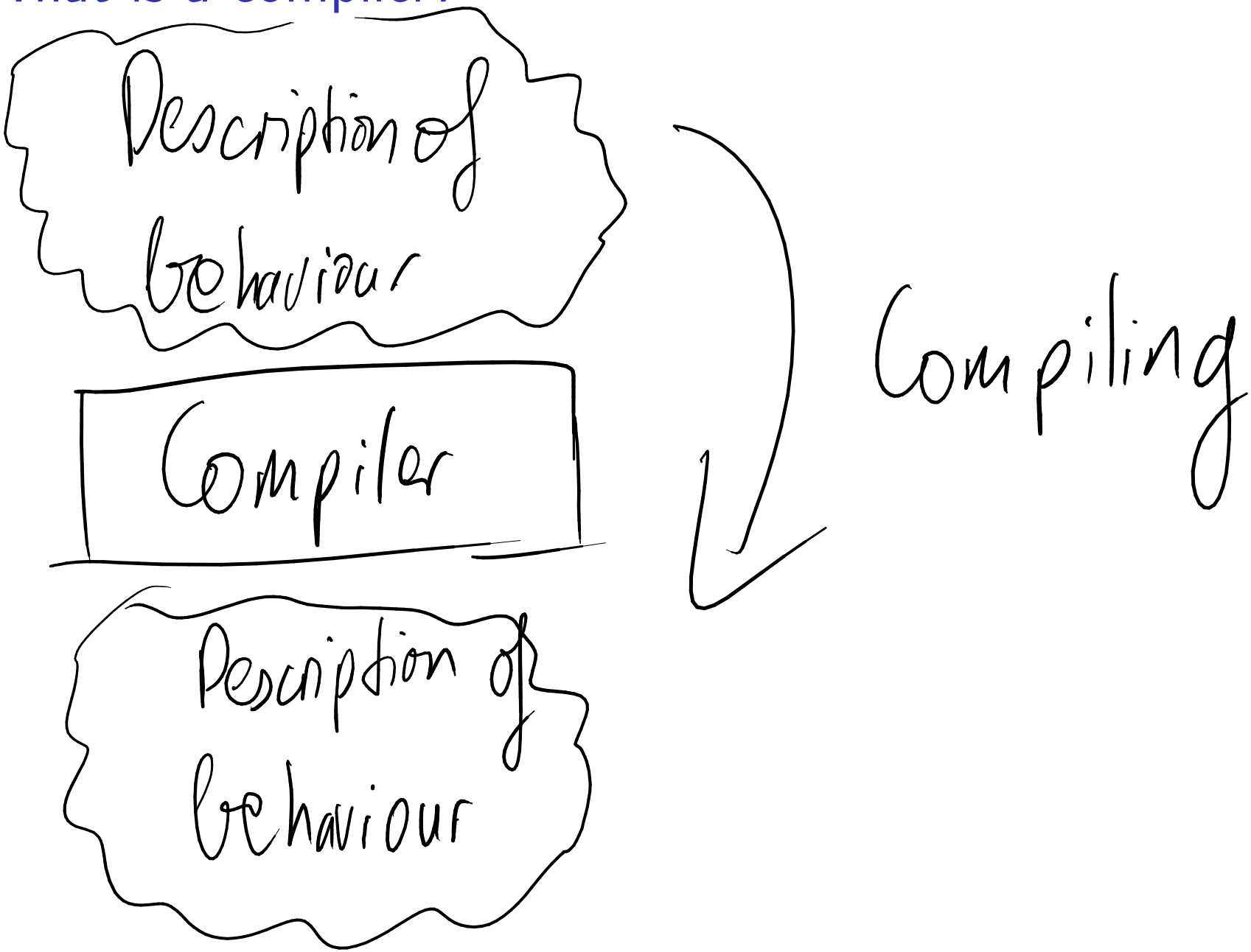
Live demo on a Raspberry Pi

Q & A

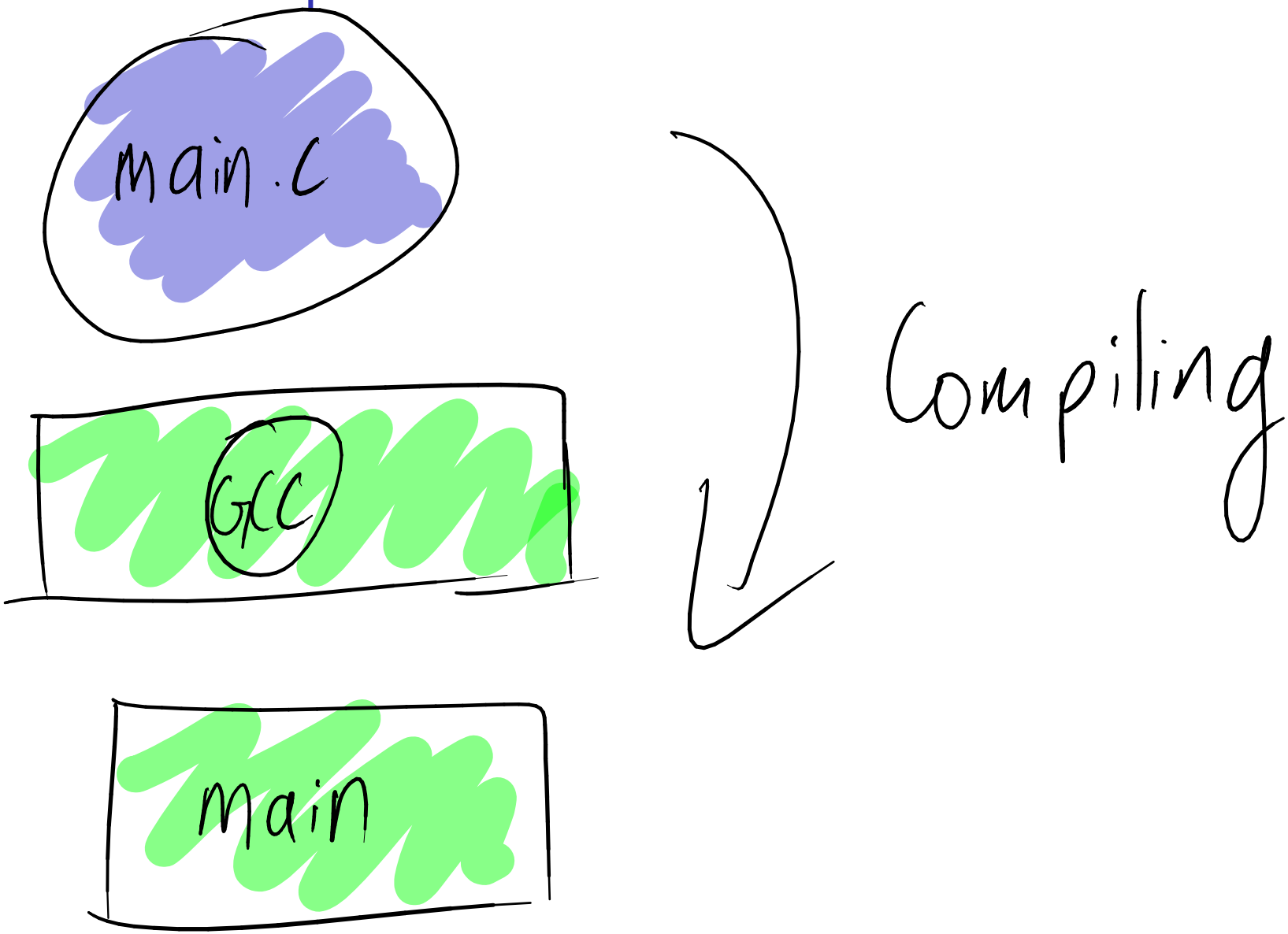
What is a compiler?



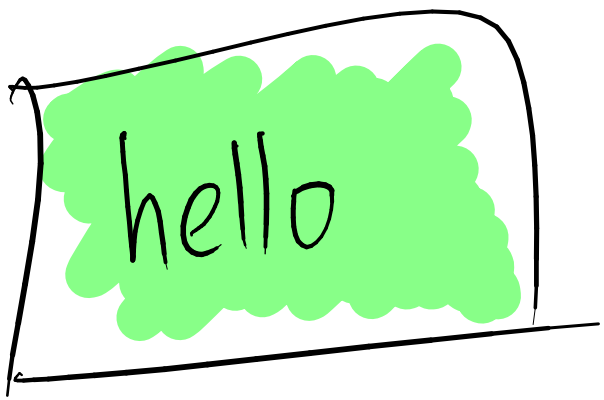
What is a compiler?



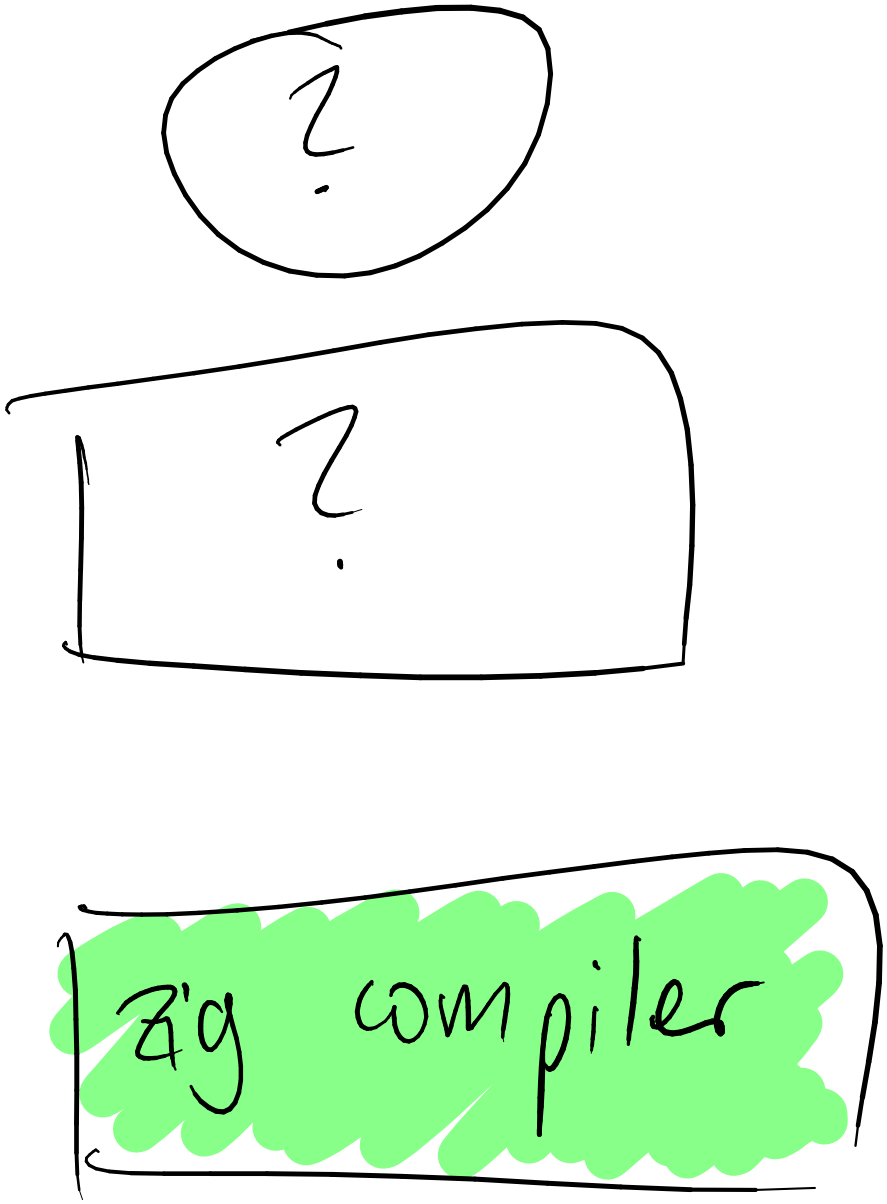
What is a compiler?



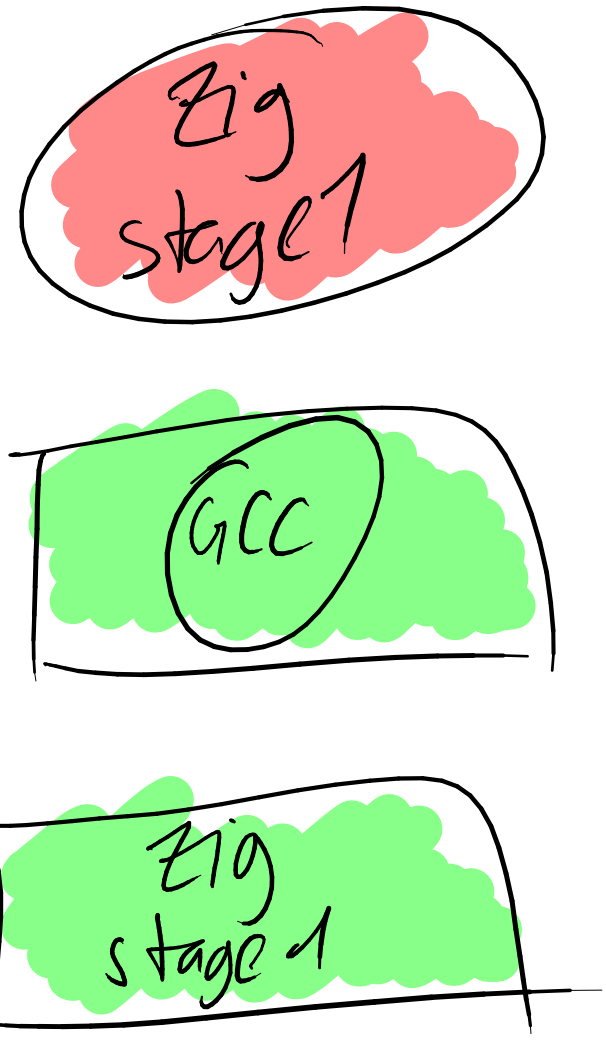
Compiling Hello World in zig



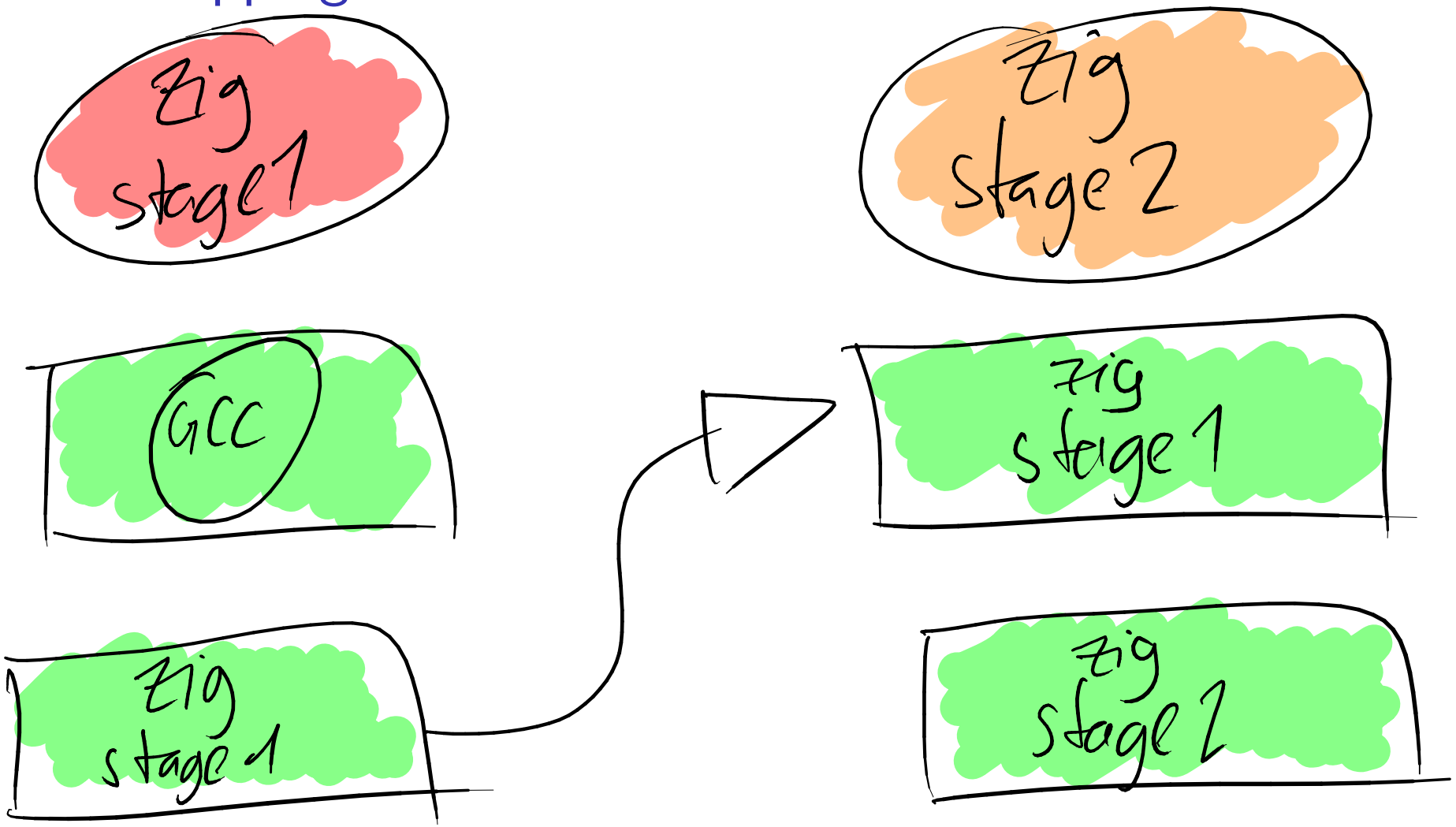
Where did the zig compiler come from?



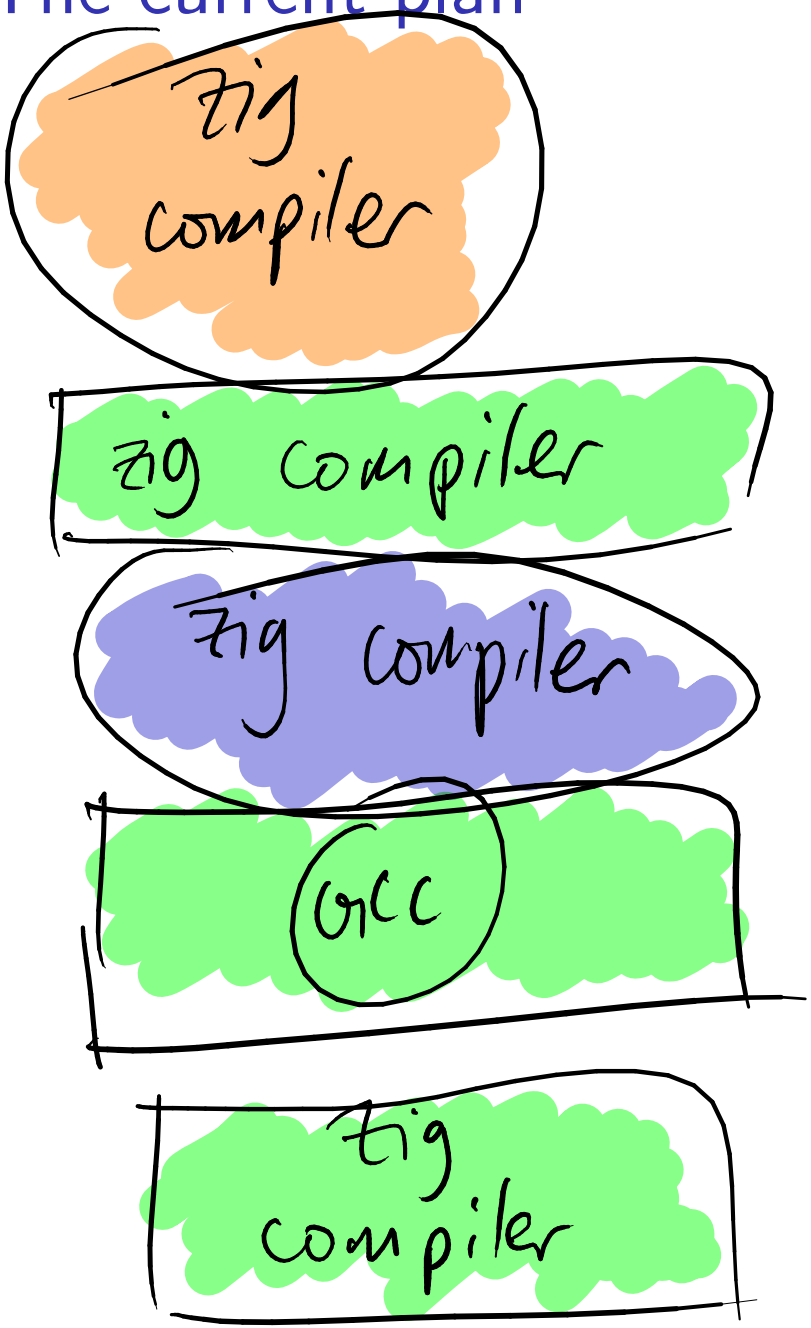
Bootstrapping



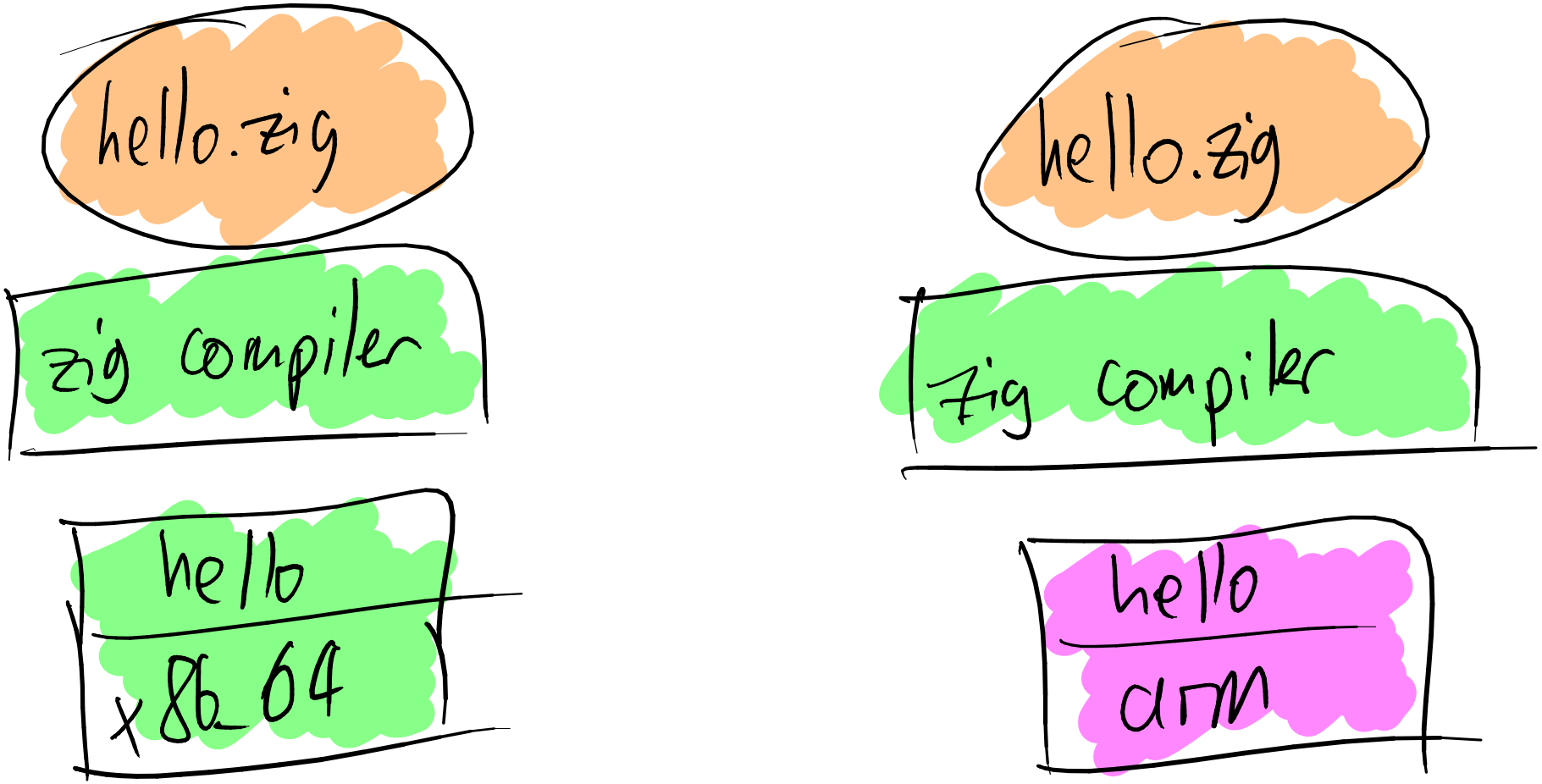
Bootstrapping



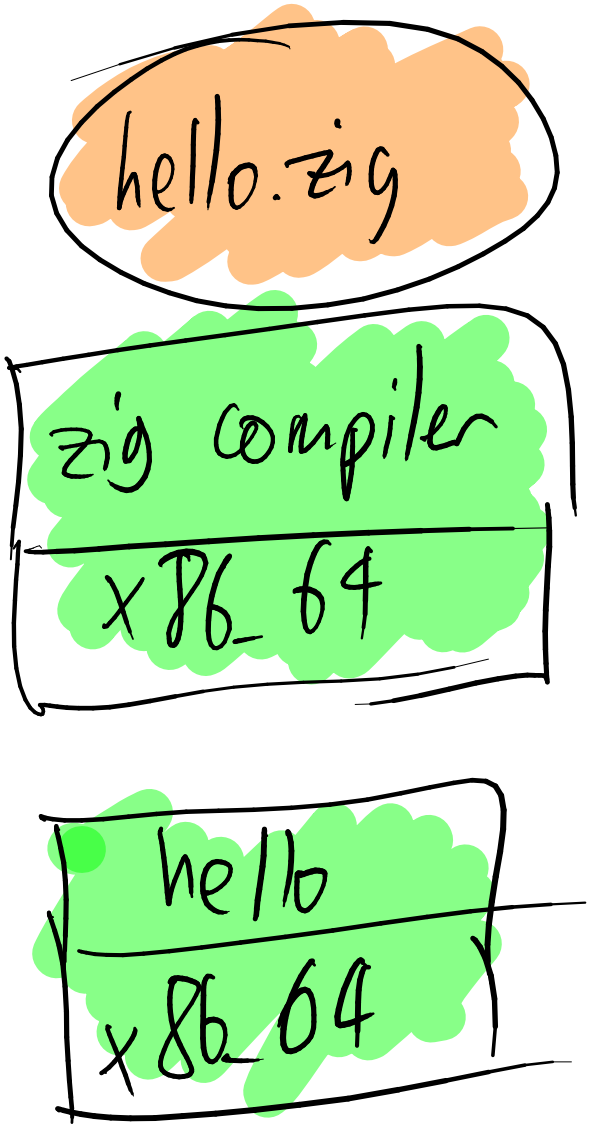
The current plan



Backends


hello.zig
zig compiler
hello
x86_64
hello.zig
zig compiler
hello
arm

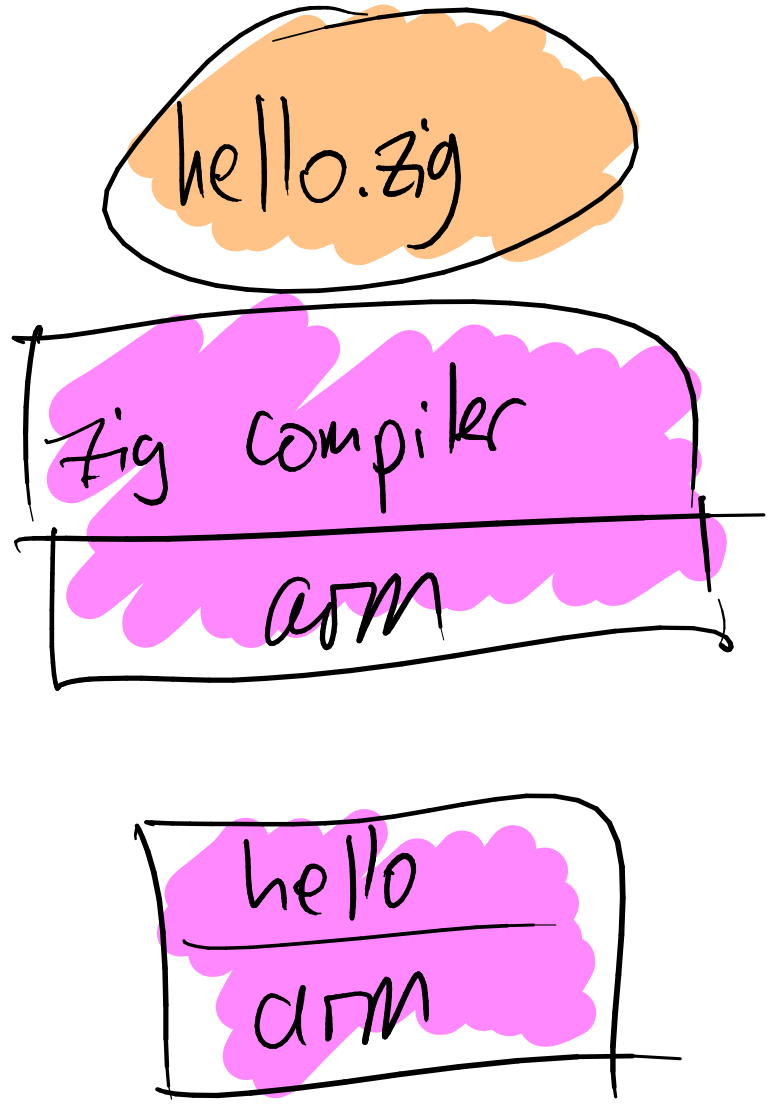
Backends


hello.zig
zig compiler
x86_64
hello
x86_64

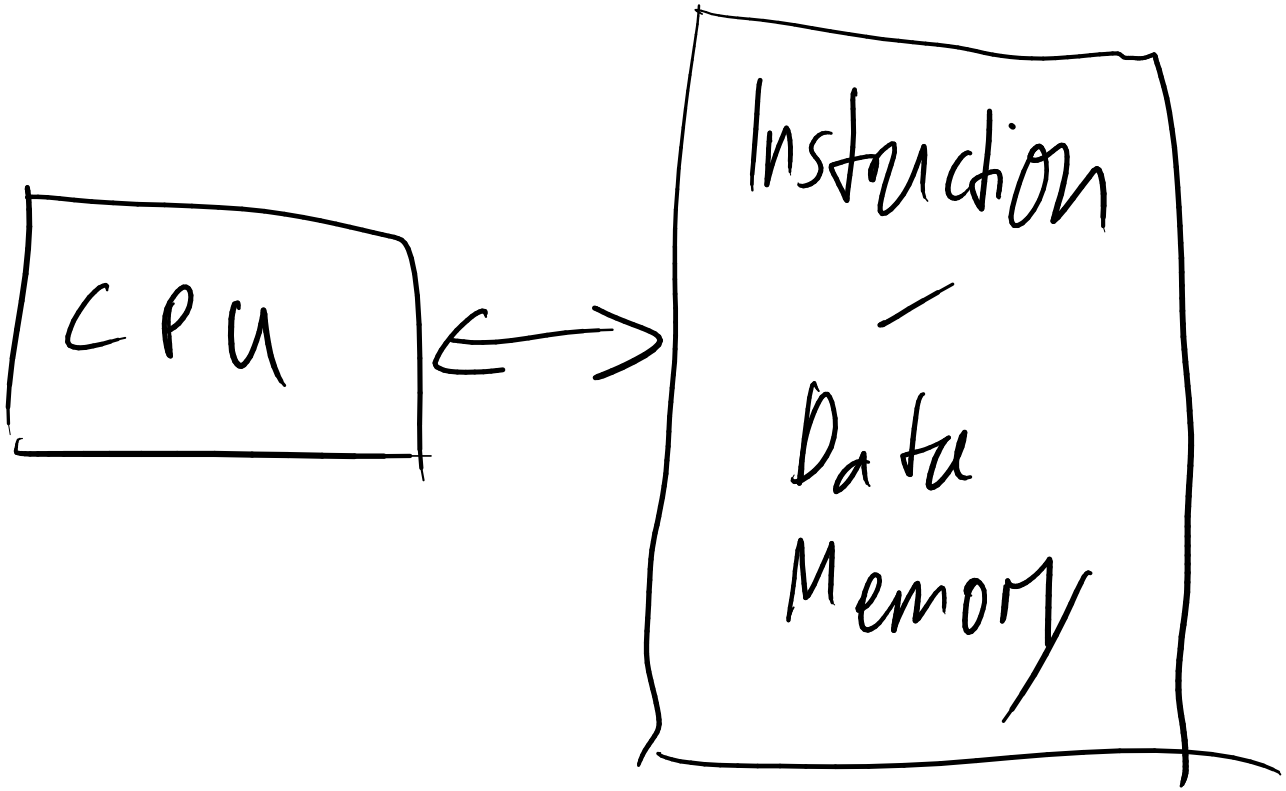

hello.zig
zig compiler
x86_64
hello
arm

Backends

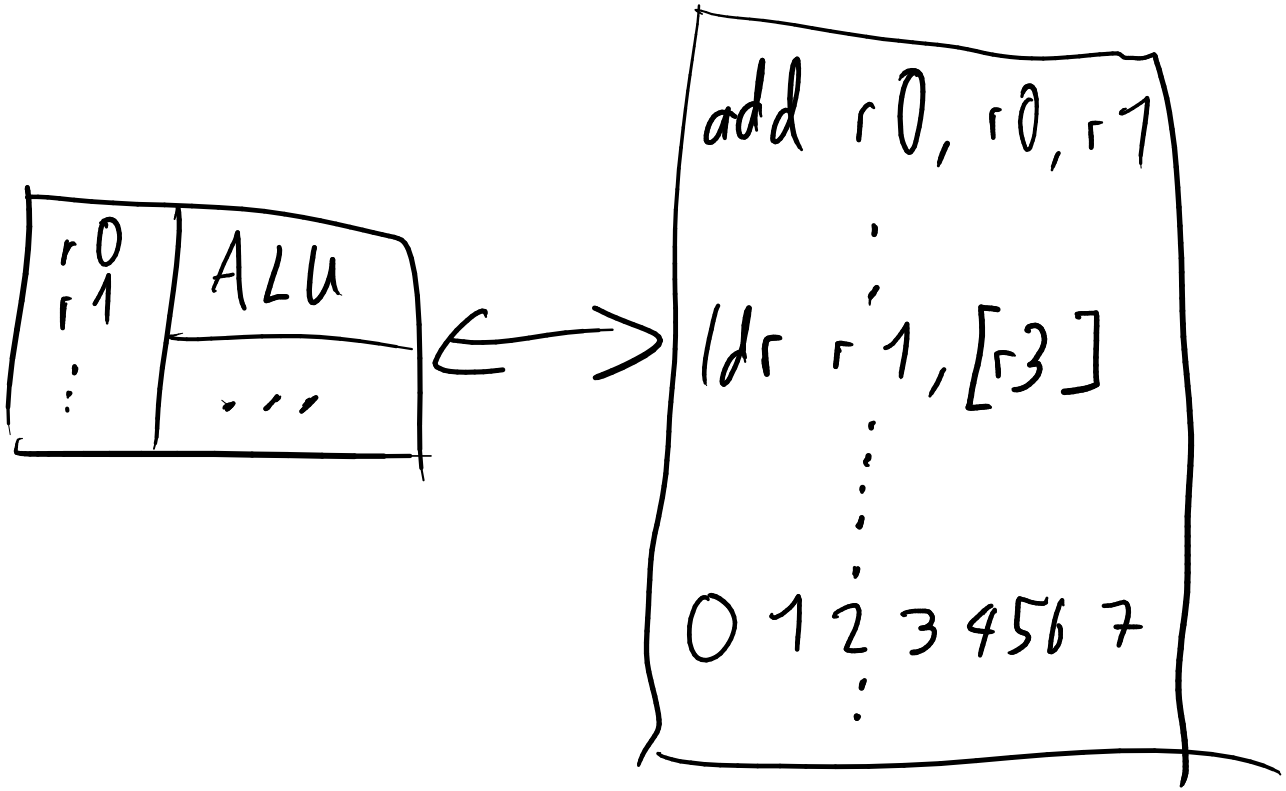

hello.zig
zig compiler
arm
hello
x86_64


hello.zig
zig compiler
arm
hello
arm

von Neumann architecture



von Neumann architecture



CISC

```
export fn foo() u32 {
    var x: u32 = 42;
    var y: u32 = 23;
    return x + y;
}
```

```
...
mov     dword ptr [rbp - 8], 42
mov     dword ptr [rbp - 12], 23
mov     eax, dword ptr [rbp - 8]
add     eax, dword ptr [rbp - 12]
...
```

https://zig.godbolt.org/z/qoMfEK

RISC

```
export fn foo() u32 {
    var x: u32 = 42;
    var y: u32 = 23;
    return x + y;
}
```

```
...
movw    r0, #42
str     r0, [sp, #8]
movw    r0, #23
str     r0, [sp, #4]
ldr     r0, [sp, #8]
ldr     r1, [sp, #4]
adds    r0, r0, r1
...
```

https://zig.godbolt.org/z/Mee9ro

ARM instruction set

- Data Processing

```
add r0, r1, #2
```

ARM instruction set

- Data Processing

```
add r0, r1, #2
```

- Memory

```
ldr r0, [sp, #4]
```

ARM instruction set

- Data Processing

```
add r0, r1, #2
```

- Memory

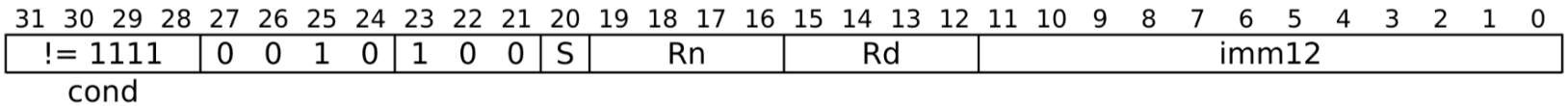
```
ldr r0, [sp, #4]
```

- Branching

```
b label
bx lr
```

A brief look at the add instruction

```
add r0, r1, #2
```



ADD (S == 0 && Rn != 11x1)

```
ADD{<c>}{<q>} {<Rd>,} <Rn>, #<const>
```

Figure: Encoding of add; Source: [2]

Condition codes

```
export fn isAnswer(x: u32) u32 {
    return if (x == 42) 32
        else 12;
}
```

```
mov     r1, #12
cmp     r0, #42
movweq  r1, #32
mov     r0, r1
bx      lr
```

https://zig.godbolt.org/z/q5rP8e

Live demo

Figure: Source: [1]

Q & A

- GitHub: `https://github.com/joachimschmidt557`
- Discord: `joachim.schmidt557#6869`
- Matrix: `@joachimschmidt557:matrix.org`

References


https://www.flickr.com/photos/rexroof/3802694376/


https://static.docs.arm.com/ddi0597/h/ISA_AArch32_xml_v86A-2020-06.pdf