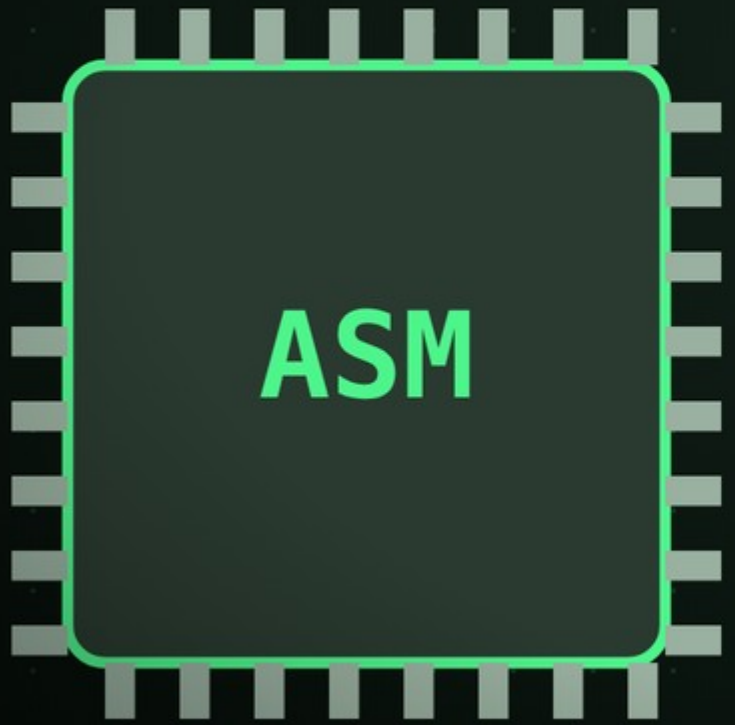
@MyWebUniversity

ASM · FPGA

Programming Close to the Metal

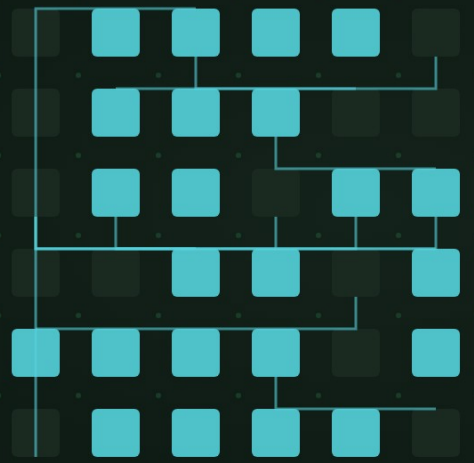



```
mov rax, 1 · syscall
```

WHEN IS ASSEMBLY WORTH IT?

```
_start:
  mov  rax, 1
  mov  rdi, 1
  lea  rsi, [msg]
  mov  rdx, 14
  syscall
```

assembly



FPGA

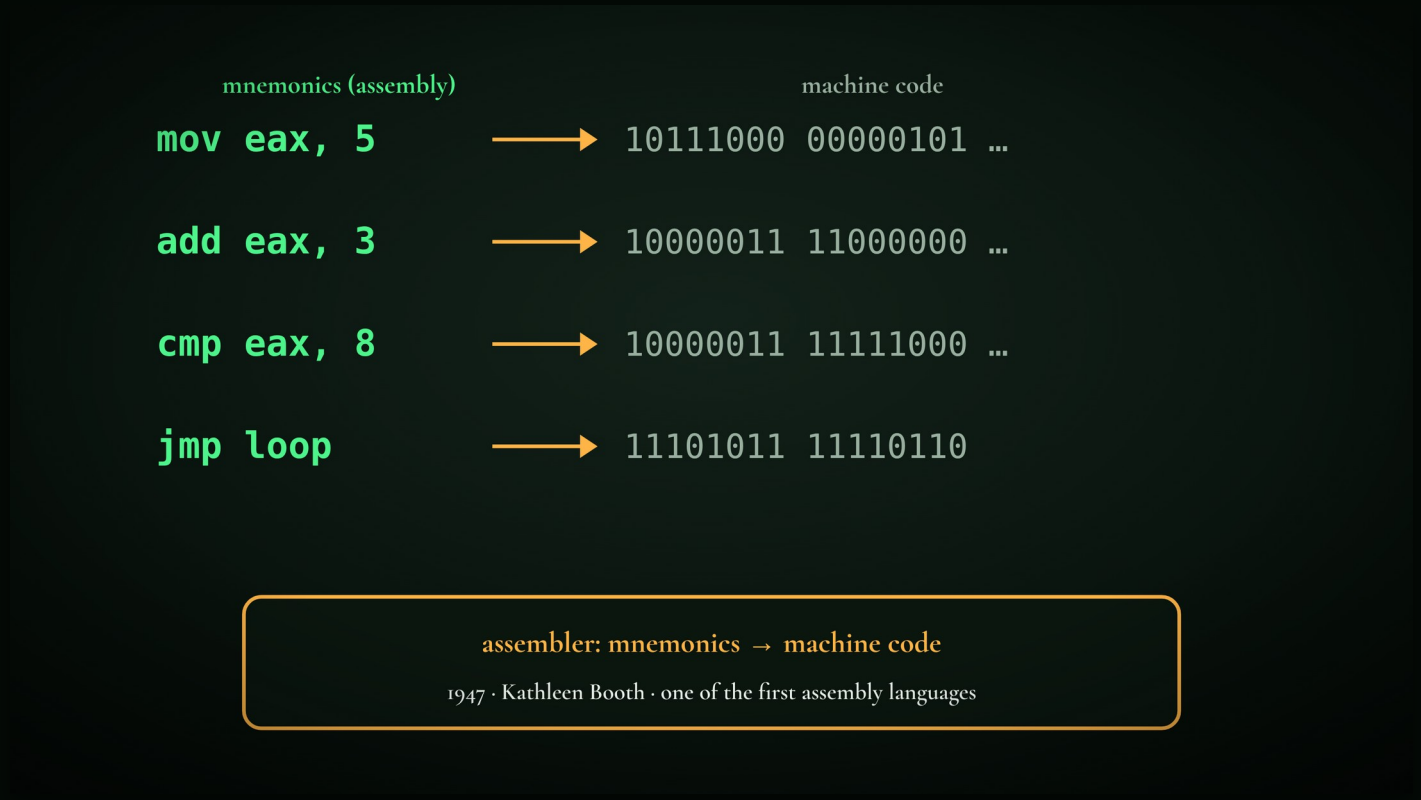
A MYWEBUNIVERSITY STORY

ASM · FPGA

Programming Close to the Metal

x86 · ARM · RISC-V · Verilog

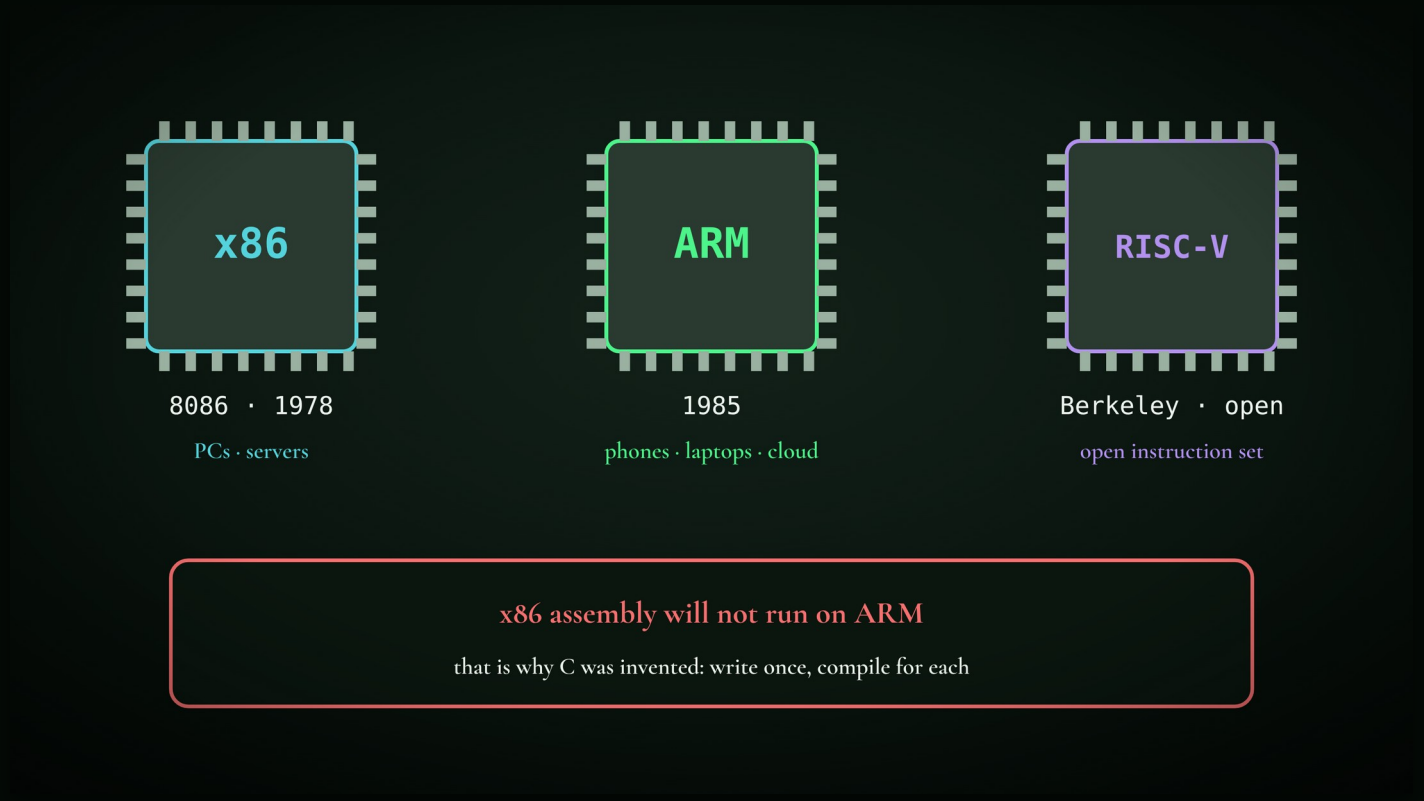
The Language of the CPU



- CPUs understand machine code (binary)
- Assembly: mnemonics (mov, add, jmp) → assembler
- 1947: Kathleen Booth, early assembly language

01

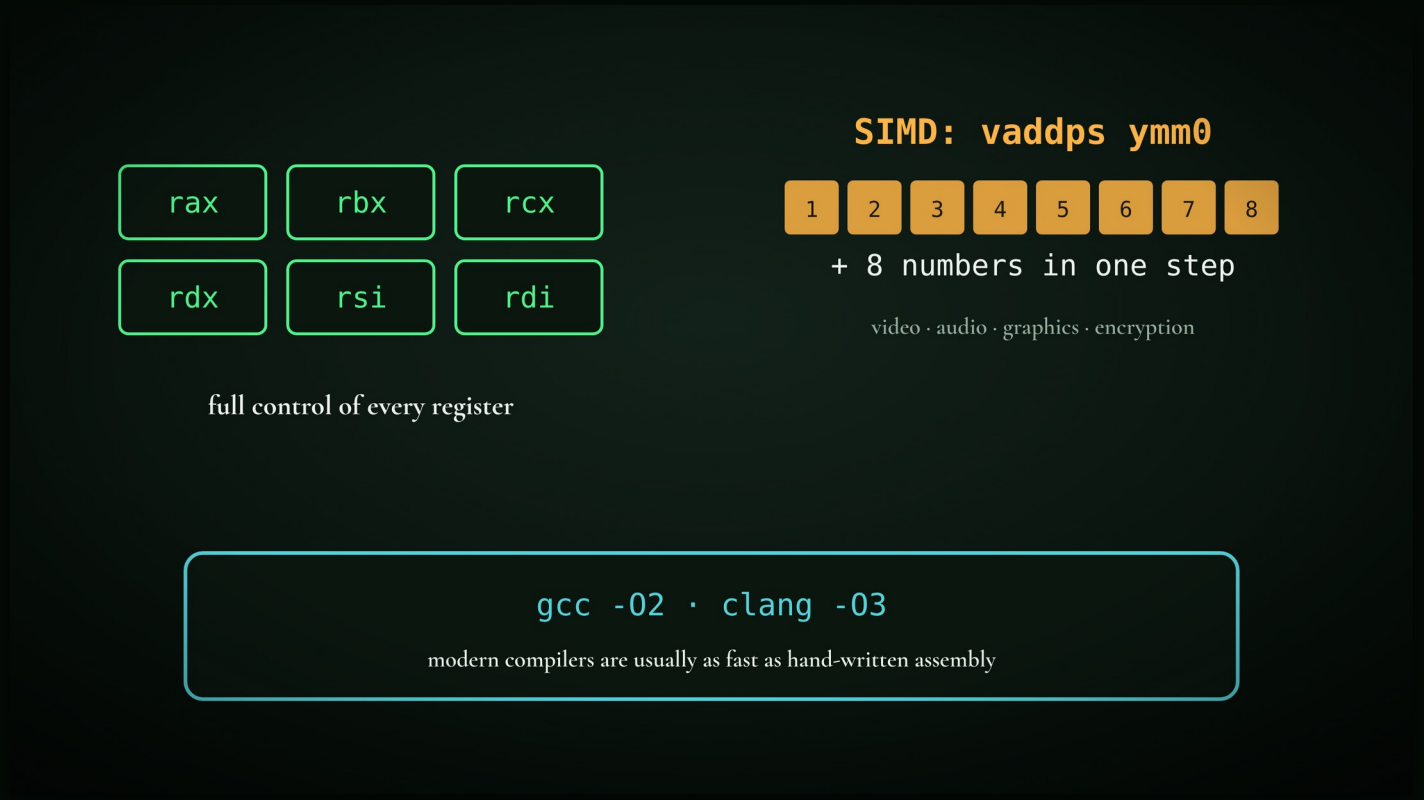
Every CPU Family Is Different



- x86 (8086, 1978): PCs and servers
- ARM (1985): phones, laptops, cloud · RISC-V: open (Berkeley)
- Assembly isn’t portable → why C exists

02

Why Assembly Can Be Faster

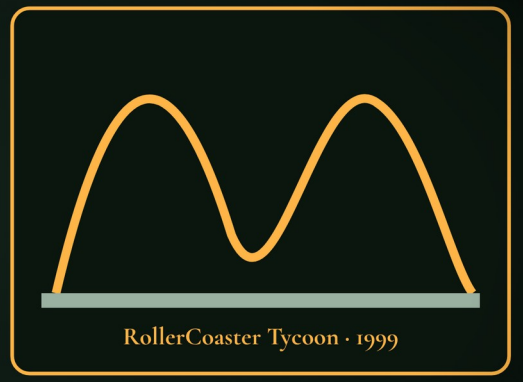


- Full control of registers and instructions
- SIMD: 8–16 numbers per instruction
- Modern optimizing compilers are usually just as good

03

Where Assembly Still Lives

- boot code
- interrupt handlers
- OS & driver hot spots
- FFmpeg · OpenSSL loops
- microcontrollers



Chris Sawyer: almost all x86 assembly

- Boot code · interrupt handlers · driver/OS hot spots
- FFmpeg and OpenSSL hot loops · microcontrollers
- 1999: RollerCoaster Tycoon, almost all x86 assembly

04

The Cost of Assembly

```
total += a[i] * b[i];
```

C: one line

drivers & game engines today:

C / C++

+ small pieces of assembly / intrinsics

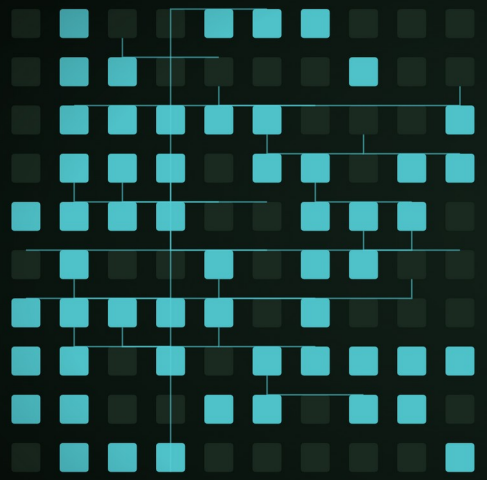
```
mov  rax, [rbp-16]
mov  rdx, [rbp-24]
mov  ecx, [rax+rdi*4]
imul ecx, [rdx+rdi*4]
add  [rbp-4], ecx
inc  rdi
cmp  rdi, rsi
jl   .loop
; …and one wrong register = crash
```

assembly: many lines, not portable

- Not portable · slow to write · easy to break
- Drivers and engines: mostly C/C++
- Small pieces in assembly or intrinsics

05

FPGA: Hardware You Can Reprogram



logic blocks + wires you can rewire

1985 · first commercial FPGA (Xilinx)

```
module adder(
  input  [7:0] a, b,
  output [8:0] sum);
  assign sum = a + b;
endmodule
// Verilog: describes HARDWARE
```

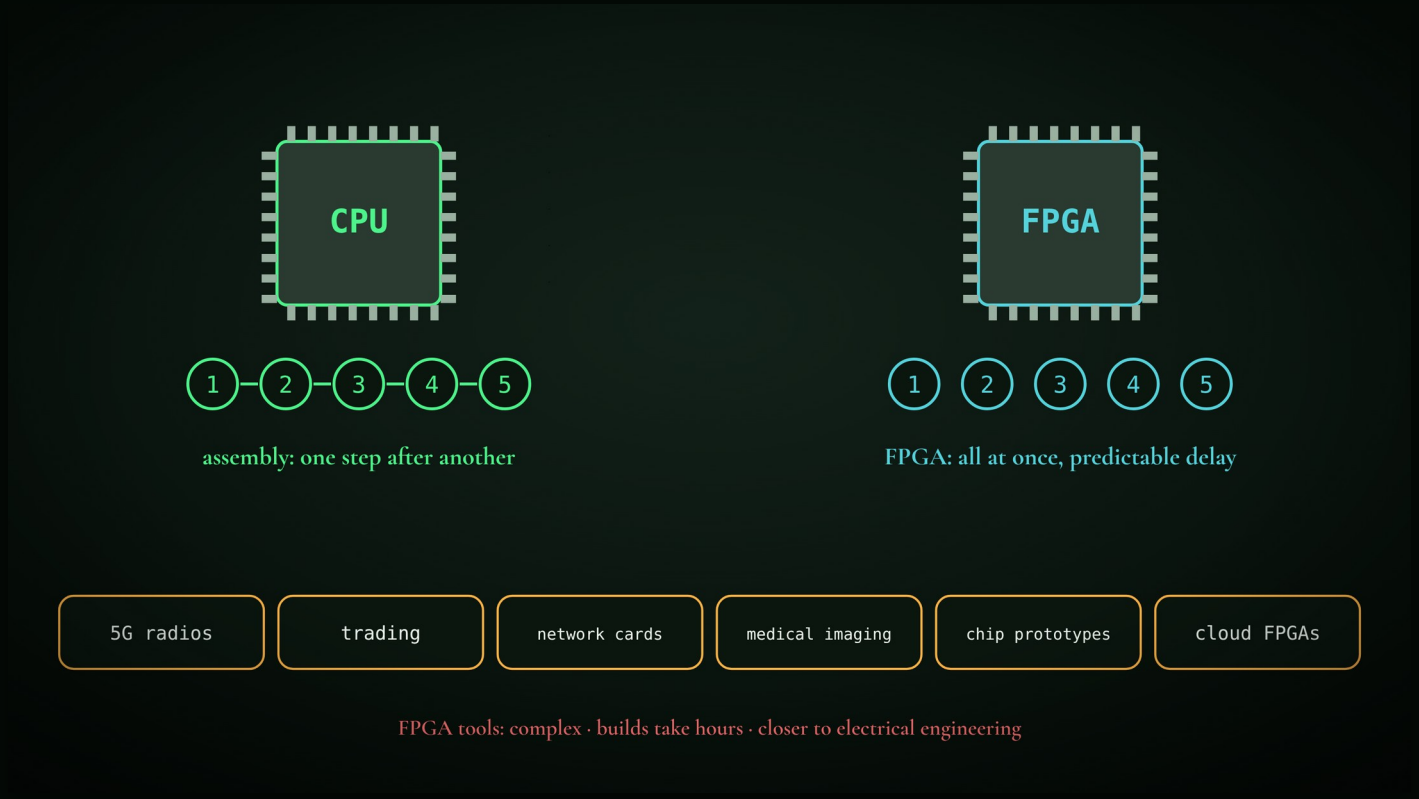
not steps — circuits

thousands of operations at the same time

- Field-Programmable Gate Array: reconfigurable circuits
- 1985: first commercial FPGA (Xilinx)
- Verilog · VHDL: describe hardware, massive parallelism

06

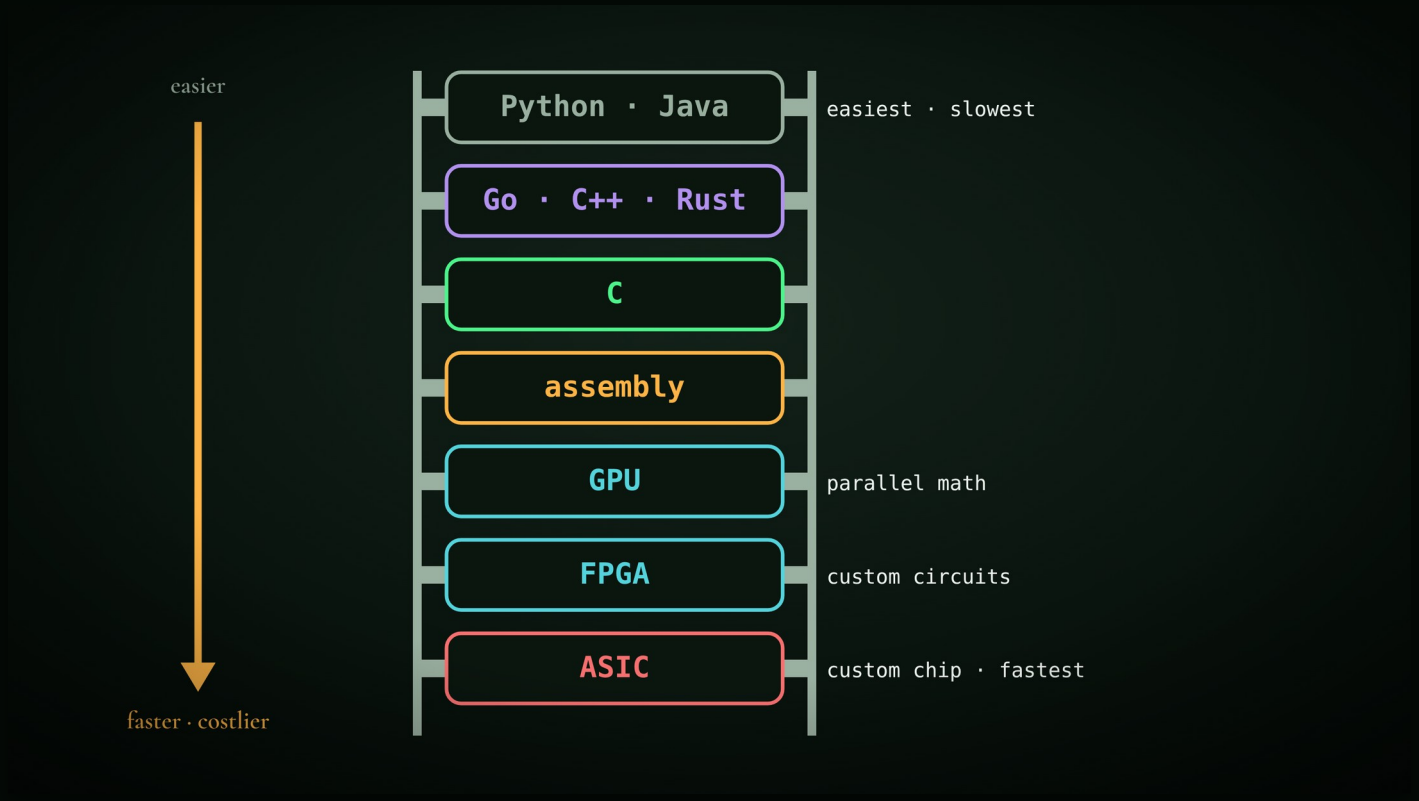
Assembly or FPGA?



- Assembly: one CPU, sequential, as fast as possible
- FPGA: custom parallel circuit, low predictable latency
- 5G, trading, network cards, imaging, chip prototyping, cloud FPGAs

07

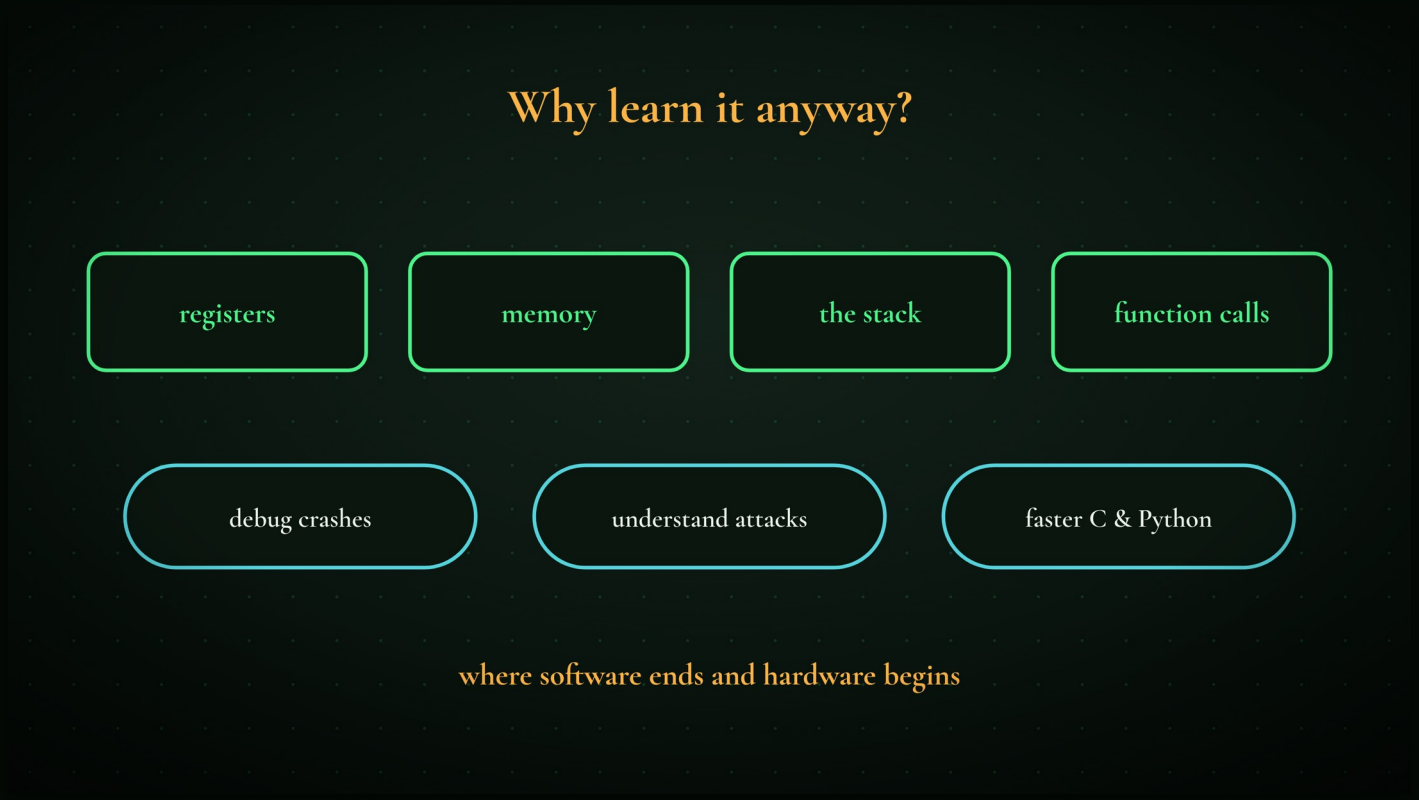
The Ladder of Choices



- Python/Java → Go/C++/Rust → C → assembly
- → GPU → FPGA → ASIC
- Go only as low as the problem needs

08

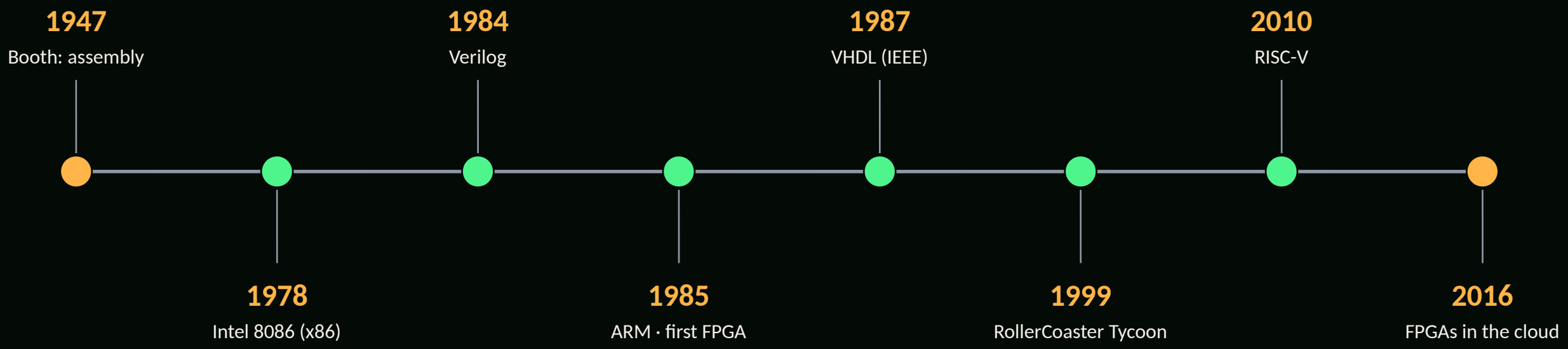
Why Learn It Anyway?



- Registers · memory · stack · calls
- Better debugging and security understanding
- Where software ends and hardware begins

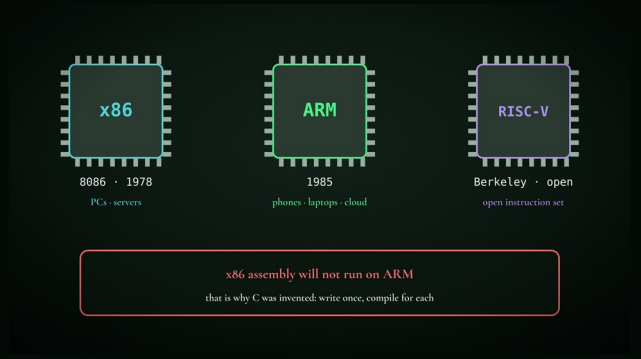
09

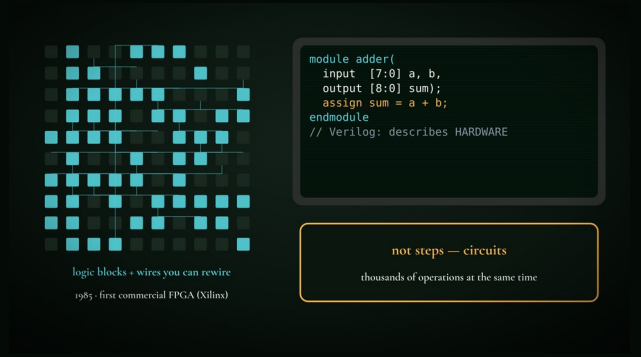
Timeline: 1947 to 2016



Sources and Further Reading

1. Intel® 64 and IA-32 Architectures Software Developer's Manuals — intel.com
2. Arm Architecture Reference Manual — developer.arm.com · RISC-V International — riscv.org
3. Xilinx (AMD), history of the FPGA (XC2064, 1985)
4. IEEE 1364 (Verilog) and IEEE 1076 (VHDL) standards
5. FFmpeg project (hand-written assembly) — ffmpeg.org





Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari