Intro to Internals

A Peek At What's Inside

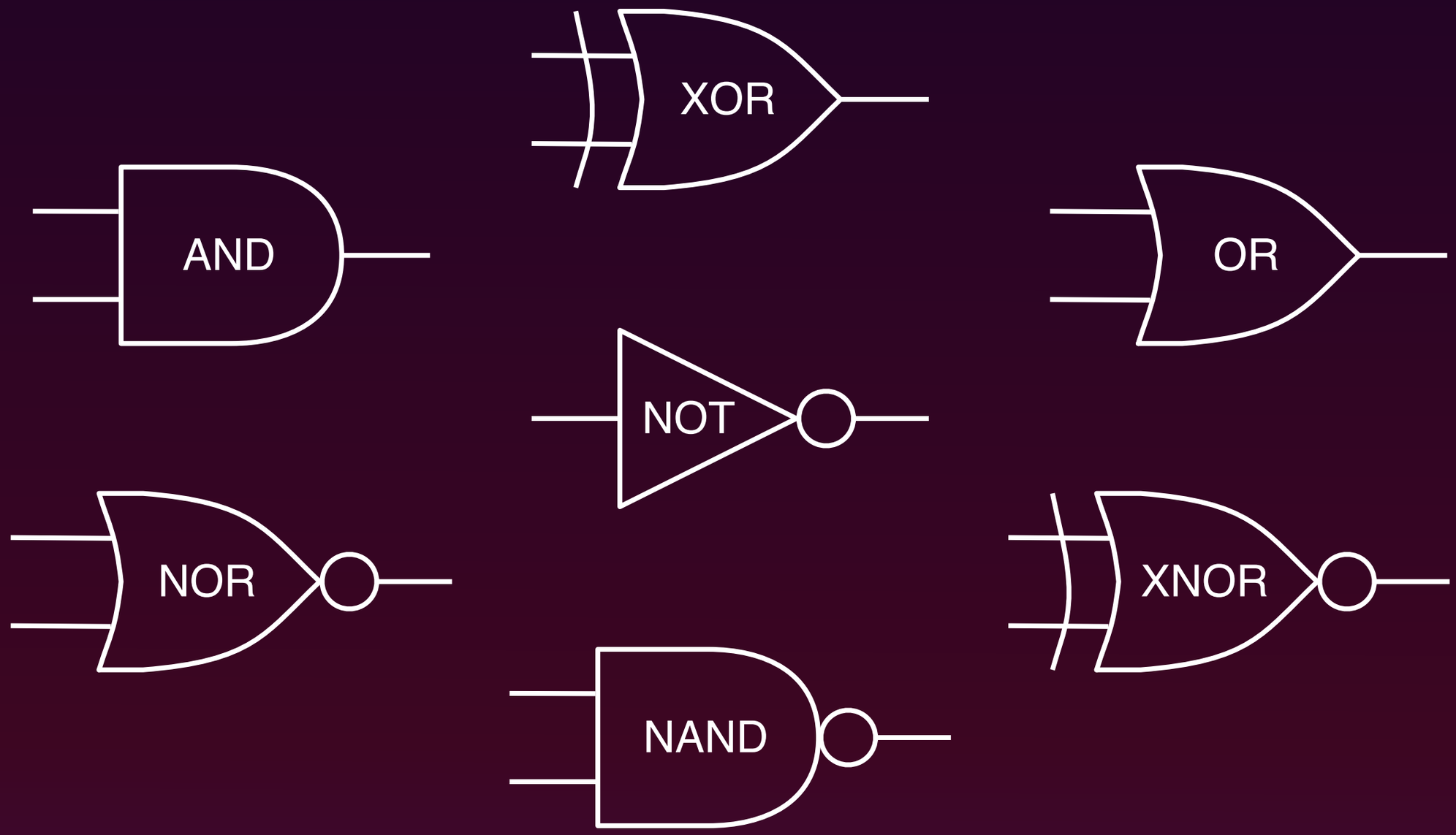
What’s inside the machine?

- Von Neumann architecture - Input, output, processing, and memory
- Motherboard
- CPU
- Memory
- I/O devices

Binary

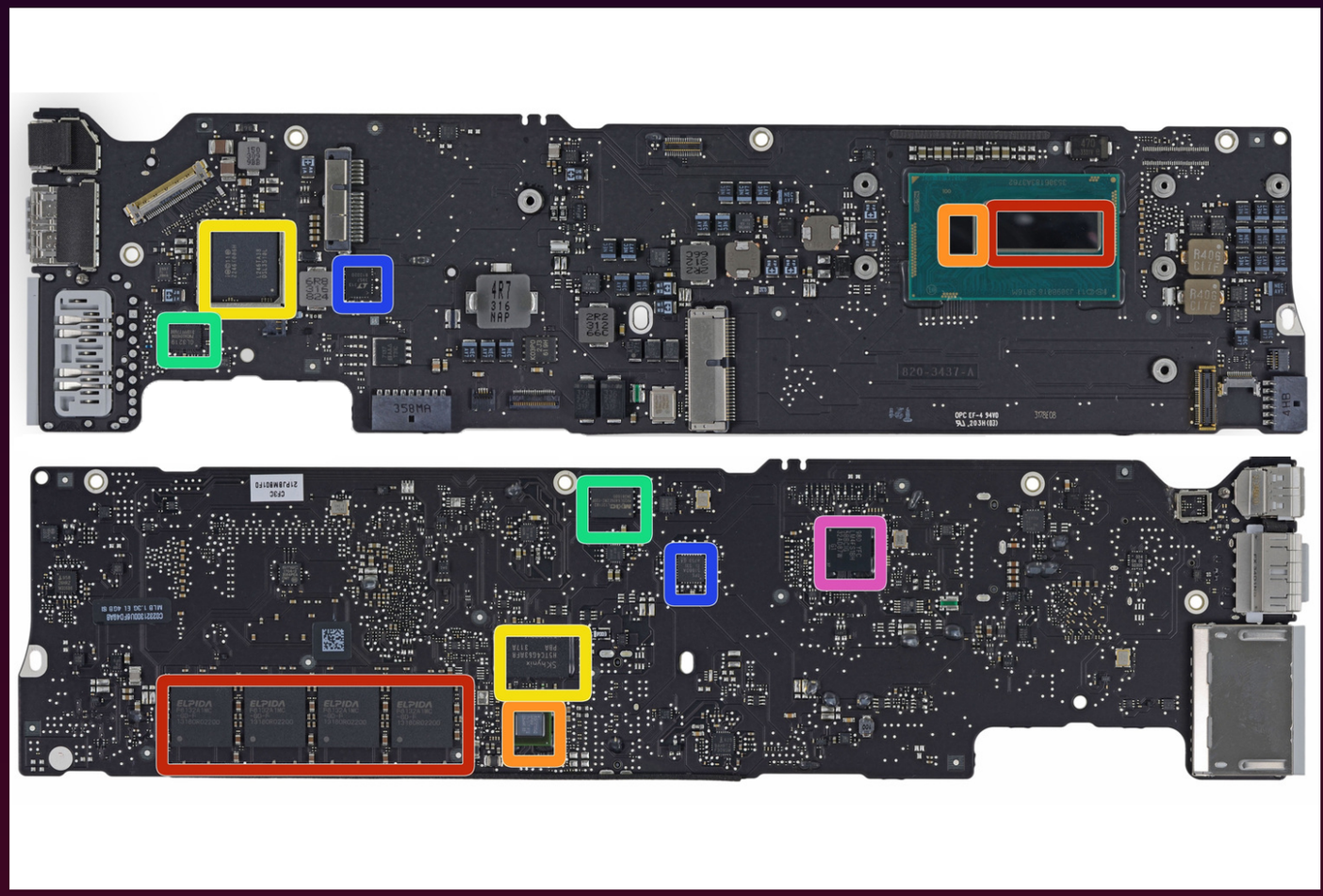
- Base 2 math - 0 and 1
- Binary arithmetic has the same rules as decimal
- Electronic logic is binary
- Morse Code is a binary language

Logic and Gates



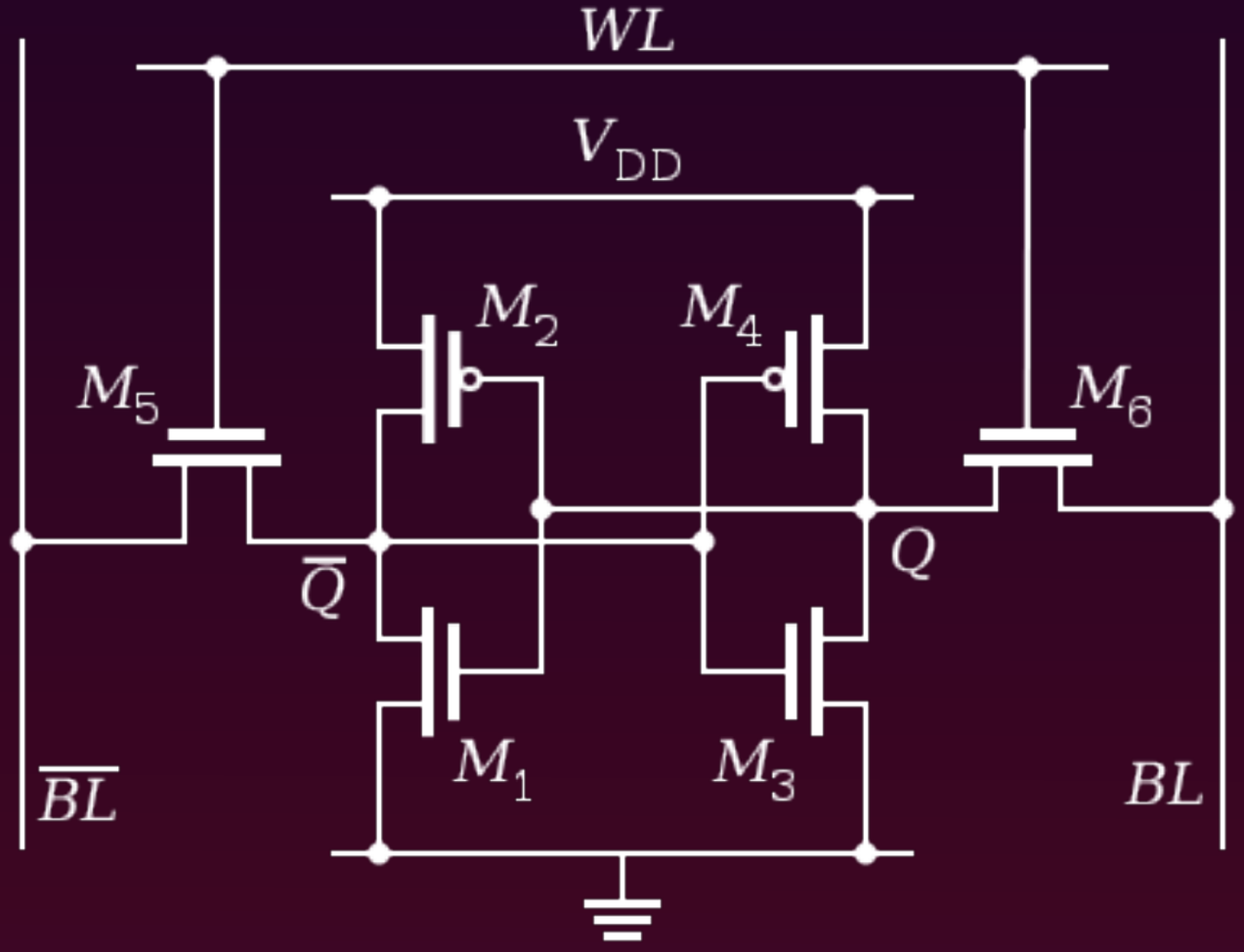
System Components

- Motherboard
- Chipset
- Interconnects
- Peripherals
- Memory, and lots of it!



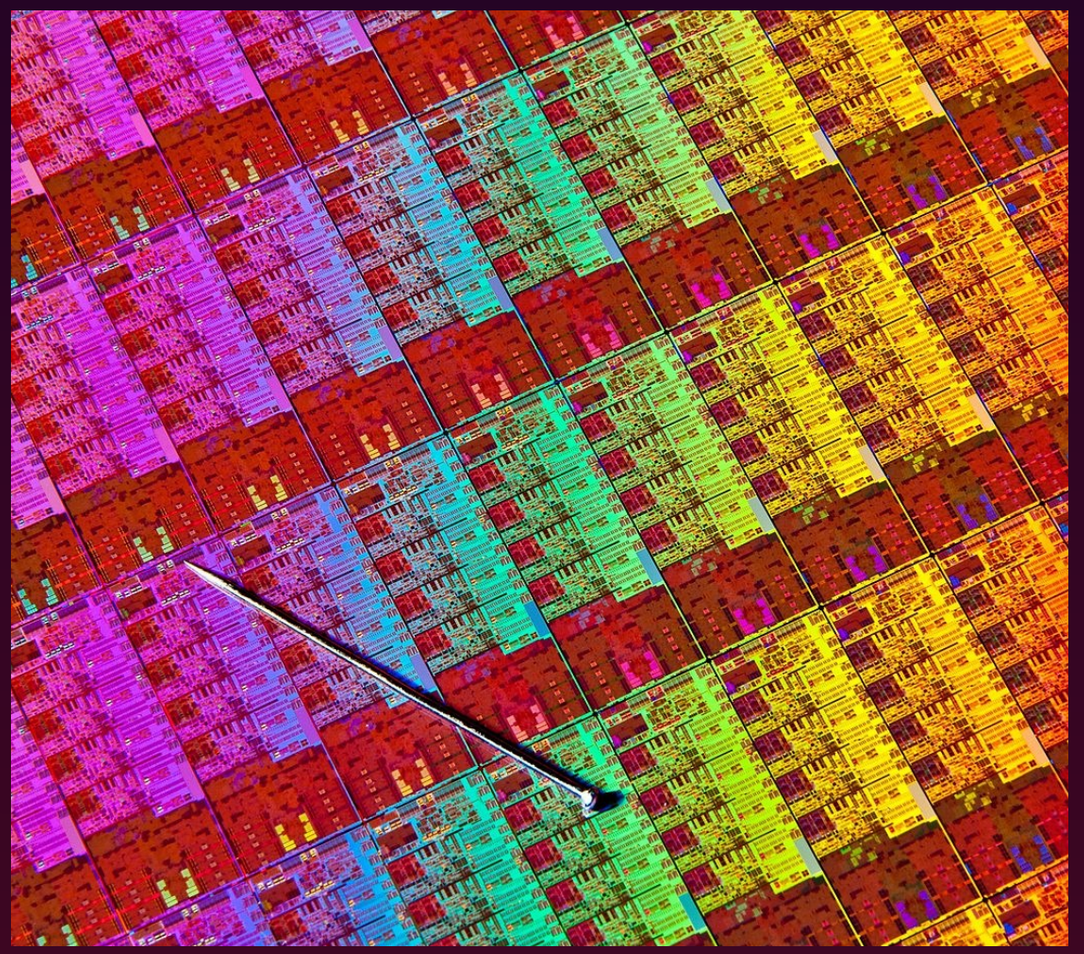
Memory - The essence of state

- Uses for memory
  - CPU registers
  - ROM
  - RAM
  - Caches
  - VRAM



Central Processing Unit

- Executes sequences of instructions
- Architectures define how a CPU operates
- All processing is ultimately simple



Assembly Language

```c
// C
int main(int argc, char **argv) {
    puts("Hello, world!\n");
}
```

```x86asm
// x86_64
pushq   %rbp
movq    %rsp, %rbp
leaq    L_str(%rip), %rdi
callq   _puts
xorq    %rax, %rax
popq    %rbp
ret
L_str:
db      "Hello, world!",10,0
```

```arm64
// arm64
stp   fp, lr, [sp, #-16]!
add   fp, sp, 0
adrp  x0, L_str@PAGE
add   x0, L_str@PAGEOFF
bl    _puts
movz  w0, #0
add   sp, fp, 0
ldp   fp, lr, [sp], #16
ret   lr
L_str:
.asciiz "Hello, world!\n"
```

Firmware - What lies beneath

- Lowest-level code typically executing on a CPU
- Concerned with details of hardware and booting operating systems
- Open Firmware and UEFI
- Functions as a simple OS

The Kernel - A lonely soul

- Takes over from firmware
- Controls hardware resources and manages memory
- Isolates developers from the low-level details of the system