

CS 4410 Operating Systems

Prof. Mike George

<http://courses.cs.cornell.edu/cs4410/2018su>

- What is OS?
- Why study OS?
- Logistics
- Soviet-era keyboard exercise

What is OS?

- Abstraction layer
- sits on top of hardware

- resource management
 - memory
 - CPU
 - ↳ caches
 - ↳ devices ...

- "applications" (isolate each program from other)

uniformity

- no need to know details

virtualization
(illusion of total control)

Components/Outline

- Devices (I/O, keyboard, screen ...)
 - interaction between applications / threads
 • writing correct multithreaded programs
 - memory.
 - filesystems / disk
 - networking
 - security
- } scheduling.

Why take OS?

- Studying software designs
 - design by example.
- Abstractions are leaky.
 - e.g. performance
 - need to know how abstractions are implemented.
- Need to build OS?

Logistics

- Weekly HW
 - due Mondays before class. (out on Mondays)
- Weekly quiz (end of class on Mon).
- Final exam. (in-class, last day).

Academic integrity

- o Do your own work
 - don't share solutions
 - don't copy solutions
 - don't look up solutions to similar problems.
 - don't copy my course materials.

Build a keyboard

- gates (and/or/etc...)

- switches for keys

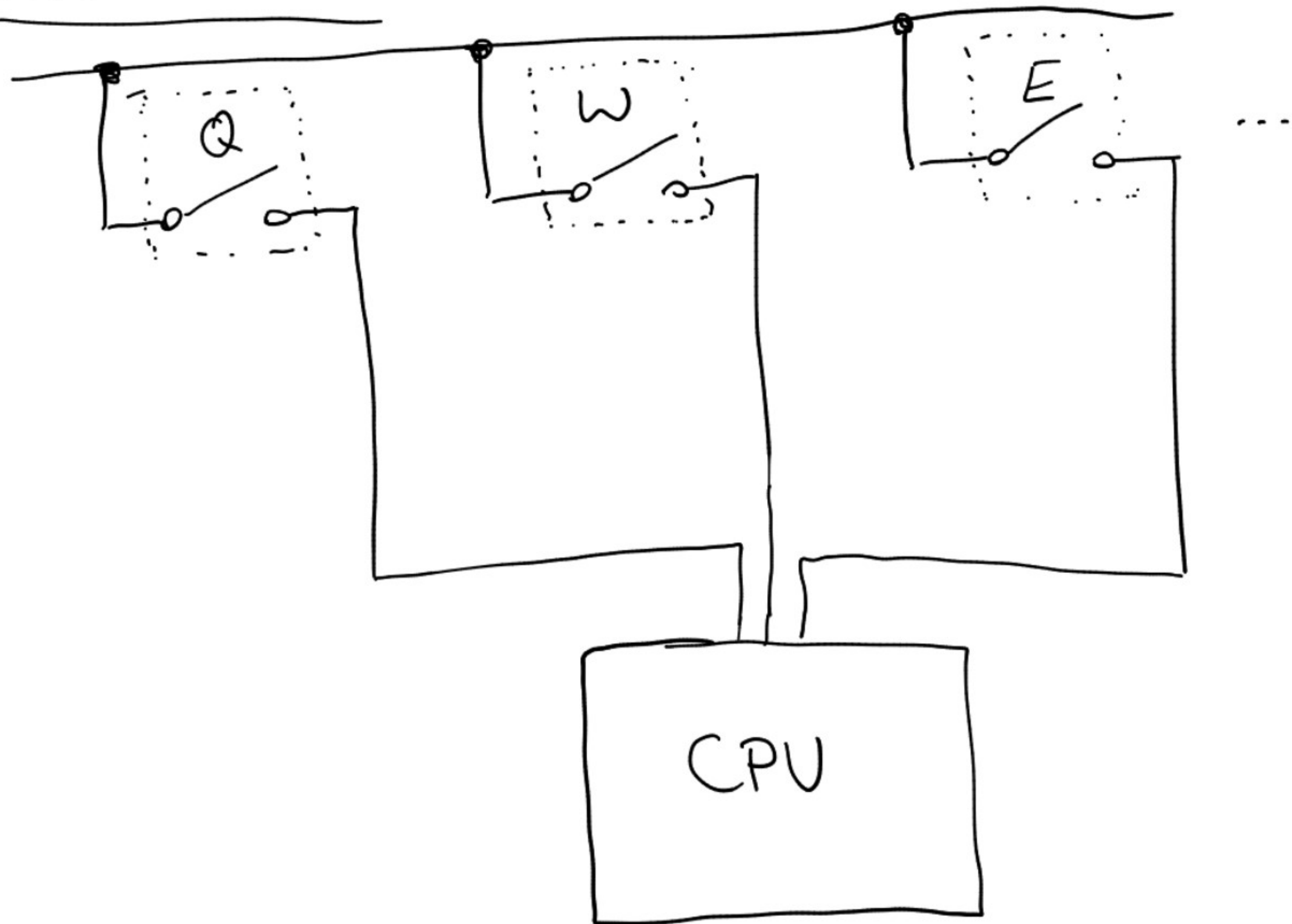


- encoders, muxors, latches, ...

- tri-state buffer

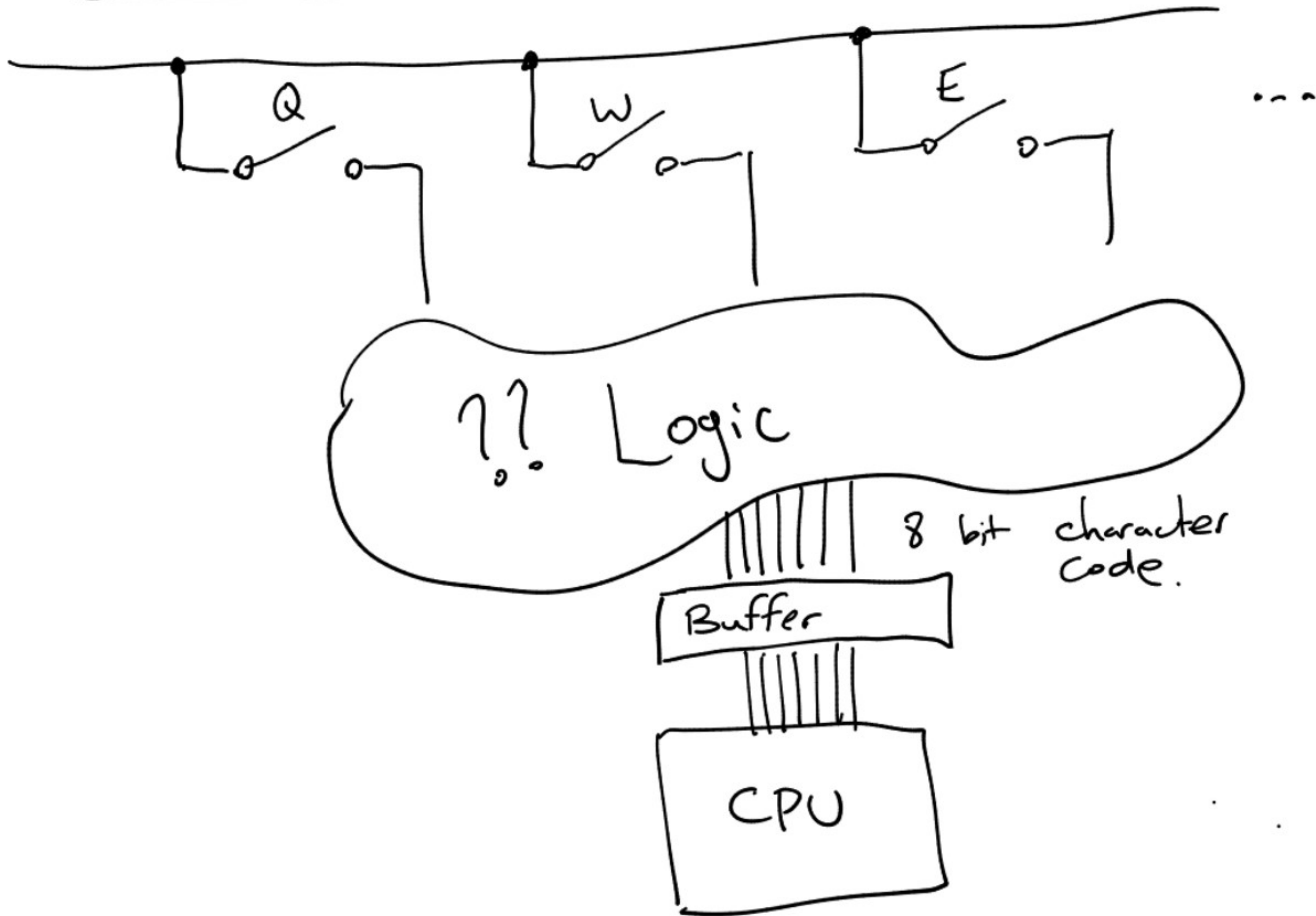


Strawman 1



Easy! just add
104 pins to CPU
- Cons: lots of
soldering.

Strawman 2

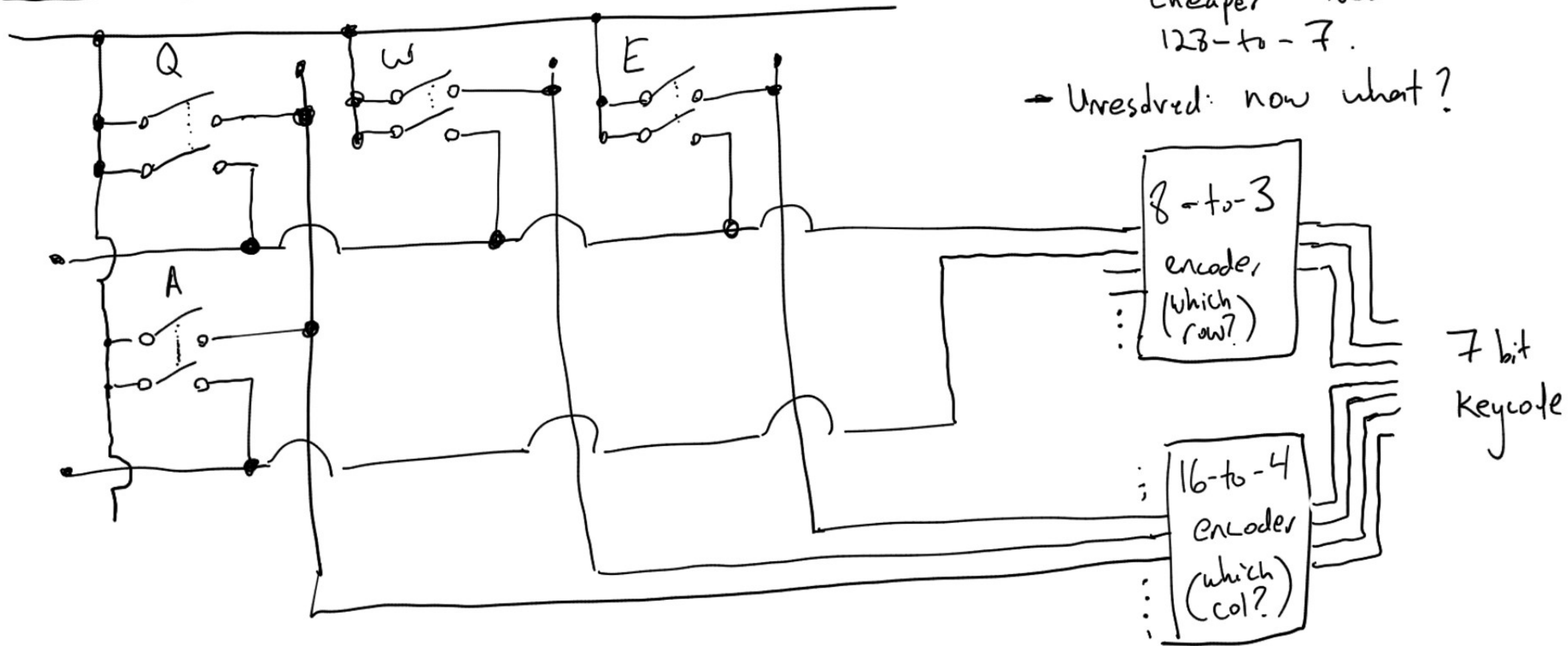


Pros: less soldering

Cons: - still soldering.
- "logic": 104-to-7 encoder, kinda expensive

Unresolved:
- how to interact w/ program?

Strawman 3

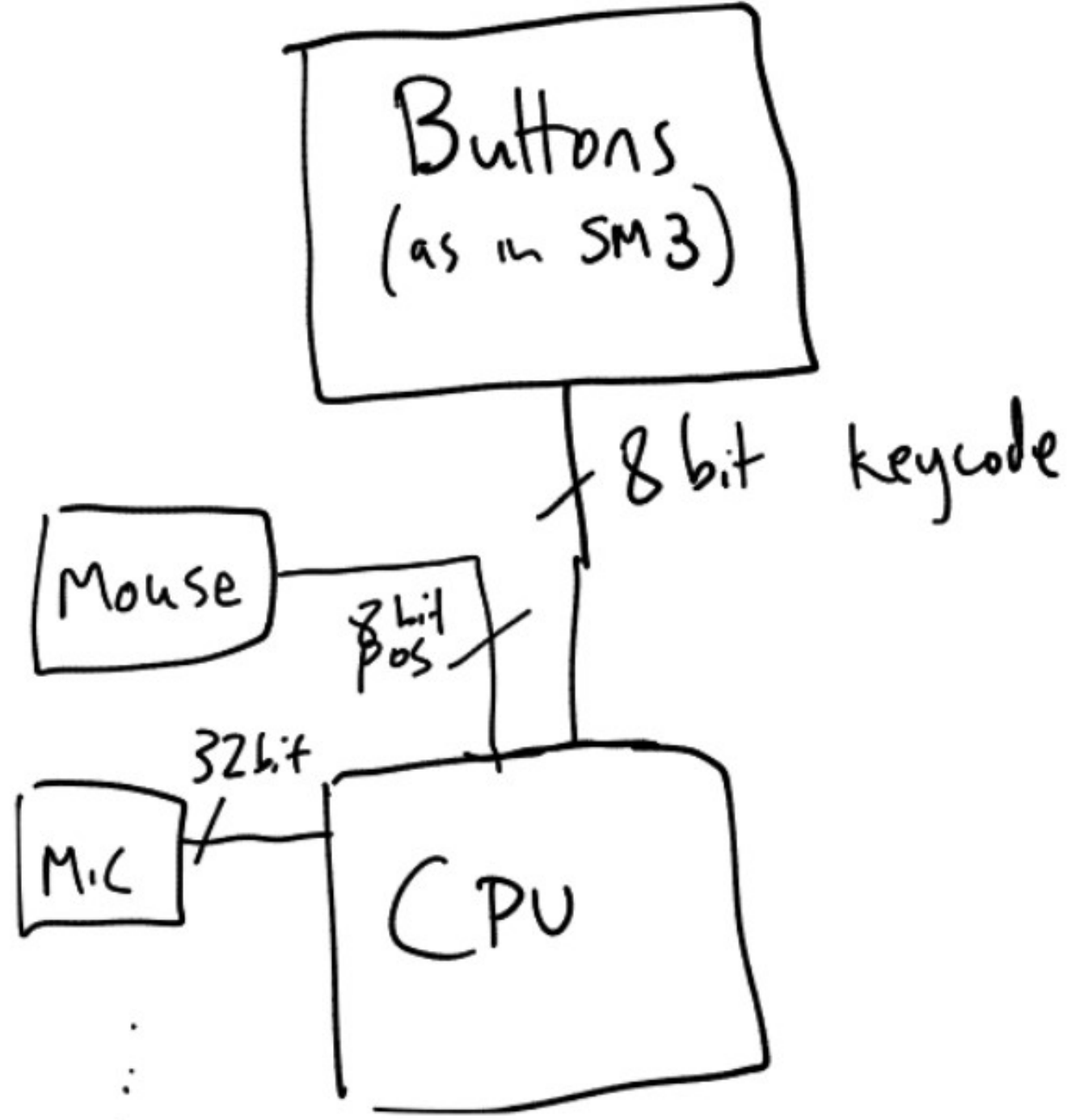


→ Pro: 8-to-3 + 16-to-4 much cheaper than 128-to-7.

→ Unresolved: now what?

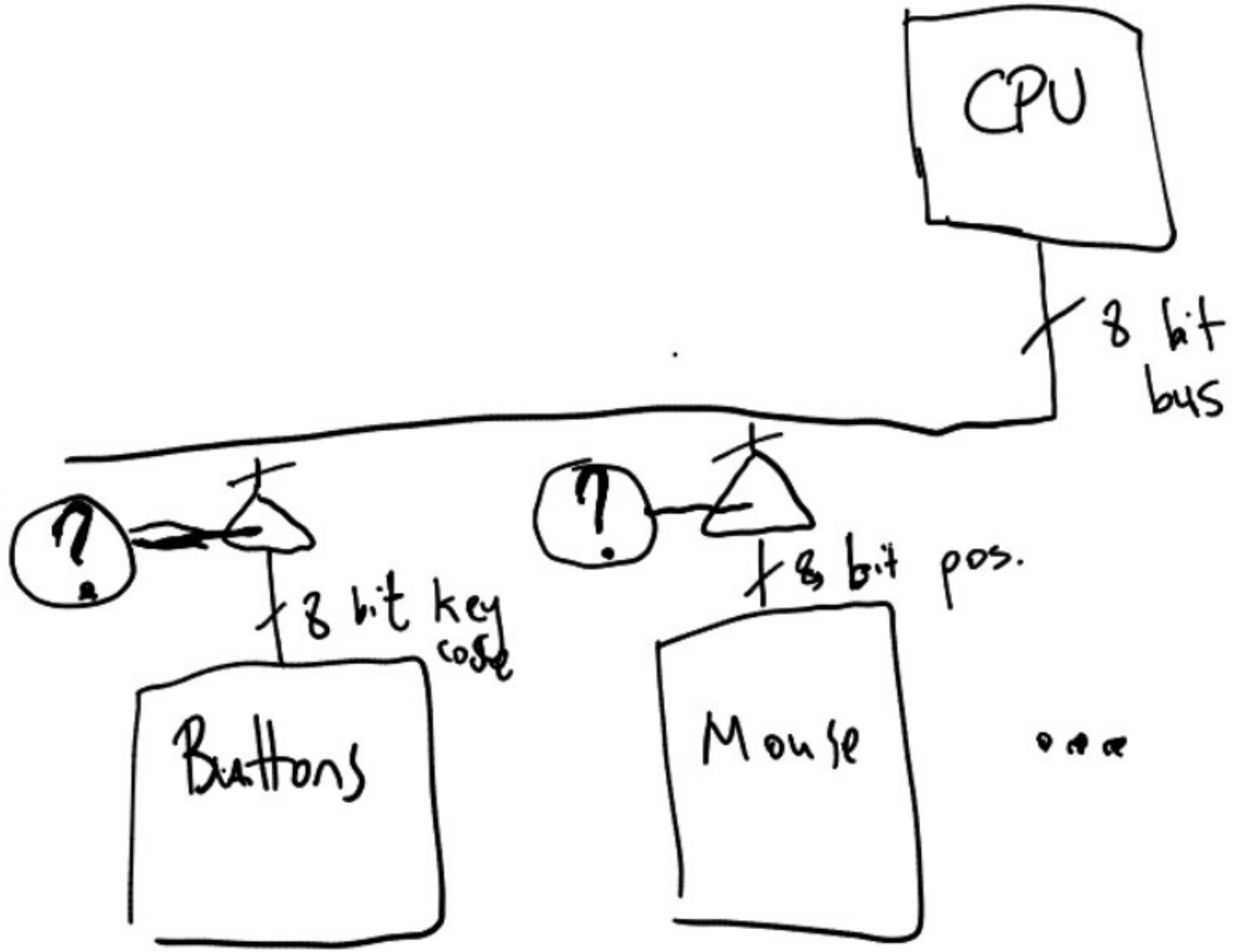
7 bit
keycode

Strawman 4



Con: Soldering galore!

Strawman 5



Unresolved:

- ① where does control ? come from
- ② what happens to keystrokes while disconnected?
- ③ what does "hello world" program look like?