6.S188

Build a Digital Clock from the Eighties

Lecture 3A:
Clocks and Oscillators

Couple Remaining Things

- This will be our last lecture/recitation/thing
- After this one more optional lab (Hit the Bit)
- And then build your digital clock

Oscillators

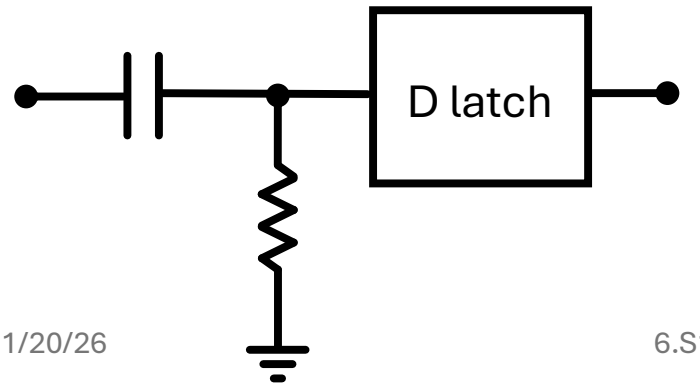
How do you oscillate?

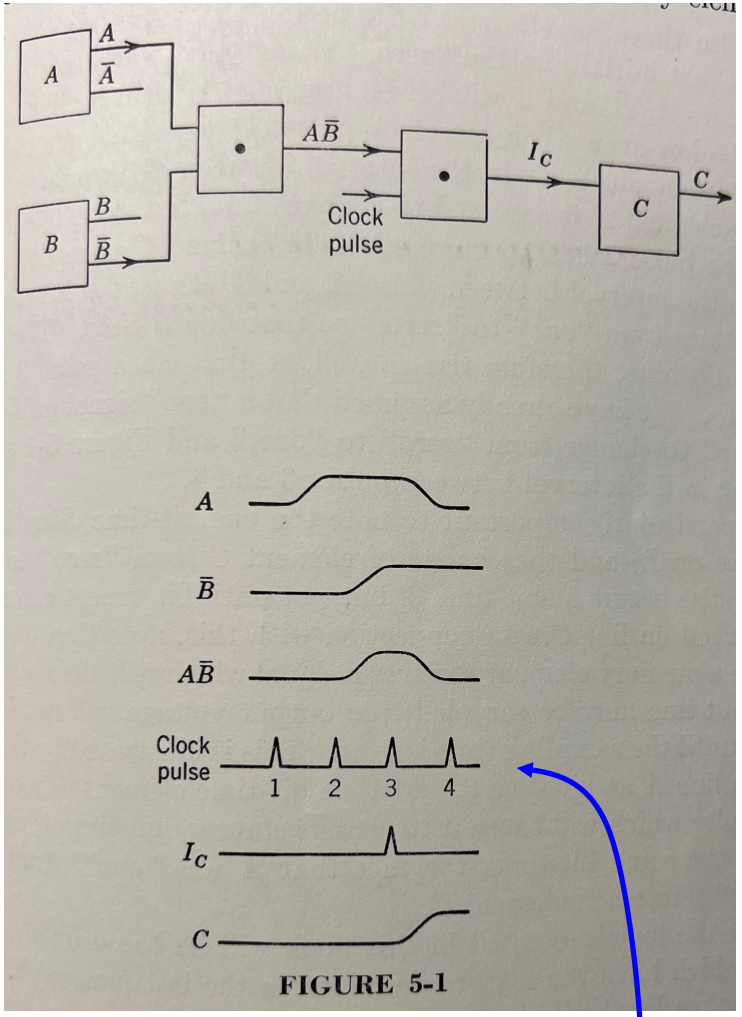
A Big Missing Piece…

- We’ve basically covered most of digital logic with the exception of clock generation.
- Most digital systems work operate using periodic signals to synchronize all the logic together.
- These signals came to be known as “clock signals”

Originally…

- The pulses were just kinda poorly regulated RC-pulses.
- A lot of early digital circuits had capacitors on input for this reason.
- The RC circuit Adam did last week to get an “edge” was actually how they’d do it.

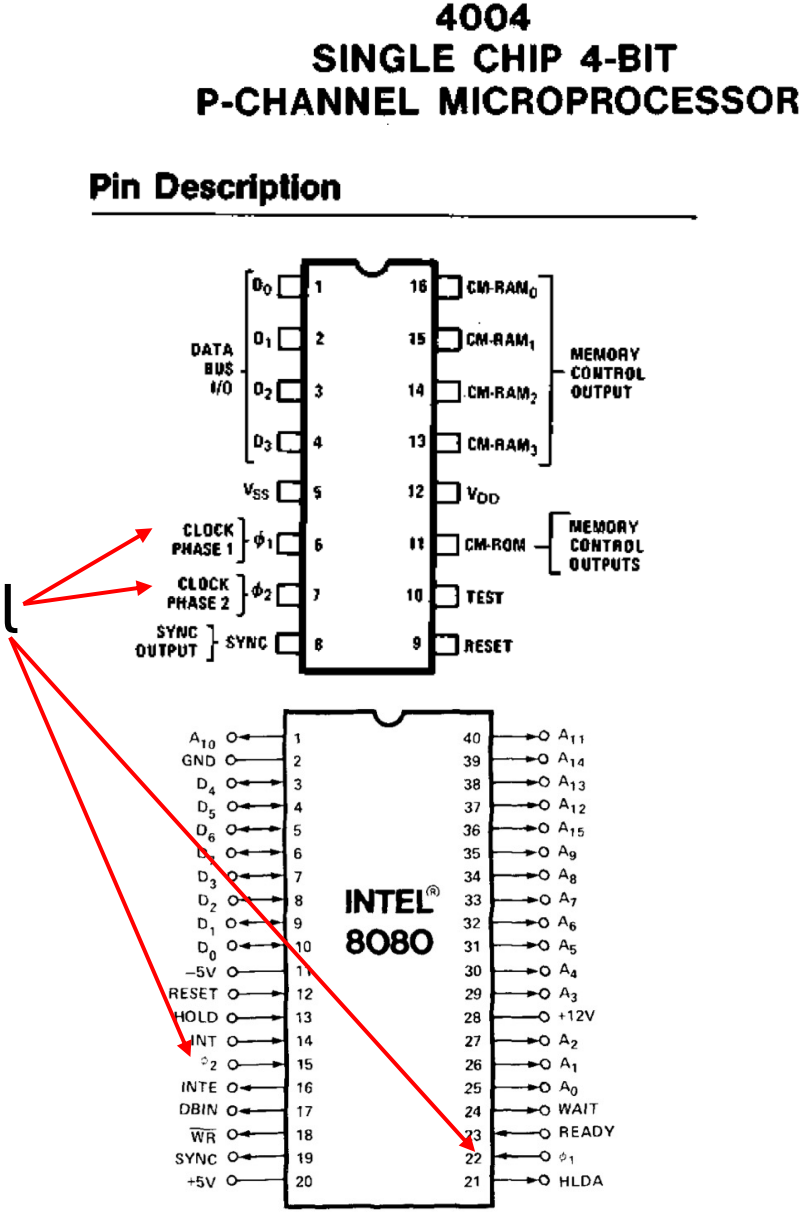




Digital Design book from 1959. A very non-square-wave-looking clock signal

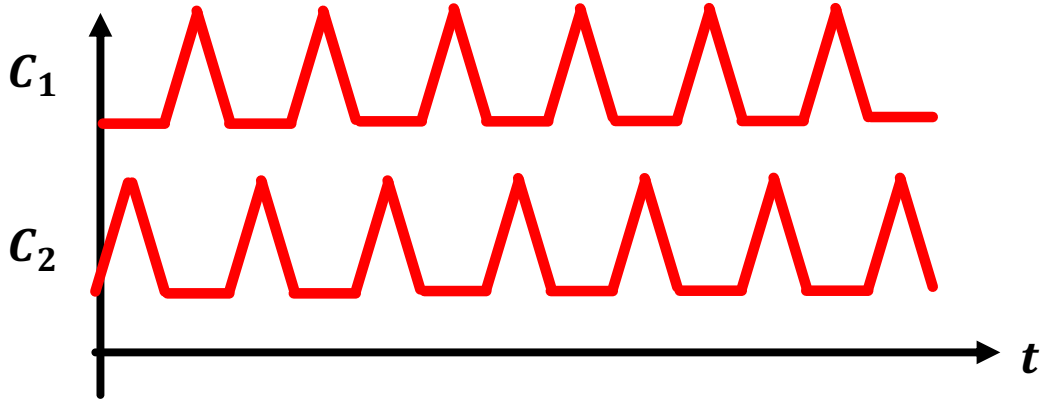
Hard to Scale

- As designs got more complicated, in order to pipeline things more effectively more
- Many systems would need more than one periodic signal

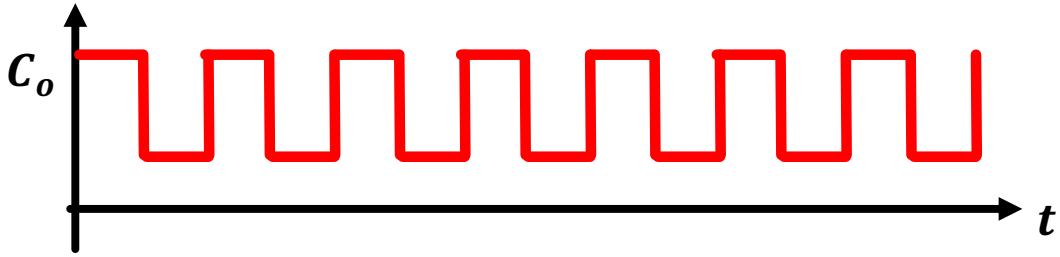


Just Make Nice Periodic signals

- Instead of having mess of offset pulses that were hard to maintain:



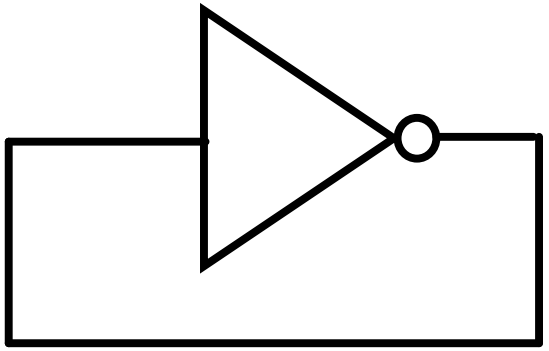
- Prioritize clean square-waves:



So how to make this?

Single Inverter in Direct Feedback

- Connect the output of an inverter back to its input…

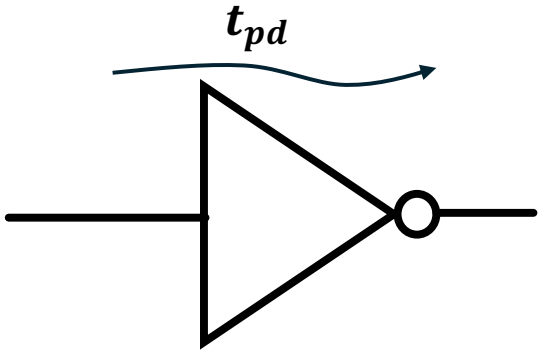


- What does it do?

Input	Output
0	1
1	0

Single Inverter in Direct Feedback

- Logic gates don't transition immediately.
- Everything needs time.
- The value t_{pd} indicates how long a logic gate should take to respond to an input

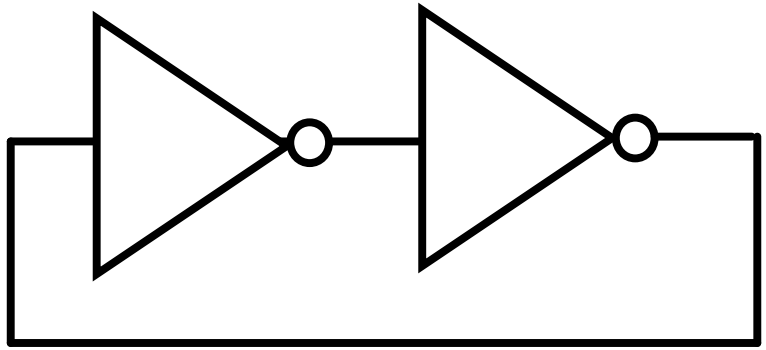


t_{pd}

Input	Output
0	1
1	0

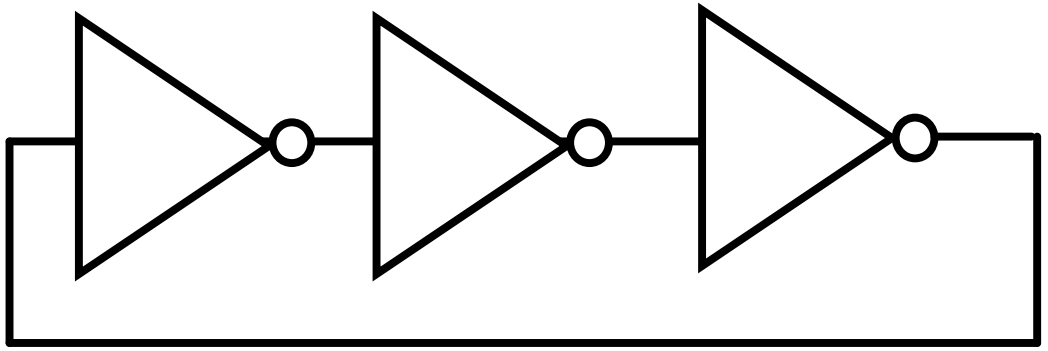
Two Inverters in Direct Feedback

- What will this do?



Three Inverters in Direct Feedback

- What will this do?

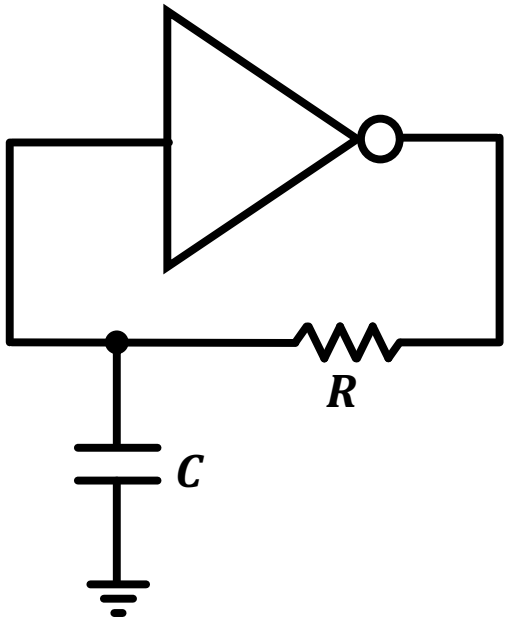


In reality…

- Odd chains of inverters will tend to oscillate (***with the exception of one inverter which will usually mush out into analog existence***)
- Call these ring oscillators
- Even chains of inverters will find stable operating points
- Ring oscillators are highly unstable…unless you need sources of randomness they are very randomly used.

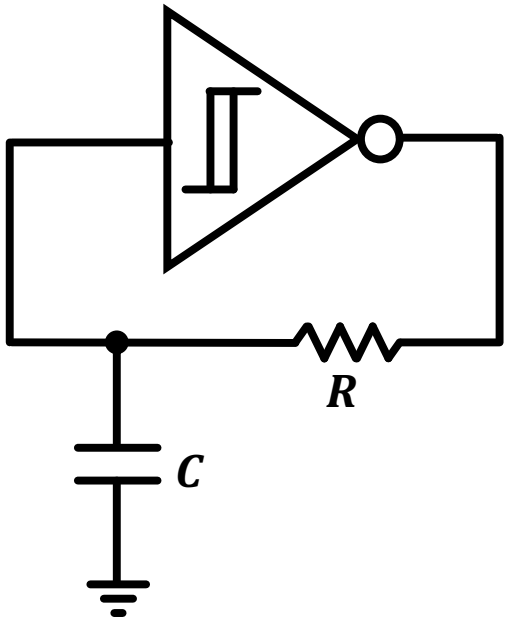
One Way to Kinda Fix…

- Add an external source of timing…
- Adding a sizable RC circuit can slow down the ability of the NOT's output to drive its input.
- Still isn't perfect though



This is better…

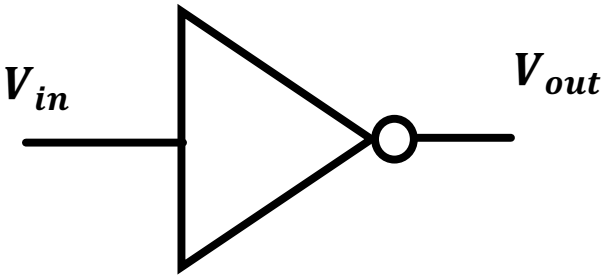
- Add in the Schmitt Trigger…
- The Schmitt Trigger's hysteresis means it will

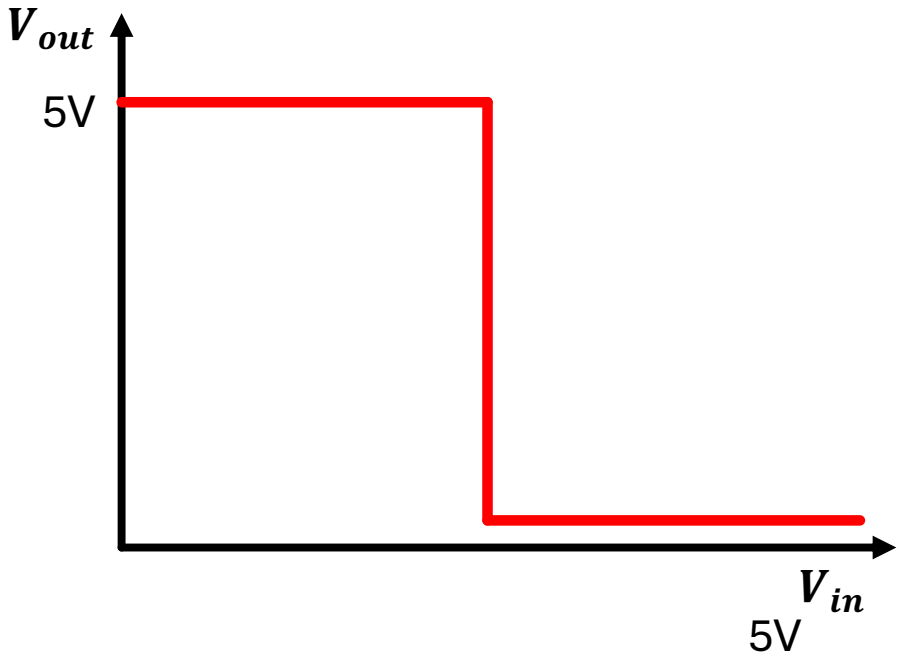


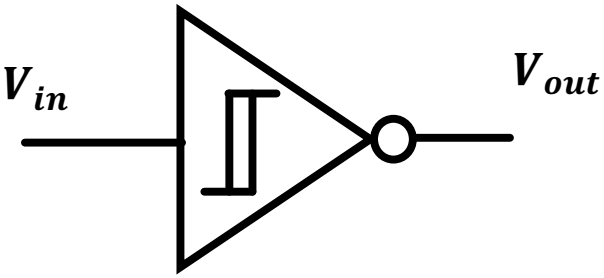
Hysteresis

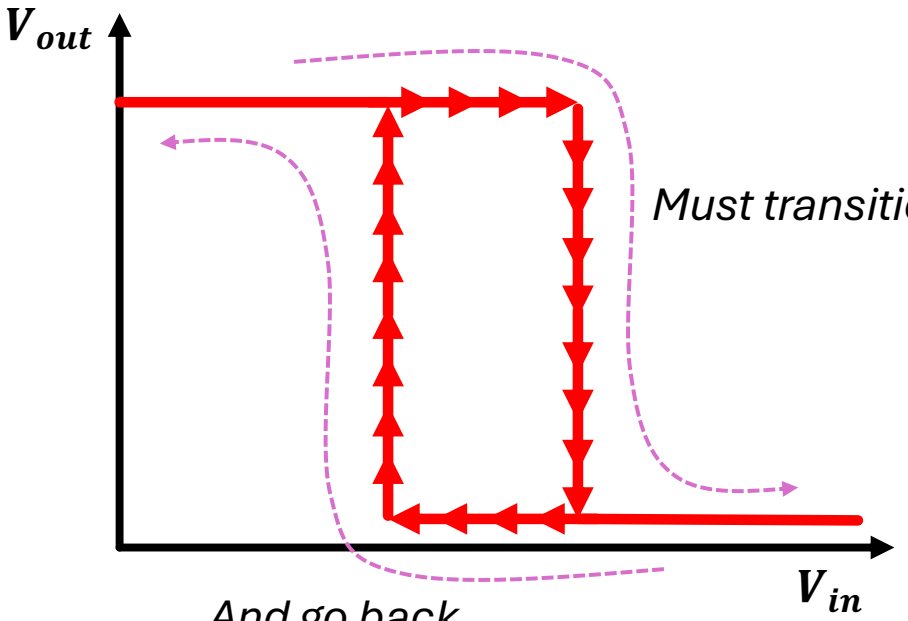
- Once you start going, you can’t go back (positive feedback and some other internal stuff)
- Makes a circuit much less prone to noise
- Also great at cleaning up signals and converting analog stuff to digital stuff.

Regular Inverter



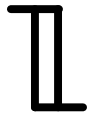






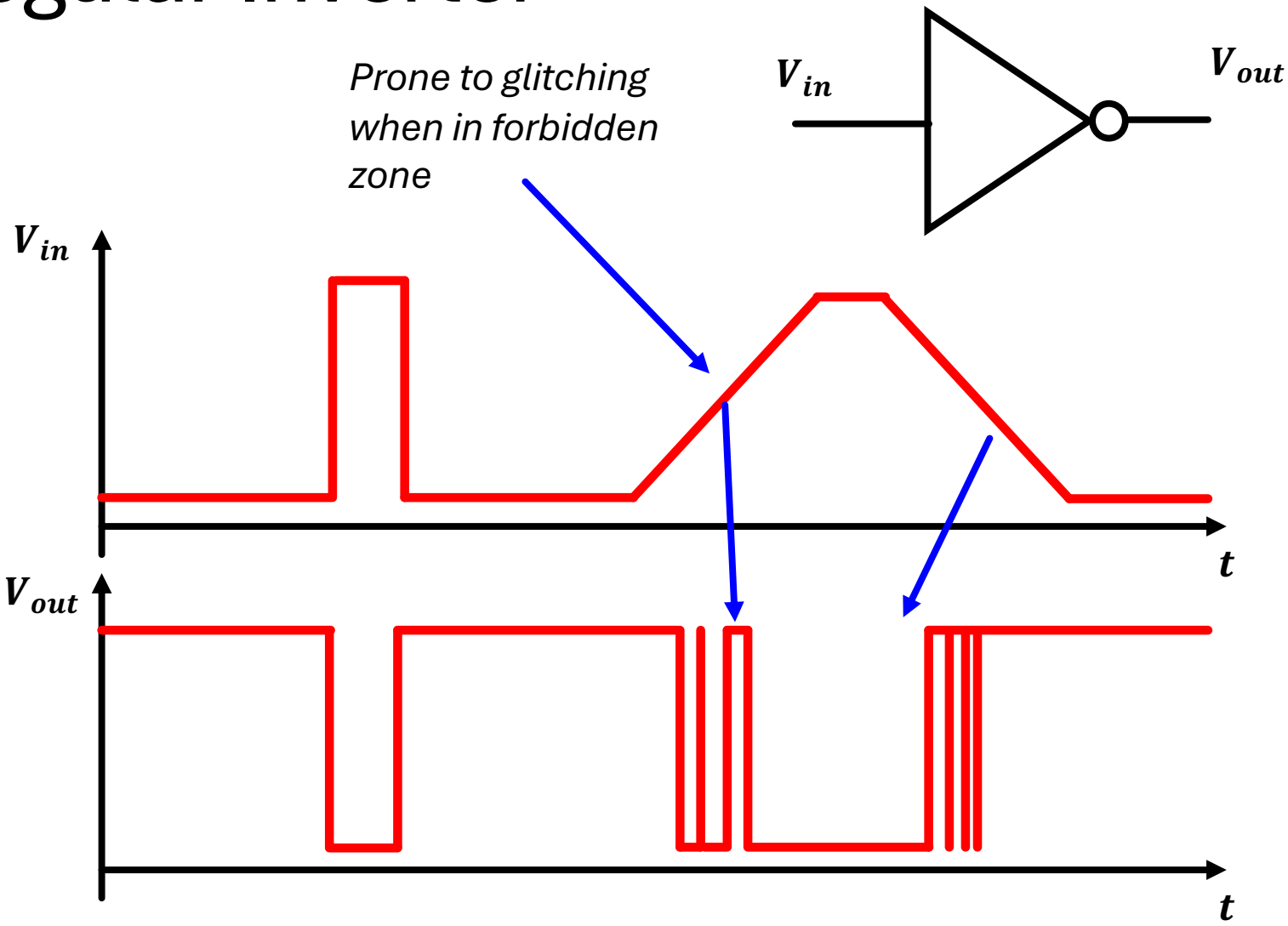
Must transition in this way

And go back this way

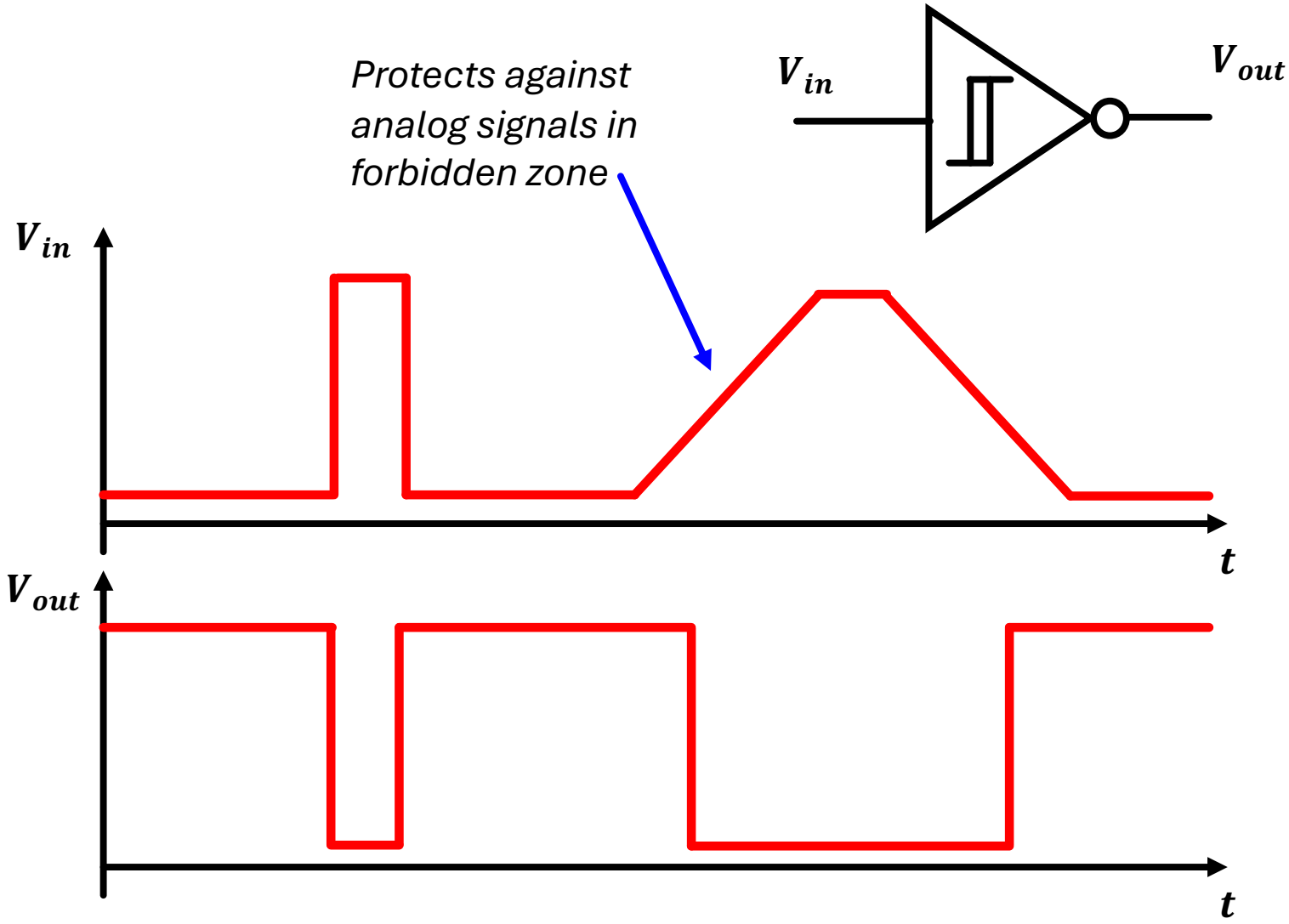


Sometimes drawn this way too. Symbol is graphical representation of hysteresis curve, not fancy S

Regular Inverter



Schmitt Inverter



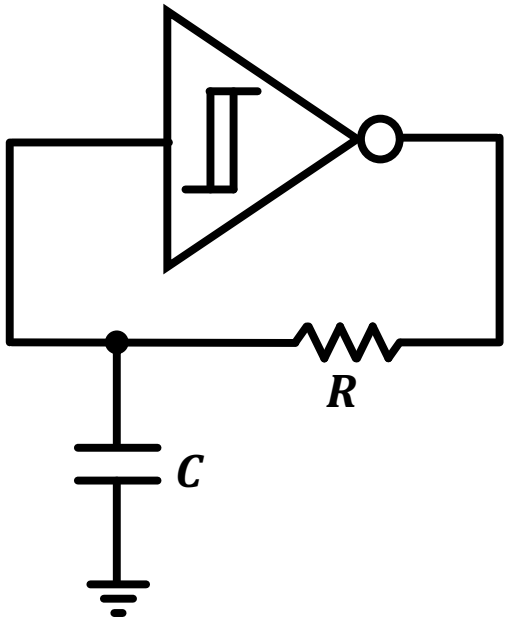
Aside…Biomimicry

- “Schmitt Triggers” receive their name from Otto Schmitt who came up with the idea for them after studying

- When our neurons fire there’s actually hysteretic behavior happening that makes things actually work.

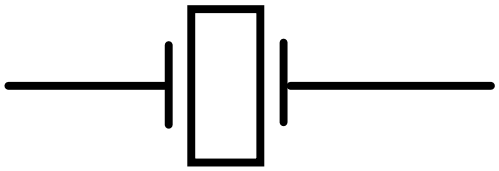
Even this has problems

- If you do lab 03a, even with hysteresis, you'll see the period of oscillation will not be stable since R's and C's can drift a bit with temperature and things.
- Not reliable



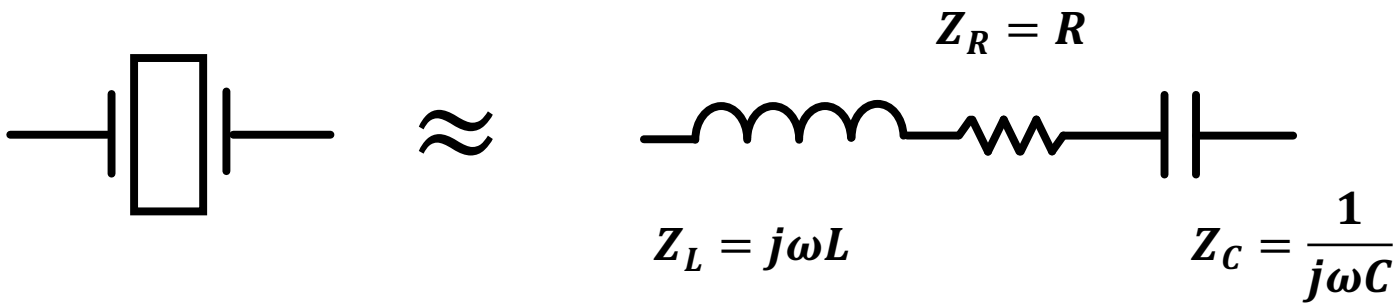
Solution…use crystals

- Take Quartz, cut it into particular shapes and utilize its mechanical resonances as a very high quality and stable frequency reference



https://en.wikipedia.org/wiki/Crystal_oscillator

Crystal is equivalent to LRC circuit*



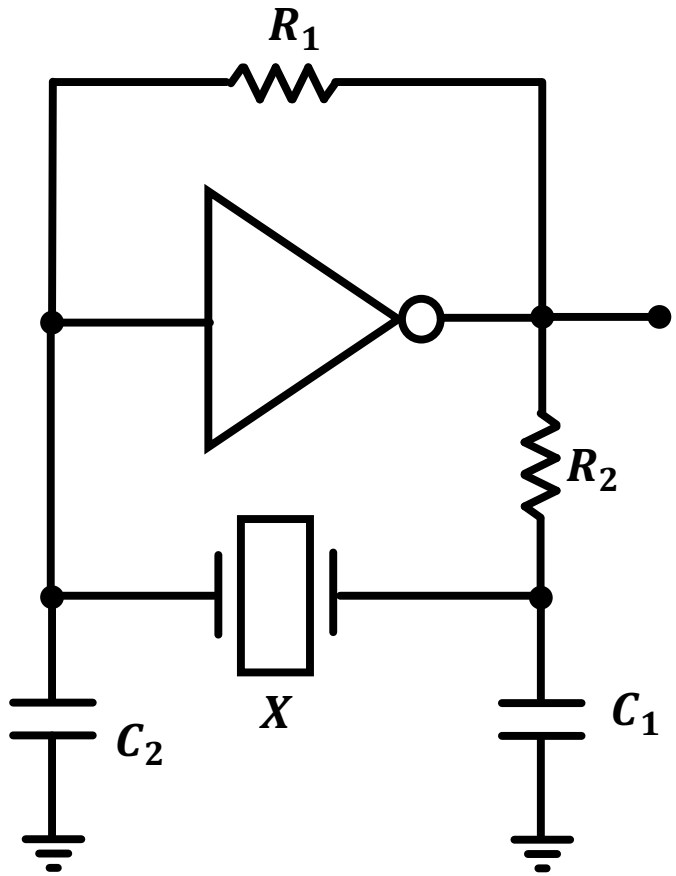
$$f_{res} = \frac{1}{2\pi\sqrt{LC}}$$

Resonant frequency

*and a few other things I'll ignore for now

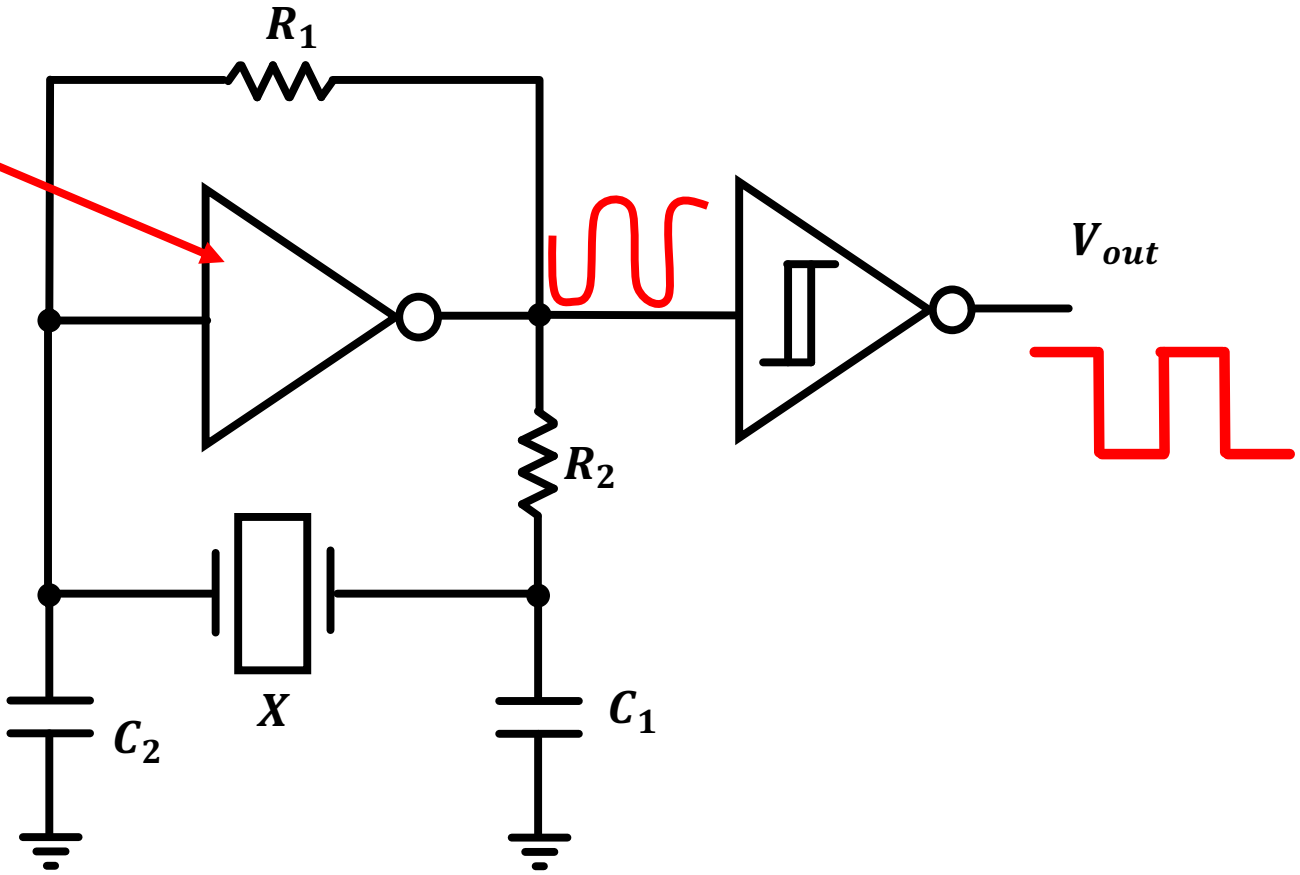
Pierce Oscillator

- All these components form a very stable bandpass filter in feedback at the frequency of interest.
- It will lead to square wave oscillations at the output
- Forms the heart of all modern clock generators!!!



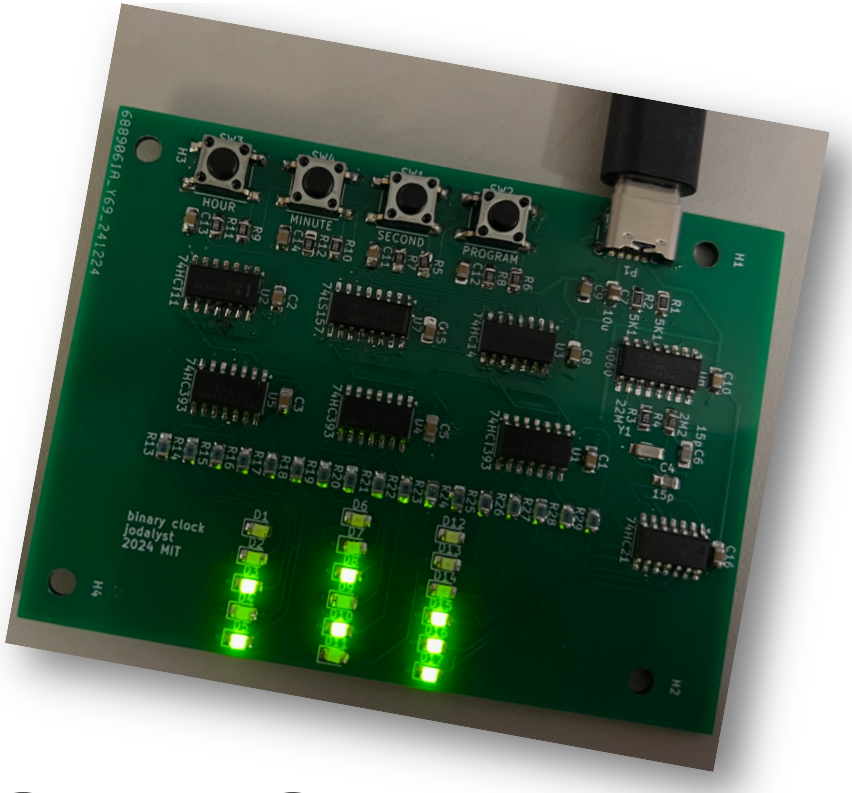
Usually add a cleanup Schmitt as well…

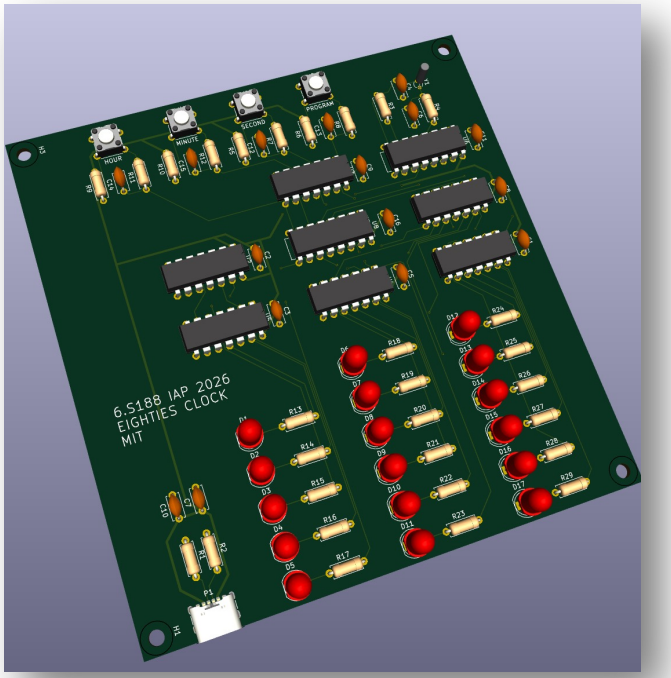
*The inverter here is **kinda** being used as an analog component so its output signals is not a digitally clean signal*



Limits on Crystals

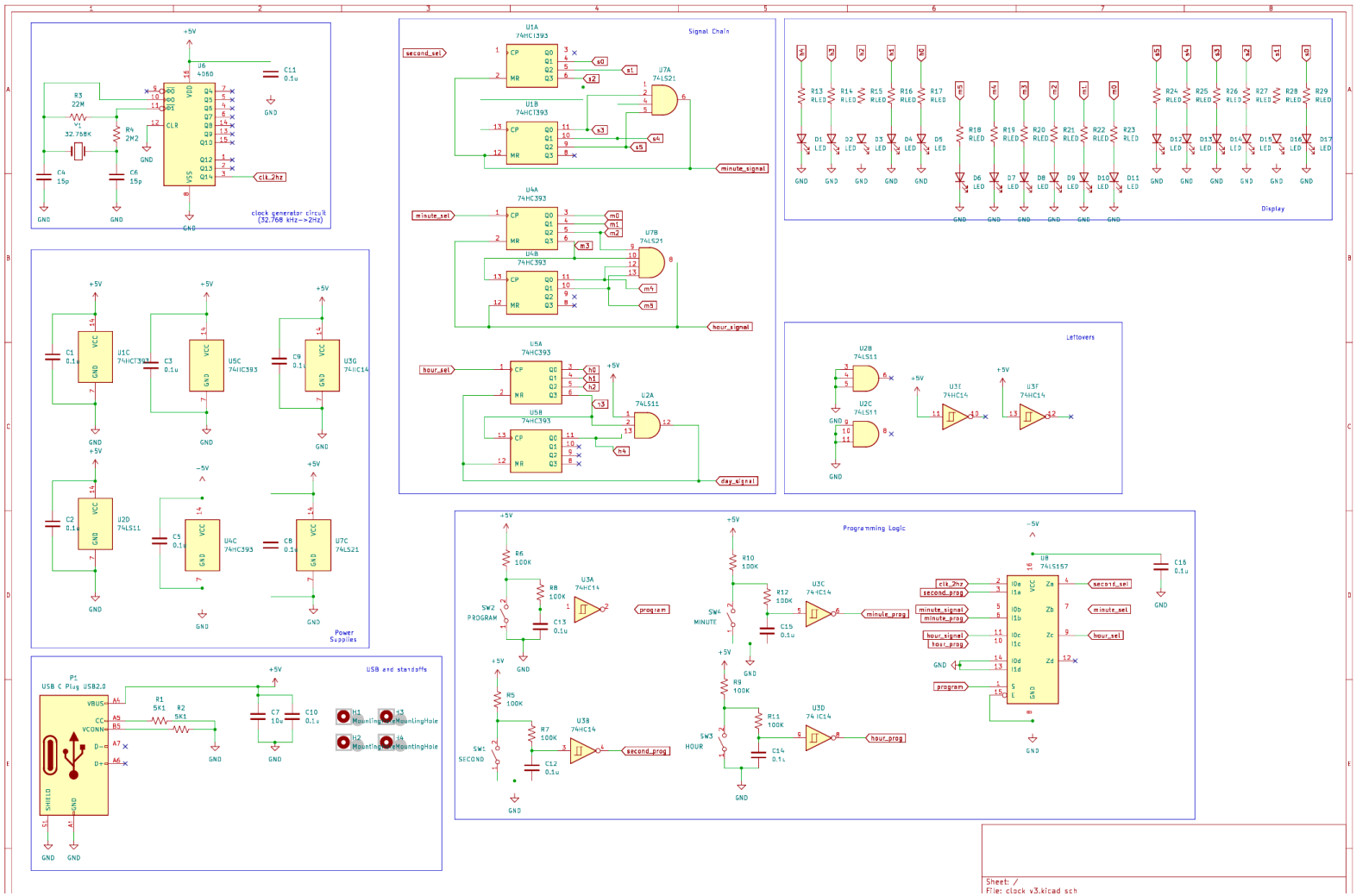
- Very tough to make quartz crystals with low resonant frequencies (get very large)…generally lowest frequency you'll see is ~32 kHz
- Also tough to make really high frequency crystals (~1 GHz about the largest you'll see).
- We use support circuits to derive other clocks from them


HOUR
MINUTE
SECOND
PROGRAM
binary clock
jodalyst
2024 MIT


6.S188 IAP 2026
EIGHTIES CLOCK
MIT

Our Clock

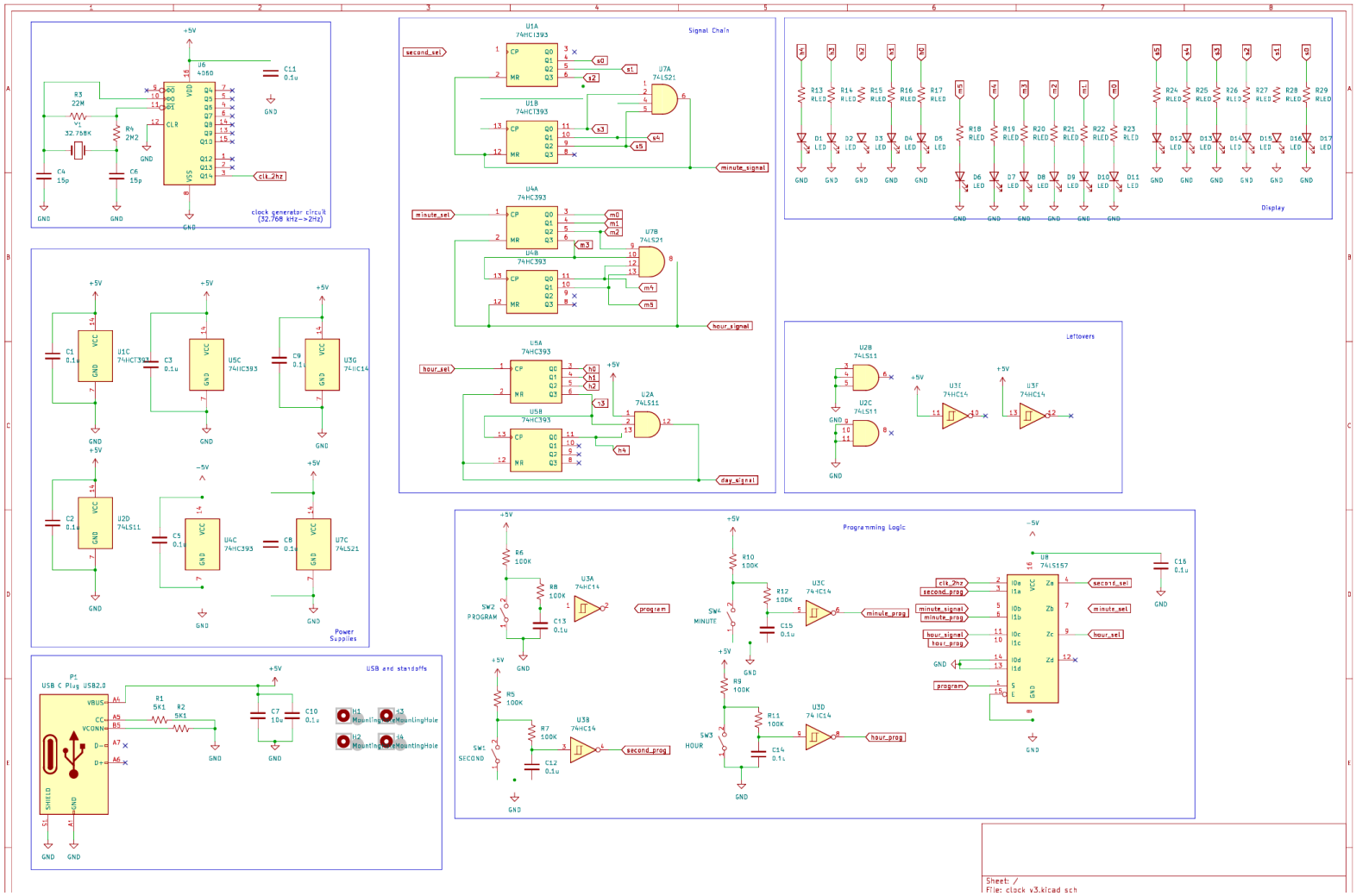
The Clock


clock generator circuit
(32.768 kHz->2Hz)
Signal Chain
Display
Leftovers
Power Supplies
USB and standoffs
Programming Logic
Sheet: /
File: clock v3.kicad sch

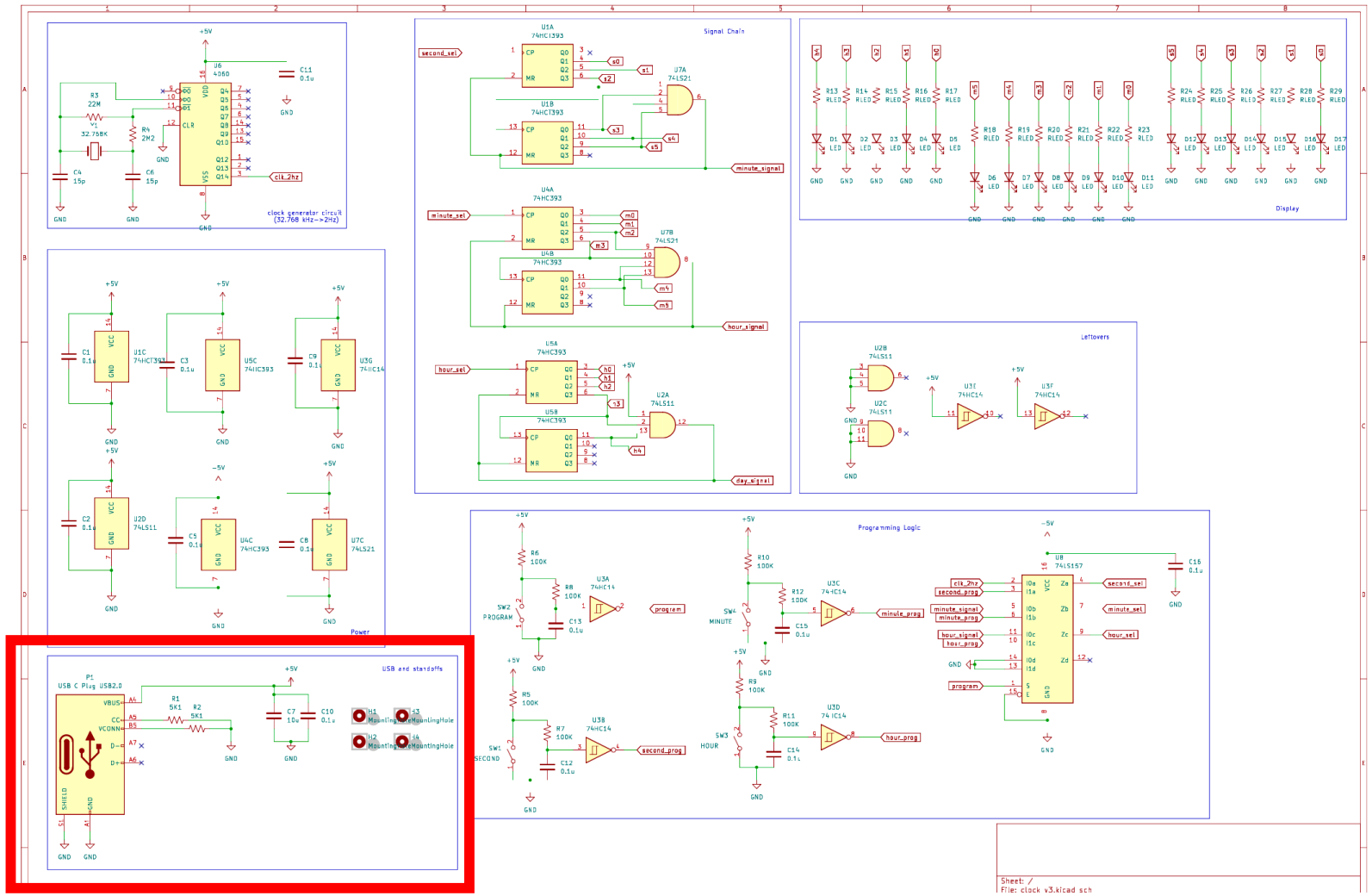
Build Options

- Two pre-made PCBs:
  - One using surface-mount components
  - One using through-hole components
- Breadboard your own clock from scratch using schematic (with modifications as you see fit)
- Solder up your own clock from scratch on perfboard using schematic (with modifications as you see fit)
- Design your own PCB off of starting schematic and we can send out (though likely won't get back prior to end of IAP)

Overview

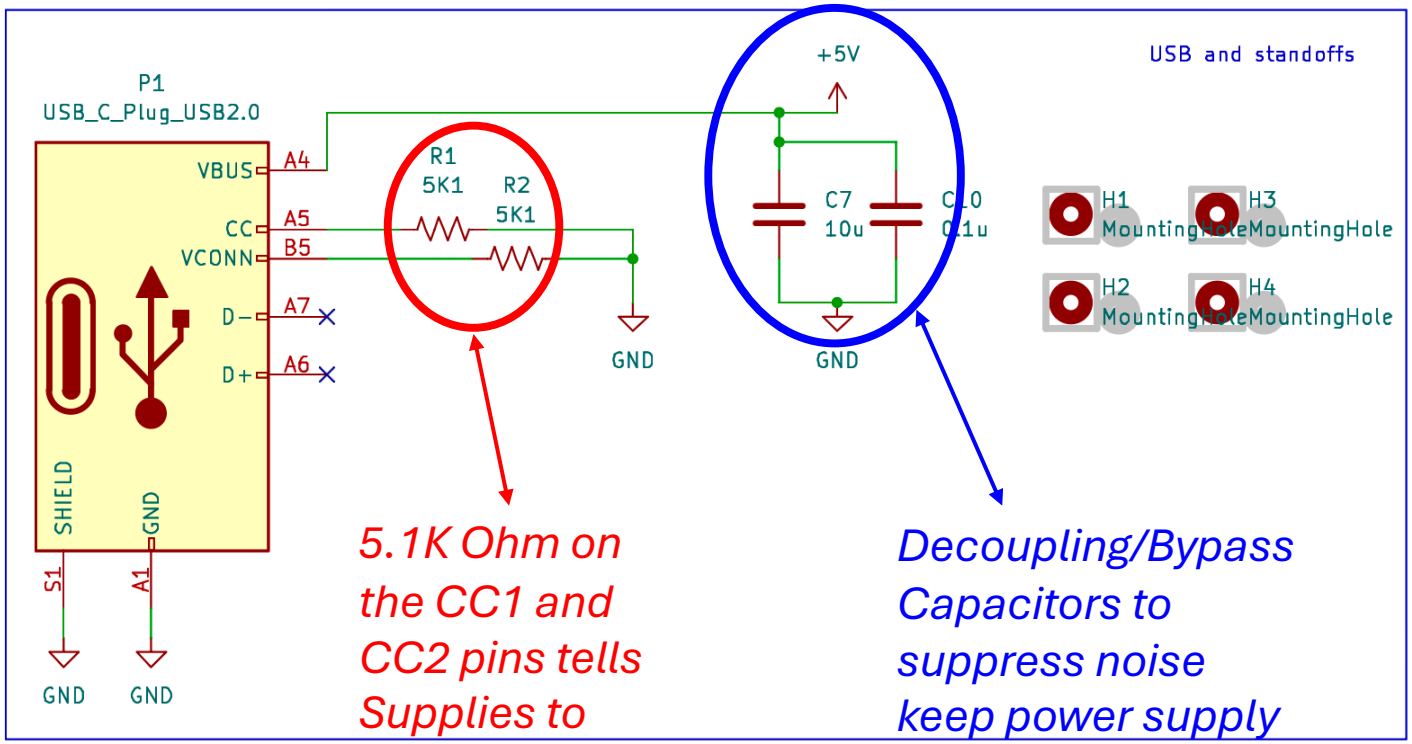

clock generator circuit (32.768 kHz->2Hz)
Signal Chain
Display
Leftovers
Power Supplies
Programming Logic
USB and standoffs
clk_2hz
second_set
minute_set
hour_set
second_signal
minute_signal
hour_signal
day_signal
second_prog
minute_prog
hour_prog
program
Sheet: /
File: clock v3.kicad_sch

USB Supply

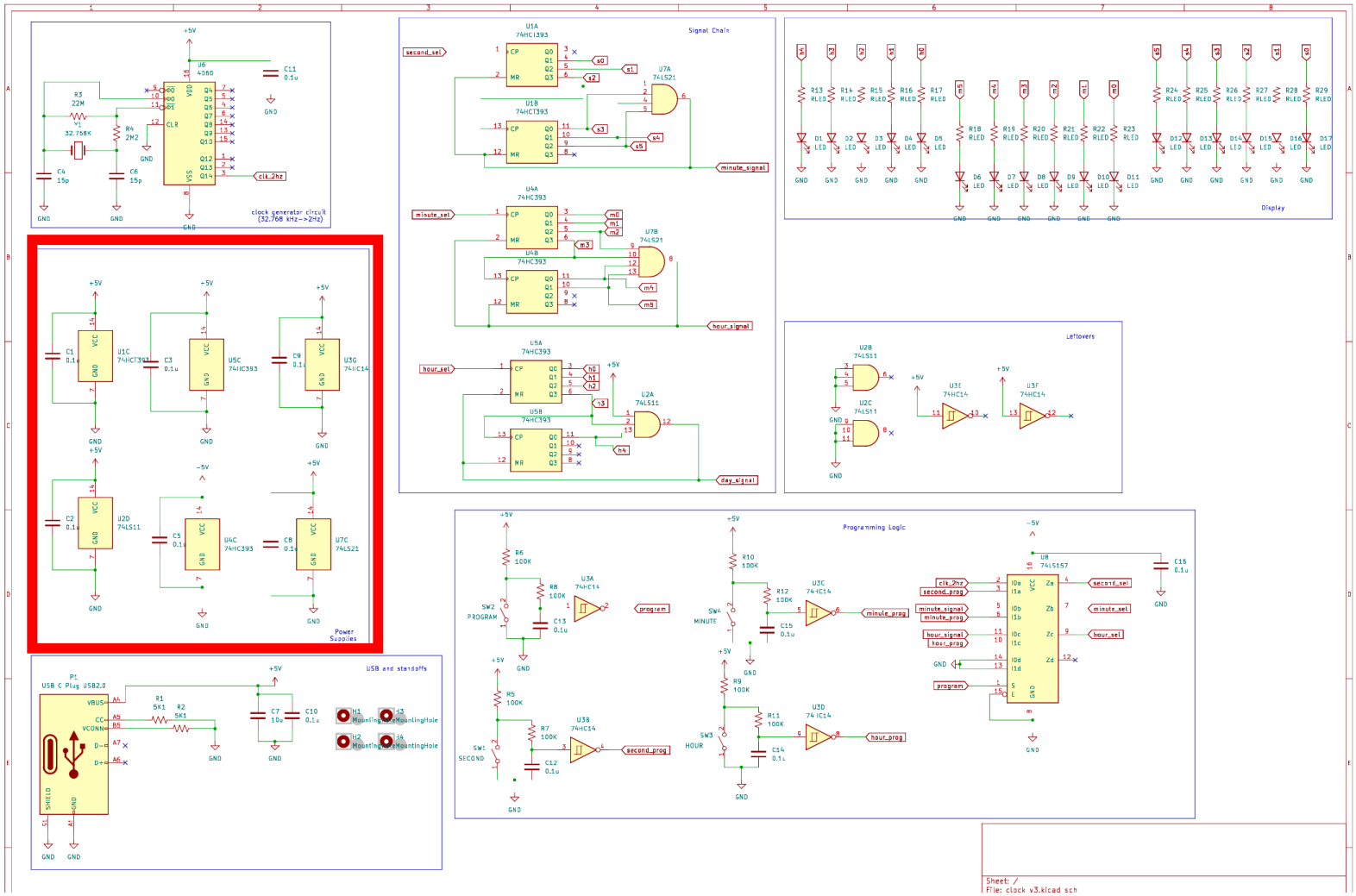

clock generator circuit (32.768 kHz->2Hz)
Signal Chain
Display
Leftovers
Programming Logic
Power
USB and standoffs
Sheet: /
File: clock v3.kicad_sch

USB Supply

- Using USB-C connector

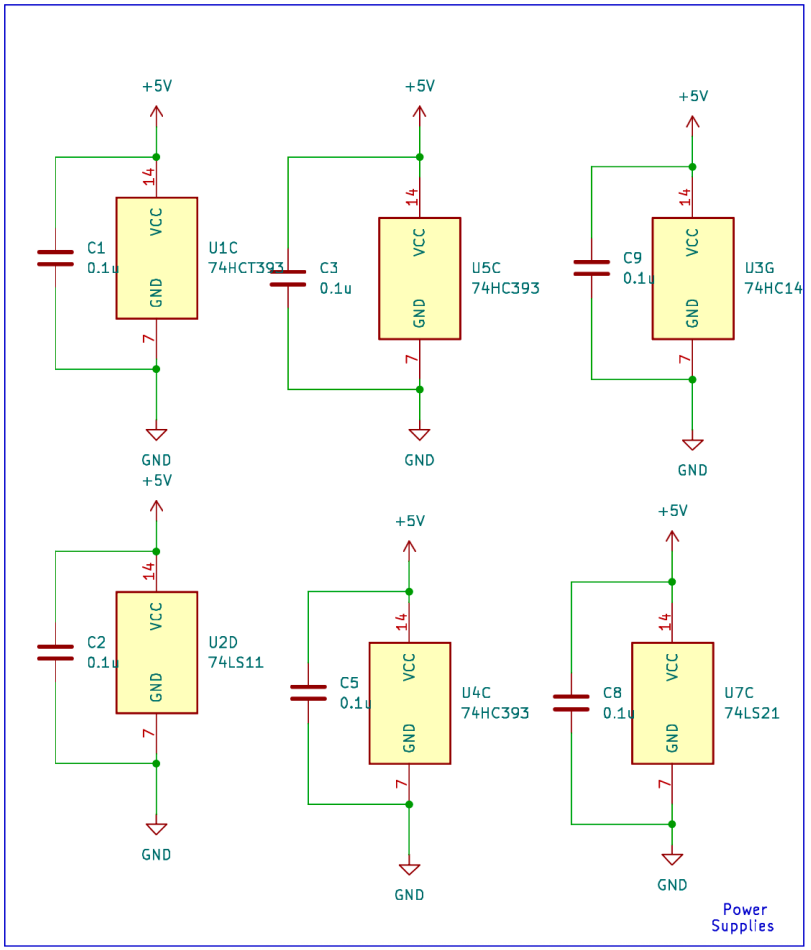


Power Supplies



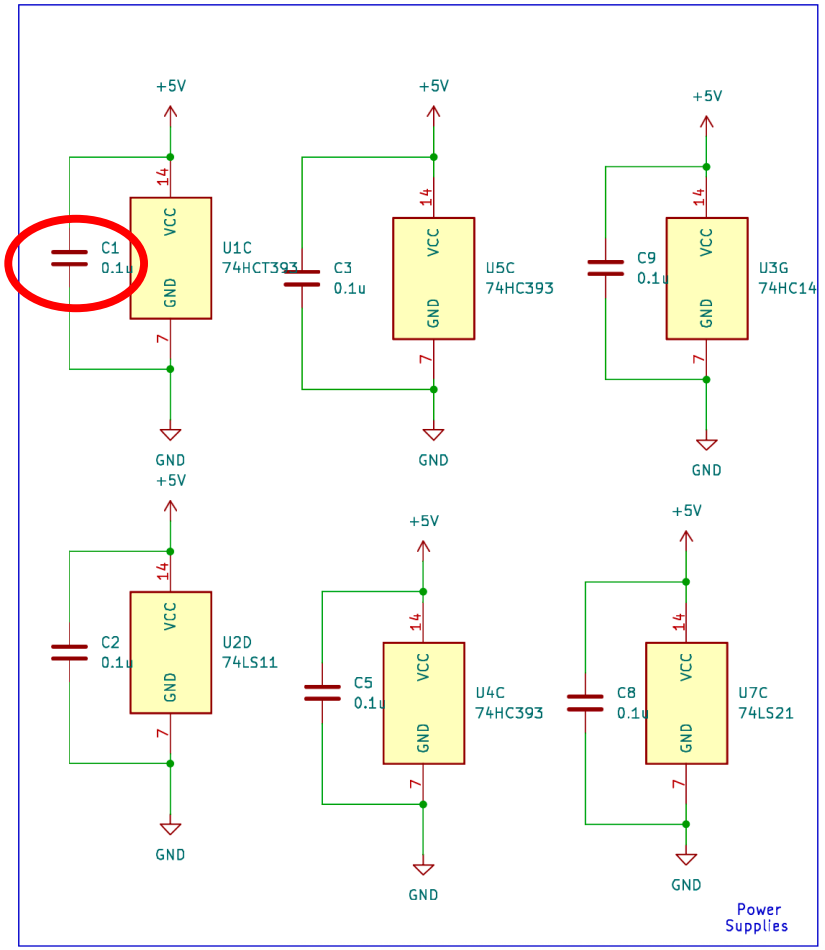
Power Supplies

- Many times in logic chips that have multiple internal devices (gates) but a common power supply, you'll see a separate symbol for the power supply hookup



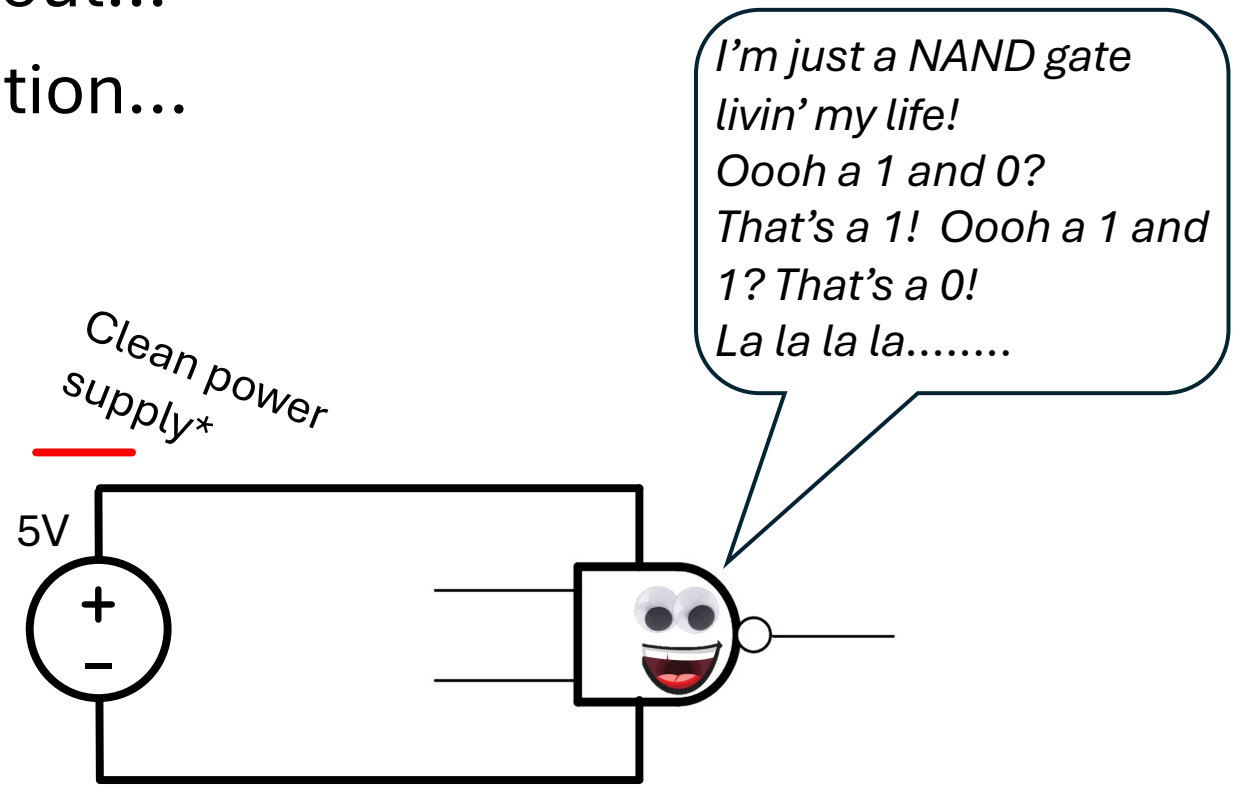
Decoupling Capacitors

- Very close to each chip you should place a capacitor of around 0.1 to 1 uF.
- This capacitor protects the chip against electrical noise which might cause it to have problems



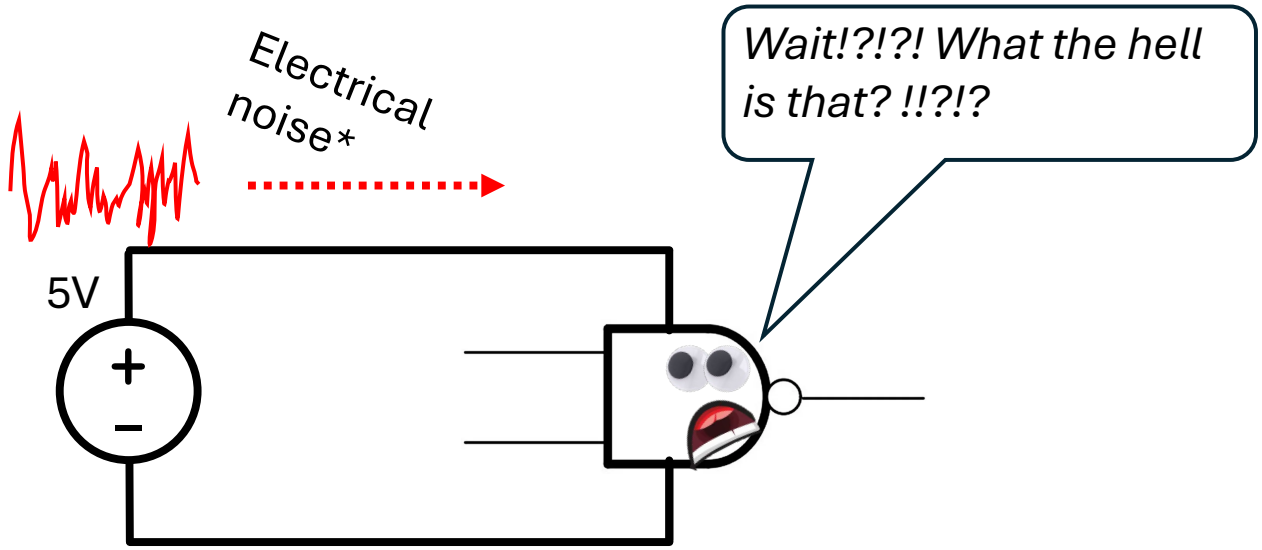
Why Decouple?

- Scenario without…
- Normal operation…



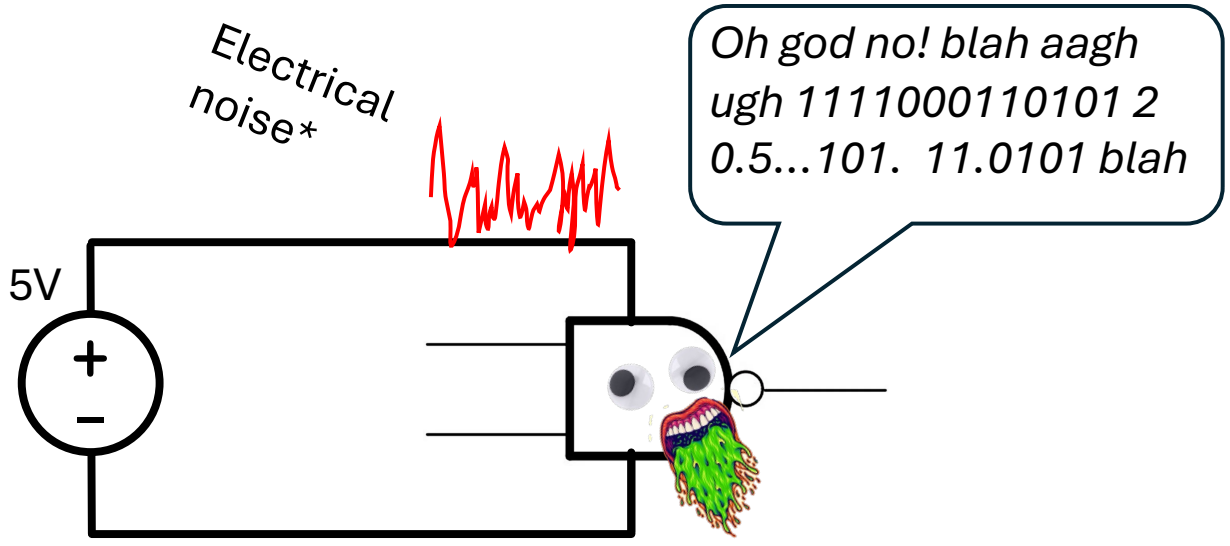
Why Decouple?

- Electrical noise gets on the power supply…



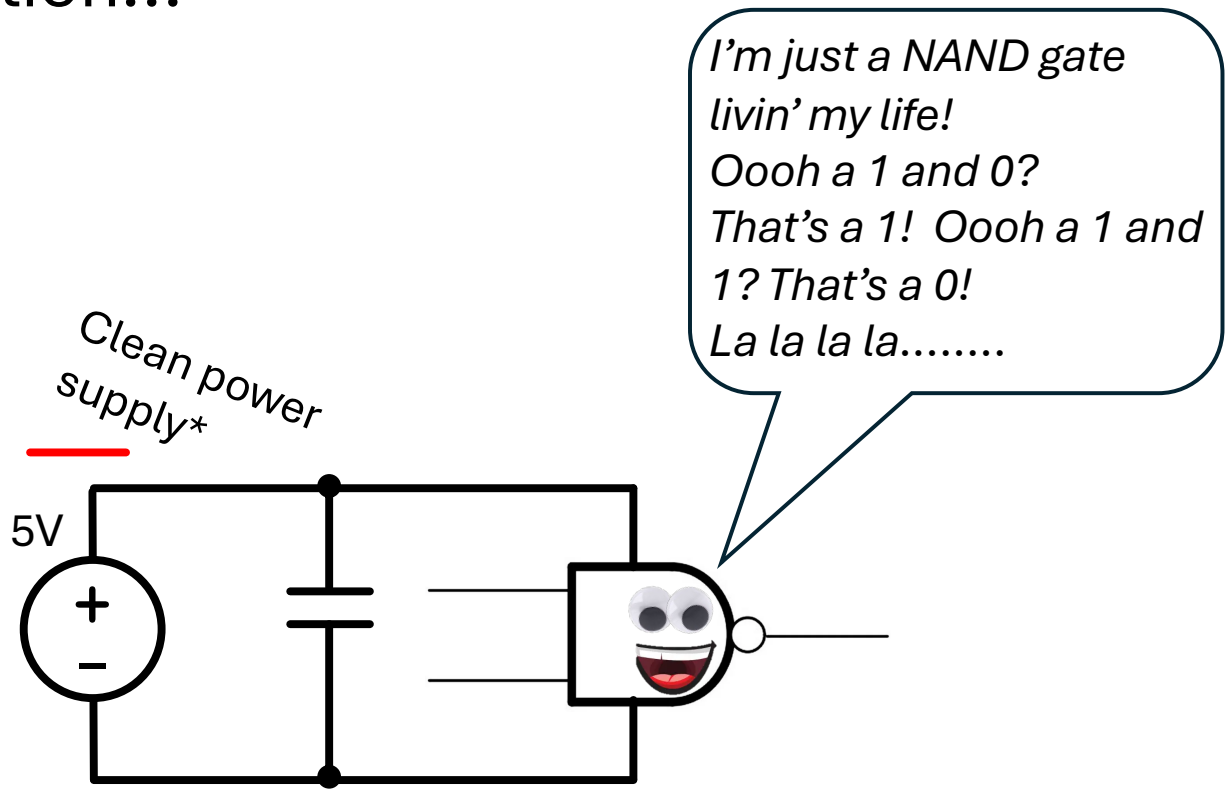
Why Decouple?

- Variations in the power supply can cause the digital chip to stop being:
  - Reliable
  - Digital
- NOT GOOD



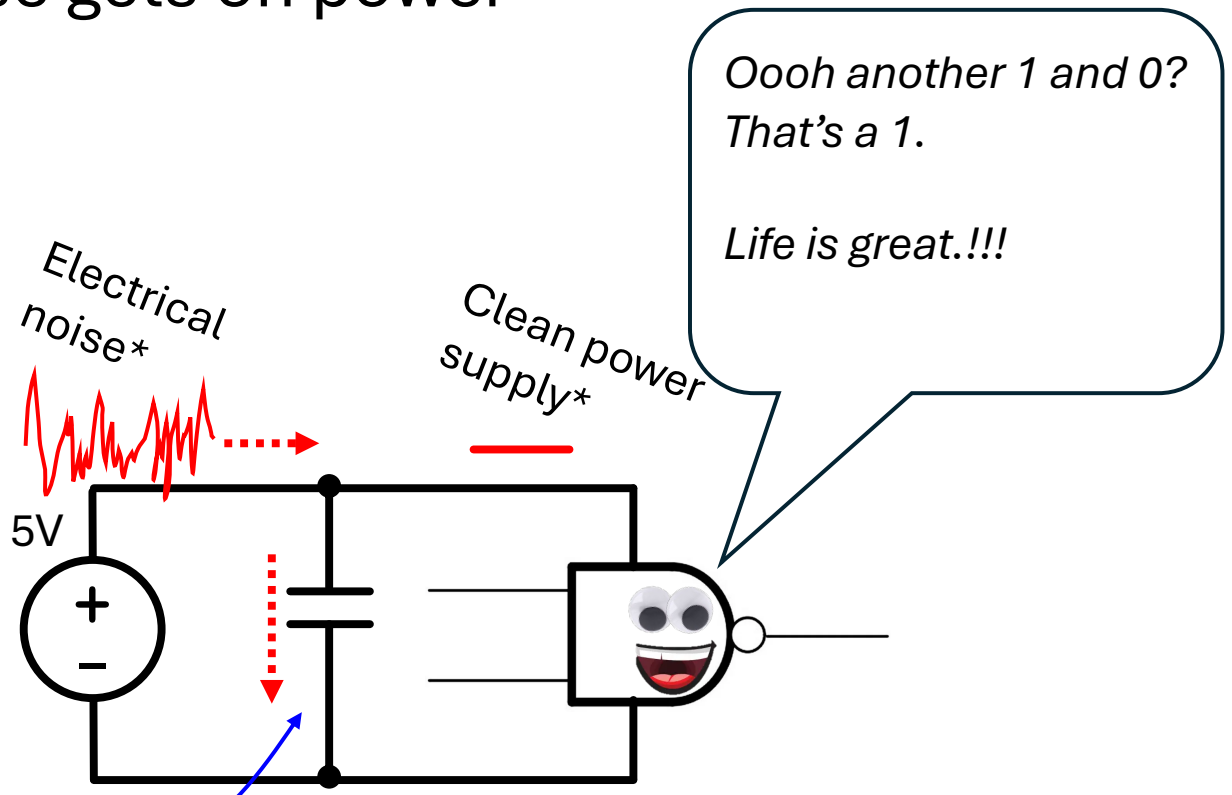
With Capacitor in Place…

- Normal operation…



With Capacitor in Place…

- Electrical noise gets on power supply line…



Impedance of capacitor is low for high frequency stuff but infinite for DC

$$Z_C = \frac{1}{j\omega C}$$

Protects device from noise

Conclusion: Always Put Decoupling Capacitors in your Circuit!

- You don’t always need them but they protect against unforeseen issues, like seatbelts or insurance.

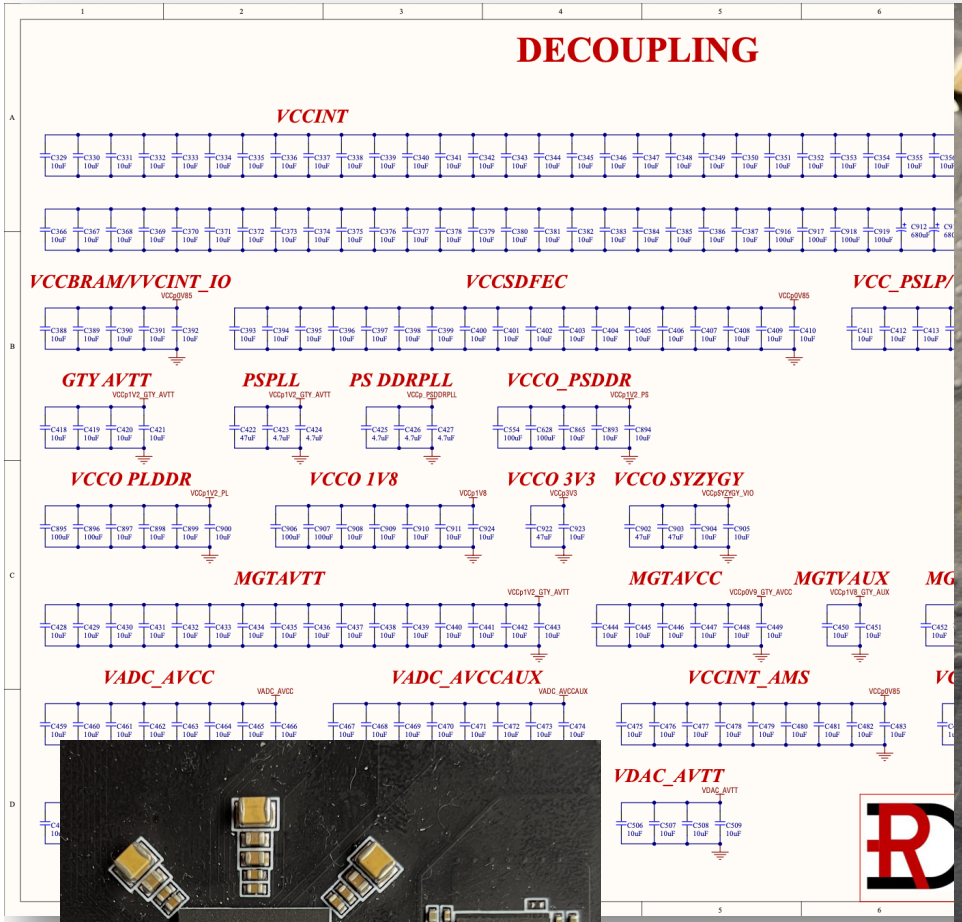
- Always include them.

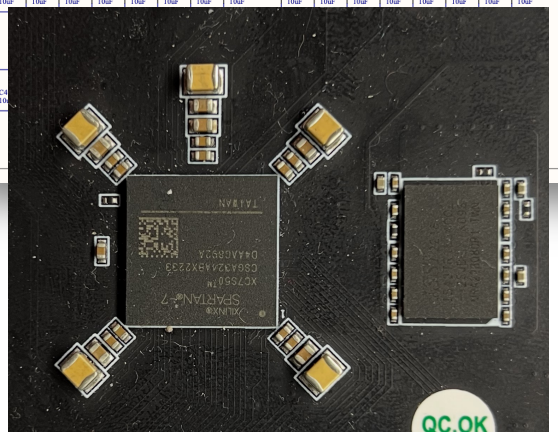
- Seriously!

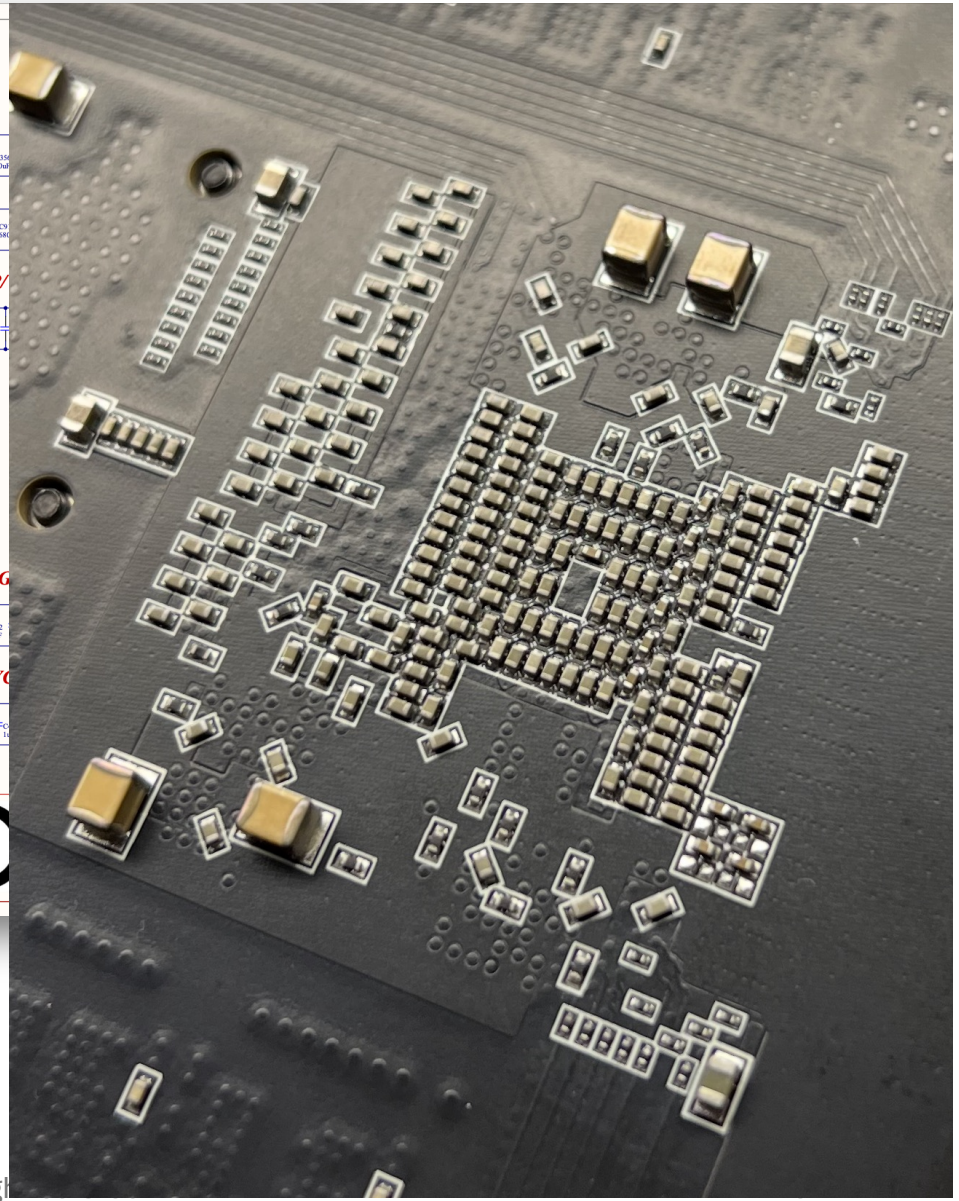
Decoupling Capacitors

- In modern devices you'll always see at least one decoupling cap per set of power supply pins (and many modern chips have many pins)

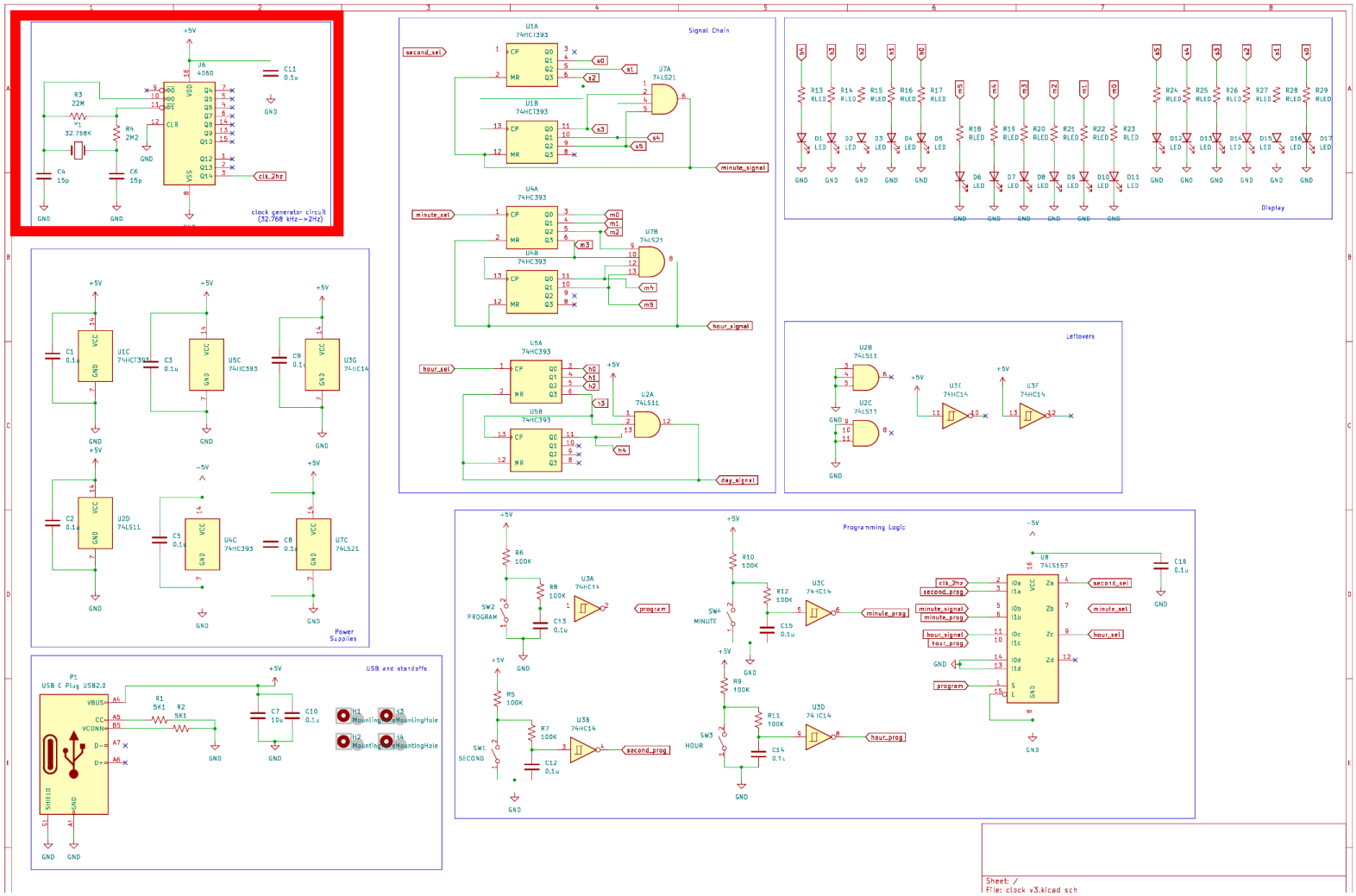
Decoupling Everywhere...





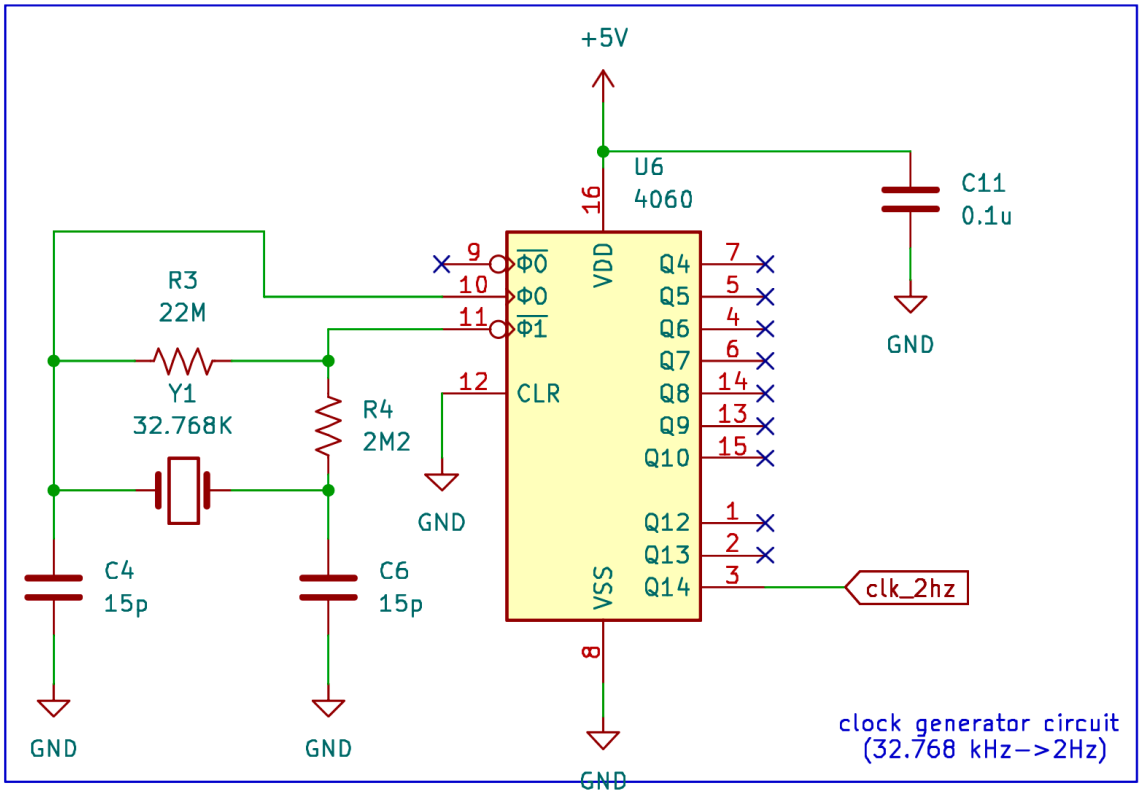


Clock Generator



Clock Generator

- What we talked about earlier except this chip does it all.



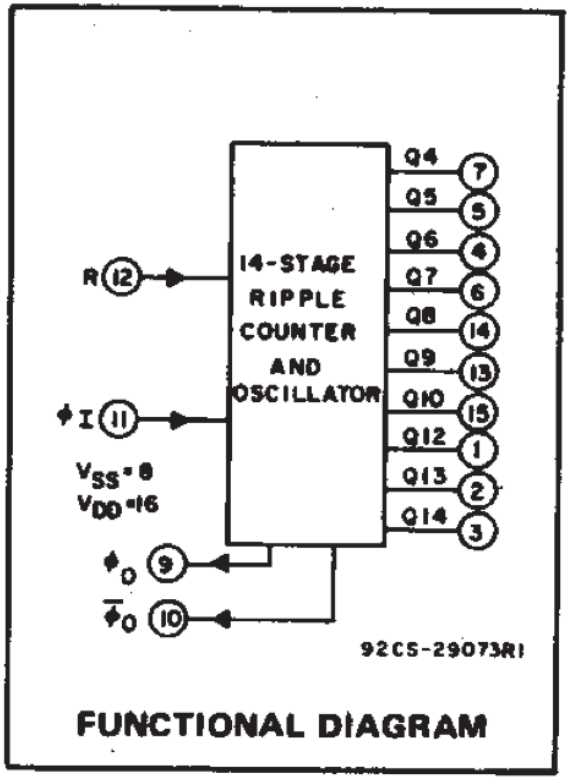
What is the 4060?

- Has built-in clock circuit, Schmitt trigger and other supporting logic to clean up clock.
- Also has a built-in 14-stage counter… So we can divide down clock

TEXAS INSTRUMENTS

Data sheet acquired from Harris Semiconductor
SCHS049C – Revised October 2003

CMOS 14-Stage Ripple-Carry Binary Counter/Divider and Oscillator

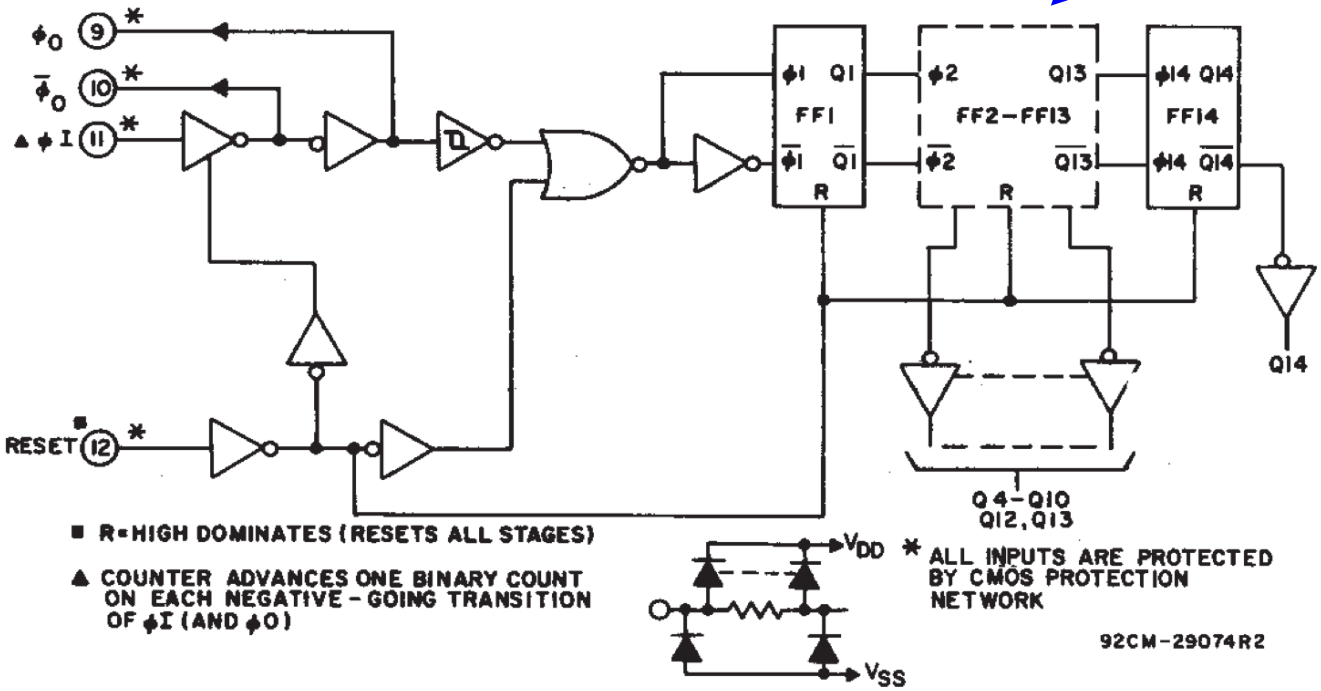


FUNCTIONAL DIAGRAM

4060 Guts

Chain of flip flops to divide clock down

Crystal and supporting parts go here

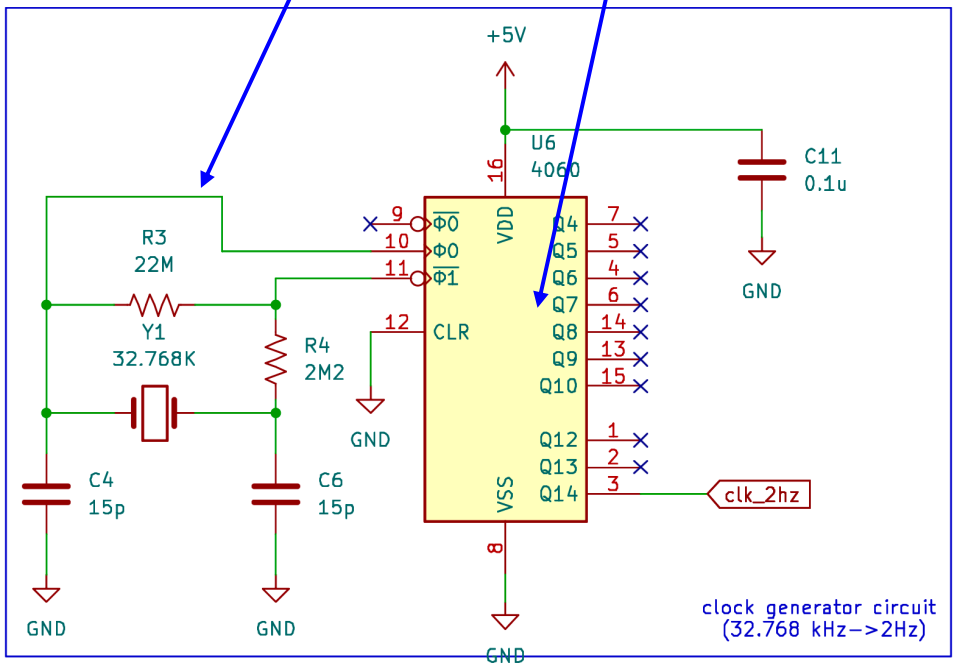


Clock Generator

32.768 kHz quartz oscillator
±10ppm

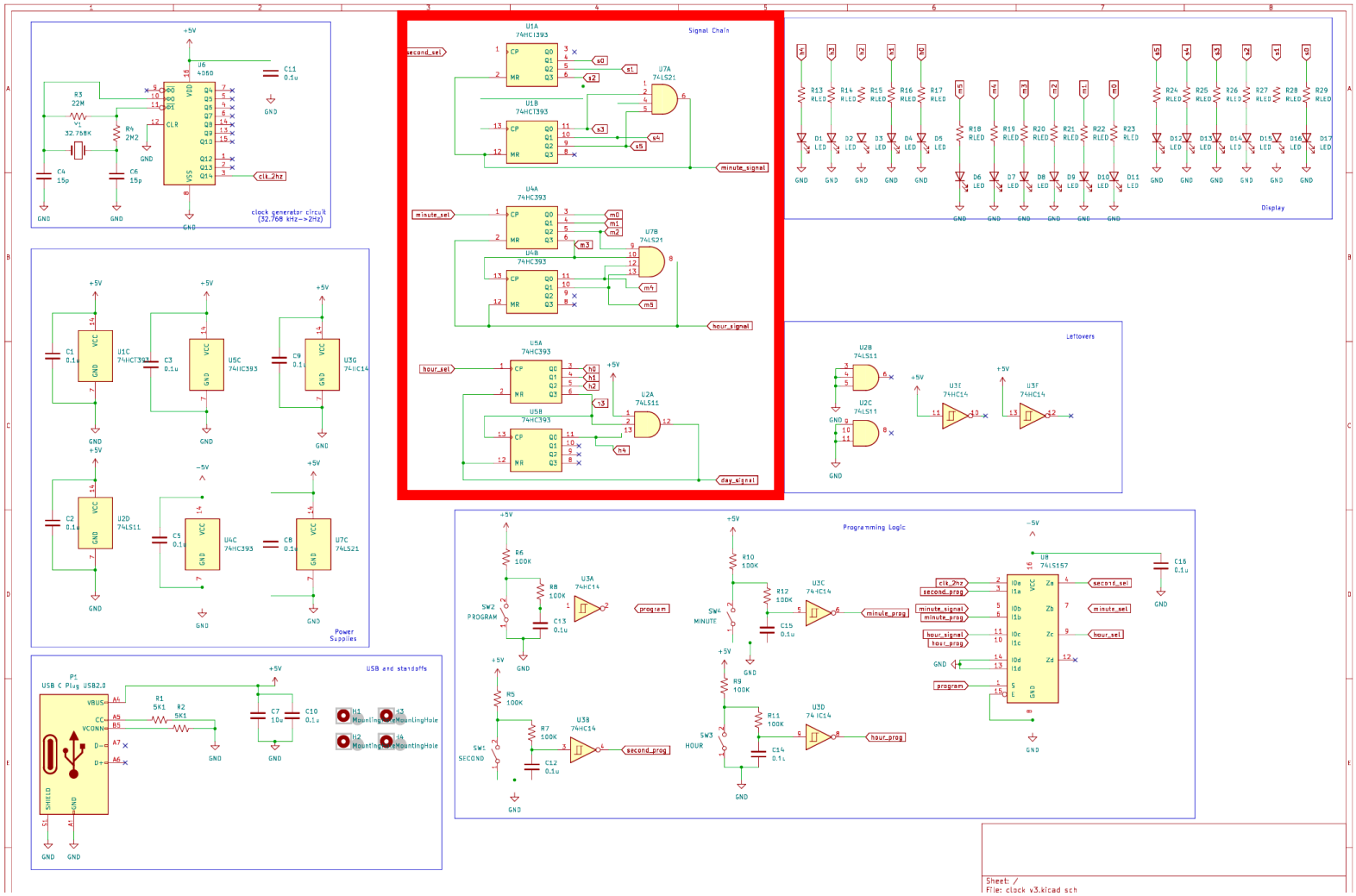
Long Chain of divide-by-2 counters

Pierce Oscillator



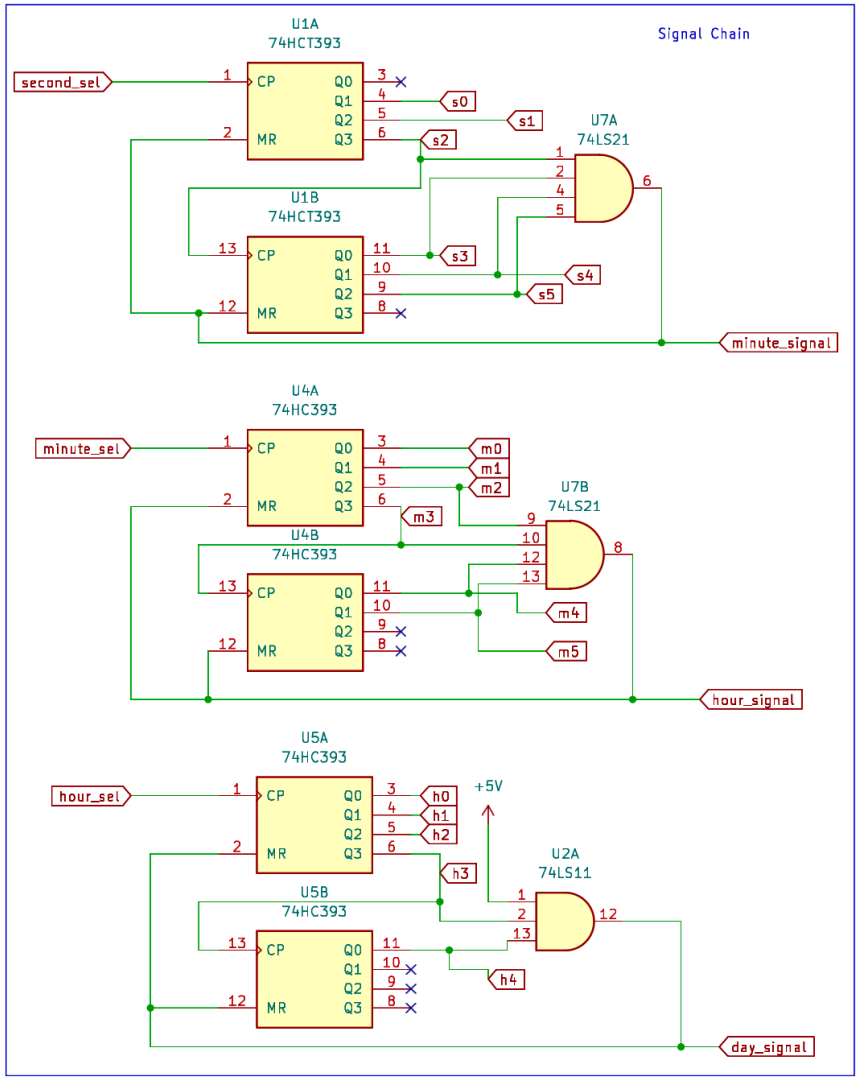
- 32.768 kHz crystal is our ultimate clock source

Counters

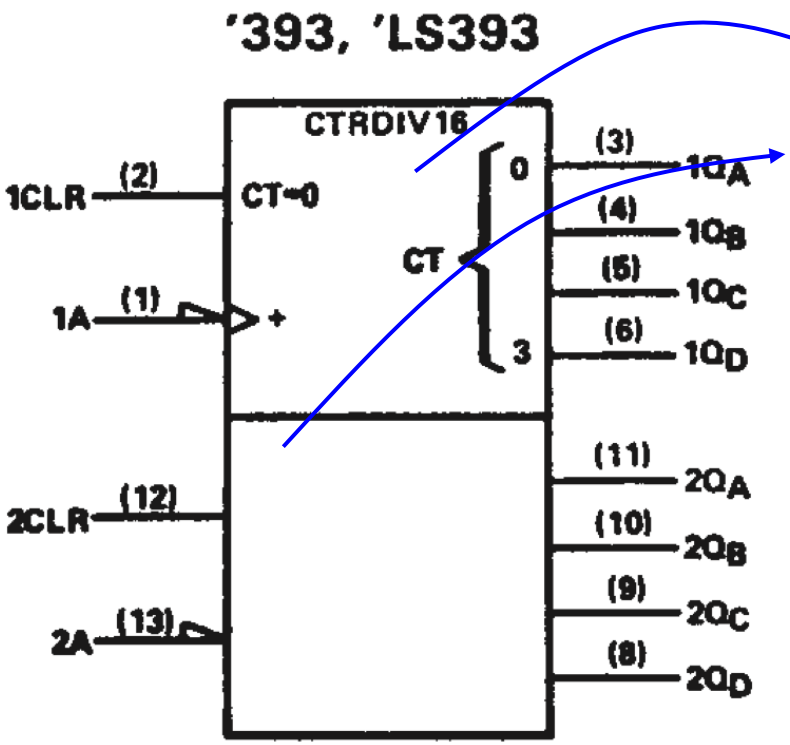

Signal Chain
Display
Leftovers
Programming Logic
Power Supplies
USB and standoffs
clock generator circuit (32.768 kHz->2Hz)
second_set
minute_set
hour_set
day_signal
minute_signal
hour_signal
clk_2hz
program
second_prog
minute_prog
hour_prog
Sheet: /
File: clock v3.kicad_sch

Counters

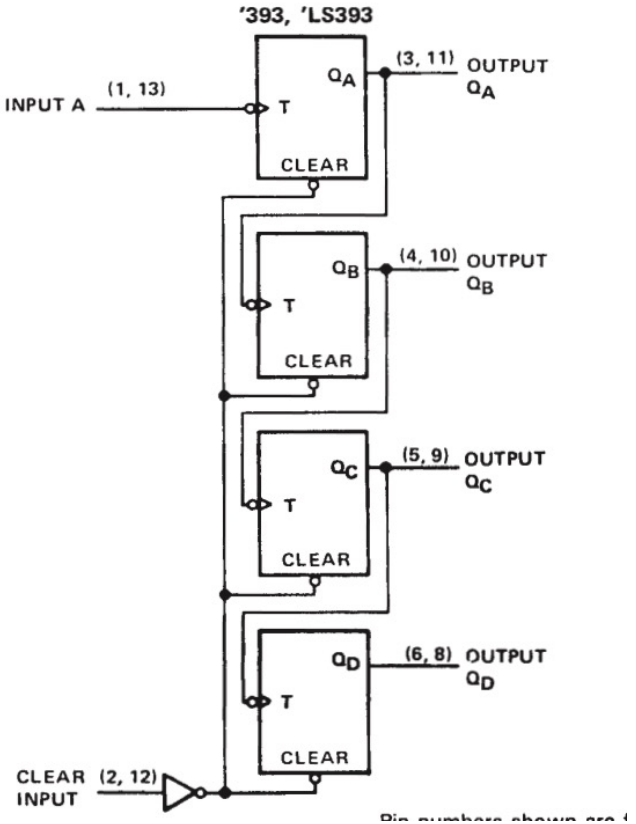
- Three separate counters for seconds, minutes, and hours
- Same exact type of counters we talked about previously…just now all packaged up a bit to save on some wiring.



74393...dual four bit counters

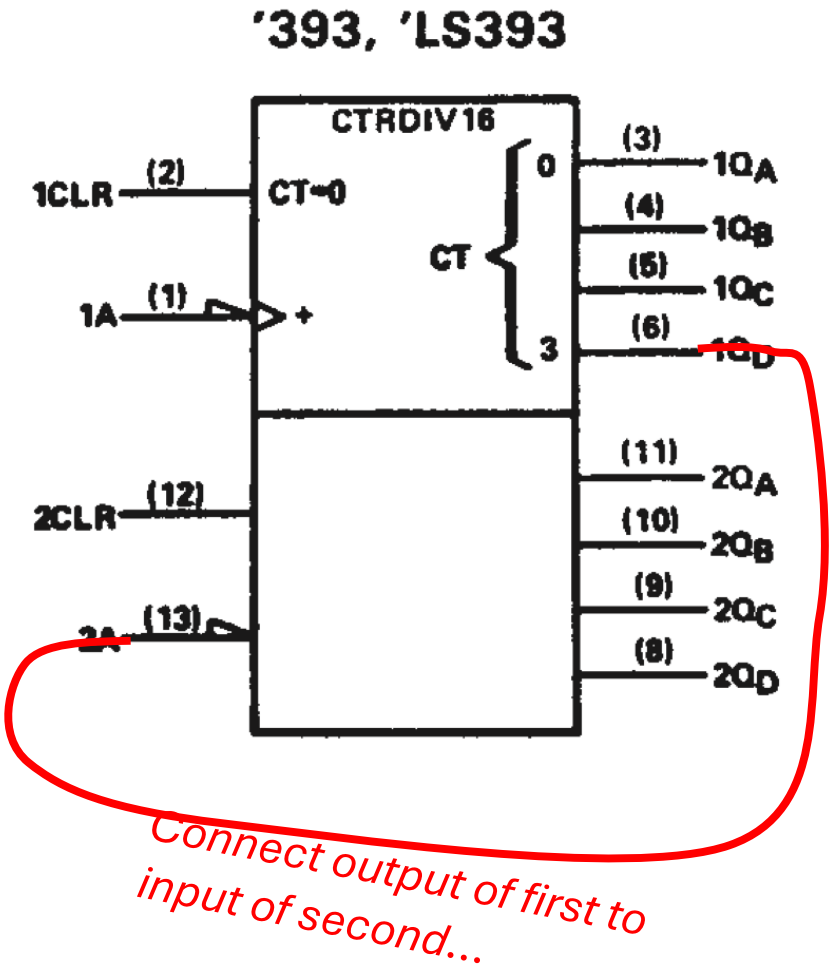


Each half is one of these



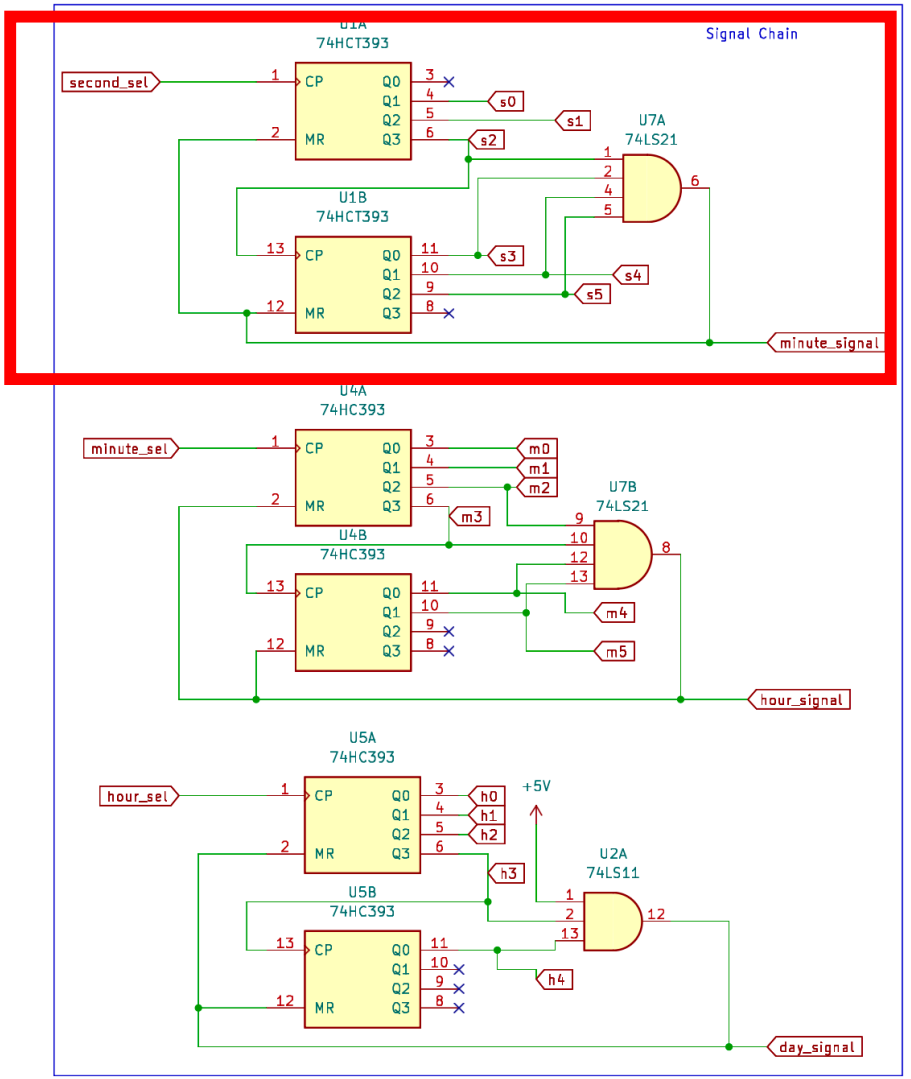
Using 74393

- We'll use one 393 for seconds, one for minutes, one for hours.
- On each chip we'll chain each 4 bit counter togethe so each chip can count up to...
- 255
- ...but we'll actually reset them early using logic for the time chunks



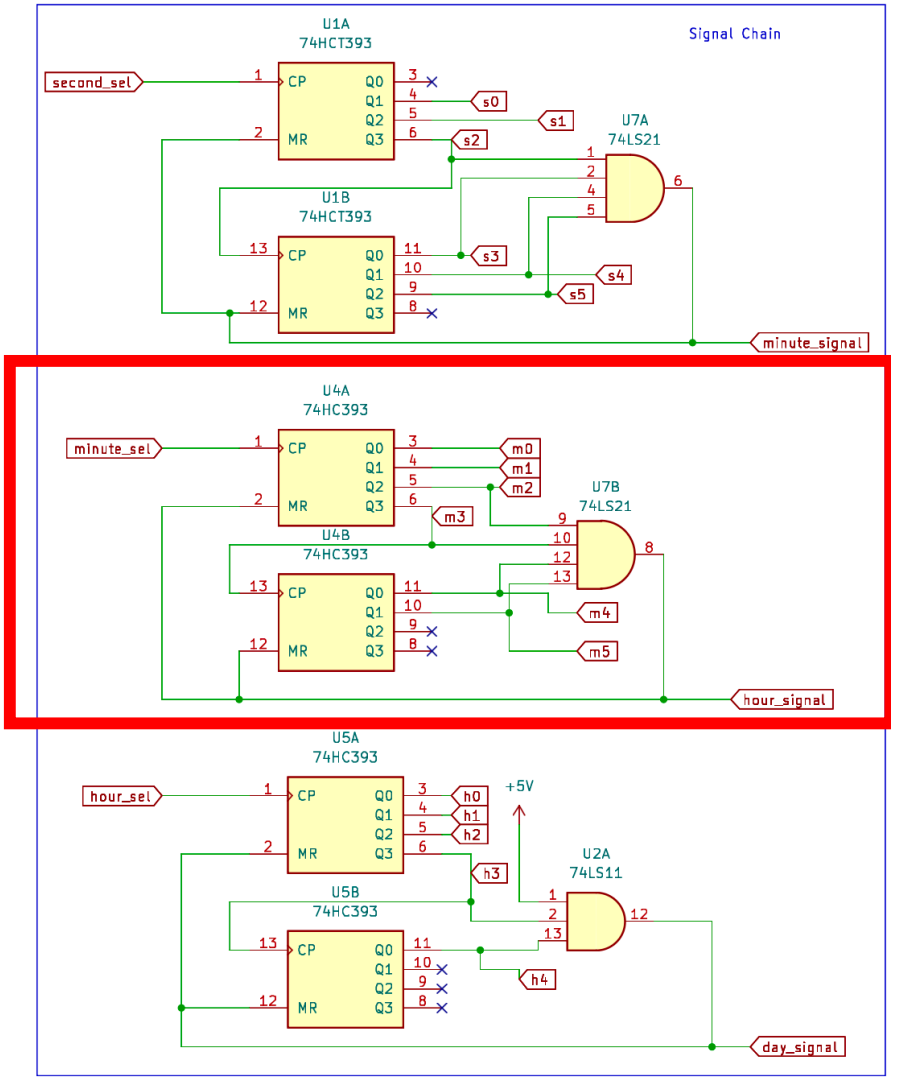
Counters

- Three separate counters
- First counts to 120 ½-seconds…use logic to generate “minute signal” both to reset the second clock and to provide a 1 minute pulse



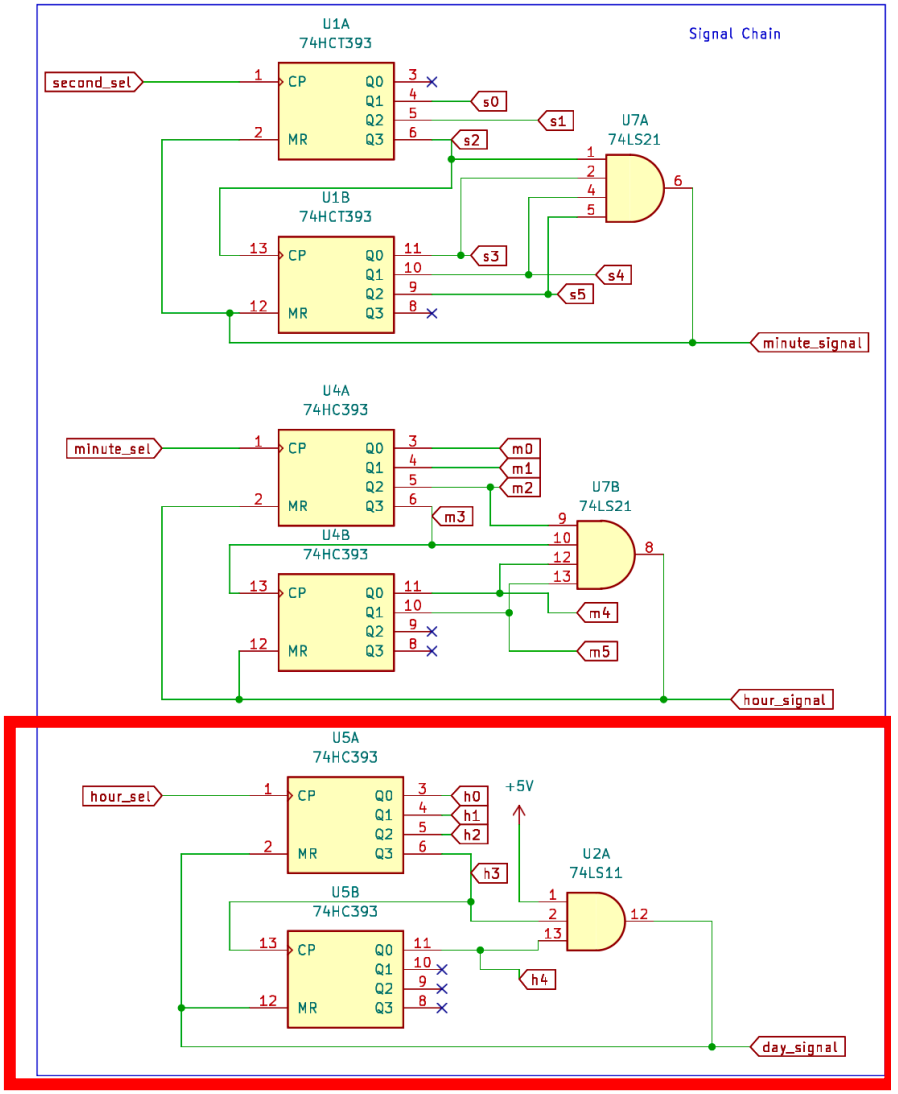
Counters

- Three separate counters
- Second counts to 60 minutes…use logic to generate “hour signal” both to reset the minute clock and to provide a 1 hour pulse



Counters

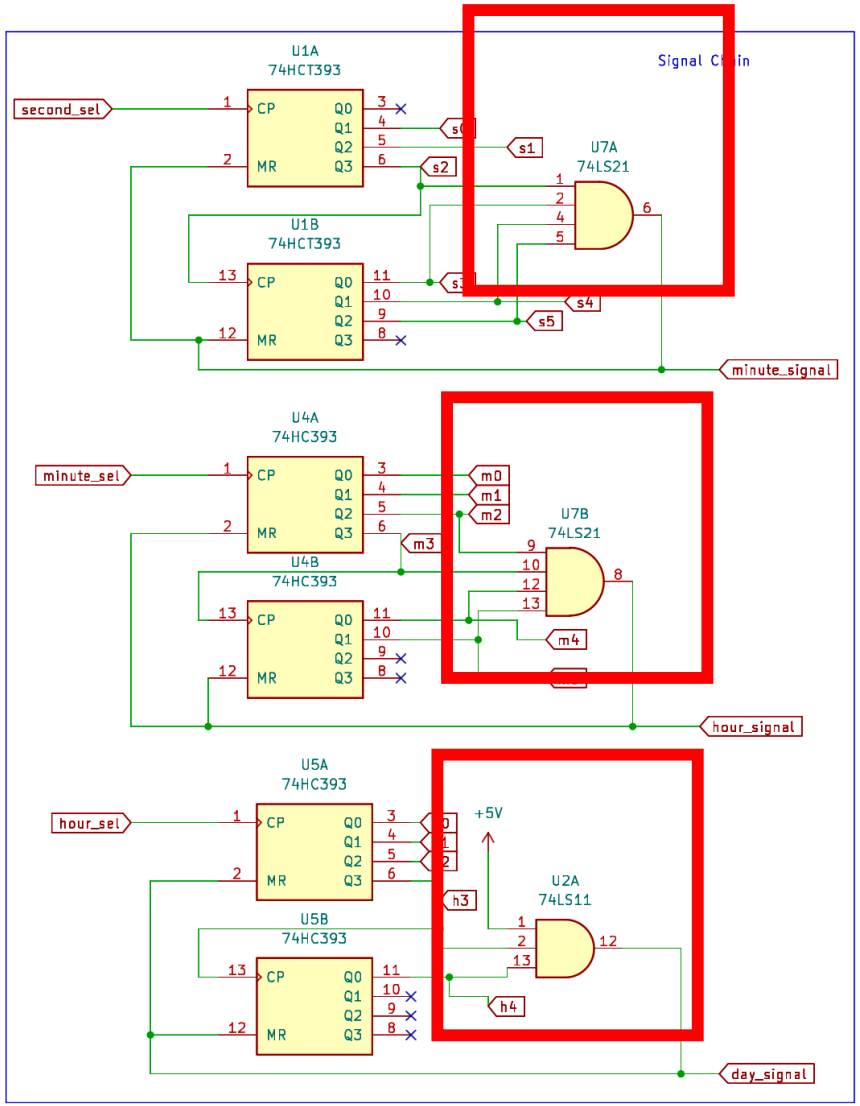
- Three separate counters
- Third counts to 24 hours…use logic to generate “day signal” to reset the hour clock



Reset Logic

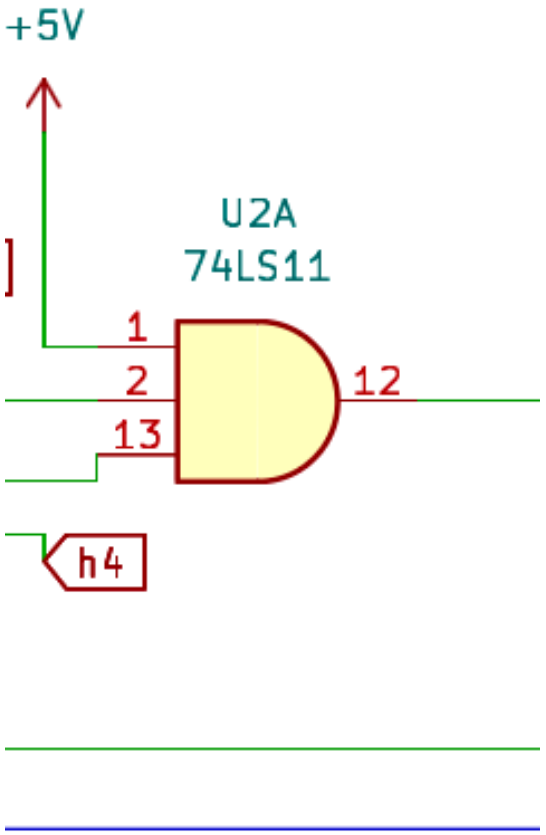
- Use 74LS21 to reset seconds, minutes
- What value to reset at (in binary?)

- Use 74LS11 to reset the hour
- What value to reset at (in binary?)...

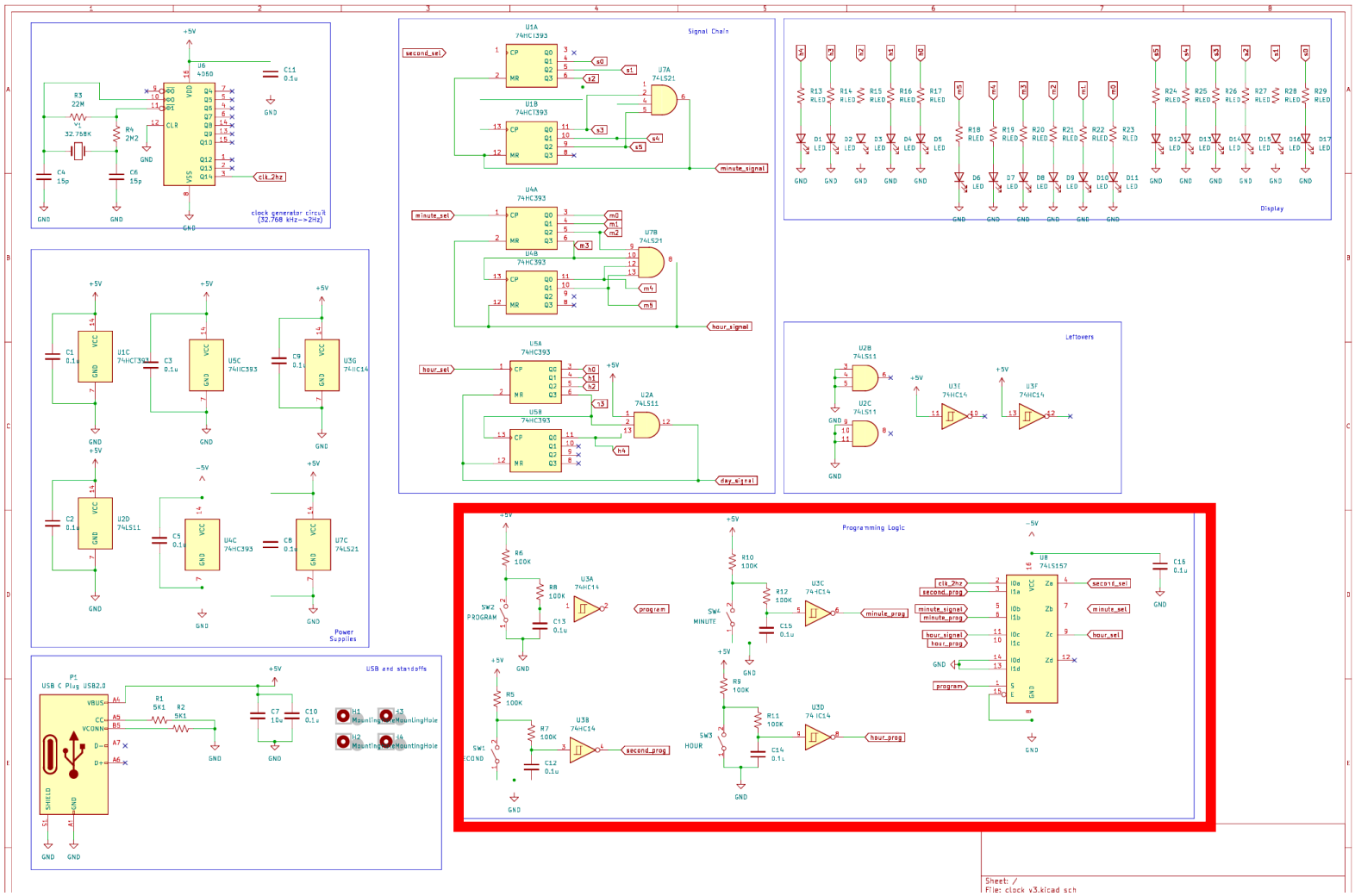


Why did I use a three-input AND gate instead of a two-input AND gate?

- Because I had a bunch extra and I'm stupid.



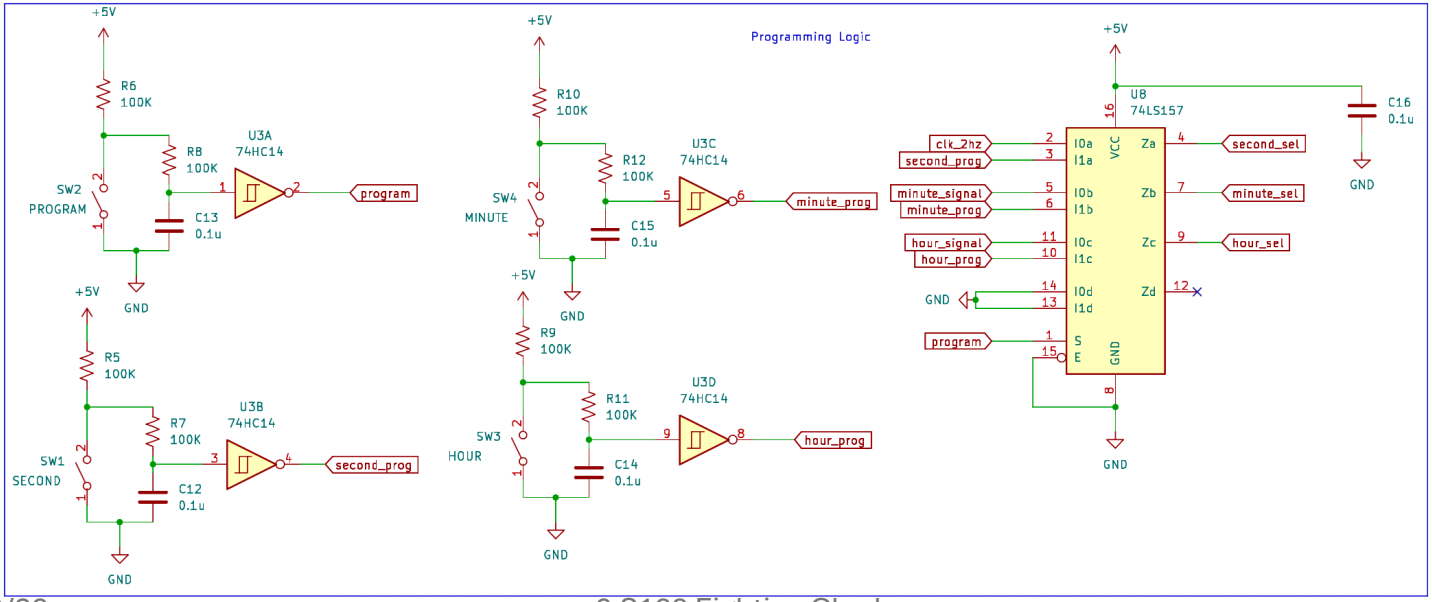
Programming Logic



Programming Logic

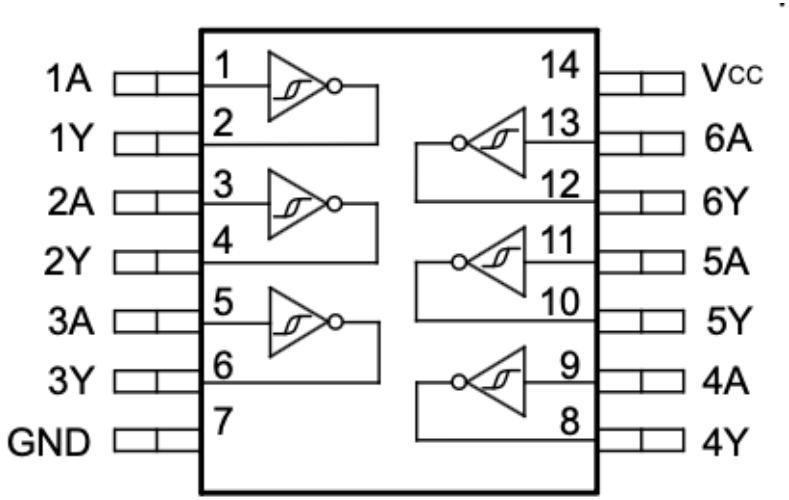
You 110% need to debounce this! Do not ignore this part!

- Two parts:
  - Debounced buttons for setting time
  - Multiplexer to choose between button pulses (for programming) and clock pulses (for normal operation)



74HC14 Schmitt Trigger

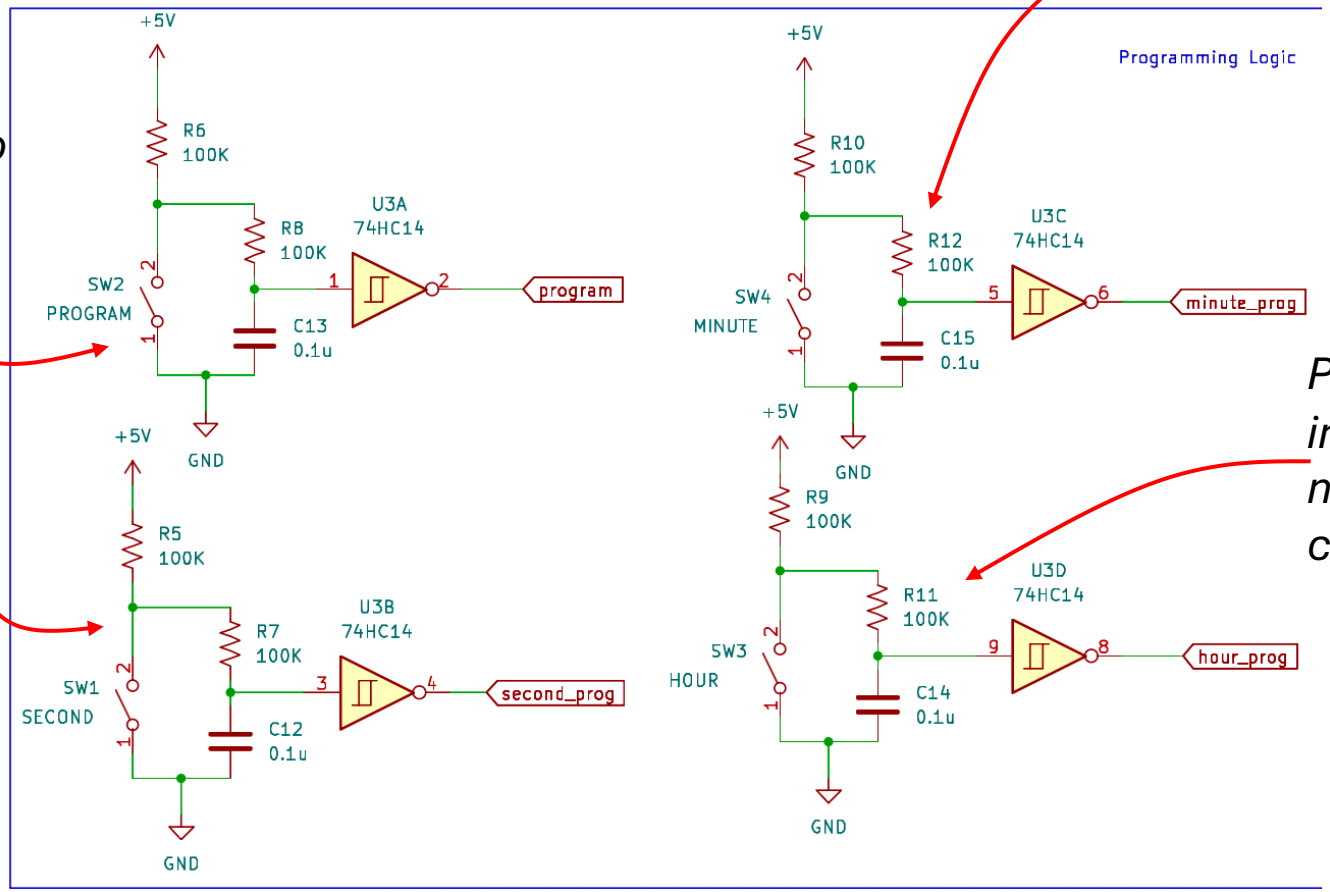
- To help with debouncing



Programming Buttons

Push to increment minute counter

Push/hold to put into program mode



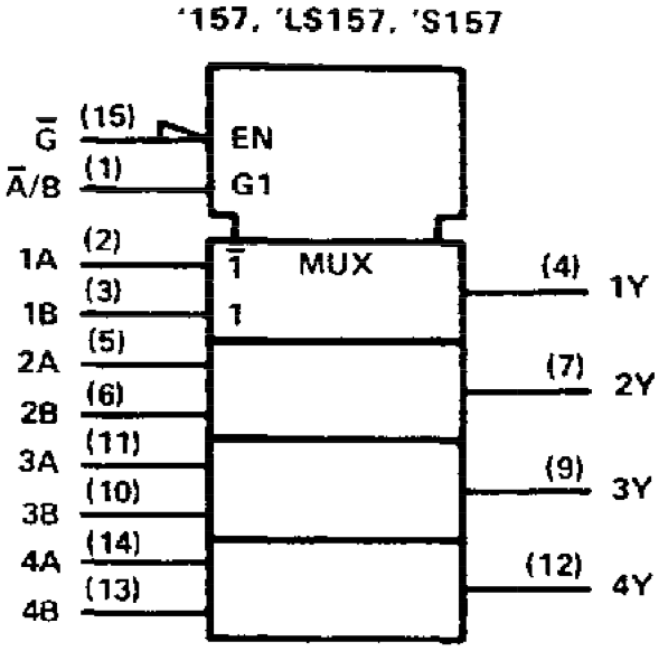
*Push to increment second counter**

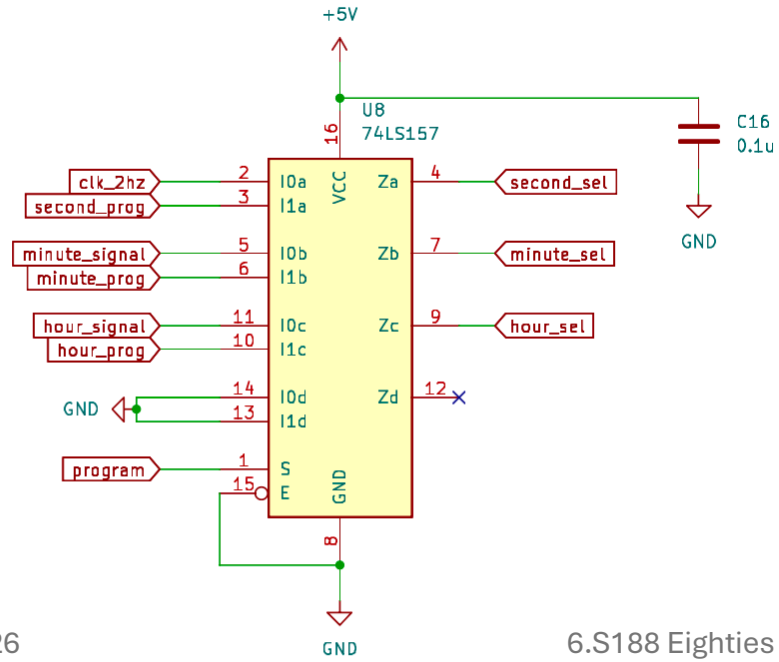
Push to increment minute counter

*in half-second increments

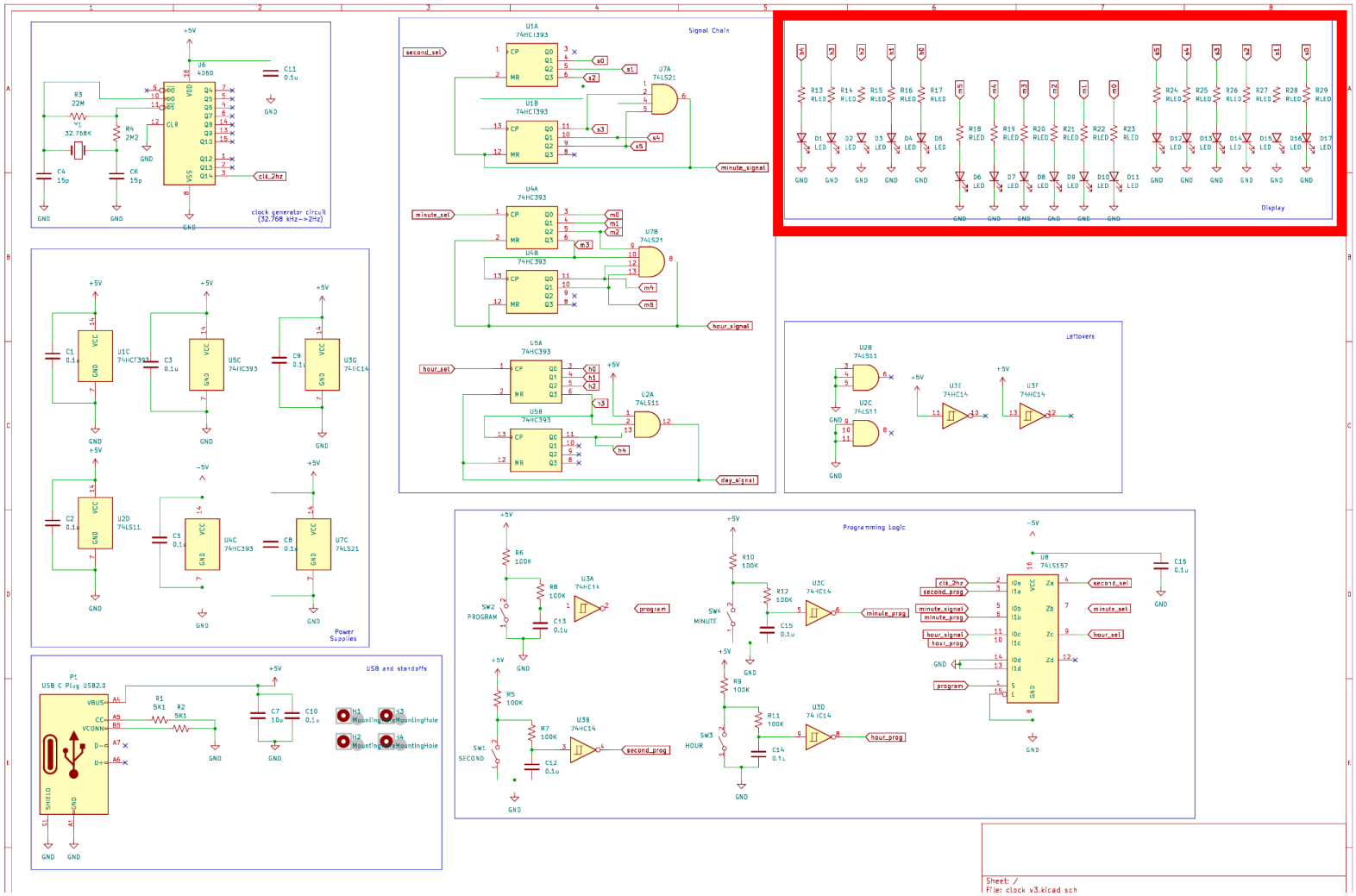
Selection done with 74157

- Quad 2-to-1 Multiplexer
- (we derived circuit for these in week 1!!!)



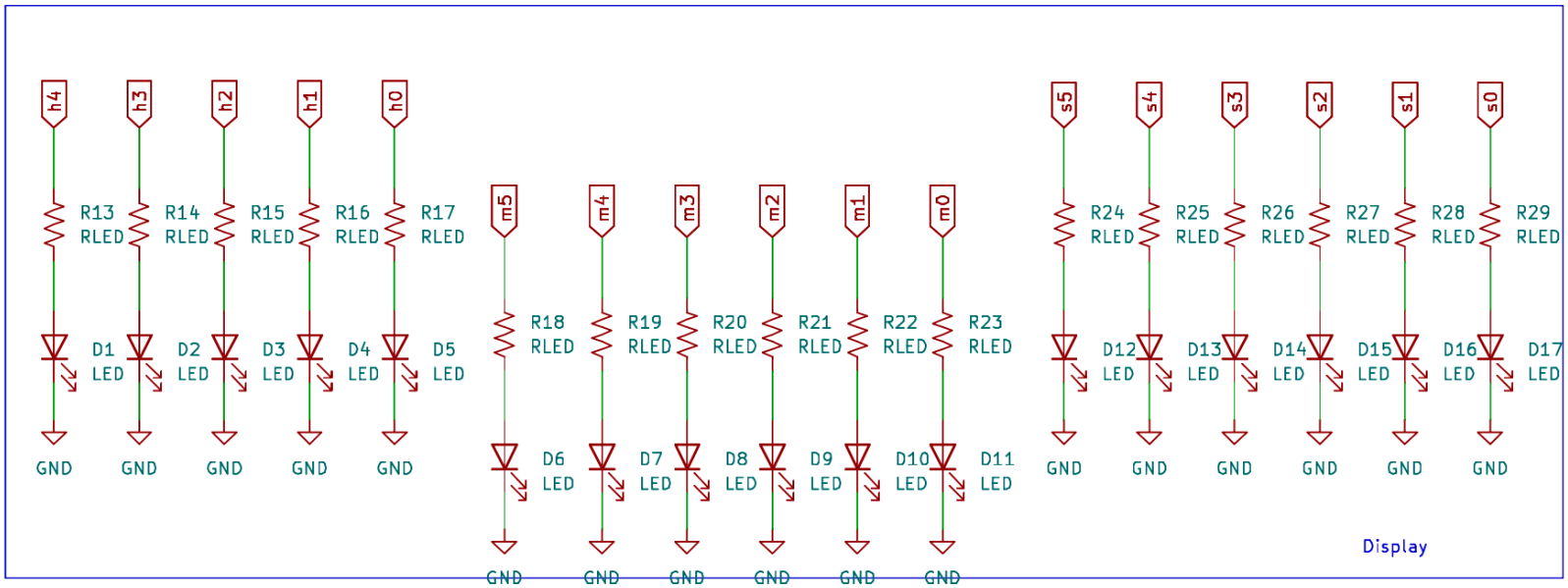


Display

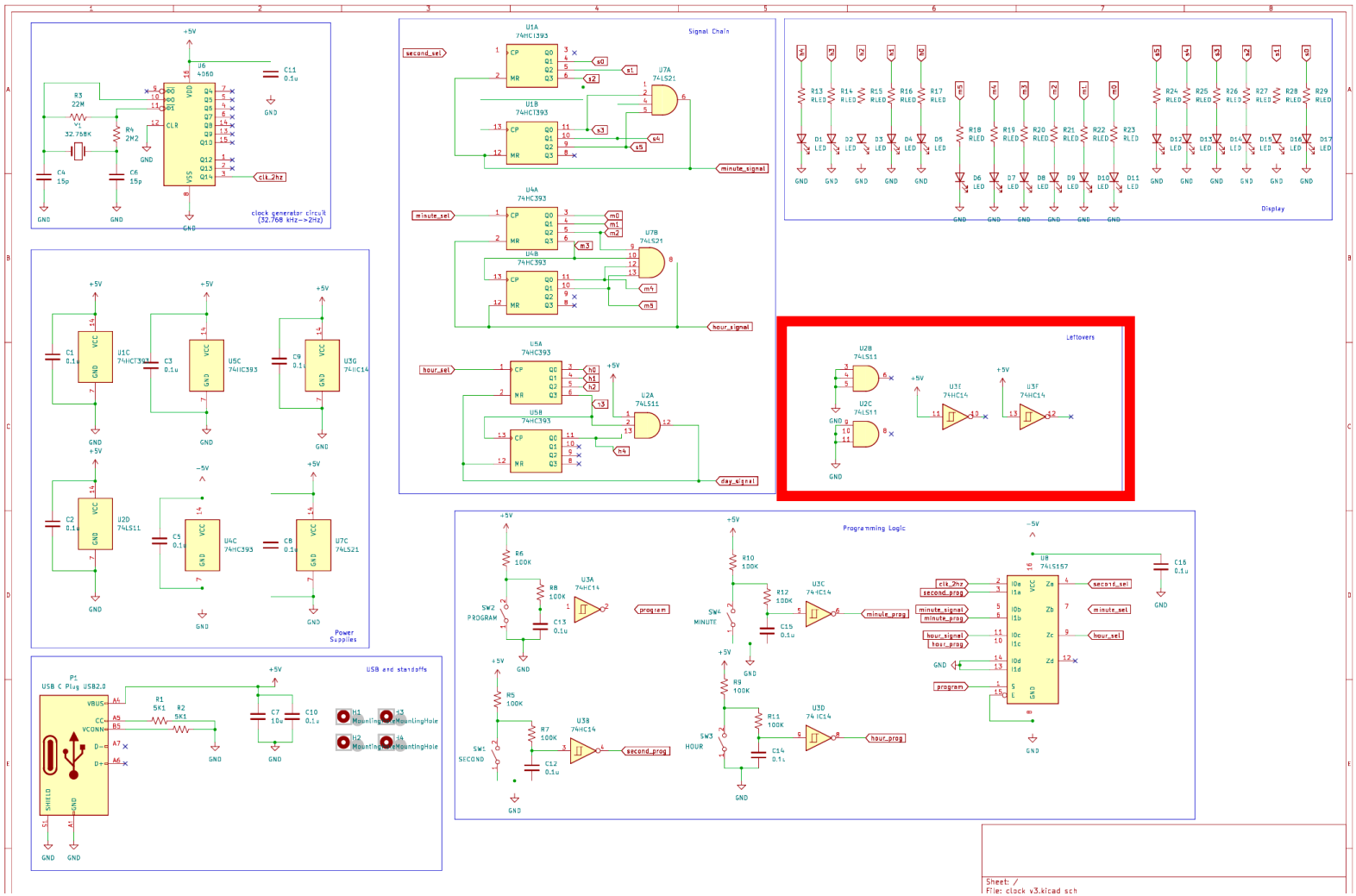

clock generator circuit (32.768 kHz->2Hz)
Signal Chain
Display
Leftovers
Power Supplies
USB and standoffs
Programming Logic
Sheet: /
File: clock v3.kicad sch

Display

- Just a bunch of resistors and LEDs. Nothing else to see here…move along

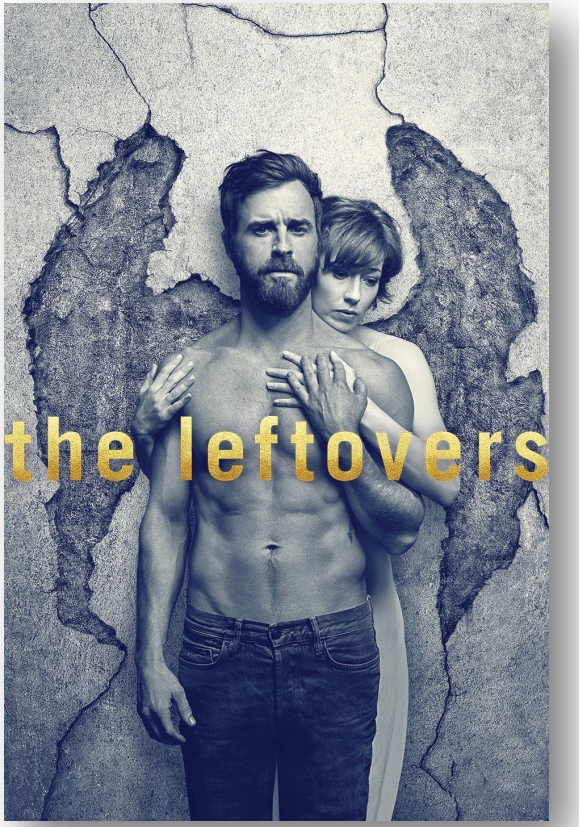


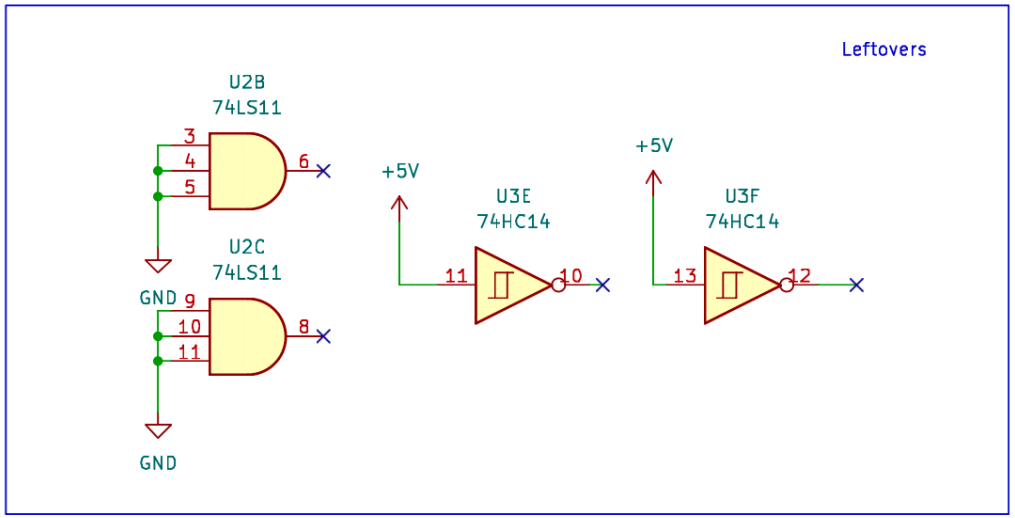
Extra Gates


clock generator circuit (32.768 kHz->2Hz)
Signal Chain
Display
Leftovers
Power Supplies
USB and standoffs
Programming Logic
clk_2hz
second_set
minute_set
hour_set
second_signal
minute_signal
hour_signal
day_signal
second_prog
minute_prog
hour_prog
program
Sheet: /
File: clock v3.kicad_sch

Extra Gates

- Tie the inputs of unused gates to some value so they don't randomly flip on/off from electrical noise (causing noise and using power in the process)





OK so Build Options:

- Two pre-made PCBs:
  - One using surface-mount components
  - One using through-hole components
- Breadboard your own clock from scratch using schematic (with modifications as you see fit)
- Solder up your own clock from scratch on perfboard using schematic (with modifications as you see fit)
- Design your own PCB off of starting schematic and we can send out (though likely won't get back prior to end of IAP at this rate…)