Physical Attacks

MIT Secure HW Design Spring 2024

Mengjia Yan & Joseph Ravichandran

Image: Proto G Engineering, "Oscilloscope Art"

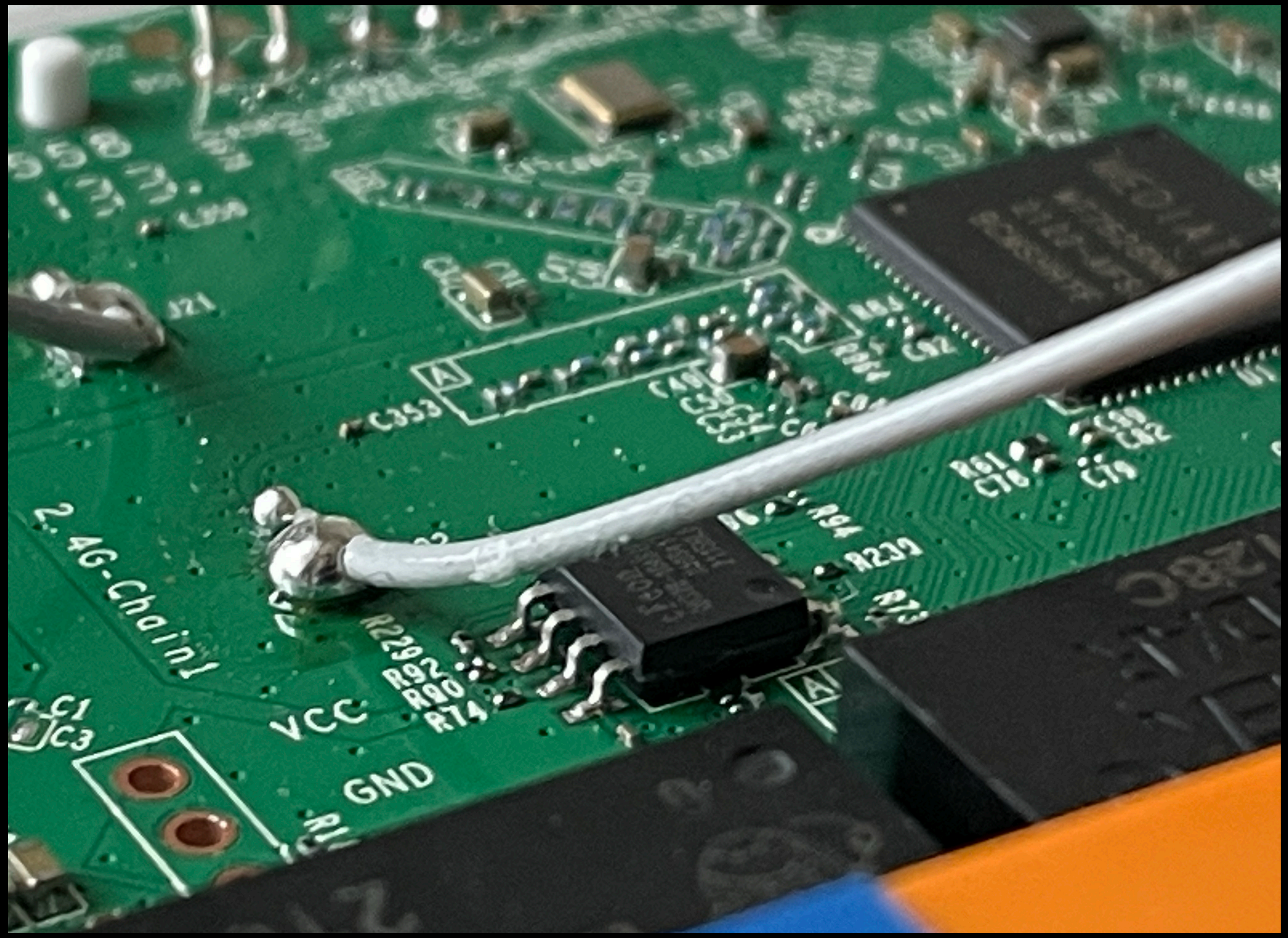
What's a Computer?

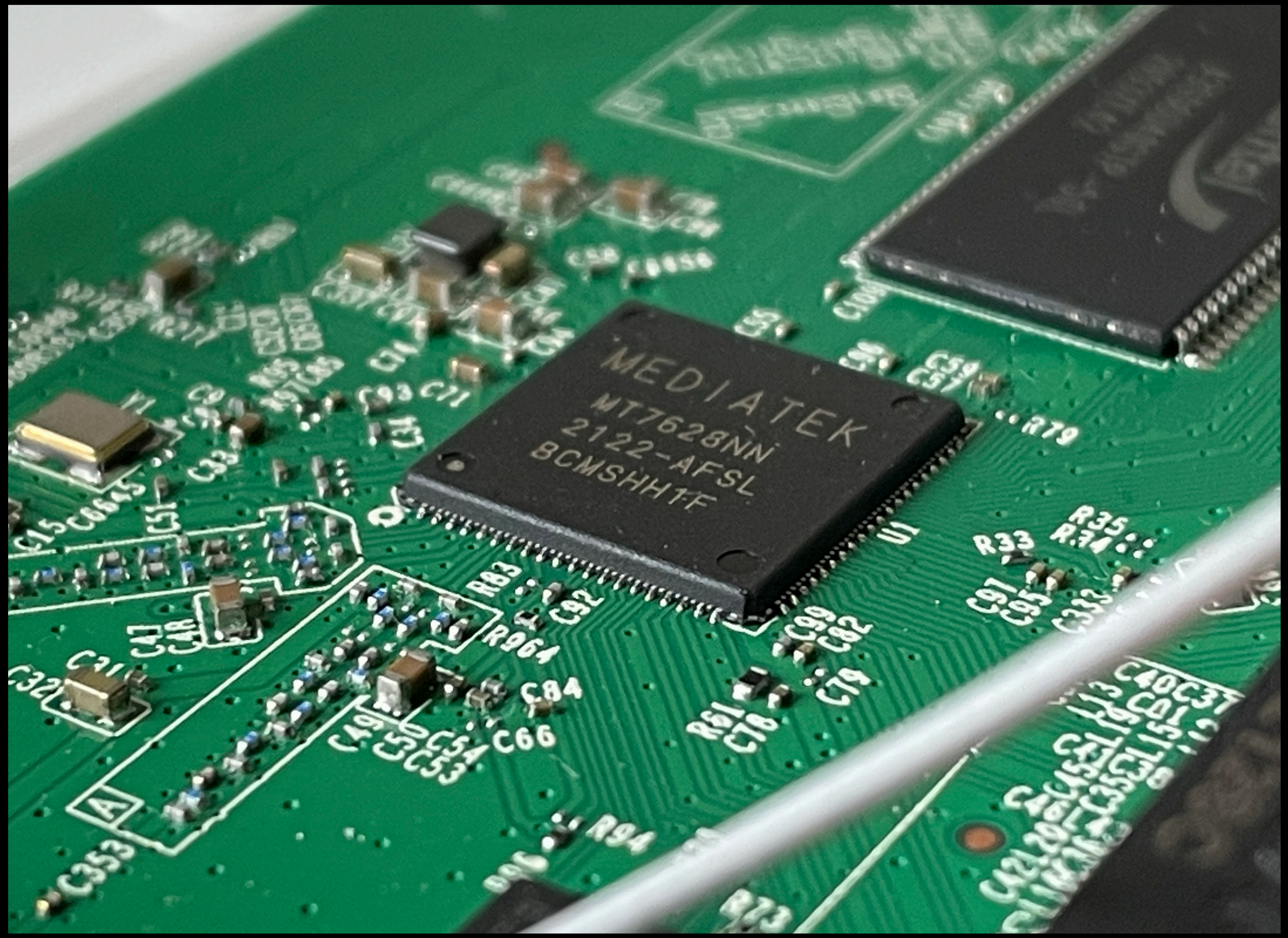
What's a Computer?

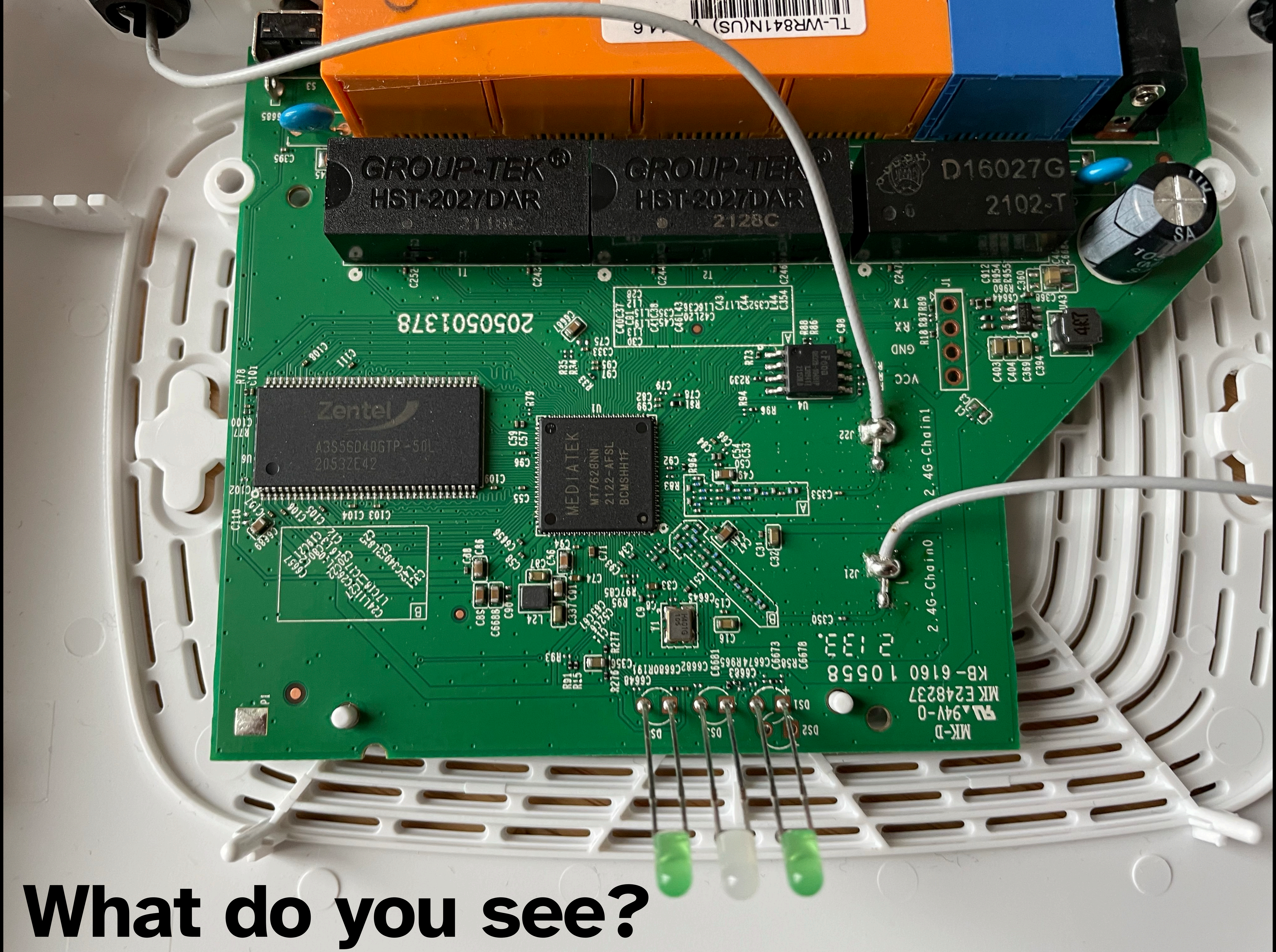
- Processor
- Memory
- Storage


What's Inside?
TP-Link WR841N

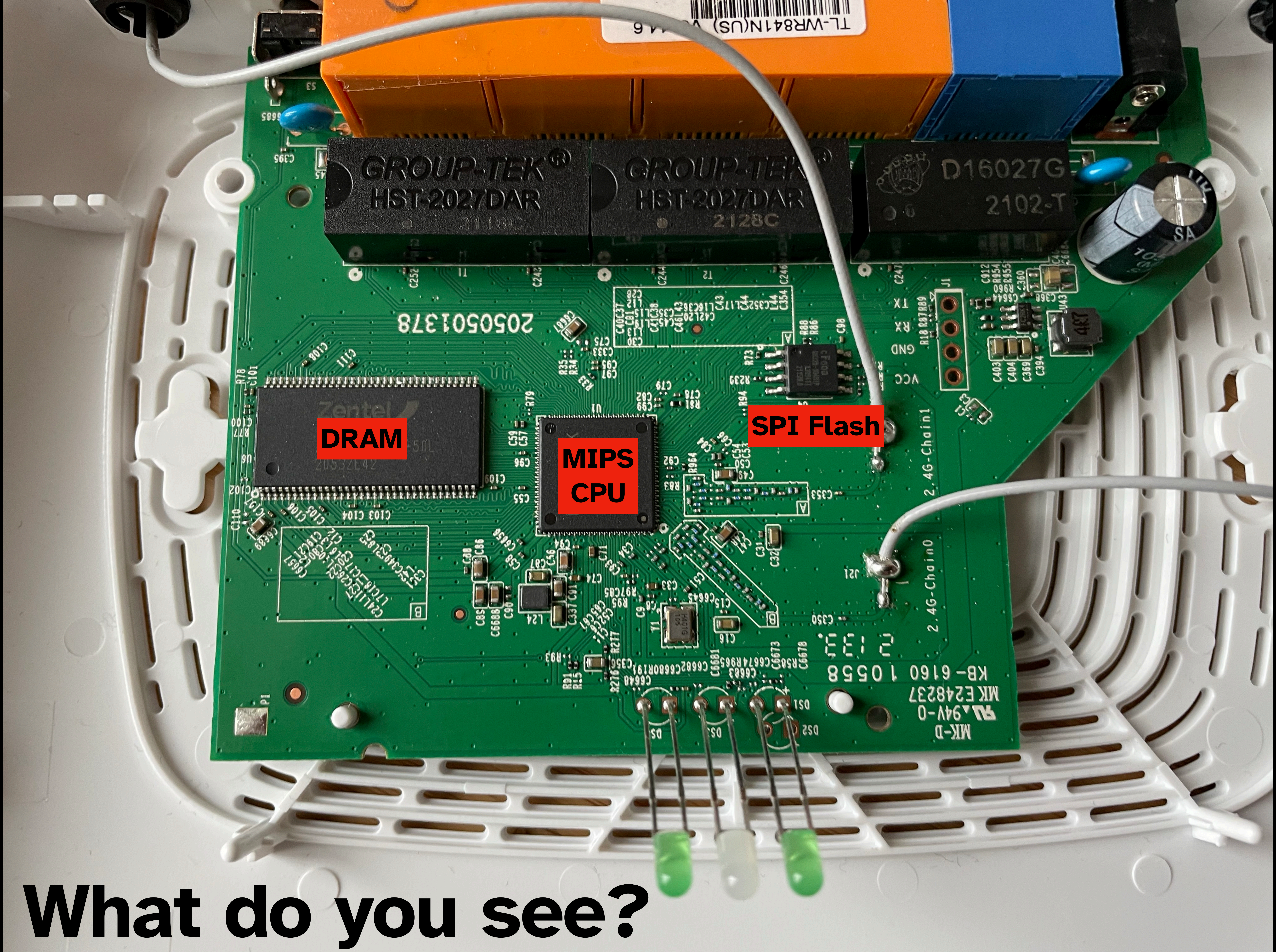
Let's find out.



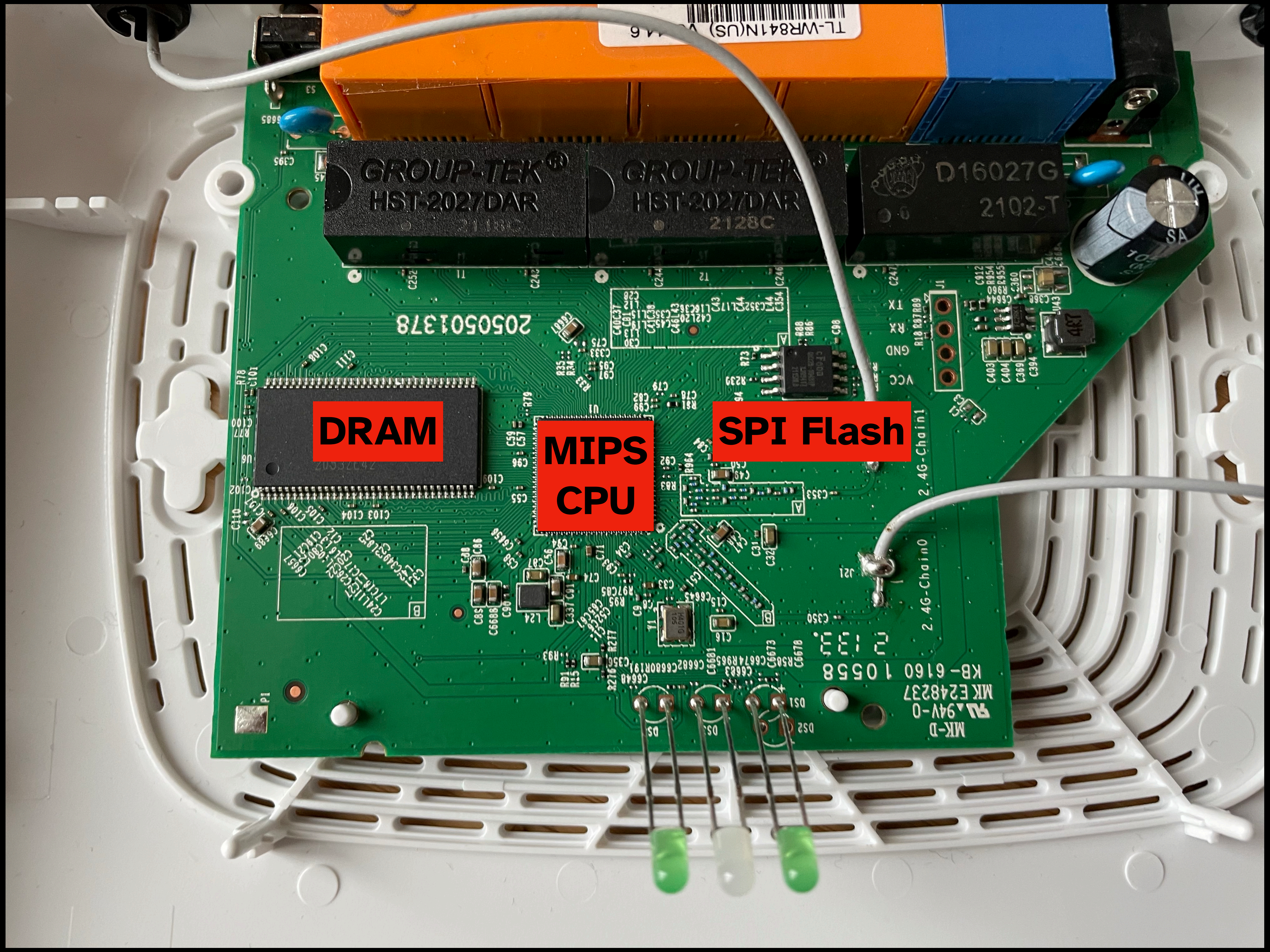




What do you see?



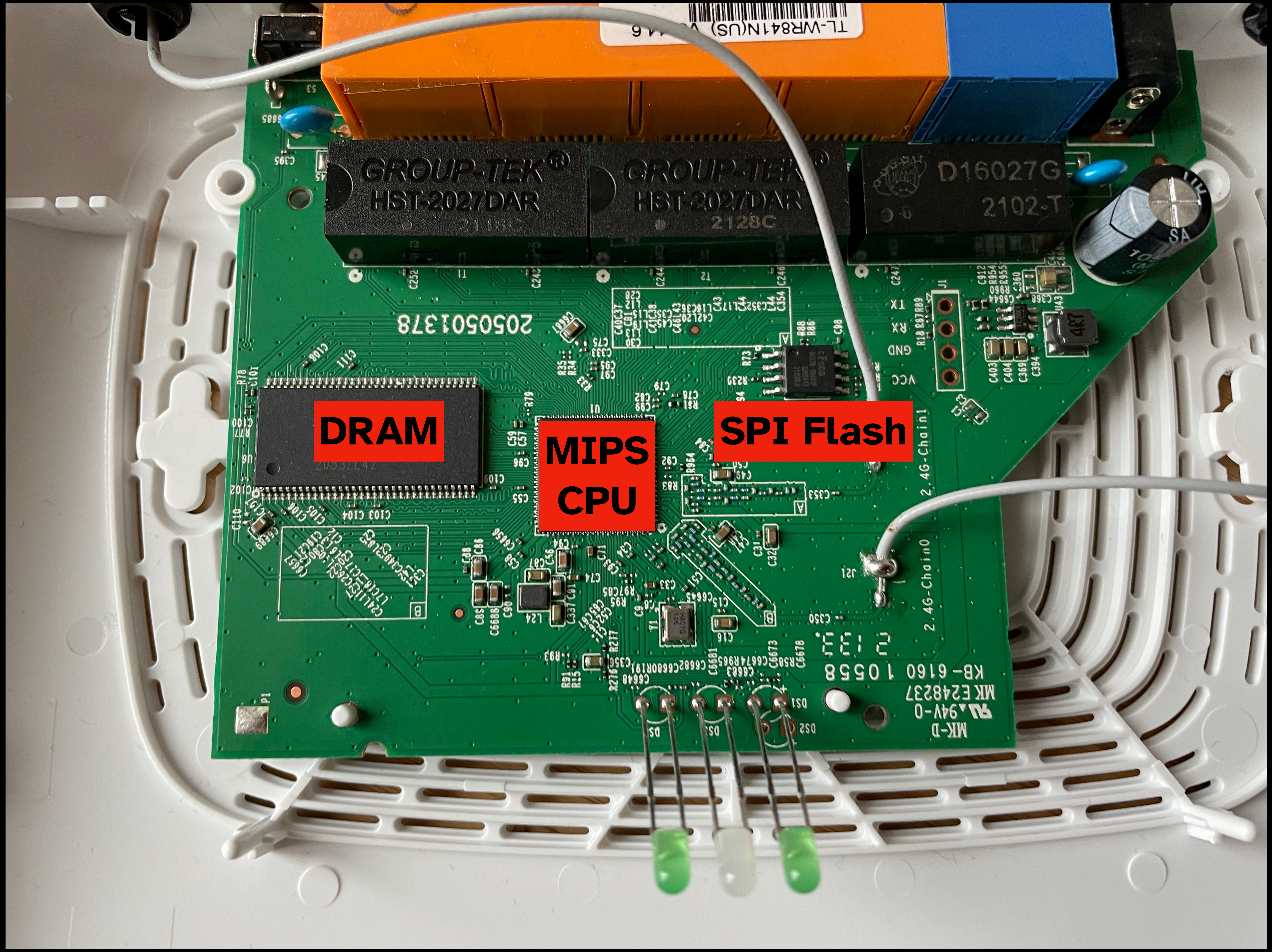
What do you see?



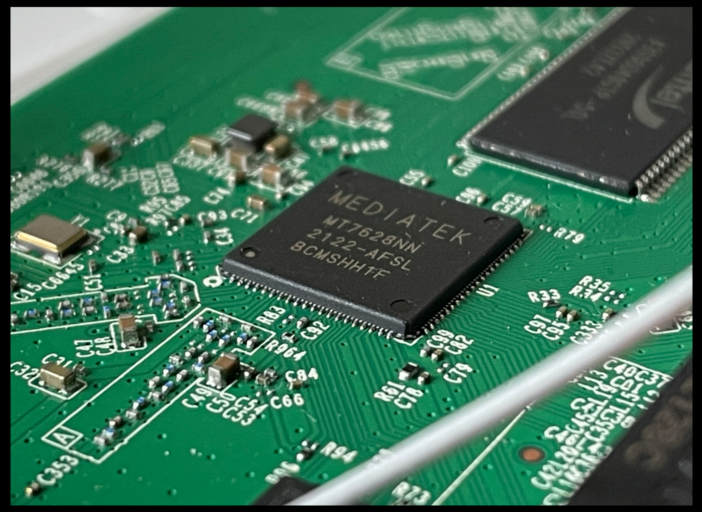
Processor

Memory

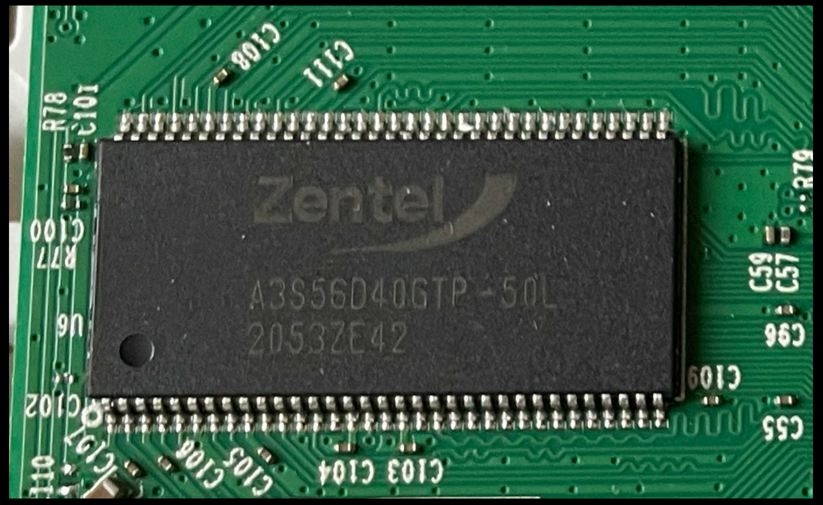
Storage



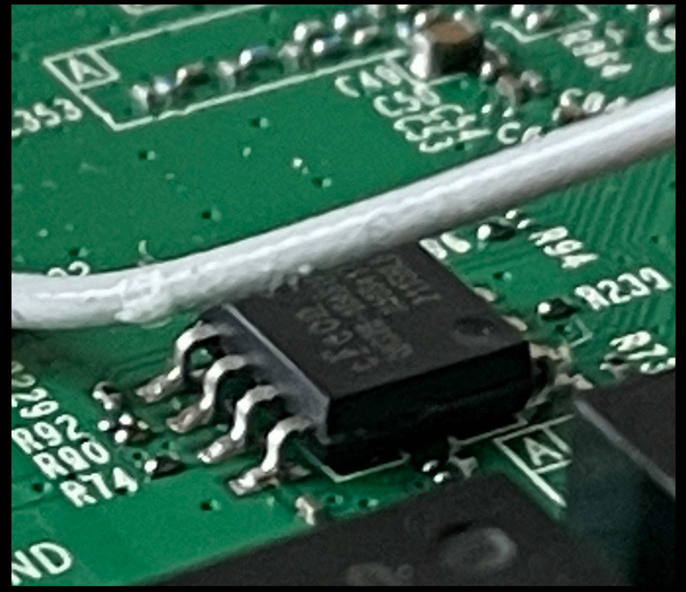
Processor

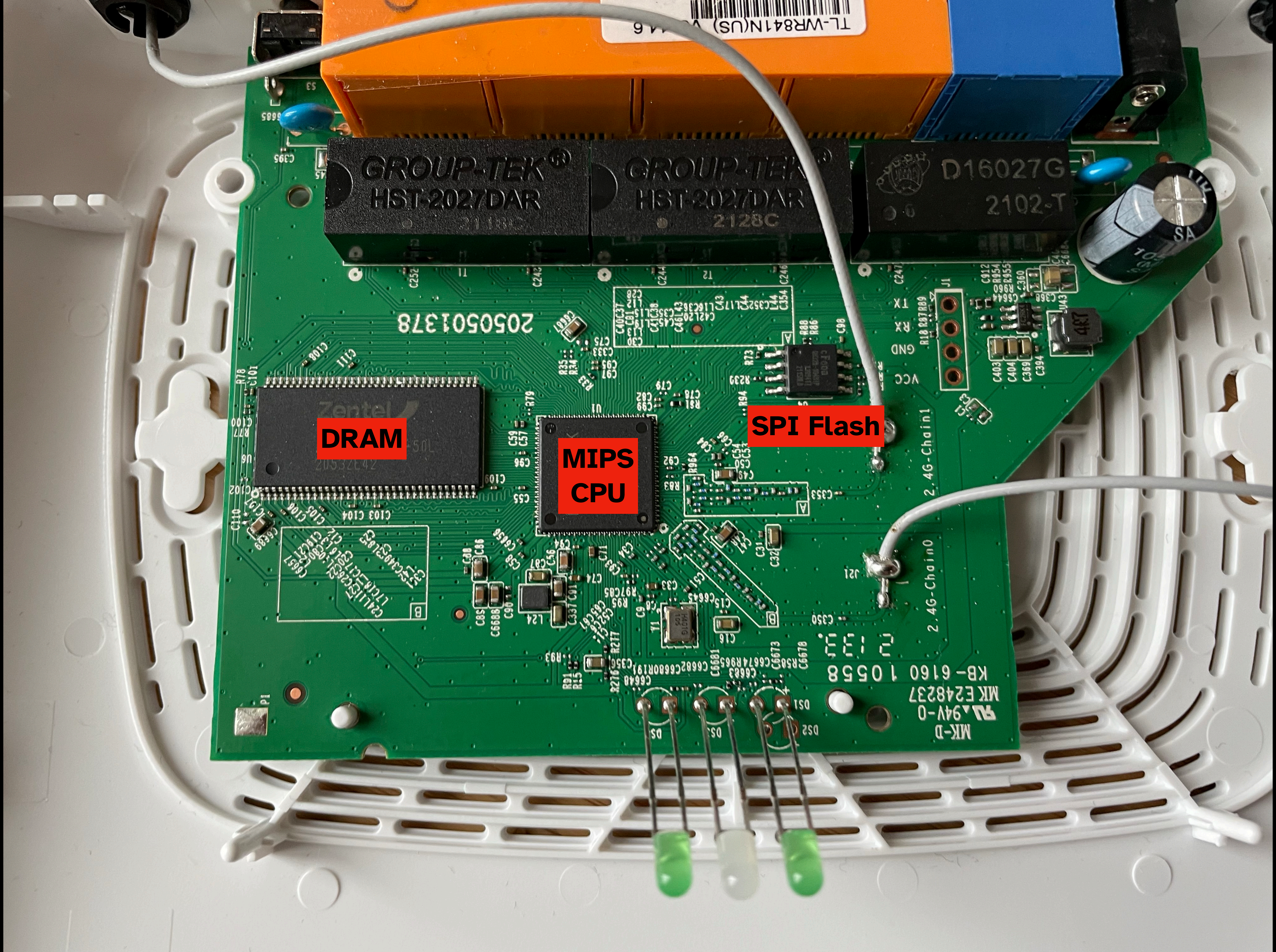


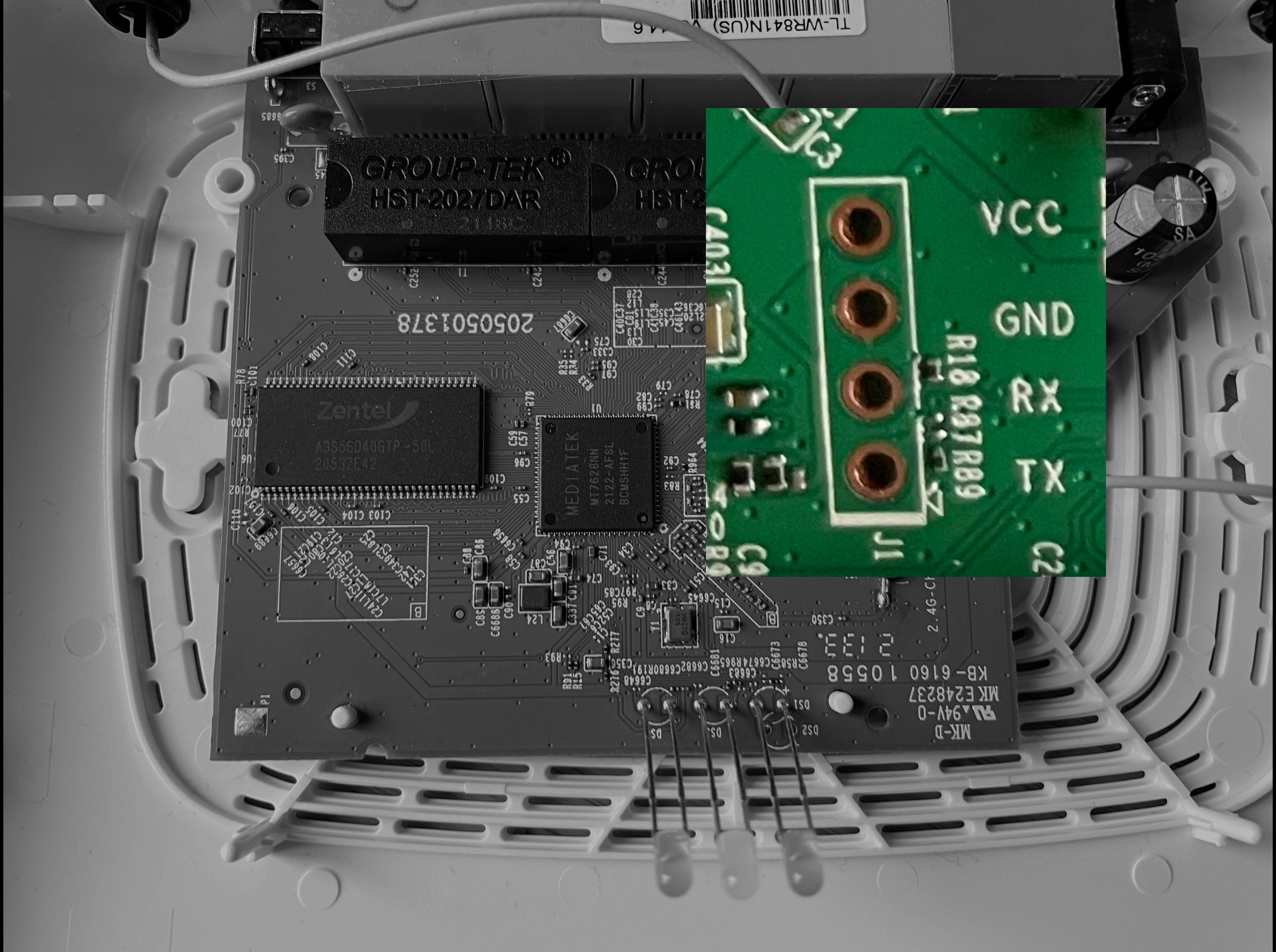
Memory



Storage




DRAM
MIPS CPU
SPI Flash
GROUP-TEK
HST-2027DAR
D16027G
2102-T
Zentel
2050501378


TL-WR841N(US)
GROUP-TEK
HST-2027DAR
2050501378
Zentel
MEDIATEK
MT7628NN
VCC
GND
RX
TX
J1
KB-6160 10558

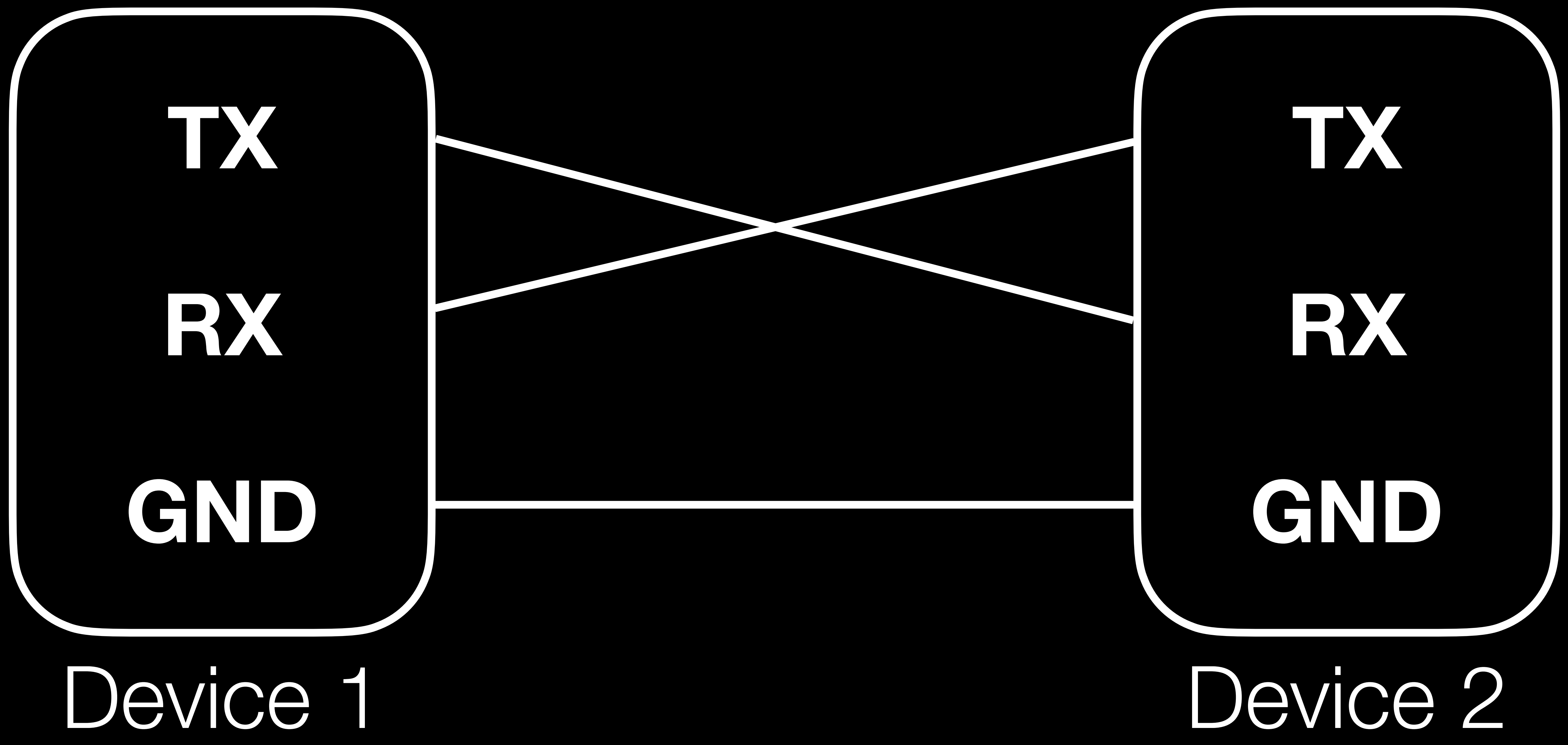

Hmmmm.....
2.4G-Chain1
VCC
GND
RX
TX
R18
R97
R89
R74
R90
R92
C98
C1
C3
C403
C404
C369

Demo 1

"What if the vendor just leaves the backdoor open?"

UART

Universal Asynchronous Receiver/ Transmitter



What other interfaces are out there?

UART/USART

Serial Protocol, a lot of the times just gives a root shell for free

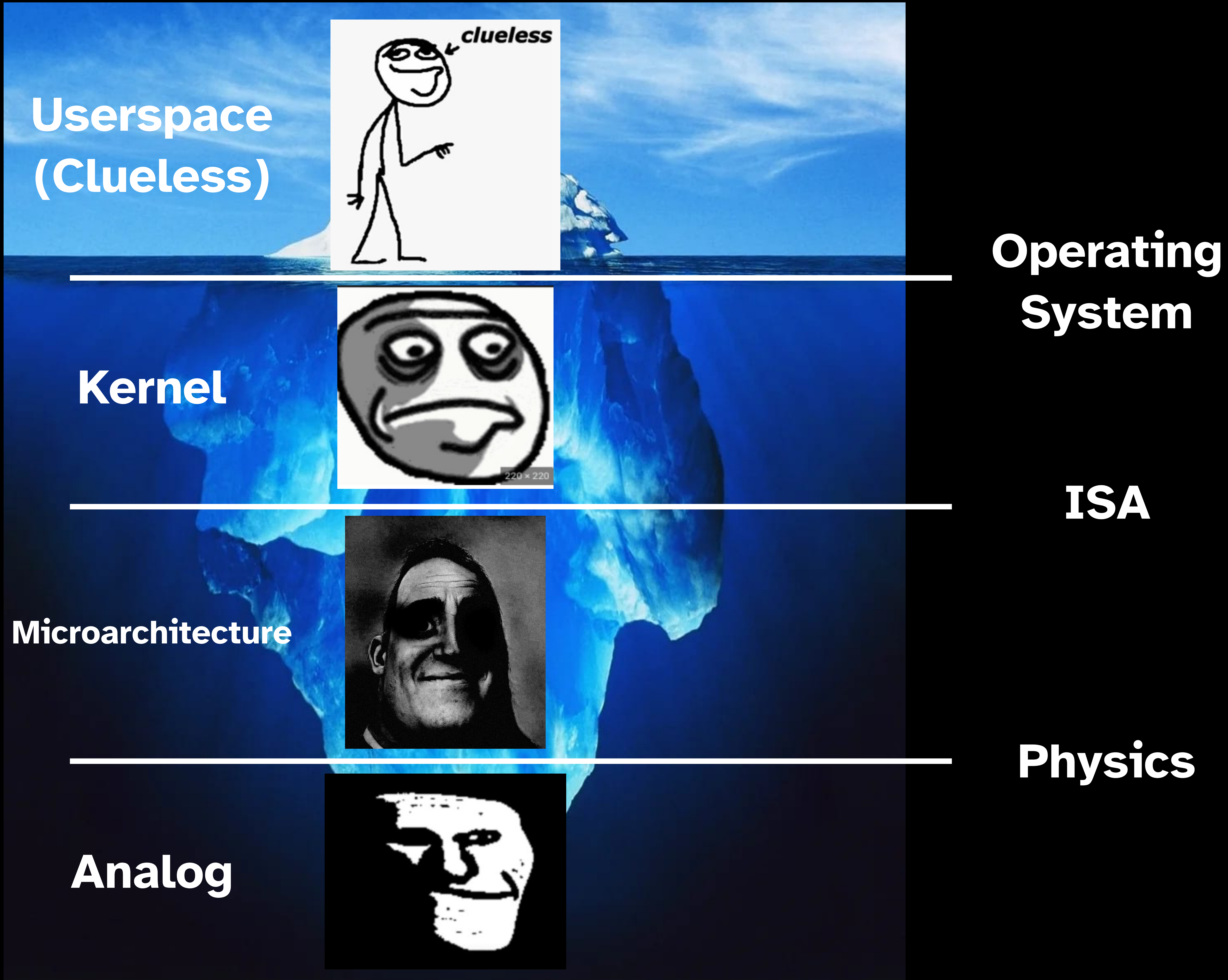
JTAG/ SWD

Dump firmware, debug CPU, upload your own firmware

I2C/ SPI

Protocol used to let chips talk to each other. PC BIOS uses SPI.

The HW Security Iceberg



Userspace
(Clueless)
clueless
Kernel
Microarchitecture
Analog
Operating
System
ISA
Physics

Active

Inject new signals

Modify existing signals in new ways

Passive

No modification of signals

Only observe regular operation

Expensive
Focused Ion Beam
Optical Imaging
Laser Etching
Non-Invasive
Invasive
IC Decapping
Power Tracing
Fault Injection
Logical Interfaces
Cheap

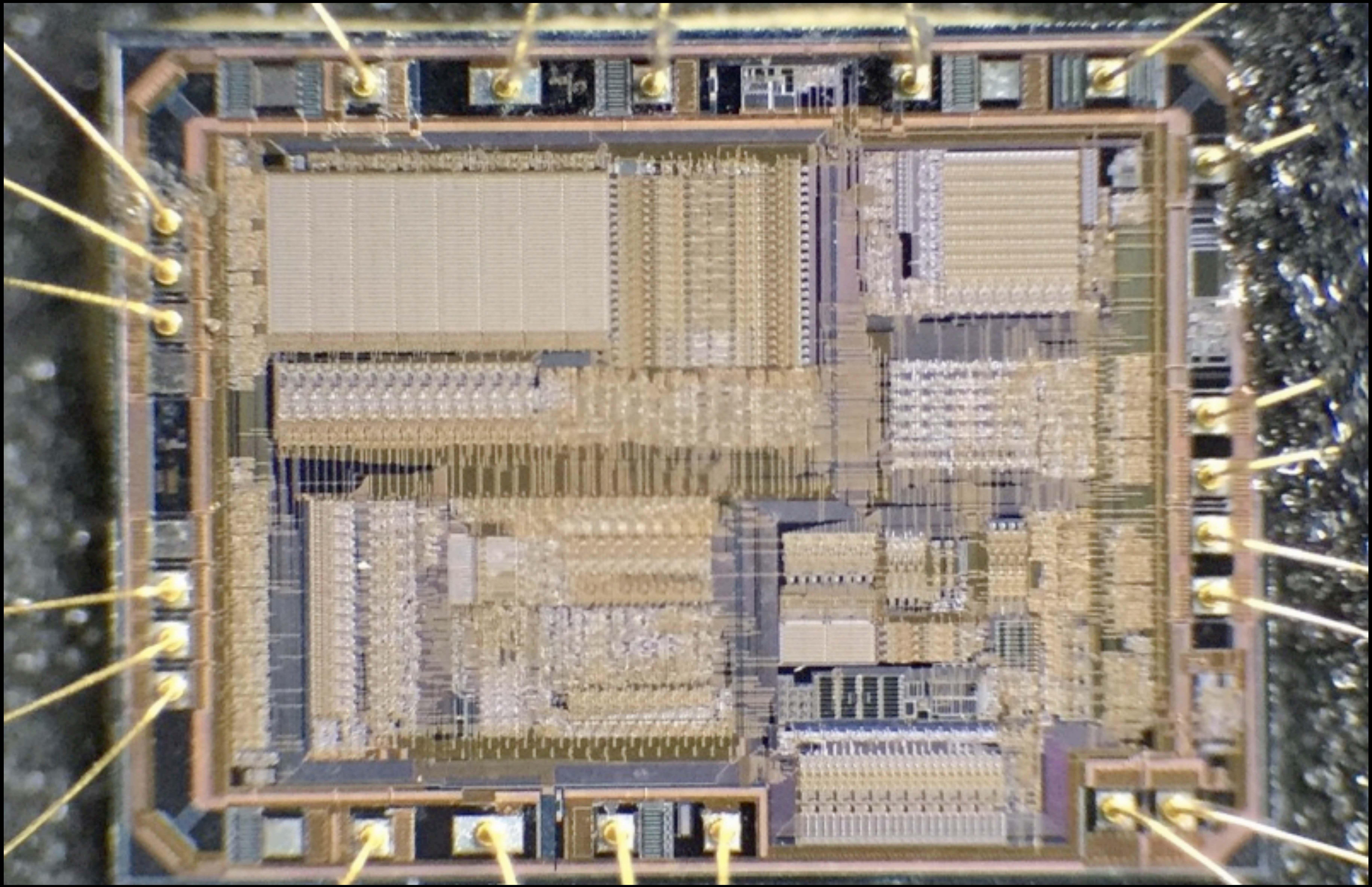


Image: Hackaday

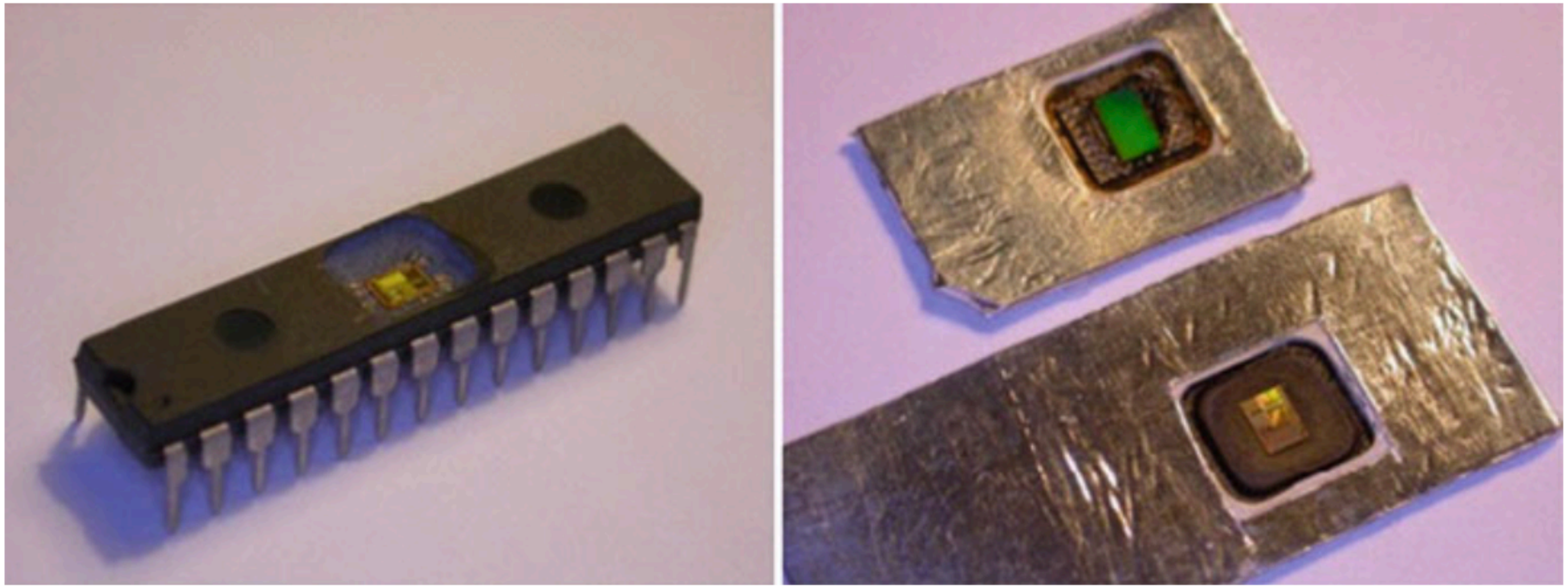
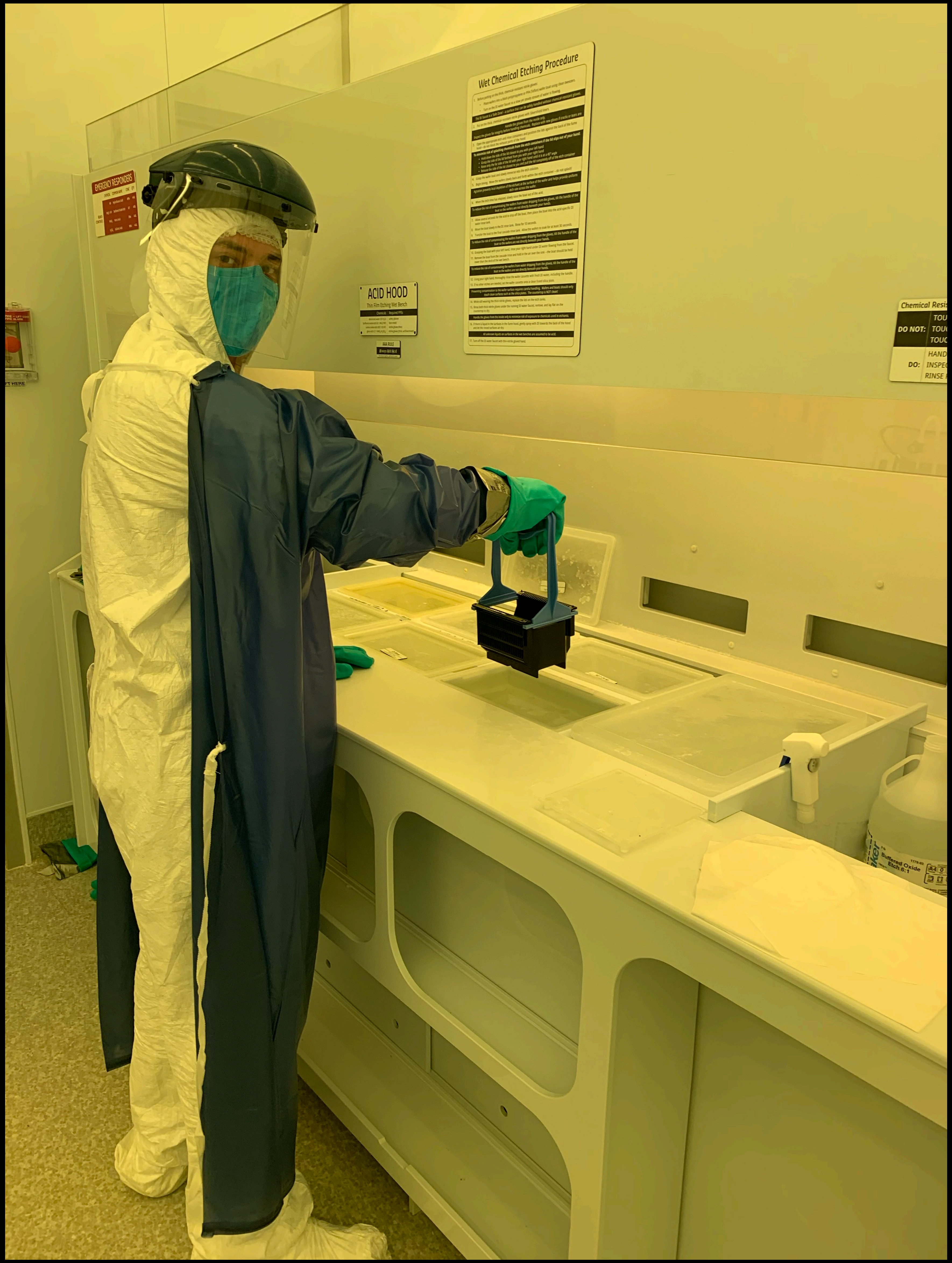


Fig. 7.3 Decapsulated chips


Wet Chemical Etching Procedure
ACID HOOD
DO NOT:
DO:

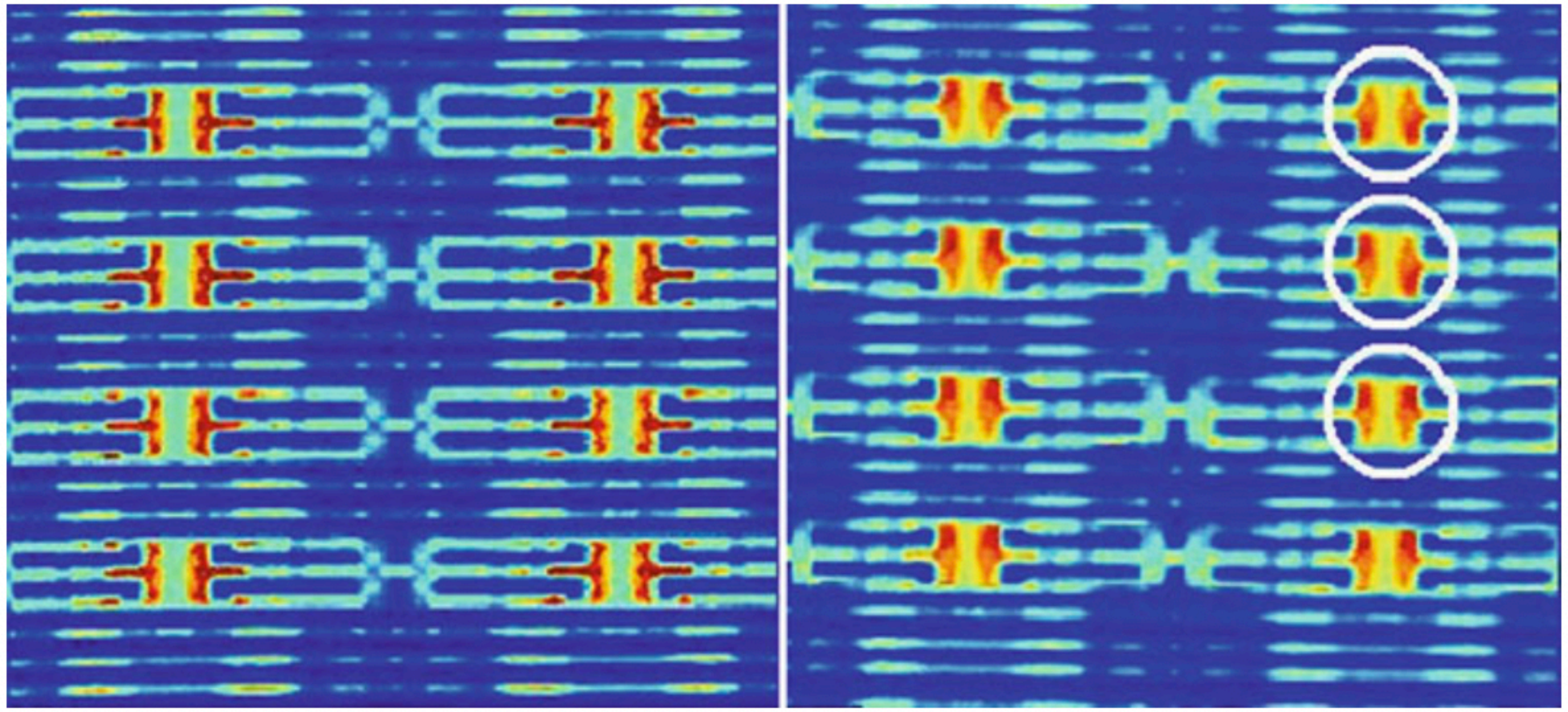


Fig. 7.6 Laser scan of unpowered and powered-up SRAM in PIC16F84 microcontroller

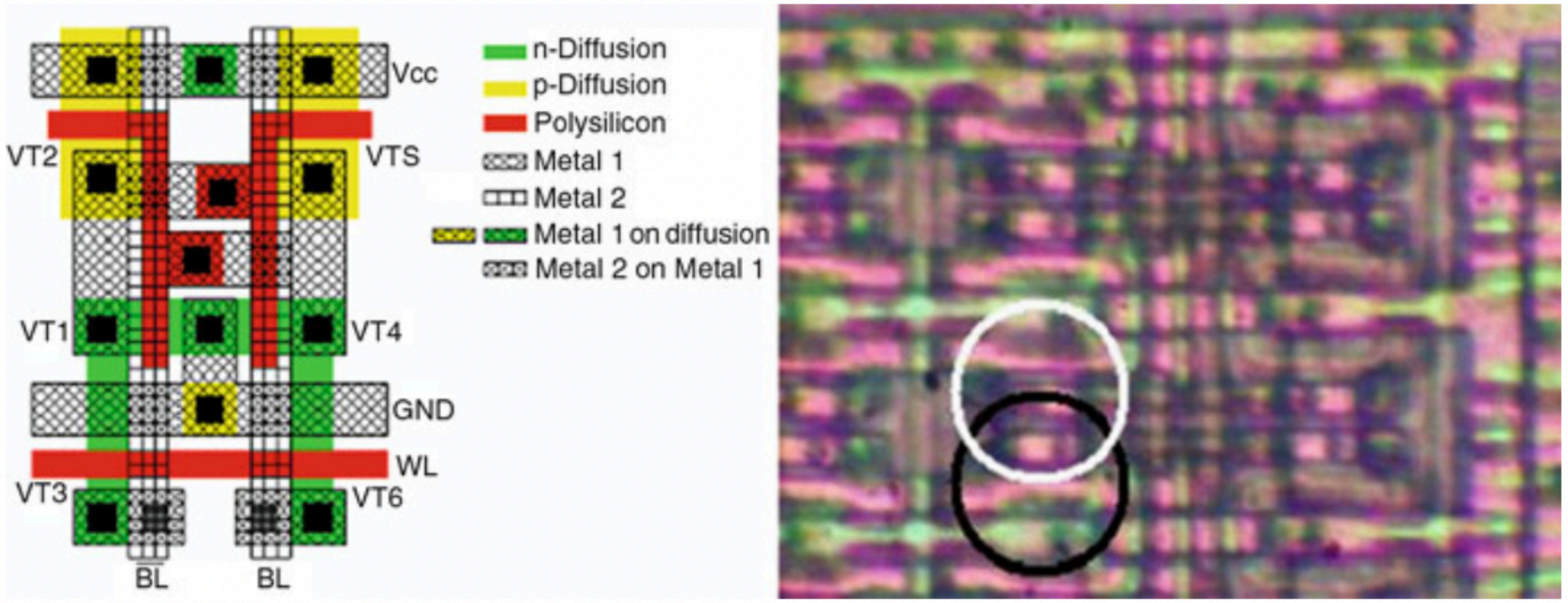


Fig. 7.7 Layout of SRAM cell and SRAM area in PIC16F84 microcontroller

4 Attacks

in this class.

Active

Fault Injection

UART

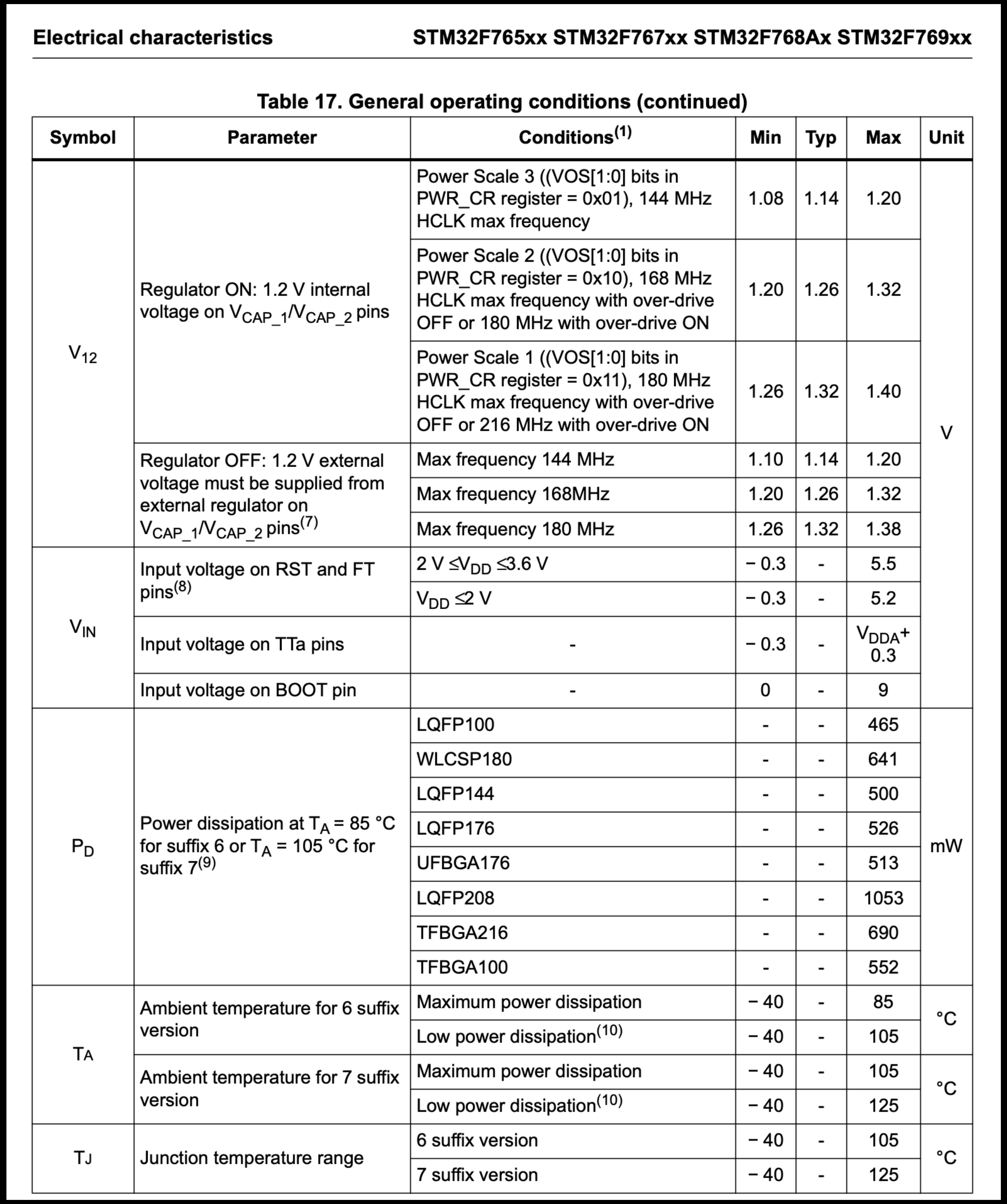
Passive

Power Analysis

Timing Analysis

Fault Injection

Chips have strict operating conditions



Electrical characteristics — STM32F765xx STM32F767xx STM32F768Ax STM32F769xx

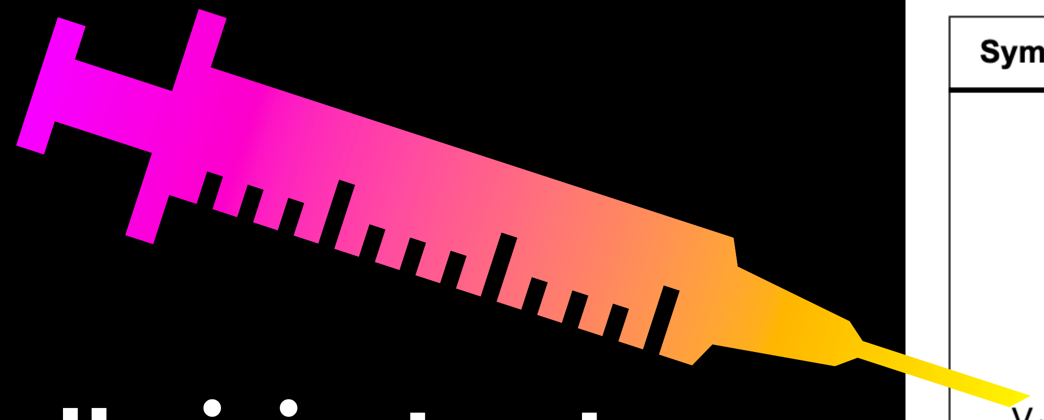
Table 17. General operating conditions (continued)

Symbol	Parameter	Conditions(1)	Min	Typ	Max	Unit
V_{12}	Regulator ON: 1.2 V internal voltage on V_{CAP_1}/V_{CAP_2} pins	Power Scale 3 ((VOS[1:0] bits in PWR_CR register = 0x01), 144 MHz HCLK max frequency	1.08	1.14	1.20	V
		Power Scale 2 ((VOS[1:0] bits in PWR_CR register = 0x10), 168 MHz HCLK max frequency with over-drive OFF or 180 MHz with over-drive ON	1.20	1.26	1.32	
		Power Scale 1 ((VOS[1:0] bits in PWR_CR register = 0x11), 180 MHz HCLK max frequency with over-drive OFF or 216 MHz with over-drive ON	1.26	1.32	1.40	
	Regulator OFF: 1.2 V external voltage must be supplied from external regulator on V_{CAP_1}/V_{CAP_2} pins(7)	Max frequency 144 MHz	1.10	1.14	1.20	
		Max frequency 168MHz	1.20	1.26	1.32	
		Max frequency 180 MHz	1.26	1.32	1.38	
V_{IN}	Input voltage on RST and FT pins(8)	2 V $\leq V_{DD} \leq$ 3.6 V	− 0.3	-	5.5	
		$V_{DD} \leq$ 2 V	− 0.3	-	5.2	
	Input voltage on TTa pins	-	− 0.3	-	V_{DDA}+ 0.3	
	Input voltage on BOOT pin	-	0	-	9	
P_D	Power dissipation at T_A = 85 °C for suffix 6 or T_A = 105 °C for suffix 7(9)	LQFP100	-	-	465	mW
		WLCSP180	-	-	641	
		LQFP144	-	-	500	
		LQFP176	-	-	526	
		UFBGA176	-	-	513	
		LQFP208	-	-	1053	
		TFBGA216	-	-	690	
		TFBGA100	-	-	552	
T_A	Ambient temperature for 6 suffix version	Maximum power dissipation	− 40	-	85	°C
		Low power dissipation(10)	− 40	-	105	
	Ambient temperature for 7 suffix version	Maximum power dissipation	− 40	-	105	°C
		Low power dissipation(10)	− 40	-	125	
T_J	Junction temperature range	6 suffix version	− 40	-	105	°C
		7 suffix version	− 40	-	125	

"Datasheet"

STMicroelectronics. STM32F767ZI Datasheet.

Chips have strict operating conditions



Intentionally inject out-of-specification inputs to (hopefully) break the chip

Table 17. General operating conditions (continued)

Symbol	Parameter	Conditions[1]	Min	Typ	Max	Unit
V_{12}	Regulator ON: 1.2 V internal voltage on V_{CAP_1}/V_{CAP_2} pins	Power Scale 3 ((VOS[1:0] bits in PWR_CR register = 0x01), 144 MHz HCLK max frequency	1.08	1.14	1.20	V
		Power Scale 2 ((VOS[1:0] bits in PWR_CR register = 0x10), 168 MHz HCLK max frequency with over-drive OFF or 180 MHz with over-drive ON	1.20	1.26	1.32	
		Power Scale 1 ((VOS[1:0] bits in PWR_CR register = 0x11), 180 MHz HCLK max frequency with over-drive OFF or 216 MHz with over-drive ON	1.26	1.32	1.40	
	Regulator OFF: 1.2 V external voltage must be supplied from external regulator on V_{CAP_1}/V_{CAP_2} pins[7]	Max frequency 144 MHz	1.10	1.14	1.20	
		Max frequency 168MHz	1.20	1.26	1.32	
		Max frequency 180 MHz	1.26	1.32	1.38	
V_{IN}	Input voltage on RST and FT pins[8]	$2\ V \leq V_{DD} \leq 3.6\ V$	– 0.3	-	5.5	
		$V_{DD} \leq 2\ V$	– 0.3	-	5.2	
	Input voltage on TTa pins	-	– 0.3	-	V_{DDA}+ 0.3	
	Input voltage on BOOT pin	-	0	-	9	
P_D	Power dissipation at T_A = 85 °C for suffix 6 or T_A = 105 °C for suffix 7[9]	LQFP100	-	-	465	mW
		WLCSP180	-	-	641	
		LQFP144	-	-	500	
		LQFP176	-	-	526	
		UFBGA176	-	-	513	
		LQFP208	-	-	1053	
		TFBGA216	-	-	690	
		TFBGA100	-	-	552	
TA	Ambient temperature for 6 suffix version	Maximum power dissipation	– 40	-	85	°C
		Low power dissipation[10]	– 40	-	105	
	Ambient temperature for 7 suffix version	Maximum power dissipation	– 40	-	105	°C
		Low power dissipation[10]	– 40	-	125	
TJ	Junction temperature range	6 suffix version	– 40	-	105	°C
		7 suffix version	– 40	-	125	

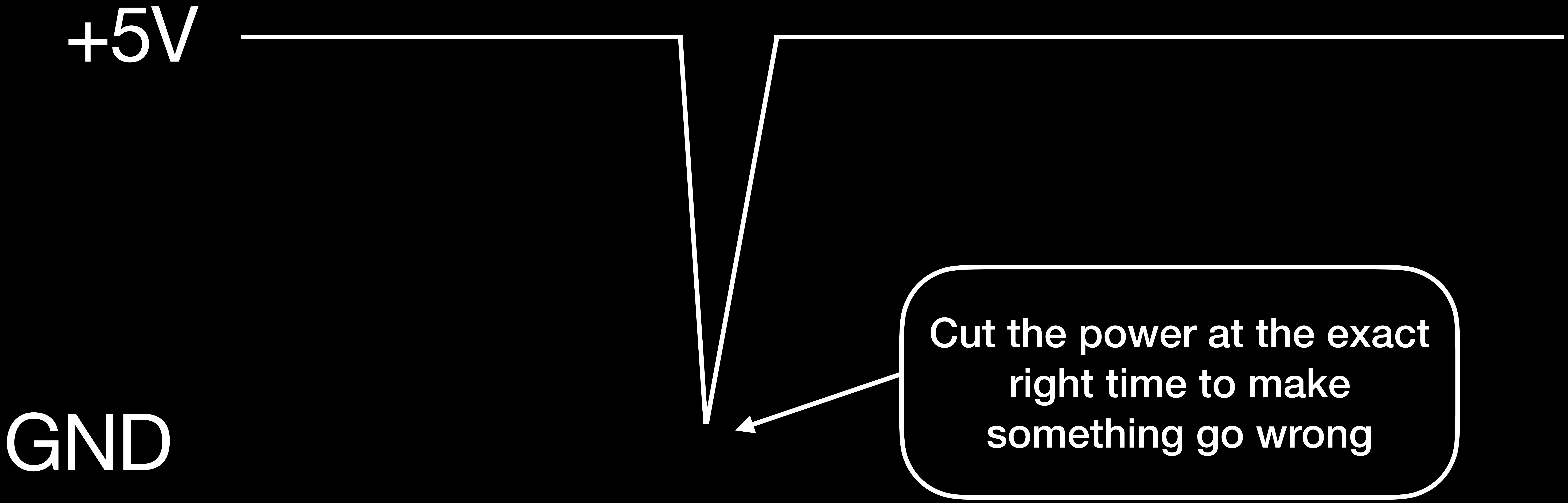
STMicroelectronics. STM32F767ZI Datasheet.

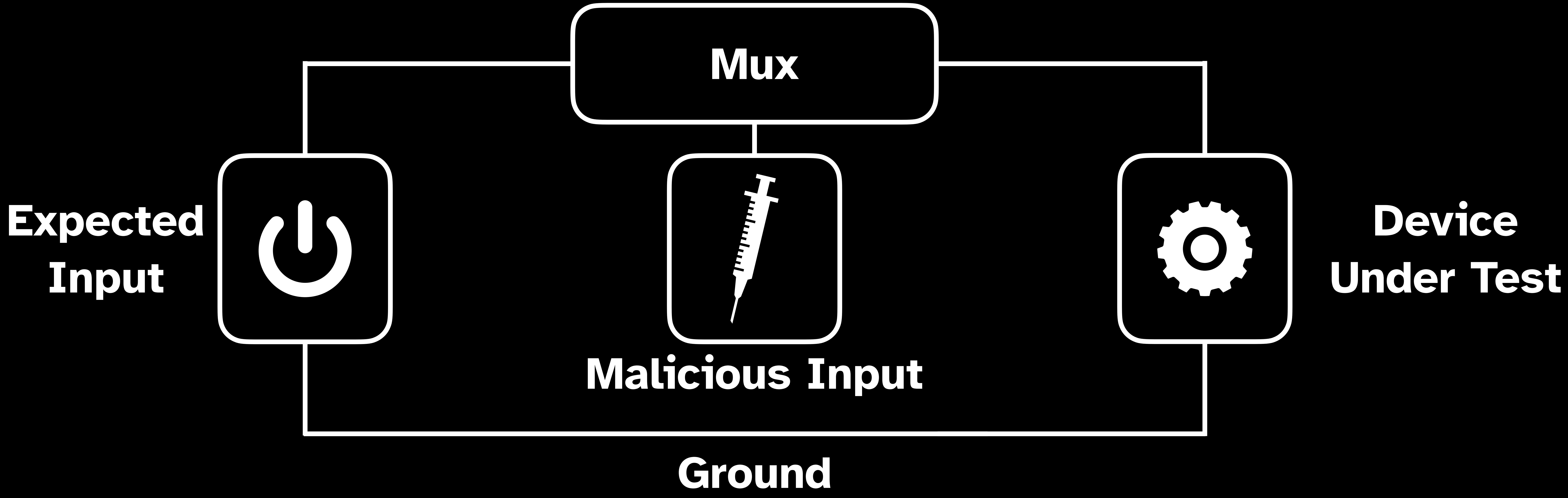
Normal Input Voltage (Vcc)

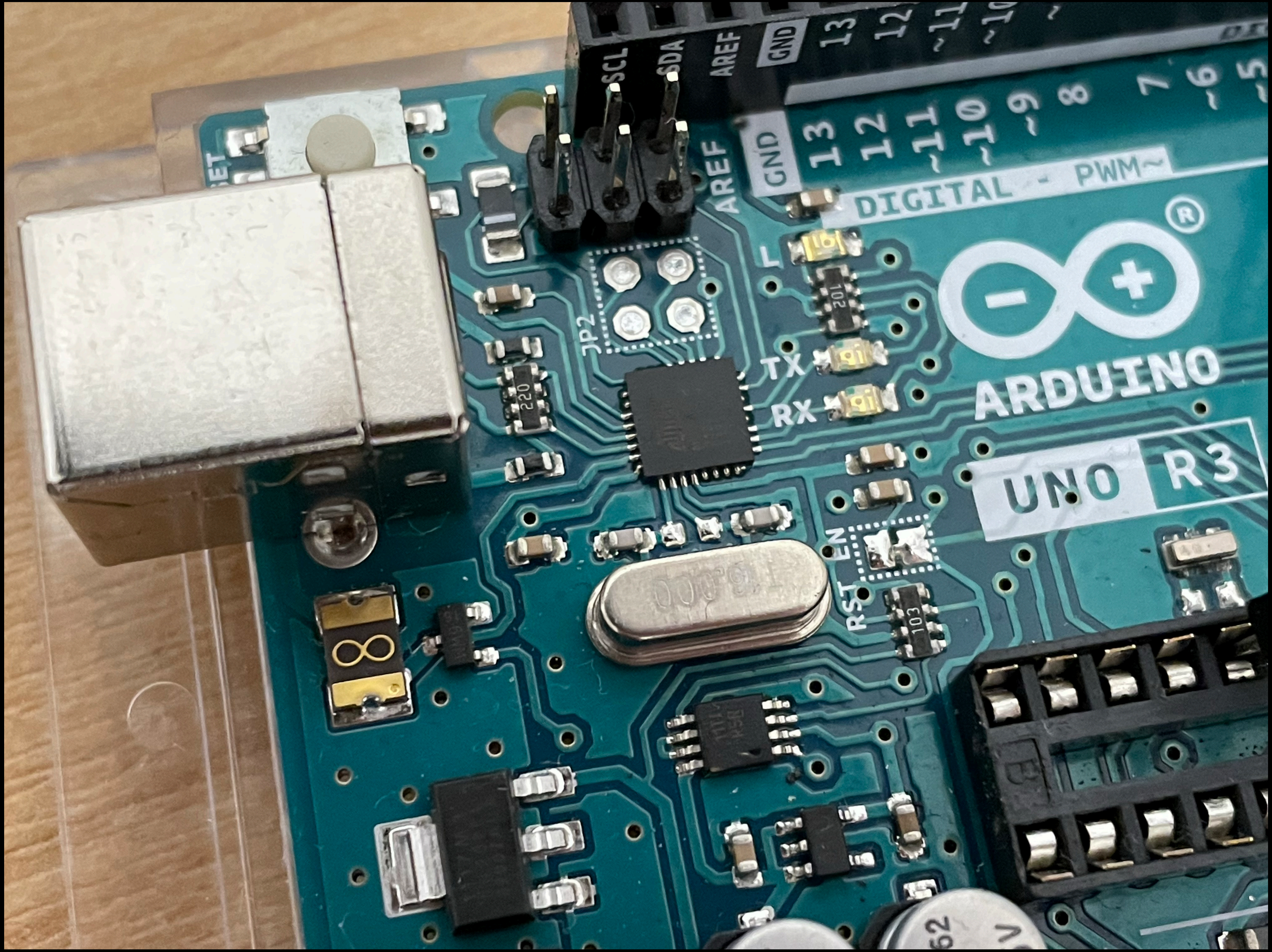
+5V

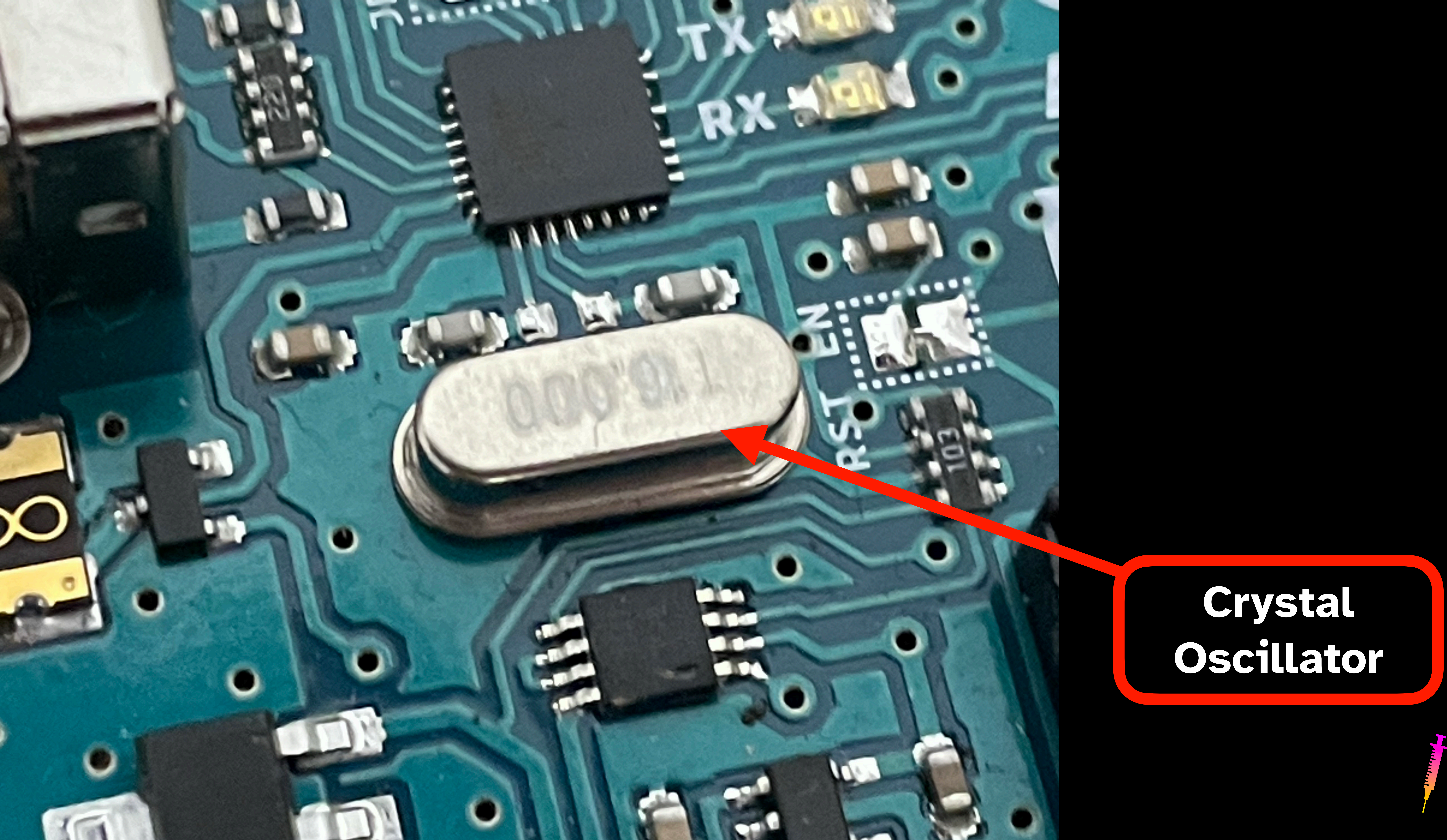
GND

Voltage Glitching

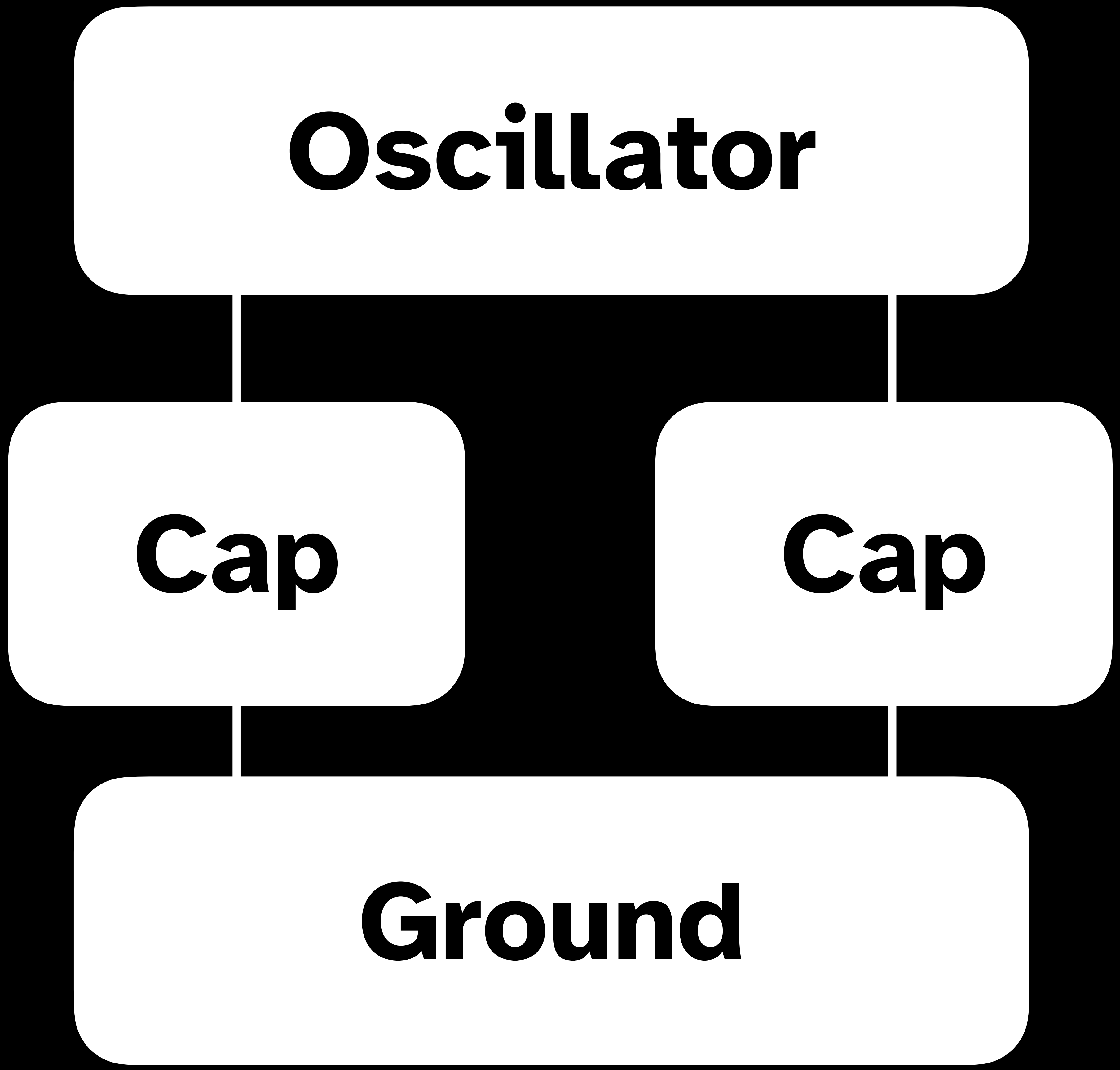


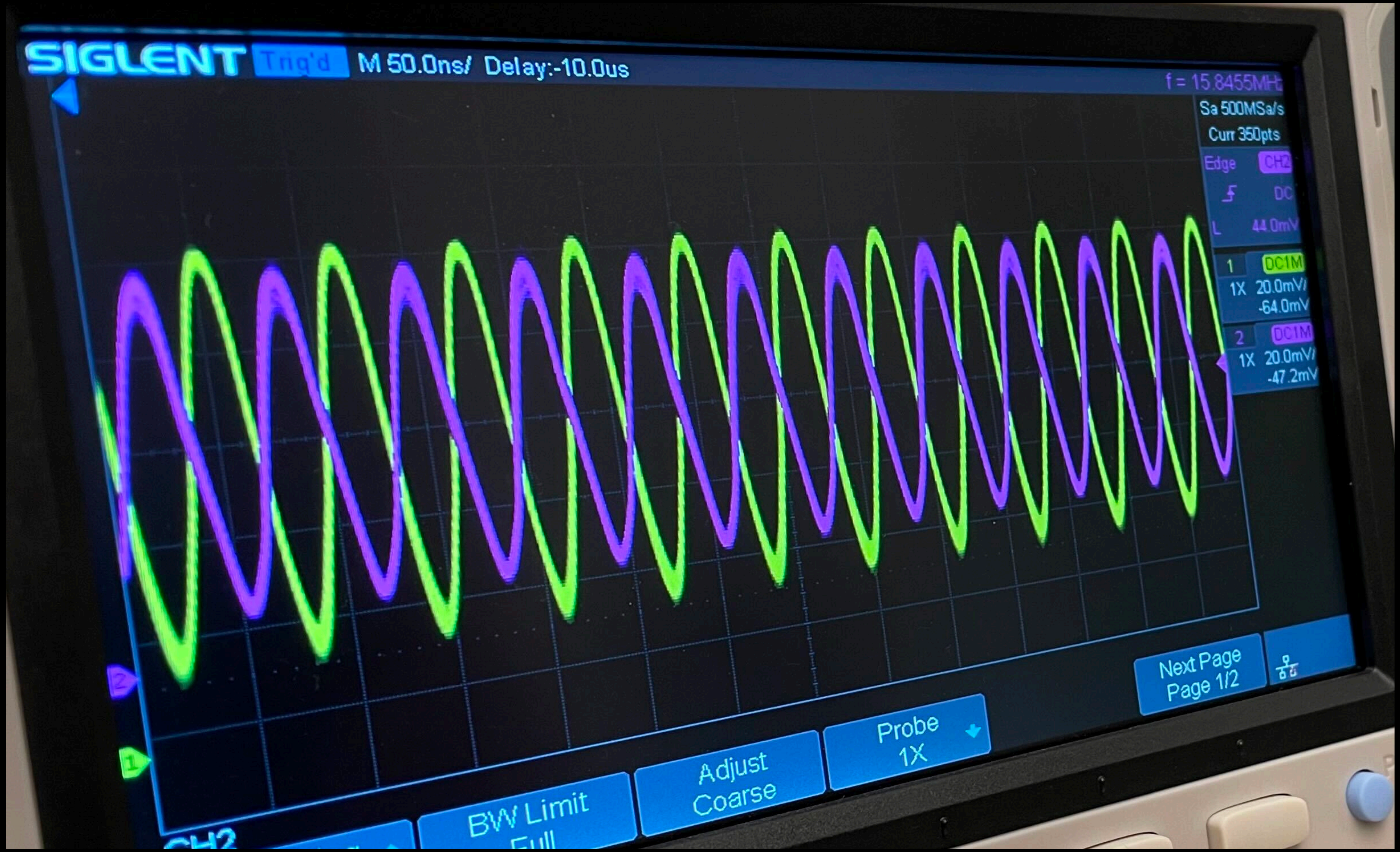

Mux
Expected
Input
Device
Under Test
Malicious Input
Ground


SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
DIGITAL - PWM~
L
TX
RX
JP2
ARDUINO
UNO
R3
EN
RST
220
102
103


TX
RX
EN
RST
Crystal Oscillator

Clock Glitching





Crystal Oscillator

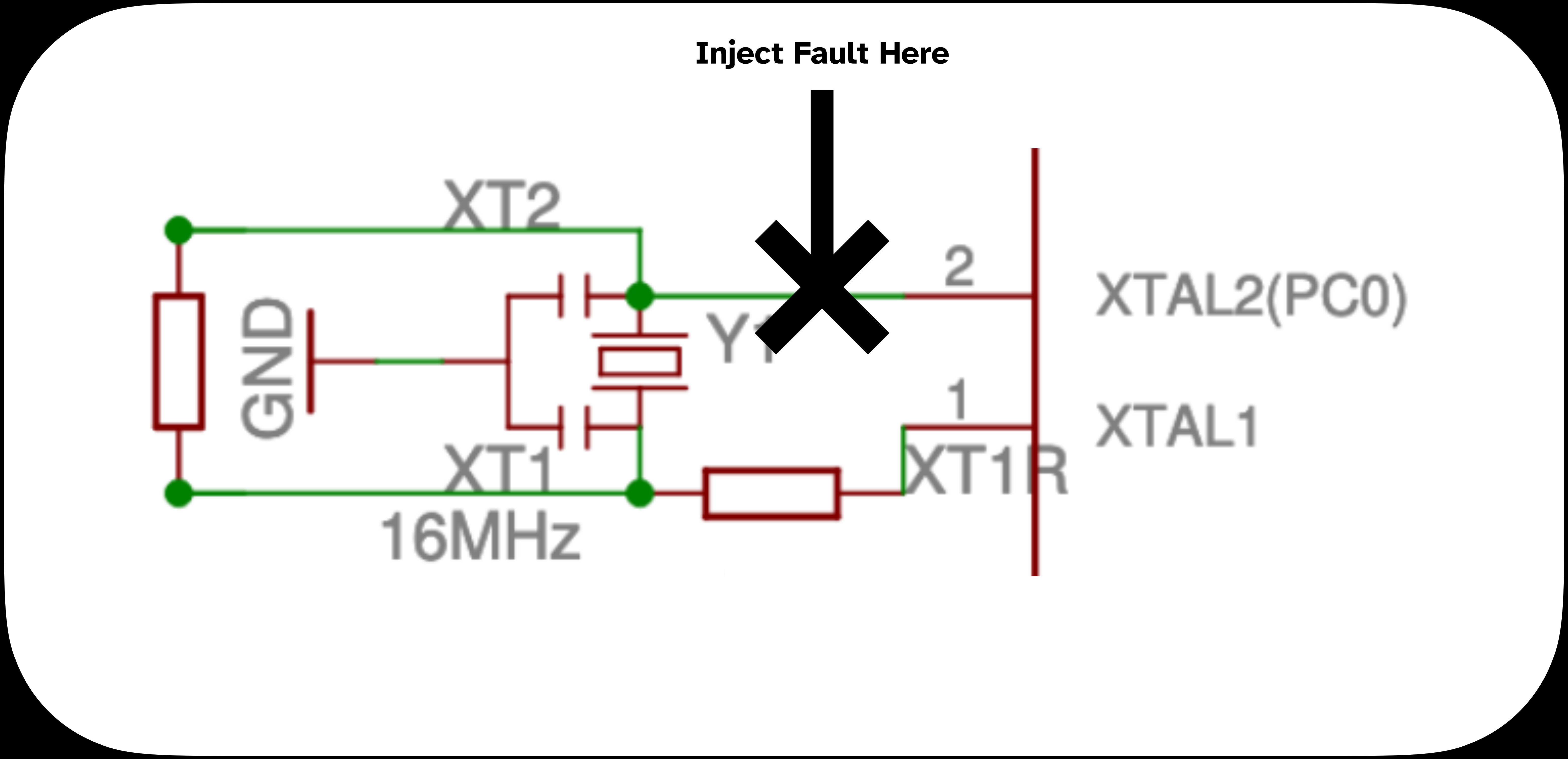
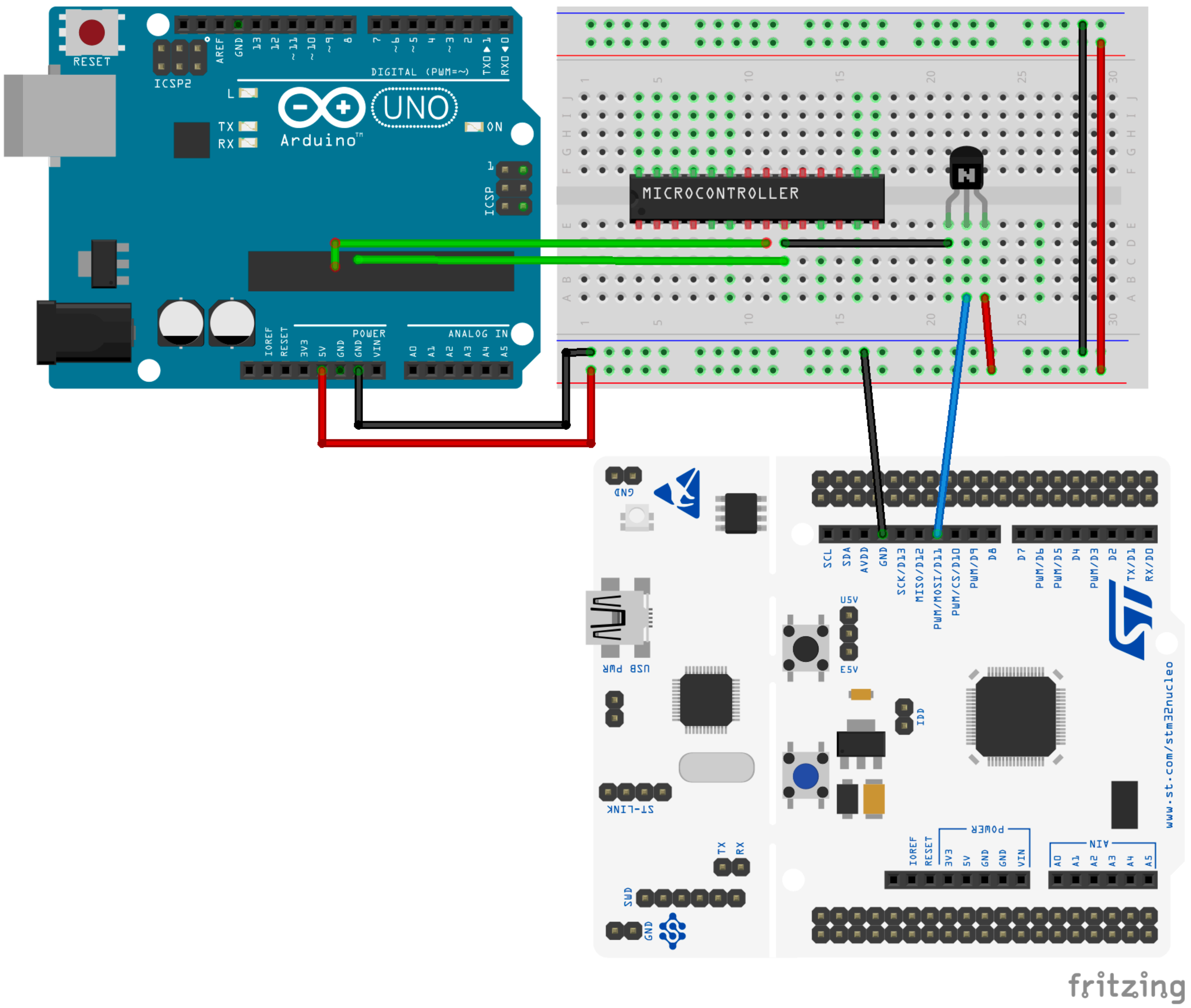
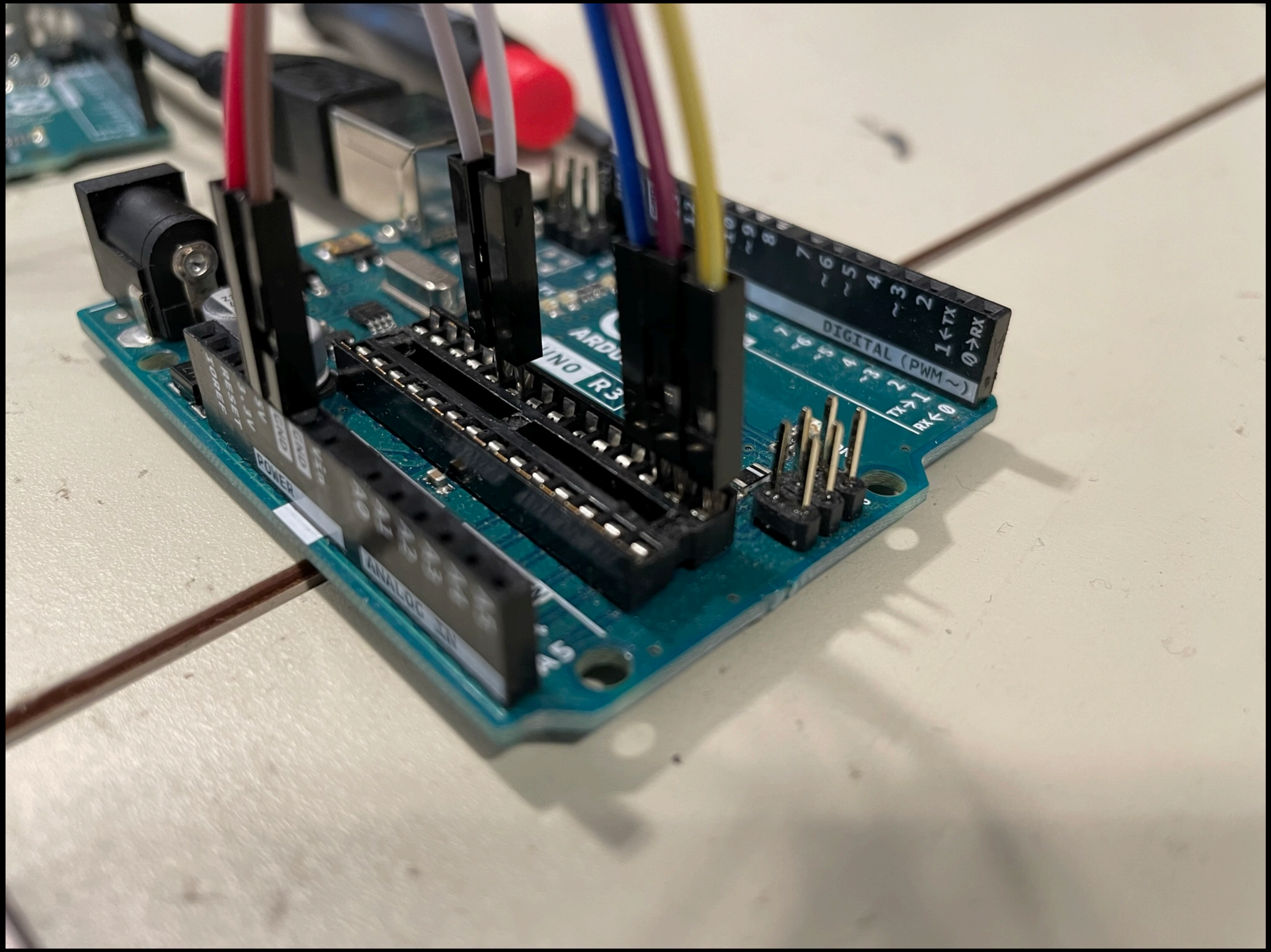
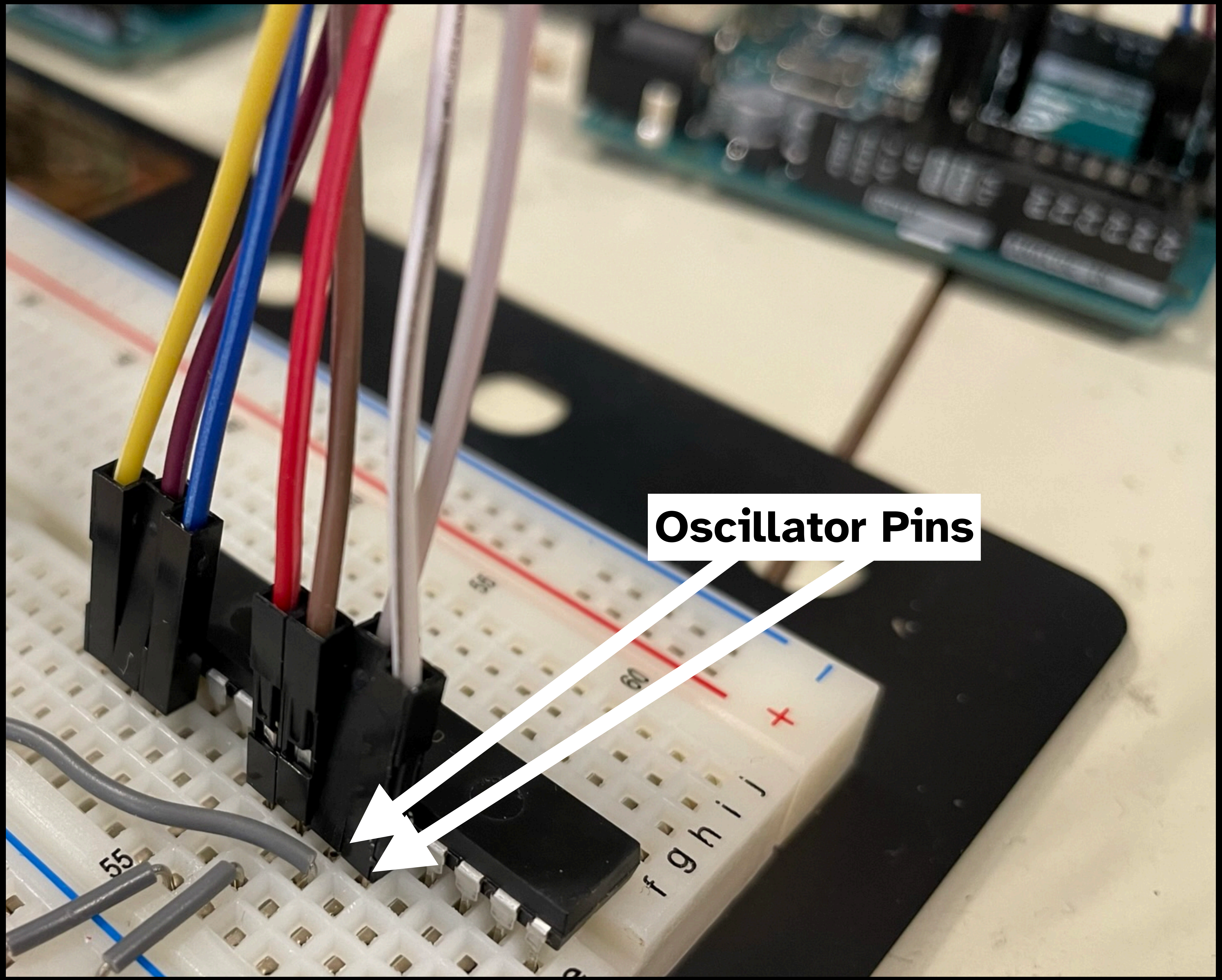
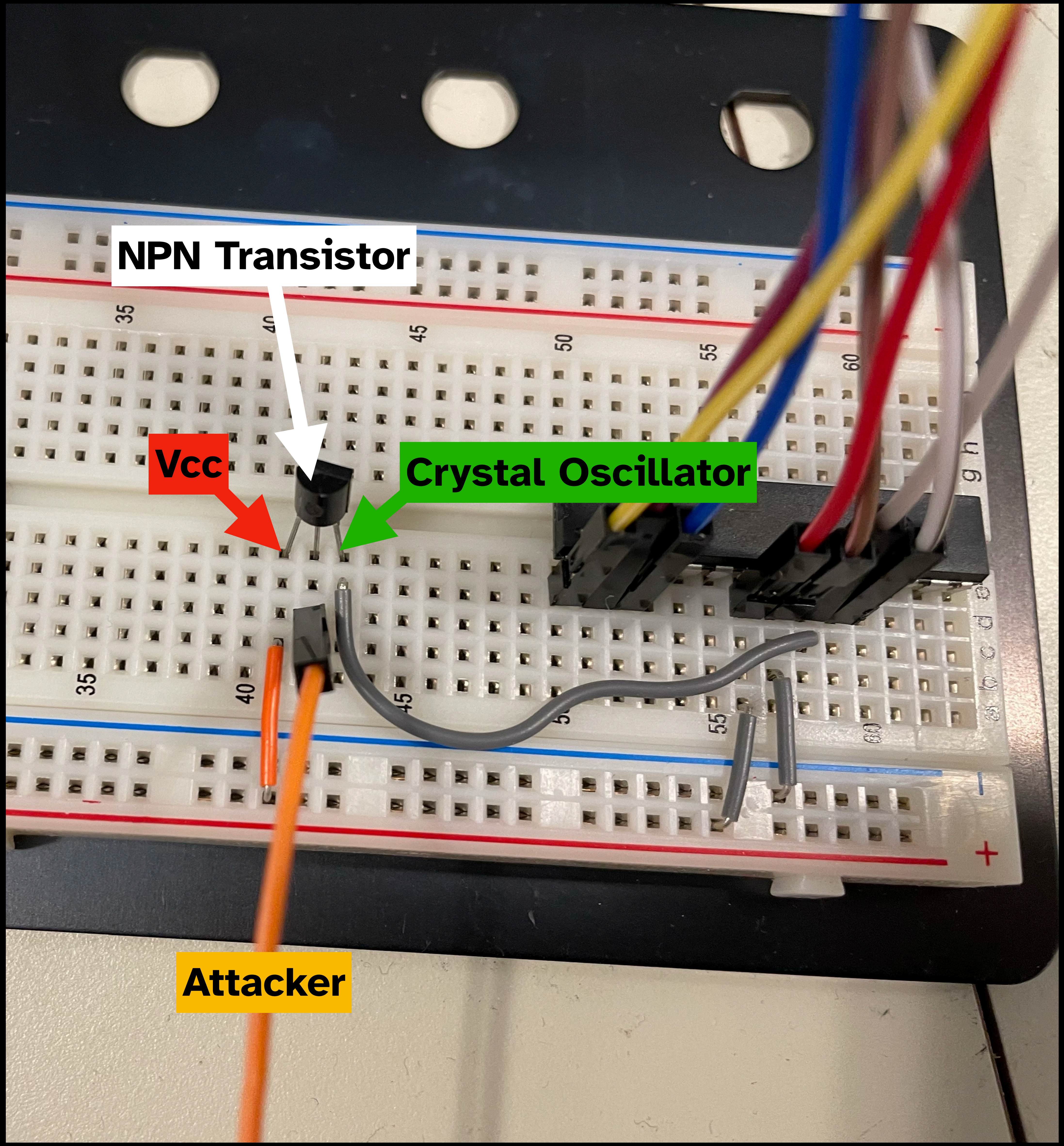


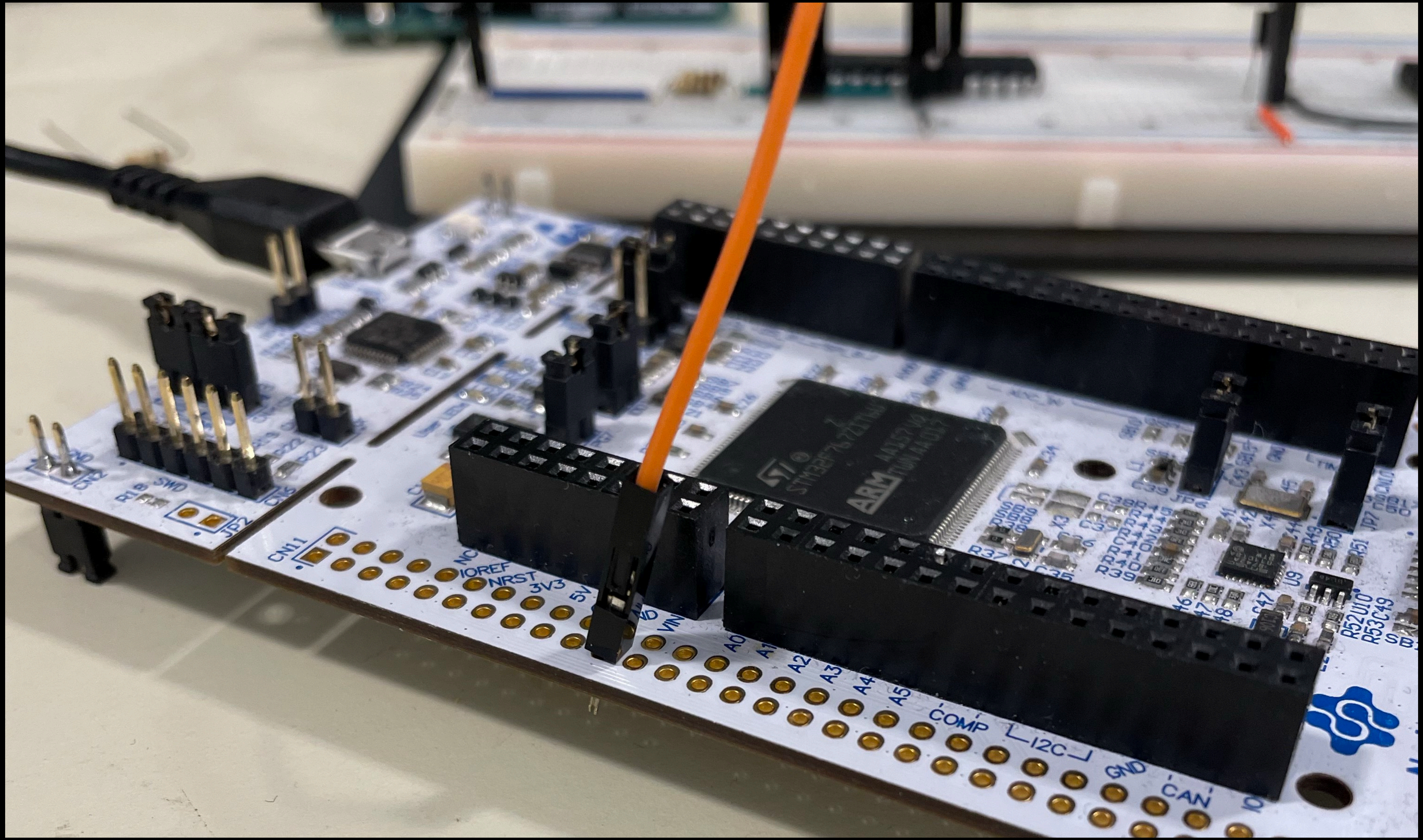
Image: Arduino Uno R3 Reference Design


RESET
ICSP2
UNO
Arduino
DIGITAL (PWM=~)
POWER
ANALOG IN
ON
ICSP
MICROCONTROLLER
fritzing




Oscillator Pins


NPN Transistor
Vcc
Crystal Oscillator
Attacker


SWD
CN11
NC
IOREF
NRST
3V3
5V
VIN
A0
A1
A2
A3
A4
A5
COMP
I2C
GND
CAN
ARM

Inject Fault here

```
while(1 == 1) {
    print("Locked! %d", iter);
    iter++;
}
print("MIT{flag}");
```

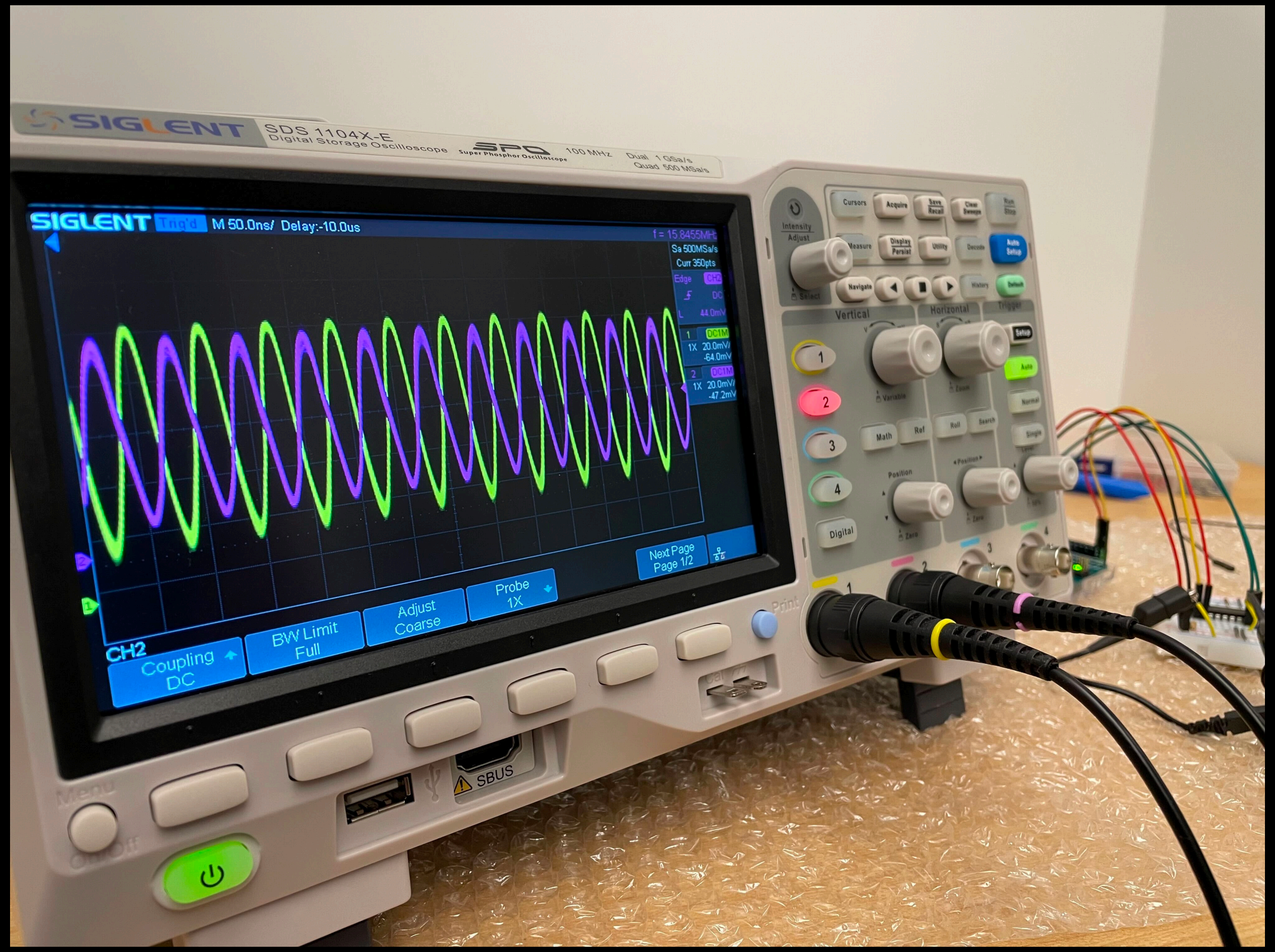
Demo 2

"What if we intentionally violate the chip's expected operating conditions?"

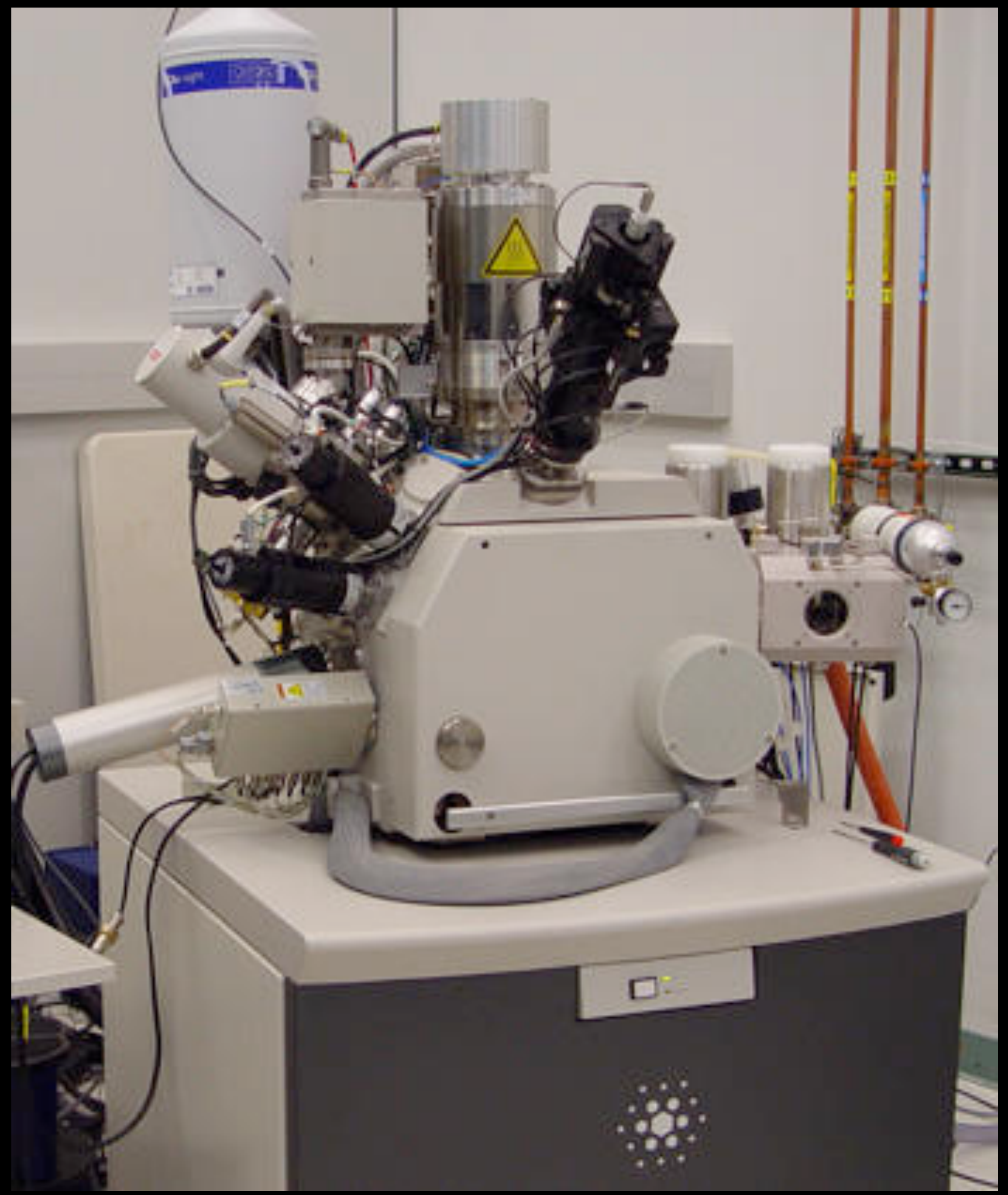
Tools

Cheap

Affordable



Crazy Expensive



Yes, Really

hackaday.com

HACKADAY

HOME BLOG HACKADAY.IO TINDIE HACKADAY PRIZE SUBMIT ABOUT

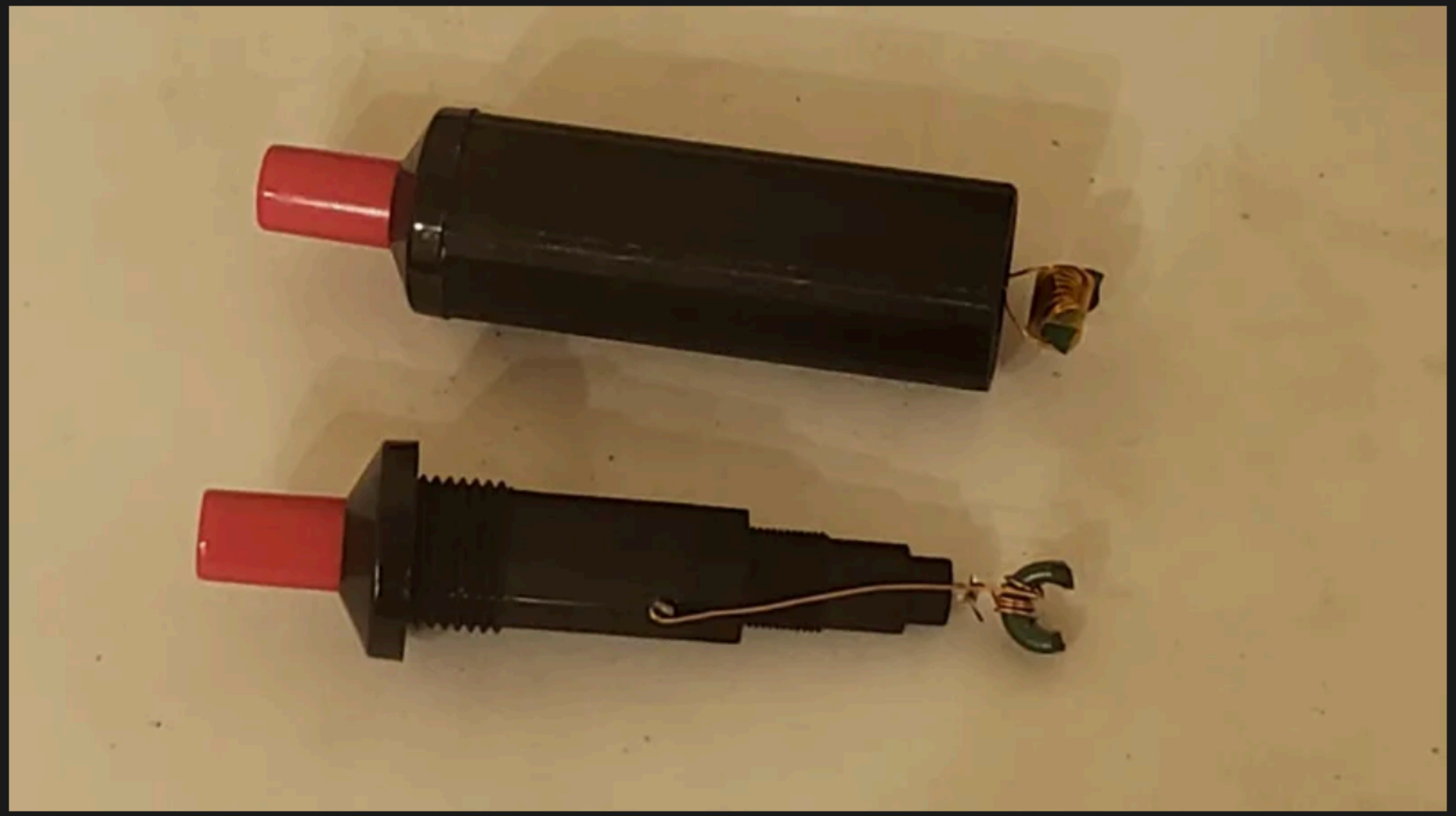
March 8, 2022

BLAST CHIPS WITH THIS BBQ LIGHTER FAULT INJECTION TOOL

by: Dan Maloney

16 Comments

January 29, 2022

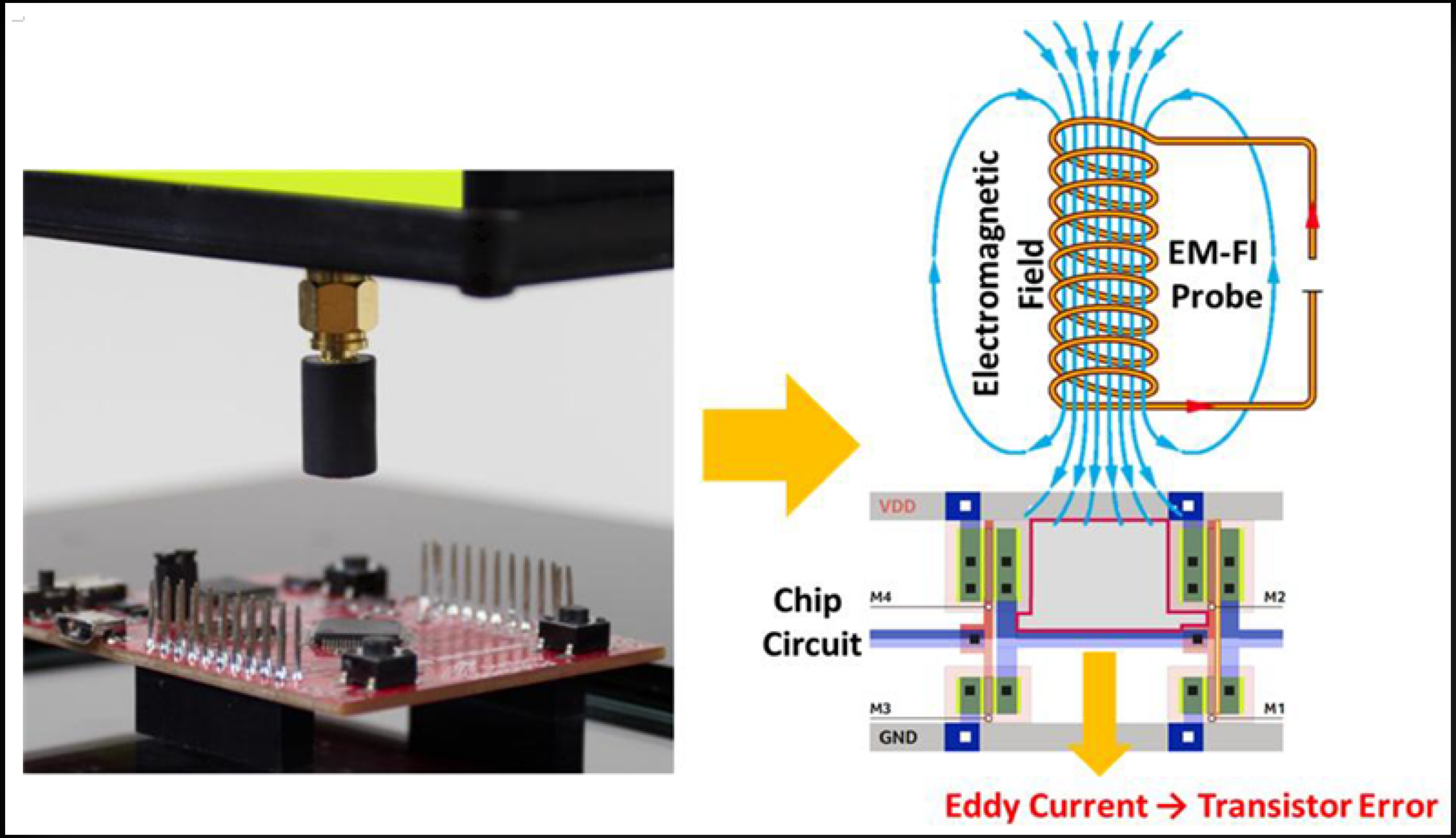


Looking to get into fault injection for your reverse engineering projects, but don't have the cash to lay out for the necessary hardware? Fear not, for the tools to glitch a chip may be as close as the nearest barbecue grill.

SEARCH

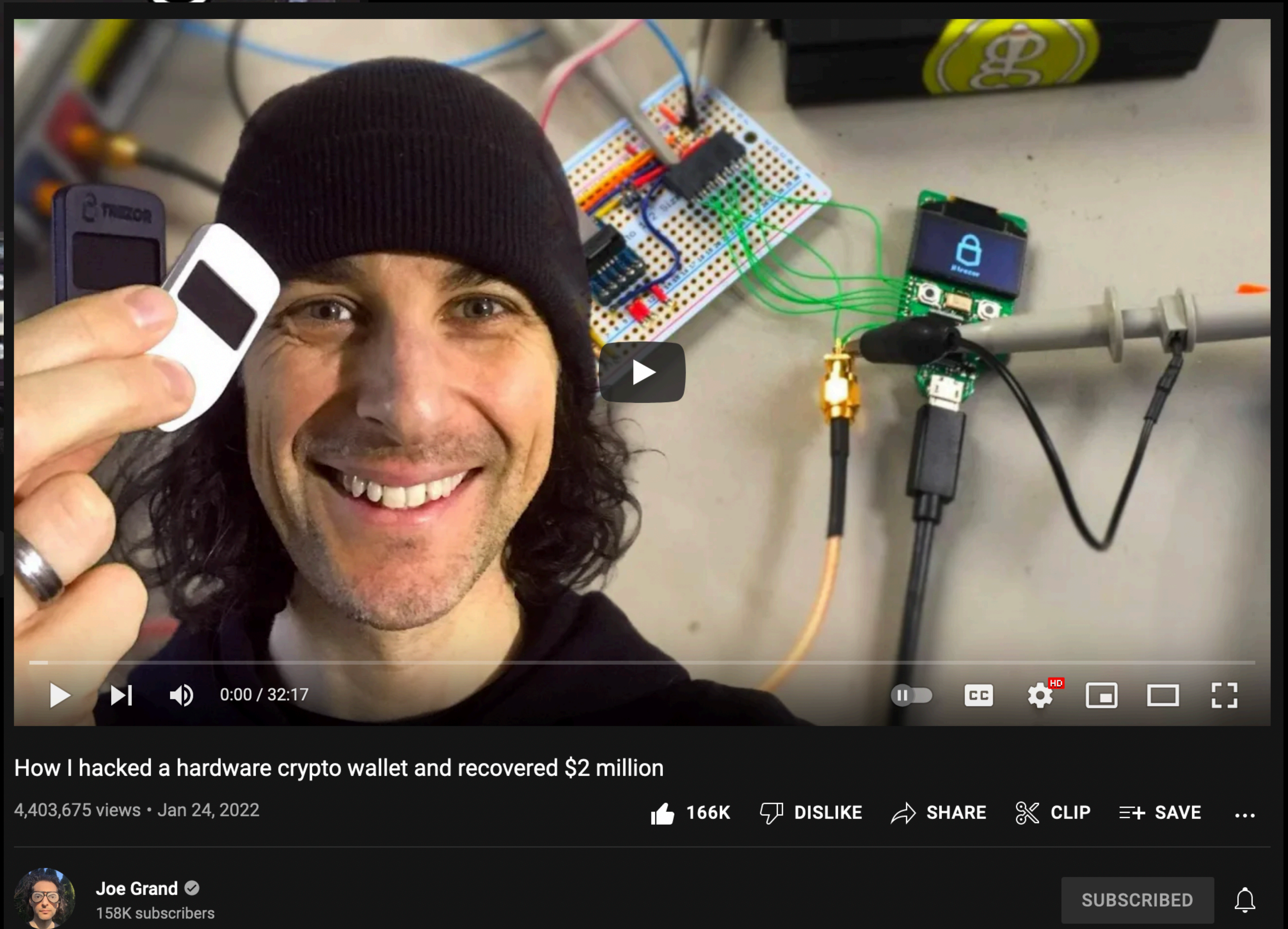
Search ... SEARCH

EM or Photonic Signals Work, Too.



Lim et al. Novel Fault Injection Attack without Artificial Trigger. Applied Science

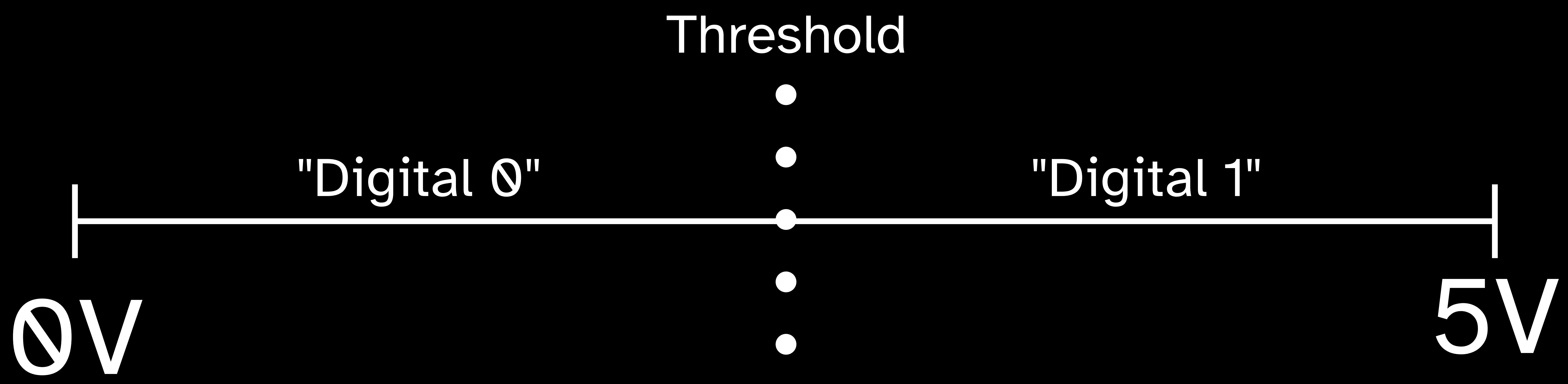
Notable Examples



So, why does that work?

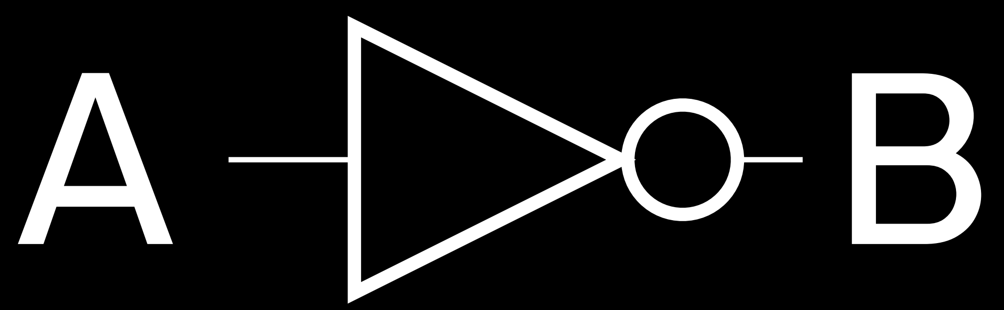
Representing 0s and 1s

Representing 0s and 1s

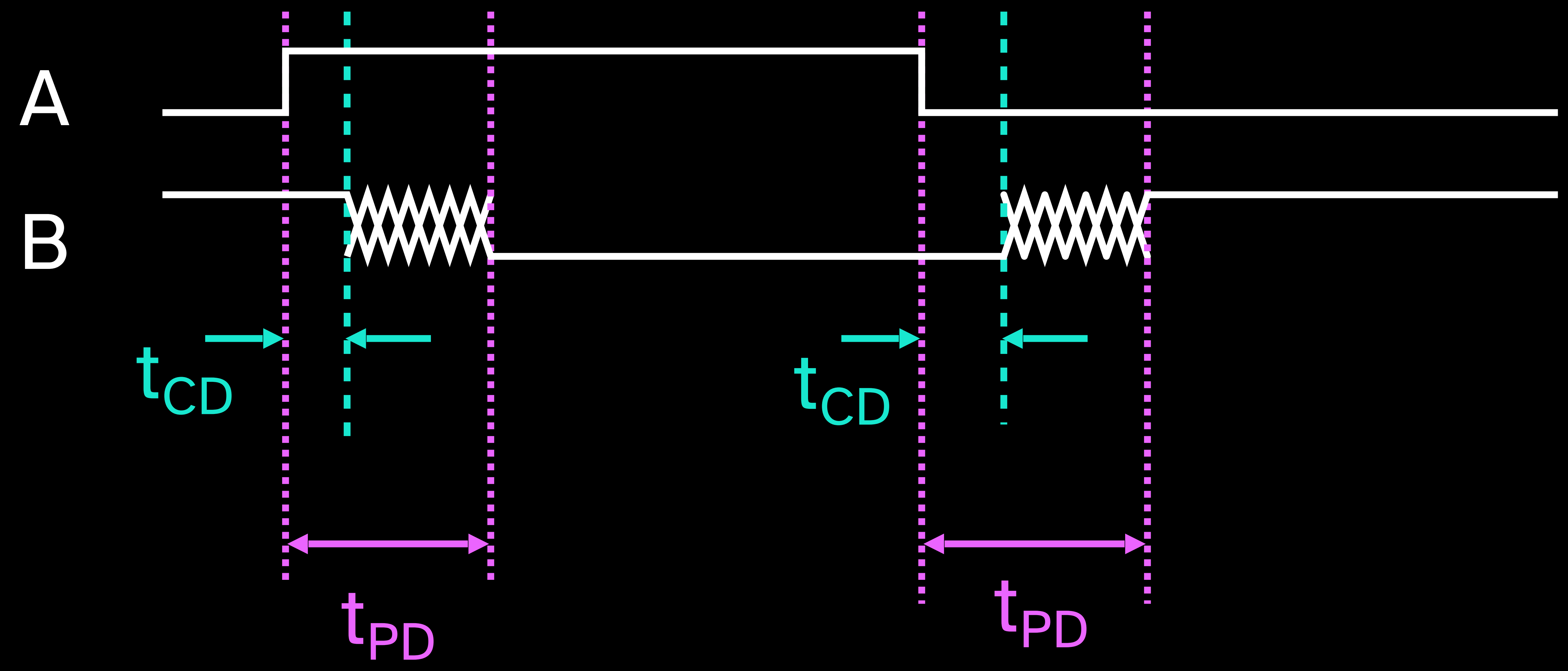


Representing 0s and 1s

Real-World Circuits Take Time



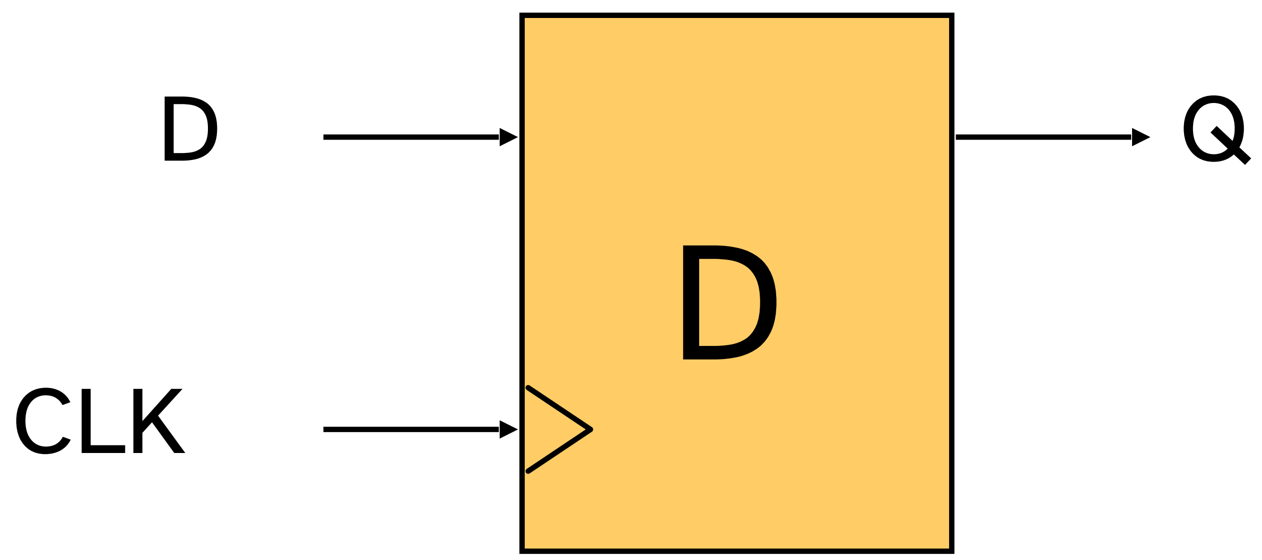
A	B
0	1
1	0

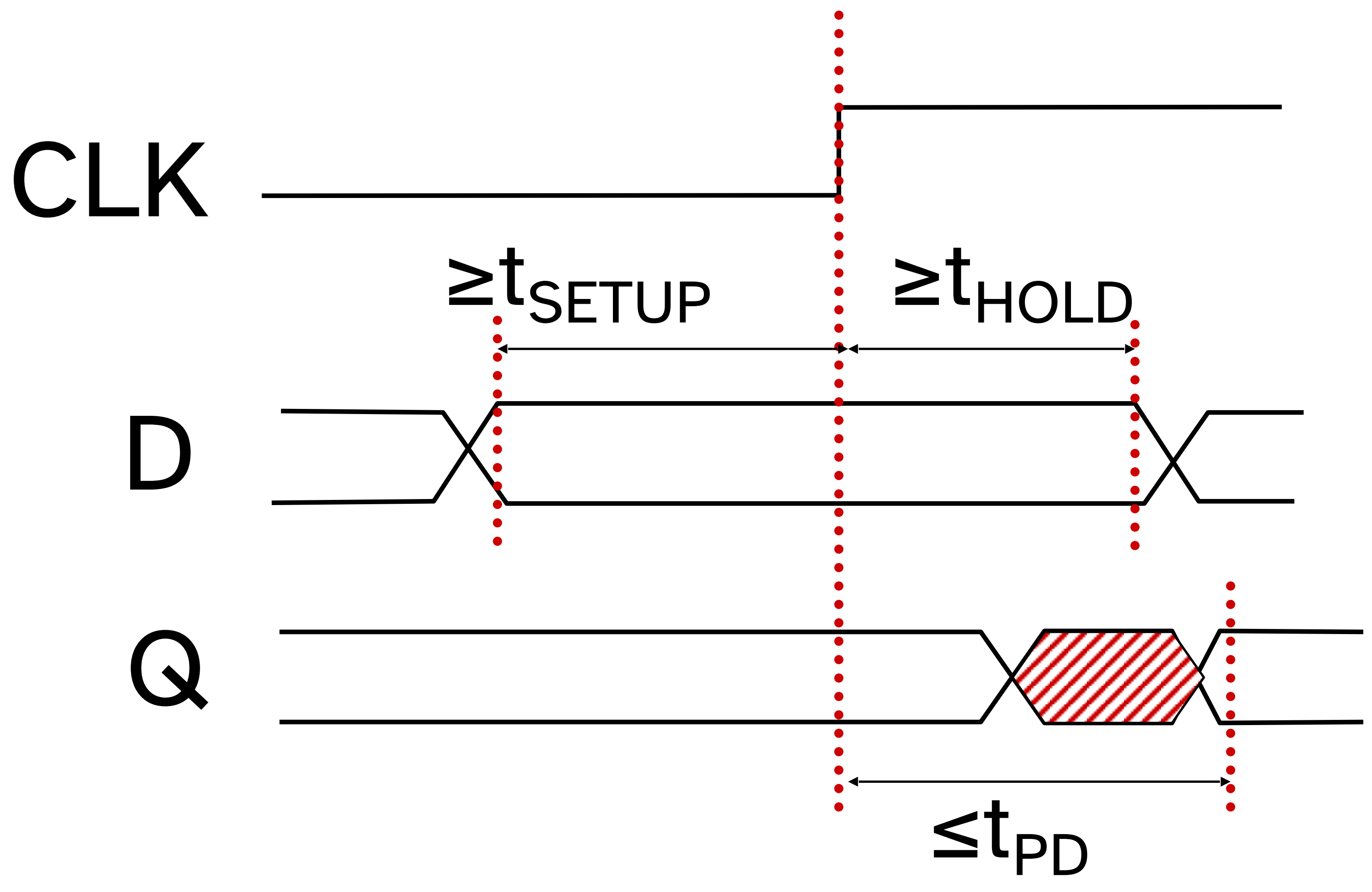


t_{PD} Propagation Delay

t_{CD} Contamination Delay

D Flip-Flop Timing (CLK Edge Trigger)





- Flip-flop input D should not change around the rising edge of the clock to avoid ***metastability***
- Formally, D should be a stable and valid digital value:
  - For at least t_{SETUP} before the rising edge of the clock
  - For at least t_{HOLD} after the rising edge of the clock
- Violating the timing constraints leaves the circuit in a metastability state. A contaminated value will be loaded into the register.

Metastability

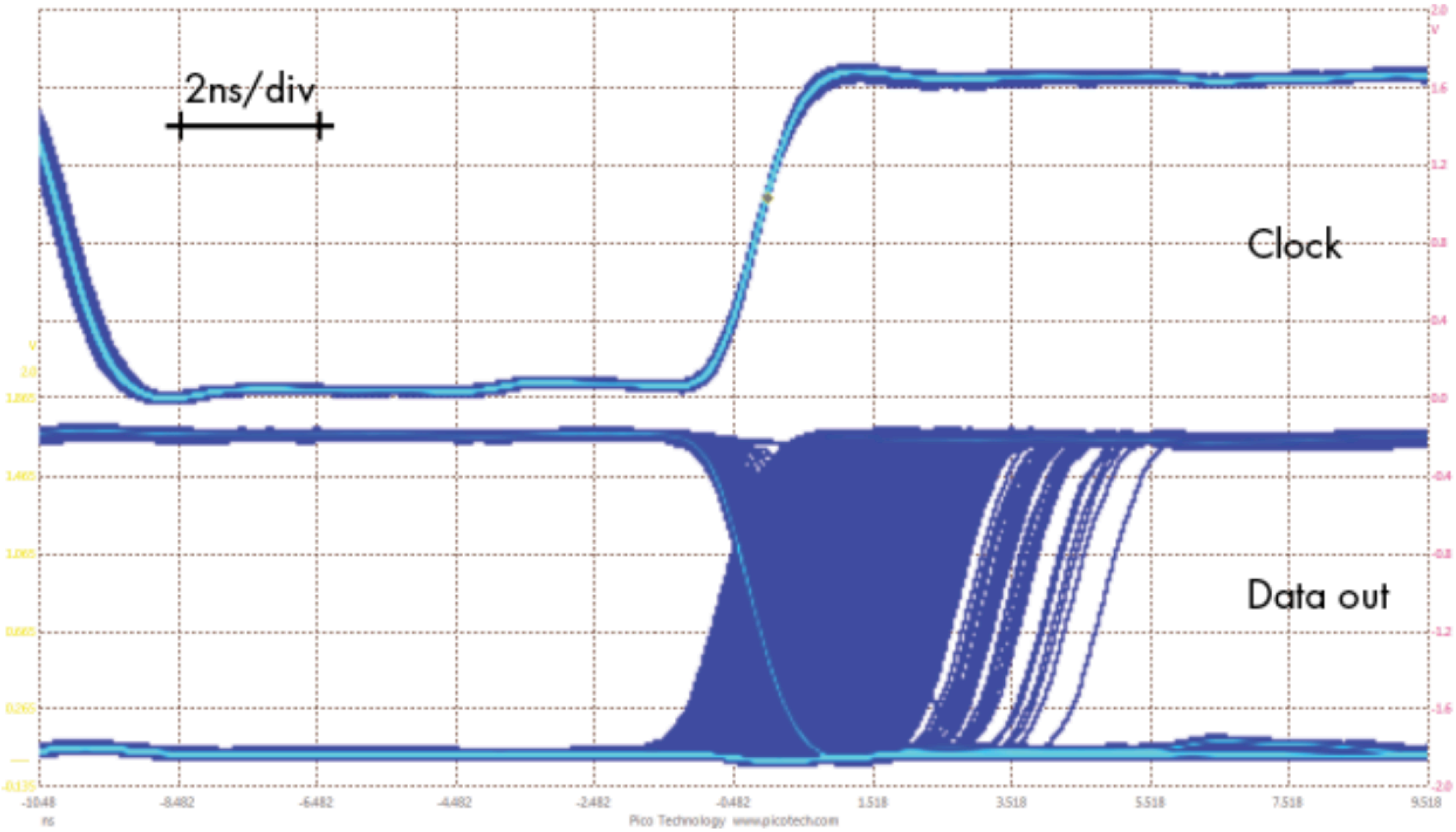
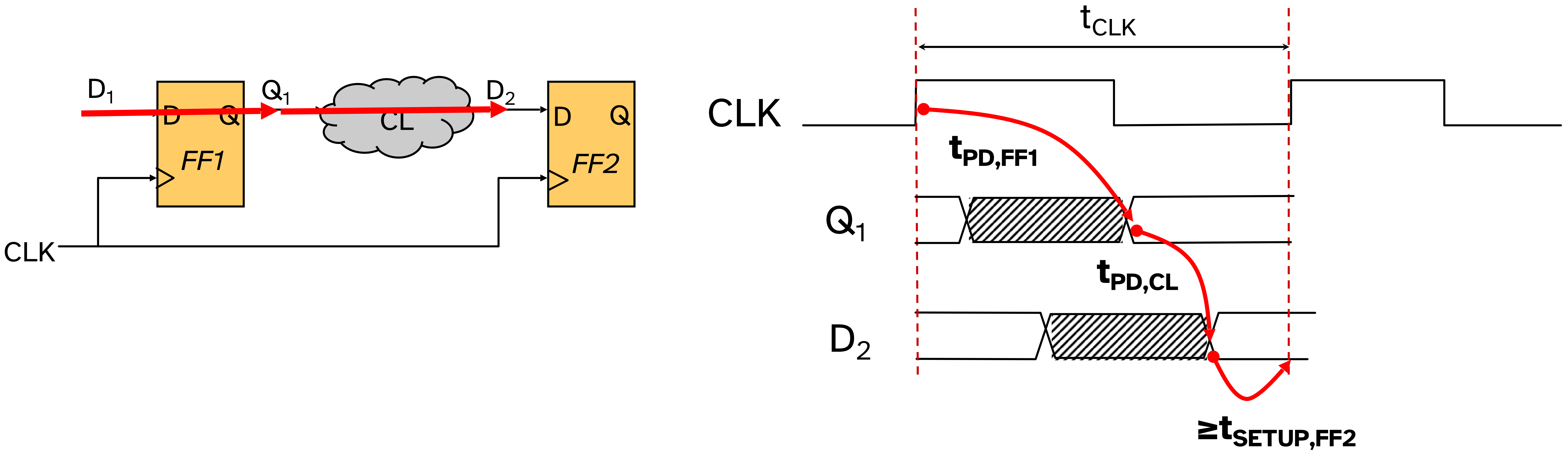


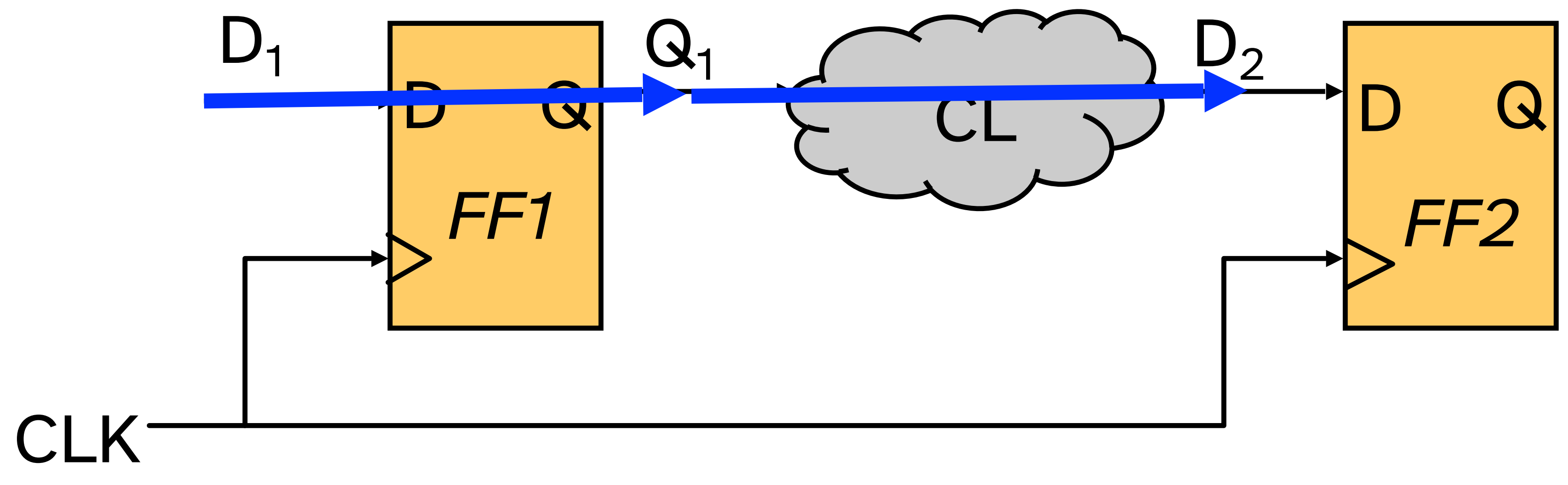
Figure 5-7: Metastable data output from shifting the clock edge to cause timing violations (low-voltage operation)

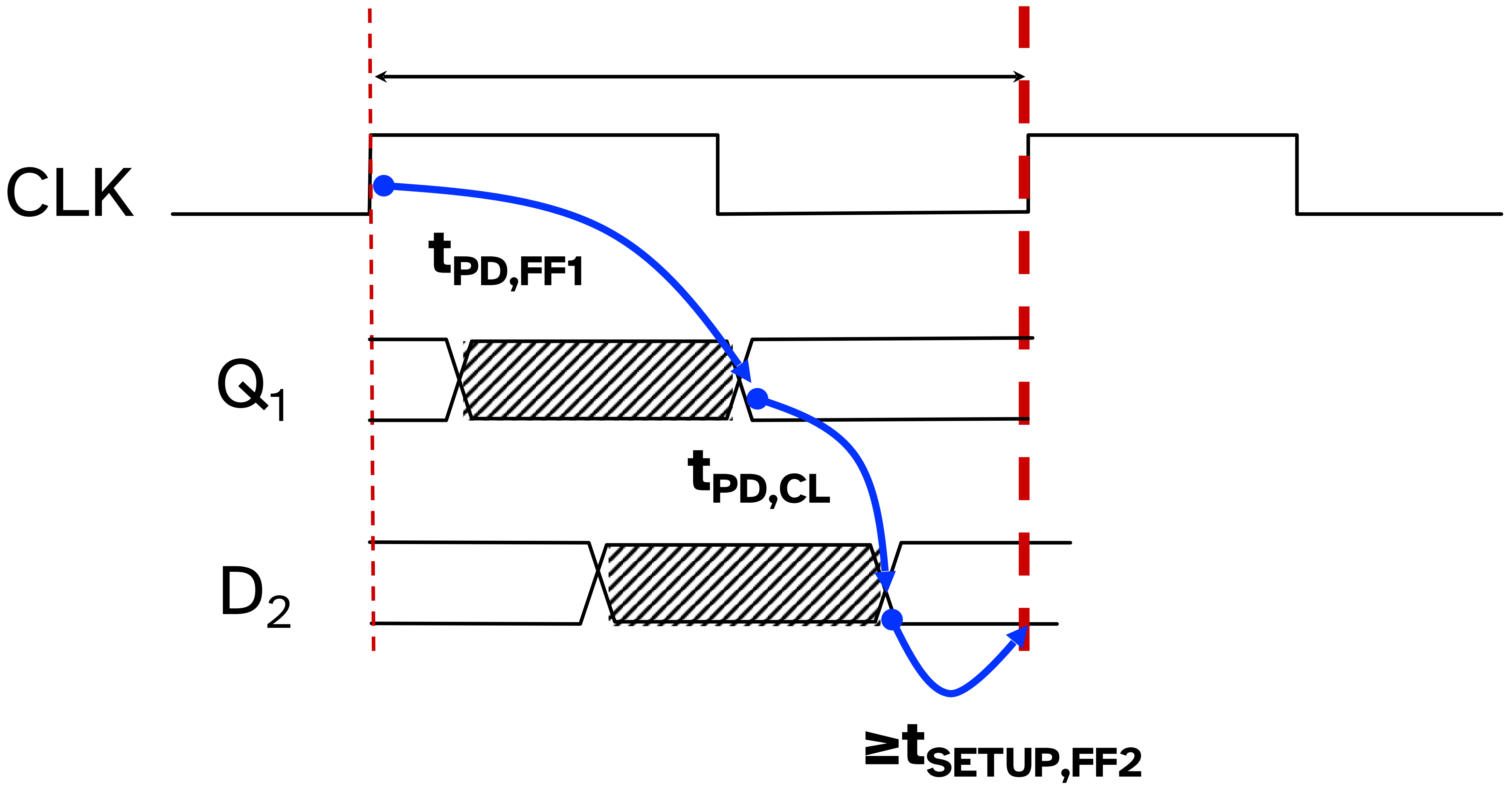
Colin O'Flynn. The Hardware Hacking Handbook. Chapter 5 Figure 5-8. No Starch Press.

Sequential Circuit Timing (Setup Time)



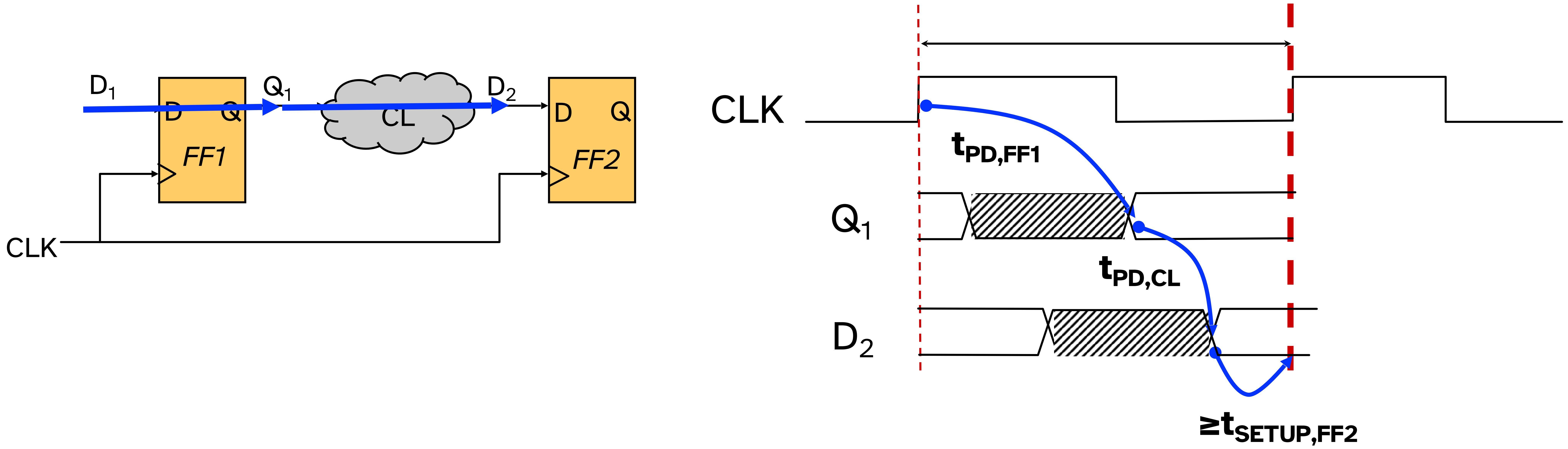
Fault Injection Attacks





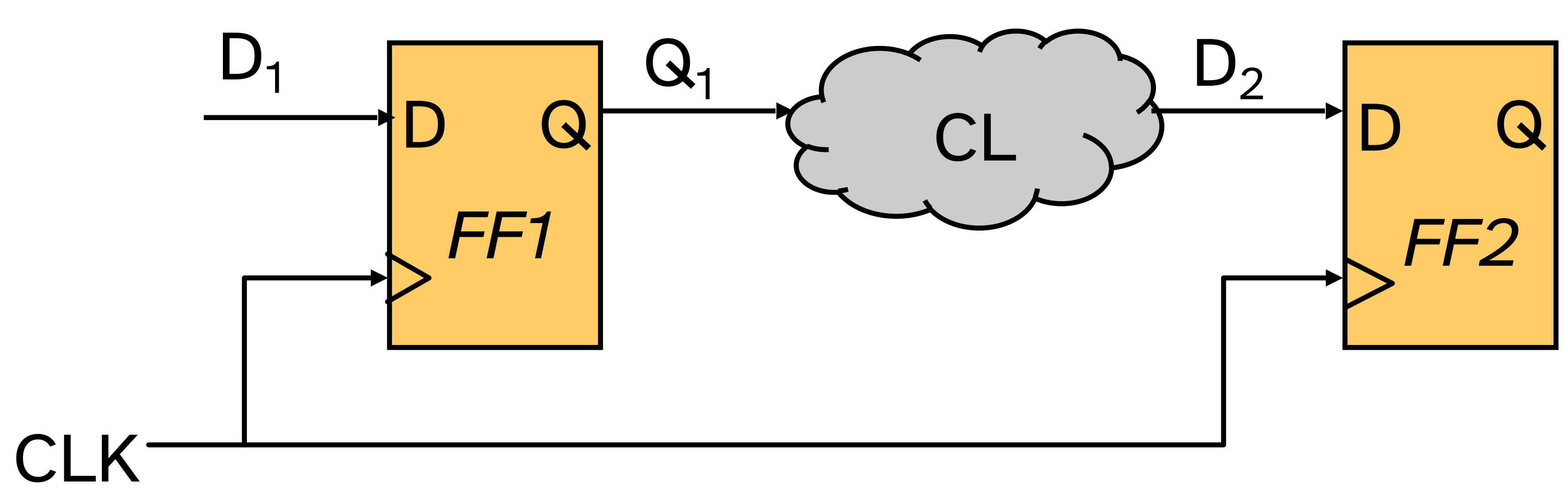
What if the clock comes earlier?

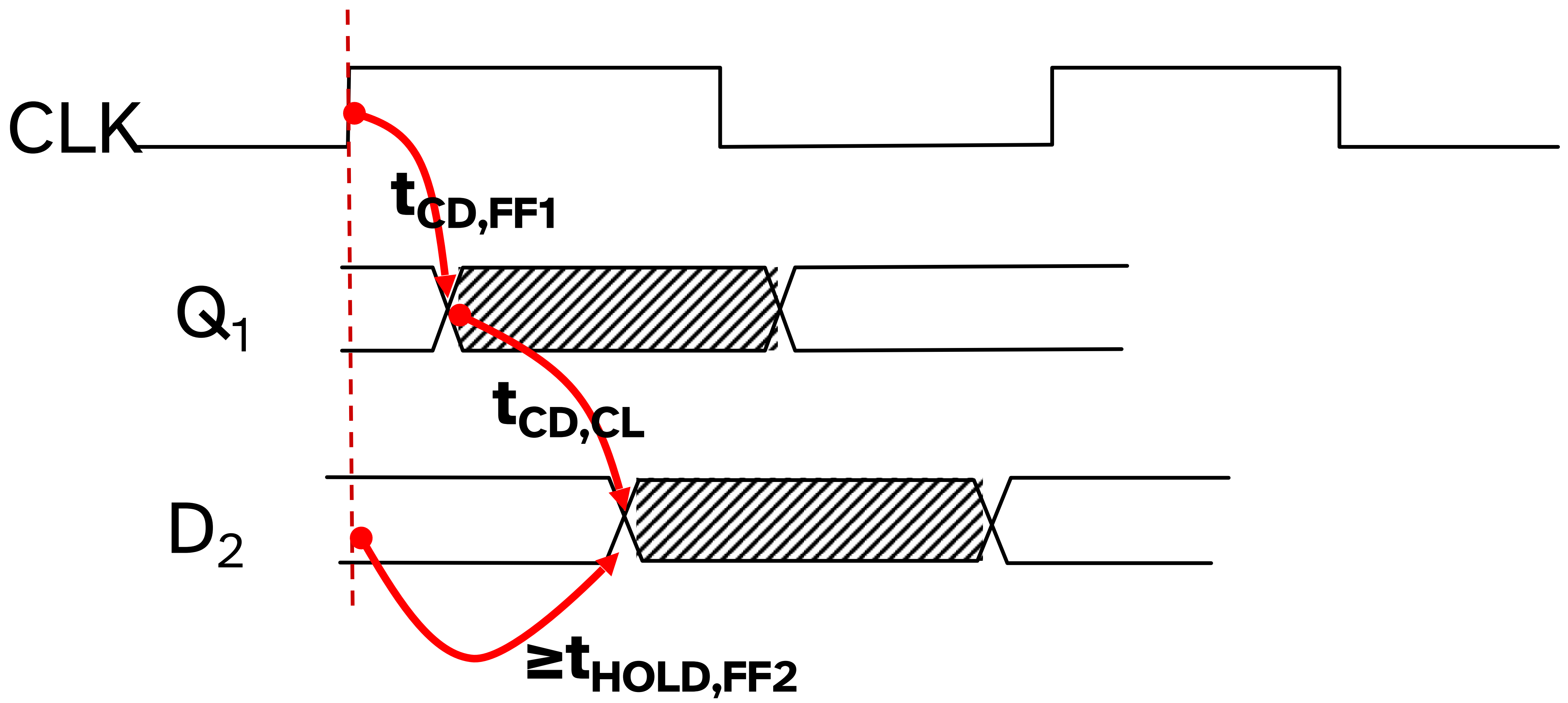
Fault Injection Attacks



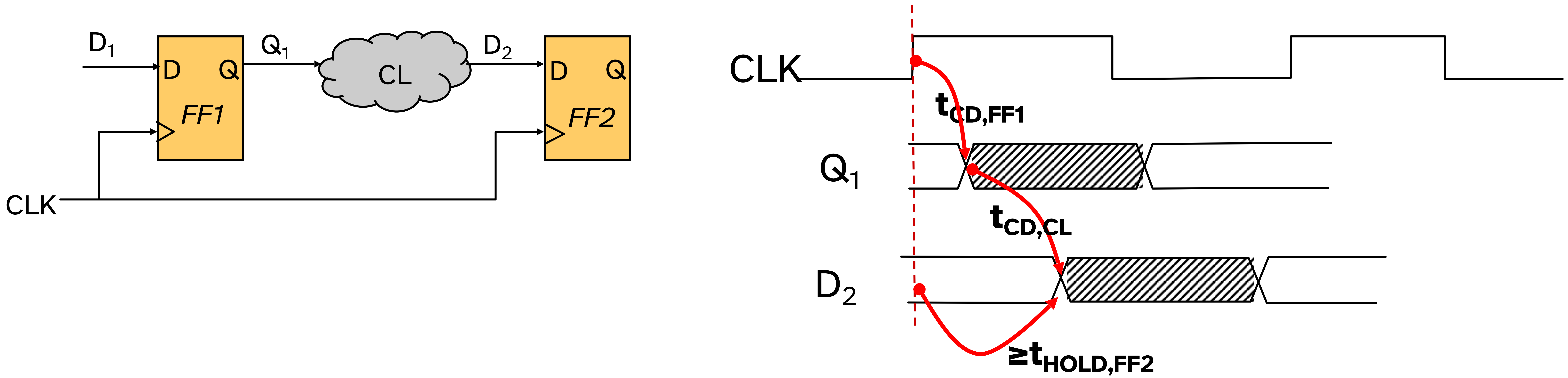
Decreasing the voltage increases propagation delay

Sequential Circuit Timing (Hold Time)





Voltage Glitching Attacks



Increasing voltage decreases contamination time

Can we stop it?

Mitigations

Redundancy

Think "two cores running the same thing". Can be expensive.

Example: OpenTitan.

Non-Determinism

Add randomness to the timing of certain chip operations.

Reduces accuracy of attack.

Fault Injection

Power Analysis

UART

Timing Analysis

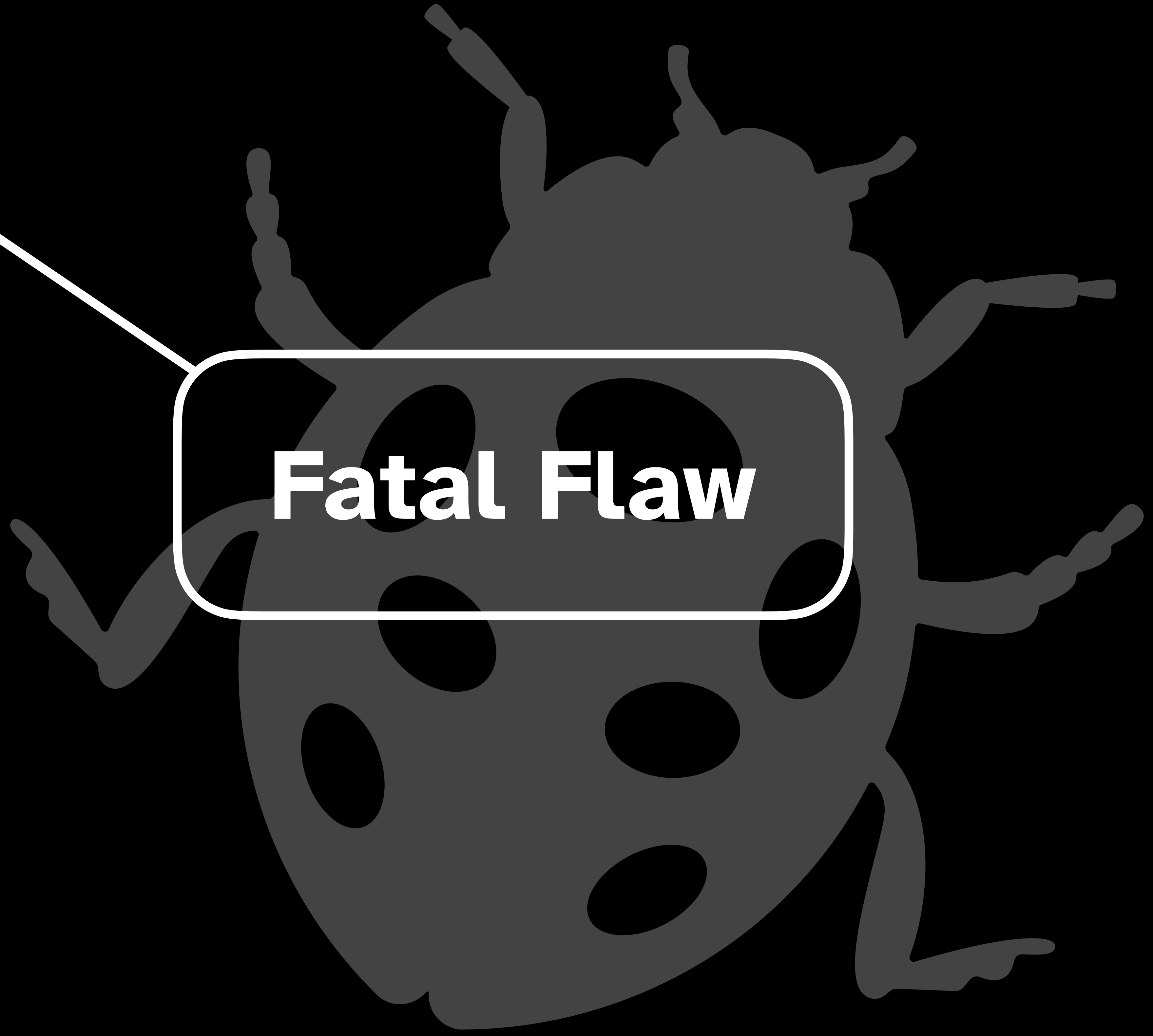
Timing Analysis

Spot the Bug

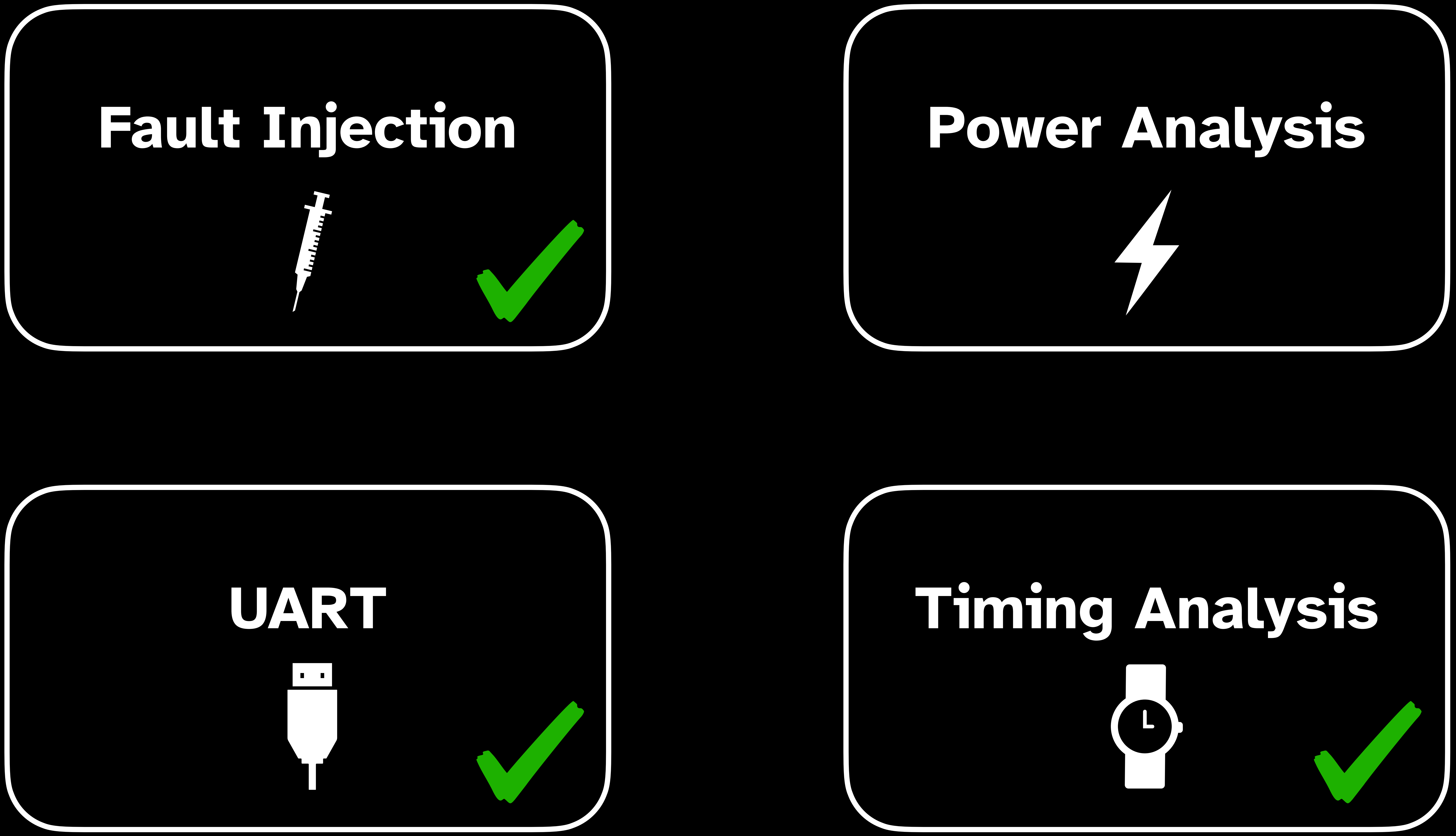
```
bool memcmp (char *buf1, char *buf2, size_t len) {
    for (int i = 0; i < len; i++) {
        if (buf1[i] != buf2[i]) {
            return false;
        }
    }
    return true;
}
```

Spot the Bug

```
bool memcmp (char *buf1, char *buf2, size_t len) {
    for (int i = 0; i < len; i++) {
        if (buf1[i] != buf2[i]) {
            return false;
        }
    }
    return true;
}
```



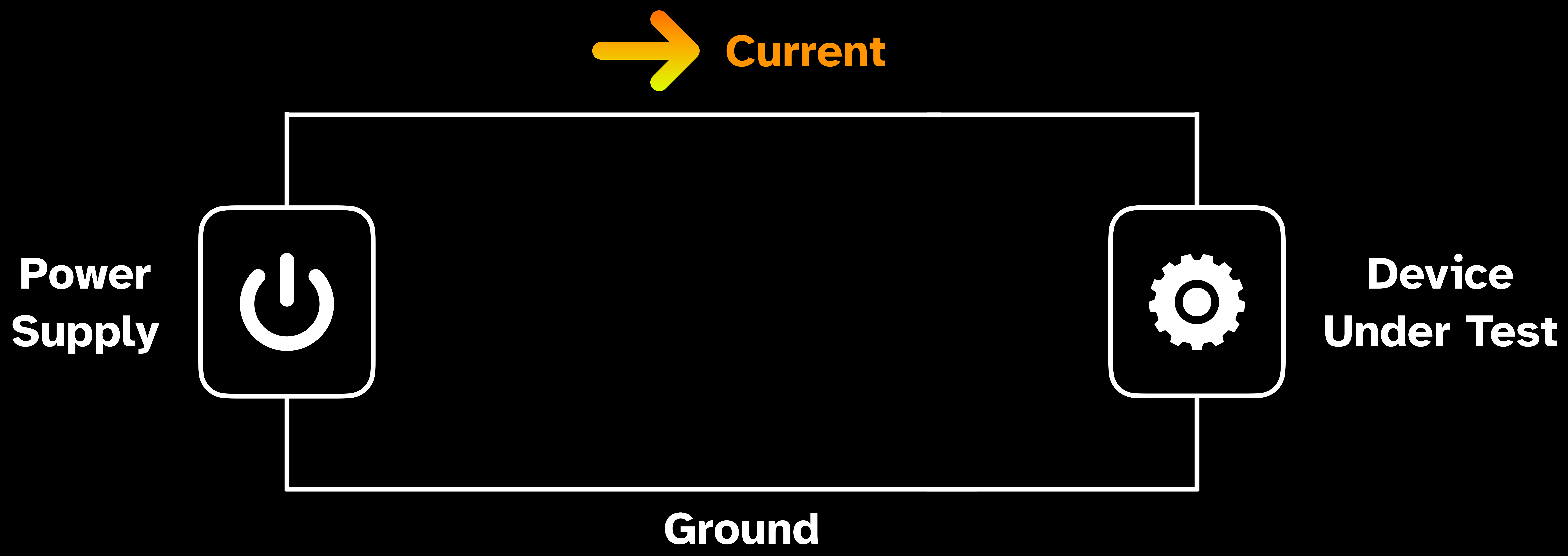
No Demo:
You will do this in recitation!



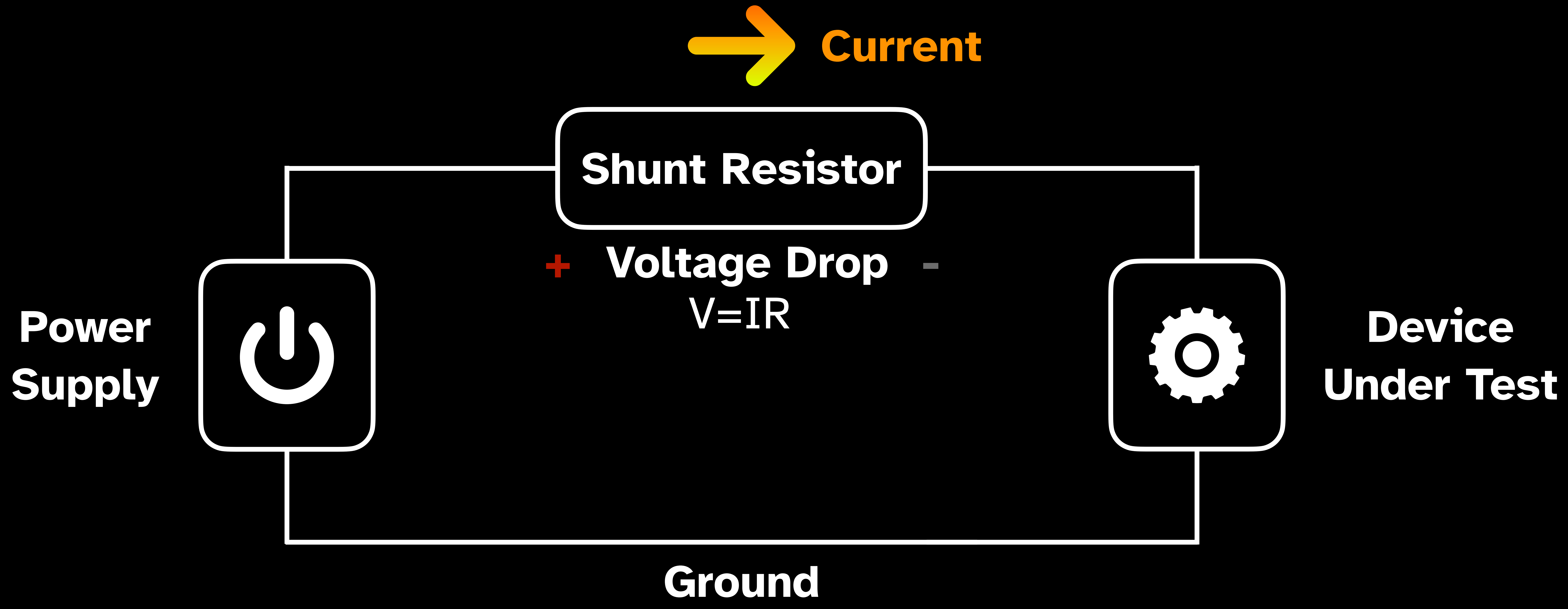
Fault Injection
Power Analysis
UART
Timing Analysis

Power Analysis

Power = Voltage x Current


Current
Power Supply
Device Under Test
Ground

How do you measure current on an oscilloscope?

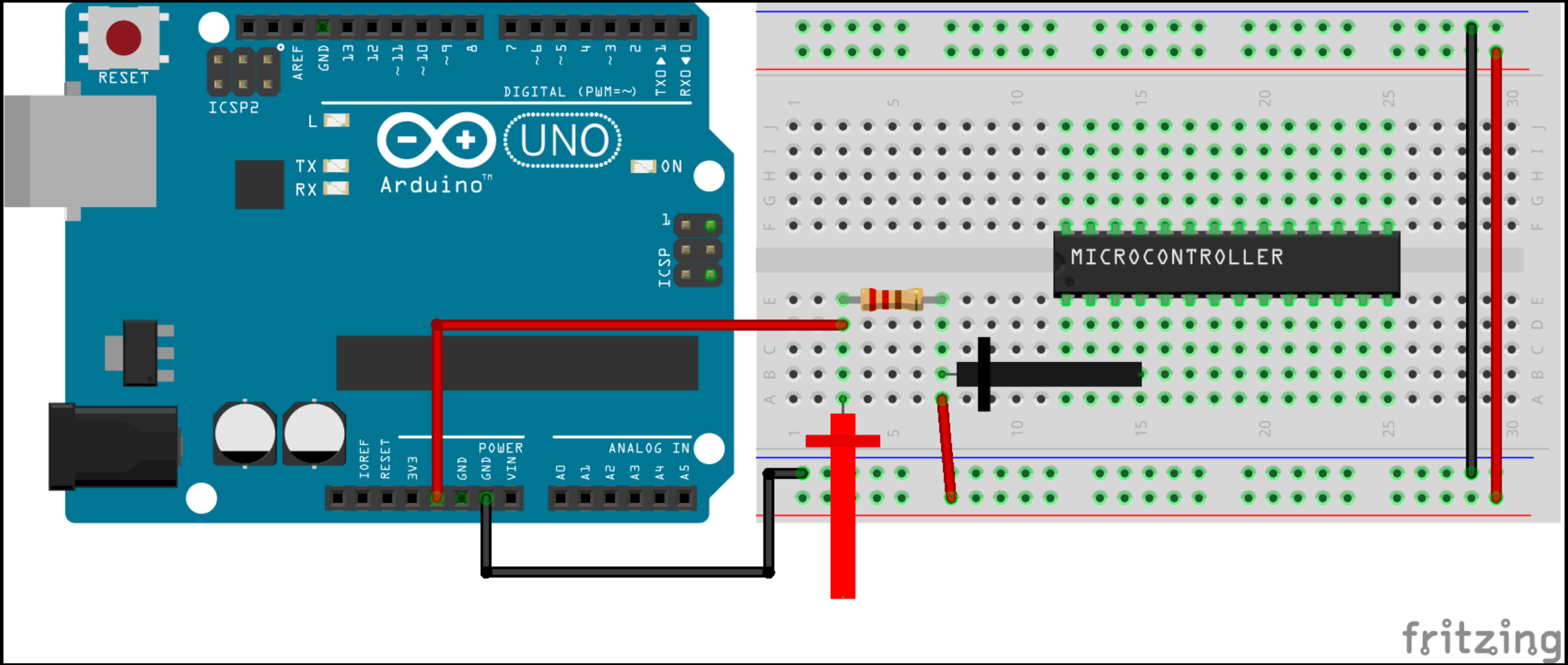

Current
Shunt Resistor
+ Voltage Drop -
V=IR
Power Supply
Device Under Test
Ground

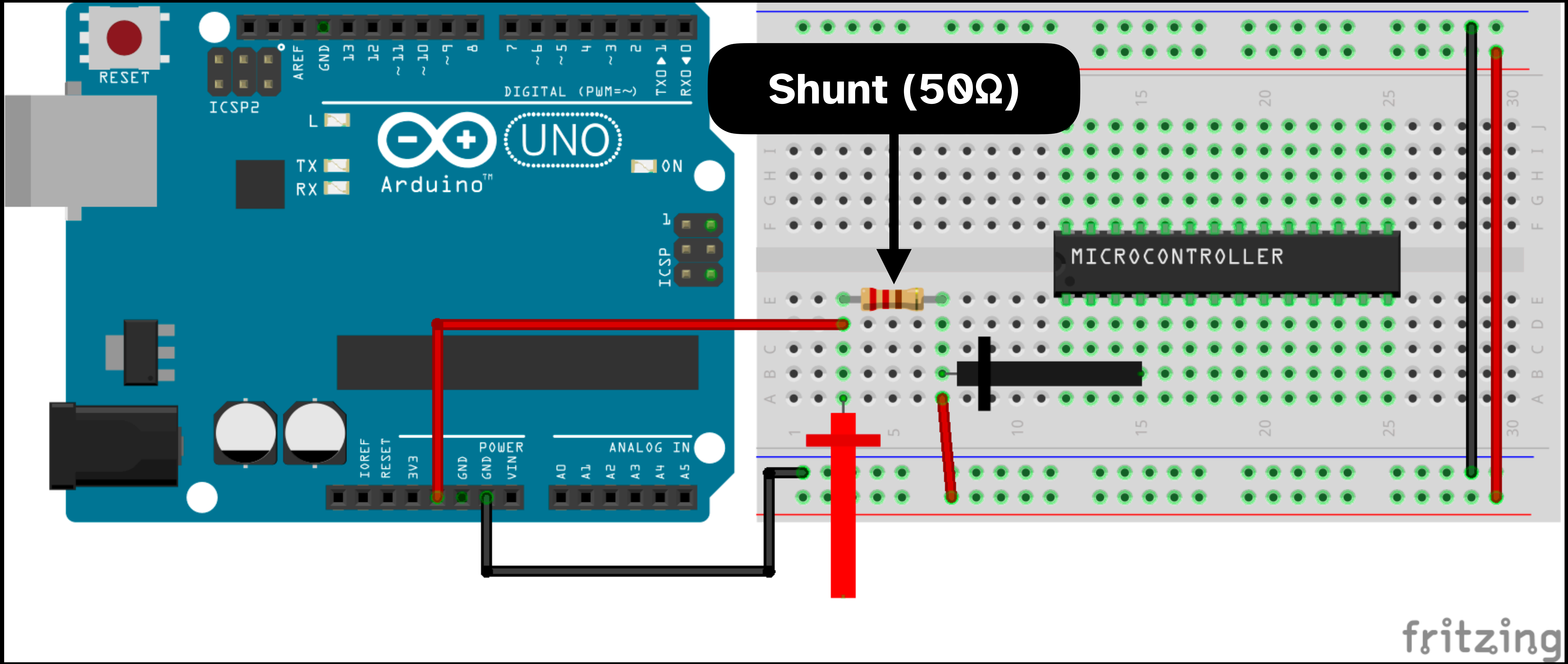
Apply Ohm's Law

Voltage (V) = Current (I) * Resistance (R)

Or in other words,

$$I = V / R$$


RESET
ICSP2
AREF
GND
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino
ON
ICSP
IOREF
RESET
3V3
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
MICROCONTROLLER
fritzing


Shunt (50Ω)
MICROCONTROLLER
RESET
ICSP2
DIGITAL (PWM=~)
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN
fritzing

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

RSA Modular Exponentiation


```
int rsa_modExp(int b, int e, int m) {
  int product = 1;
  b = b % m;
  while ( e > 0){
    if (e & 1){
      product = modmult(product, b, m);
    }
    b = modmult(b, b, m);

    e >>= 1;
  }
  return product;
}
```

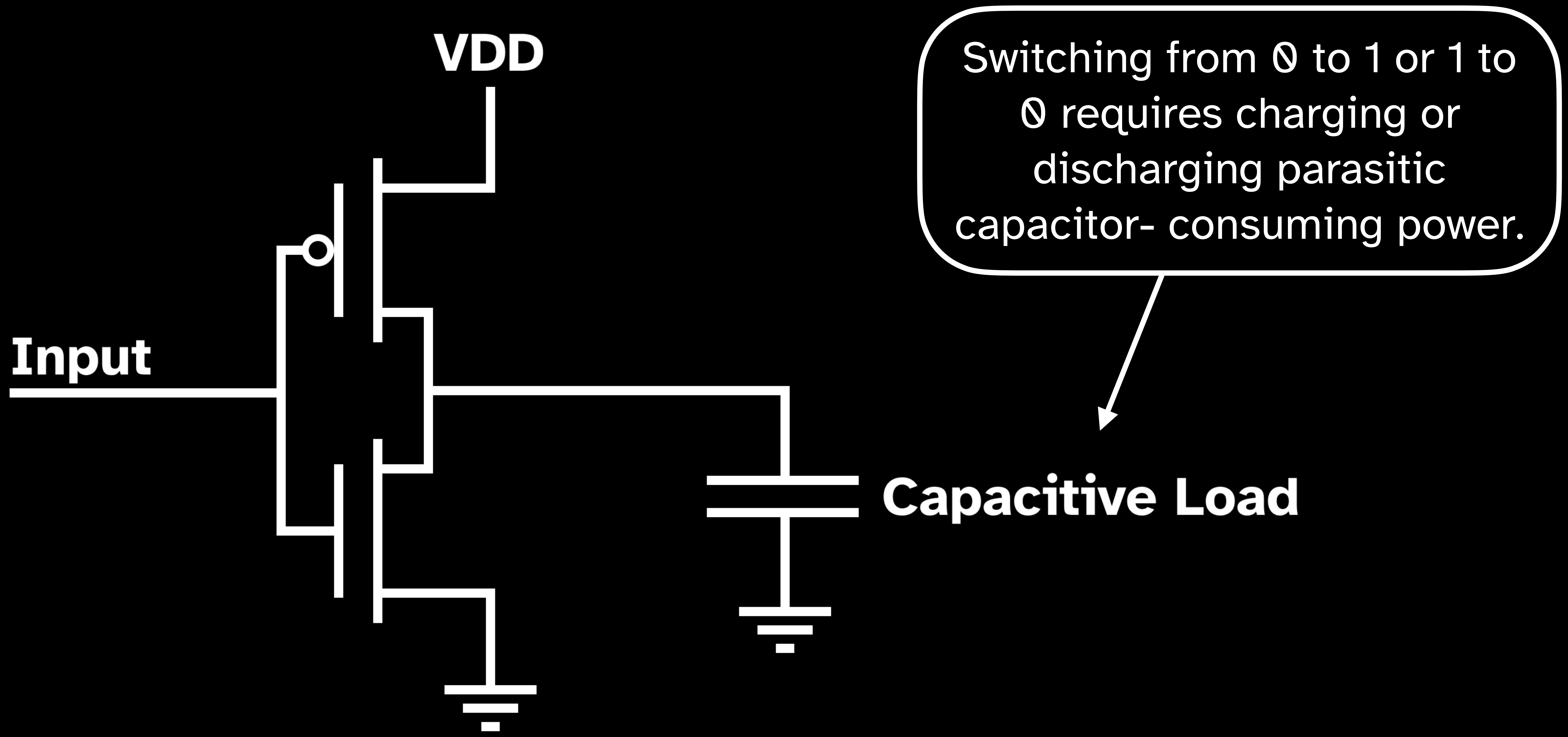
e = 0xf0

Demo 4

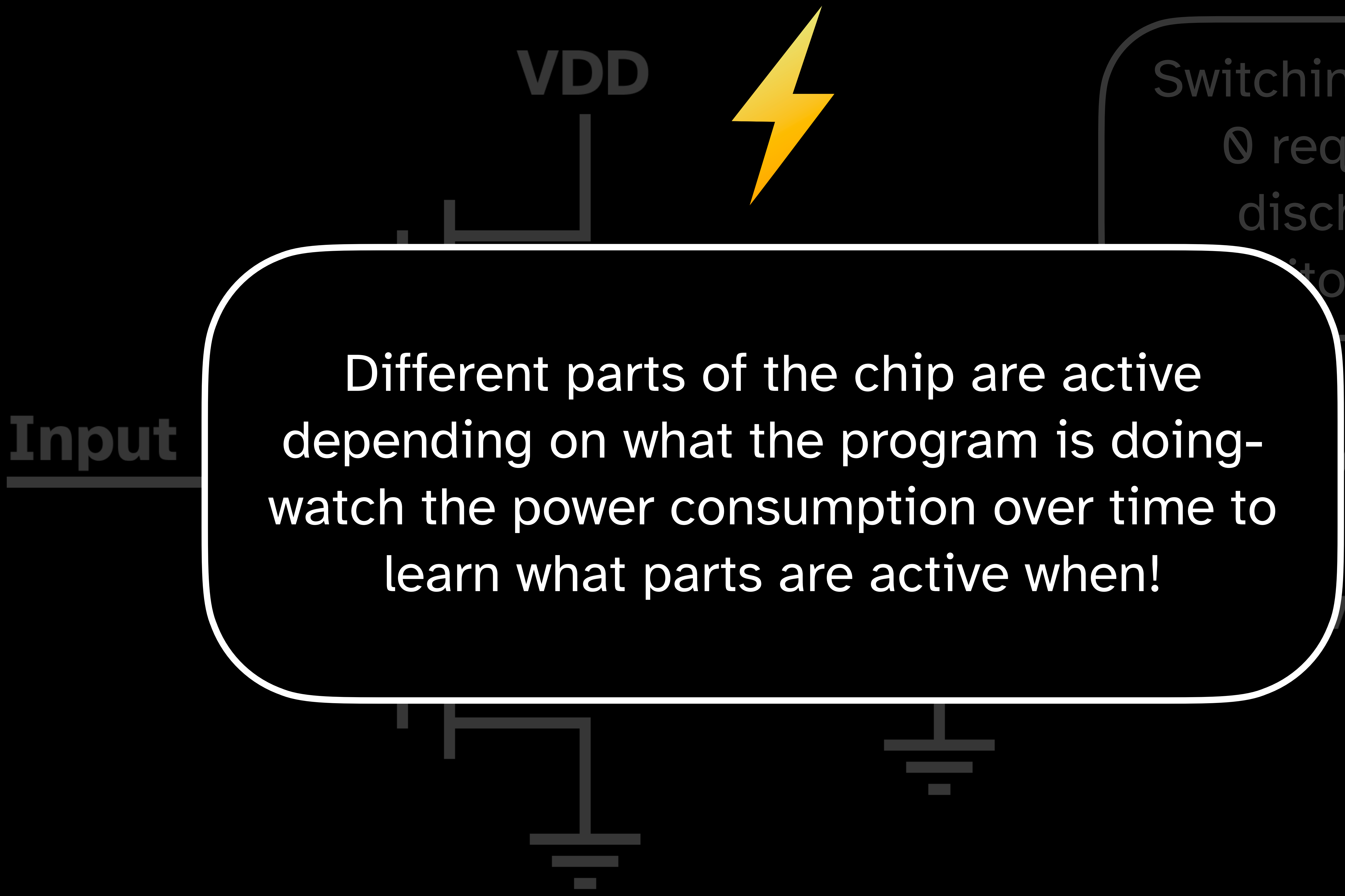
"What if we watch the chip's current draw?"

So, why does that work?

CMOS Inverter In Reality

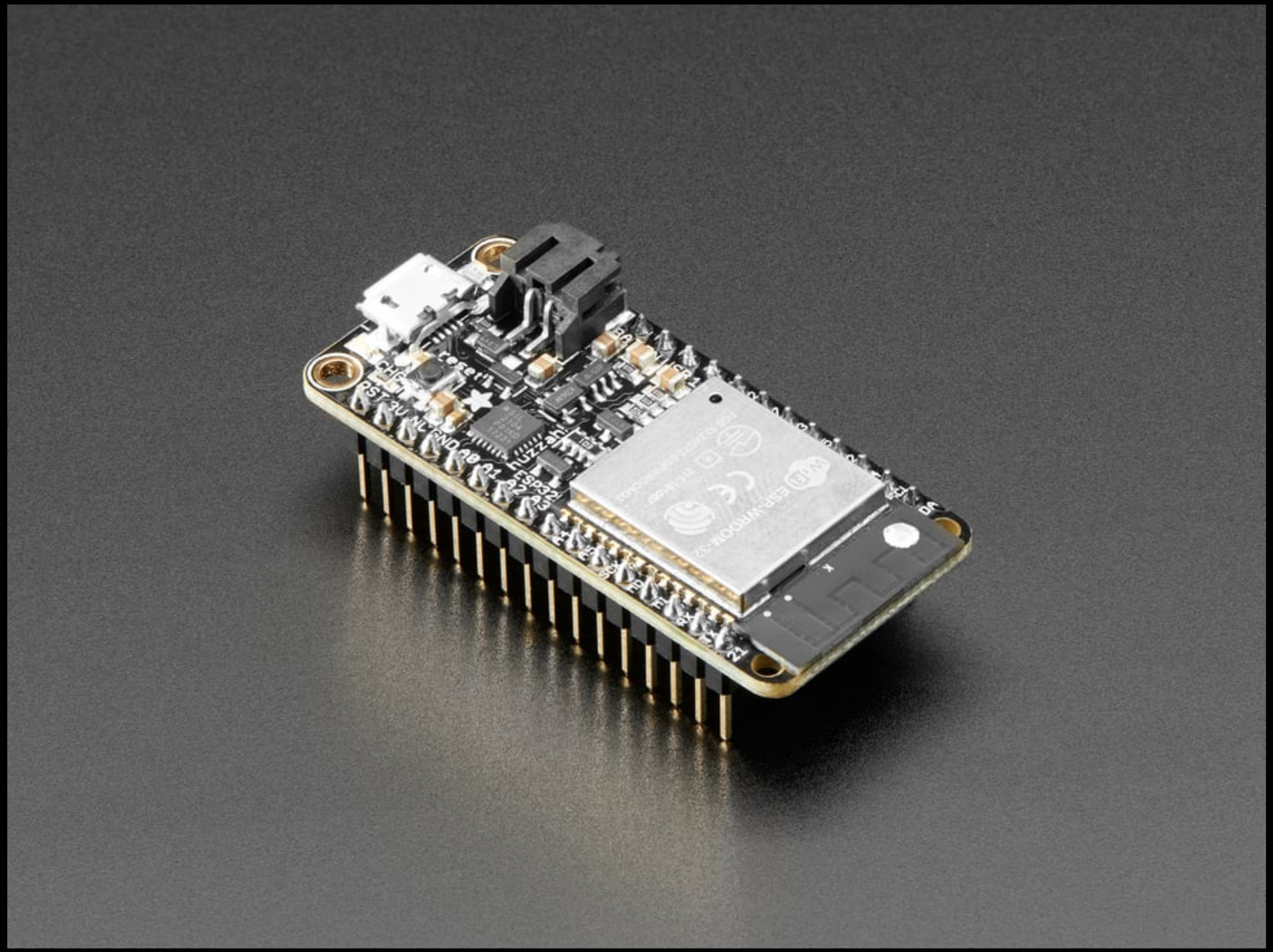


CMOS Inverter In Reality



Switching from 0 to 1 or 1 to 0 requires charging or discharging parasitic itor- consuming power.

Different parts of the chip are active depending on what the program is doing- watch the power consumption over time to learn what parts are active when!



Next:
Your Turn...

Image: Adafruit

Bring a USB Micro or USB-C cable if you have it.

Install the Arduino IDE as well- instructions will be posted on Piazza.