Power II

6.9000

Spring 2026

Last Week

- Talked about power
- How to measure
- How to use
- How to store
- Now…
- How to harvest
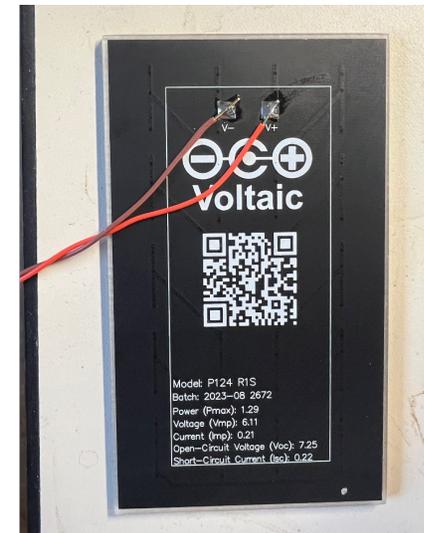- How to compute (with power in mind)

Recharging Cells?

- If you need to harvest energy, doing so from solar cells is most likely the best way to do so.

- There are other options theoretically, but practically solar is likely the best choice.

- Cheap, well-characterized, Generally robust, etc...
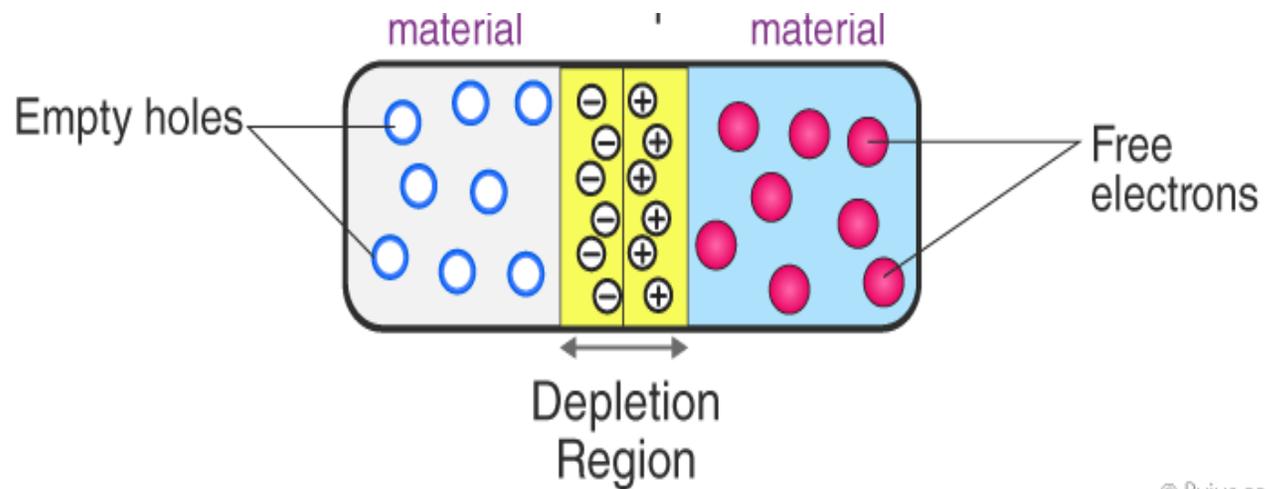
Front:

Back:

Used in lab 4 also some teams used last year
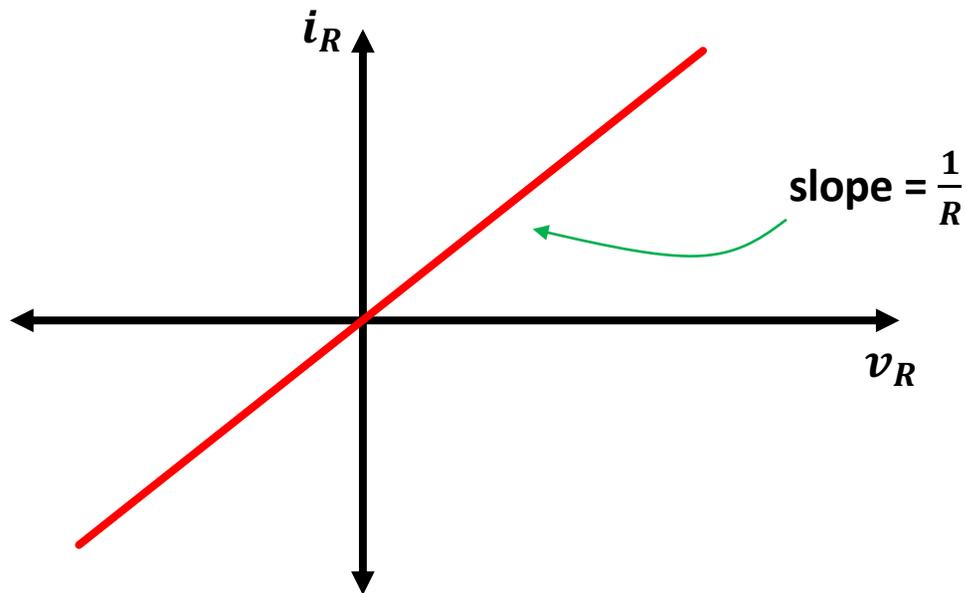
PN Junction
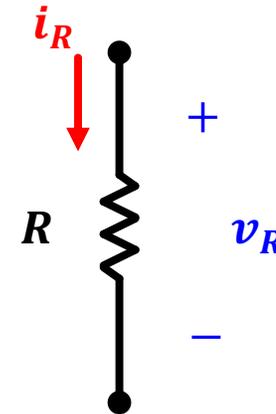
- A Solar cell is, interestingly, just a PN diode
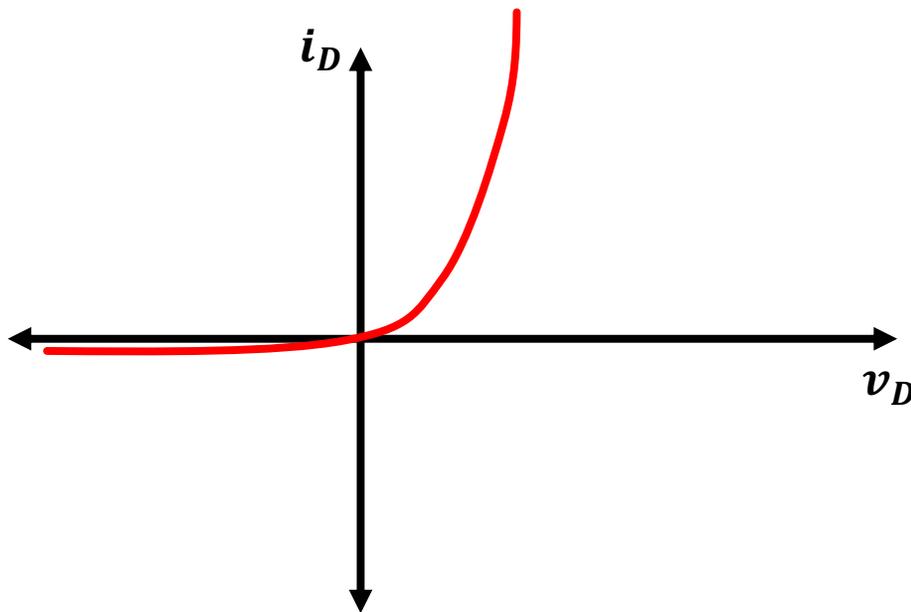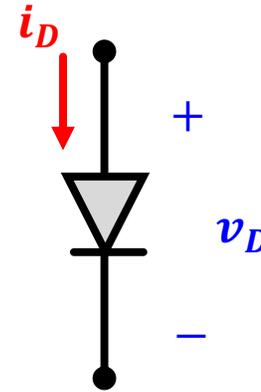
https://byjus.com/physics/p-n-junction/

What is the I-V relationship of a diode?

- A resistor, you'll remember is:

i_R

$+$

R v_R

$-$

i_R

slope $= \dfrac{1}{R}$

v_R

What is the I-V relationship of a diode?

- Now a Diode

i_D

$+$

v_D

$-$

i_D

v_D

Interesting Feature to Notice

- What quadrants of the I-V space do these plots both live in?
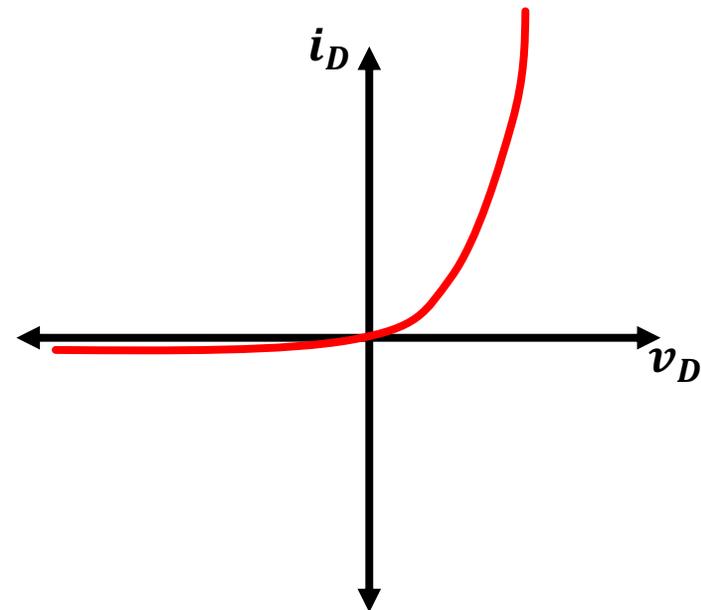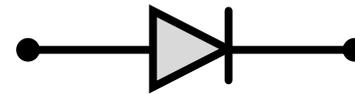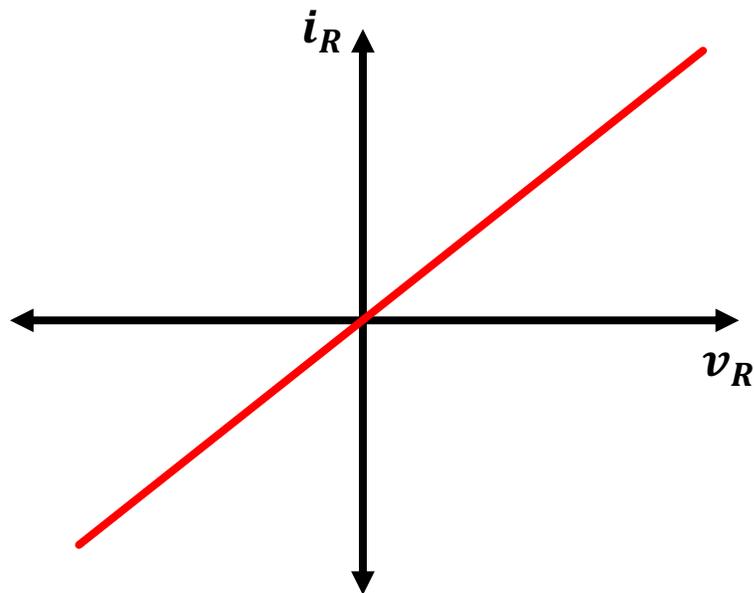
I-V Quadrants

- Diodes and resistors (and all non-power-generating devices) will be stuck in quadrants I and III
- In those quadrants: $p = v \cdot i$
  - Quadrant I: $p = (+) \cdot (+) \rightarrow (+)$
  - Quadrant III: $p = (-) \cdot (-) \rightarrow (+)$
- This is really meaningful…why?

- To be supplying power you need to be in Quadrants II or IV:
  - Quadrant II: $p = (-) \cdot (+) \rightarrow (-)$
  - Quadrant IV: $p = (+) \cdot (-) \rightarrow (-)$

A PN junction exposed to radiation

- Light starts to shine on a PN junction
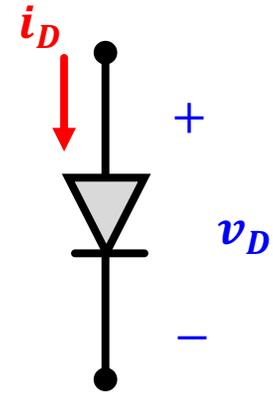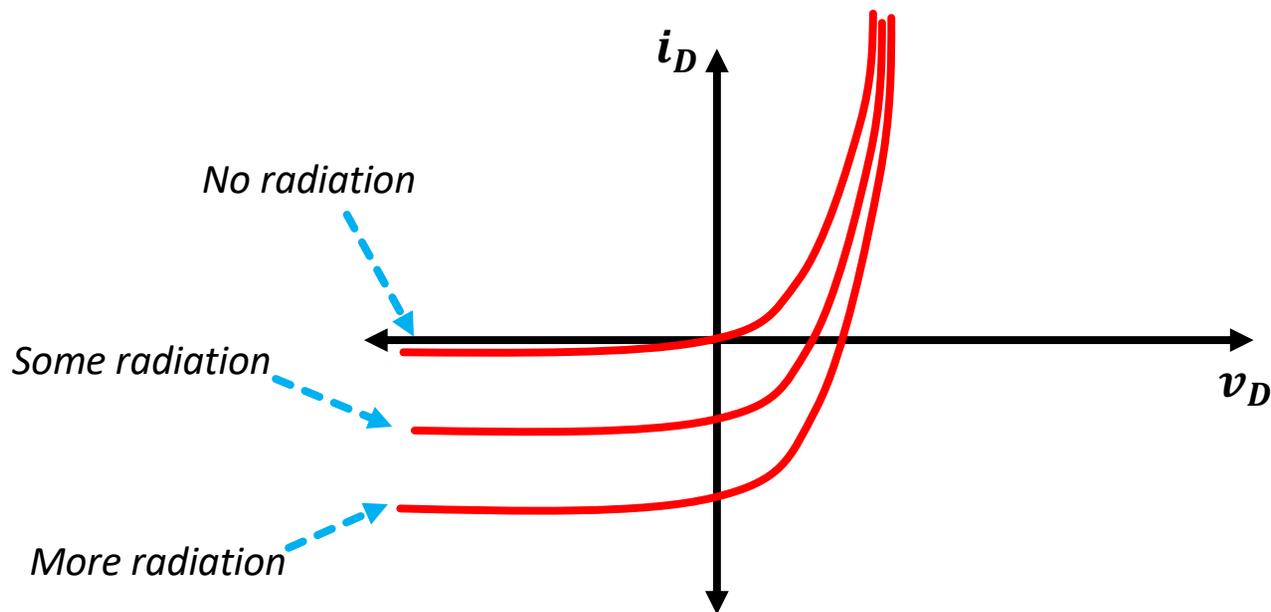
- What Do you Notice About these curves?

A PN junction exposed to radiation

- Light starts to shine on a PN junction

i_D

$+$

v_D

$-$

i_D

No radiation

Some radiation

More radiation

v_D

Quadrant IV !

$$p = (-) \cdot (+)$$
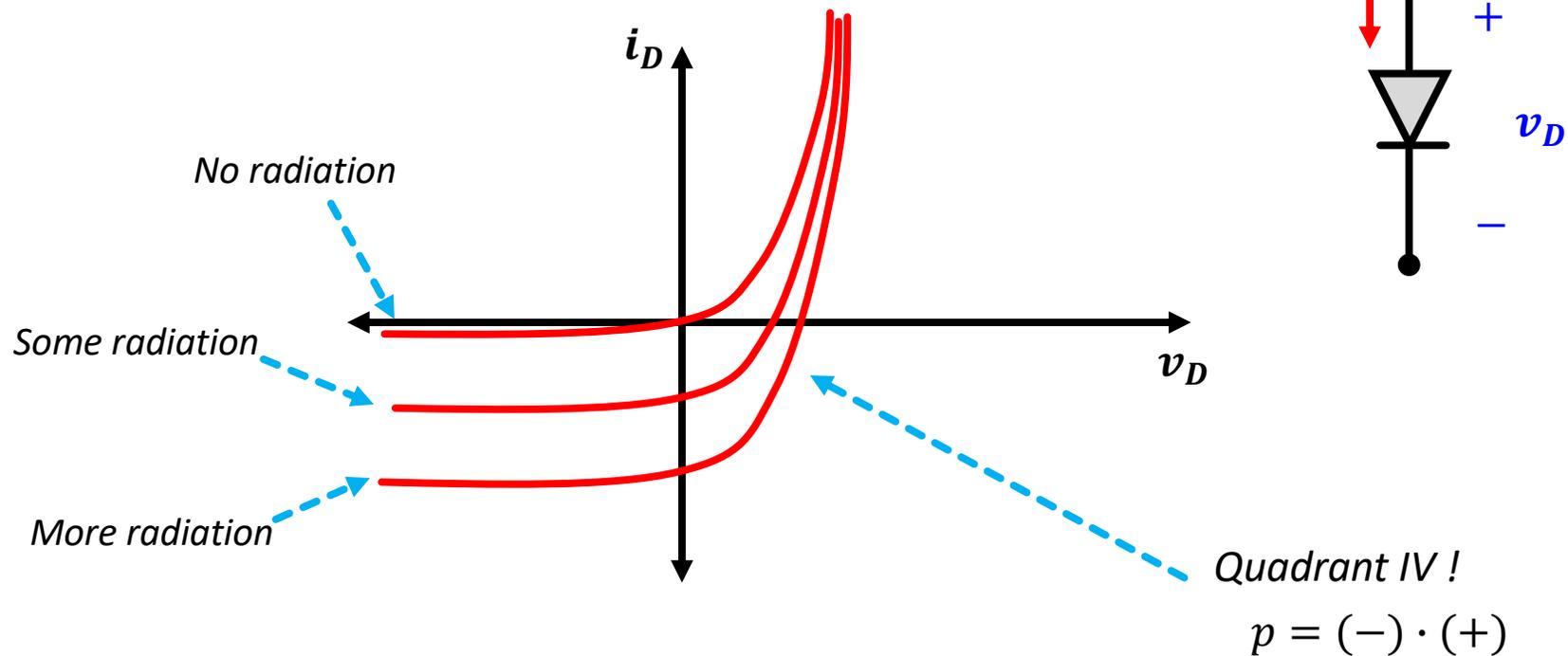
- What Do you Notice About these curves?

They Usually Flip Axes

- When drawing Power-supplying devices we usually flip the axes (as mentioned in class last week!!!)
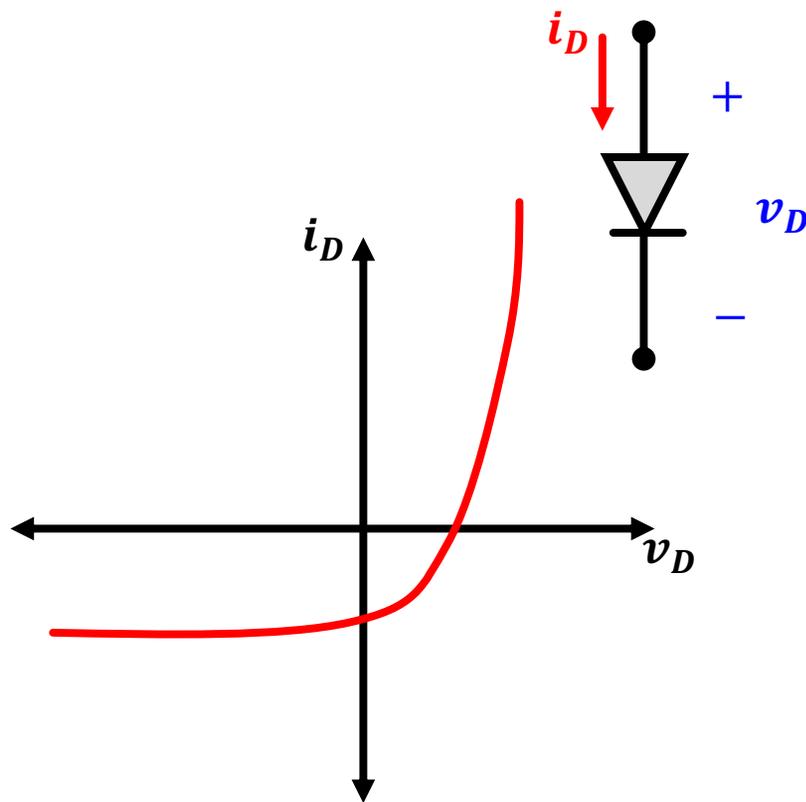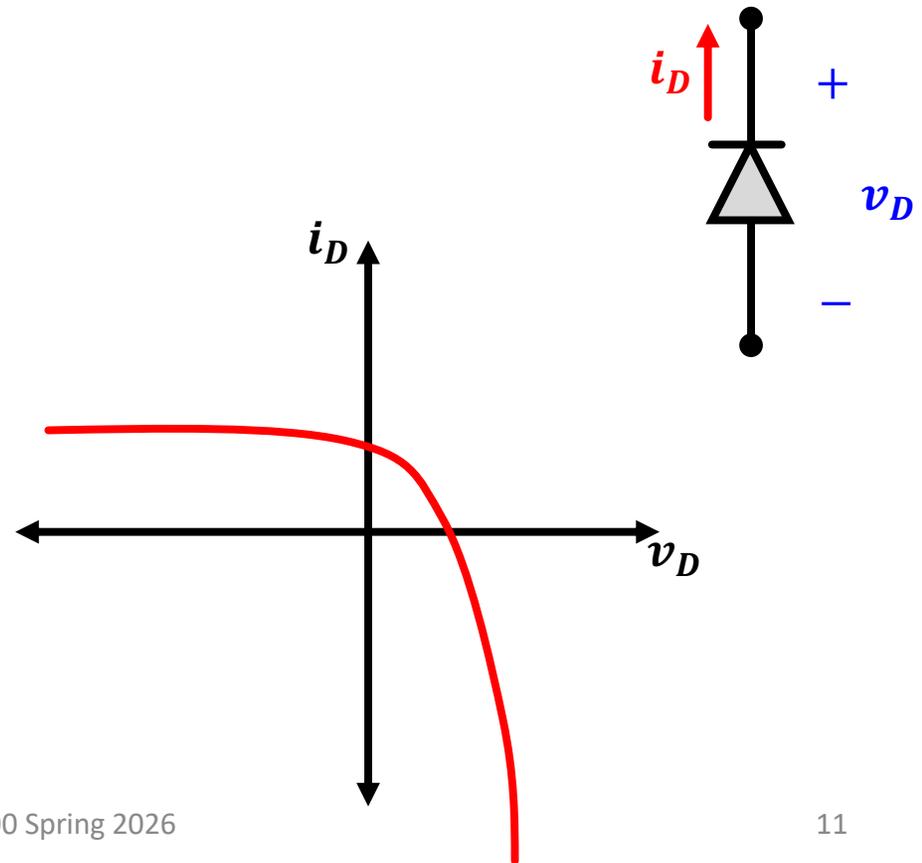
The I-V curve of a PV Cell

- If Power is the product of voltage and current, how much power can a PV cell provide?
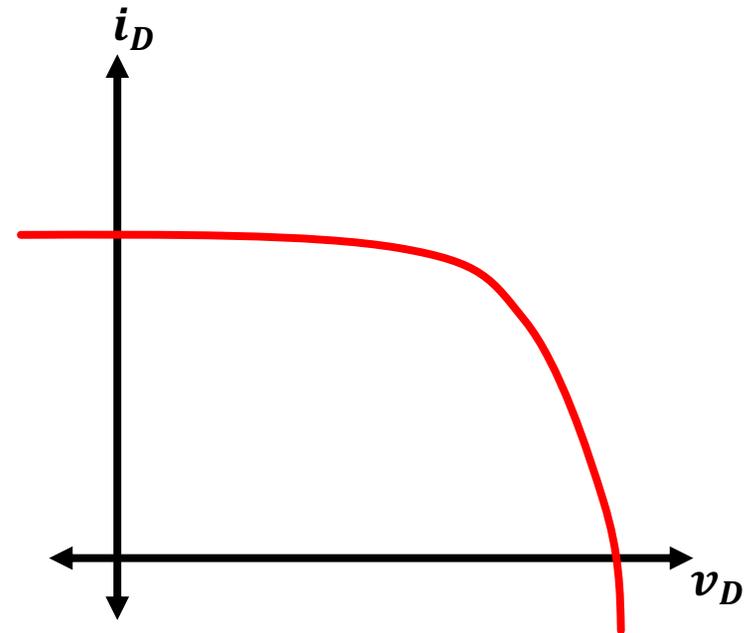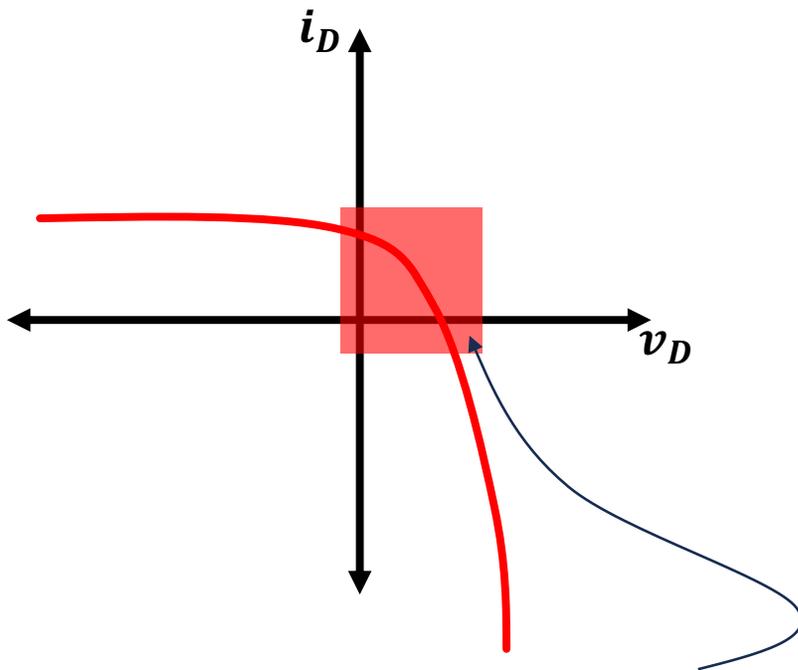
Zoooooooooooom in on this region!
Since that's what we care about

Operating a PV Cell

- A PV cell is *absolutely not* a voltage source nor current source… it is a complex thing with its own wants and needs as expressed through its I-V curve

- It will not give you the power you want in the form *you* want. It will give you power in a form that it wants (certain I and V) and maybe that isn't what you want.

- But this is how relationships work
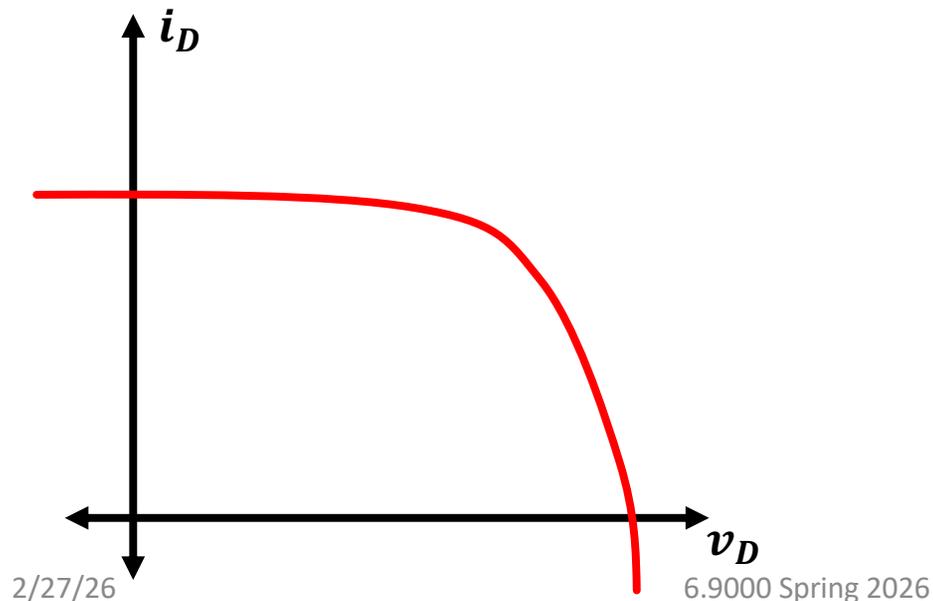
PV Curve

- Extracting the most power from a PV Cell requires using it at exactly one point on its I-V curve.

- Using it at anywhere else, will be a non-optimal efficiency.

- This is called the Maximum Power Point*

No power delivered
At these extrema
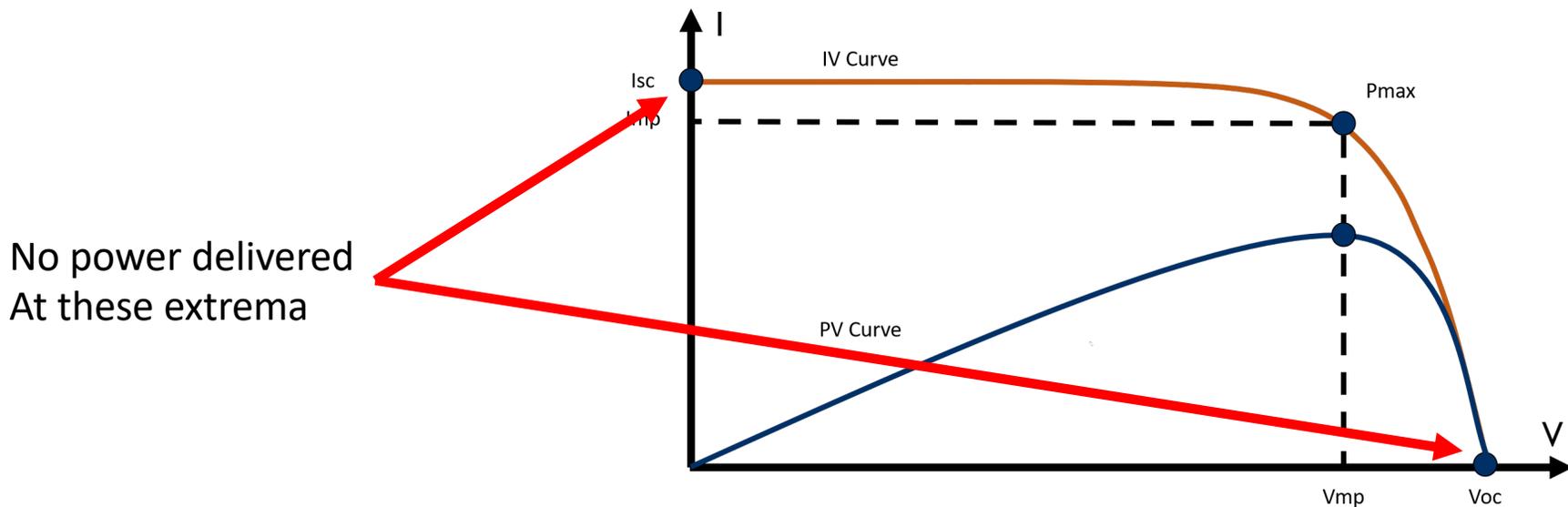
MPPT

- To get the most out of our PV cell, we need to be running it at its Maximum Power Point (MPP).

- Ideally we'd even want to to "track" that MPP. We'll call this MPP Tracking or MPPT

To MPPT or not to MPPT?

- It might sound like a no-brainer to do this. There are chips and things for this:
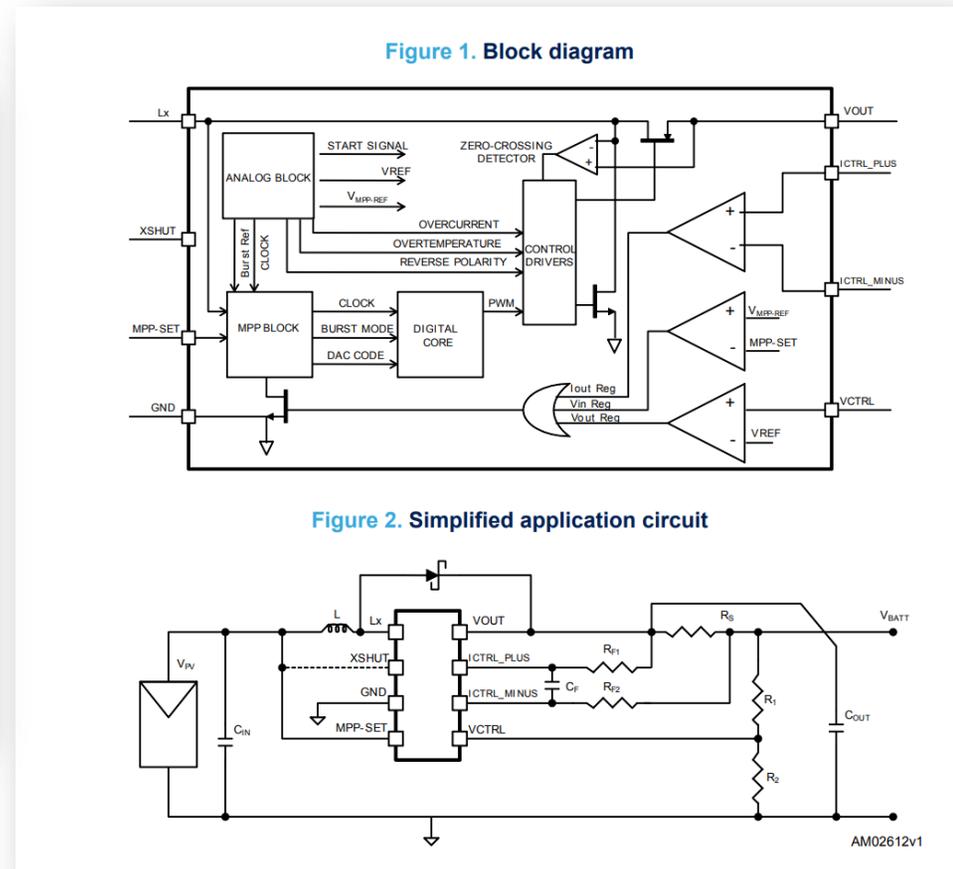
To MPPT or not to MPPT?

- But these chips and their supporting circuitry cost money, and you may only be improving your operating efficiency by ~10% in doing so.

- You have to ask if this is worth it or not.

- Do you have lots of sun so you can use a PV cell inefficiently? Or are you just scavenging enough to breathe and so can't afford to waste anything?

- Also what is the cost/benefit of just buying a larger cell (and continuing to use it at a similar efficiency) vs. using a smaller one more efficiently?

- There's no single right answer so needs experimenting.

And Interestingly…

BQ25185 1-Cell, 1A Standalone Linear Battery Charger with Power Path, Factory Mode, and Battery Tracking VINDPM

- Battery tracking input voltage dynamic power management (VINDPM) for high-impedance input sources

- The input-tracking capabilities actually get near-MPPT-like performance from many solar cells

Computation/Power

6.9000 Spring 2026

All Computation Requires Power

- Power is related to energy
- All computation uses energy
- For a given computational technology…
  - The more computation you do, the more energy you use
  - The faster you do your computation, the more energy used per unit time, the more *power* your system uses

Lower Limit on Computation

- There is a lower limit: it takes about $\mathbf{3{\times}10^{-21}}$ **Joules** to erase a bit no matter what
    - Called Landauer Limit
    - Experimentally shown in 2012 (Berut et al., Nature 2012)

- Modern architectures are taking approximately
    - $\mathbf{1{\times}10^{-15}}$ **Joules (estimate/approximation)**
- Between those two numbers are the ***inefficiencies*** and ***limitations*** of circuits

https://spectrum.ieee.org/computing/hardware/landauer-limit-demonstrated
https://en.wikipedia.org/wiki/Landauer%27s_principle

What Are Those _Inefficiencies_ of Modern Digital Circuits?

- *Why* do our modern computing devices use the power that they do?

- Important to remember the famous statement… "All Models are wrong…some models are useful."

https://en.m.wikipedia.org/wiki/File:CMOS_NAND.svg

Basic Modern CMOS NAND Gate
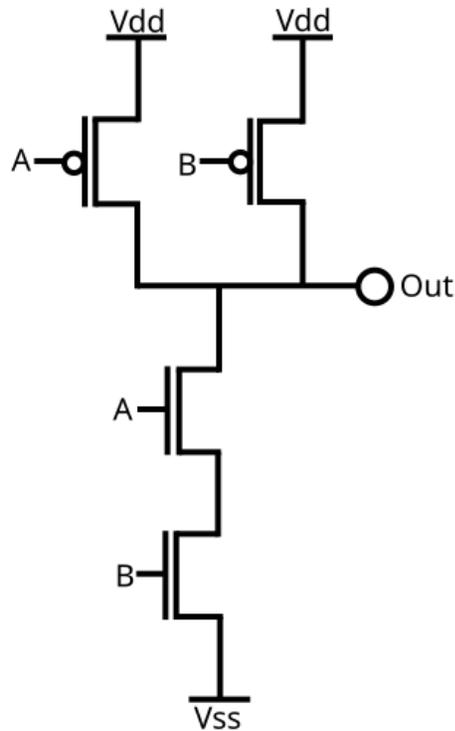
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

P-Channel FET

S

G

D

N-Channel FET

D

S
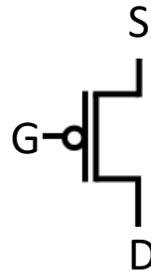
https://en.m.wikipedia.org/wiki/File:CMOS_NAND.svg

Two Types of Transistors

- Both Types of transistors can act as voltage-controlled-switches.
- Two Transistors work in opposite...dare I say... "**C**omplementary" ways

P-Channel FET

S

G⊸

D
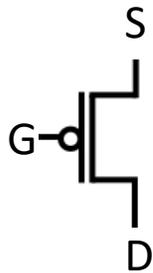
When $V_G = 0$, device conducts

S

G=0

D
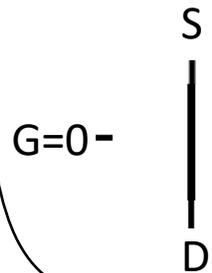
When $V_G = 1$, device open

S

G=1

D

N-Channel FET

D

G⊣

S
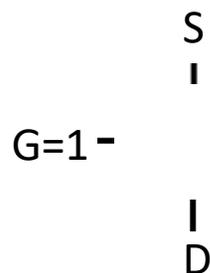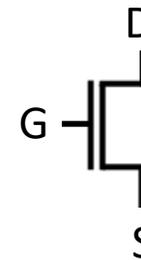
When $V_G = 0$, device open

D

G=0

S
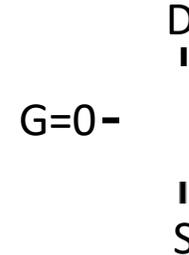
When $V_G = 1$, device conducts

D

G=1

S
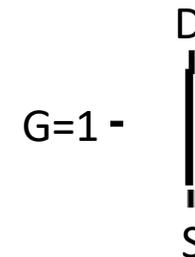
Theoretical NAND Gate Operation
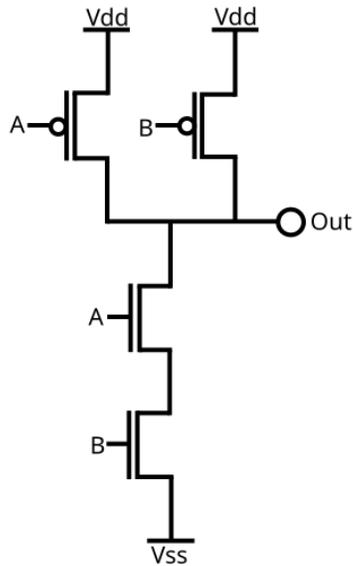
A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0
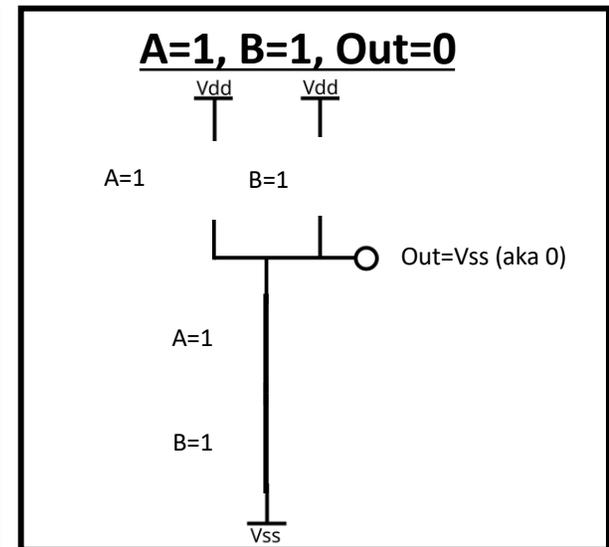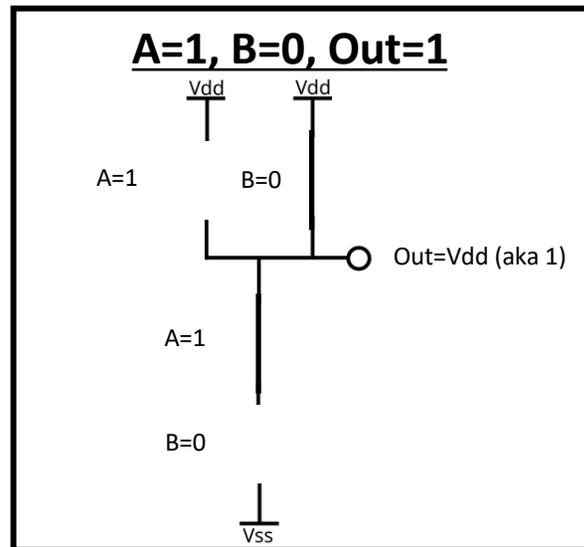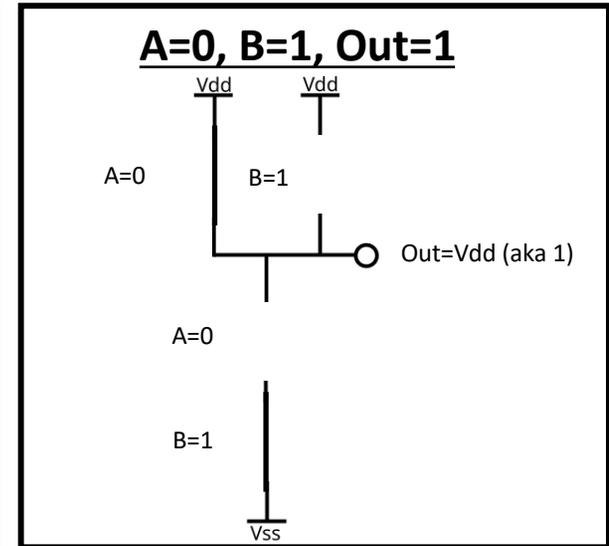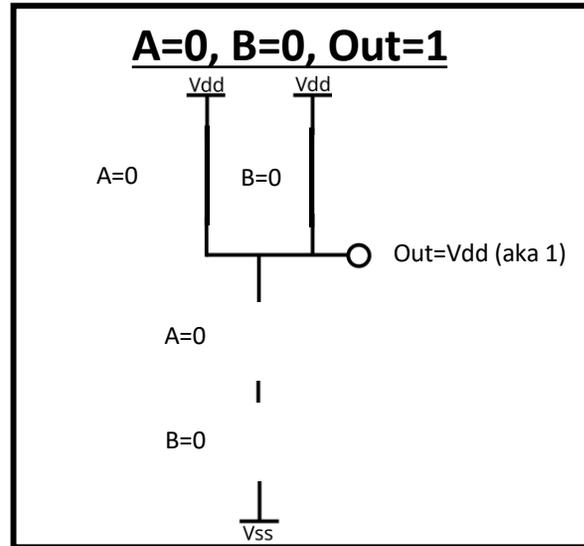
Characteristics…

- In *None* of these four circuit states is there a direct path from Vdd to Vss

- So one should expect no current flow in the circuit itself

- Therefore no *static power consumption*

A=0, B=0, Out=1
Vdd   Vdd
A=0   B=0
Out=Vdd (aka 1)
A=0
B=0
Vss

A=0, B=1, Out=1
Vdd   Vdd
A=0   B=1
Out=Vdd (aka 1)
A=0
B=1
Vss

A=1, B=0, Out=1
Vdd   Vdd
A=1   B=0
Out=Vdd (aka 1)
A=1
B=0
Vss

A=1, B=1, Out=0
Vdd   Vdd
A=1   B=1
Out=Vss (aka 0)
A=1
B=1
Vss
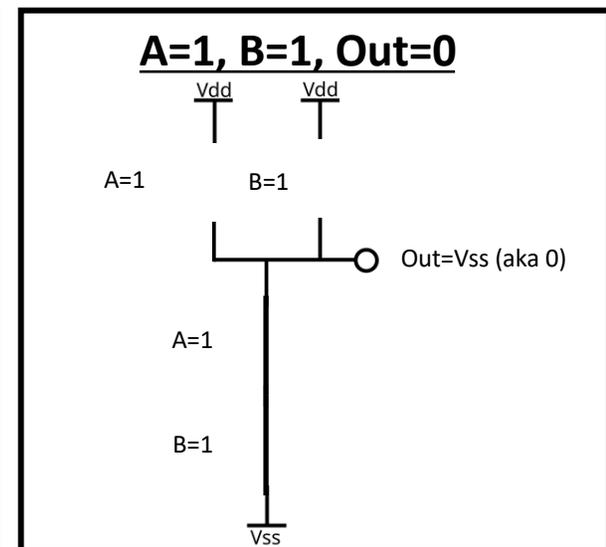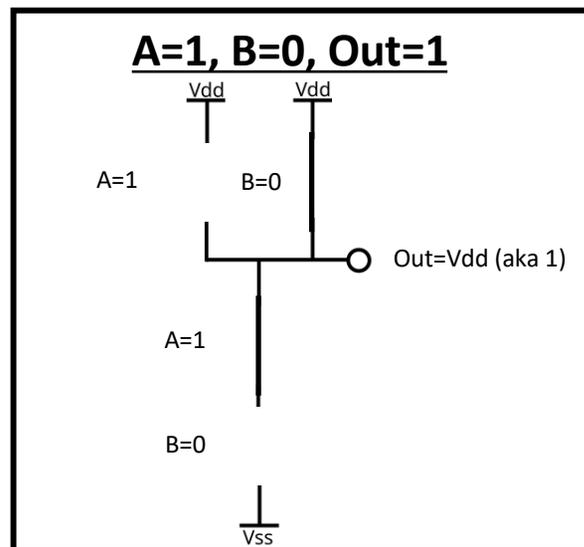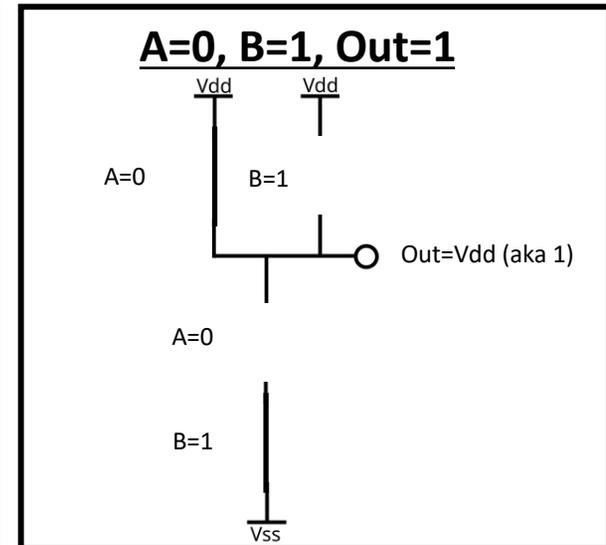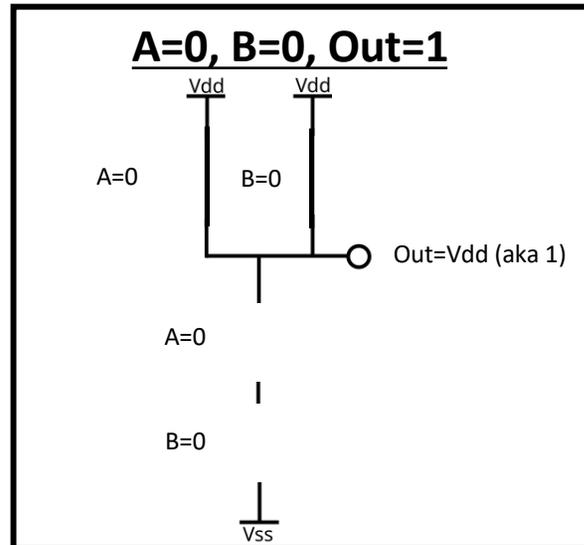
Characteristics...

- It takes no energy to turn on/turn off a switch... They're "massless"
- We can transition between states instantaneously
- Therefore no ***dynamic power consumption***

A=0, B=0, Out=1
Vdd   Vdd

A=0   B=0

               ○ Out=Vdd (aka 1)

A=0

B=0

Vss

A=0, B=1, Out=1
Vdd   Vdd

A=0   B=1

               ○ Out=Vdd (aka 1)

A=0

B=1

Vss

A=1, B=0, Out=1
Vdd   Vdd

A=1   B=0

               ○ Out=Vdd (aka 1)

A=1

B=0

Vss

A=1, B=1, Out=0
Vdd   Vdd

A=1   B=1

               ○ Out=Vss (aka 0)

A=1

B=1

Vss
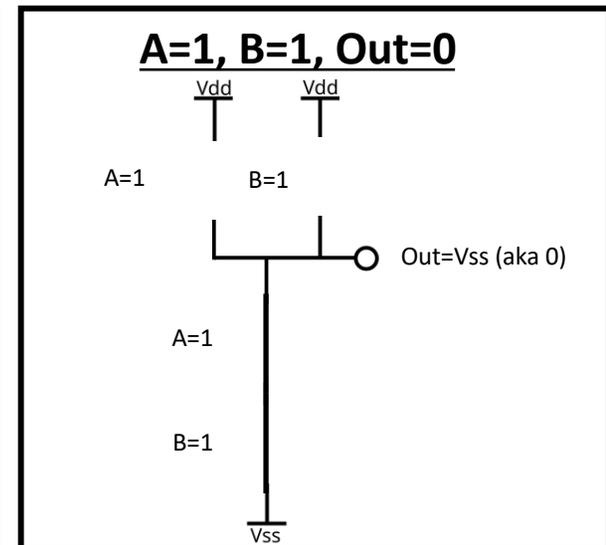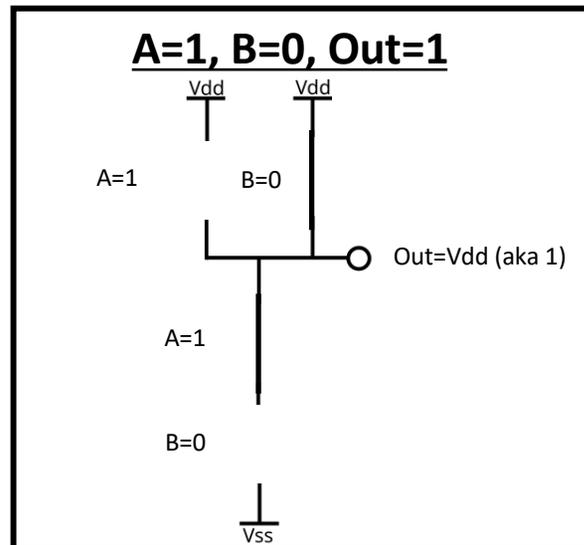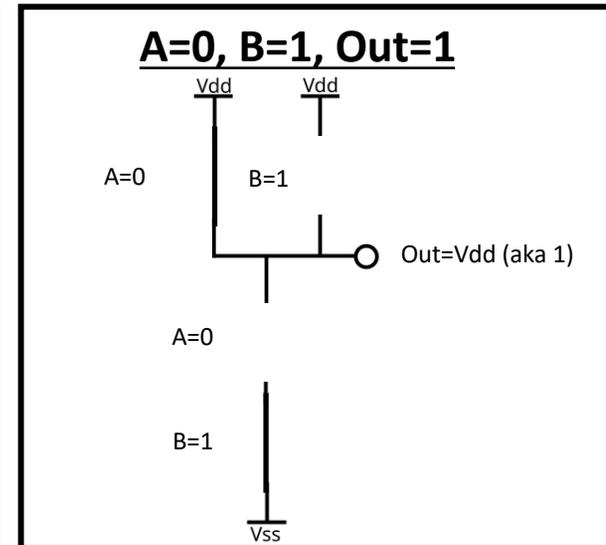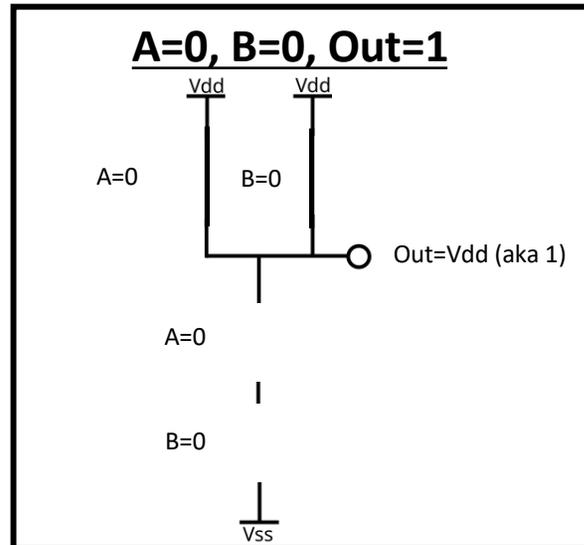
An Ideal Switch I-V Relationship

Open Switch:

i → $\quad$ v $\quad$ + $\quad$ −

i

v

Perfectly horizontal

Closed Switch:

i → $\quad$ v $\quad$ + $\quad$ −

i

v

Perfectly vertical
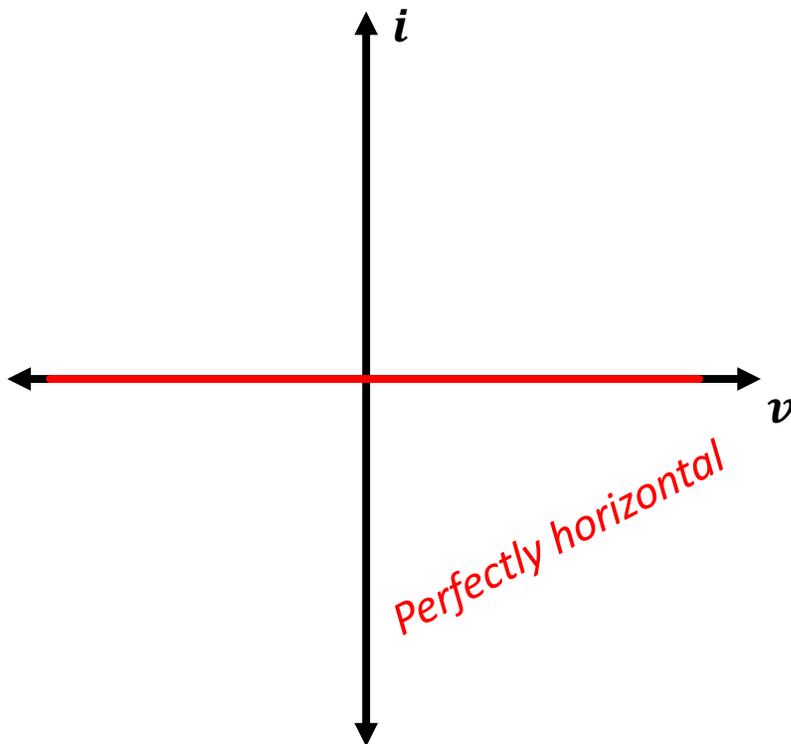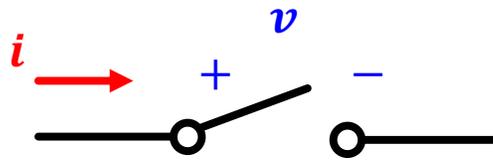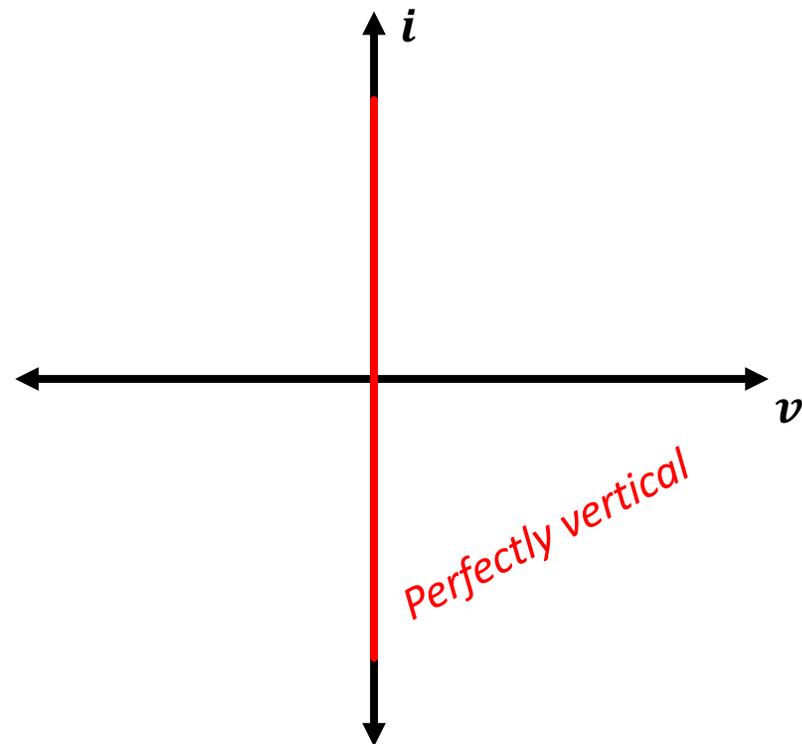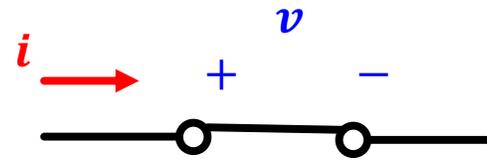
Reality Always Ruins Things Though...

- Transistors Do Not Act Like Ideal Switches
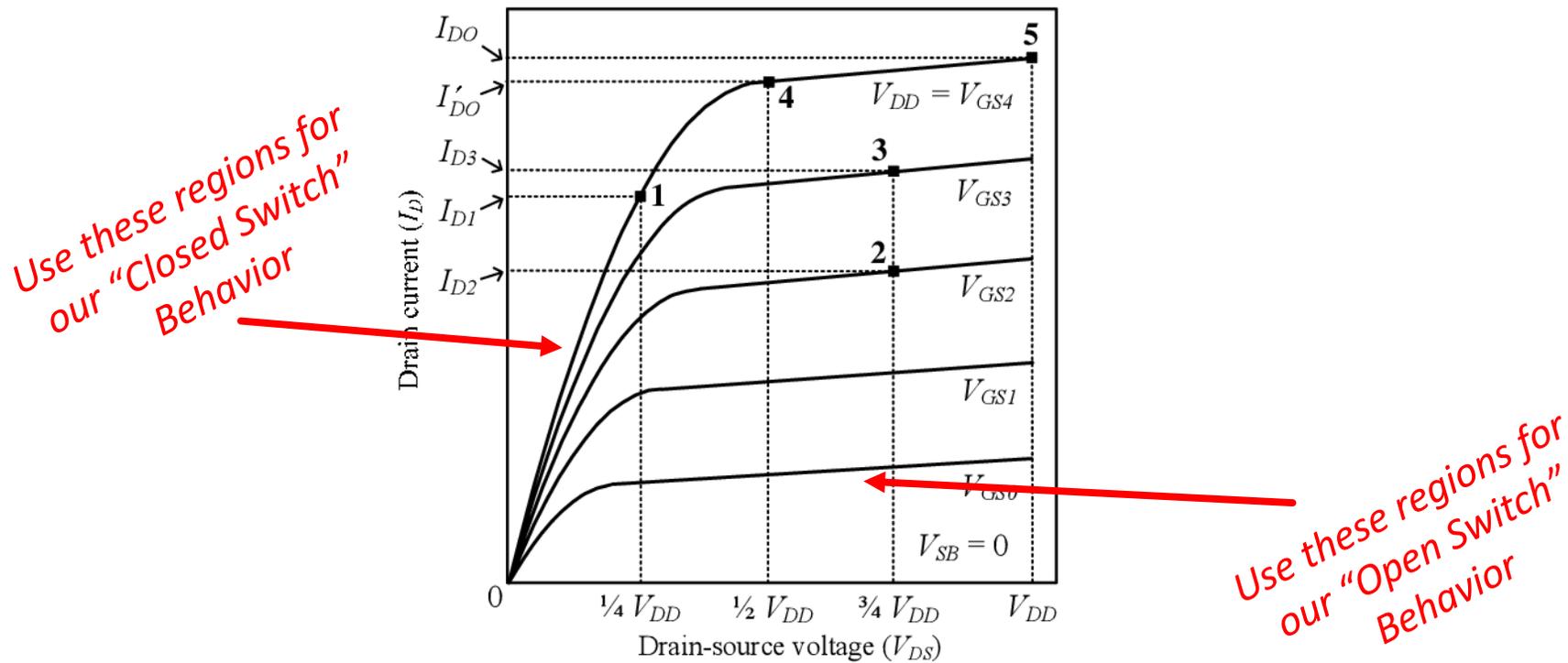
Figure 1. Selected points for NMOS parameters extraction

https://www.semanticscholar.org/paper/An-Accurate-and-Compact-MOSFET-I-V-
Model-for-CMOS-Bisdounis/edcd356940b102e895b83ad2e7ebcbca499710e0

Also…A MOSFET

Gate

Source

Drain

Physical Design:

Gate
Conductive

Oxide
(insulator)

Source
Conductive

Channel
(variably conductive)

Drain
Conductive

Substrate (insulator)

Input (Gate) of a MOSFET

Schematic Symbol:

Gate

Forms capacitor

CAPACITIVE!

Metal
Oxide
Semiconductor

Source

Drain

Conductor
Insulator
Conductor

Gate
Conductive

Oxide
(insulator)

Source
Conductive

Channel
(variably conductive)

Drain
Conductive

Substrate (insulator)
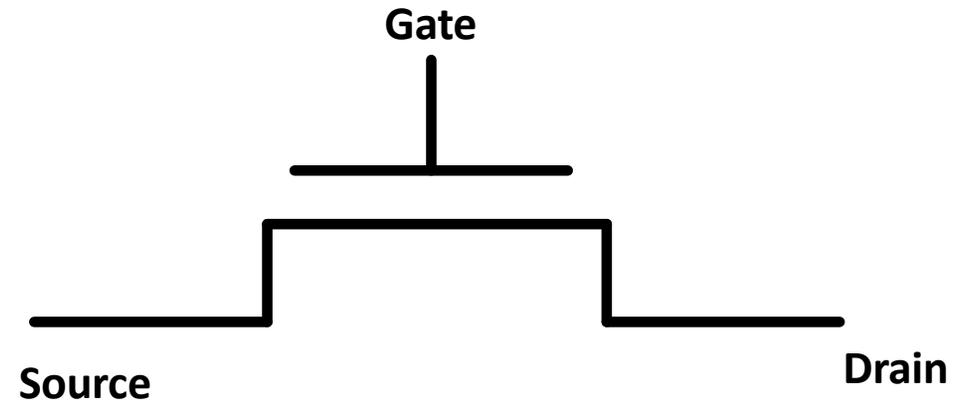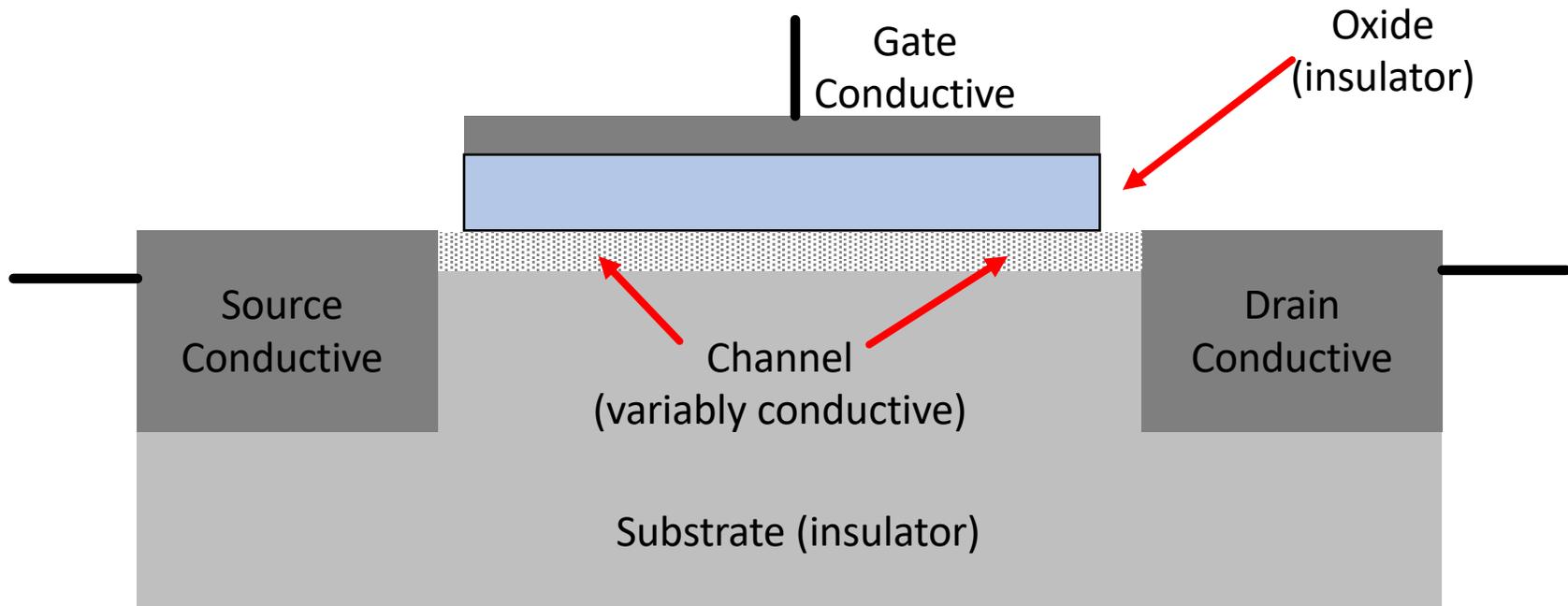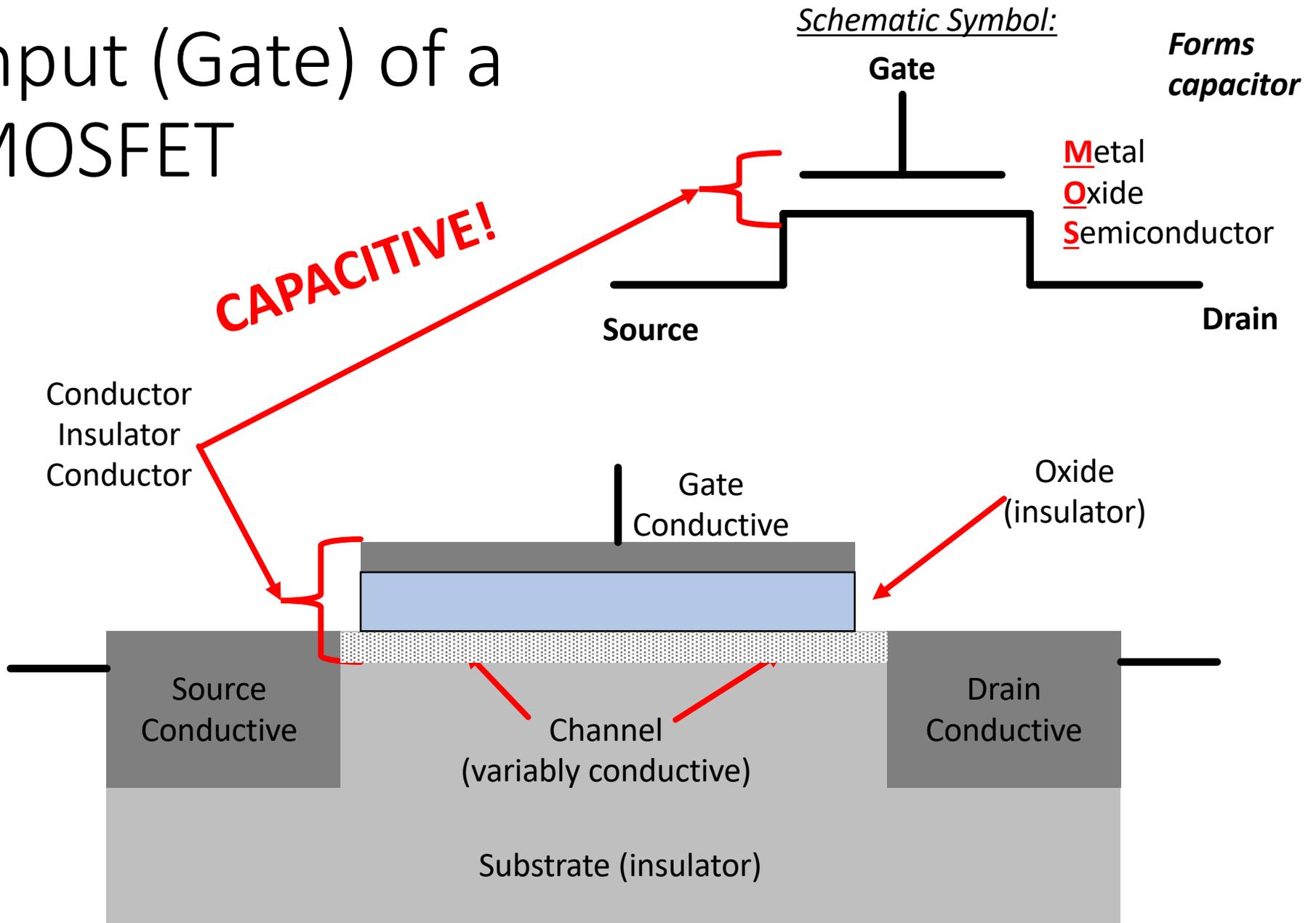
So when we start to build complicated digital circuits...
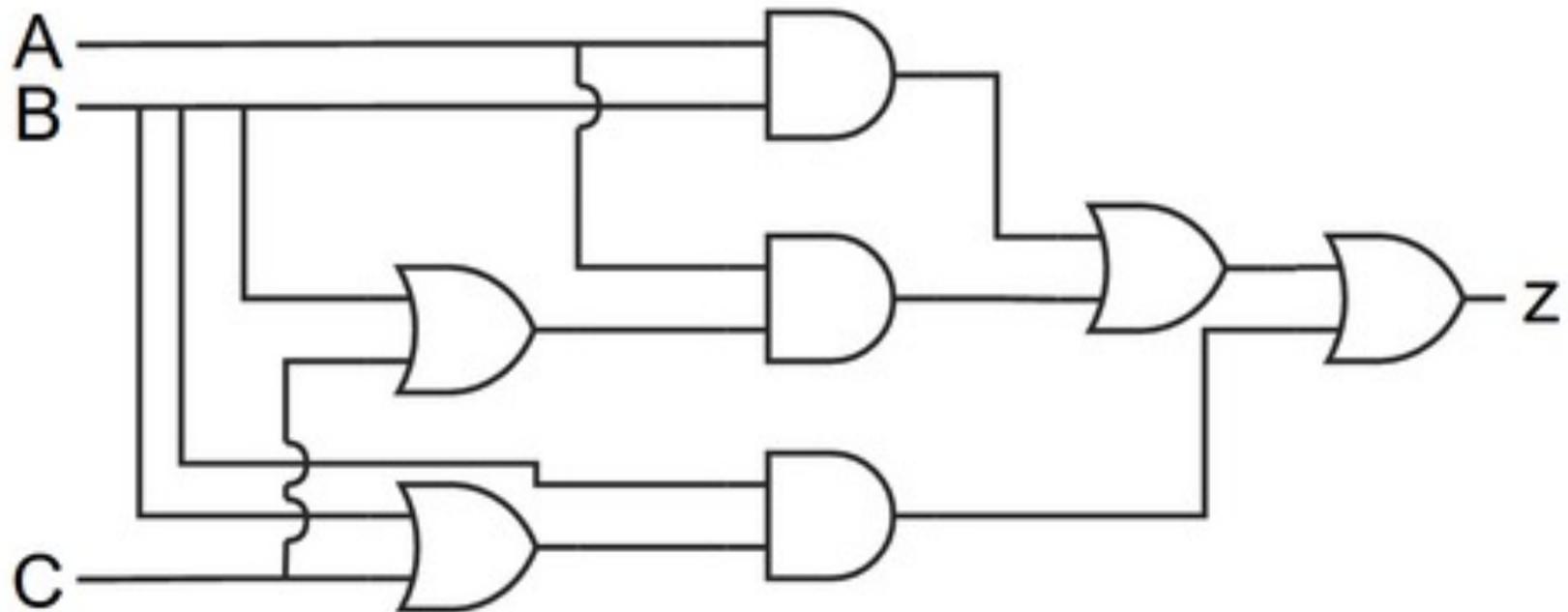
CMOS Circuit drives another CMOS Circuit...
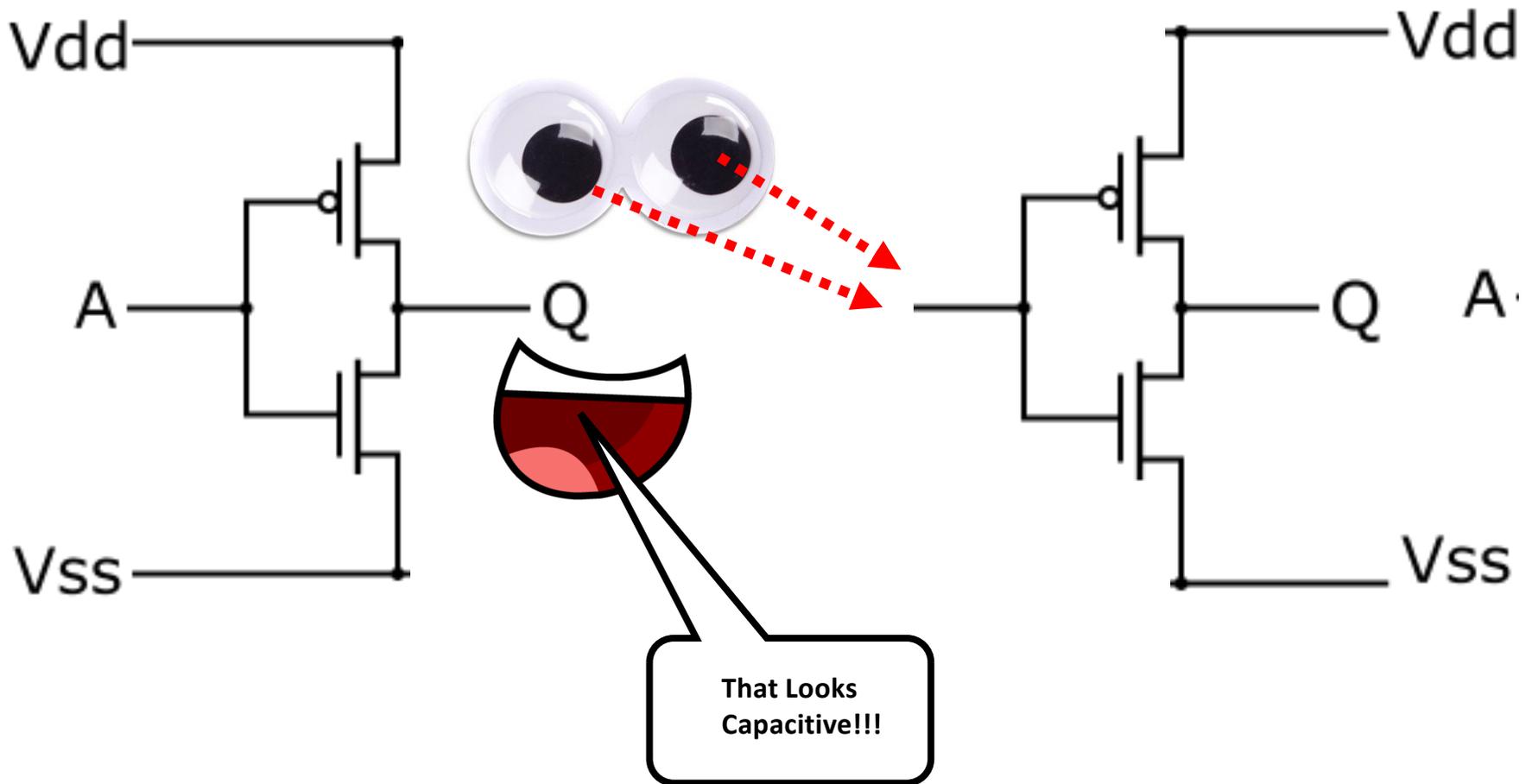
It is CMOS all the way down

CMOS Circuit drives another CMOS Circuit...

It is CMOS all the way down

C_L represents the next stage's input capacitance

I am an imperfect voltage source

CMOS Circuit drives another CMOS Circuit...

It is CMOS all the way down

R_P

V_S

C_L

C_L represents the next stage's input capacitance

I am an imperfect voltage source
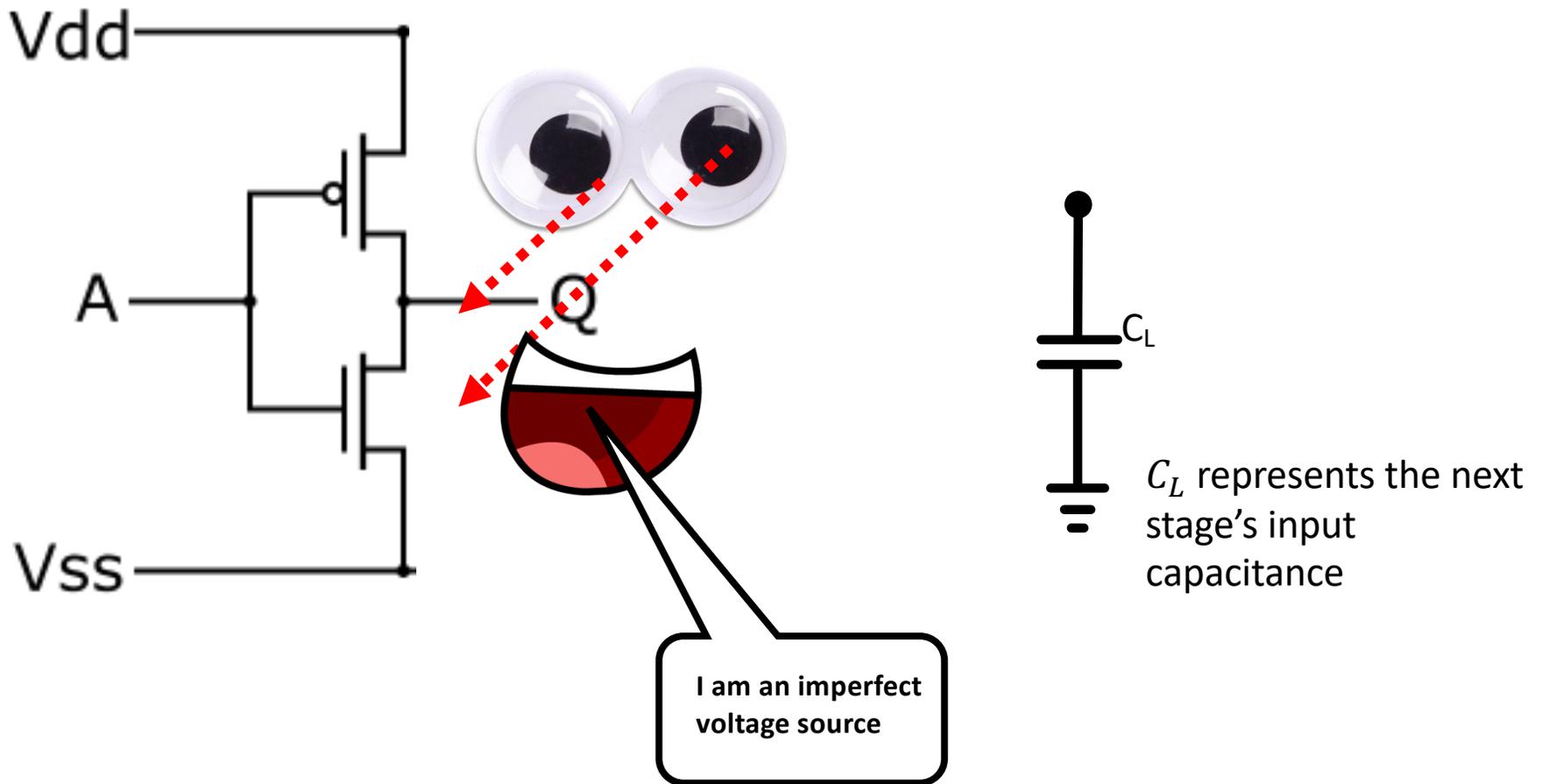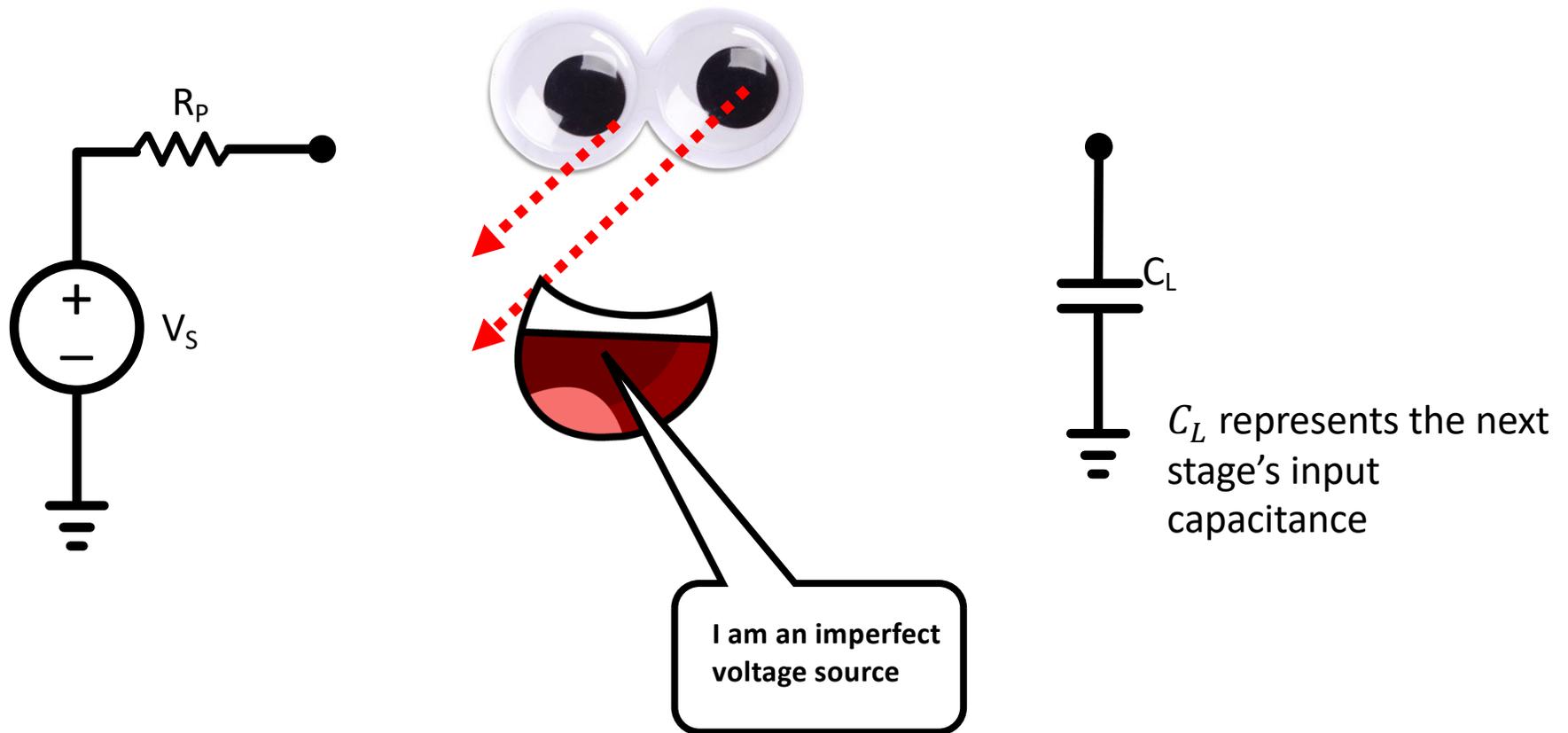
Implications

- If a circuit has to charge or discharge up a capacitive input that means…:

- There is an inherent speed limit from RC time constants that will limit the switching speed

- The fact that you have to move charge onto that capacitor through a resistor means energy is needed…and that energy will never get recovered.

Takes energy to charge <u>up</u> capacitors (Dynamic Power Consumption)

conducts ....

V_{DD}

Non-ideal conductance of transistor channel means some power lost as heat during charging of capacitor

1

V_{IN}

0

I

V_{OUT}

1

0

C_L

doesn't conduct

C_L represents the next stage's input capacitance
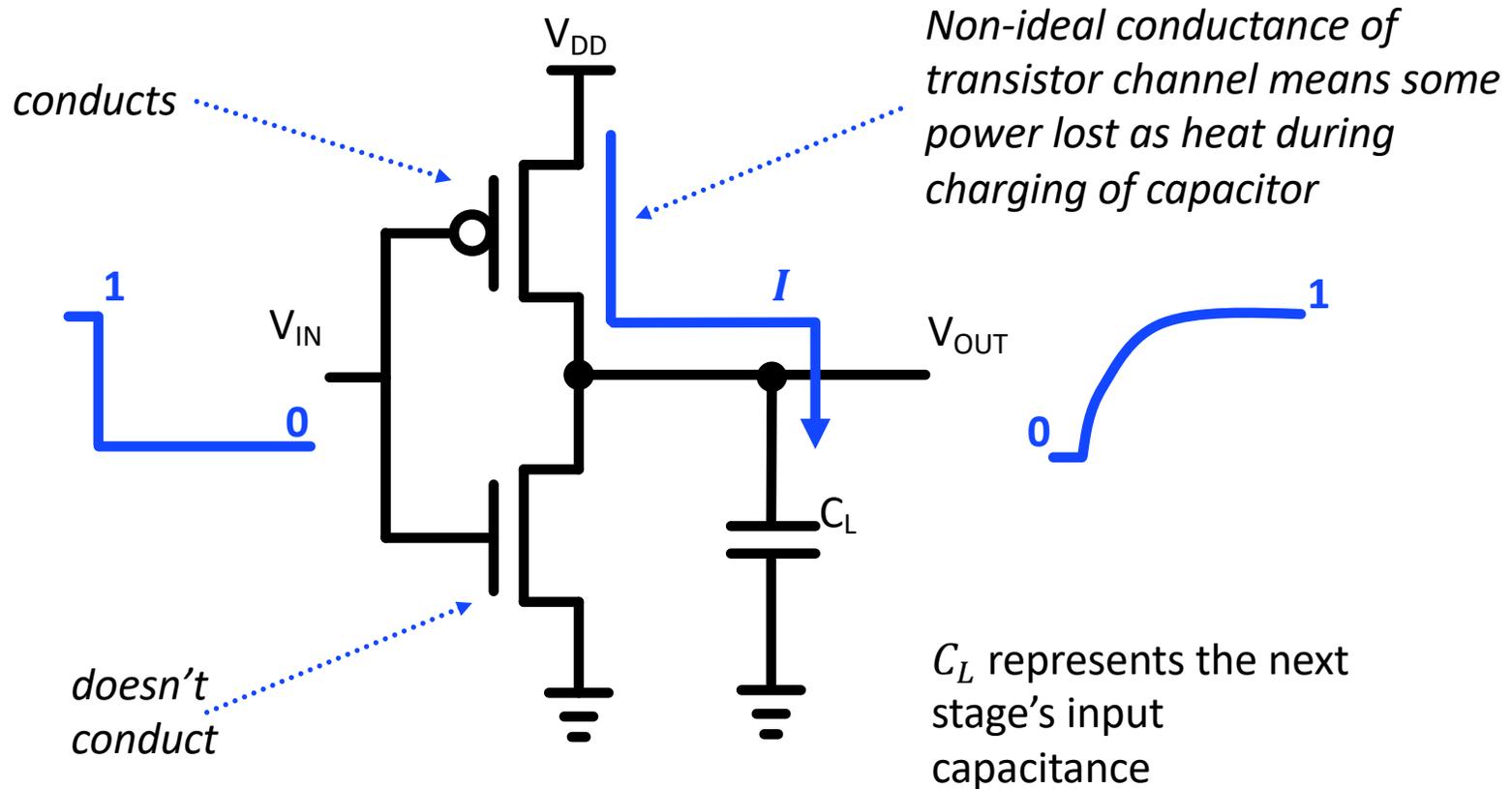
Takes energy to charge _down_ capacitors (Dynamic Power Consumption)

V_{DD}

doesn't conduct

1

V_{IN}

0

I

V_{OUT}

1

0

C_L

Non-ideal conductance of transistor channel means some power lost as heat during charging of capacitor

conducts

C_L represents the next stage's input capacitance
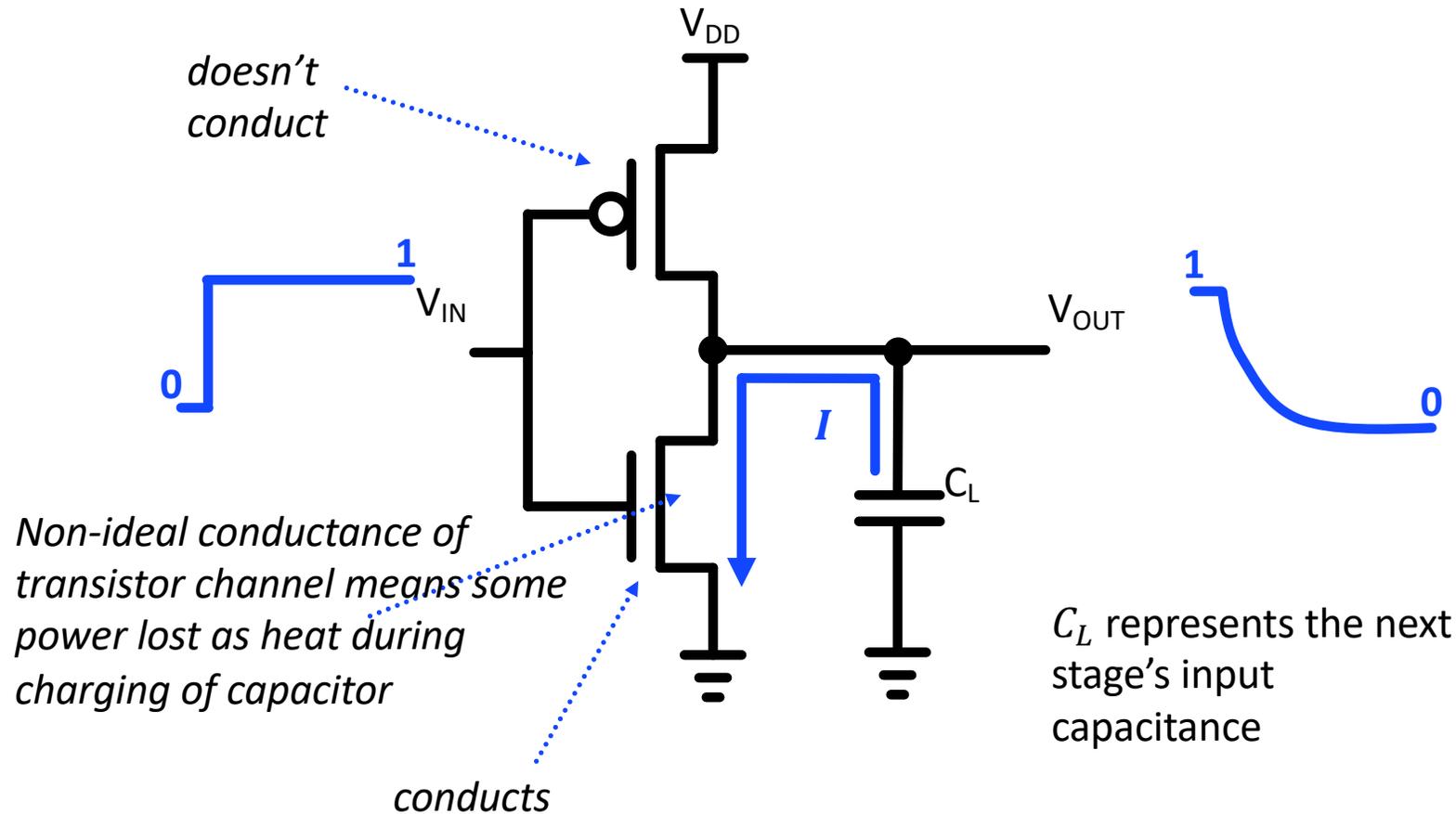
During Transition, there may be overlap of conductances

V_{DD}

Kinda conducting

Both cooking off heat during transition

I_{peak}

V_{IN}

1

0

V_{OUT}

1

0

C_L

*Transitions are **never** instantaneous*

Kinda conducting

C_L represents the next stage's input capacitance
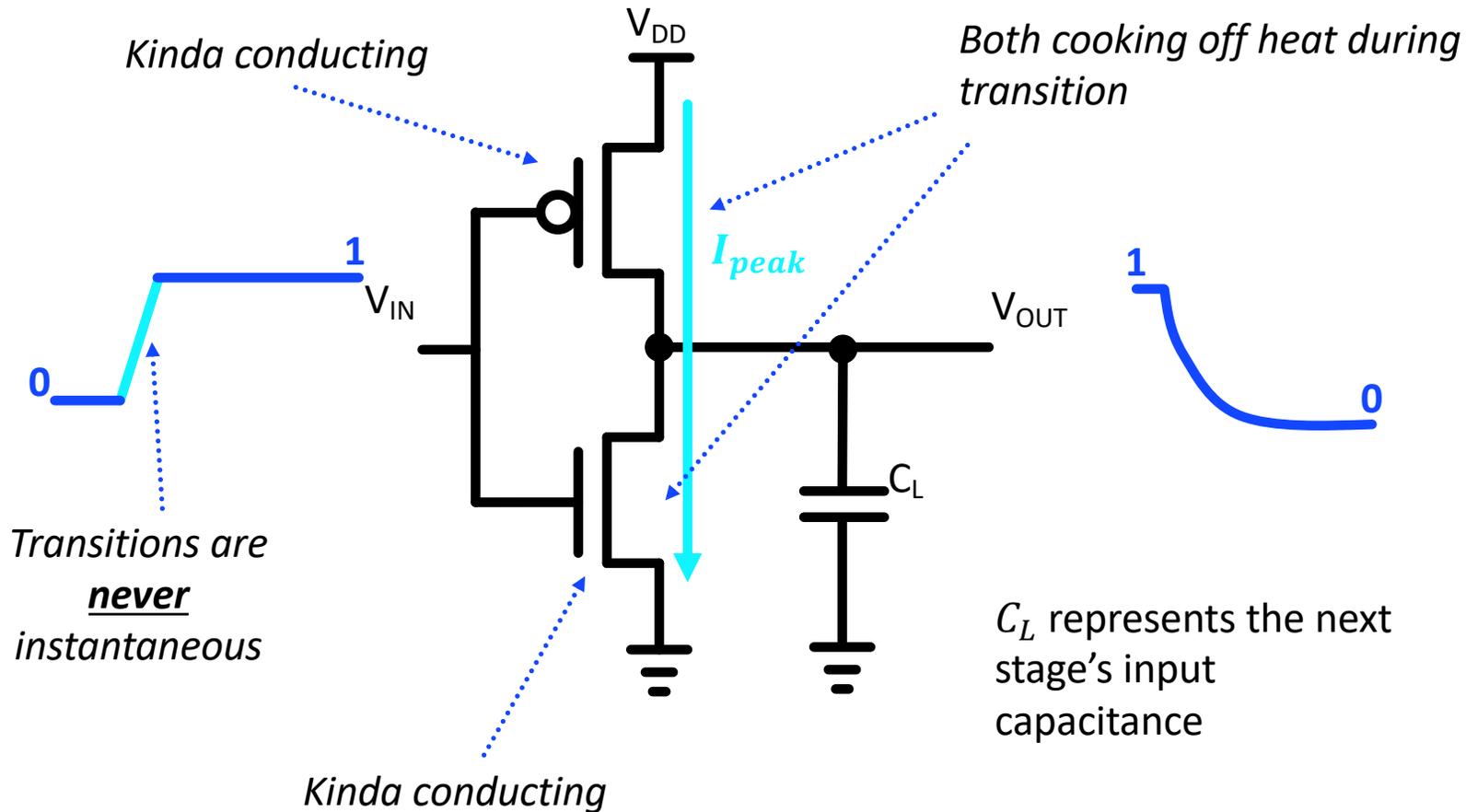
Dynamic Power Consumption
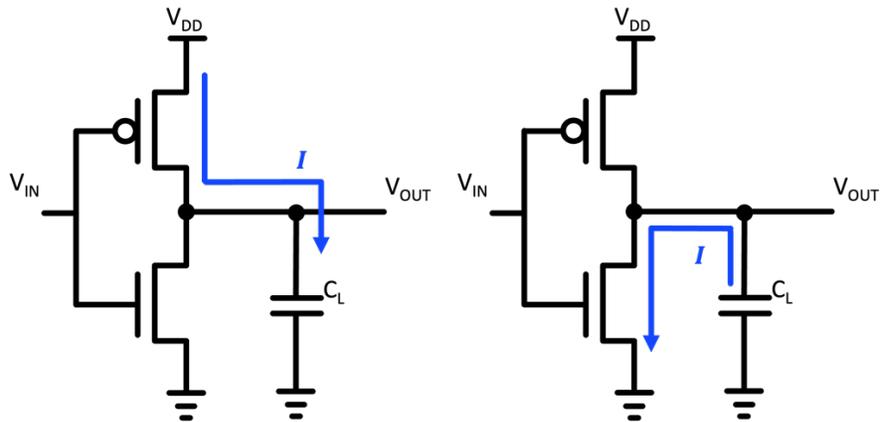
Capacitive Dis/Charging

- Caused by need to store up finite charge
- $P \propto CV^2 f$
- C: capacitance of gate
- V: V_{DD} of system
- f: frequency of switching
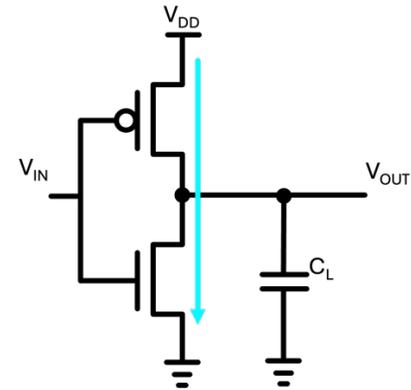
Short Circuit

- $P \propto t_{sc} V I_{peak}$
- t_{sc}: in crossover
- V: V_{DD} of system
- I_{peak}: Max current at crossover
- Good news this is usually rather small compared to **capacitive**

The Downward Scaling of Transistors

- Most gains in computation have arisen from making smaller transistors...

- Fit more on a chip.

- They use less power (less gate capacitance!!!)

- Can do more for less!

Moore's Law: The number of transistors on microchips doubles every two years

As MOSFETs get Very Small…

Oxide
Acts less and less like an insulator

Gate
Conductive

Source
Conductive

Drain
Conductive

Channel
Acts less and less "Off" when
in "Off" mode

Static Power Consumption

Transistors start to not be as ON-OFF as they were when bigger…

V$_{DD}$

I_{gate}

Gate Leakage

I_{sub}

V$_{IN}$

V$_{OUT}$

Gate Leakage

I_{gate}

C$_L$
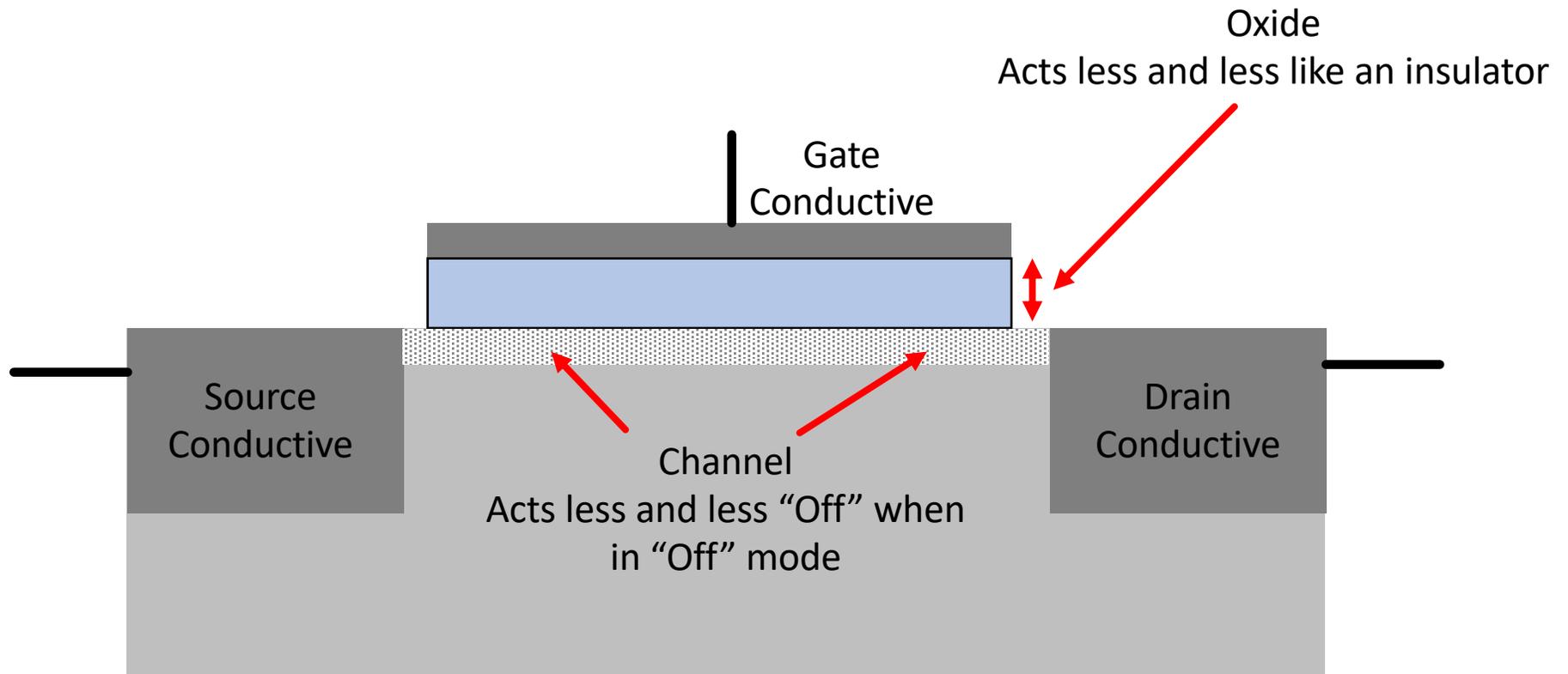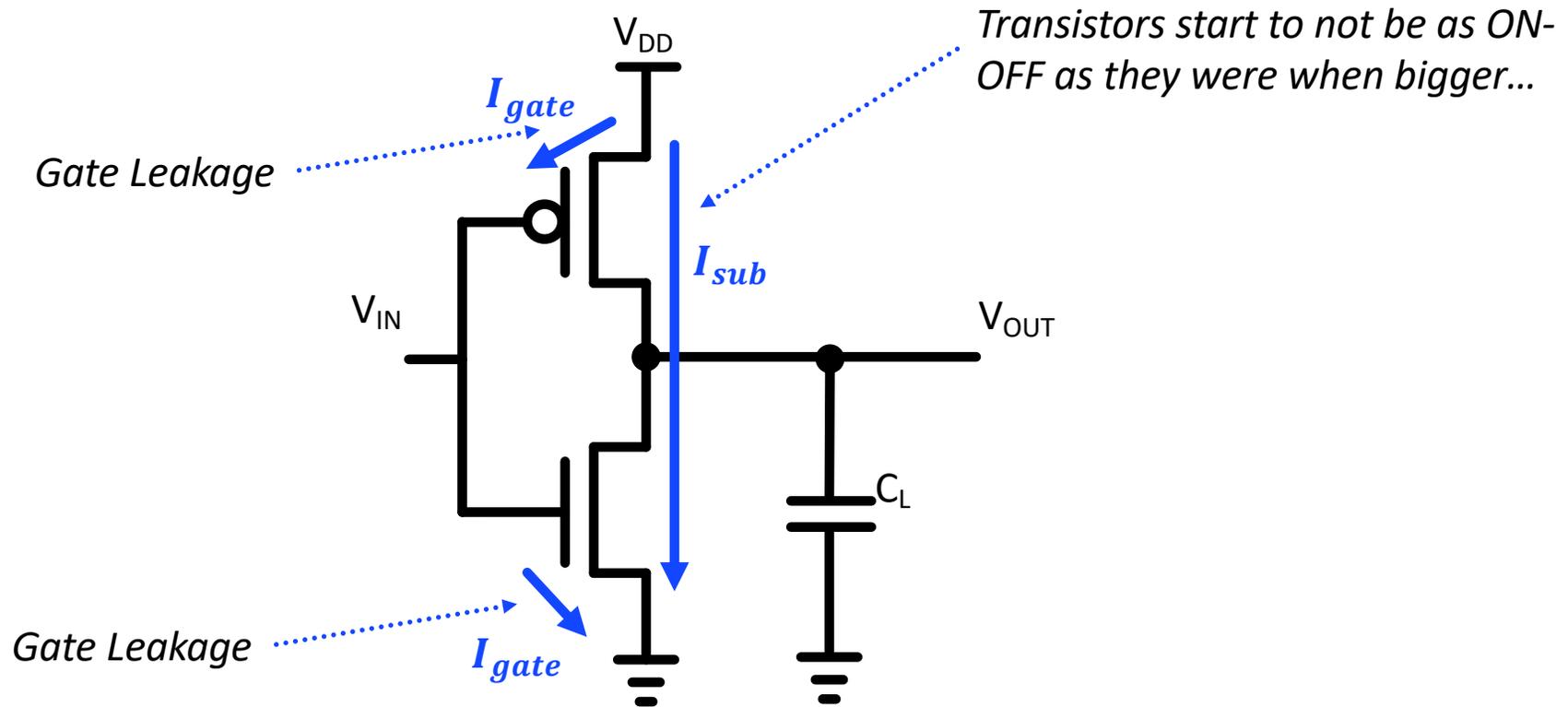
Static Power Consumption

- **Gate Leakage:** The gates don't act like perfect insulators so you leak current (power) through them

- **Sub-threshold Leakage:** The transistors don't turn On and off as sharply, so there's more mushing and overlap and start to conduct significant amounts all the time (wasted leaks)

Static Power Losses

- As opposed to dynamic power losses which arise from the act of switching bits...

- Static losses take place just by being powered up and existing.

- So the total power consumed ends up being roughly explained by this equation:

$$P_{total} = P_{dynamic} + P_{static}$$

Two Major Ways Power is Consumed on a Device

- **Dynamic Power Consumption:** By flipping bits in the process of calculations, you will be burning energy. The more bits flipped, the more energy burned.

- **Static Power Consumption:** Just by "being on" you're eating power...even if you're not doing anything.

Summary: Digital Power Consumption

- P : total power consumed
- $\alpha_{0\to1}$: fraction of gates switching
- C : Capacitance of gates, busses, interconnects
- V : Operating voltage (V_{dd})
- f : frequency of operation
- I_{leak} : Leakage Current:
  - Sub-threshold leakage
  - Gate-Leakage

Dynamic power consumption

$$P = \alpha_{0\to1}CV^2f + VI_{leak}$$

Static power consumption

How to make P go down?

Drop system voltage

Improve Fabrication

$$P = \alpha_{0 \to 1} CV^2 f + VI_{leak}$$

Cut down on bits flipping

Smaller capacitors, Keep wires/inter connects short!

Scale down system voltage

Drop frequency?

Be ON for less time
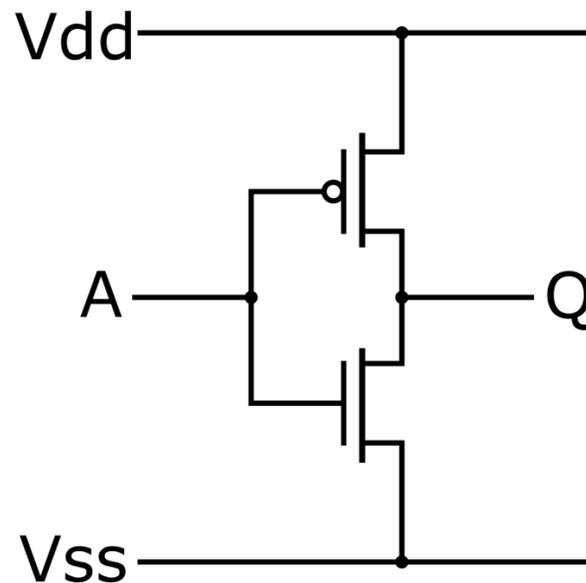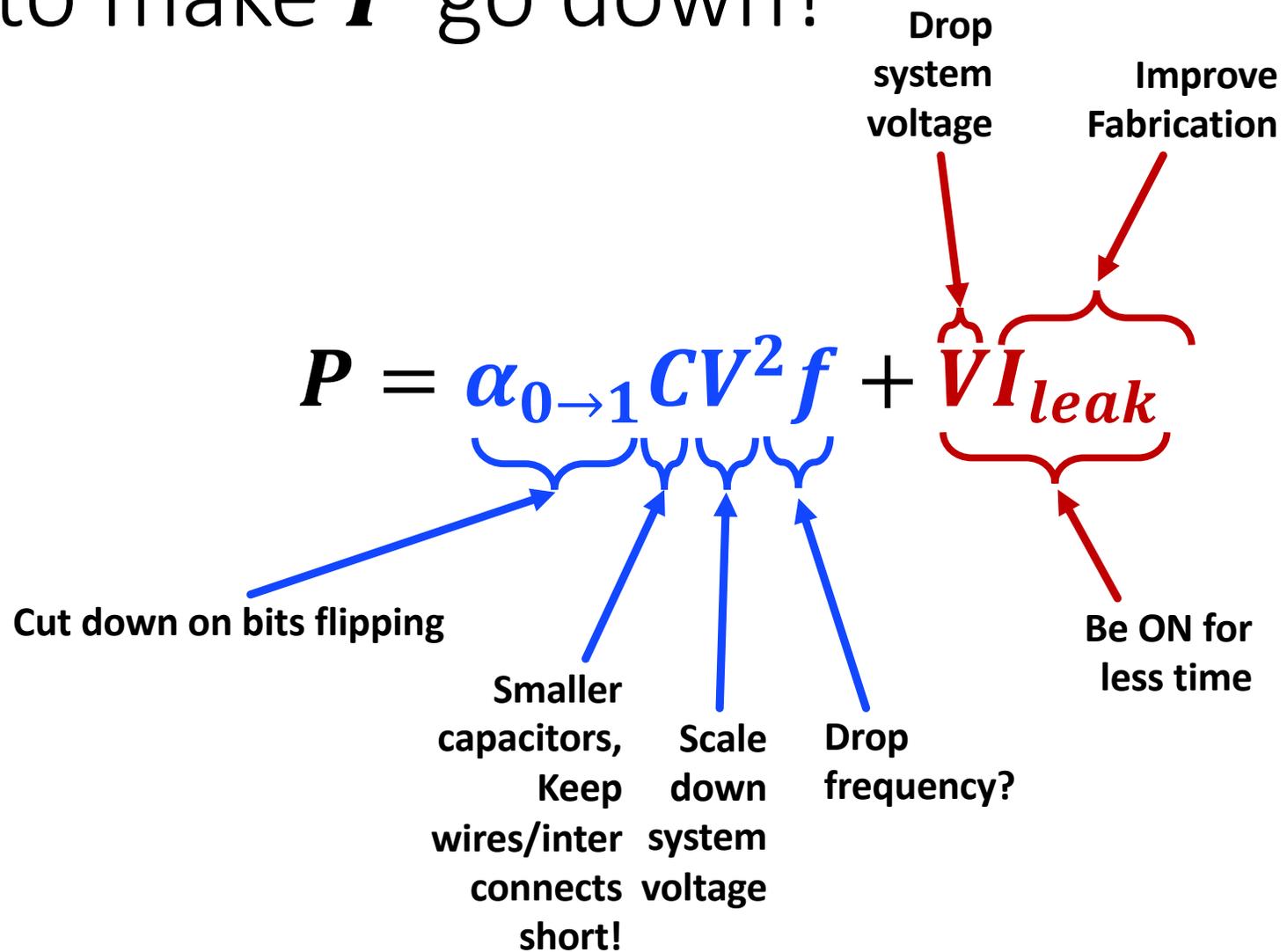
How to Manage?

- If you're designing chips from scratch you have some options:

- **Dynamic Power Consumption:**
  - ***Clock Gating***: Stop the clock that is going to your logic…this will essentially "freeze" the state of your module and prevent if from running in place

- **Static Power Consumption:**
  - ***Power Gating***: Actually _turn off_ your logic…this will basically remove it from your system and not bleed power by just being on.

Clock-Gating
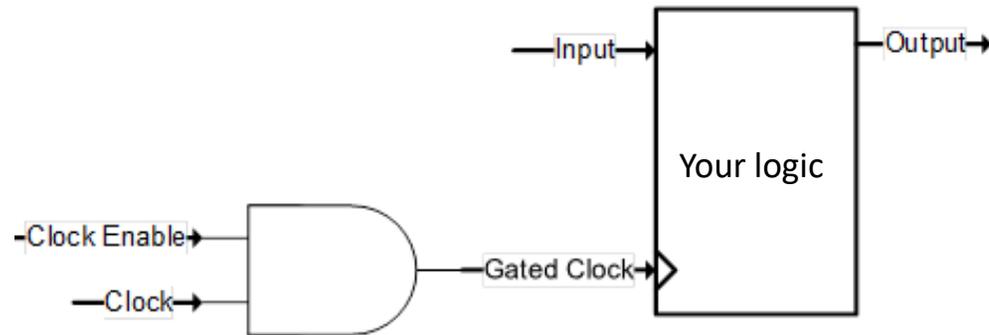
- Since most modern digital logic is synchronous and only updates on the edges of clocks (sequential logic), simply stopping the clock (making its period go to infinity) can "freeze" the system
- The system will not flip bits for no reason
- Relatively easy to implement into chip design
- It is still powered, just frozen.

https://anysilicon.com/the-ultimate-guide-to-clock-gating/

Power-Gating

- More involved than clock gating! You actually deactivate whole portions of the circuit (have large high side or low side enable transistors)

- Takes up a lot more real estate on chip

- Requires a lot more devices to make work…Have to:
  - Isolate the "dead" portion from "live portions"
  - Save and reload state of system before after turnoff
  - Worry about pipelining and propagation of removal/reappearance
  - Longer shutdown and startup times

- But does minimize power wasted!!!!

https://anysilicon.com/power-gating/
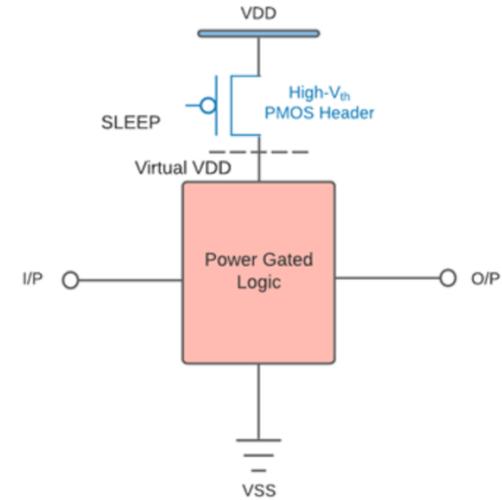
Fig. 1.3: Header Switch Cell
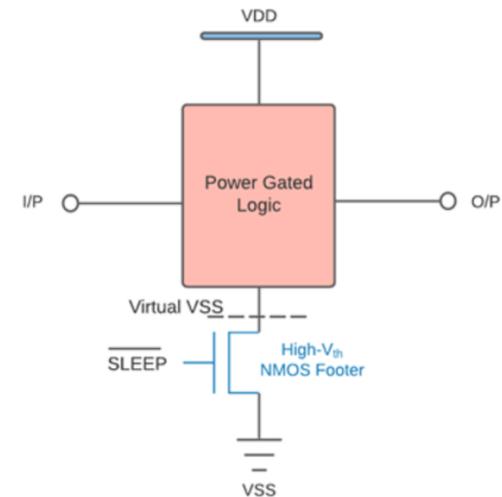
Fig. 1.4: Footer Switch cell

Both Functionalities Are Available To Us in 6.9000!

- The ESP32C3 is, if nothing else, a very power-conscious microcontroller.

- It has both clock gating and power gating of various peripherals that are at your fingertips.

- We need to take advantage of these whenever possible to save on energy

More Generally

- Try to minimize:
  - Time it takes to do something
  - Time we're "ON"
  - If we're "ON" make sure we're doing as much as possible
  - No lollygagging around.

Espressif and ESP32 C3

- Tons of Documentation

- Quite a lot of functionality

ESP32 C3 Power Modes

- Various Portions of the ESP32 C3 SOC can be turned on/off as needed
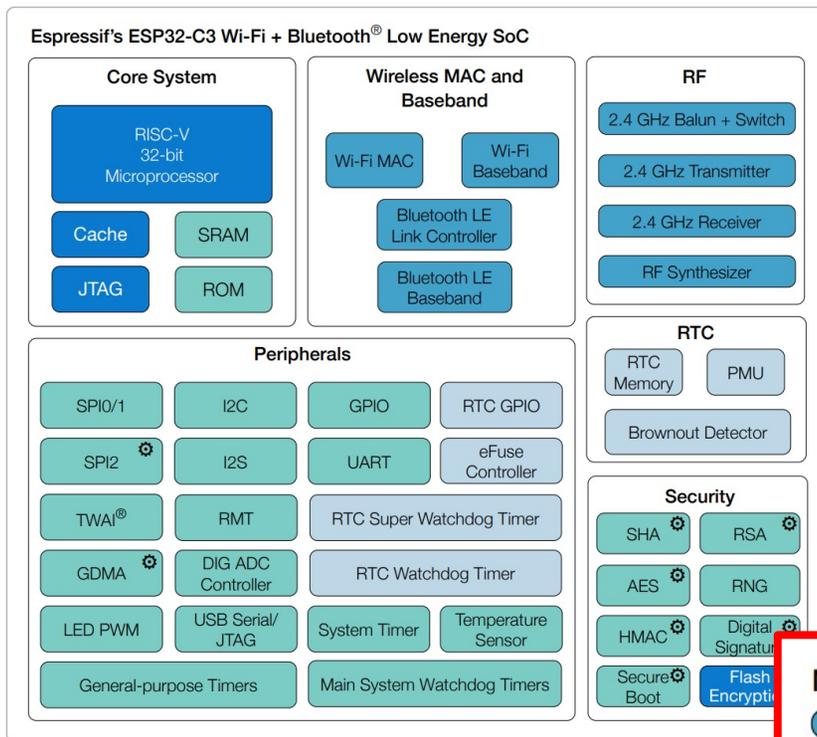
Work mode		Description	Peak (mA)
Active (RF working)	TX	802.11b, 1 Mbps, @20.5 dBm	345
		802.11g, 54 Mbps, @18 dBm	285
		802.11n, HT20, MCS7, @17.5 dBm	280
		802.11n, HT40, MCS7, @17 dBm	280
	RX	802.11b/g/n, HT20	82
		802.11n, HT40	84

Mode	CPU Frequency (MHz)	Description	Typ	
			All Peripherals Clocks Disabled (mA)	All Peripherals Clocks Enabled (mA)[1]
Modem-sleep[2,3]	160	CPU is idle	16	21
		CPU is running	23	28
	80	CPU is idle	13	18
		CPU is running	17	22

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

Modules having power in specific power modes:

- Active
- Active and Modem-sleep
- Active, Modem-sleep, and Light-sleep;   ⚙ optional in Light-sleep
- All modes

Real Hierarchy of Needs Here

All Functionality

Less Functionality

Even Less Functionality

Bare Minimum

Increasing Functionality

Decreasing Power Consumption
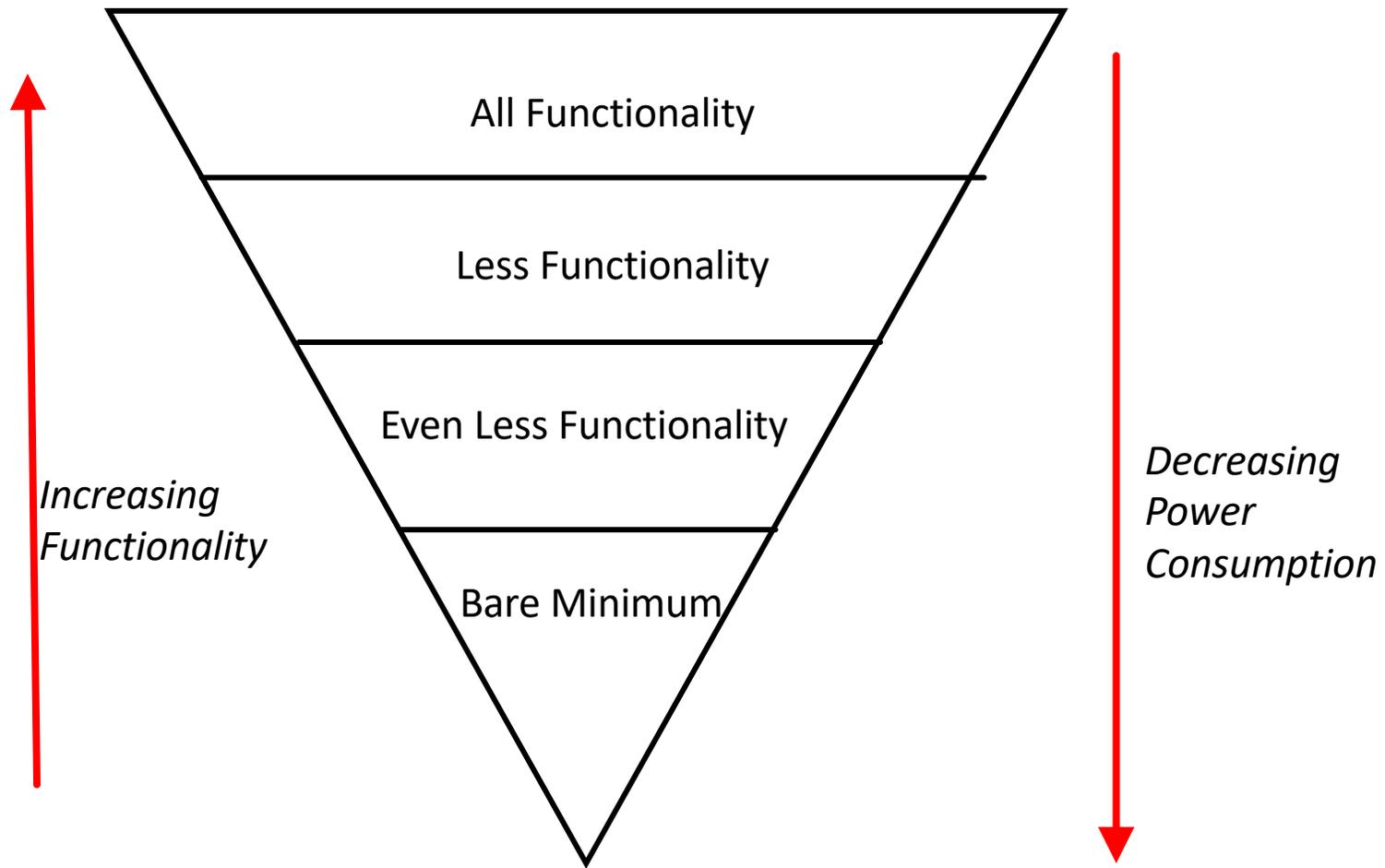
What is On/Off During "Active"

Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC

Modules having power in specific power modes:
- Active
- Active and Modem-sleep
- Active, Modem-sleep, and Light-sleep; ⚙ optional in Light-sleep
- All modes
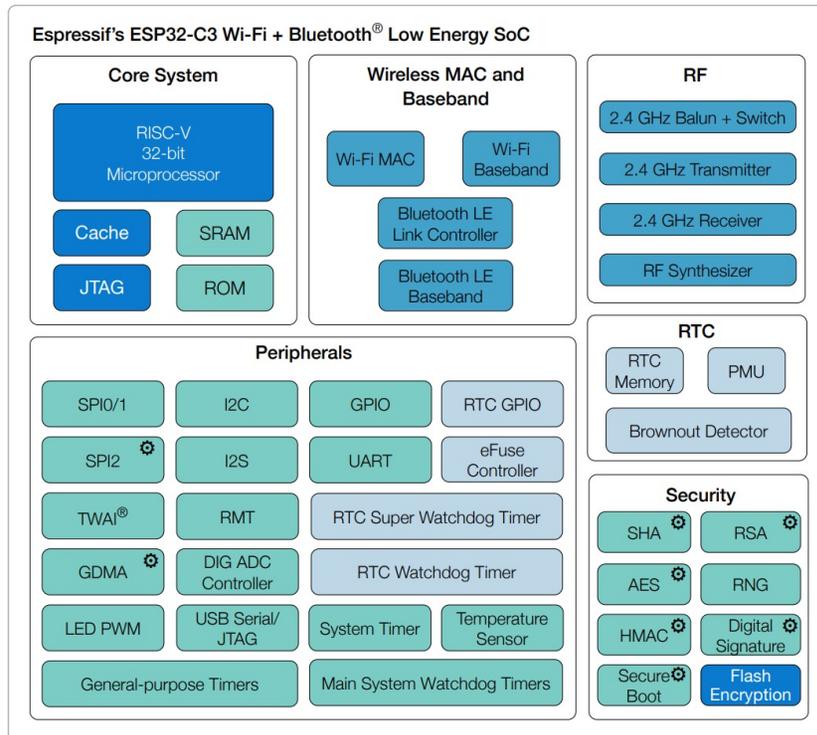
- **Refers to having WiFi and/or BLE enabled**

- **Everything is running**

- **Consumes massive amount of power in various stages (TX uses more than RX...but do need to TX in order to RX)**

Work mode		Description	Peak (mA)
Active (RF working)	TX	802.11b, 1 Mbps, @20.5 dBm	345
		802.11g, 54 Mbps, @18 dBm	285
		802.11n, HT20, MCS7, @17.5 dBm	280
		802.11n, HT40, MCS7, @17 dBm	280
	RX	802.11b/g/n, HT20	82
		802.11n, HT40	84

What is On/Off During Modem-Sleep

Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC
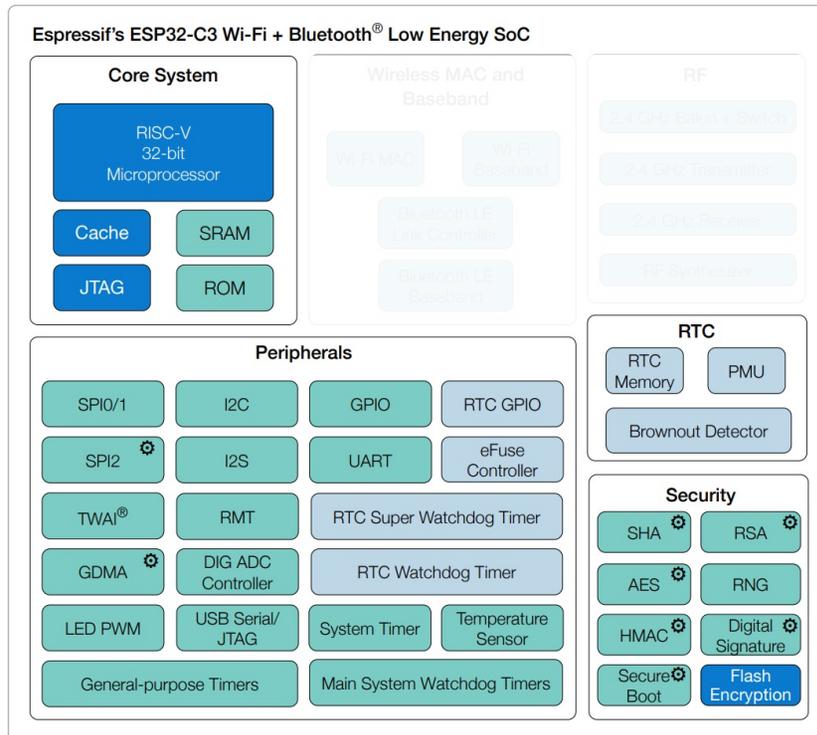
- All RF portions are turned off
- Device is basically just a microcontroller at that point

What is On/Off During Light Sleep
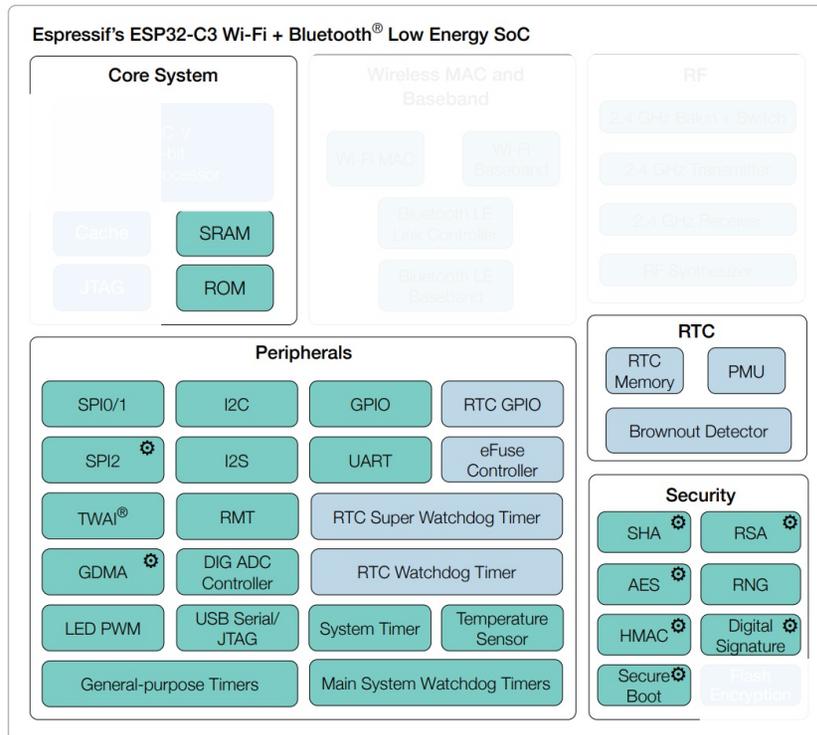
Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC

- Everything listed so far
- The processor is off
- Cache is off
- JTAG is off

Light Sleep Really messes with USB

Light Sleep

When entering light sleep, the APB and USB PHY clock are gated. Thus, the USB Serial/JTAG controller is no longer able to receive or respond to any USB transactions from the connected host (including periodic CDC Data IN transactions). As a result:

- when entering light sleep, the USB Serial/JTAG device is unresponsive to the host/PC's USB CDC driver. The host/PC may then report the USB Serial/JTAG device as disconnected or erroneous (even if the USB cable is still physically connected).
- when exiting light sleep, it is possible that the host/PC does not re-enumerate (i.e., reconnect) the USB Serial/JTAG device given that the USB PHY's D+ line remains pulled up state during light sleep. Users may need to physically disconnect and then reconnect the USB cable.

- In my experience, with light sleep the USB port does not work after waking again...your computer gets confused and you need to reset ESP32/restart it.

- Clock gating makes USB look "frozen" to host/computer

What is On/Off During Deep Sleep
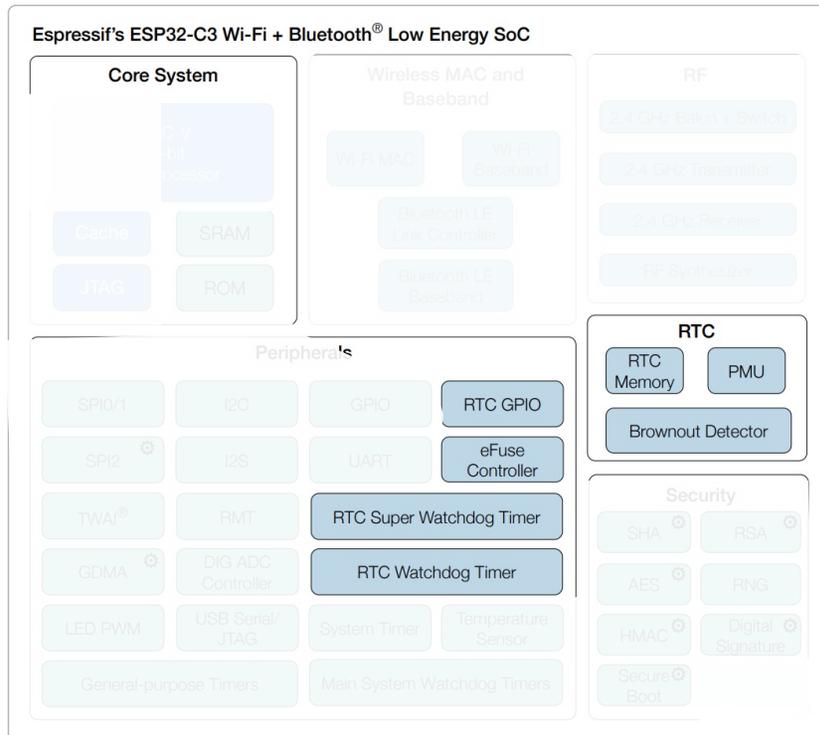
- Everything listed so far
- The processor is off
- Cache is off
- JTAG is off
- Almost All Peripherals
- Just a minimum set of functionalities that can carry you forward

The Mammalian Dive Reflex

- When any mammal holds its breath and has its sinuses exposed to low temperatures…

- Heart rate drops

- Vasculature in extremities contracts

- Blood flow prioritizes vital organs

- Just like an ESP32 C3 with power!

Deep Sleep Messes with USB Less

Deep Sleep

When entering deep sleep, both the USB Serial/JTAG controller and the USB PHY are powered off, leading to the USB PHY's D+ line no longer being pulled up. As a result:

- When entering deep sleep, the USB Serial/JTAG device appears disconnected from the host/PC (even if the USB cable is still physically connected).
- When exiting deep sleep, the USB Serial/JTAG device reconnects to the host/PC.

- USB doesn't work in deep sleep, but since it is powered off the computer doesn't freak out about it. Upon coming back from Deep sleep, USB controller on ESP32 C3 reboots so actually will come back up fine

ESP32 Power Consumption is Very Complicated

Not our ESP32
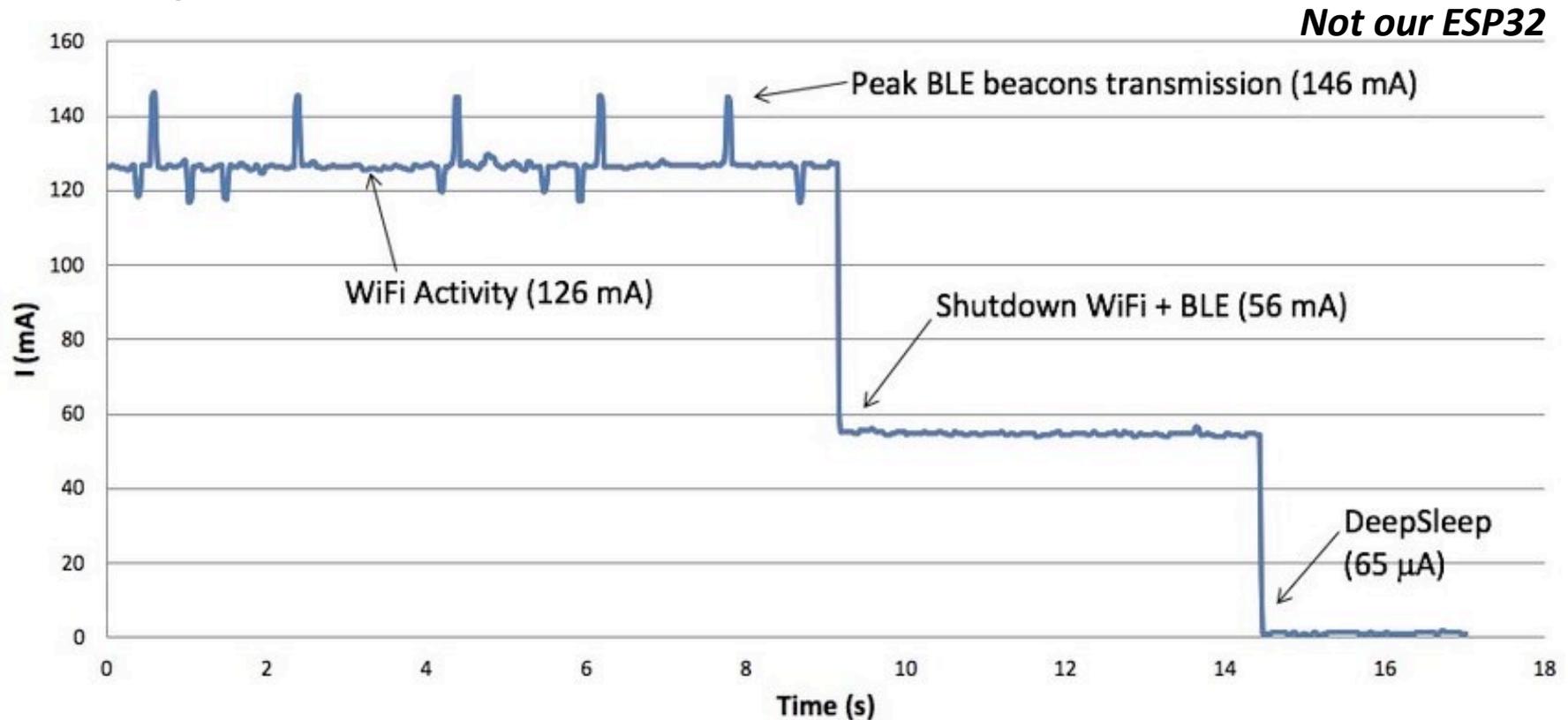
Take the JouleScope and measure power consumption in different modes

- 6.9000 Lesson: Never Trust Anybody
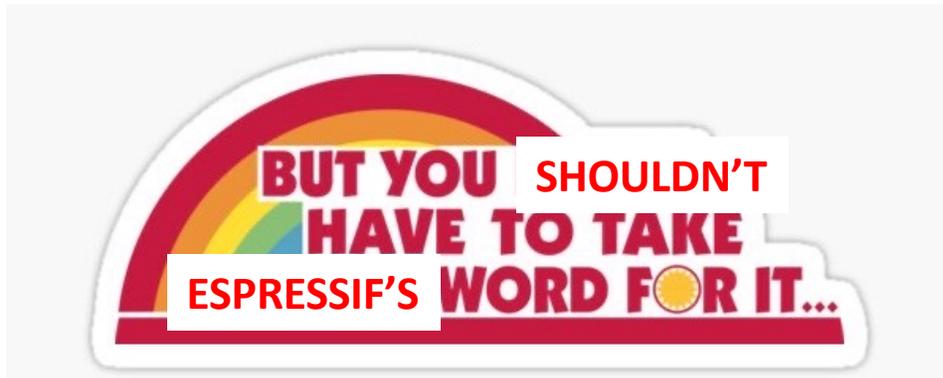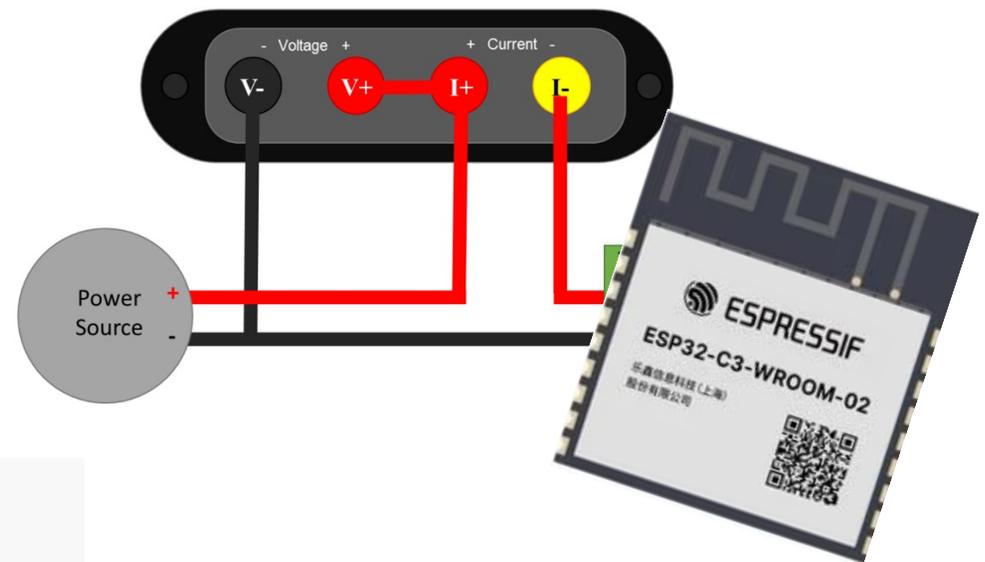
*Espressif is manufacturer of ESP32

ESP32 C3 Measurements (Code from Lab 01)

During regular running no wifi:


```
+18.3432  mA     +0.77182  std
                 +14.2044  min
                 +34.3144  max
                 +20.1099  p2p

+5.07666  V      +0.00230  std
                 +5.05914  min
                 +5.08998  max
                 +0.03084  p2p

+93.1219  mW     +3.90985  std
                 +72.1465  min
                 +173.624  max
                 +101.477  p2p
```

Mode	CPU Frequency (MHz)	Description	Typ	
			All Peripherals Clocks Disabled (mA)	All Peripherals Clocks Enabled (mA)[1]
Modem-sleep[2,3]	160	CPU is idle	16	21
		CPU is running	23	28
	80	CPU is idle	13	18
		CPU is running	17	22

Probably right???

During running with Wifi at some point during the request/contact cycle


```
+81.7395  mA     +1.41357  std
                 +73.8820  min
                 +106.359  max
                 +32.4775  p2p

+5.02088  V      +0.00422  std
                 +4.99804  min
                 +5.03247  max
                 +0.03442  p2p

+410.403  mW     +7.01329  std
                 +370.860  min
                 +531.903  max
                 +161.042  p2p
```

Work mode		Description	Peak (mA)
Active (RF working)	TX	802.11b, 1 Mbps, @20.5 dBm	345
		802.11g, 54 Mbps, @18 dBm	285
		802.11n, HT20, MCS7, @17.5 dBm	280
		802.11n, HT40, MCS7, @17 dBm	280
	RX	802.11b/g/n, HT20	82
		802.11n, HT40	84

Probably right???

*these measurements made with linear regulator so 5V not 3.3V

The power consumption is complicated

Connecting to WiFi

Requests to Numbers API

Worry about Power but also Energy
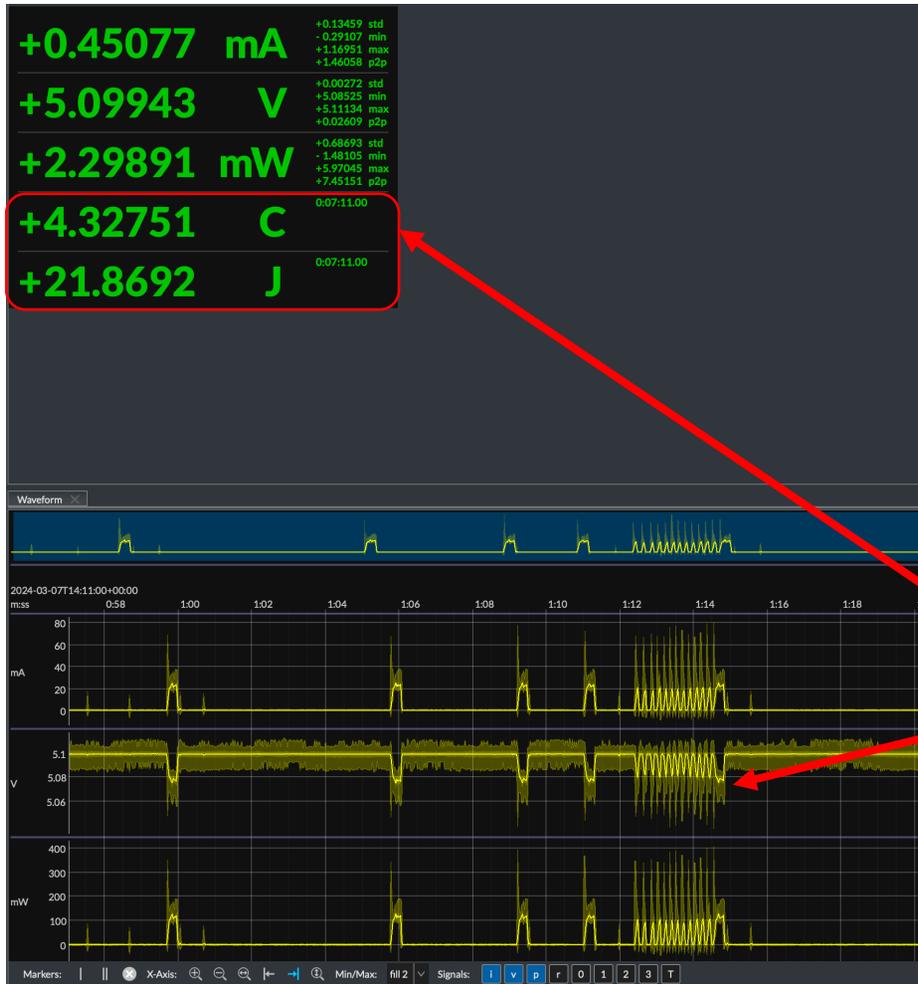
- The power pattern that appears during complicated tasks such as a WiFi request may not be best encompassed by an instantaneous power

- May need to worry more about energy per message

- Or something else related to integrating power over time!!!!

Must dig through docs and app notes!

https://www.espressif.com/sites/default/files/documentation/ESP32-C3%20Wireless%20Adventure.pdf

CHAPTER 12

https://docs.espressif.com/projects/esp-idf/en/stable/esp32c3/api-reference/system/power_management.html

Lots of Information Out there

https://blog.voltaicsystems.com/how-to-put-an-esp32-into-deep-sleep/

https://randomnerdtutorials.com/esp32-timer-wake-up-deep-sleep/

Last Few Years

- Prior to this year, we used the generic ESP32C3 M1 devkits
- These are cheap and easy to use
- They are not built with power in mind!!!
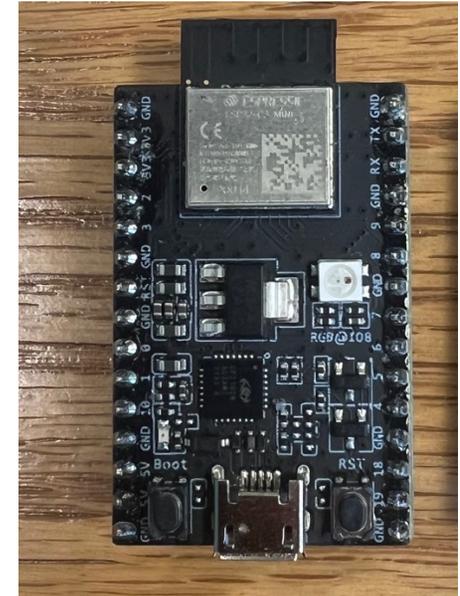
Experiments

- Wrote Some Test Files That Put the microcontroller into different modes

- This line lets the ESP wake itself from various sleep modes after a certain amount of time:

```
esp_sleep_enable_timer_wakeup(5000000); // how long to sleep in us.
```

- Then you can either go into light sleep:

```
esp_light_sleep_start();
```

- Or Deep Sleep:

```
esp_deep_sleep_start();
```

So for reference regular running with no wifi/BLE was like…

Mode	CPU Frequency (MHz)	Description	Typ	
			All Peripherals Clocks Disabled (mA)	All Peripherals Clocks Enabled (mA)[1]
Modem-sleep[2,3]	160	CPU is idle	16	21
		CPU is running	23	28
	80	CPU is idle	13	18
		CPU is running	17	22

Multimeter ✕

Device default ⌄   JS220-001003

[Hold] [Accrue]   500 ms

+17.4238 mA
+3.16899 std
+11.9755 min
+28.4362 max
+16.4607 p2p

+3.26740 V
+0.00328 std
+3.25271 min
+3.27684 max
+0.02413 p2p

+56.9405 mW
+10.4115 std
+39.0548 min
+92.8768 max
+53.8219 p2p

Light Sleep:

`esp_light_sleep_start();`

Seems to go to sleep on a timer…

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

+0.80021 mA
+1.94107 std
- 9.19773 min
+20.5256 max
+29.7234 p2p

+3.29538 V
+0.00296 std
+3.28734 min
+3.31964 max
+0.03230 p2p

+2.63824 mW
+6.40195 std
- 30.3345 min
+67.7252 max
+98.0597 p2p

Definitely Lower Power but is it good enough?

Why is it 6 times as large?

Maybe all pins aren't high-impedance?

Not sure…

And then there's Deep Sleep

`esp_deep_sleep_start();`

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

+0.48471 mA	+2.01551 std - 9.09876 min +28.2694 max +37.3681 p2p
+3.29508 V	+0.00431 std +3.28454 min +3.32107 max +0.03653 p2p
+1.59945 mW	+6.65719 std - 30.0296 min +93.2978 max +123.327 p2p

Definitely Lower Power but is it good enough?

Why is it 100 ish times larger?

Should we worry?

Still…Something Not Right

- The numbers were qualitatively correct when looked at together, but quantitatively in disagreement with the datasheet!

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

+0.79970 mA	+2.31550 std
	- 8.56741 min
	+24.0256 max
	+32.5930 p2p
+3.29425 V	+0.00410 std
	+3.28410 min
	+3.31909 max
	+0.03498 p2p
+2.63796 mW	+7.64561 std
	- 28.2709 min
	+79.2723 max
	+107.543 p2p

+0.48471 mA	+2.01551 std
	- 9.09876 min
	+28.2694 max
	+37.3681 p2p
+3.29508 V	+0.00431 std
	+3.28454 min
	+3.32107 max
	+0.03653 p2p
+1.59945 mW	+6.65719 std
	- 30.0296 min
	+93.2978 max
	+123.327 p2p
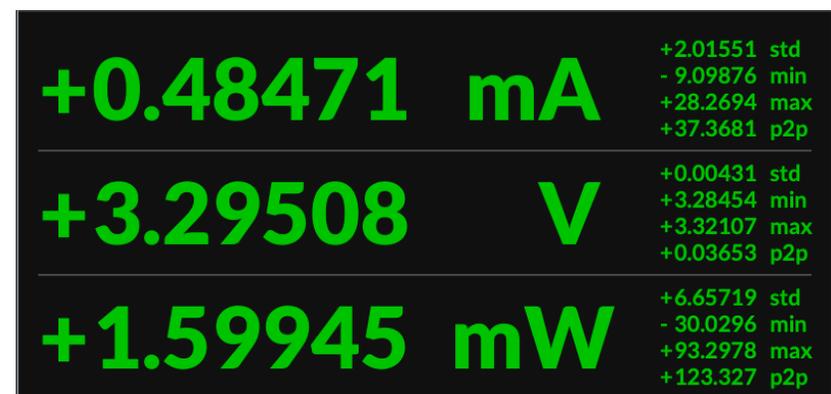
Remember Board

The actual ESP32 C3 is a chip inside this metal thing

LED that Comes on Sometimes

CP2102 USB-UART Chip

Always-On LED
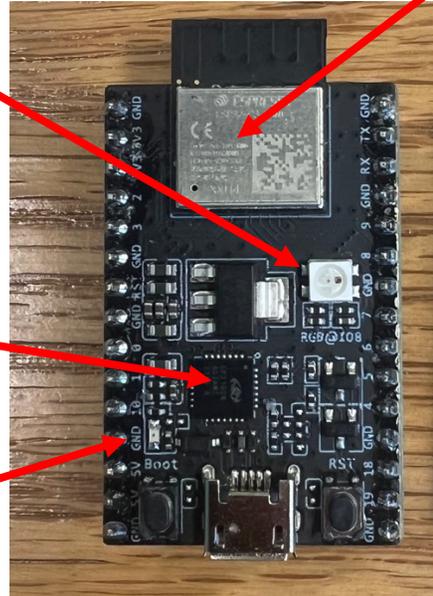
These numbers may have been getting impacted by these other rando parts....what are/were they?

ESP32 C3 Devkit M1 Schematic
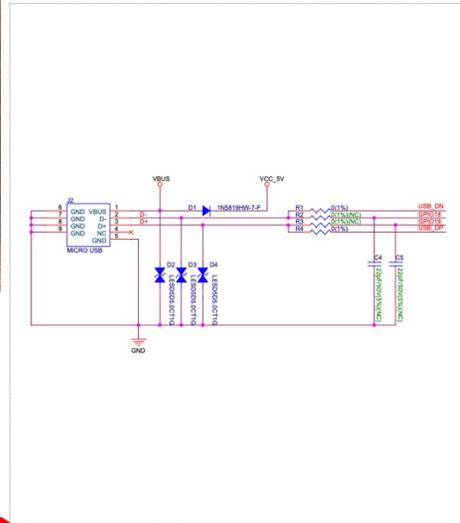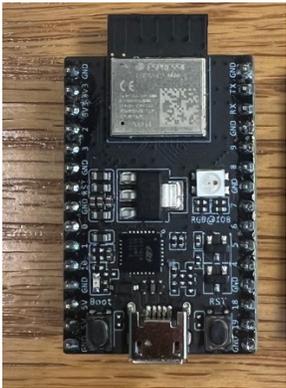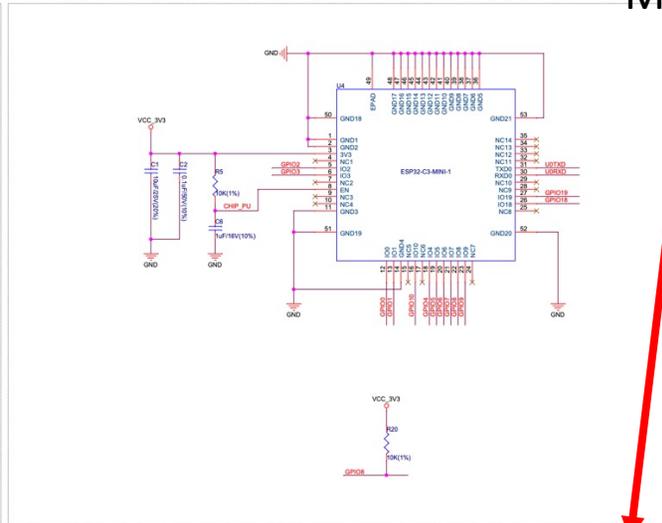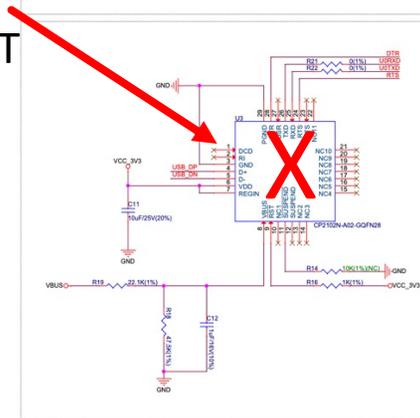
Regulator Maybe?
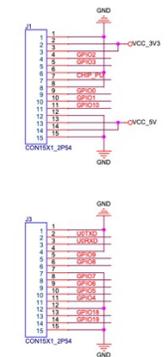
Always-On LED only works with 5V

CP2102 USB-UART Chip
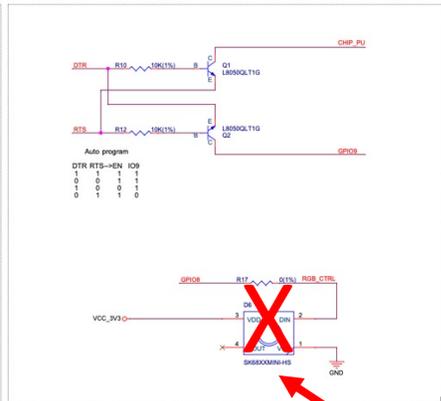
LED that Comes on Sometimes and blinds you
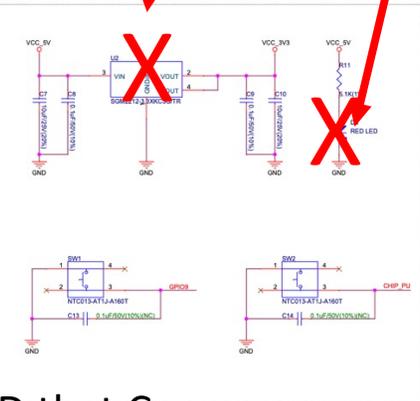
CP2102 USB-UART Interface

So maybe about 80 uA from that chip just sitting there and doing nothing useful
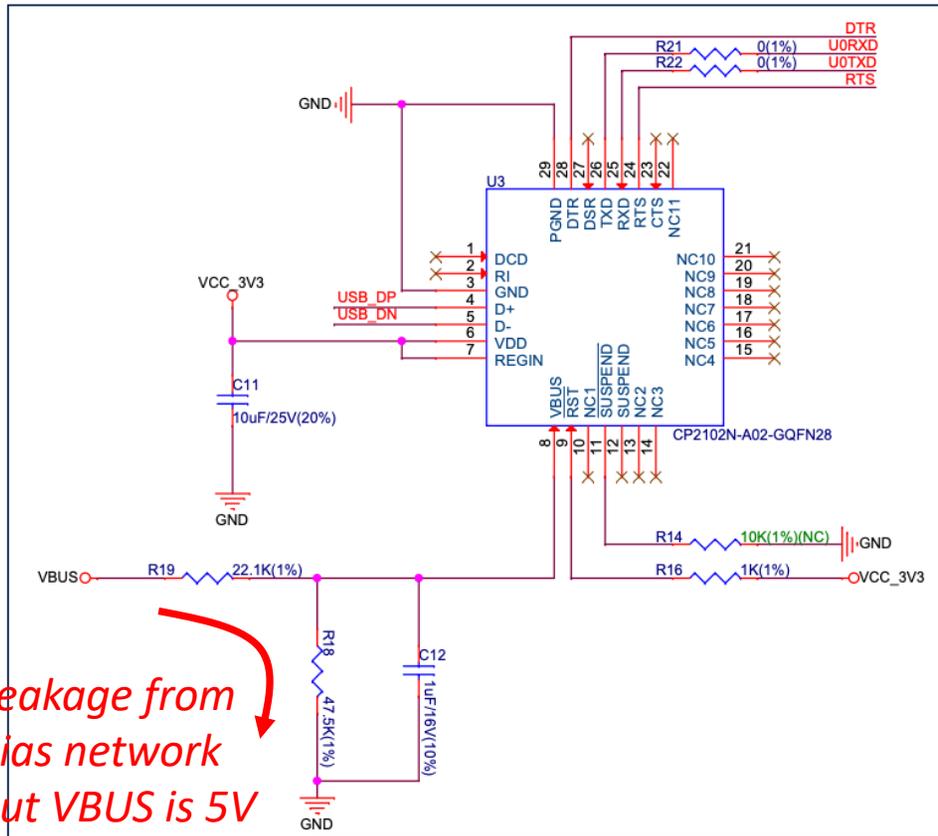
Leakage from bias network but VBUS is 5V and we're not giving board that so ignore

CP2102						
Supply Current—Normal[3]	I_{REGIN}	Normal Operation; V_{REG} Enabled	—	20	26	mA
Supply Current—Suspended[3]		Bus Powered; V_{REG} Enabled	—	80	100	µA

The Bright LED: SK68XXMINI
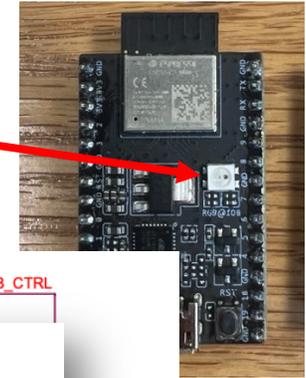
shenzhen Rose Lighting Technology Co,. Ltd
Rose Lighting
rose-lighting.com   https://www.rose-lighting.com/   www.ulstrip.com
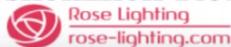
3. Description:

● **Top SMD internal integrated high quality external control line serial cascade constant current IC;**

● **control circuit and the RGB chip in SMD 3535 components, to form a complete control of pixel, color mixing uniformity and consistency;**

● **built-in data shaping circuit, a pixel signal is received after wave shaping and output waveform distortion will not guarantee a line;**

● **The built-in power on reset and reset circuit, the power does not work;**

● **gray level adjusting circuit (256 level gray scale adjustable);**

● **red drive special treatment, color balance;**

● **line data transmission;**

● **plastic forward strengthening technology, the transmission distance between two points over 10M;**

● **Using a typical data transmission frequency of 800 Kbps, when the refresh rate of 30 frames per sec**

GPIO8   R17 —/\/\/— 0(1%)   RGB_CTRL

The chip supply voltage	VDD	---	5.2	---	V	---
The signal input flip threshold	VIH	0.7*VDD	---	---	V	VDD=5.0V
	VIL	---	---	0.3*VDD	V	
The frequency of PWM	FPWM	---	1.2	---	KHZ	---
Static power consumption	IDD	---	1	---	mA	---

Is there anything else in the Mini Module?

Removed top off of Mini Module:

This is the reference design of the module.

The values of C1 and C2 vary with the selection of the crystal.

The value of R1 varies with the actual PCB board.

The values of C8, L2 and C9 vary with the actual PCB board.

NC: No component.
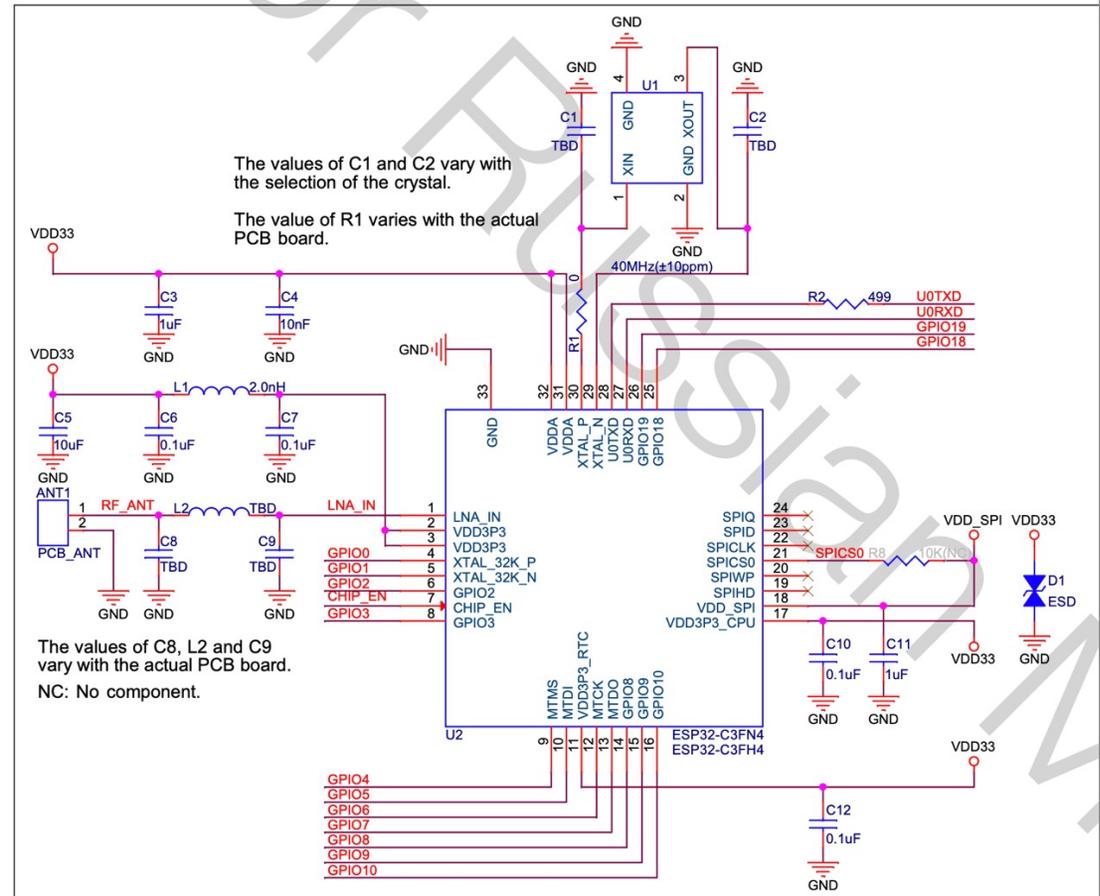
Figure 5: ESP32-C3-MINI-1 Schematics

https://www.espressif.com/sites/default/files/russianDocumentation/esp32-c3-mini-1_datasheet_en.pdf
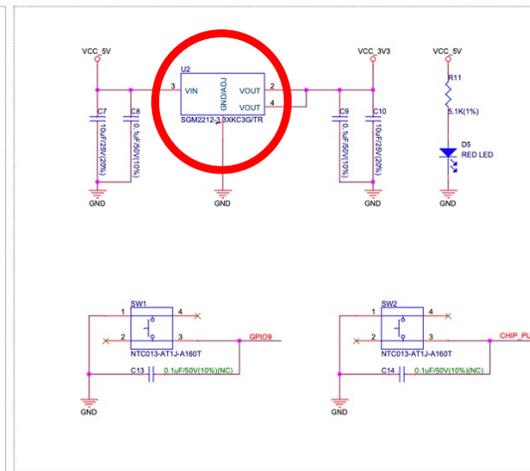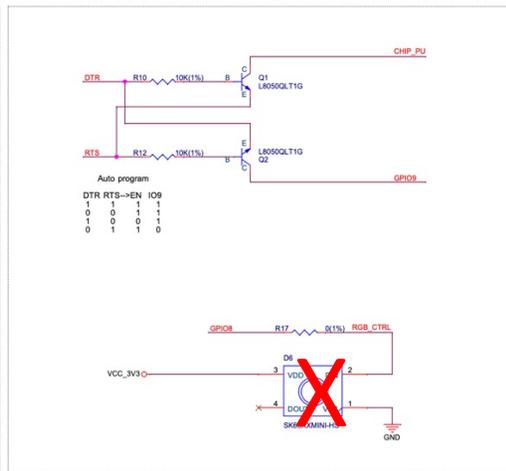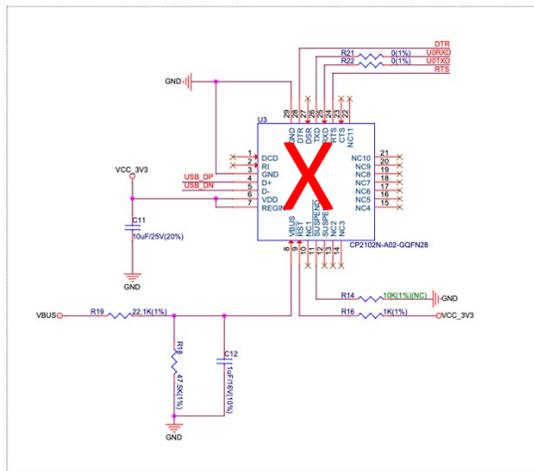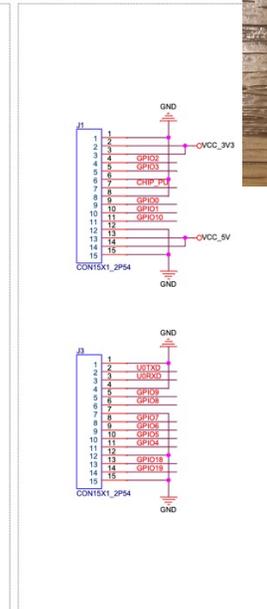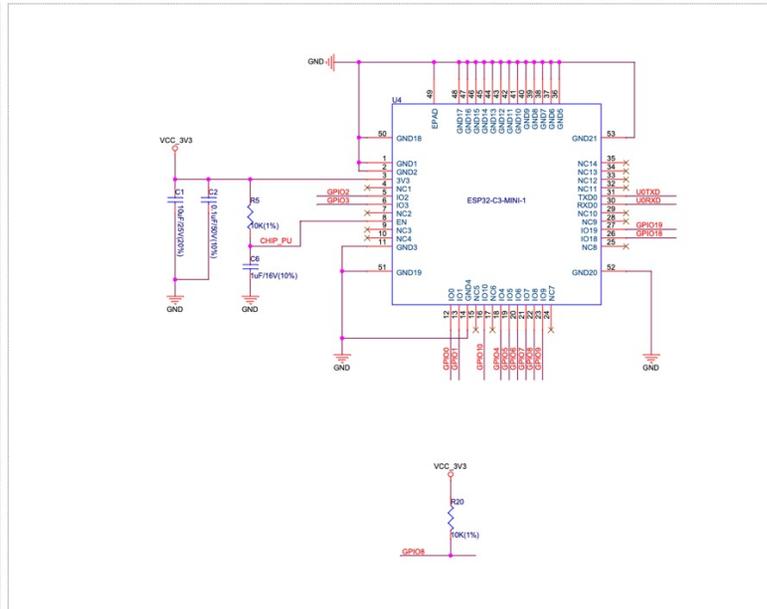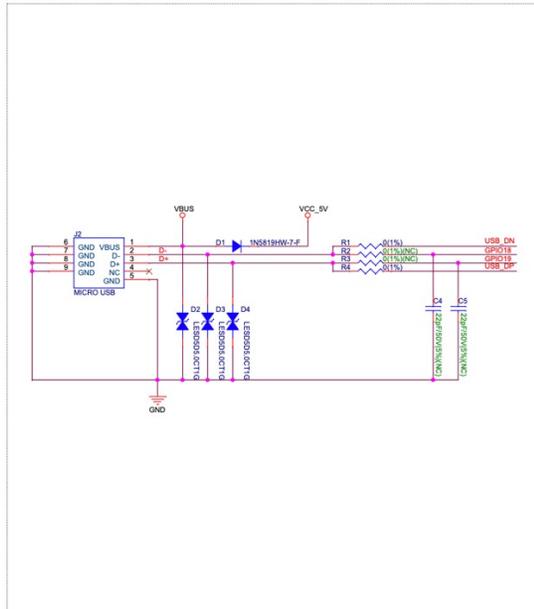
Is there anything else in the Mini Module?

Nothing…

Ugh….

This is the reference design of the module.

Figure 5: ESP32-C3-MINI-1 Schematics

Back to the Whole Board

We saw a Linear Regulator Last Week

- Constant Current Device (KCL maintained)
- Can only regulate down in voltage

v can vary

Always @$desired\ Voltage$

Reg

i

i

VUSB

ESP32

i

We saw a Linear Regulator Last Week

- Constant Current Device (KCL maintained)
- Can only regulate down in voltage

v can vary

Always @$desired\ Voltage$

Reg

VUSB

i_{TOT}

i_{ESP32}

i_{Reg}

ESP32

i_{ESP32}
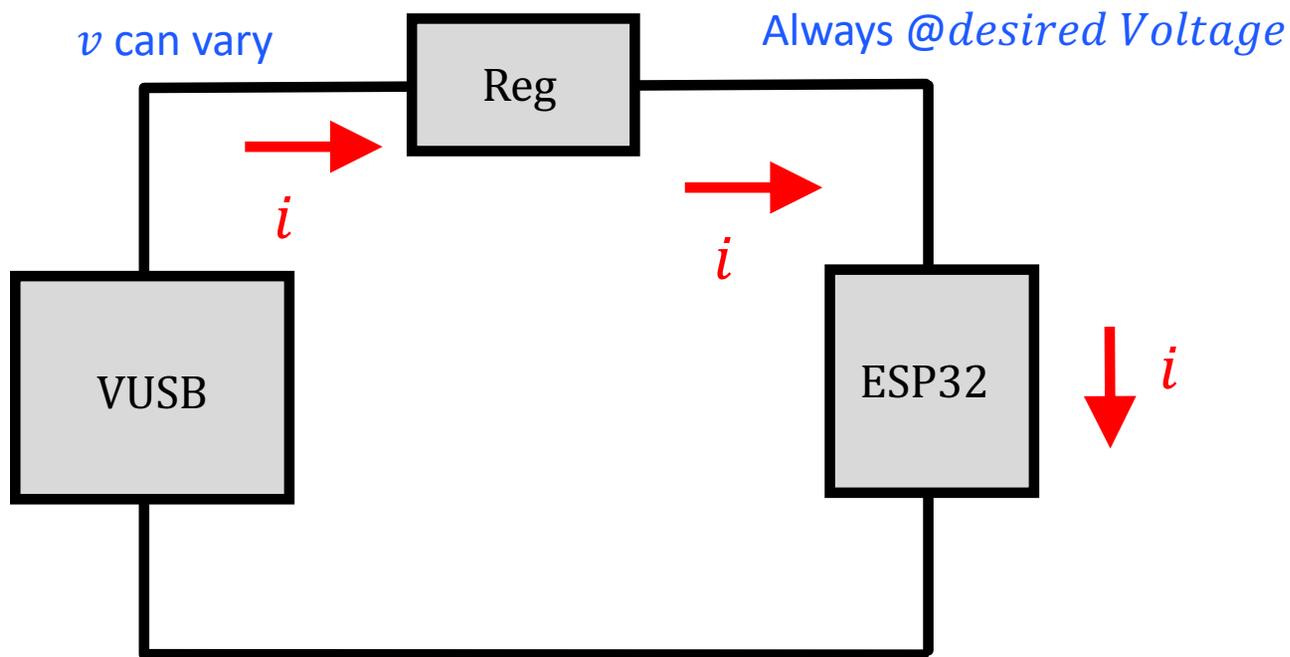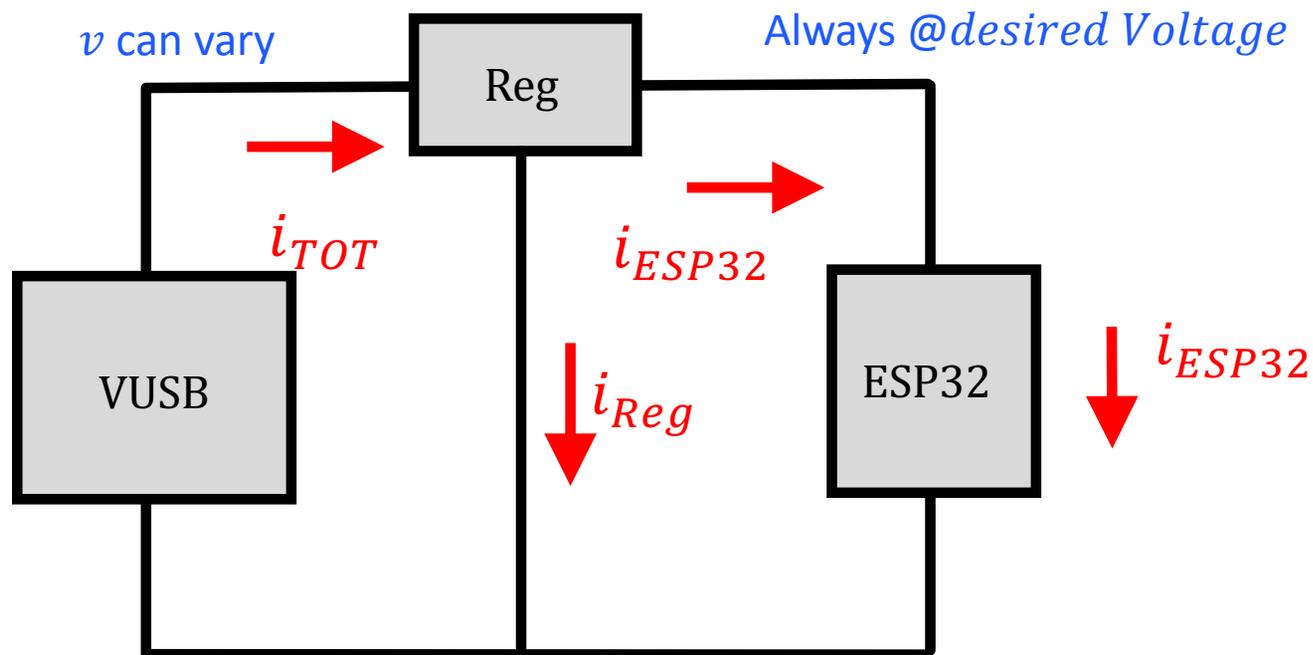
We saw a Linear Regulator Last Week

- No idea how the regulator will work when not actively powered but it

Set @ 3.3V using external supply

Reg

$i_{Leak}?$

$i_{Leak}?$

ESP32

i_{ESP32}
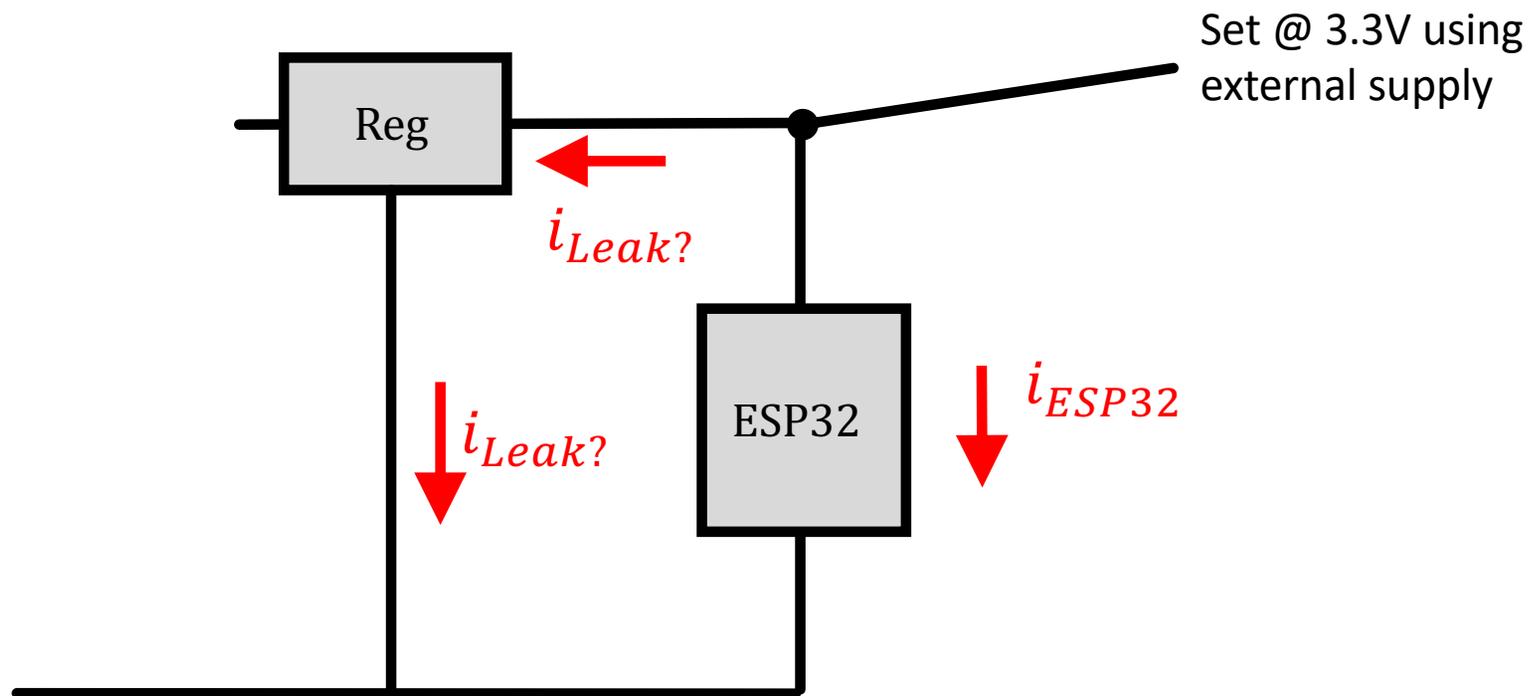
Any particular spots?
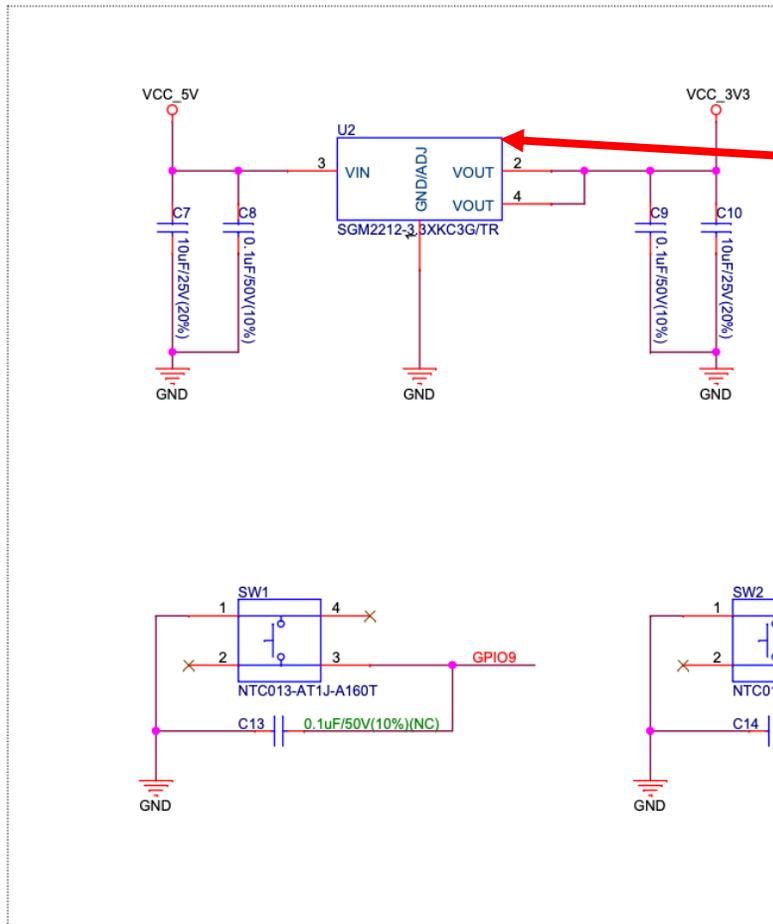
Even though the Linear regulator is driven by 5V and there is no 5V in our system as tested, the regulator might be leaking some current from its VOUT terminal to GND

SGM2212

When powered with 5V but no draw on the 3.3V about 80 uA consumed, but no indication how this regulator works when not powered by 5V, but VOUT connected to 3.3V (kinda out of spec)

SGM2212
800mA, Low Noise, Low Quiescent Current, High PSRR, Low Dropout Linear Regulator

GENERAL DESCRIPTION

The SGM2212 is a low noise, low quiescent current, high PSRR, fast transient response and low dropout voltage linear regulator which is designed using CMOS technology. It provides 800mA output current capability. The operating input voltage range is from 2.7V to 20V. The fixed output voltages are 1.8V, 2.5V, 2.8V, 3.3V, 5V and adjustable output voltage range is from 1.25V to 15V.

Other features include short-circuit current limit and thermal shutdown protection.

The SGM2212 is available in Green TO-252-2, TO-263-3, SOT-223-3 and TDFN-3×3-8L packages. It operates over an operating temperature range of -40℃ to +125℃.

FEATURES

- Input Voltage Range: 2.7V to 20V
- Output Current: 800mA
- Fixed Output Voltages:1.8V, 2.5V, 2.8V, 3.3V, 5V
- Adjustable Output Voltage Range: 1.25V to 15V
- Output Voltage Accuracy: ±1% at +25℃
- Line Regulation: 0.012% (MAX)
- Load Regulation: 0.4% (MAX)
- Stable with Small Case Size Ceramic Capacitors
- Output Current Limit
- Thermal Shutdown Protection
- -40℃ to +125℃ Operating Temperature Ra...
- Available in Green TO-252-2, TO-263-3, SOT-223-3 and TDFN-3×3-8L Packages

APPLICATIONS

Portable Electronic Device

APPLICATION INFORMATION

The SGM2212 is a low noise, fast transient response high performance LDO, it consumes only 80µA (TYP) quiescent current and provides 800mA output current. The SGM2212 provides the protection function for output overload, output short-circuit condition and overheating.

The SGM2212 is suitable for application which has noise sensitive circuit such as battery-powered equipment and smartphones.

https://www.sg-micro.com/uploads/soft/20220506/1651829970.pdf

Conclusion: Those Components were Leeches

- They can serve a purpose on the development board for sure, but in this context they are leeches sucking away our precious natural bodily fluids

- Needed to purge them

Bias Network

SGM2212

CP2102

SK68XXMINI

ESP32C3
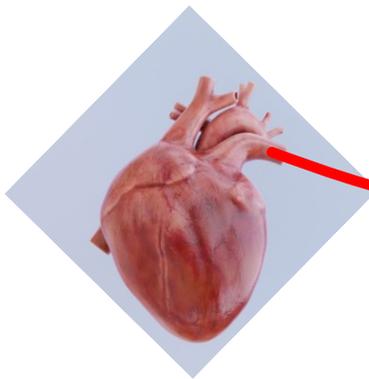
18650 Lithium Battery

Purged them in 2026

Regular Dev Board:

Improved Board this year
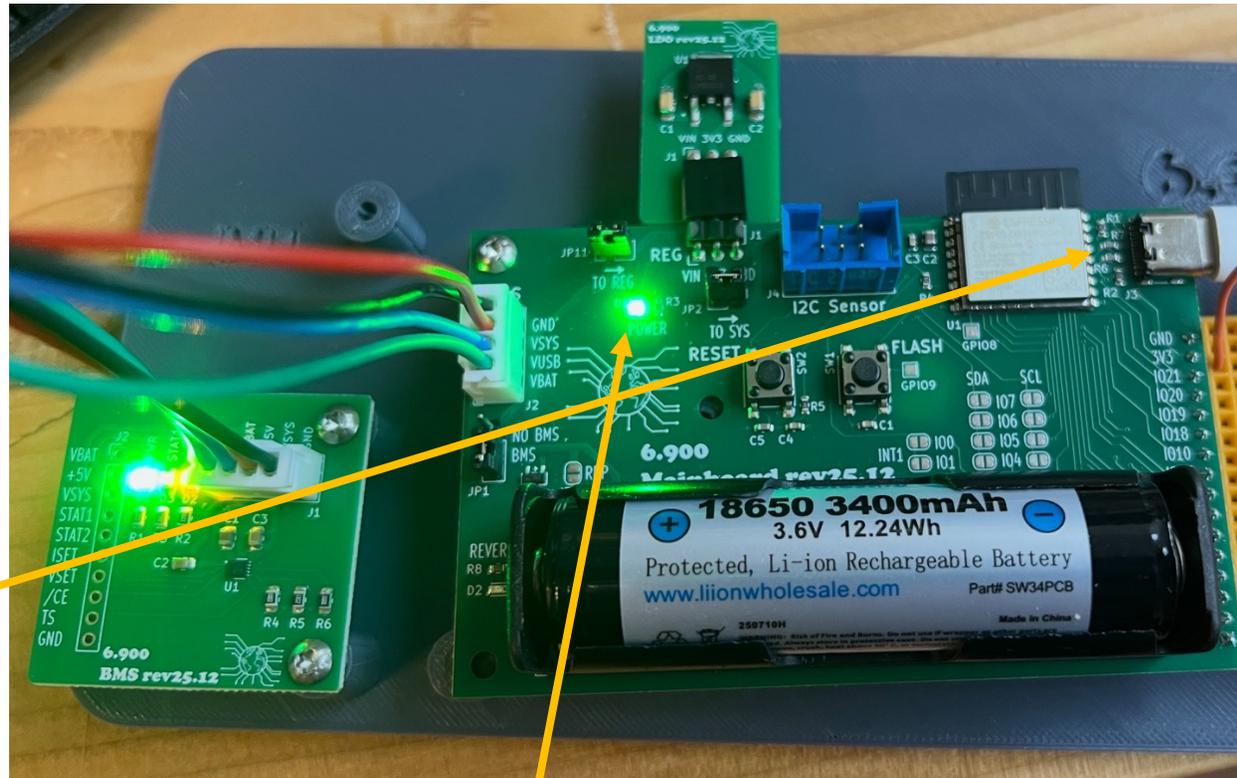
Removed CP2102

Still bright LED, but on the battery-side of regulator so we can easily remove it from measurements

Remove them

Regular Running:


```
Multimeter  X
Device  default  v    JS220-001003
[Hold] [Accrue]  500 ms

+17.4238  mA      +3.16899 std
                  +11.9755 min
                  +28.4362 max
                  +16.4607 p2p

+3.26740   V      +0.00328 std
                  +3.25271 min
                  +3.27684 max
                  +0.02413 p2p

+56.9405  mW      +10.4115 std
                  +39.0548 min
                  +92.8768 max
                  +53.8219 p2p
```

WiFi Running:

```
+84.4405  mA      +18.7619 std
                  +48.7327 min
                  +428.993 max
                  +380.260 p2p

+3.29937   V      +0.00537 std
                  +3.21240 min
                  +3.30766 max
                  +0.09525 p2p

+0.27850   W      +0.06008 std
                  +0.16102 min
                  +1.38511 max
                  +1.22408 p2p
```

esp_light_sleep_start();

```
+379.998  µA      +16.2147 std
                  +341.864 min
                  +465.412 max
                  +123.547 p2p

+3.30803   V      +0.00059 std
                  +3.30484 min
                  +3.31115 max
                  +0.00631 p2p

+1.25704  mW      +0.05365 std
                  +1.13067 min
                  +1.54008 max
                  +0.40940 p2p
```

esp_deep_sleep_start();

```
+6.85262  µA      +0.10983 std
                  +6.40330 min
                  +7.44173 max
                  +1.03842 p2p

+3.30806   V      +0.00059 std
                  +3.30519 min
                  +3.31109 max
                  +0.00590 p2p

+22.6652  µW      +0.36355 std
                  +21.1745 min
                  +24.6167 max
                  +3.44216 p2p
```

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

Current Plot during WiFi Transaction
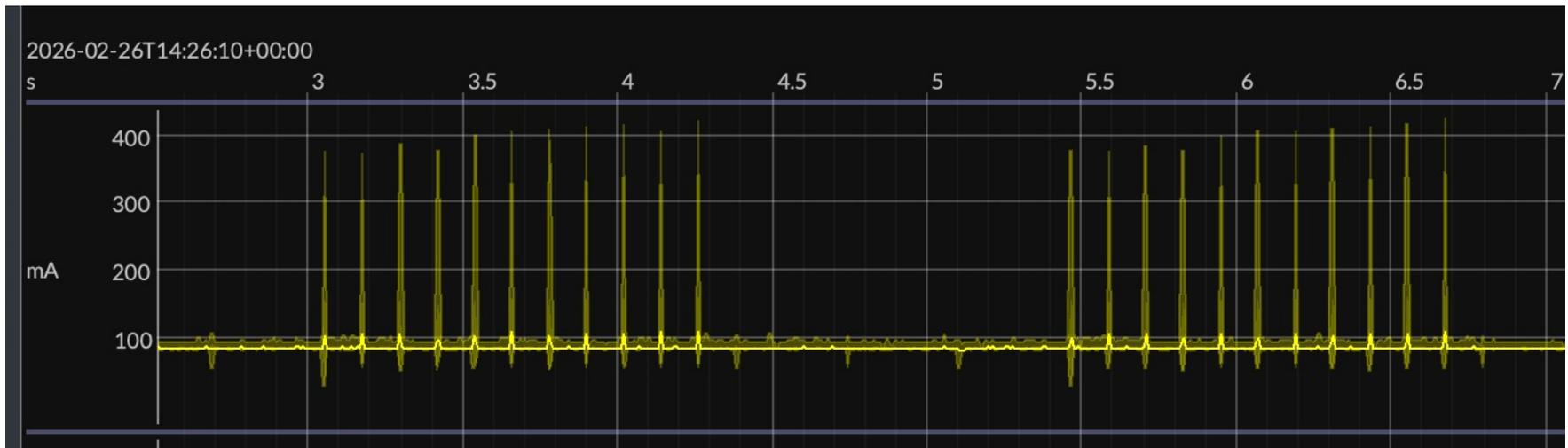
Joulescope samples @ 1MSPS

Spike up to 320mA during transmit
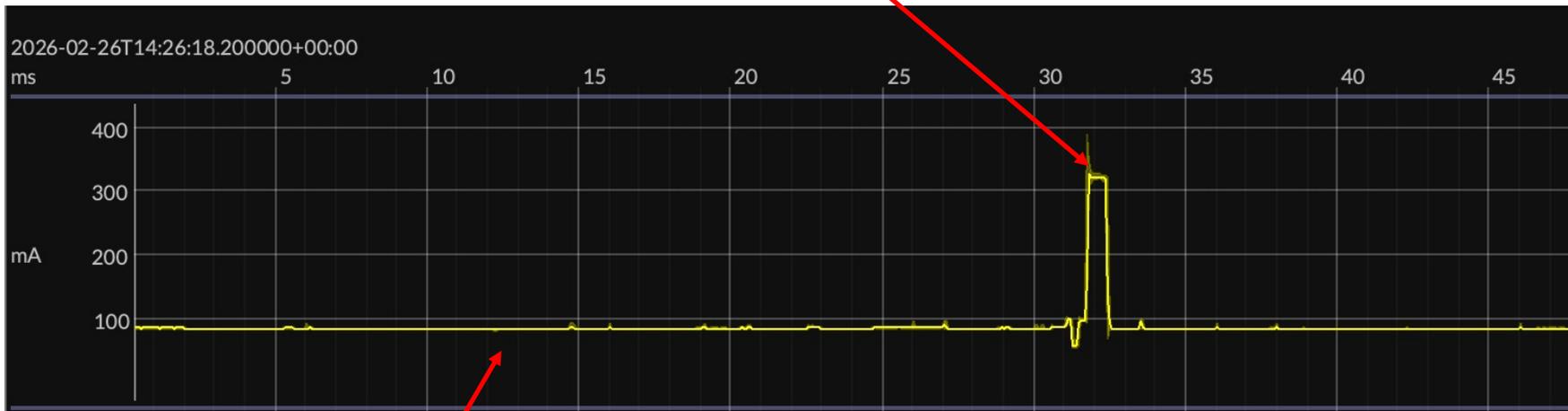
82 mA normally on

Interestingly in Light Sleep

With ESP32C3 plugged into USB   ***Without*** *ESP32C3 plugged into USB*
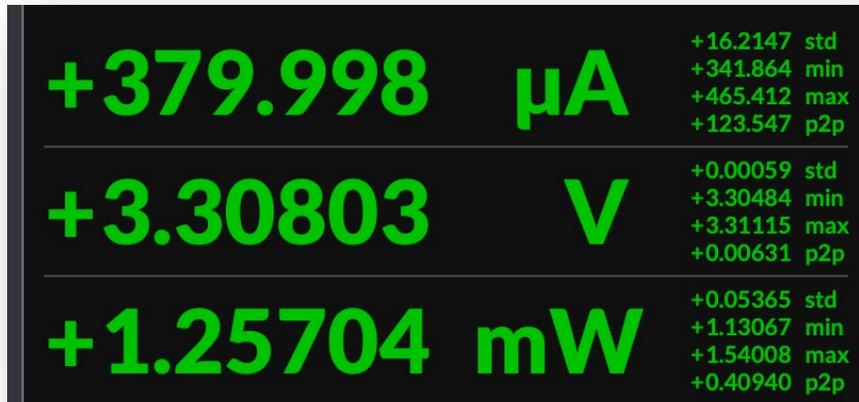
Interestingly in Light Sleep

Without *ESP32C3 plugged into USB*

+201.696 µA	+7.79759 std +163.035 min +237.861 max +74.8252 p2p
+3.30773 V	+0.00059 std +3.30487 min +3.31048 max +0.00561 p2p
+667.156 µW	+25.8096 std +539.310 min +787.027 max +247.716 p2p

Still quite a disconnect

Mode	Description	Typ (μA)
Light-sleep	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance	130
Deep-sleep	RTC timer + RTC memory	5
Power off	CHIP_EN is set to low level, the chip is powered off	1

Probably coming from a lot of optional things...
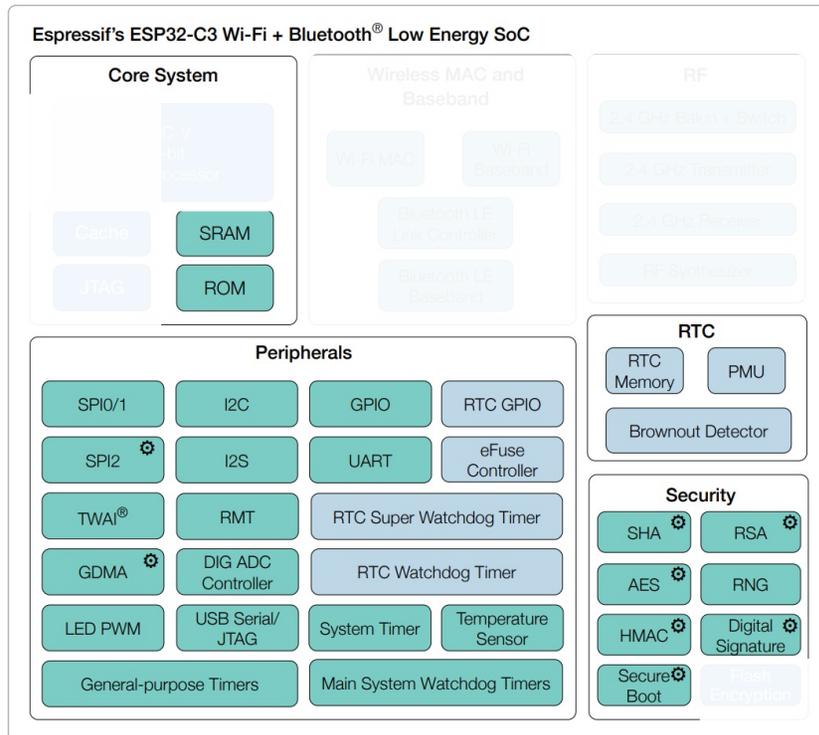
Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC

- SPI, TWAI, GPIO things.

optional in Light-sleep

Probably need to do some more digging

The Espressif Stable of Devices

ESP32 · ESP32-S2 · ESP32-S3 · ESP32-C2 · ESP32-C3 · ESP32-C5 · ESP32-C6

Espressif SoC Product Portfolio
Comprehensive Connectivity, AI and HMI SoCs

ESPRESSIF — Product Selector

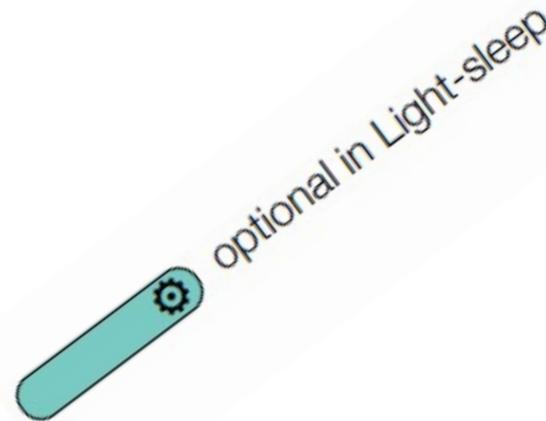| Feature		P-Series ESP32-P4	S-Series ESP32-S3	S-Series ESP32-S2	C-Series ESP32-C5	C-Series ESP32-C6	C-Series ESP32-C61	C-Series ESP32-C3	C-Series ESP32-C2	H-Series ESP32-H4	H-Series ESP32-H2	32-Series ESP32
Wi-Fi	Wi-Fi Generation		11b/g/n	11b/g/n	11a/b/g/n/ac/ax	11b/g/n/ax	11b/g/n/ax	11b/g/n	11b/g/n			11b/g/n
	Data Rate		150 Mbps	150 Mbps	150 Mbps	150 Mbps	150 Mbps	150 Mbps	72.2 Mbps			150 Mbps
	Dual Band				✓							
	Wi-Fi 6				✓	✓	✓					
	SoftAP		✓	✓	✓	✓	✓	✓	✓			✓
	Sniffing		✓	✓	✓	✓	✓	✓	✓			✓
	WPA Support		WPA3	WPA3	WPA3	WPA3	WPA3	WPA3	WPA3			
	Wi-Fi CSI		✓	✓	✓	✓	✓	✓	✓			✓
Bluetooth	Bluetooth Low Energy		5.0		5.3	5.3	5.0	5.0	5.0	5.4	5.0	BT Classic/ 4.2
	LE-Audio									✓		
	Direction Finding									✓		
	Long Range		✓		✓	✓	✓	✓	✓	✓	✓	
	2Mbps Data Rate		✓		✓	✓	✓	✓	✓	✓	✓	
	Extended Advertising		✓		✓	✓	✓	✓	✓	✓	✓	
	BLE-Mesh		✓		✓	✓	✓	✓	✓	✓	✓	
	Thread				✓	✓				✓	✓	
	Zigbee					✓				✓	✓	
	Matter		Wi-Fi		Wi-Fi, Thread	Wi-Fi, Thread	Wi-Fi	Wi-Fi	Wi-Fi	Thread	Thread	Wi-Fi
Type	Ethernet	✓										
	SoC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Module		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Radio	Frequency		2.4 GHz	2.4 GHz	2.4 GHz 5 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz
	Tx-Power		0 to 20 dBm	0 to 20 dBm	0 to 20 dBm	0 to 20 dBm	0 to 20 dBm	0 to 20 dBm	0 to 20 dBm	-15 to 9 dBm	-15 to 20 dBm	0 to 20 dBm
	Rx-Sensitivity (dBm)		-98.4 (Wi-Fi-1M) -104 (BLE-125k)	-97 (Wi-Fi-1M)	-101(Wi-Fi-1M) -99(Wi-FiSG-6M) -107(BLE-125k) -104(802.15.4)	-99.2 (Wi-Fi-1M) -106 (BLE-125k) -104 (802.15.4)	-99.5(Wi-Fi-1M) -105(BLE-125k)	-98.4 (Wi-Fi-1M) -105 (BLE-125k)	-99 (Wi-Fi-1M) -106 (BLE-125k)	-(BLE-125k) -(802.15.4)	-106.5 (BLE-125k) -102.5 (802.15.4)	-98 (Wi-Fi-1M) -94 (BLE-1M)
CPU & Memory	CPU	Dual Core RISC-V	Dual Core Xtensa	Single Core Xtensa	Single Core RISC-V	Single Core RISC-V	Single Core RISC-V	Single Core RISC-V	Single Core RISC-V	Dual Core RISC-V	Single Core RISC-V	Single/Dual Core Xtensa
	Clock	400MHz	240MHz	240MHz	240MHz	160MHz	160MHz	160MHz	120MHz	96MHz	96MHz	240MHz
	FPU Extension	✓	✓							✓		
	AI Extensions	✓	✓									
	LP CPU	40MHz RISC-V	17.5MHz RISC-V ULP Core	8MHz RISCV + ULP Core	40MHz RISC-V	20MHz RISC-V						ULP Core
	SRAM	768KB (HP) 16KB (LP)	512KB 16KB (LP)	320KB (HP) 16KB (LP)	384KB (HP) 16KB (LP)	512KB (HP) 16KB (LP)	320KB	400KB 8KB (RTC)	272KB	320KB	320KB 4KB (RTC)	520KB 16KB (RTC)
	External Flash	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PSRAM Support	✓	✓	✓	✓	✓		✓		✓		✓
	Secure Boot	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Flash Encryption	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Security	Crypto Accelerators	RSA, AES, ECC, HMAC	RSA, AES, HMAC	RSA, AES, HMAC	AES, ECC, HMAC, RSA	RSA, AES, ECC, HMAC	ECC,SHA	RSA, AES, HMAC	ECC,SHA	AES, ECC, HMAC	RSA, AES, ECC, HMAC	RSA, AES, SHA
	Digital Signature	ECDSA, RSA	RSA	RSA	ECDSA, RSA	RSA	ECDSA	RSA		ECDSA	ECDSA	
	APM/TEE	✓			✓					✓	✓	
	Key Manager				✓							
HMI	MIPI-CSI	2-lane										
	H.264 Encoder	1080p@30fps										
	MIPI-DSI	2-lane										
	RGB Display	✓	✓	✓								✓
	Camera Interface	✓	✓	✓								✓
	Touch Inputs	14	14	14						14		10
Peripherals	GPIO Pins	55	45	43	29	30 or 22	22 or 18	22 or 16	14	35	19	34
	SDIO Host	1	1									1
	SDIO Slave				1	1	1					1
	I3C	1										
	SPI	4x SPI 1x LP SPI	4	4	2x SPI 1x GP SPI	2x SPI 1x GP SPI	3	3	3	4	3	4
	USB OTG	1x HS 1x FS	1x FS	1x FS						1		
	USB Serial	1	1	1	1	1	1	1		1	1	1
	CAN-FD/TWAI	3x TWAI	1x TWAI	1x TWAIx1	2x CAN-FD	2x TWAI		1x TWAI		1x CAN-FD	1x TWAI	1x TWAI
	UART	5x UART 1x LP UART	3	2	2	2x UART 1x LP UART	3	2	2	2	2	3
	I2C	2x I2C 2x LP I2C	2	2	1x I2C 1x LP I2C	1x I2C 1x LP I2C	1	1	1	2	2	2
	I2S	3x I2S 1x LP I2S	2	1	1	1	1	1		1	1	2
	PWM	1x LED 8ch 2x MCPWM 6ch	1x LED 8ch 2x MCPWM	1x LED 8ch	1x LED 6ch 1x MCPWM 6ch	1x LED 6ch 1x MCPWM	1x LED 6ch	1x LED 6ch	1x LED 6ch	1x LED 8ch 2x MCPWM	1x LED 6ch 1x MCPWM	1x LED 16ch 2x MCPWM
	ADC	2x 12 bit	2x 12 bit	2x 13 bit 20ch	1x 12 bit 6ch	1x 12 bit 7ch	1x 12 bit	2x 12 bit 6ch	1x 12 bit 5ch	1x 12 bit	1x 12 bit 5ch	1x 12 bit 18ch
	DAC			2x 8 bit								2x 8 bit
	Timers, WDT	4x HP GPT, 1x ST, 3x HP WDT,1x LPST 1x LPGPT, 1x LP WDT,1x ASWDT	4x GPT, 1x ST, 3x WDT	1x GPT, 1x ST, 3x WDT, 1x SWDT, 2x XTAL32k WDT	1x ST, 2x GPT, 1x RTCT, 3x DWDT, 1x AWDT	1x ST, 2x GPT, 3x DWDT, 1x AWDT	2x GPT, 1x ST, 2x SWDT, 3x WDT	2x GPT, 3x DWDT, 1x AWDT	1x GPT, 2x WDT, 1x ST	2x GPT, 1x ST, 3x WDT	2x GPT, 1x ST, 3x WDT	3x WDT, 4x GPT
	Temp Sensor	1	1	1	1	1	1	1	1	1	1	1
Operating Temperature		-45 to 85	-40 to 105	-40 to 105	-40 to 85	-40 to 105	-40 to 105	-40 to 105	-40 to 105	-40 to 105	-40 to 105	-40 to 125
Supply Voltage		3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	3.0 to 3.6	1.7 to 5.0	1.8 to 3.6	2.3/3.0 to 3.6
Package Type		QFN104	QFN56 LGA56	QFN56	QFN48	QFN40 QFN32	QFN32 QFN28	QFN32	QFN24	QFN52	QFN32	QFN48 LGA48

ESP32C6

- C6 has a **regular core** and a **low-power core** that uses roughly ~1mA while capable of doing I2C, UART, etc...

- The LP core can wake up the HP core when needed.

Espressif's ESP32-C6 Wi-Fi + Bluetooth® Low Energy + 802.15.4 SoC

CPU System
- HP RISC-V 32-bit Microprocessor
- LP RISC-V 32-bit Microprocessor
- Cache
- SRAM
- LP Memory
- JTAG
- ROM

Wireless MAC and Baseband
- Wi-Fi Baseband
- Wi-Fi MAC
- Bluetooth LE Baseband
- Bluetooth LE Link Controller
- 802.15.4 Baseband
- 802.15.4 MAC

RF
- 2.4 GHz Balun + Switch
- 2.4 GHz Transmitter
- 2.4 GHz Receiver
- RF Synthesizer

Peripherals
- SPI
- I2C
- GPIO
- LP IO
- TWAI®
- I2S
- UART
- RTC Watchdog Timer
- GDMA
- PCNT
- RMT
- Super Watchdog
- LED PWM
- ETM
- ADC
- LP UART
- PARLIO
- MCPWM
- USB Serial/JTAG
- LP I2C
- SDIO 2.0 Slave
- Brownout Detector
- Temperature Sensor
- eFuse Controller
- System Timer
- General-Purpose Timers
- Main System Watchdog Timers

Power Management
- Power Management Unit

Security
- SHA
- RSA
- ECC
- AES
- Digital Signature
- HMAC
- RNG
- Clock Glitch Filter
- TEE Controller
- Secure Boot
- Flash Encryption
- APM

Modules having power in specific power modes:
- Active
- Active and Modem-sleep
- Active, Modem-sleep, Light-sleep; optional in Light-sleep
- All modes / optional in Deep-sleep

ESP32P4

- P4 has two full power cores and then a third low-power core that is even more capable than the C6's variant.
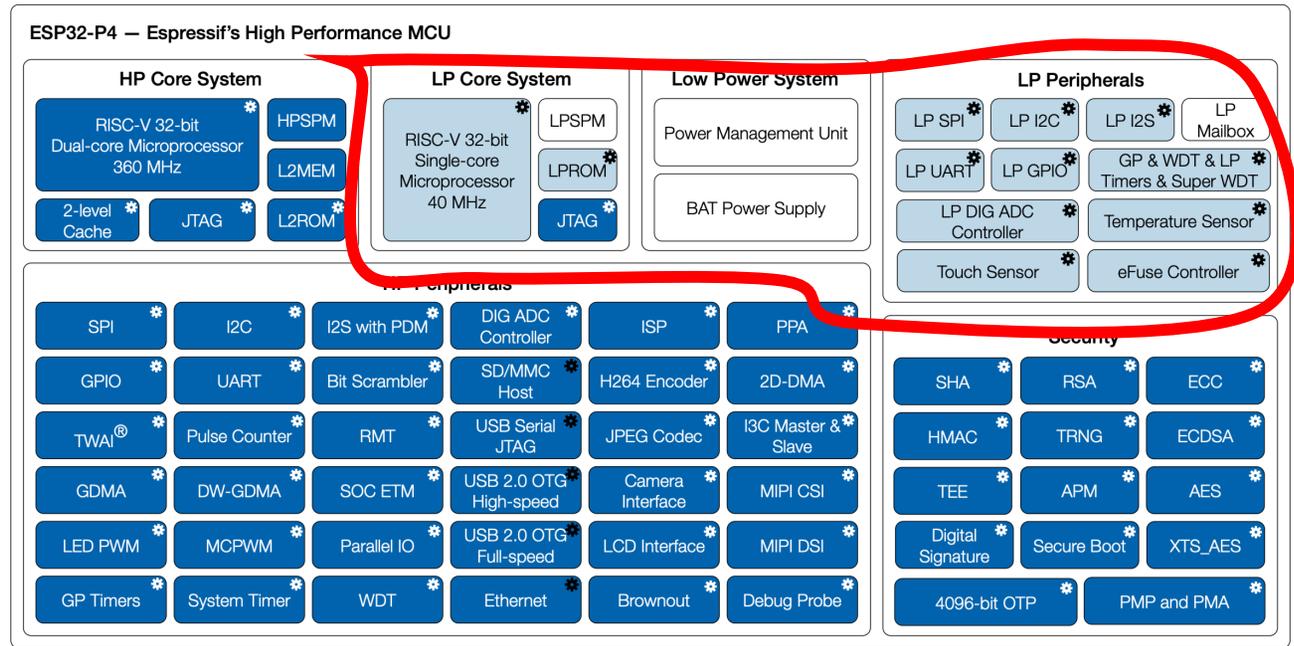
- I have not tested this

LP Caveats!

- The Low-Power cores of the ESP series are pretty well-documented in terms of example code, but very poorly documented in terms of actual specs and measurements.

- If you choose to use them just be aware. There may be some useful functionality there, but also maybe not relative to just ESP32C3

Other Parts?

- Each Part that you end up using needs to be vetted just like how our Federal government rigorously vets people they appoint to lead various departments and offices
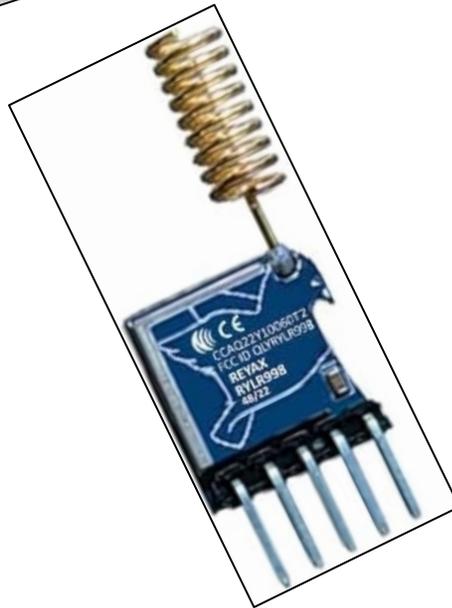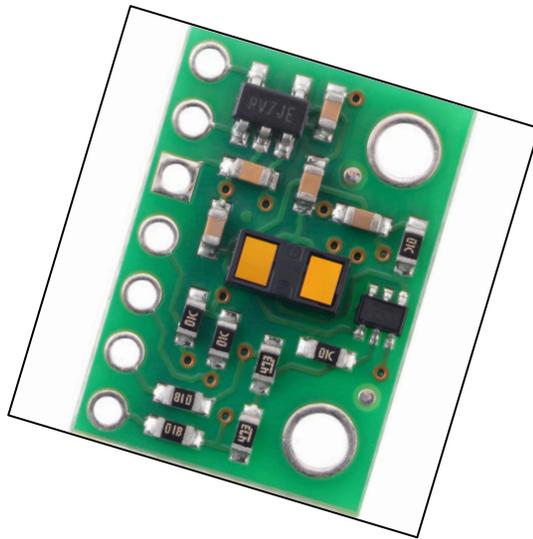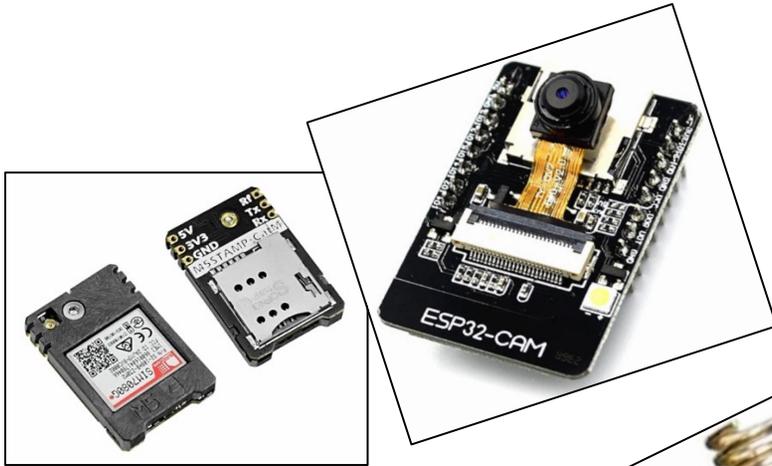
The cellular module from last couple years is rough...

:/

Module type	SIM7080G
Support CAT-M band	B1/B2/B3/B4/B5/B8/B12/B13/B14/B18/B19/B20/B25/B26/B27/B28/B66/B85
Support CAT-NB band	B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66/B71/B85
Cat-M upstream and downstream speed	Uplink: 1119Kbps Downlink:589Kbps
NB-IoT upstream and downstream speed	Uplink: 150Kbps Downlink:136Kbps
RF Power Class	Class 5 (Typ. 21dbm)
SIM Card Slot Specifications	MicroSIM
Standby Operating Current	DC5V/46mA
Network current	DC5V/71mA
Communication Interface	UART: baud 115200 8N1
Net Weight	4.5g
Gross Weight	17.8g
Product Size	30.1*20.1*\5.5mm
Package Size	91.1*135.5mm

https://shop.m5stack.com/products/m5stamp-cat-m-module-sim7080g

SIM7080G Module

- The internet and various semi-legitimate sources seemed to suggest there is a sleep mode for this module…

- At that point, it is supposed to consume 1.2 mA (not great not terrible)

https://www.waveshare.com/wiki/SIM7080G_Cat-M/NB-IoT_HAT

- Operating Frequency
    - GNSS L1: 1575.42±1.023MHz
    - GLONASS: 1597.5~1605.8 MHz
    - BeiDou: 1559.05~1563.14 MHz
    - Galileo L1: 1575.42±1.023MHz
- Refreshing frequency: 1 Hz (by default)
- GNSS data format: NMEA-0183
- GNSS antenna: active antenna

Other Parameters

- Power supply: 5V
- Logic level: 5V / 3.3V (Switchable via jumper caps)
- Overall current (idle mode): 39mA
- Module sole current (VBAT=3.8V)
- Idle mode: 10mA
- Sleep mode: 1.2mA
- PSM mode: 3.2uA
- eDRX mode: 0.59mA (eDRX=81.92s)
- Operating temperature: -40℃ ~ 85℃
- Storage temperature: -45℃ ~ 90℃
- Dimension: 30.5mm x 65mm

Maker of Chip explains how to get there

Program it

Might need to use DTR pin??

https://www.waveshare.com/w/upload/e/e7/SIM7080G_Hardware_Design_V1.03.pdf

DTR Pin?
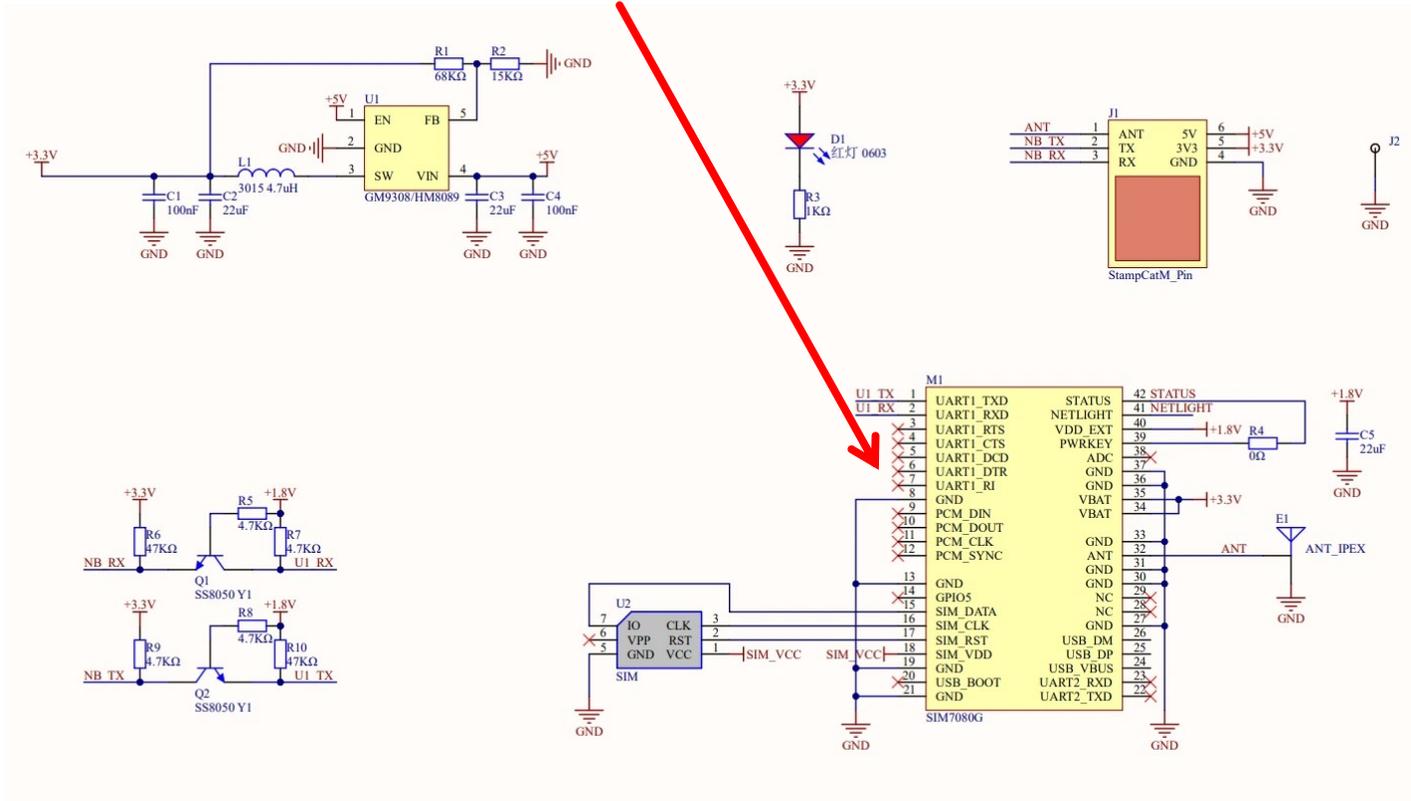
https://docs.m5stack.com/en/stamp/stamp_catm

Needed to solder a wire onto the dev board thing

- That's ok,
- The pins are pretty accessible

- But this needs to be figured out!
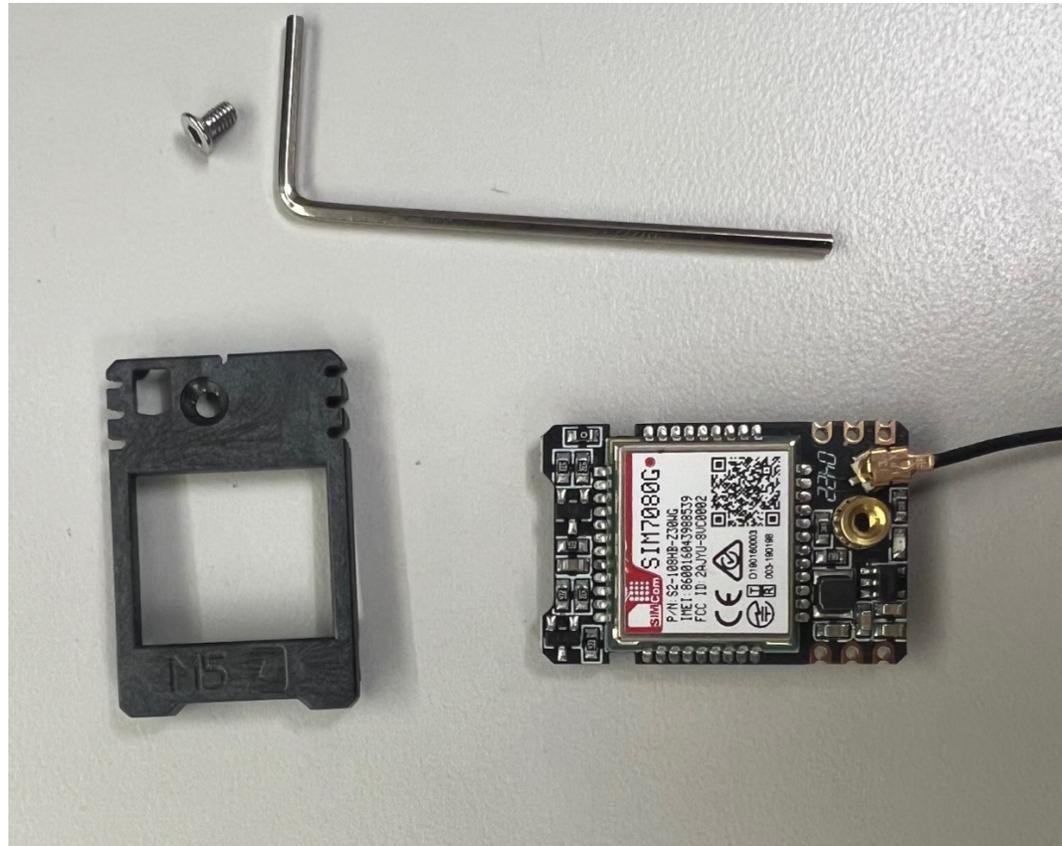
Cell Module

- After all that, I'm actually not even sure if they actually got 1.2 mA out of this thing last year anyways... I seem to recall this was not the case.

- it was an energy hog.

- Maybe figure out another way!

Other Parts?

- Each one is going to be a journey...you will need to do some reverse engineering and thinking about stuff and how it is being used!
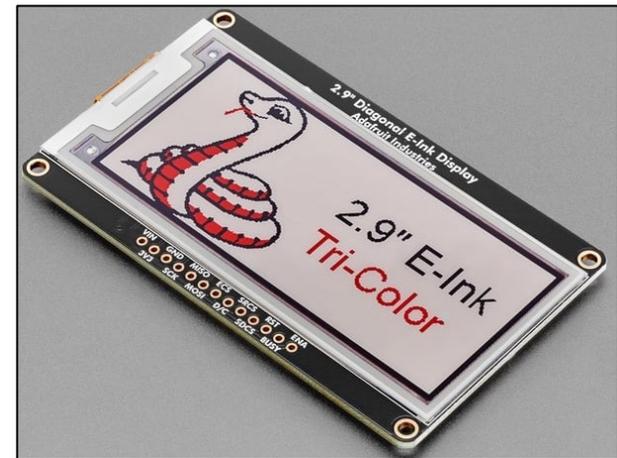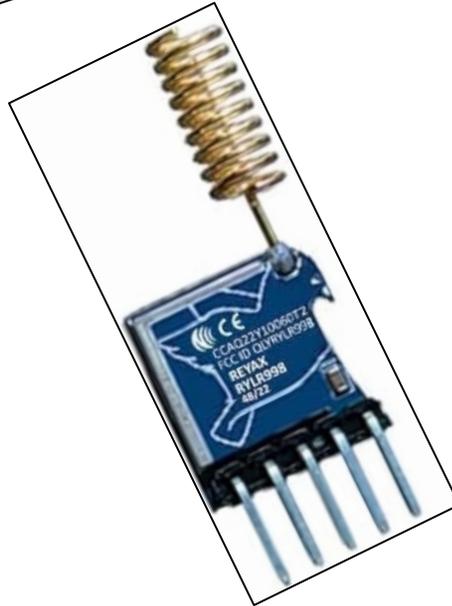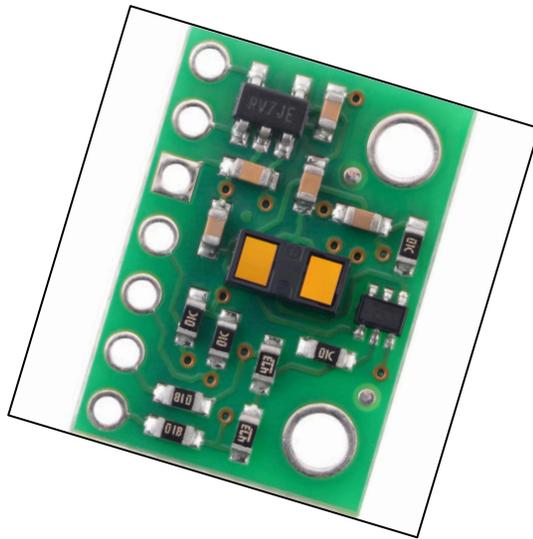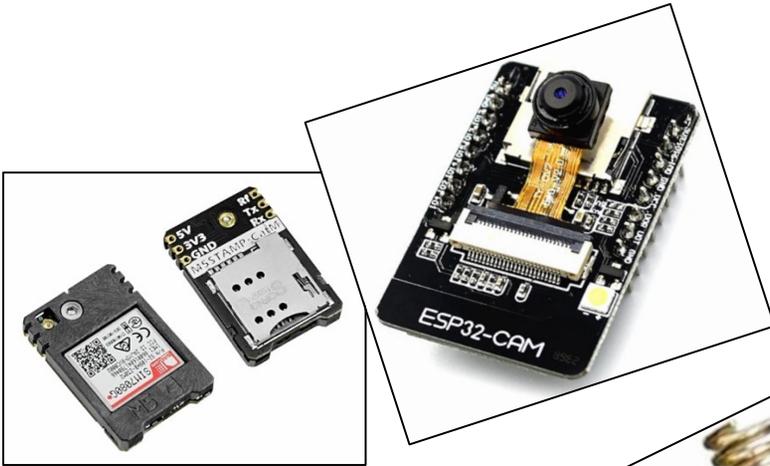
Controlling Other Components

- Ideally external components can be commanded to go into low-power modes (over the comms protocols like UART, I2C, SPI, etc...)
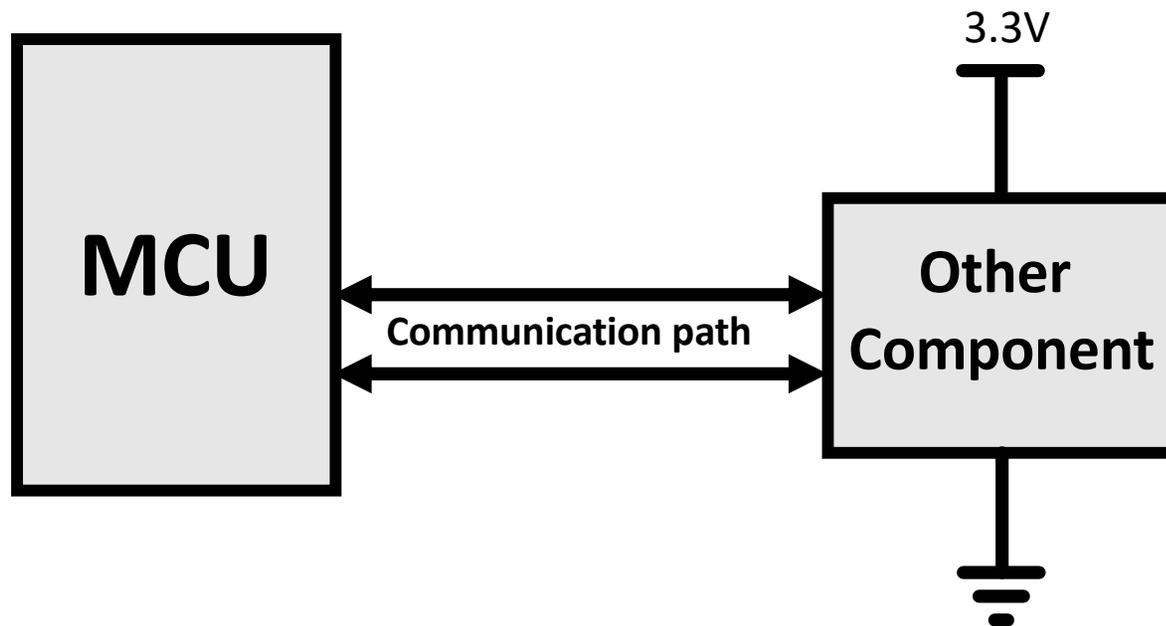
With that said…

- Some types of components may not have power-saving modes

- OR their power-saving modes may not be sufficient

3.3V

MCU

Communication path

Other Component
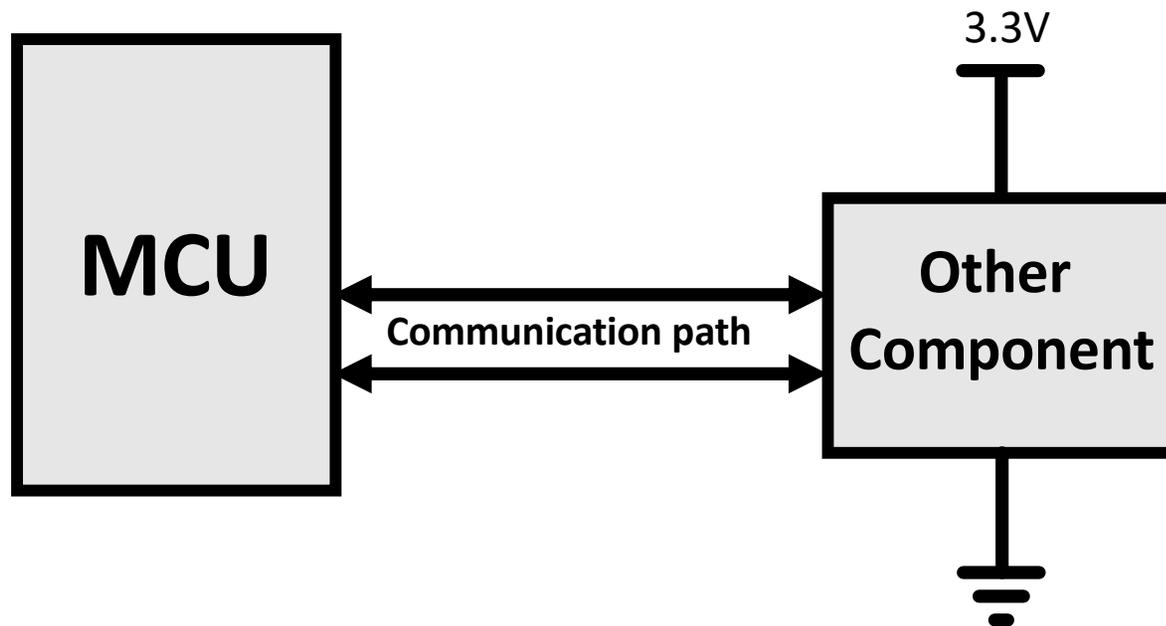
In that case…

- If component needs only a few mA or so to run, it may be possible to power it through a IO pin of the microcontroller

3.3V

MCU

Communication path

Other Component

Should measure and test though…"few mA" at steady state does not mean startup current is few mA
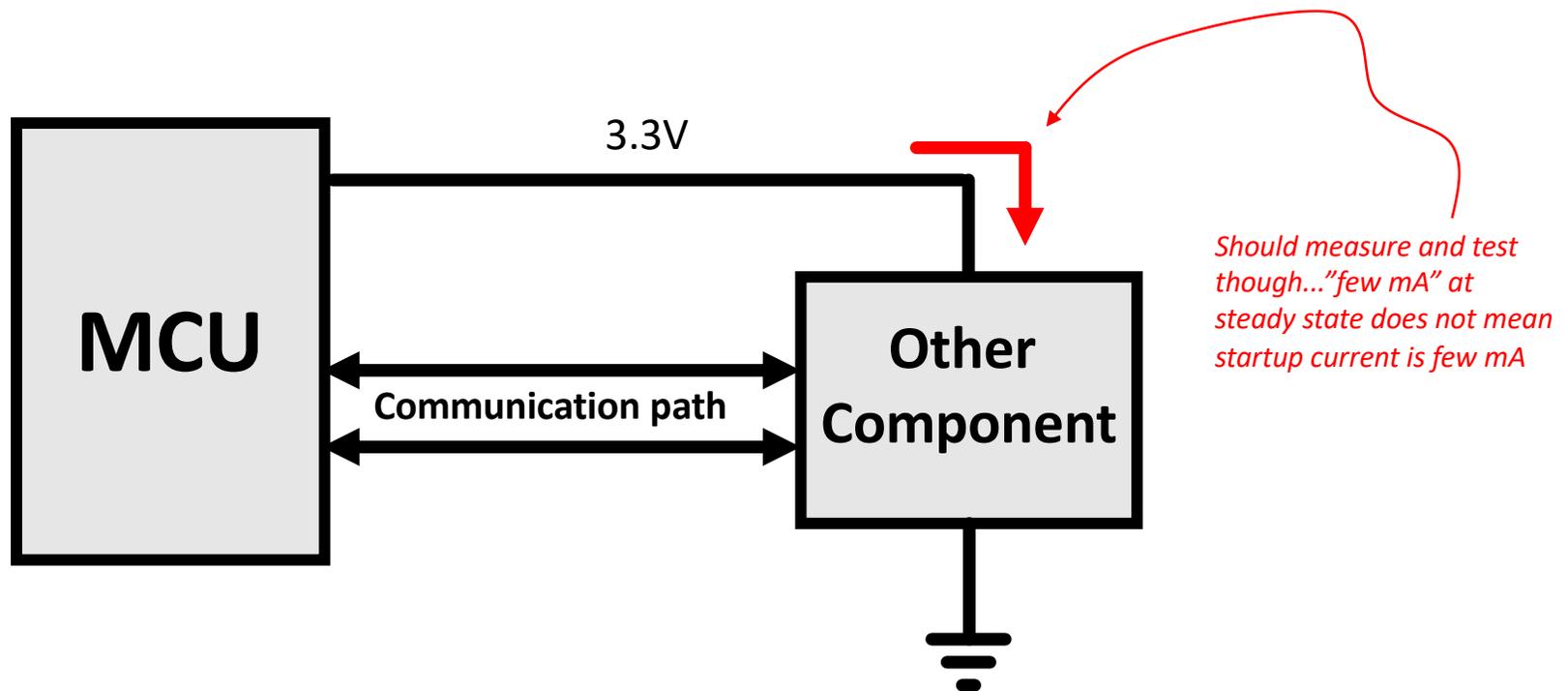
If supply current is higher

- Will likely need a load-switch:
- EX03 you looked at these!
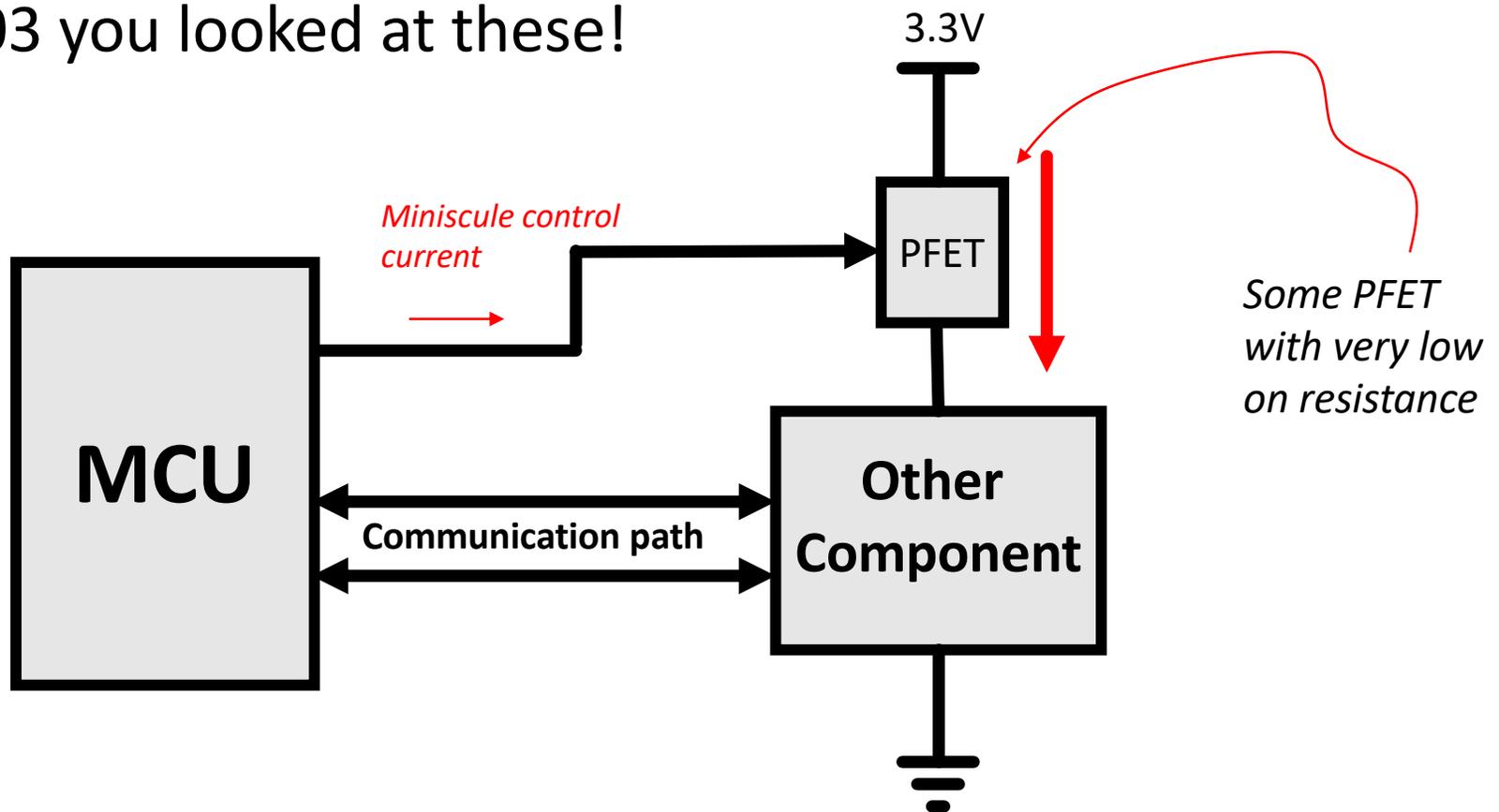
3.3V

PFET

Miniscule control current

MCU

Communication path

Other Component

Some PFET with very low on resistance

Caveat!!!!!

- If power-gating, may need to add isolation!!!

3.3V

PFET **OFF**

Some PFET with very low on resistance

Communication path

Parasitic draw
Out of spec behavior
Phantom-powering
Especially common in i2C

Caveat

- If power-gating, may need to add isolation!!!

3.3V

PFET **OFF**

MCU

Isolation chip, FETs, Buffers

Some PFET with very low on resistance
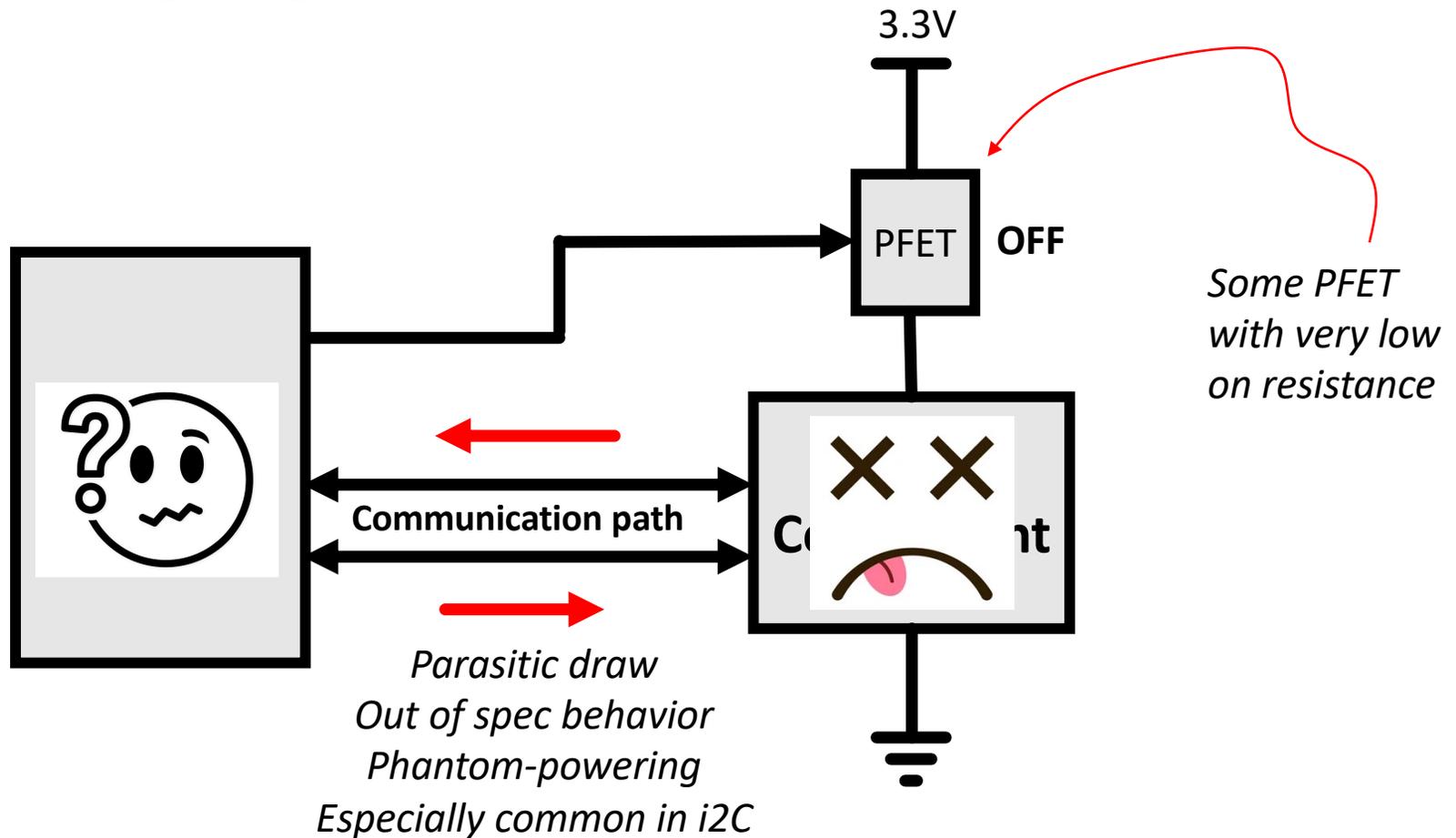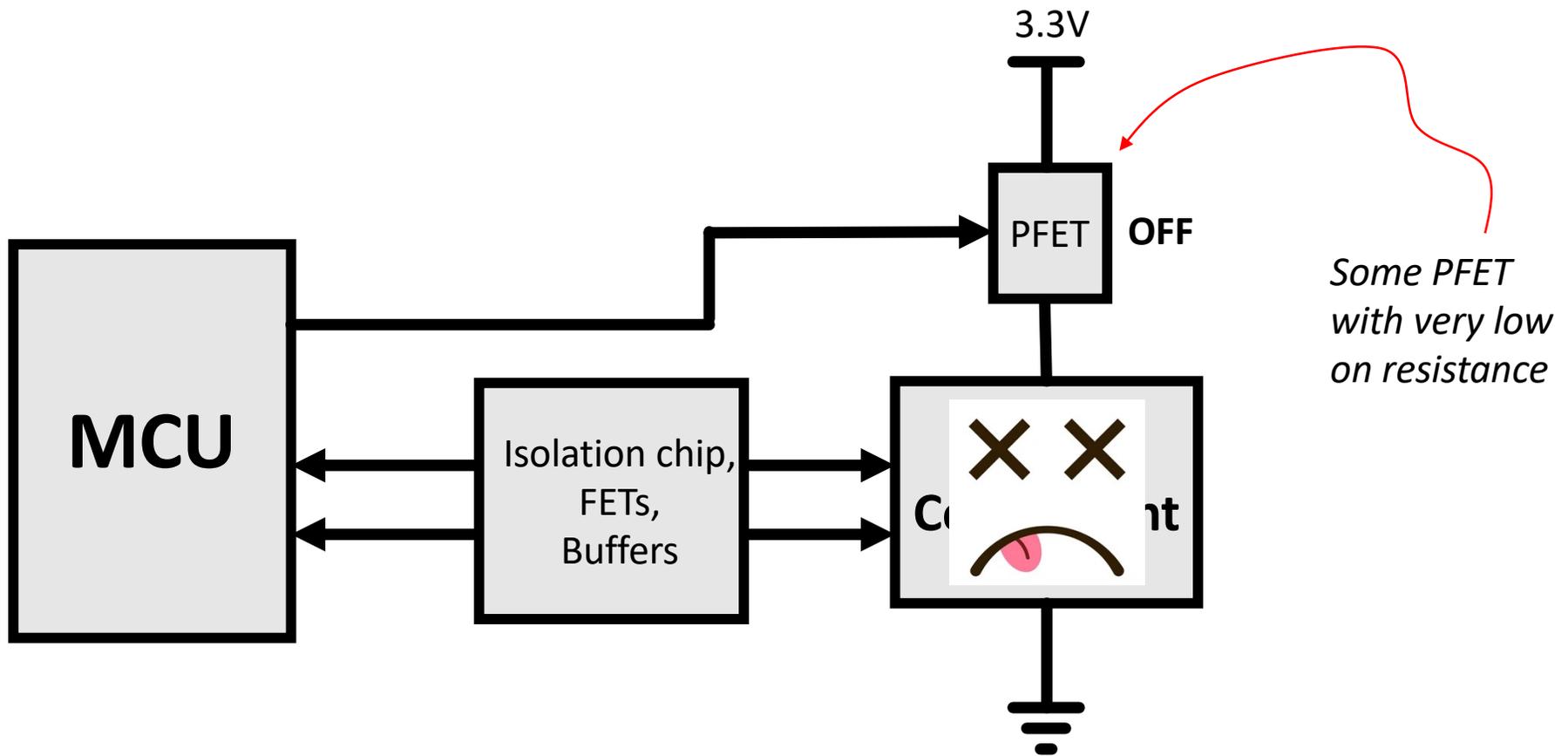
Power Budgeting!

Power Budget

- Now that you getting an idea of the power of your components you need to engineer with that.

- We'll be relying on placing devices into different power modes in loops whenever possible to make them utilize less energy

- These patterns of power consumption can then be roughly time-averaged and help us determine how much our average consumption will be.

- This can then help us figure out how much we need to be putting into the system (from solar, or big battery, etc…)

- And how much we need to store up for when we don't have power input (at night for example)

Duty Cycling

- The core of this budgeting will be duty-cycling the components and behavior when possible

- Turn on in bursts, do what is needed, and turn off otherwise.

- The ratio of the "ON" to overall is called the duty cycle

$$duty\ cycle = \frac{t_{ON}}{T} = \frac{t_{ON}}{t_{ON} + t_{OFF}} \cdot 100\%$$
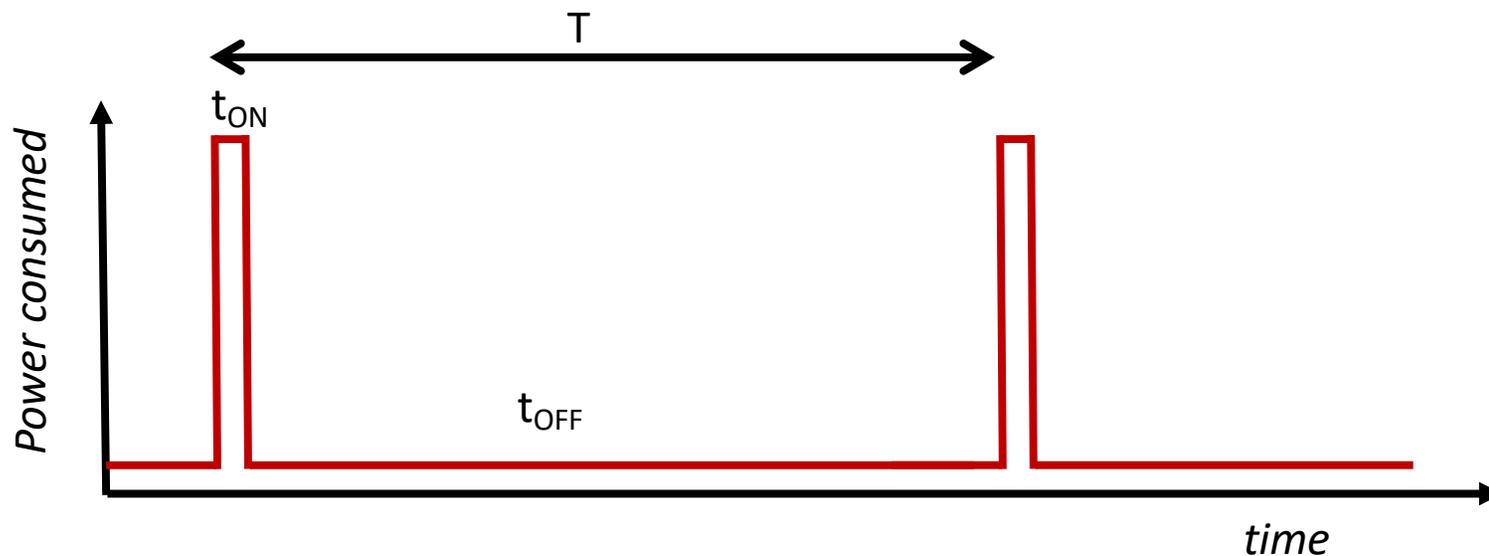
Duty Cycling

- Example:
  - We have an ESP32 powered with a 200 mAh coin-cell battery. Convert to 3.3V at 100% efficiency
  - We want it to run for one month
  - We want to take measurements and send them over WiFi, but **to measure and connect to WiFi requires a t_{ON}= 10 sec.**
  - How often can we make measurements/send them to the server if we want the coin cell to last for one month?

$$duty\ cycle = \frac{t_{ON}}{T} = \frac{t_{ON}}{t_{ON} + t_{SLEEP}} \cdot 100\%$$

T

t_{ON}

t_{SLEEP}

time
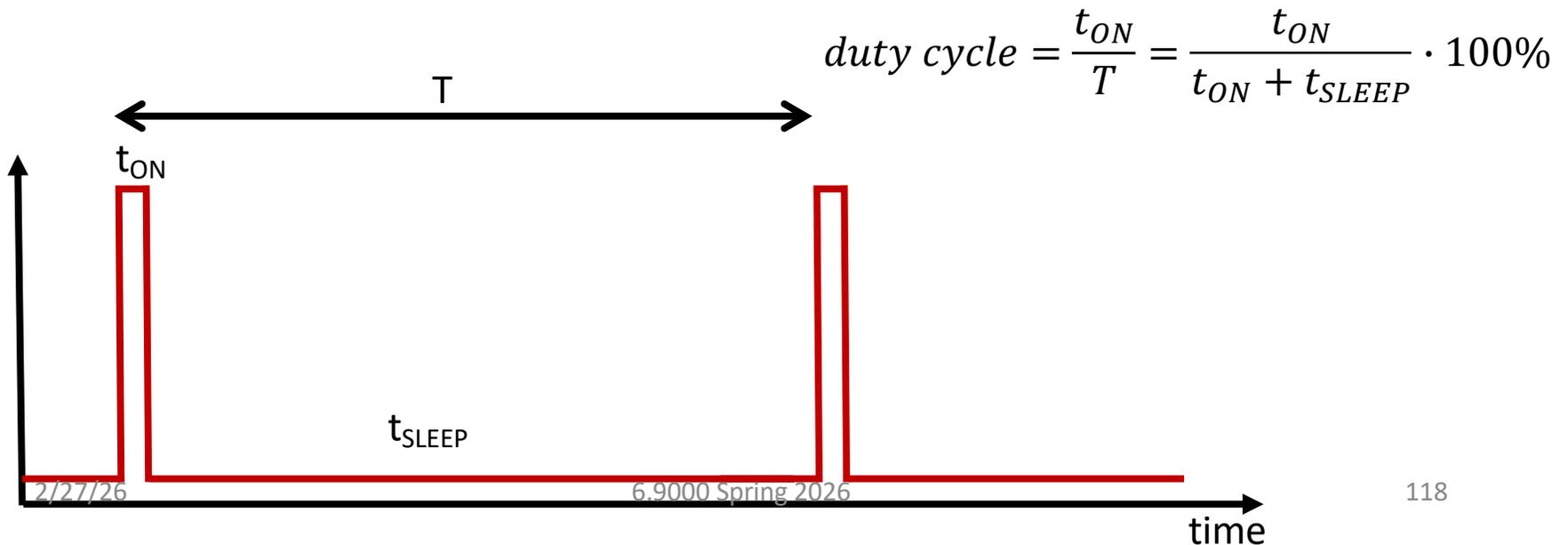
Duty Cycling

- Adjust Duty Cycle to achieve Average Current:

$$= \frac{t_{on} \cdot i_{on} + t_{sleep} \cdot i_{sleep}}{T} = 0.278 \text{ mA}$$

Plug in known values and rewrite in terms of things we're solving for:

$$= \frac{10s \cdot 150mA + t_{sleep} \cdot 0.005mA}{10s + t_{sleep}} = 0.278 \text{ mA}$$

Solve for t_{sleep}:

$$t_{sleep} \approx 5395s$$

So we'd need to:
Come on for 10 seconds
Sleep for 5400 seconds
Duty Cycle: 0.185%

$$duty\ cycle = \frac{t_{ON}}{T} = \frac{t_{ON}}{t_{ON} + t_{OFF}} \cdot 100\%$$
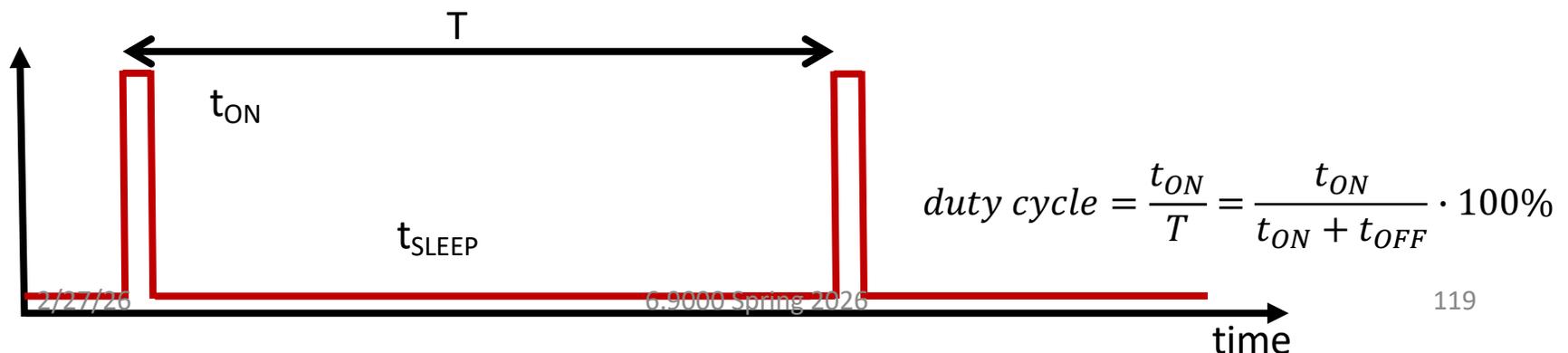
Power Budget

- Now that you're getting an idea of the power of your components you need to engineer with that.

- The first step is to identify your entire system's flowchart of operation.

- Functionality must come first since we need this system to do a thing so you should start with your "ideal" flowchart of operations

Power Budget

- An example from two years ago using the dev boards
- Very similar to some functionality you'll need this year.

- For each state go through and determine what is "on" what is "off" and all the shades of grey in between

Power Budget

- Can do things in either mW or mA...often see mA if you can assume a constant system voltage

- Determine the draw (measured ideal or estimated less-ideal) of each component in the mode it should be in in that state

	A	B	C	D
	Power budget			
		Read sensors		
	MCU subsystem			
	ESP32C3	28.00		
	LED	5.00		
		33.00		
	Sensor subsystem			
	SGP41	3.00		
	SHTC3	0.90		
	LED	5.00		
	I2C	0.66		
		9.56		
	Power subsystem			
	MCP73871	0.03		
	AP7361C	0.06		
	LEDs	5.00		
	Therm	0.05		
	PROG1	3.70		
		8.84		
	cycle time (ms)	51.40		
	10000	70		
		3.6		
		0.4%		

Different System States:

- Fill out the budget as you go along

- Evolve it week to week!

Power budget	Read sensors	Math	Xmit data	Sleep	
MCU subsystem					
ESP32C3	28.00	28.00	345.00	0.13	
LED	5.00	5.00	5.00	5.00	
	33.00	33.00	350.00	5.13	
Sensor subsystem					
SGP41	3.00	0.03	0.03	0.03	
SHTC3	0.90	0.07	0.07	0.07	
LED	5.00	5.00	5.00	5.00	
I2C	0.66	0.33	0.33	0.00	
	9.56	5.43	5.43	5.10	
Power subsystem					
MCP73871	0.03	0.03	0.03	0.03	
AP7361C	0.06	0.06	0.06	0.06	
LEDs	5.00	5.00	5.00	5.00	
Therm	0.05	0.05	0.05	0.05	
PROG1	3.70	3.70	3.70	3.70	
	8.84	8.84	8.84	8.84	
cycle time (ms)	51.40	47.27	364.27	19.07	total current (mA)
10000	70	100	200	9630	duration (ms)
	3.6	4.7	72.9	183.7	charge (mA-sec)
	1.4%	1.8%	27.5%	69.4%	% energy
			Average current/cycle	26.49	mA
			Battery capacity	2200	mA-h
			Lifetime	83.1	h

- Identify problem regions and adjust

- LEDs were always on on these dev boards…

Power budget

	Read sensors	Math	Xmit data	Sleep	
MCU subsystem					
ESP32C3	28.00	28.00	345.00	0.13	
LED	5.00	0.00	0.00	0.00	
	33.00	28.00	345.00	0.13	
Sensor subsystem					
SGP41	3.00	0.03	0.03	0.03	
SHTC3	0.90	0.07	0.07	0.07	
LED	5.00	0.00	0.00	0.00	
I2C	0.66	0.33	0.33	0.00	
	9.56	0.43	0.43	0.10	
Power subsystem					
MCP73871	0.03	0.03	0.03	0.03	
AP7361C	0.06	0.06	0.06	0.06	
LEDs	5.00	5.00	5.00	5.00	
Therm	0.05	0.05	0.05	0.05	
PROG1	3.70	3.70	3.70	3.70	
	8.84	8.84	8.84	8.84	
cycle time (ms)	51.40	37.27	354.27	9.07	total current (mA)
10000	70	100	200	9630	duration (ms)
	3.6	3.7	70.9	87.4	charge (mA-sec)
	2.2%	2.3%	42.8%	52.8%	% energy
			Average current/cycle	16.56	mA
			Battery capacity	2200	mA-h
			Lifetime	132.9	h

Focus

- You need to figure out what your average current draw is going to be!

- ***That number must be less than the average that your source can provide***

- And ideally at least several factors smaller than what the source can provide to account for life sucking and never working out how you want

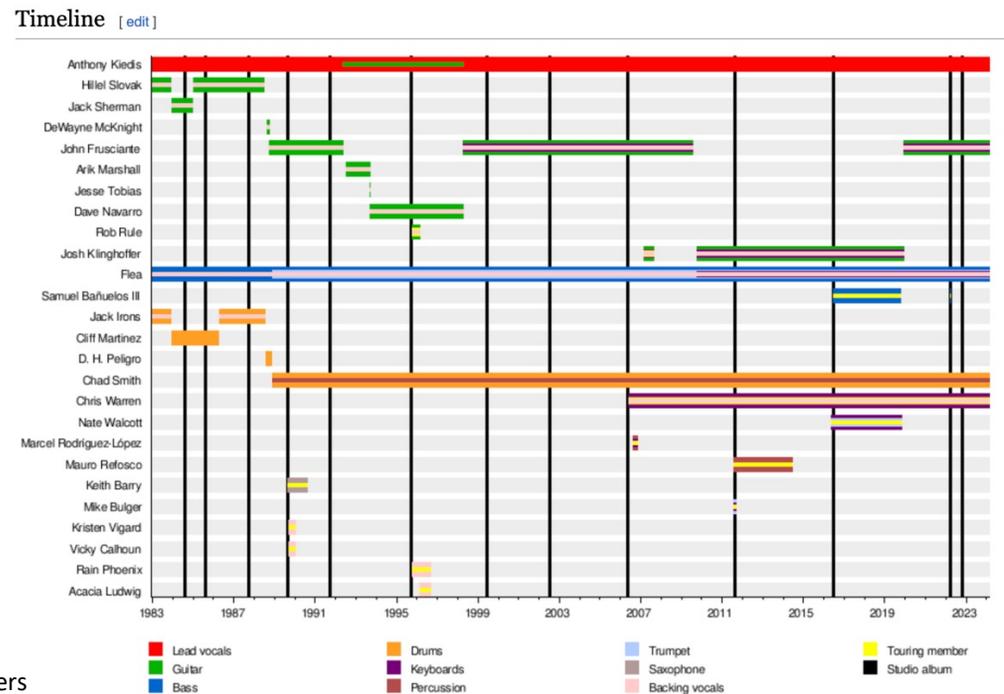|		
Average current/cycle	16.56	mA
Battery capacity	2200	mA-h
Lifetime	132.9	h

- Lifetime can also give you a sense of how long you can go when it is dark out or cloudy or whatever

Timeline

- When you have many components coming on in various states another way to draw this out might be like what they do to show membership of bands on Wikipedia

- Sort of like a Gant chart

Timeline

- But we also want to see actual numbers calculated

- And a spreadsheet will likely be the best choice for this.

	A	B	C
		Read sensors	Math
Power budget			
MCU subsystem			
	ESP32C3	28.00	28.00
	LED	5.00	5.00
		33.00	33.00
Sensor subsystem			
	SGP41	3.00	0.03
	SHTC3	0.90	0.07
	LED	5.00	5.00
	I2C	0.66	0.33
		9.56	5.43
Power subsystem			
	MCP73871	0.03	0.03
	AP7361C	0.06	0.06
	LEDs	5.00	5.00
	Therm	0.05	0.05
	PROG1	3.70	3.70
		8.84	8.84
cycle time (ms)		51.40	47.27
10000		70	100
		3.6	4.7
		0.4%	0.5%

Your System

- Your system may have a much more complicated flow chart

- Likely inner loops…outer loops.

- So you'll be duty-cycling the duty-cyles (and that's fine)

- Using the RTOS and tasks getting pinned to cores can also further complicate things!

Also Keep in Mind Start-Up Costs

- If you send a device to sleep, it cannot just immediately go to sleep and then wake up and start doing stuff again. There will be shutdown and startup costs!

- In modern processors, this is usually 10,000's of clock cycles (meaning potentially milliseconds lost). So if you're going to sleep you better make sure it is worth it

Sleep Mode

Preparing to sleep
(50 us)

Waking Up
(here for 1 ms)

Active Mode
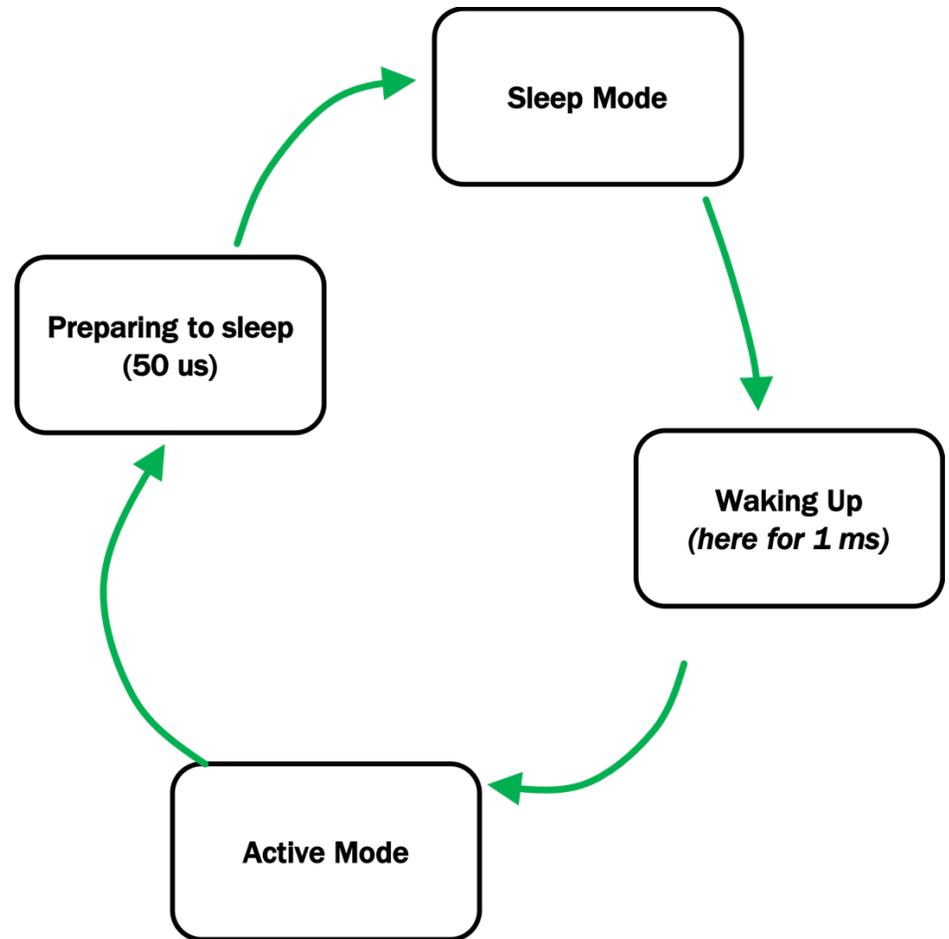
Start-Up Costs

- The ESP32 C3 can come back pretty quick (**50 microseconds for light sleep…500 microseconds from deep I believe**)

- Certain functionalities on the ESP may be much more problematic:
  - Turning on WiFi/getting connections is costly in terms of time (and power)
  - Sending a given packet is relatively cheap
  - May motivate building up data if transferring it over WiFi, though that's limited via other things.
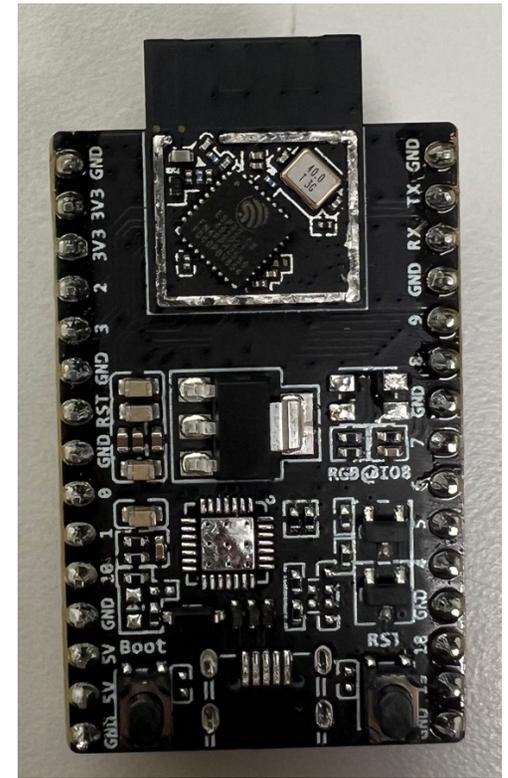
Plot of Sleeping, Waking, Connecting to WiFi

- Connecting to Wifi Fresh takes a lot of energy as opposed to maintaining a periodic connection
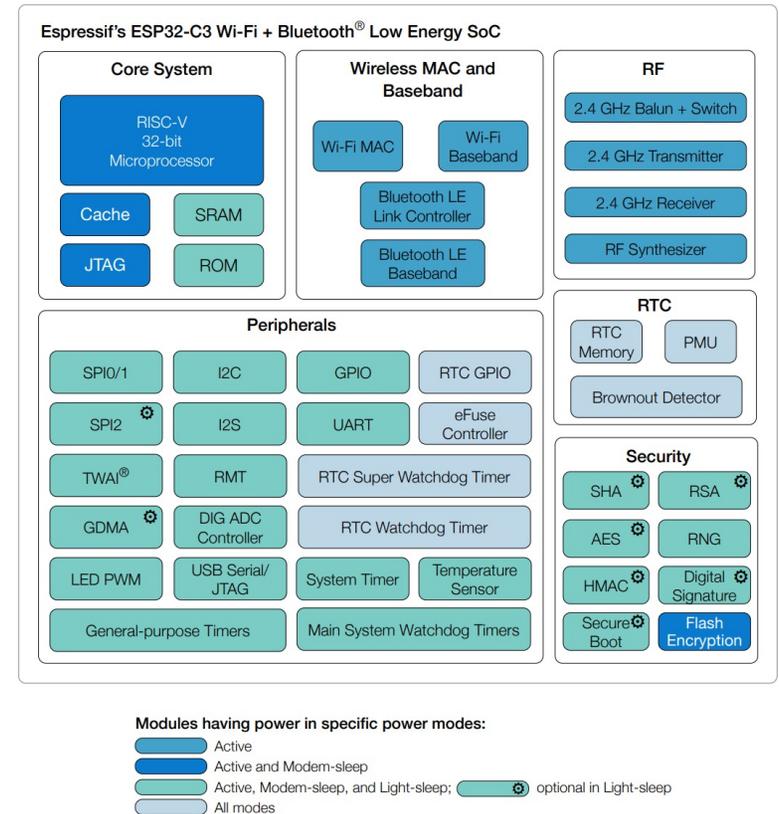- Some nonlinear optimizations need to be figured out
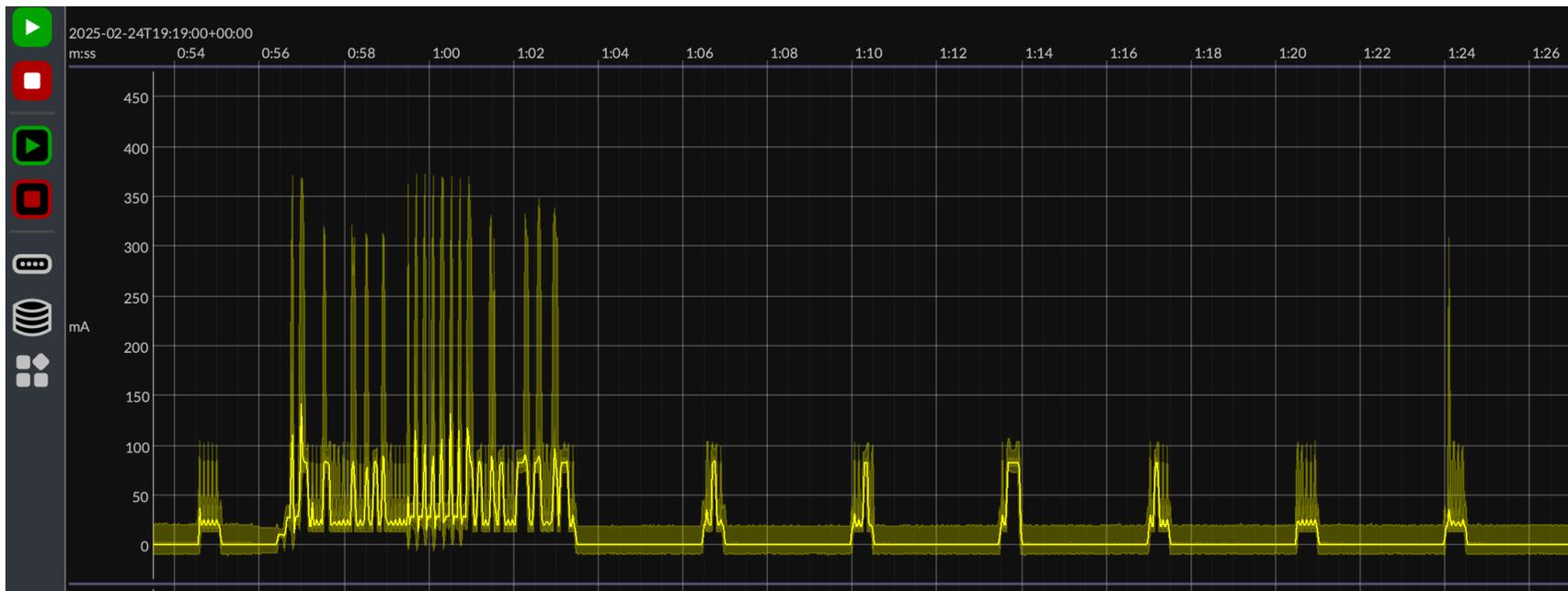
Startup Costs: Other Components

- It takes quite a few minutes for this thing to wake up and find a network successfully...that will need to be figured into everything

- Less well-documented.

- Will need to be studied and measured

- Same might go for other modules!!!

Memory

- Doing periodic sleeps can save energy, but if you're planning to gradually build up data from measurements and then periodically post it to server you have to think about how to save data during those sleeps!!!

- Light Sleep: All memory stays active

- Deep sleep only 8 kB of RTC memory is available/everything else gets reset

Espressif's ESP32-C3 Wi-Fi + Bluetooth® Low Energy SoC

Modules having power in specific power modes:
- Active
- Active and Modem-sleep
- Active, Modem-sleep, and Light-sleep; optional in Light-sleep
- All modes

Making Code More Efficient

- The faster your code runs, the better it is:
  - Can turn off quicker, or can go to the next task quicker…
  - Either way bleeding less power just by being on

- This applies to on the server as well.
  - Just because the power to run your python comes from a power plant in Canada or New Jersey doesn't mean we should ignore it
  - Data centers consume an ever-increasing portion of the US energy consumption in the US (~4.0% currently, expected to be as high as 9.0% by end of decade)

No matter what you will then need to get actual data!!!
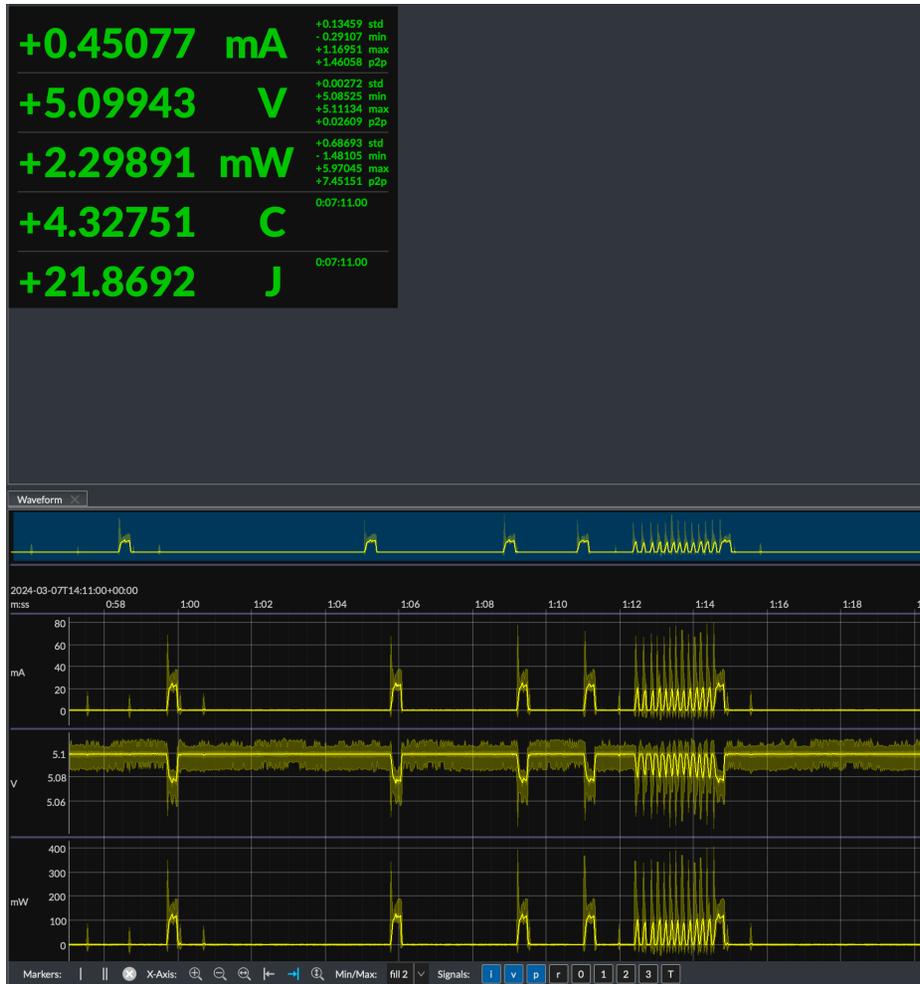
- Compare your estimated power budget records to actual corrected data.

- Compare and contrast and find where the issues are!

Joulescope!!!

+947.371 μA
+9.23014 std
+908.889 min
+985.173 max
+76.2841 p2p

+3.30697 V
+0.00060 std
+3.30414 min
+3.30957 max
+0.00542 p2p

+3.13292 mW
+0.03059 std
+3.00544 min
+3.25831 max
+0.25286 p2p

+391.629 mC
27.5 s

+1.29364 J
27.5 s

Can actually integrate energy used over time

Run your system for a while and get actual real comparisons to your approximated numbers