

TFE4152 - Lecture 13

CMOS Logic

Source

| Week | Book | Monday | Book | Friday |
|------|-------------------------|---|-------------------------|--------------------------------------|
| 34 | | Introduction, what are we going to do WH 1 , WH 15 in this course. Why do you need it? | | Manufacturing of integrated circuits |
| 35 | CJM 1.1 | pn Junctions | CJM 1.2 WH 1.3, 2.1-2.4 | Mosfet transistors |
| 36 | CJM 1.2 WH 1.3, 2.1-2.4 | Mosfet transistors | CJM 1.3 - 1.6 | Modeling and passive devices |
| 37 | | Guest Lecture - Sony | CJM 3.1, 3.5, 3.6 | Current mirrors |
| 38 | CJM 3.2, 3.3,3.4 3.7 | Amplifiers | CJM, CJM 2 WH 1.5 | SPICE simulation |
| 39 | | Verilog | | Verilog |
| 40 | WH 1.4 WH 2.5 | CMOS Logic | WH 3 | Speed |
| 41 | WH 4 | Power | WH 5 | Wires |
| 42 | WH 6 | Scaling Reliability and Variability | WH 8 | Gates |
| 43 | WH 9 | Sequencing | WH 10 | Datapaths - Adders |
| 44 | WH 10 | Datapaths - Multipliers, Counters | WH 11 | Memories |
| 45 | WH 12 | Packaging | WH 14 | Test |
| 46 | | Guest lecture - Nordic Semiconductor | | |
| 47 | CJM | Recap of CJM | WH | Recap of WH |

Goal for today

Analog transistor to digital transistor

Fundamental static logic cells

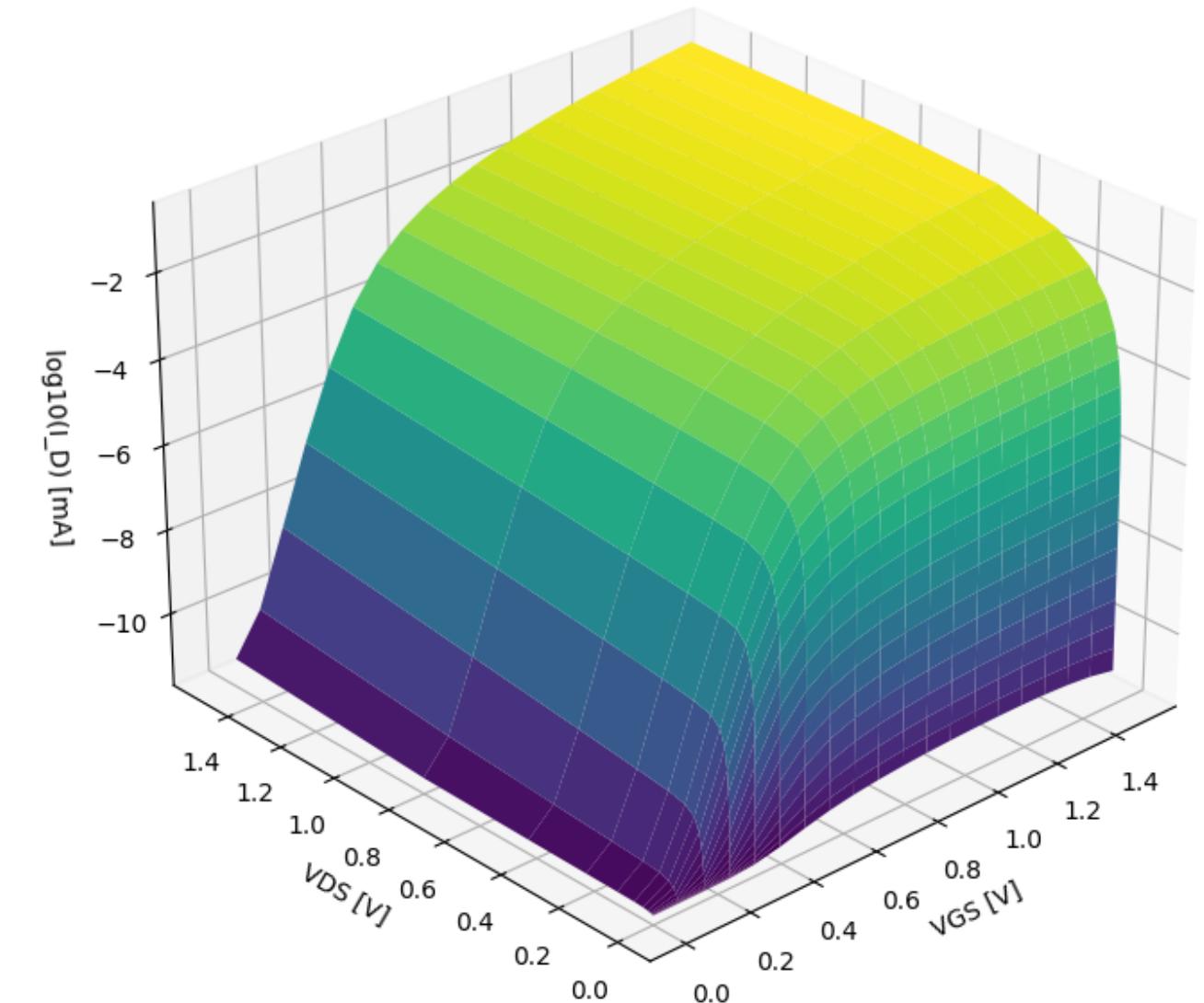
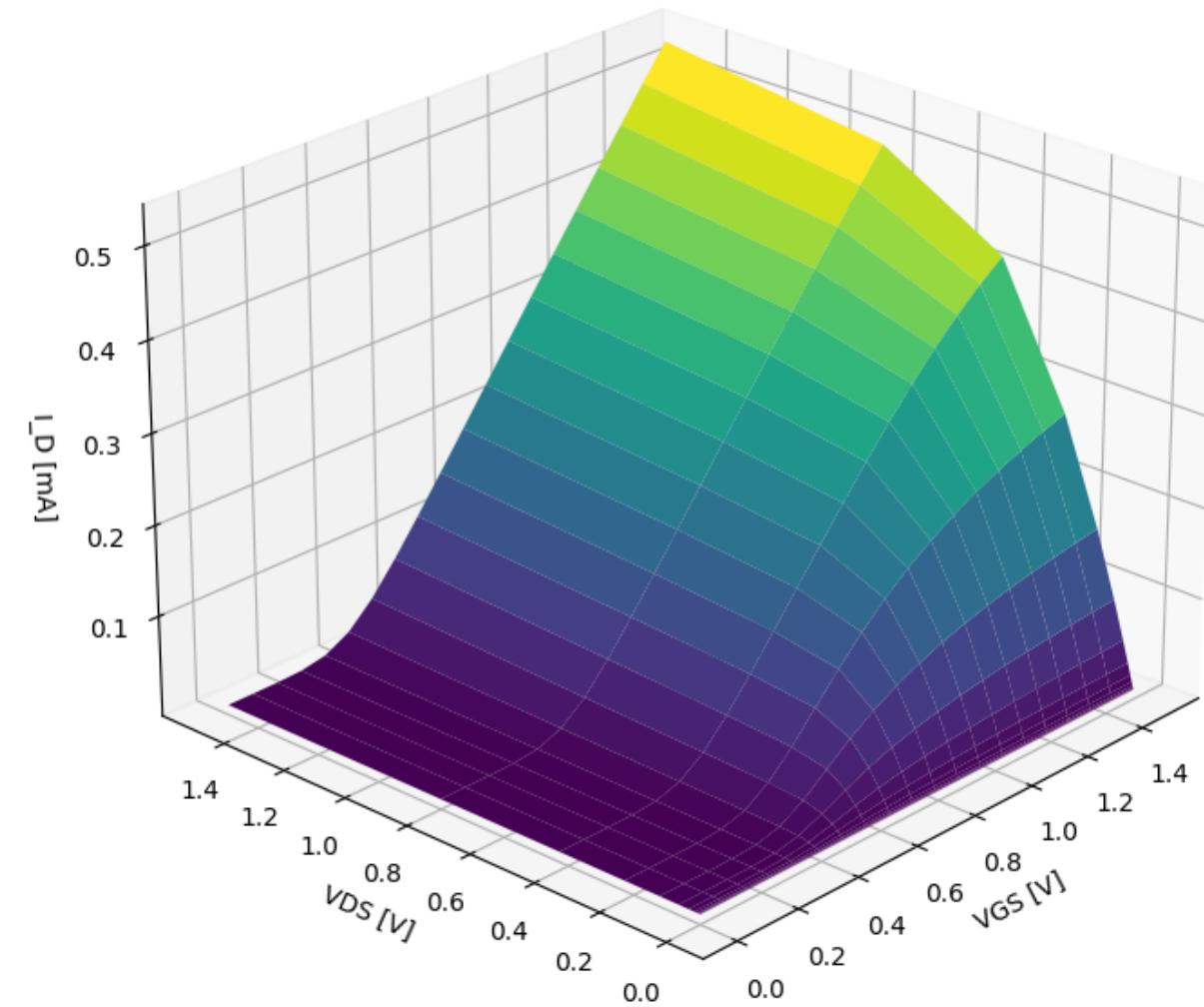
Other static logic cells

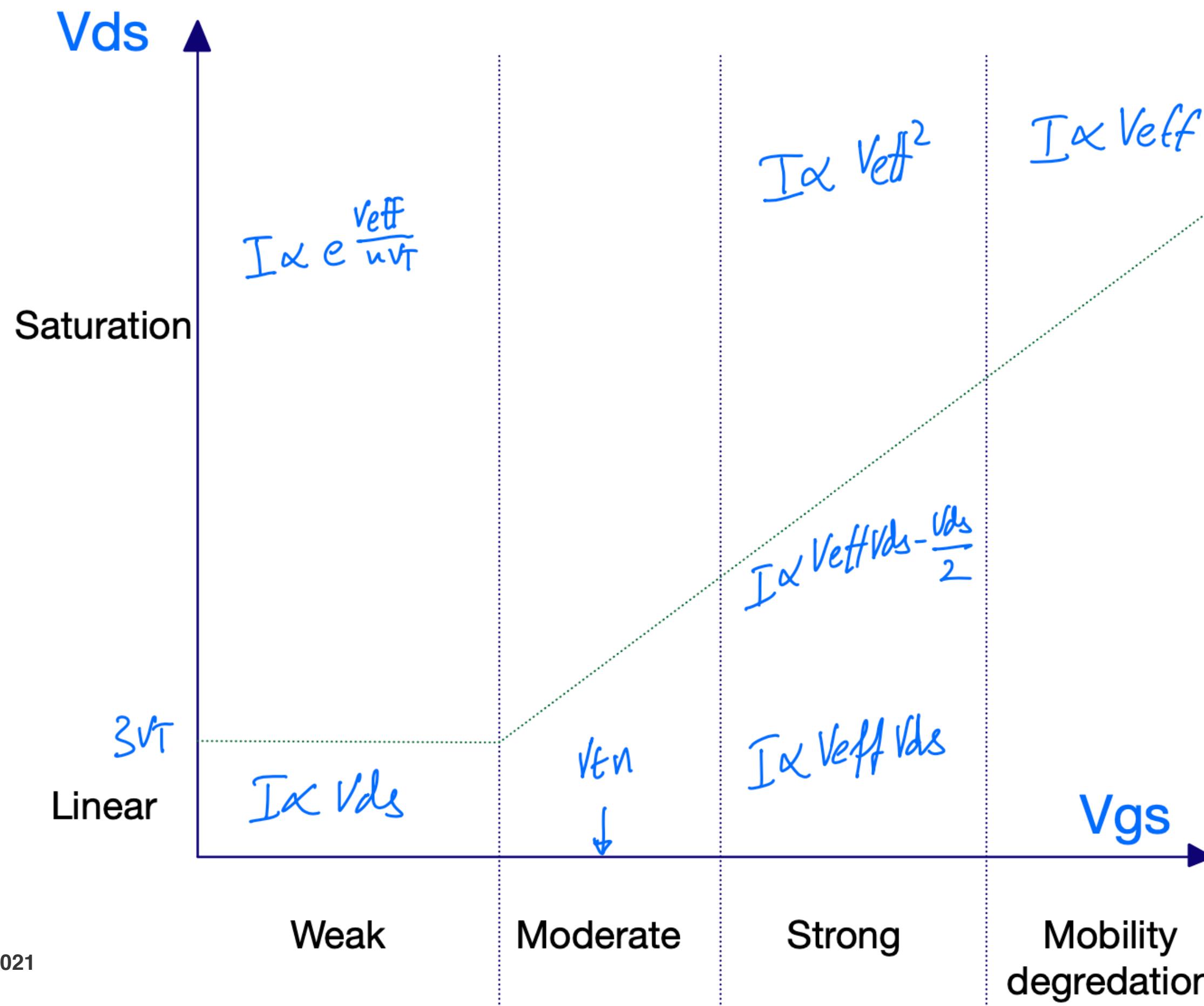
Project Questions

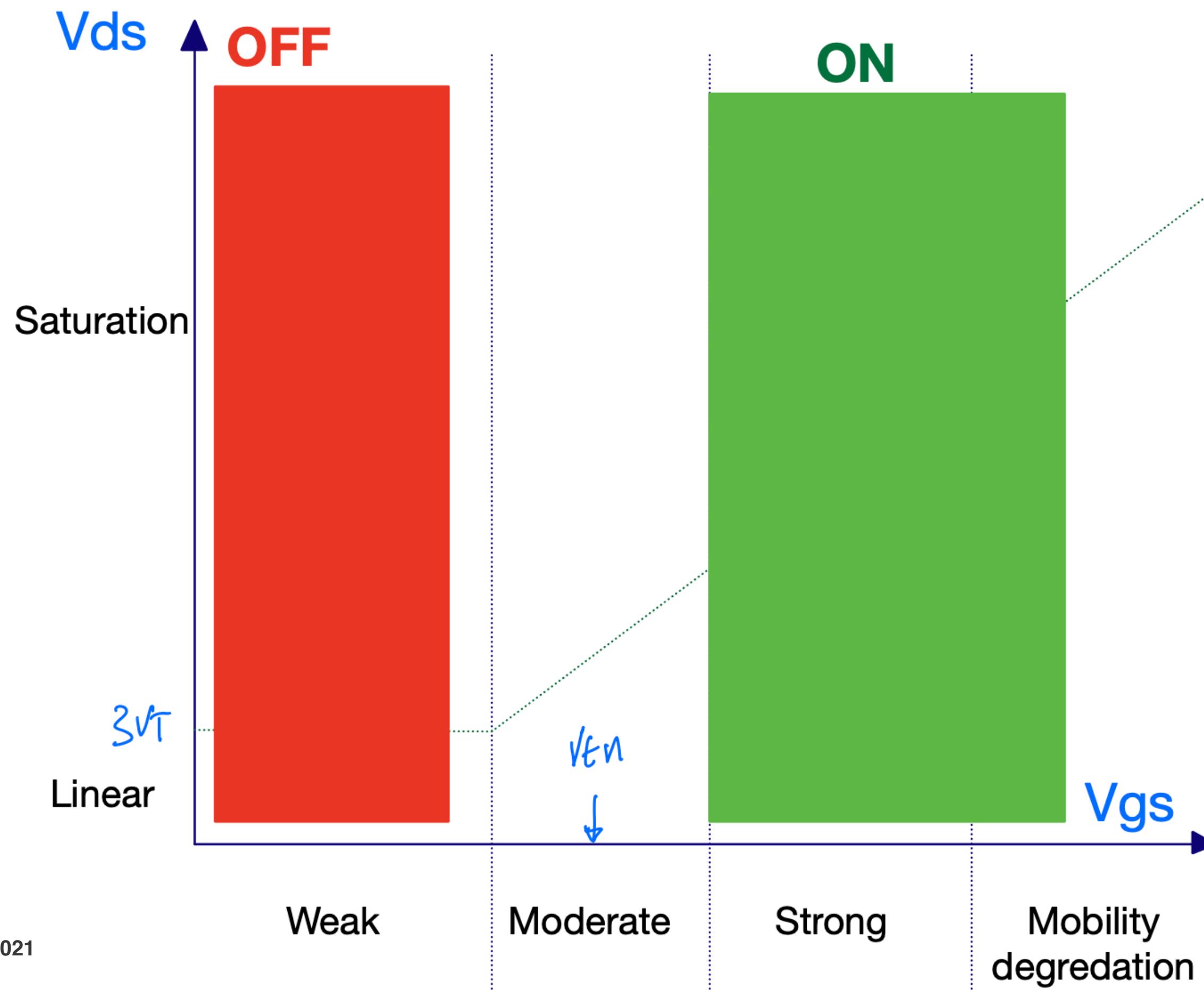
(If time: [Sesame Demo](#))

Analog transistor to digital transistor

NMOS current ($W = 0.4\mu$ $L=0.15\mu$) as a function of V_{GS} and V_{DS}







Gate**NMOS****PMOS**

VDD

ON

OFF

VDD -> VSS

X

X

VSS -> VDD

X

X

VSS

OFF

ON

Gate

NMOS

PMOS

1

ON

OFF

$1 \rightarrow 0$

X

X

$0 \rightarrow 1$

X

X

0

OFF

ON

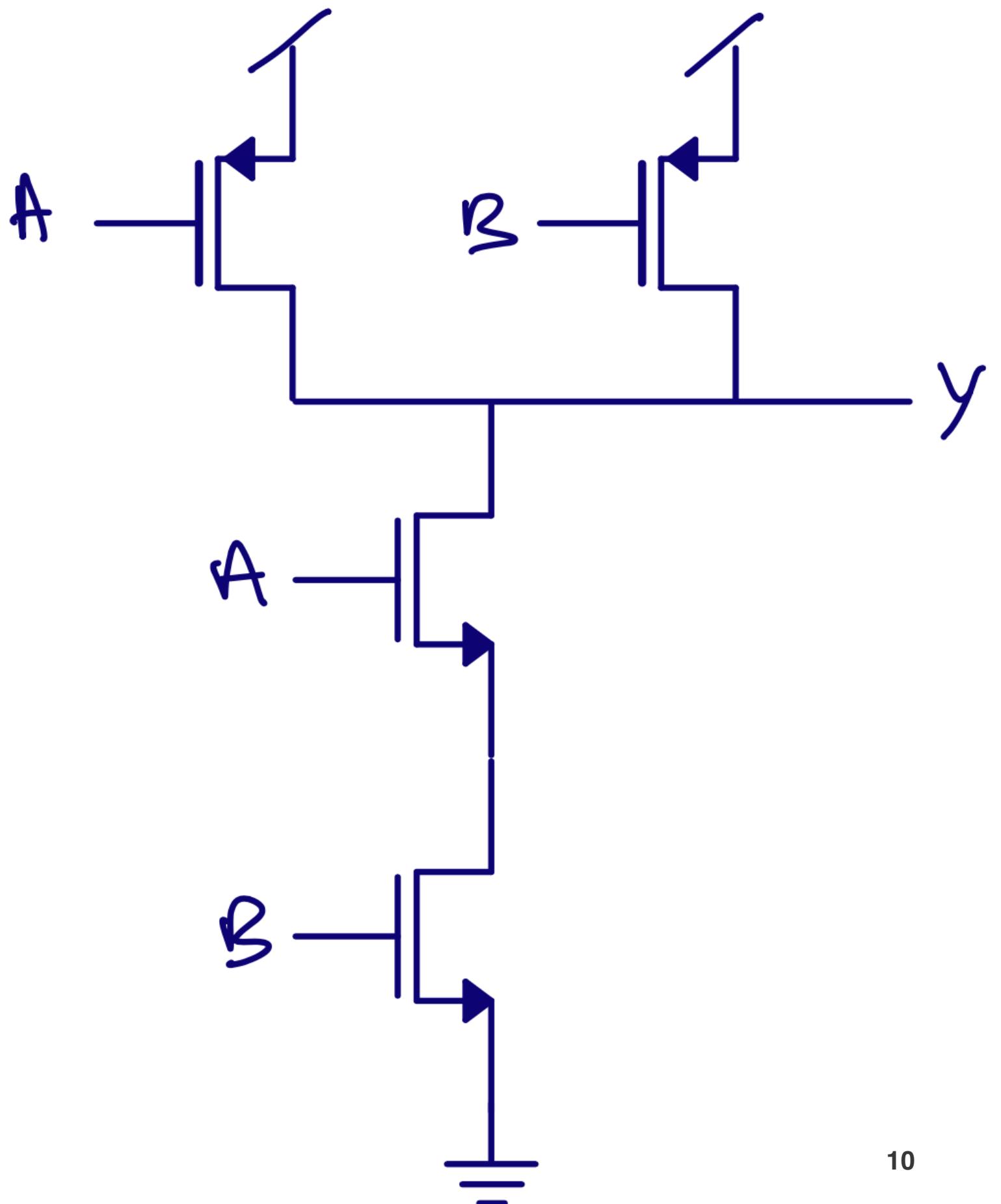
CMOS static logic assumptions

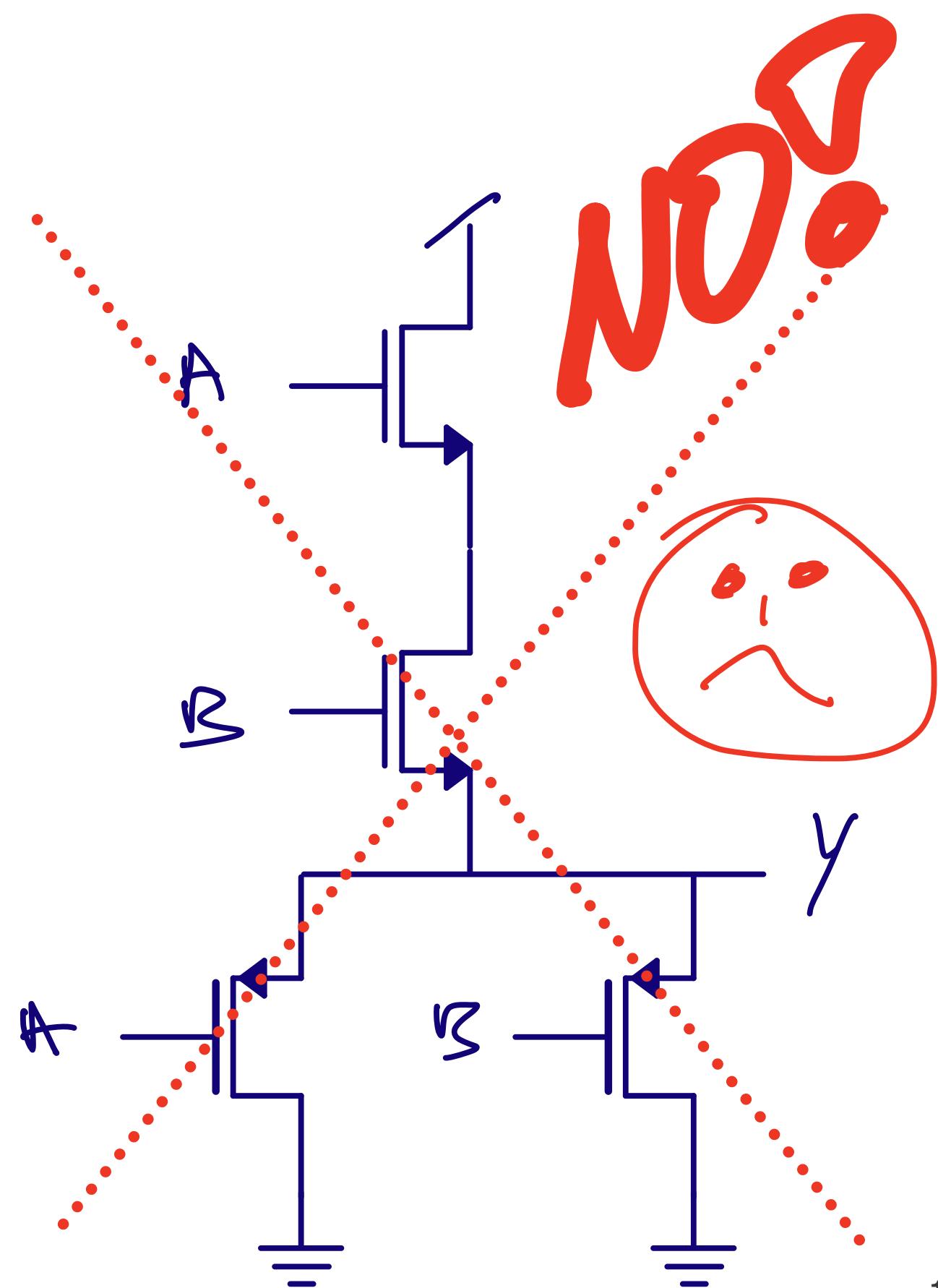
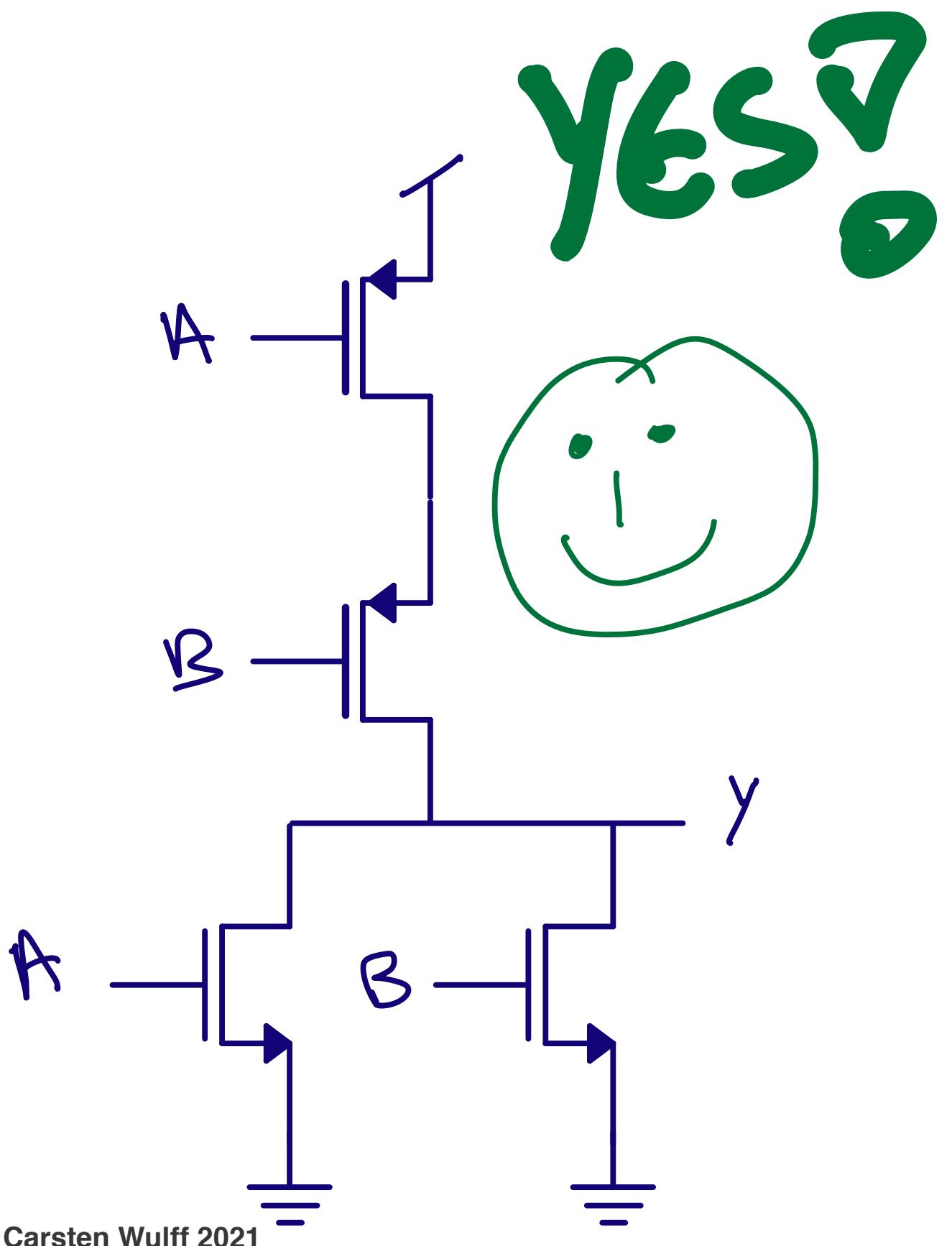
NMOS source is connected to low potential

$$V_{GS} > V_{TH} \text{ when } V_G = V_{DD}$$

PMOS source is connected to high potential

$$V_{GS} < V_{TH} \text{ when } V_G = 0$$





Don't break rules unless you know
exactly why it will be OK

Logic cells

| A | B | \overline{AB} | $\overline{A+B}$ | AB | $A+B$ |
|-----|-----|-----------------|------------------|------|-------|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |

$$\overline{AB} = \overline{\overline{A} + \overline{B}} \quad \leftarrow DM$$

$$\overline{A+B} = \overline{\overline{A} \overline{B}} \quad \leftarrow DM$$

$$AB = \overline{\overline{\overline{A} + \overline{B}}}$$

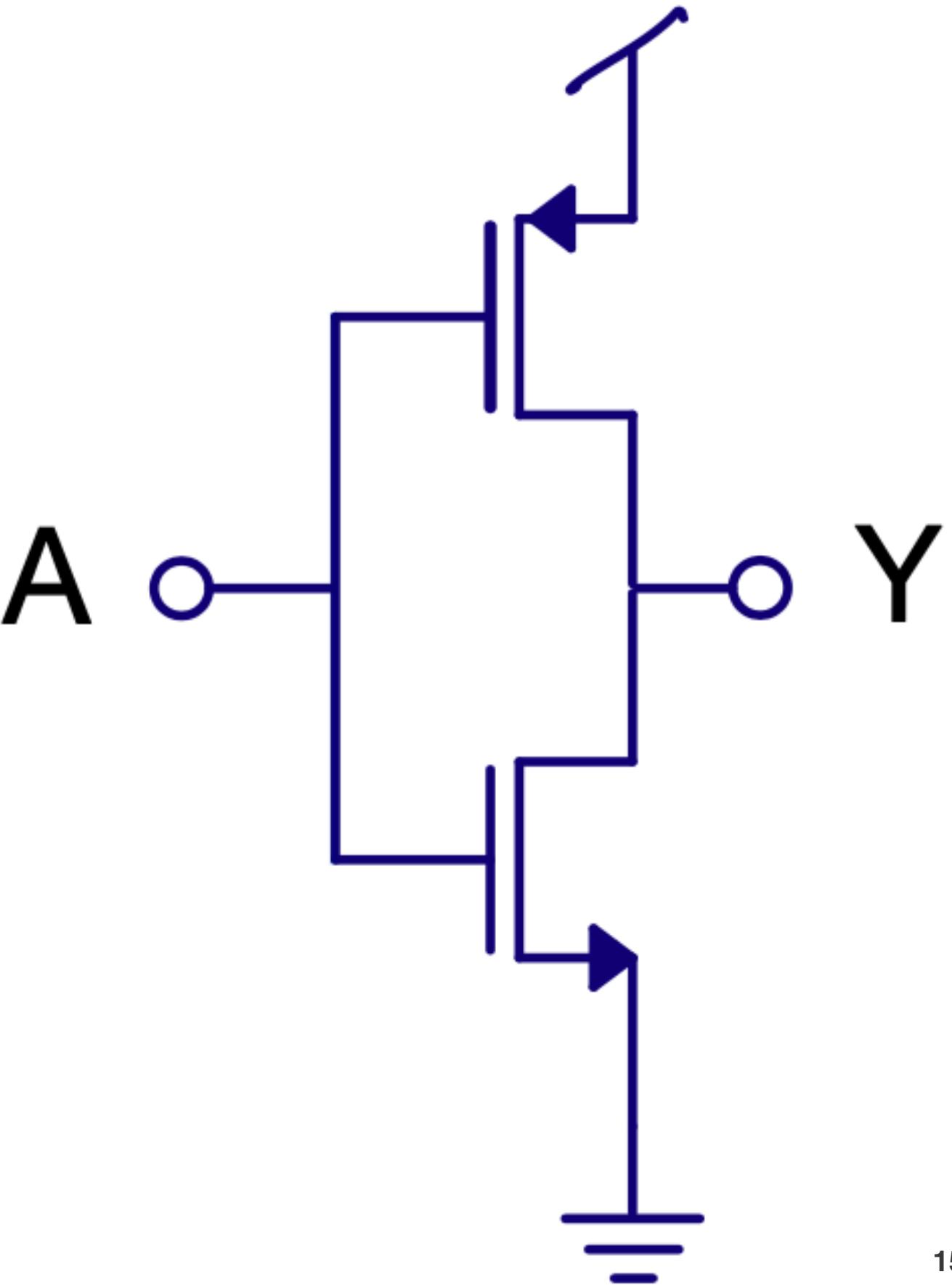
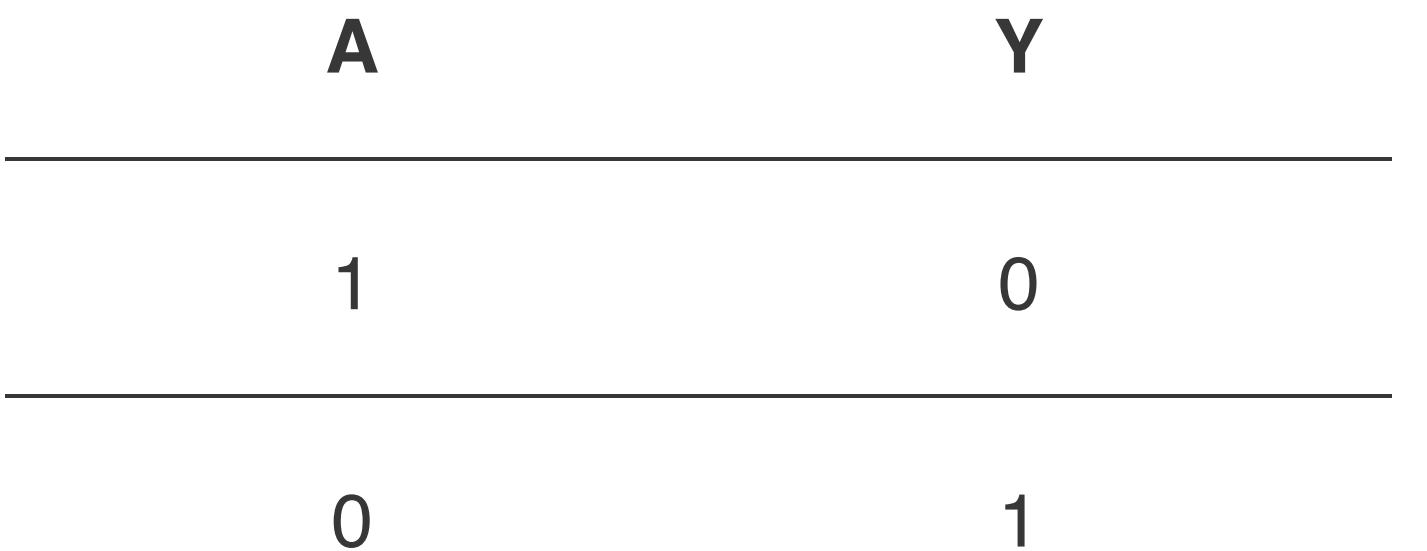
$$A+B = \overline{\overline{AB}}$$

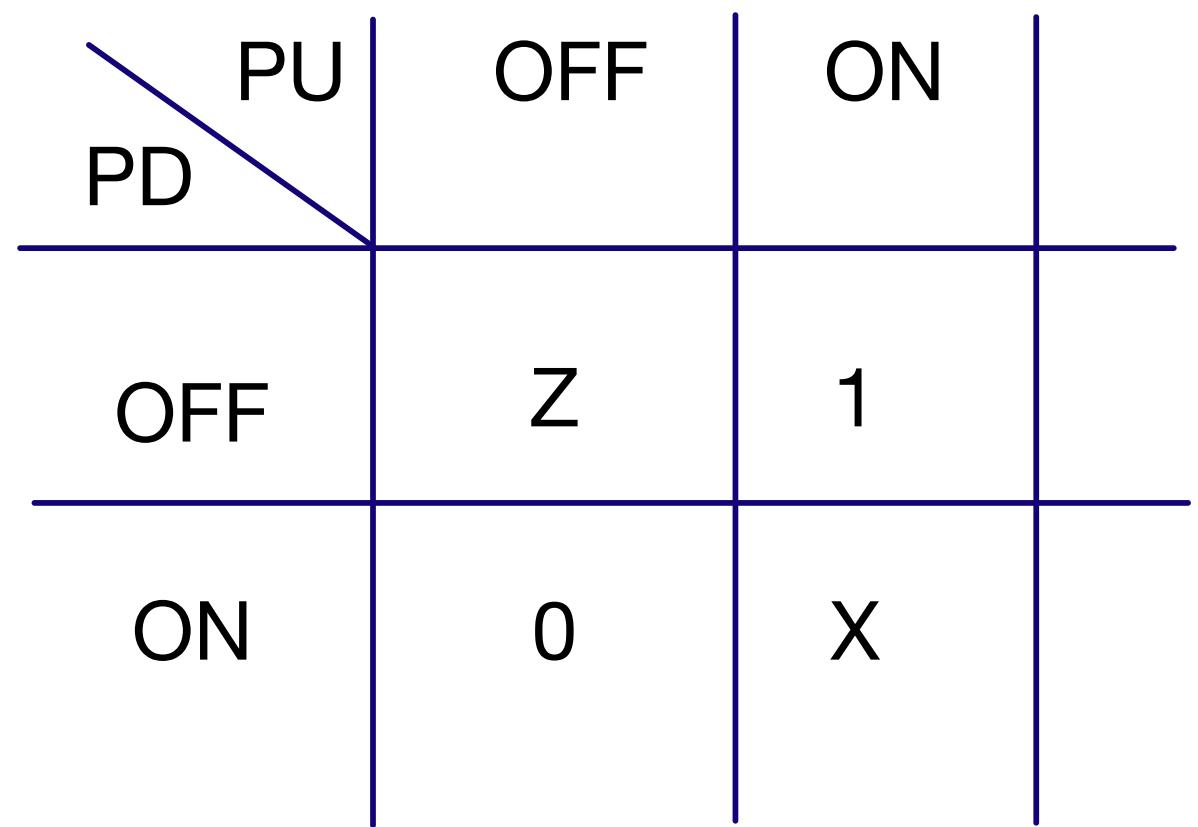
| \overline{A} | \overline{B} | $\overline{A} + \overline{B}$ | \overline{AB} | $\overline{\overline{A} + \overline{B}}$ | $\overline{\overline{AB}}$ |
|----------------|----------------|-------------------------------|-----------------|--|----------------------------|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |

$$\overline{AB} \neq \overline{\overline{A} \overline{B}}$$

$$\overline{A+B} \neq \overline{\overline{A} + \overline{B}}$$

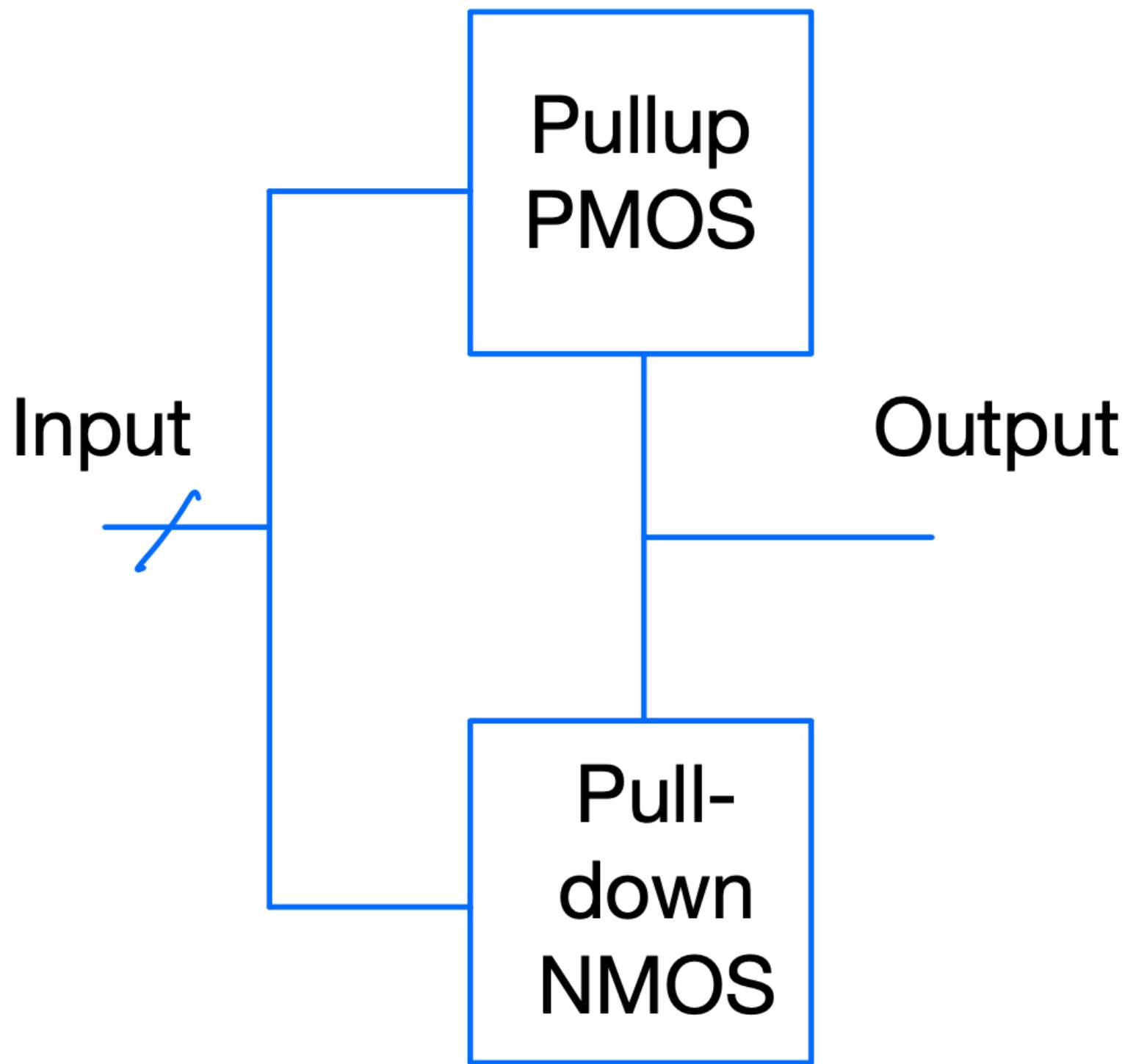
CMOS static logic is inverting





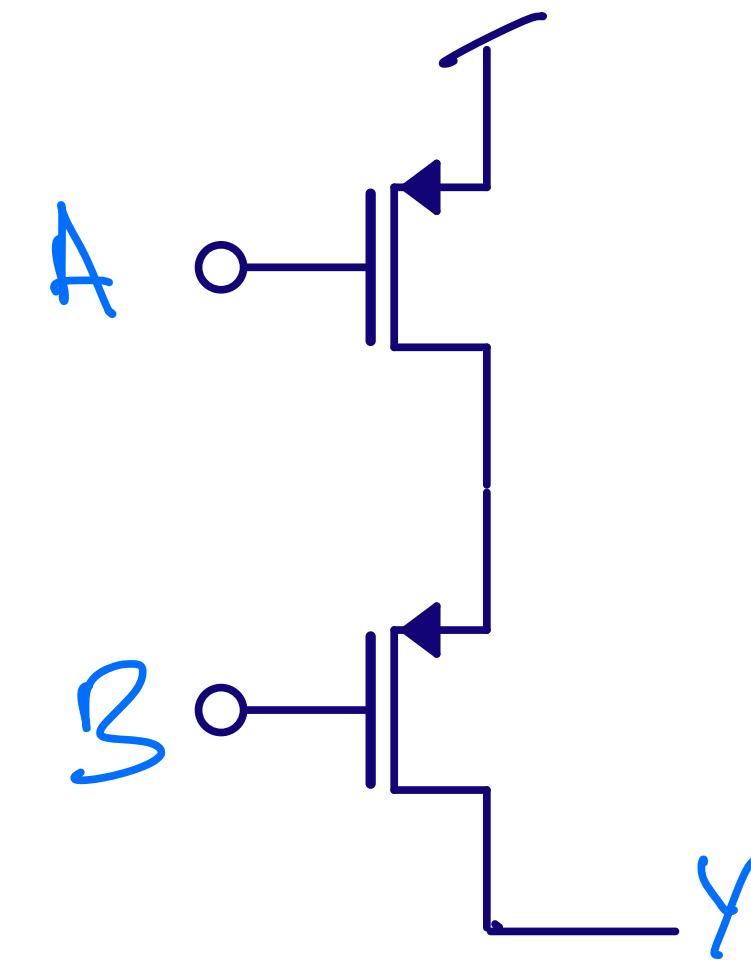
PD = Pull-down PU = Pull-up

logic => [0,1,Z,X];



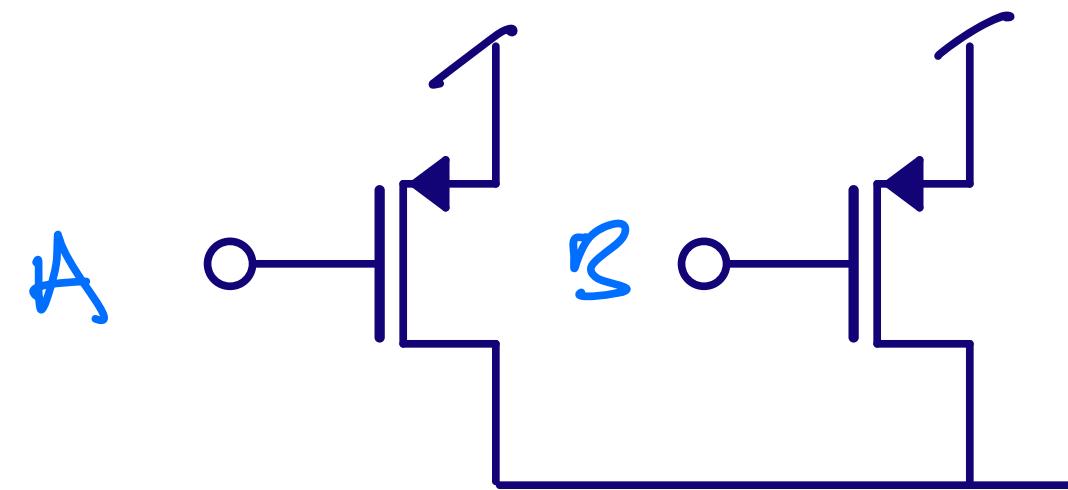
Pull-up series

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | Z |
| 1 | 0 | Z |
| 1 | 1 | Z |



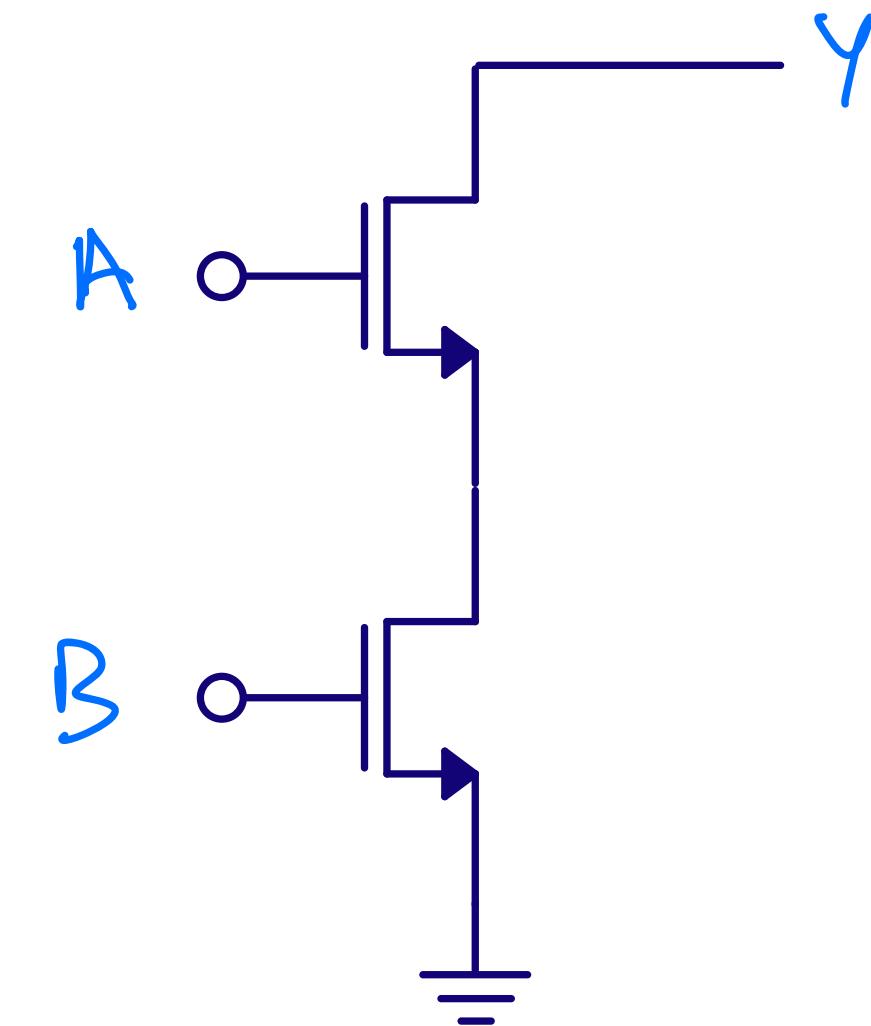
Pull-up parallel

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | Z |



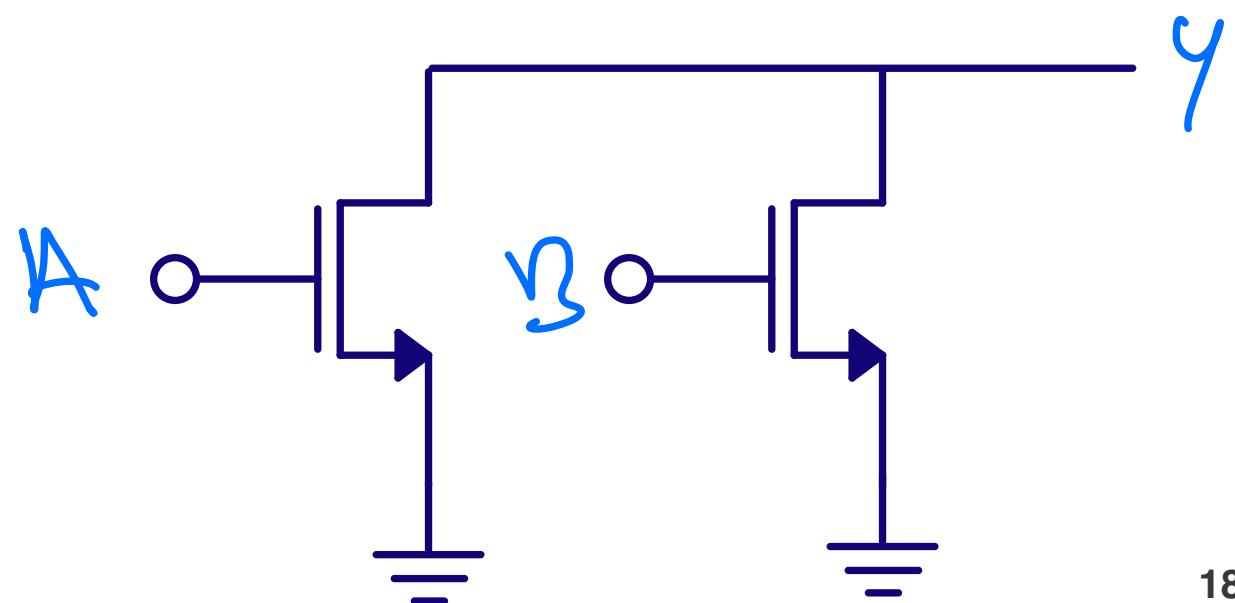
Pull-down series

| A | B | Y |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | Z |
| 1 | 1 | 0 |



Pull-down parallel

| A | B | Y |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



Rules for inverting logic

Pull-up

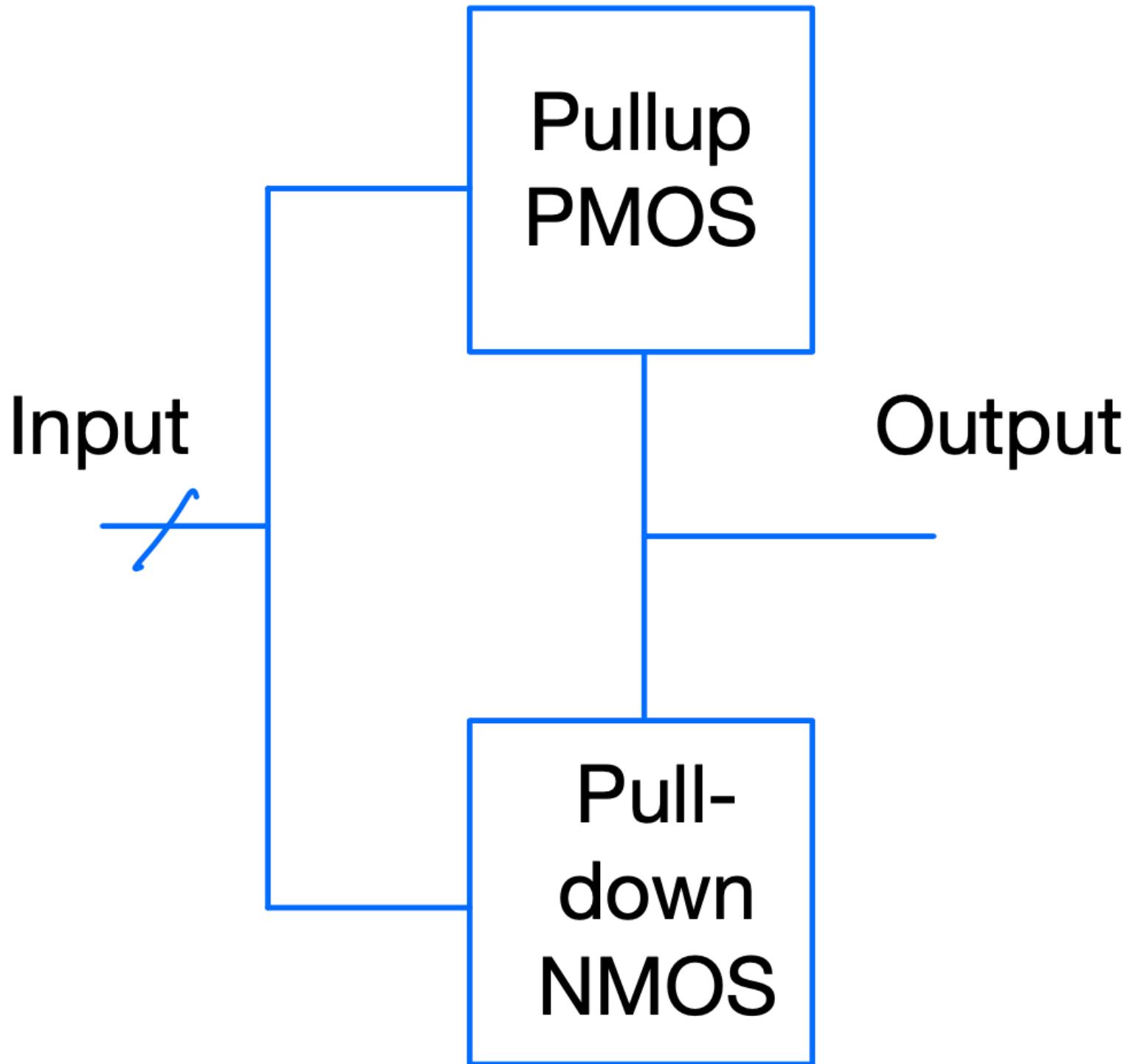
OR \Rightarrow PMOS in series \Rightarrow POS

AND \Rightarrow PMOS in parallel \Rightarrow PAP

Pull-down

OR \Rightarrow NMOS in parallel \Rightarrow NOP

AND \Rightarrow NMOS in series \Rightarrow NAS

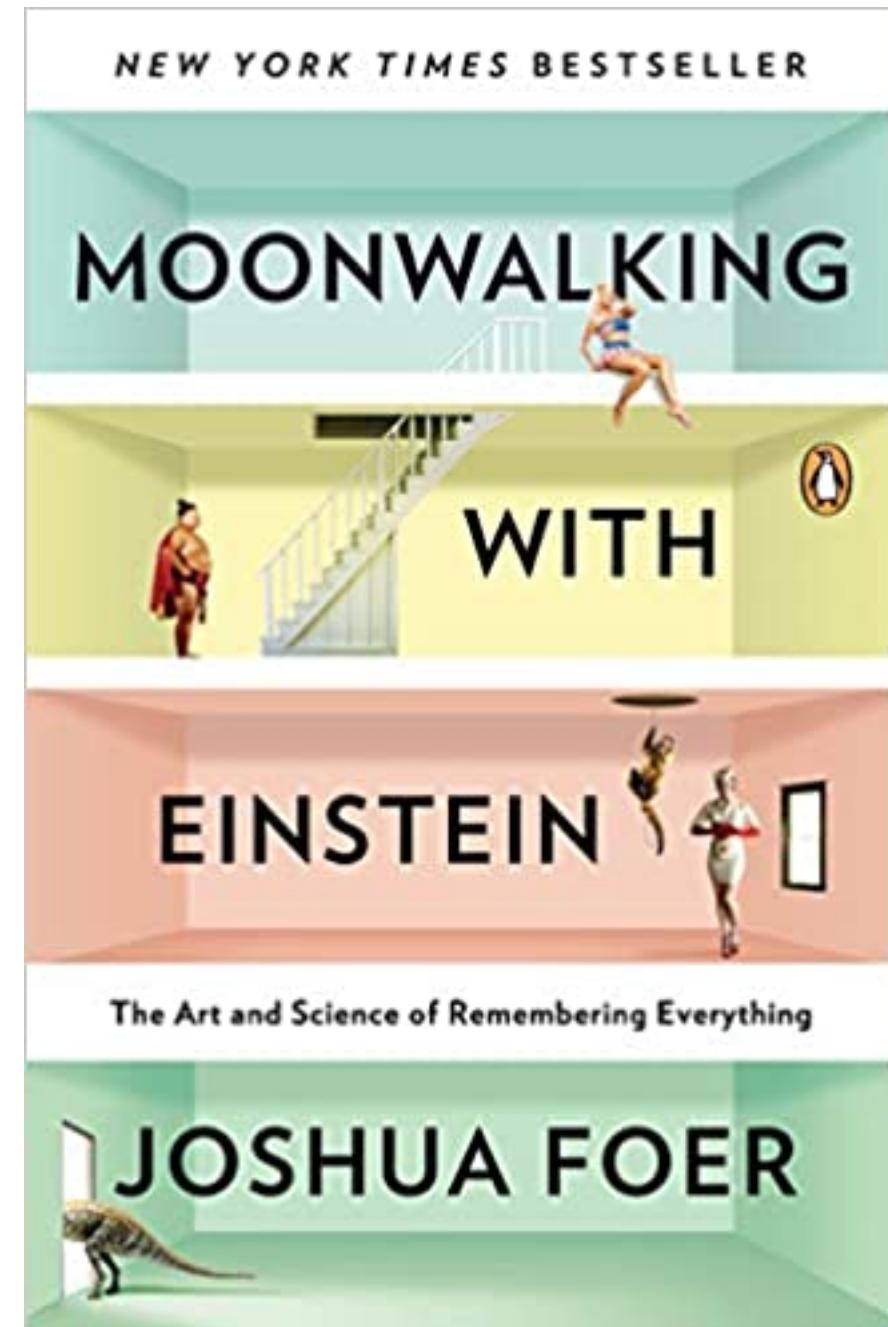


Memnonic

Crazier the better

POS, PAP, NOP, NAS

A Postraumatic Papaya was walking on the Moon. Nope, it was a NASA astronaut.



[Moonwalking with Einstein](#)

$$Y = \overline{AB} = \text{NOT} (\text{A AND B})$$

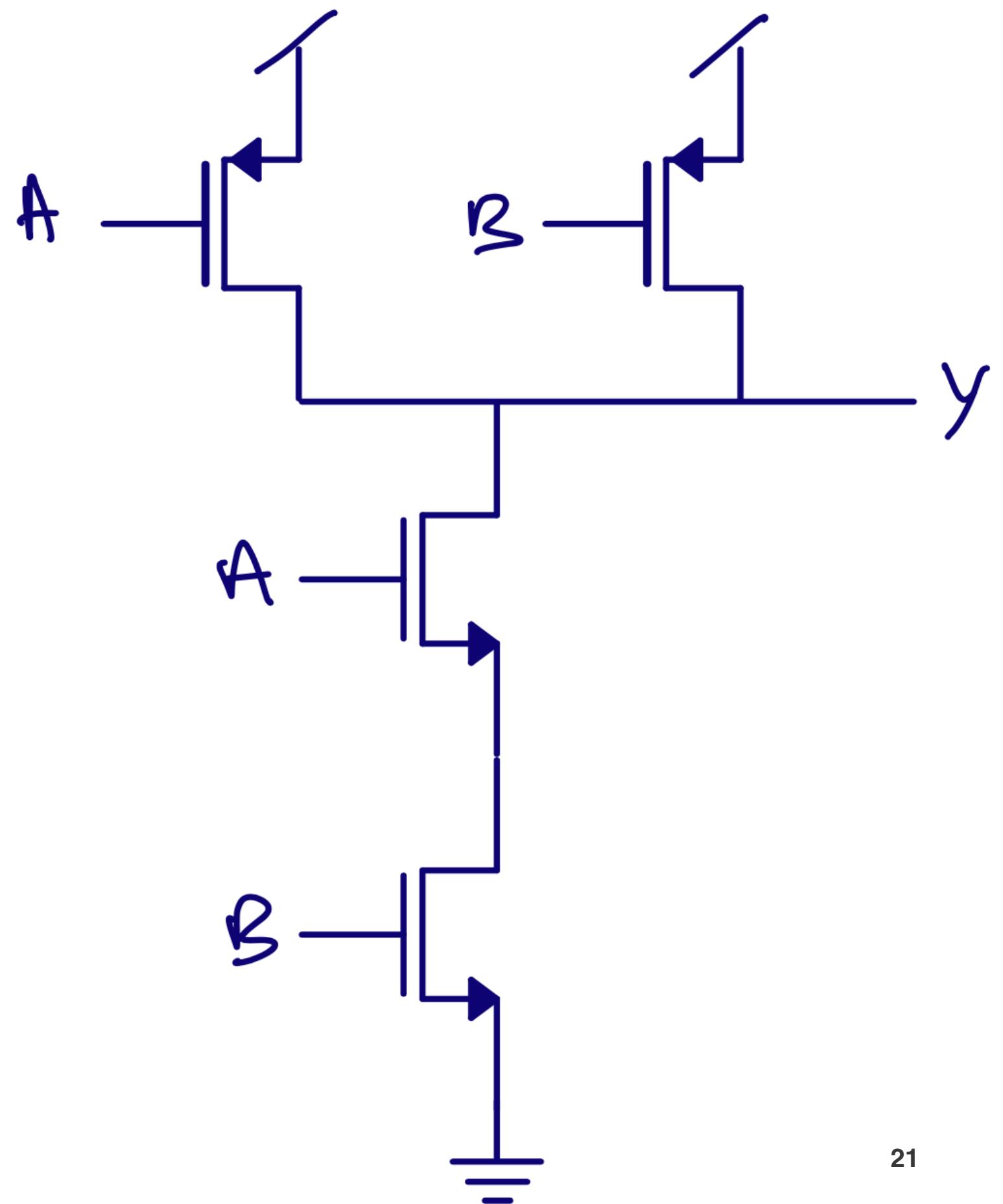
AND

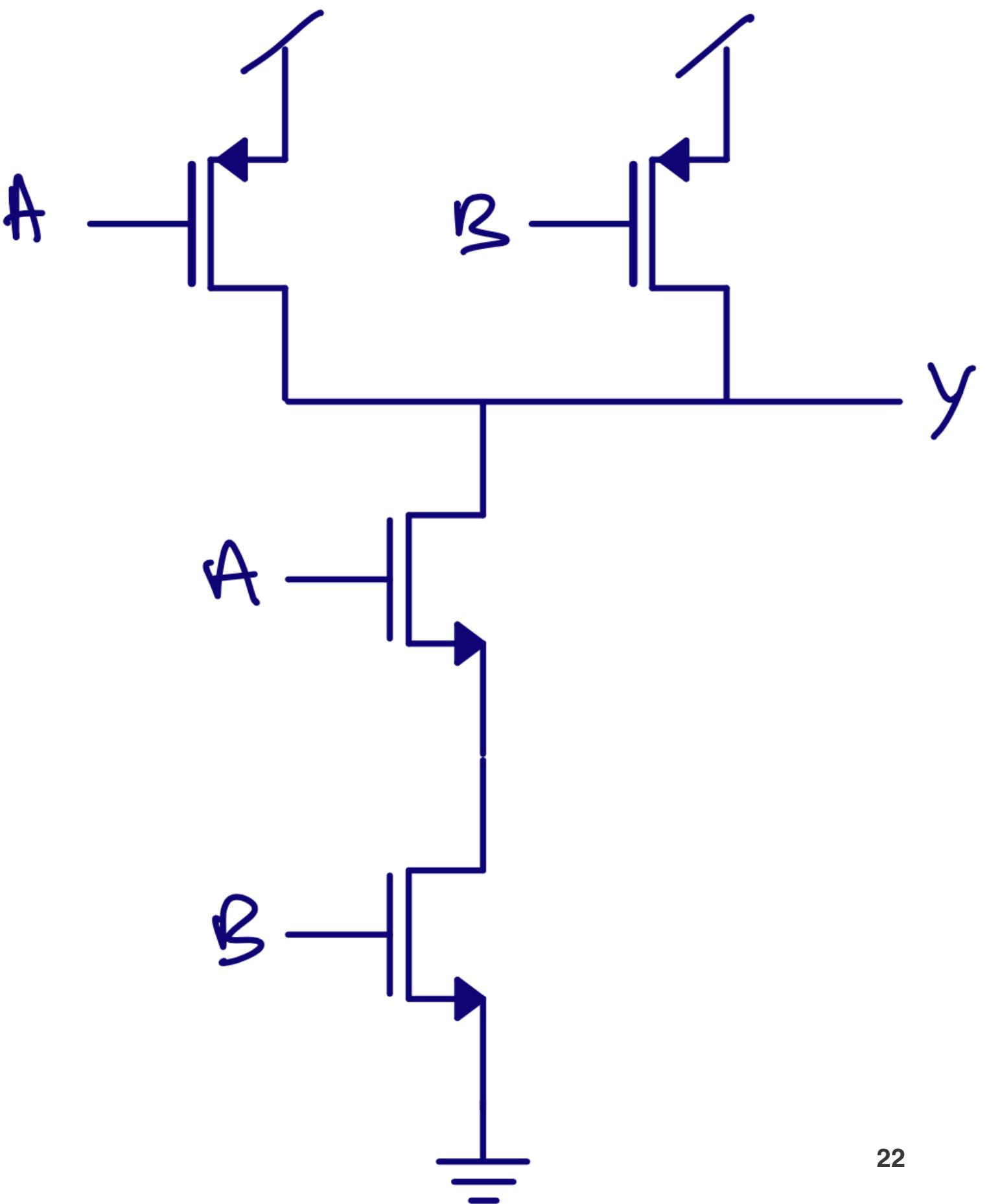
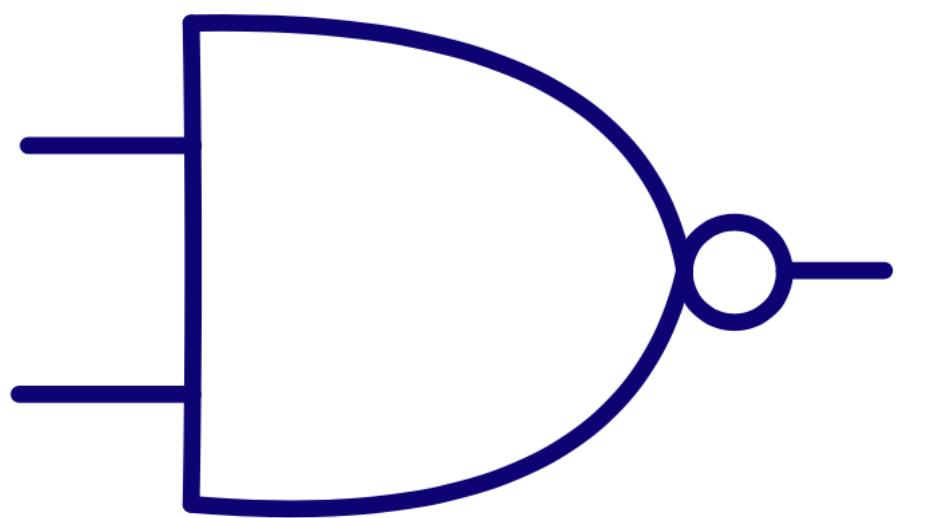
PU \Rightarrow PMOS in parallel

PD \Rightarrow NMOS in series

A Posttraumatic Papaya was walking on the Moon. Nope, it was a NASA astronaut.

| A | B | NOT(A AND B) |
|---|---|--------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |





$$Y = \overline{A + B} = \text{NOT} (A \text{ OR } B)$$

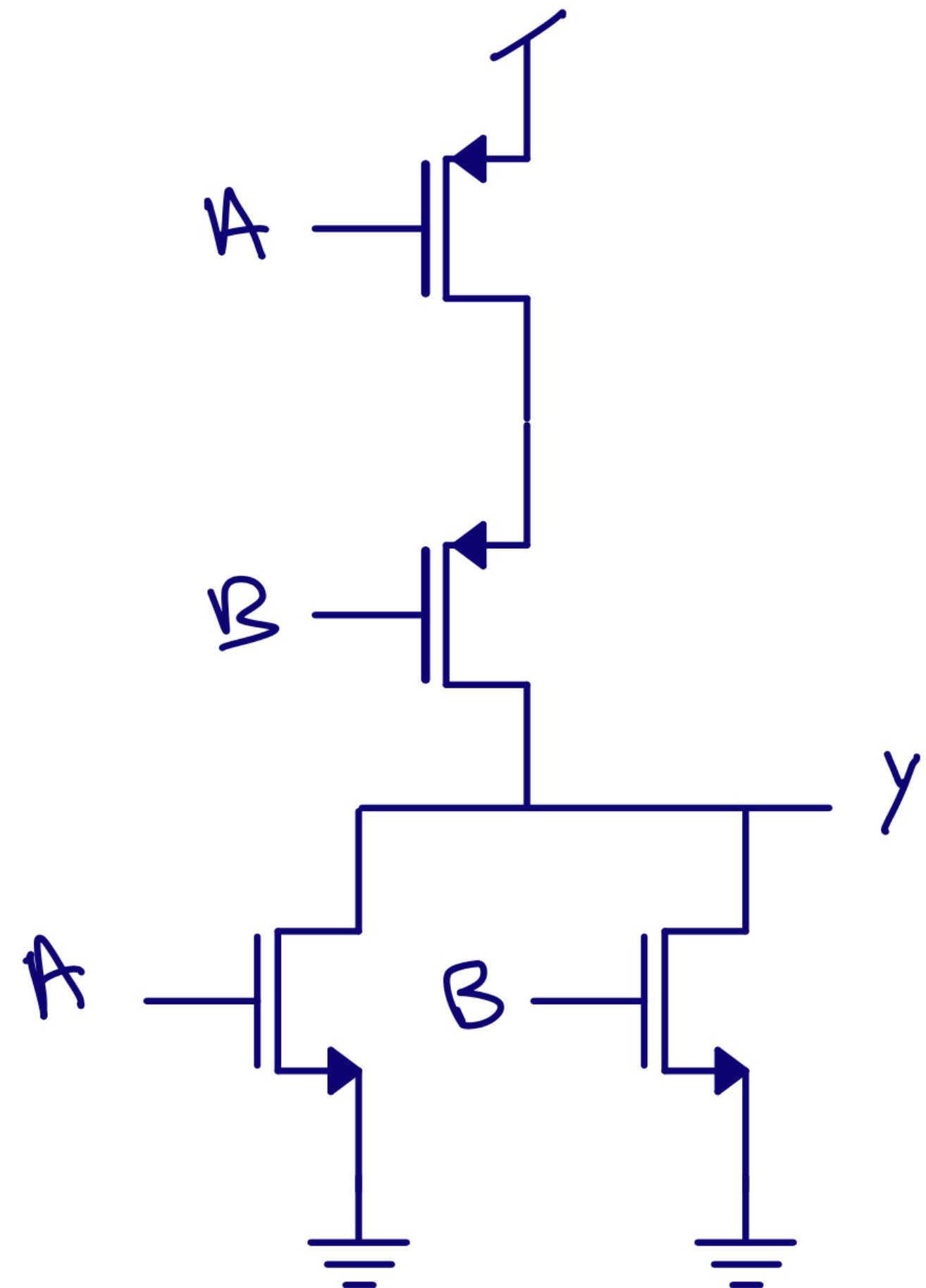
OR

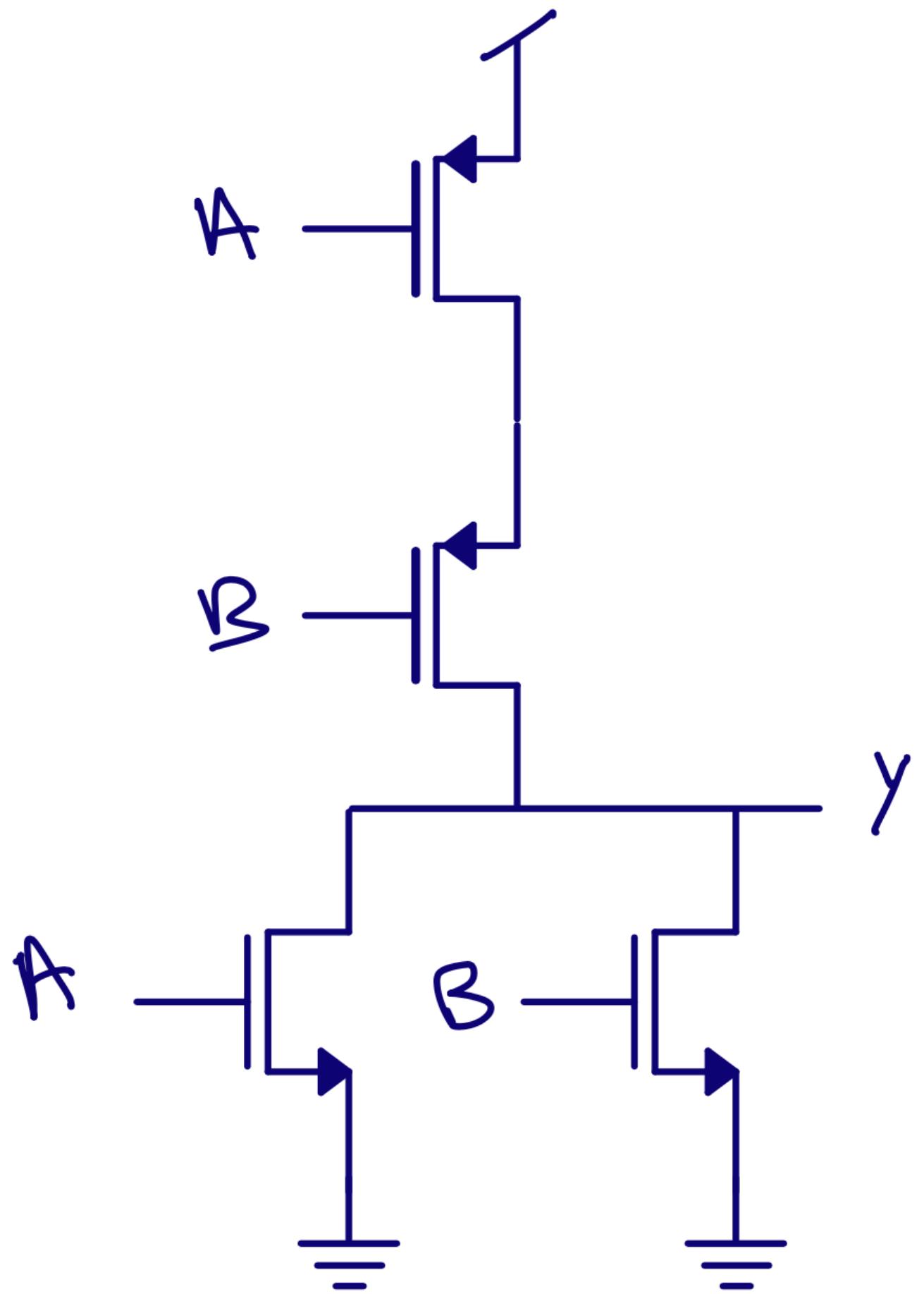
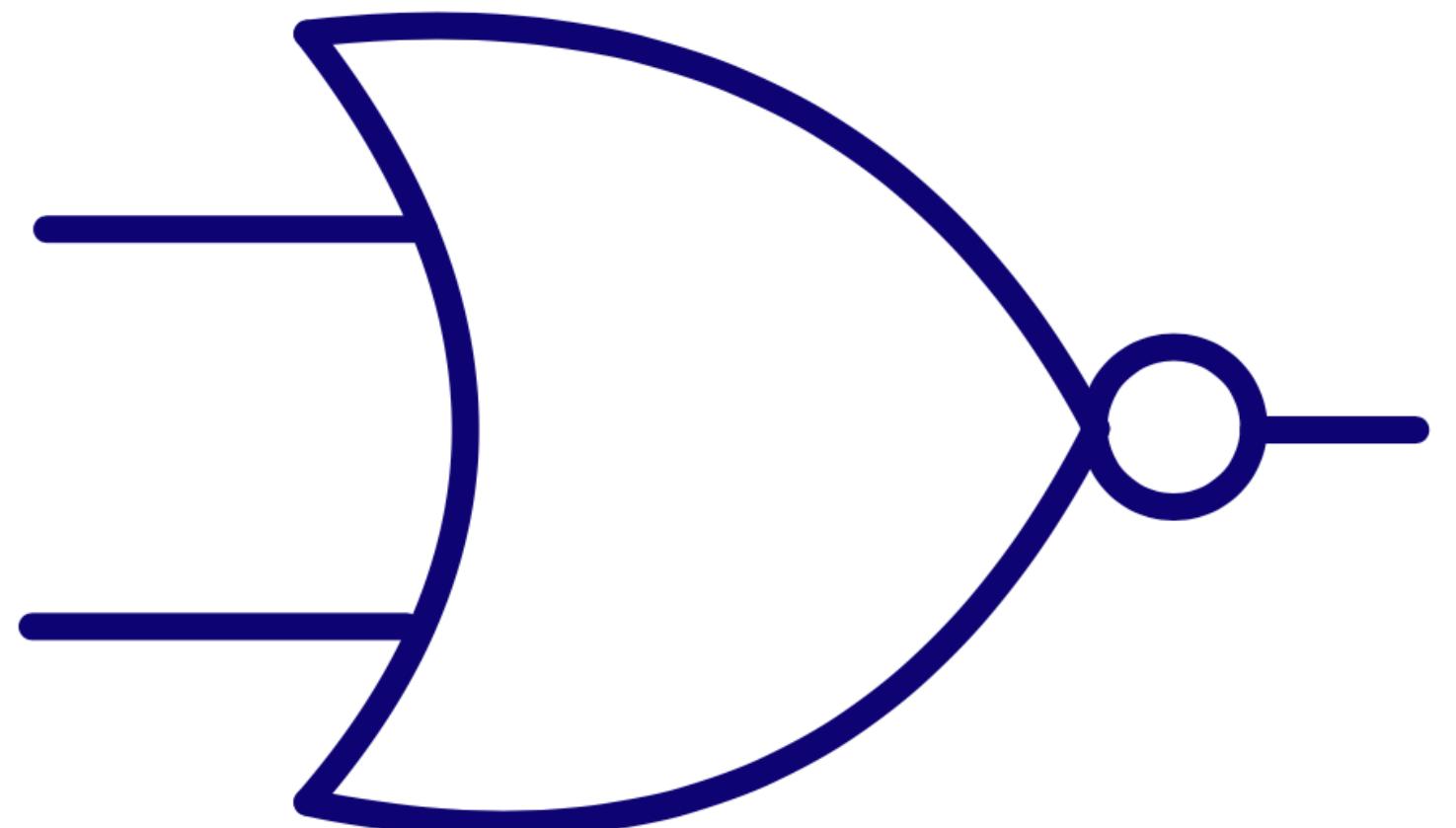
PU \Rightarrow PMOS in series

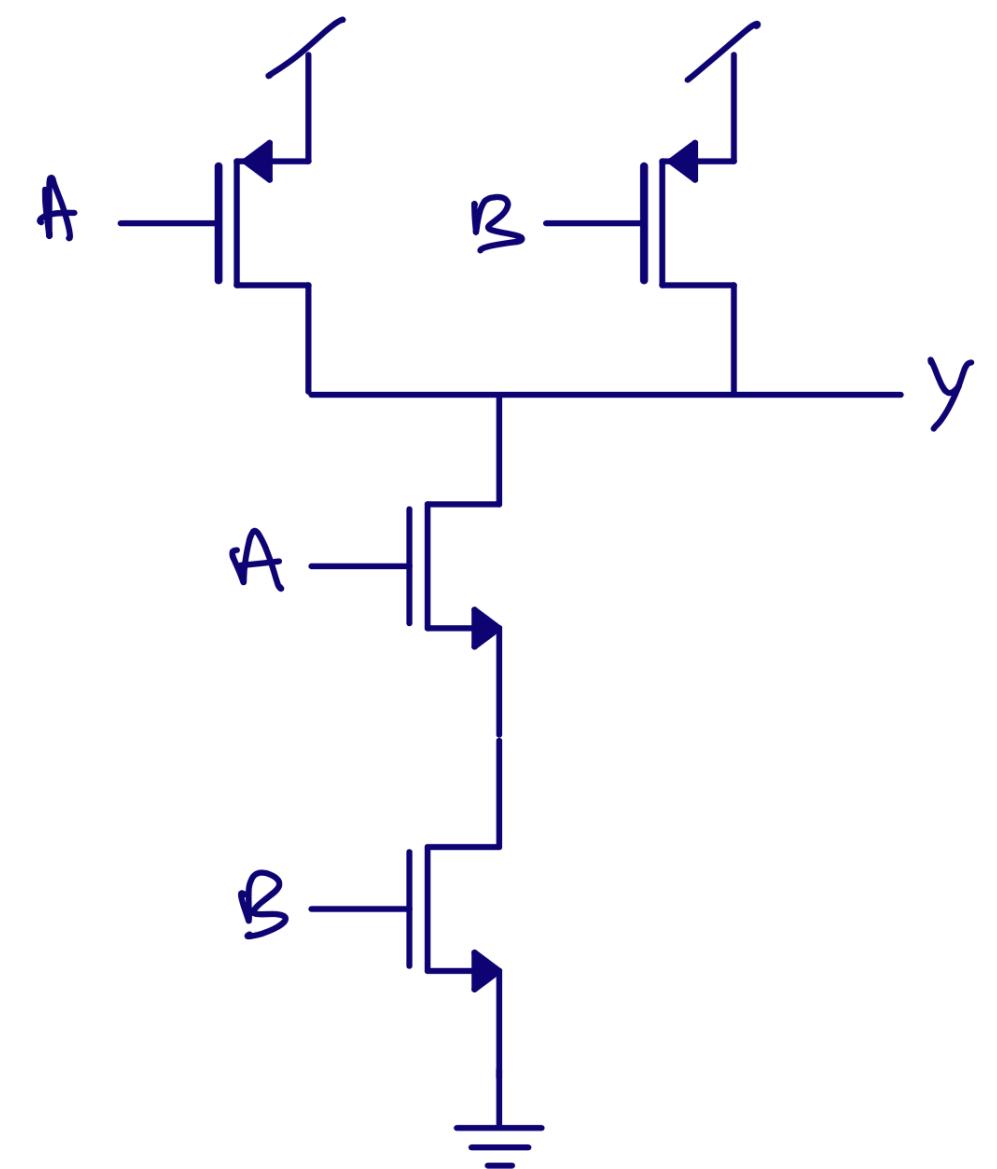
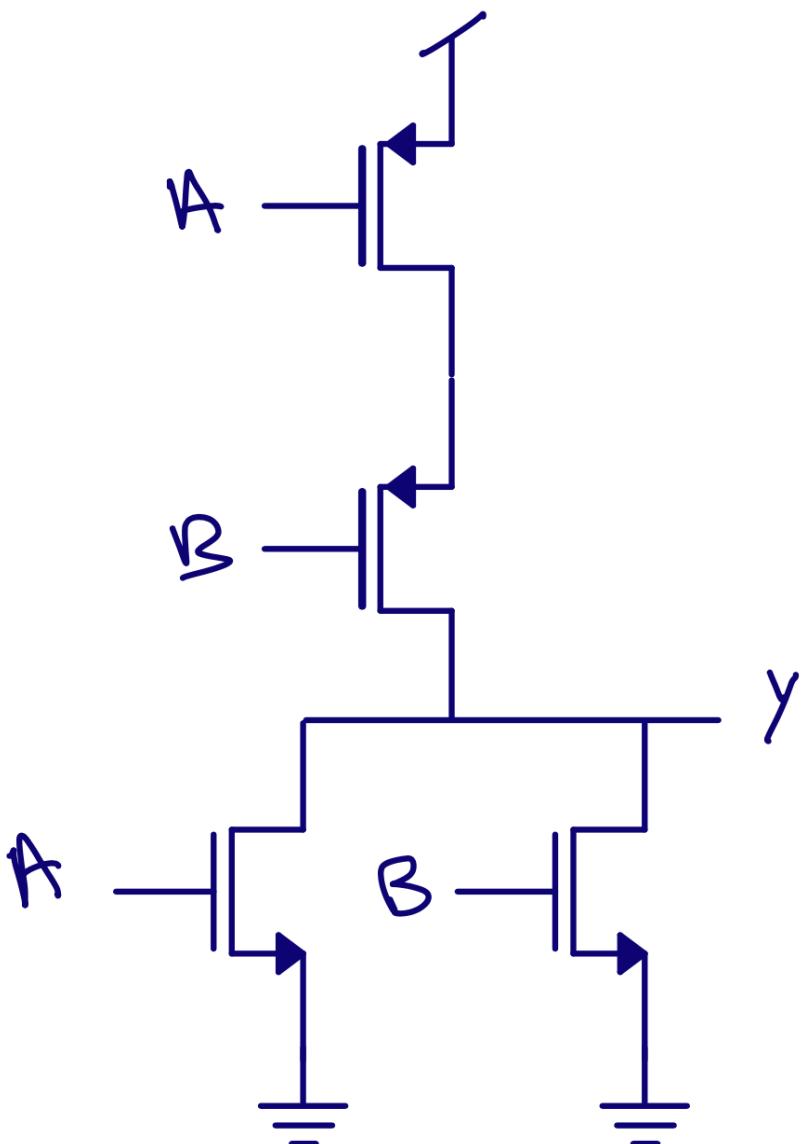
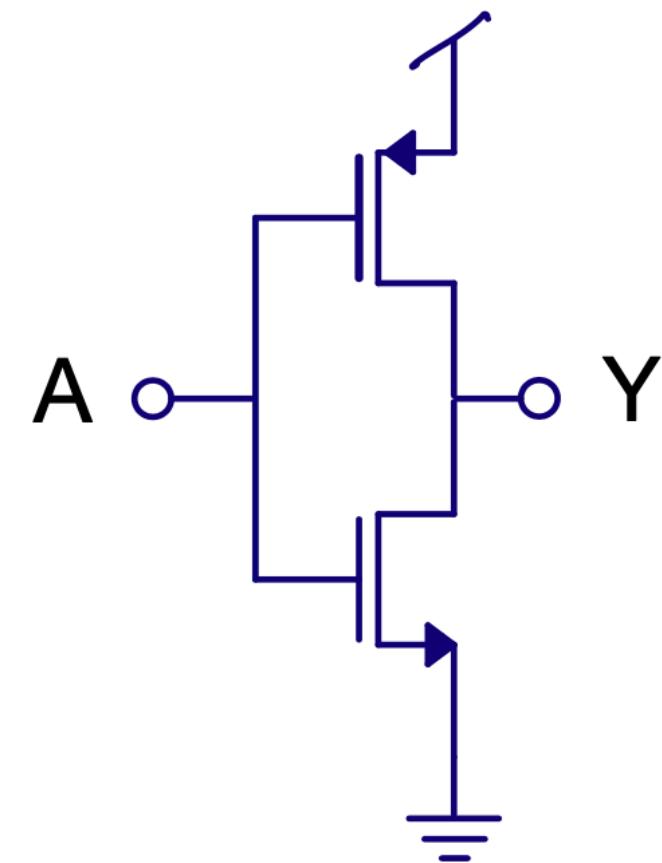
PD \Rightarrow NMOS in parallel

A Posttraumatic Papaya was walking on the Moon. Nope, it was a NASA astronaut.

| A | B | NOT(A OR B) |
|---|---|-------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |







SR-Latch

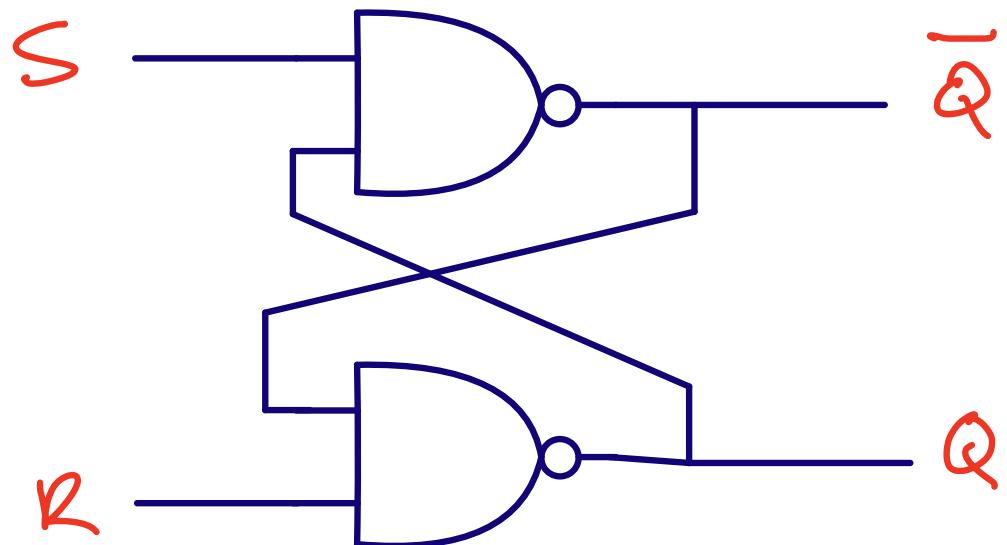
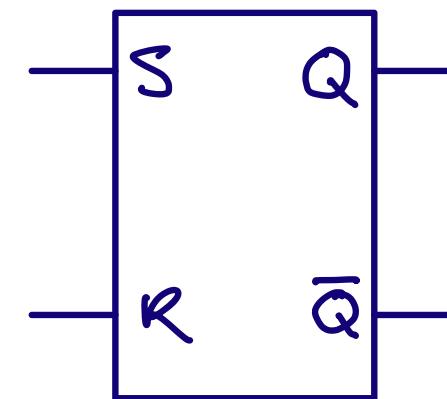
Use boolean expressions to figure out how gates work.

Remember De-Morgan

$$\begin{aligned}\overline{AB} &= \overline{A} + \overline{B} \\ \overline{A + B} &= \overline{A} \cdot \overline{B}\end{aligned}$$

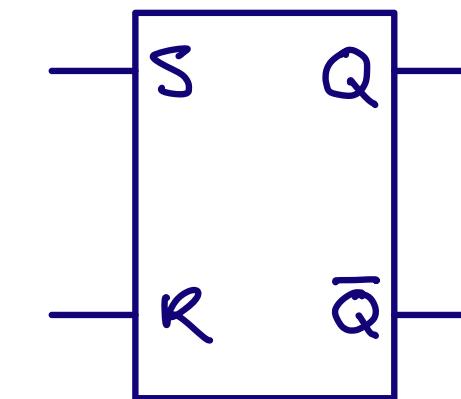
$$Q = \overline{R\overline{Q}} = \overline{R} + \overline{\overline{Q}} = \overline{R} + Q$$

$$\overline{Q} = \overline{S\overline{Q}} = \overline{S} + \overline{\overline{Q}} = \overline{S} + \overline{Q}$$

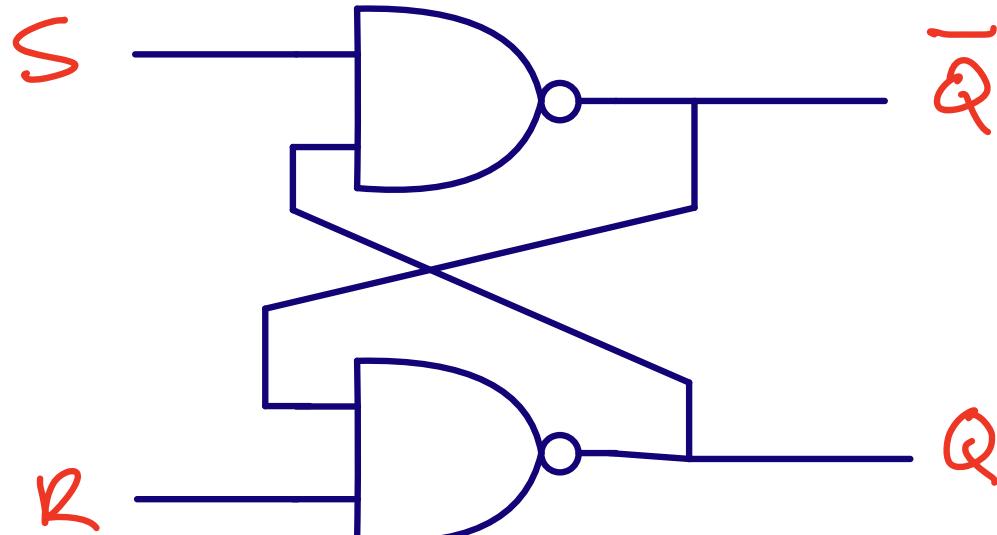


SR-Latch

$$Q = \overline{R} + Q, \overline{Q} = \overline{S} + \overline{Q}$$

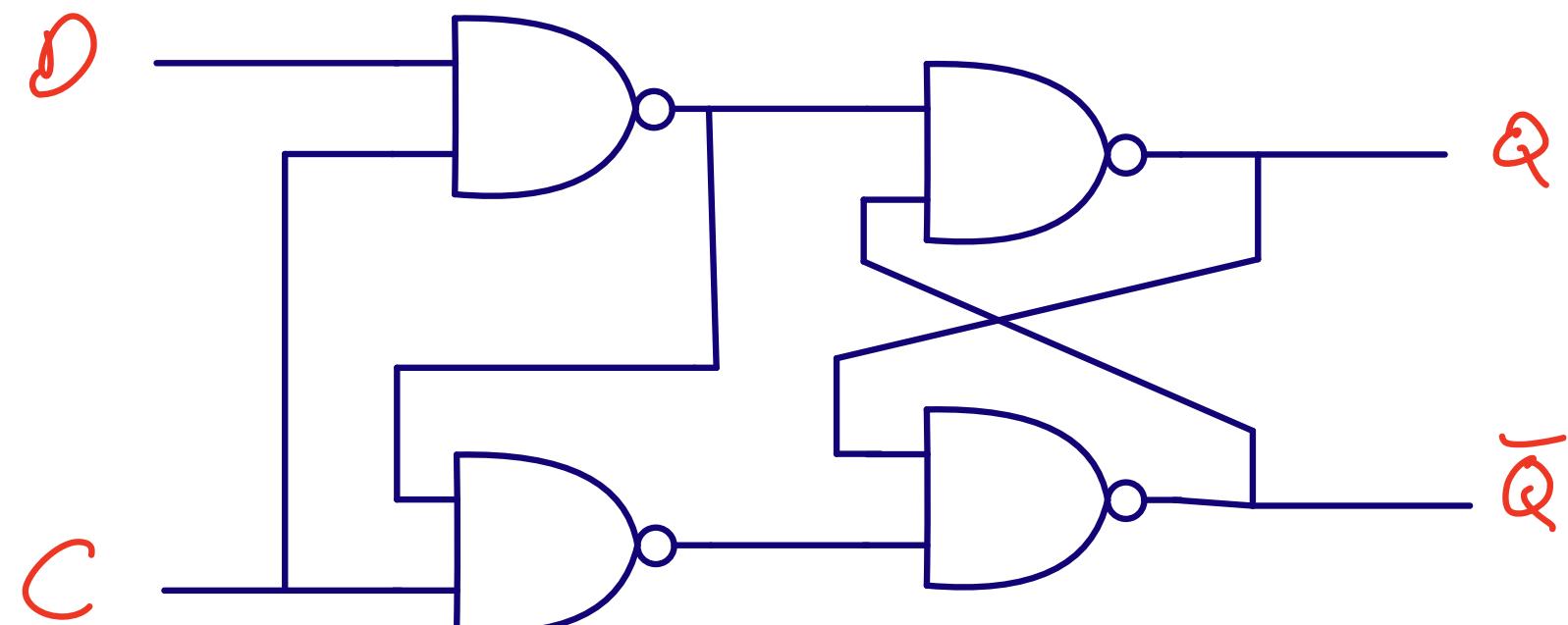
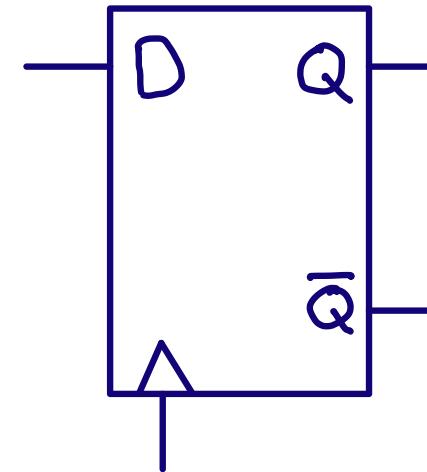


| S | R | Q | $\sim Q$ |
|---|---|---|----------|
| 0 | 0 | X | X |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | Q | $\sim Q$ |



D-Latch (16 transistors)

| C | D | Q | $\sim Q$ |
|---|---|---|----------|
| 0 | X | Q | $\sim Q$ |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |



Digital can be synthesized in
conductive peanut butter

Barrie Gilbert?

Other logic cells

What about $Y = AB$ and $Y = A + B$?

$$Y = AB = \overline{\overline{AB}}$$

$Y = A \text{ AND } B = \text{NOT(NOT(} A \text{ AND } B \text{))}$



$$Y = A + B = \overline{\overline{A+B}}$$

$Y = A \text{ OR } B = \text{NOT(NOT(} A \text{ OR } B \text{))}$

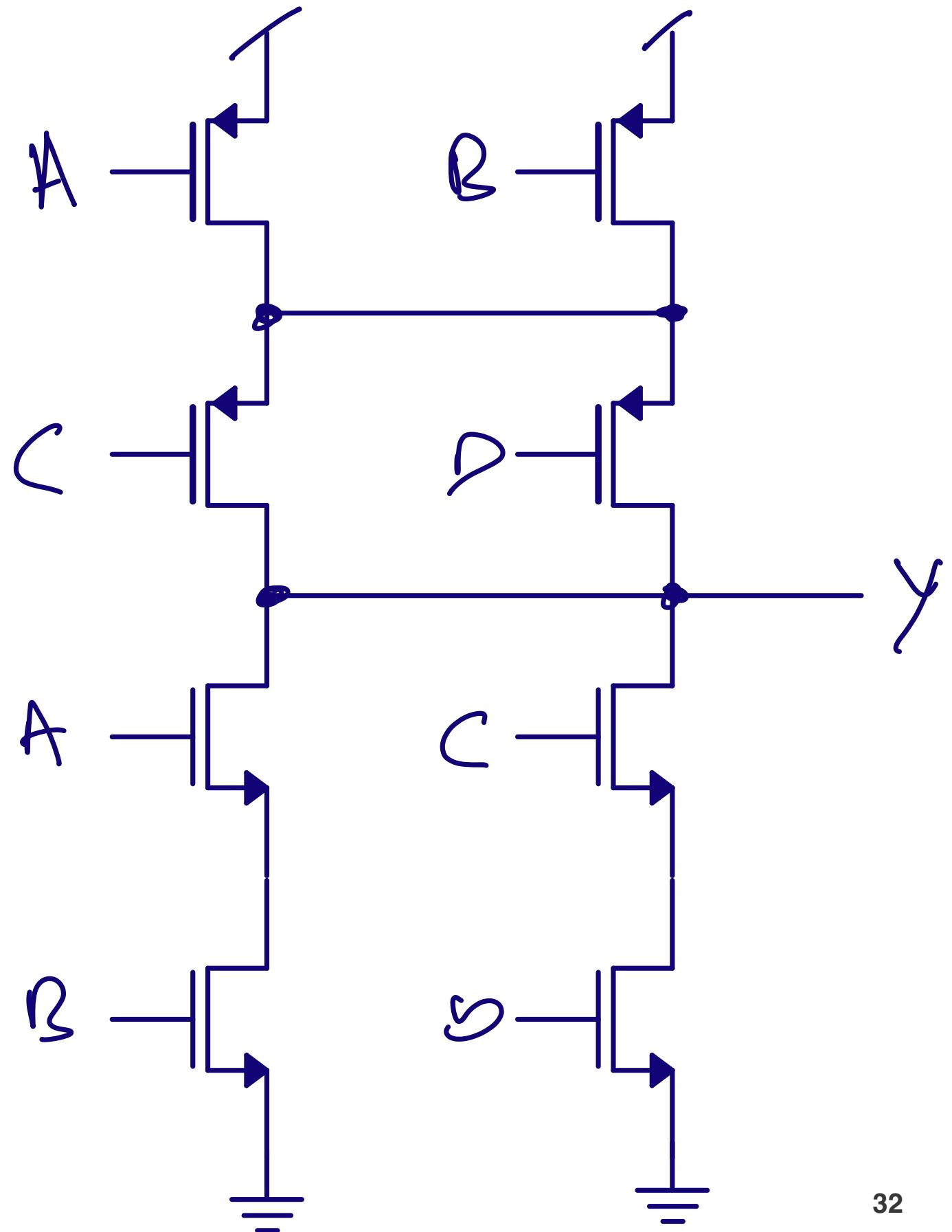


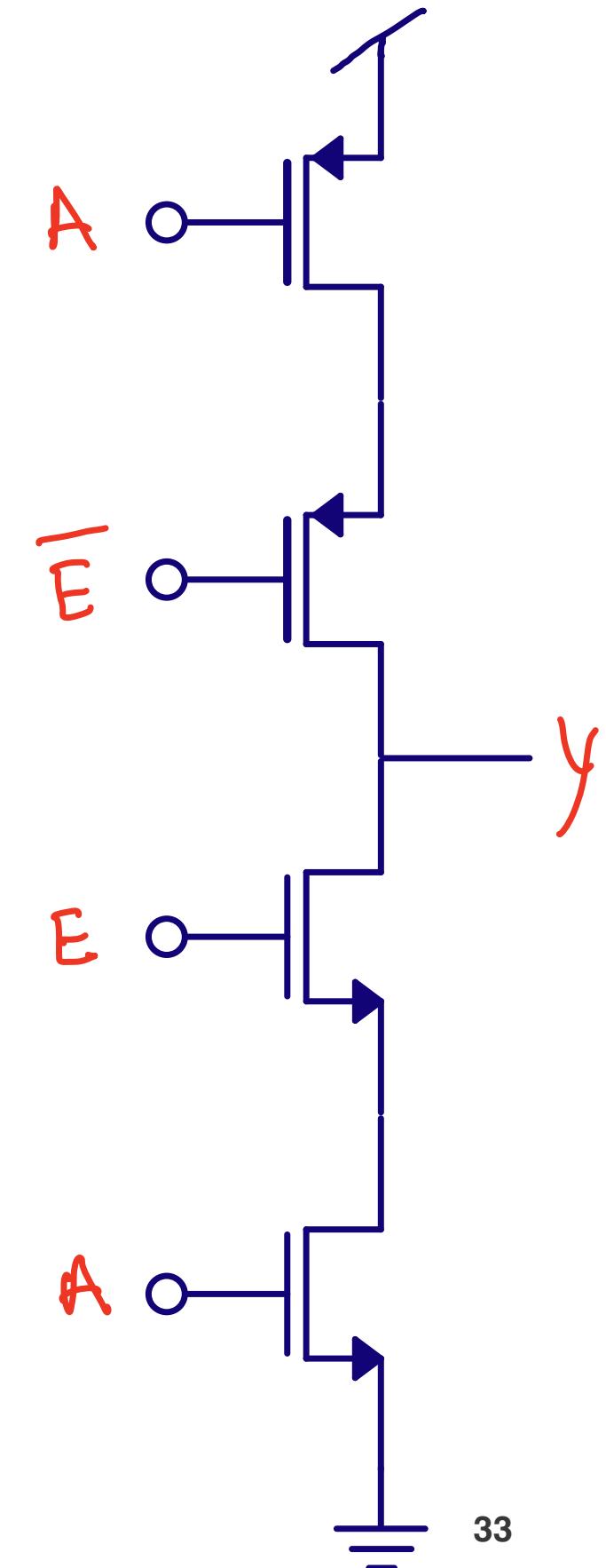
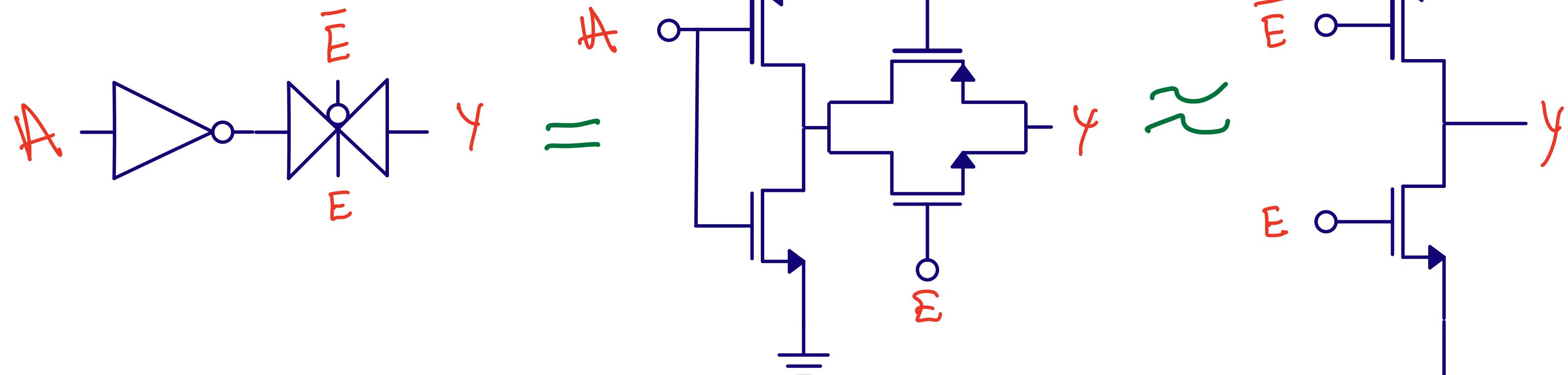
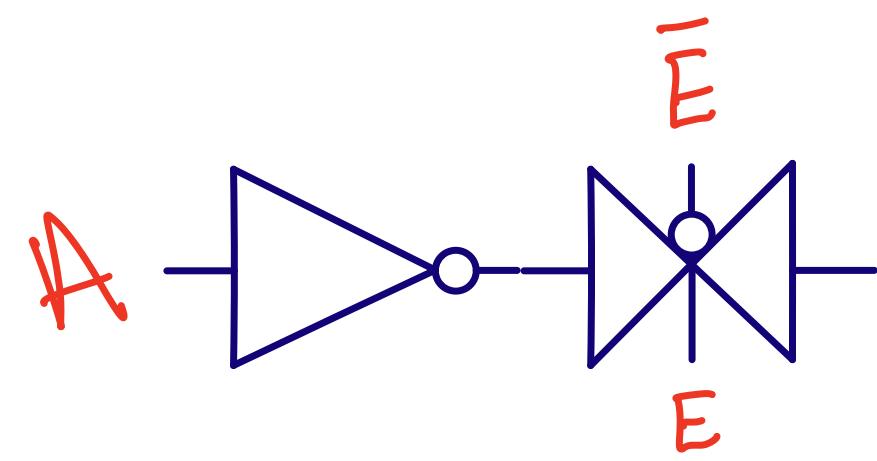
AOI22: and or invert

$$Y = \text{NOT}(\mathbf{A} \text{ AND } \mathbf{B} \text{ OR } \mathbf{C} \text{ AND } \mathbf{D})$$

$$Y = \overline{\mathbf{AB} + \mathbf{CD}}$$

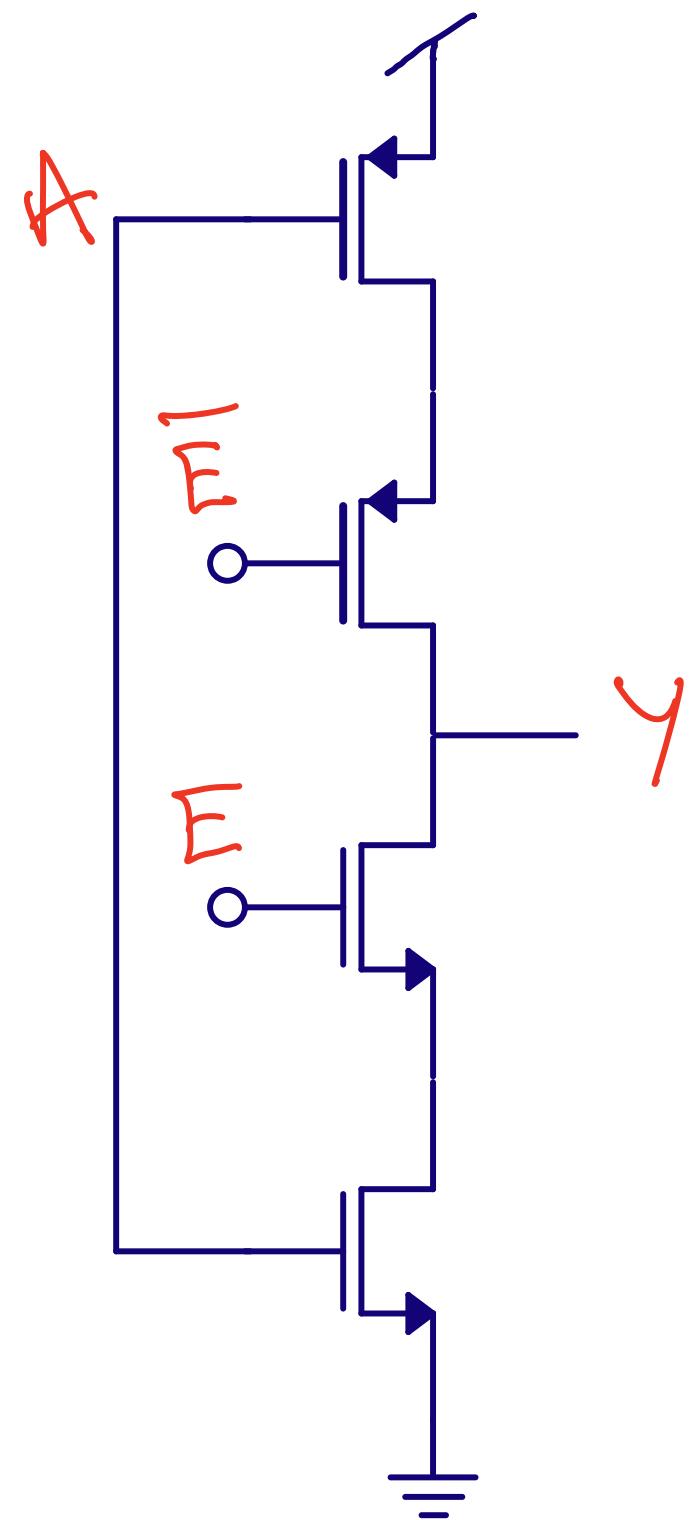
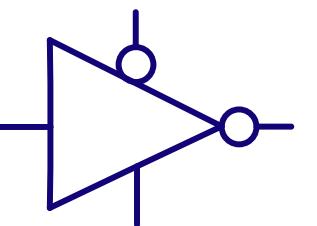
A Posttraumatic Papaya was walking on
the Moon. Nope, it was a NASA
astronaut.





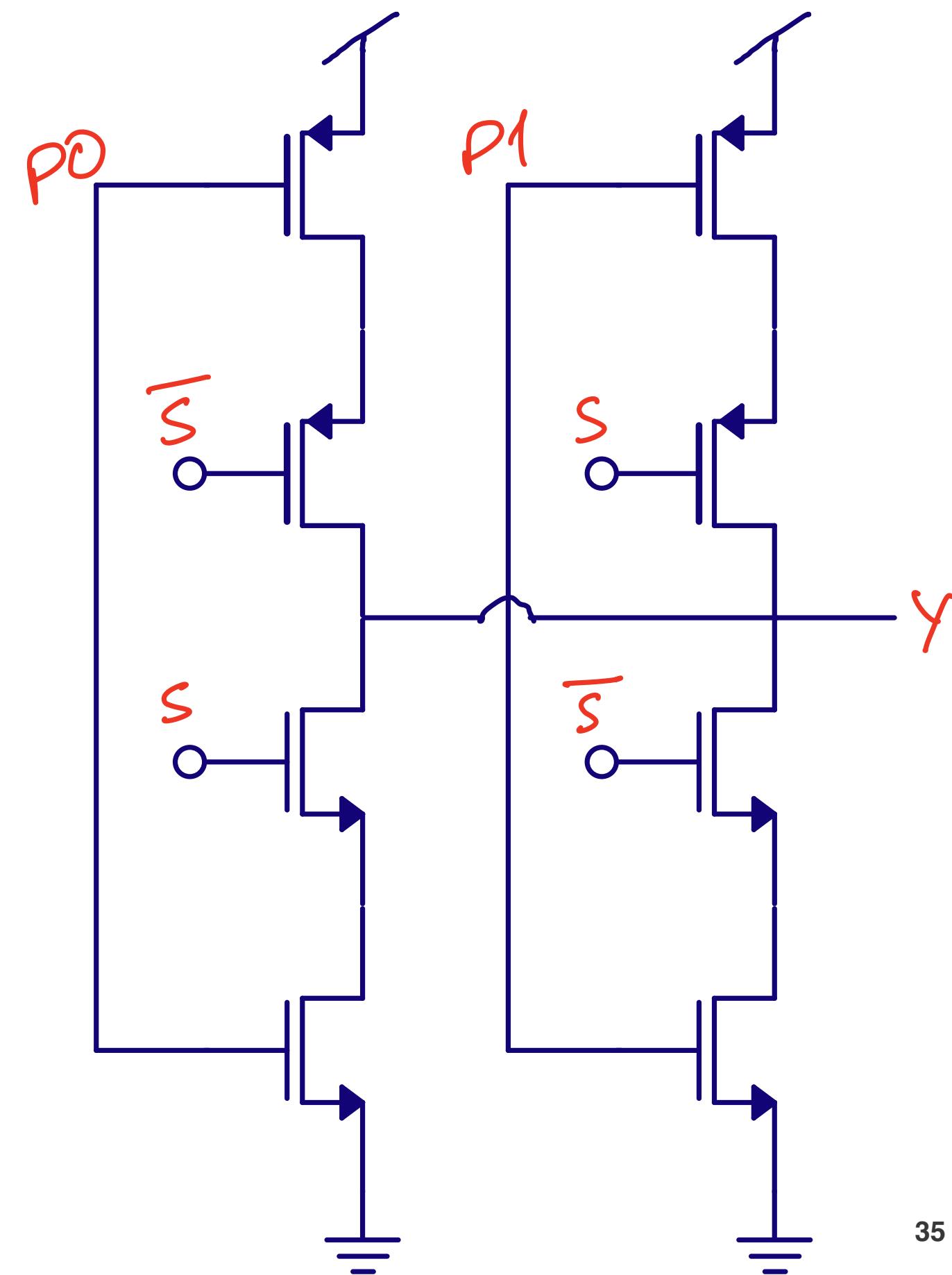
Tristate inverter

| E | A | Y |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

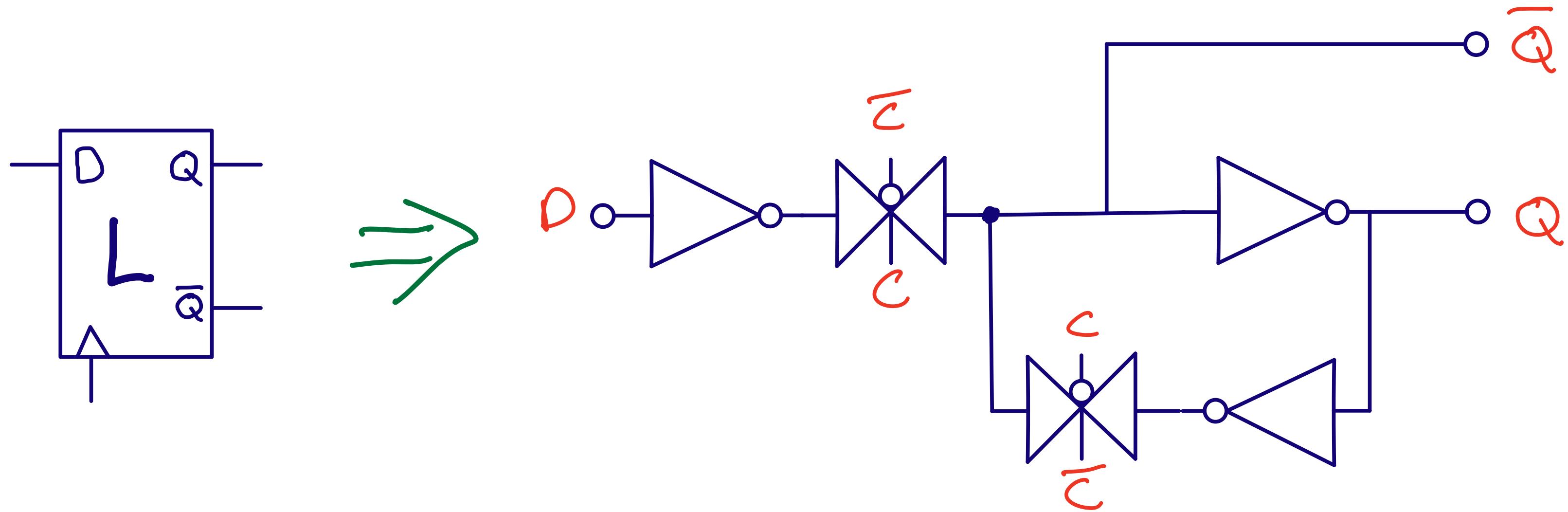


Mux

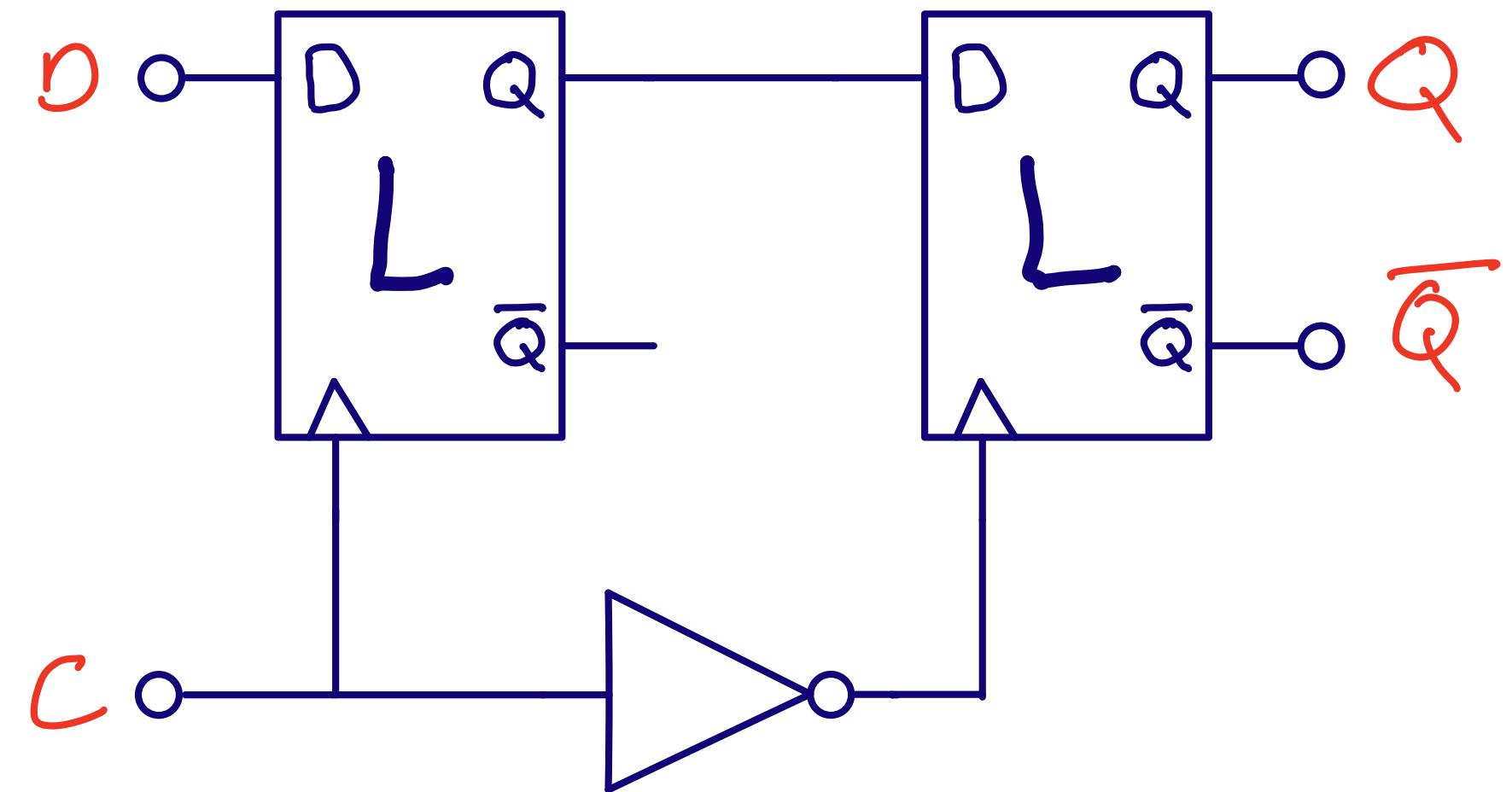
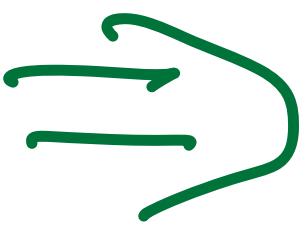
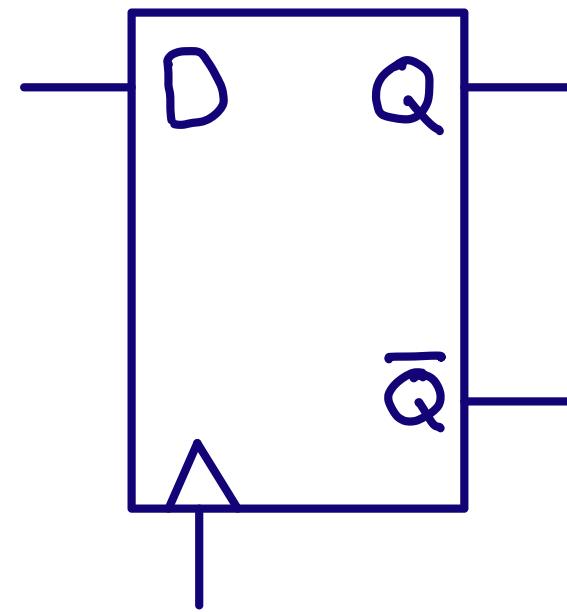
| S | Y |
|---|---------|
| 0 | NOT(P1) |
| 0 | NOT(P1) |
| 1 | NOT(P0) |
| 1 | NOT(P0) |

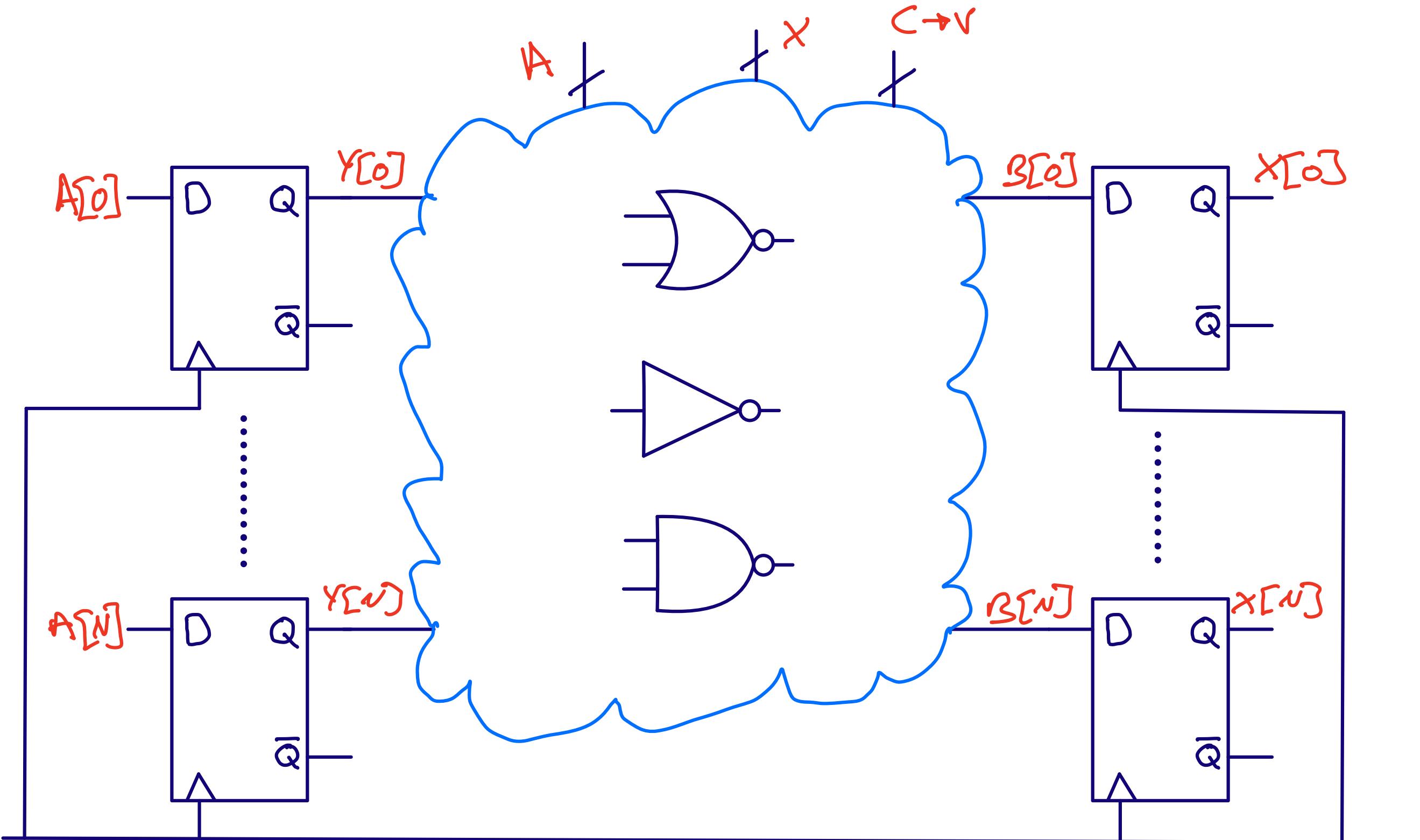


D-Latch (12 transistors)



D-Flip Flop (< 26 transistors)





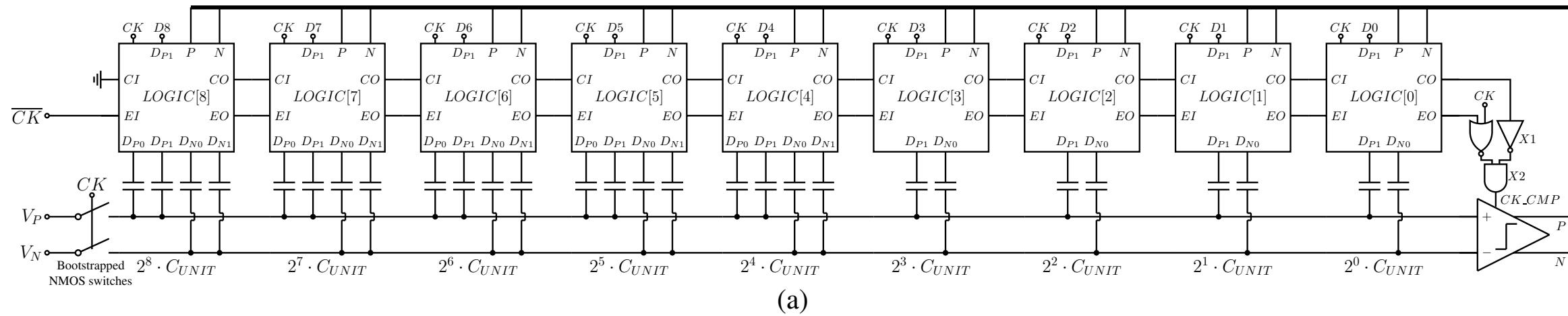
Carsten Wulff 2021 `always_ff`

`always_comb`

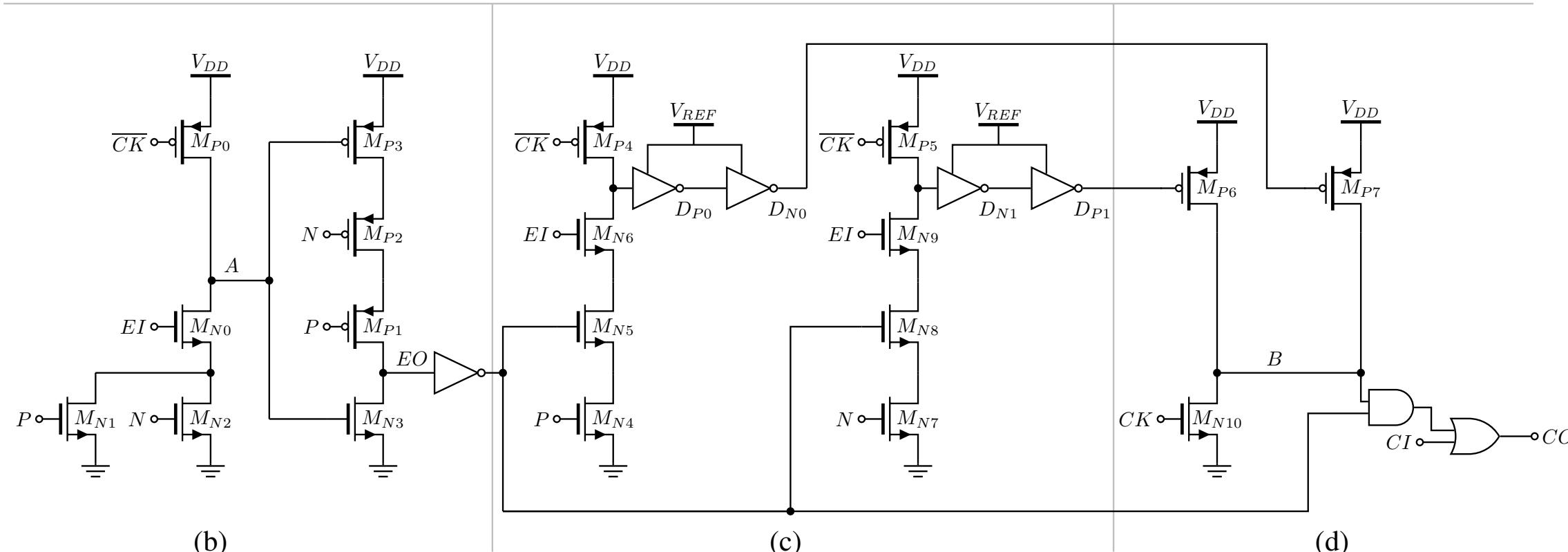
`always_ff`

There are other types of logic

- True single phase clock (TSPC) logic
 - Pass transistor logic
 - Transmission gate logic
 - Differential logic
 - Dynamic logic
- Consider other types of logic "rule breaking", so you should know why you need it.



(a)



(b)

(c)

(d)

Dynamic logic => A Compiled 9-bit 20-MS/s 3.5-fJ/conv.step SAR ADC in 28-nm FDSOI for Bluetooth Low Energy Receivers

Zen of electronics design ¹

Beautiful is better than ugly.

Although practicality beats purity.

Explicit is better than implicit.

In the face of ambiguity, refuse the temptation to guess.

Simple is better than complex.

Now is better than never.

Complex is better than complicated.

Although never is often better than *right* now.

Readability counts.

If the implementation is hard to explain, it's a bad idea.

Special cases aren't special enough to break the rules.

If the implementation is easy to explain, it may be a good idea.

¹ Zen of Python

Sesame

Sesame is a Python3 package for solving the drift diffusion Poisson equations for multi-dimensional systems using finite differences.

Install instructions

Semiconductor current-flow equations (diffusion and degeneracy),

R.Stratton,

IEEE Transactions on Electron Devices

<https://ieeexplore.ieee.org/document/1477063>

Semiconductor Current-Flow Equations (Diffusion and Degeneracy)

ROBERT STRATTON

The correct form for the current-flow equation in semiconductors in the presence of density and temperature gradients, as well as electric fields, is derived from a perturbation solution of Boltzmann's equation. The conditions under which the various widely used approximate forms of the current-flow equation are valid are clearly discussed. A new term that occurs if the relaxation time depends on position is derived, and is shown to be comparable in magnitude to the other terms in the current-flow equation.

Manuscript received March 15, 1972; revised July 20, 1972.
The author is with Texas Instruments, Inc., Dallas, Tex. 75222

I. INTRODUCTION

THE ANALYSIS of the electrical characteristics of semiconductor devices invariably involves the electron current-flow equation, i.e.,

$$j = j_F + j_D \quad (1)$$

where j_F is the conduction current due to the electric field F and j_D is the diffusion current, as one of a set of simultaneous equations that must be solved for a given

**Grid**

Each axis of the grid is a concatenation of sets of evenly spaced nodes. Edit the form with (x1, x2, number of nodes), (x2, x3 number of nodes),...

Grid x-axis ,100,(3e-4,4e-4,100 cm

Grid y-axis (0,1e-5,5) cm

Materials

Material 1

New

Save

Remove

Save a material before adding a new one.

Location $x < 2e-4$

Tip: Define the region for $y < 1.5 \mu\text{m}$ or $y > 2.5 \mu\text{m}$ with $(y < 1.5e-6) | (y > 2.5e-6)$. Use the bitwise operators | for 'or', and & for 'and'.

| | Value | Unit |
|---------|---------|------------------|
| N_D | 0.0 | cm^{-3} |
| N_A | 1e+15 | cm^{-3} |
| Nc | 3.2e+19 | cm^{-3} |
| Nv | 1.8e+19 | cm^{-3} |
| Eg | 1.1 | eV |
| epsilon | 11.8 | NA |

Illumination

monochromatic 1 sun

Wavelength [nm]

Power [W cm^{-2}]

Absorption

User defined from file

alpha [cm^{-1}]

absorption file

Browse...

Manual Generation rate

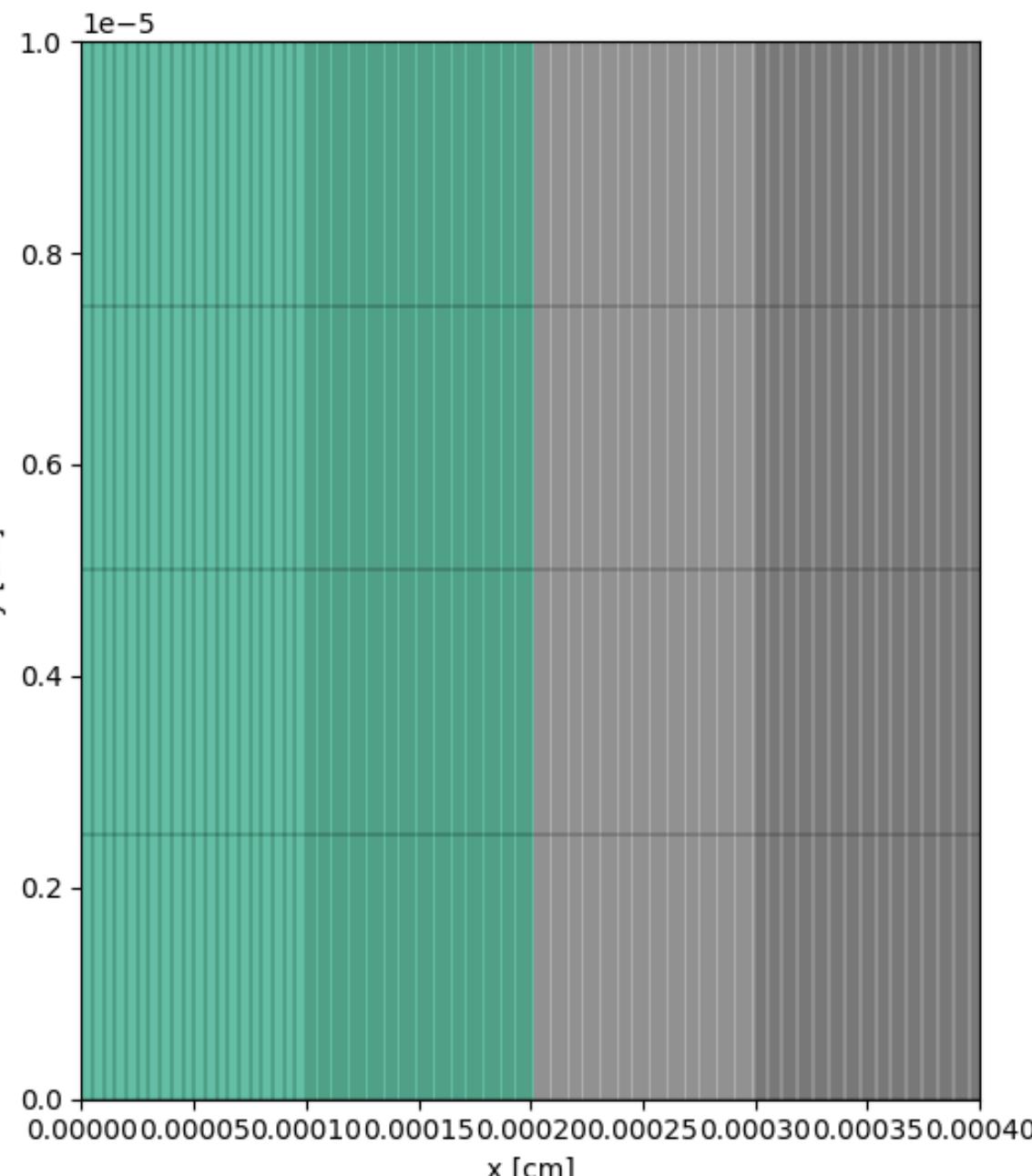
Use manual generation

Provide a number for uniform illumination, or a space-dependent function, or simply nothing for dark conditions.

A single variable parameter is allowed and will be looped over during the simulation.

Expression [$\text{cm}^{-3}\text{s}^{-1}$] 0

Paramater name

View system

**Basic settings**Loop over Voltages Generation rates

Loop values [0.3|0.0,-1]

Working directory same/wulff/pndiode/ Output file name pndiode **Boundary conditions**Contact boundary conditions at x=0 Ohmic Schottky Neutral

Electron recombination velocity in x=0 [cm/s] 1e5

Hole recombination velocity in x=0 [cm/s] 1e5

Metal work function [eV] 4.7

Contact boundary conditions at x=L Ohmic Schottky Neutral

Electron recombination velocity in x=L [cm/s] 1e5

Hole recombination velocity in x=L [cm/s] 1e5

Metal work function [eV] 4.7

Transverse boundary conditions Periodic Hardwall**Algorithm settings**

Generation ramp 0

Algorithm precision 1e-6

Maximum steps 100

Mumps library Yes NoIterative solver Yes No

Iterative solver precision 1e-6

Newton homotopy 1

Simulation log

INFO: step 25,error = 0.9999995808573537

INFO: step 26,error = 0.9999988596577988

INFO: step 27,error = 0.999996898179034

INFO: step 28,error = 0.9999915642702679

INFO: step 29,error = 0.999977061282639

INFO: step 30,error = 0.9999376312764978

INFO: step 31,error = 0.9998304427570286

INFO: step 32,error = 0.9995391106703418

INFO: step 33,error = 0.9987476260090542

INFO: step 34,error = 0.9965997108182367

INFO: step 35,error = 0.9907879485052914

INFO: step 36,error = 0.9751877095968543

INFO: step 37,error = 0.9342041943279747

INFO: step 38,error = 0.8325334222710672

INFO: step 39,error = 0.614963842698804

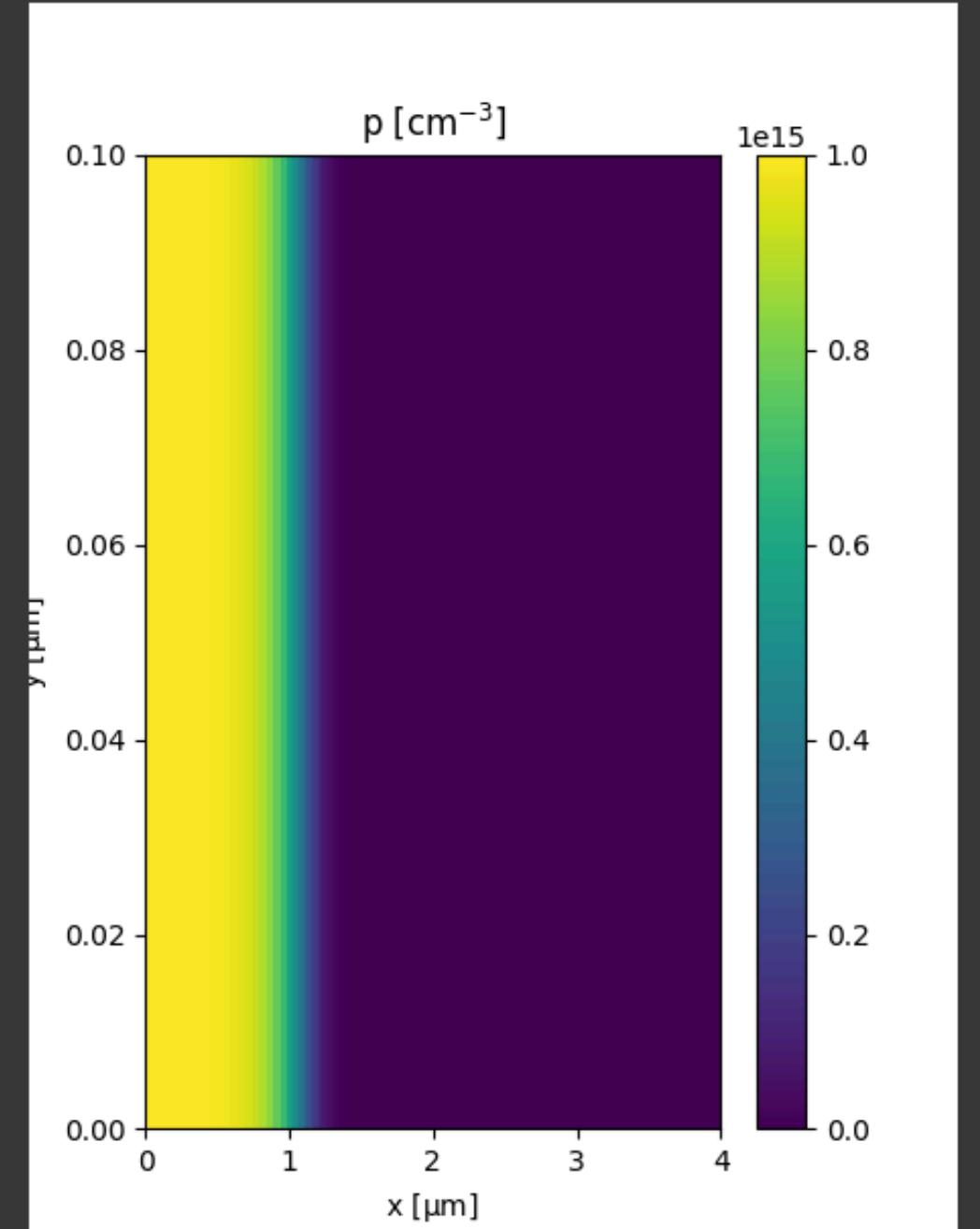
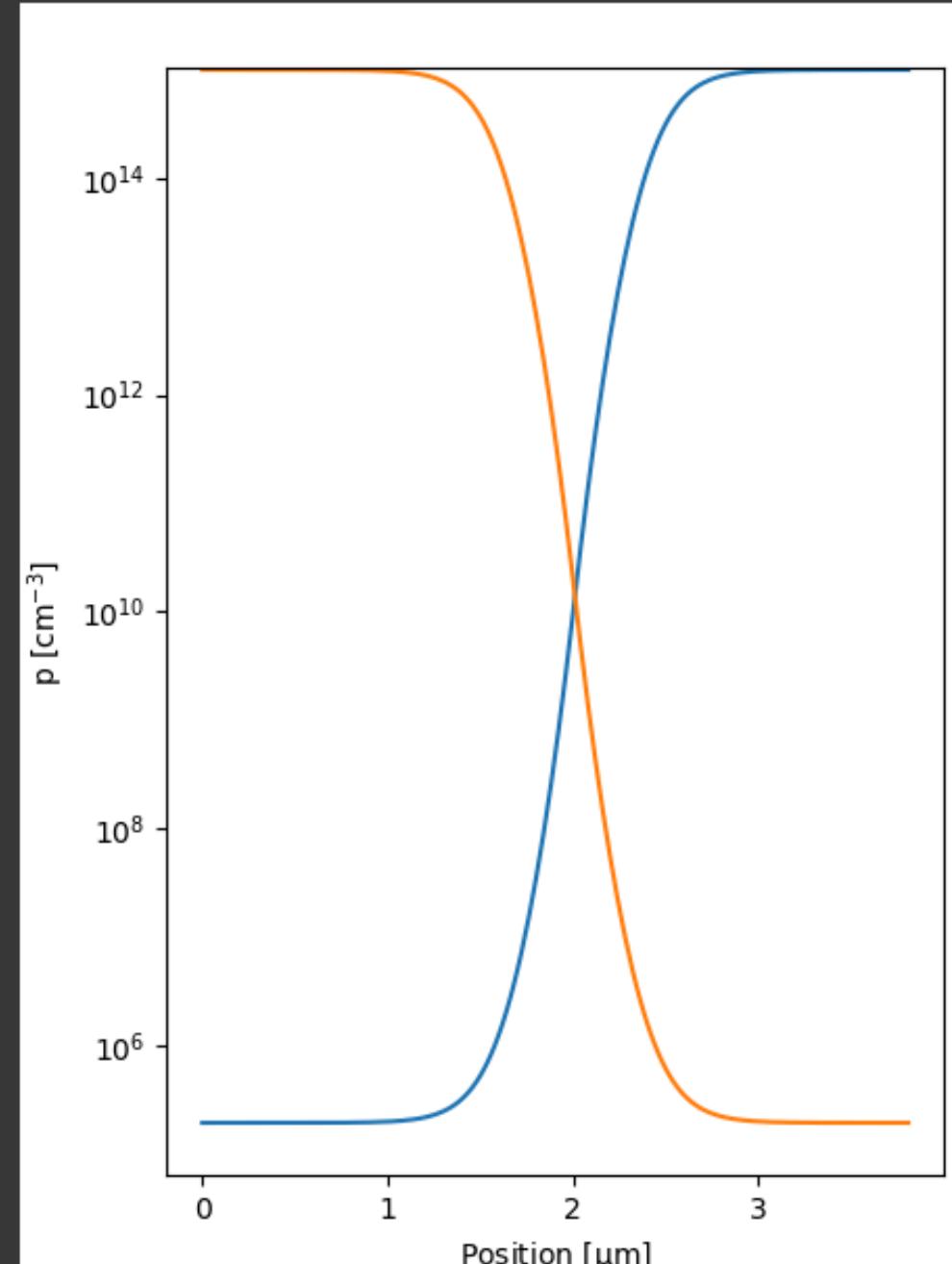
INFO: step 40,error = 0.28784490108657

INFO: step 41,error = 0.05031109907804321

INFO: step 42,error = 0.001309360091800371

INFO: step 43,error = 8.591859291703617e-07

INFO: ** Calculations completed successfully **

**Import data****Upload files...****Remove selected****pndiode_0.gzp**
pndiode_1.gzp
pndiode_2.gzp**Surface plot****Linear plot****Surface plot****Hole density****Plot****Linear plot****X data** Loop values Position

(0,0),(3.8e-4,0)

Y data **Hole density****Clear****Plot****Export**

Thanks!