ESE3700: Circuit-Level Modeling, Design and Optimization for Digital Systems

Lec 2: January 24, 2024

Transistor Introduction and

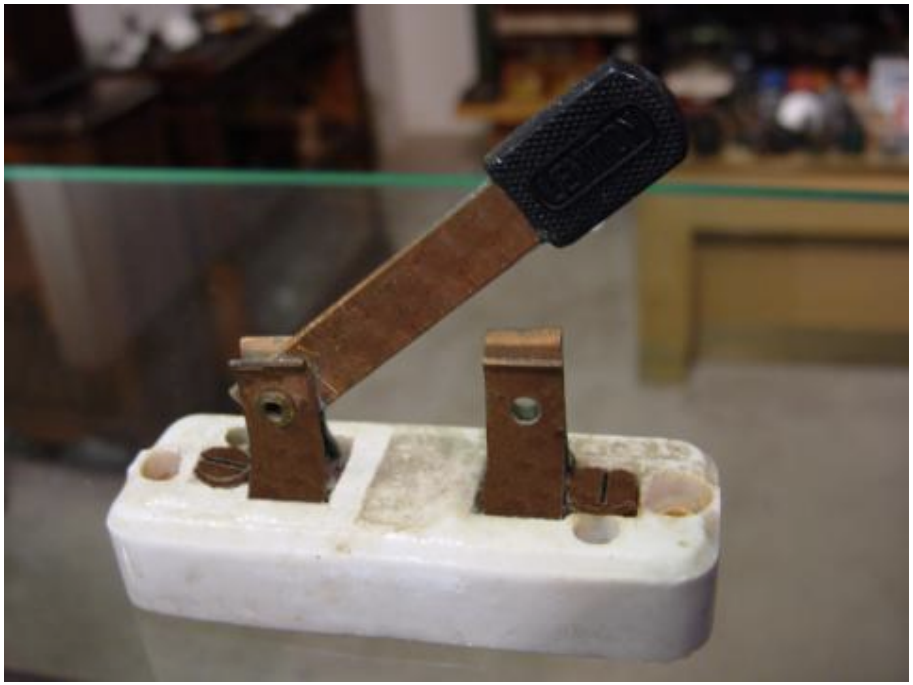
Gates from Transistors

Today

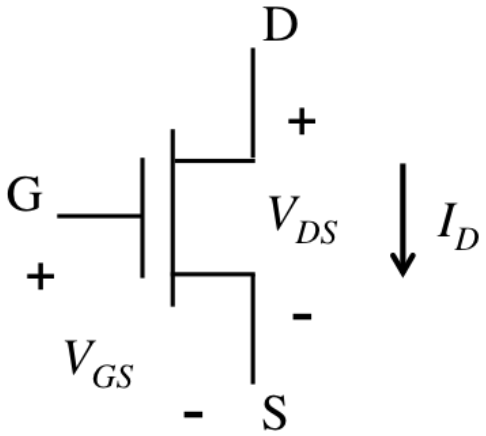
- Transistors – MOSFET
- Zero-th order transistor model
  - Good enough for [what?]
- How to construct static CMOS gates
  - Gate function identification
    - CMOS gate structure
    - Pullup/pulldown networks
- Basic Digital Gates
- Boolean Logic
  - Basic Algebra
  - Minimum Sum of Products/K-maps

Transistor

- Electrical switch to conduct electricity
  - Instead of physically connecting conducting materials to conduct electricity, apply a voltage to conduct



MOSFET



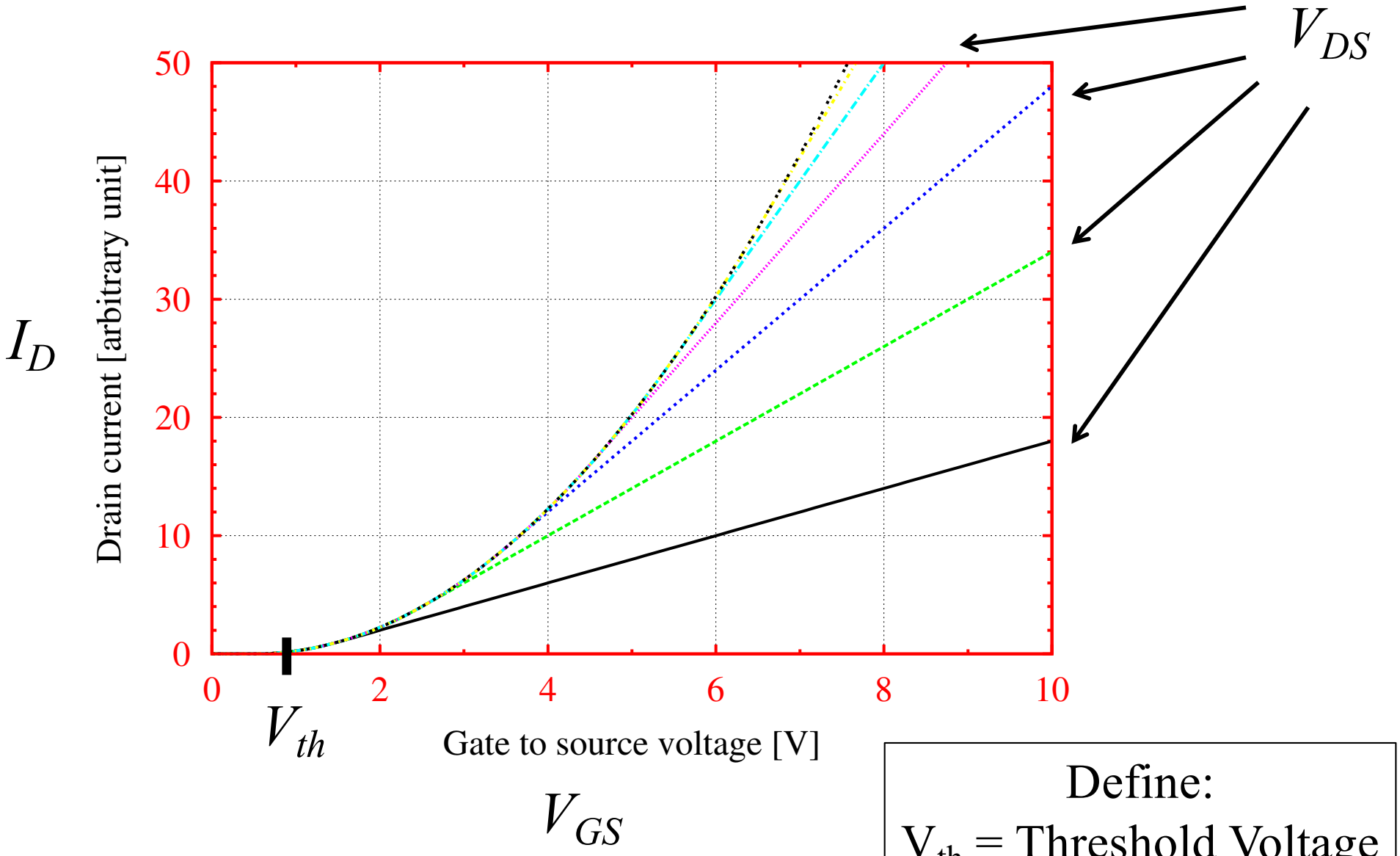
$$V_{GS} = V_G - V_S$$

$$V_{DS} = V_D - V_S$$

$$I_D = f(V_{DS}, V_{GS})$$

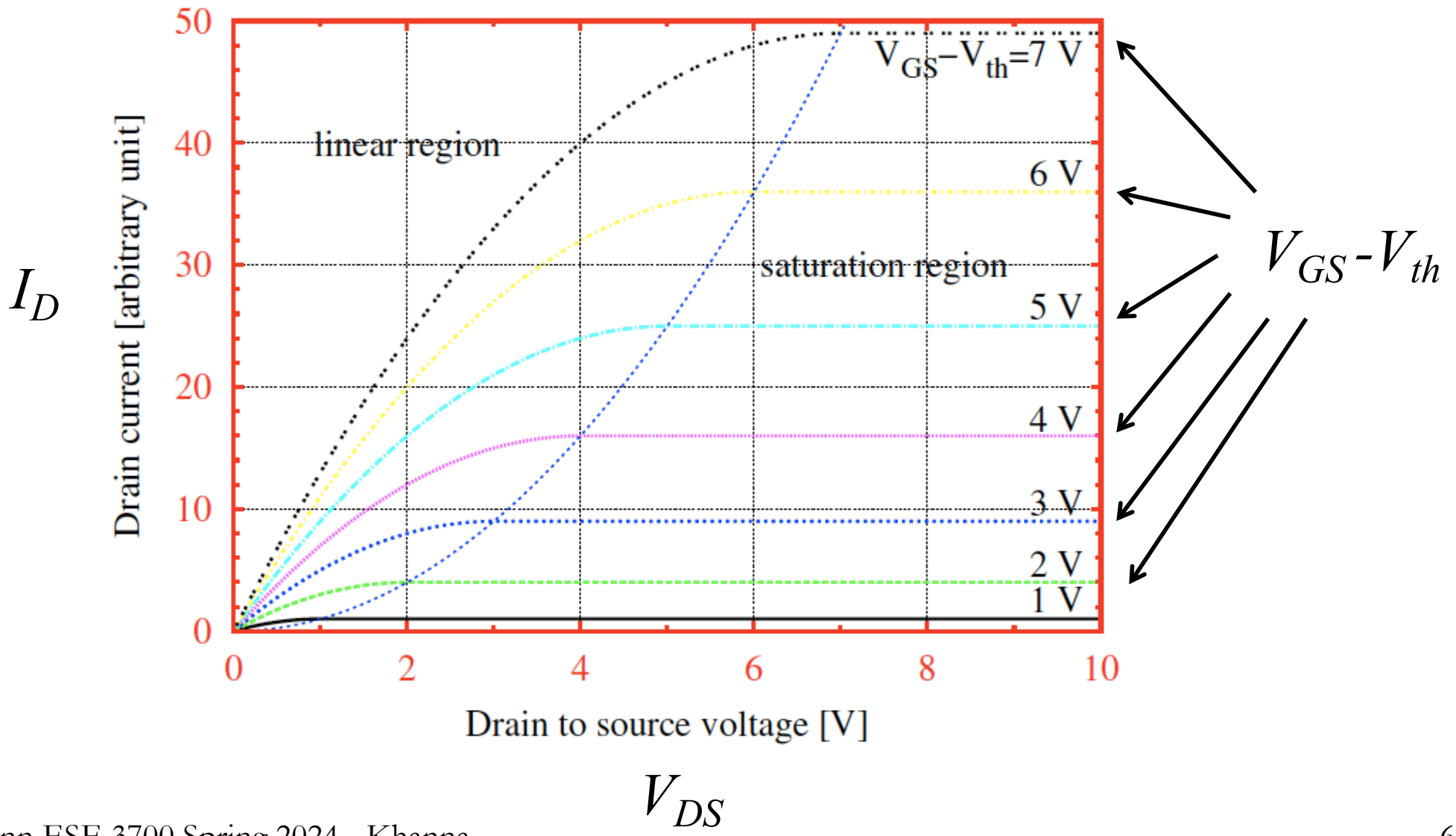
- Metal Oxide Semiconductor Field Effect Transistor
  - Primary **active** component for the term
  - Three terminal device
    - Voltage at gate controls conduction between two other terminals (source, drain)

MOSFET – IV Characteristics

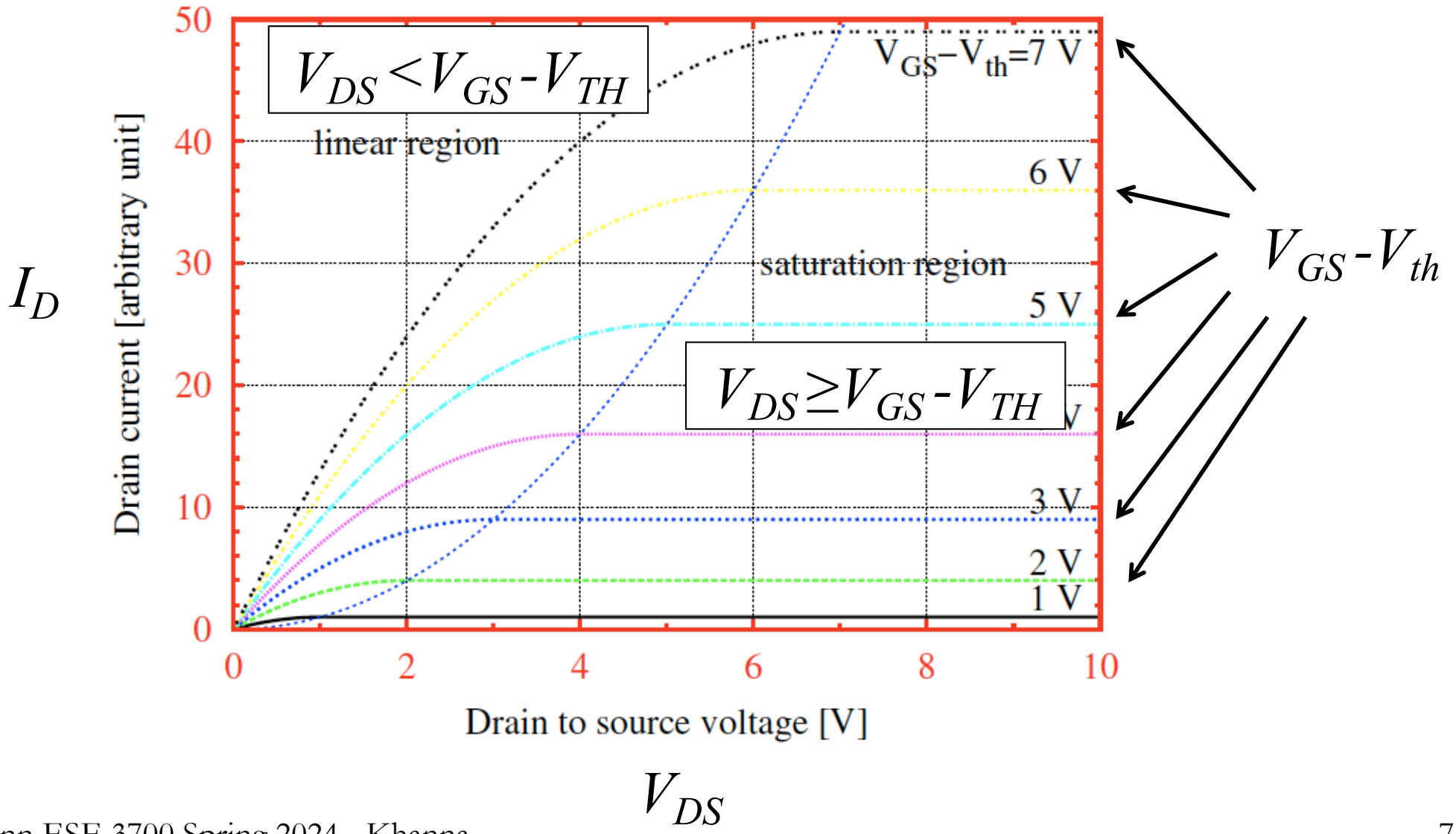


Define:
V_{th} = Threshold Voltage

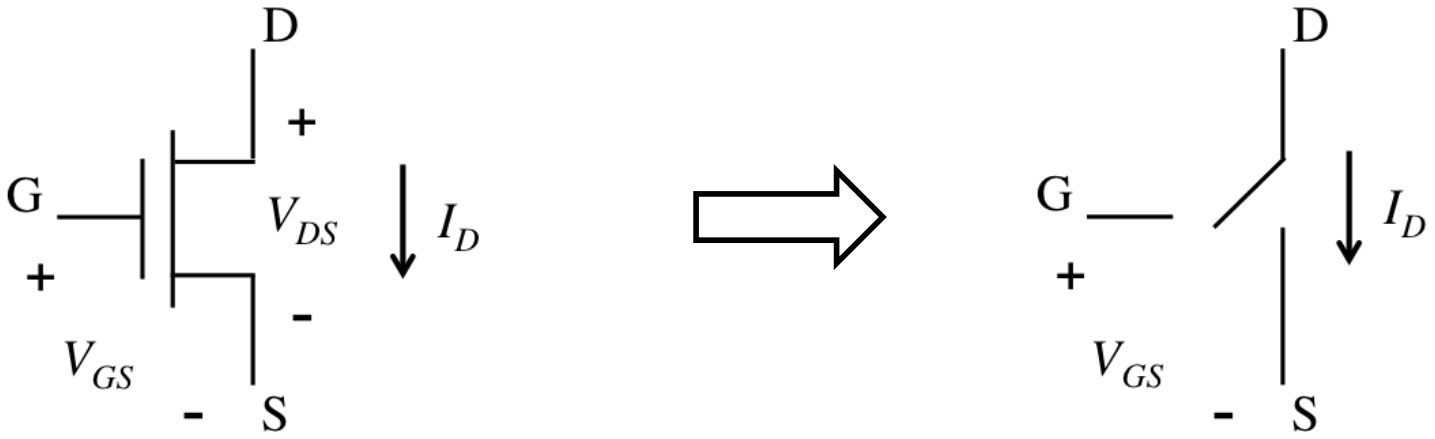
MOSFET – IV Characteristics



MOSFET – IV Characteristics

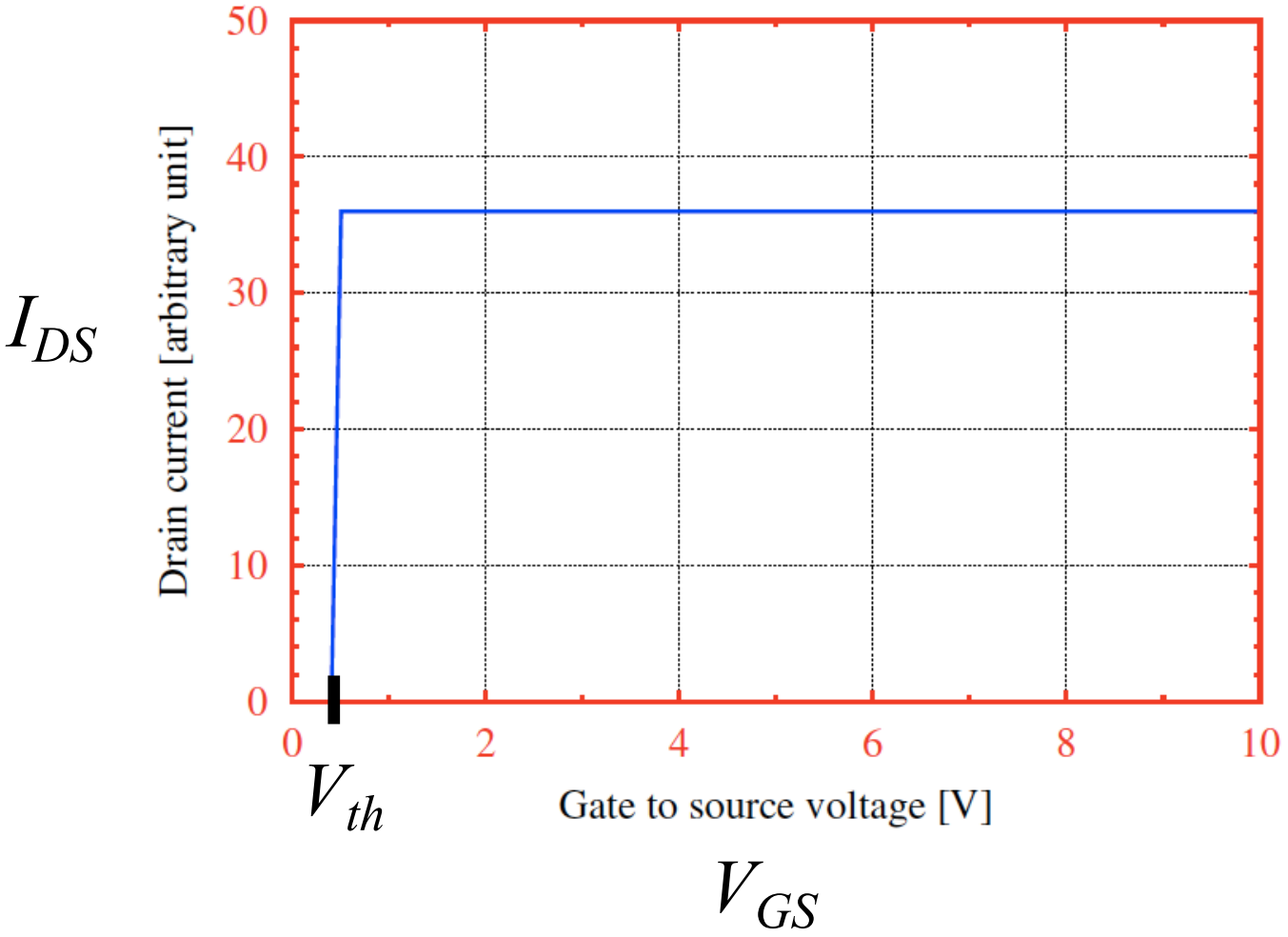


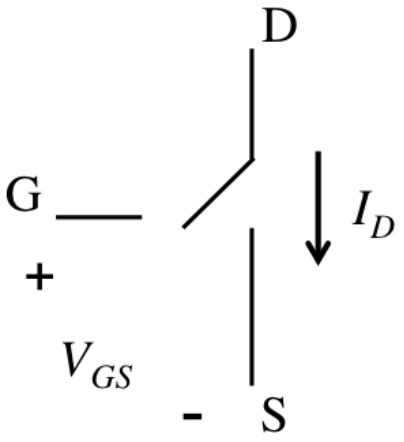
MOSFET – Zeroeth Order Model



- Ideal Switch
  - $V_{GS} > V_{th}$ → switch is closed, conducts, "on"
  - $V_{GS} < V_{th}$ → switch is open, does not conduct, "off"
- Gate draws no current from input
  - Loads input capacitively (gate capacitance)

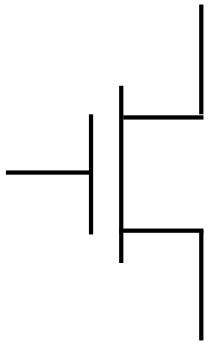
MOSFET – Zeroeth Order Model





MOSFET - Symmetric

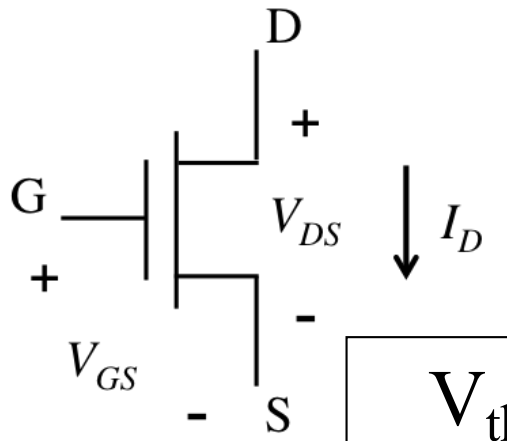
- Switch turned on for positive V_{GS}
  - Which side is drain or source?



MOSFET – N-Type

- Switch turned on for positive V_{GS}

$V_D > V_S$



$V_{th,n} > 0$

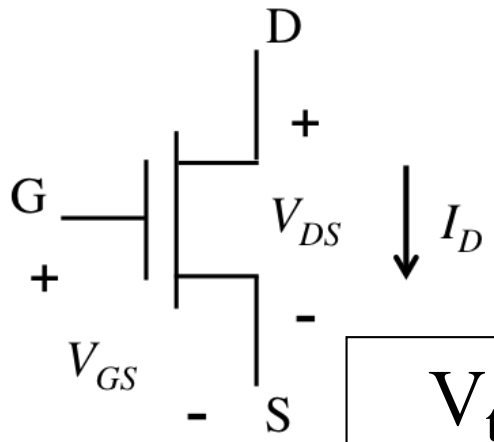
$V_{GS} > V_{th,n}$

to conduct

MOSFET – N-Type, P-Type

- Switch turned on for positive V_{GS}
- Switch turned on for negative V_{GS}

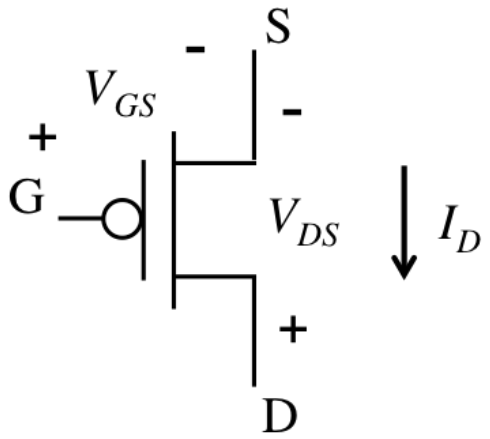
$V_D > V_S$



$V_{th,n} > 0$

$V_{GS} > V_{th,n}$ to conduct

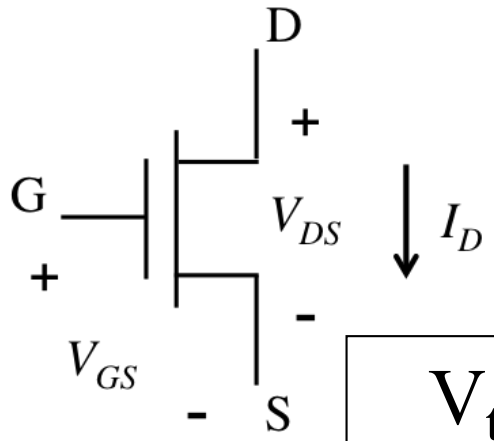
$V_S > V_D$



MOSFET – N-Type, P-Type

- Switch turned on for positive V_{GS}
- Switch turned on for negative V_{GS}

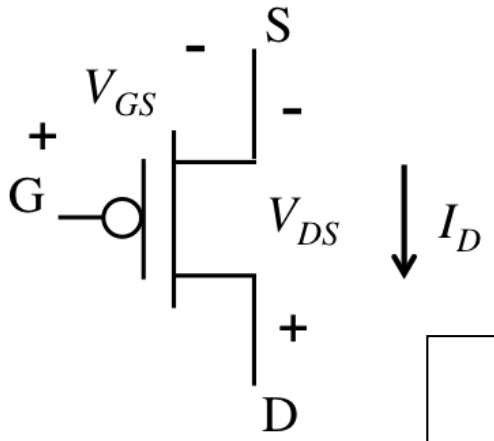
$V_D > V_S$



$V_{th,n} > 0$

$V_{GS} > V_{th,n}$ to conduct

$V_S > V_D$



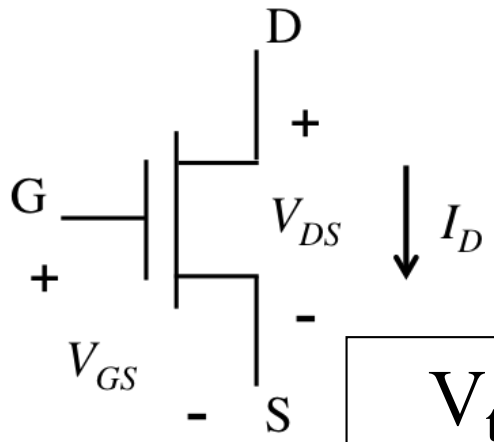
$V_{th,p} < 0$

$V_{GS} < V_{th,p}$ to conduct

MOSFET – N-Type, P-Type

- Switch turned on for positive V_{GS}
- Switch turned on for negative V_{GS}

$V_D > V_S$

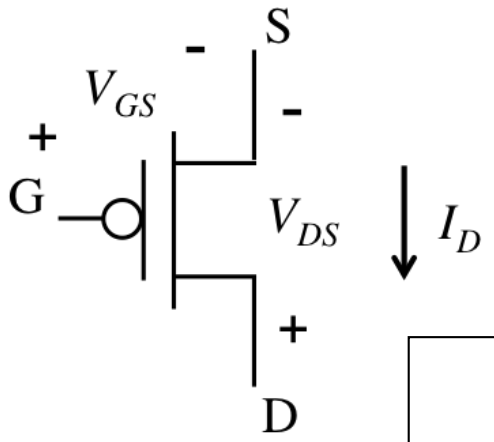


$V_{th,n} > 0$

$V_{GS} > V_{th,n}$

to conduct

$V_S > V_D$



$|V_{th,p}| > 0$

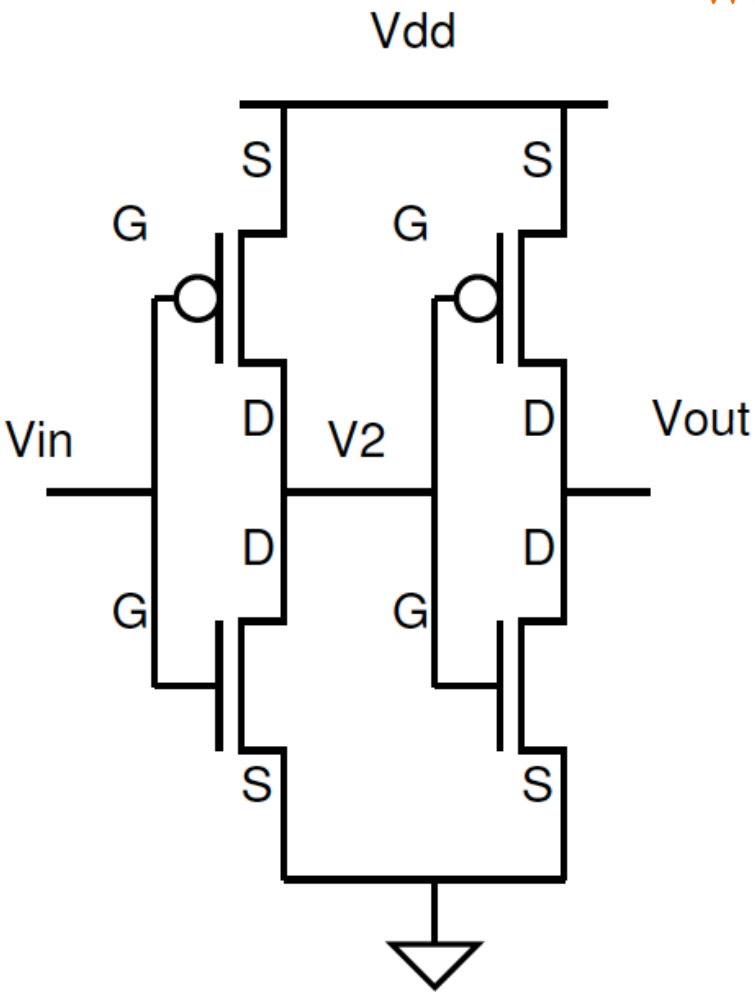
$|V_{GS}| > |V_{th,p}|$

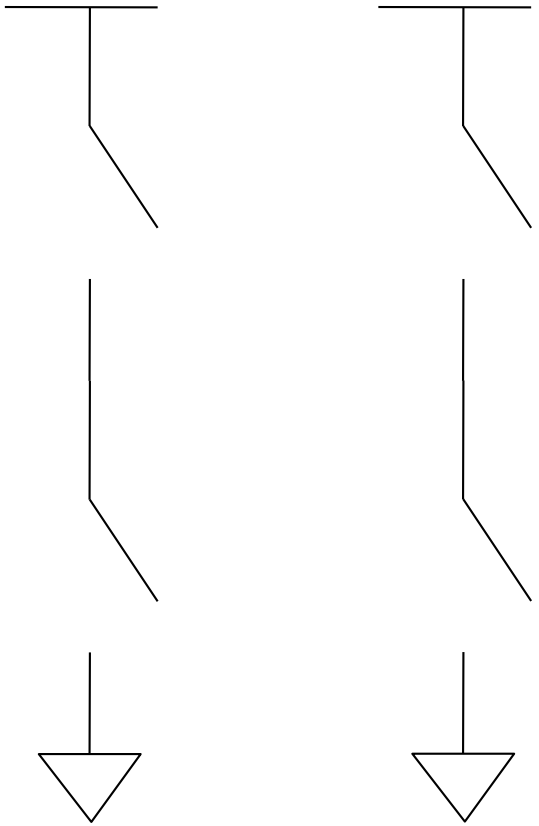
to conduct

Apply zero-order model

Note S, D annotation on this slide (won't be labeled in future)

Why is it this way?



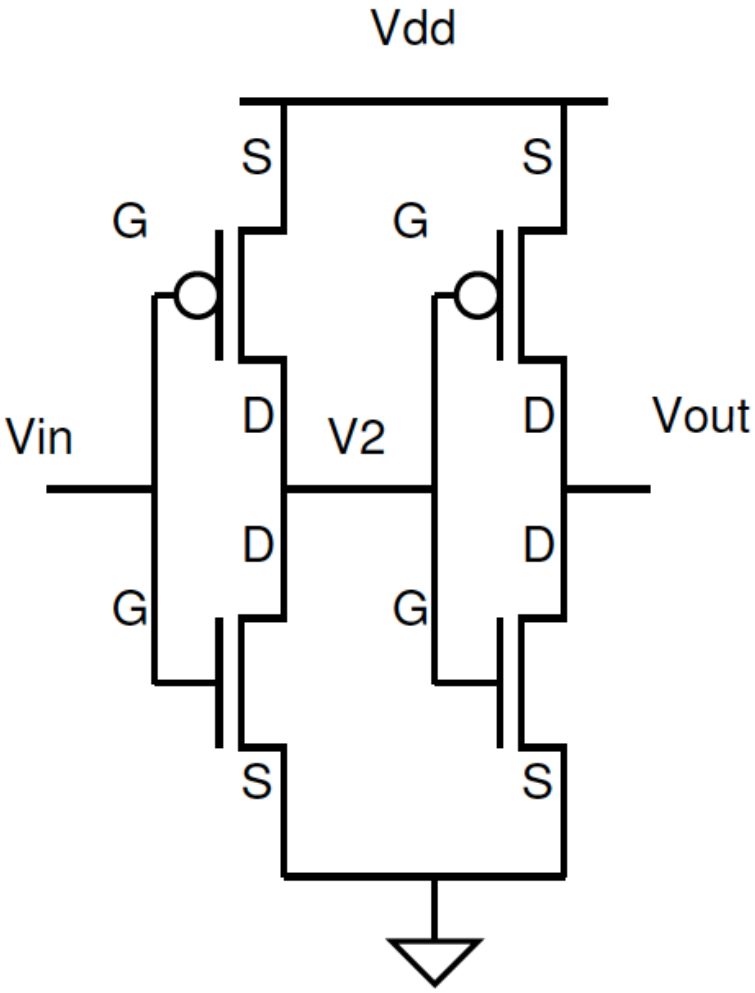


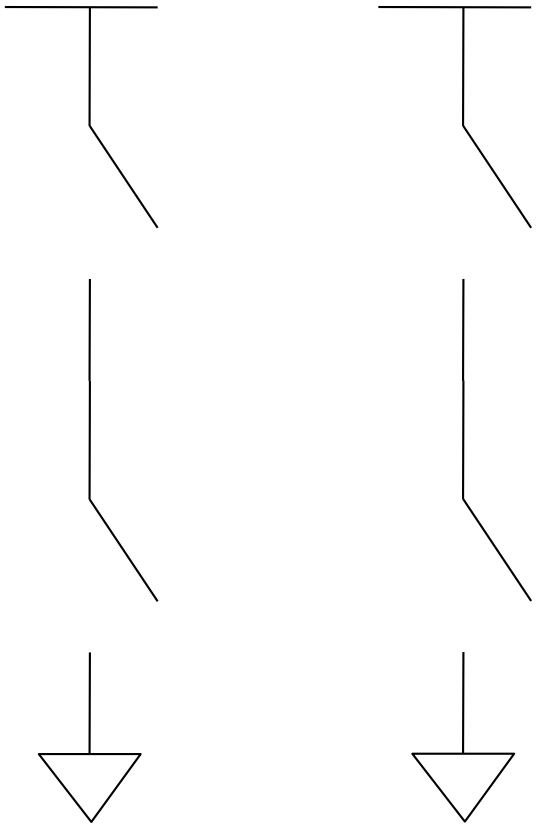
Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



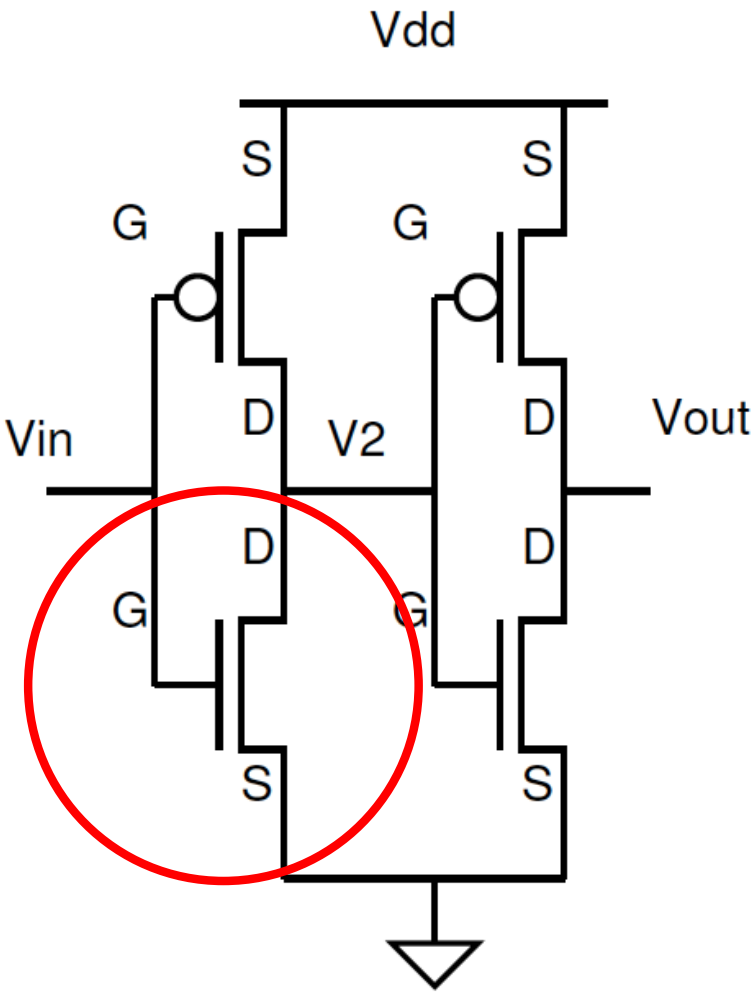


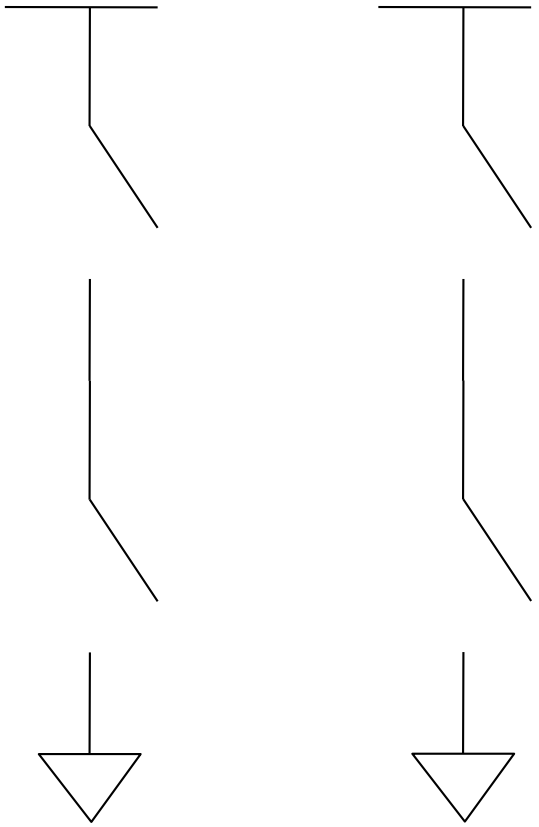
Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$

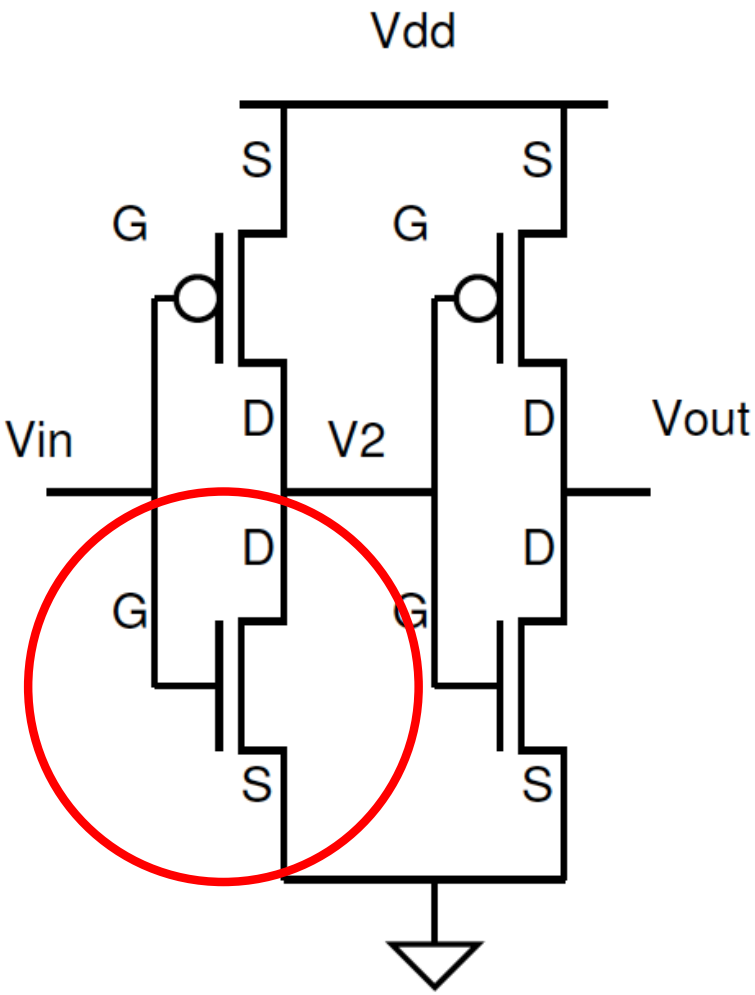


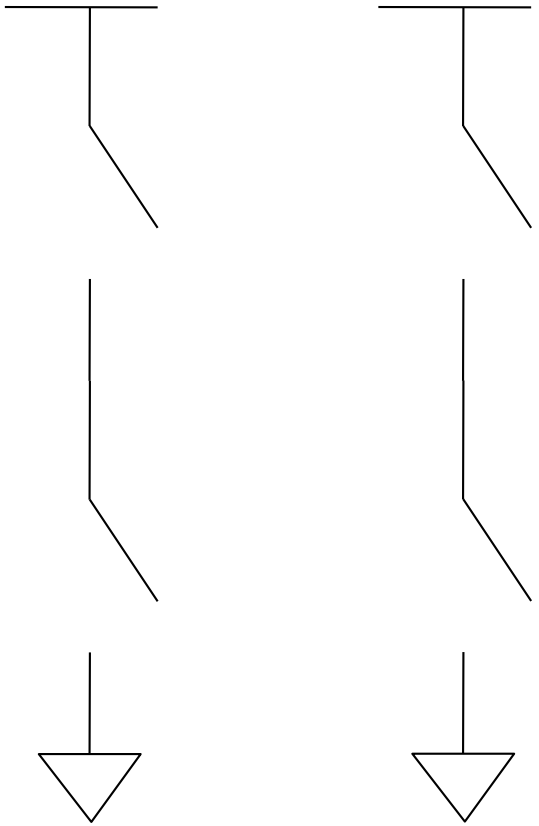


Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$
$V_{GS} = V_G - V_S$



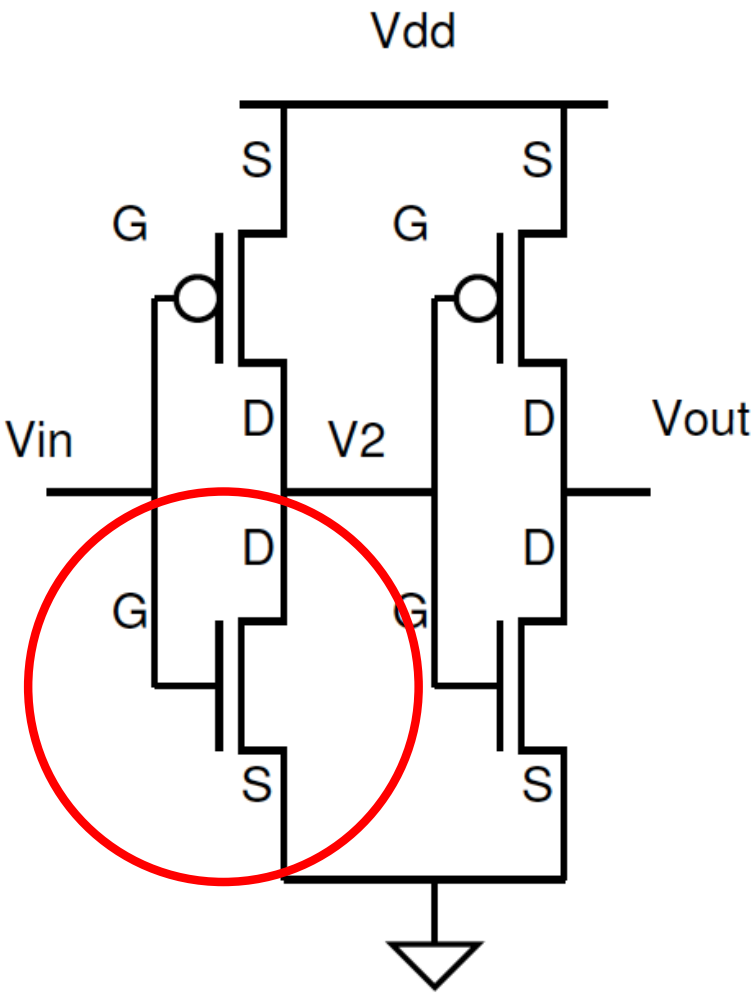


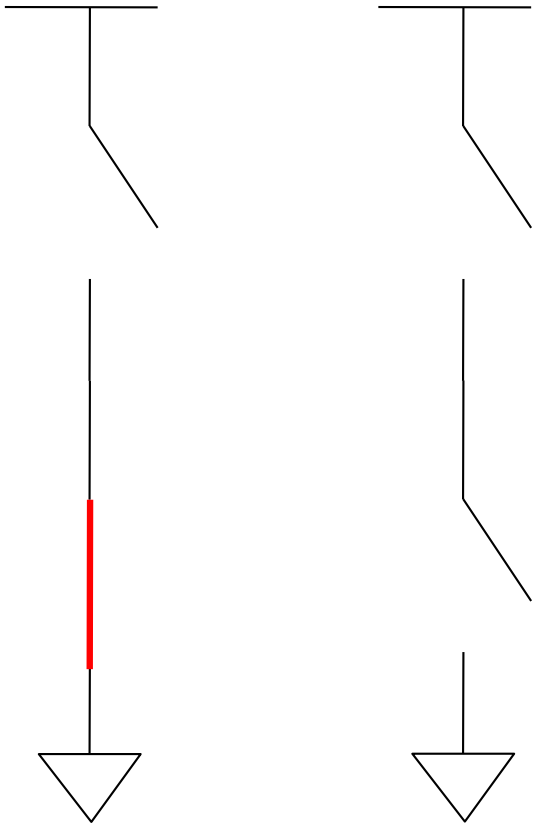
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$
$V_{GS} = V_G - V_S$



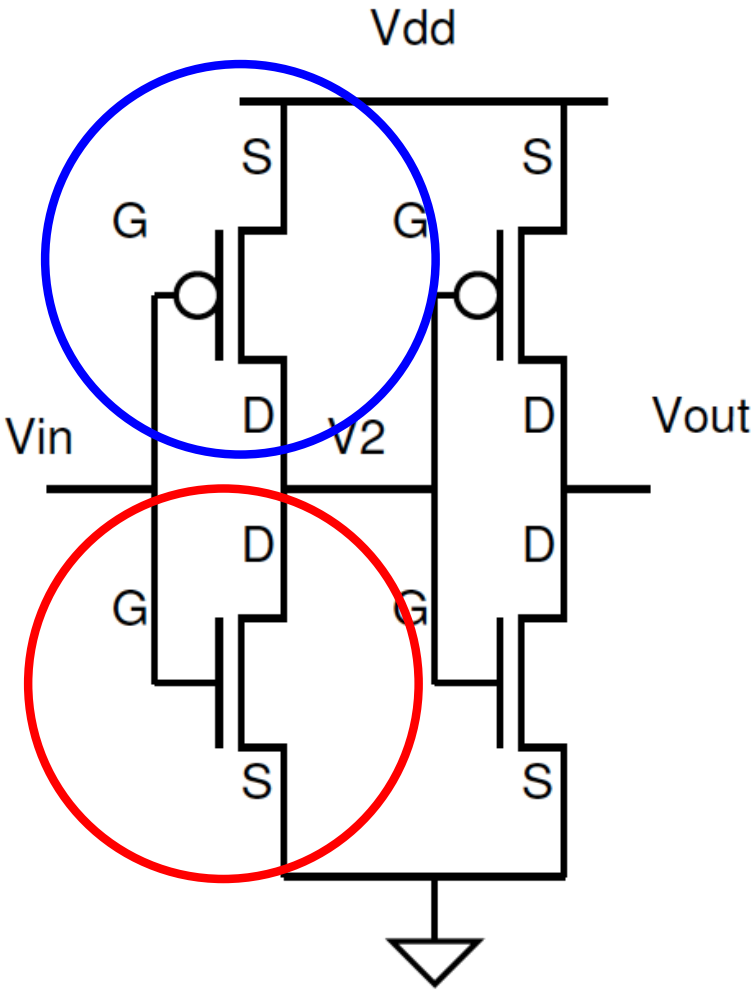


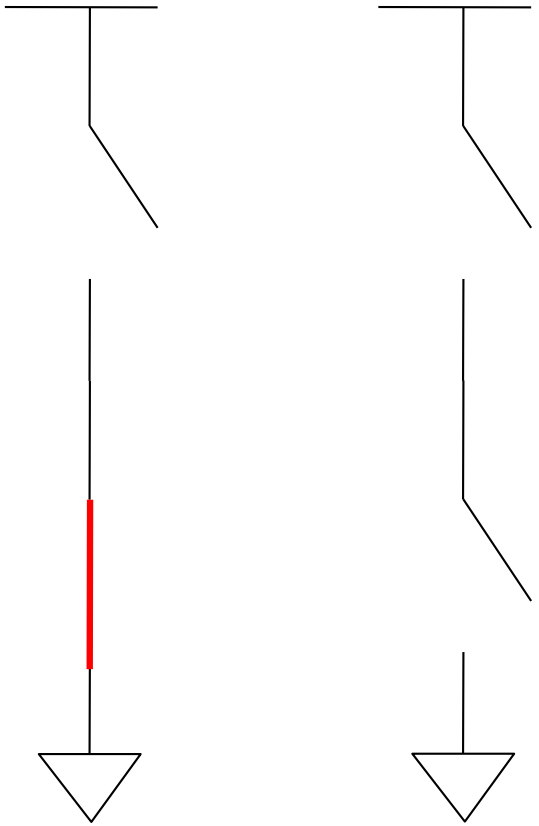
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$
$V_{GS} = V_G - V_S$





$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

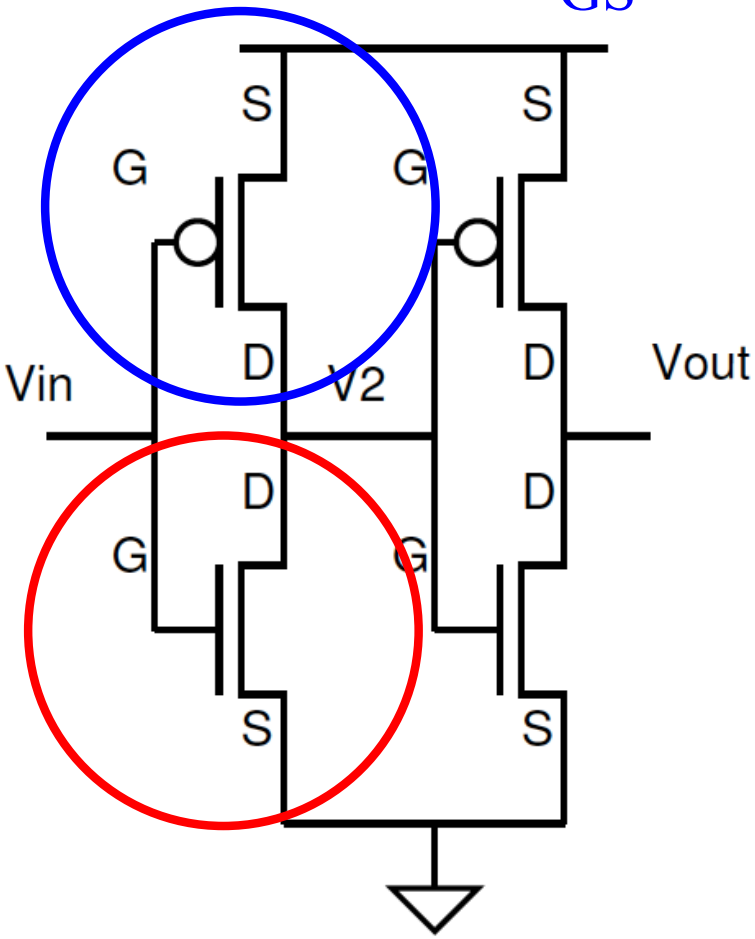
Apply zero-order model?

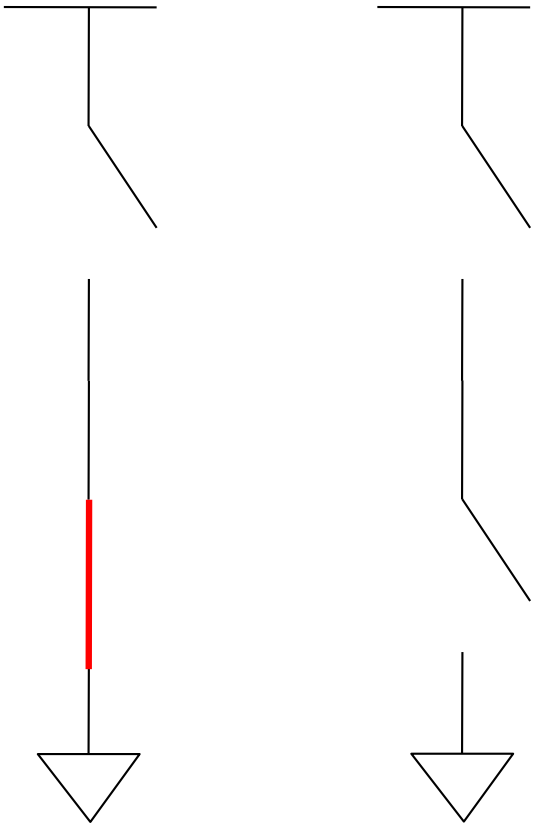
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$

Vdd $V_{GS} = 0 > V_{th,p}$





$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

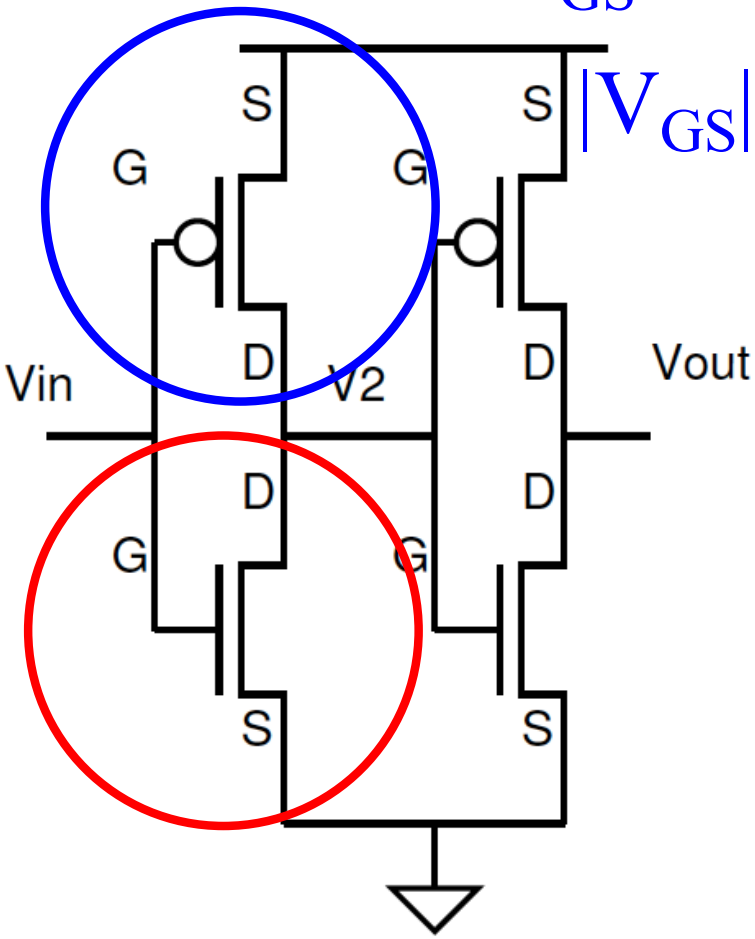
What happens when $V_{in} = V_{dd} > V_{thn}$?

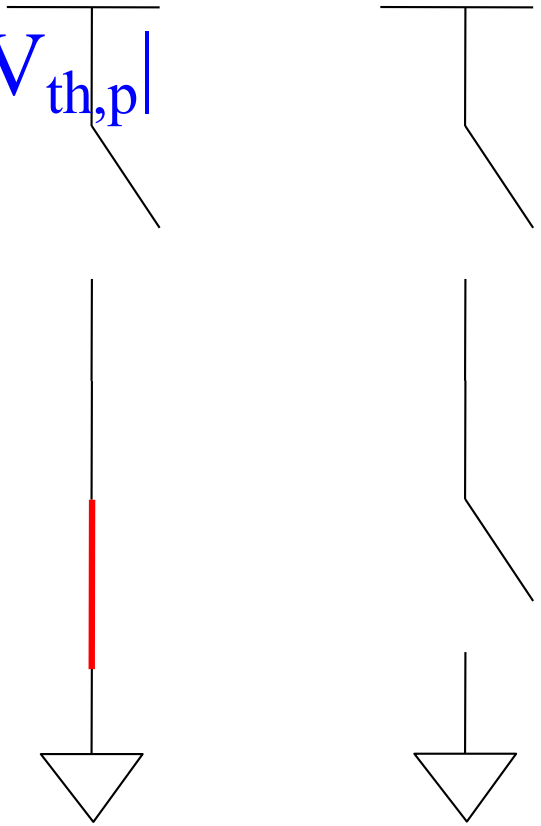
$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$

Vdd $V_{GS} = 0 > V_{th,p}$

$|V_{GS}| = 0 < |V_{th,p}|$





$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

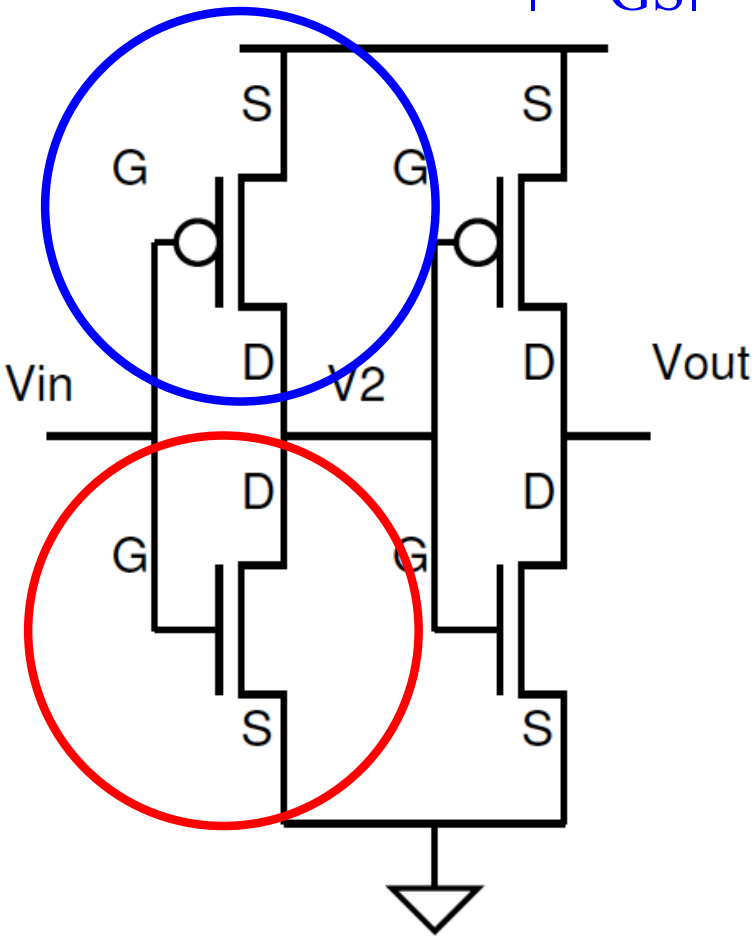
Apply zero-order model?

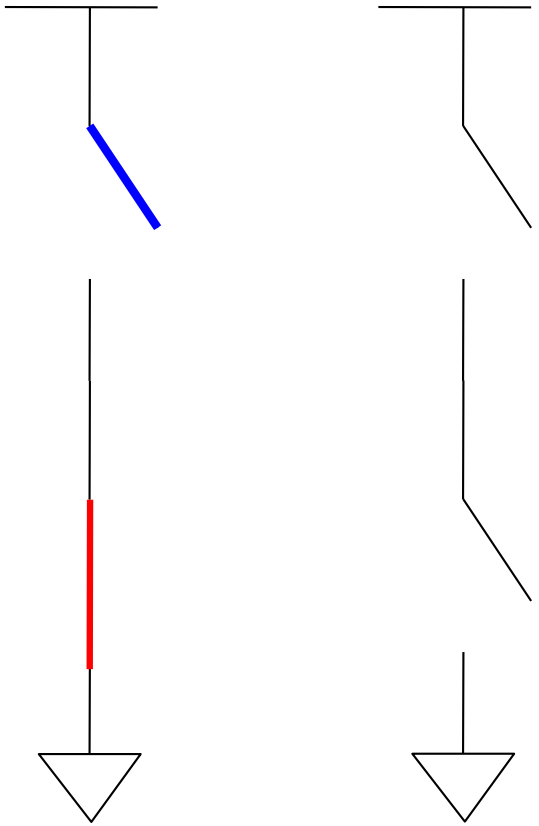
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$

$|V_{GS}| = 0 < |V_{th,p}|$





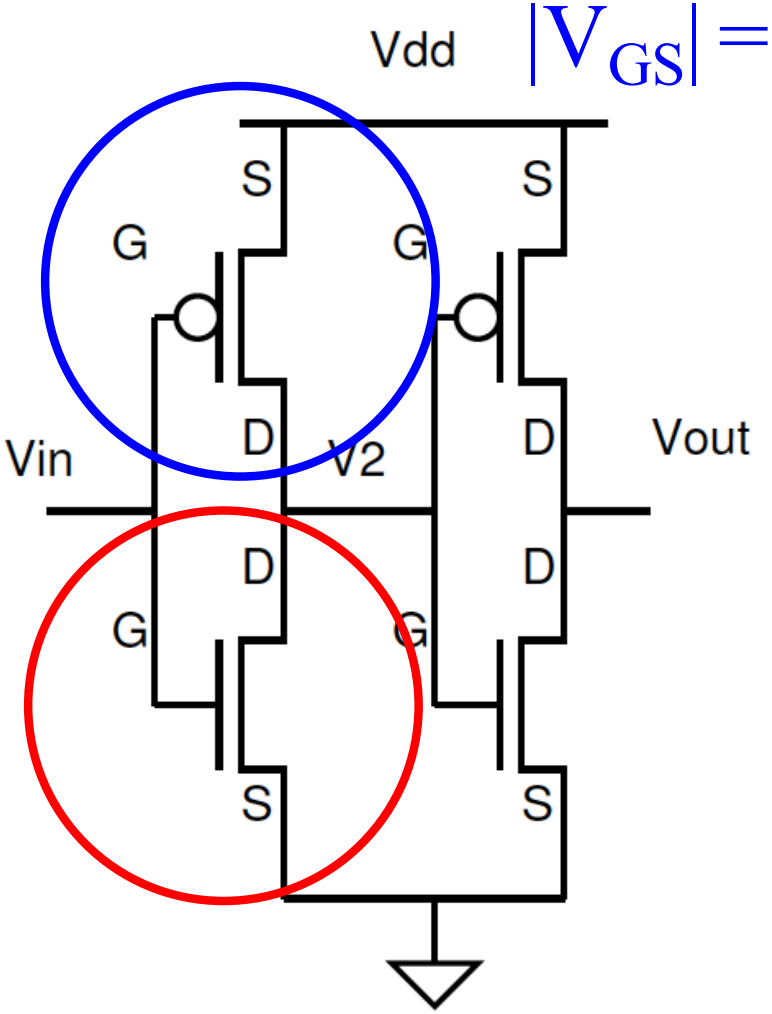
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

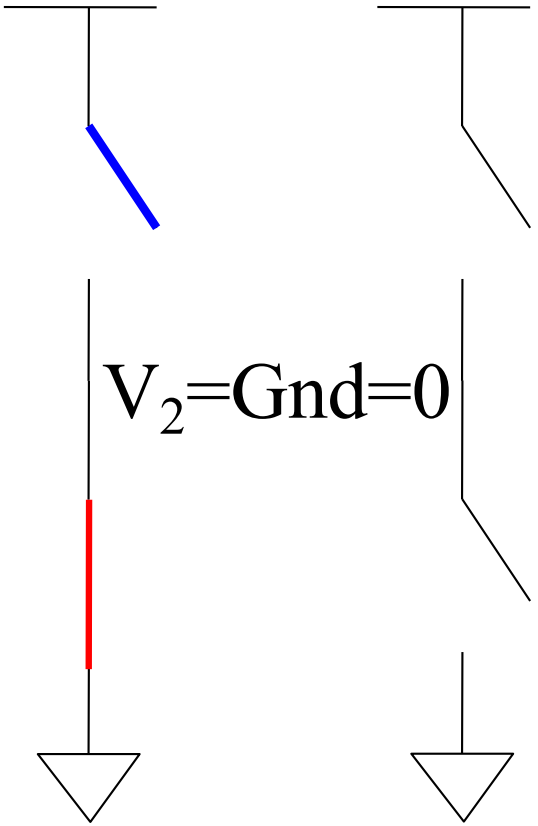
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$



$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

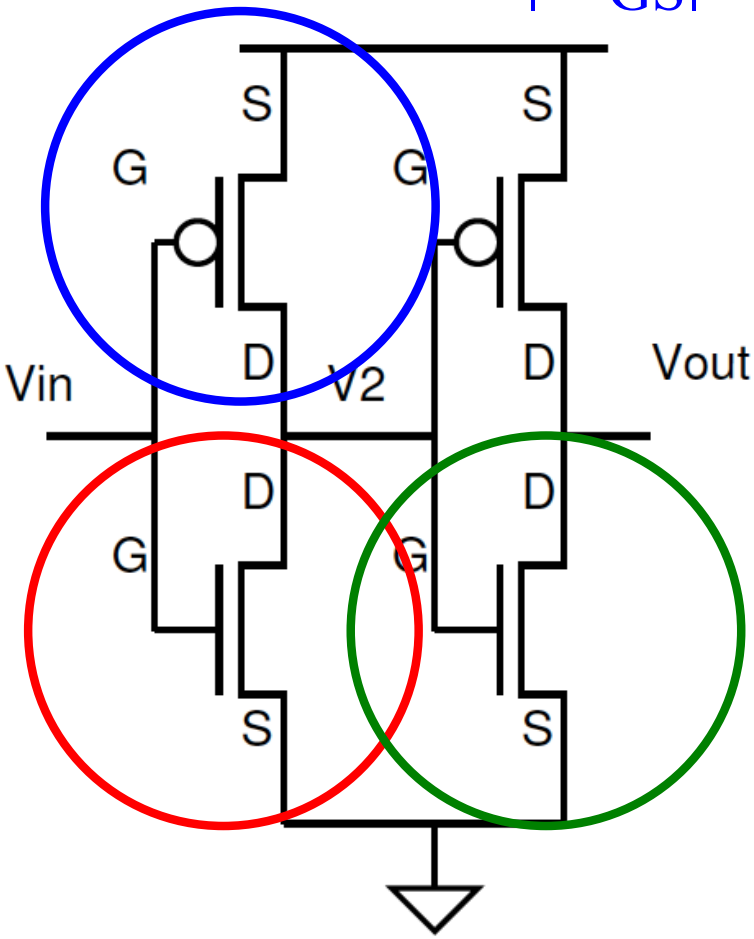
Apply zero-order model?

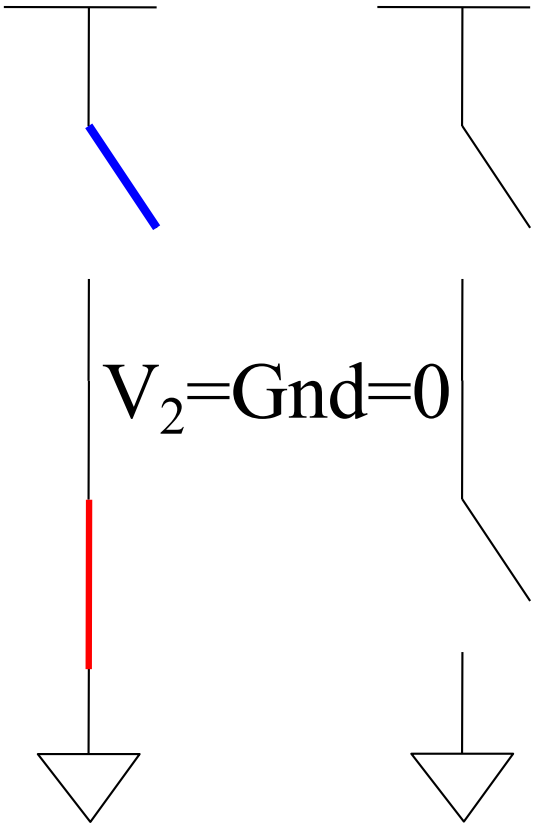
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$

$|V_{GS}| = 0 < |V_{th,p}|$





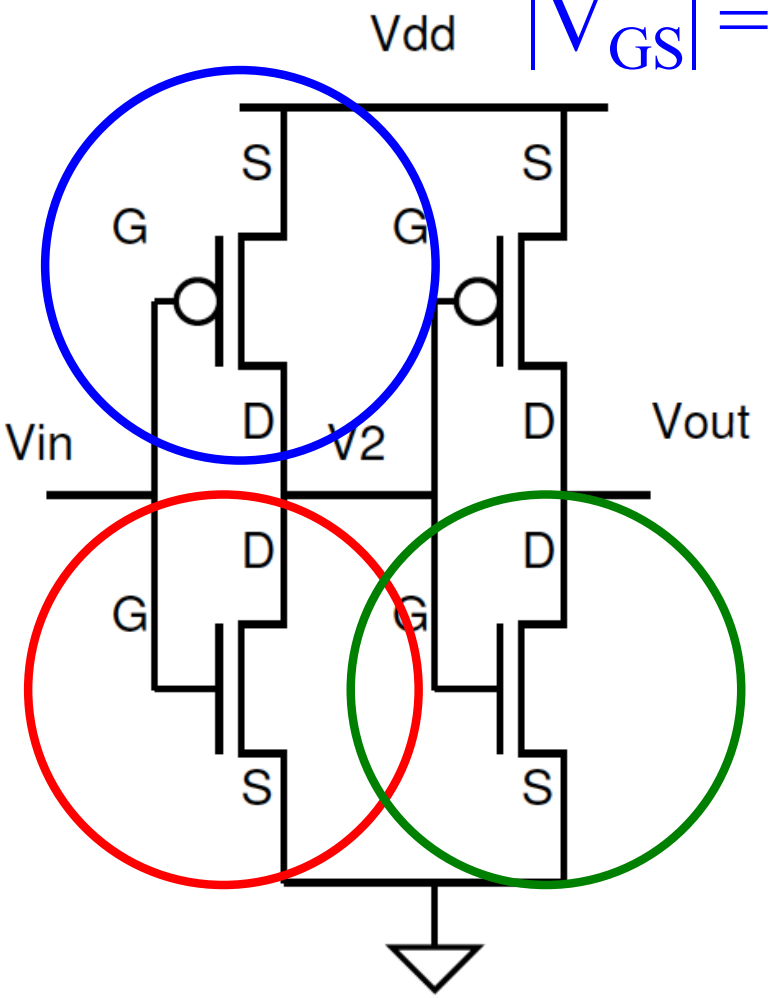
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

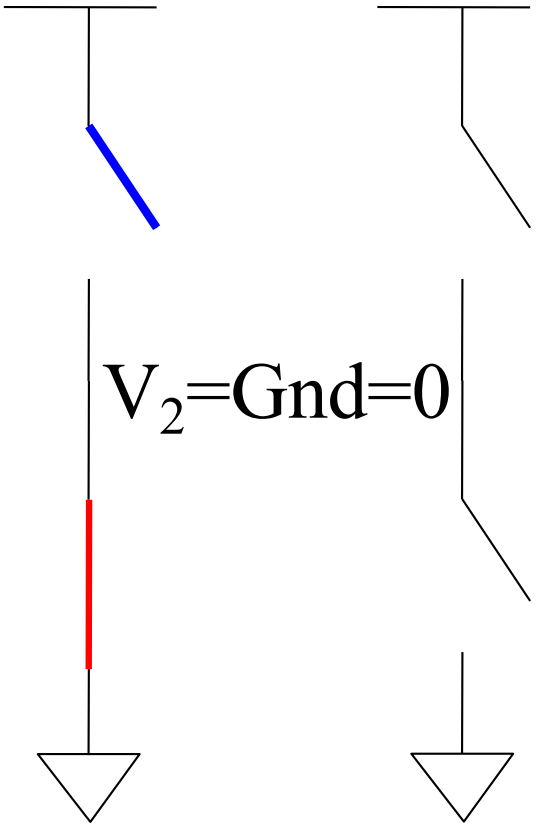
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$



$V_{GS} = 0 < V_{th,n}$

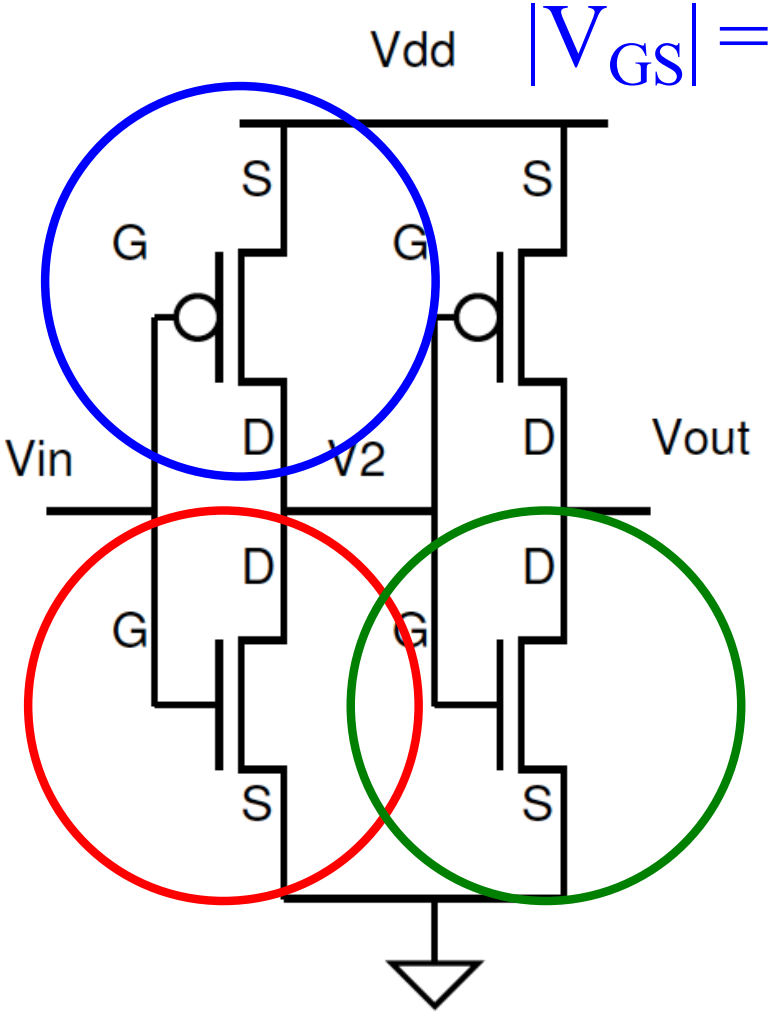
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

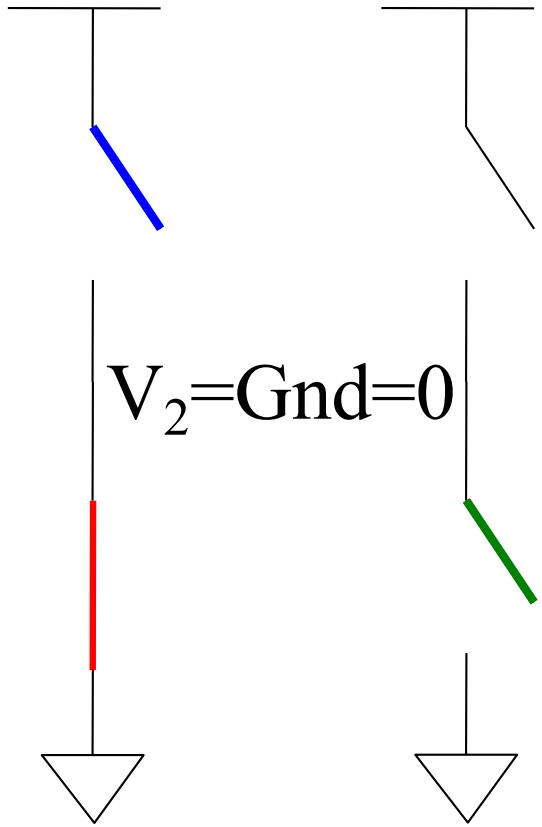
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$



$V_{GS} = 0 < V_{th,n}$

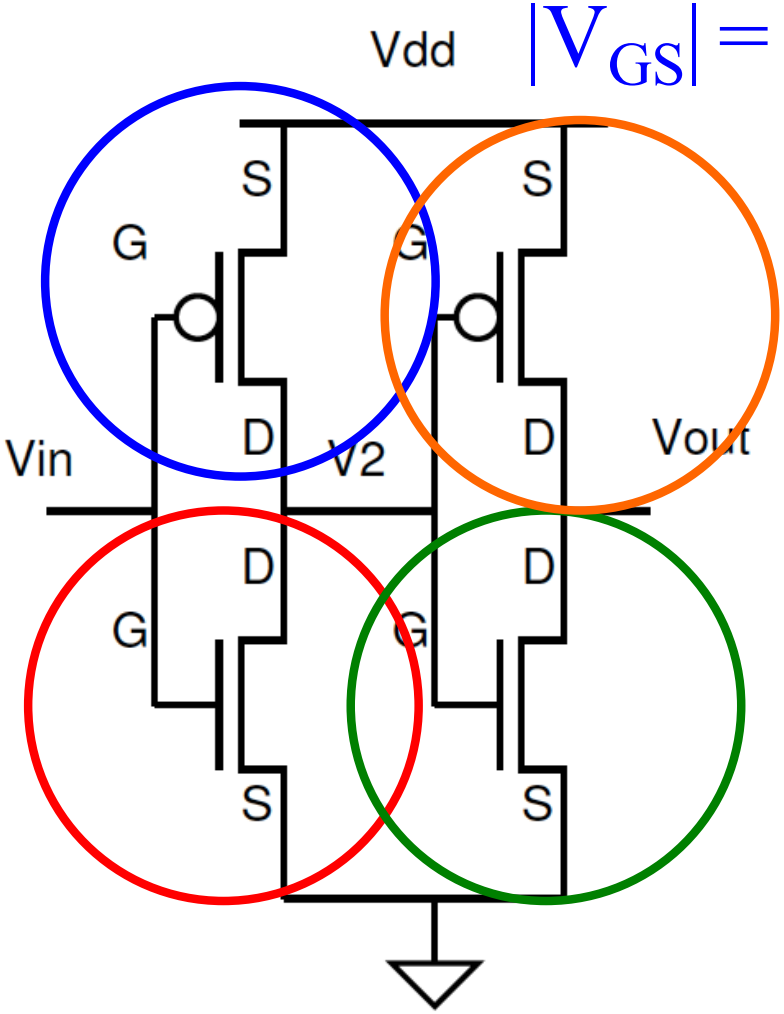
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

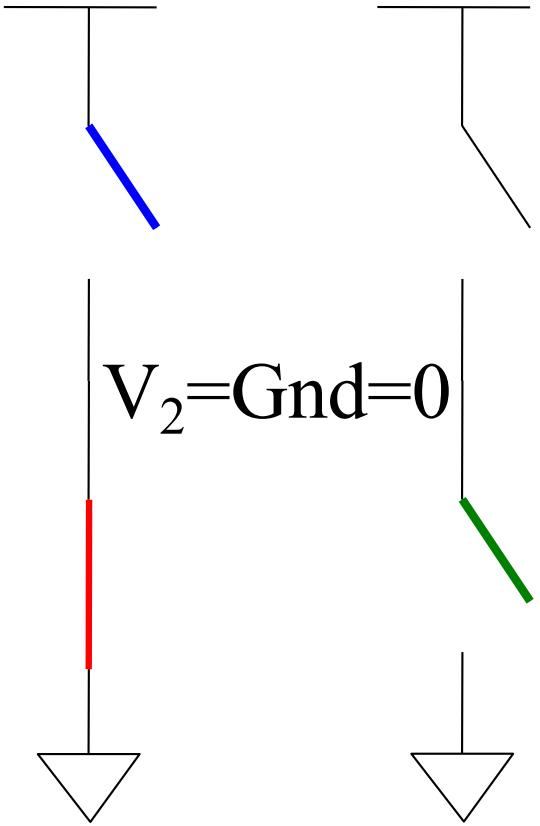
What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$



$V_{GS} = 0 < V_{th,n}$

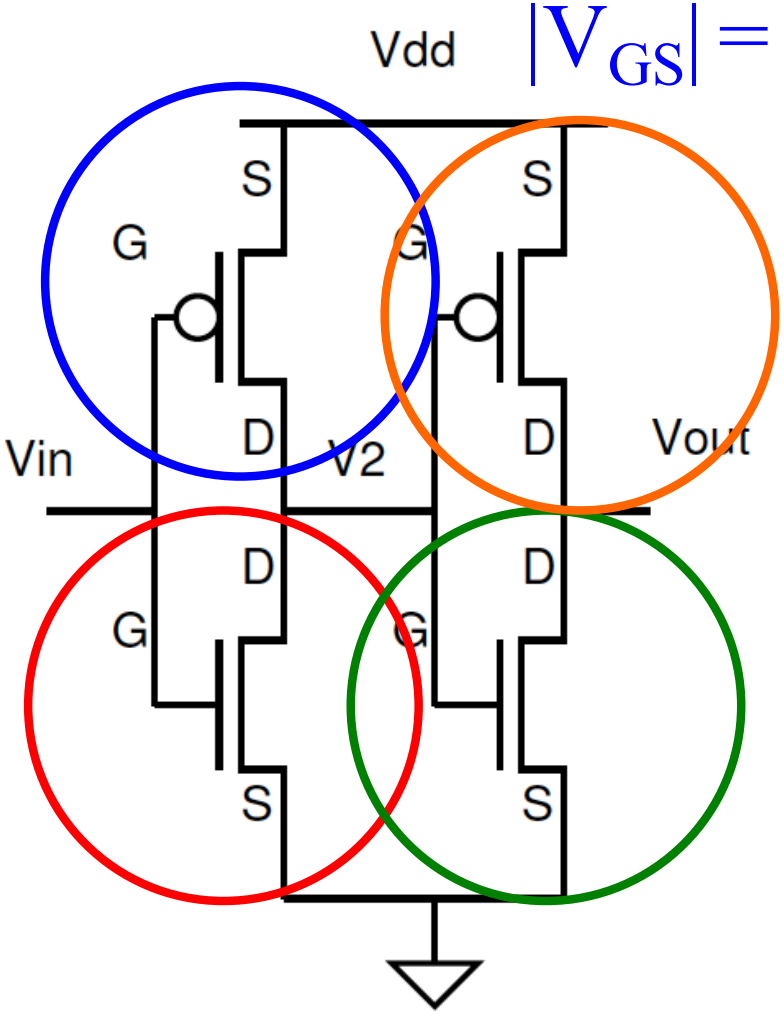
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$

$V_{GS} = -V_{dd} < V_{th,p}$

V_2=Gnd=0

$V_{GS} = 0 < V_{th,n}$

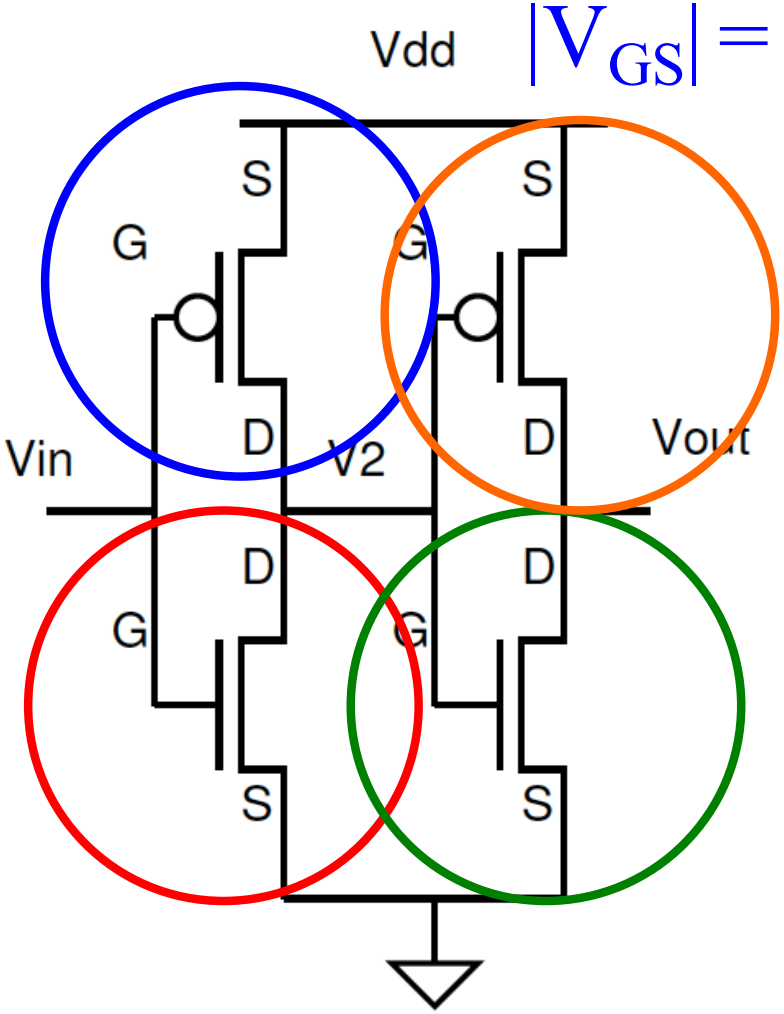
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$

$V_{GS} = -V_{dd} < V_{th,p}$

$|V_{GS}| = V_{dd} > |V_{th,p}|$

V_2=Gnd=0

$V_{GS} = 0 < V_{th,n}$

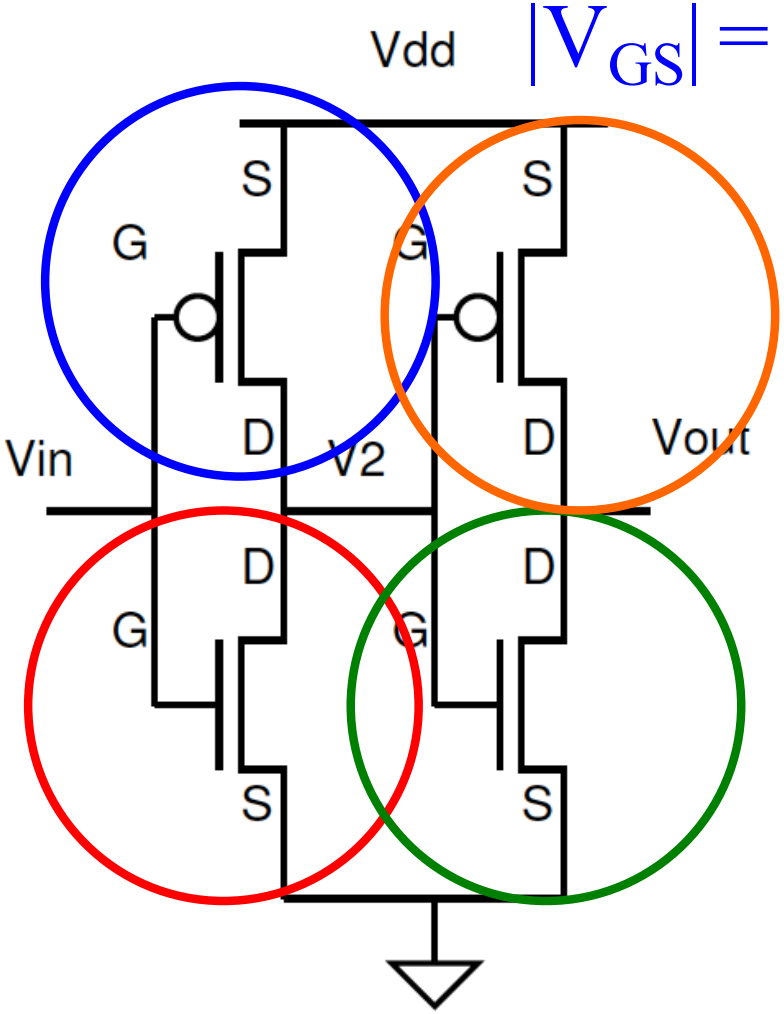
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

Apply zero-order model?

What happens when $V_{in} = V_{dd} > V_{thn}$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



$|V_{GS}| = 0 < |V_{th,p}|$

$V_{GS} = -V_{dd} < V_{th,p}$

V_2=Gnd=0

$V_{out} = V_{dd}$

$V_{GS} = 0 < V_{th,n}$

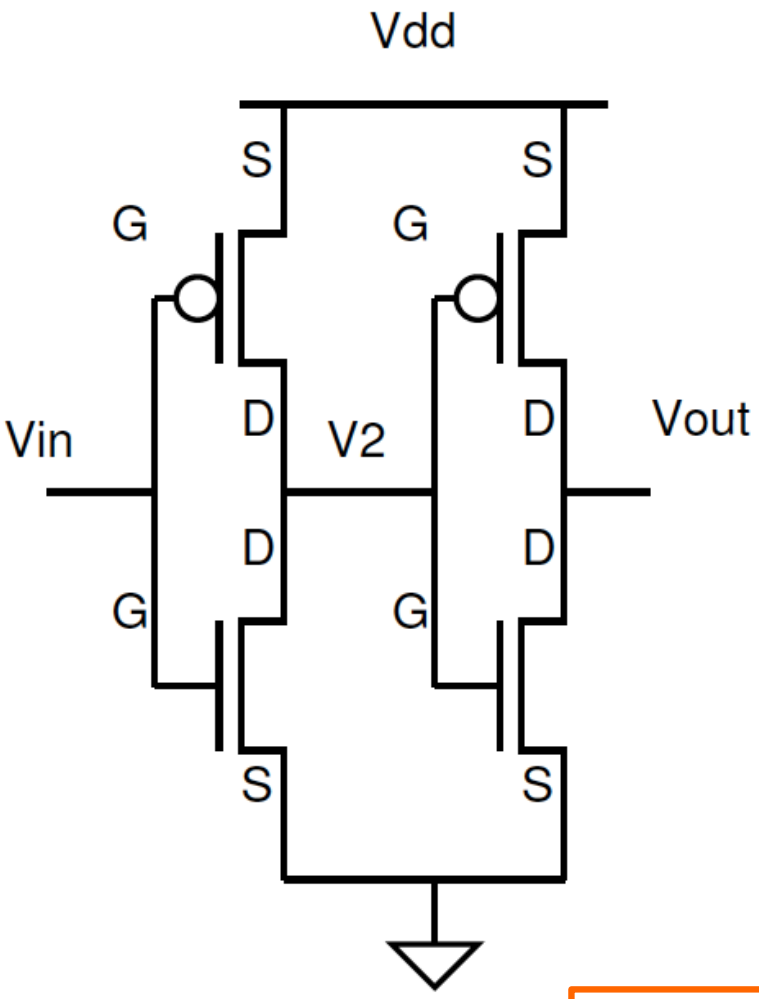
$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$

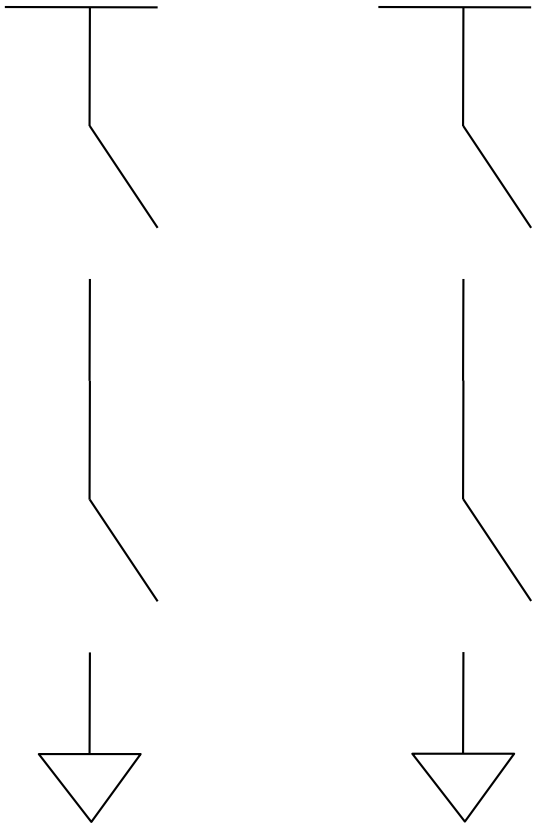
Apply zero-order model?

What happens when $V_{in} = 0$?

$V_{th,p} = -V_{th,n}$

$V_{GS} = V_G - V_S$



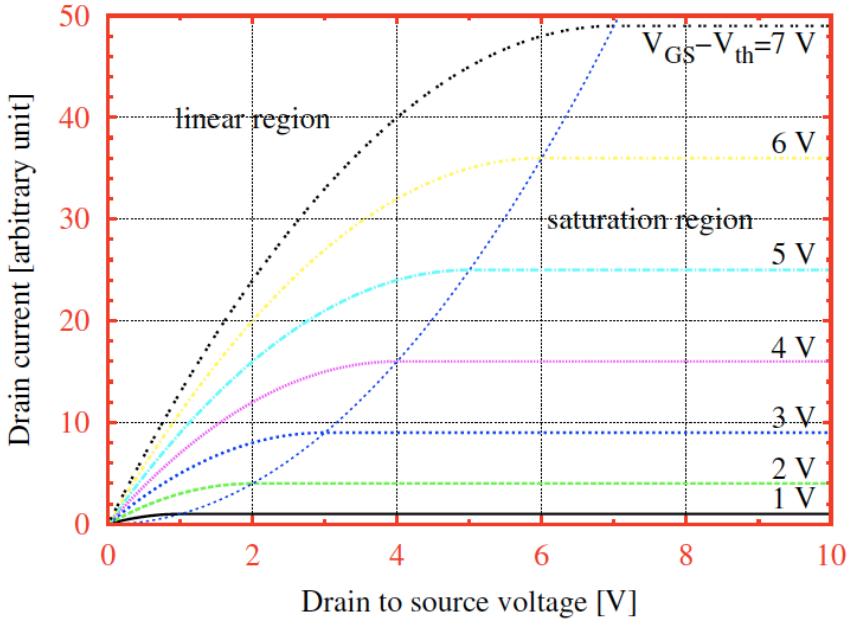


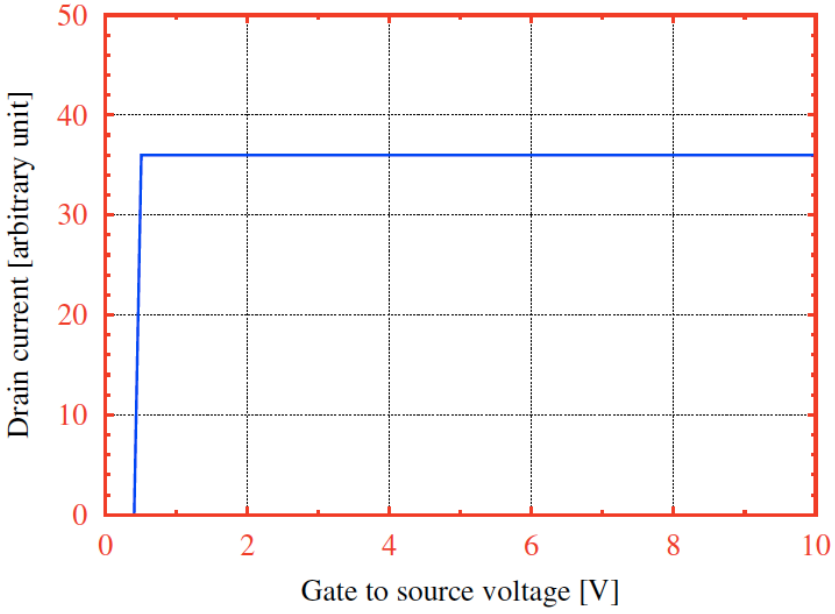
Convince yourself that $V_{out} = 0$.

Zeroeth Order Model

- Allows us to reason (mostly) at logic level about steady-state functionality of typical gate circuits before worrying about performance (speed, power, etc.) details

What is missing in Zeroeth Order Model?

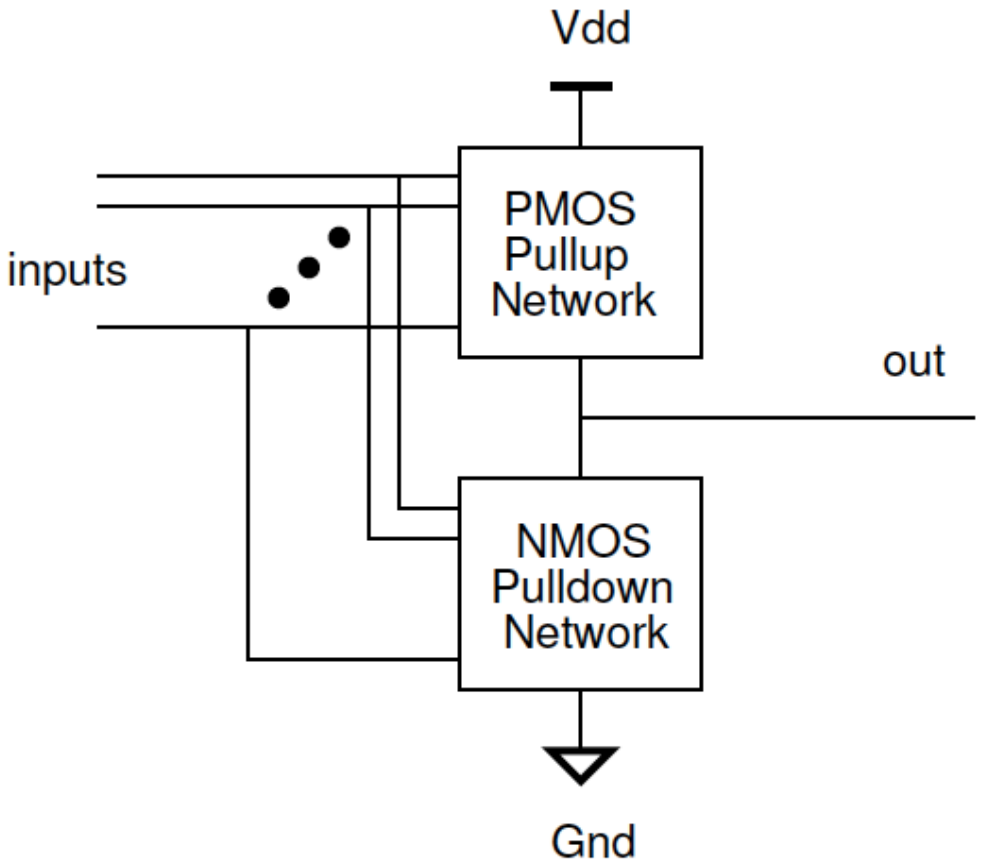




- Delay
  - Parasitic capacitances and resistances
- Dynamics
- Zeroeth Order captures behaviour if our circuit is not:
  - Capacitively loaded, acyclic (if there are Loops)

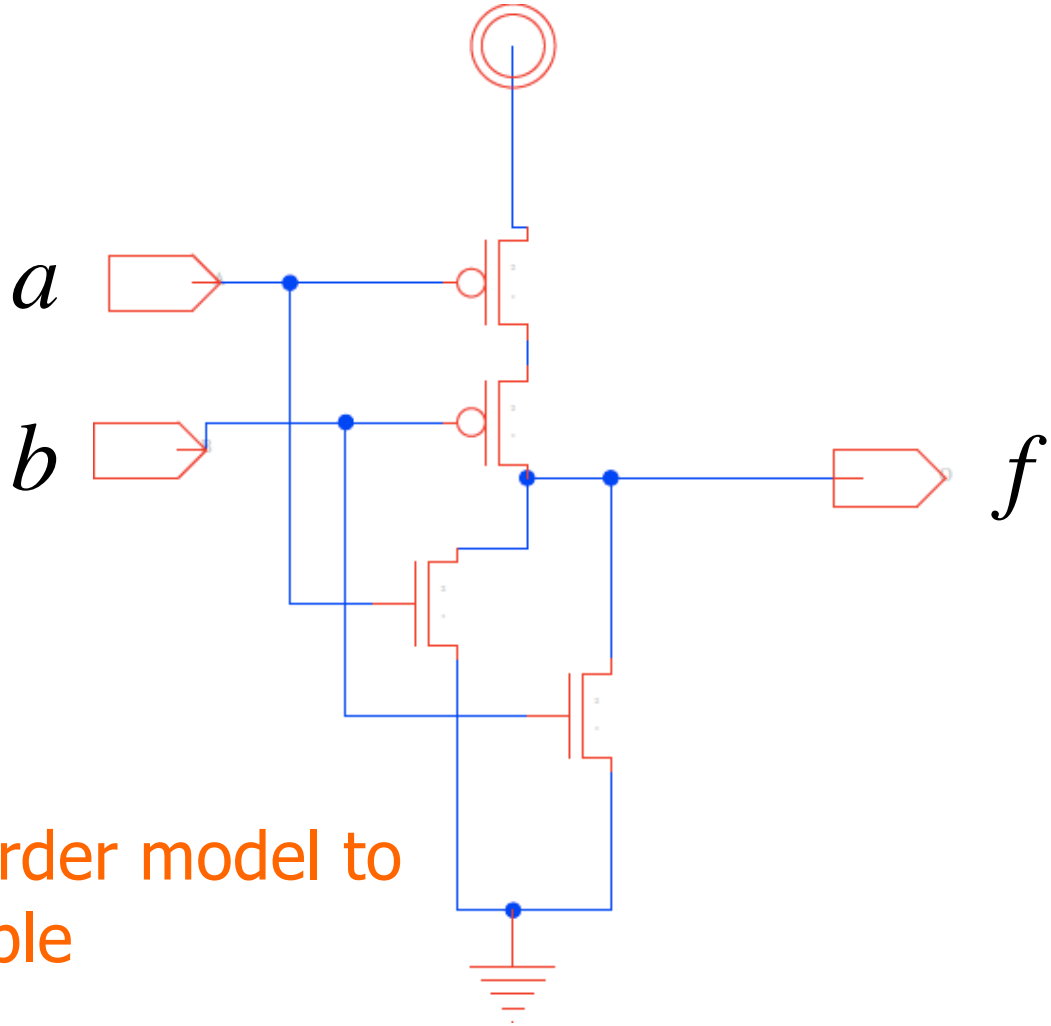
CMOS Gates

How to construct static CMOS gates



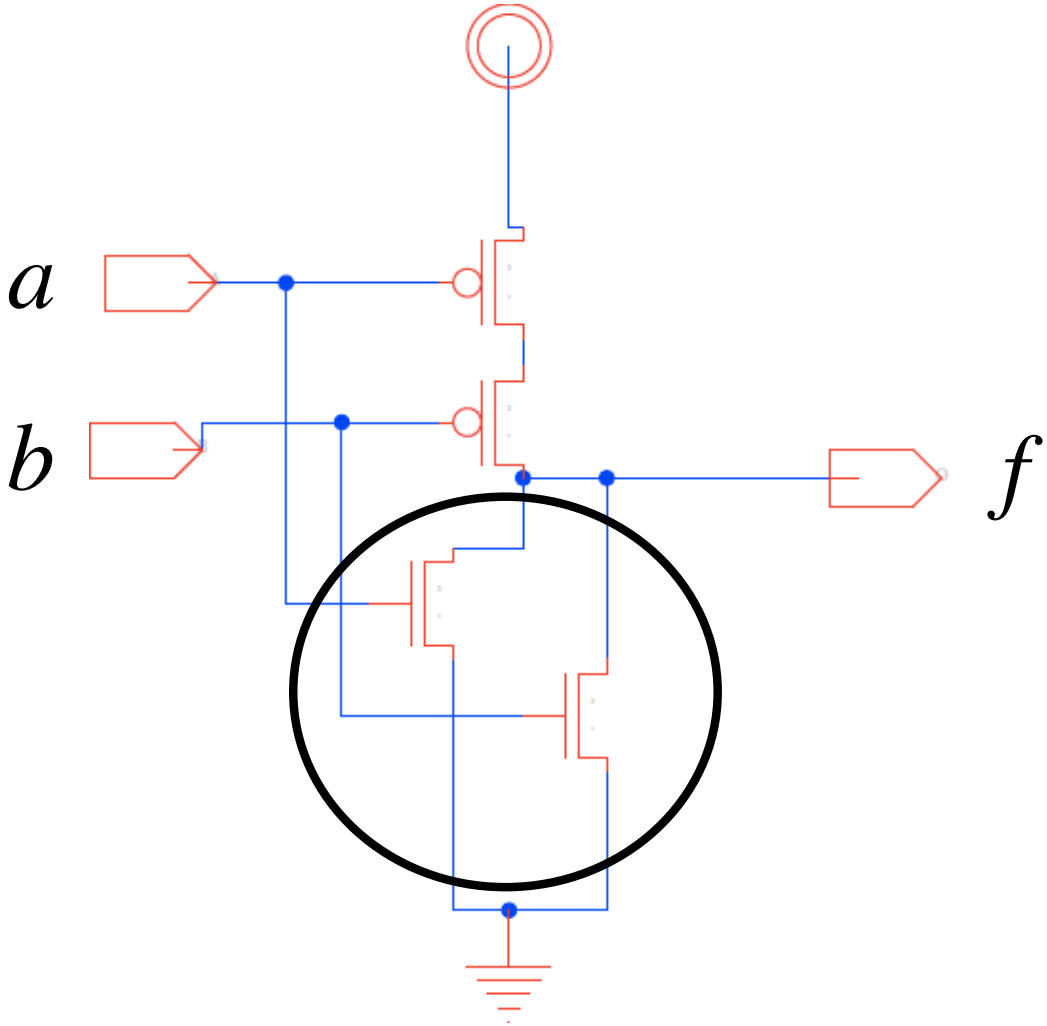
- Complementary Metal Oxide Semiconductor

What gate is this? Preclass 1

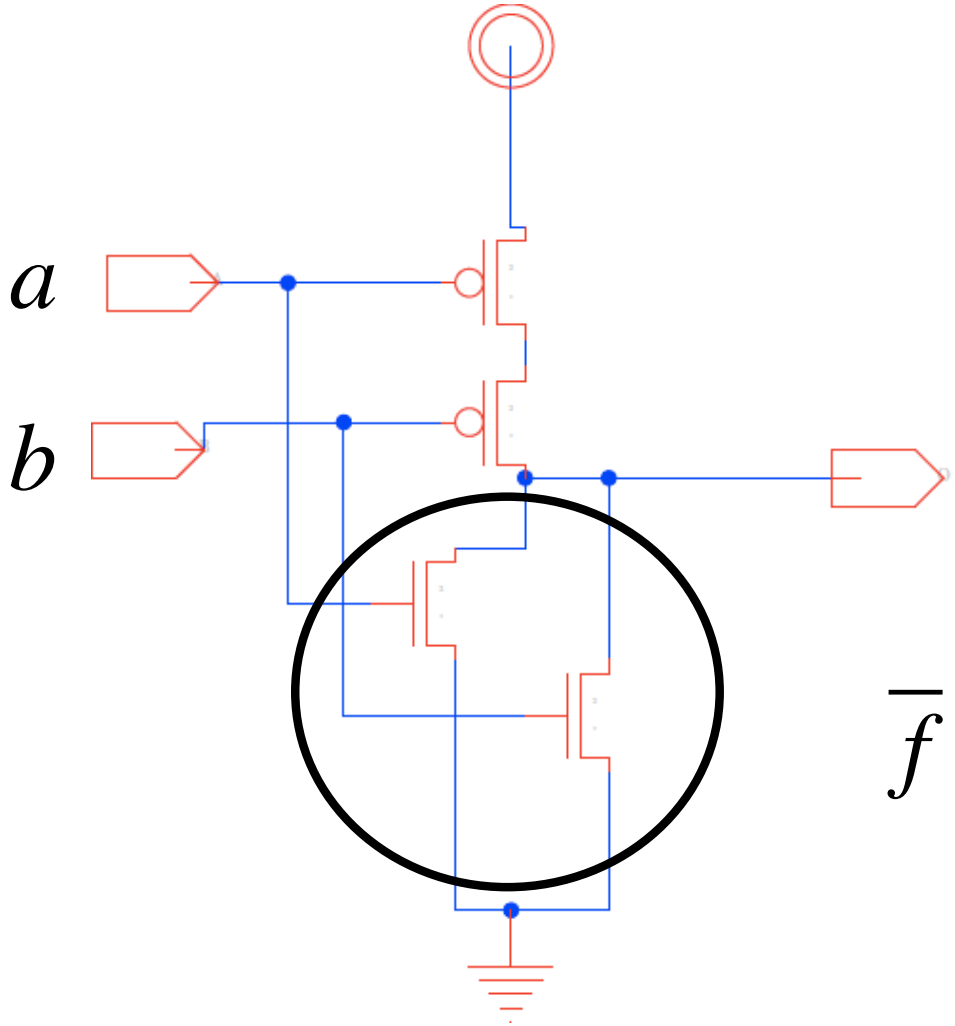


Hint: use zero order model to make a truth table

What gate is this? Preclass 1



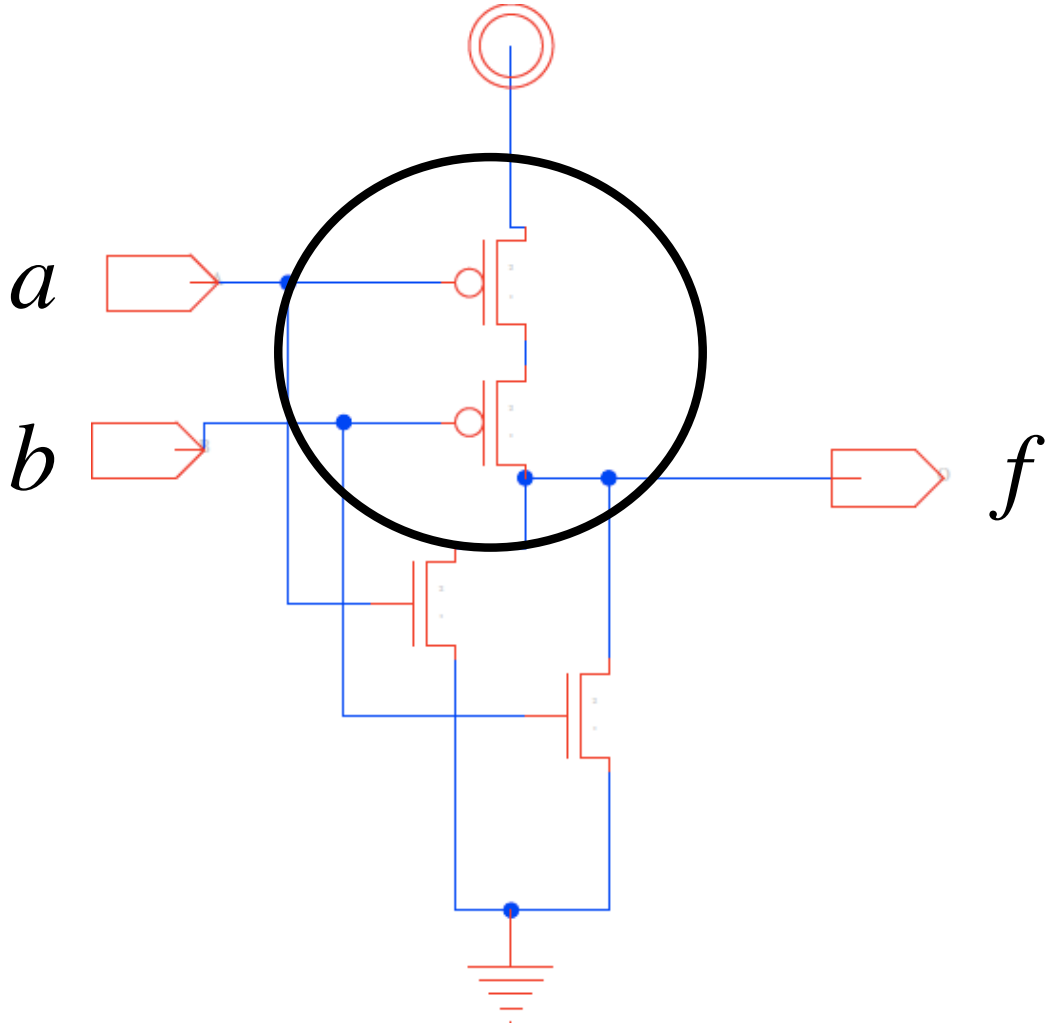
What gate is this? Preclass 2



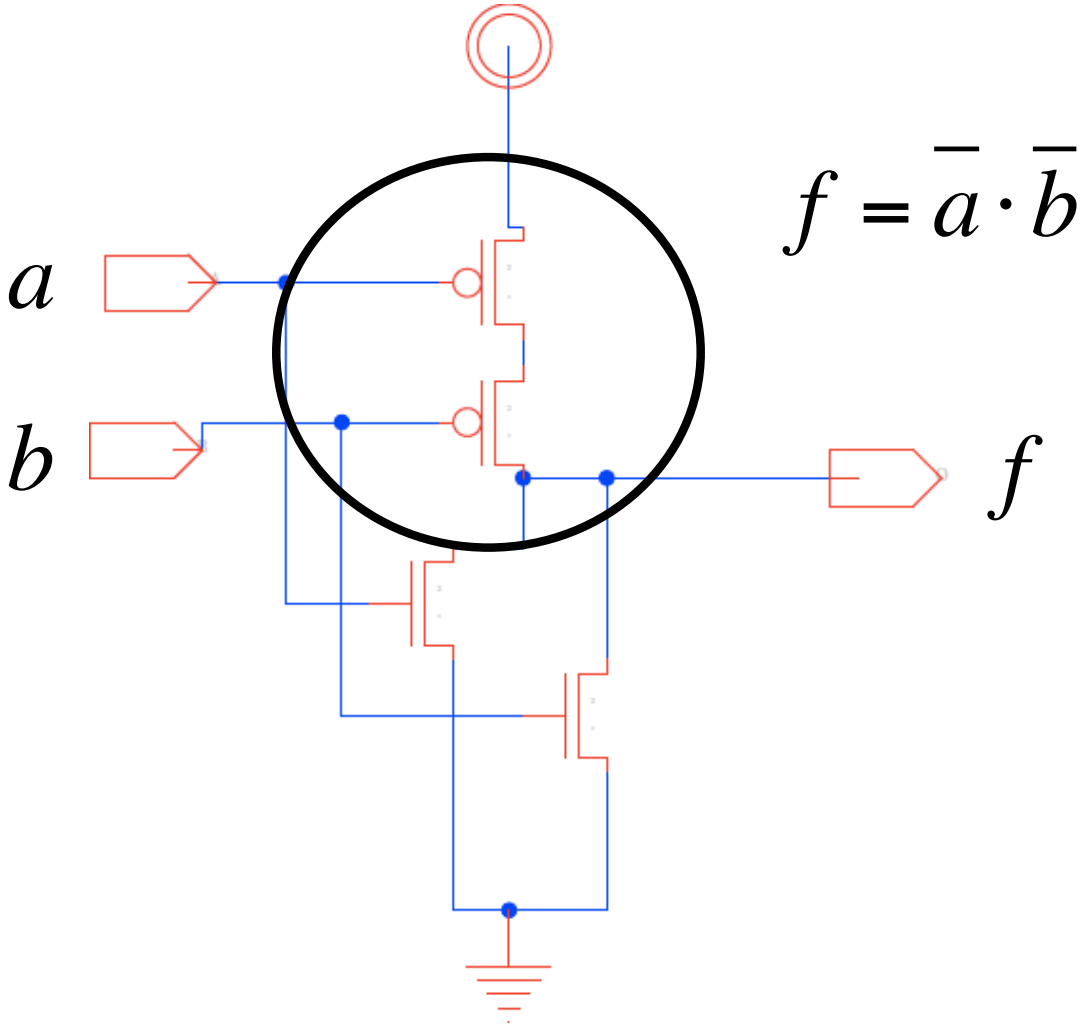
$$\overline{f} = a + b$$

What is f in minimum-sum-of-products form?

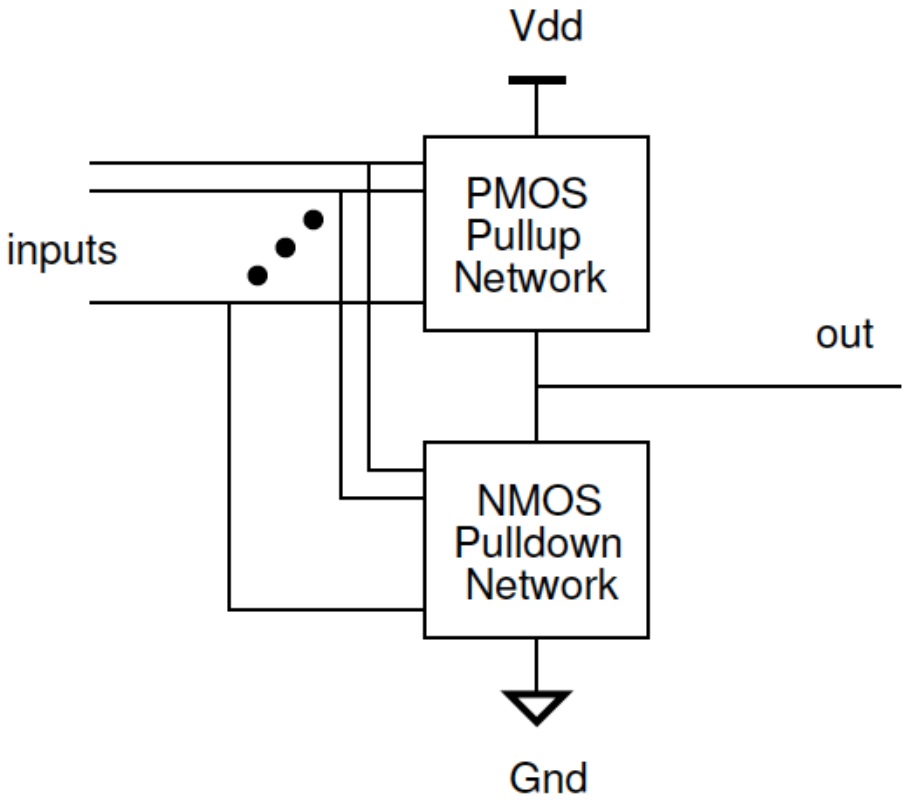
What gate is this? (Preclass 1)

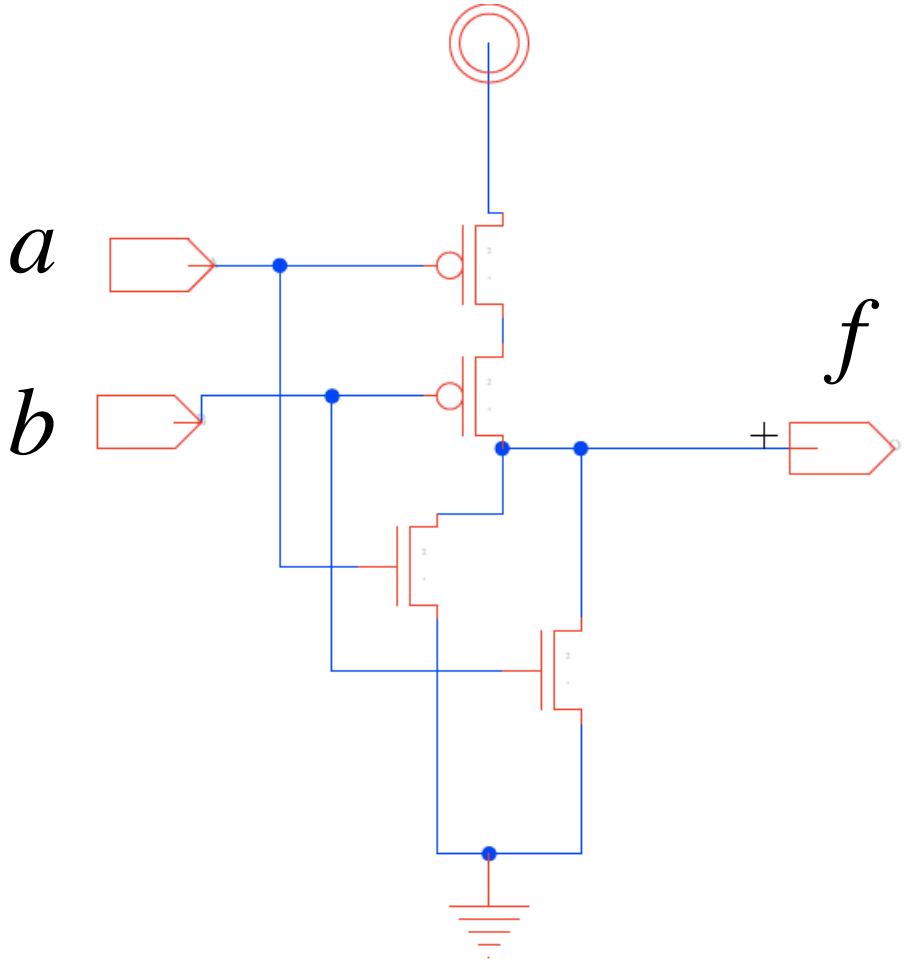


What gate is this? (Preclass 1)



Static CMOS Gate Structure

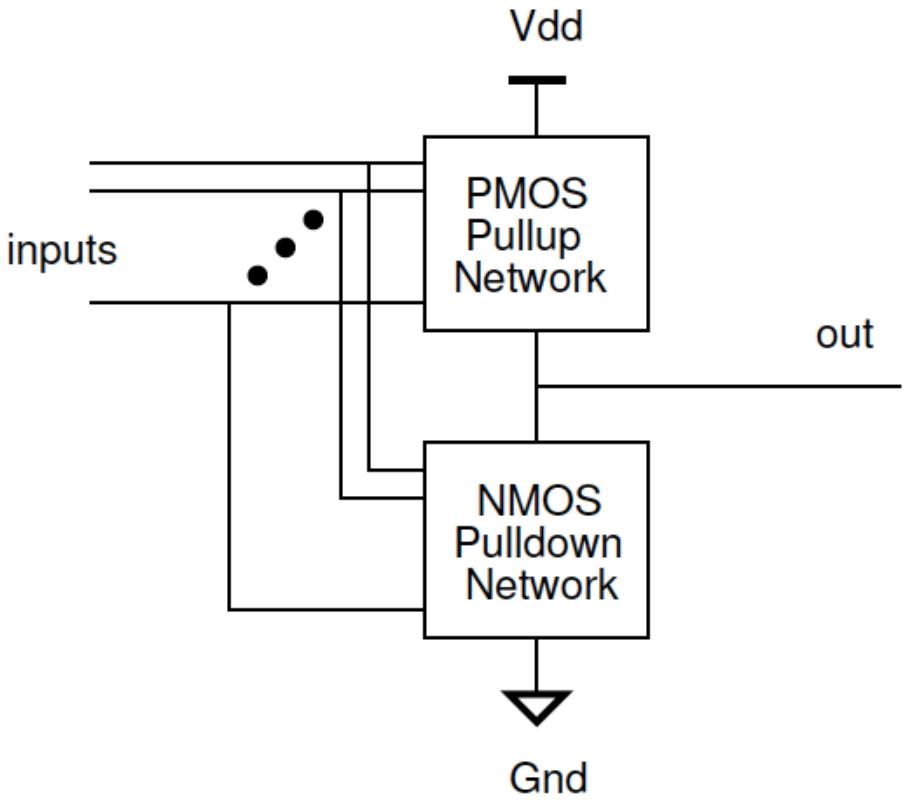




CMOS

- Complementary Metal Oxide Semicondutor
- Uses *complementary* transistors
  - NMOS, PMOS
- Pull-down and pull-up networks are *complements* of each other
  - Exactly one network active (on) at a time to charge or discharge output to V_{dd} or Gnd respectively

Static CMOS Gate Structure

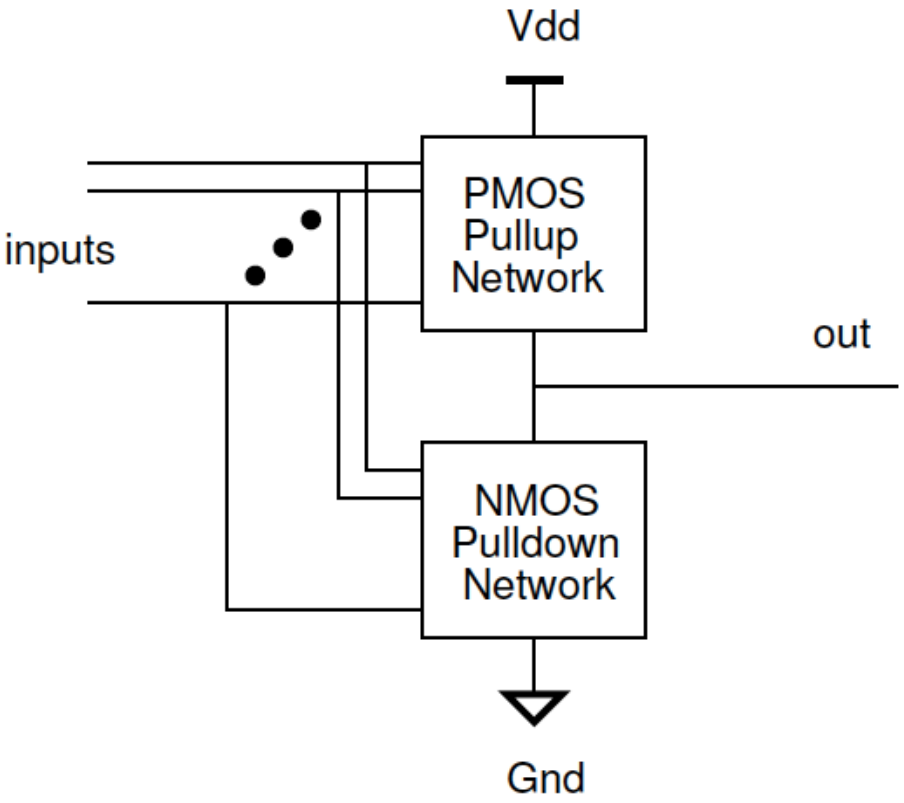


- Drives rail-to-rail
  - Power rails are V_{dd} and Gnd
  - output is V_{dd} or Gnd
- Input connects to gates → load is capacitive
- Once output node is charged doesn't use energy (no static current)
- Output actively driven

Gate Design Example Preclass 3

- Design gate to perform: $f = (\bar{a} + \bar{b}) \cdot \bar{c}$

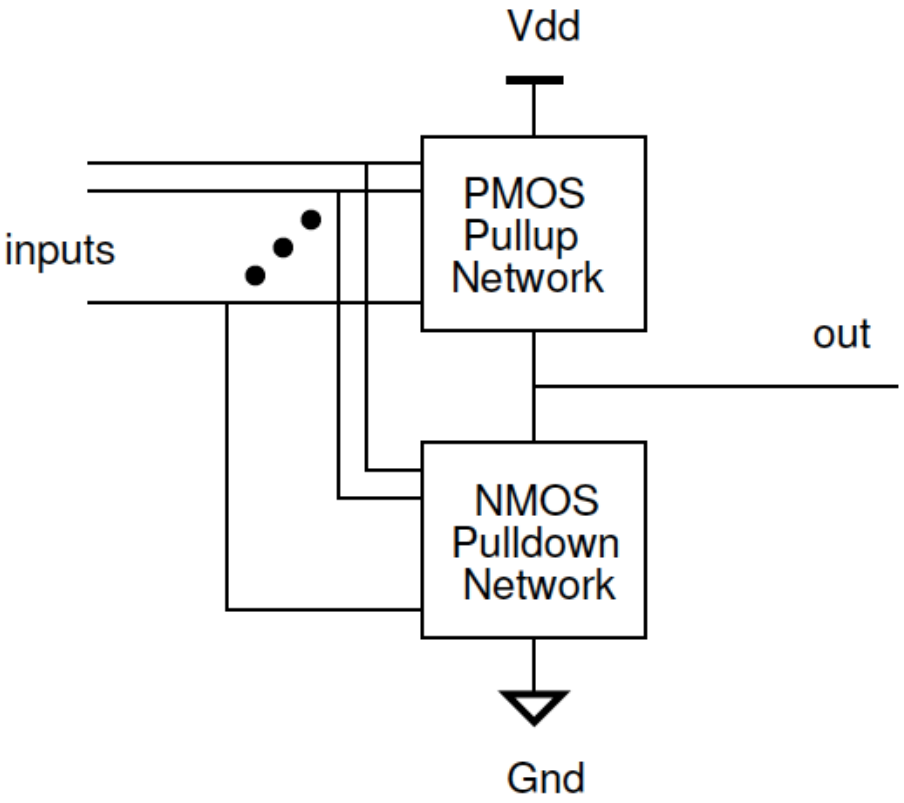
- Strategy:
  1. Use static CMOS structure
  2. Design PMOS pullup for *f*
  3. Use DeMorgan's Law to determine *f'*
  4. Design NMOS pulldown for *f'*



Gate Design Example

- Design gate to perform: $f = (\bar{a} + \bar{b}) \cdot \bar{c}$

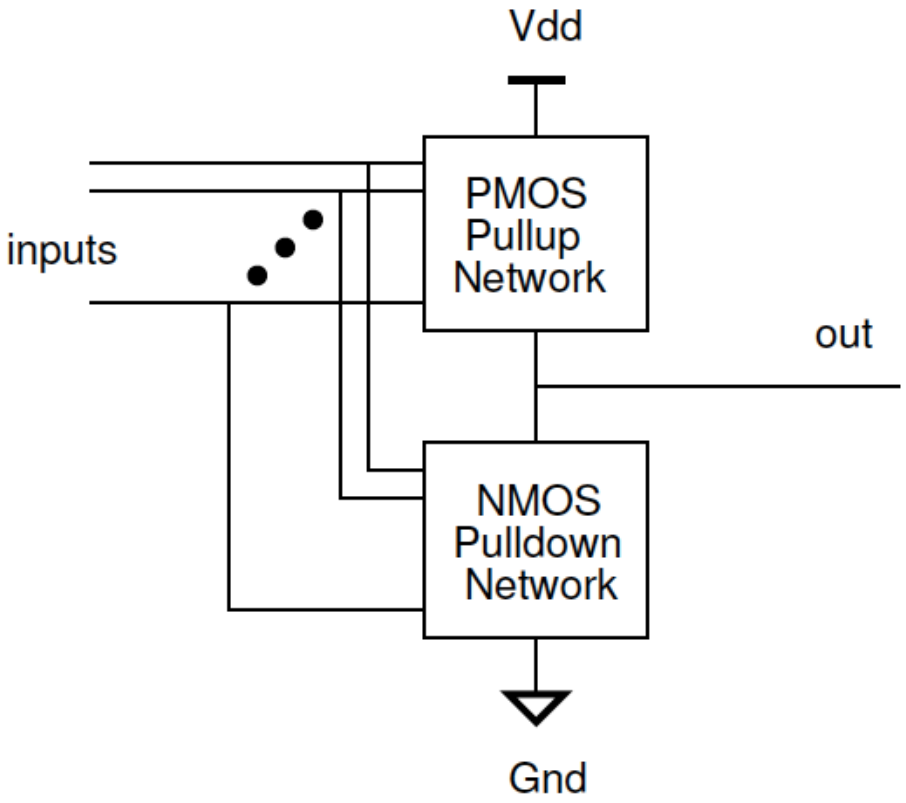
- PMOS Pullup for *f*?



Gate Design Example

- Design gate to perform: $f = (\bar{a} + \bar{b}) \cdot \bar{c}$

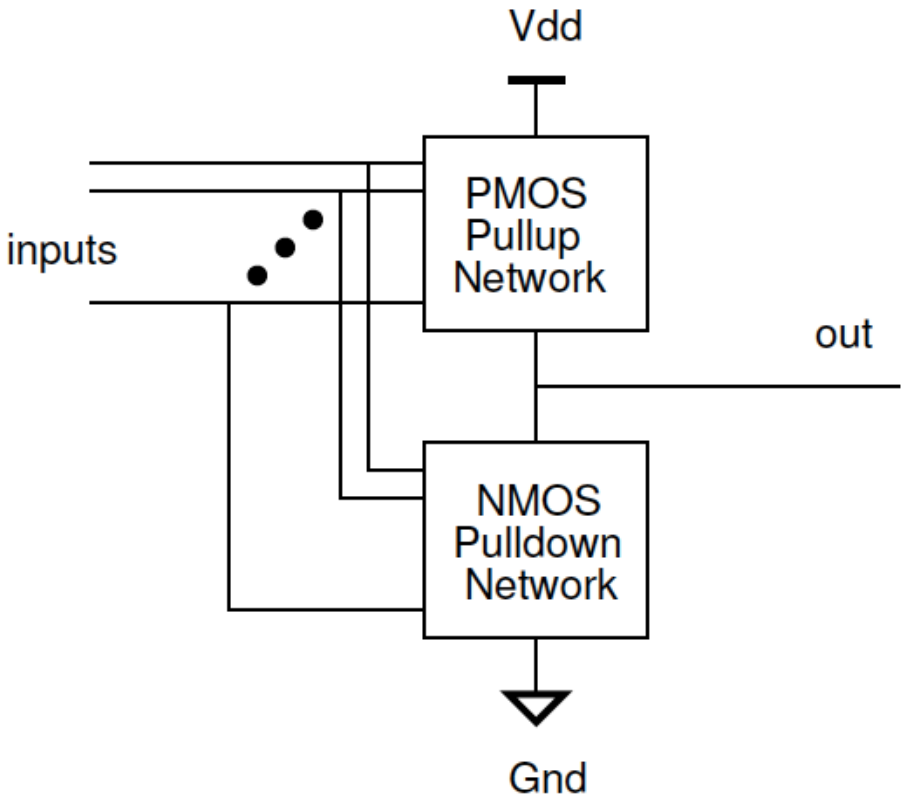
- What is *f'*?
  - DeMorgan's Law



Gate Design Example

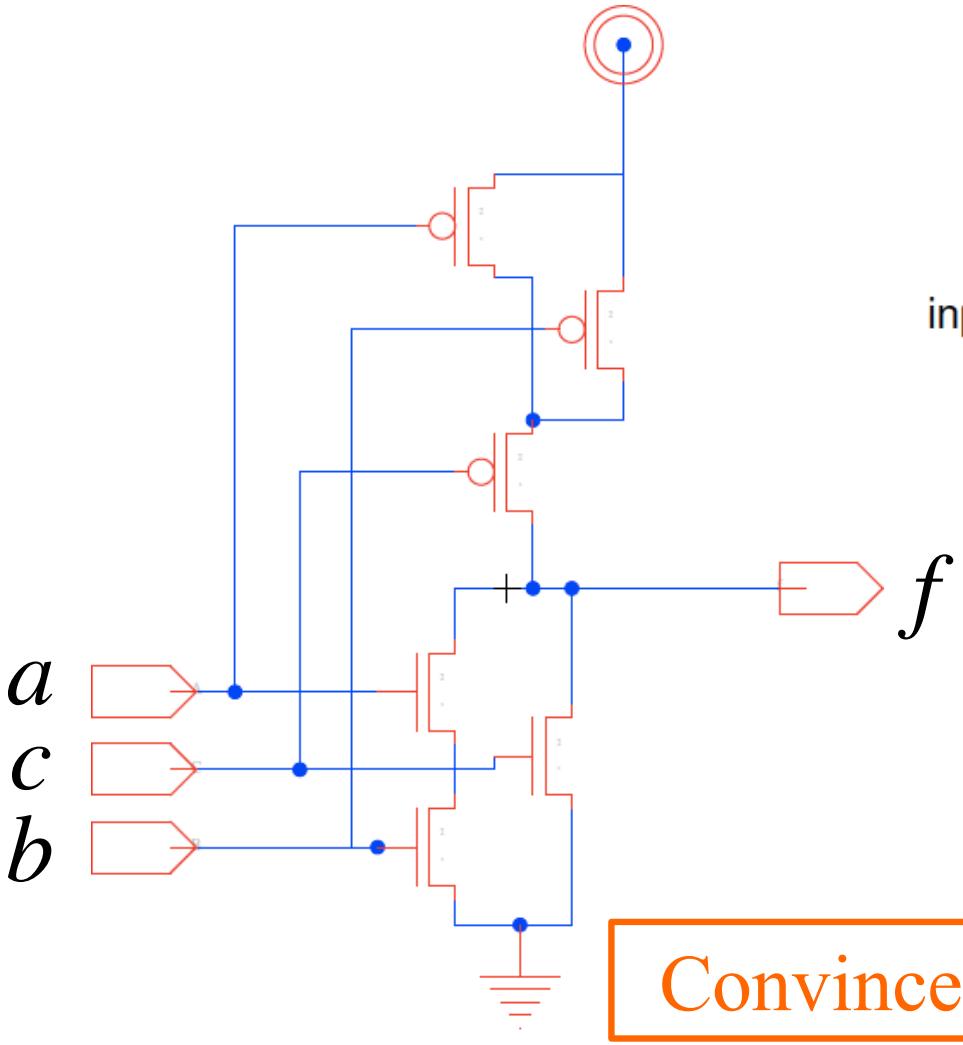
- Design gate to perform: $f = (\bar{a} + \bar{b}) \cdot \bar{c}$

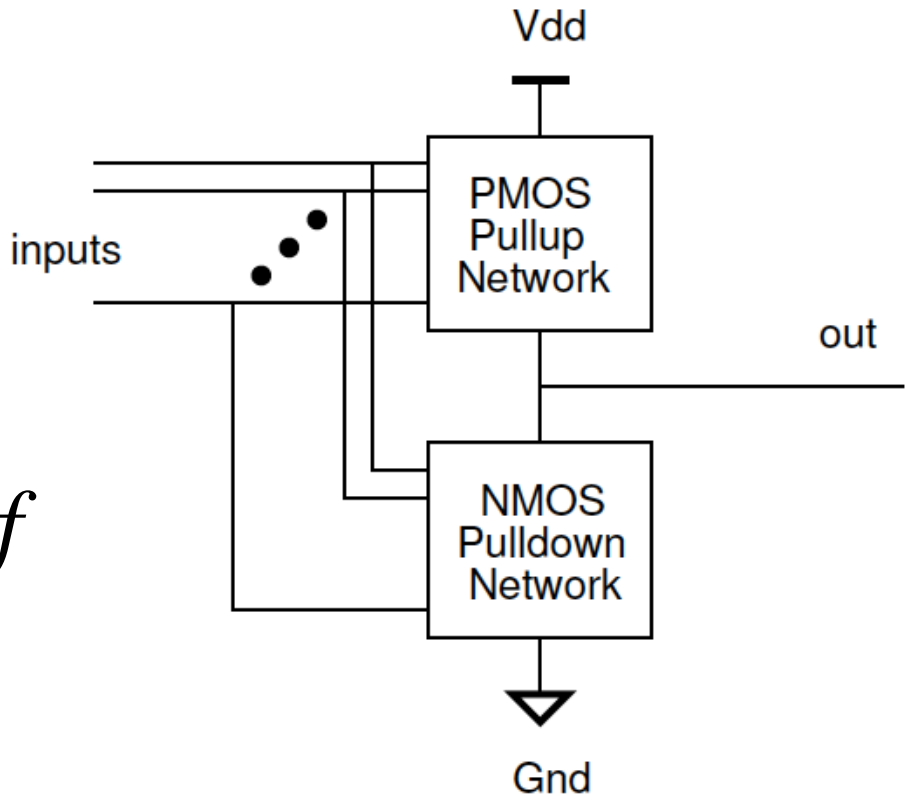
- Design NMOS pulldown for *f'*



Gate Design Example

- Design gate to perform: $f = (\overline{a} + \overline{b}) \cdot \overline{c}$

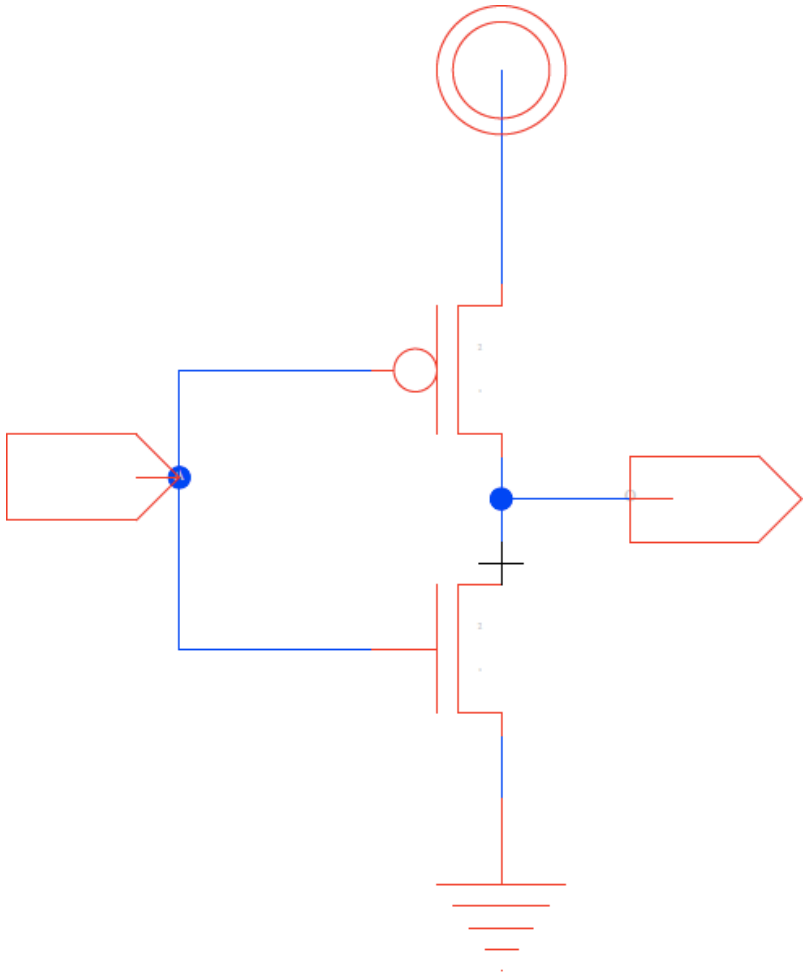




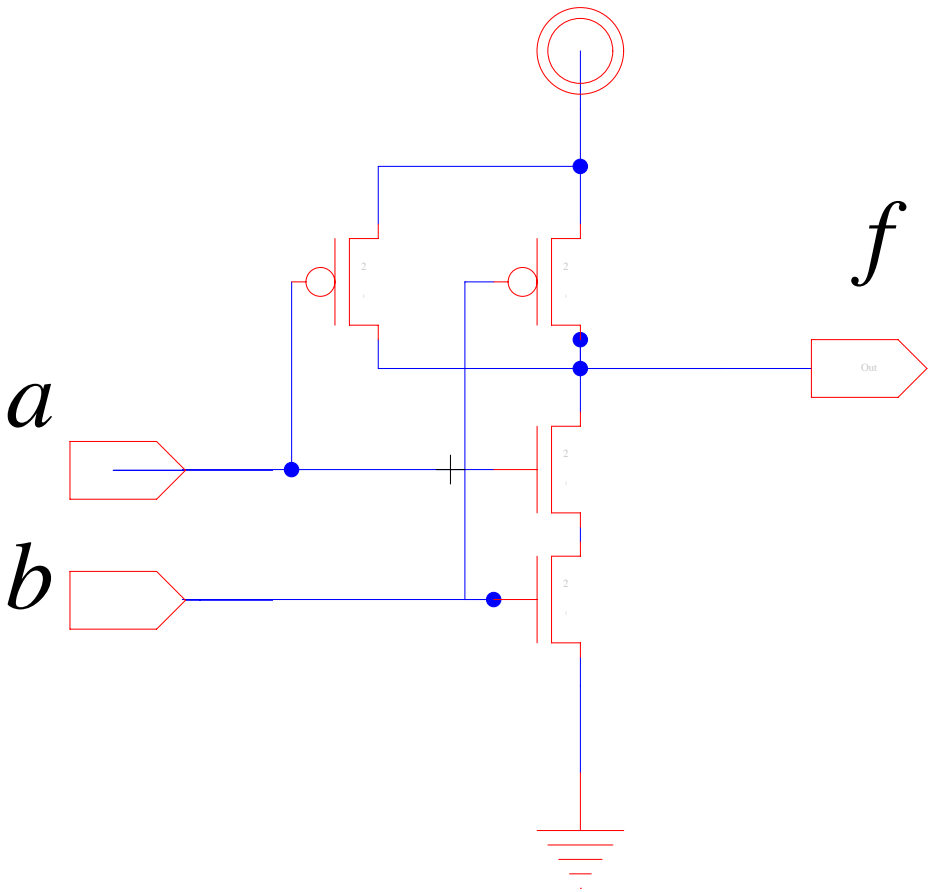
Convince yourself with a truth table.

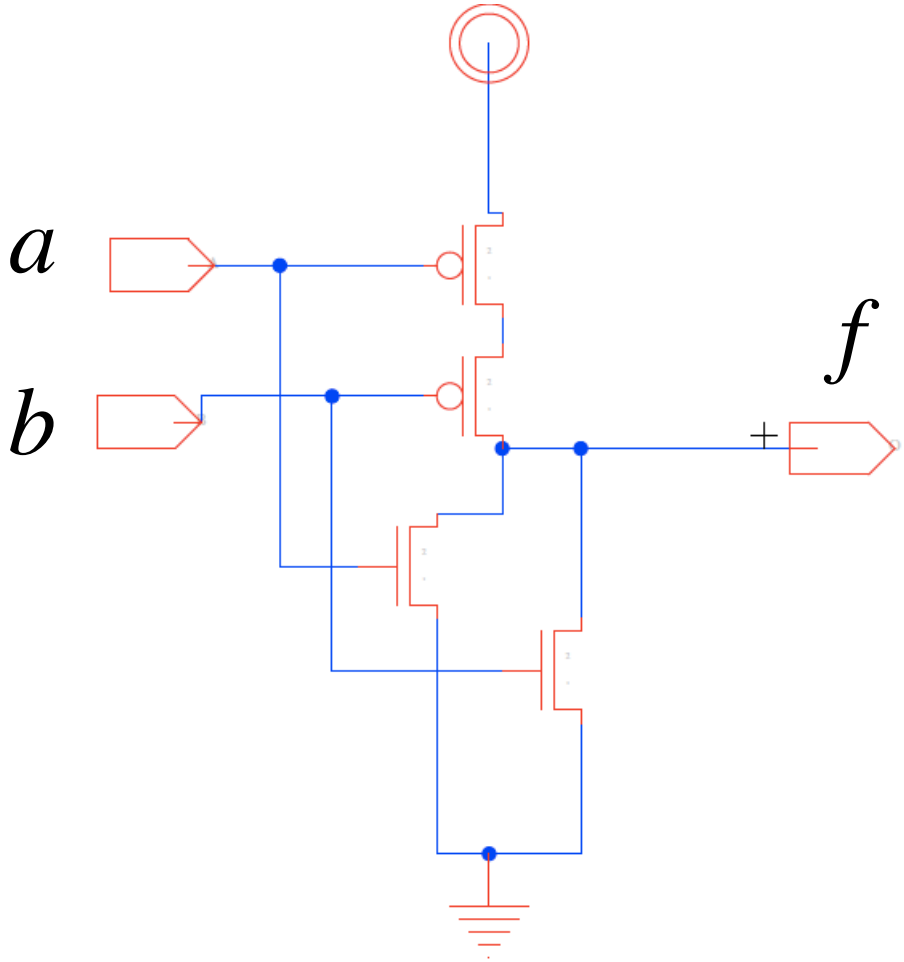
Inverting Stage

- Each stage of Static CMOS gate is inherently inverting



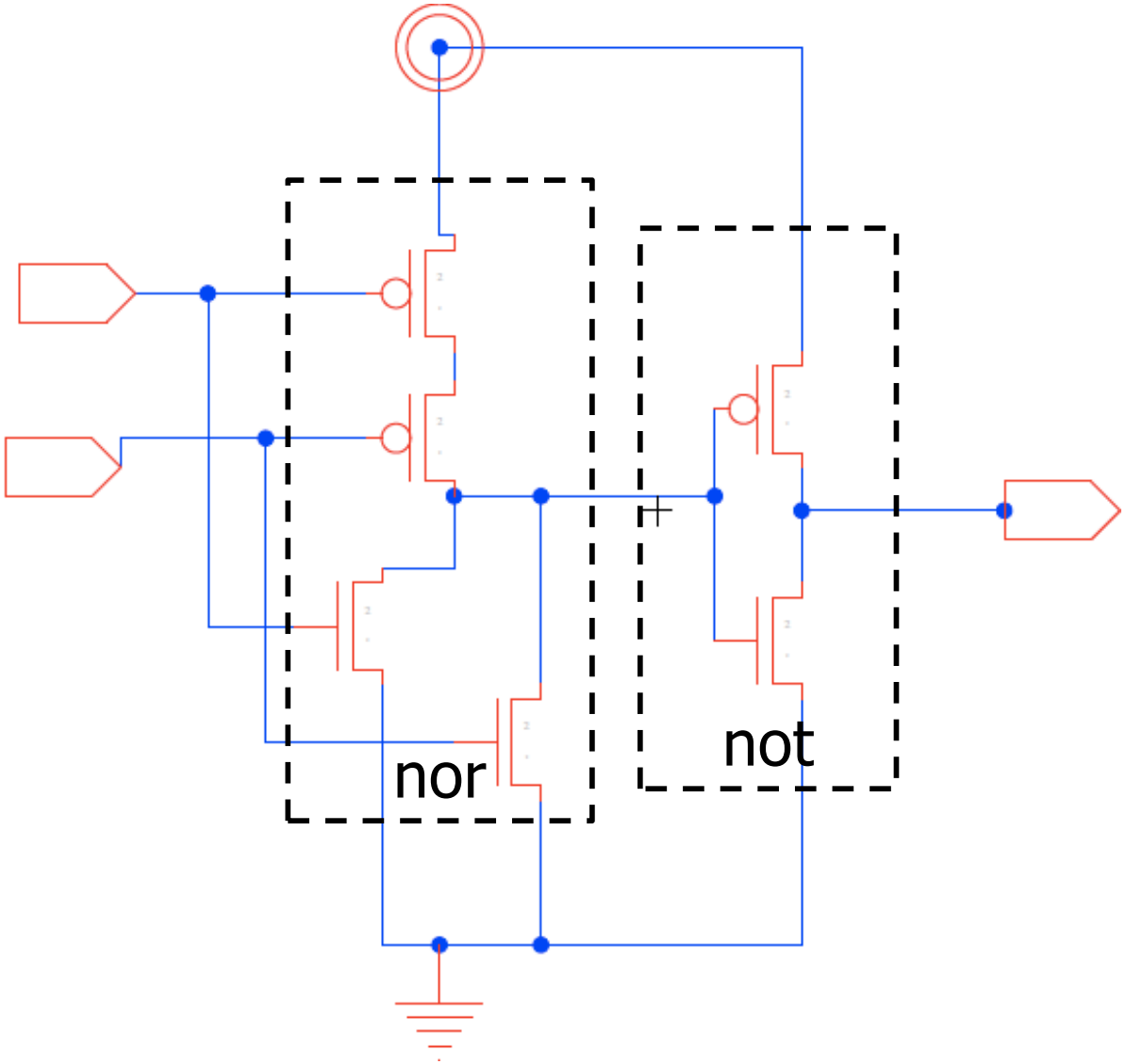
NAND/NOR Fundamental Gates





How implement OR?

How implement OR?



Cascading Stages

- Can always cascade "stages" to build more complex gates
- Could simply build nor2 or nand2 at circuit level and assemble arbitrary logic by combining – universality
  - but may not be smallest/fastest/least power

Implement: $f = a \cdot \bar{b}$

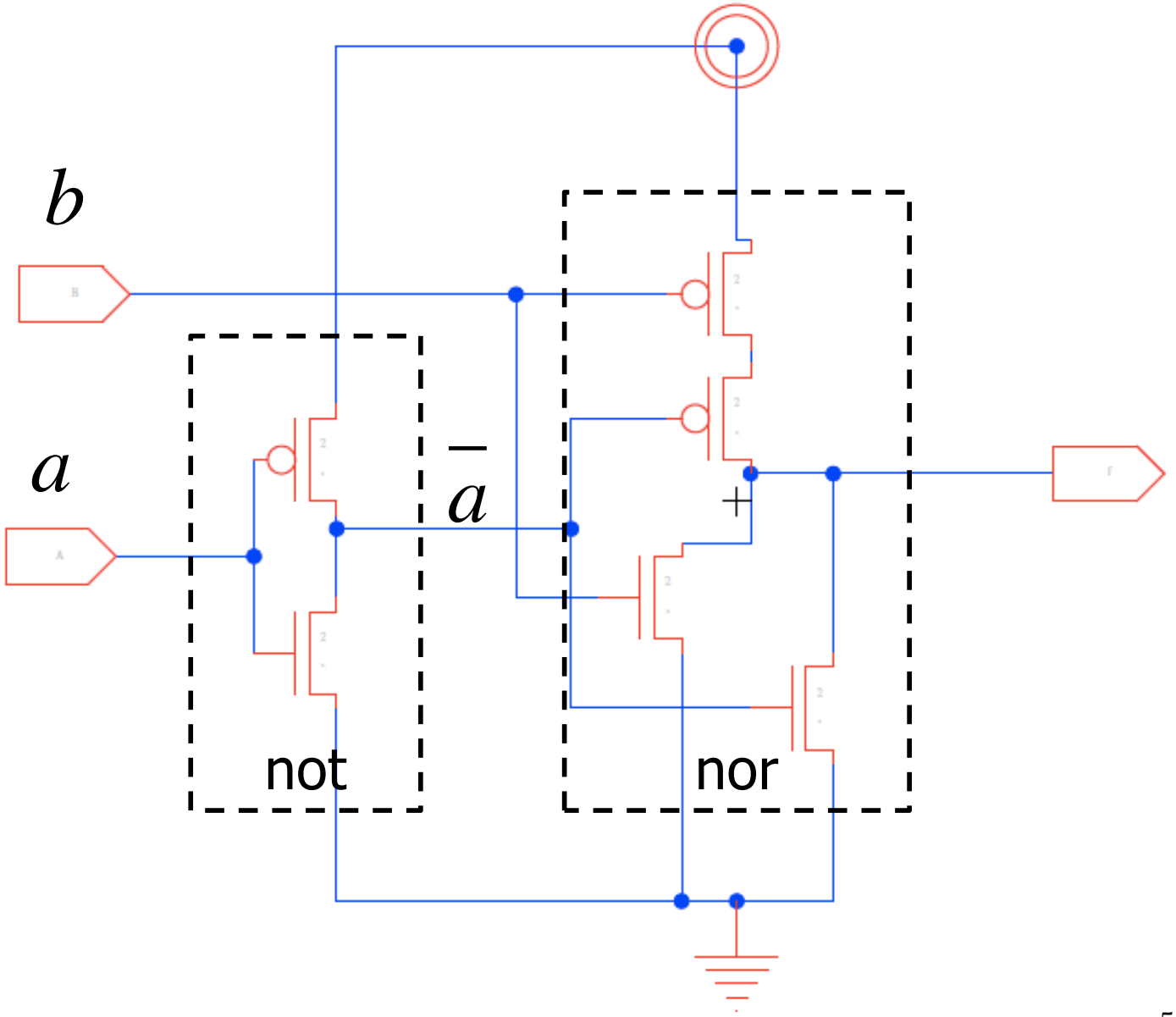
- Pullup?

- Pulldown?

Hint: use cascading stages

Implement: $f = a \cdot \bar{b}$

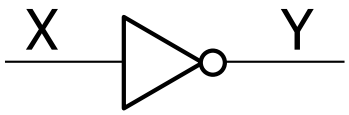
- Pullup?
- Pulldown?



Digital Logic

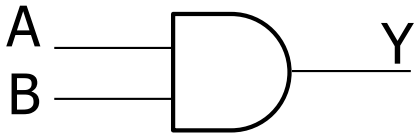
Basic Digital Gates

NOT
$Y = \overline{X}$



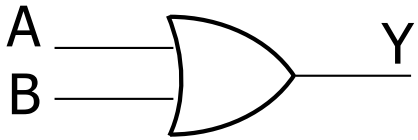
X	Y
0	1
1	0

AND
Y = A•B



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

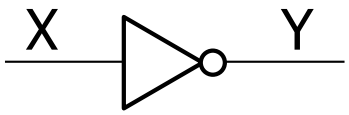
OR
Y = A+B



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

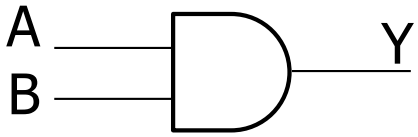
Basic Digital Gates

NOT
$Y = \overline{X}$



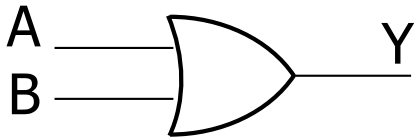
X	Y
0	1
1	0

AND
Y = A•B



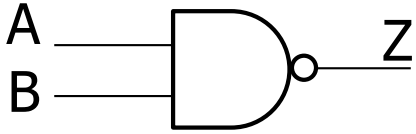
A	B	Y	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

OR
Y = A+B

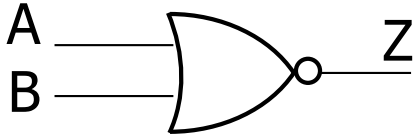


A	B	Y	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

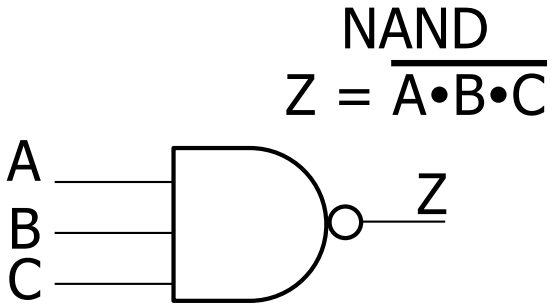
NAND
$Z = \overline{A \cdot B}$



NOR
$Z = \overline{A+B}$



Basic Digital Gates



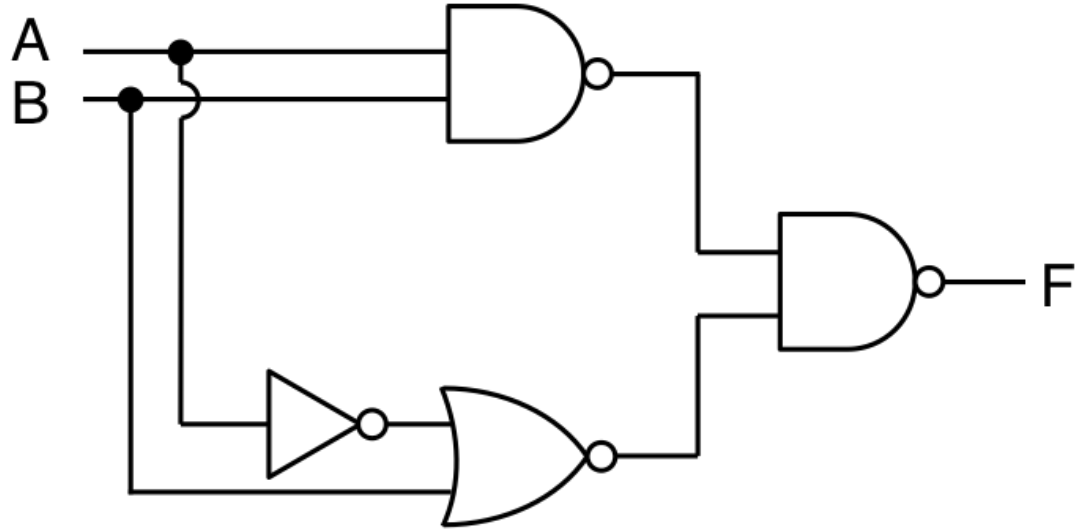
A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Boolean Algebra

TABLE 2-3
Basic Identities of Boolean Algebra

1.	$X+0=X$	2.	$X\cdot 1=X$	
3.	$X+1=1$	4.	$X\cdot 0=0$	
5.	$X+X=X$	6.	$X\cdot X=X$	
7.	$X+\overline{X}=1$	8.	$X\cdot\overline{X}=0$	
9.	$\overline{\overline{X}}=X$			
10.	$X+Y=Y+X$	11.	$XY=YX$	Commutative
12.	$X+(Y+Z)=(X+Y)+Z$	13.	$X(YZ)=(XY)Z$	Associative
14.	$X(Y+Z)=XY+XZ$	15.	$X+YZ=(X+Y)(X+Z)$	Distributive
16.	$\overline{X+Y}=\overline{X}\cdot\overline{Y}$	17.	$\overline{X\cdot Y}=\overline{X}+\overline{Y}$	DeMorgan's

Combination



Boolean Expressions

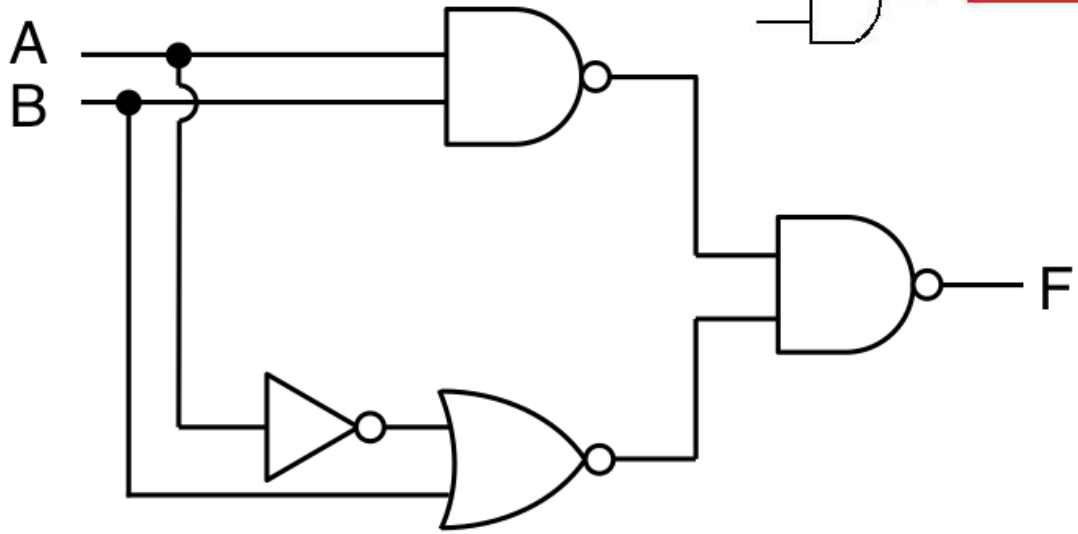
- Sum-of-products form (SOP)
  - Eg. ABC+DEF+GHI
- Product-of-sums form (POS)
  - Eg. (A+B+C)(D+E+F)(G+H+I)

- Convert between the two with Boolean algebra
  - DeMorgan's Law

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

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

Combination



Canonical Form

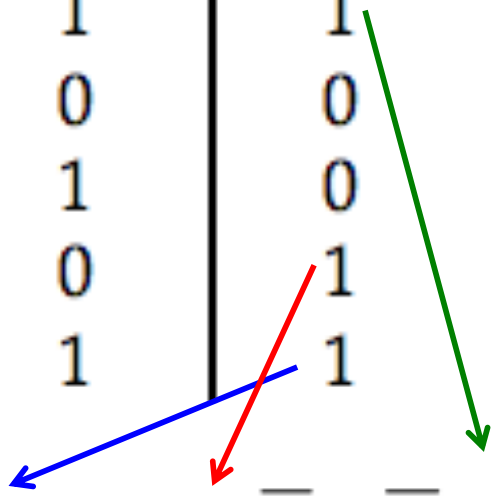
- Every minterm in your expression has every variable

Row Number	A	B	C	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Canonical Form

- Every minterm in your expression has every variable

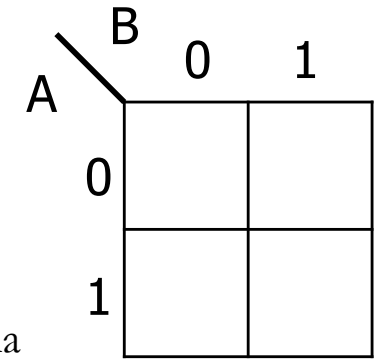
Row Number	A	B	C	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

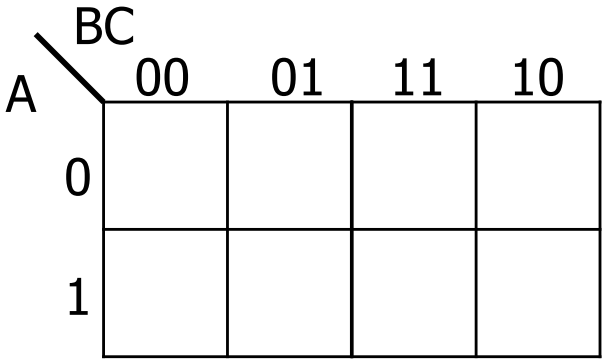


$$f(A,B,C) = ABC + AB\overline{C} + \overline{A}BC$$

What is a K(arnaugh)-map?

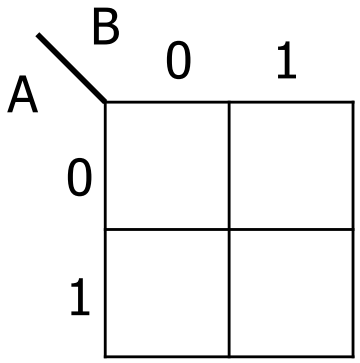
- A grid of squares (representing truth table)
- Each square represents one minterm
- The minterms are ordered according to **Gray code**
  - Only one variable changes between adjacent squares
- Squares on edges are considered adjacent to squares on opposite edges
  - I.e Table wraps around
- K-maps are clumsy with more than 4 variables





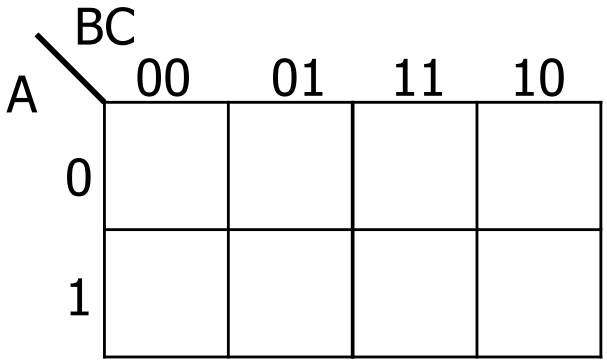
K-map Examples (Preclass 4)

- 2-variable



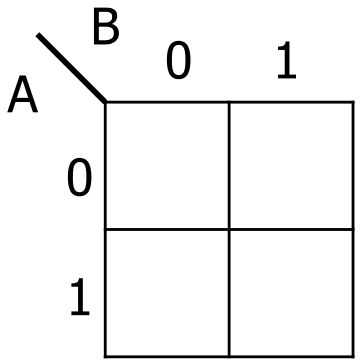
Eg: Z = A′B′ + AB′ + A′B

- 3-variable



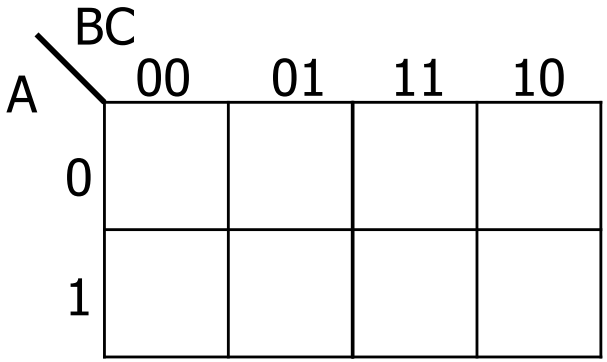
K-map Examples (Preclass 5)

- 2-variable



Eg: Z = A′B′ + AB′ + A′B

- 3-variable



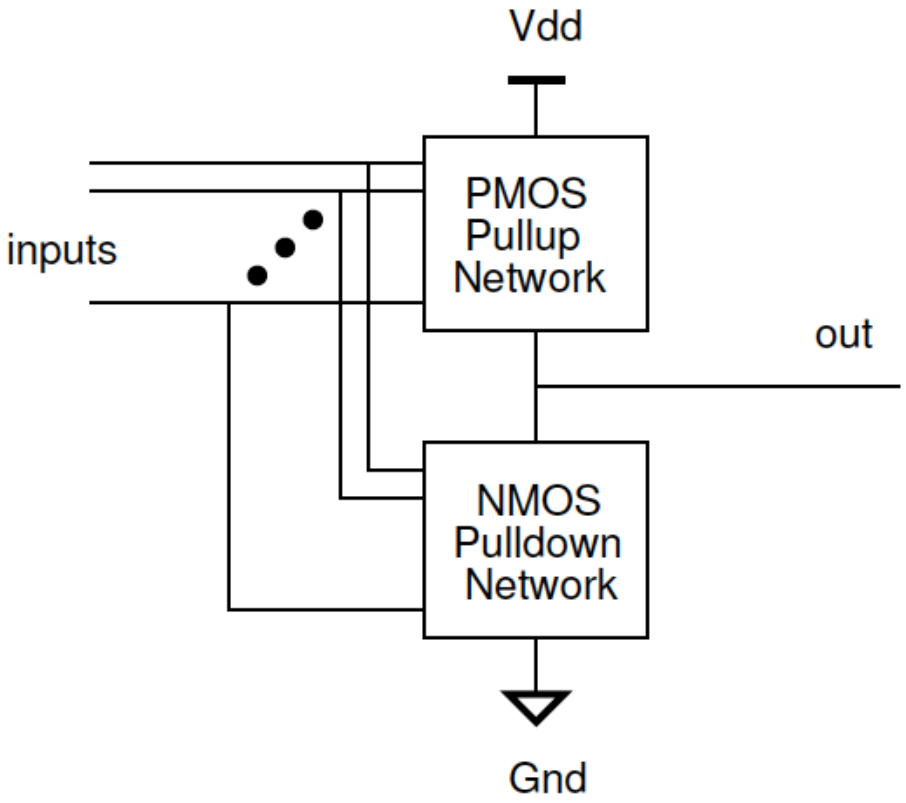
Eg: Z = A′B′C′ + A′B + ABC′ + AC

Big Idea

- MOSFET Transistor as switch
- Functionality-driven simplified modeling (Zero order)
  - Aid reasoning
  - Sanity check
  - Simplify design
- CMOS Gate Design
  - Complementary pull-up and pull-down networks

Big Idea

- Systematic construction of any gate from transistors
  1. Use static CMOS structure
  2. Design PMOS pullup for *f*
  3. Use DeMorgan's Law to determine *f'*
  4. Design NMOS pulldown for *f'*



Admin

- Diagnostic Quiz (in Canvas) – due by F 1/26
  - Review as needed
- Monday 1/29 – no lecture, Lab in Detkin
- HW 1 – due F 2/2
  - Need lab and future lectures to finish
  - Submit in Gradescope
    - Demo on Gradescope in Class Recordings on Canvas
- TA office hours announced on Ed Discussion
  - Still working on permanent locations

Grace Hopper Lecture

- Thursday 3-4pm in Glandt Forum (Singh Center)

ESE Grace Hopper Lecture – "Disrupting NextG"

January 25 at 3:00 PM - 4:00 PM

As 5G takes to the airwaves, we now turn our imagination to the next generation of wireless technology. The promise of this technology has created an international race to innovate, with significant investment by government as well as industry. And much innovation is needed as 6G aspires to not only support significantly higher data rates than 5G, but also improved reliability along with excellent coverage indoors and out, including for underserved areas. New architectures including edge computing must be designed to drastically enhance efficient resource allocation while also reducing latency for real-time control. Breakthrough energy-efficiency architectures, algorithms and hardware will be needed so that wireless devices can be powered by tiny batteries, energy-harvesting, or over-the-air power transfer. And machine learning may will play a big role in the underlying technologies for NextG as well as the "killer apps" that will drive its deployment and success. This talk will describe what the wireless future might look like along with some of the innovations and breakthroughs required to realize this vision.

Andrea Goldsmith
Dean of the School of Engineering and Applied Science & The Arthur LeGrand Doty Professor of Electrical and Computer Engineering, Princeton University

Andrea Goldsmith is the Dean of Engineering and Applied Science and the Arthur LeGrand Doty Professor of Electrical and Computer Engineering at Princeton University. She was previously the Stephen Harris Professor of Engineering and Professor of Electrical Engineering at Stanford University, where she is now Harris Professor Emerita. Her research interests are in information theory, communication theory, and signal processing, and their application to wireless communications, interconnected systems, and biomedical devices. She founded and served as Chief Technical Officer of Plume WiFi (formerly Accelera, Inc.) and of Quantenna (QTNA), Inc, and she serves on the Board of Directors for Intel (INTC), Medtronic (MDT), Crown Castle Inc (CCI), and the Marconi Society. She also serves on the Presidential Council of Advisors on Science and Technology (PCAST). Dr. Goldsmith is a

https://events.seas.upenn.edu/event/ese-grace-hopper-lecture-title-tbd/

Acknowledgement

- Prof. André DeHon (University of Pennsylvania)
- Prof. Jing Li (University of Pennsylvania)