

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

---

Lec 2: January 22, 2025

Transistor Introduction and  
Gates from Transistors



# Today

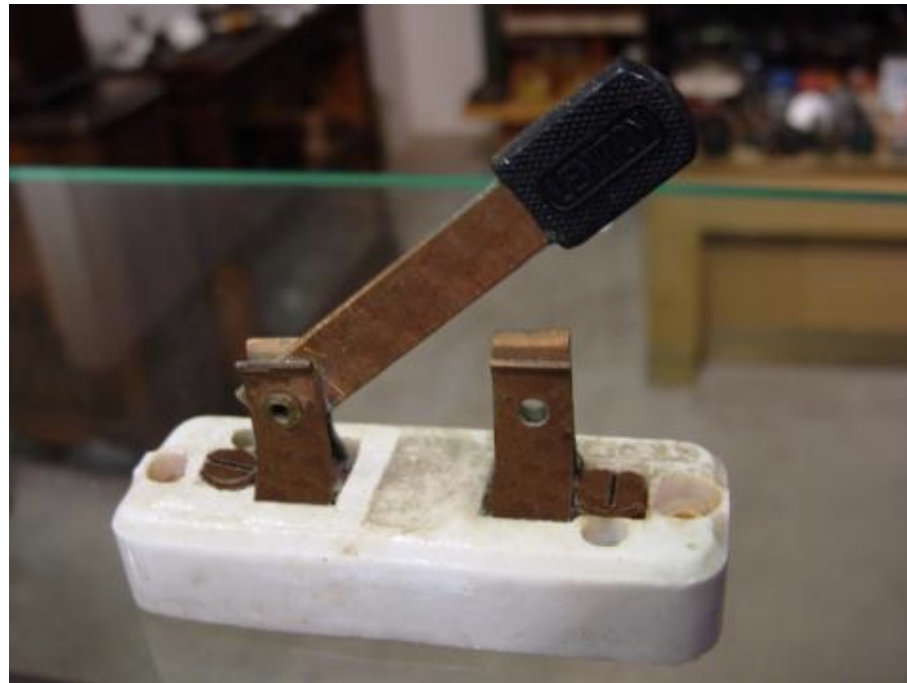
---

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

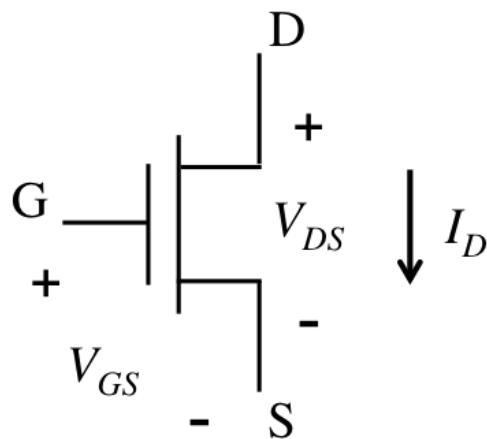
# Transistor

---

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



# MOSFET



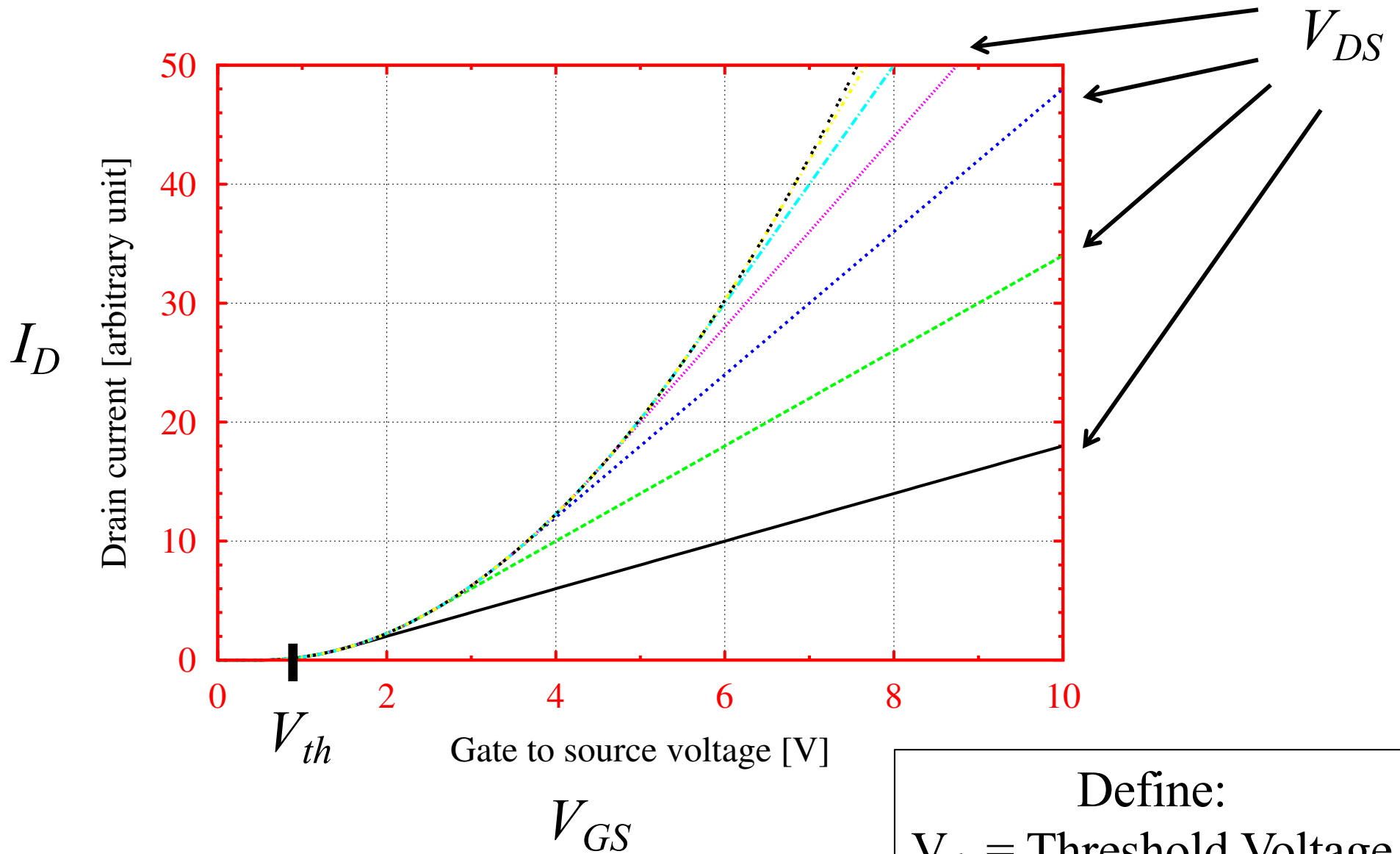
$$V_{GS} = V_G - V_S$$

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

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

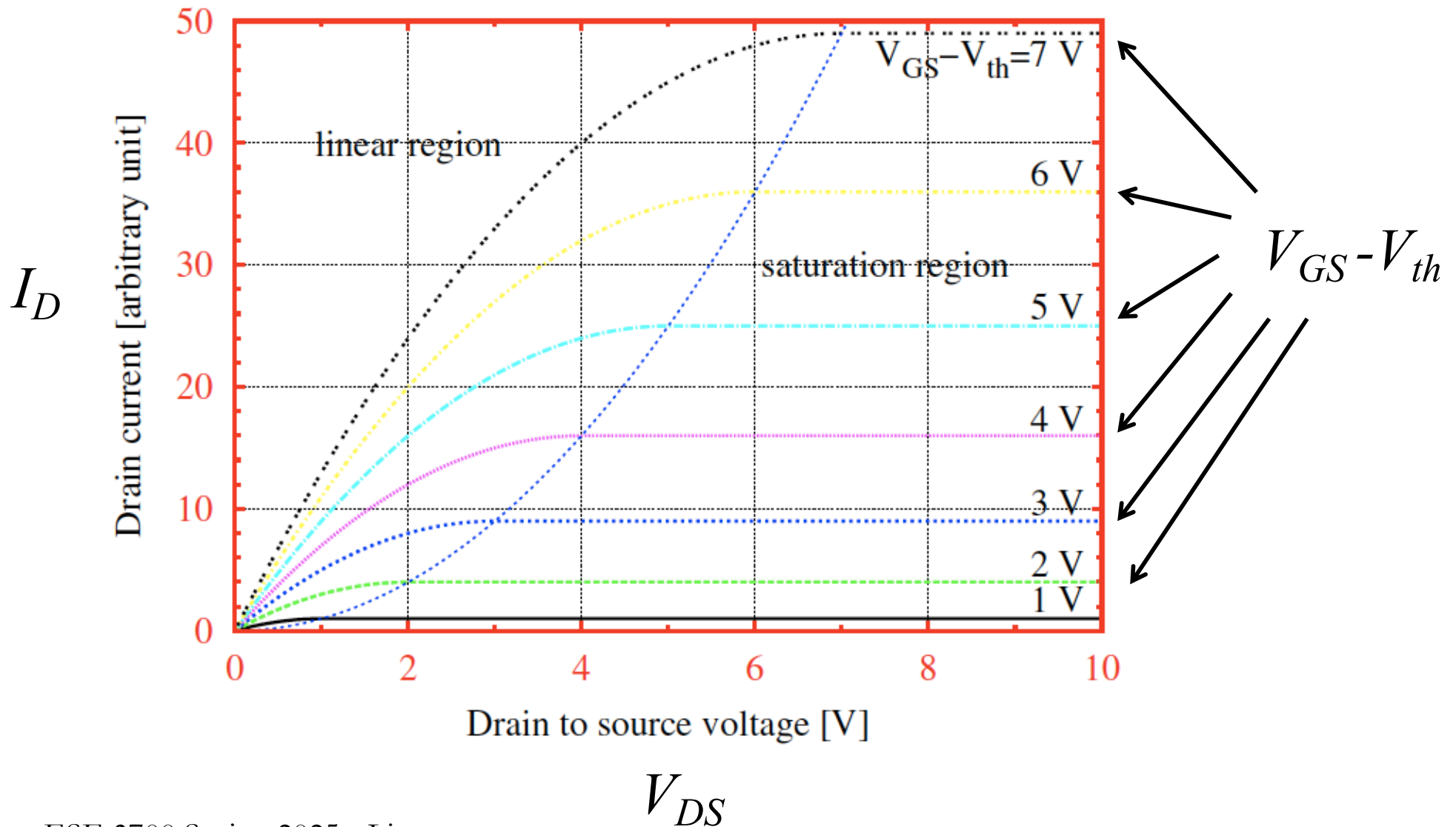
- Metal Oxide Semiconductor Field Effect Transistor
  - Primary **active** component for the term
  - Three terminal device
    - Voltage at gate controls conduction between two other terminals (source, drain)

# MOSFET – IV Characteristics

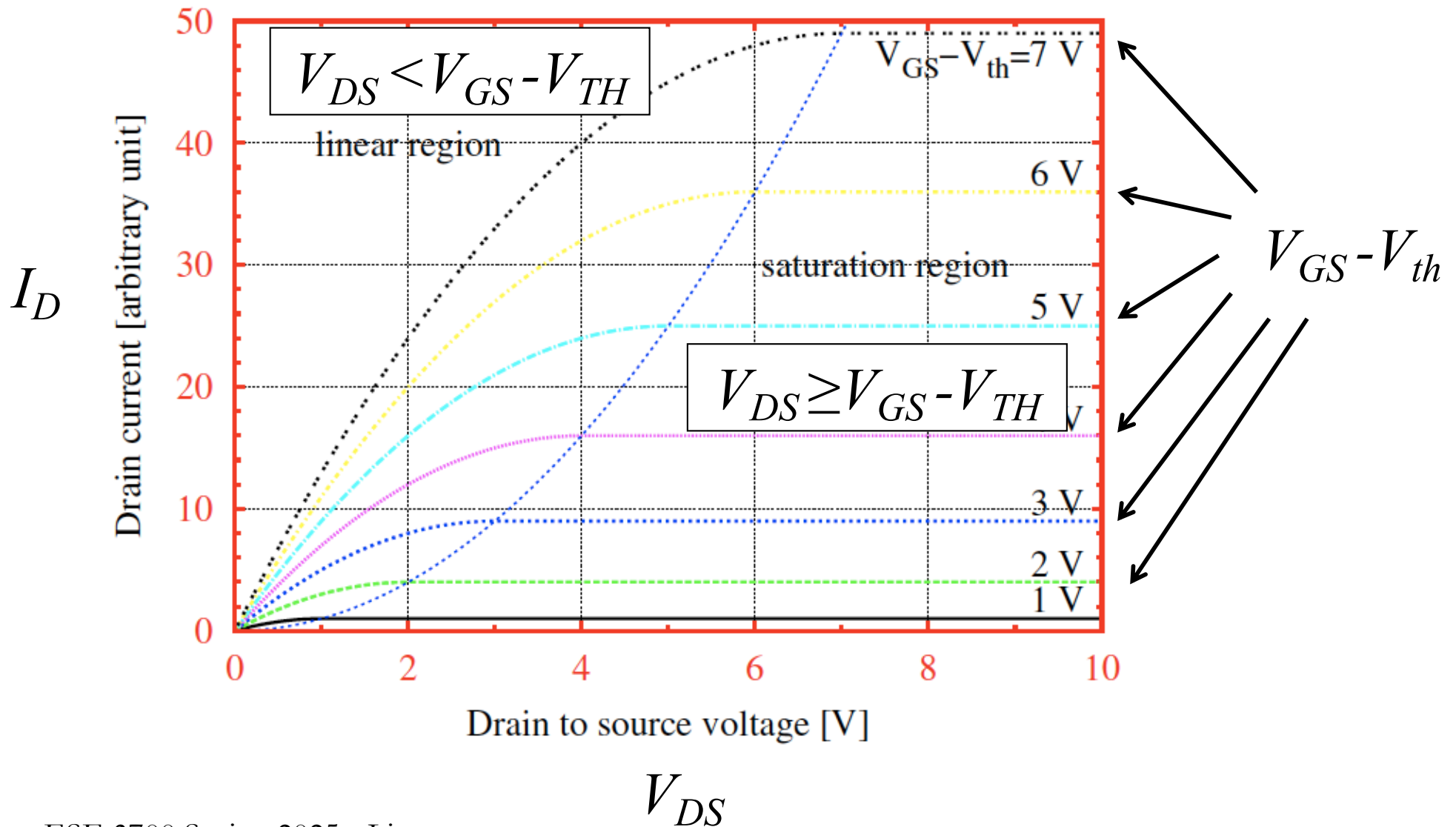


Define:  
 $V_{th} = \text{Threshold Voltage}$

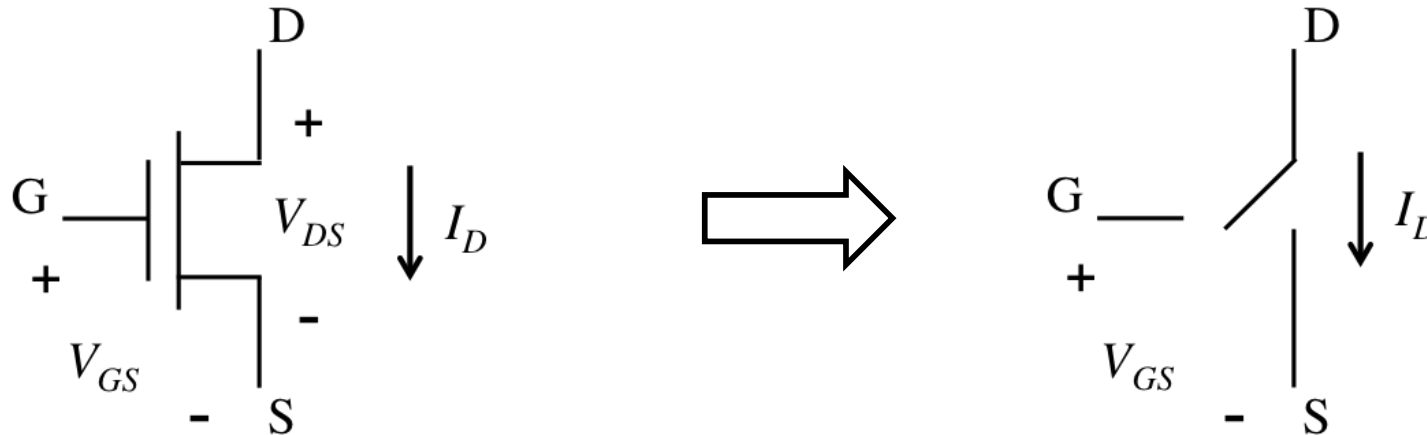
# MOSFET – IV Characteristics



# MOSFET – IV Characteristics



# MOSFET – Zeroth Order Model



## □ Ideal Switch

$V_{GS} > V_{th} \rightarrow$  switch is closed, conducts

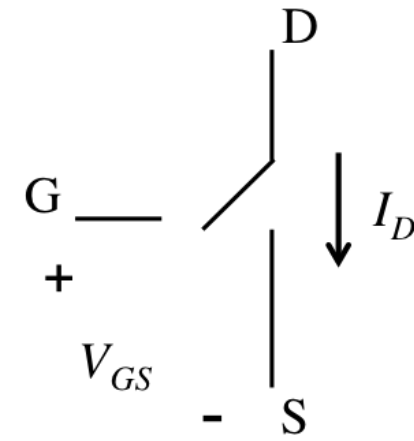
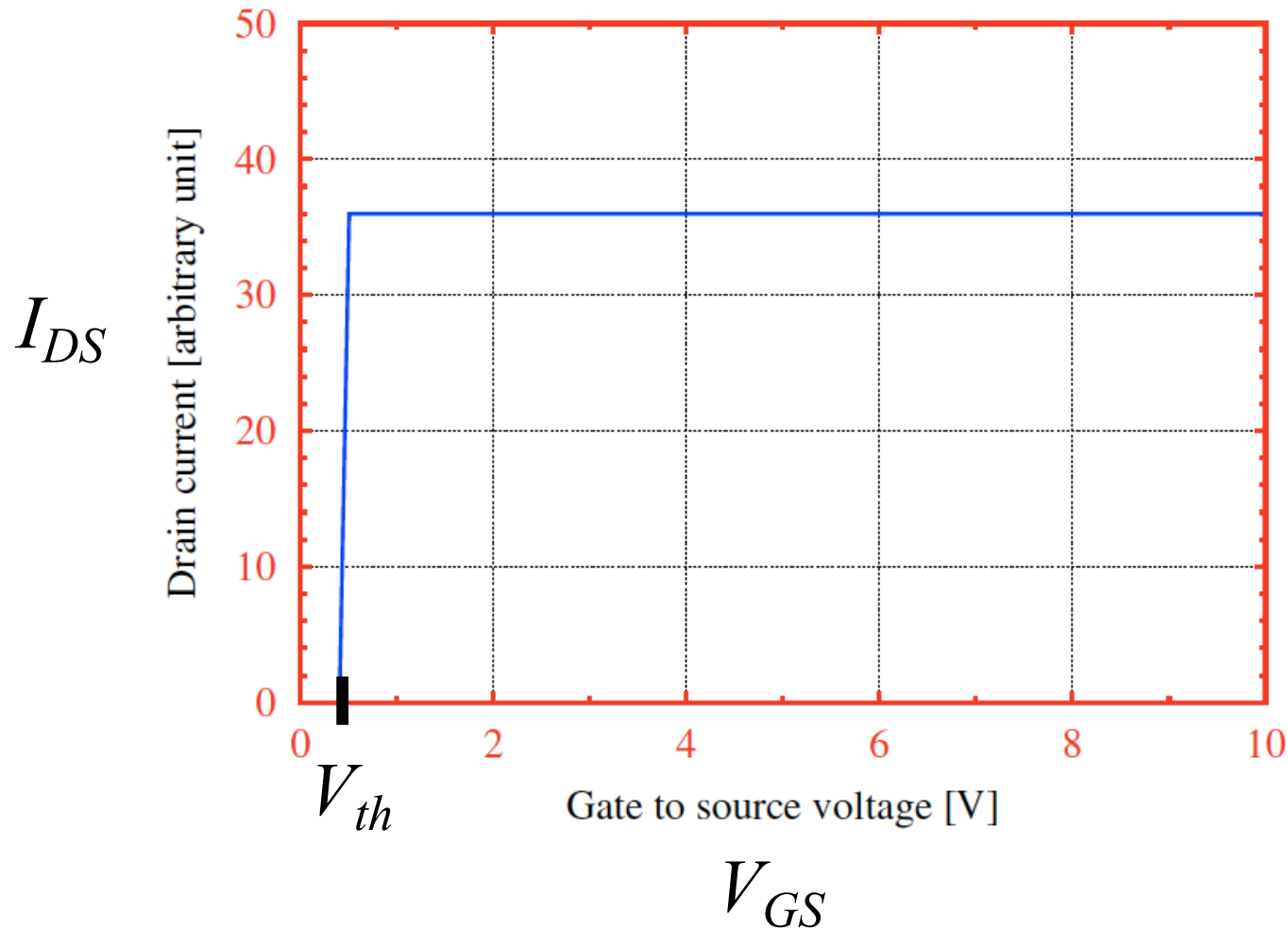
$V_{GS} < V_{th} \rightarrow$  switch is open, does not conduct

## □ Gate draws no current from input

- Loads input capacitively (gate capacitance)



# MOSFET – Zeroeth Order Model

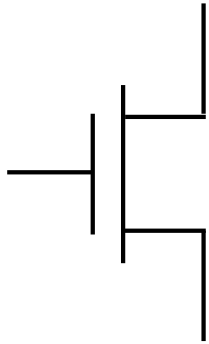




# MOSFET - Symmetric

---

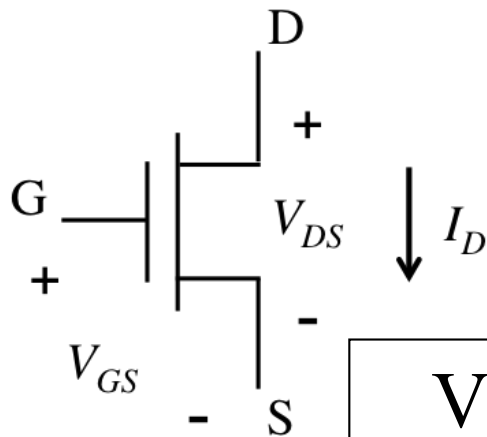
- ❑ Switch turned on for positive  $V_{GS}$ 
  - Which side is drain or source?



# MOSFET – N-Type

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

$$V_D > V_S$$



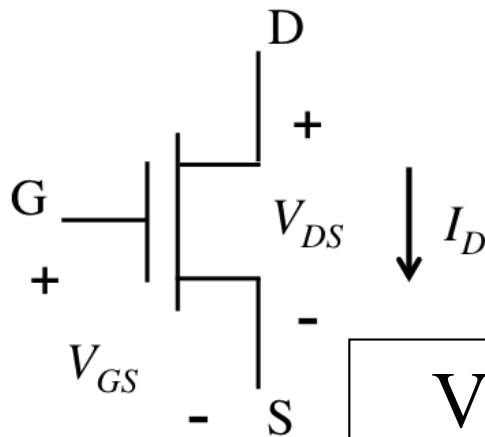
$V_{th,n} > 0$   
 $V_{GS} > V_{th,n}$   
to conduct

# MOSFET – N-Type, P-Type

□ Switch turned on for  
positive  $V_{GS}$

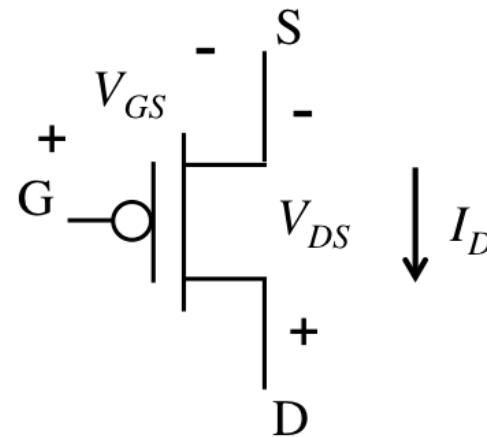
□ Switch turned on for  
negative  $V_{GS}$

$$V_D > V_S$$



$V_{th,n} > 0$   
 $V_{GS} > V_{th,n}$   
to conduct

$$V_S > V_D$$

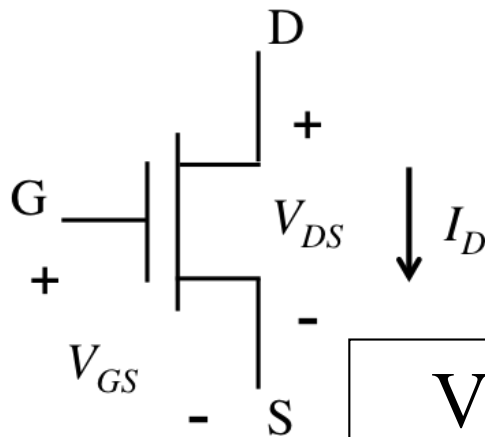


# MOSFET – N-Type, P-Type

□ Switch turned on for  
positive  $V_{GS}$

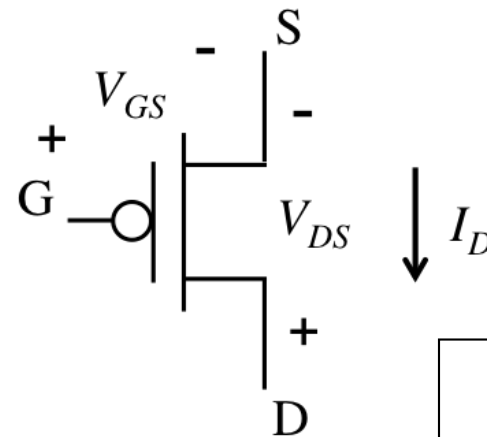
□ Switch turned on for  
negative  $V_{GS}$

$$V_D > V_S$$



$$\begin{aligned} V_{th,n} &> 0 \\ V_{GS} &> V_{th,n} \\ \text{to conduct} \end{aligned}$$

$$V_S > V_D$$

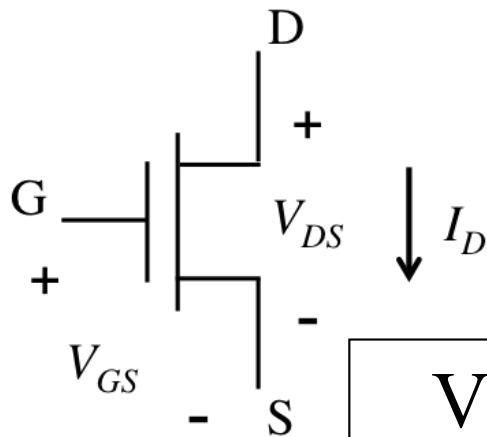


$$\begin{aligned} V_{th,p} &< 0 \\ V_{GS} &< V_{th,p} \\ \text{to conduct} \end{aligned}$$

# MOSFET – N-Type, P-Type

□ Switch turned on for  
positive  $V_{GS}$

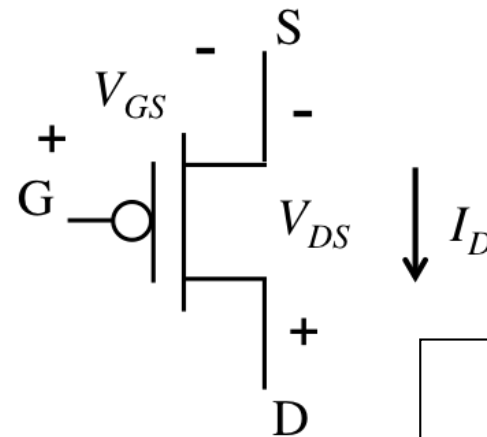
$$V_D > V_S$$



$V_{th,n} > 0$   
 $V_{GS} > V_{th,n}$   
to conduct

□ Switch turned on for  
negative  $V_{GS}$

$$V_S > V_D$$

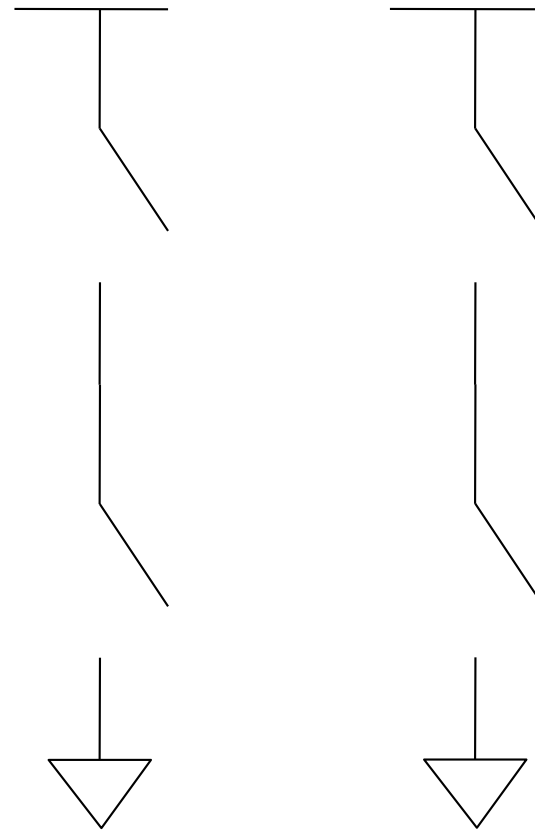
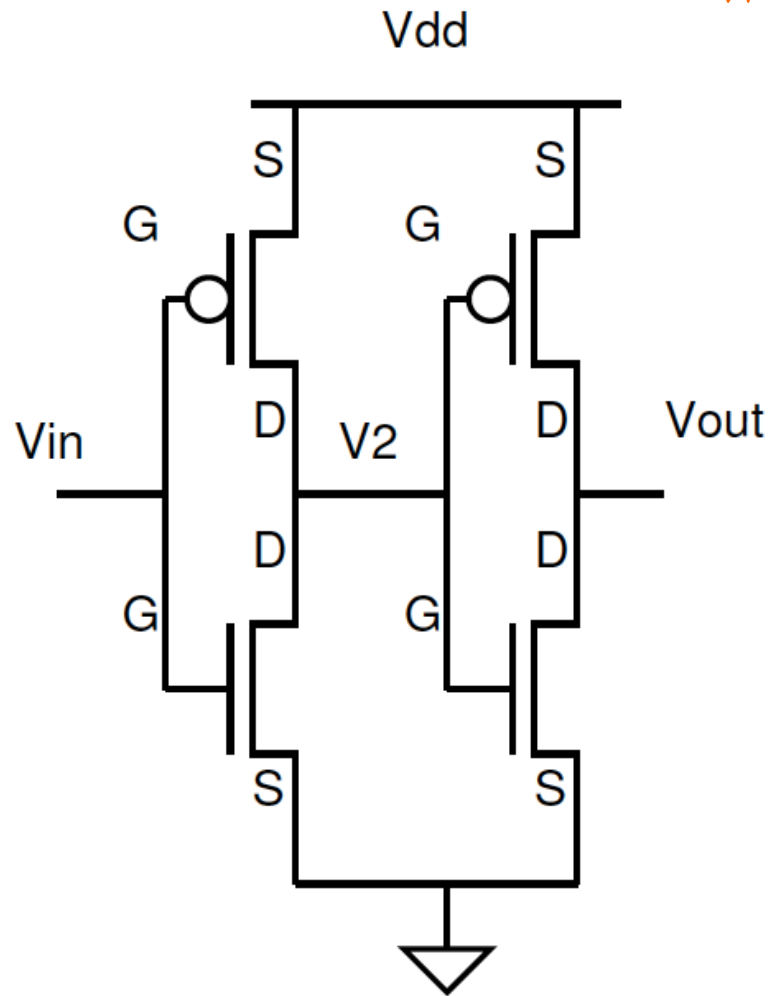


$|V_{th,p}| > 0$   
 $|V_{GS}| > |V_{th,p}|$   
to conduct

# Apply zero-order model

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

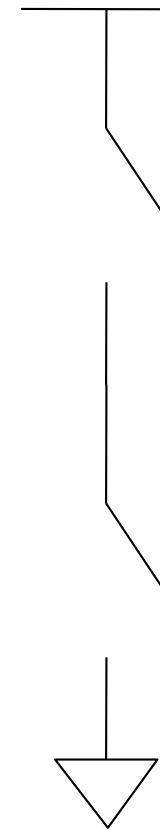
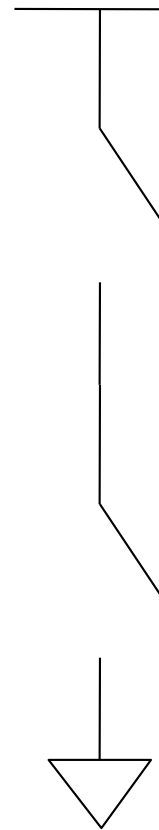
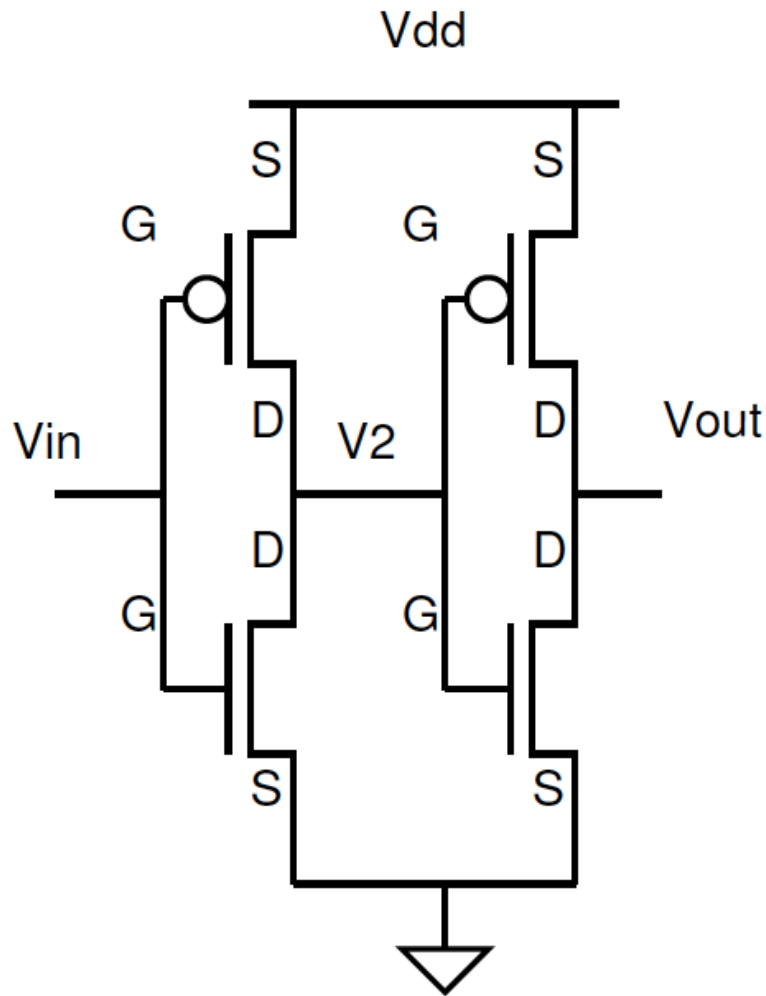
Why is it this way?



# Apply zero-order model?

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

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

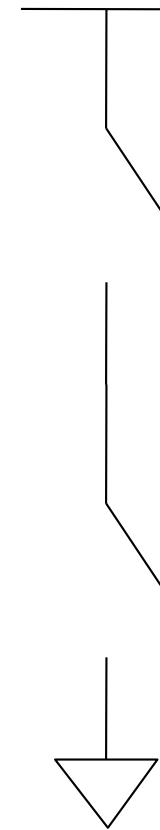
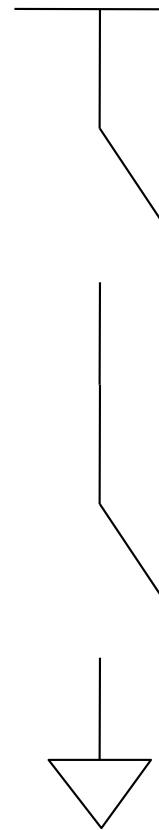
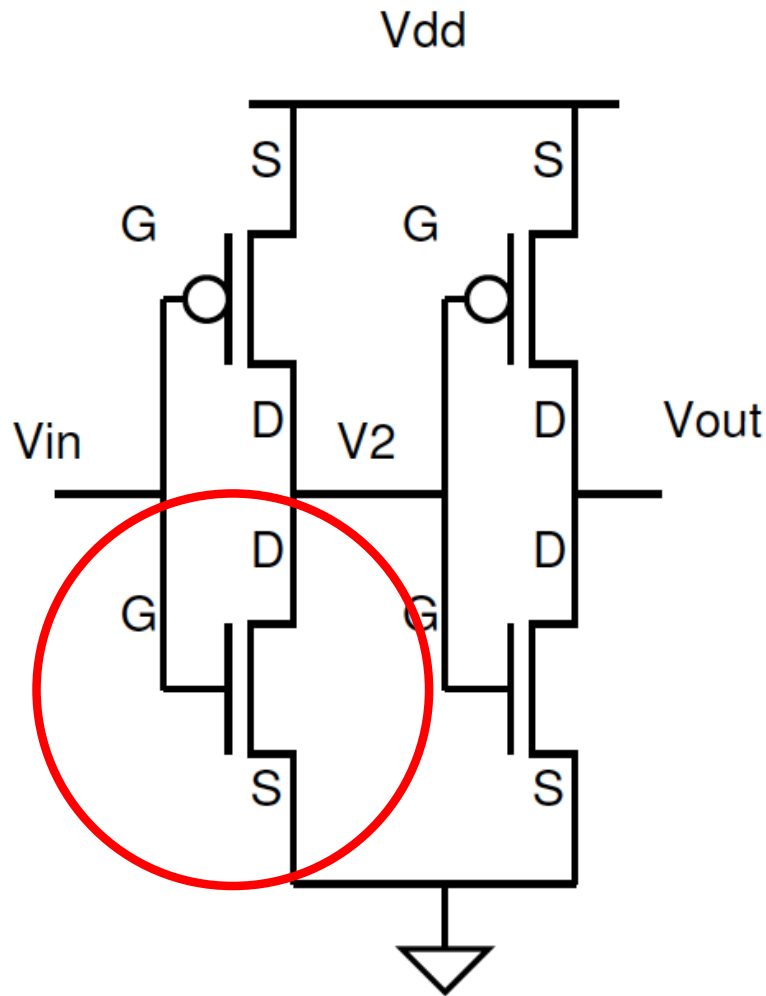




# Apply zero-order model?

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

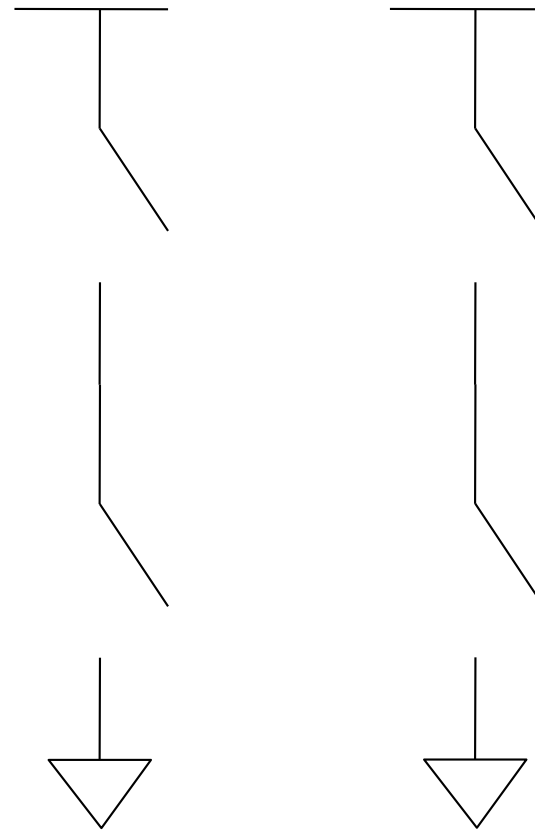
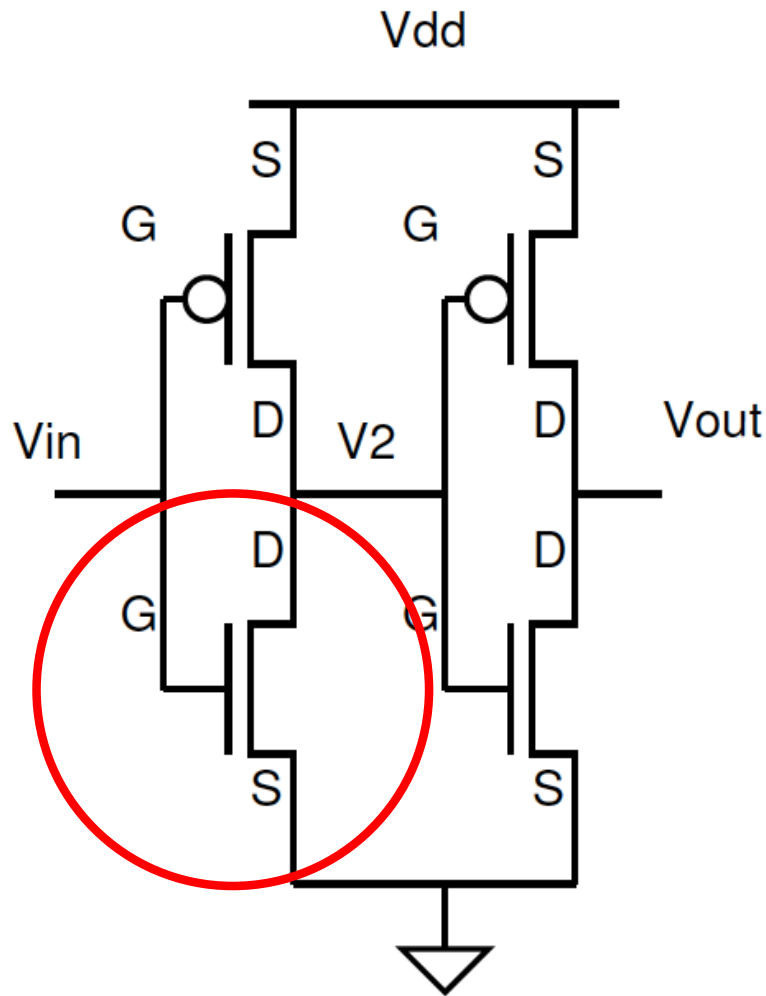
$$V_{th,p} = -V_{th,n}$$
$$V_{GS} = V_G - V_S$$



# Apply zero-order model?

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

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

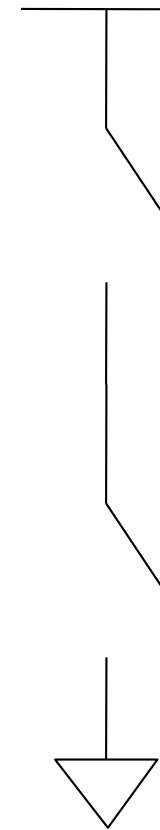
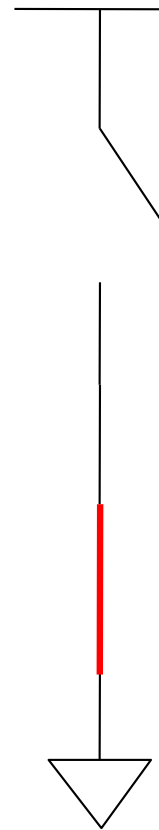
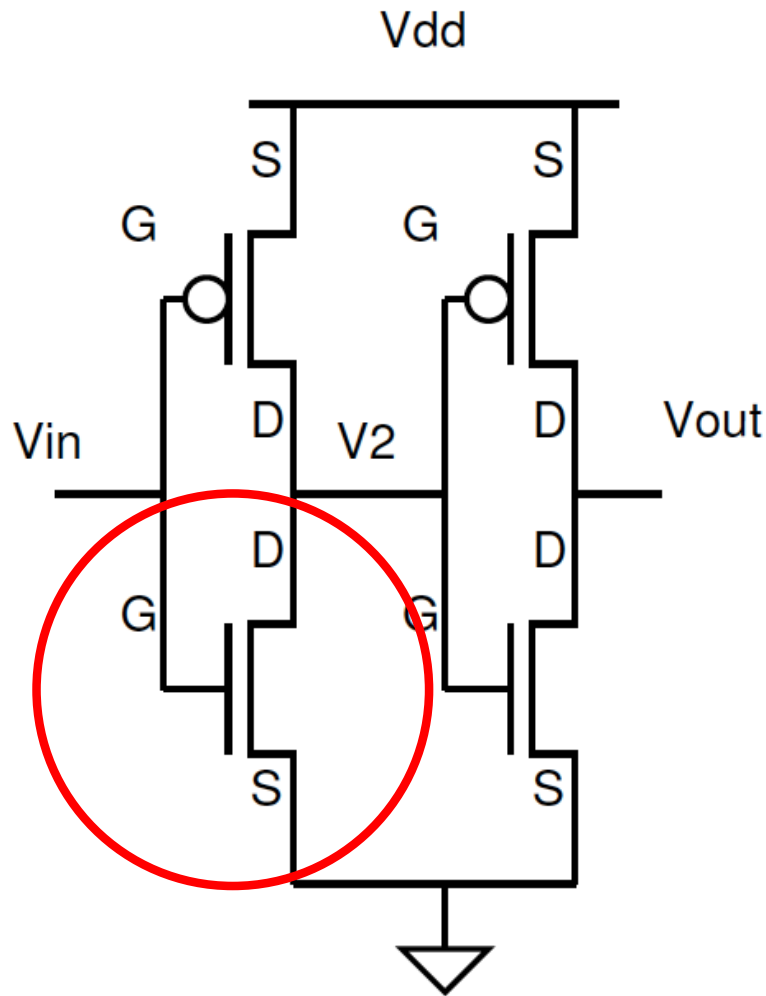


$$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$$

# Apply zero-order model?

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

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

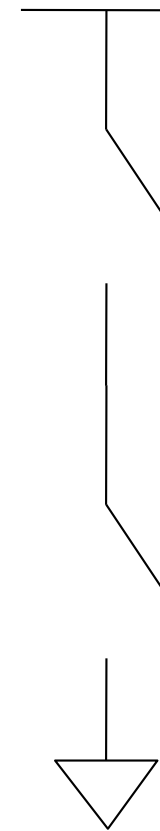
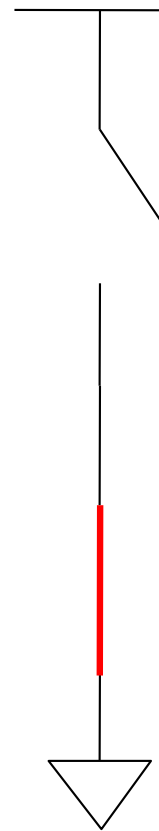
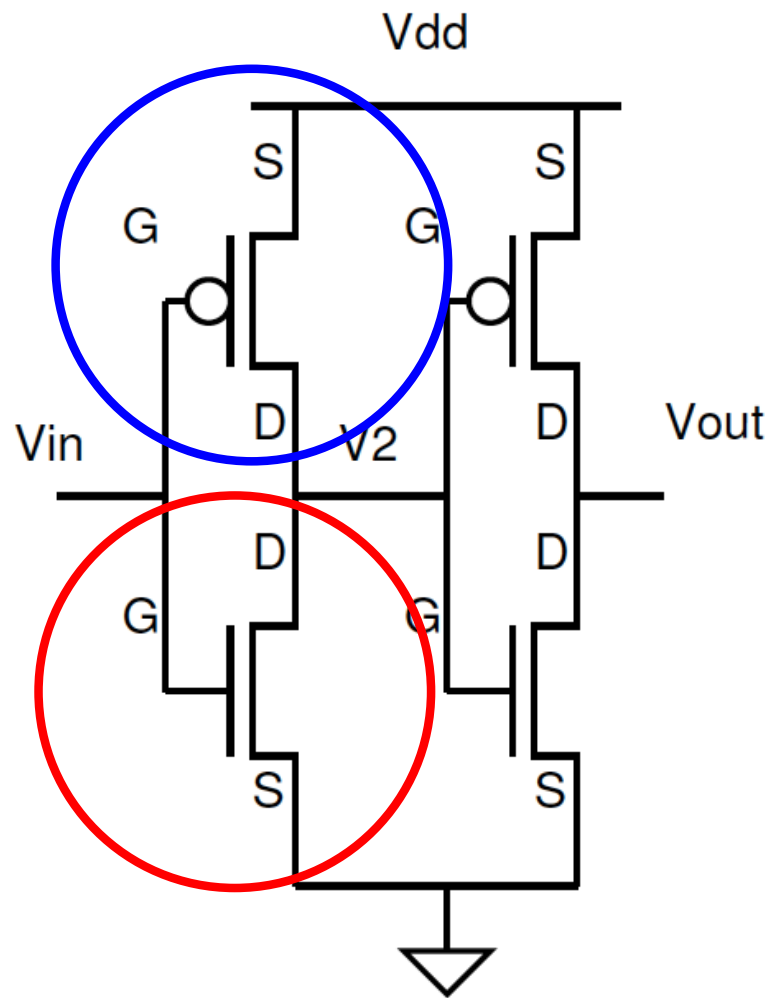


$$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$$

# Apply zero-order model?

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

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



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

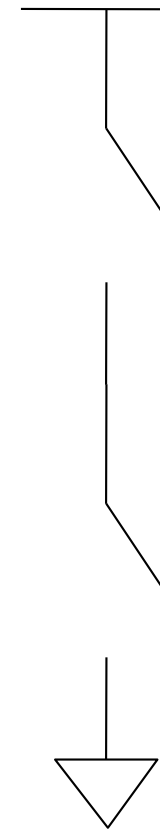
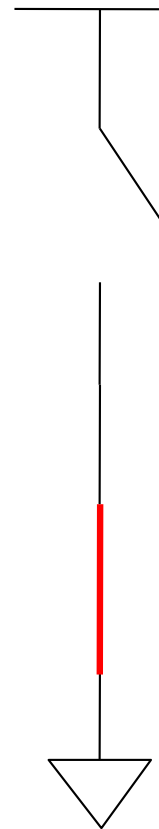
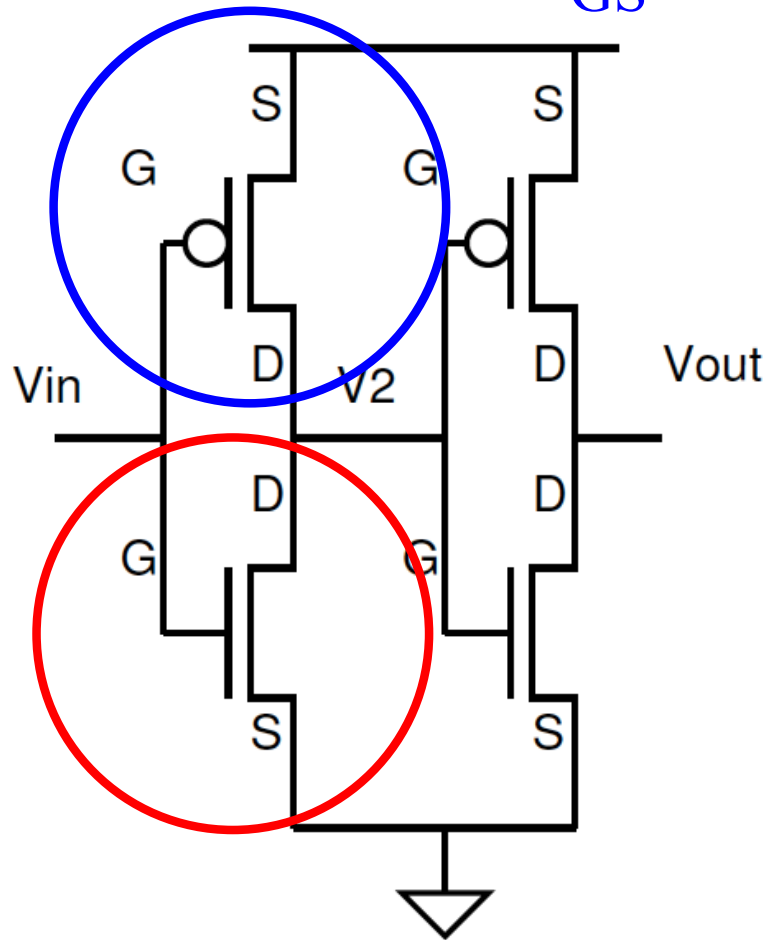
# Apply zero-order model?

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

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

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

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



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

# Apply zero-order model?

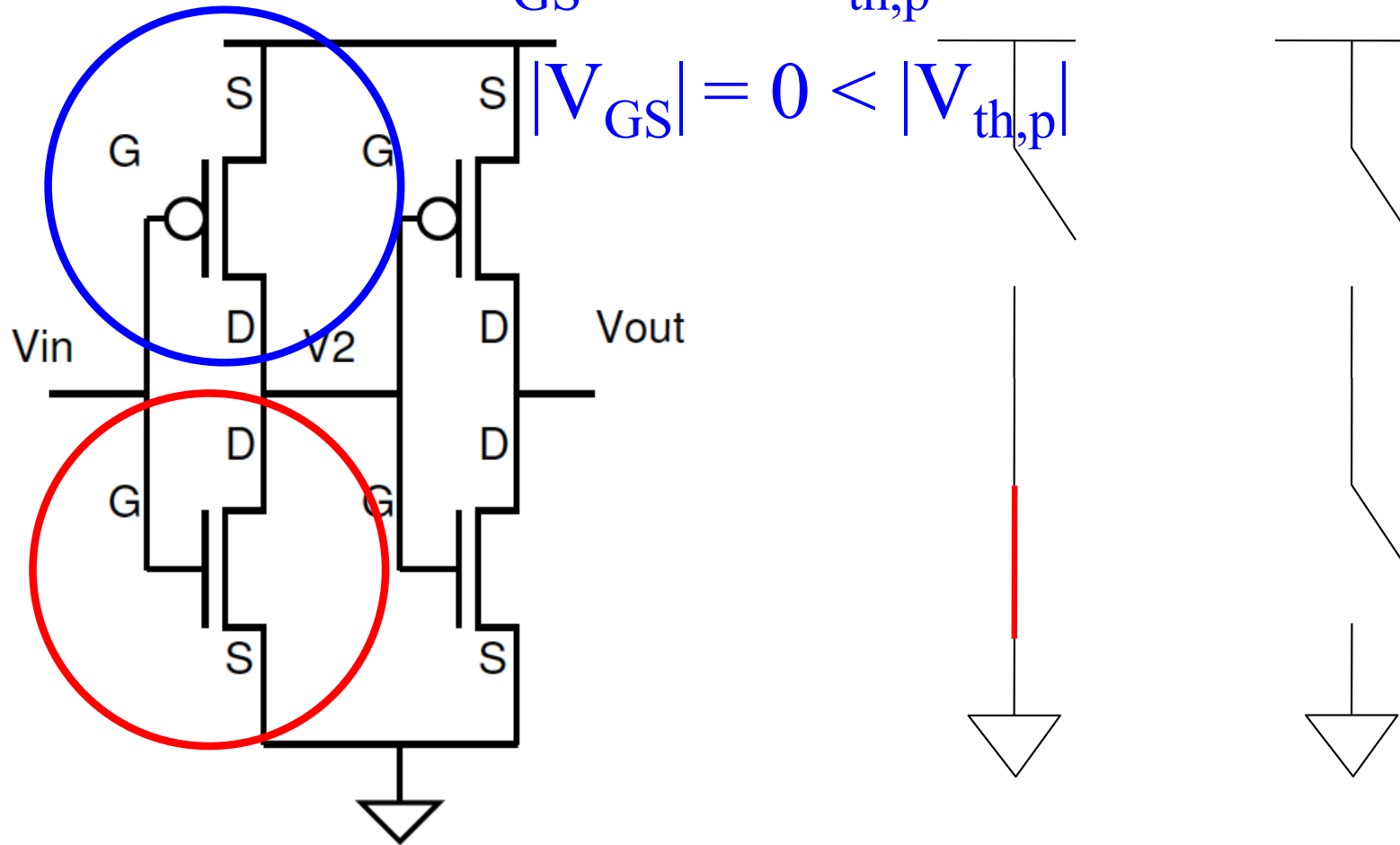
What happens when  $V_{in} = V_{dd} > V_{thn}$ ?

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

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

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

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



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

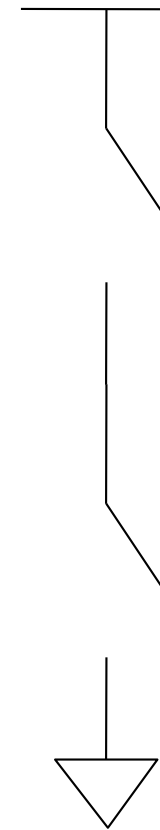
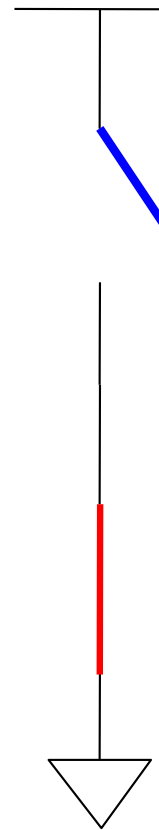
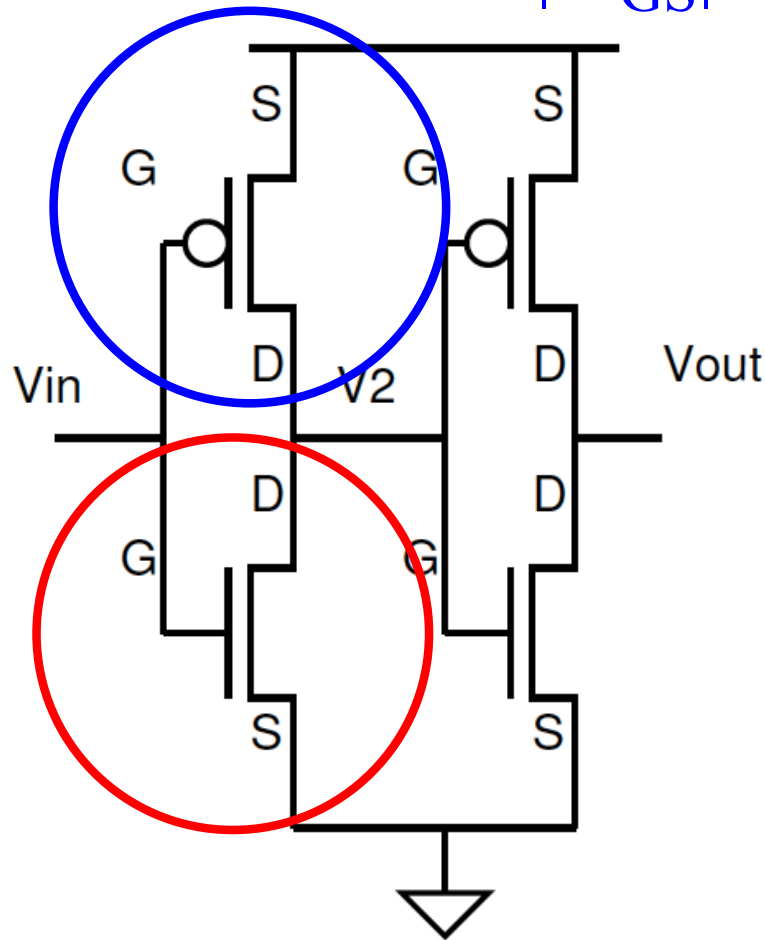
# Apply zero-order model?

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

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

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

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



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

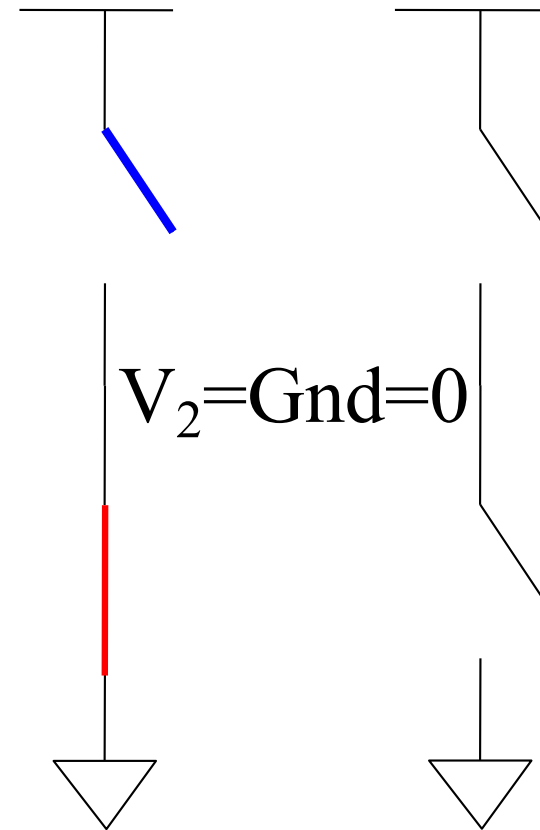
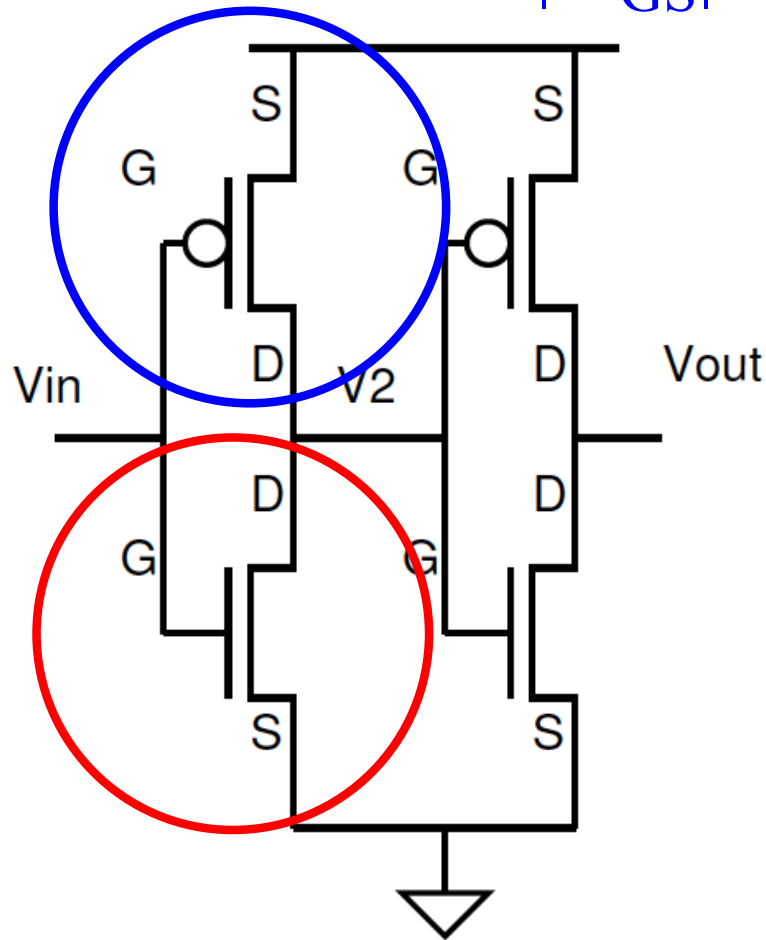
# Apply zero-order model?

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

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

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

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



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



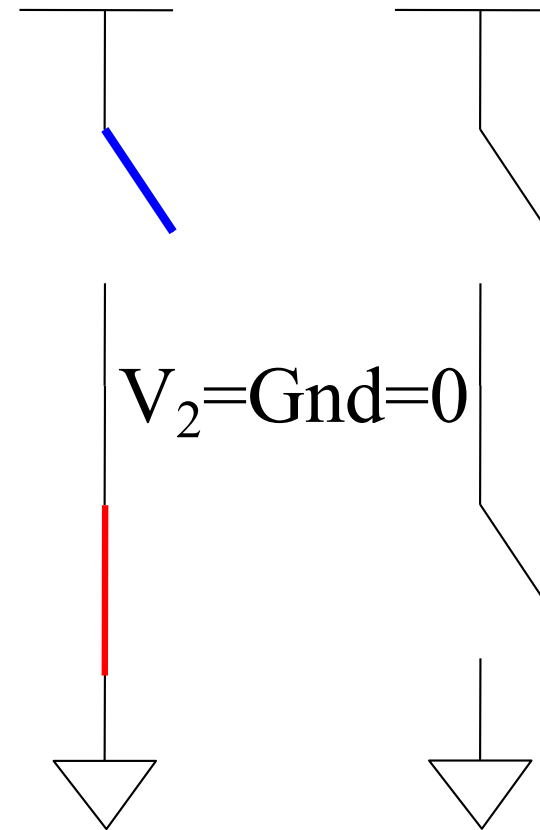
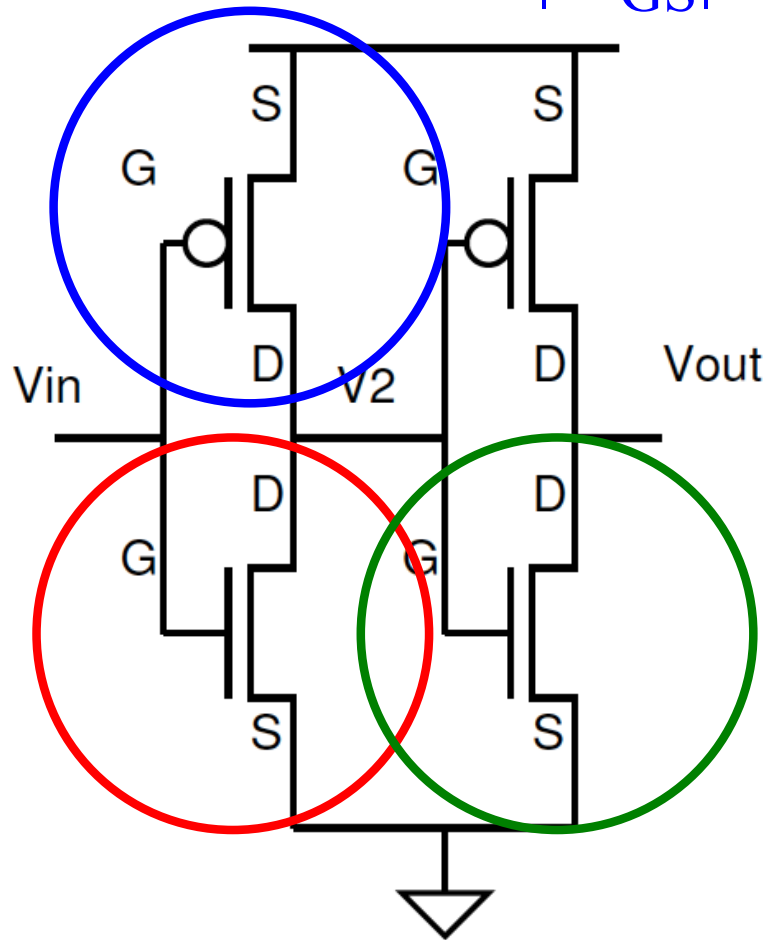
# Apply zero-order model?

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

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

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

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



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

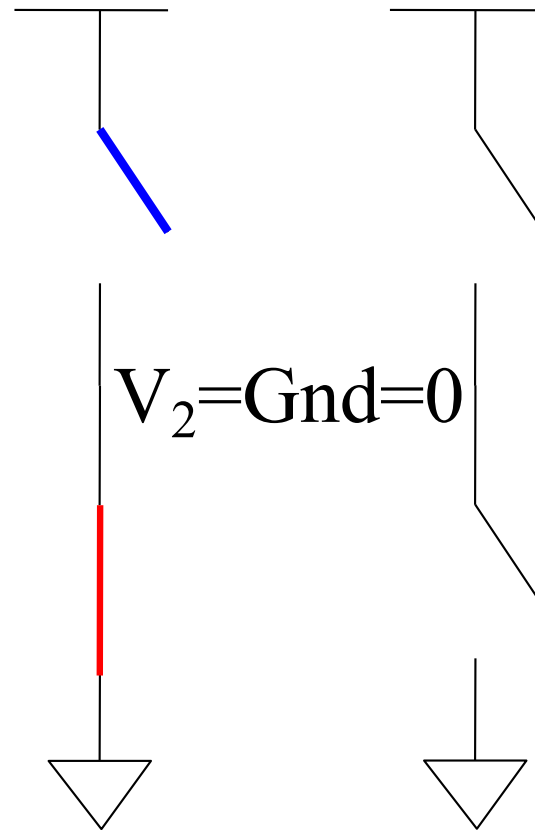
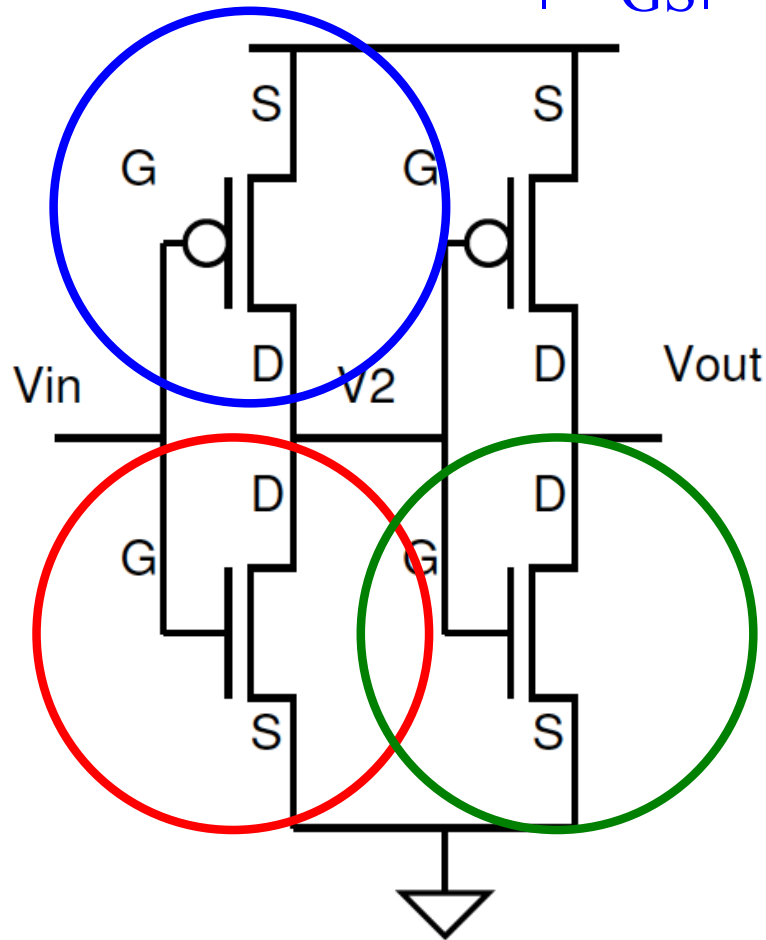
# Apply zero-order model?

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

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

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

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



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

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

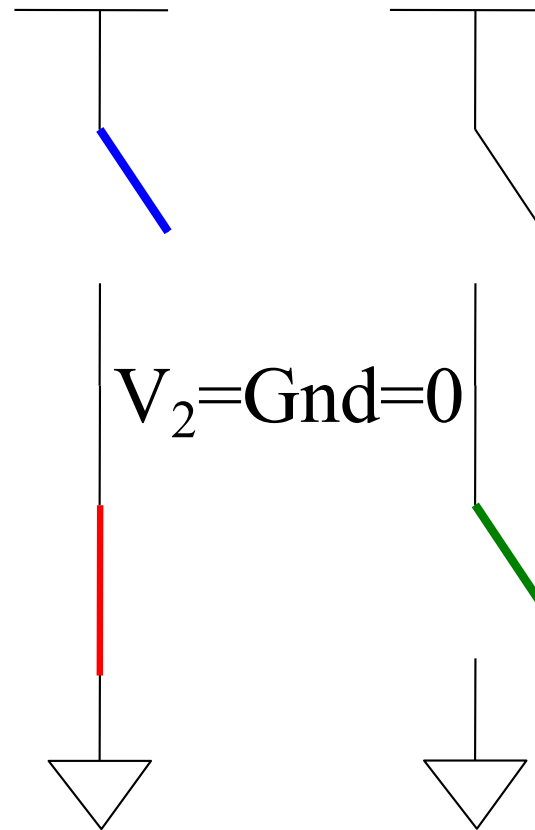
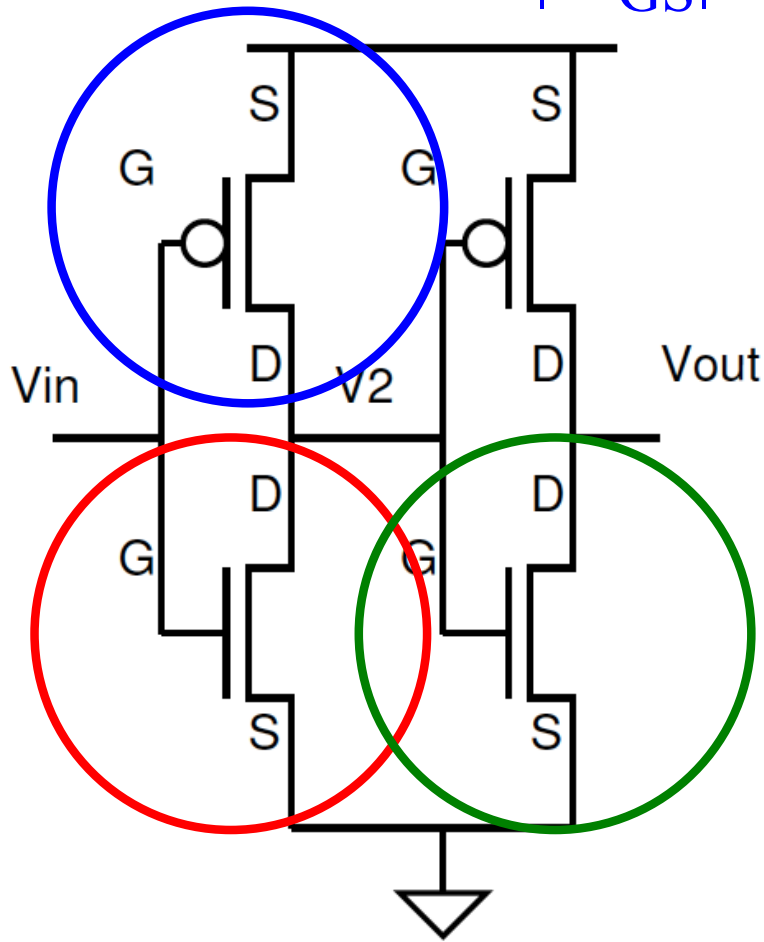
# Apply zero-order model?

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

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

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

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



$$V_2 = \text{Gnd} = 0$$

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

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

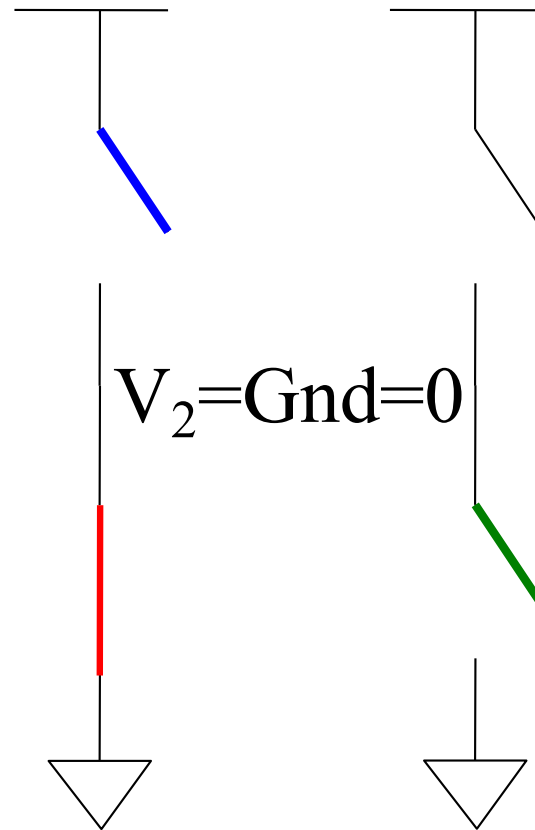
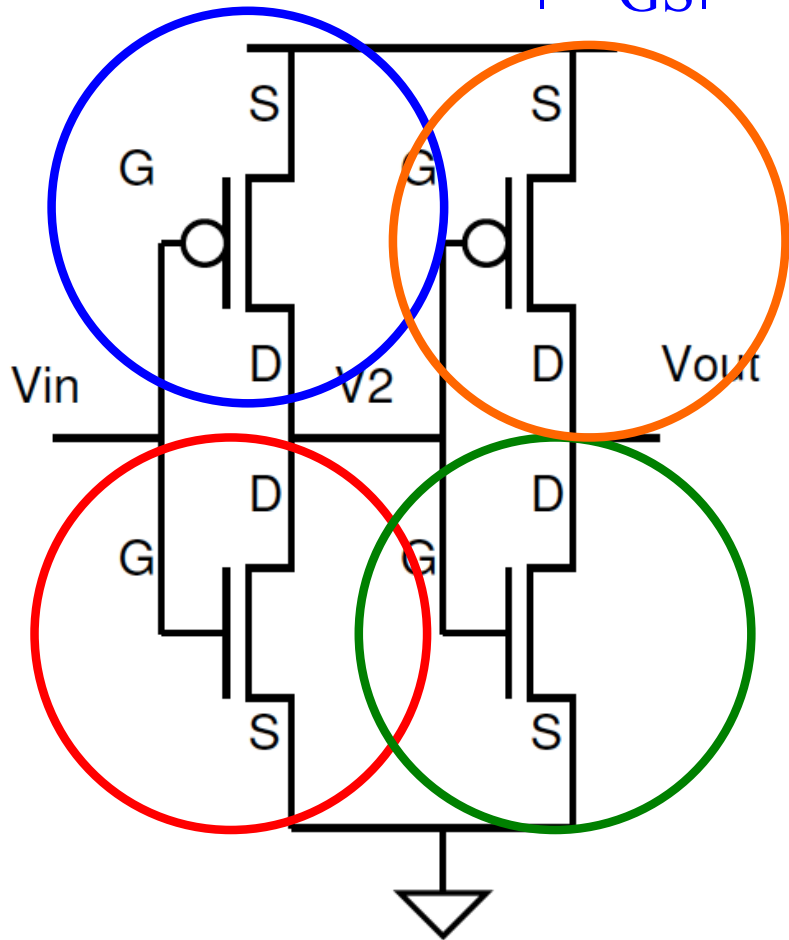
# Apply zero-order model?

What happens when  $V_{in} = V_{dd} > V_{th,n}$ ?

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

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

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



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

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

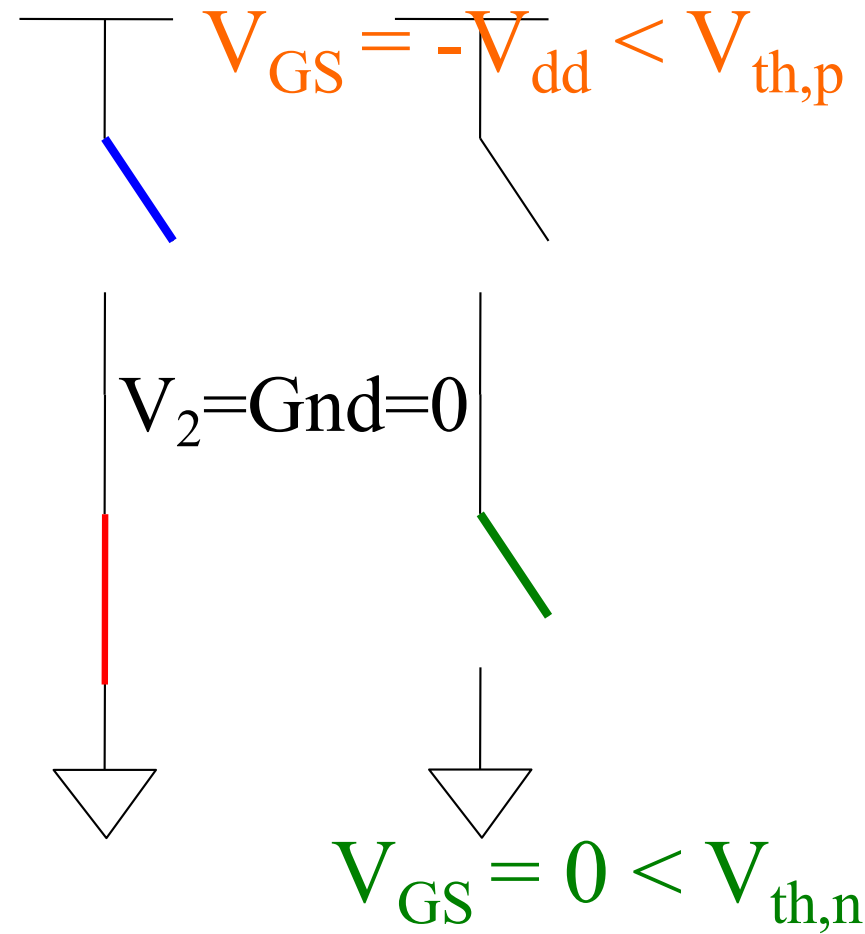
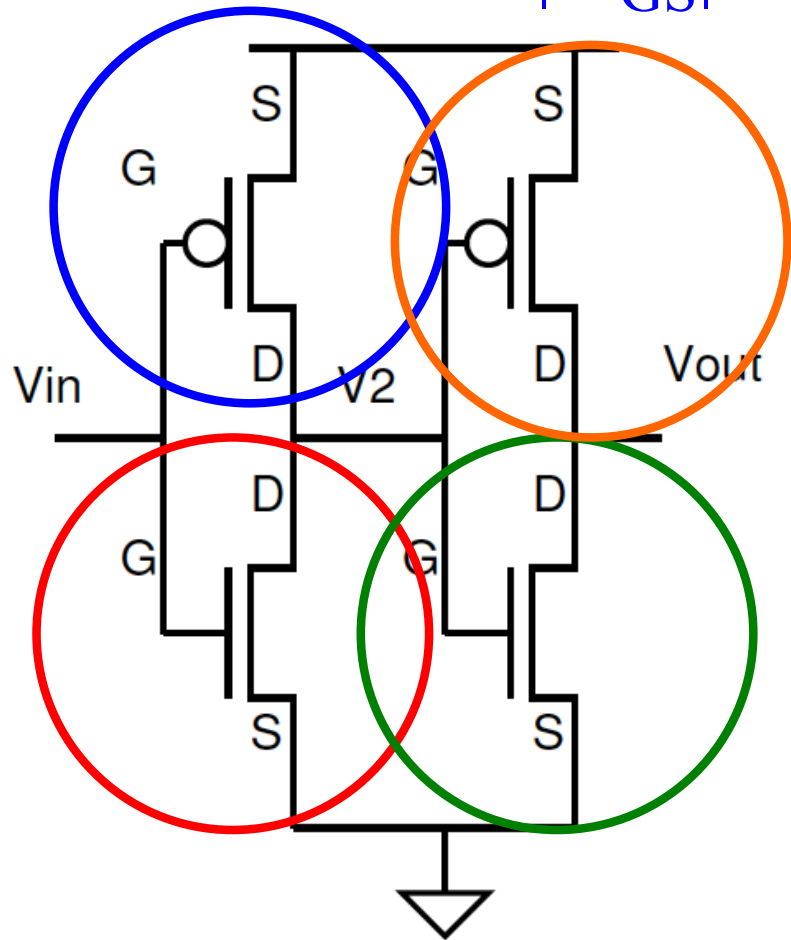
# Apply zero-order model?

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

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

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

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



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

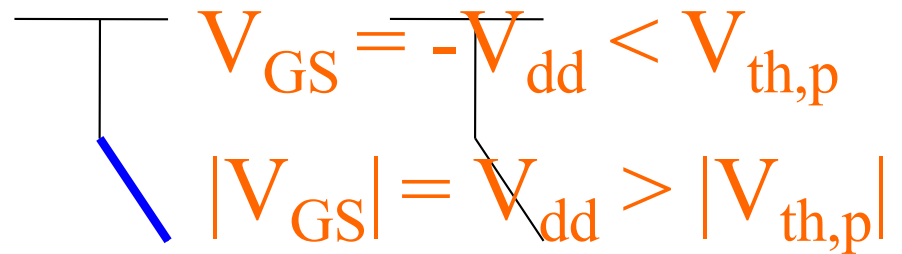
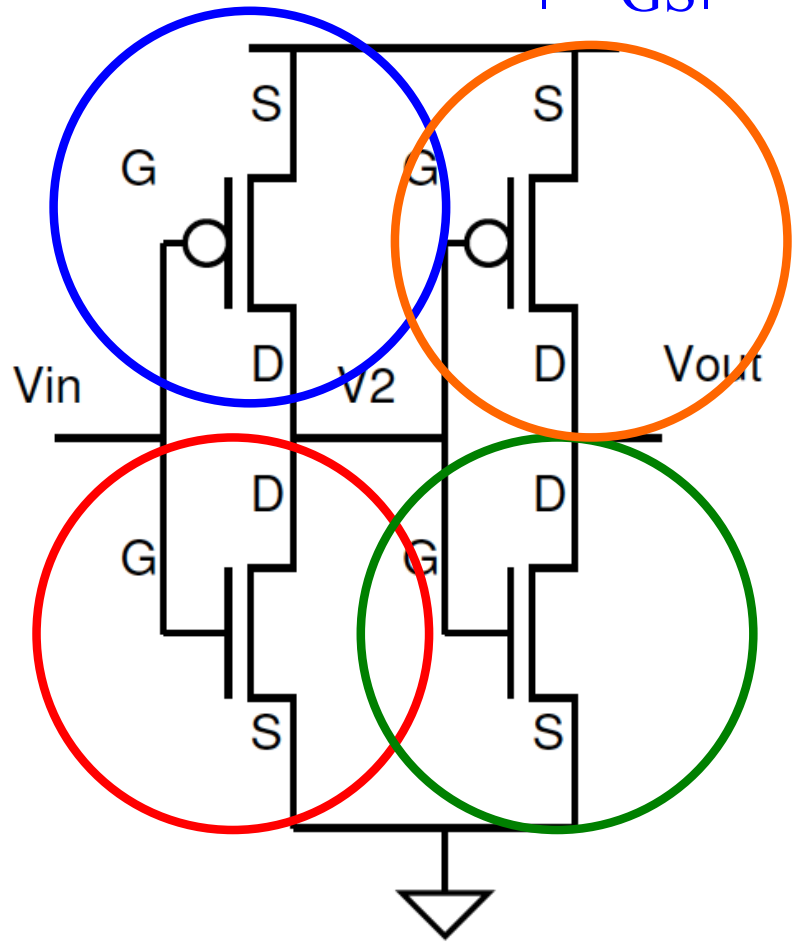
# Apply zero-order model?

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

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

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

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



$V_2 = \text{Gnd} = 0$

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

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

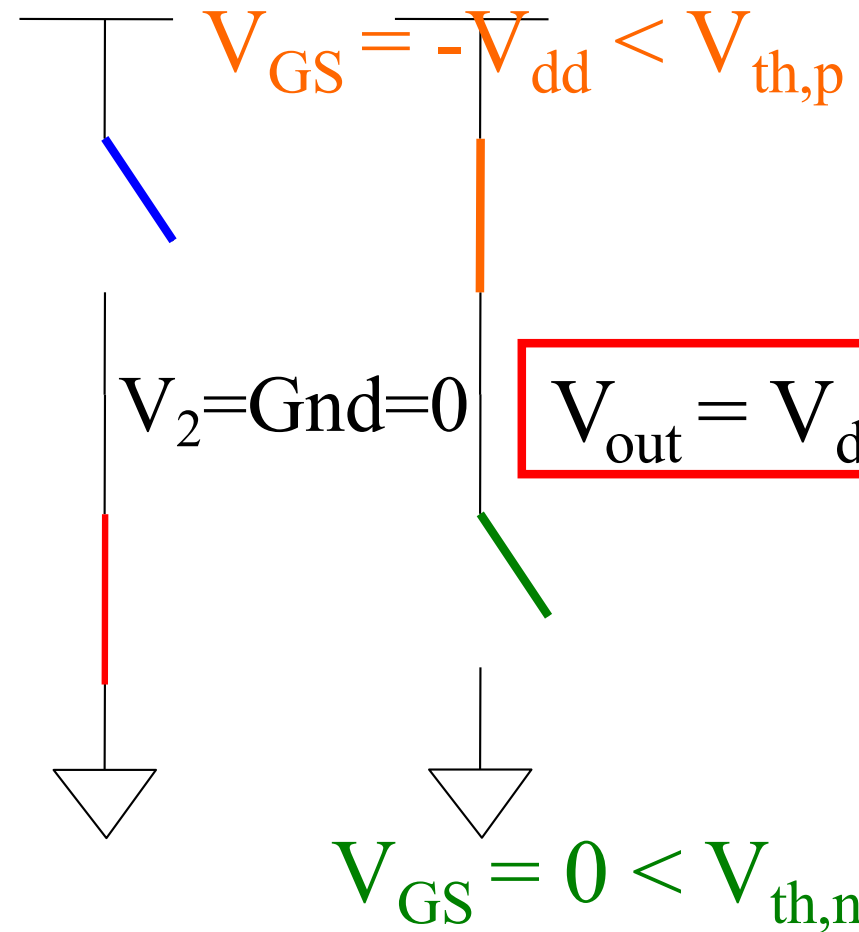
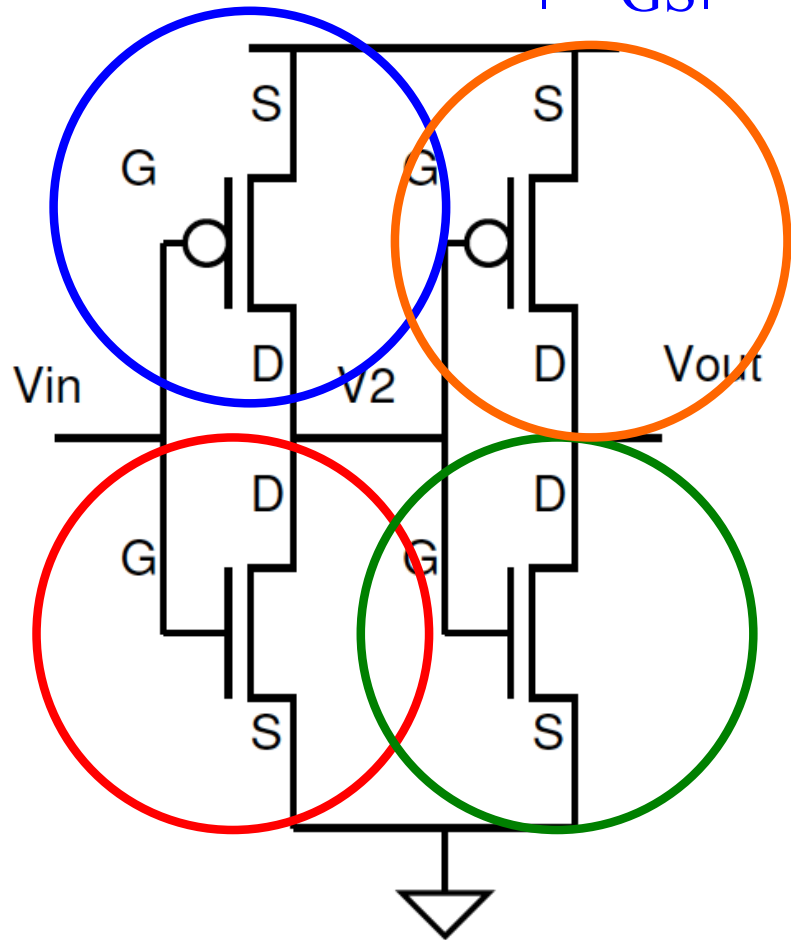
# Apply zero-order model?

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

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

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

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

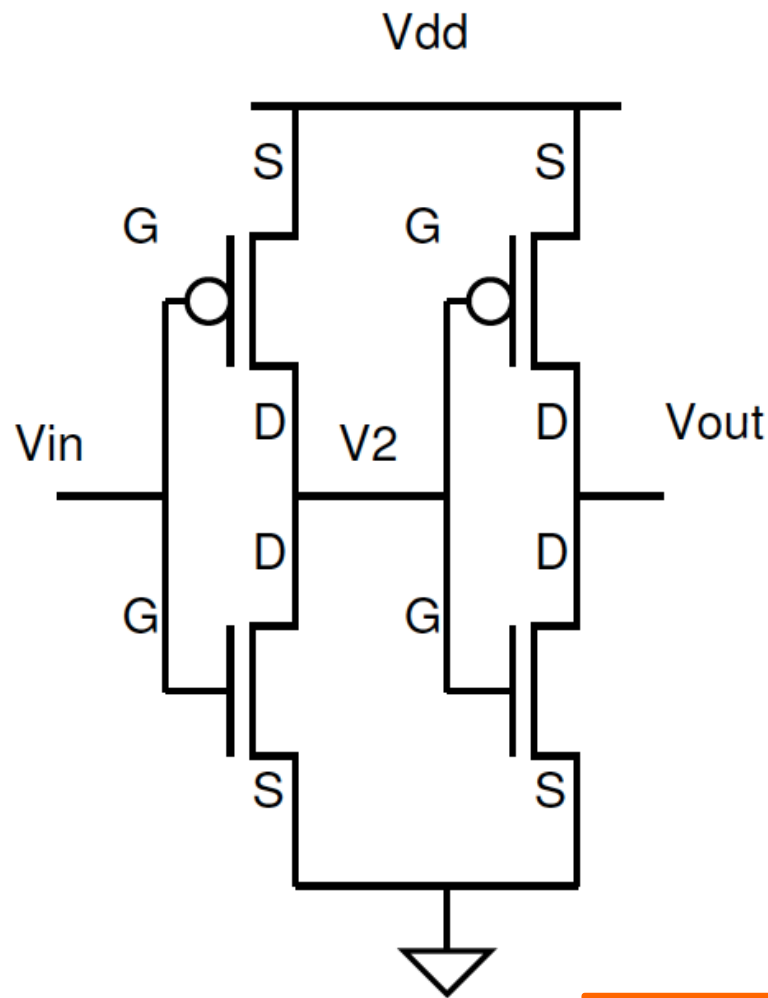


$$V_{out} = V_{dd}$$

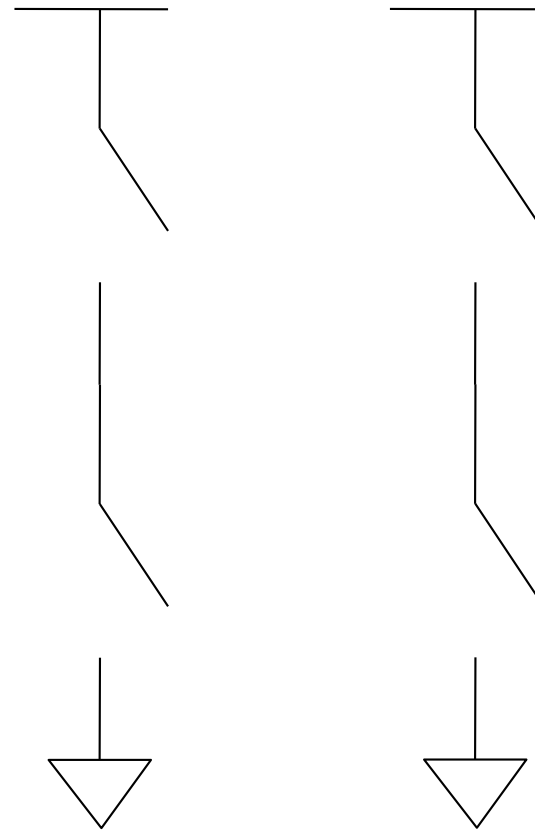
$$V_{GS} = V_G - V_S = V_{dd} > V_{th,n}$$

# 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  $V_{out} = 0$ .**



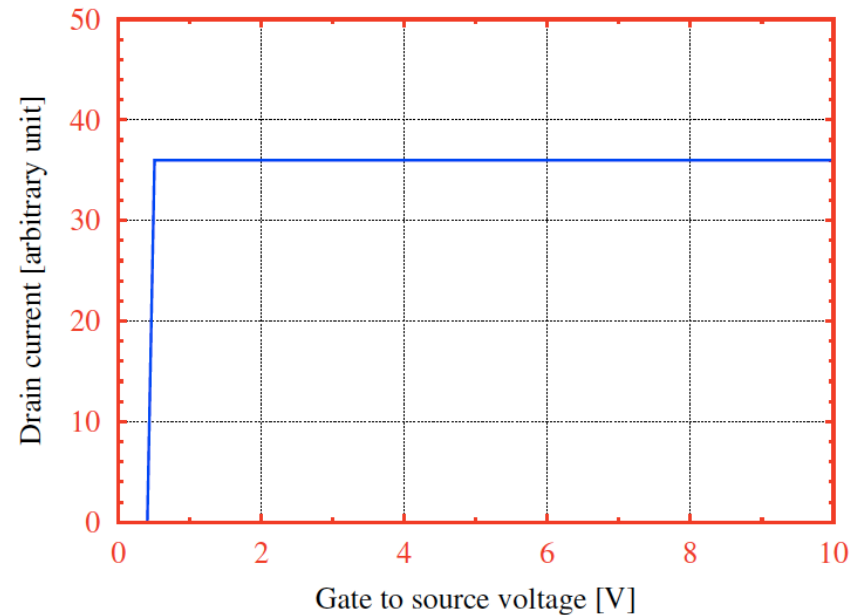
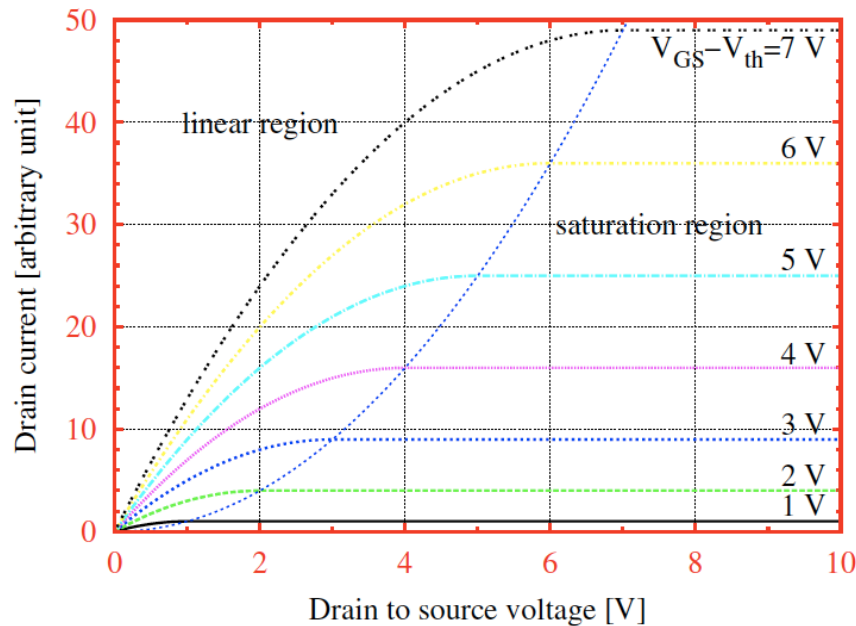


# Zeroeth Order Model

---

- ❑ 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?



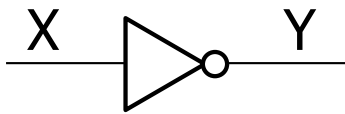
- ❑ Delay
  - Parasitic capacitances and resistances
- ❑ Dynamics
- ❑ Zeroeth Order captures behaviour if our circuit is **not**:
  - Capacitively loaded, acyclic (if there are Loops)

---

# Digital Logic

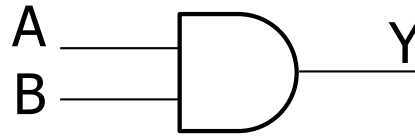
# Basic Digital Gates

NOT  
 $Y = \bar{X}$



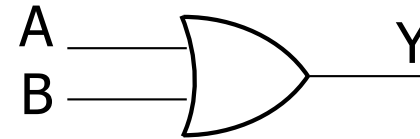
X	Y
0	1
1	0

AND  
 $Y = A \cdot B$



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

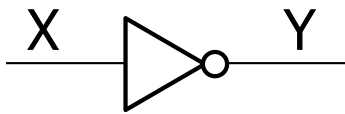
OR  
 $Y = A + B$



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

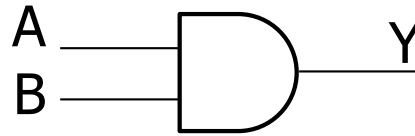
# Basic Digital Gates

NOT  
 $Y = \bar{X}$



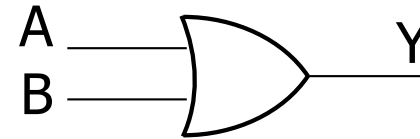
X	Y
0	1
1	0

AND  
 $Y = A \cdot B$



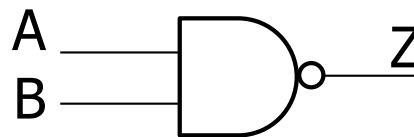
A	B	Y	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

OR  
 $Y = A + B$

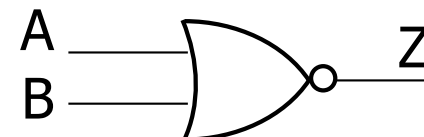


A	B	Y	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

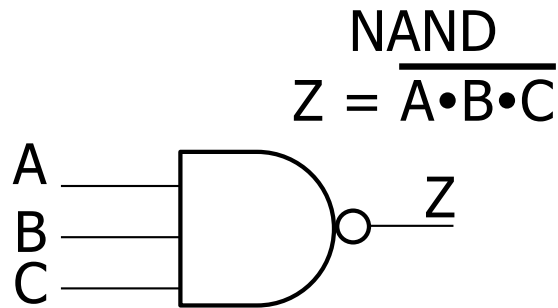
NAND  
 $Z = \overline{A \cdot B}$



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

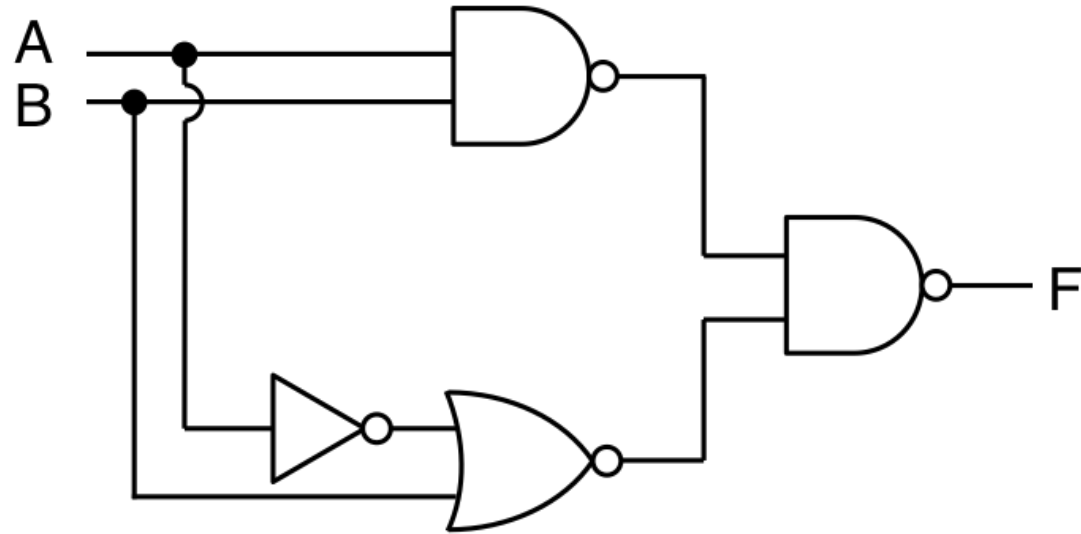
□ **TABLE 2-3**  
**Basic Identities of Boolean Algebra**

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. $X + \bar{X} = 1$	8. $X \cdot \bar{X} = 0$	
9. $\overline{\bar{X}} = X$		
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. $\overline{X + Y} = \bar{X} \cdot \bar{Y}$	17. $\overline{X \cdot Y} = \bar{X} + \bar{Y}$	DeMorgan's



# Combination

---





# Boolean Expressions

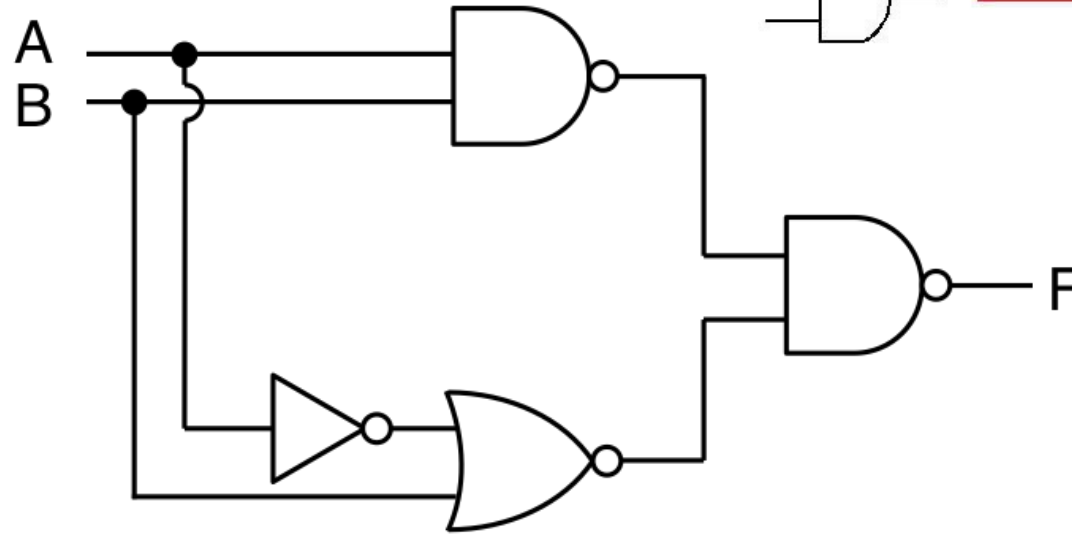
- Sum-of-products form (SOP)
  - Eg.  $ABC+DEF+GHI$
- Product-of-sums form (POS)
  - Eg.  $(A+B+C)(D+E+F)(G+H+I)$
- Convert between the two with Boolean algebra
  - DeMorgan's Law

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

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



# 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

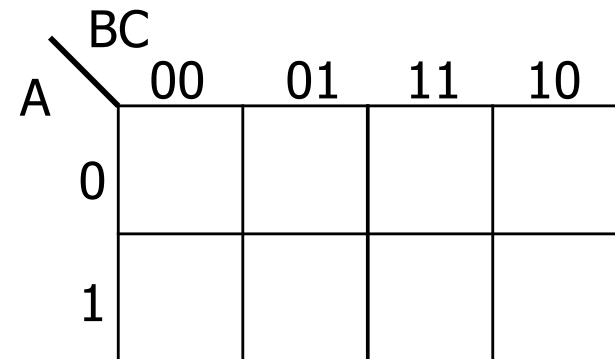
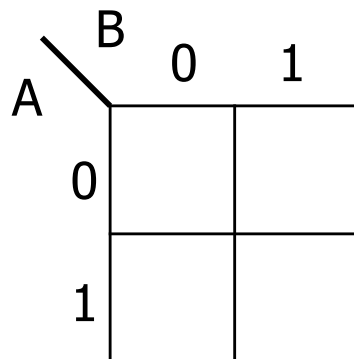
- Every minterm in your expression has every variable

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

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


# 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 1)

□ 2-variable

		B	
		0	1
A	0		
	1		

$$\text{Eg: } Z = A'B' + AB' + A'B$$

□ 3-variable

		BC			
		00	01	11	10
A	0				
	1				

# K-map Examples (Preclass 2)

□ 2-variable

		B	0	1
A	0			
	1			

$$\text{Eg: } Z = A'B' + AB' + A'B$$

□ 3-variable

				BC	00	01	11	10
A	0							
	1							

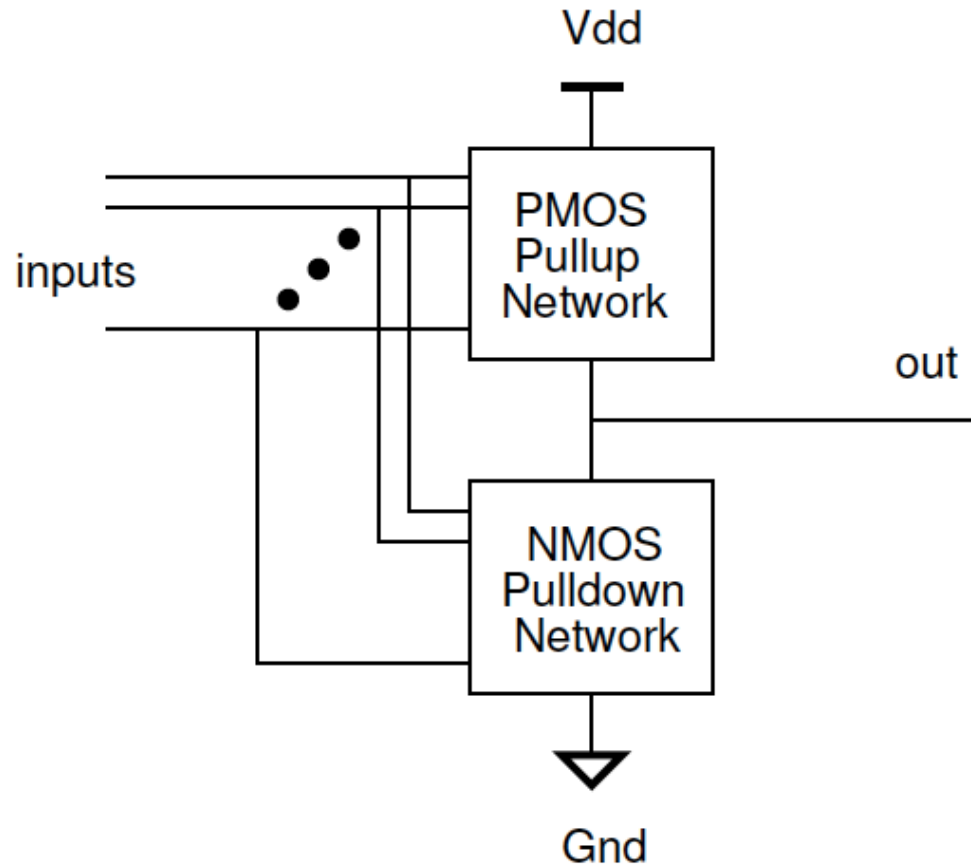
$$\text{Eg: } Z = A'B'C' + A'B + ABC' + AC$$

---

# CMOS Gates

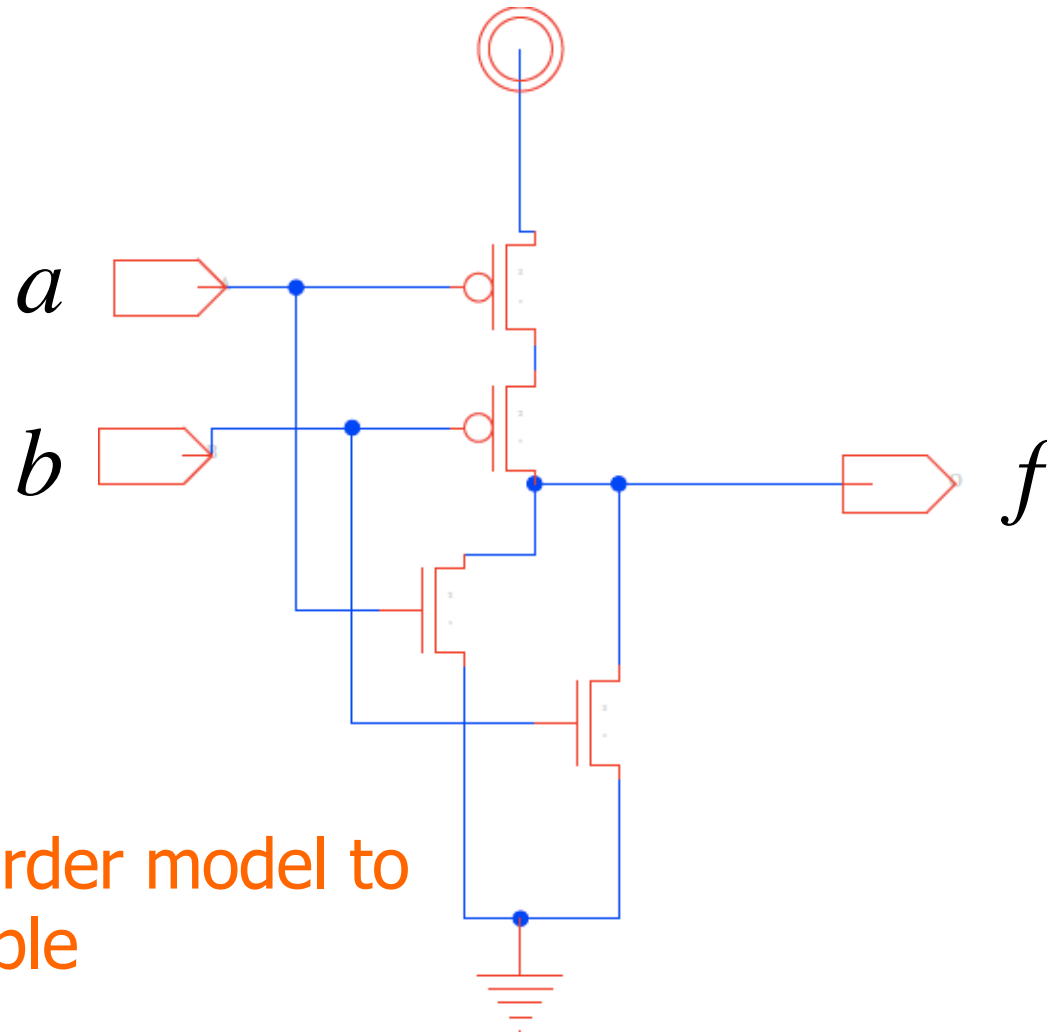


# How to construct static CMOS gates



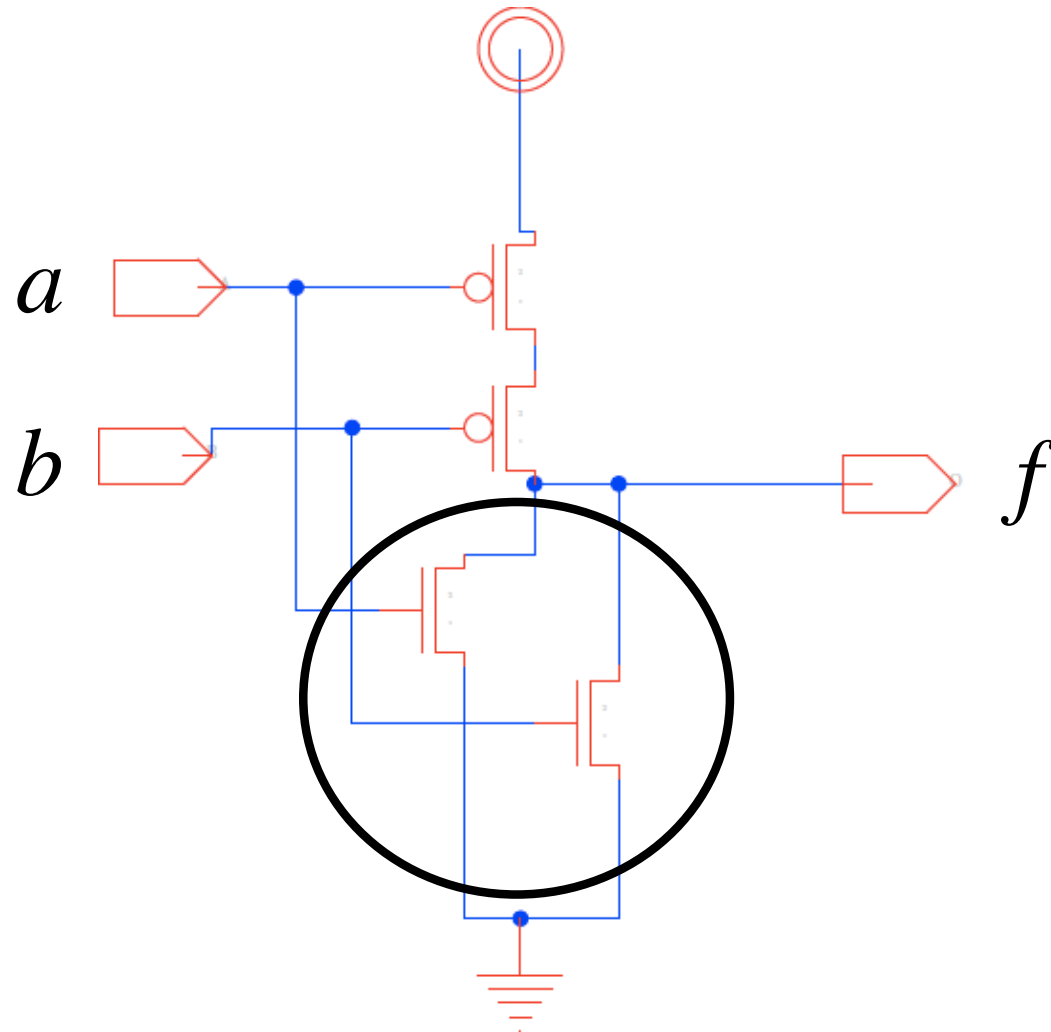
- ❑ Complementary Metal Oxide Semiconductor

# What gate is this? Preclass 3

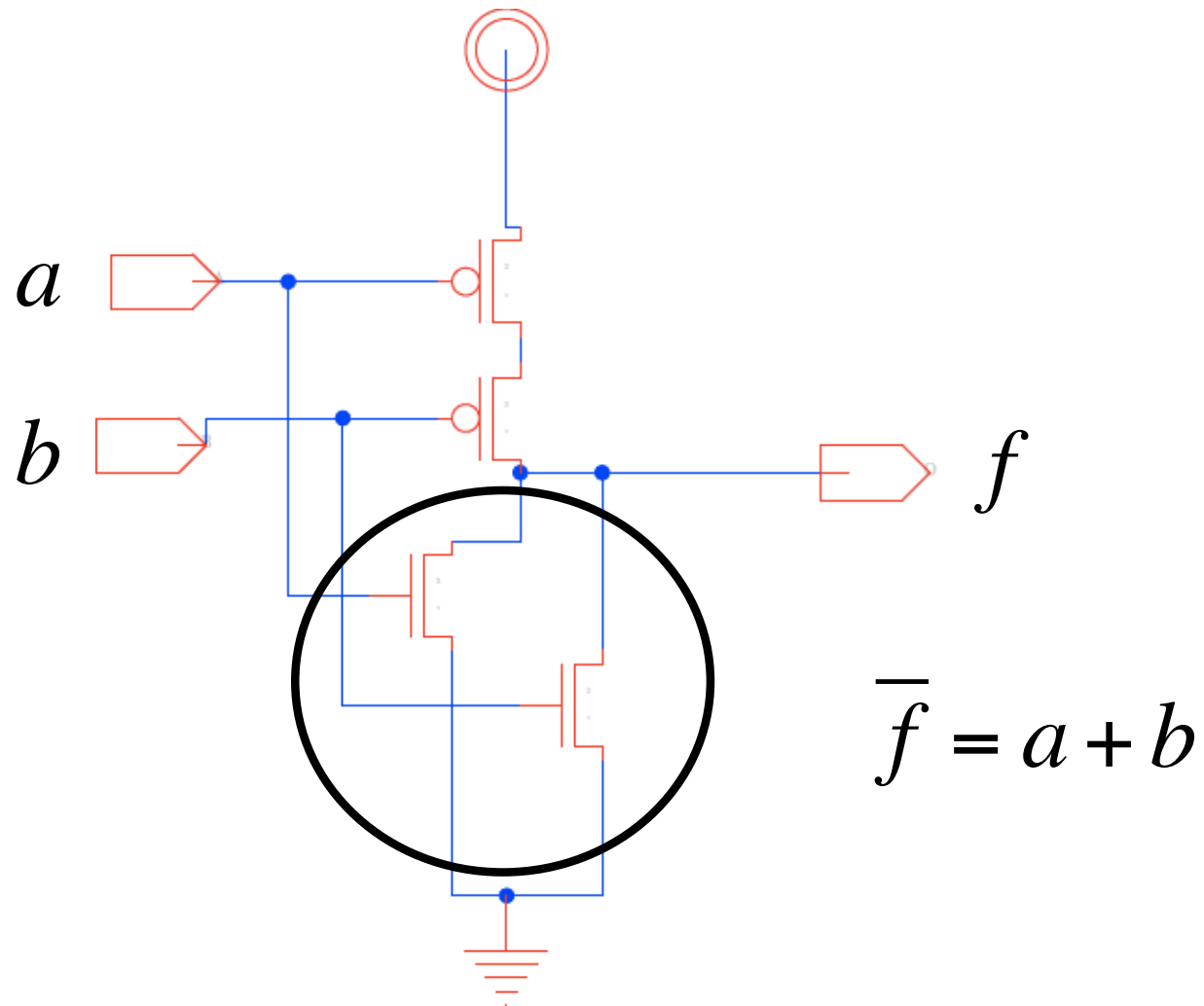


Hint: use zero order model to make a truth table

# What gate is this?

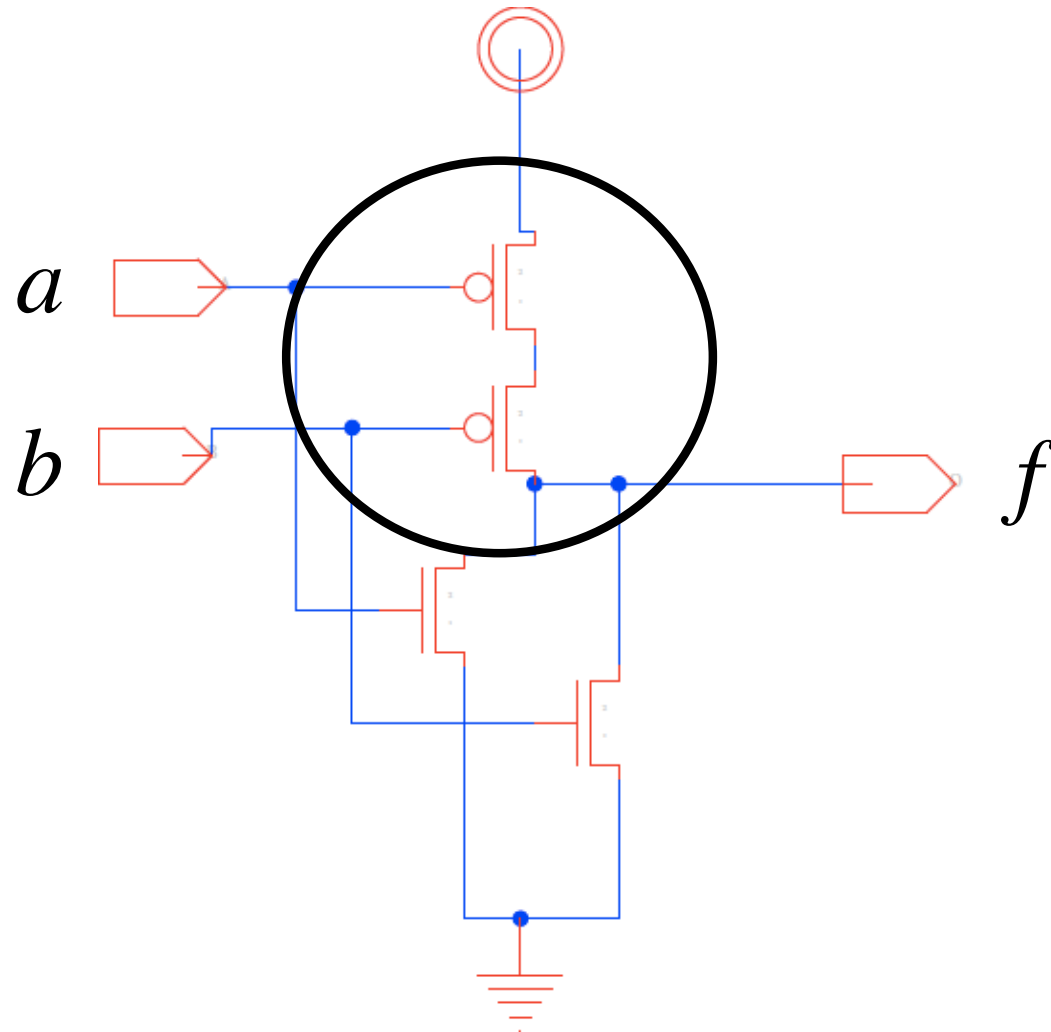


# What gate is this? Preclass 4

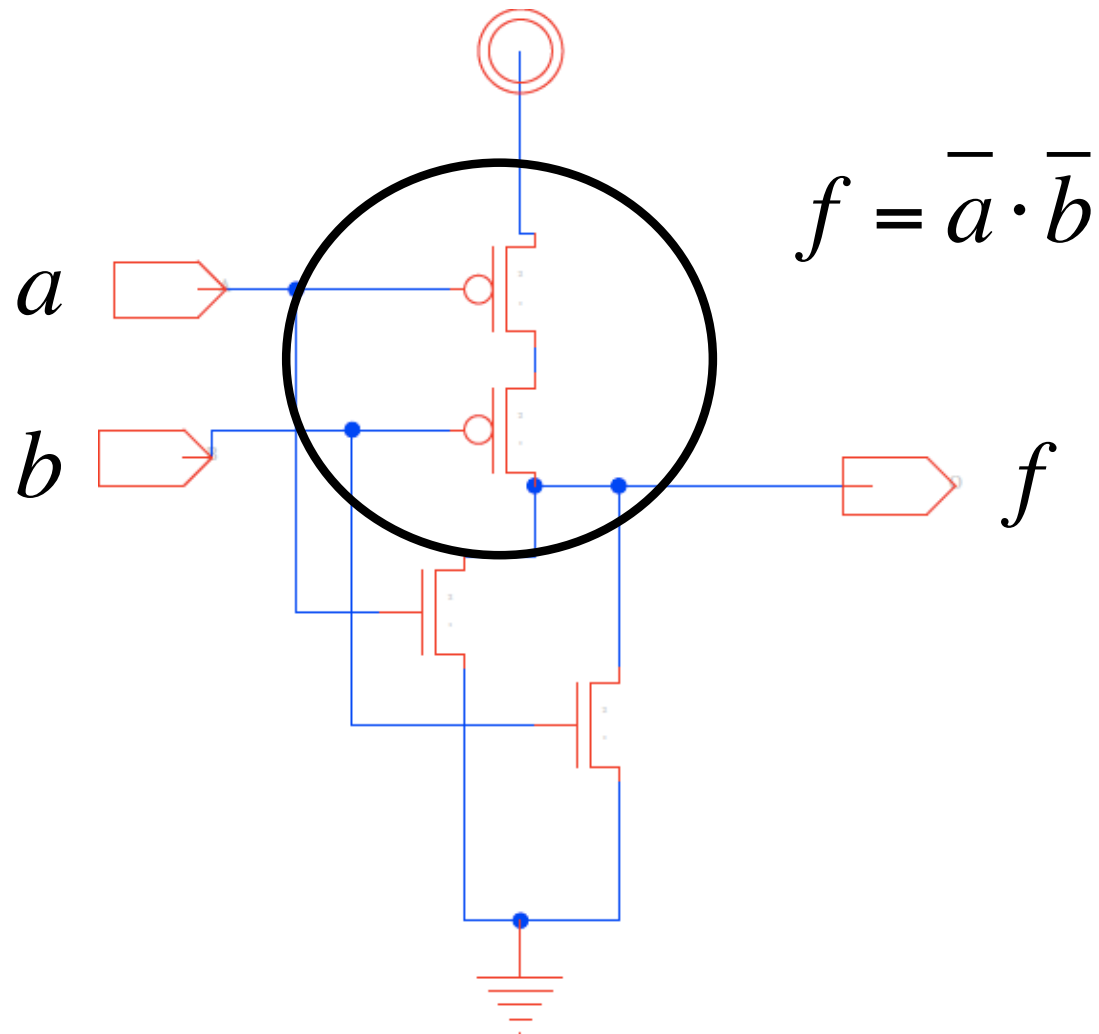


What is  $f$  in minimum-sum-of-products form?

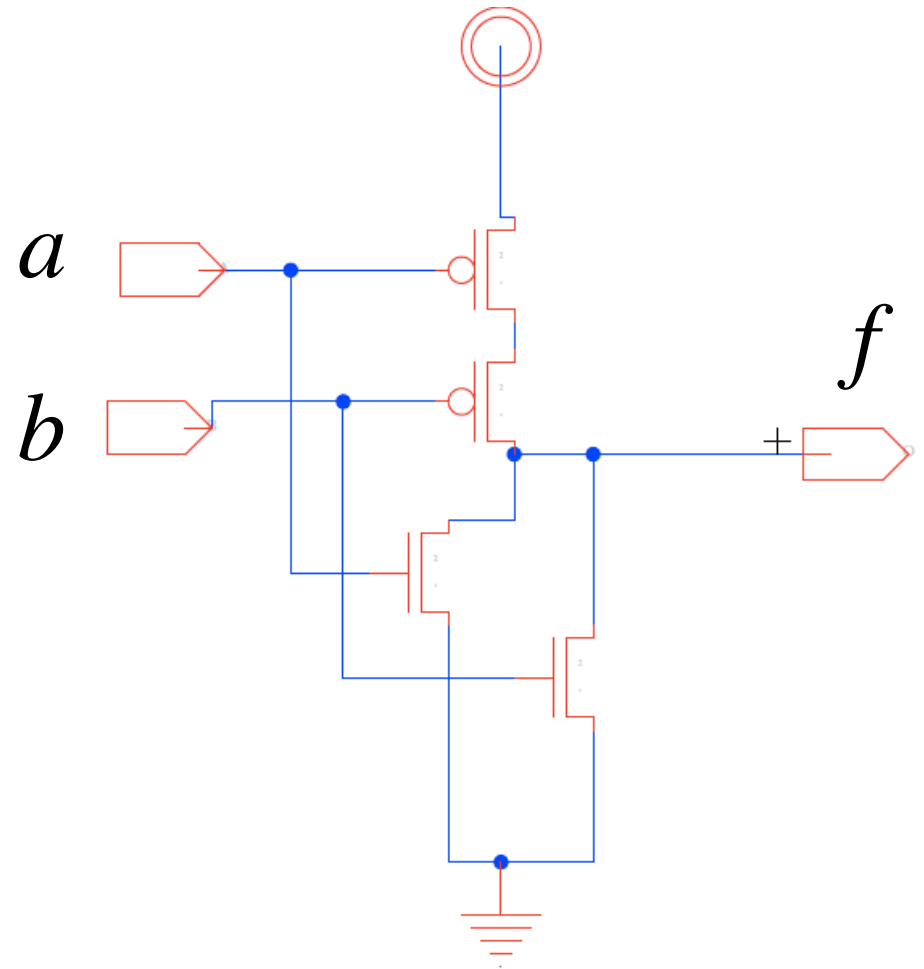
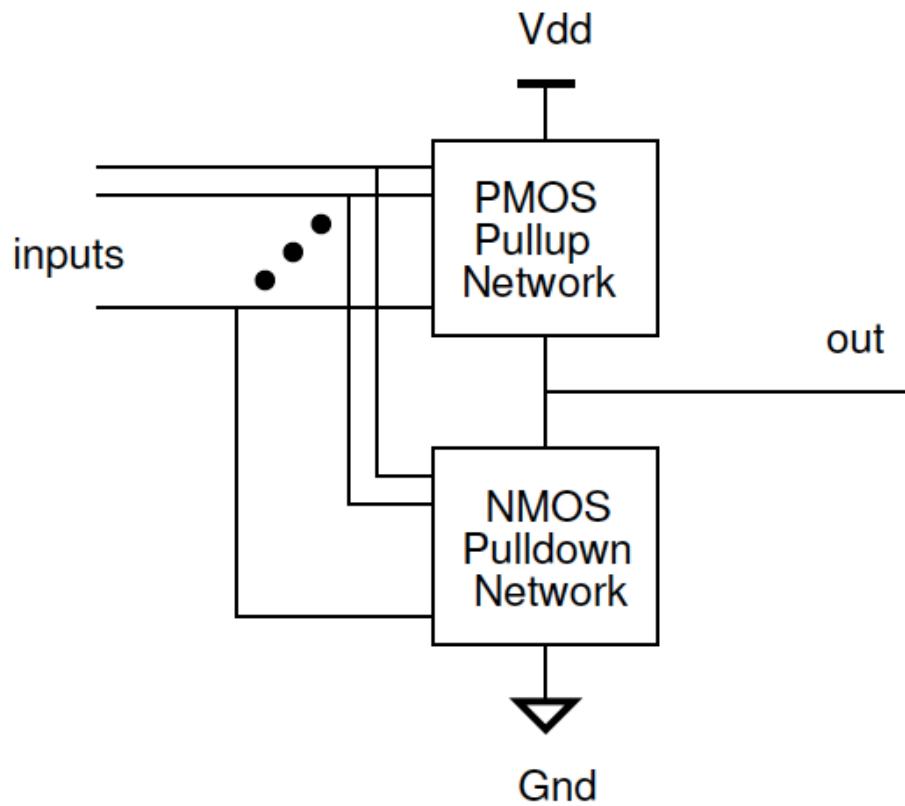
# What gate is this?



# What gate is this?



# Static CMOS Gate Structure





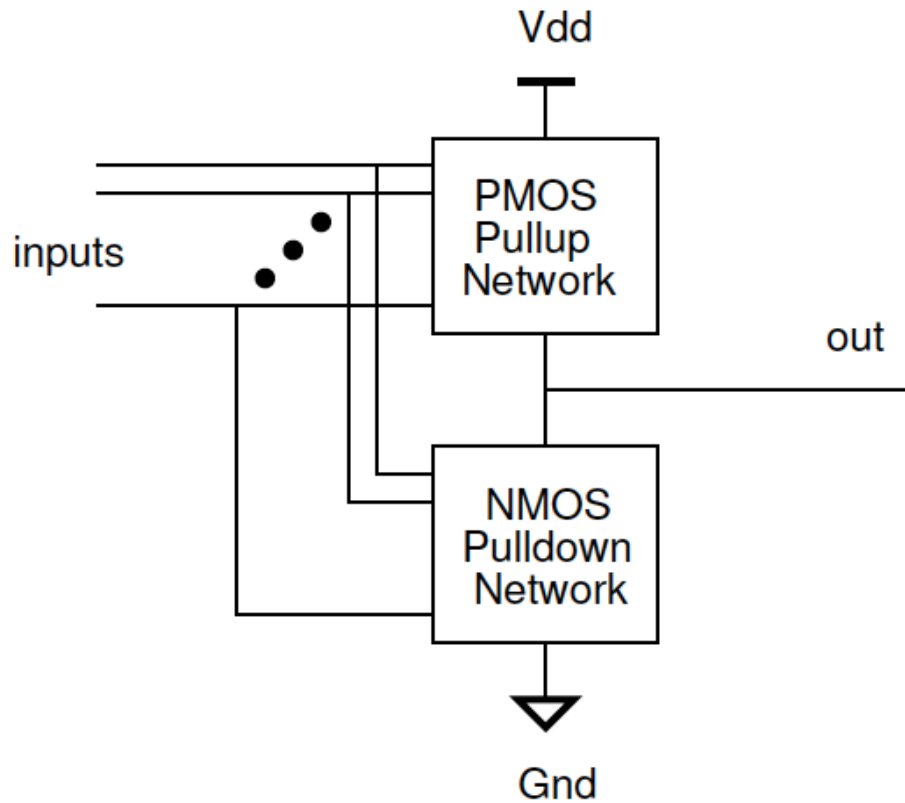
# CMOS

---

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



# Static CMOS Gate Structure



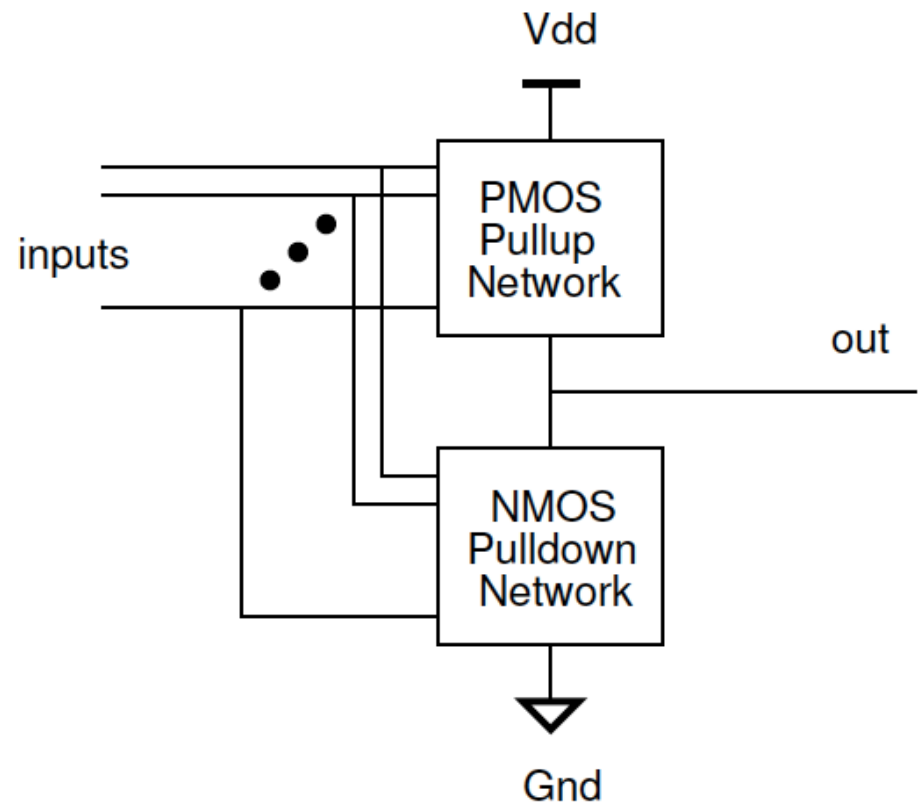
- ❑ Drives rail-to-rail
  - Power rails are  $V_{dd}$  and  $Gnd$
  - output is  $V_{dd}$  or  $Gnd$
- ❑ Input connects to gates
  - load is capacitive
- ❑ Once output node is charged doesn't use energy (no static current)
- ❑ Output actively driven

# Gate Design Example Preclass 5

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

□ Strategy:

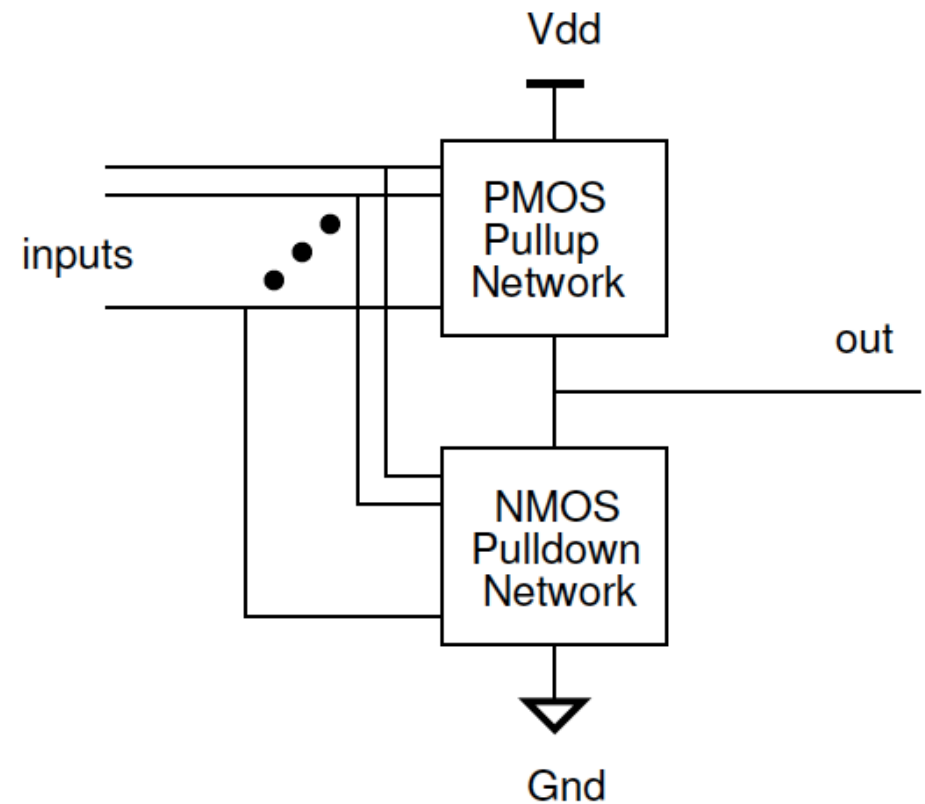
1. Use static CMOS structure
2. Design PMOS pullup for  $f$
3. Use DeMorgan's Law to determine  $f'$
4. Design NMOS pulldown for  $f'$



# Gate Design Example

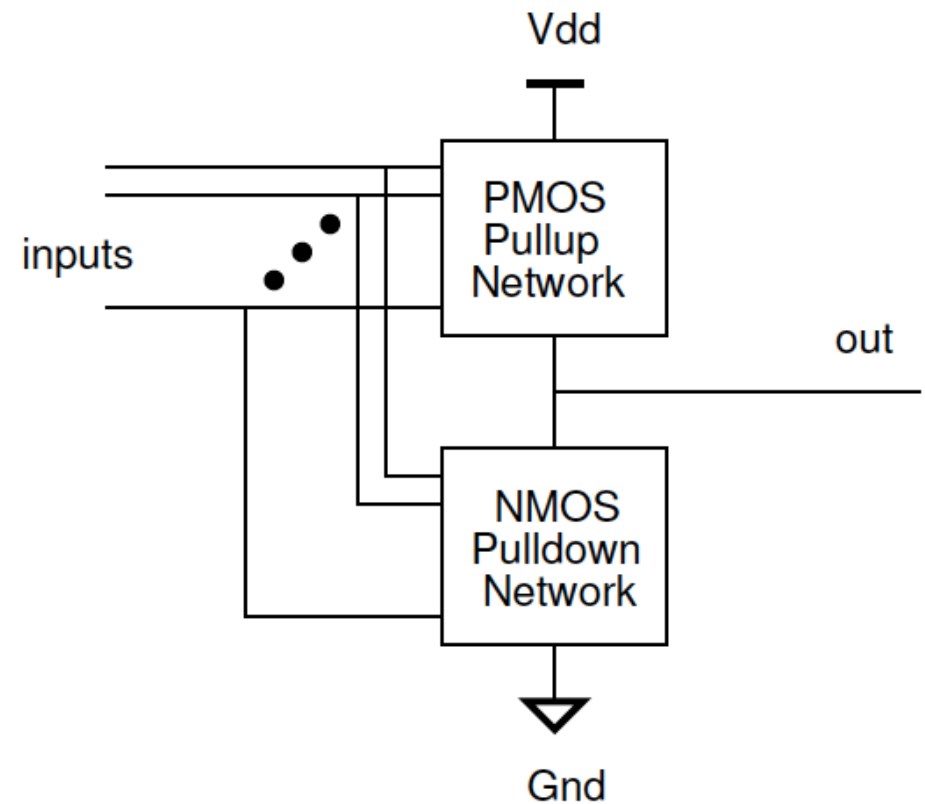
□ Design gate to perform:  $f = (\bar{a} + \bar{b}) \cdot \bar{c}$

□ PMOS Pullup for  $f$ ?



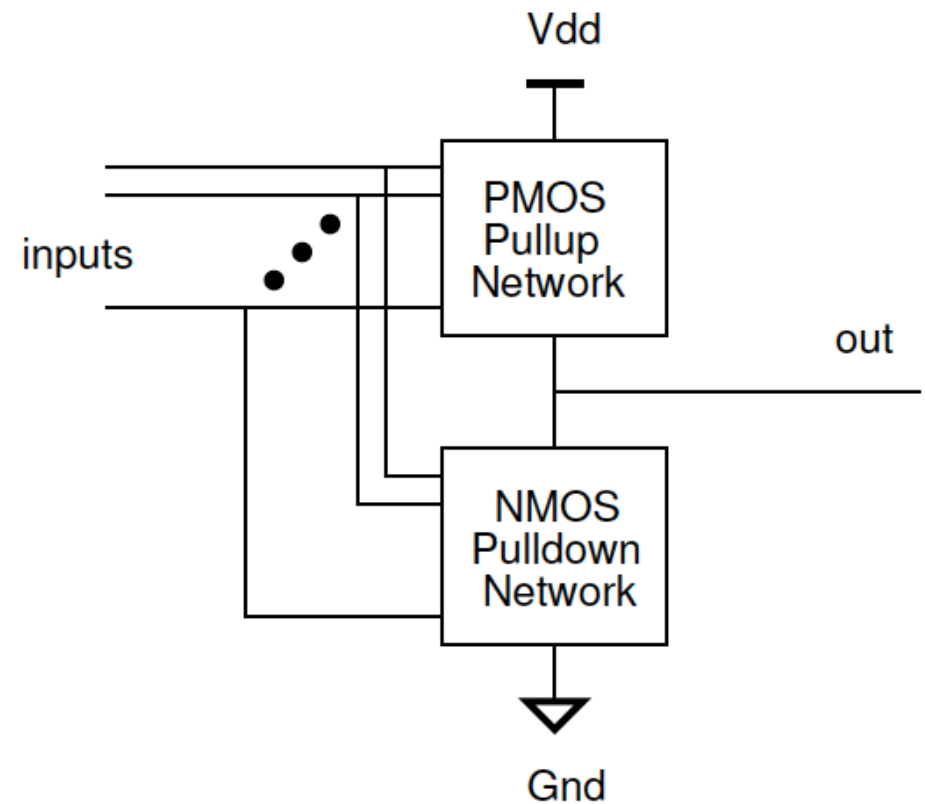
# Gate Design Example

- Design gate to perform:  $f = (\bar{a} + \bar{b}) \cdot \bar{c}$
- What is  $f'$ ?
  - DeMorgan's Law



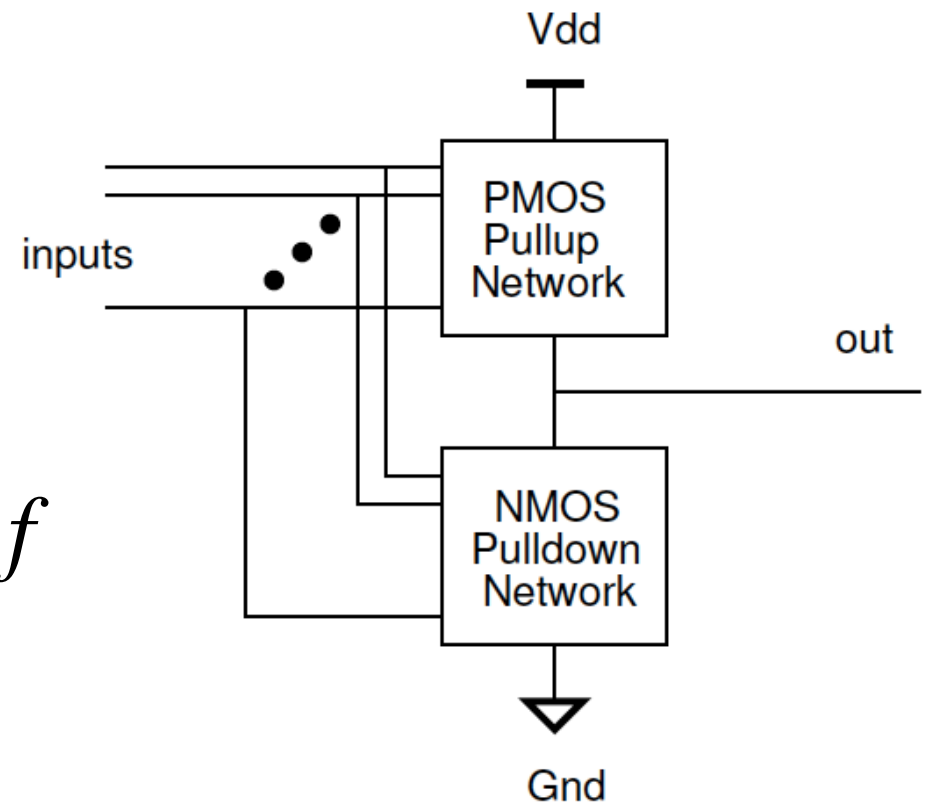
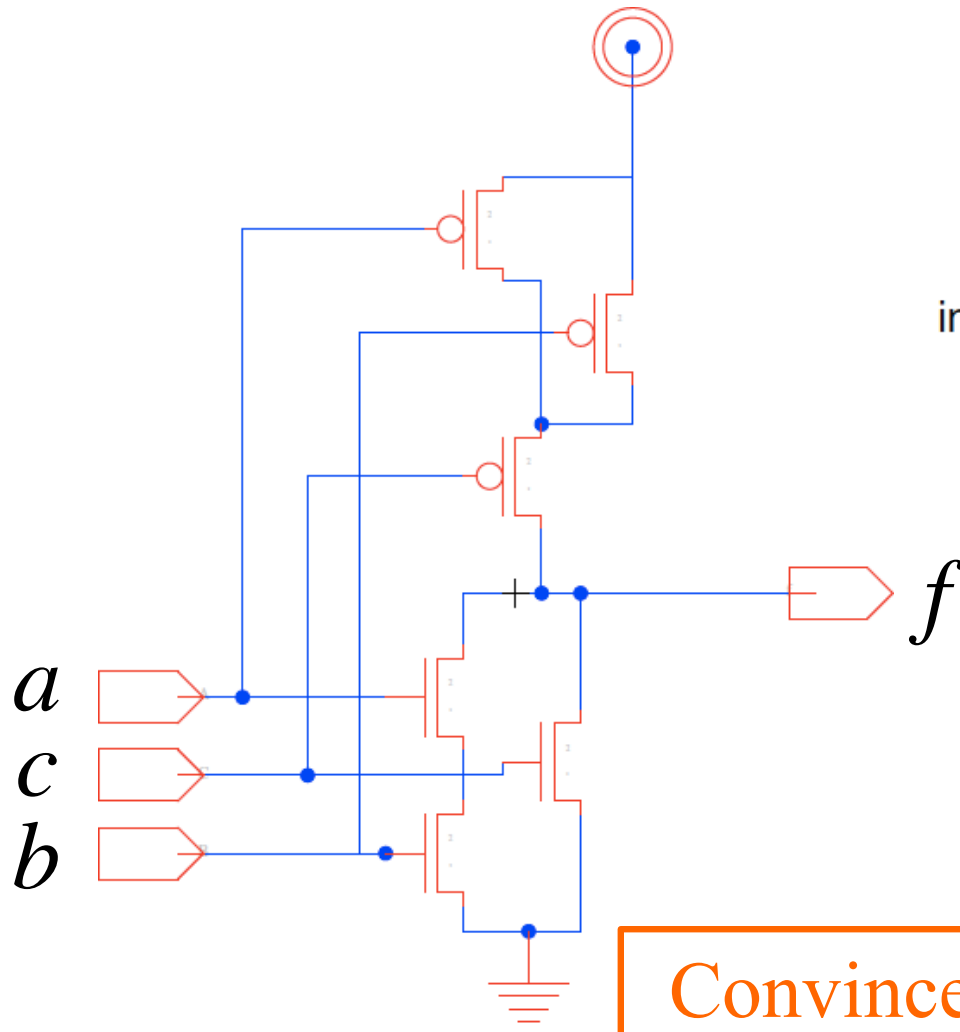
# Gate Design Example

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



# Gate Design Example

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



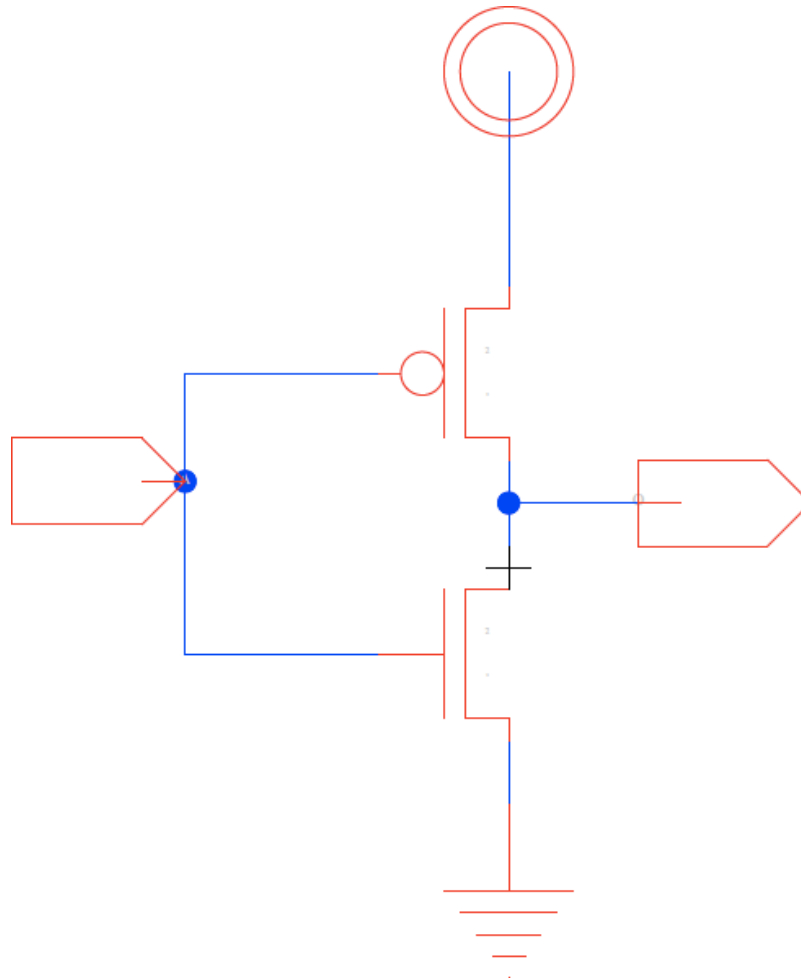
**Convince yourself with a truth table.**



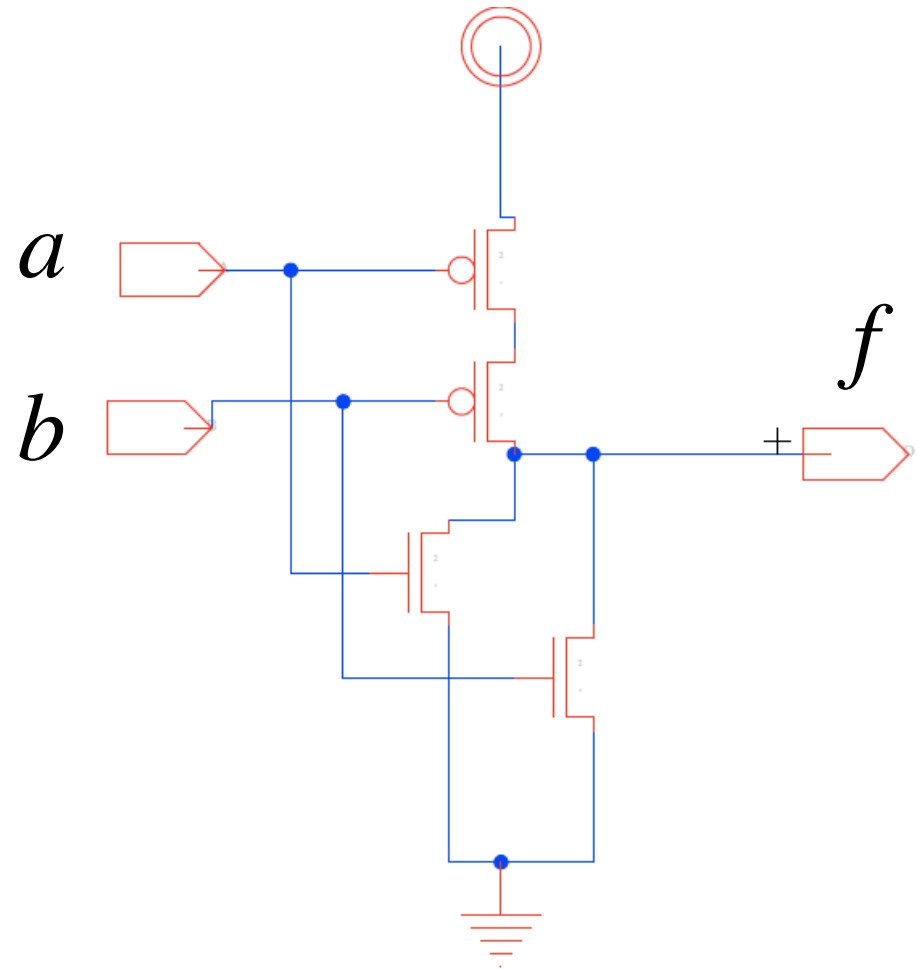
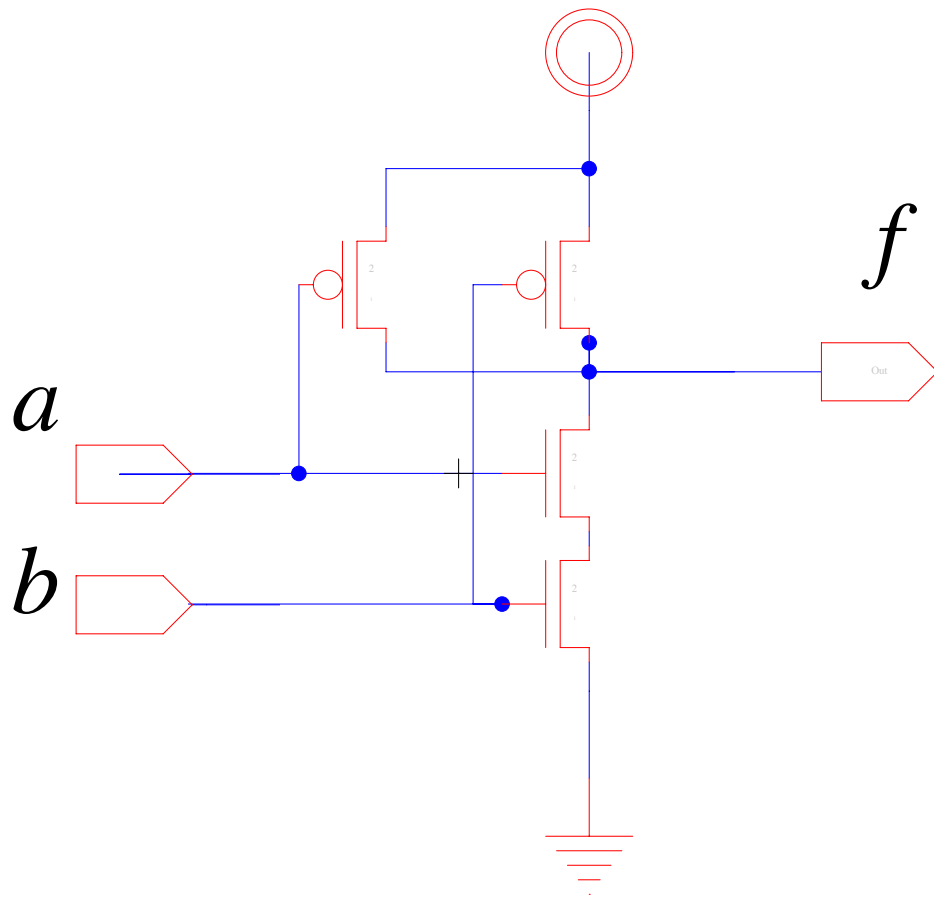
# Inverting Stage

---

- ❑ Each stage of Static CMOS gate is inherently inverting



# NAND/NOR Fundamental Gates





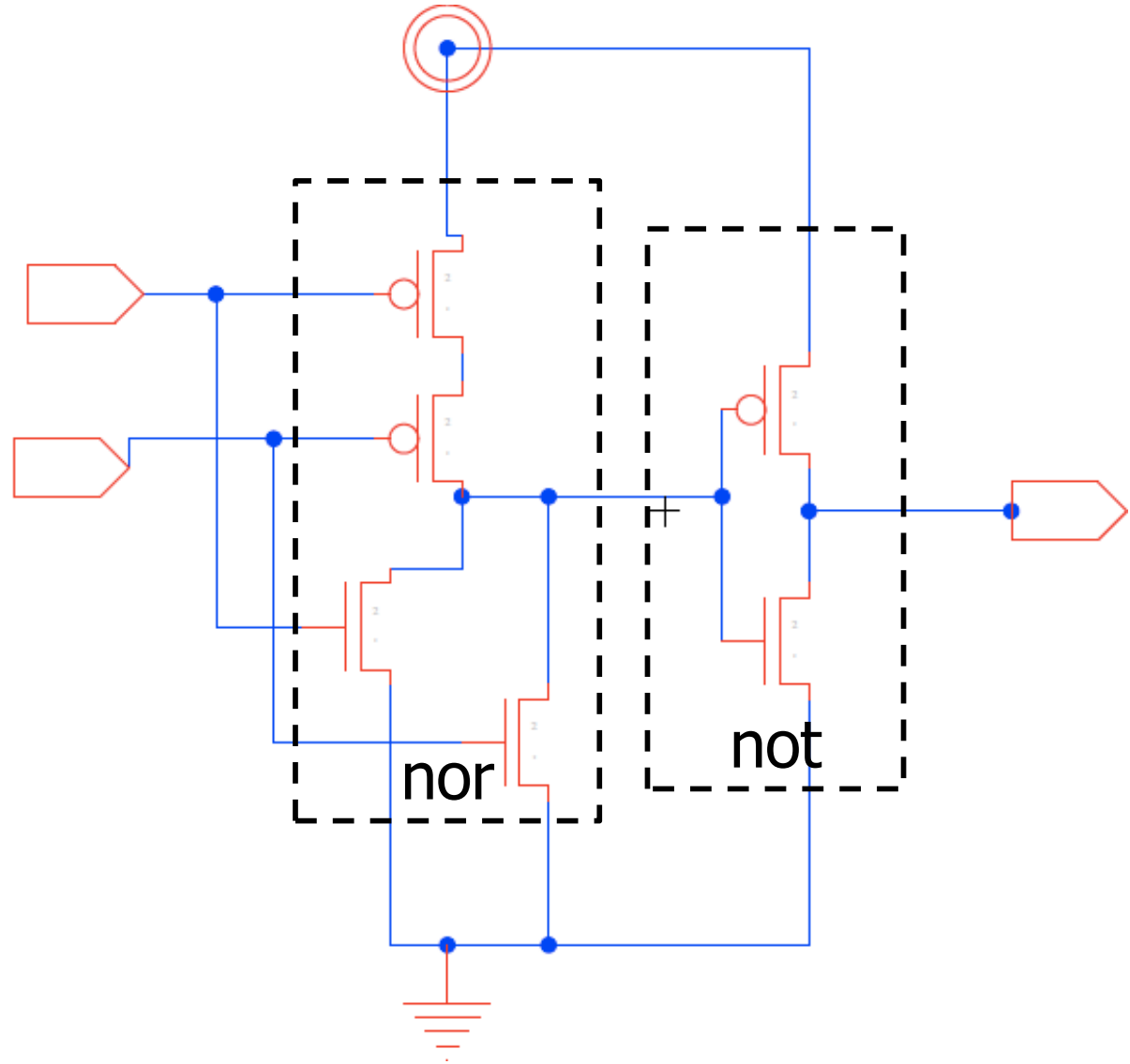


# How implement OR?

---



# 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?

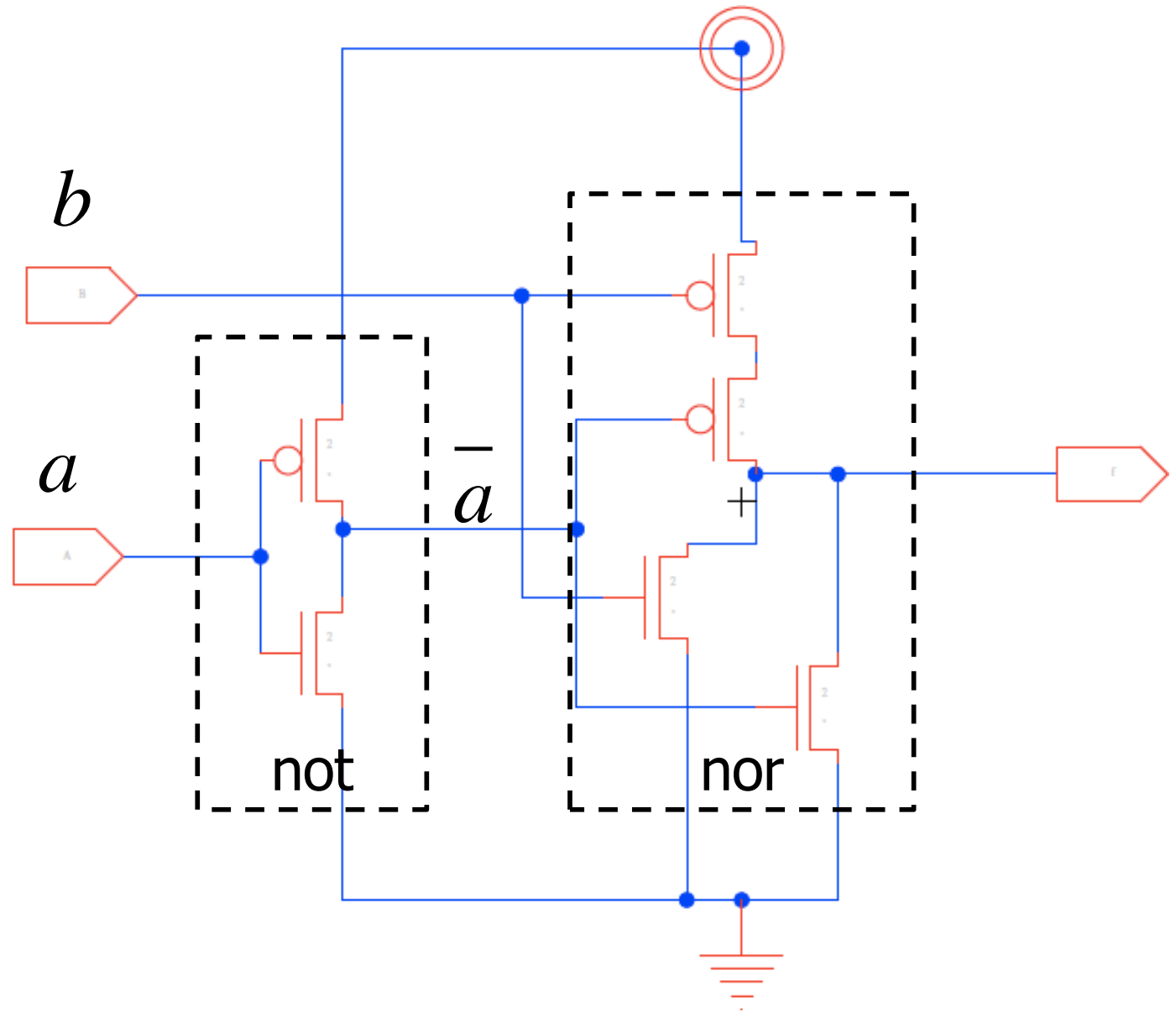
Hint: use cascading stages



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

□ Pullup?

□ Pulldown?





# Big Idea

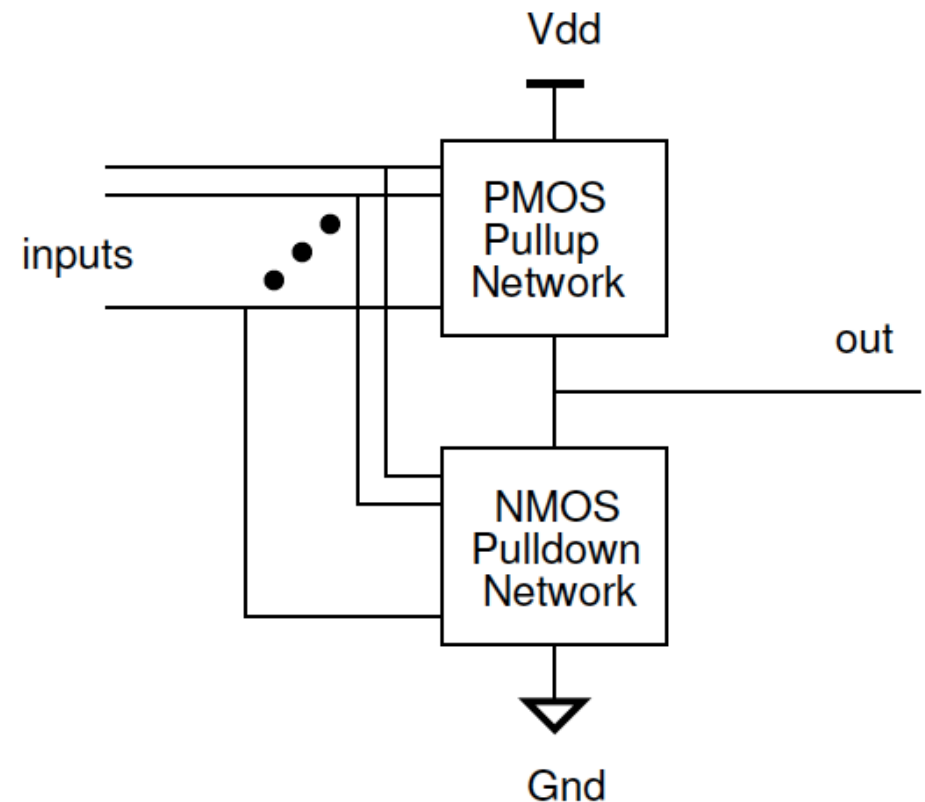
---

- ❑ 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

---

- ❑ HW 1 out – due 1/31
  - Will take time to learn Electric CAD tool, start early
- ❑ Monday (1/27) – no lecture, Lab in Detkin
- ❑ TA office hours announced on Ed Discussion (Mon- Friday)





# Acknowledgement

---

- ❑ Prof. André DeHon (University of Pennsylvania)
- ❑ Prof. Tania Khanna (University of Pennsylvania)