

# Lecture 4: Circuits & Layout

# Assignment for 9/12

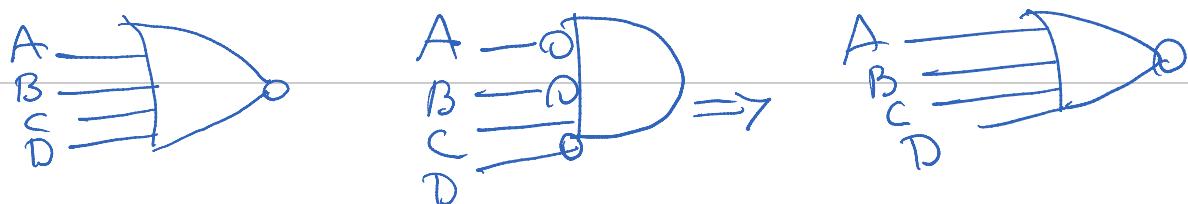
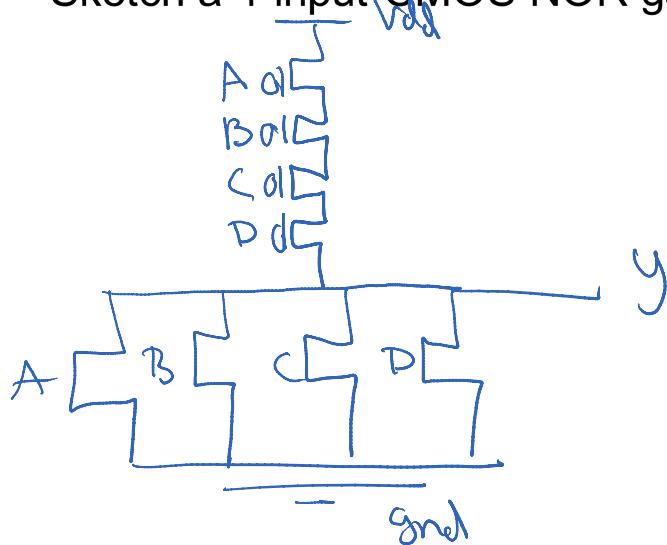
- Complete Lab2
- Text book reading sections 1.1 – 1.5 (28 pages)
- Prepare for quiz 2 (based on today's lecture) *on 9/10*

# Outline

- CMOS Gate Design
- Pass Transistors
- CMOS Latches & Flip-Flops
- Standard Cell Layouts
- Stick Diagrams

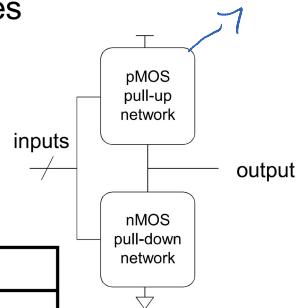
# CMOS Gate Design

- Activity:  $\overline{A+B+C+D} = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}}$
- Sketch a 4-input CMOS NOR gate

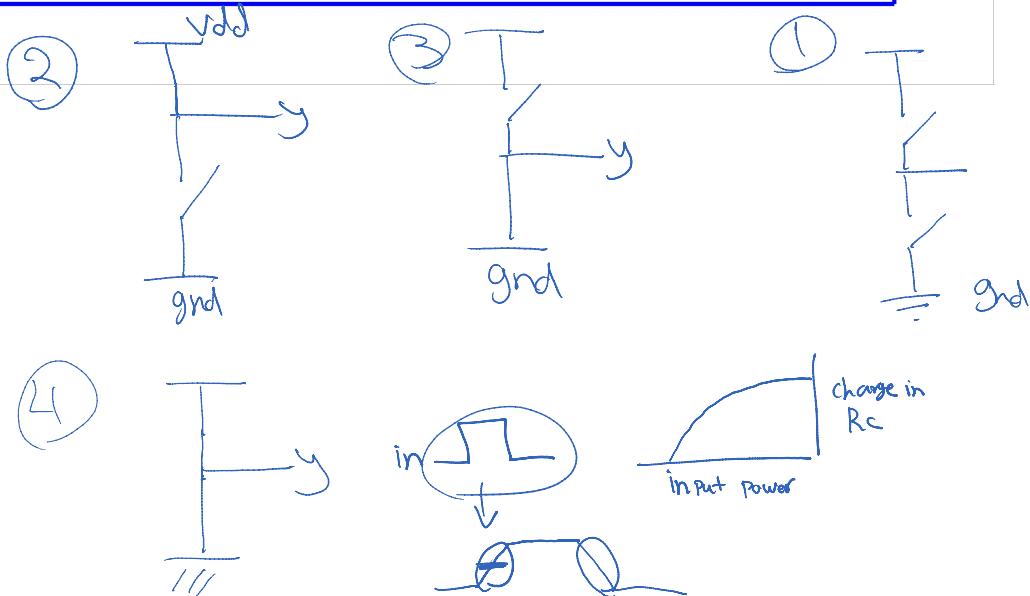


# Complementary CMOS

- Complementary CMOS logic gates
  - nMOS pull-down network
  - pMOS pull-up network
  - a.k.a. static CMOS

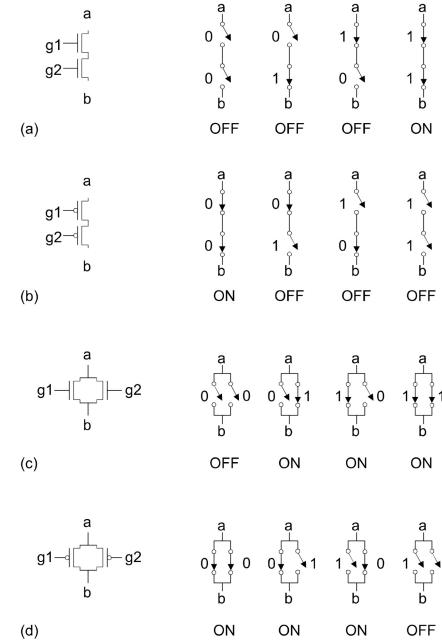


	Pull-up OFF	Pull-up ON
Pull-down OFF	Z (float) ①	1 ②
Pull-down ON	0 ③	X (crowbar) ④



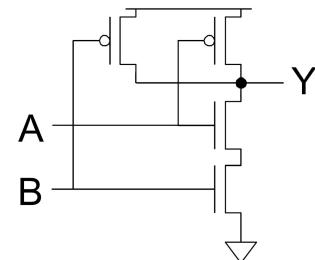
# Series and Parallel

- nMOS: 1 = ON
- pMOS: 0 = ON
- Series: both must be ON
- Parallel: either can be ON



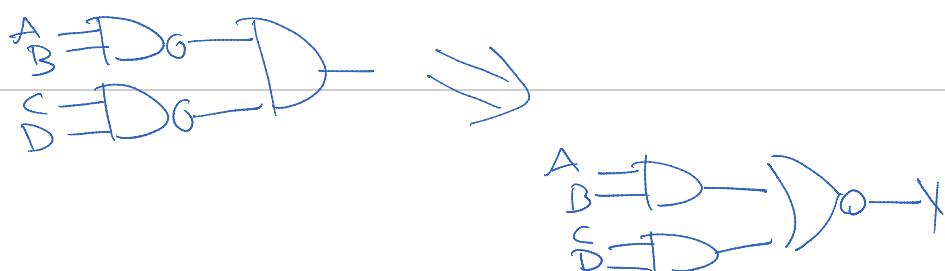
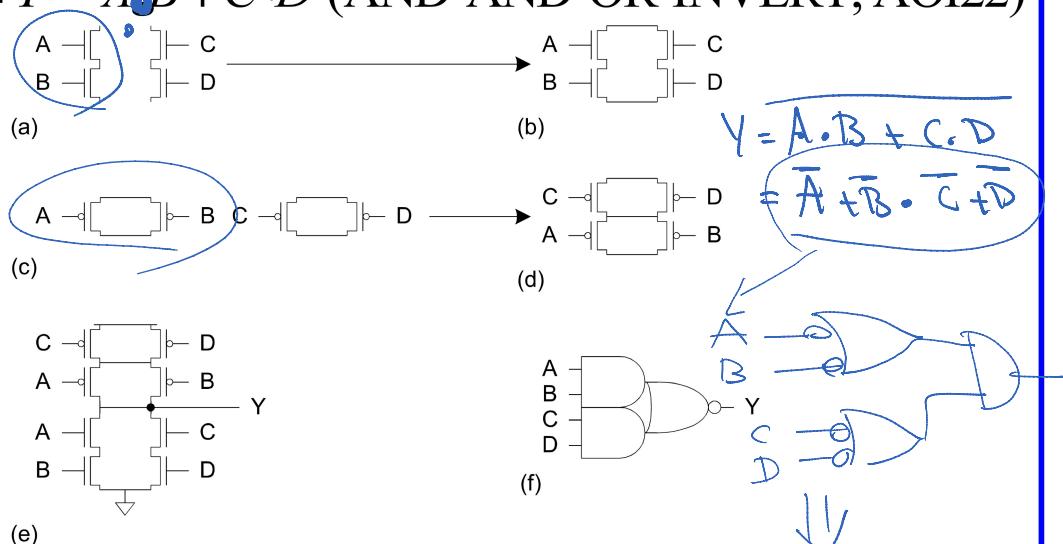
# Conduction Complement

- Complementary CMOS gates always produce 0 or 1
- Ex: NAND gate
  - Series nMOS:  $Y=0$  when both inputs are 1
  - Thus  $Y=1$  when either input is 0
  - Requires parallel pMOS
- Rule of *Conduction Complements*
  - Pull-up network is complement of pull-down
  - Parallel  $\rightarrow$  series, series  $\rightarrow$  parallel



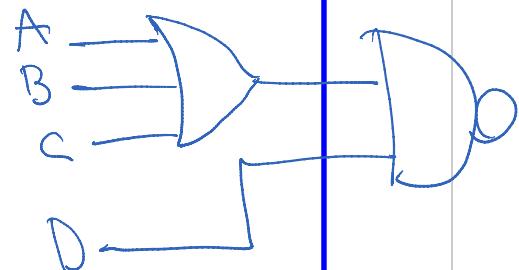
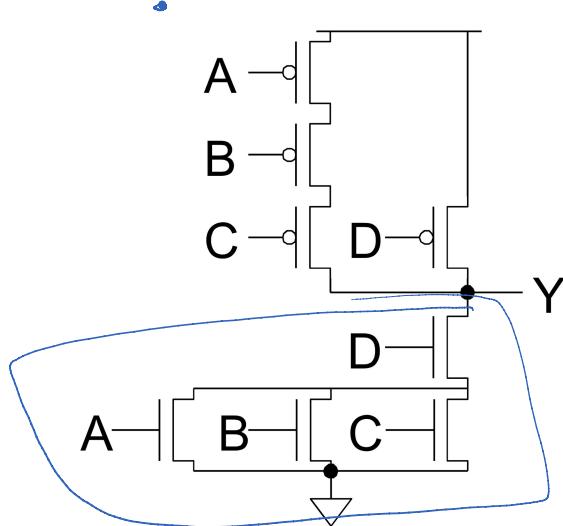
# Compound Gates

- Compound gates can do any inverting function
- Ex:  $Y = \overline{A \cdot B} + C \cdot D$  (AND-AND-OR-INVERT, AOI22)



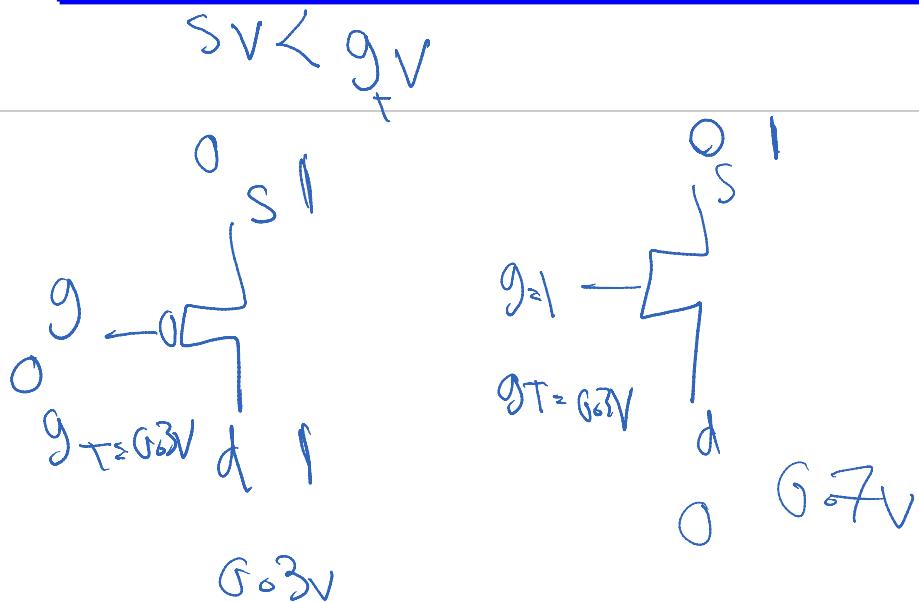
## Example: O3AI

□  $Y = \overline{(A + B + C)}D$



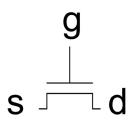
# Signal Strength

- *Strength of signal*
  - How close it approximates ideal voltage source
- $V_{DD}$  and GND rails are strongest 1 and 0
- nMOS pass strong 0
  - But degraded or weak 1
- pMOS pass strong 1
  - But degraded or weak 0
- Thus nMOS are best for pull-down network



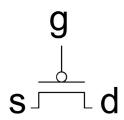
# Pass Transistors

- Transistors can be used as switches



$g = 0$   
 $s \rightarrow o \circ d$   
 $g = 1$   
 $s \rightarrow o \circ d$

Input  $g = 1$  Output  
 $0 \rightarrow o \circ$  strong 0  
 $g = 1$   
 $1 \rightarrow o \circ$  degraded 1

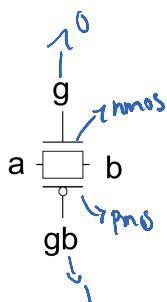


$g = 0$   
 $s \rightarrow o \circ d$   
 $g = 1$   
 $s \rightarrow o \circ d$

Input  $g = 0$  Output  
 $0 \rightarrow o \circ$  degraded 0  
 $g = 0$   
 $1 \rightarrow o \circ$  strong 1

# Transmission Gates

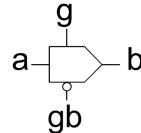
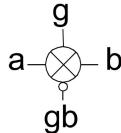
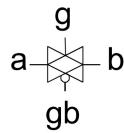
- Pass transistors produce degraded outputs
- *Transmission gates* pass both 0 and 1 well



$g = 0, gb = 1$   
 $a \circlearrowright \rightarrow b$

$g = 1, gb = 0$   
 $a \circlearrowright \rightarrow b$

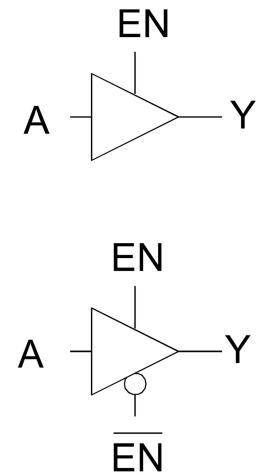
Input	Output
$g = 1, gb = 0$ $0 \circlearrowright \rightarrow$	$strong\ 0$
$g = 1, gb = 0$ $1 \circlearrowright \rightarrow$	$strong\ 1$



# Tristates

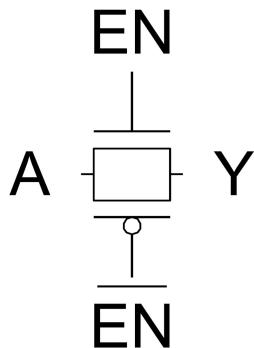
- Tristate buffer produces Z when not enabled

EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1



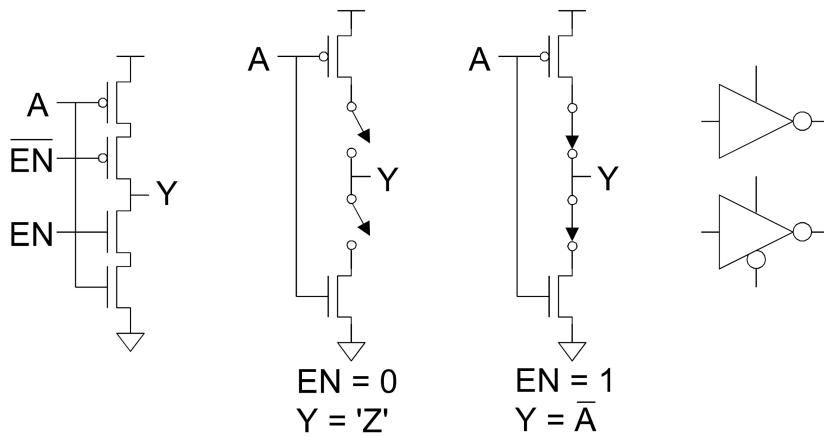
# Nonrestoring Tristate

- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y



# Tristate Inverter

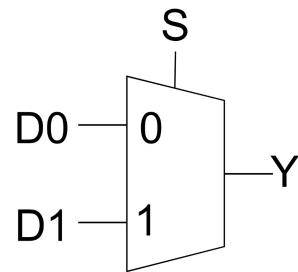
- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



# Multiplexers

- ~~4:1~~ multiplexer chooses between two inputs  
2:1

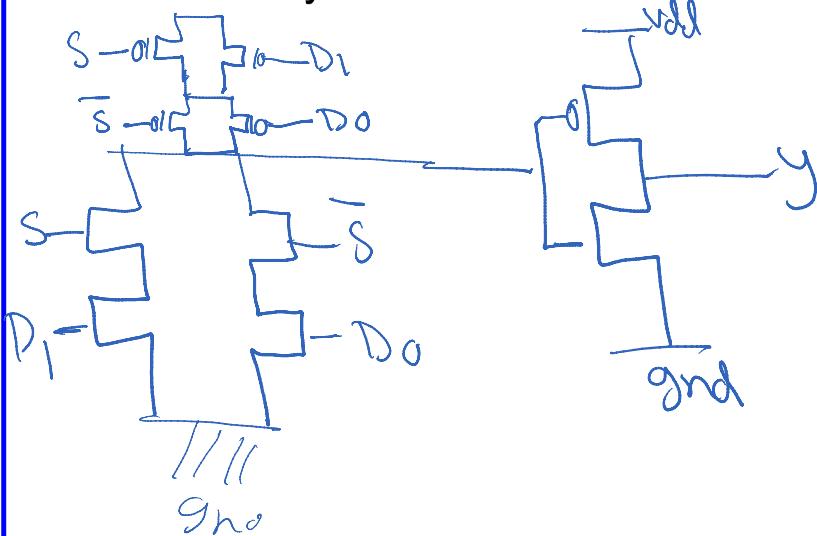
S	D1	D0	Y
0	X	0	D <sub>0,0</sub>
0	X	1	D <sub>0,1</sub>
1	0	X	D <sub>1,0</sub>
1	1	X	D <sub>1,1</sub>



# Gate-Level Mux Design

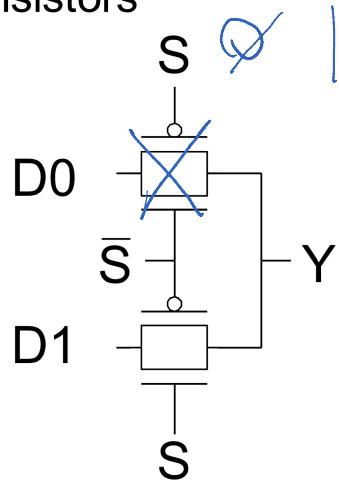
- $Y = SD_1 + \bar{S}D_0$  (too many transistors)
- How many transistors are needed?

$$Y = S \cdot D_1 + \bar{S} \cdot D_0$$



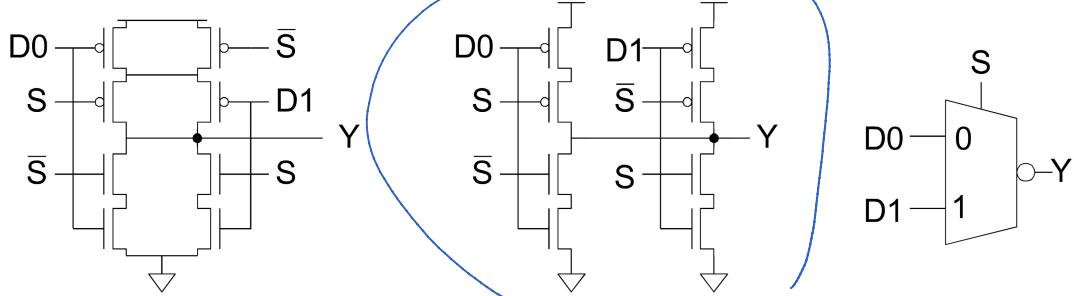
# Transmission Gate Mux

- ❑ Nonrestoring mux uses two transmission gates
  - Only 4 transistors



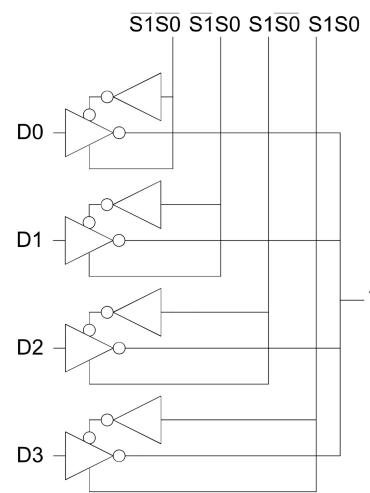
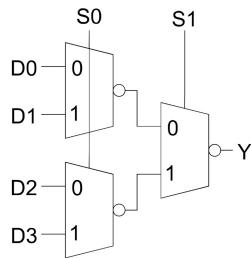
# Inverting Mux

- Inverting multiplexer
  - Use compound AOI22
  - Or pair of tristate inverters
  - Essentially the same thing
- Noninverting multiplexer adds an inverter



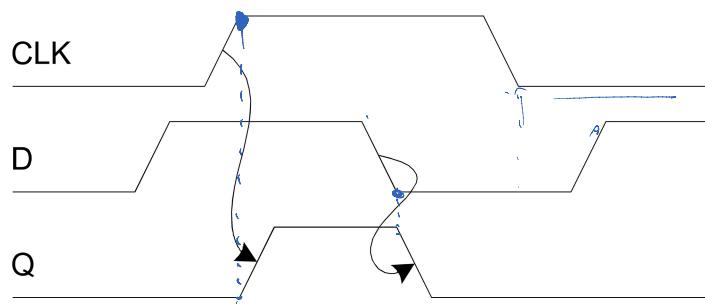
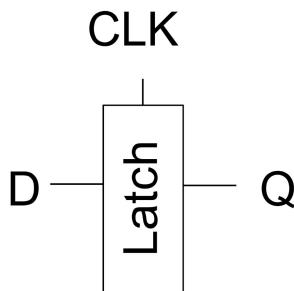
# 4:1 Multiplexer

- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 4:1 muxes
  - Or four tristates



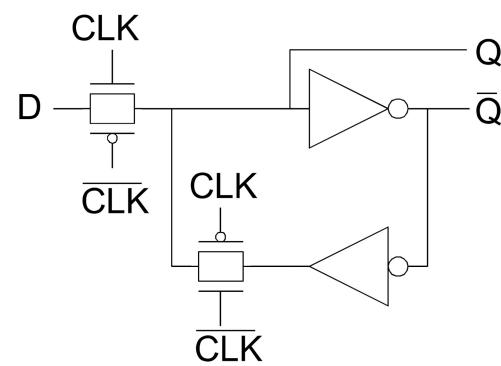
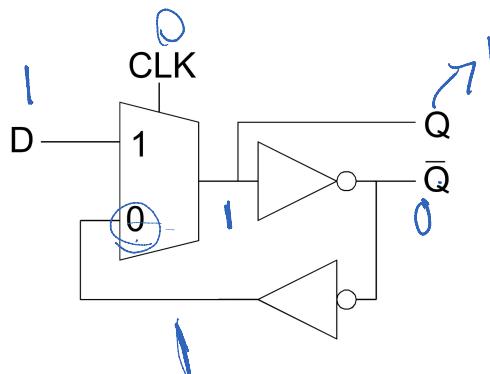
# D Latch

- When  $\text{CLK} = 1$ , latch is *transparent*
  - D flows through to Q like a buffer
- When  $\text{CLK} = 0$ , the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

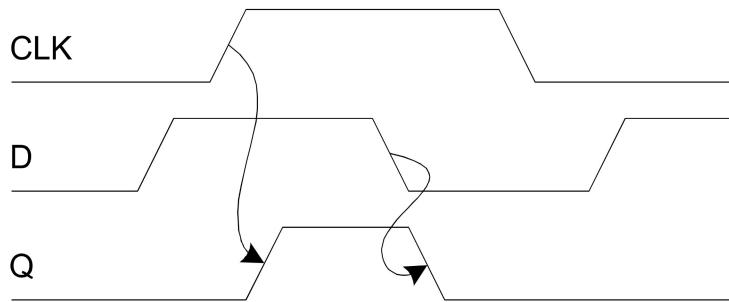
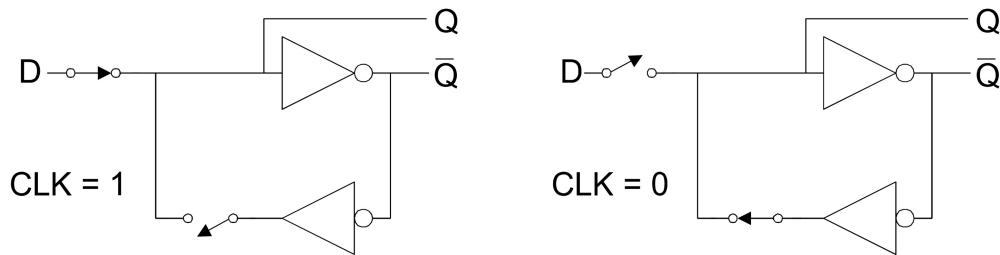


# D Latch Design

- Multiplexer chooses D or old Q

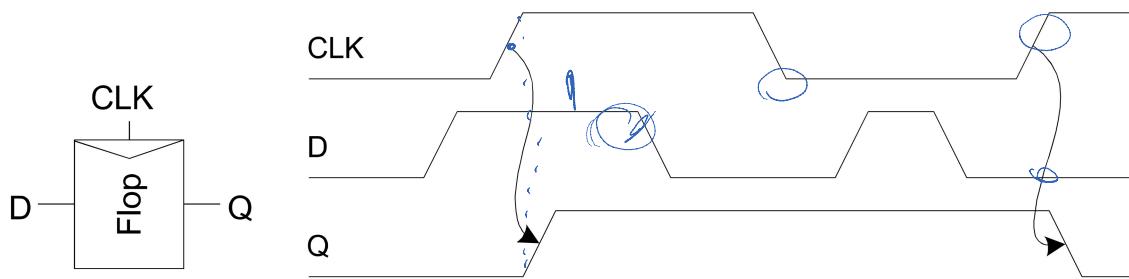


# D Latch Operation



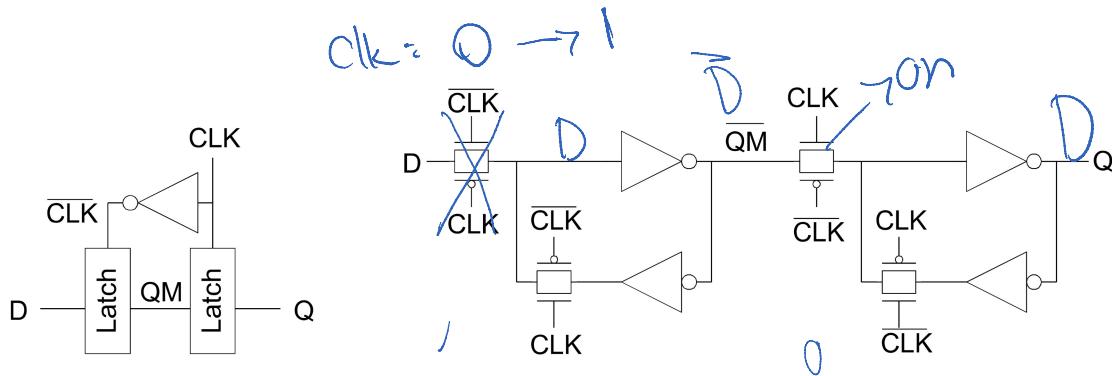
# D Flip-flop

- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop, master-slave flip-flop*

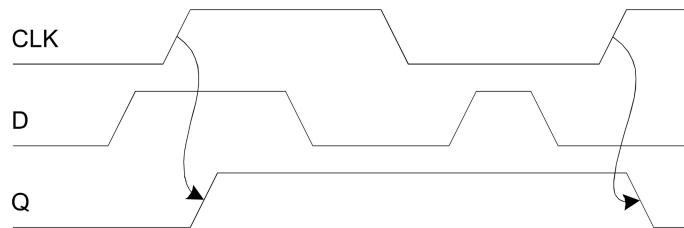
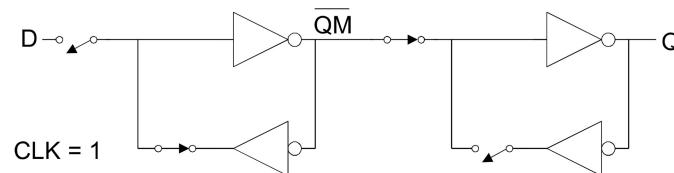
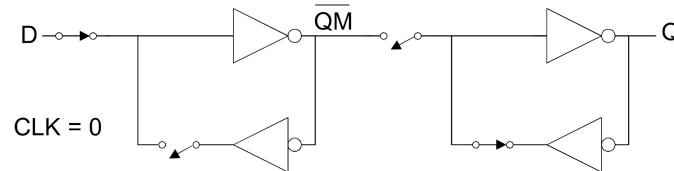


# D Flip-flop Design

- Built from master and slave D latches

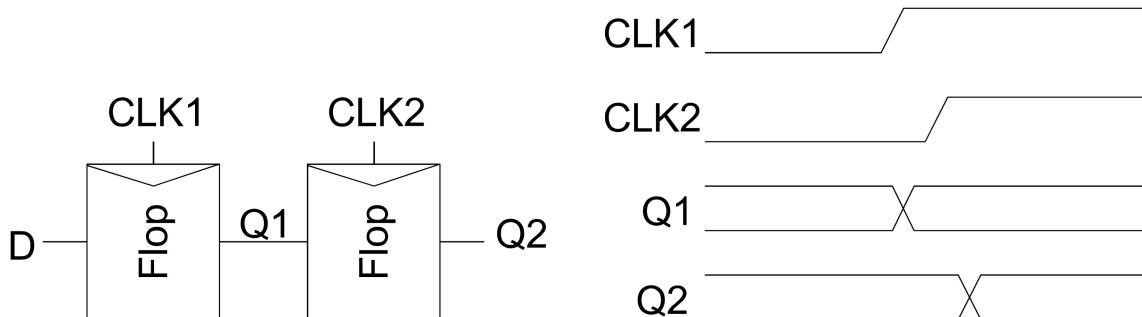


# D Flip-flop Operation



# Race Condition

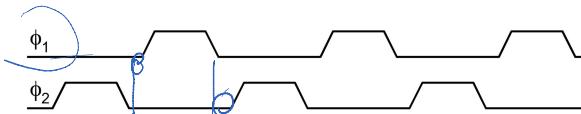
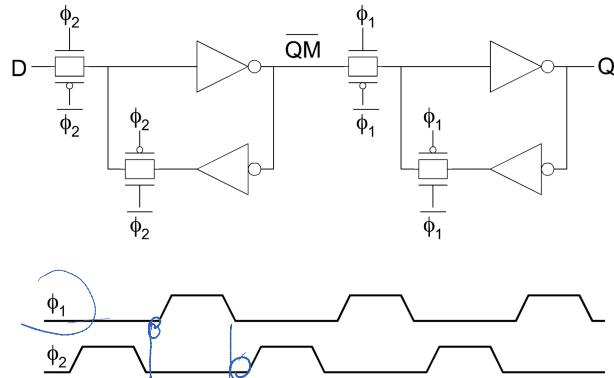
- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*



$\text{clk}$   $\rightarrow$   $\overline{\text{clk}}$

# Nonoverlapping Clocks

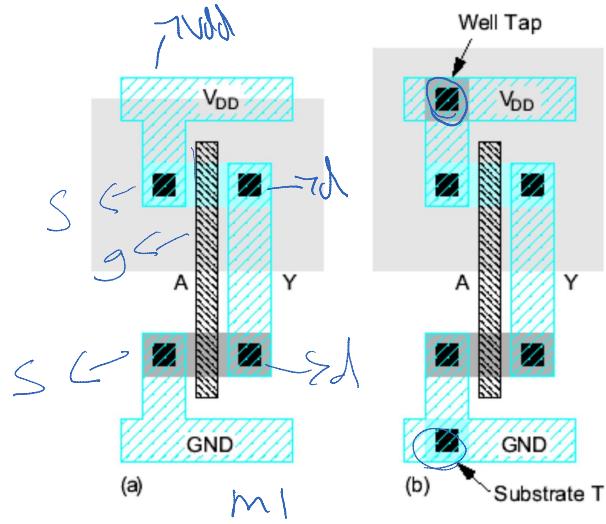
- Nonoverlapping clocks can prevent races
  - As long as nonoverlap exceeds clock skew
- We will use them in this class for safe design
  - Industry manages skew more carefully instead



# Gate Layout

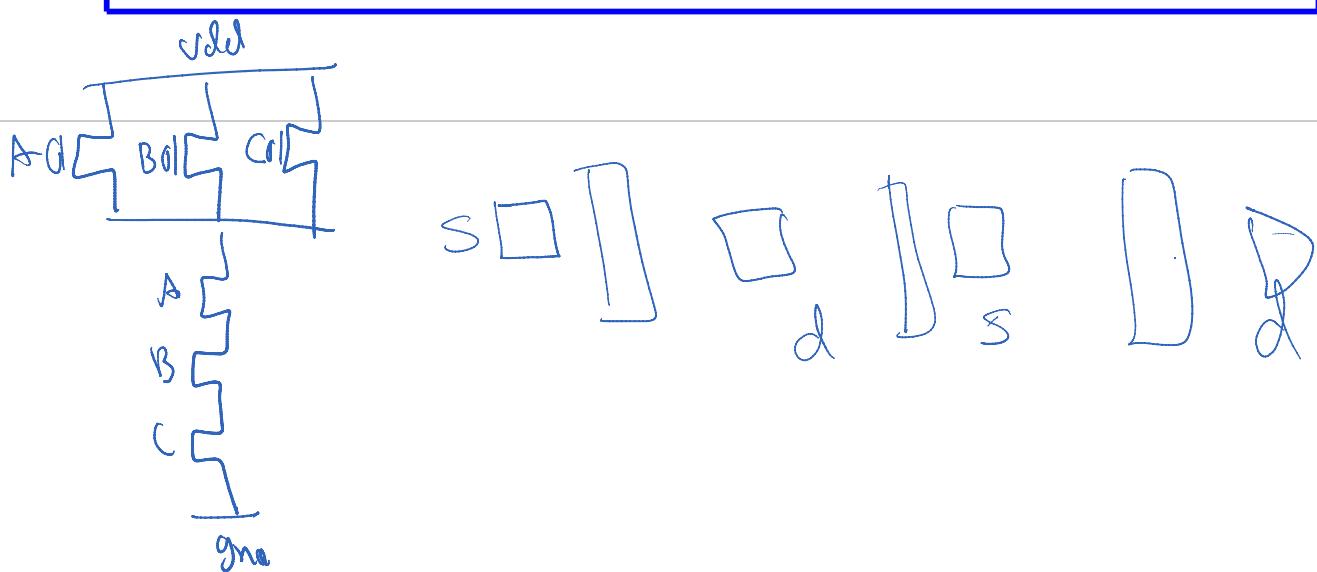
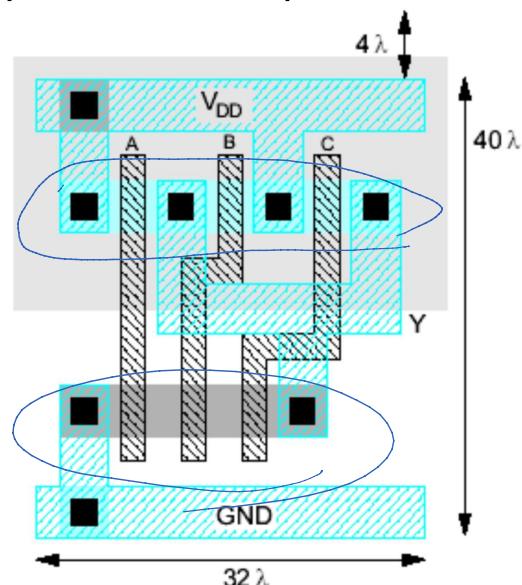
- Layout can be very time consuming
  - Design gates to fit together nicely
  - Build a library of standard cells
- Standard cell design methodology
  - $V_{DD}$  and GND should abut (standard height)
  - Adjacent gates should satisfy design rules
  - nMOS at bottom and pMOS at top
  - All gates include well and substrate contacts

# Example: Inverter



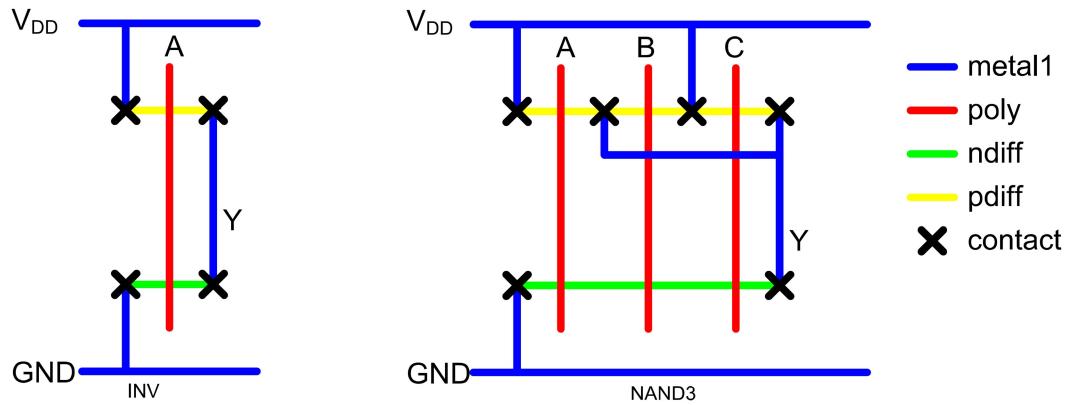
# Example: NAND3

- Horizontal N-diffusion and p-diffusion strips
- Vertical polysilicon gates
- Metal1  $V_{DD}$  rail at top
- Metal1 GND rail at bottom
- $32 \lambda$  by  $40 \lambda$



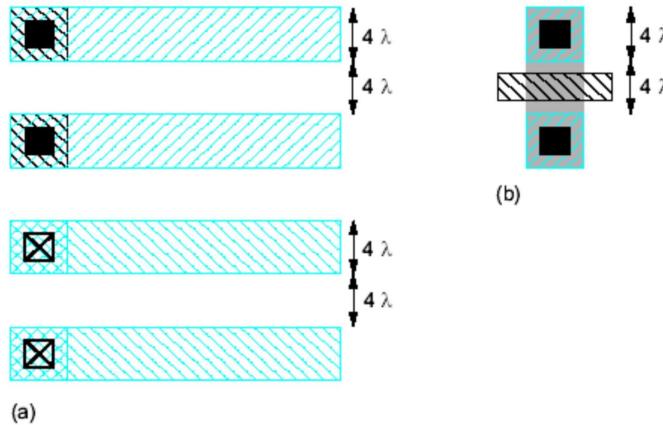
# Stick Diagrams

- Stick diagrams help plan layout quickly
  - Need not be to scale
  - Draw with color pencils or dry-erase markers



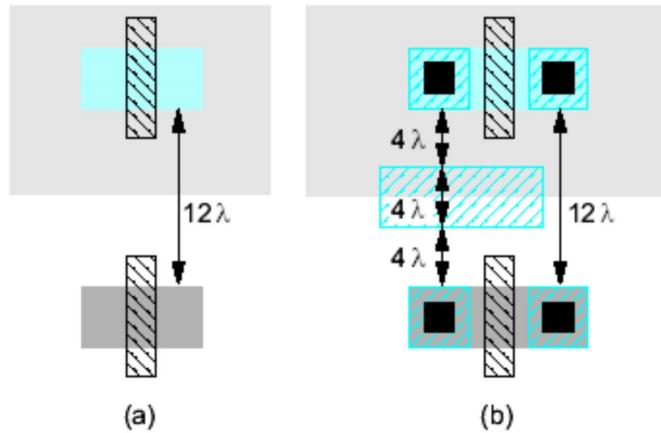
# Wiring Tracks

- A *wiring track* is the space required for a wire
  - $4 \lambda$  width,  $4 \lambda$  spacing from neighbor =  $8 \lambda$  pitch
- Transistors also consume one wiring track



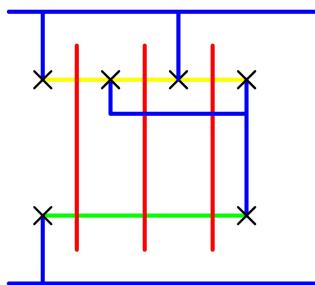
# Well spacing

- Wells must surround transistors by  $6 \lambda$ 
  - Implies  $12 \lambda$  between opposite transistor flavors
  - Leaves room for one wire track



# Area Estimation

- Estimate area by counting wiring tracks
  - Multiply by 8 to express in  $\lambda$



# Example: O3AI

- Sketch a stick diagram for O3AI and estimate area

- $Y = \overline{(A + B + C)}D$

