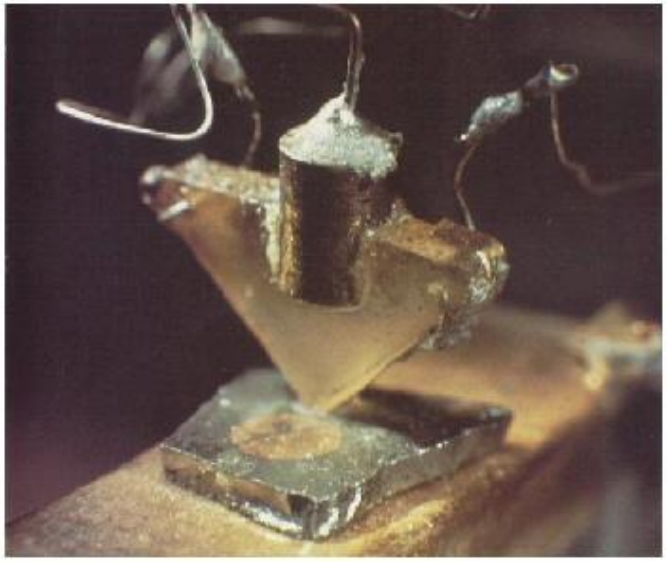
Lecture 2:
CMOS logic design and layout
DC response

Invention of the Transistor

- Vacuum tubes ruled in first half of 20th century: Large, expensive, power-hungry, unreliable
- 1947: first point contact transistor
  - John Bardeen and Walter Brattain at Bell Labs
  - Read *Crystal Fire* by Riordan, Hoddeson

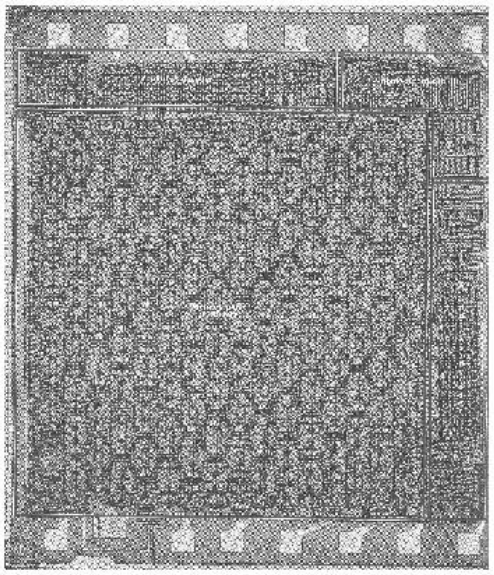


Transistor Types

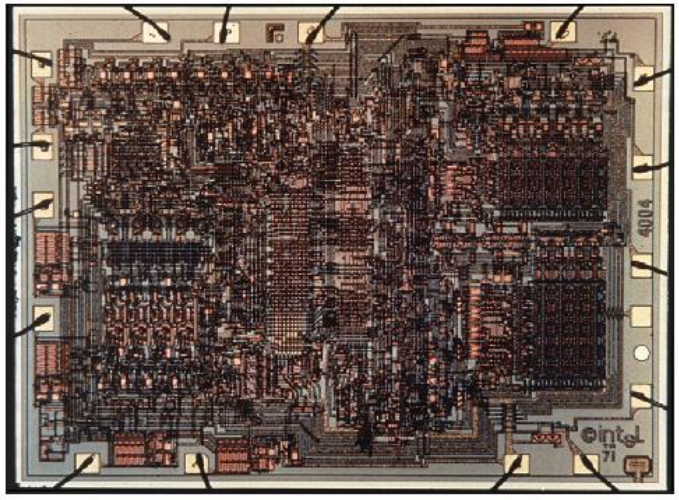
- Bipolar transistors
  - npn or pnp silicon structure
  - Small current into very thin base layer controls large currents between emitter and collector
  - Base currents limit integration density
- Metal Oxide Semiconductor Field Effect Transistors
  - nMOS and pMOS MOSFETS
  - Voltage applied to insulated gate controls current between source and drain
  - Low power allows very high integration

MOS Integrated Circuits

- 1970's processes usually had only nMOS transistors
  - Inexpensive, but consume power while idle



Intel 1101 256-bit SRAM

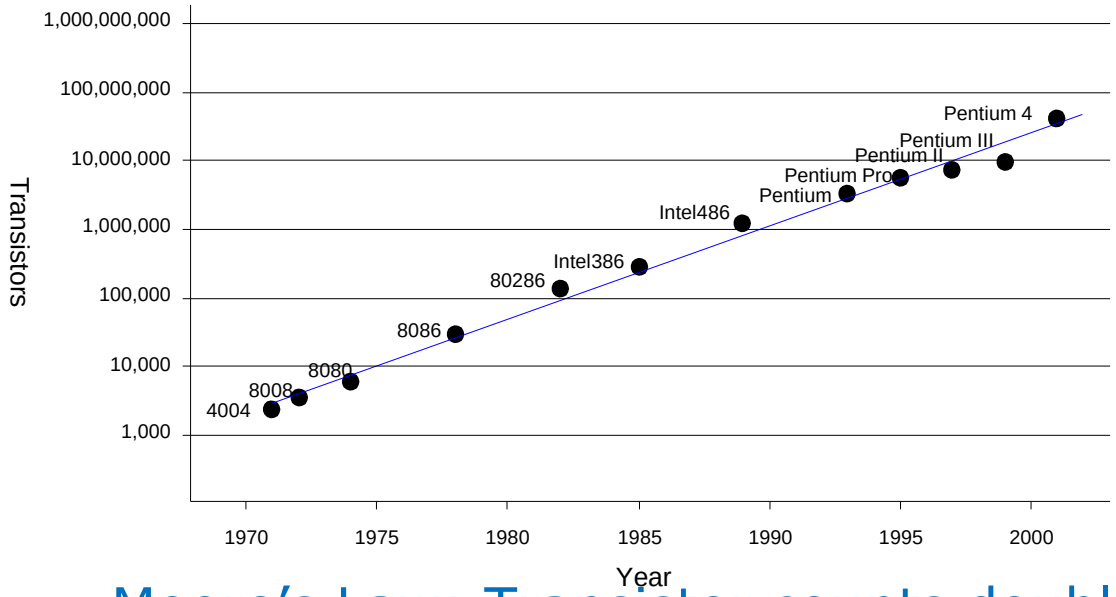


Intel 4004 4-bit μProc

- 1980s-present: CMOS processes for low idle power

Moore's Law

- 1965: Gordon Moore plotted transistor on each chip
  - Fit straight line on semilog scale: most economical!!
  - Transistor counts of Intel uPs have doubled every 26 months (following fiugure)



Integration Levels

SSI: 10 gates

MSI: 1000 gates

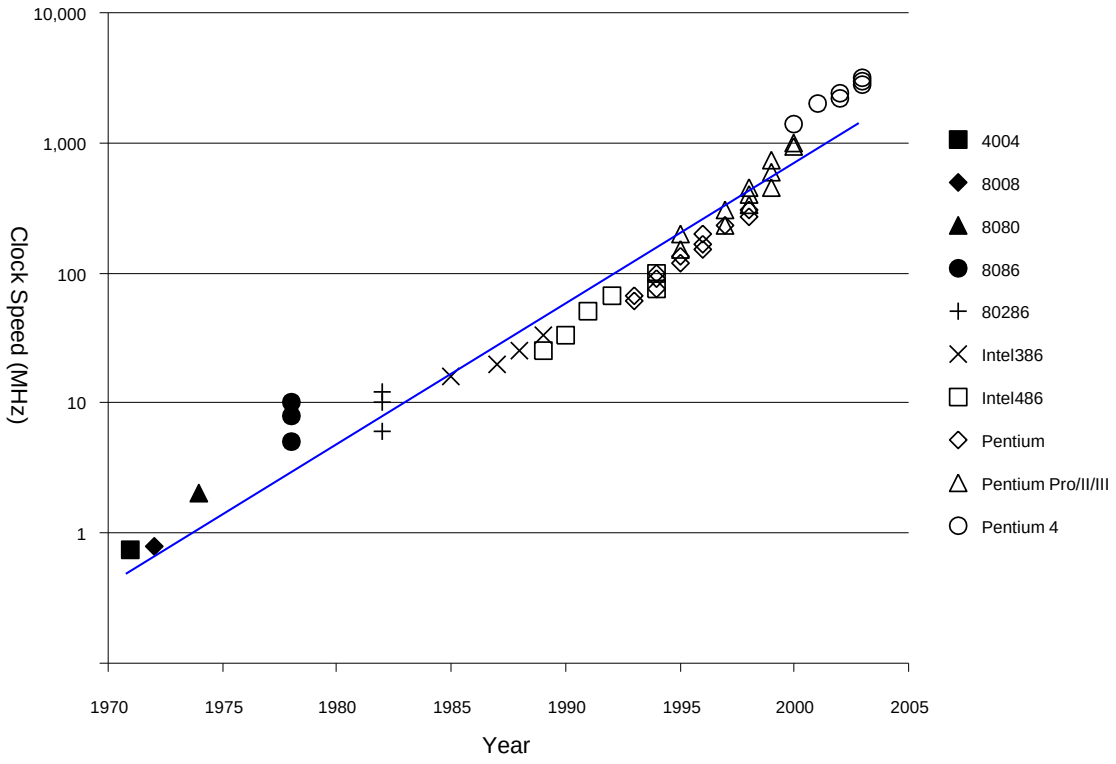
LSI: 10,000 gates

VLSI: > 10k gates

  - Moore's Law: Transistor counts doubles every 18 months.

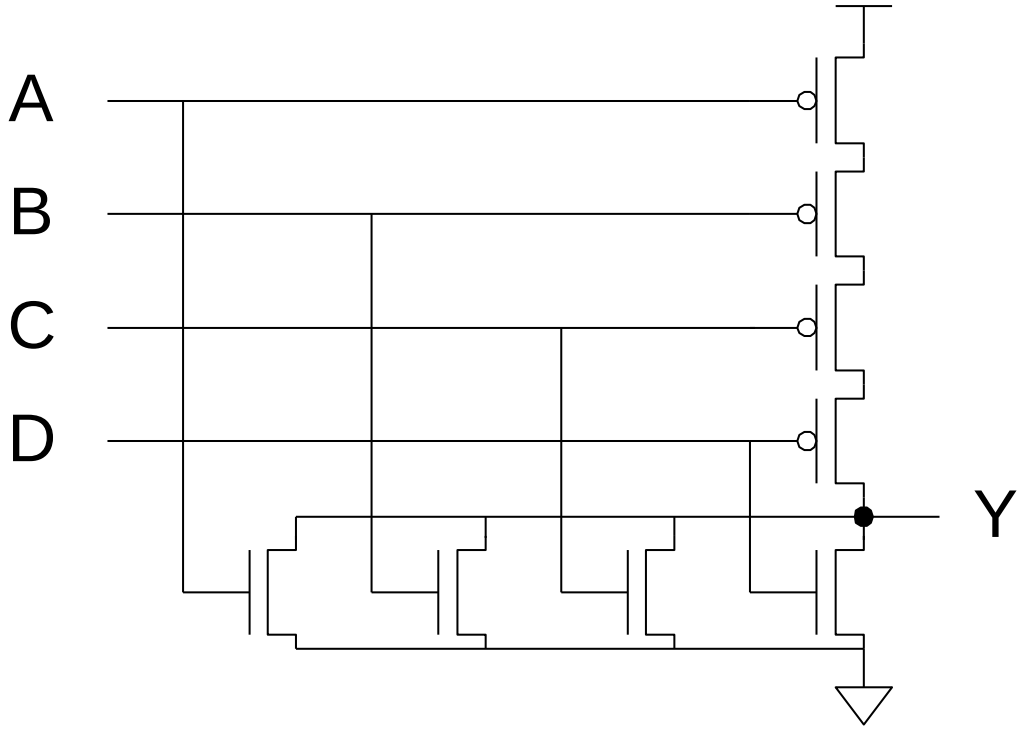
Corollaries

- Many other factors grow exponentially
  - Ex: clock frequency, processor performance



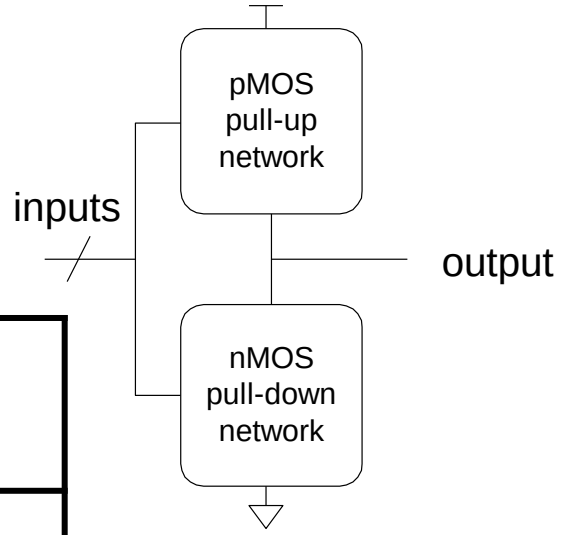
CMOS Gate Design

- Activity:
  - Sketch a 4-input CMOS NOR gate



Complementary CMOS

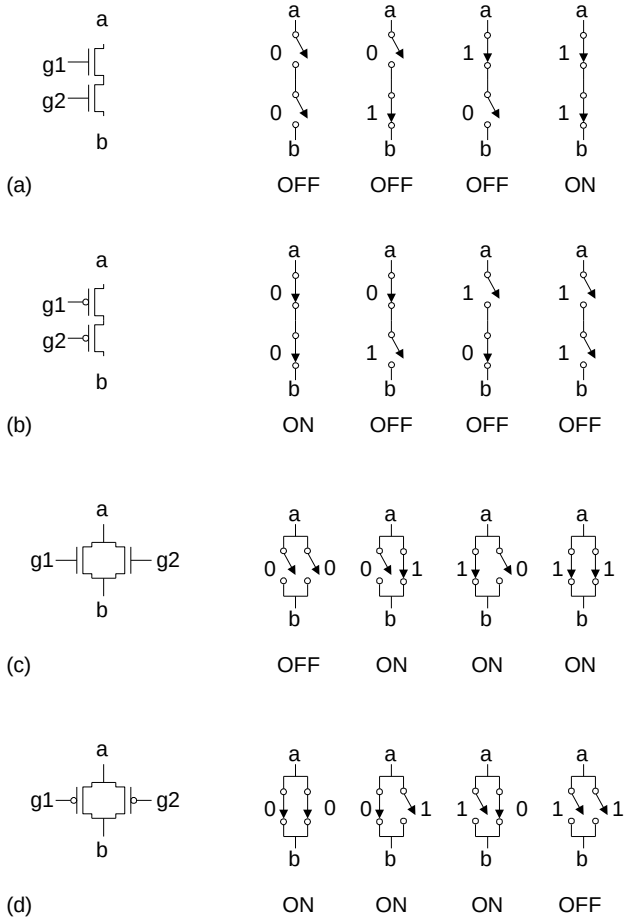
- Complementary CMOS logic gates
  - nMOS *pull-down network*
  - pMOS *pull-up network*
  - a.k.a. static CMOS



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

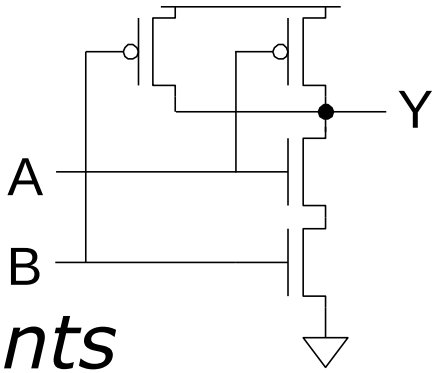
Series and Parallel

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



Conduction Complement

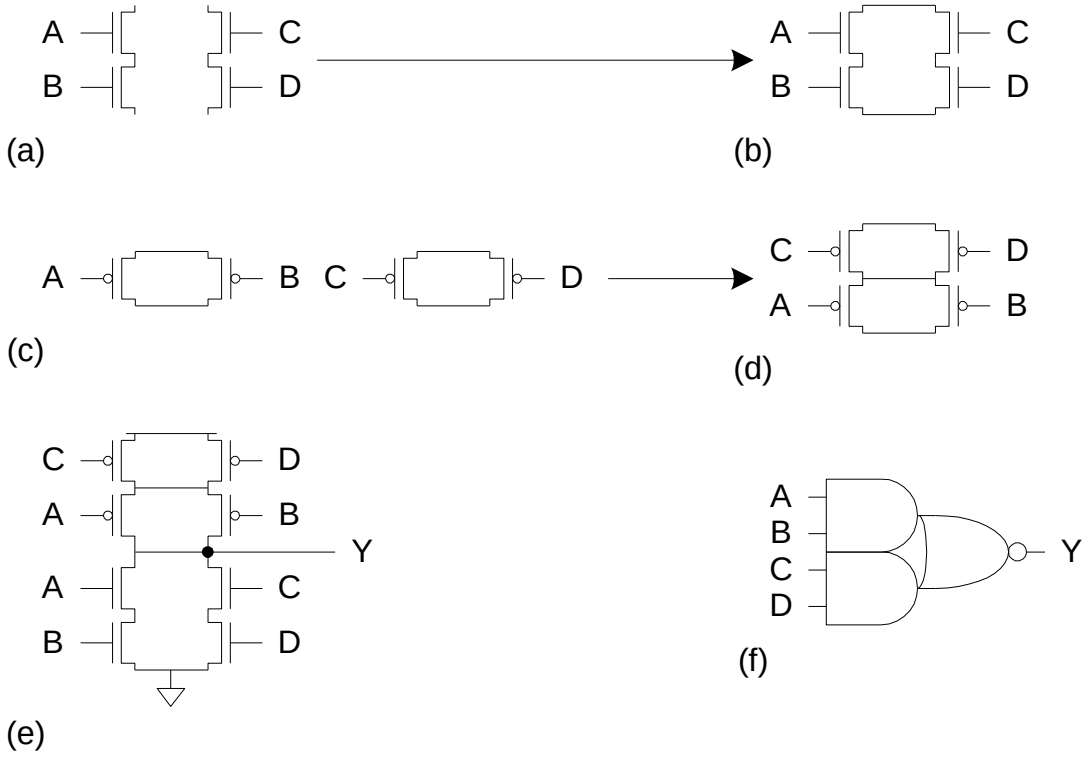
- Complementary CMOS gates always produce 0 or 1
- Ex: NAND gate
  - Series nMOS: Y=0 when both inputs are 1
  - Thus Y=1 when either input is 0
  - Requires parallel pMOS



- Rule of *Conduction Complements*
  - Pull-up network is complement of pull-down
  - Parallel -> series, series -> parallel

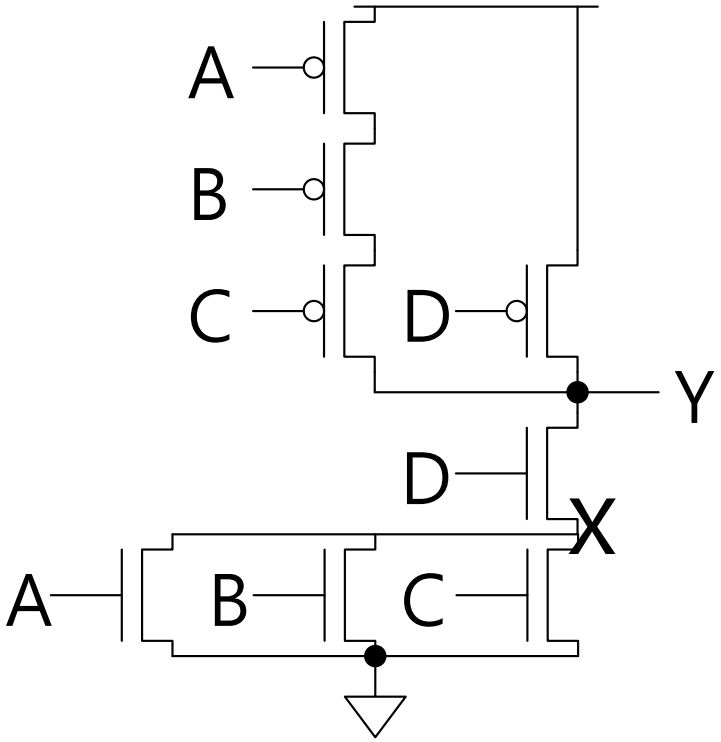
Compound Gates

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



Example: O3AI

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

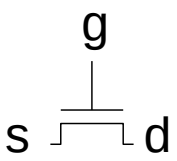


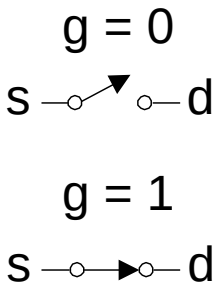
Signal Strength

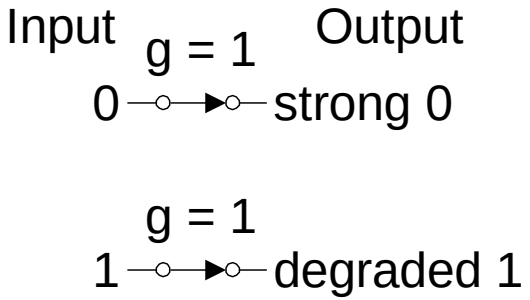
- *Strength* of signal
  - How close it approximates ideal voltage source
- V_{DD} and GND rails are strongest 1 and 0
- nMOS pass strong 0
  - But degraded or weak 1
- pMOS pass strong 1
  - But degraded or weak 0
- Thus nMOS are best for pull-down network

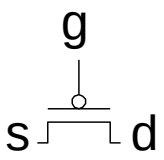
Pass Transistors

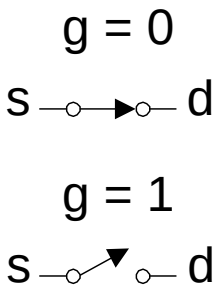
- Transistors can be used as switches

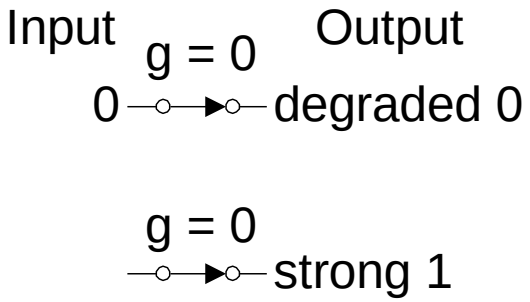






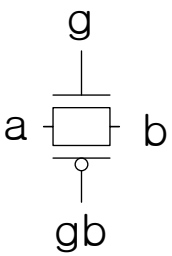


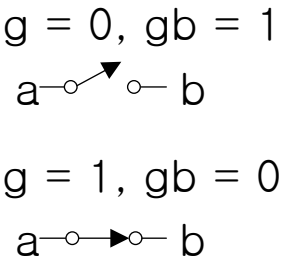


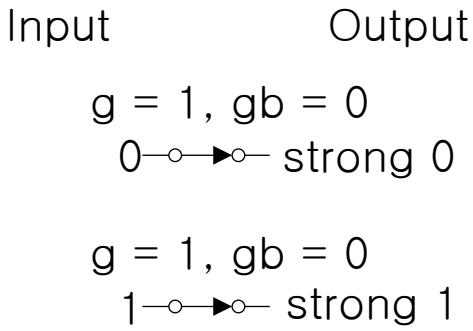


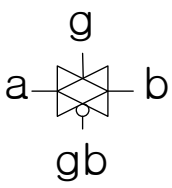
Transmission Gates

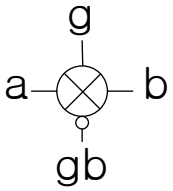
- Pass transistors produce degraded outputs
- *Transmission gates* pass both 0 and 1 well

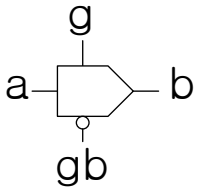








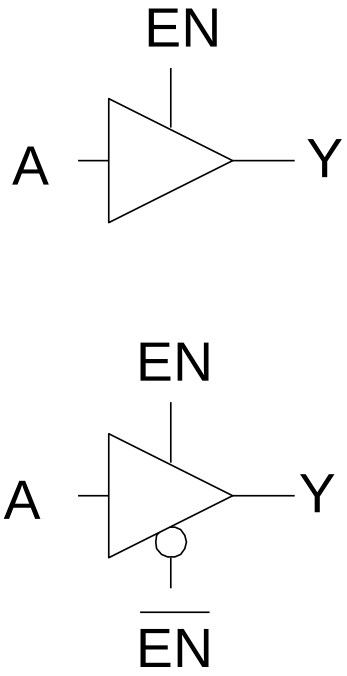




Tristates

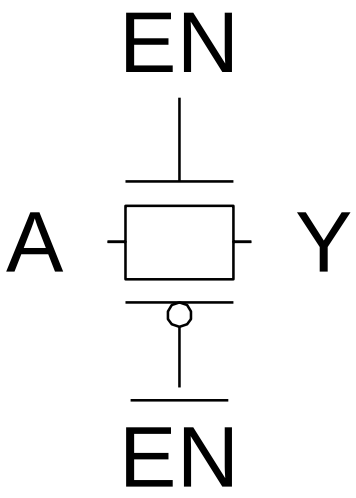
- *Tristate buffer* produces Z when not enabled

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



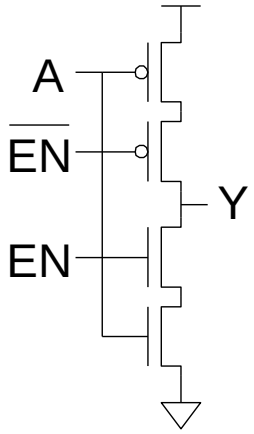
Nonrestoring Tristate

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



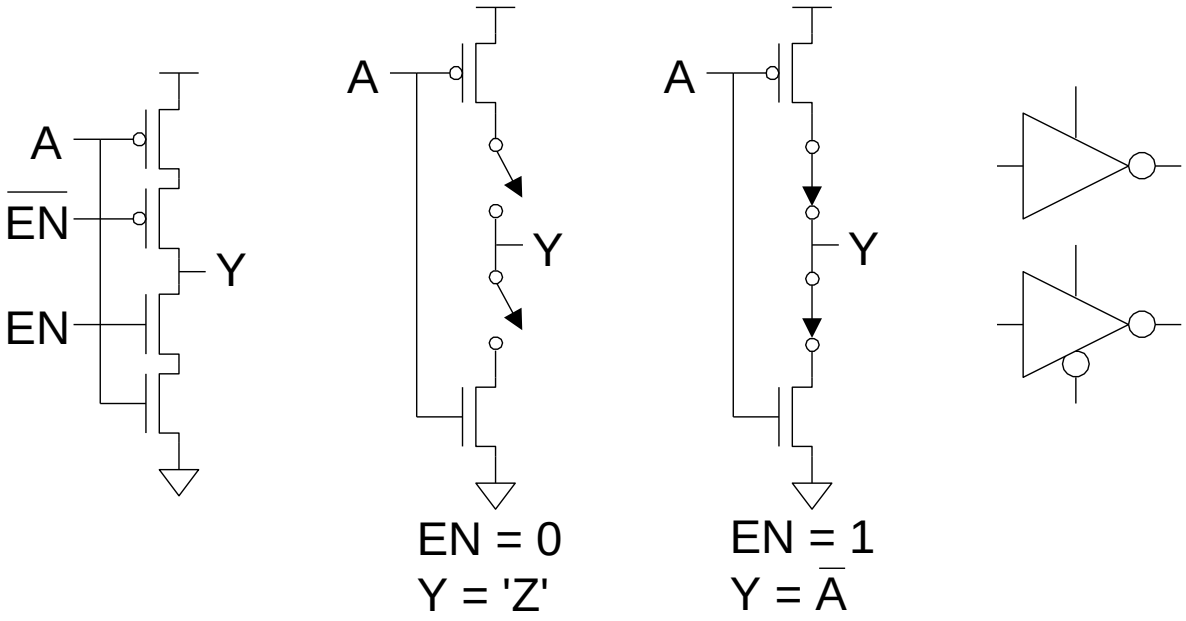
Tristate Inverter

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



Tristate Inverter

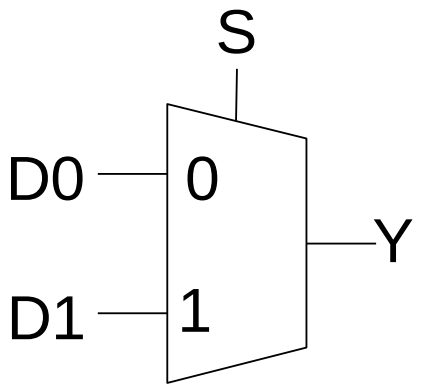
- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



Multiplexers

- 2:1 *multiplexer* chooses between two inputs

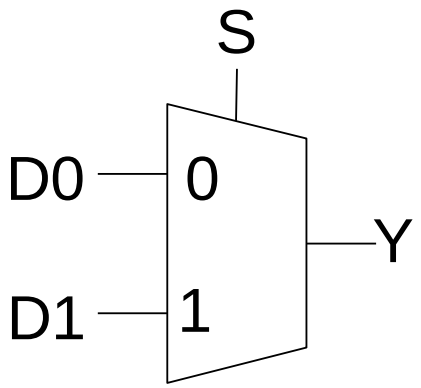
S	D1	D0	Y
0	X	0	
0	X	1	
1	0	X	
1	1	X	



Multiplexers

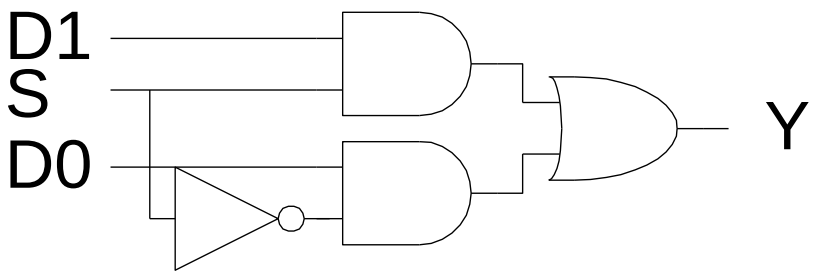
- 2:1 multiplexer chooses between two inputs

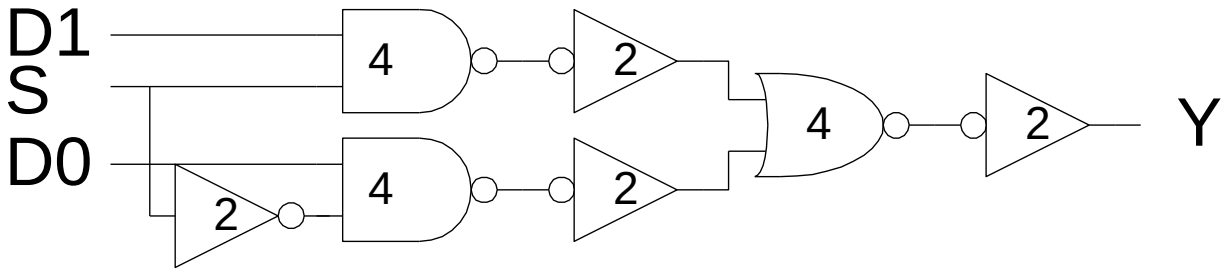
S	D1	D0	Y
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1



Gate-Level Mux Design

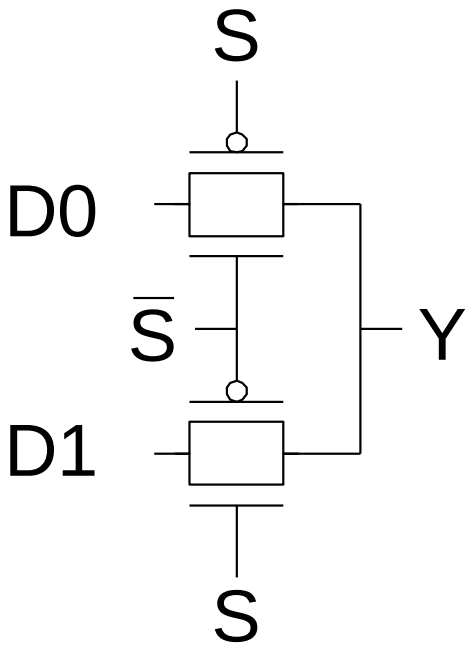
- $Y = SD_1 + \overline{S}D_0$ (too many transistors)
- How many transistors are needed? 20





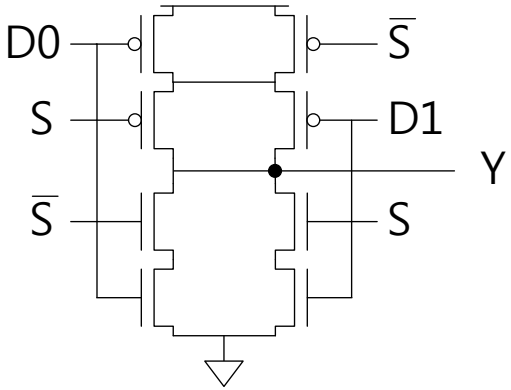
Transmission Gate Mux

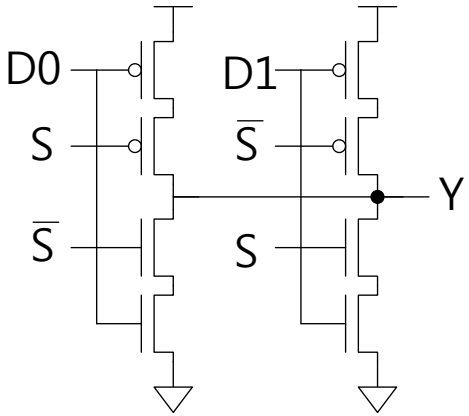
- Nonrestoring mux uses two transmission gates
  - Only 4 transistors

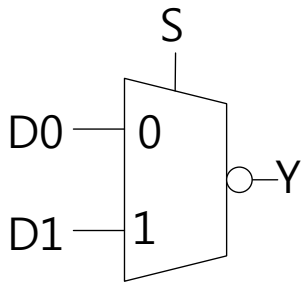


Inverting Mux

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

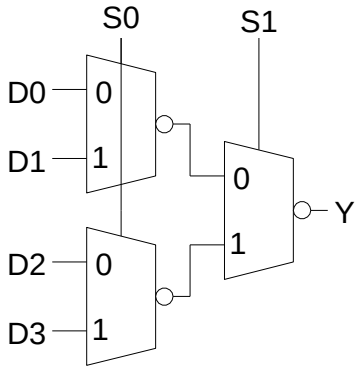


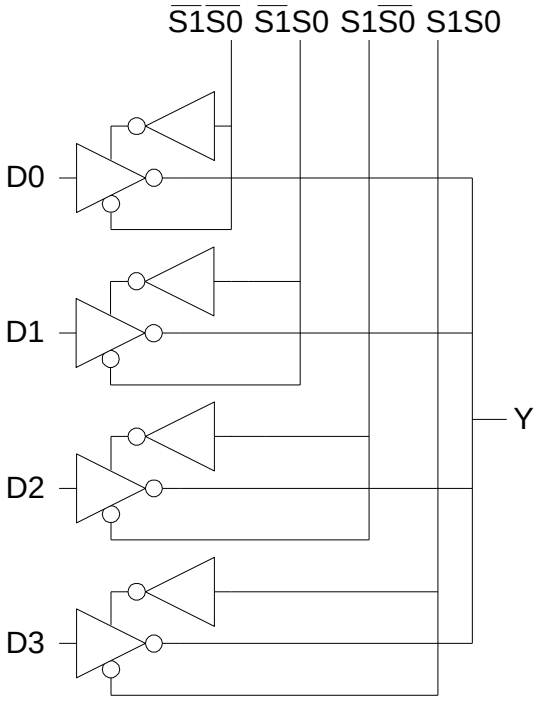




4:1 Multiplexer

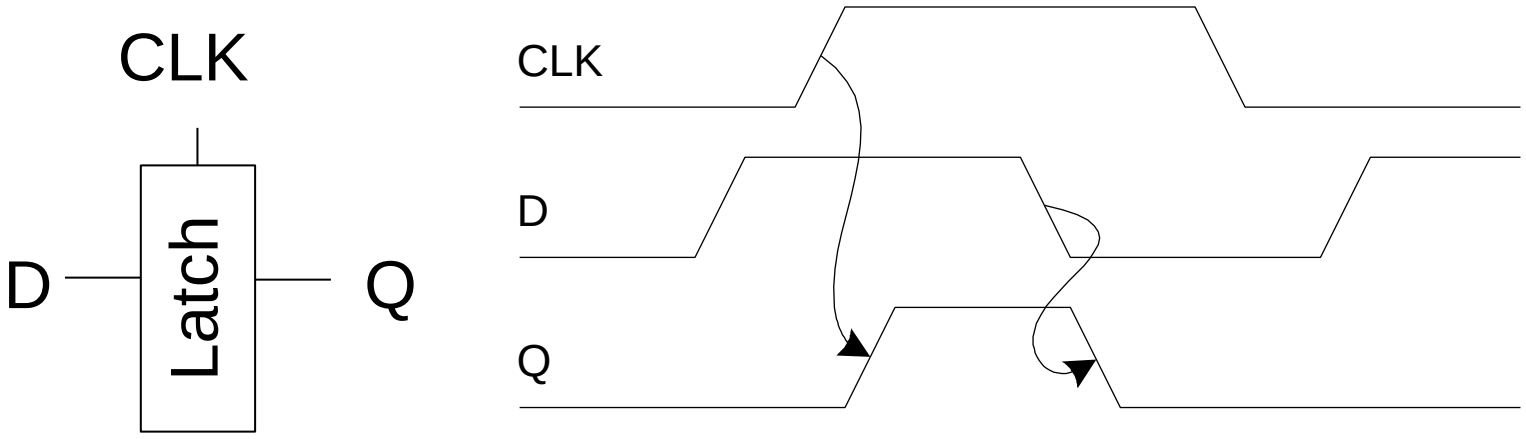
- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 2:1 muxes
  - Or four tristates





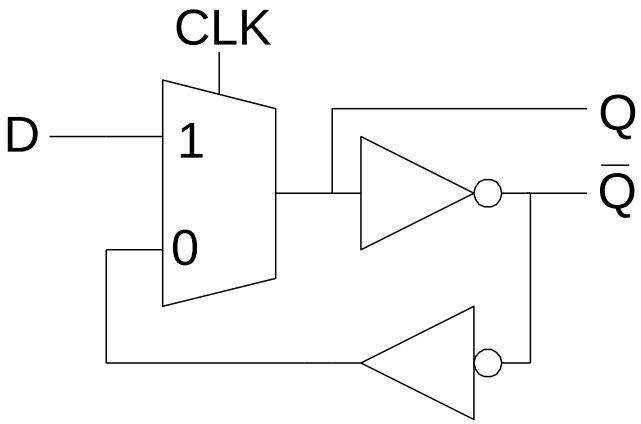
D Latch

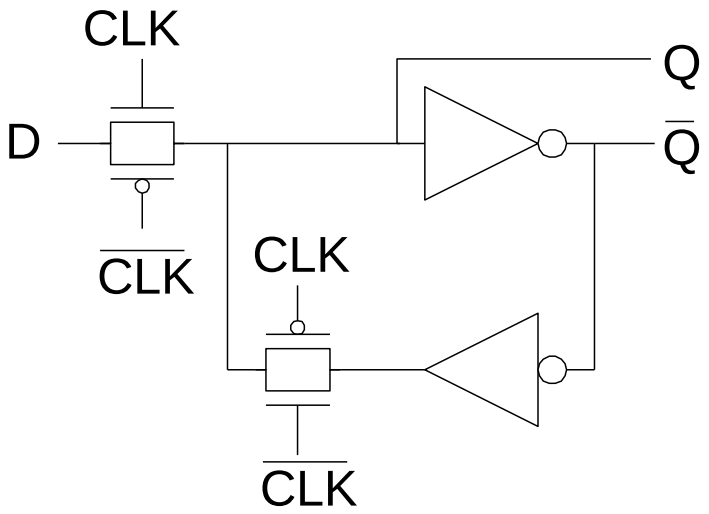
- When CLK = 1, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = 0, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*



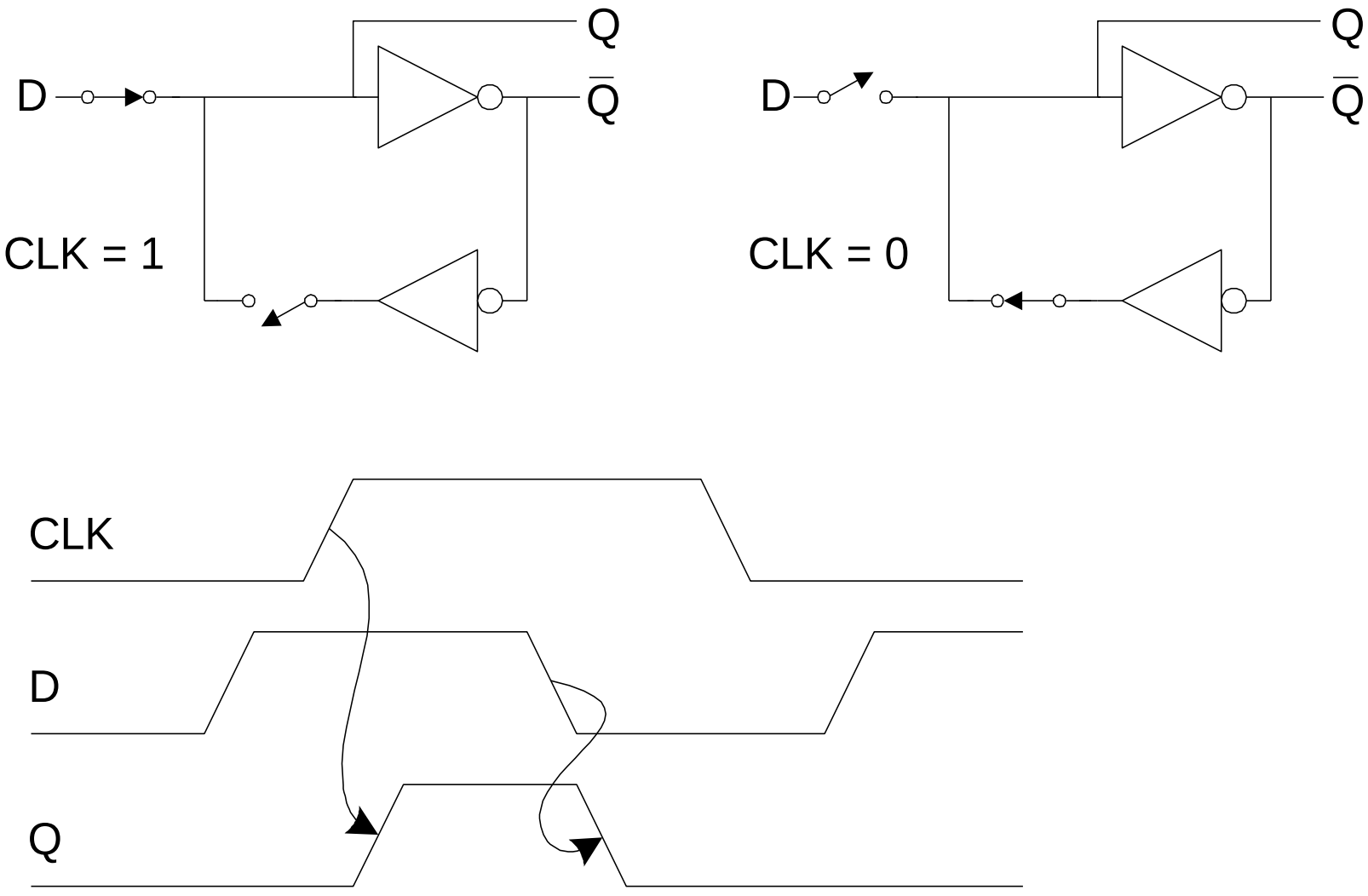
D Latch Design

- Multiplexer chooses D or old Q



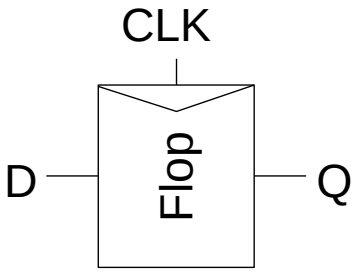


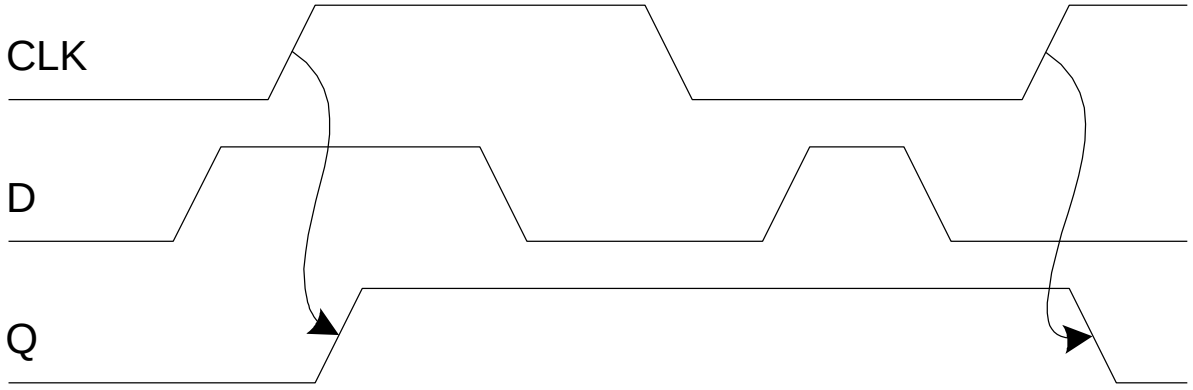
D Latch Operation



D Flip-flop

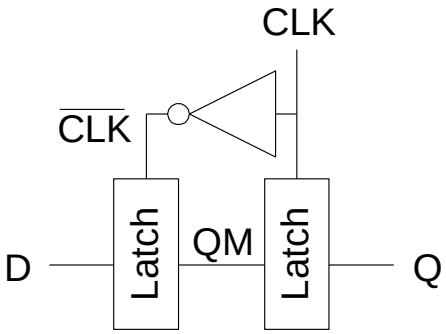
- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

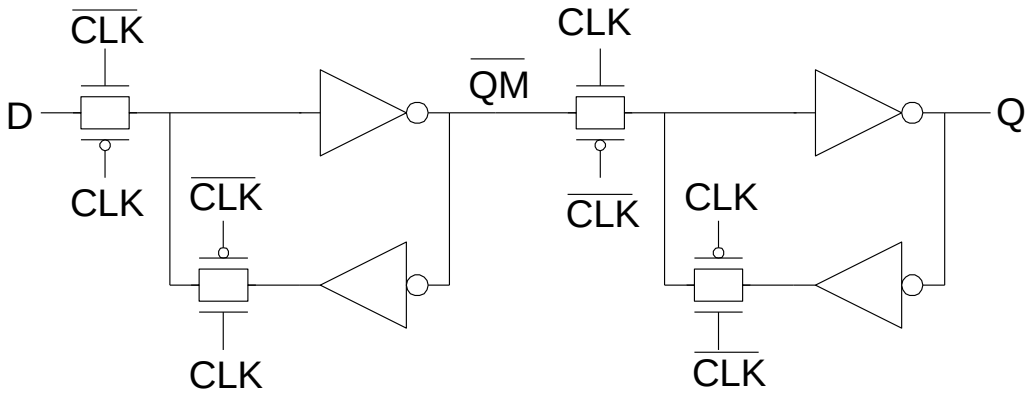




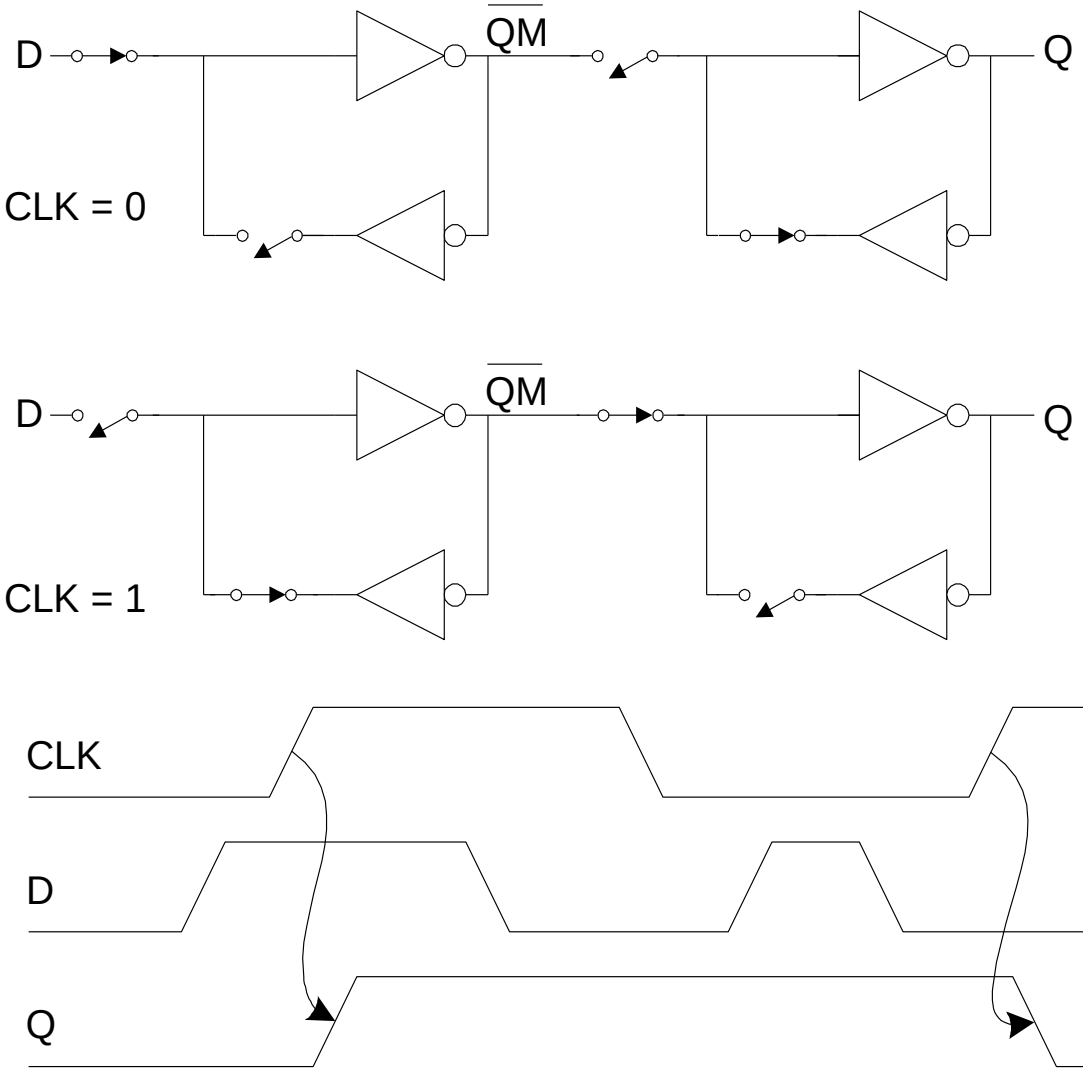
D Flip-flop Design

- Built from master and slave D latches



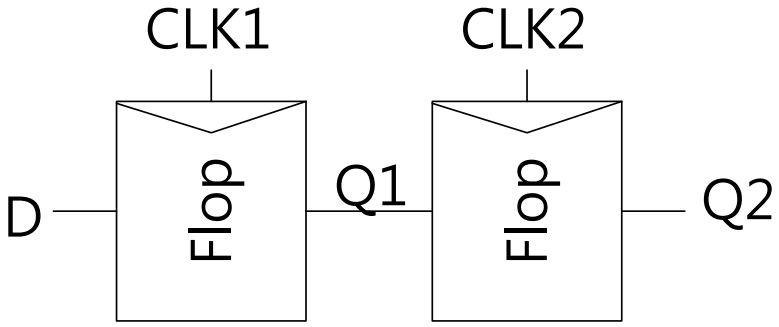


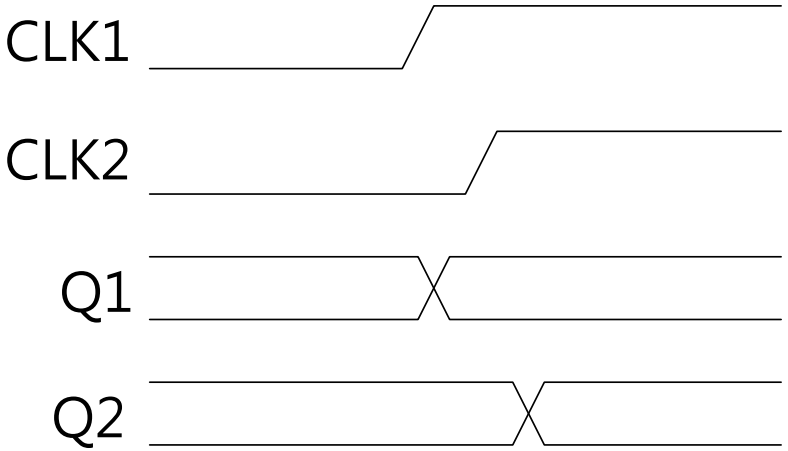
D Flip-flop Operation



Race Condition

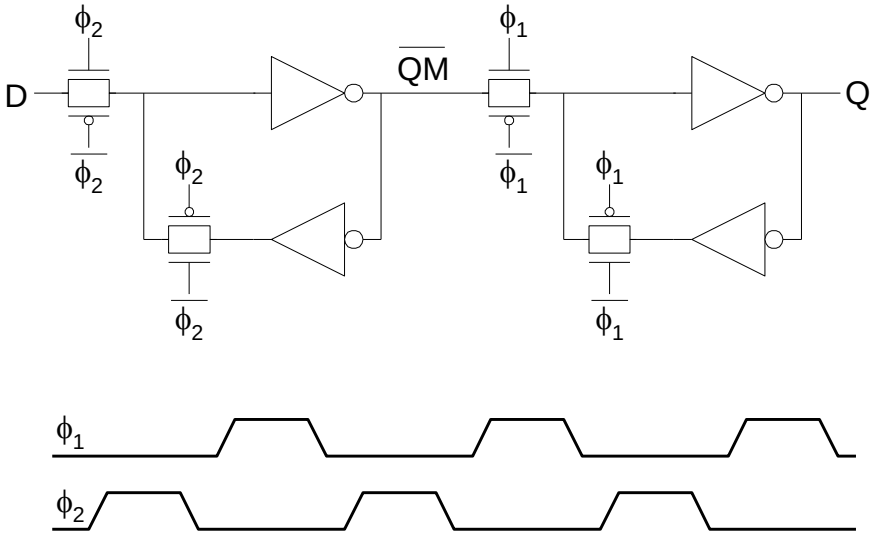
- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*





Nonoverlapping Clocks

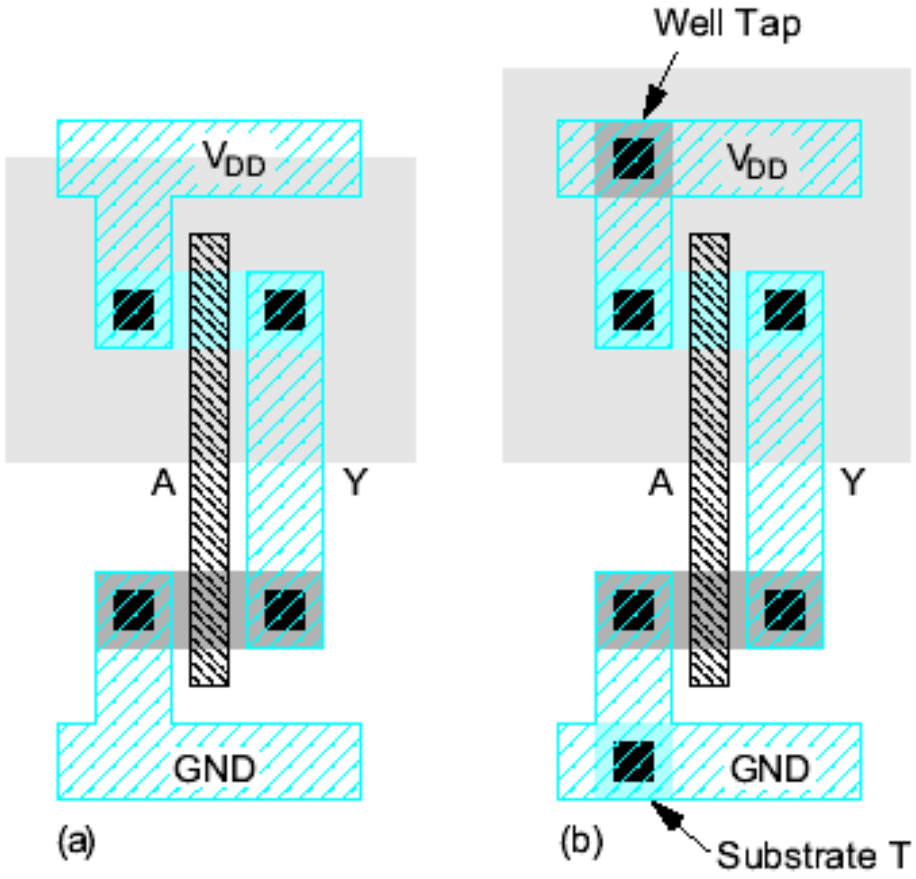
- Nonoverlapping clocks can prevent races
  - As long as nonoverlap exceeds clock skew
- We will use them in this class for safe design
  - Industry manages skew more carefully instead



Gate Layout

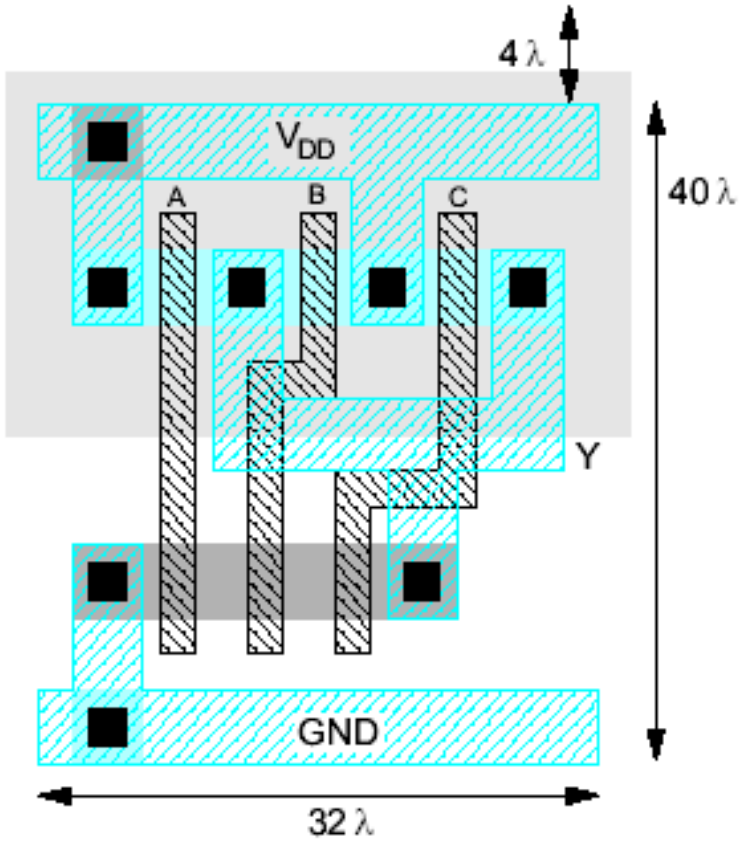
- Layout can be very time consuming
  - Design gates to fit together nicely
  - Build a library of standard cells
- Standard cell design methodology
  - V_{DD} and GND should abut (standard height)
  - Adjacent gates should satisfy design rules
  - nMOS at bottom and pMOS at top
  - All gates include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

Example: Inverter



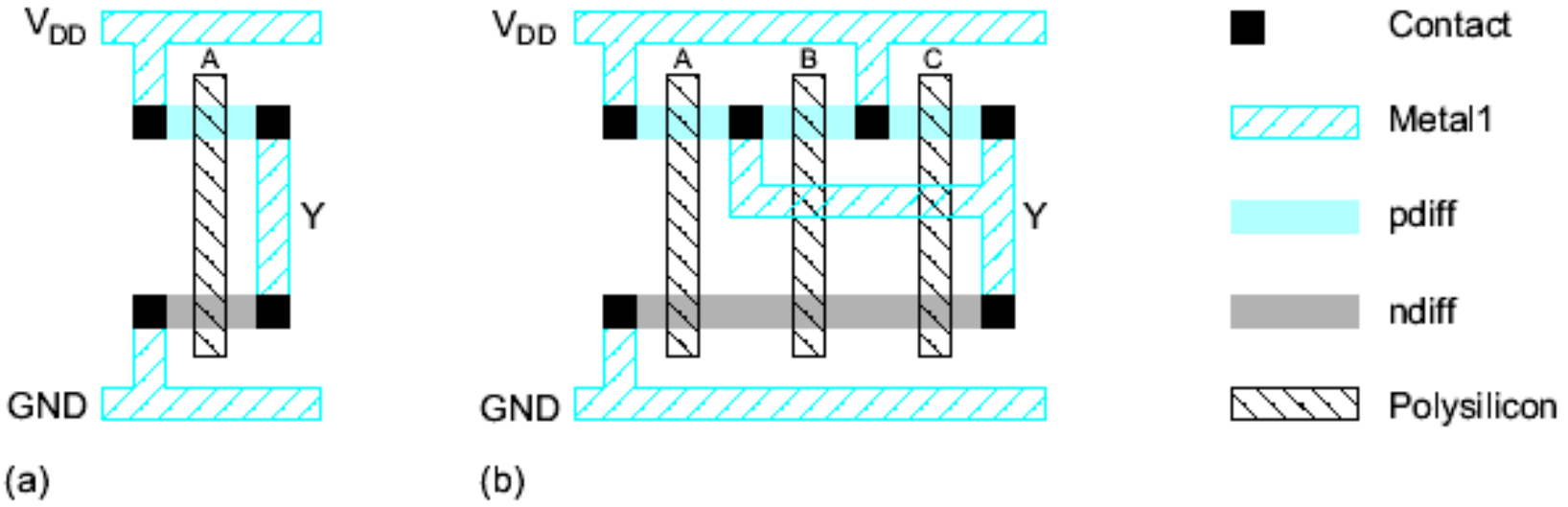
Example: NAND3

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



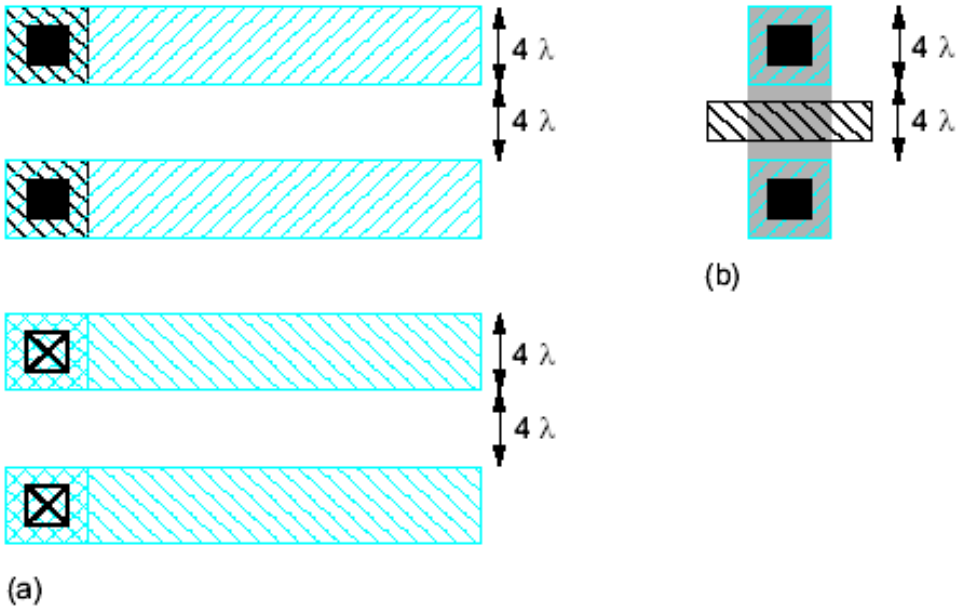
Stick Diagrams

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



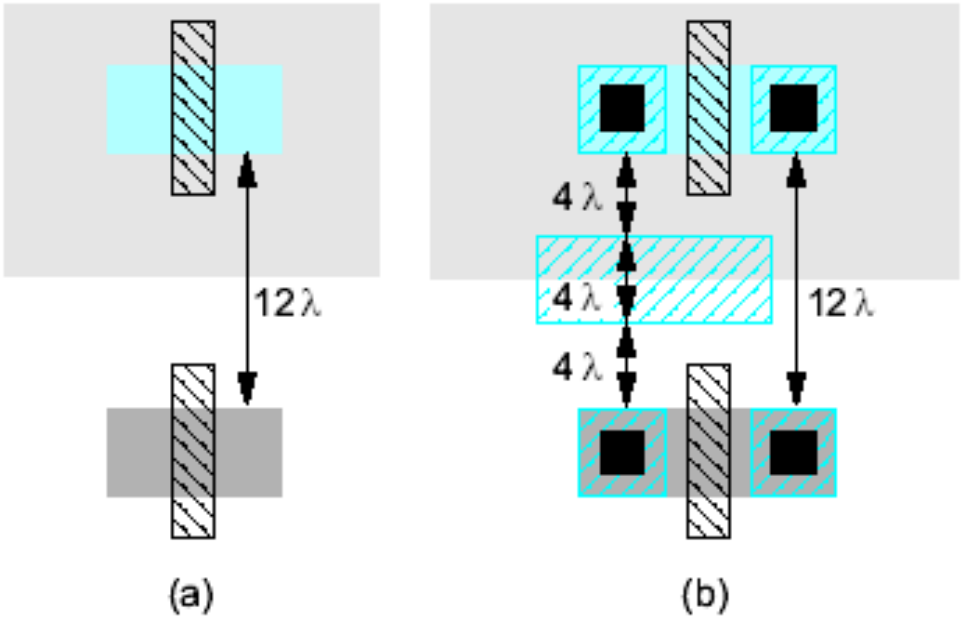
Wiring Tracks

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



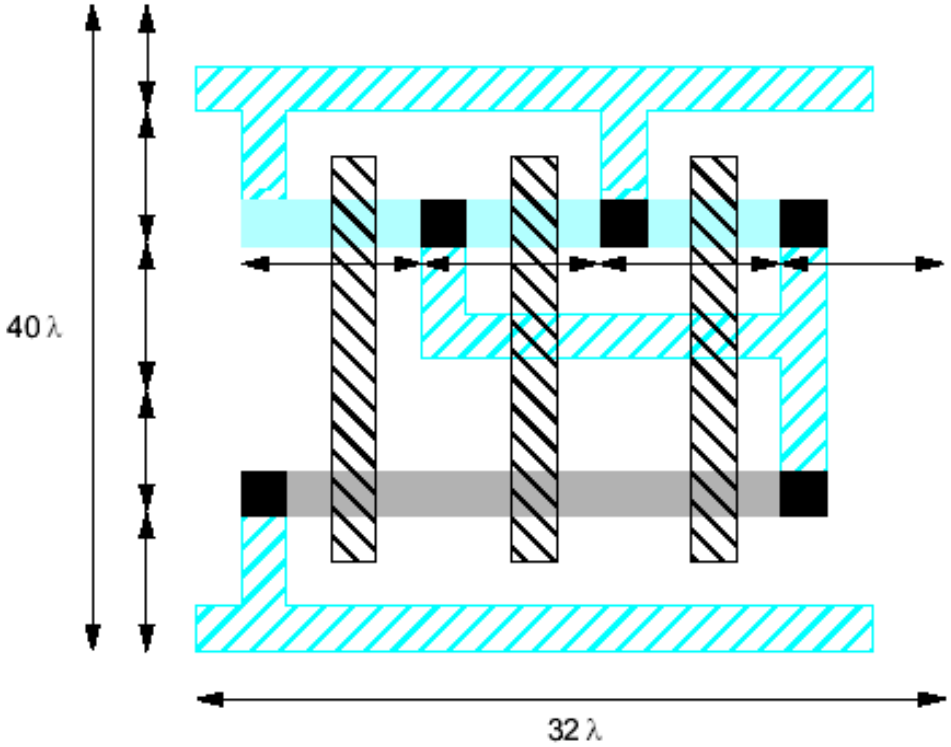
Well spacing

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



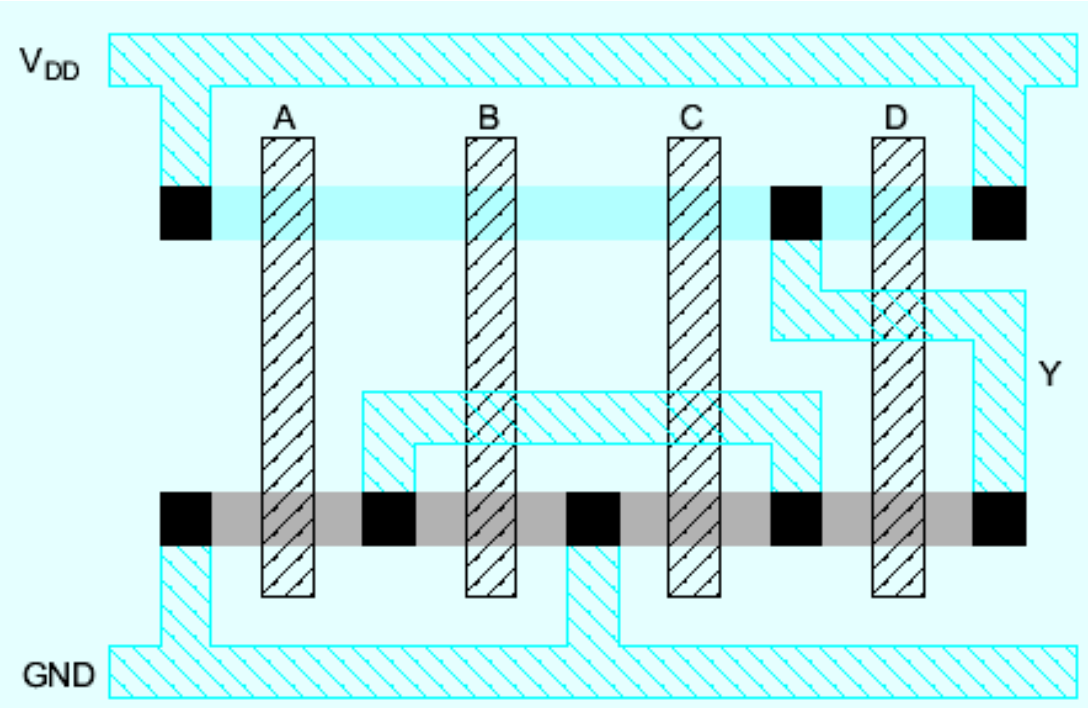
Area Estimation

- Estimate area by counting wiring tracks
  - Multiply by 8 to express in λ



Example: O3AI

- Sketch a stick diagram for O3AI and estimate area
  - $Y = \overline{(A + B + C) \cdot D}$



Transistor Sizing

- Decide the W and L of a CMOS circuit to provide equal current driving capability in both directions (PUN and PDN) as the basic inverter.
  - If $(W/L)_n = n,\ (W/L)_p = p,\ p = (\mu n/\mu p)\cdot n$ *for a matched design.*

- PDN's capacitor discharge current should be at least equal that of NMOS TR of inverter.
- PUD's capacitor charging current should be at least equal that of PMOS TR of inverter.

⇒ the above two conditions will guarantee a worst-cast gate delay equal to that of the basic inverter.

⇒ we need to find the equivalent W/L ratio of a network of MOS transistor as that of an inverter.

Transistor Sizing

- Equivalent series resistance when a number of MOSFETs are connected in series

$$R_{series} = r_{DS1} + r_{DS2} + \cdots = \frac{constant}{(W/L)_1} + \frac{constant}{(W/L)_2} + \cdots$$

$$= constant\left[\frac{1}{(W/L)_1} + \frac{1}{(W/L)_2} + \cdots\right]$$

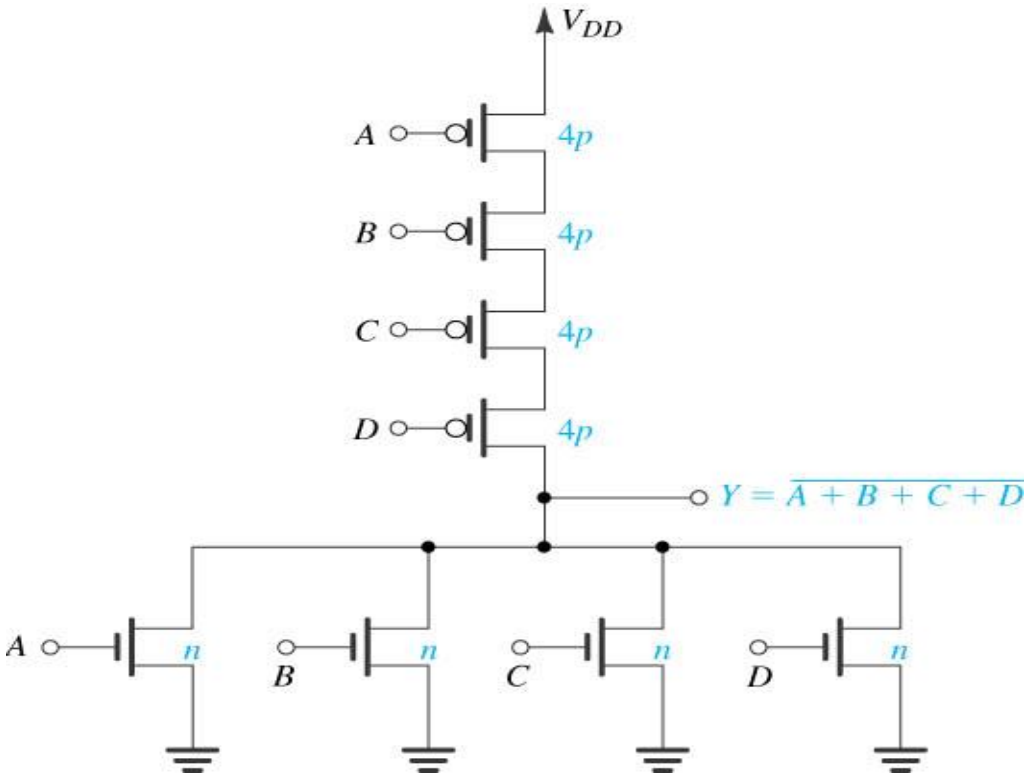
$$= \frac{constant}{(W/L)_{eq}}$$

$$(W/L)_{eq} = \frac{1}{\frac{1}{(W/L)_1} + \frac{1}{(W/L)_2} + \cdots}$$

- In the same manner, for parallel connection of TRs

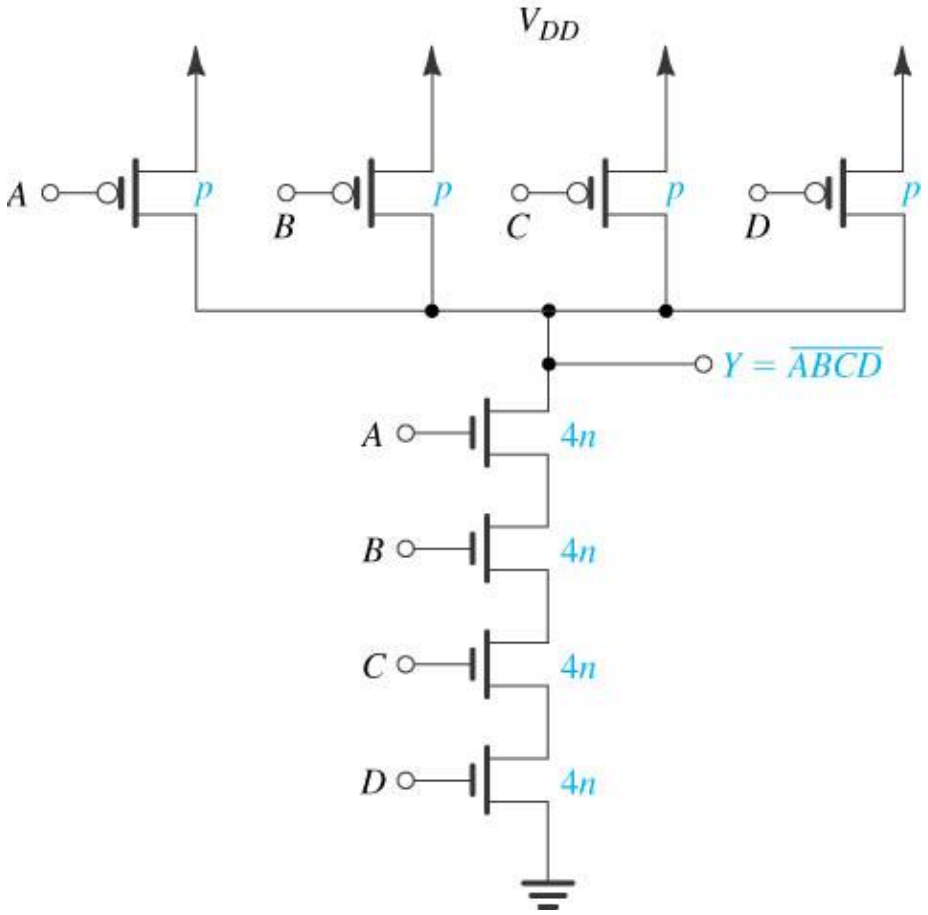
$$(W/L)_{eq} = (W/L)_1 + (W/L)_2 + \cdots$$

Transistor Sizing



Proper transistor sizing for a four-input NOR gate. Note that *n* and *p* denote the (*W*/*L*) ratios of Q_N and Q_P, respectively, of the basic inverter.

Transistor Sizing



Proper transistor sizing for a four-input NAND gate. Note that *n* and *p* denote the (*W*/*L*) ratios of Q_N and Q_P, respectively, of the basic inverter.

Transistor Sizing

$$Q_{NB} : W / L = 2n = 3 = 0.75 / 0.25$$

$$Q_{NC} : W / L = 2n = 3 = 0.75 / 0.25$$

$$Q_{ND} : W / L = 2n = 3 = 0.75 / 0.25$$

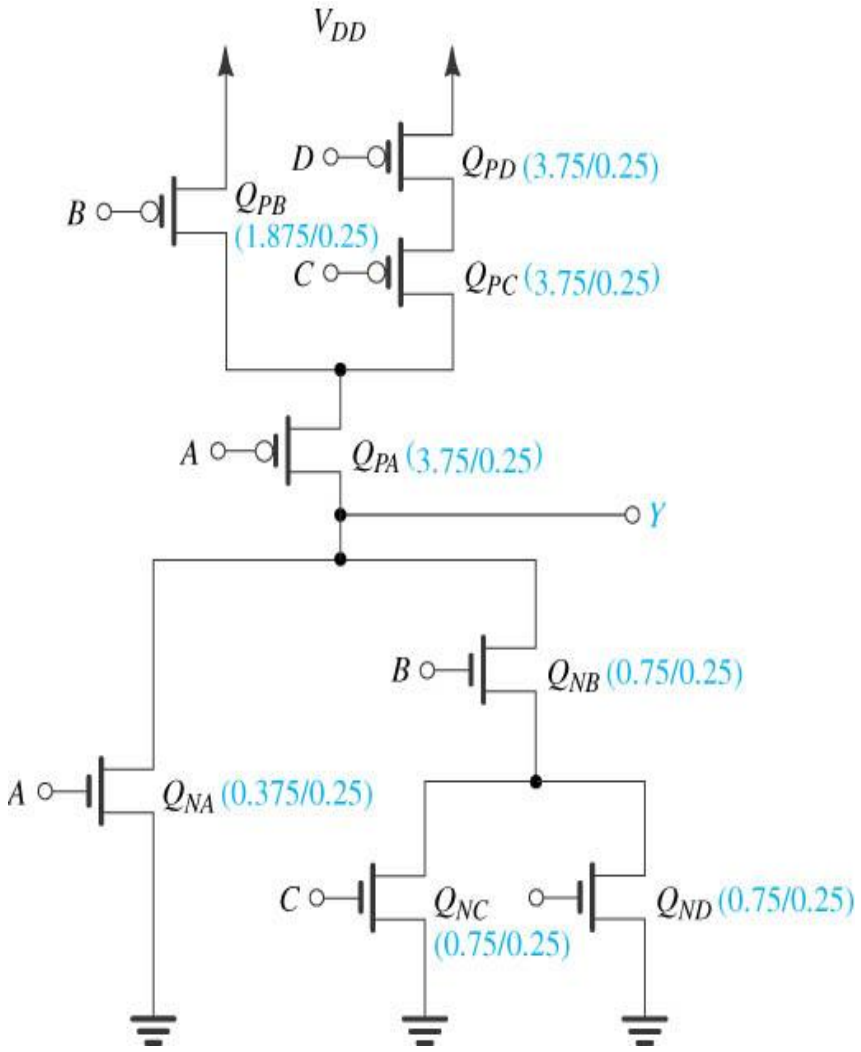
$$Q_{NA} : W / L = n = 1.5 = 0.375 / 0.25$$

$$Q_{PA} : W / L = 3p = 15 = 3.75 / 0.25$$

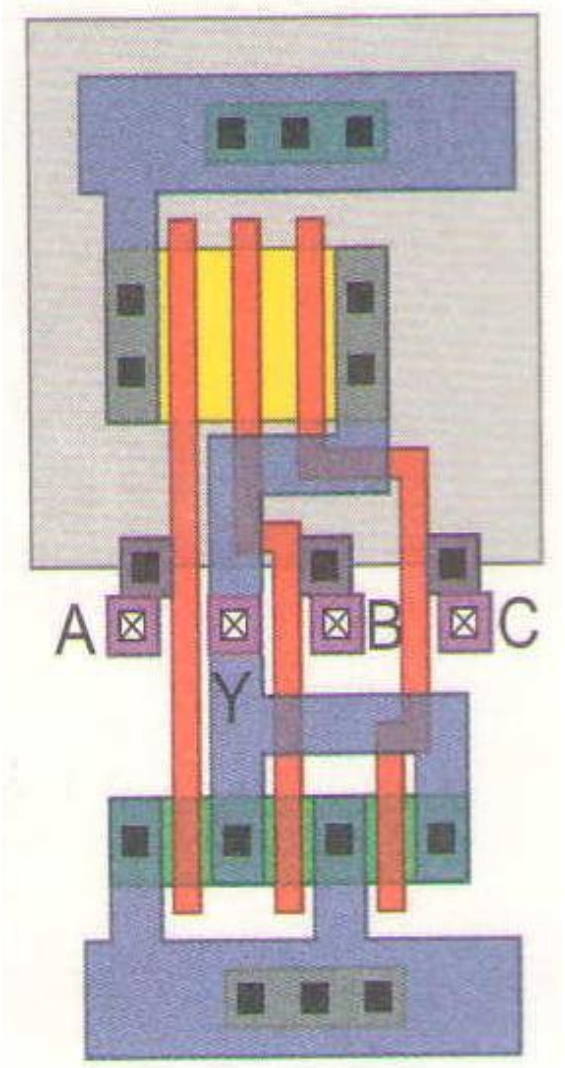
$$Q_{PC} : W / L = 3p = 15 = 3.75 / 0.25$$

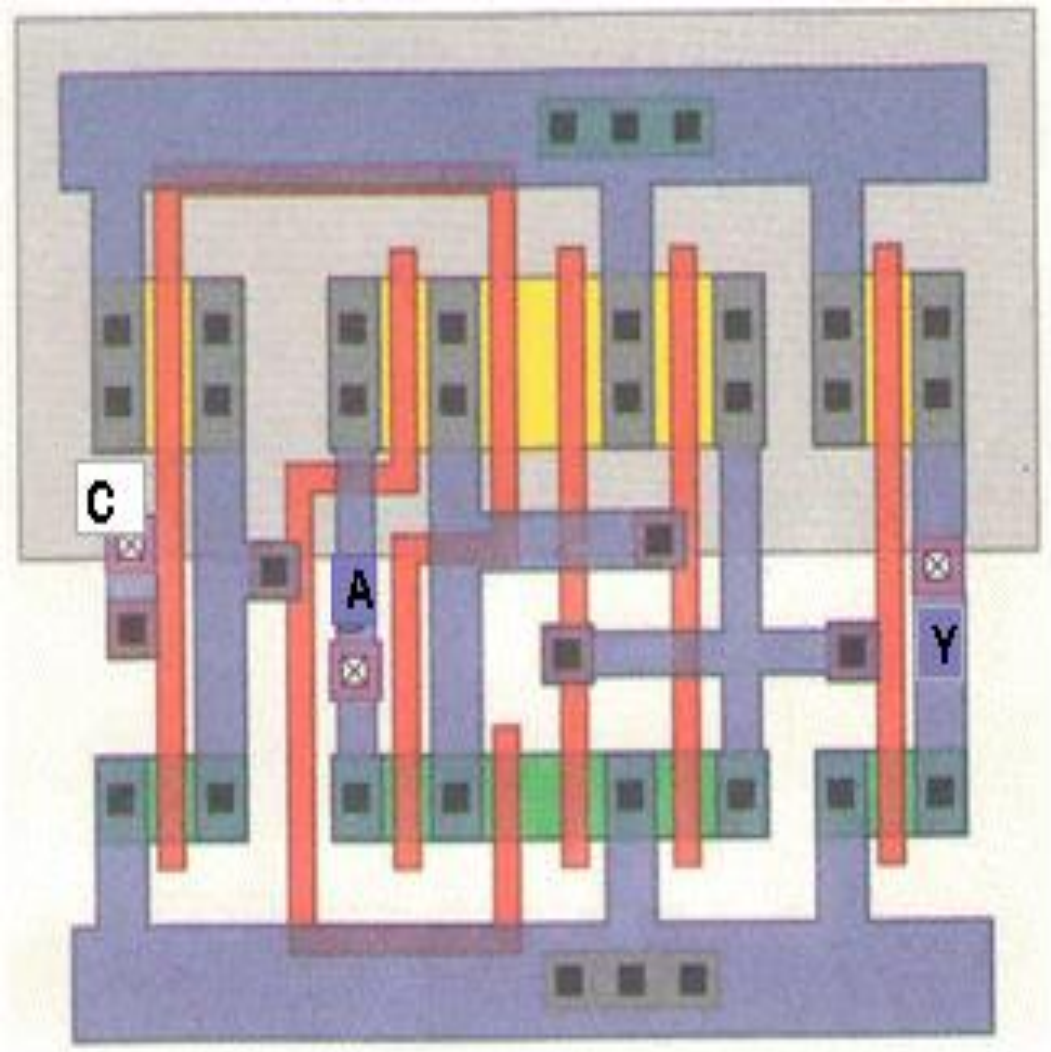
$$Q_{PD} : W / L = 3p = 15 = 3.75 / 0.25$$

$$Q_{PB} : W / L = 1.5p = 7.5 = 1.875 / 0.25$$



Transistor Sizing





DC & Transient Response

Outline

- DC Response
- Logic Levels and Noise Margins
- Transient Response
- Delay Estimation

Activity

1) If the width of a transistor increases, the current will

increase decrease not change

2) If the length of a transistor increases, the current will

increase decrease not change

3) If the supply voltage of a chip increases, the maximum transistor current will

increase decrease not change

4) If the width of a transistor increases, its gate capacitance will

increase decrease not change

5) If the length of a transistor increases, its gate capacitance will

increase decrease not change

6) If the supply voltage of a chip increases, the gate capacitance of each transistor will

increase decrease not change

Activity

1) If the width of a transistor increases, the current will

increase decrease not change

2) If the length of a transistor increases, the current will

increase **decrease** not change

3) If the supply voltage of a chip increases, the maximum transistor current will

increase decrease not change

4) If the width of a transistor increases, its gate capacitance will

increase decrease not change

5) If the length of a transistor increases, its gate capacitance will

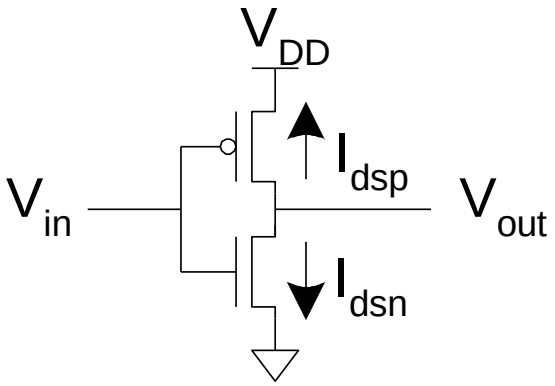
increase decrease not change

6) If the supply voltage of a chip increases, the gate capacitance of each transistor will

increase decrease **not change**

DC Response

- DC Response: V_{out} vs. V_{in} for a gate
- Ex: Inverter
  - When $V_{in} = 0$ -> $V_{out} = V_{DD}$
  - When $V_{in} = V_{DD}$ -> $V_{out} = 0$
  - In between, V_{out} depends on transistor size and current
  - By KCL, must settle such that $I_{dsn} = |I_{dsp}|$
  - We could solve equations
  - But graphical solution gives more insight

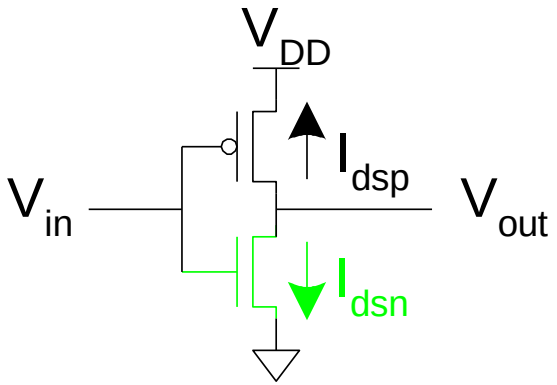


Transistor Operation

- Current depends on region of transistor behavior
- For what V_{in} and V_{out} are nMOS and pMOS in
  - Cutoff?
  - Linear?
  - Saturation?

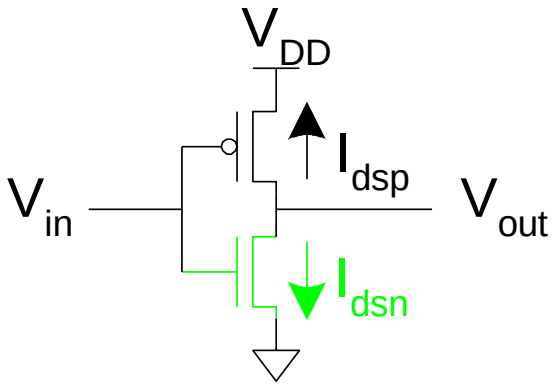
nMOS Operation

Cutoff	Linear	Saturated
V_{gsn} <	V_{gsn} >	V_{gsn} >
	V_{dsn} <	V_{dsn} >



nMOS Operation

Cutoff	Linear	Saturated
$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$ $V_{dsn} < V_{gsn} - V_{tn}$	$V_{gsn} > V_{tn}$ $V_{dsn} > V_{gsn} - V_{tn}$

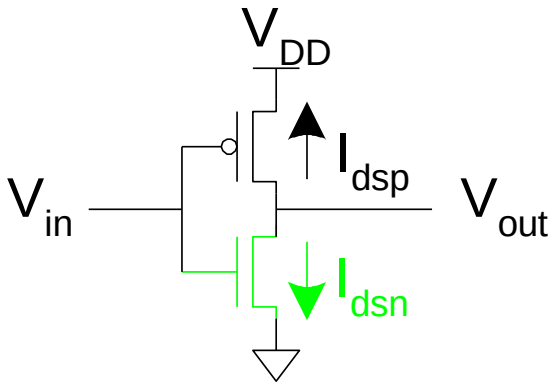


nMOS Operation

Cutoff	Linear	Saturated
$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$ $V_{dsn} < V_{gsn} - V_{tn}$	$V_{gsn} > V_{tn}$ $V_{dsn} > V_{gsn} - V_{tn}$

$V_{gsn} = V_{in}$

$V_{dsn} = V_{out}$

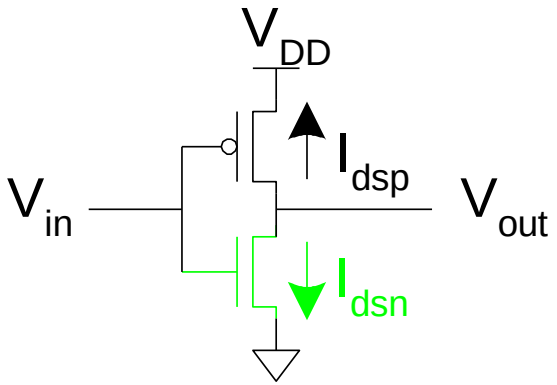


nMOS Operation

Cutoff	Linear	Saturated
$V_{gsn} < V_{tn}$ $V_{in} < V_{tn}$	$V_{gsn} > V_{tn}$ $V_{in} > V_{tn}$ $V_{dsn} < V_{gsn} - V_{tn}$ $V_{out} < V_{in} - V_{tn}$	$V_{gsn} > V_{tn}$ $V_{in} > V_{tn}$ $V_{dsn} > V_{gsn} - V_{tn}$ $V_{out} > V_{in} - V_{tn}$

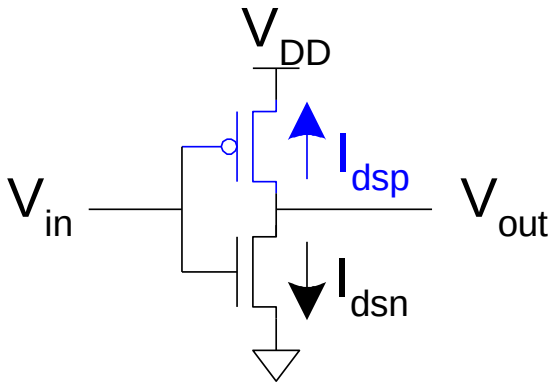
$V_{gsn} = V_{in}$

$V_{dsn} = V_{out}$



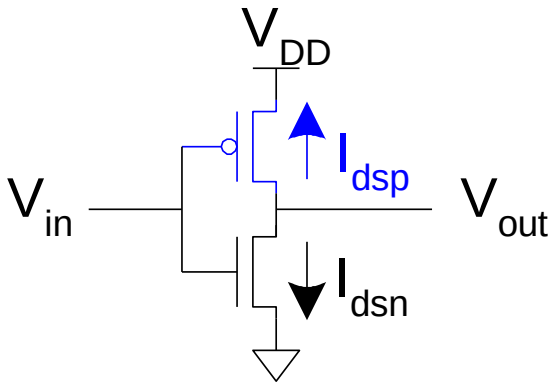
pMOS Operation

Cutoff	Linear	Saturated
$V_{gsp} >$	$V_{gsp} <$ $V_{dsp} >$	$V_{gsp} <$ $V_{dsp} <$



pMOS Operation

Cutoff	Linear	Saturated
$V_{gsp} > V_{tp}$	$V_{gsp} < V_{tp}$ $V_{dsp} > V_{gsp} - V_{tp}$	$V_{gsp} < V_{tp}$ $V_{dsp} < V_{gsp} - V_{tp}$



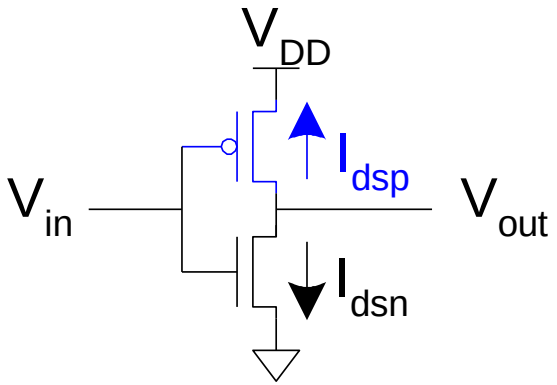
pMOS Operation

Cutoff	Linear	Saturated
$V_{gsp} > V_{tp}$	$V_{gsp} < V_{tp}$ $V_{dsp} > V_{gsp} - V_{tp}$	$V_{gsp} < V_{tp}$ $V_{dsp} < V_{gsp} - V_{tp}$

$V_{gsp} = V_{in} - V_{DD}$

$V_{dsp} = V_{out} - V_{DD}$

$V_{tp} < 0$



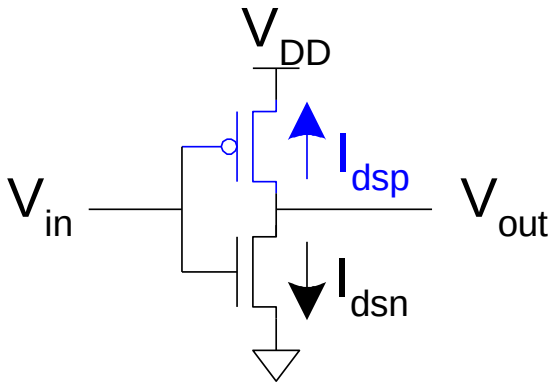
pMOS Operation

Cutoff	Linear	Saturated
$V_{gsp} > V_{tp}$ $V_{in} > V_{DD} + V_{tp}$	$V_{gsp} < V_{tp}$ $V_{in} < V_{DD} + V_{tp}$ $V_{dsp} > V_{gsp} - V_{tp}$ $V_{out} > V_{in} - V_{tp}$	$V_{gsp} < V_{tp}$ $V_{in} < V_{DD} + V_{tp}$ $V_{dsp} < V_{gsp} - V_{tp}$ $V_{out} < V_{in} - V_{tp}$

$V_{gsp} = V_{in} - V_{DD}$

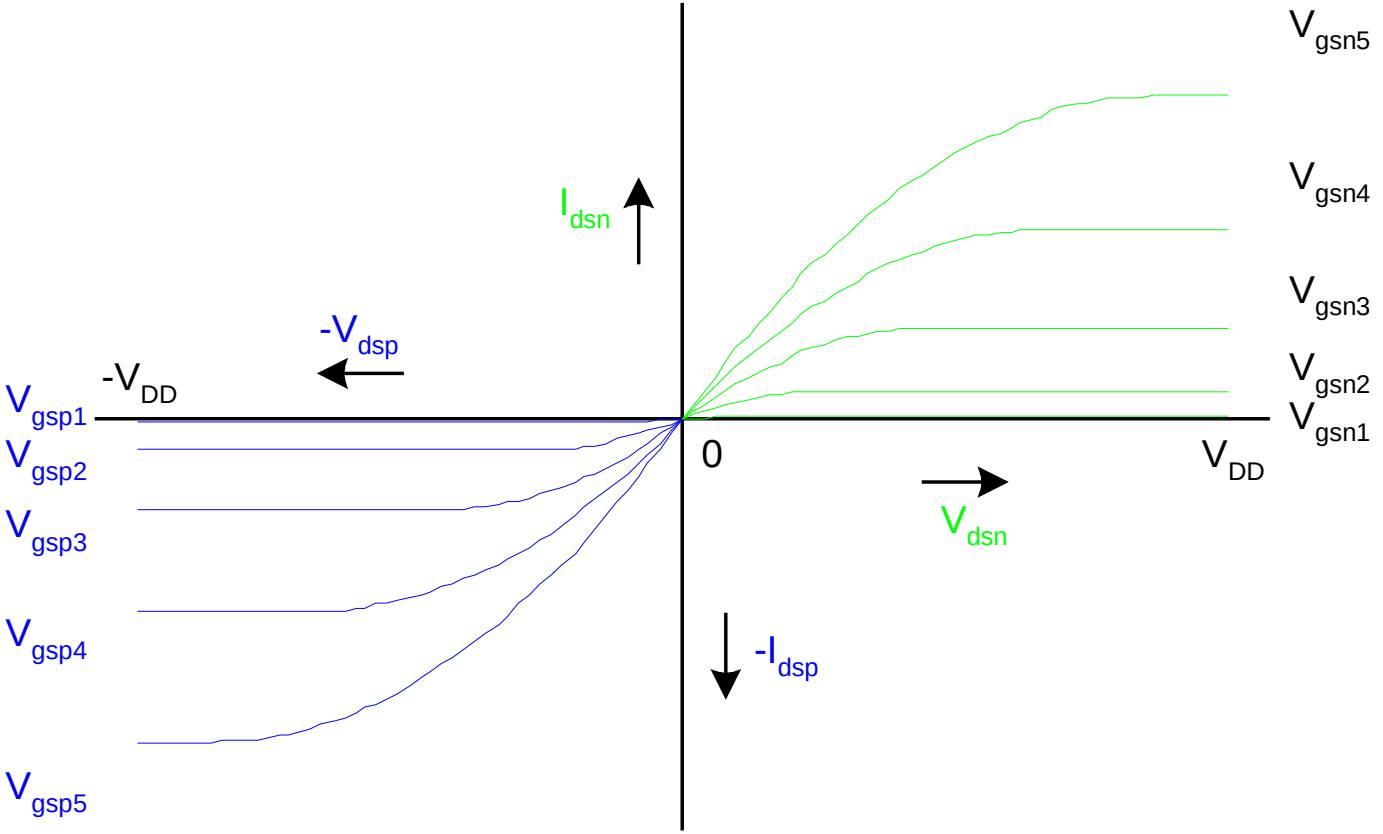
$V_{dsp} = V_{out} - V_{DD}$

$V_{tp} < 0$

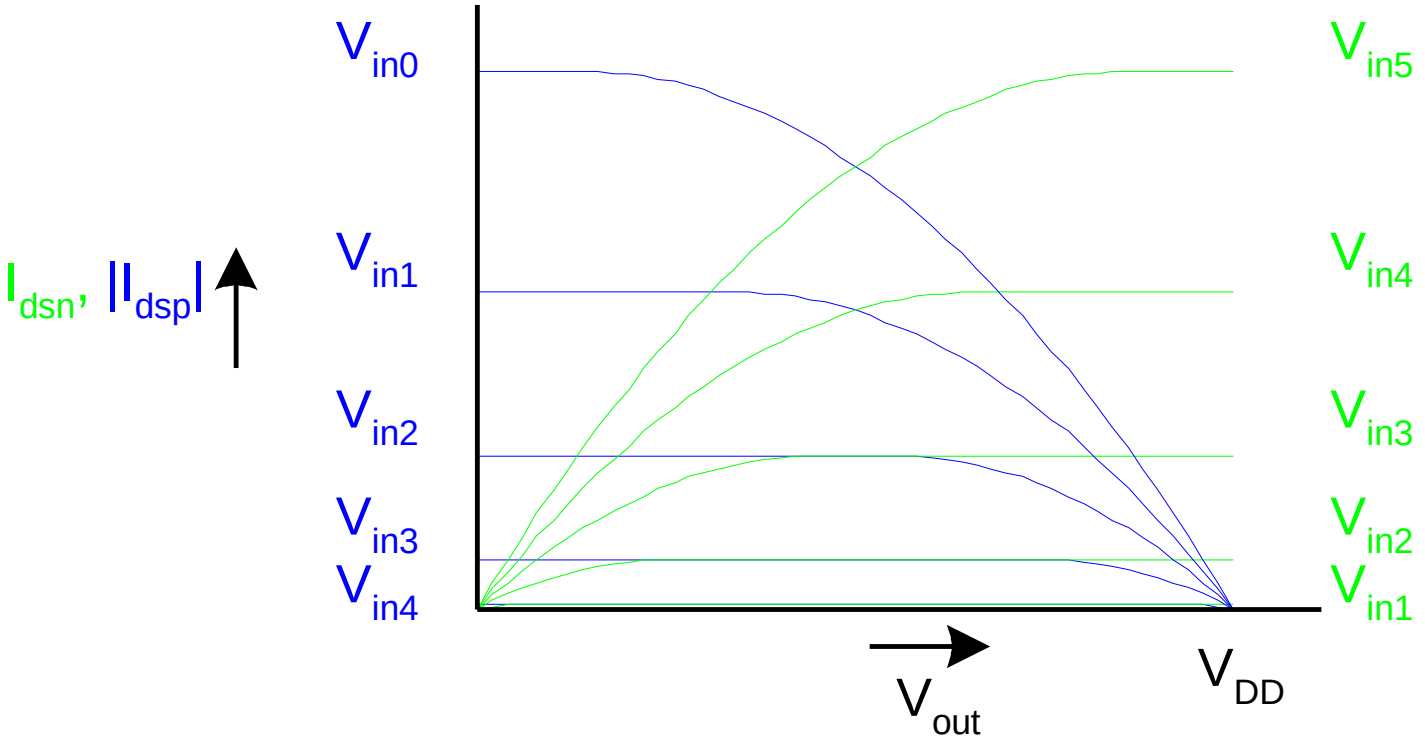


I-V Characteristics

- Make pMOS is wider than nMOS such that $\beta_n = \beta_p$

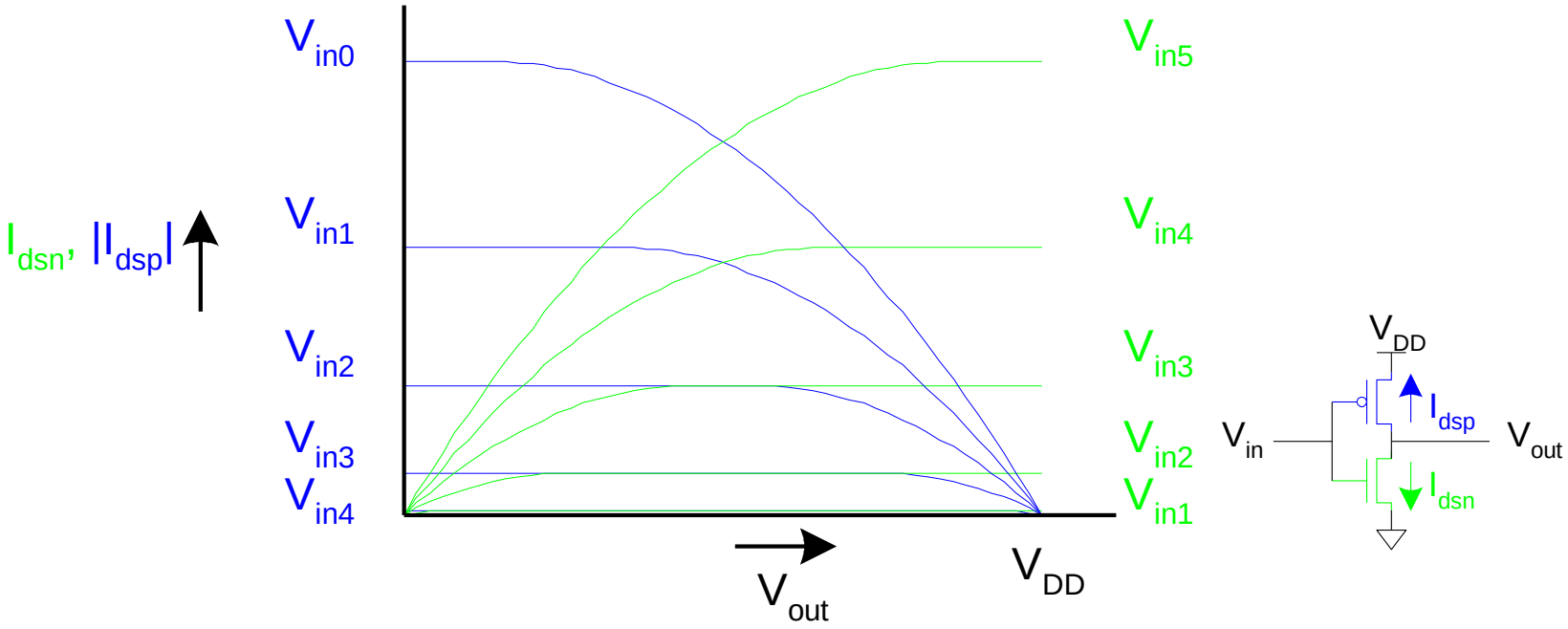


Current vs. V_{out}, V_{in}



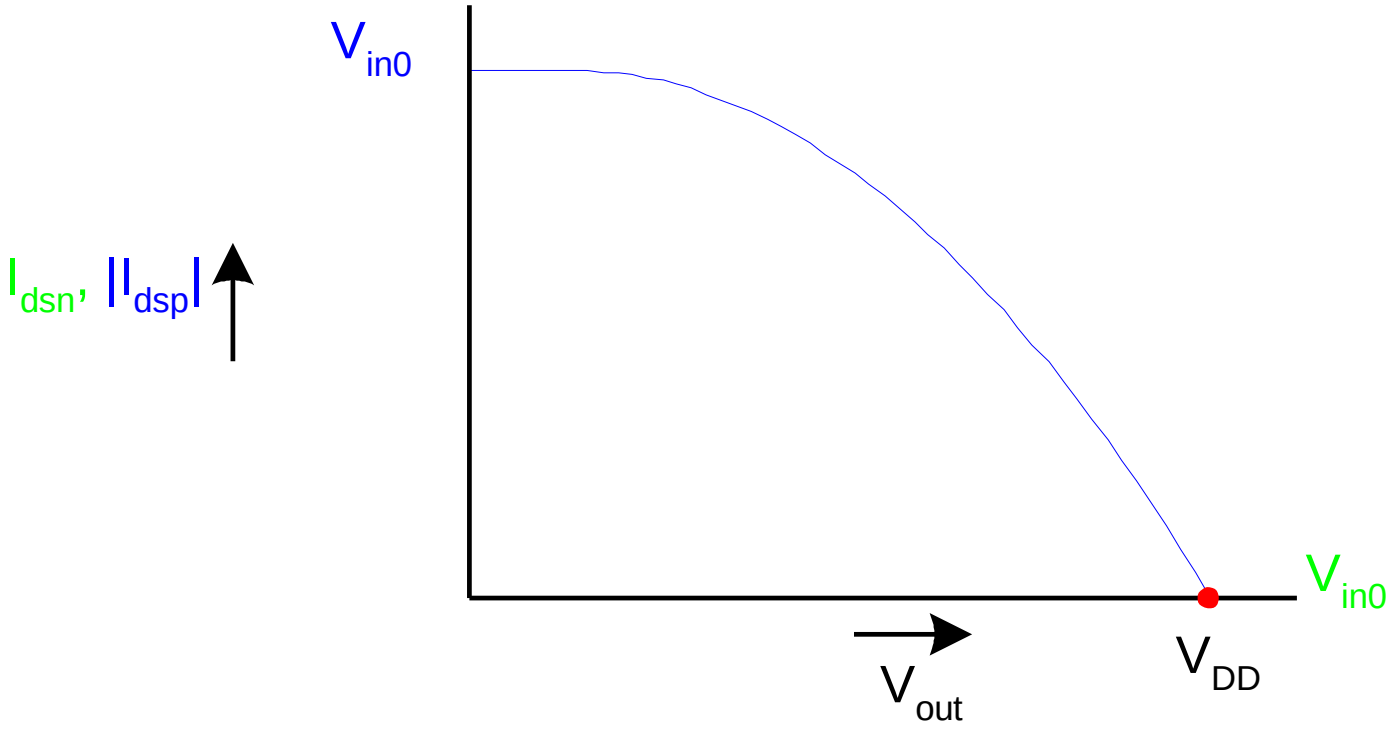
Load Line Analysis

- For a given V_{in}:
  - Plot I_{dsn}, I_{dsp} vs. V_{out}
  - V_{out} must be where |currents| are equal in



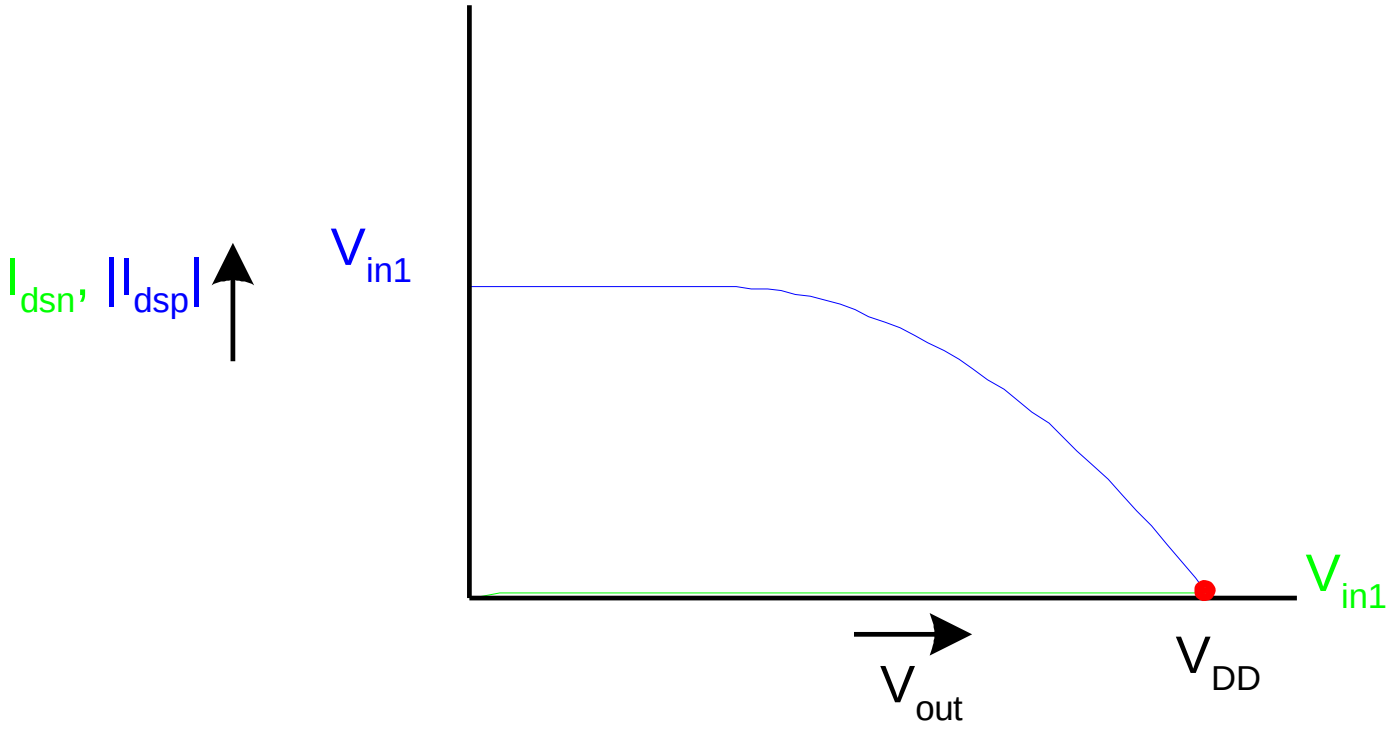
Load Line Analysis

- $V_{in} = 0$



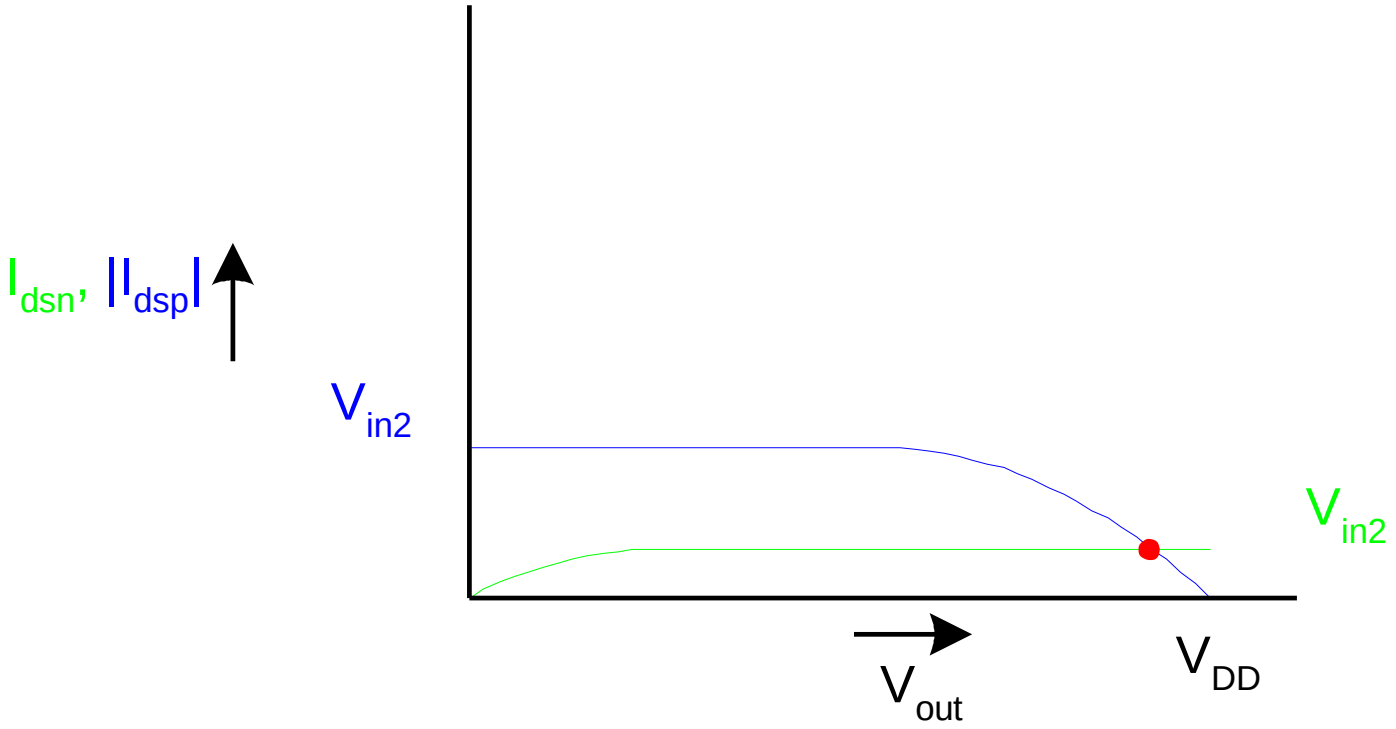
Load Line Analysis

- $V_{in} = 0.2V_{DD}$



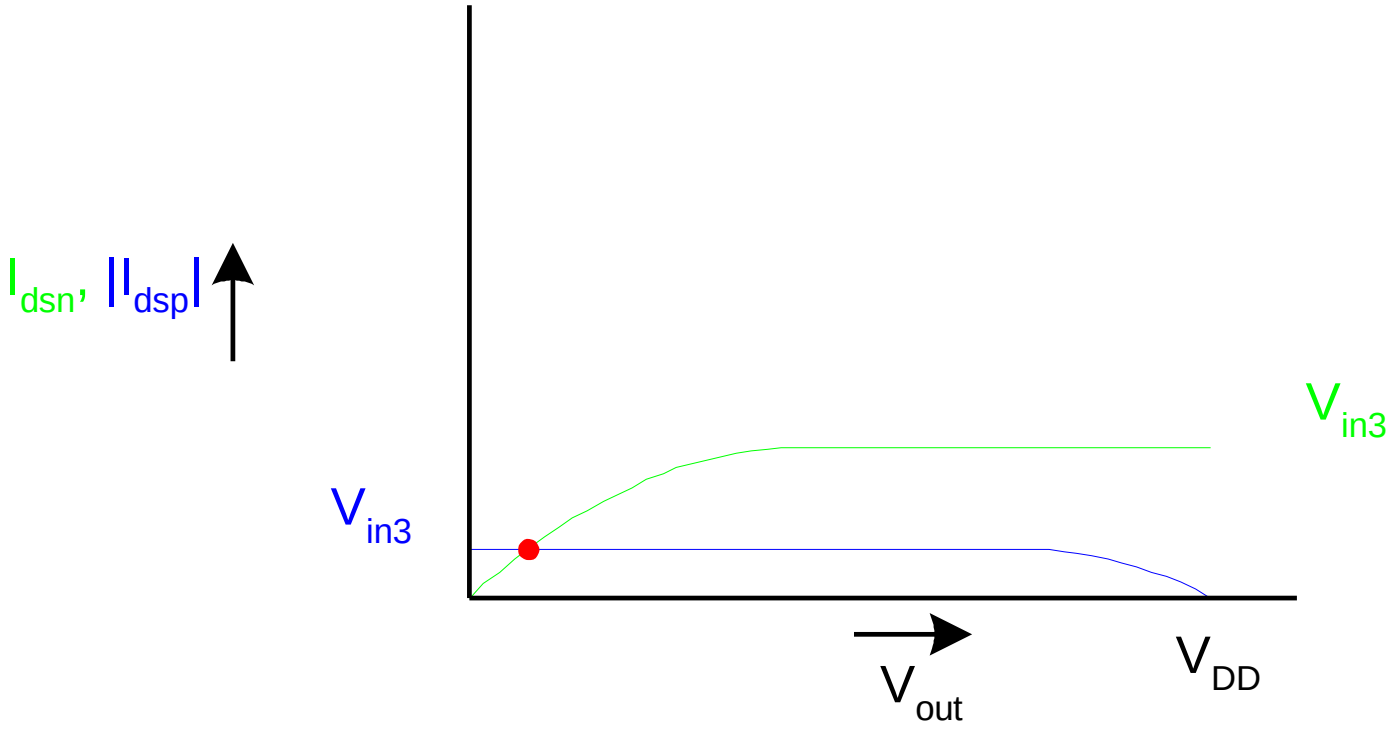
Load Line Analysis

- $V_{in} = 0.4V_{DD}$



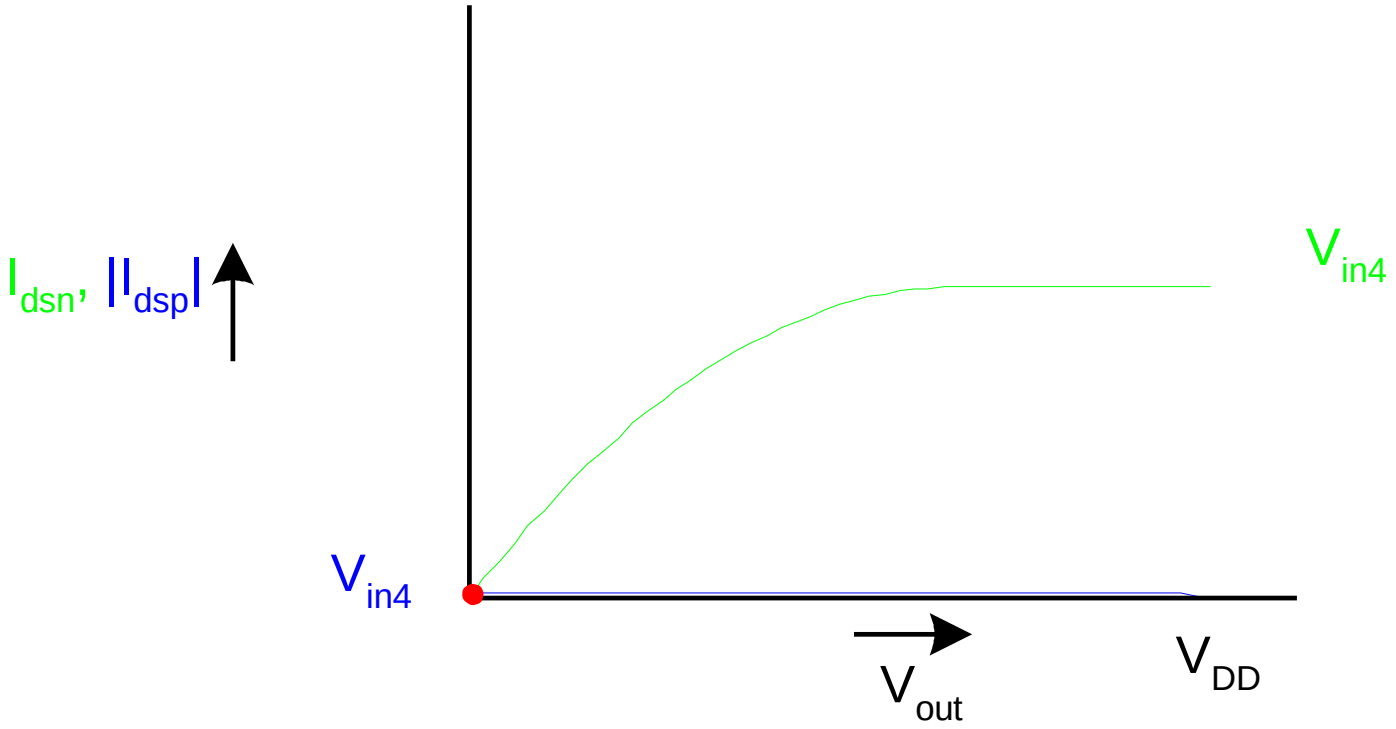
Load Line Analysis

- $V_{in} = 0.6V_{DD}$



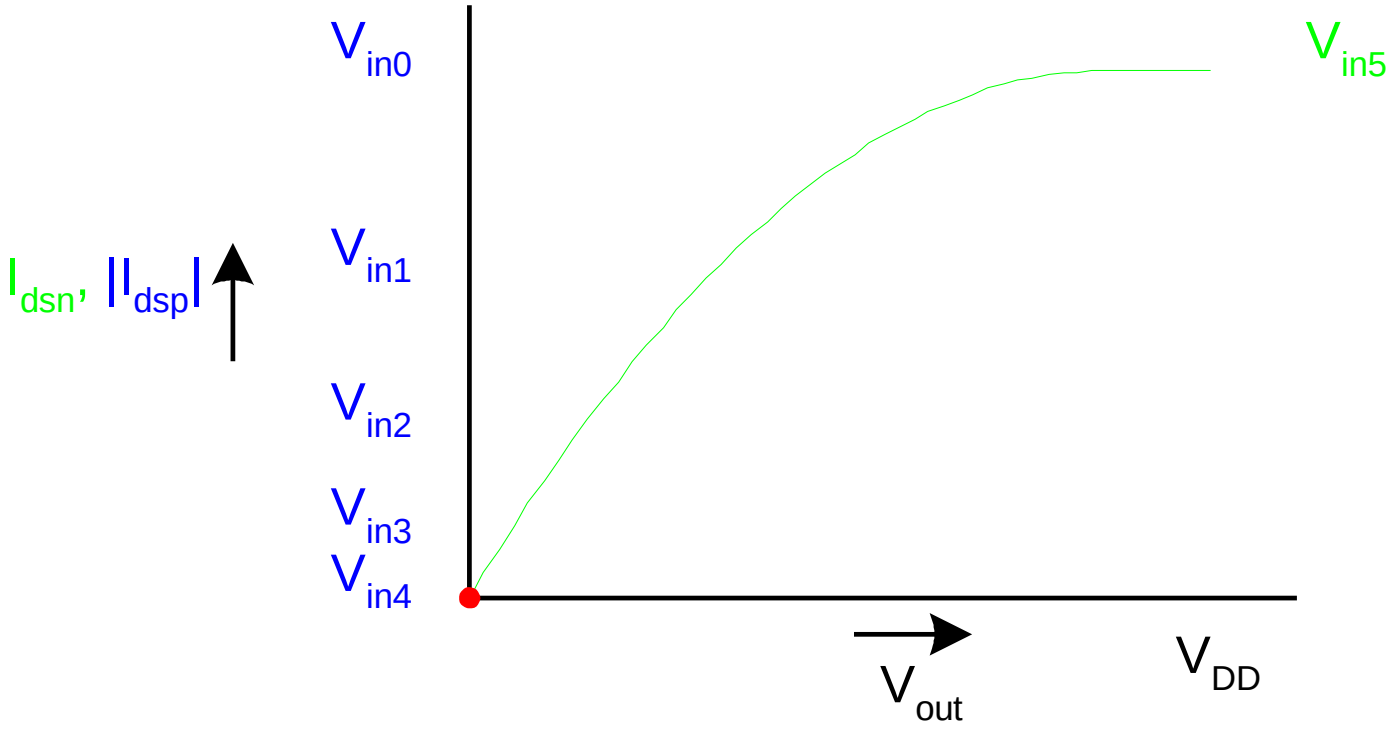
Load Line Analysis

- $V_{in} = 0.8V_{DD}$

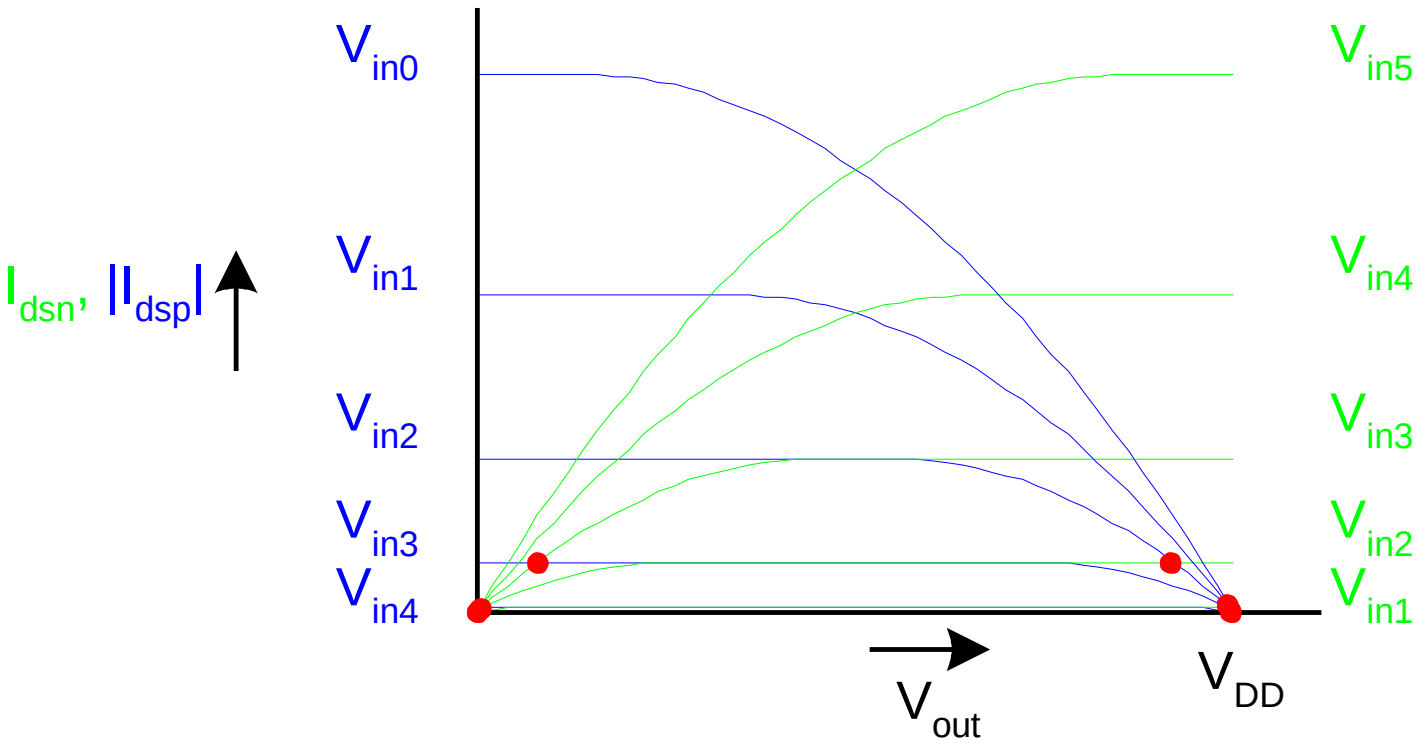


Load Line Analysis

- $V_{in} = V_{DD}$

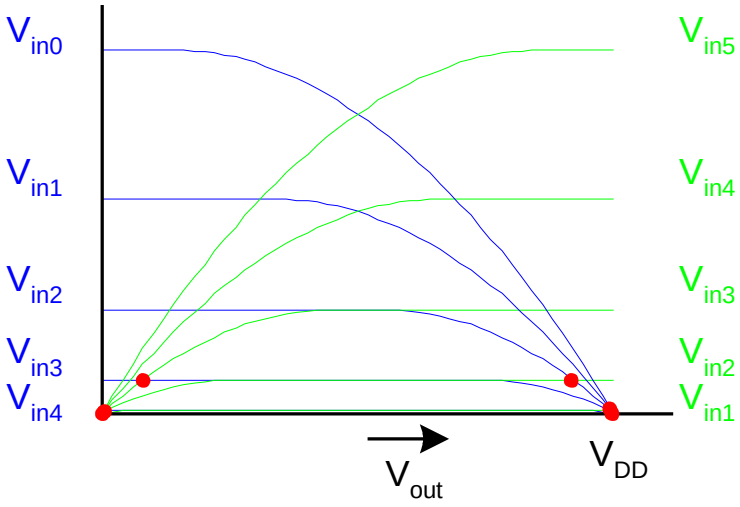


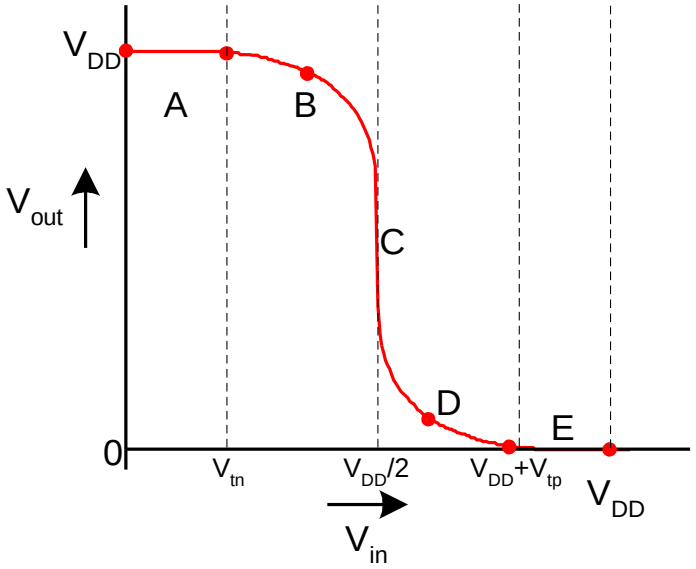
Load Line Summary



DC Transfer Curve

- Transcribe points onto V_{in} vs. V_{out} plot

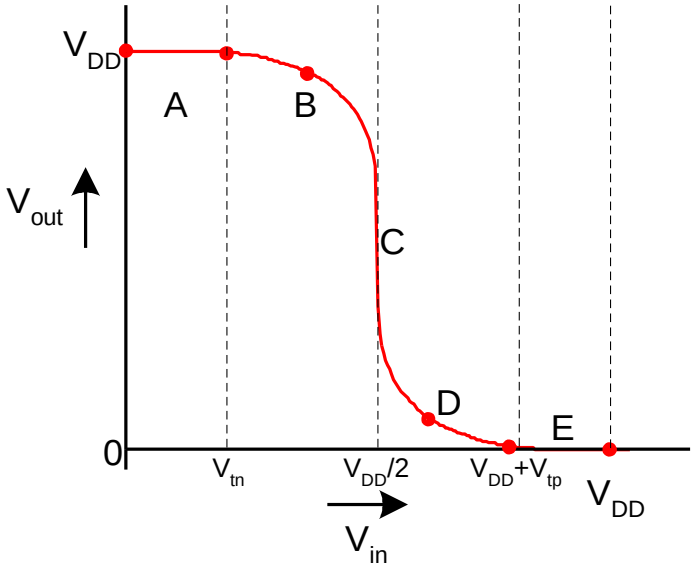




Operating Regions

- Revisit transistor operating regions

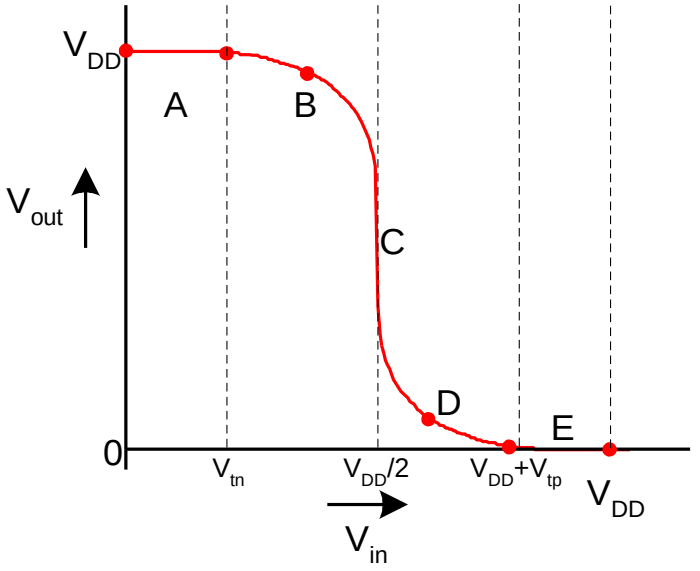
Region	nMOS	pMOS
A		
B		
C		
D		
E		



Operating Regions

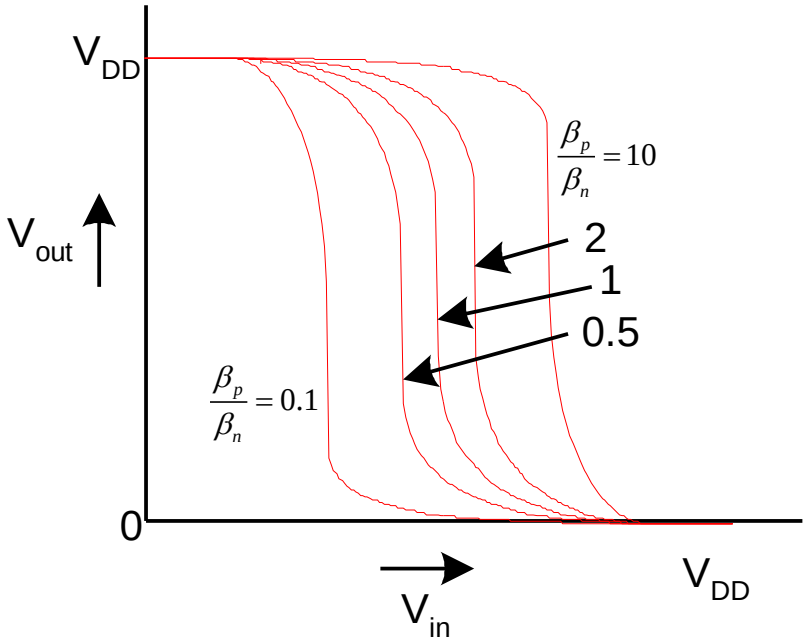
- Revisit transistor operating regions

Region	nMOS	pMOS
A	Cutoff	Linear
B	Saturation	Linear
C	Saturation	Saturation
D	Linear	Saturation
E	Linear	Cutoff



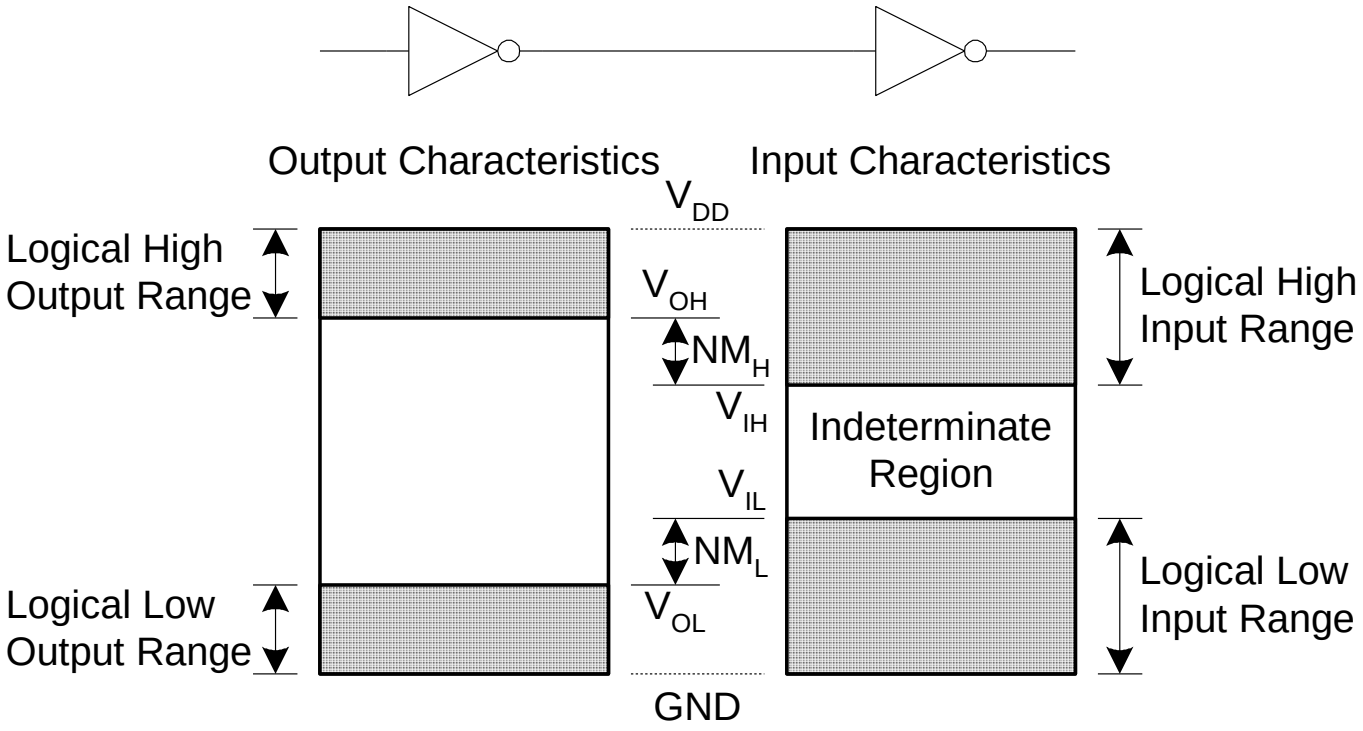
Beta Ratio

- If $\beta_p / \beta_n \neq 1$, switching point will move from $V_{DD}/2$
- Called *skewed* gate
- Other gates: collapse into equivalent inverter



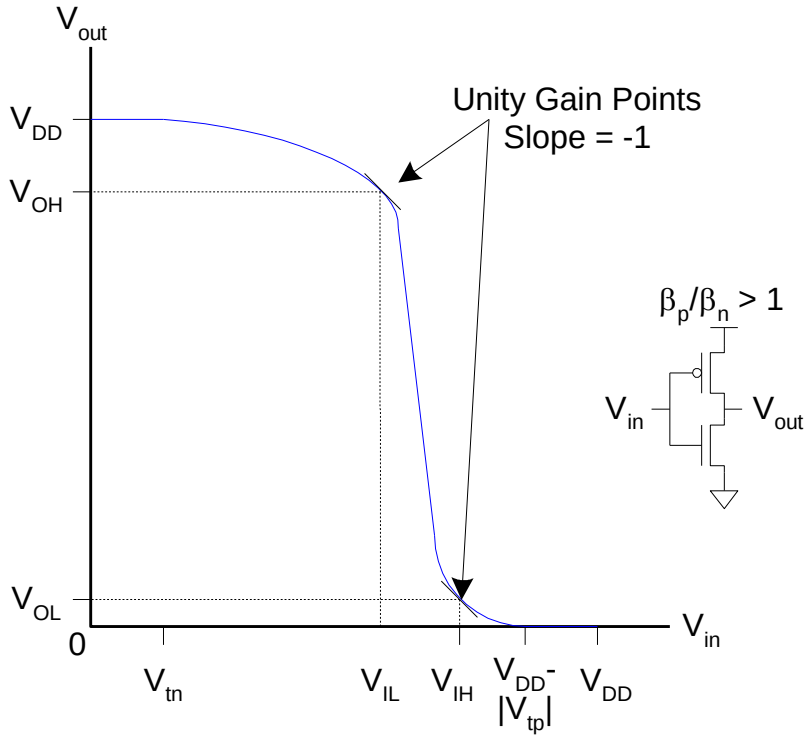
Noise Margins

- How much noise can a gate input see before it does not recognize the input?



Logic Levels

- To maximize noise margins, select logic levels at
  - unity gain point of DC transfer characteristic



Transient Response

- *DC analysis* tells us V_{out} if V_{in} is constant
- *Transient analysis* tells us $V_{out}(t)$ if $V_{in}(t)$ changes
  - Requires solving differential equations
- Input is usually considered to be a step or ramp
  - From 0 to V_{DD} or vice versa

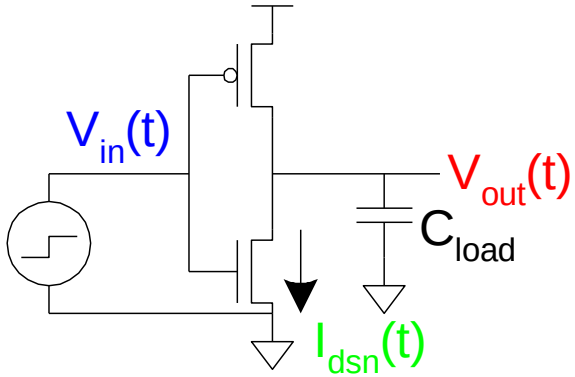
Inverter Step Response

- Ex: find step response of inverter driving load cap

$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$



$$I_{dsn}(t) = \begin{cases} t \le t_0 \\ V_{out} > V_{DD} - V_t \\ V_{out} < V_{DD} - V_t \end{cases}$$

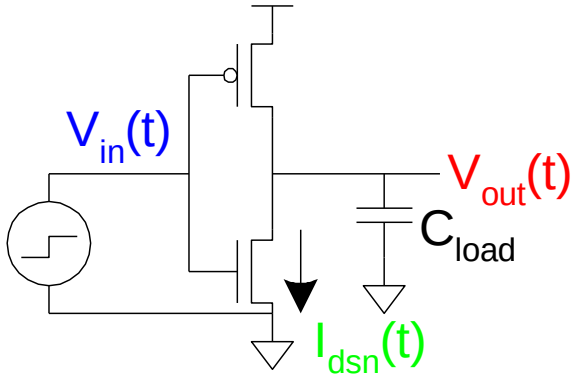
Inverter Step Response

- Ex: find step response of inverter driving load cap

$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$



$$I_{dsn}(t) = \begin{cases} 0 & t \le t_0 \\ \frac{\beta}{2}(V_{DD} - V)^2 & V_{out} > V_{DD} - V_t \\ \beta\left(V_{DD} - V_t - V_{out}(t)/2\right)V_{out}(t) & V_{out} < V_{DD} - V_t \end{cases}$$

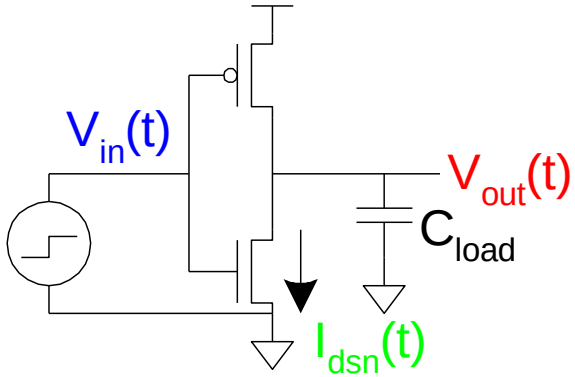
Inverter Step Response

- Ex: find step response of inverter driving load cap

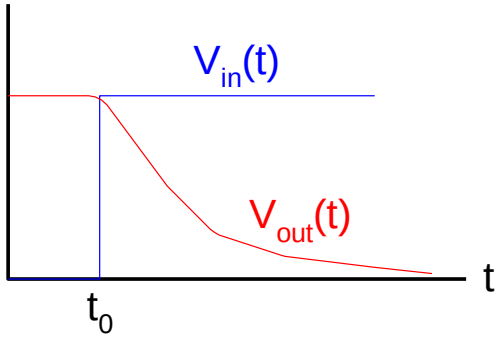
$$V_{in}(t) = u(t - t_0)V_{DD}$$

$$V_{out}(t < t_0) = V_{DD}$$

$$\frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$$



$$I_{dsn}(t) = \begin{cases} 0 & t \le t_0 \\ \frac{\beta}{2}(V_{DD} - V)^2 & V_{out} > V_{DD} - V_t \\ \beta\left(V_{DD} - V_t - V_{out}(t)/2\right)V_{out}(t) & V_{out} < V_{DD} - V_t \end{cases}$$



Delay Definitions

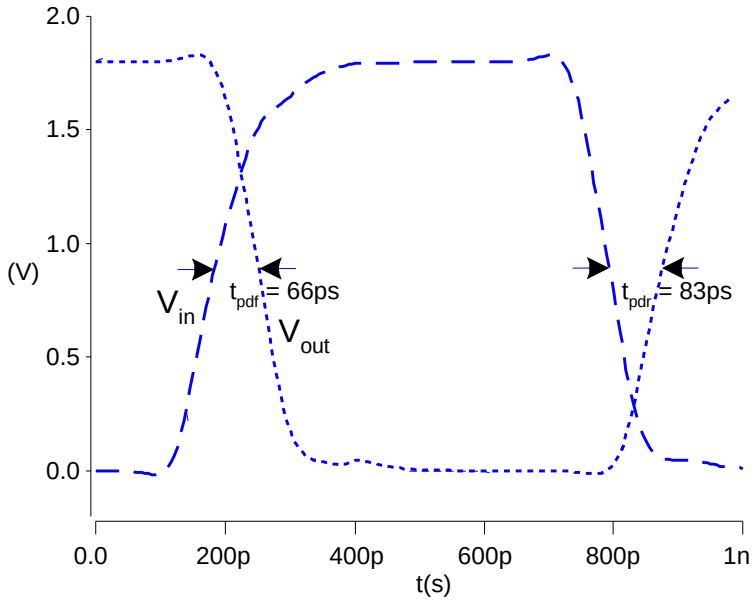
- **t_{pdr}**: maximum *rising propagation delay*
  - From input to rising output crossing $V_{DD}/2$
- **t_{pdf}**: maximum *falling propagation delay*
  - From input to falling output crossing $V_{DD}/2$
- **t_{pd}**: *average propagation delay*
  - $t_{pd} = (t_{pdr} + t_{pdf})/2$
- **t_r**: *rise time*
  - From output crossing 0.2 V_{DD} to 0.8 V_{DD}
- **t_f**: *fall time*
  - From output crossing 0.8 V_{DD} to 0.2 V_{DD}

Delay Definitions

- **t_{cdr}**: minimum *rising contamination delay*
  - From input to rising output crossing $V_{DD}/2$
- **t_{cdf}**: minimum *falling contamination delay*
  - From input to falling output crossing $V_{DD}/2$
- **t_{cd}**: *average contamination delay*
  - $t_{pd} = (t_{cdr} + t_{cdf})/2$

Simulated Inverter Delay

- Solving differential equations by hand is too hard
- SPICE simulator solves the equations numerically
  - Uses more accurate I-V models too!
- But simulations take time to write

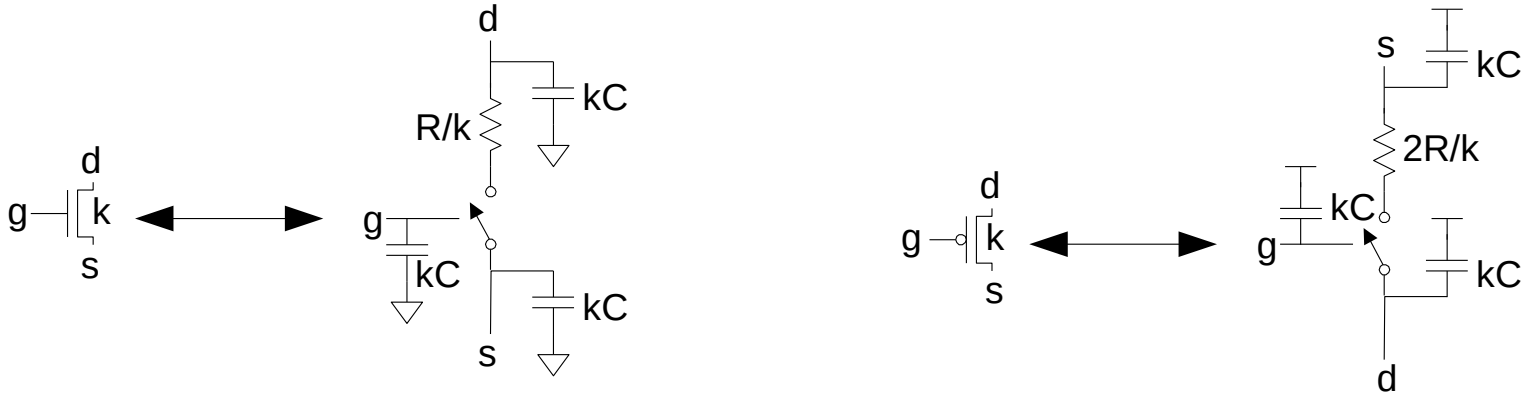


Delay Estimation

- We would like to be able to easily estimate delay
  - Not as accurate as simulation
  - But easier to ask “What if?”
- The step response usually looks like a 1st order RC response with a decaying exponential.
- Use RC delay models to estimate delay
  - C = total capacitance on output node
  - Use *effective resistance* R
  - So that t_{pd} = RC
- Characterize transistors by finding their effective R
  - Depends on average current as gate switches

RC Delay Models

- Use equivalent circuits for MOS transistors
  - Ideal switch + capacitance and ON resistance
  - Unit nMOS has resistance R, capacitance C
  - Unit pMOS has resistance 2R, capacitance C
- Capacitance proportional to width
- Resistance inversely proportional to width

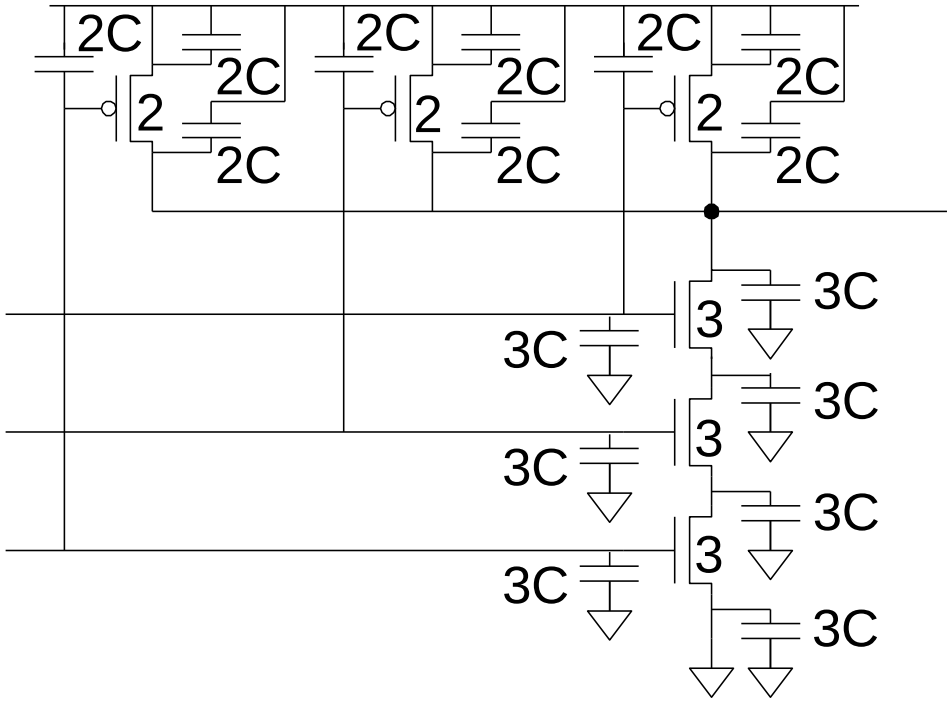


Example: 3-input NAND

- Sketch a 3-input NAND with transistor widths chosen to achieve effective rise and fall resistances equal to a unit inverter (R).

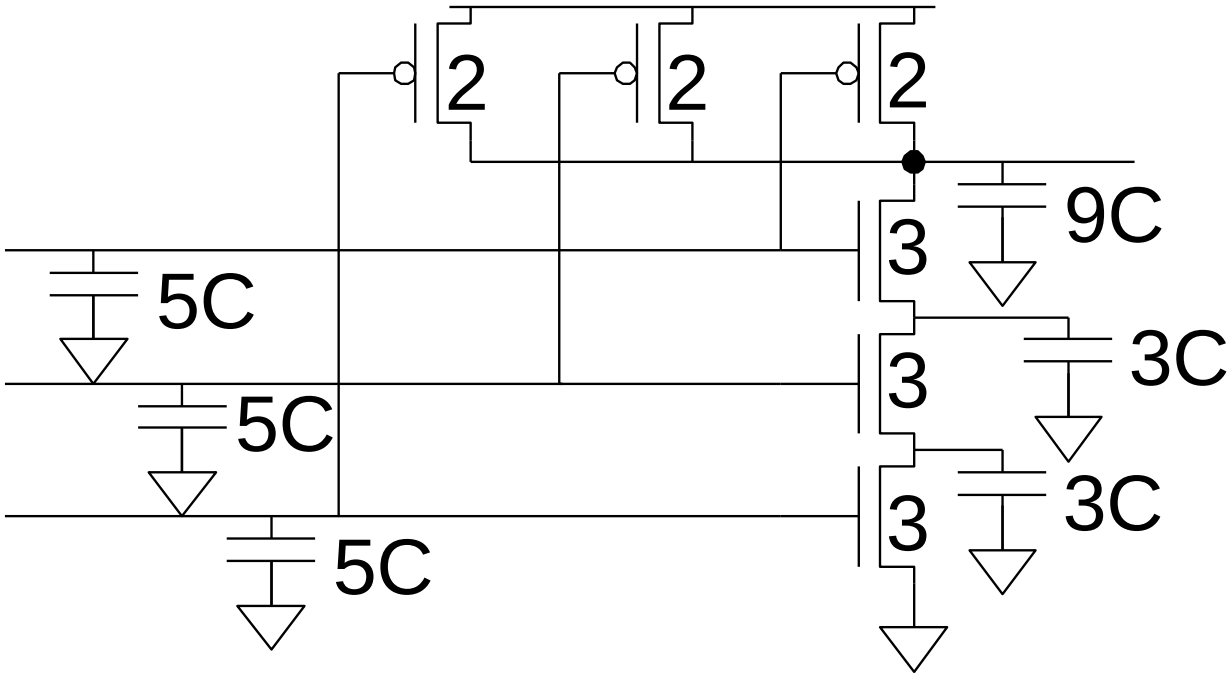
3-input NAND Caps

- Annotate the 3-input NAND gate with gate and diffusion capacitance.



3-input NAND Caps

- Annotate the 3-input NAND gate with gate and diffusion capacitance.

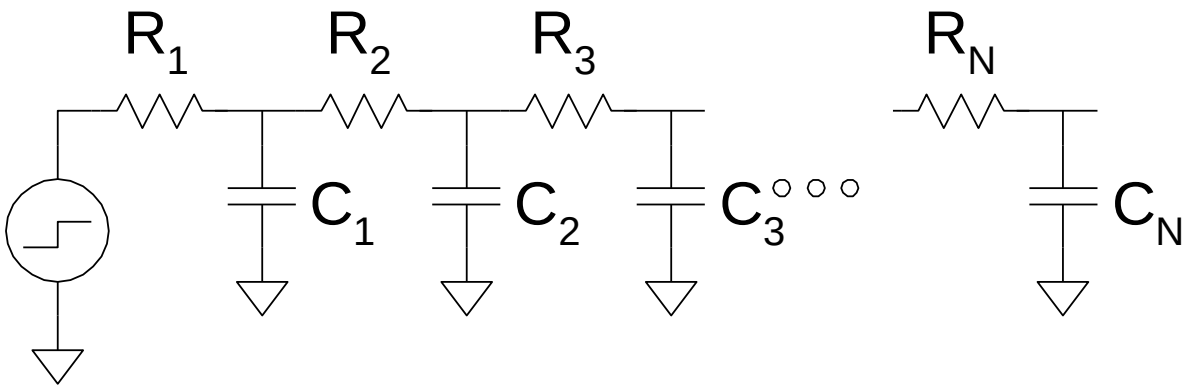


Elmore Delay

- ON transistors look like resistors
- Pullup or pulldown network modeled as *RC ladder*
- Elmore delay of RC ladder

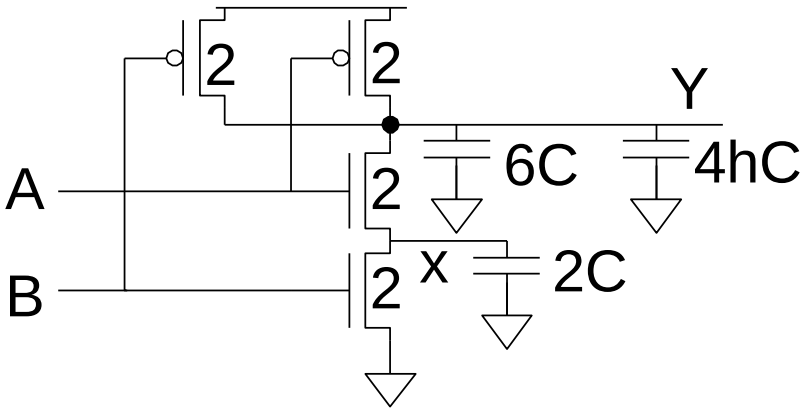
$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$

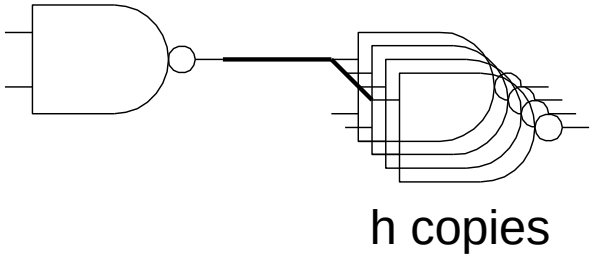
$$= R_1 C_1 + (R_1 + R_2) C_2 + \ldots + (R_1 + R_2 + \ldots + R_N) C_N$$



Example: 2-input NAND

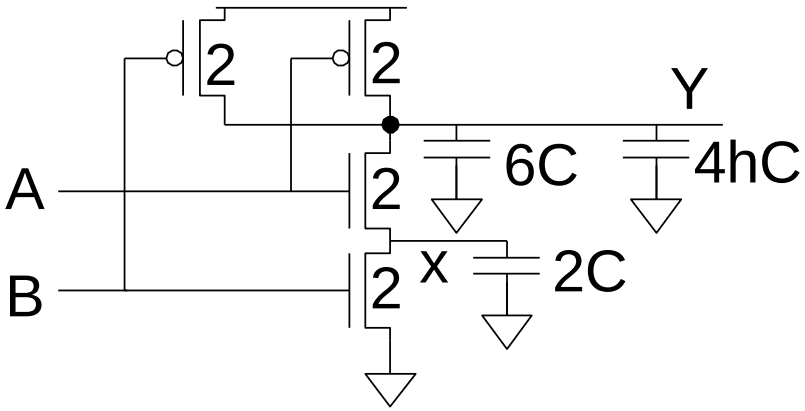
- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.

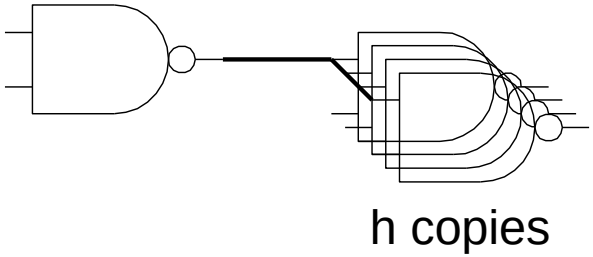


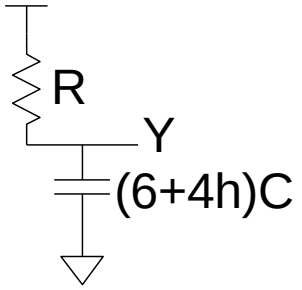


Example: 2-input NAND

- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.



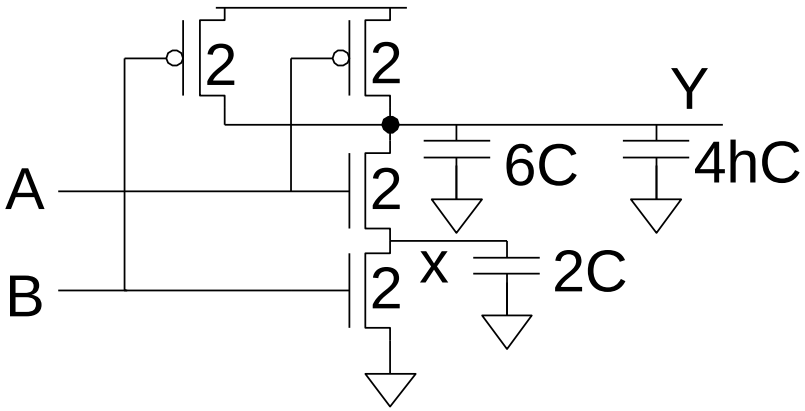


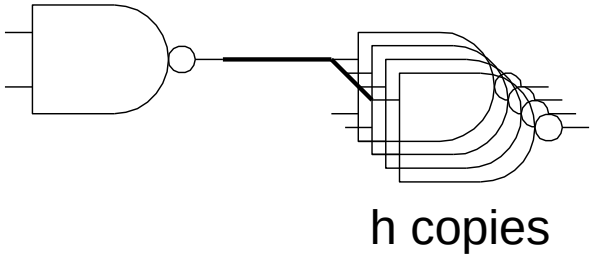


$$t_{pdr} = (6+4h)RC$$

Example: 2-input NAND

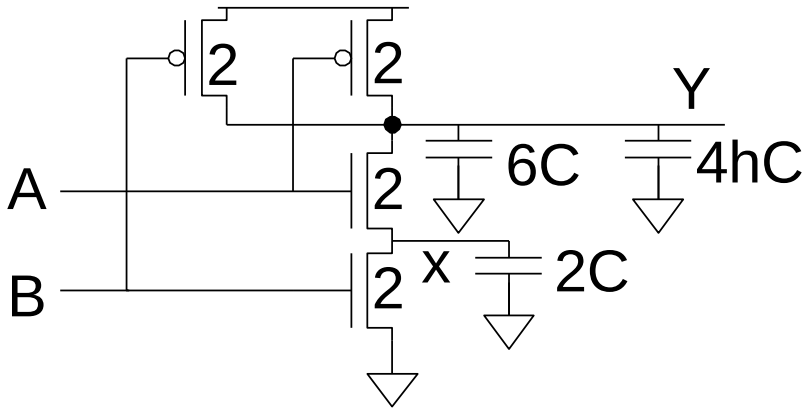
- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.

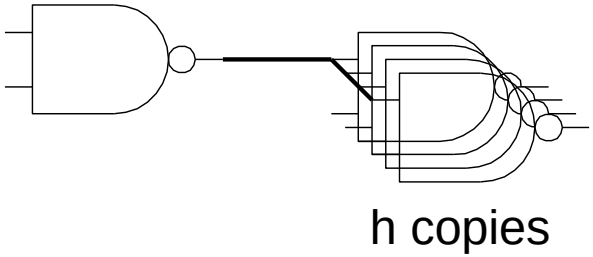


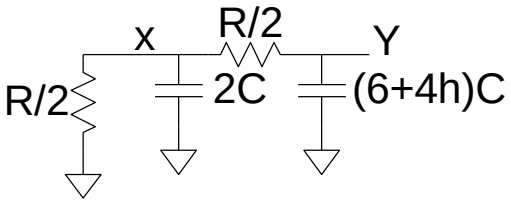


Example: 2-input NAND

- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.



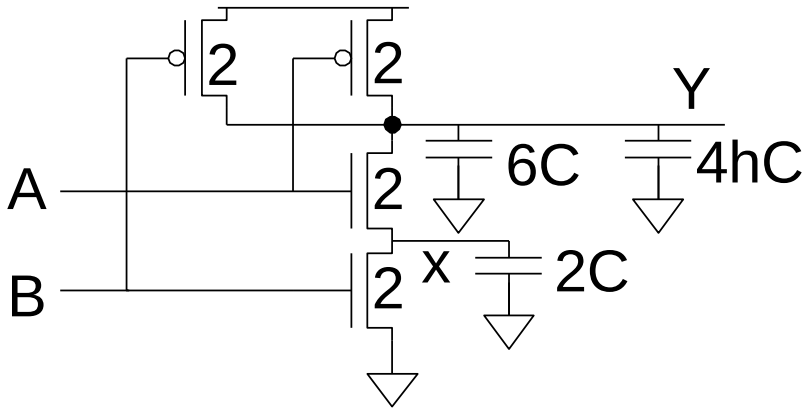


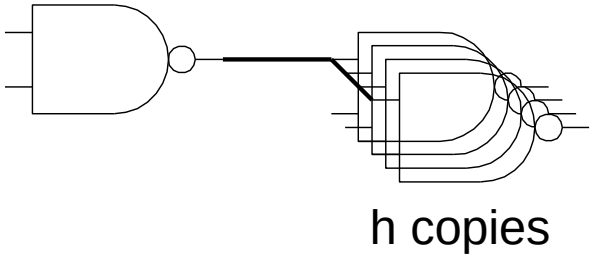


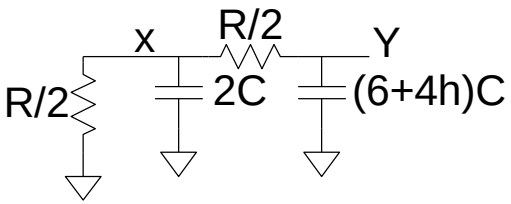
$$t_{pdf} =$$

Example: 2-input NAND

- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.







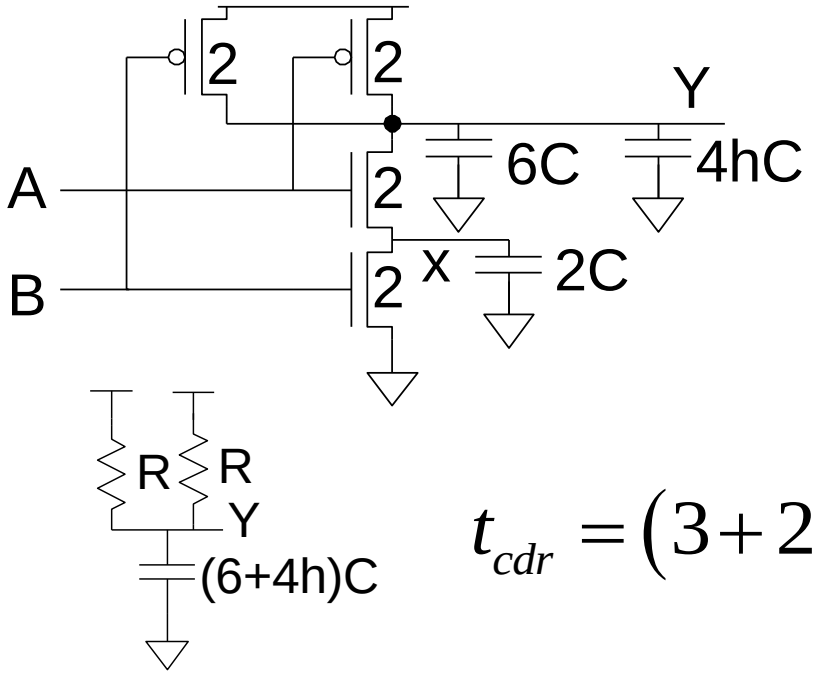
$$t_{pdf} = (2C)\left(\frac{R}{2}\right) + \left[(6+4h)C\right]\left(\frac{R}{2}+\frac{R}{2}\right)$$
$$= (7+4h)RC$$

Delay Components

- Delay has two parts
  - *Parasitic delay*
    - 6 or 7 RC
    - Independent of load
  - *Effort delay*
    - 4h RC
    - Proportional to load capacitance

Contamination Delay

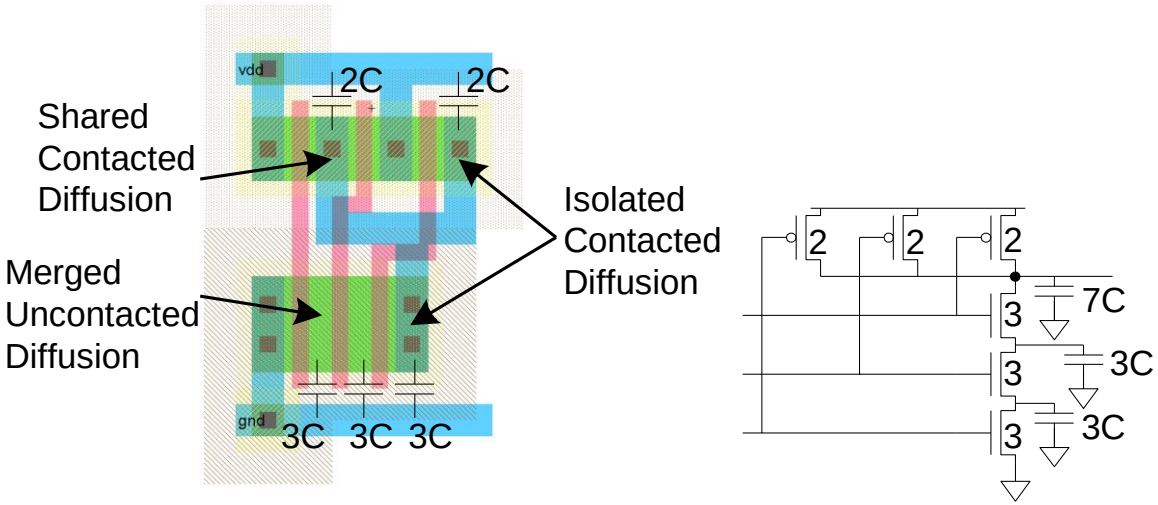
- Best-case (contamination) delay can be substantially less than propagation delay.
- Ex: If both inputs fall simultaneously



$$t_{cdr} = (3+2h)RC$$

Diffusion Capacitance

- we assumed contacted diffusion on every s / d.
- Good layout minimizes diffusion area
- Ex: NAND3 layout shares one diffusion contact
  - Reduces output capacitance by 2C
  - Merged uncontacted diffusion might help too



Layout Comparison

- Which layout is better?