ECE1388 VLSI Design Methodology

Lecture 5:
DC & Transient Response

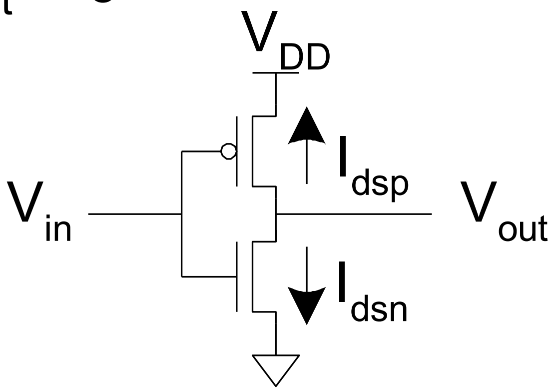
Adapted from Weste, Harris CMOS VLSI Design

Outline

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

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

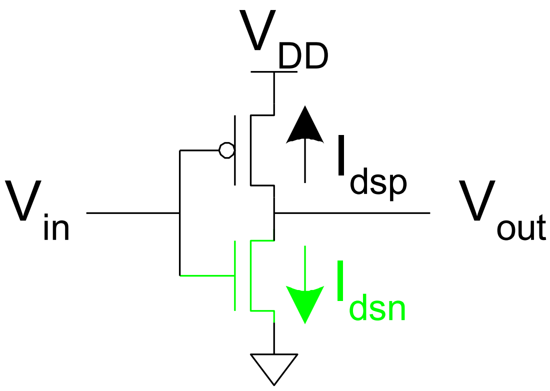
- Current depends on region of transistor behavior
- For what V_{in} and V_{out} are nMOS and pMOS in
  - Cutoff?
  - Linear?
  - Saturation?

nMOS Operation

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

$V_{gsn} = V_{in}$

$V_{dsn} = V_{out}$



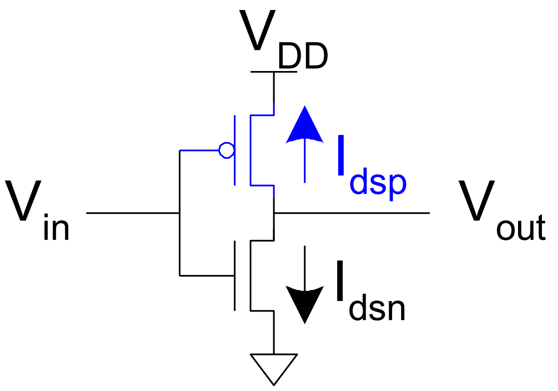
pMOS Operation

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

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

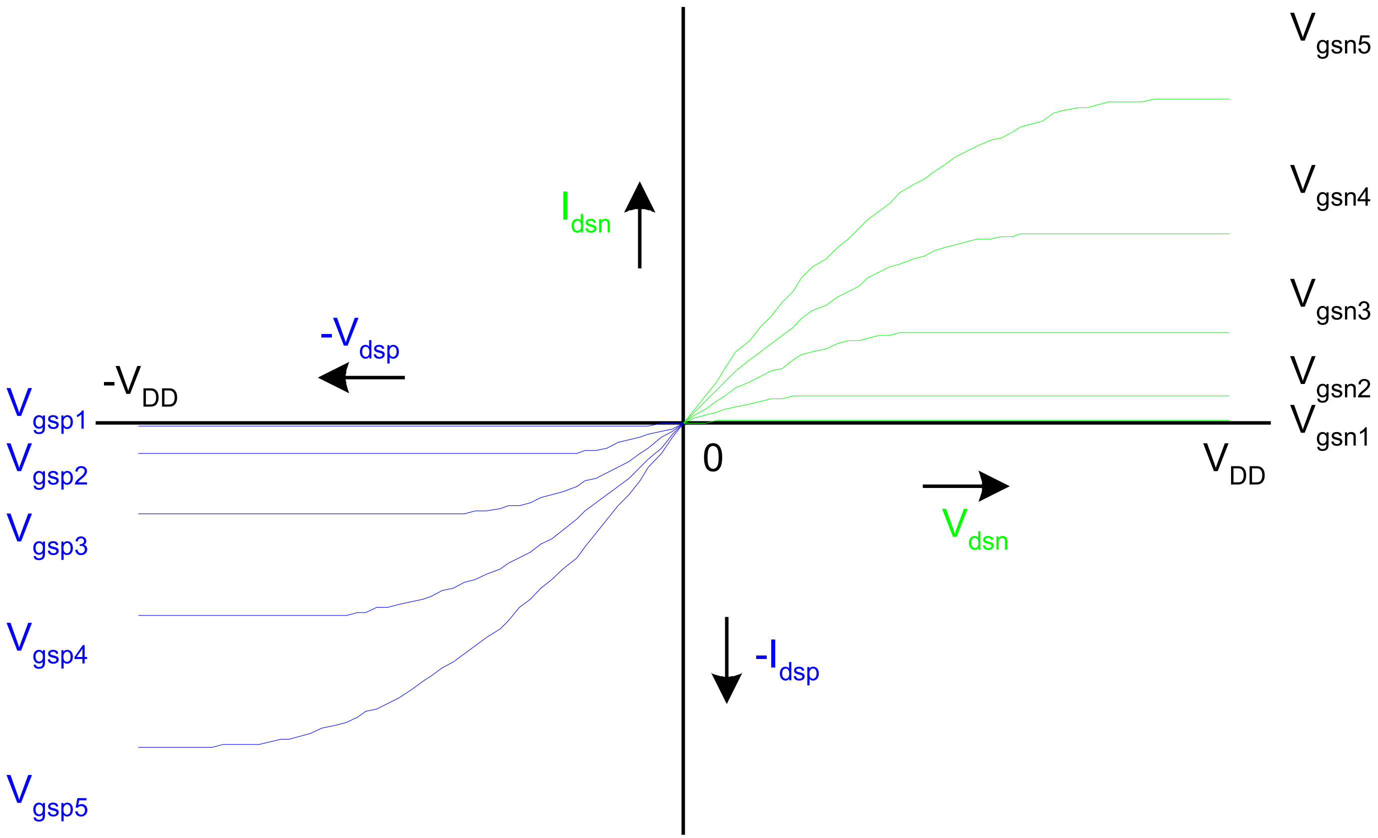
$V_{dsp} = V_{out} - V_{DD}$

$V_{tp} < 0$

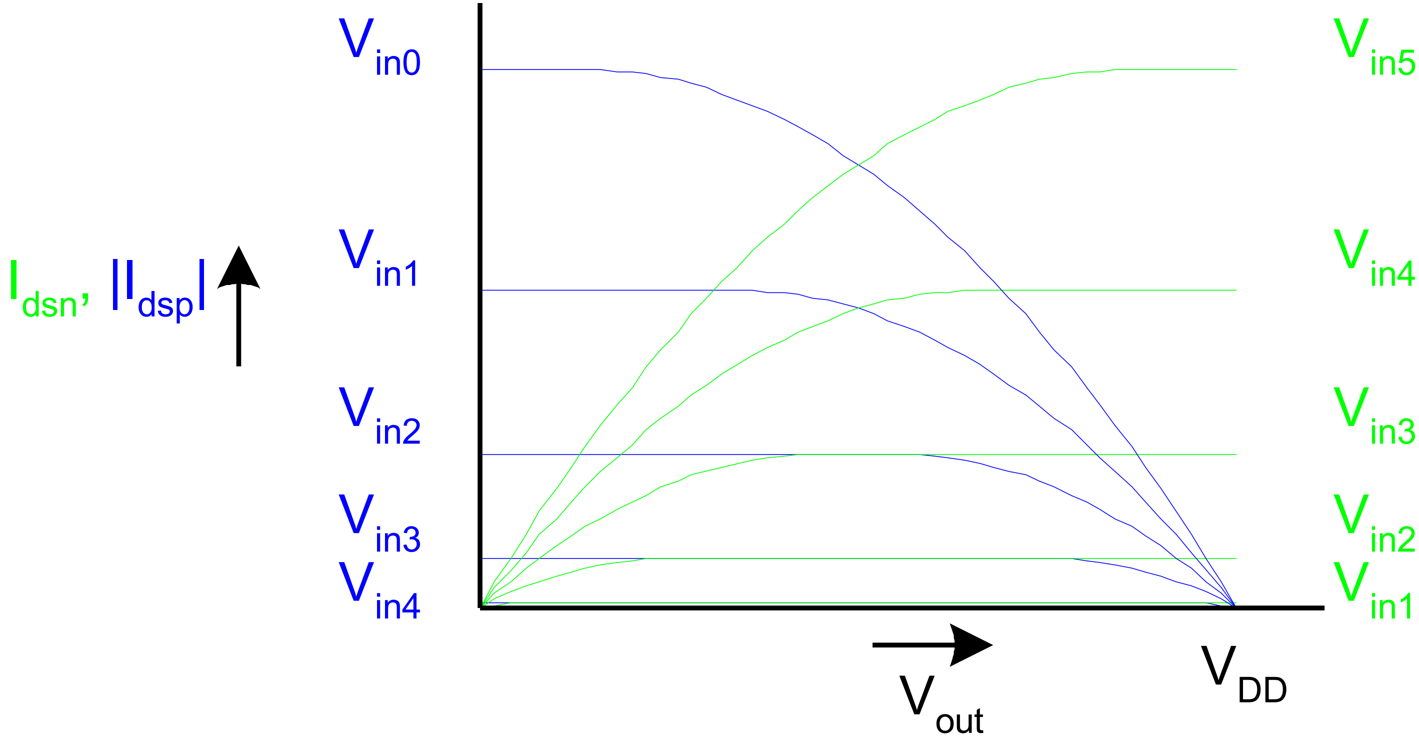


I-V Characteristics

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

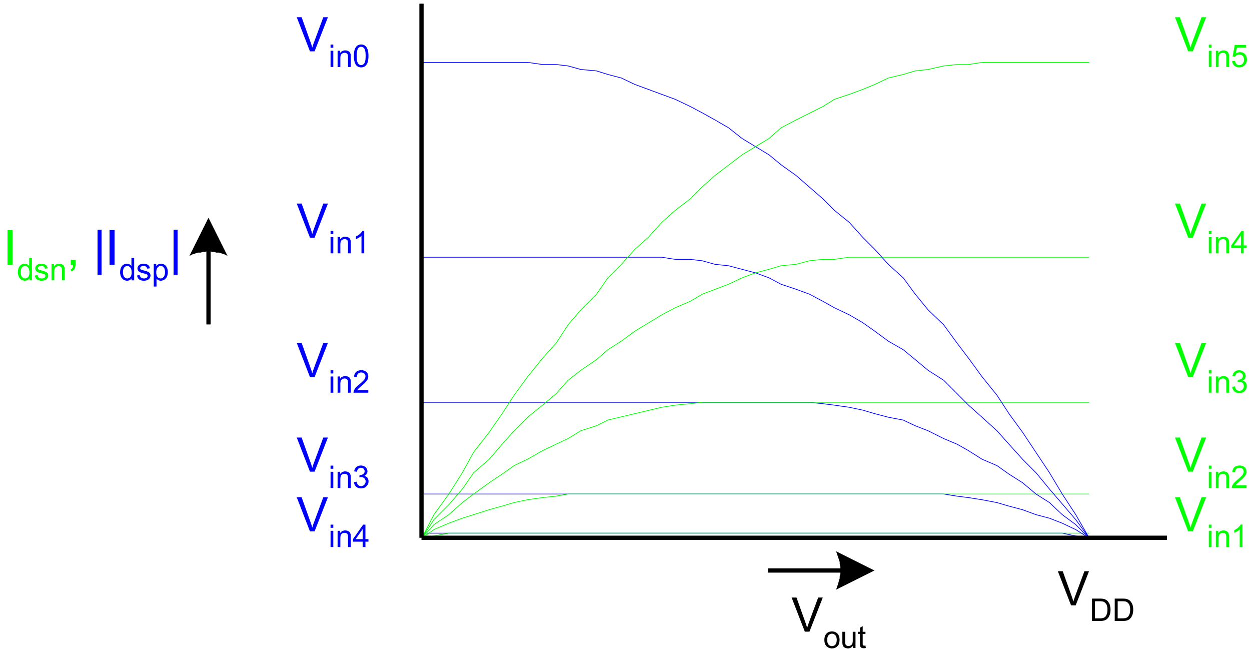


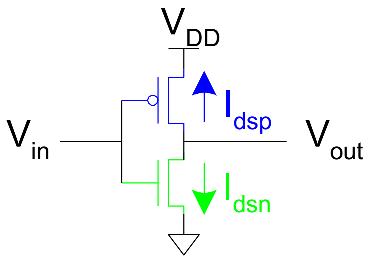
Current vs. V_{out}, V_{in}



Load Line Analysis

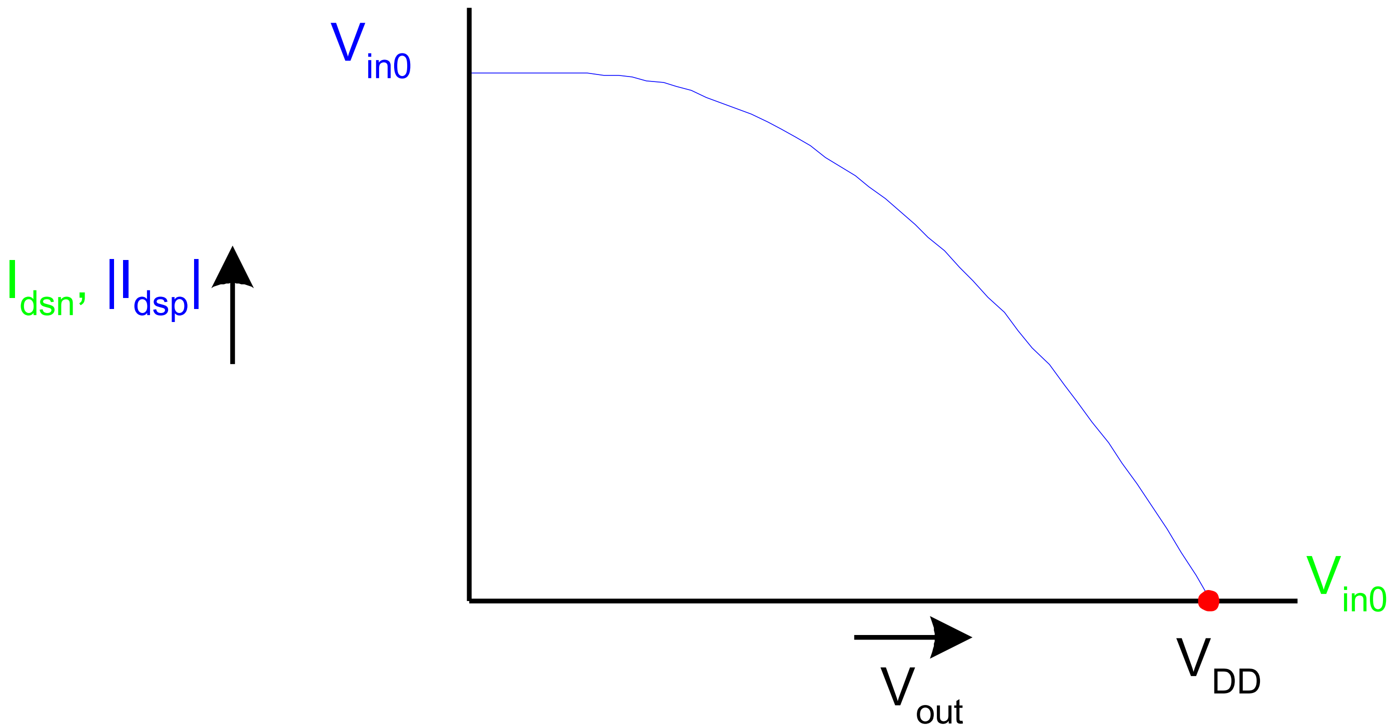
- For a given V_{in}:
  - Plot I_{dsn}, I_{dsp} vs. V_{out}
  - V_{out} must be where |currents| are equal in





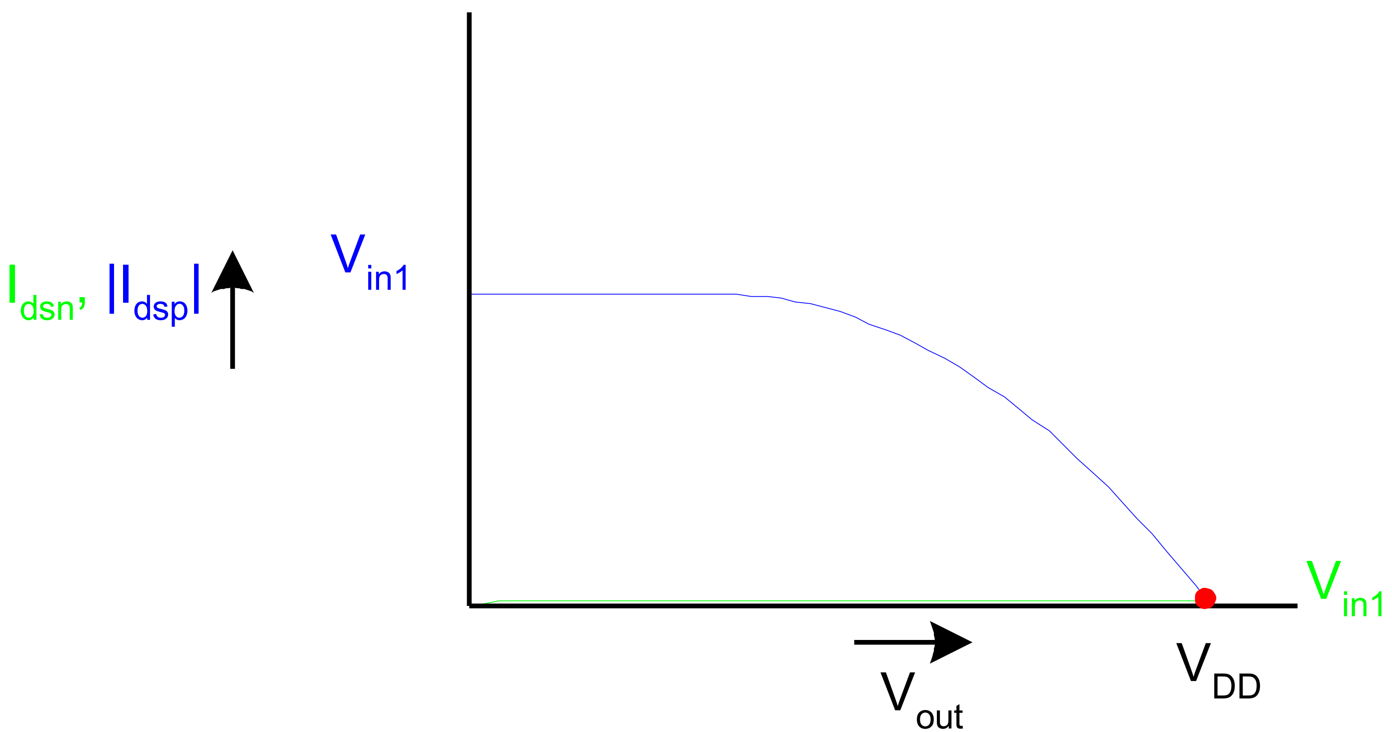
Load Line Analysis

- $V_{in} = 0$



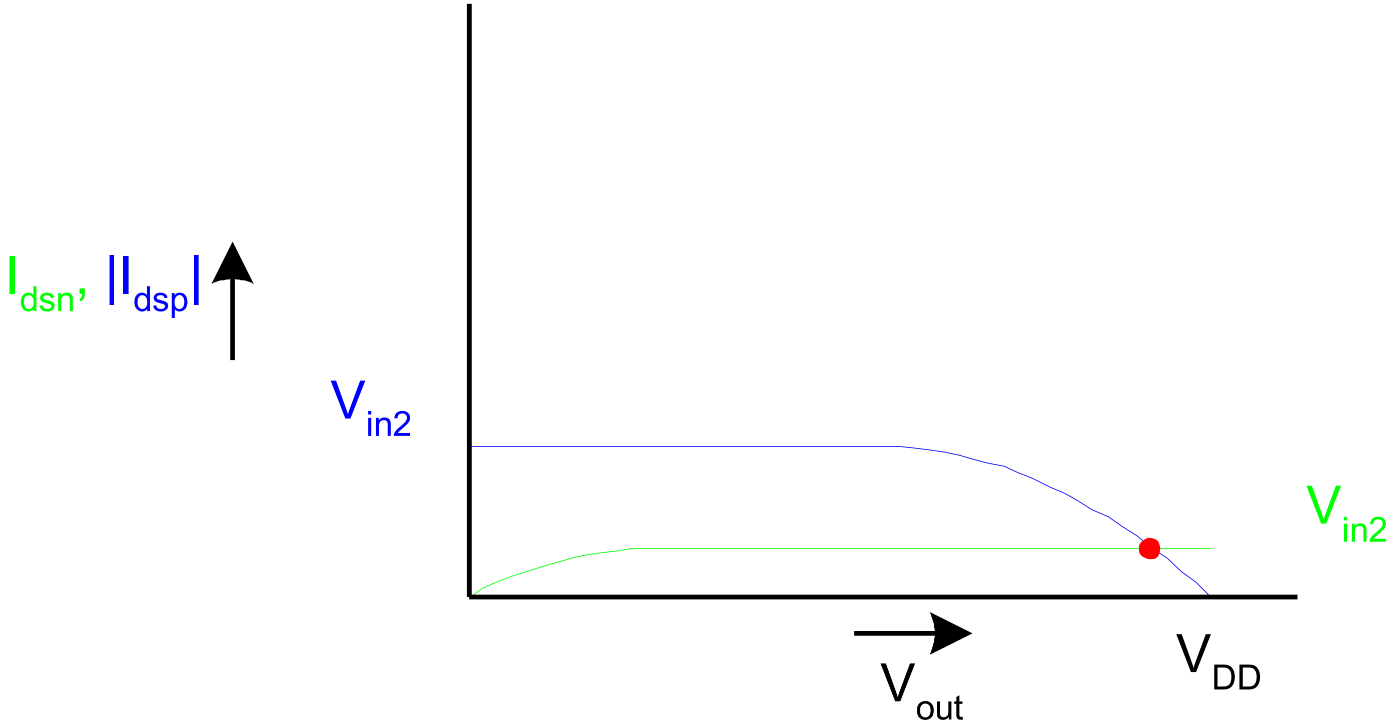
Load Line Analysis

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



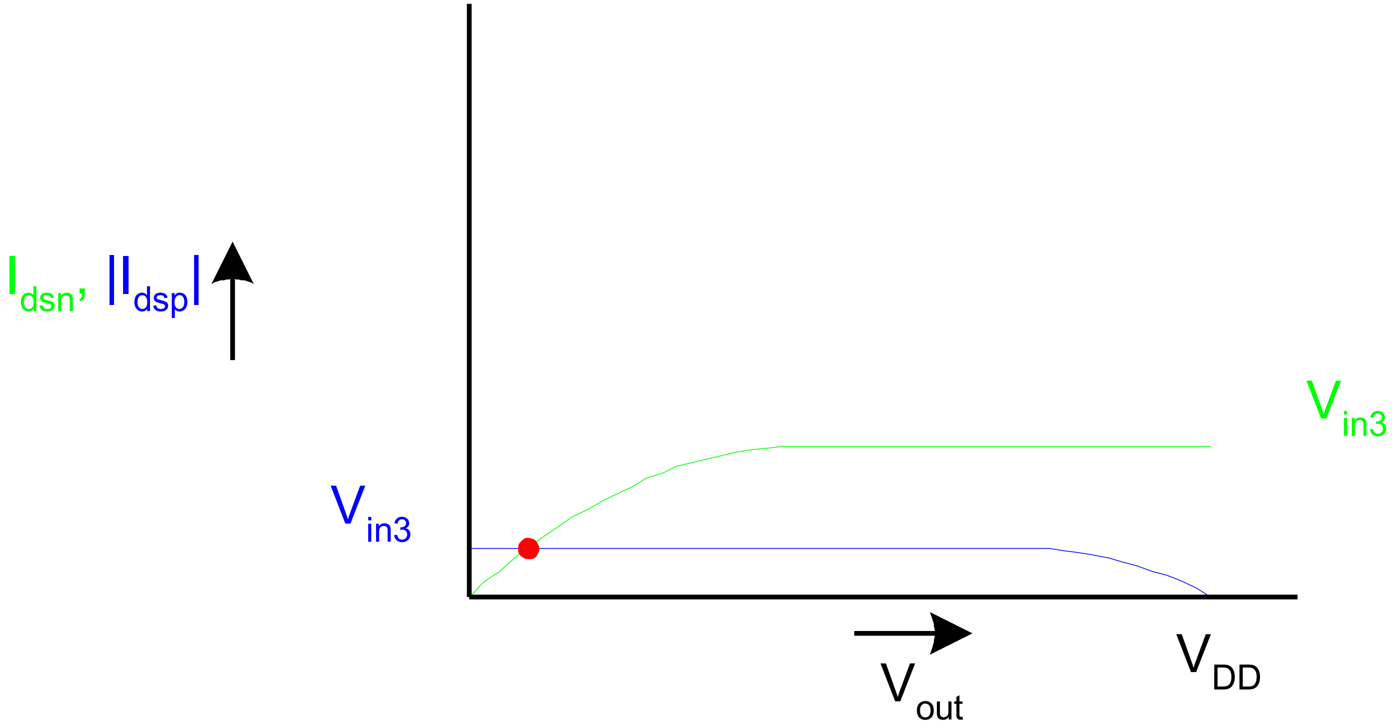
Load Line Analysis

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



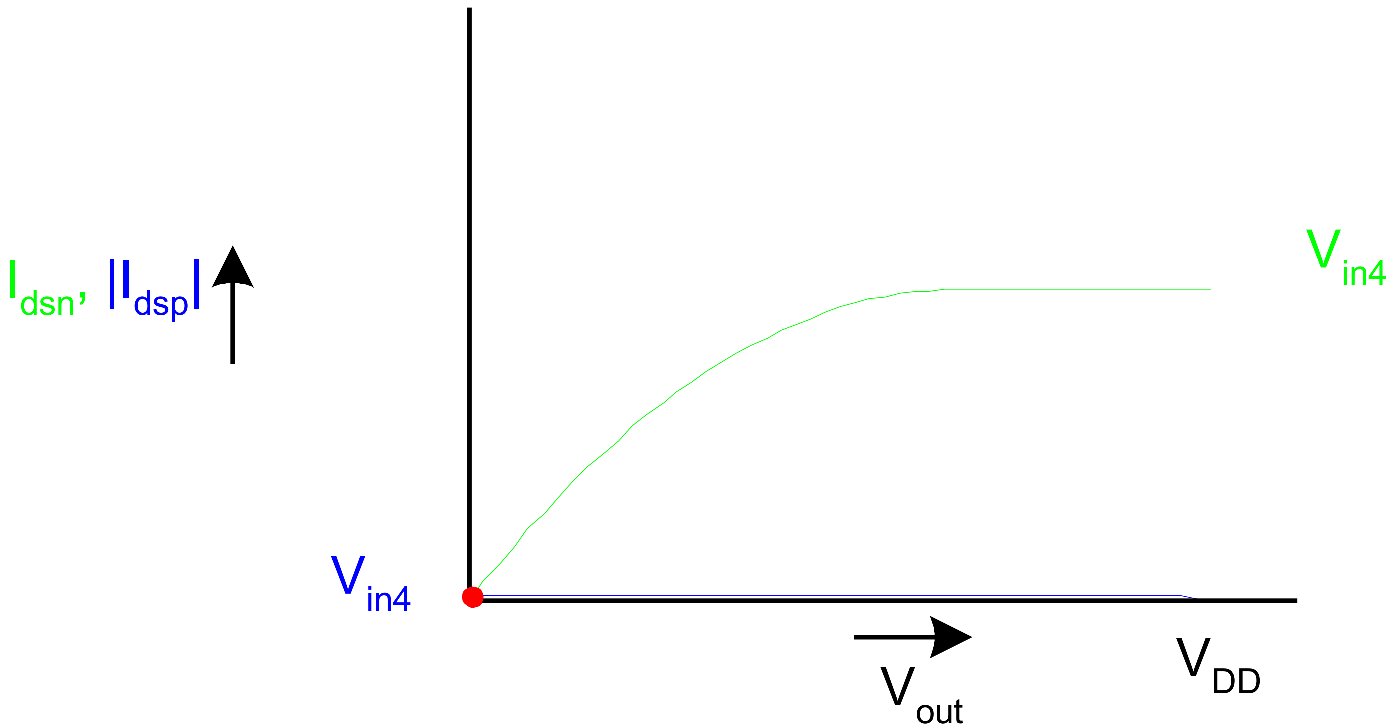
Load Line Analysis

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



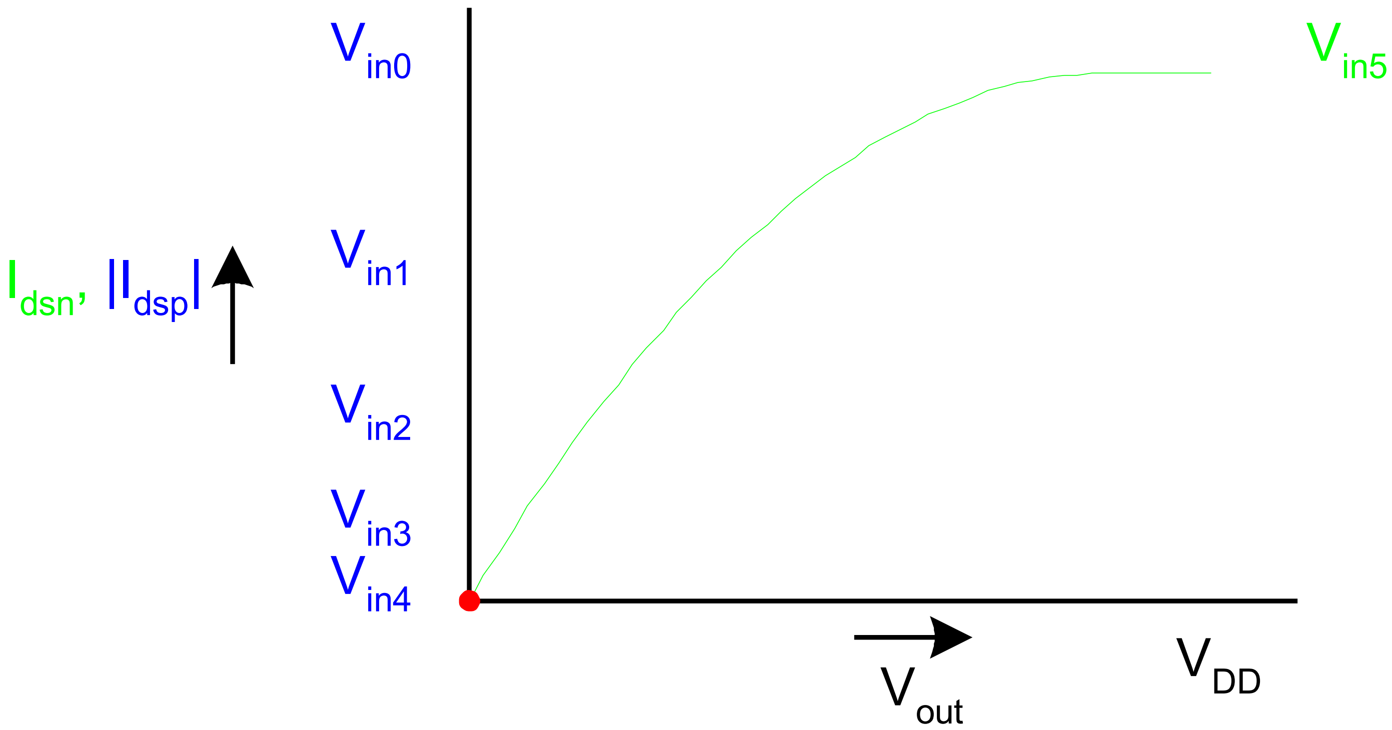
Load Line Analysis

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

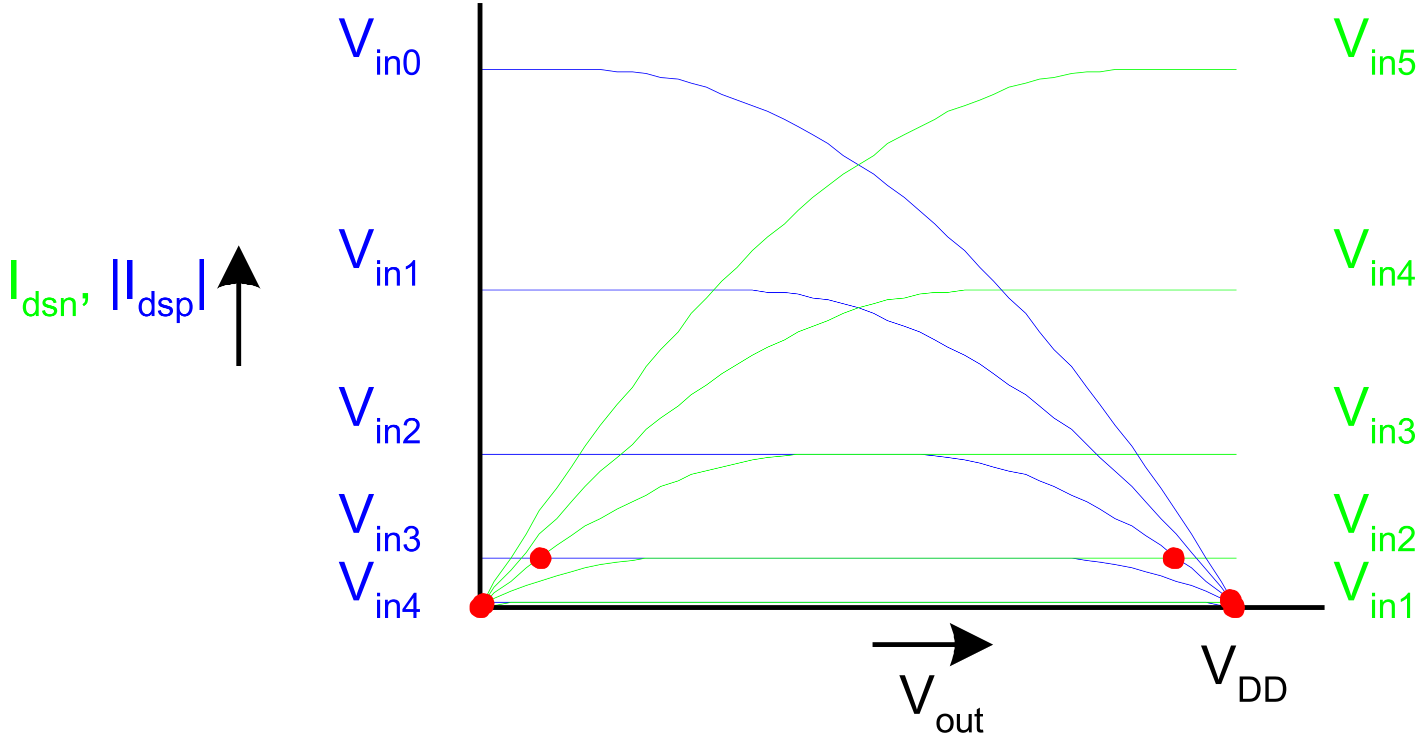


Load Line Analysis

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

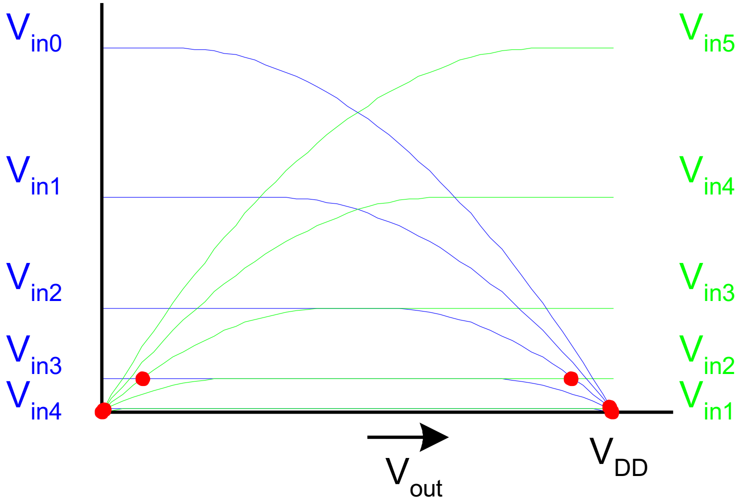


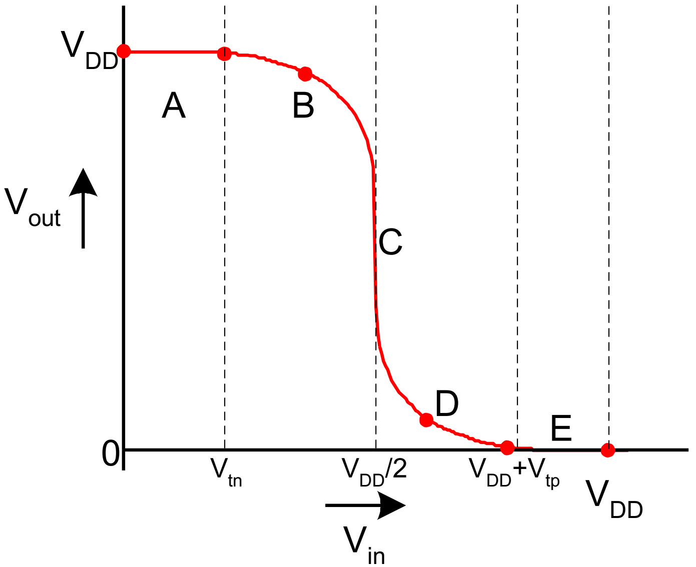
Load Line Summary

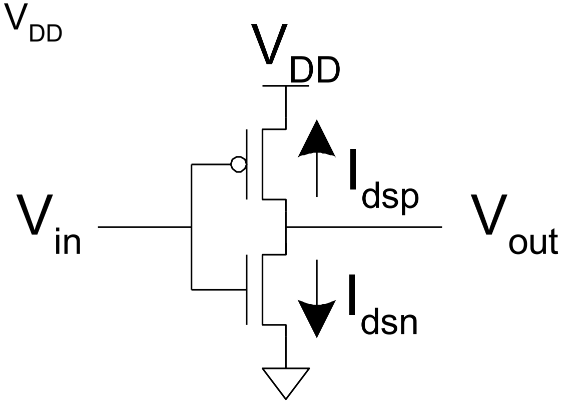


DC Transfer Curve

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



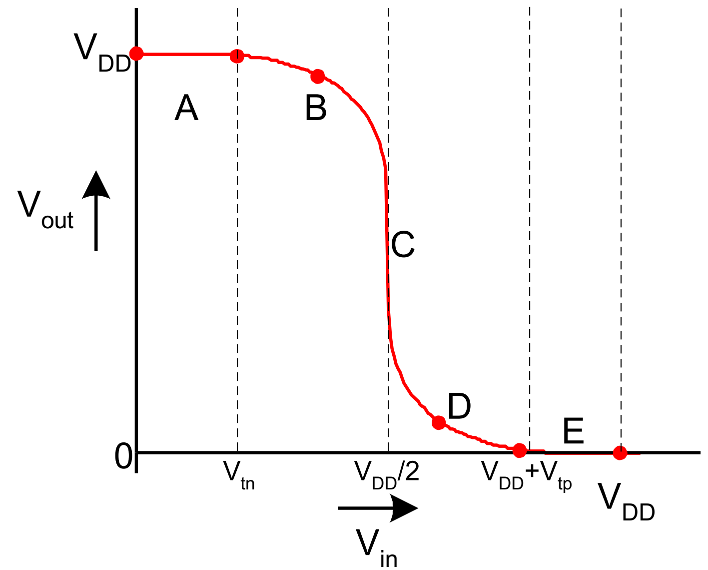


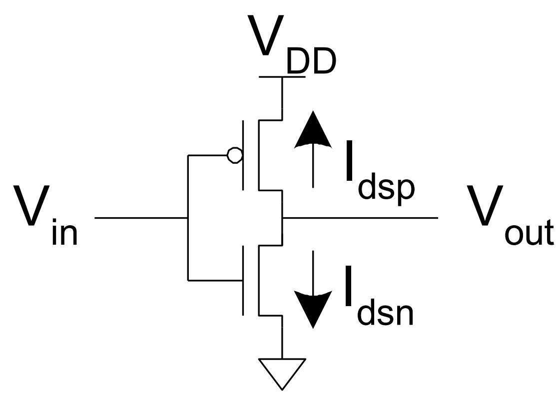


Operating Regions

- Revisit transistor operating regions

Region	nMOS	pMOS
A		
B		
C		
D		
E		

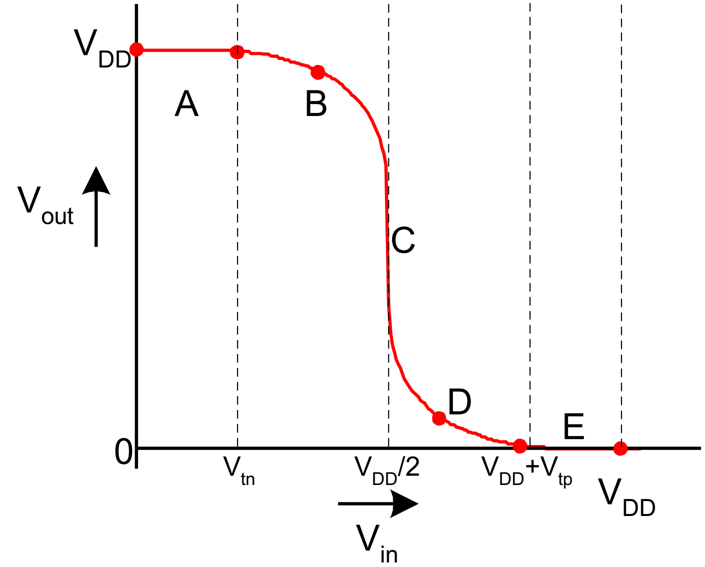


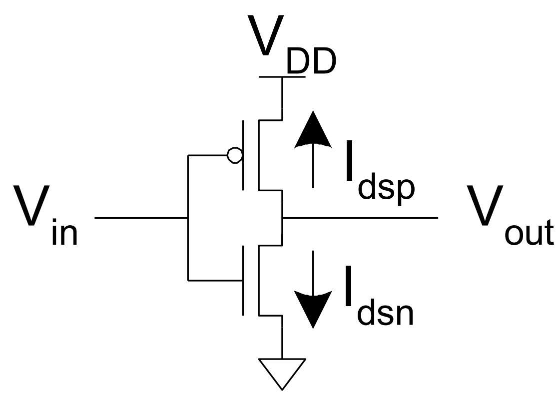


Operating Regions

- Revisit transistor operating regions

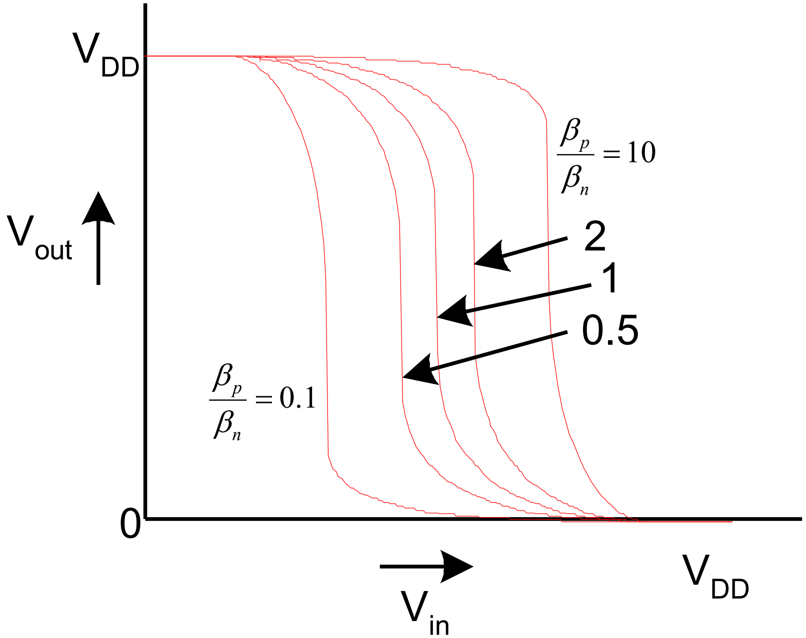
Region	nMOS	pMOS
A	Cutoff	Linear
B	Saturation	Linear
C	Saturation	Saturation
D	Linear	Saturation
E	Linear	Cutoff





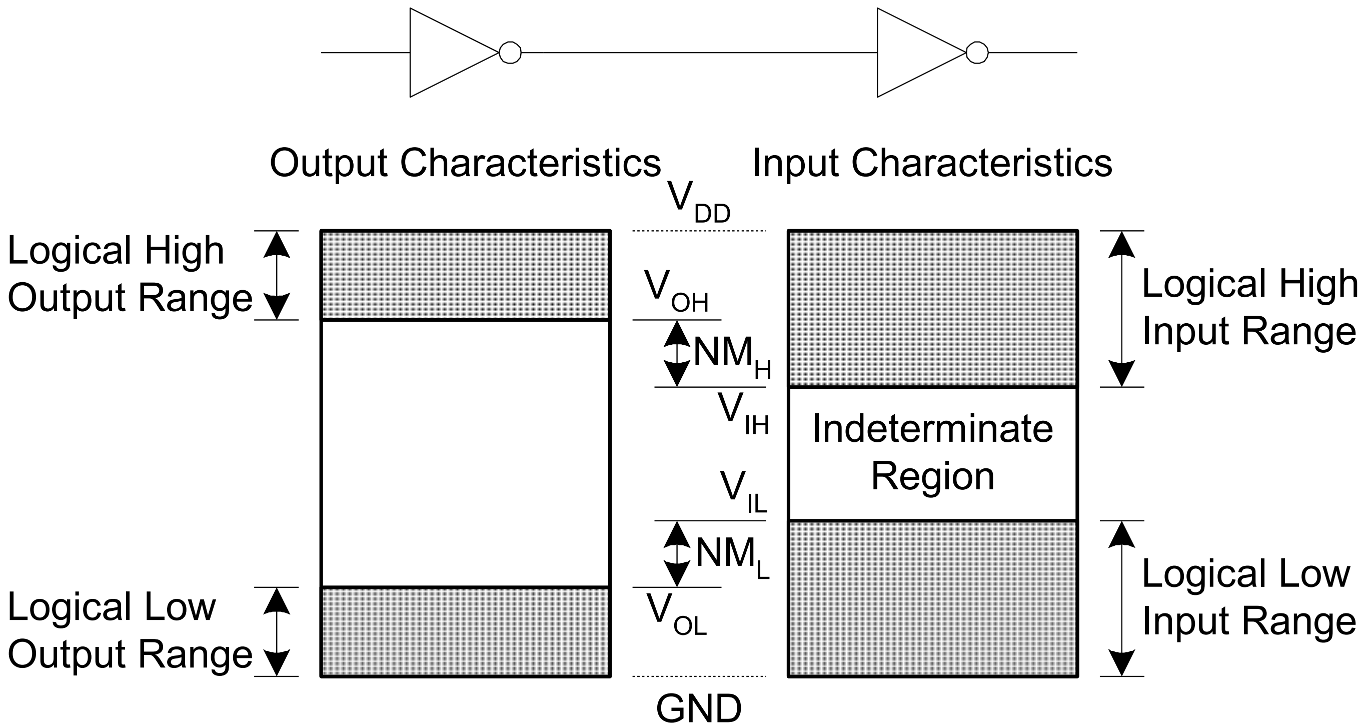
Beta Ratio

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



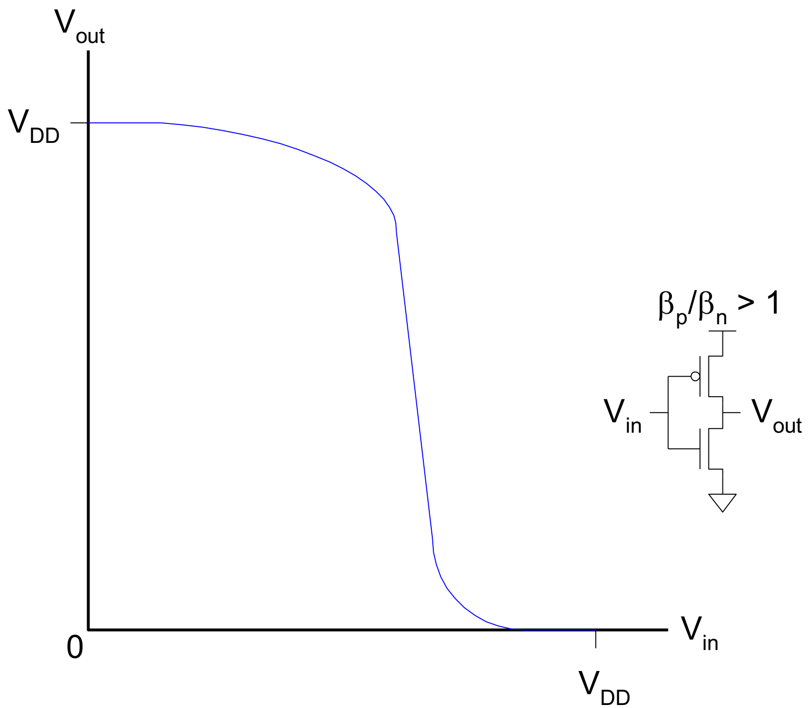
Noise Margins

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



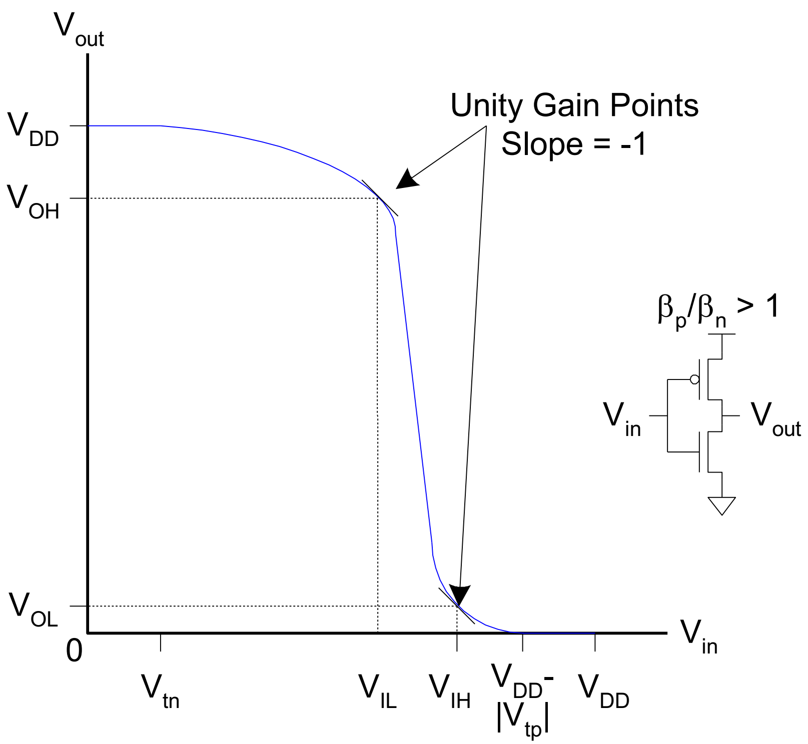
Logic Levels

- To maximize noise margins, select logic levels at



Logic Levels

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



For best noise margins:
$W_p/W_n=\mu_n/\mu_p$

For highest density:
$W_p/W_n=1$ – min size

Btw, for highest speed:
$W_p/W_n=(\mu_n/\mu_p)^{1/2}$

(The last formulae is a preview of what's to come: it will be derived in lecture 10, slide 18;
An alternative proof is also available in K. Martin, Digital Integrated Circuits Design, p. 208)

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

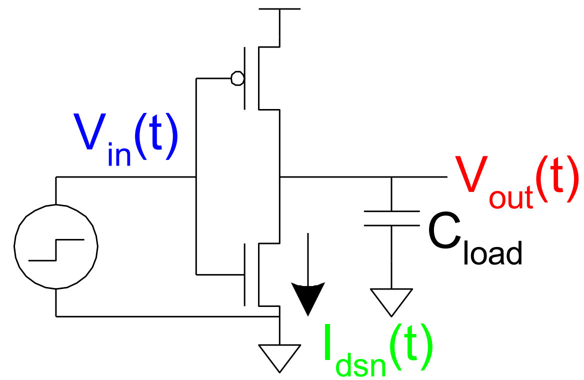
Inverter Step Response

- Ex: find step response of inverter driving load cap

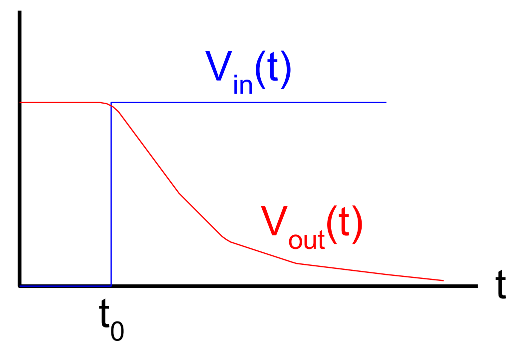
$$V_{in}(t) = u(t - t_0)V_{DD}$$

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

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



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



Delay Definitions

Propagation delay: maximum time

- **t_{pdr}**: *rising propagation delay*
  - From input to rising output, both crossing $V_{DD}/2$
- **t_{pdf}**: *falling propagation delay*
  - From input to falling output, both 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}

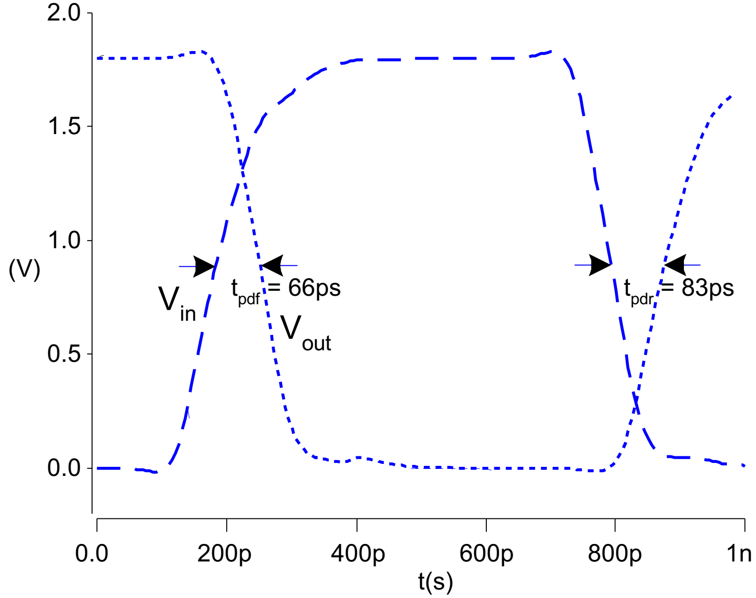
Delay Definitions

Contamination delay: minimum time

- $\mathbf{t_{cdr}}$: *rising contamination delay*
  - From input to rising output, both crossing $V_{DD}/2$
- $\mathbf{t_{cdf}}$: *falling contamination delay*
  - From input to falling output, both crossing $V_{DD}/2$
- $\mathbf{t_{cd}}$: *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 up/set up

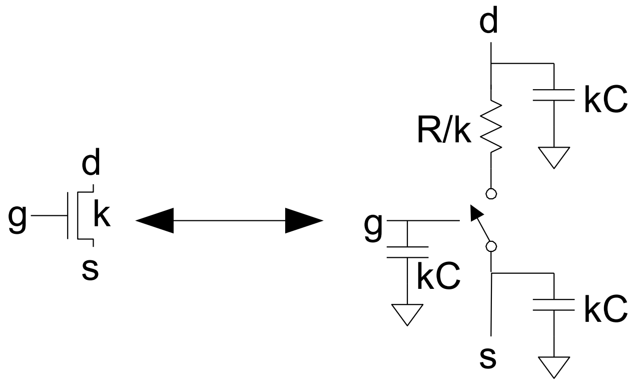


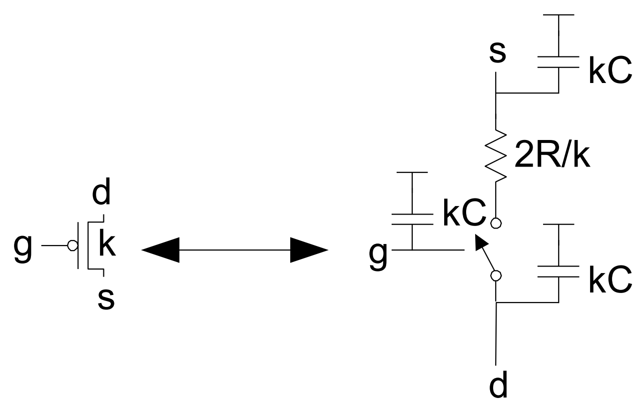
Delay Estimation

- We would like to be able to easily estimate delay
  - Not as accurate as simulation
- The step response usually looks like a 1st order RC response with an asymptotically settling 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, cap C on each node
  - Unit pMOS has resistance 2R, cap C on each node
- Capacitance - proportional to width
- Resistance - inversely proportional to width



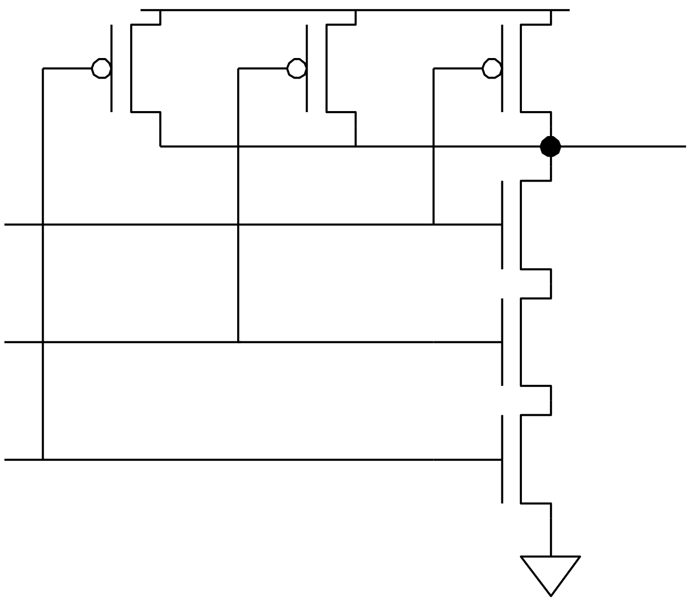


Example: 3-input NAND

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

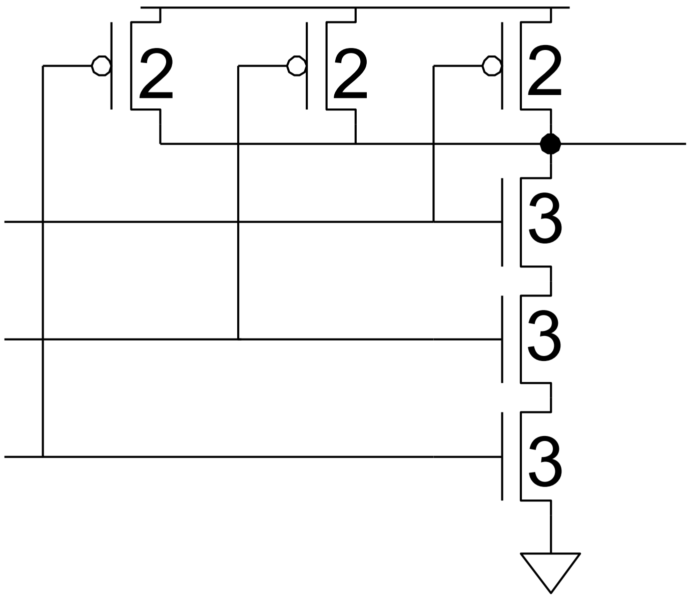
Example: 3-input NAND

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



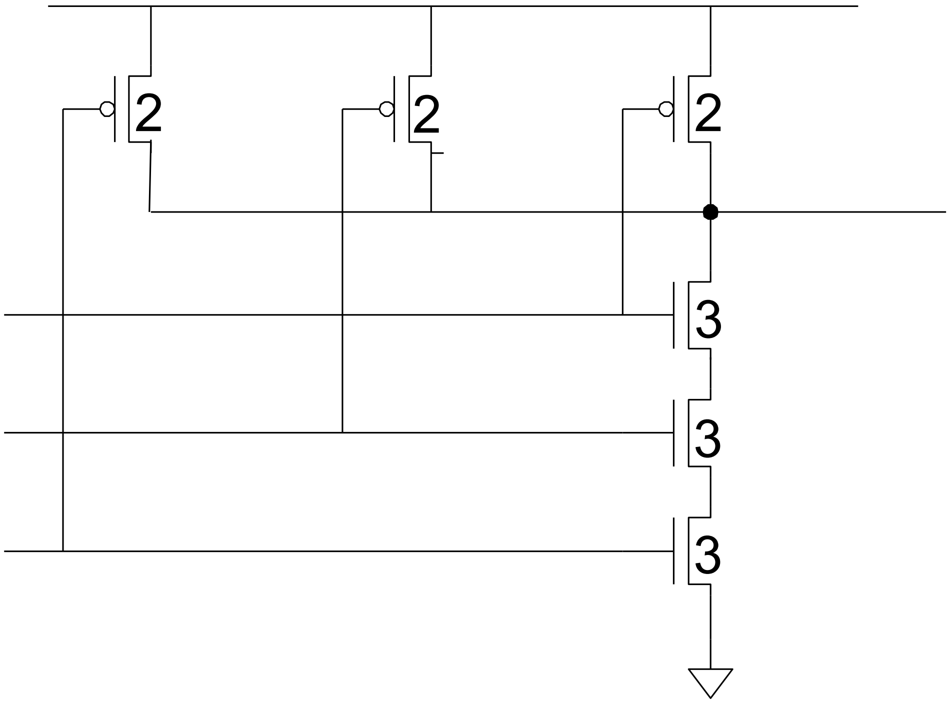
Example: 3-input NAND

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



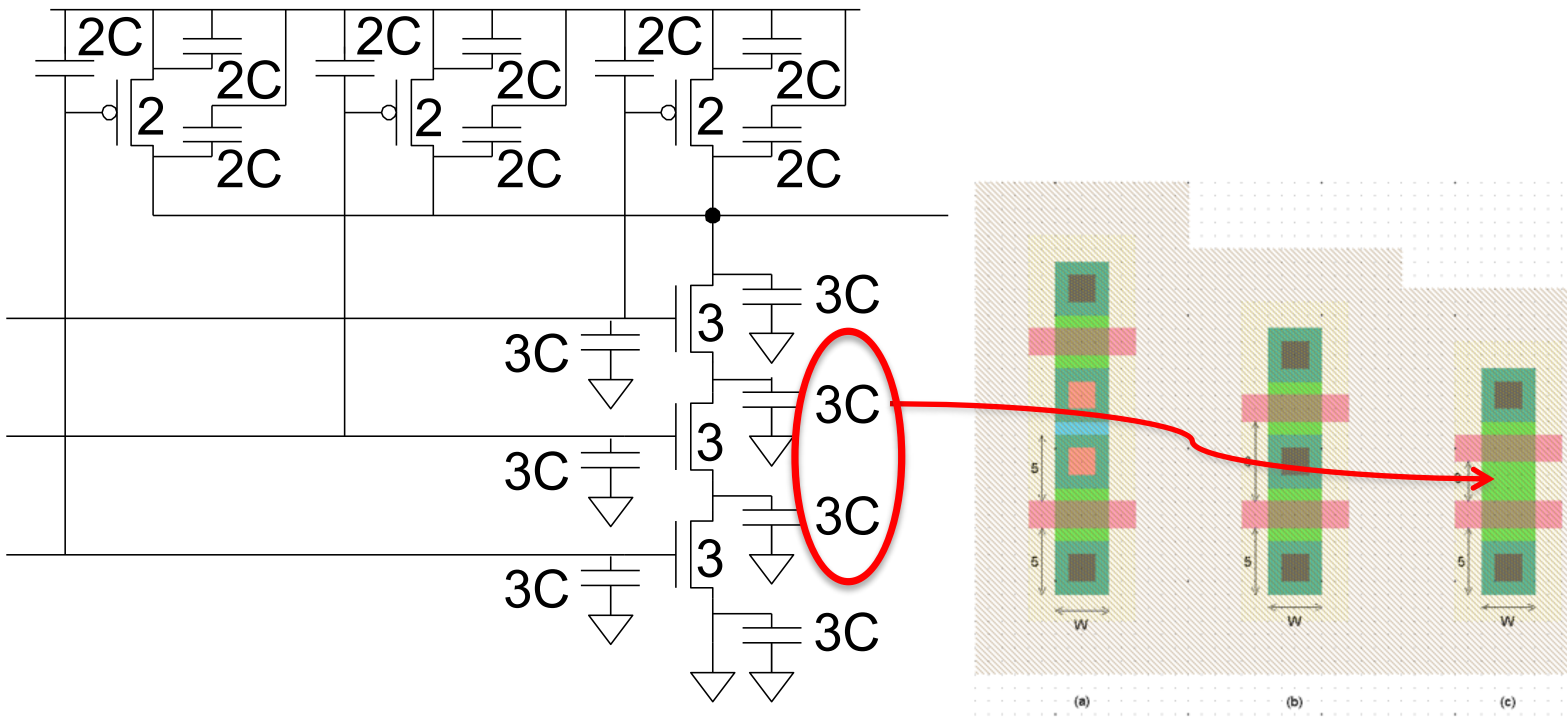
3-input NAND Caps

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



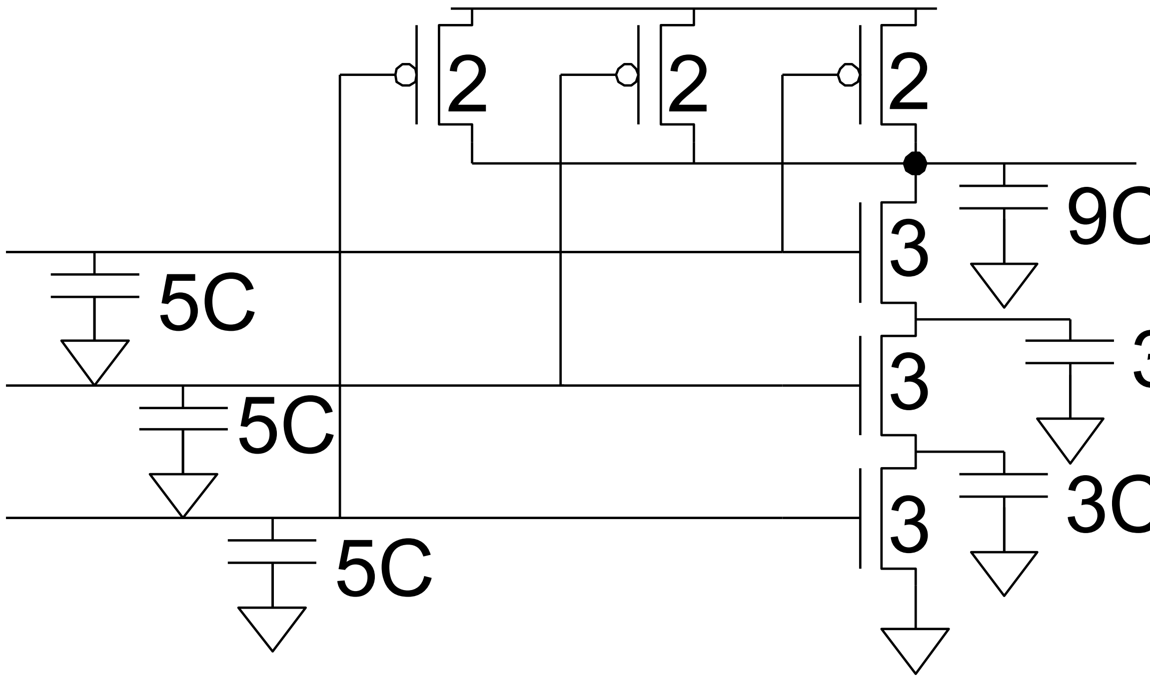
3-input NAND Caps

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



3-input NAND Caps

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

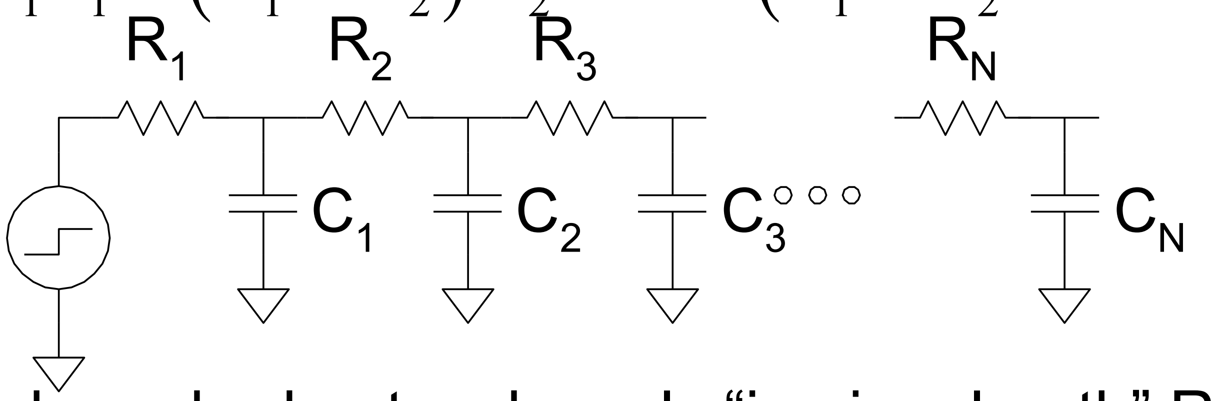


Elmore Delay

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

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

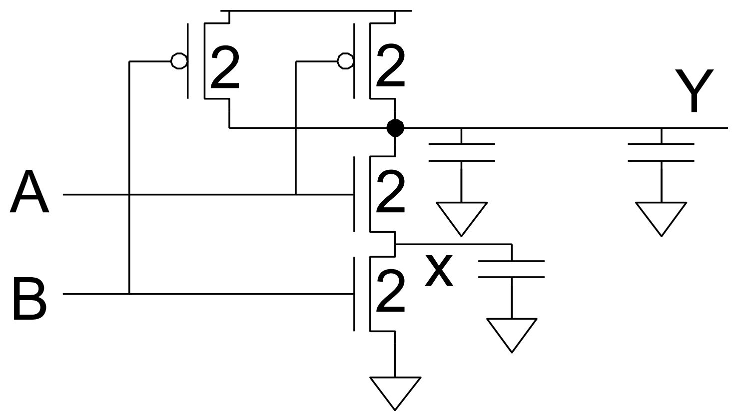
$$= R_1 C_1 + \left(R_1 + R_2\right) C_2 + \ldots + \left(R_1 + R_2 + \ldots + R_N\right) C_N$$

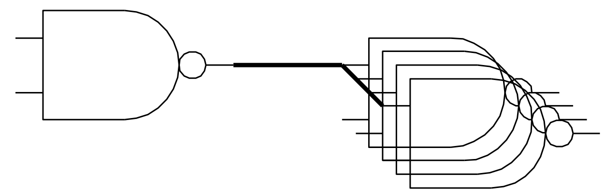


- For branched networks only "in-signal-path" R counts

Example: 2-input NAND

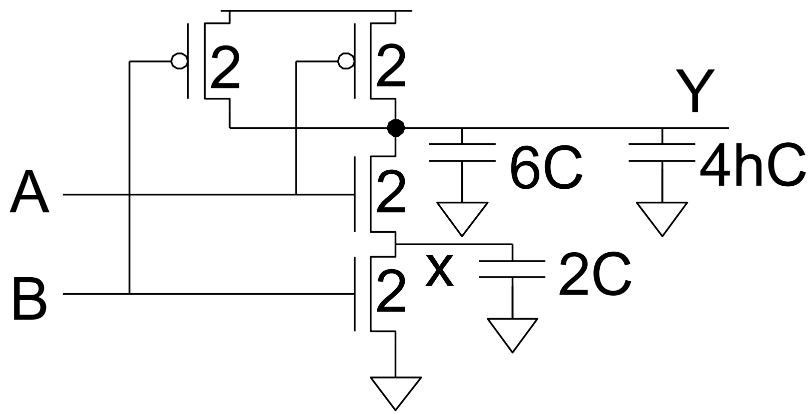
- Estimate rising and falling worst-case (propagation) delays of a 2-input NAND driving *h* identical gates.

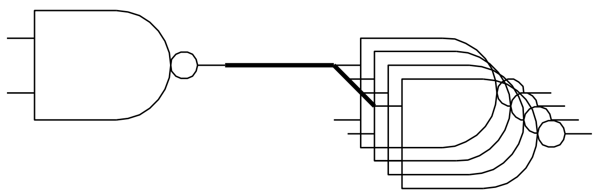




Example: 2-input NAND

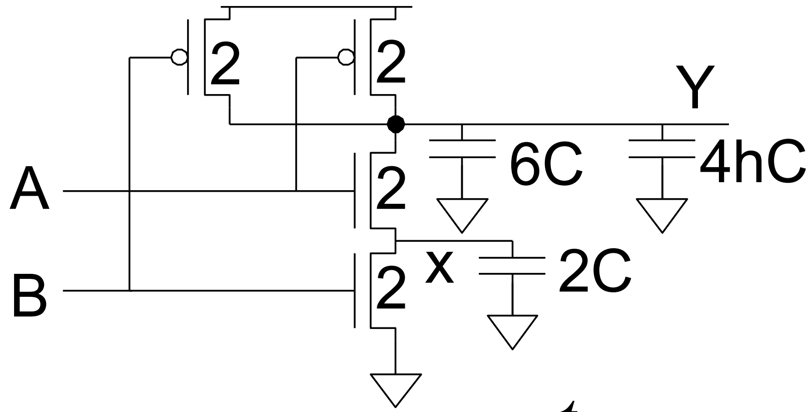
- Estimate rising and falling worst-case (propagation) delays of a 2-input NAND driving *h* identical gates.

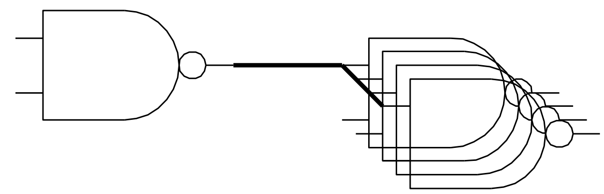




Example: 2-input NAND

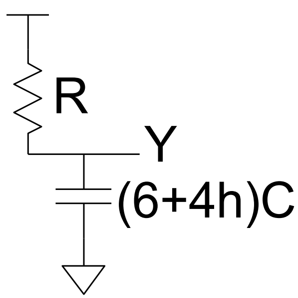
- Estimate rising and falling worst-case (propagation) delays of a 2-input NAND driving *h* identical gates.





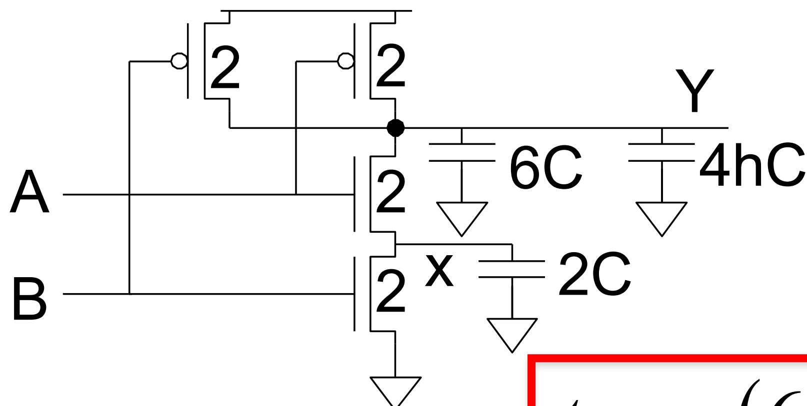
h copies

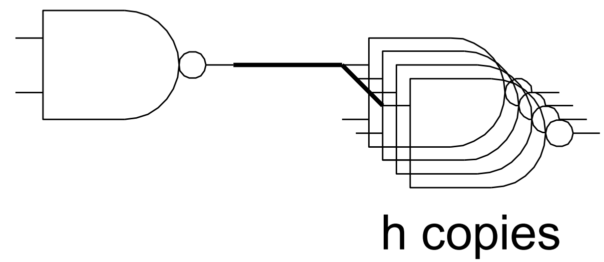
$$t_{pdr} =$$



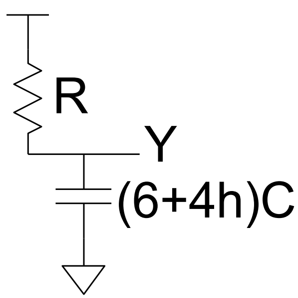
Example: 2-input NAND

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





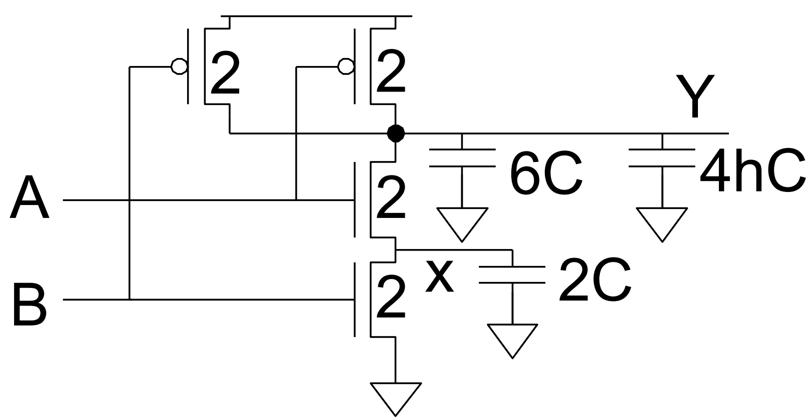
$$\boxed{t_{pdr} = \left(6+4h\right)RC}$$

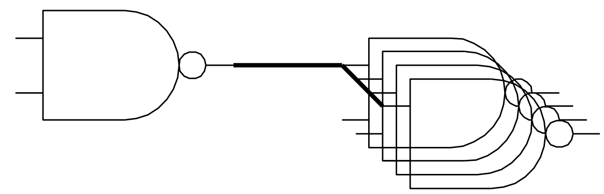


If B switches off first, node X contributes an extra 2RC delay (2C*R – resistance of in-path pMOS) which we ignore as critical signals should connect to node A

Example: 2-input NAND

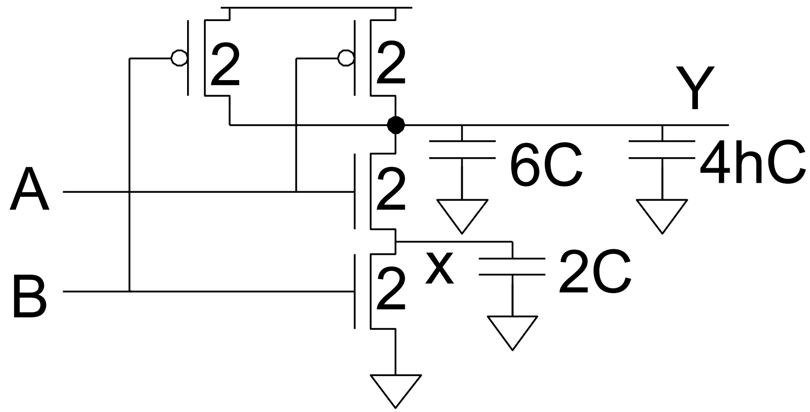
- Estimate rising and falling propagation delays of a 2-input NAND driving *h* identical gates.

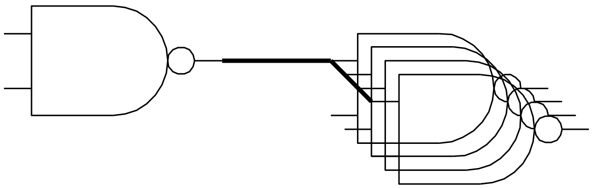


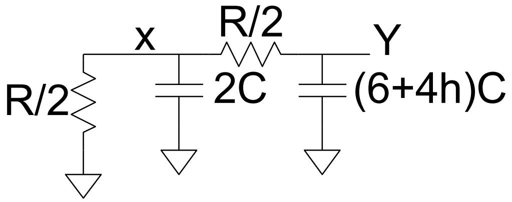


Example: 2-input NAND

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



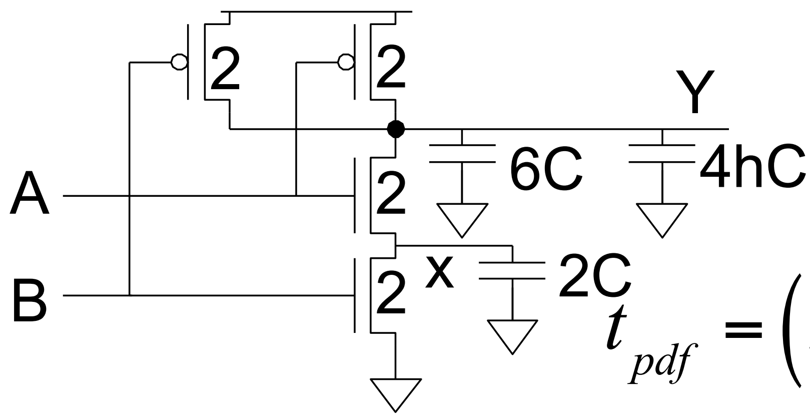


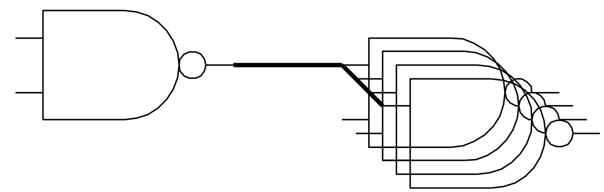


$$t_{pdf} =$$

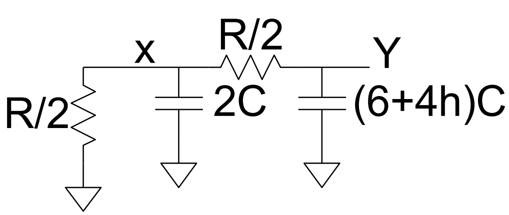
Example: 2-input NAND

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





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



If A arrives last, node X is already discharged and we ignore it:

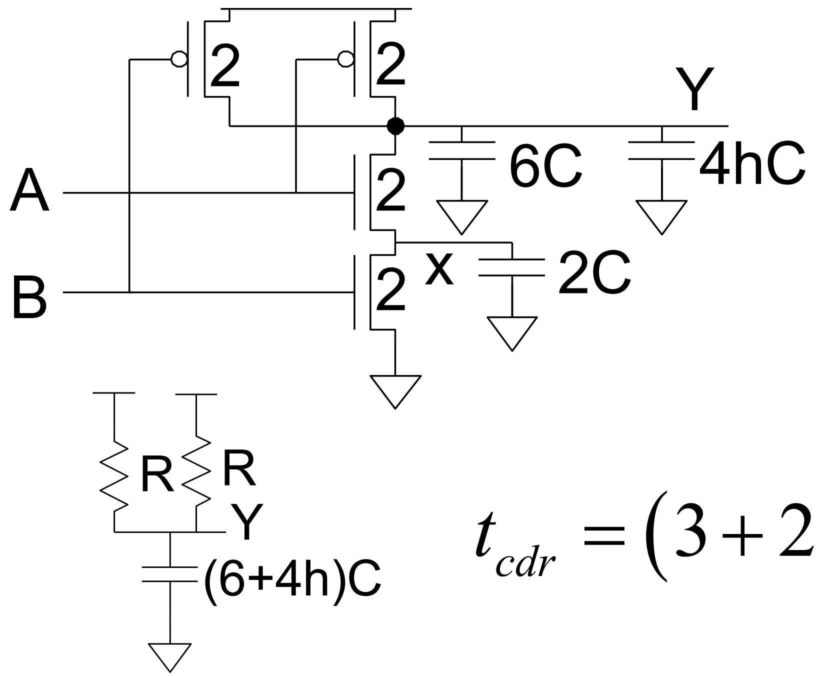
$$\boxed{t_{pdf} = \left(6+4h\right)RC}$$

Delay Components

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

Contamination Delay

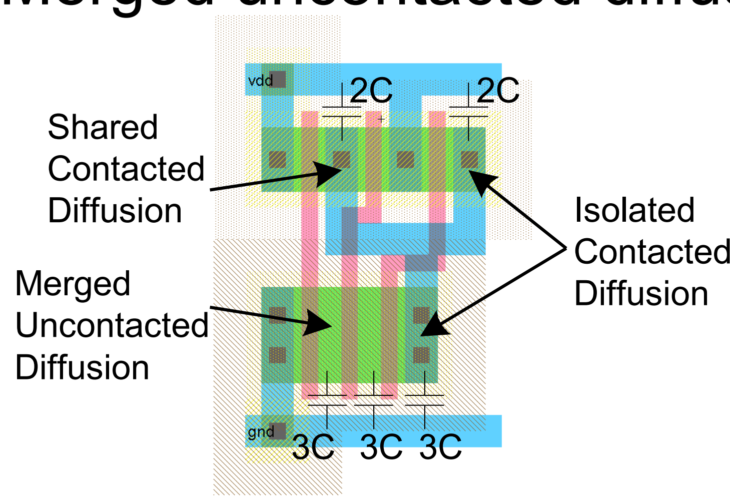
- Best-case (contamination) delay can be substantially less than propagation delay.
- Ex: If both inputs fall simultaneously

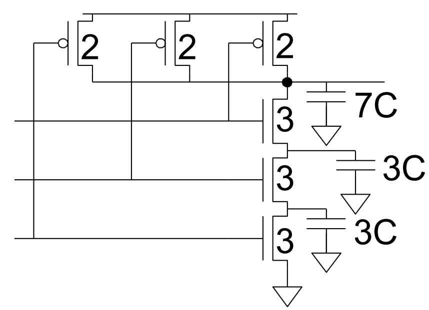


$$t_{cdr} = \left(3 + 2h\right) RC$$

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





(7C, not 9C when assuming contacted s / d)

Layout Comparison

- Which layout is better?