Lecture 2
CMOS Circuits

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese566/

Quantitative CMOS Model

- Threshold Voltage

$$V_T = V_{T0} + \gamma\left(\sqrt{|-2\phi_F| + V_{SB}} - \sqrt{|-2\phi_F|}\right)$$

$$\phi_0 = \frac{kT}{q}\ln\frac{N_A N_D}{n_i^2}$$

- I-V Curve
  - linear/triode $I_D = \frac{\mu_n C_{ox}}{2}\frac{W}{L}\left(V_{gs} - V_{th}\right)V_{ds} - \frac{V_{ds}^2}{2}$
  - saturation $I_D = \frac{\mu_n C_{ox}}{2}\frac{W}{L}\left(V_{gs} - V_{th}\right)^2\left(1 + \lambda V_{ds}\right)$

- Parasitic Capacitance
  - gate cap
  - junction cap $C_{j0} = \sqrt{\frac{\varepsilon_{si} q}{2}\left(\frac{N_A N_D}{N_A + N_D}\right)}\frac{1}{\sqrt{\phi_0}}$

CMOS Capacitance

- Overlap (gate) Cap
  - C_{gso}
- Junction Cap
  - C_{diff}

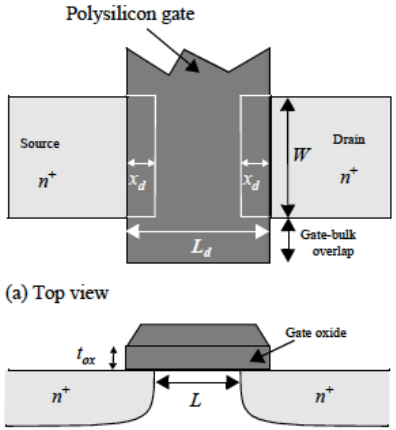


Figure 3.28 MOSFET overlap capacitance.

$$C_{diff} = C_{bottom} + C_{sw} = C_j \times AREA + C_{jsw} \times PERIMETER$$
$$= C_j L_S W + C_{jsw}(2L_S + W)$$

$$C_{GSO} = C_{GDO} = C_{ox} x_d W = C_o W \qquad (3.45)$$

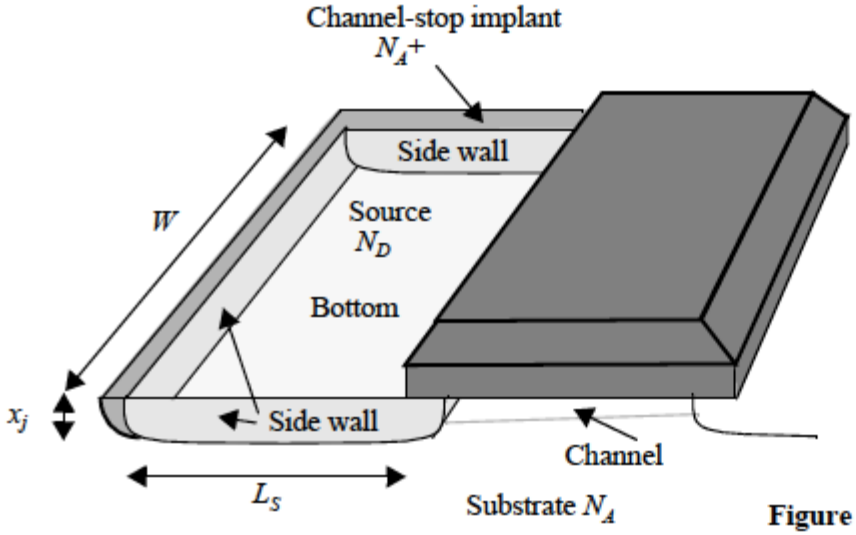


Figure 3.32 Detailed view of source junction.

Outline

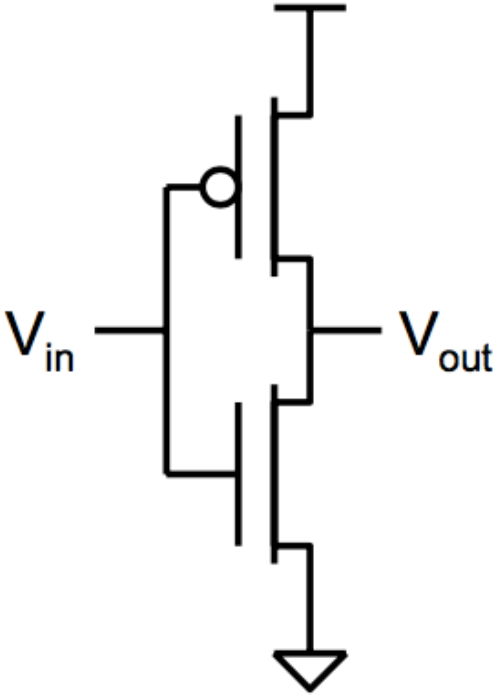
CMOS Inverter

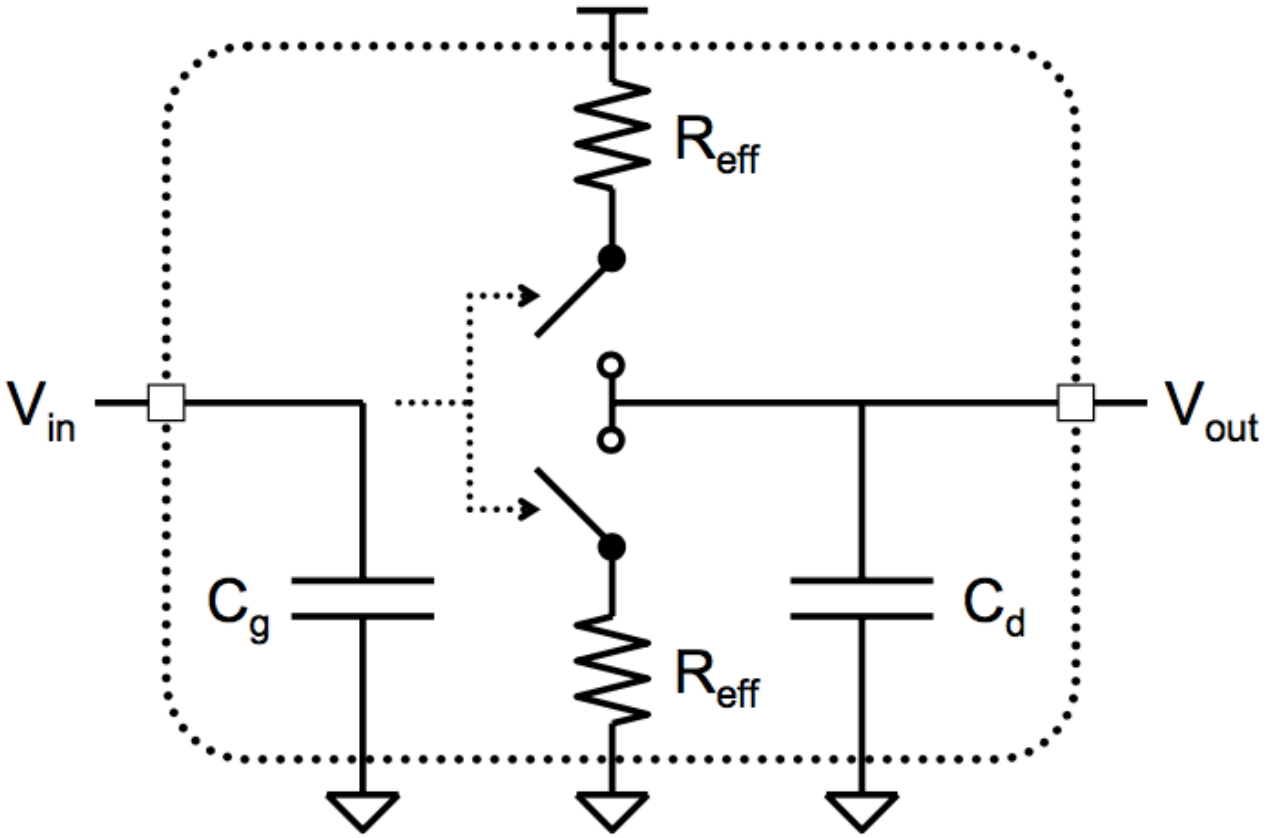
Combinational Logic

Sequential Circuits

Memory

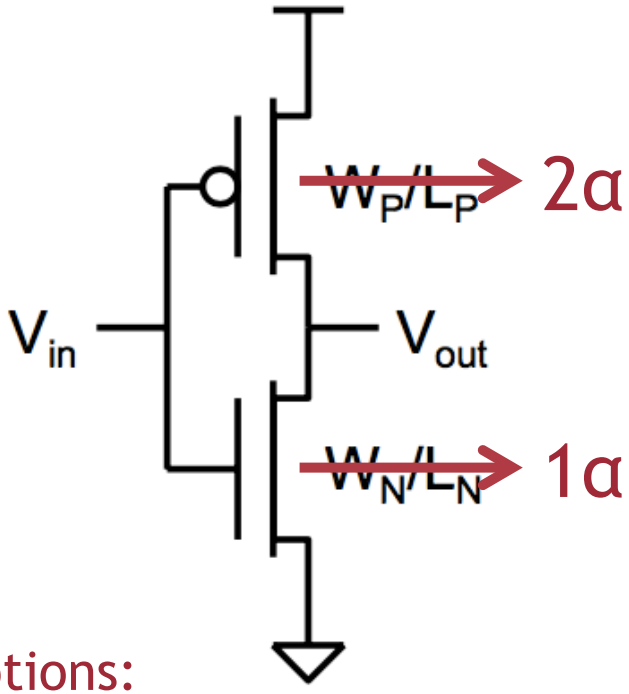
Inverter: Simple RC Model

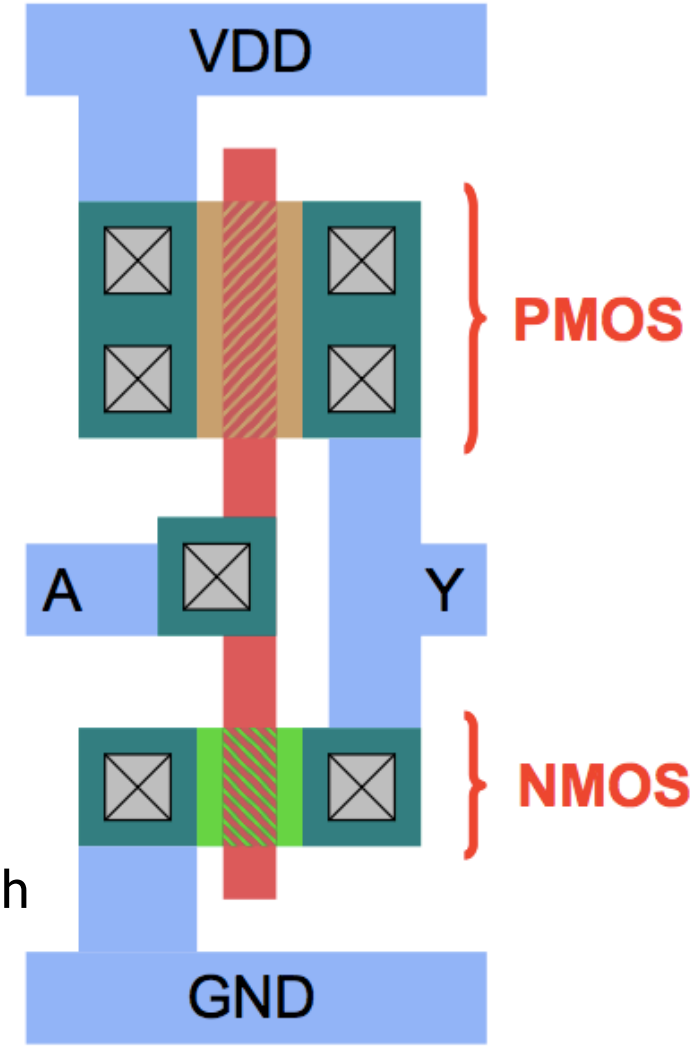




$$R_{eff} = R_{eff,N} = R_{eff,P}$$
$$C_g = C_{g,N} + C_{g,P}$$
$$C_d = C_{d,N} + C_{d,P}$$

Inverter Layout

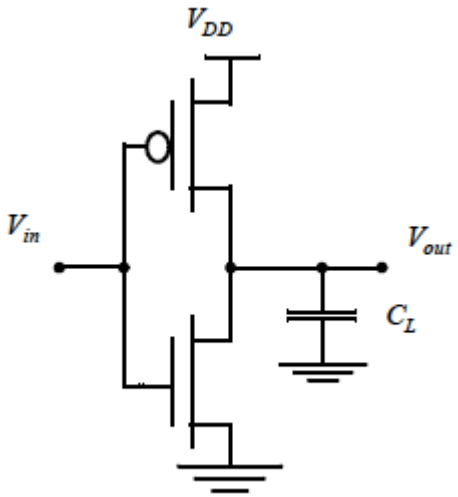




Assumptions:

1. All transistors are minimum length
2. All gate have equal rise/fall time (PMOS twice as wide)
3. Normalize transistor width to minimum NMOS

Inverter Voltage Transfer Characteristics (VTC)



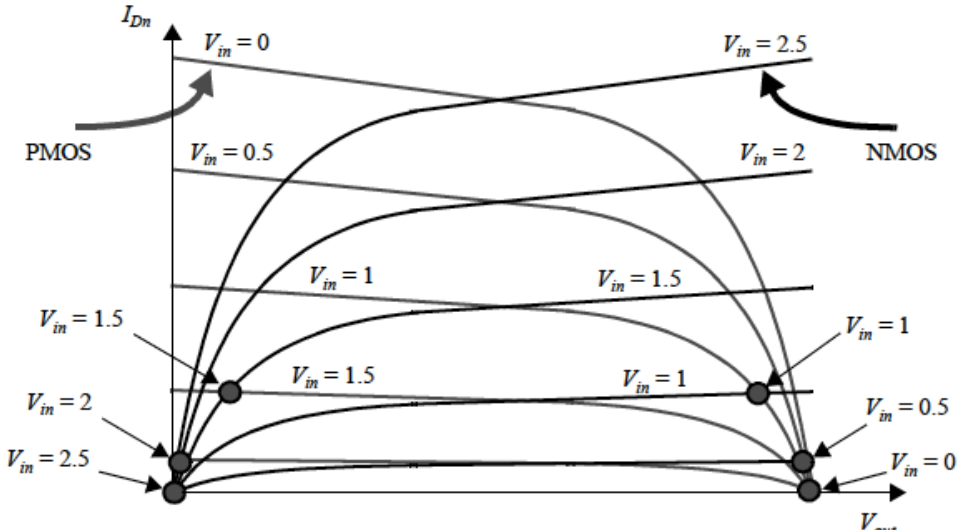


Figure 5.4 Load curves for NMOS and PMOS transistors of the static CMOS inverter (V_{DD} = 2.5 V). The dots represent the dc operation points for various input voltages.

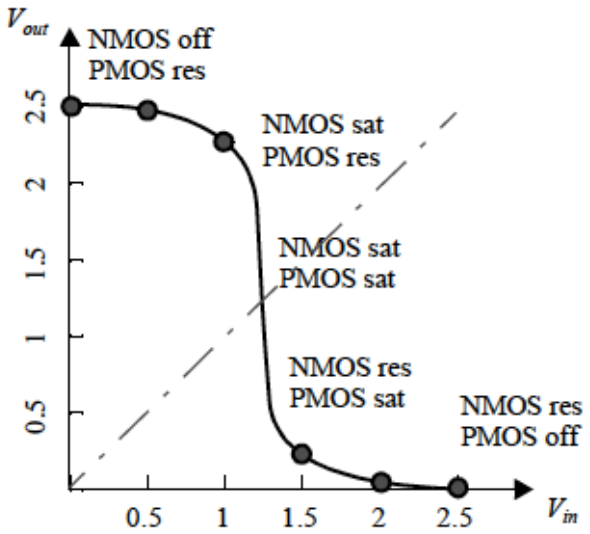
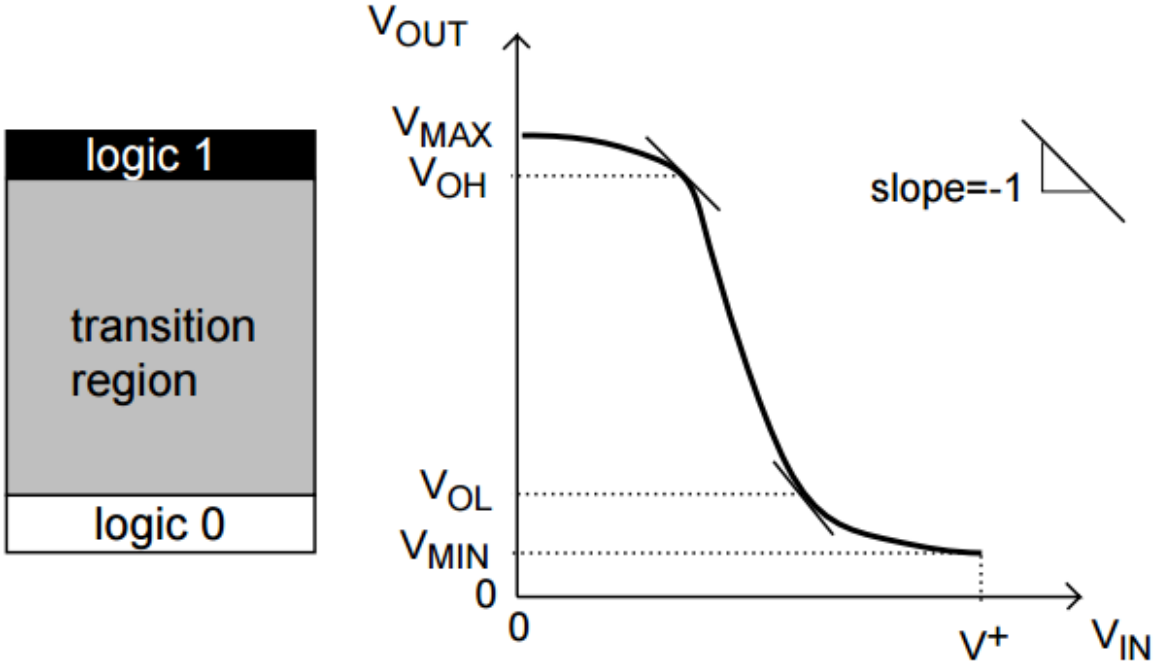


Figure 5.5 VTC of static CMOS inverter, derived from Figure 5.4 (V_{DD} = 2.5 V). For each operation region, the modes of the transistors are annotated — off, res(istive), or sat(urated).

Inverter Noise Margin

- V_{IH}, V_{IL}, V_{OH}, V_{OL} Definition
  - where the dV_{out}/dV_{in} slope is -1

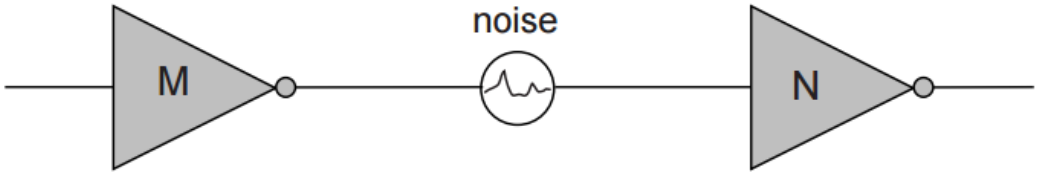


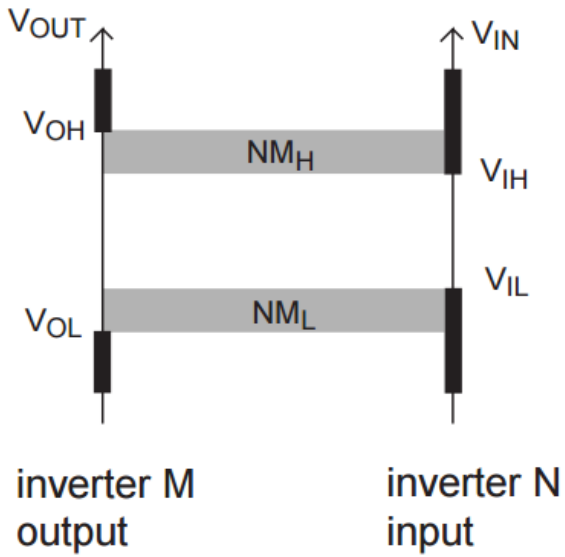
- Noise Margin
  - if range of output V_{OL} to V_{OH} is wider than the range of input values V_{IL} to V_{IH}, then the inverter exhibits noise immunity. (|Voltage Gain|>1)

Inverter Noise Margin

- NMH, NML Definition

$$NM_H \equiv V_{OH} - V_{IH}$$
$$NM_L \equiv V_{IL} - V_{OL}$$





Inverter Dynamic Power

- Capacitive Power

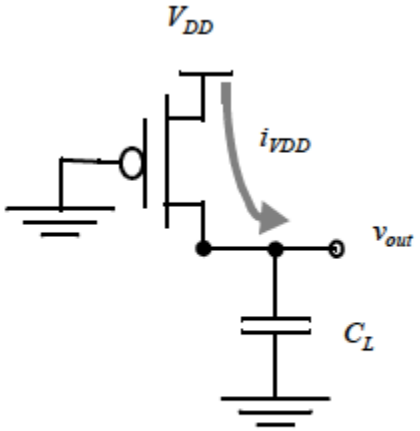


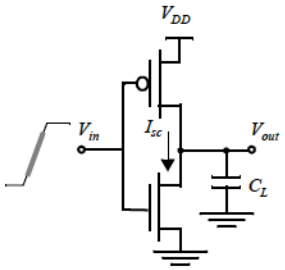
Figure 5.25 Equivalent circuit during the low-to-high transition.

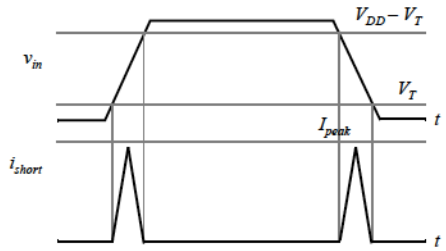
$$E_{VDD} = \int_0^\infty i_{VDD}(t)V_{DD}dt = V_{DD}\int_0^\infty C_L\frac{dv_{out}}{dt}dt = C_L V_{DD}\int_0^{V_{DD}} dv_{out} = C_L V_{DD}^2$$

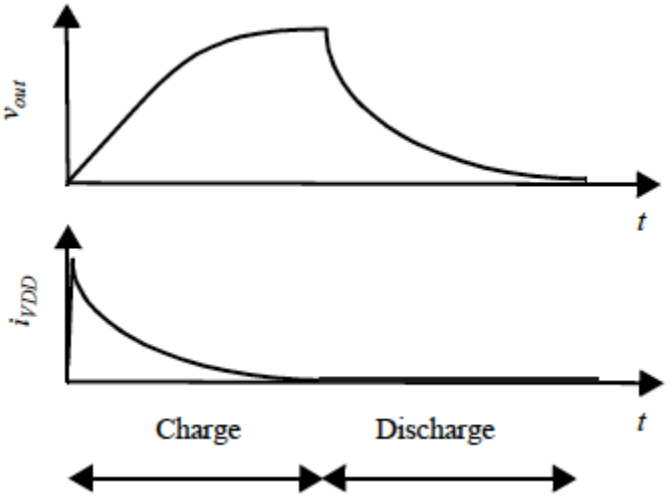
$$E_C = \int_0^\infty i_{VDD}(t)v_{out}dt = \int_0^\infty C_L\frac{dv_{out}}{dt}v_{out}dt = C_L\int_0^{V_{DD}} v_{out}dv_{out} = \frac{C_L V_{DD}^2}{2}$$

$$P_{dyn} = C_L V_{DD}^2 f_{0\to 1} = C_L V_{DD}^2 P_{0\to 1} f = C_{EFF} V_{DD}^2 f$$

- Short-circuit Power

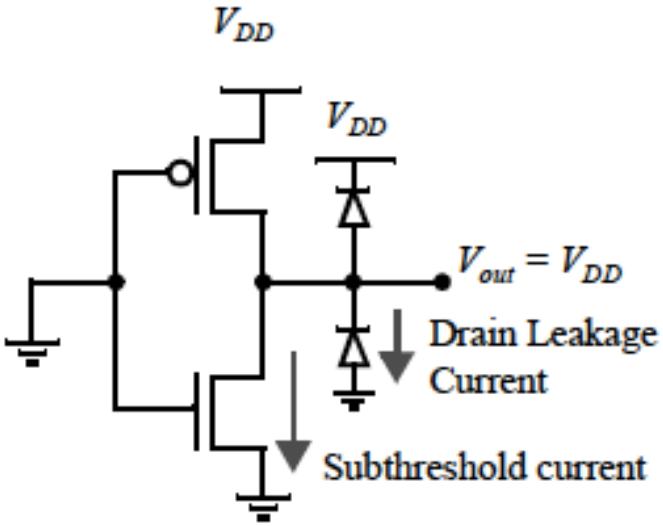






Inverter Static Power

- Sub-threshold Leakage



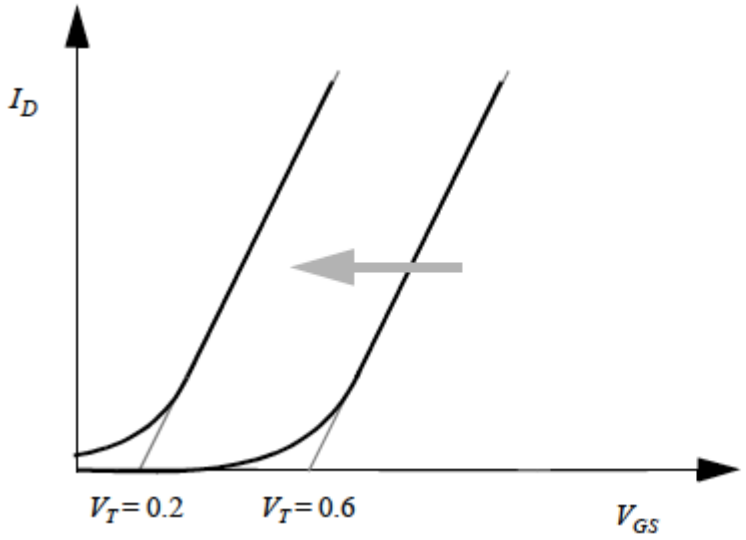
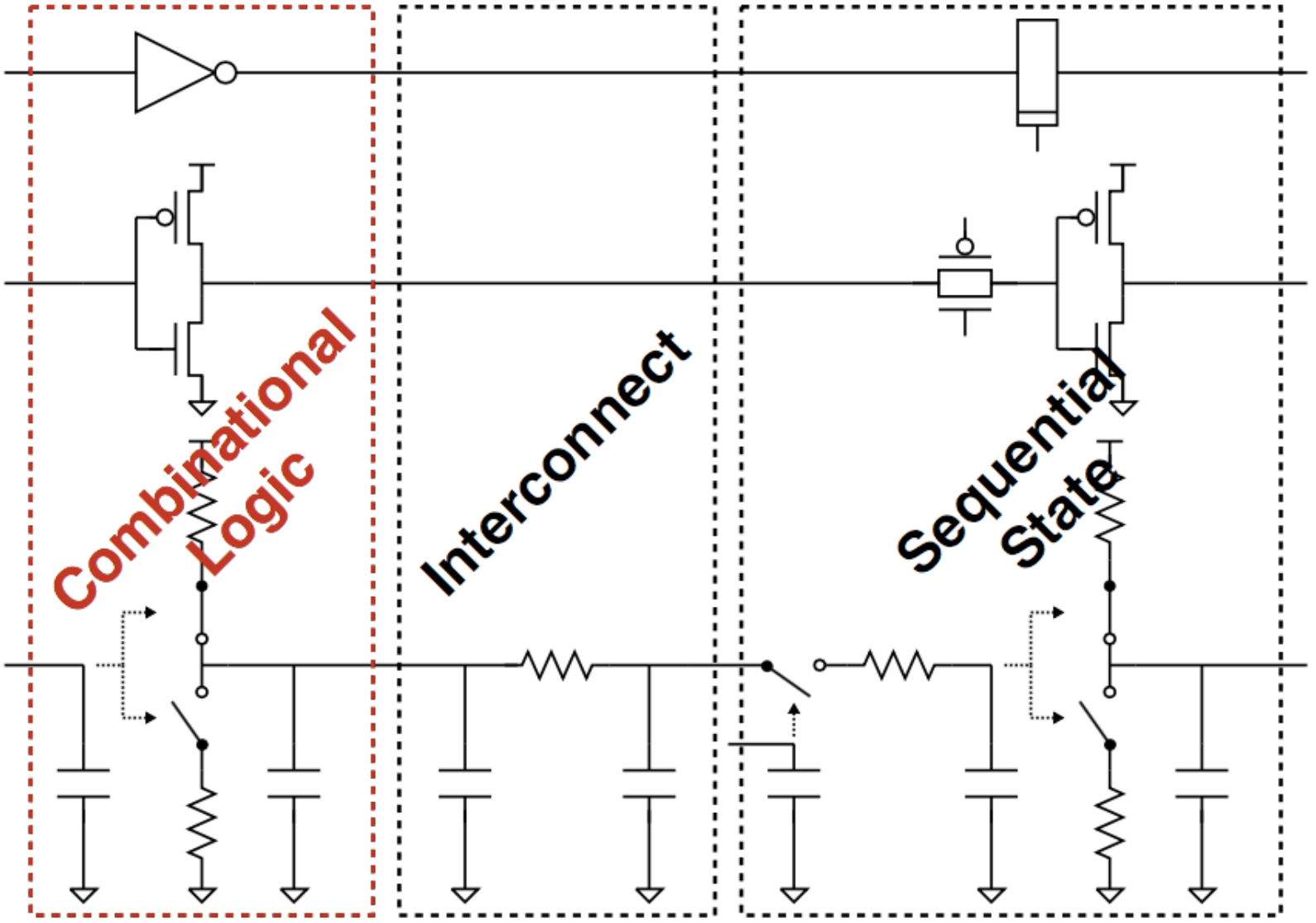
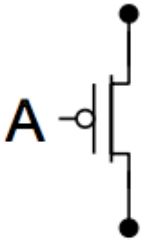


Figure 5.35 Decreasing the threshold increases the subthreshold current at $V_{GS} = 0$.

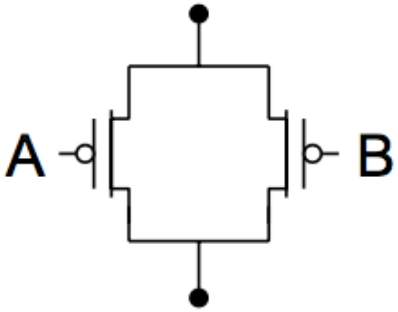
Outline



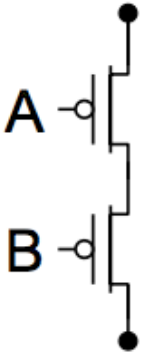
CMOS Duality



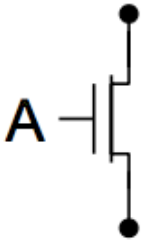
Conducts if A=0



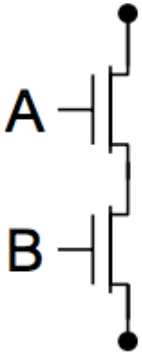
Conducts if A=0 **OR** B=0



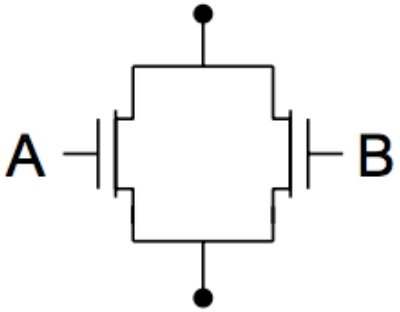
Conducts if A=0 **AND** B=0



Conducts if A=1



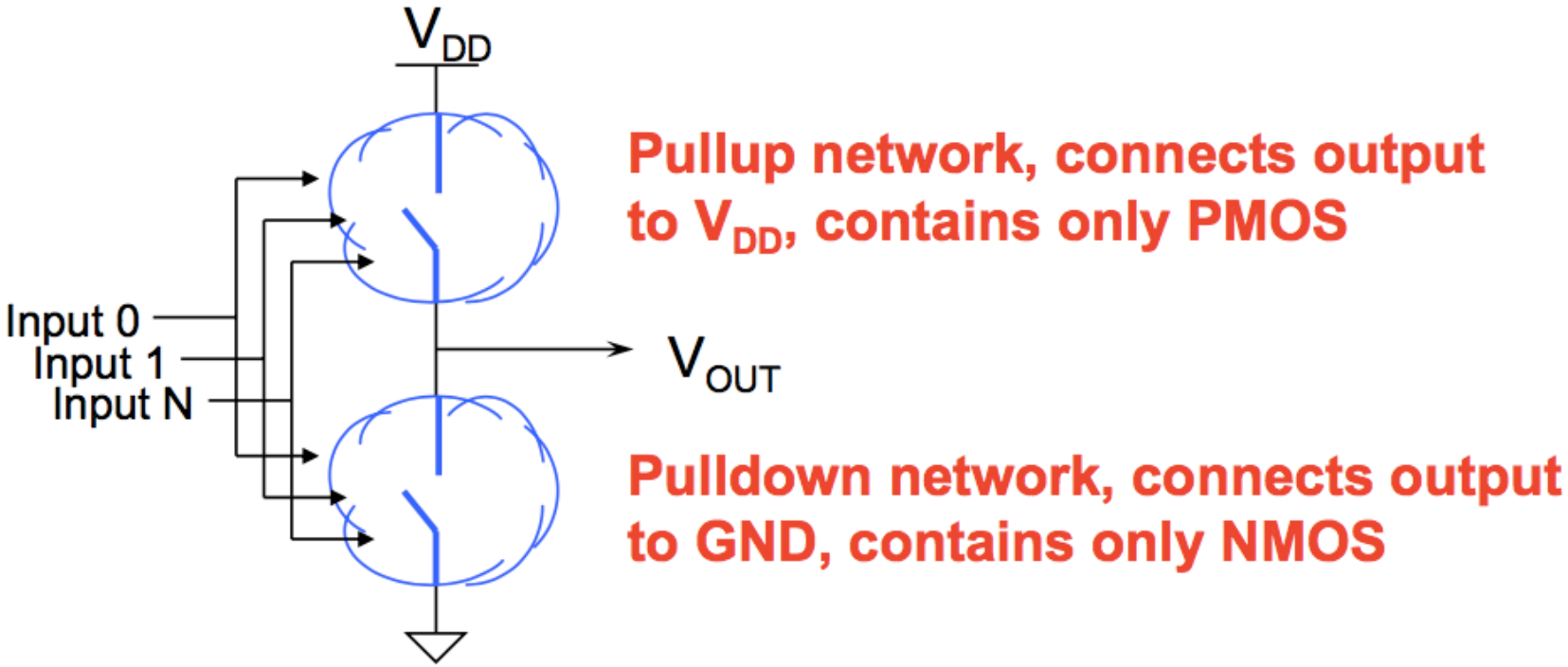
Conducts if A=1 **AND** B=1



Conducts if A=1 **OR** B=1

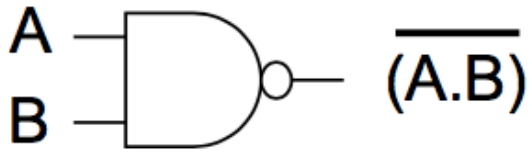
CMOS Static Logic Style

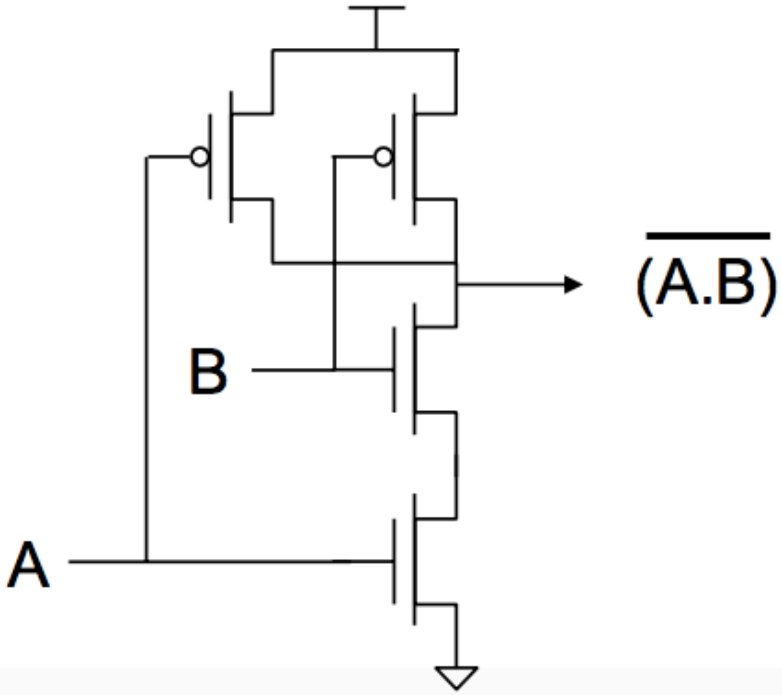
- For every set of input logic values, either pull-up or pull-down network connects to VDD or GND
  - power rails would be shorted if both connected
  - output would float (tri-state logic), if neither connected



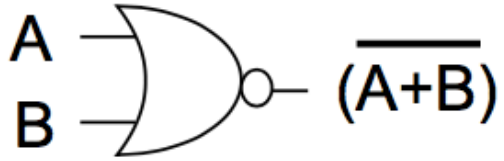
NAND/NOR Static Logic Gates

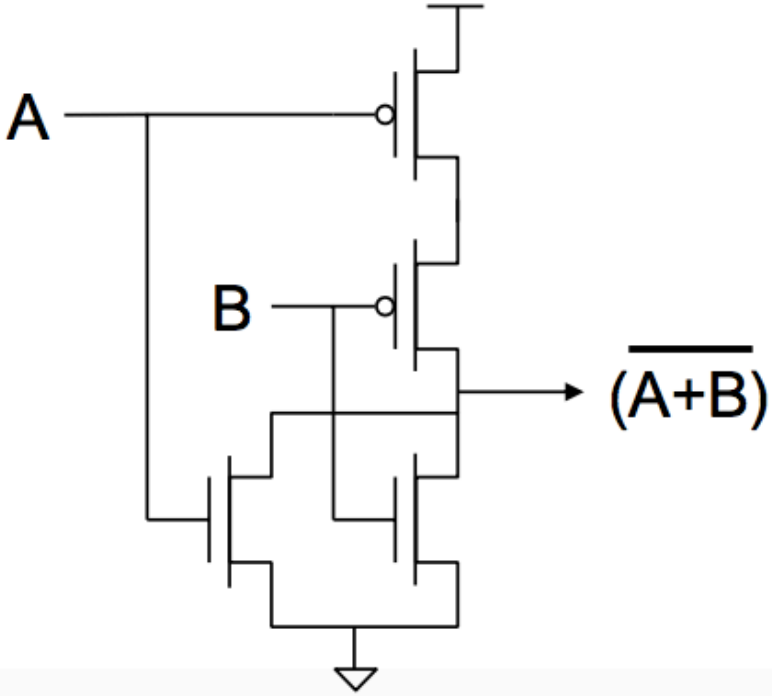
NAND Gate





NOR Gate

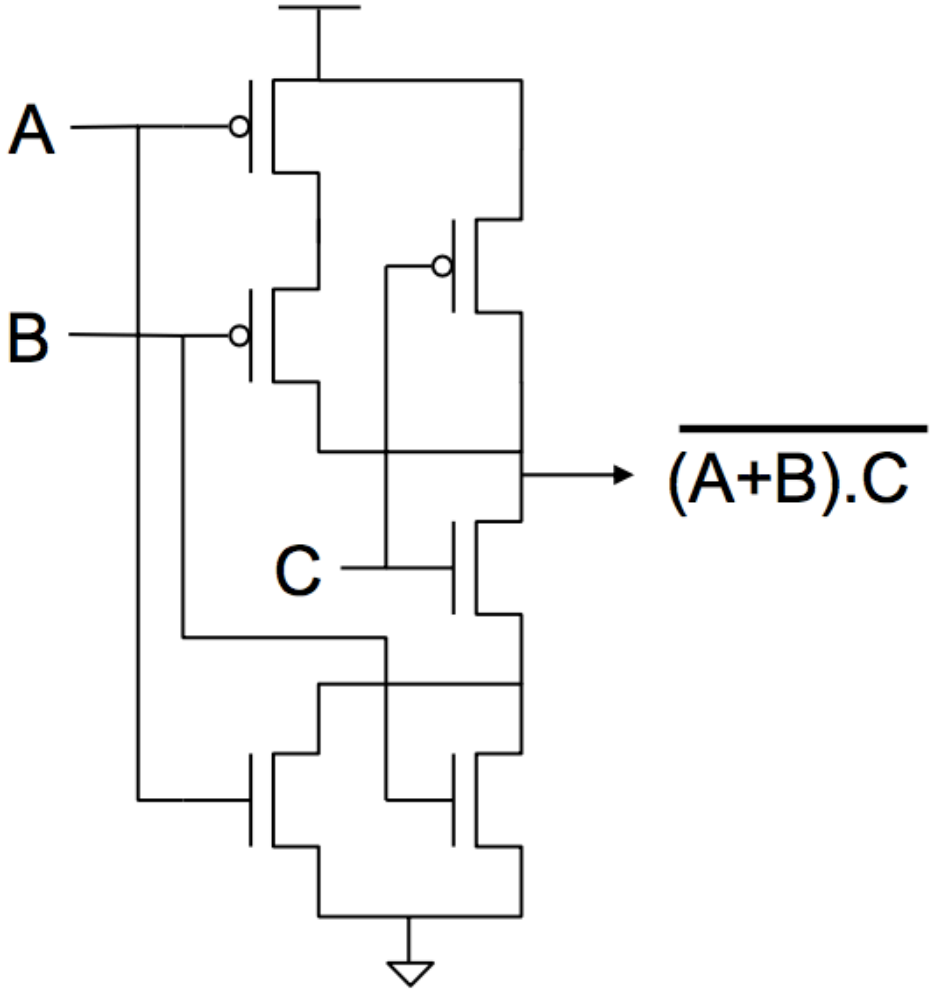




Design More Complex Gates

- Goal is to create a logic function $f(x_1, x_2, \ldots)$
  - can only implement inverting logic with one stage
- Implement pull-down network
  - write PD = $/f(x_1, x_2, \ldots)$
  - use parallel NMOS for OR of inputs
  - use series NMOS for AND of inputs
- Implement pull-up network
  - write PU = $f(x_1, x_2, \ldots) = g(/x_1, /x_2, \ldots)$
  - use parallel PMOS for OR of inverted inputs
  - use series PMOS for AND of inverted inputs

Complex Logic Gate Example



$$f = \overline{(A + B) \cdot C}$$

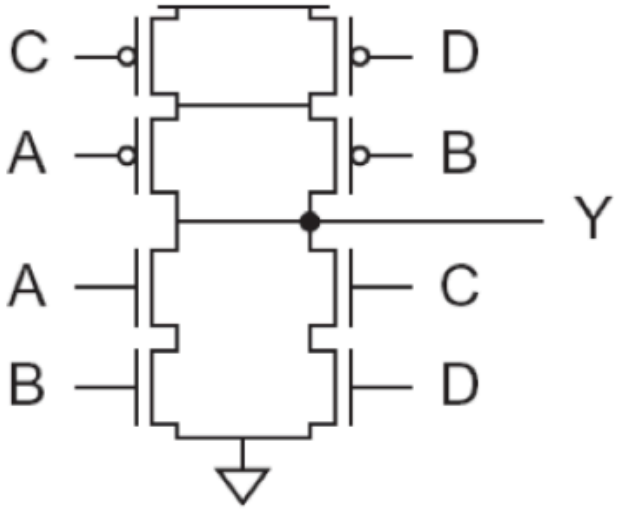
$$PD = (A + B) \cdot C$$

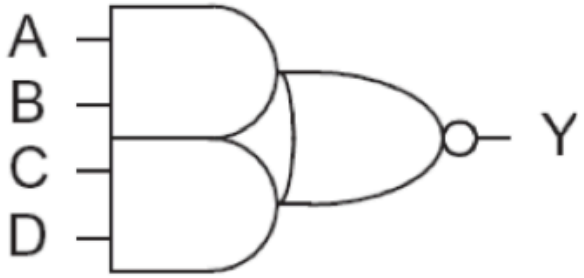
$$PU = \overline{(A + B) \cdot C}$$

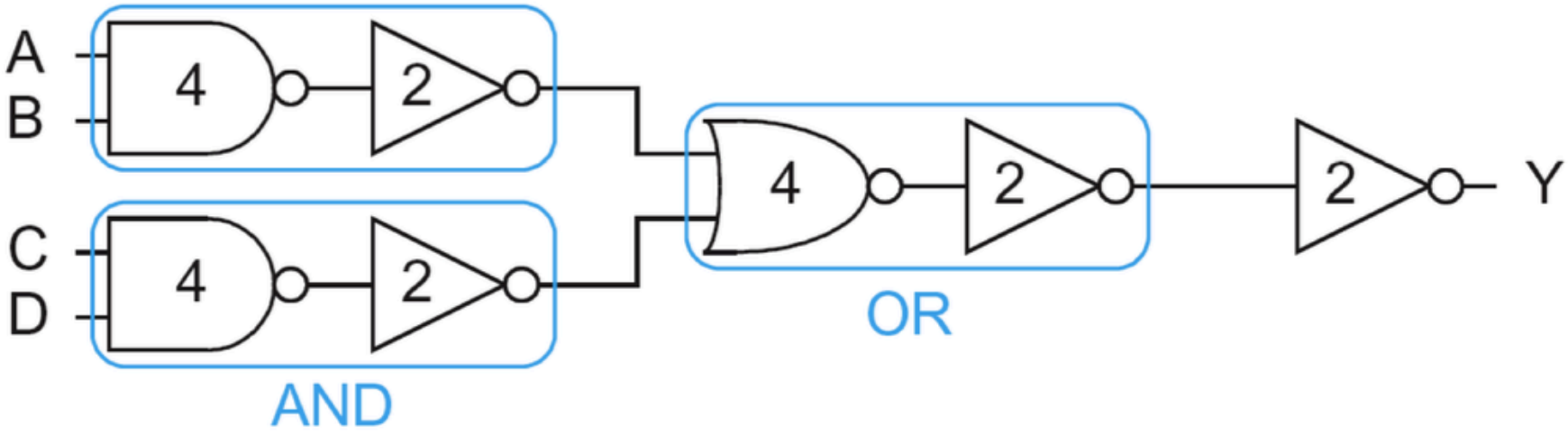
$$= \overline{(A + B)} + \overline{C}$$

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

Multi-Stage Static Logic

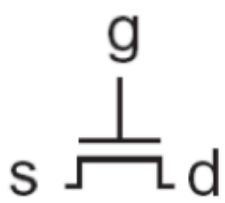


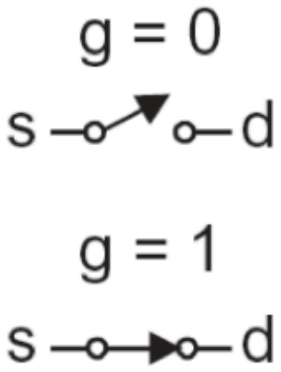


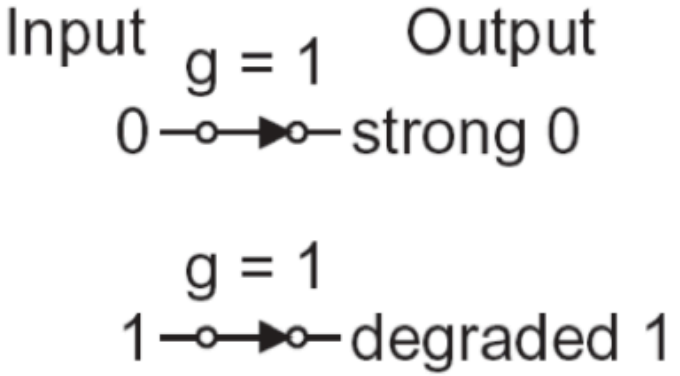


CMOS Pass-Gate Logic Style

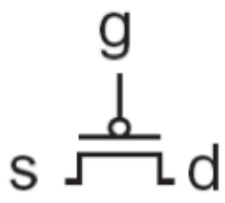
nMOS

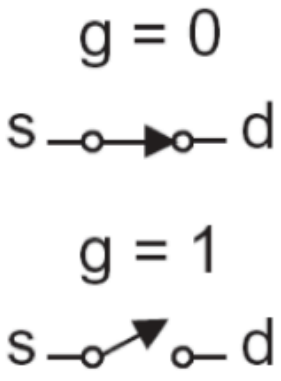


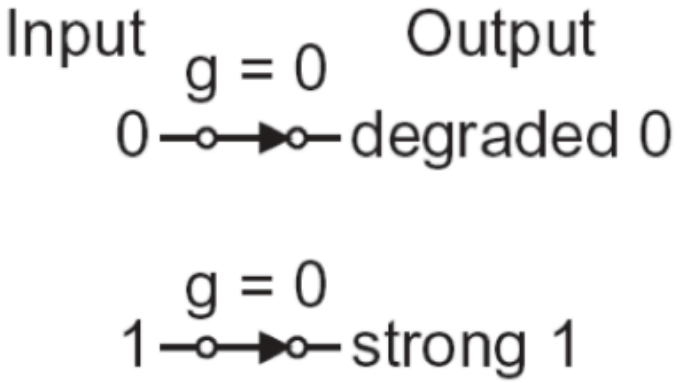




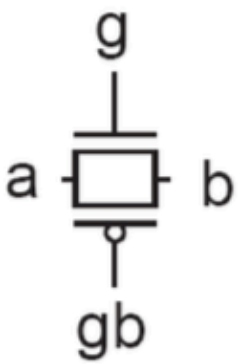
pMOS

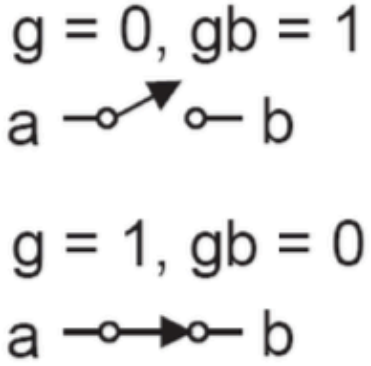


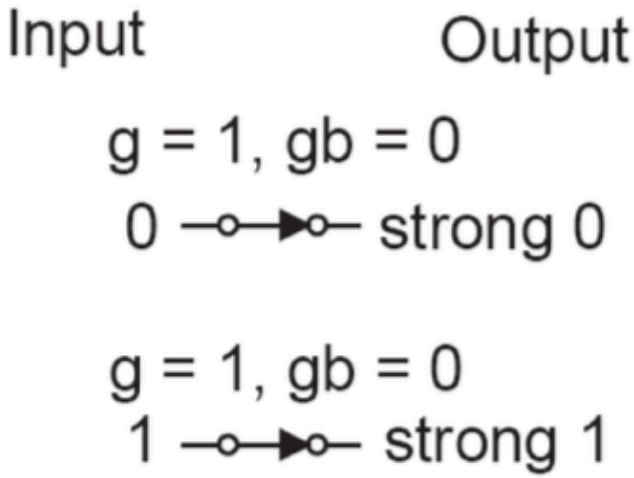


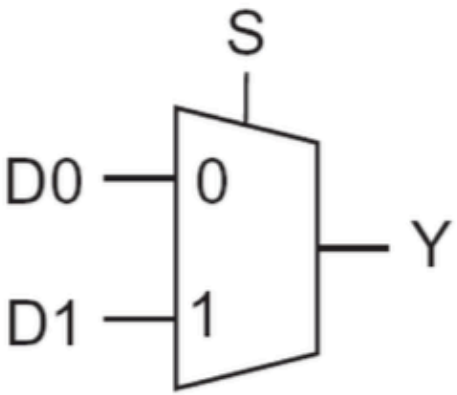


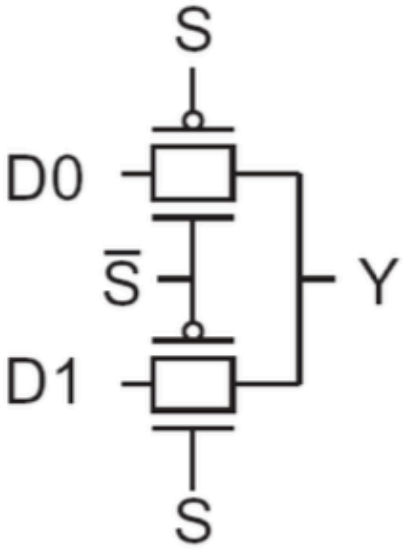
CMOS Transmission Gate Multiplexer



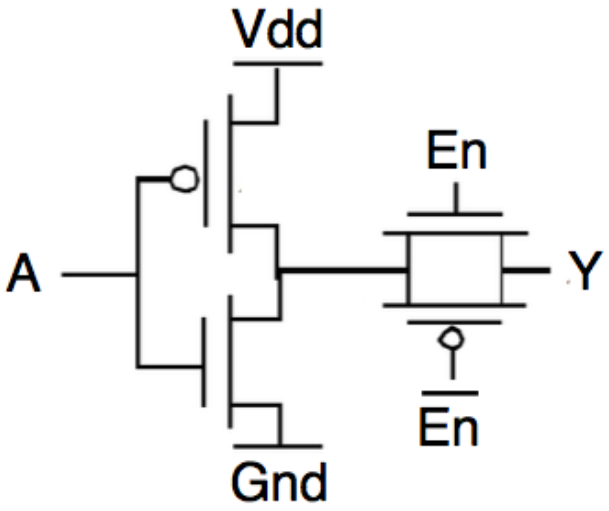


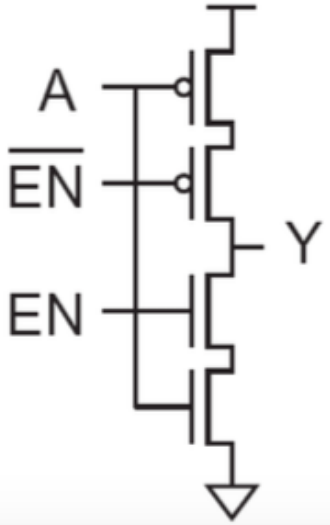


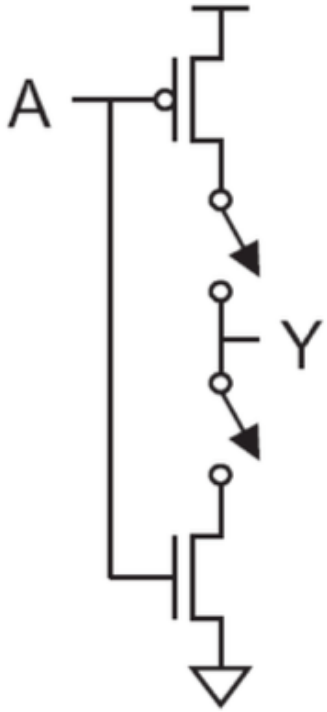




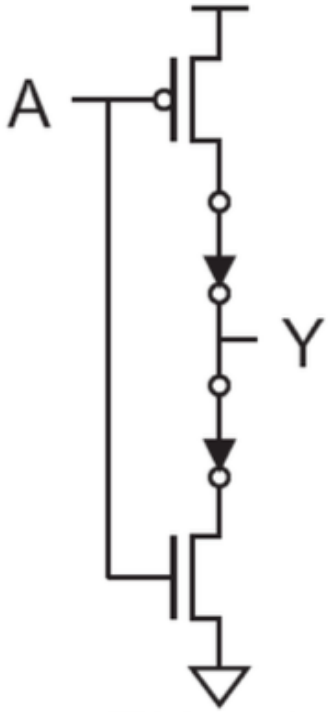
CMOS Tri-State Buffers



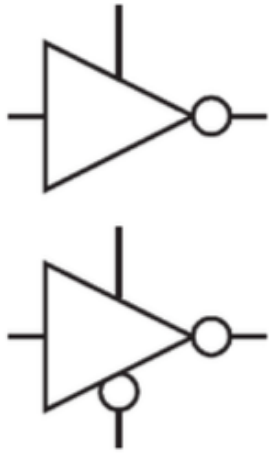




EN = 0
Y = 'Z'

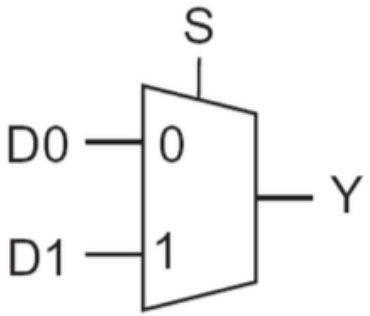


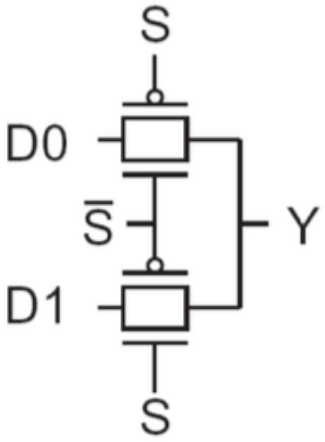
EN = 1
Y = $\overline{A}$

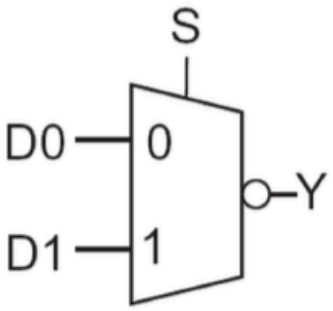


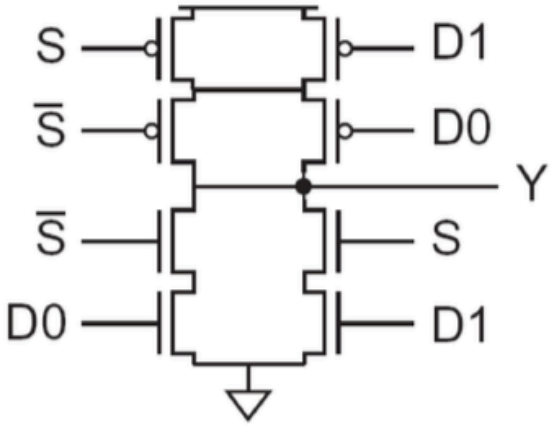
Various Multiplexer Implementations

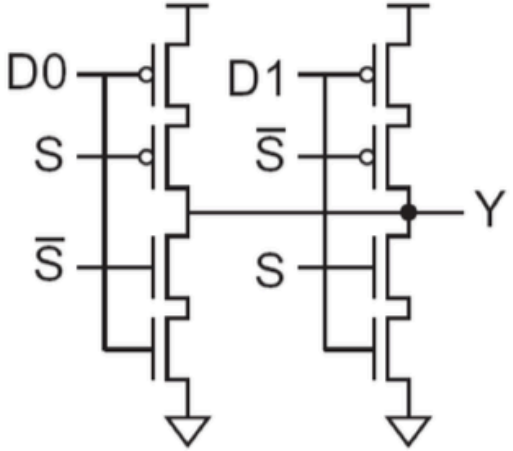
- Delay, Area, Energy Trade-offs
  - simple first-order analysis



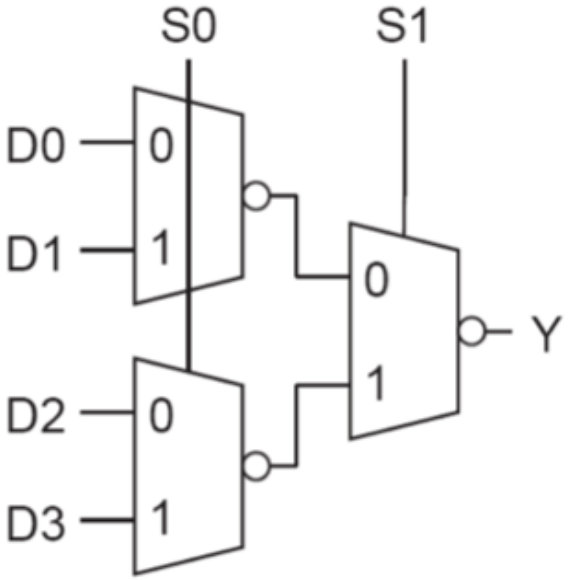


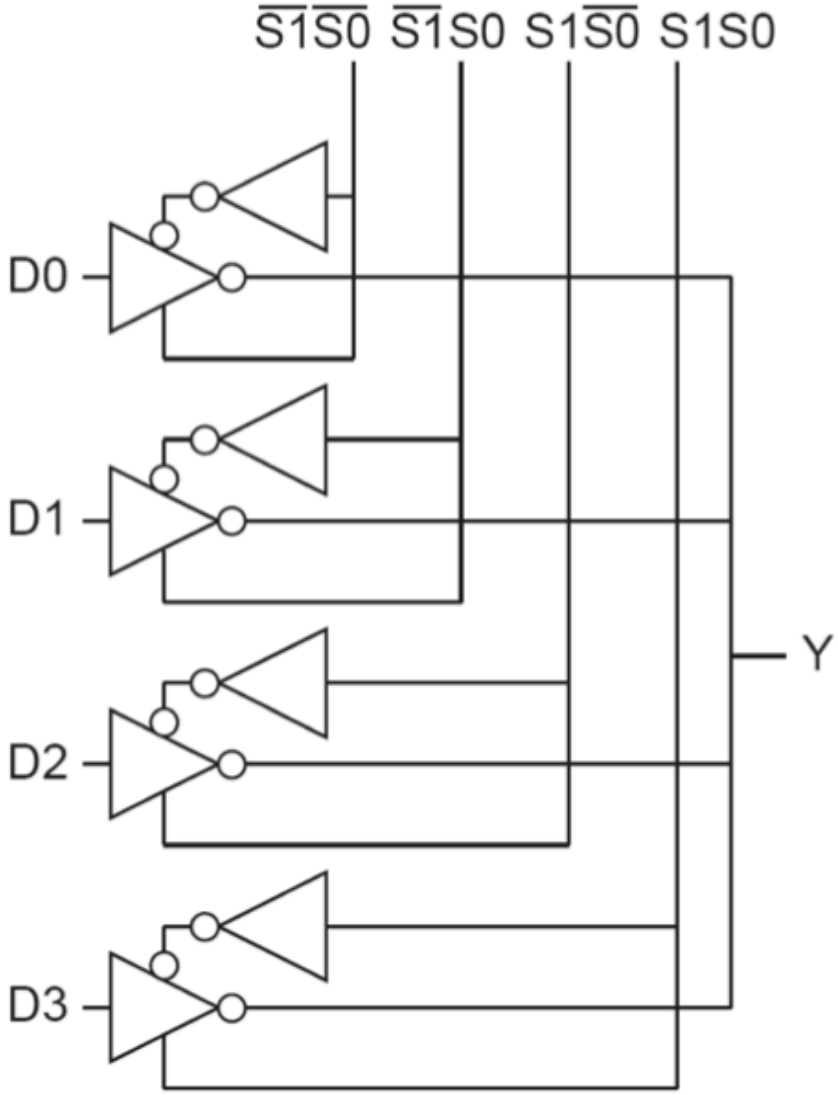






Larger Tri-State Multiplexers





Homework #2

- Posted on class website
- Due on 1/30 at 2:30pm
- Provide extension to the lecture

RC Delay

- Lumped Model
  - C only
  - RC model

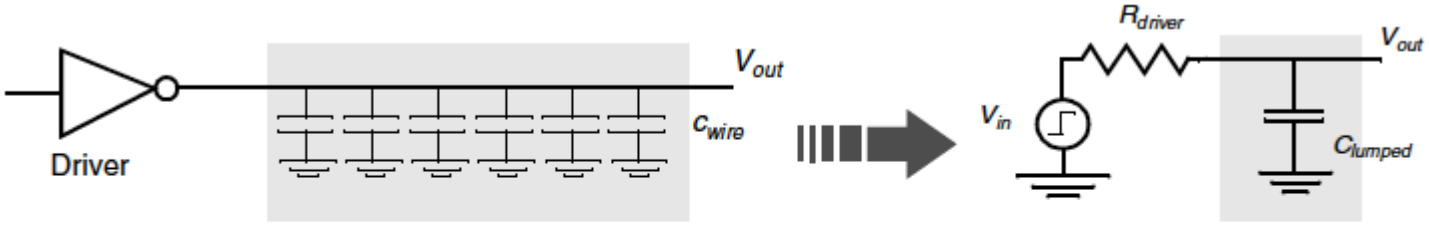
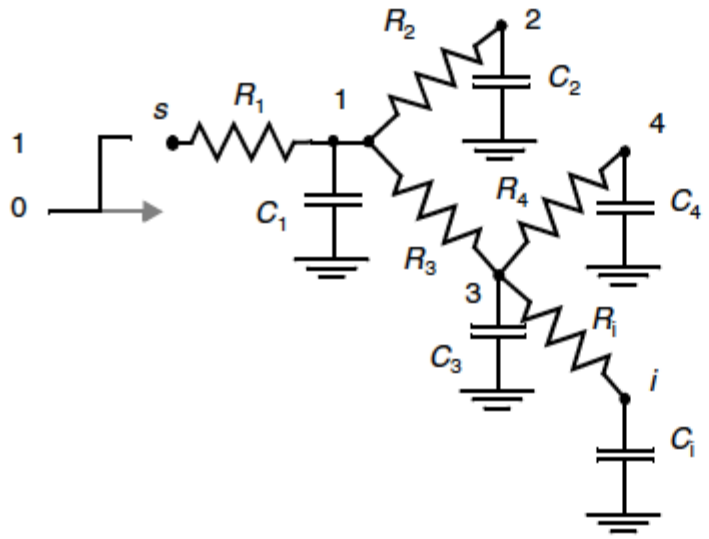


Figure 4.11 Distributed versus lumped capacitance model of wire. $C_{lumped} = L \times c_{wire}$, with L the length of the wire and c_{wire} the capacitance per unit length. The driver is modeled as a voltage source and a source resistance R_{driver}.

Elmore Delay Formula

- Assumptions regarding the RC network
  - the network has a single input node
  - all the capacitors are between a node and the ground
  - the network does not contain any resistive loops (tree)
- Unique resistive path
  - path resistance $R_{44} = R_1 + R_3 + R_4$
  - shared path resistance $R_{ik} = \sum R_j \Rightarrow (R_j \in [path(s \rightarrow i) \cap path(s \rightarrow k)])$



$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

Figure 4.12 Tree-structured *RC* network.

Questions?

Comments?

Discussion?

Acknowledgement

N. Weste and D. Harris, “CMOS VLSI Design”, 2011
Jan Rabaey, “Digital Integrated Circuits”, 2006
Cornell University, ECE 5745
UC Berkeley, CS 230
MIT, 6.371