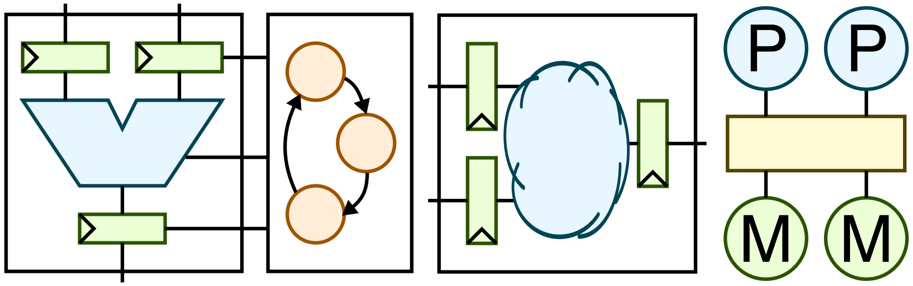
ECE 6745 Complex Digital ASIC Design
Topic 3: CMOS Circuits

Christopher Batten

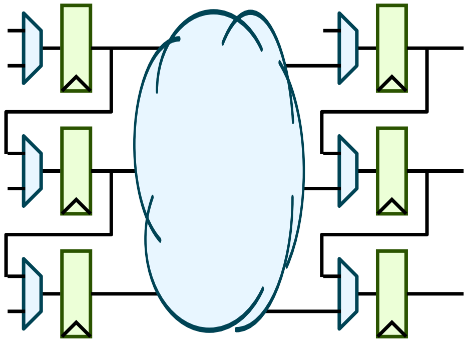
School of Electrical and Computer Engineering
Cornell University

http://www.csl.cornell.edu/courses/ece6745

Part 1: ASIC Design Overview



Topic 1
Hardware Description Languages

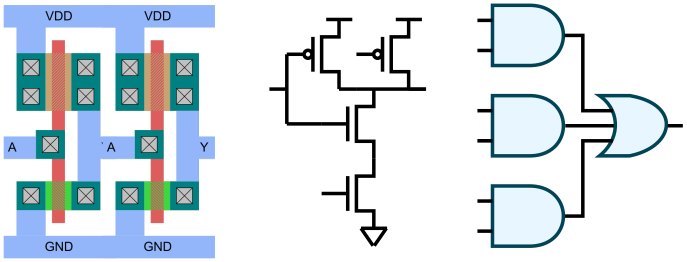


Topic 8
Testing and Verification

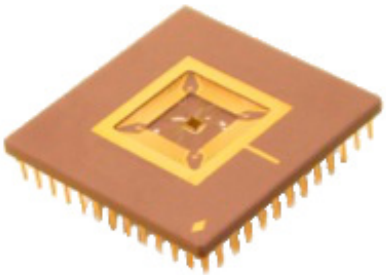
Topic 4
Full-Custom Design Methodology

Topic 6
Closing the Gap

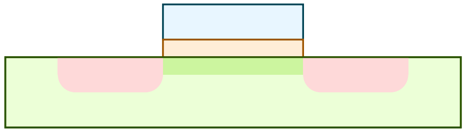
Topic 5
Automated Design Methodologies



Topic 3
CMOS Circuits

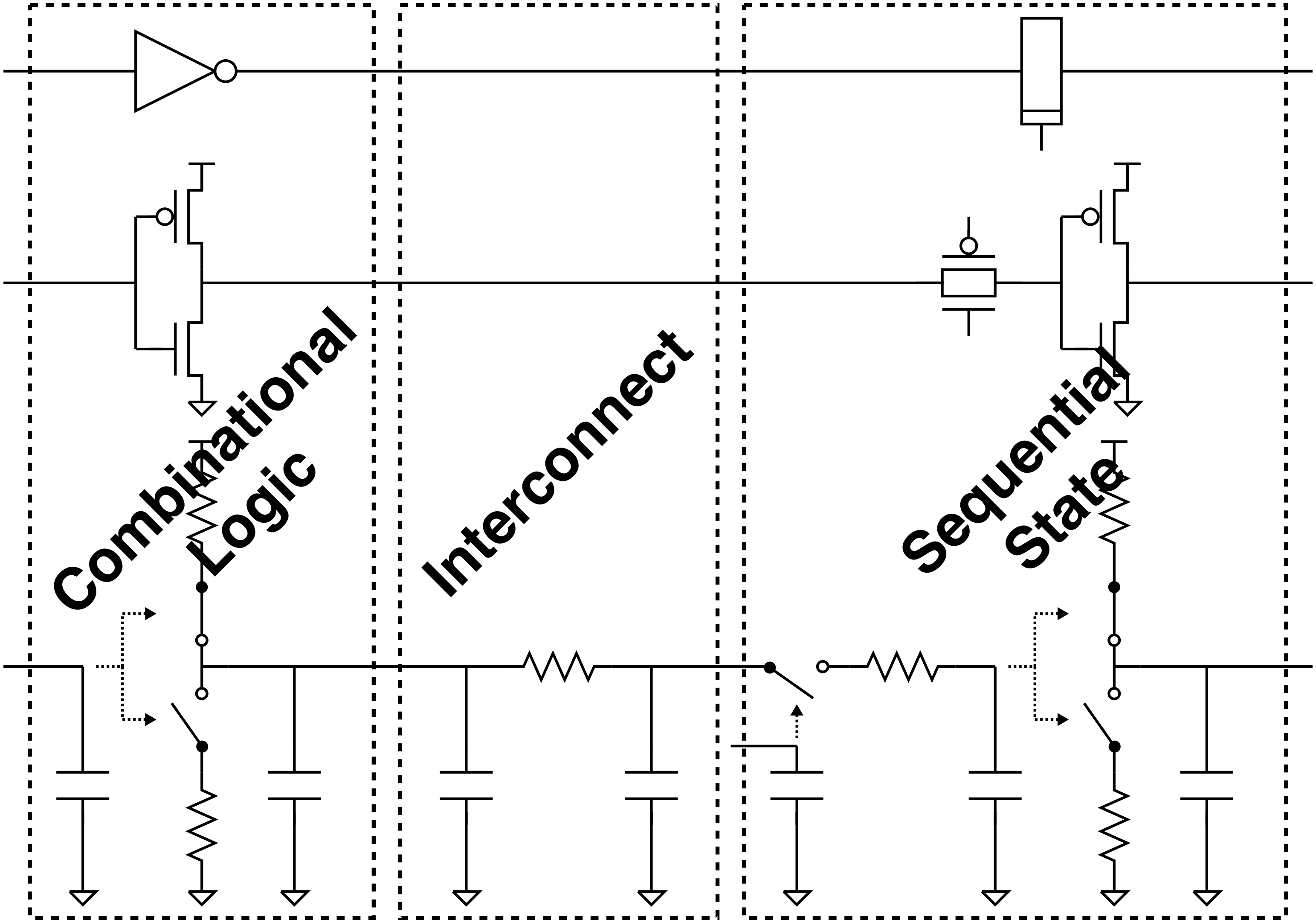


Topic 7
Clocking, Power Distribution, Packaging, and I/O

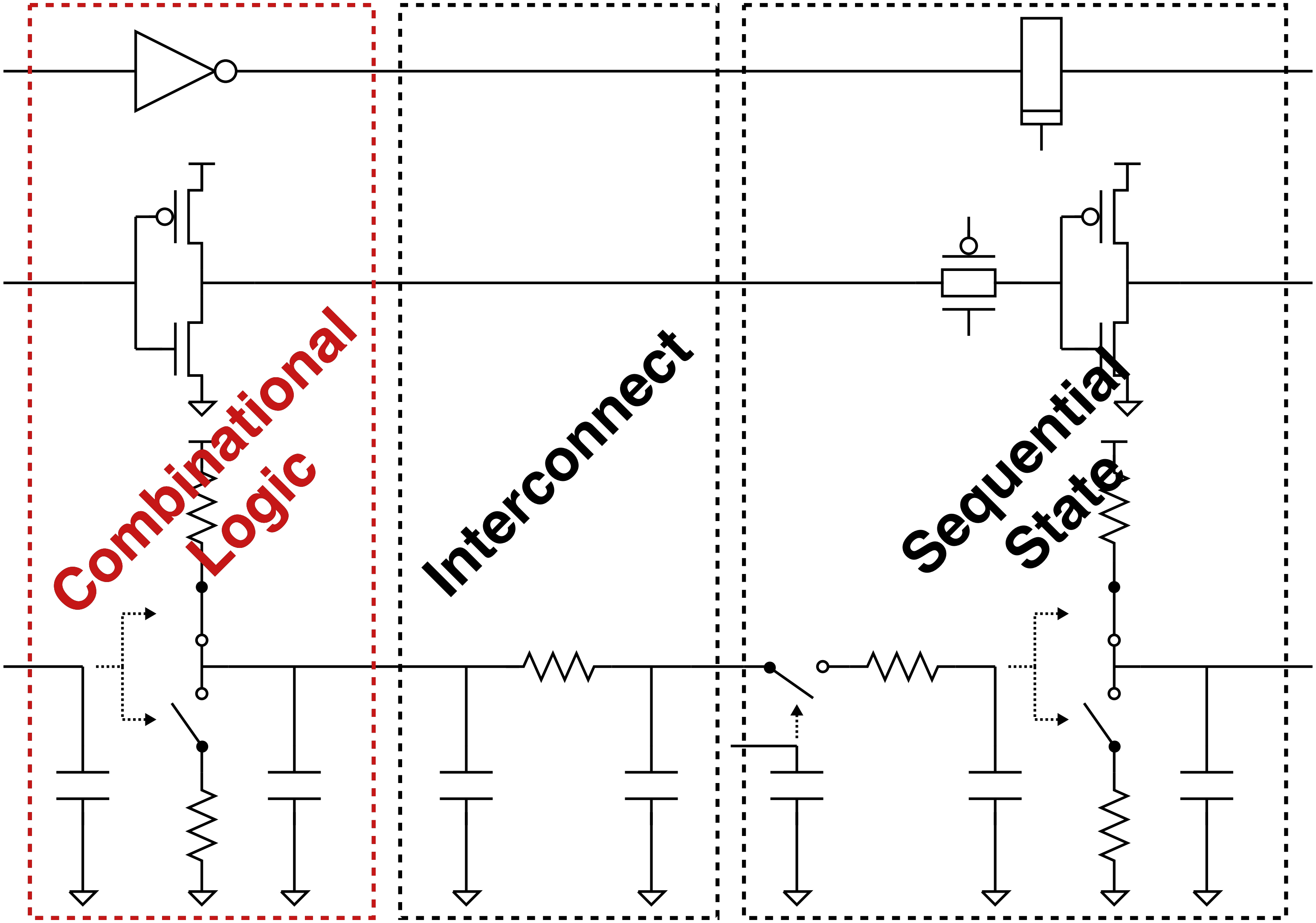


Topic 2
CMOS Devices

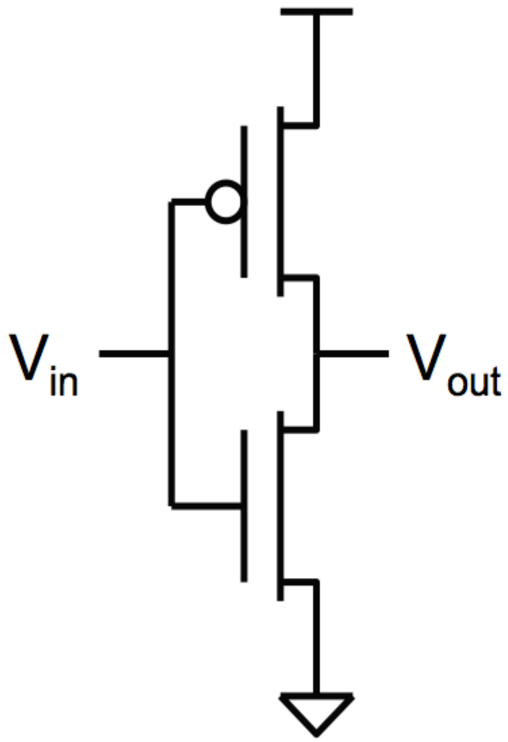
CMOS Logic, State, Interconnect

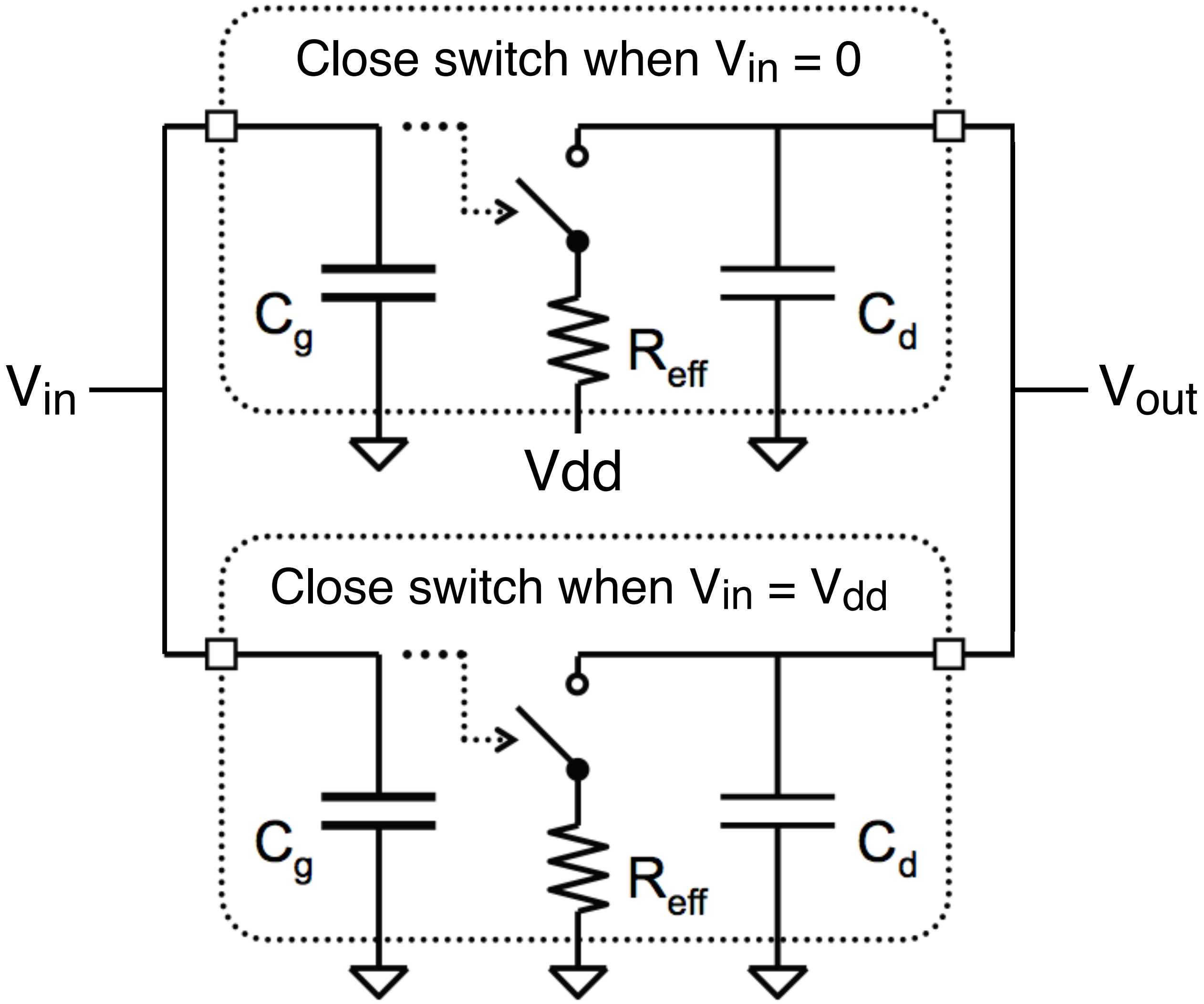


CMOS Logic, State, Interconnect

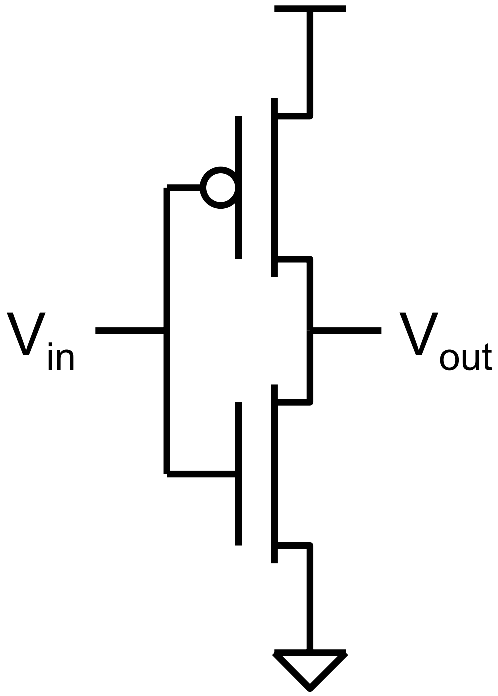


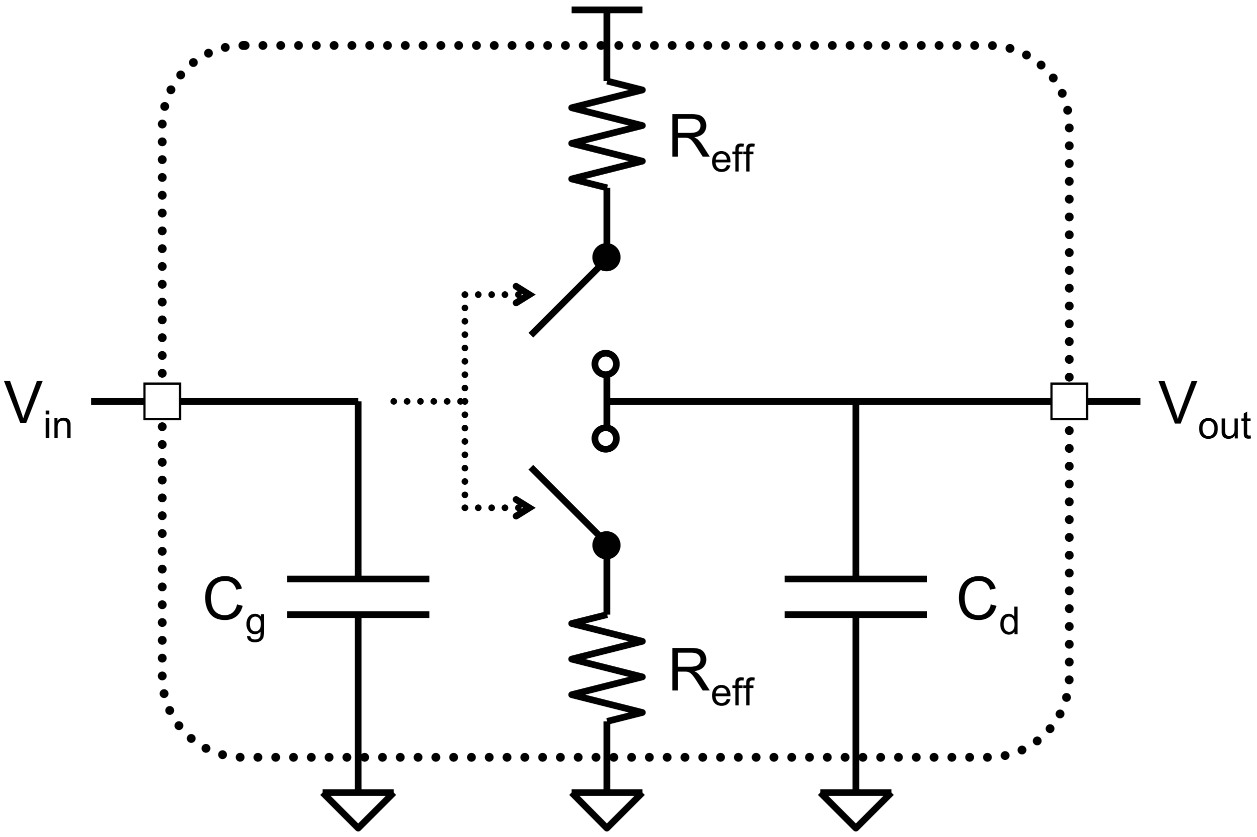
CMOS Inverter Simple RC Model




Close switch when $V_{in} = 0$
C_g
R_{eff}
C_d
V_{in}
V_{out}
Vdd
Close switch when $V_{in} = V_{dd}$
C_g
R_{eff}
C_d

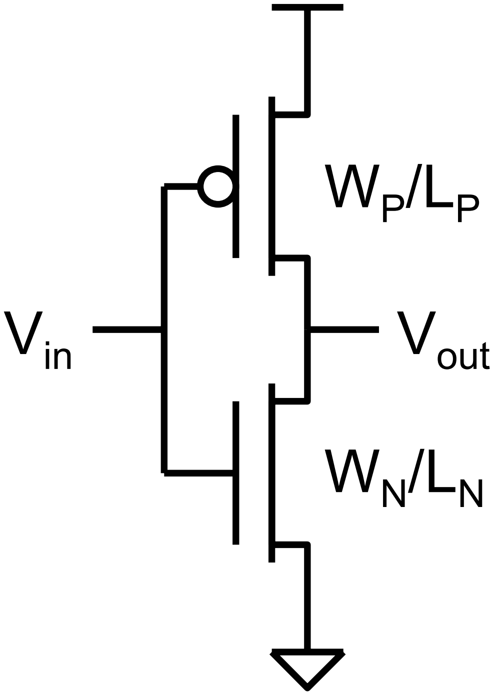
CMOS Inverter Simple RC Model

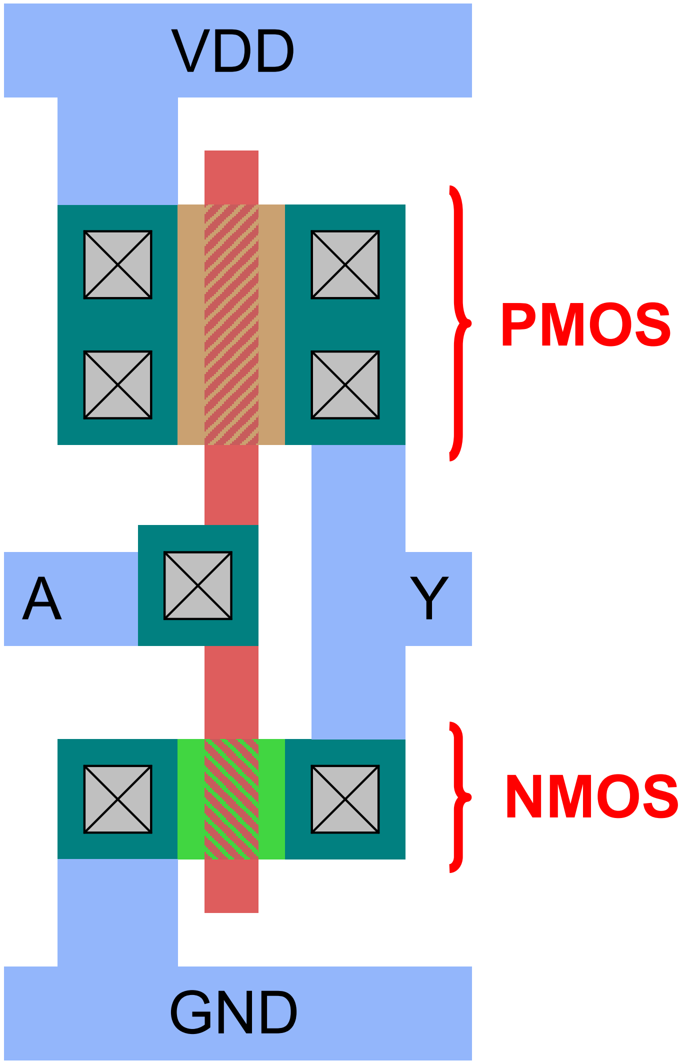




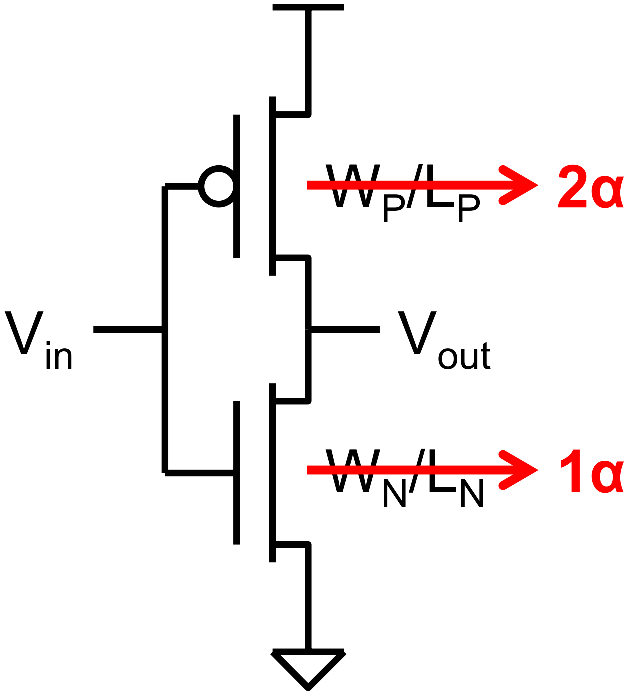
$$R_{eff} = R_{eff,N} = R_{eff,P}$$
$$C_g = C_{g,N} + C_{g,P}$$
$$C_d = C_{d,N} + C_{d,P}$$

CMOS Inverter Layout





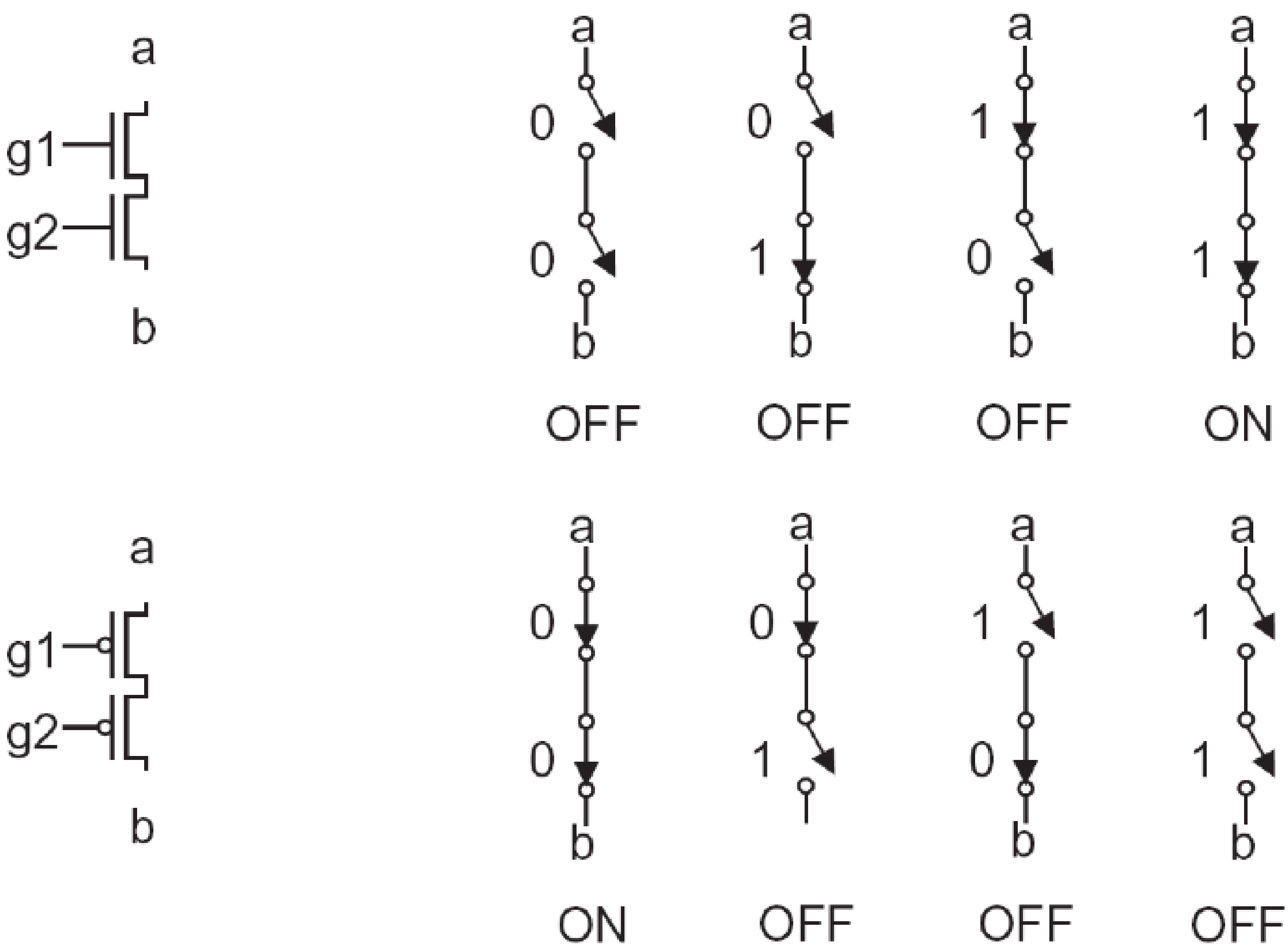
CMOS Inverter



Let's make the following assumptions

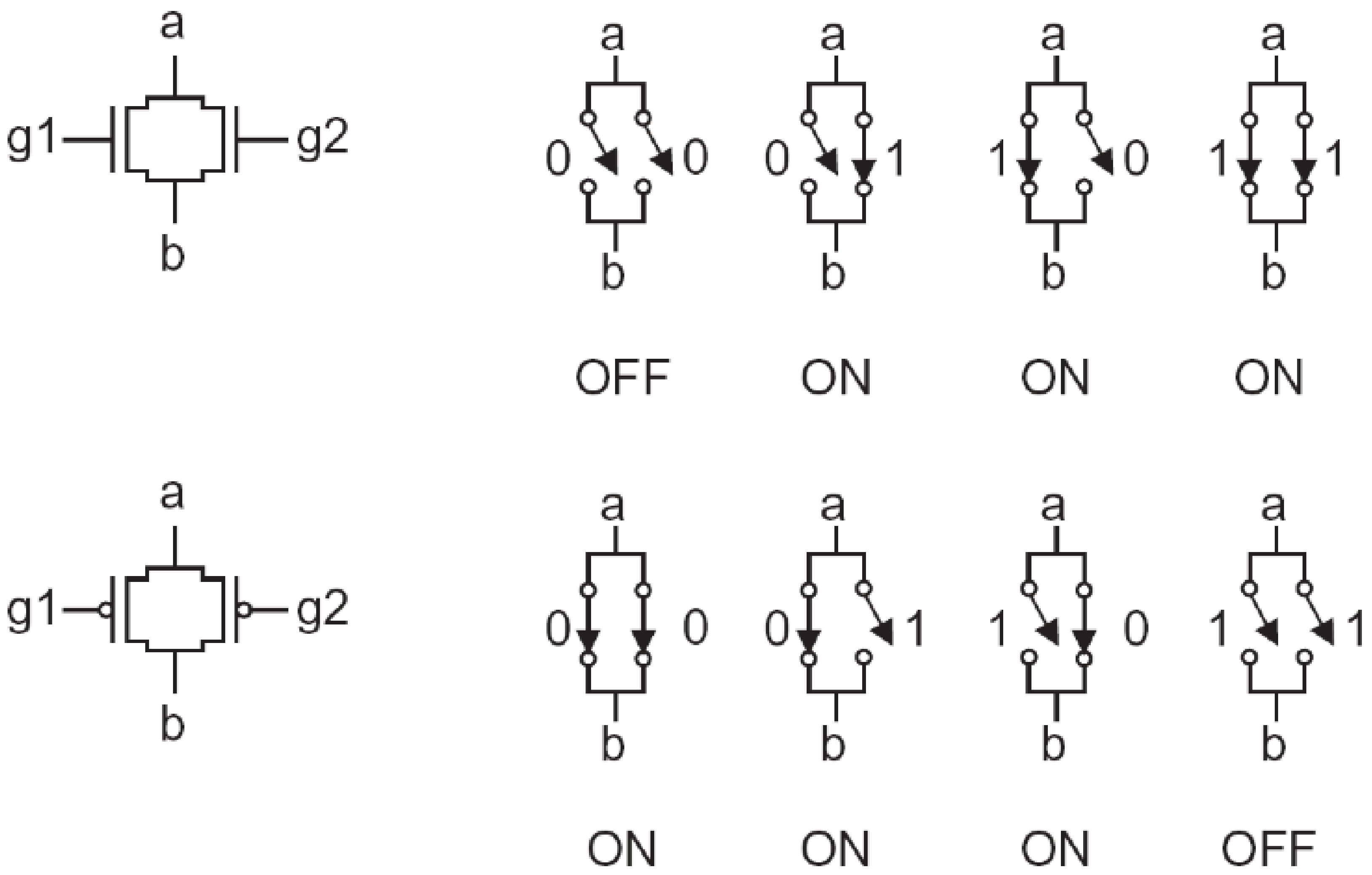
1. All transistors are minimum length
2. All gates should have equal rise/fall times. Since PMOS are twice as slow as NMOS they must be twice as wide to have the same effective resistance
3. Normalize all transistor widths to minimum width NMOS

Series Transistors



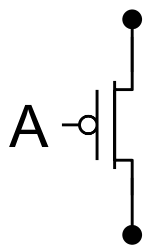
Adapted from [Weste'11]

Parallel Transistors

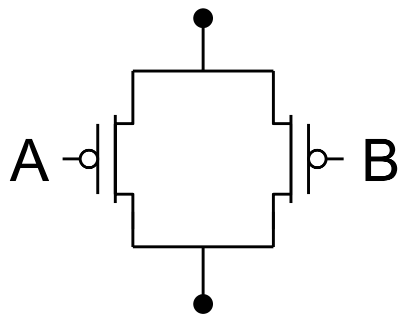


Adapted from [Weste'11]

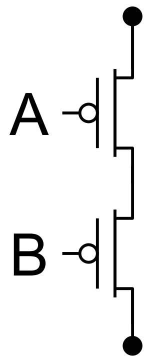
Series/Parallel Transistor Networks are Natural Duals



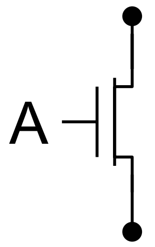
Conducts if A=0



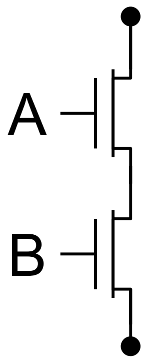
Conducts if A=0 **OR** B=0



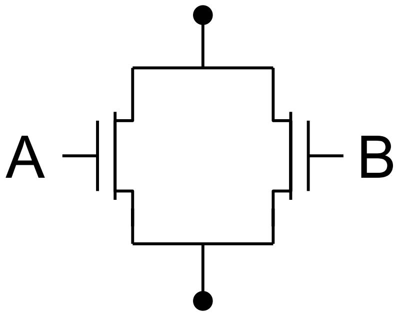
Conducts if A=0 **AND** B=0



Conducts if A=1

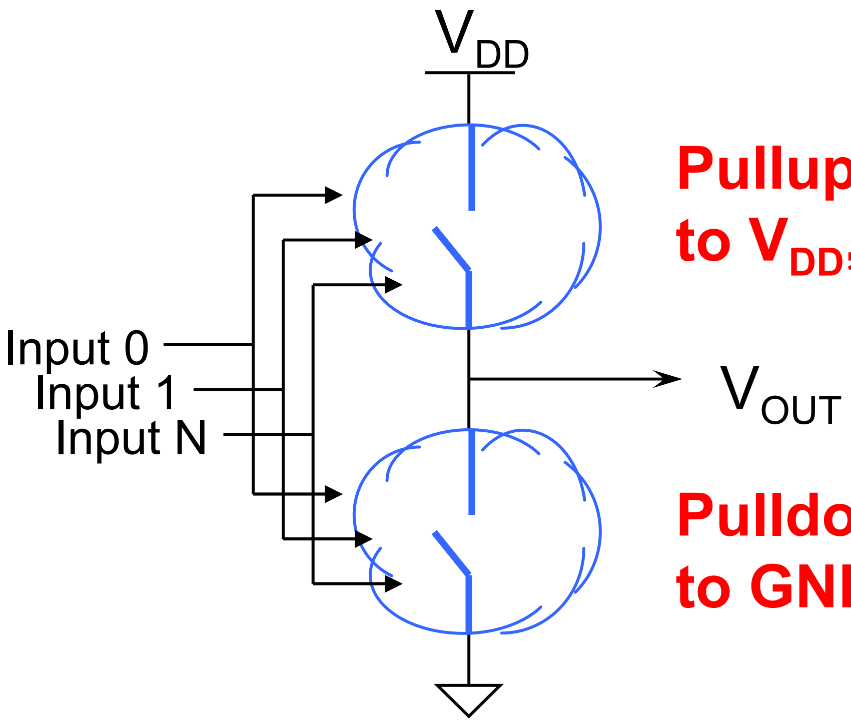


Conducts if A=1 **AND** B=1



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

CMOS Static Logic Style



Pullup network, connects output to V_{DD}, contains only PMOS

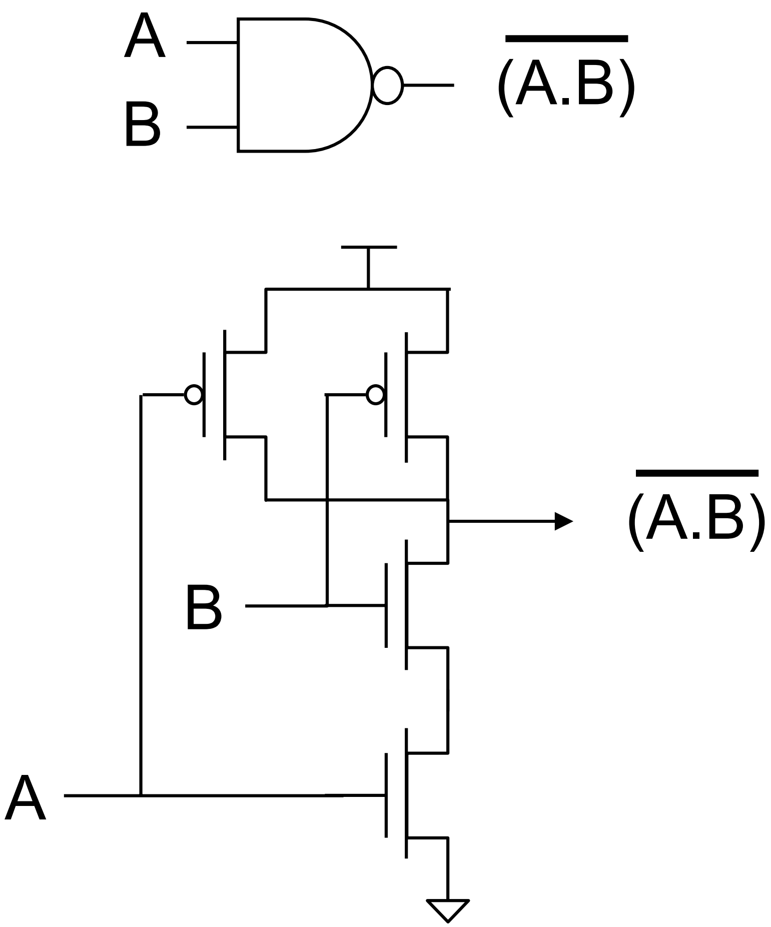
Pulldown network, connects output to GND, contains only NMOS

For every set of input logic values, either pullup or pulldown network makes connection to VDD or GND

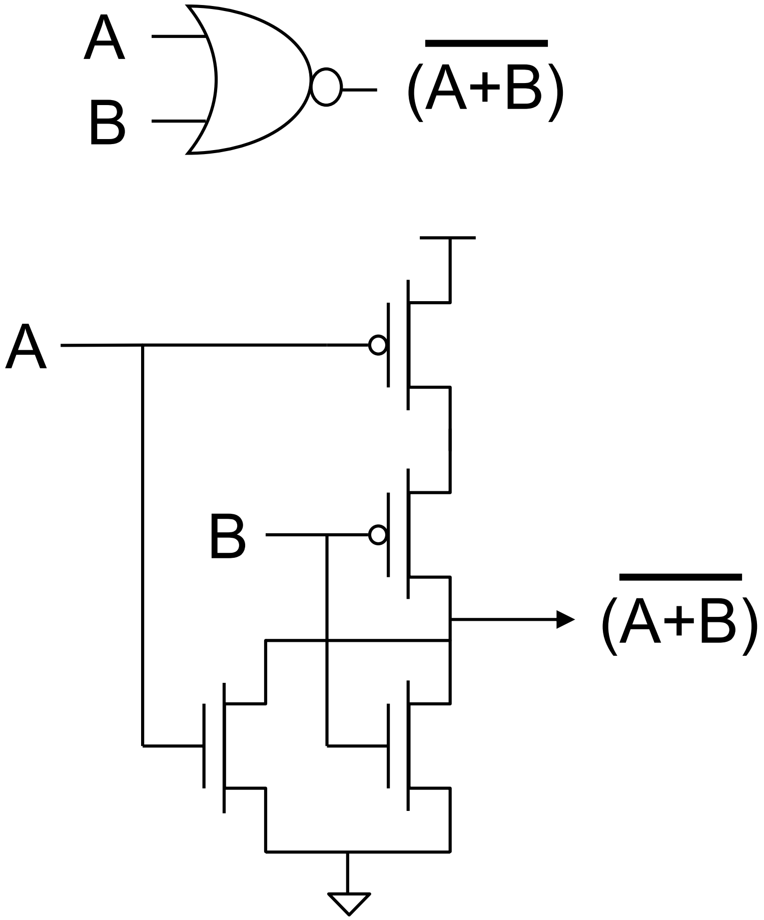
- If both connected, power rails would be shorted together
- If neither connected, output would float (tristate logic)

NAND/NOR Static CMOS Logic Gates

NAND Gate



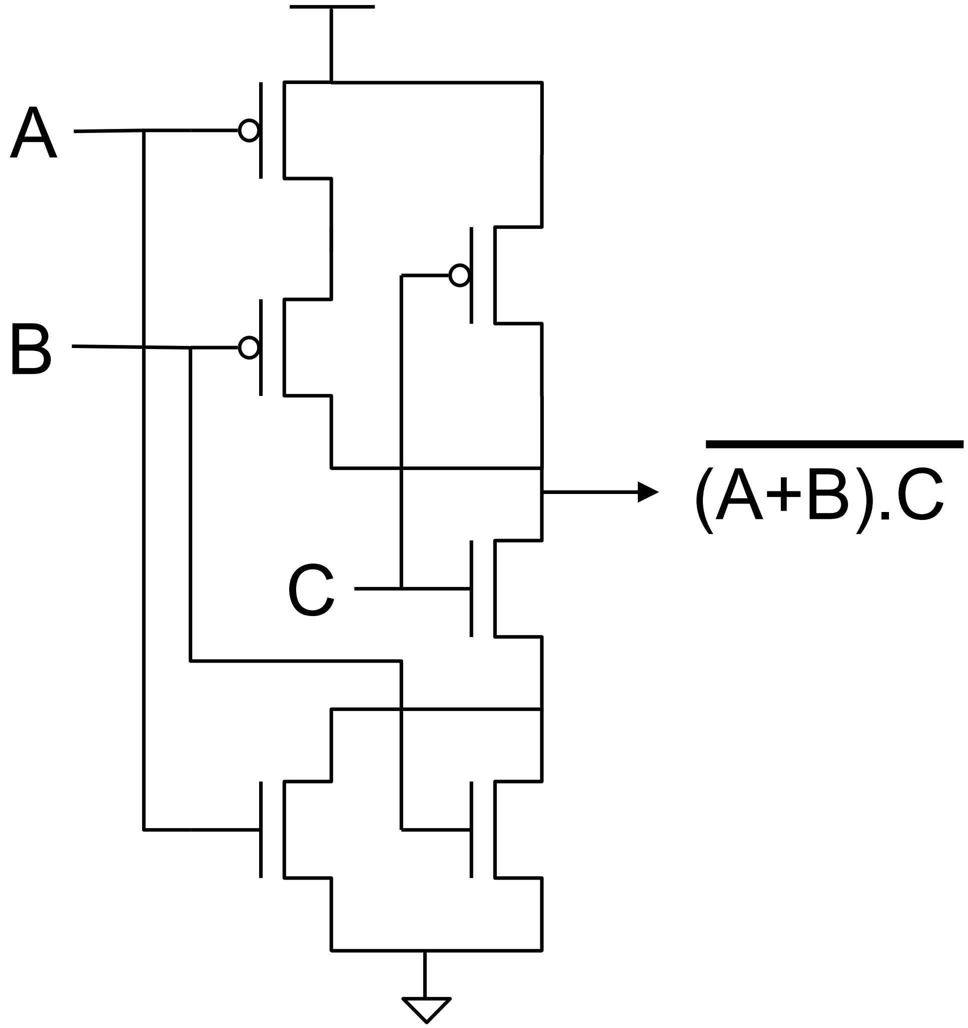
NOR Gate



Approach for Designing More Complex Gates

- Goal is to create a logic function $f(x_1, x_2, \ldots)$
  - We can only implement inverting logic with one CMOS stage
- Implement pulldown network
  - Write $PD = \overline{f(x_1, x_2, \ldots)}$
  - Use parallel NMOS for OR of inputs
  - Use series NMOS for AND of inputs
- Implement pullup network
  - Write $PU = f(x_1, x_2, \ldots) = g(\overline{x_1}, \overline{x_2}, \ldots)$
  - Use parallel PMOS for OR of inverted inputs
  - Use series PMOS for AND of inverted inputs

Complex Logic Gate Example

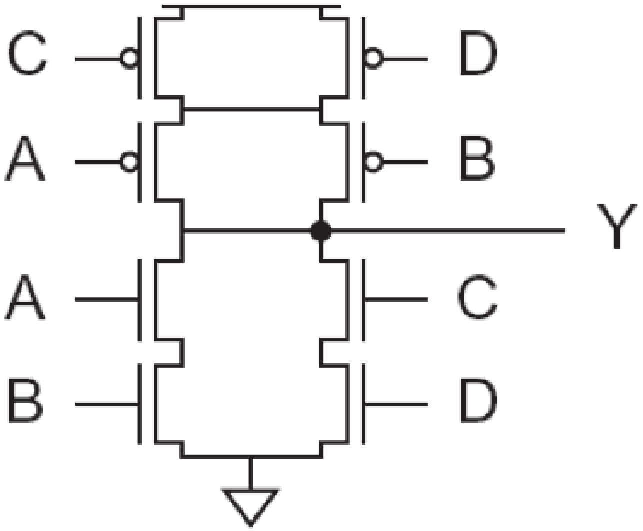


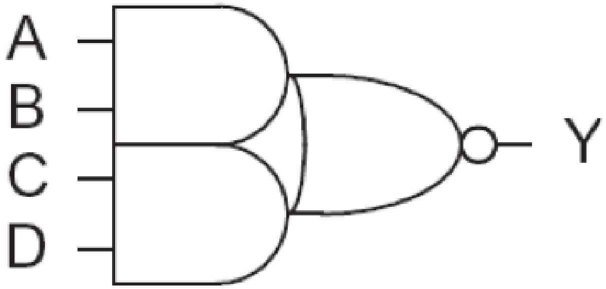
$$f = \overline{(A + B) \cdot C}$$

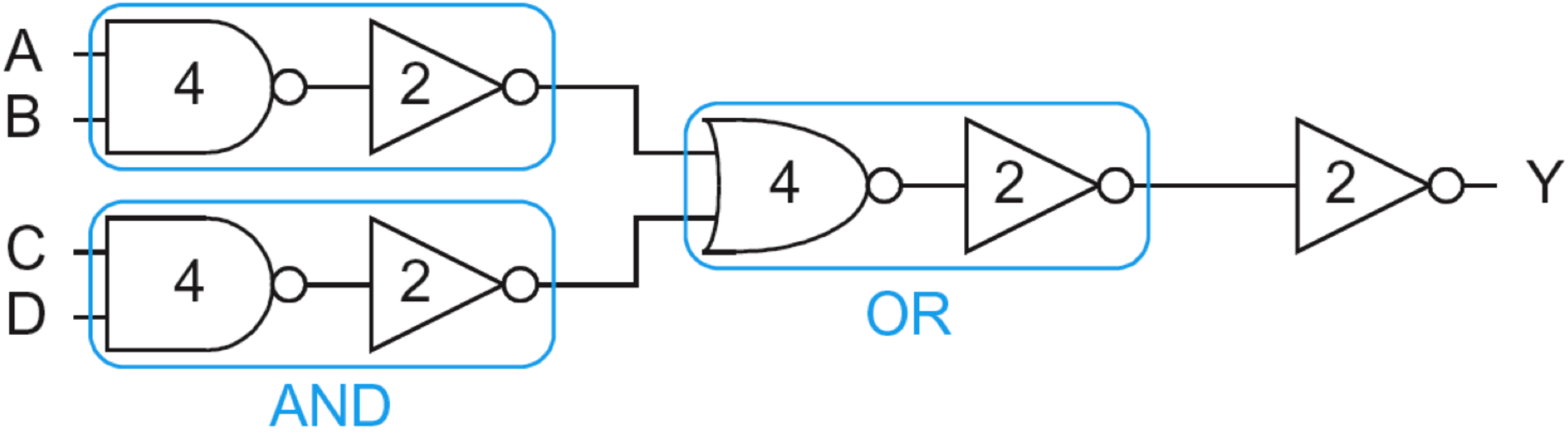
$$PD = (A + B) \cdot C$$

$$\begin{aligned} PU &= \overline{(A + B) \cdot C} \\ &= \overline{(A + B)} + \overline{C} \\ &= (\overline{A} \cdot \overline{B}) + \overline{C} \end{aligned}$$

Single- vs. Multi-Stage Static CMOS Logic

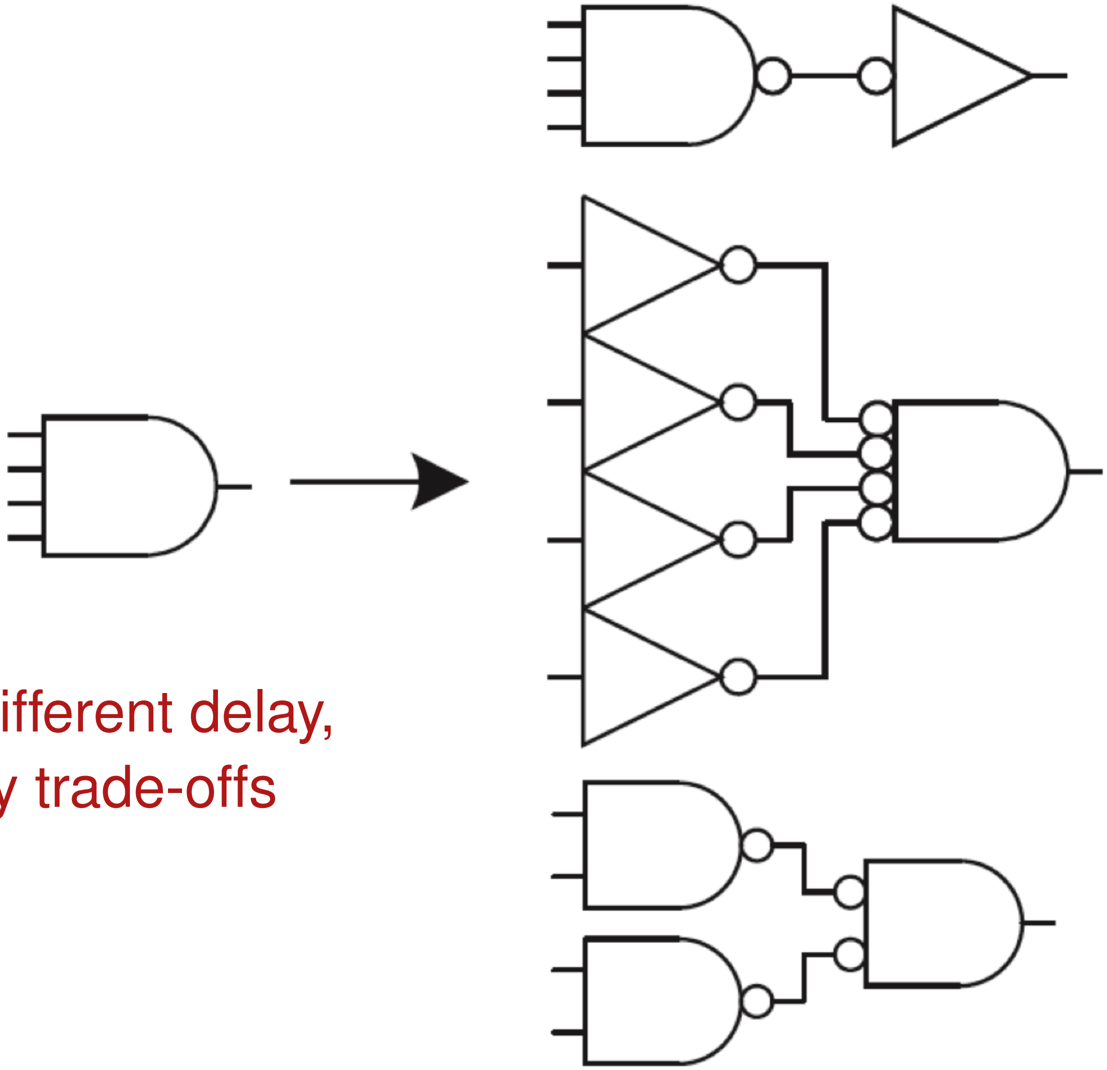






Adapted from [Weste'11]

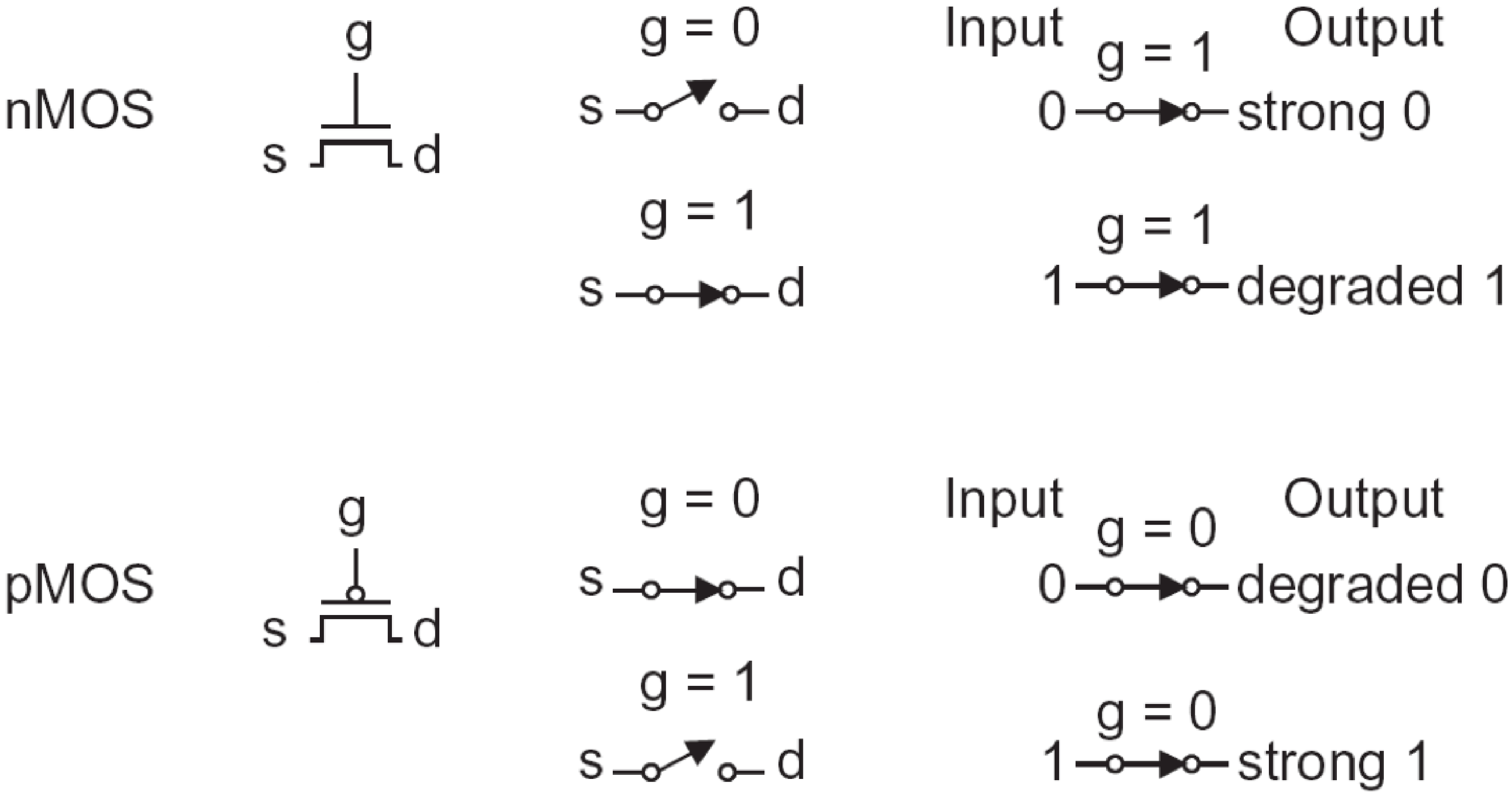
Multiple Stages of Static CMOS Logic



Each design has different delay, area, and energy trade-offs

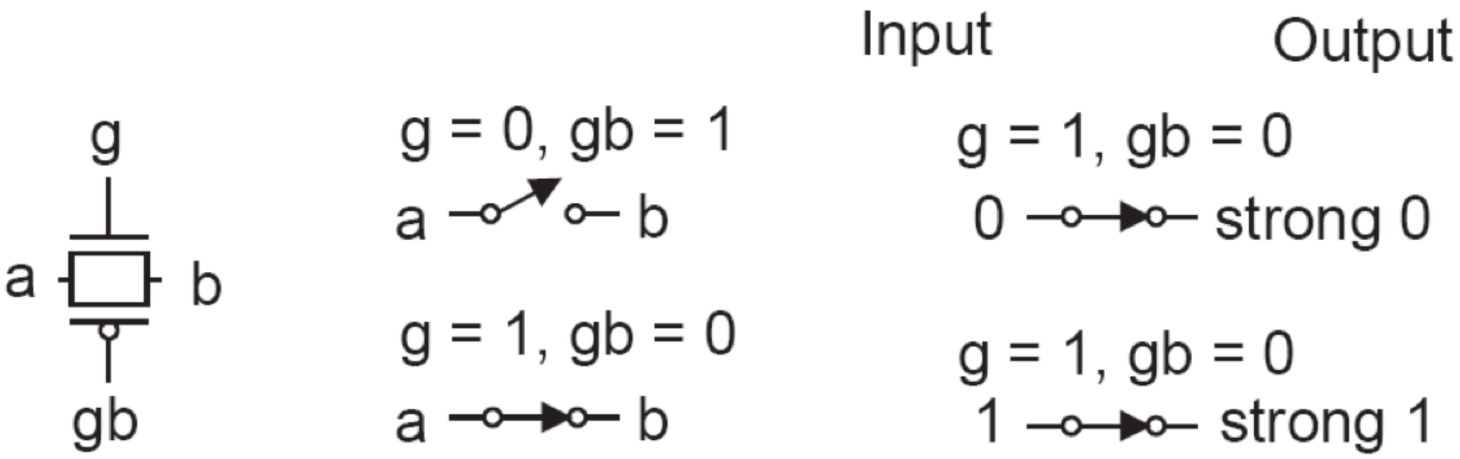
Adapted from [Weste'11]

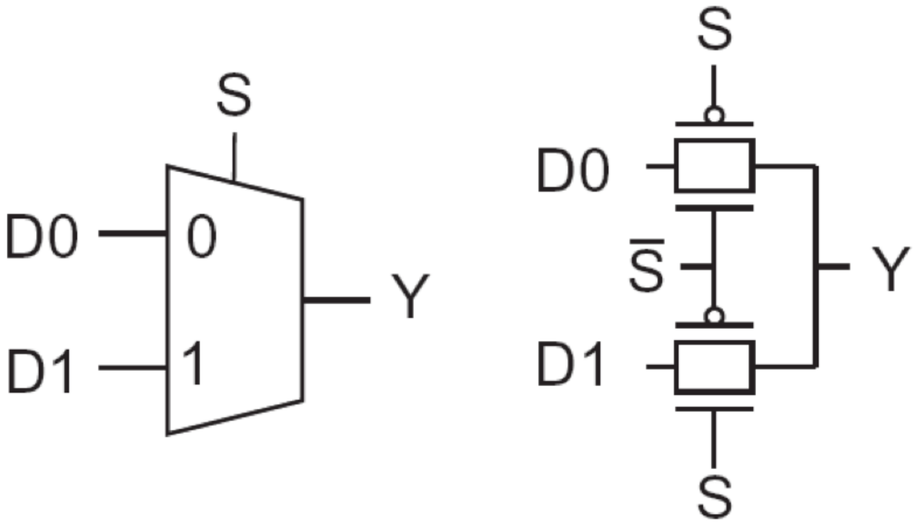
CMOS Pass-Transistor Logic Style



Adapted from [Weste'11]

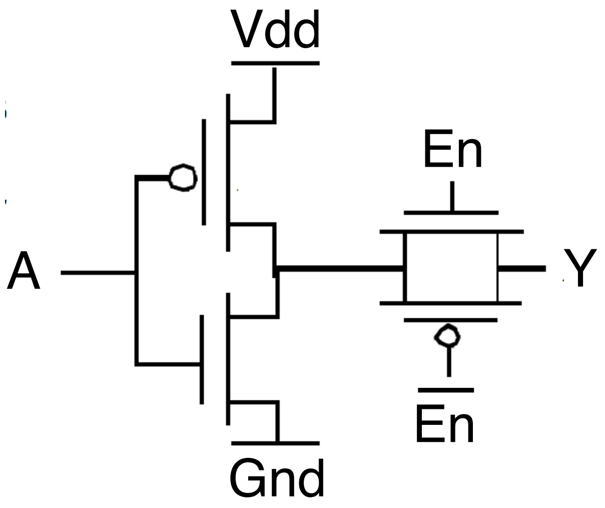
CMOS Transmission Gate Multiplexer

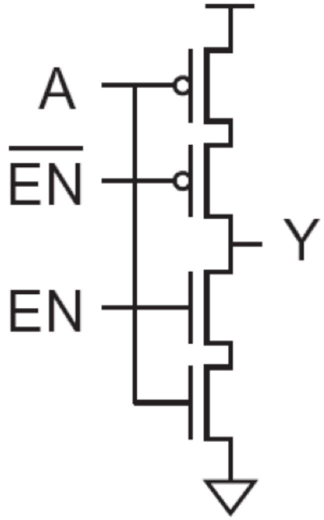


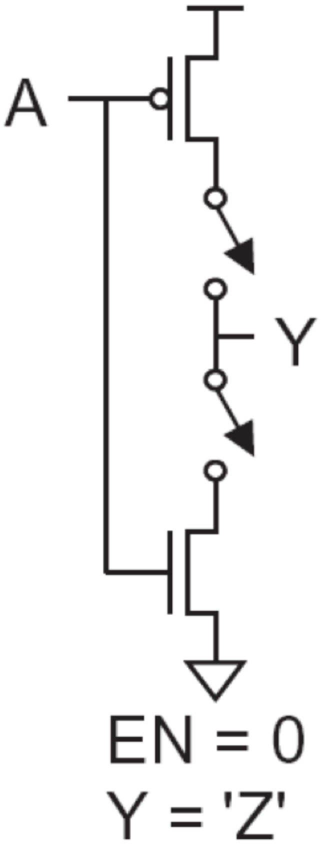


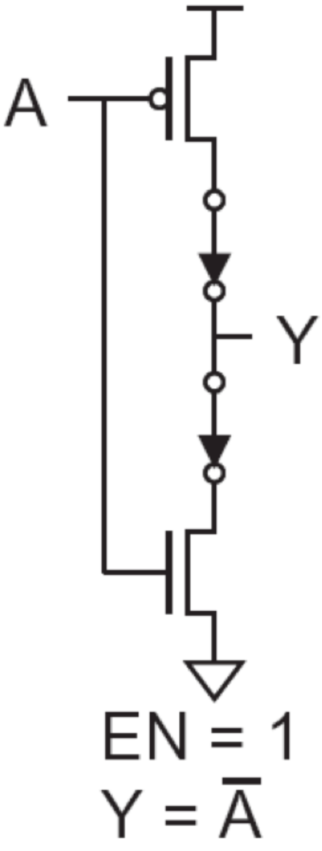
Adapted from [Weste'11]

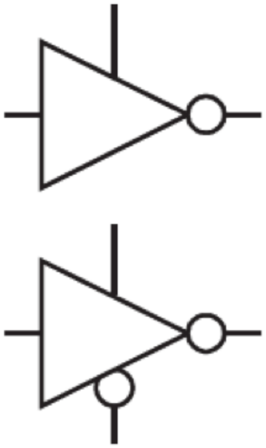
CMOS Tri-State Buffers





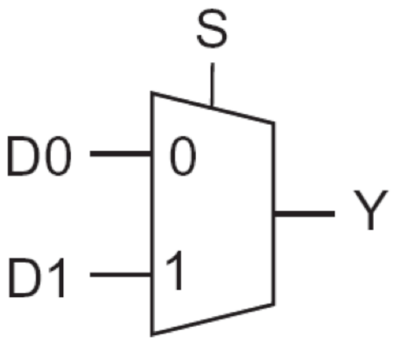


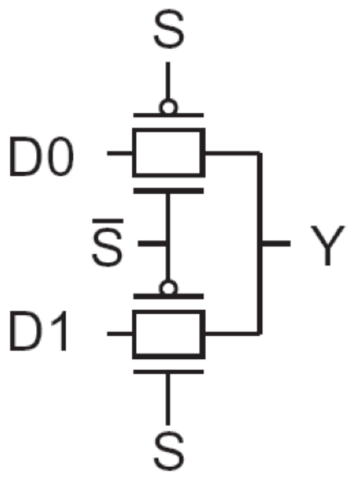




Adapted from [Weste'11]

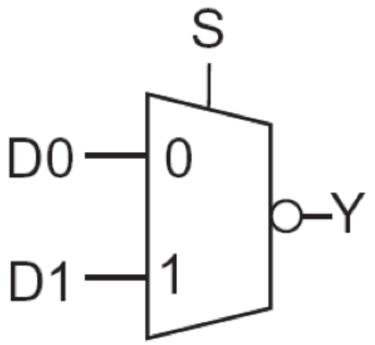
Various Multiplexer Implementations

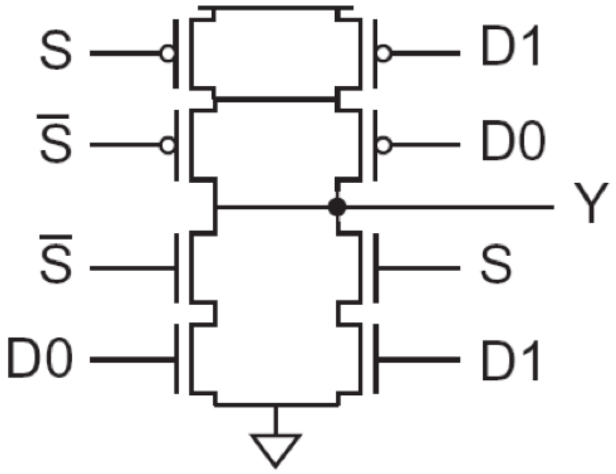


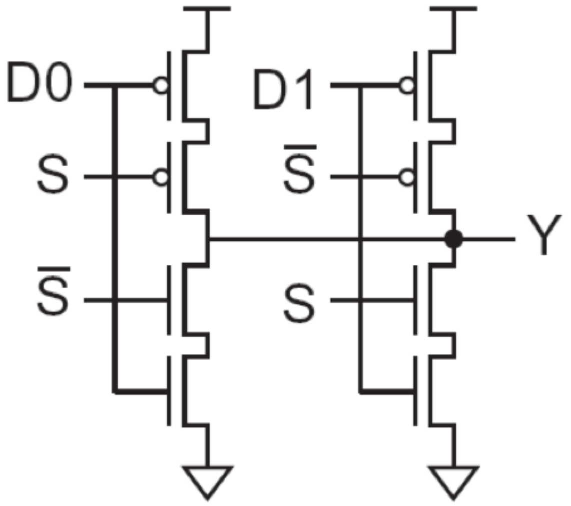


Each design has different delay, area, and energy trade-offs

Simple first-order analysis can help suggest some of these trade-offs

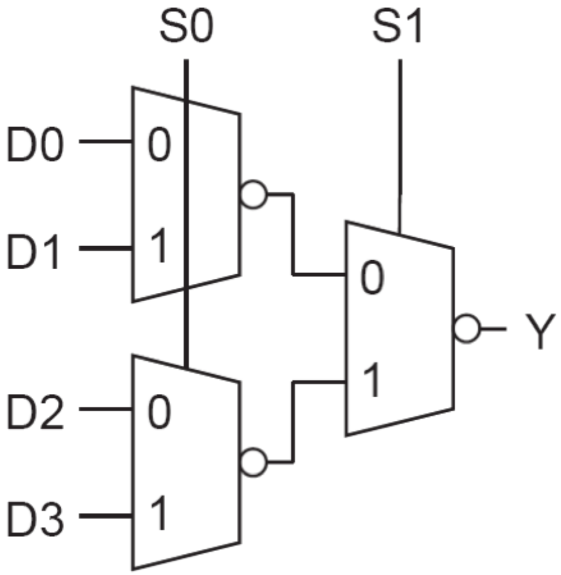


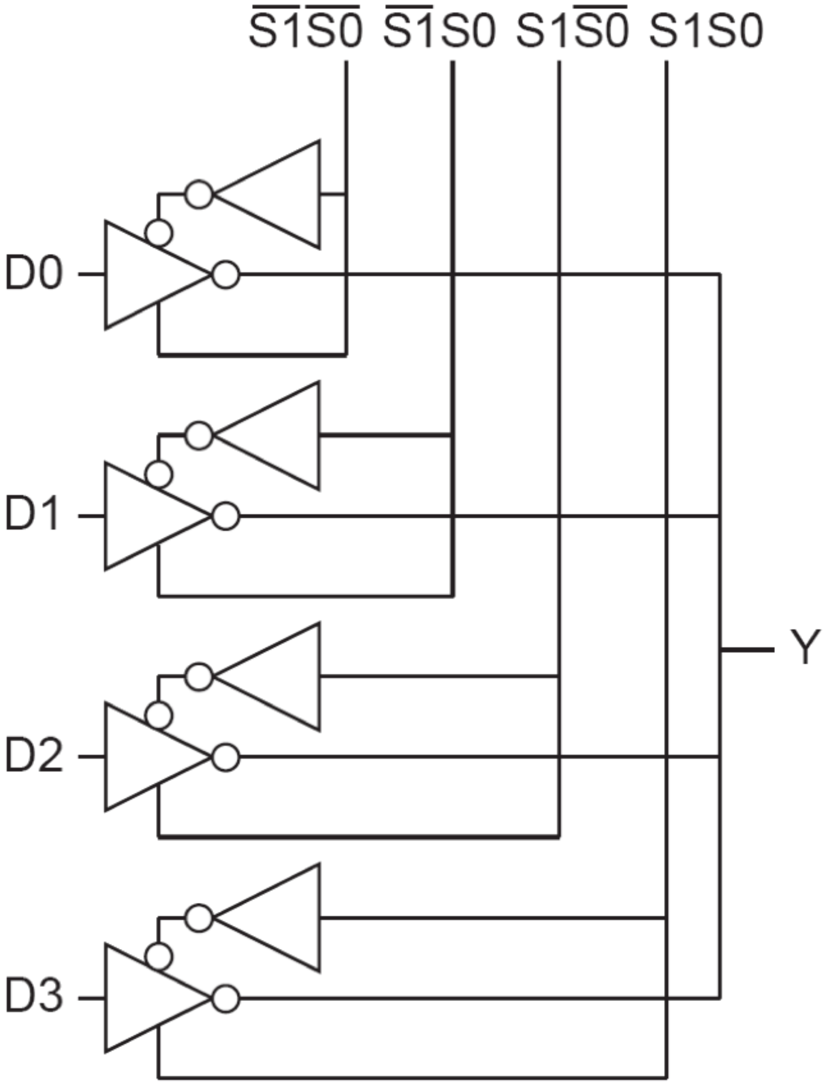




Adapted from [Weste'11]

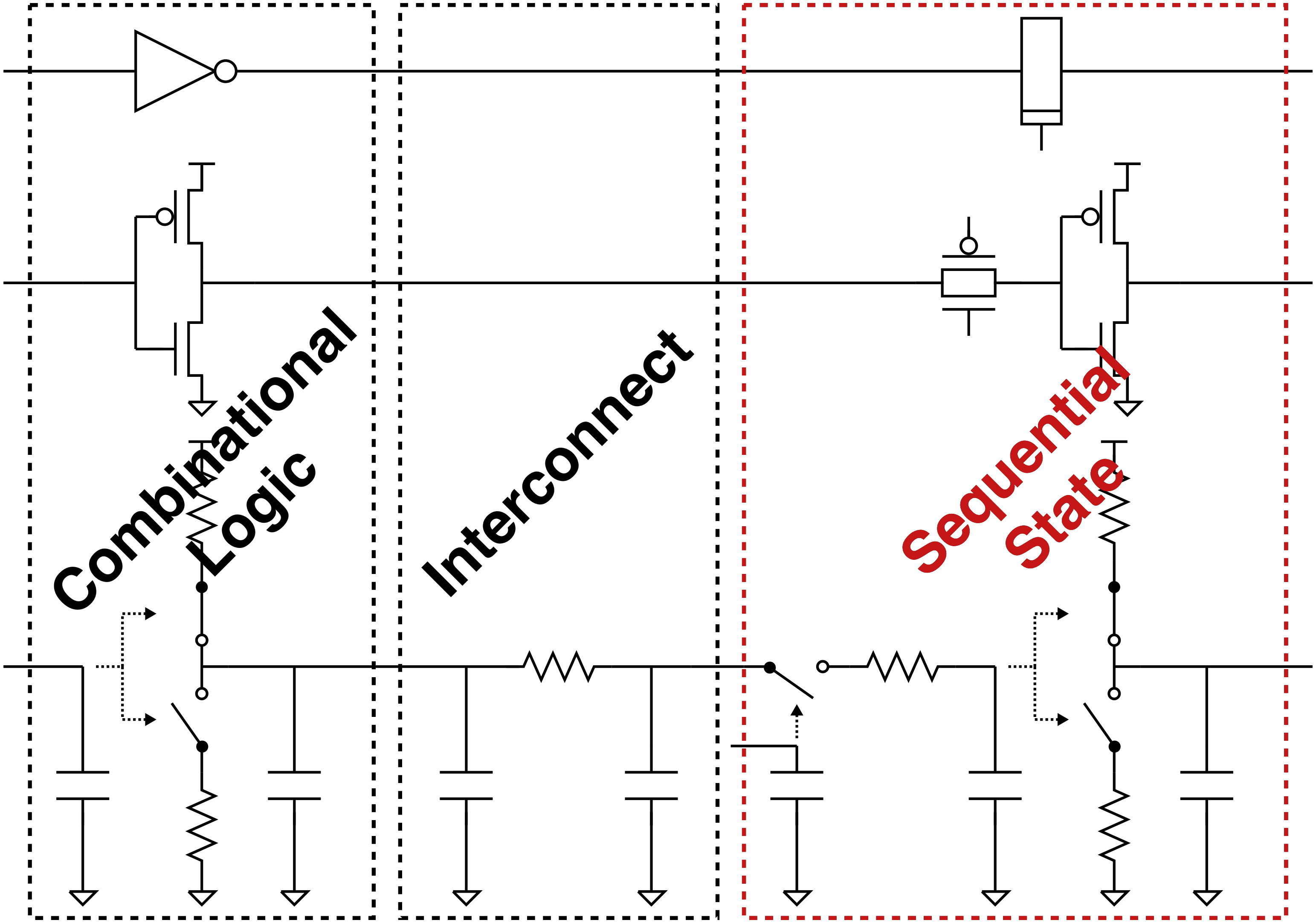
Larger Tri-State Multiplexers



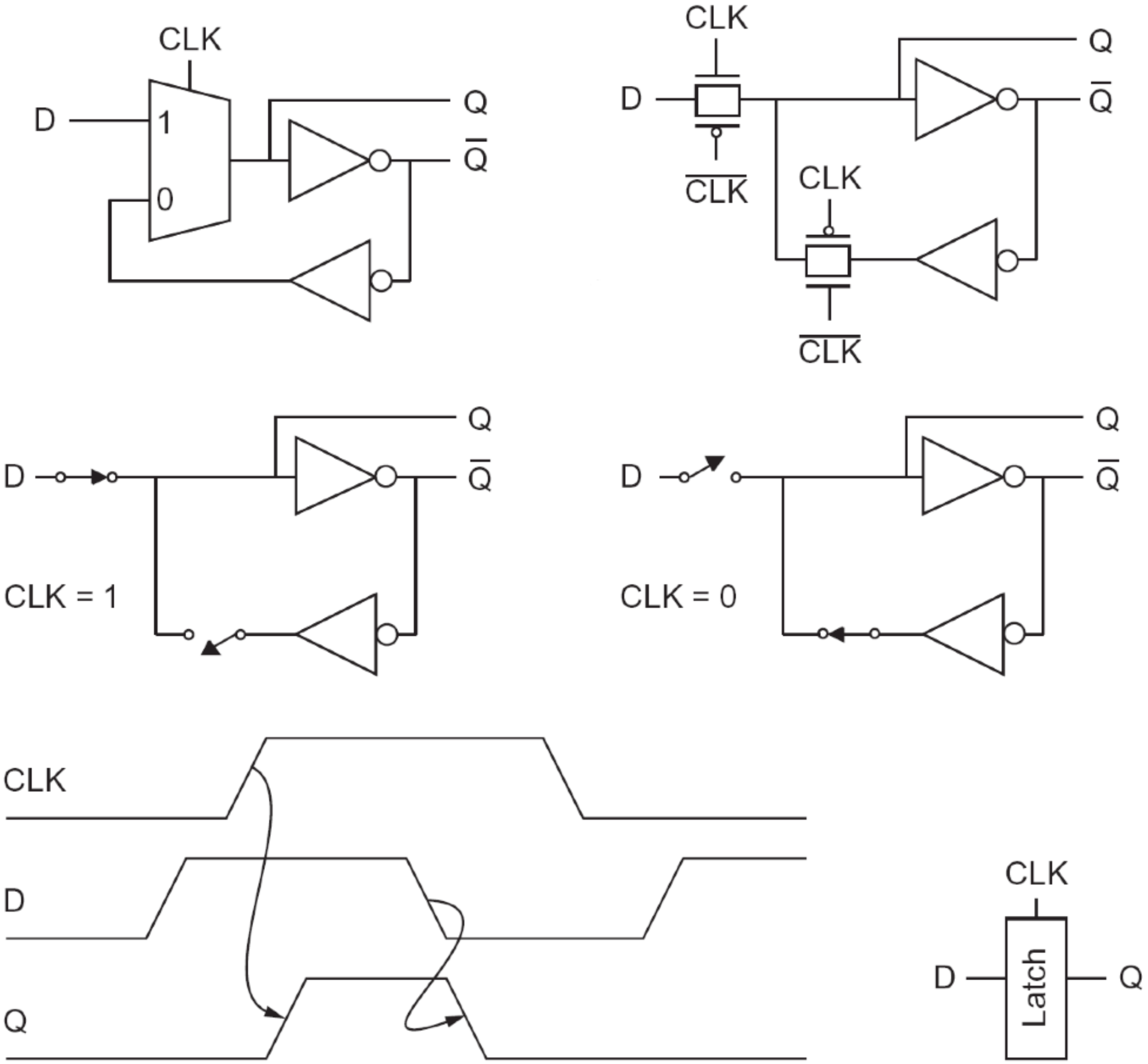


Adapted from [Weste'11]

CMOS Logic, State, Interconnect

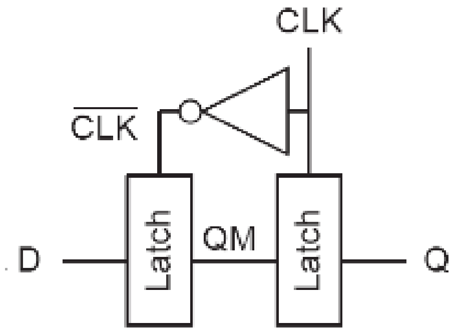


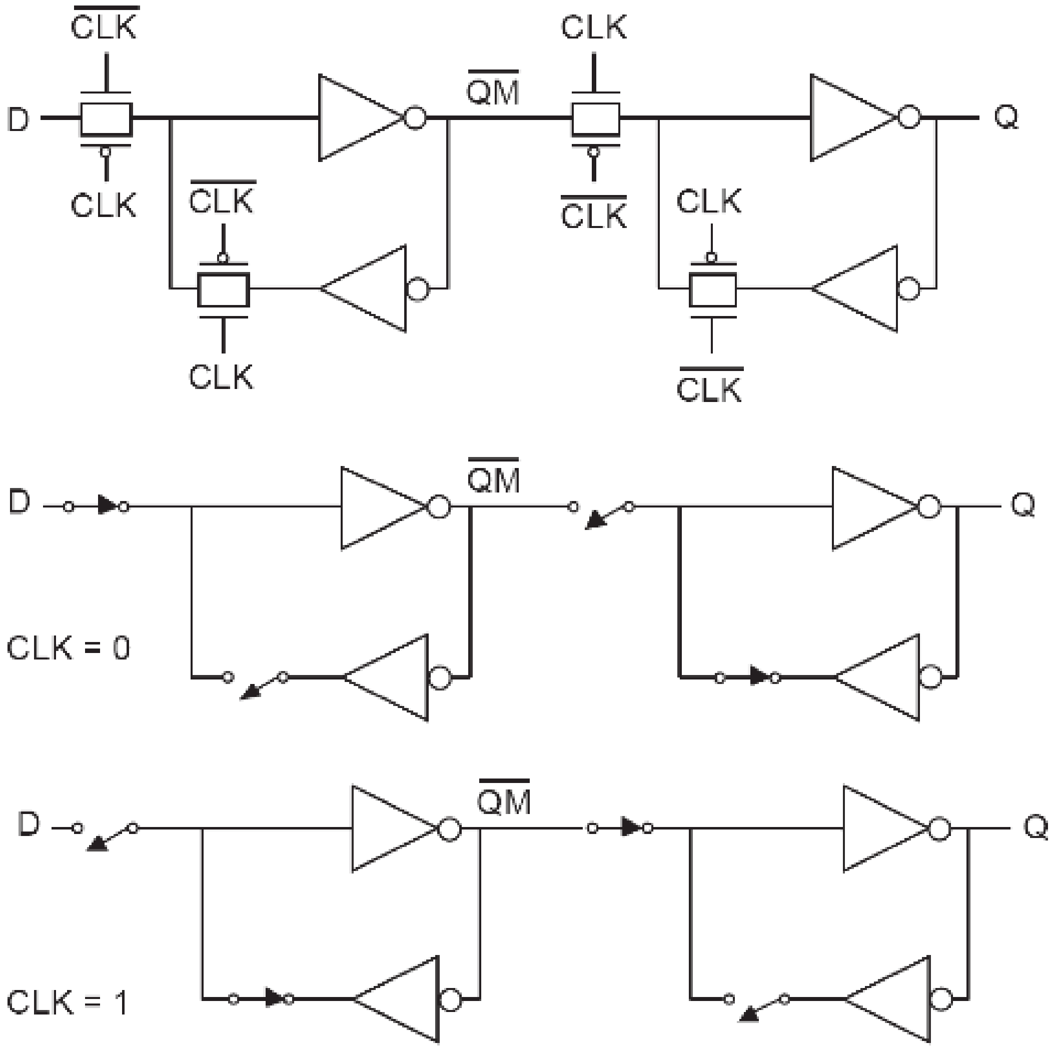
Level-High Latch



Adapted from [Weste'11]

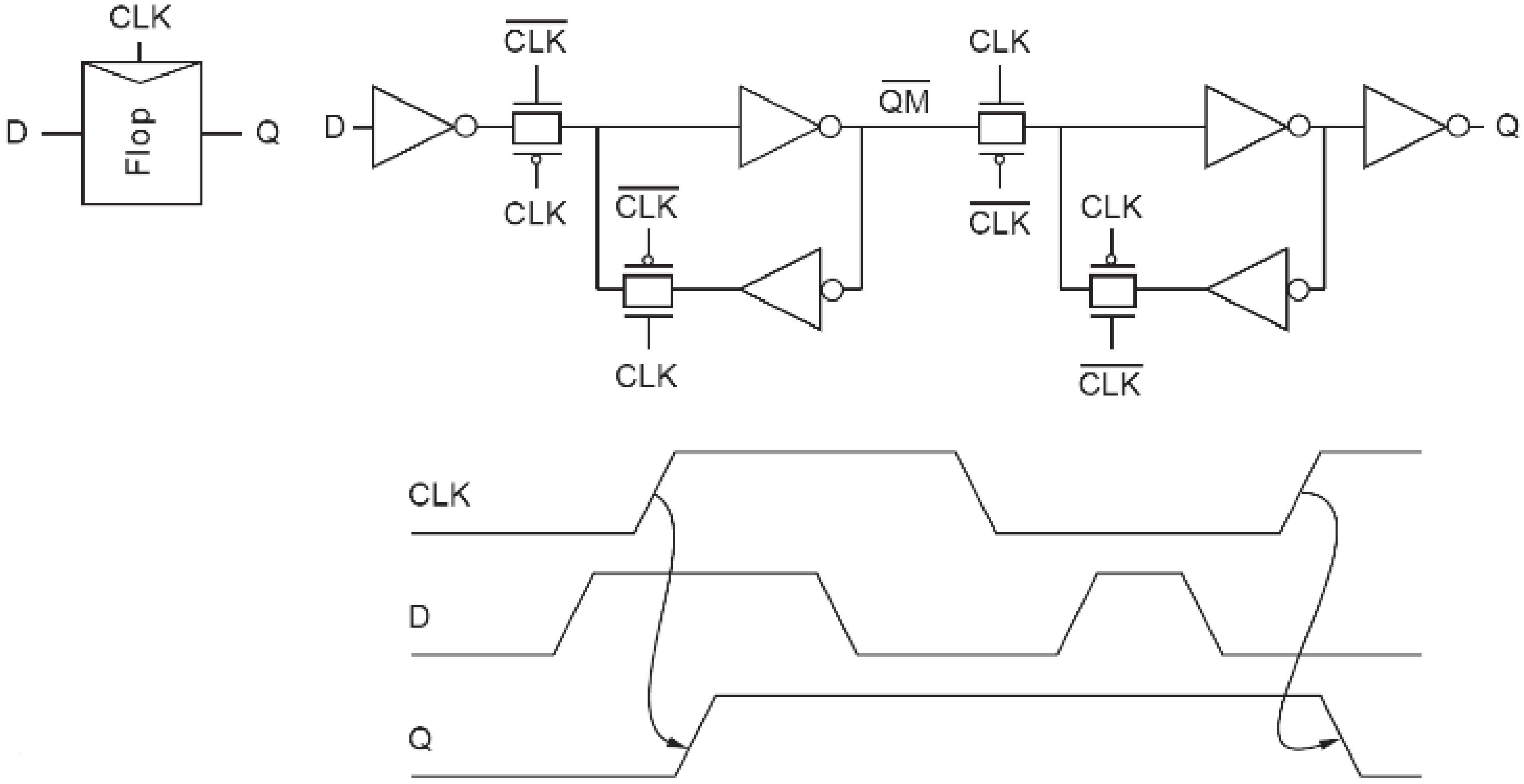
Positive-Edge Triggered Flip-Flop





Adapted from [Weste'11]

Positive-Edge Triggered Flip-Flop



Adapted from [Weste'11]

Take-Away Points

- We have reviewed basic CMOS circuit implementations
  - Combinational Logic: static CMOS, pass-transistor, tri-state buffers
  - Sequential State: latches, flip-flops
- In the next two sections, we will explore various methodologies which enable mapping designs written in a hardware-description language down into these circuits
- In the next part of the course, we will explore the details of how to quantitatively evaluate the cycle time, area, and energy of these circuits

Acknowledgments

- [Weste’11] N. Weste and D. Harris, “CMOS VLSI Design: A Circuits and Systems Perspective,” 4th ed, Addison Wesley, 2011.