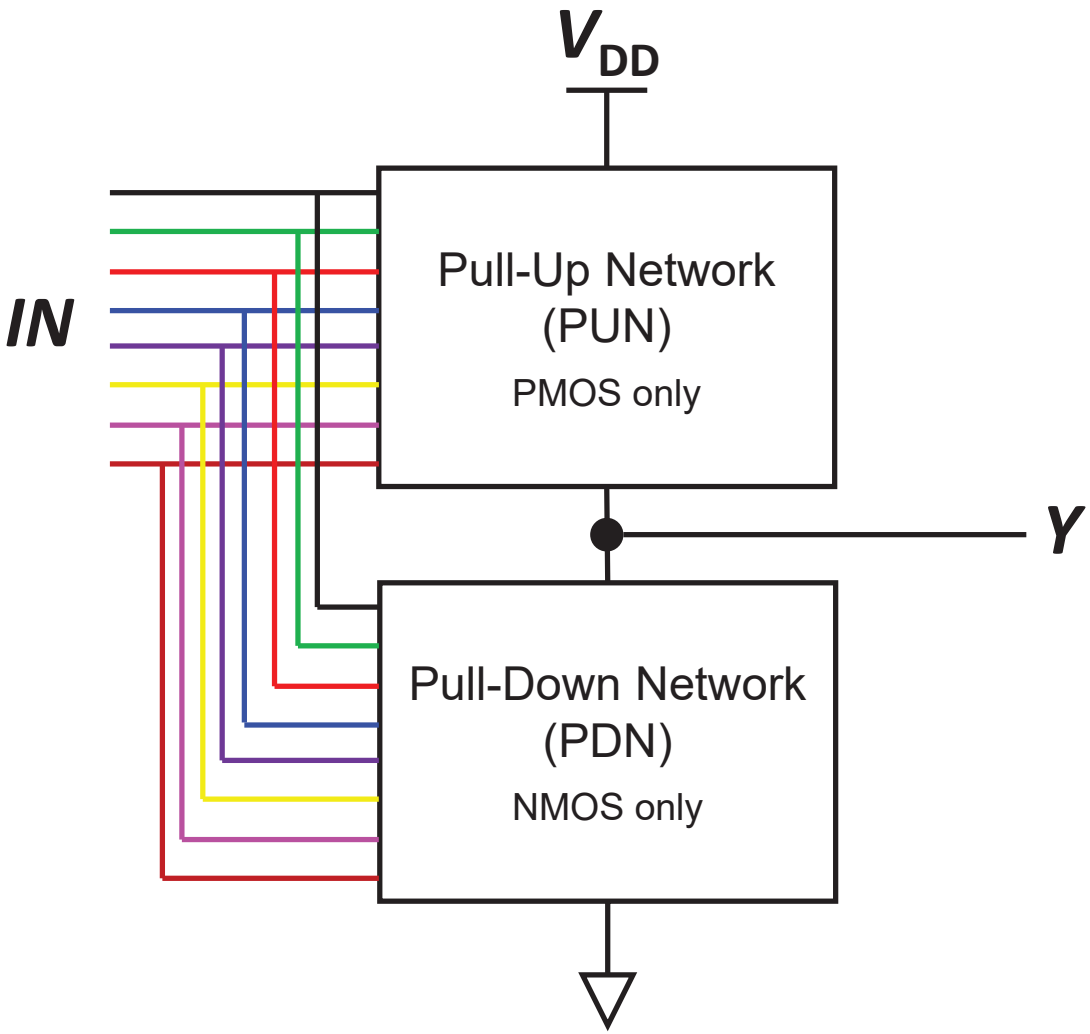
Week 3-1

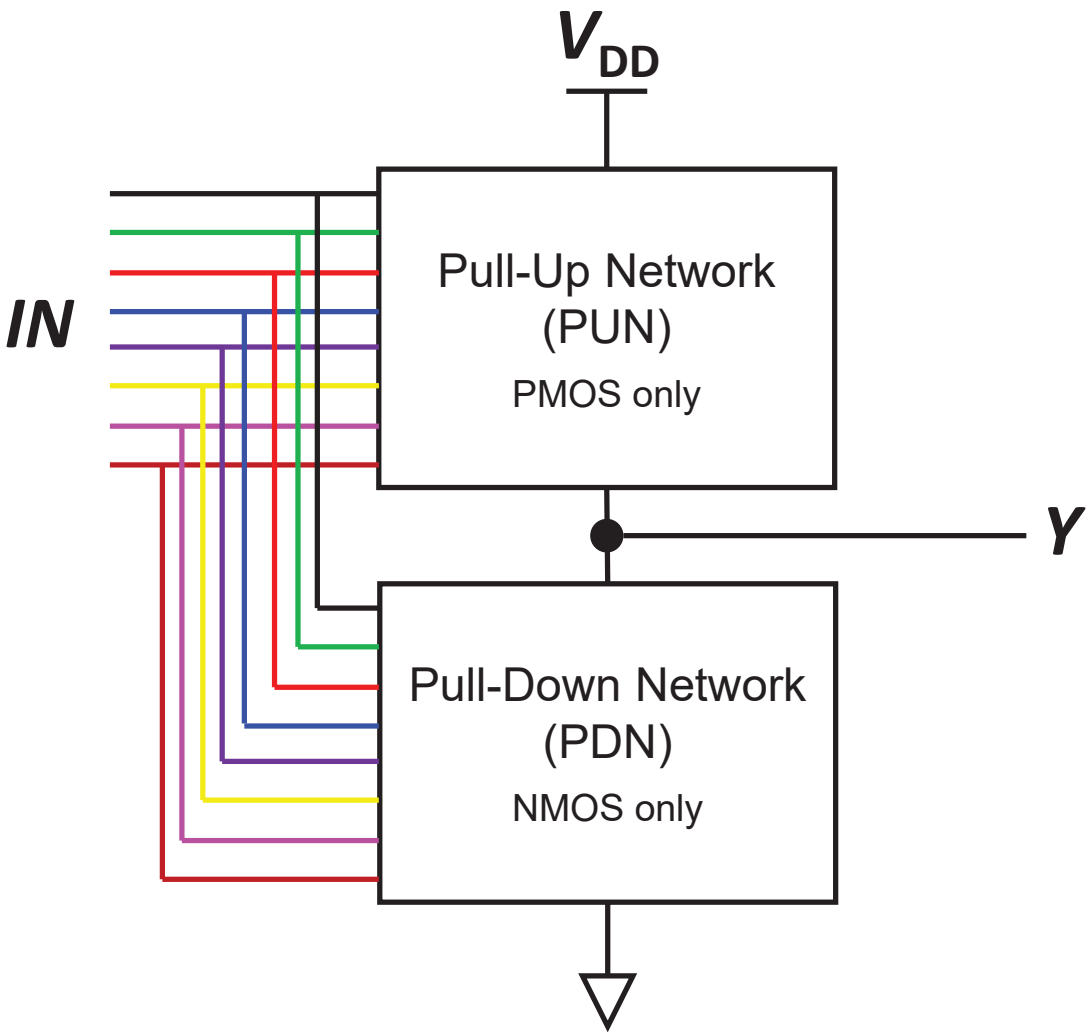
Introduction to the Static CMOS Logic Gate

Static CMOS Combinational Logic Gates



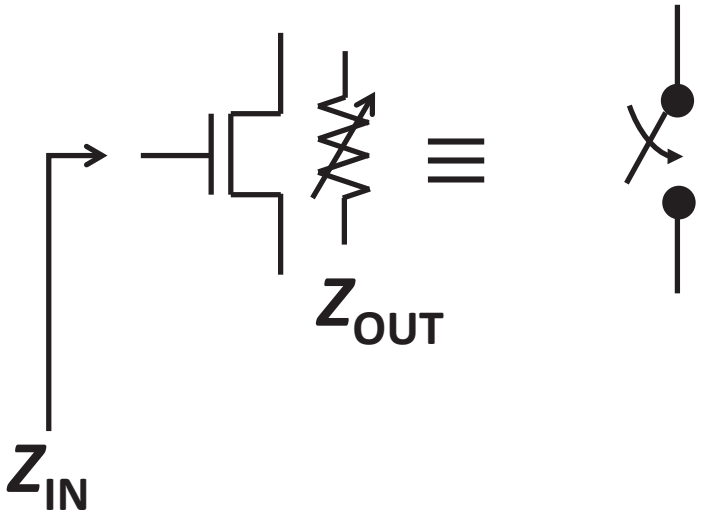
- Use MOSFETs as digital switches
  - NMOS: normally-open switch
  - PMOS: normally-closed switch
  - "normally": when gate is '0'
- Static CMOS Logic Gate
  - Design PDN such that ***Y*** is connected to **GND** when the inputs to the logic gate results output of '0' at its output
    - No path to **GND** otherwise
  - Design PUN such that ***Y*** is connected to $\boldsymbol{V_{DD}}$ when the inputs to the logic gate results output of '1' at its output
    - No path to $\boldsymbol{V_{DD}}$ otherwise
  - No short circuit path through PDN and PUN at the same time
    - No short circuit between $\boldsymbol{V_{DD}}$ and **GND**

Designing Static CMOS Logic Gates



- Designing any logic gate requires:
  - Drawing the circuit schematic of the logic gate (transistors) corresponding to Register-Transfer Level (RTL)
  - Determining the parameters of all elements in the circuit (transistor "sizes", V_T, etc.)
  - Drawing layout (the real physical implementation of the logic gate)
  - Estimating/evaluating/analyzing relationships between design choices and metrics
    - Power analysis
      - Battery life, heat generation
    - Timing analysis (e.g. static timing analysis, STA)
      - Performance/speed
    - Layout analysis
      - Area, cost
  - Scientific method for comparing/justifying design choices
    - Make use of Figures of Merit (FoM)

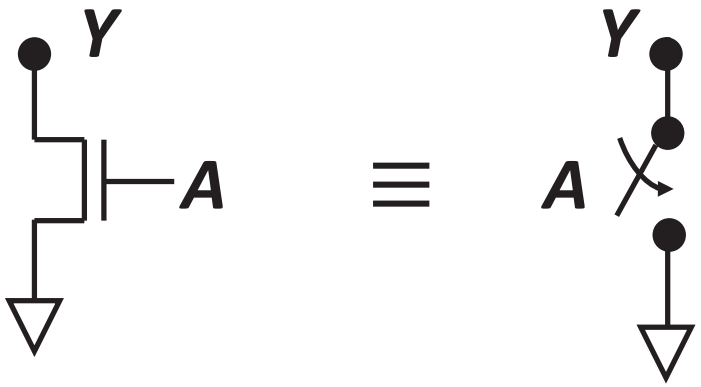
Transistors as Electronic (Digital) Switches



- Voltage on gate terminal of transistor controls resistance between its source and drain terminals
  - Low resistance = switch is ON or CLOSED
  - High resistance = switch is OFF or OPEN

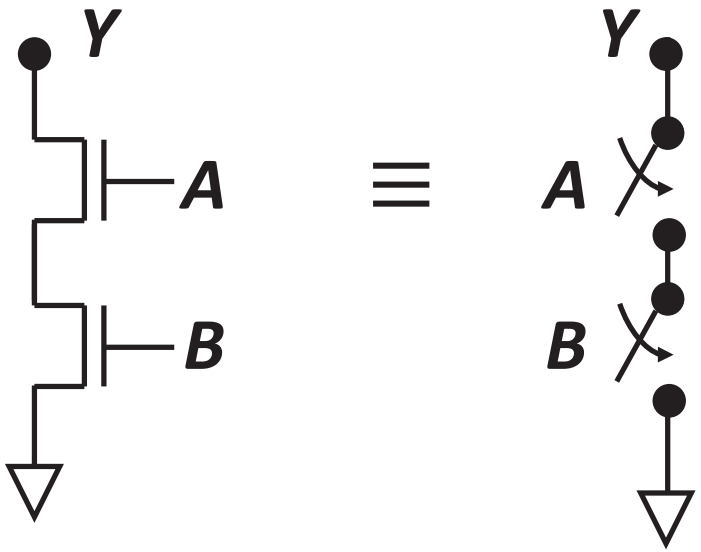
1. **Consider the transistor as an ideal switch to design the topology of the static CMOS combinational logic gate**
2. Take capacitances and channel resistance into consideration when determining propagation delays, current drive strength, and power dissipation
   - Tune by designing (W_{ch}/L_{ch}) ratios (mainly W_{ch}) of transistors

Implementing Logic using Switches



- When is ***Y*** = '0' (***V*****$_Y$** = **GND**)?
  - Switch is CLOSED
  - ***A*** = '1' (***V*****$_A$** = ***V*****$_{DD}$**)

Implementing Logic using Switches – Series Connection



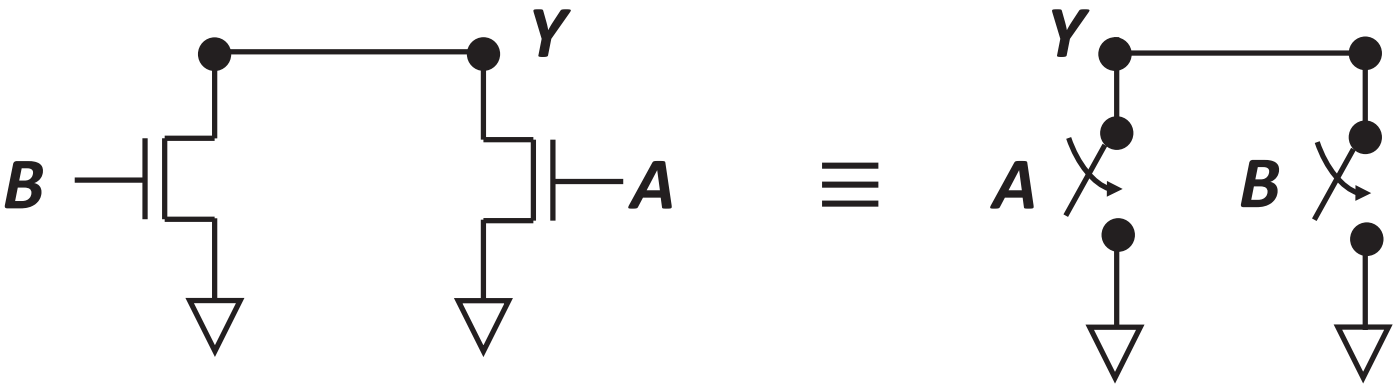
- When is ***Y*** = '0' (***V***$_Y$ = **GND**)?
  - Both switches are CLOSED
  - ***A*** AND ***B*** = '1'

Y = '0' ↔ ***A*** AND ***B*** = '1'

Like $Y = \overline{A.B}$

- ***Y*** is in high impedance (or high-***Z***) state if ANY switch is open
- Otherwise, PDN passes '0' to ***Y***

Implementing Logic using Switches – Parallel Connection



- When is ***Y*** = '0' (***V***$_Y$ = **GND**)?
  - Either switch is CLOSED
  - ***A*** OR ***B*** = '1'

Y = '0' ↔ ***A*** OR ***B*** = '1'

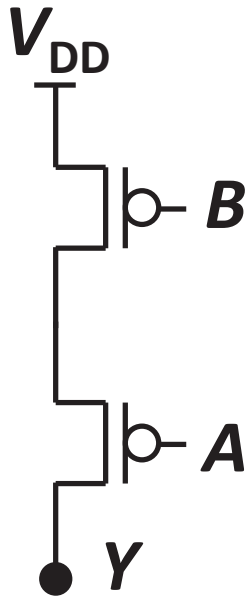
Like $Y = \overline{A + B}$

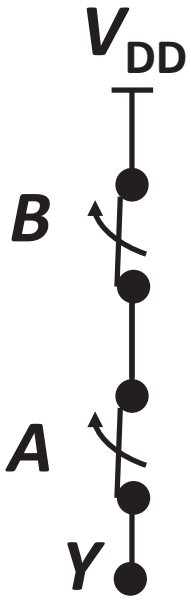
- ***Y*** is in high impedance (or high-***Z***) state if BOTH switches are open
- Otherwise, PDN passes '0' to ***Y***

Implementing Logic using PMOS Transistors

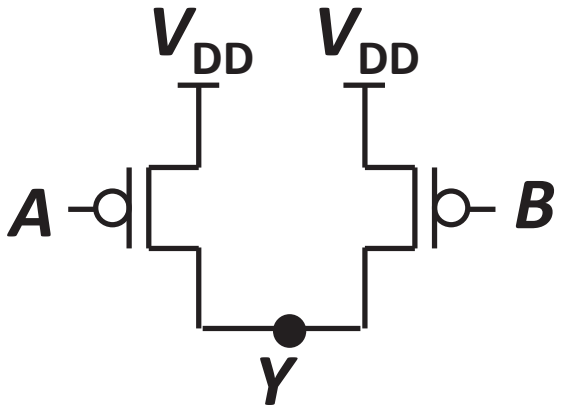
When is Y = '1' ($V_Y = V_{DD}$)?

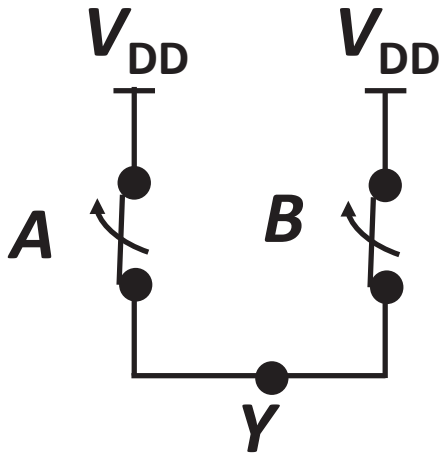
- '1' is passed to Y
- High-Z when otherwise





- Both switches are CLOSED
  - $\overline{A} \,.\, \overline{B}$ = '1'
  - $(\overline{A + B})$ = '1' (De Morgan)
- Same as parallel connected NMOS





- Either switch is CLOSED
  - $\overline{A} + \overline{B}$ = '1'
  - $(\overline{A.B})$ = '1' (De Morgan)
- Same as series connected NMOS

Week 3-2

Fundamental Design Concepts of the Static CMOS Logic Gate – The Schematic or Topology

Observation from Logic Truth Tables

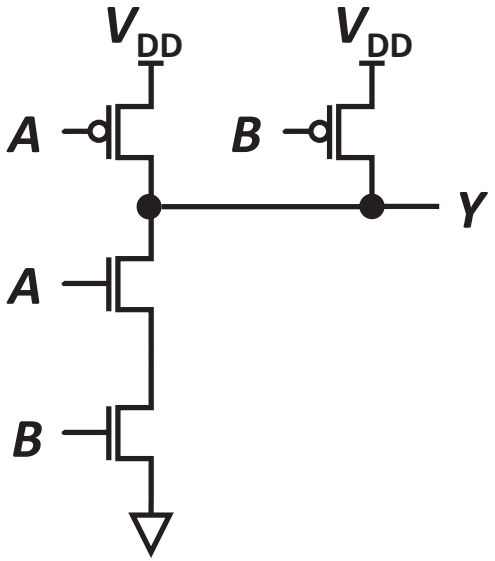
NAND		
A	***B***	Output, ***Y***
0	0	1
0	1	1
1	0	1
1	1	0

NOR		
A	***B***	Output, ***Y***
0	0	1
0	1	0
1	0	0
1	1	0

- For any combination of inputs, the output, ***Y***, must be either '0' or '1'
- Use pull-down network, PDN, consisting of only NMOS transistors to **pass '0' according to the truth table**
  - High-***Z*** state otherwise
- Use pull-up network, PUN, consisting of only PMOS transistors to **pass '1' according to the truth table**
  - High-***Z*** state otherwise
- When NMOS network pass '0', PMOS network pass high-***Z***: No conflict
- When PMOS network pass '1', NMOS network pass high-***Z***: No conflict

Example 1: 2-input NAND Gate

NAND		
A	***B***	Output, ***Y***
0	0	1
0	1	1
1	0	1
1	1	0



Condition for when PDN passes '0': $A \cdot B$

Condition for when PUN passes '1': $\bar{A} + \bar{B}$

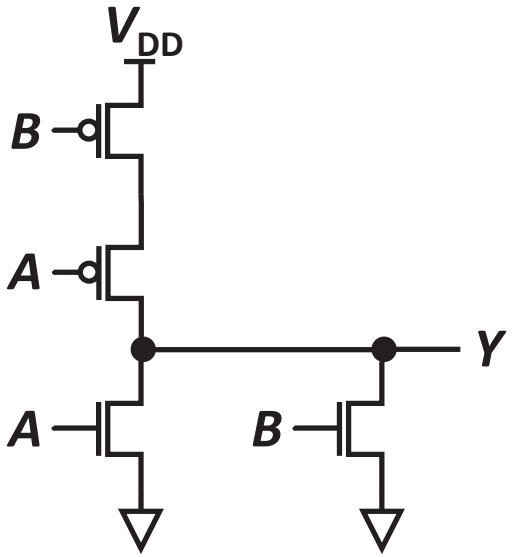
Apply De Morgan's Theorem to Boolean function of PDN:

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

PUN is dual network of PDN

Example 2: 2-input NOR Gate

NOR		
A	***B***	Output, ***Y***
0	0	1
0	1	0
1	0	0
1	1	0



Condition for when PDN passes '0': $A + B$

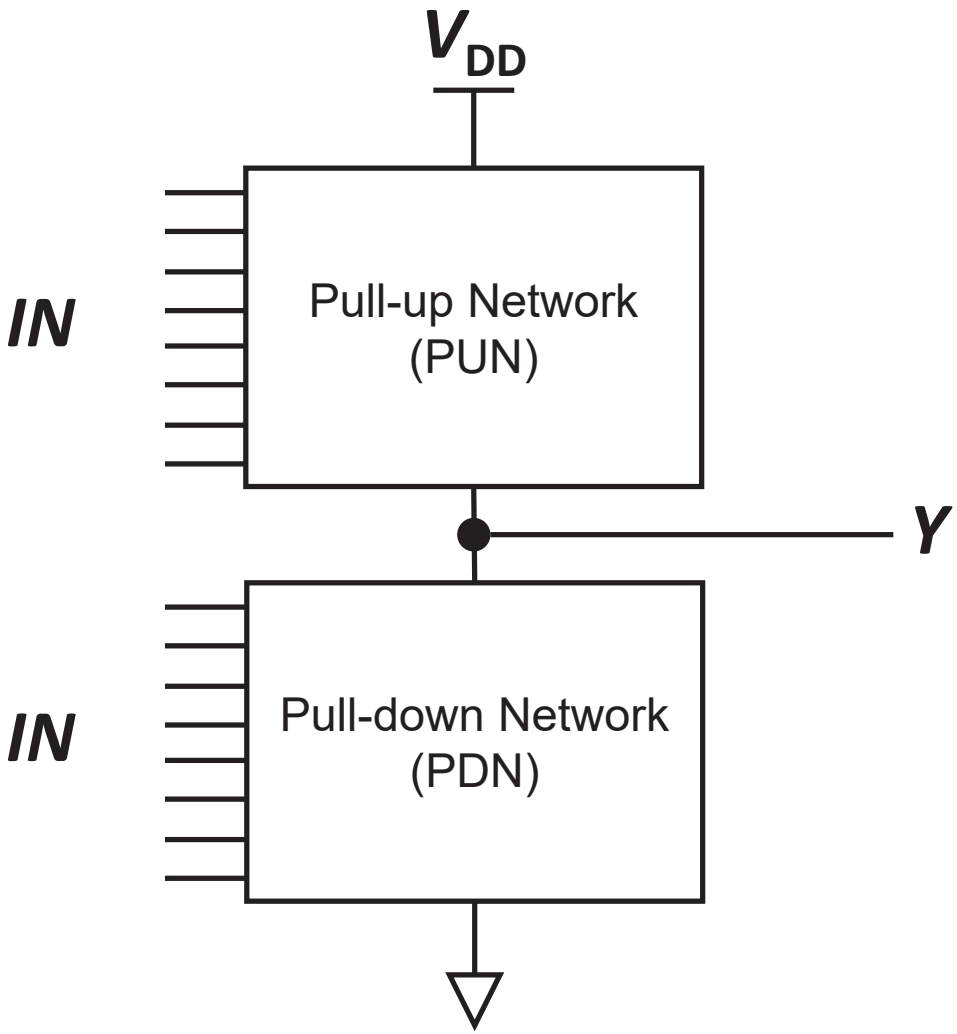
Condition for when PUN passes '1': $\bar{A} \cdot \bar{B}$

Apply De Morgan's Theorem to Boolean function of PDN:

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

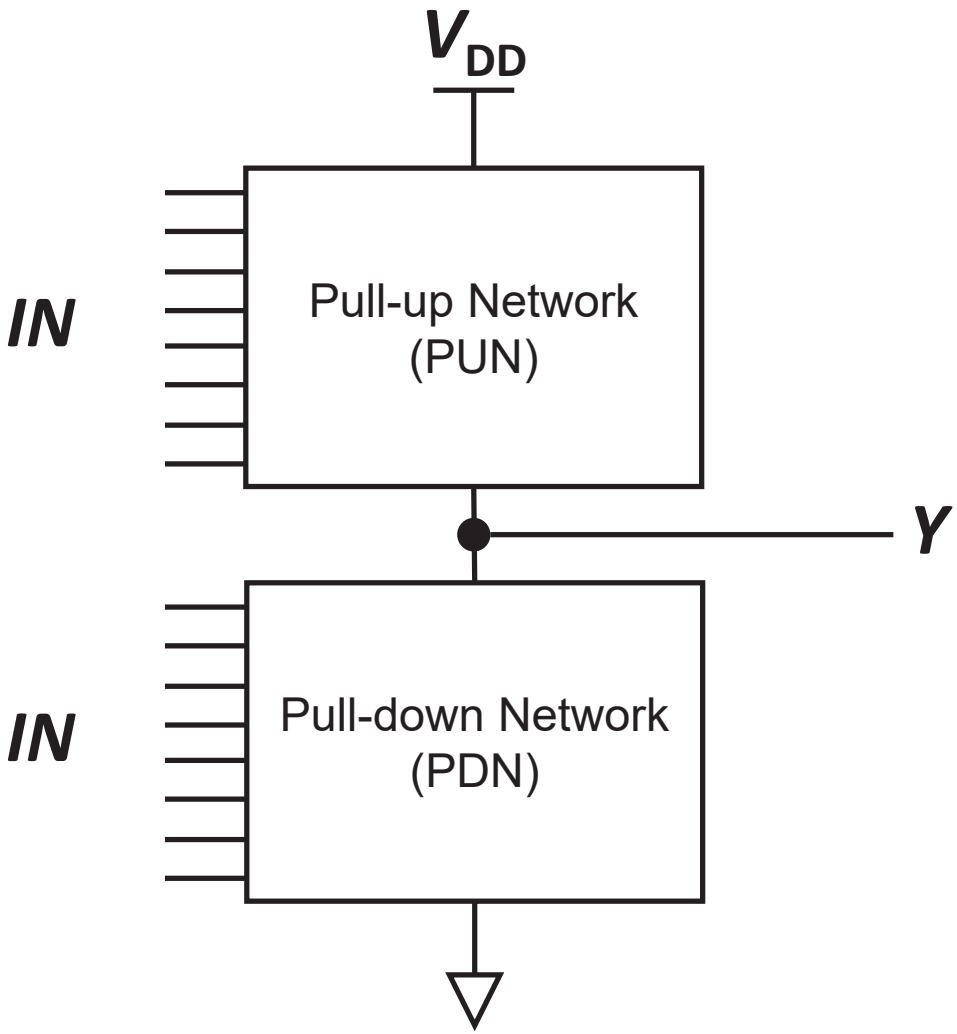
PUN is dual network of PDN

General Method to Design Topology of Static Logic Gates



- Implement PDN based on logic function:
  - $\bar{Y} = f(IN)$
- Implement the PUN (the dual of PDN)
  - $Y = f^{DUAL}(IN)$
- Connect outputs of PDN and PUN together
- Only one set of inputs:
  - Every input will be connected to one n-MOSFET and one p-MOSFET
- Only one of PDN or PUN will short circuit node ***Y*** to the corresponding power rail

Characteristics of Static Logic Gates



- As long as power is supplied to the logic gate, and all the inputs are at held voltages corresponding to legal logic values:
  - The output voltage will be either **GND** or V_{DD}
  - The output voltage will correspond to the logic value of the Boolean function acting on the inputs to the logic gate
  - The output voltage will only be between **GND** and V_{DD} during switching
- Logic is inverting
  - NAND, NOR, NOT
  - AND
    - NAND followed by NOT
  - OR
    - NOR followed by NOT
- If ***N*** is the number of inputs to the static CMOS logic gate, the number of transistors in the logic gate is 2***N***

Week 3-3

An Example of Designing the Schematic of a Static CMOS Logic Gate

Example 1: $Z = \overline{C \cdot (A + B) + D}$

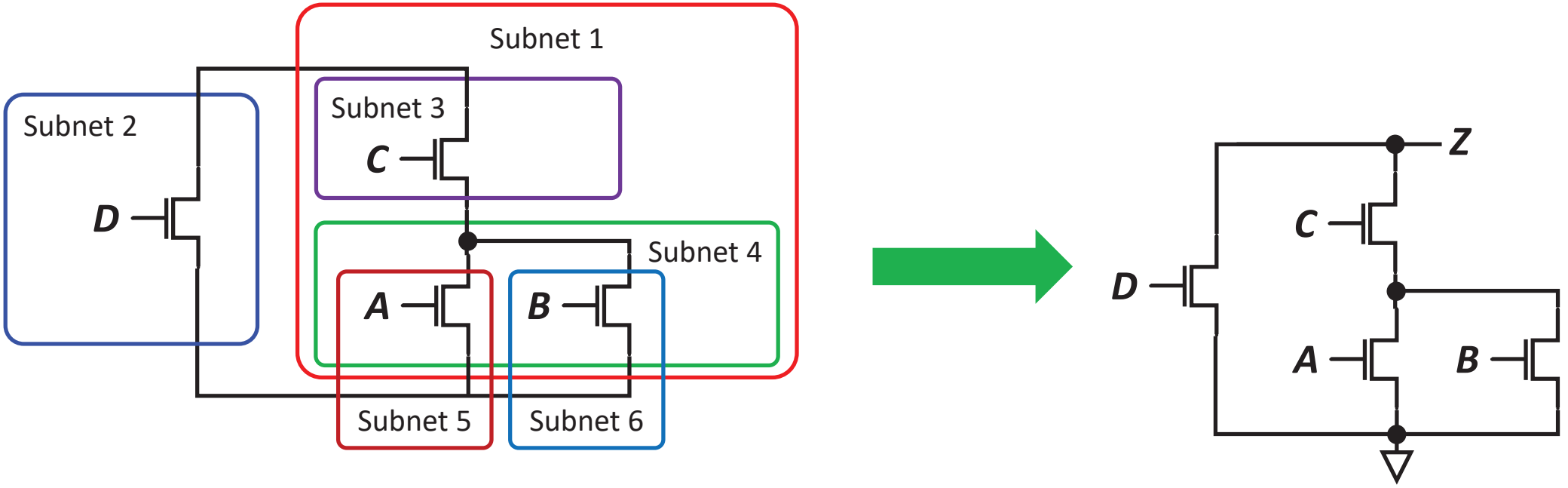
1. Draw the schematic of the PDN

$$\bar{Z} = C \cdot (A + B) + D$$

Subnet 1: $C \cdot (A + B)$ is connected in parallel with Subnet 2: D

Subnet 3: C is connected in series with Subnet 4: $A + B$

Subnet 5: A is connected in parallel with Subnet 6: B



Example 1: $Z = \overline{C \cdot (A + B) + D}$

2. Draw the schmeatic of the PUN

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

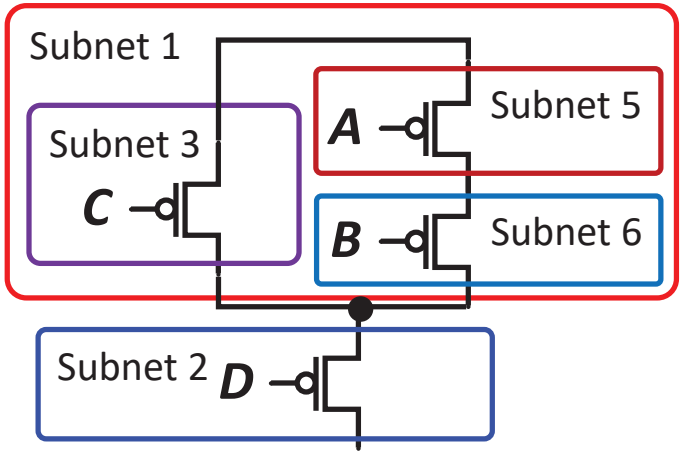
Apply De Morgan's Theorem:

$$Z = \overline{C \cdot (A + B)} \cdot \bar{D} = \left(\bar{C} + \overline{(A + B)}\right) \cdot \bar{D} = (\bar{C} + \bar{A} \cdot \bar{B}) \cdot \bar{D}$$

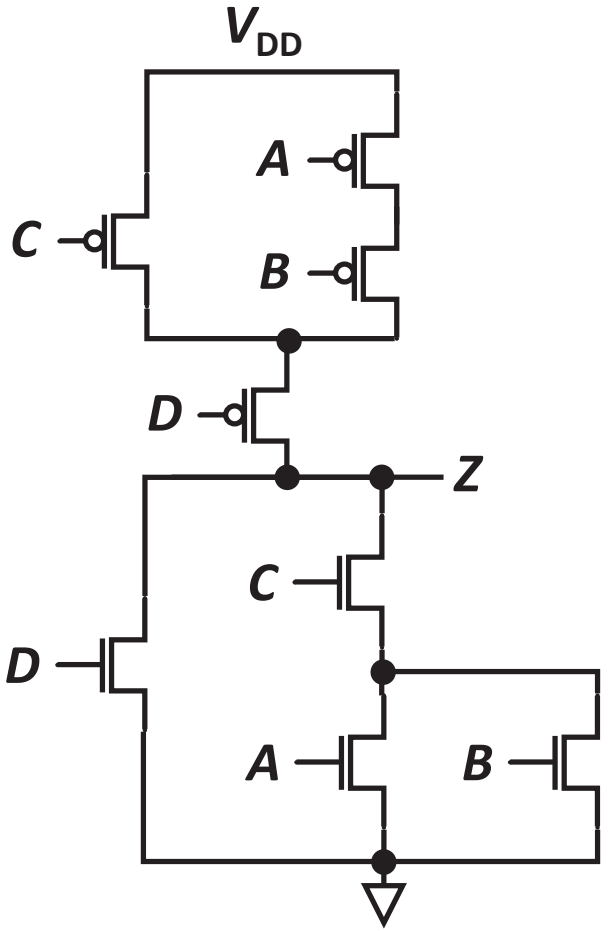
Subnet 1: $\bar{C} + \bar{A} \cdot \bar{B}$ is connected in series with Subnet 2: $\bar{D}$

Subnet 3: $\bar{C}$ is connected in parallel with Subnet 4: $\bar{A} \cdot \bar{B}$

Subnet 5: $\bar{A}$ is connected in series with Subnet 6: $\bar{B}$



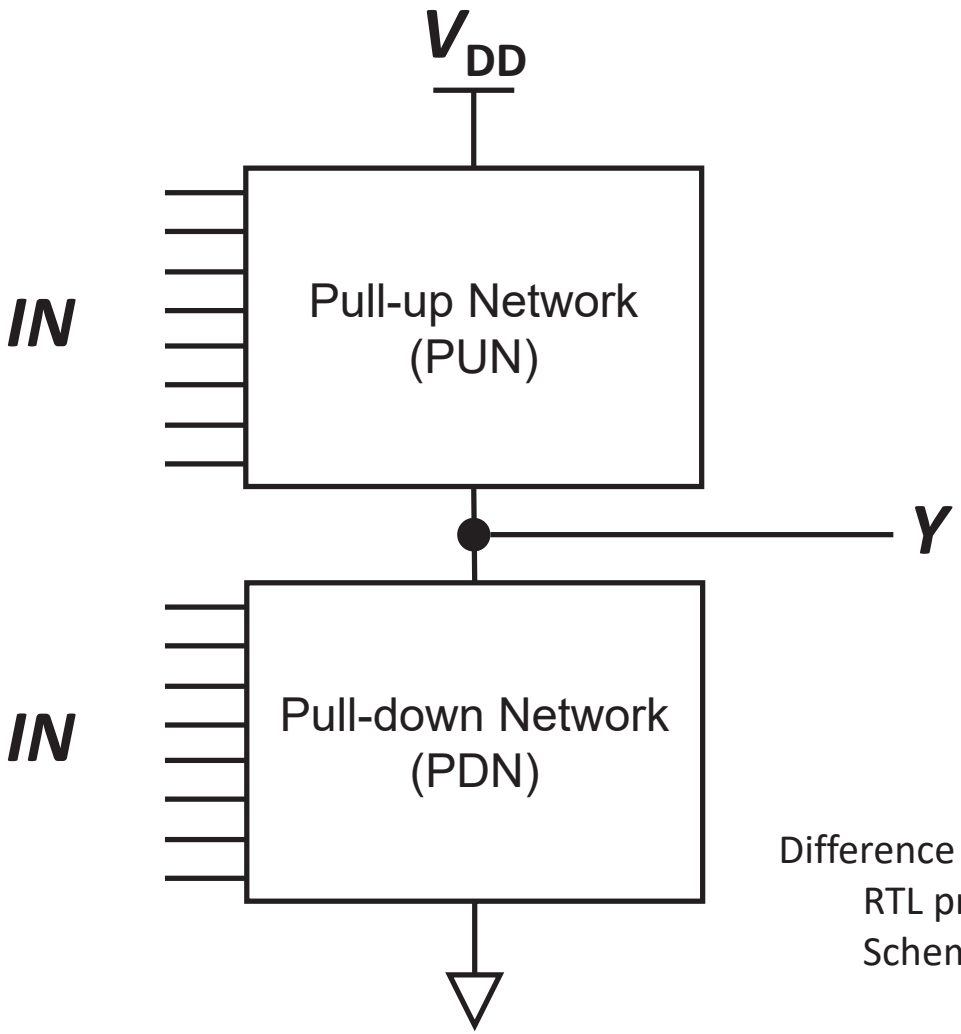
PUN and PDN are dual networks
- Same number of transistors
- PMOS ⇔ NMOS
- Connection of subnets:
  - Series ⇔ Parallel



Week 3-4

Fundamental Design Concepts of the Static CMOS Logic Gate – Transistor Sizing

Designing Static CMOS Logic Gates



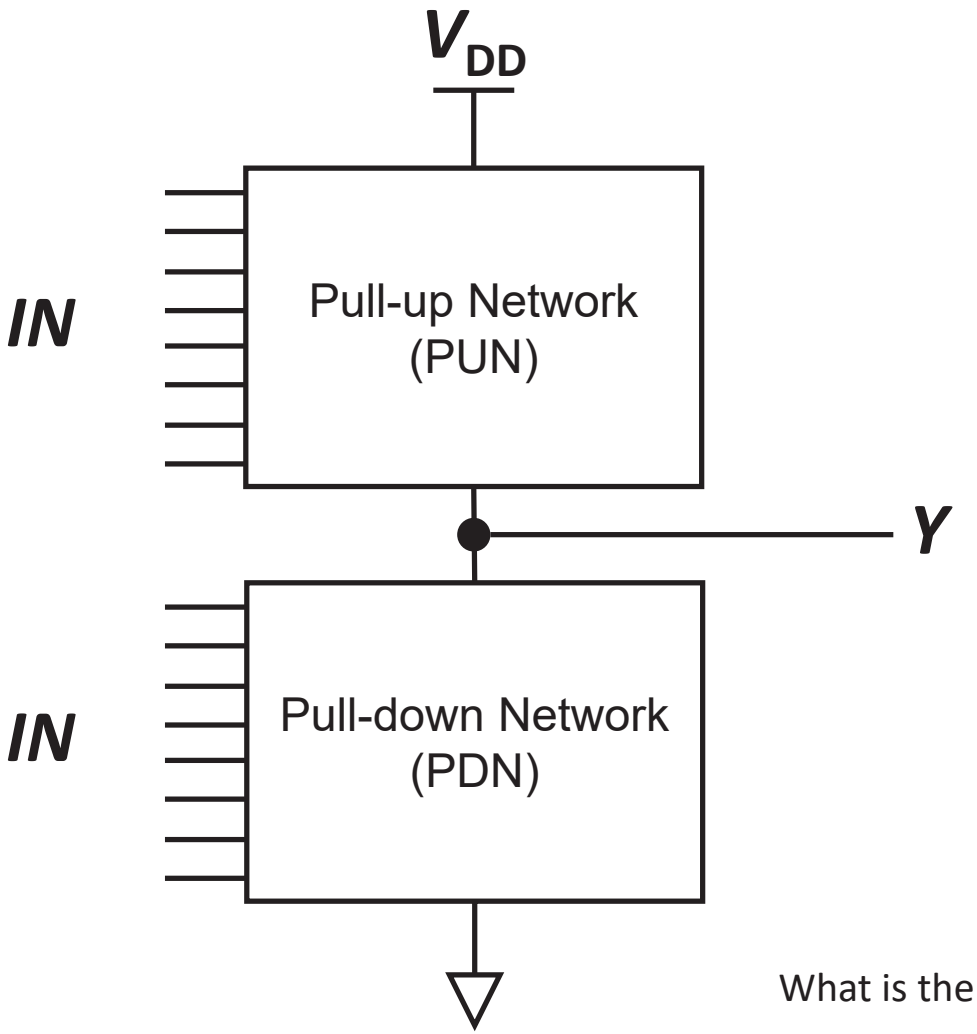
- Designing any logic gate requires:
  - Drawing the circuit schematic of the logic gate (transistors) corresponding to Register-Transfer Level (RTL)
  - Determining the parameters of all elements in the circuit (transistor "sizes", V_T, etc.)
  - Drawing layout (the real physical implementation of the logic gate)
  - Estimating/evaluating/analyzing relationships between design choices and metrics
    - Power analysis
      - Battery life, heat generation
    - Timing analysis (e.g. static timing analysis, STA)
      - Performance/speed
    - Layout analysis
      - Area, cost

Difference between RTL and schematic:

RTL provides a functional and logical description (like a C program)

Schematic is a physical implementation that can be manufactured

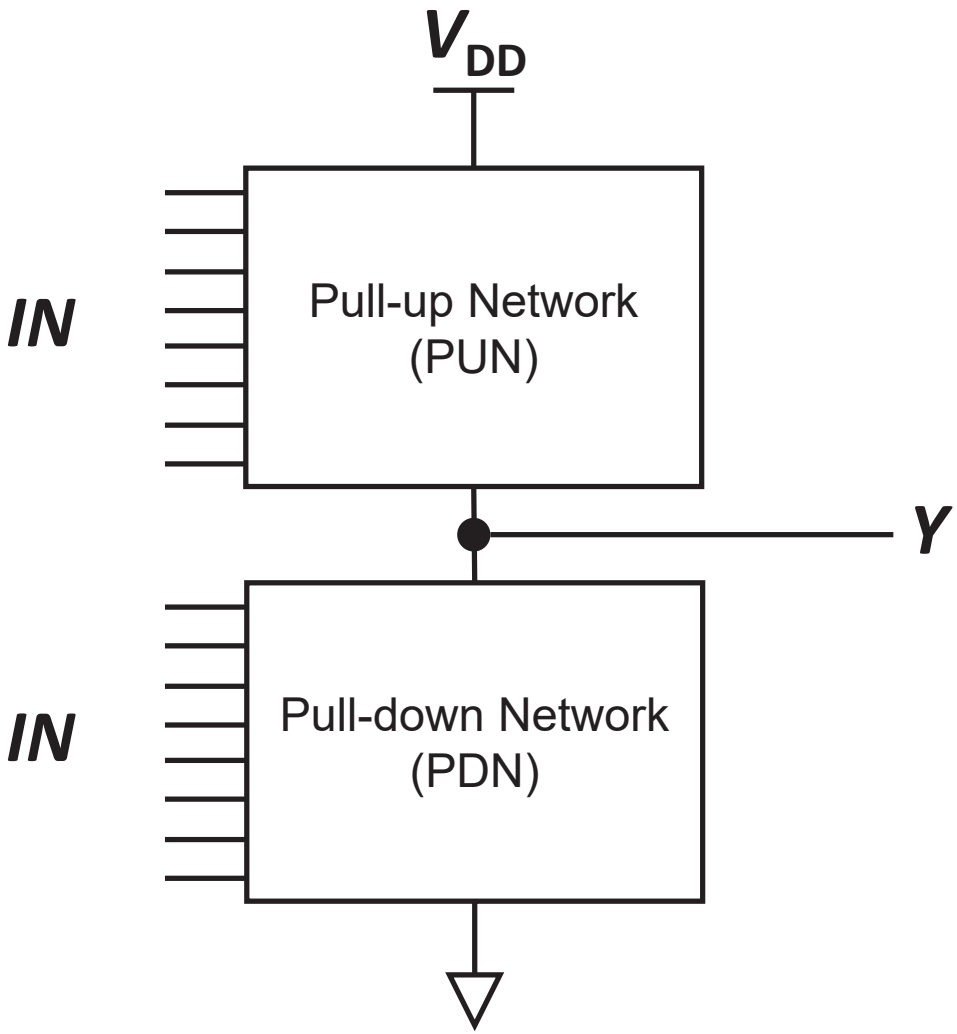
General Method to Size Transistors of Static Logic Gates



1. Identify ALL possible short-circuit paths through PDN and sort them by the number of n-MOSFETs (in descending order)
2. Starting with paths having the greatest number of n-MOSFETs, assign sizes to all transistors along the path
   - Once an n-MOSFET is assigned with a size, it will not be changed in the next steps
3. Reiterate Step #2 from paths with greatest number of n-MOSFETs to the fewest, until all n-MOSFETs have been assigned a size
   - Once a transistor is assigned with a size, it will not be changed in the next steps
4. Reiterate from Step #1 for PUN and work on p-MOSFETs until all p-MOSFETs have been assigned a size

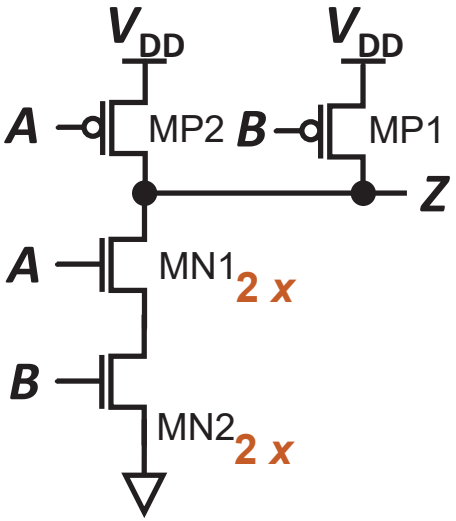
What is the rule for assigning size to a transistor?

Assigning Size to Transistor



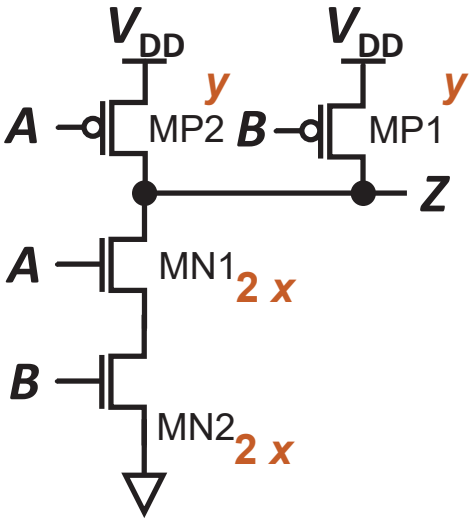
- The most common approach:
  - Match R_{EQ} of the short-circuit path to a reference R_{REF}
    - Typical reference circuit: smallest static CMOS inverter
    - For a transistor:
      - $R_{ON} \propto 1/I_{ON}$
      - $I_{ON} \propto (W/L)$
      - $R_{ON} \propto 1/(W/L)$
  - Reference circuit provides a benchmark for the drive strength and delay of a logic gate
    - This method assigns sizes to transistors in the circuit so that the logic gate will have the same drive strength and delay as the benchmark circuit
  - Simplifies static timing analysis

Example 1: $Z = \overline{A \cdot B}$



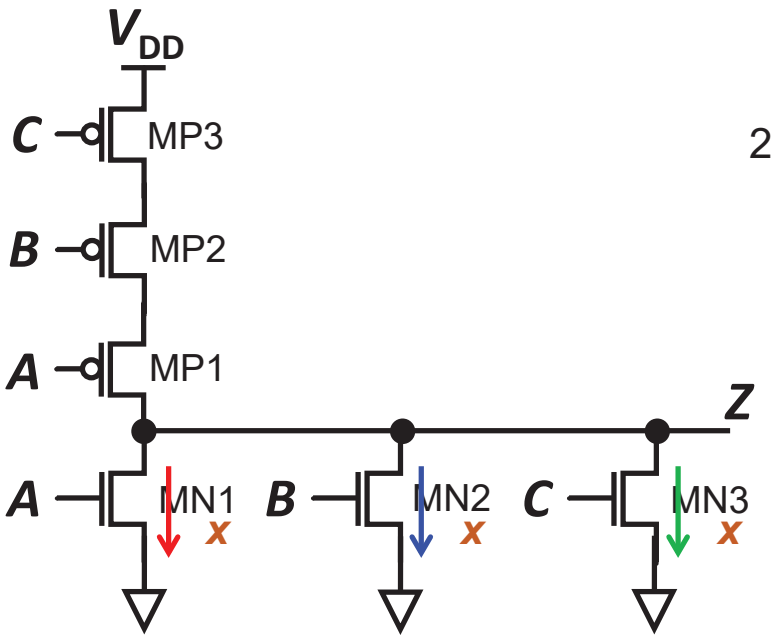
1. Identify all possible short-circuit paths between output and GND through PDN and sort in descending order of number of n-MOSFETs
   - MN1-MN2
2. Assign sizes to n-MOSFETs such that $\boldsymbol{R_{EQ}}$ of all paths are the same as that for an n-MOSFET having (W/L) = $(W/L)_{REF,N}$, starting from the longest path
   - Let resistance of n-MOSFET having (W/L) = $(W/L)_{REF,N}$ = $\boldsymbol{x}$ be $\boldsymbol{R_N}$
   - If both n-MOSFETs along MN1-MN2 has (W/L) = $\boldsymbol{x}$, then each n-MOSFET has resistance of $\boldsymbol{R_N}$, and the total resistance along the path is $\boldsymbol{R_N} + \boldsymbol{R_N} = 2\boldsymbol{R_N}$
   - If we assign MN1 and MN2 with (W/L) = 2 $\boldsymbol{x}$, the total resistance along the path is $0.5\,\boldsymbol{R_N} + 0.5\,\boldsymbol{R_N} = \boldsymbol{R_N}$
   - Thus, MN1 and MN2 should be assigned (W/L) = 2 $\boldsymbol{x}$
   - A rule of thumb: when sizing transistors in a path, *assign same size to transistors that have not been assigned a size yet*

Example 1: $Z = \overline{A \cdot B}$



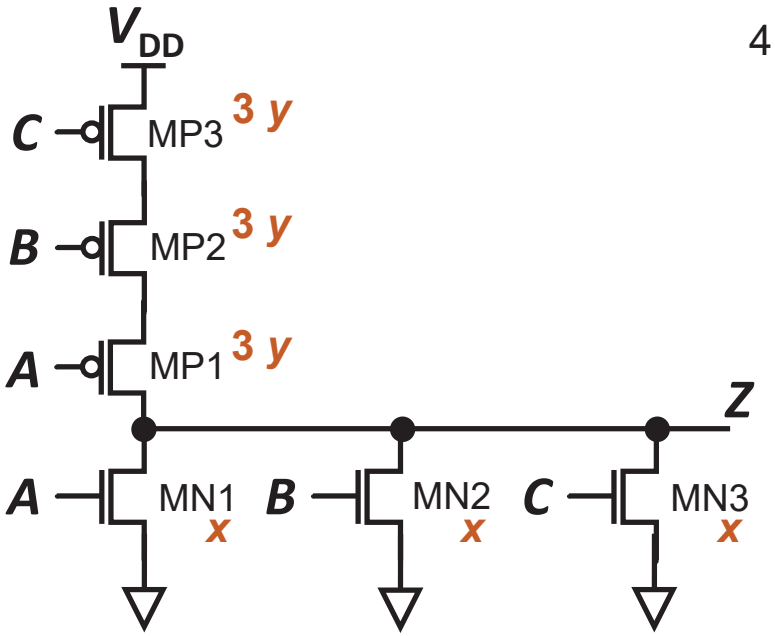
3. Identify all possible short-circuit paths between output and V_{DD} through PUN and sort in descending order of number of p-MOSFETs
   - MP1
   - MP2
4. Assign sizes to p-MOSFETs such that R_{EQ} of all paths are the same as that for an p-MOSFET having (W/L) = $(W/L)_{REF,P}$, starting from the longest path
   - Let resistance of p-MOSFET having (W/L) = $(W/L)_{REF,P}$ = y be R_P
   - If MP1 have (W/L) = y, then the total resistance along the path is R_P, which is the same as the reference
   - Thus, MP1 should be assigned (W/L) = y
   - A rule of thumb: transistors in a path that has only one transistor should *have the same size as the reference*

Example 2: $Z = \overline{A + B + C}$



1. Identify all possible short-circuit paths between output and GND through PDN and sort in descending order of number of n-MOSFETs
   - MN1
   - MN2
   - MN3
2. Assign sizes to n-MOSFETs such that $\boldsymbol{R_{EQ}}$ of all paths are the same as that for an n-MOSFET having (W/L) = $(W/L)_{REF,N}$, starting from the longest path
   - Let resistance of n-MOSFET having (W/L) = $(W/L)_{REF,N}$ = $\boldsymbol{x}$ be $\boldsymbol{R_N}$
   - Transistors in a path that has only one transistor should *have the same size as the reference*
   - Thus, MP1 should be assigned (W/L) = $\boldsymbol{x}$

Example 2: $Z = \overline{A + B + C}$

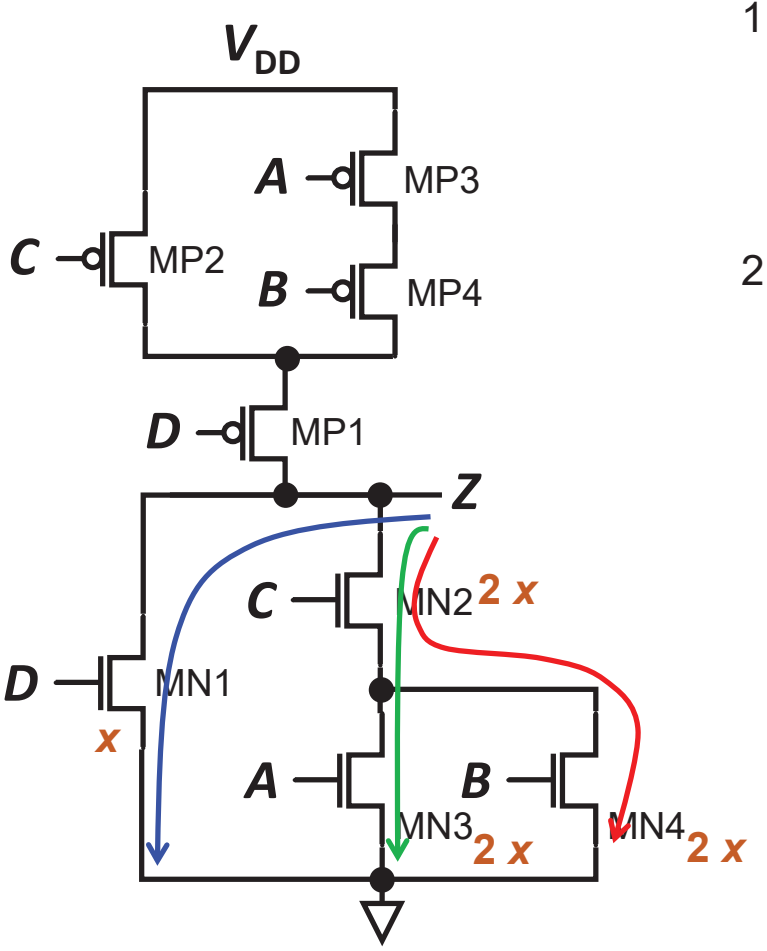


3. Identify all possible short-circuit paths between output and V_{DD} through PUN and sort in descending order of number of p-MOSFETs
   - MP1-MP2-MP3
4. Assign sizes to p-MOSFETs such that R_{EQ} of all paths are the same as that for an p-MOSFET having (W/L) = $(W/L)_{REF,P}$, starting from the longest path
   - Let resistance of p-MOSFET having (W/L) = $(W/L)_{REF,P}$ = y be R_P
   - When sizing transistors in a path, *assign same size to transistors that have not been assigned a size yet*
   - Stack of N transistors with size y has $R_{EQ} = N R_P$
   - Size all transistors in stack to have (W/L) = $N y$
   - $N = 3 \Rightarrow$ assign (W/L) = $3y$
   - Check: resistance of pull-up path
     - $R_P/3 + R_P/3 + R_P/3 = R_P$ (same as reference ✓)

Week 3-5

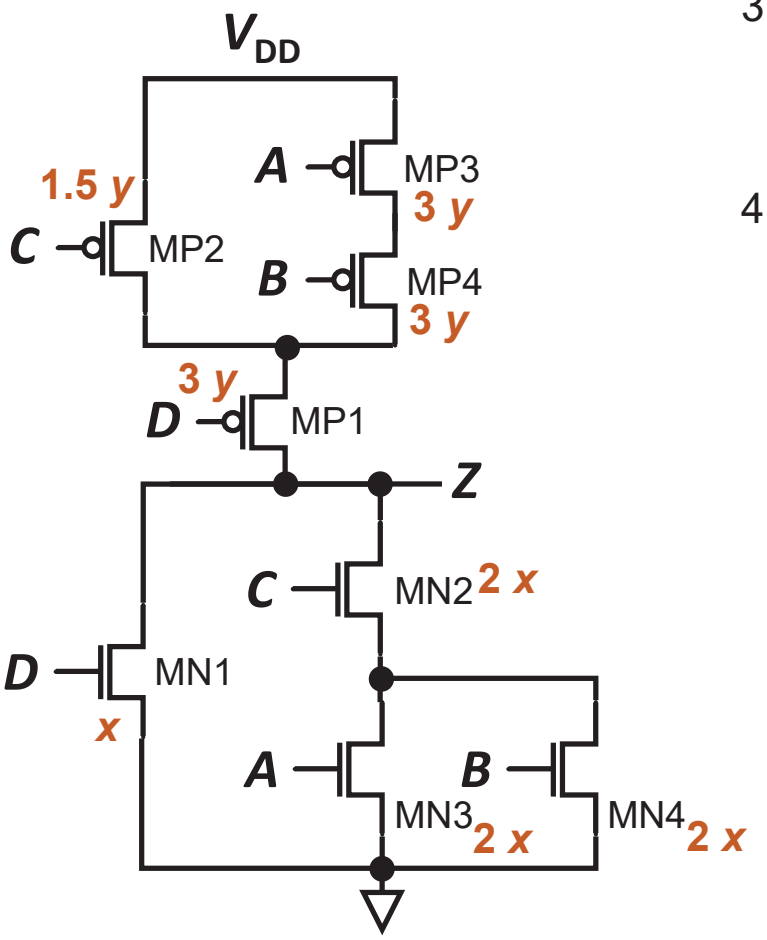
An Example of Assigning Transistor Sizes for a Static CMOS Logic Gate

Example 1: $Z = \overline{C \cdot (A + B) + D}$



1. Identify all possible short-circuit paths between output and GND through PDN and sort in descending order of number of n-MOSFETs
   - MN2-MN4
   - MN2-MN3
   - MN1
2. Assign sizes to n-MOSFETs such that R_{EQ} of all paths are the same as that for an n-MOSFET having (W/L) = $(W/L)_{REF,N}$, starting from the longest path
   - Let resistance of n-MOSFET having (W/L) = $(W/L)_{REF,N}$ = x be R_N
   - If both n-MOSFETs along MN2-MN4 has (W/L) = x, then each n-MOSFET has resistance of R_N, and the total resistance along the path is $R_N + R_N = 2R_N$
   - If we assign MN2 and MN4 with (W/L) = 2 x, the total resistance along the path is $0.5\,R_N + 0.5\,R_N = R_N$
   - Thus, MN2 and MN4 should be assigned (W/L) = 2 x
   - In MN2-MN3, only MN3 does not have a size
     - Need resistance of MN3 to be $R_N - 0.5\,R_N = 0.5\,R_N$
     - Hence, MN3 should be assigned (W/L) = 2 x
   - In all paths having a single transistor, assign (W/L) = x
     - Hence, MN1 should be assigned (W/L) = x

Example 1: $Z = \overline{C \cdot (A + B) + D}$

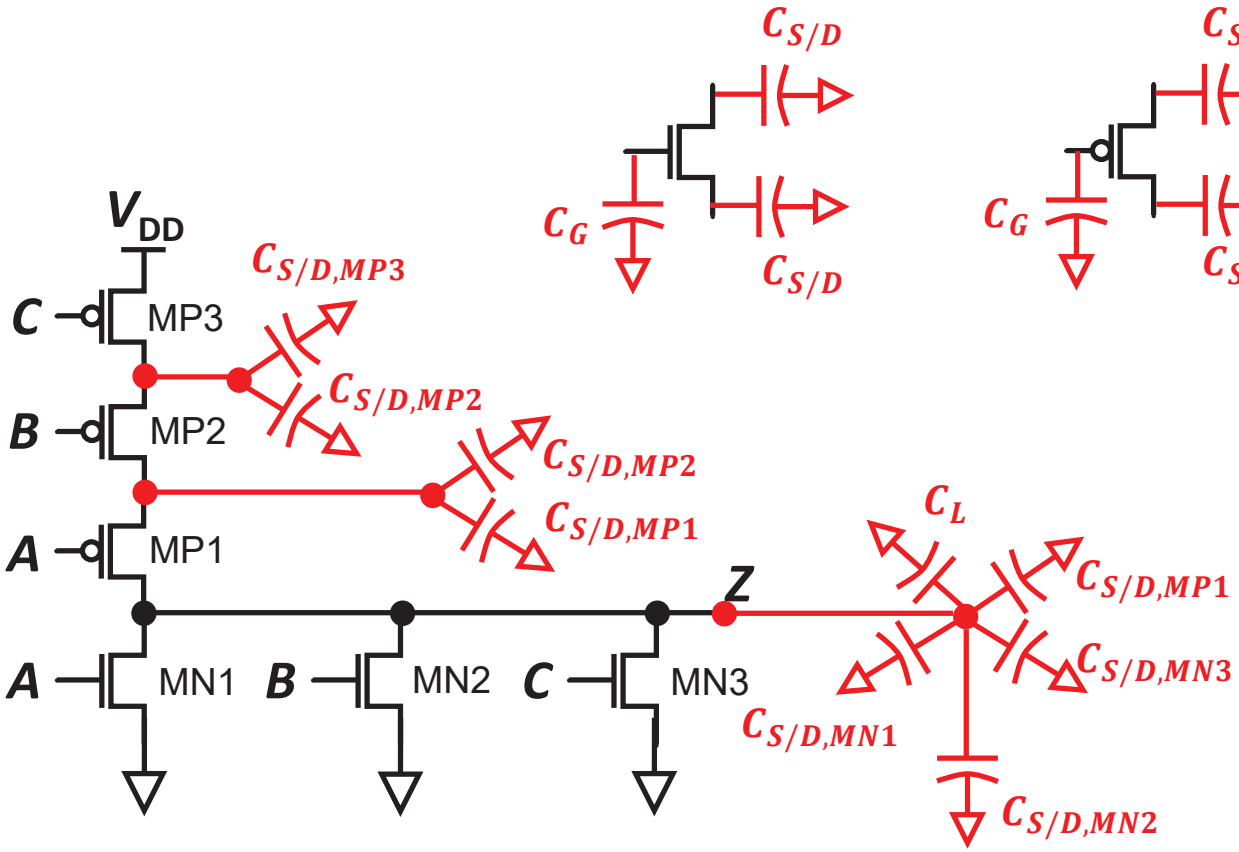


3. Identify all possible short-circuit paths between output and V_{DD} through PUN and sort in descending order of number of p-MOSFETs
   - MP1-MP4-MP3
   - MP1-MP2
4. Assign sizes to p-MOSFETs such that R_{EQ} of all paths are the same as that for an p-MOSFET having (W/L) = $(W/L)_{REF,P}$, starting from the longest path
   - Let resistance of p-MOSFET having (W/L) = $(W/L)_{REF,P}$ = y be R_P
   - If all p-MOSFETs along MP1-MP4-MP3 have (W/L) = y, then each p-MOSFET has resistance of R_P, and the total resistance along the path is $R_P + R_P + R_P = 3R_P$
   - If we assign MP1, MP3 and MP4 with (W/L) = 3 y, the total resistance along the path is $R_P/3 + R_P/3 + R_P/3 = R_P$
   - Thus, MP1, MP3 and MP4 should be assigned (W/L) = 3 y
   - In MP1-MP2, only MP2 does not have a size
     - Need resistance of MP2 to be $R_P - R_P/3 = 2R_P/3$
     - Hence, MP2 should be assigned (W/L) = (3/2) y = 1.5 y

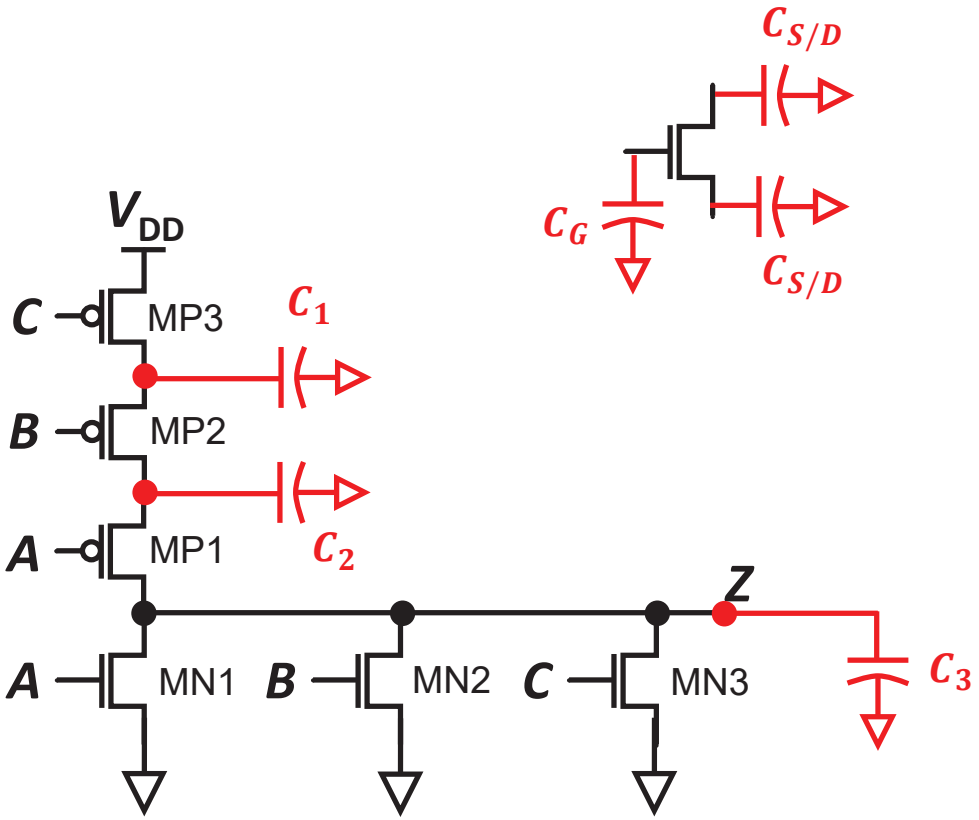
Week 3-6

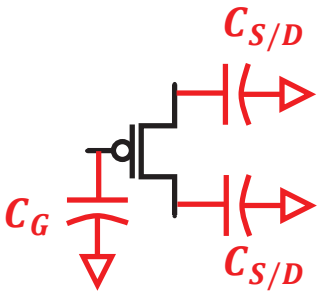
The Elmore Delay Model for Propagation Delays of Static CMOS Logic Gates

Transient Response in Digital Circuits

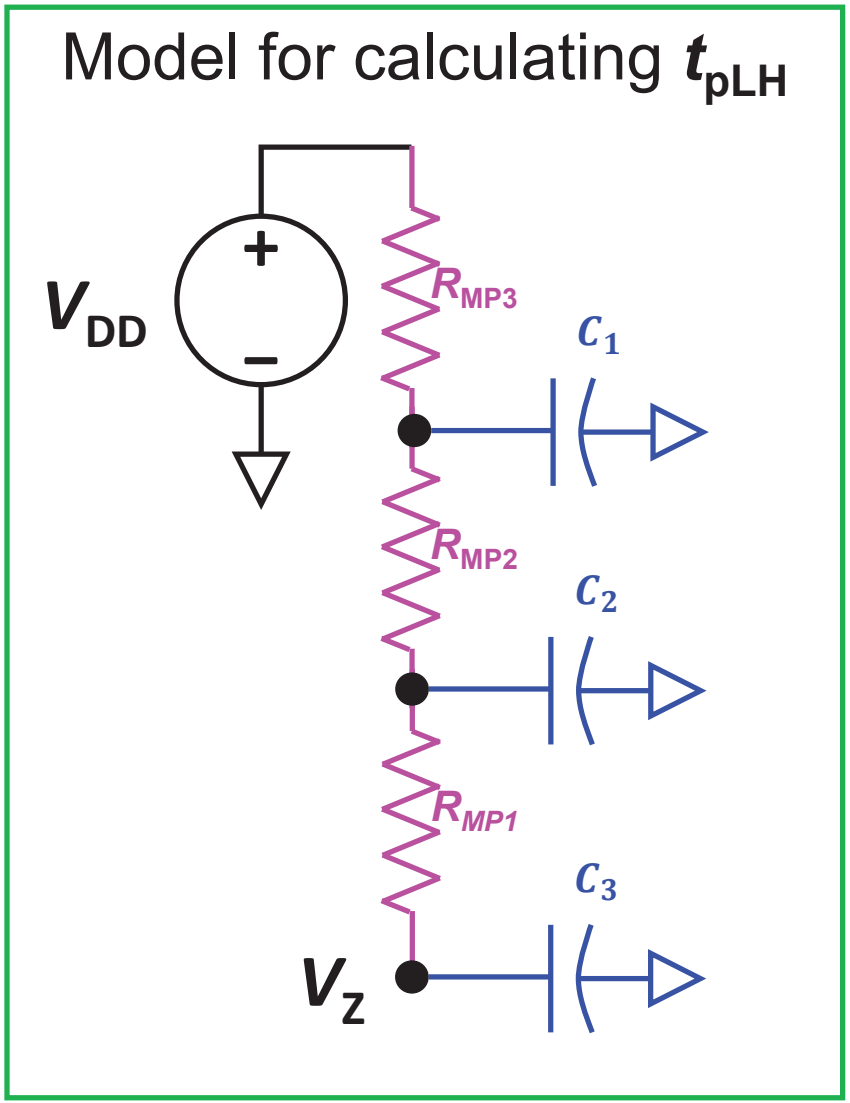


Transient Response in Digital Circuits



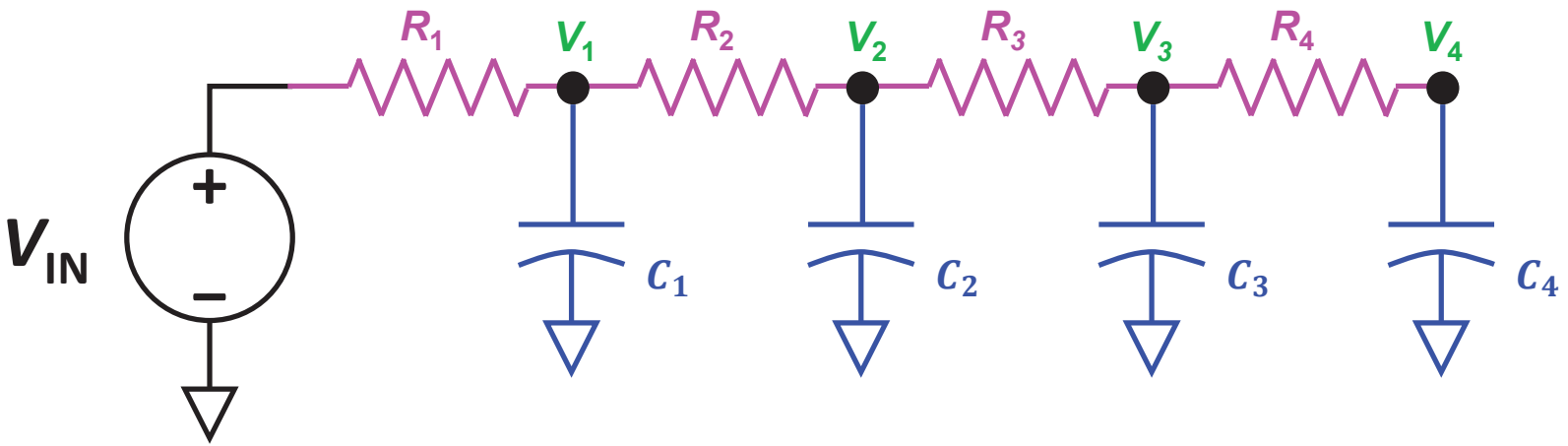


$$C_1 = C_{S/D,MP3} + C_{S/D,MP2}$$
$$C_2 = C_{S/D,MP2} + C_{S/D,MP1}$$
$$C_3 = C_{S/D,MP1} + C_{S/D,MN1} + C_{S/D,MN2} + C_{S/D,MN3} + C_L$$



Delay Estimation for Higher Order RC Circuits

- Equivalent circuits can be more complicated than 1st order RC circuit



$$(V_3 - V_4)/R_4 = C_4 \frac{dV_4}{dt}$$

$$(V_1 - V_2)/R_2 - (V_2 - V_3)/R_3 = C_2 \frac{dV_2}{dt}$$

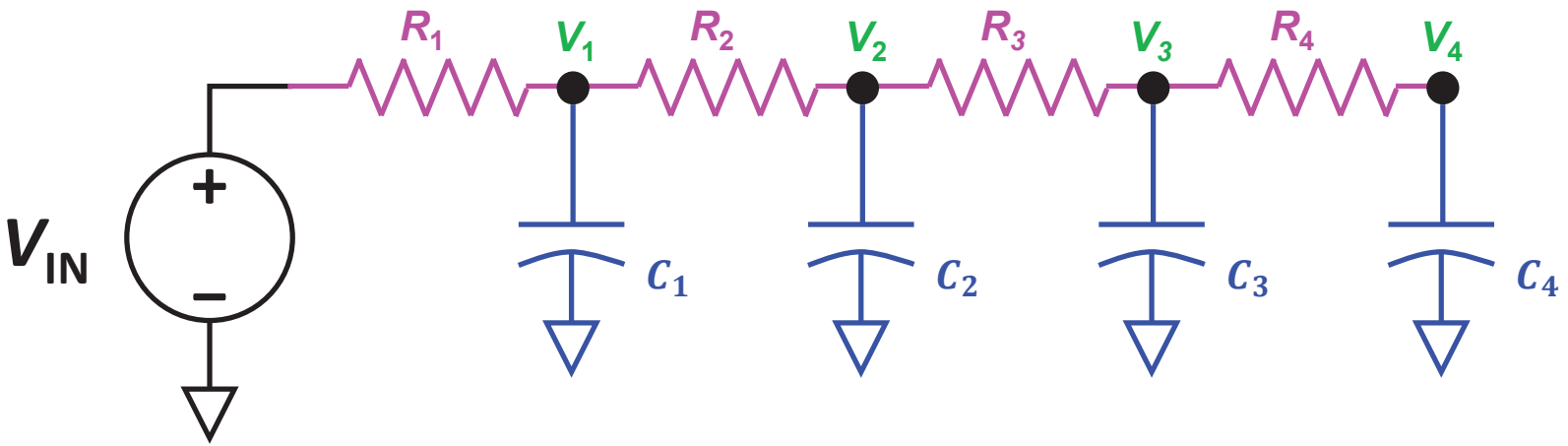
$$(V_2 - V_3)/R_3 - (V_3 - V_4)/R_4 = C_3 \frac{dV_3}{dt}$$

$$(V_{IN} - V_1)/R_1 - (V_1 - V_2)/R_2 = C_1 \frac{dV_1}{dt}$$

Need to solve ALL equations simultaneously!

The Elmore Delay Model

- Approximate total delay using smaller delays



$$\tau_1 = 0.69R_1C_1 \qquad \tau_3 = 0.69(R_1 + R_2 + R_3)C_3$$

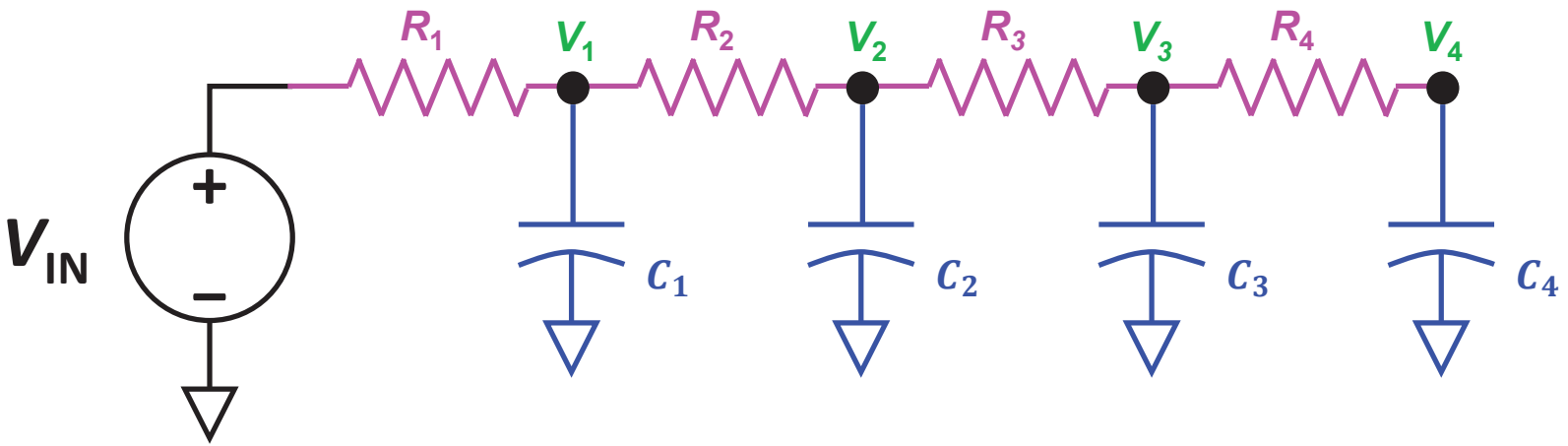
$$\tau_2 = 0.69(R_1 + R_2)C_2 \qquad \tau_4 = 0.69(R_1 + R_2 + R_3 + R_4)C_4$$

$$\tau_{4,actual} \approx \tau_1 + \tau_2 + \tau_3 + \tau_4$$

$$\tau_{4,actual} \approx 0.69[R_1(C_1 + C_2 + C_3 + C_4) + R_2(C_2 + C_3 + C_4) + R_3(C_3 + C_4) + R_4C_4]$$

The Elmore Delay Model

- Approximate total delay using smaller delays



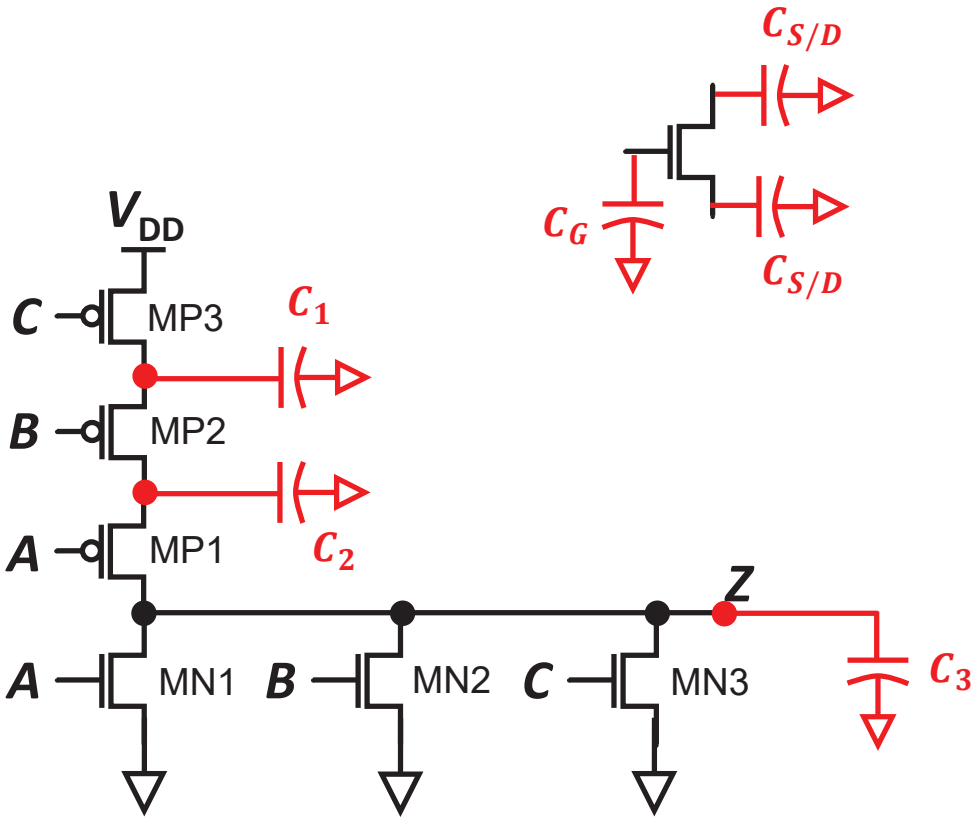
$$\tau_1 = 0.69R_1C_1 \qquad \tau_3 = 0.69(R_1 + R_2 + R_3)C_3$$

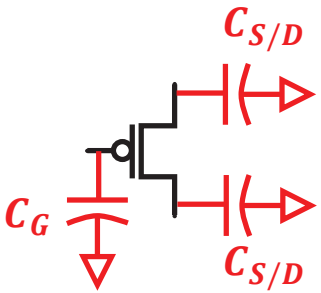
$$\tau_2 = 0.69(R_1 + R_2)C_2 \qquad \tau_4 = 0.69(R_1 + R_2 + R_3 + R_4)C_4$$

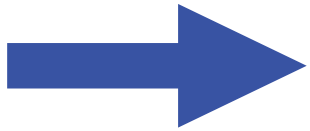
$$\tau_{4,actual} \approx \tau_1 + \tau_2 + \tau_3 + \tau_4$$

$$\tau_{M,actual} \approx 0.69 \sum_{i=1}^{M} R_i \left(\sum_{j=i}^{M} C_j \right)$$

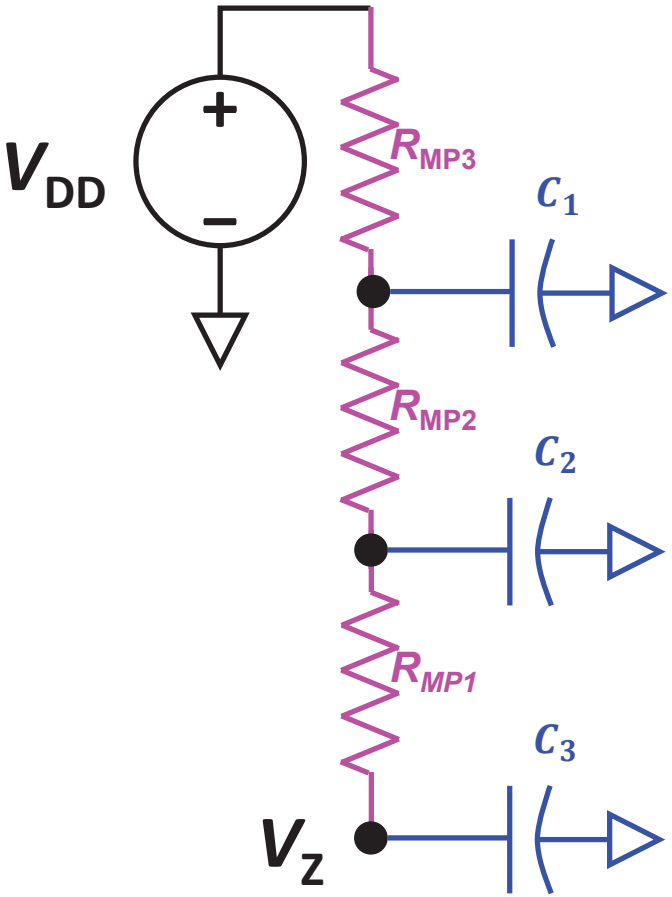
Propagation Delay with Intrinsic/Internal Capacitances





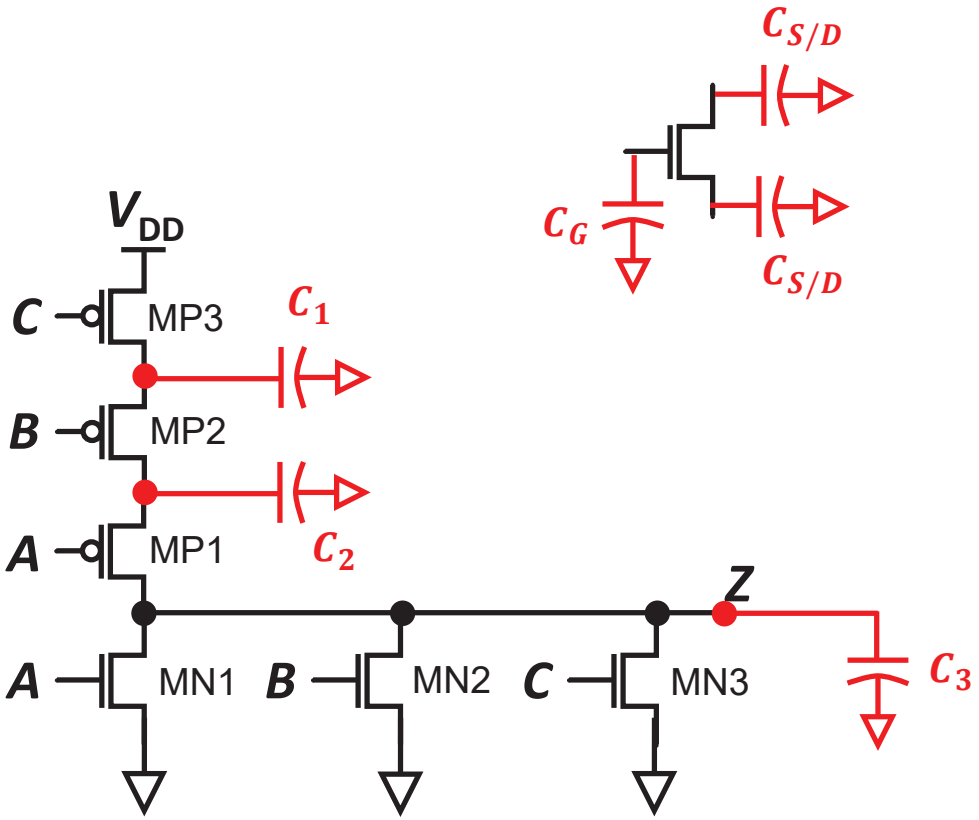


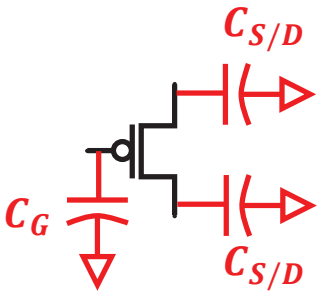
Model for calculating t_{pLH}

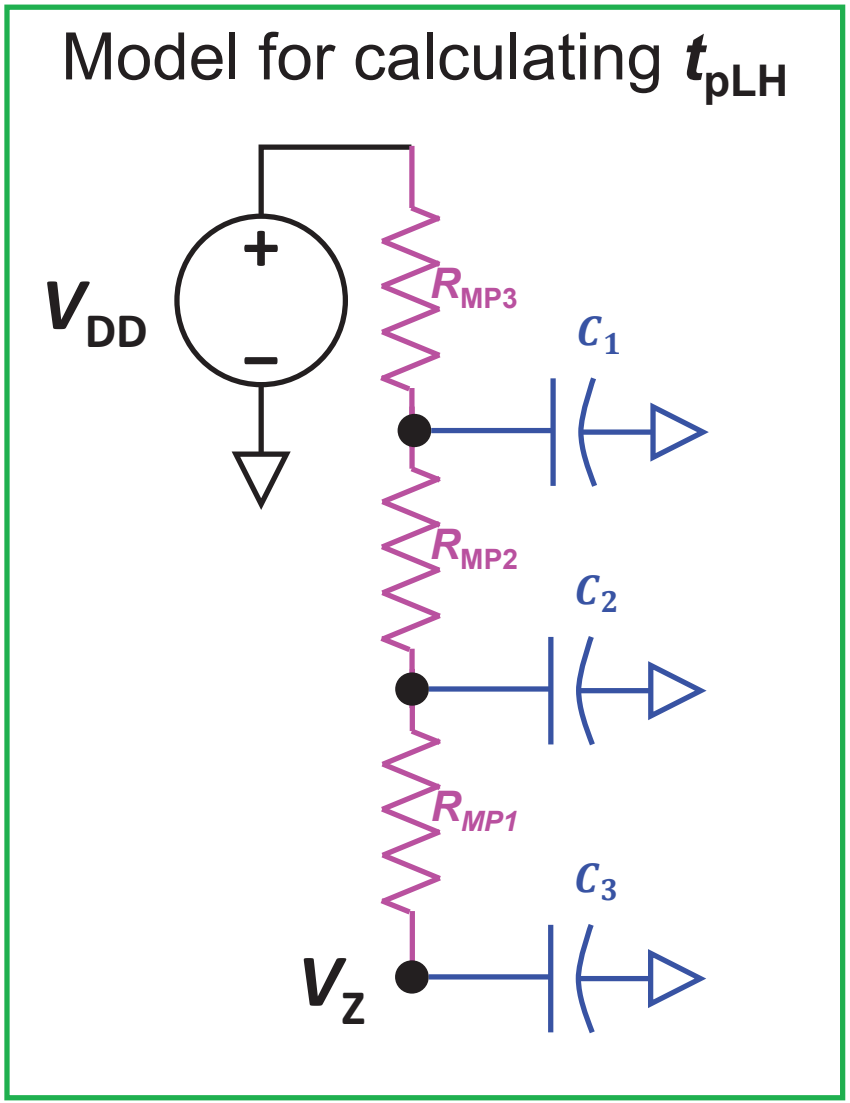


$$C_1 = C_{S/D,MP3} + C_{S/D,MP2}$$
$$C_2 = C_{S/D,MP2} + C_{S/D,MP1}$$
$$C_3 = C_{S/D,MP1} + C_{S/D,MN1} + C_{S/D,MN2} + C_{S/D,MN3} + C_L$$

Propagation Delay with Intrinsic/Internal Capacitances







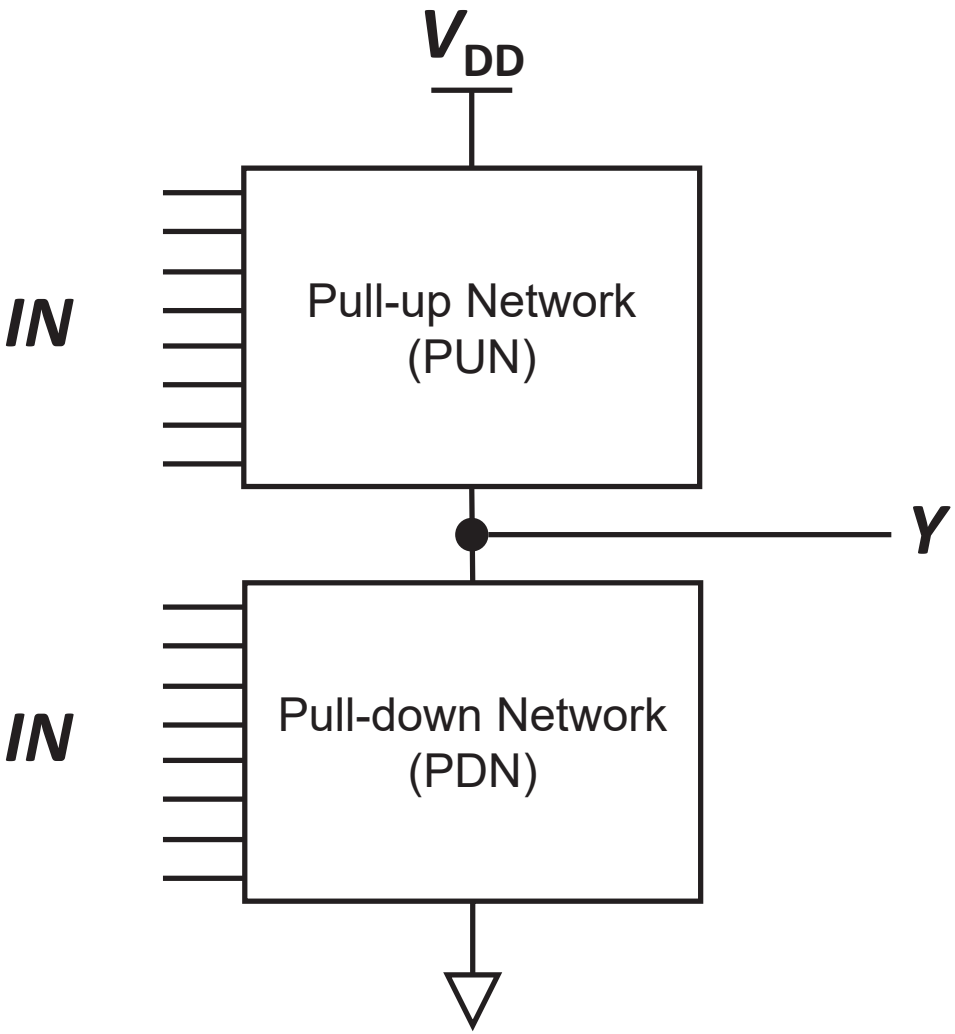
$$C_{3,i} = C_{S/D,MP1} + C_{S/D,MN1} + C_{S/D,MN2} + C_{S/D,MN3}$$

$$t_{pLH} = 0.69[R_{MP3}C_1 + (R_{MP3} + R_{MP2})C_2 + (R_{MP3} + R_{MP2} + R_{MP1})C_{3,1}] + 0.69(R_{MP1} + R_{MP2} + R_{MP3})C_L$$

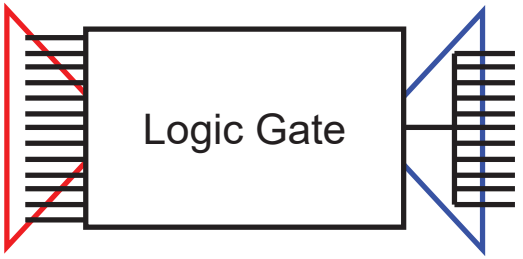
Week 3-7

Fan-in and Fan-out

Fan-in and Fan-out

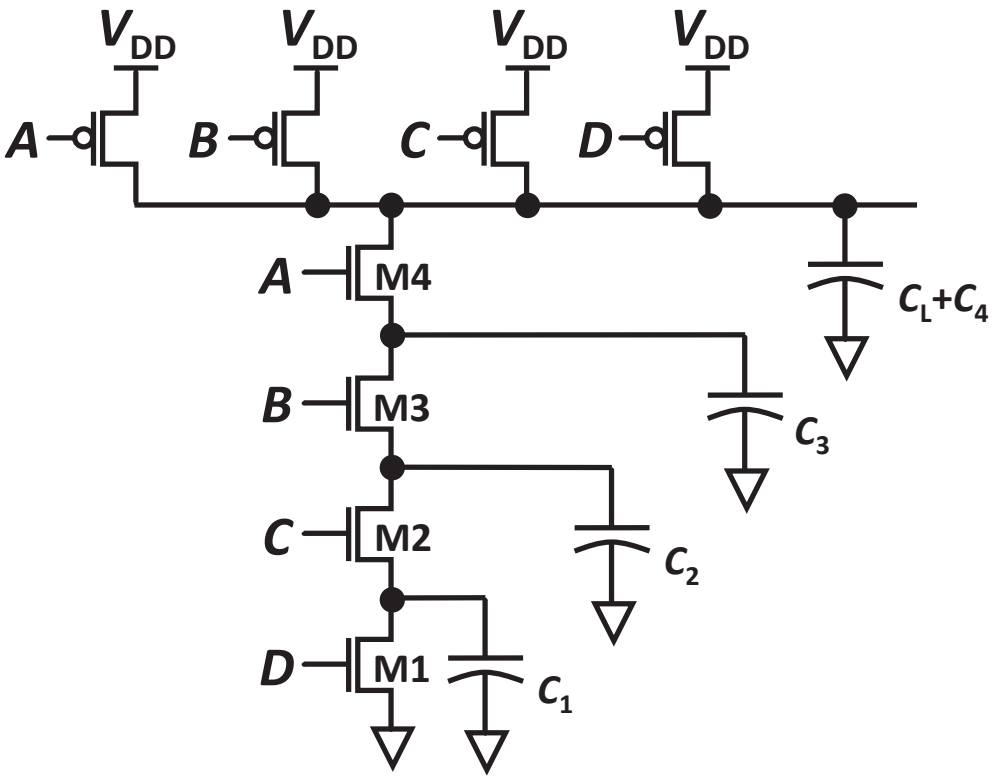


- Broadly speaking:
  - Fan-in: number of inputs to the logic gate
  - Fan-out: number of gates the output is connected to



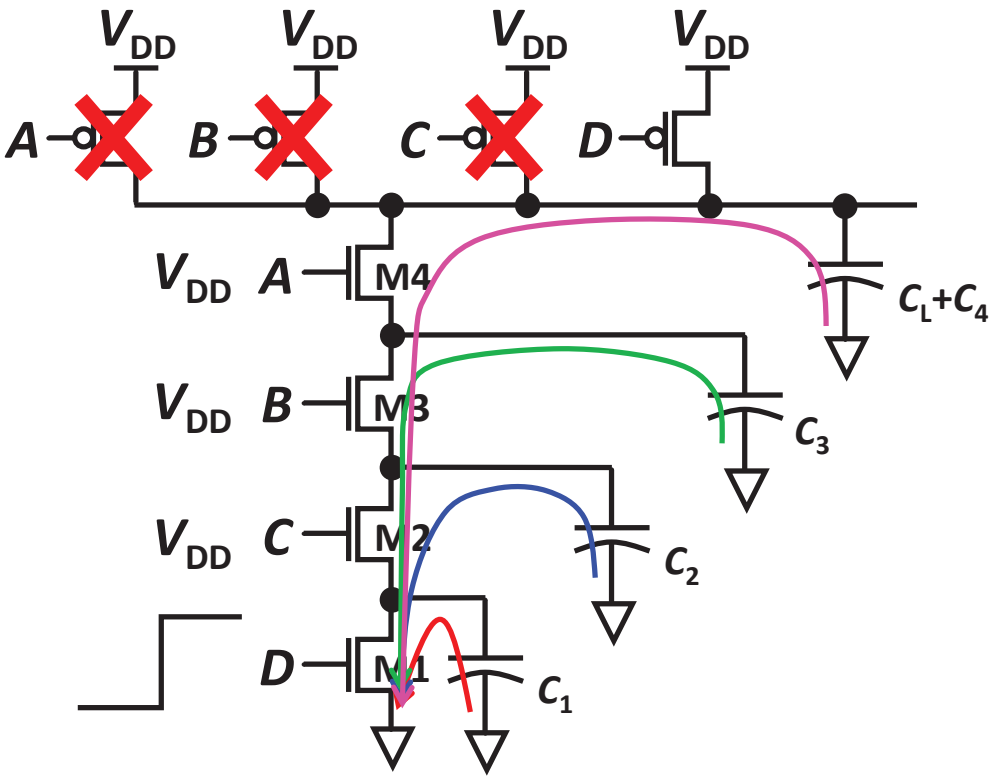
- How do fan-in and fan-out affect the key metrics of logic gates?
  - Fan-out: affects C_L and thus, affects delay and active power dissipation
  - Fan-in: affects number of transistors in the logic gate, which determines internal parasitic capacitances

Case Study: 4-input NAND Gate



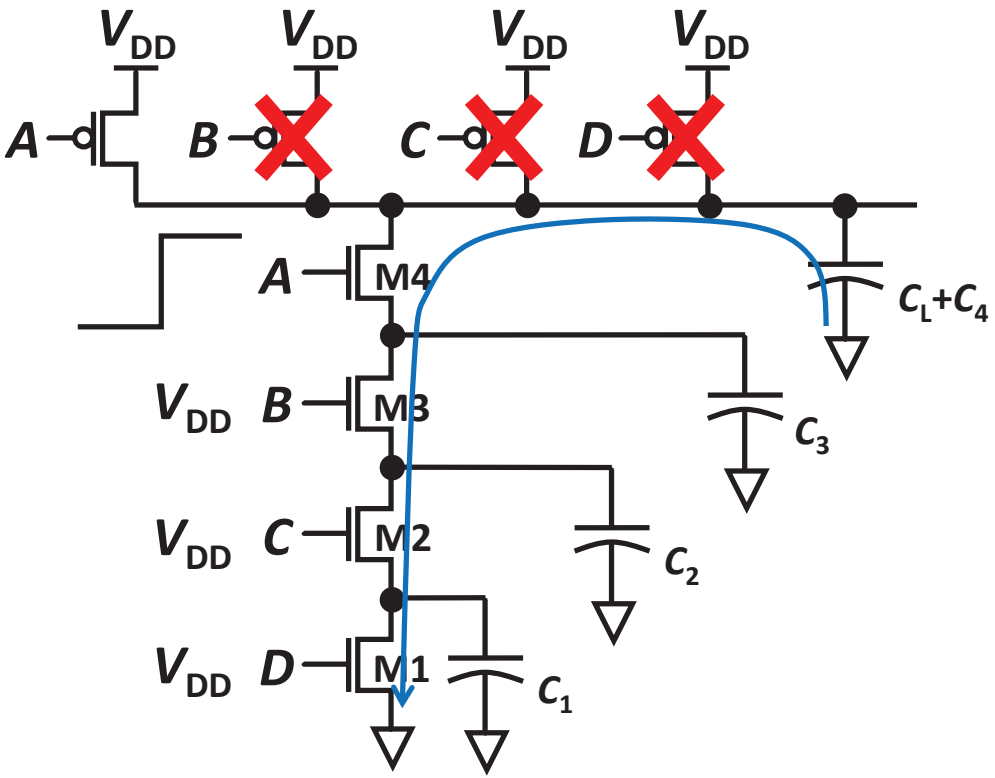
- Compare the delay for when *ABCD* changes to 1111 from:
  i. 1110
  ii. 0111
- Initial conditions:
  i. C_1, C_2, C_3, and $C_L + C_4$ charged to $\boldsymbol{V_{DD}}$
  ii. C_L charged to $\boldsymbol{V_{DD}}$ whereas C_1, C_2, C_3, charged to GND
- Final conditions:
  - C_1, C_2, C_3, and C_L all discharged to GND in both cases
- Signal delays based on Elmore delay model:

Case Study: 4-input NAND Gate



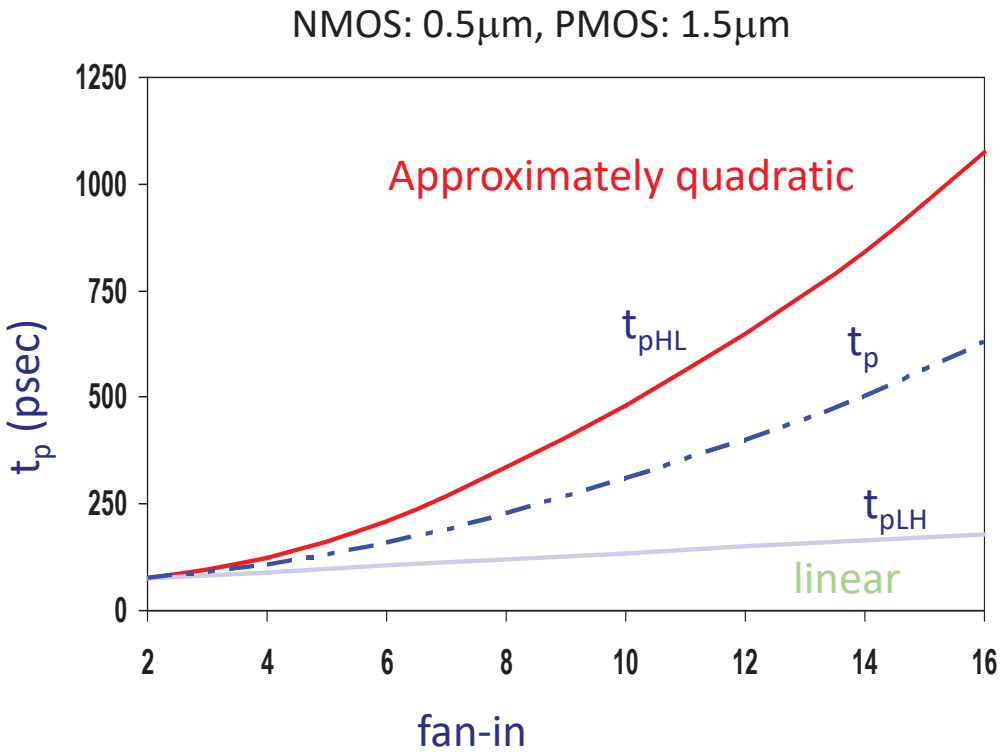
- Compare the delay for when *ABCD* changes to 1111 from:
  - i. 1110
  - ii. 0111
- Initial conditions:
  - *i.* C_1, C_2, C_3, and $C_L + C_4$ charged to $\boldsymbol{V_{DD}}$
  - *ii.* C_L charged to $\boldsymbol{V_{DD}}$ whereas C_1, C_2, C_3, charged to GND
- Final conditions:
  - C_1, C_2, C_3, and C_L all discharged to GND in both cases
- Signal delays based on Elmore delay model:
  - *i.* $R_{M1}C_1+C_2(R_{M1}+R_{M2})+C_3(R_{M1}+R_{M2}+R_{M3})$ $+(C_L+C_4)(R_{M1}+R_{M2}+R_{M3}+R_{M4})$

Case Study: 4-input NAND Gate



- Compare the delay for when *ABCD* changes to 1111 from:
  - i. 1110
  - ii. 0111
- Initial conditions:
  - *i.* C_1, C_2, C_3, and $C_L + C_4$ charged to $\boldsymbol{V_{DD}}$
  - *ii.* C_L charged to $\boldsymbol{V_{DD}}$ whereas C_1, C_2, C_3, charged to GND
- Final conditions:
  - C_1, C_2, C_3, and C_L all discharged to GND in both cases
- Signal delays based on Elmore delay model:
  - *i.* $R_{M1}C_1+C_2(R_{M1}+R_{M2})+C_3(R_{M1}+R_{M2}+R_{M3})+(C_L+C_4)(R_{M1}+R_{M2}+R_{M3}+R_{M4})$
  - ii. $(C_L+C_4)(R_{M1}+R_{M2}+R_{M3}+R_{M4})$

Generalization: *N*-input NAND Gate



NAND gate:

- Intrinsic capacitance increase linearly with the fan-in
- Series connection of the transistor causes slow-down (t_{pHL})
- Gates with a fan-in greater than 4 should be avoided.

If R_{Mj} are the same and all C_j are the same,

$$\tau_{d,worst} = 0.69\left[NRC_L + RC\sum_{i=1}^{N} i\right] = 0.69R\left[C\left(\frac{N(N+1)}{2}\right) + NC_L\right]$$

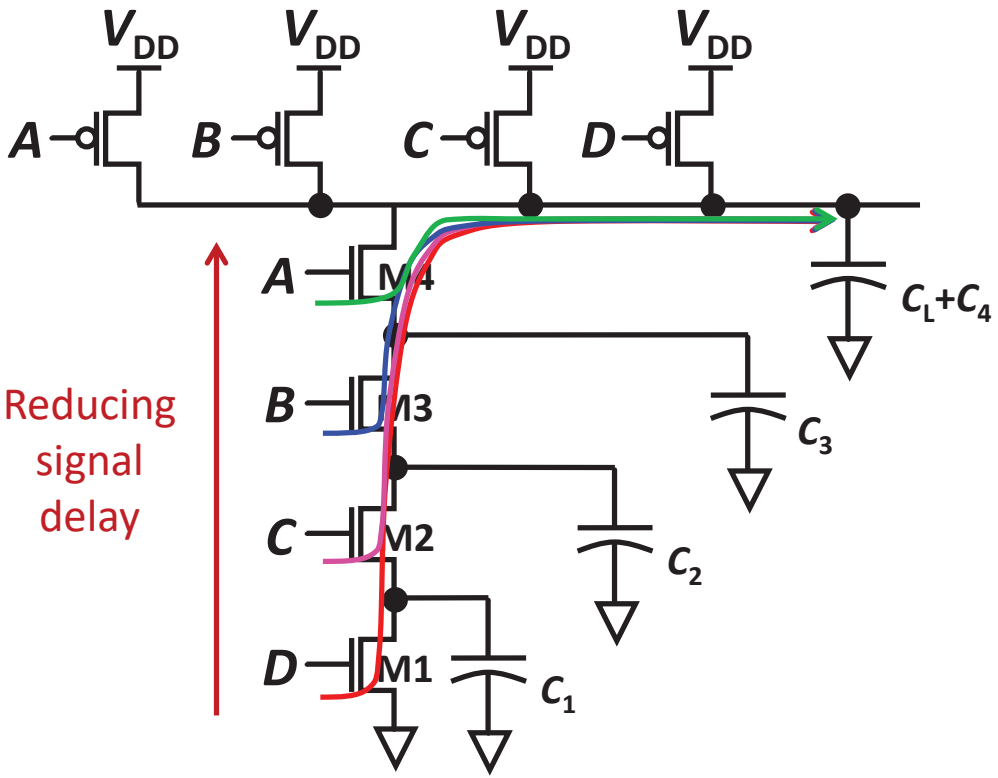
Design Techniques for Large Fan-in

- Transistor sizing
  - Only effective when fan-out dominates the delay

$$\tau_{d,worst} = 0.69R\left[C\left(\frac{N(N+1)}{2}\right) + NC_L\right] \approx 0.69NRC_L$$

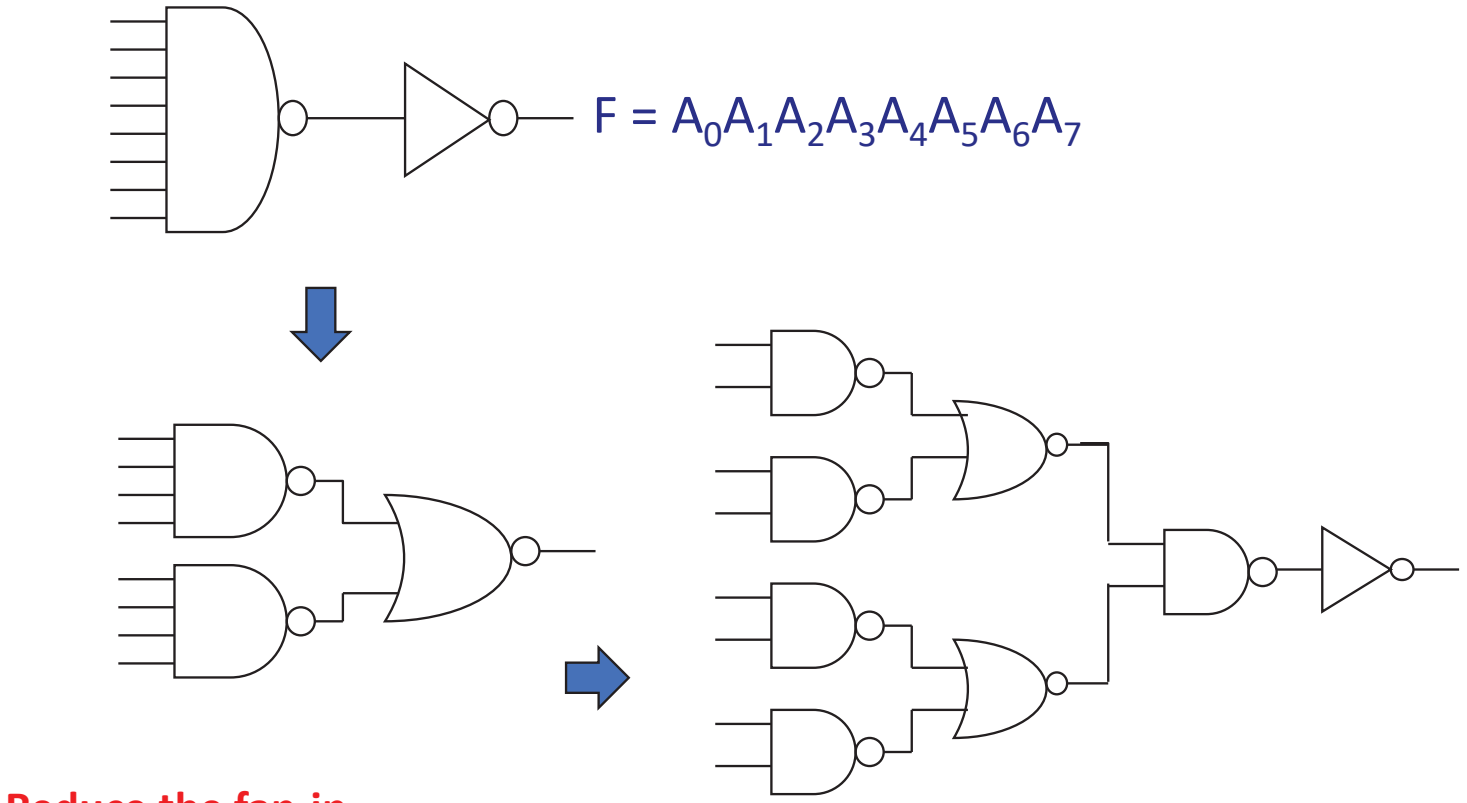
    Sizing to reduce *R* and delay
- Progressive transistor sizing
  - M1 > M2 > M3 > … > MN (distributed delay line)
  - Easy to do in schematic but may be difficult in layout due to design rules
- Input reordering
- Logic restructuring

Input Reordering



- Compare the delay for when *ABCD* changes to 1111 from:
  - i. 1110
  - ii. 0111
- Initial conditions:
  - *i.* C_1, C_2, C_3, and C_L all charged to $\boldsymbol{V_{DD}}$
  - *ii.* C_L all charged to $\boldsymbol{V_{DD}}$ whereas C_1, C_2, C_3, charged to GND
- Final conditions:
  - C_1, C_2, C_3, and C_L all discharged to GND in both cases
- Signal delays based on Elmore delay model:
  - *i.* $R_{M1}C_1+C_2(R_{M1}+R_{M2})+C_3(R_{M1}+R_{M2}+R_{M3})+(C_L+C_4)(R_{M1}+R_{M2}+R_{M3}+R_{M4})$
  - ii. $(C_L+C_4)(R_{M1}+R_{M2}+R_{M3}+R_{M4})$

Logic Restructuring



Reduce the fan-in

Similar to buffer insertion

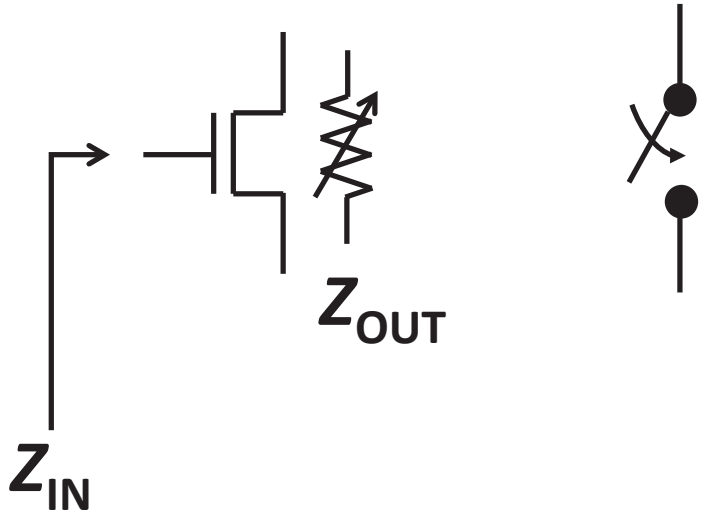
Week 3-8

Pass Transistor (Pass Gate) and Transmission Gate Logic

NMOS Transistor as an Electronic (Digital) Switch

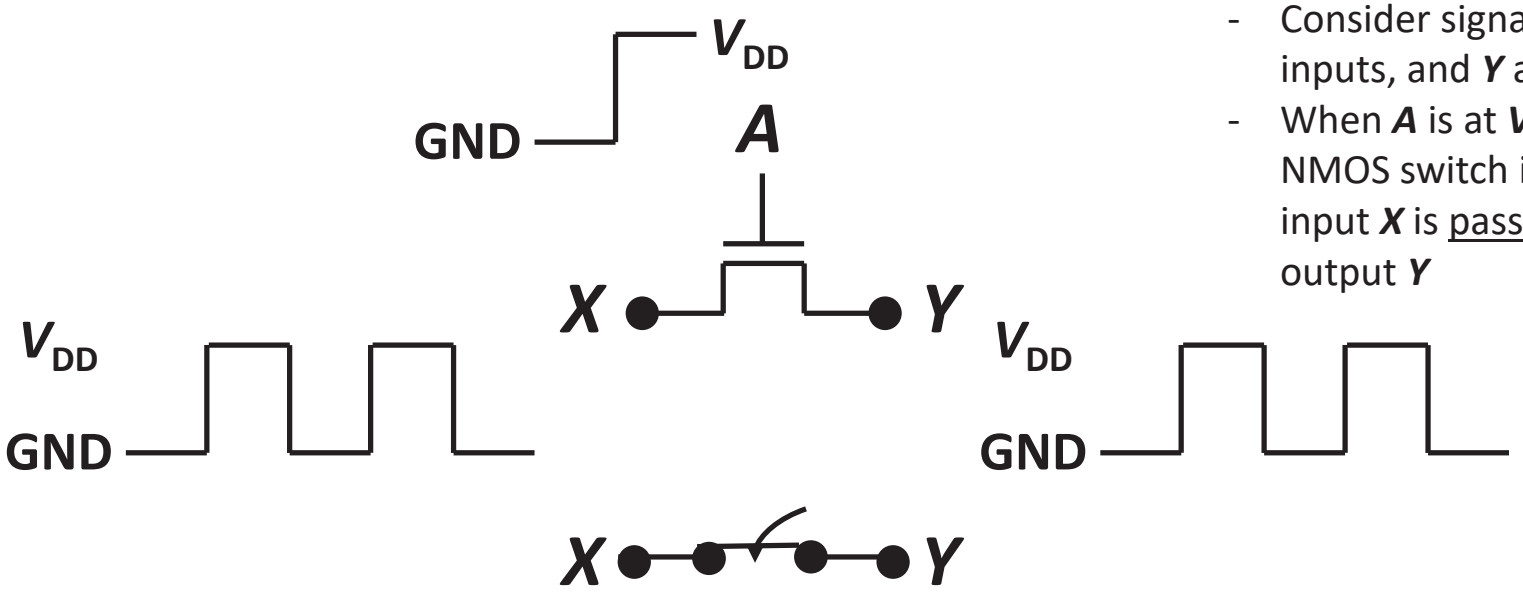
What about cascading logic gates? (Output of a logic gate is connected to input of another logic gate)

- Model for input impedance (Z_{IN})
- Model for output impedance (Z_{OUT})



- Voltage on gate terminal of transistor controls resistance between its source and drain terminals
  - Low resistance = switch is ON or CLOSED
  - High resistance = switch is OFF or OPEN

NMOS Transistor as an Electronic (Digital) Switch



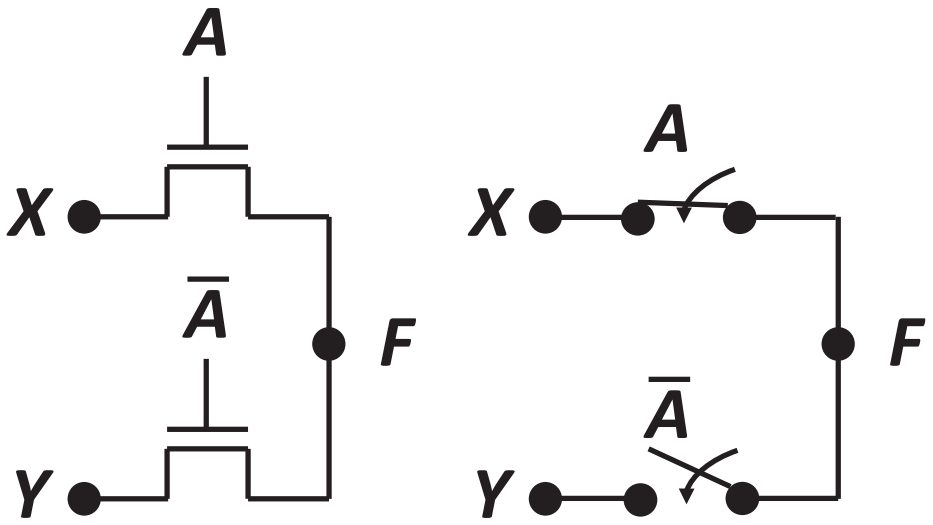
- Consider signals ***A*** and ***X*** as inputs, and ***Y*** as an output
- When ***A*** is at V_{DD} ('1'), the NMOS switch is closed, and the input ***X*** is passed to the output ***Y***

What if ***A*** is at **GND** ('0')?

The NMOS switch is open and ***Y*** is in the high-***Z*** state

- When used in this manner, the NMOS transistor is called a *pass transistor* or *pass gate*

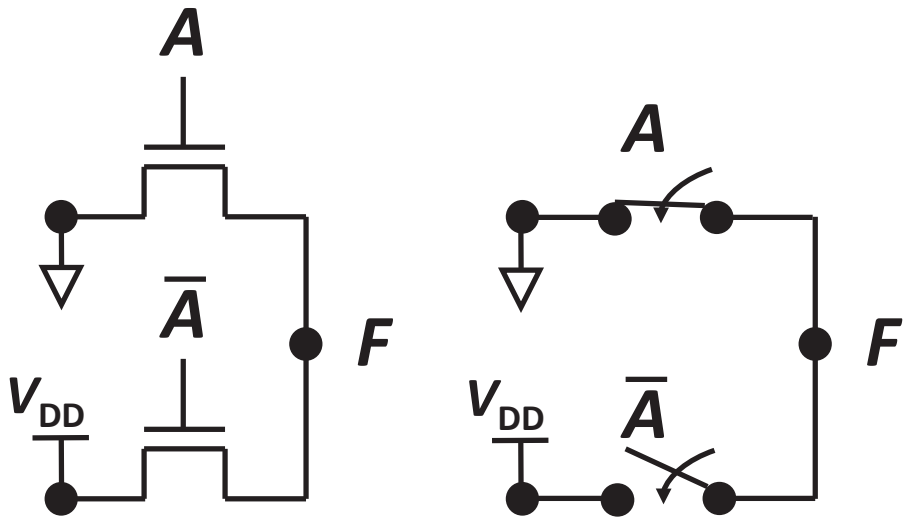
The Pass Transistor Multiplexers



- What is the logic function of ***F***?
- ***A*** = 0: ***F*** = ***Y***
- ***A*** = 1: ***F*** = ***X***

$$F = AX + \bar{A}Y$$ **(Multiplexer)**

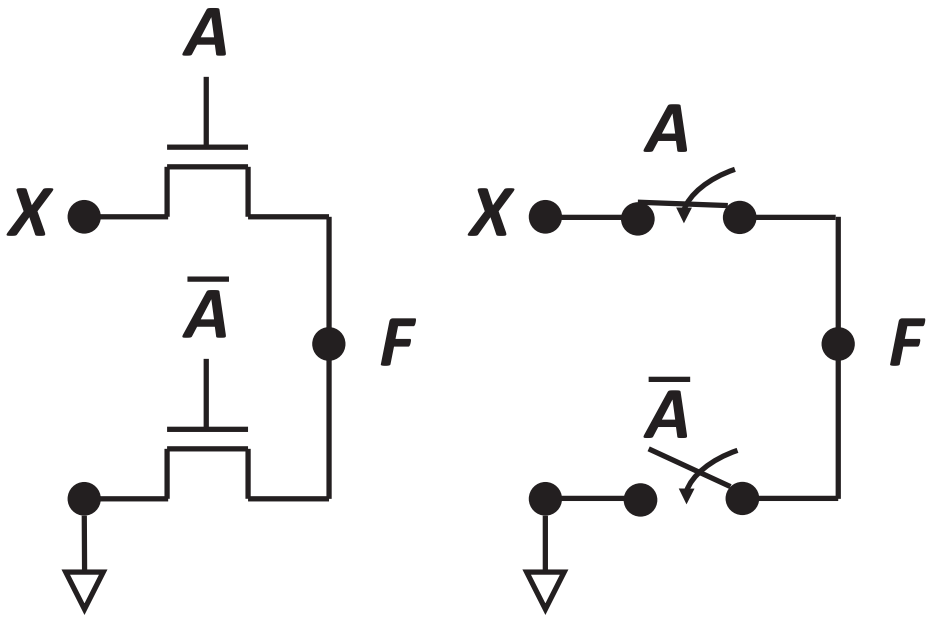
Example 1: The Pass Transistor Inverter Gate



- What is the logic function of ***F*** if we fix ***Y*** = 1 and ***X*** = 0?

$$F = AX + \overline{A}Y$$
$$= \overline{A}$$

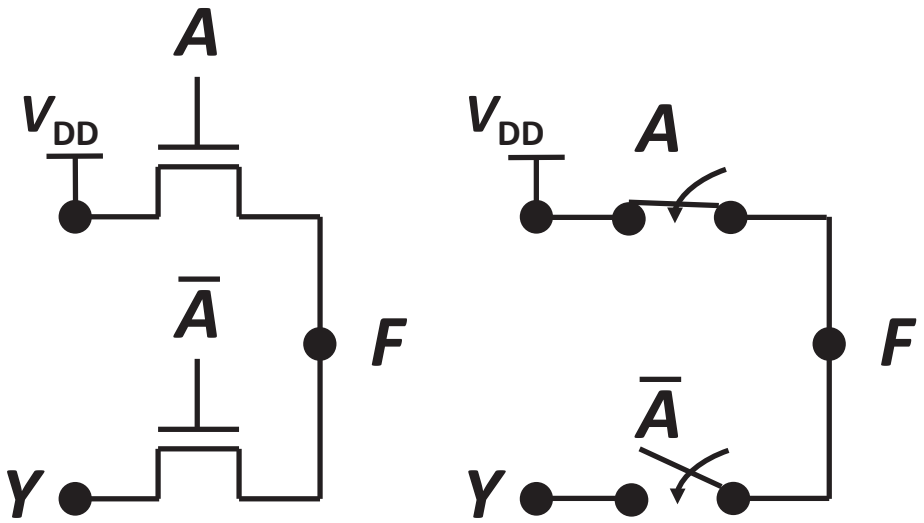
Example 2: The Pass Transistor AND Logic Gate



- What is the logic function of ***F*** if we fix ***Y*** = 0?

$$F = AX + \overline{A}Y$$
$$= AX$$

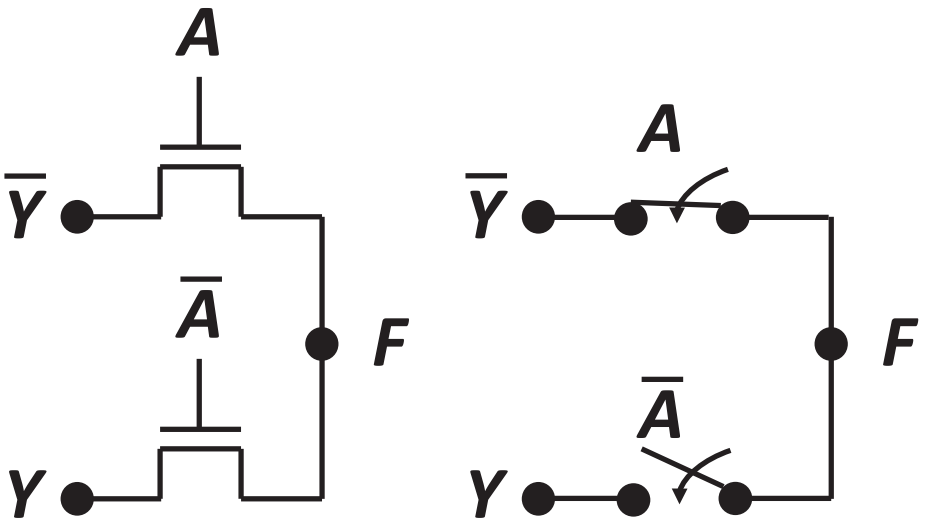
Example 3: The Pass Transistor OR Logic Gate



- What is the logic function of ***F*** if we fix ***X*** = 1?

$$F = A + \overline{A}Y$$
$$= A + Y$$

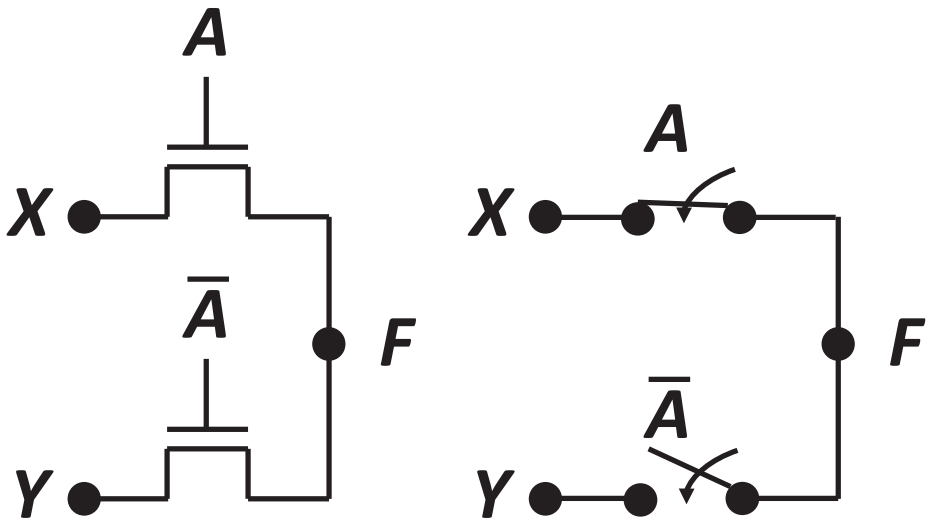
Example 4: The Pass Transistor XOR Logic Gate



- What is the logic function of ***F*** if we fix ***X*** = $\overline{\boldsymbol{Y}}$?

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{AX} + \overline{\boldsymbol{A}}\boldsymbol{Y} \\ &= \boldsymbol{A}\overline{\boldsymbol{Y}} + \overline{\boldsymbol{A}}\boldsymbol{Y} \\ &= \boldsymbol{A} \oplus \boldsymbol{Y} \end{aligned}$$

Pass Gate Logic using Multiplexers



$F = AX + \bar{A}Y$ **(Multiplexer)**

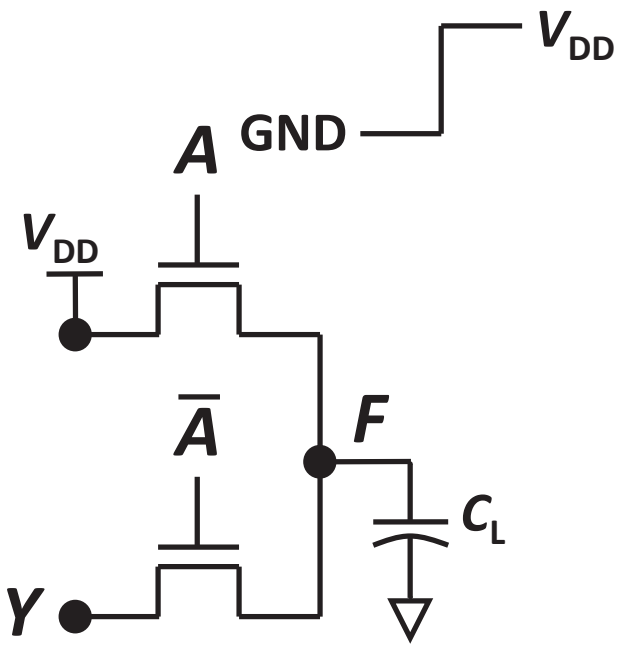
Advantages of pass gate logic:

- Able to implement non-inverting logic
  - AND/OR vs. NAND/NOR
- Possible to implement logic using few transistors (save area)

Pass gate logic sounds great! Are there any problems?

May need to generate complementary signals

Design Issues in Pass Gate Logic



Consider that initially, $A = Y = F = $ '0'

Switch signal A to '1'

Current flows from V_{DD} to charge V_F to V_{DD}

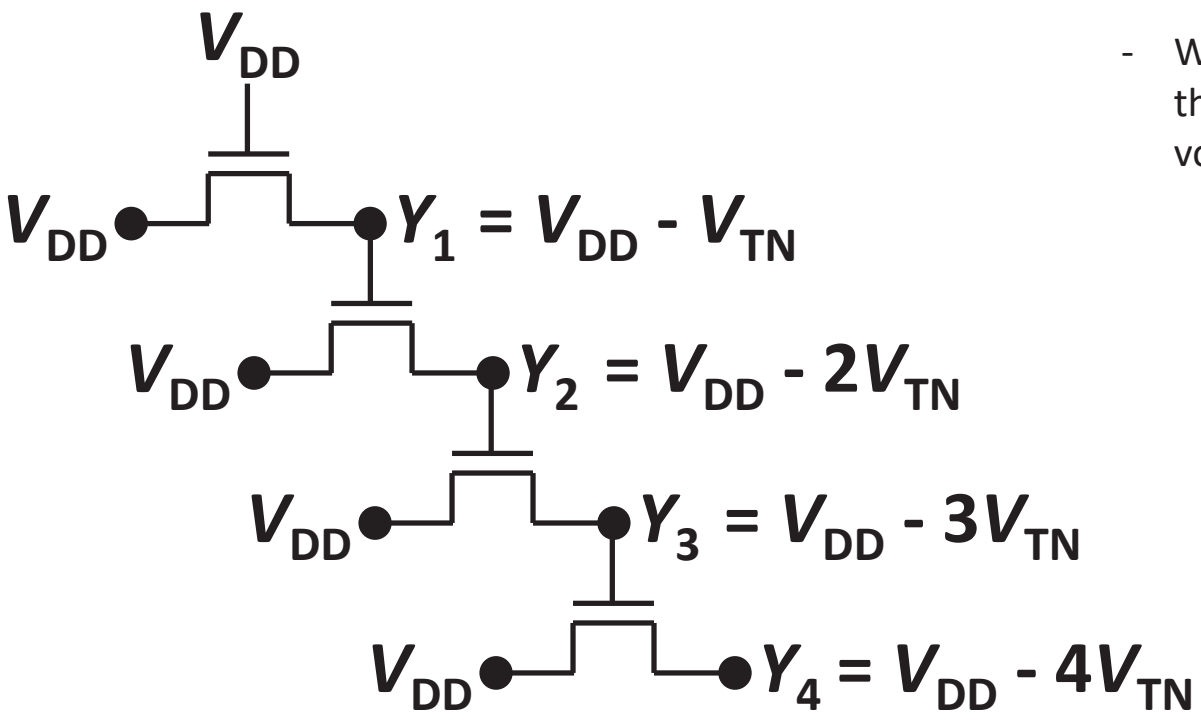
The source of the NMOS pass gate is at F (drain is at V_{DD}) so $V_{GS} = V_{DS} = V_{DD}$ immediately after A become '1'

$V_F = V_S$ keeps increasing until $I_D = 0$:

Two possible conditions for $I_D = 0$: $V_{GS} = V_{TN}$ or $V_{DS} = 0$ V

There is a V_{TN} drop when trying to pass '1' (weak '1')
No problem with passing '0' (strong '0')

Cascading Pass Transistor Logic Gates - I

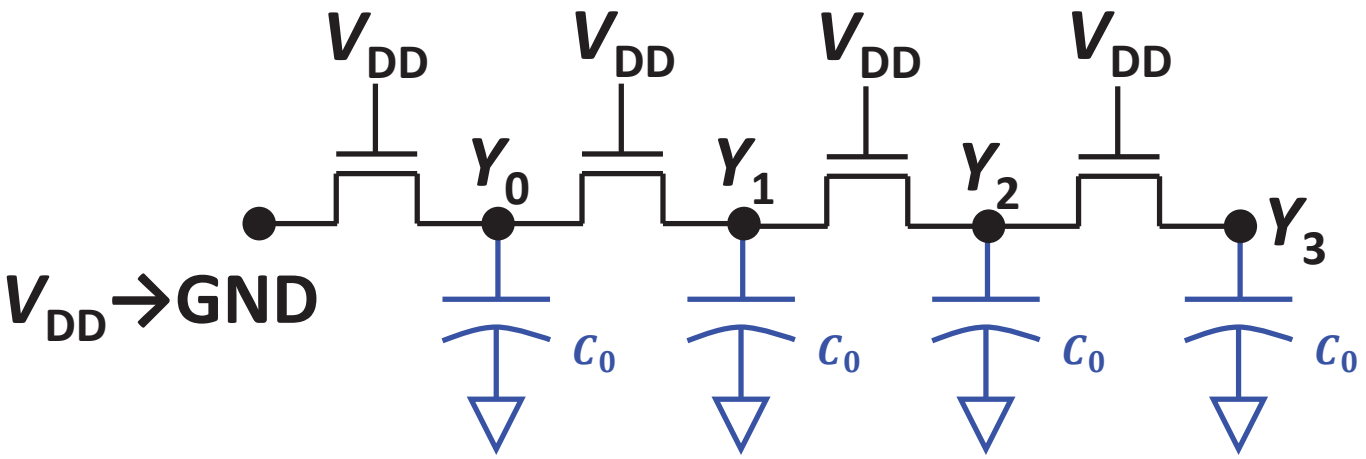


- What are the voltages on the gates of the pass transistors? (Assume all node voltages were initially at **GND**)

Highest voltage passed is voltage on gate minus V_{TN}.

Too much voltage loss may lead to loss of functionality!

Cascading Pass Transistor Logic Gates - II



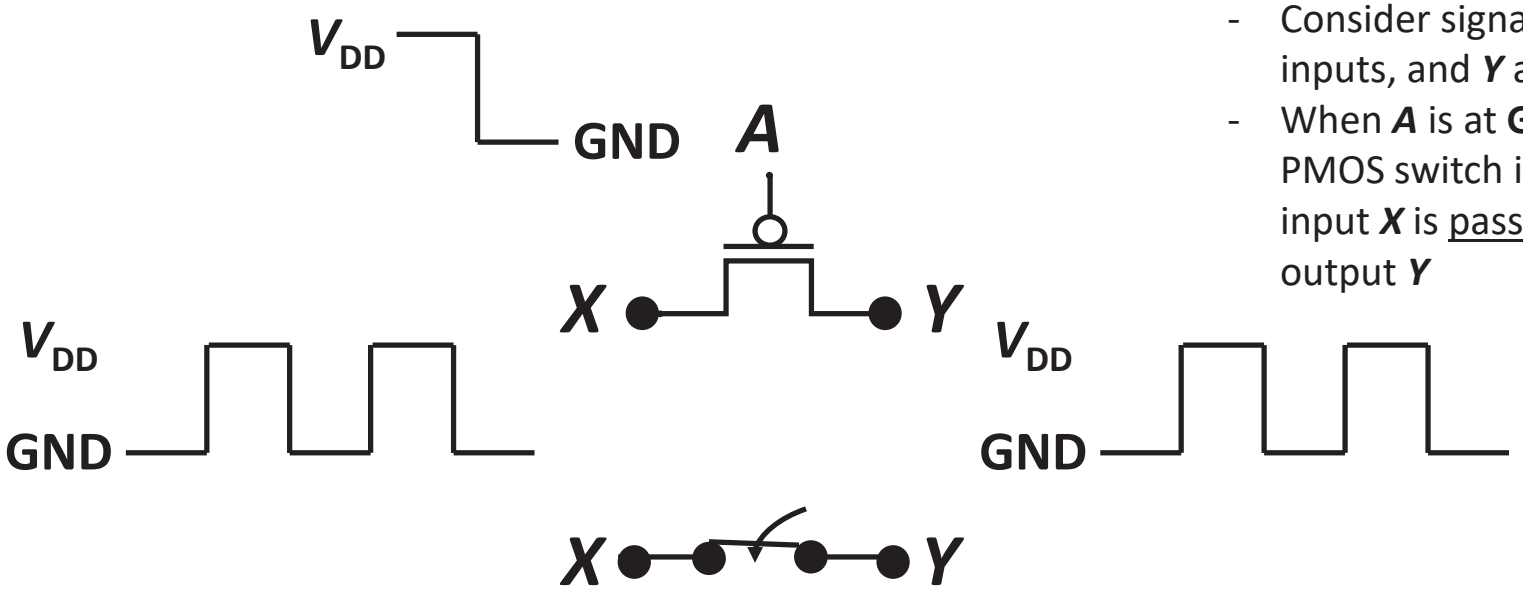
- What is the signal delay for Y_3? (Assume all intermediate node voltages were initially at **VDD** and all pass transistors have same R_{ON})

$$\tau = 0.69 R_{ON} \left(\frac{N(N-1)}{2} \right) C_0 \propto N^2$$

N: number of series connected pass transistors

$$R_{ON} \propto 1/I_D$$
$$I_D \propto W/L$$
$$R_{ON} \propto L/W$$

PMOS Transistor as an Electronic (Digital) Switch

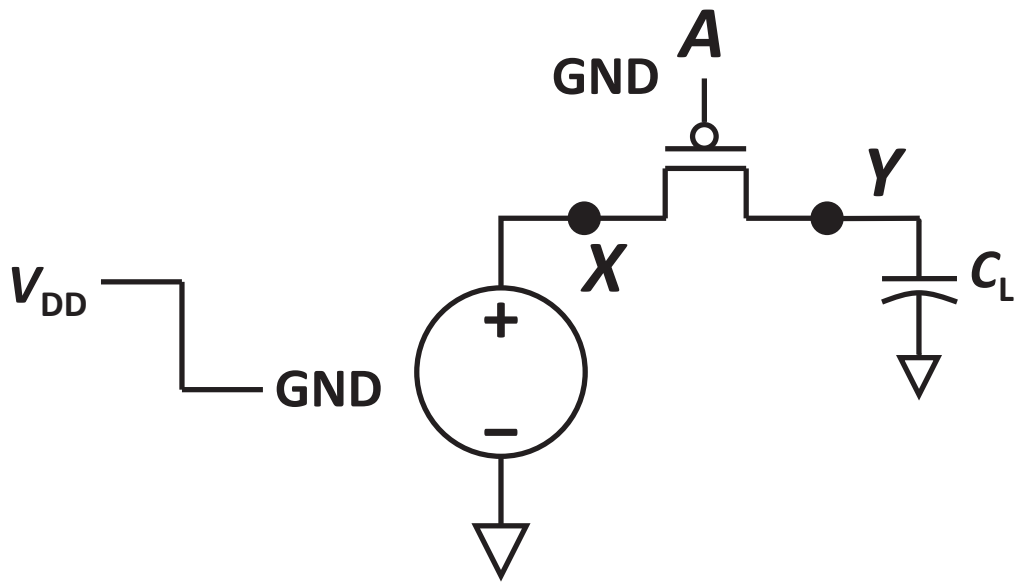


- Consider signals ***A*** and ***X*** as inputs, and ***Y*** as an output
- When ***A*** is at **GND** (‘0’), the PMOS switch is closed and the input ***X*** is passed to the output ***Y***

Since the PMOS transistor can also behave as a digital switch like the NMOS transistor, we can also use it as a pass gate

Not often done because of size and the problem we will discuss next

PMOS Transistor as an Electronic (Analog) Switch - II



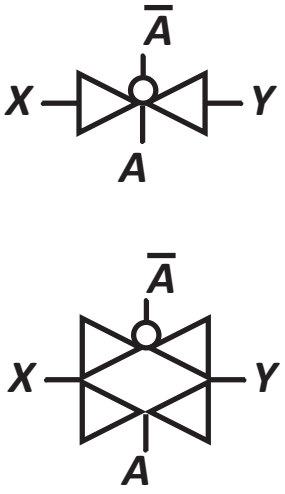
- Assume initially $V_Y = V_{DD}$
- If ***X*** is switched to **GND**, current flows from ***Y*** to ***X*** to discharge V_Y to **GND**
- Current flows from source to drain in PMOS transistor
  - ***Y*** is source of p-MOSFET
- I_D will flow until $V_{GS} = V_{TP}$ and V_Y can only discharge to $-V_{TP}$

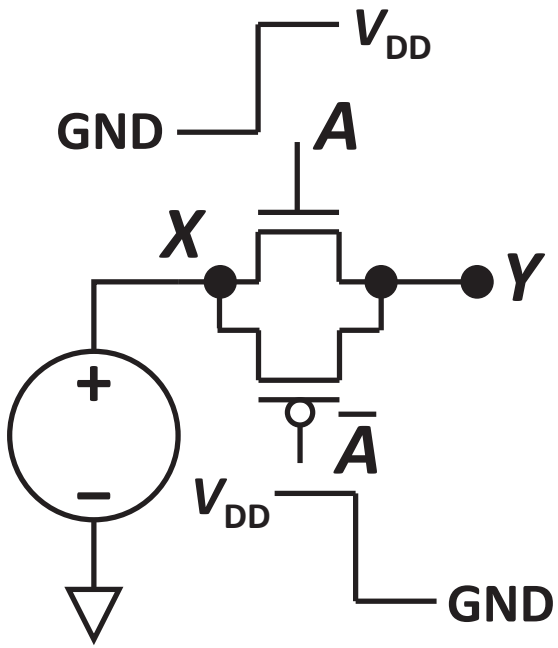
There is a V_{TP} drop when trying to pass '0' (weak '0')
No problem with passing '1' (strong '1')

NMOS: Strong '0' but weak '1'
PMOS: Strong '1' but weak '0'
} Connect both in parallel to form a switch

The Transmission Gate

Circuit Symbols

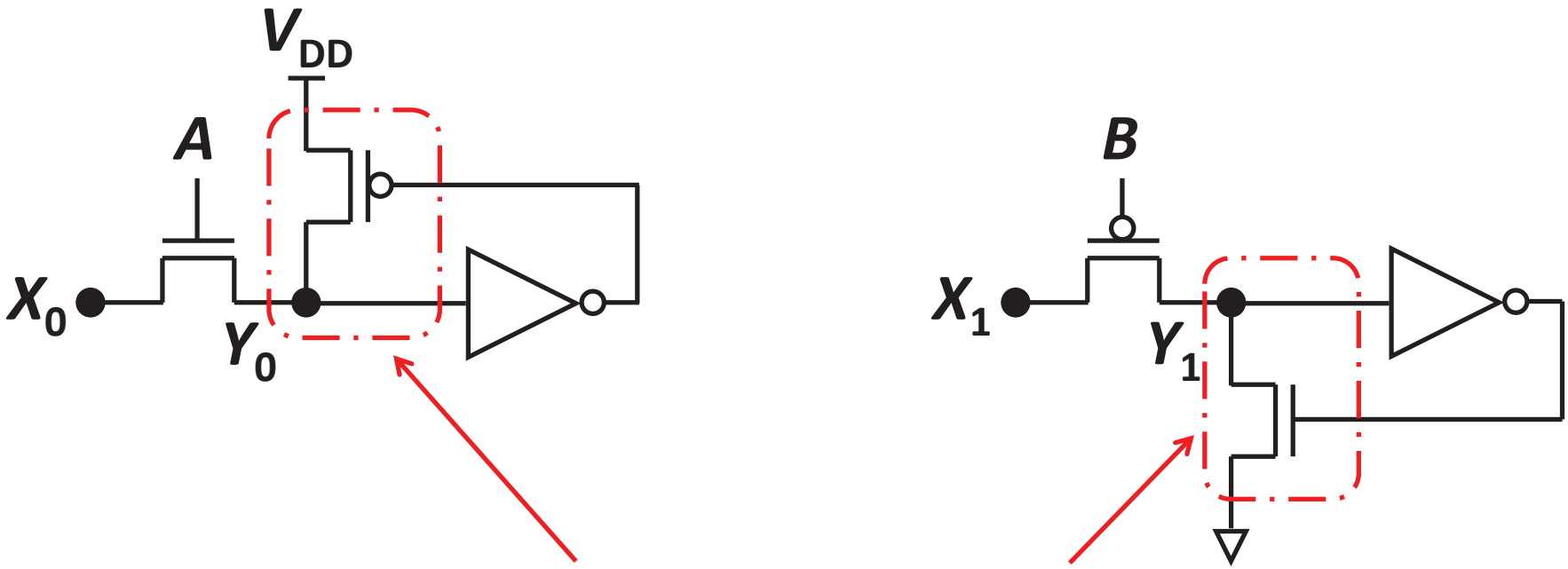




- A transmission gate is formed using a p-MOSFET and an n-MOSFET connected as shown
  - The gates are connected to complementary signals so that both of them are either ON or OFF at the same time
  - The source terminal of the n-MOSFET is connected to the drain terminal of the p-MOSFET
  - The drain terminal of the n-MOSFET is connected to the source terminal of the p-MOSFET
- The n-MOSFET helps to pass strong ‘0’
- The p-MOSFET helps to pass strong ‘1’

Need complementary signals AND twice the number of MOSFETs to pass strong ‘0’ and strong ‘1’!
Does not solve problem of delays in long chains of transmission gates!

Level Restoring Transistor



Level Restoring Transistors

1) Need extra inverter
2) Need to design W/L of pass transistor and level restoring transistor carefully
   - Voltage input to inverter depends on relative values of $R_{ON,N}$ and $R_{ON,P}$ (designed so that inverter will turn keeper OFF)