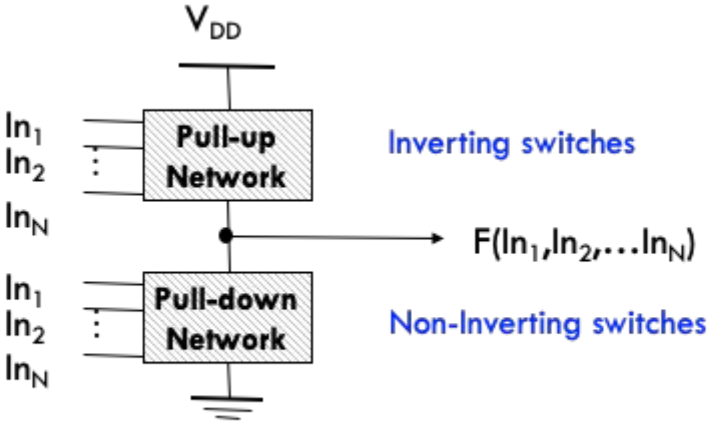
Discussion 10

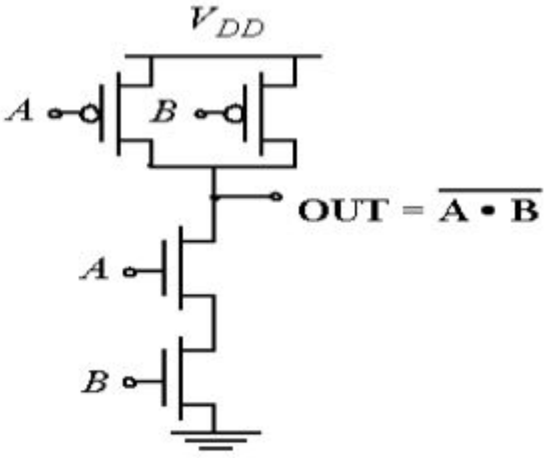
Gate Sizing, RC Delay, Logical Effort, Power/Energy, Adders

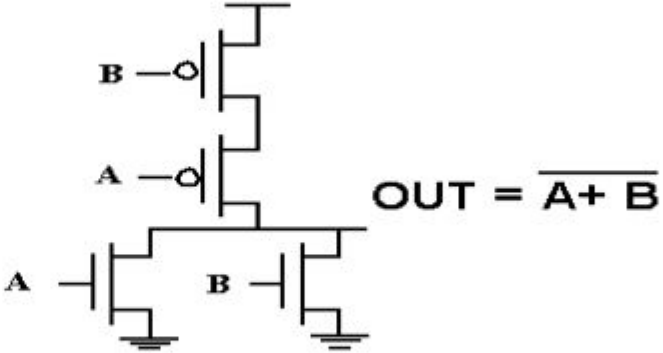
CMOS Logic



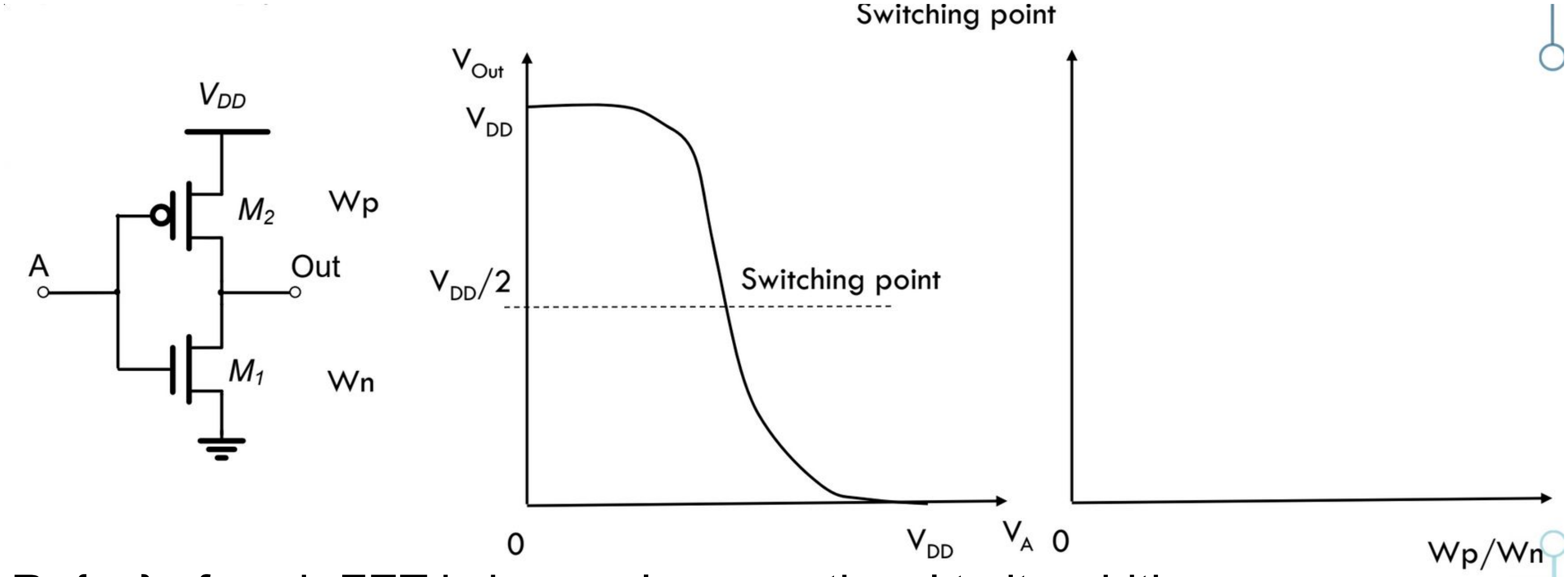
PUN and PDN are dual logic networks

PUN and PDN functions are complementary



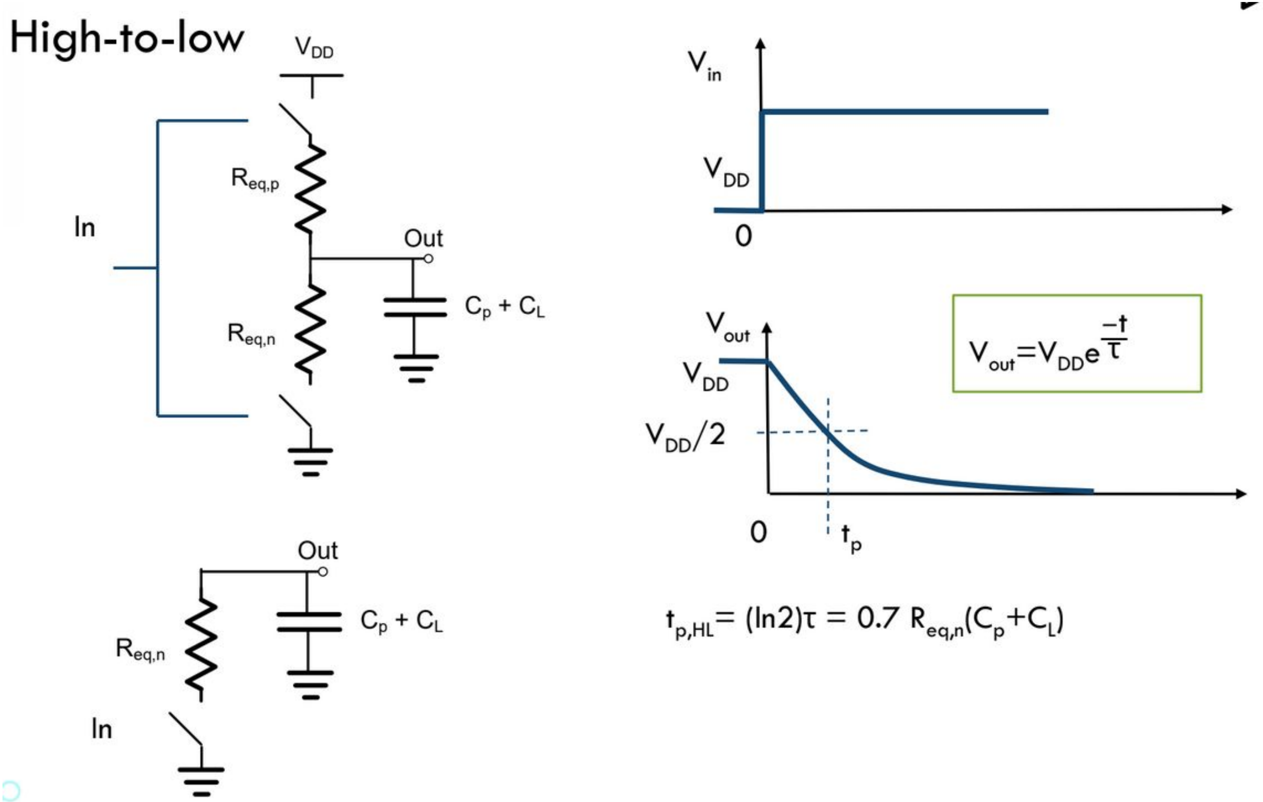


Inverter Sizing



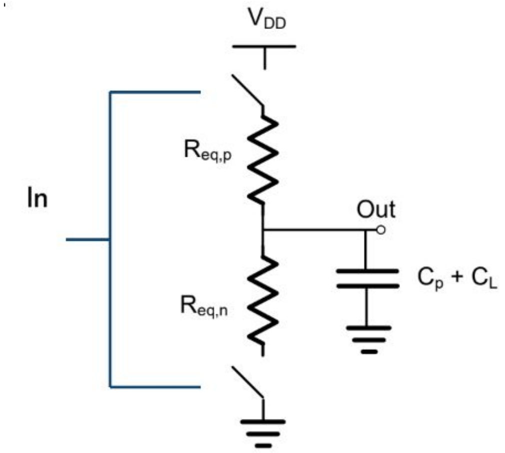
- R_{on} of each FET is inversely proportional to its width
- Assume V_{th,n} = V_{th,p}
- Assume R_{on,n} = R_{on,p} for the same width unless otherwise specified
- Wp = Wn = switching threshold of Vdd/2
- If Wp >> Wn, Vm approaches VDD | If Wn >> Wp, Vm approaches 0 (exponentially)

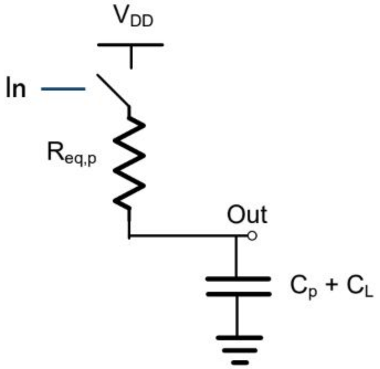
Inverter RC Delay (High -> Low)

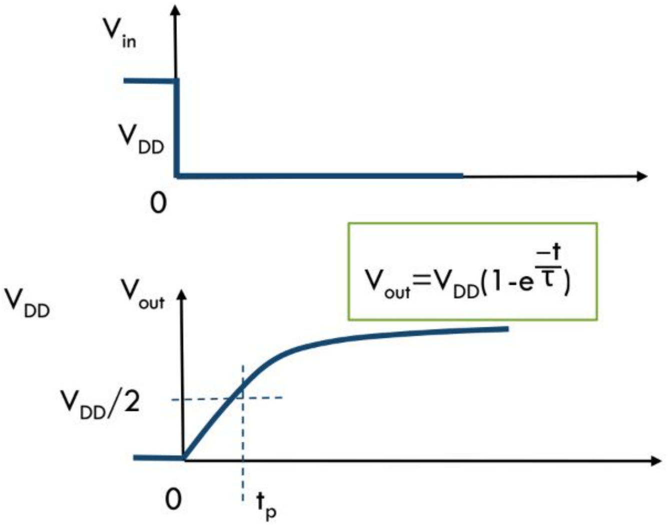


- High to low output transition time governed by strength of NMOS pulling low
- Can lower the time constant by increasing the NMOS width

Inverter RC Delay (Low -> High)



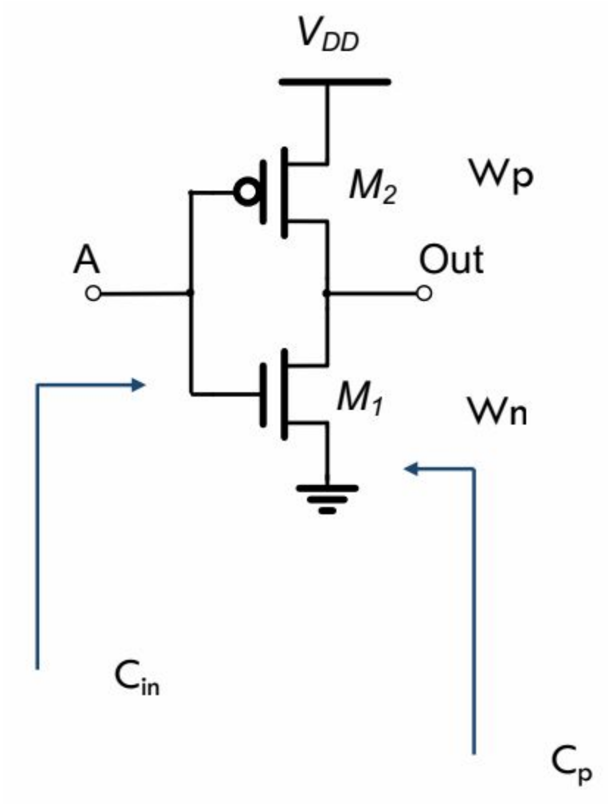




$$t_{p,LH} = (\ln 2)\tau = 0.7\ R_{eq,p}(C_p+C_L)$$

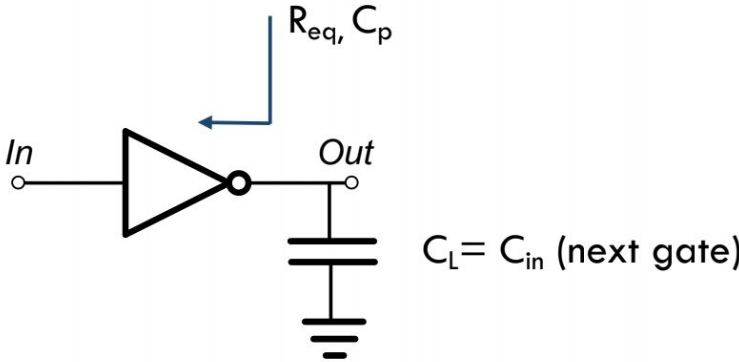
- Low to high output transition time governed by strength of PMOS pulling high
- Can lower the time constant by increasing the PMOS width

Inverter Sizing



- Cin (gate input capacitance) and Cp (intrinsic drain capacitance) are proportional to W
  - You may see Cp referred to Cd in other sources
  - Cp = γ Cin
- How does the inverter delay change if either PMOS/NMOS width is doubled?
- If both widths are doubled, does the intrinsic (unloaded) delay improve?
- Inverters are usually sized to equalize high->low and low->high delays

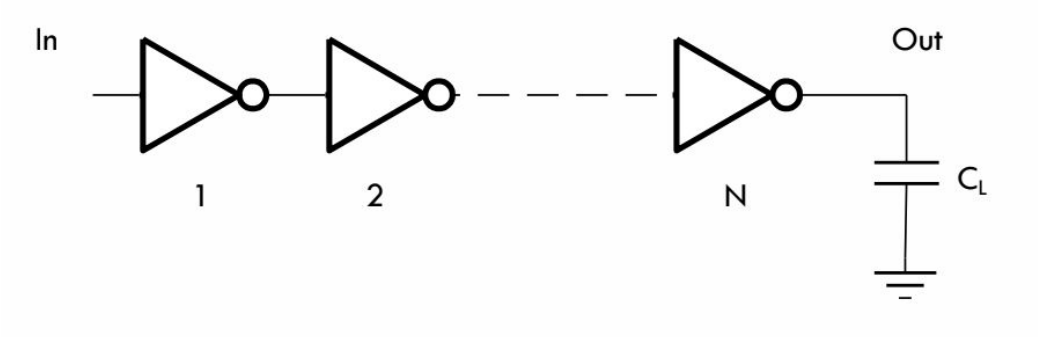
Inverter Delay



t_{p,inv} = R_{eq} (C_{p,tot} + C_L)
= R_{eq} C_{p,tot} (1 + (C_L / C_{p,tot}))
= R_{eq} C_{p,tot} (1 + (C_L / C_{in} γ))
= R_{eq} C_{p,tot} (1 + (f / γ))
= tau_{inv} (1 + (f / γ))

- Delay can be split into 2 parts, intrinsic and extrinsic
- Fanout f = ratio between output and input cap (CL / Cin)
- This delay formula can be generalized for any CMOS gate
    - t_{p,gate} = tau_{inv} (p + gf/γ)
    - Often assume γ = 1

Sizing an Inverter Chain



- We want to minimize the delay of this inverter chain
- Assume the 1st inverter has a size of 1

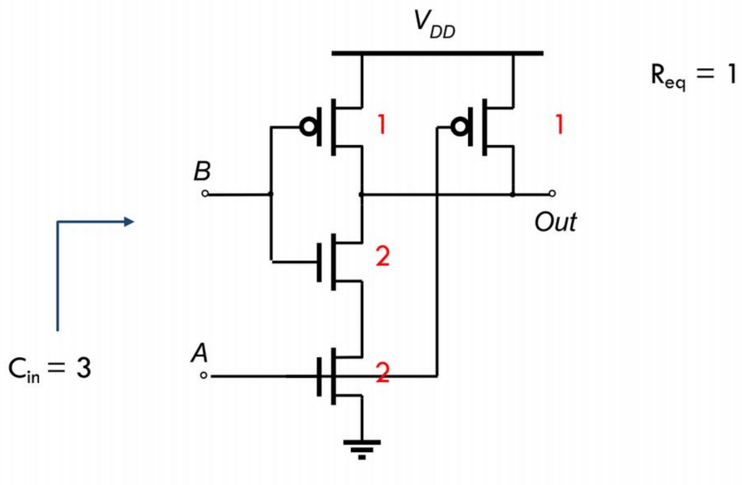
Delay = tp1 + … + tpn = (1+f1) + (1+f2) + … + (1+fn)

Take partial derivatives of this expression wrt C2, … Cn

Solution is f1 = f2 = f3 = … = fN

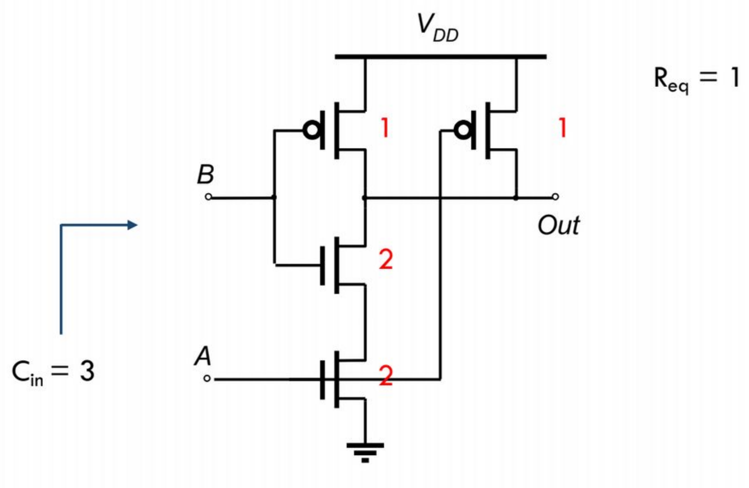
Each inverter has a fanout = the n-th root of F (total fanout)

Logical Effort



- g = logical effort
- To find g for a CMOS gate:
  1. Size the gate to have R_{eq} equal to that of an inverter
  2. Find the input capacitance of the gate (for a particular input)
  3. Take the ratio C_{in,gate} / C_{in,inv}
- For a NAND2 gate: g = 3/2
- For a NOR2 gate: g = 3/2
- How do these change if the PMOS is half as strong as an nMOS?
- Can the logical effort be different for different inputs?

Intrinsic Delay

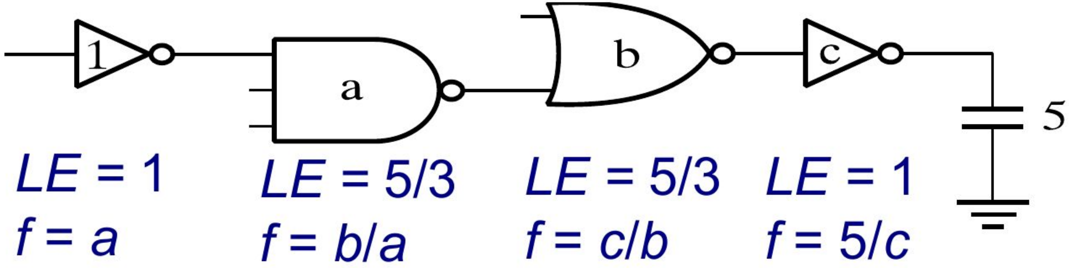


- p = intrinsic delay
- To find p for a CMOS gate:
  1. Size the gate to have R_{eq} equal to that of an inverter
  2. Find the output (intrinsic) capacitance of the gate
  3. Take the ratio C_{out,gate} / C_{out,inv}
- For a NAND2 gate: g = 4/2
- For a NOR2 gate: g = 4/2
- How do these change if the PMOS is half as strong as an nMOS?

Path Delay

- Path logical effort G = g1 * g2 * … gN
- Path fanout F = CL / C_{in,1}
  - This is called H in lecture
- Branching factor bi = ratio of total cap seen / on-path cap
- Path total effort H = GFB
  - This is called F in lecture
- We want to minimize the delay of a path through a series of gates
  - Each stage i in the chain has an effective fanout gi * fi
  - The solution is to make the effective fanout of each stage the same
  - EF_{opt} = H^{1/N}
  - EF = gi * fi = gi * (C_{load} / C_{in})
  - C_{in} (i.e. the gate size) = gi * C_{load} / EF
- The optimal path delay D = N * H^{1/N} + p1 + … + pN

Sizing a Logic Path for Minimum Delay



LE = 1
f = *a*

LE = 5/3
f = *b*/*a*

LE = 5/3
f = *c*/*b*

LE = 1
f = 5/*c*

Electrical fanout, F = 5
Π LE = 25/9
PE = 125/9
EF/stage = 1.93
a = 1.93
b = 2.23
c = 2.59

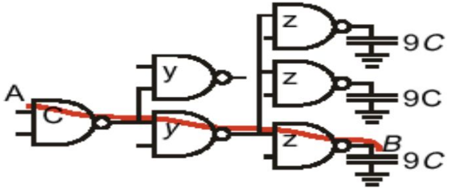
From the back

5/c = 1.93
(5/3)c/b = 1.93
(5/3)b/a = 1.93

- This example assumes the NMOS is twice as strong as the PMOS
- EF/stage is calculated as the 4th-root of G*F
- gi * fi = EF of stage i
- Easier to calculate gate sizes from the load cap backwards

Sizing a Logic Path for Minimum Delay (Branching)

Select gate sizes *y* and *z* to minimize delay from *A* to *B*



Logical Effort: $LE = (4/3)^3$

Electrical Fanout: $F = C_{out}/C_{in} = 9$

Branching Effort: $B = 2 \cdot 3 = 6$

Path Effort: $PE = \prod LE \cdot F \cdot B = 128$

Best Effective Fanout: $EF = PE^{1/3} \approx 5$

Delay: $D = 3 \cdot 5 + 3 \cdot 2 = 21$

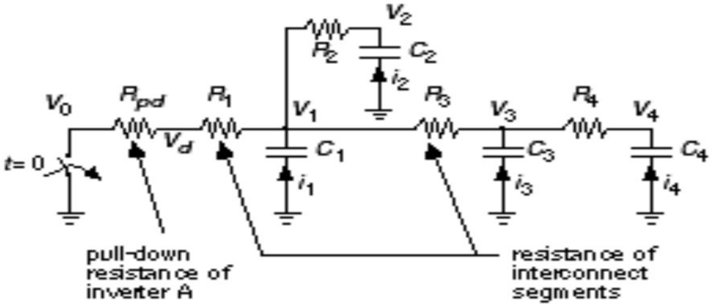
Work backward for sizes:

$$z = \frac{9C \cdot (4/3)}{5} = 2.4C$$

$$y = \frac{3z \cdot (4/3)}{5} = 1.9C$$

- This example also assumes the NMOS is twice as strong as the PMOS
- Note that when calculating the size y, you must consider the total load cap (3z) instead of just z
- Recall: D_{opt} = N * H^{1/N} + p1 + … + pN

Elmore Delay Approximation


V2
R2
C2
i2
V0
Rpd
R1
V1
R3
V3
R4
V4
Vd
C1
C3
C4
t=0
i1
i3
i4
pull-down resistance of inverter A
resistance of interconnect segments

$$
\begin{aligned}
\tau_{D4} = \sum_{k=0}^{4} C_k R_{4k} = {} & C_0 R_{pd} \; + \\
& C_1(R_{pd} + R_1) + \\
& C_2(Rpd + R_1) + \\
& C_3(Rpd + R_1 + R_3) + \\
& C_4(Rpd + R_1 + R_3 + R_4) = \\
R_{pd}(C_0 + C_1 + C_2 + C_3 + C_4) + \; & R_1(C_1 + C_2 + C_3 + C_4) \\
+ R_3(C_3 + C_4) + R_4 C_4 &
\end{aligned}
$$

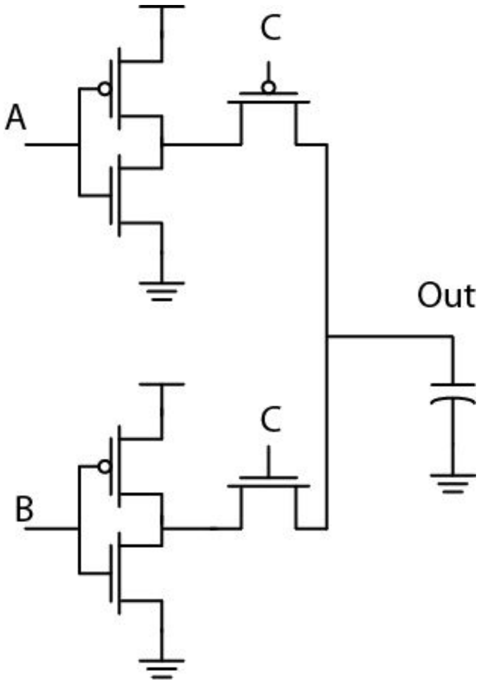
- Approximates the dominant time constant of a RC network for a given input and output node
- 2 methods (same result):
  1. Take every capacitor and multiply it by the sum of resistors on the path charging it
  2. Take every resistor on the path and multiply it by the sum of capacitances it charges

Energy and Power

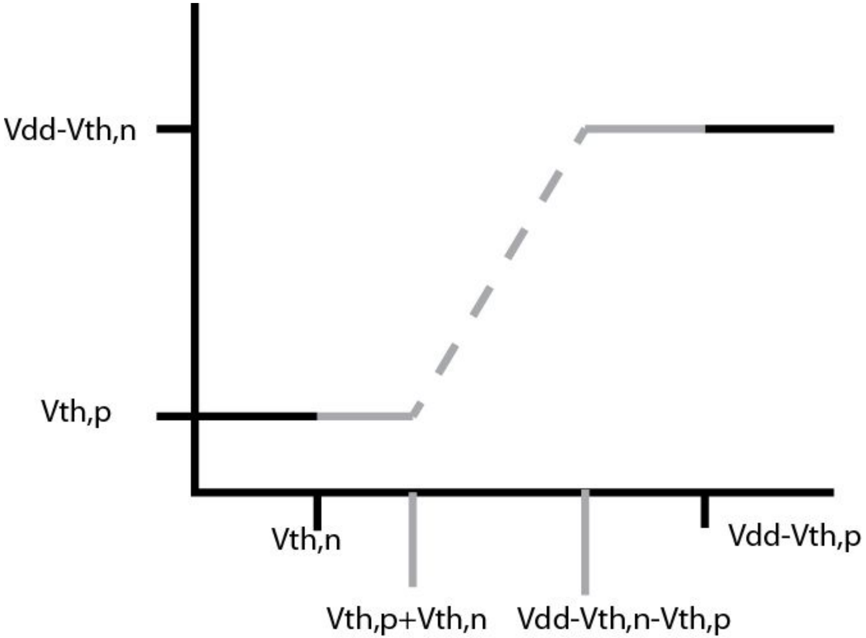
- CV^2 joules drawn from supply when charging cap to VDD
- ½ CV^2 stored on capacitor
- No additional charge drawn from supply when the cap discharges
- The other ½ CV^2 is burned on the charging resistance and dissipates as heat
- Dynamic switching power = f C V^2 alpha_{0 -> 1}

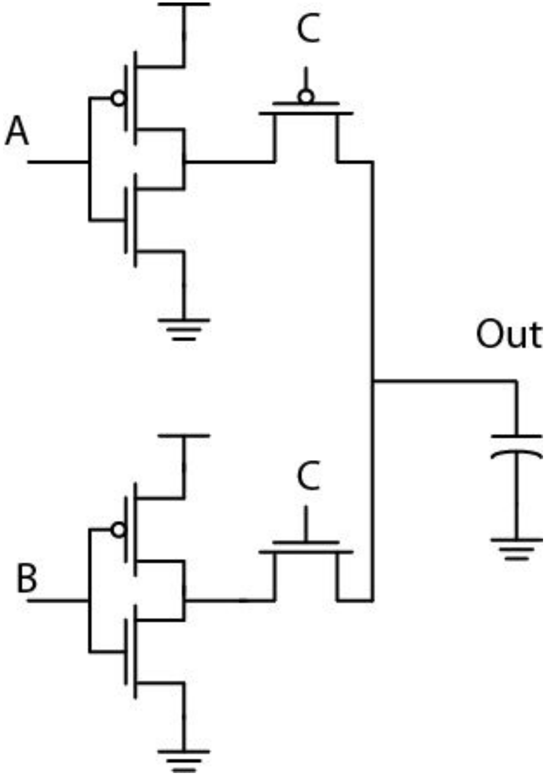
Problem

- Assume A = VDD, B = 0V
- Vthn = 0.2V
- |Vthp| = 0.3V
- Rp = Rn
- Draw a VTC when Vc is swept from 0 to VDD



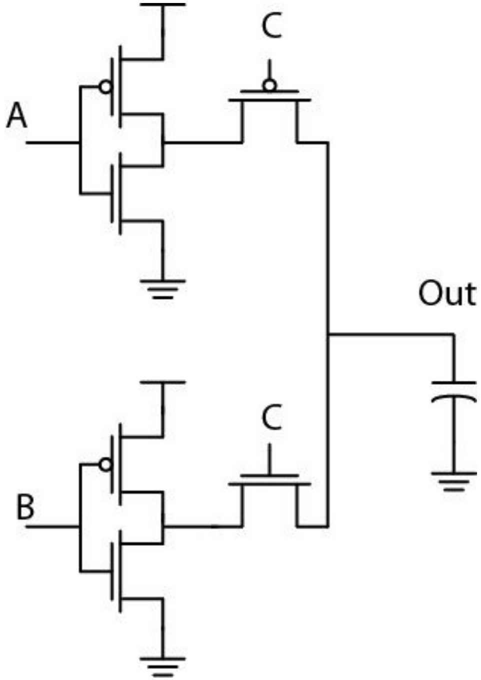
Problem





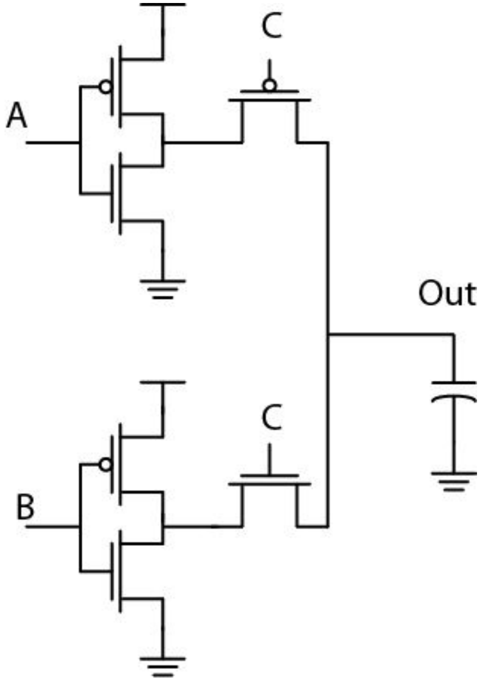
Problem

- What is the energy pulled from the supply when B = 1, C = 0, and A goes from 1 to 0? Assume CL is initially at 0V.
  - The output cap is charged from 0V to VDD
  - C * VDD^2
- Using the final voltage from the last part as the new initial condition, how much energy is dissipated when C goes from 0 to 1?
  - ½ C * VDD^2



Problem

- What is the energy pulled from the supply when A = 1, C = 1, and B goes from 1 to 0? Assume CL is initially 0V.
  - Recall E_VDD = CL * VDD * V_{delta}, where V_{delta} is the change in voltage across the capacitor
  - CL * VDD * (VDD - Vthn)
- Using the final voltage from the last part, how much energy is dissipated when C goes from 1 to 0?
  - ½ CL ((VDD - Vthn)^2 - (Vthp)^2)
  - Cap starts at VDD - Vthn and ends at Vthp. The energy difference is dissipated.



Adders

- Let’s look at last semester’s discussion slides: http://inst.eecs.berkeley.edu/~eecs151/sp19/files/discussion11.pdf