CMOS Logic Gates
a delay model

Introducing logical effort

Lecture 5

Weekly schedule

- Week 1
  - Intro & the transistor
- Week 2
  - The CMOS inverter – one or more (tapered buffer)
- Week 3
  - Lab 1 Inverter schematic entry
  - Delay model for CMOS gates – optimal path delay + ILAs
- Week 4
  - Lab 2 Carry circuit ILA schematic entry
  - Physical design, layout, geometrical design rules
- Week 5
  - Lab 3 Carry circuit ILA layout and verification
  - Wires, delay with wires & gates
- Week 6
  - Lab 4 Clock tree w. wires text entry & simulation
  - Sequential & adders
- Week 7
  - Power & speeding up adders
- Week 8
  - Review

Week 3

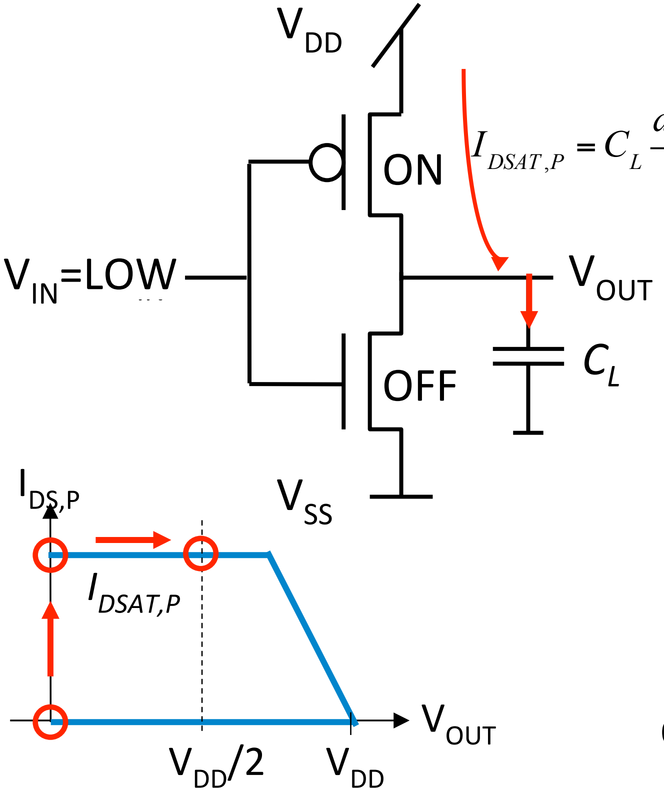
- Monday lab1
  - CMOS inverter static and dynamic
- Tuesday
  - Lecture Delay with gates, logical effort
  - Postlab review lab 1
- Thursday
  - Prelab lab 2, Lecture on optimal path delay
  - Tutorial POTW (Victor)
- Friday Deadline prelab 2 –
  - Schematic entry of carry ckt of full adder

From MUD cards

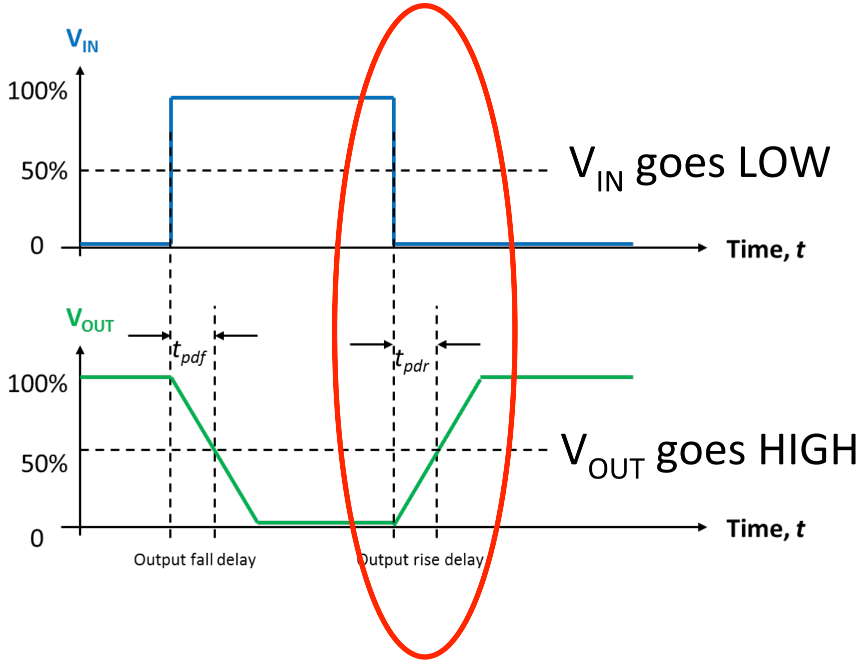
- Definitions
  - Why is rise and fall time defined as 20 % to 80 %?
  - Why is propagation delay defined from 50 % to 50 %?
- Model of inverter
  - Explanation of the step response model
  - How to derive the equivalent model of one inverter
  - Inverter output changes from input voltage confusing
- One inverter driving on other inverter
  - How resistance R_{eff} and C_G could give time constant?
  - The relationship between C_L and width.
- Fanout-of-four (FO4) delay
  - How to derive the delay
  - How does scaling work with FO4
- Tapered buffer (will review on Thursday)
  - What is fanout (H) and the concept of optimum fanout.
  - How calculate the number of inverters needed to minimize delay.

Step-response model

1. Charging the load capacitor through the p-channel MOSFET



$$I_{DSAT,P} = C_L \frac{dV_{OUT}}{dt} = C_L \frac{\Delta V_{OUT}}{\Delta t}$$

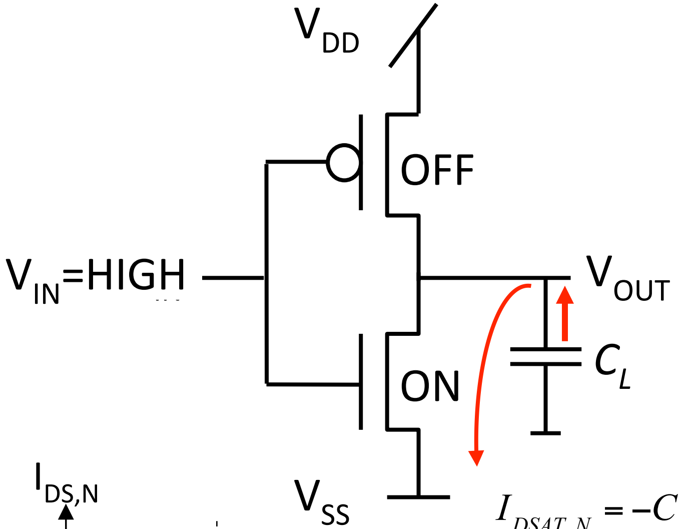


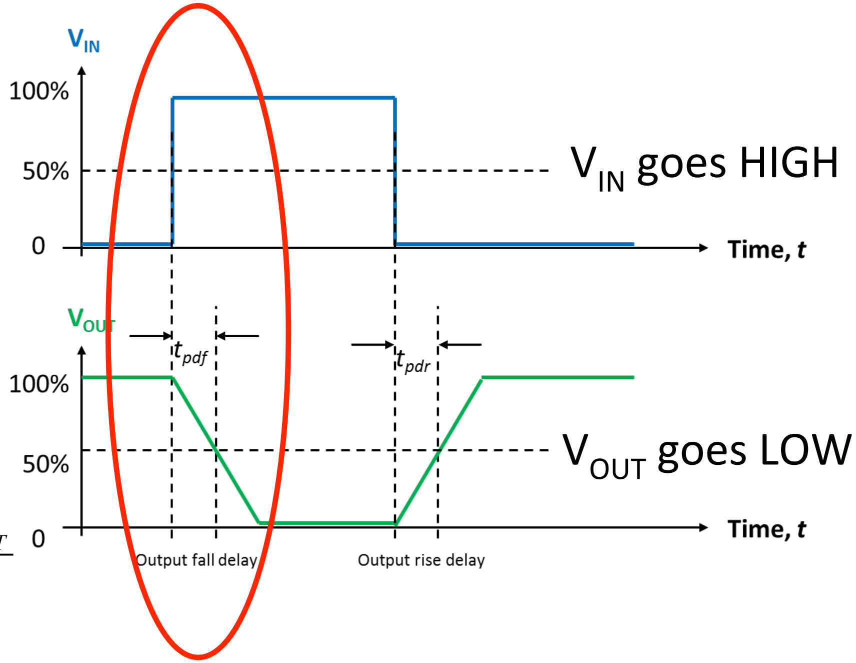
Output rise delay $\Delta t = t_{pdr} = C_L \frac{\Delta V_{OUT}}{I_{DSAT,P}} = C_L \frac{V_{DD}/2}{I_{DSAT,P}}$

pMOS current flow in detail

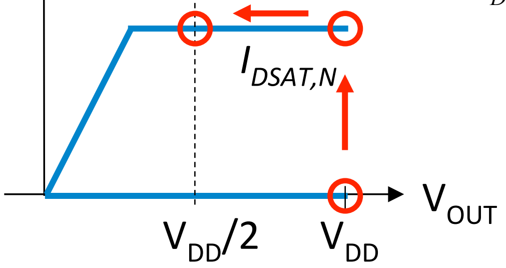
Step-response model

2. Discharging the load capacitor through the n-channel MOSFET





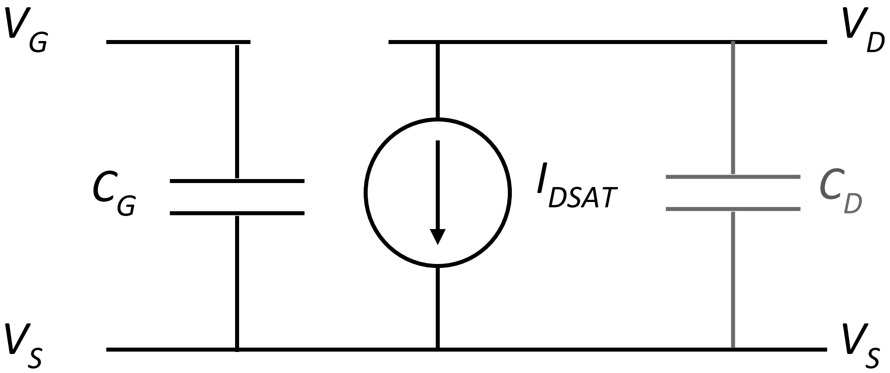
$$I_{DSAT,N} = -C_L \frac{dV_{OUT}}{dt} = -C_L \frac{\Delta V_{OUT}}{\Delta t}$$



nMOS current flow in detail

Output fall delay $$\Delta t = t_{pdf} = C_L \frac{\Delta V_{OUT}}{I_{DSAT,N}} = C_L \frac{V_{DD}/2}{I_{DSAT,N}}$$

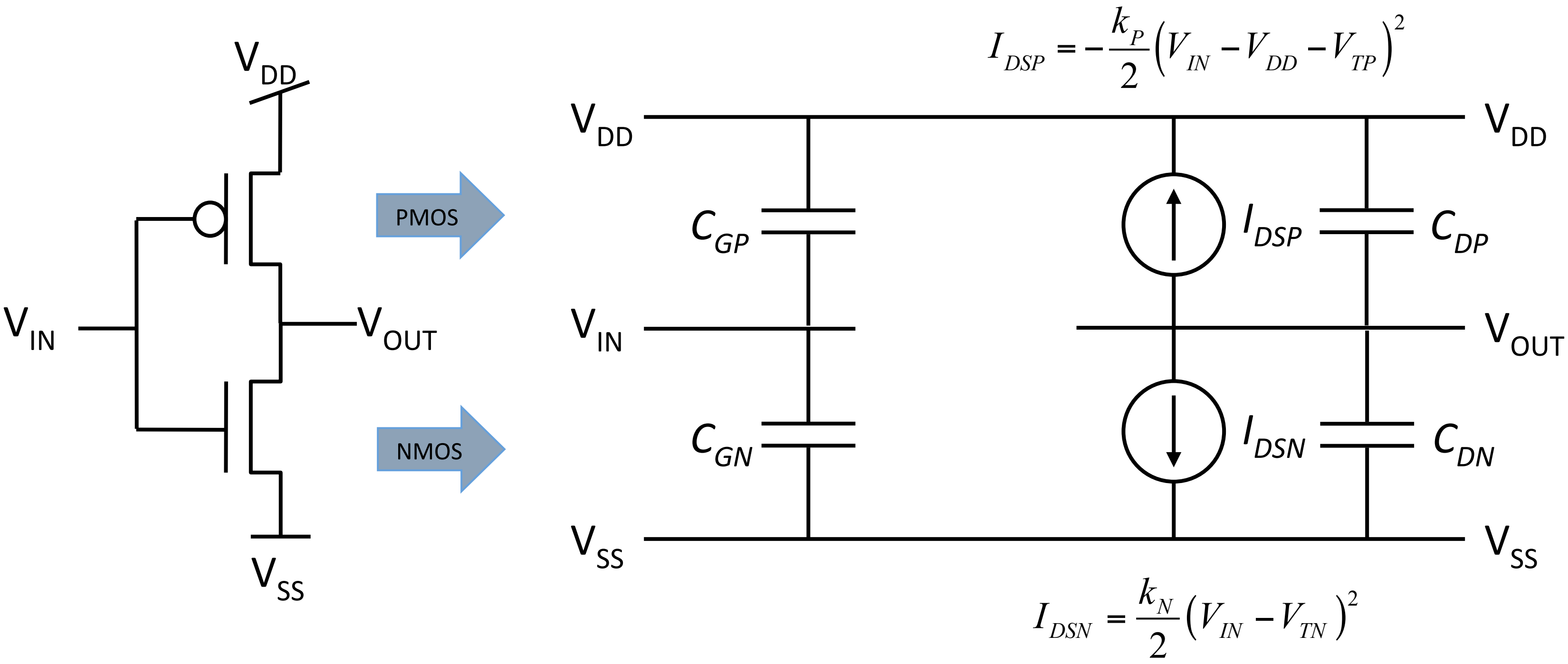
Electrical model for saturated MOSFET (from lecture 2)



Model for saturated MOSFET
Valid for both nMOS and pMOS transistors

The inverter and its electrical model

Replace the MOSFETs with their equivalent electrical circuits!



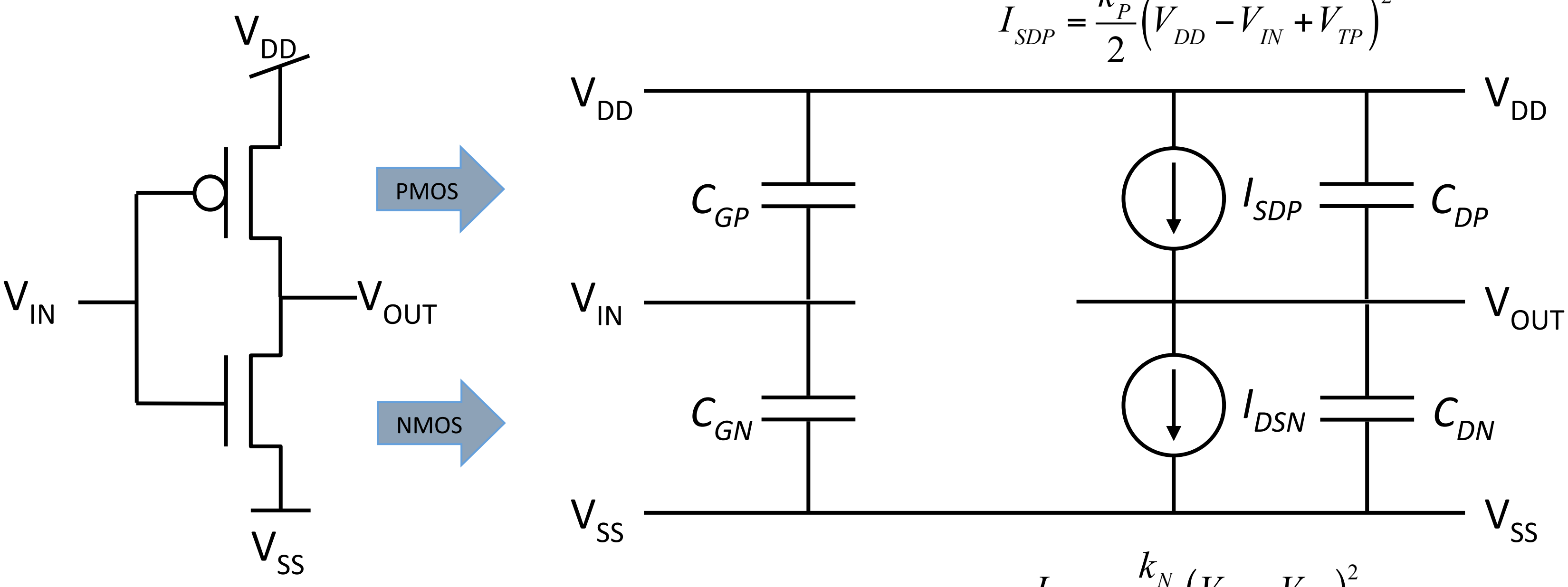
$$I_{DSP} = -\frac{k_P}{2}\left(V_{IN} - V_{DD} - V_{TP}\right)^2$$

$$I_{DSN} = \frac{k_N}{2}\left(V_{IN} - V_{TN}\right)^2$$

The inverter and its electrical model

Change the sign of the pMOS current so that all currents are positive
(We also made the pMOS voltages positive, although it is not necessary)

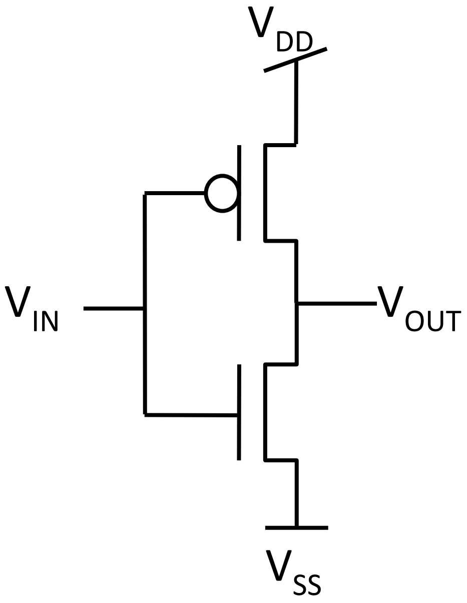
$$I_{SDP} = \frac{k_P}{2}\left(V_{DD} - V_{IN} + V_{TP}\right)^2$$



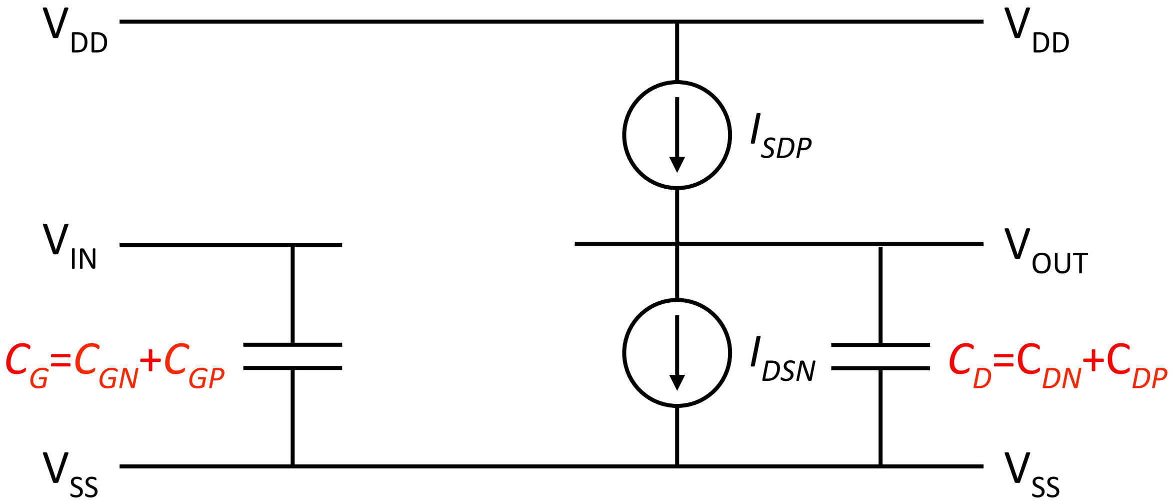
$$I_{DSN} = \frac{k_N}{2}\left(V_{IN} - V_{TN}\right)^2$$

The inverter and its electrical model

Place all capacitors to signal ground! Both rails are constant voltages, no dV/dt



$$I_{SDP} = \frac{k_P}{2}\left(V_{DD} - V_{IN} + V_{TP}\right)^2$$

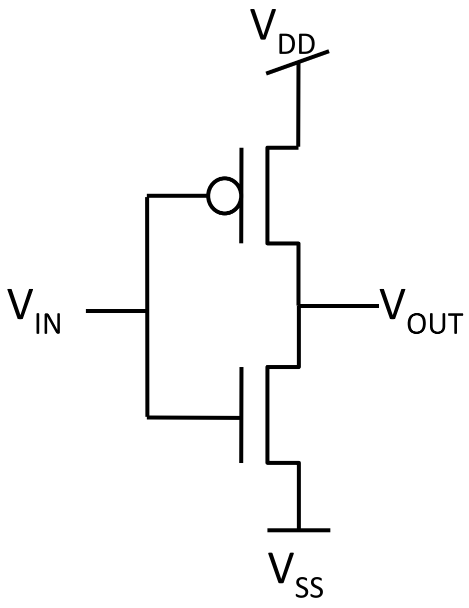


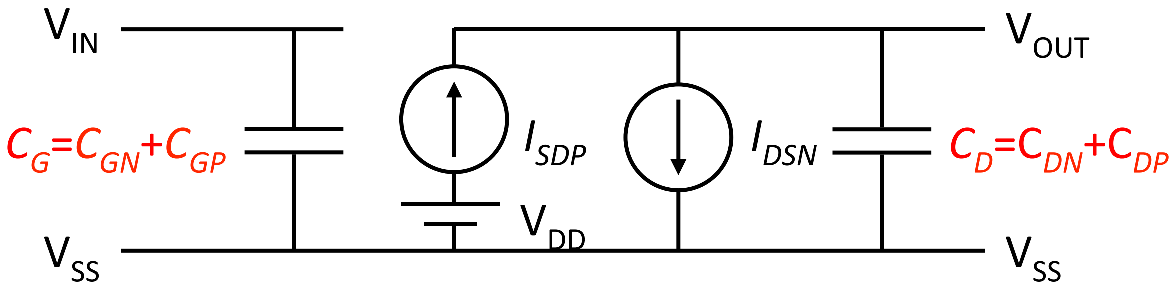
$$I_{DSN} = \frac{k_N}{2}\left(V_{IN} - V_{TN}\right)^2$$

Inverter input capacitance: $C_G=C_{GN}+C_{GP}$; MOSFET gate capacitances add!
Inverter parasitic output capacitance: $C_D=C_{DN}+C_{DP}$. Drain caps also add!

The inverter and its electrical model

Eliminate V_{DD} rail by inserting power supply to signal ground!

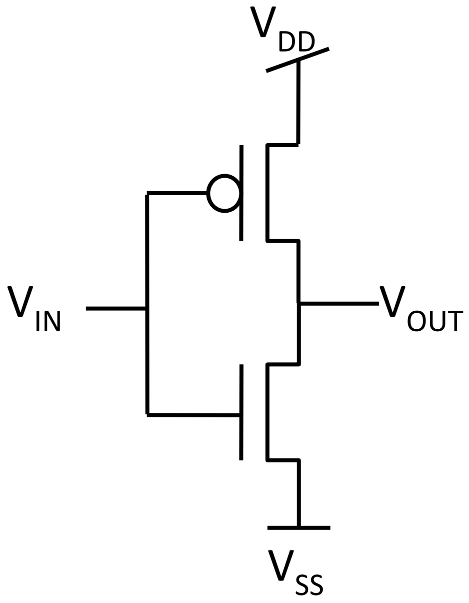




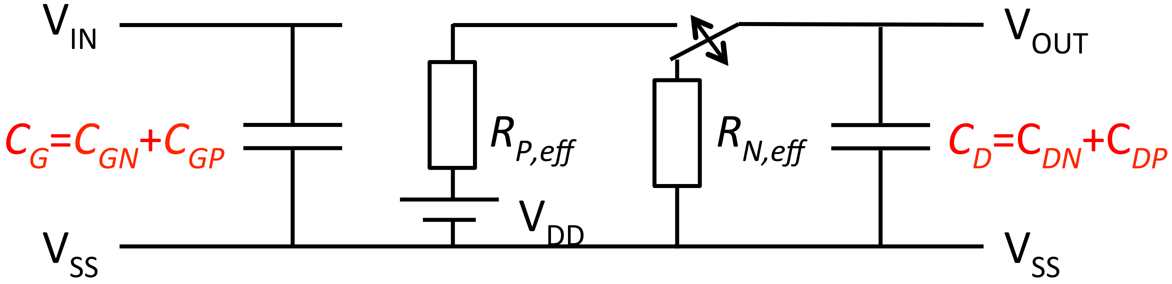
Inverter input capacitance: $C_G=C_{GN}+C_{GP}$; MOSFET gate capacitances add!
Inverter parasitic output capacitance: $C_D=C_{DN}+C_{DP}$. Drain caps also add!

The inverter and its electrical model

Replace MOSFET constant-current sources with their effective resistances!



$R_{N,eff}$=2 kΩ·μm $R_{P,eff}$=4 kΩ·μm

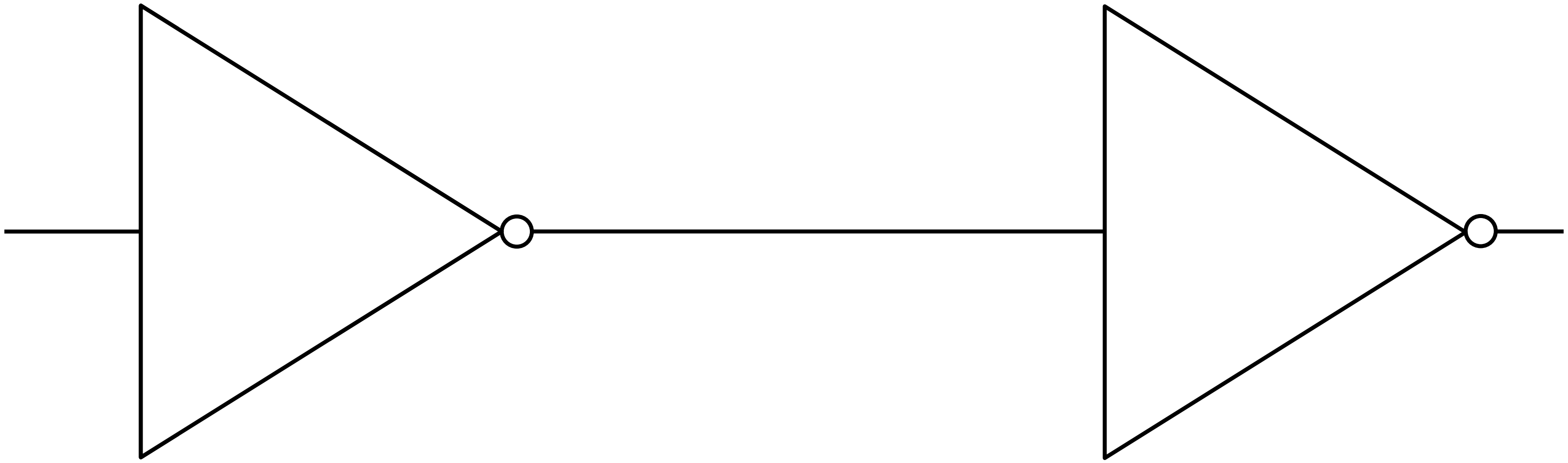


Inverter input capacitance: $C_G=C_{GN}+C_{GP}$; MOSFET gate capacitances add!
Inverter parasitic output capacitance: $C_D=C_{DN}+C_{DP}$. Drain caps also add!

Inverter pair delay

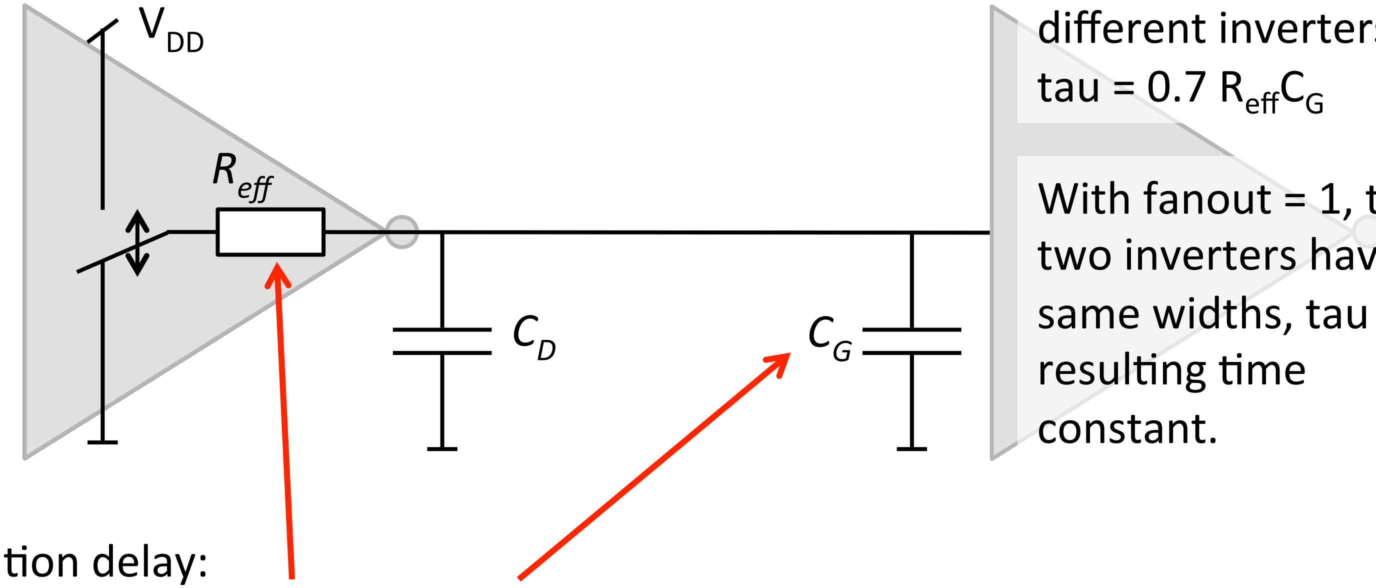
Task: Calculate the inverter pair delay!

Note: identical inverters



Inverter pair delay

Equivalent electrical circuit for propagation delay calculations



R_{eff} and C_G come from different inverters to form tau = 0.7 $R_{eff}C_G$

With fanout = 1, that is the two inverters having the same widths, tau is the resulting time constant.

Propagation delay:

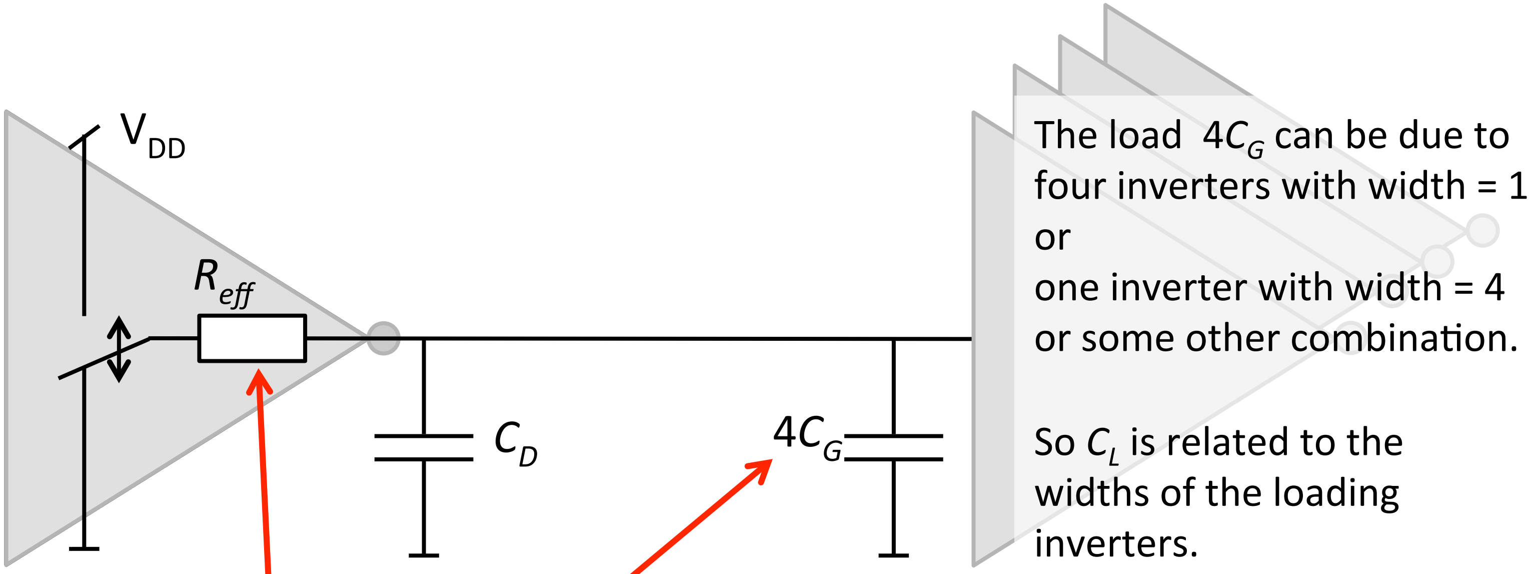
$$t_{pd} = 0.7R_{eff}\left(C_D + C_G\right) = 0.7R_{eff}C_G\left(p_{inv} + 1\right) = 5\text{ ps} \times 2 = 10\text{ ps}$$

All delay calculations are made wrt to this technology time constant tau (τ)

$$\tau = 0.7R_{eff}C_G = 0.7 \times \left(2\text{ k}\Omega \times \mu\text{m}\right) \times \left(3.6\,fF / \mu\text{m}\right) = 5\text{ ps}$$

FO4 delay

Equivalent electrical circuit for propagation delay calculations



The load $4C_G$ can be due to four inverters with width = 1
or
one inverter with width = 4
or some other combination.

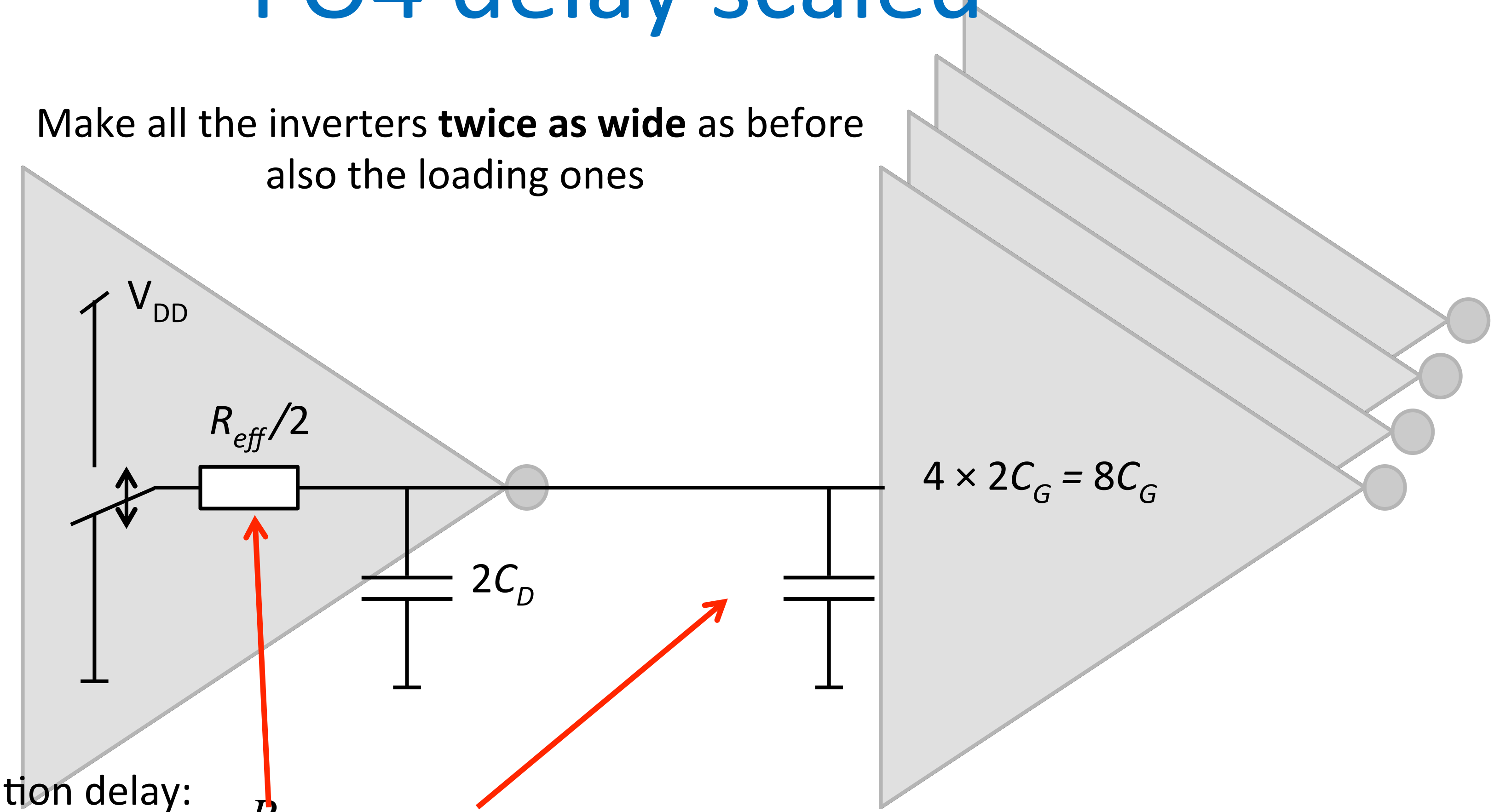
So C_L is related to the widths of the loading inverters.

Propagation delay:

$$t_{pd} = 0.7R_{eff}\left(C_D + 4C_G\right) = 0.7R_{eff}C_G\left(p_{inv} + 4\right) \approx 5\text{ ps} \times 5 = 25\text{ ps}$$

FO4 delay scaled

Make all the inverters **twice as wide** as before
also the loading ones

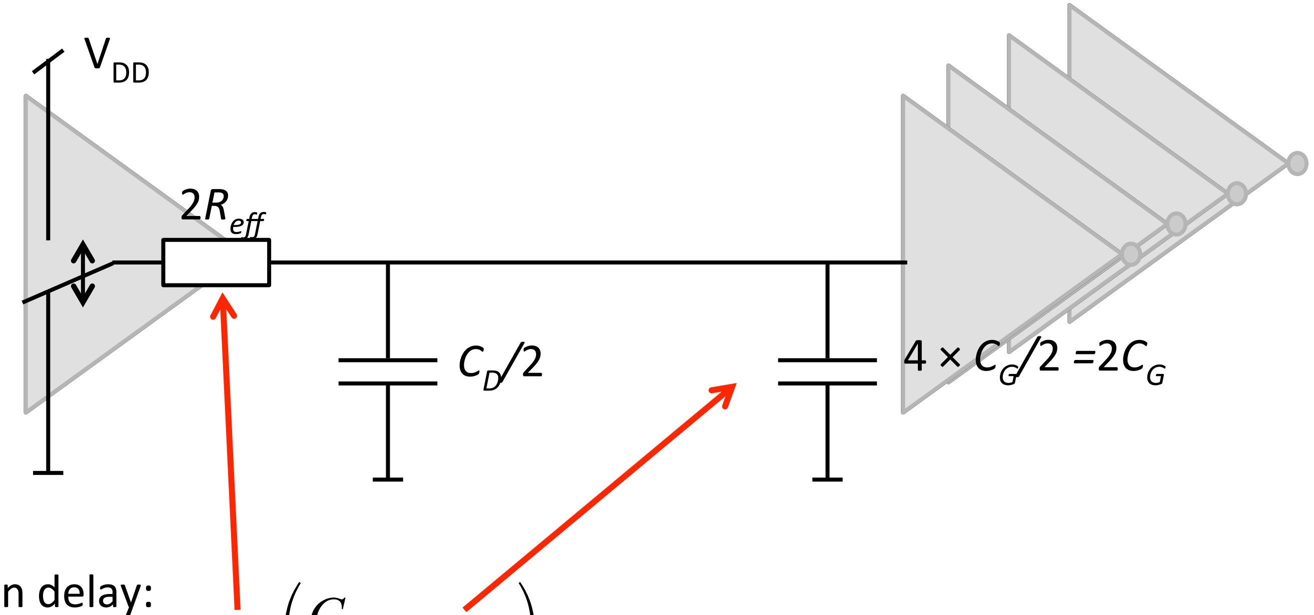


Propagation delay:

$$t_{pd} = 0.7\cdot\frac{R_{eff}}{2}\left(2C_D + 8C_G\right) = 0.7R_{eff}C_G\left(p_{inv} + 4\right) \approx 5\text{ ps}\times 5 = 25\text{ ps}$$

FO4 delay scaled

Make all the inverters **half as wide** as before – also the loading ones

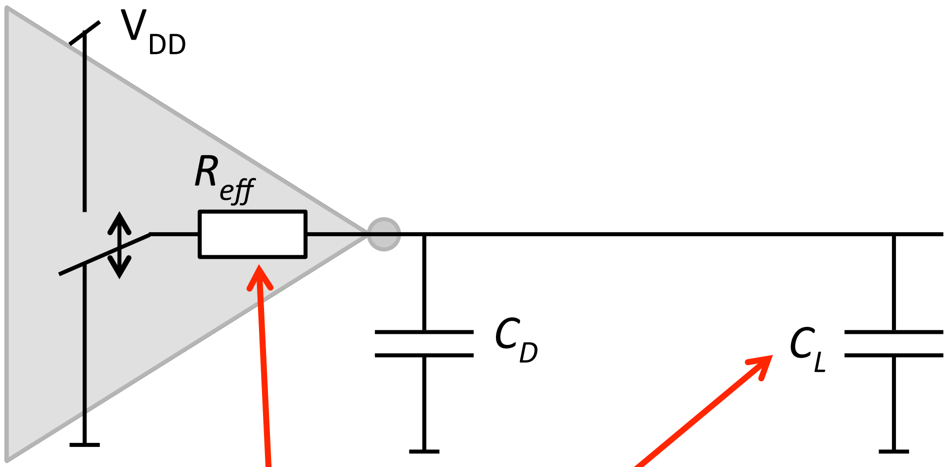


Propagation delay:

$$t_{pd} = 0.7 \cdot 2R_{eff}\left(\frac{C_D}{2} + 2C_G\right) = 0.7 R_{eff} C_G \left(p_{inv} + 4\right) \approx 5\ \text{ps} \times 5 = 25\ \text{ps}$$

Delay with any load

Equivalent electrical circuit for propagation delay calculations



C_L is the placeholder for the load capacitance.
C_L can be due to anything capacitive connected to the driving inverter’s output.

If C_L stems from inverters then C_L is related to the widths of the loading inverters.

Propagation delay:

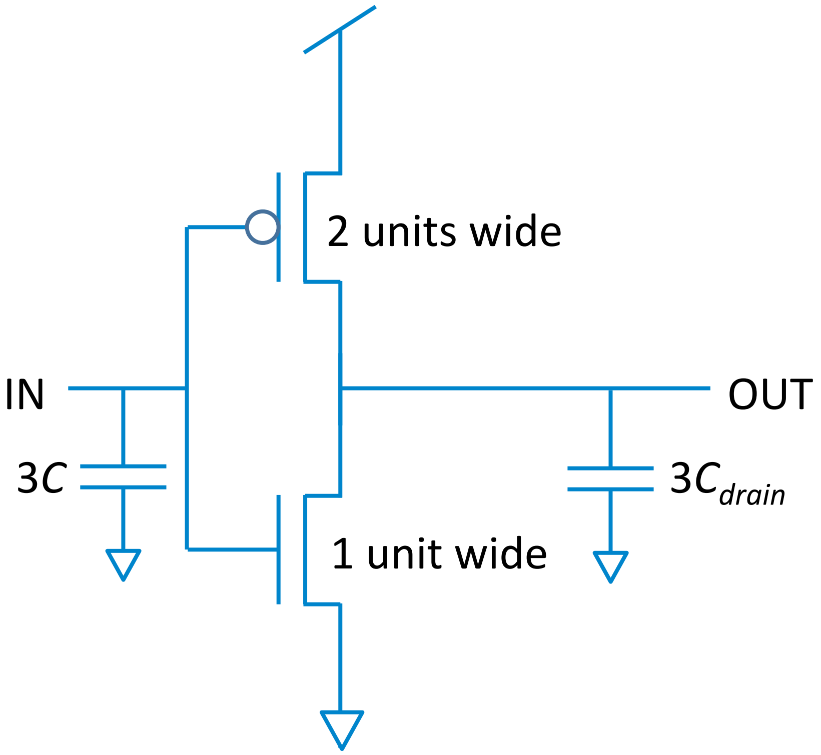
$$t_{pd} = 0.7R_{eff}\left(C_D + C_L\right) = 0.7R_{eff}C_G\left(p_{inv} + \frac{C_L}{C_G}\right) = 0.7R_{eff}C_G\left(p_{inv} + h\right)$$

h = electrical effort

Background

- In previous lecture, lecture 4, we have developed a propagation delay model for CMOS inverters.
- For equal rise and fall delays, we have decided to use p-channel devices twice as wide as the n-channel device.
- An n-channel MOSFET of unit width was assumed to have effective resistance *R* and gate capacitance *C*.
- Hence, a p-channel device of two units width has the same effective resistance *R*, but gate capacitance 2*C*.
- The inverter designed with such MOSFETs has input capacitance 3*C*, and equal pull-up and pull-down resistances.

CMOS inverter –Transistor sizing

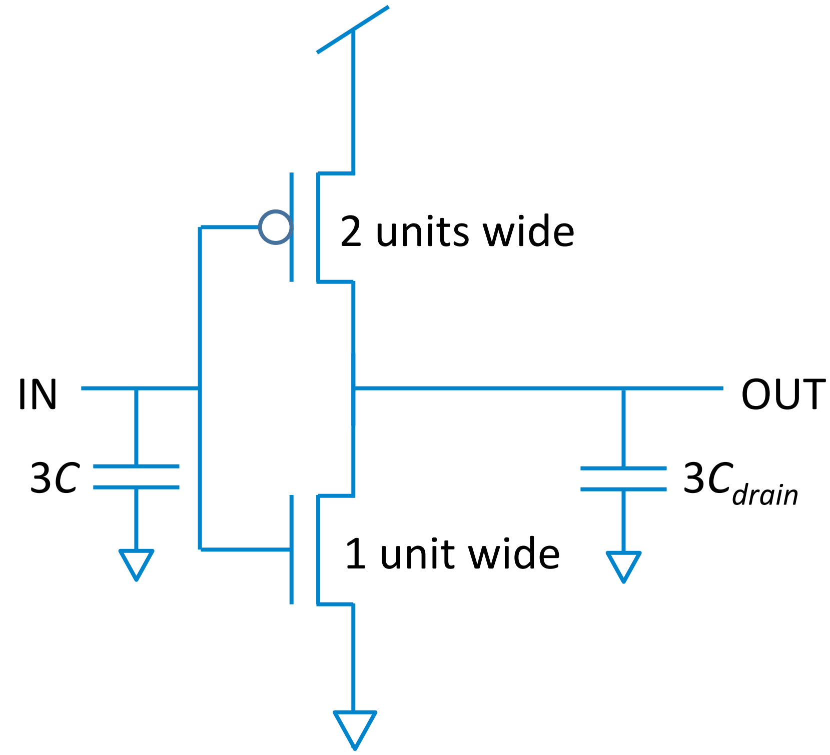


Note:

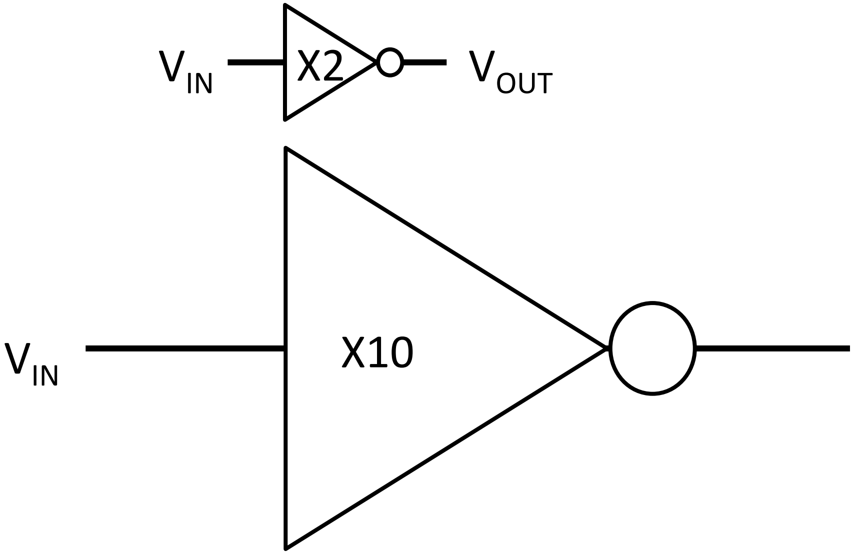
C is the gate capacitance of a MOSFET 1 unit wide

C_{drain} is the drain capacitance of a MOSFET 1 unit wide

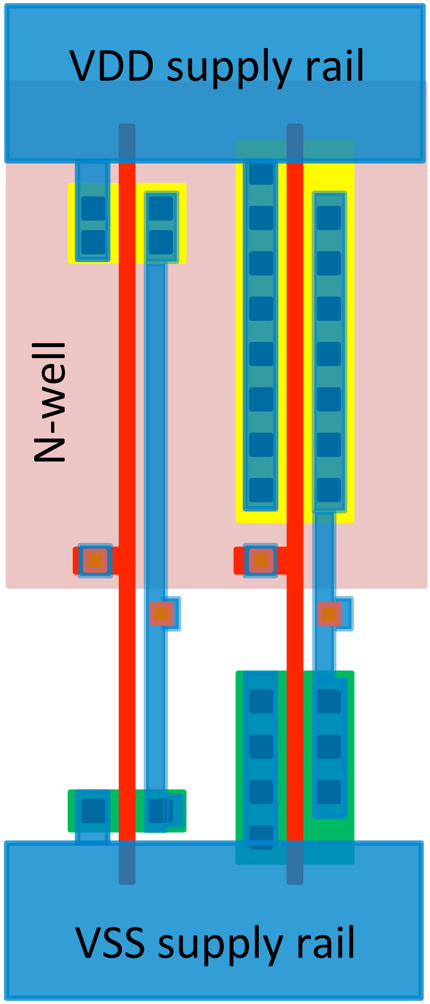
CMOS inverter – cell sizing



Size X	WN [nm]	WP [nm]
2	200	400
3	280	560
4	370	740
5	510	1020
8	720	1440
10	1000	2000

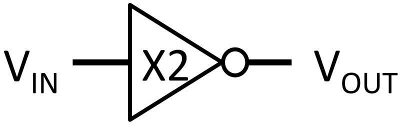


CMOS inverter – cell sizing

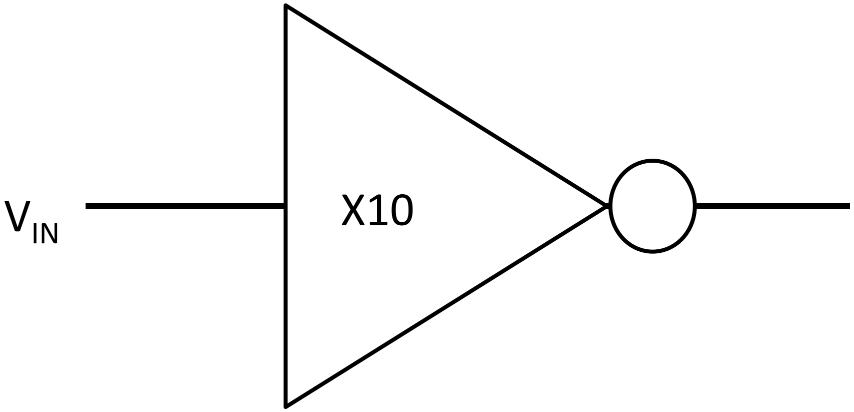


In ST cell library, size X refers to the input cap (and for inverters also to the driving capability)!

Size X	WN [nm]	WP [nm]
2	200	400
3	280	560
4	370	740
5	510	1020
8	720	1440
10	1000	2000

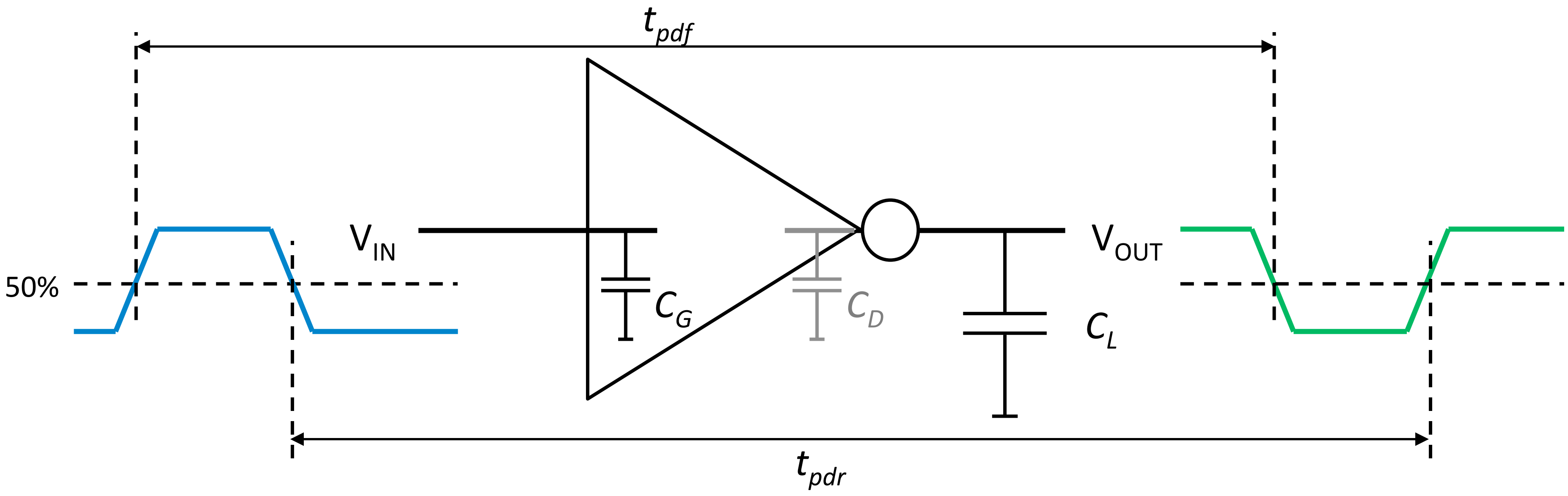


V_{IN}



Inverter propagation delay model with load

Propagation delay definitions: rise and fall delays

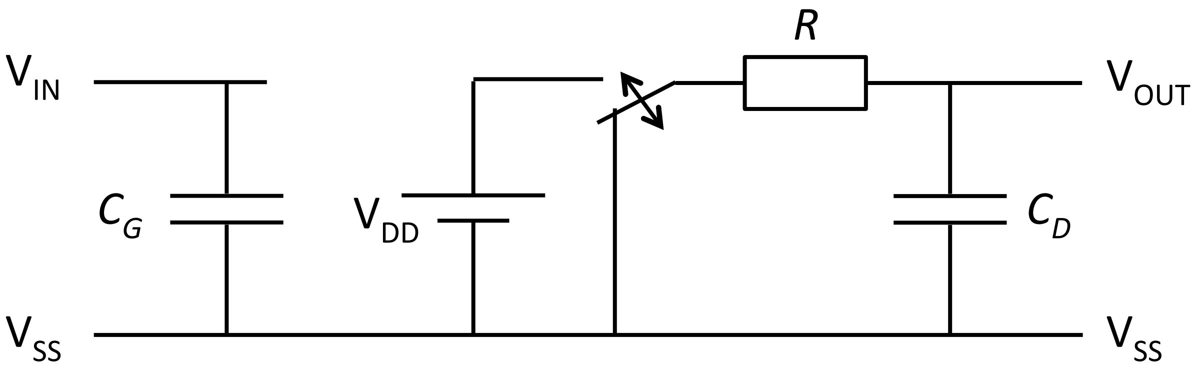


$$t_{pd} = 0.7\left(C_D + C_L\right)\frac{V_{DD}}{I_{DSAT}} = \underbrace{0.7R_{eff}C_G}_{\text{tau}}\left(\frac{C_D}{C_G} + \frac{C_L}{C_G}\right) = \tau\left(p_{inv} + h\right)$$

We defined time constant tau equal to 5 ps in 65 nm CMOS technology. Furthermore, we introduced **parasitic delay** $p_{inv}=C_D/C_G$, and **electrical effort** $h=C_L/C_G$.

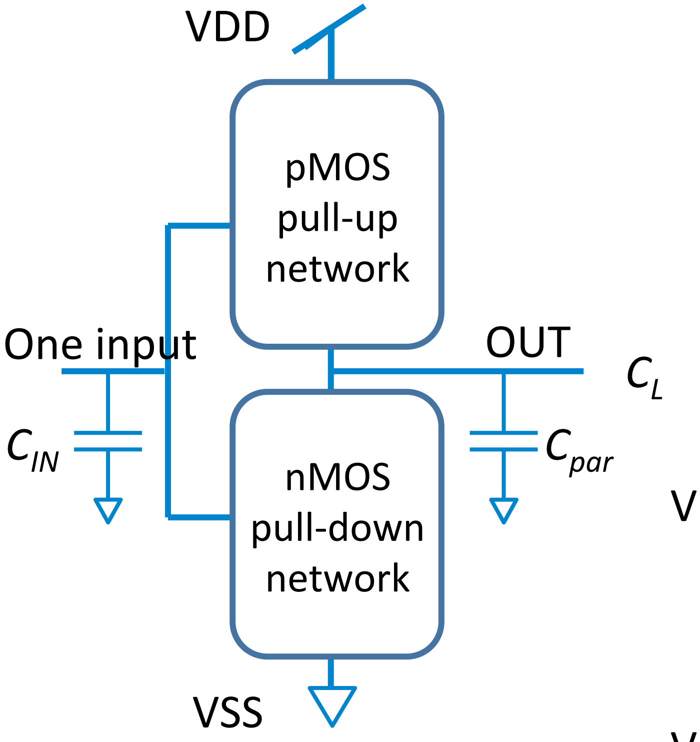
Inverter propagation delay model

Two-port electrical representation of propagation delay model

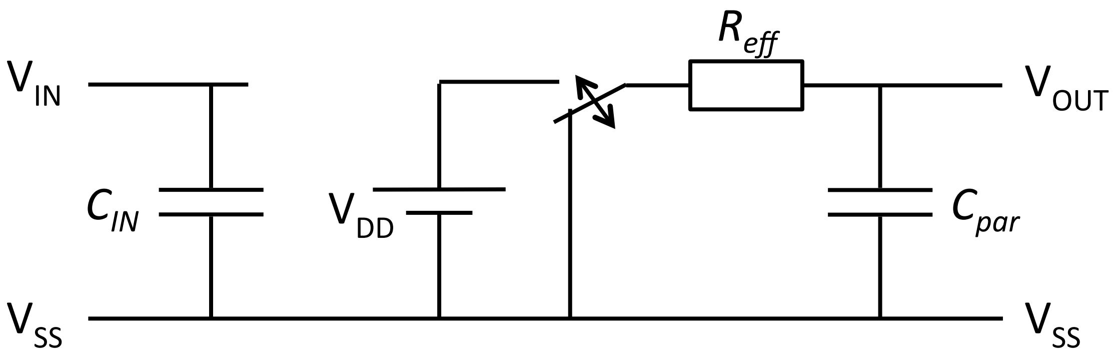


Logic gate propagation delay model

Now we want to apply the same model to any CMOS logic gate



The pull-up and pull-down networks may have many different paths, but we want to design for equal effective resistance R_{eff} in all paths!
But . . . what price do we have to pay for this in terms of input capacitance, C_{IN}?
And in parasitic capacitance, C_{par}?

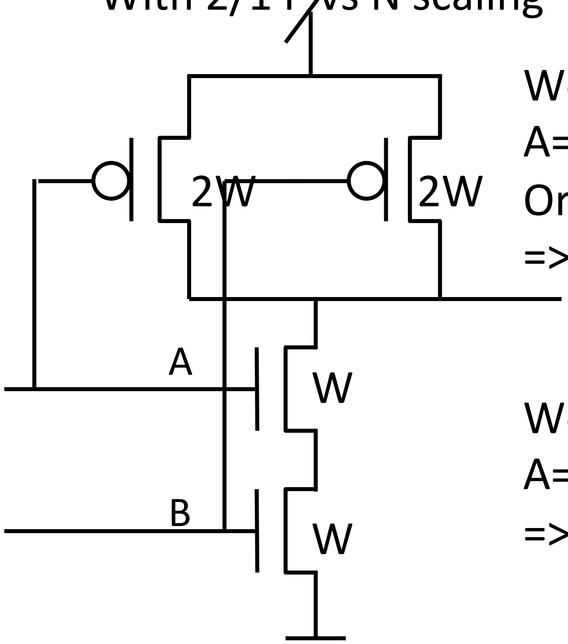


C_{IN} is the input capacitance for one of the inputs to the logic gate
C_{par} is the total parasitic capacitance at the gate output

Prerequisite: same worst-case resistance in all paths

Example: 2–input NAND gate

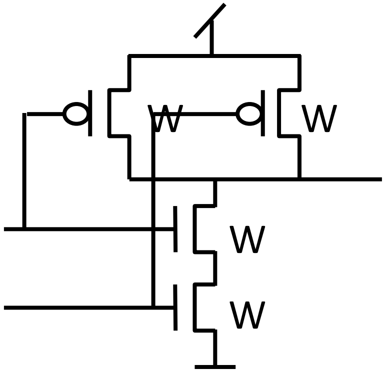
With 2/1 P vs N scaling

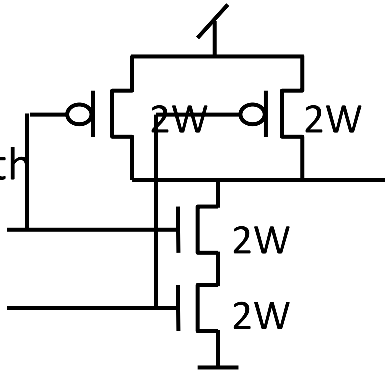


Worst case:
A=0 & B=1
Or B=1 & A=1
=> $R_{eff}/2$

Worst case:
A=1 & B=1
=> R_{eff}

Conclusion:
Need to change widths in p-net or n-net (or both).



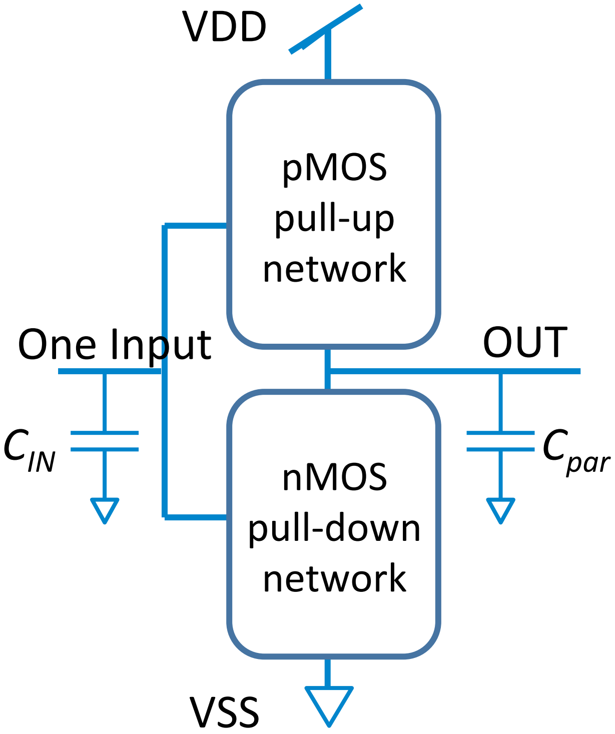


Many possible solutions with same worst-case resistance in n-net and p-net

If we don’t scale for same resistance we cannot use equivalent gate model with R_{eff} to the right of the switch.

Logic gate propagation delay model

Now we want to apply the same model to any CMOS logic gate

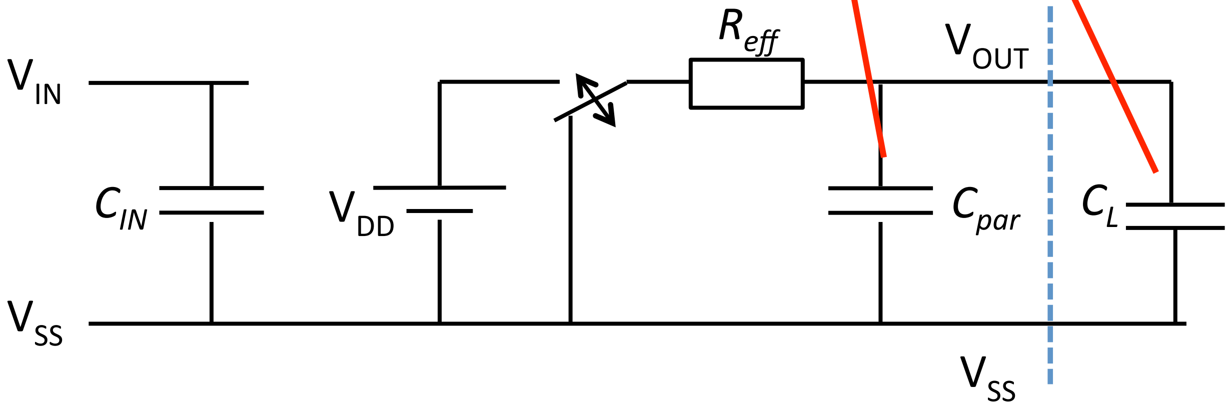


Assume all pull-up/down paths have same resistance R_{eff}. Obviously, the logic gate will have larger RC product than the inverter! What is the delay with load C_L from one of the inputs to the logic gate?

As before, normalize to process time constant tau!

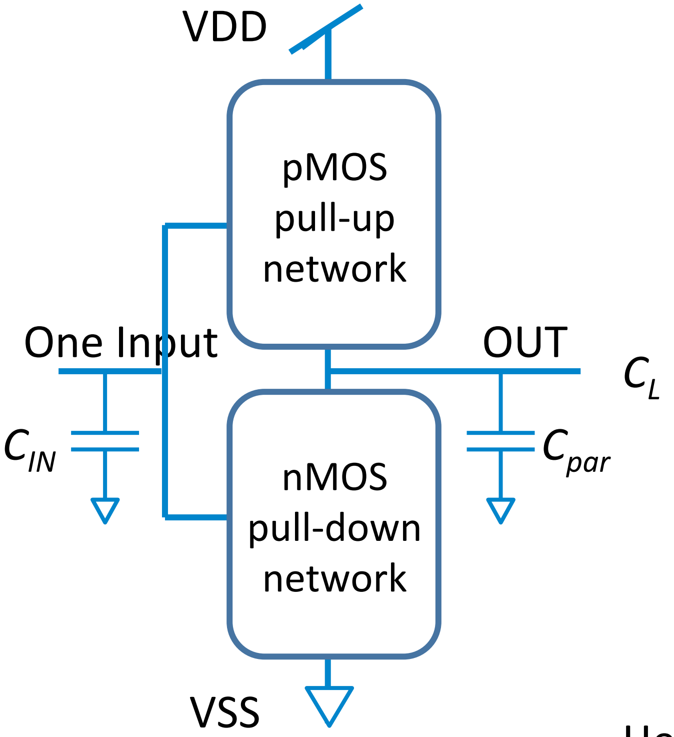
C_L

$$t_{pd} = 0.7R_{eff}\left(C_{par} + C_L\right) = \tau\left(\frac{R_{eff}C_{par}}{\left(RC_G\right)_{inv}} + \frac{R_{eff}C_L}{\left(RC_G\right)_{inv}}\right)$$



Logic gate propagation delay model

Now we want to apply the same model to any CMOS logic gate



Consider one delay term at a time!

$$t_{pd} = 0.7R_{eff}\left(C_{par} + C_L\right) = \tau\left(\frac{R_{eff}C_{par}}{\left(RC_G\right)_{inv}} + \frac{R_{eff}C_L}{\left(RC_G\right)_{inv}}\right)$$

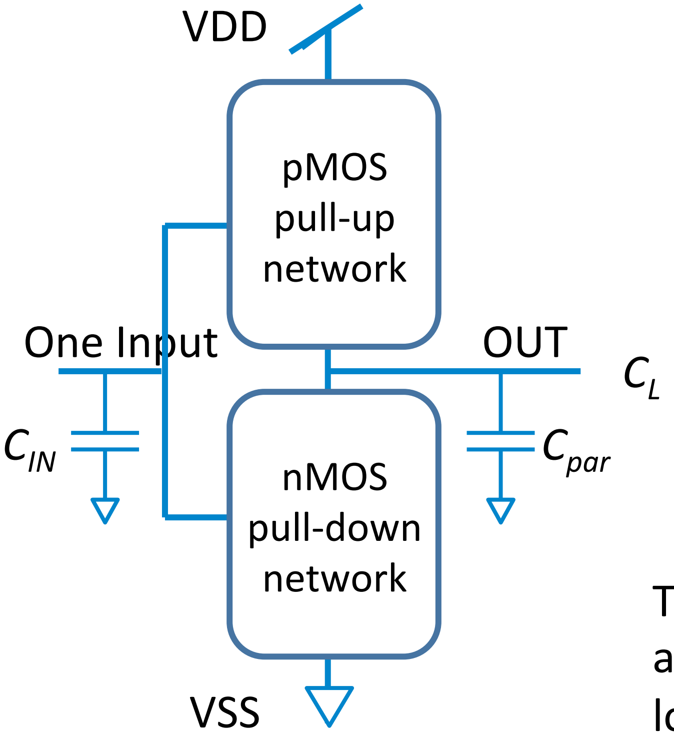
$$\text{parasitic delay: } p = \frac{R_{eff}C_{par}}{\left(RC_G\right)_{inv}} = \frac{R_{eff}C_{par}}{\left(RC_D\right)_{inv}}\underbrace{\frac{C_D}{C_G}}_{p_{inv}}$$

$$\text{stage effort: } f = \frac{R_{eff}C_L}{\left(RC_G\right)_{inv}} = \underbrace{\frac{R_{eff}C_{IN}}{\left(RC_G\right)_{inv}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

Here, we have defined the **logical effort** of a logic gate, *g*
The logical effort tells us how much larger the logic gate RC product is wrt inverter RC product!

Logic gate propagation delay model

To simplify, we suggest sizing MOSFETs for equal effective resistances, i.e. $R_{eff}=R$



Consider one delay term at a time!

$$\text{parasitic delay: } p = \frac{RC_{par}}{\left(RC_G\right)_{inv}} = \frac{C_{par}}{C_D}\underbrace{\frac{C_D}{C_G}}_{p_{inv}}$$

$$\text{stage effort : } f = \frac{RC_L}{\left(RC_G\right)_{inv}} = \underbrace{\frac{C_{IN}}{C_G}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

This method conveniently separates the driving strength of a logic gate in terms of its **logical effort**, ***g***, and its external load in terms of its **electrical effort, *h***. And all with respect to the properties of the inverter.

Overview important concepts

$$d = f + p$$

$$f = gh$$

d: (normalized) stage delay

f: stage effort

g: logical effort

h: electrical effort

p: parasitic delay

All concepts related
to **reference inverter**
with $(RC_G)_{inv}$, $p_{inv} = C_{Dinv}/C_{Ginv}$

The gate itself has parameters
C_{IN}, R_{eff} and C_{par}
and is connected to load C_L

Note:
C_{IN} can be different for different gate inputs =>
Different *g* for different inputs!

If $R_{eff} \neq R_{inv}$

$$f = \underbrace{\frac{R_{eff}C_{IN}}{(RC_G)_{inv}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

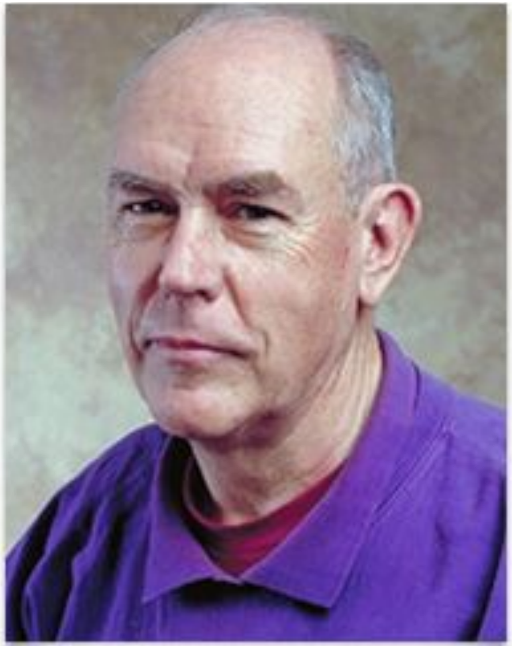
$$p = \frac{R_{eff}C_{par}}{(RC_D)_{inv}}\underbrace{\frac{C_{Dinv}}{C_G}}_{p_{inv}}$$

If $R_{eff} = R_{inv}$

$$f = \underbrace{\frac{C_{IN}}{C_{Ginv}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

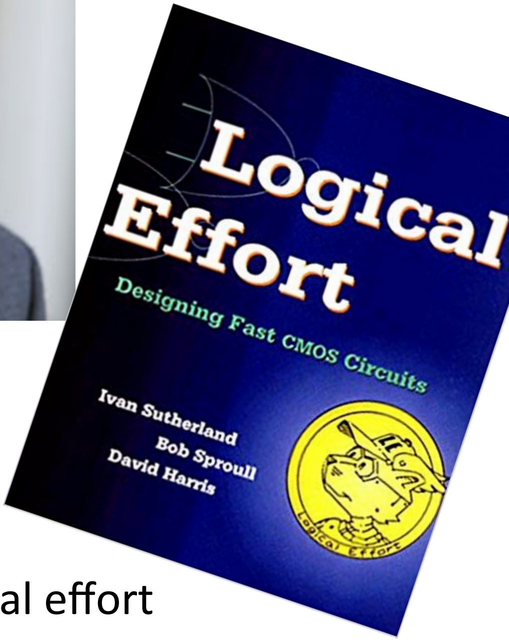
$$p = \frac{C_{par}}{C_{Dinv}}\underbrace{\frac{C_{Dinv}}{C_{Ginv}}}_{p_{inv}}$$

Who invented logical effort?



Ivan Sutherland

Bob Sproull



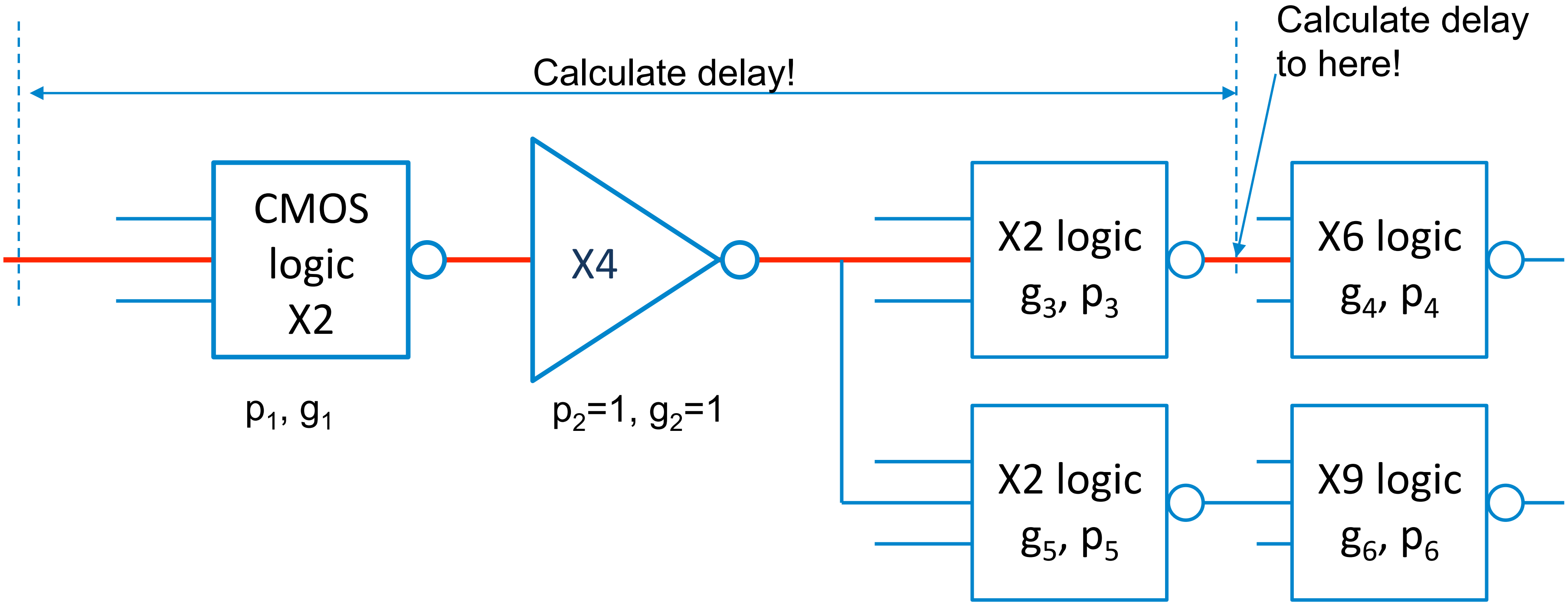
“The father of computer graphics”
Together they worked on head-mounted displays.
Work on graphics hardware spurred the invention of logical effort

Quiz time!

Go to socrative.com
Select Student login
Go to room: “MCC0922018”

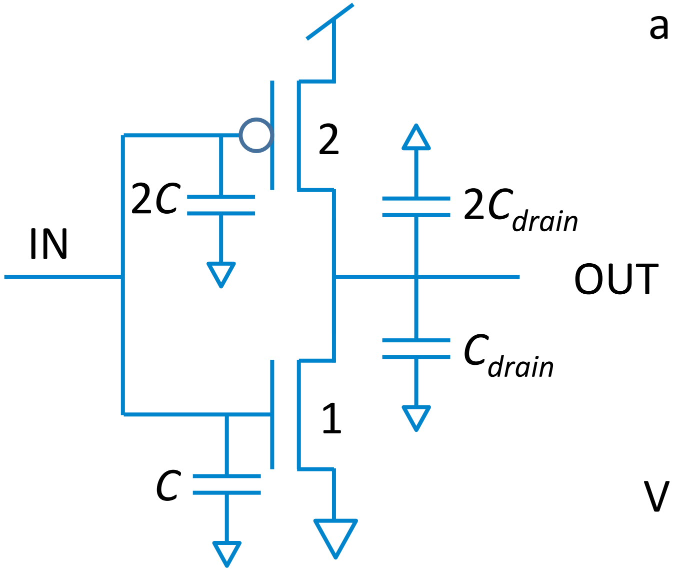
Delay estimations using logical effort

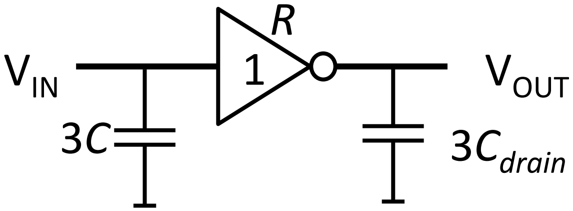
Once the logical effort of a logic gate is known we can easily calculate the propagation delay of any critical timing path

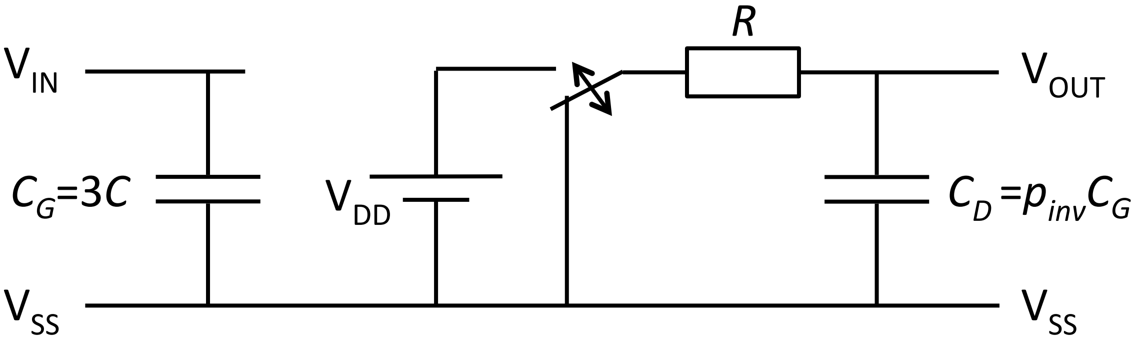


Inverter propagation delay model

Two-port electrical representation of propagation delay model
Consider a unit size inverter with a unit size n-channel device and a p-channel device of twice that width

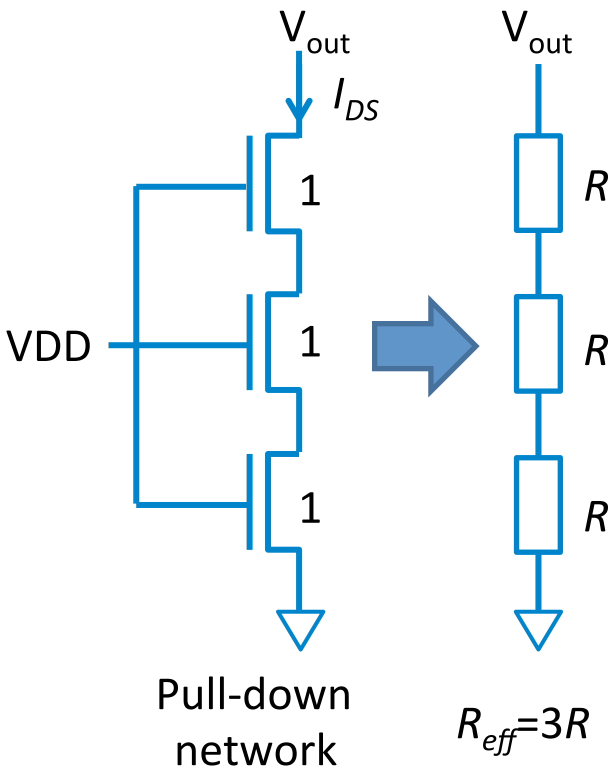




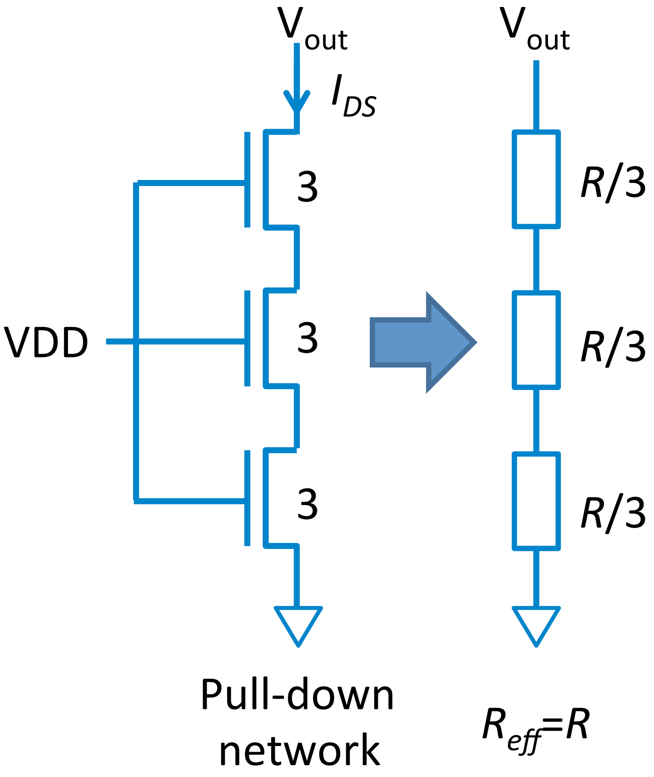


Effective resistance

- Pull-up/down paths may have MOSFETs in series! Effective resistances add up!

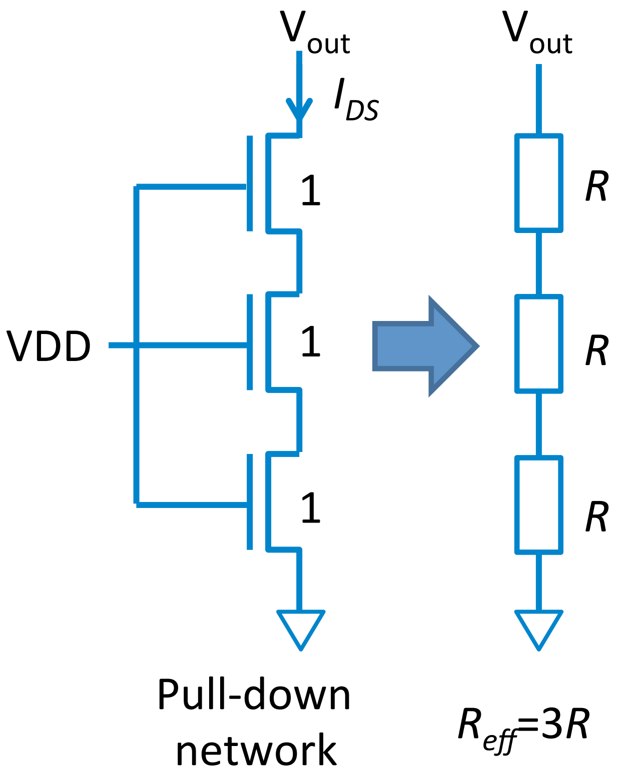


Solution: widen the MOSFETs to obtain effective resistance *R*! N MOSFETs in series – make them 3 units wide!

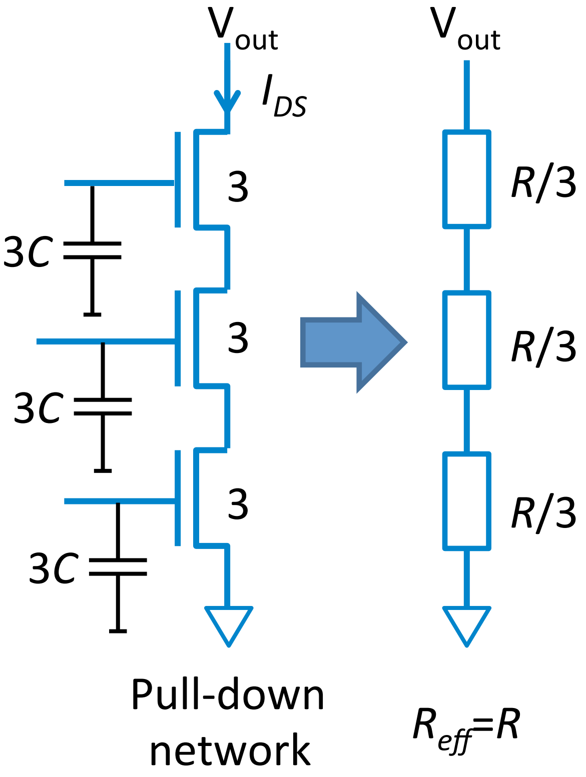


Effective resistance

- Pull-up/down paths may have MOSFETs in series! Effective resistances add up!

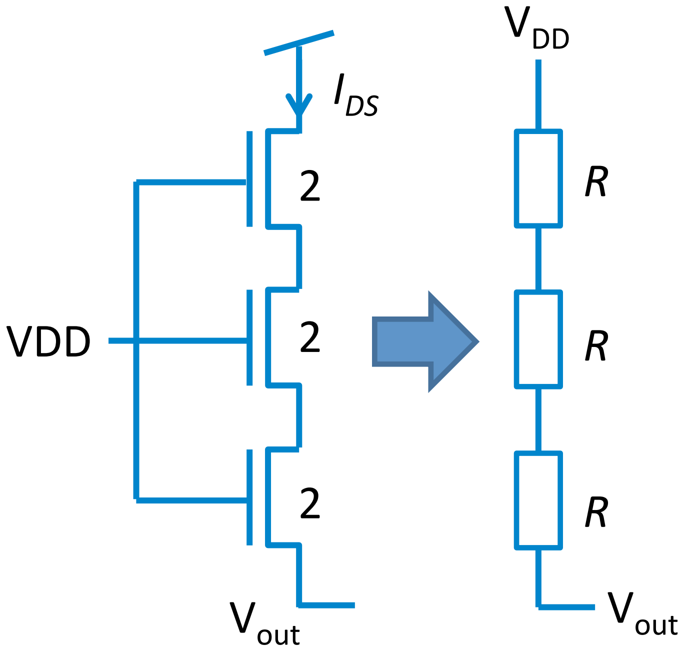


Cost: Increased input capacitance!



Effective resistance

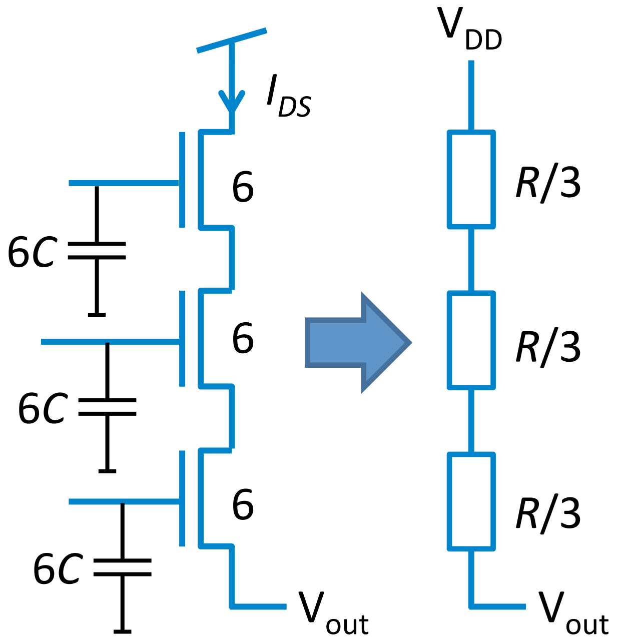
Pull-up/down paths may have MOSFETs in series! Effective resistances add up! This is for p-channel devices!



Pull-up network

$R_{eff}=3R$

Again, widen MOSFET a factor 3 since they are 3 in series. Cost: Increased input capacitance!



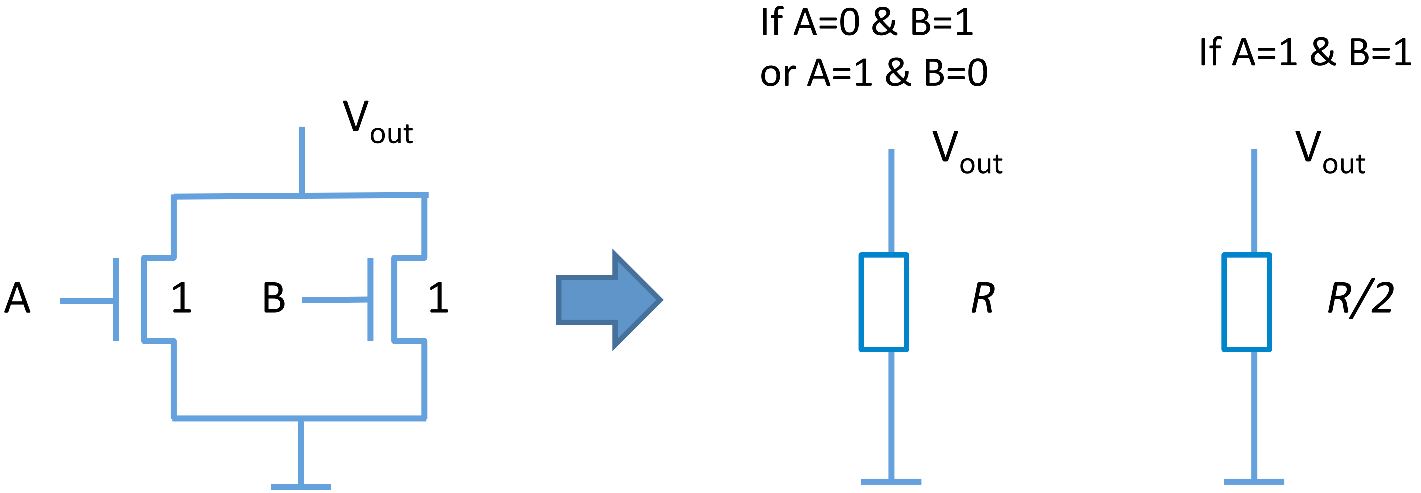
Pull-up network

$R_{eff}=R$

Effective resistance

The propagation delay, t_{pd} is defined as the **worst-case** delay, that is the longest delay. (We did not mention that in lecture 4 because an inverter only has one input.)

Pull-up/down paths may have MOSFETs parallel! Effective resistances can be combined as resistors in parallel. But what is the worst-case resistance?



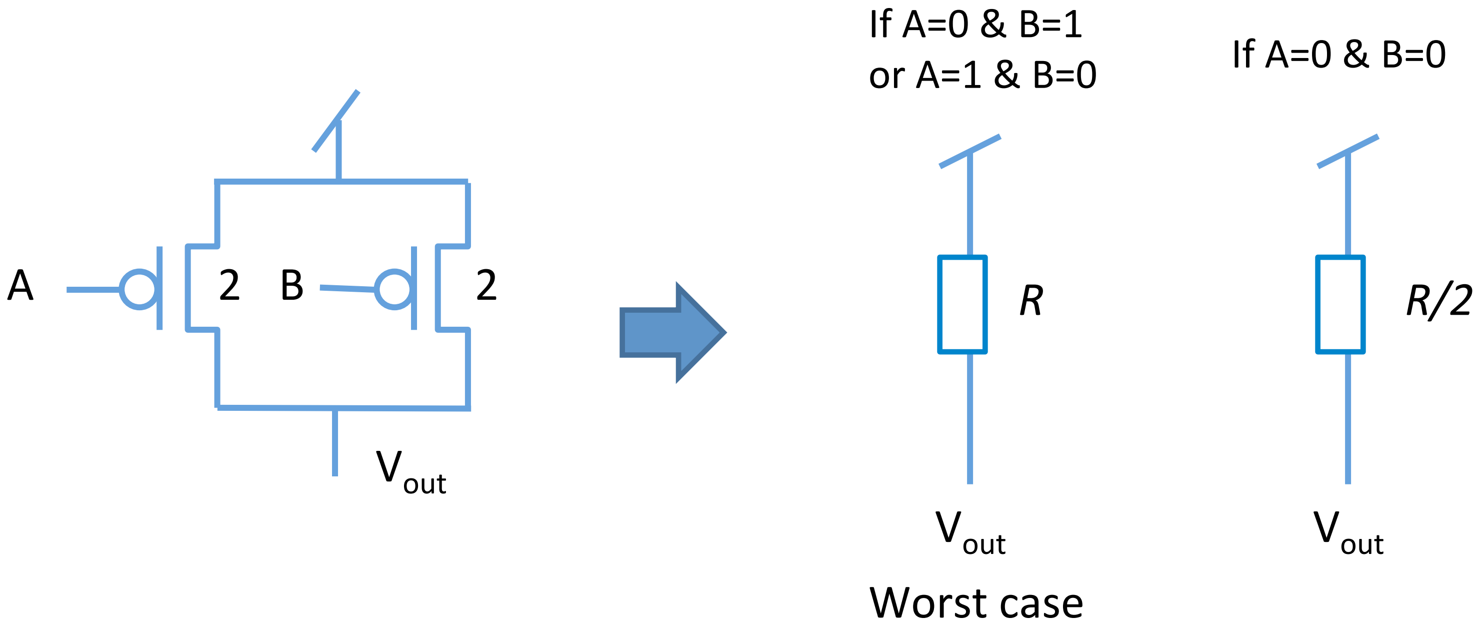
If A=0 & B=1
or A=1 & B=0

If A=1 & B=1

Worst case
because highest *R*

Effective resistance

The same but for pMOS FETs in parallel.



If A=0 & B=1
or A=1 & B=0

If A=0 & B=0

Worst case

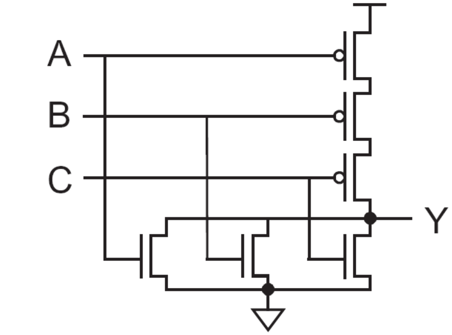
The NAND/NOR VTC revisited

Which VTC is NAND and which VTC is NOR?
Assume all transistors have the same k and all three inputs switch together.

Now we can use our knowledge about R_{eff} to find the effective k_n and k_p. R_{eff} is inversely related to the maximum saturation current. So for entire path combined R gives us the inverse of the k factors.

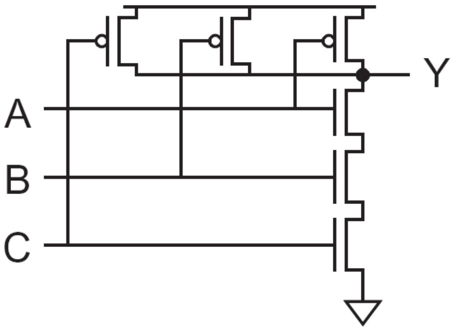
$$V_{sw} = V_{TN} + \frac{\Delta V}{1+\sqrt{x}}$$

where x = k_n/k_p



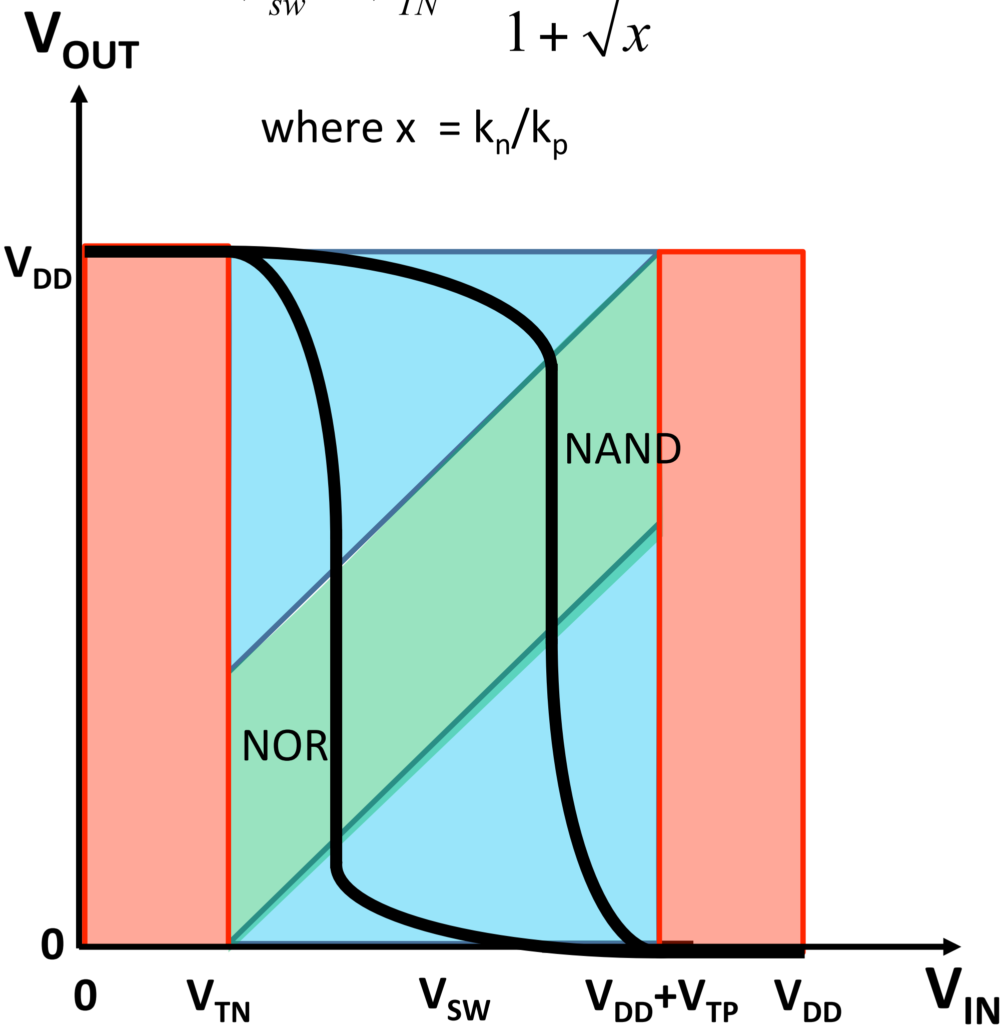
3R => k_p = 1/3

R/3 => k_n = 3



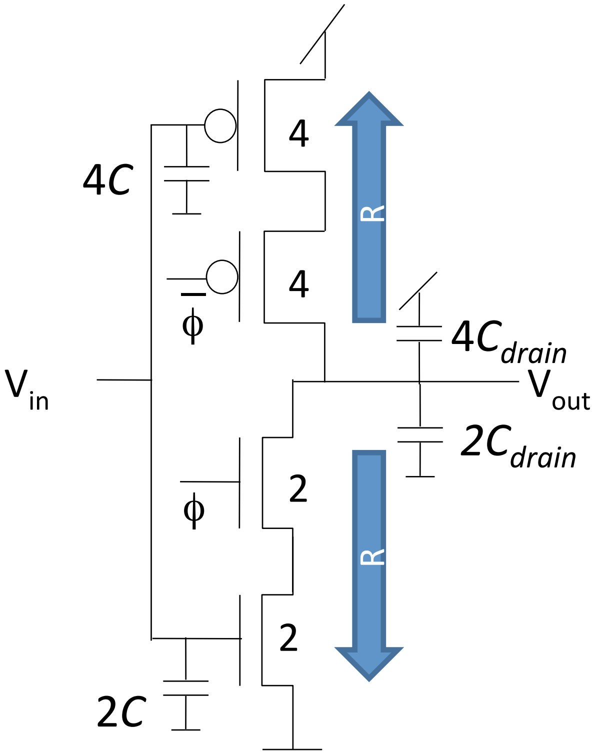
R/3 => k_p = 3

3R => k_n = 1/3



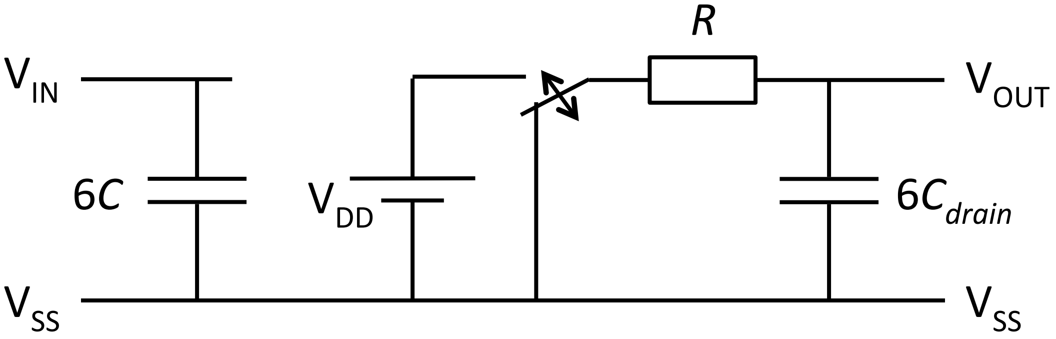
Example 1: The tri-state inverter

Make all MOSFETs twice as wide to cut effective resistance in half!

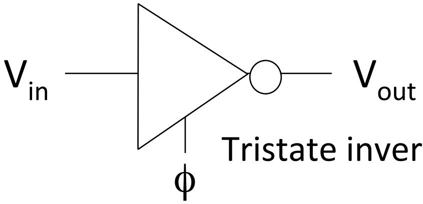


Input cap = twice that of reference inverter: C_G=6*C*
where *C* is gate capacitance of a unit width MOSFET

Parasitic output cap = 6C_{drain} which is twice that of the unit inverter



While inverter *RC*-product is 3*RC*, tristate inverter *RC*-product is 6*RC*, i.e. twice that of the inverter!

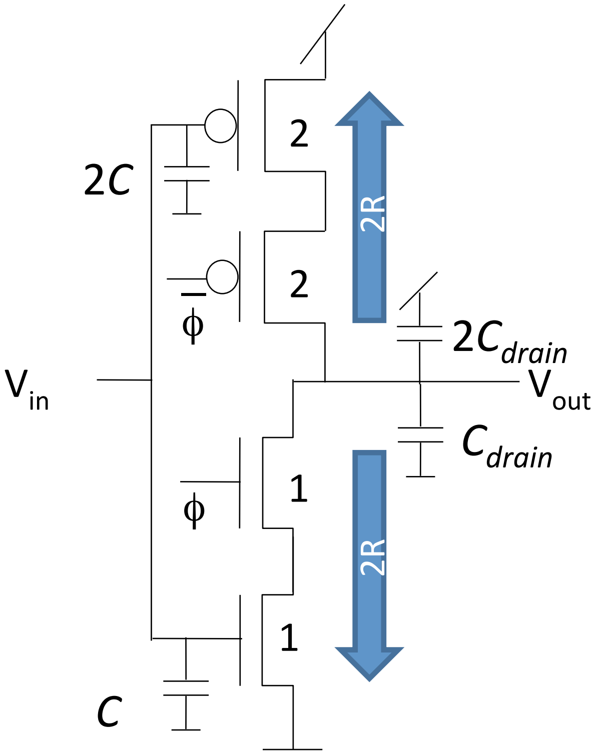


Tristate inverter symbol

This ratio is defined as the "logical effort", *g*, and *g*=6*C*/3*C*=2!
Parasitic delay =6C_{drain}/3*C*=2p_{inv}

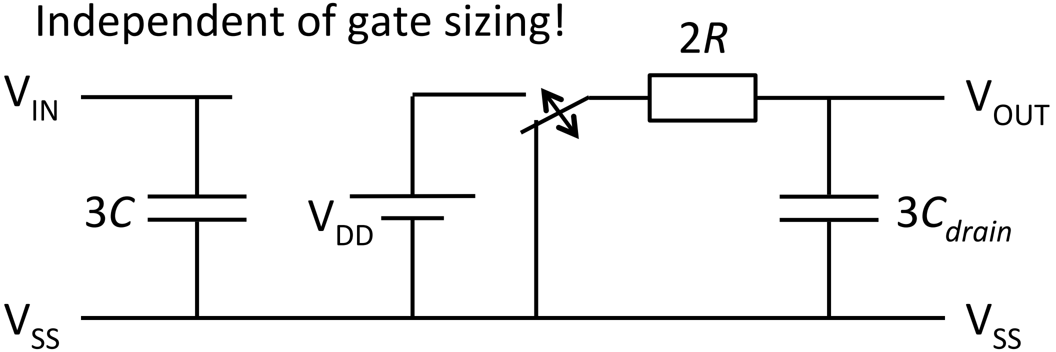
Example 1: The tri-state inverter

Make all MOSFETs at original widths to keep input capacitances @ 3*C*.



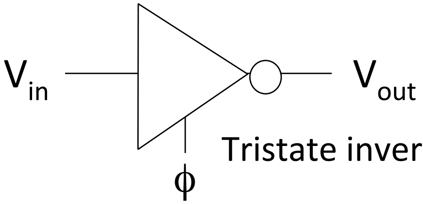
Say that we want input capacitance to be same as for the inverter! Then path resistances become 2R!

But parasitic delay and logical effort are the same! Independent of gate sizing!



While inverter *RC*-product is 3*RC*, tristate inverter *RC*-product is 6*RC*, i.e. twice that of the inverter!

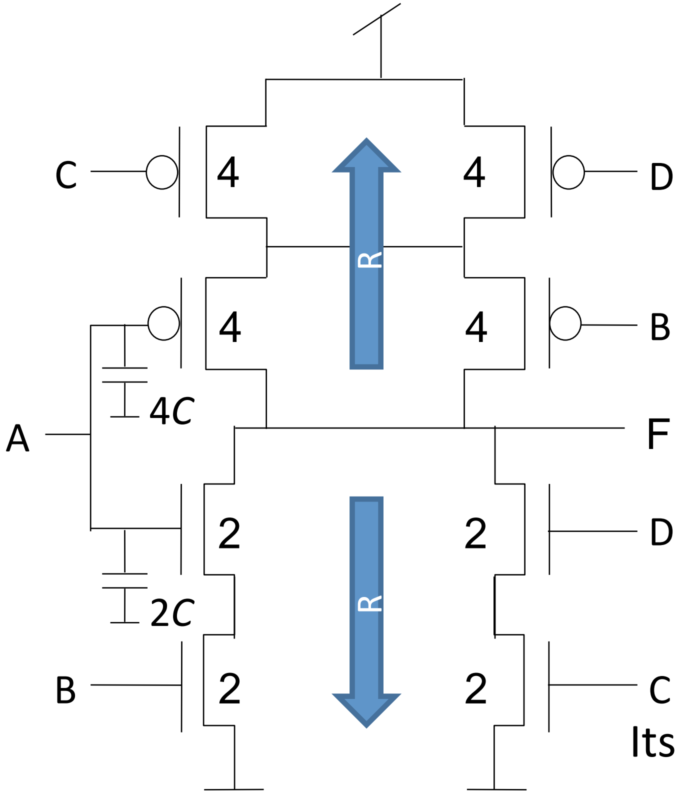
Because $R \sim \frac{1}{W}$ and $C \sim W$. Hence RC is constant!



Tristate inverter symbol

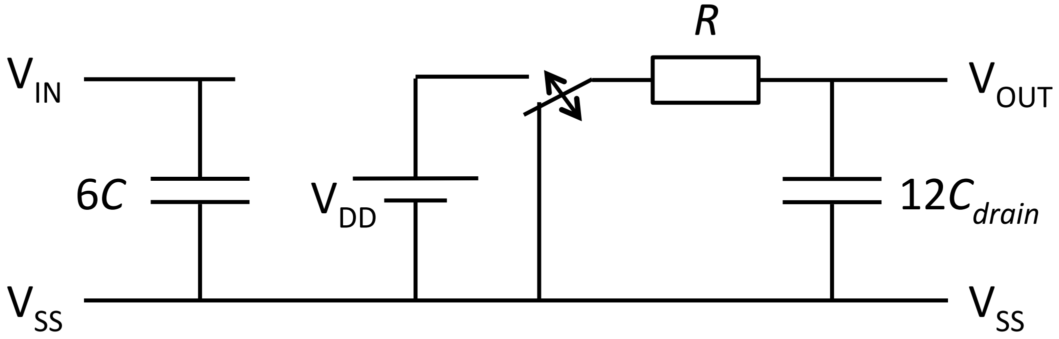
Example 2: 2+2 AND-OR-invert (AOI22)

Make all MOSFETs twice as wide to cut effective resistance in half!



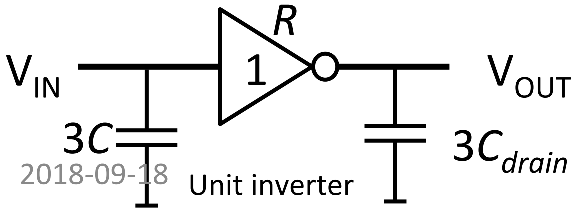
This gate also has logical effort, *g*=2 for all inputs A-D

$$g = \frac{\left(R_{eff} C_{IN}\right)}{3RC} = \frac{6RC}{3RC} = 2$$

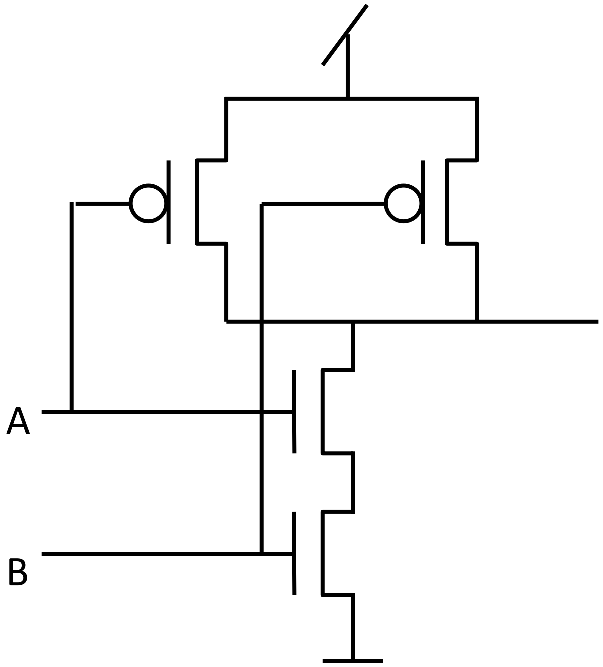


Its parasitic *RC*-product *p, the parasitic delay,* is defined as

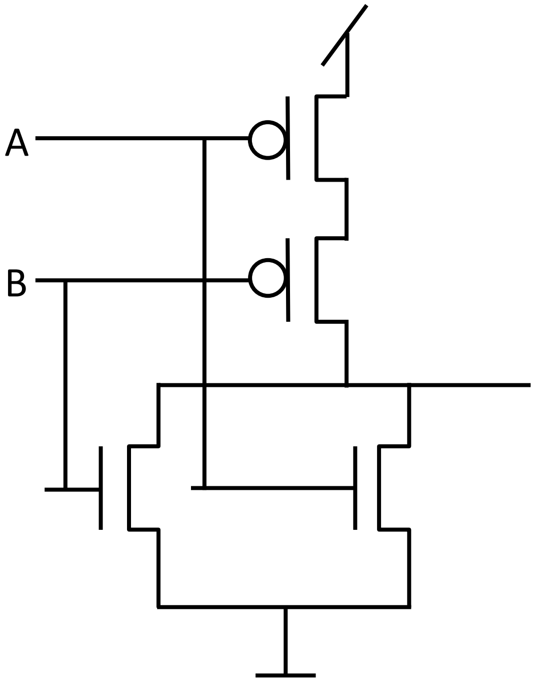
$$p = \frac{\left(R_{eff} C_{par}\right)}{3RC} = \frac{12RC_{drain}}{3RC} = 4\frac{C_{drain}}{C} = \begin{cases} 4 \text{ if } p_{inv} = 1.0 \\ 3.2 \text{ if } p_{inv} = 0.8 \end{cases}$$



Exercise:
Calculate NAND2, NOR2 logical efforts

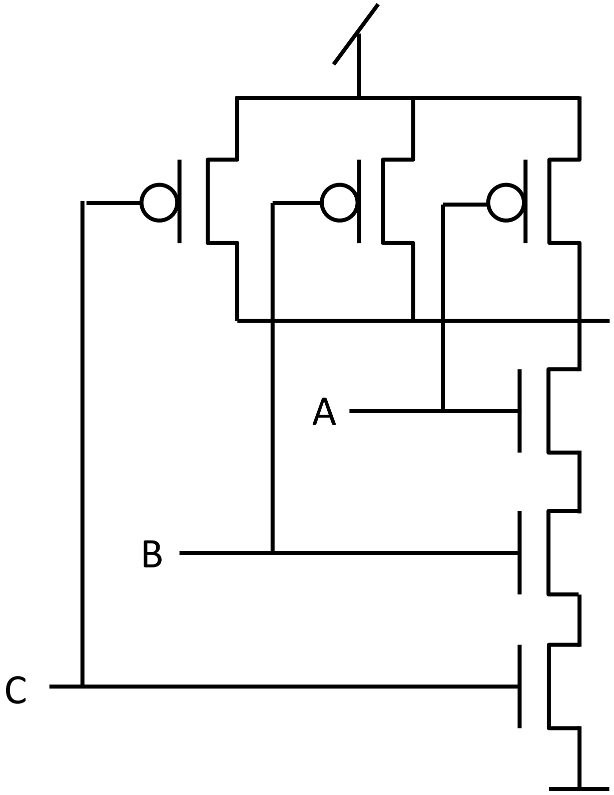


NAND2

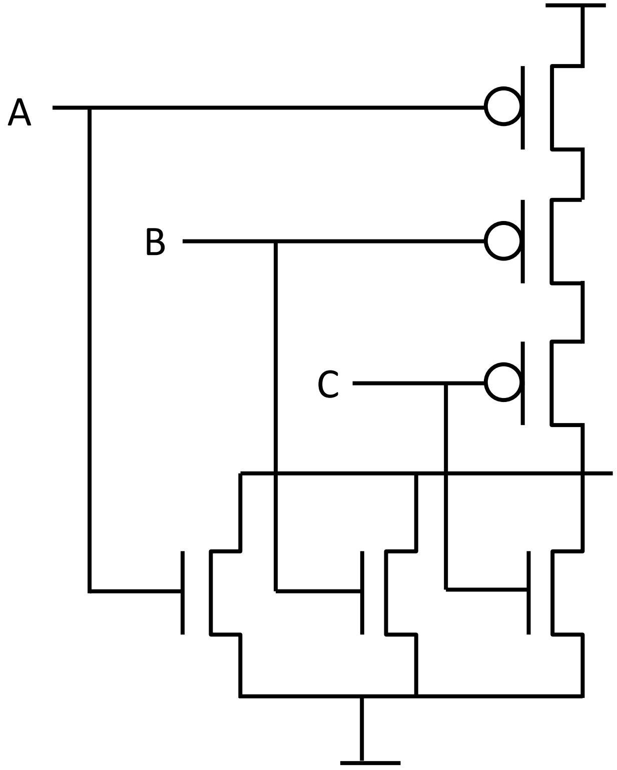


NOR2

Exercise:
Calculate NAND3 & NOR3 logical efforts



NAND3



NOR2

Exercise: compound gate OR-AND-INVERT 3+1

Note: Different inputs to a gate can have different logical efforts!

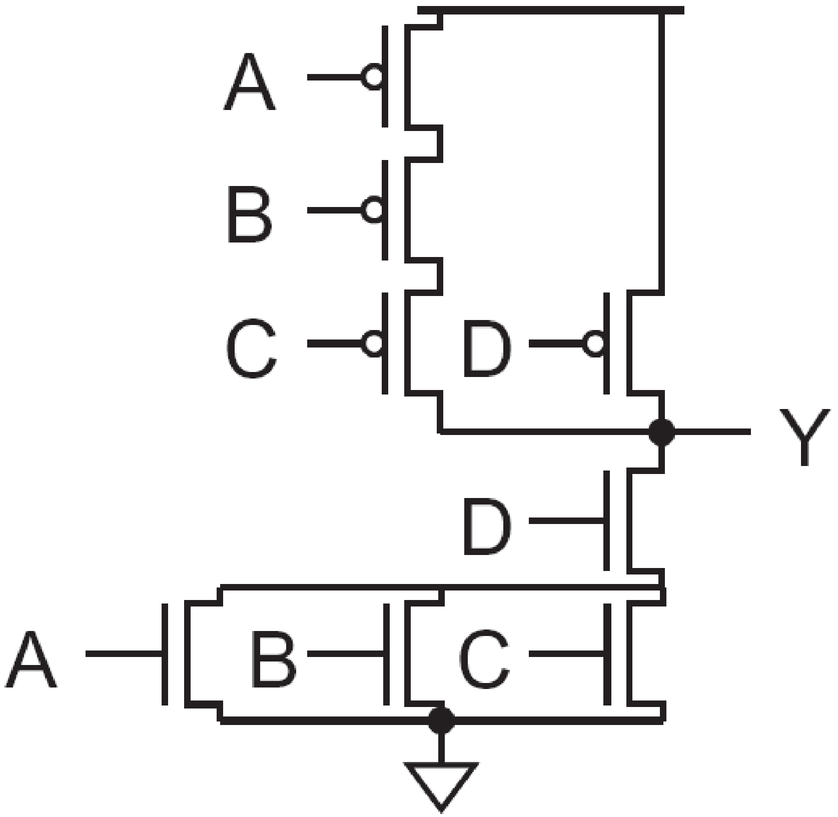


FIGURE 1.19
CMOS compound gate for function
$Y = \overline{(A + B + C) \cdot D}$

OAI invert 3+1 gate done as quiz

- Question 1: What is the logical effort of ABC inputs?
- Question 2: What is the logical effort of D input?
- Question 3: What is the parasitic delay expressed as p x pinv?

- Work in small groups

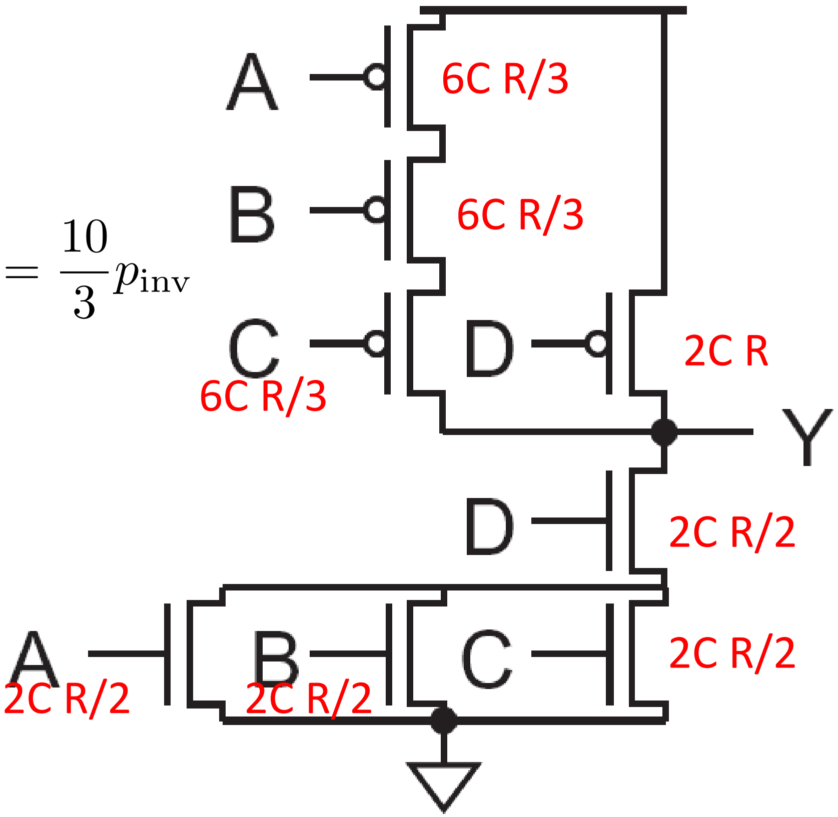
Exercise: compound gate OR-AND-INVERT 3+1 (solution)

Gate capacitance is proportional to transistor width. Here we write out capacitance for the scaling rather than width. Drain capacitance is also proportional to width.

$$p = \frac{6C_{\text{drain}} + 2C_{\text{drain}} + 2C_{\text{drain}}}{3C} = \frac{10}{3}\frac{C_{\text{drain}}}{C} = \frac{10}{3}p_{\text{inv}}$$

$$g_{A,B,C} = \frac{2C + 6C}{3C} = \frac{8}{3}$$

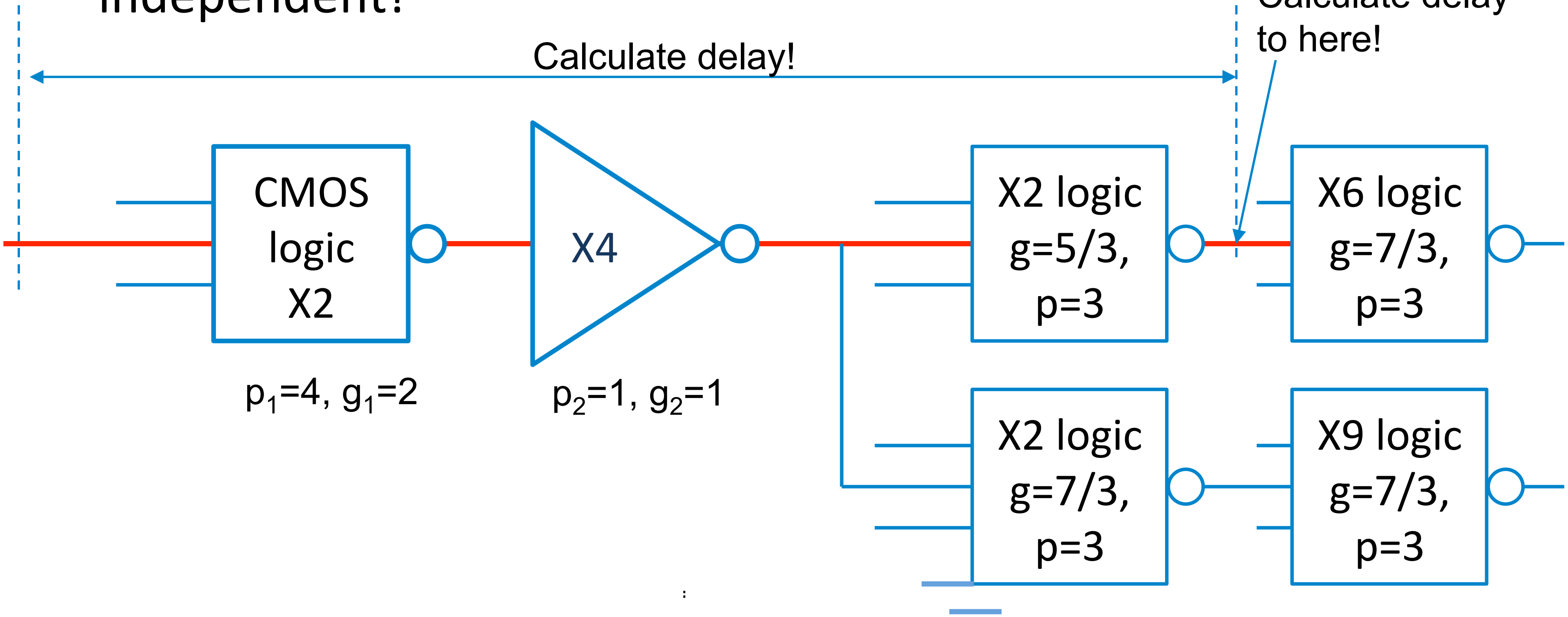
$$g_D = \frac{2C + 2C}{3C} = \frac{4}{3}$$



Why all this?

In ST cell library, size X refers to the input cap!

- Simplify delay calculations and making them technology independent!



$p_1=4,\ g_1=2$

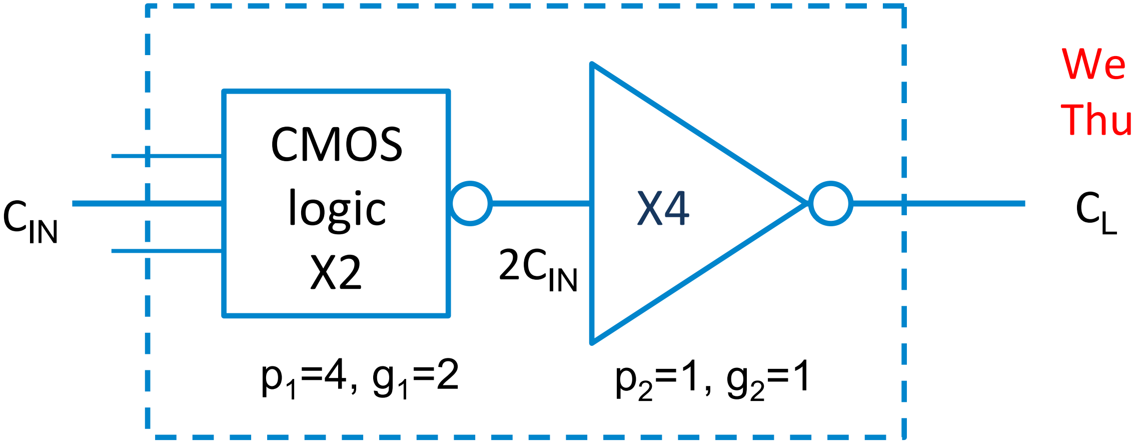
$p_2=1,\ g_2=1$

Why all this?

- Minimize path delays with gates other than inverters.
- We will do that in lecture 6.

Model for a non-inverting gate

d = f + p should hold also for non-inverting gate
f is the part that depends on C_L, f = gh where h is defined as C_L/C_{IN}
Thus $f_{gate} = C_{IN}/C_{INinv} \times C_L/C_{IN} = g_{gate} \times h_{gate}$
p is static part of the delay: the part that does not depend on C_L



We will do this on Thursday!

$$d = \underbrace{4 + 2\frac{X4}{X2}}_{\text{1st stage}} + \underbrace{1 + \frac{C_L}{2C_{IN}}}_{\text{inverter}} = \underbrace{9}_{p} + \underbrace{\frac{C_{IN}}{2C_{IN}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = \underbrace{9}_{p} + \underbrace{\frac{1}{2}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h}$$

The parasitic delay *p*=9
Logical effort *g*=0.5

Overview important concepts

$$d = f + p$$

$$f = gh$$

d: (normalized) stage delay

f: stage effort

g: logical effort

h: electrical effort

p: parasitic delay

All concepts related
to **reference inverter**
with $(RC_G)_{inv}$, $p_{inv} = C_{Dinv}/C_{Ginv}$

The gate itself has parameters
C_{IN}, R_{eff} and C_{par}
and is connected to load C_L

Note:
C_{IN} can be different for different gate inputs =>
Different *g* for different inputs!

If $R_{eff} \neq R_{inv}$

$$f = \underbrace{\frac{R_{eff}C_{IN}}{(RC_G)_{inv}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

$$p = \frac{R_{eff}C_{par}}{(RC_D)_{inv}}\underbrace{\frac{C_{Dinv}}{C_G}}_{p_{inv}}$$

If $R_{eff} = R_{inv}$

$$f = \underbrace{\frac{C_{IN}}{C_{Ginv}}}_{g}\underbrace{\frac{C_L}{C_{IN}}}_{h} = gh$$

$$p = \frac{C_{par}}{C_{Dinv}}\underbrace{\frac{C_{Dinv}}{C_{Ginv}}}_{p_{inv}}$$

Conclusion

In this lecture we have

- Adapted the two-port delay model previously developed for inverters to any CMOS logic gate
- Learnt how to size MOSFETs for equal worst case rise and fall delay
- Noticed that logic gates have larger $R_{eff}C_G$ products than inverters
- Introduced two new concepts: ***logical effort*** and ***parasitic delay***
- Logical effort and parasitic delay quantifies
  - Logical effort: $(R_{eff}C_G)_{gate}$ relative to $(3RC)_{inv}$
  - Parasitic delay: $(R_{eff}C_{par})_{gate}$ relative to $(3RC)_{inv}$
- Learnt how to calculate these parameters for any logic gate
- Learnt how to calculate the electrical effort $h= C_{LOAD}/C_{IN}$
- Learnt to calculate all propagation delays as a multiple of tau, where tau=5 ps in STMicroelectronics 65 nm CMOS process