Fundamentals of CMOS Design II

Ronald Valenzuela
ronald.valenzuela@synopsys.com
Corporate Application Engineer - Synopsys

Outline

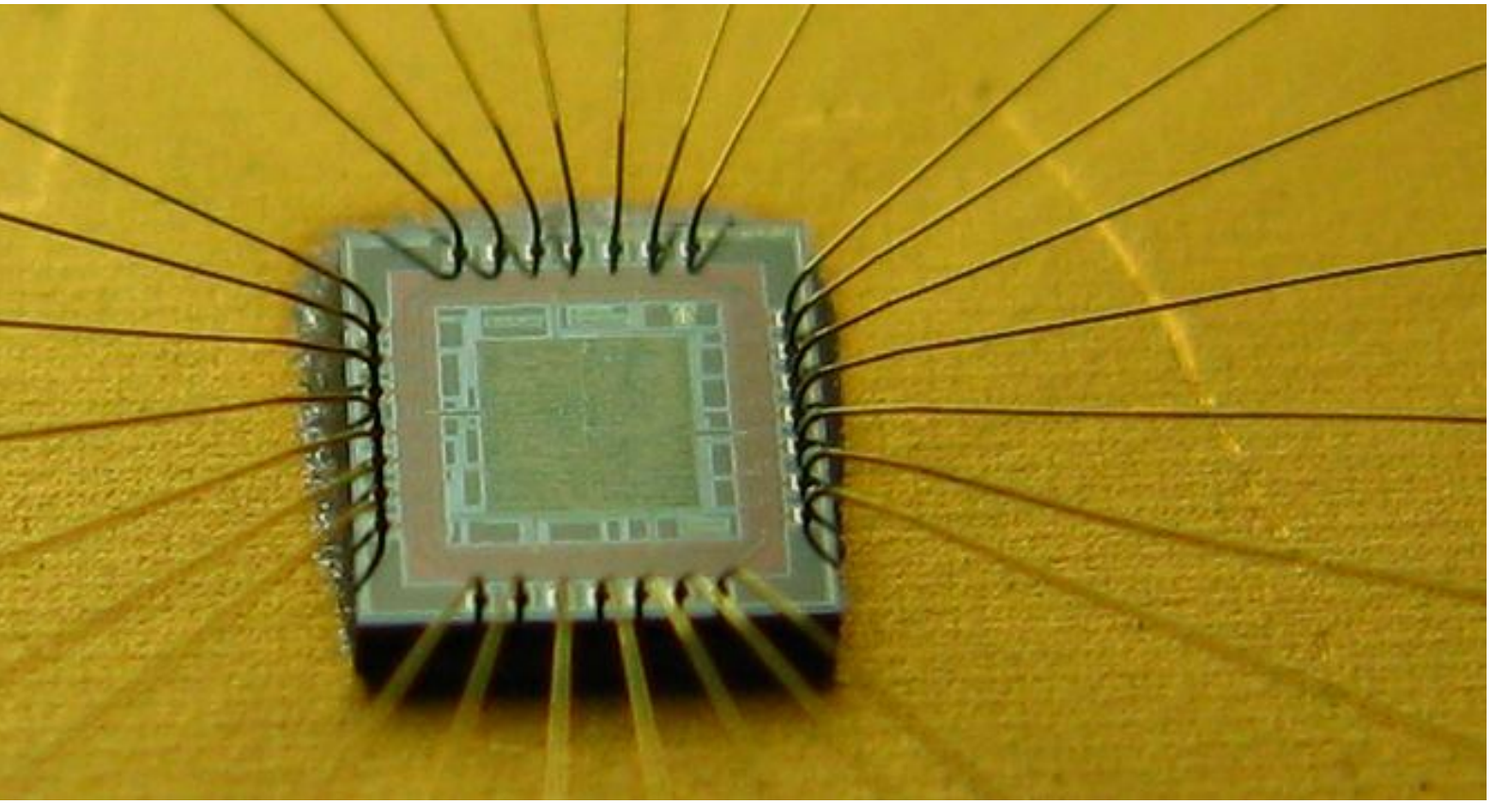
Gate Performance

Gate Power

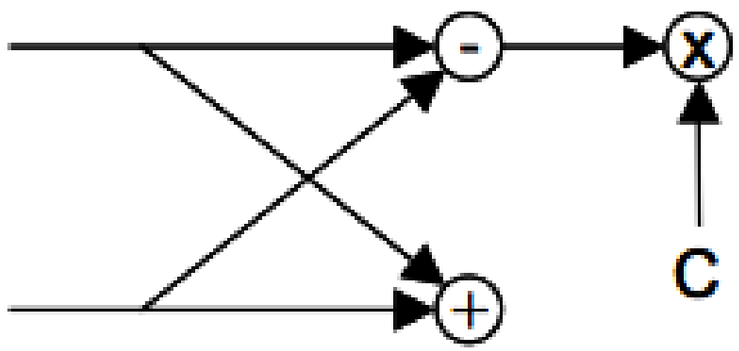
System Analysis

Gate Performance

We want to build Chips

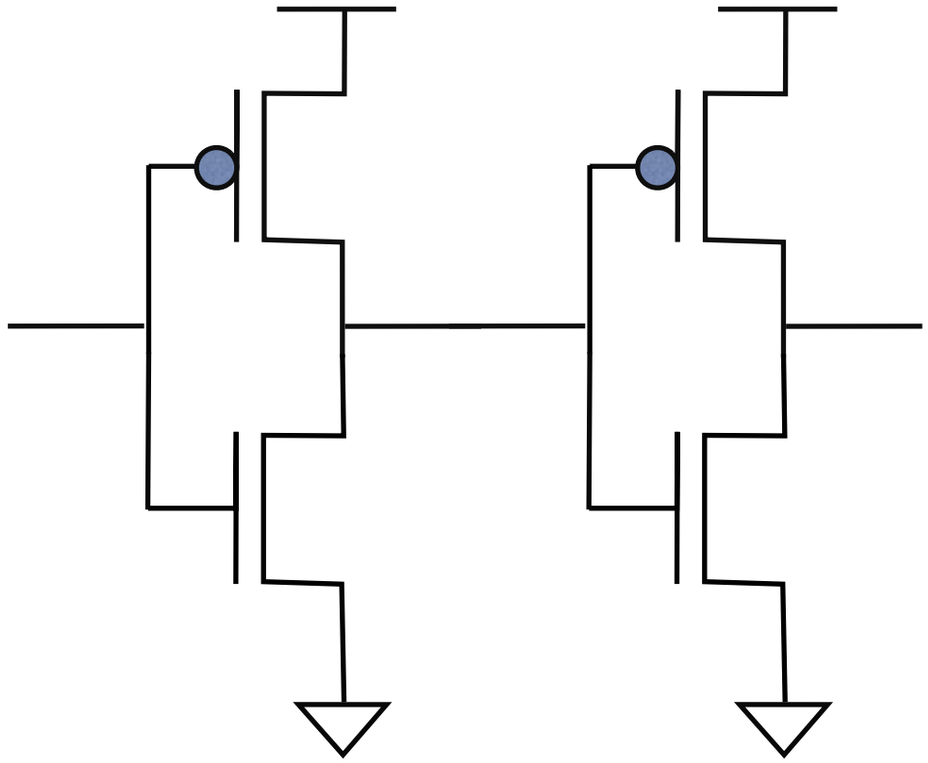


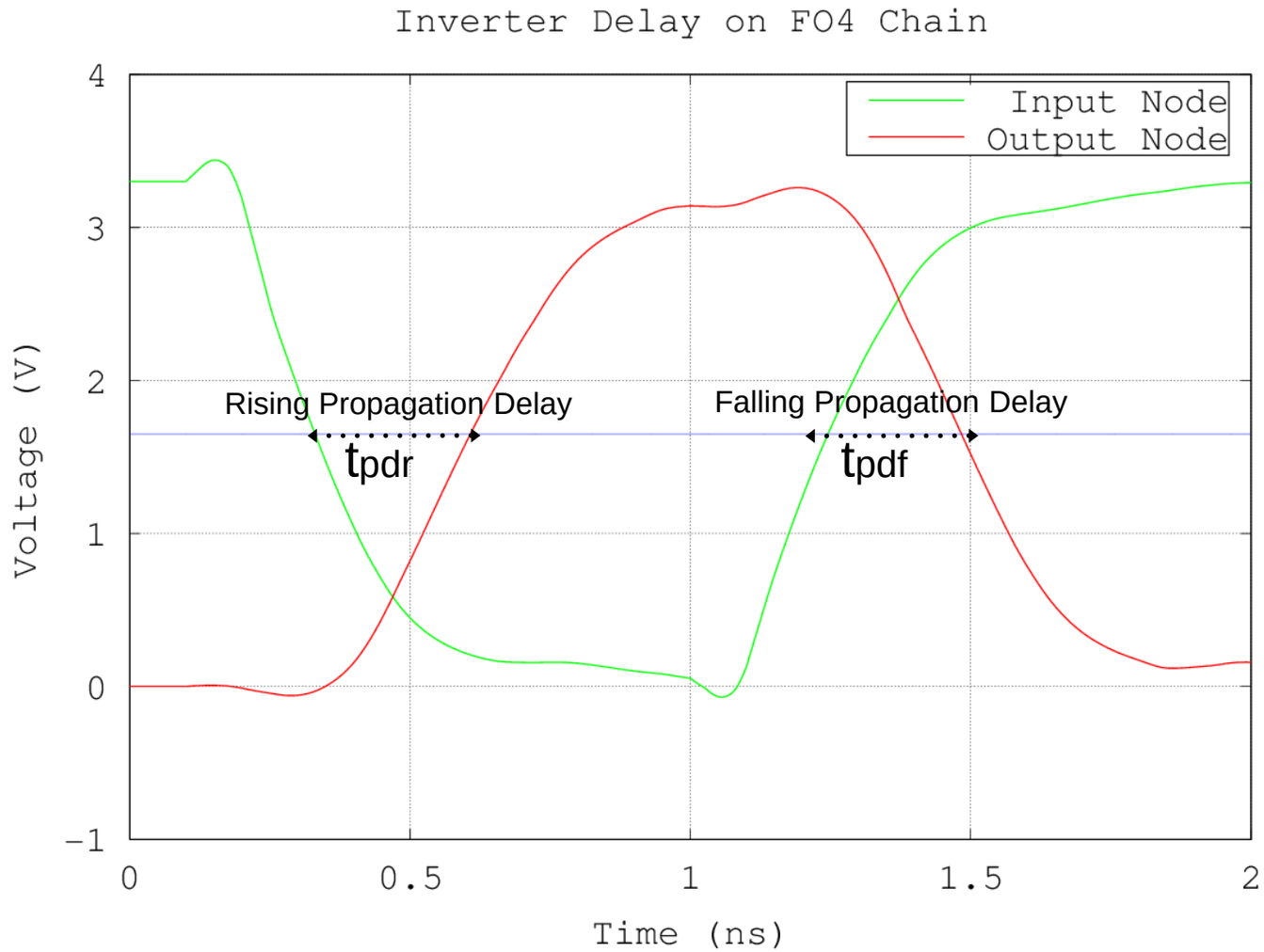
0011100110011



To perform complex binary calculations

But what we are really are building CIRCUITS

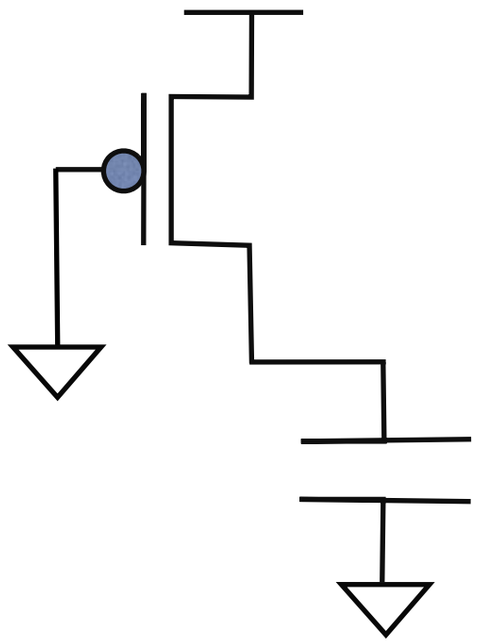


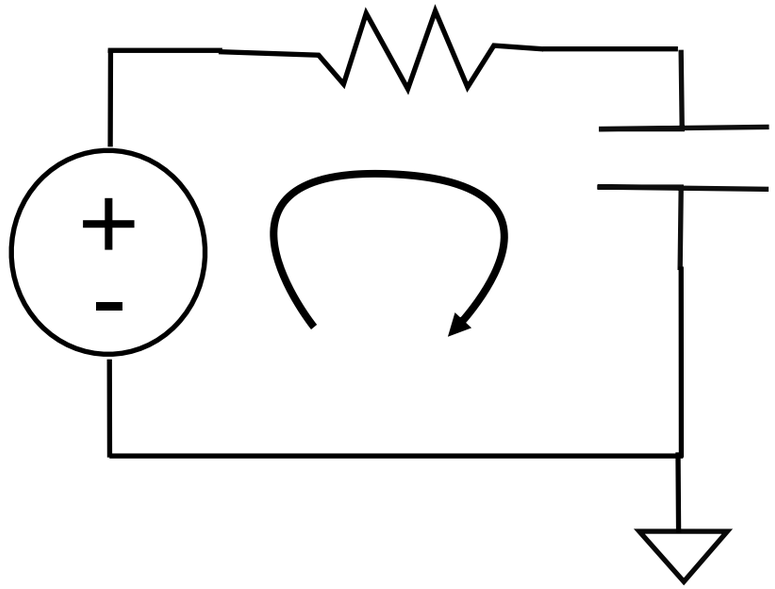


And circuits have transient response

Pull Up Delay

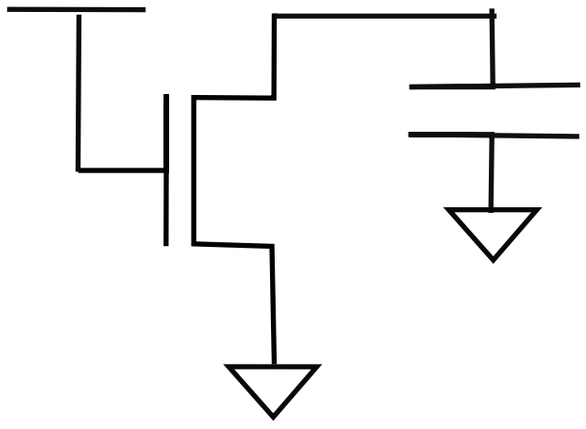
- Rise time calculation

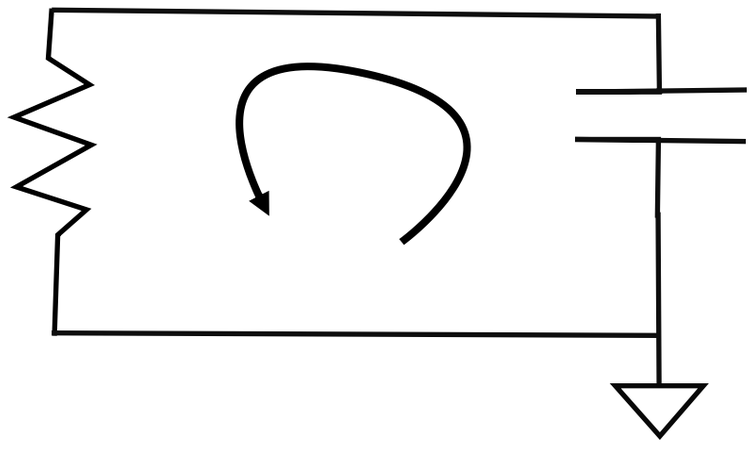




Pull Down Delay

- Fall time calculation

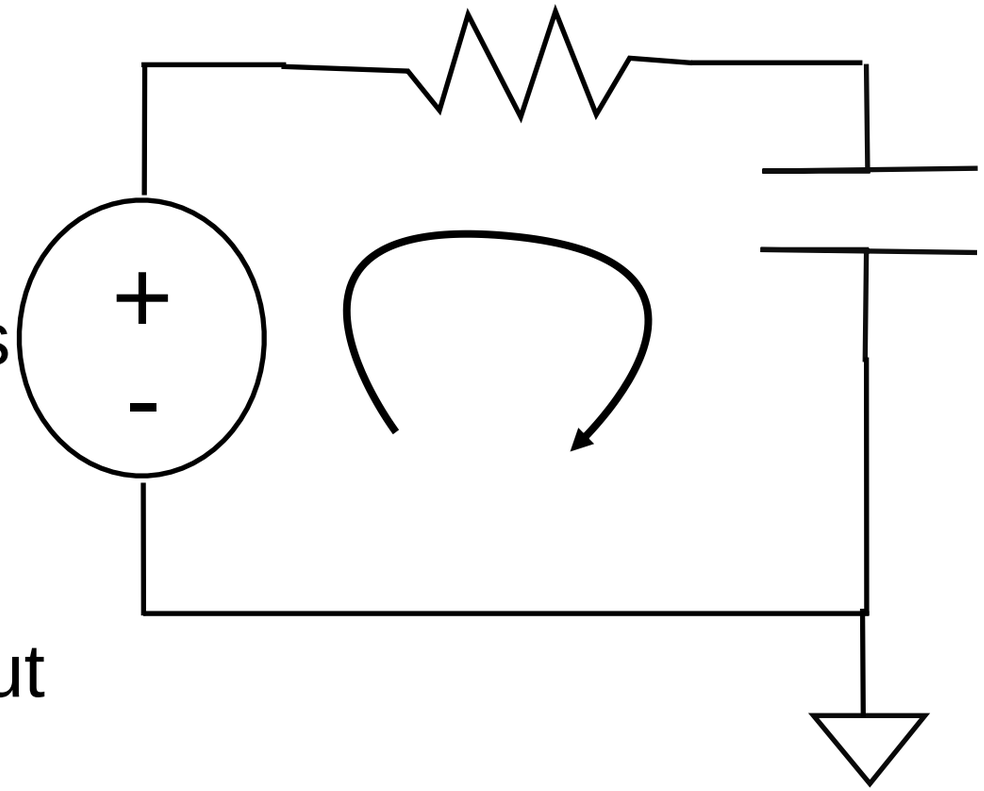




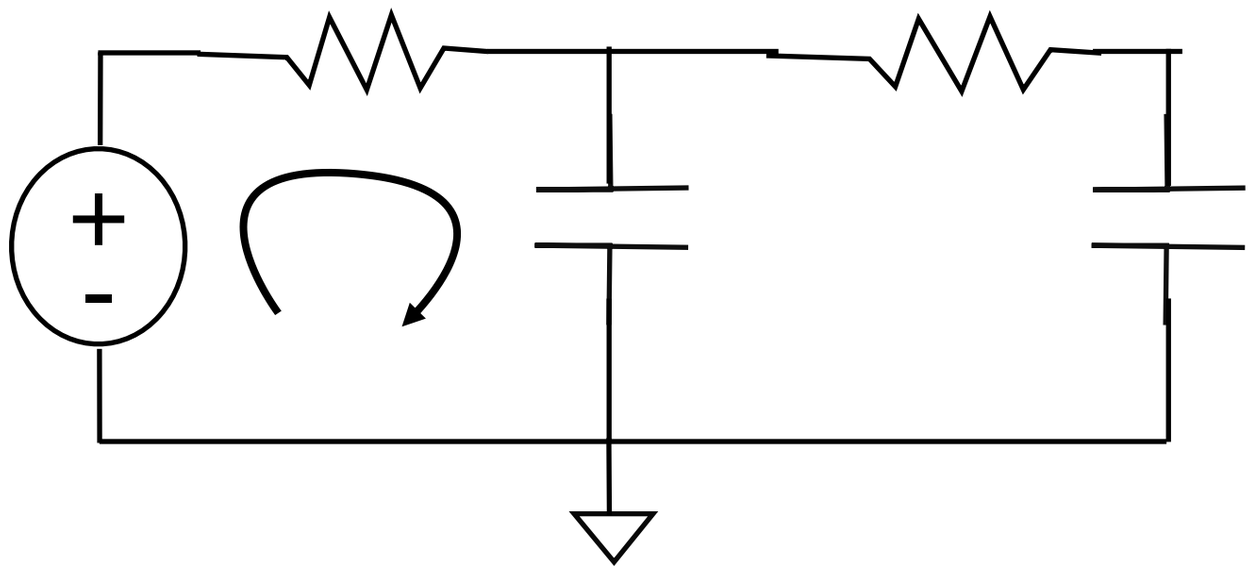
Delay - And then...

$$H(s) = \frac{1}{1 + sRC} \qquad \Rightarrow \qquad V_{out} = V_{dd}(1 - e^{\frac{-t}{R_P C}})$$

- Propagation delay is the time when when Vout reaches ½ Vdd
- $t_{pd} = RC * ln2 = 0.69 * RC$
- So simple that we will take this value as the delay for hand calculations.
- Define gate delay as:
  Time from input crossing Vdd/2 to output crossing Vdd/2 and approximate to RC



Unfortunately, we don't always have a 1st order system

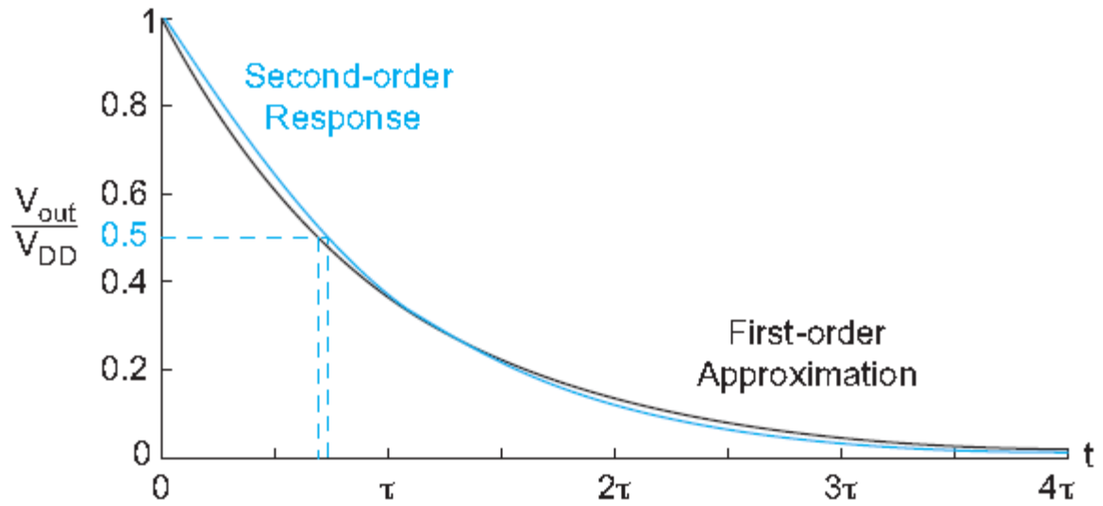


$$H(s) = \frac{1}{1 + s(R_1C_1 + (R_1 + R_2)C_2) + s^2R_1C_1R_2C_2}$$

This is too complex. It defeats the purpose of simplifying a CMOS circuit into an equivalent RC network

Need something simpler!

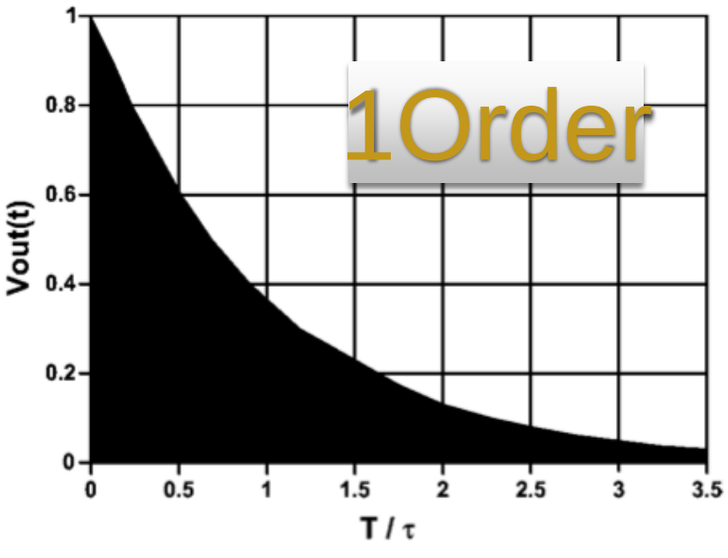
Sometimes we can approximate the response considering just the dominant pole

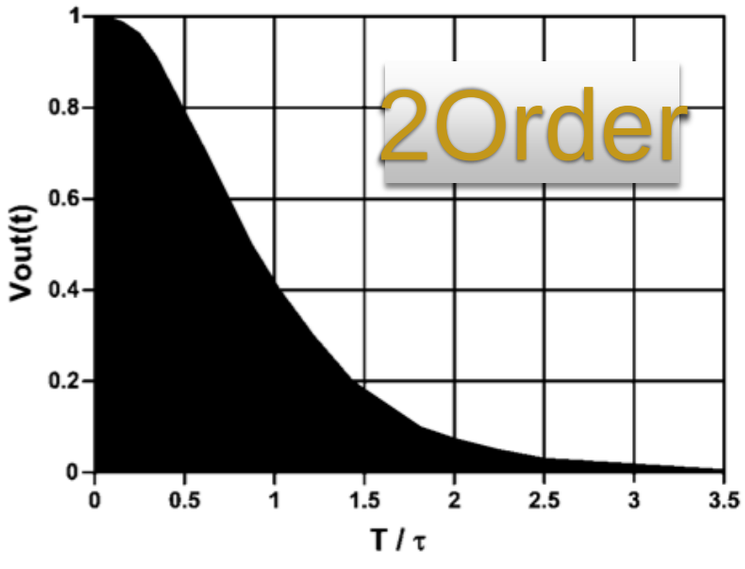


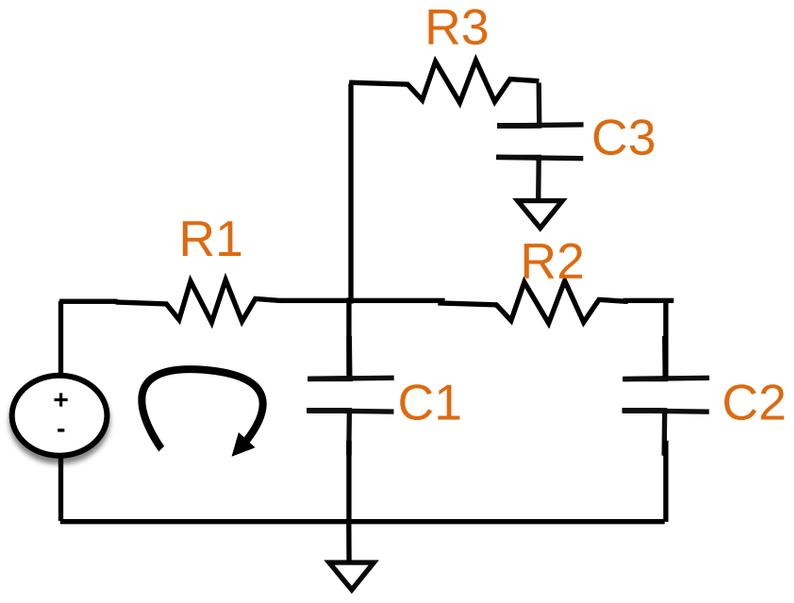
- This works very well when the time constant of the dominant pole is much bigger than the other (error < 7%)
- Worst case (when time constants are equal) has error < 15%
- The problem here is that this approximation doesn’t describe well intermediate nodes

Elmore Delay

- This approximation is based on finding an equivalent 1st – order reponse for the actual response of our circuit
- To find an equivalent τ we match the area under the step response curve of our circuit to the area of a 1st-order response





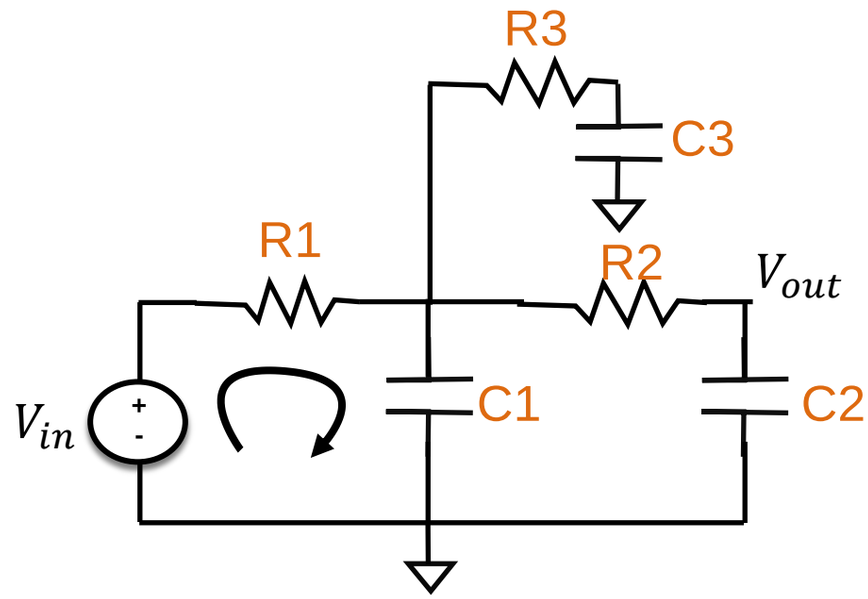


$$\int_0^\infty V_0' dt = \int_0^\infty V_0 dt = \sum RI = \sum RC\Delta V$$

Elmore Delay

The Elmore time constant τ_E is the sum over each node in the ladder of the resistance R_{n-1} between that node and the source V_{in}, multiplied by the capacitance on the node C_i

$$t_{pd} = \tau_E * ln2$$



In this example, the τ_D at V_{out} is calculated as

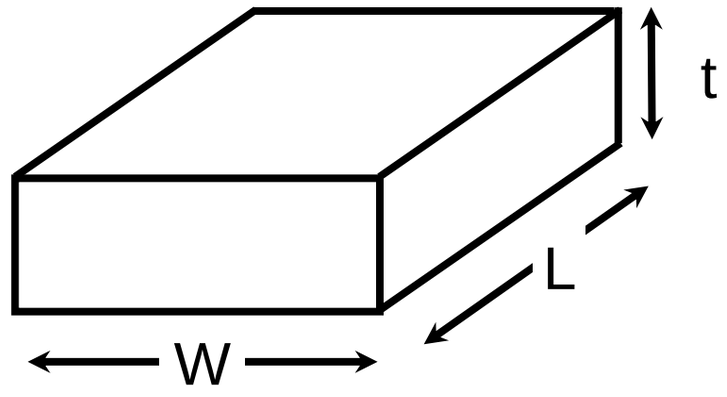
$$\tau_D = R_1C_1 + (R_1 + R_2)C_2 + R_1C_3$$

Note that $R_3$is dropped because it is not in the signal path (C_3 gets connected directly to the node)

Delay - Can we make this assumption?

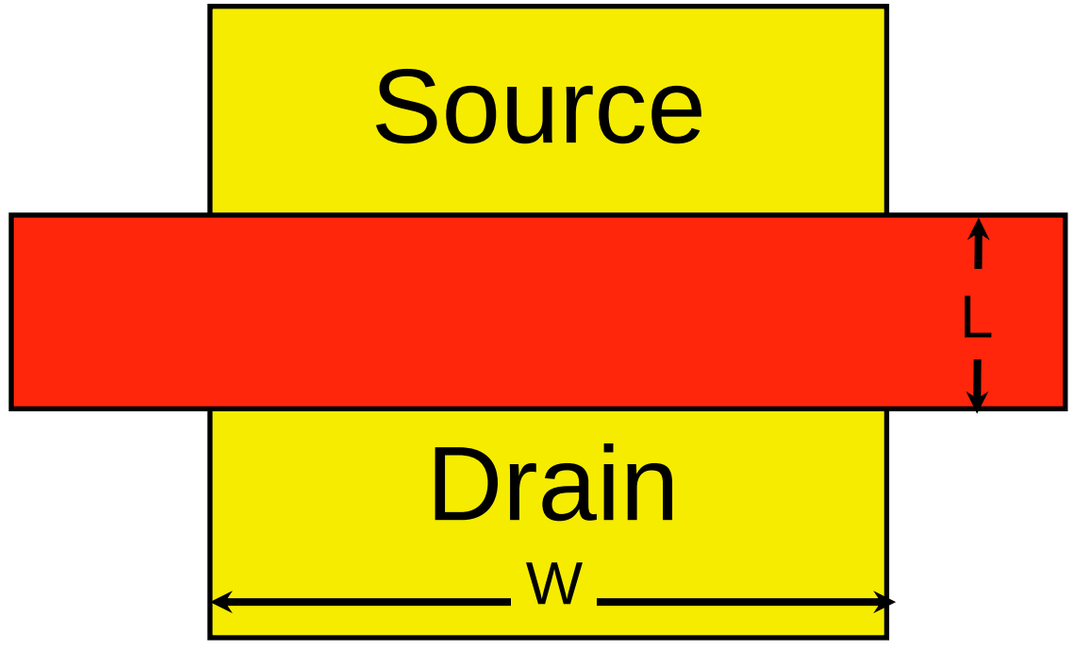
- Conductor Resistance:

$$R = \frac{\rho L}{tW}$$



- Transistor resistance
  - Designer can set W and L
  - Actually I_{ds}^{-1}
  - Prop^{-1} to Vdd.
  - Approximated by:

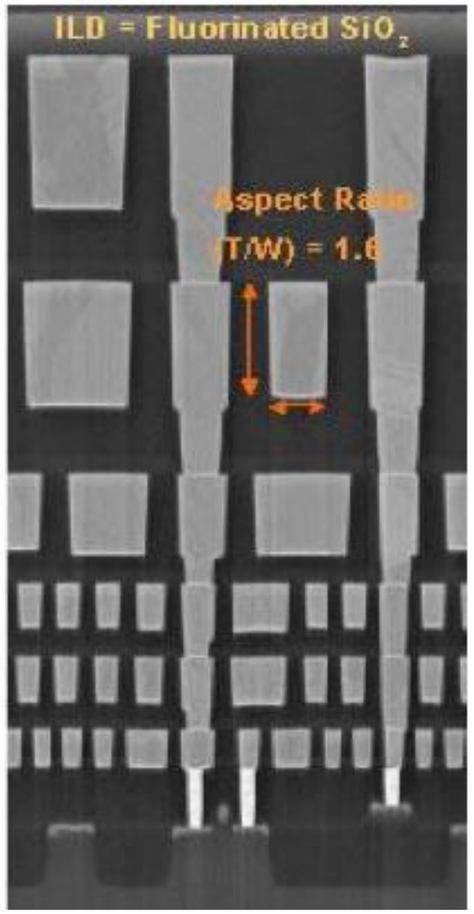
$$R \approx \frac{L}{W} K_{technology}$$

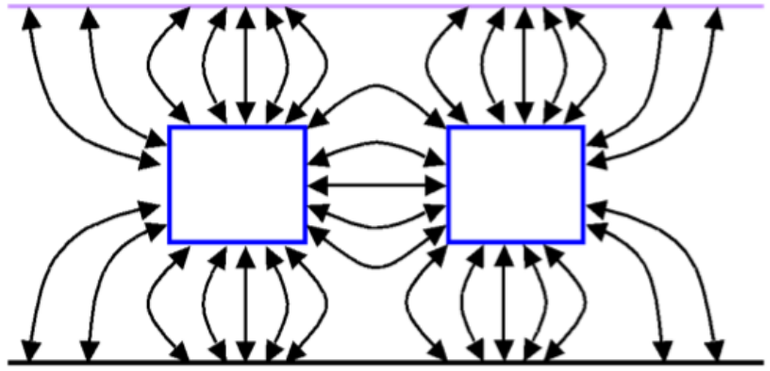


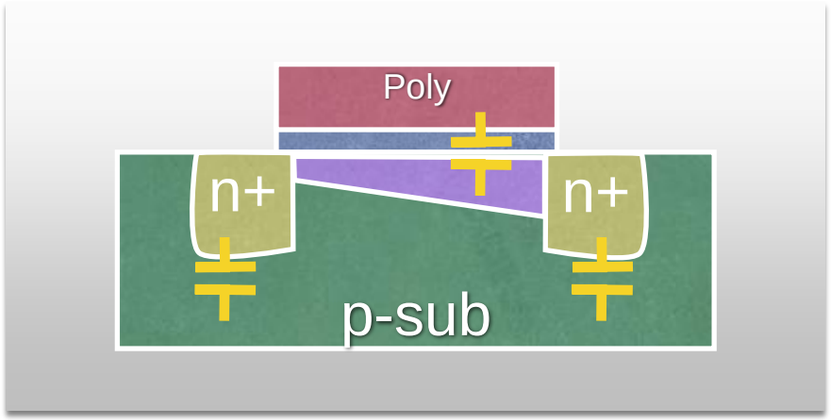
Delay - Capacitance

C_{load} comes mainly from three factors:

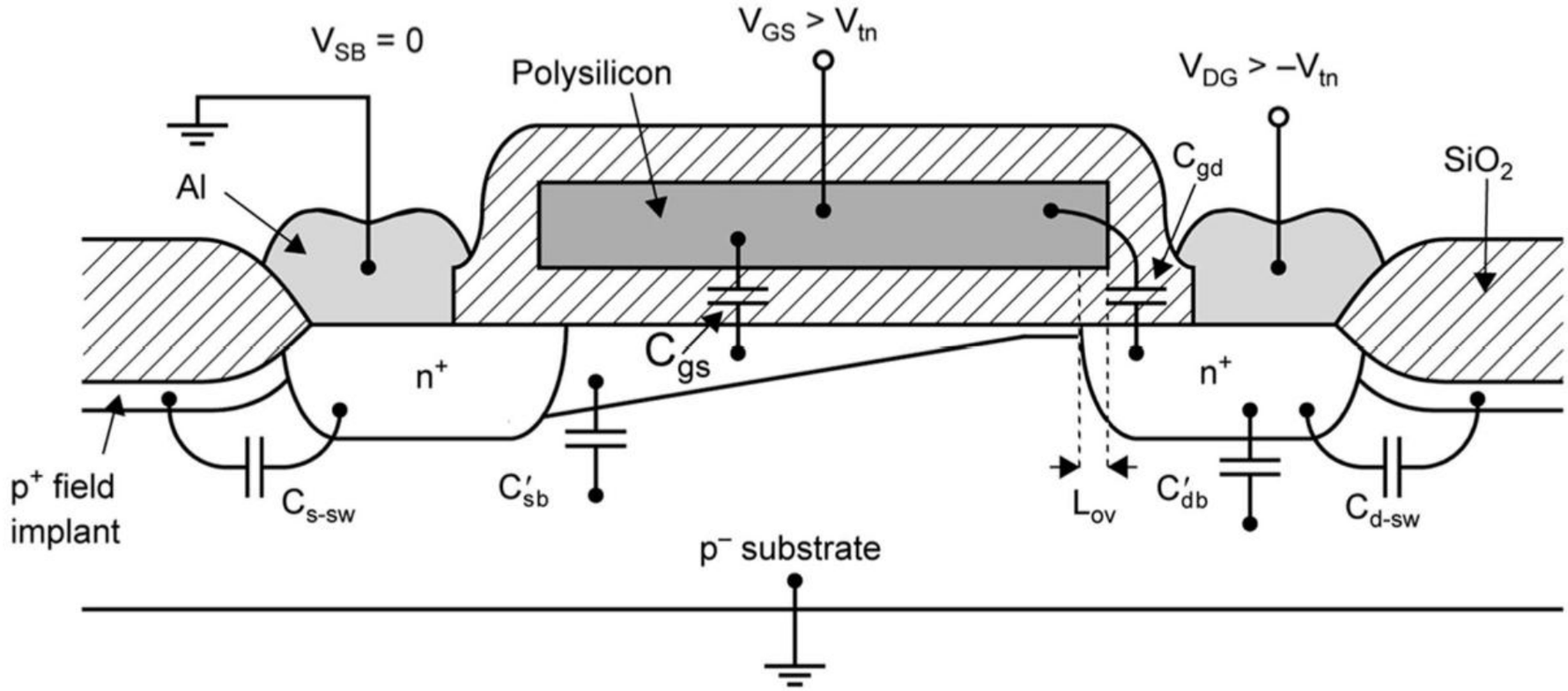
1. Gate capacitance of driven transistors.
2. Diffusion capacitance (due depletion region) of source/drain connected to the wire.
3. Wire capacitance.





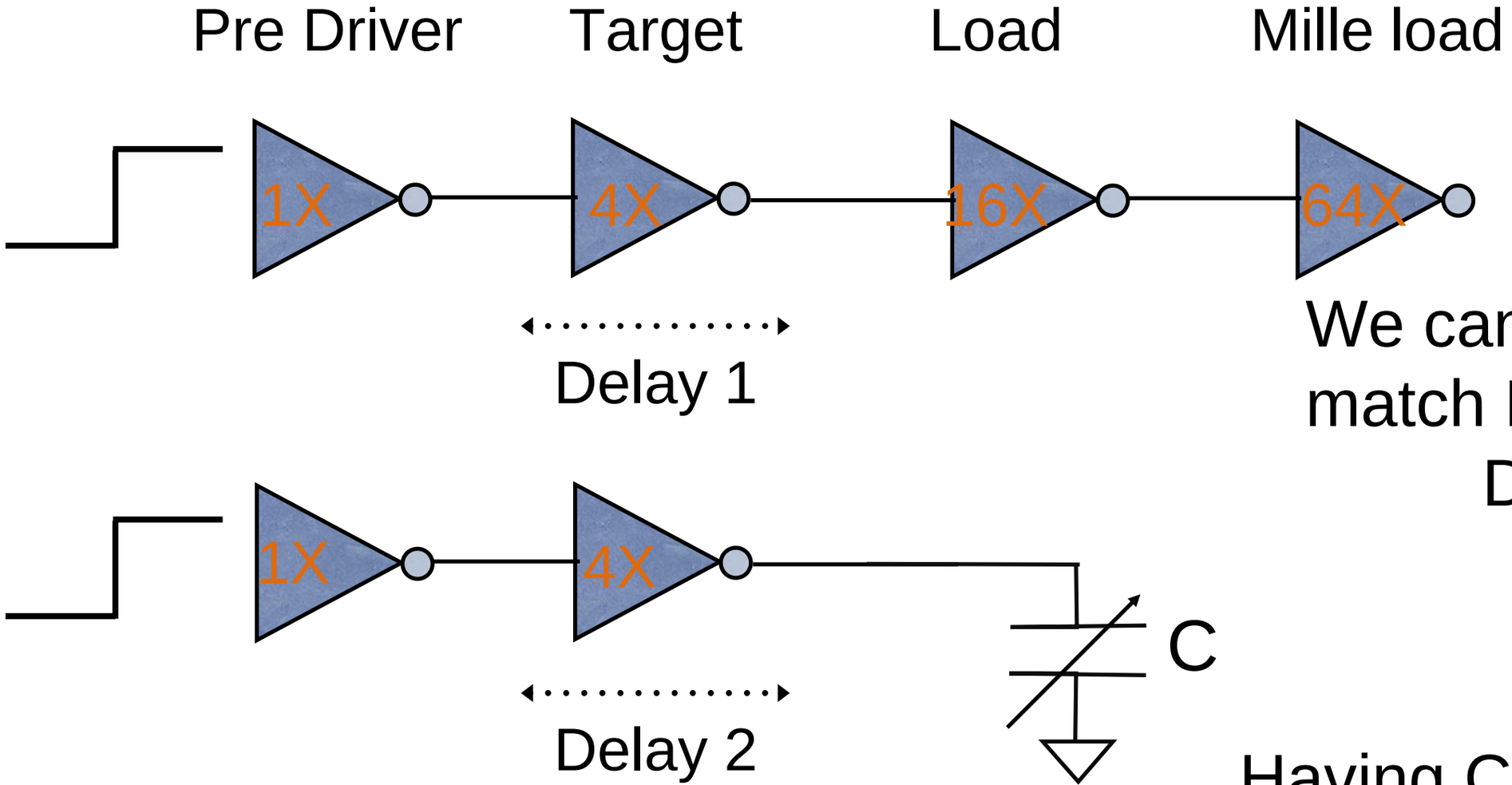


$$C_g = C_{ox}WL = C_{permicron}W, \qquad C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}}$$



Transistor Capacitance: Many Sources

+ Process Calibration



We can adjust C to match Delay 1 with Delay2

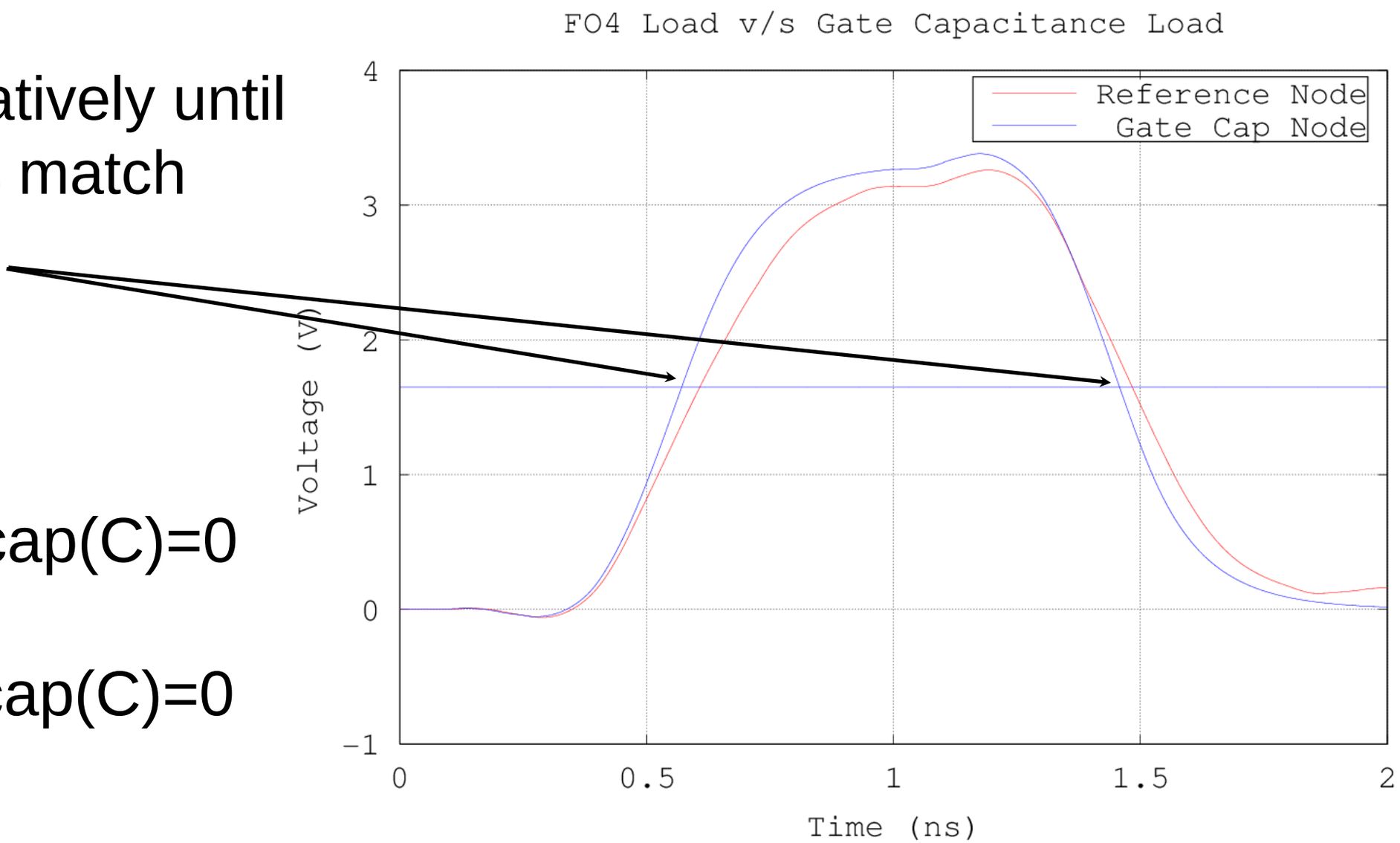
Having C, then R is just Delay / C*ln(2)

+ Process Calibration

Find C iteratively until Delays match

t_{pdr}- $t_{pdr\_cap(C)=0}$

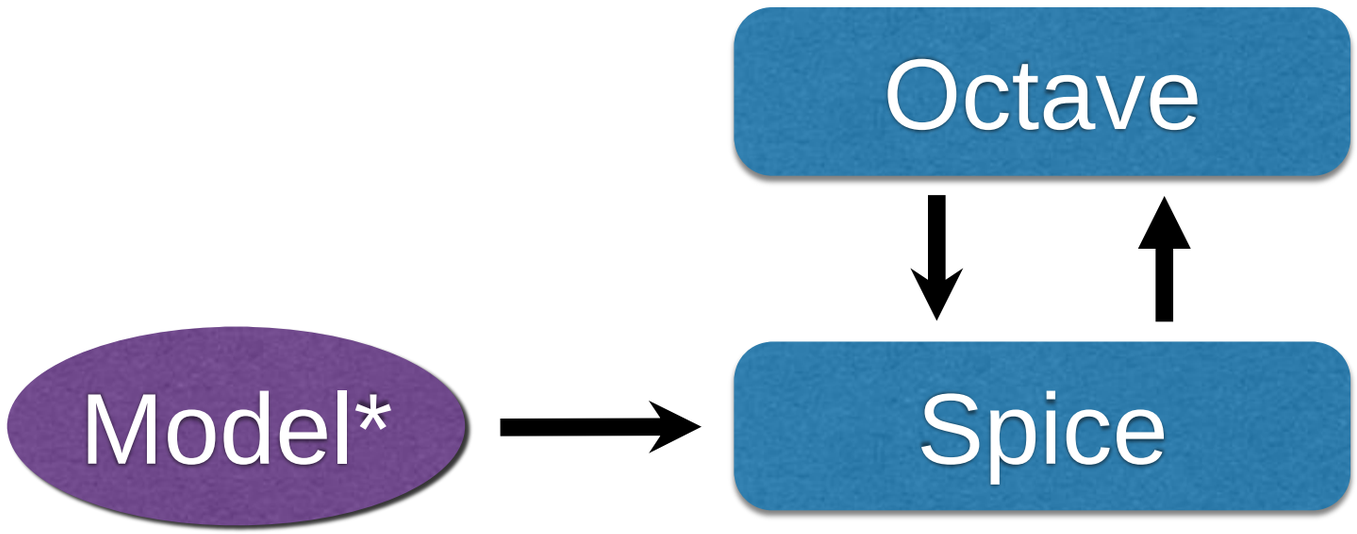
t_{pdf}- $t_{pdf\_cap(C)=0}$



Example

```
octave:1> source calib_opt.m
supply =  3.3000
cload_rise =    8.9632e-13
cload_fall =    7.6845e-13
cload =    8.3238e-13
WNDIFF_rise =  0.0015621
WNDIFF_fall =  0.0017749
WNDIFF =  0.0016685
delay =    2.6400e-10
cg =    1.4451e-15
Cdiff =    4.9887e-16
gamma =  0.23014
RN =  7321.1
RP =    1.4642e+04
Rsq =    3.6605e+04

========================
Cg = 1.445110 fF/um, Cd = 0.498875 fF/um Rsq = 36.605 kOhm/sq
```

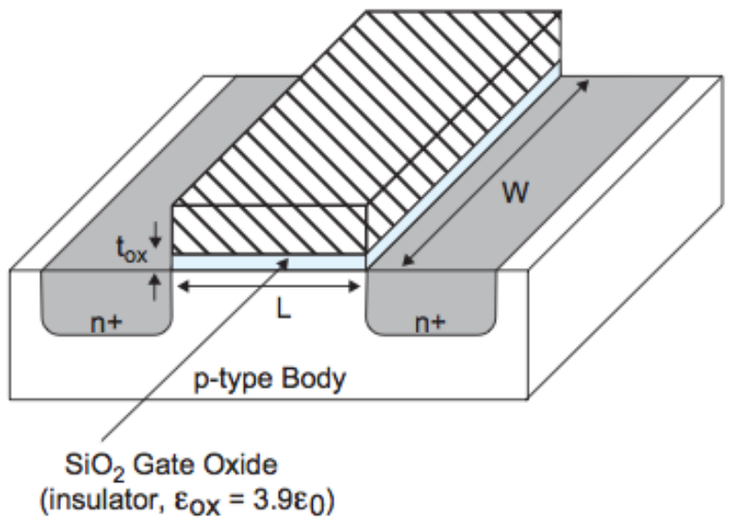


Process	Cgate	Cdiff	Rsq
0.5 um Pr05ocess	1.445 fF/um	0.499 fF/um	36.6 KOhm

* http://www.mosis.com/files/test_data/ami_c5n_corner_bsim3.txt

Current in a MOSFET

$$I_{ds} = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{gs} - V_t)^2$$



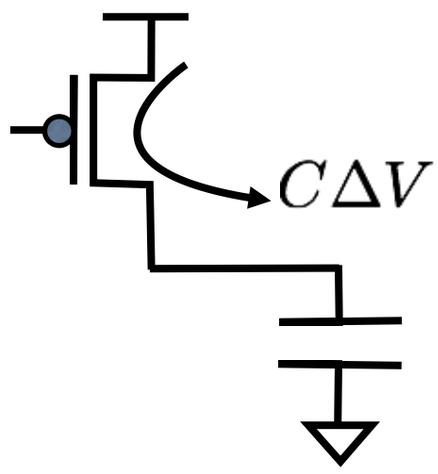
* Read Weste/Harris Ch3 for a complete derivation. Notice this is not even close to current in a real transistor.

Gate Power

What about Power?

- Well, Power = Energy / time
- And takes energy to move charges to a voltage source

  E = Q [cb] * V [J/cb]



- Now to drive a gate we need C*ΔV charges:

  E = C V^2

$$E_{source} = \int_0^{Tc} i_{dd} V_{dd} dt = V_{dd} C_{load} \int_0^{Tc} \frac{dV_{dd}}{dt} dt = V_{dd} C_{load} \int_0^{V_{dd}} dV = C_{load} V_{dd}^2$$

$$E_C = \int_0^{Tc} i_C V_C dt = C \int_0^{Tc} V_C \frac{dV_C}{dt} dt = C_{load} \int_0^{V_{dd}} V_C dV = \frac{1}{2} C_{load} V_{dd}^2$$

$$E_{disipated} = E_{source} - E_C = \frac{1}{2} C_{load} {V_{dd}}^2$$

Dynamic Power

- Now we know each **transition** dissipates some Energy:

$$E_{disipated} = \frac{1}{2} C_{load} {V_{dd}}^2$$

- Now, think about Power. How many times will a clock line take this Energy? Over which time?

$$P_{switching} = \frac{E}{T} = \frac{T f_{sw} (\frac{C_{load} {V_{dd}}^2}{2})}{T} = f_{sw} (\frac{C_{load} {V_{dd}}^2}{2})$$

- How many times does any signal transition per clock cycle? Let's say alpha.

$$P_{switching} = \alpha f_{sw} (\frac{C_{load} {V_{dd}}^2}{2})$$

Leakage Power

- Transistor don’t really shut down abruptly.
  - There is current even when Vgs < Vth
  - So cells leak current and this depend on Vth:

$$I_{leak} = I_0 e^{\frac{V_{gs} - Vth}{nV_T}}$$

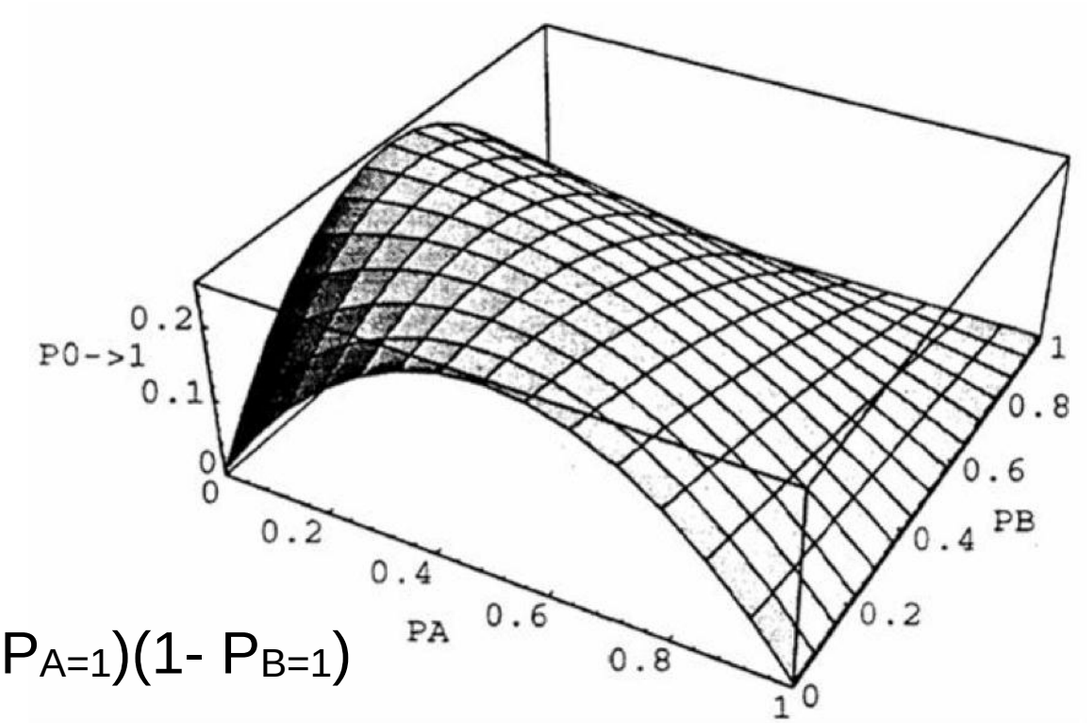
$$I_0 = I_{ds_{(Vgs=Vth)}}$$

*Leakage can drop 10x for each 80mV of Vth at room tº

Transition Probability

- Assume
  - $P_{A=1} = 1/2$
  - $P_{B=1} = 1/2$
- Then:
  - $P_{Y=1} = 1/4$
  - $P_{Y=0\rightarrow1} = P_{Y=0} * P_{Y=1} = 3/16$

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0



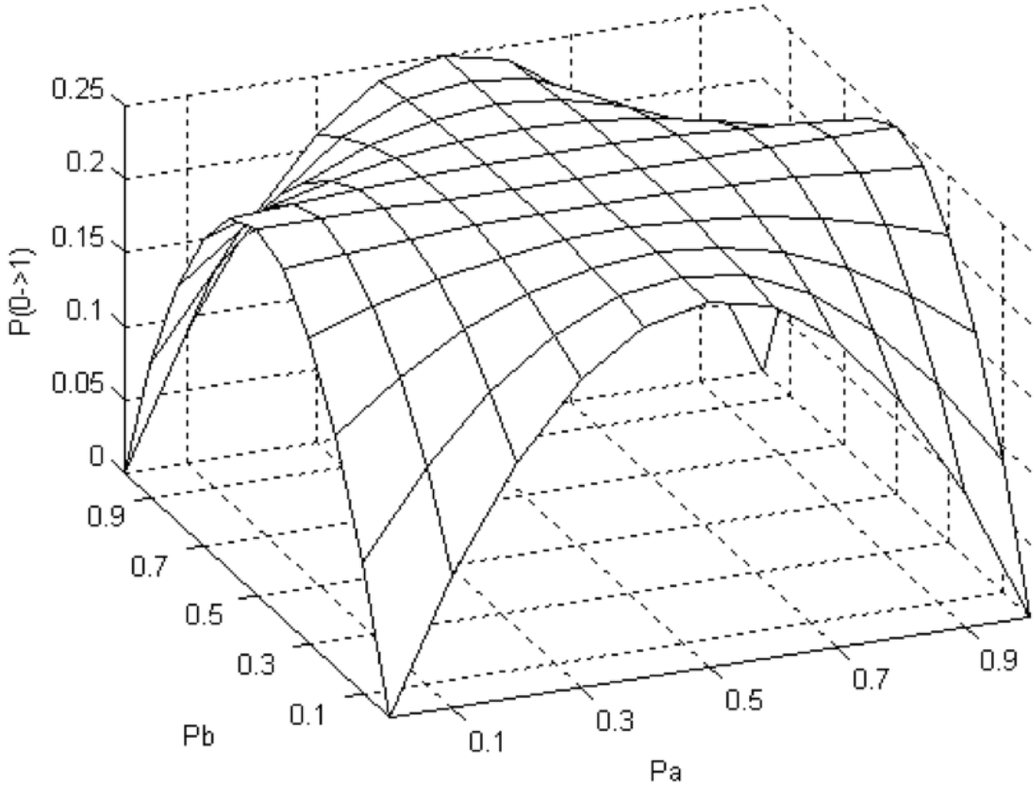
- Analytic expression:
  - $P_{Y=1} = (1- P_{A=1})(1- P_{B=1})$
  - $P_{Y=0\rightarrow1} = P_{Y=0}\, P_{Y=1} = (1-(1- P_{A=1})(1- P_{B=1}))(1- P_{A=1})(1- P_{B=1})$

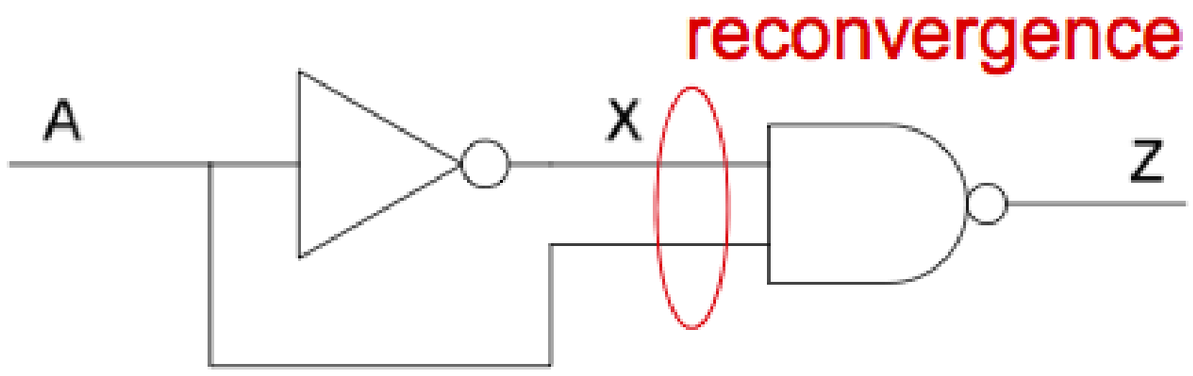
Transition Probability

- Assume
  - $P_{A=1}$= 1/2
  - $P_{B=1}$ = 1/2
- Then:
  - $P_{Y=1}$ = 1/2
  - $P_{Y=0\rightarrow1}$= $P_{Y=0}$*$P_{Y=1}$
  - $\alpha_{0\rightarrow1}$= 1/4
- $P_1 = P_A(1-P_B)+P_B(1-P_A) = P_A + P_B - 2P_AP_B$
- $P_{0\rightarrow1}$= P_0P_1

  =(1-P_A + P_B - 2P_AP_B)(P_A + P_B - 2P_AP_B)

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0





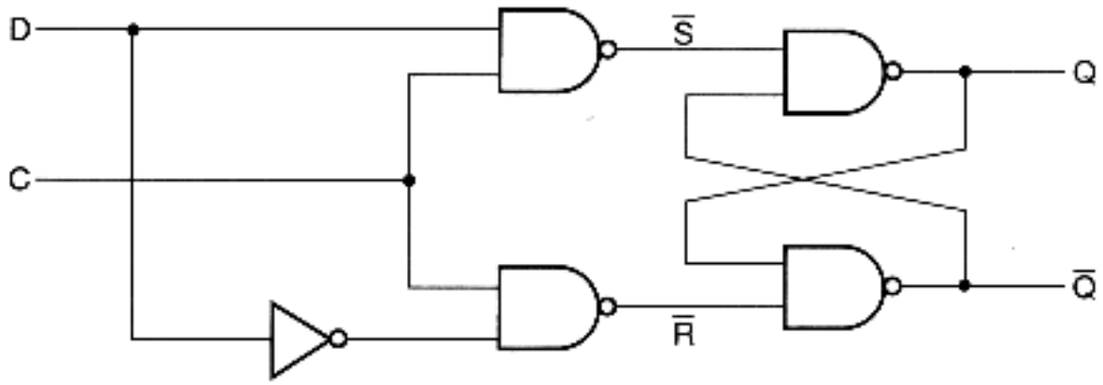
Reconvergence →Correlated inputs (NOT INDEPENDENT EVENTS)

$$P_Z = 1 - P_A * P(X/A) = 1$$

Conditional Probability
Probability of X provided A happens first

Reconvergent circuits quickly become hard to track and analyze

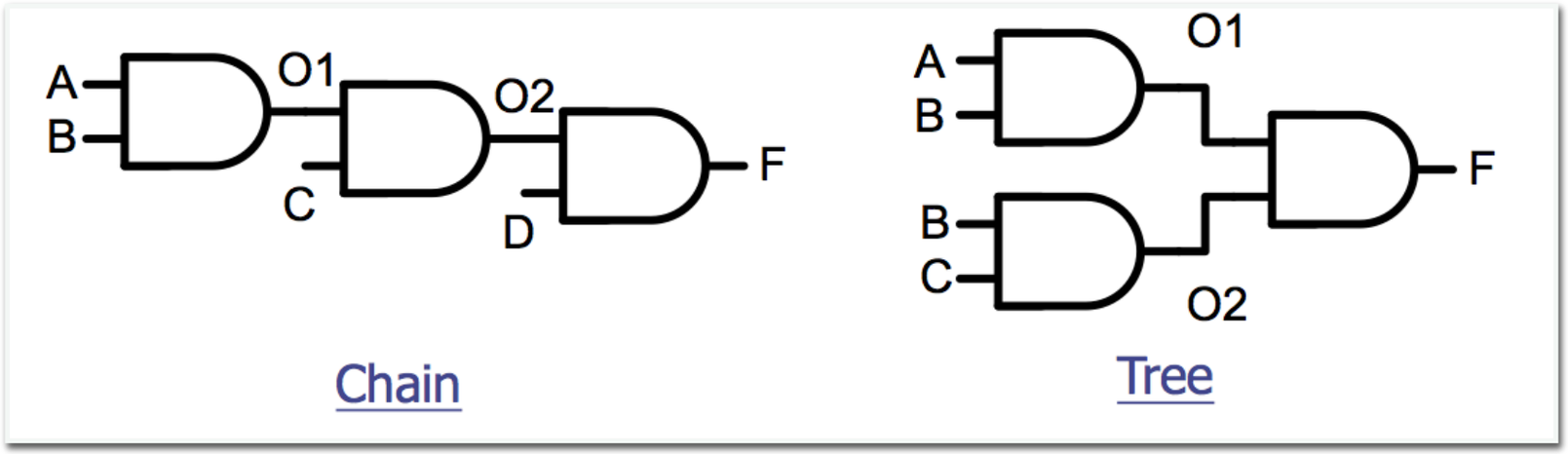
Power Estimation - Example



$$P = \frac{\alpha}{2} C V^2 f$$

Node	C	D	S'	R'	Q	Q'
Cycle 1	0	0	1	1	0	1
Cycle 1	1	1	0	1	1	0
Cycle 2	0	1	1	1	1	0
C						
Cycle 2	1	1	0	1	1	0
Transitions	**2**	**1**	**2**	**0**	**1**	**1**
Activity Factor Factor	**1**	**0.5**	**1**	**0**	**0.5**	**0.5**

* What if gates don't change instantly:
  - Need to consider transient states too (hint: glitches)

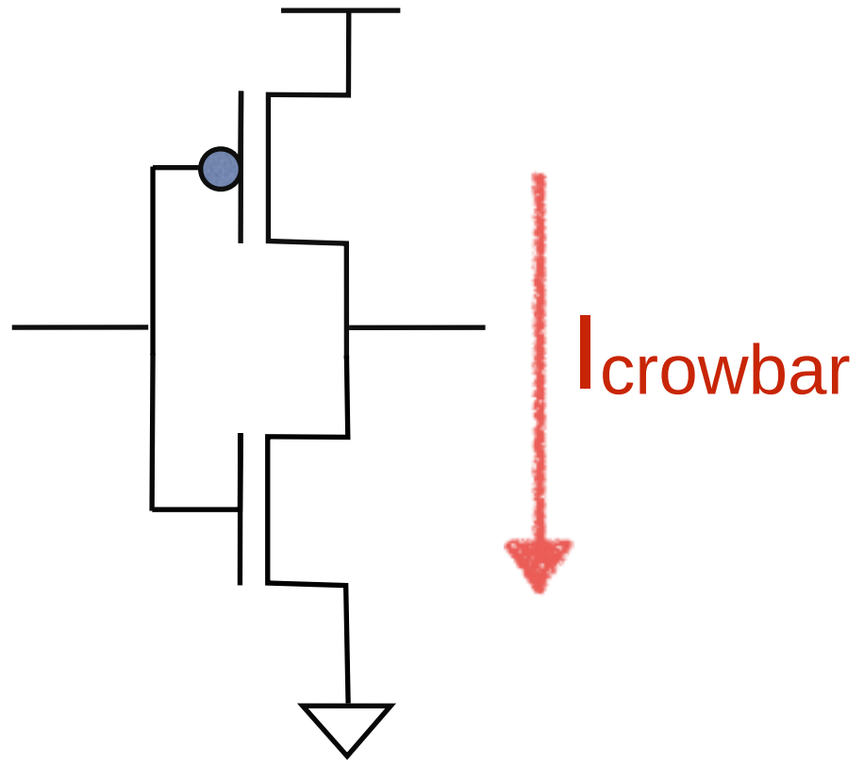


Which has higher activity factor?
(more in homework)

+Crowbar current

- During circuit commutation both transistor may be active at the same time.

- On actual chips long transition time can make power dissipation to increase within 5-10% due crowbar current



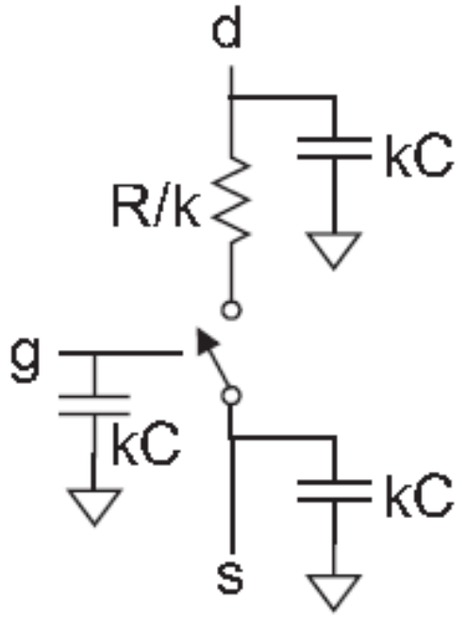
Can add this 5-10% to your hand calculations

System Analysis

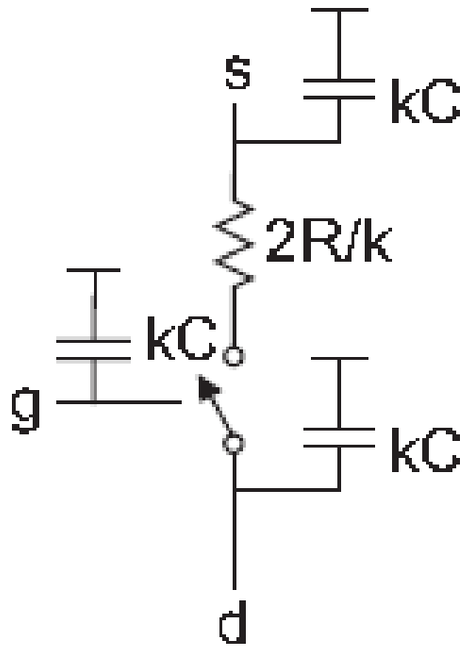
RC Delay model

- This model approximates the nonlinear transistor I-V and C-V characteristics with an average R and C
- Works well for delay estimation (not so well for analog behavior)
- Equivalent resistance R is the ratio Vds/Ids averaged across the switching interval of interest.
- A unit nMOS transistor is defined to have a resistance of R and gate and diffusion capacitances of C
- Transistors can be modeled based on the unit transistors

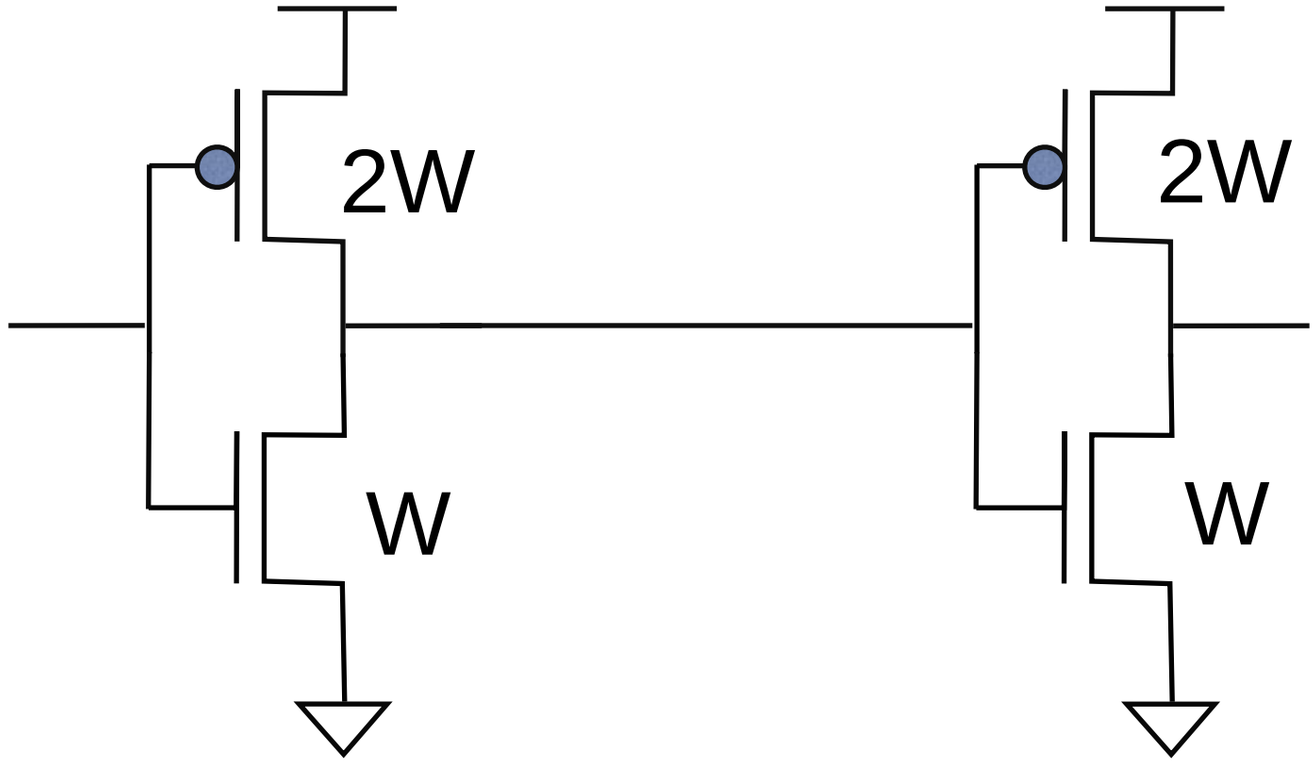
nMOS



pMOS



Gate Performance

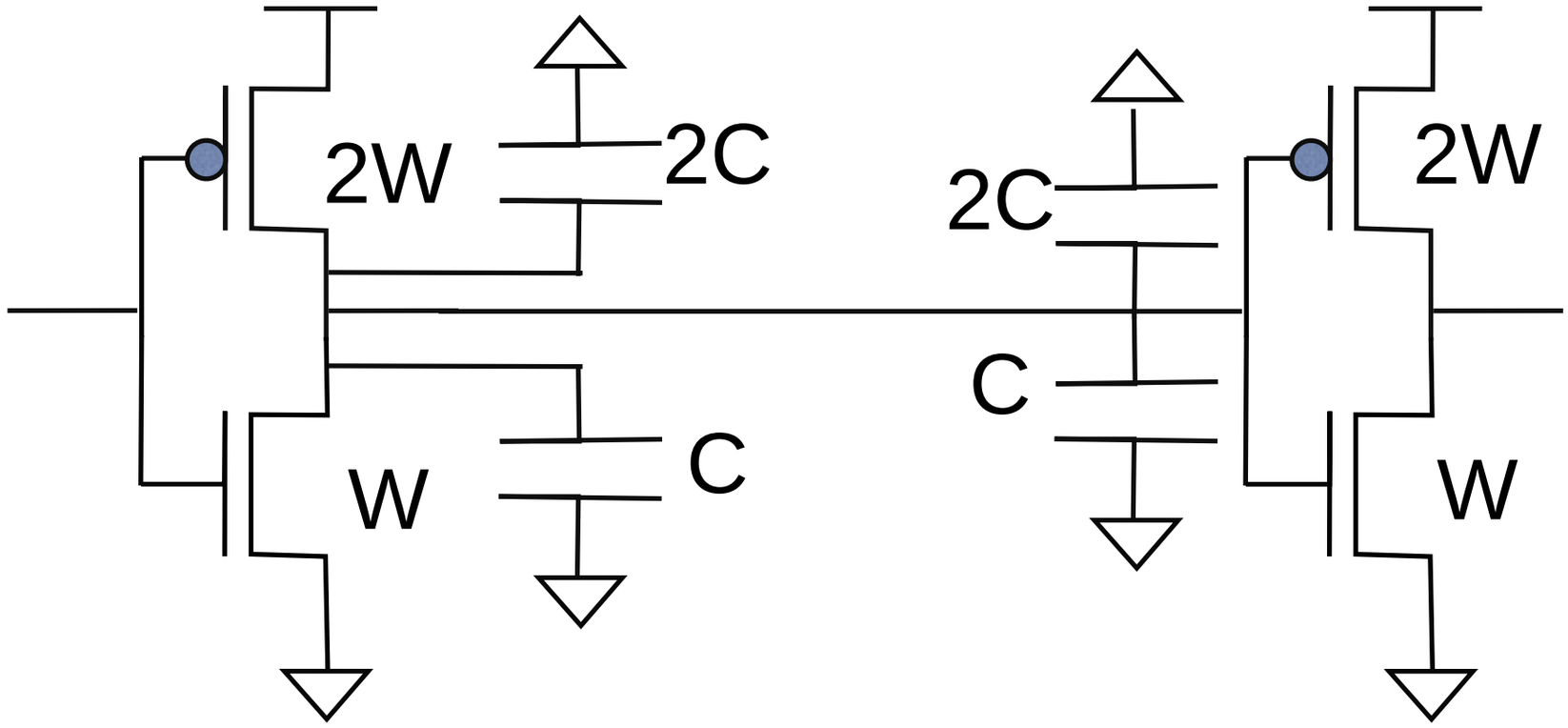


- Can you explain the transistor sizes?
- What is the value of Cload?

$$I_{DS} = \frac{\mu C_{ox} W}{L} (V_{gs} - V_t)^2$$

Gate Performance

To perform delay analysis, we need to calculate R and C!



- Let’s say that both Cg and Cdiff are proportional to W

$$C_g = C_{g,cons}W \qquad C_{par} = C_{p,cons}W$$

- Resistance can be approximated to $R = \frac{W}{L}R_{sq}$

Sizing for minimum delay

- Wider transistors reduce R ($R = \frac{L}{W} R_{sq}$), thus delay (RC) should decrease. Right?
- Wait a second... What about $C_g = C_{g,cons} W$?

Will also grow, making the previous gate slower!

Modelling inverter delay

$$t_{inv} = R_{drive}(C_{par} + C_{load})$$

$$C_{in} = 3WC_g$$

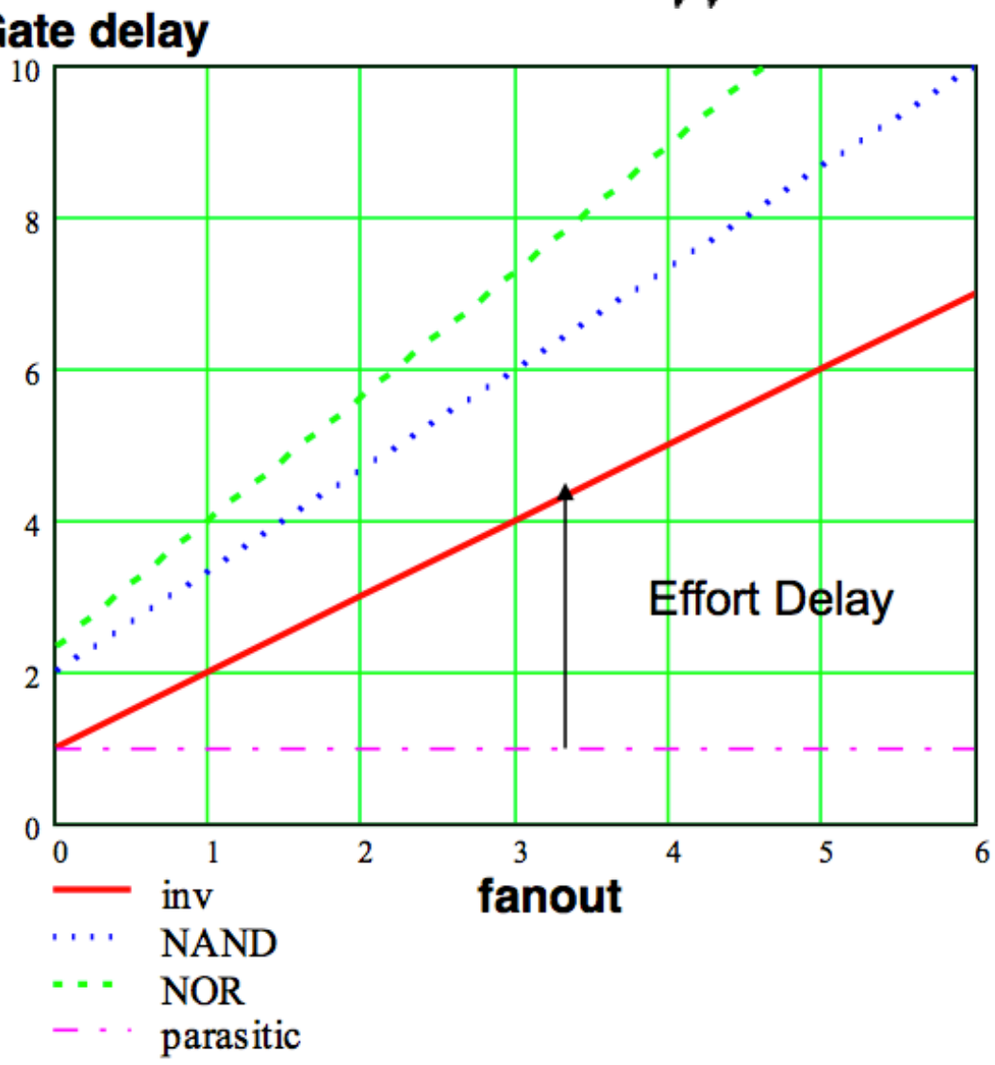
$$C_{par} = \gamma C_{in}$$

$$R_{drive} = R_{sq}\frac{L}{W}$$

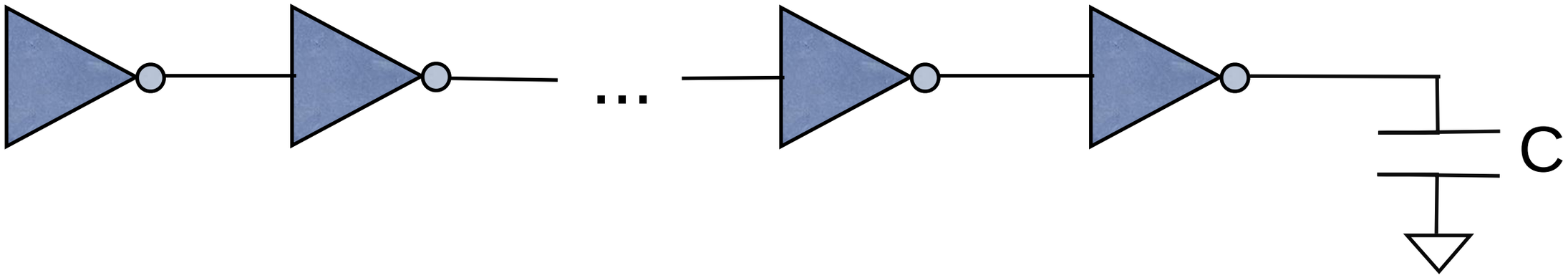
$$t_{inv} = R_{drive}C_{in}(\frac{C_{par}}{C_{in}} + \frac{C_{load}}{C_{in}})$$

- Some algebra and...

$$t_{inv} = 3R_{sq}C_g(\gamma + \underbrace{\frac{C_{load}}{C_{in}}}_{\text{Fanout}})$$



Optimizing an inverter chain



$$TotalDelay = \sum_{j=1}^{N} \tau_{inv}(\gamma + \frac{C_{in}^{j+1}}{C_{in}^{j}})$$

$$\tau_{inv} = 3R_gC_g$$

* How can we find optimum?

Optimizing an inverter chain

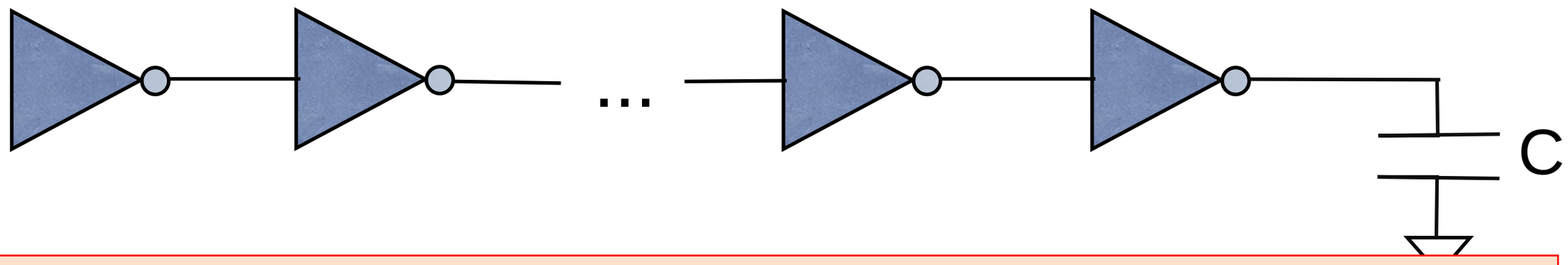
$$\frac{d}{dC_{in}^{j}} Delay = \tau_{inv}(\frac{1}{C_{in}^{j-1}}) - \tau_{inv}(\frac{C_{in}^{j+1}}{(C_{in}^{j})^2})$$

When equal to zero:

$$C_j = \sqrt{C^{j+1} * C^{j-1}}$$

Or

$$\frac{C_{in}^{j}}{C_{in}^{j-1}} = \frac{C_{in}^{j+1}}{C_{in}^{j}}$$



At the optimum, each inverter does the same effort!

If all do the same effort, how much then?

- Let's say at the optimum *f* is the fanout of every gate. If we multiply all the fanouts, we have

$$f = \sqrt[N]{\frac{\cancel{C2}}{C1}\frac{\cancel{C3}}{\cancel{C2}}\cdots\frac{C_{N+1}}{\cancel{C_N}}}$$

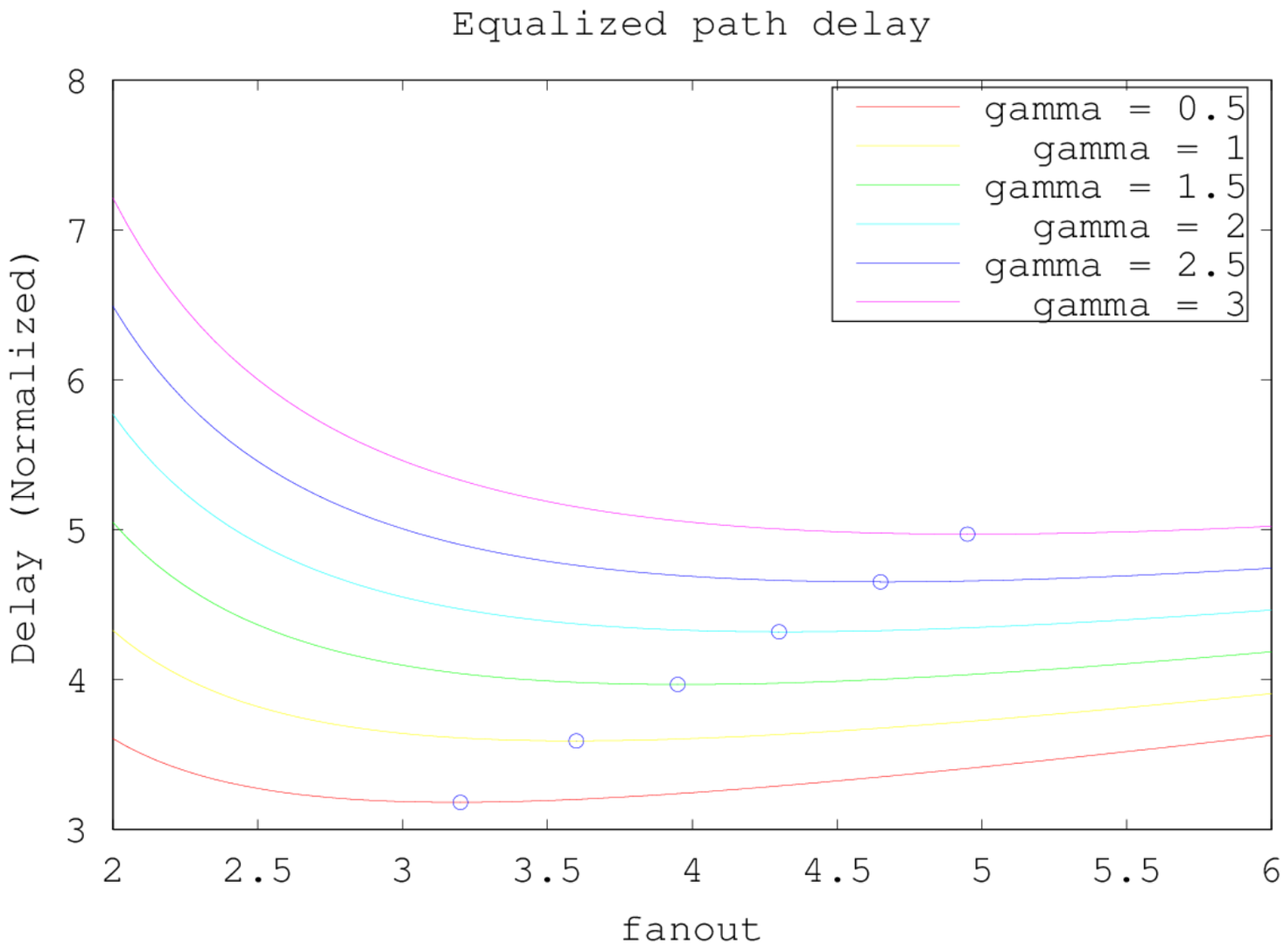
$$N = \frac{ln(\frac{C_{out}}{C_{in}})}{ln(f)}$$

- On the other hand, at the optimum we have

$$TD = \sum_{j=1}^{N} \tau_{inv}(\gamma + \frac{C_{in}^{j+1}}{C_{in}^{j}}) = N\tau_{inv}(\gamma + f)$$

$$TD = ln\left(\frac{C_{out}}{C_{in}}\right)\tau_{inv}\frac{(\gamma + f)}{ln(f)}$$

If all do the same effort, how much then?



To find the optimum, we derive *TD* with respect to *f*

$$TD = ln\left(\frac{C_{out}}{C_{in}}\right)\tau_{inv}\frac{(\gamma + f)}{ln(f)}$$

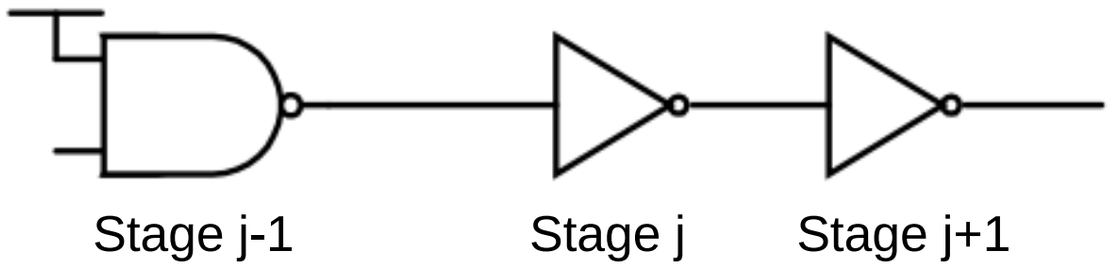
This is done graphically…

Optimum at roughly *f = 4* (FO4)

But in reality…

Circuits are not made of inverters only

$$\tau_{j-1} \neq \tau_j \neq \tau_{j+1} \cdots$$
$$\gamma_{j-1} \neq \gamma_j \neq \gamma_{j+1} \cdots$$



At optimum

$$\tau_{nand} f_{nand} = \tau_{inv} f_{inv}$$

$$\tau_{nand} \frac{C_j}{C_{j-1}} = \tau_{inv} \frac{C_{j+1}}{C_j}$$

Note that now fanouts are not equal, they depend on τ

To optimize a heterogeneous chain, we normalize by τ_{inv}

Keep an eye on this term

$$\frac{\tau_{nand}}{\tau_{inv}} \frac{C_j}{C_{j-1}} = \frac{\tau_{inv}}{\tau_{inv}} \frac{C_{j+1}}{C_j}$$

Logical Effort

$$LE_{gate} = \frac{\tau_{gate}}{\tau_{inv}} = \frac{C_{in,gate} R_{drive,gate}}{C_{in,inv} R_{drive,gate}}$$

If we normalize the fanout ecuation at optimum, it becomes:

$$\tau_{nand} \frac{C_j}{C_{j-1}} = \tau_{inv} \frac{C_{j+1}}{C_j} \Leftrightarrow LE_{nand} \frac{C_j}{C_{j-1}} = LE_{inv} \frac{C_{j+1}}{C_j}$$

Note that by definition $LE_{inv} = 1$

***LE* of a logic gate tells how much more slowly the gate will drive a load compared to an inverter**

Delay with Logical Effort

If we remember,:

$$TD = \sum_{j=1}^{N} \tau_j \left(\gamma + \frac{{C_{in}}^{j+1}}{{C_{in}}^{j}}\right)$$

Normalizing by τ_{inv}:,

$$TD_{norm} = \sum_{j=1}^{N} LE_j \left(\gamma + \frac{{C_{in}}^{j+1}}{{C_{in}}^{j}}\right)$$

Then, deriving and equaling to 0, at optimum we have

$$LE_1 \frac{C_2}{C_1} = LE_2 \frac{C_3}{C_2} = \cdots = LE_{N-1} \frac{C_N}{C_{N-1}}$$

Stage Effort (SE)

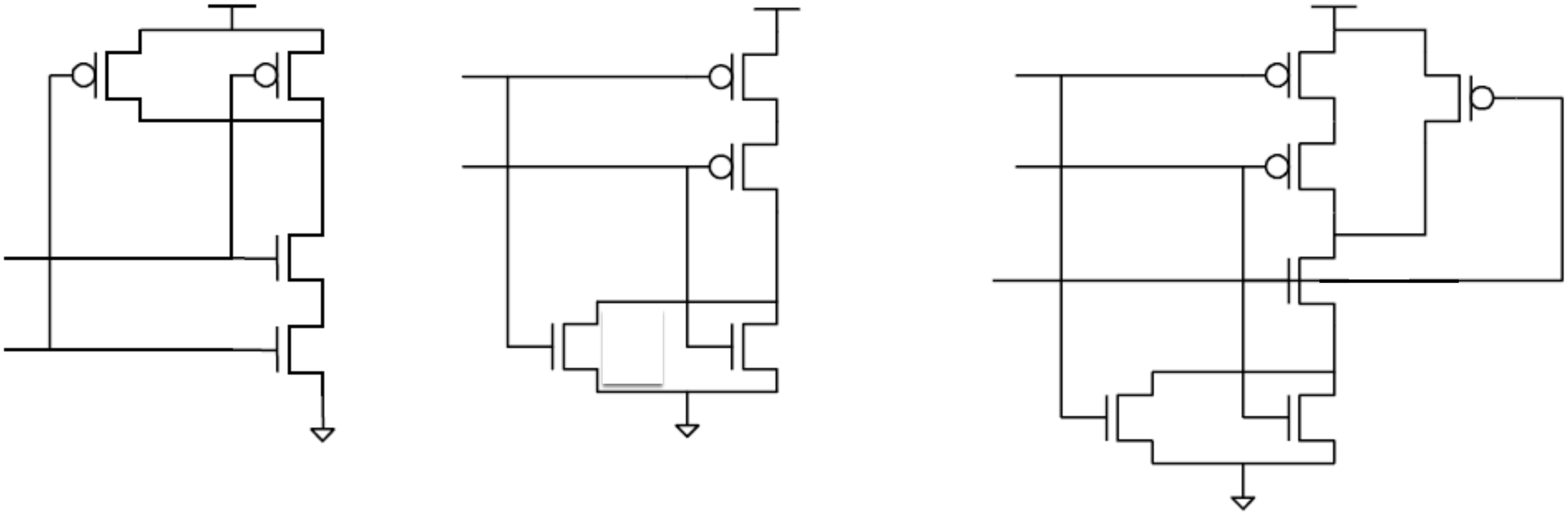
We can calculate the *Effective Fanout* as

$$f_{eff} = \sqrt[N]{\prod_{j=1}^{N} LE_j \frac{C_{in,j+1}}{C_{in,j}}} = \sqrt[N]{\frac{C_{in,N}}{C_{in,1}} \prod_{j=1}^{N} LE_j}$$

Path Logical Effort (LE_{path})

Path Fanout (f_{path})

Sizing - Example



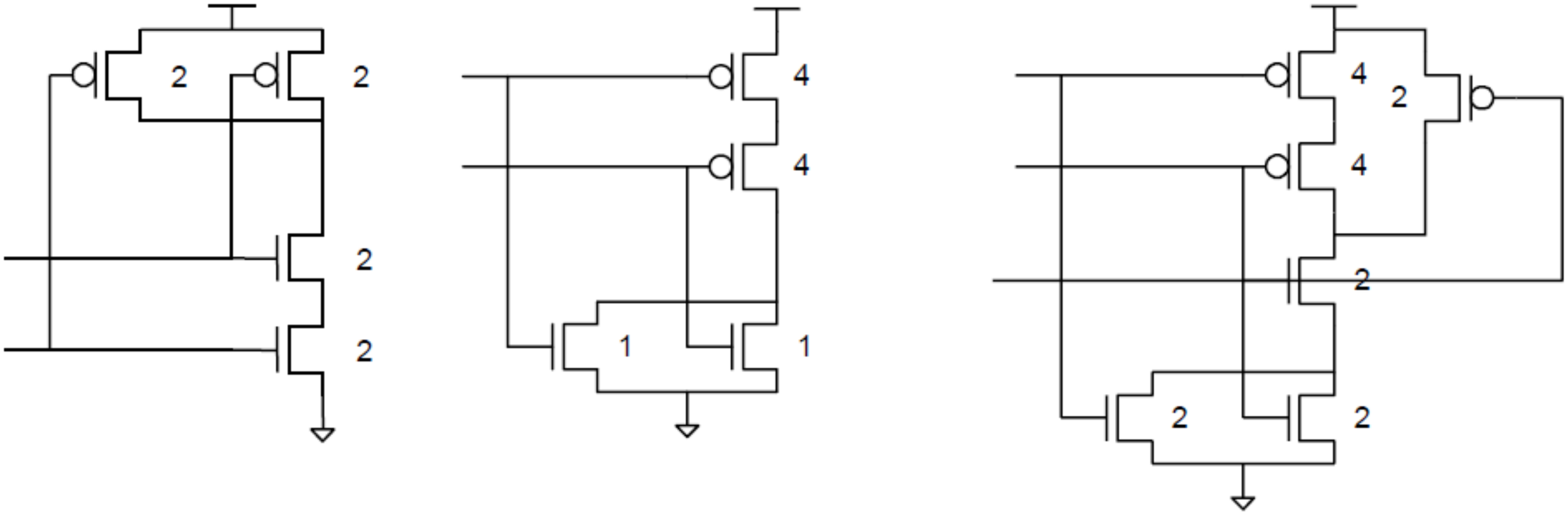
How do we size?
Is LE equal for every output?

$$C_g = C_{g,cons} W$$

$$R = \frac{W}{L} R_{sq}$$

Hint: We want falling and rising times to be equal
Also, rising and falling times should match inverter times

Sizing - Example



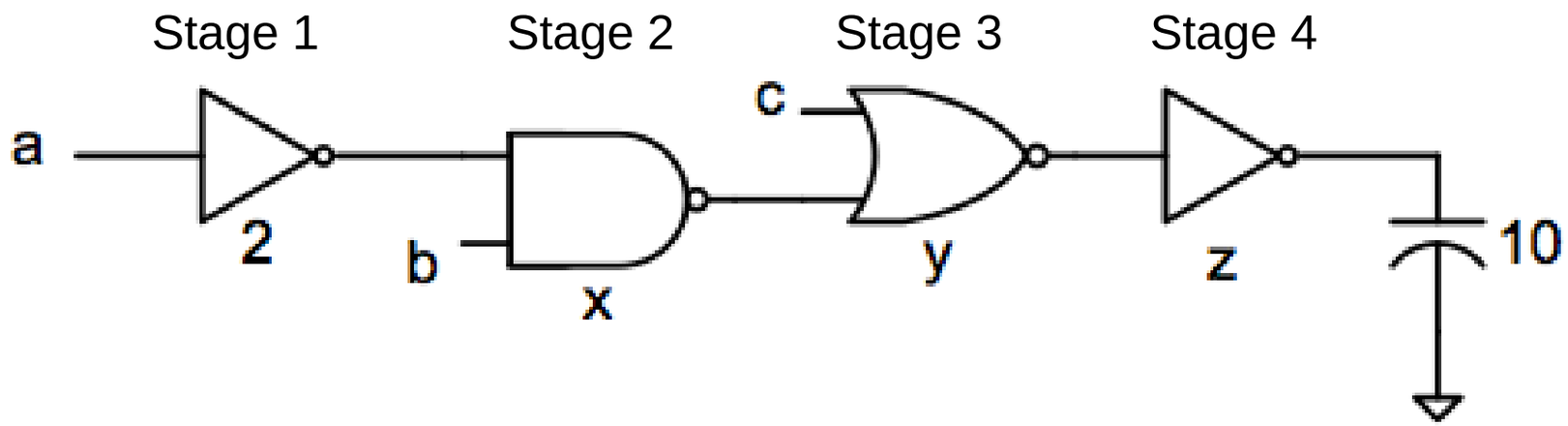
How do we size?

Is LE equal for every output?

Hint: We want falling and rising times to be equal

Also, rising and falling times should match inverter times

Optimizing Path Delay Example



Stage	1	2	3	4	
LE	1	4/3	5/3	1	$LE_{path} = 2.2$
Fanout	x/2	y/x	z/y	10/z	$f_{eff} = 1.8$
Size	2	3.6	4.86	5.25	

$$SE = f_{eff}$$

$$LE_j \frac{C_{j+1}}{C_j} = f_{eff}$$

$$\sqrt[4]{\frac{10}{2} * 2.2} = 1.8$$

$$\frac{x}{2} \cdot 1 = 1.8$$

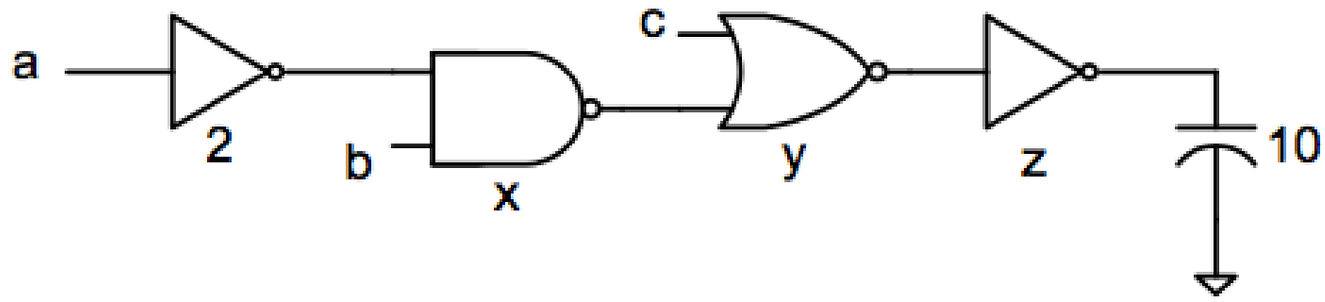
$$\frac{y}{x} \cdot \frac{4}{3} = 1.8$$

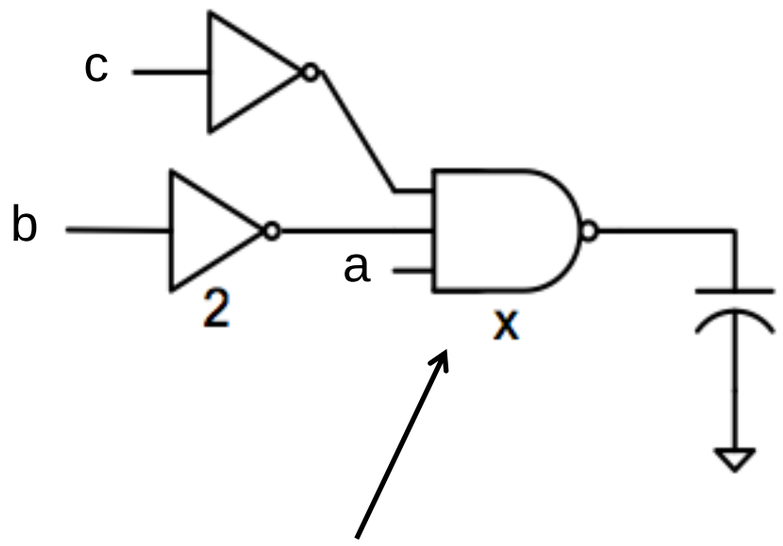
$$\dots$$

Change Topology so effective fanout is closer to 4

$$f = a + \bar{b}a = \overline{\overline{c + b + \bar{a}}}$$

$$f = \overline{\overline{(c+b)} \cdot a} = \overline{\bar{c}\bar{b}a}$$





$LE_{3-input\_NAND} = 5/3$

$$LE_{path} = 1 * \frac{5}{3} = 1.66$$

$$f_{eff} = \sqrt{1.66 * \frac{10}{2}} = 2.88$$

Better!

Review: To Size a Path

1. Find Effective Fanout

$$f_{eff} = \sqrt[N]{\frac{C_{in,N}}{C_{in,1}} \prod_{j=1}^{N} LE_j} = \sqrt[N]{f_{path}}$$

2. Estimate optimal number of stages. We want f_{eff} to be close to 4 at optimum

$$\widehat{f_{eff}} = 4 = \sqrt[\widehat{N}]{f_{path}} \Leftrightarrow \widehat{N} = log_4(f_{path})$$

3. Re-implement using $\widehat{N}$ stages. Re-calculate LE_{path} and f_{eff}

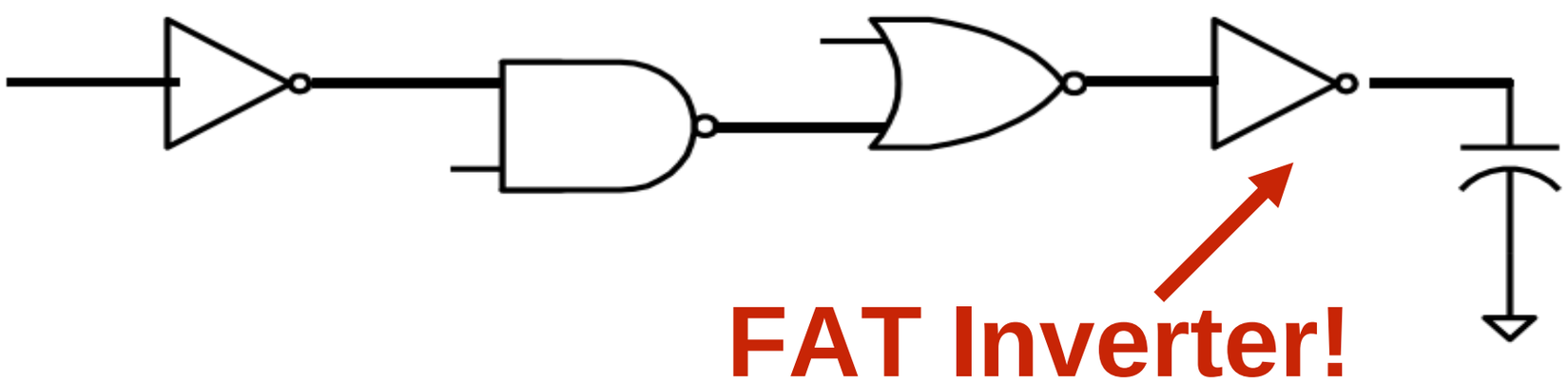
4. Calculate sizes, start from either end

$$LE_j \frac{C_{in,j+1}}{C_{in,j}} = f_{eff}$$

LE Energy Considerations

$$-\frac{\partial E/\partial W_i}{\partial D/\partial W_i} = -\frac{\partial E/\partial C_i}{\partial D/\partial C_i} \cdot \frac{\partial C_i/\partial W_i}{\partial C_i/\partial W_i} = -\frac{ec_i/C_i}{\tau \cdot \left(\frac{1}{C_{i-1}} - \frac{C_{i+1}}{C_i^2}\right)} = \frac{ec_i}{\tau \cdot (h_{eff_i} - h_{eff_{i-1}})}$$

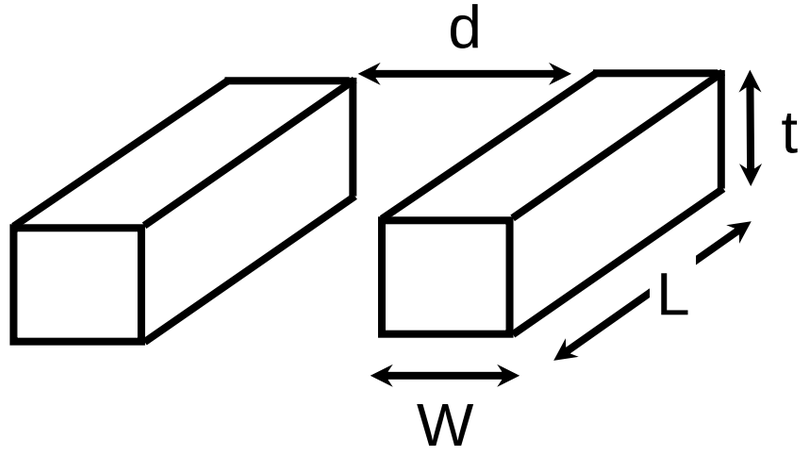
Bigger cells give less performance per energy spent.
If we want to <u>equalize marginal cost,</u> then **BIG** cells should have **HIGHER** fanout.



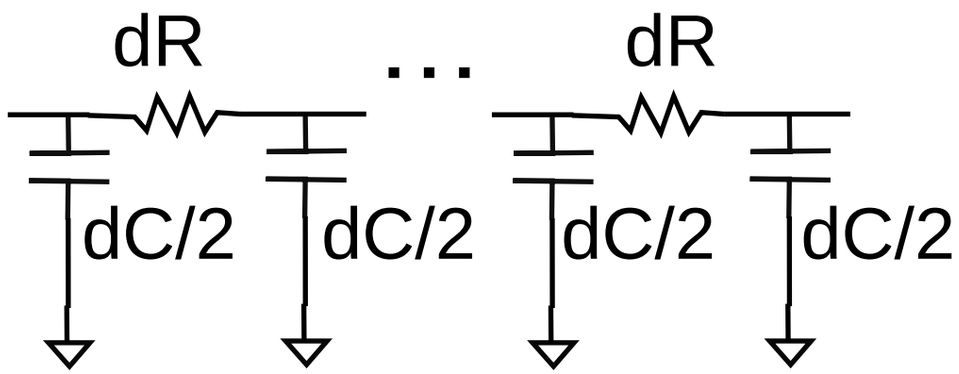
Notes on:
The Cost of Wires

- Characterized by R & C
  (Oversimplified aproximation)

$$R_w = \frac{\rho}{tW} \times L \quad C_w = \frac{\varepsilon t}{d} \times L$$



- But propagations follows pi-model



$$Delay = \frac{1}{2} R_w C_w$$

- **R α L, C α L ⇒ Delay α L^2**

Using optimally spaced repeaters we can make it α L

Summary

- Delay is proportional to ln(2)*RC.
- Power can be dynamic (P= a/2 CV^2 x f) and static (Ileak*Vdd).
- C is load + parasitic.
- Optimum chain delay require equalized effective fanout.

References

- Weste, Harris - CMOS VLSI Design 4th Edition
- Rabaey, Digital Integrated Circuits
- Sutherland, Sproull, Harris - Logical Effort: Designing Fast CMOS Circuits
- Kahng, Lienig, Markov, Hu - VLSI Physical Design: “From Graph Partitioning to Timing Closure”