ECE 6745 Complex Digital ASIC Design
Topic 9: CMOS Combinational Logic

School of Electrical and Computer Engineering
Cornell University

revision: 2025-02-13-11-19

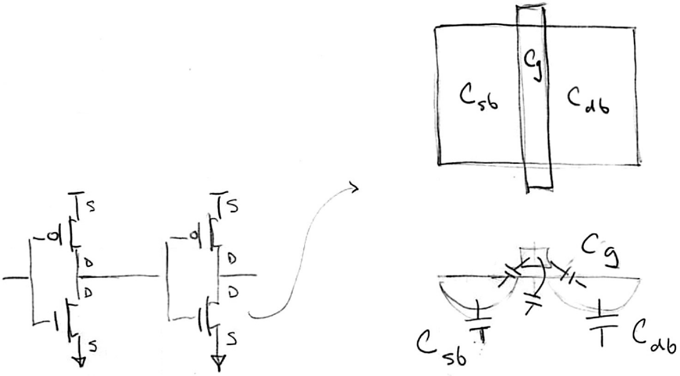
1 RC Modeling 3

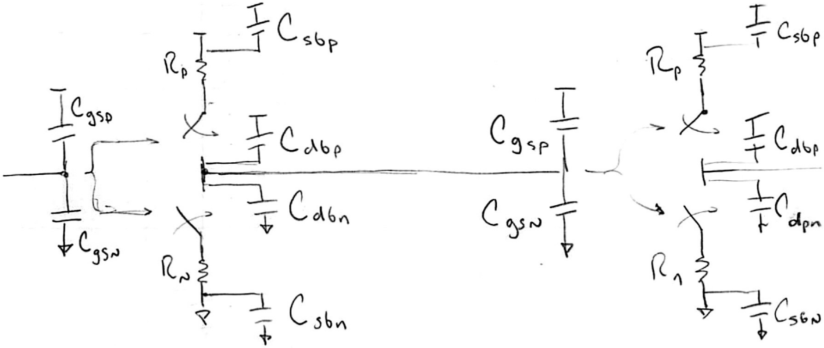
2 Delay 5
- 2.1. RC Delay of Inverter 5
- 2.2. RC Delay of 2-Input NAND Gate 6
- 2.3. Equal Rise/Fall Times 9
- 2.4. Equal Drive Strength 10
- 2.5. Larger Gates 11
- 2.6. Larger Loads 12
- 2.7. Comparison of Inverter, NAND, NOR Gates 13
- 2.8. Logical Effort: Single Stage 16
- 2.9. Logical Effort: Multiple Stages 22

3 Energy 37

4 Area 44

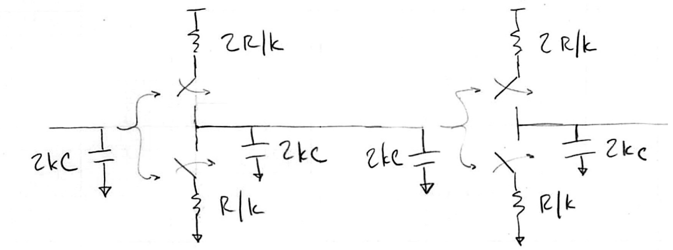
1. RC Modeling



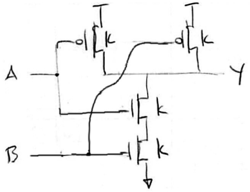


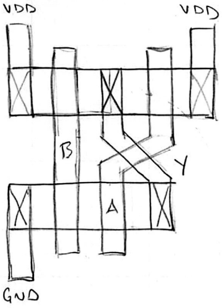
- C_{sb} capacitors do not actually switch, so ignore
- Lump $C_{dbp} + C_{dbn}$ since both tied to constant nodes
- Lump $C_{gsp} + C_{gsn}$ since both tied to constant nodes
- Assume PMOS mobility is 2× worse than NMOS mobility

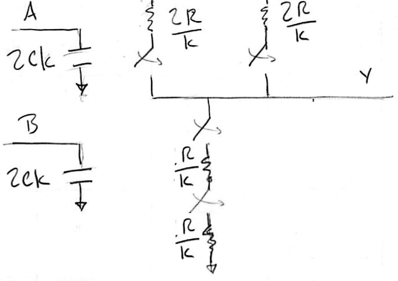
- Let *C* be the gate capacitance of minimum sized NMOS
- Let *R* be the effective resistance of a minimum sized NMOS
- Let *k* be width of a transistor relative to minimum sized NMOS



2-Input NAND Gate





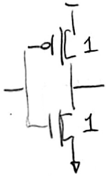


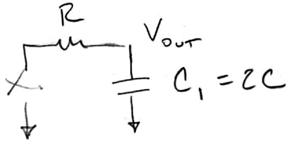
Draw and label the parasitic capacitances.

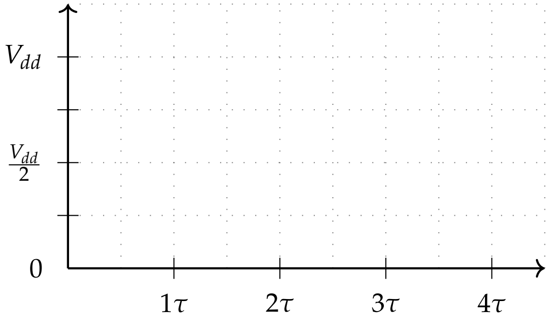
2. Delay

- We will initially use RC modeling to estimate delay
- We will then use RC modeling to derive logical effort (LE)
- LE is a fast way to estimate delay for simple static CMOS circuits
- Often need to use a mix of RC modeling and LE

2.1. RC Delay of Inverter







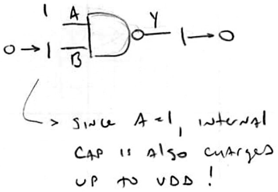
- Let t_{pd} be the propagation delay, time until $V_{out} = V_{dd}/2$

$$
\begin{aligned}
V_{out} &= V_{dd}\, e^{-t/\tau} \\
\frac{V_{dd}}{2} &= V_{dd}\, e^{-t/\tau} \\
\frac{1}{V_{dd}}\frac{V_{dd}}{2} &= e^{-t/\tau} \\
\ln\left(\frac{1}{2}\right) &= \frac{-t}{\tau} \\
-\tau \ln\left(\frac{1}{2}\right) &= t \\
t &= \tau \ln(2)
\end{aligned}
$$

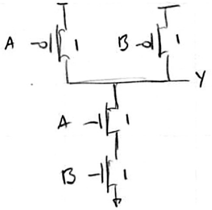
- So $t_{pd} = \ln(2) \cdot RC_1$
- Let $R' = \ln(2) \cdot R$, so $t_{pd} = R'C_1$
- For inverter on previous page, $t_{pd} = 2R'C$
- We usually just assume effective resistance is scaled by ln(2)
- So propagation delay of inverter on previous page:

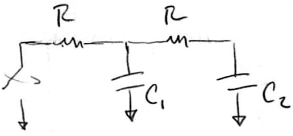
$$\boxed{t_{pd} = 2RC}$$

2.2. RC Delay of 2-Input NAND Gate



Since A = 1, internal cap is also charged up to VDD!





$C_1 = C$

$C_2 = 3C$

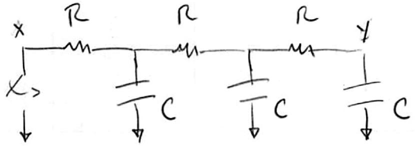
- Requires complicated 2nd order model
- We can use a simple approximation

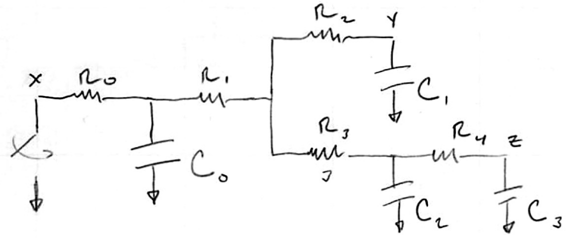
$$
\begin{aligned}
\tau = \tau_1 + \tau_2 &= RC_1 + (R+R)C_2 \\
&= RC + (2R)(3C) \\
&= RC + 6RC = 7RC \quad \text{(3.5× slower than inverter)}
\end{aligned}
$$

- Best when one τ much larger than the other τ
- Even if $\tau_1 = \tau_2$, error is $< 15\%$

Generalized Elmore Delay

$$t_{pd} = \sum_{i}^{\text{all nodes}} R_{ij} C_i$$





Assume all resistances are R and all capacitances are C

- Delay of path from x to y is impacted by branch to z
- Delay of path from x to z is impacted by branch to y
- For path x to y, lump $C_2 + C_3$ and use shared resistance $R_0 + R_1$
- For path x to z, lump C_1 and use shared resistance $R_0 + R_1$
- This extra term estimates impact of delay due to "branch"

$$\begin{aligned} T_{pd,xy} &= R_0C_0 + (R_0 + R_1 + R_2)C_1 + (R_0 + R_1)(C_2 + C_3) \\ &= RC + 3RC + 4RC = 8RC \end{aligned}$$

$$\begin{aligned} T_{pd,xz} &= R_0C_0 + (R_0 + R_1 + R_3)C_2 + (R_0 + R_1 + R_3 + R_4)C_3 + (R_0 + R_1)C_1 \\ &= RC + 3RC + 4RC + 2RC = 10RC \end{aligned}$$

Use Elmore Delay to Estimate Rise/Fall Times for 2-Input NAND Gate

$t_{pd,1\rightarrow 0}$

$A = 1$
$B = 0 \rightarrow 1$

$t_{pd,1\rightarrow 0}$

$A = 0 \rightarrow 1$
$B = 1$

$t_{pd,0\rightarrow 1}$

$A = 1 \rightarrow 0$
$B = 1 \rightarrow 0$

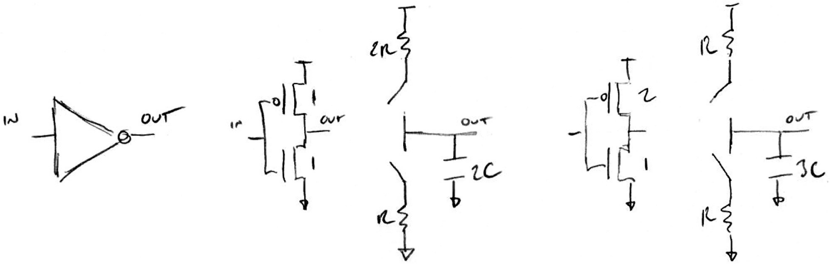
$t_{pd,0\rightarrow 1}$

$A = 1$
$B = 1 \rightarrow 0$

$t_{pd,0\rightarrow 1}$

$A = 1 \rightarrow 0$
$B = 1$

2.3. Equal Rise/Fall Times



$t_{pd,1\to 0}$

in = $0 \to 1$

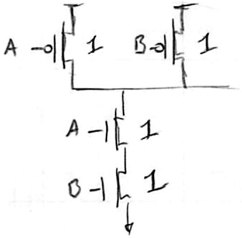
$t_{pd,0\to 1}$

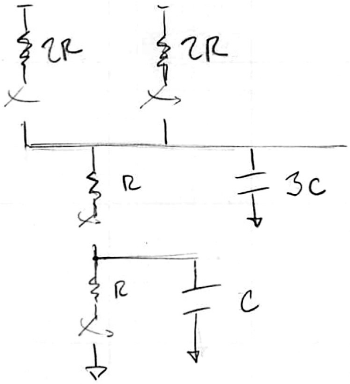
in = $1 \to 0$

- For equal rise/fall times, the effective resistance of pullup must equal effective resistance of pulldown
- If we assume PMOS mobility 2× worse than NMOS, then PMOS must be 2× size of NMOS in an inverter for equal rise/fall times

2.4. Equal Drive Strength

- Size transistors so worst case effective resistance is equal in both the pullup and pulldown networks.

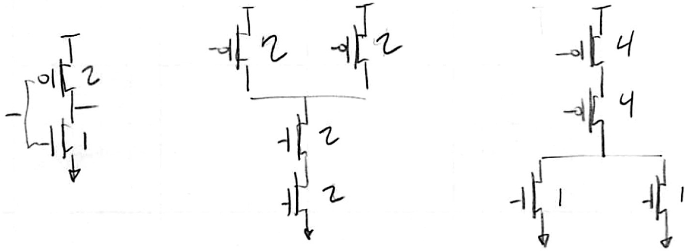




		$t_{pd,1\to 0}$		$t_{pd,0\to 1}$	
		worst	best	worst	best
inverter					
2-input NAND	w/o internal cap				
2-input NAND	w/ internal cap				

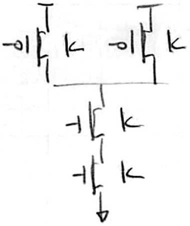
- Is this a fair comparison? No, we are not normalizing anything across these gates. We need to either normalize:
  - Input gate gap (i.e., load on previous gate)
  - Drive strength (i.e., effective resistance)

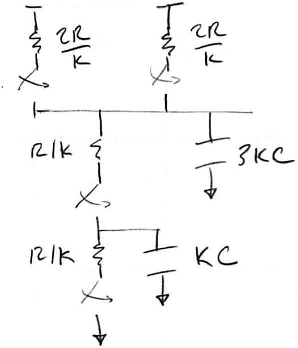
- All three gates with equal rise/fall times *and* equal drive strengths



R_{EFF} =	R	R	R
C_{IN} =	3C	4C	5C
C_P =	3C	6C	6C

2.5. Larger Gates



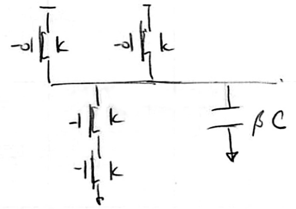


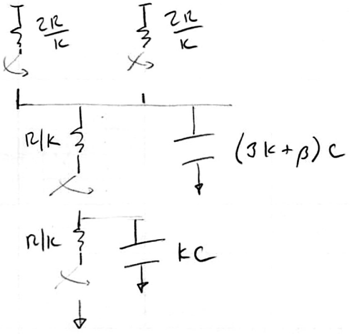
$t_{pd,1\rightarrow 0} =$

$t_{pd,0\rightarrow 1} =$

- This is the parasitic delay, independent of size (k)

2.6. Larger Loads



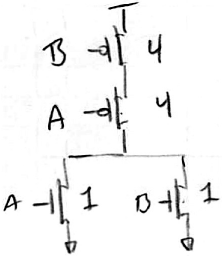


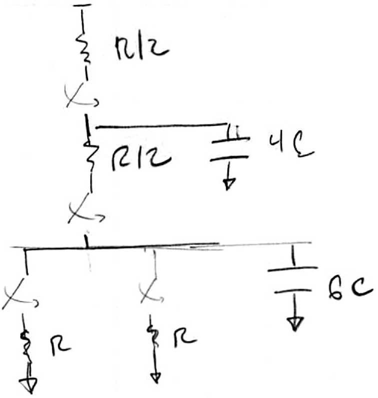
$t_{pd,1\to 0} =$

2.7. Comparison of Inverter, NAND, NOR Gates

- Complete a fair comparison assuming equal rise/fall times, equal drive strength, only parasitic delay

		$t_{pd,1\to 0}$		$t_{pd,0\to 1}$	
		worst	best	worst	best
inverter					
2-input NAND	w/o internal cap				
2-input NAND	w/ internal cap				
2-input NOR	w/o internal cap				
2-input NOR	w/ internal cap				





Use Elmore Delay to Estimate Rise/Fall Times for 2-Input NOR Gate

$t_{pd,1\to 0}$

$A = 0 \to 1$
$B = 0 \to 1$

$t_{pd,1\to 0}$

$A = 0$
$B = 0 \to 1$

$t_{pd,1\to 0}$

$A = 0 \to 1$
$B = 0$

$t_{pd,0\to 1}$

$A = 0$
$B = 1 \to 0$

$t_{pd,0\to 1}$

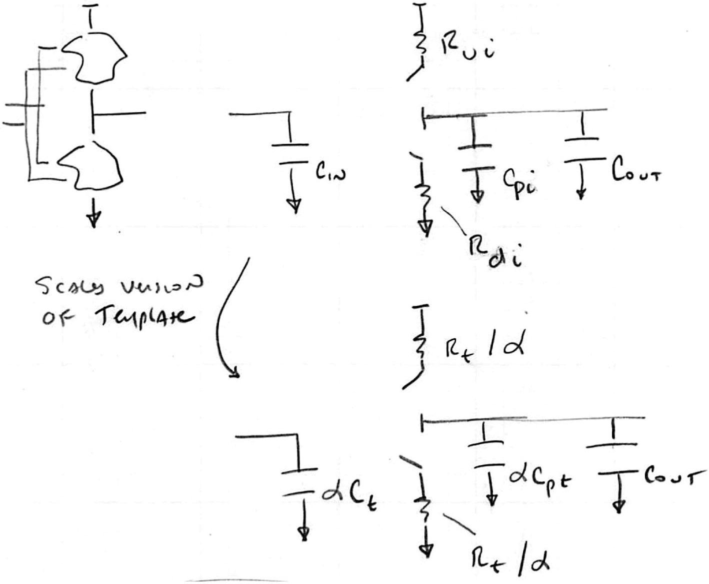
$A = 1 \to 0$
$B = 0$

Use RC Modeling to Estimate Delay of 2-Input NAND and NOR Gates

- Ignore internal capacitance
- Assume worst case delay
- Assume an output load of 15C

2.8. Logical Effort: Single Stage

- Logic effort (LE) is just an abstraction over RC modeling
- Logic effort (LE) is a *linear delay model*
- Useful for building intuition for static CMOS modeling
- Keep in mind often need to use a mix of RC modeling and LE



$$C_{in} = \alpha C_t$$
$$R_i = R_{ui} = R_{di} = R_t/\alpha$$
$$C_{pi} = \alpha C_{pt}$$

- We know the propagation delay of the gate instance is:

$$t_{pd} = R_i(C_{out} + C_{pi})$$

- Let's rewrite this in terms of the template

$$\begin{aligned}
C_{in} &= \alpha C_t \\
R_i &= R_{ui} = R_{di} = R_t/\alpha \\
C_{pi} &= \alpha C_{pt}
\end{aligned}$$

$$\begin{aligned}
t_{pd} &= R_i(C_{out} + C_{pi}) \\
&= R_i C_{out} + R_i C_{pi} \\
&= \frac{R_t}{\alpha} C_{out} + \frac{R_t}{\alpha} C_{pi} \\
&= \frac{R_t}{\alpha} C_{out} + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
&= \frac{R_t}{\alpha} \left(\frac{C_{in}}{C_{in}} \right) C_{out} + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
&= \frac{R_t}{\alpha} \alpha C_t \left(\frac{C_{out}}{C_{in}} \right) + \frac{R_t}{\alpha} (\alpha C_{pt}) \\
&= R_t C_t \left(\frac{C_{out}}{C_{in}} \right) + R_t C_{pt}
\end{aligned}$$

- We don't want to deal with absolute delay
- Let's rewrite our propagation delay equation to be a "relative" delay
- Relative to the delay of a single unloaded minimal inverter
- Let's start by defining some new parameters

$$\tau = R_{inv}C_{inv} \qquad \text{"relative delay units"}$$

$$g = R_tC_t / R_{inv}C_{inv} \qquad \text{logical effort}$$

$$h = C_{out} / C_{in} \qquad \text{electrical effort}$$

$$p = R_tC_{pt} / R_{inv}C_{inv} \qquad \text{parasitic delay}$$

- Let's rewrite our propagation delay equation in terms of τ

$\tau = R_{inv}C_{inv}$ "relative delay units"

$g = R_tC_t / R_{inv}C_{inv}$ logical effort

$h = C_{out} / C_{in}$ electrical effort

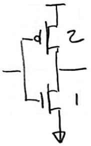
$p = R_tC_{pt} / R_{inv}C_{inv}$ parasitic delay

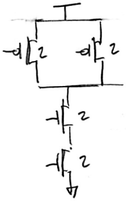
$$\begin{aligned}
t_{pd} = d_{abs} &= R_tC_t\left(\frac{C_{out}}{C_{in}}\right) + R_tC_{pt} \\
&= \left(\frac{R_{inv}C_{inv}}{R_{inv}C_{inv}}\right)R_tC_t\left(\frac{C_{out}}{C_{in}}\right) + \left(\frac{R_{inv}C_{inv}}{R_{inv}C_{inv}}\right)R_tC_{pt} \\
&= R_{inv}C_{inv}\left(\frac{R_tC_t}{R_{inv}C_{inv}}\right)\left(\frac{C_{out}}{C_{in}}\right) + R_{inv}C_{inv}\left(\frac{R_tC_{pt}}{R_{inv}C_{inv}}\right) \\
&= \tau gh + \tau p \\
&= \tau(gh + p)
\end{aligned}$$

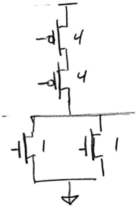
$$\boxed{d_{abs} = \tau(gh + p)}$$

- Let d be the delay in units of τ (i.e., $d = gh + p$)

Templates for Inverter, NAND, NOR Gates

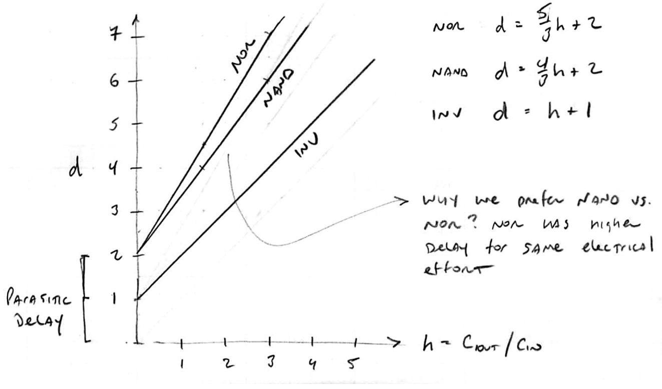






R_{EFF}	R	R	R
C_{IN}	$3C$	$4C$	$5C$
$\frac{R_t C_t}{R_{inv} C_{inv}}$	$\frac{R3C}{R3C}$	$\frac{R4C}{R3C}$	$\frac{R5C}{R3C}$
g	1	$4/3$	$5/3$
$\frac{R_t C_{pt}}{R_{INV} C_{INV}}$	1	$\frac{6}{3} = 2$	$\frac{6}{3} = 2$

RECALL $d = gh + p$ (LINEAR DELAY MODEL)
PLOT d AS A FUNCTION OF h



NOR $d = \frac{5}{3}h + 2$

NAND $d = \frac{4}{3}h + 2$

INV $d = h + 1$

WHY WE PREFER NAND VS. NOR? NOR HAS HIGHER DELAY FOR SAME ELECTRICAL EFFORT

Use LE to Estimate Delay of 2-Input NAND and NOR Gates

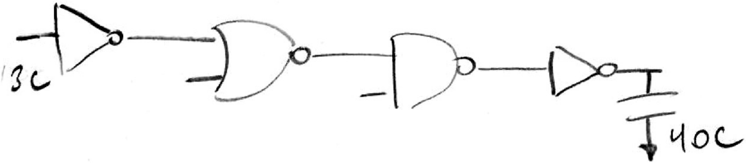
- Assume an output load of 15C

Let's list the many approximations we have made

2.9. Logical Effort: Multiple Stages

- Path delay is the sum of the delay of each stage

$$D = \sum d_i = \sum (g_i h_i + p_i)$$



- Calculate path delay assuming canonical sized gates

g_i	1	5/3	4/3	1
h_i	5/3	4/5	3/4	40/3
$g_i \cdot h_i$	5/3	4/3	1	40/3
p_i	1	2	2	1
d_i	2.67	3.33	3	14.33
D				**23.33**

- Calculate path delay assuming final gate is X16

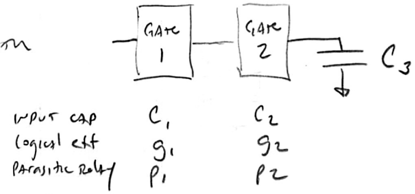
g_i	1	5/3	4/3	1
h_i	5/3	4/5	48/4	40/48
$g_i \cdot h_i$	5/3	4/3	16	40/48
p_i	1	2	2	1
d_i	2.67	3.33	18	1.83
D				**25.83**

Q1: How should we size gates to minimize total delay?

- Independent variables are h_i (i.e., internal gate sizing)
- We want to choose h_i to minimize D
- Take the partial derivative of D with respect to h_i, set to zero, and solve for optimum h_i

Consider two stage path

Assume C_1 and C_3 are given



$$D = (g_1 h_1 + p_1) + (g_2 h_2 + p_2)$$

- Note that h_1 and h_2 are constrained since C_1 and C_3 are given and input cap of gate 2 is output cap for gate 1

$$h_1 = \frac{C_2}{C_1} \qquad h_2 = \frac{C_3}{C_2} \qquad h_1 h_2 = \frac{C_2}{C_1}\frac{C_3}{C_2} = \frac{C_3}{C_1}$$

- Let $H = h_1 h_2 = C_3/C_1$, H is a constant since C_1 and C_3 are given
- Let's rework D to get it in terms of just one variable

$$\begin{aligned}
D &= (g_1 h_1 + p_1) + (g_2 h_2 + p_2) \\
D &= g_1 h_1 + g_2 h_2 + (p_1 + p_2) \\
&= g_1 h_1 + g_2 \frac{H}{h_1} + (p_1 + p_2) \\
&= g_1 h_1 + g_2 H h_1^{-1} + (p_1 + p_2)
\end{aligned}$$

- Take partial derivative with respect to the only variable h_1

$$\begin{aligned}
D &= g_1 h_1 + g_2 H h_1^{-1} + (p_1 + p_2) \\
\frac{\partial D}{\partial h_1} &= g_1 - g_2 H h_1^{-2} + 0 \\
&= g_1 - \frac{g_2 H}{h_1^2}
\end{aligned}$$

- Set partial derivative to zero and solve for h_1

$$\begin{aligned}
\frac{\partial D}{\partial h_1} = g_1 - \frac{g_2 H}{h_1^2} &= 0 \\
g_1 &= \frac{g_2 H}{h_1^2} \\
g_1 h_1^2 &= g_2 H \\
h_1^2 &= \frac{g_2}{g_1} H \\
h_1 &= \sqrt{\frac{g_2}{g_1} H}
\end{aligned}$$

- Can use similar approach to find optimal h_i for more than 2 stages
- However, there is actually a much more interesting result!

$$\begin{aligned}
g_1 h_1^2 &= g_2 H \\
g_1 h_1^2 &= g_2 h_1 h_2 \\
g_1 h_1 &= g_2 h_2 \\
f_1 &= f_2
\end{aligned}$$

- Delay is minimized when stage effort (f_i) is the same in both stages!
- Let $\widehat{f}$ be the optimal stage effort (i.e., $\widehat{f} = f_1 = f_2$)
- We can use a trick to quickly calculate $\widehat{f}$

$$\begin{aligned} \widehat{f} &= \sqrt{\widehat{f}^{\,2}} = \sqrt{\widehat{f}\,\widehat{f}} = \sqrt{f_1\, f_2} \\ &= \sqrt{(g_1 h_1)(g_2 h_2)} \\ &= \sqrt{(g_1 g_2)(h_1 h_2)} \end{aligned}$$

- Let $G = g_1 g_2$, this is the path logical effort
- Let $H = h_1 h_2 = C_{out}/C_{in}$, this is the path electrical effort
- Let $F = GH$, this is the path effort

$$\begin{aligned} \widehat{f} &= \sqrt{(g_1 g_1)(h_1 h_2)} \\ &= \sqrt{GH} \\ &= \sqrt{F} \end{aligned}$$

- We can calculate $\widehat{f}$ without finding the optimal size of each gate!
- Minimal delay with optimal sizing can be quickly calculated using:

$$\widehat{D} = 2\widehat{f} + (p_1 + p_2)$$

- This generalizes to paths with any number of stages

$$G = \prod g_i \qquad \text{path logical effort}$$

$$H = \prod h_i = \frac{C_{out}}{C_{in}} \qquad \text{path electrical effort}$$

$$F = GH \qquad \text{path effort}$$

$$\widehat{f} = F^{1/N} \qquad \text{optimal stage effort}$$

$$P = \sum p_i \qquad \text{path parasitic delay}$$

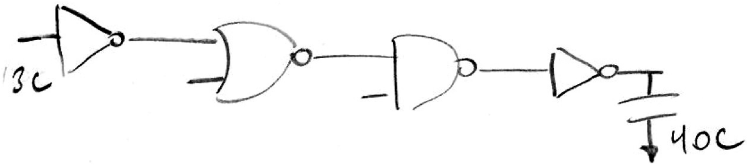
$$\widehat{D} = N\widehat{f} + P \qquad \text{min delay with opt sizing}$$

Method for optimal sizing

1. Calculate path effort ($F = GH$)
2. Calculate effort for each stage ($\widehat{f} = F^{1/N}$)
3. Estimate minimum delay with optimal sizing ($\widehat{D} = N\widehat{f} + P$)
4. Starting with last stage, work backwards sizing each gate

$$\widehat{f} = gh = g\frac{C_{out}}{C_{in}} \qquad C_{in} = \frac{g}{\widehat{f}}C_{out}$$

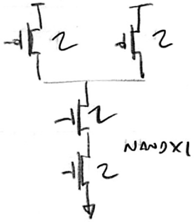
Revisit earlier example

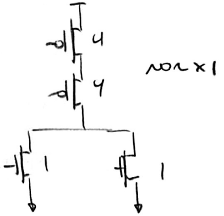


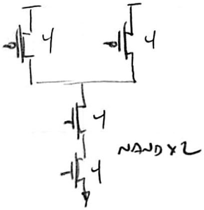
g_i	1	5/3	4/3	1
p_i	1	2	2	1

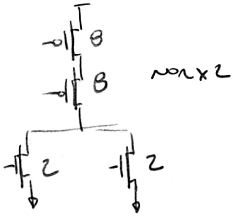
Optimal sizing with standard cells

- This assumes we can size gates arbitrarily using full custom design
- What about if we are using a standard cell library?
- Assume we have a standard cell library with the following cells
  - INVX1, INVX2, INVX4, INVX8
  - NANDX1, NANDX2, NANDX4
  - NORX1, NORX2, NORX4









- Assume we have determined optimal sizing in C_{in}
- How do we figure out which standard cell to use?

- Given optimum C_{in} from before, what is α?

C_{in}	g	$C_{in}/(g \times 3C)$	=	α	gate
17.17C	1	17.17C/(1 × 3C)	=	5.72	INVX4
9.83C	4/3	9.83C/((4/3) × 3C)	=	2.45	NANDX2
7.03C	5/3	7.03C/((5/3) × 3C)	=	1.41	NORX1
3.02C	1	3.02C/(1 × 3C)	=	1.00	INVX1

- Recalculate actual delay given these gates
- First calculate actual C_{in} for each standard cell gate

gate	α	g	$\alpha \times g \times 3C$	=	C_{in}
INVX4	4	1	4 × 1 × 3C	=	12C
NANDX2	2	4/3	2 × 4/3 × 3C	=	8C
NORX1	1	5/3	1 × 5/3 × 3C	=	5C
INVX1	1	1	1 × 1 × 3C	=	3C

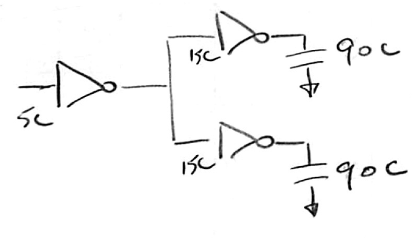
- Now use path delay equation

$$\begin{aligned} D &= \sum gh + \sum p \\ &= (1 \times \frac{40}{12}) + (\frac{4}{3} \times \frac{12}{8}) + (\frac{5}{3} \times \frac{8}{5}) + (1 \times \frac{5}{3}) + (1 + 2 + 2 + 1) \\ &= 3.33 + 2 + 2.67 + 1.67 + 6 = 9.67 + 6 = 15.67 \end{aligned}$$

- Compare with optimal delay which is 15.32, off by 2.3%

What about branching?

- Consider the following simple example



$$G = 1 \times 1 = 1$$

$$H = 90C/5C = 18$$

$$F = GH = 18$$

$$F = \prod g_i h_i$$

$$= (1 \times 6) \times (1 \times 6) = 36$$

- So in this example $F = 2GH$
- The factor of two is called the branching effort
- Key Idea: some drive current is directed off path we are analyzing
- Similar to Elmore delay for trees

$$b = \frac{C_{\text{onpath}} + C_{\text{offpath}}}{C_{\text{onpath}}} \qquad \text{stage branching effort}$$

$$B = \prod b_i \qquad \text{path branching effort}$$

- So our new path effort equation is now:

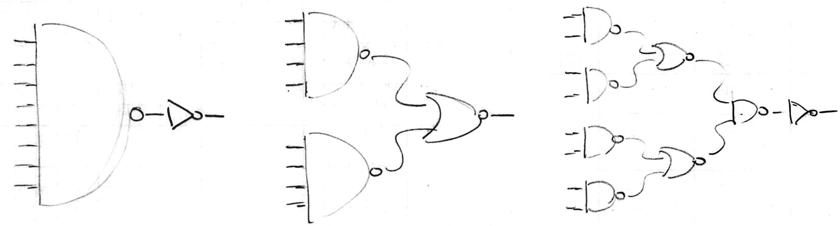
$$F = \prod f_i = GBH$$

- Note that path effort depends on circuit topology and loading of entire path, but *not* size of transistors in network

- Note that path effort does not change if we add or remove inverters!

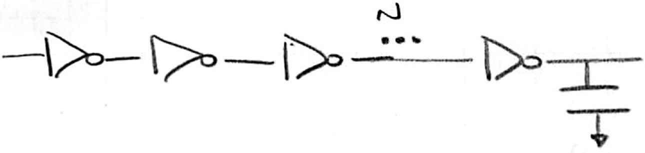
Q2: How should we change topology to minimize delay?

- Assume we want to implement an eight input AND gate
- Calculate min delay assuming optimal sizing for three topologies
- First assume $H = 1$, then assume $H = 12$



	$H = 1$			$H = 12$		
Topology	$NF^{1/N}$	P	$\widehat{D}$	$NF^{1/N}$	P	$\widehat{D}$
NAND8						
NAND4						
NAND2						

Determine optimal number of stages for chain of inverters



$$\begin{aligned}
\widehat{D} &= NF^{1/N} + NP_{inv} \\
\frac{\partial \widehat{D}}{\partial N} &= F^{1/N} - F^{1/N}\ln(F^{1/N}) + P_{inv} = 0
\end{aligned}$$

- If $P_{inv} = 0$

$$\begin{aligned}
\frac{\partial \widehat{D}}{\partial N} &= F^{1/N} - F^{1/N}\ln(F^{1/N}) = 0 \\
\ln(F^{1/N}) &= 1 \\
F^{1/N} &= e \\
\widehat{f} &= e
\end{aligned}$$

- So if we assume $P_{inv} = 0$, then the optimal number of stages results in a stage effort of e (i.e., 2.718) for every stage
- Since $G = 1$ for an inverter, this means $h = 2.718$ for every stage

- If $P_{inv} = 1$, then we need to solve this nonlinear equation:

$$F^{1/N} - F^{1/N} \ln(F^{1/N}) + 1 = 0$$

- Let $\rho = F^{1/\widehat{N}}$ where $\widehat{N}$ is optimal number of stages

$$1 + \rho(1 - \ln(\rho)) = 0$$

- We can solve this numerically to find that $\rho \approx 3.59$
- So if we assume $P_{inv} = 1$, then the optimal number of stages results in a stage effort of 3.59 for every stage
- Since $G = 1$ for an inverter, this means $h = 3.59$ for every stage
- We can roughly approximate 3.59 to be 4
- Let's solve for $\widehat{N}$ as a function of F

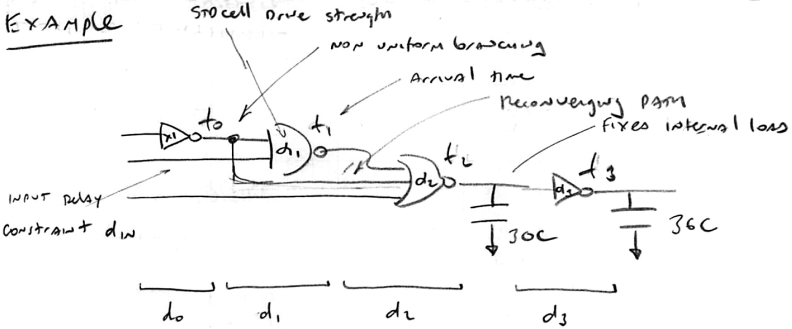
$$\begin{aligned} F^{1/\widehat{N}} &= 4 \\ \log(F^{1/\widehat{N}}) &= \log(4) \\ \frac{1}{\widehat{N}} \log(F) &= \log(4) \\ \widehat{N} &= \frac{\log(F)}{\log(4)} = \log_4(F) \end{aligned}$$

- This is actually a pretty good estimate even for a path of gates which are not inverters!

Logical effort can help give us intuition on how to size gates + choose a topology to minimize delay but it has many limitations.

To deal with more complicated scenarios we can also write the delay equations for each gate in system and minimize the latest arrival time.

Example



Let's write our linear delay equation as a function of α

$d = gh + p$ $\qquad g = \frac{R_T C_T}{R_{inv} C_{inv}}$ $\qquad C_{in} = \alpha C_T$ $\qquad C_T = \frac{C_{in}}{\alpha}$

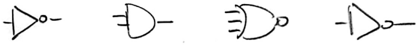
$d = \frac{\cancel{C_{in}}}{\alpha C_{inv}} \cdot \frac{C_{out}}{\cancel{C_{in}}} + p$ $\qquad g = \frac{C_{in}}{\alpha C_{inv}}$

$\sim 3C$

$$\boxed{d = \frac{C_{out}}{3\alpha} + p}$$ Delay as function of α (std cell drive)

Recall $\boxed{C_{in} = 3g\alpha}$

Now write Delay equations for each stage



	d_0	d_1	d_2	d_3
α	1	α_1	α_2	α_3
g	1	$4/3$	$7/3$	1
$C_{IN} = 3g\alpha$	3	$4\alpha_1$	$7\alpha_2$	$3\alpha_3$
p	1	2	3	1

$$d_0 = \frac{(4\alpha_1 + 7\alpha_2)}{3 \cdot 1} + 1 = \frac{4}{3}\alpha_1 + \frac{7}{3}\alpha_2 + 1$$

$$d_1 = \frac{(7\alpha_2)}{3 \cdot \alpha_1} + 2 = \frac{7}{3}\frac{\alpha_2}{\alpha_1} + 2$$

$$d_2 = \frac{(30 + 3\alpha_3)}{3\alpha_2} + 3 = \frac{10}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + 3$$

$$d_3 = \frac{(36)}{3\alpha_3} + 1 = \frac{12}{\alpha_3} + 1$$

Arrival TIMES

$$t_0 = d_0$$

$$t_1 = \max(t_0, d_{IN}) + d_1$$

$$t_2 = \max(t_0, t_1) + d_2$$

$$t_3 = t_2 + d_3$$

$$t_3 = \max(t_0, t_1) + d_2 + d_3$$

$$= \max(d_0, \max(d_0, d_{IN}) + d_1) + d_2 + d_3$$

$$t_3 = \max(d_0, \max(d_0, d_{in}) + d_1) + d_2 + d_3$$

Minimize t_3 subject to above constraints with d_1, d_2 d_3 as the independent variables

Actually in synthesis we really want to minimize area (or energy) subject to constraint on t_3.

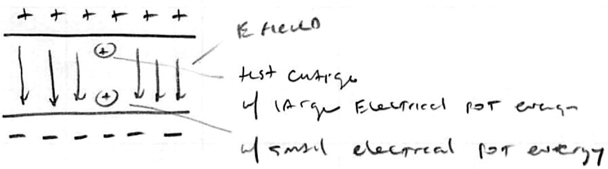
So we could craft optimization problem to be minimize sum of d_1 d_2 d_3 (proxy for area) subject to constraint:

$$t_{\substack{clk \\ period}} > \max(d_0, \max(d_0, d_{in}) + d_{1n}) + d_2 + d_3$$

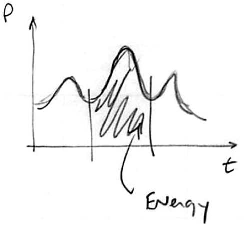
↳ Clock period constraint

3. Energy

- Energy is a measure of work
- Power is the rate at which work is done

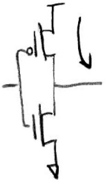


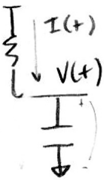
Electric Potential Energy	Capacity for doing work which arises from position of a charge in an electric field	Joules
Electric Potential	Electric potential energy of a position per unit charge	Volts 1V = 1J/C $\Delta V = \Delta E/Q$
Current	Rate at which charge flows past position	Amps 1A = 1C/S $I = Q/\Delta t$
Power	Rate at which electric energy is supplied or consumed	Watts 1W = 1J/S $P = \Delta E/\Delta t = \frac{\Delta V \cdot Q}{Q/I} = VI$



$$E = \int_0^T P(t)\,dt$$

Energy Stored on a Capacitor

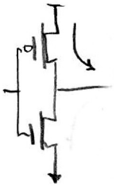


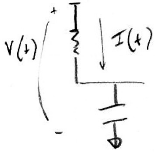


$$E_C = \int_0^\infty P(t)dt = \int_0^\infty V(t)I(t)dt$$
$$= \int_0^\infty V(t)\frac{dQ}{dt}dt = \int_0^\infty V(t)\frac{Cdv}{dt}dt$$
$$= C\int_0^{V_{DD}} V(t)dV = \frac{1}{2}C{V_{DD}}^2$$

- So on 1 → 0 input transition, $\frac{1}{2}C{V_{DD}}^2$ is stored on capacitor
- This energy is released on 0 → 1 input transition

Energy Delivered From Power Supply





$$E_{\text{supply}} = \int_0^\infty P(t)dt = \int_0^\infty V_{DD}I(t)dt$$
$$= V_{DD}\int_0^\infty \frac{dQ}{dt}dt = V_{DD}\int_0^\infty \frac{Cdv}{dt}dt$$
$$= CV_{DD}\int_0^{V_{DD}} dV = C{V_{DD}}^2$$

- 0 → 1 output transition
  - $C{V_{DD}}^2$ energy is delivered from power supply
  - half this energy dissipated as heat in PMOS
  - half this energy is stored on the capacitor
- 1 → 0 output transition
  - no energy is delivered from power supply
  - remaining energy on capacitor dissipated as heat in NMOS

- On average, each bit transition requires $\frac{1}{2}C{V_{DD}}^2$
- Let α be the activity factor, probability of a bit transitions per cycle

$$\boxed{E_{\text{node}} = \alpha\tfrac{1}{2}C{V_{DD}}^2}$$

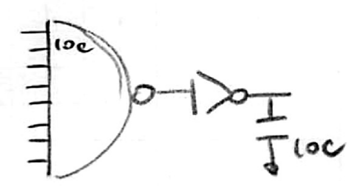
Power Consumption

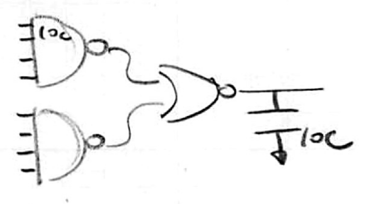
$$\begin{aligned} P_{\text{tot}} &= P_{\text{switching}} + P_{\text{static}} \\ &= \alpha f \frac{1}{2} C{V_{DD}}^2 + V_{DD} I_{\text{off}} \end{aligned}$$

- Sometimes engineers will assume α is the probability of just a $0 \rightarrow 1$ output transition instead of the probability of any transition
  - α = probability of any transition
  - α' = probability of a $0 \rightarrow 1$ transition only
- If you use α' then do not include the factor of $1/2$
- Note that book uses α but it is really α' in our notation!

Comparing Energy

- Calculate the total switched cap in worst case





g	$10/3$	1
p	8	1
G	$10/3$	
H	1	
B	1	
F	$10/3$	
$\hat{f}$	1.8	

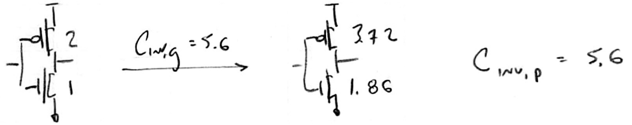
g	2	$5/3$
p	4	2
G	$10/3$	
H	1	
B	1	
F	$10/3$	
$\hat{f}$	1.8	

$$\begin{aligned} C_{\text{inv},g} &= \frac{1}{1.8} \times 10 = 5.6 \\ C_{\text{nand},g} &= \frac{10/3}{1.8} \times 5.6 = 10.4 \\ C_{\text{tot},g} &= C_{\text{inv},g} + 8C_{\text{nand},g} \\ &= 88.8C \end{aligned}$$

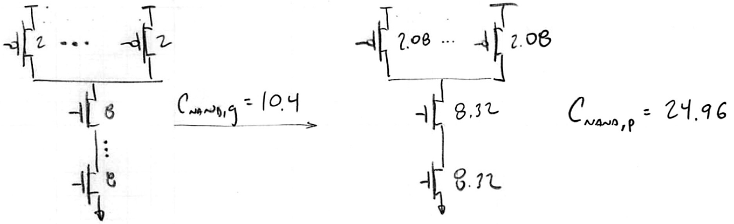
$$\begin{aligned} C_{\text{nor},g} &= \frac{5/3}{1.8} \times 10 = 9.3 \\ C_{\text{nand},g} &= \frac{2}{1.8} \times 9.3 = 10.3 \\ C_{\text{tot},g} &= 2C_{\text{nor},g} + 8C_{\text{nand},g} \\ &= 101C \end{aligned}$$

- To determine parasitic cap need to understand how gate cap is distributed across transistors

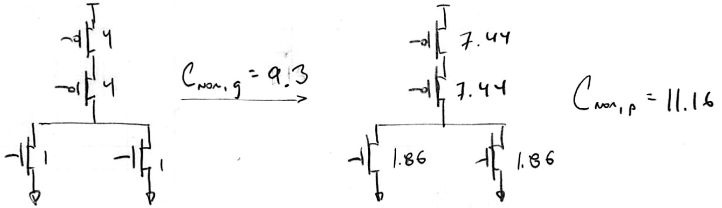
inverter



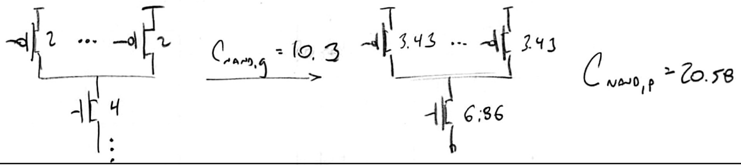
8 INPUT NAND GATE



2 INPUT NOR GATE



4INPUT NAND GATE



$$E_{\text{node}} = \alpha \tfrac{1}{2} C {V_{DD}}^2$$

- Assume $\alpha = 0.1$ and $V_{DD} = 1\text{V}$ for both
- Only difference is amount of switched cap

- For 8-input NAND topology

$$C_{\text{tot}} = C_{\text{tot},g} + C_{\text{tot},p} = 88.8 + (5.6 + 24.96) = 119.36C$$

- For 4-input NAND topology

$$C_{\text{tot}} = C_{\text{tot},g} + C_{\text{tot},p} = 101 + (11.16 + 2 \times 20.58) = 153.32C$$

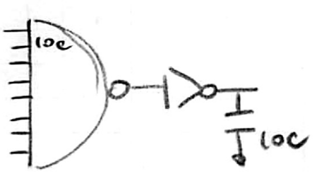
- So second topology requires ≈30% more energy in the *worst* case
- Worst case is when all capacitance is switched
- This ignores the energy for switching the output load
- Let's assume $C = 0.5$ fF (see extra notes)
- Assume clock frequency is 500 MHz

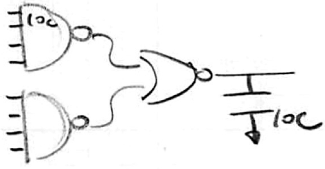
$$E = \alpha \frac{1}{2} C {V_{DD}}^2 = 0.1 \times \frac{1}{2} \times 120C \times \frac{0.5 fF}{C} \times (1V)^2 = 3\text{fJ}$$

$$P = \alpha f \frac{1}{2} C {V_{DD}}^2 = (0.5 \times 10^9)(30 \times 10^{-15}) = 1.5\mu\text{W}$$

Activity Factors

- Previous example used fixed $\alpha = 0.1$ for all nodes
- Can improve accuracy by:
  - Propagate activity factor of inputs to internal nodes
  - Use RTL to calculate activity of inputs, then propagate
  - Use gate-level simulation to find activity of each node





$$P_i = \text{probability node is one on cycle } i$$

$$\overline{P}_i = 1 - P_i = \text{probability node is zero on cycle } i$$

$$\alpha = P_{i-1}\overline{P}_i + \overline{P}_{i-1}P_i$$

$$\alpha' = \overline{P}_{i-1}P_i$$

- Assuming inputs have uncorrelated random data
- Each of these is equally likely: 0→0, 0→1, 1→0, 1→1

$$\alpha = P_{i-1}\overline{P}_i + \overline{P}_{i-1}P_i = 0.5$$

$$\alpha' = \overline{P}_{i-1}P_i = 0.25$$

$$\alpha' = \frac{1}{2}\alpha$$

Output Activity Factor of NAND2

- Calculate output activity factor of a NAND2 gate
- Assume inputs are uncorrelated random data
- Output of NAND2 is zero if both inputs one, otherwise output is one

$$\begin{aligned}\alpha'_{out} &= \overline{P}_{out,i-1} P_{out,i} \\ &= (P_A P_B)(1 - P_A P_B) \\ &= (0.5 \times 0.5)(1 - 0.5 \times 0.5) \\ &= (0.25)(1 - 0.25) \\ &= 0.1875\end{aligned}$$

Output Activity Factor of NAND8

4. Area

- Sum the transistor widths across all transistors in design
- Use standard cell footprints