EECS 151/251A Spring 2018

Digital Design and Integrated Circuits

Instructors:
Nick Weaver & John Wawrzynek

Lecture 9

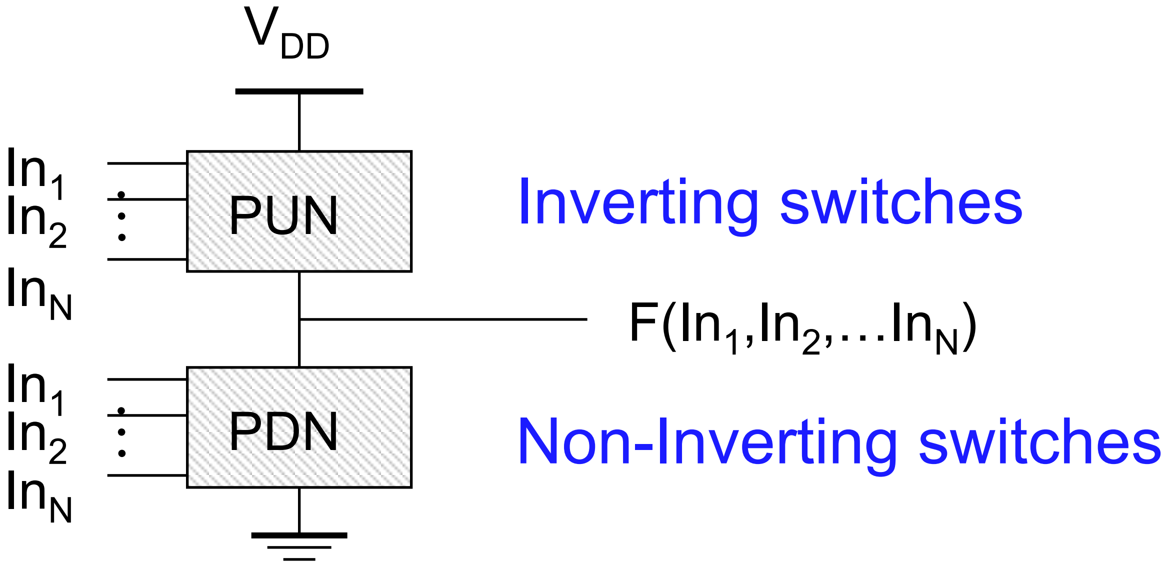
Administration

- First Midterm Thursday

- *Decided on closed book (sorry), but will give you extra time. We will make an exam that is expected to take 90 minutes, but will give you 3 hours.*
  - 5-8PM
  - 405 Soda
  - Material: Everything up to slide 35 of Thursdays lecture (Lecture 8) – that is, CMOS logic is included.

Previous Lecture

Static Complementary CMOS

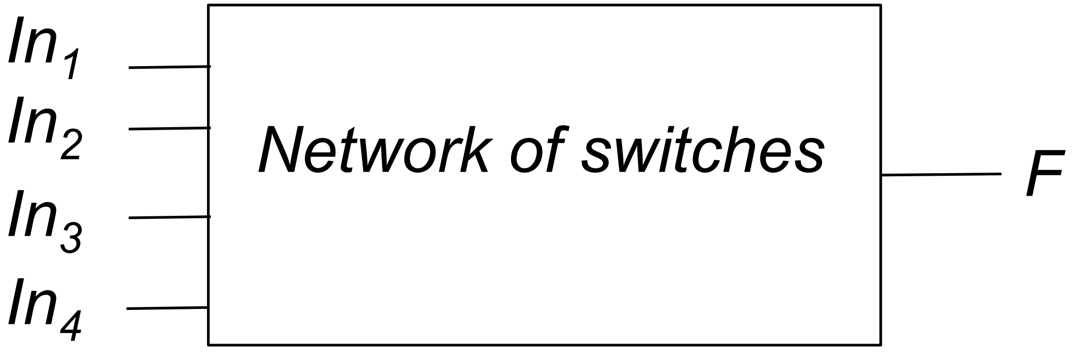


PUN and PDN are dual logic networks
PUN and PDN functions are complementary

- ❑ Full rail-to-rail swing
- ❑ Symmetrical VTC
- ❑ No (…) static power dissipation
- ❑ Direct path current during switching

Switch (Transmission Gate Logic)

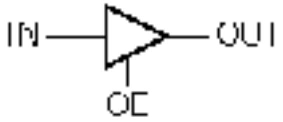
Static:
Output always defined by GND or VDD, never both



No connections to GND or VDD

Tri-state Buffers

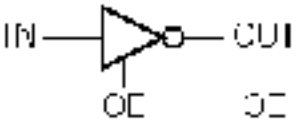
Tri-state Buffer:



OE	IN	OUT
0	0	Z
0	1	Z
1	0	0
1	1	1

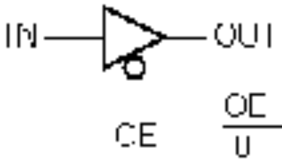
"high impedance" (output disconnected)

Variations:



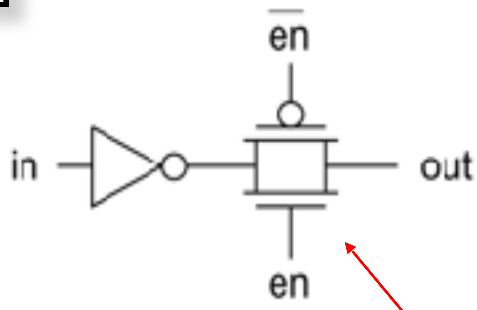
OE	IN	OUT
0	–	Z
1	0	1
1	1	0

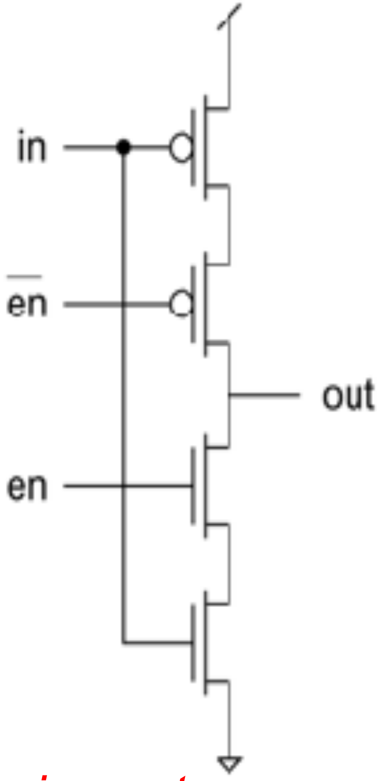
Inverting buffer



OE	IN	OUT
0	0	0
0	1	1
1	–	Z

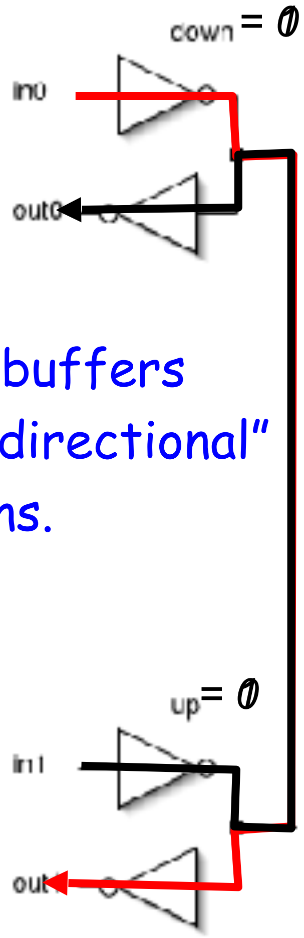
Inverted enable





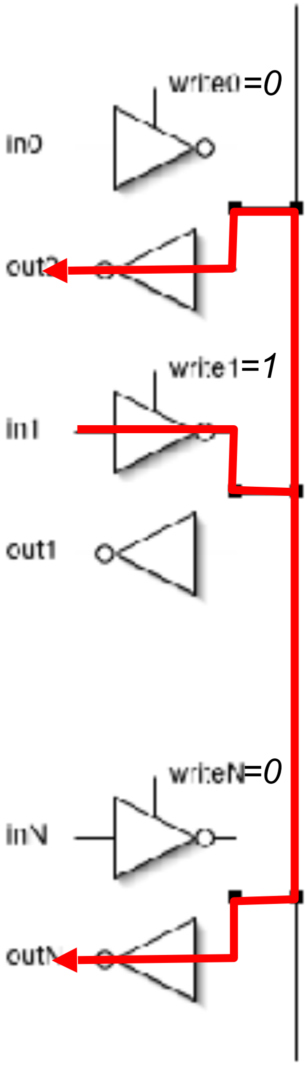
transmission gate useful in implementation

Tri-state Buffers



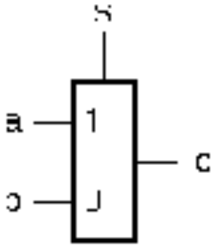
Tri-state buffers enable “bidirectional” connections.

Tri-state buffers are used when multiple circuits all connect to a common wire. Only one circuit at a time is allowed to drive the bus. All others “disconnect” their outputs, but can “listen”.

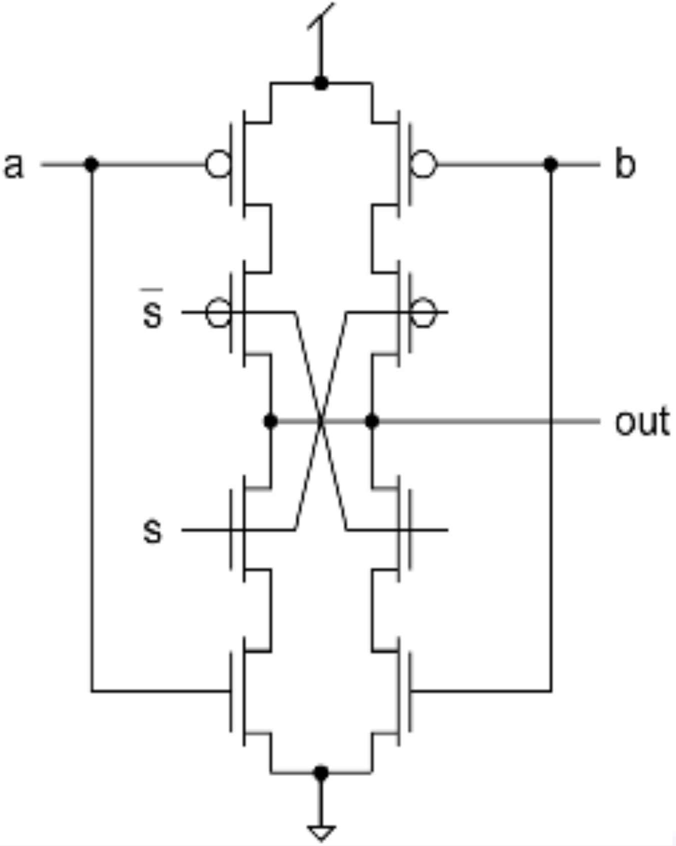


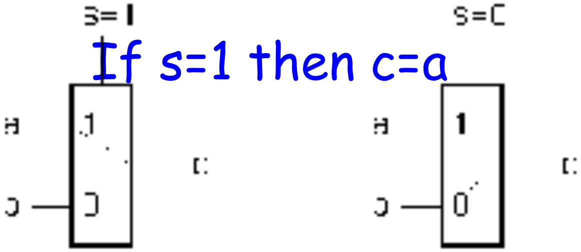
Tri-state Based Multiplexor

Multiplexor



Transistor Circuit for inverting multiplexor:

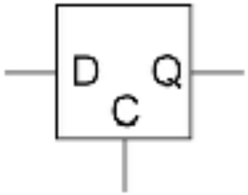


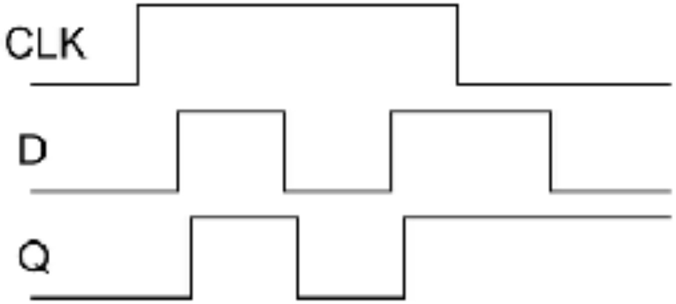


If s=1 then c=a

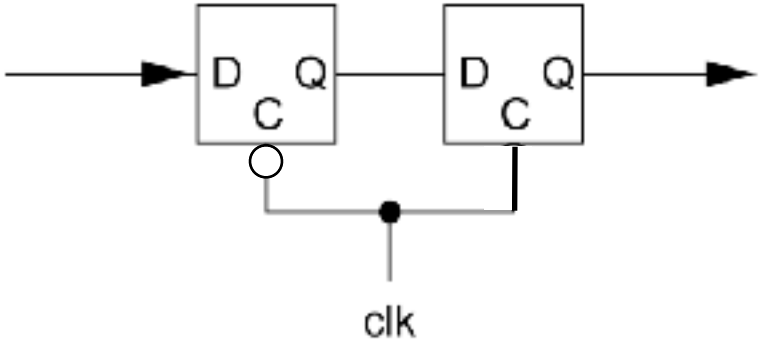
Latches and Flip-flops

Positive Level-sensitive ***latch***:

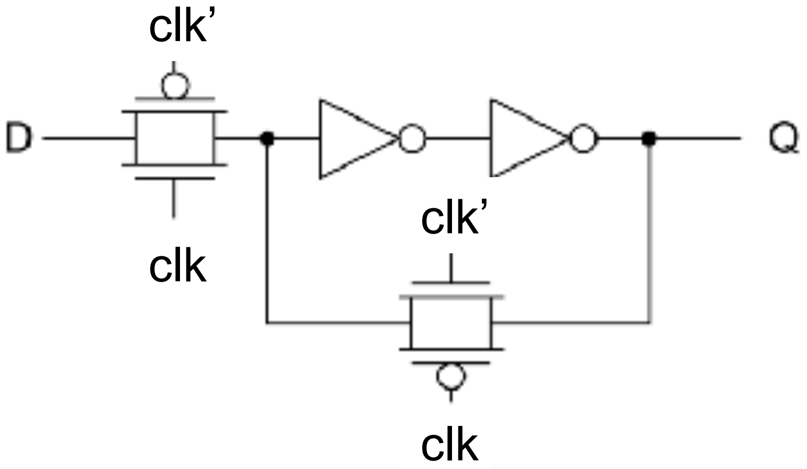




Positive Edge-triggered **flip-flop** built from two level-sensitive latches:



Latch Implementation:



Digital abstraction

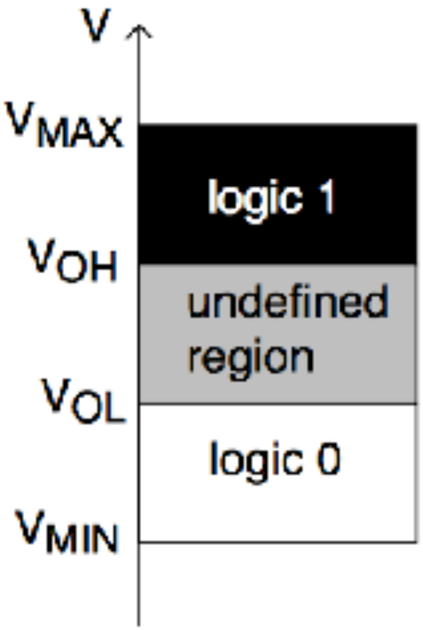
Noise and Digital Systems

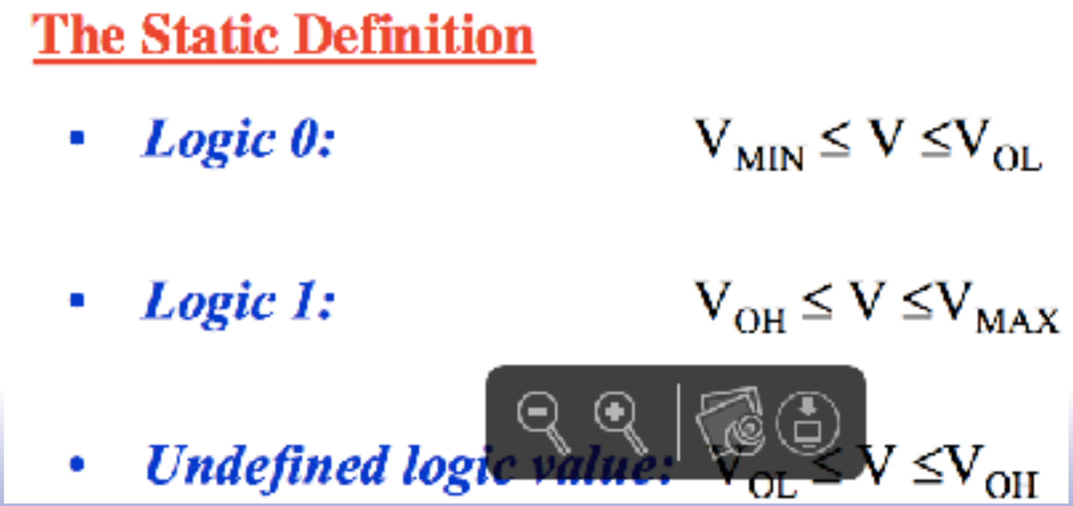
- Circuit needs to works despite “analog” noise
  - Digital gates can and must reject noise
  - This is actually how digital systems are defined

- Digital system is one where:
  - Discrete values mapped to analog levels and back
  - Elements (gates) can reject noise
    - For “small” amounts of noise, output noise is less than input noise
  - Thus, for sufficiently “small” noise, the system acts as if it was noiseless
  - This is called regeneration

Bridging the digital and the analog worlds

- ❑ How to represent 0’s and 1’s in a world that is analog?





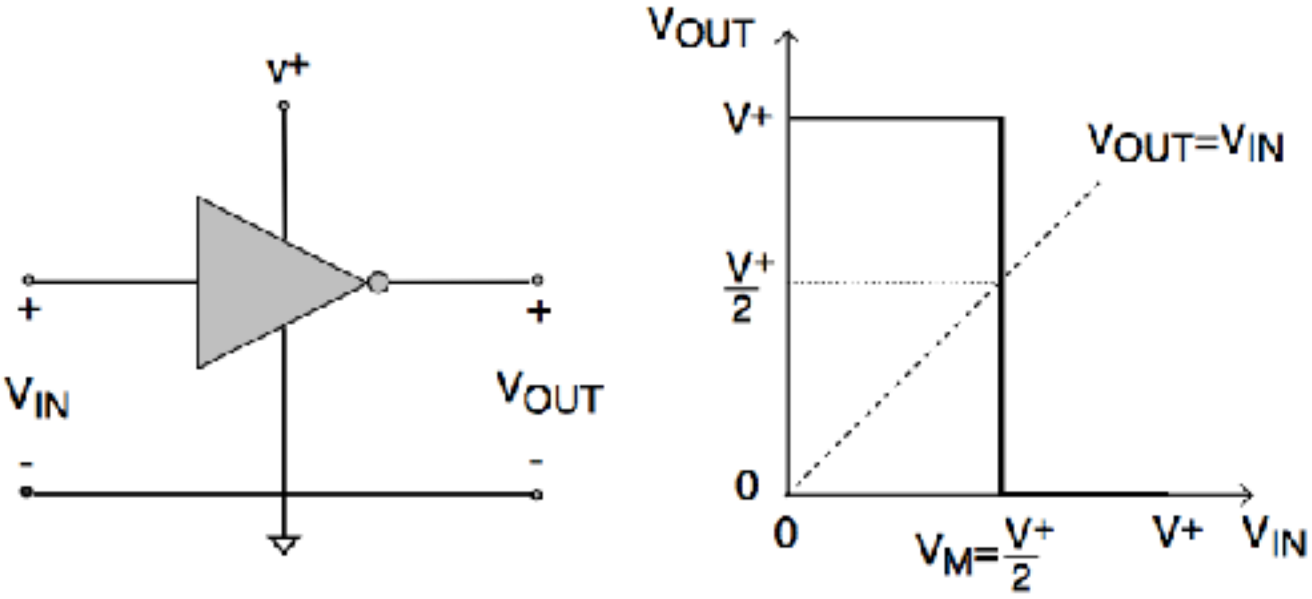
The Static Definition

- *Logic 0:* $V_{MIN} \leq V \leq V_{OL}$
- *Logic 1:* $V_{OH} \leq V \leq V_{MAX}$
- *Undefined logic value:* $V_{OL} \leq V \leq V_{OH}$

From MIT 6.012 Spring 2007, Lecture 11

Ideal Inverter

Circuit representation and ideal transfer function:



Define ***switching point*** or ***logic threshold*** :

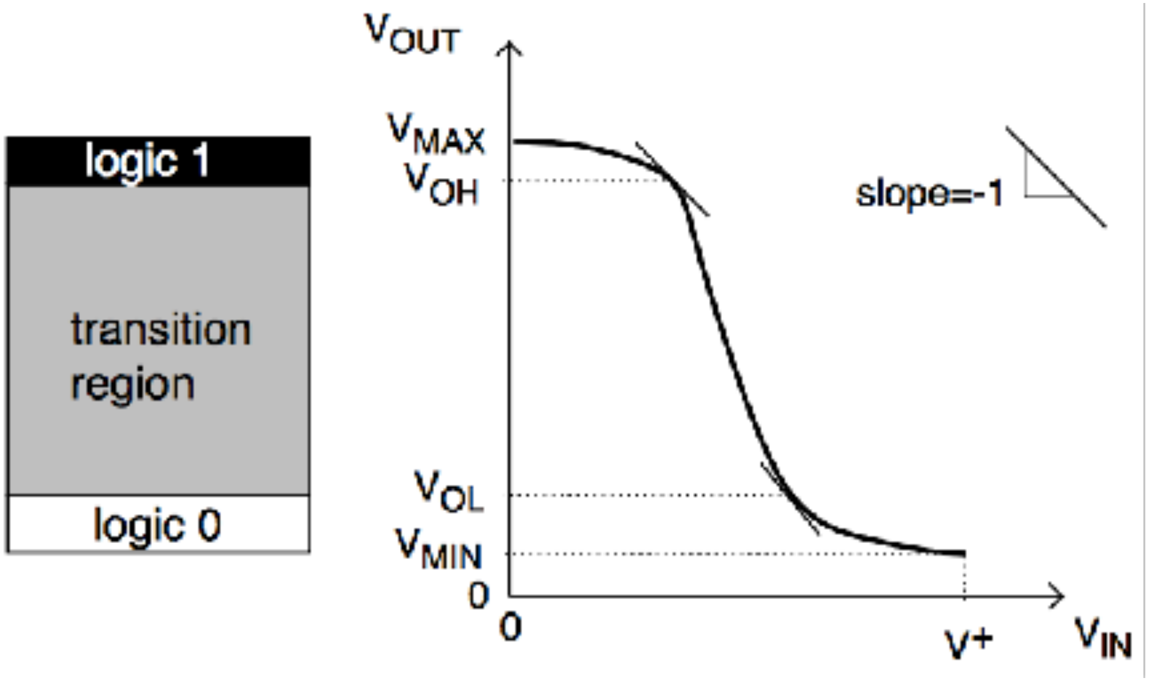
- $V_M \equiv$ input voltage for which $V_{OUT} = V_{IN}$
  - For $0 \le V_{IN} < V_M \quad \Rightarrow V_{OUT} = V^+$
  - For $V_M < V_{IN} \le V^+ \quad \Rightarrow V_{OUT} = 0$

Ideal inverter returns well defined logical outputs (0 or V^+) even in the presence of considerable noise in V_{IN} (from voltage spikes, crosstalk, etc.)

⇒ signal is ***regenerated***!

From MIT 6.012 Spring 2007, Lecture 11

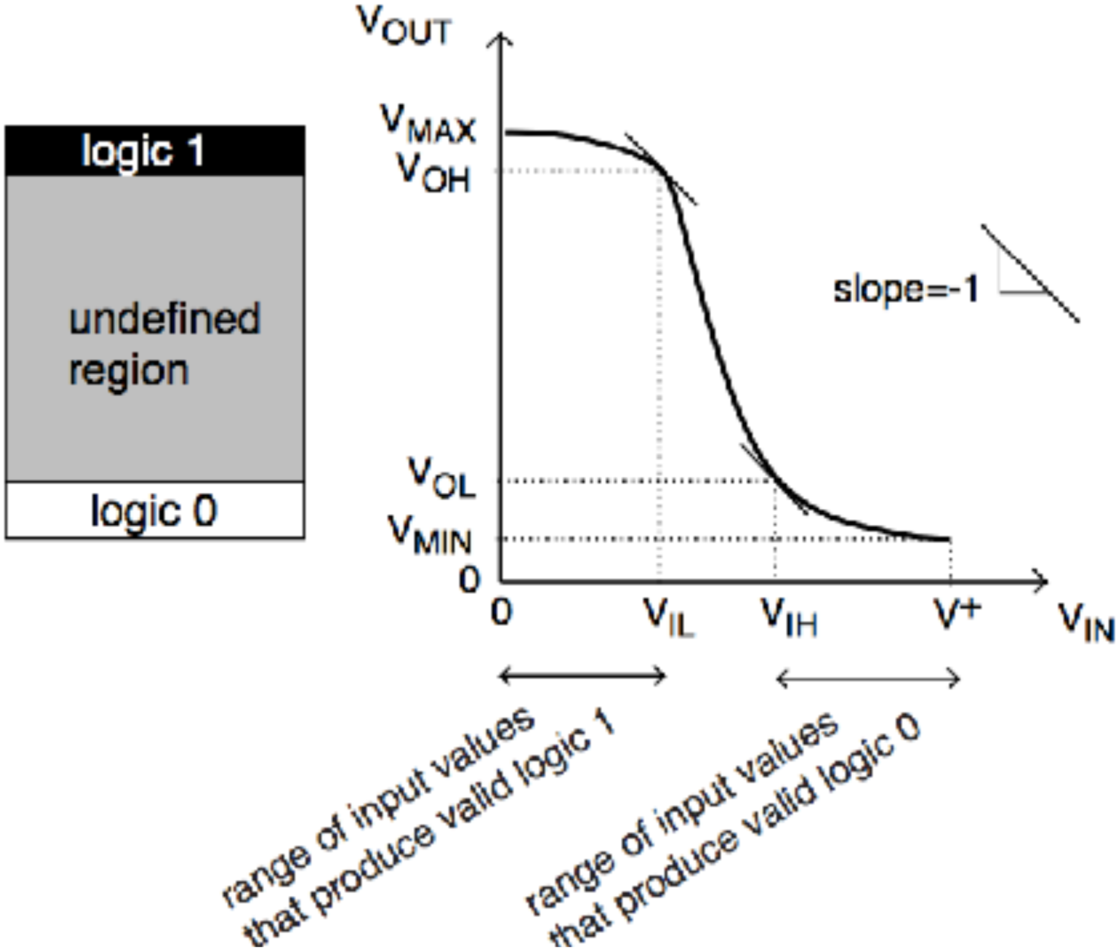
“Real Inverter”



- Logic 0:
  - $V_{MIN} \equiv$ output voltage for which $V_{IN} = V^+$
  - $V_{OL} \equiv$ smallest output voltage where slope = -1
- Logic 1:
  - $V_{OH} \equiv$ largest output voltage where slope = -1
  - $V_{MAX} \equiv$ output voltage for which $V_{IN} = 0$

From MIT 6.012 Spring 2007, Lecture 11

Valid Input Ranges

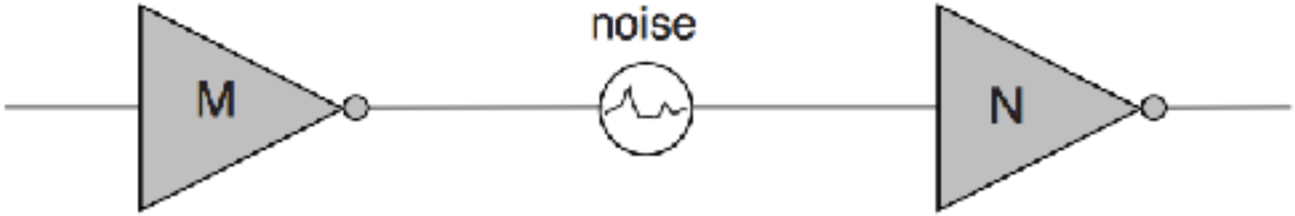


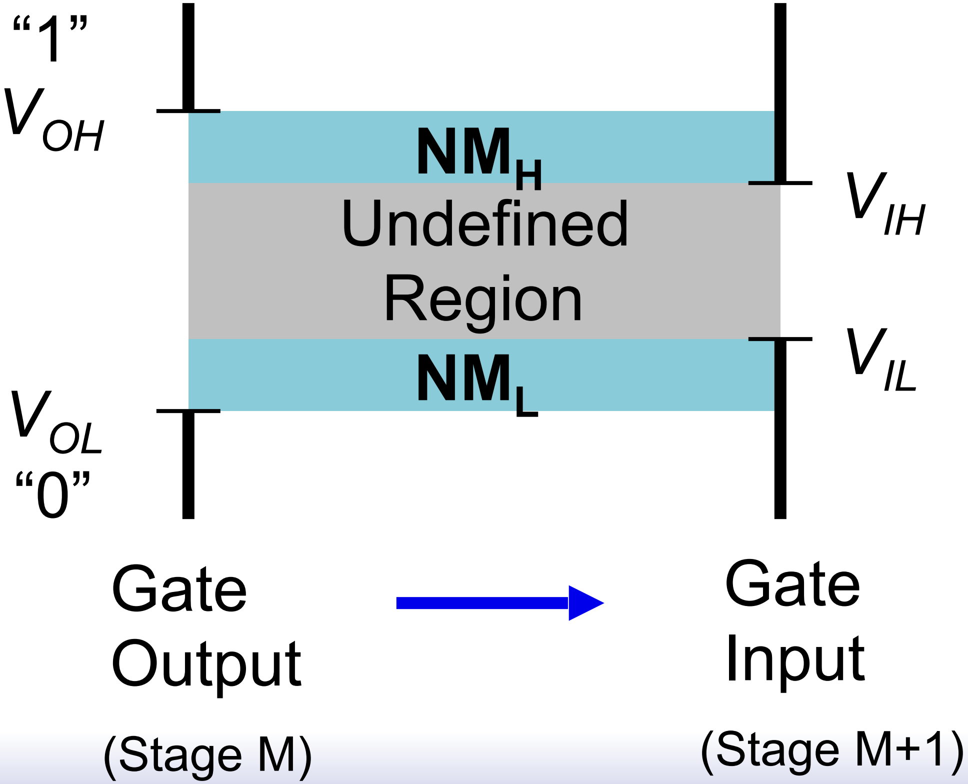
If range of output values V_{OL} to V_{OH} is ***wider*** than the range of input values V_{IL} to V_{IH}, then the inverter exhibits some noise immunity. ($|\text{Voltage gain}| > 1$)

Quantify this through ***noise margins***.

From MIT 6.012 Spring 2007, Lecture 11

Definition of Noise Margins





"1"

V_{OH}

$\mathbf{NM_H}$

Undefined Region

$\mathbf{NM_L}$

V_{OL}

"0"

V_{IH}

V_{IL}

Gate Output

(Stage M)

Gate Input

(Stage M+1)

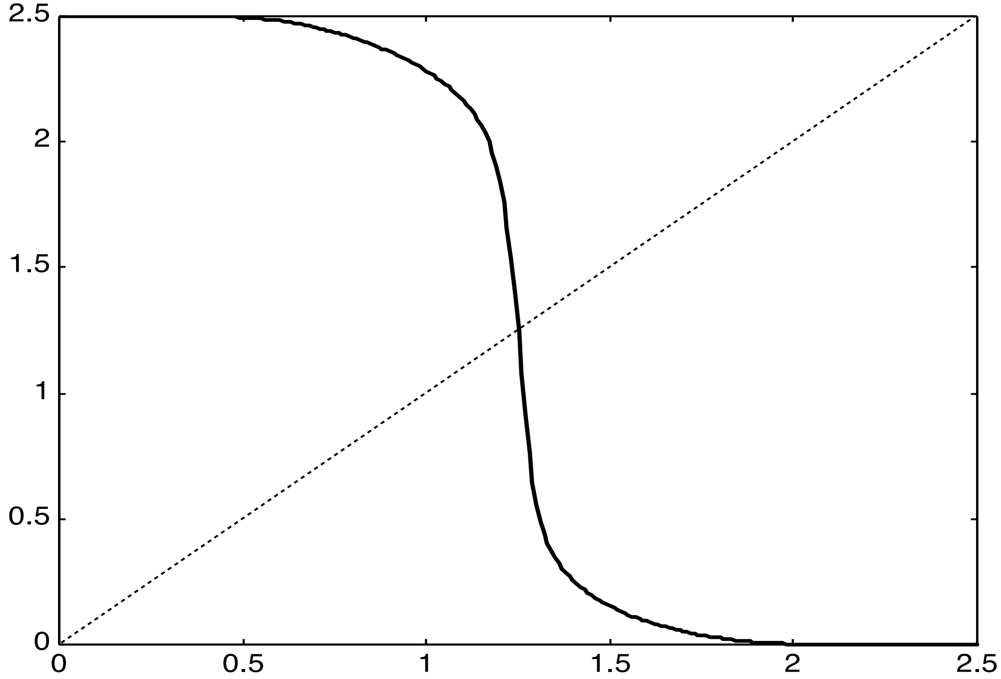
Noise margin high:

$NM_H = V_{OH} – V_{IH}$

Noise margin low:

$NM_L = V_{IL} – V_{OL}$

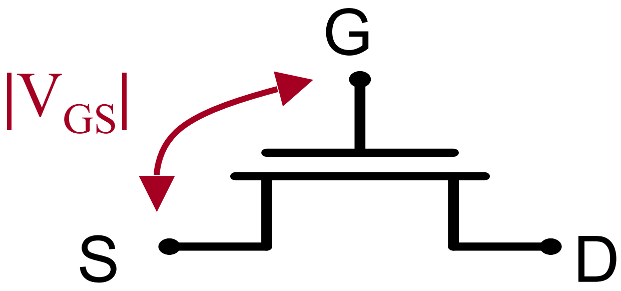
Simulated Inverter VTC (Spice)

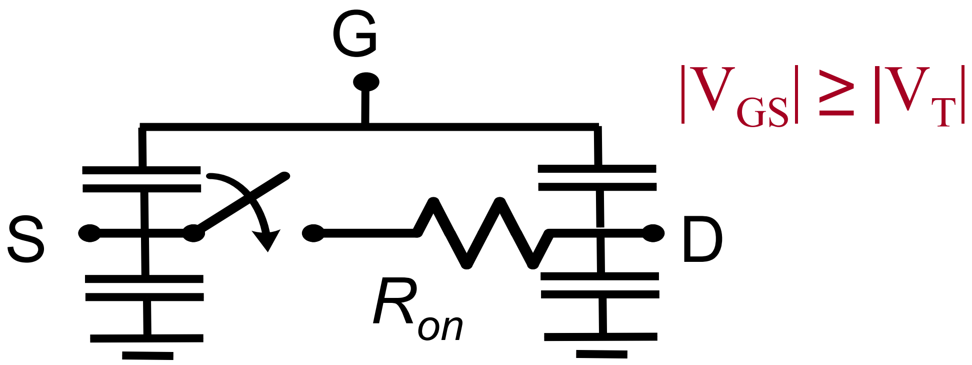


- ❑ V_{OH} =
- ❑ V_{OL} =
- ❑ V_{IL} =
- ❑ V_{IH} =
- ❑ N_{MH} =
- ❑ N_{ML} =
- ❑ V_{M} =

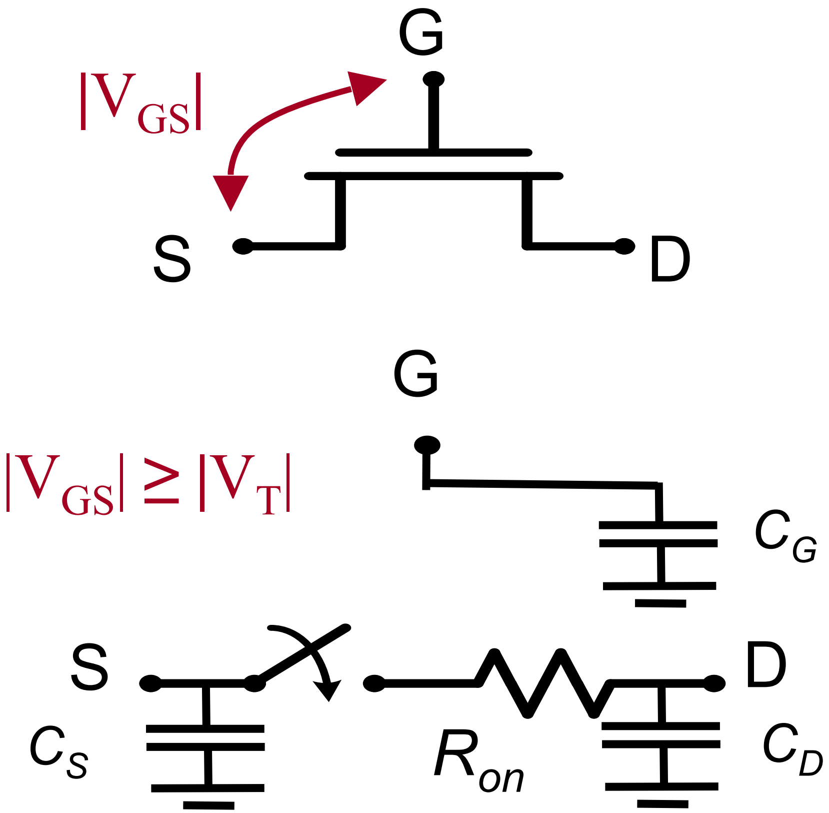
Transient properties

The Switch – Dynamic Model





The Switch – Dynamic Model (Simplified)

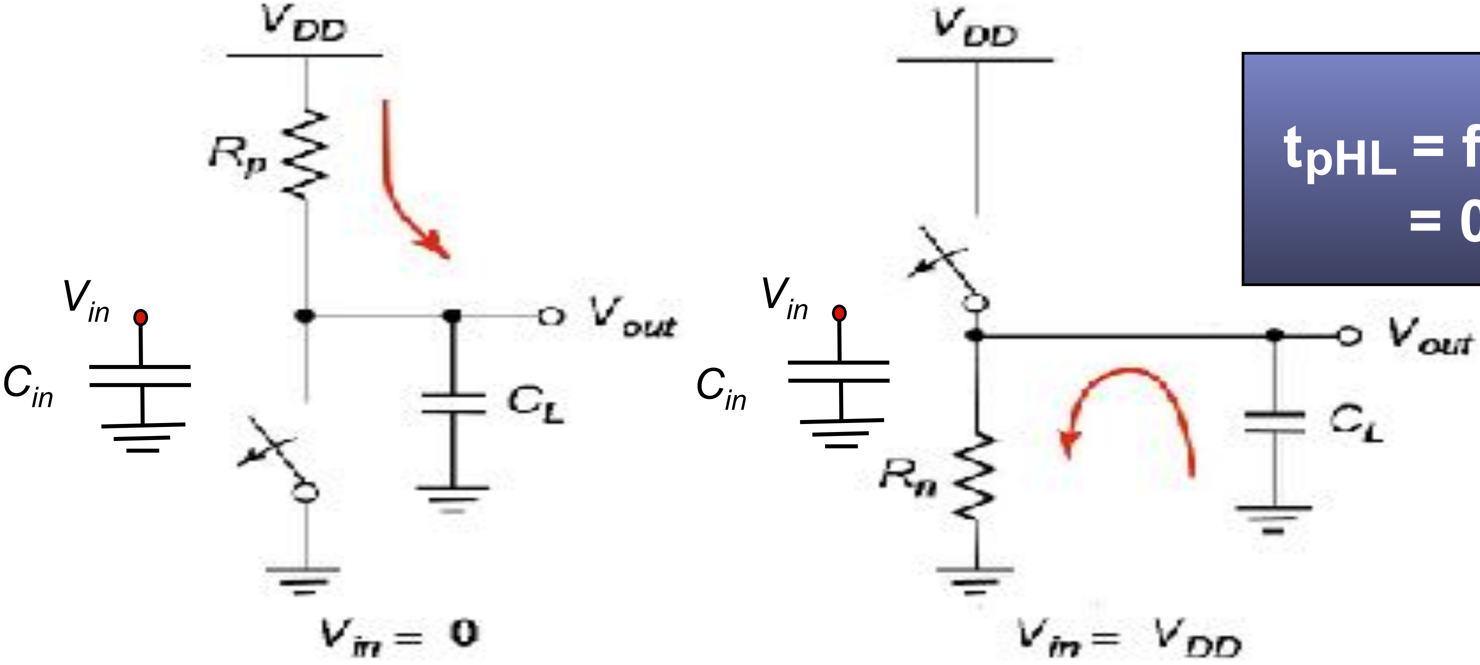


The Switch Inverter: Transient Response

$$V(t) = V_0\, e^{-t/RC}$$

$$t_{1/2} = \ln(2) \times RC$$

$t_{pHL} = f(R_{on}C_L)$
$= 0.69\, R_n C_L$

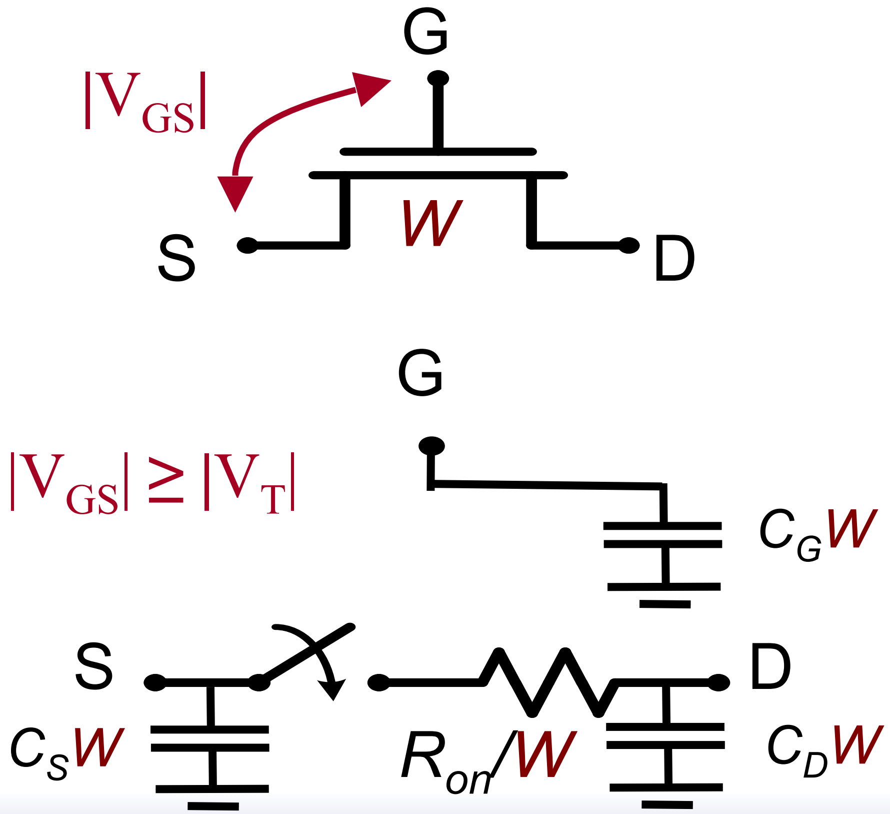


(a) Low-to-high

(b) High-to-low

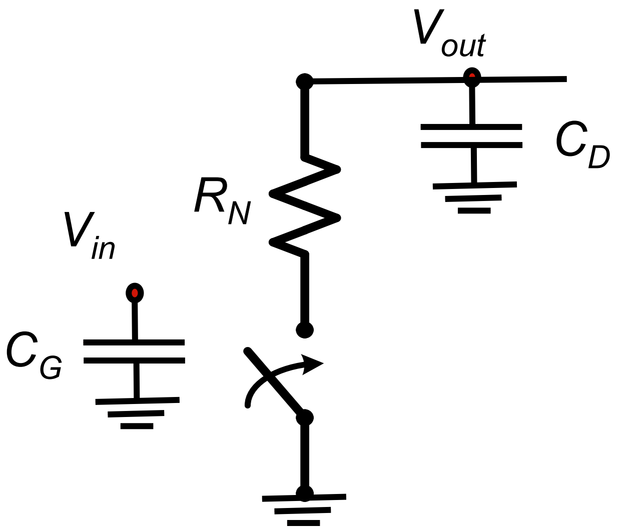
Switch Sizing

What happens if we make a switch W times larger (wider)

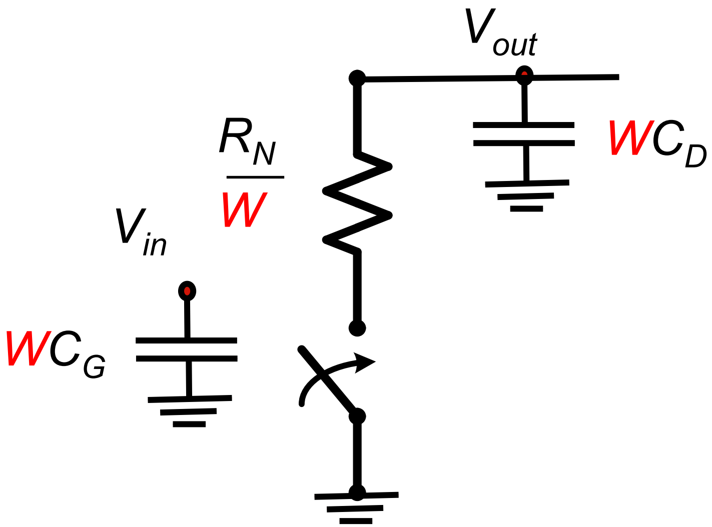


Switch Parasitic Model

The pull-down switch (NMOS)



Minimum-size switch



Sizing the transistor (factor *W*)

We assume transistors of minimal length (or at least constant length). R's and C's in units of per unit width.

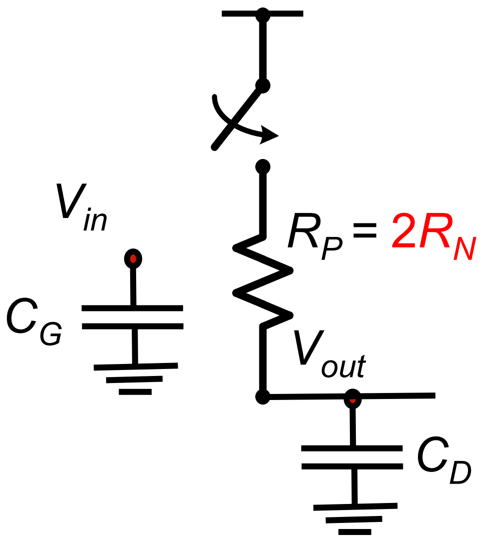
PMOS Sizing

The PMOS challenge:

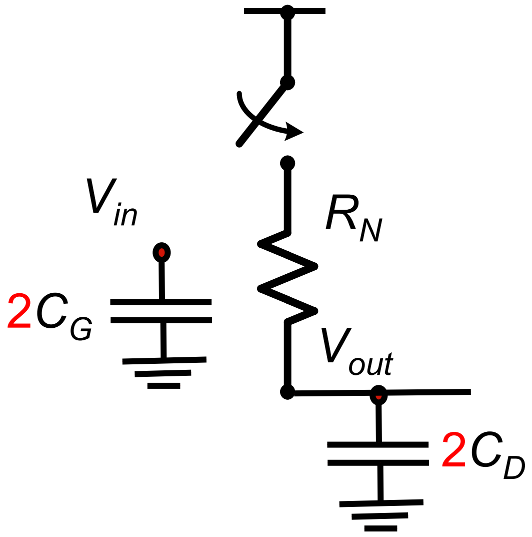
For the same voltages, it provides less current (approximately 2 times less)

Switch Parasitic Model

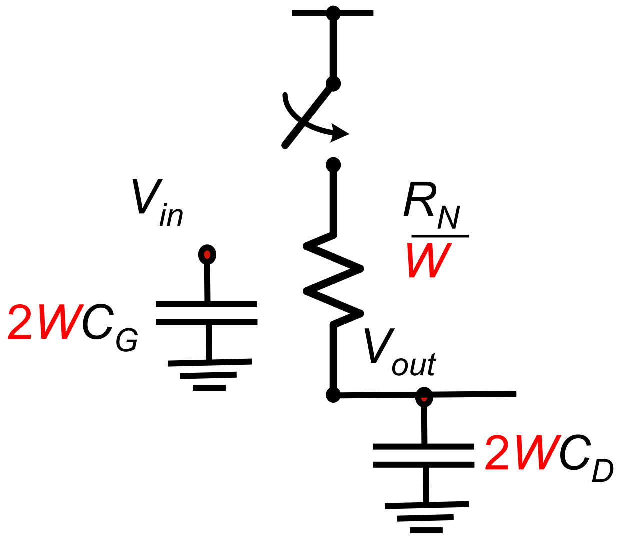
The pull-up switch (PMOS)



Minimum-size switch

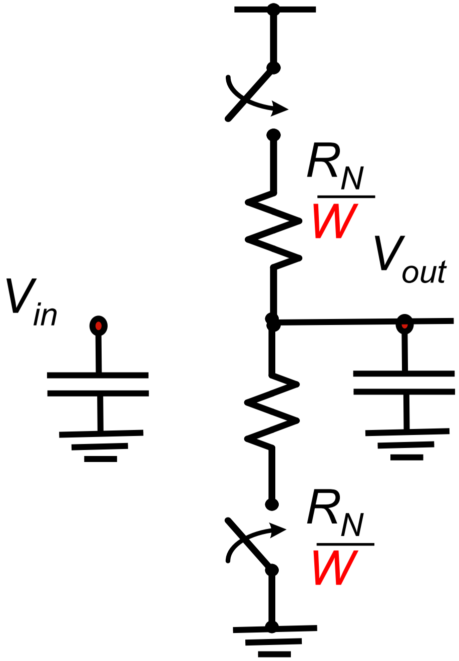


Sized for symmetry



General sizing

Inverter Parasitic Model



$C_{in} = 3WC_G$

$C_{int} = 3WC_D = 3W\gamma C_G$

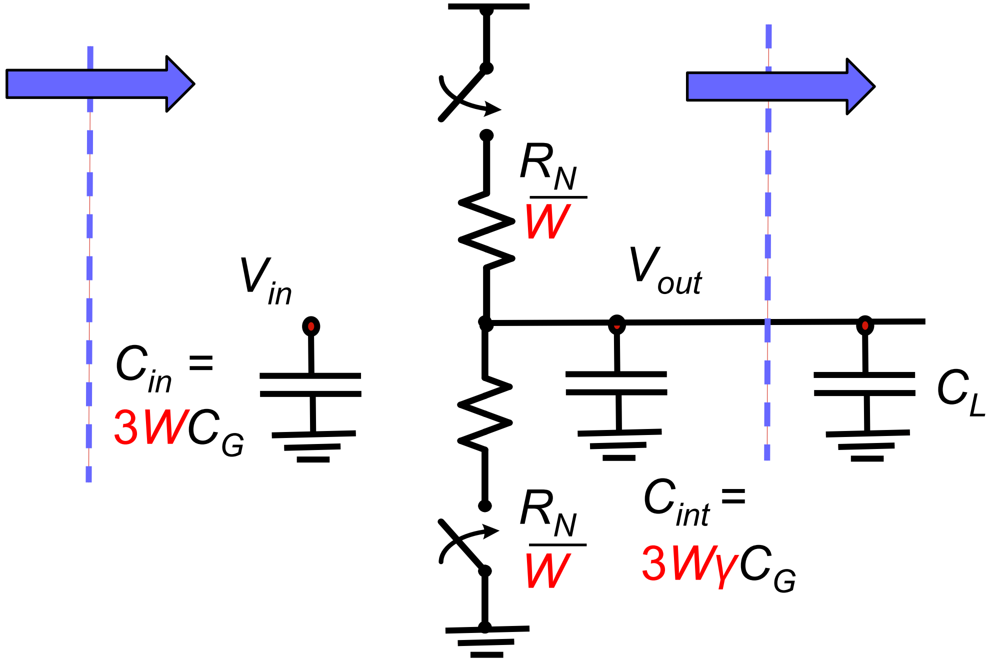
Drain and gate capacitance of transistor are **directly** related by process (γ≈1)

$$\boxed{C_D = \gamma C_G}$$

$$\boxed{t_p = 0.69\left(\frac{R_N}{W}\right)(3W\gamma C_G) = 0.69(3\gamma)R_N C_G}$$

Intrinsic delay of inverter
independent of size

Inverter with Load Capacitance



$$\begin{aligned} t_p &= 0.69\left(\frac{R_N}{W}\right)(C_{\text{int}} + C_L) \\ &= 0.69\left(\frac{R_N}{W}\right)(3W\gamma C_G + C_L) \\ &= 0.69(3C_G R_N)(\gamma + \frac{C_L}{C_{in}}) \\ &= t_{inv}(\gamma + \frac{C_L}{C_{in}}) = t_0(\gamma + f) \end{aligned}$$

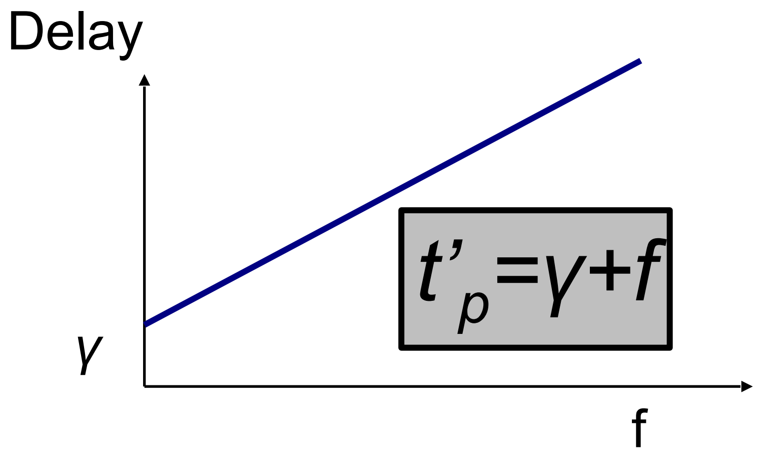
f = **fanout** = ratio between load and input capacitance of gate

Inverter Delay Model

$$\boxed{t_p=t_{inv}(\gamma+f)}$$

t_{inv} technology constant

- Can be dropped from expression
- Delay unit-less variable (expressed in unit delays)

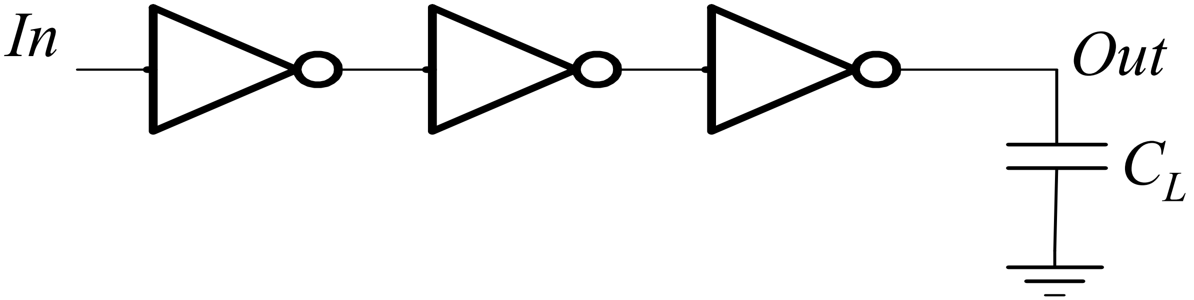


$$\boxed{t'_p=\gamma+f}$$

Question: how does transistor sizing (W) impact delay?

Inverter Delay Optimization

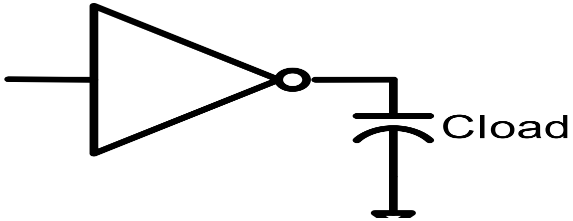
Inverter Chain



- For some given C_L:
  - How many stages are needed to minimize delay?
  - How to size the inverters?
- Anyone want to guess the solution?

Careful about Optimization Problems

- Get fastest delay if build one **very** big inverter
  - So big that delay is set only by self-loading



- Likely not the problem you're interested in
  - Someone has to drive this inverter…

Engineering Optimization Problems in General

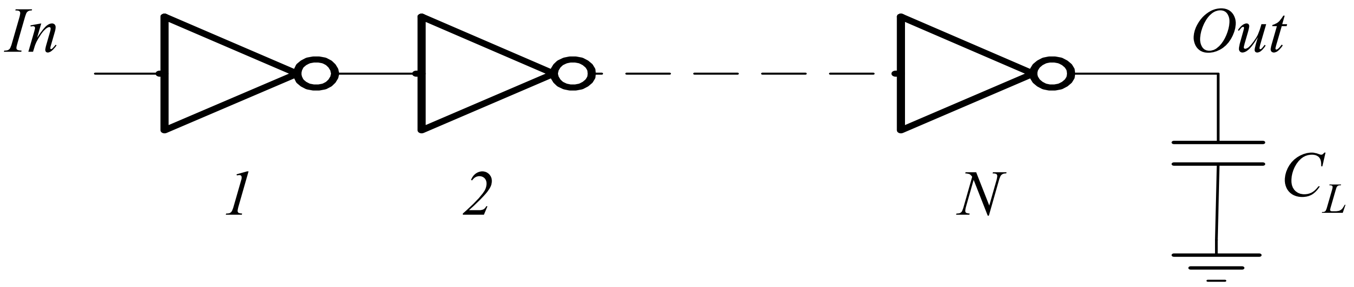
- Need to have a set of constraints
- Constraints key to:
  - Making the result useful
  - Making the problem have a ‘clean’ solution

- For sizing problem:
  - Need to constrain size of first inverter

Delay Optimization Problem #1

- You are given:
  - A fixed number of inverters
  - The size of the first inverter
  - The size of the load that needs to be driven
- Your goal:
  - Minimize the delay of the inverter chain
- Need model for inverter delay vs. size

Apply to Inverter Chain



$$t_p = t_{p1} + t_{p2} + \ldots + t_{pN}$$

$$t_{pj} = t_{inv}\left(\gamma + \frac{C_{in,j+1}}{C_{in,j}}\right)$$

$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{inv} \sum_{i=1}^{N} \left(\gamma + \frac{C_{in,j+1}}{C_{in,j}}\right), \quad C_{in,N+1} = C_L$$

Optimal Sizing for Given N

- Delay equation has N-1 unknowns, $C_{in,2} \dots C_{in,N}$

- **To minimize the delay,** find N-1 partial derivatives:

$$t_p = \ldots + t_{inv}\frac{C_{in,j}}{C_{in,j-1}} + t_{inv}\frac{C_{in,j+1}}{C_{in,j}} + \ldots$$

$$\frac{dt_p}{dC_{in,j}} = t_{inv}\frac{1}{C_{in,j-1}} - t_{inv}\frac{C_{in,j+1}}{C_{in,j}^2} = 0$$

Optimal Sizing for Given N (cont'd)

- Result: every stage has **equal fanout (f)**:

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

- Size of each stage is geometric mean of two neighbors:

$$\boxed{C_{in,j} = \sqrt{C_{in,j-1}C_{in,j+1}}}$$

- Equal fanout → every stage will have same delay

Optimum Delay and Number of Stages

- When each stage has same fanout f:

$$f^{N} = F = C_{L} / C_{in,1}$$

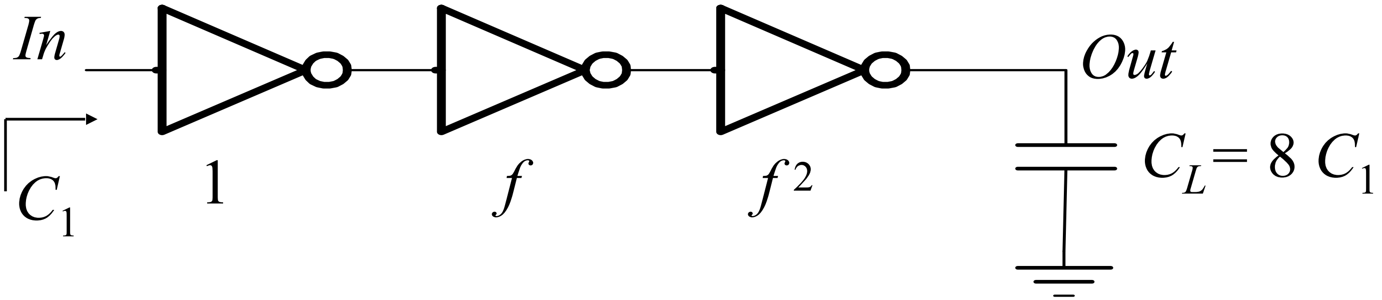
- Fanout of each stage:

$$f = \sqrt[N]{F}$$

- Minimum path delay:

$$t_{p} = Nt_{inv}\left(\gamma + \sqrt[N]{F}\right)$$

Example



C_L/C_1 has to be evenly distributed across $N = 3$ stages:

Delay Optimization Problem #2

- You are given:
  - The size of the first inverter
  - The size of the load that needs to be driven
- Your goal:
  - Minimize delay by finding optimal number and sizes of gates
- So, need to find N that minimizes:

$$t_p = N t_{inv} \left(\gamma + \sqrt[N]{C_L / C_{in}} \right)$$

Untangling the Optimization Problem

- Rewrite N in terms of fanout/stage *f*:

$$f^N = C_L / C_{in} \rightarrow \boxed{N = \frac{\ln\left(C_L / C_{in}\right)}{\ln f}}$$

$$t_p = N t_{inv}\left(\left(C_L / C_{in}\right)^{1/N} + \gamma\right) = t_{inv} \ln\left(C_L / C_{in}\right)\left(\frac{f + \gamma}{\ln f}\right)$$

$$\frac{\partial t_p}{\partial f} = t_{inv} \ln\left(C_L / C_{in}\right) \cdot \frac{\ln f - 1 - \gamma / f}{\ln^2 f} = 0$$

$$\boxed{f = \exp\left(1 + \gamma / f\right)}$$

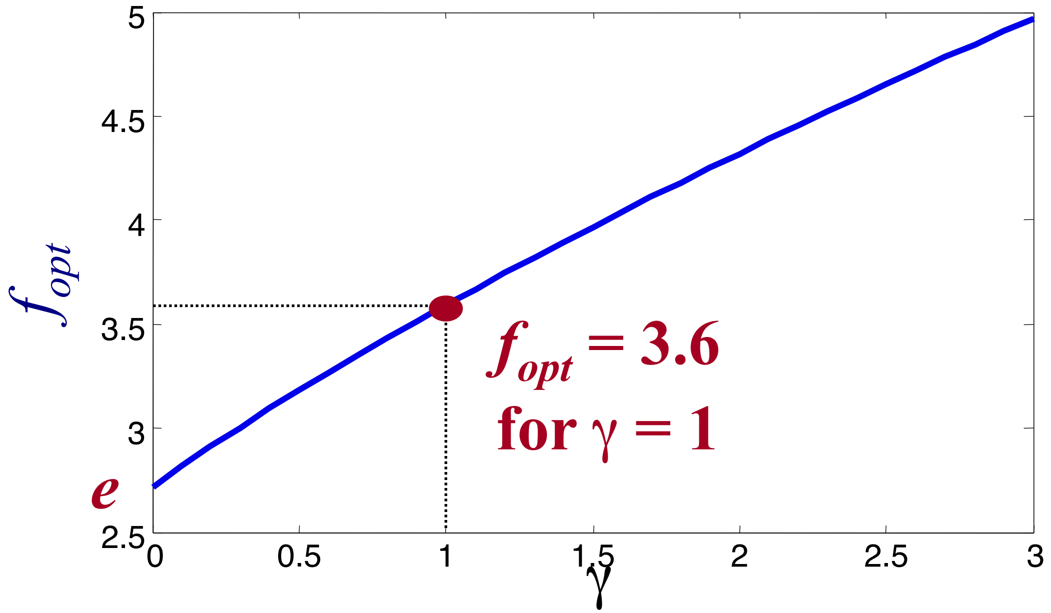
(no explicit solution)

For $\gamma = 0$, $f = e$, $N = \ln\,(C_L/C_{in})$

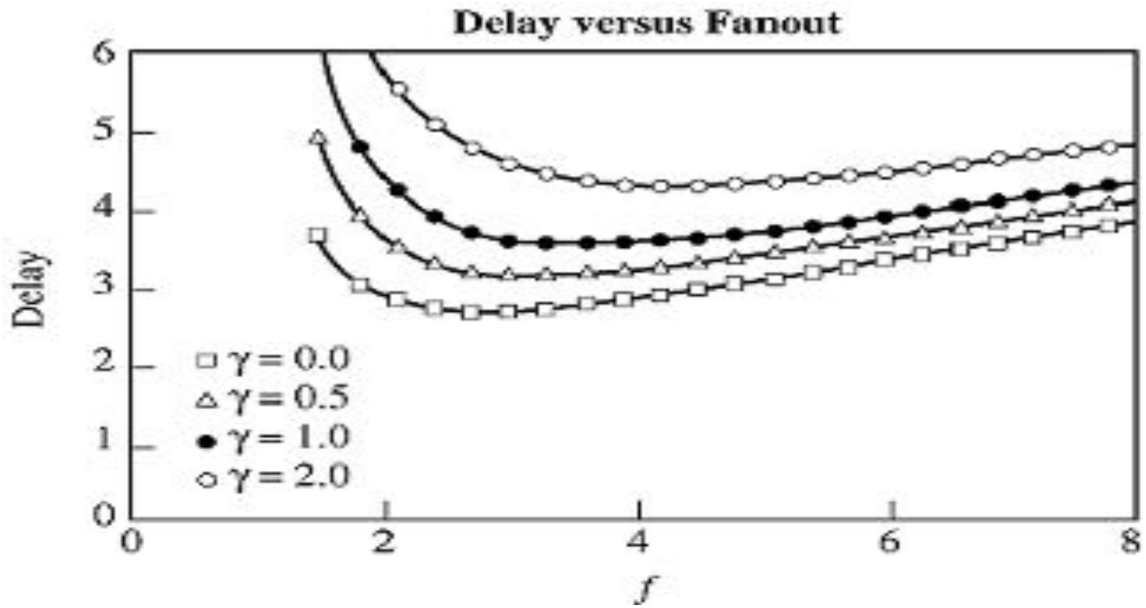
Optimum Effective Fanout f

- ❑ Optimum f for given process defined by γ

$$f = \exp(1 + \gamma / f)$$



In Practice: Plot of Total Delay

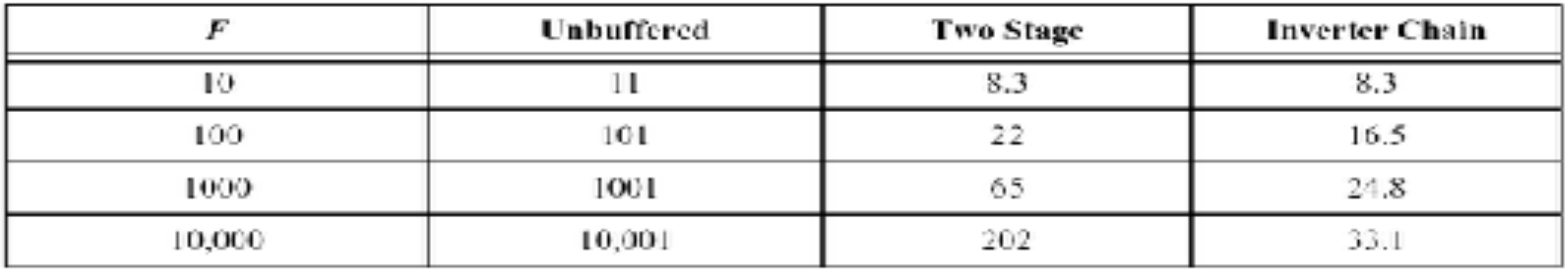


[Hodges, p.281]

- Why the shape?
- Curves very flat for *f* > 2
  - Simplest/most common choice: *f* = 4

Normalized Delay As a Function of F

$$t_p = N t_{inv} \left(\gamma + \sqrt[N]{F} \right), \; F = C_L / C_{in}$$

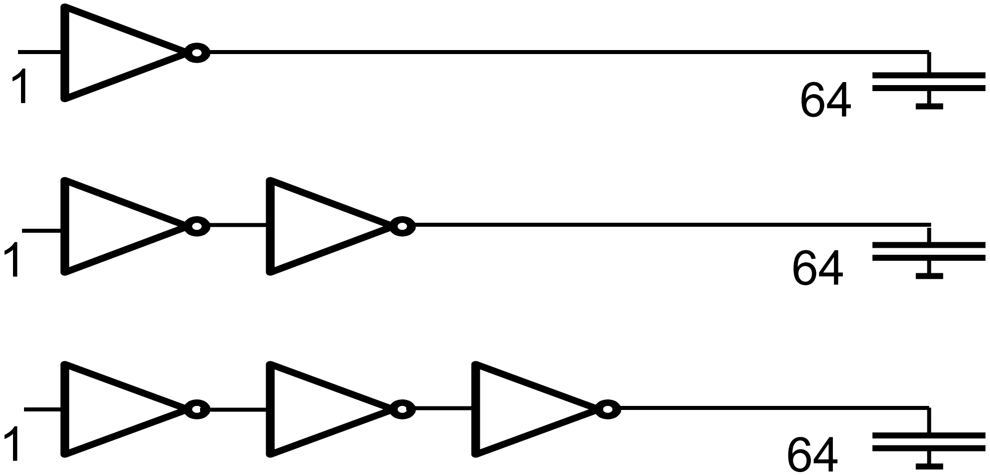


F	Unbuffered	Two Stage	Inverter Chain
10	11	8.3	8.3
100	101	22	16.5
1000	1001	65	24.8
10,000	10,001	202	33.1

($\gamma = 1$)

[Rabaey: page 210]

Buffer Design

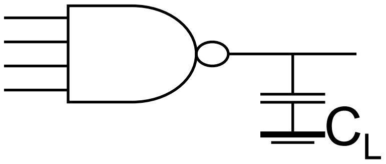


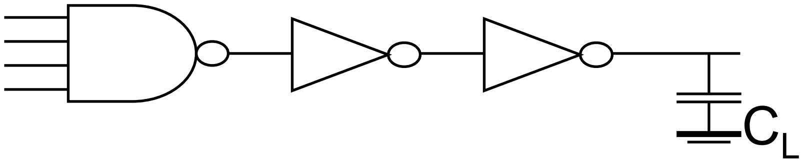
N	f	t_p
1		
2		
3		

Logical Effort

Question #1

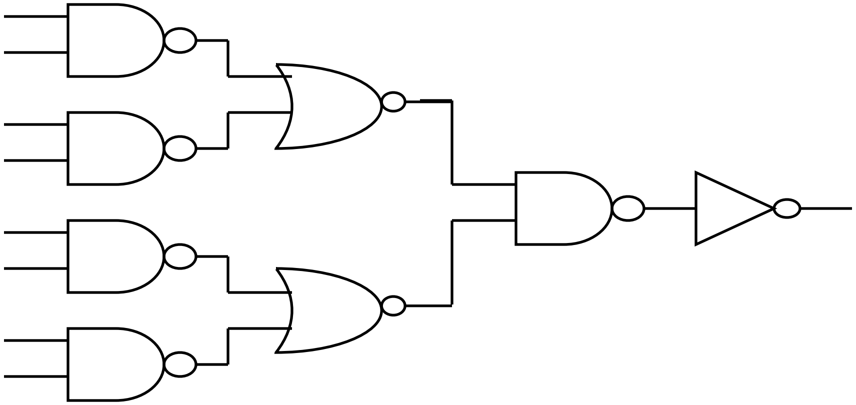
- How to best combine logic and drive for a big capacitive load?

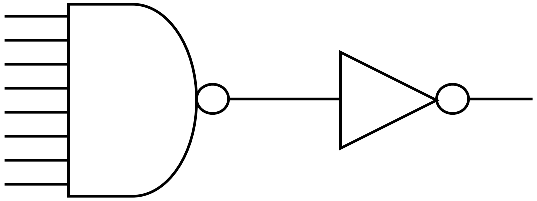


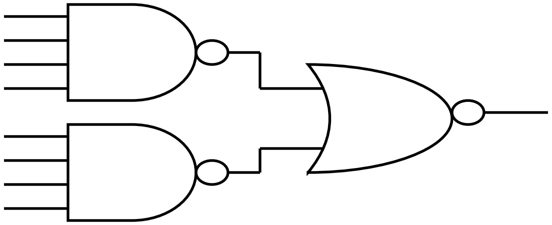


Question #2

- All of these are "decoders"
  - Which one is "best"?







Method to answer both of these questions

- ❑ Extension of buffer sizing problem
- ❑ Logical effort

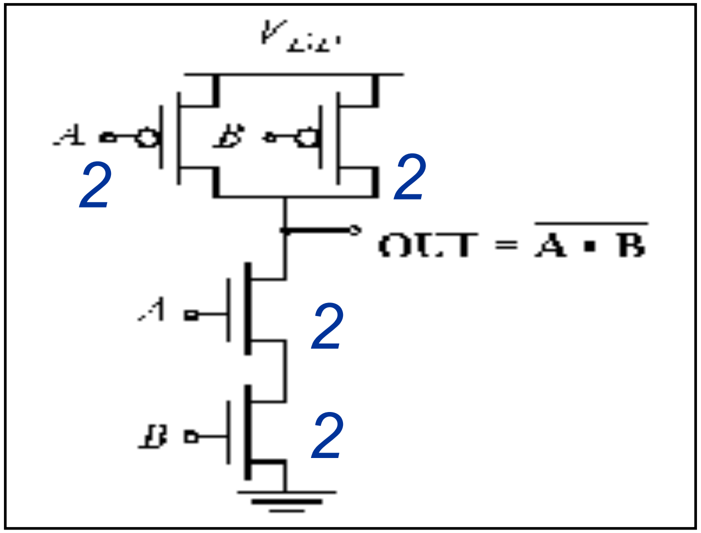
Complex Gate Sizing

Complex Gate Sizing: NAND-2 Example

$C_{gnand} = 4C_G = (4/3)\ C_{ginv}$

$C_{dnand} = 6C_D = 6\gamma C_G = 2\gamma C_{ginv}$

$f = C_L/C_{gnand} = (3/4)\ C_L/C_{ginv}$



$$
\begin{aligned}
t_{pNAND} &= kR_N(C_{dnand} + C_L) \\
&= kR_N(2\gamma C_{ginv} + C_L) \\
&= kR_N C_{ginv}\,(2\gamma + C_L/C_{ginv}) \\
&= t_{inv}\,(2\gamma + (4/3)f)
\end{aligned}
$$

Logical Effort

- Defines ease of gate to drive external capacitance
- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates
- Logical effort LE is defined as:
  - $(R_{eq,gate}C_{in,gate})/(R_{eq,inv}C_{in,inv})$
  - Easiest way to calculate (usually):
    - Size gate to deliver same current as an inverter, take ratio of gate input capacitance to inverter capacitance
- LE increases with gate complexity

Logical Effort

$$\boxed{t_{pgate} = t_{inv} \left(p + LEf\right)}$$

Measure everything in units of t_{inv} (divide by t_{inv}):

p – intrinsic delay - gate parameter $\neq$ f(W)
LE – logical effort – gate parameter $\neq$ f(W)
f – electrical fanout = C_L/C_{in} = f (W)

Normalize everything to an inverter:
$LE_{inv} = 1$, $p_{inv} = \gamma$

Delay of a Logic Gate

Gate delay:

$$Delay = EF + p \quad \text{(measured in units of } t_{inv}\text{)}$$

effective fanout   intrinsic delay

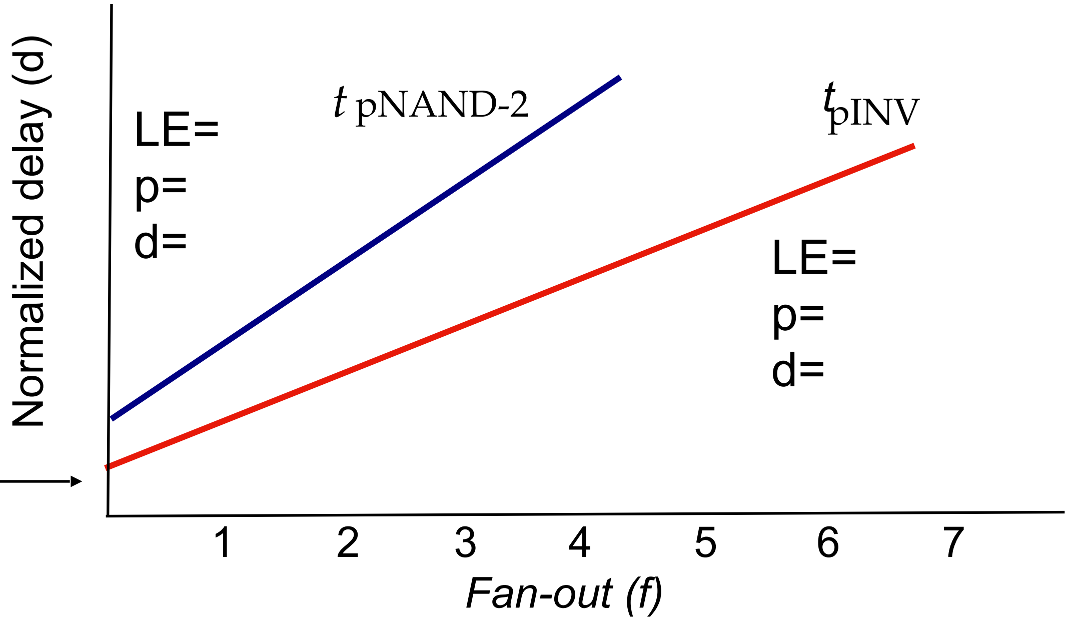
Effective fanout:

$$EF = LE\ f$$

logical effort   electrical fanout $= C_L/C_{in}$

Logical effort is a function of topology, independent of sizing
Effective fanout is a function of load/gate size

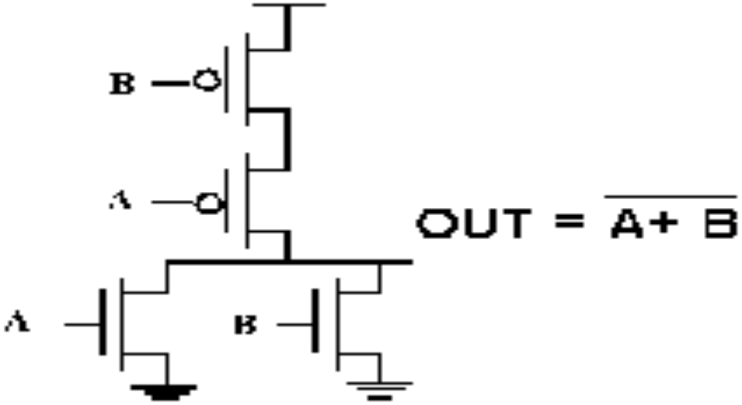
Logical Effort of Gates



p = γ·Fan-in
(for top input)

Delay Of NOR-2 Gate

1. Size for same resistance as inverter
2. LE = ratio of input cap of gate versus inverter



Intrinsic capacitance (C_{dnor}) =

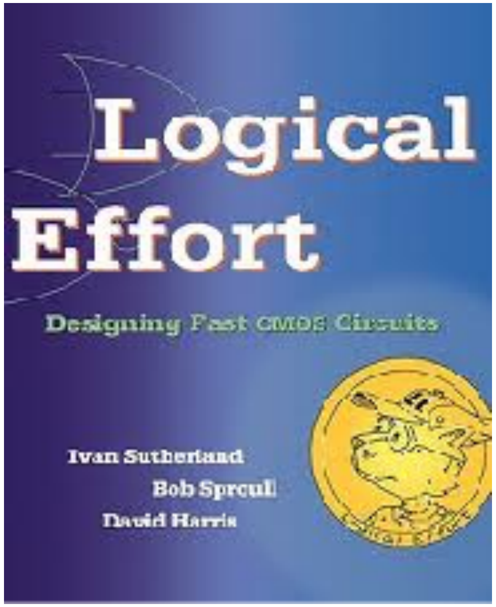
t_{pint} (NOR) =

Question

Any logic function can be implemented using NOR gates only or NAND gates only!

Which of the two approaches is preferable in CMOS (from a performance perspective)?

Logical Effort



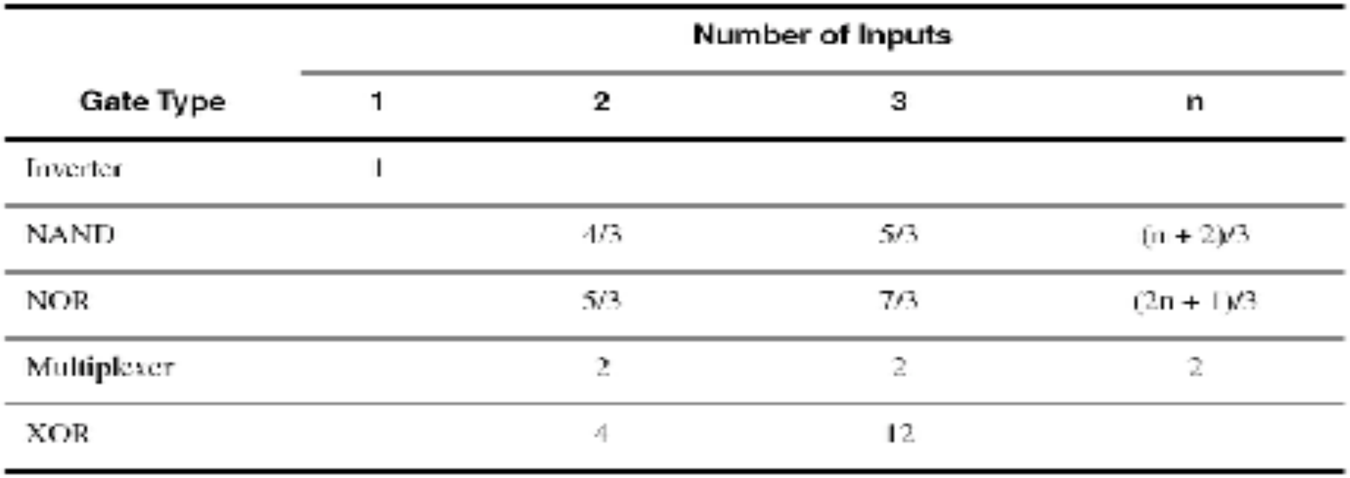
Logical Effort

Designing Fast CMOS Circuits

Ivan Sutherland

Bob Sproull

David Harris



	Number of Inputs			
Gate Type	1	2	3	n
Inverter	1			
NAND		4/3	5/3	(n + 2)/3
NOR		5/3	7/3	(2n + 1)/3
Multiplexer		2	2	2
XOR		4	12	

[From Sutherland, Sproull, Harris]

Optimizing Complex Combinational Logic

Multistage Networks

$$Delay = \sum_{i=1}^{N} \left(p_i + LE_i \cdot f_i \right)$$

Effective fanout: $EF_i = LE_i f_i$

Only for tree networks

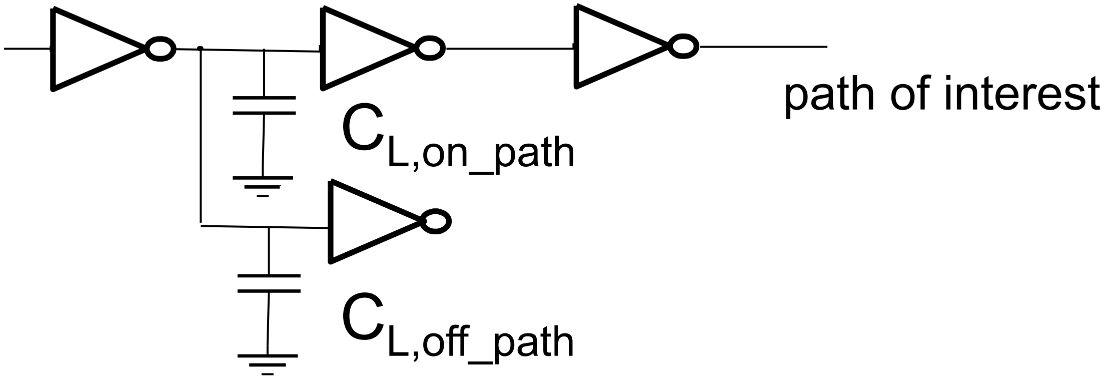
Path delay $D = \Sigma d_i = \Sigma p_i + \Sigma EF_i$

Path electrical fanout: $F = C_L/C_{in} = \Pi f_i$

Path logical effort: $\Pi LE = LE_1 LE_2 \ldots LE_N$

Path effort: $PE = \Pi LE\ F$

Adding branching



Branching effort: $$b = \frac{C_{L,on-path} + C_{L,off-path}}{C_{L,on-path}}$$

Multistage Networks

$$\boxed{Delay = \sum_{i=1}^{N} \left(p_i + LE_i \cdot f_i \right)}$$

Effective fanout: $\mathrm{EF}_i = \mathrm{LE}_i f_i$

Path delay $D = \Sigma d_i = \Sigma p_i + \Sigma \mathrm{EF}_i$

Path electrical fanout: $F = C_L / C_{in}$

Branching effort: $\Pi B = b_1 b_2 \ldots b_N$

$\Pi f_i = \Pi \mathrm{B}\ F$ *(assuming all paths in the tree are important)*

Path logical effort: $\Pi LE = LE_1 LE_2 \ldots LE_N$

Path effort: $PE = \Pi LE\ \Pi \mathrm{B}\ F$

Optimum Effort per Stage

When each stage bears the same effort (effective fanout):

$$EF^{N} = PE$$

$$EF = \sqrt[N]{PE}$$

Effective fanouts: $LE_1 f_1 = LE_2 f_2 = \ldots = LE_N f_N$

Minimum path delay

$$\hat{D} = \sum_{i=1}^{N} \left(LE_i f_i + p_i \right) = N \cdot PE^{1/N} + \sum_{i=1}^{N} p_i$$

Optimal Number of Stages

For a given load,
and given input capacitance of the first gate
Find optimal number of stages and optimal sizing

$$D = N \cdot PE^{1/N} + \sum p_i$$

Remember: we can always add inverters to the end of the chain

The ‘best effective fanout’ $EF = PE^{1/\hat{N}}$ is still around 4
(3.6 with γ=1)

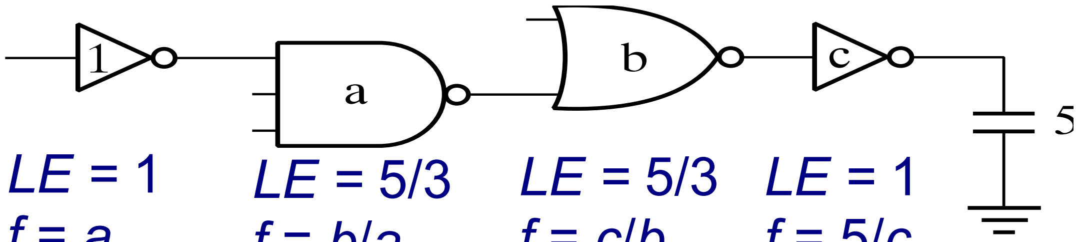
Method of Logical Effort: Summary

- Compute the path effort: PE = $(\Pi \text{LE})BF$
- Find the best number of stages $N \sim \log_4 \text{PE}$
- Compute the effective fanout/stage EF = $\text{PE}^{1/N}$
- Sketch the path with this number of stages
- Work either from either end, find sizes: $C_{in} = C_{out}$*LE/EF

Reference: Sutherland, Sproull, Harris, “Logical Effort”, Morgan-Kaufmann 1999.

Optimizing Complex Combinational Logic: Examples

Example 1: No branching



LE = 1
f = *a*

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

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

LE = 1
f = 5/*c*

Electrical fanout, F =

Π *LE* =

PE =

EF/stage =

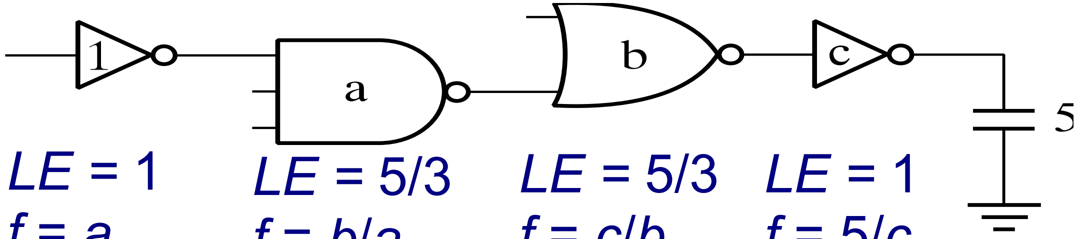
a =

b =

c =

Example 1: No branching

a, b, c are input capacitances normalized to the unit inverter



LE = 1
f = *a*

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

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

LE = 1
f = 5/*c*

Electrical fanout, F = 5

Π *LE* = 25/9

PE = 125/9

EF/stage = 1.93

a = 1.93

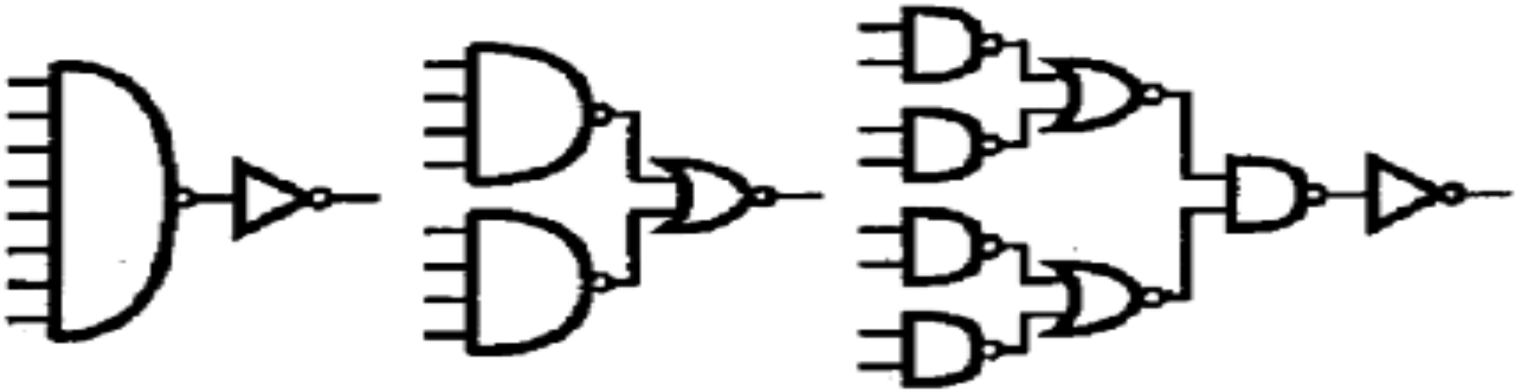
b = 2.23

c = 2.59

From the back

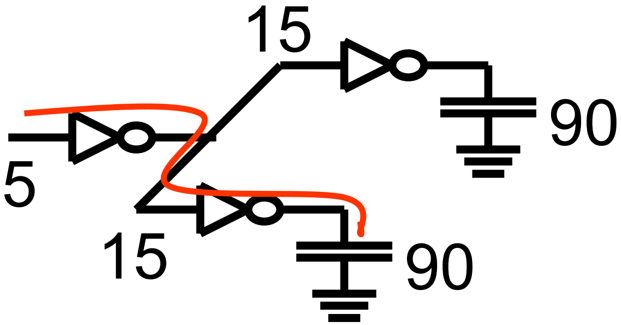
5/c = 1.93
(5/3)c/b = 1.93
(5/3)b/a = 1.93

Our old problem: which one is better?



LE=10/3 1	*LE*=2 5/3	*LE*=4/3 5/3 4/3 1
ΠLE = 10/3	ΠLE = 10/3	ΠLE = 80/27
P = 8 + 1	P = 4 + 2	P = 2 + 2 + 2 + 1

Adding Branching



$LE = 1$

$F = 90/5 = 18$

$PE = 18$ (wrong!)

$EF_1 = (15+15)/5 = 6$

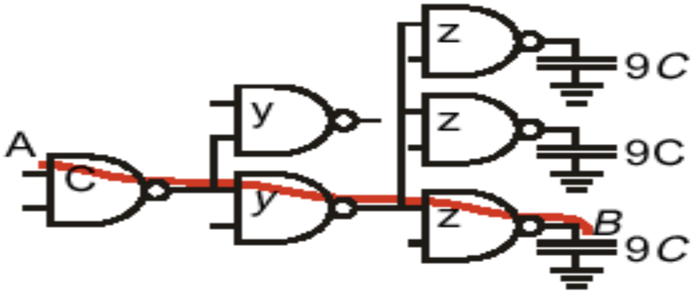
$EF_2 = 90/15 = 6$

$PE =$ 36, not 18!

Better: $PE = F \cdot LE \cdot B = 18 \cdot 1 \cdot 2 = 36$

Example 2 with Branching

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



Logical Effort: *LE* =

Electrical Fanout: *F* =

Branching Effort: *B* =

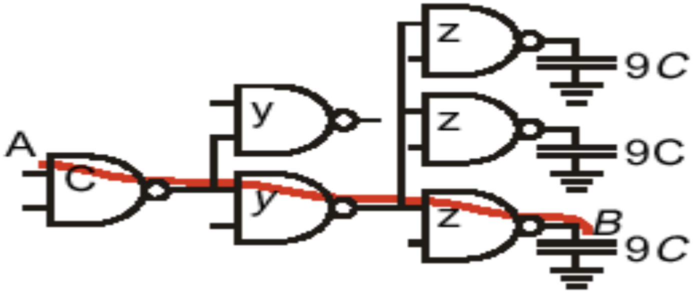
Path Effort: *PE* =

Best Effective Fanout: *EF* =

Delay: *D* =

Example 2 with Branching

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



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

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

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

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

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

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

Work backward for sizes:

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

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