ECE 225
High-Speed Digital IC Design

Lecture 3

Review of Basic CMOS Design-1

Prof. Kaustav Banerjee
Electrical and Computer Engineering
E-mail: kaustav@ece.ucsb.edu

Static and Dynamic CMOS

- In static circuits at every point in time (except when switching) the output is connected to either GND or V_{DD} via a low resistance path.
  - fan-in of *n* requires 2*n* (*n* N-type + *n* P-type) devices

- Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.
  - requires only *n* + 2 (*n*+1 N-type and 1 P-type) transistors

Static and Dynamic CMOS Circuit Families

- Static:
  - Complementary CMOS
    - (robustness, low power, large fan-in expensive in terms of area and performance)
  - Ratioed Logic (pseudo-NMOS, DCVSL)
    - (simple and fast at the expense of reduced NM and static power)
  - Pass-Transistor Logic (Transmission Gate)
  - -attractive for specific circuits: MUX, XOR-dominated logic such as Adders)
- Dynamic:
  - good for fast and complex gates, design process is harder due to parasitic effects, leakage puts an upper limit on the operating frequency of the circuit
    - Domino Logic
    - np-CMOS

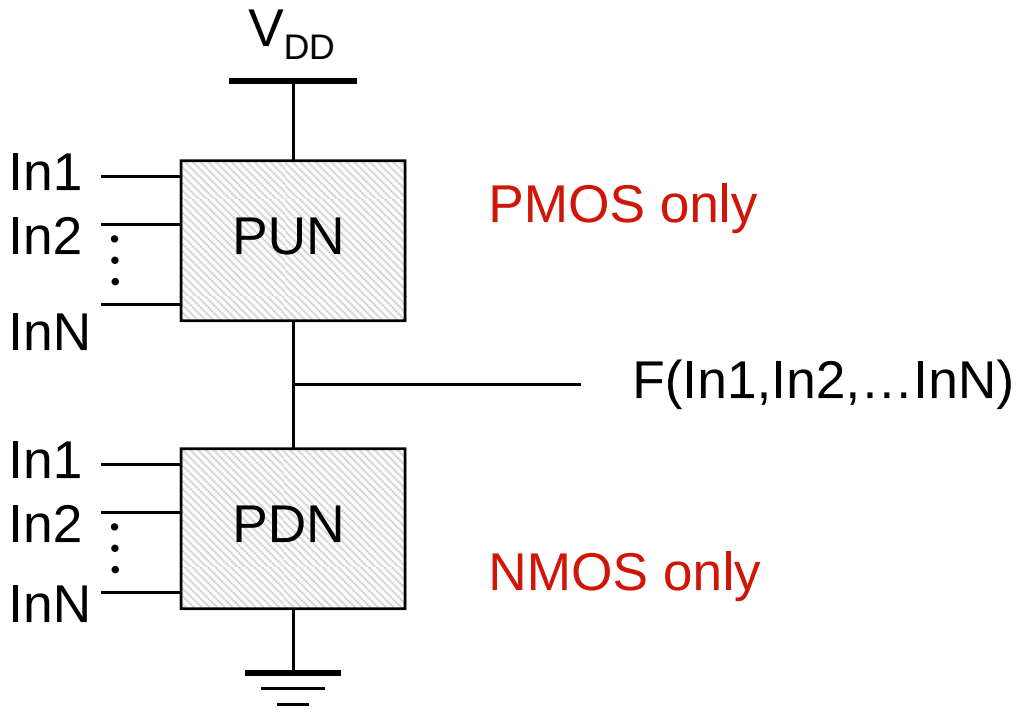
Which style is best?

……depends on ease of design, performance, power, area and robustness.

Complementary CMOS Logic

- Full rail-to-rail swing; high noise margins
- Logic levels not dependent upon the relative device sizes; ratioless
- Always a path to Vdd or Gnd in steady state; low output impedance
- Extremely high input resistance; nearly zero steady-state input current
- No direct path steady state between power and ground; no static power dissipation
- Propagation delay function of load capacitance and resistance of transistors

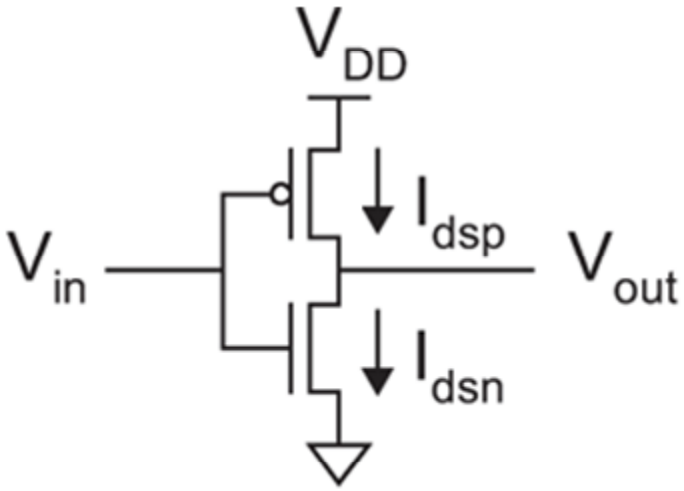
Static Complementary CMOS



PUN and PDN are dual logic networks

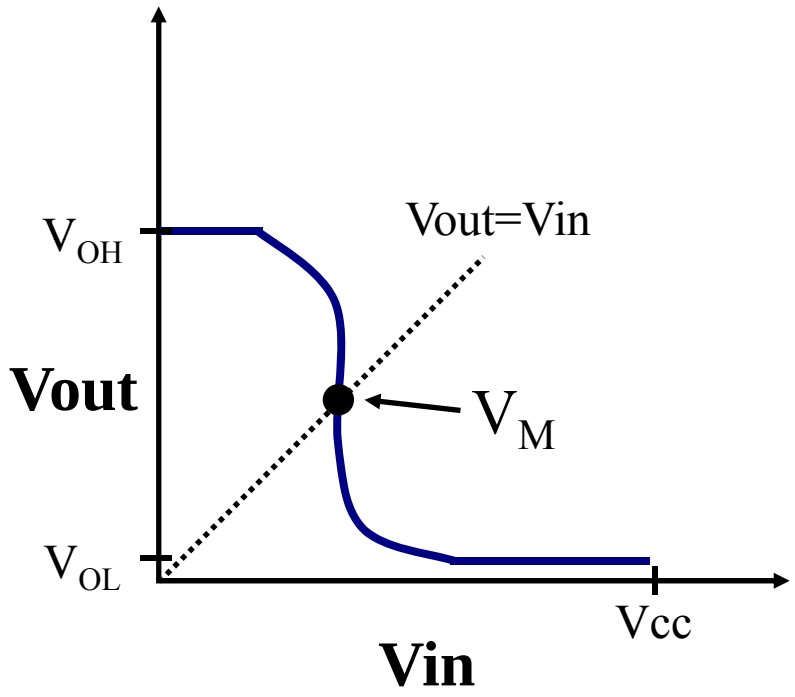
CMOS Inverter

The most widely used gate….



A CMOS inverter

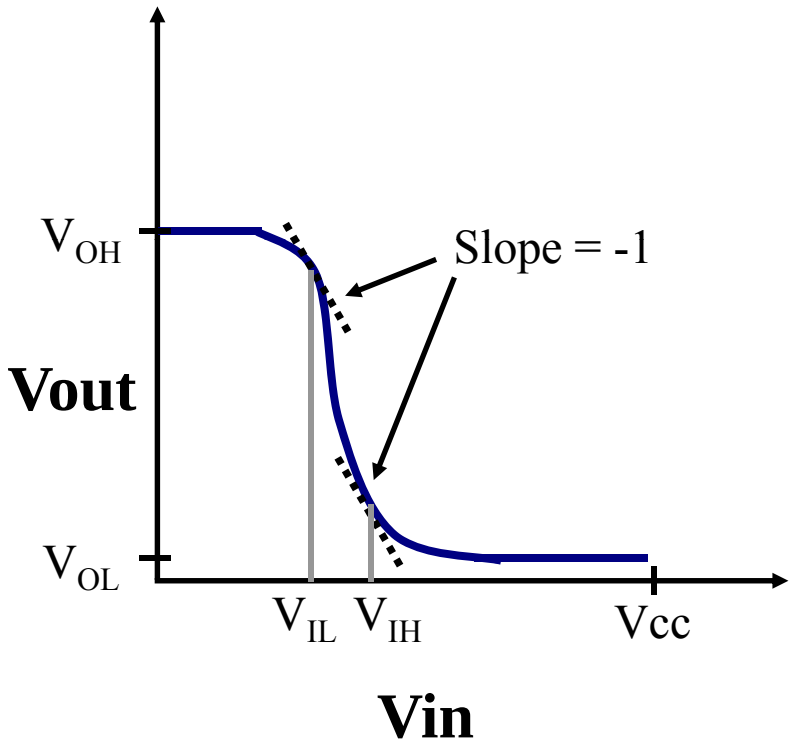
Inverter Threshold



Inverter switching threshold:

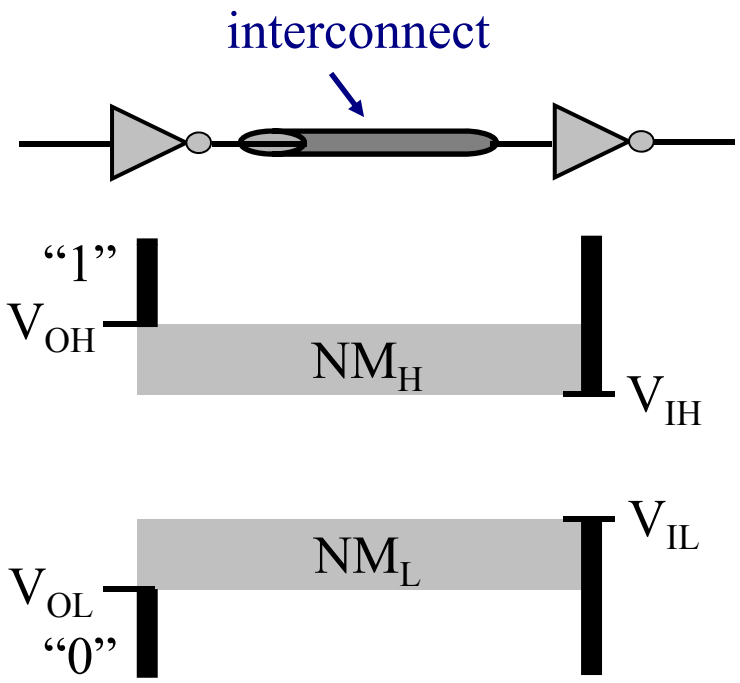
- Point where voltage transfer curve intersects line Vout=Vin
- Represents the point at which the inverter switches state
- Normally, $V_M \approx Vcc/2$

Noise Margins



- V_{IL} and V_{IH} measure effect of input voltage on inverter output
- V_{IL} = largest input voltage recognized as logic ‘0’
- V_{IH} = smallest input voltage recognized as logic ‘1’
- Defined as point on VTC where slope = -1

Noise Margin (cont)

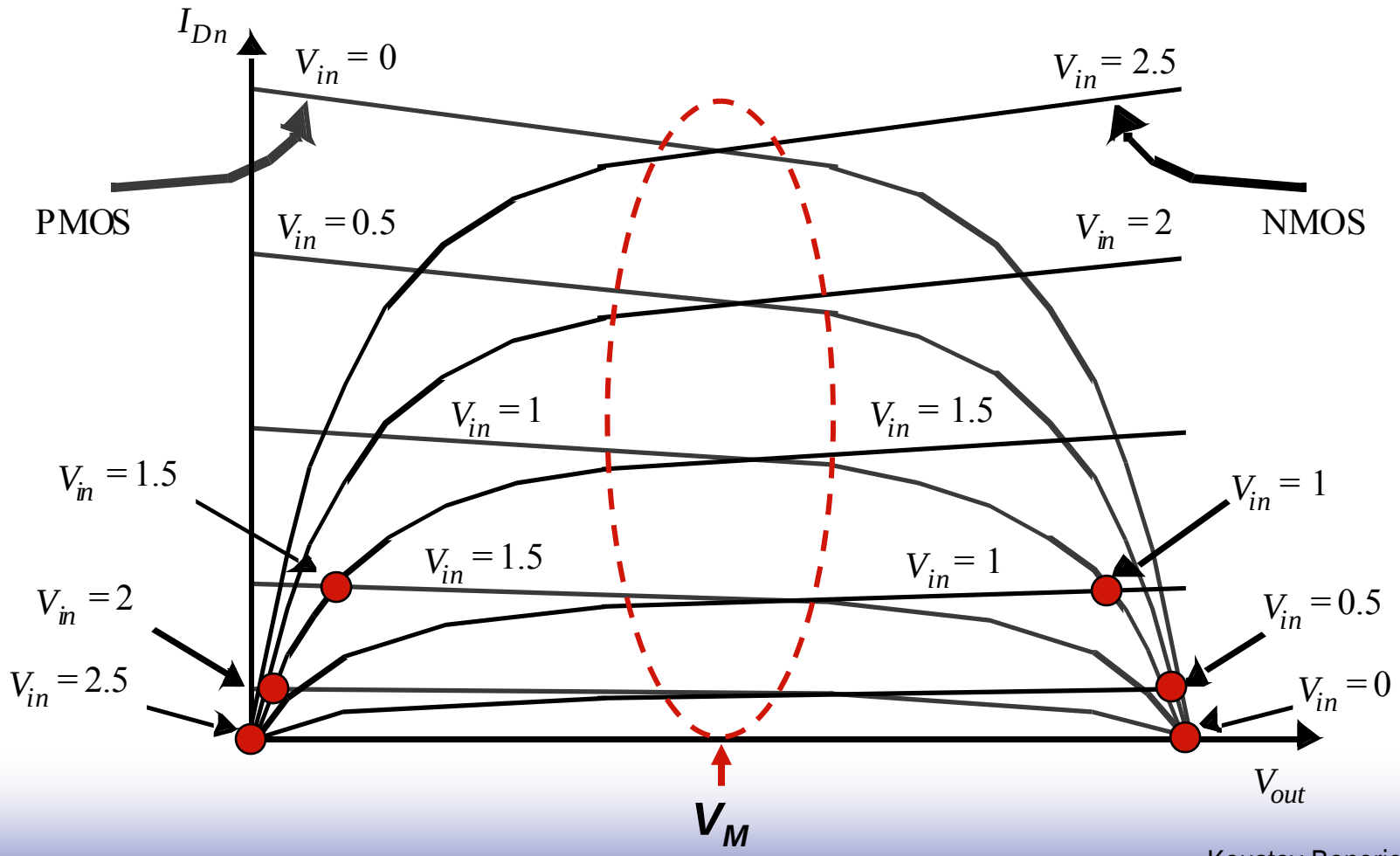


Ideally, noise margin should be as large as possible

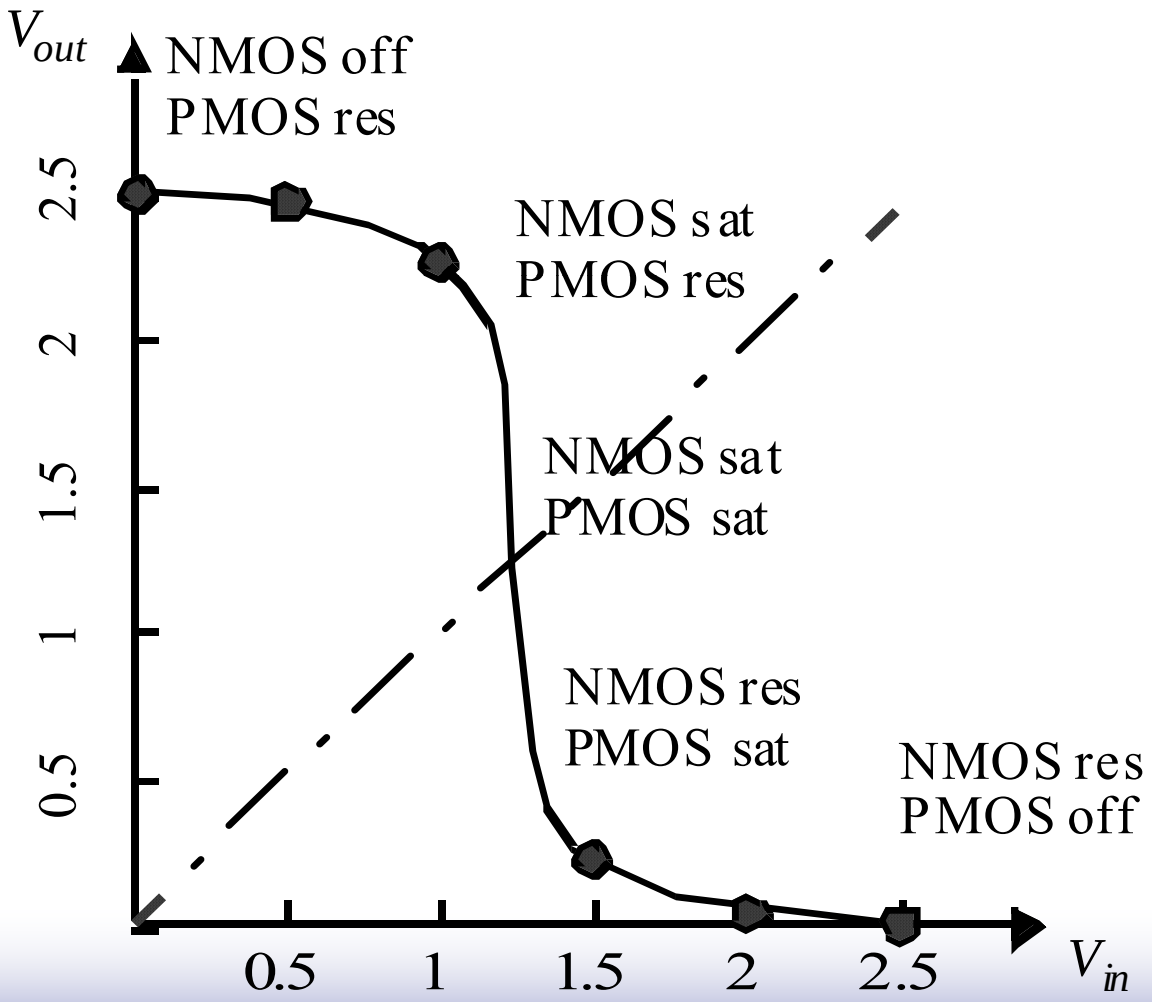
- Noise margin is a measure of the *robustness* of an inverter
  - $N_{ML} = V_{IL} - V_{OL}$
  - $N_{MH} = V_{OH} - V_{IH}$
- Models a chain of inverters. Example:
  - First inverter output is V_{OH}
  - Second inverter recognizes input $> V_{IH}$ as logic '1'
  - Difference V_{OH}-V_{IH} is "safety zone" for noise

CMOS Inverter Load Characteristics

B. For a DC operating point to be valid, currents through NMOS and PMOS must be equal (for a given V_{in}), hence find the points of intersection.



CMOS Inverter VTC

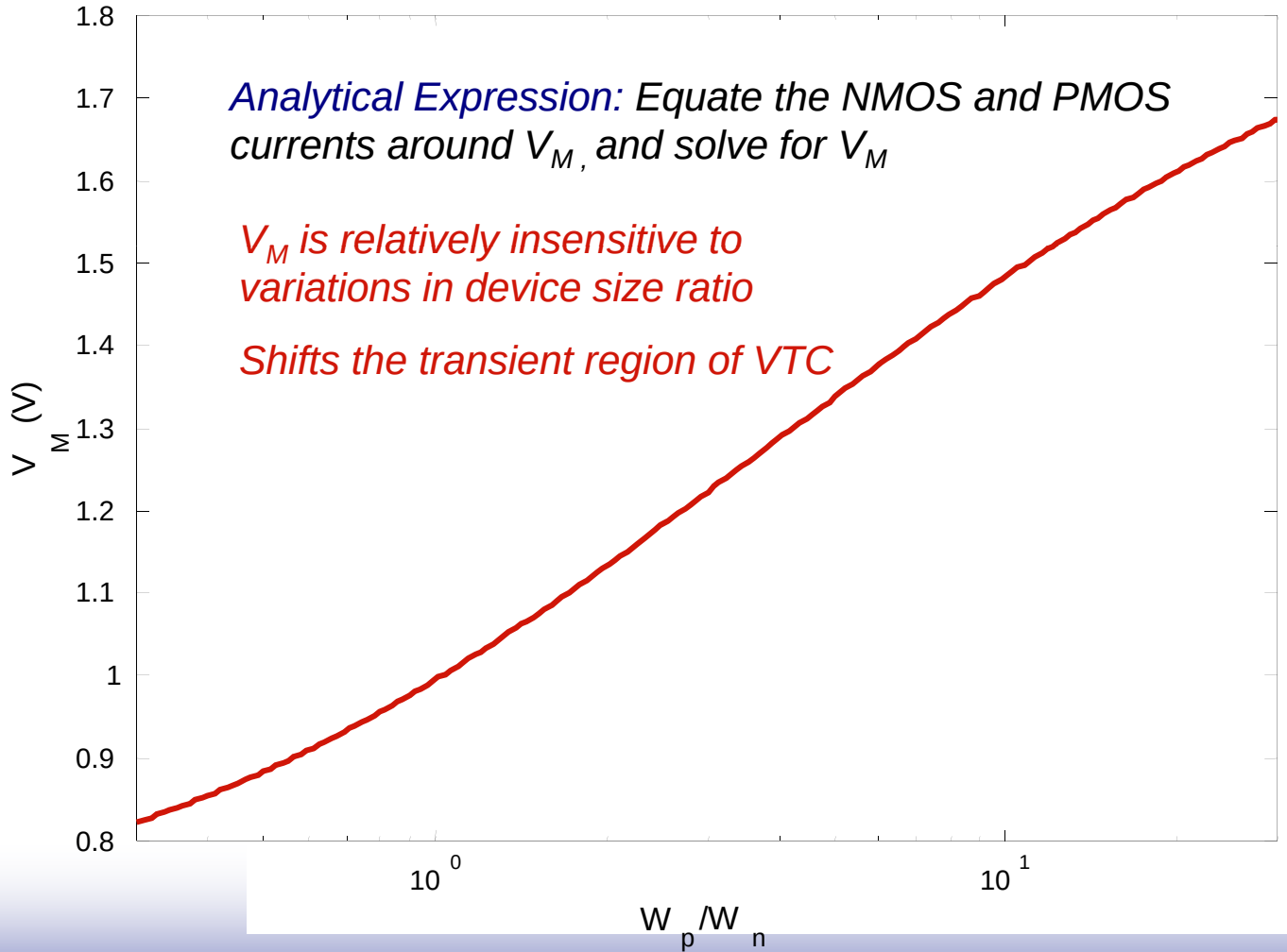


CMOS Inverter Operation (summary)

Table 2.2 Relationships between voltages for the three regions of operation of a CMOS inverter

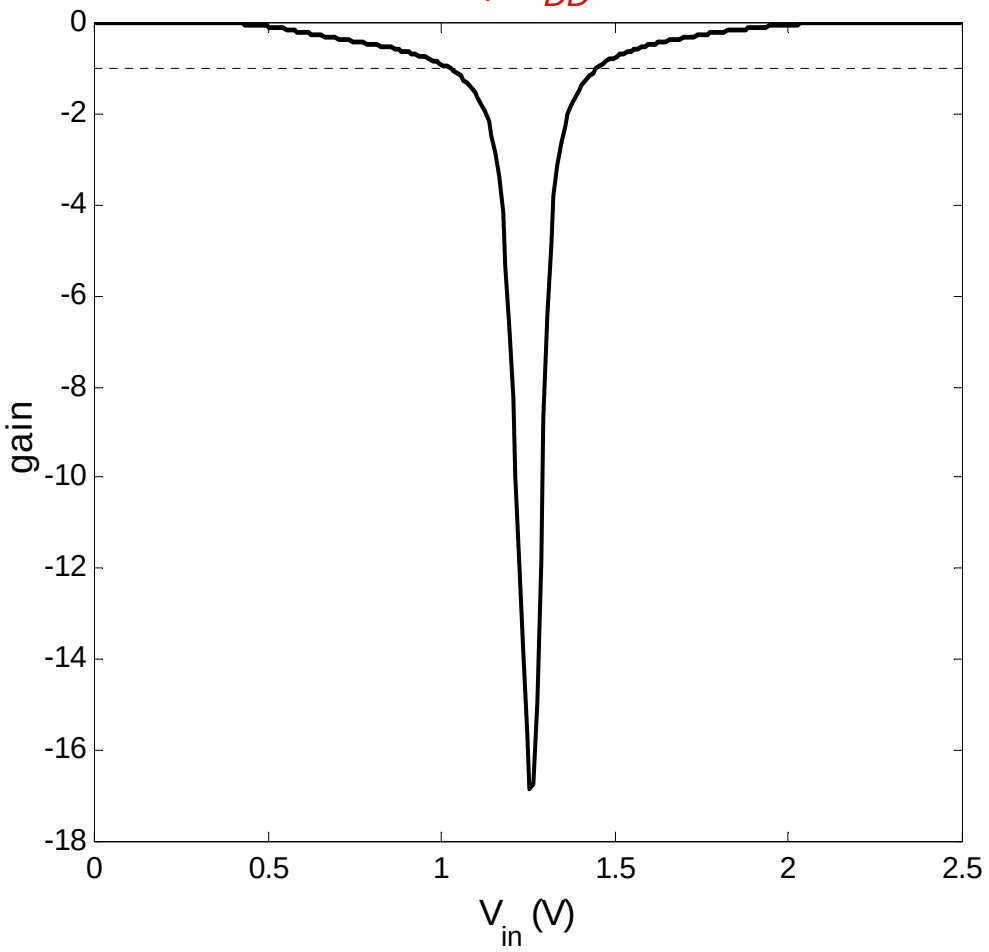
	Cutoff	Linear	Saturated
nMOS	$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$	$V_{gsn} > V_{tn}$
	$V_{in} < V_{tn}$	$V_{in} > V_{tn}$	$V_{in} > V_{tn}$
		$V_{dsn} < V_{gsn} - V_{tn}$	$V_{dsn} > V_{gsn} - V_{tn}$
		$V_{out} < V_{in} - V_{tn}$	$V_{out} > V_{in} - V_{tn}$
pMOS	$V_{gsp} > V_{tp}$	$V_{gsp} < V_{tp}$	$V_{gsp} < V_{tp}$
	$V_{in} > V_{tp} + V_{DD}$	$V_{in} < V_{tp} + V_{DD}$	$V_{in} < V_{tp} + V_{DD}$
		$V_{dsp} > V_{gsp} - V_{tp}$	$V_{dsp} < V_{gsp} - V_{tp}$
		$V_{out} > V_{in} - V_{tp}$	$V_{out} < V_{in} - V_{tp}$

Switching Threshold as a function of Transistor Ratio



Inverter Gain

0.25 um CMOS, V_{DD} = 2.5V



Gain is mostly determined by technology parameters, especially channel length modulation, but also by V_{DD}

Note: approximately $V_M \propto V_{DD}$

Inverter Skew

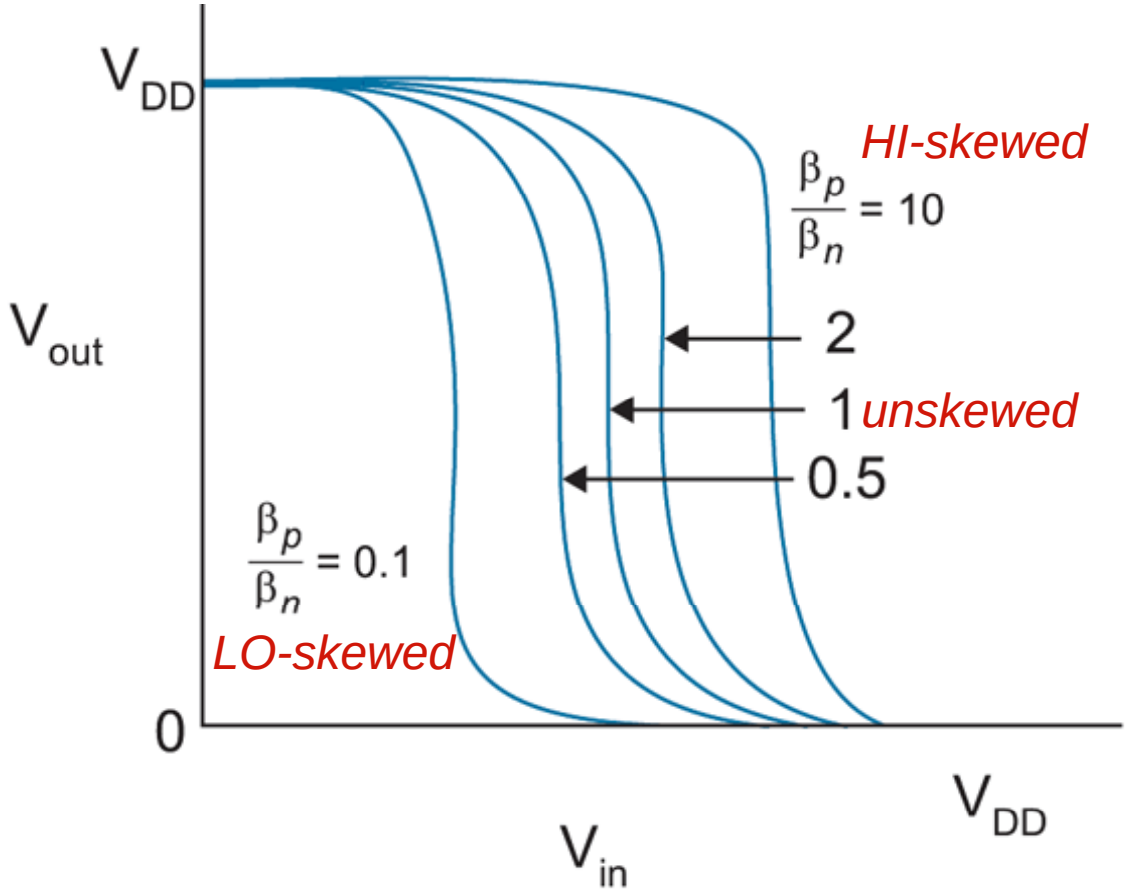
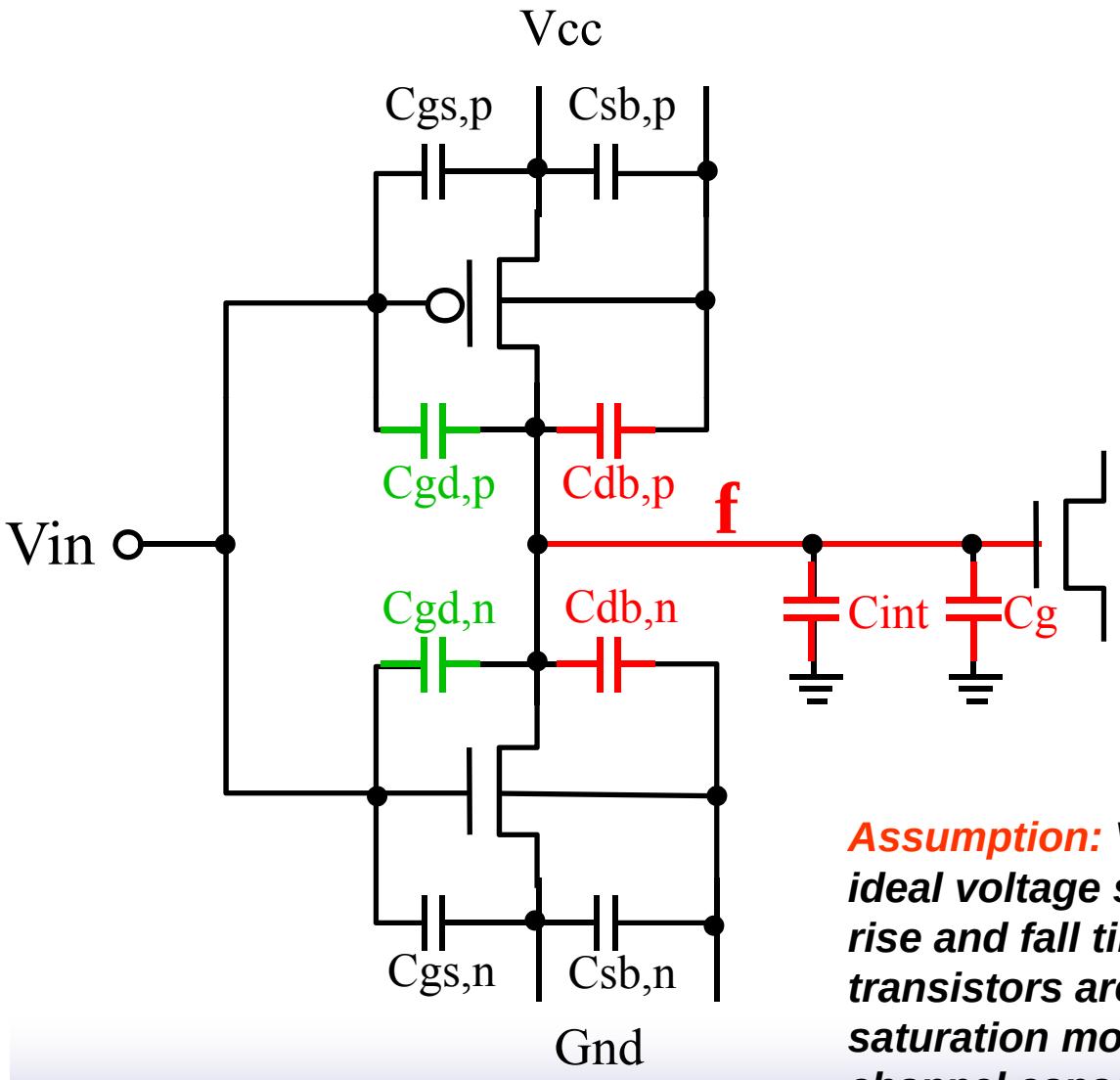


FIG 2.26 Transfer characteristics of skewed inverters

Propagation Delay

CMOS inverter capacitances



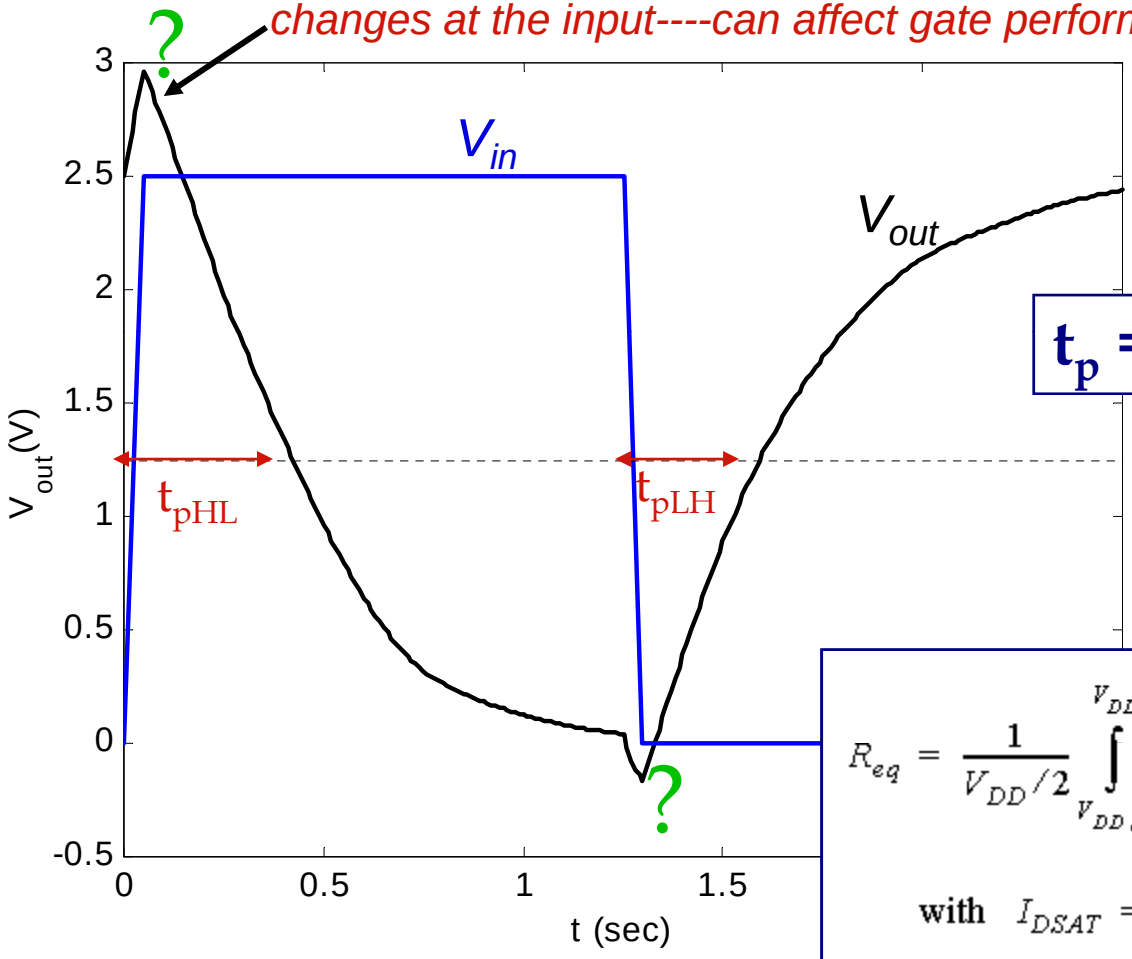
Cap on Node f:

- Junction cap: $C_{db,p}$ and $C_{db,n}$
- Gate (overlap) capacitance $C_{gd,p}$ and $C_{gd,n}$ (beware of Miller effect)
- Interconnect cap: C_{int}
- Receiver gate cap: C_g

Assumption: ***Vin is driven by an ideal voltage source….with zero rise and fall times….hence the transistors are either in cut-off or saturation mode…hence, no channel capacitance***

Transient Response

Due to C_{gd} of transistors: directly couples voltage at input to output before the transistors can even start to react to changes at the input----can affect gate performance



Symmetric inverter has $t_{pHL}=t_{pLH}$

$$t_p = 0.69\, C_L\, (R_{eqn}+R_{eqp})/2$$

$$R_{eq} = \frac{1}{V_{DD}/2}\int_{V_{DD}/2}^{V_{DD}} \frac{V}{I_{DSAT}(1+\lambda V)}dV \approx \frac{3}{4}\frac{V_{DD}}{I_{DSAT}}\left(1-\frac{7}{9}\lambda V_{DD}\right)$$

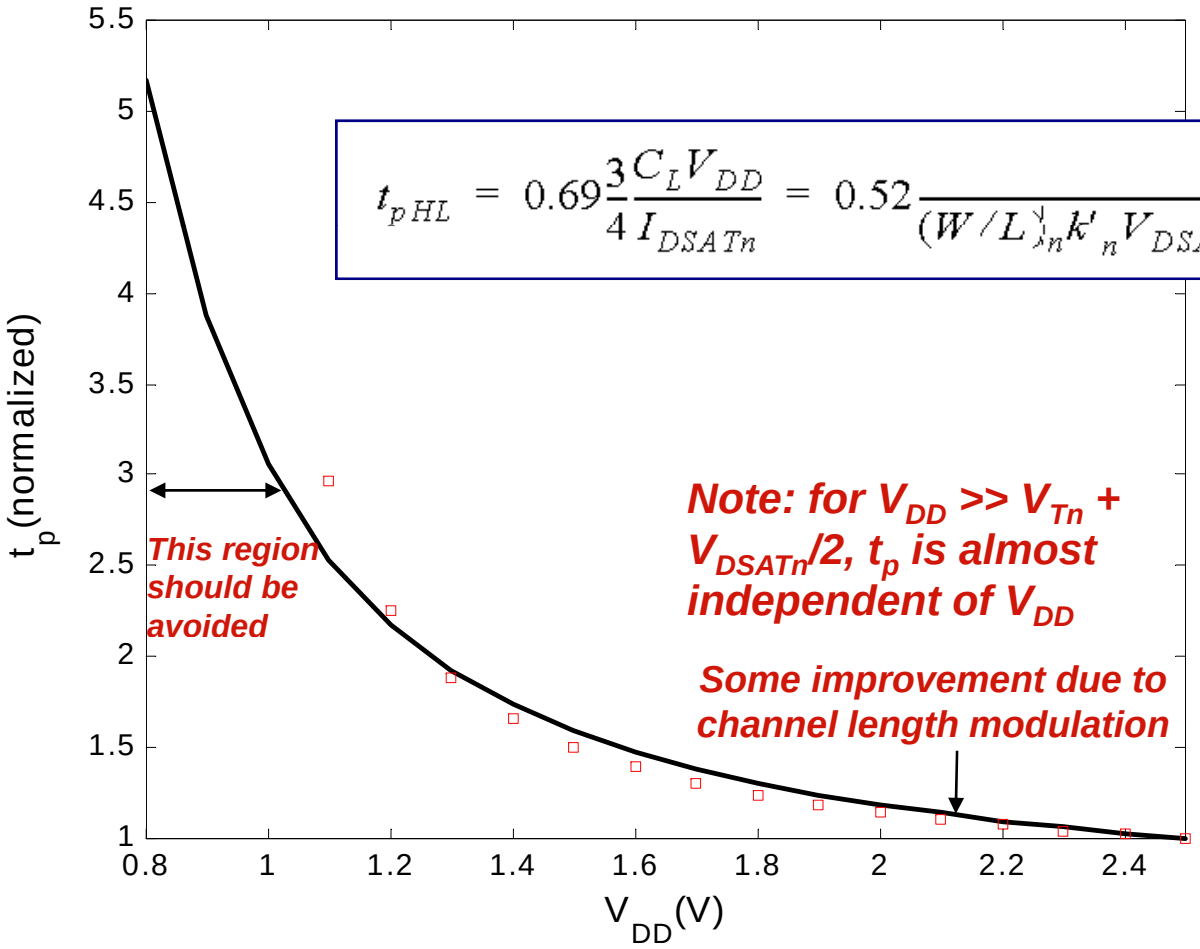
$$\text{with}\quad I_{DSAT} = k'\frac{W}{L}\left((V_{DD}-V_T)V_{DSAT}-\frac{V_{DSAT}^2}{2}\right)$$

Delay as a function of V_{DD}

Same as the ON resistance of a transistor….

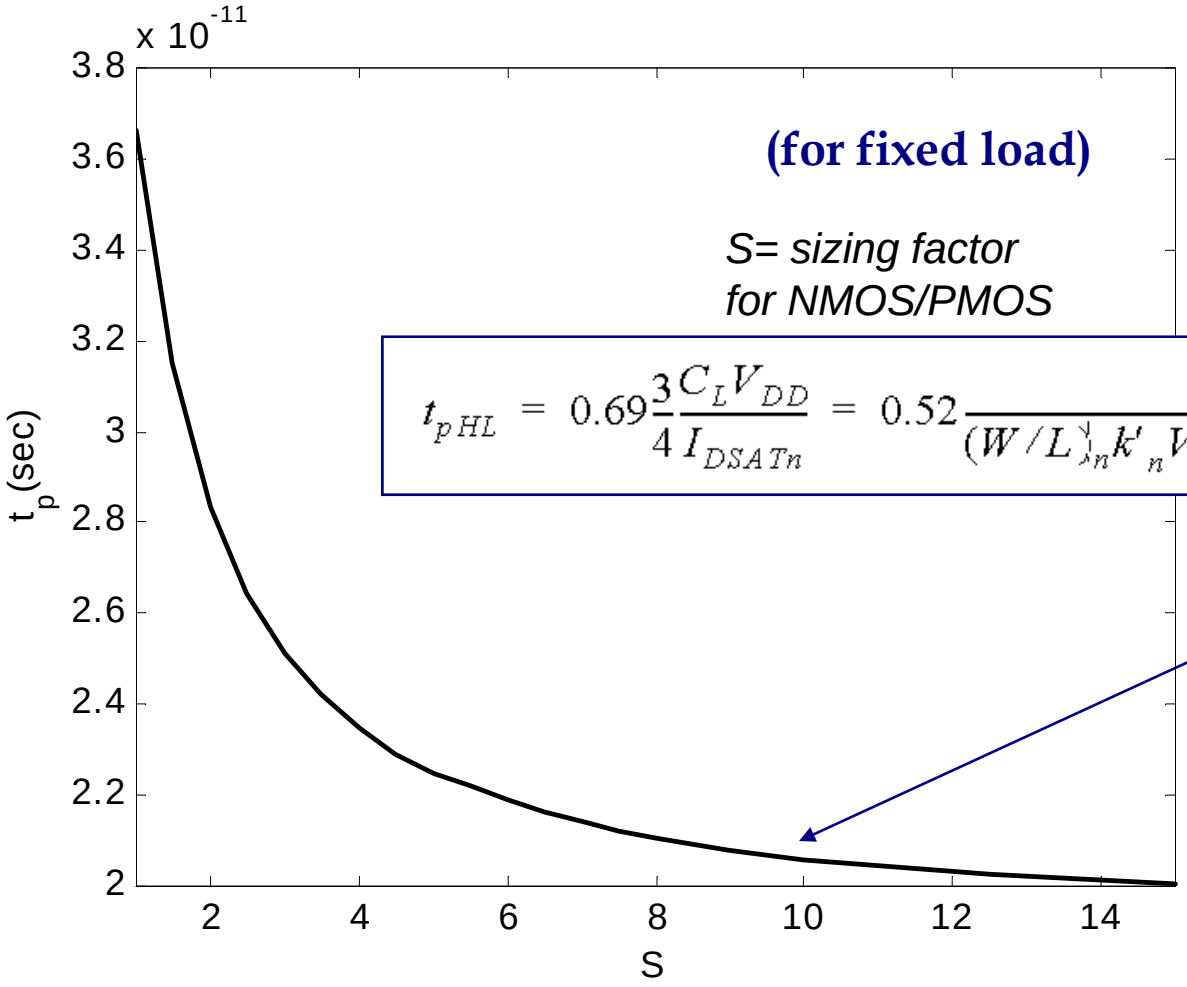
Trade off energy dissipation vs performance…..

$$t_{pHL} = 0.69\frac{3}{4}\frac{C_L V_{DD}}{I_{DSATn}} = 0.52\frac{C_L V_{DD}}{(W/L)_n k'_n V_{DSATn}(V_{DD} - V_{Tn} - V_{DSATn}/2)}$$



Reliability concerns at high V_{DD}….

Effect of Device Sizing



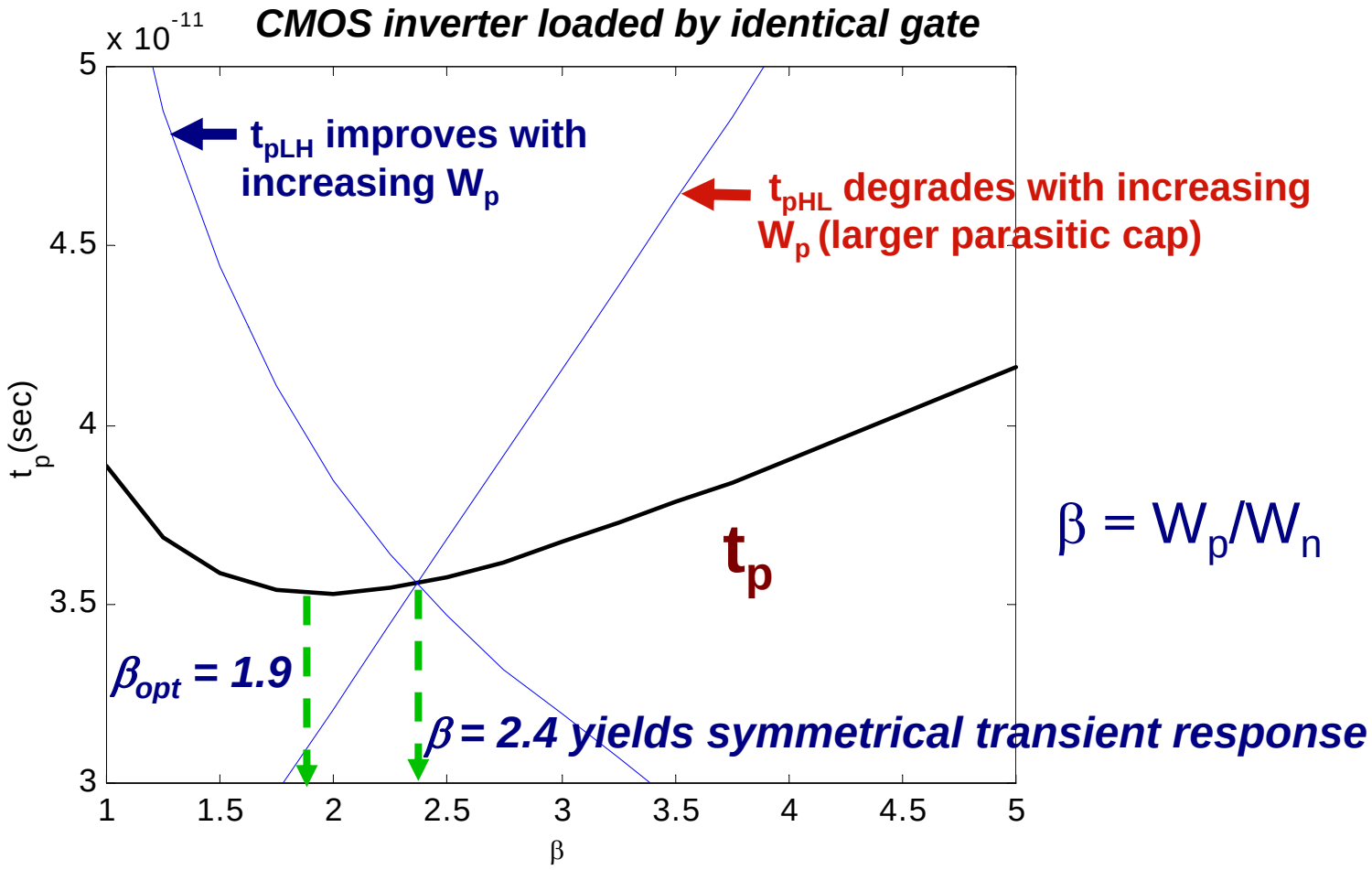
(for fixed load)

S= sizing factor for NMOS/PMOS

$$t_{pHL} = 0.69\frac{3}{4}\frac{C_L V_{DD}}{I_{DSATn}} = 0.52\frac{C_L V_{DD}}{(W/L)_n k'_n V_{DSATn}(V_{DD} - V_{Tn} - V_{DSATn}/2)}$$

Self-loading effect: Intrinsic (diffusion) capacitances dominate

NMOS/PMOS ratio



$\beta = W_p/W_n$

If symmetry and noise margins are not of prime concern, inverter delay can be reduced by reducing the width of PMOS….

Design for Performance

- Keep load capacitances (C_L) small
  - Recall that three major components contribute to the load cap.
    - internal diffusion + overlap caps
    - interconnect cap.
    - fan-out (gate cap)
- Increase transistor sizes
  - watch out for self-loading!
- Increase V_{DD} (??)
  - watch out for reliability issues!

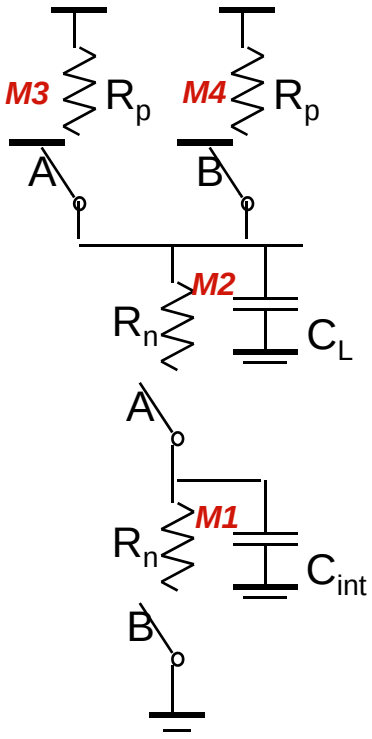
Designing Combinational Logic Circuits

Kaustav Banerjee

Equivalent Inverter

- CMOS gates: many paths to Vcc and Gnd
  - Multiple values for V_M, V_{IL}, V_{OL}, etc
  - Different delays for each input combination
- Equivalent inverter
  - Represent each gate as an inverter with appropriate device width
  - Include only transistors which are on or switching
  - Calculate V_M, delays, etc using inverter equations

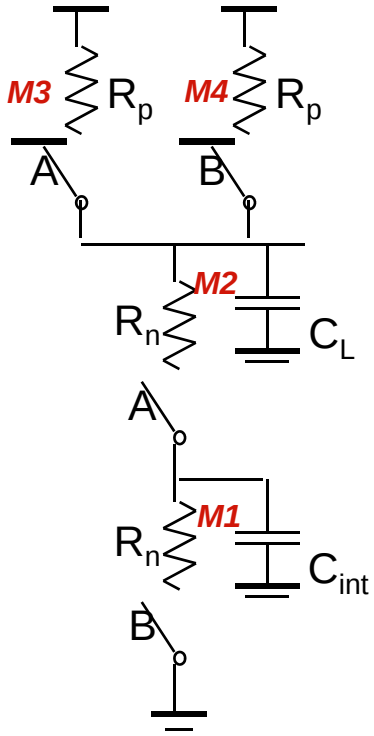
Input Pattern Effects on Delay



2-input NAND

- Delay is dependent on the pattern of inputs
- Low to high transition
  - both inputs go low
    - delay is 0.69 $R_p/2$ C_L
  - one input goes low
    - delay is 0.69 R_p C_L
- High to low transition
  - both inputs go high
    - delay is 0.69 $2R_n$ C_L

Delay Dependence on Input Patterns



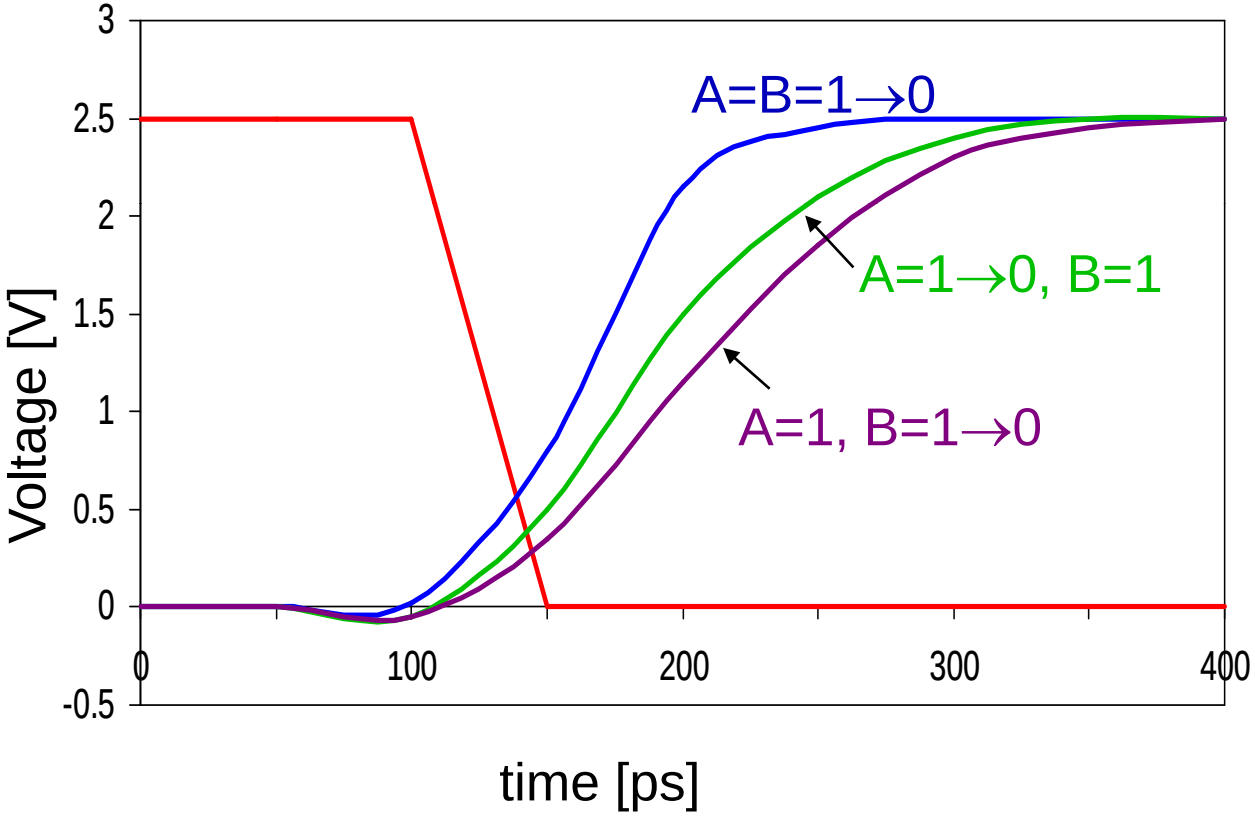
2-input NAND

- **High to Low Delay** depends on the initial state of the internal nodes
- Example: when both inputs transition from 0 to 1, it is important to know the state of the internal node (between M1 and M2)
- **Worst case**: when internal node is initially charged up to V_{DD}- V_{tn}. This can be ensured by pulsing the A input from 1-0-1, while B only makes the 0-1 transition.

Bottom Line: Delay estimation can be a fairly complex affair and requires careful consideration of the internal node caps. and data patterns

Delay Dependence on Input Patterns

Simulated Low to High delay for a 2-input NAND



Input Data Pattern	Delay (psec)
A=B=0→1	69
A=1, B=0→1	62
A= 0→1, B=1	50
A=B=1→0	35
A=1, B=1→0	76
A= 1→0, B=1	57

NMOS = 0.5μm/0.25 μm
PMOS = 0.75μm/0.25 μm
C_L = 100 fF

Note: worst case L-H delay depends on which input (A or B) is driven low (due to internal node cap of PDN stack: source of M2)

Transistor Sizing

- Sizing for switching threshold
  - All inputs switch together
- Sizing for delay
  - Find worst-case input combination
- Find equivalent inverter, use inverter analysis to set device sizes

Transistor Sizing

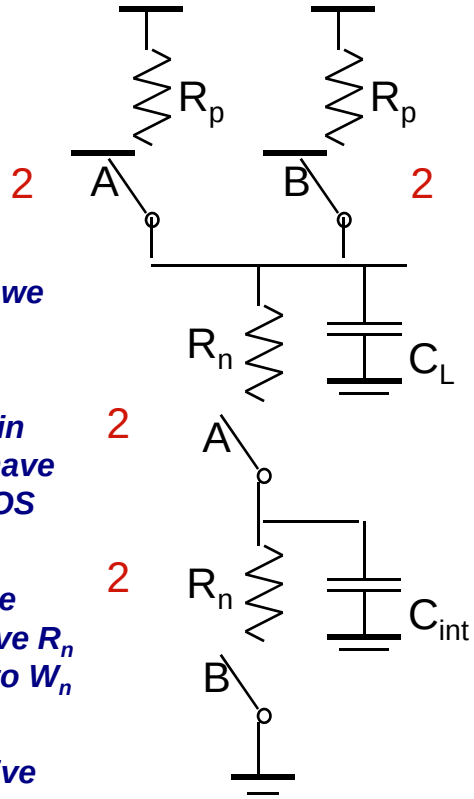
What sizing will lead to a 2:1 equivalent inverter?

For effective $R_p = R_n$, we need $W_p = 2W_n$

Since two NMOS are in series, each should have $R_n/2$, hence each NMOS size, $W_n = 2$

Each PMOS should be such that R_p = effective R_n (which corresponds to $W_n = 1$)

Hence, $W_p = 2$ (effective W_n) = 2



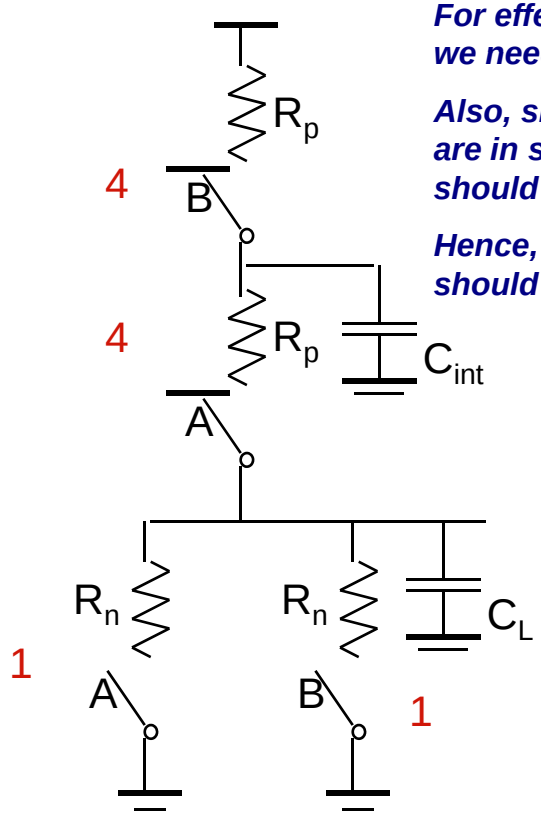
2-input NAND

For effective $R_p = R_n$, we need $W_p = 2W_n$

Also, since two PMOS are in series, each should have $R_p/2$

Hence, W_p of each should be $4W_n$

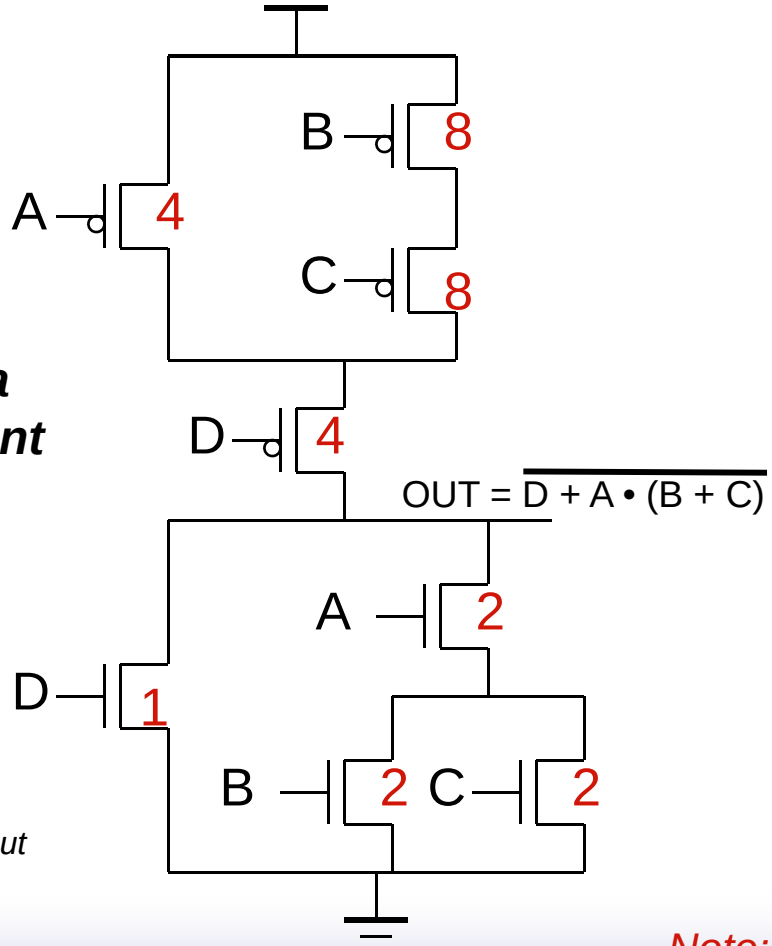
Avoid stacking PMOS transistors in series….



2-input NOR

Transistor Sizing a Complex CMOS Gate

What sizing will lead to a 2:1 equivalent inverter?



Note: A= 3 and B=C=D=6 will also give a 2:1 inverter but that will give higher output parasitic capacitance (due to larger D)

1. Start with a transistor in the PDN, that is (preferably) isolated (i.e., it can pull down the output node on its own)----D in this case
2. Assign a minimum size to this transistor (W=1 for D)
3. Now identify other paths in the PDN that can also discharge the output node: AB and AC in this case.
4. Size A, B, and C such that: each path will be electrically equivalent to the path through D (i.e., with same resistance), hence A, B and C should have W=2
5. Repeat the same for the PUN, keeping in mind that i) effective resistance of any path must equal the effective resistance of any path in the PDN, and ii) PMOS transistors will have to be sized (twice as large in this case) appropriately.

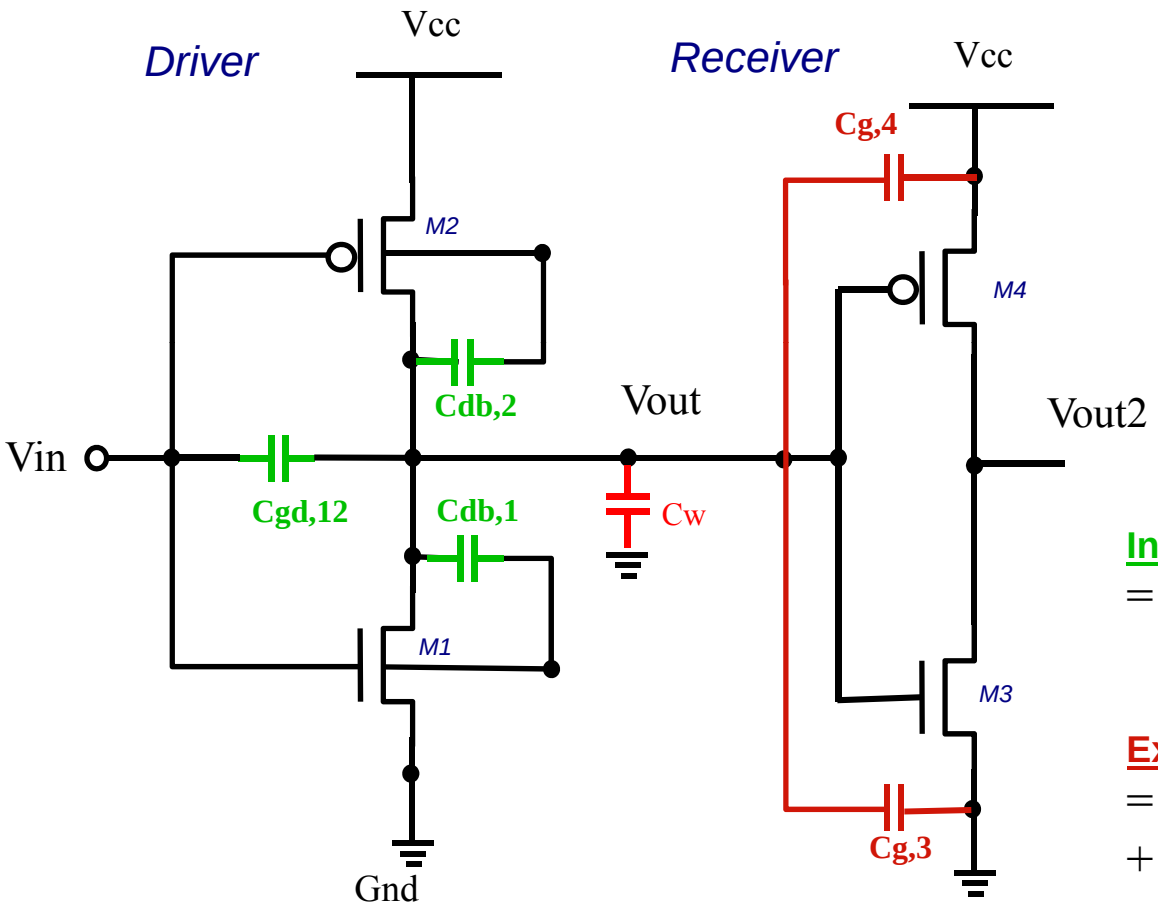
Note: here we ignored internal caps.

CMOS gate design: summary

- Designing a CMOS gate:
  - Find pulldown NMOS network from logic function or by inspection
  - Find pullup PMOS network
    - By inspection
    - Using logic function
    - Using dual network approach
  - Size transistors using equivalent inverter
    - Find worst-case pullup and pulldown paths
    - Size to meet rise/fall or threshold requirements

Inverter Chain Sizing

Load capacitances



$$C_L = C_{int} + C_{ext}$$

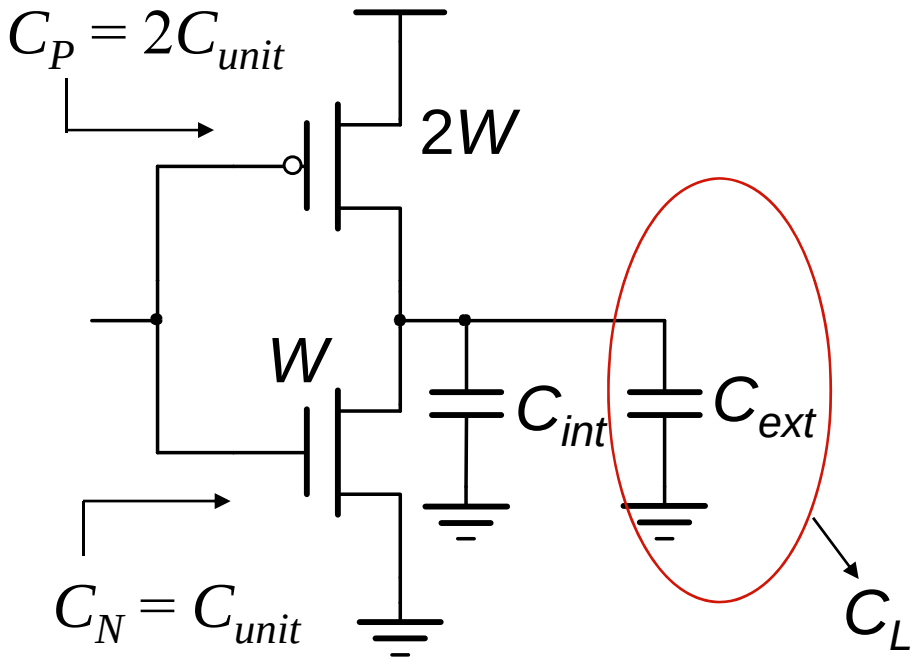
Internal Caps of Driver (Cint):

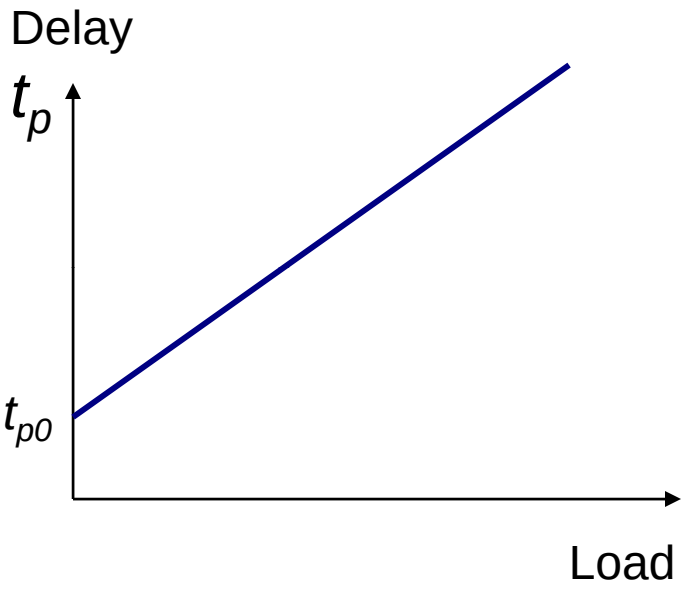
= Junction caps: $C_{db,12}$ +
Gate caps: $C_{gd,12}$ (including Miller Caps.)

External Caps (Cext):

= Interconnect cap: C_w
+ Receiver gate caps: $C_{g,43}$

Inverter with Load





Delay (t_p) = $kR_W(C_{int} + C_{ext}) = kR_WC_{int} + kR_WC_{ext} = \underbrace{kR_W C_{int}}_{t_{p0}\ (intrinsic\ delay)}(1+ C_{ext}/C_{int})$

Intrinsic delay of CMOS inverter

Let R_{eq} be the equivalent resistance of the gate (inverter), then delay (t_p) is defined as:

$$t_p = 0.69\, R_{eq} \left(C_{int} + C_{ext}\right)$$

$$= 0.69\, R_{eq}\, C_{int} \left(1 + \frac{C_{ext}}{C_{int}}\right)$$

$$= t_{p0} \left(1 + \frac{C_{ext}}{C_{int}}\right)$$

t_{p0} is the intrinsic delay

Impact of sizing on gate delay

Let S be the sizing factor

R_{ref} be the resistance of a reference gate (usually a minimum size gate)

C_{iref} be the internal capacitance of the reference gate

$$C_{\text{int}} = S\,C_{iref}\ , \quad R_{eq} = \frac{R_{ref}}{S}$$

$$t_p = 0.69\left(\frac{R_{ref}}{S}\right)\left(S\,C_{iref}\right)\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

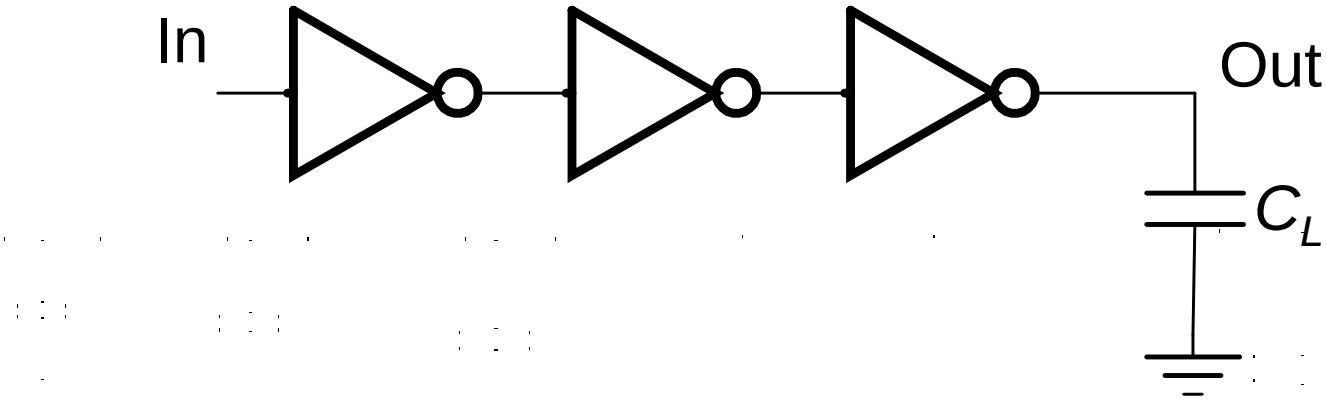
$$= 0.69\,R_{ref}\;C_{iref}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

$$= t_{p0}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

Hence:

1. *Intrinsic delay is independent of gate sizing, and is determined only by technology and inverter layout*
2. *If S is made very large, gate delay approaches the intrinsic value but increases the area significantly*

Inverter Chain



If C_L is given:
- How many stages are needed to minimize the delay?
- How to size the inverters?

May need some additional constraints.

Delay Formula: inverter chain

$$Delay \sim R_{eq}\left(C_{int} + C_{ext}\right)$$

$$t_p = 0.69\, R_{eq} C_{int}\left(1 + C_{ext} / \gamma C_{gin}\right) = t_{p0}\left(1 + f / \gamma\right)$$

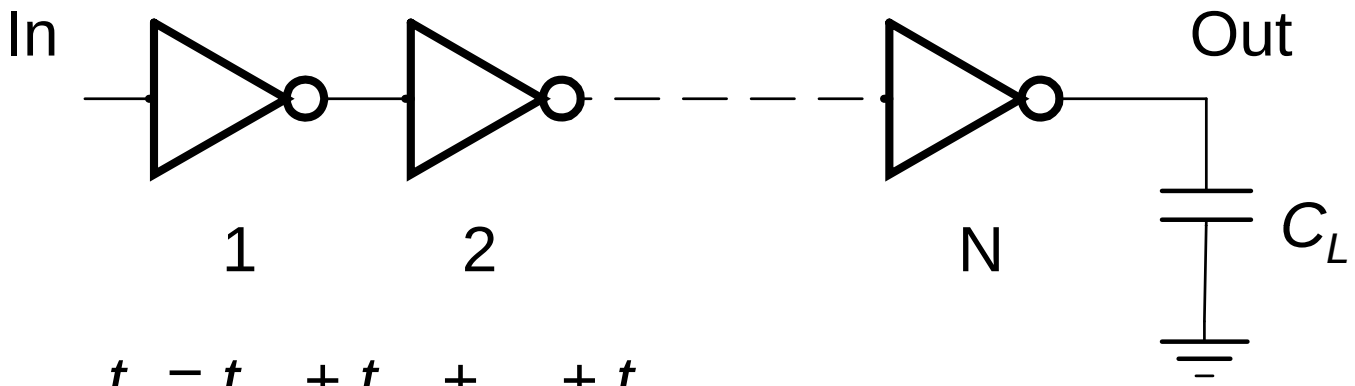
t_{p0}

relates the input gate cap. and the intrinsic output cap. of the inverter…

Let $C_{int} = \gamma C_{gin}$ with $\gamma \approx 1$

$f = C_{ext}/C_{gin}$ - effective fanout

Apply to Inverter Chain



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

$$t_{p,\,j} = t_{p0}\left(1 + \frac{C_{gin,\,j+1}}{\gamma C_{gin,\,j}}\right)$$

$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0}\sum_{j=1}^{N}\left(1 + \frac{C_{gin,\,j+1}}{\gamma C_{gin,\,j}}\right),\ C_{gin,N+1} = C_L$$

Optimal Tapering for Given N

Delay equation has *N* - 1 unknowns, $C_{g,2}$ …. $C_{g,N}$

Minimize the delay, find *N* - 1 partial derivatives and equate them to zero, or $\left(\partial t_p / \partial C_{g,j}\right) = 0$

Result: $C_{g,j+1}/C_{g,j} = C_{g,j}/C_{g,j-1}$ With j = 2,…..,N

Size of each stage is the geometric mean of two neighbors

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

- each stage has the same effective fanout ($f_j = f = C_{ext}/C_{g,j}$)
- hence, each stage has the same delay: $t_p = t_{p0}\,(1 + f/\gamma)$

Optimum Delay and Number of Stages

When each stage is sized by *f* and has same eff. fanout *f*:

$$\frac{C_L}{C_{g,N}} = \frac{C_{g,N}}{C_{g,N-1}} = \cdots\cdots\cdots = \frac{C_{g,2}}{C_{g,1}} = f$$

$$Hence,\ f^N = C_L / C_{g,1} = F$$

F is the overall effective fanout of the circuit

Effective fanout of each stage: $f = \sqrt[N]{F}$ ***If C_L and $C_{g,1}$ are known….***

Minimum path delay:

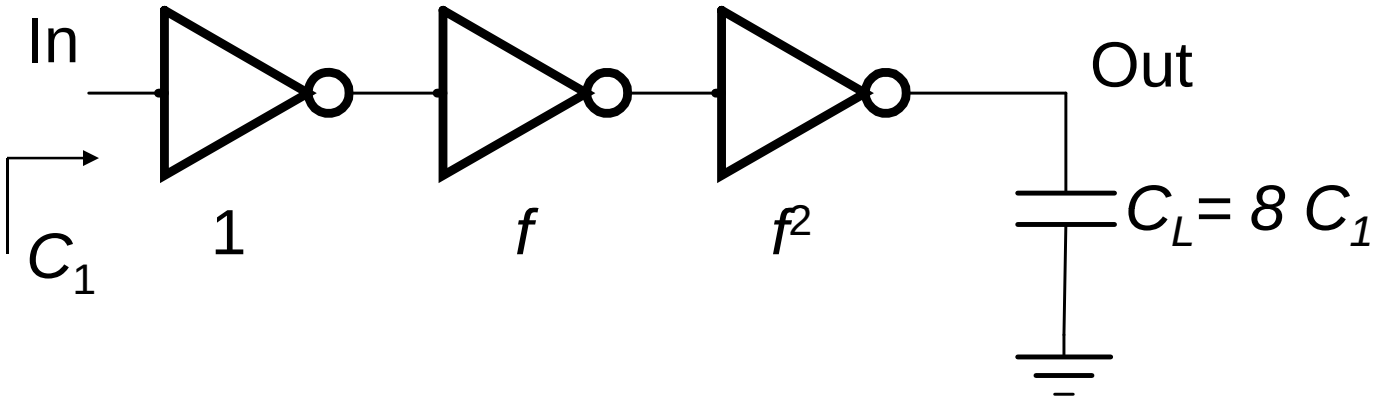
$$t_p = Nt_{p0}\left(1 + \sqrt[N]{F}/\gamma\right)$$

If N is too large, intrinsic delay of stages dominate, while if N is small, effective fanout of each stage (f) is large and the second term dominates

How to choose N?

Example

If N is given….



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

$$f = \sqrt[3]{8} = 2$$

Optimum Number of Stages

For a given load, C_L and given input capacitance C_{in}
Find optimal sizing f

$$C_L = F \cdot C_{in} = f^N C_{in} \quad with \quad N = \frac{\ln F}{\ln f}$$

$$t_p = N t_{p0}\left(F^{1/N}/\gamma + 1\right) = \frac{t_{p0} \ln F}{\gamma}\left(\frac{f}{\ln f} + \frac{\gamma}{\ln f}\right)$$

$$\frac{\partial t_p}{\partial f} = \frac{t_{p0} \ln F}{\gamma} \cdot \frac{\ln f - 1 - \gamma / f}{\ln^2 f} = 0$$

If self-loading is ignored….

For $\gamma = 0$, $f = e = 2.718$, $N = \ln F$

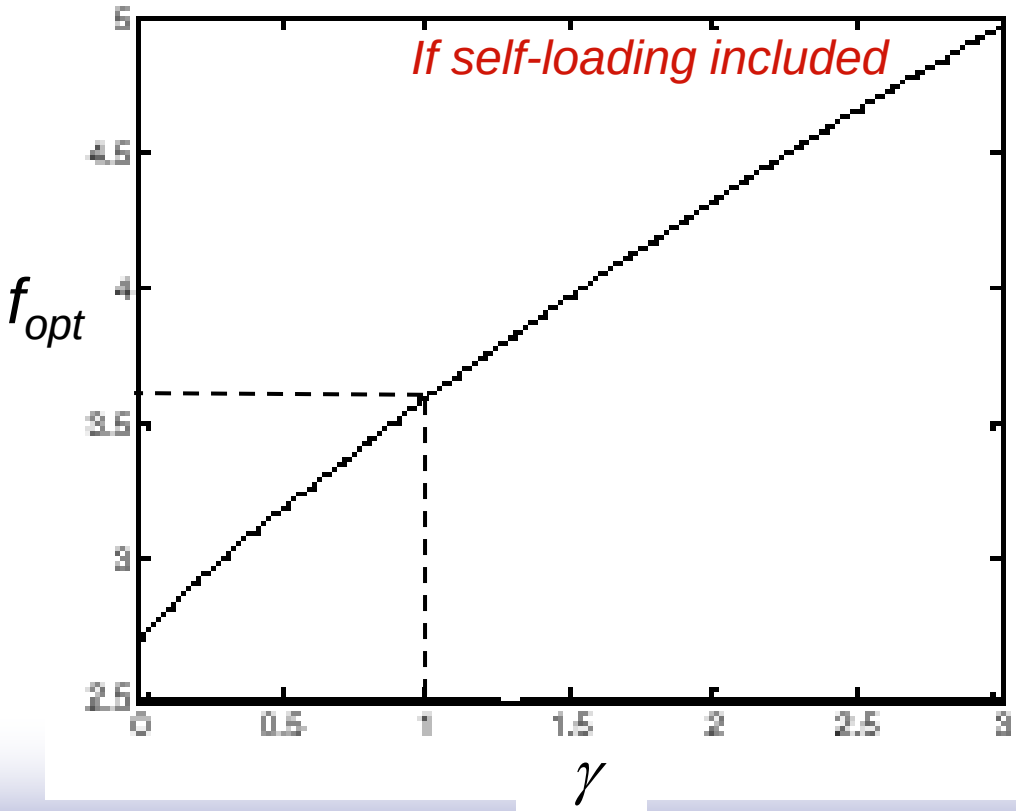
Otherwise….

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

Optimum Effective Fanout f

Optimum *f* for given process defined by γ

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

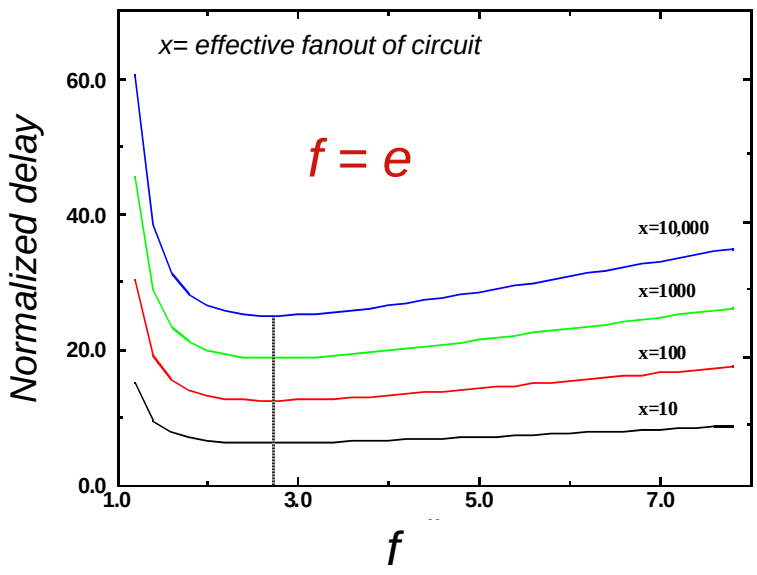


Optimum tapering factor:

f_{opt} = 3.6
for γ=1 (typical case)

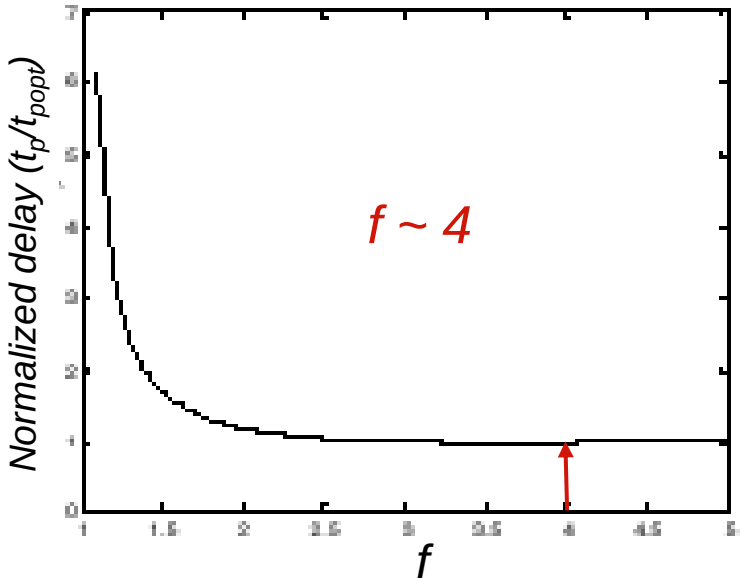
Impact of Self-Loading on tp

No Self-Loading, γ=0



Optimal number of stages, N= ln(F)

With Self-Loading γ=1



If $f<f_{opt}$ (too many stages) will result in delay to increase

Normalized delay function of F

$$t_p = Nt_{p0}\left(1 + \sqrt[N]{F}/\gamma\right)$$

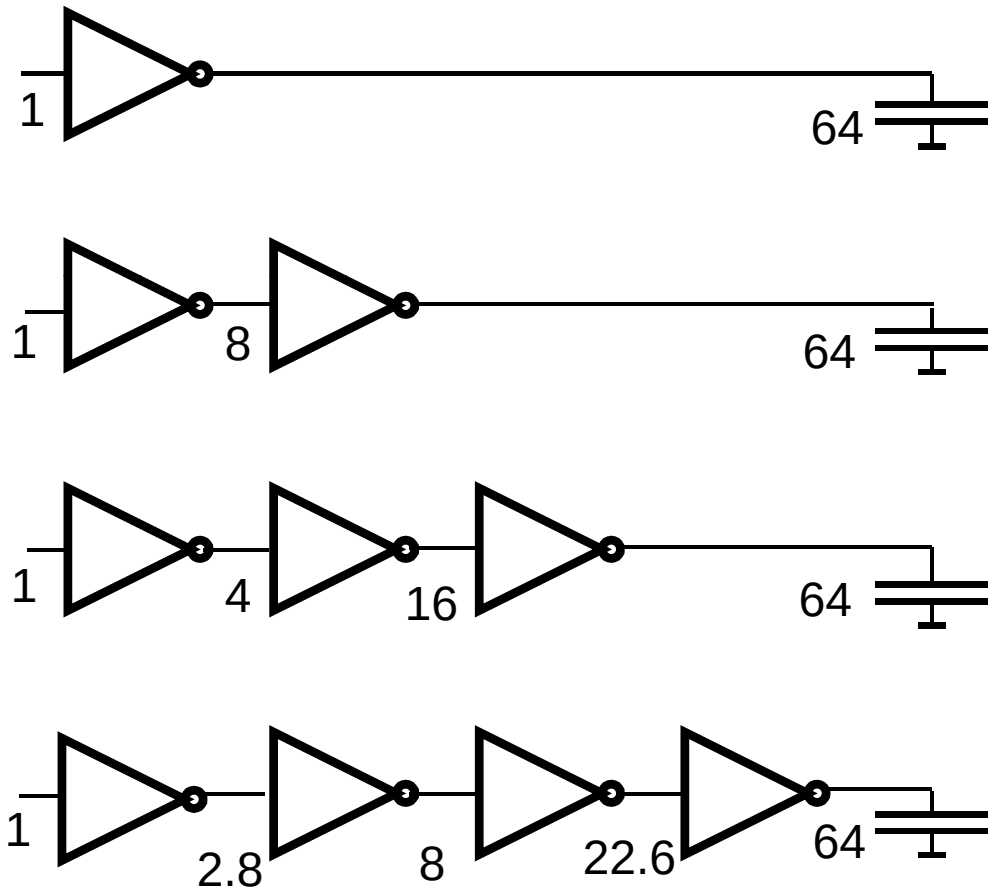
t_{popt}/t_{p0} for γ=1

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

As F increases, the differences between the unbuffered case (or two-stage buffer case) and the case of inverter chain increases…..

Buffer Design

$f = F^{1/N}$



N	f	t_p
1	64	65
2	8	18
3	4	15
4	2.8	15.3

Sizing Logic Paths for Speed

- Frequently, input capacitance of a logic path is constrained
- Logic also has to drive some capacitance
- Example: ALU load in an Intel's microprocessor is 0.5pF
- How do we size the ALU datapath to achieve maximum speed?
- We have already solved this for the inverter chain – can we generalize it for any type of logic?

Minimizing Delay in Complex Logic Networks

$$Delay = t_{p0}\left(1+\frac{f}{\gamma}\right)\ (inverter)$$

$$= t_{p0}\left(p+\frac{g\cdot f}{\gamma}\right)(Complex\ gate)$$

Everything Normalized w.r.t an inverter:
g_{inv} =1, p_{inv} = 1

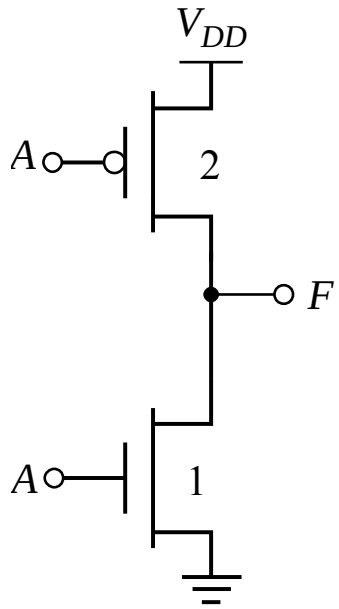
f – effective fanout *(ratio of external load and input cap. of gate)*
p – ratio of intrinsic delays of complex gate and inverter
(value increases with complexity of gate)
g – logical effort: how much more input capacitance is presented by the complex gate to deliver the same output current as an inverter (depends only on circuit topology)

Logical Effort

- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates
- Logical effort of a gate is the ratio of its input capacitance to the inverter capacitance when sized to deliver the same current
- Logical effort increases with gate complexity

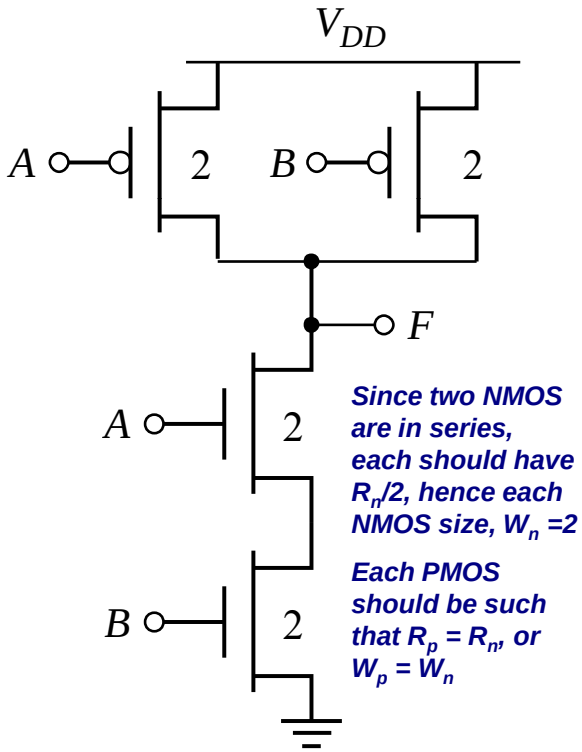
Logical Effort

Logical effort is the ratio of input capacitance of a gate to the input capacitance of an inverter with the same output current



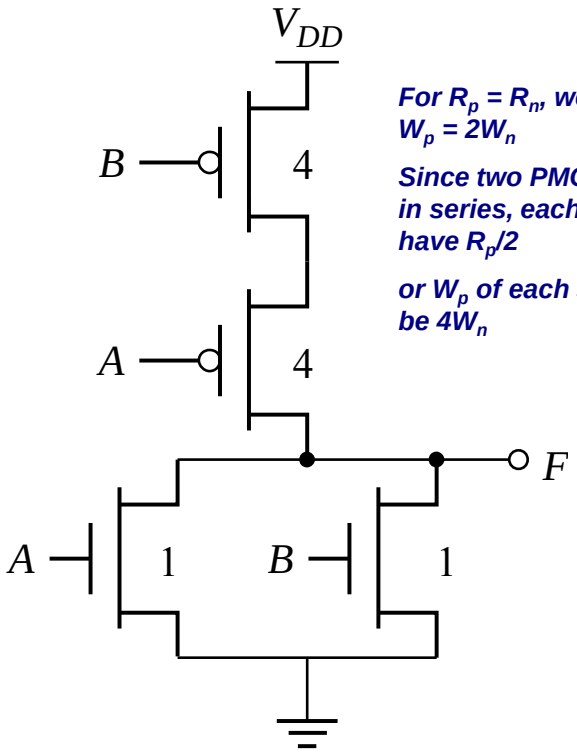
Inverter

$g = 1$



2-input NAND

$g = 4/3$



2-input NOR

$g = 5/3$

Delay in a Logic Gate

Gate delay:

$$d = h + p$$

effort delay **intrinsic delay**

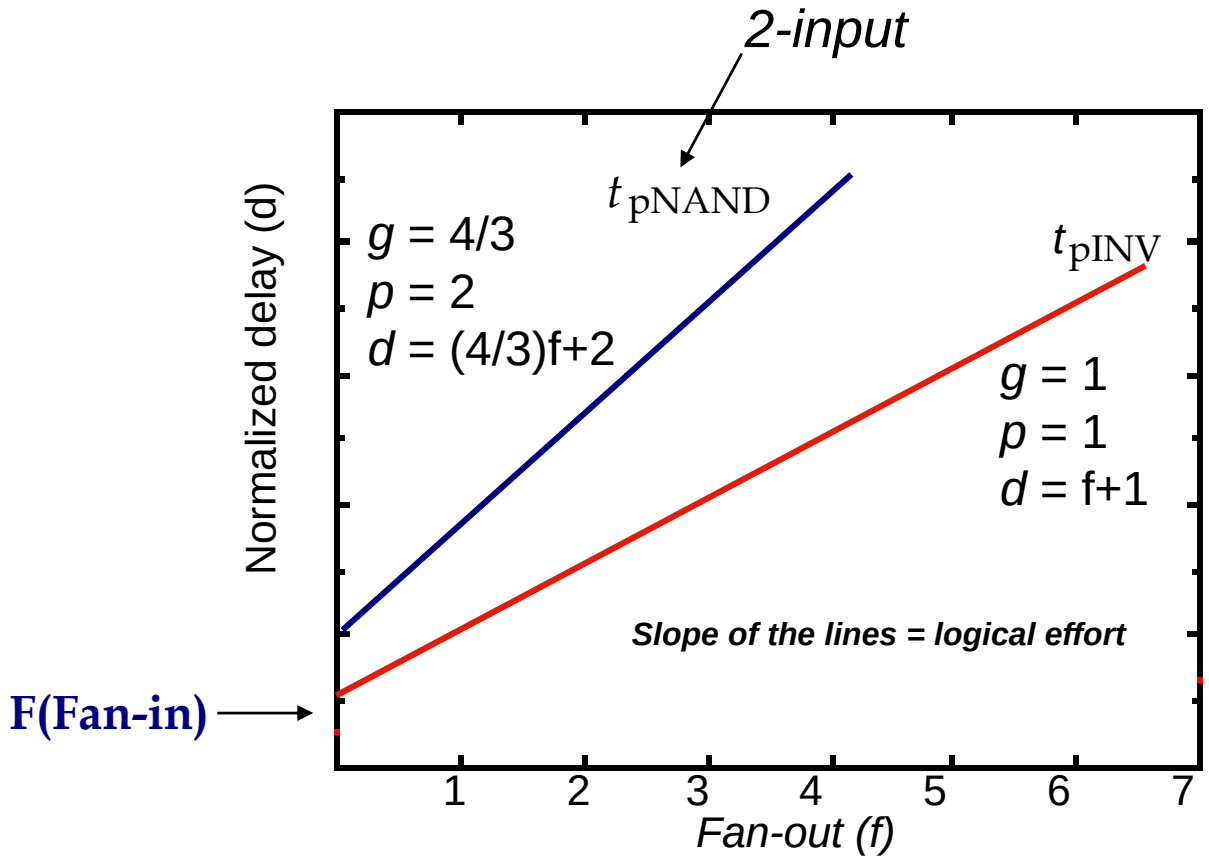
Effort delay (or gate effort):

$$h = g\,f$$

logical effort **effective fanout** $= C_{ext}/C_{in}$

- **Logical effort** is a function of topology, independent of sizing
- Effective fanout (**electrical effort**) is a function of load/gate size

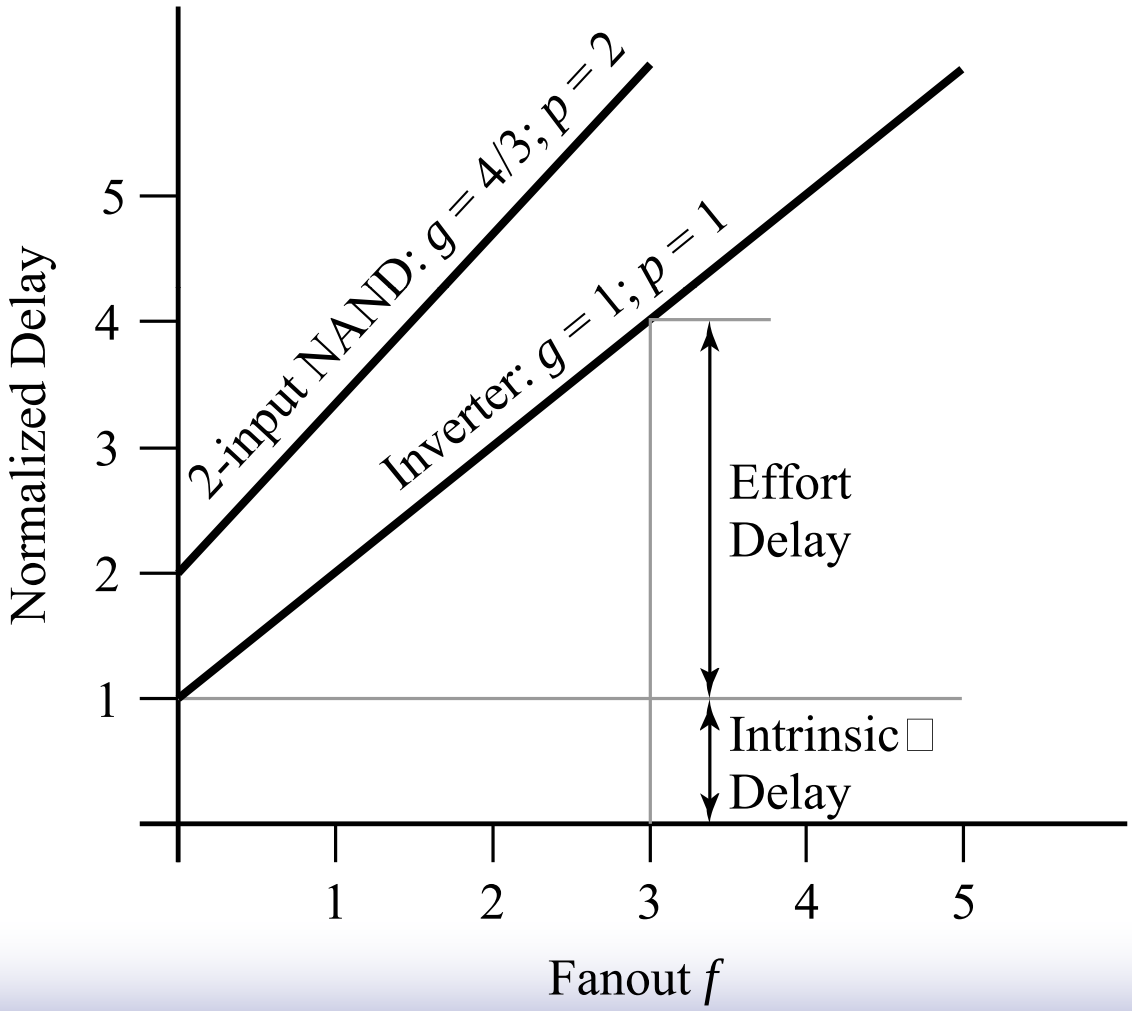
Logical Effort of Gates



$$d = p + g\,f$$

- **Delay can be adjusted by**:
  - *transistor sizing that changes the effective fanout*
  - *Choosing a gate with different g*

Logical Effort of Gates



Logical Effort

Gate Type	Number of Inputs			
	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

Total delay through a combinational logic block

$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0} \sum_{j=1}^{N} \left(p_j + \frac{f_j g_j}{\gamma} \right)$$

Similar to inverter chain delay….find N-1 partial derivatives and equate them to zero….

For minimal delay : $g_1 f_1 = g_2 f_2 = \ldots = g_N f_N$ (each stage should have the same gate effort, h)

$$Path\ Logic\ Effort = G = \prod_{1}^{N} g_i$$

$$Path\ Effective\ Fanout = F = \frac{C_L}{C_{g1}}$$

(or Path Electrical Effort)

Branching Effort

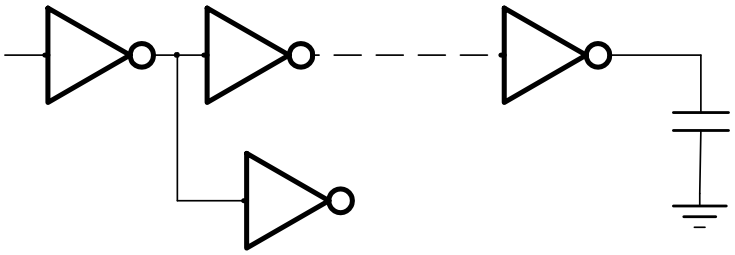
To relate F to the effective fanouts of the individual gates, one must account for the logical fanout within the network

When fanout occurs at the output of a node, some of the available drive current is directed along the path being analyzed

Branching effort of a logic gate:

$$b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$$

Load capacitance of the gate along the path under study



$$Path\ Branching\ Effort = B = \prod_{1}^{N} b_i$$

Total Path Effort

- Path electrical effort can be related to the electrical and branching efforts of the individual gates:

$$F = \prod_{1}^{N} \frac{f_i}{b_i} = \frac{\prod f_i}{B}$$

- Total path effort can be defined as:

$$H = \prod_{1}^{N} h_i = \prod_{1}^{N} g_i \; f_i = GFB$$

- Gate effort that minimizes the path delay = ?

- Minimum delay through path = ?

Multistage Networks

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

Gate effort: $h_i = g_i f_i$

Path electrical effort: $F = C_L / C_{gin}$

Path logical effort: $G = g_1 g_2 \ldots g_N$

Path branching effort: $B = b_1 b_2 \ldots b_N$

*Path effort: $H = GFB$

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

Optimal Number of Stages

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

$$D = NH^{1/N} + Np_{inv}$$

$$\frac{\partial D}{\partial N} = -H^{1/N}\ln\left(H^{1/N}\right) + H^{1/N} + p_{inv} = 0$$

Substitute 'best gate effort': $h = H^{1/N}$ ⇒ *Gate effort that minimizes path delay*

A path achieves least delay by using N = $\log_4 H$ stages

Optimum Effort per Stage

When each stage bears the same effort:

$$h^N = H$$

$$h = \sqrt[N]{H}$$

gate efforts: $g_1 f_1 = g_2 f_2 = \ldots = g_N f_N$

Effective fanout of each gate: $f_i = h / g_i$

Minimum path delay:

$$D = t_{p0}\left(\sum_{j=1}^{N} p_j + \frac{N\left(\sqrt[N]{H}\right)}{\gamma} \right)$$

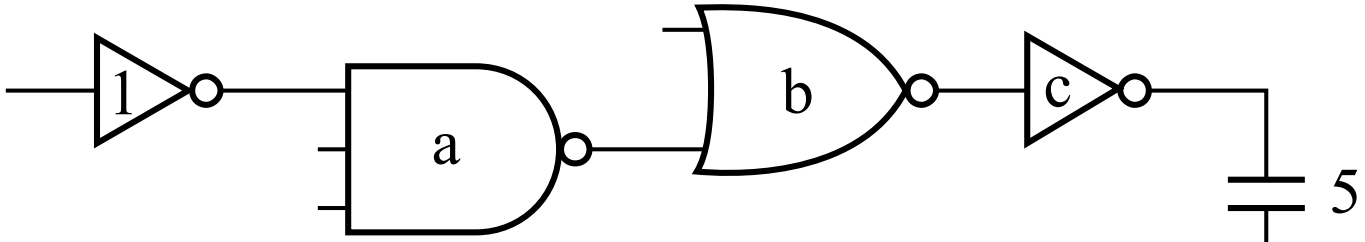
Sizing of Chain of Gates

- Consider chain s_i
- Sizing factors for each gate in the chain can be derived by working out from front to end (or vice versa).
- Assume that a unit-size gate has a driving capability equal to a minimum-size inverter
- Hence, $C_{gin} = g\, C_{in\_ref}$
- If s_1 is the sizing factor for gate 1:
  - $C_{g1} = s_1\, g_1\, C_{in\_ref}$
  - Input capacitance of gate 2 is larger by f_1/b_1:

    That is, $C_{g2} = f_1/b_1\, C_{g1} = s_2\, g_2\, C_{in\_ref}$
  - For gate i in the chain:

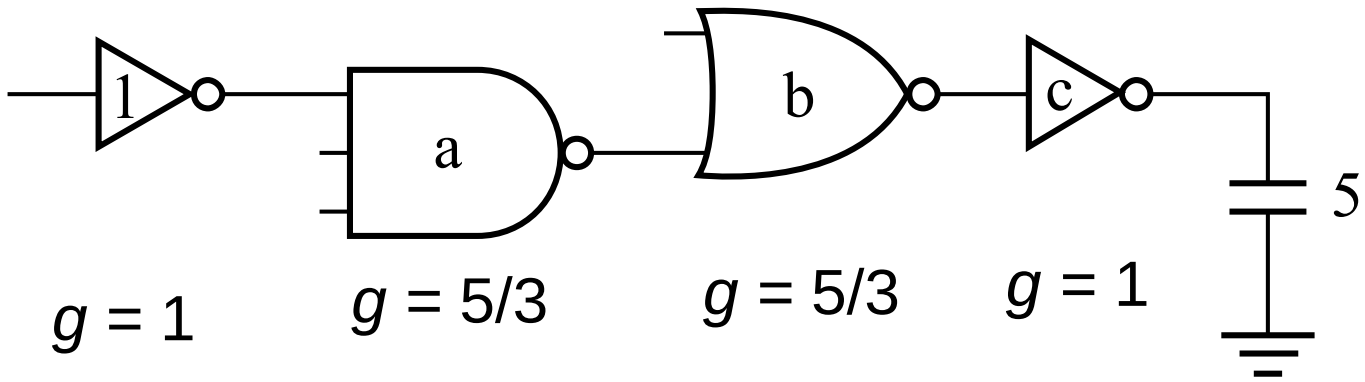
$$s_i = \left(\frac{g_1 s_1}{g_i}\right)\prod_{j=1}^{i-1}\left(\frac{f_j}{b_j}\right)$$

Example: Optimize Path



$g = 1$	$g = 5/3$	$g = 5/3$	$g = 1$
$f = a$	$f = b/a$	$f = c/b$	$f = 5/c$

Example: Optimize Path



Effective fanout, F = 5/1 = 5
G = 1 x 5/3 x 5/3 x 1 = 25/9
B=1 (no branching)
H = GFB = 125/9 = 13.9
h = $H^{1/4}$ =1.93 (optimal gate effort)

Derive Fanout Factors (taking gate types into account):
f1 = 1.93 (since f=gh)
f2 = 1.93 (*3/5)* = 1.16
f3 = 1.16
f4=1.93

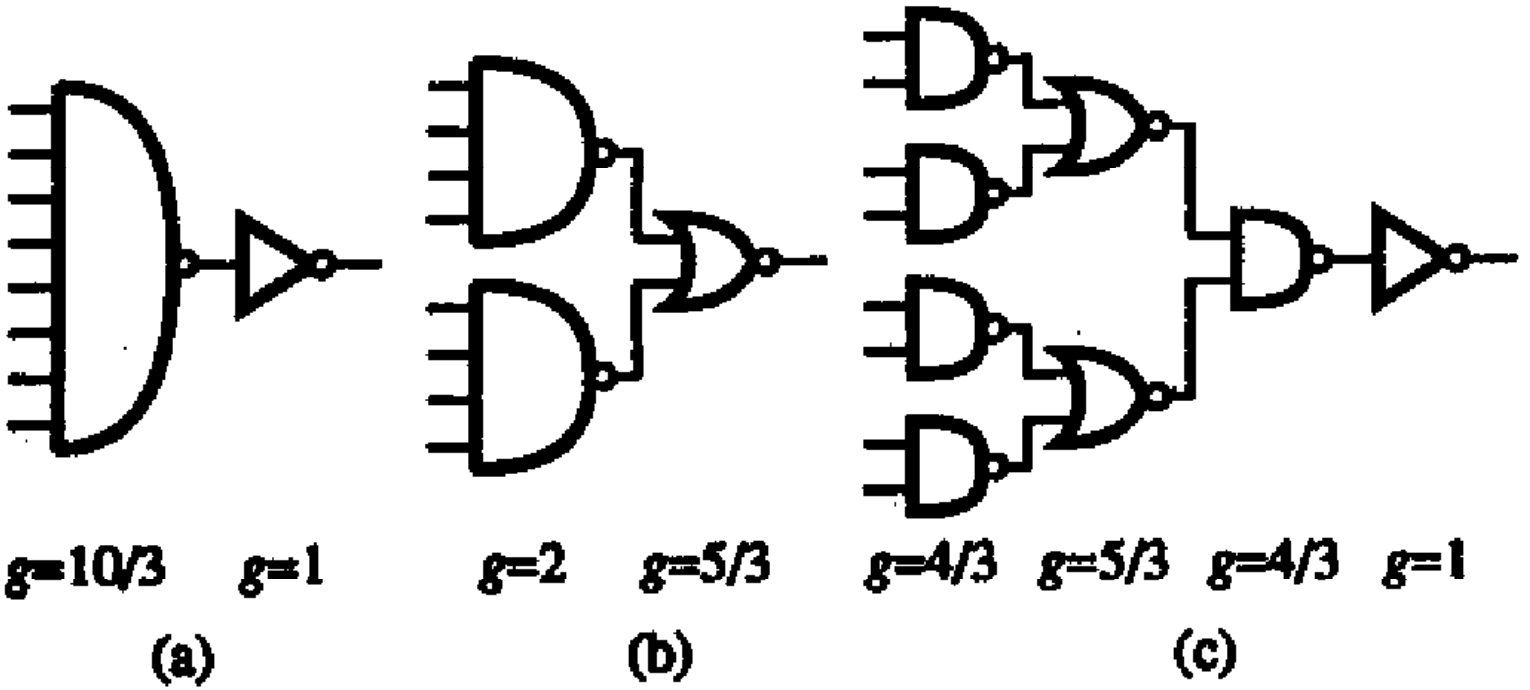
Derive Gate Sizes:

$a = f_1 g_1 / g_2 = 1.16$

$b = f_1 f_2 g_1 / g_3 = 1.34$

$c = f_1 f_2 f_3 g_1 / g_4 = 2.6$

Example – 8-input AND



Method of Logical Effort

- Compute the path effort: $H = GFB$
- Find the best number of stages $N \sim \log_4 H$
- Compute the stage effort $h = H^{1/N}$
- Sketch the path with this number of stages
- Work from either end, find sizes:
  $C_{in} = C_{out} * g/h$

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

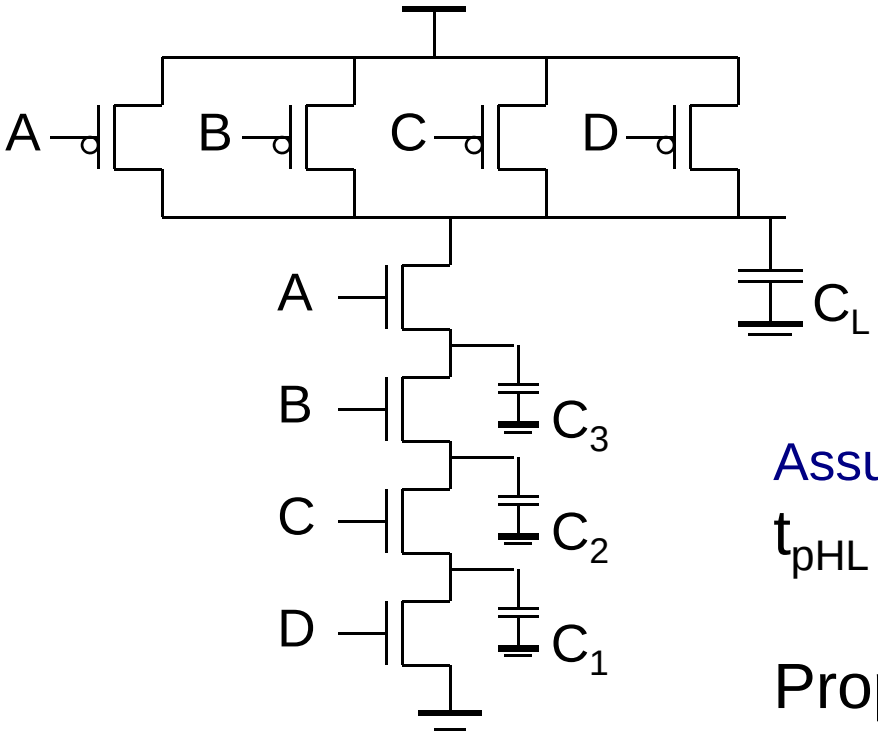
Summary

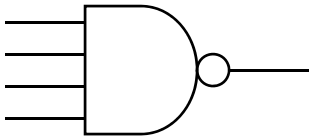
Table 4: Key Definitions of Logical Effort

Term	Stage expression	Path expression
Logical effort	g	$G = \prod g_i$
Electrical effort	$f = \frac{C_{out}}{C_{in}}$	$F = \frac{C_{out\ (path)}}{C_{in\ (path)}}$
Branching effort	n/a	$B = \prod b_i$
Effort	$h = gf$	$H = GFB$
Effort delay	h	$D_H = \sum h_i$
Number of stages	1	N
Parasitic delay	p	$P = \sum p_i$
Delay	$d = h + p$	$D = D_H + P$

Sutherland, Sproull and Harris

Fan-In Considerations





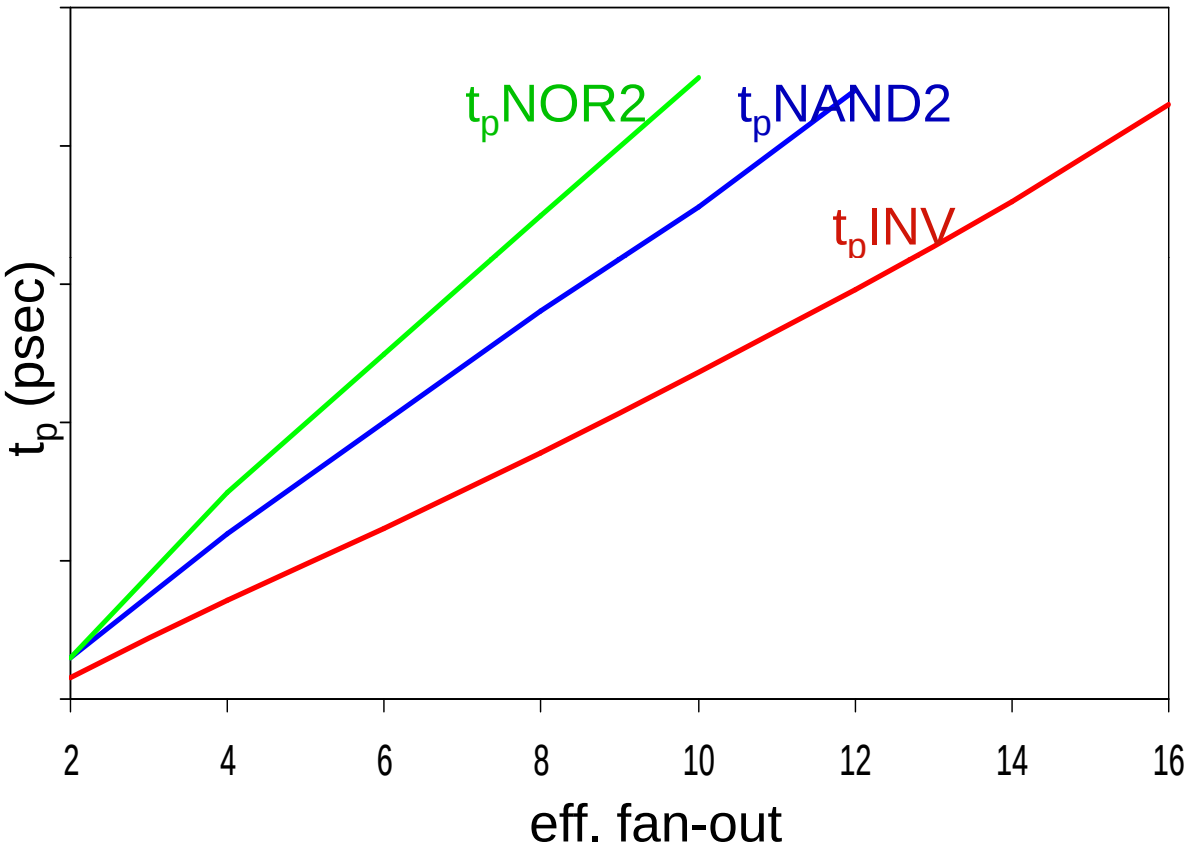
Distributed RC model
(Elmore delay)

Assuming all NMOS are equal size

$t_{pHL} = 0.69\ R_{eqn}(C_1+2C_2+3C_3+4C_L)$

Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

t_p as a Function of Fan-Out



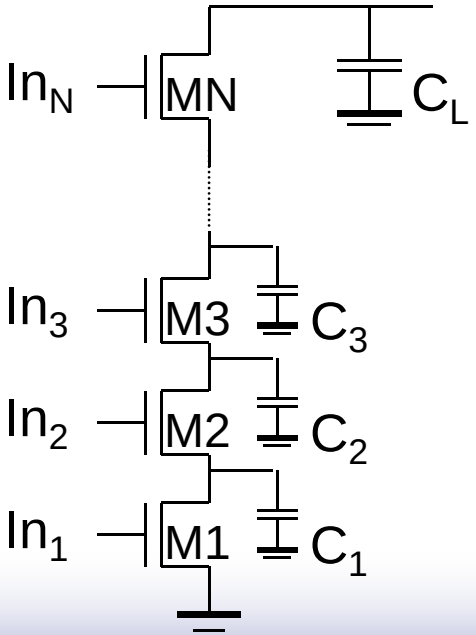
All gates have the same drive current.

Slope is a function of "driving strength"

Note: i) these lines are not going through the origin, ii) NAND and NOR have same intrinsic delay….

Fast Complex Gates: Design Technique 1

- Transistor sizing
  - as long as fan-out capacitance dominates
- Progressive sizing (not so simple to layout!)



Distributed RC line

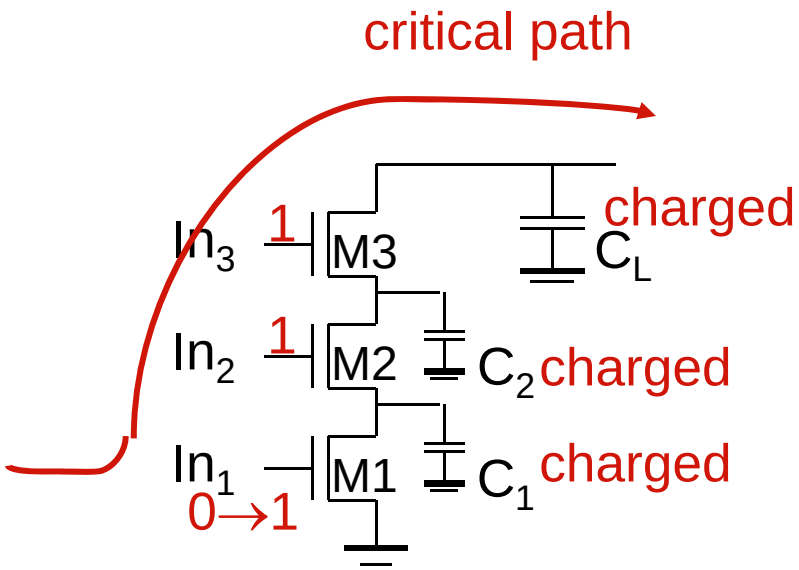
M1 > M2 > M3 > … > MN
(the transistor closest to the output is the smallest: has biggest R)

Can reduce delay by more than 20%; decreasing gains as technology shrinks

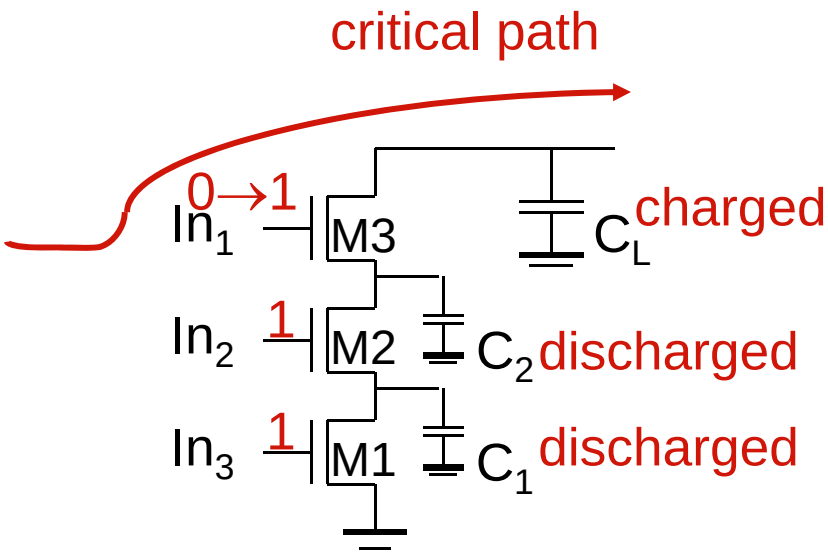
Fast Complex Gates: Design Technique 2

- Transistor ordering

Critical path is determined by the slowest arriving signal: In1



delay determined by time to discharge C_L, C_1 and C_2



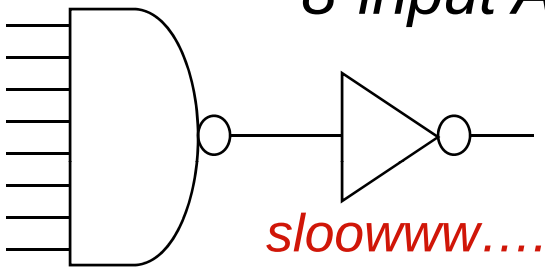
delay determined by time to discharge C_L

Fast Complex Gates: Design Technique 3

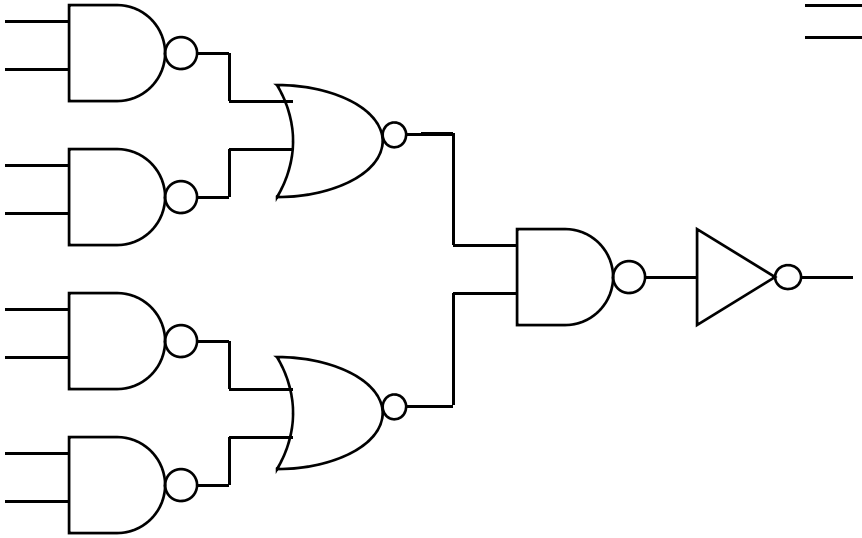
- Alternative logic structures

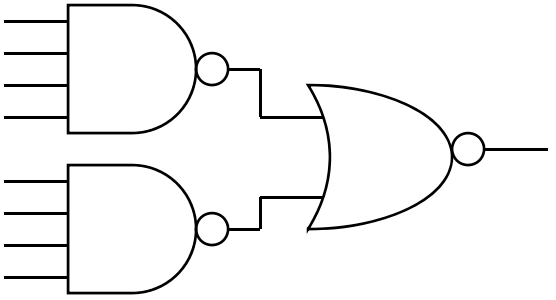
F = ABCDEFGH

8-input AND



sloowww….

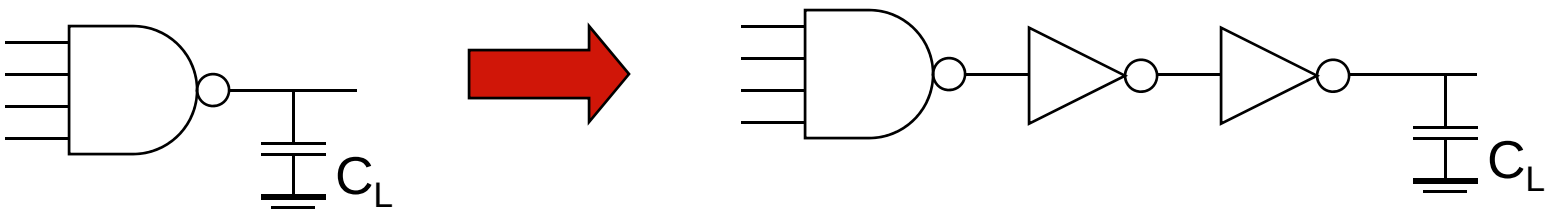




Using De Morgan's Law…

Fast Complex Gates: Design Technique 4

- Isolating fan-in from fan-out using buffer insertion



Fast Complex Gates: Design Technique 5

- Reducing the voltage swing

$$t_{pHL} = 0.69\ (3/4\ (C_L\ V_{DD})/\ I_{DSATn}\ )$$

$$= 0.69\ (3/4\ (C_L\ V_{swing})/\ I_{DSATn}\ )$$

  - linear reduction in delay
  - also reduces power consumption
- But the following gate is much slower!
- Or requires use of "sense amplifiers" on the receiving end to restore the signal level (memory design)