ECE 225
High-Speed Digital IC Design

Lecture 3

Review of Basic Circuit Styles

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

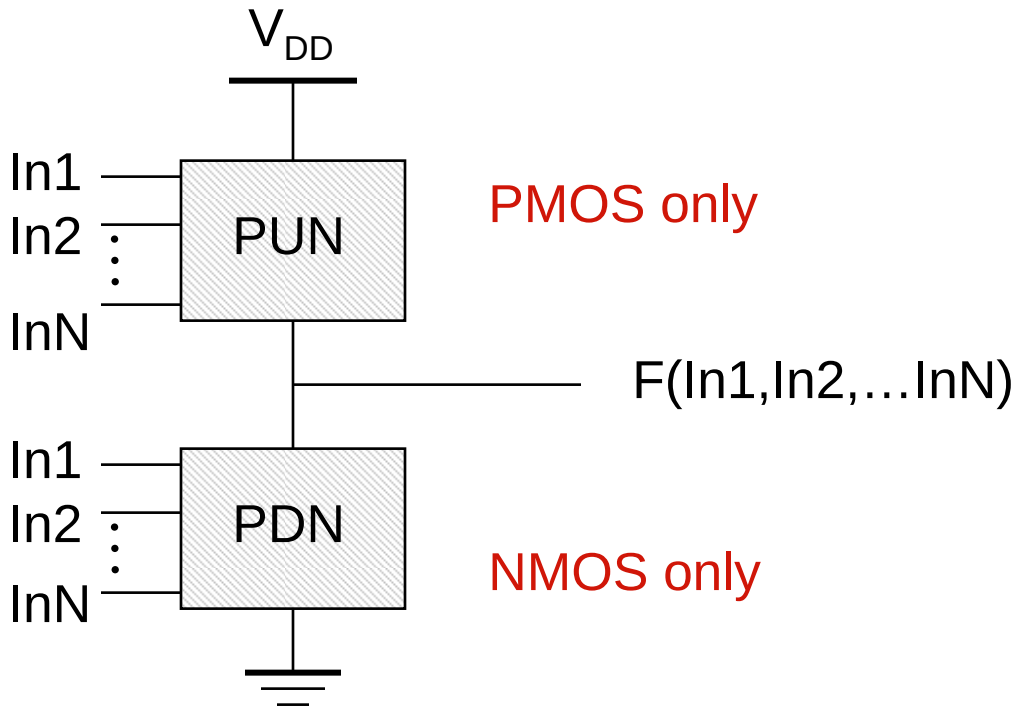
Which style is best?

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

Complementary CMOS Logic

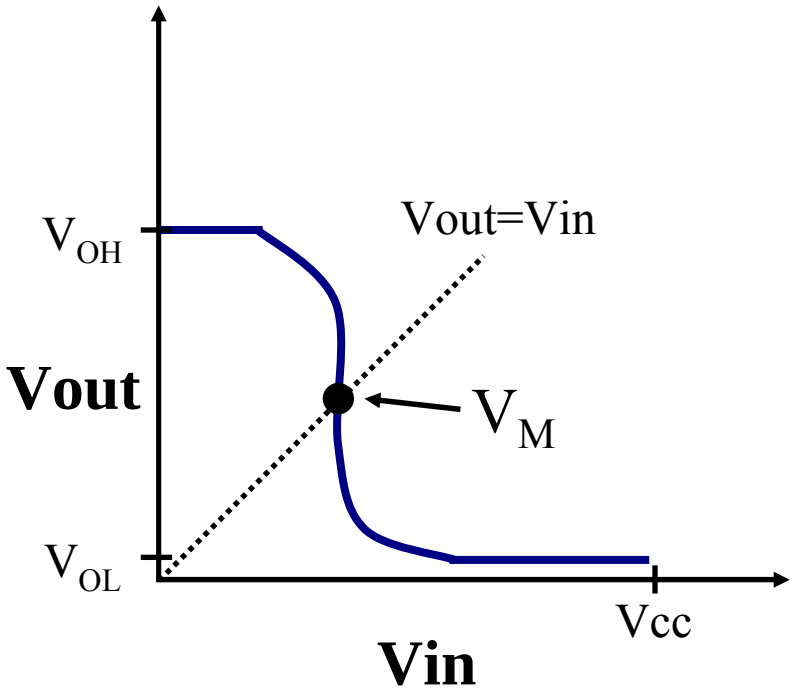
- Full rail-to-rail swing; high noise margins
- Logic levels not dependent upon the relative device sizes; ratioless
- Always a path to Vdd or Gnd in steady state; low output impedance
- Extremely high input resistance; nearly zero steady-state input current
- No direct path steady state between power 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

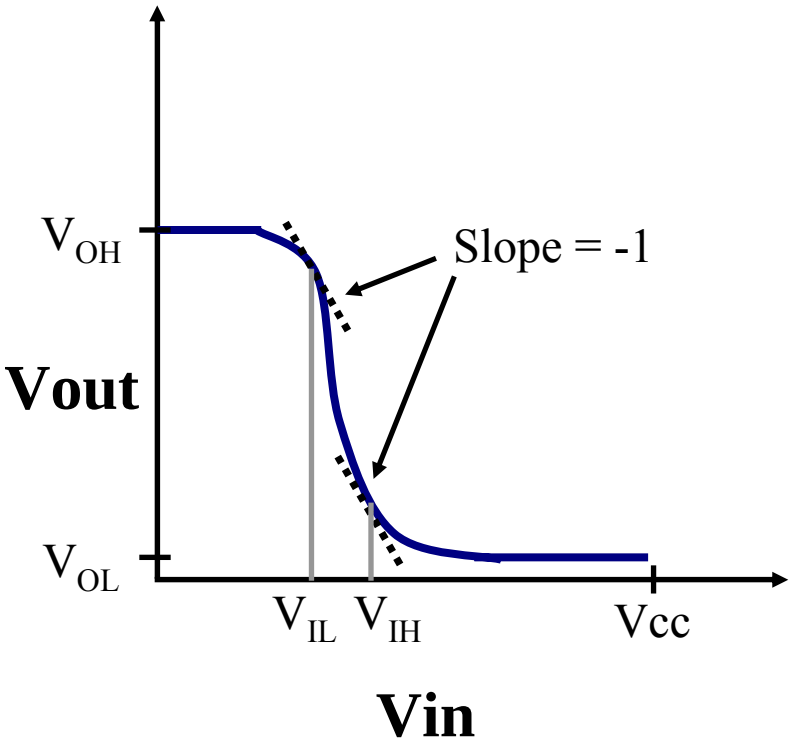
Inverter Threshold



Inverter switching threshold:

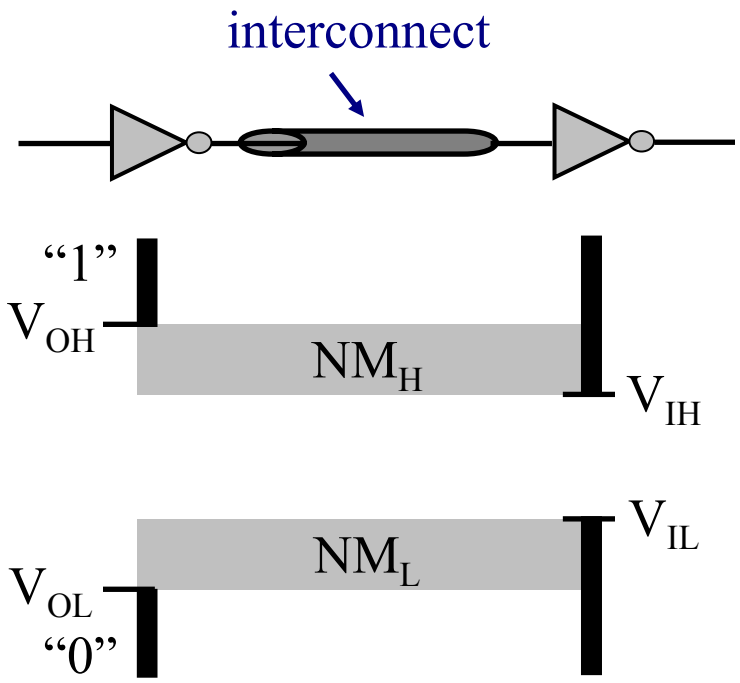
- Point where voltage transfer curve intersects line Vout=Vin
- Represents the point at which the inverter switches state
- Normally, $V_M \approx Vcc/2$

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

Noise Margin (cont)

- Why are V_{IL}, V_{IH} defined as unity-gain point on VTC curve?
  - Assume there is noise on input voltage V_{in}

$$V_{out} = f(V_{in} + V_{noise})$$

  - First-order approximation (Taylor Series):

$$V_{out} = f(V_{in}) + \frac{dV_{out}}{dV_{in}} V_{noise}$$

Note: **$dV_{out}/dV_{in} = 0$ occurs only at the beginning and at the end of the VTC curve, elsewhere it is negative**

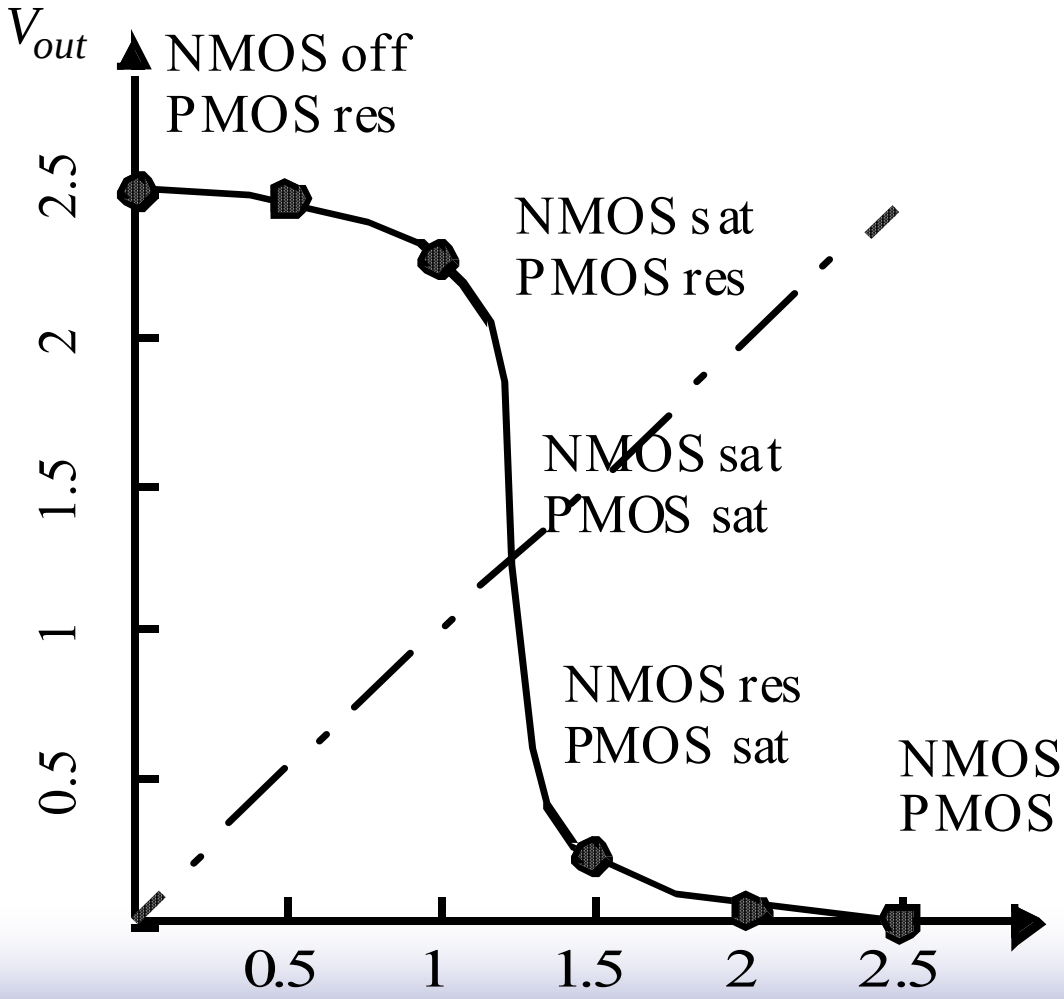
  - If gain (dV_{out}/dV_{in}) > 1, noise will be amplified.
  - If gain < 1, noise is filtered. Therefore V_{IL}, V_{IH} ensure that gain < 1

CMOS Inverter Operation (summary)

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

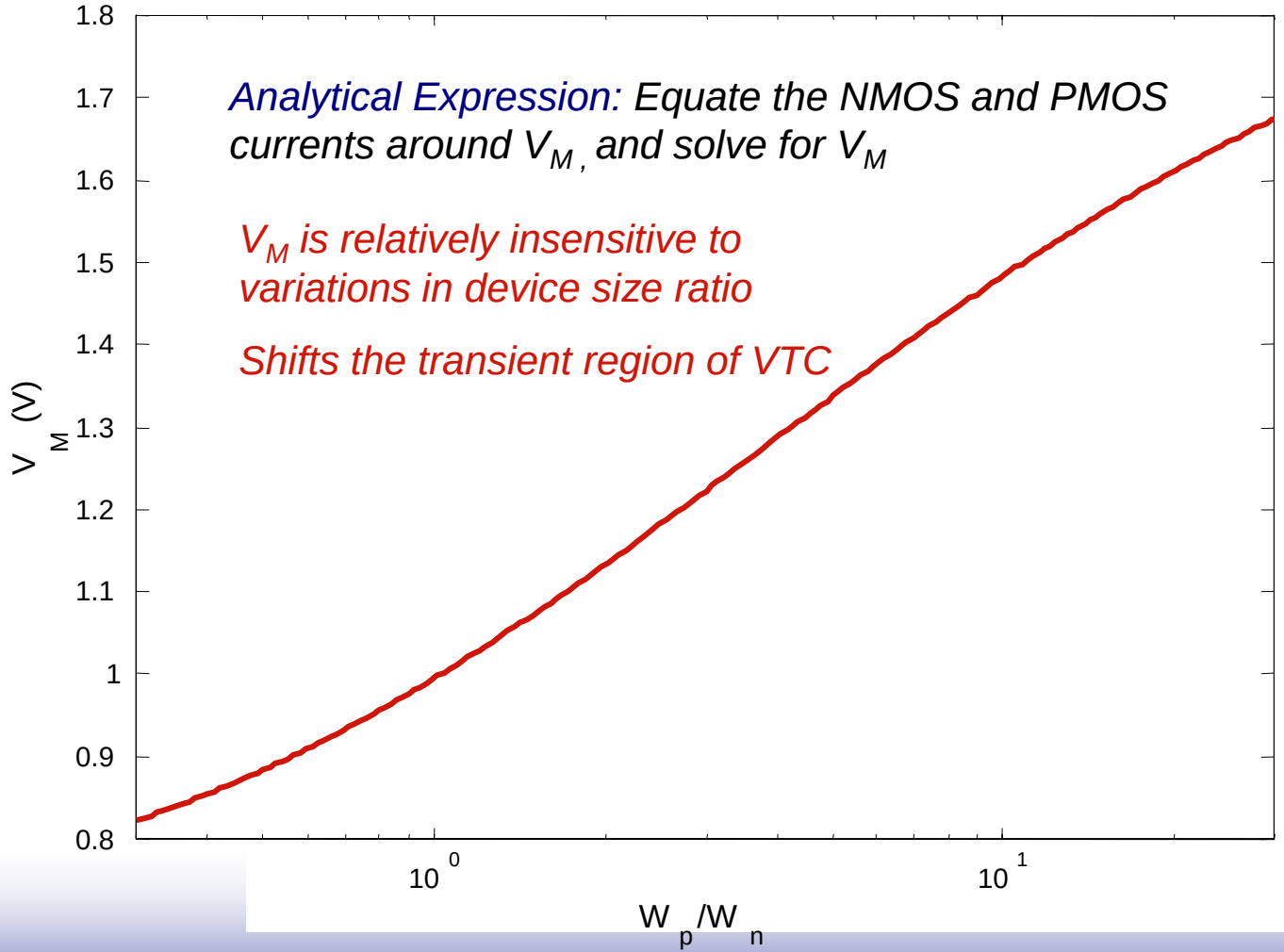
	Cutoff	Linear	Saturated
nMOS	$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$	$V_{gsn} > V_{tn}$
	$V_{in} < V_{tn}$	$V_{in} > V_{tn}$	$V_{in} > V_{tn}$
		$V_{dsn} < V_{gsn} - V_{tn}$	$V_{dsn} > V_{gsn} - V_{tn}$
		$V_{out} < V_{in} - V_{tn}$	$V_{out} > V_{in} - V_{tn}$
pMOS	$V_{gsp} > V_{tp}$	$V_{gsp} < V_{tp}$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

CMOS Inverter VTC



In which state do the transistors spend most of their time?

Switching Threshold as a function of Transistor Ratio



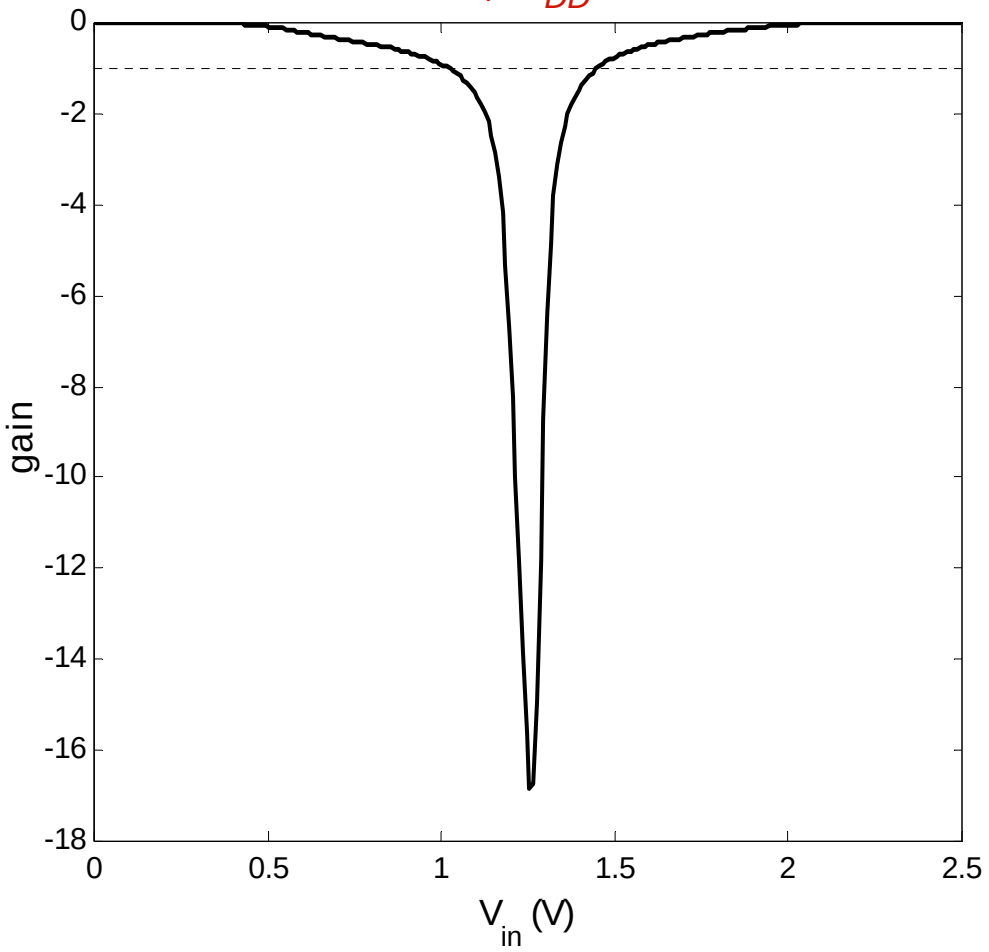
Analytical Expression: Equate the NMOS and PMOS currents around V_M, and solve for V_M

V_M is relatively insensitive to variations in device size ratio

Shifts the transient region of VTC

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

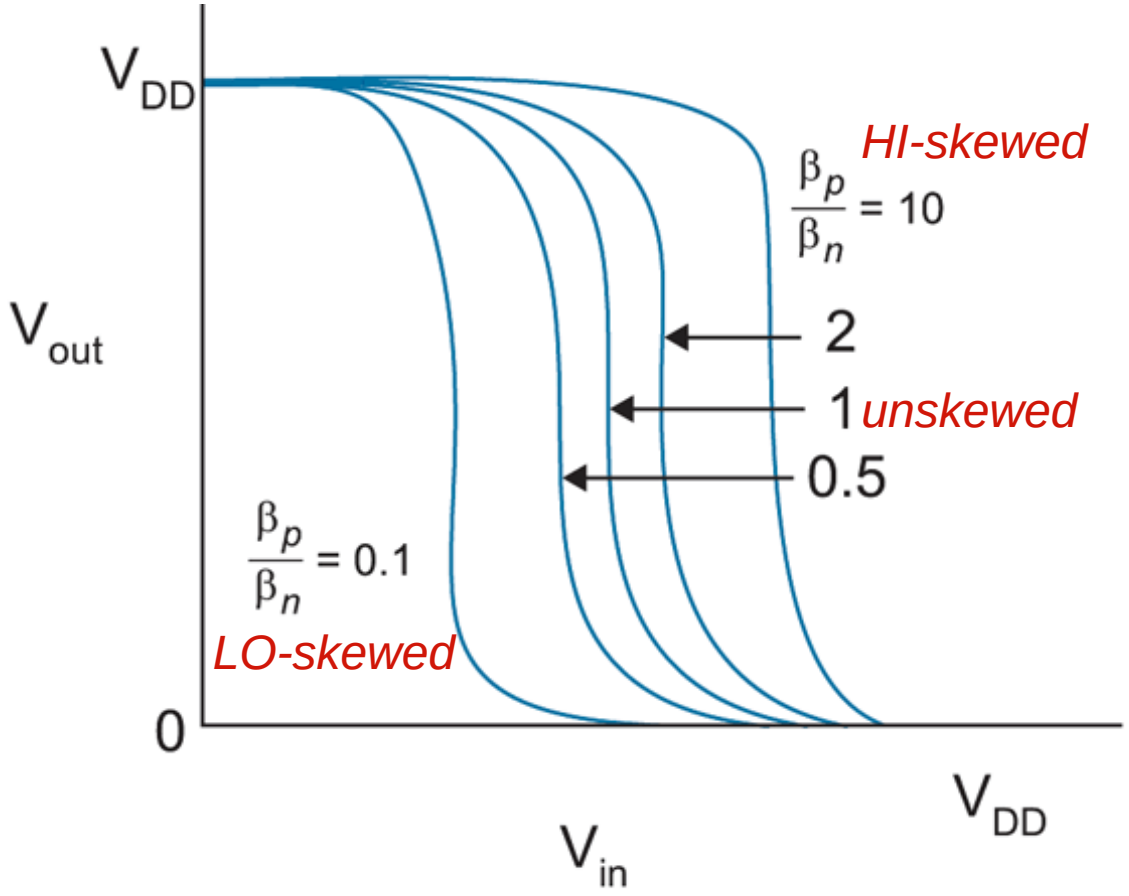
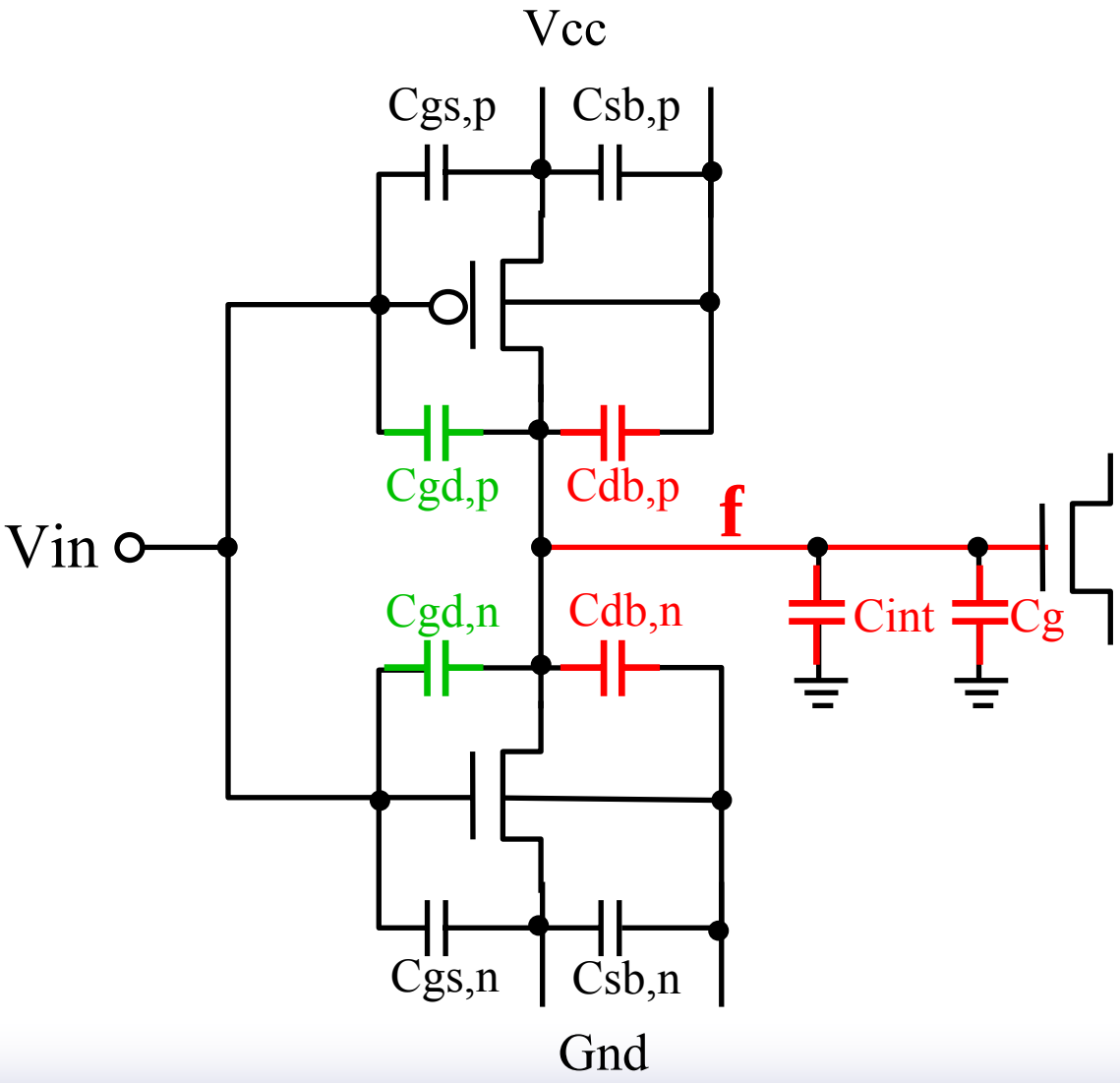


FIG 2.26 Transfer characteristics of skewed inverters

Propagation Delay

CMOS inverter capacitances

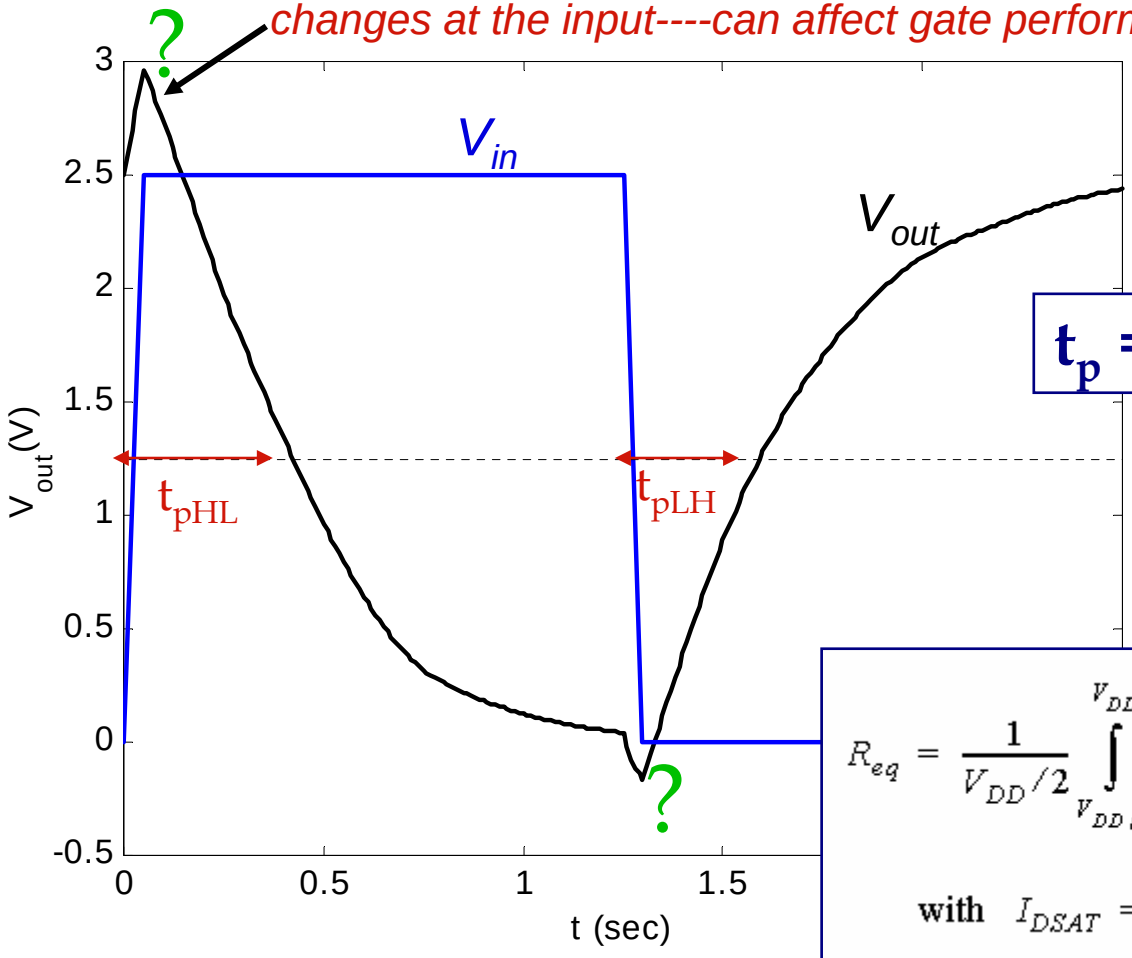


Cap on Node f:

- 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

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 $t_{pHL}=t_{pLH}$

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

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

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

Design for Performance

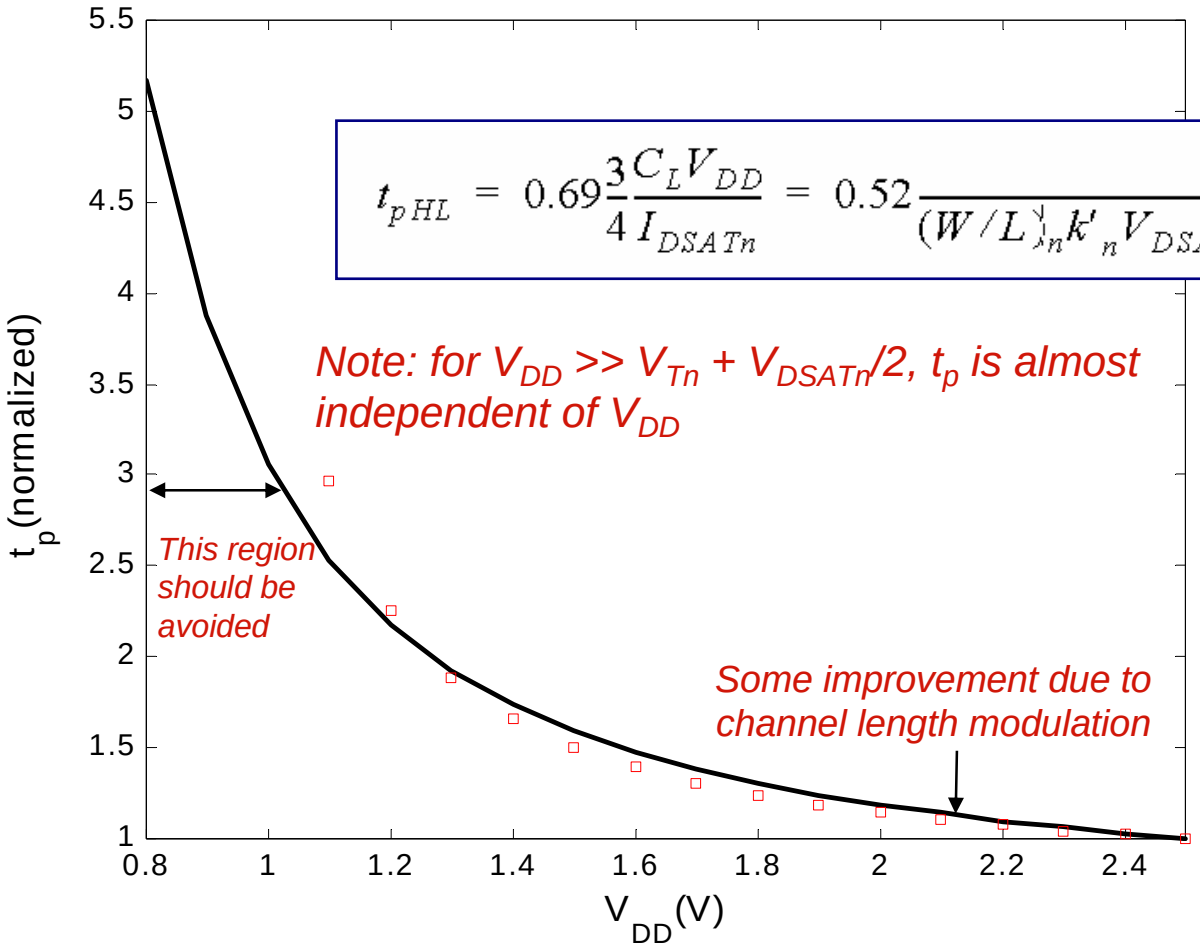
- Keep load capacitances (C_L) small
  - Recall that three major components contribute to the load cap.
    - internal diffusion + overlap caps
    - interconnect cap.
    - fan-out (gate cap)
- Increase transistor sizes
  - watch out for self-loading!
- Increase V_{DD} (??)
  - watch out for reliability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

Delay as a function of V_{DD}

Same as the ON resistance of a transistor….

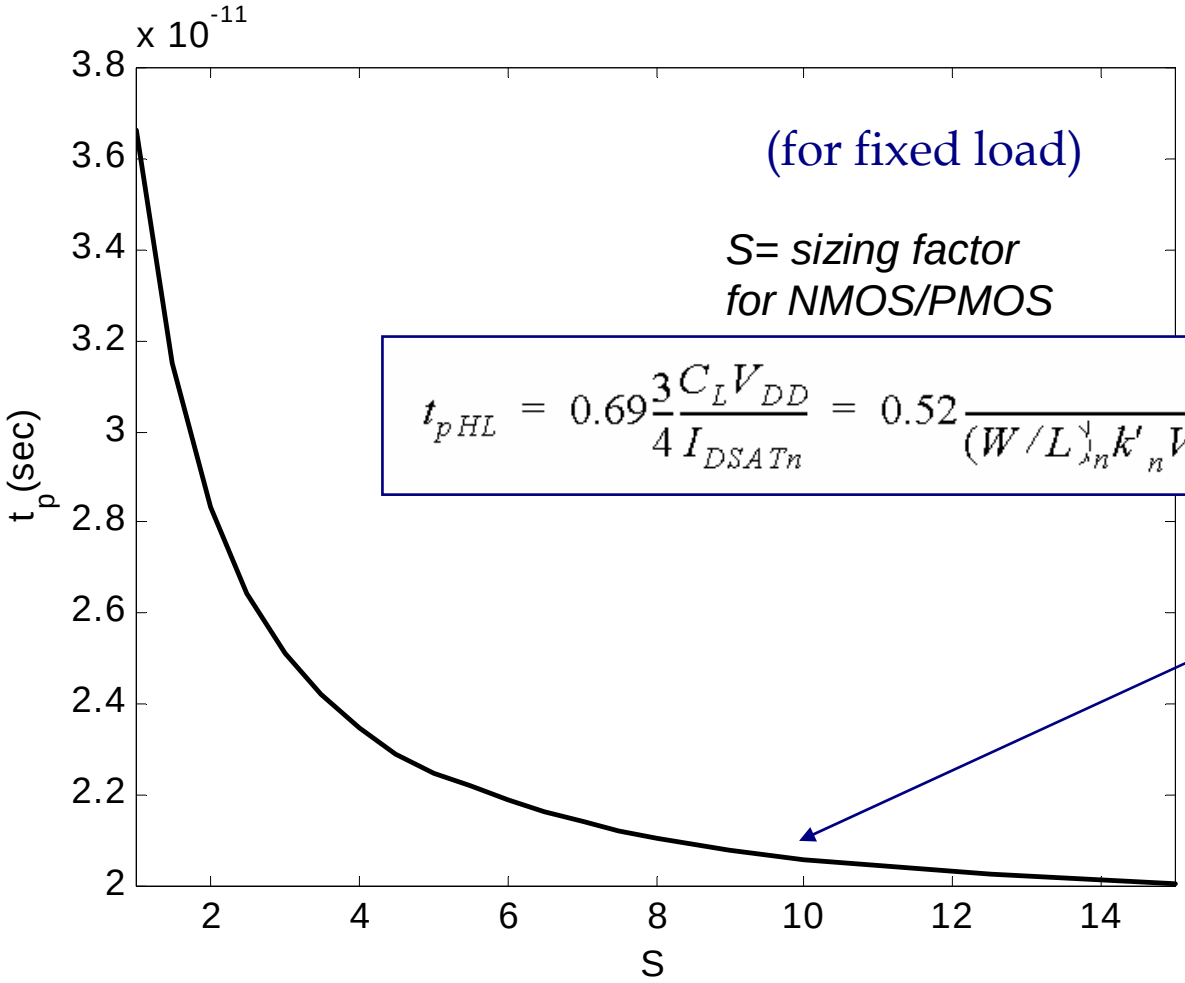
Trade off energy dissipation vs performance…..

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



Reliability concerns at high V_{DD}….

Device Sizing



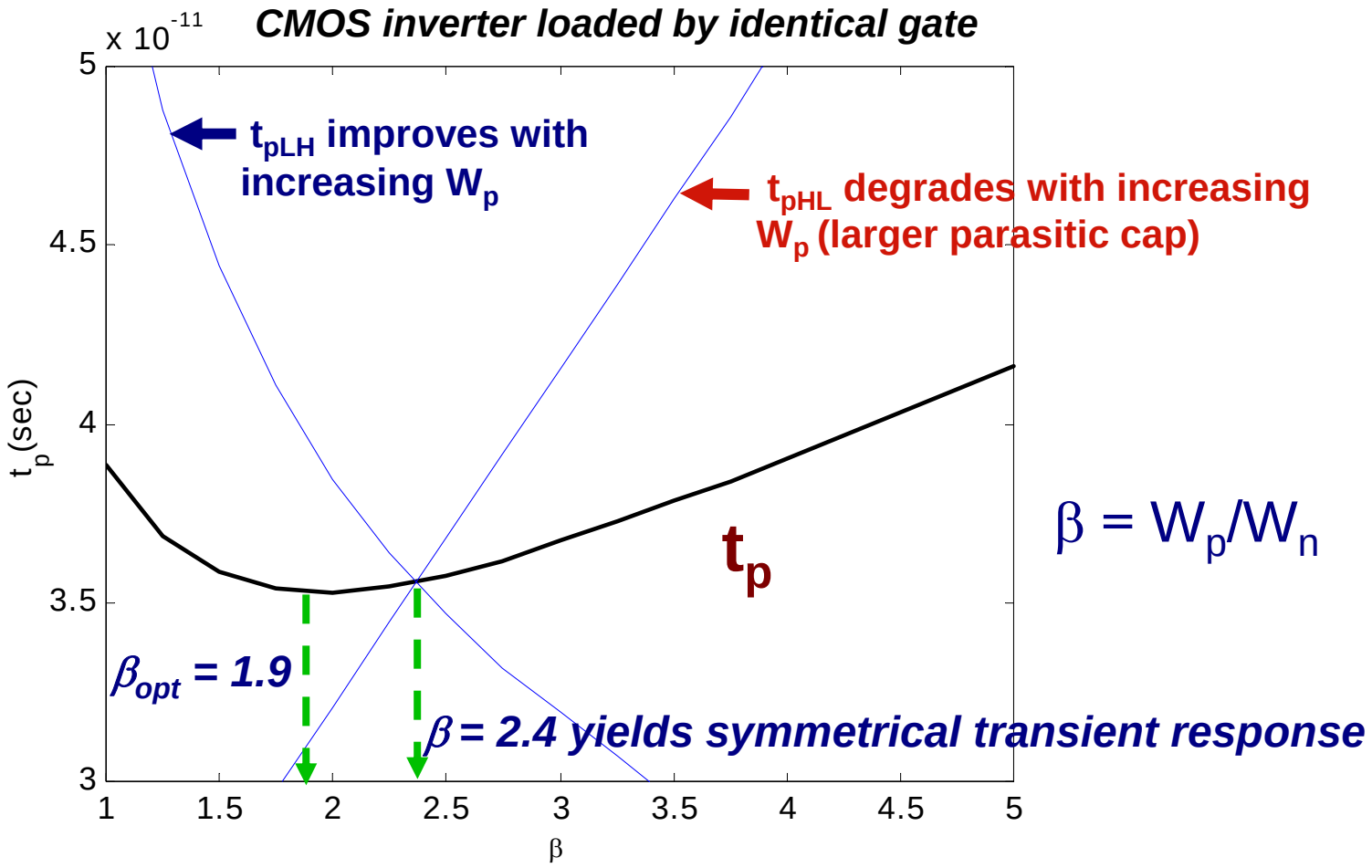
(for fixed load)

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 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….

Designing Combinational Logic Circuits

Analysis of CMOS gates

- Represent "on" transistors as resistors

- Transistors in series → resistances in series
  - Effective resistance = 2R
  - Effective width = ½ W

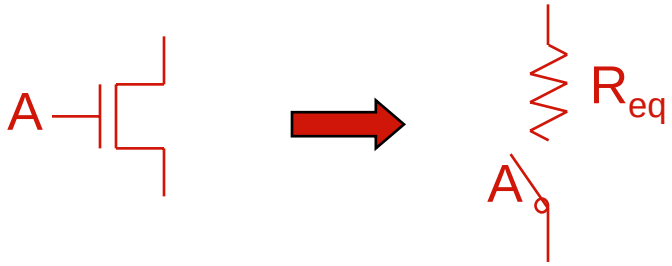
Note: 1) As transistor width increases, its on resistance (R) decreases

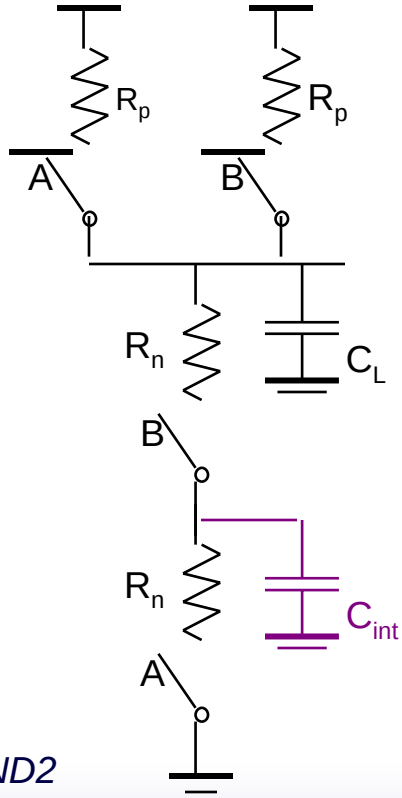
2) If transistor channel length increases, R increases

Equivalent Inverter

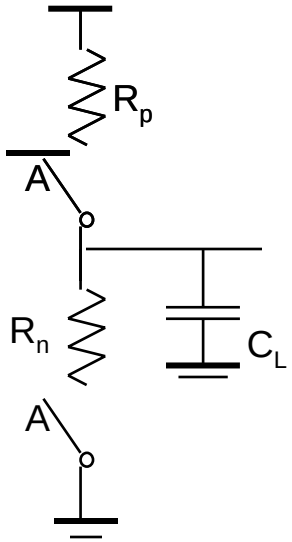
- CMOS gates: many paths to Vcc and Gnd
  - Multiple values for V_M, V_{IL}, V_{OL}, etc
  - Different delays for each input combination
- Equivalent inverter
  - Represent each gate as an inverter with appropriate device width
  - Include only transistors which are on or switching
  - Calculate V_M,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

Switch Delay Model

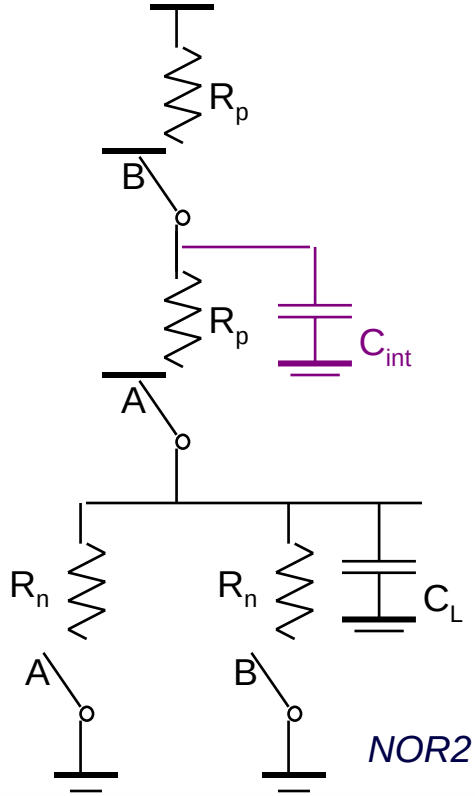




NAND2

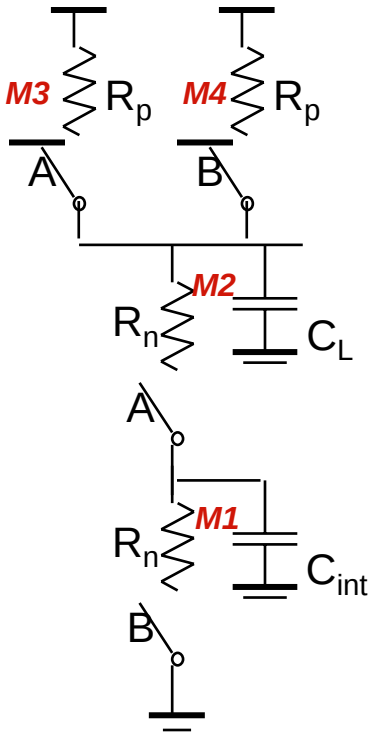


INV



NOR2

Input Pattern Effects on Delay

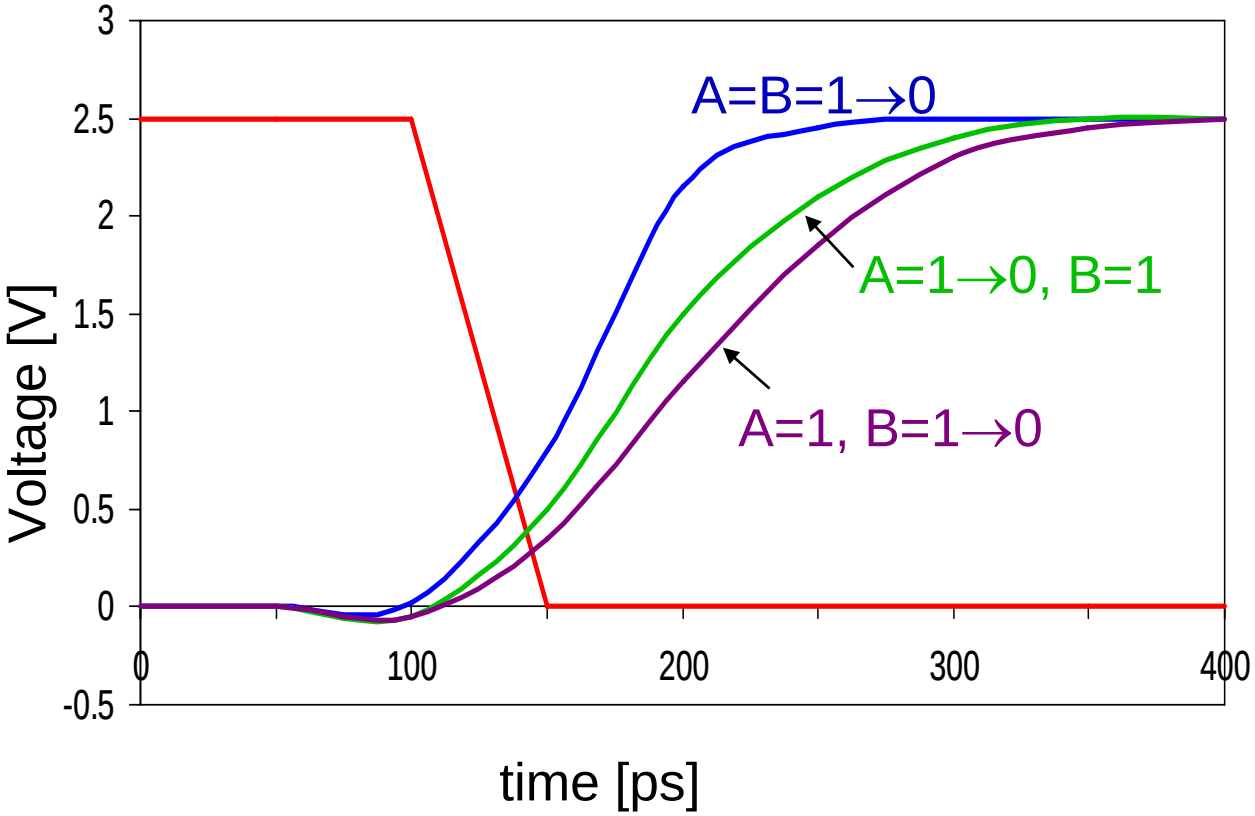


2-input NAND

- 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

Simulated Low to High delay for a 2-input NAND

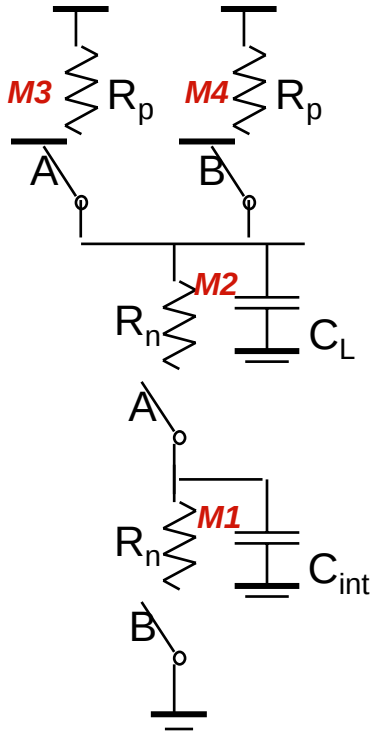


Input Data Pattern	Delay (psec)
A=B=0→1	69
A=1, B=0→1	62
A= 0→1, B=1	50
A=B=1→0	35
A=1, B=1→0	76
A= 1→0, B=1	57

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)

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

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

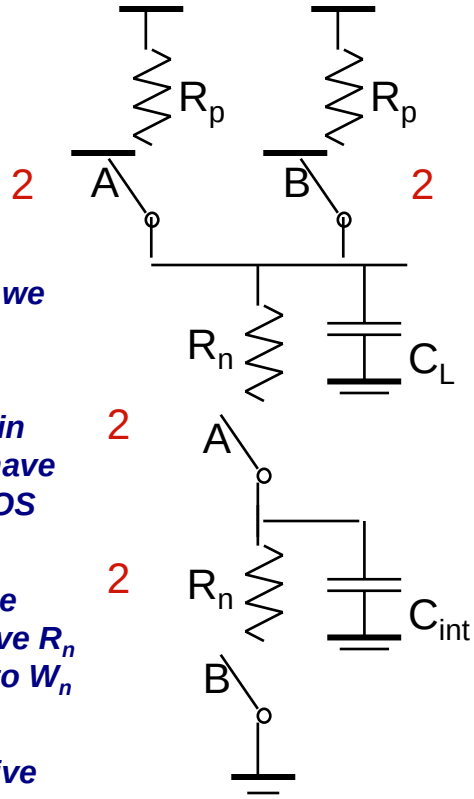
What sizing will lead to a 2:1 equivalent inverter?

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



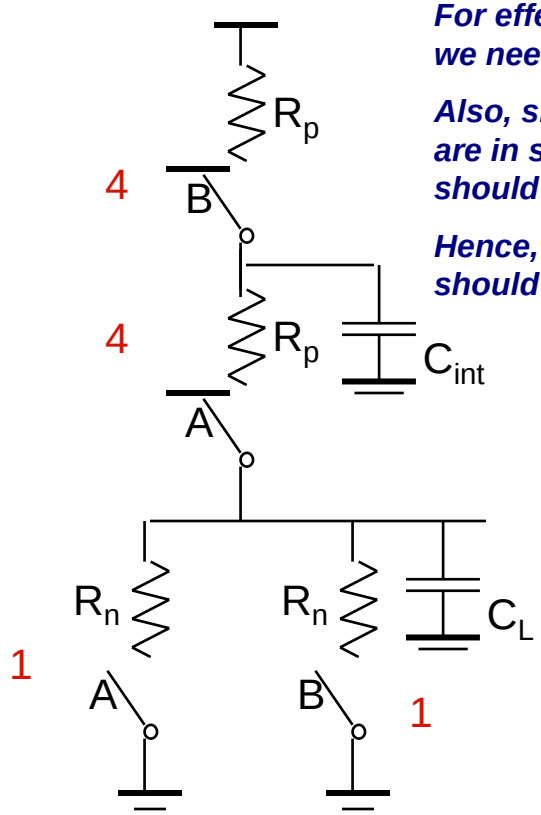
2-input NAND

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 $4W_n$

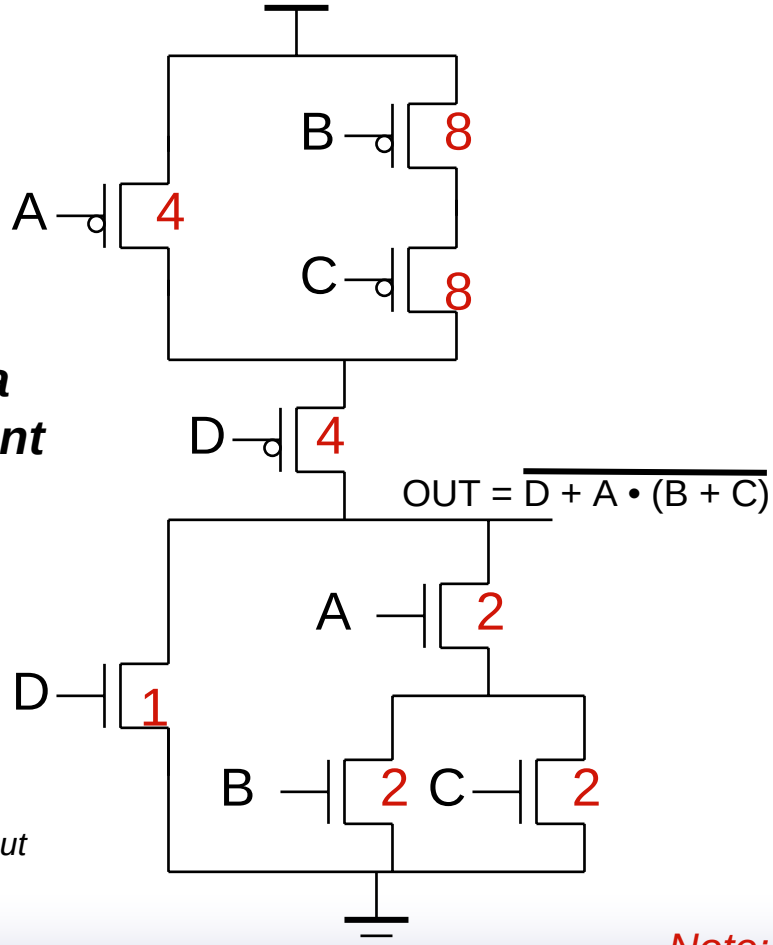
Avoid stacking PMOS transistors in series….



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.

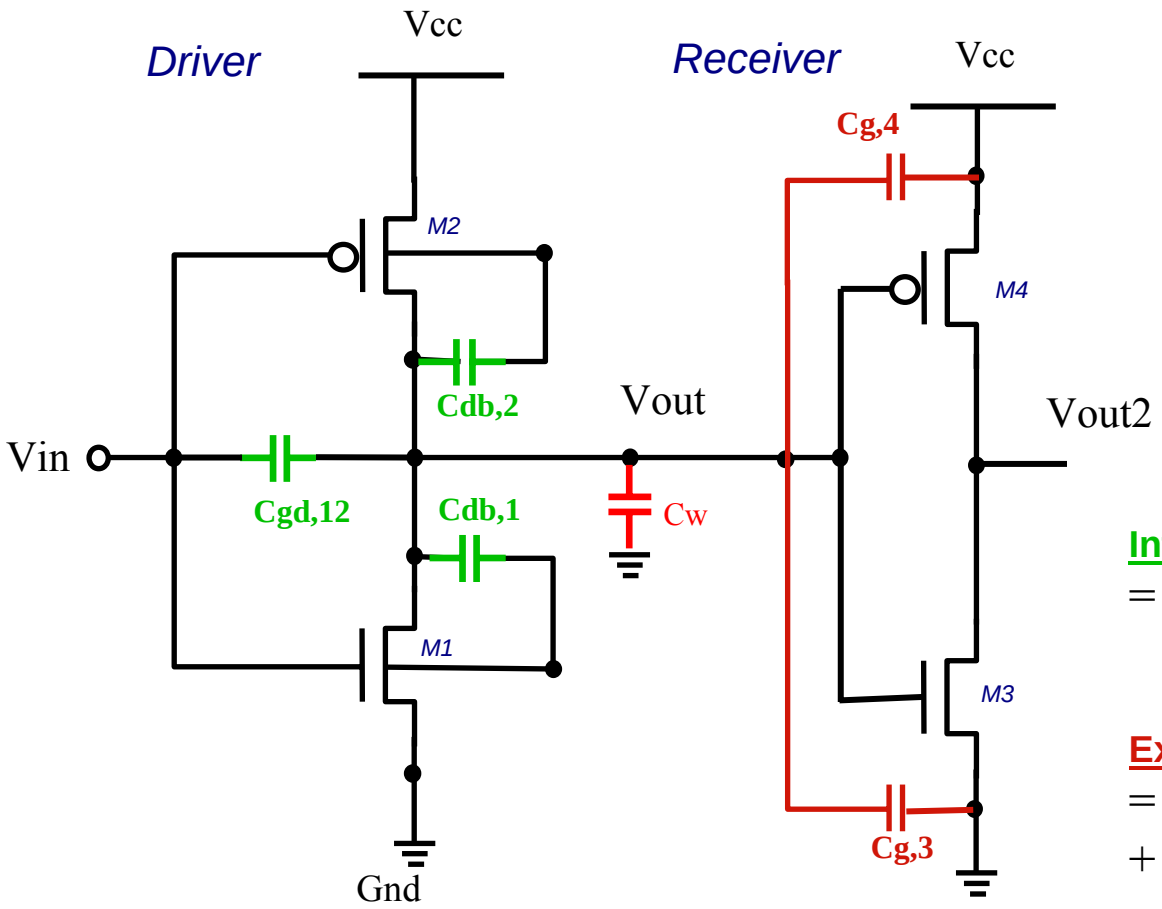
Note: here we ignored internal caps.

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 capacitance



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

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: $C_{g,43}$

Intrinsic delay of CMOS inverter

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

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

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

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

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

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

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

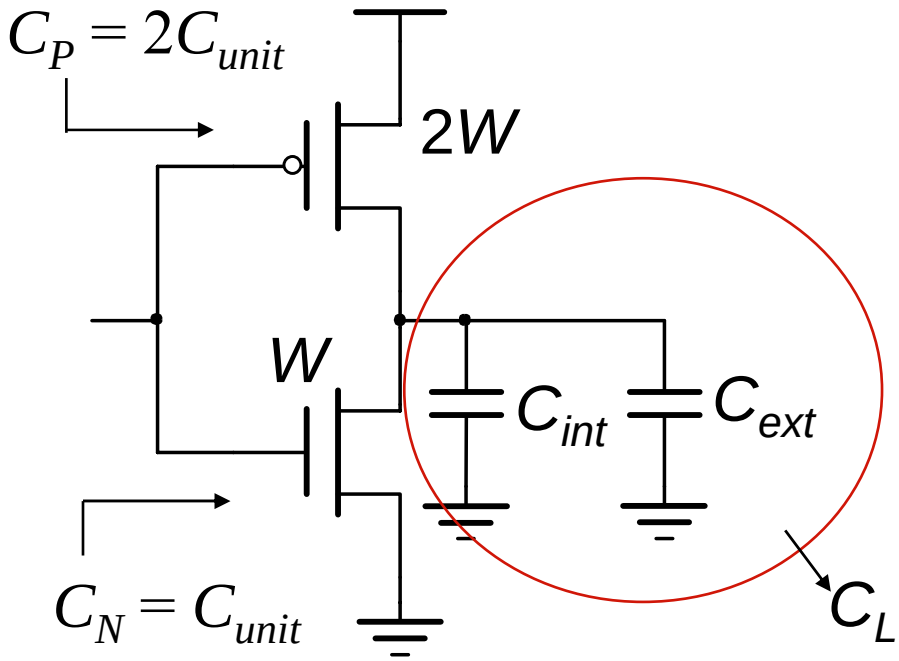
$$= 0.69\,R_{ref}\;C_{iref}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

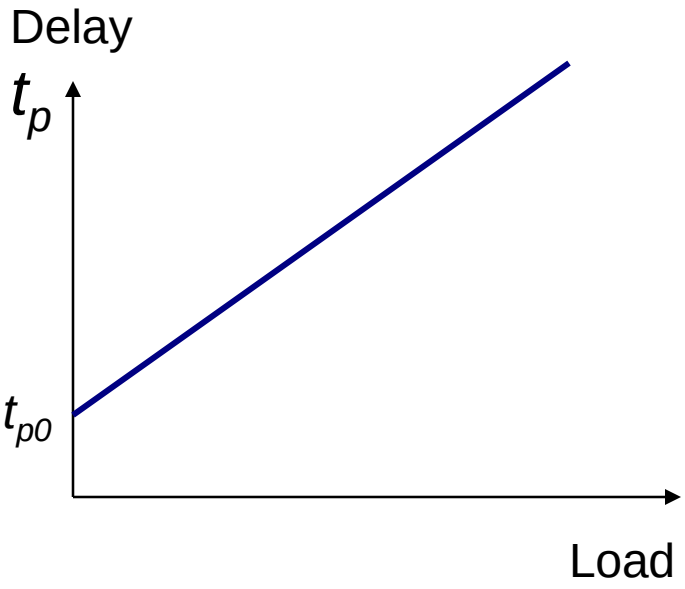
$$= t_{p0}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

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 with Load

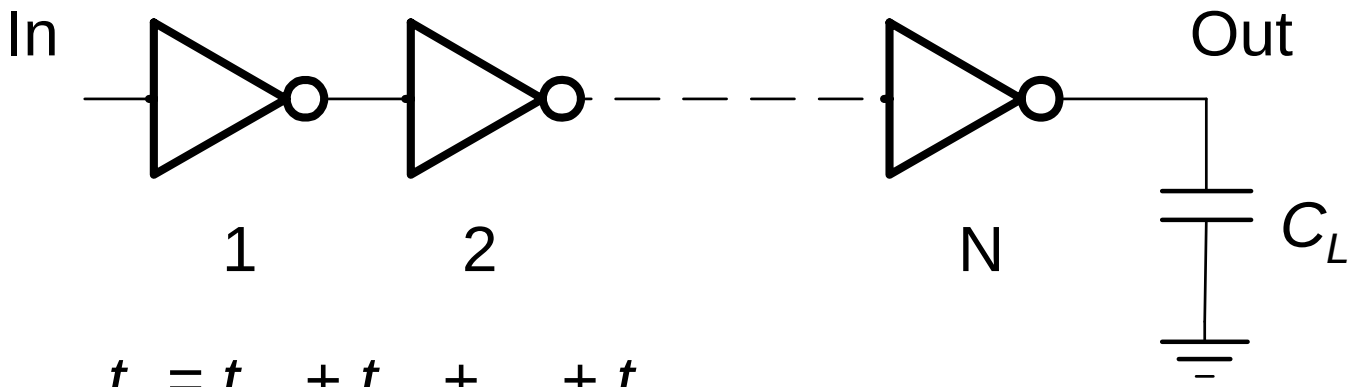




Delay (t_p) = $kR_W(C_{int} + C_{ext}) = kR_WC_{int} + kR_WC_{ext} = kR_W C_{int}(1+ C_{ext}/C_{int})$

t_{p0} *(intrinsic delay)*

Apply to Inverter Chain



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

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

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

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

Minimum path delay:

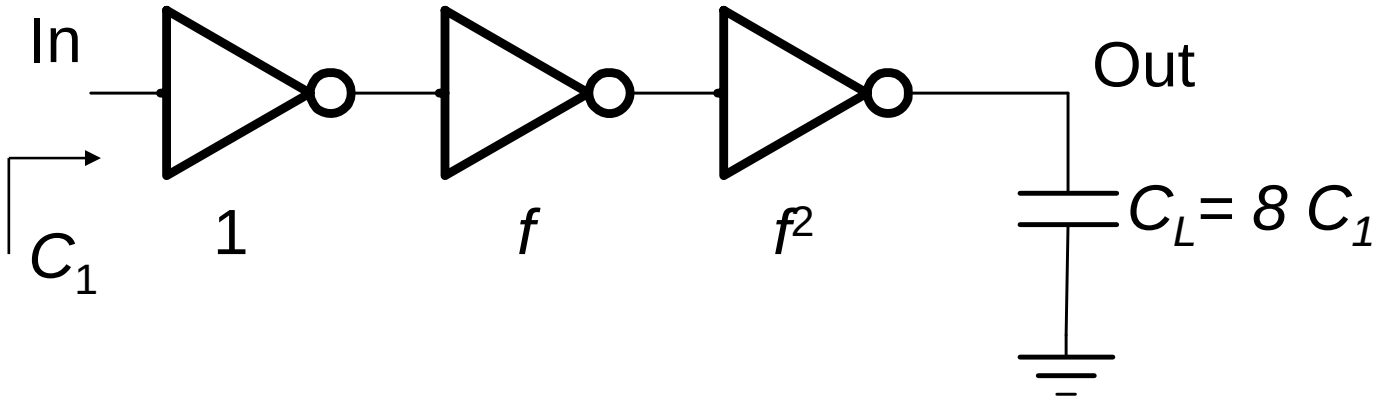
$$t_p = Nt_{p0}\left(1 + \sqrt[N]{F}/\gamma\right)$$

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

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

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

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

If self-loading is ignored….

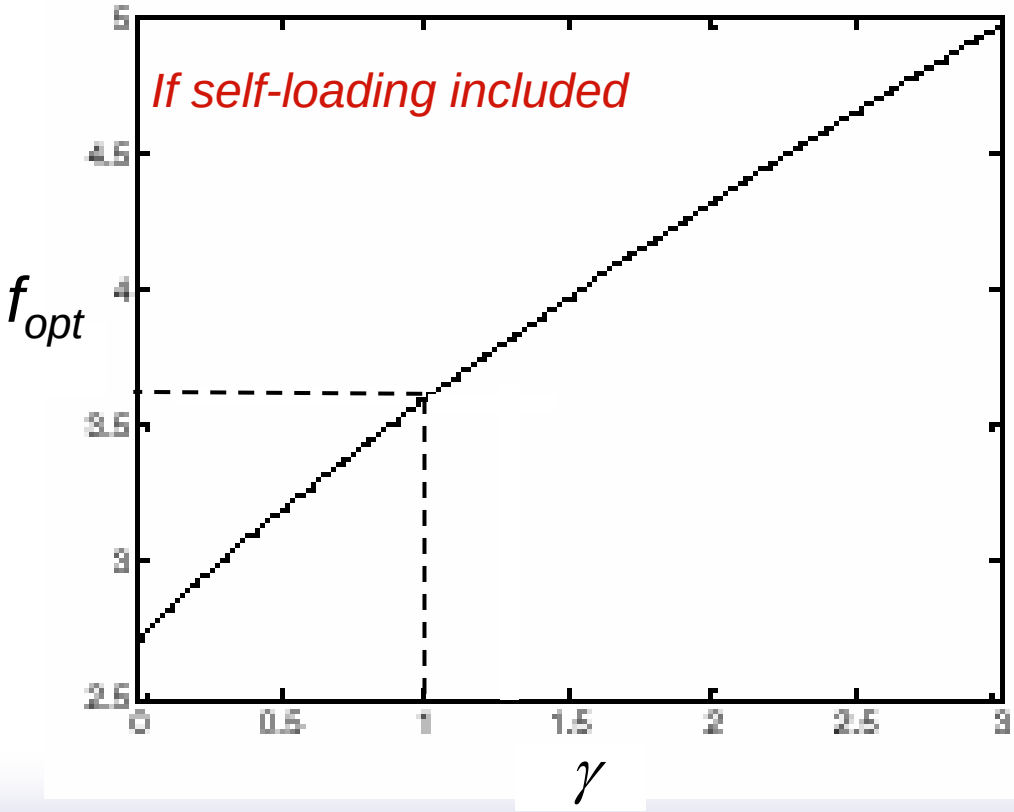
For $\gamma = 0$, $f = e$, $N = \ln F$

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

Optimum Effective Fanout f

Optimum *f* for given process defined by γ

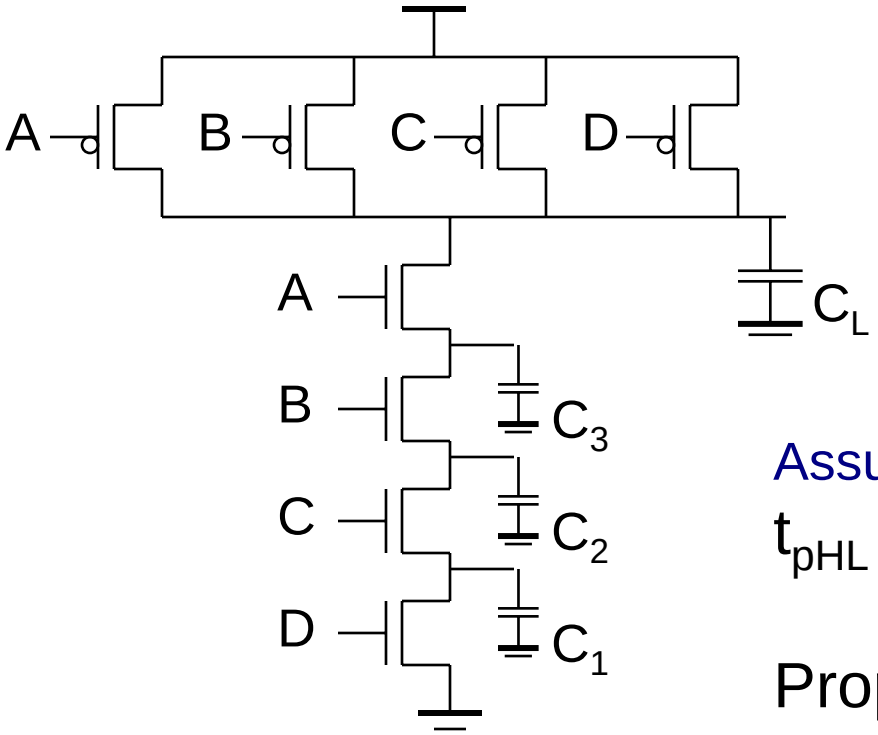
$$f = \exp\left(1 + \gamma / f\right)$$

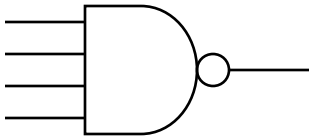


Optimum tapering factor:

f_{opt} = 3.6 for γ=1

Fan-In Considerations





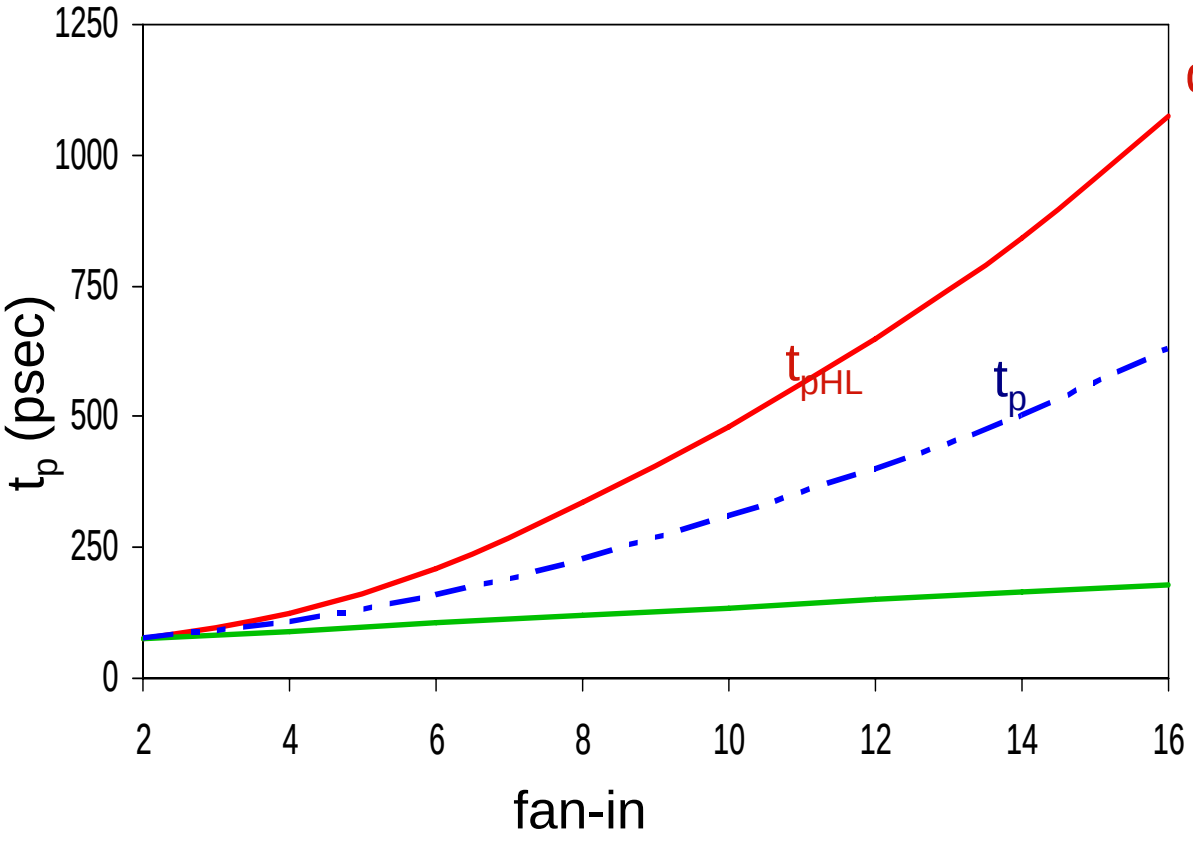
Distributed RC model
(Elmore delay)

Assuming all NMOS are equal size

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

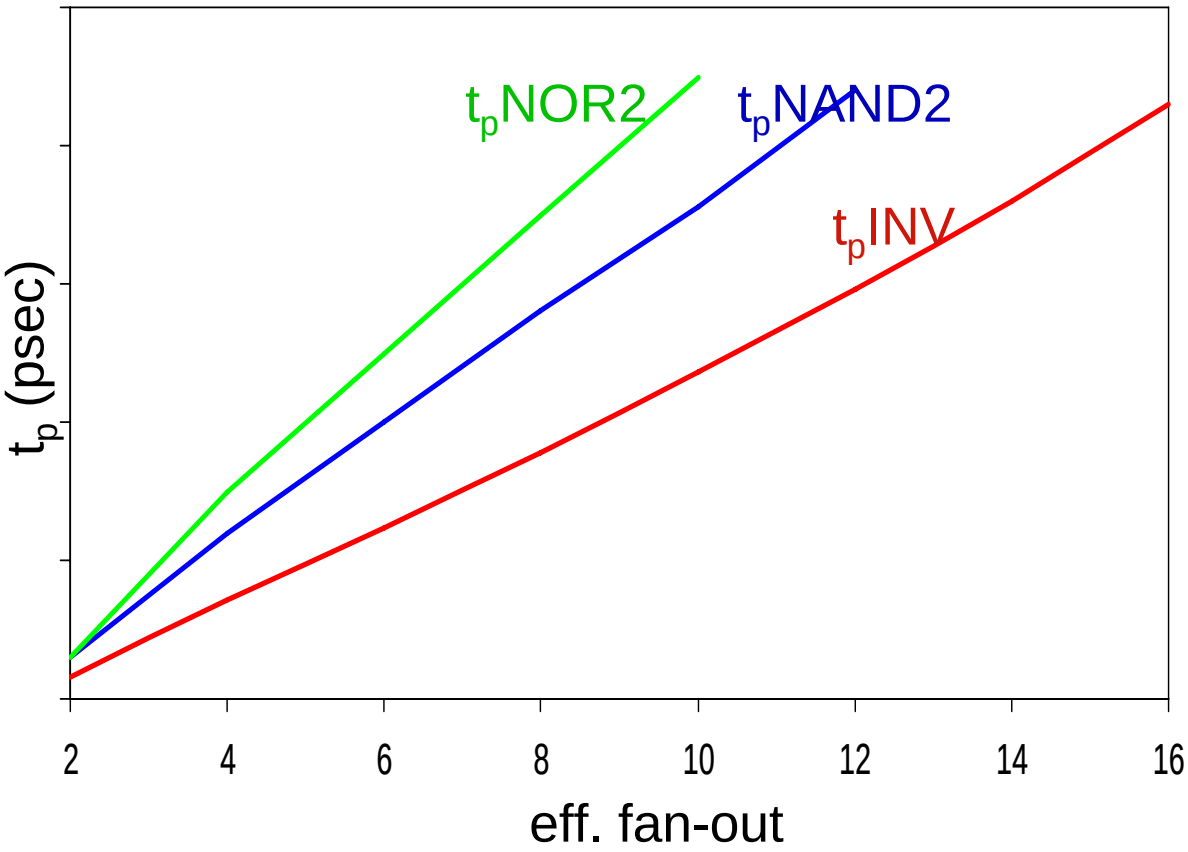
Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

t_p as a Function of Fan-In



Gates with a fan-in greater than 4 should be avoided.

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….

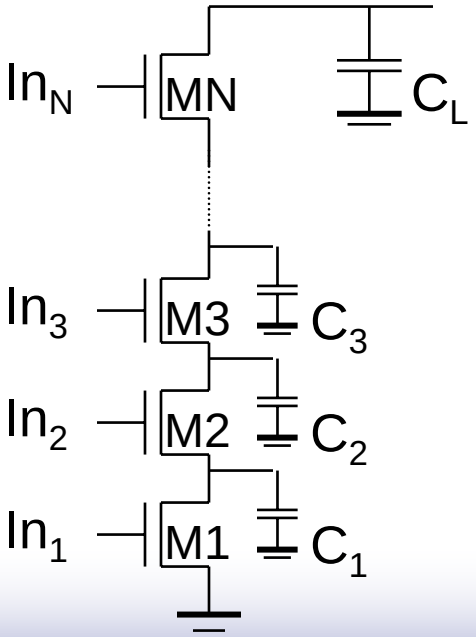
t_p as a Function of Fan-In and Fan-Out

- Fan-in: quadratic due to increasing resistance and capacitance
- Fan-out: each additional fan-out gate adds two gate capacitances to C_L

$$t_p = a_1 FI + a_2 FI^2 + a_3 FO$$

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

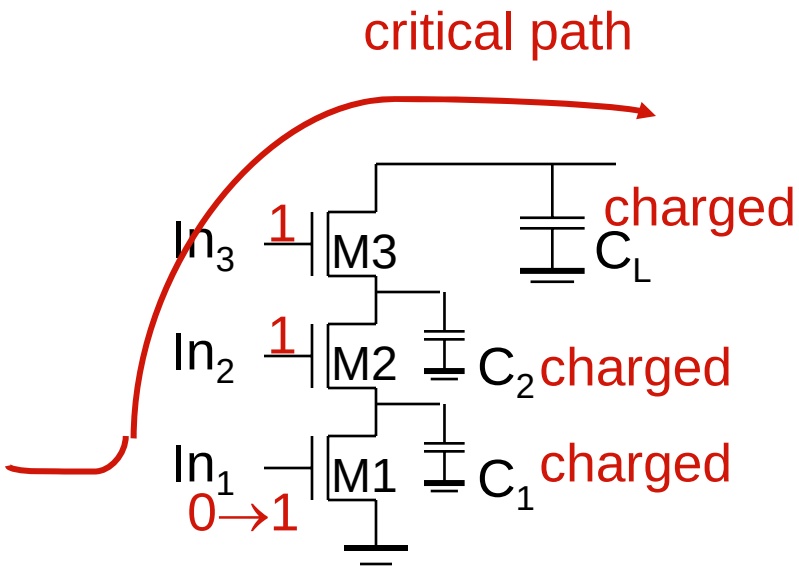
M1 > M2 > M3 > … > MN
(the transistor closest to the output is the smallest: has biggest R)

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

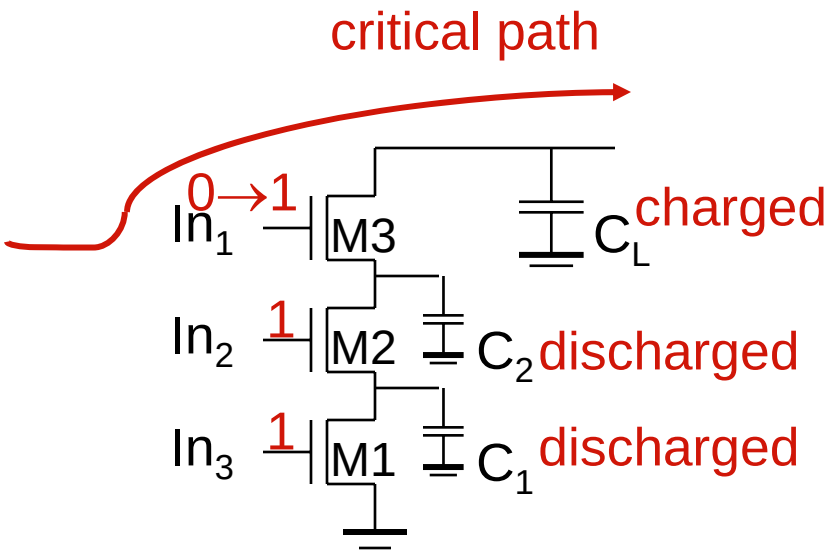
Fast Complex Gates: Design Technique 2

- 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



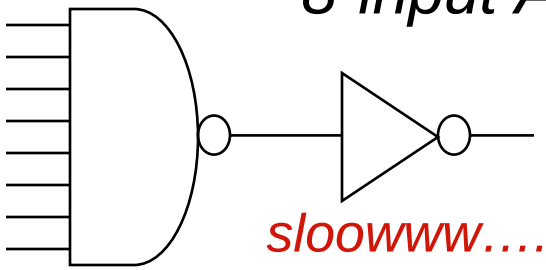
delay determined by time to discharge C_L

Fast Complex Gates: Design Technique 3

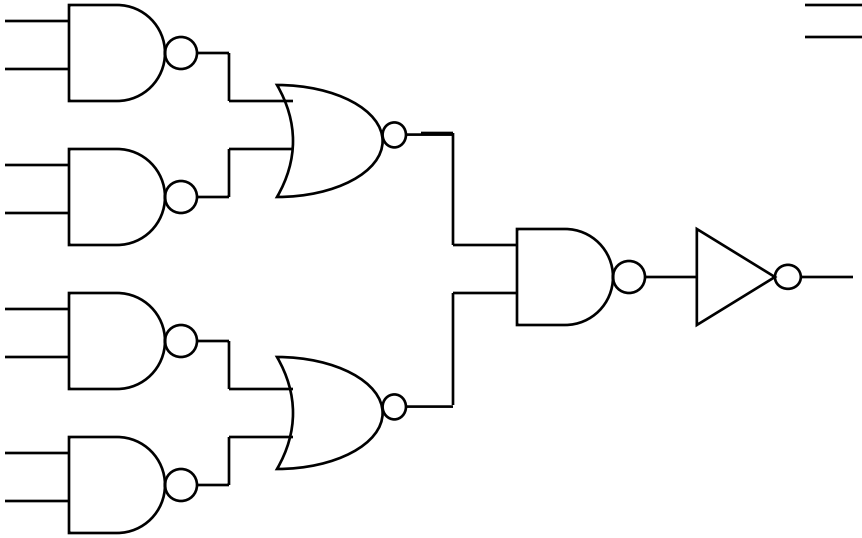
- Alternative logic structures

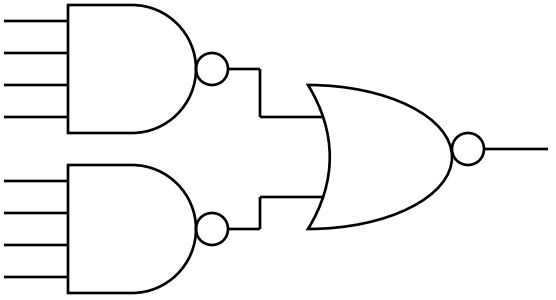
F = ABCDEFGH

8-input AND



sloowww….

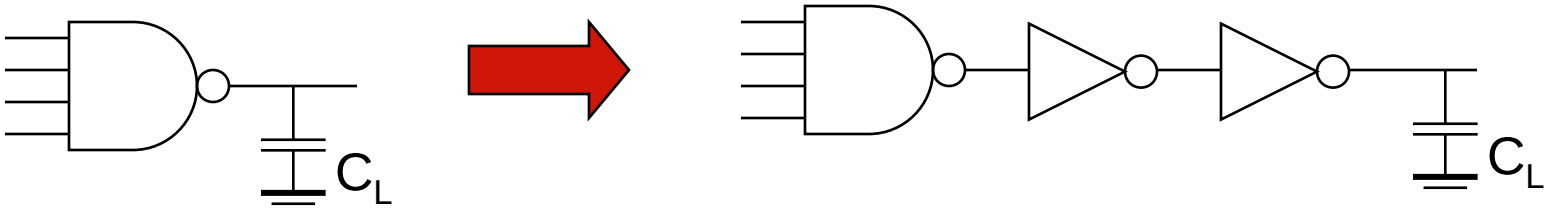




Using De Morgan's Law…

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)

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

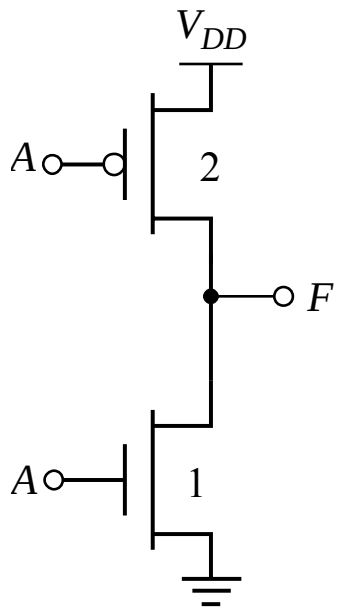
f – effective fanout *(ratio of external load and input cap. of gate)*
p – ratio of intrinsic delays of complex gate and inverter
(value increases with complexity of gate)
g – logical effort: how much more input capacitance is presented by the complex gate to deliver the same output current as an inverter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

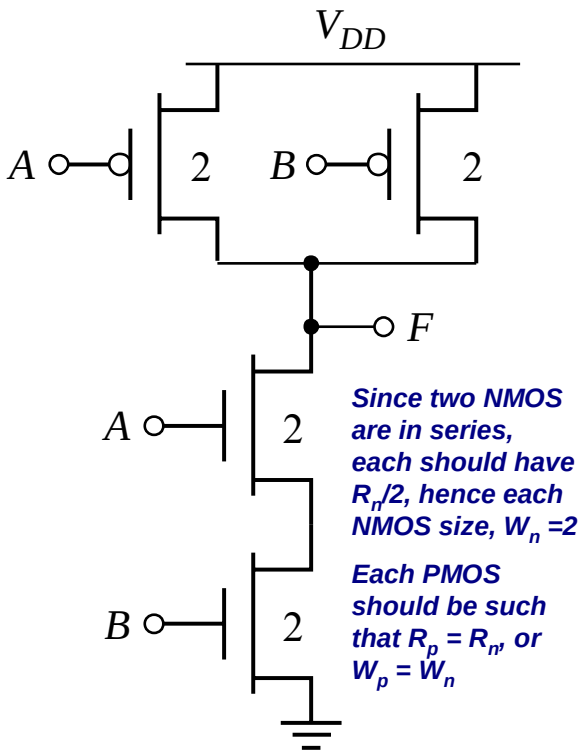
Logical Effort

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



Inverter

$g = 1$

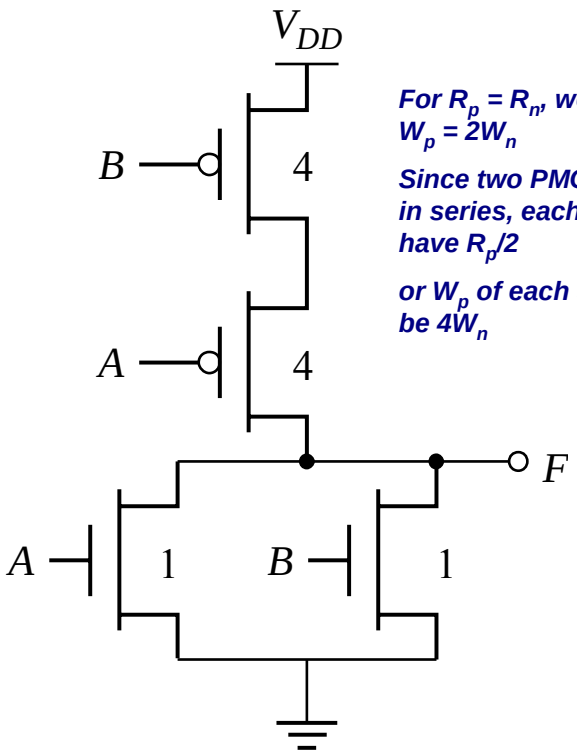


Since two NMOS are in series, each should have $R_n/2$, hence each NMOS size, $W_n = 2$

Each PMOS should be such that $R_p = R_n$, or $W_p = W_n$

2-input NAND

$g = 4/3$



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

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

or W_p of each should be $4W_n$

2-input NOR

$g = 5/3$

Delay in a Logic Gate

Gate delay:

$$d = h + p$$

effort delay **intrinsic delay**

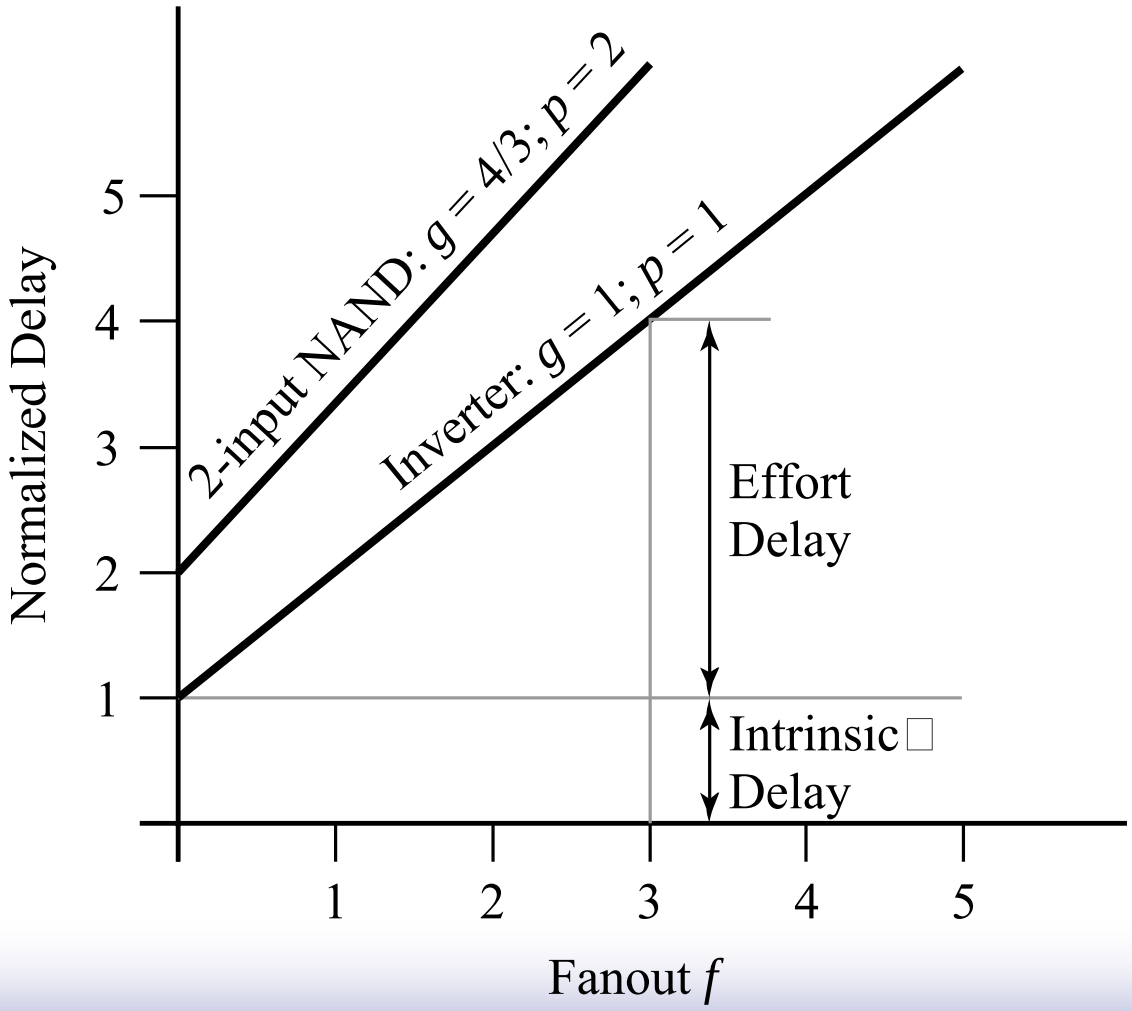
Effort delay (or gate effort):

$$h = g\,f$$

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

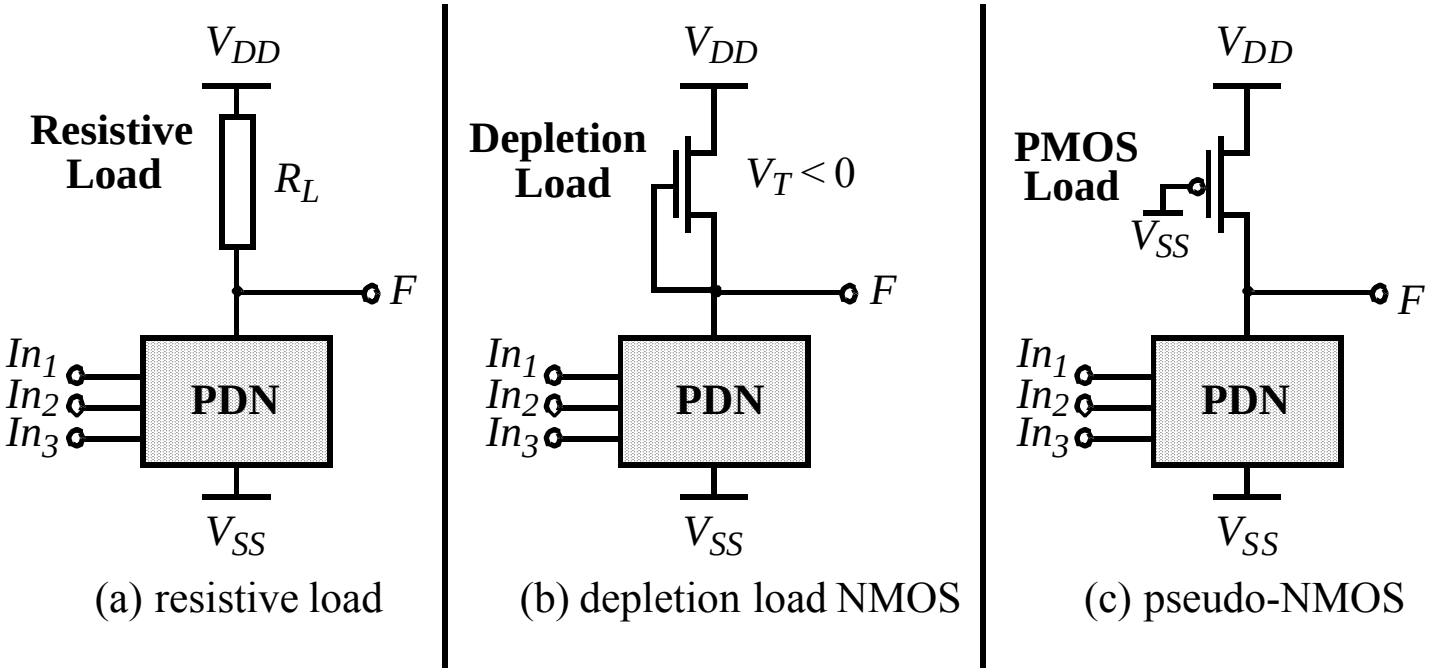
Logical Effort of Gates



Ratioed Logic

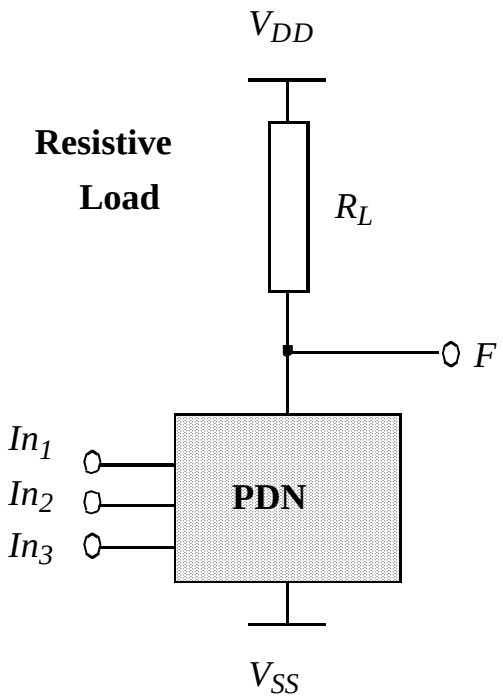
Ratioed Logic

Need N+1 transistors vs 2N for complementary CMOS



Goal: to reduce the number of devices over complementary CMOS

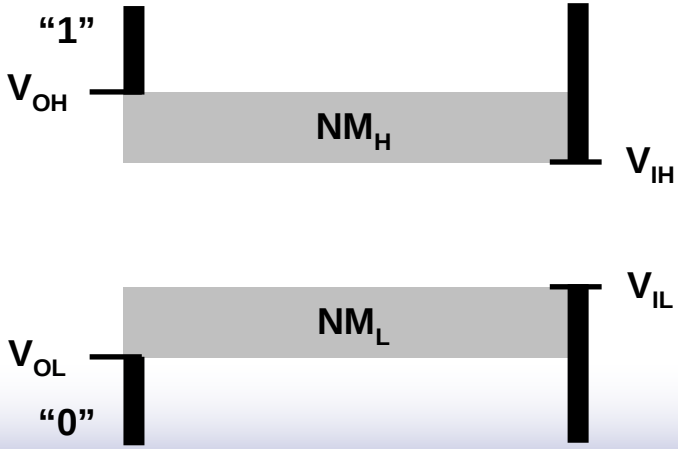
Ratioed Logic: Resistive Load



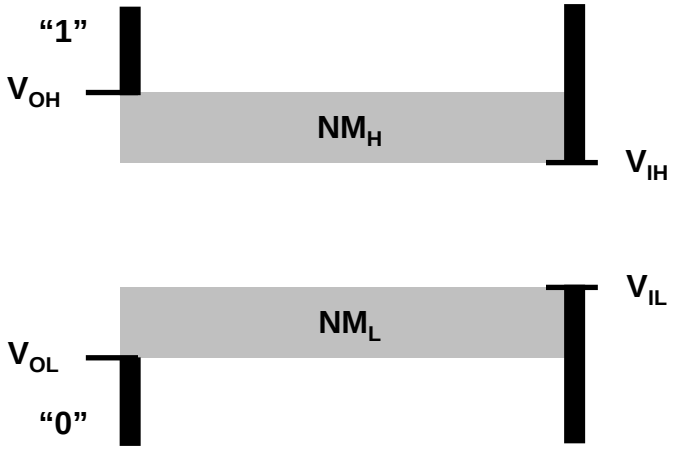
- **N transistors + Load**
- **$V_{OH} = V_{DD}$**
- **$V_{OL} = \frac{R_{PN}}{R_{PN} + R_L}$**

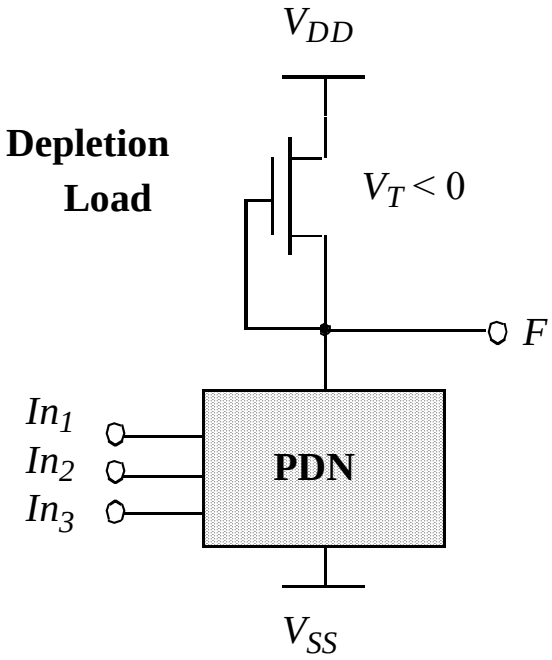
Ideally V_{OL} should be as small as possible. Hence, R_L should be large…

- **Assymetrical response**
- **Static power consumption**
- **$t_{pL} = 0.69\, R_L C_L$**

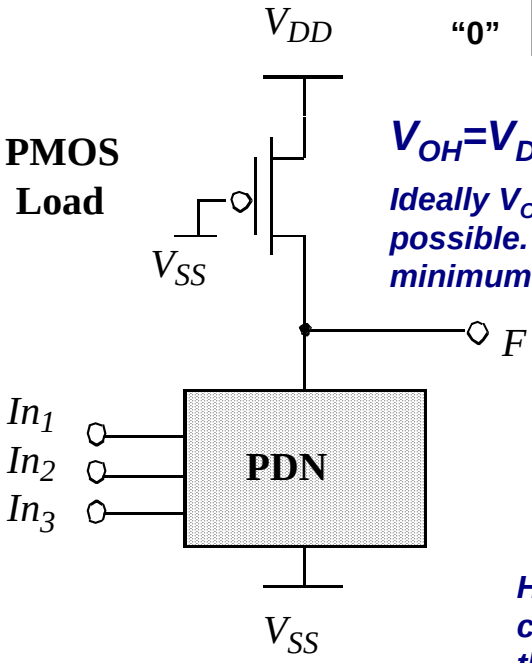


Active Loads





depletion load NMOS



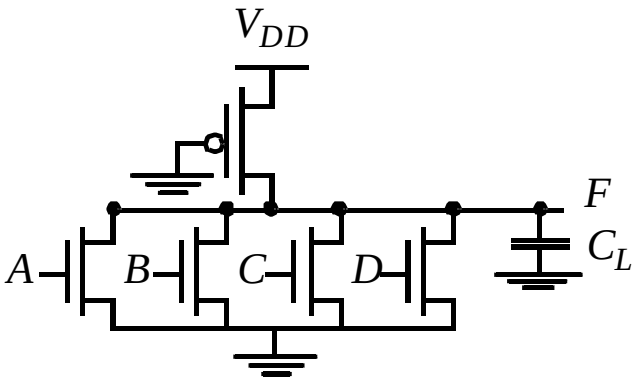
pseudo-NMOS

$V_{OH}=V_{DD}$ (assuming $V_{OL}<V_{tn}$)

Ideally V_{OL} should be as small as possible. Hence, PMOS device should be minimum sized…

However, since V_{OL} is not 0 V, contention between PMOS and the PDN lowers the NM and results in static power dissipation.

Pseudo-NMOS



$V_{OH} = V_{DD}$ (similar to complementary CMOS)

To Find V_{OL}:

$$k_n\left((V_{DD}-V_{Tn})V_{OL}-\frac{V_{OL}^2}{2}\right)+k_p\left((-V_{DD}-V_{Tp})V_{DSATp}-V_{DSATp}^2/2\right)=0$$

Note: NMOS in linear mode, since ideally the output=0 V

Note: PMOS in saturation mode

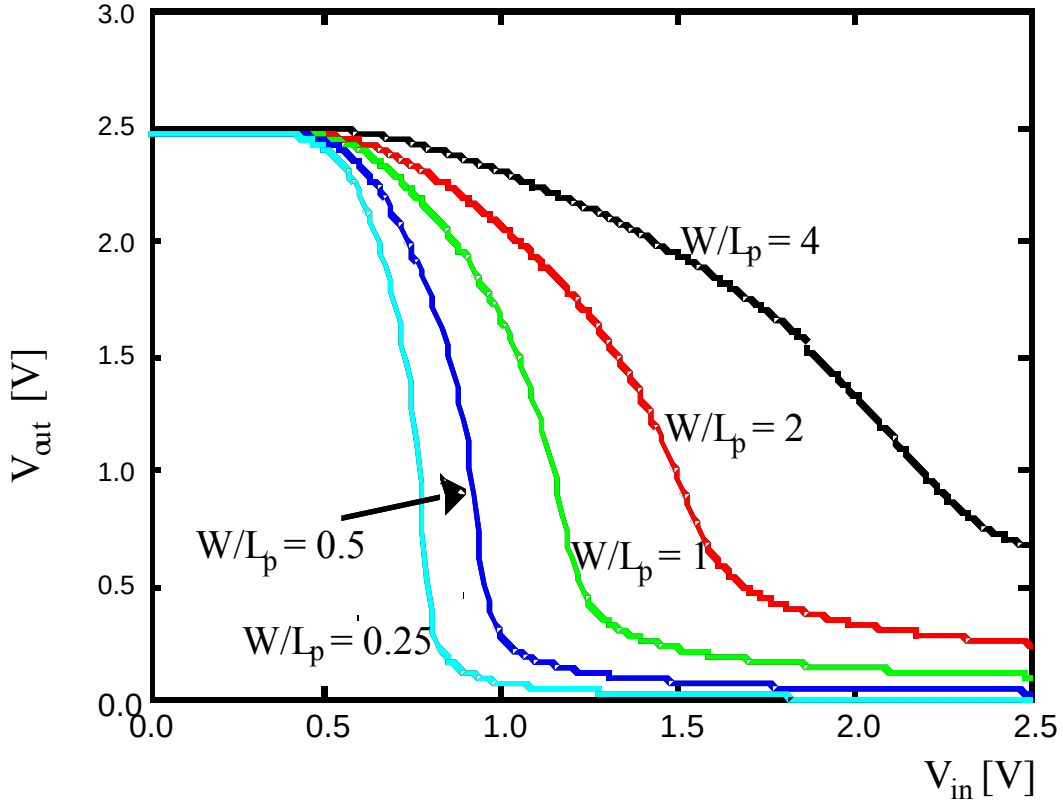
$$V_{OL} = \mu_p/\mu_n\, W_p/W_n\, V_{DSATp}$$

Assuming V_{OL} is small relative to gate drive, (V_{DD}-V_T), and V_{Tp}=V_{Tn}

SMALLER AREA & LOAD <u>BUT</u> STATIC POWER DISSIPATION!!!

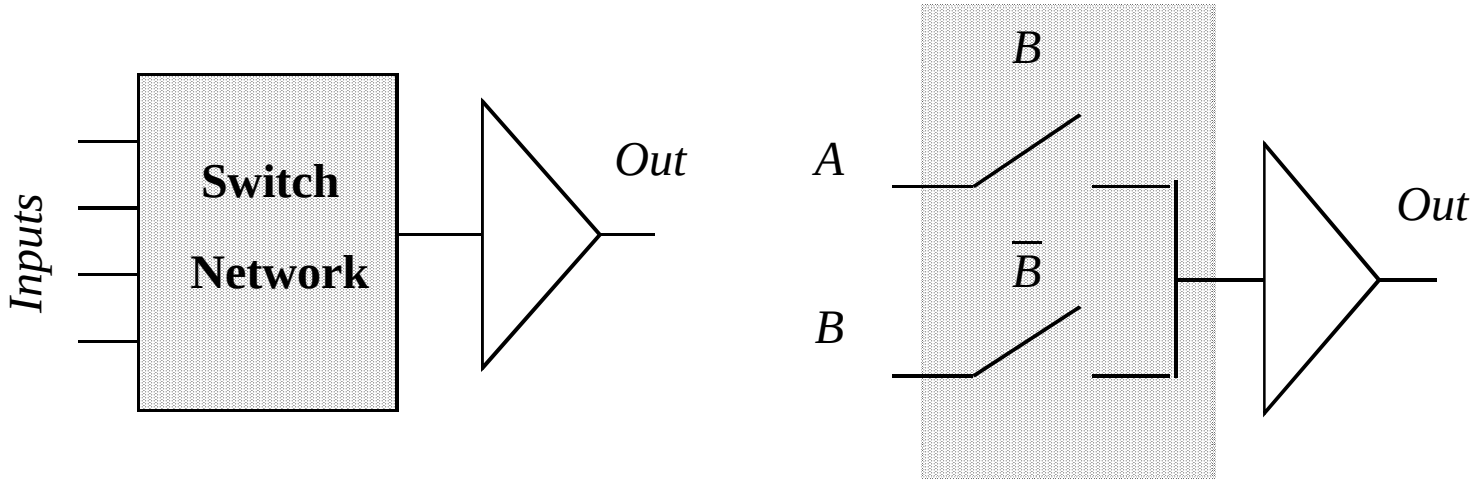
Pseudo-NMOS VTC

Sizing of the load device can be used to trade off parameters such as NM, delay, and power……



A larger pull-up device improves performance but increases static power and degrades NM by increasing V_{OL}

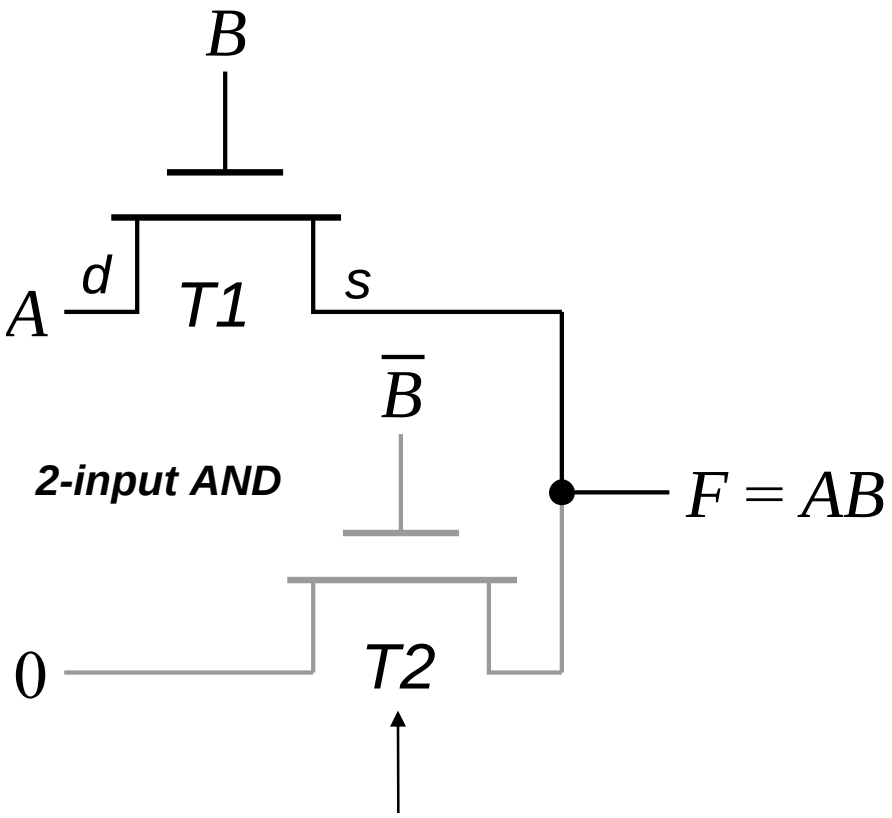
Pass-Transistor Logic



- **N transistors**
- **No static consumption**

Allows primary inputs to drive gate terminals as well as source-drain terminals

Example: AND Gate



Ensures that gate is static: provides low impedance path when B=0

***If B=1** then T1 is ON and T2 is OFF*

***Then A=F**, i.e., if A=1, F=1 and if A=0, F=0*

***When B=0**, T2 is ON and passes a Zero*

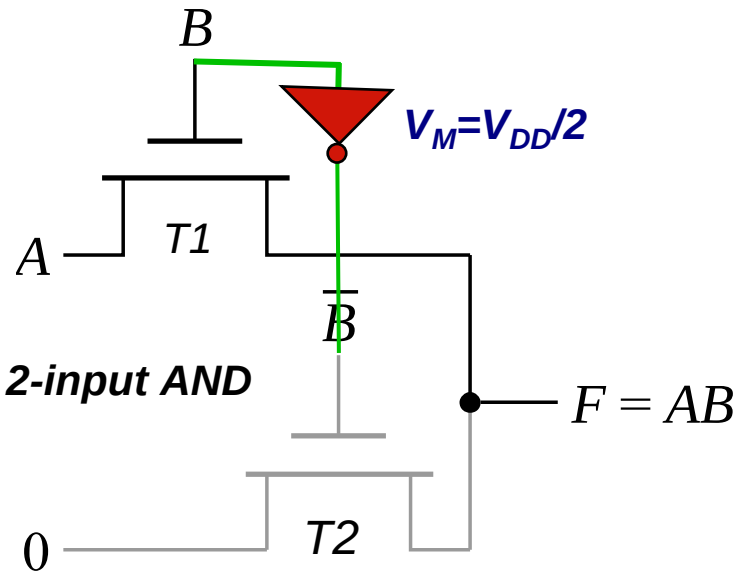
***Need fewer transistors**: 4 to implement the AND: lower cap.*

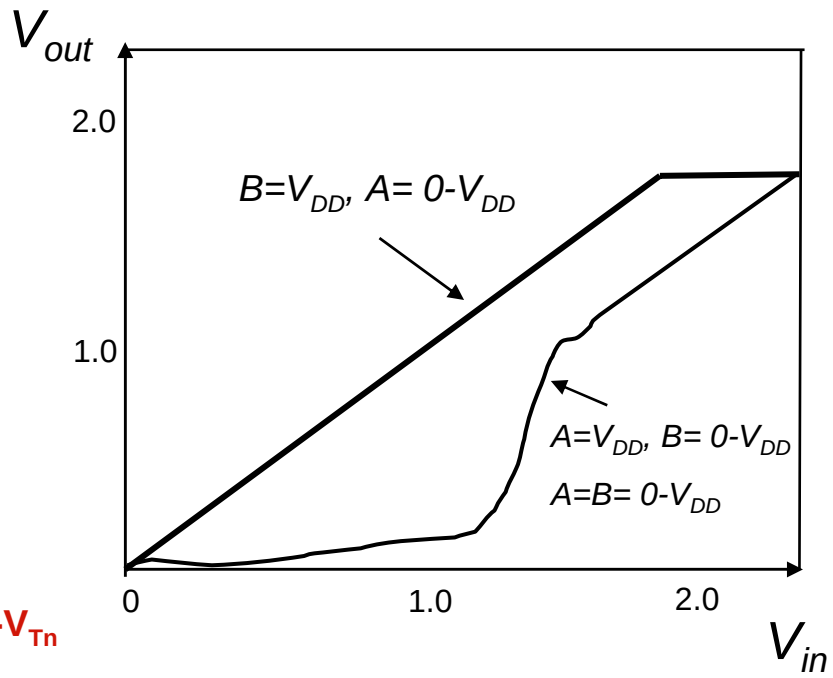
Need 6 to implement in static CMOS (4 for NAND and 2 for INV)

Note: F will charge only up to V_{DD}- V_{tn}

Also, V_{Tn} will be a function of V_F (increase due to RBB)

VTC of Pass-Transistor AND Gate





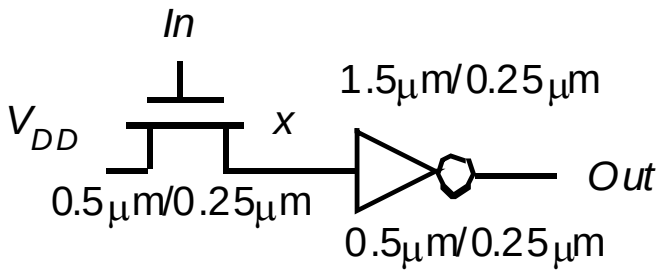
When $B=V_{DD}$, T1 is ON until the input reaches V_{DD}-V_{Tn}

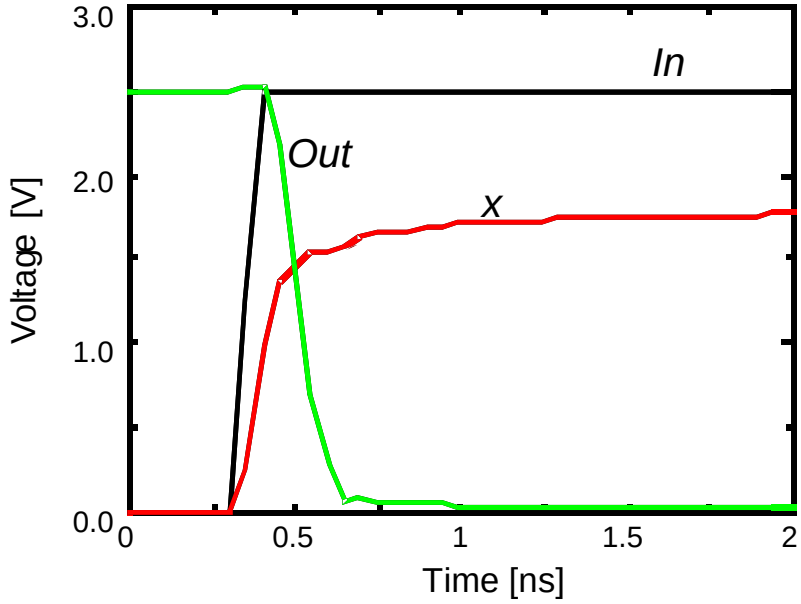
When $A=V_{DD}$, and B makes a transition from 0 to 1, T2 is turned on until $V_{DD}/2$ and Output =0. Once T2 is turned off, output follows the input B minus a threshold drop.

VTC of Pass Transistor Logic is data dependent

NMOS-Only Logic

Voltage Drop Issue:

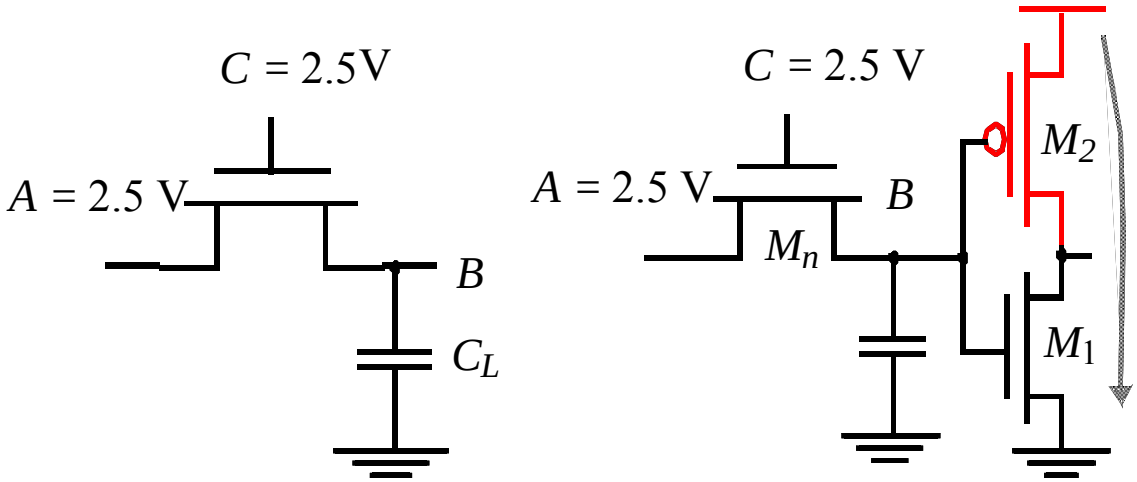




$$V_x = V_{dd} - V_{Tn}(V_x)$$

Hence, pass transistor gates cannot be cascaded by connecting the output of a pass gate to the gate input of another pass transistor. They can only be cascaded in series….

NMOS-only Switch



Lower switching energy!

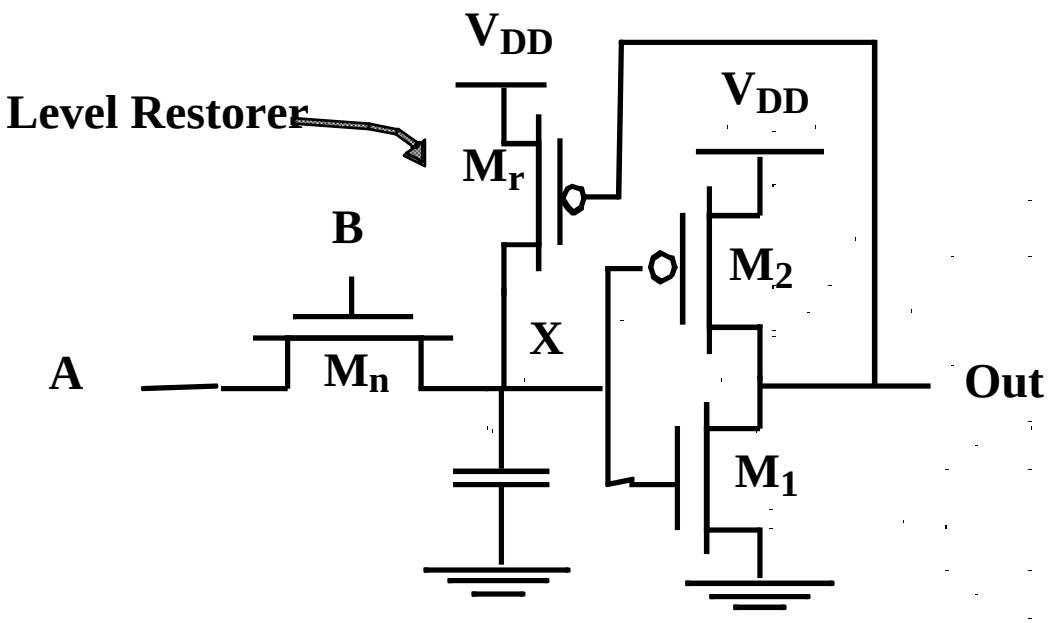
Difficult to switch off the PMOS!

V_B does not pull up to 2.5V, but 2.5V -V_{Tn}

Voltage loss causes static power consumption

NMOS has higher threshold than PMOS (body effect)

Solutions to the Voltage Drop Problem:
Solution 1: Level Restoring Transistor



Pass Transistor Logic suffers from static power dissipation and reduced NMs

At B=V_{DD}, if A: 0 to V_{DD}
$V_x = V_{DD}-V_{Tn}$, Out=0,
M_r=ON and $V_x=V_{DD}$

Eliminates static power in the Inverter

No static power between M_r and M_n

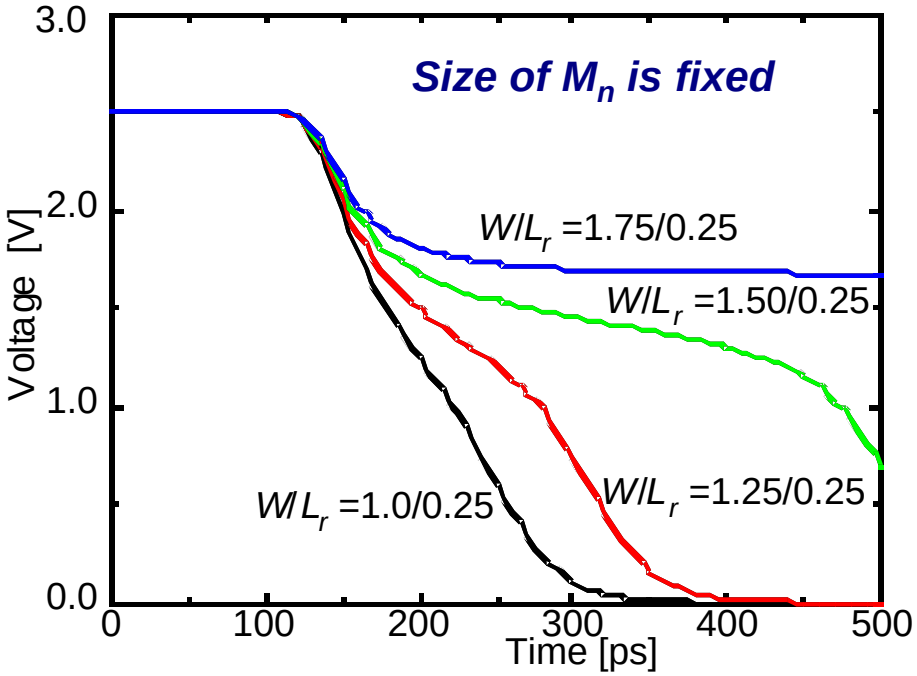
- **Advantage: Full Swing**

- **Restorer adds capacitance, takes away pull down current at X (for high to low transition at X, M_n must be stronger than M_r), can slow down gate**

- **Ratio problem**

Restorer Sizing

Need to size M_n and M_r to bring $V_x < V_M(=V_{DD}/2)$ (V_M is a function of R1 and R2) *R1 and R2 are the equivalent on-resistances of M1 and M2*

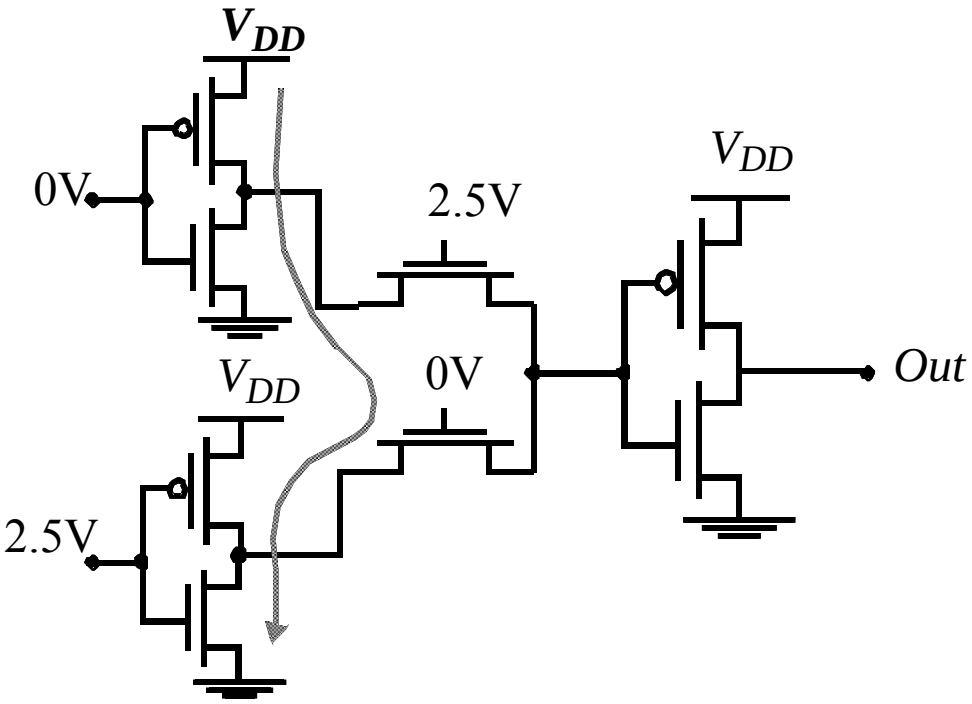


- **Upper limit on restorer size** when too large (R_r too small), V_x can't be brought below V_M
- Pass-transistor pull-down can have several transistors in stack

Transient Response: V_x vs. time

Solution 2: Single Transistor Pass Gate with V_T=0

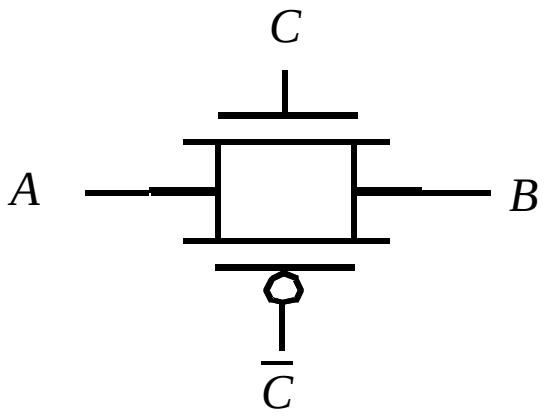
Only Pass Transistor Devices have V_T=0

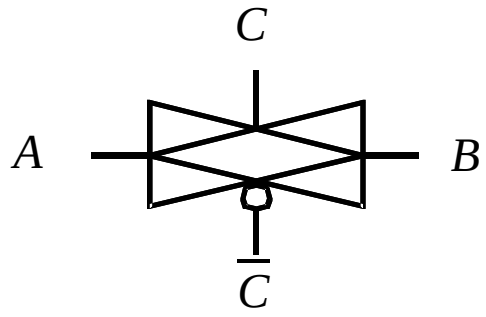


But even if V_T=0, there is still body effect…which prevents full swing!

WATCH OUT FOR LEAKAGE CURRENTS in the IDLE State!!!

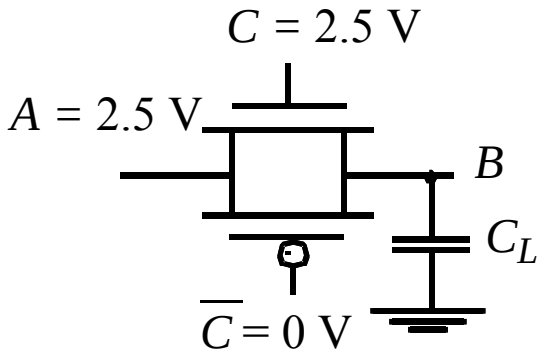
Solution 3: Transmission Gate





Acts like a bidirectional switch controlled by the gate signal C

When C=1, both MOSFETS are ON allowing the signal to pass through the gate (A=B, if C=1)



Because of the PMOS, C_L charges to Vdd

Because of NMOS C_L discharges to 0

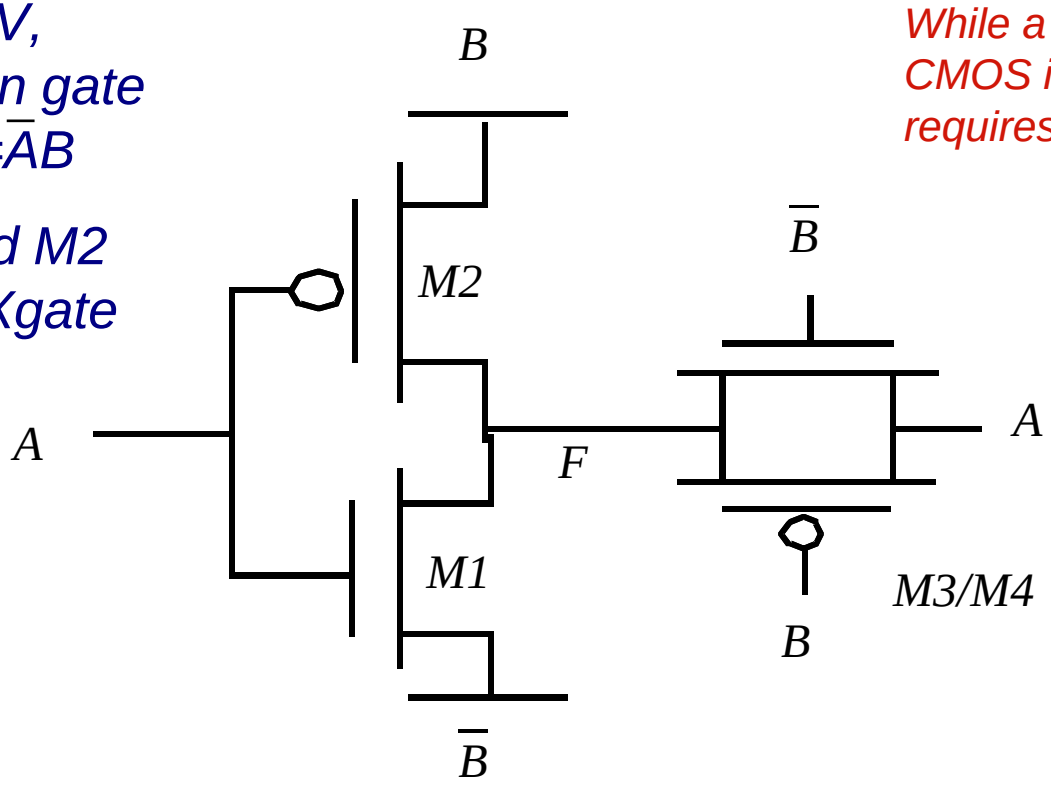
Transmission Gate XOR

B=1: *M1 and M2 act as an INV, Transmission gate is off and* $F=\bar{A}B$

B=0: *M1 and M2 are off and Xgate is on,* $F=A\bar{B}$

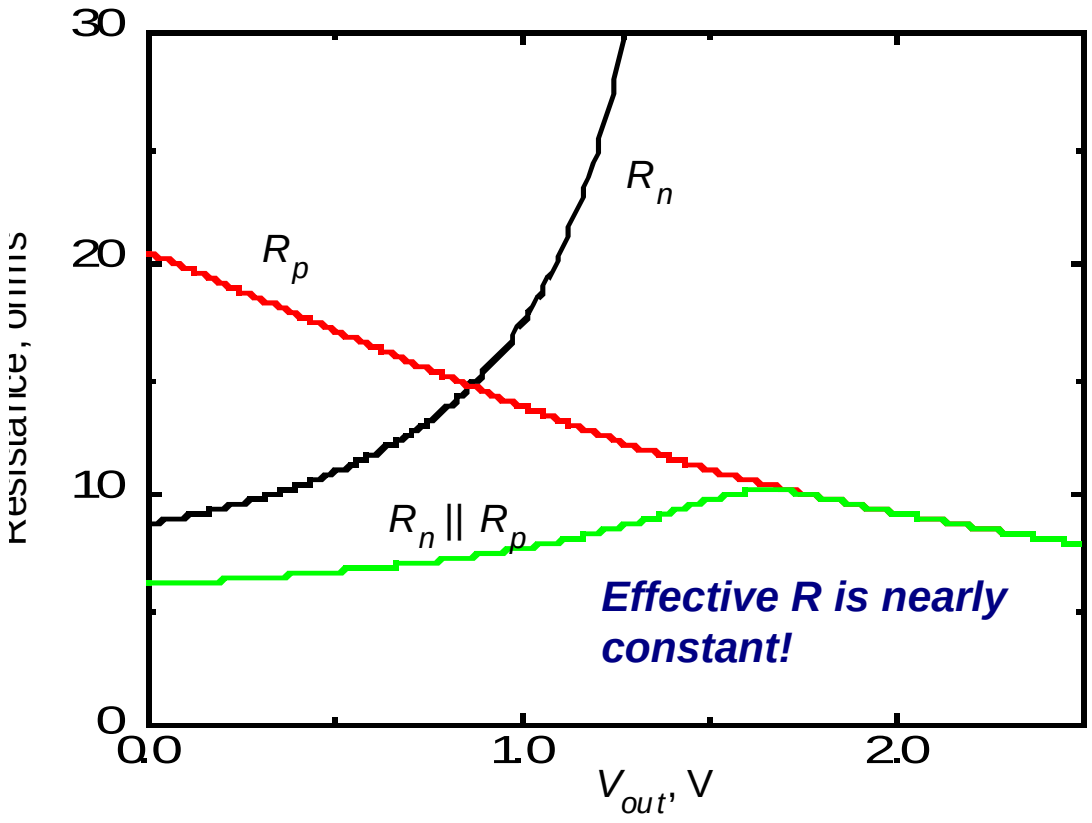
Implementation requires only 6 transistors!

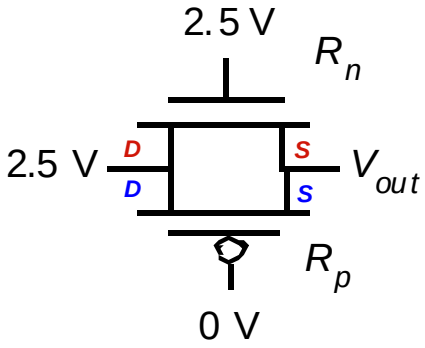
While a complementary CMOS implementation requires ???? transistors



Fast adder circuits and registers can also be implemented with Xgates

Resistance of Transmission Gate



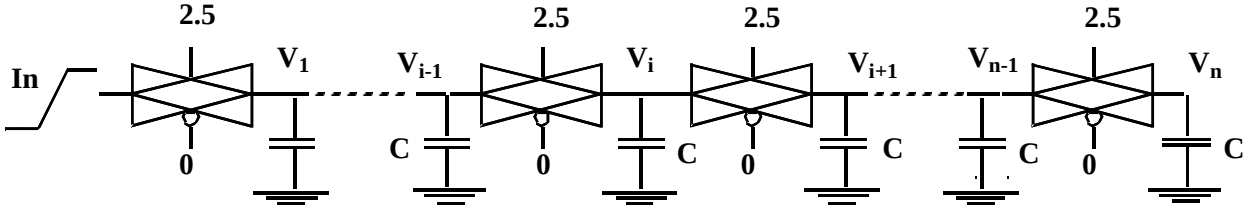


$R_n = (V_{DD}-V_{out})/I_{dn}$

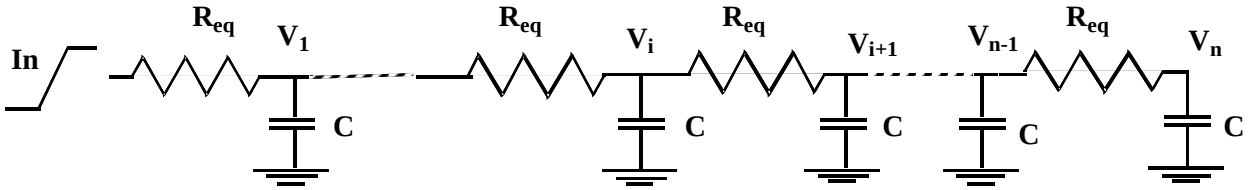
$R_p = (V_{out}-V_{DD})/I_{dp}$

Delay in Transmission Gate Networks

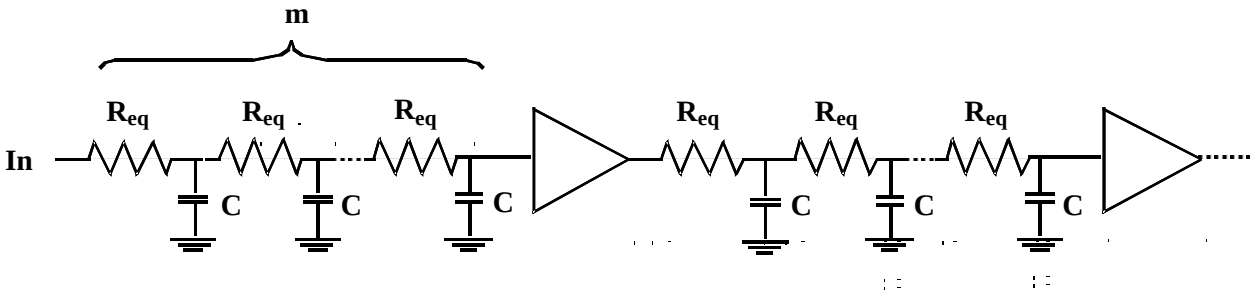
Delay of a chain of n Xgates (used in adders and deep MUXes) can be modeled using Elmore delay



(a) A chain of n Xgates



(b) Equivalent RC representation

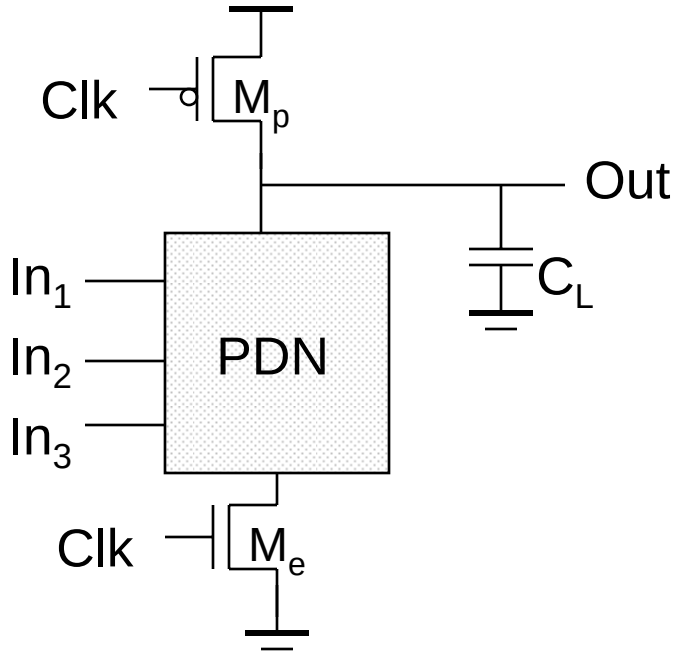


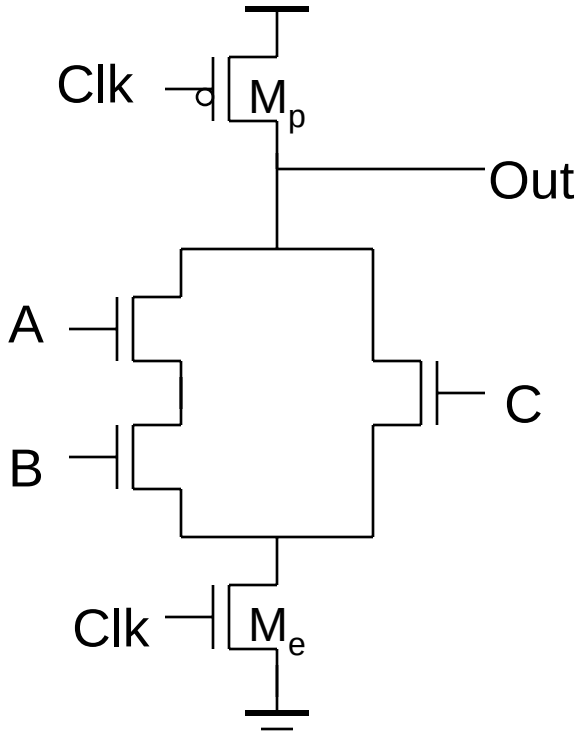
(c) Buffer insertion in a chain of Xgates to lower delay

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

Dynamic Gate



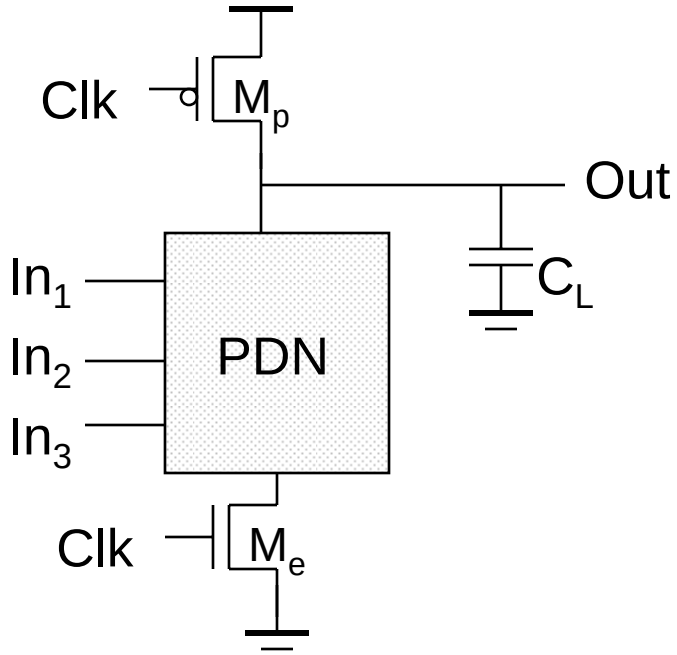


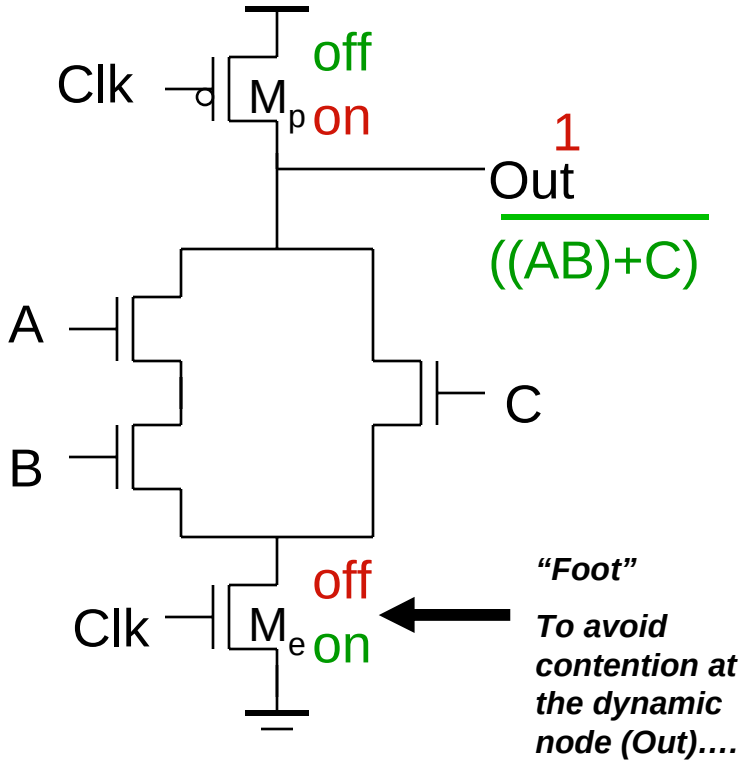
Two phase operation

Precharge (CLK = 0)

Evaluate (CLK = 1)

Dynamic Gate





Two phase operation

Precharge (Clk = 0)

Evaluate (Clk = 1)

$Out = \overline{CLK} + \overline{(AB)+C} \cdot CLK$

Conditions on Output

- During evaluation phase, the only possible path between output node and supply rail is to ground. Hence, once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make at most one transition during evaluation.
- Output can be in the high-impedance state during and after evaluation (if PDN is off), state is stored on C_L

Properties of Dynamic Gates

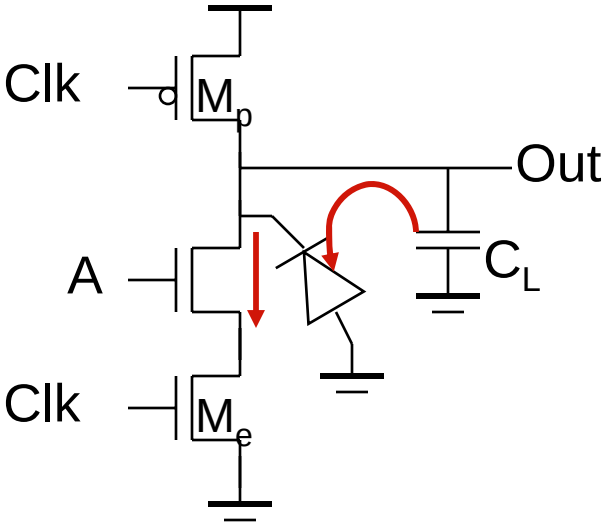
- Logic function is implemented by the PDN only
  - number of transistors is N + 2 (versus 2N for static complementary CMOS)
- Full swing outputs (V_{OL} = GND and V_{OH} = V_{DD})
- Non-ratioed - sizing of the devices does not affect the logic levels
- PDN starts to work as soon as the input signals exceed V_{Tn}, so V_M, V_{IH} and V_{IL} equal to V_{Tn}
- Low noise margin (NM_L)
- Needs a precharge/evaluate clock
- Faster switching speeds
  - reduced load capacitance due to lower input capacitance (C_{in}) due to lower number of transistors per gate and single transistor load per fan-in (reduced logical effort, 2/3 for a 2-input dynamic NOR)
  - no I_{sc}, so all the current provided by PDN goes into discharging C_L

Properties of Dynamic Gates

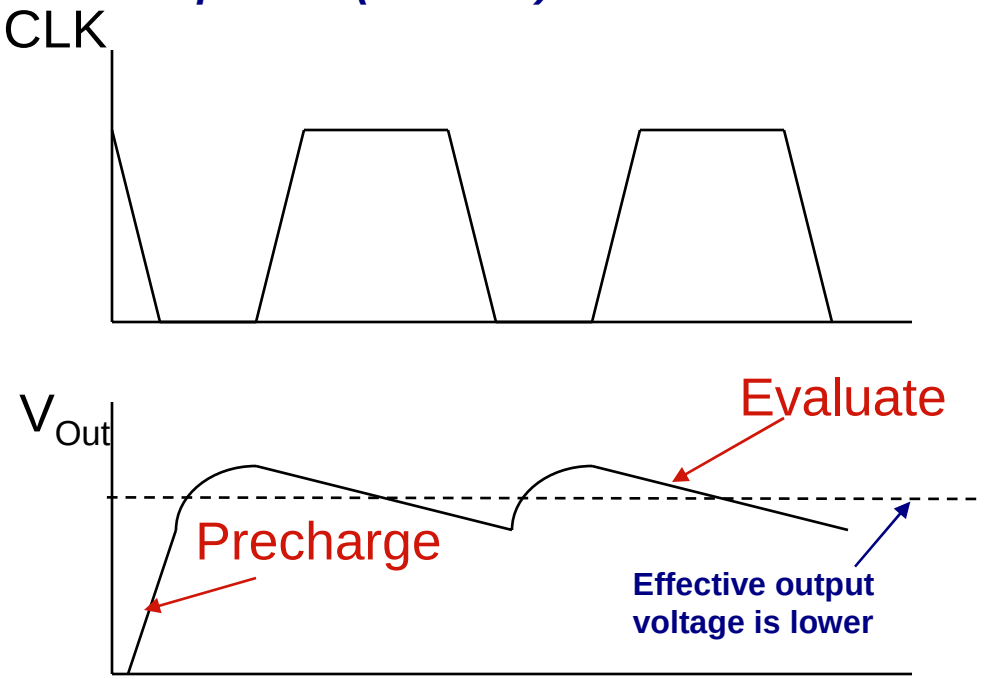
- Advantages:
  - Lower physical capacitance: uses fewer transistors
  - No glitching (dynamic gates can have at most one transition per CLK cycle)
  - Only consumes dynamic power…..no static current path ever exists between V_{DD} and GND (including P_{sc})
- In spite of the above…..overall power dissipation usually higher than static CMOS
  - CLK power can be significant: extra load on Clk + transition every CLK cycle
  - Number of transistors is more than the minimal set required for implementing logic
  - Higher switching activity due to higher transition probabilities

Issues in Dynamic Design 1: Charge Leakage

A dynamic inverter



Due to leakage, a minimal CLK rate required…(few KHz)



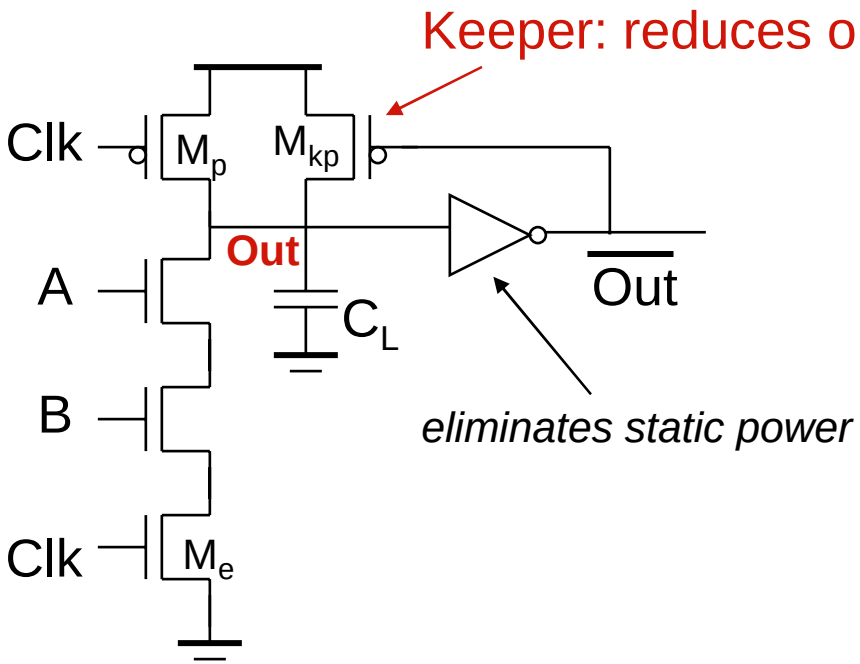
Leakage sources: reverse biased diode and subthreshold

Dominant component is subthreshold current

Note: leakage of precharge PMOS can partially compensate for the charge loss at the dynamic node

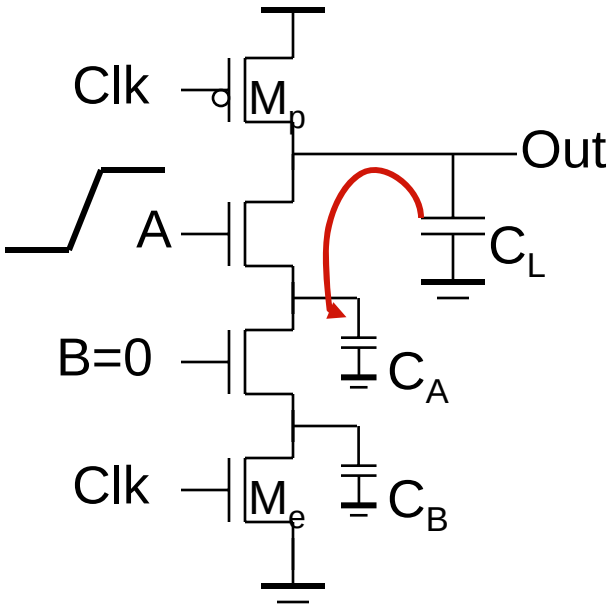
Solution to Charge Leakage

Same approach as level restorer for pass-transistor logic



Contention between keeper and PDN---strength of keeper must be less than that of the PDN to lower the out node well below the switching threshold of the next gate. Hence keeper size should be small.

Issues in Dynamic Design 2: Charge Sharing



Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to reduced robustness

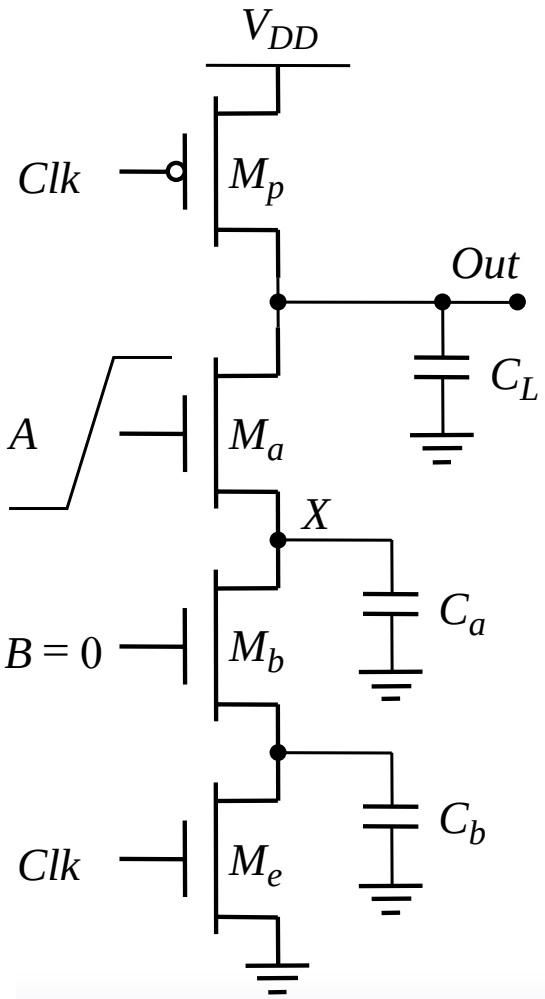
Output node voltage drops and cannot be recovered due to the dynamic nature of the circuit.

Charge Sharing

All inputs = 0 during pre-charge

Initial conditions: V_{out} (t=0)=V_{DD} and V_x(t=0)=0

2 possible scenarios:



case 1) if $\Delta V_{out} < V_{Tn}$

Final value of V_X

$$C_L V_{DD} = C_L V_{out}(t) + C_a(V_{DD} - V_{Tn}(V_X))$$

or

$$\Delta V_{out} = V_{out}(t) - V_{DD} = -\frac{C_a}{C_L}(V_{DD} - V_{Tn}(V_X))$$

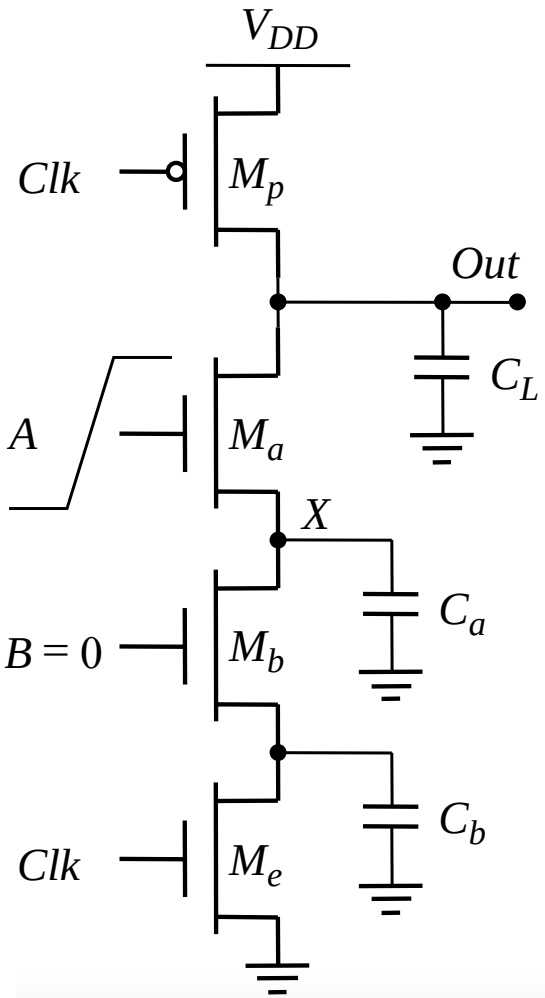
case 2) if $\Delta V_{out} > V_{Tn}$

V_{OUT} and V_X then reach the same value…..

$$\Delta V_{out} = -V_{DD}\left(\frac{C_a}{C_a + C_L}\right)$$

Which of these scenarios is valid?

Charge Sharing



Initial conditions: $V_{out}(t=0)=V_{DD}$ and $V_x(t=0)=0$

2 possible scenarios:

$\Delta V_{out} < V_{Tn}$ ….case I

$\Delta V_{out} > V_{Tn}$ ….case II

Which of these scenarios is valid?

First find the capacitance ratio: C_a/C_L

The boundary condition between the two cases can be determined by setting $\Delta V_{out} = V_{Tn}$. Hence,

$$\frac{C_a}{C_L} = \frac{V_{Tn}}{V_{DD} - V_{Tn}}$$

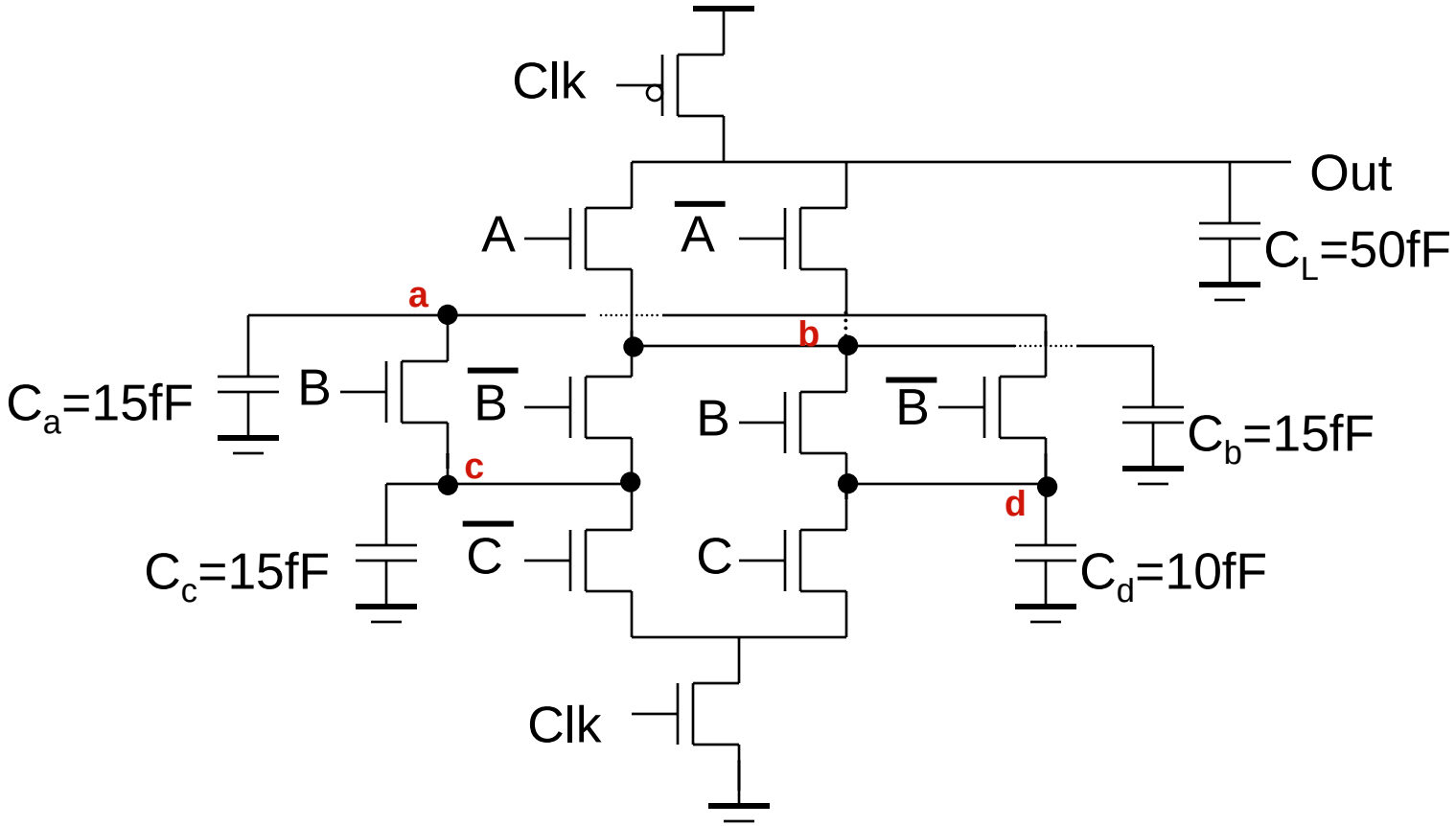
Case I holds when the C_a/C_L ratio is smaller than the value defined above, otherwise Case II holds.

Overall, it is desirable to keep $\Delta V_{out} < |V_{Tp}|$ ---since the output of dynamic gate might be connected to a static inverter---low level of V_{out} will cause static power consumption. Also, V_{out} must not go below V_M of the inverter.

Charge Sharing Example

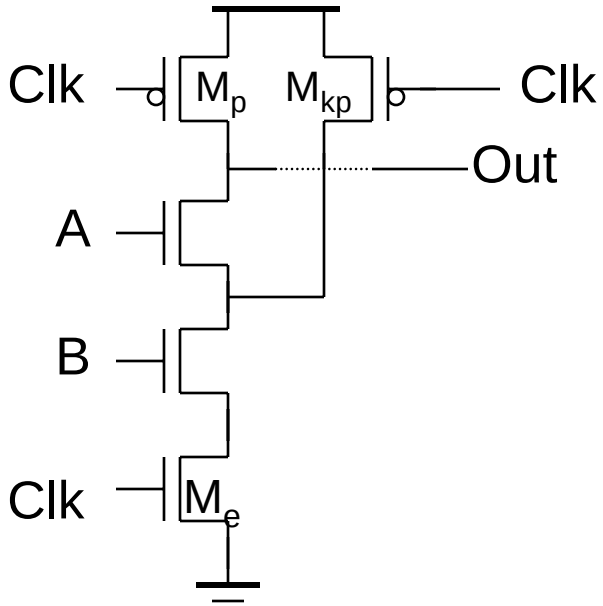
Dynamic 3-input EXOR gate

$$Out = A \oplus B \oplus C$$



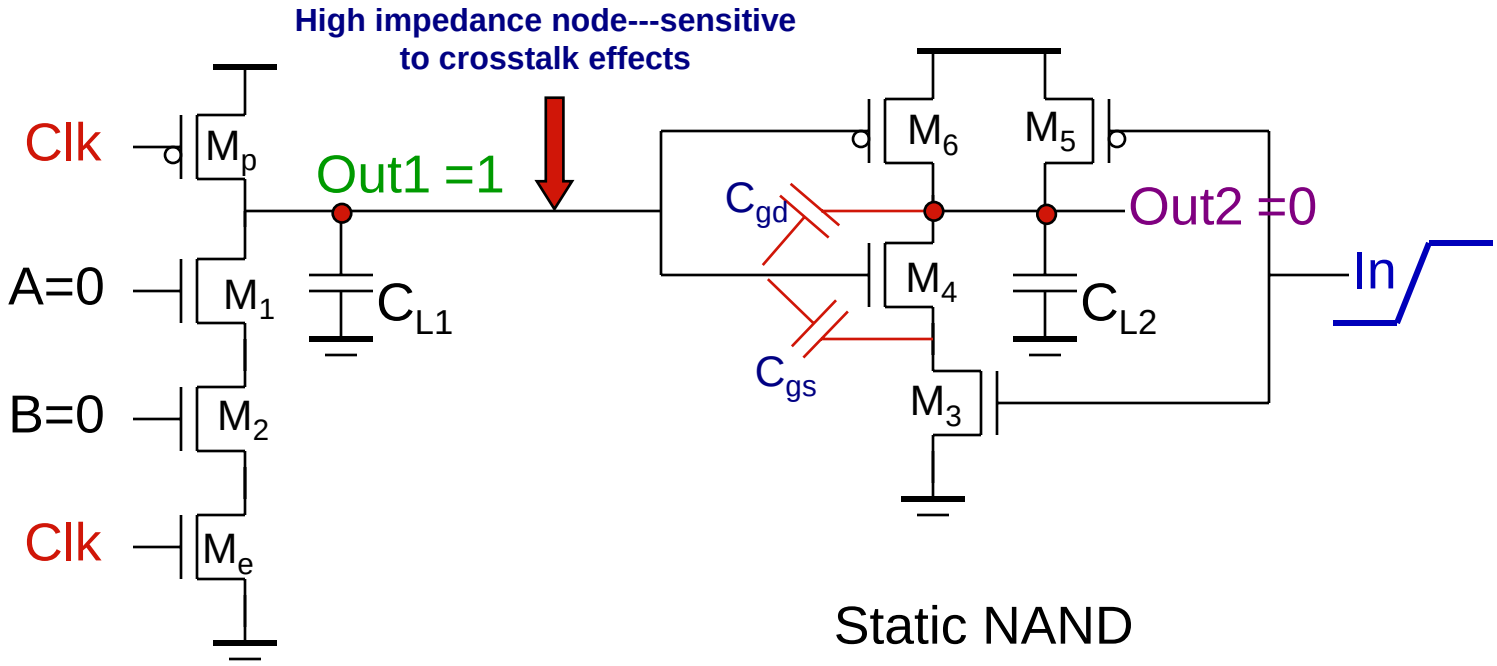
Worst case change in Output is obtained by exposing the maximum number of internal capacitances to the output: this happens for $\overline{A}BC$ or $A\overline{B}C$

Solution to Charge Redistribution



Precharge internal nodes (to V_{DD}) using a clock-driven transistor (at the cost of increased area and power)

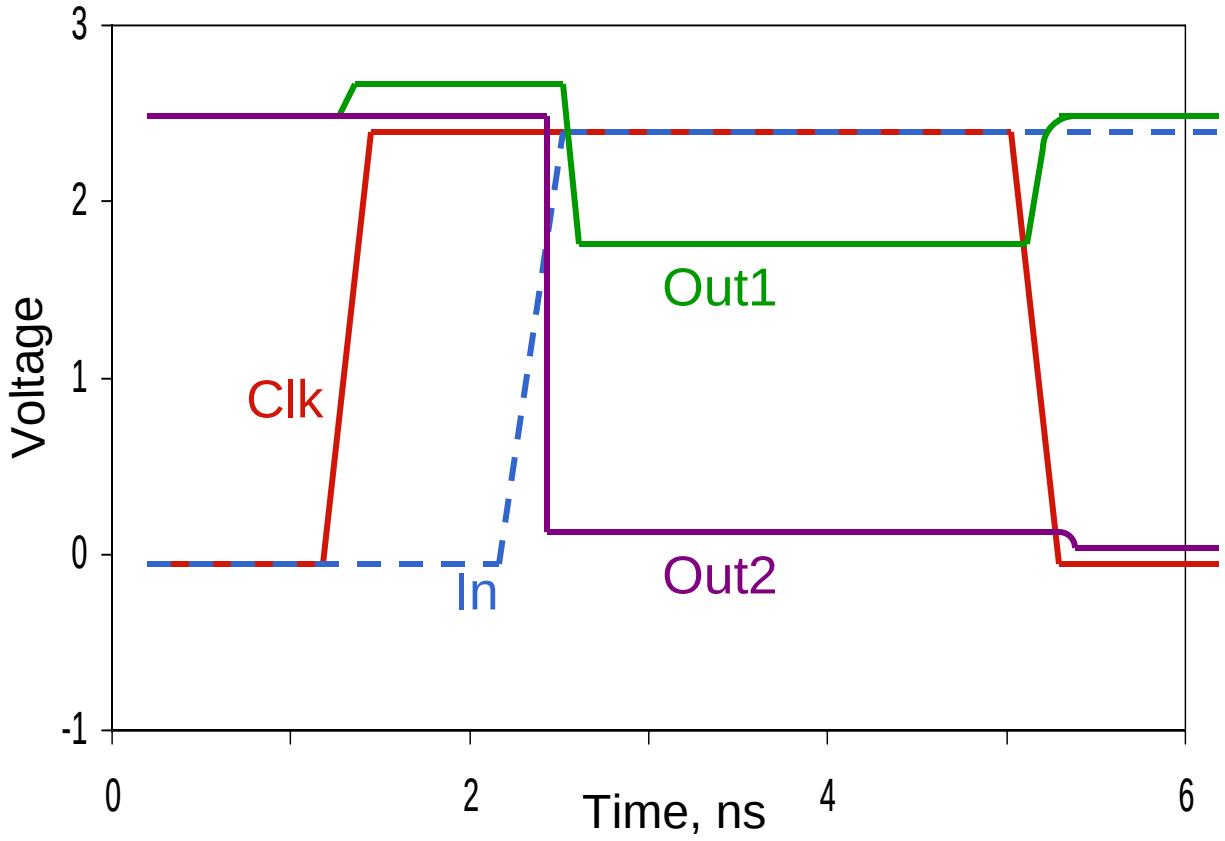
Issues in Dynamic Design 3: Backgate Coupling



Capacitive coupling between dynamic node Out1 and H-L transition at Out2 through the gate-drain and gate-source capacitance of M4

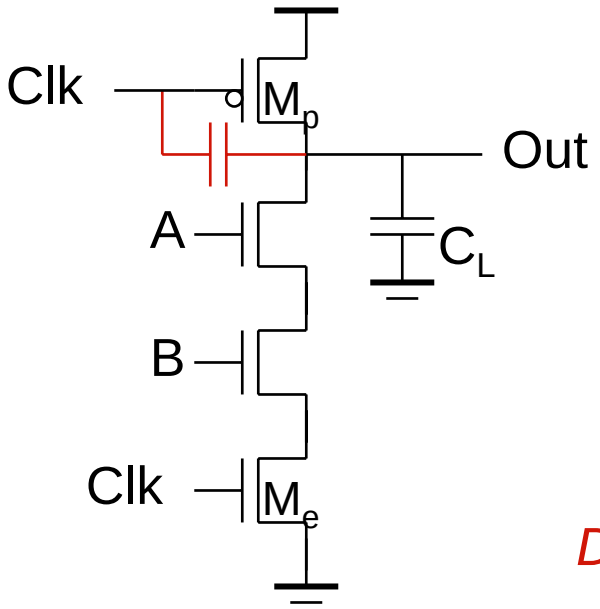
Backgate Coupling Effect

Simulation result



Coupling causes dynamic node Out1 to drop significantly, which further prevents Out2 from dropping all the way to zero---static power dissipation.

Issues in Dynamic Design 4: Clock Feedthrough



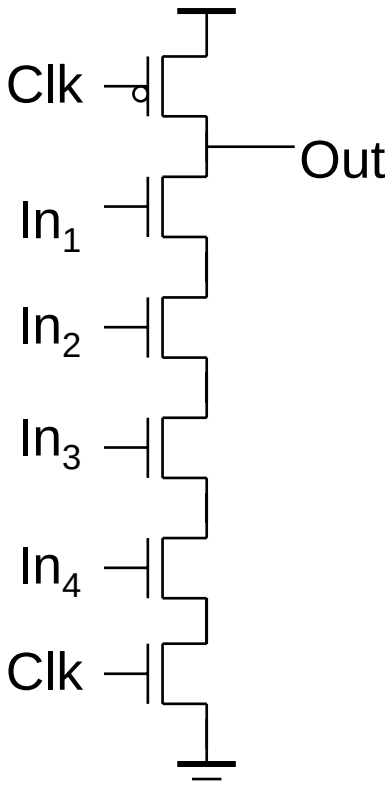
Coupling between Out and Clk input of the precharge device due to gate to drain capacitance (includes both overlap and channel).

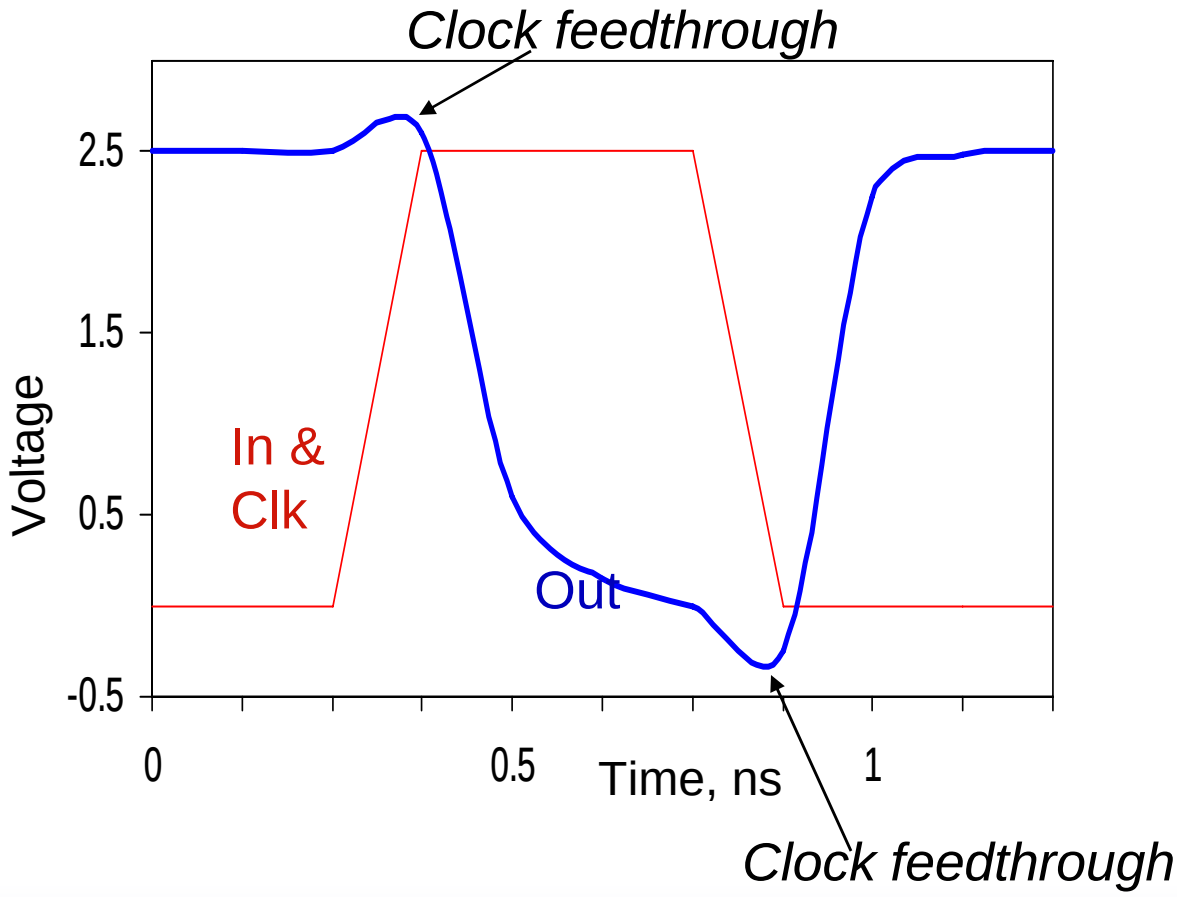
Hence, voltage of Out can rise above V_{DD} on the L-H Clk transition (assuming PDN is off). The fast rising (and falling edges) of the clock couple to Out.

Dynamic circuits need careful simulation!

Clk feedthrough can cause normally reverse biased junction diodes of the precharge transistor to become forward biased---causing electron injection into the substrate that can be collected by a nearby high-impedance node in the 1 state, eventually resulting in faulty operation.

Clock Feedthrough



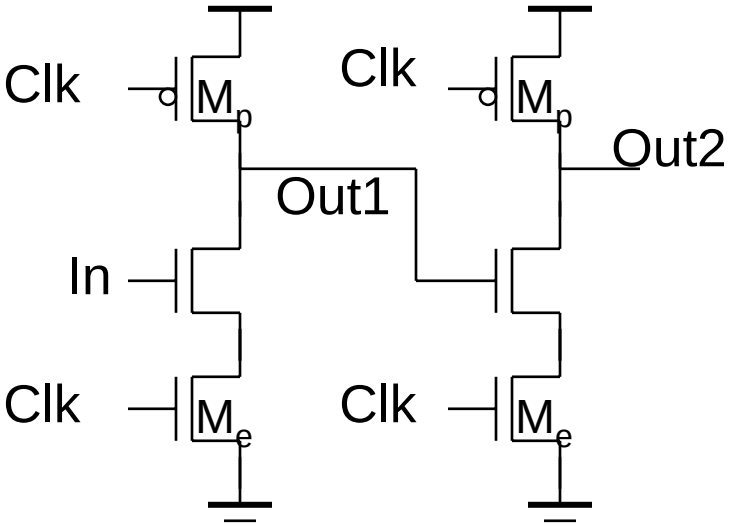


Other Effects

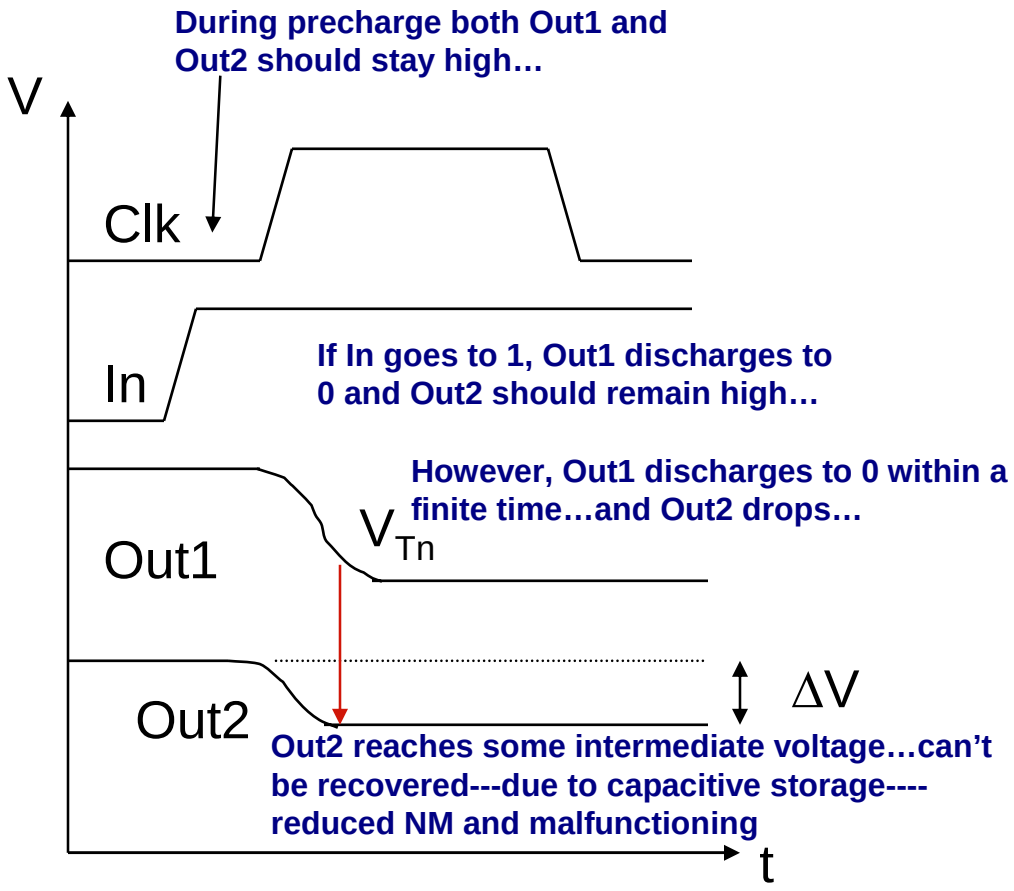
- Capacitive coupling
- Substrate coupling
- Minority charge injection
- Supply noise (ground bounce)

Cascading Dynamic Gates

Simple cascoding doesn't work…



As long as Out1>V_M (~V_{Tn}) of the second inverter, Out2 will decrease leading to reduced NMs

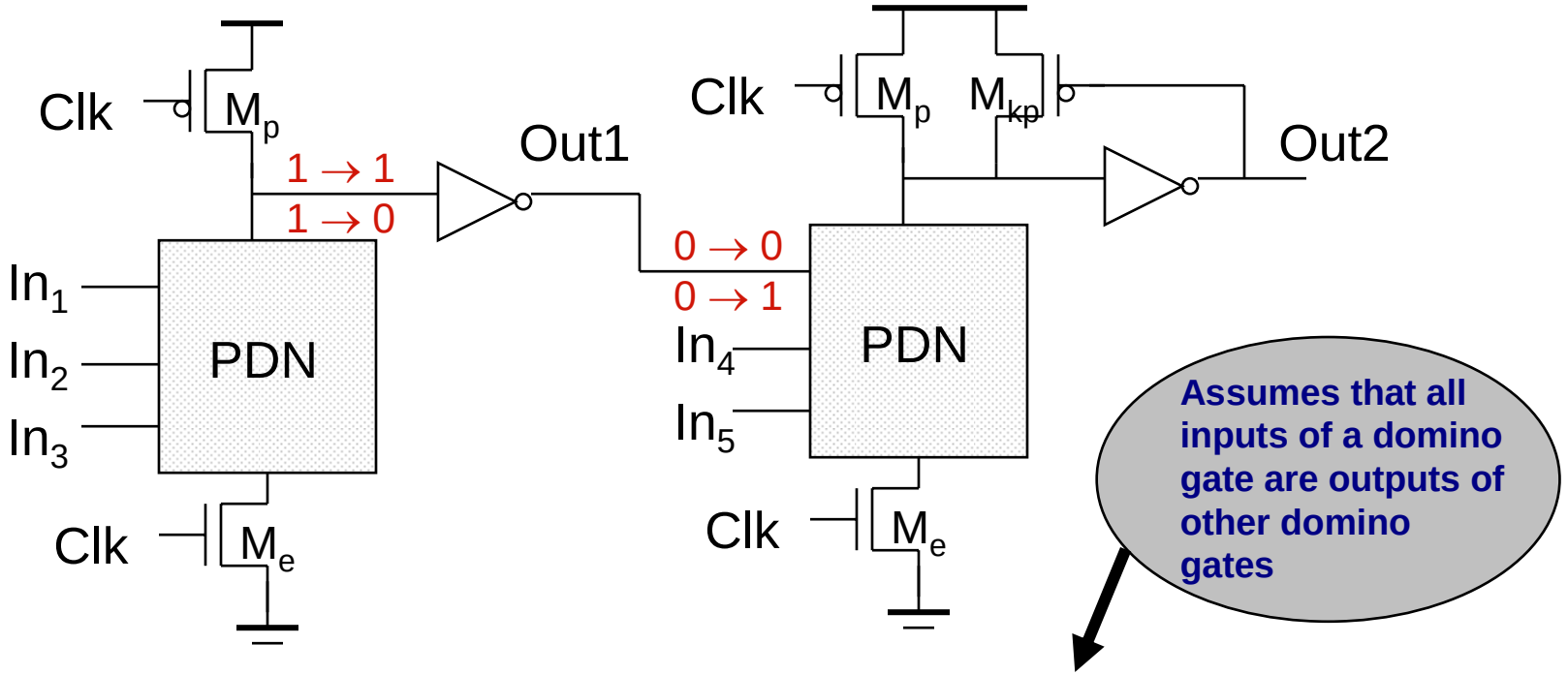


Solution: Set all inputs to 0 during precharge

For correct operation only 0 → 1 transitions should be allowed at inputs!

Domino Logic

An n-type dynamic logic followed by a static inverter…

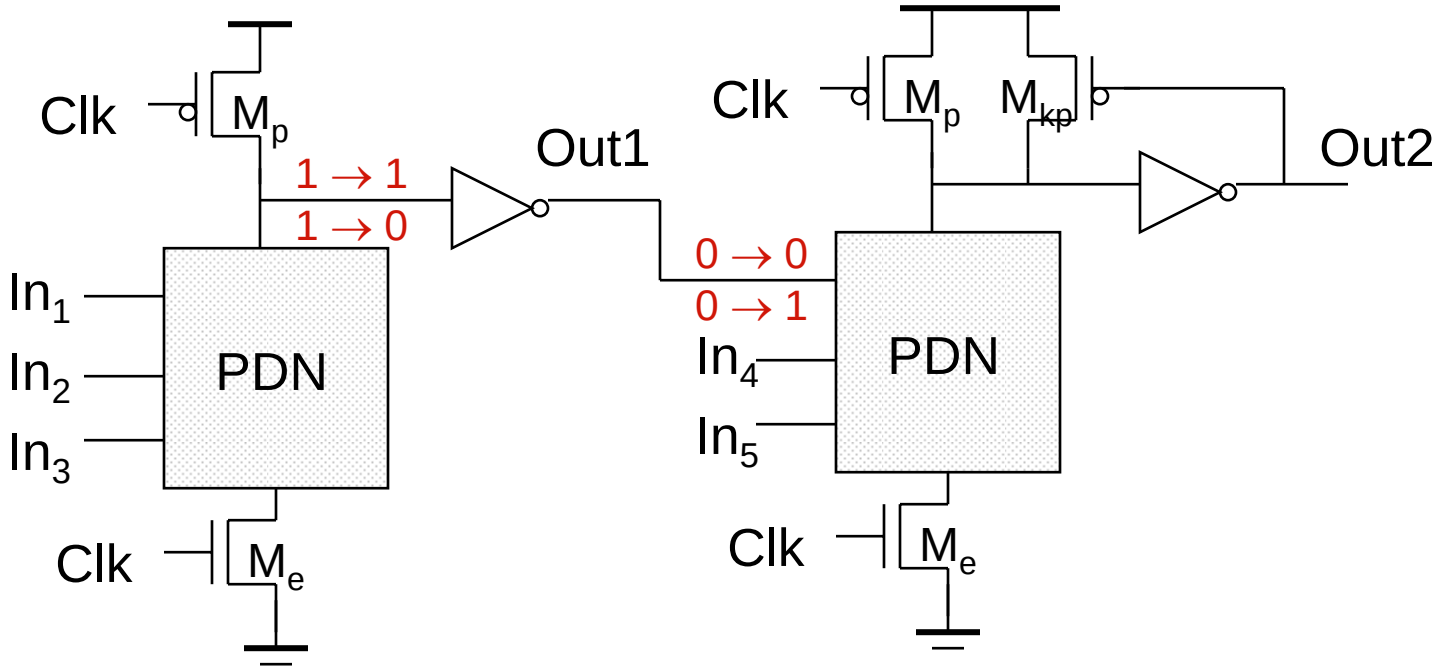


All inputs (are outputs of other Domino gates) are set to 0 at the end of precharge phase

Only 0 to 1 transition at the inputs during evaluation phase:
during evaluation, dynamic gate conditionally discharges and the output of the inverter makes a conditional transition from 0 to 1.

Domino Logic

An n-type dynamic logic followed by a static inverter…

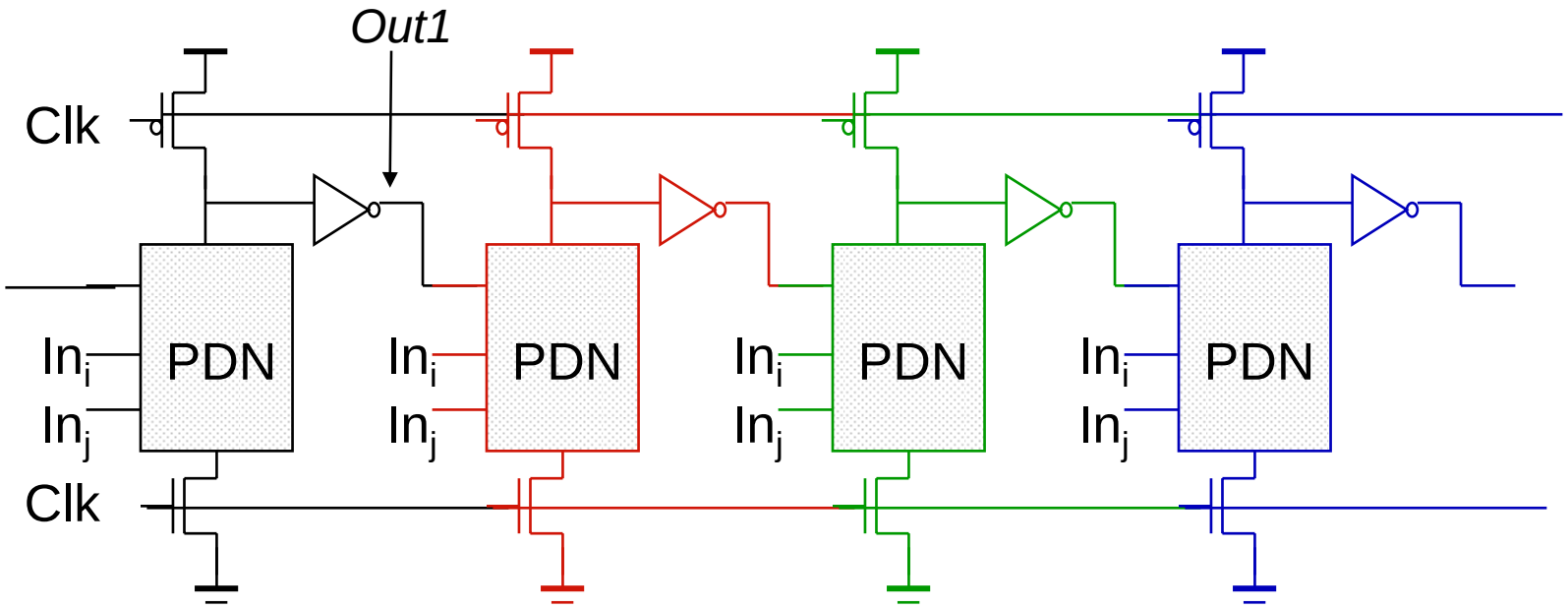


The static inverter reduces the capacitance of the dynamic output node by separating internal and load capacitances…..it also increases the NM (due to the low-impedance output)

The inverter can also be used to drive a keeper device to combat leakage and charge redistribution.

Why Domino?

A Domino chain



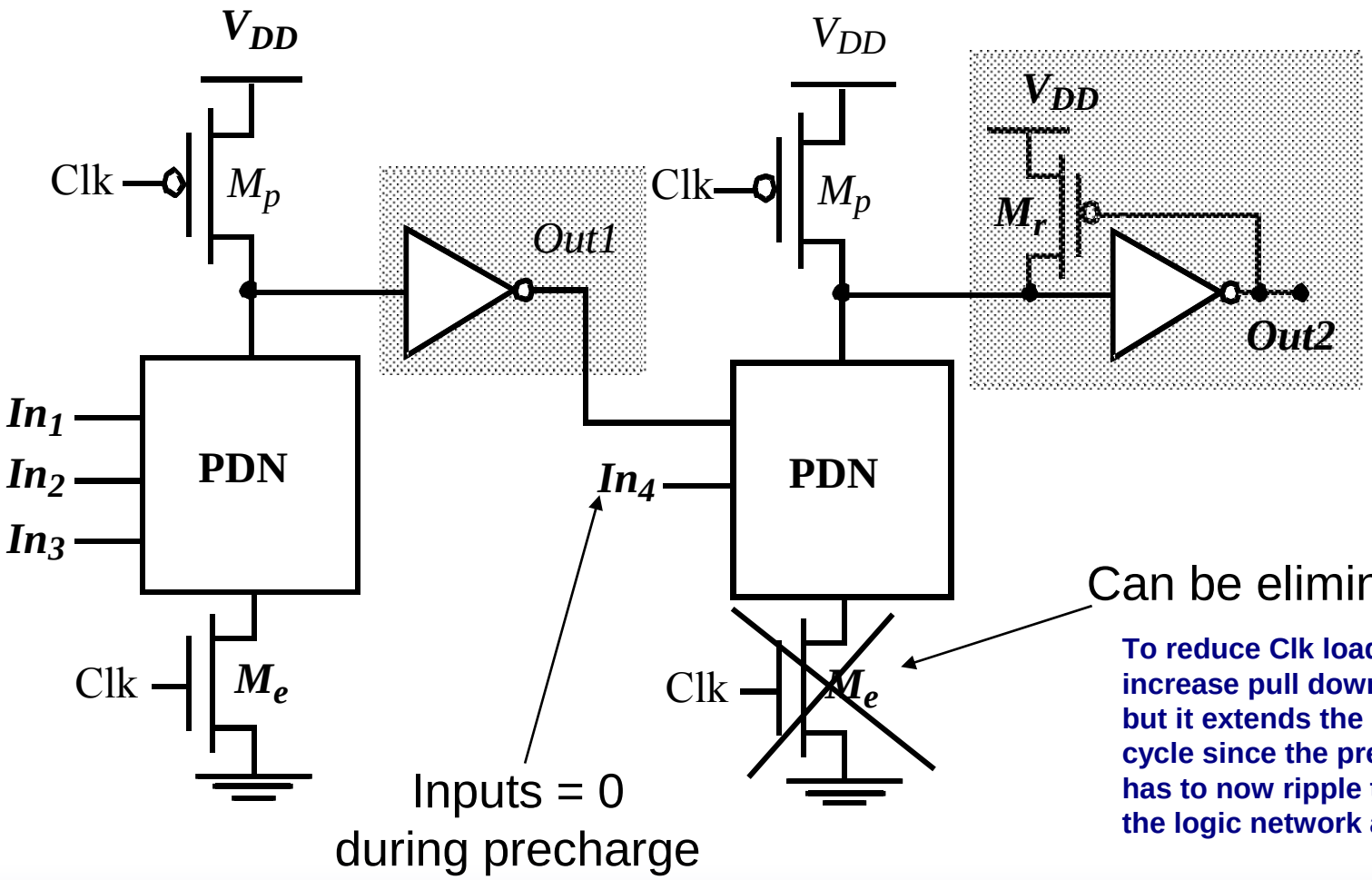
Precharge: all inputs=0

Evaluation: Output of domino1 either stays at 0 or makes a transition from 0 to 1, affecting the second gate. This effect might ripple through the whole chain…***like a line of falling dominos!***

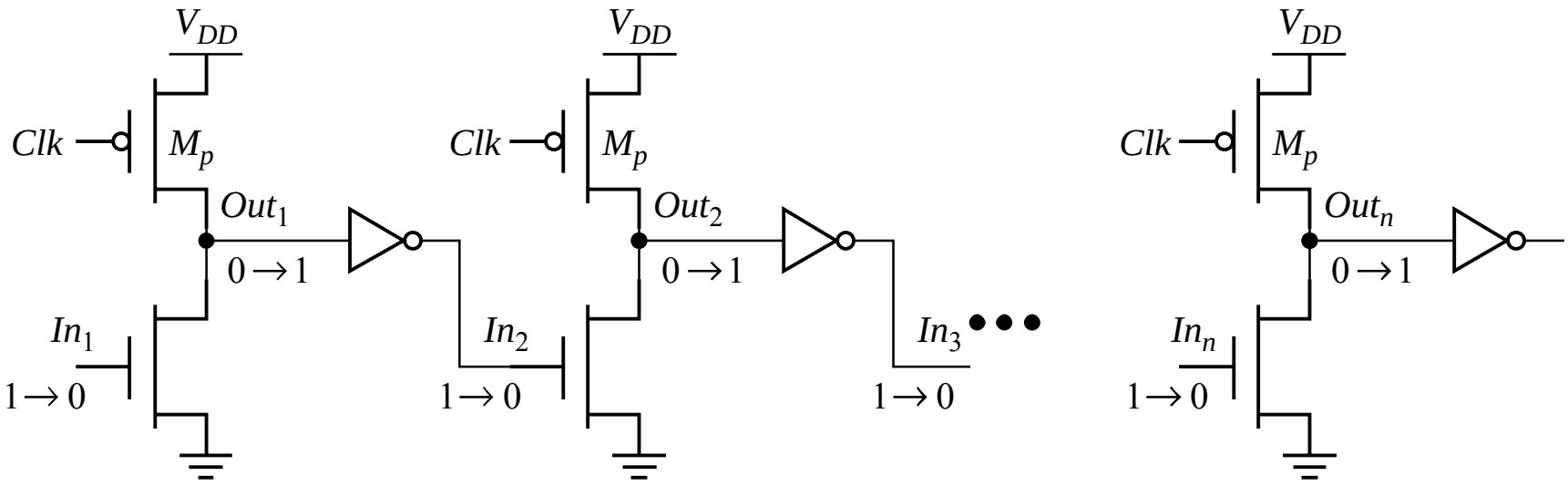
Properties of Domino CMOS Logic

- Only non-inverting logic can be implemented (due to the static inverter)
  - Major limitation
  - Can be overcome using dual-rail domino (an expensive solution)
- Very high speed
  - Only rising edge delays, and $t_{pHL}=0$
  - static inverter can be skewed to match the fanout, which is already much smaller than in the complimentary case, since only a single gate capacitance needs to be accounted for per fan-out gate.
  - Input capacitance reduced – smaller logical effort

Designing with Domino Logic



Footless Domino



If In_1=1, out_1=0 and In_2=1

On the falling edge of CLK, let In_1 =0: but it takes two gate delays for In_2 to be 0, during which second gate cannot pre-charge its output (Out_2), since PDN is fighting the precharge-PMOS

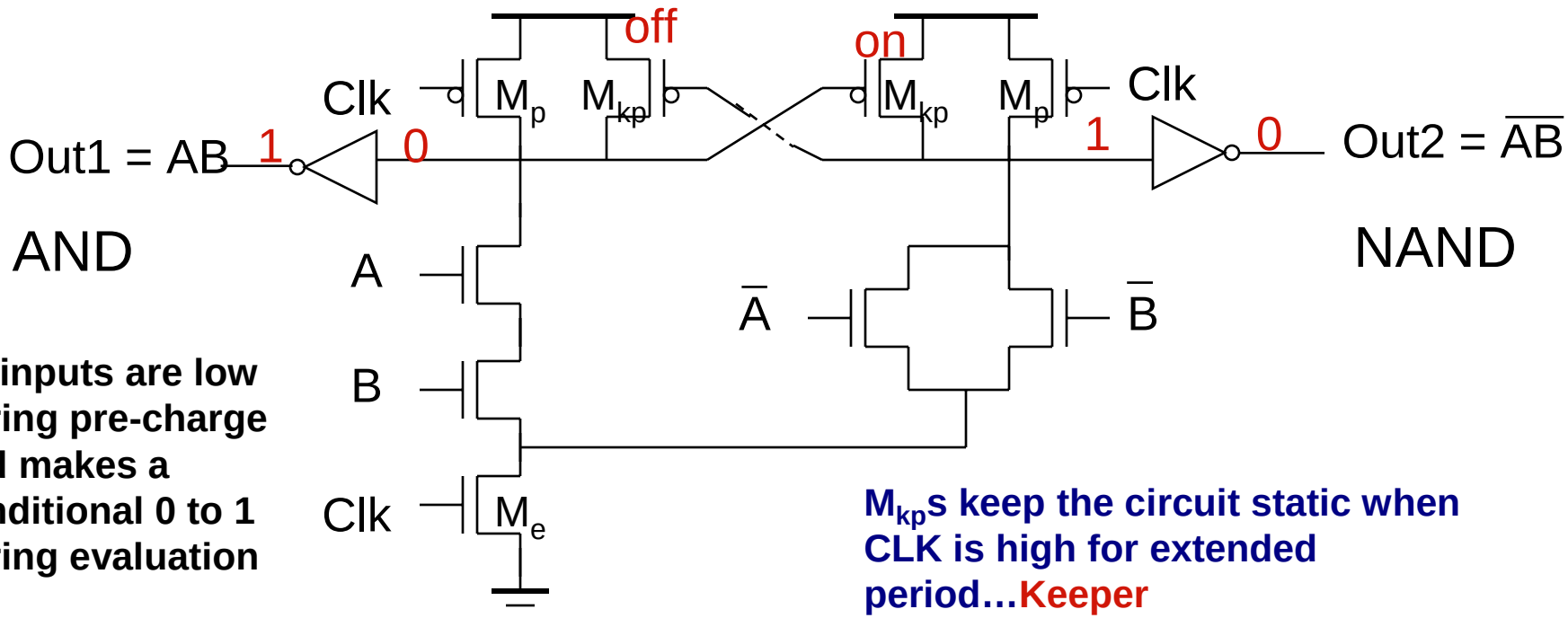
Time taken to precharge equals the critical path delay!
Better to use the evaluation device….

Pre-charge is rippling – short-circuit current
A solution is to delay the clock for each stage

Differential (Dual Rail) Domino

Overcomes the non-inverting property of Domino Logic: used commercially in several microprocessors

Uses a pre-charged load….



All inputs are low during pre-charge and makes a conditional 0 to 1 during evaluation

M_{kp}s keep the circuit static when CLK is high for extended period…Keeper

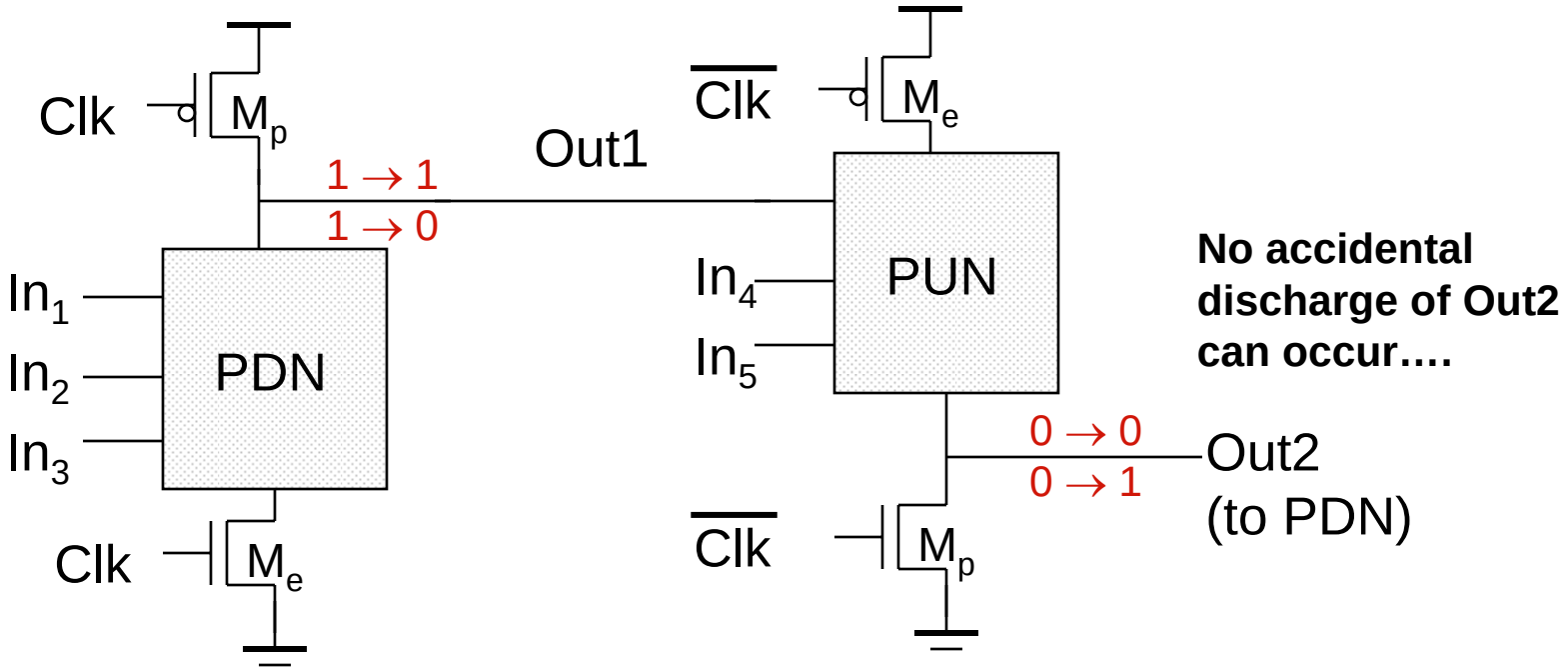
Possible to implement any arbitrary function….but comes at the expense of increased power since a transition is guaranteed every CLK cycle irrespective of the input values….either Out1 or Out2 must make a 0 to 1 transition.

np-CMOS

Alternative to cascading dynamic gates….uses n-type and p-type dynamic logic

Exploits duality between n-tree and p-tree logic gates to eliminate cascading problem

No extra inverter at the outputs….unless output of n-tree (p-tree) needs to be connected to another n-tree (p-tree) gates



Only 0 → 1 transitions allowed at inputs of PDN

Only 1 → 0 transitions allowed at inputs of PUN

Drawback: p-tree gates are slower than the n-tree gates…needs proper skewing of PMOS….area penalty

No buffers---so dynamic nodes need to be routed between gates