ECE 225
High-Speed Digital IC Design

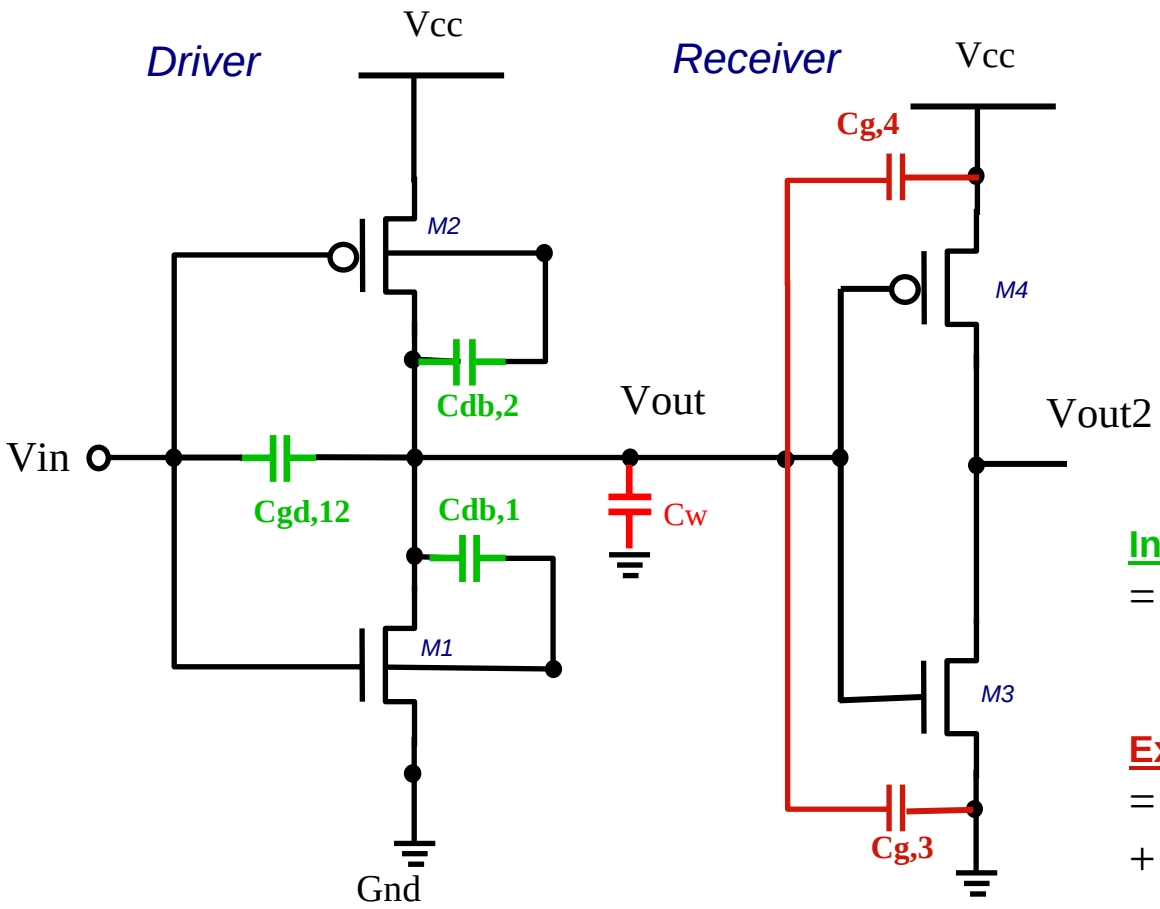
Lecture 4

Review of Basic Circuit Styles-2

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

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

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

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

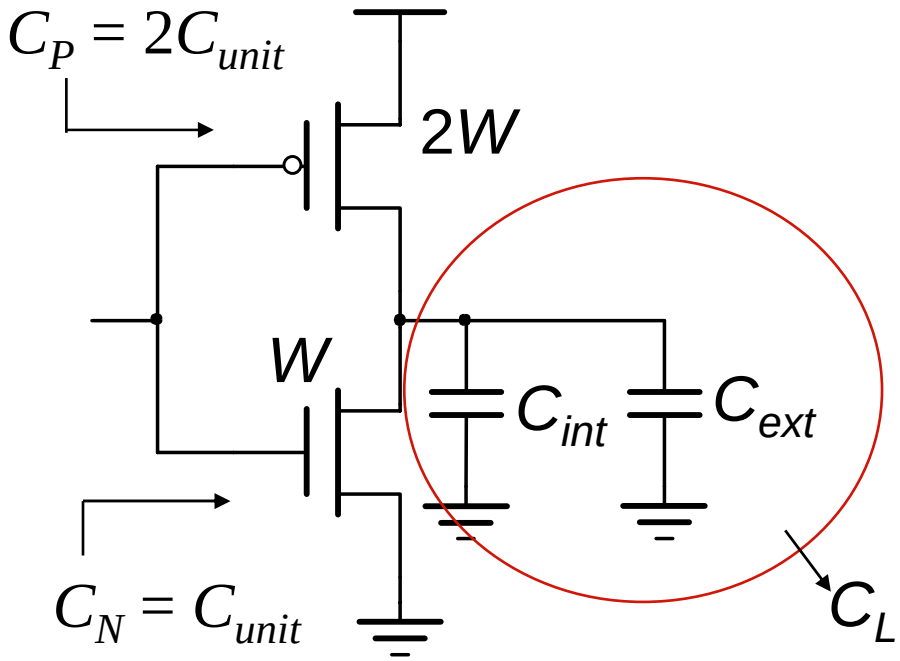
$$= 0.69\,R_{ref}\;C_{iref}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

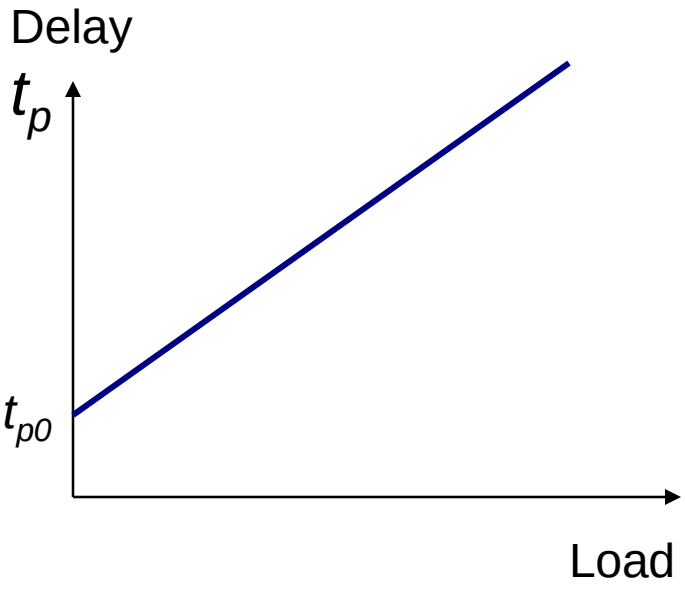
$$= t_{p0}\left(1+\frac{C_{ext}}{SC_{iref}}\right)$$

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

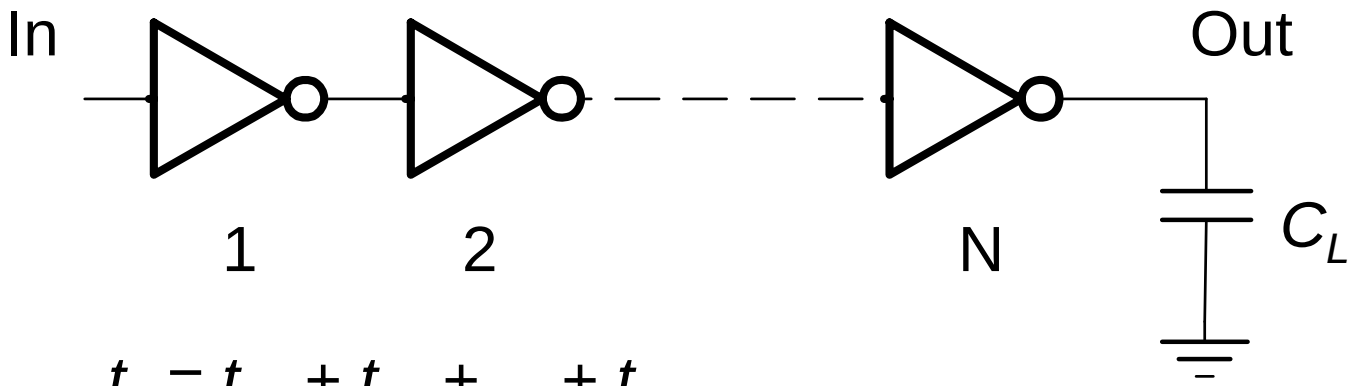




Delay (t_p) = $kR_W(C_{int} + C_{ext}) = kR_WC_{int} + kR_WC_{ext} = kR_W C_{int}(1+ C_{ext} /C_{int})$

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:

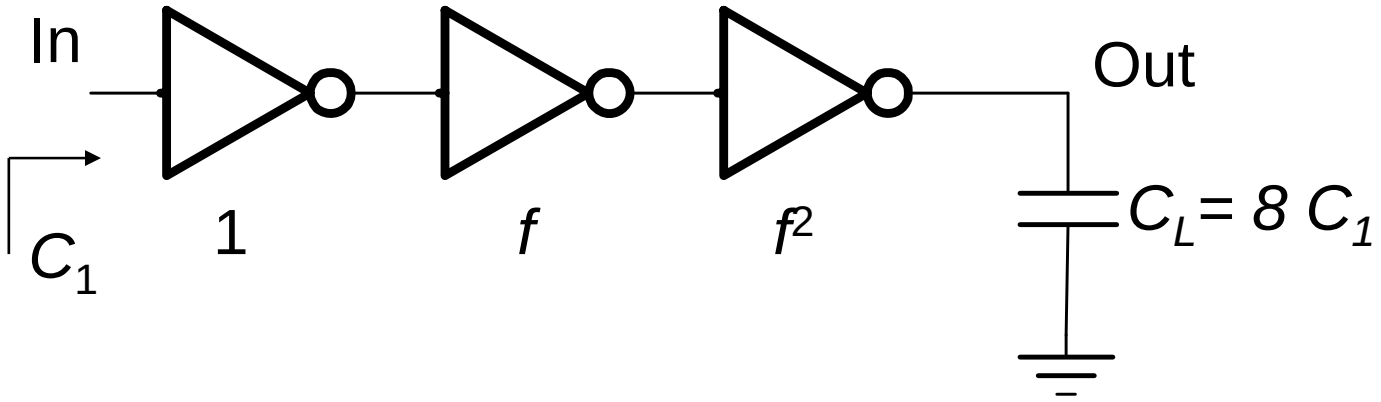
$$t_p = Nt_{p0}\left(1 + \sqrt[N]{F}/\gamma\right)$$

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….

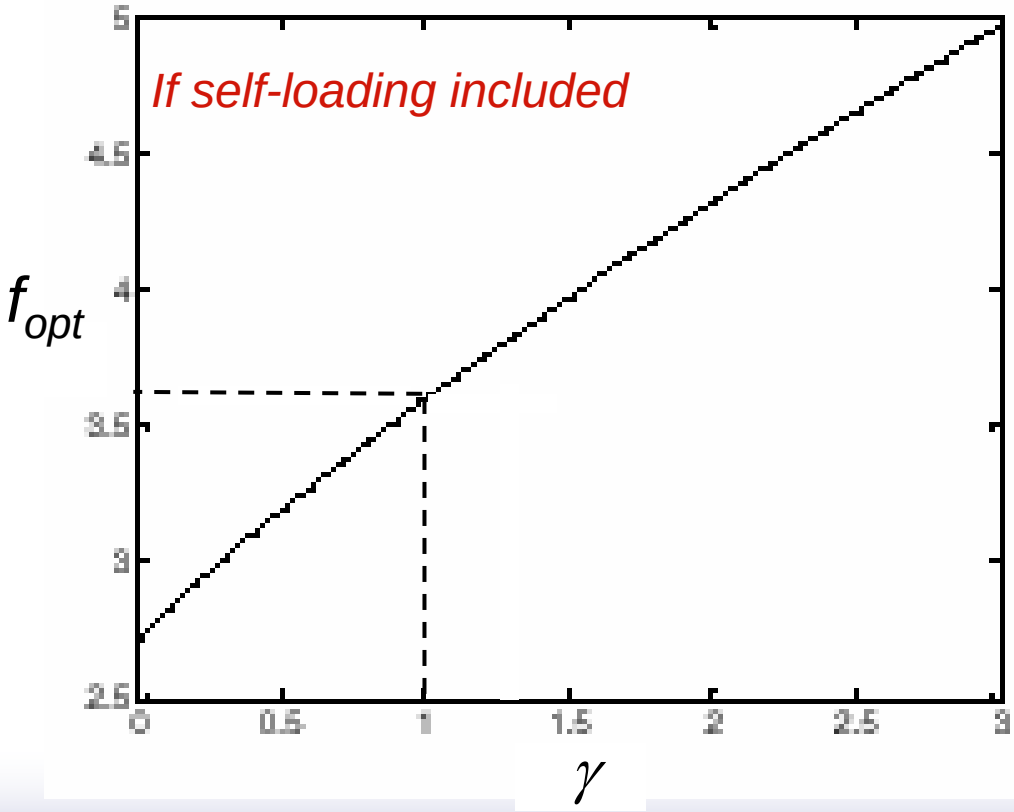
For $\gamma = 0$, $f = e$, $N = \ln F$

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

Optimum Effective Fanout f

Optimum *f* for given process defined by γ

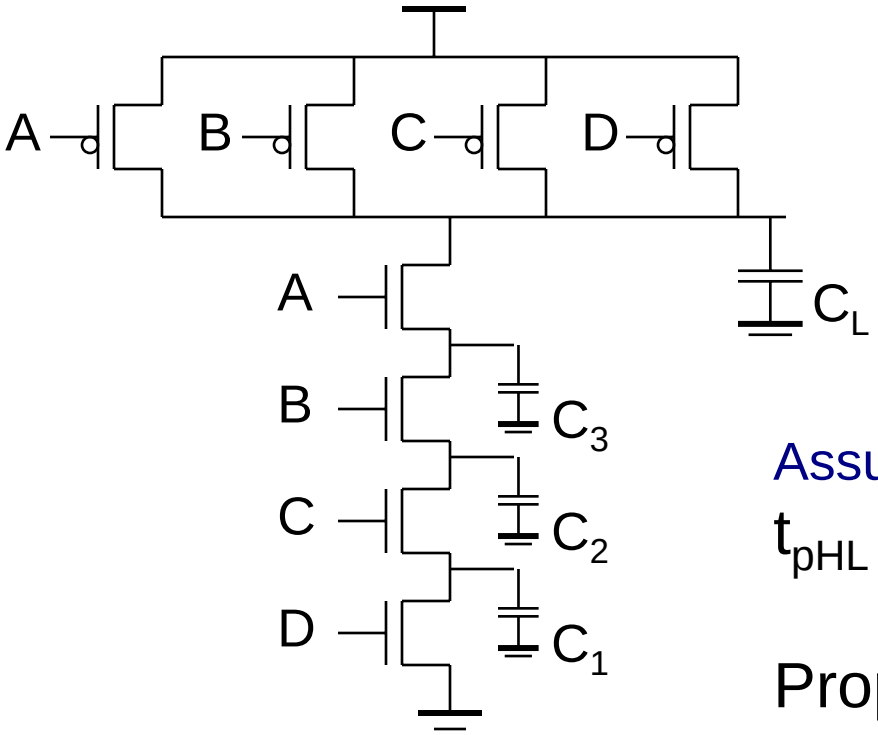
$$f = \exp\left(1 + \gamma / f\right)$$

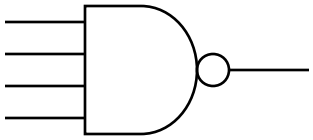


Optimum tapering factor:

f_{opt} = 3.6 for γ=1

Fan-In Considerations





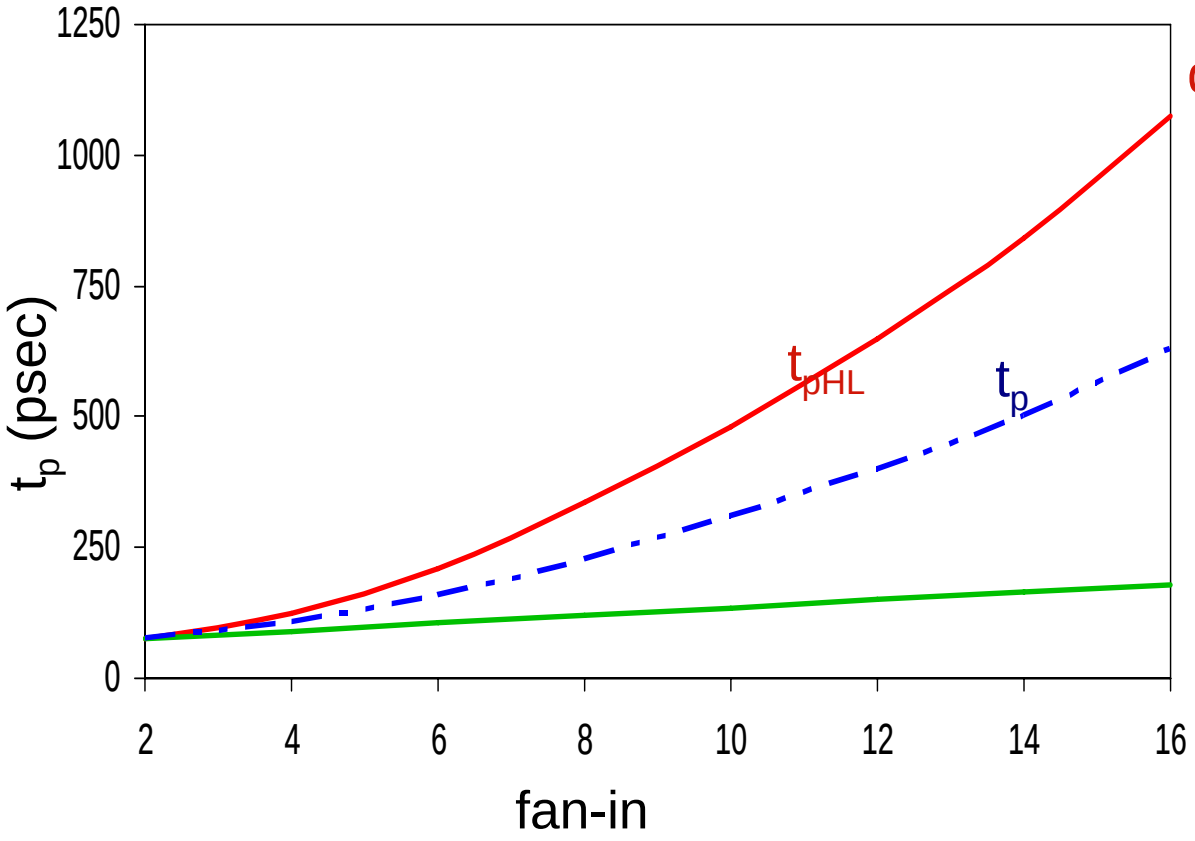
Distributed RC model (Elmore delay)

Assuming all NMOS are equal size

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

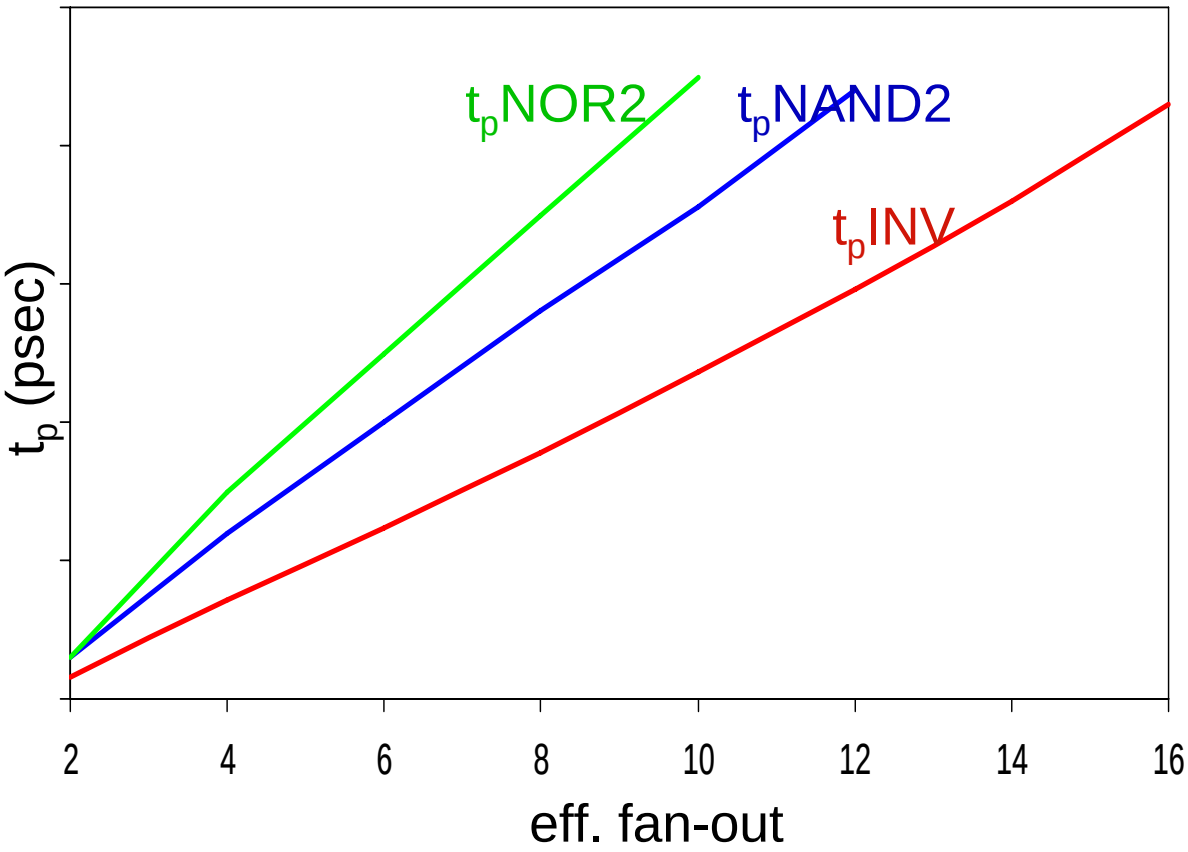
Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

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….

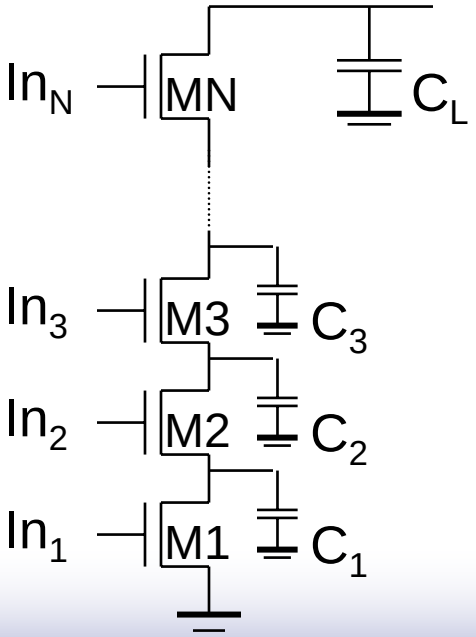
t_p as a Function of Fan-In and Fan-Out

- 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

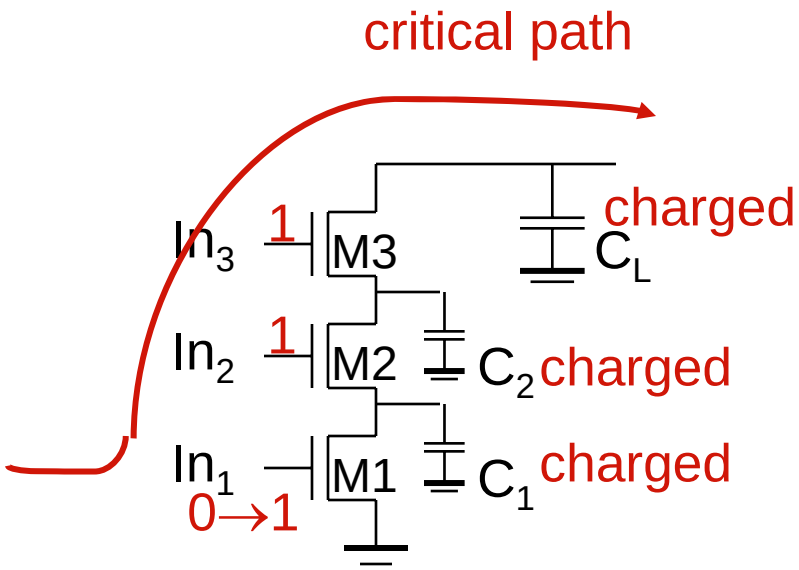
M1 > M2 > M3 > … > MN
(the transistor closest to the output is the smallest: has biggest R)

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

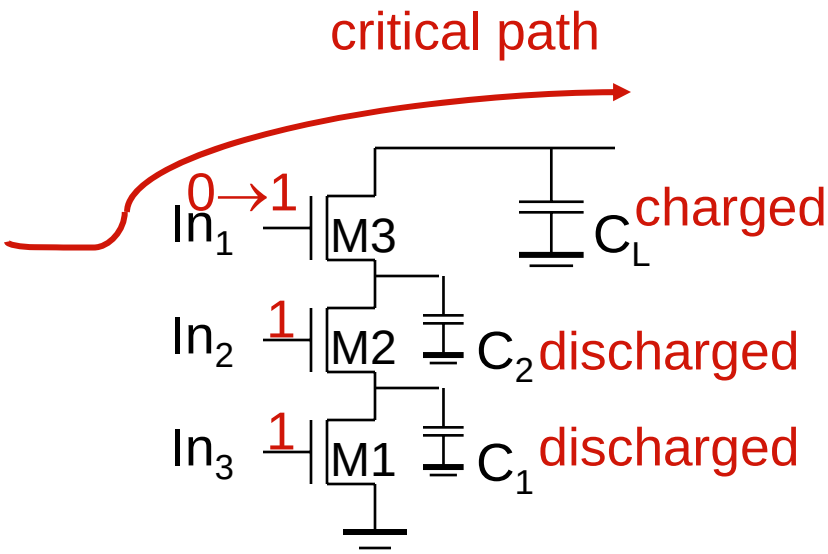
Fast Complex Gates: Design Technique 2

- 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



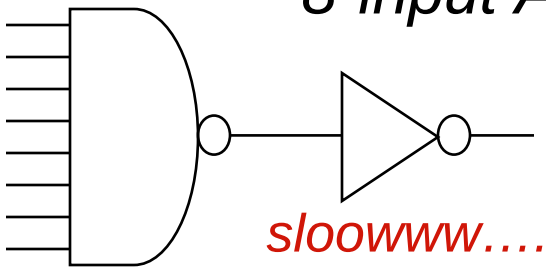
delay determined by time to discharge C_L

Fast Complex Gates: Design Technique 3

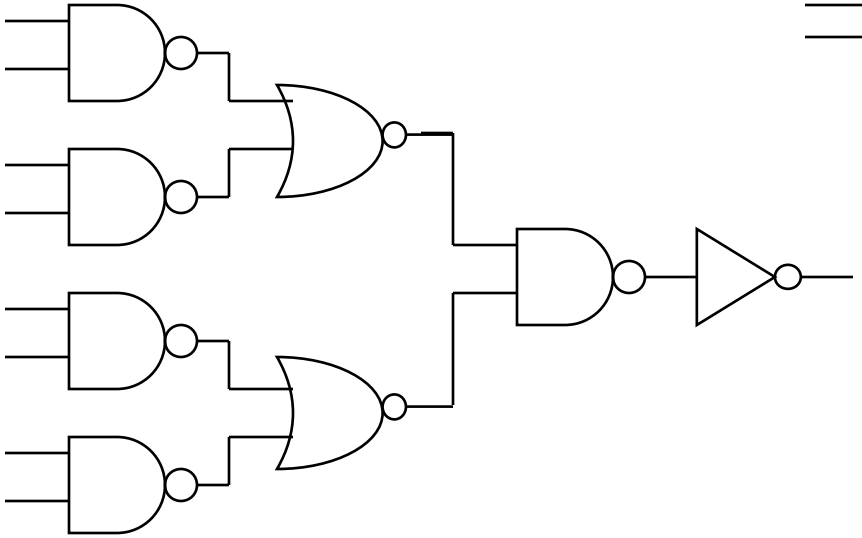
- Alternative logic structures

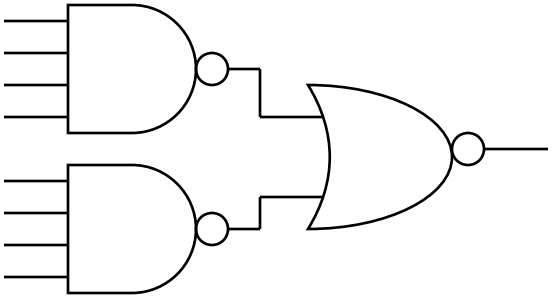
F = ABCDEFGH

8-input AND



sloowww….

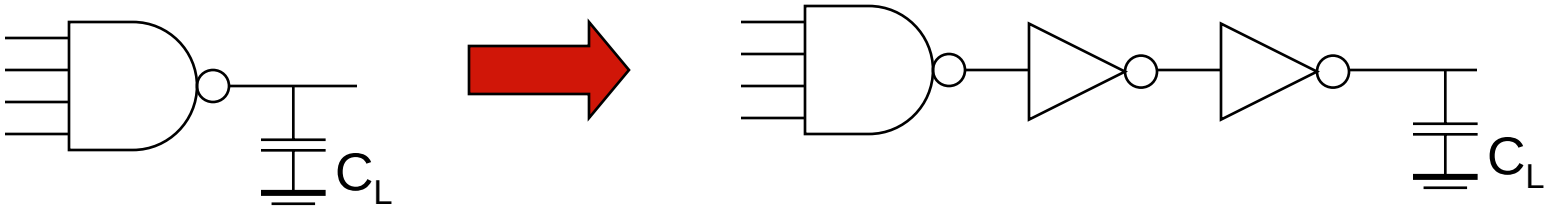




Using De Morgan's Law…

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

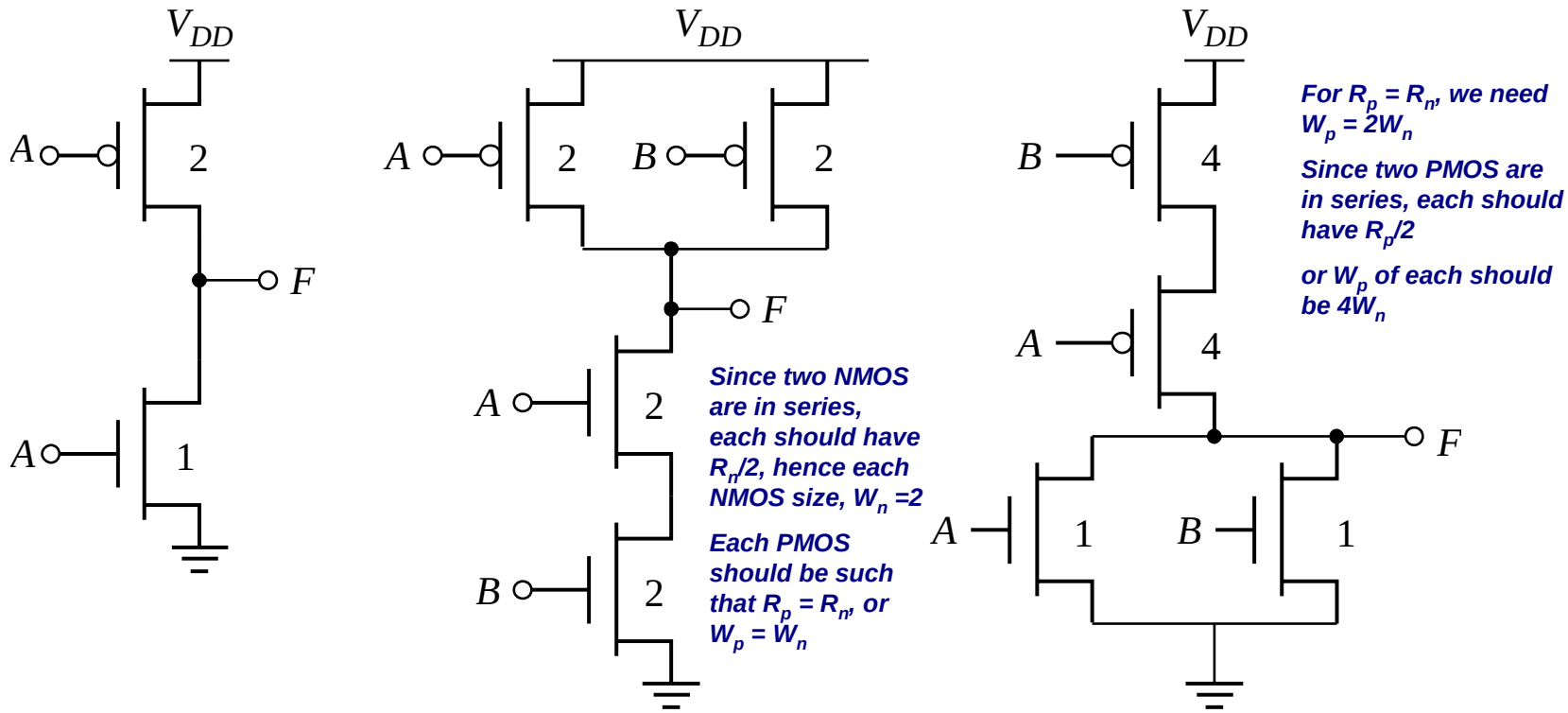
f – effective fanout *(ratio of external load and input cap. of gate)*
p – ratio of intrinsic delays of complex gate and inverter
(value increases with complexity of gate)
g – logical effort: how much more input capacitance is presented by the complex gate to deliver the same output current as an inverter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

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**

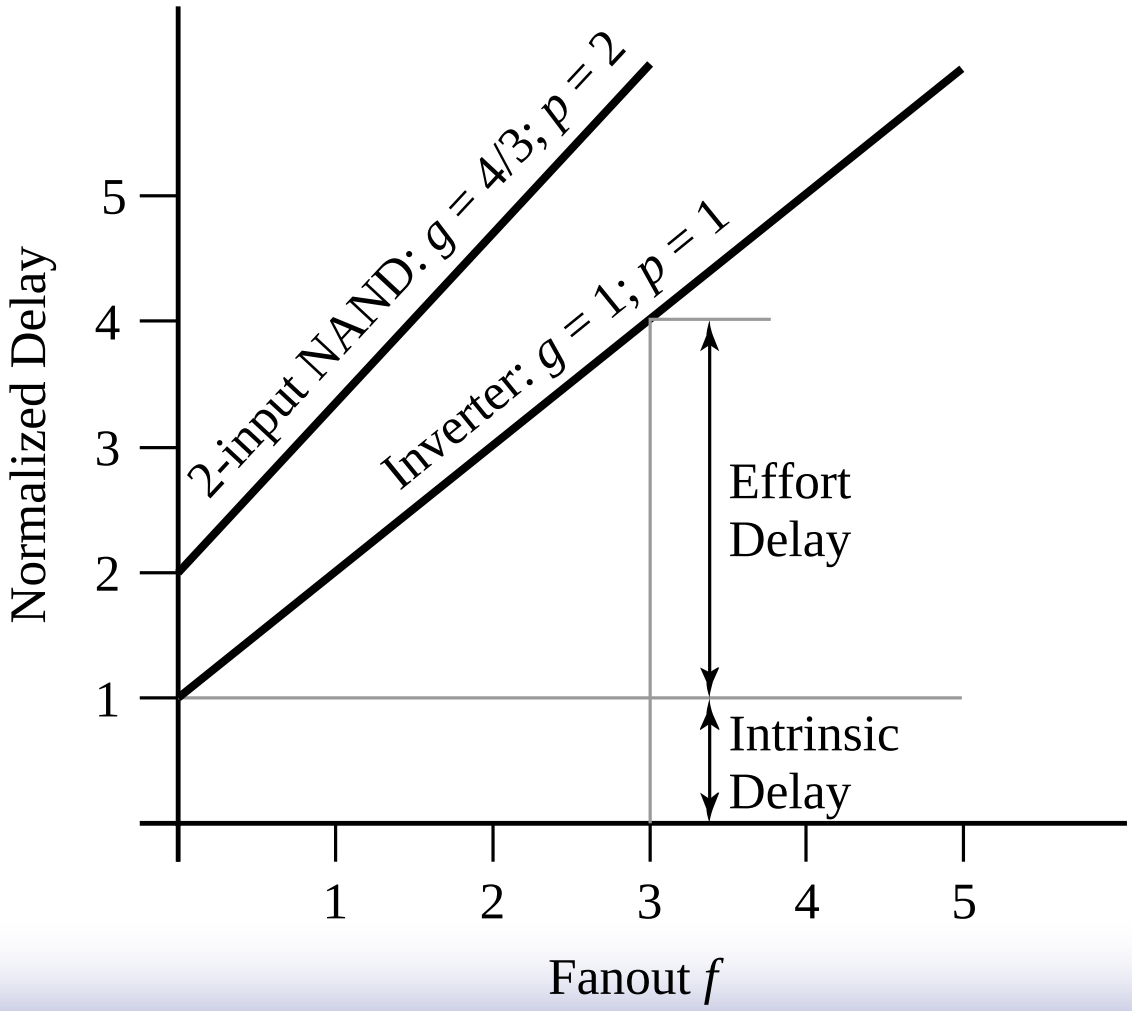
Effort delay (or gate effort):

$$h = g\,f$$

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

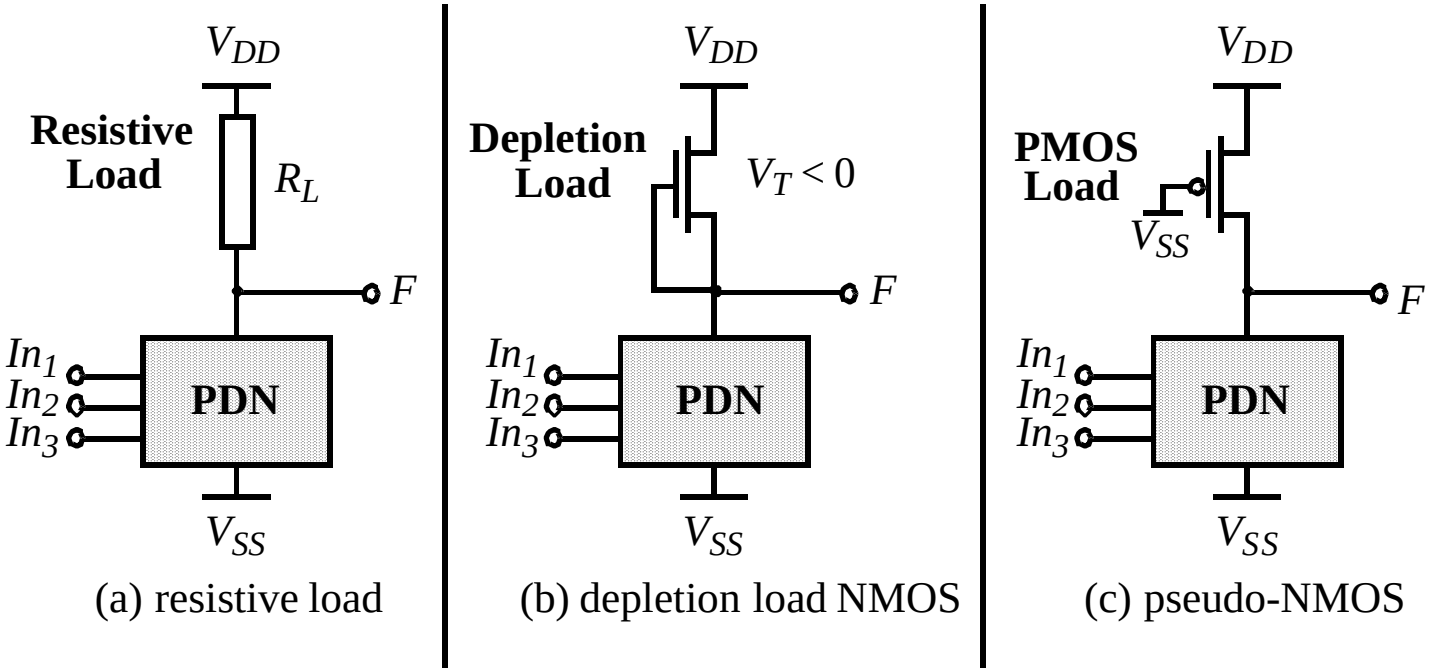
Logical Effort of Gates



Ratioed Logic

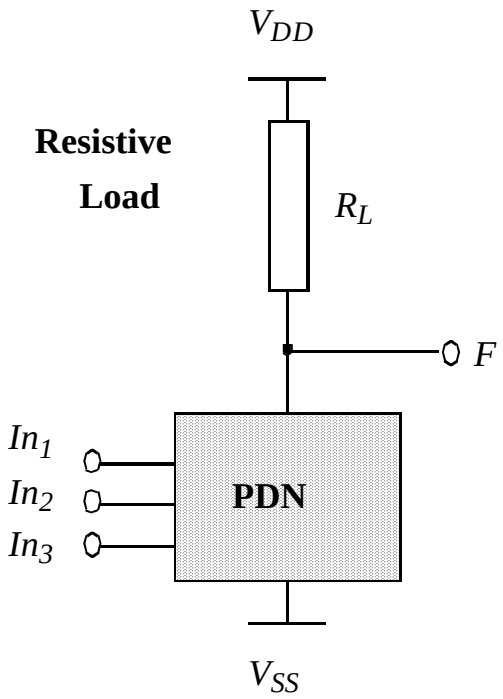
Ratioed Logic

Need N+1 transistors vs 2N for complementary CMOS



Goal: to reduce the number of devices over complementary CMOS

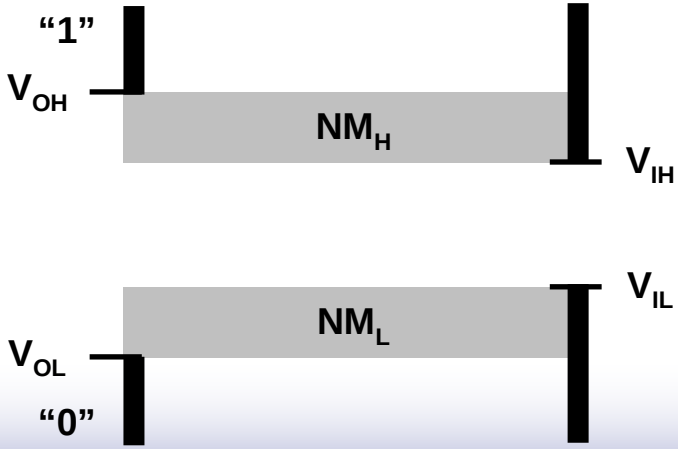
Ratioed Logic: Resistive Load



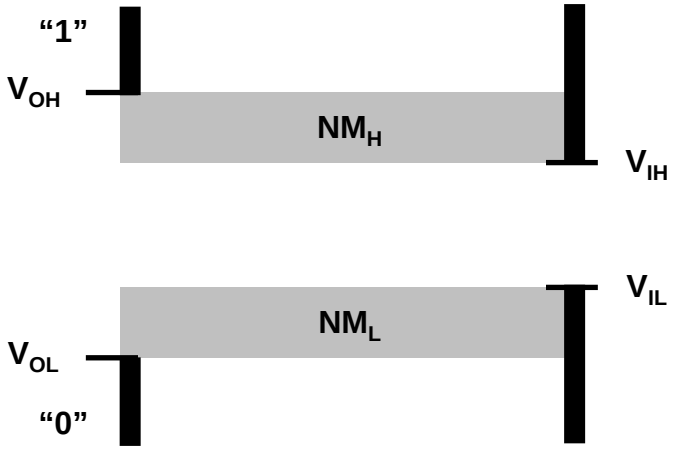
- **N transistors + Load**
- **$V_{OH} = V_{DD}$**
- **$V_{OL} = \frac{R_{PN}}{R_{PN} + R_L}$**

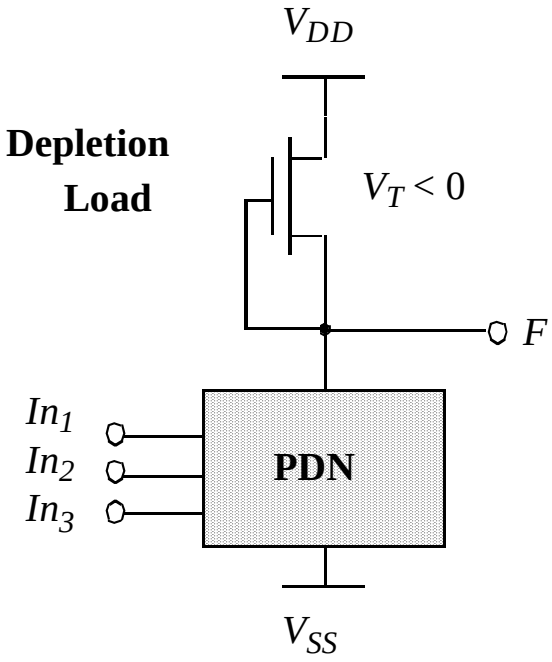
Ideally V_{OL} should be as small as possible. Hence, R_L should be large…

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

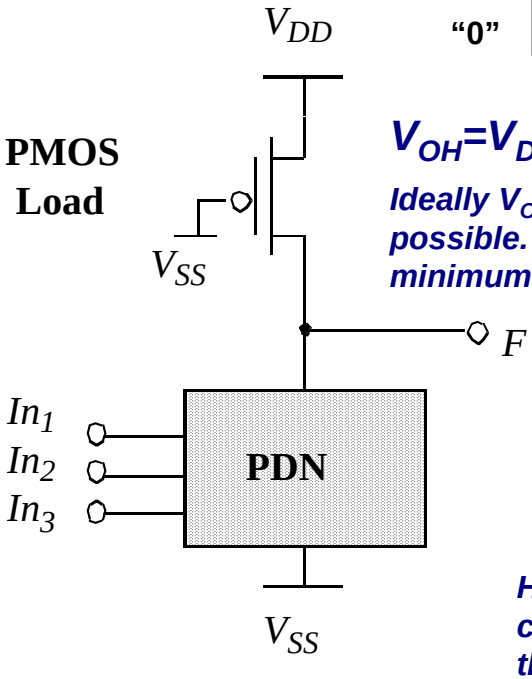


Active Loads





depletion load NMOS



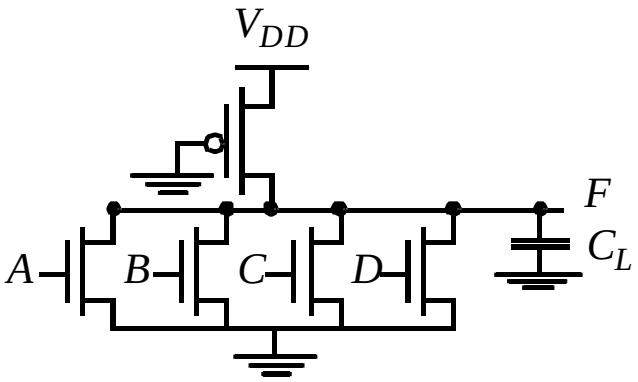
pseudo-NMOS

$V_{OH}=V_{DD}$ (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

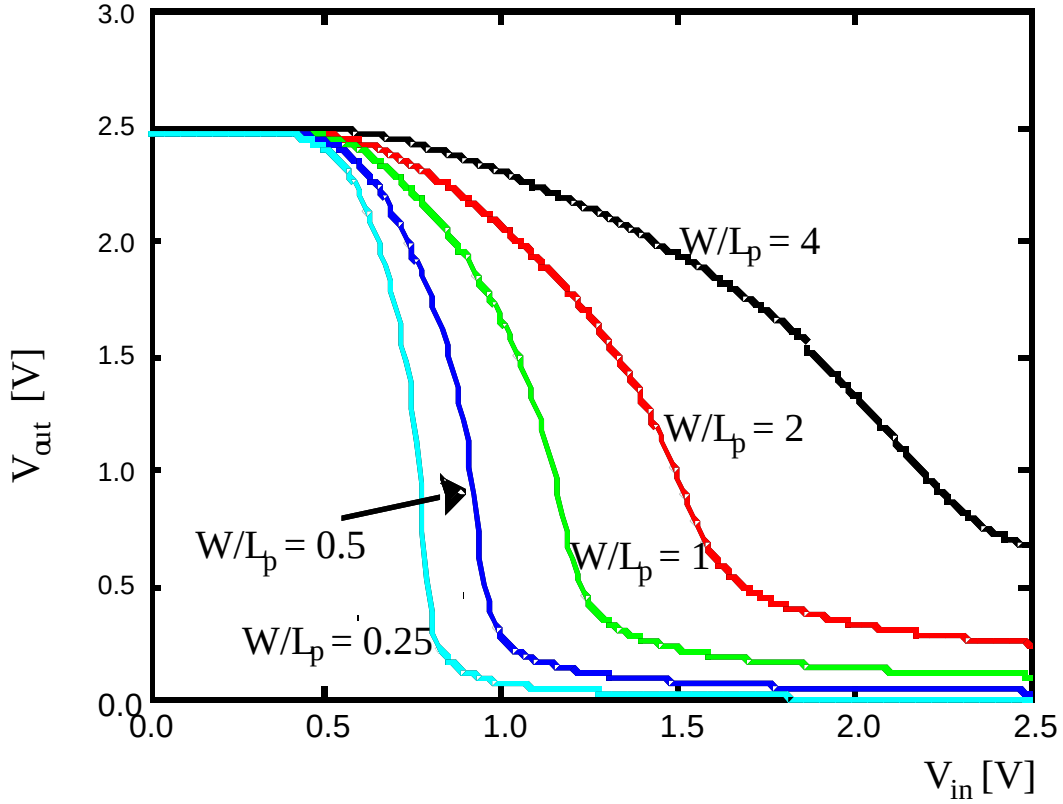
$$V_{OL} = \mu_p/\mu_n\, W_p/W_n\, V_{DSATp}$$

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!!!

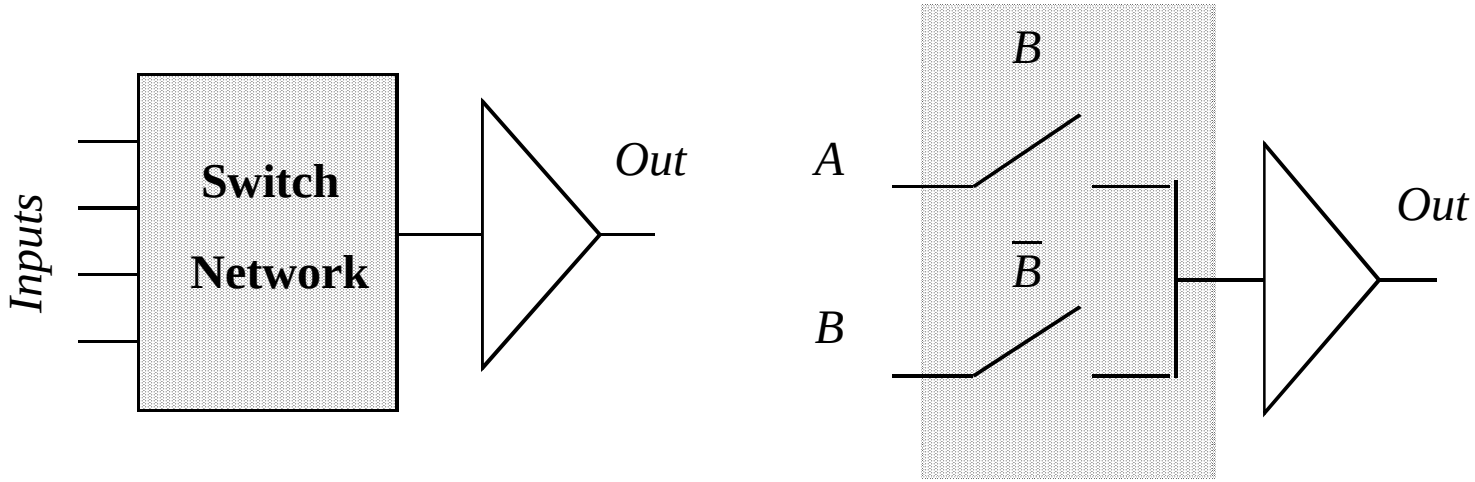
Pseudo-NMOS VTC

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}

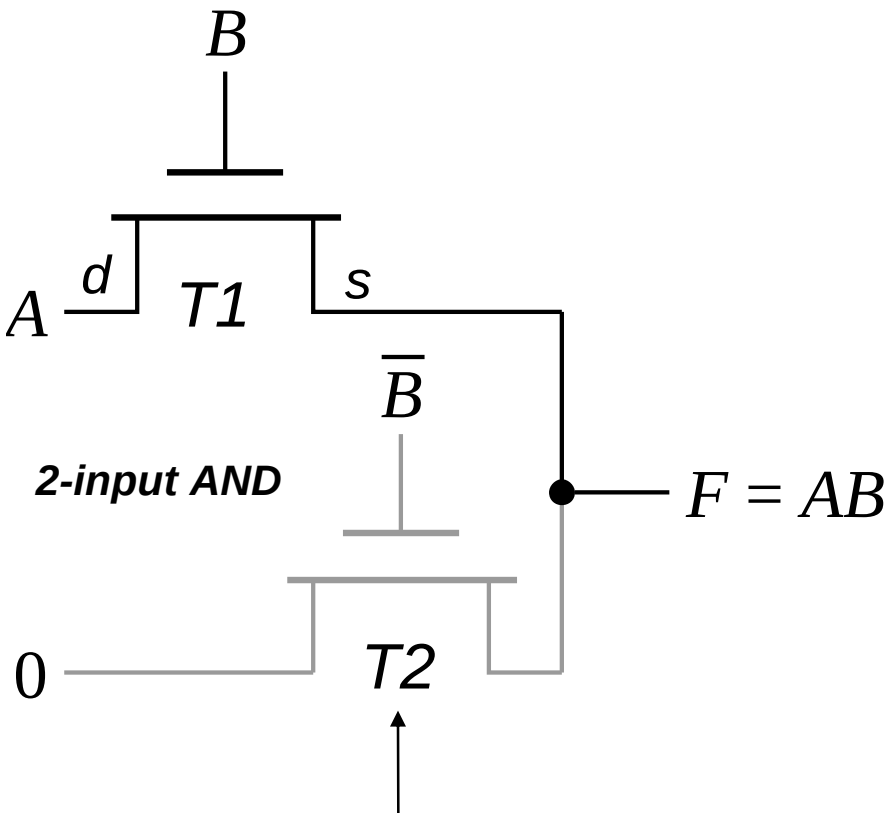
Pass-Transistor Logic



- **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*

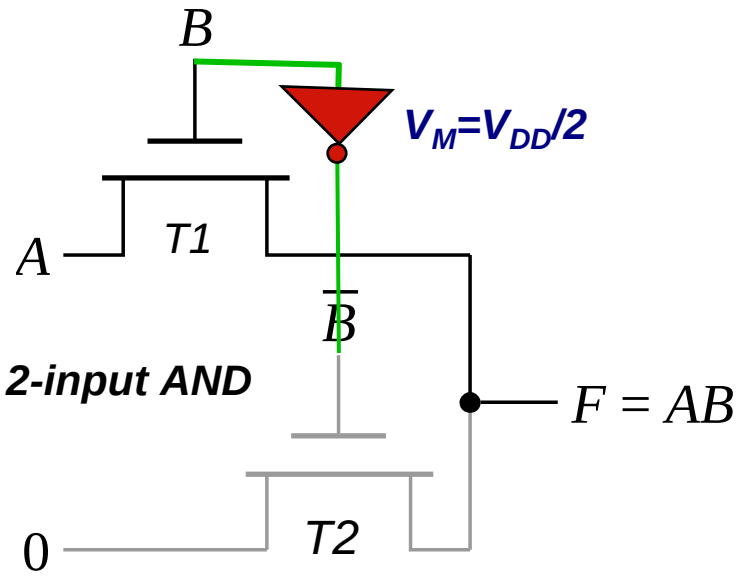
***Need fewer transistors**: 4 to implement the AND: lower cap.*

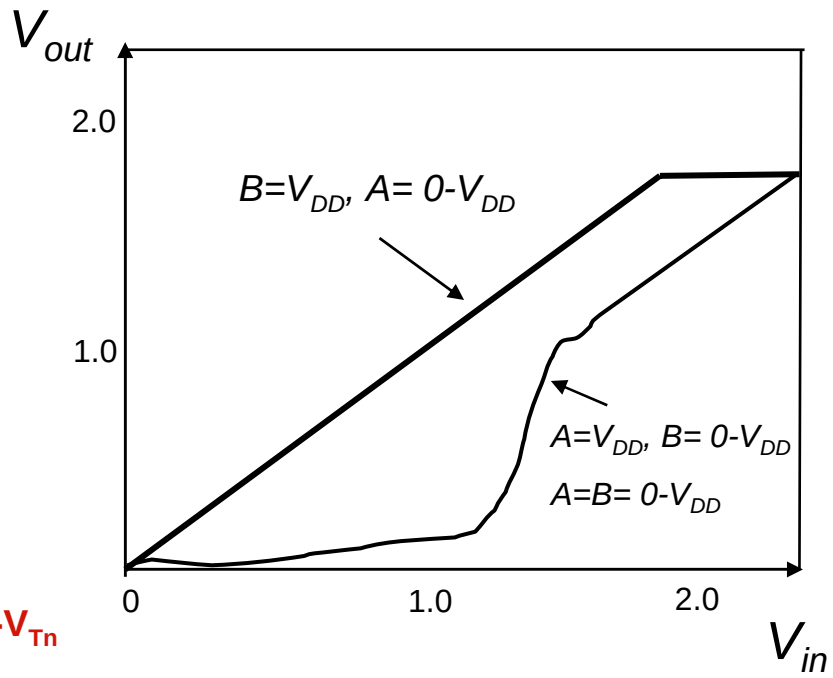
Need 6 to implement in static CMOS (4 for NAND and 2 for INV)

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





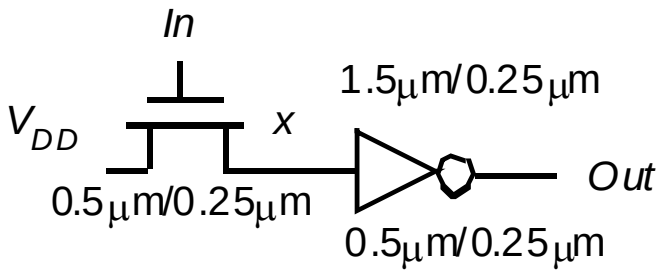
When $B=V_{DD}$, T1 is ON until the input reaches V_{DD}-V_{Tn}

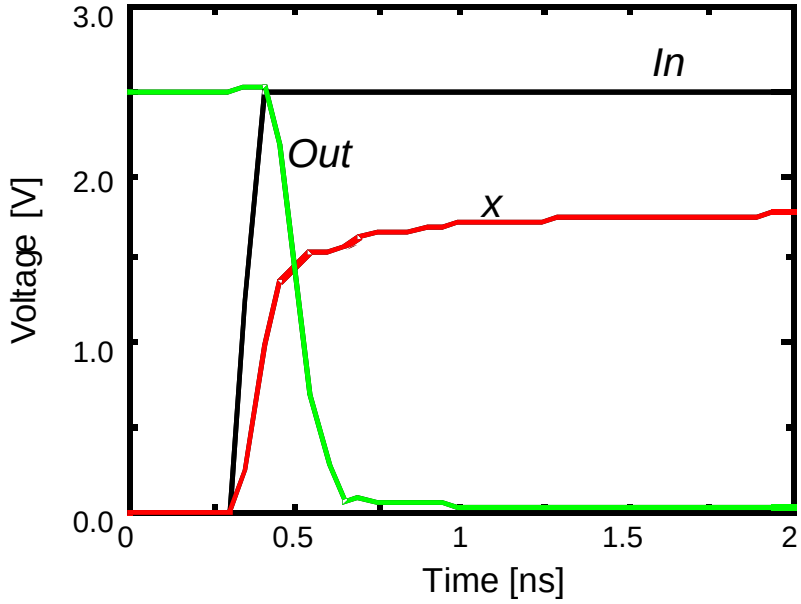
When $A=V_{DD}$, and B makes a transition from 0 to 1, T2 is turned on until $V_{DD}/2$ and Output =0. Once T2 is turned off, output follows the input B minus a threshold drop.

VTC of Pass Transistor Logic is data dependent

NMOS-Only Logic

Voltage Drop Issue:

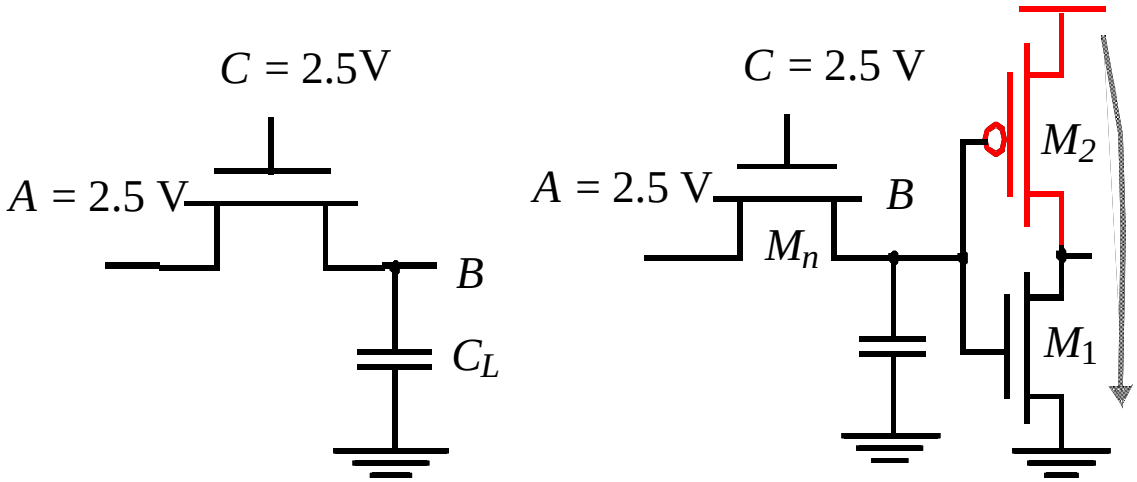




$$V_x = V_{dd} - V_{Tn}(V_x)$$

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!

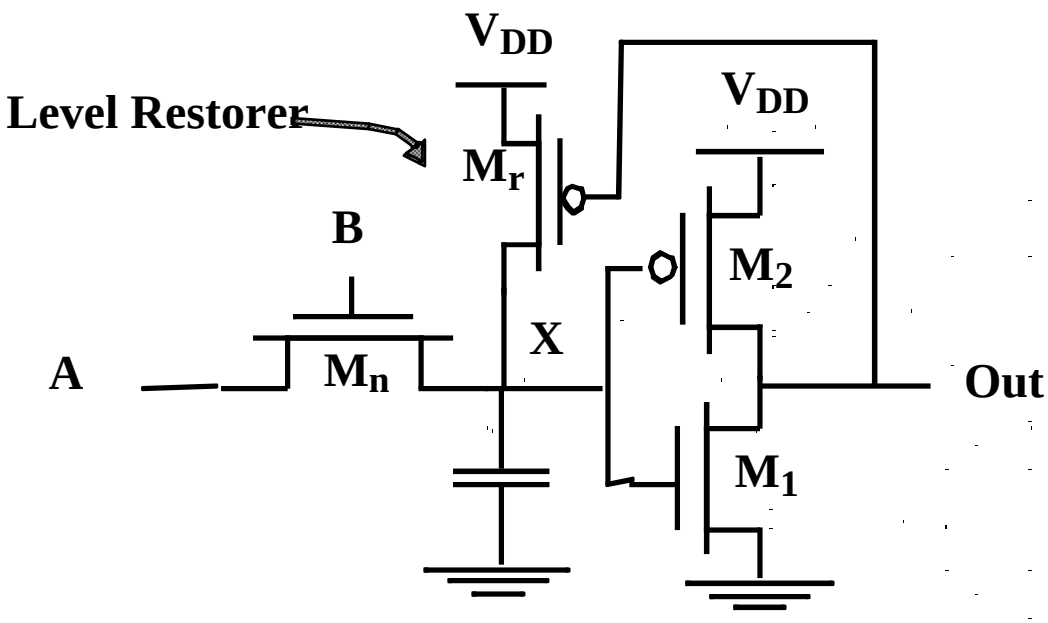
Difficult to switch off the PMOS!

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

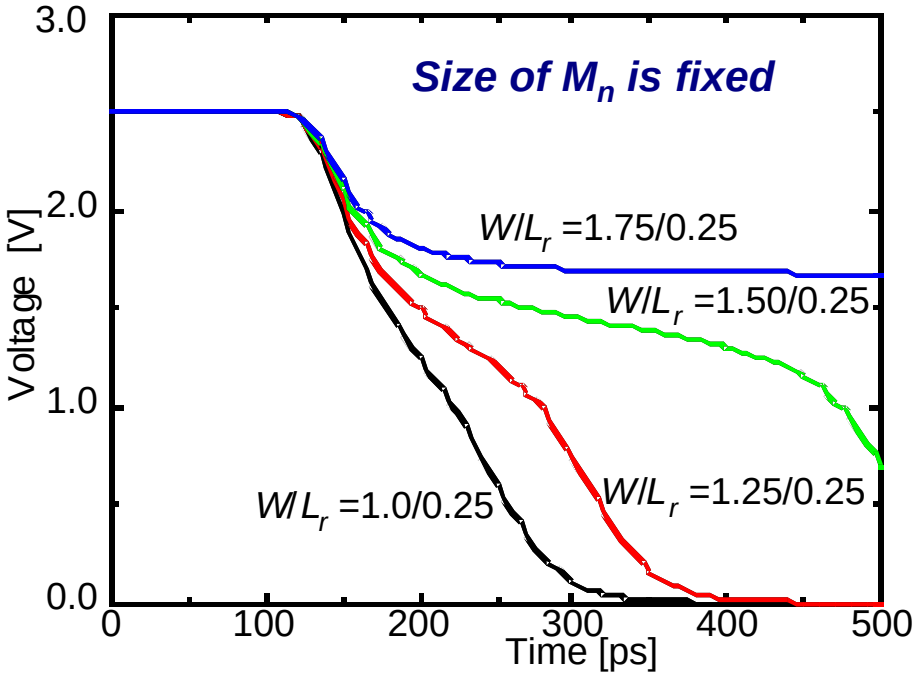
- **Advantage: Full Swing**

- **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*

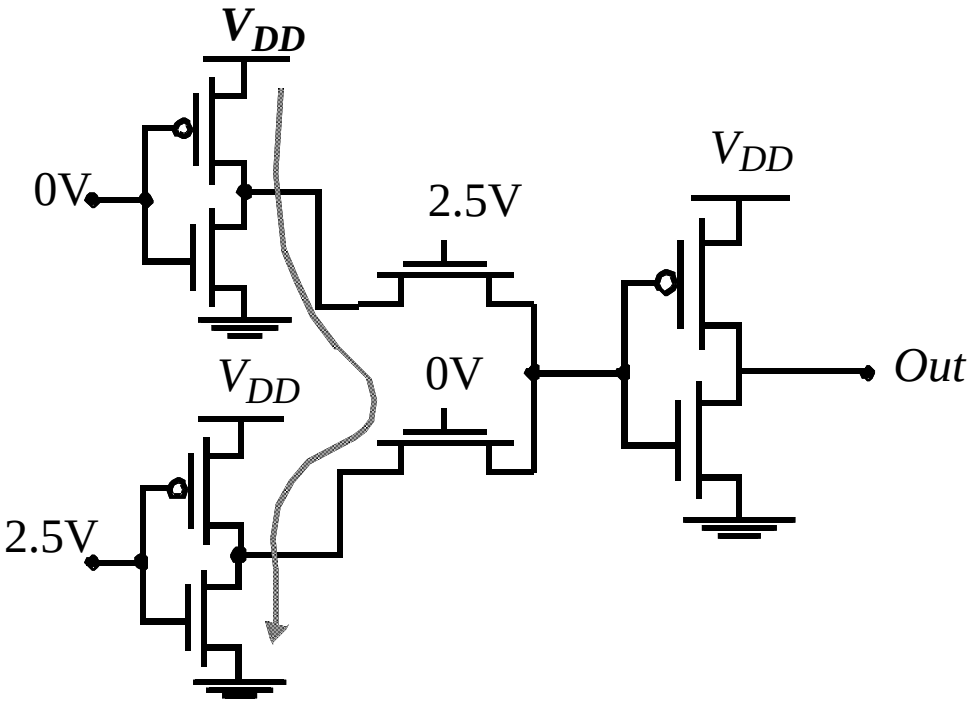


- **Upper limit on restorer size** when too large (R_r too small), V_x can't be brought below V_M
- Pass-transistor pull-down can have several transistors in stack

Transient Response: V_x vs. time

Solution 2: Single Transistor Pass Gate with $V_T=0$

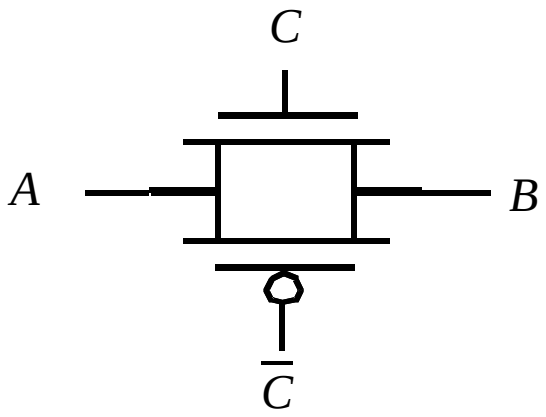
Only Pass Transistor Devices have $V_T=0$

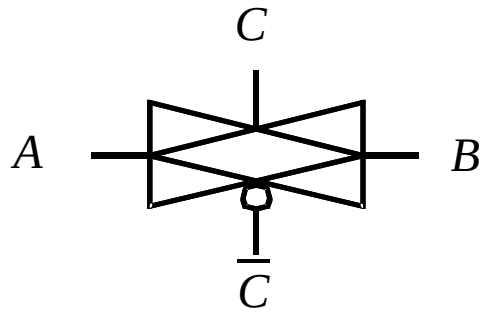


But even if $V_T=0$, there is still body effect…which prevents full swing!

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

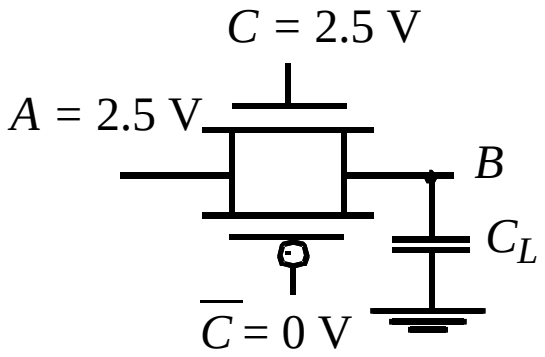
Solution 3: Transmission Gate





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

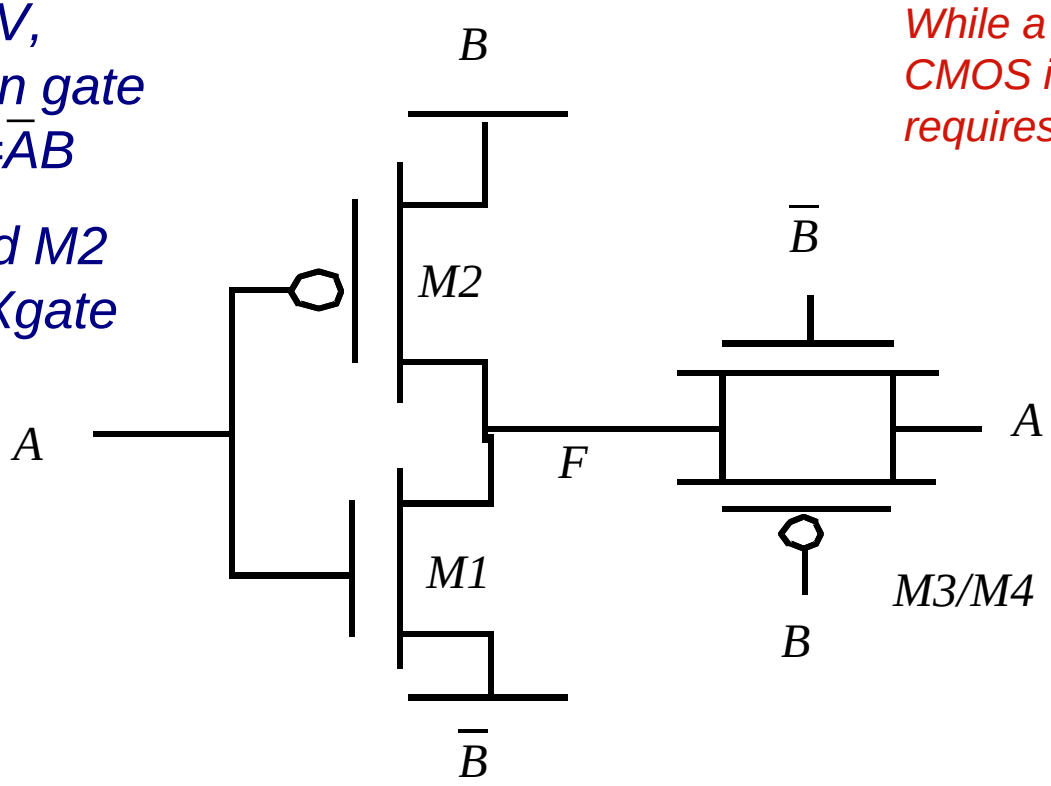
Transmission Gate XOR

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,* $F=A\bar{B}$

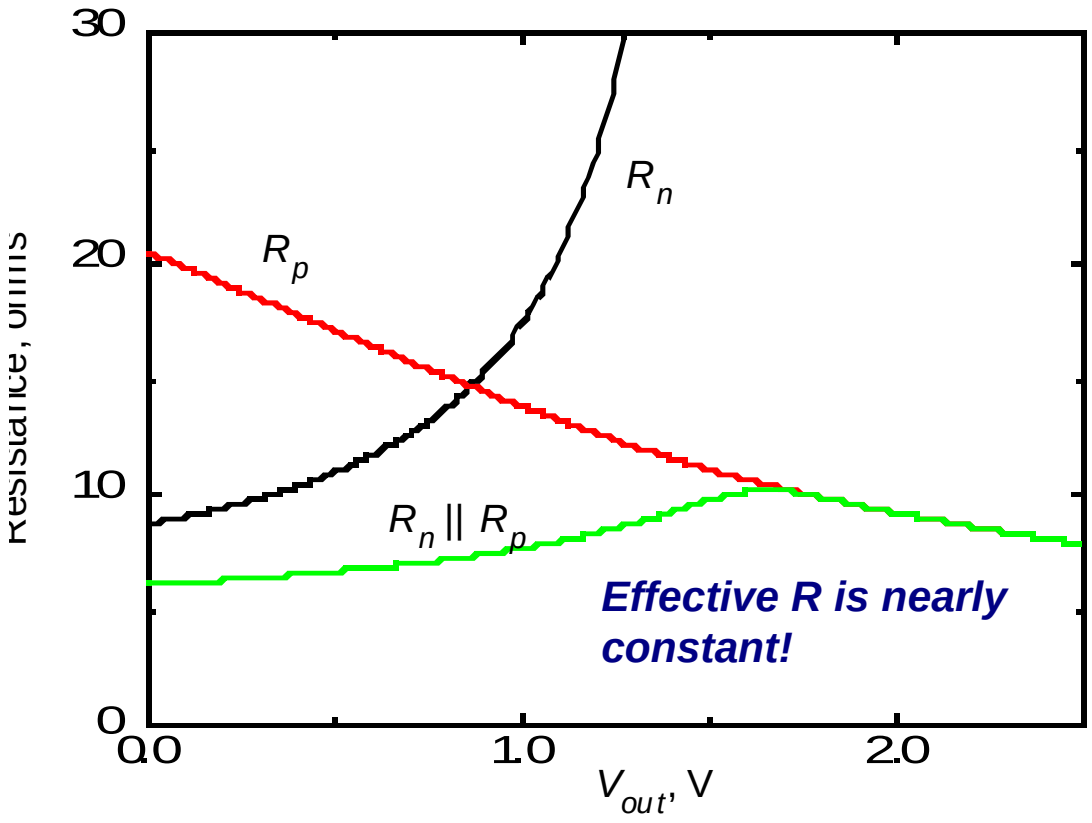
Implementation requires only 6 transistors!

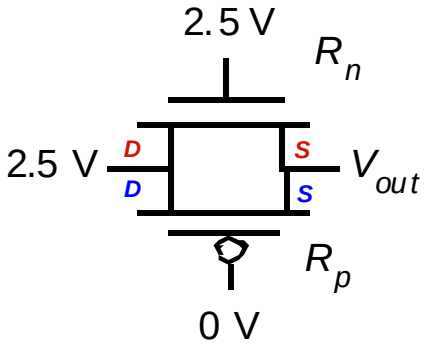
While a complementary CMOS implementation requires ???? transistors



Fast adder circuits and registers can also be implemented with Xgates

Resistance of Transmission Gate



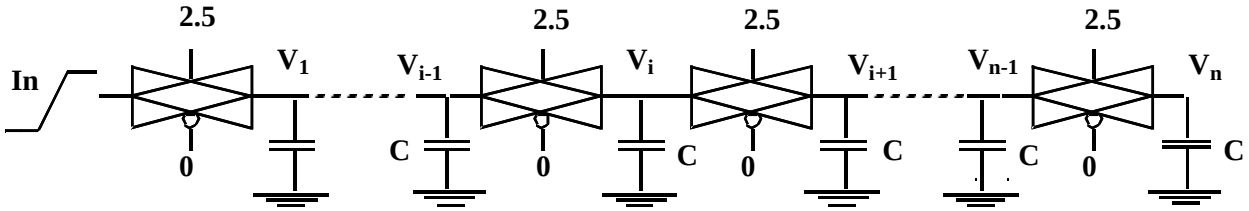


$R_n = (V_{DD}-V_{out})/I_{dn}$

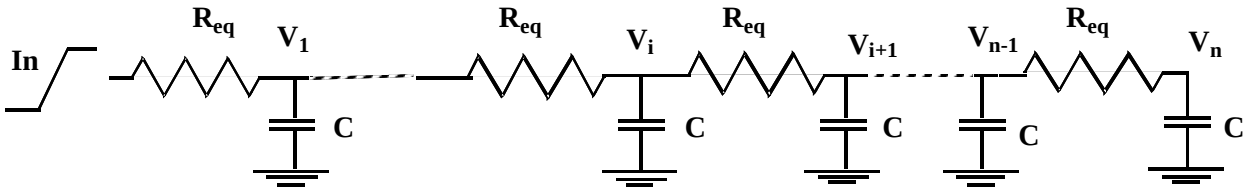
$R_p = (V_{out}-V_{DD})/I_{dp}$

Delay in Transmission Gate Networks

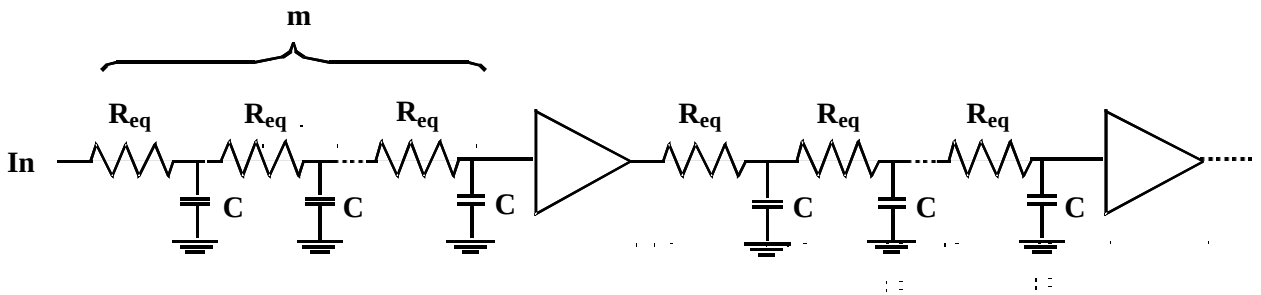
Delay of a chain of n Xgates (used in adders and deep MUXes) can be modeled using Elmore delay



(a) A chain of n Xgates



(b) Equivalent RC representation



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