EE5311- Digital IC Design

Module 4 - Combinational Circuit Design

Janakiraman V

Assistant Professor
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai

October 16, 2018

Learning Objectives

- Explain logical effort (LE) and electrical effort (EE)
- Derive the optimum number of buffers with their sizes to drive a load.
- Implement any arbitrary boolean function in Static CMOS logic
- Derive logical effort for any gate built in any style of logic
- Optimize the path delay of arbitrary gates driving a load capacitance
- Implement logic functions using ratio'd logic and dynamic logic
- Use the pass transistor to implement simple gates like MUX and XORs 8. Explain basic domino logic

Outline

- CMOS gates
- Gate sizing
- Capacitance estimation
- Delay estimation
- Logical effort
- Path delay optimizaion
- Buffer insertion
- Circuit Families
  - Static CMOS
  - Ratioed gates
  - Cascode Voltage Switch Logic (CVSL)
  - Dynamic clrcuits
  - Pass Transistor circuits

CMOS Gates

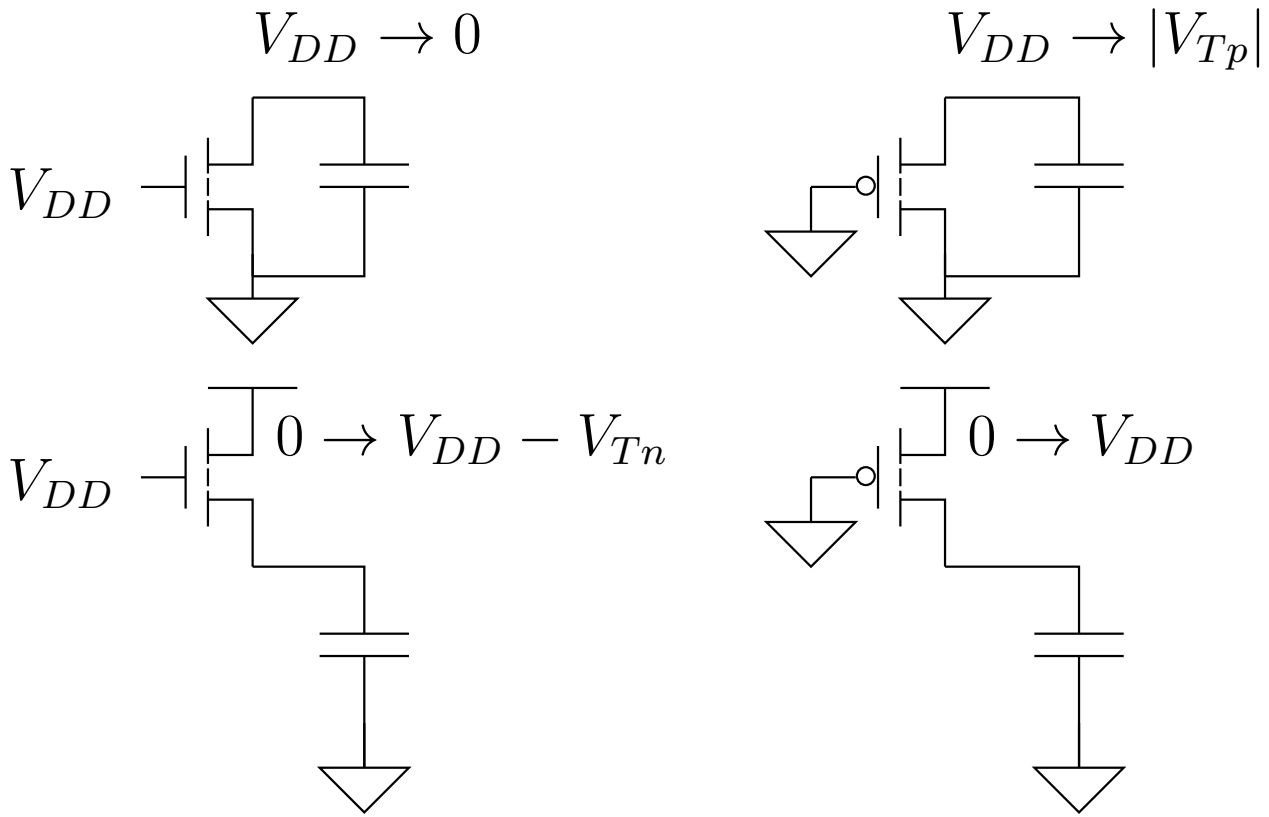


Figure: A CMOS Gate

- Pull down to *GND* with NMOS
- Pull up to V_{DD} with PMOS

CMOS Gates

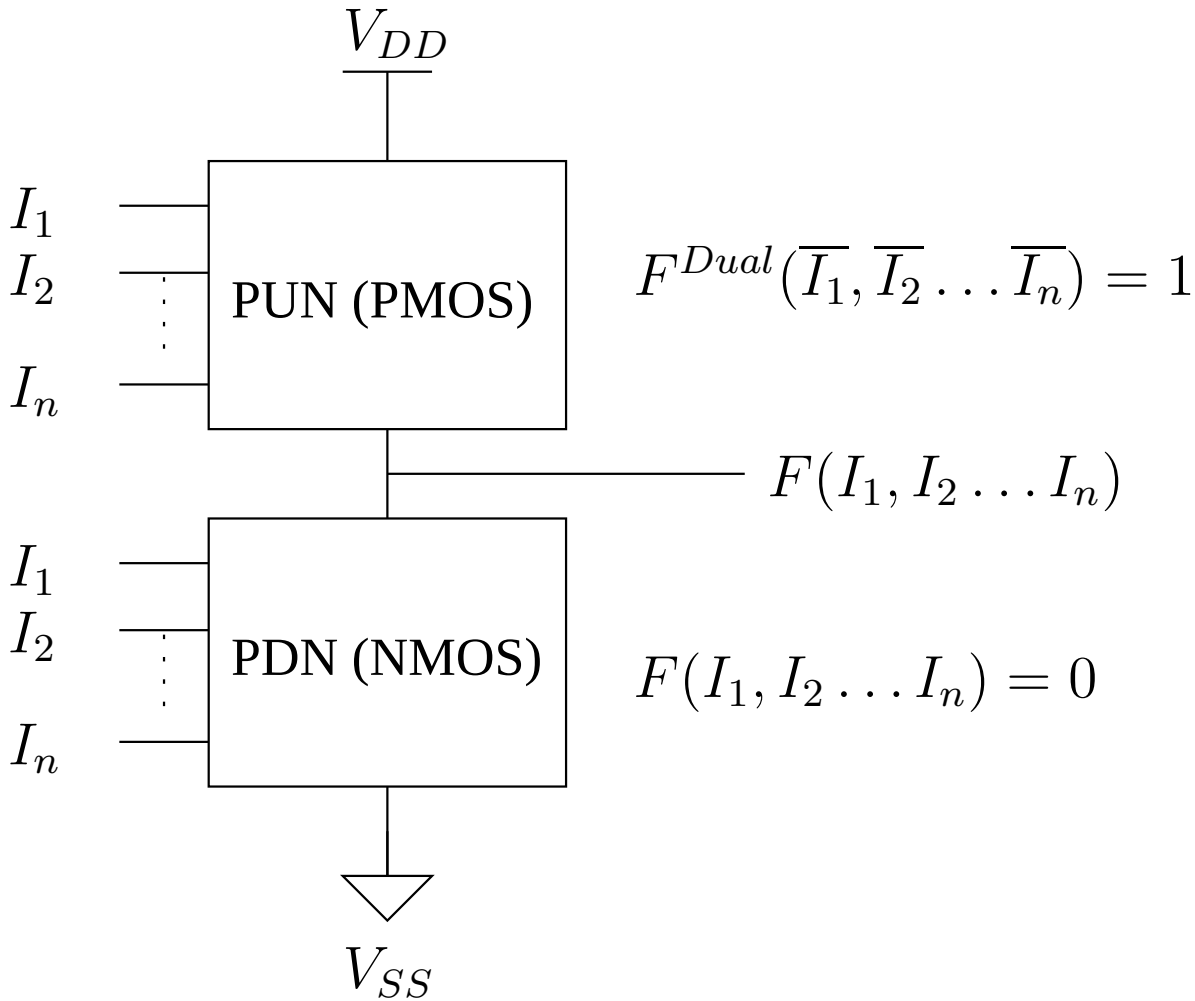


Figure: A CMOS Gate

CMOS Gates

- Pull up network (PUN) - PMOS only
- Pull down network (PDN) - NMOS only
- Series transistors provide an AND logic
- Parallel transistors provide an OR logic
- Use the SOP form to obtain the PDN
- Use De Morgan's laws to obtain the dual PUN
- CMOS gates are naturally inverting.
- N-input logic gate requires 2N transistors

Examples

- $Y = \overline{AB}$
- $Y = \overline{A + B}$
- $Y = \overline{A + BC}$
- $Y = \overline{D + A.(B + C)}$

Simple CMOS Gates - Gate Sizing

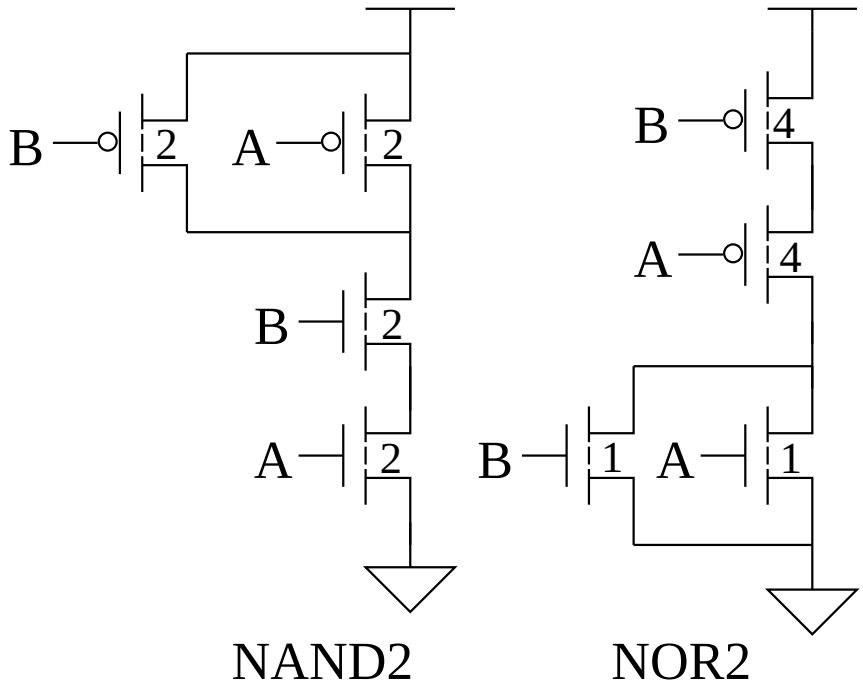


Figure: NOR2 and NAND2 Gate

- Transistor size is chosen so that rise and fall propagation delays are equal to that of a unit inverter

CMOS Gates - Capacitance

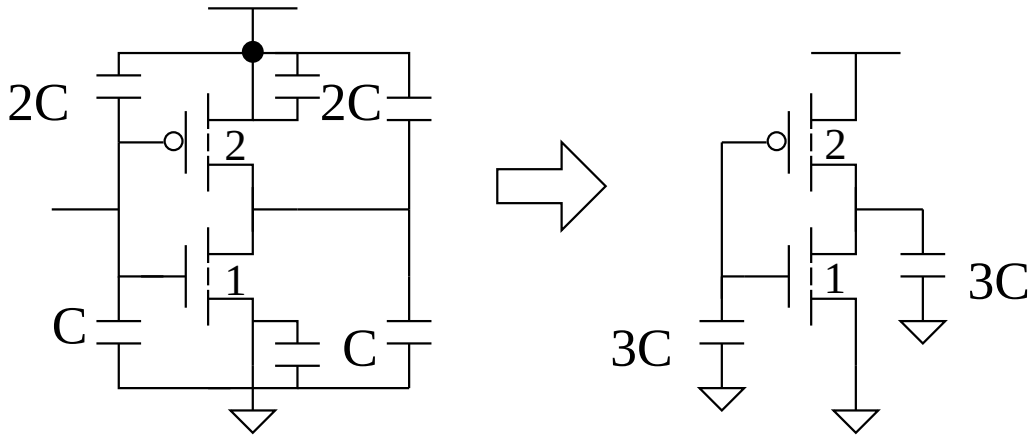
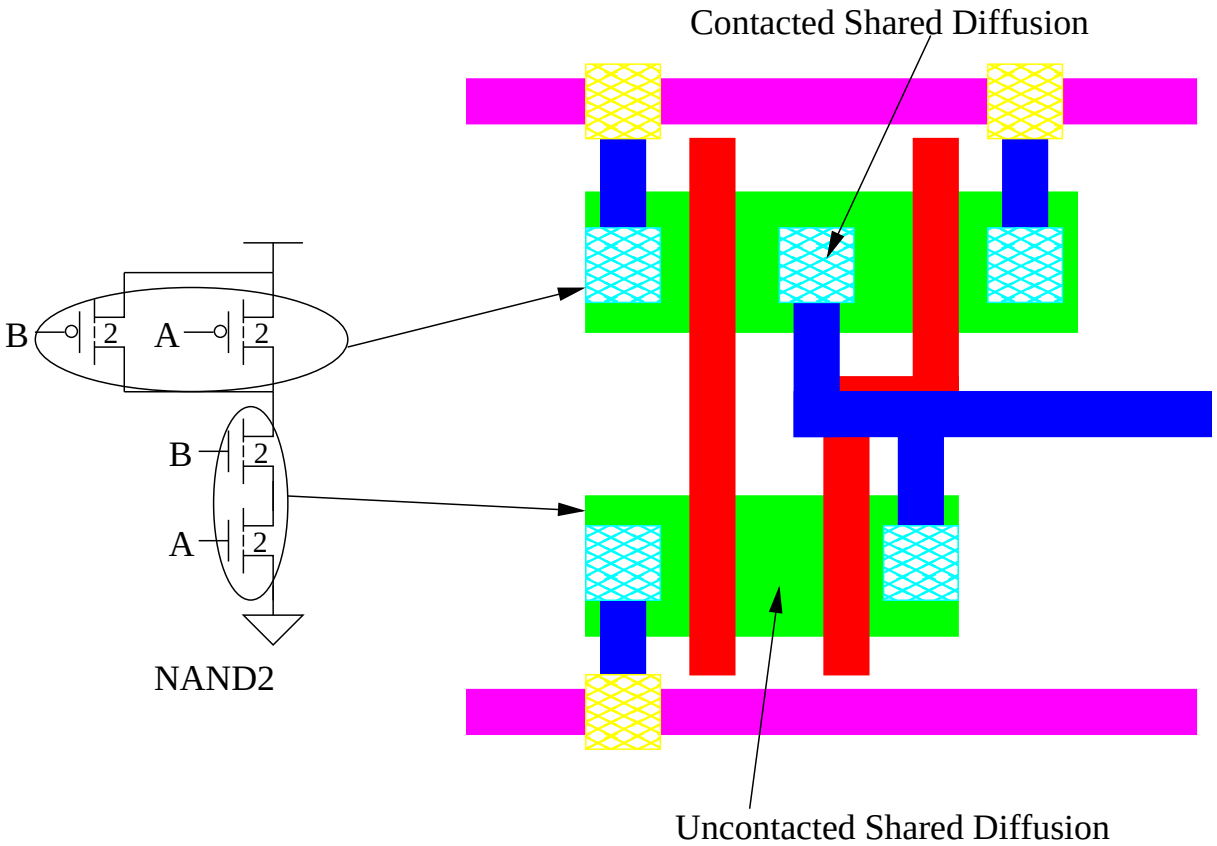


Figure: Inverter Capacitance Approximation

Useful Approximations

- Capacitance at all three terminals are equal (C)
- Capacitances are between the terminal and ground

Shared Diffusion - Capacitance



- Contacted shared diffusions need to be larger - Higher cap
- Uncontacted shared diffusions are smaller - Lower cap

NAND3 - Capacitance

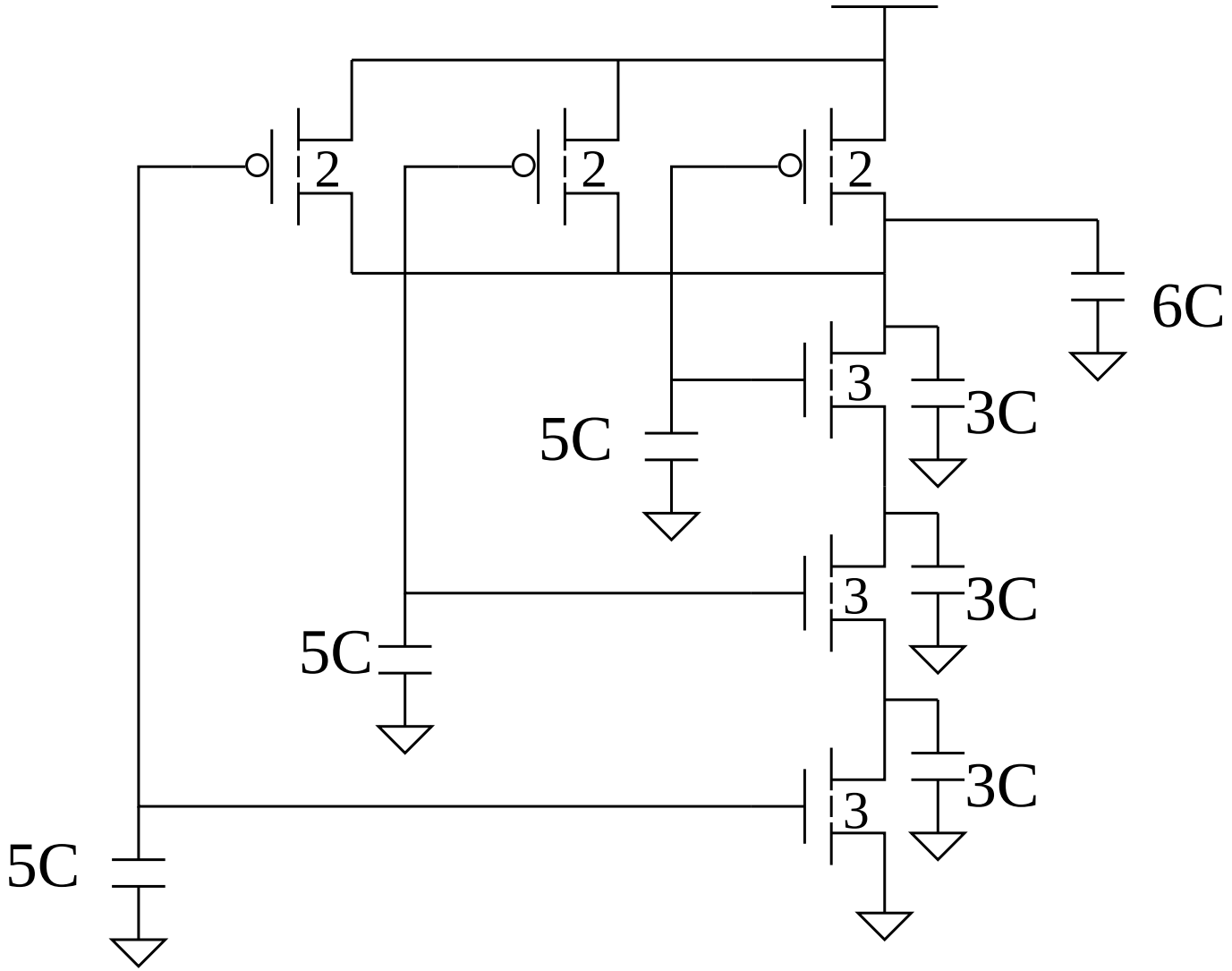


Figure: Uncontacted shared diffusion cap counted only once

Parasitic Delay

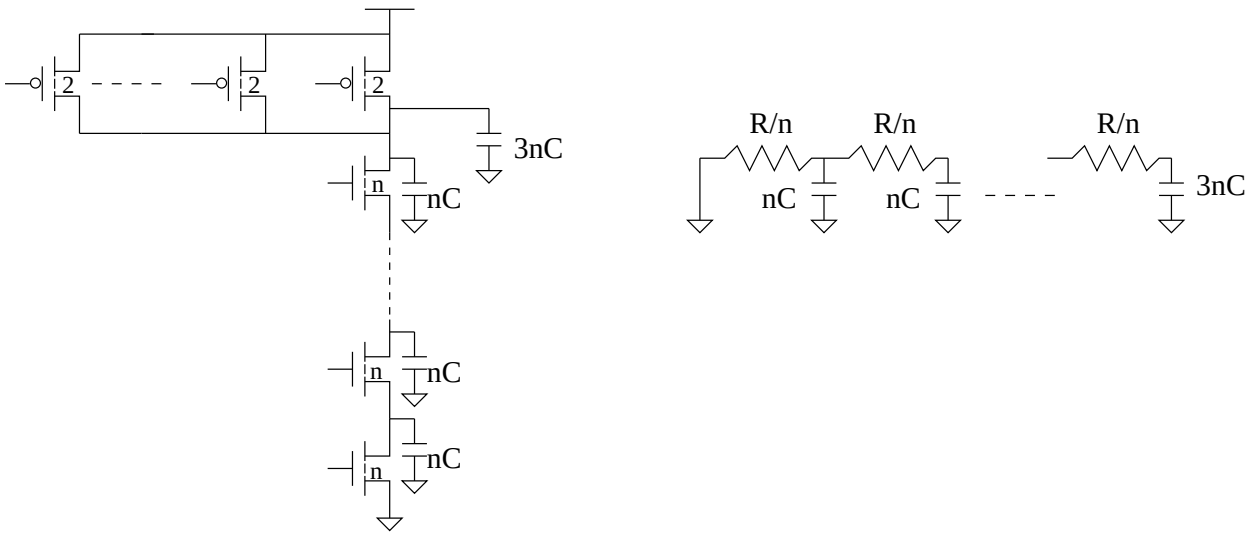


Figure: Parasitic delay for an n input NAND

$$t_{pd} = 3RC + \sum_{i=1}^{n-1} \frac{iR}{n} nC = (\frac{n^2}{2} + \frac{5n}{2})RC$$

- Delay grows quadratically as the number of inputs
- Keep the number of inputs down to 4

Parasitic Delay - Approximation

Gate Type	No. of Inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate Mux	2	4	6	8	2n

$$p_{inv} = 3RC$$

- Count diffusion capacitance only on output node
- Normalize to parasitic delay of an inverter
- Parasitic delay is independent of gate size to a first order

NAND2 - Delay

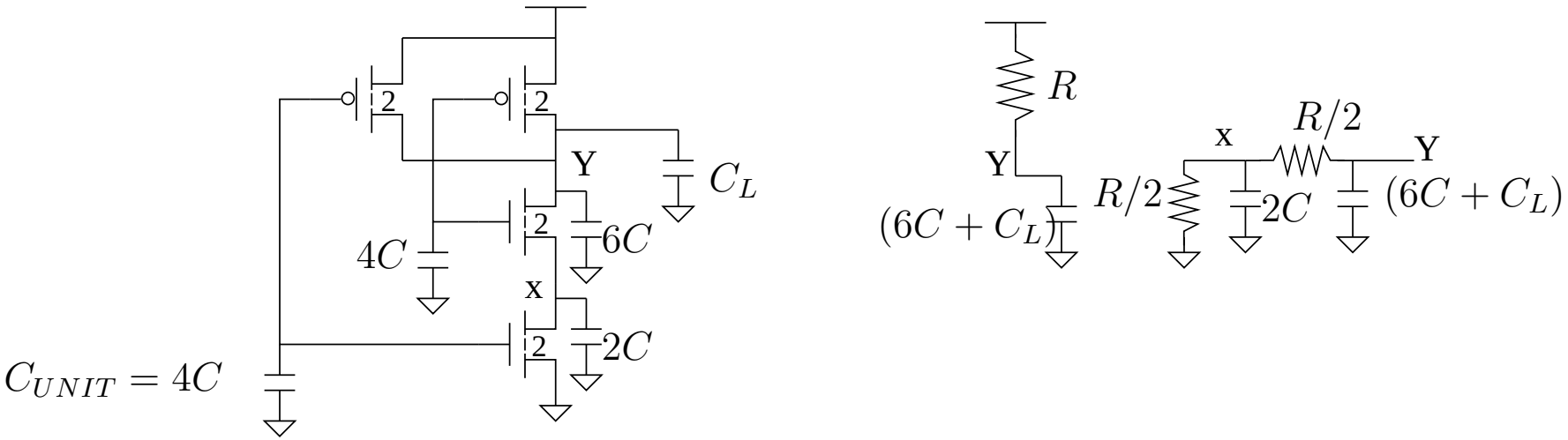


Figure: NAND2 gate driving an arbitrary load capacitance

- Rise Time Propagation Delay $= R(6C + C_L)$
- Fall Time Propagation Delay $= (R/2)(2C) + R(6C + C_L)$
- Rise Time Contamination Delay $= (R/2)(6C + C_L)$
- Fall Time Contamination Delay $= R(6C + C_L)$

NAND2 Upsized - Delay

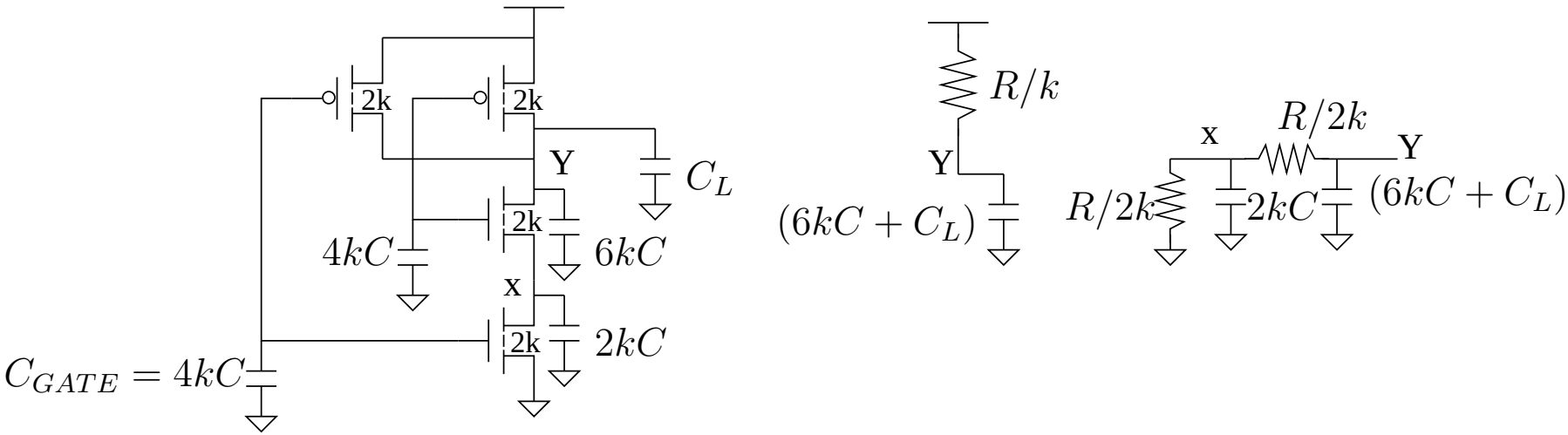


Figure: NAND2 Upsized by a factor of k

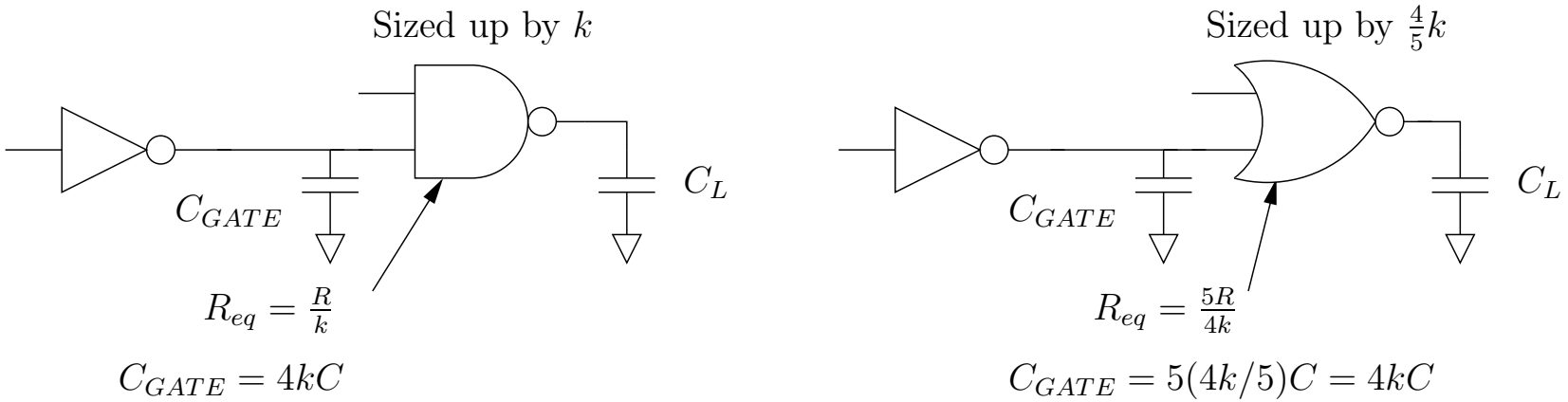
- Rise Time Propagation Delay $= (6RC + RC_L/k))$
- Fall Time Propagation Delay $= (7RC + (RC_L/k))$

NAND2 Delay - Observations

$$t_{pdr} = (6RC + RC_L/k)$$
$$t_{pdf} = (7RC + (RC_L/k))$$
$$k = C_{GATE}/4C$$
$$t_{pd} = t_{par} + 4RC\left(\frac{C_L}{C_{GATE}}\right)$$

- Parasitic delay $= t_{par} = 6RC$ or $7RC$ - Independent of k
- Effort delay $= 4RC\left(\frac{C_L}{C_{GATE}}\right)$ Depends on
  - $h = \left(\frac{C_L}{GATE}\right)$ - Ratio of load capacitance to input gate capacitance - Size dependent (Electrical Effort)
  - Gate capacitance of the unit gate (4C) - Decided purely by the logic complexity of the gate (Logical Effort)

Logical Effort Intuition



- A reference gate is placed in a circuit such its load capacitance is C_L and it offers a gate capacitance of C_{GATE}
- If the reference gate is swapped by another logically equivalent gate that is *sized to offer exactly the same* gate capacitance C_{GATE}
- The delay will be lesser for the gate with lesser logical effort

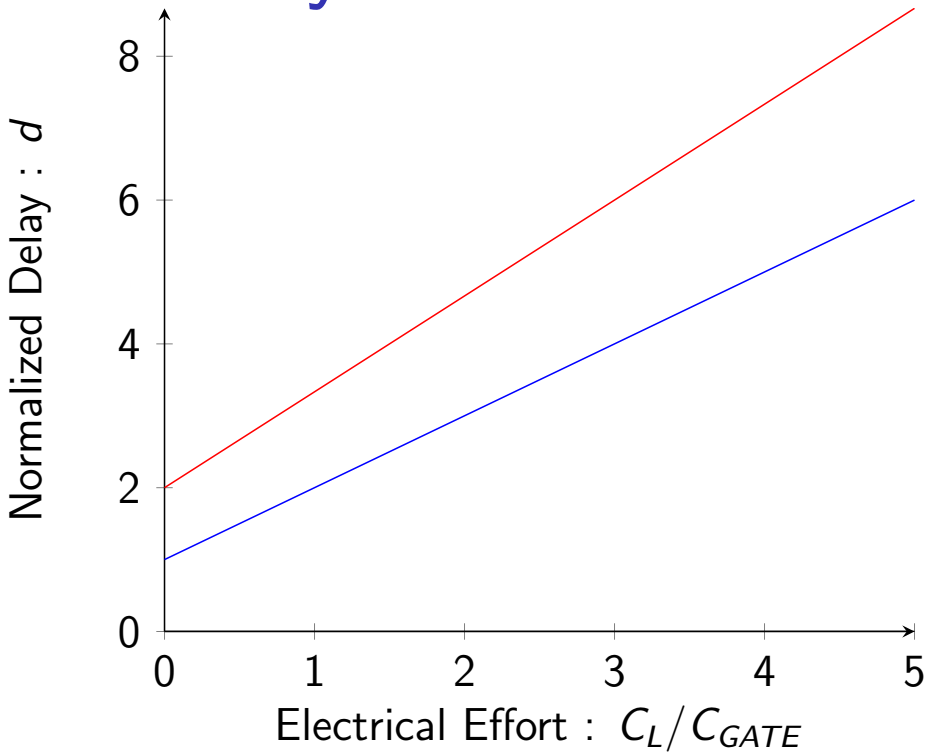
NAND2 - Normalized Delay

$$t_{pd} = 6RC + 4RC\left(\frac{C_L}{C_{GATE}}\right)$$

- Good to have a process independent representation
- Ideal unit inverter driving a FO1 load with no parasitic capacitance $= \tau = 3RC$

$$d = t_{pd}/\tau$$
$$d = ((4/3)h + 2)$$

Linear Delay Model



$$d = h + 1$$

$$d = (4/3)h + 2$$

$$d = gh + p$$

$$h = \frac{C_L}{C_{GATE}}$$

Logical Effort

Def: Ratio of input gate capacitance of the gate to that of an inverter that can deliver the same output current

Gate Type	No. of Inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
Tristate Mux	2	2	2	2	2
X(N)OR		4,4	6,12,6,	8, 16, 16, 8	

Gate Delay Estimation Examples

- Estimate the delay of an FO4 inverter
  - $d = 4 + 1 = 5$
  - $\tau = 0.2 T_{FO4}$
  - FO4 delay of a process $\approx 2\lambda/3 - 2\lambda/2$ ps
  - 180nm Process - FO4 delay $\approx 60 - 90ps$
- Estimate the delay of an Ring Osicllator
  - $d = 2N$
  - $T_{period} = 2 * 2N$
  - $f_{RO} = 1/4N\tau$

Gate Sizing

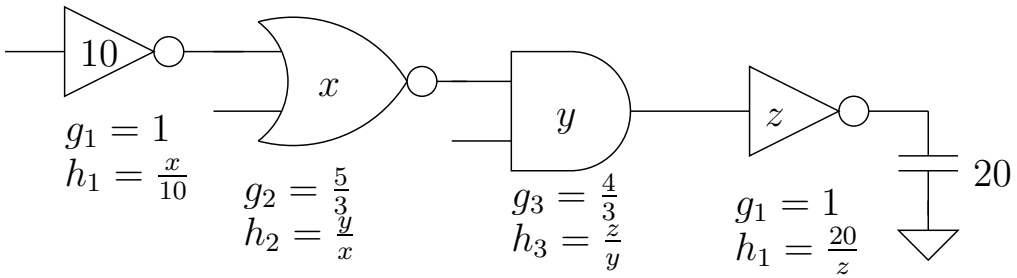


Figure: Gate Sizing

$$t_{pd} = \sum_{i=1}^{N} g_i h_i + p_i$$

$$\prod_{i=1}^{N} g_i = G$$

$$\prod_{i=1}^{N} h_i = H = \frac{C_{out}}{C_{in}}$$

Gate Sizing

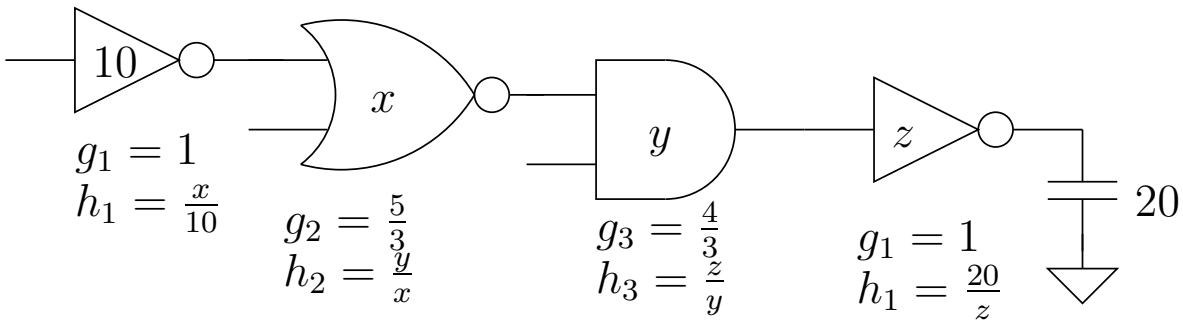


Figure: Gate Sizing

- $P = \sum_{i=1}^{N} p_i$ - Independent of gate size
- G - Path logical effort
- H - Path electrical effort - Constant
- $F = GH$ - Path effort - Constant
- Minimize $\sum_{i=1}^{N} g_i h_i$;
- Use $\frac{\sum_{i=1}^{N} g_i h_i}{N} \geq (\prod_{i=1}^{N} g_i h_i)^{\frac{1}{N}} = (F)^{\frac{1}{N}}$ (AM $\geq$ GM)
- Optimal delay $= NF^{1/N} + P$

Gate Sizing

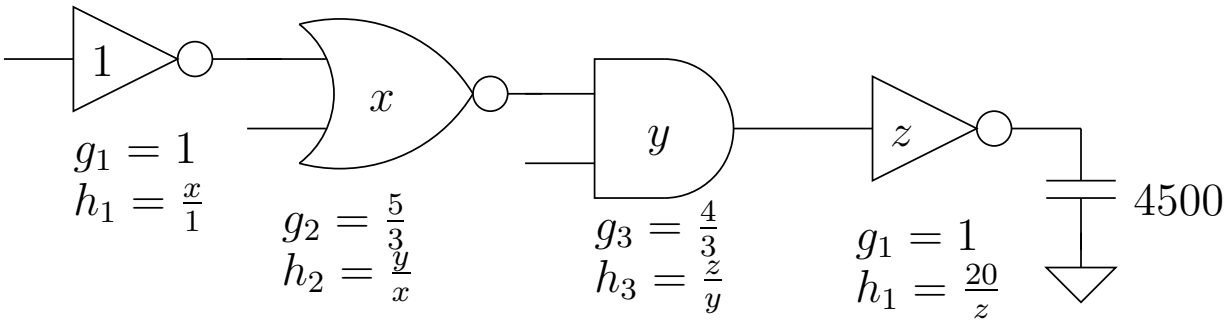


Figure: Gate Sizing - Large Caps

$$G = 4/3 \times 5/3$$

$$H = 4500/1$$

$$F = 10000$$

$$\tau_{opt} = NF^{1/N} + \sum_{i=1}^{4} p_i = 4 * 10 + \sum_{i=1}^{4} p_i$$

Can we do better?

Buffer Insertion

- Inserting inverters does not alter the logical effort.
- Useful to change the electrical effort
- Can potentially reduce the path delay
- Beware of the inversion

Buffer Insertion

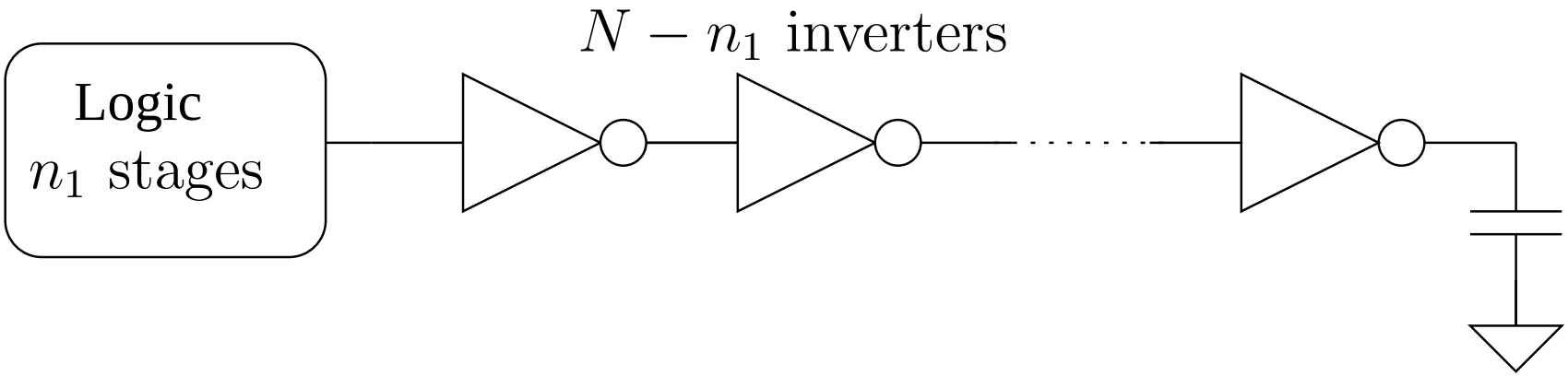


Figure: Buffer Insertion

$$D = NF^{1/N} + \sum_{i=1}^{n_1} p_i + (N - n_1)p_{inv}$$

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

$$p_{inv} + \rho(1 - ln(\rho)) = 0$$

Buffer Insertion

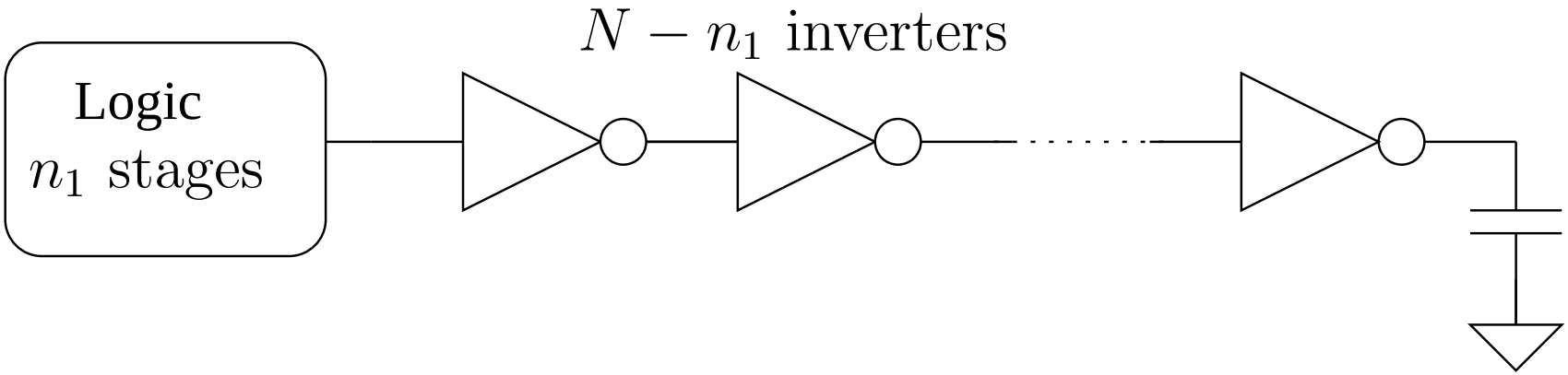
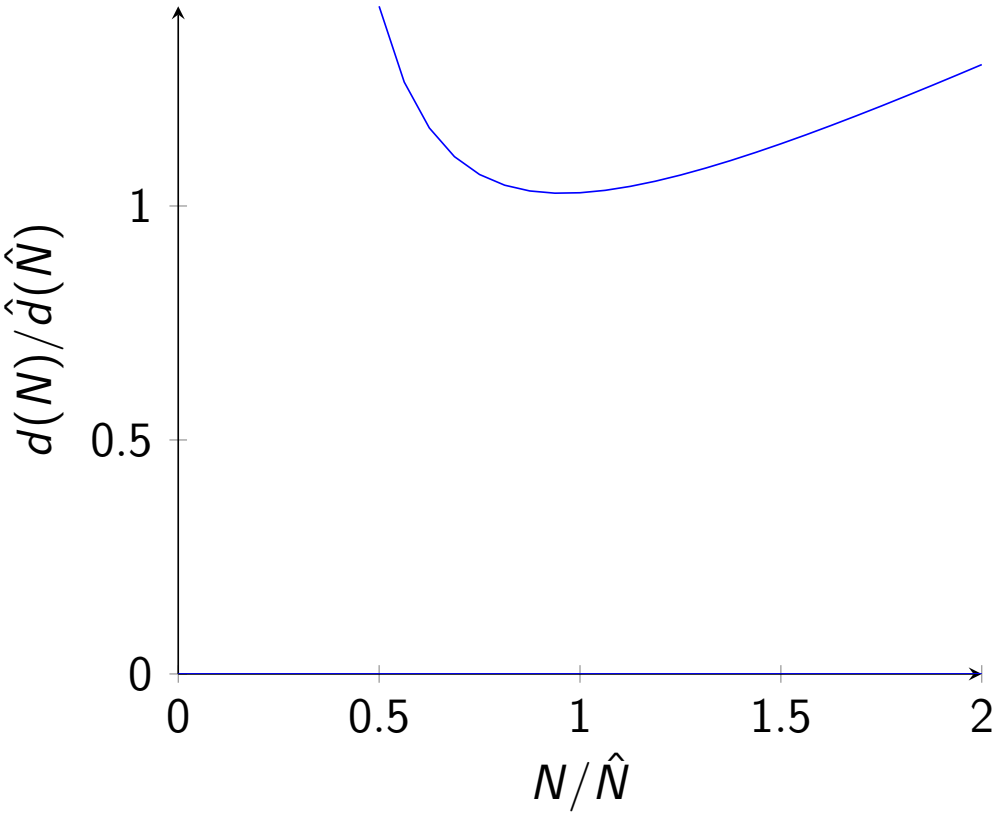


Figure: Buffer Insertion

$$\rho = F^{1/N}$$

$$p_{inv} = 1$$

$$p_{inv} + \rho(1 - ln(\rho)) = 0$$



- Solving we get $\rho_{opt} = 3.59$
- $\hat{N} = log_{\rho}F$
- Fan out of 4 is a practical and near optimum choice

Buffer Sizing

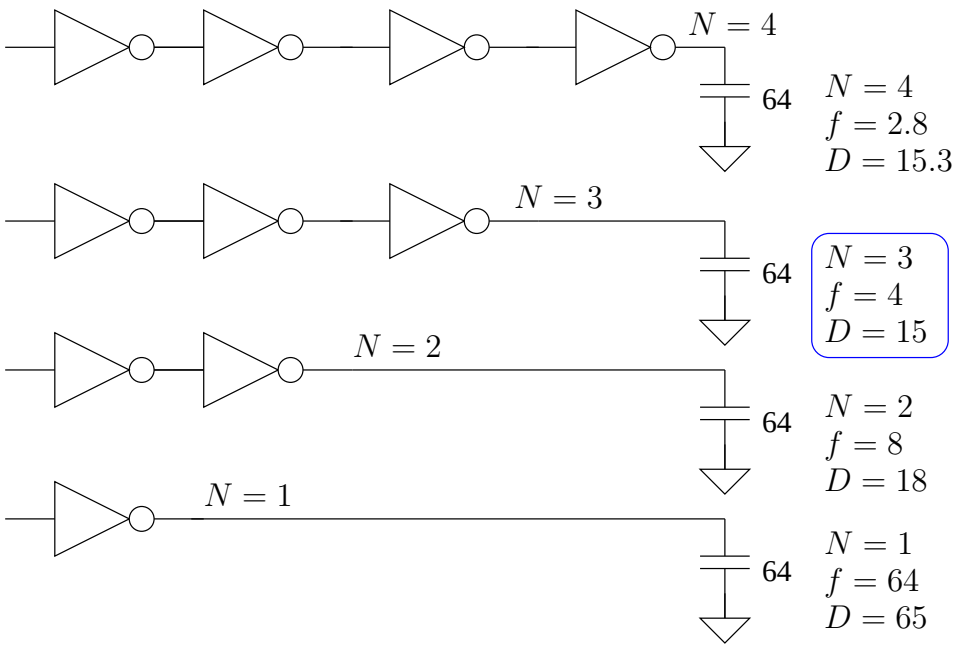


Figure: Buffer Sizing

- $D = NF^{1/N} + N$
- $F = 64$
- Optimal stage effort ≈ 4

Sizing - Summary

- Gate sizing (N is fixed)
  - Stage delay is made equal for all gates
  - $\tau_{opt} = NF^{1/N} + p$
- Buffer insertion (N is variable)
  - Reduce $NF^{1/N}$ by increasing N
  - Optimal stage delay ≈ 4
  - $\hat{N} = log_4(F)$
  - $\tau_{opt} = \hat{N}F^{1/\hat{N}} + p$
- How do you reduce $F = GH$?
  - $H = C_{out}/C_{in}$ - Fixed
  - $G = \prod_{i=1}^{N} g_i$ - Can we reduce g_i?

Circuit Families

- Static CMOS
- Ratioed gates
- Cascode Voltage Switch Logic (CVSL)
- Dynamic clrcuits
- Pass Transistor circuits

Static CMOS

- Good noise margins
- Fast
- Low power
- Insensitive to device variations
- Easy to design
- Supported by CAD tools
- Available in all standard cell libraries

Static CMOS

- Fundamentally inverting gates in nature
- Think NAND-NOR instead of AND-OR
- Use Demorgan's laws

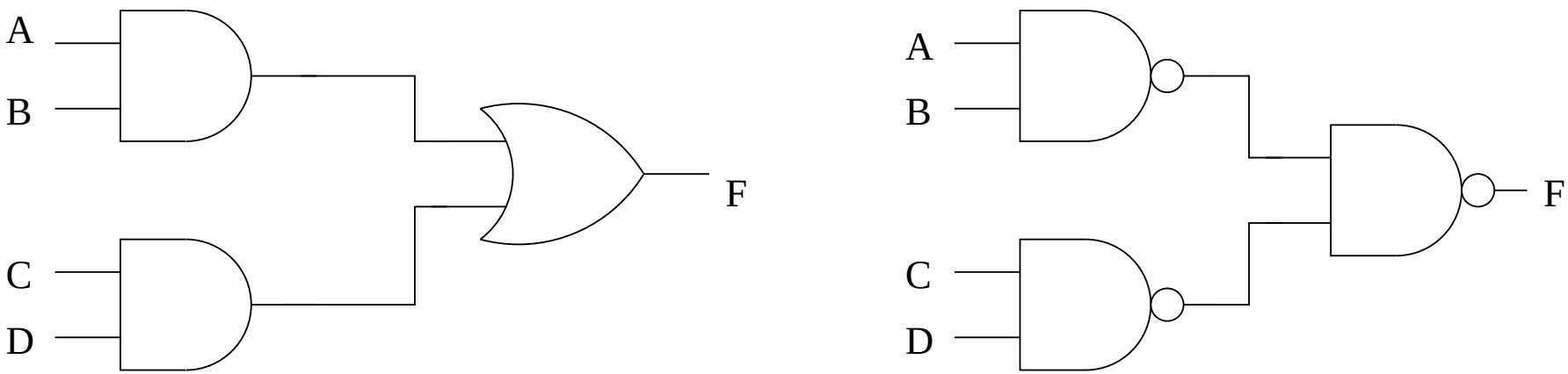


Figure: $F = AB + CD$

Complex Static CMOS

Sizing

- Worst case PU and PD drive strength of the gate must be similar to that of the unit inverter

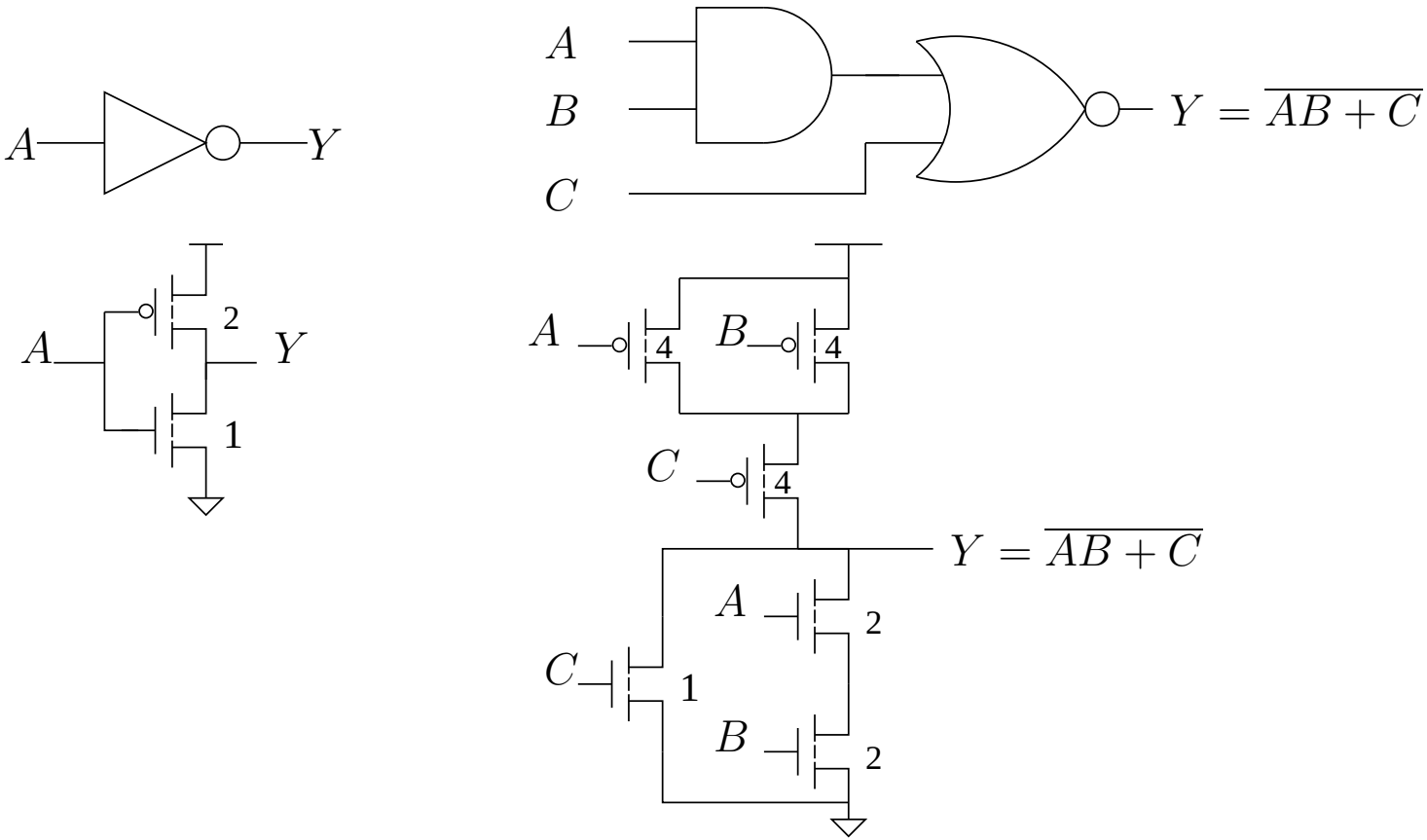
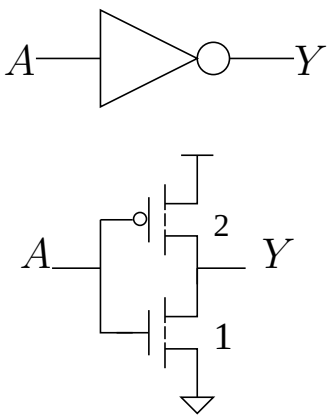
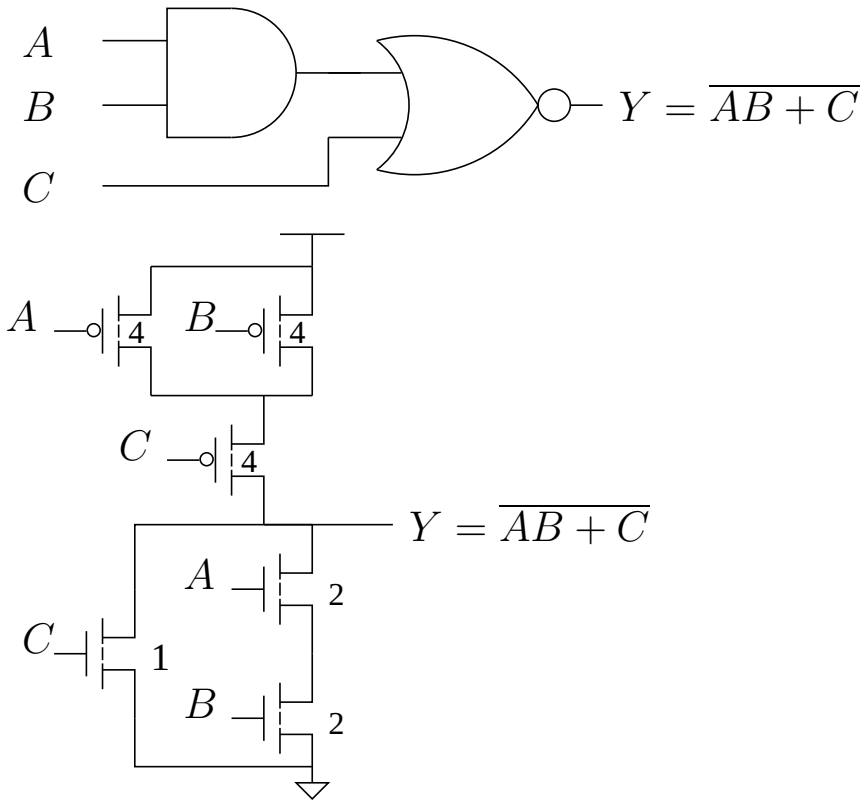


Figure: $F = \overline{AB + C}$

Complex Static CMOS - Logical Effort

- Logical effort of complex gates can be different for dfferent inputs
- $g_A = g_B = 6/3$
- $g_C = 5/3$
- $p = 7/3$





Complex Static CMOS

Draw the Static CMOS circuit that implements $Y = \overline{A(B + C) + DE}$ indicating the sizes and the logical effort each input

Special Functions

A	B	C		Y
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		1
1	1	1		1

$$Y = AB + BC + CA$$

Special Functions

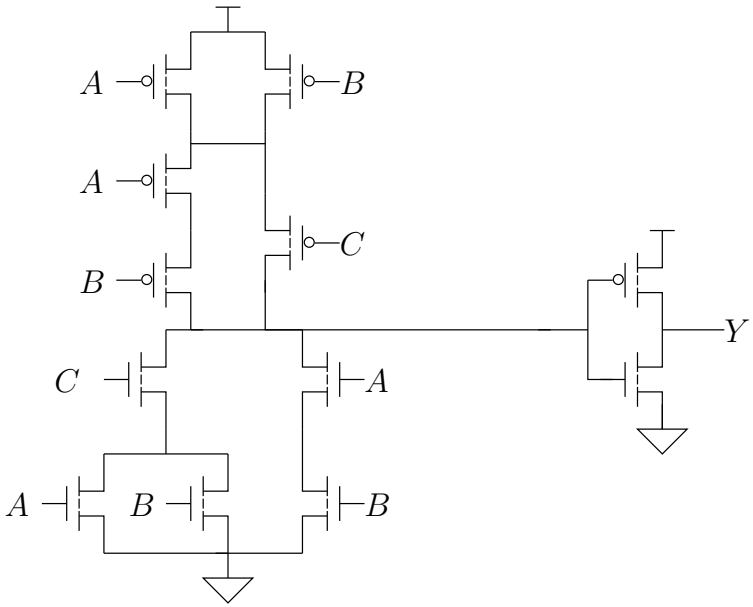


Figure: Full Adder Carry Circuit

$$Y = AB + BC + CA$$

Special Functions

Logic functions are written in SoP form (Sum of minterms)

$$Y = \sum m(3, 5, 6, 7)$$
$$\overline{Y} = \sum m(0, 1, 2, 4)$$

Inverting Property - **IF**

$$Y = F(A, B, C)$$
$$\overline{Y} = F(\overline{A}, \overline{B}, \overline{C})$$

The PMOS dual n/w *can be* identical to the NMOS network

Full Adder

Inverting property valid for both SUM and CARRY

A	B	C_i		S	C_o
0	0	0		0	0
0	0	1		1	0
0	1	0		1	0
0	1	1		0	1
1	0	0		1	0
1	0	1		0	1
1	1	0		0	1
1	1	1		1	1

Table: Full Adder: $S = \sum m(1,2,4,7)$, $C_o = \sum m(3,5,6,7)$

$$\overline{S} = \sum m(0,3,5,6)$$

$$\overline{C_o} = \sum m(0,1,2,4)$$

Input Ordering

- Delay when B rises last = $(R/2)2C + R(6C) = 7RC = 2.33\tau$
- Delay when A rises last = $R(6C) = 6RC = 2\tau$
- Outer input is close to the supply/ ground rail
- Inner input is close to the output
- Input that arrives last is connected to the inner input

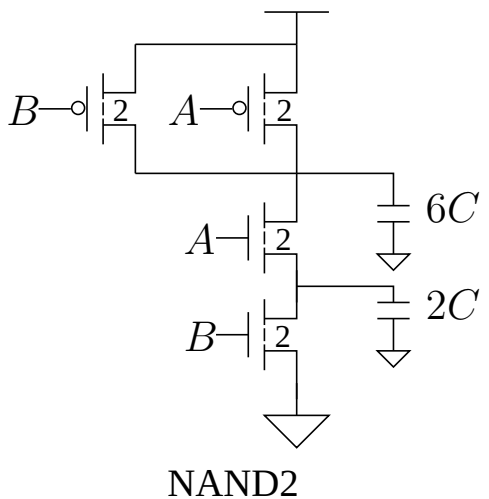


Figure: Outer input (B) and inner input (A)

Asymmetric Gates

- Intentionally skew the sizes to favour more important inputs
- Reset is not a high priority signal
- Pull down resistance $= R/4 + 3R/4 = R$
- Logical effort for input A is $g_A = 10/9 < 4/3$
- In the limit $g_A = 1$ - Large RESET transistor

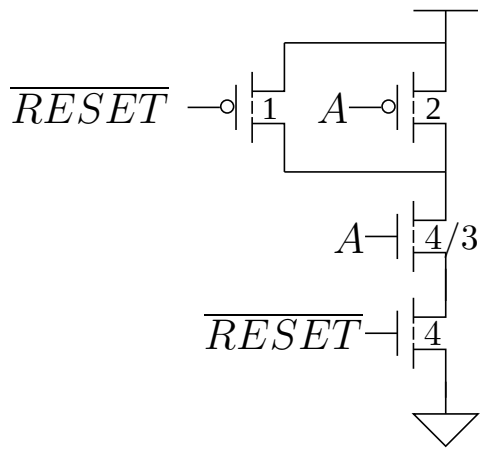


Figure: Input *A* preferred over *RESET*

Skewed Gates

- Intentionally skew the sizes to favour more important transition
- Logical Effort : Ratio of input capacitance of the gate to that of an unskewed inverter of same drive current
- Logical effort for rise $g_u = 2.5/3$
- Logical effort for fall $g_d = 2.5/1.5$
- For least average delay $W_P = \sqrt{2} \implies g_u = 1.15$, $g_d = 0.81$

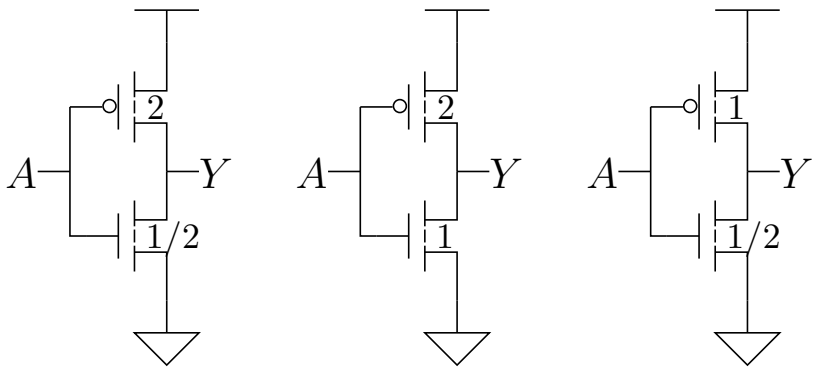
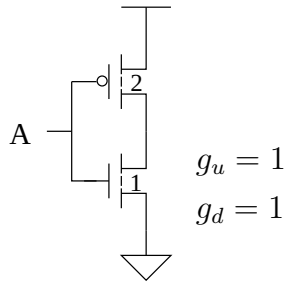
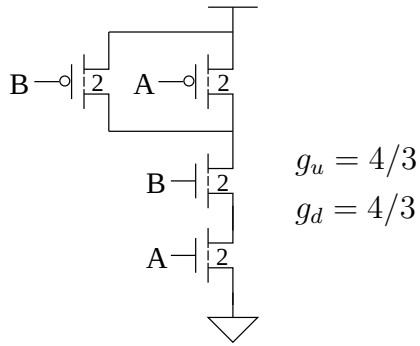
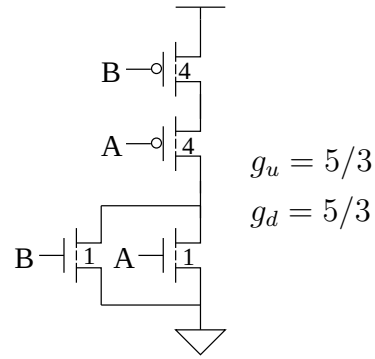
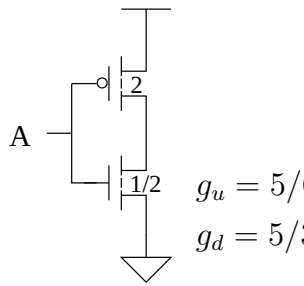
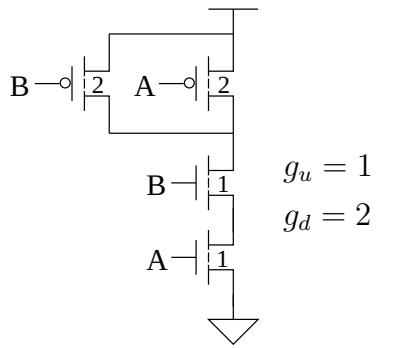
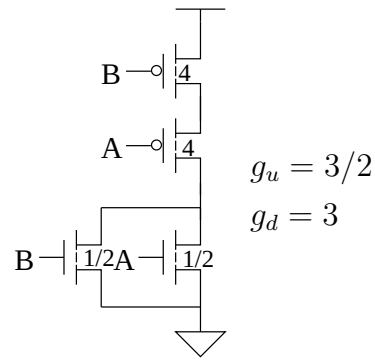
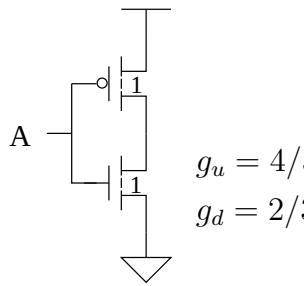
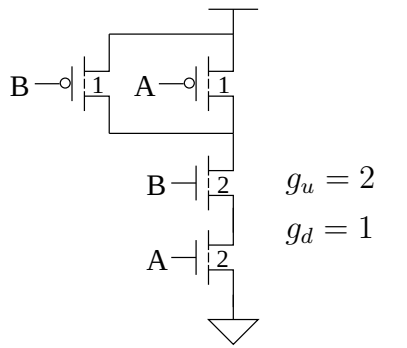
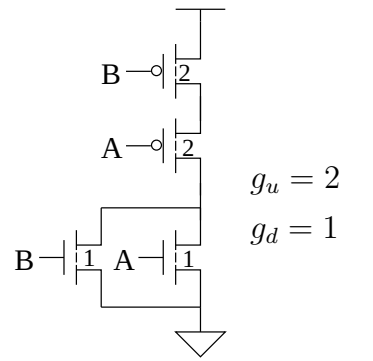


Figure: Logical effort for a HI Skew inverter

Skewed Gates

	INV	NAND2	NOR2
UNSKEWED	 $g_u = 1$ $g_d = 1$	 $g_u = 4/3$ $g_d = 4/3$	 $g_u = 5/3$ $g_d = 5/3$
HI−SKEW	 $g_u = 5/6$ $g_d = 5/3$	 $g_u = 1$ $g_d = 2$	 $g_u = 3/2$ $g_d = 3$
LOW−SKEW	 $g_u = 4/3$ $g_d = 2/3$	 $g_u = 2$ $g_d = 1$	 $g_u = 2$ $g_d = 1$

Ratioed Circuits

- Static CMOS requires $2N$ transistors, where N is the number of inputs
- N transistors are PMOS - Each is at least twice the width of the NMOS
- Ratioed logic mitiagates this problem

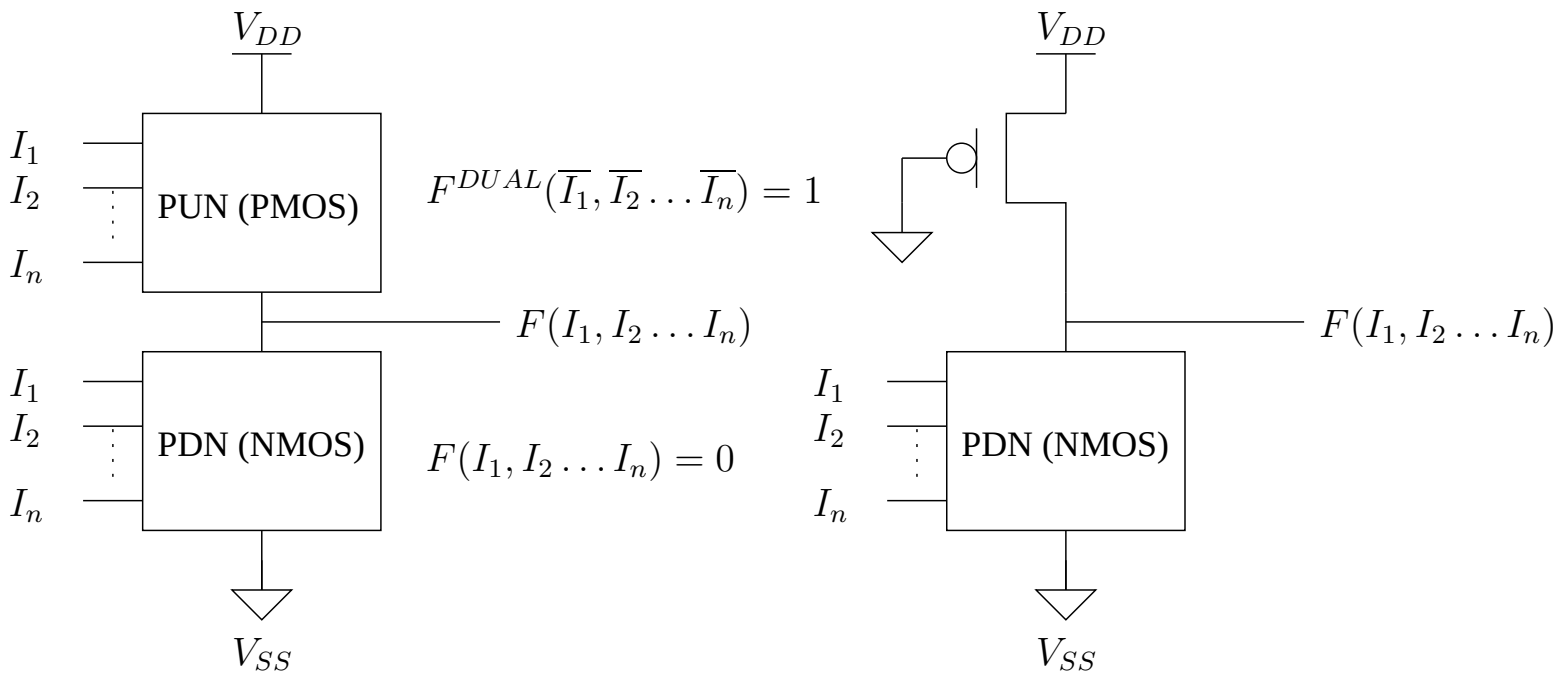
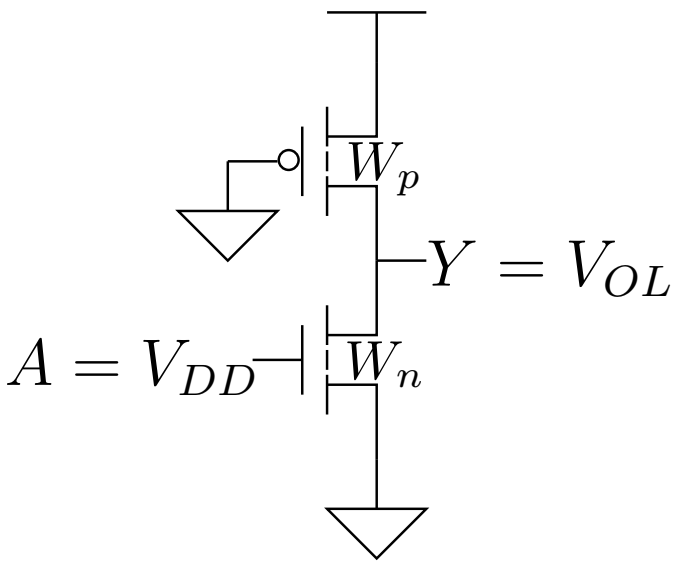


Figure: Pseudo NMOS Circuit

Pseudo NMOS Gate - V_{OL}

- Pull up PMOS is always ON
- $A = 0 \implies V_{OH} = V_{DD}$
- $A = V_{DD} \implies Y = V_{OL}$? - NMOS and PMOS fight
- NMOS in Linear Region
- PMOS in Velocity Saturation



Pseudo NMOS Gate - V_{OL}

$$I_{DSn} = \frac{k'_n W_n}{L} V_{OL}(V_{DD} - V_{Tn} - \frac{V_{OL}}{2})$$

$$I_{DSp} = \frac{k'_p W_p}{L} V_{DSATp}(-V_{DD} - V_{Tp} - \frac{V_{DSATp}}{2})$$

$$I_{DSn} = -I_{DSp}$$

$$V_{OL} = \frac{k'_p W_p}{k'_n W_n} \left(\frac{V_{DD} + V_{Tp} + \frac{V_{DSATp}}{2}}{V_{DD} - V_{Tn}} \right) V_{DSATp}$$

$$V_{OL} \approx \frac{\mu_p W_p}{\mu_n W_n} |V_{DSATp}|$$

- W_p/W_n should be as small as possible
- Rise delay severely affected if W_p is very small

Ratioed Circuits - Pseudo NMOS Gates

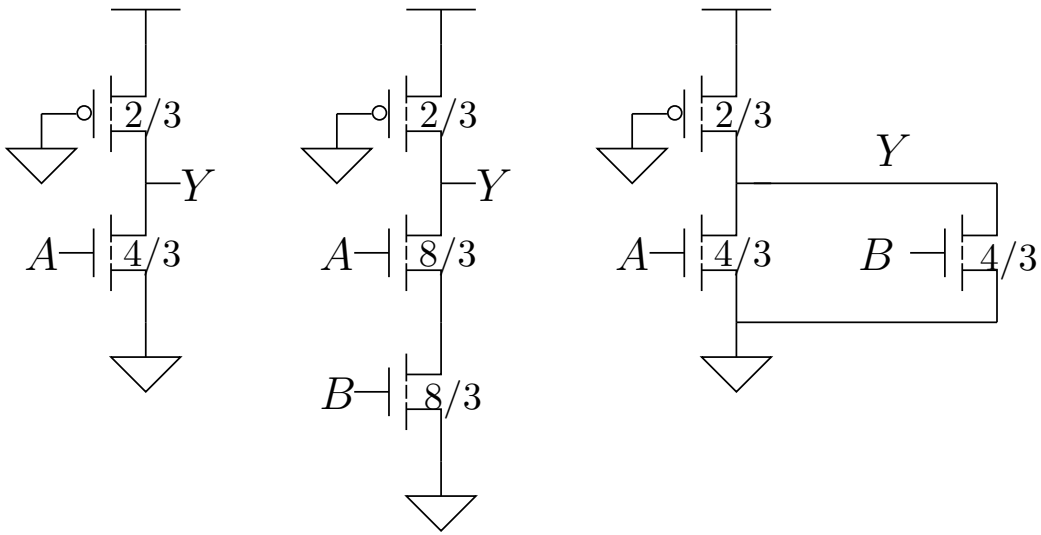


Figure: Pseudo NMOS Gates INV, NAND2 and NOR2

Pseudo NMOS Gates- Logical Effort

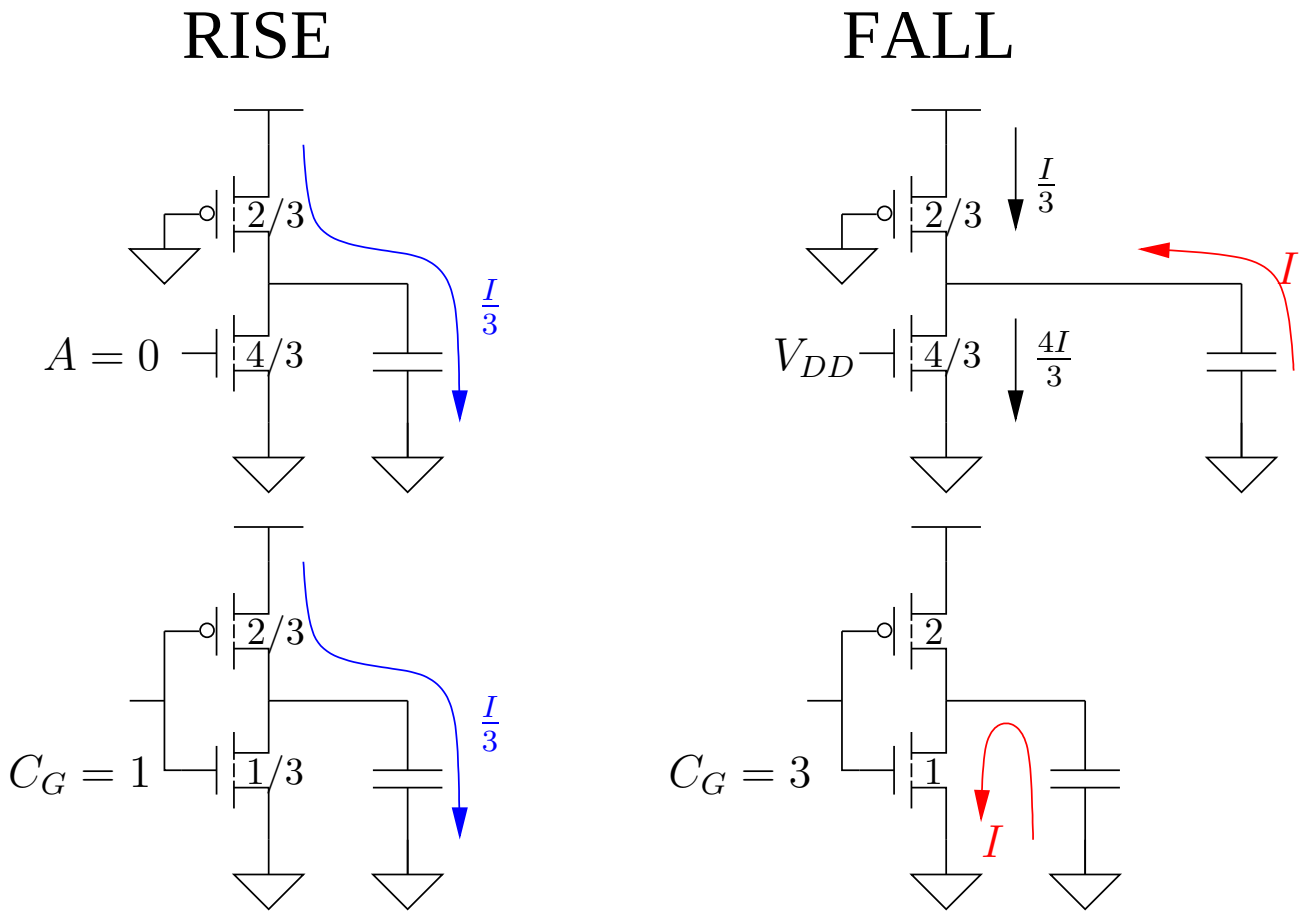


Figure: Logical Effort : Reference symmetric inverter

Pseudo NMOS - Logical Effort

Gate	g_u	g_d	g_{avg}	p_u	p_d	p_{avg}
INV	4/3	4/9	8/9	18/9	6/9	12/9
NAND2	8/3	8/9	16/9	30/9	10/9	20/9
NOR2	4/3	4/9	8/9	30/9	10/9	20/9

- Average delay for NAND2 is larger than Static CMOS
- Works well for fast-wide NORs - Logical effort is independent of number of inputs
- Beware of SF process corner - Slow NMOS and Fast PMOS

Dynamic Circuits

Ratioed circuits suffer from

- Weak pull up transistor
- Contention on a falling transition
- Significant static power
- Non-zero V_{OL}
- Dynamic Circuits circumvent these problems by using a clocked Pull up

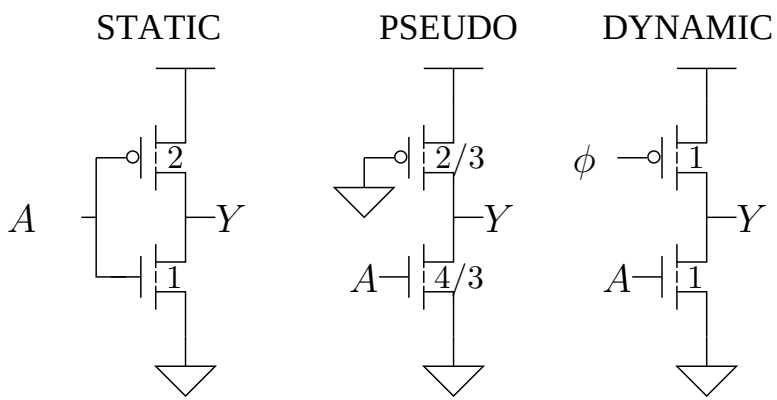


Figure: Dynamic Circuits

Dynamic Circuits - Evaluate and Precharge

Dynamic circuits function in two phases

- Pre-charge
- Evaluate
- In precharge phase, A should NOT be HIGH
- Add a footer device as well

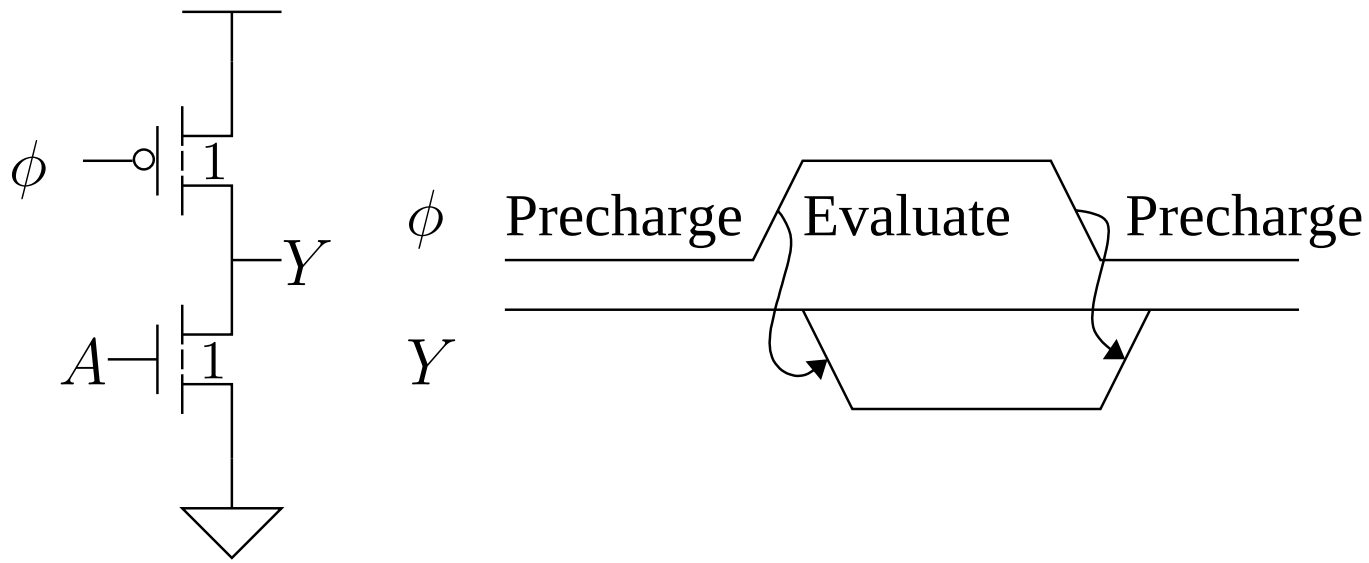
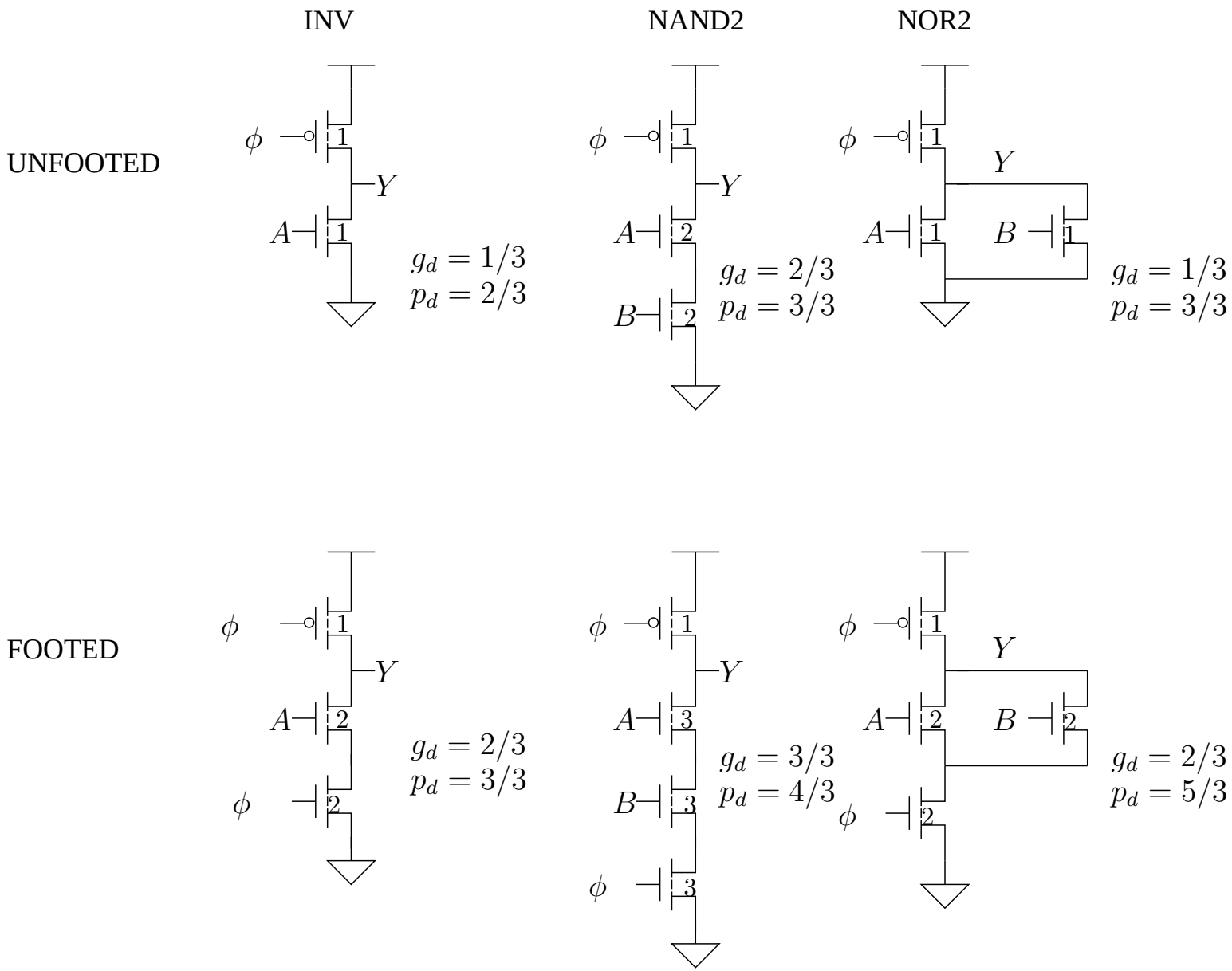


Figure: Two phases

Dynamic Circuits - Footed and Unfooted



Dynamic Circuits - Monotonicity

During Evaluate

- Input should be monotonically rising
- Input CANNOT start HIGH and fall LOW

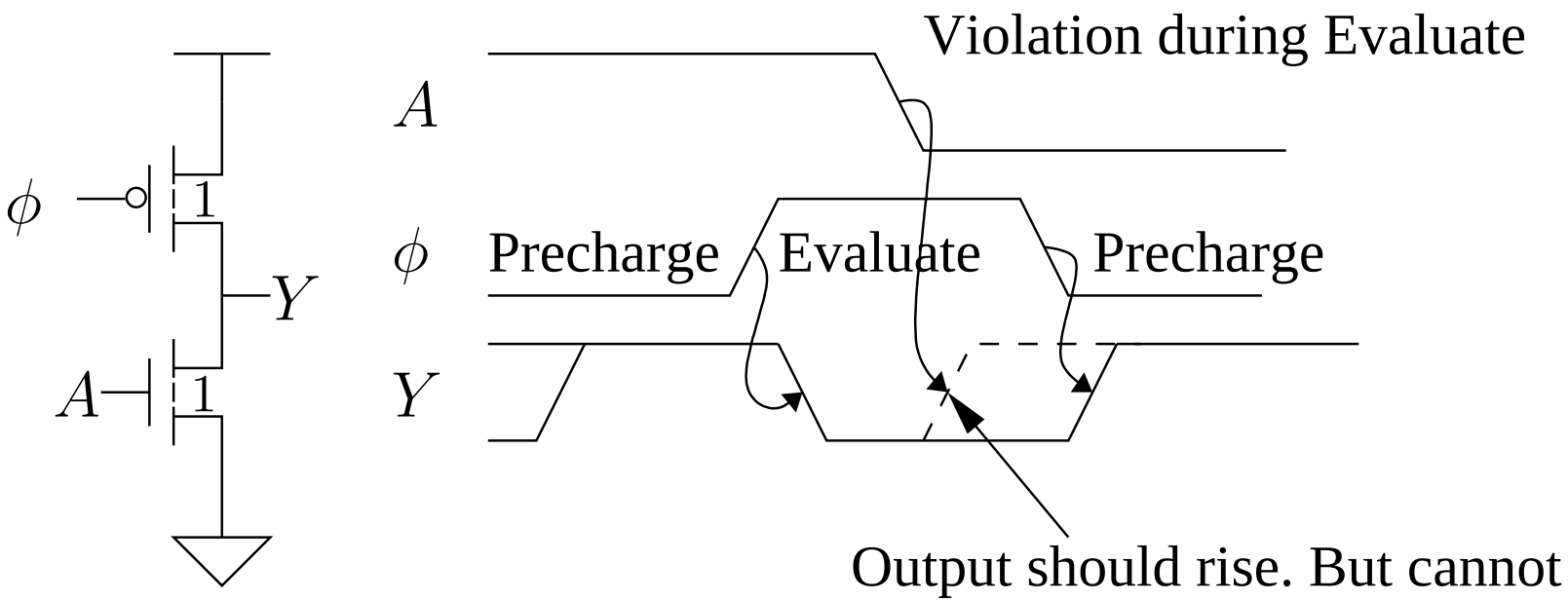
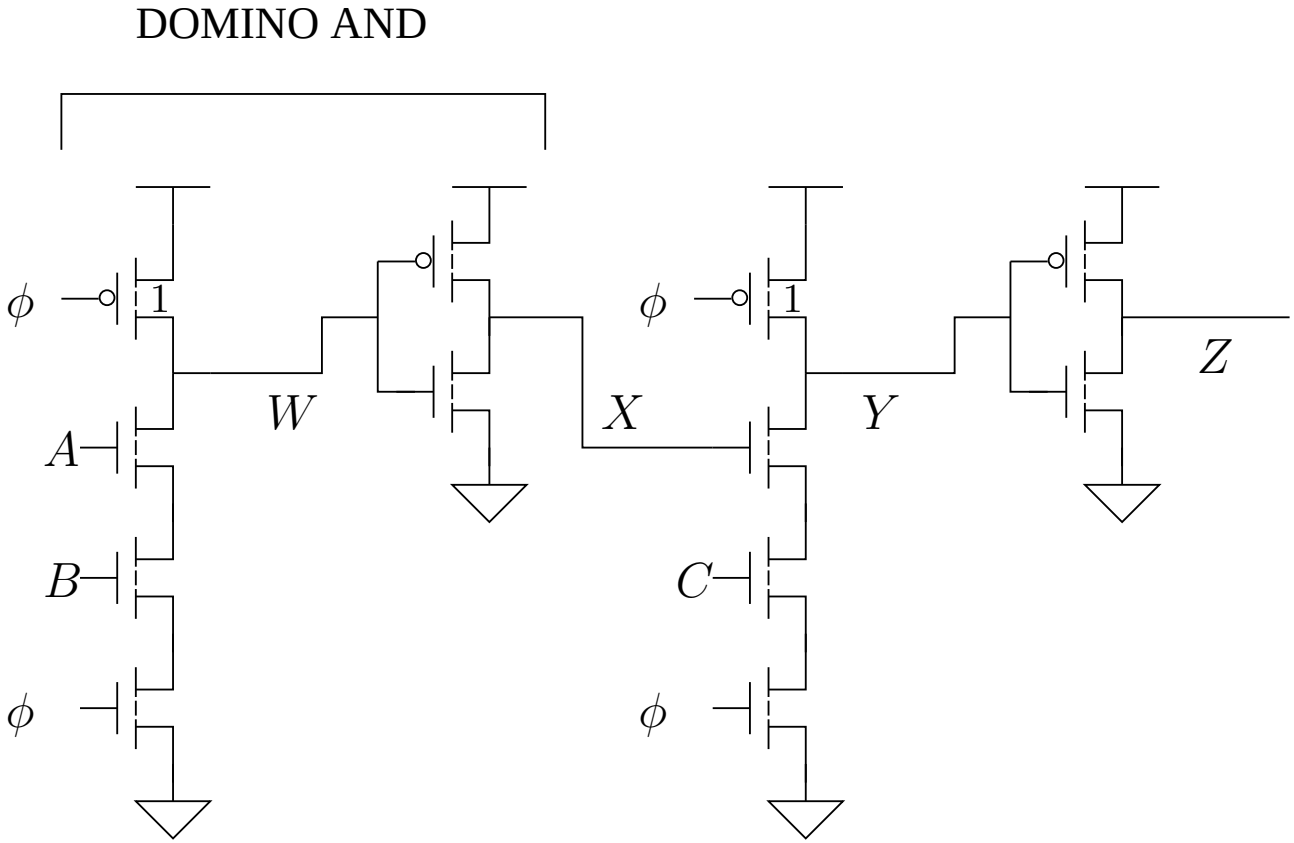


Figure: Input Monotonicity

Dynamic Circuits - Domino Logic

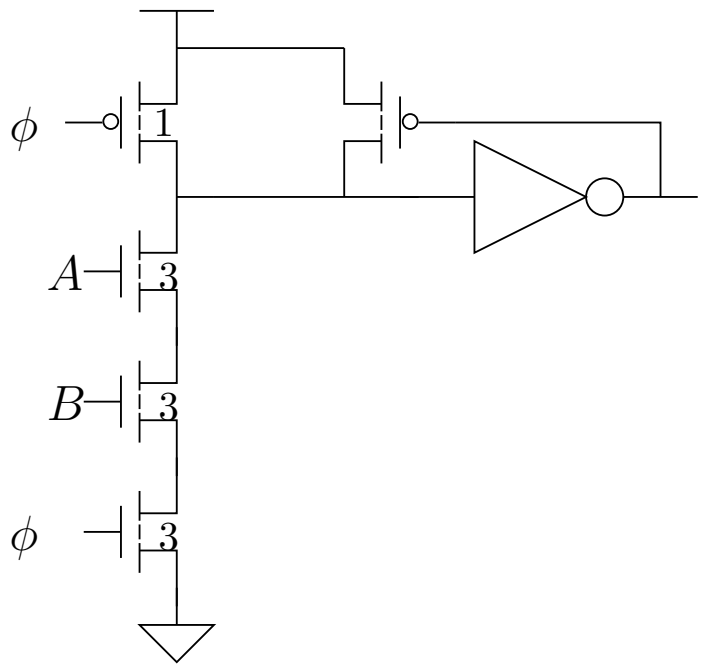
During Evaluate

- Insert a static CMOS INVERTER in between
- All dynamic gates evaluate and fall like a Domino
- All inverters rise - Skew HI inverter



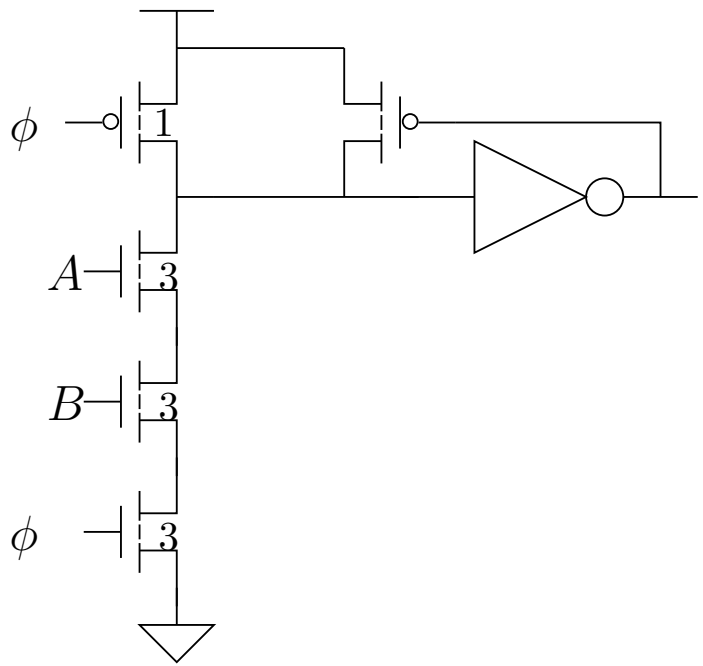
Dynamic Circuits - Keeper Transistors

- Dynamic gates suffer from charge leakage
- In evaluate mode the dynamic node is FLOATING
- Leakage current can cause it to drift and discharge



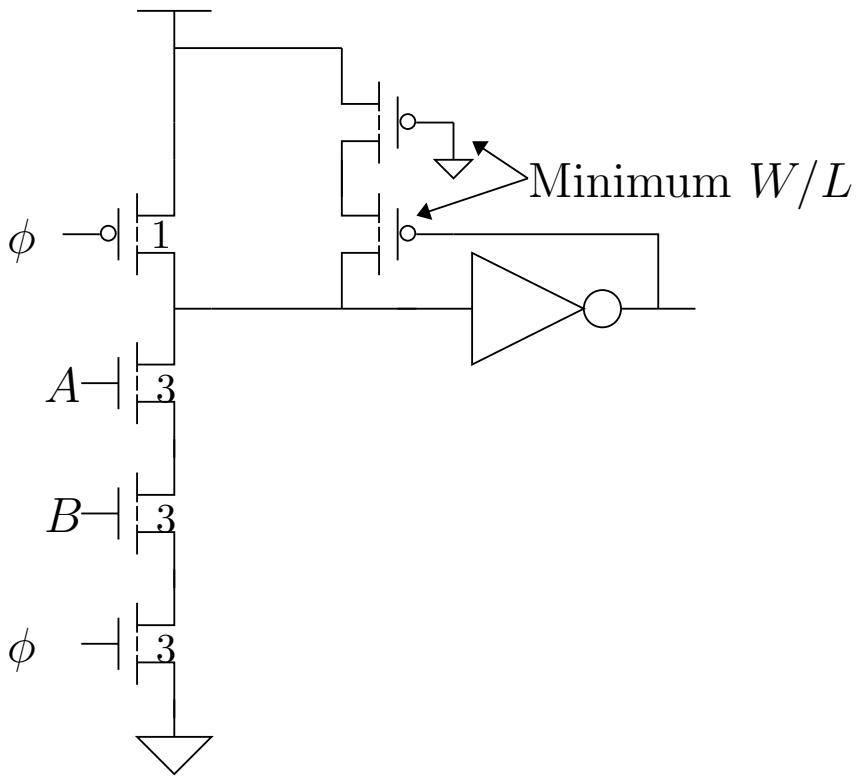
Dynamic Circuits - Keeper Transistors

- Weak keeper keeps the
- In evaluate mode the dynamic node is FLOATING
- Leakage current can cause it to drift and discharge



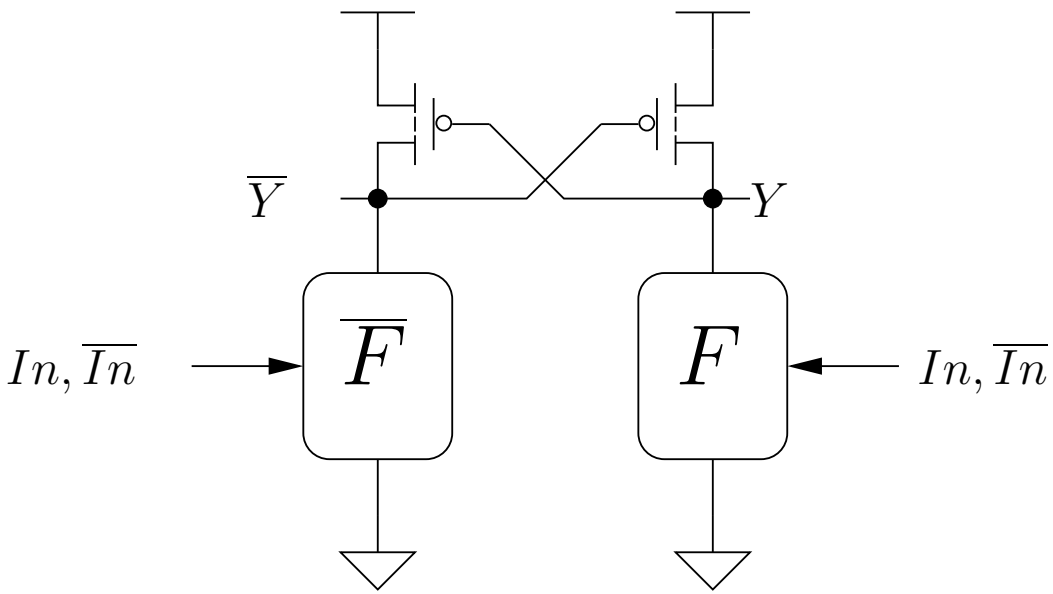
Dynamic Circuits - Weak Keepers

- Keeper size is critical
- Typically as small as possible
- Can increase L instead of decreasing W - $C \uparrow$
- Weak keeper helps solve the problem



Cascode Voltage Switch Logic

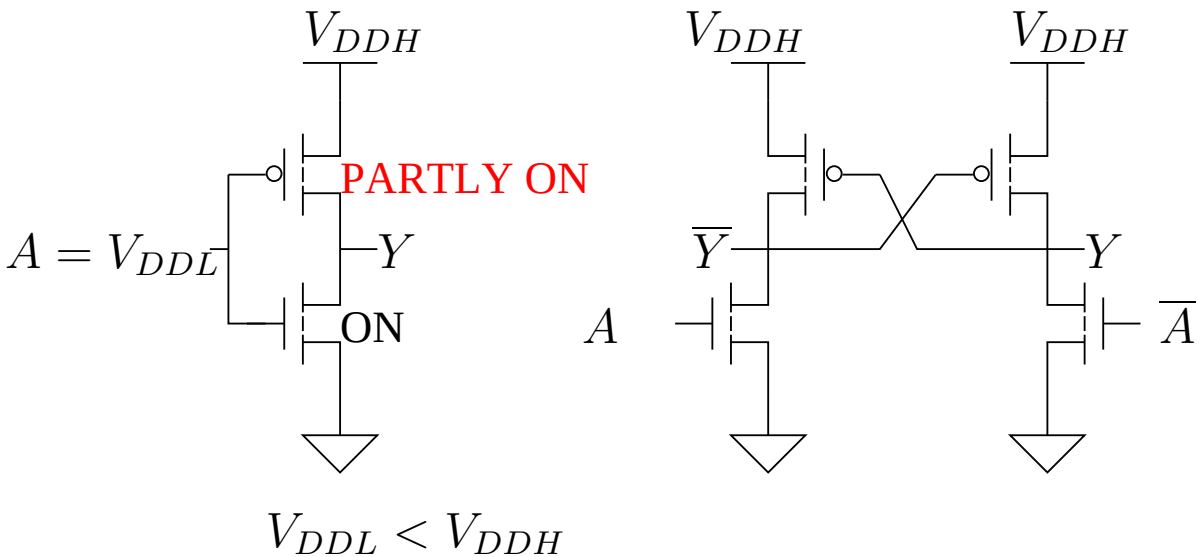
- Performance of ratioed circuits
- Avoid static leakage at the same time
- Cross coupling helps turn off the leaking side
- Needs true and complement versions of the inputs
- Output is true and complement form



Dynamic version is dual rail domino logic

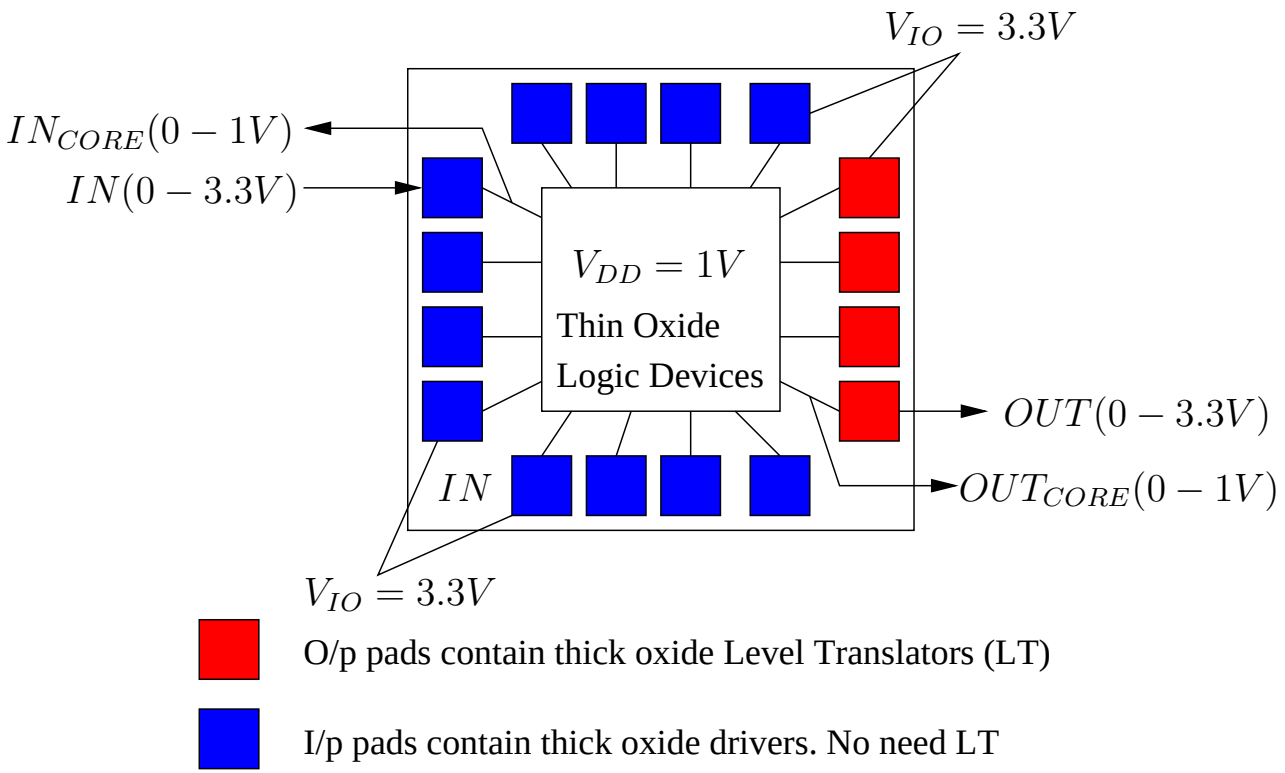
Level Translators

- Low voltage domain(V_{DDL}) driving high voltage domain (V_{DDH})- Static leakage problem
- PMOS partially turned ON when NMOS is fully ON
- Cross coupling the PMOS elimanates the issue
- High voltage driving low voltage is not an issue



Chip IO Pads

- Withstands larger voltage swing - Protects core
- IO pads use thick oxide devices $\Longrightarrow$ Larger V_{TH}
- Requires larger supply voltage (V_{IO}) to over come V_{TH}
- Typically $V_{IO} = 3.3V$ and logic $V_{DD} = 1.0V$
- Pads also contain ESD protect diodes



Transmission Gates

- Usually input is fed to the GATE terminal
- In some, they can be fed to the DRAIN/ SOURCE
- NMOS can pass ZERO and PMOS can pass ONE
- Combine the two - Pass gate or Transmission gate
- Typically used in implementing
  - XOR2
  - MUX-2
  - MUX-N

Multiplexer Implementation

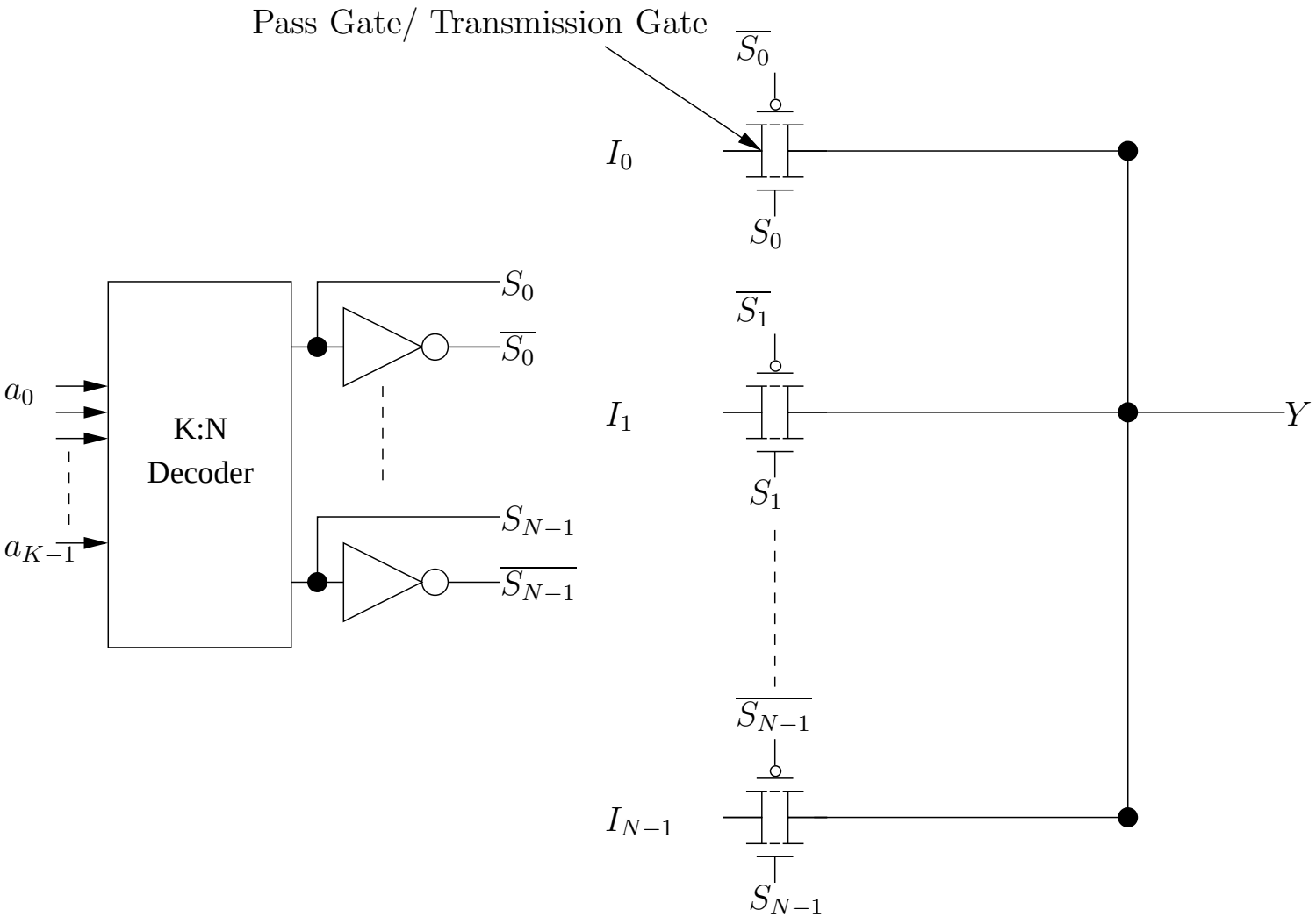


Figure: $N:1$ Mux. $N \leq 2^K$

Transmission Gate Mux 2

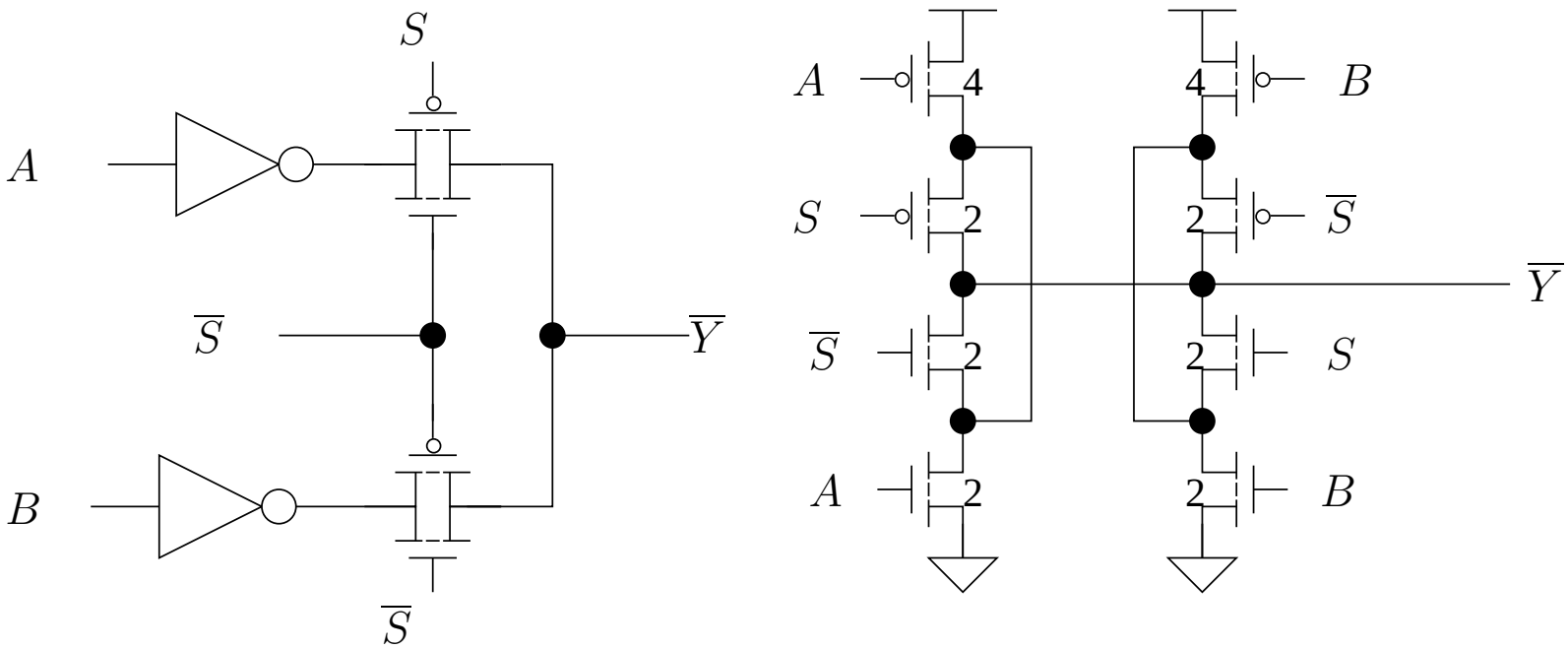


Figure: TG sizing

- NMOS and PMOS size is the same in the TG - They turn ON in parallel
- Delay of the TG gate depends on the driving gate's strength

Tri-State Inverter

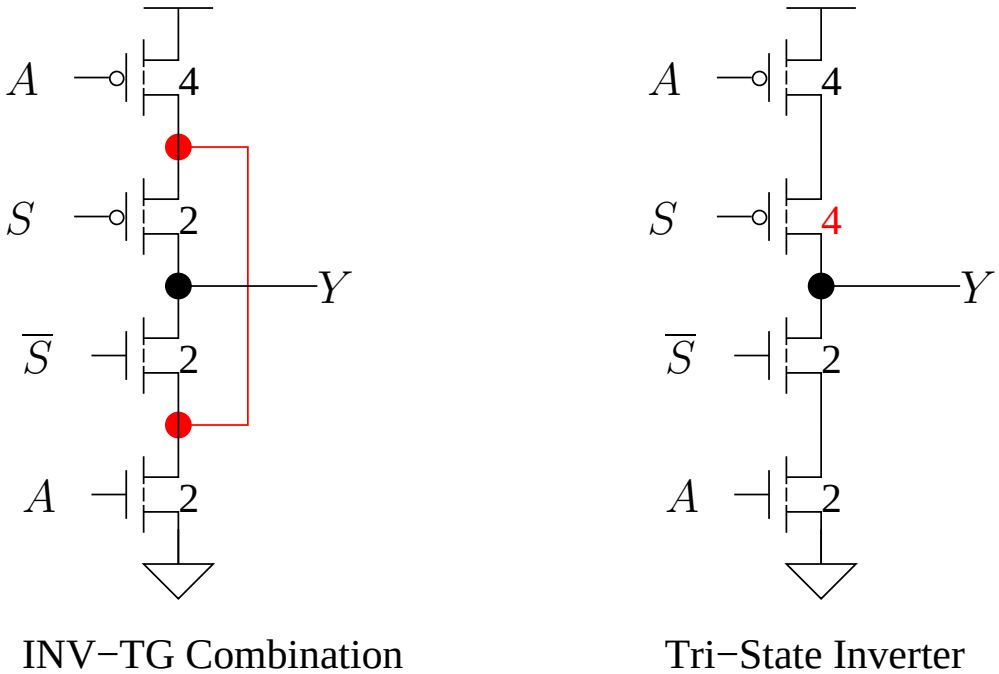
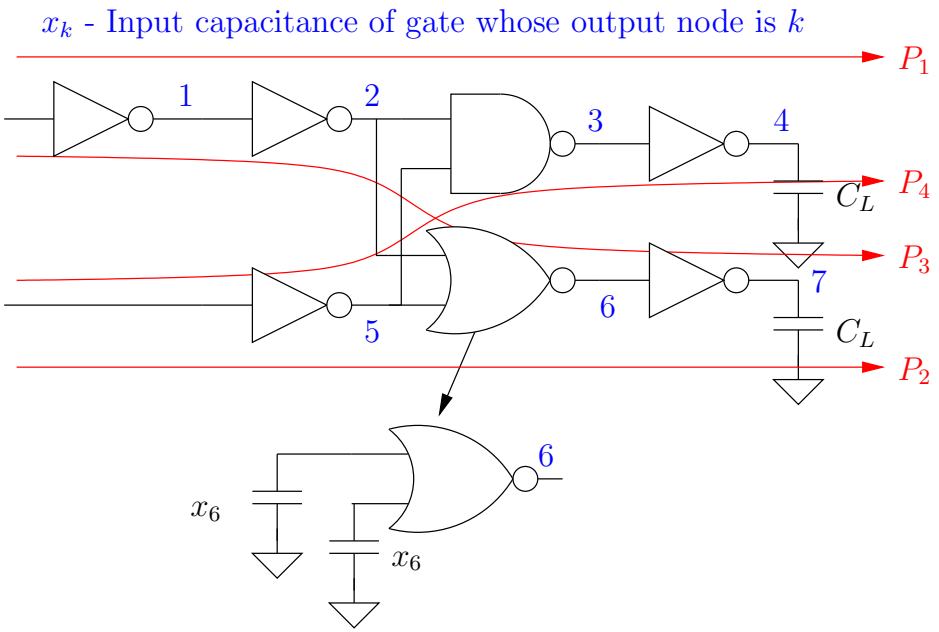


Figure: Tri State Inverter

Circuit Sizing



- Find x_k's such that all outputs settle within a time T_{Spec}
- Path P_3 is potentially the longest path - Optimize the path
- Optimizing P_3 can increase delay of P_4 and cause a timing violation
- How do we ensure timing across all paths?

Node Based Timing

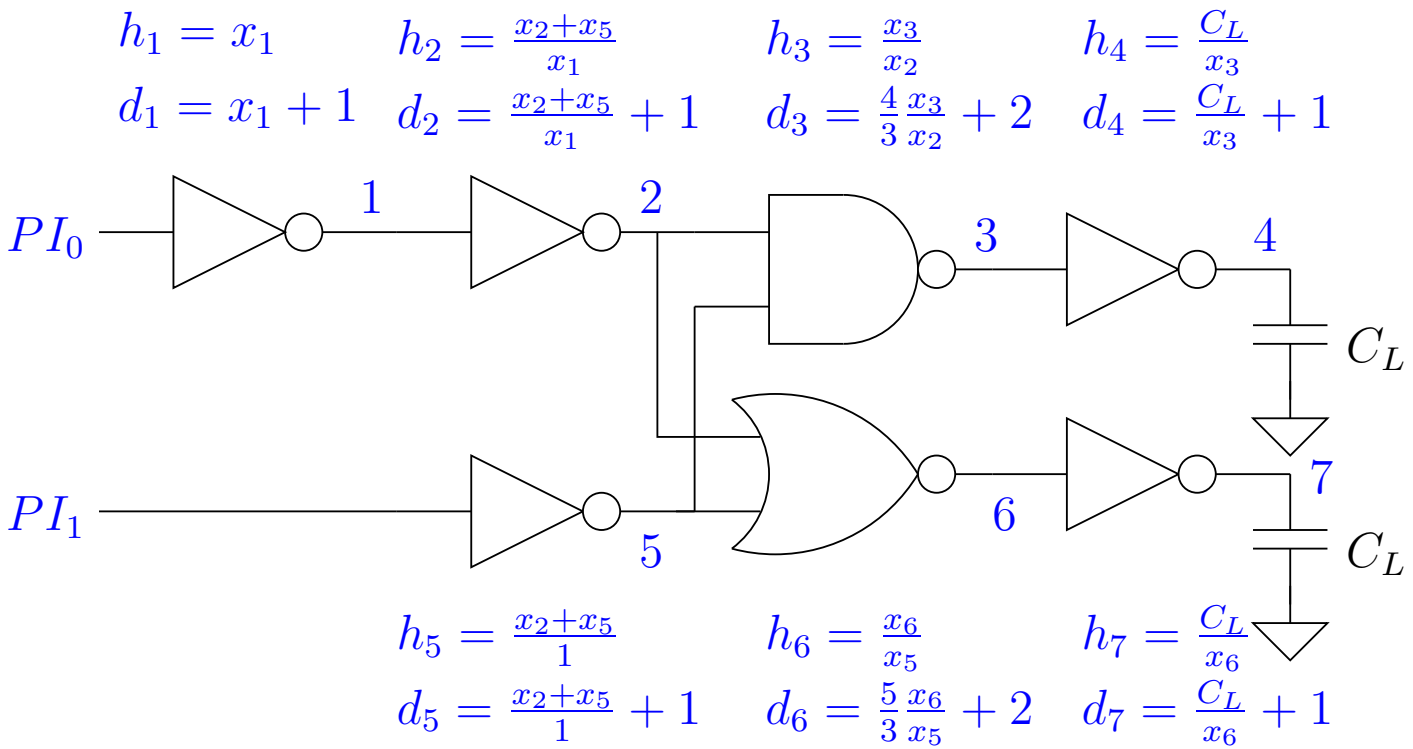
- Need to move away from path analysis
- Need to formulate a node based optimization problem

Define arrival time a_{jk} at the output of a gate due to input j

$$a_{jk} = a_j + d_{jk}$$
$$a_k = max(a_1, a_2 \ldots a_j) \ldots j \in INPUTS(Gate_k)$$
$$a_k = 0 \ldots k \in PI$$

- The arrival time(a_k) at the output of gate k is maximum of the a_{jk}'s across all inputs
- The arrival time at the primary inputs (PI) is zero.

Circuit Arrival Times



$$a_1 = d_1$$
$$a_2 = a_1 + d_2$$
$$a_3 = max(a_2 + d_3, a_5 + d_3)$$
$$a_4 = a_3 + d_4$$

$$a_5 = d_5$$
$$a_6 = max(a_2 + d_6, a_5 + d_6)$$
$$a_7 = a_6 + d_7$$
$$max(a_4, a_7) \leq T_{Spec}$$

Handling MAX operation

- MAX is not good for optimization problem
- Equalities in the constraints are also not good
- The arrival time constraints are modified as shown below

Old	Modified
$a_3 = max(a_2 + d_3, a_5 + d_3)$	$a_2 + d_3 \leq a_3$ and $a_5 + d_3 \leq a_3$
$a_6 = max(a_2 + d_6, a_5 + d_6)$	$a_2 + d_6 \leq a_6$ and $a_5 + d_6 \leq a_6$
$max(a_4, a_7) \leq T_{Spec}$	$a_4 \leq T_{Spec}$ and $a_7 \leq T_{Spec}$

Similarly, the equality constraints of remaining arrival times are converted to inequalities.

Formulation : Sizing for Minimum Delay

Objective Function : $minimize(T_{Spec})$
Subject to:

$$\begin{array}{ccc} d_1 \leq a_1 & a_2 + d_3 \leq a_3 & a_5 + d_3 \leq a_3 \\ a_1 + d_1 \leq a_2 & a_2 + d_6 \leq a_6 & a_5 + d_6 \leq a_6 \\ d_5 \leq a_5 & a_4 \leq T_{Spec} & a_7 \leq T_{Spec} \end{array}$$

- This is a convex problem and hence the solution obtained is a *Global Optimum*
- For a circuit with N gates the number of constraints is $O(N)$

Formulation : Minimum Area for a given T_{spec}

Objective Function : $minimize(\sum_{k=1}^{6} x_k)$
Subject to:

$$\begin{array}{ccc} d_1 \leq a_1 & a_2 + d_3 \leq a_3 & a_5 + d_3 \leq a_3 \\ a_1 + d_1 \leq a_2 & a_2 + d_6 \leq a_6 & a_5 + d_6 \leq a_6 \\ d_5 \leq a_5 & a_4 \leq T_{Spec} & a_7 \leq T_{Spec} \end{array}$$

This is a convex problem and hence the solution obtained is a *Global Optimum*

References

Most of the material presented here is based on the first reference [Weste]. The circuit sizing alone is based on references 2 and 3 [Boyd].

1. CMOS VLSI Design, Neil H.E. Weste, David Harris and Ayan Banerjee, 3rd Edition, Pearson Education
2. S. Boyd, S.-J. Kim, D. D. Patil, M. A. Horowitz, "Digital circuit optimization via geometric programming", Operations Research, vol. 53, no. 6, pp. 899-932, Nov. 2005.
3. http://web.stanford.edu/ boyd/papers/pdf/date05.pdf