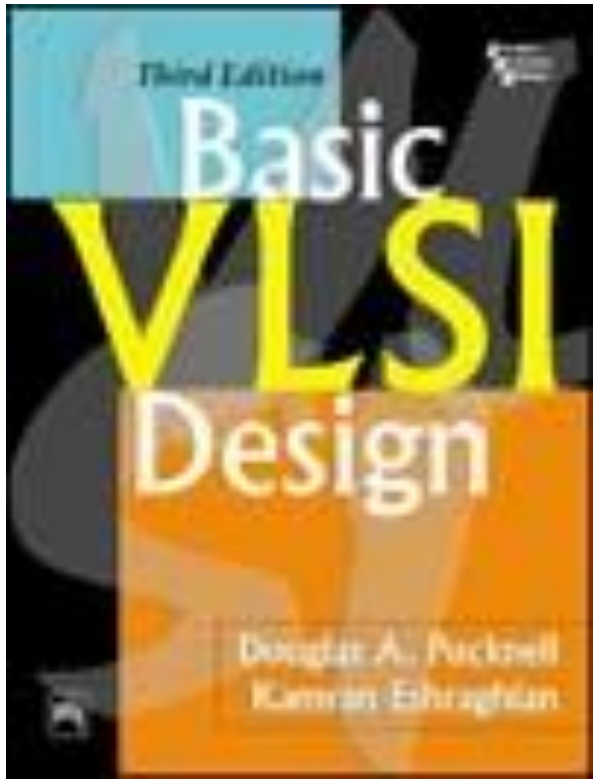
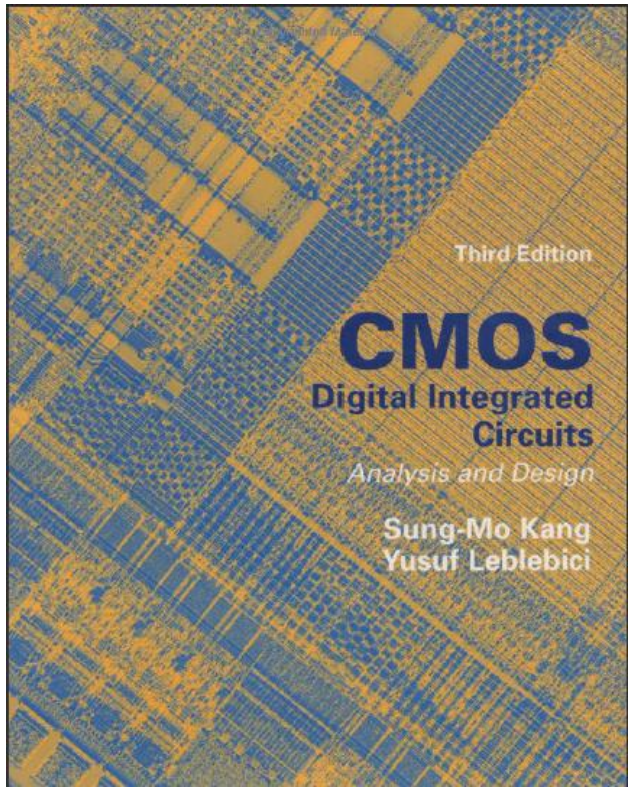
VLSI Design

Reference Books

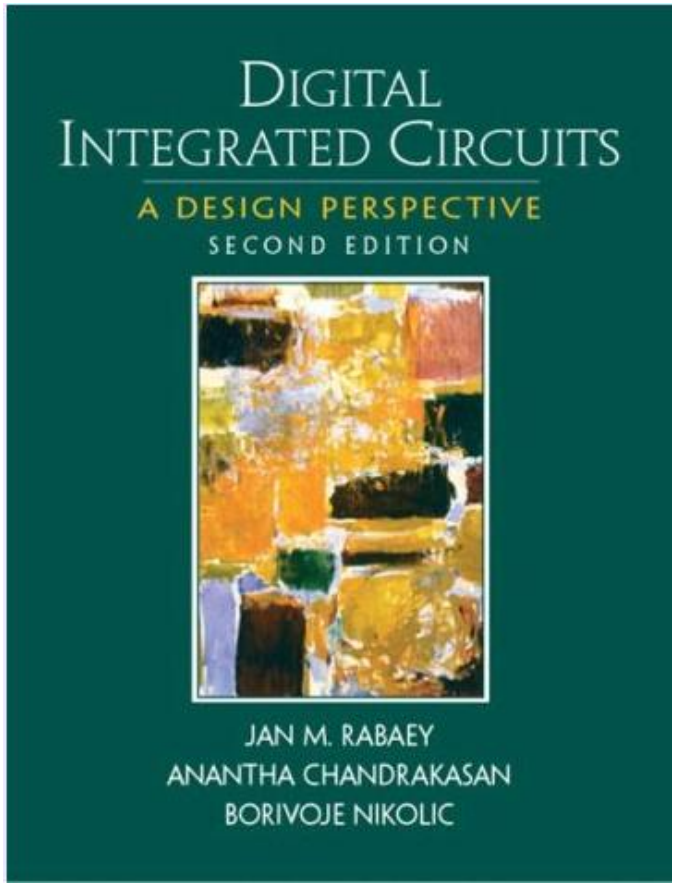


“Basic VLSI Design”, 3rd Ed., PHI
By: PUCKNELL DOUGLAS A.ESHRAGHIAN, KAMRAN

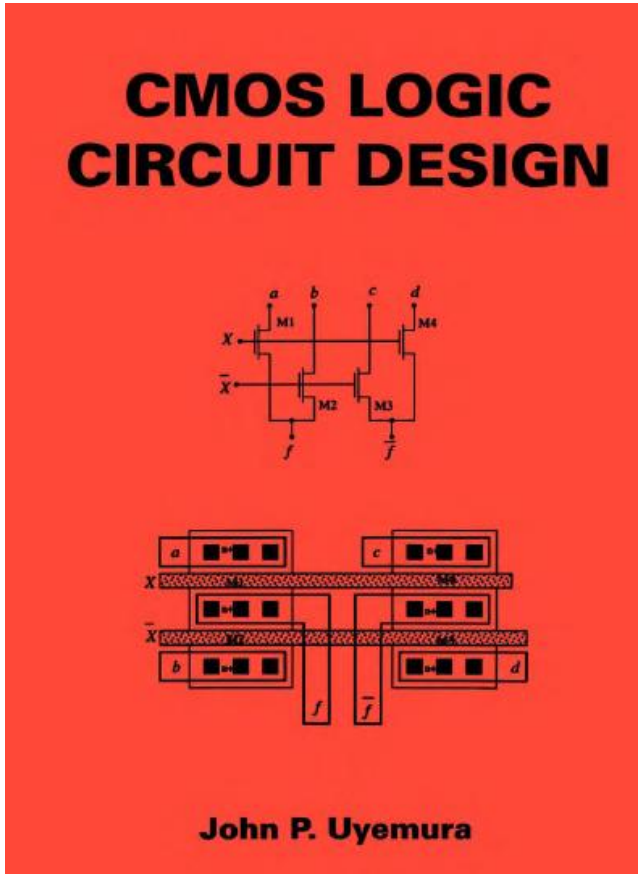


“CMOS DIGITAL INTEGRATED CIRCUITS:Analysis and Design”
3rd Ed. McGraw-Hill.
By:SUNG-MO (STEVE) KANG

Reference Books

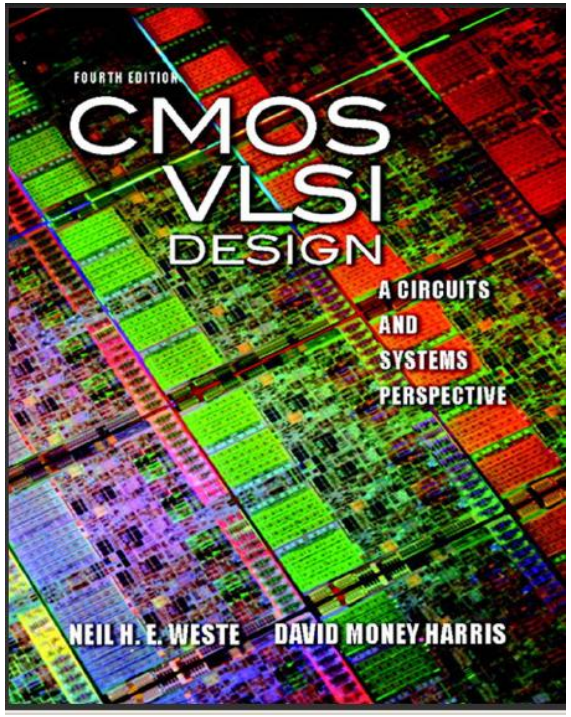


Digital Integrated Circuits – A Design Perspective”, 2nd ed. by J. Rabaey, A. Chandrakasan, B. Nikolic



“CMOS LOGIC CIRCUIT DESIGN”
KLUWER ACADEMIC PUBLISHERS
By: John P. Uyemura

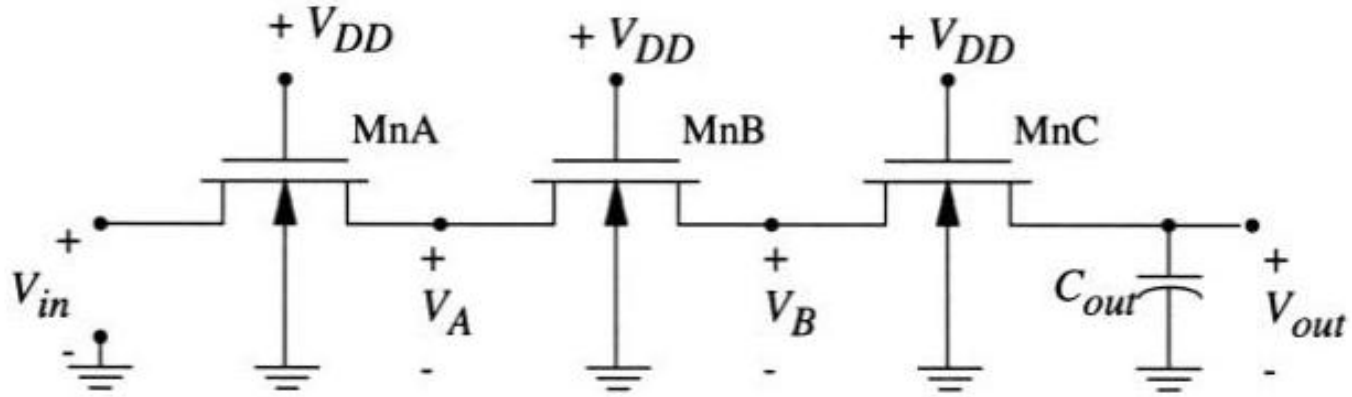
Reference Books

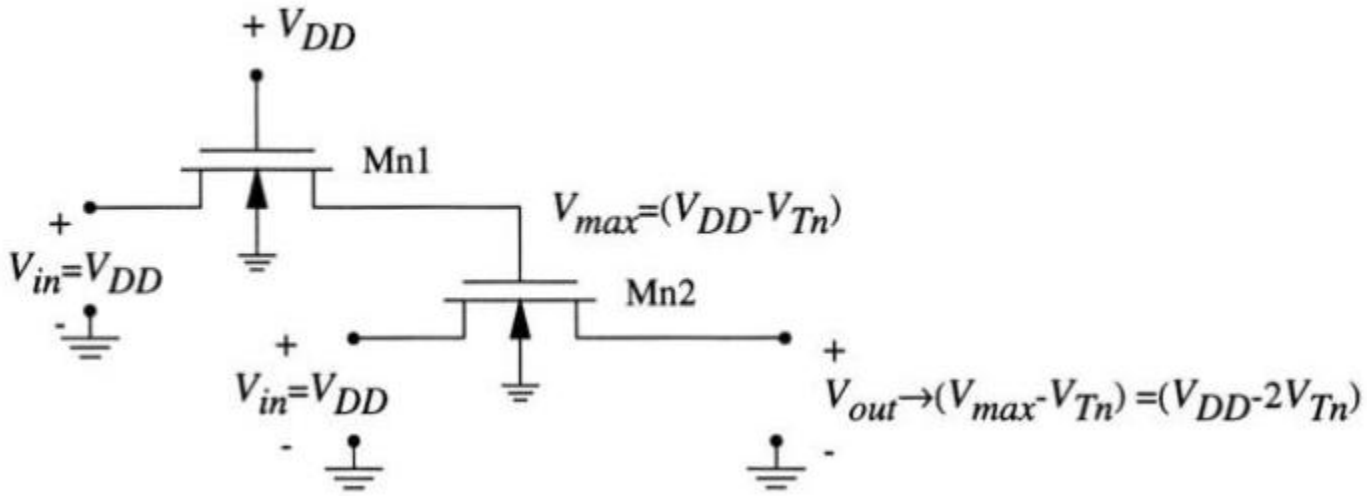


"CMOS VLSI Design:A Circuits and Systems Perspective" 4th ed. Addison-Wesley
By: Neil H. E. Weste, David Money Harris

Pass Transistor and Transmission Gate

Pass Transistors (NMOS)

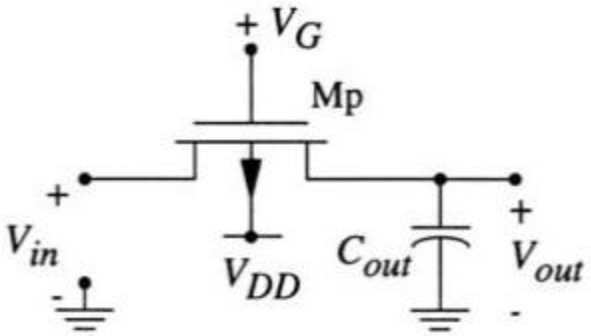


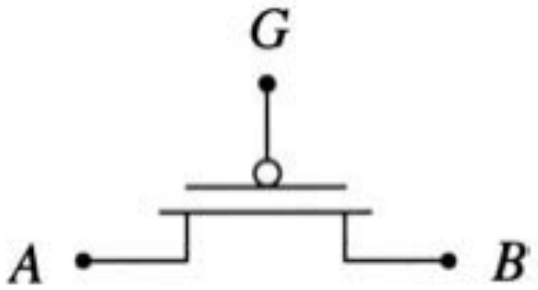


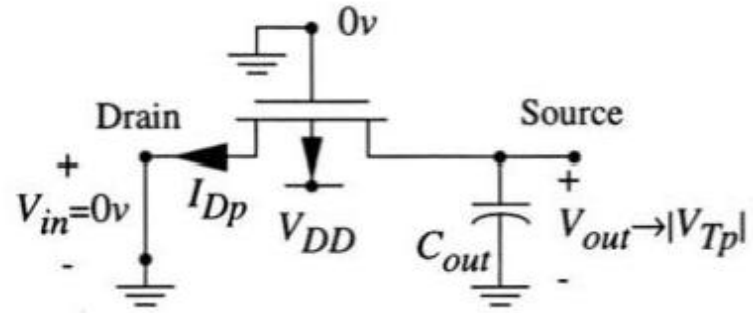
NMOS

- NMOS Pass transistor passes strong logic 0.
- NMOS Pass transistor passes weak logic 1.

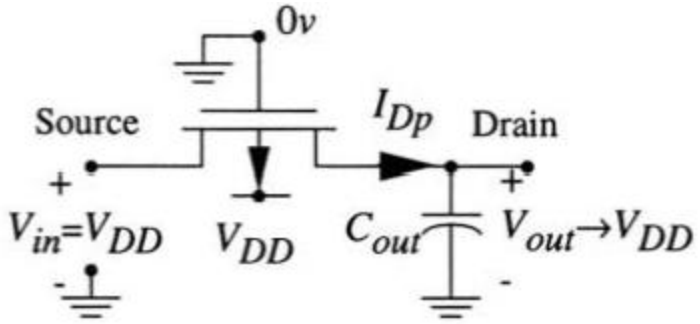
Pass Transistors (PMOS)





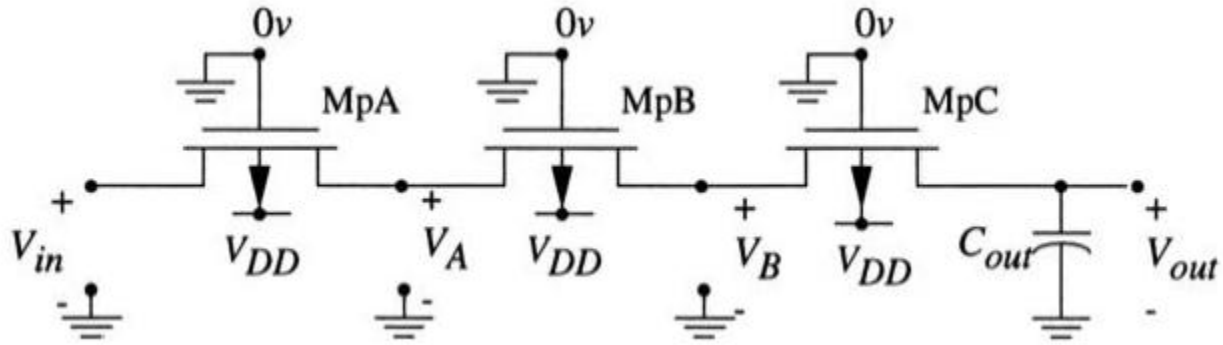


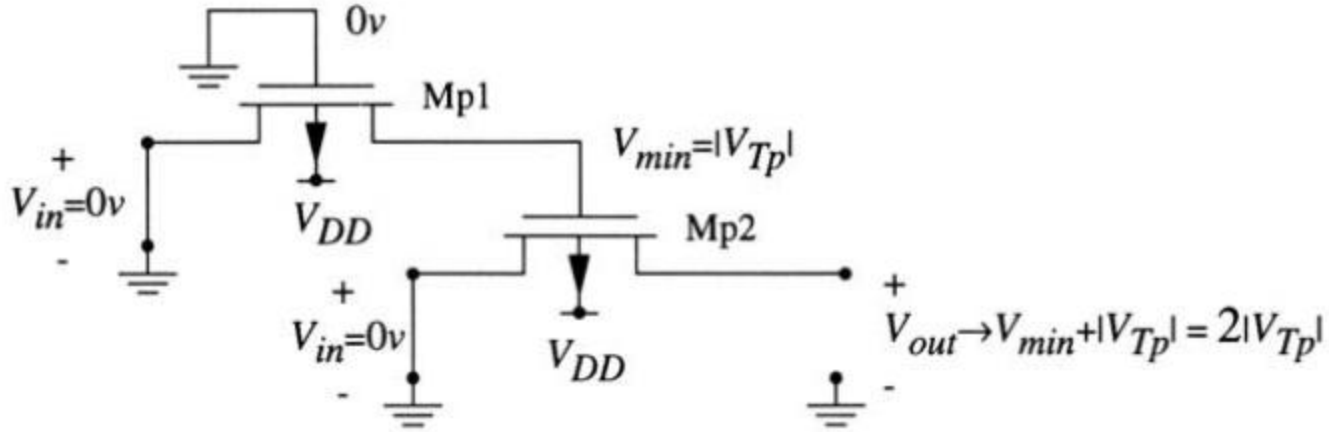
(a) Logic 0 input



(b) Logic 1 input

Pass Transistors (PMOS)

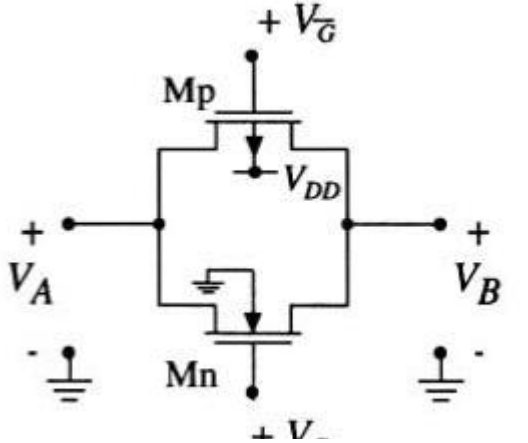


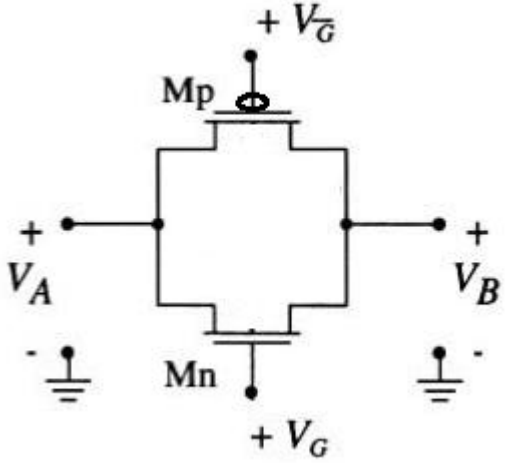


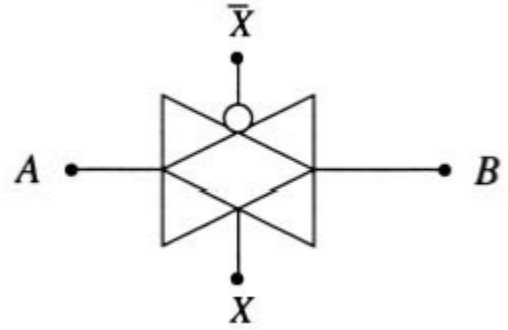
PMOS

- PMOS Pass transistor passes weak logic 0.
- PMOS Pass transistor passes strong logic 1.

Transmission Gate



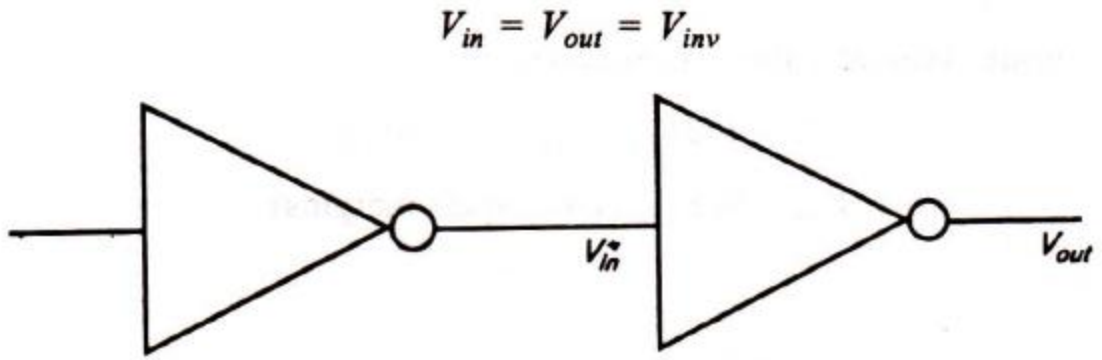




X	A	B
1	0	0
1	1	1
0	0	?
0	1	?

PMOS Pass transistor passes strong logic 1.
NMOS Pass transistor passes strong logic 0.

Determination of pull-up to pull-down ratio *(Zp.U/ Zp.D.)* For An Nmos Inverter Driven By Another Nmos Inverter



V_{inv}=0.5V_{DD}

$$I_{ds} = K\frac{W}{L}\frac{(V_{gs} - V_t)^2}{2}$$

Note→ $K = \frac{\mu\varepsilon_{ins}\varepsilon_0}{t_{ox}}$

In the depletion mode

$$I_{ds} = K\frac{W_{p.u.}}{L_{p.u.}}\frac{(-V_{td})^2}{2} \text{ since } V_{gs} = 0$$

and in the enhancement mode

$$I_{ds} = K\frac{W_{p.d.}}{L_{p.d.}}\frac{(V_{inv} - V_t)^2}{2} \text{ since } V_{gs} = V_{inv}$$

Equating (since currents are the same) we have

$$\frac{W_{p.d.}}{L_{p.d.}}(V_{inv} - V_t)^2 = \frac{W_{p.u.}}{L_{p.u.}}(-V_{td})^2$$

$$Z_{p.d.} = \frac{L_{p.d.}}{W_{p.d.}};\ Z_{p.u.} = \frac{L_{p.u.}}{W_{p.u.}}$$

$$\frac{1}{Z_{p.d.}}(V_{inv} - V_t)^2 = \frac{1}{Z_{p.u.}}(-V_{td})^2$$

$$V_{inv} = V_t - \frac{V_{td}}{\sqrt{Z_{p.u.}/Z_{p.d.}}}$$

substitute typical values as follows

$$V_t = 0.2V_{DD};\ V_{td} = -0.6V_{DD}$$

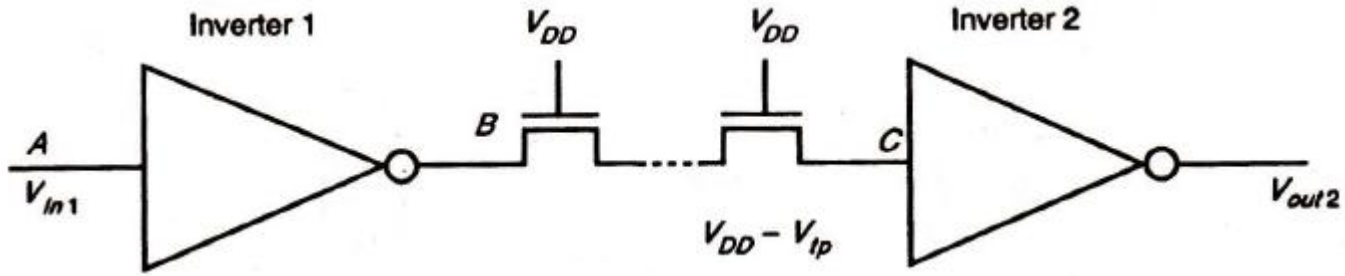
$$V_{inv} = 0.5V_{DD} \text{ (for equal margins)}$$

$$0.5 = 0.2 + \frac{0.6}{\sqrt{Z_{p.u.}/Z_{p.d.}}}$$

$$\sqrt{Z_{p.u.}/Z_{p.d.}} = 2$$

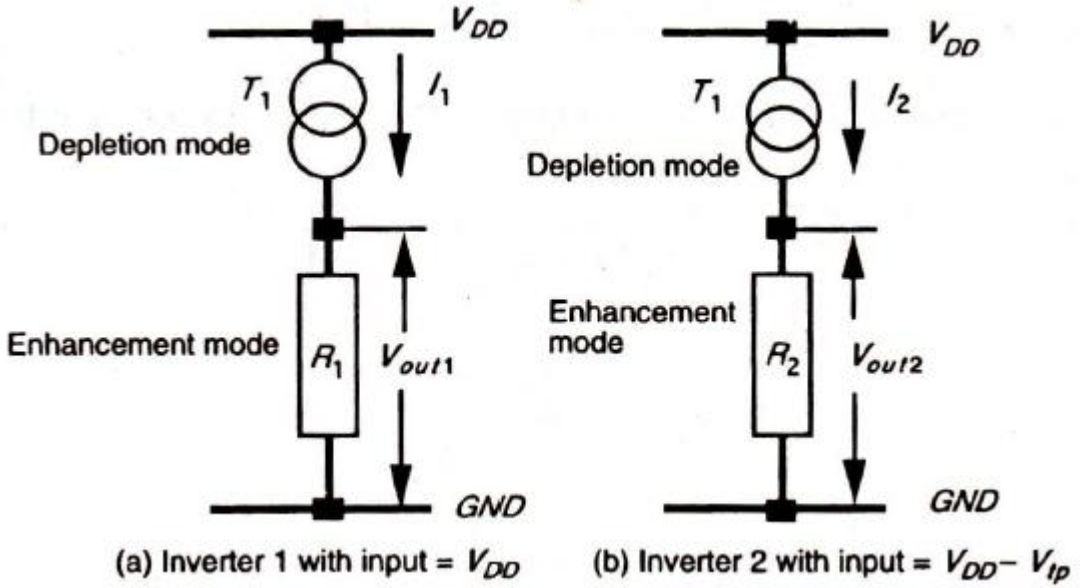
$$Z_{p.u.}/Z_{p.d.} = 4/1$$

Pull-up to pull-down ratio for an Nmos inverter driven through one or more pass transistors



$$V_{in2} = V_{DD} - V_{tp}$$

V_{tp} = threshold voltage for a pass transistor



For the p.d. transistor

$$I_{ds} = K \frac{W_{p.d.1}}{L_{p.d.1}} \left((V_{DD} - V_t) V_{ds1} - \frac{V_{ds1}^2}{2} \right)$$

$$R_1 = \frac{V_{ds1}}{I_{ds}} = \frac{1}{K} \frac{L_{p.d.1}}{W_{p.d.1}} \left(\frac{1}{V_{DD} - V_t - \frac{V_{ds1}}{2}} \right)$$

$$R_1 \doteq \frac{1}{K} Z_{p.d.1} \left(\frac{1}{V_{DD} - V_t} \right)$$

Now, for depletion mode p.u. in saturation with $V_{GS} = 0$

$$I_1 = I_{ds} = K \frac{W_{p.u.1}}{L_{p.u.1}} \frac{(-V_{td})^2}{2}$$

Note→ $K = \frac{\mu \varepsilon_{ins} \varepsilon_0}{t_{ox}}$

$$I_1 R_1 = V_{out\,1}$$

$$V_{out\,1} = I_1 R_1 = \frac{Z_{p.d.1}}{Z_{p.u.1}} \left(\frac{1}{V_{DD} - V_t} \right) \frac{(V_{td})^2}{2}$$

Consider inverter 2 (Figure 2.10(b)) when input = *VDD*- V_{tp}.

$$R_2 \doteq \frac{1}{K} Z_{p.d.2} \frac{1}{((V_{DD} - V_{tp}) - V_t}$$

$$I_2 = K \frac{1}{Z_{p.u.2}} \frac{(-V_{td})^2}{2}$$

Taking typical values

$$V_t = 0.2V_{DD}$$

$$V_{tp} = 0.3V_{DD}*$$

$$\frac{Z_{p.u.2}}{Z_{p.d.2}} = \frac{Z_{p.u.1}}{Z_{p.d.1}} \frac{0.8}{\mathbf{0.5}}$$

$$\frac{Z_{p.u.2}}{Z_{p.d.2}} \div 2 \frac{Z_{p.u.1}}{Z_{p.d.1}} = \frac{8}{1}$$

2016 End Sem

Compare BJT and MOSFET with respect to following parameters:

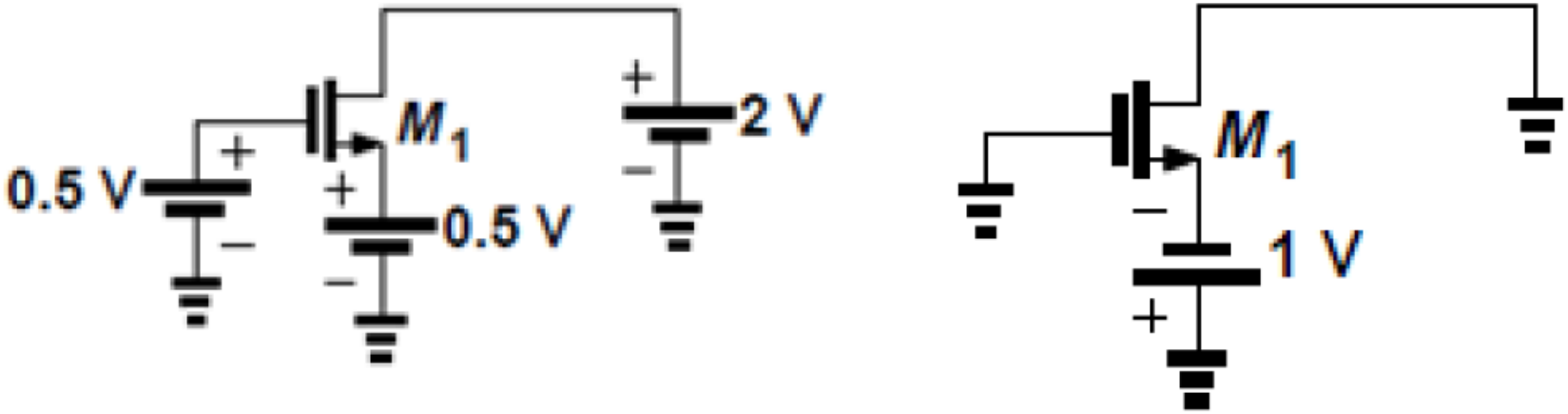
(i) Transconductance value

(ii) Input impedance

(iii) Noise

(iv) Current handling capability

Find the region of operation for NMOS based circuits given in **FIG. 3C**. Assume VT = 0.4V



Find the region of operation for the circuit given in **FIG. Q 2C** (i) and (ii). Threshold voltage VTH = 0.4 V.

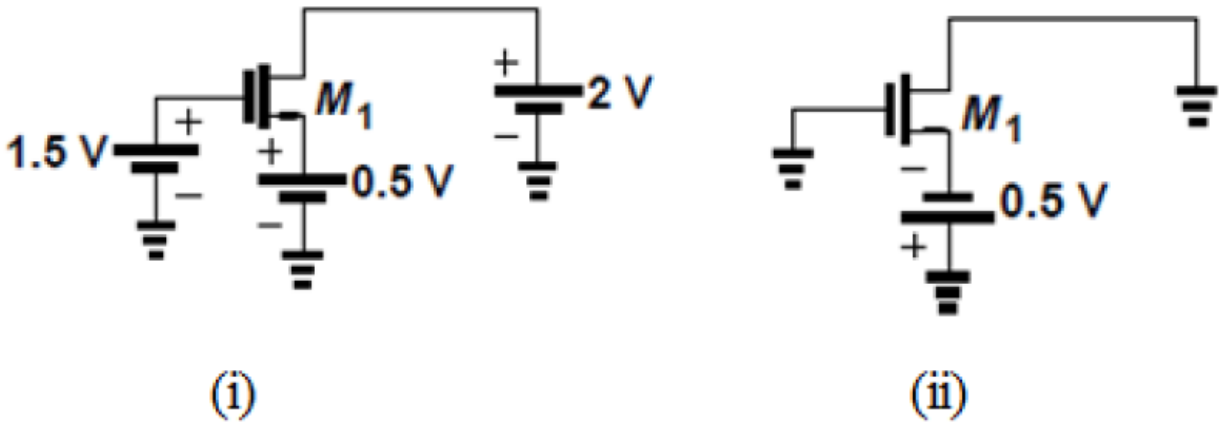


FIG. Q 2C

BiCMOS Inverter

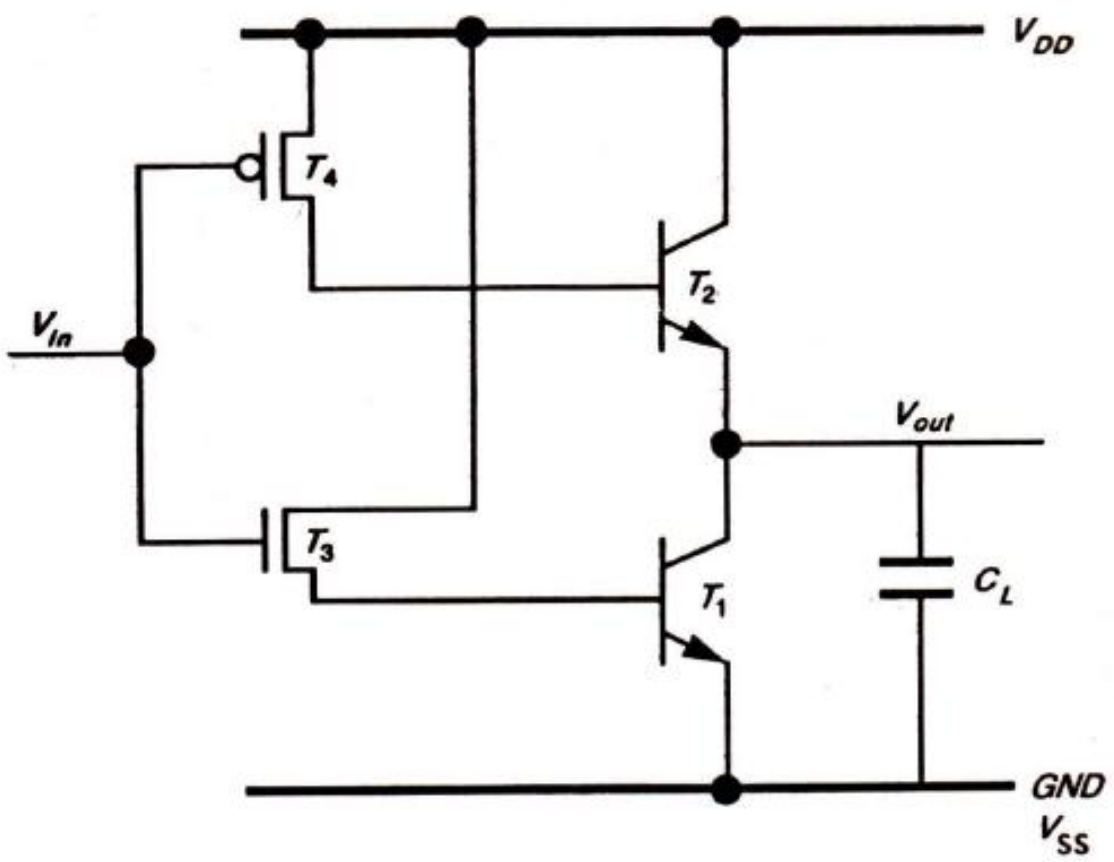
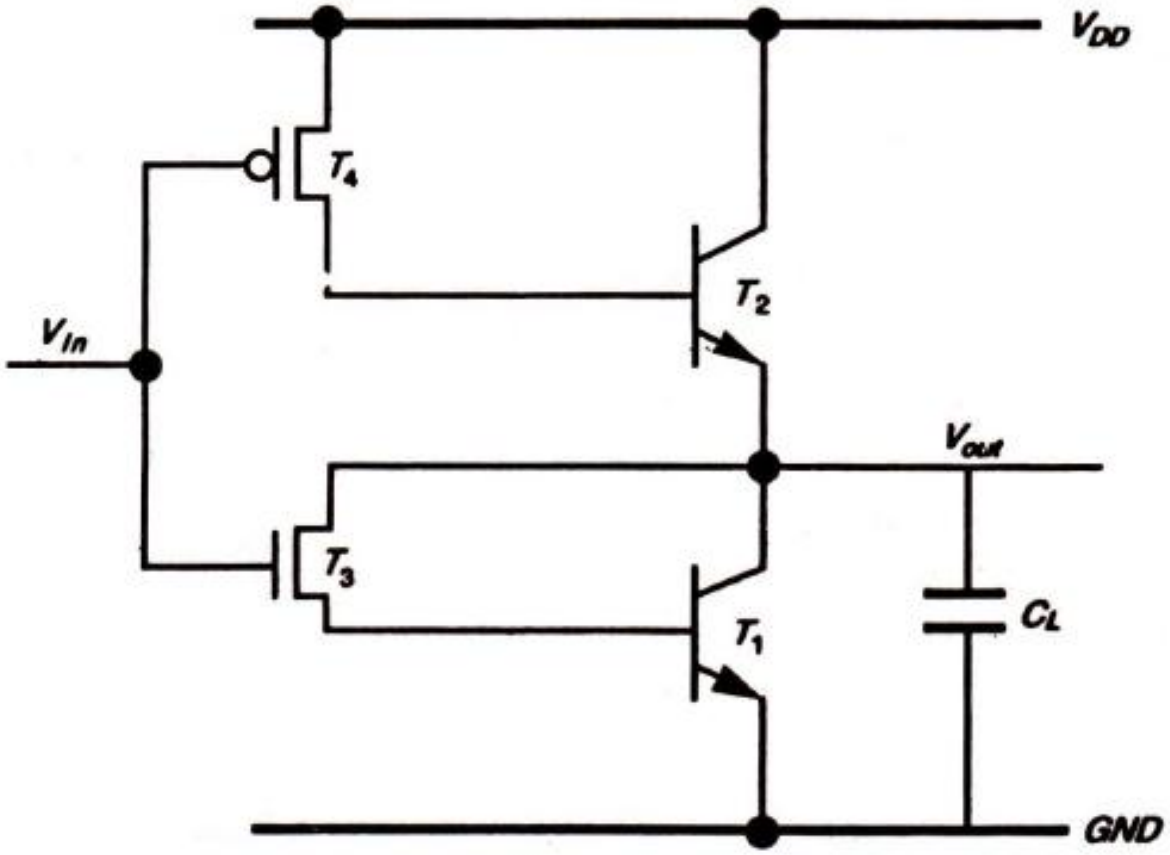
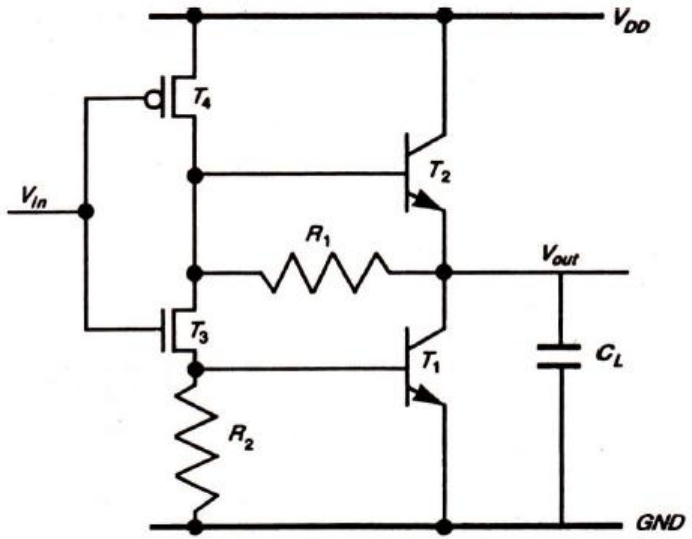


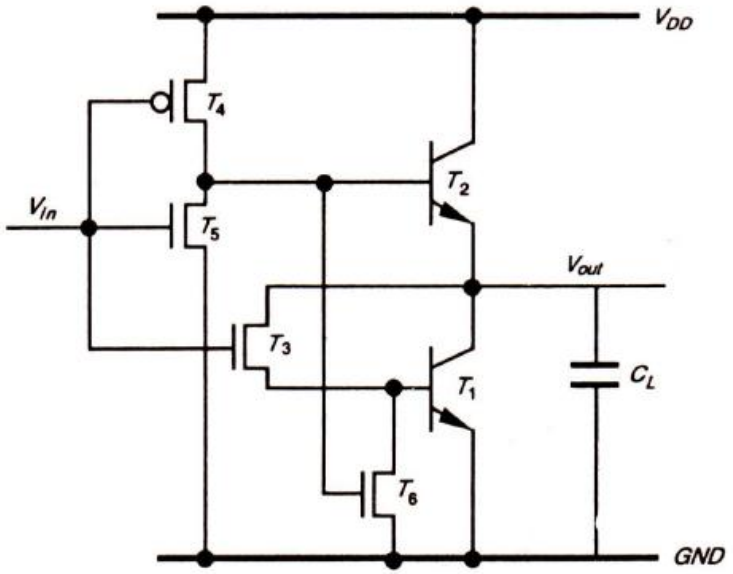
FIGURE 2.17 A simple BiCMOS inverter.



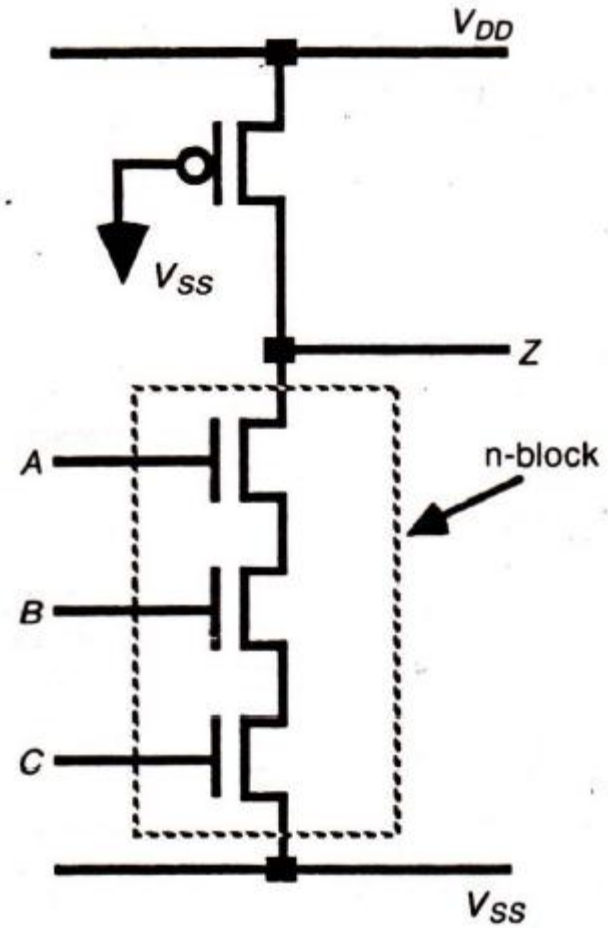
BiCMOS Inverter

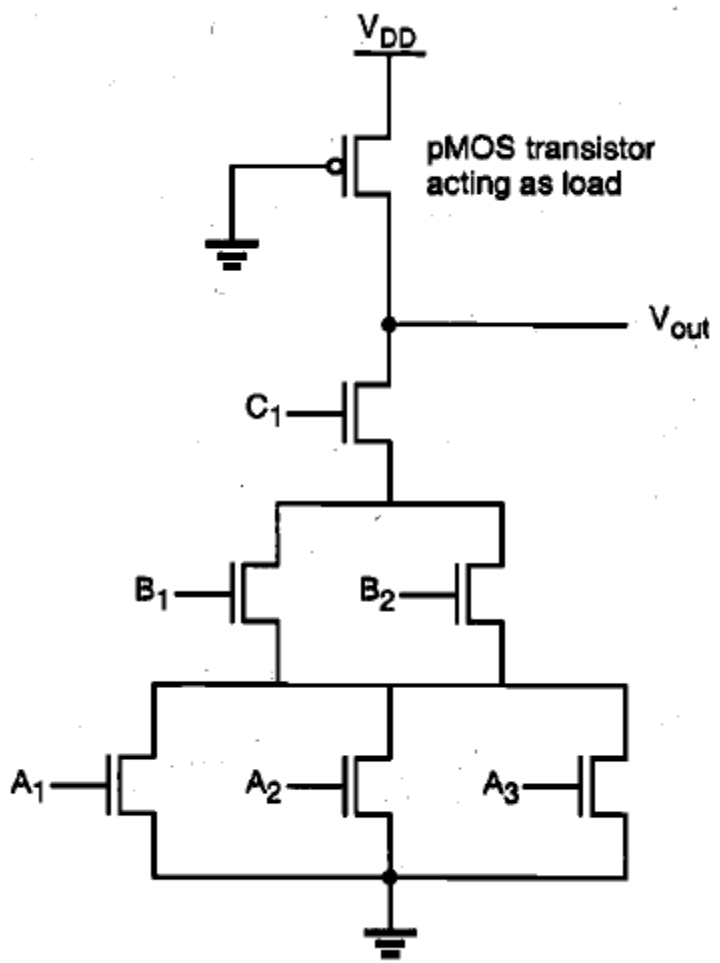


RE 2.19 An improved BiCMOS inverter with better output logic levels.

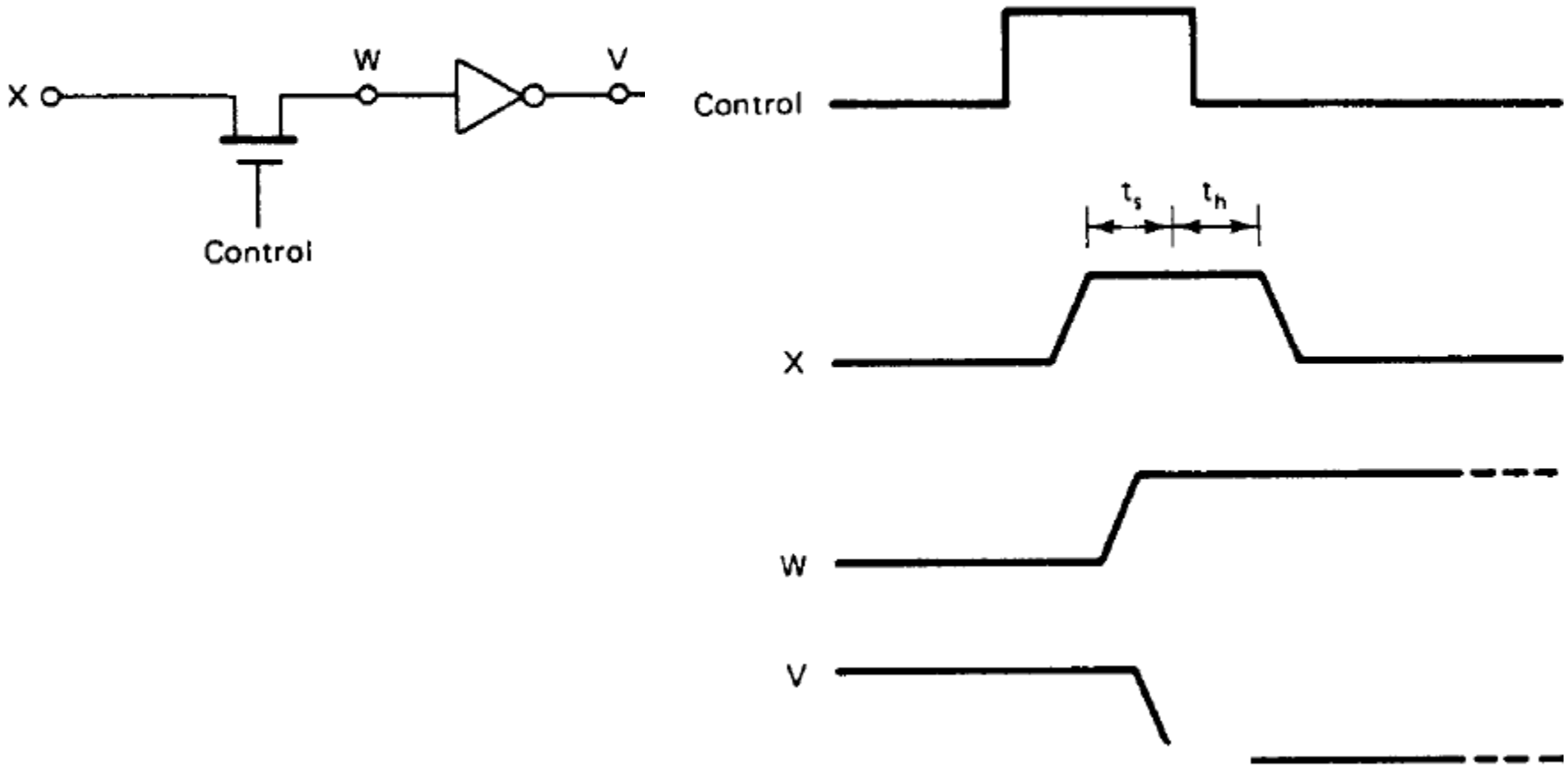


Pseudo NMOS Logic

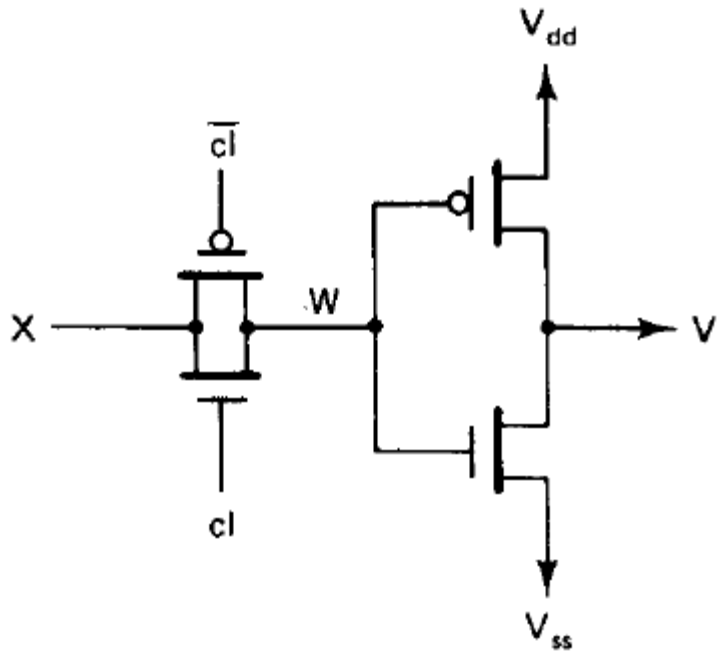




NMOS Dynamic latch



CMOS dynamic latch

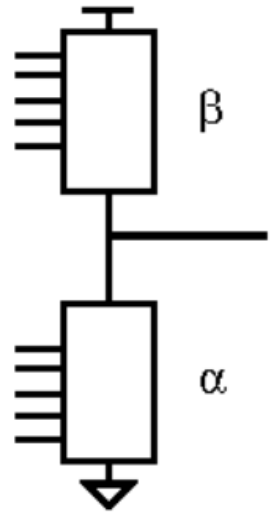


Dynamic Logic

Static logic

To build a logic gate $\bar{f}(x_1, \ldots, x_n)$, need to build two switch networks:

The pullup network connects the output to Vdd when f is false.



pMOS only, since only passes 1

The pulldown network connects the output to Gnd when f is true.

nMOS only, since only passes 0

Pulldown

$$\alpha(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$$

Pullup

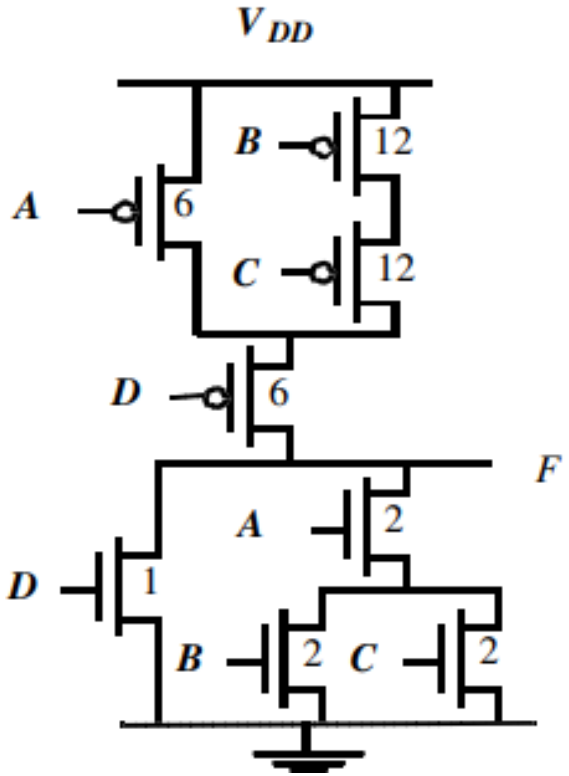
$$\beta(\bar{x}_1, \ldots, \bar{x}_n) = \bar{f}(x_1, \ldots, x_n)$$ (since pMOS invert inputs)

Static CMOS logic

1. **High noise margins** : V_{OH} and V_{OL} are at V_{DD} and GND, respectively.
2. **No static power consumption** : There never exists a direct path between VDD and VSS (GND) in steady-state mode.
3. **Comparable rise and fall times**: (under the appropriate scaling conditions)

Static CMOS Sizing

- for symmetrical response (dc, ac)
- for performance



Input Dependent

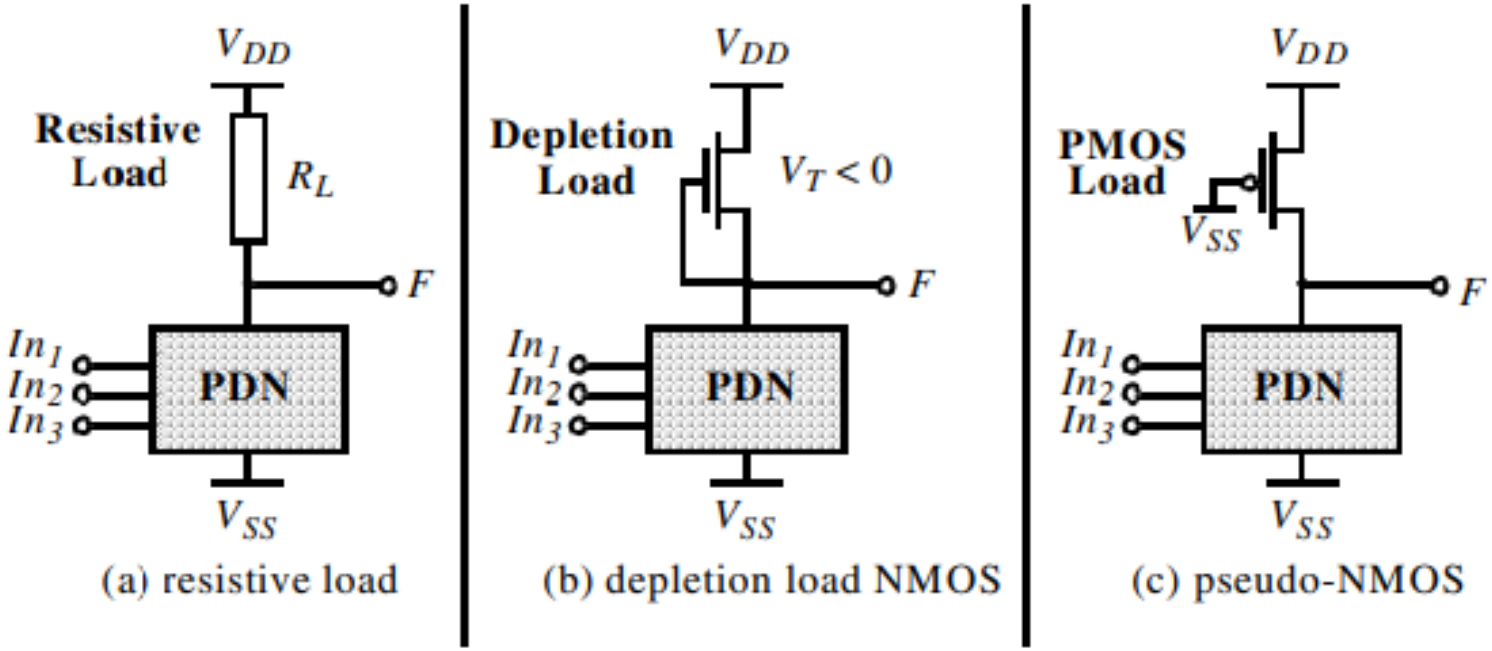
Focus on worst-case

- assume $\mu_n = 3^* \mu_p$ (i.e. n-channel transistors has 3 times the transconductance as that of p-channel.)

Comparing with equivalent Inverter

Problem: Area, as number of transistor are doubled.
Too many series transistors can effect switching speed.

Ratioed logic



Goal: to reduce the number of devices over complementary CMOS

PMOS size
0.25 of NMOS

Psuedo-NMOS

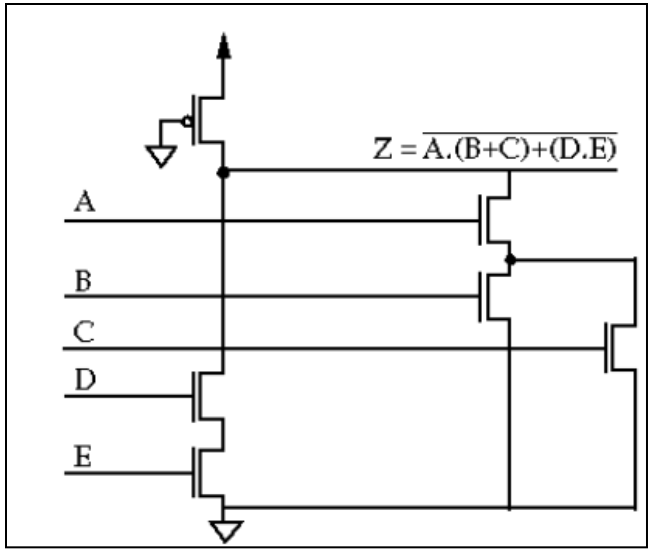
Implement:

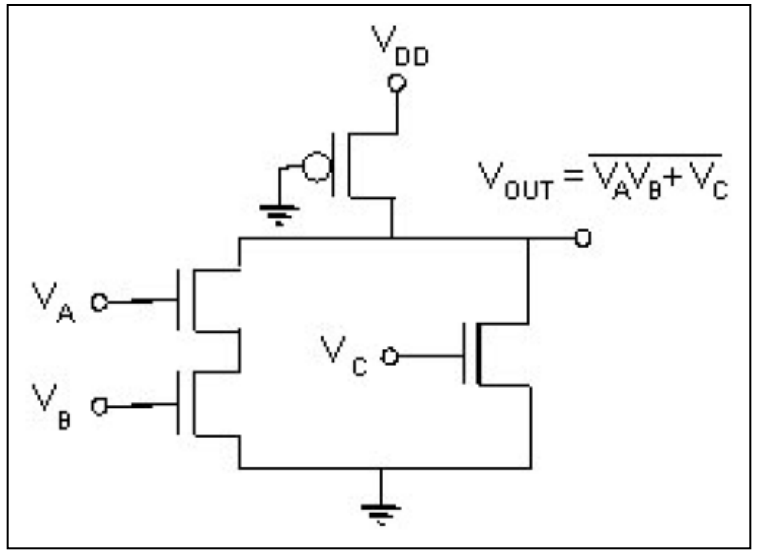
2-input NAND Logic

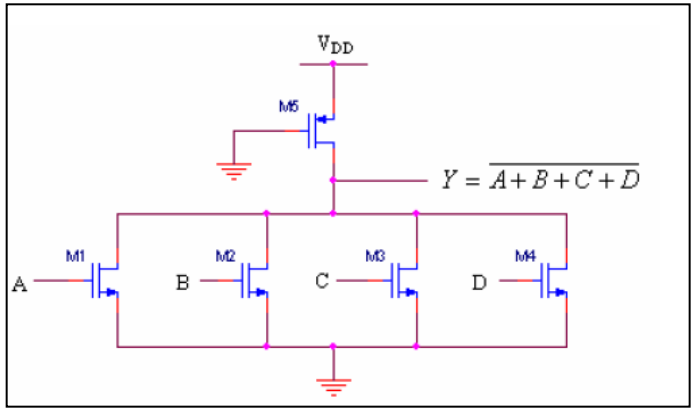
$$V_{OUT} = \overline{V_A V_B + V_C}$$

$$Y = \overline{A + B + C + D}$$

$$Z = \overline{A.(B+C)+(D.E)}$$

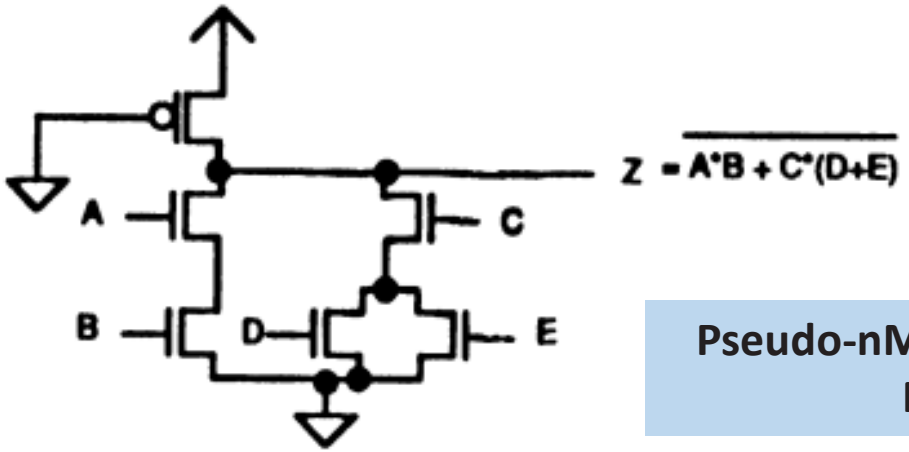






Psuedo-NMOS

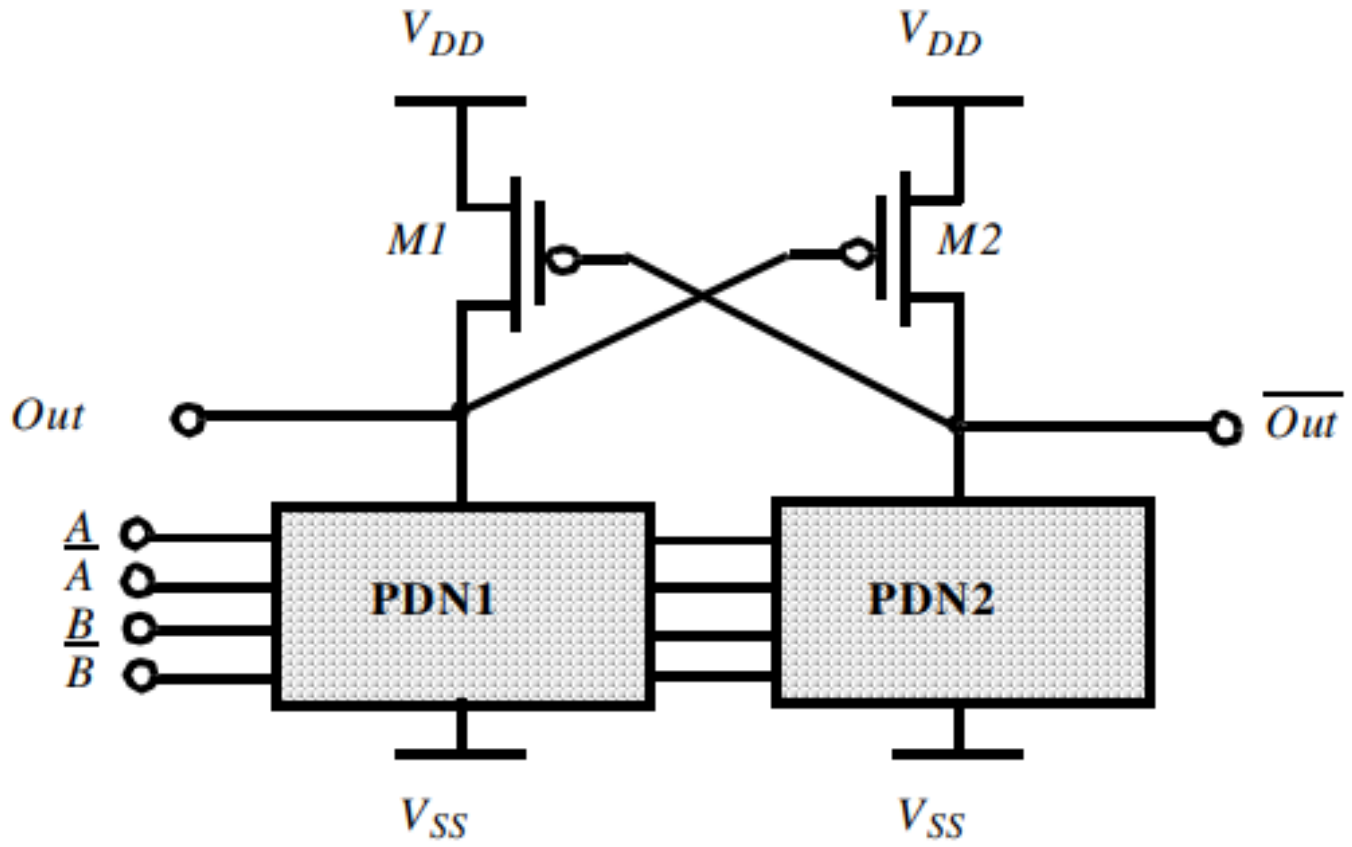
- *Disadvantages of previous circuit* :
  - Almost twice as many transistors as equivalent NMOS implementation.
  - If there are too many series transistors in the tree, switching speed is reduced.
- Try a pseudo NMOS circuit:-



Pseudo-nMOS worked well for wide NOR structures

- The pull-up p-channel transistor is always conducting.
  - *Disadvantages*: high d.c. dissipation & slow rise time.

Dual Cascode Voltage Switch Logic (DCVSL)



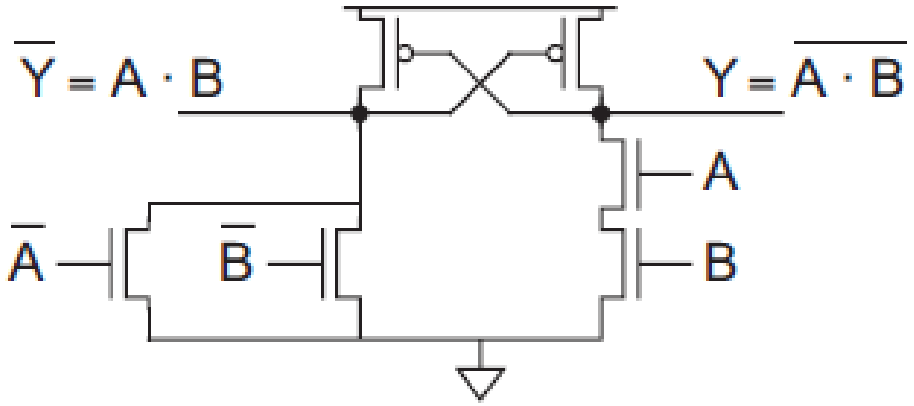
Dual Cascode Voltage Switch Logic (DCVSL)

Dual Cascode Voltage Switch Logic (DCVSL)

- Comprises benefits of ratioed circuits without the static power consumption.
- It uses both true and complementary input signals and computes both true and complementary outputs using a pair of nMOS pulldown networks.
- For any given input pattern, one of the pulldown networks will be ON and the other OFF.
- CVSL has a potential **speed advantage** because all of the logic is performed with nMOS transistors

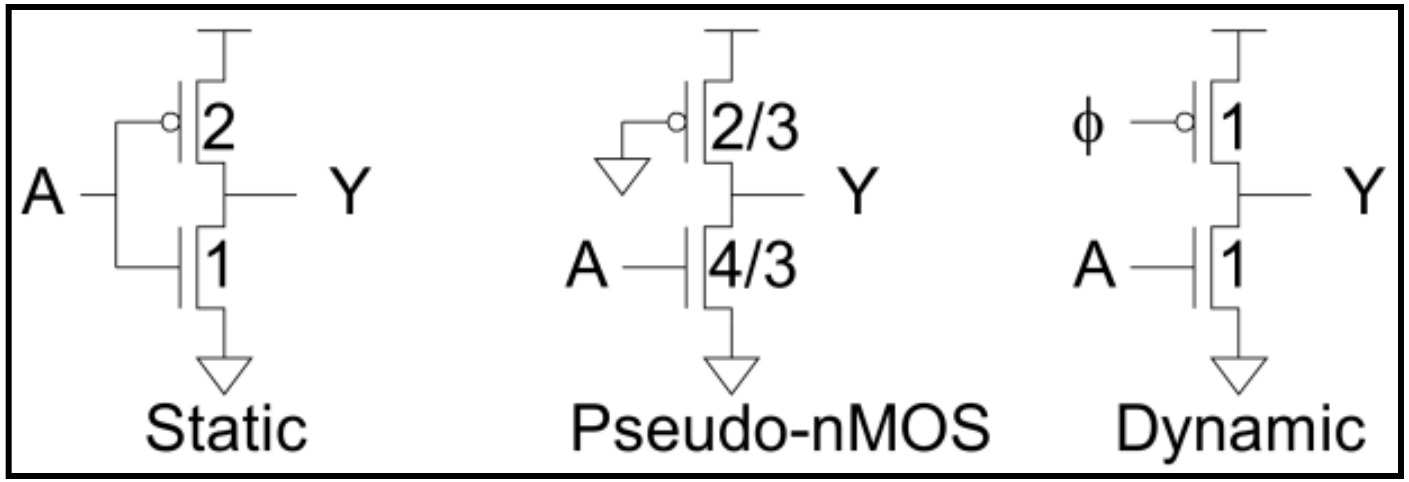
Dual Cascode Voltage Switch Logic (DCVSL)

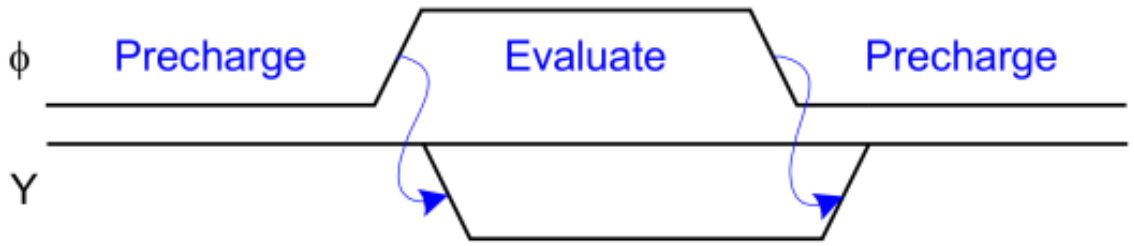
Implement AND-NAND gate using DCVSL



Dynamic Logic

- Dynamic gates use a clocked pMOS pullup
- Two modes of operation: precharge and evaluate



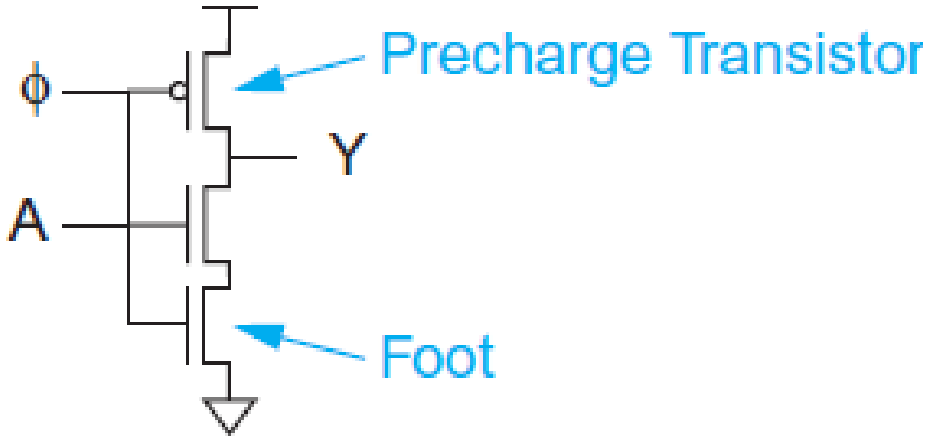


What if input *A* is 1 during precharge?

Dynamic Logic

- When the input cannot be guaranteed to be 0 during precharge, an extra clocked evaluation transistor can be added to the bottom of the nMOS stack to avoid contention.

No Inverting Input

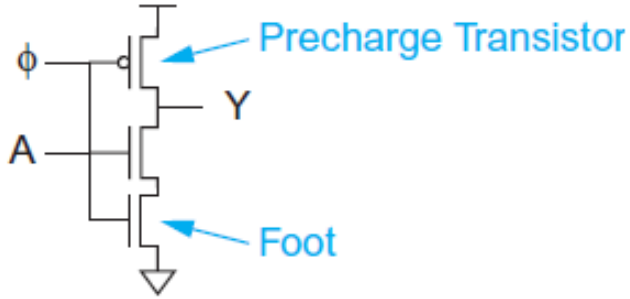


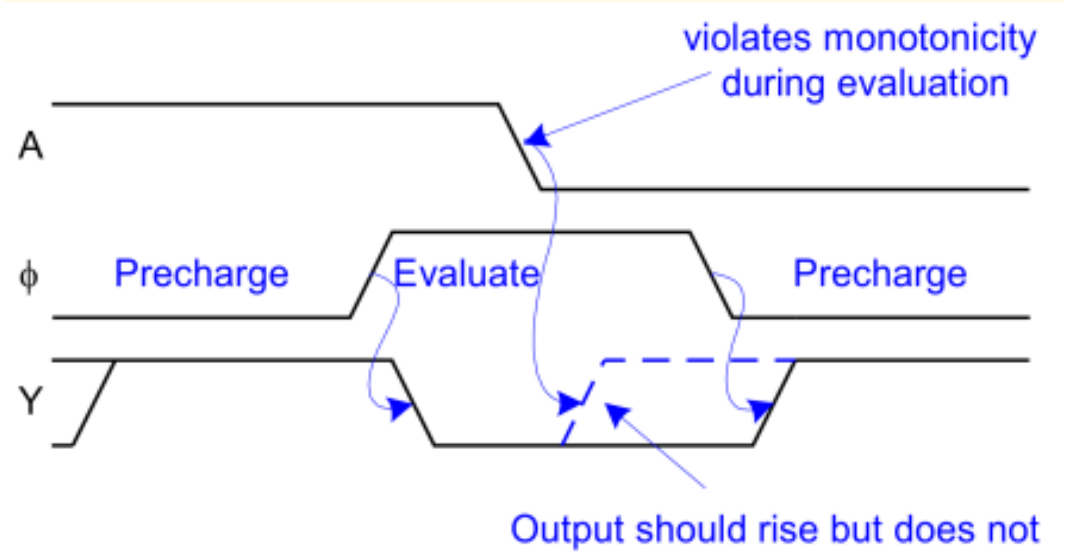
This circuit is dynamic because during evaluation, the output high level at Y is maintained by the stray capacitance at the output node. If Ø stays high (i.e. evaluation period) for a long time, Y may eventually discharge to a low logic level.

Dynamic Logic - Monotonicity

Dynamic gates require monotonically rising inputs during evaluation:

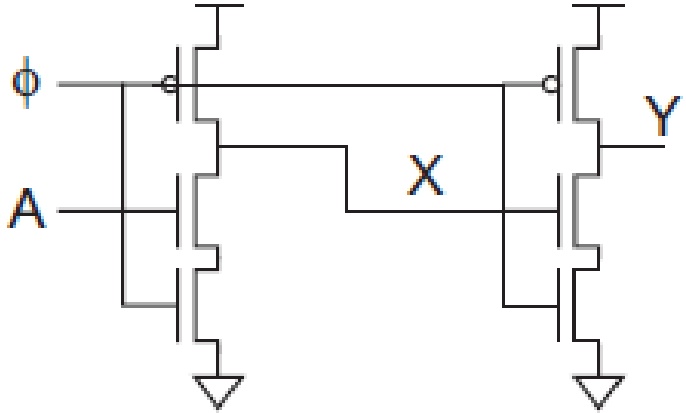
$0 \rightarrow 0$
$0 \rightarrow 1$
$1 \rightarrow 1$
But not $1 \rightarrow 0$

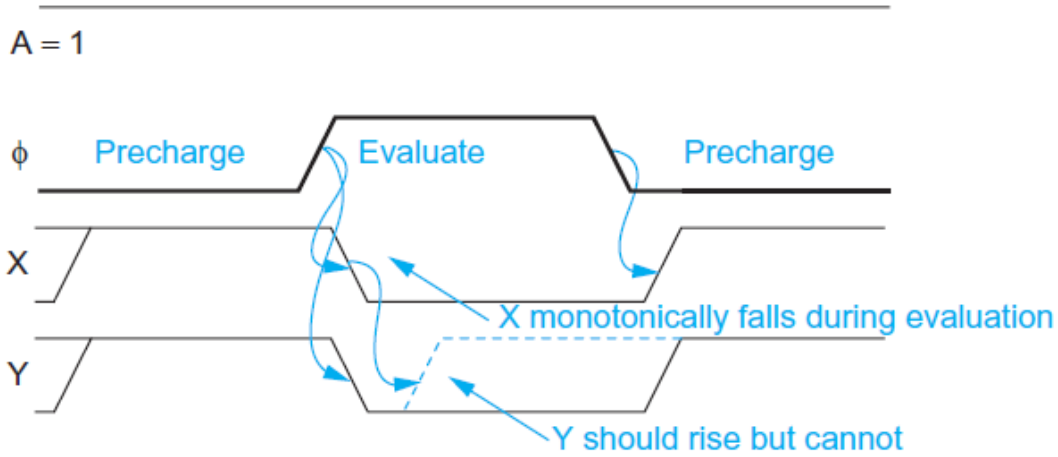




Dynamic Logic - Monotonicity

Even $1 \rightarrow 1$ also create s problem in series Gates:





Dynamic gates sharing the same clock cannot be directly connected.

This problem is often overcome with domino logic

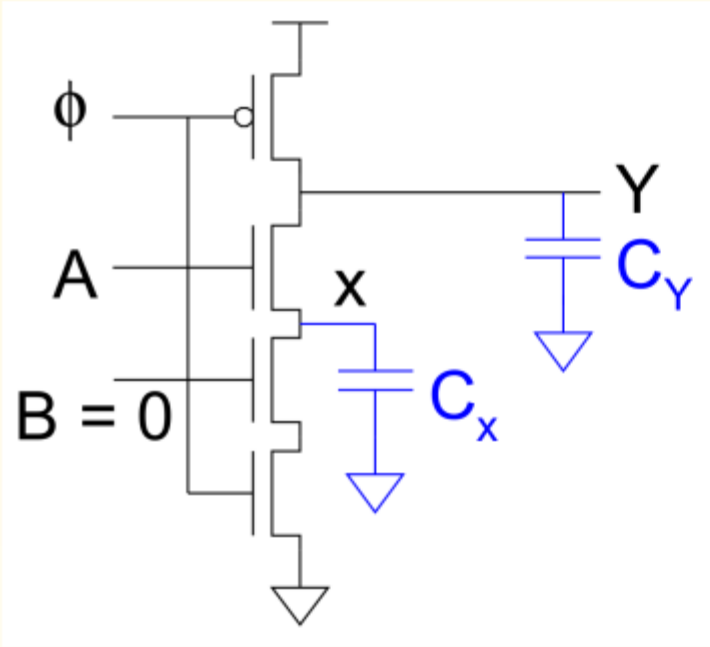
Cascading Problem in Dynamic CMOS Logic

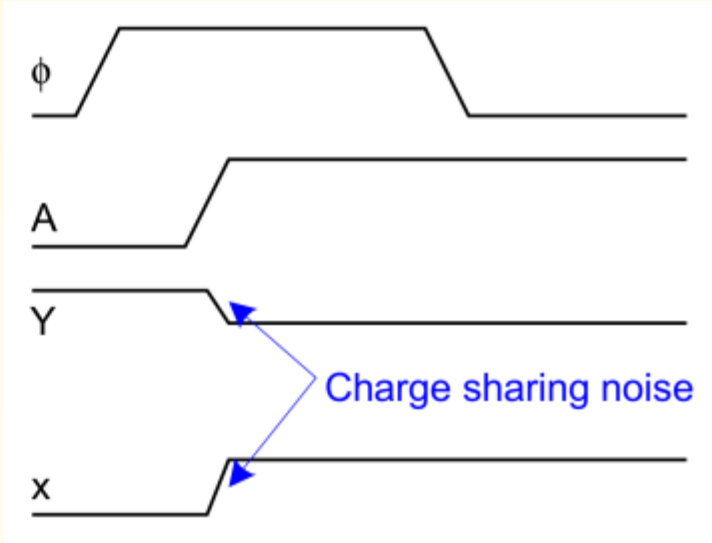
- If several stages of the previous CMOS dynamic logic circuit are cascaded together using the same clock ϕ, a problem in evaluation involving a built-in "race condition" will exist
- Consider the two-stage dynamic logic circuit below:
  - During **pre-charge**, both Vout1 and Vout2 are pre-charged to Vdd
  - When ϕ goes high to begin **evaluation**, all inputs at stage 1 require some finite time to resolve, but during this time, the charge may erroneously be discharged from Vout2
    - e.g. assume that eventually the 1st stage NMOS logic tree conducts and fully discharges Vout1, but since all the inputs to the N-tree all not immediately resolved, it takes some time for the N-tree to discharge Vout1 to GND finally.
    - If, during this time delay, the 2nd stage has the input condition shown with the bottom NMOS transistor gate at logic 1, then Vout2 will start to fall and discharge its load capacitance until Vout1 finally evaluates and turns off the top series NMOS transistor in stage 2
  - The result is an error in the output of the 2nd stage Vout2

Dynamic Logic

Charge Sharing

- Dynamic gates suffer from charge sharing



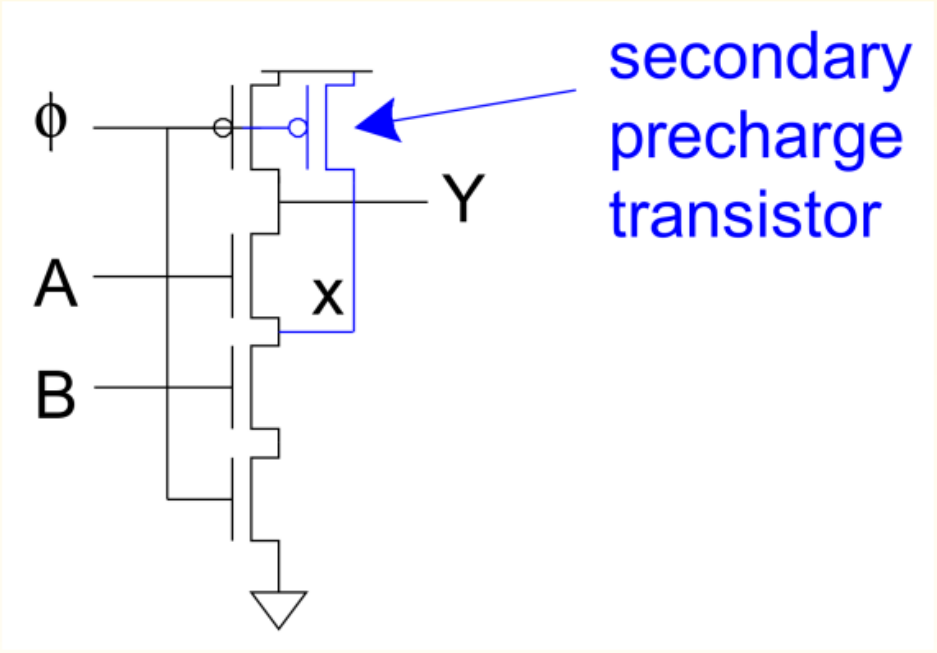


$$V_x = V_y = \frac{C_y}{C_x + C_y} V_{DD}$$

Dynamic Logic

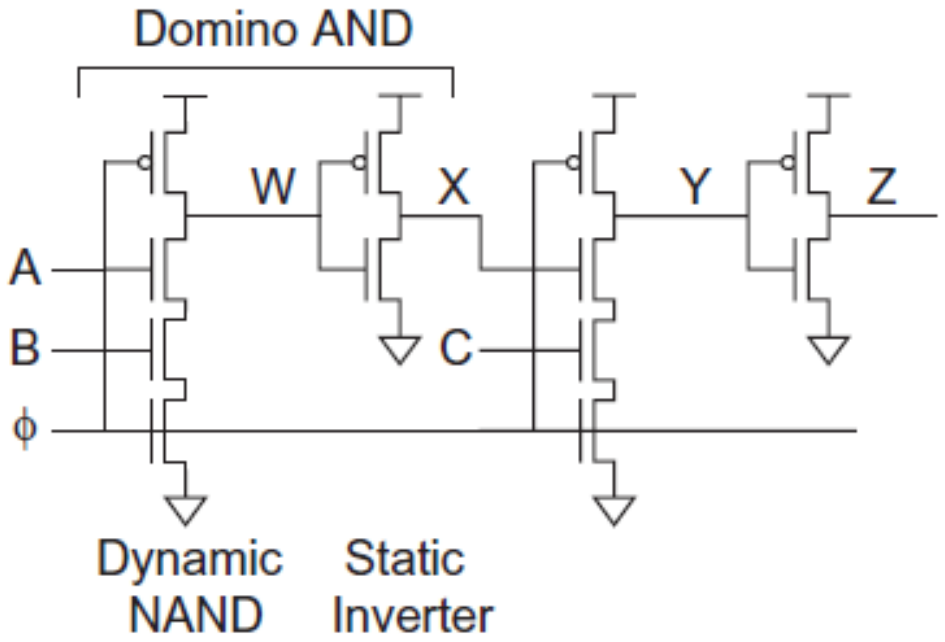
Secondary Precharge

- Solution: add secondary precharge transistors
  - Typically need to precharge every other node
- Big load capacitance on Y helps as well

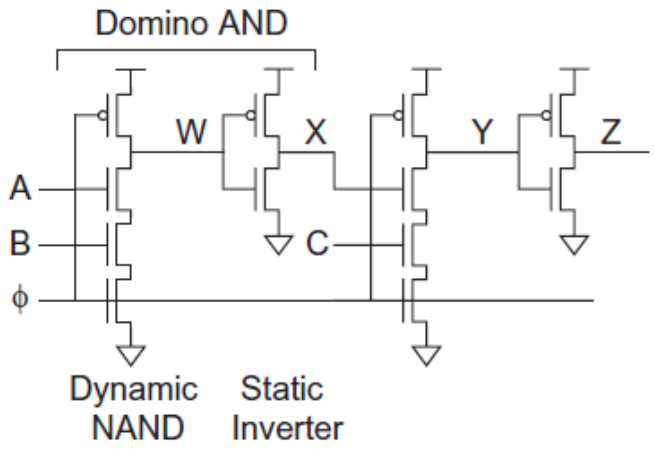


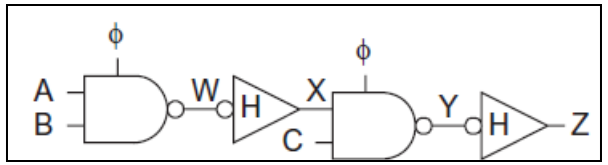
Dynamic Logic - Domino

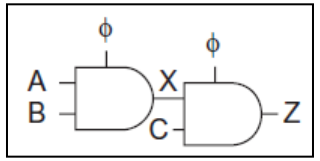
- Follow dynamic stage with inverting static gate.
- Dynamic/static pair is called domino gate Produces monotonic outputs.

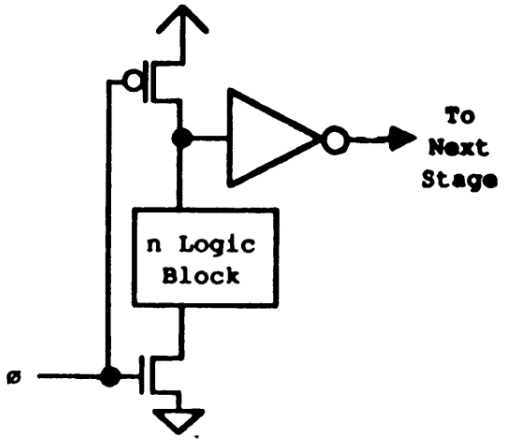


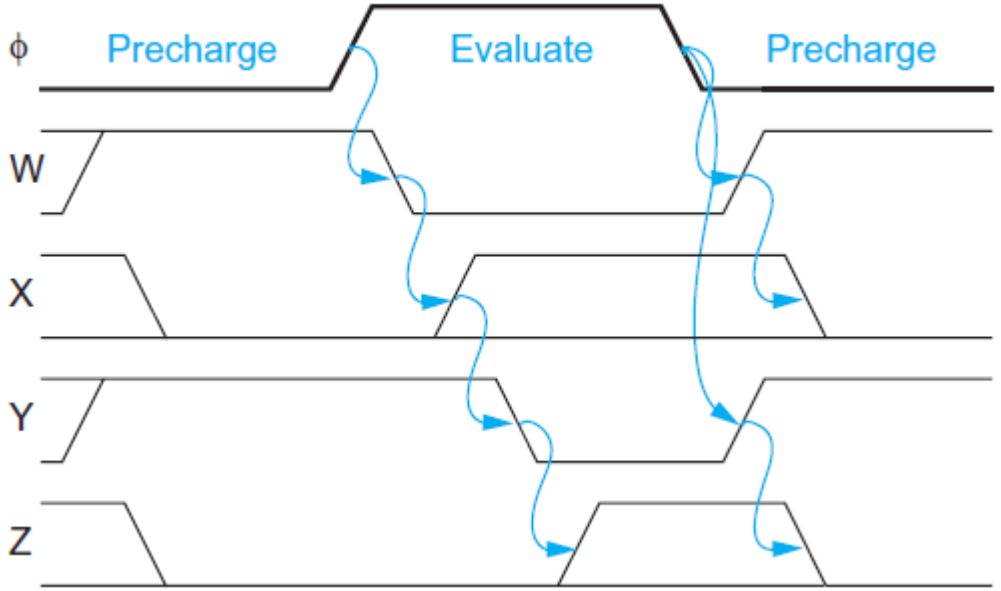
Dynamic Logic - Domino











Precharge occurs in parallel, but evaluation occurs sequentially

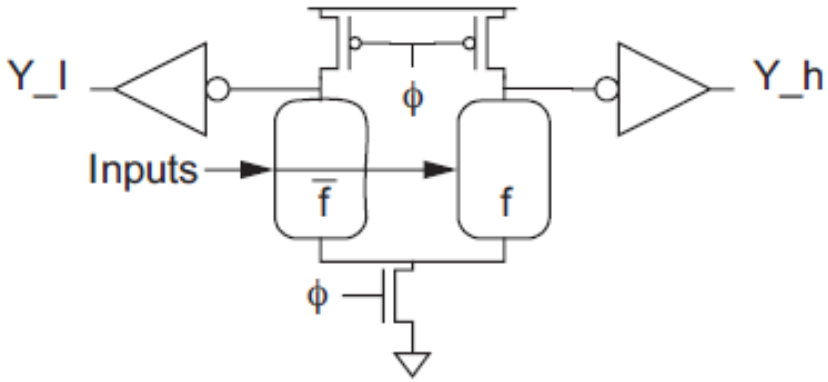
Dynamic Logic - Domino

Disadvantages of domino logic:-

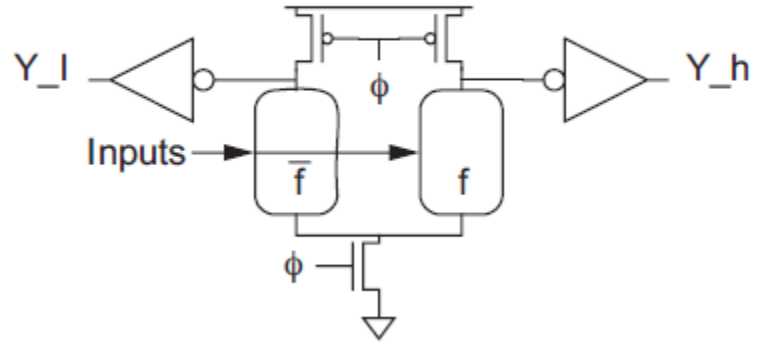
- Only non-inverting logic is possible, i.e. AND, OR but not NAND, NOR, or XOR.
- Each gate needs an inverter; hence more transistors.
- Suffer from charge sharing effect (considered later)

Dual-rail domino solves this problem

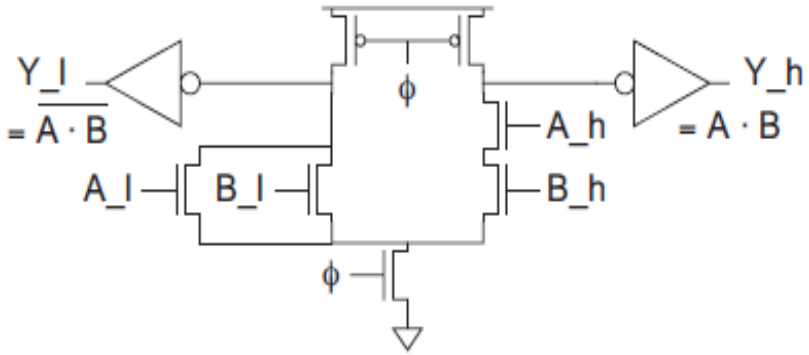
- Takes true and complementary inputs.
- Produces true and complementary outputs



Dynamic Logic – Dual rail Domino



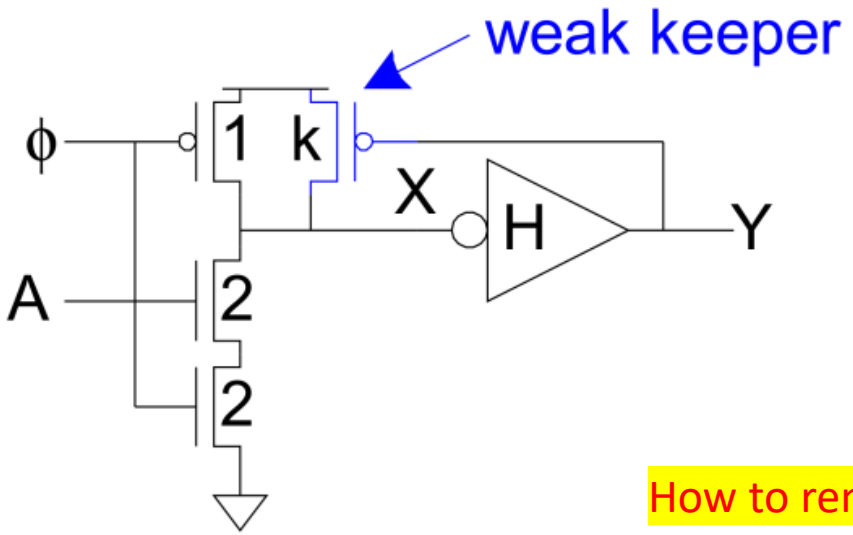
sig_h	*sig_l*	Meaning
0	0	Precharged
0	1	'0'
1	0	'1'
1	1	Invalid



AND/NAND Logic

Domino Logic

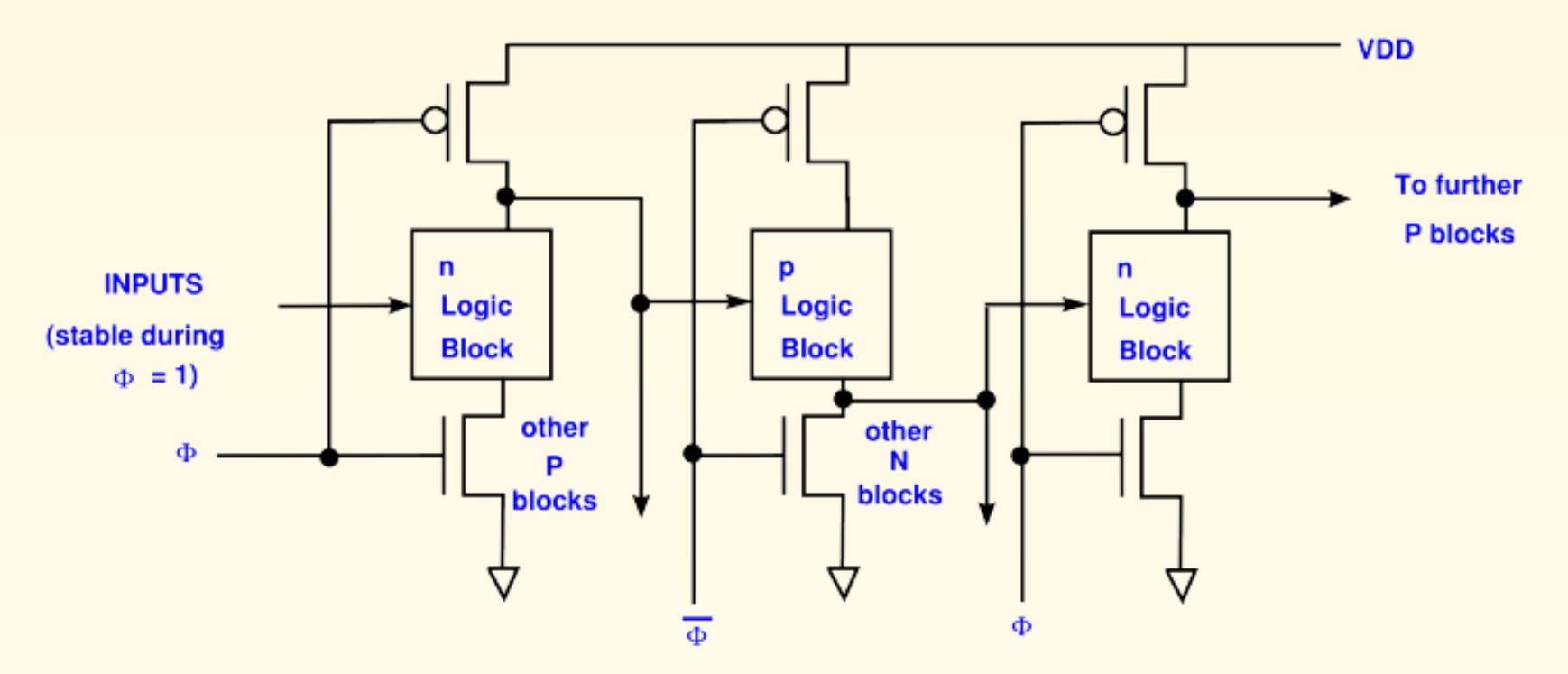
- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{off} \neq 0$)
  - Dynamic value will leak away over time
  - Formerly milliseconds, now nanoseconds!
- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation
- Leakage Power!



How to remove the static inverter??

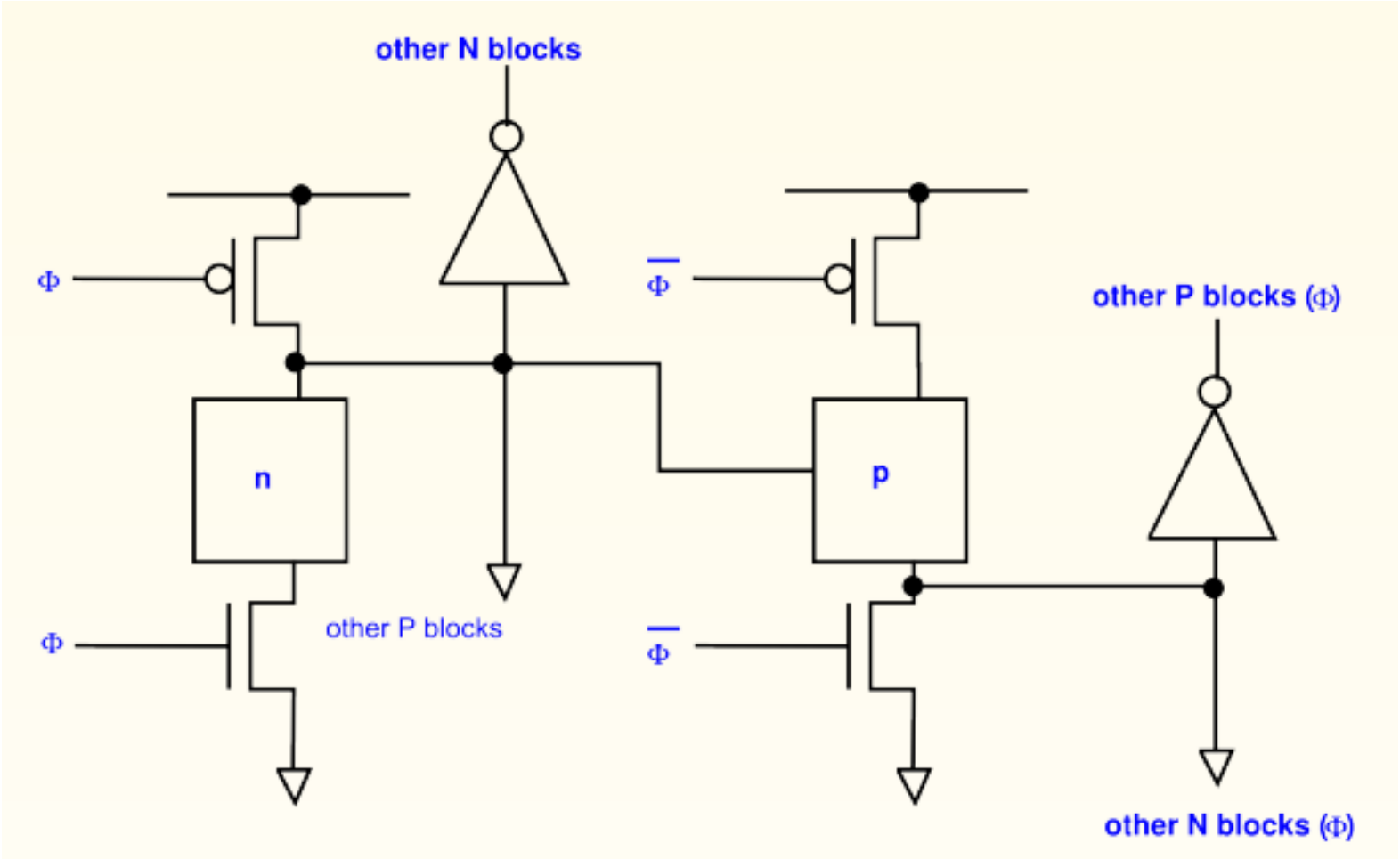
Domino Logic

N-P Domino



Domino Logic

N-P Domino

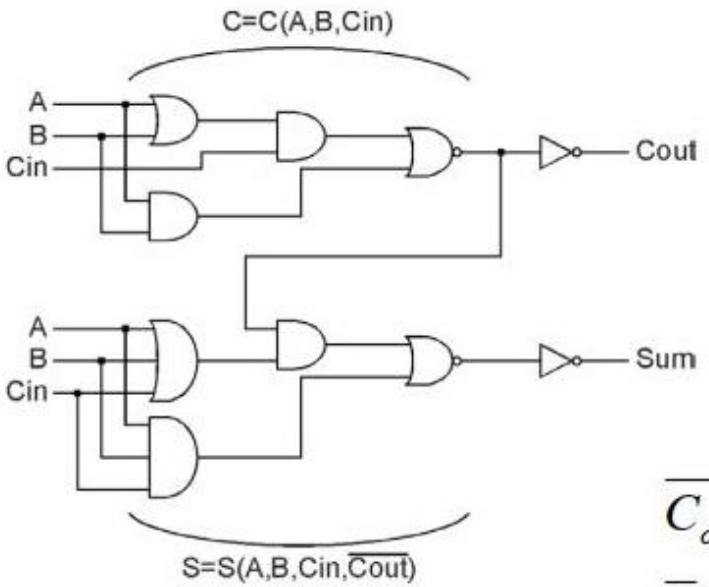


Domino Logic - Summary

- Domino logic is attractive for high-speed circuits
  - 1.5 2x faster than static CMOS
- Many Challenges
  - Monotonicity
  - Leakage
  - Charge sharing
  - Noise
- Used in previous generation high-performance microprocessors and in some recent embedded processors

Example

Implement a Full Adder Using Domino Logic



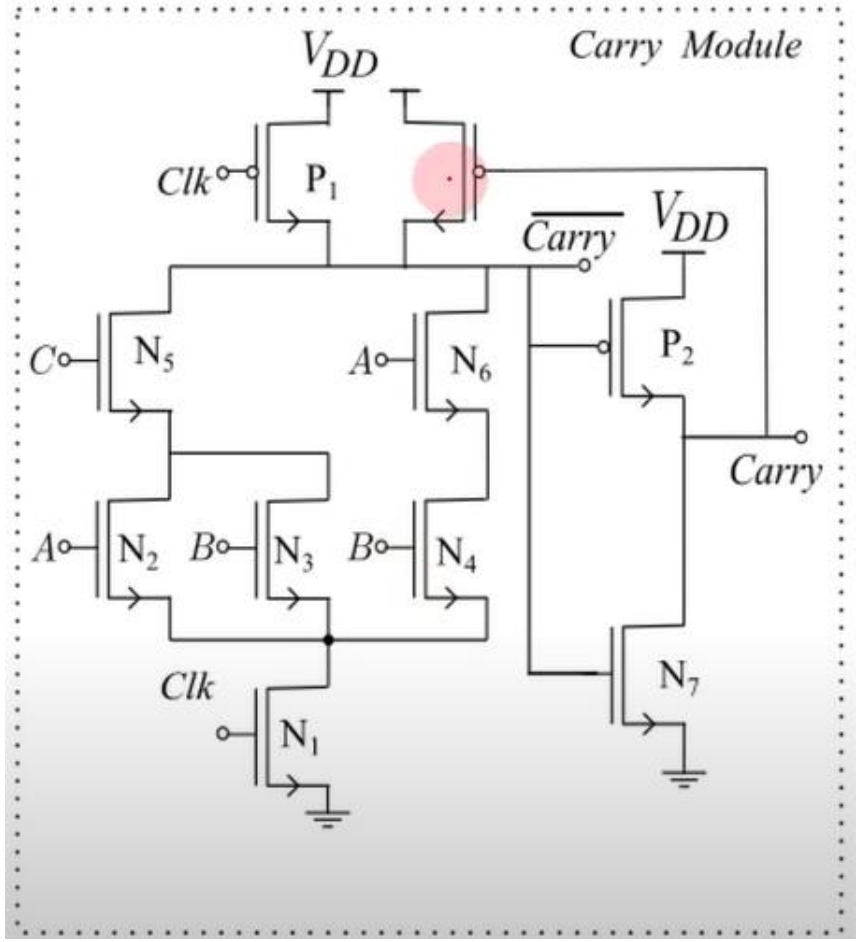
Full Adder

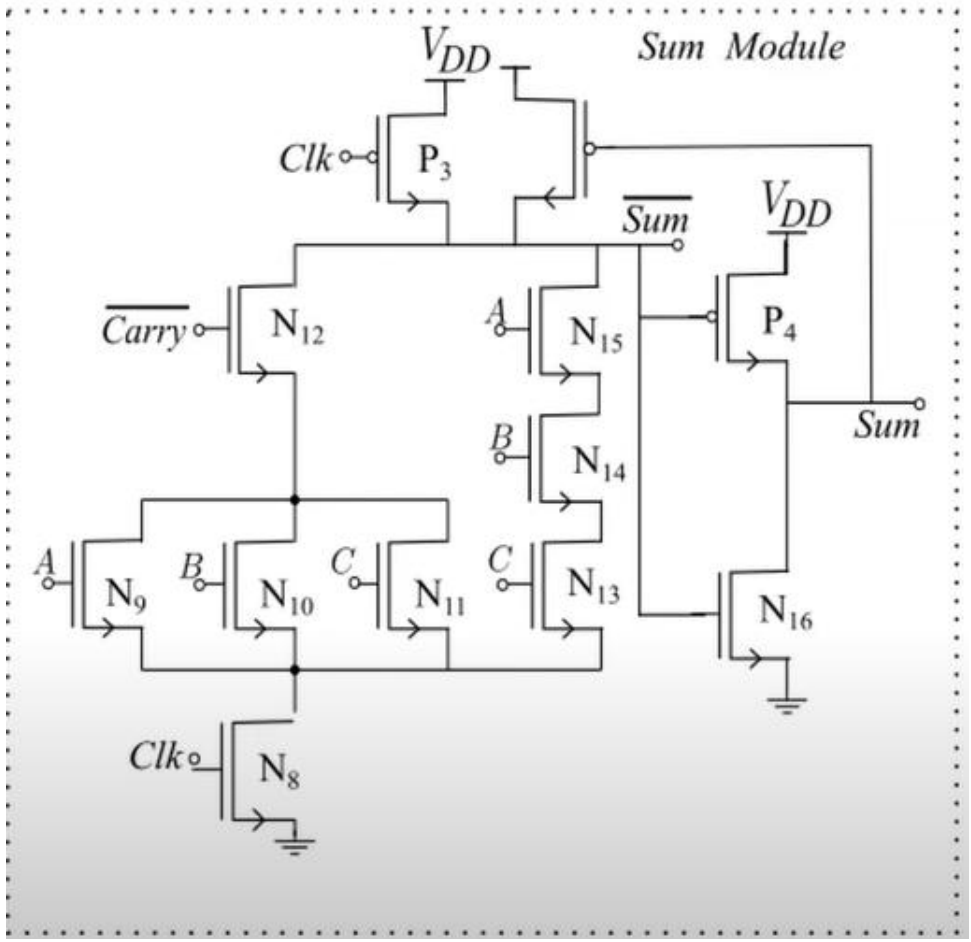
$$\overline{C_{out}} = \overline{AB + C_{in}(A+B)}$$

$$\overline{S} = \overline{C_{out}}(A+B+C_{in}) + ABC_{in}$$

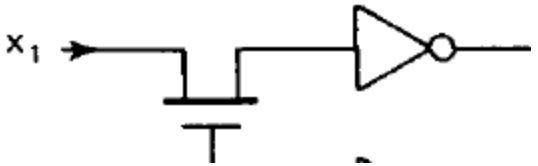
Signal from circuit for $\overline{C_{out}}$ can be re-used for S (→ two stages)

Example

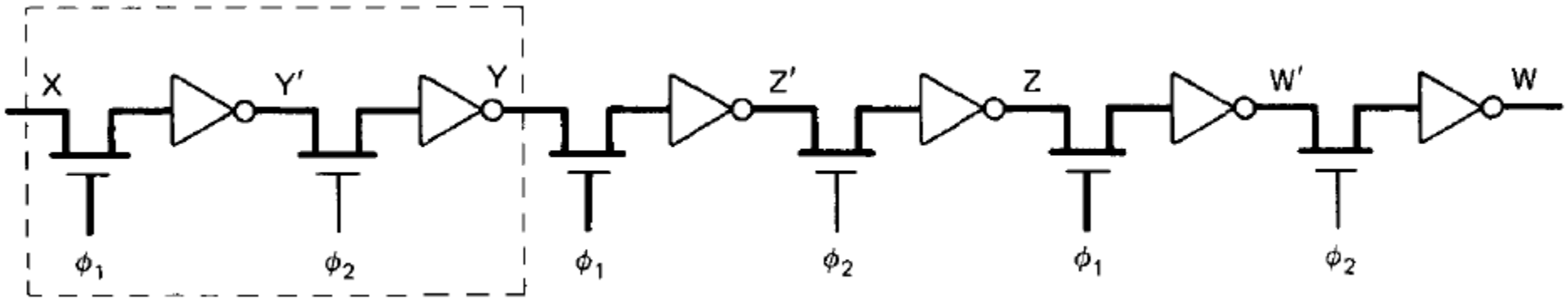

Carry Module
V_{DD}
Clk
P_1
$\overline{Carry}$
V_{DD}
C
N_5
A
N_6
P_2
Carry
A
N_2
B
N_3
B
N_4
Clk
N_1
N_7


Sum Module
V_{DD}
Clk
P_3
$\overline{Sum}$
V_{DD}
$\overline{Carry}$
N_{12}
A
N_{15}
P_4
Sum
B
N_{14}
A
N_9
B
N_{10}
C
N_{11}
C
N_{13}
N_{16}
Clk
N_8

Dynamic Shift register

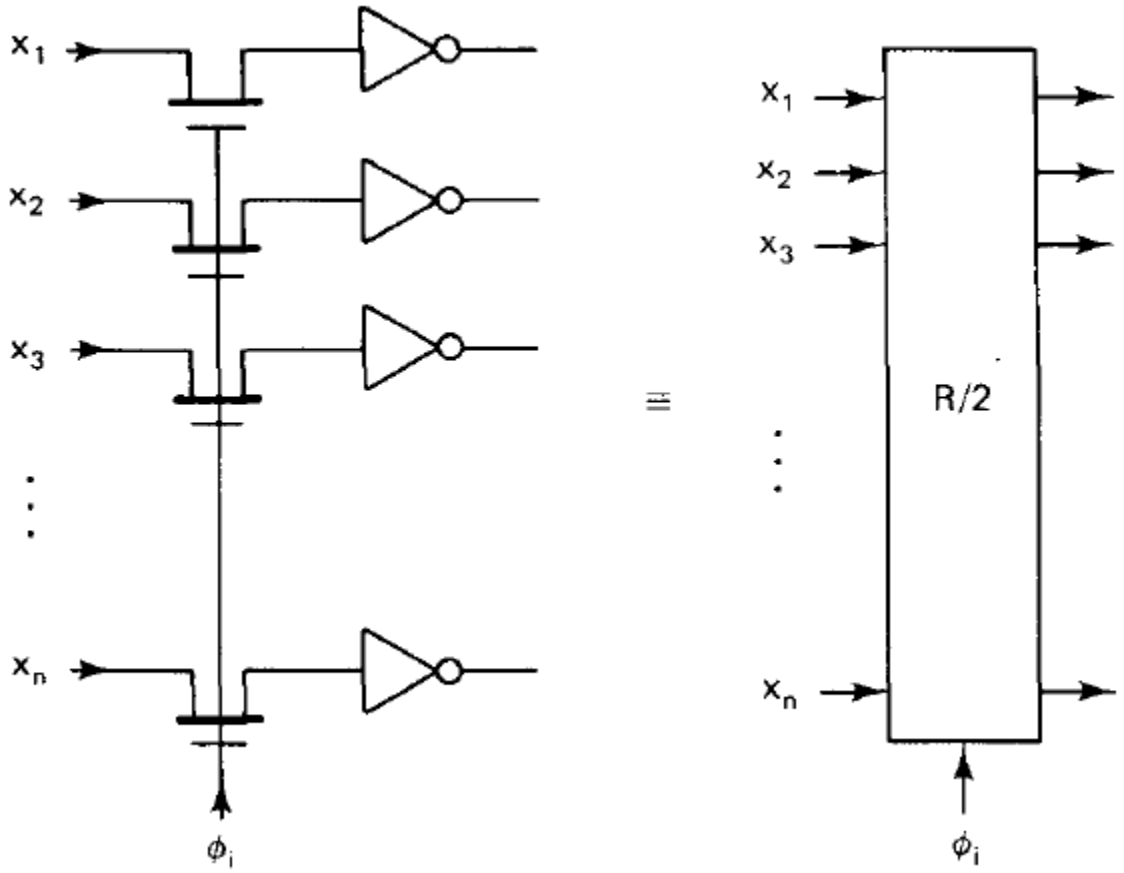


One bit Half-register stage

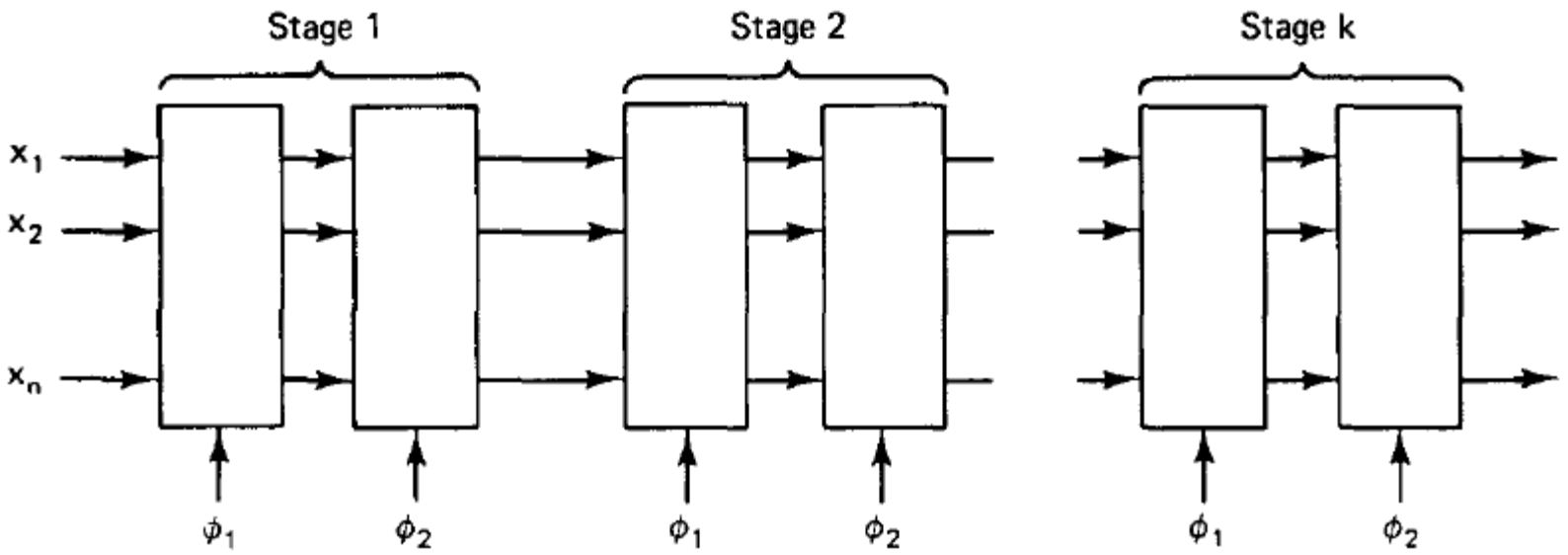


Dynamic shift register for one bit

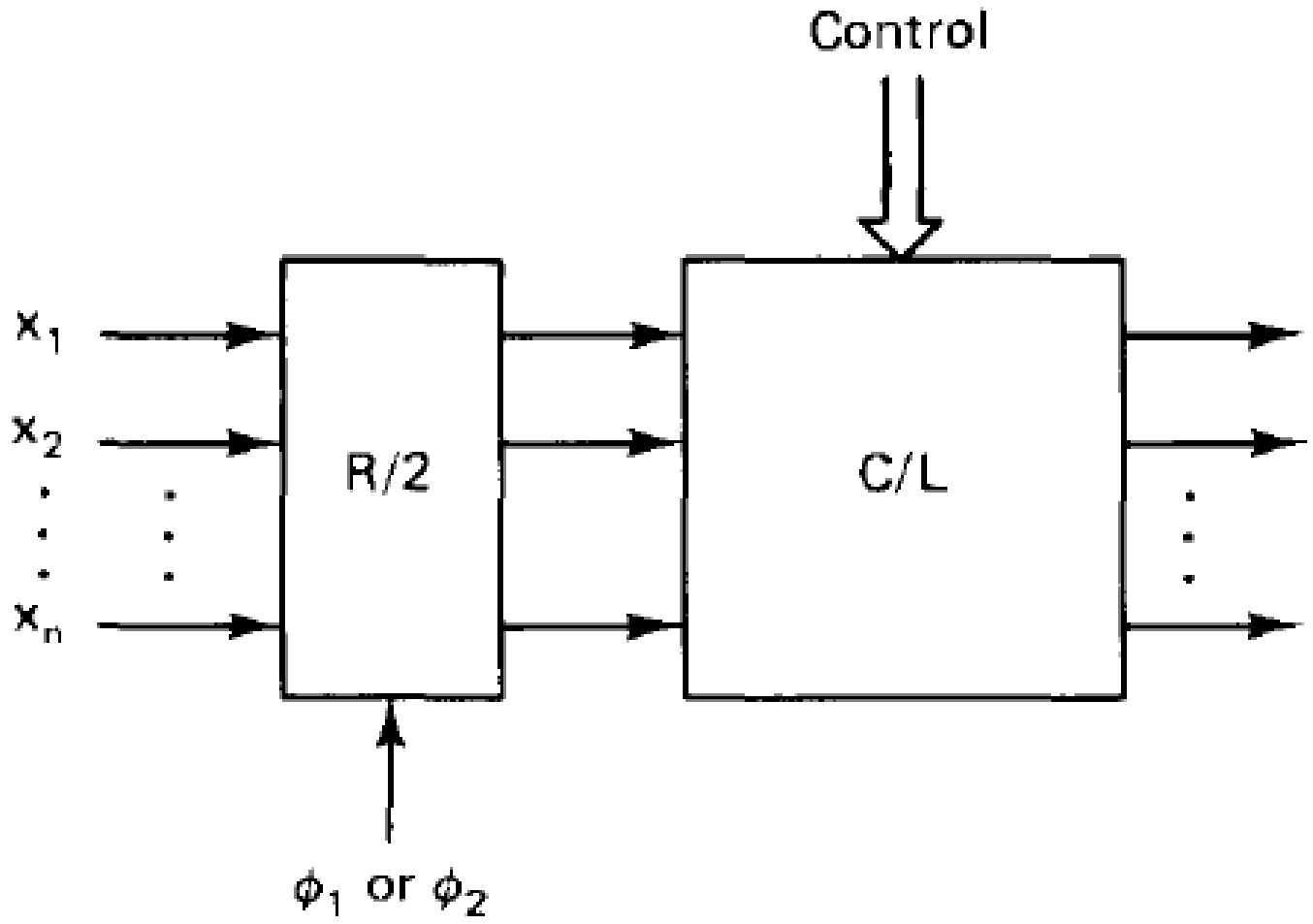
N-Bit Half Register Stage



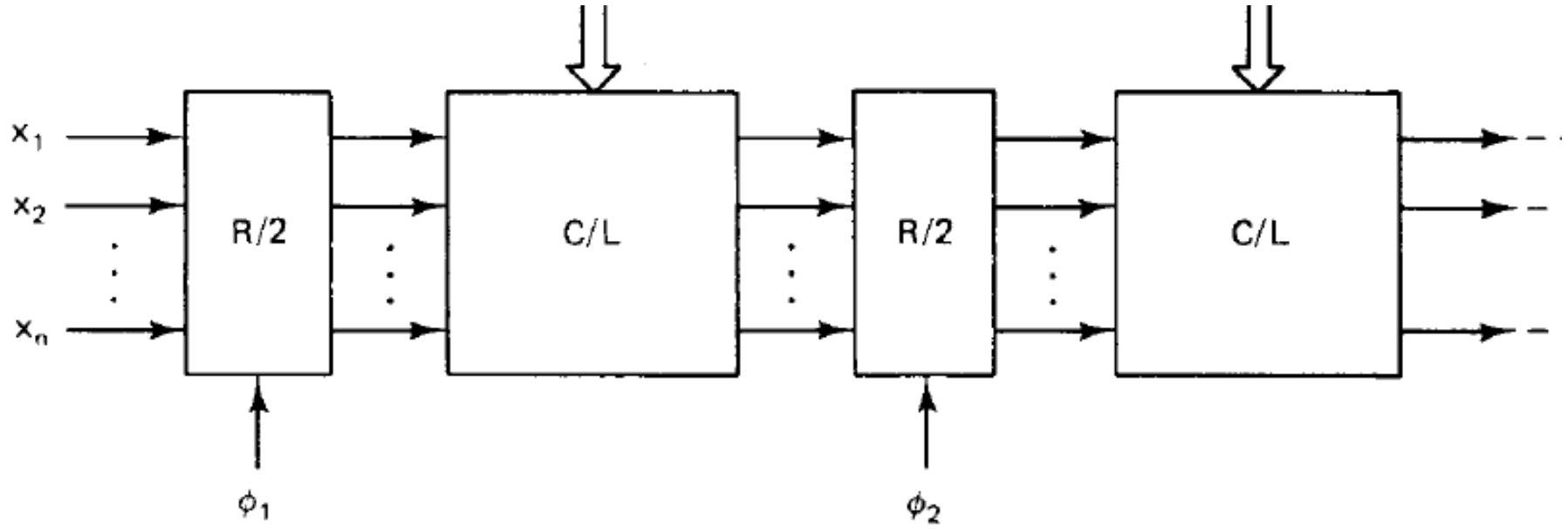
N-bit shift register K-Stage



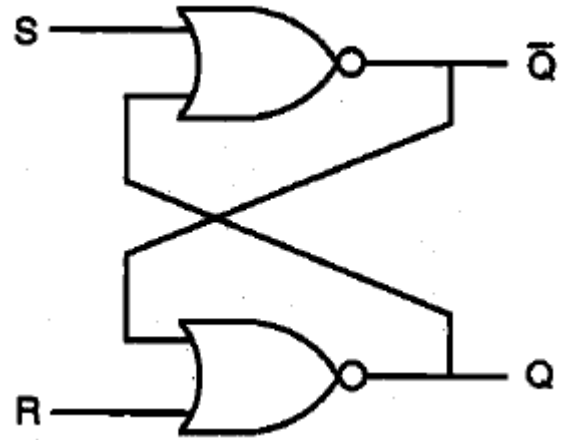
Half-stage of a data path.



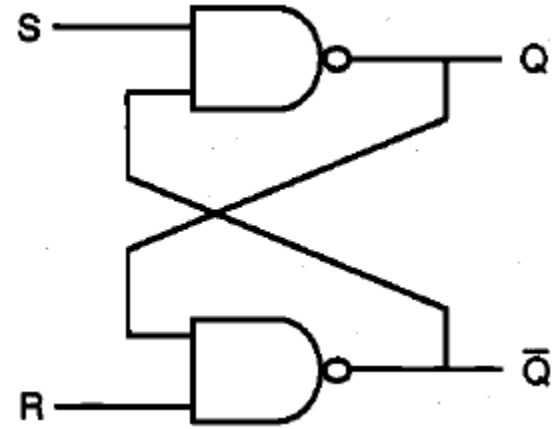
An *n* MOS data path



Static Storage Element

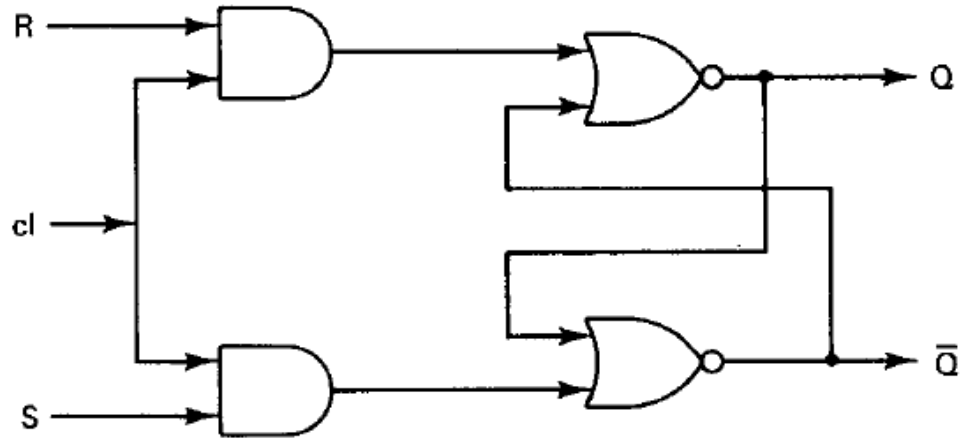


S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0	0	Q_n	$\overline{Q_n}$	hold
1	0	1	0	set
0	1	0	1	reset
1	1	0	0	not allowed



S	R	Q_{n+1}	$\overline{Q_{n+1}}$	Operation
0	0	1	1	not allowed
0	1	1	0	set
1	0	0	1	reset
1	1	Q_n	$\overline{Q_n}$	hold

Static Storage Element



$S(t)$	$R(t)$	$Q(t+1)$	$\overline{Q}(t+1)$
1	0	1	0
0	1	0	1
0	0	$Q(t)$	$\overline{Q}(t)$
1	1	Not	allowed