LOVELY
PROFESSIONAL
UNIVERSITY

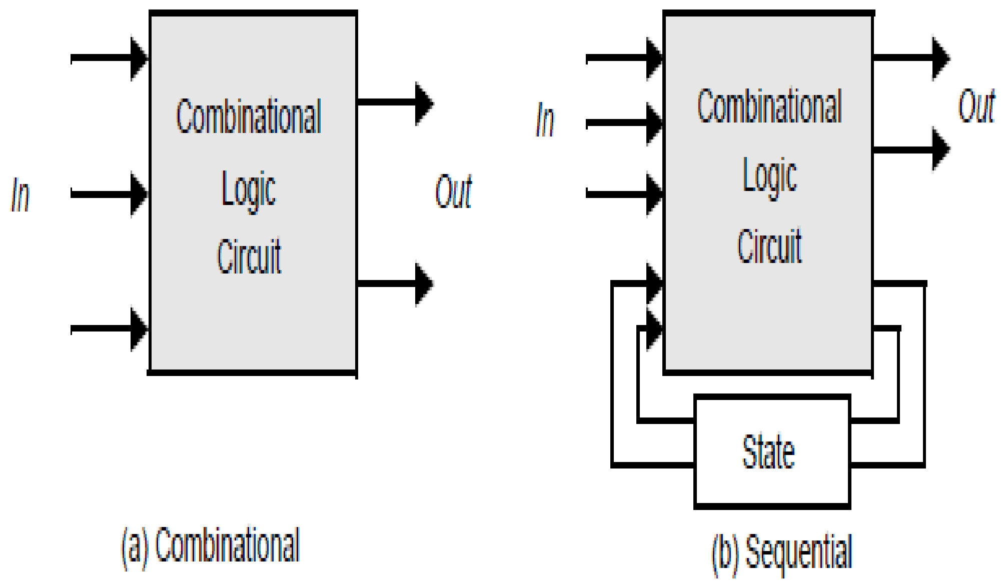
CMOS VLSI Design

ECE318

Unit4: Combinational MOS Logic Circuits

Classification Digital Circuit

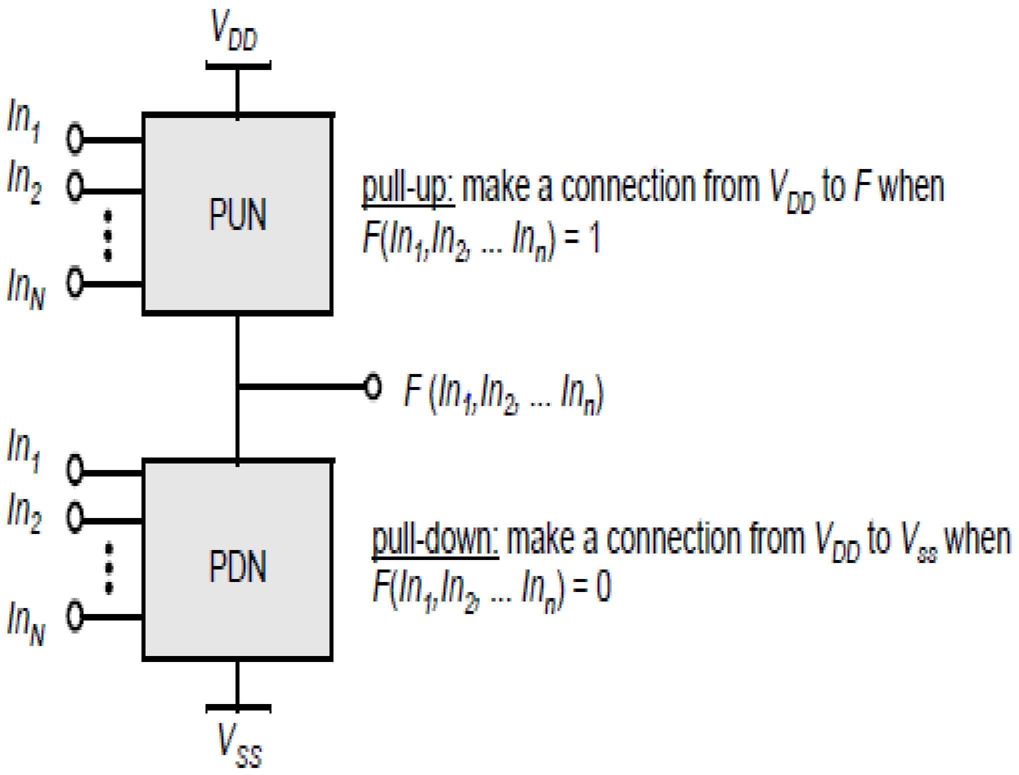
- Combinational and
- The focus is on *combinational logic* (or *non-regenerative*) circuits; this is, circuits that have the property that at any point in time, the output of the circuit is related to its current input signals by some Boolean expression
- Sequential Circuit:
- This is in contrast to another class of circuits, known as *sequential* or *regenerative*, for which the output is not only a function of the current input data, but also of previous values of the input signals



High level classification of logic circuits.

Complementary CMOS

- A static CMOS gate is a combination of two networks, called the *pull-up network* (PUN) and the *pull-down* network (PDN)
- A transistor can be thought of as a switch controlled by its gate signal.
- An NMOS switch is *on* when the controlling signal is high and is *off* when the controlling signal is low.
- A PMOS transistor acts as an inverse switch that is *on* when the controlling signal is low and *off* when the controlling signal is high.



pull-up: make a connection from V_{DD} to F when $F(In_1, In_2, \ldots In_n) = 1$

pull-down: make a connection from V_{DD} to V_{ss} when $F(In_1, In_2, \ldots In_n) = 0$

MCQ

- PDN and PUN network are made by

a) PMOS, NMOS transistors

b) NMOS, PMOS transistors

c) NMOS, NMOS transistors

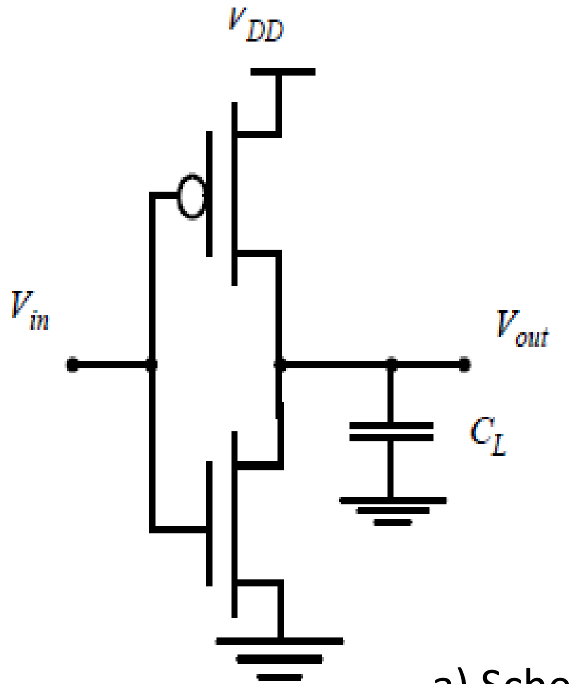
d) PMOS, PMOS transistors

CMOS Inverter

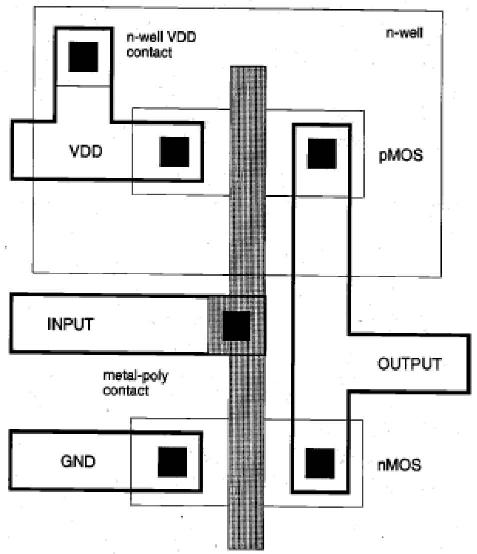
Static CMOS inverter. V_{DD} stands for the supply voltage.

Design metrics:

- *cost,* expressed by the complexity and area
- *integrity and robustness*, expressed by the static (or steady-state) beh
- *performance*, determined by the dynamic (or transient) response
- *energy efficiency,* set by the energy and power consumption



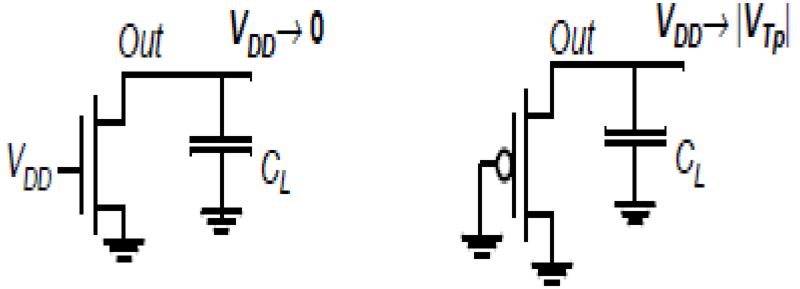
a) Schematic



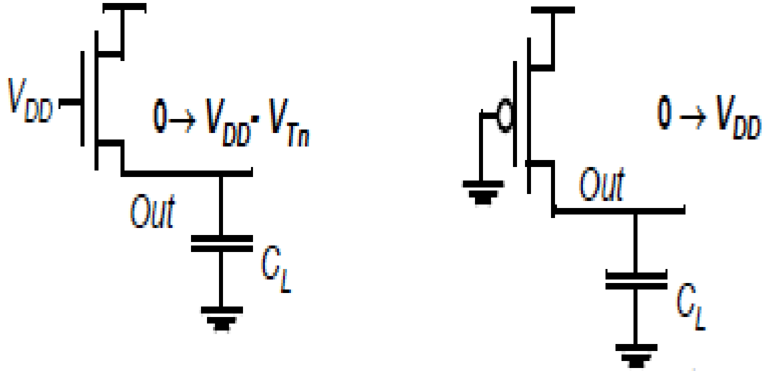
b) Layout

Complementary CMOS

- The PDN is constructed using NMOS devices, while PMOS transistors are used in the PUN.
- The primary reason for this choice is that NMOS transistors produce “strong zeros,” and PMOS devices generate “strong ones”. The output capacitance is initially charged to *VDD*.
- Two possible discharge scenarios are shown. An NMOS device pulls the output all the way down to GND, while a PMOS lowers the output no further than |*VTp*| — the PMOS turns *off* at that point, and stops contributing discharge current. NMOS transistors are hence the preferred devices in the PDN. Similarly,
- Same way, two alternative approaches to charging up a capacitor with the output initially at GND.
- A PMOS switch succeeds in charging the output all the way to *VDD*, while the NMOS device fails to raise the output above *VDD-VTn*. This explains why PMOS transistors are preferentially used in a PUN.

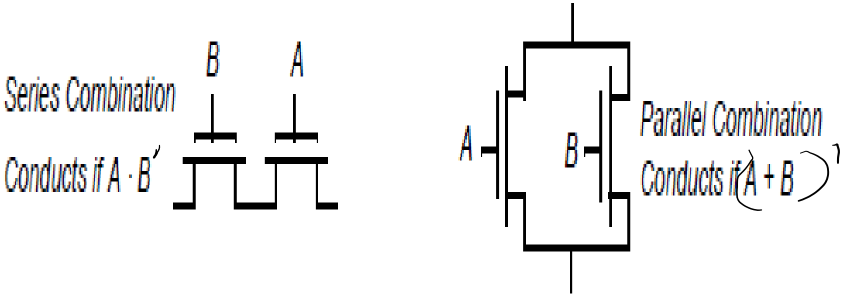


(a) pulling down a node using NMOS and PMOS switches



(b) pulling ~~down~~ up a node using NMOS and PMOS switches

Logic Rule



(a) series

(b) parallel

NMOS Logic Rule

- Using De Morgan's theorems (*A* + *B*)' = *A*'·*B*' and (*A*·*B*)'= (*A*' + *B*'), it can be shown that the pull-up and pull-down networks of a complementary CMOS structure are *dual* networks.
- This means that a parallel connection of transistors in the pull-up network corresponds to a series connection of the corresponding devices in the pull-down and vice versa
- Therefore, to construct a CMOS gate, one of the networks (e.g., PDN) is implemented using combinations of series and parallel devices.
- The other network (i.e., PUN) is obtained using duality principle by walking the hierarchy, replacing series sub-nets with parallel sub-nets, and parallel sub-nets with series sub-nets.
- The complete CMOS gate is constructed by combining the PDN with the PUN.

NMOS connected series in PDN:
F=(A.B)'

NMOS connected parallel in PDN
F=(A+B)'

MCQ

In PDN, when transistors in series, the output node follows

a) AND function

b) OR function

c) NAND function

d) NOR function

2-Input NOR and NAND and Layout

- Figure 7.14 shows a sample layout of a CMOS NOR2 gate, using single layer metal and single-layer polysilicon.
- In this example, the p-type diffusion area for pMOS transistors and the n-type diffusion area for nMOS transistors are aligned in parallel to allow simple routing of the gate signals via two parallel polysilicon lines running vertically.

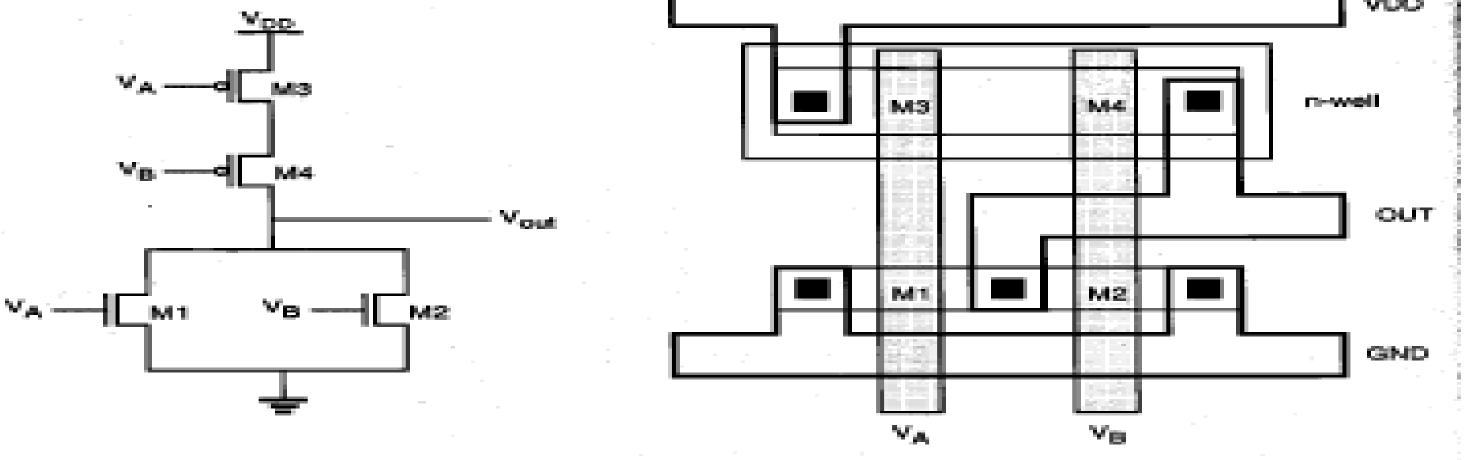
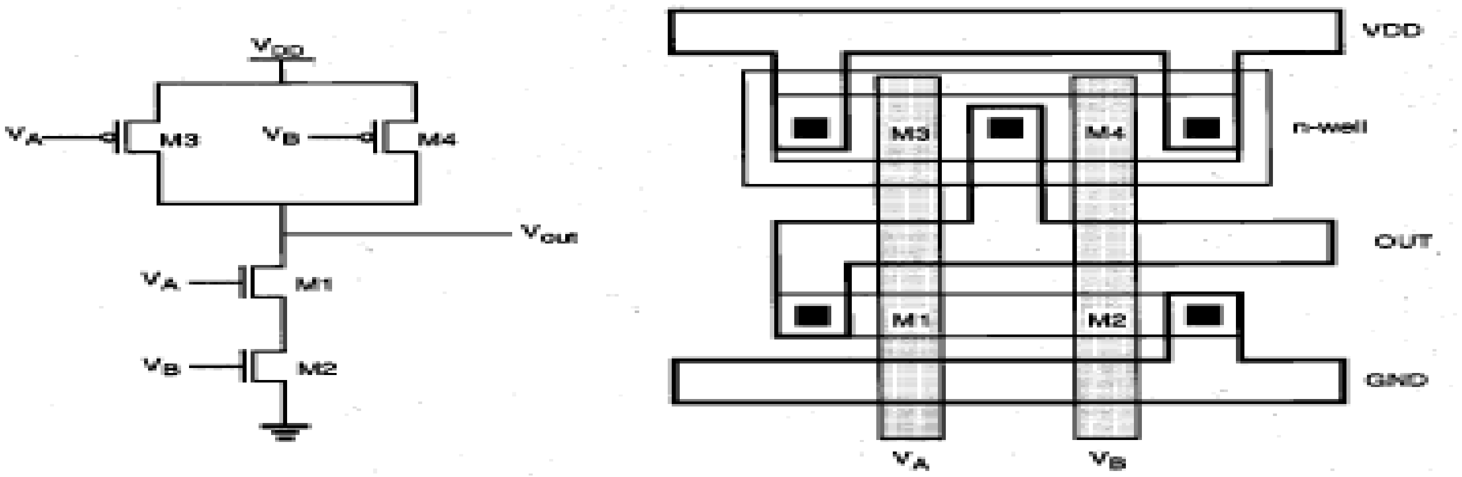
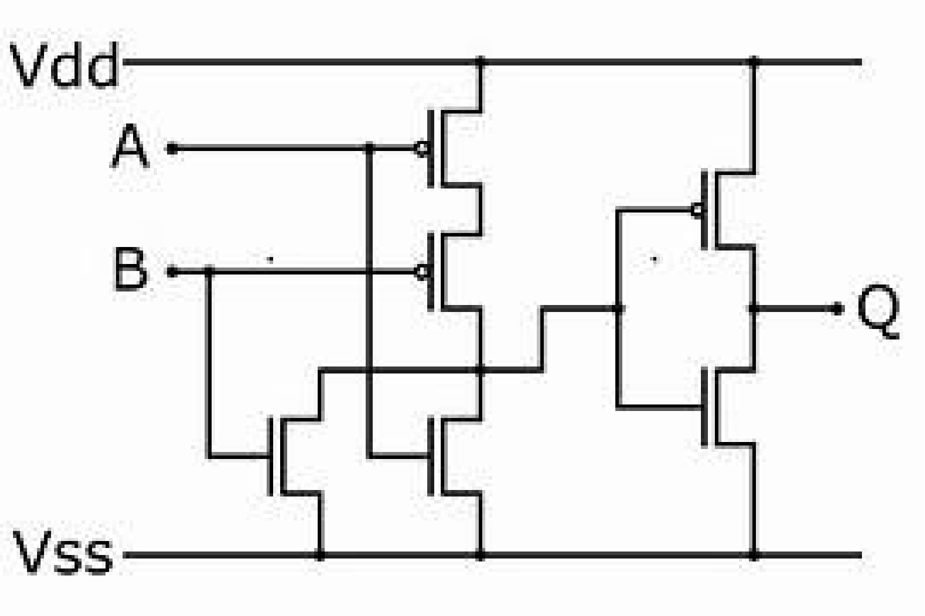
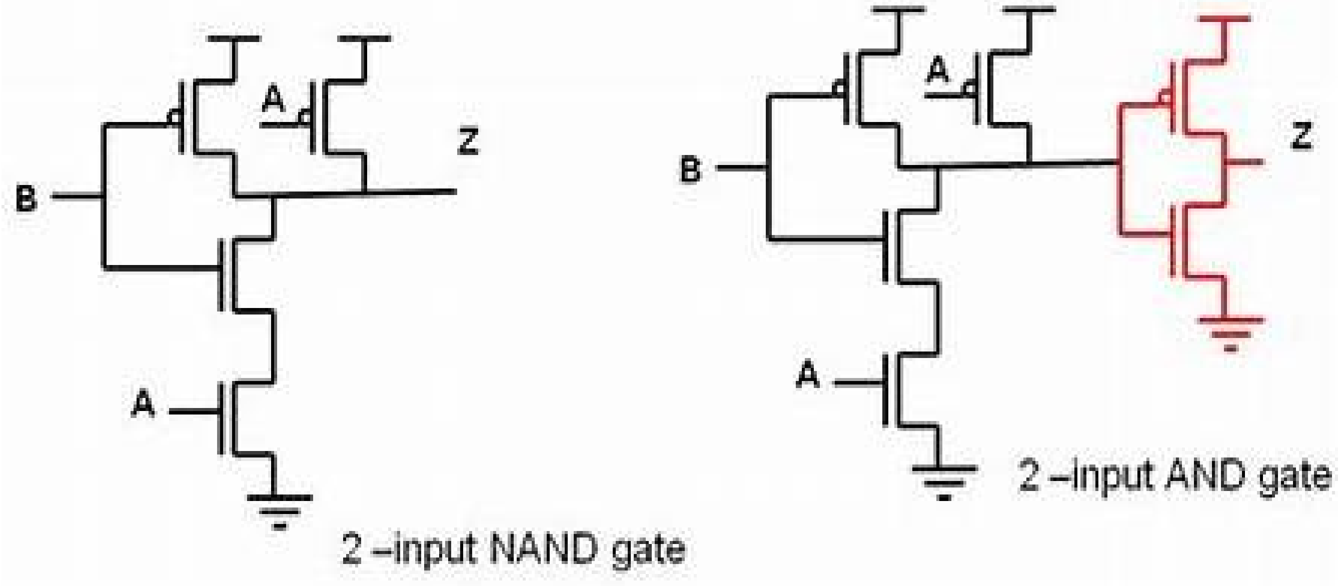


Figure 7.14. Sample layout of the CMOS NOR2 gate.





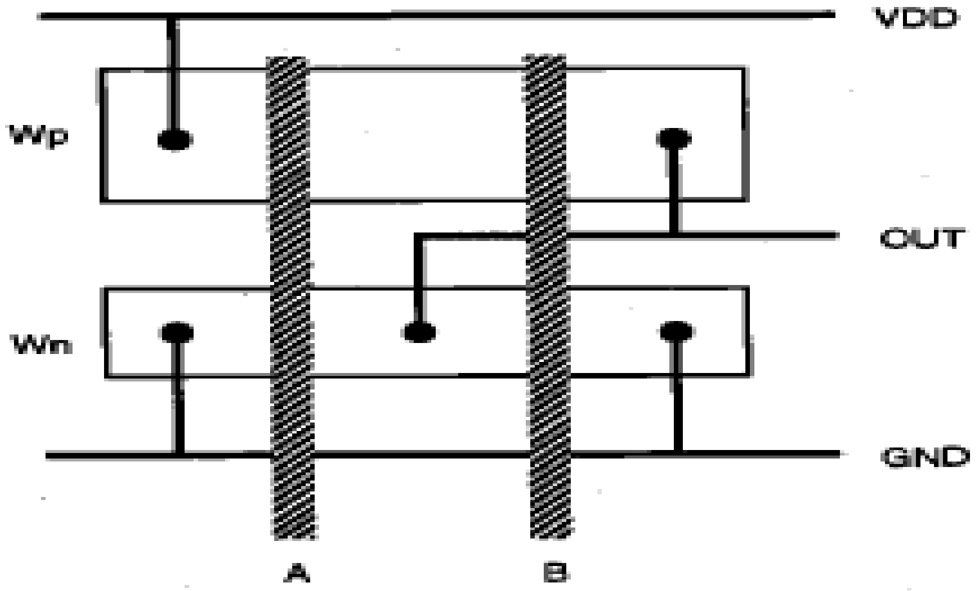
OR Gate using CMOS



NAND and AND gate using CMOS

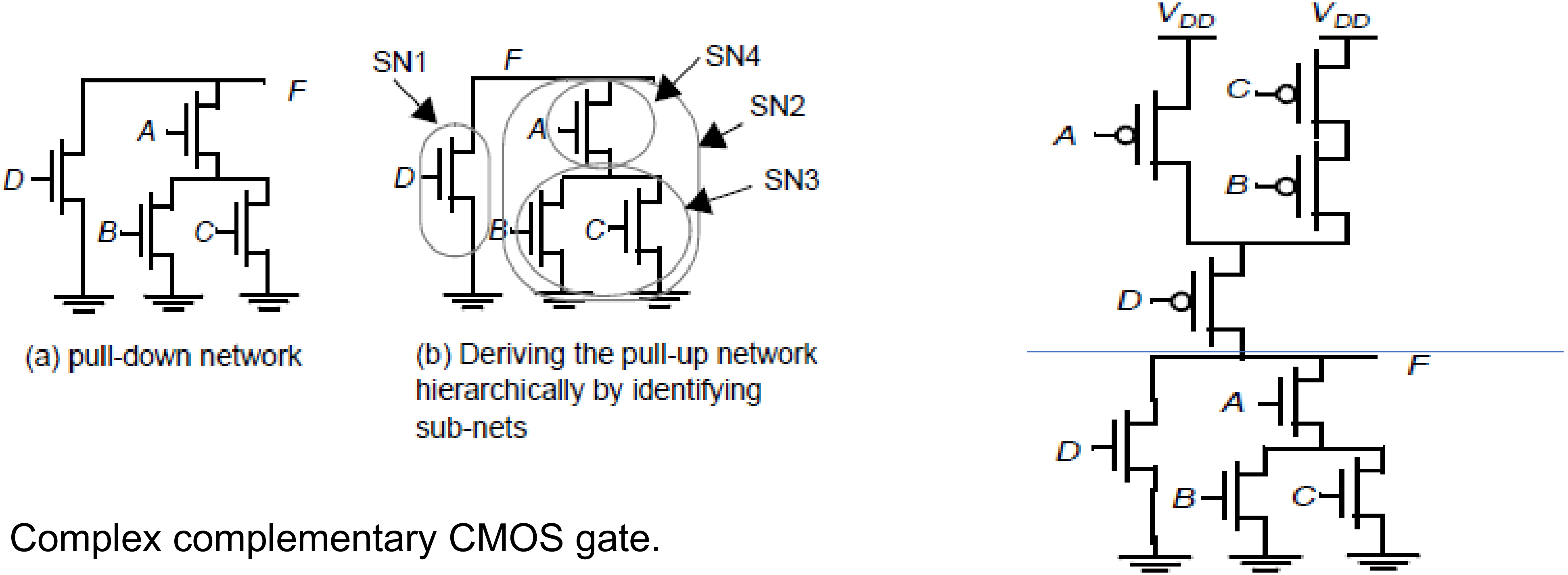
Stick Diagram

- The stick-diagram layout does not carry any information on the actual geometry relations of the individual features, but it conveys valuable information on the relative placement of the transistors and their interconnections.



Synthesis of complex CMOS Gate

- Using complementary CMOS logic, consider the synthesis of a complex CMOS gate whose function is
- *F* = (*D* + *A*· (*B* +*C*))′



(a) pull-down network

(b) Deriving the pull-up network hierarchically by identifying sub-nets

(c) complete gate

Complex complementary CMOS gate.

Problem 1

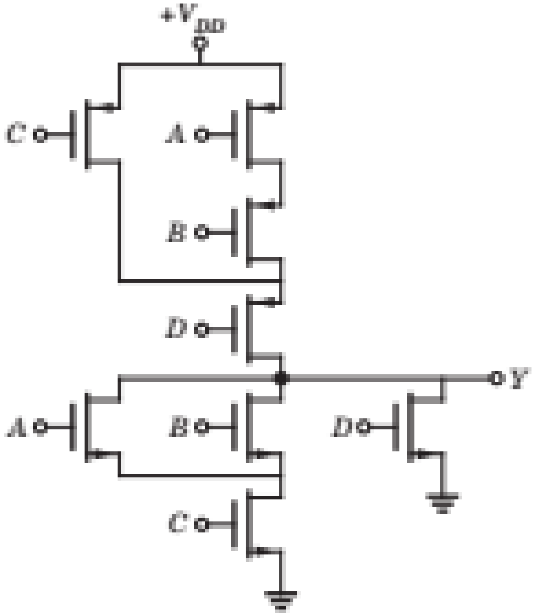
- The circuit shown in fig. implements the function:

(A) $(A + B)C + D$

(B) $\overline{(AB + C)D}$

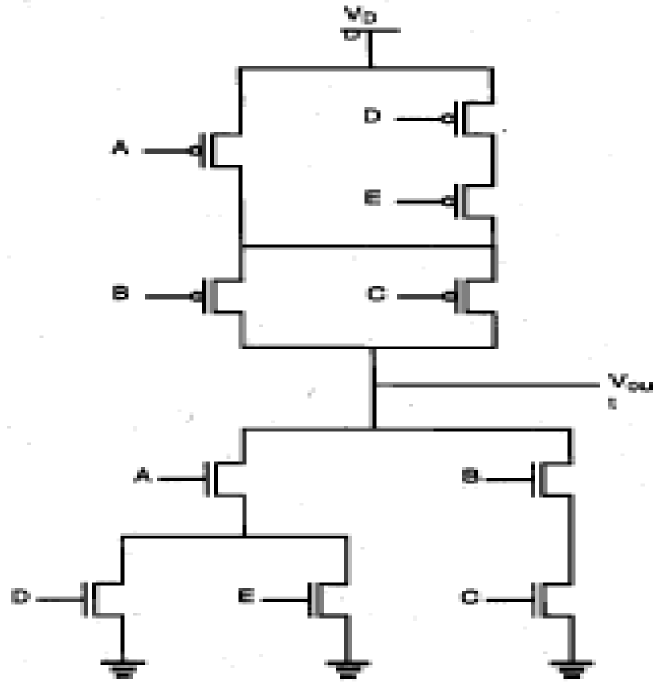
(C) $\overline{(A + B)C + D}$

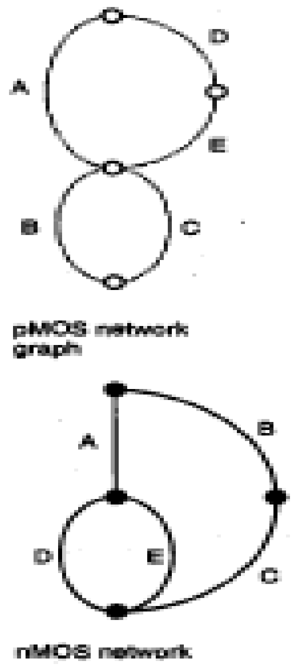
(D) $(AB + C)D$



Problem

Q1. Identify the logic expression for given function

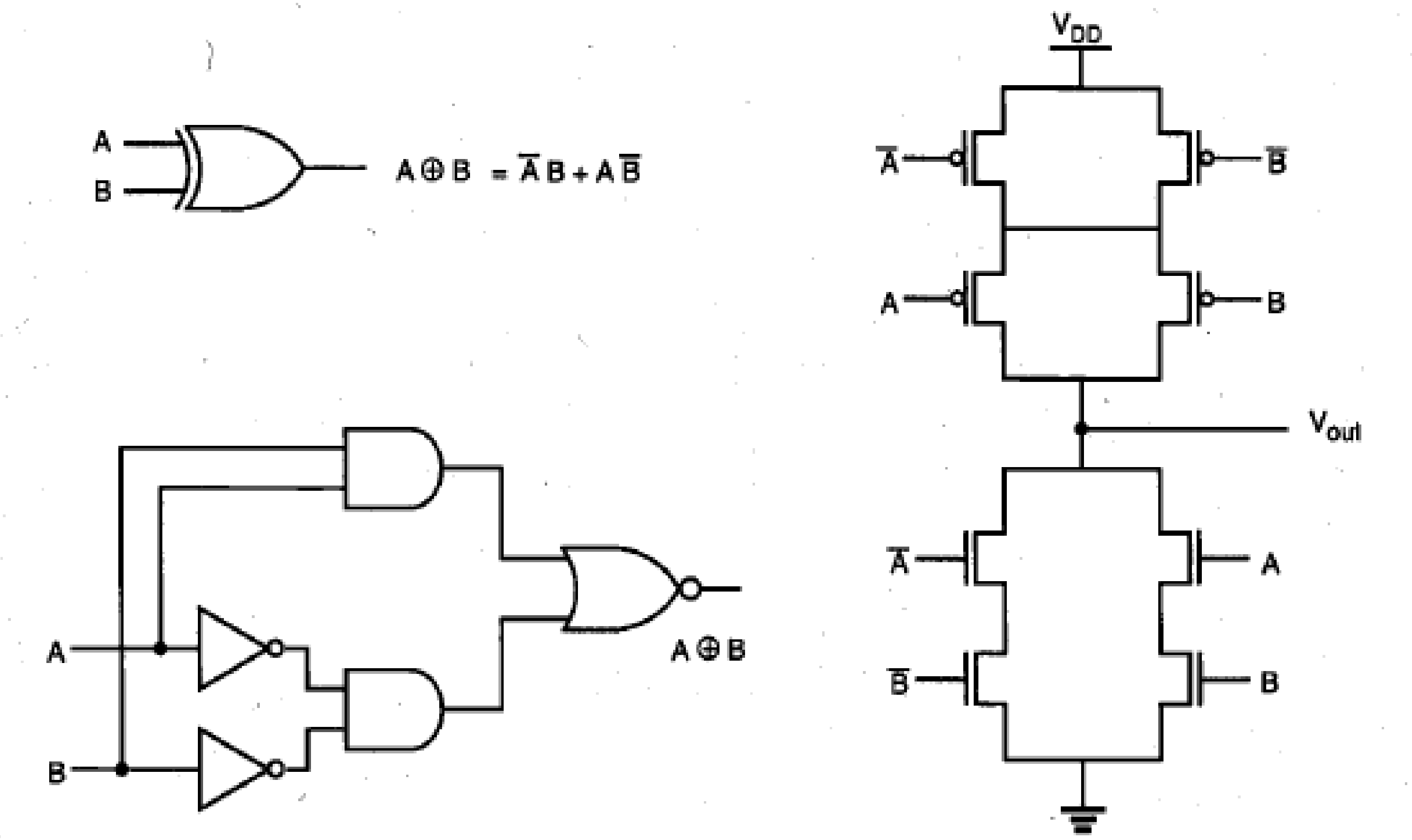




Q2. F=((A.B+C).D)’ :design given function using tanner EDA

Full-CMOS implementation of the exclusive-OR (XOR) function

- Ex-OR Gate



MCQ

Q. How many minimum number of transistors required to implement Ex-NOR gate

A) 8

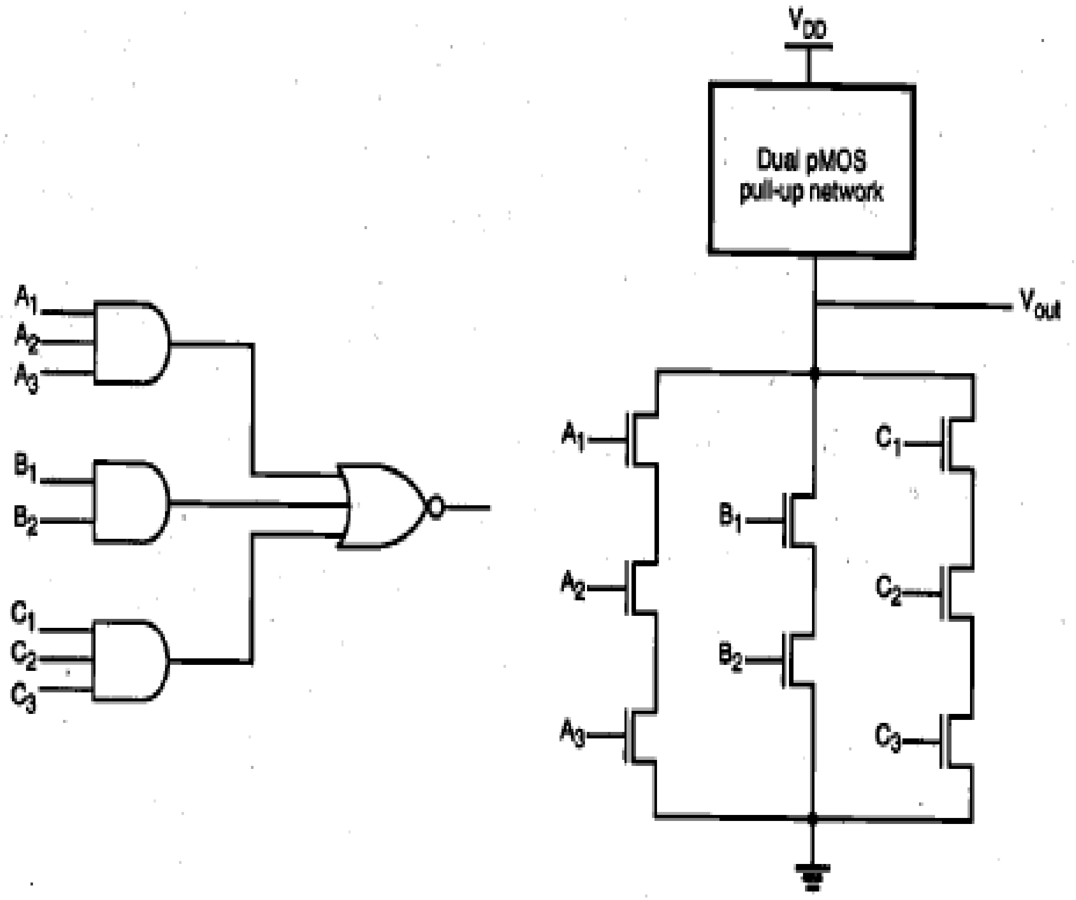
B) 10

C) 12

D) 14

AOI and OAI Gates

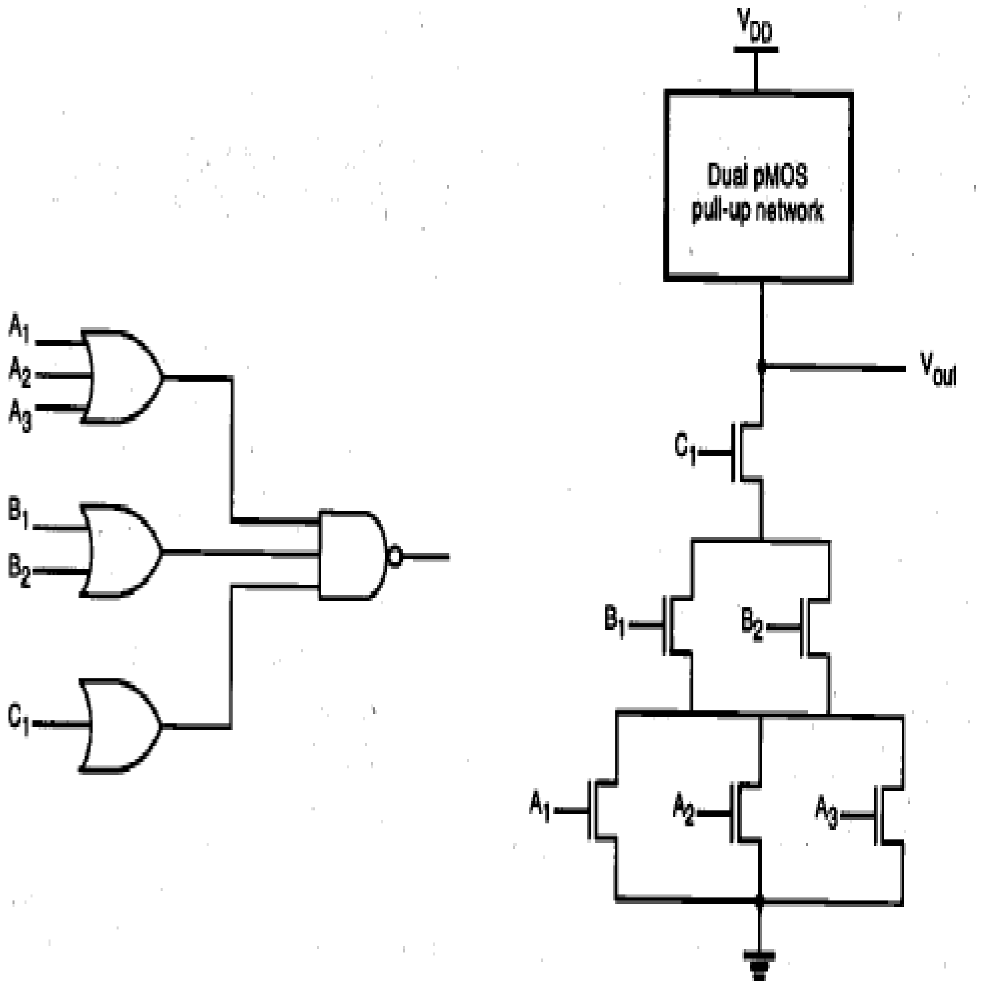
- Two important circuit categories as subsets of the general complex CMOS gate topology. These are the AND-OR-INVERT (AOI) gates and the OR-AND-INVERT (OAI) gates.
- The AOI gate, enables the sum-of-products realization of a Boolean function in one logic stage (Fig.) The pull-down net of the AOI gate consists of parallel branches of series-connected nMOS driver transistors



An AND-OR-INVERT (AOI) gate and the corresponding pull-down net

OR AND Invert

- OAI gate, on the other hand, enables the product-of-sums realization of a Boolean function in one logic stage (Fig.).
- The pull-down net of the OAI gate consists of series branches of parallel-connected nMOS driver transistors, while the corresponding p-type pull-up network can be found using the dual-graph concept.



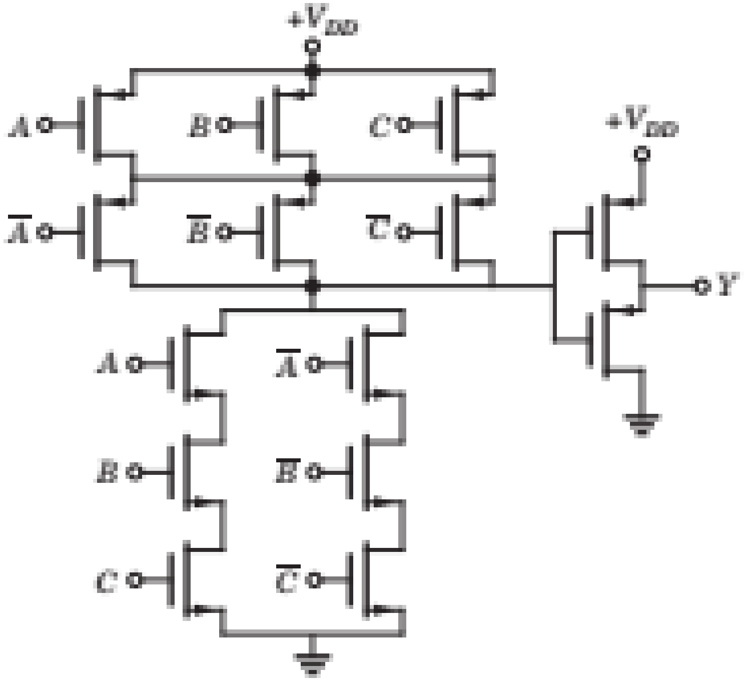
An OR-AND-INVERT (OAI) gate, and the corresponding pull-down net.

Problem 2

The circuit shown in fig. implements the function

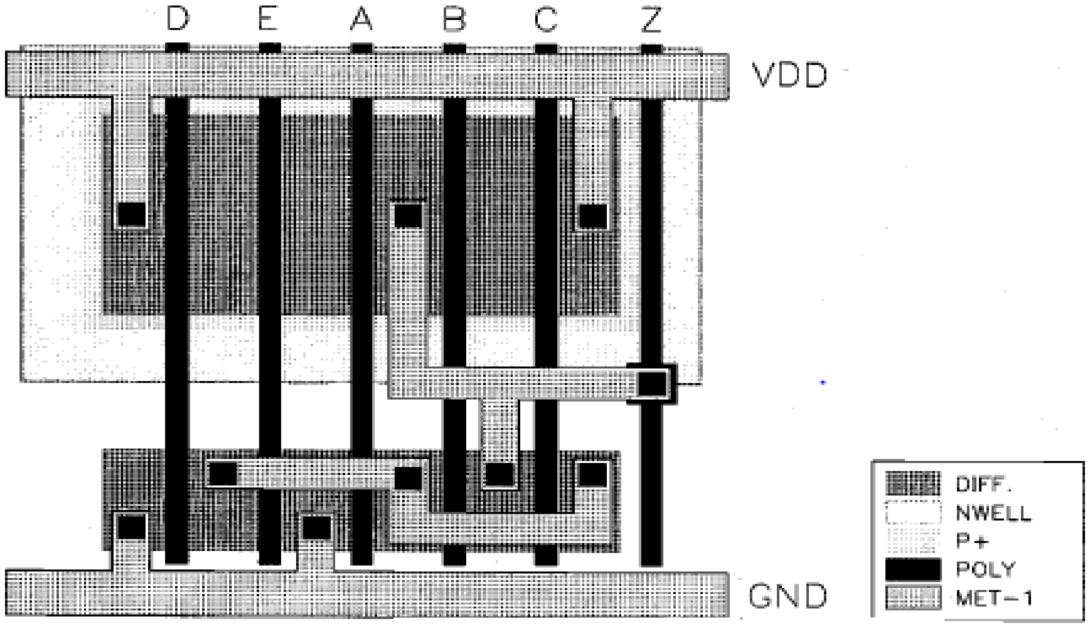
(A) $ABC + \overline{ABC}$ (B $ABC + \overline{(A + B + C})$

(C) $\overline{ABC} + \overline{(A + B + C})$ (D) None of the above



Example

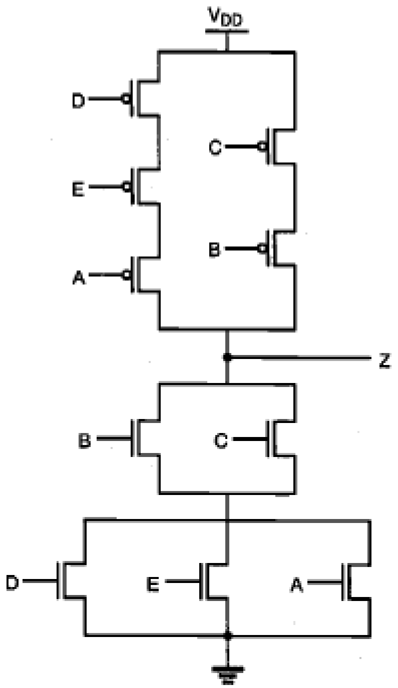
- Draw the corresponding circuit diagram, and find an equivalent CMOS inverter circuit for simultaneous switching of all inputs, assuming that *(W/L)p* = 15 for all pMOS transistors and *(W/L)n* = 10 for all nMOS transistors.



The Boolean function realized by this circuit is

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

The circuit diagram can be found fr



Example contd.

- *(W/L)p* = 15 for all pMOS transistors and *(W/L)n* = 10
- The equivalent *(W/L)* ratios of the nMOS network and the pMOS network are determined by using the series-parallel equivalency rules.

$$\left(\frac{W}{L}\right)_{n,eq} = \frac{1}{\dfrac{1}{\left(\frac{W}{L}\right)_D + \left(\frac{W}{L}\right)_E + \left(\frac{W}{L}\right)_A} + \dfrac{1}{\left(\frac{W}{L}\right)_B + \left(\frac{W}{L}\right)_C}}$$

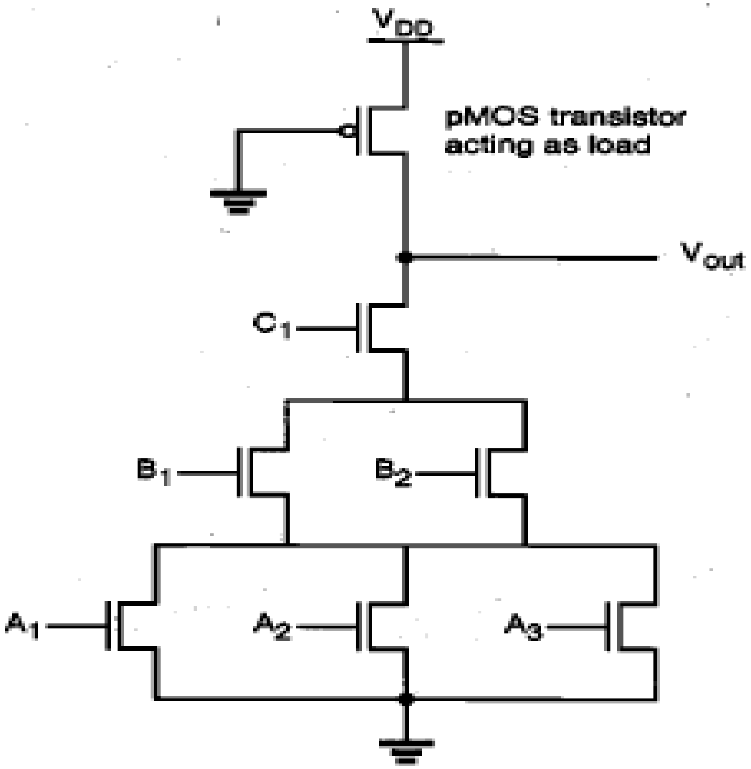
$$= \frac{1}{\dfrac{1}{30} + \dfrac{1}{20}} = 12$$

$$\left(\frac{W}{L}\right)_{p,eq} = \frac{1}{\dfrac{1}{\left(\frac{W}{L}\right)_D} + \dfrac{1}{\left(\frac{W}{L}\right)_E} + \dfrac{1}{\left(\frac{W}{L}\right)_A}} + \frac{1}{\dfrac{1}{\left(\frac{W}{L}\right)_B} + \dfrac{1}{\left(\frac{W}{L}\right)_C}}$$

$$= \frac{1}{\dfrac{1}{15} + \dfrac{1}{15} + \dfrac{1}{15}} + \frac{1}{\dfrac{1}{15} + \dfrac{1}{15}} = 12.5$$

Pseudo-nMOS Gates

- CMOS technology become more complex because large number of transistors.
- In Pseudo-nMOS Gates PUN can be replaced by single PMOS only.
- The most significant disadvantage of using a pseudo-nMOS gate instead of a full- CMOS gate is the nonzero static power dissipation, since the always-on pMOS load device conducts a steady-state current when the output voltage is lower than *VDD.*
- Also, the value of *VOL* and the noise margins are now determined by the *ratio* of the pMOS load transconductance to the pull-down or driver transconductance.

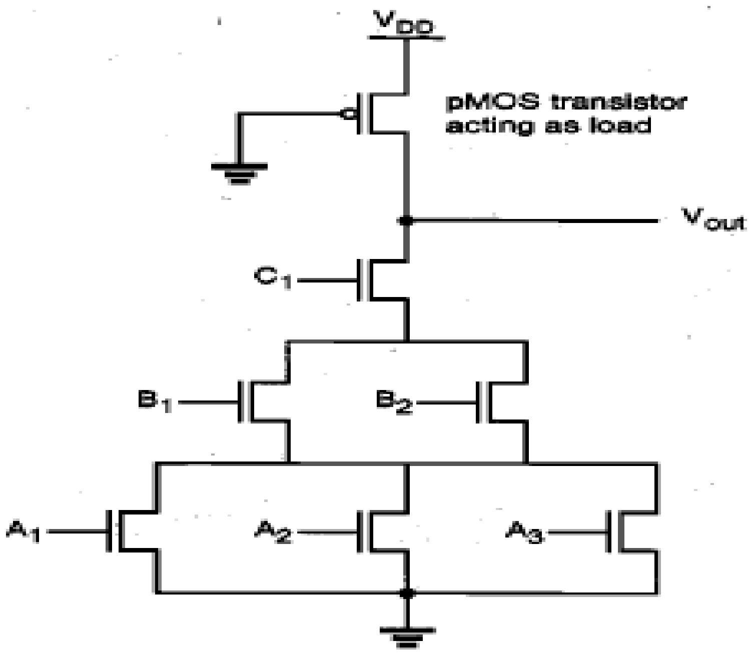


pseudo-nMOS implementation of the OAI gate

Pseudo-nMOS Gates

- CMOS technology become more complex because large number of transistors.
- In Pseudo-nMOS Gates PUN can be replaced by single PMOS only.
- The most significant disadvantage of using a pseudo-nMOS gate instead of a full- CMOS gate is the nonzero static power dissipation, since the always-on pMOS load device conducts a steady-state current when the output voltage is lower than *VDD.*
- Also, the value of *VOL* and the noise margins are now determined by the *ratio* of the pMOS load transcondu to the pull-down or driver transconductance.

No. of transistor=N+1



pseudo-nMOS implementation of the OAI gate

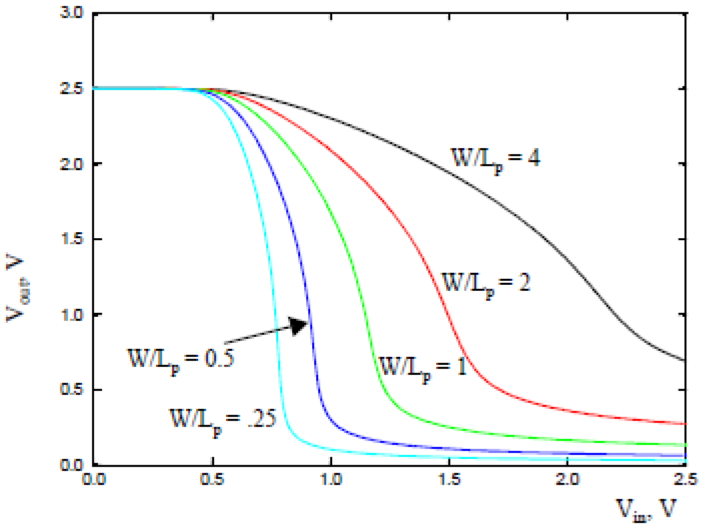


Figure 6.28 Voltage-transfer curves of the pseudo-NMOS inverter as a function of the PMOS size.

MCQ

The approximate number of transistor required for the implementation of N input Boolean function using pseudo NMOS, are:

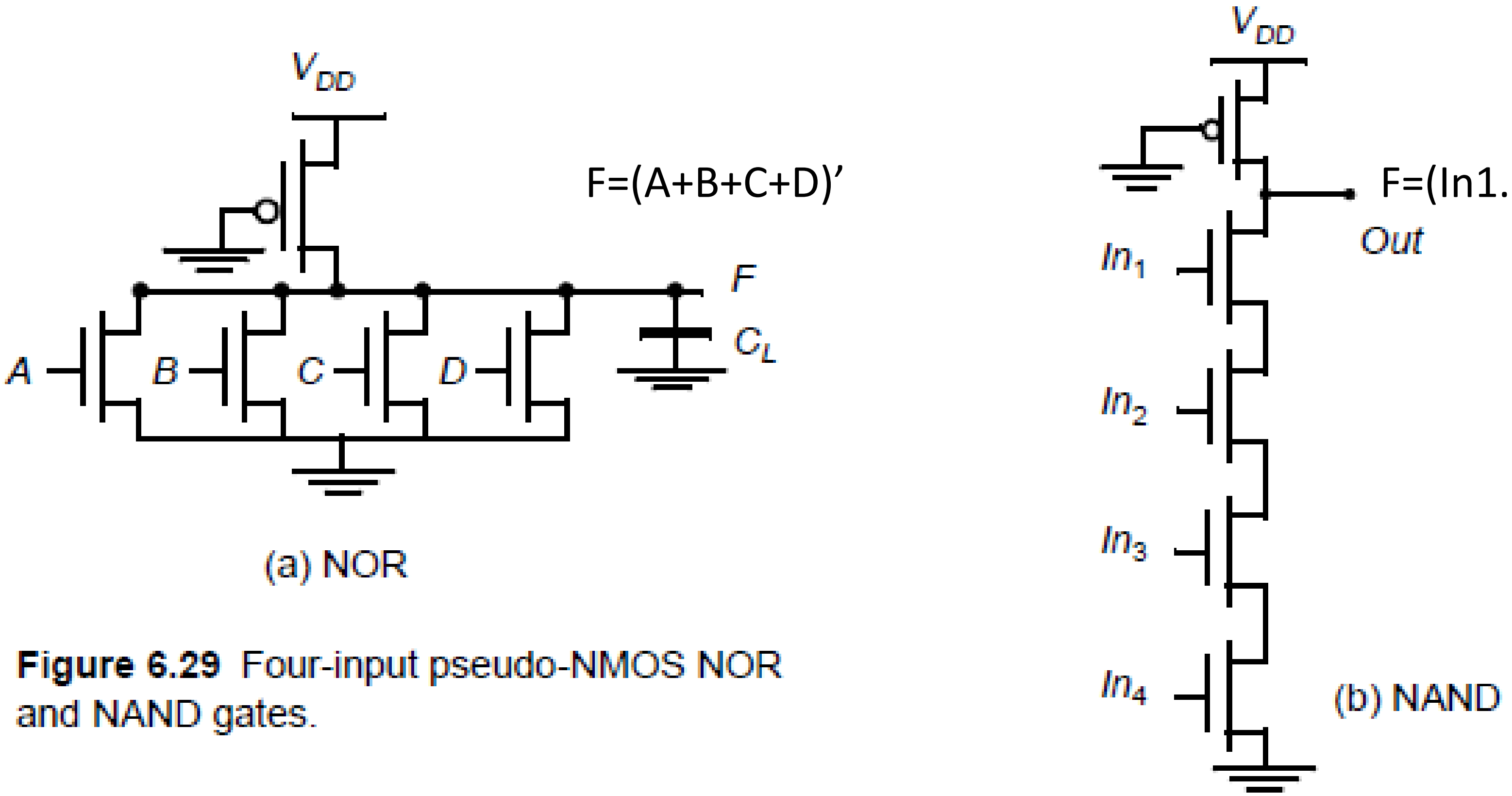
a) 2N+2

b) 2N+1

c) N+1

d) N

Example



F=(A+B+C+D)’

F=(In1.In2.In3.In4)’

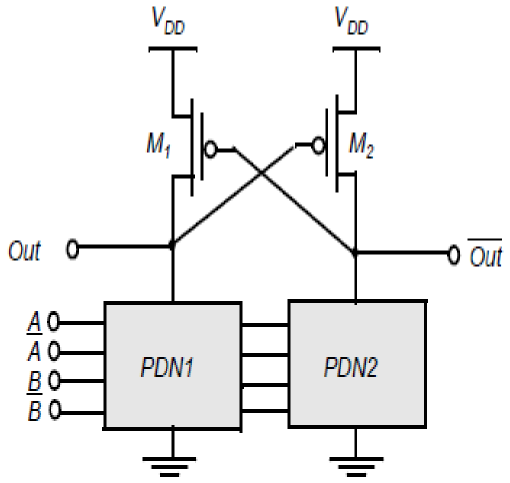
(a) NOR

(b) NAND

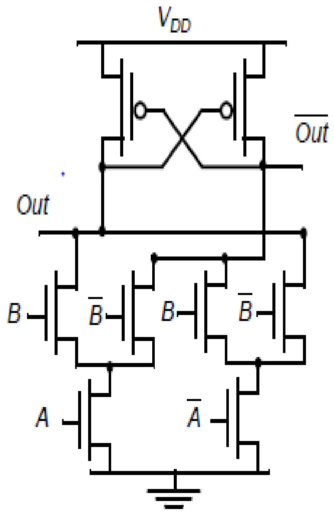
Figure 6.29 Four-input pseudo-NMOS NOR and NAND gates.

Differential Cascode Voltage Switch Logic

- It is possible to create a ratioed logic style that completely eliminates static currents and provides rail-to-rail swing. Such a gate combines two concepts: *differential logic* and *positive feedback*.
- A differential gate requires that each input is provided in complementary format, and produces complementary outputs in turn. The feedback mechanism ensures that the load device is turned off when not needed.
- A example of such a logic family, called *Differential Cascode Voltage Switch Logic* (or DCVSL)
- Here pull-down networks PDN1 and PDN2 use NMOS devices and are mutually exclusive (this is, when PDN1 conducts, PDN2 is off, and when PDN1 is off, PDN2 conducts)



(a) Basic principle



(b) XOR-XNOR gate

Pass Transistor Logic

- A popular and widely-used alternative to complementary CMOS is pass-transistor logic, which attempts to reduce the number of transistors required to implement logic by allowing the primary inputs to drive gate terminals as well as source/drain terminals.
- This is in contrast to logic families that we have studied so far, which only allow primary inputs to drive the gate terminals of MOSFETS.
- An NMOS device is effective at passing a 0 but is poor at pulling a node to *VDD*. When the pass transistor pulls a node high, the output only charges up to *VDD* -V_{Tn}.

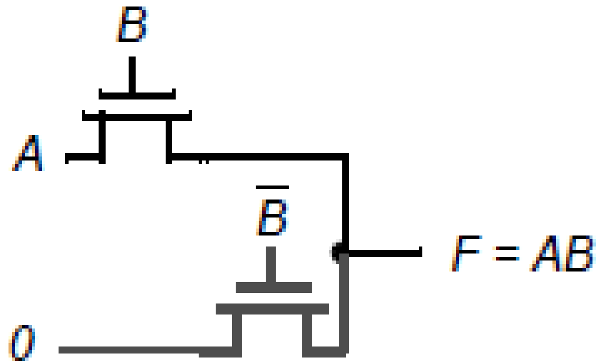
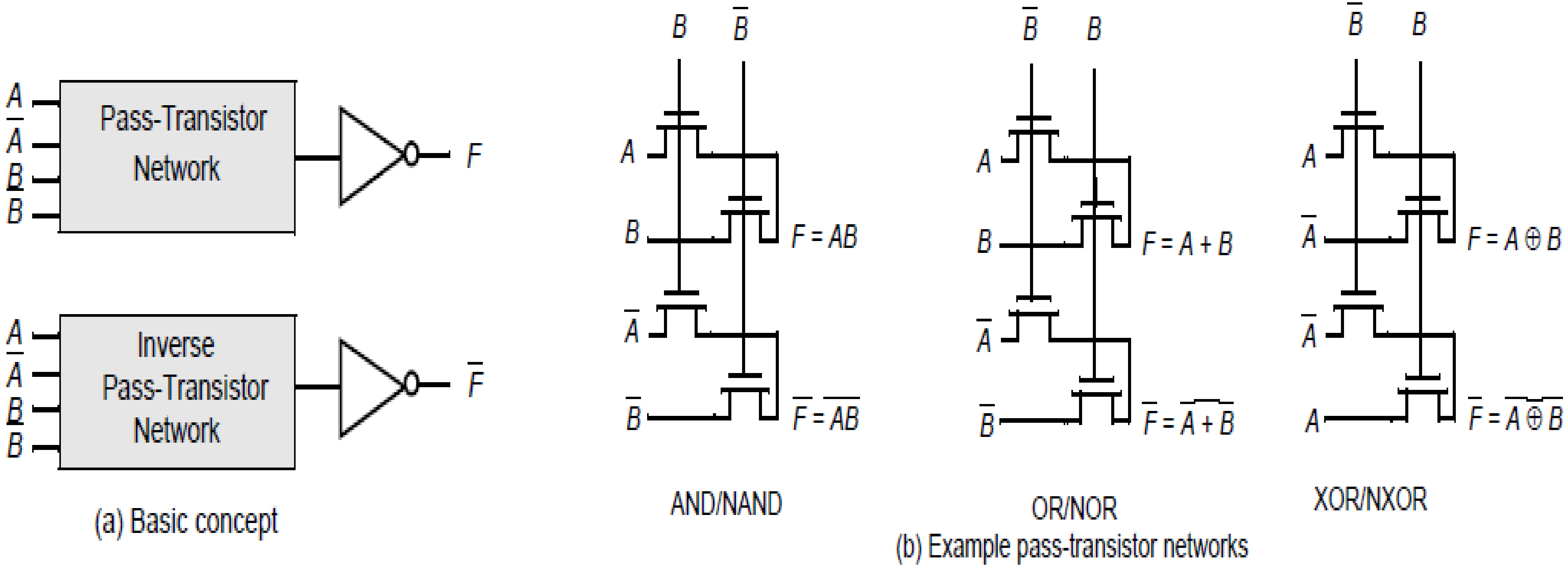


Figure 6.33 Pass-transistor implementation of an AND gate.

F=A+B

Pass Transistor Logic



(a) Basic concept

AND/NAND

OR/NOR

XOR/NXOR

(b) Example pass-transistor networks

Figure 6.37 Complementary pass-transistor logic (CPL).

Q1. Design half adder using pass transistor logic

Sum=AB’+A’B

Carry=A.B

Q2. Half subtractor using pass transistor logic

Diff=AB’+A’B

Borrow=A’B

Q3. Design half adder using pseudo NMOS

Sum=AB’+A’B

Carry=A.B

Q4. Half subtractor using pseudo NMOS

Diff=AB’+A’B

Borrow=A’B

Q5. Compare all above designs with CMOS technology in terms of power, delay and noise margin, and number of transistor

Robust and Efficient Pass-Transistor Design

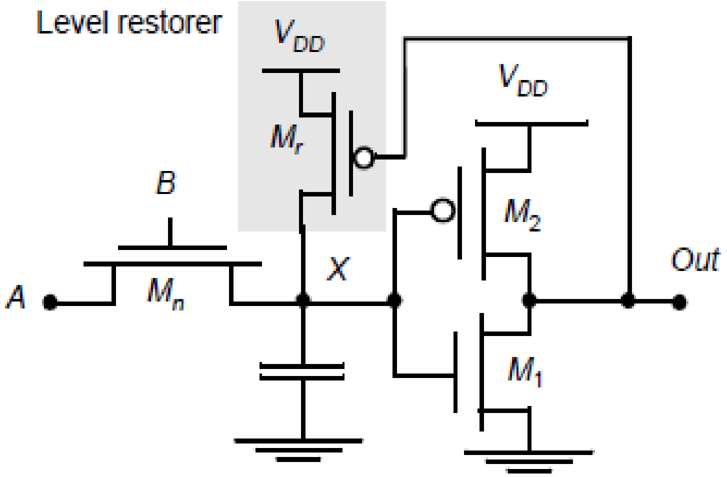
- Unfortunately, differential pass-transistor logic, like single-ended pass-transistor logic, suffers from static power dissipation and reduced noise margins, since the high input to the signal-restoring inverter only charges up to V_{DD}-V_{Tn}.

Solution 1: Level Restoration

- A common solution to the voltage drop problem is the use of a *level restorer*, which is a single PMOS configured in a feedback path
- The gate of the PMOS device is connected to the output of the inverter, its drain connected to the input of the inverter and the source to *VDD*.

- If input *A* makes a 0 to *VDD* transition, *Mn* only charges up node *X* to *VDD*-*VTn*. This is, however, enough to switch the output of the inverter low, turning on the feedback device *Mr* and pulling node *X* all theway to *VDD*.

- No static current path can exist through the level restorer and the pass-transistor, since the restorer is only active when *A* is high.
- In summary, this circuit has the advantage that all voltage levels are either at *GND* or *VDD*, and no static power is consumed.



Level-restoring circuit

Solution 2: Multiple-Threshold Transistors

- Technology solution to the voltage-drop problem associated with pass-transistor logic is the use of multiple-threshold devices.
- Using *zero threshold* devices for the NMOS pass-transistors eliminates most of the threshold drop, and passes a signal close to *VDD*.
- Notice that even if the devices threshold was implanted to be exactly equal to zero, the body effect of the device prevents a swing to *VDD*.
- All devices other than the pass transistors (i.e., the inverters) are implemented using standard high-threshold devices.
- The use of multiple-threshold transistors is becoming more common, and involves simple modifications to existing process flows.

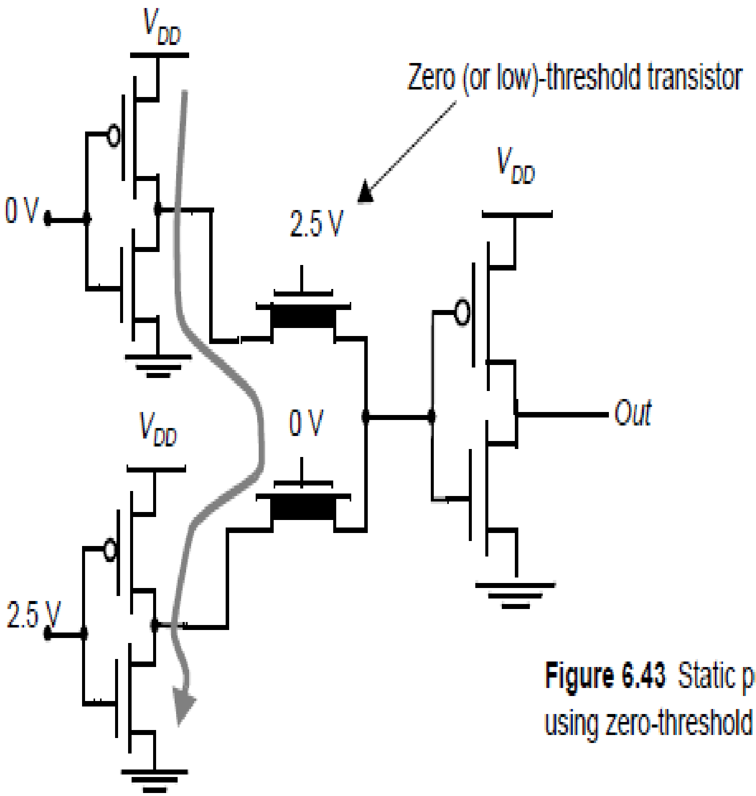


Figure 6.43 Static power consumption when using zero-threshold pass-transistors.

CMOS Transmission Gates

- The most widely-used solution to deal with the voltage-drop problem is the use of *transmission gates*
- CMOS transmission gate consists of one nMOS and one pMOS transistor, connected in parallel. The gate voltages applied to these two transistors are also set to be complementary signals.
- As such, the CMOS TG operates as a bidirectional switch between the nodes A and B which is controlled by signal C.
- If the control signal C is logic-high, i.e., equal to *VDD,* then both transistors are turned on and provide a low-resistance current path between the nodes A and B.
- If, on the other hand, the control signal C is low, then both transistors will be off, and the path between the nodes A and B will be an open circuit. This condition is also called the high-impedance state.

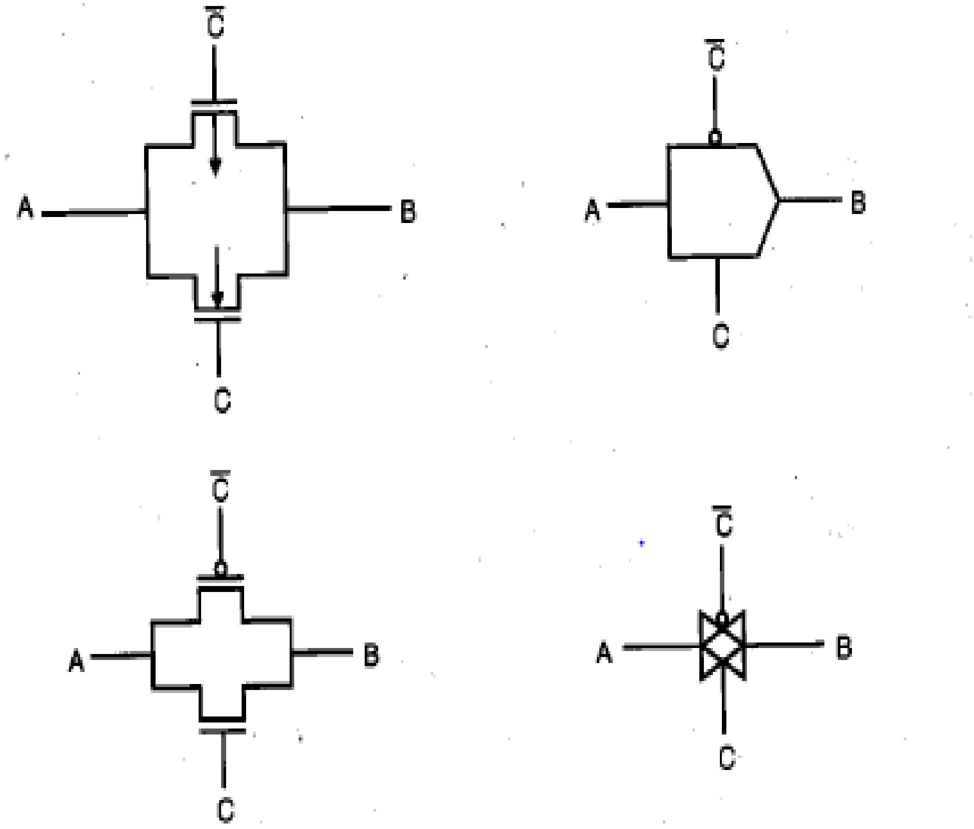


Fig: Four different representations of the CMOS transmission gate (TG).

Example CMOS TGs

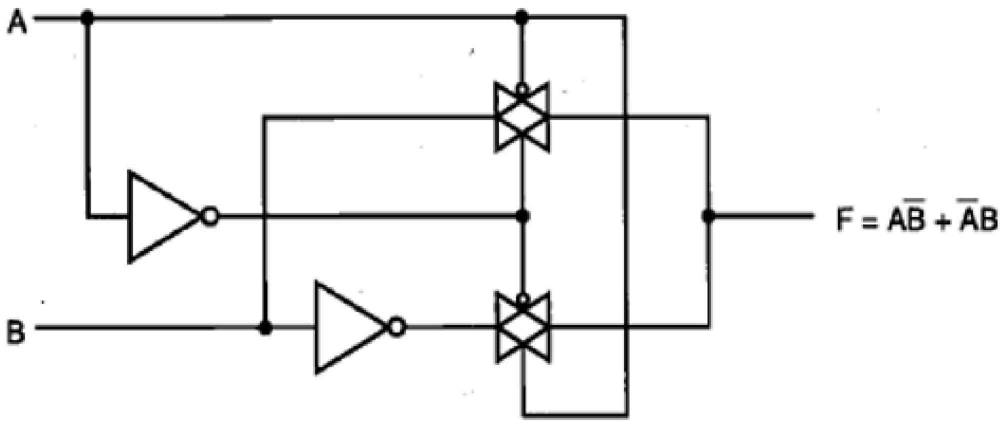


Fig. XOR gate using CMOS TG

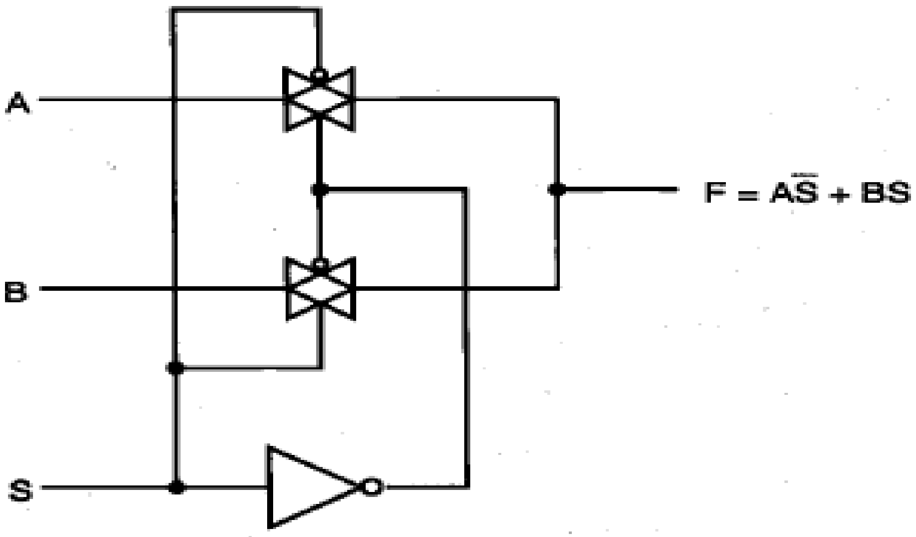


Fig. 2x1 Mux using CMOS TGs.

Static Properties of Complementary CMOS Gates

- Complementary CMOS gates inherit all the nice properties of the basic CMOS inverter.
- They exhibit rail to rail swing with *VOH* = *VDD* and *VOL* = *GND*.
- The circuits also have no static power dissipation, since the circuits are designed such that the pull-down and pullup networks are mutually exclusive.
- The DC voltage transfer characteristics and the noise margins depend upon the data input patterns applied to gate.
- For the NMOS devices to turn on, both gate-to-source voltages must be above *VTn*, with *VGS2* = *VA* - *VDS1* and *VGS1* = *VB*.
- voltage of transistor *M*2 will be higher than transistor *M*1 due to the body effect. The threshold voltages of the two devices are given by:

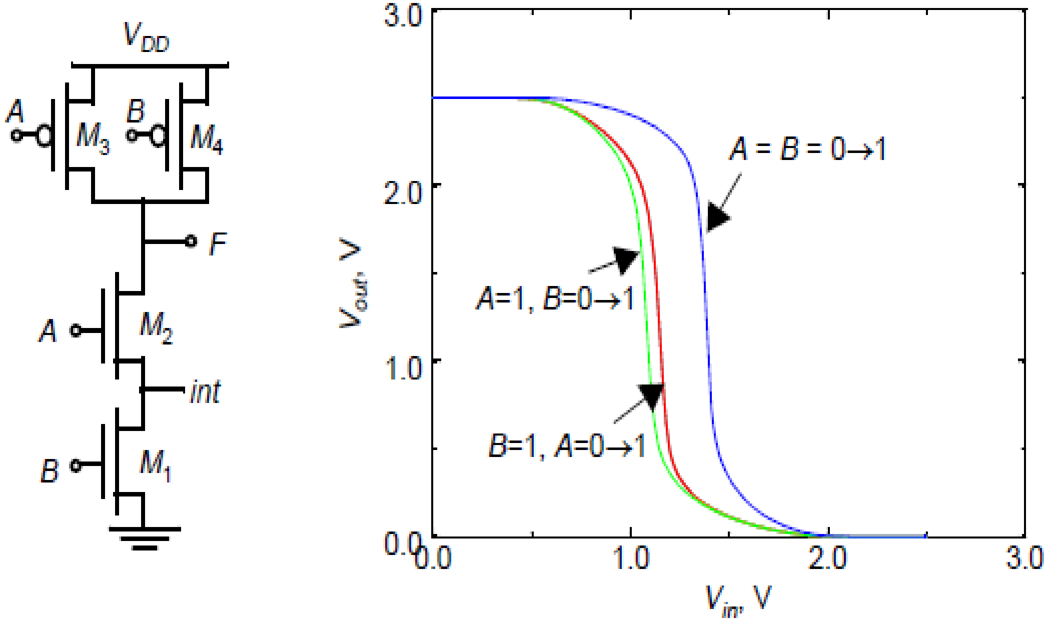


Fig:VTC of a two-input NAND is data-dependent. NMOS devices are 0.5mm/0.25mm while the PMOS devices are sized at 0.75mm/0.25mm.

$$V_{Tn2} = V_{tn0} + \gamma((\sqrt{|2\phi_f| + V_{int}}) - \sqrt{|2\phi_f|})$$

$$V_{Tn1} = V_{tn0}$$

Design Consideration

Propagation Delay of Complementary CMOS Gates

The computation of propagation delay proceeds in a fashion similar to the static inverter.

The logic is transformed into an equivalent RC network that includes the effect of internal node capacitances

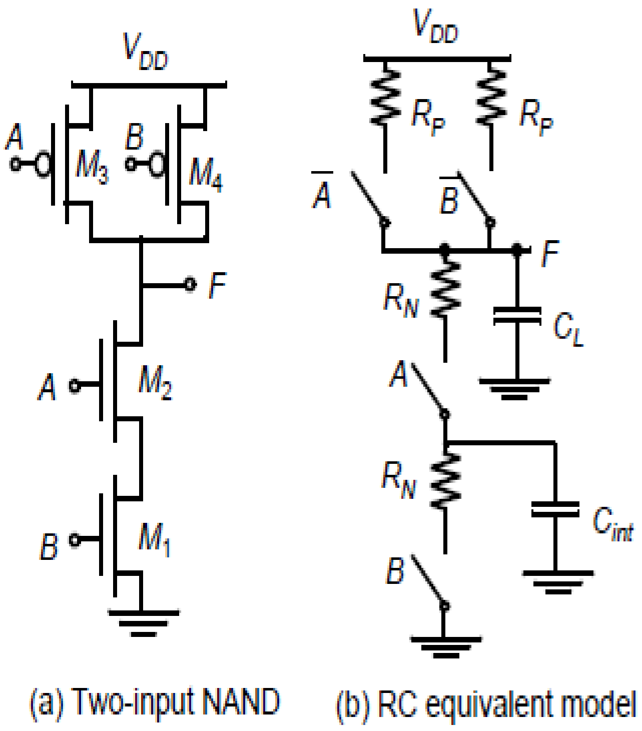


Fig6: Equivalent RC model for a 2-input NAND gate.

Delay

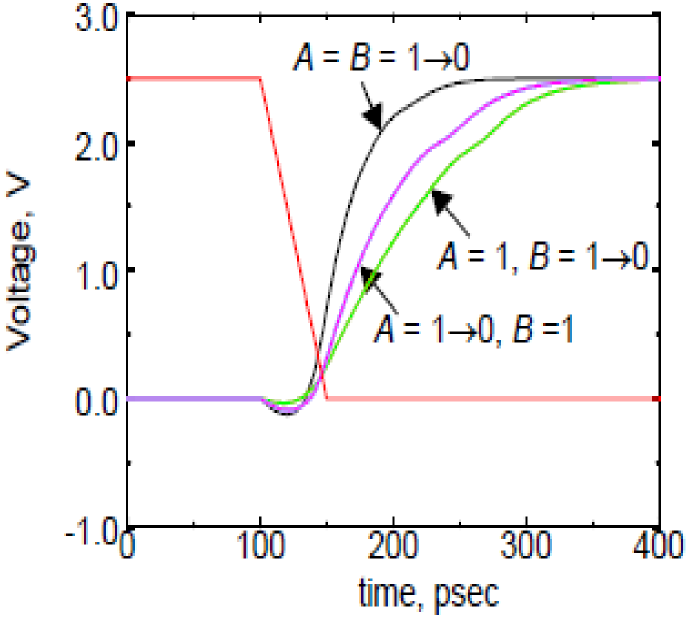
- If both inputs are driven low, the two PMOS devices are on. The delay in this case is
- Tp=0.69x(Rp/2)xCL, since the two resistors are in parallel.
- This is not the worst-case low-tohigh transition, which occurs when only one device turns on, and is given by 0.69x$RpxCL$.
- For the pull-down path, the output is discharged only if both A and B are switched high, and the delay is given by 0.69x (2RN)xCL to a first order

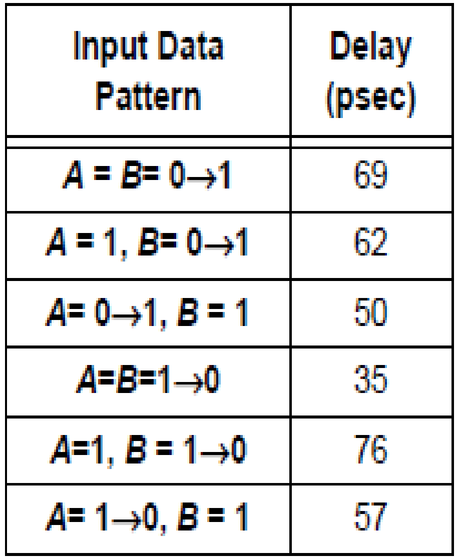
Example:Delay dependence on input patterns

Delay dependence on input patterns

Consider the NAND gate of Figure 6. Assume NMOS and PMOS devices of 0.5mm/0.25mm and 0.75mm/0.25mm, respectively. This sizing should result in approximately equal worst-case rise and fall times (since the effective resistance of the pull-down is designed to be equal to the pull-up resistance).

- Figure 7 shows the simulated low-to-high delay for different input patterns. As expected, the case where both inputs transition go low (*A* = *B* = 1→0) results in a smaller delay, compared to the case where only one input is driven low.



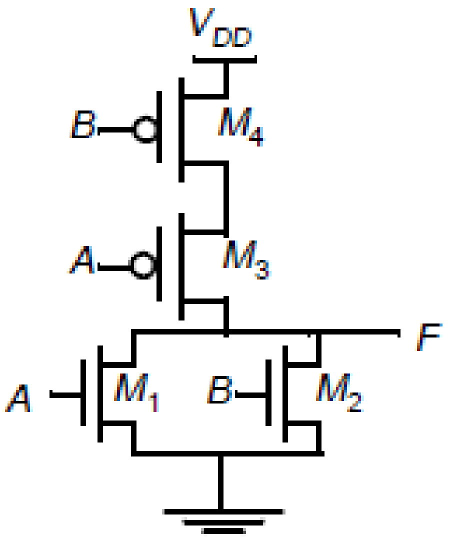


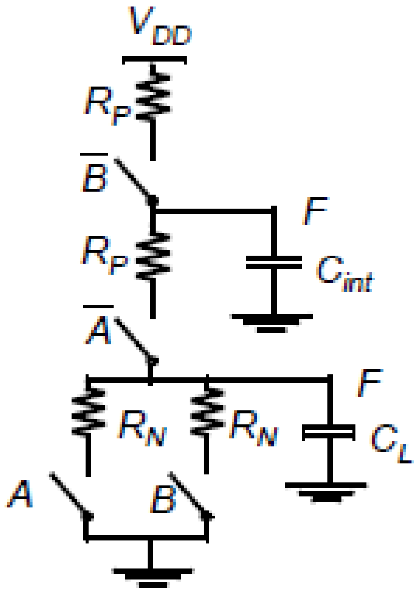
Input Data Pattern	Delay (psec)
A = *B*= 0→1	69
A = 1, *B*= 0→1	62
A= 0→1, *B* = 1	50
A=*B*=1→0	35
A=1, *B* = 1→0	76
A= 1→0, *B* = 1	57

Fig7: Example showing the delay dependence on input patterns

Sizing of NOR Gate

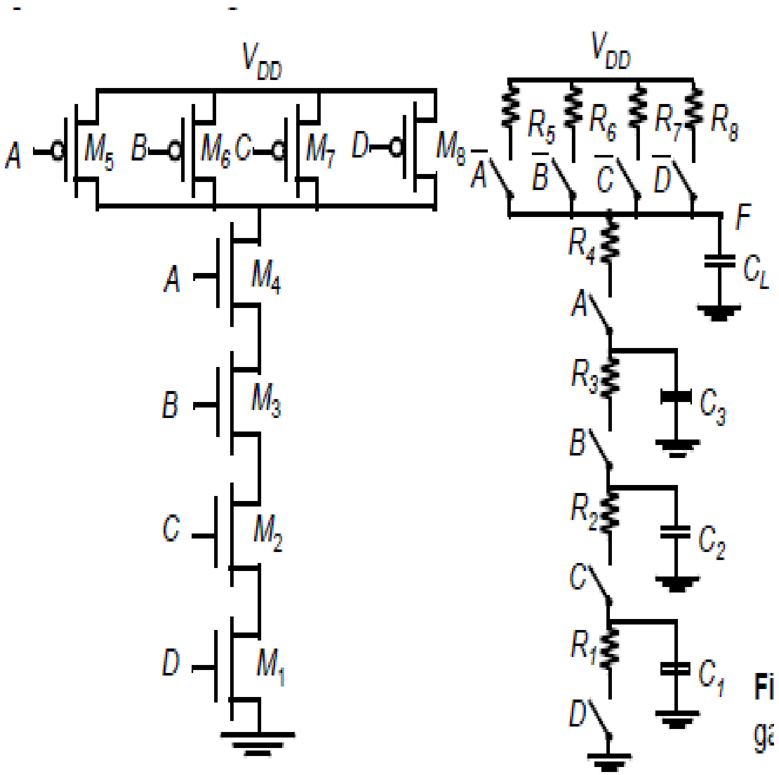
- A NAND implementation is clearly preferred over a NOR implementation for implementing generic logic.
- Sizing of a NOR gate to produce the same delay as an inverter with size of NMOS: 0.5µm/0.25µm and PMOS: *F* 1.5µm/0.25µm.





4-Input NAND Gate

- So far in the analysis of propagation delay, we have ignored the effect of internal node capacitances. This is often a reasonable assumption for a first-order analysis. However, in more complex logic gates that have large *fan-in*, the internal node capacitances can become significant.
- Consider a 4-input NAND gate as shown in Figure 9, which shows the equivalent RC model of the gate, including the internal node capacitances.
- The internal capacitances consist of the junction capacitance of the transistors, as well as the gate-to-source and gate-to-drain capacitances.
- The latter are turned into capacitances to ground using the Miller equivalence. The delay analysis for such a circuit involves solving distributed RC networks, a problem we already encountered when analyzing the delay of interconnect networks.

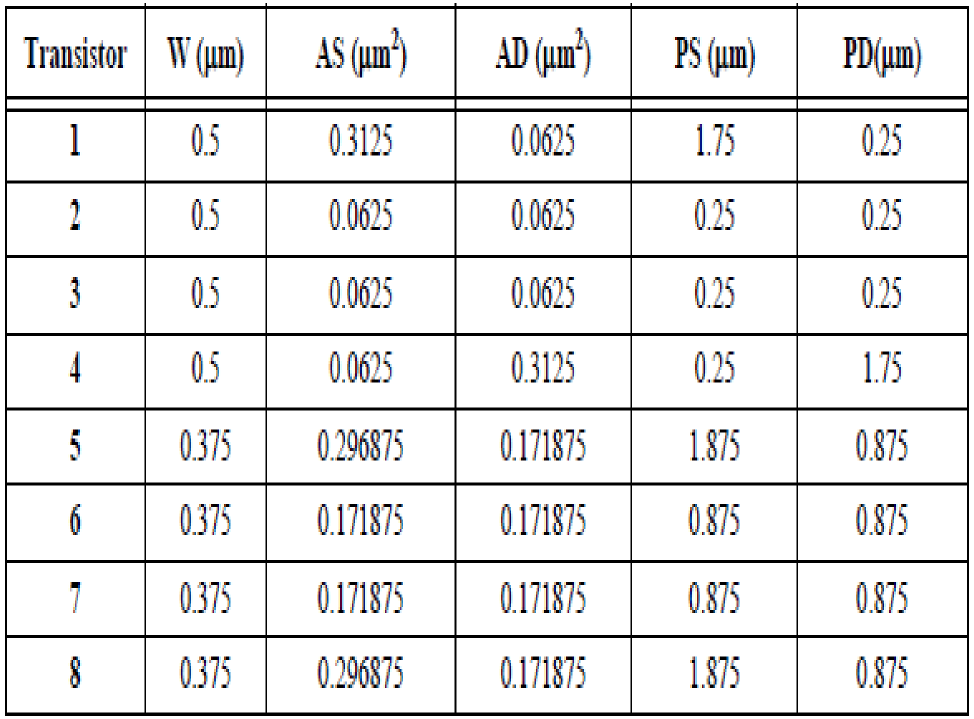


Fi
ga

4-Input NAND Gate Delay

- The propagation delay can be computed using the Elmore delay model and is approximated as:
- $t_{pHL} = 0.69(R1 \times C1 + (R1 + R2) \times C2 + (R1 + R2 + R3) \times C3 + (R1 + R2 + R3 + R4) \times CL)$ --3
- Assuming that all NMOS ndevices have an equal size, Eq. (3) simplifies to
- $t_{pHL} = 0.69R_N(C1 + 2 \times C2 + 3 \times C3 + 4 \times C_L)$ --4

Table 6.1 Area and perimeter of transistors in 4 input NAND gate.



Transistor	W (µm)	AS (µm^2)	AD (µm^2)	PS (µm)	PD(µm)
1	0.5	0.3125	0.0625	1.75	0.25
2	0.5	0.0625	0.0625	0.25	0.25
3	0.5	0.0625	0.0625	0.25	0.25
4	0.5	0.0625	0.3125	0.25	1.75
5	0.375	0.296875	0.171875	1.875	0.875
6	0.375	0.171875	0.171875	0.875	0.875
7	0.375	0.171875	0.171875	0.875	0.875
8	0.375	0.296875	0.171875	1.875	0.875

Layout

- Assume that all NMOS devices have a *W*/*L* of 0.5mm/0.25mm, and all PMOS devices have a device size of 0.375mm/0.25mm. The layout of a four-input NAND gate is shown in Figure

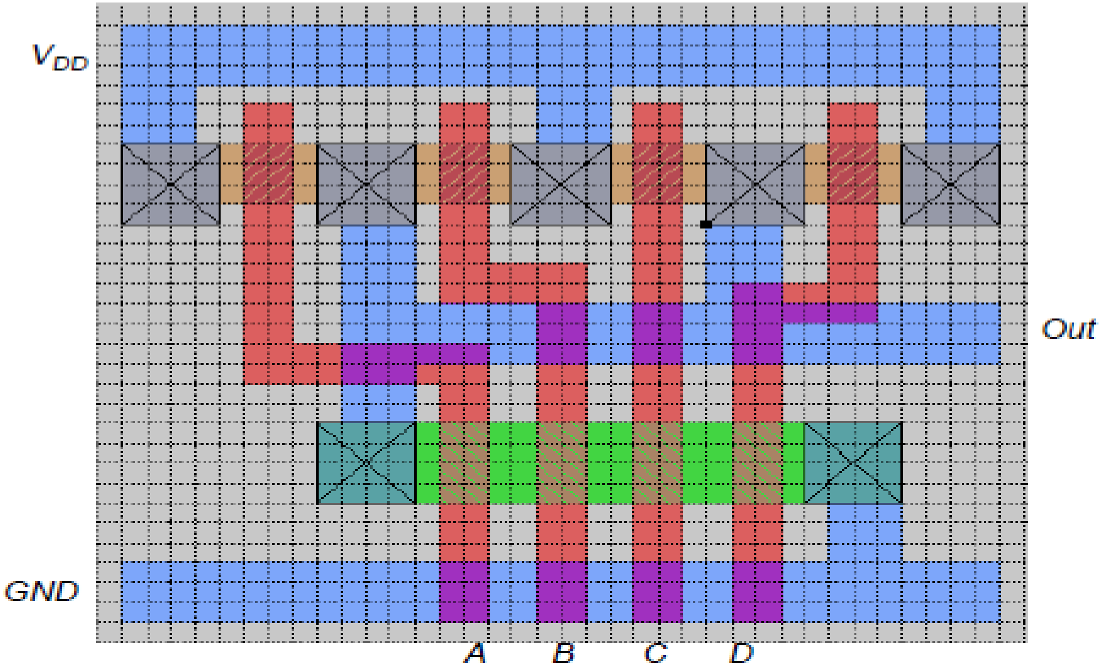


Fig: Layout a four-input NAND gate in complementary CMOS

4 input NAND

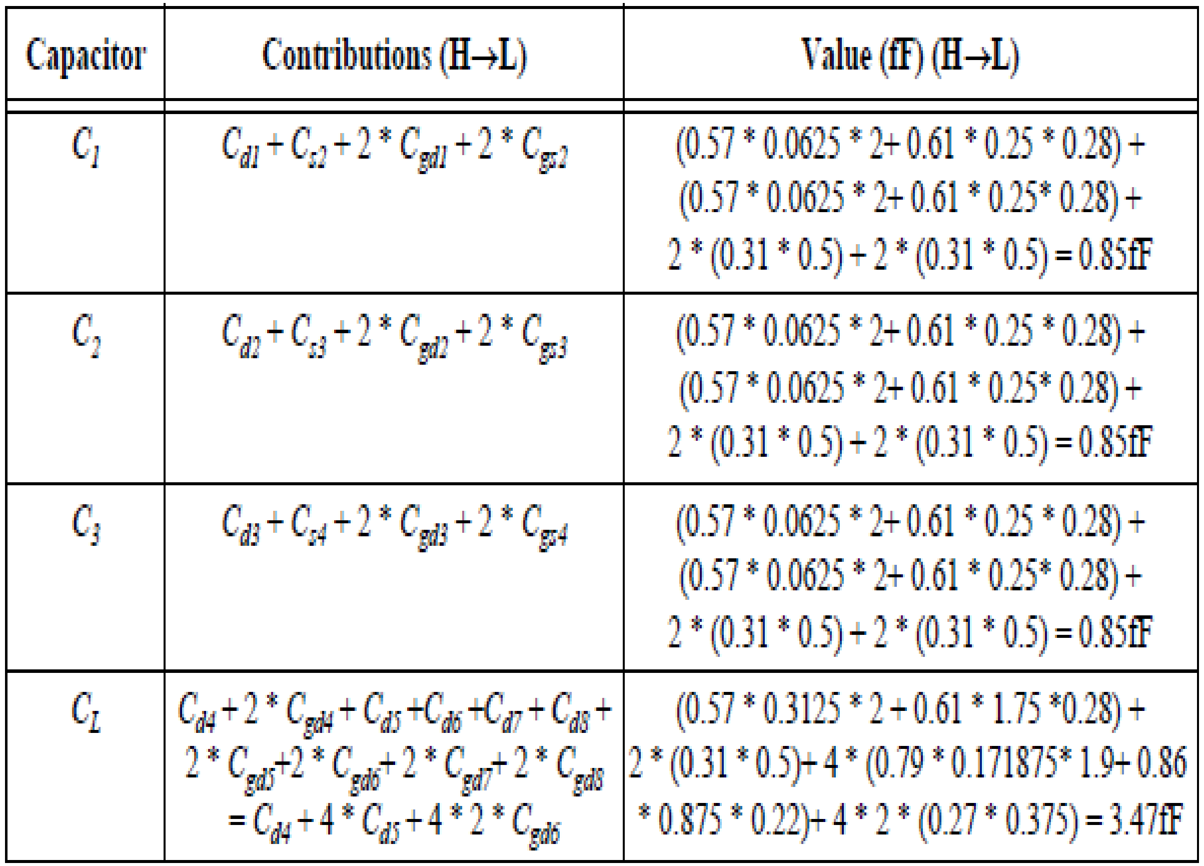
- Delay

t_{pHL}= 0.69 (13$K\Omega$ /2) (0.85fF + 2 x 0.85fF + 3x0.85fF + 4x 3.47fF)= 85ps

FAN IN FAN OUT

$tp = a1FI + a2FI^2 + a3FO$

where *FI* and *FO* are the *fan-in* and *fan-out* of the gate, respectively, and *a1, a2* and *a3* are weighting factors that are a function of the technology



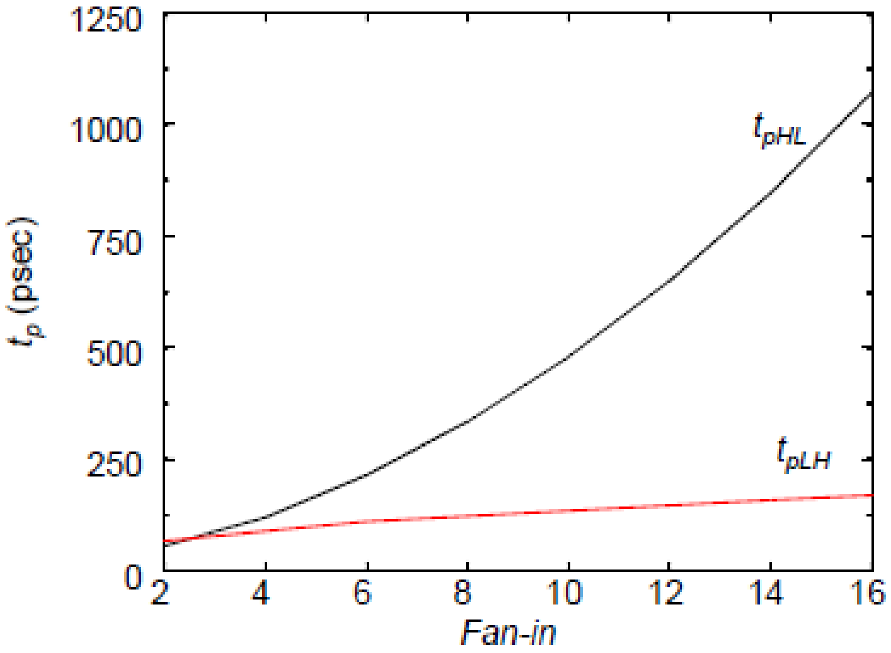
Capacitor	Contributions (H→L)	Value (fF) (H→L)
C_1	$C_{d1} + C_{s2} + 2 * C_{gd1} + 2 * C_{gs2}$	$(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $2 * (0.31 * 0.5) + 2 * (0.31 * 0.5) = 0.85\text{fF}$
C_2	$C_{d2} + C_{s3} + 2 * C_{gd2} + 2 * C_{gs3}$	$(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $2 * (0.31 * 0.5) + 2 * (0.31 * 0.5) = 0.85\text{fF}$
C_3	$C_{d3} + C_{s4} + 2 * C_{gd3} + 2 * C_{gs4}$	$(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $(0.57 * 0.0625 * 2 + 0.61 * 0.25 * 0.28) +$ $2 * (0.31 * 0.5) + 2 * (0.31 * 0.5) = 0.85\text{fF}$
C_L	$C_{d4} + 2 * C_{gd4} + C_{d5} + C_{d6} + C_{d7} + C_{d8} +$ $2 * C_{gd5} + 2 * C_{gd6} + 2 * C_{gd7} + 2 * C_{gd8}$ $= C_{d4} + 4 * C_{d5} + 4 * 2 * C_{gd6}$	$(0.57 * 0.3125 * 2 + 0.61 * 1.75 * 0.28) +$ $2 * (0.31 * 0.5) + 4 * (0.79 * 0.171875 * 1.9 + 0.86$ $* 0.875 * 0.22) + 4 * 2 * (0.27 * 0.375) = 3.47\text{fF}$

Design Technique for Large FAN-IN

- Propagation delay of CMOS NAND gate as a function of fan-in. A fan-out of one inverter is assumed, and all pull-down transistors are minimal size.

1. Transistor Sizing

- For larger Fan-In, overall transistor size can be increased. This lowers the resistance of devices in series and lowers the time constant.
- However, increasing the transistor size, results in larger parasitic capacitors, which do not only affect the *propagation delay* of the gate but also present a larger load to the preceding gate.



Delay versus Fan-In

Design Technique for Large FAN-IN

2. Progressive Transistor Sizing

- An alternate approach to uniform sizing (in which each transistor is scaled up uniformly), is to use progressive transistor sizing
- Farthest transistor from output node should be largest in size beacuse that contribute to delay due to each node.
- M1>M2>M3>M4

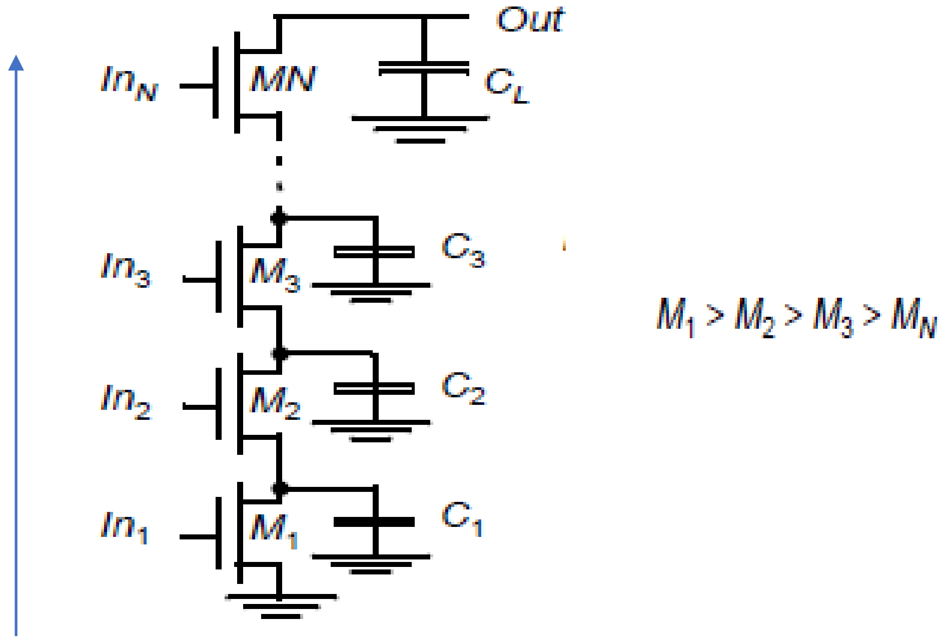


Fig: Progressive sizing of transistors in large transistor chains copes with the extra load of internal capacitances

Problems

Q. Compare 2 input and 4 input NAND gate in terms of design and performance parameters like delay and power consumption.

MCQ

Q. With Increased Fan-In, Delay and power consumption in CMOS Logic

a) increases, increases
b) increases, decreases
c) decreases, increases
d) decreases, decreases

Q. Transistor delay can be reduced by

a) Transistor sizing
b) Reduced Fan-in
c) Increased Fan-In
d) a and b
e) **2,7,11,14,18**