

FUNDAMENTALS OF VLSI DESIGN

(19EC6DCFOV)

Module-1

An overview of VLSI : Complexity and design, basic concepts (Text book-2)

Basic MOS Technology: Introduction to MOS transistors, nMOS fabrication, CMOS fabrication, Bi-CMOS technology. (Text book-1)

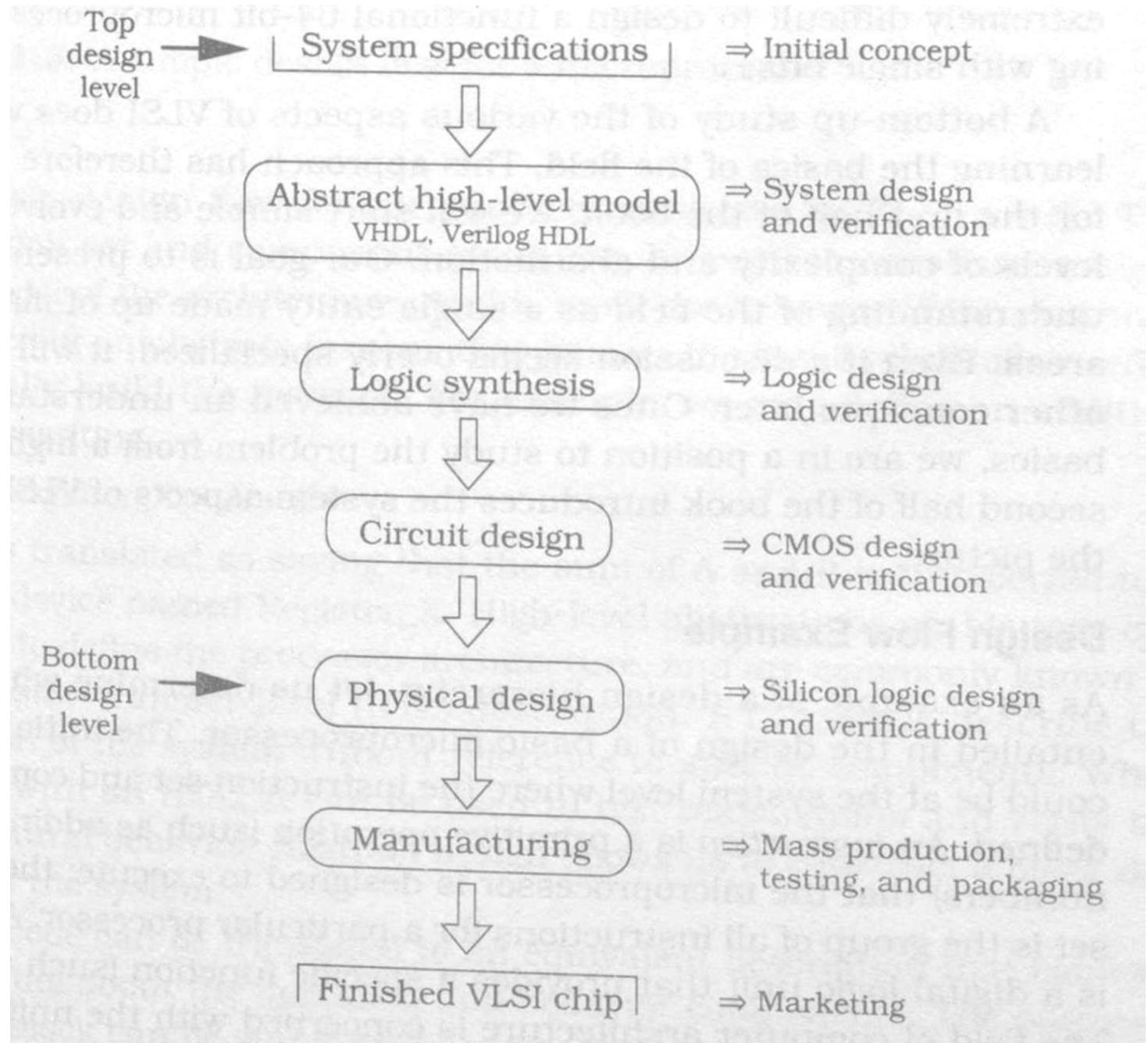
Logic Design with MOSFETs: Ideal switches and Boolean operations, MOSFETs as switches, Basic logic gates in CMOS, Complex logic gates in CMOS, Transmission gates (Text book-2)

1. Douglas A. Pucknell, Kamran E., "Basic VLSI Design", 3rd Edition, PHI Publication, India.
2. John P.Uyemura, "Introduction to VLSI Circuits and Systems", Wiley India Edition, 3rd print, 2007

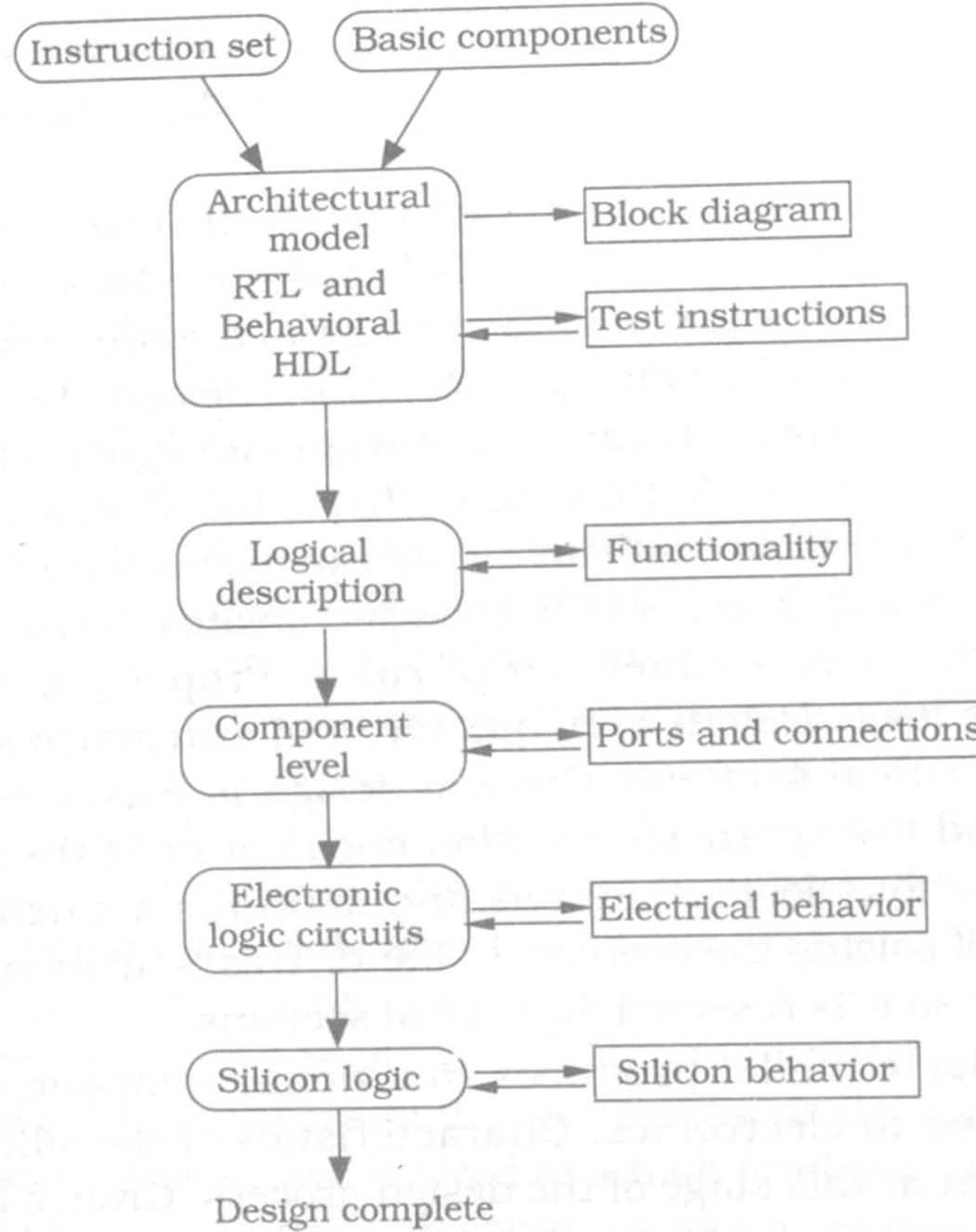
Complexity and design

Two Approaches:

- Top down approach
- Bottom up approach



Design flow example: Microprocessor



VLSI Chip Types:

Classified by the approach used to implement and build the circuit

- Full custom Design
- ASICs
- Semi custom Design

Basic MOS Transistor

Two modes of Operation:

1. Enhancement mode
2. Depletion mode

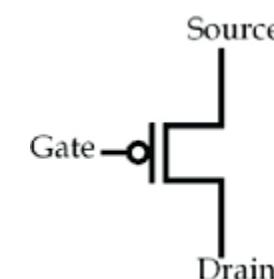
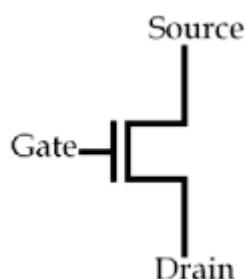
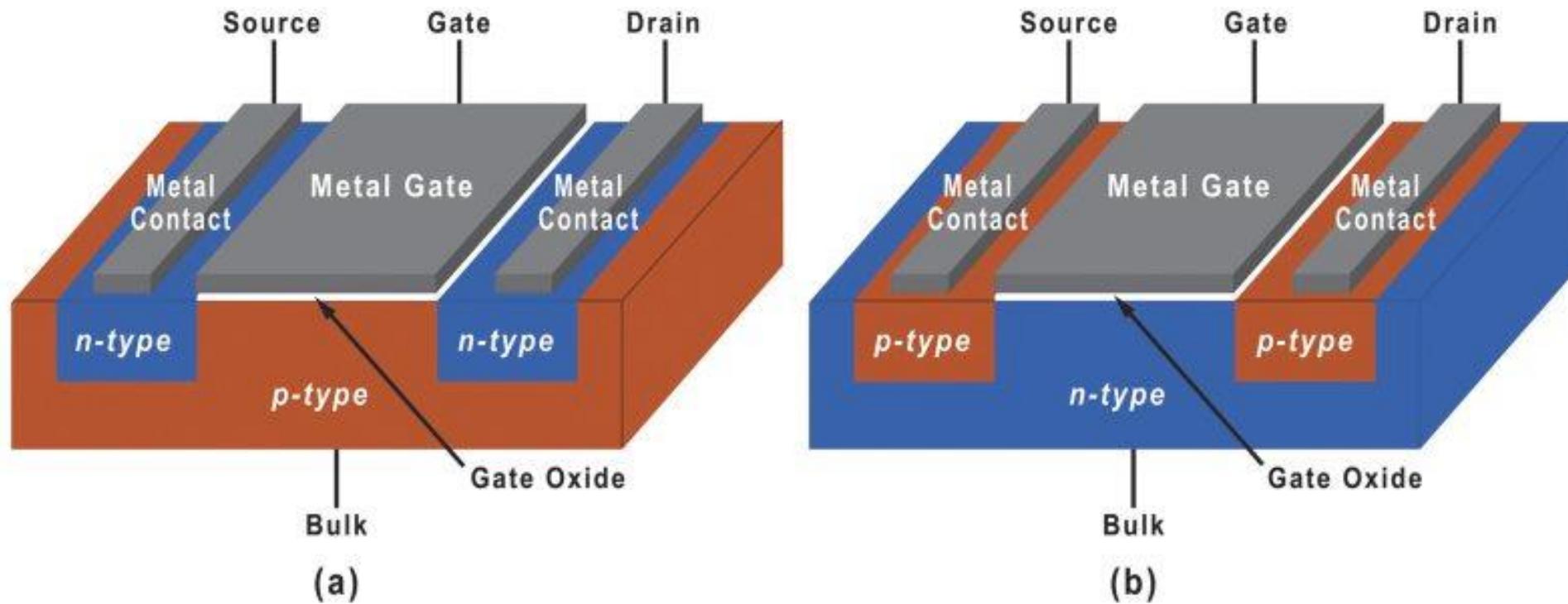
Enhancement type:

- Normally OFF
- requires V_{gs} to switch the device “ON”
- Normally Open switch

Depletion type:

- Normally ON
- requires V_{gs} to switch the device “OFF”
- Normally Closed switch

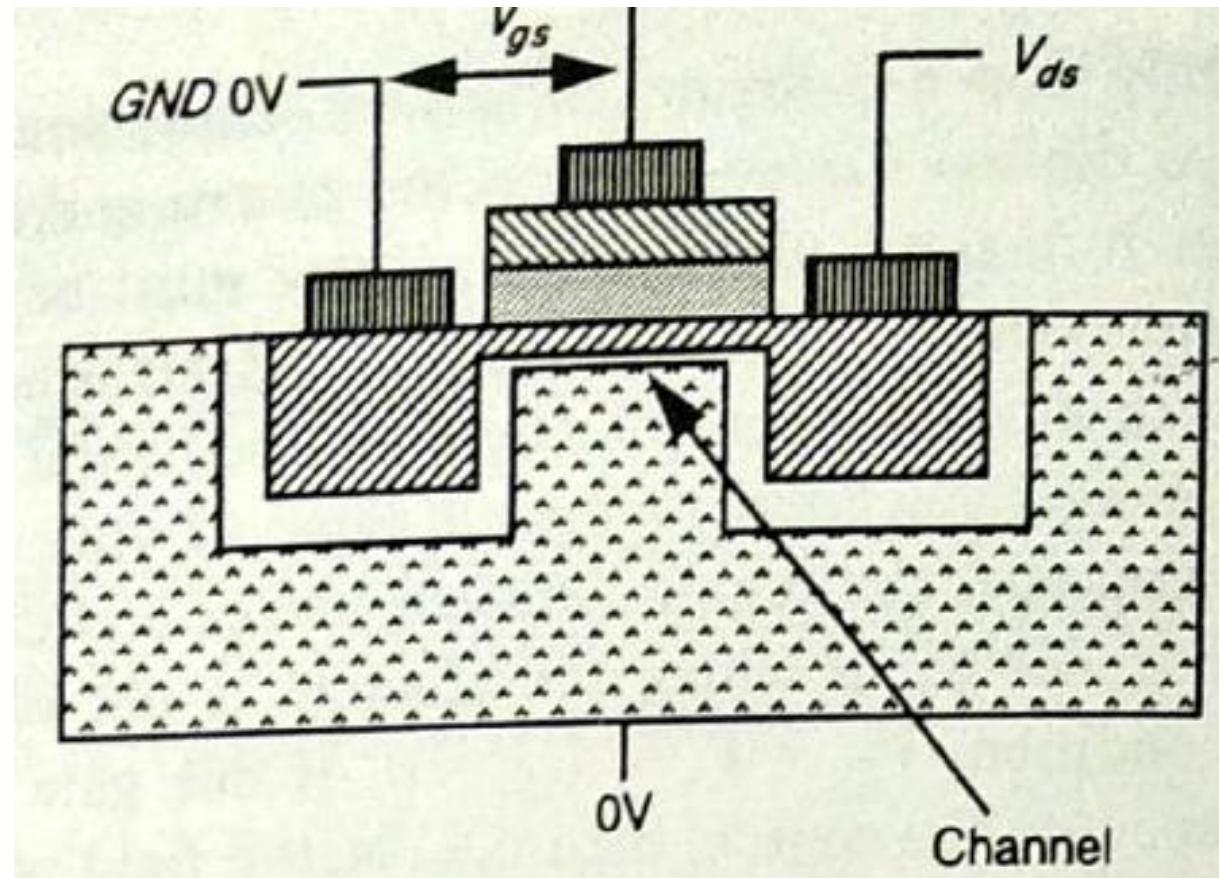
Type of Enhancement Transistor and Circuit symbol



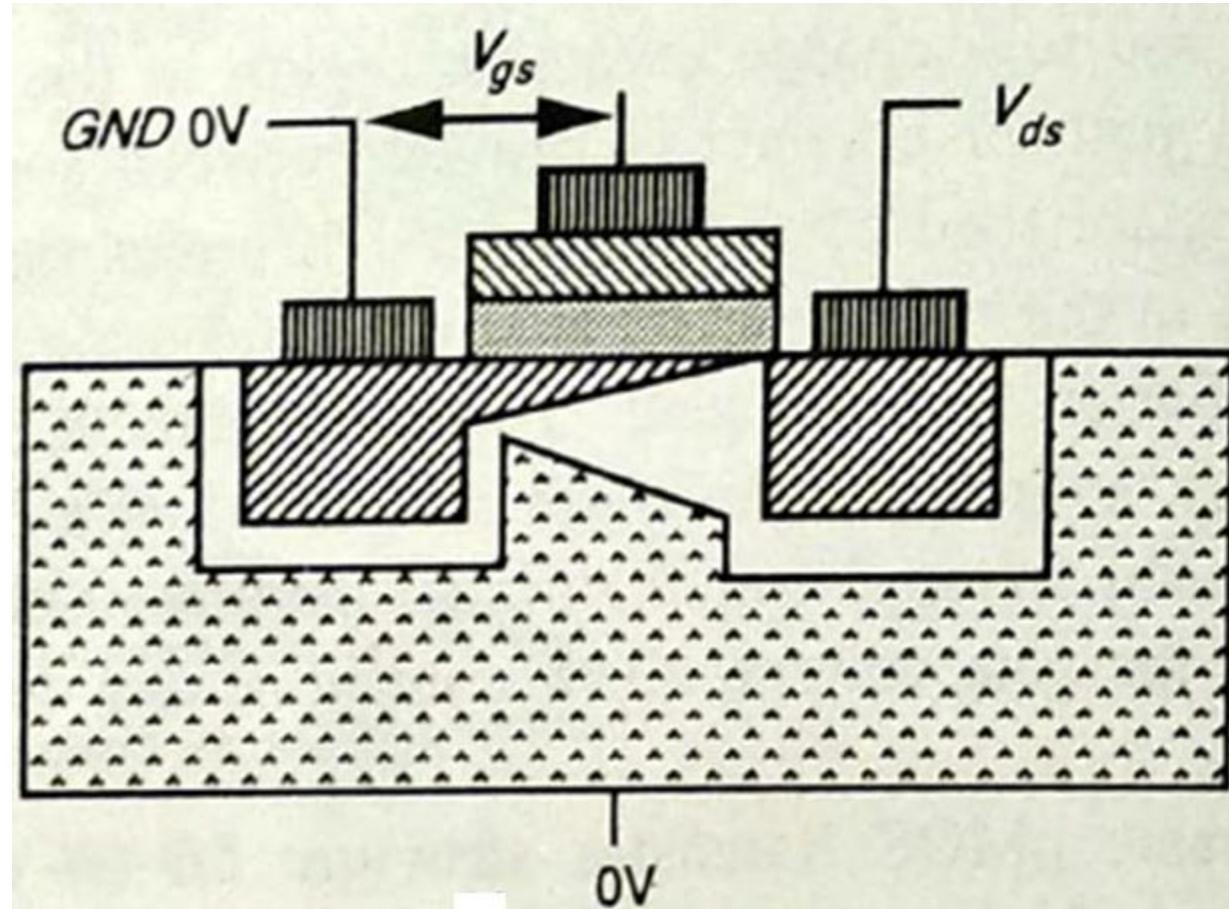
Enhancement mode nMOS Transistor operation

Three sets of conditions

(i) $V_{gs} > V_t$ and $V_{ds} = 0V$

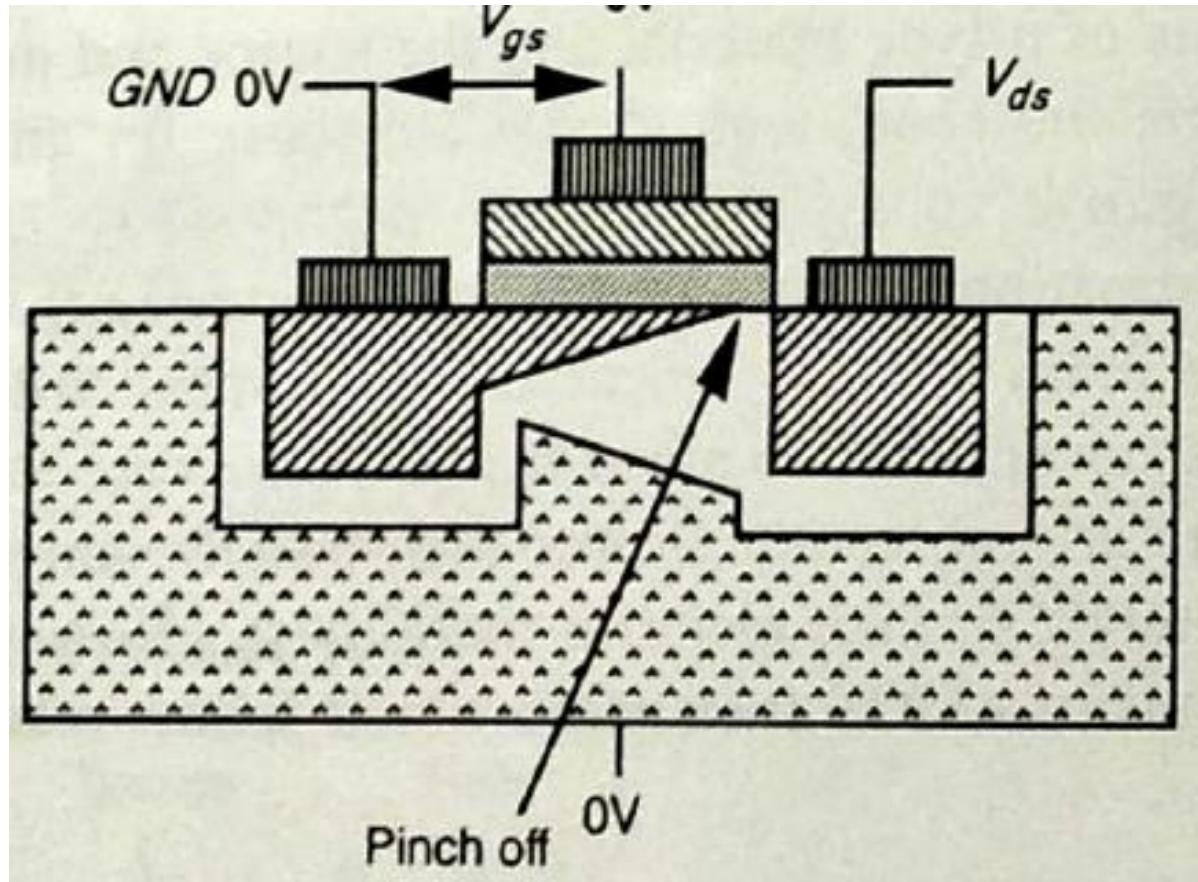


(ii) $V_{gs} > V_t$ and $V_{ds} < V_{gs} - V_t$



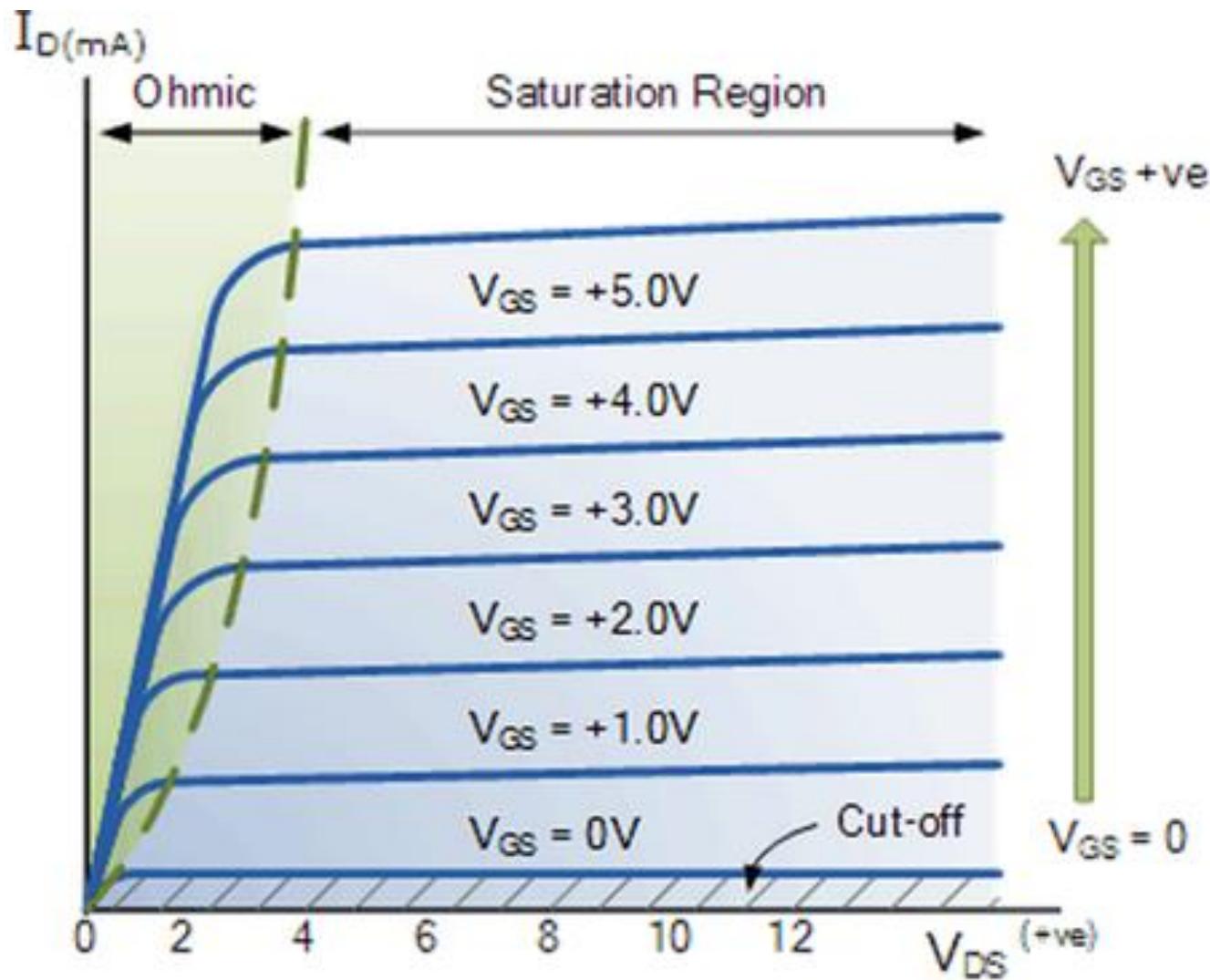
- Channel width decreases in drain side
- For all voltages, $V_{ds} < V_{gs} - V_t \Rightarrow$ **Non saturation region of Operation**

(iii) $V_{gs} > V_t$ and $V_{ds} > V_{gs} - V_t$

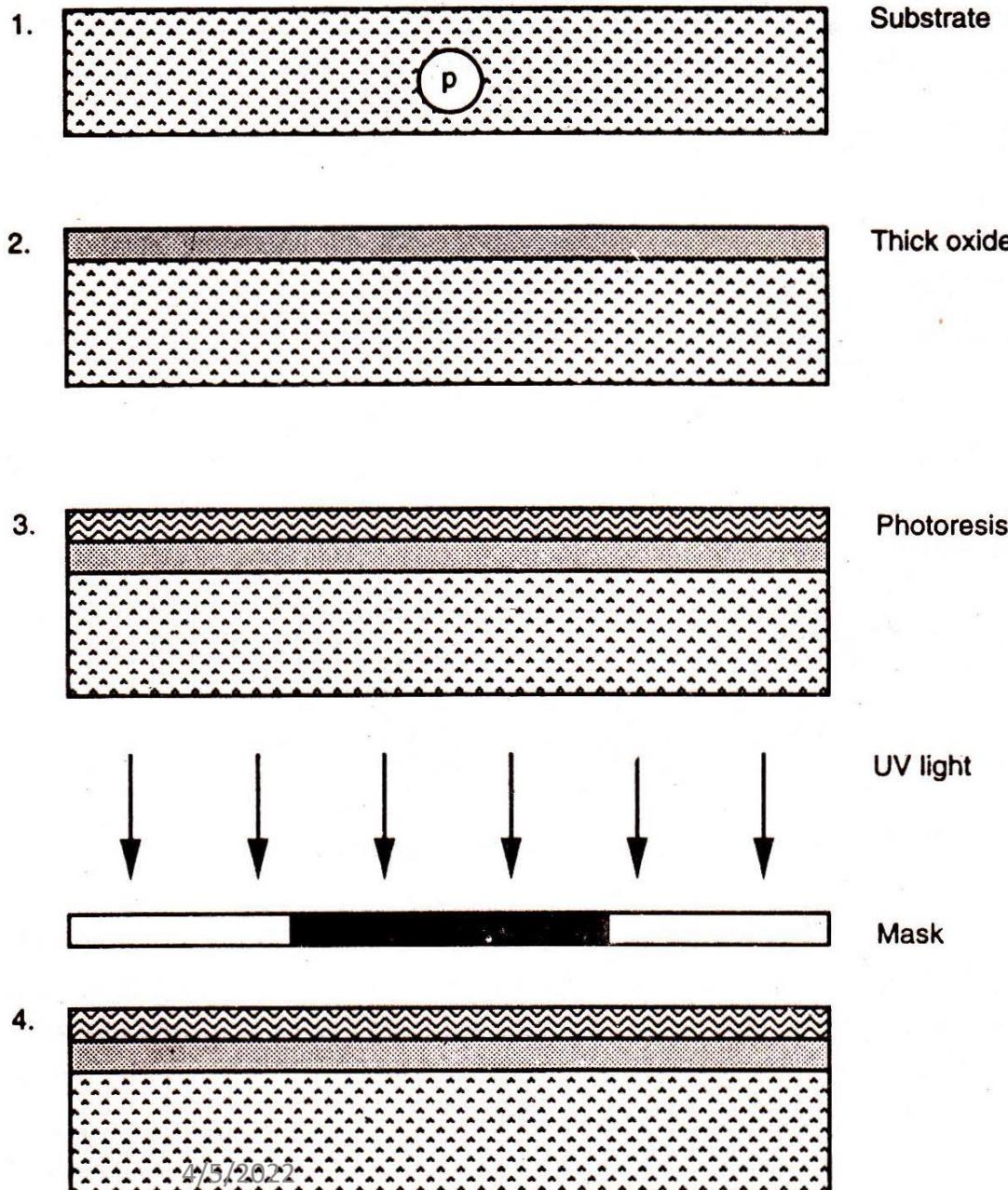


- Channel is Pinch off
- For all voltages above $V_{ds} = V_{gs} - V_t \Rightarrow$ **Saturation region of Operation**

Current Voltage (I-V) characteristics:

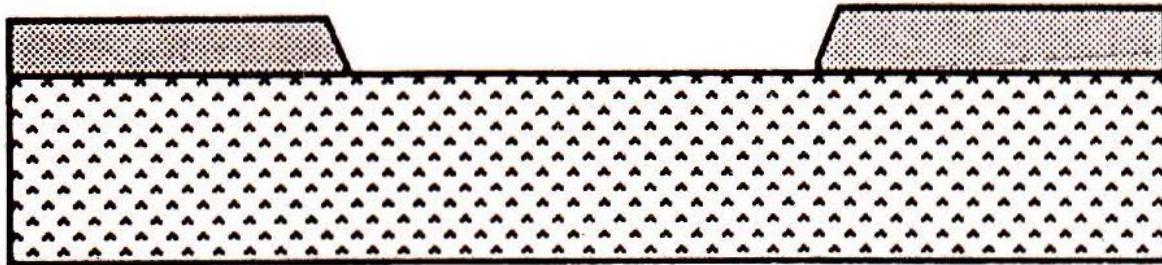


nMOS FABRICATION



- 1) Processing is carried out on a thin wafer cut from a single crystal of silicon of high purity into which the required **p-impurities are introduced** as the crystal is grown.
- 2) A **layer of silicon dioxide (SiO_2) is grown all over the surface** of the wafer to protect the surface, **act as a barrier** to dopants during processing, and provide a generally insulating substrate on to which other layers may be deposited and patterned.
- 3) The surface is now **covered with a photoresist** which is deposited onto the wafer and spun **to achieve an even distribution** of the required thickness.
- 4) The photoresist layer is then **exposed to ultraviolet light through a mask** which defines **those regions** into which **diffusion is to take place** together with transistor channels.

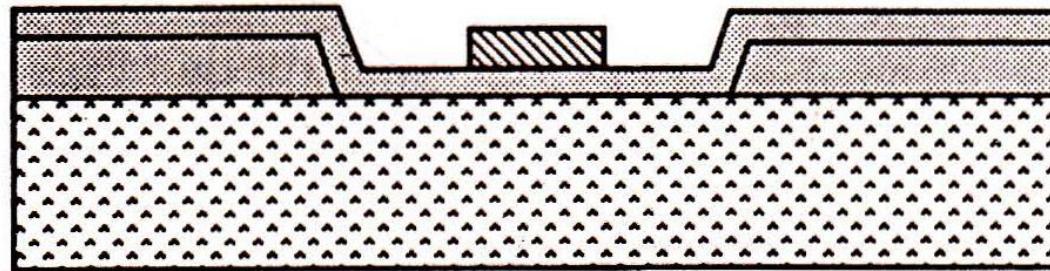
5.



Window in oxide

5. These areas are subsequently **readily etched away** together with the **underlying silicon dioxide** so that the wafer surface is exposed in the window defined by the mask.

6.



Patterned poly. (1–2 μm)
on thin oxide (800–1000 Å)

6.

- **remaining photoresist is removed**
- **thin layer of SiO₂ is grown over the entire chip surface**
- **polysilicon** is deposited on top of this **to form the gate structure**. The polysilicon layer consists of heavily doped polysilicon deposited by **chemical vapor deposition (CVD)**.

7.

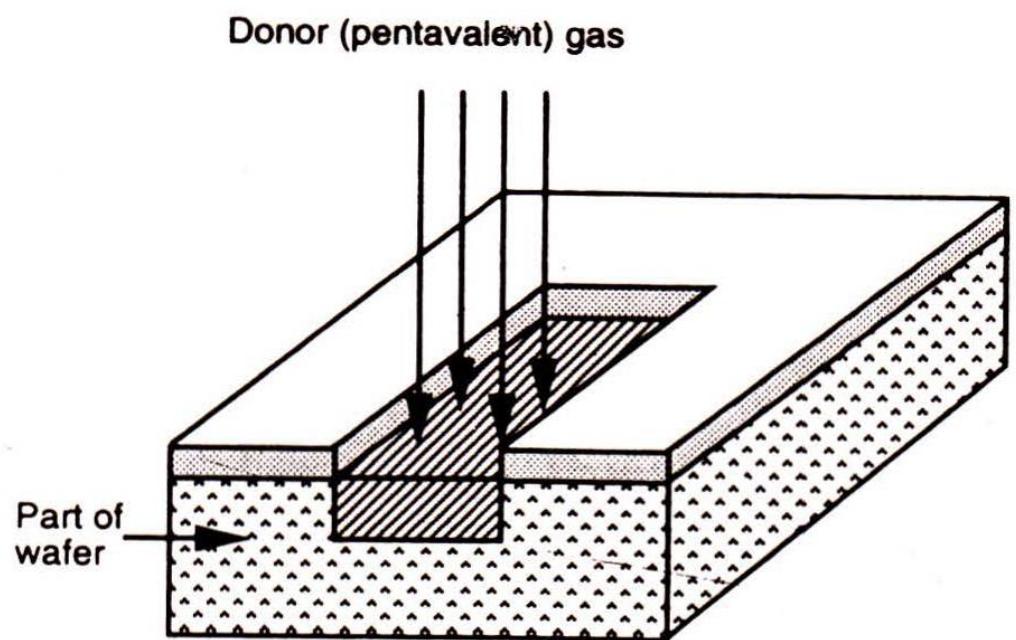
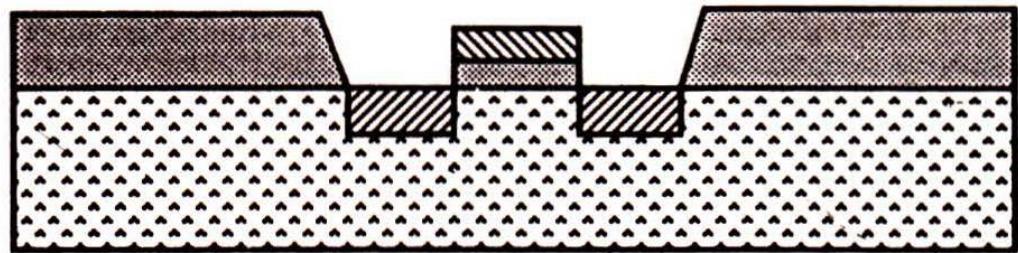
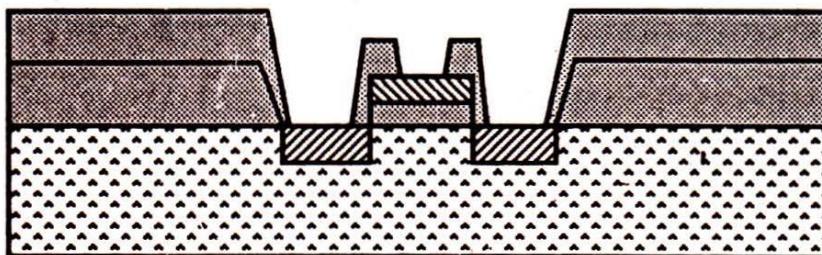


FIGURE 1.8 Diffusion process.

7.

- **photoresist coating and masking allows the polysilicon to be patterned** (as shown in Step 6)
- **thin oxide is removed** to expose areas into which **n-type impurities are to be diffused to form the source and drain**.
- **Diffusion is achieved by heating the wafer** to a high temperature and **passing a gas containing the desired n-type impurity** (for example, phosphorus) over the surface as indicated in Figure.
- Note that the **polysilicon with underlying thin oxide act as masks during diffusion--the process is self-aligning**.

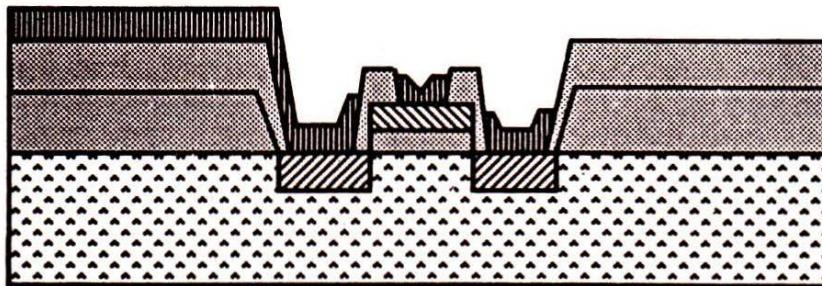
8.



Contact holes (cuts)

8. **Thick oxide (SiO_2) is grown over all again** and is then masked with photoresist and etched to expose selected areas of the polysilicon gate and the drain and source areas **where connections** (i.e. contact cuts) are to be made.

9.

Patterned metallization
(aluminum 1 μm)

9. The whole chip then has **metal (aluminum) deposited over its surface** to a thickness. This metal layer is then masked and etched to form the required interconnection pattern.

Summary of An nMOS Process

- Processing takes place on a p-doped silicon crystal wafer on which is grown a 'thick' layer of SiO₂.

Mask 1-Pattern SiO₂ to expose the silicon surface in areas where paths in the diffusion layer or gate areas of transistors are required. Deposit thin oxide over all. For this reason, this mask is often known as the '**'thinox'**' mask but some texts refer to it as the diffusion mask.

Mask 2-Pattern the ion implantation within the thinox region where depletion mode devices are to be produced-self-aligning.

Mask 3-Deposit polysilicon over all, then pattern using Mask 3. Using the same mask, remove thin oxide layer where it is not covered by polysilicon.

- Diffuse n + regions into areas where thin oxide has been removed. Transistor drains and sources are thus self-aligning with respect to the gate structures.

Mask 4-Grow thick oxide over all and then etch for contact cuts.

Mask 5-Deposit metal and pattern with Mask 5

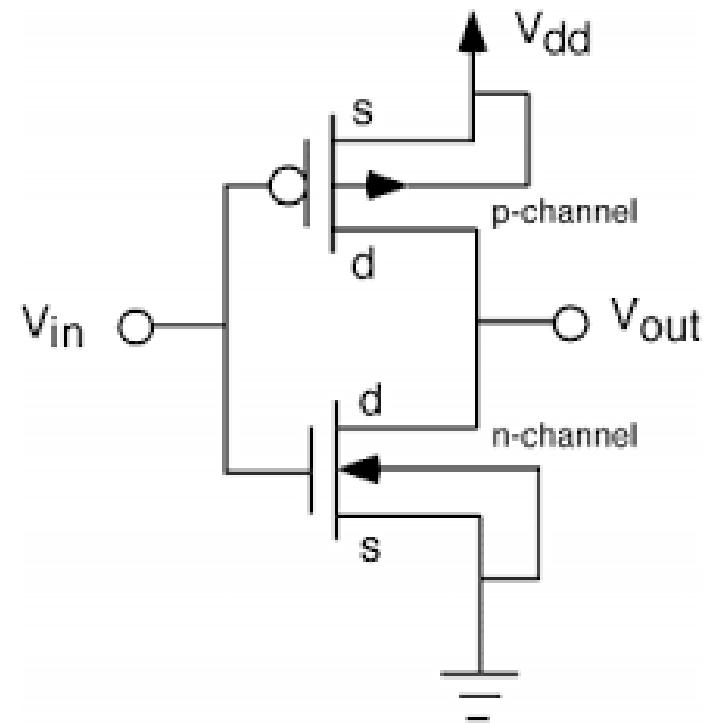
Mask 6-Would be required for the overglossing process step.

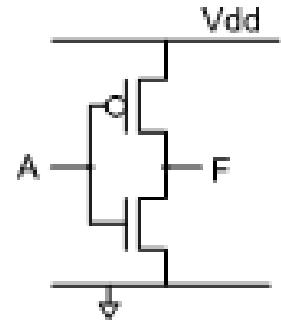
CMOS Technology

- One important MOS technology
- Consider both n-channel and p-channel MOSFETs
- high noise immunity and low power consumption.

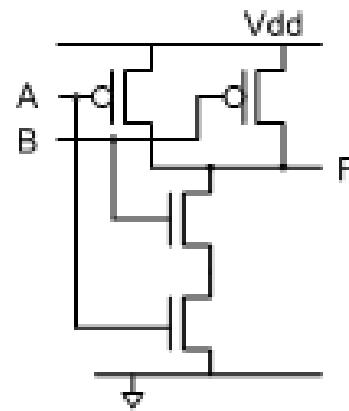
Fabrication Techniques:

1. P-well structure
2. N-well structure
3. Twin-well structure

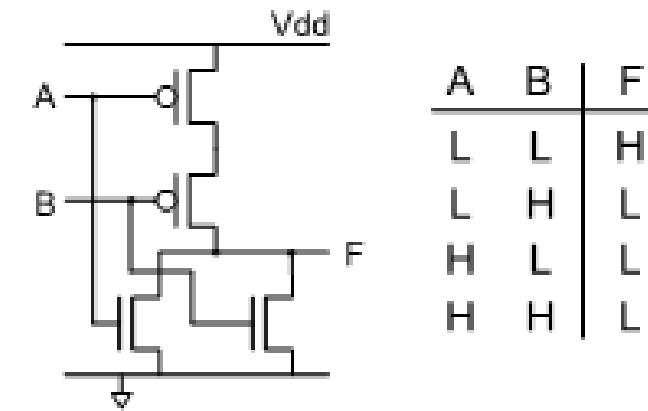




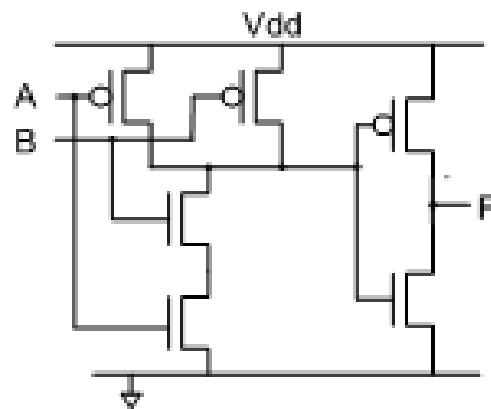
	A	F
A	L	H
	H	L



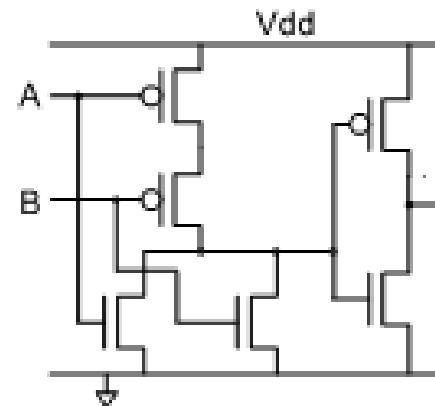
A	B	F
L	L	H
L	H	H
H	L	H
H	H	L



A	B	F
L	L	H
L	H	L
H	L	L
H	H	L



A	B	F
L	L	L
L	H	L
H	L	L
H	H	H



A	B	F
L	L	L
L	H	H
H	L	H
H	H	H

The p-well Process

- The structure consists of an **n-type substrate** in which **p-devices may be formed** by suitable masking and diffusion and, in order to accommodate **n-type devices, a deep p-well is diffused** into the n-type substrate as shown.
- This diffusion must be carried out with special care since the **p-well doping concentration and depth will affect the threshold voltages as well as the breakdown voltages** of the n-transistors.
- To **achieve low threshold voltages** (0 .6 to 1.0 V) we need **either deep-well diffusion or high-well resistivity**. However, deep wells require larger spacing between the n- and p-type transistors and wires due to lateral diffusion and therefore a larger chip area.
- The p-wells act as substrates for the n-devices within the parent n-substrate, **the two areas are electrically isolated**. However, since there are now **in effect two substrates, two substrate connections (VDD and Vss) are required**, as shown in Figure 1.1 0.

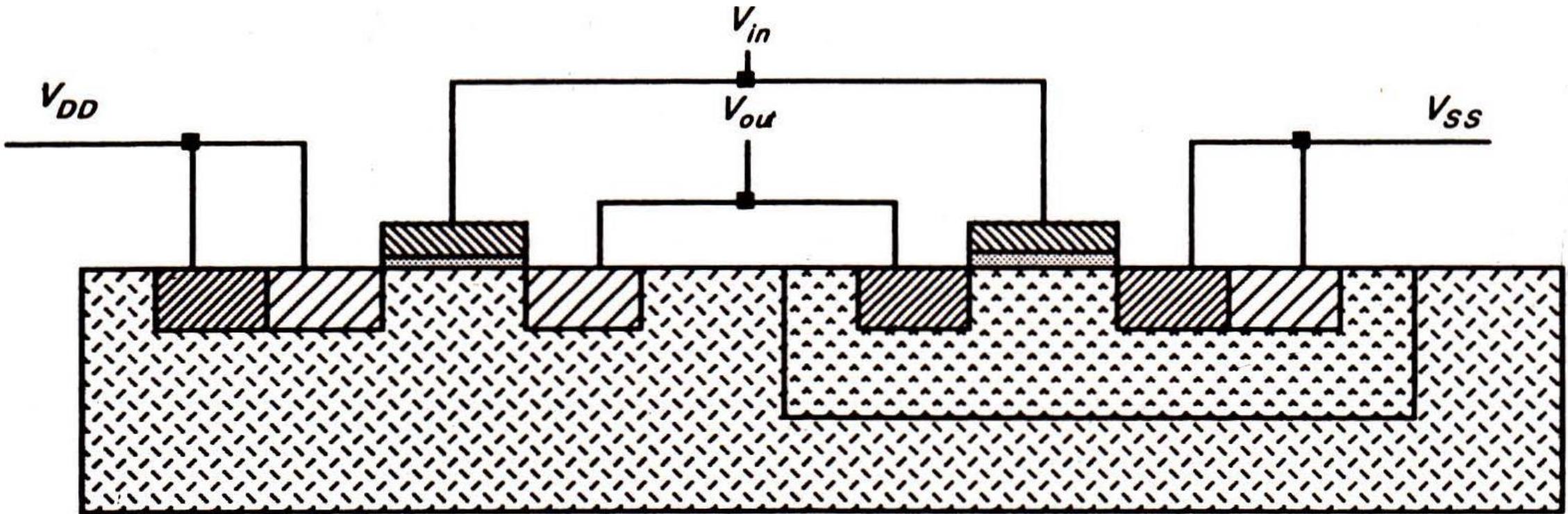
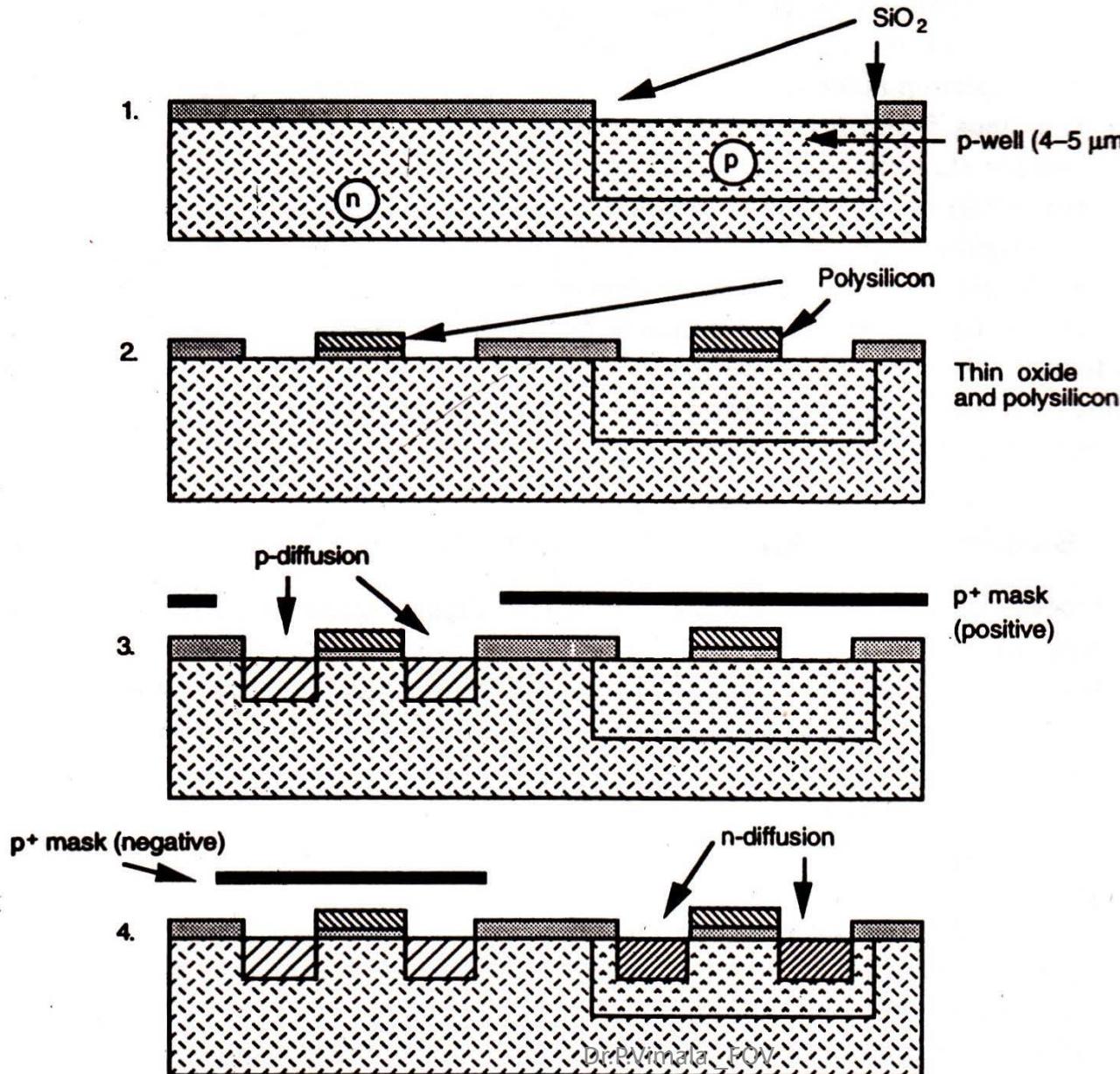


FIGURE 1.10 CMOS p-well inverter showing V_{DD} and V_{SS} substrate connections.

A brief overview of the fabrication steps may be obtained with reference to Figure, noting that the basic processing steps are of the same nature as those used for nMOS.



In all other respects-masking, patterning, and diffusion-the process is similar to nMOS fabrication.

In summary, typical processing steps are:

- **Mask 1** - **defines the areas** in which the deep **p-well diffusions** are to take place.
- **Mask 2** - defines the thinox regions, namely those areas where the **thick oxide is to be stripped and thin oxide grown to accommodate p- and n-transistors and wires**.
- **Mask 3** - **used to pattern the polysilicon layer** which is deposited after the thin oxide.
- **Mask 4** - **A p-plus mask is now used** (to be in effect "Anded" with Mask 2) to define all areas where **p-diffusion** is to take place.
- **Mask 5** - This is usually performed using **the negative form of the p-plus mask** and defines those areas where **n-type diffusion** is to take place.
- **Mask 6** - **Contact cuts** are now defined.
- **Mask 7** - The **metal layer pattern** is defined by this mask.
- **Mask 8** - An overall passivation (overglass) layer is now applied and Mask 8 is needed to define the openings for access to bonding pads.

The n-well Process

- p-well process is widely used, **n-well fabrication** has also gained wide acceptance, **initially as a retrofit to nMOS lines.**
- **N-well CMOS circuits are also superior** to p-well because of the **lower substrate bias effects** on transistor threshold voltage and inherently **lower parasitic capacitances** associated with source and drain regions.

n-well fabrication steps:

- The **first mask defines the n-well regions**. This is followed by a low dose phosphorus implant driven in by a **high temperature diffusion step to form the n-wells**. The well depth is optimized to ensure against p-substrate to p+ diffusion breakdown without compromising the n-well to n+ mask separation.
- The **next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the polysilicon, carry out the diffusions, make contact cuts, and finally metalize as before.**

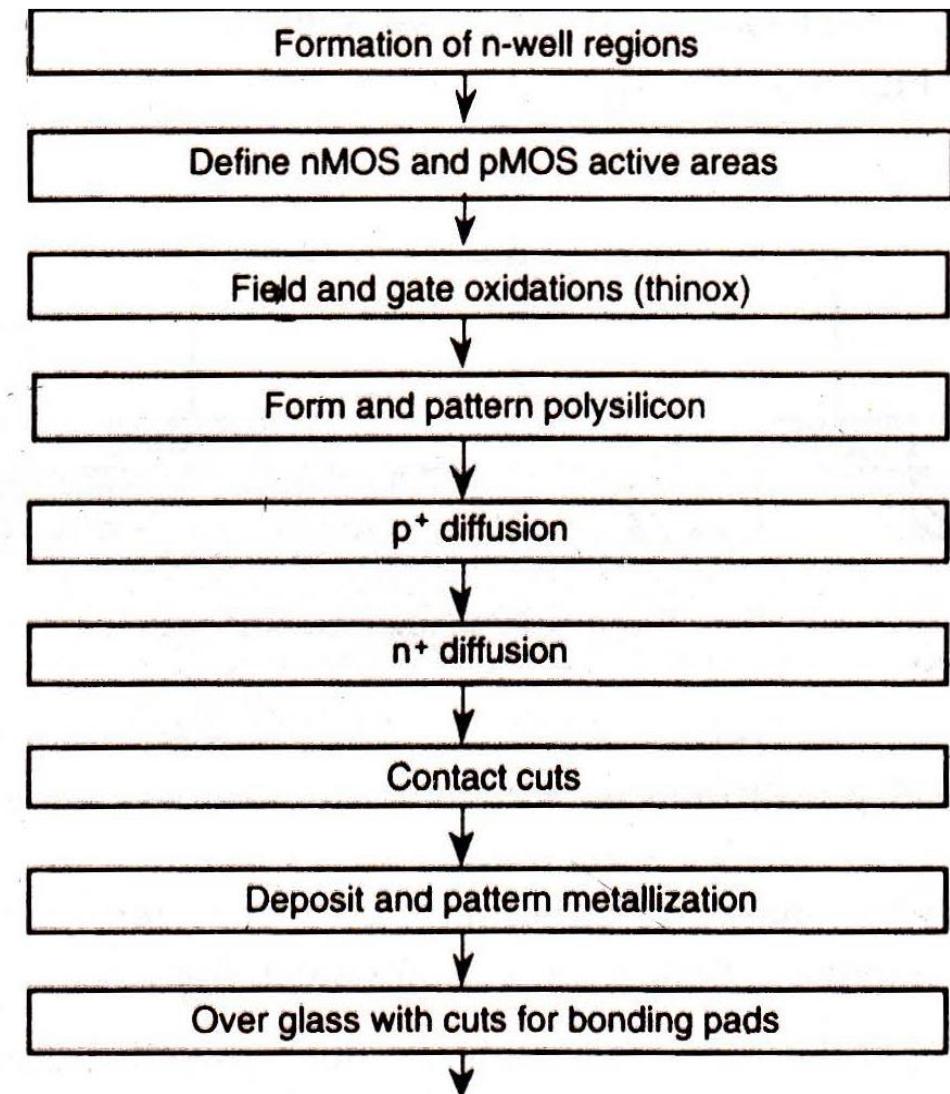


FIGURE 1.11 Main steps in a typical n-well process.

- It will be seen that an **n+ mask and its complement** may be used to define the n- and p-diffusion regions respectively. These same masks **also include the V_{DD} and V_{SS} contacts** (respectively). It should be noted that, alternatively, we could have used a p+ mask and its complement; since the n+ and p+ masks are generally complementary.

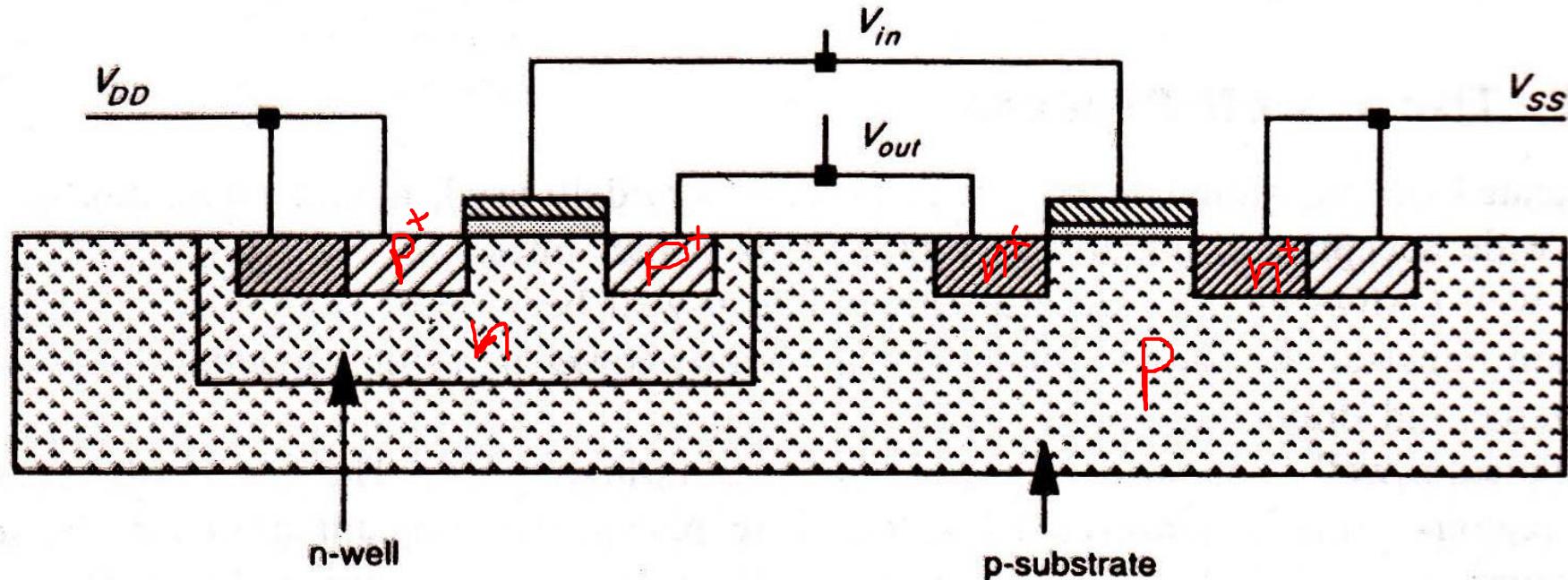


FIGURE 1.12 Cross-sectional view of n-well CMOS inverter.

- Latch-up problems can be considerably reduced.** However, a factor of the n-well process is that the **performance of the already poorly performing p-transistor** is even further degraded.

The Twin-Tub Process

- A **logical extension of the p-well and n-well approaches** is the twin-tub fabrication process.
- Here we start with a **substrate of high resistivity n-type material** and then **create both n-well and p-well regions**. Through this process it is possible to preserve the performance of n-transistors without compromising the p-transistors.
- **Doping control is more readily achieved** and some relaxation in manufacturing tolerances results.
- This is particularly **important as far as latch-up** is concerned.
- In general, the twin-tub process **allows separate optimization of the n- and p-transistors**.

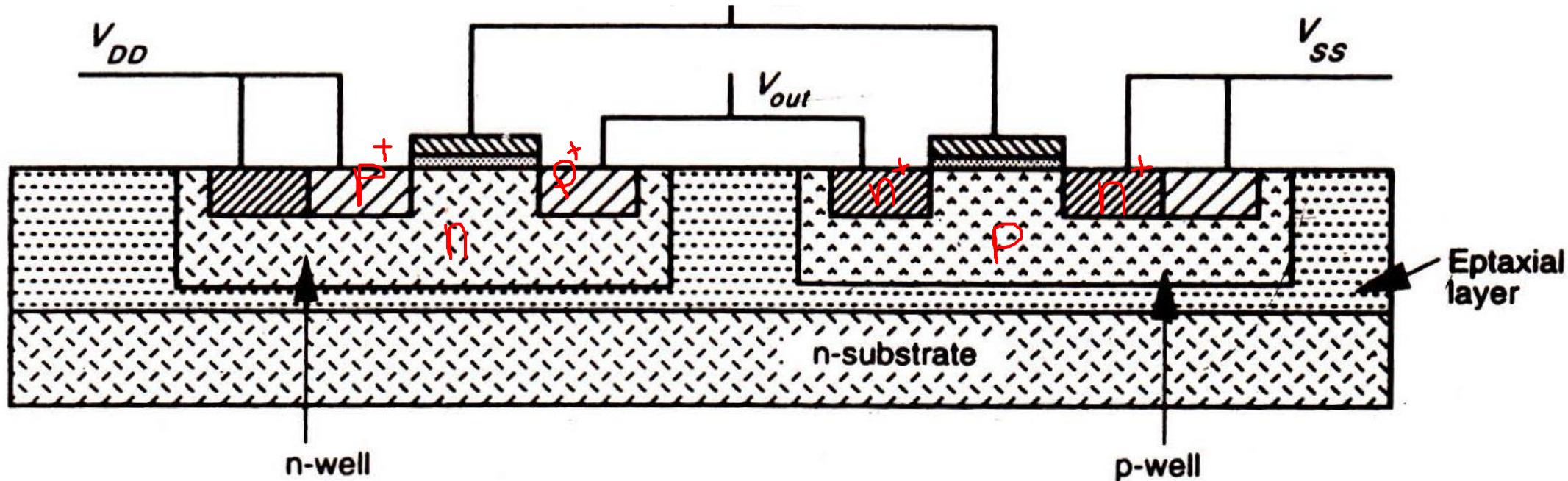


FIGURE 1.14 Twin-tub structure.

Thermal Aspects of Processing

- The processes involved in making nMOS and CMOS devices have **differing high temperature sequences**.
- The CMOS p-well process, for example, has a **high temperature p-well diffusion process** (1100 to 1250°C), the **nMOS process having no such requirement**.
- Because of the **simplicity, ease of fabrication, and high density per unit area of nMOS circuits**, many of the earlier IC designs, still in current use, have been fabricated using nMOS technology.
- nMOS and CMOS system designs will continue to co-exist** for some time to come.

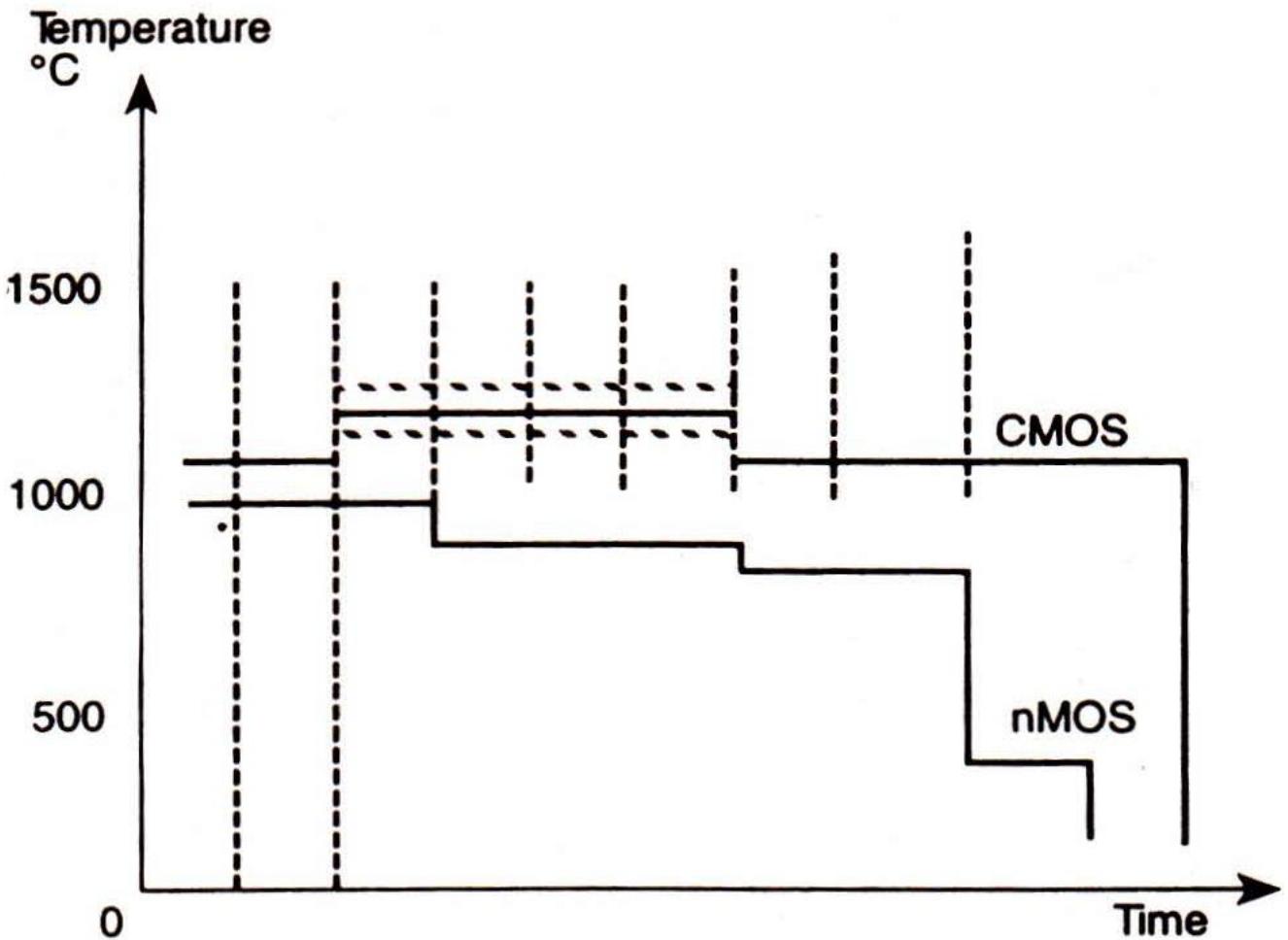


FIGURE 1.15 Thermal sequence difference between nMOS and CMOS processes.

BiCMOS TECHNOLOGY

TABLE 1.2 Comparison between CMOS and bipolar technologies

- Deficiency of MOS technology -limited load driving capabilities
- BiCMOS- integrates BJT and CMOS technology.
- It provides:
 - **High Gain**
 - **High driving Capability**
 - **Better noise and high frequency**

<i>CMOS technology</i>	<i>Bipolar technology</i>
<ul style="list-style-type: none">• Low static power dissipation• High input impedance (low drive current)• Scalable threshold voltage• High noise margin• High packing density• High delay sensitivity to load (fan-out limitations)• Low output drive current• Low g_m ($g_m \propto V_{in}$)• Bidirectional capability (drain and source are interchangeable)• A near ideal switching device	<ul style="list-style-type: none">• High power dissipation• Low input impedance (high drive current)• Low voltage swing logic• Low packing density• Low delay sensitivity to load• High output drive current• High g_m ($g_m \propto e^{V_{in}}$)• High f_t at low currents• Essentially unidirectional

Logic Design with MOSFETs: Ideal switches and Boolean operations, MOSFETs as switches, Basic logic gates in CMOS, Complex logic gates in CMOS, Transmission gates

Ideal switches and Boolean operations

- Logic Gates are created by using sets of **Controlled switches**
- **Two types:** Assert-high and Assert-low

Assert-High Switches:

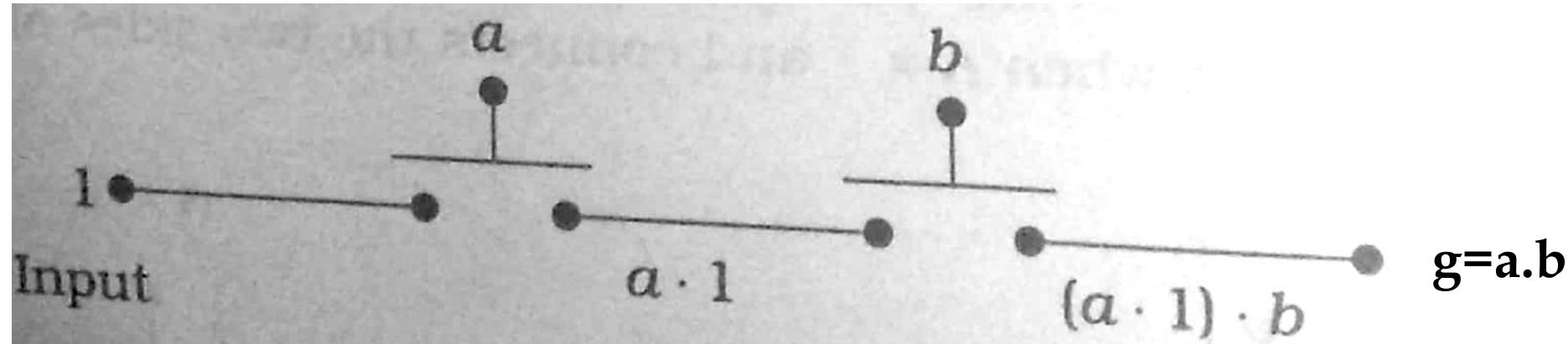


(a) Open

(b) Closed

$$y = x \cdot A \quad \text{iff} \quad A = 1$$

- Create a logic network by combining the concept of ideal switches,
- 1) Two switches are **in series**

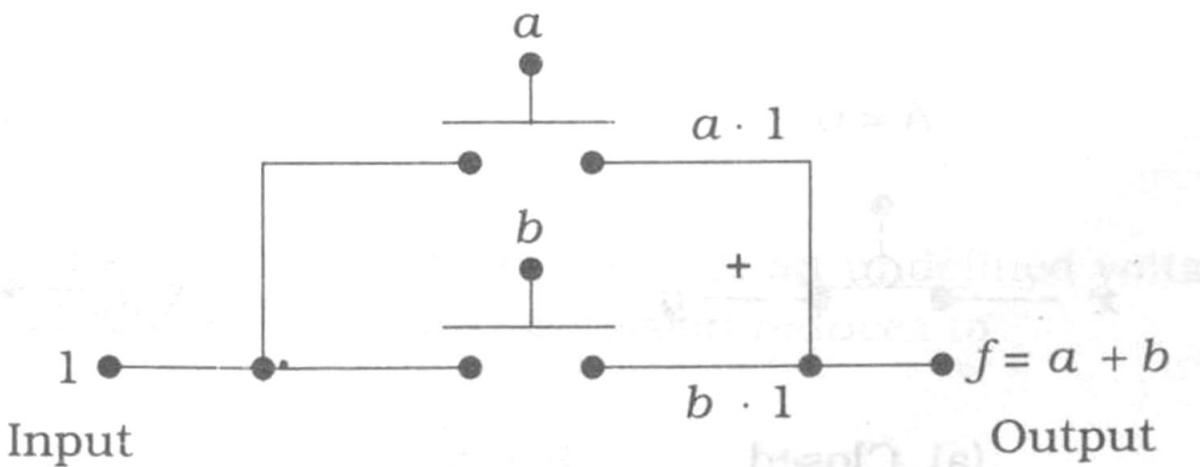


$$g = (a \cdot 1) \cdot b = a \cdot b$$

- $(a,b) = (1,1) ; g=1$
- $(a,b) = (1,0), (0,1), (0,0) ; g=0,$
but logic equations says g -undefined

AND Operation

2) Two switches are **in Parallel**

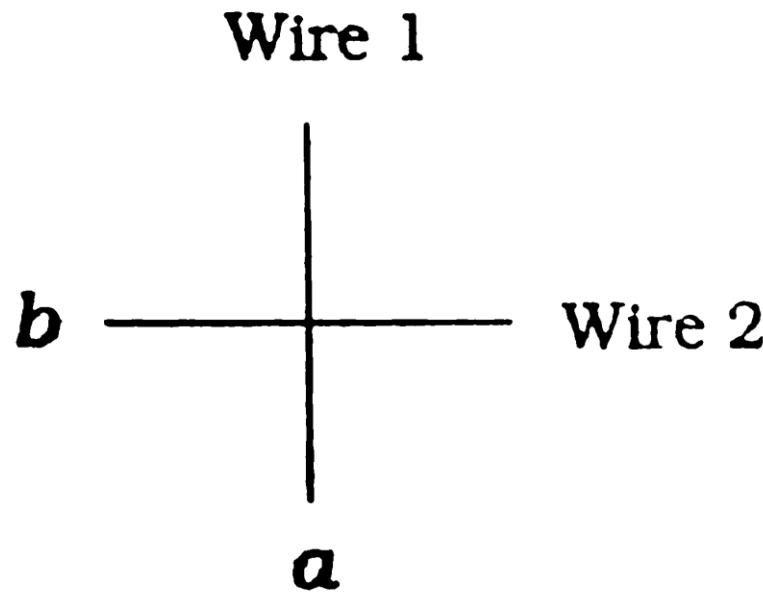


- $(a,b) = (1,1), (1,0), (0,1) \quad f=1$
- $(a,b) = (0,0) ; \quad f=0$

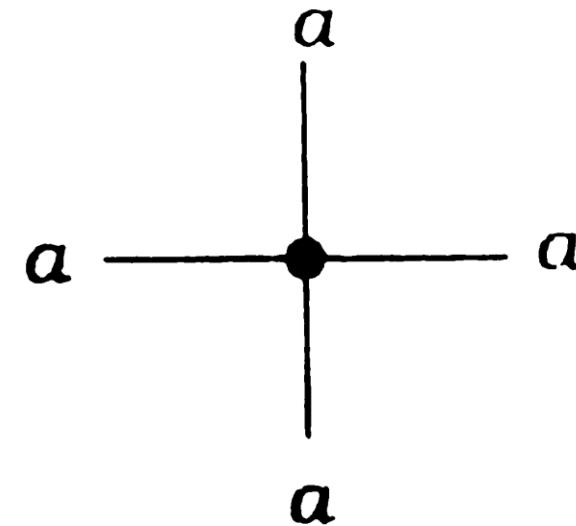
OR Operation

but logic equations says f-undefined

- Switch drawings will be called as **Schematic Diagrams**
- They shows the **scheme used in the wiring**



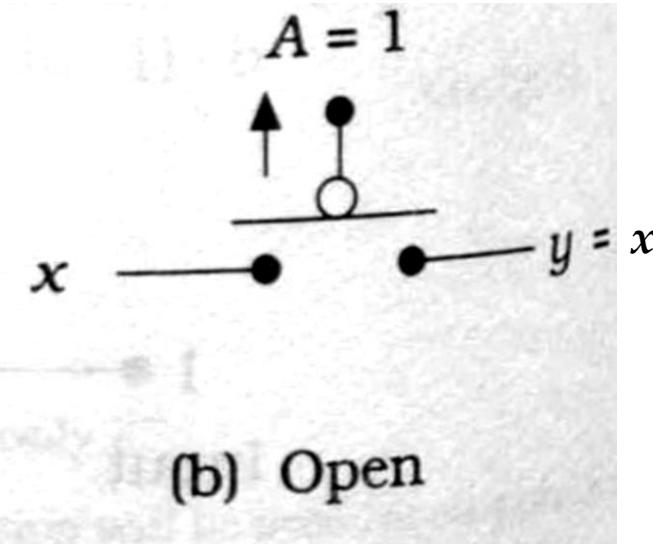
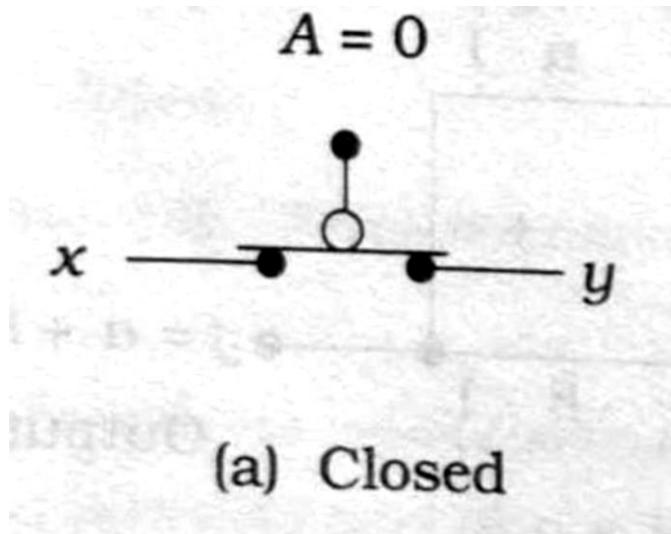
(a) No connection



(b) Connection

Assert-Low Switches

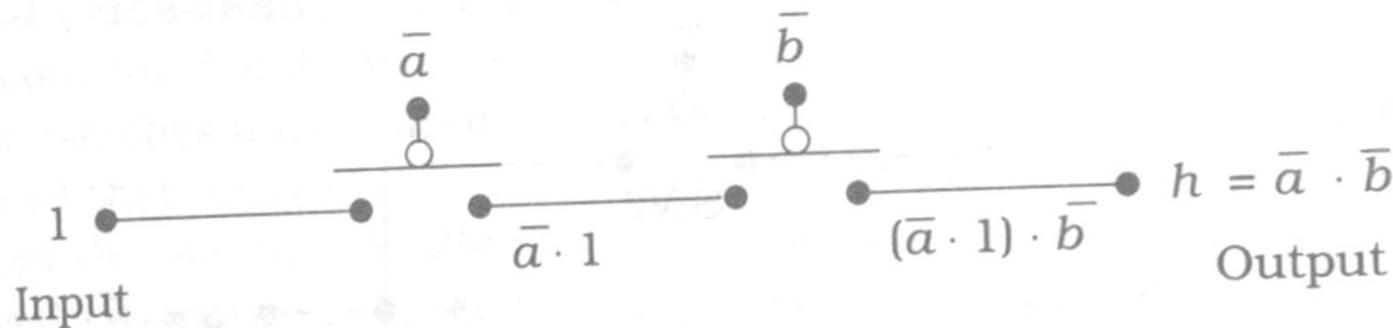
- Added a logic “bubble” to the top of the symbol to distinguish from assert-high switch.



$$y = x \cdot \bar{A} \quad \text{iff } A = 0$$

Create a logic network by combining the concept of ideal switches,

- Two switches are **in series**



$$h = (\bar{a} \cdot 1) \cdot \bar{b}$$

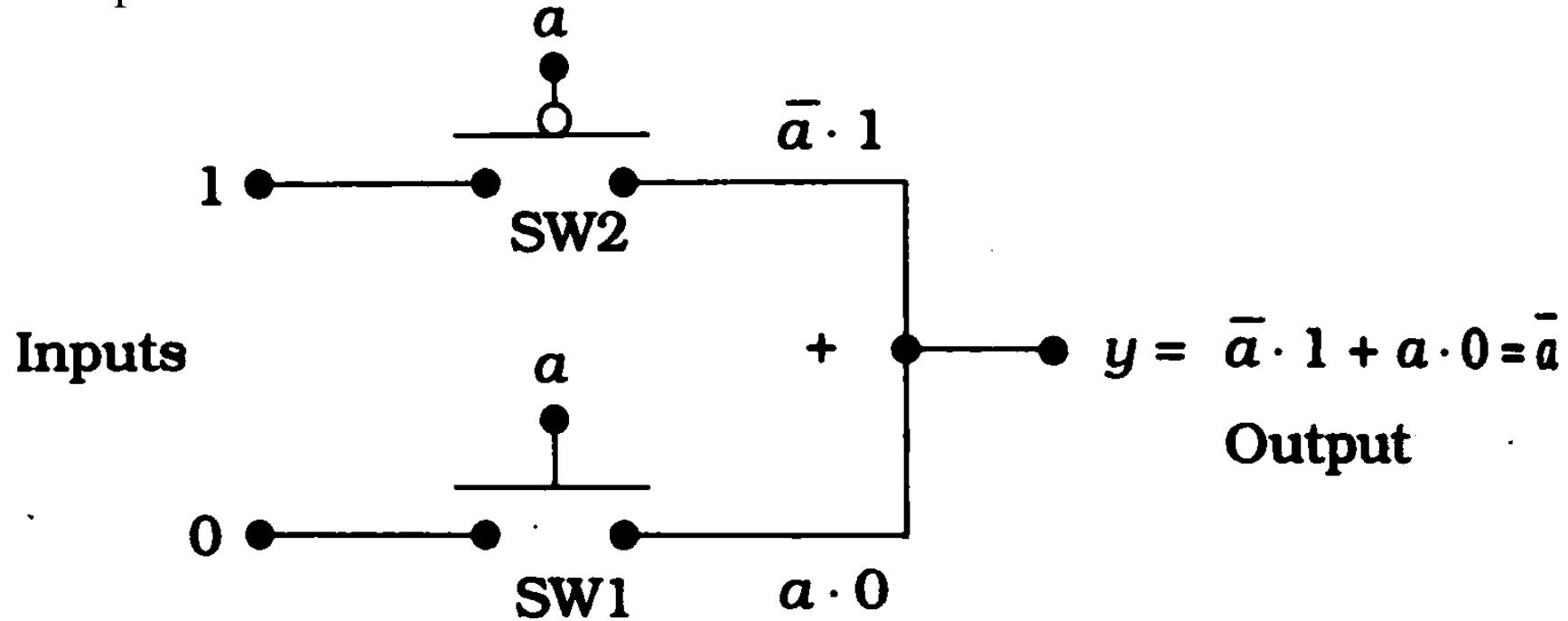
$$= \overline{a + b}$$

- $(a,b) = (0,0) ; h=1$
- $(a,b) = (1,0), (0,1), (1,1); h=0,$
but logic equations says h-undefined

NOR Operation

Both type of switches in single network:

- provide both logic 1 and logic 0 inputs in an effort to produce an output that is defined for all possible input combinations.

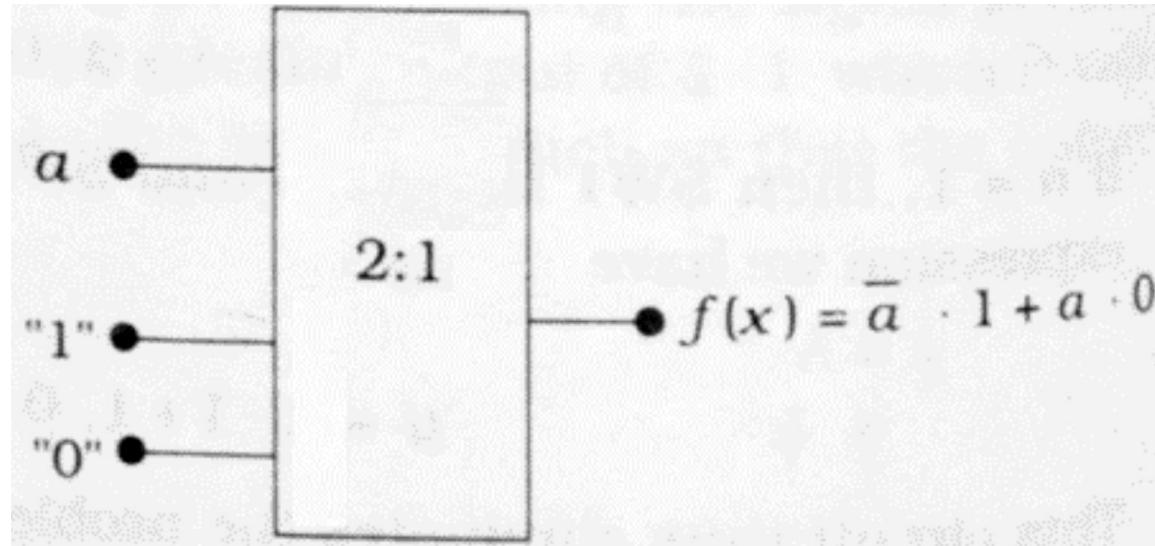


- If $a=0$, $y = \bar{0} \cdot 1 + 0 \cdot 0 = 1$

$$y = \text{NOT}(a) = \bar{a}$$

- If $a=1$, $y = \bar{1} \cdot 1 + 1 \cdot 0 = 0$

NOT circuit is based on the behaviour of 2:1 Mux



$$y = \bar{a} \cdot 1 + a \cdot 0$$

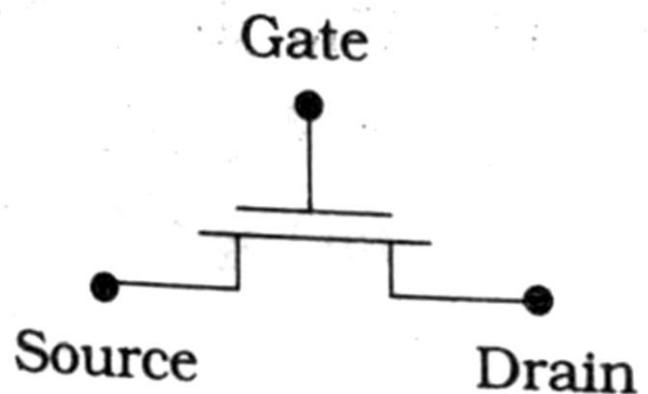
MOSFETs as Switches

Difference compare to idealized switches are

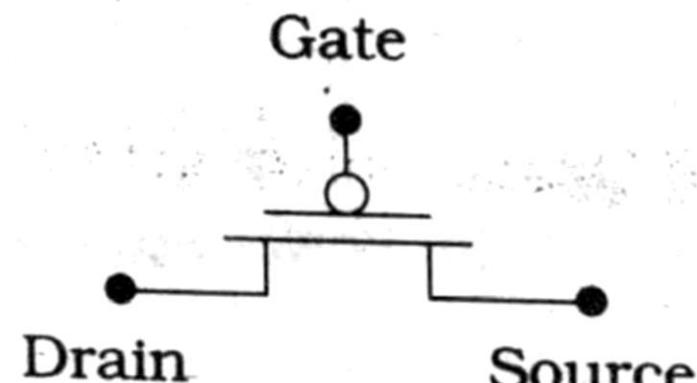
- must obey circuit equations
- Their ultimate performance is limited by the laws of physics.

CMOS uses two types of MOSFETs

- nFET
- pFET

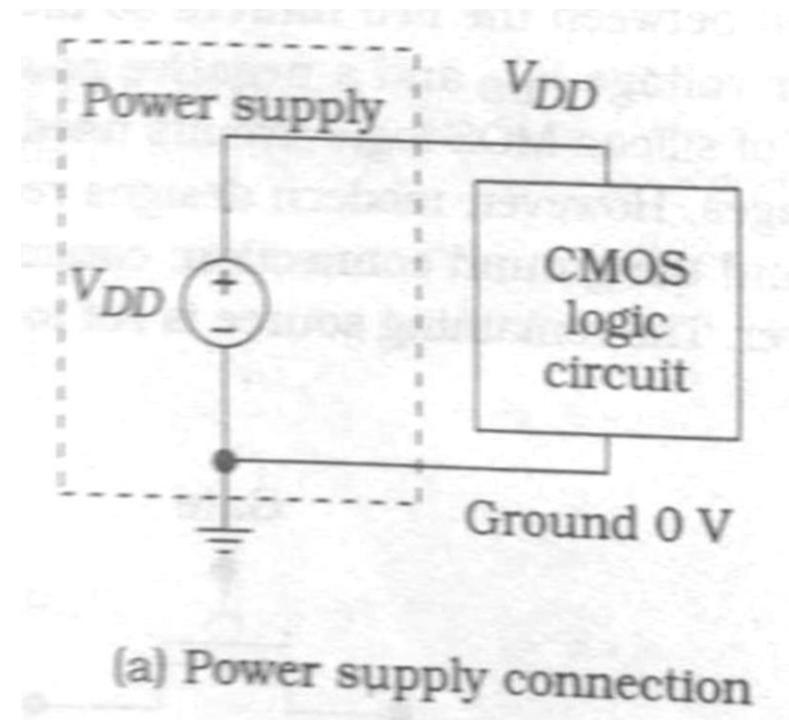
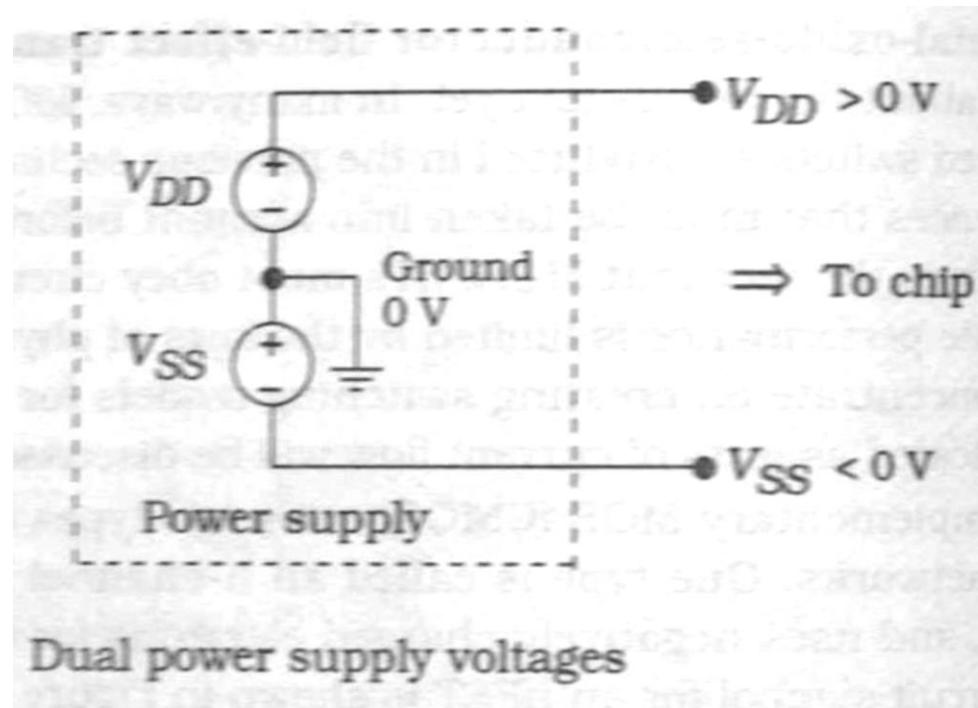


(a) nFET symbol



(b) pFET symbol

- How to translate between Boolean values and electrical parameters
- Two power supply voltages V_{DD} and V_{SS} are defined



- Modern designs require only single positive voltage, V_{DD} and Ground.
- V_{DD} ranges from 5 V to 3.3 V

- Now we can define the relationship between logic variables and voltages.
- Boolean variables are discrete and carry values $x = 0$ or $x = 1$
- At the circuit level **variable x using a voltage V_x**

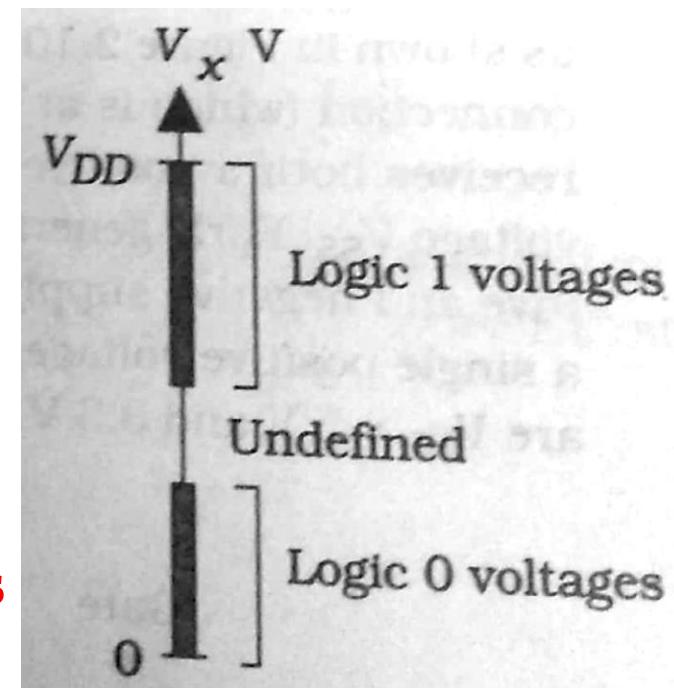
$$0 \leq V_x \leq V_{DD}$$

- The **positive logic convention** defines the ideal logic 0 and logic 1 voltages as,

$x = 0$ means that $V_x = 0$ V

$x = 1$ means that $V_x = V_{DD}$

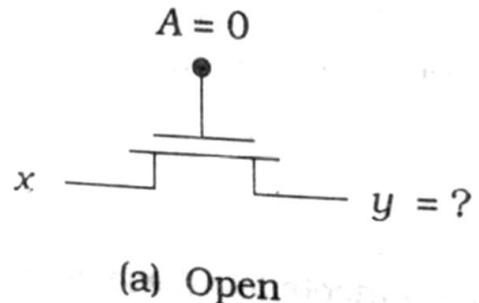
- Low voltages correspond to logic 0 values
- High voltages correspond to logic 1 values
- The actual extent of both **voltage ranges is determined by the characteristics of the logic circuits**



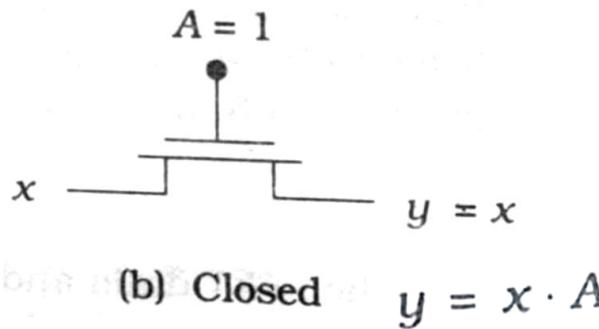
Switching characteristics of MOSFETs

A is the logic variable applied to the gate.

- nFET behaves like an assert-high switch.



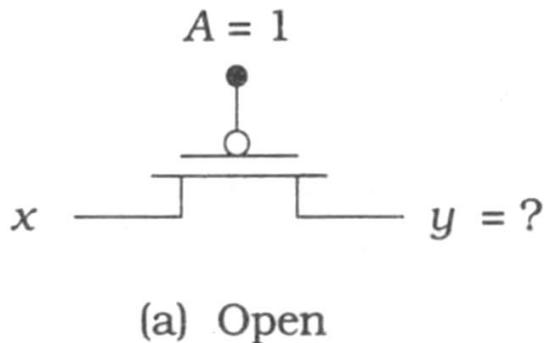
(a) Open



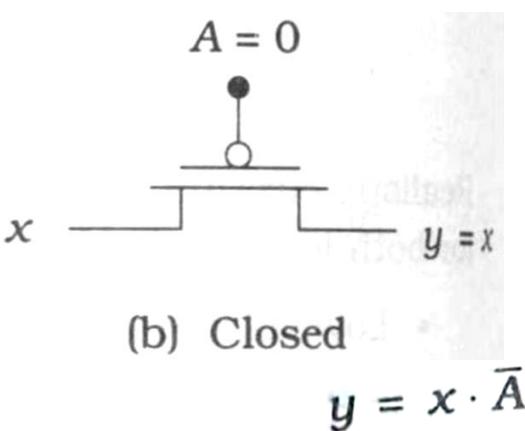
(b) Closed

- MOSFETs allow us to design logic circuits using the technique assert-high and assert-low switching networks.

- pFET behaves like an assert-low switch.



(a) Open



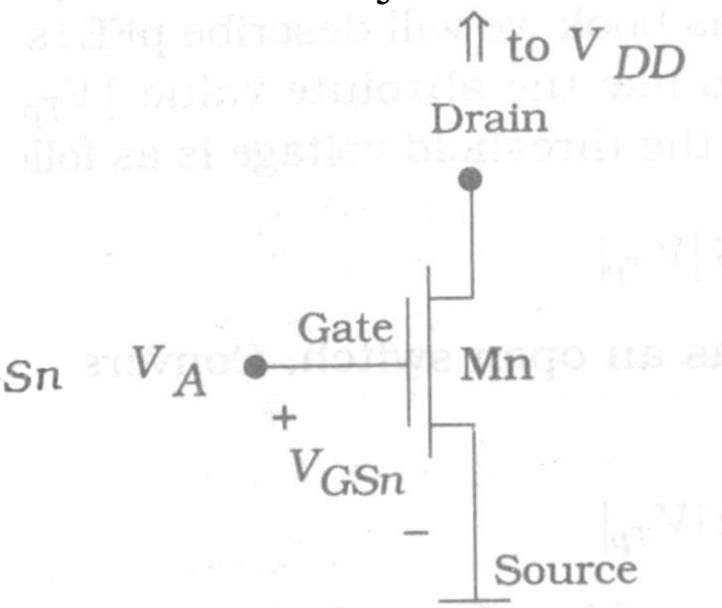
- However, FETs are physical devices that do not behave exactly like the ideal switch models

FET Threshold Voltages

- The switching equations assume that the binary variable A applied to the gate of a FET is either 0 or 1.
- Define a range of voltages for the control signal (V_A)
- Threshold voltage helps to define the gate voltage ranges

nFET:

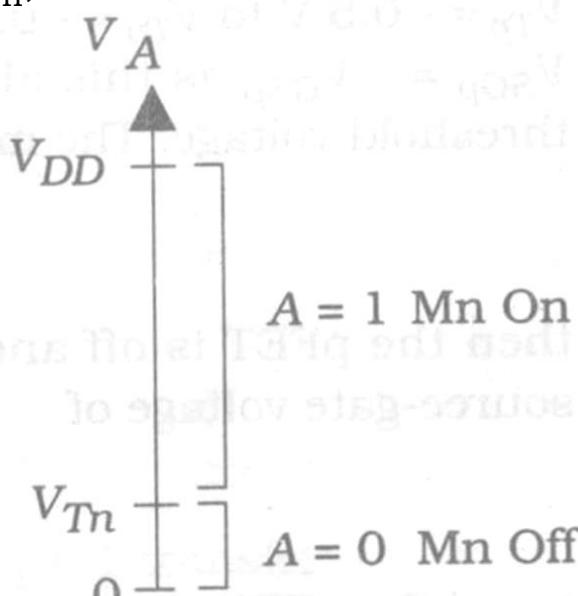
nFET is characterized by a threshold voltage, V_{Tn} , values around 0.5 V to 0.7 V.



When,

$$A=0; V_A \leq V_{Tn}$$

$$A=1; V_A \geq V_{Tn}$$



(b) Logic translation

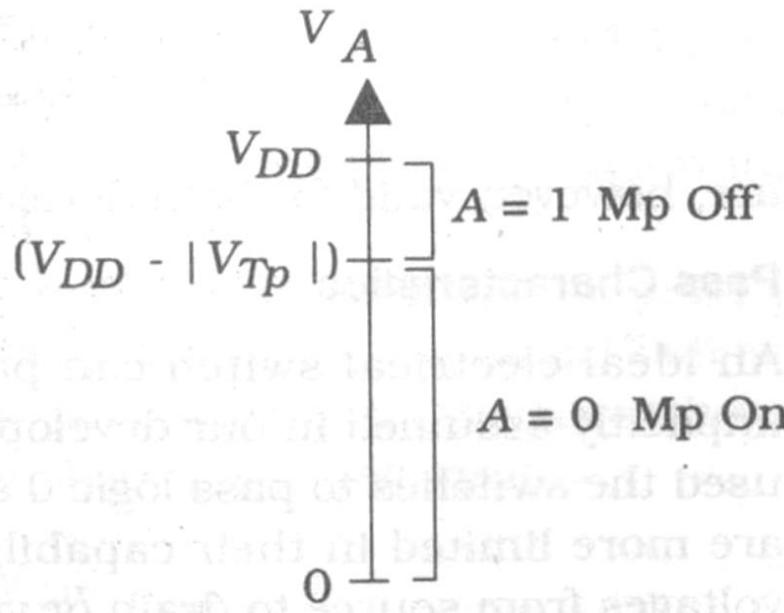
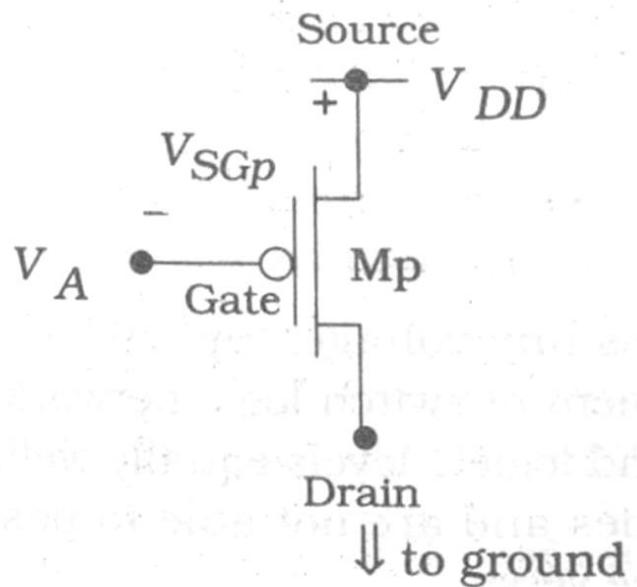
When, $V_{GSn} \leq V_{Tn}$

- nFET-Open switch
- no current flow
- Transistor is off

When, $V_{GSn} \geq V_{Tn}$

- nFET-Closed Switch
- Conducts current
- Transistor is on

- pFET:**
- pFET behaves in a **complementary** manner
 - pFET is characterized by a **threshold voltage**, V_{Tp} values around -0.5 V to -0.8 V.



- $V_A=0$, large V_{SGp} ; **pFET is on**
- $V_A=1$, V_{SGp} is small ; **pFET is off.**

$$V_A + V_{SGp} = V_{DD}$$

$$V_A = V_{DD} - V_{SGp}$$

When, $V_{SGp} \leq |V_{Tp}|$

- pFET-Open switch
- no current flow
- Transistor is off

When, $V_{SGp} \geq |V_{Tp}|$

- pFET-Closed Switch
- Conducts current
- Transistor is on

- logic 0 and logic 1 voltage ranges **V_A are different for the two types of FETs.**
- One way around this problem is to note that **there are regions of overlap for both $A = 0$ and $A = 1$** values that can be used if a uniform definition is needed.
- The ideal values of,

$$V_A = 0 \text{ V}$$

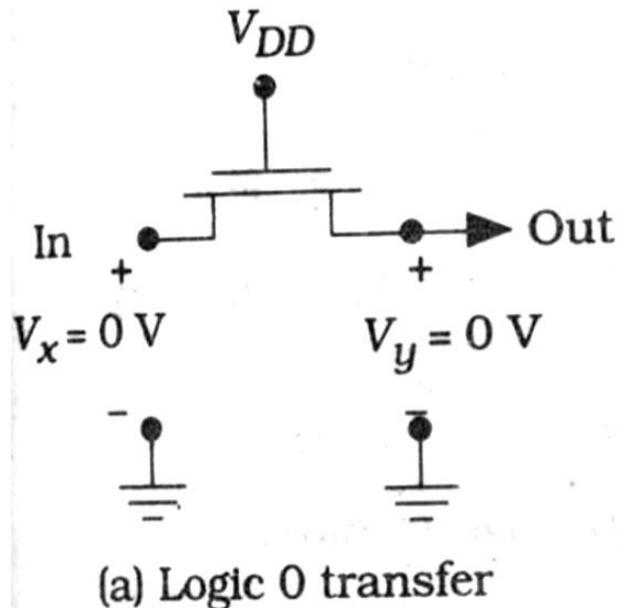
$$V_A = V_{DD}$$

valid for both devices.

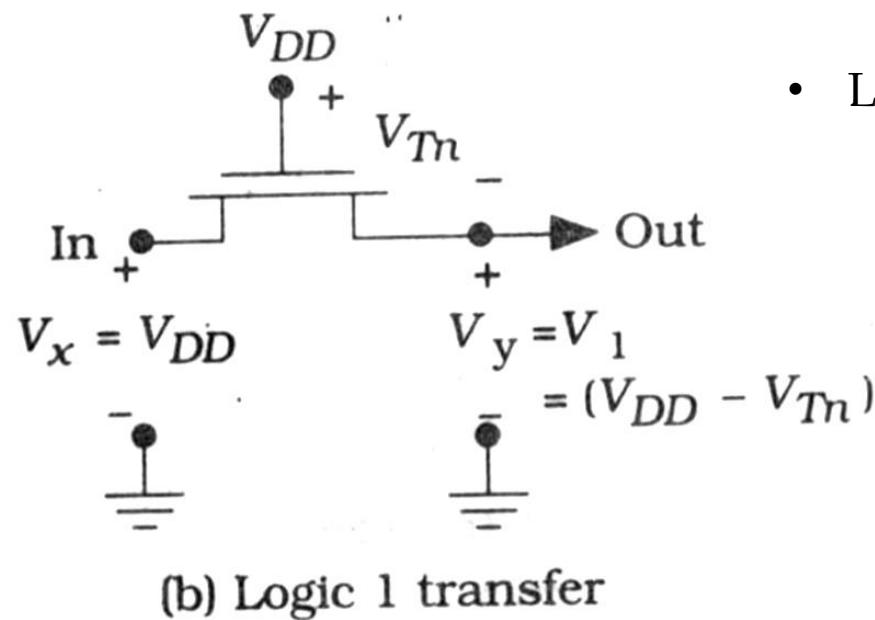
Pass Characteristics

- An ideal electrical switch can pass any voltage applied to it.
- MOSFETs are more limited in their capabilities

nFET:



(a) Logic 0 transfer



(b) Logic 1 transfer

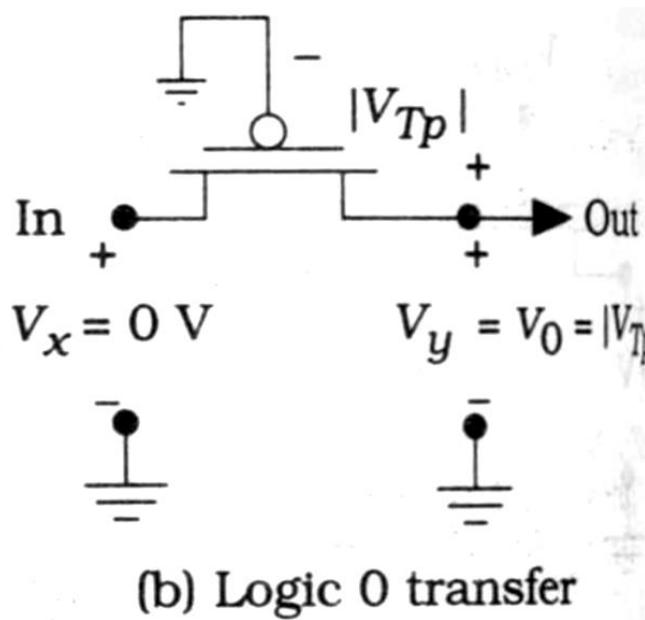
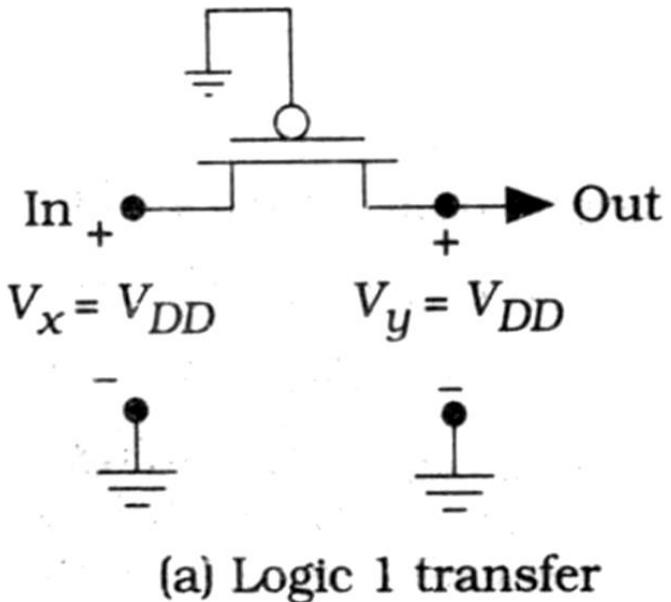
- Logic-0 voltage of $V_x=0\text{V}$
 $V_y=0\text{V}$
 - Logic-1 voltage of $V_x=V_{DD}$
 V_y reduced to,
$$V_1 = V_{DD} - V_{Tn}$$
- Threshold voltage loss**

nFET can pass,

- **A week logic 1**
- **A strong logic 0**

voltage range $[0, V_1]$

pFET:



- Logic-1 voltage of $V_x=V_{DD}$
 $V_y=V_{DD}$
- Logic-0 voltage of $V_x=0$
 V_y reduced to,
 $V_y = |V_{Tp}|$

Threshold voltage loss

FET can pass,
•A Strong logic 1
•A week logic 0

voltage range $[V_{DD}, V_0]$

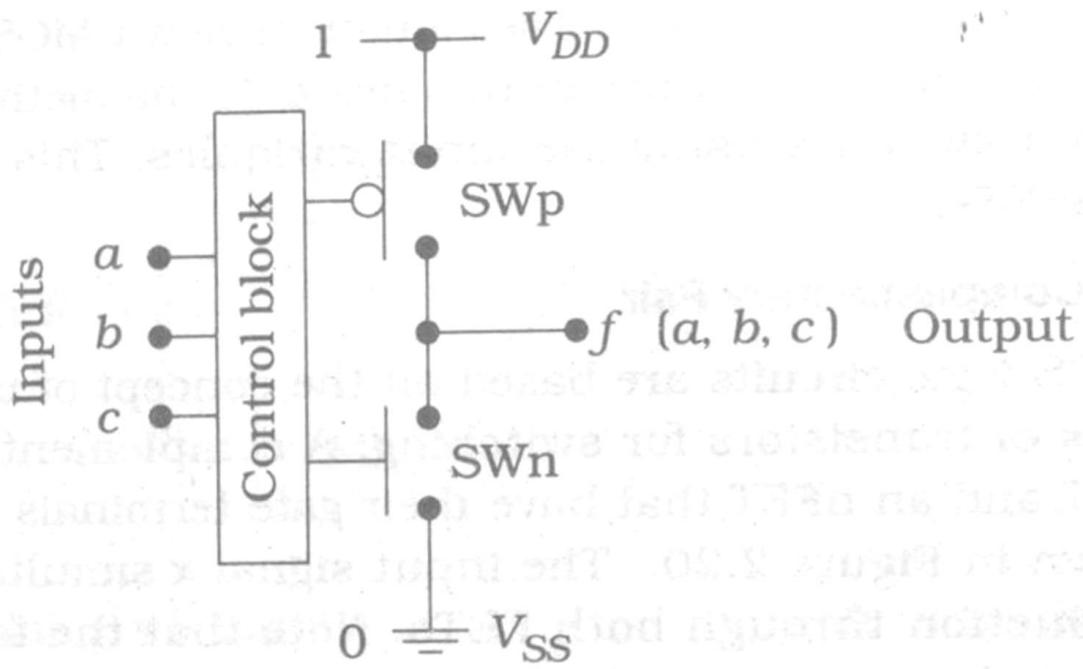
- nFETs pass strong logic 0 voltages, but weak logic 1 values
- pFETs pass strong logic 1 voltages, but weak logic 0 levels

Complementary MOS (CMOS) circuits are designed to account for the transmission levels.

- 1. Use pFETs to pass logic 1 voltages of V_{DD}
- 2. Use nFETs to pass logic 0 voltages of $V_{SS} = 0 \text{ V}$

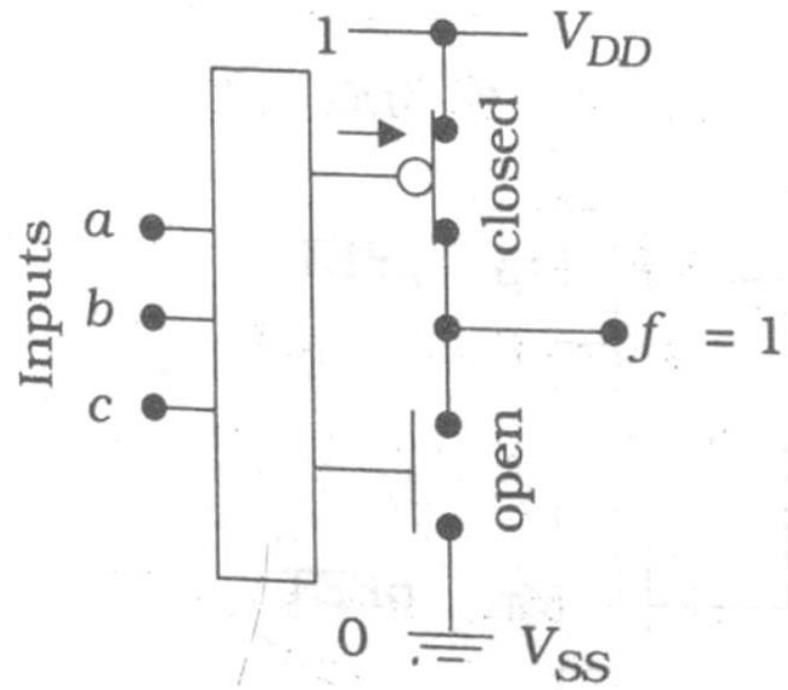
Basic logic gates in CMOS

General CMOS logic gate:



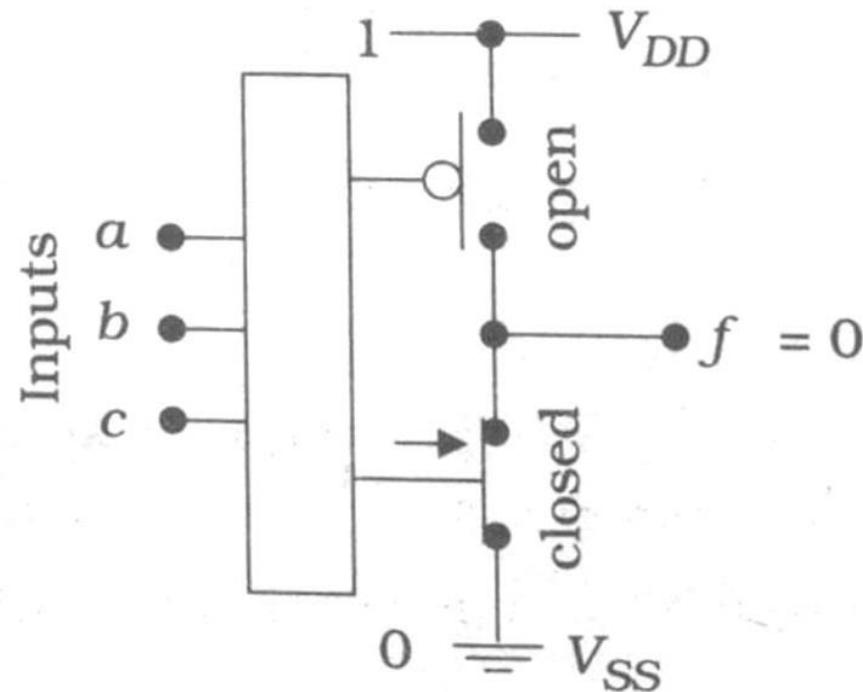
- a,b,c are input bits combine to give the output function bit $f(a, b, c)$.
- In Digital circuit, all of the quantities are restricted to values of 0 or 1.
- Digital logic circuits are **nonlinear networks that use transistors as electronic switches** to divert *one* of the supply voltages V_{DD} or 0V to the output.
- Corresponds to a logical result of $f = 1$ or $f = 0$.
- Output network of the gate as consisting of two switches:
 - **SWp** (an assert-low device)
 - **SWn** (an assert-high device)
- These are **wired in to Insure** that one switch is closed while the other switch is open.

Operation of a CMOS logic gate:



(a) $f = 1$ output

- upper switch is closed while the lower switch is open.
- Output is connected to the power supply and yields a value of $f = 1$.

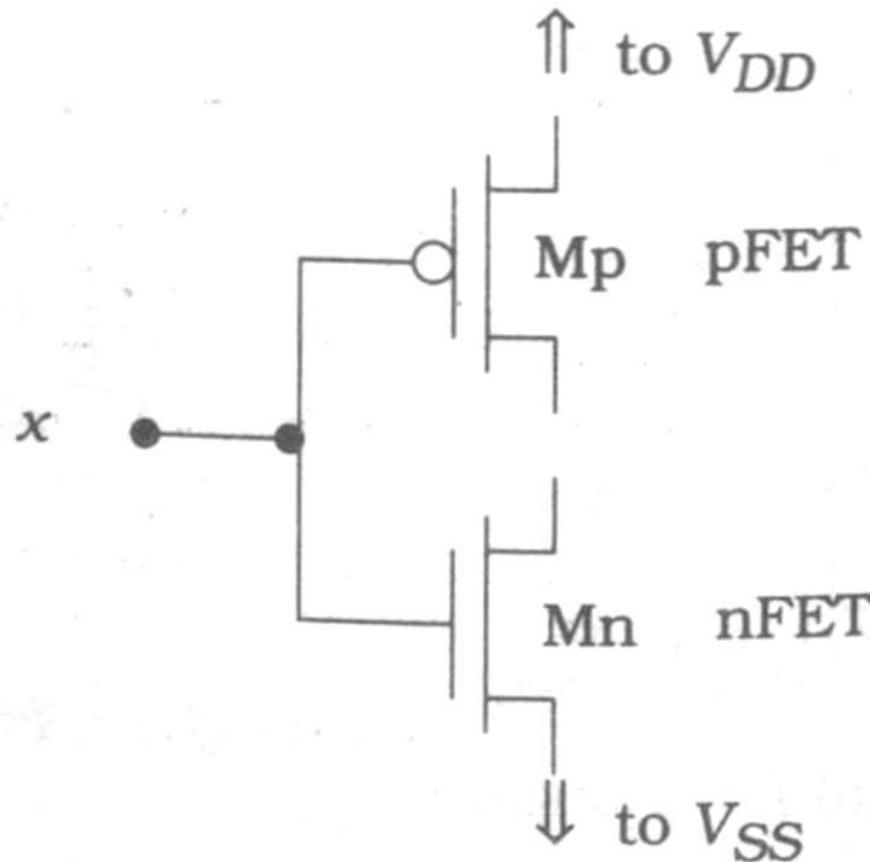


(b) $f = 0$ output

- upper switch is open and the lower switch is closed.
- Output is now connected to $V_{SS} = 0$, the logical result is $f = 0$.

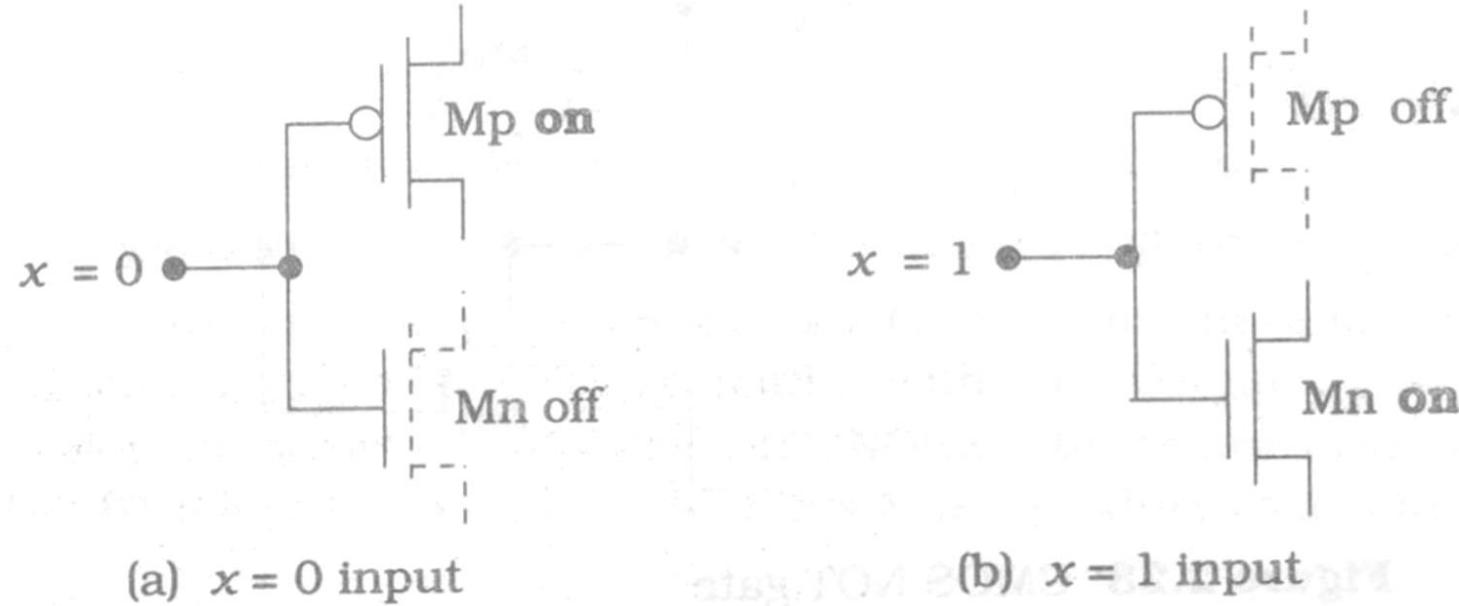
Missing feature in this model : method used to control the output switches using the input variables.

A CMOS complementary pair:



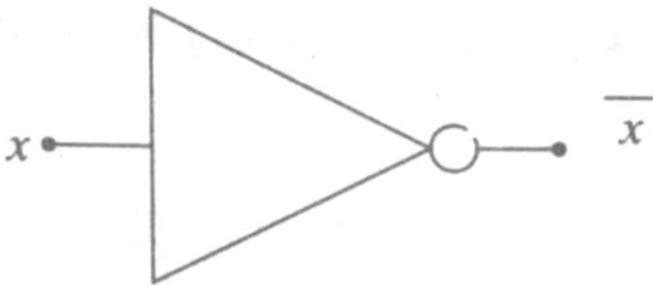
- Based on the concept of using **complementary pairs** of transistors for switching.
- Gate terminals of both transistors are connected together .
- The input signal x simultaneously control the conduction through both FETs.

Operation of the complementary pair:



The NOT Gate

$$f(x) = \text{NOT}(x) = \overline{x}$$



(a) Logic symbol

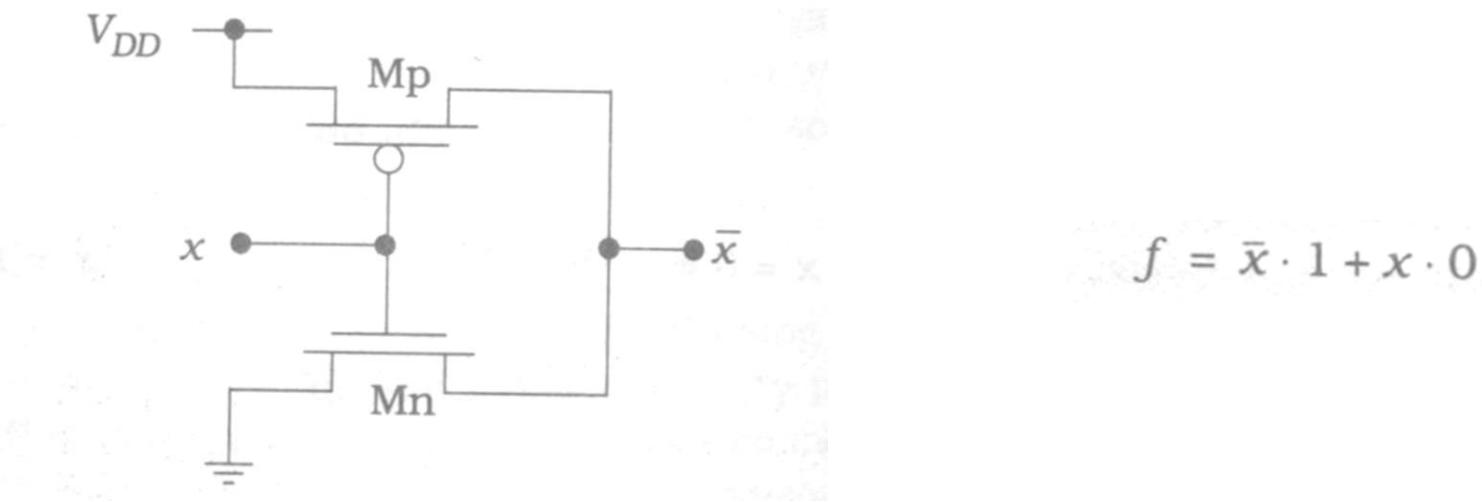
x	\bar{x}
0	1
1	0

(b) Truth table

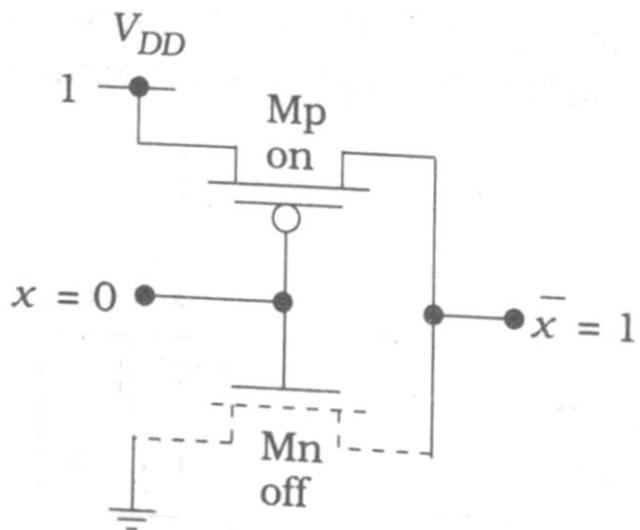
If $x = 0$ then $\bar{x} = 1$

If $x = 1$ then $\bar{x} = 0$

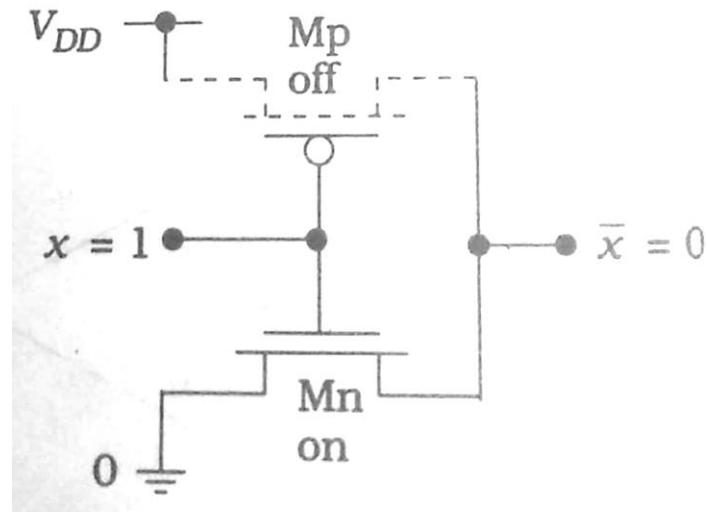
CMOS NOT gate:



Operation of CMOS NOT gate:



(a) $x = 0$ input



(b) $x = 1$ input

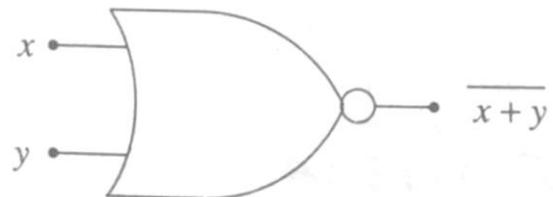
Important characteristics:

- Input logic state of $x = 0$ or 1
- output is connected to either VDD or ground.

Important guidelines helps us design logic circuits:

- Use a complementary nFET/pFET pair for each input
- Connect the output node to the power supply VDD through pFETs
- Connect the output node to ground through nFETs
- Insure that the output is always a well-defined high or low voltage

CMOS NOR gate

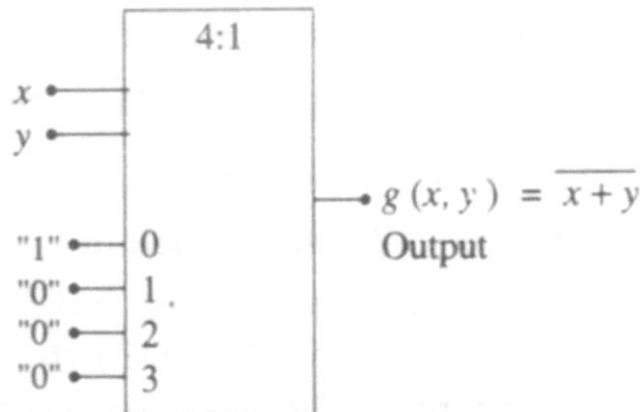


(a) Logic symbol

x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

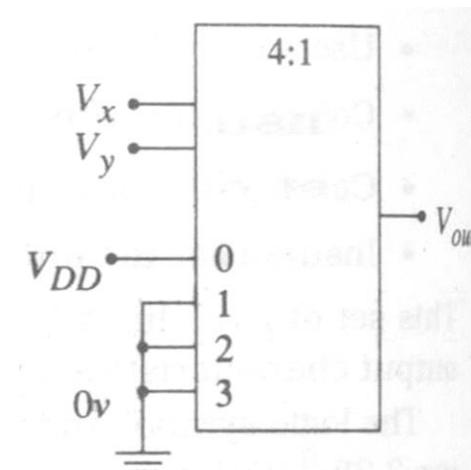
(b) Truth table

Basic for constructing CMOS NOR2 circuit



(a) Logic diagram

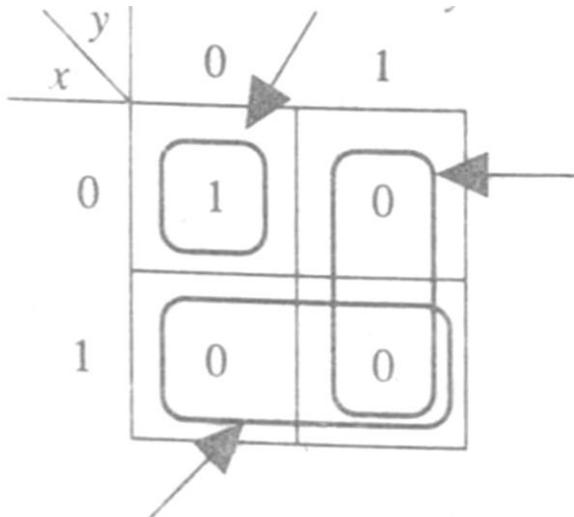
$$g(x, y) = \overline{x+y}$$



(b) Voltage network

$$g(x, y) = \bar{x} \cdot \bar{y} \cdot 1 + \bar{x} \cdot y \cdot 0 + x \cdot \bar{y} \cdot 0 + x \cdot y \cdot 0$$

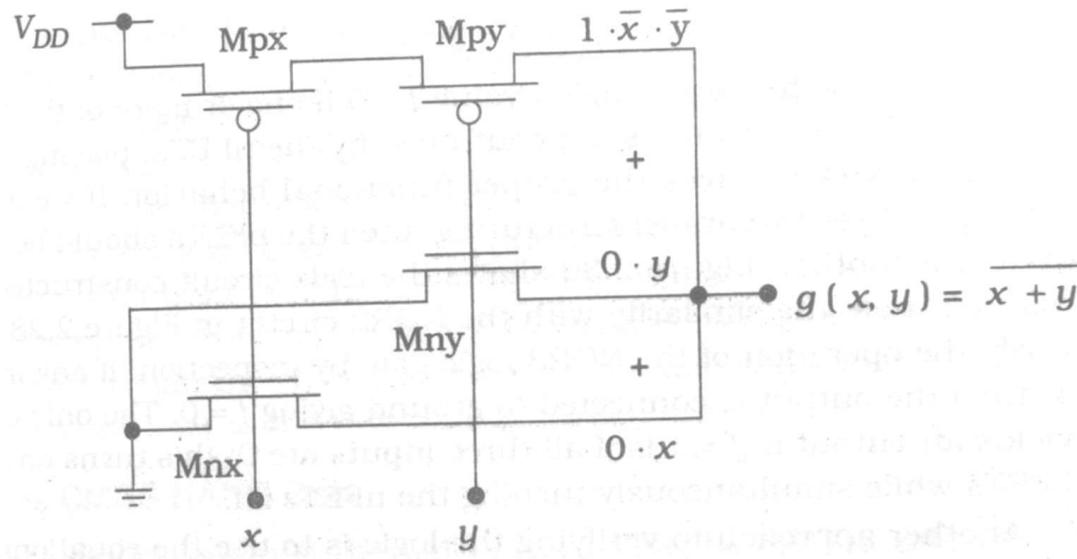
K-map:



$$g(x, y) = \bar{x} \cdot \bar{y} \cdot 1 + x \cdot 0 + y \cdot 0$$

CMOS NOR2 circuit

x	y	Mpx	Mpy	Mnx	Mny	g
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

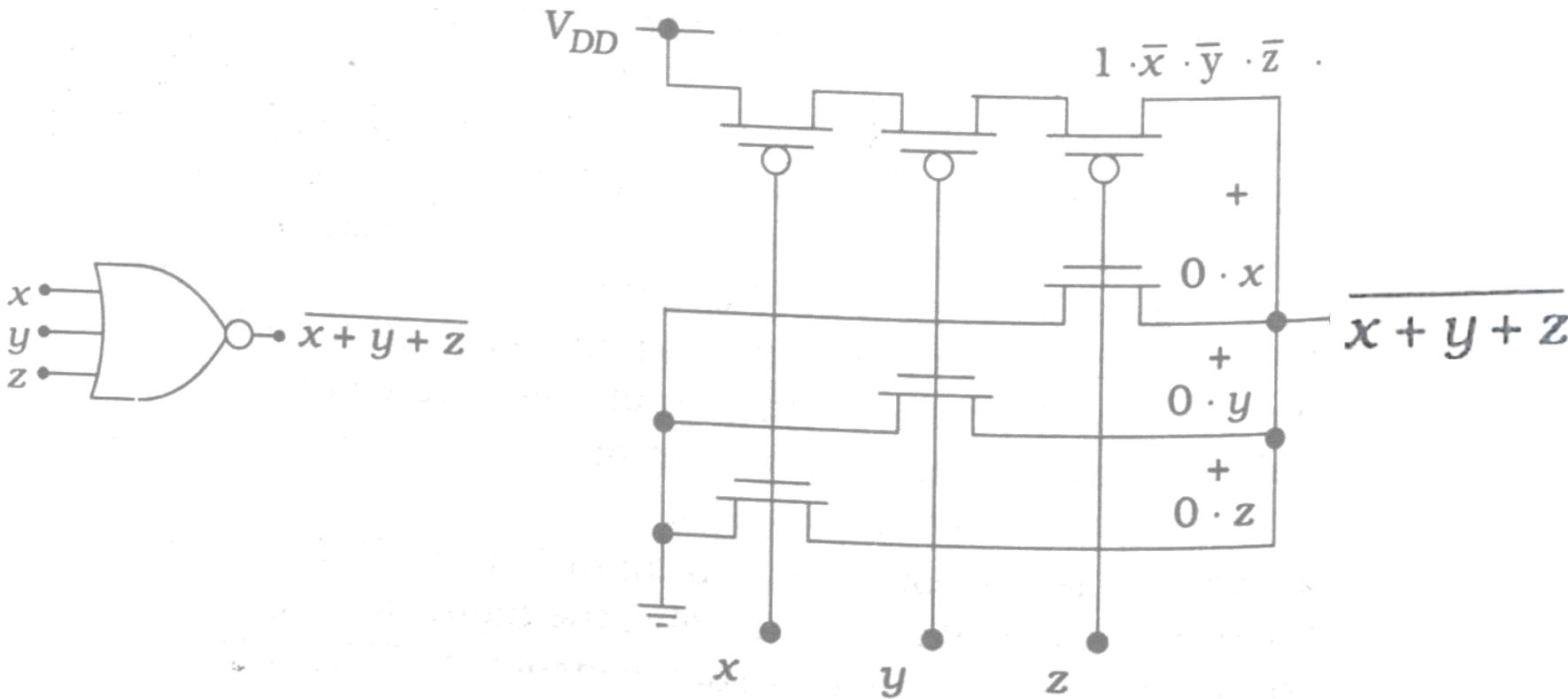


Series-parallel transistor arrangement

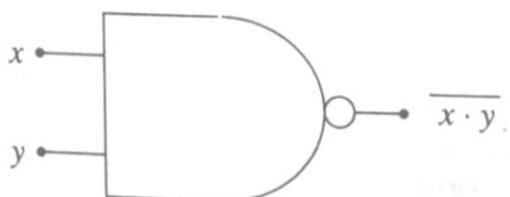
CMOS NOR3 circuit:

$$f = 1 \cdot \bar{x} \cdot \bar{y} \cdot \bar{z} + 0 \cdot x + 0 \cdot y + 0 \cdot z$$

$$f = 1 \cdot \bar{x} \cdot \bar{y} \cdot \bar{z} = \overline{x + y + z}$$



CMOS NAND gate



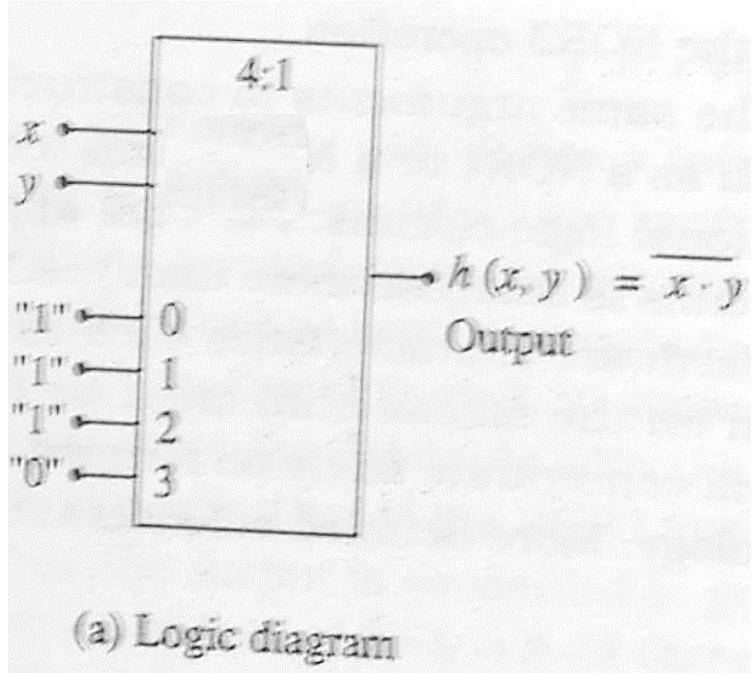
(a) Logic symbol

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0

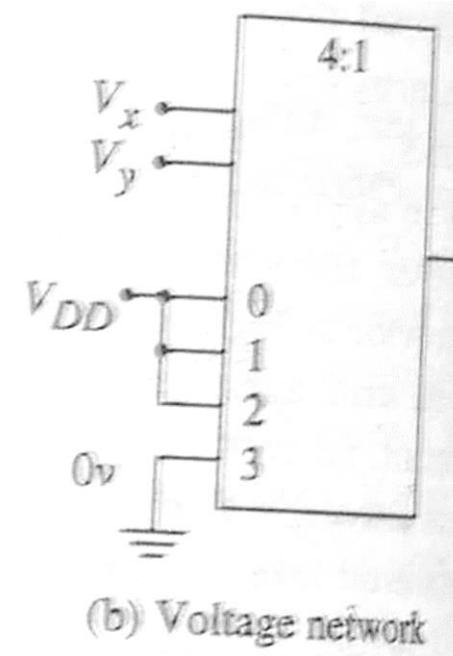
$$g(x, y) = \overline{x \cdot y}$$

(b) Truth table

Basic for constructing CMOS NAND2 circuit



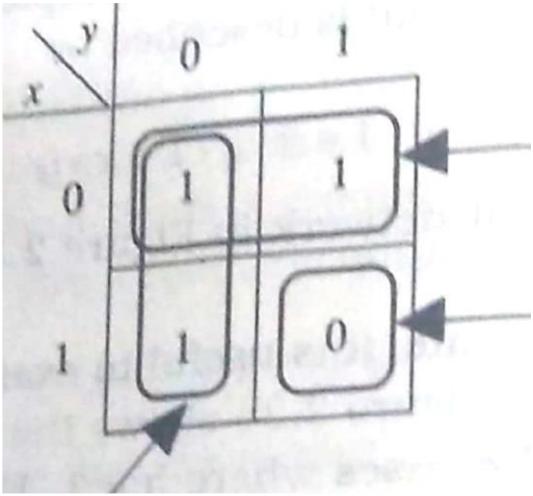
(a) Logic diagram



(b) Voltage network

$$h(x, y) = \bar{x} \cdot \bar{y} \cdot 1 + \bar{x} \cdot y \cdot 1 + x \cdot \bar{y} \cdot 1 + x \cdot y \cdot 0$$

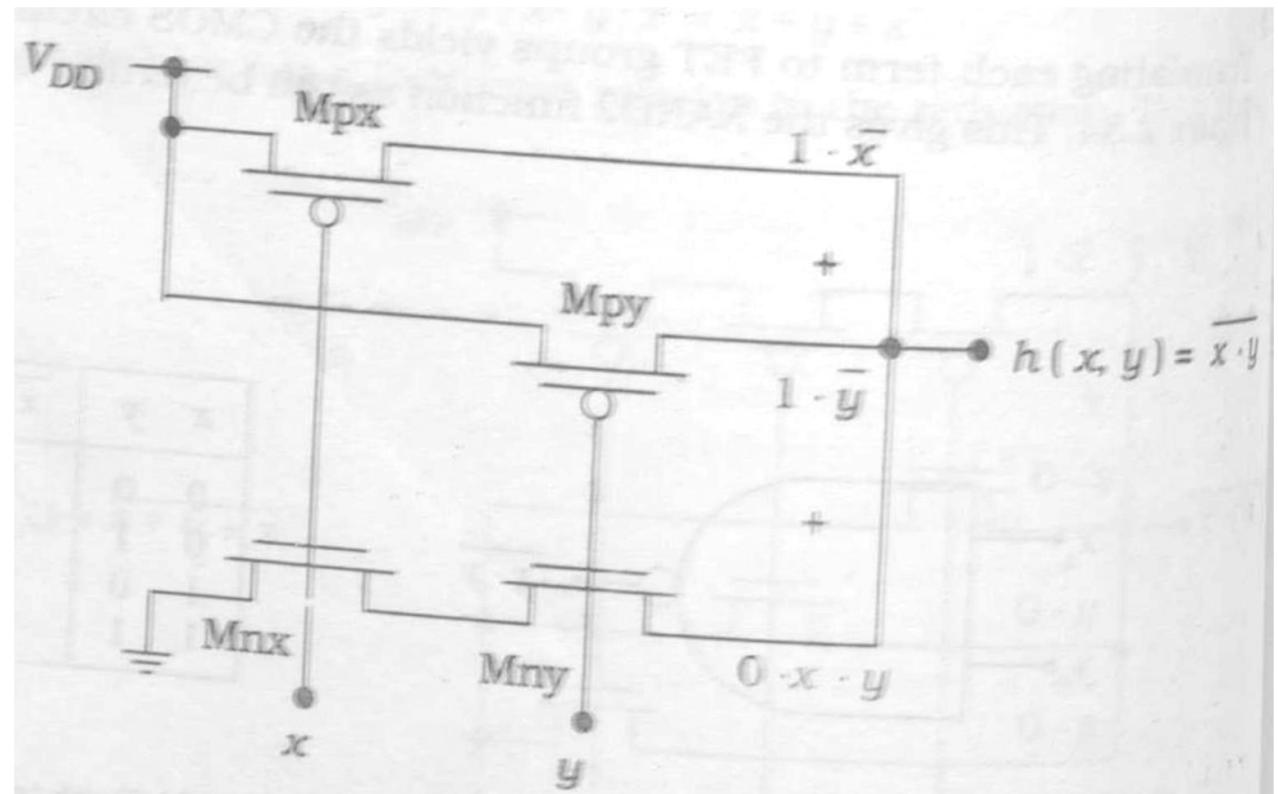
K-map:



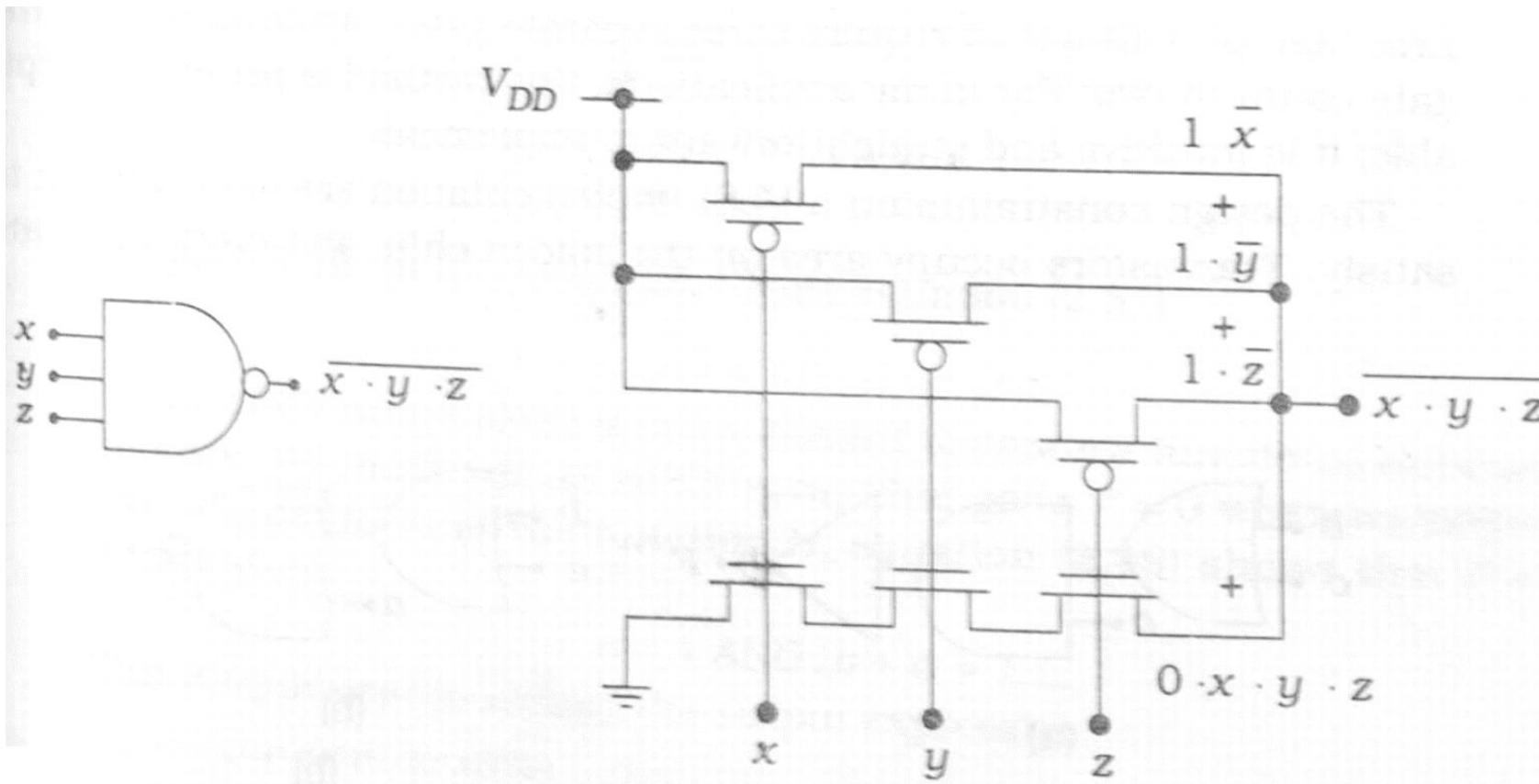
$$h(x, y) = \bar{x} \cdot 1 + \bar{y} \cdot 1 + x \cdot y \cdot 0$$

CMOS NAND2 circuit

x	y	Mpx	Mpy	Mnx	Mny	h
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	0



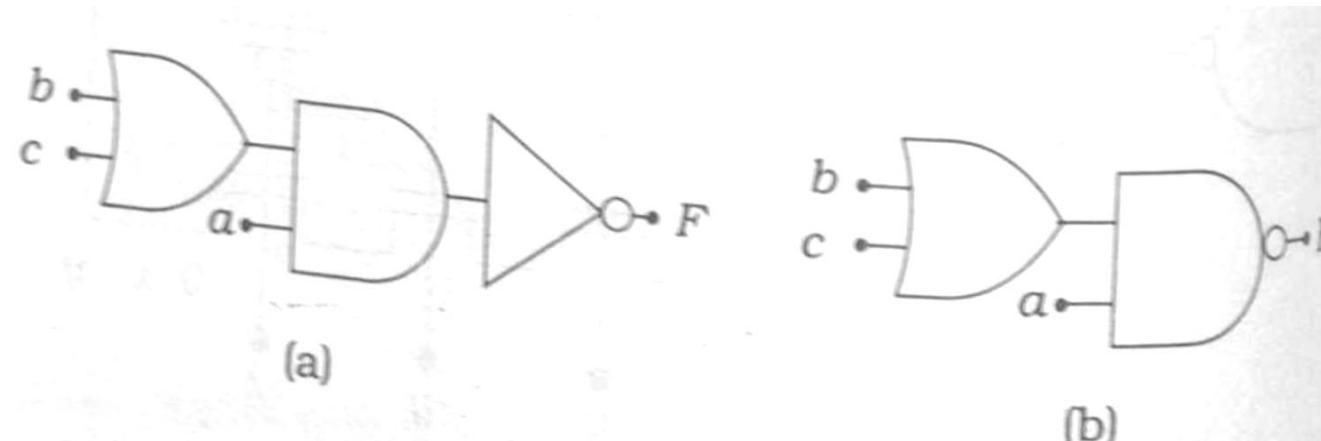
CMOS NAND3 circuit:



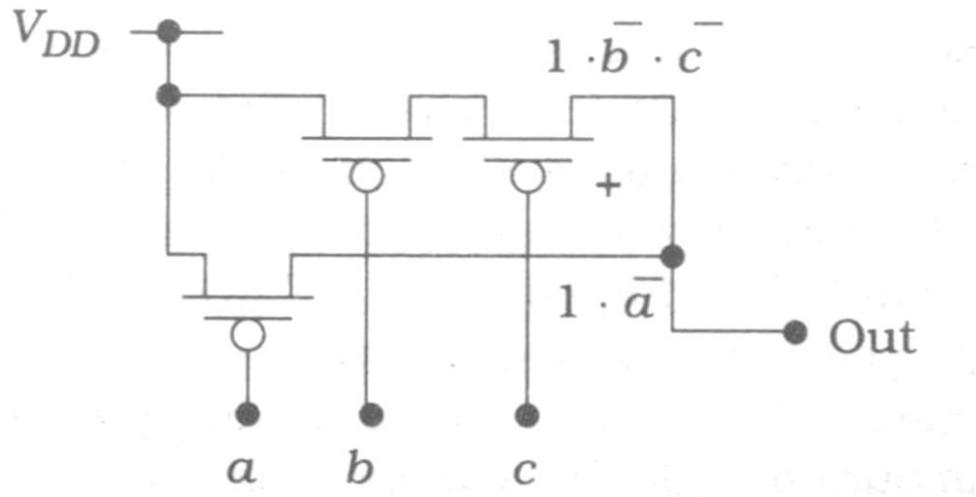
$$0 \cdot x \cdot y \cdot z + 1 \cdot \bar{x} + 1 \cdot \bar{y} + 1 \cdot \bar{z}$$

Complex logic gates in CMOS

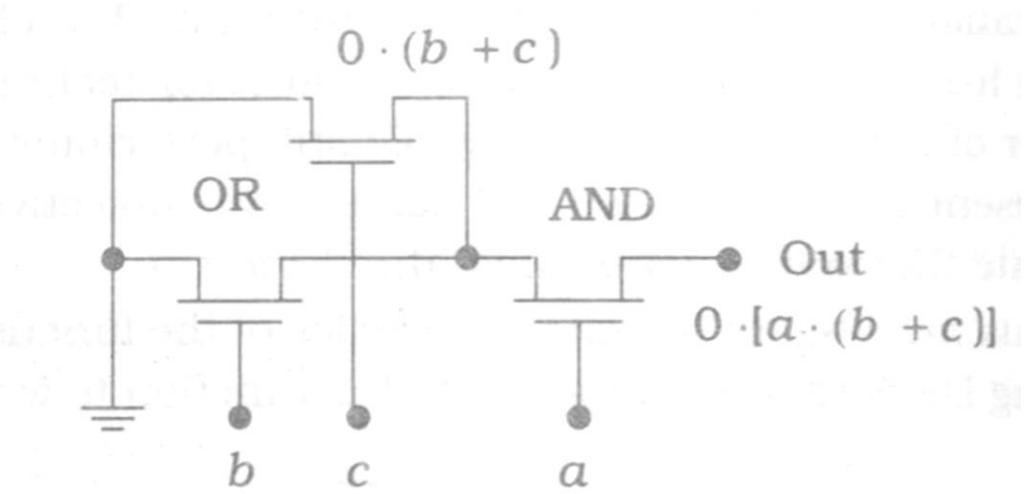
- One of the most powerful aspects of building logic circuits in CMOS is **the ability to create a single circuit that provides several primitive operations** (NOT, AND and OR) in an integrated manner.
- These will be called **complex or combinational logic gates**.
- Consider, $F(a, b, c) = \overline{a \cdot (b + c)}$



$$\begin{aligned}
 F &= \overline{a \cdot (b + c)} \\
 &= \overline{a} + \overline{(b + c)} \\
 &= [\overline{a} + (\overline{b} \cdot \overline{c})] \cdot 1 \\
 F &= \overline{a} \cdot 1 + (\overline{b} \cdot \overline{c}) \cdot 1
 \end{aligned}$$

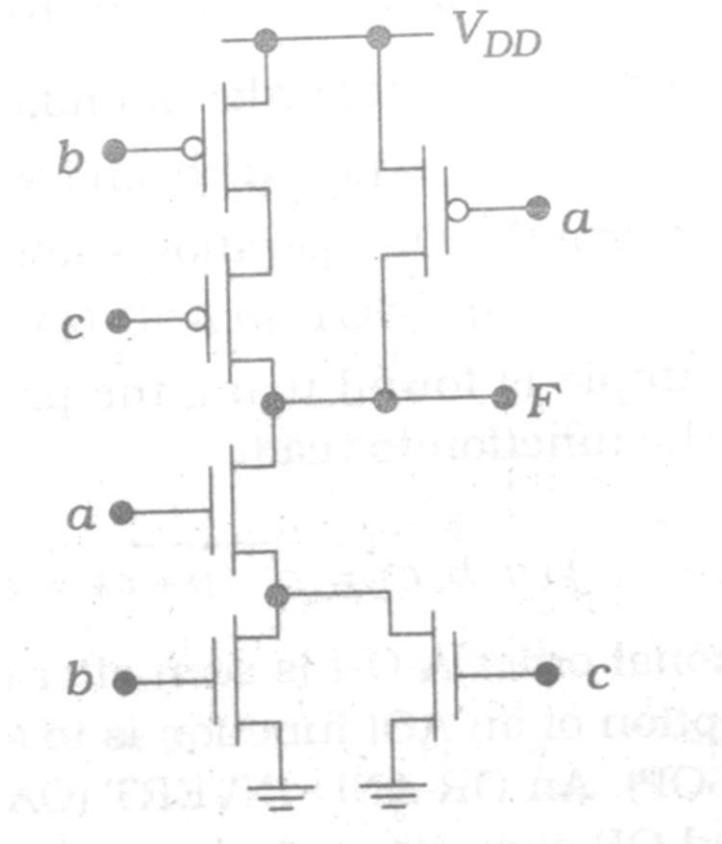


pFET circuit for F from equation



nFET logic circuit for F

$$F(a, b, c) = \overline{a \cdot (b + c)}$$



Finished complex CMOS logic gate circuit

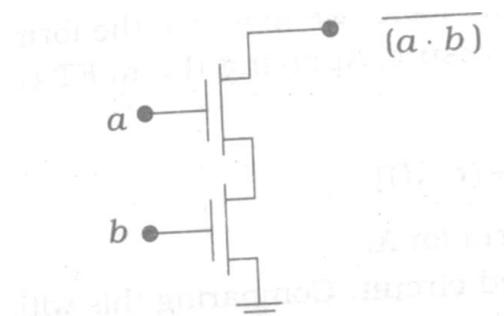
Structured Logic Design

- A structured approach to designing complex logic gates **can be developed by focusing on the circuit characteristics.**
- CMOS logic gates are intrinsically **inverting**;
 - The output always produces a NOT operation acting on the input variables.
- The inverting nature of CMOS logic circuits allows us to construct logic circuits for AOI and OAI logic expressions using a structured approach.
 - **AOI logic** - AND then OR then NOT (invert).

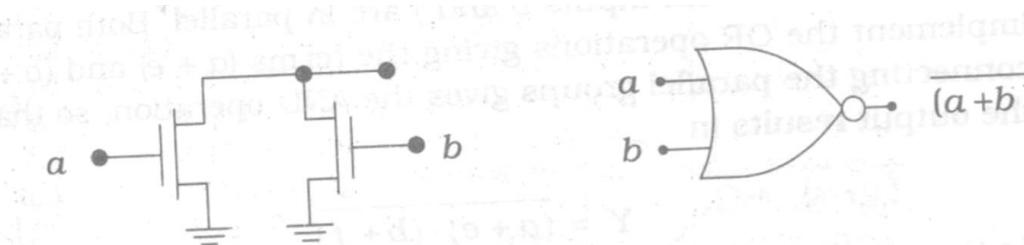
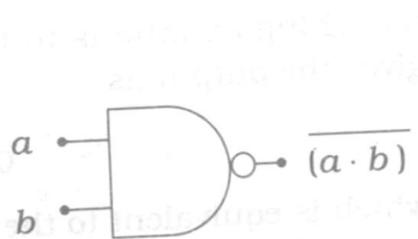
$$g(a, b, c, d) = \overline{a \cdot b + c \cdot d} ; g = \text{NOT } [(a \text{ AND } b) \text{ OR } (c \text{ AND } d)] \quad \text{Inverted SOP}$$

- **OAI logic** - OR then AND then NOT (invert).

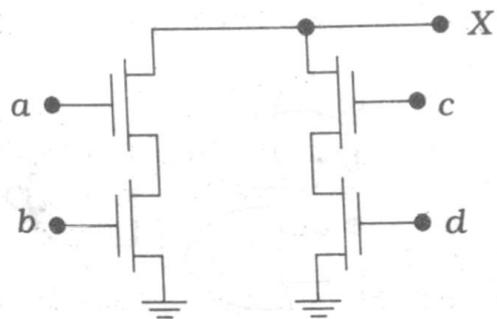
$$h(x, y, z, w) = \overline{(x + y) \cdot (z + w)} ; h = \text{NOT } [(x \text{ OR } y) \text{ AND } (z \text{ OR } w)] \quad \text{Inverted POS}$$



(a) Series-connected nFETs

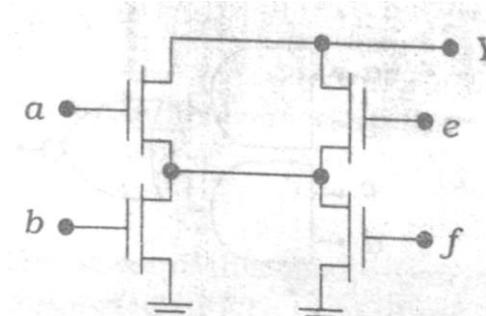
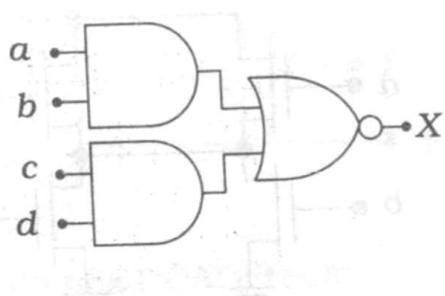


(b) Parallel-connected nFETs



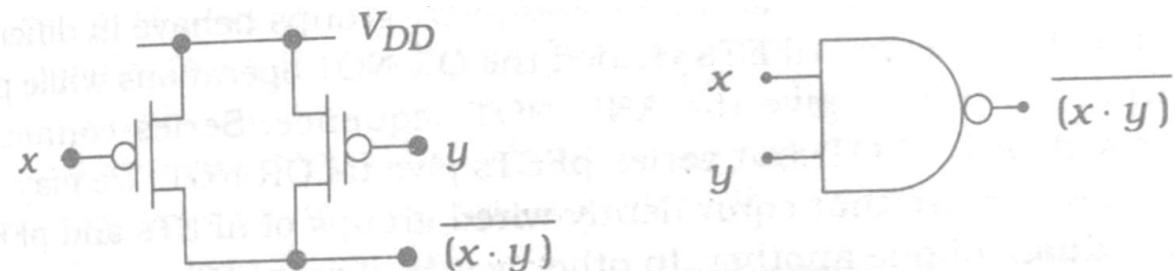
nFET AOI logic

$$X = \overline{(a \cdot b) + (c \cdot d)}$$

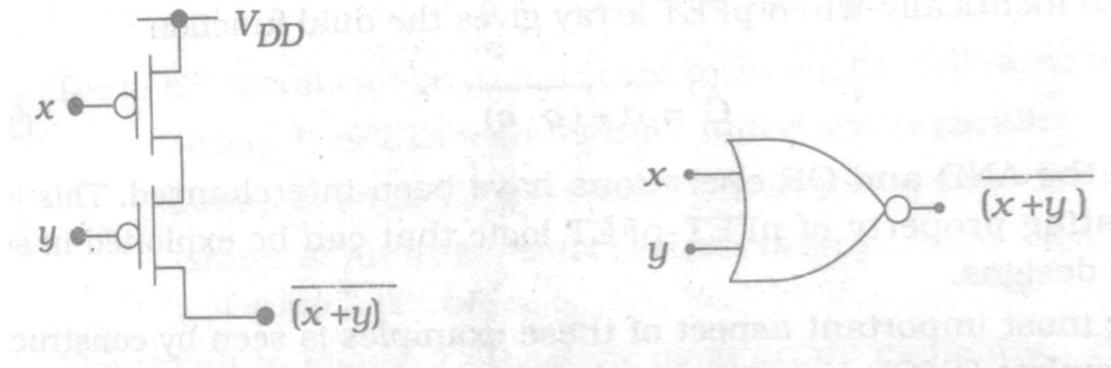


nFET OAI logic

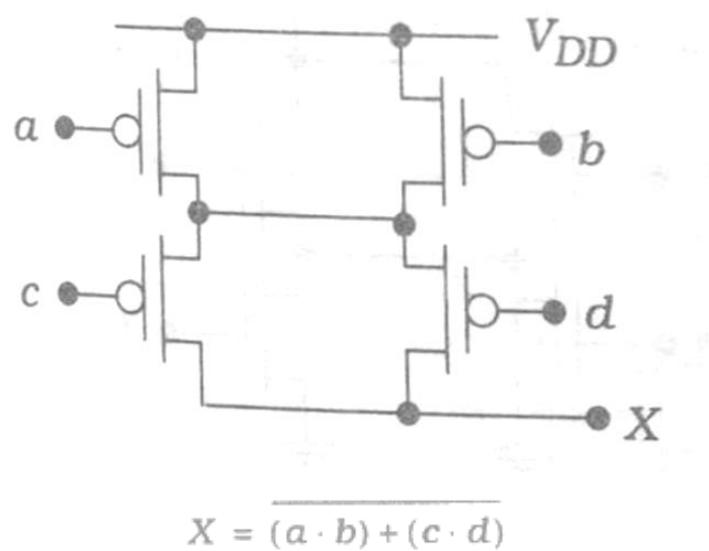
$$Y = \overline{(a+e) \cdot (b+f)}$$



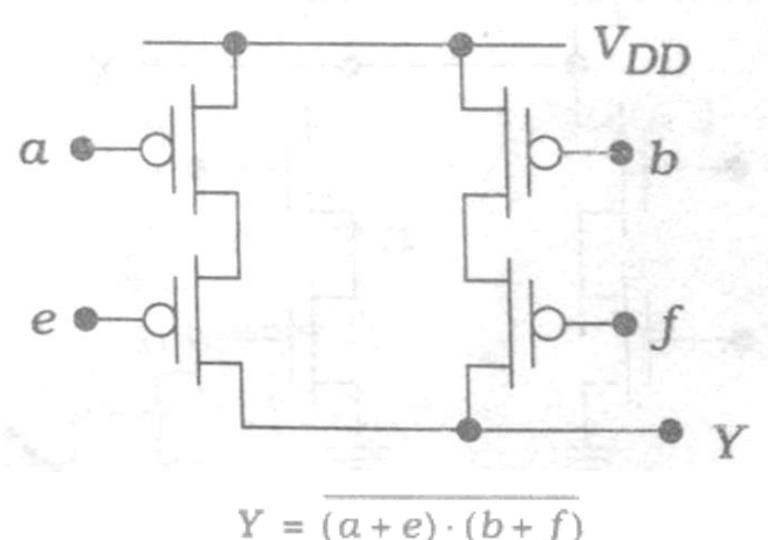
(a) Parallel-connected pFETs



(b) Series-connected pFETs



(a) pFET AOI circuit



(b) pFET OAI circuit

Common procedure used for all type of circuits:

NMOS

OR operation (+)

AND operation (.)

parallel
series

PMOS

OR operation (+)

AND operation (.)

series
parallel

1. AND and NAND

2. OR and NOR

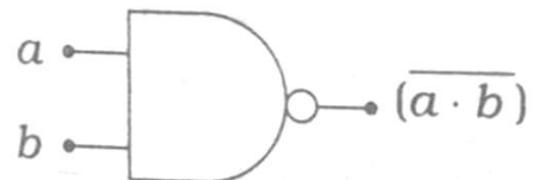
3. $Y = a.(b+c)$

4. XOR and XNOR

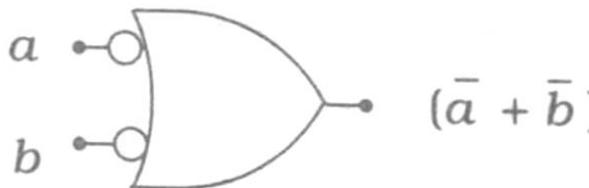
5. $Y = \overline{a + b.(c + d)}$

6. $Y = \overline{a.(c + d)}$

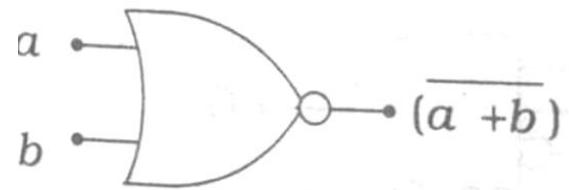
Bubble Pushing



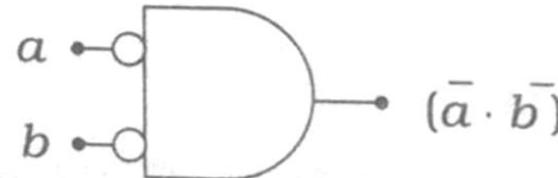
Equivalent
 \Leftrightarrow



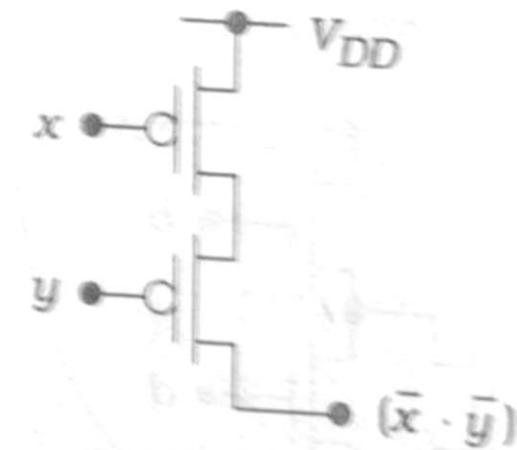
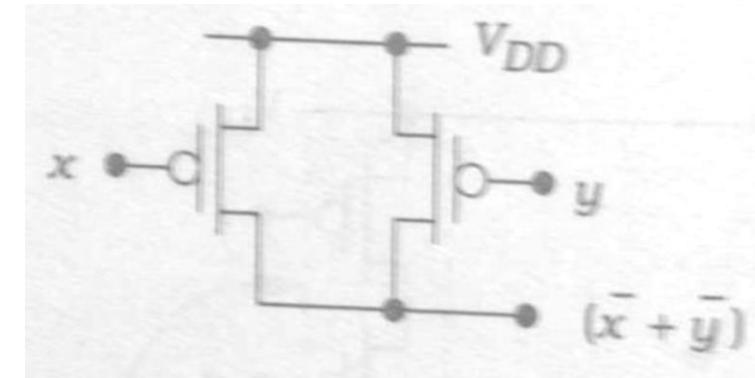
(a) NAND - OR



Equivalent
 \Leftrightarrow

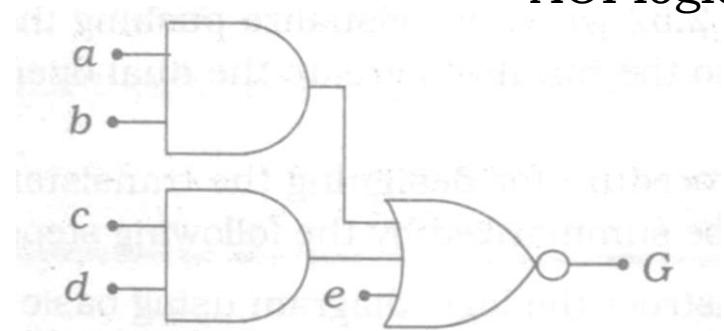


(b) NOR - AND

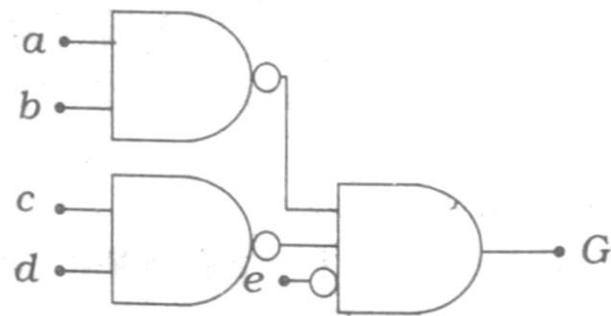


Bubble pushing using DeMorgan rules

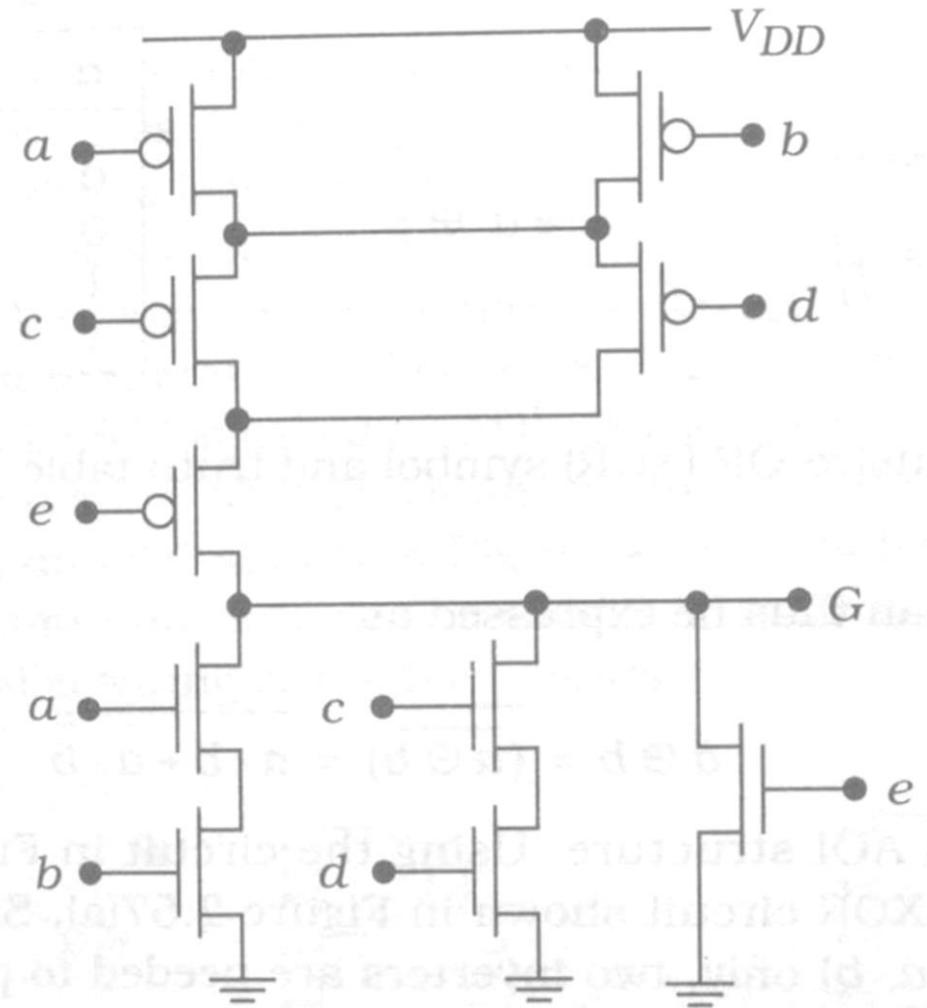
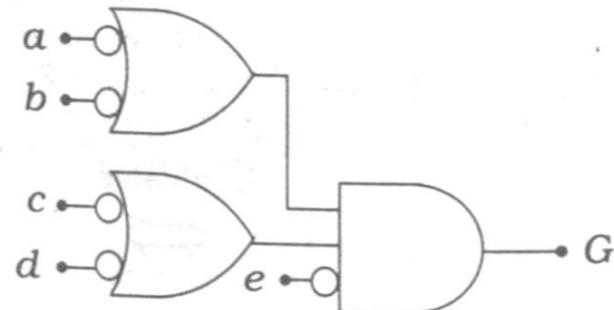
AOI logic diagram for bubble-pushing example



First transformation



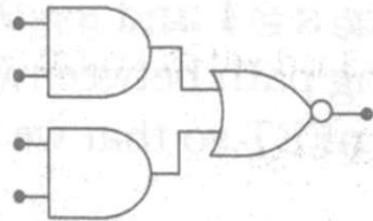
Final form



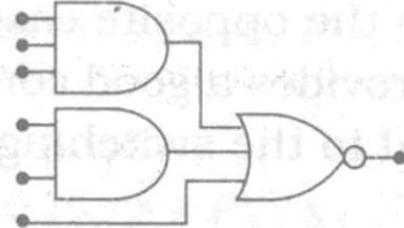
Final circuit for the bubble-pushing example

Generalized AOI and OAI Logic Gates

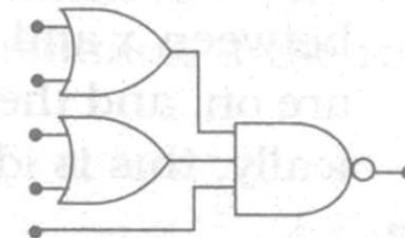
- Standard logic design is simplified using generalized multiple-input AOI and OAI logic gates.
- various input configurations:



(a) AOI22 gate



(b) AOI321 gate



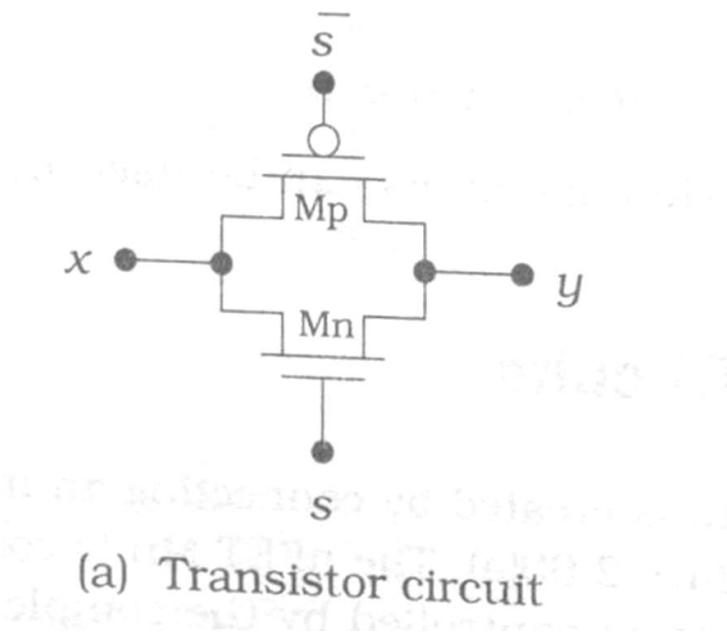
(c) OAI221 gate

AOI pattern with 2 inputs to each AND gate;

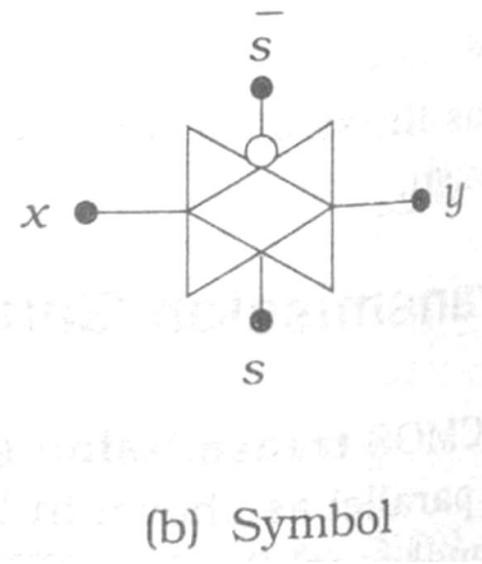
“ 1 ” label implies an input that bypasses the AND gates and is connected directly to an OR gate.

Transmission gate circuits:

- Transmission gate is created by connecting an nFET and pFET in parallel.
- TG is classified as a **bi-directional switch**.



(a) Transistor circuit



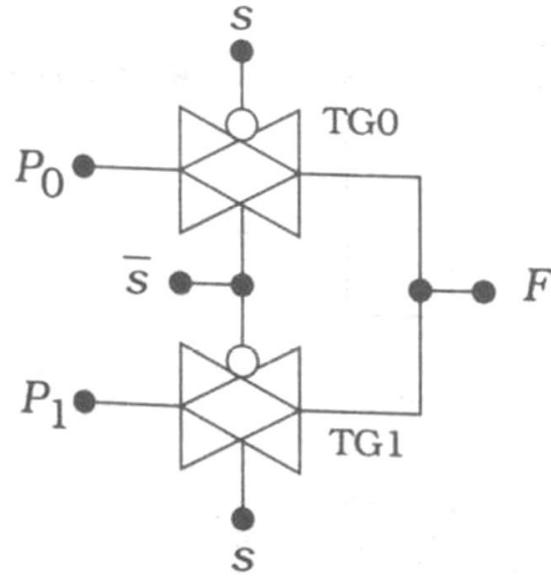
(b) Symbol

$$y = x \cdot s \quad \text{iff} \quad s = 1$$

- The symbol is created by two back to back arrows indicating that the data can flow in either direction.
- TG can transmit the entire voltage range (0, V_{DD})
- **Advantages:** simplicity of the switching and the ability to transmit the entire range of voltages
- **Drawback:** requires two FETs and an implied inverter that takes S and produces \bar{S}

Logic Design:

Multiplexors

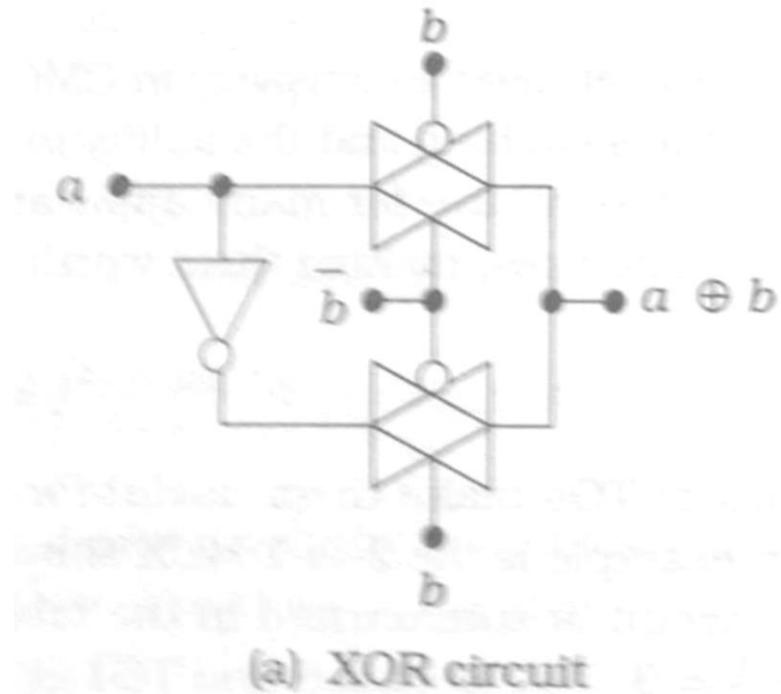


s	TG0	TG1	F
0	Closed	Open	P_0
1	Open	Closed	P_1

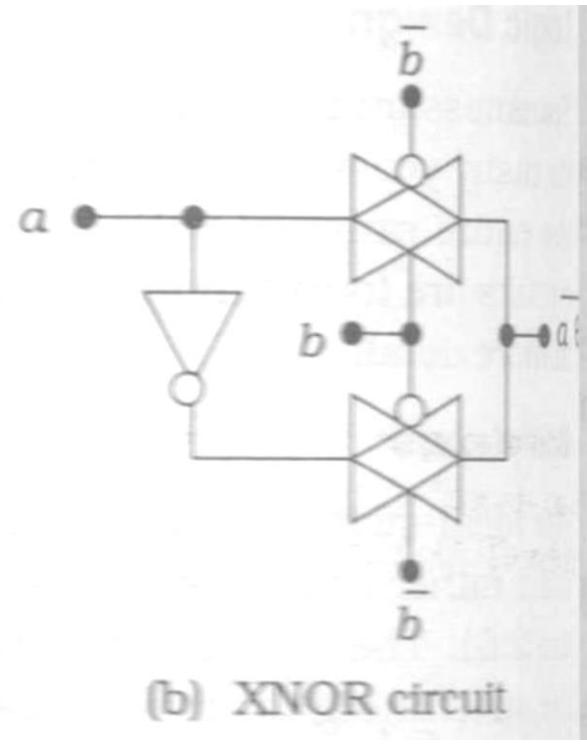
$$F = P_0 \cdot \bar{s} + P_1 \cdot s$$

- The 2-to-1 architecture can be extended to a 4:1 network by using the 2-bit selector word ($s_1\ s_0$) that has values of (0 0), (0 1), (1 0), and (1 1).
- Each input line (P_0, P_1, P_2, P_3) will have two TGs in its path such that the output is

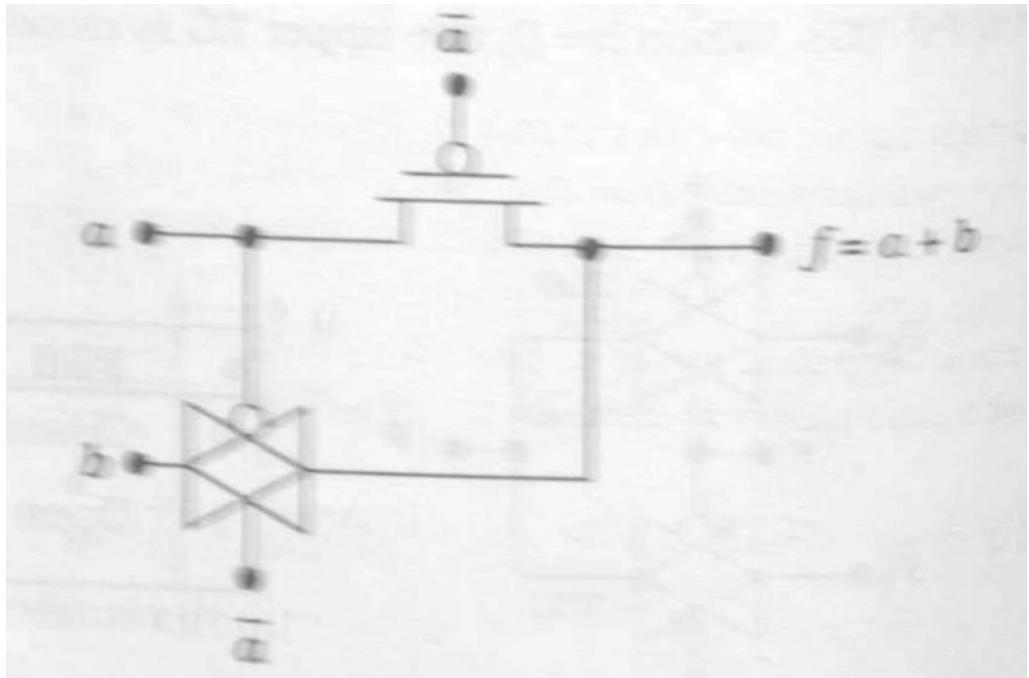
$$F = P_0 \cdot \bar{s}_1 \cdot \bar{s}_0 + P_1 \cdot \bar{s}_1 \cdot s_0 + P_2 \cdot s_1 \cdot \bar{s}_0 + P_3 \cdot s_1 \cdot s_0$$



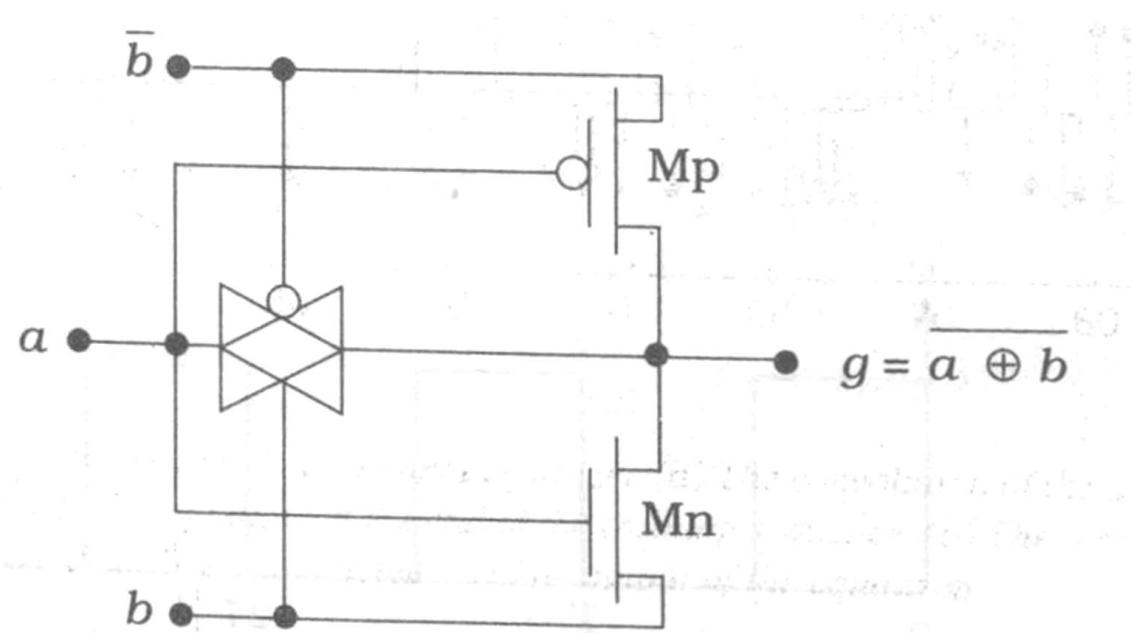
(a) XOR circuit



(b) XNOR circuit



TG based OR gate



An XNOR gate that uses both TGs and FETs