IE1204 Digital Design:

F3 : CMOS Implementation of Boolean Logic

Slides from Ahmed Hemani

KTH/ICT/ES

hemani@kth.se

Why are we studying this Lecture ?

- Boolean aglebra helps us design arbitrary logic functionality
- Most of the course is about Boolean algebra and logic design
- Physics implements the logic gates
  - Decides the cost metrics – area, speed and power consumption
- All real life scenarios require us to design logic that meets certain constraints on cost metrics:
  - Design a function ***f***
    - that does not exceed area ***A***
    - Has atleast speed ***S***
    - And does not consume more power than ***P***
  - There are multiple ways to implement function ***f*** that differs in their cost metrics

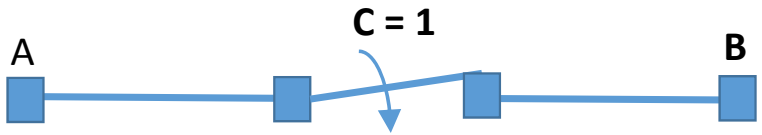
Outline

1. Semiconductor Primer
   Fundamentals necessary to understand the physics of logic gates

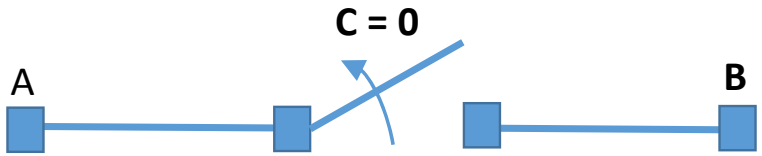
2. CMOS and Logic Implementation
   Semiconductor structure that implements gates

3. Electrical Aspects of CMOS circuits
   The essential parameters of CMOS that we will use to estimate area, speed/performance and power consumption

What we want to be able to do ?



The Switch is Closed. A and B are connected
Ideally the resistance between A and B is zero
A and B are at the same potential



The Switch is Open. A and B are disconnected
Ideally the resistance between A and B is infinite
The potential of A and B have no co-relation, i.e., potential at has no effect on potential at B and vice-versa

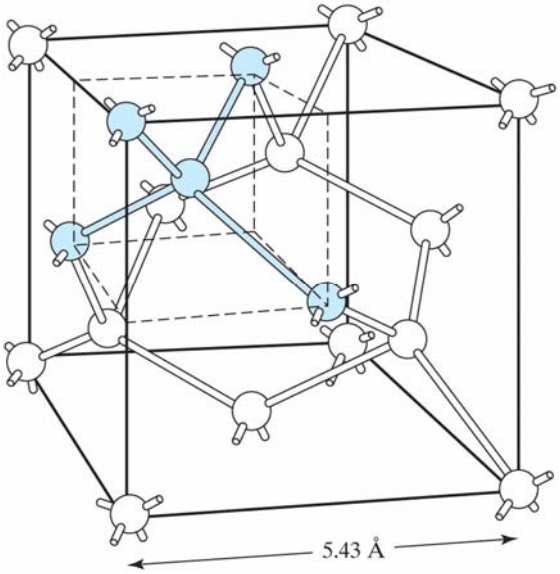
How Do We

Connect ***A*** and ***B*** and

Disconnect ***A*** and ***B***

electronically by applying a signal to ***C***

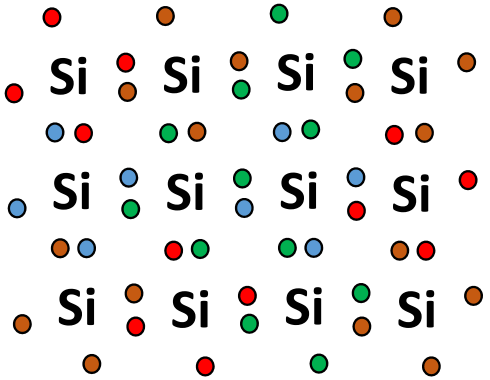
Silicon Atoms in Silicon Crystal couple with Covalent Bonds



Crystal Lattice

1 Ångström = 0.1 nano metre

Anders Jonas Ångström (1814 – 1874)



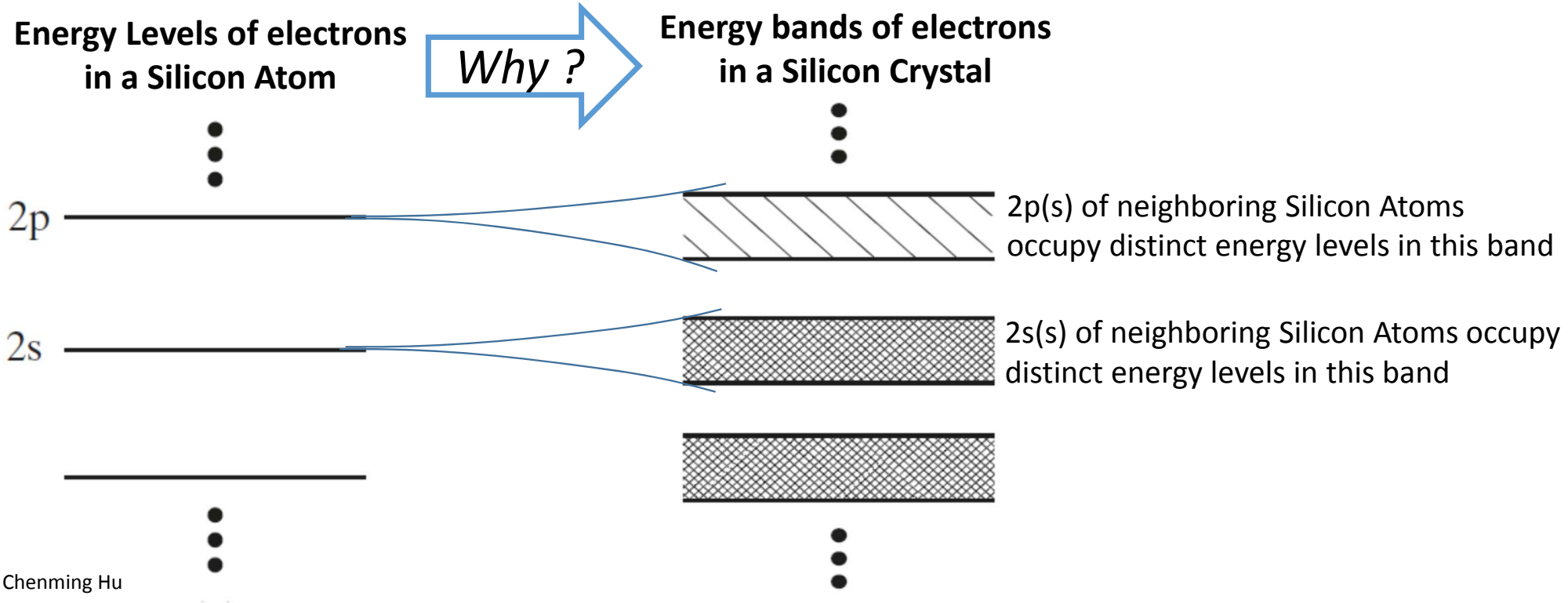
Silicon Atoms are coupled to each other by covalent bond

Two Dimensional Representation

Source: Chenming Hu

Energy Bands

1. Isolated Silicon Atoms have distinct energy levels when they are isolated as per Bohr's model in the diagram to the left
2. When they are in an a Silicon crystal, the inter atomic distance between silicon atoms is very small
3. This results in the energy level corresponding to the same orbitals (2s, 2p etc.) of neighbouring atoms in the silicon crystal to occupy slightly different levels of energy creating bands as shown in the diagram to the right

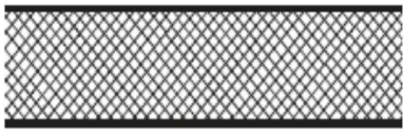


Source: Chenming Hu

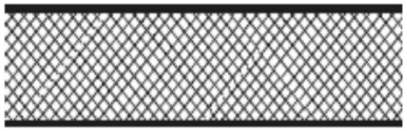
Valence vs. Conduction Bands

Upper Empty Bands.
With no electrons, these bands also do not contribute to conduction of electricity

Conduction Band
Lowest *nearly empty* band. Contributes to Conduction.
Higher Mobility *because nearly empty makes it easy for electrons to move*



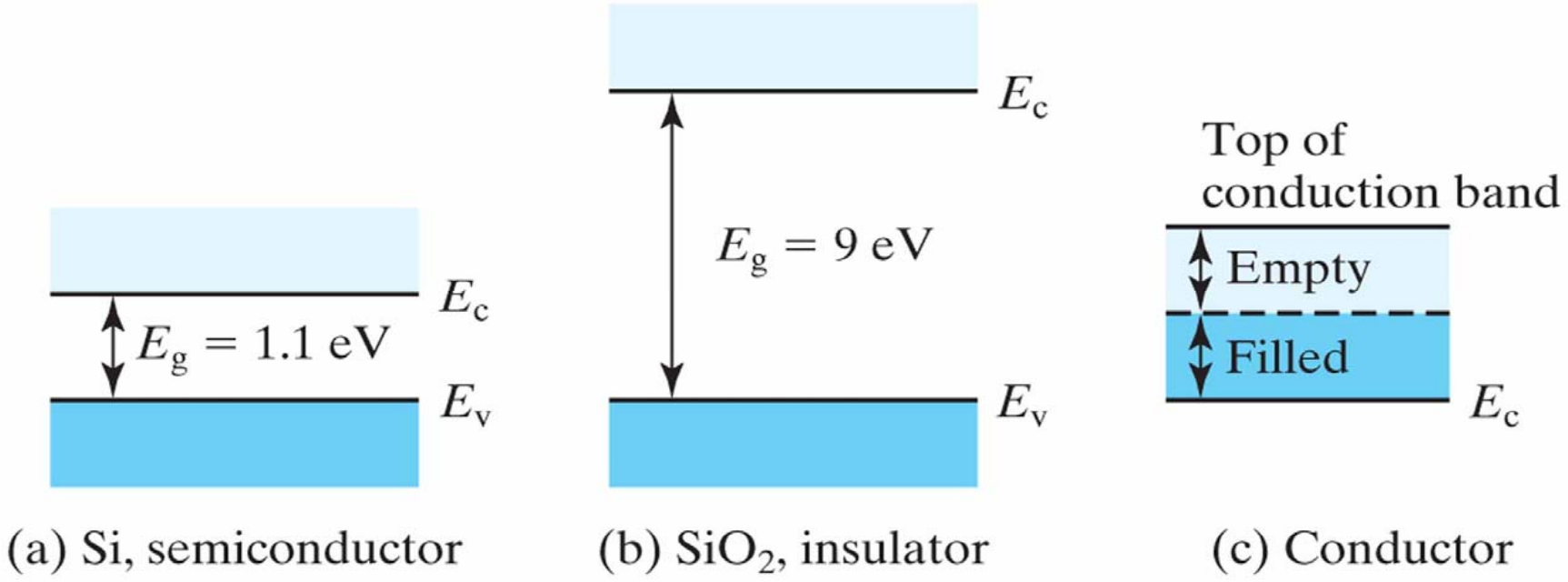
Valence Band
Highest *nearly full* band. Contributes to Conduction.
Lower Mobility *because nearly full makes it harder for electrons to move*



Lower Full Bands.
Electrons in these bands do not contribute to conduction
They do not respond to an external electric field

Source: Chenming Hu

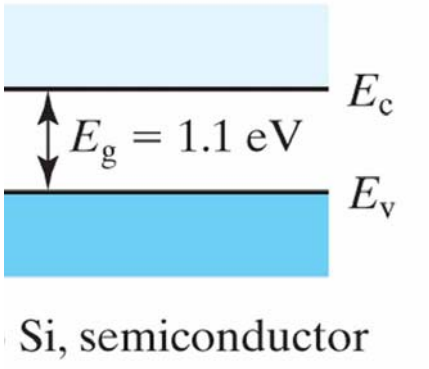
Semiconductors, Insulators, and Conductors

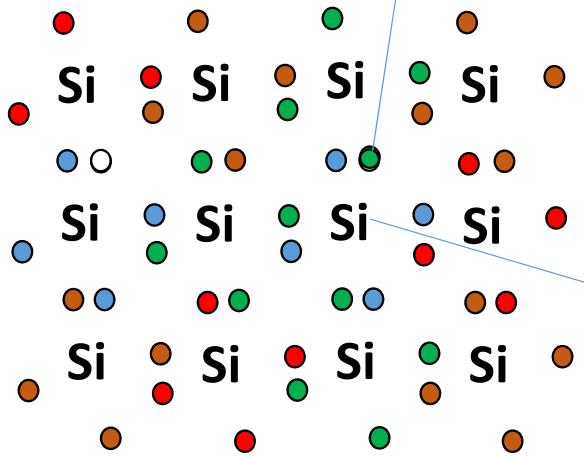


Source: Chenming Hu

Two Types of Charge Carriers Electrons and Holes

At room temperature a small fraction of electrons can break free the covalent bond and become conduction electrons





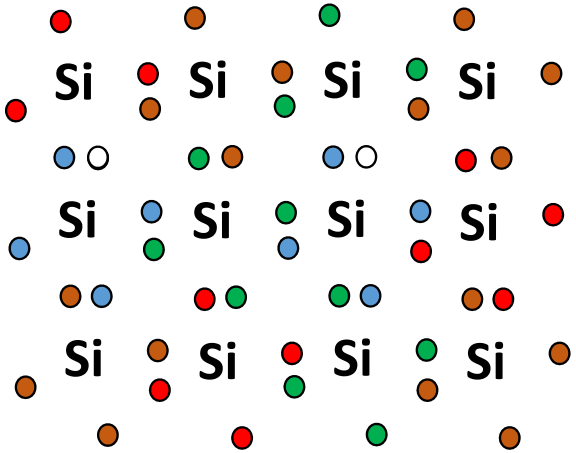
The empty slot left by the electron is called a hole
The hole is a positive charge carrier

This electron that has broken free its covalent bond is a charge carrier
Being an electron it is a negative charge carrier

This Silicon atom has been ionised and now has a net positive charge

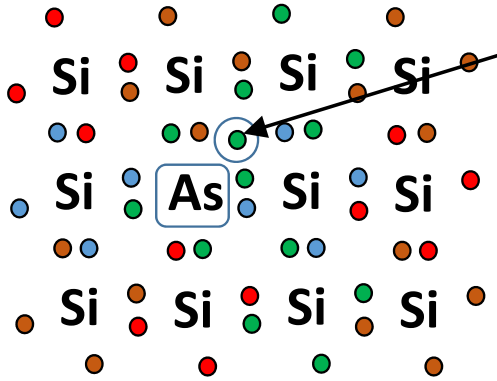
Source: Chenming Hu

Holes are the positive charge carriers in covalent band



A hole could be filled in by a neighboring electron
This effectively moves the hole in the opposite direction
This is interpreted as a positive charge moving in the opposite direction to the movement of electron

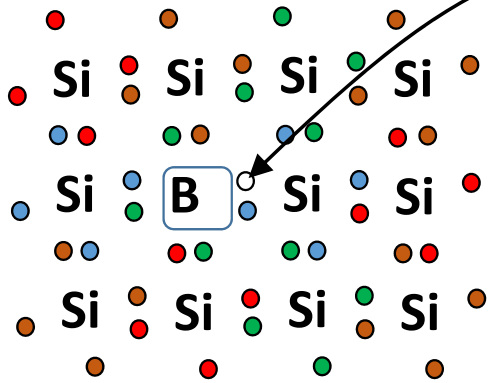
Doping – adding impurity changes the nature of Semiconductor. Donor Impurity



1. **Arsenic has five valence electrons**
2. **It has one electron more than necessary to bond in a Silicon Lattice**
3. **This electron readily breaks free to become a conduction electron**
4. **The ionisation energy that is required for such extra – donor – electrons to break free is very small ~50 mEv**
5. **At room temperature, almost all donor electrons become conduction electrons**
6. **Such donor doped material are called N type material**

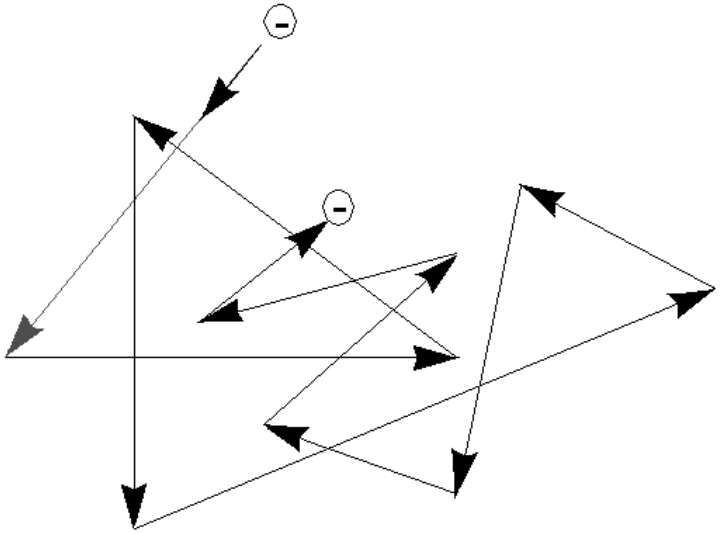
Interesting Fact: The energy with which a mosquito hits a window is 1 Tera Electron Volt

Doping – adding impurity changes the nature of Semiconductor. Acceptor Impurity



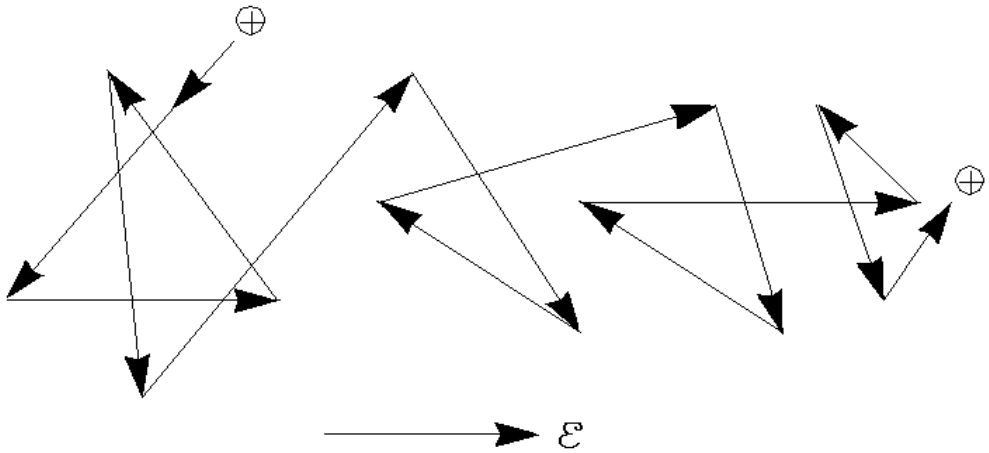
1. **Boron has three valence electrons**
2. **It has one electron lesss than necessary to bond in a Silicon Lattice**
3. **This creates plenty of holes that provides easy conduction of electricity in terms of positive charge carriers**
4. **At room temperature the valence electrons easily fill up the holes to provide conduction of holes in the opposite direction**
5. **Such acceptor doped Silicon are called P type material**

Motion of Charge Carriers



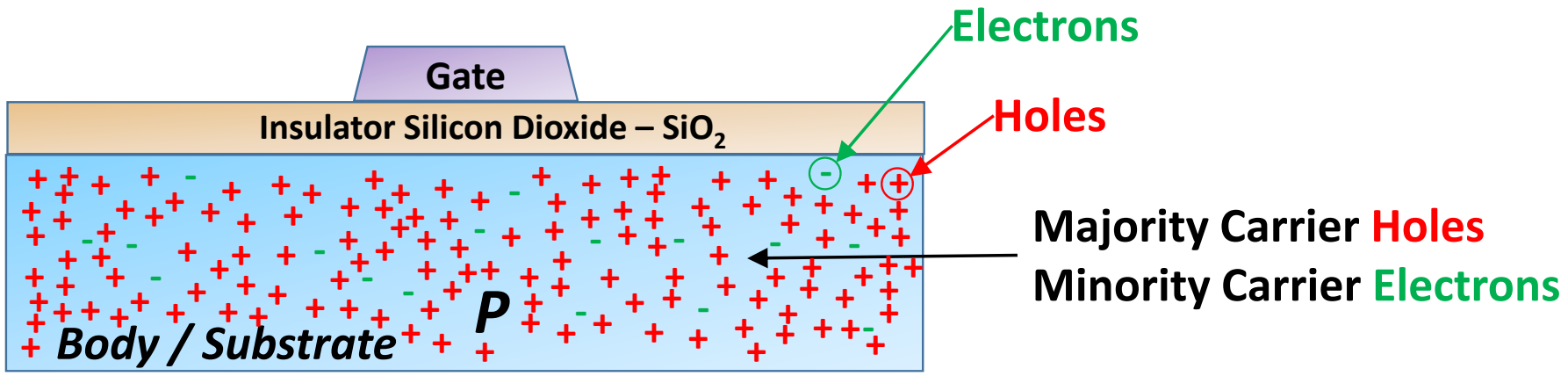
1. *In absence of electric field, the motion of charge carriers is zig-zag and the net velocity is zeron*
2. Mean time between collisions is $\tau_m \sim 0.1$ps

Source: Chenming Hu



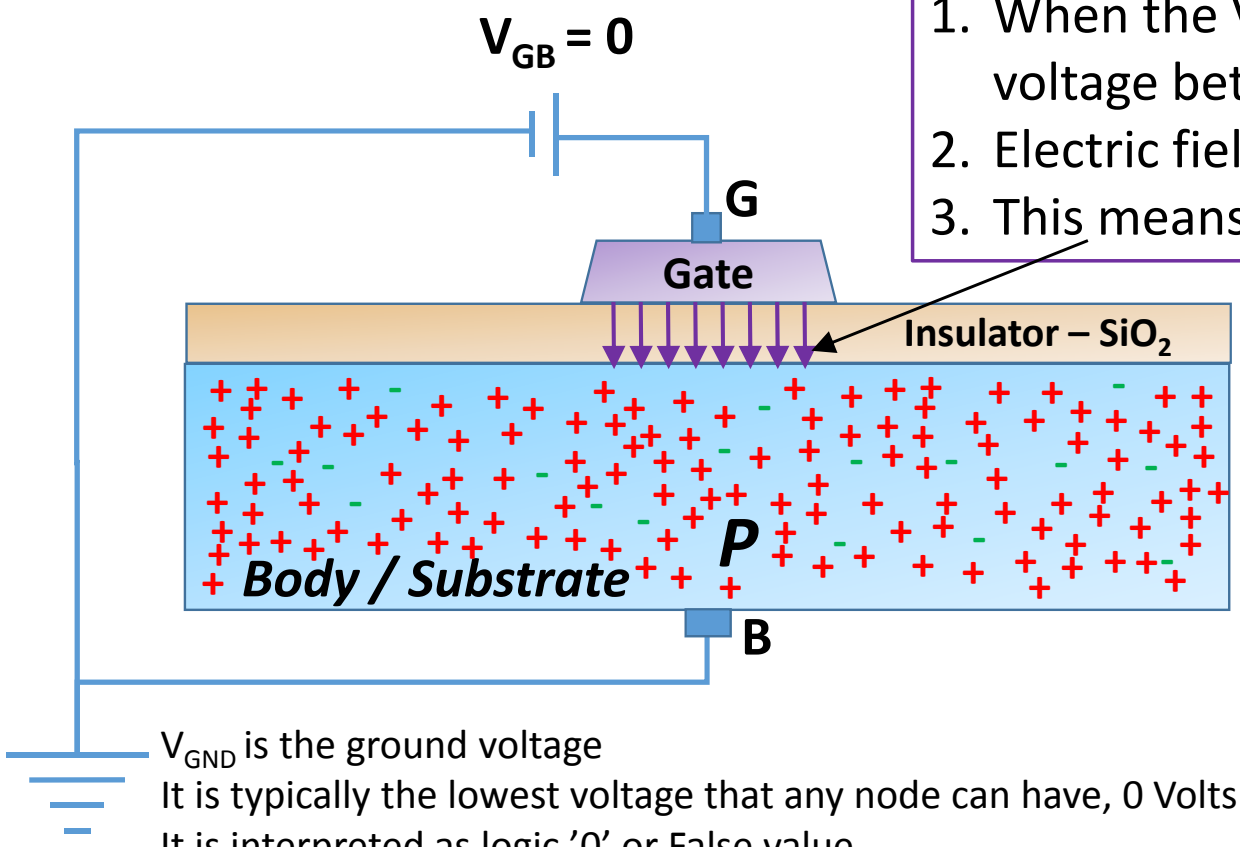
1. *When an electric field is applied, both the negative and the positive charge carriers have a net movement*
2. *The velocity is proportional to the electric field*
3. *The proportionality constant is called* ***mobility***
4. ***Mobility of electrons is nearly three times that of holes. Why ?***

Let us build a Semiconductor Switch



Let us build a Semiconductor Switch

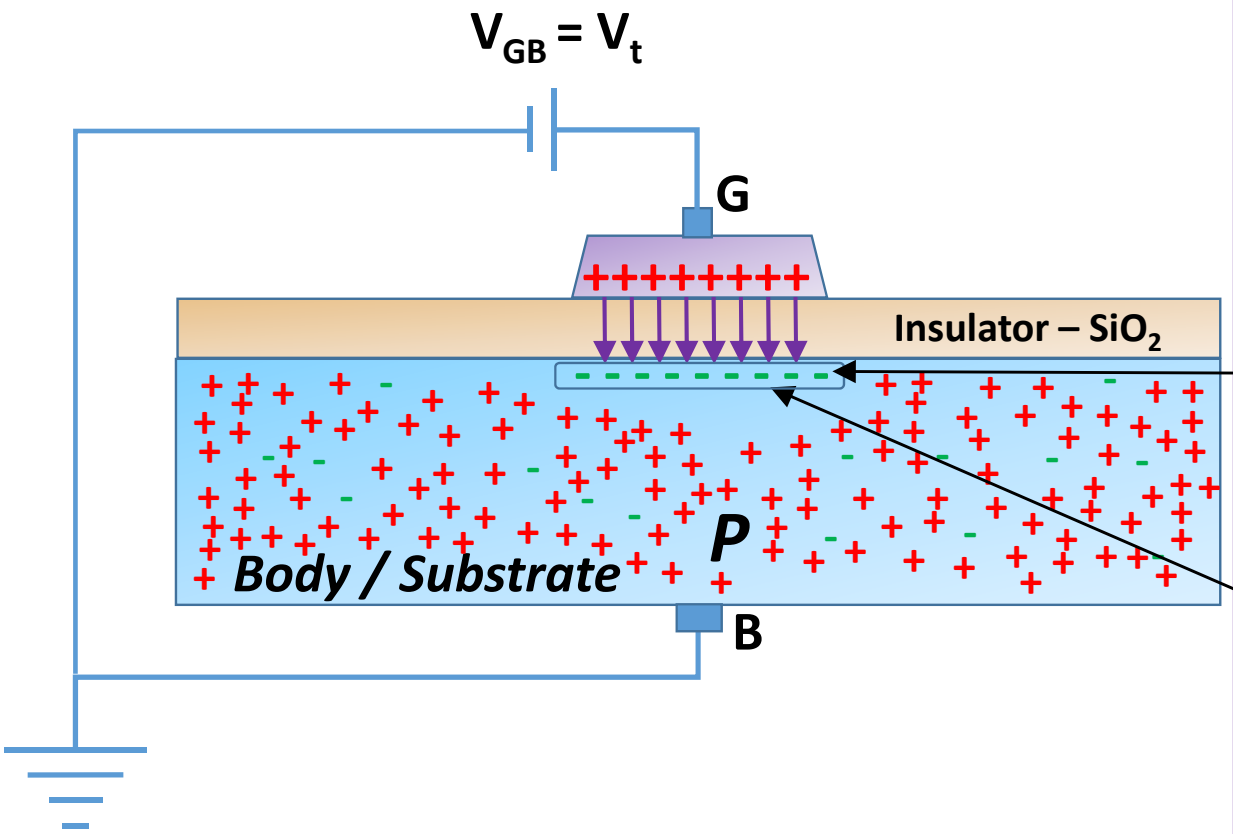
V_{GB}: Voltage between G(Gate) and B (Body)



1. When the V_{GB} is zero, there is no gradient in voltage between Gate and Body
2. Electric field is a gradient of voltage across space
3. This means that the electric field is zero

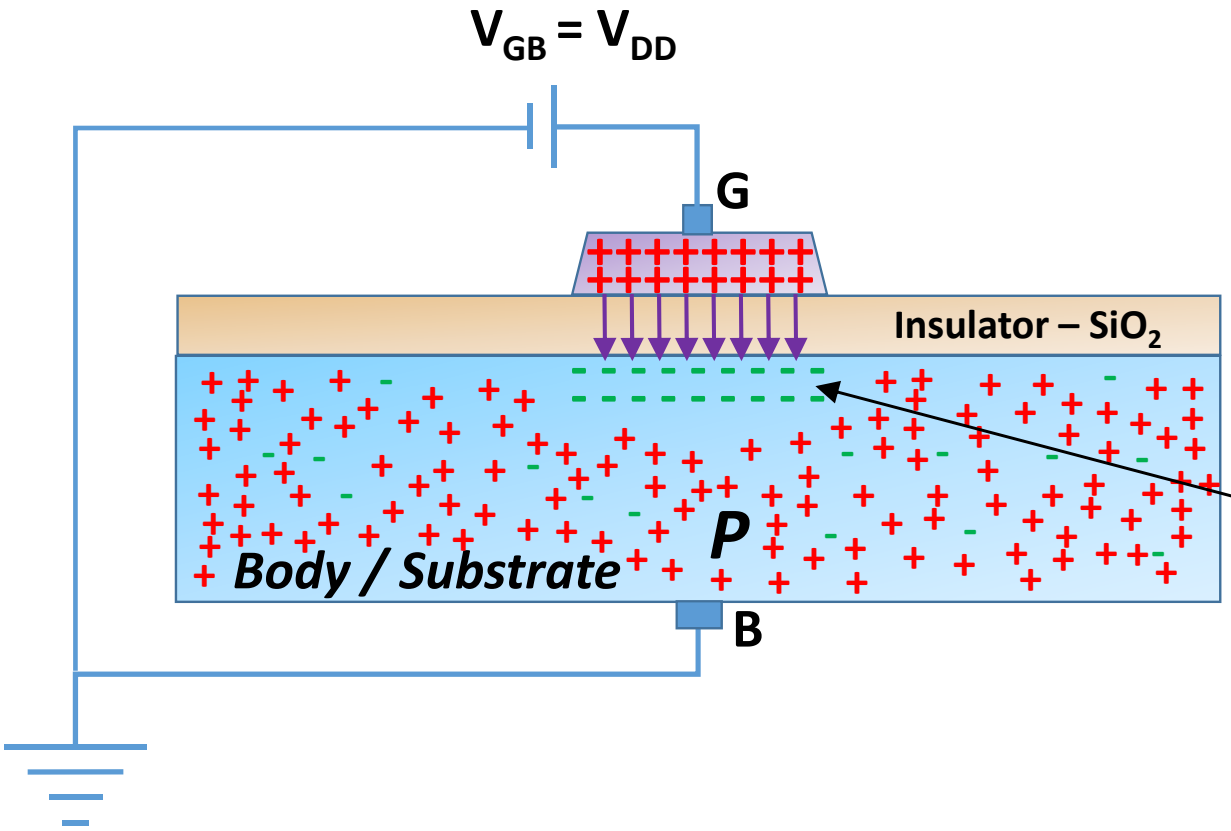
V_{GND} is the ground voltage
It is typically the lowest voltage that any node can have, 0 Volts
It is interpreted as logic '0' or False value

Let us build a Semiconductor Switch



1. When the V_{GB} is V_t (threshold voltage)
2. A gradient in voltage exists between Gate and Body
3. An electric field is created
4. The positive charge on Gate attracts the minority carriers, i.e., the electrons, to the surface of the substate
5. We say that near the surface of the Body, the P material has locally *inverted* to an N material
6. V_t is the defined as the smallest voltage at which this inversion happens

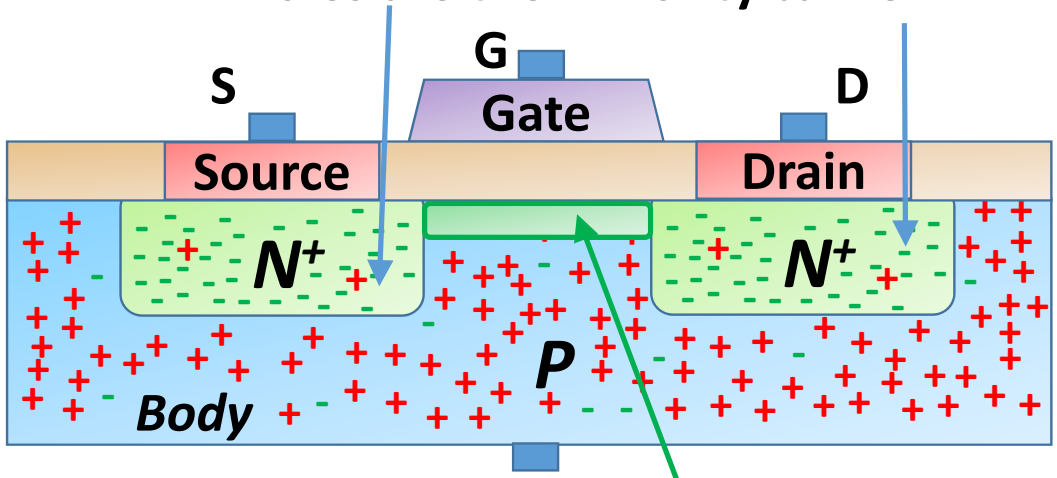
Let us build a Semiconductor Switch



1. V_{DD} is the highest voltage and is interpreted as logic 1 or True
2. When V_{GB} is raised to V_{DD}, the strongest possible electric field between Gate and Body is created
3. A strong inversion happens at the surface

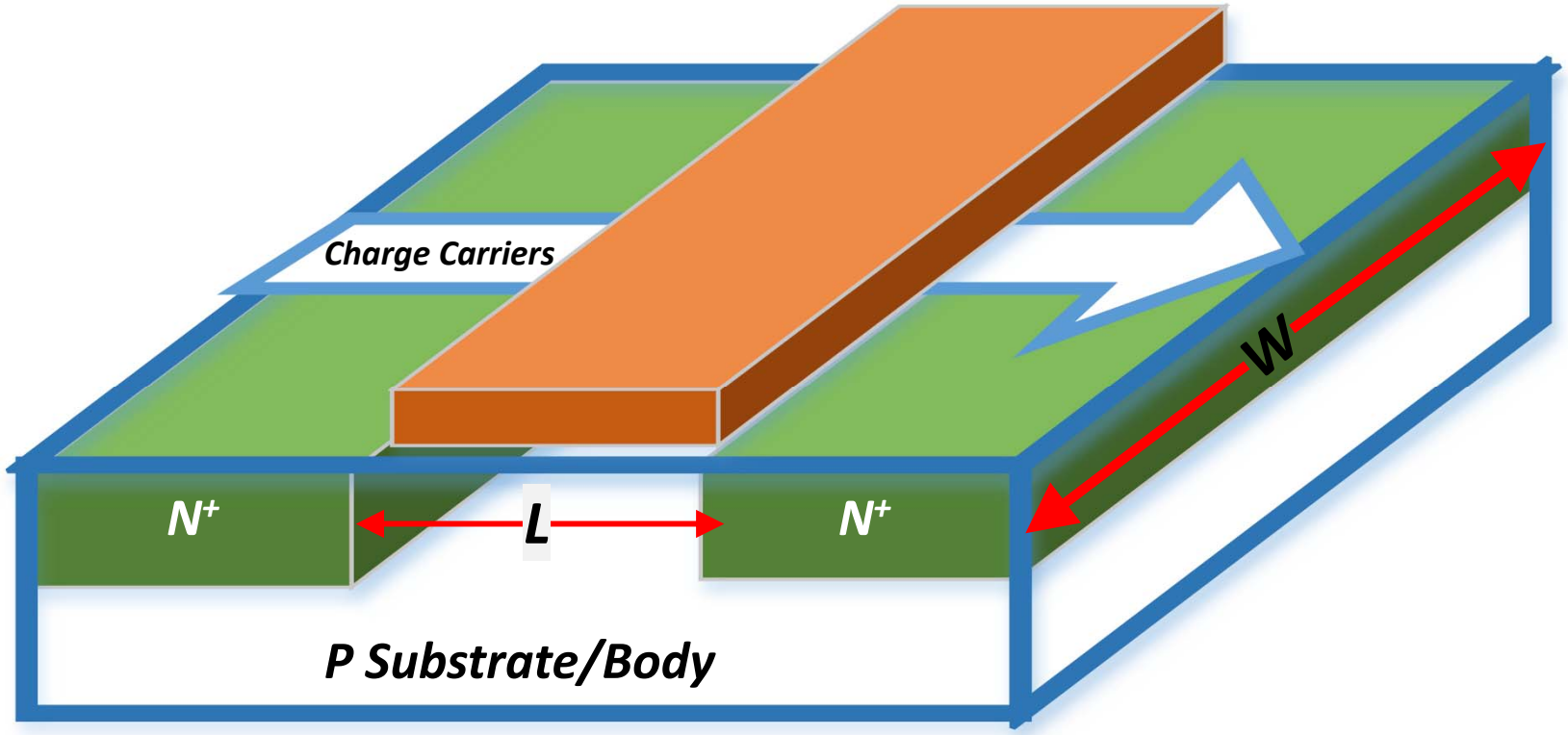
A Semiconductor Switch

Heavily Doped N Material:
Electrons are the majority carrier
Holes are the minority carrier

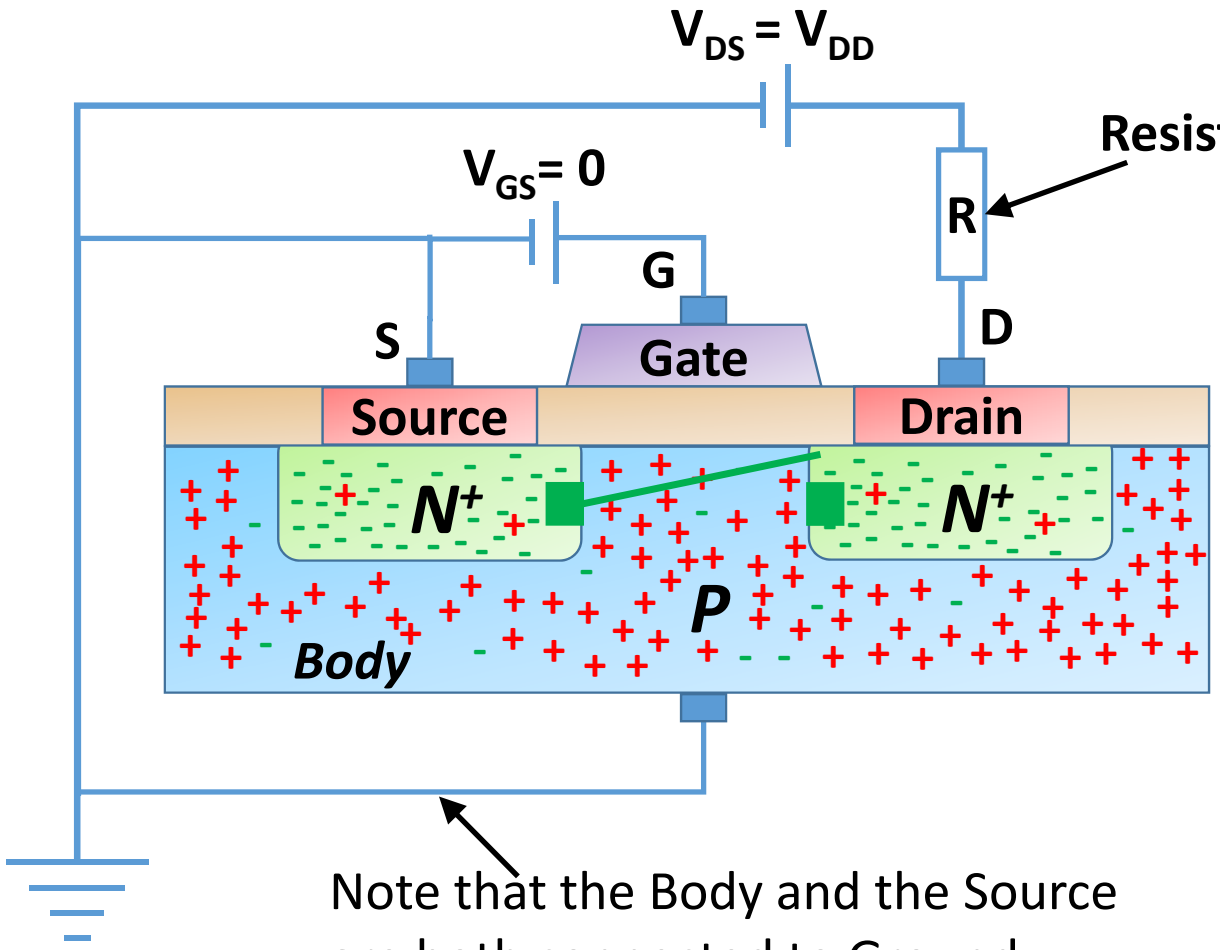


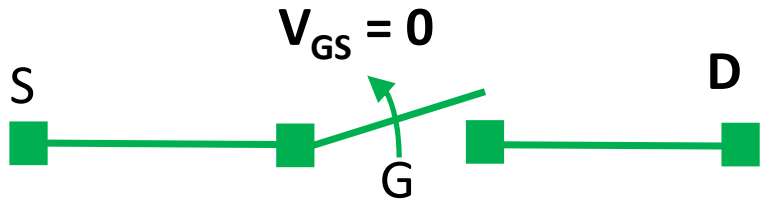
Channel
A Narrow region in Body between Source and Drain

The Semiconductor Switch is a 3D structure



How does this work as a Switch ?

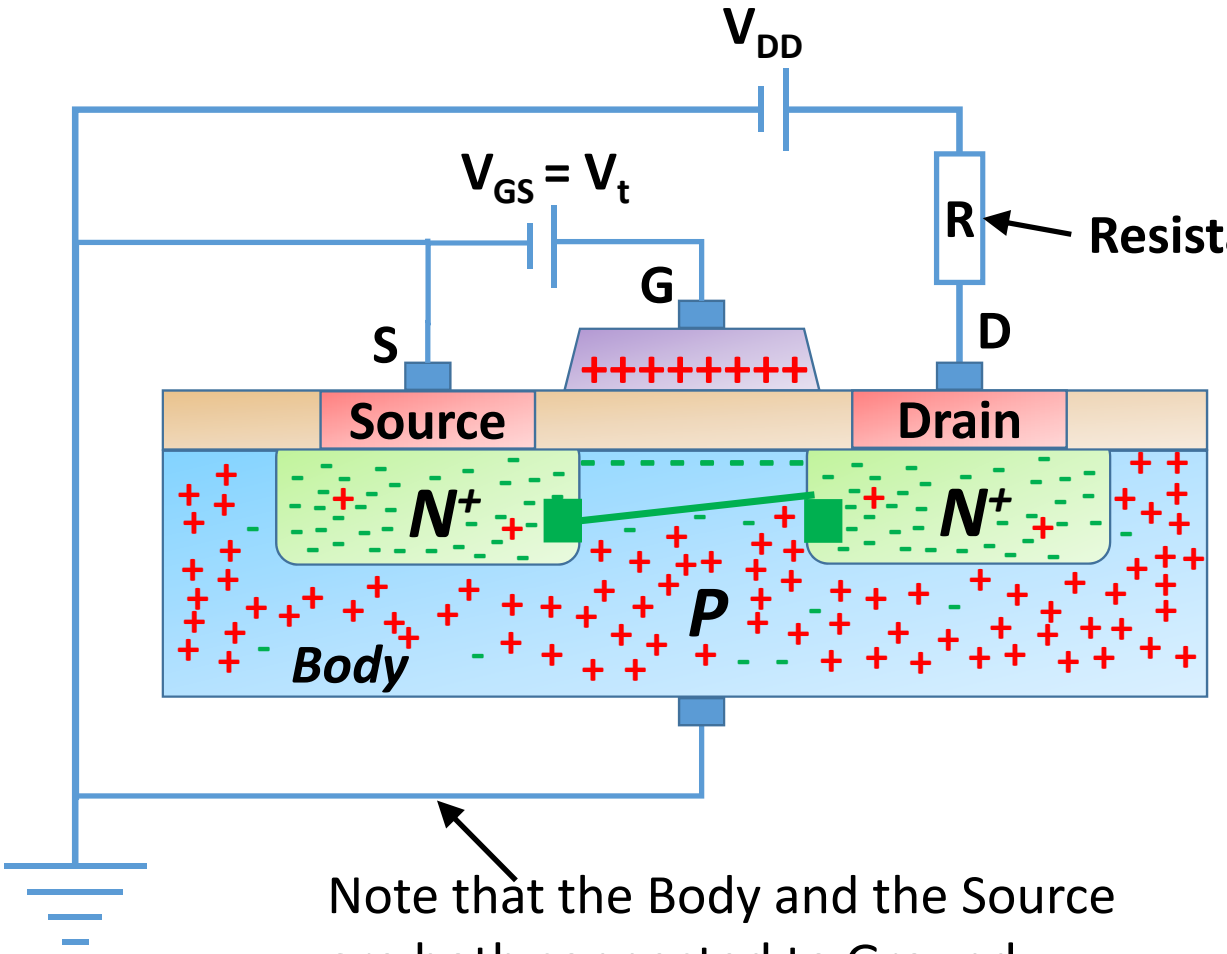




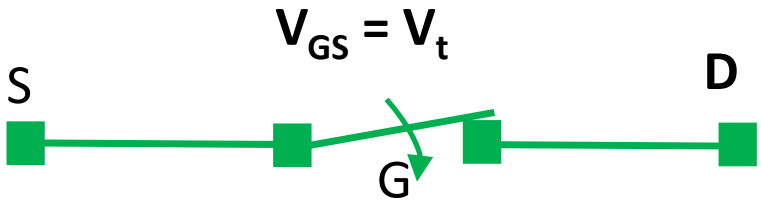
When $V_{GS} = 0$
The channel has no charge carrier of right polarity, i.e., electrons to conduct between source and drain.

Note that the Body and the Source are both connected to Ground

How does this work as a Switch ?

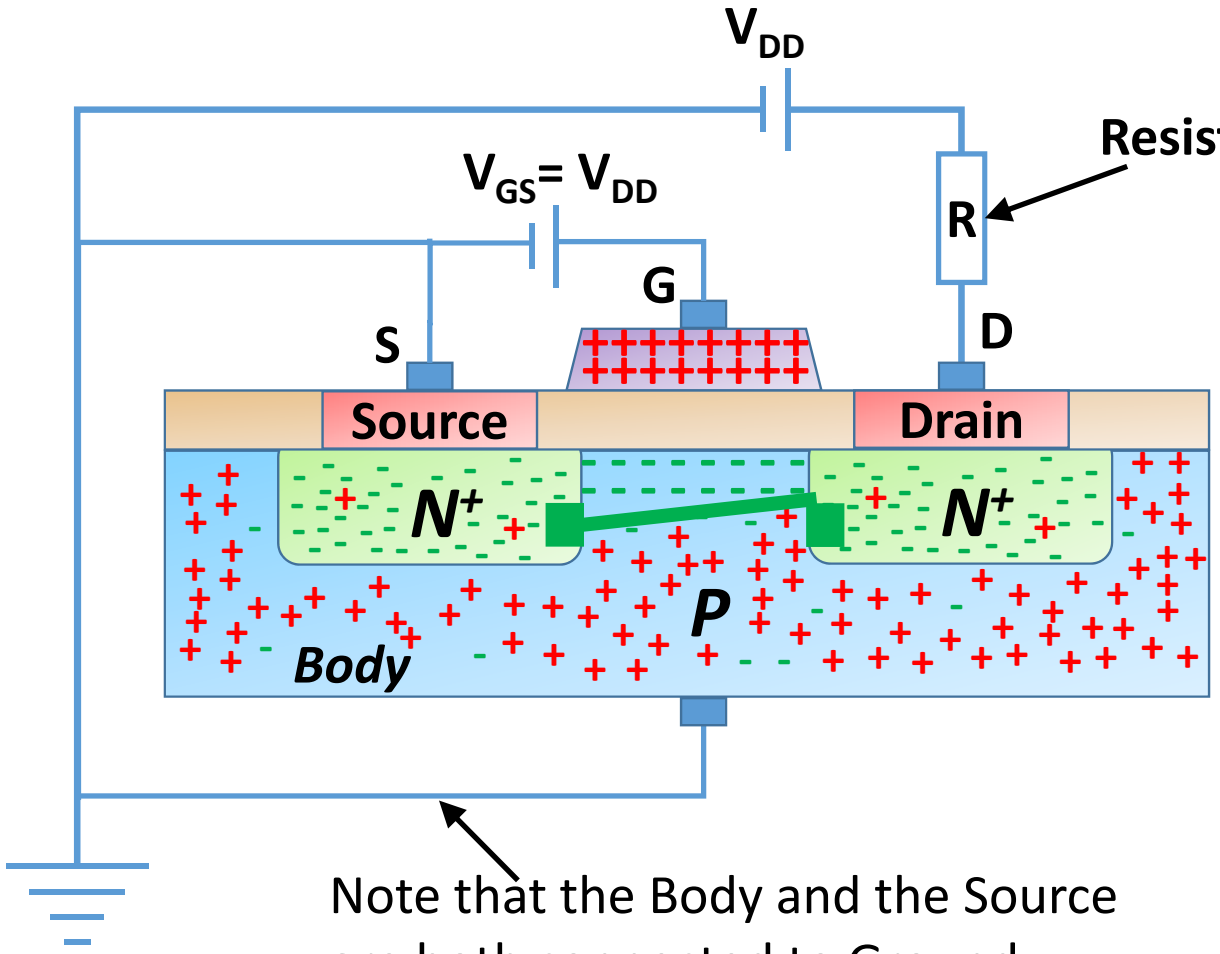


Note that the Body and the Source are both connected to Ground

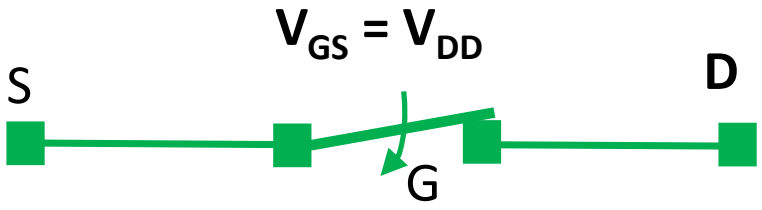


1. When $V_{GS} = V_t$ (the threshold voltage), the positive charge on gate attracts the minority carriers in substrate and forms a thin layer of conducting channel to connect source and drain
2. The electric field between Drain and Source moves the electrons from source to drain, i.e. a circuit is formed between source and drain.

How does this work as a Switch ?

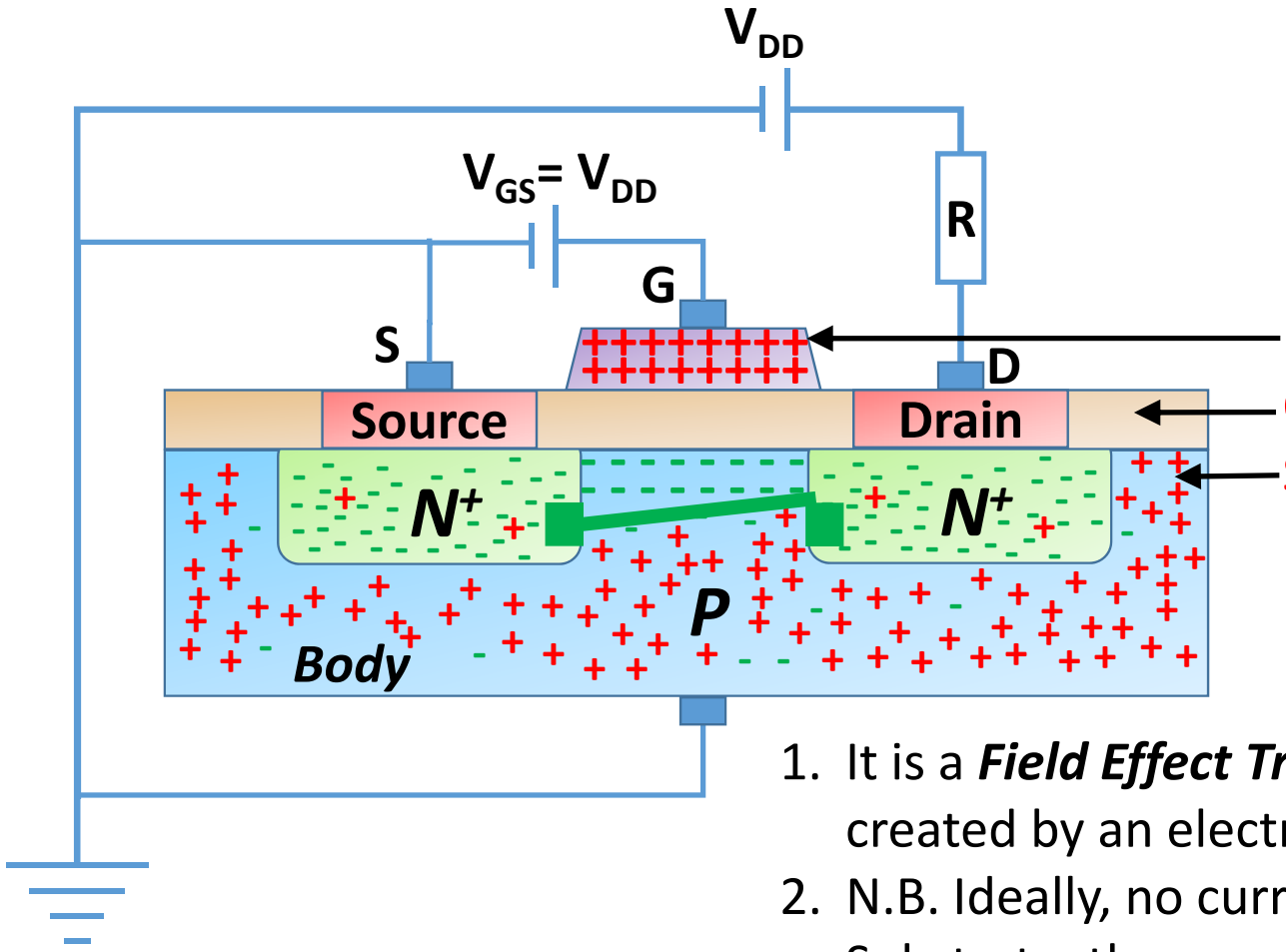


Note that the Body and the Source are both connected to Ground



1. When $V_{GS} = V_{DD}$ the large number of positive charge on gate attracts lots of minority carriers in channel to create a conducting path between source and drain
2. The voltage between Drain and Source establishes a horizontal electric field that makes the electrons flow from source to drain

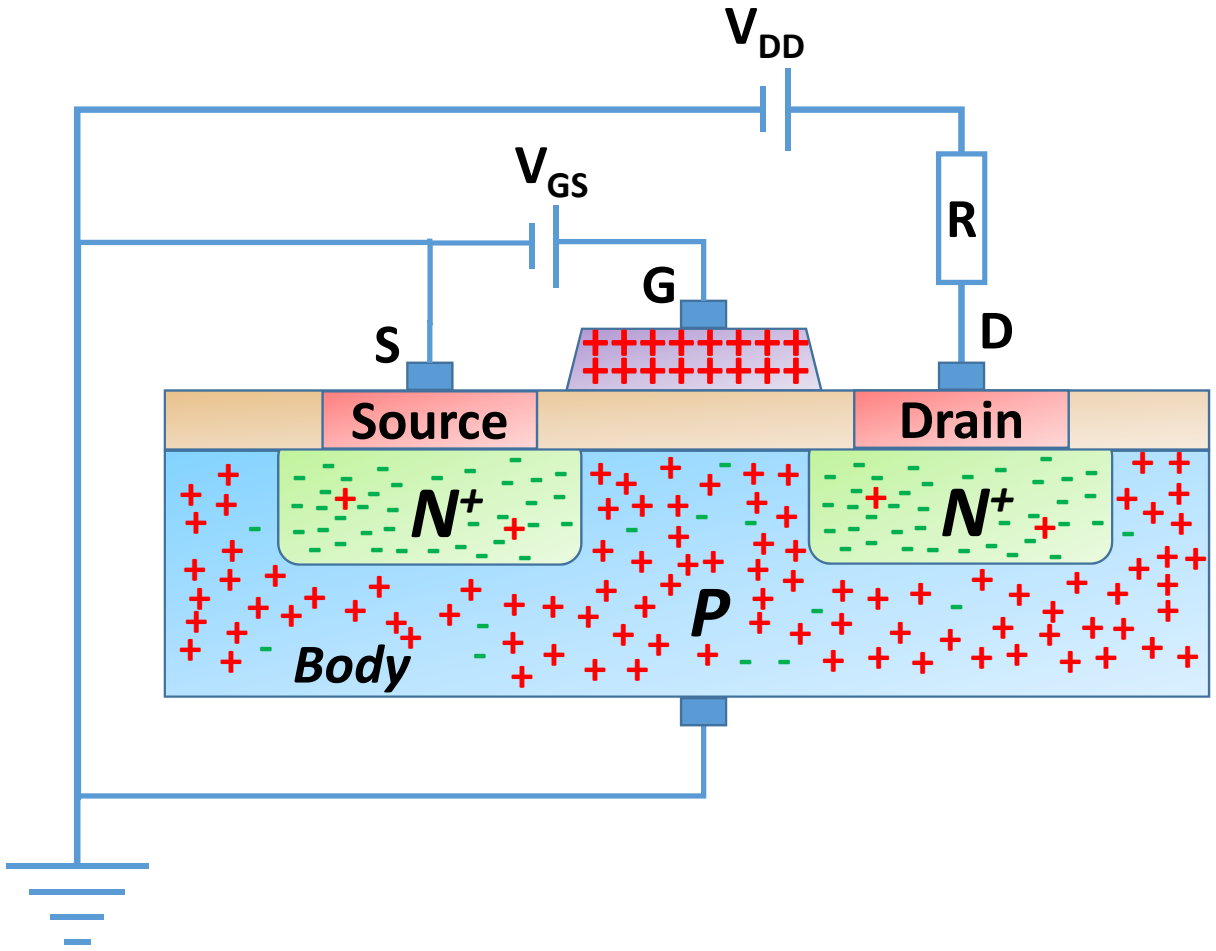
The Switch is called NMOS Field Effect Transistor (FET)

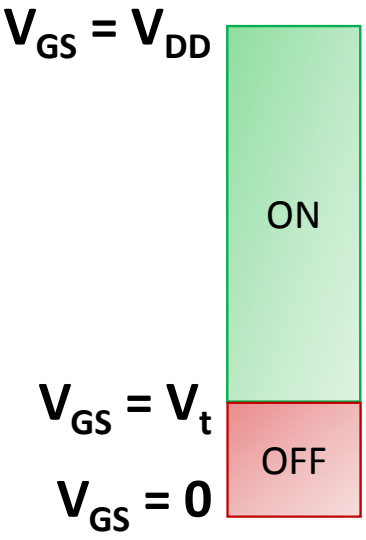


It is called NMOS because the charge carriers are N type

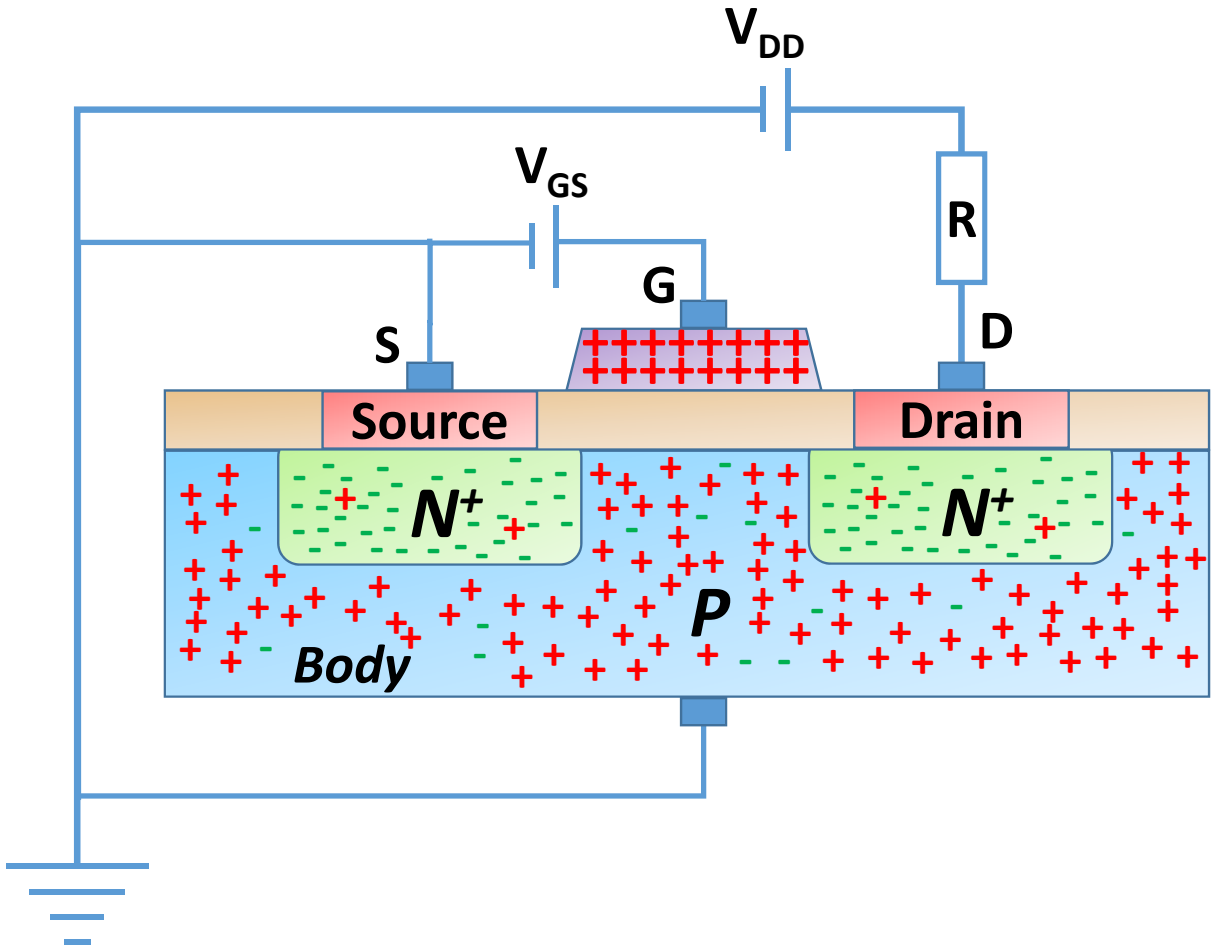
1. It is a ***Field Effect Transistor*** because the channel is created by an electric field between Gate and Body
2. N.B. Ideally, no current flows between Gate and Substrate, they are separated by an insulator

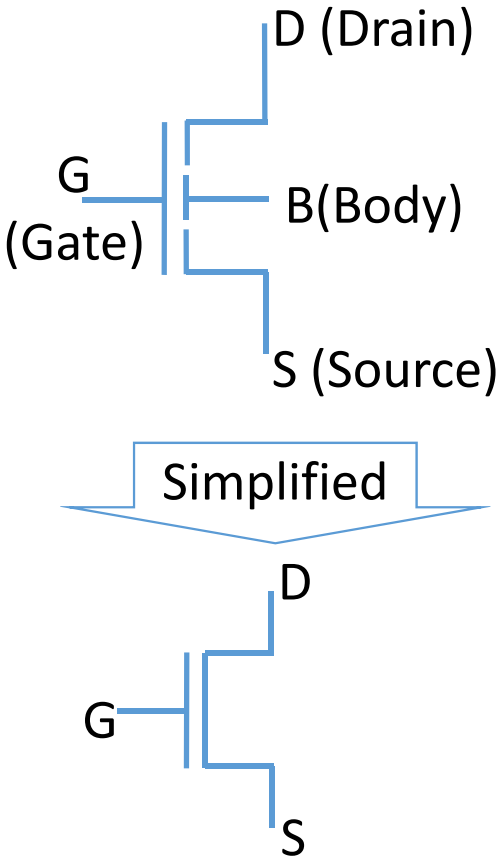
NMOS – On and Off Regions



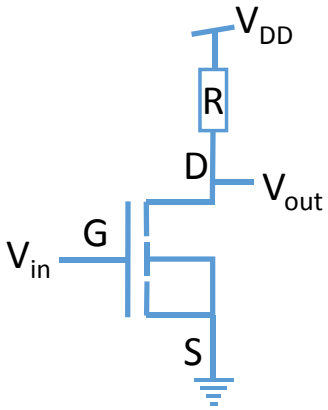


NMOS Symbol


V_{DD}
V_{GS}
R
G
S
D
Source
Drain
N^+
N^+
P
Body


D (Drain)
G
(Gate)
B(Body)
S (Source)
Simplified
D
G
S

NMOS as a Switch



V_{in}	V_{out}
0	1
1	0

When $V_{in} < V_t$ typically this means V_{in} = 0 volts (logic 0, false)

$V_{out} = V_{DD}$

Source and Drain are disconnected

Assuming no current flows through R, the voltage drop across R is zero.

This makes $V_{out} = V_{DD}$

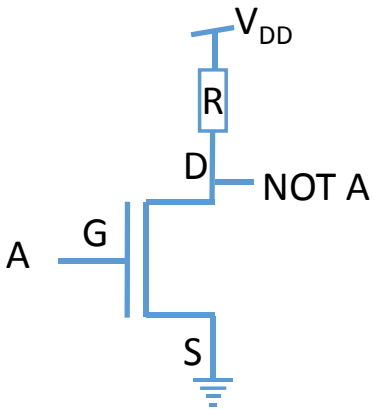
When $V_{in} > V_t$ typically this means $V_{in} = V_{DD}$ (logic 1, true)

$V_{out} = 0$

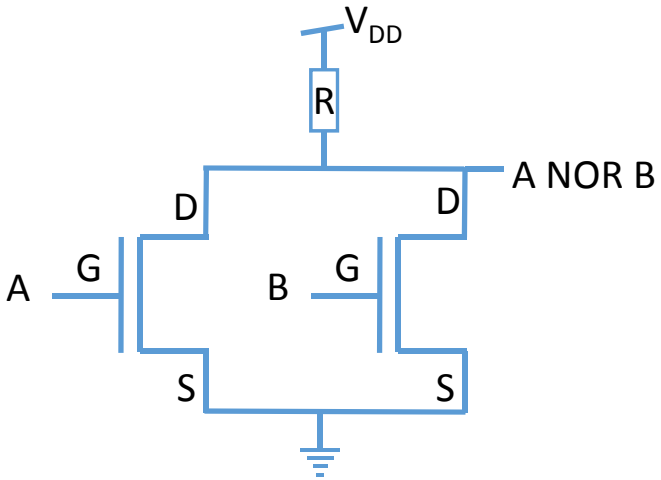
Source and Drain are connected. Ideally, this means that drain (D) is connected to source and source is connected to GND.

This makes $V_{out} = 0$

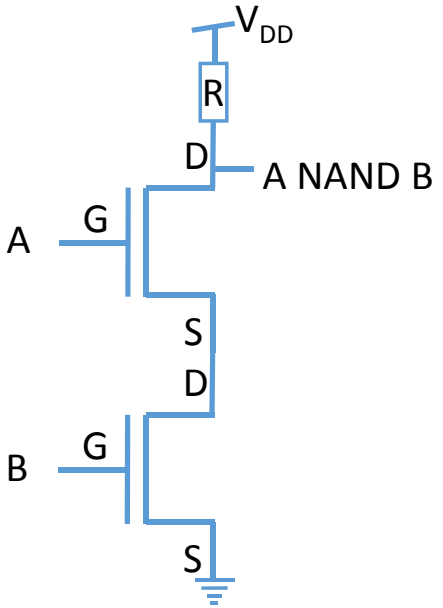
NMOS can implement NOT, NAND and NOR



A	NOT A
0	1
1	0

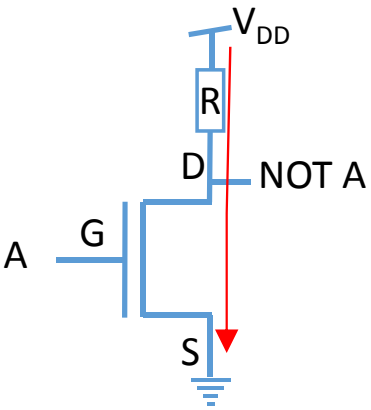


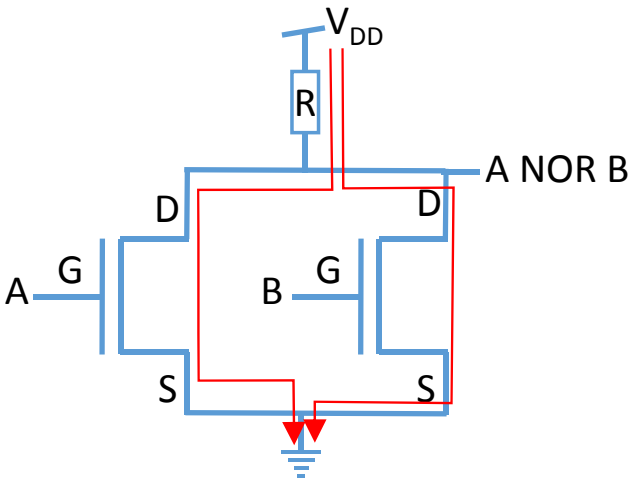
A	B	A NOR B_t
0	0	1
0	1	0
1	0	0
1	1	0

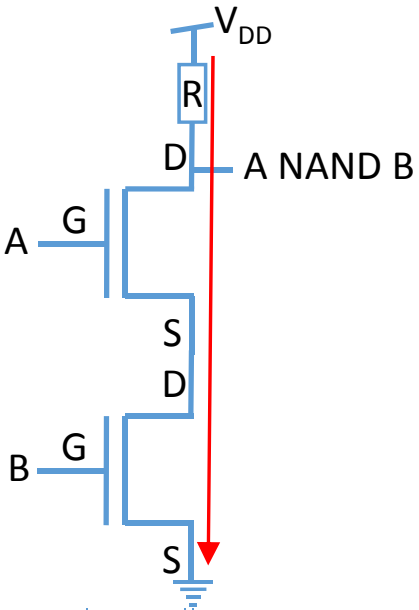


V_A	V_B	V_{out}
0	0	1
0	1	1
1	0	1
1	1	0

What is wrong with NMOS ?







Replace V_x with X to simplify Notation

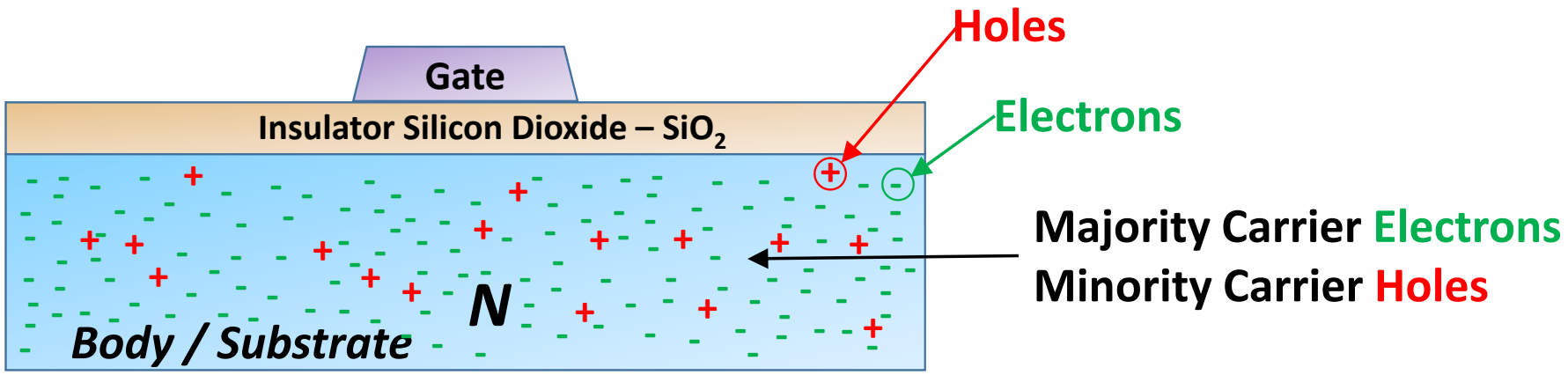
A	NOT A
0	1
1	0

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

1. When a logic operation is performed, it is justified to consume power
2. However, when no operation is being done, consuming power is considered to be a leakage
3. In these states, enclosed in red, the NMOS transistors, continuously draw current, even if the input does not change and it has to peform no logic operation

PMOS Switch

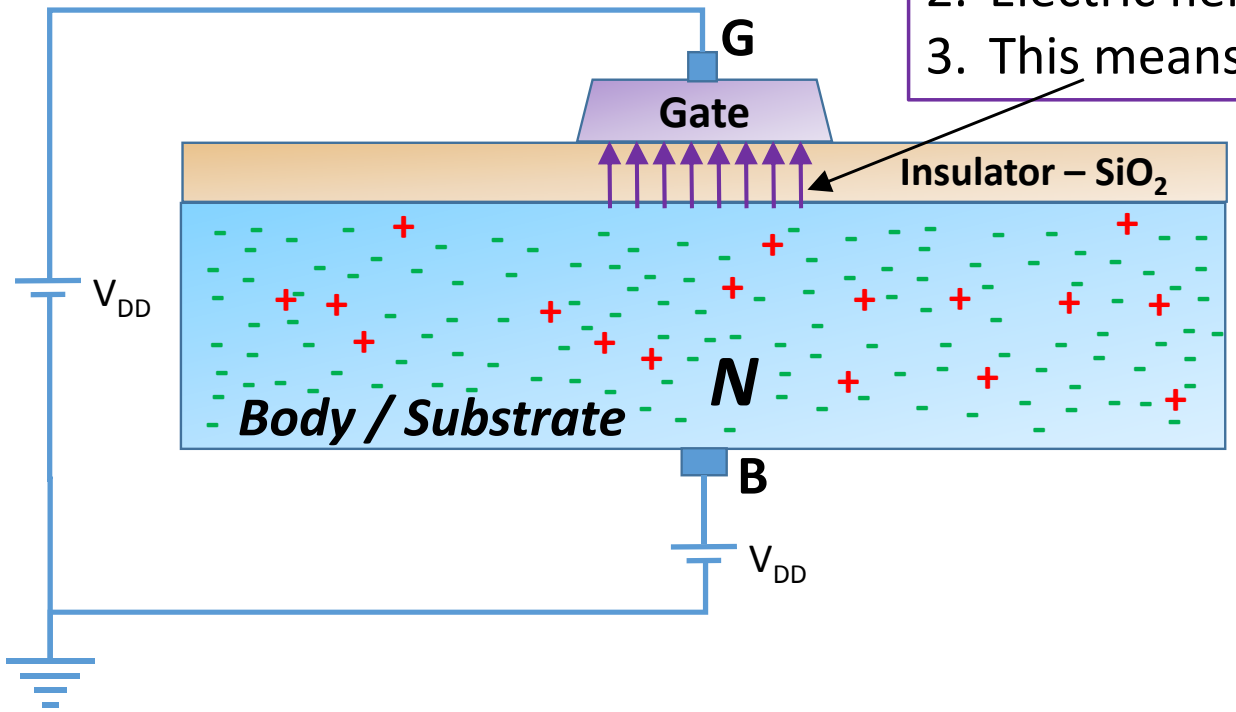

Gate
Insulator Silicon Dioxide – SiO_2
N
Body / Substrate
Holes
Electrons
Majority Carrier Electrons
Minority Carrier Holes

Let us build a Semiconductor Switch

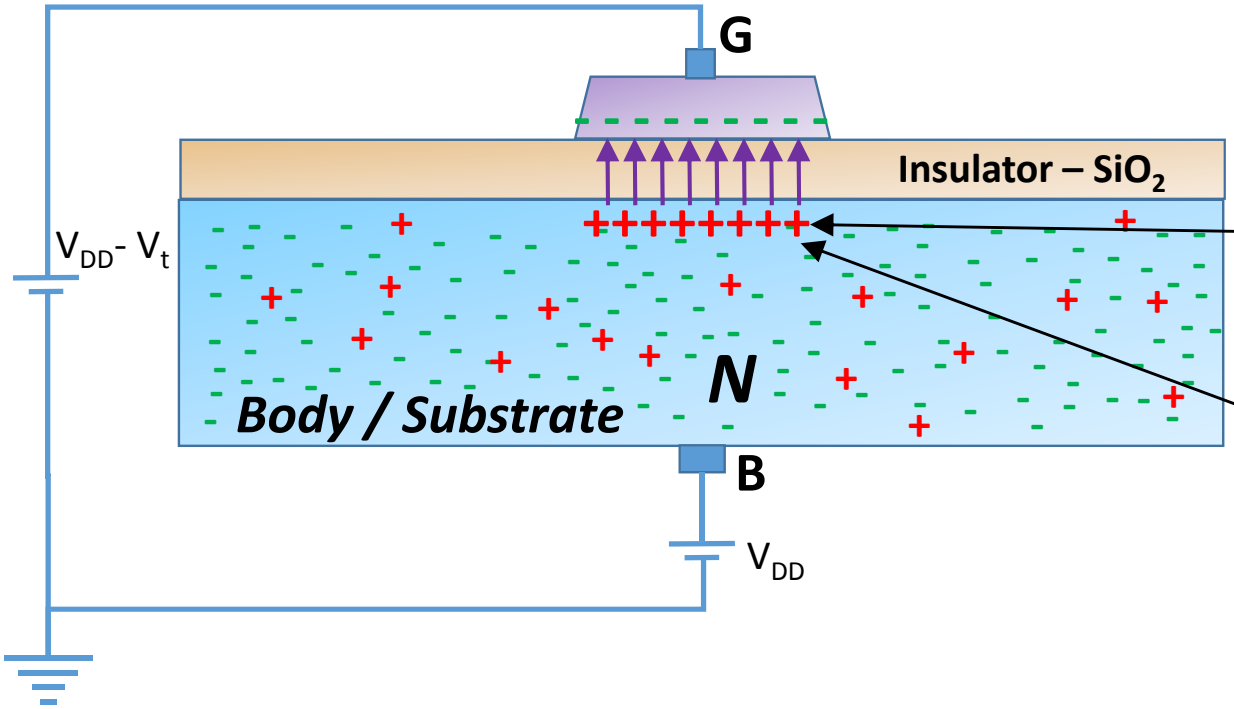
V_{GB}: Voltage between G(Gate) and S (Substrate)

$V_{GB} = V_{DD}$

1. When the V_{GB} is zero, there is no gradient in voltage between Gate and Body
2. Electric field is a gradient of voltage across space
3. This means that the electric field is zero

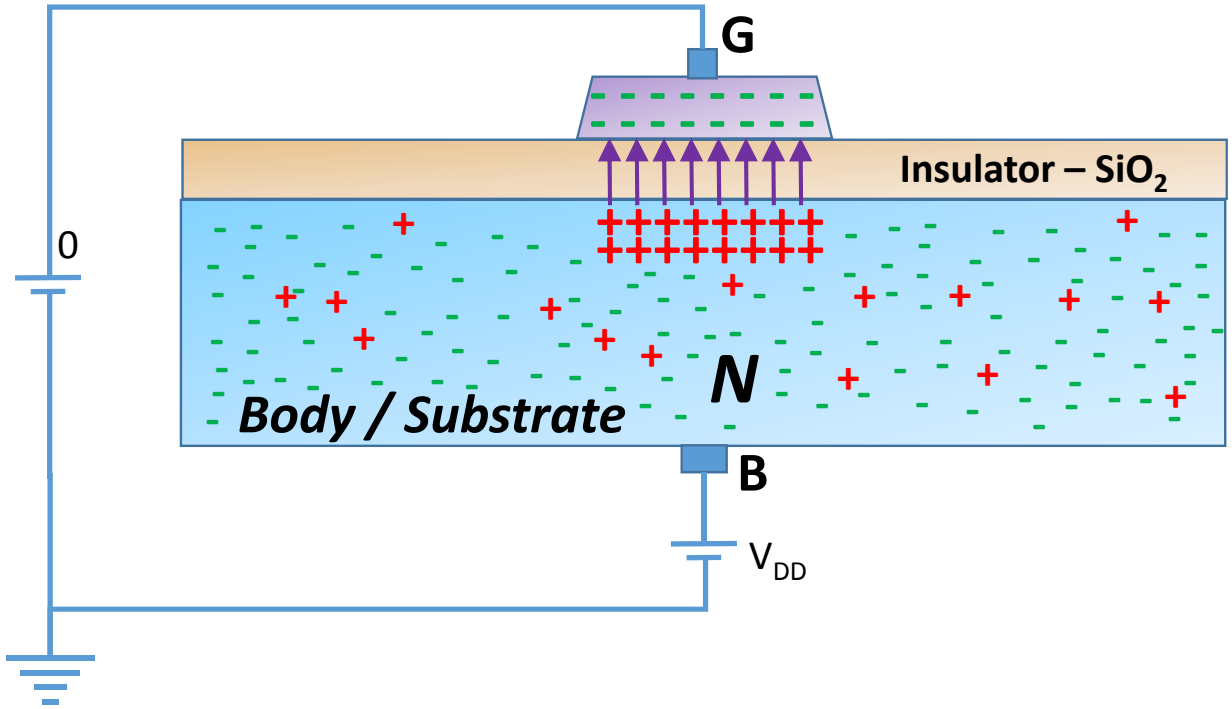


Let us build a Semiconductor Switch



1. When the V_{GB} is V_t- V_{DD} (negative voltage because $V_t < V_{DD}$)
2. A gradient in voltage exists between Gate and Body
3. An electric field is created
4. The negative charge on Gate attracts the minority carriers, i.e., the holes, to the surface of the substate
5. We say that near the surface of the Body, the N material has locally inverted to an P material
6. V_t is the defined as the largest negative voltage difference between Gate and Body at which the inversion happens

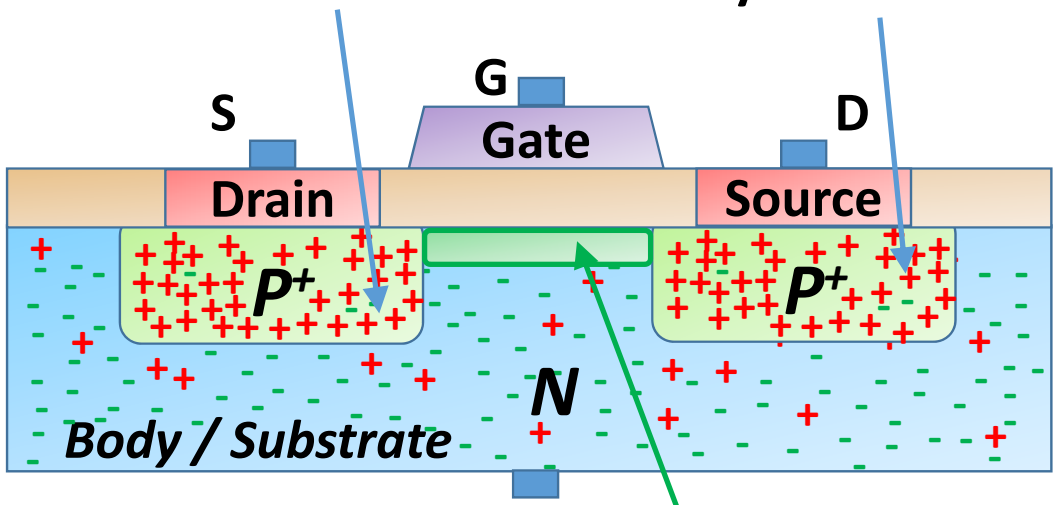
Let us build a Semiconductor Switch



1. When V_{GB} is lowered to $-V_{DD}$, the strongest possible electric field between Gate and Body is created
2. A strong inversion happens at the surface

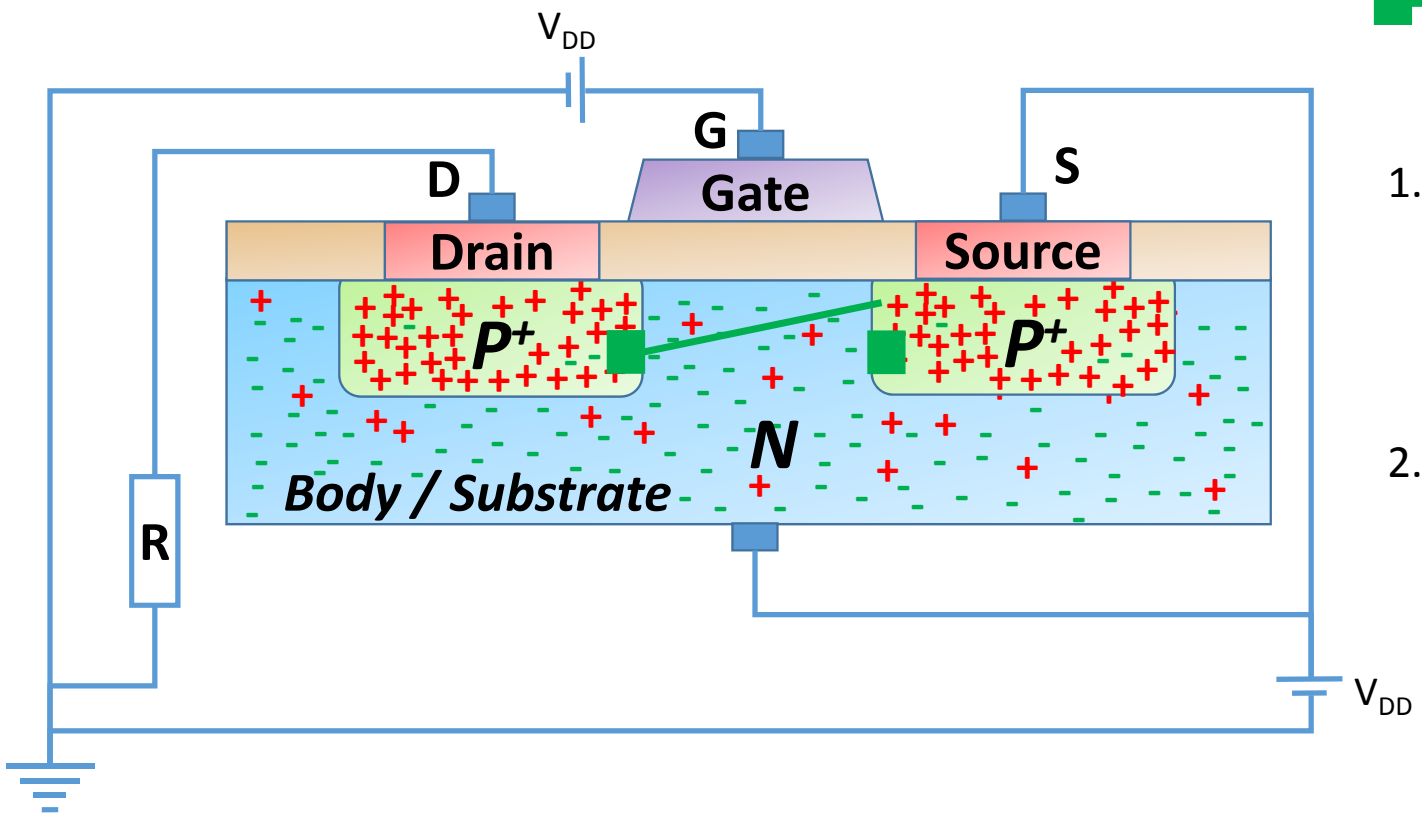
PMOS

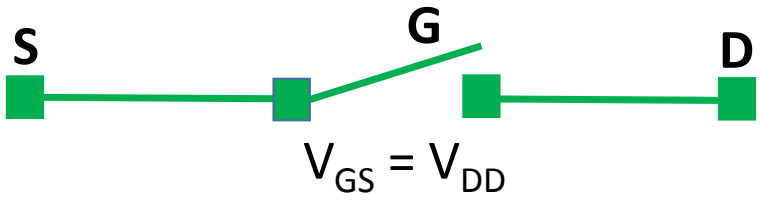
Heavily Doped P Material (P^+):
Holes are the majority carrier
Electrons are the minority carrier



Channel
A Narrow region in Body between Source and Drain

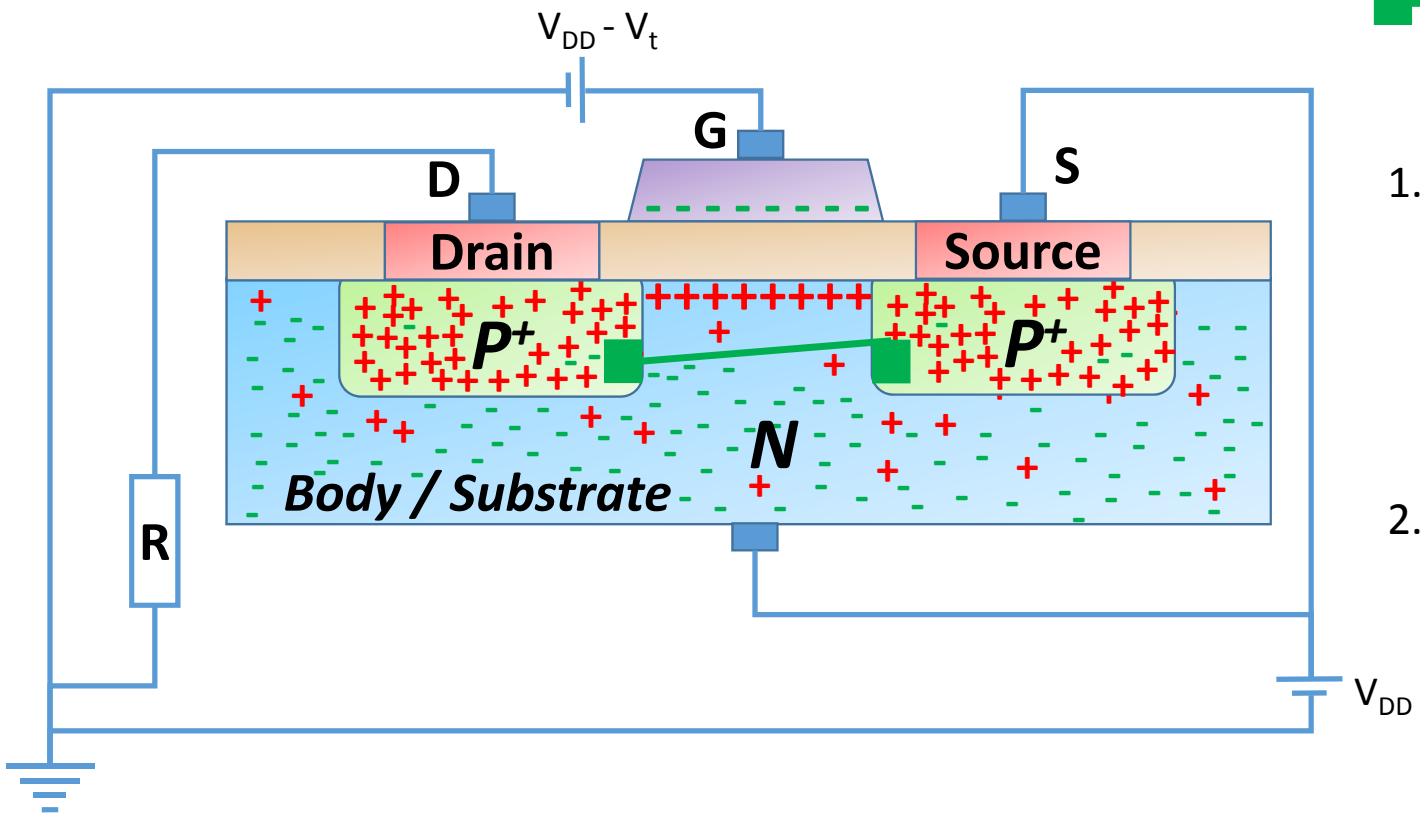
PMOS – Off Mode

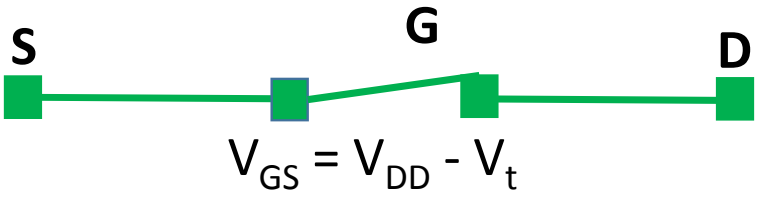




1. When the Gate and Body are both at V_{DD}, there is no electric field between Gate and Body to create a conducting channel between and Source and Drain
2. The source and drain are disconnected

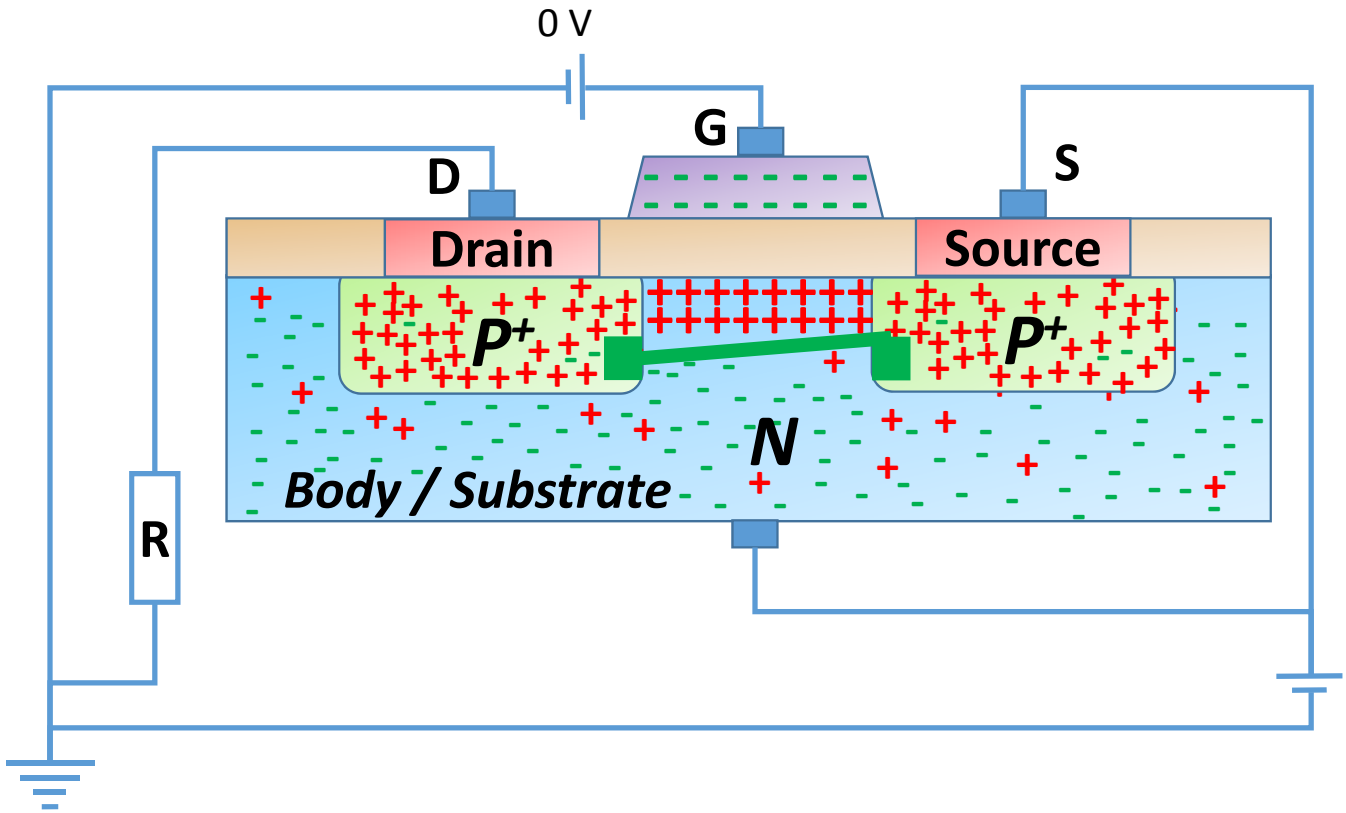
PMOS – at Threshold Voltage

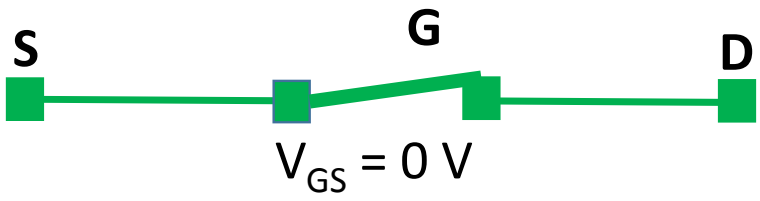




1. When the Gate Voltage is lowered to $V_{DD} – V_t$, an electric field exists between Gate and Body to cause an inversion at the surface and create a conductin channel between source and drain
2. The source and drain are now connected

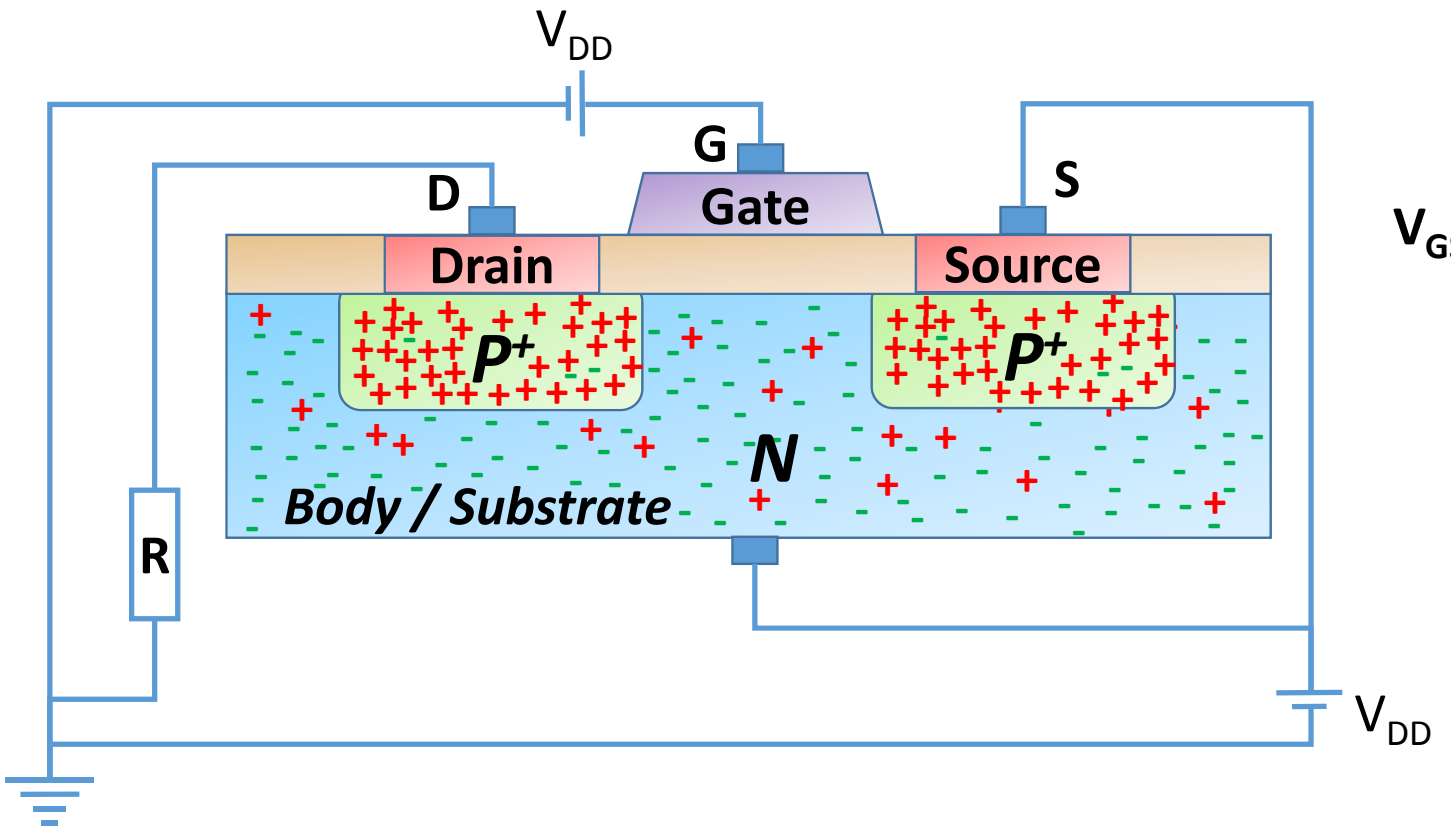
PMOS – ON Mode

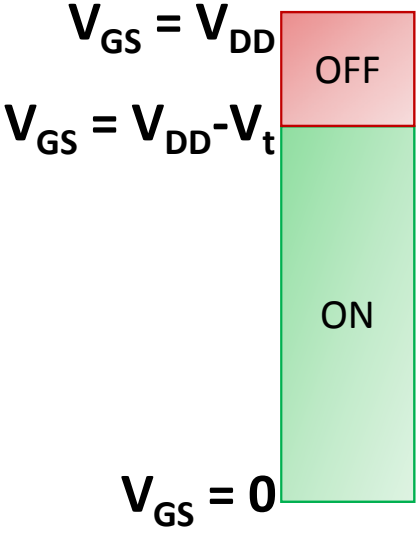




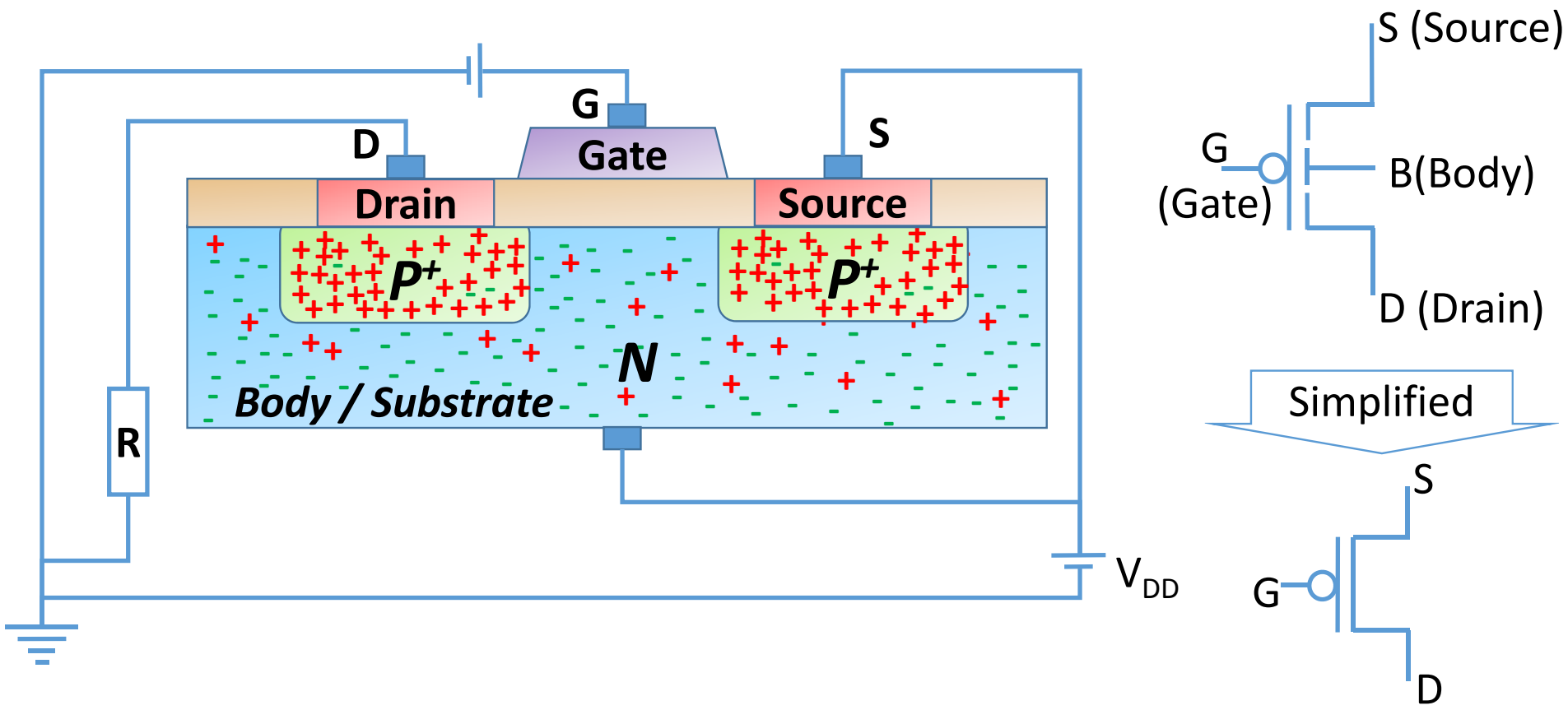
1. When the Gate Voltage is lowered to 0 V, the strongest possible electric field exists between Gate and Body to cause a strong inversion at the surface and create a good conducting channel between source and drain
2. The source and drain are now connected and the PMOS transistor is in full ON mode

PMOS – On and Off Regions



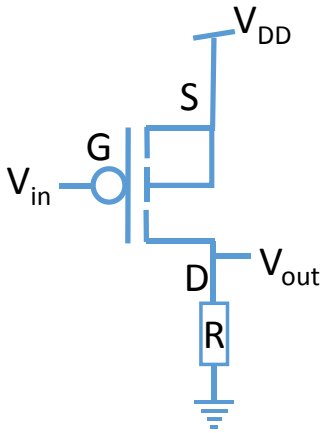


PMOS Symbol


G
Gate
D
S
Drain
Source
P+
P+
N
Body / Substrate
R
VDD

S (Source)
G
(Gate)
B(Body)
D (Drain)
Simplified
S
G
D

PMOS as a Switch



When $V_{in} > V_t$ typically this means $V_{in} = V_{DD}$ volts (logic 1, True)

$V_{out} = 0$ (Logic 0, False)

Source and Drain are disconnected

Assuming no current flows through R, the voltage drop across R is zero.

This makes $V_{out} = 0$ V

V_{in}	V_{out}
1	0
0	1

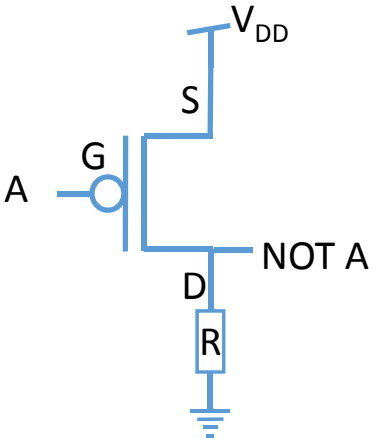
When $V_{in} < V_t$ typically this means $V_{in} = 0$ (logic 0, False)

$V_{out} = 1$

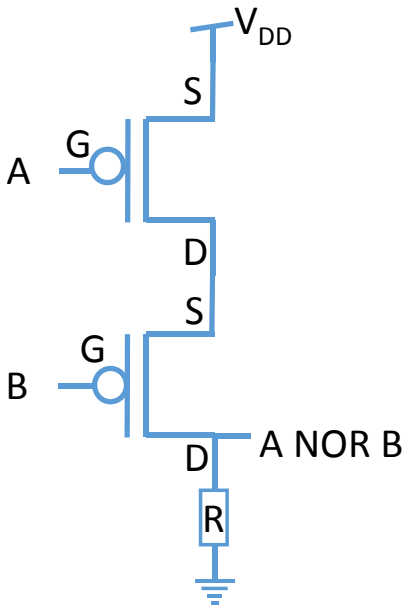
Source and Drain are connected. Ideally, this means that drain (D) are at at the same Voltage. Since Source is connected to V_{DD}, Drain is also connected to V_{DD}

This makes $V_{out} = 1$

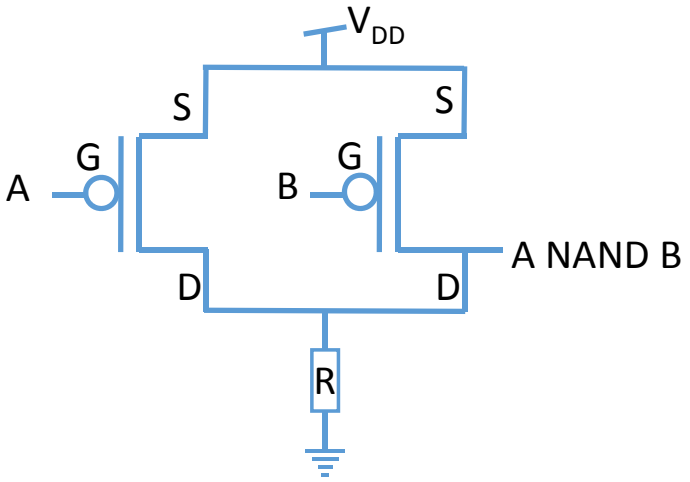
PMOS can implement NOT, NAND and NOR



A	NOT A
1	0
0	1

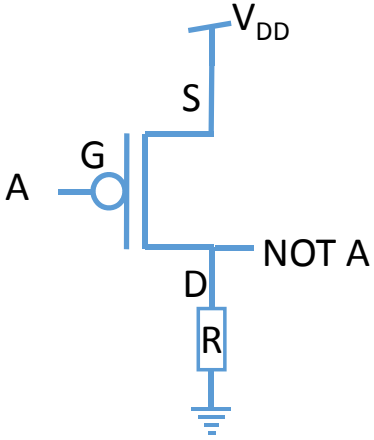


A	B	A NOR B
1	1	0
0	1	0
1	0	0
0	0	1

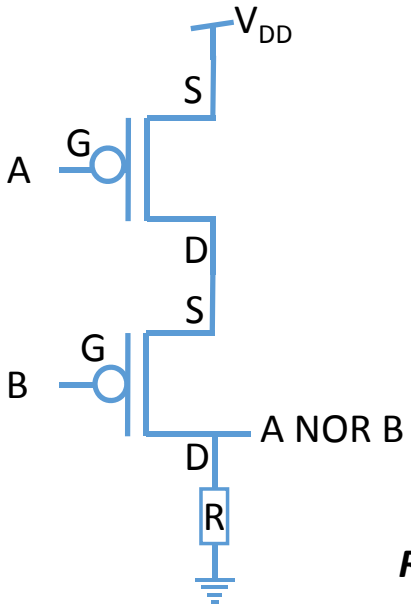


A	B	A NAND B
1	1	0
0	1	1
1	0	1
0	0	1

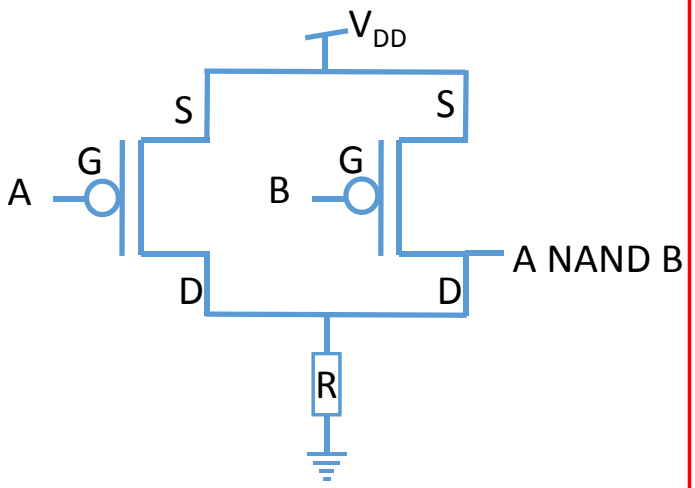
What is wrong with PMOS ?



A	NOT A
1	0
0	1



A	B	A NOR B
1	1	0
0	1	0
1	0	0
0	0	1

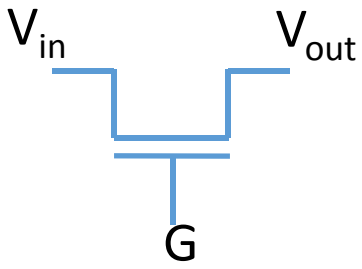


Replace V_x with X to simplify Notation

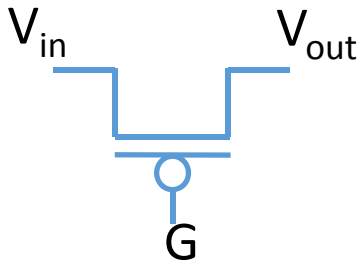
A	B	A NAND B
1	1	0
0	1	1
1	0	1
0	0	1

1. When a logic operation is performed, it is justified to consume power
2. However, when no operation is being done, consuming power is considered to be a leakage
3. In states, enclosed in red, the PMOS transistors, continuously draw current, even if the input does not change and it has to peform no logic operatio *In red enclosed states there is conducting path between V_{DD} and ground (0 V)*

There are some more issues with NMOS and PMOS

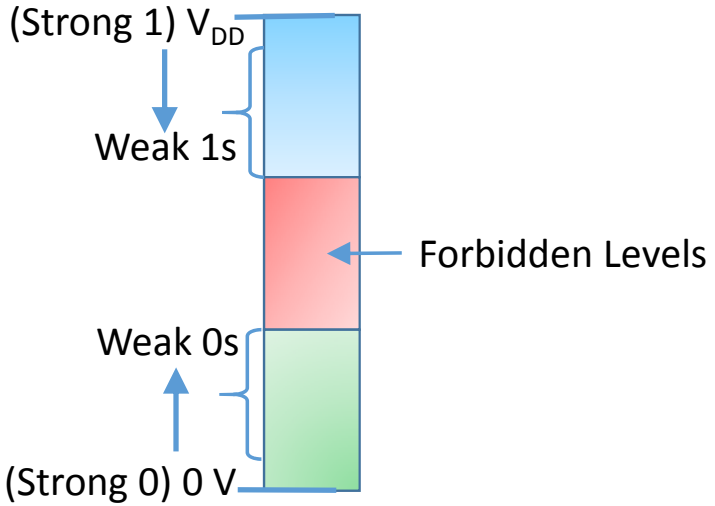


V_{in}	V_G	V_{out}
1	0	Z
0	0	Z
0	1	0
1	1	Weak 1 ($V_{DD} - V_t$)

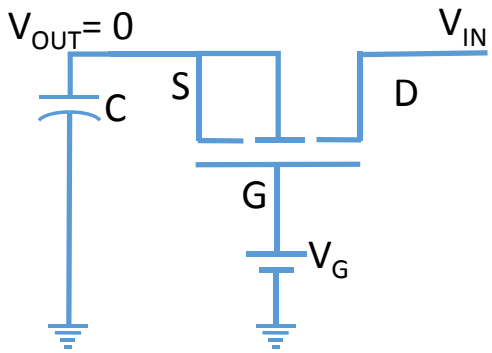


V_{in}	V_G	V_{out}
1	1	Z
0	1	Z
0	0	Weak 0 (V_t)
1	0	1

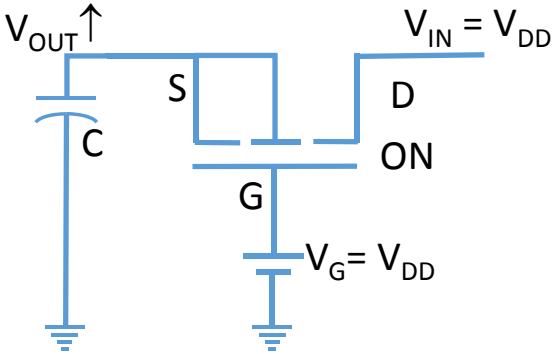
Z is a symbol used to mean that V_{out} is disconnected from V_{in} because there is a high impedance between the two terminal



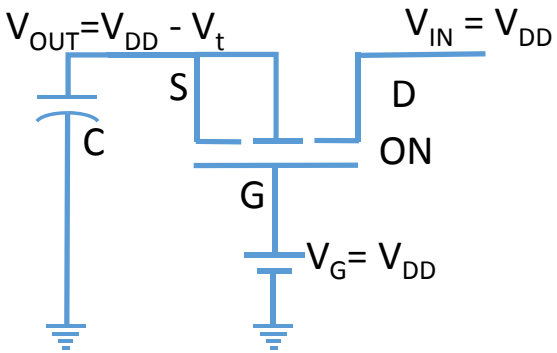
Why is NMOS bad at transmitting V_{DD} (Logic 1) ?



Assume that initially there is no charge on the capacitance C and V_{OUT} is 0V



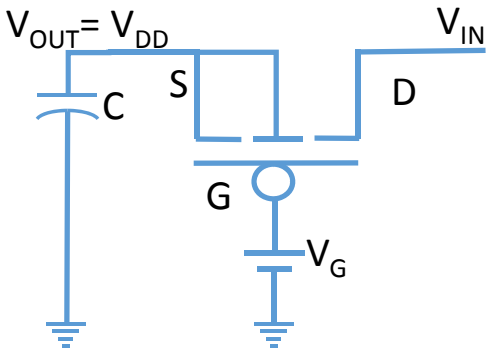
Let us drive V_{IN} and V_G to V_{DD}
Because the Gate to Body/Source voltage, V_{GS} = V_{DD}, the NMOS transistor is turned ON and current starts to flow from drain to source and charge the capacitance C and V_{OUT} starts to increase (↑)



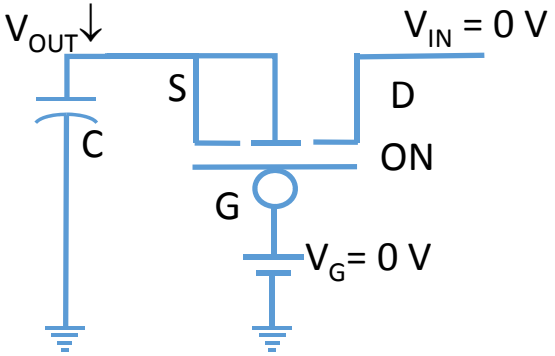
The maximum voltage that V_{OUT} can reach is $V_{DD} - V_t$
This is because for the transistor to stay ON, the voltage difference between Gate and Body/Source, V_{GS}, should be atleast V_t.
Let us say that V_{OUT} does reach V_{DD}, if it happened, the voltage difference between Gate and Source/Body would become zero, turning OFF the transistor and stop charging the capacitance

Section 3.8.7 in Brown and Vranesic Book

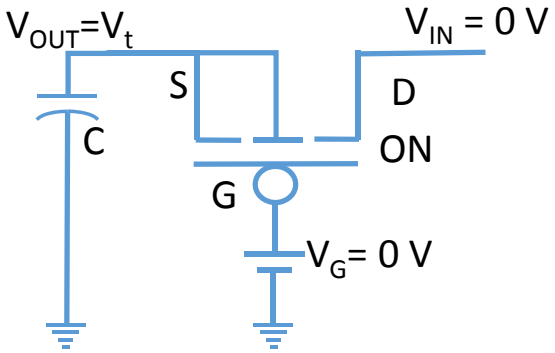
Why is PMOS bad at transmitting 0 V (Logic 0) ?



Assume that initially the capacitance is charged to V_{DD}
Since we want to transmit 0 to V_{OUT}, if we assume that it is already 0, then we will not see the effect of transmission. This is the reason, we want to set the initial condition to the opposite



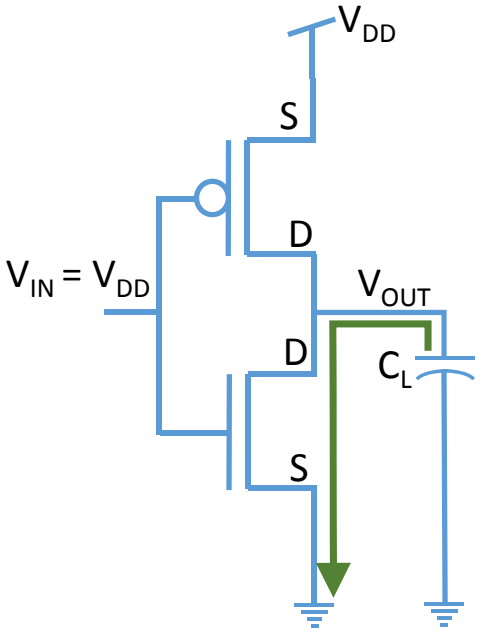
Let us drive V_{IN} and V_G to 0 V
Because the Body/Source are tied to V_{DD} (V_{OUT}) and Gate voltage has been driven to 0V, the V_{GS} is negative and fullfills the condition for PMOS to turn ON
Current starts to flow from source to drain and discharge the capacitance C and V_{OUT} starts to decrease(↓)



The minimum voltage that V_{OUT} can reach is V_t
This is because for the transistor to stay ON, the voltage difference between Gate and Body/Source, V_{GS}, should be minimum V_t.
Let us say that V_{OUT} does reach 0 V, if it happened, the voltage difference between Gate and Source/Body would become zero, turning OFF the PMOS transistor

Section 3.8.7 in Brown and Vranesic Book

By combining NMOS and PMOS we can overcome their problems



$V_{IN} = V_{DD}$

PMOS is turned OFF

The gate and body/source both are at VDD. There is no electric field to between gate and body to create a conducting channel

NMOS is turned ON

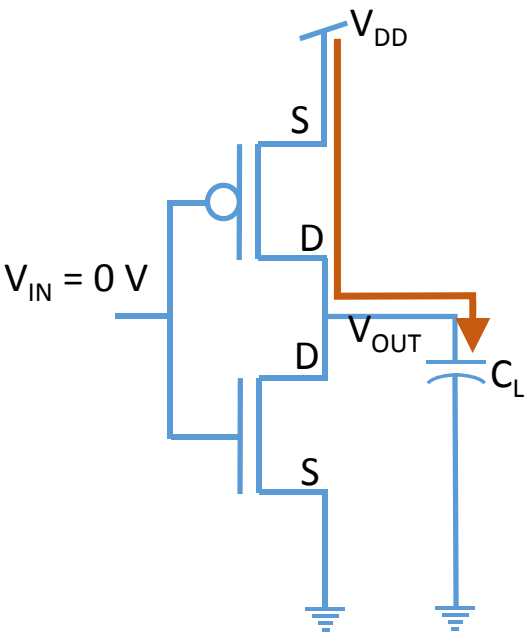
The gate is at V_{DD} and body/source is at 0V. There is strong electric field to between gate and body to create a conducting channel

What is the Voltage at V_{OUT}?

0 V

Because the N transistor is ON
*Its **S** and **D** terminals are connected by low resistance (ideally zero).*
*Since **S** is connected to GND (0 V), the charge(if any) on $\boldsymbol{C_L}$ will be discharged by the **N** transistor and $\boldsymbol{V_{OUT}}$ will become 0 V.*

By combining NMOS and PMOS we can overcome their problems



V_{IN} = 0 V

PMOS is turned ON

The gate is at 0 V and body/source is atVDD. There is a strong electric field to between gate and body to create a conducting channel

NMOS is turned OFF

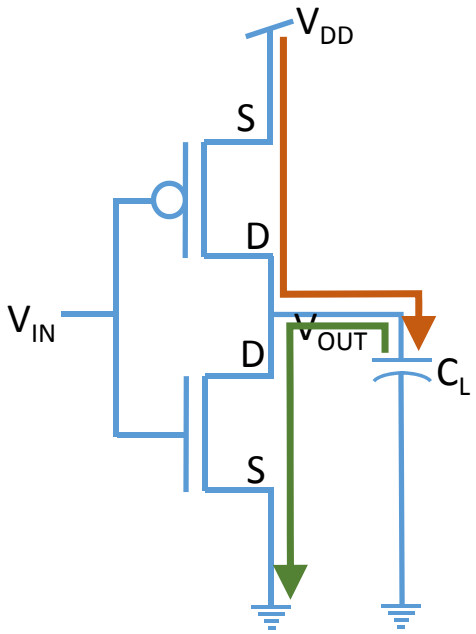
The gate and body/source both are at 0V. There is no electric field to between gate and body to create a conducting channel

What is the Voltage at V_{OUT}?

VDD

Because the P transistor is ON
*Its **S** and **D** terminals are connected by low resistance (ideally zero).*
*Since **S** is connected to V_{DD}, the V_{OUT} is connected to VDD and C_L will be charged to V_{DD}*

Does it solve any PMOS/NMOS problem ?



Transmitting 1 (V_{DD}) to the output goes via PMOS transistor

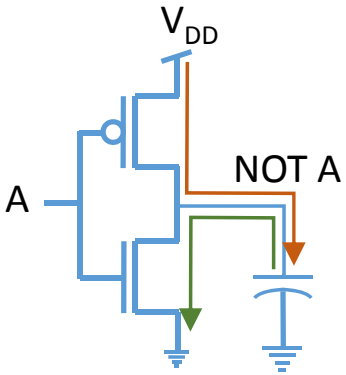
Remember PMOS is good at transmitting 1 (V_{DD})

Transmitting 0 V to the output goes via NMOS transistor

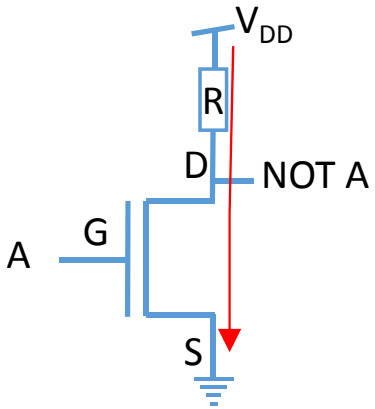
Remember NMOS is good at transmitting 0 V

This transistor structure that combines the Complementary MOS transistors is called CMOS
Almost all digital transistors are CMOS

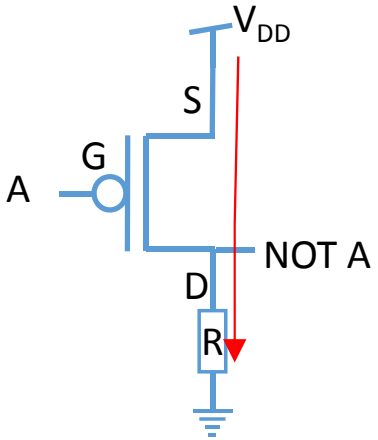
CMOS Inverter (ideally) consumes power only during switching, unlike NMOS and PMOS



A	NOT A
1	0
0	1



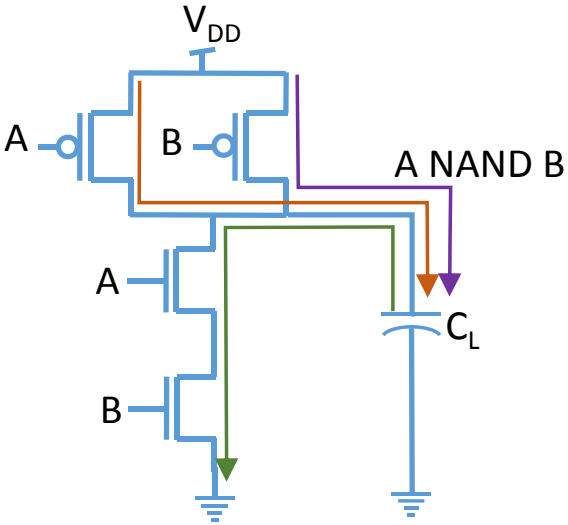
A	NOT A
0	1
1	0



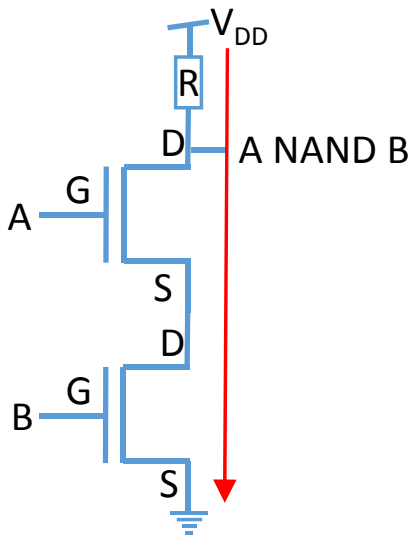
A	NOT A
1	0
0	1

In CMOS, there is no row in truth table for which there is a path between V_{DD} and GND (0 V)
This implies that power is consumed only when the inputs change – C_L is charged or discharged

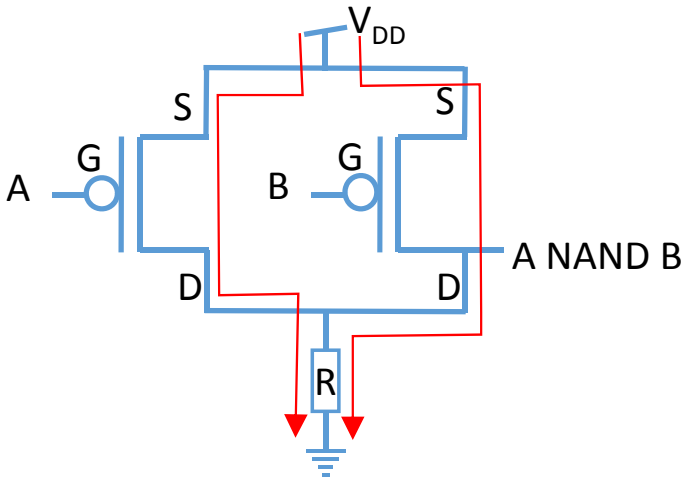
CMOS NAND (ideally) consumes power only during switching, unlike NMOS and PMOS



A	B	A NAND B
1	1	0
0	1	1
1	0	1
0	0	1



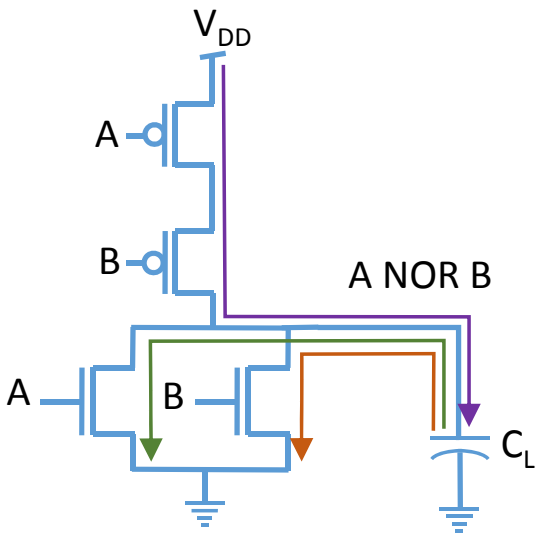
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



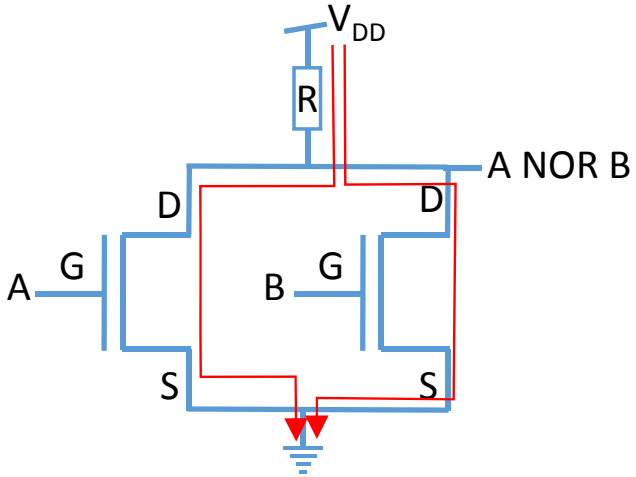
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

In CMOS, there is no row in truth table for which there is a path between V_{DD} and GND (0 V)
This implies that power is consumed only when the inputs change – C_L is charged or discharged

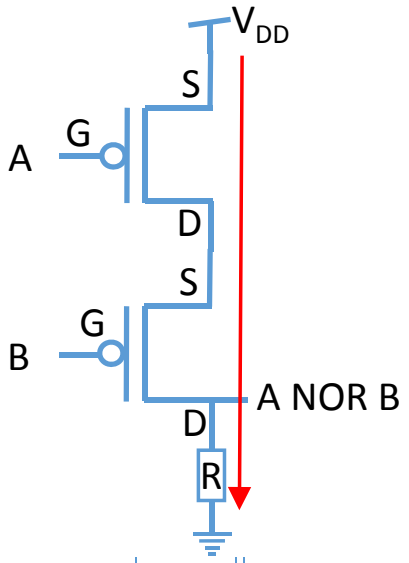
CMOS NOR (ideally) consumes power only during switching, unlike NMOS and PMOS



A	B	A NOR B
1	1	0
0	1	0
1	0	0
0	0	1



A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



A	B	A NOR B
1	1	0
0	1	0
1	0	0
0	0	1

In CMOS, there is no row in truth table for which there is a path between V_{DD} and GND (0 V)
This implies that power is consumed only when the inputs change – C_L is charged or discharged

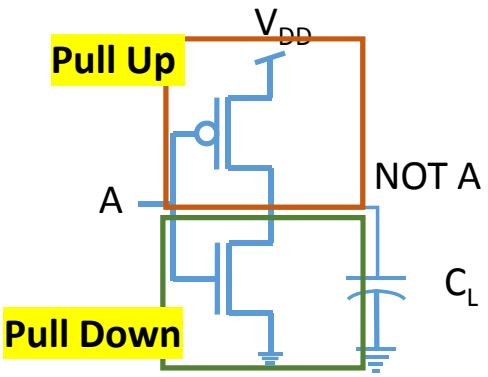
Systematic design method?

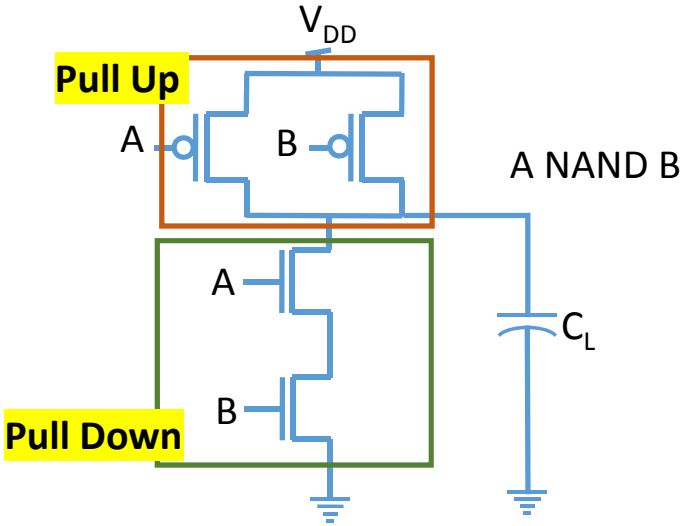
How did we find out that these particular combinations of PMOS and NMOS transistors implement NOT, NAND and NOR Gates ?

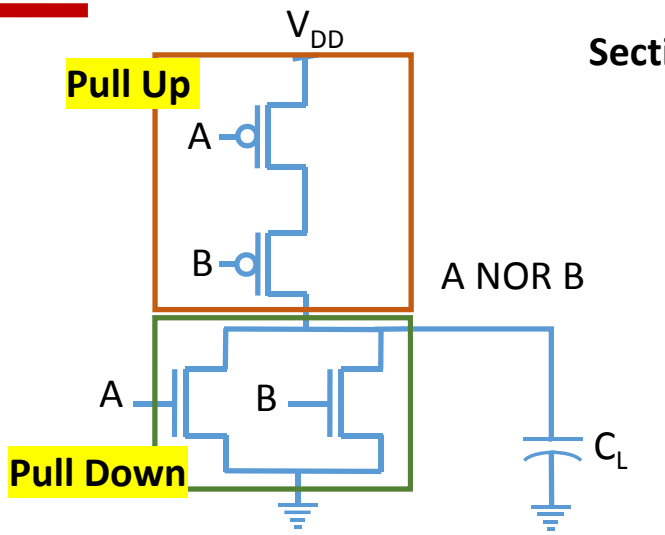
Is this trial and error or is there a systematic method ?

There is a systematic method and we will study it next,
and apply it to find systematically the right combination of PMOS and NMOS transistors for arbitrary Boolean functions, not just NOT, NAND and NOR

ON Set is implemented by PMOS Pull Up Network
OFF Set is implemented by NMOS Pull Down Network







A	NOT A	
1	0	Pull Down
0	1	Pull Up

A	B	A NAND B	
1	1	0	Pull Down, Off Set
0	1	1	Pull Up, On Set
1	0	1	
0	0	1	

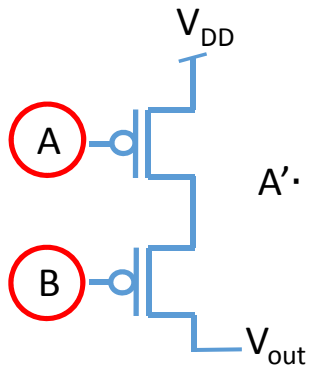
A	B	A NOR B	
1	1	0	Pull Down, Off Set
0	1	0	
1	0	0	
0	0	1	Pull Up, On Set

ON set is implemented by a network of PMOS transistors. This network is called Pull Up or Assert High Network

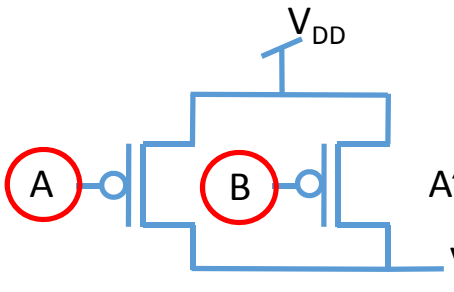
OFF set is implemented by a network of NMOS transistors. This network is called Pull Down or Assert Low Network

The Pull Up Primitives

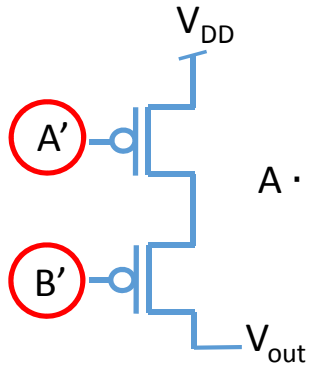
Careful: The Pull Up Primitives invert the literals when used as inputs of the PMOS transistors
Why ? Because the PMOS transistor has a buit in inversion.



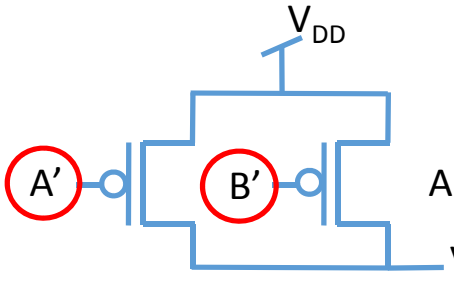
$A' \cdot B'$ → The Serial Pull Up Primitive



$A' + B'$ → The Parallel Pull Up Primitive

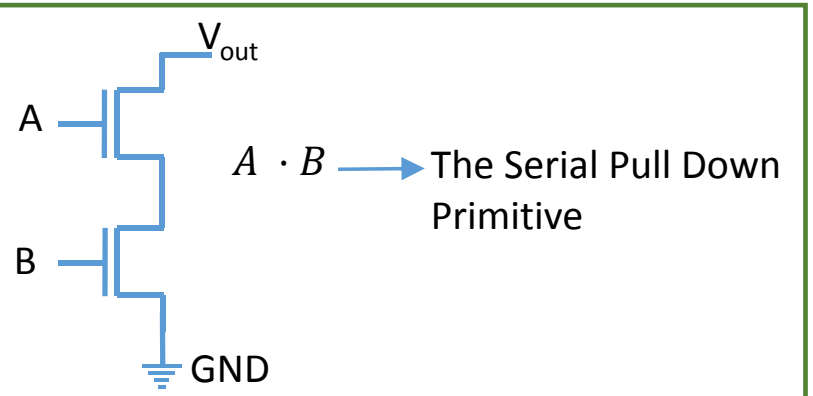


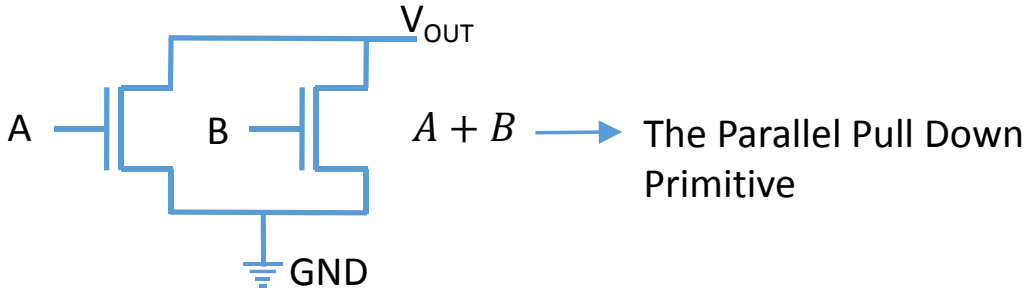
$A \cdot B$ → The Serial Pull Up Primitive

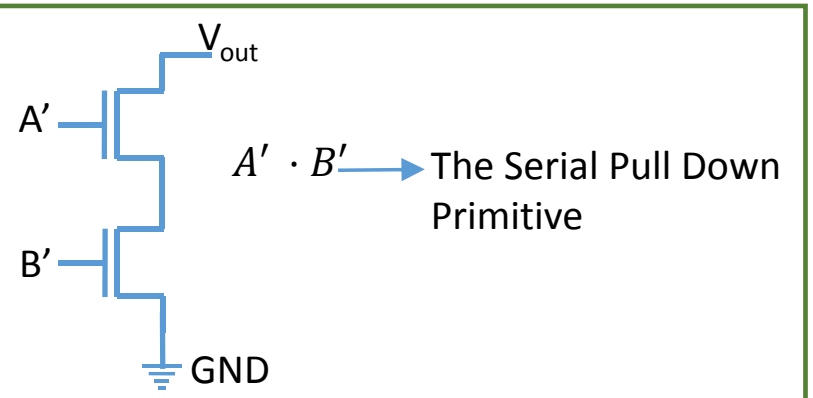


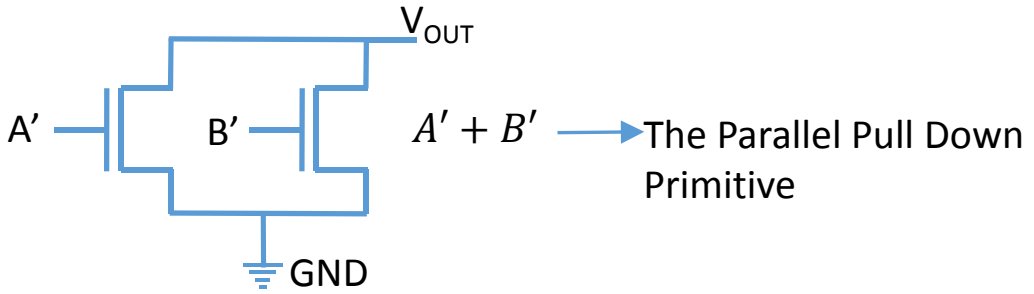
$A + B$ → The Parallel Pull Up Primitive

The Pull Down Primitives









Recipe for implementing arbitrary Boolean function in CMOS

Step 1: Simplify the function – ***f*** , so that it is composed of ONLY the serial and parallel Pull Up and Pull Down primitive $A + B$, $A \cdot B$, $A' + B'$, $A' \cdot B'$,
Note that it is not allowed to complement products (NANDs) or complement sums (NORs). Though inversion of literals is OK. This restriction is because we want the function *f* to be expressed in terms of our Pull Up and Pull Down primitives that allows only the above four forms.

Let us call this simplified function as ***fs****. Note that* ***f*** *and* ***fs*** *are functionally equivalent. They have the same truth table.*

We create two versions of ***fs****:* fs_{ON} *to implement the ON set and* fs_{OFF} *to implement the OFF set*

Step 2: fs_{ON} is the same as ***fs***. Just **remember to invert the literals**. This is done because fs_{ON} is implemented by Pull Up PMOS network. And PMOS has built in inversion of the literals.

Step 3*:* fs_{OFF} *is created by inverting the fs and again ensuring that it is in terms of the primitives. This is done because by definition* fs_{OFF} *implements the complement of* ***f*** *– the OFF set.*

NAND Gate derived by the Recipe

$$f = (A \cdot B)'$$

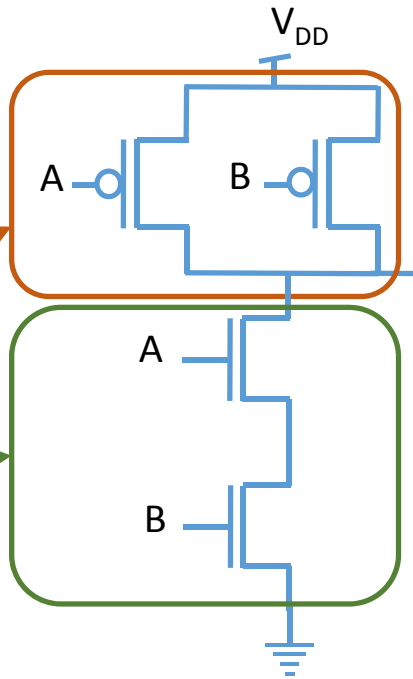
Step 1: Transform *f* to be in terms of the primitives.

$f_s = A' + B'$

Step 2: Create fs_{ON} by inverting the literal when used as inputs to the gates of PMOS Pull Up Network

Step 3: Create fs_{OFF} by inverting *f* and again simplifying it in terms of the primitives. fs_{OFF} is implemented by the Pull Down Network composed of NMOS transistors

$fs_{OFF} = f' = A \cdot B$



This Pull Up Network implements $fs_{ON} = A' + B'$

A NAND B

This Pull Down Network implements $fs_{OFF} = A \cdot B$

NOR Gate derived by the Recipe

$$f = (A + B)'$$

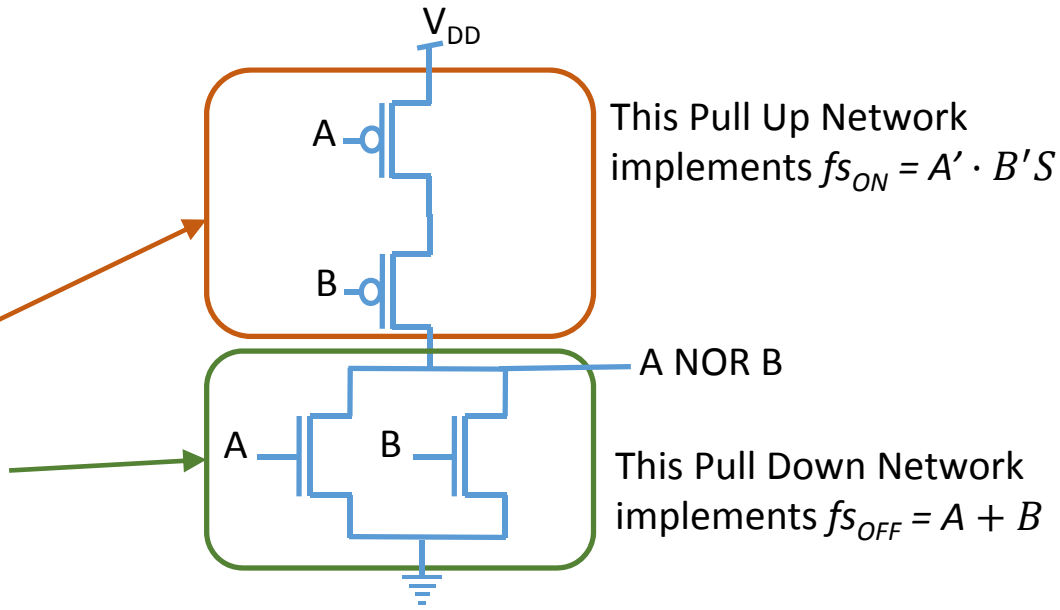
Step 1: Transform *f* to be in terms of the primitives.

$$f_s = A' \cdot B'$$

Step 2: Create fs_{ON} by inverting the literal when used as inputs to the gates of PMOS Pull Up Network

Step 3: Create fs_{OFF} by inverting *f* and again simplifying it in terms of the primitives.

$fs_{OFF} = f' = A + B$



Recipe applied to arbitrary boolean function

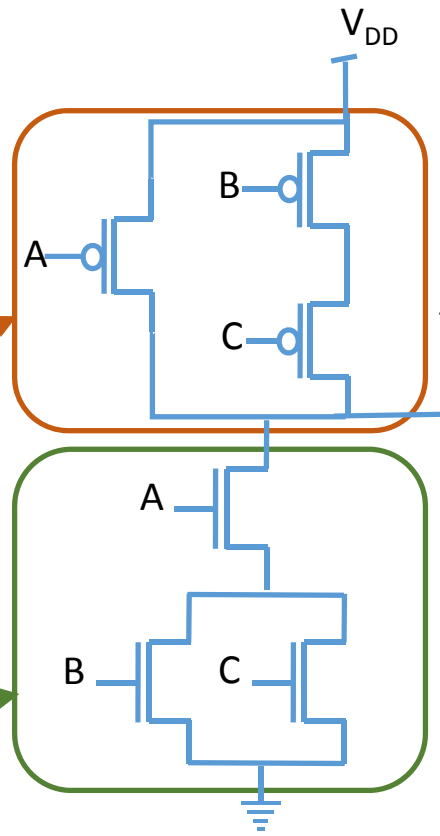
$f = [A \cdot (B + C)]'$

Step 1: Transform *f* to be in terms of the primitives.

$f_s = A' + (B' \cdot C')$

Step 2: Create fs_{ON} by inverting the literal when used as inputs to the gates of PMOS Pull Up Network
Notice the *fs* has two primitives. One is a serial primitive *B·C* and the other is a parallel primitive combining *B·C* and *A*

Step 3: Create fs_{OFF} by inverting *f* and again simplifying it in terms of the Pull Down NMOS primitives.
$fs_{OFF} = f = A \cdot (B + C)$



This Pull Up Network implements $fs_{ON} = A' + (B' \cdot C')$

A NOR B

This Pull Down Network implements $fs_{OFF} = A \cdot (B + C)$

Recipe applied to XOR Gate

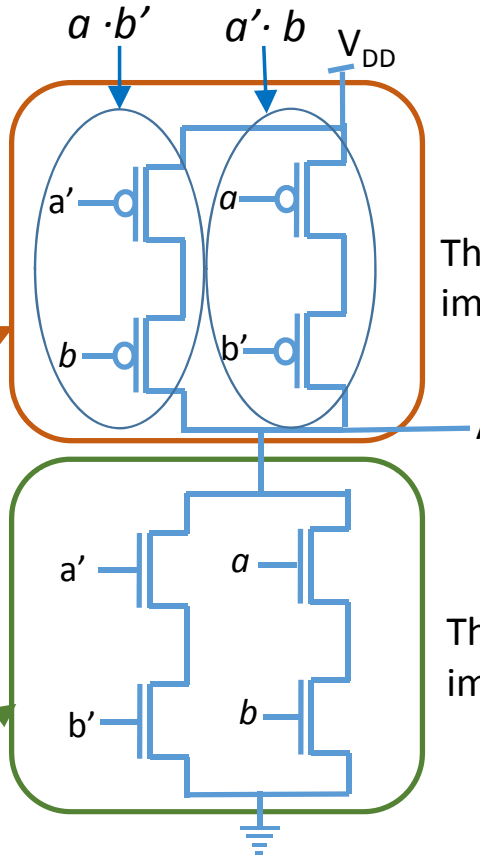
$$f = a' \cdot b + a \cdot b'$$

Step 1: Transform *f* to be in terms of the primitives. The XOR gate function is already in form of primitives
$f_s = f$

Step 2: Create fs_{ON} by inverting the literal when used as inputs to the gates of PMOS Pull Up Network.

Notice the *fs* has three primitives. Two are serial primitives a'·*b* and *a*·*b'* the third is a parallel primitive combining the two serial primitives.

Step 3: Create fs_{OFF} by inverting *f* and again simplifying it in terms of the Pull Down NMOS primitives.
$fs_{OFF} = f' = a' \cdot b' + a \cdot b$



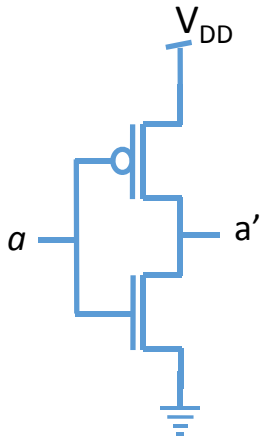
This Pull Up Network implements $fs_{ON} = a' \cdot b + a \cdot b'$

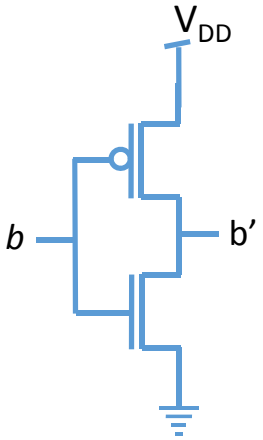
A XOR B

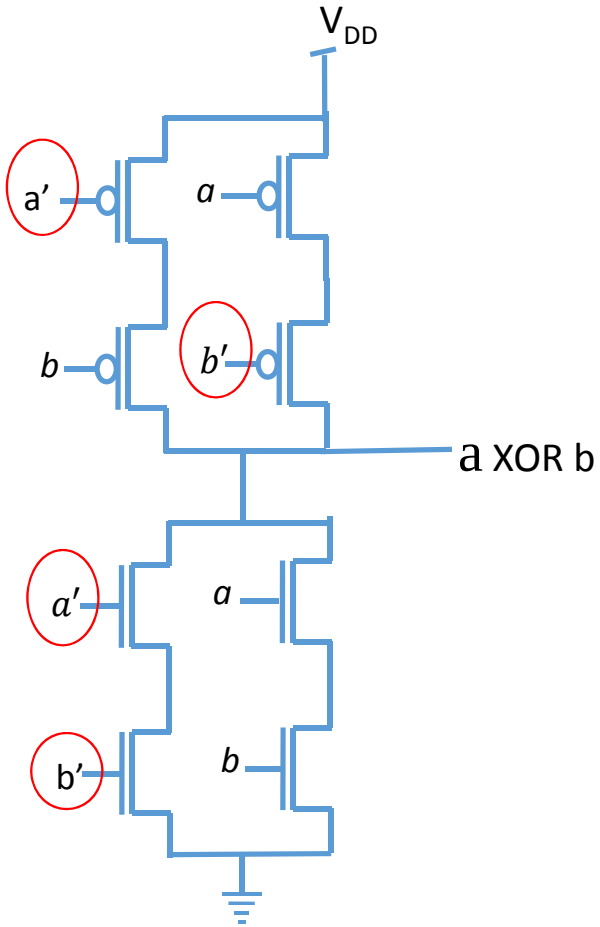
This Pull Down Network implements $fs_{OFF} = a' \cdot b' + a \cdot b$

Who provides the inverted literals ?

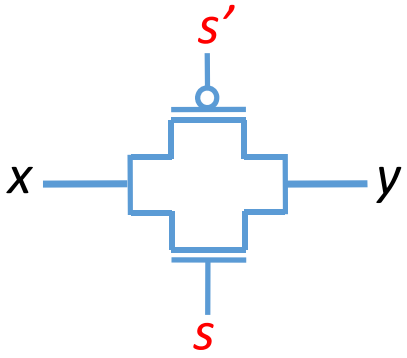
12 Transistors Needed



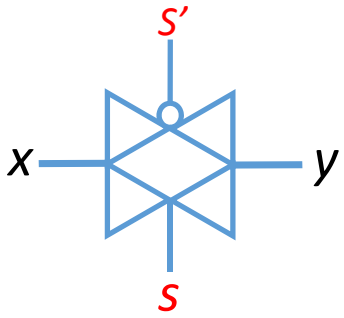




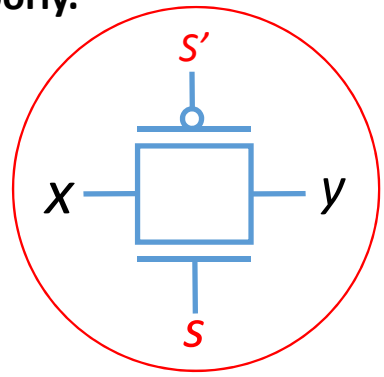
CMOS Transmission Gates (TGs)



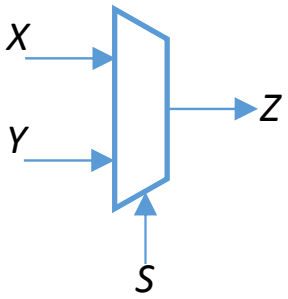
1. **Transmission Gates are also constructed from PMOS and NMOS transistors**
2. **They act as a switch whose operation can be controlled by a signal *s***
3. **When *s = 1*, both the NMOS and PMOS transistors are ON.**
   ***y = x*, i.e., *y* is connected to *x* or the resistance between *y* and *x* is negligibly low.**
   We say *x* is transmitted to *y* or *y* is driven by *x*
4. **When *s = 0*, *y* is disconnected from *x*, i.e., there is a high impedance between *y* and *x*.**
   The voltage of *y* has no corelation to the voltage at *x*.
5. **Why do we have two complementary transistors (NMOS and PMOS) ?**
   Because NMOS is good at transmitting 0 (0 V/GND) and PMOS is transmitting 1(V_{DD}). If we had only one transistor PMOS or NMOS, it would still function as a switch but it will transmit 0 poorly or 1 poorly.



Transmission Gate Symbols



Multiplexor

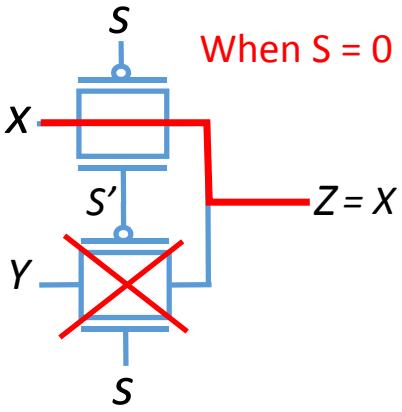


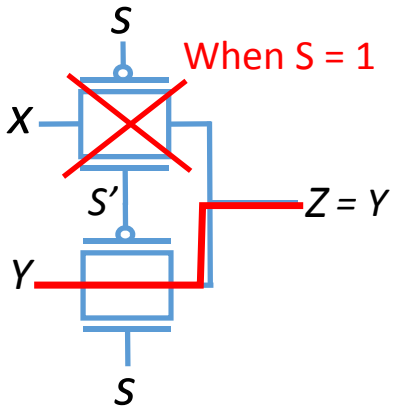
1. It is a universal building block like NAND or NOR gate because any arbitrary boolean function can be built using Muxes. We will study this in a later lecture on combinatorial circuits
2. Multiplexor are often abbreviated to Mux, plural form - Muxes

X	***Y***	***S***	***Z***
x	-	0	*x*
-	*y*	1	*y*

```
Z  <= X when S = 0 ELSE
      Y;
```

Mux Using Transmission Gate





X	Y	S	Z	
0	-	0	0	The upper TG is ON and passes X to Z
1	-	0	1	The lower TG is OFF and blocks Y
-	0	1	0	The lower TG is ON and passes Y to Z
-	1	1	1	The upper TG is OFF and blocks X

'-' symbol means don't care.
The output *Z* is unaffected by what ever value of *X* or *Y* in the rows where their value is shown as '-'

Tri State

Section 3.8 and Section 3.9

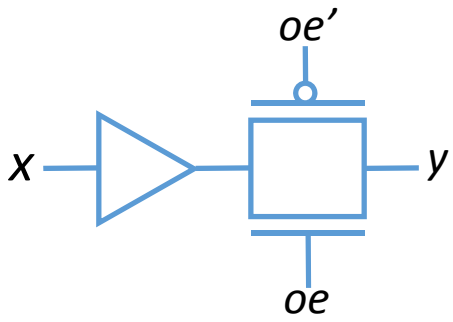
1. All nodes (input and output) of digital logic occupy, ideally logic 1 or logic 0 state.
2. These states imply that the node in logic is driven by V_{DD} or connected to GND (logic 0)
3. However, in some cases, a node in the logic may be not connected to either V_{DD} or GND.
4. This may sound as an anomaly (abnormality), however in reality this is desirable and is very useful.
5. This *third state is often called the high impedance state* because there is a high impedance between the node and the VDD / GND
6. This third state is represented by *Z*
7. Logic gates that can intentionally drive its output to the *Z* state are called tri-stated

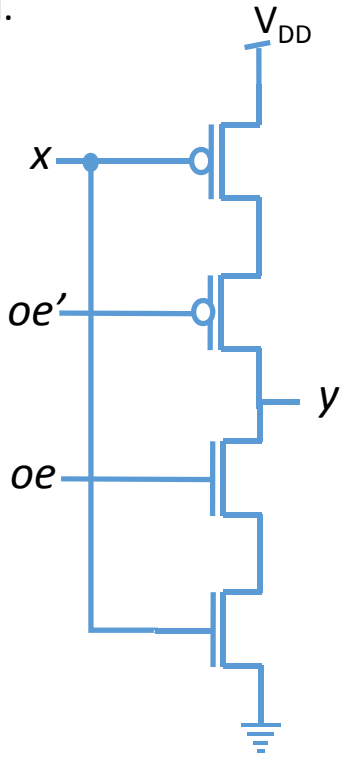
oe: output enable

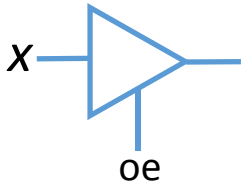
x	oe	y
x	1	x
-	0	Z

y = x when oe = 1

y = Z when oe = 0; y is disconnected from x







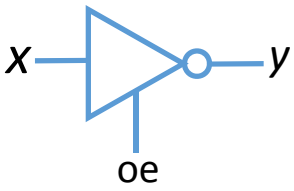
y = x when oe = 1
else Z

Tri State Variants

Section 3.8 and Section 3.9

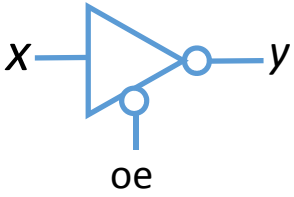
oe: output enable

x	oe	y
x	1	$\overline{x}$
-	0	Z

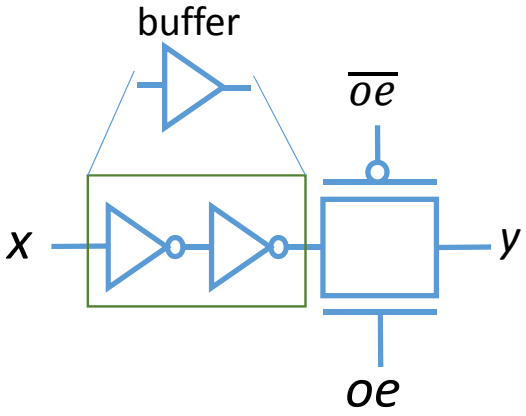


$y = \overline{x}$ *when oe = 1*
else Z

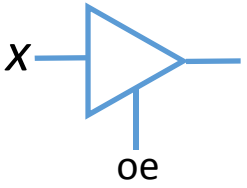
x	oe	y
x	0	$\overline{x}$
-	1	Z



$y = \overline{x}$ *when oe = 0*
else Z

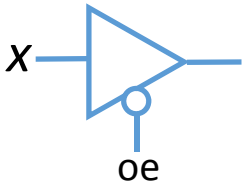


x	oe	y
x	1	x
-	0	Z



y = x when oe = 1
else Z

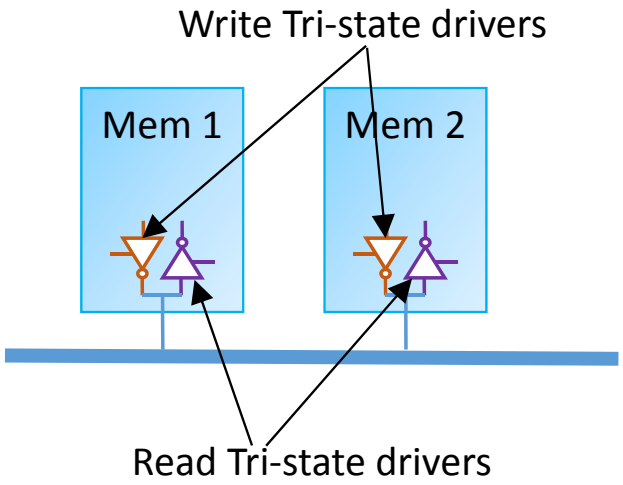
x	oe	y
x	0	x
-	1	Z



y = x when oe = 0
else Z

What are tri state buffers/drivers used for ?

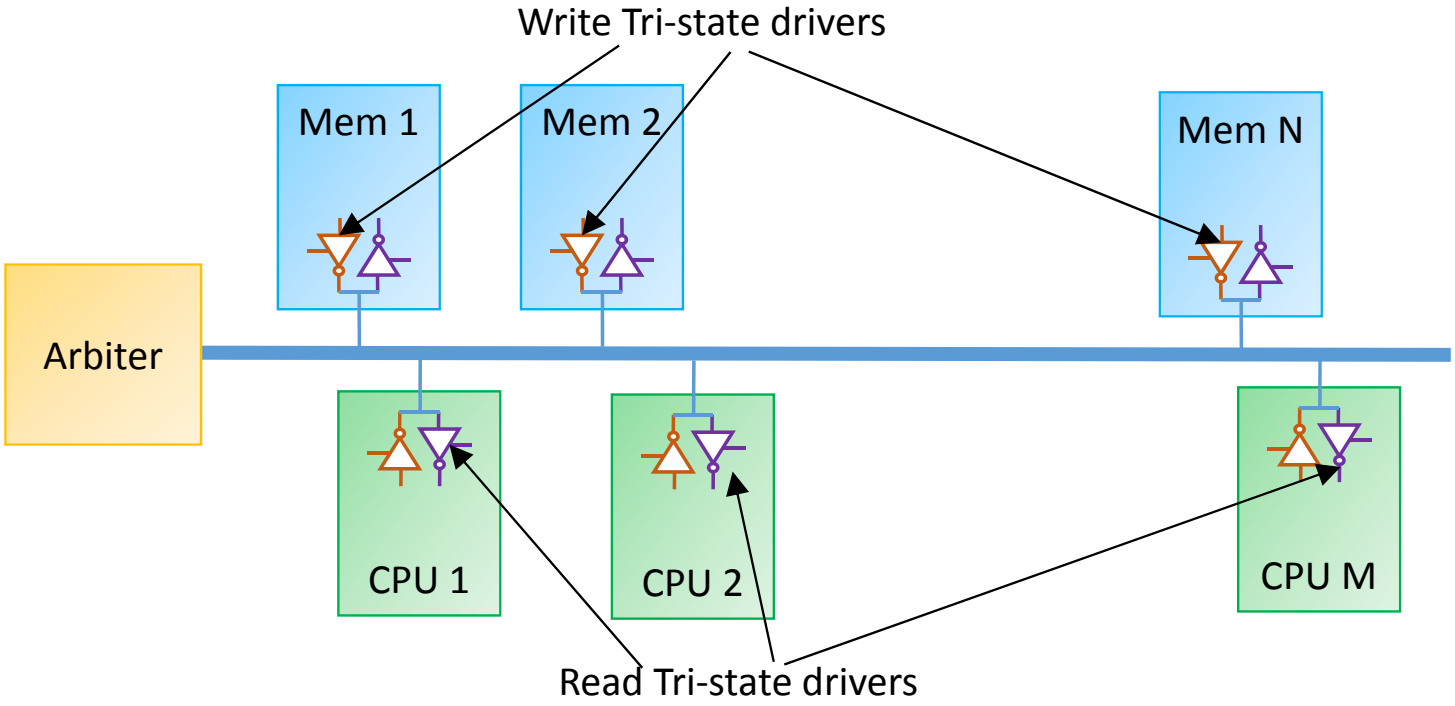
Tri States are used when we need bi-directional connections
When a device or a sub-system what to read and write from the same port



1. Let us say two Memory Devices share a bus (a bundle of wires)
2. Both memory devices can read from the bus at the same time
3. Both memory devices cannot however write to the bus at the same time
   One could be wanting to write logic 1 and ther other could be wanting write logic 0. This would create a short circuit between VDD and GND
4. Tri-state buffers / drivers help because we can disable one of the driver, while the other is writing
5. Usually there is a device called arbiter that grants access to the bus before it can write to it.
6. If all devices can read, why do we need ***read*** tri-state drivers ?
7. This is because if Mem 1 device is writing to the bus the read tri-state driver in Mem 1 is disabled. What Mem 1 is writing is meant for external devices.

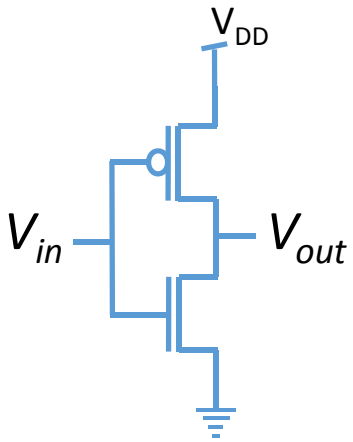
What are tri states used for ?

Tri States are used when we need bi-directional connections
When a device or a sub-system what to read and write from the same port



Only one device is allowed to drive, i.e., write to the bus. The output enable of only one output tri-state driver is activated at a time. All others are inactivate
Multiple devices can read.

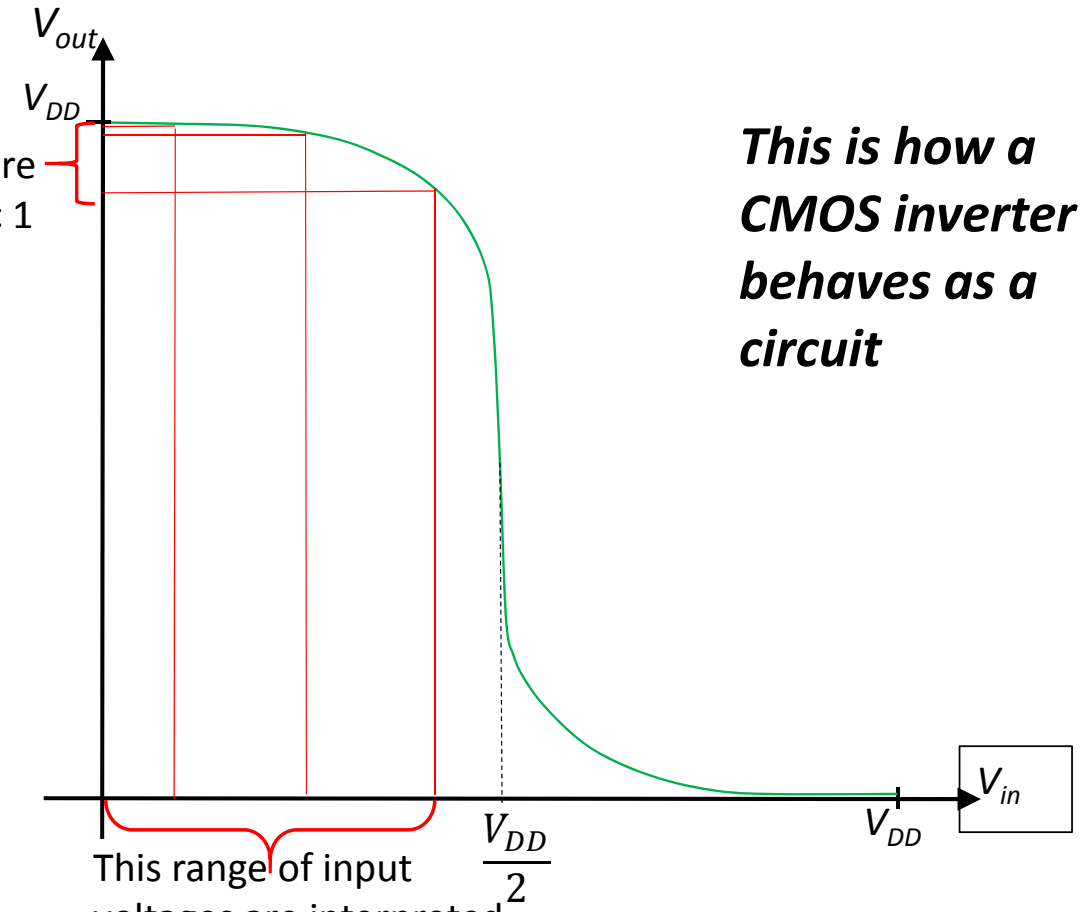
Logic 0 is a range of values at circuit level



This is the Boolean Algebraic View of an inverter

V_{in}	V_{out}
0	1
1	0

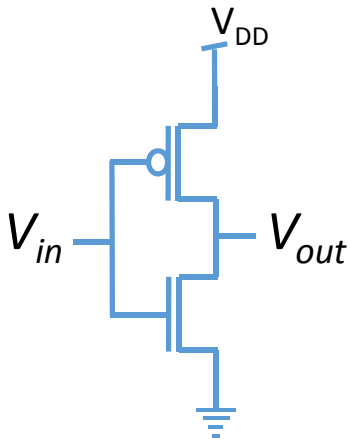
This range of output voltages are produced as logic 1



This range of input voltages are interpreted as logic 0

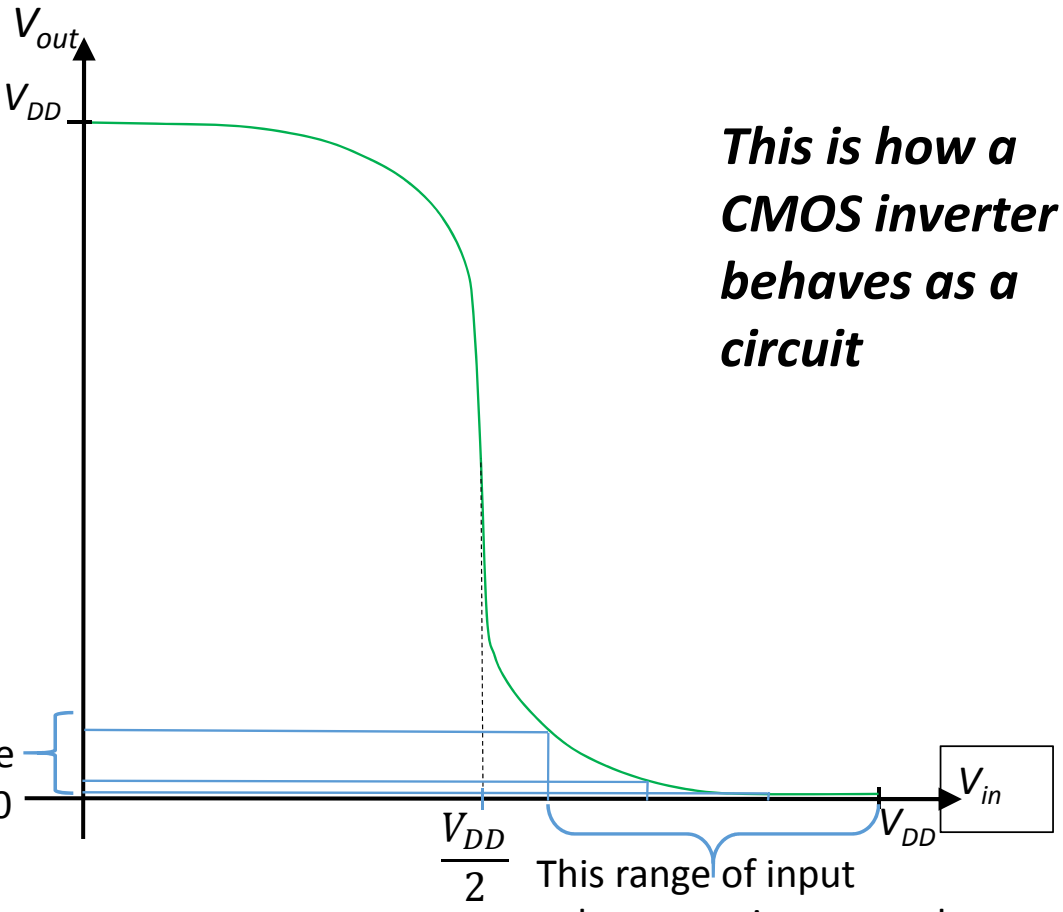
This is how a CMOS inverter behaves as a circuit

Logic 1 is also a range of values at circuit level



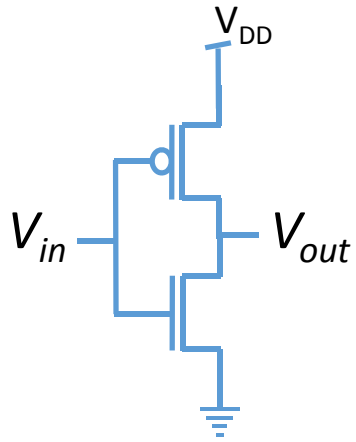
This is the Boolean Algebraic View of an inverter

V_{in}	V_{out}
0	1
1	0



This is how a CMOS inverter behaves as a circuit

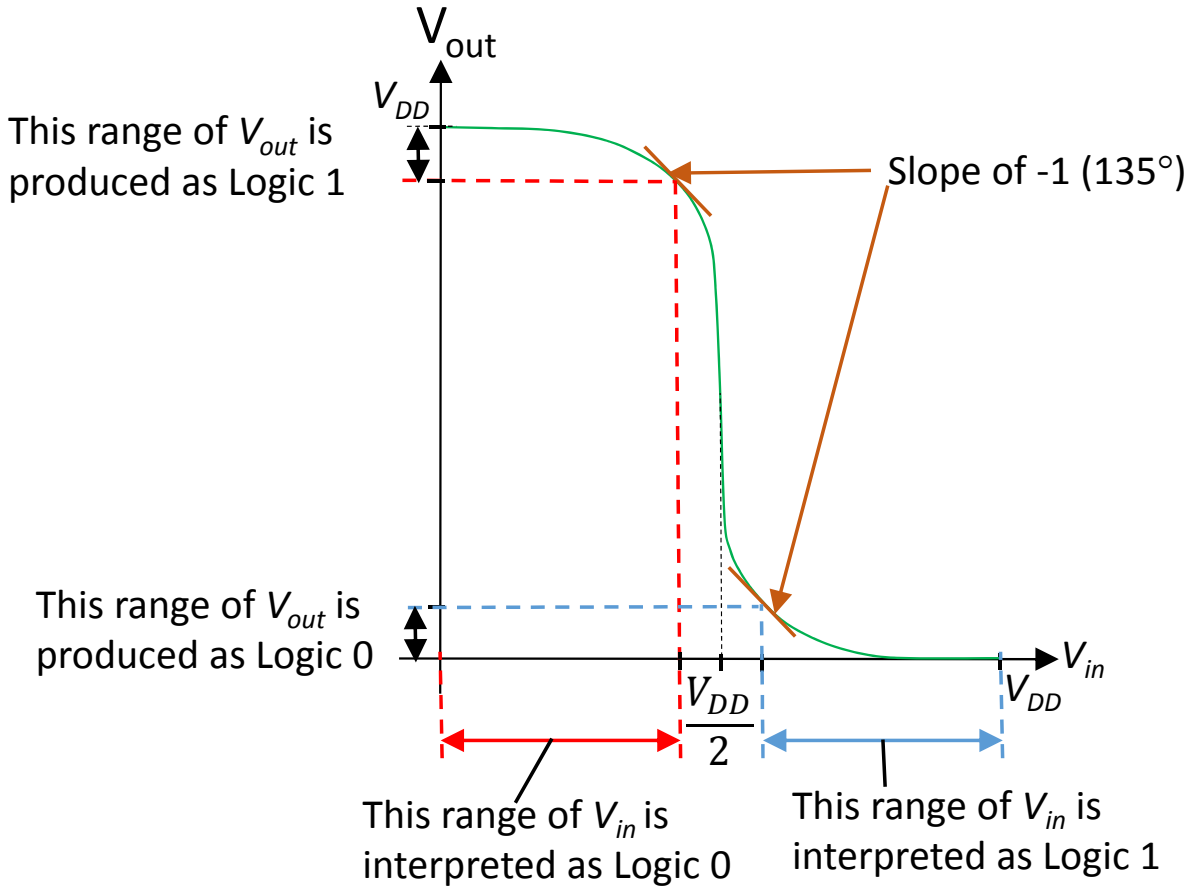
Defining Logic 0 and Logic 1 Ranges



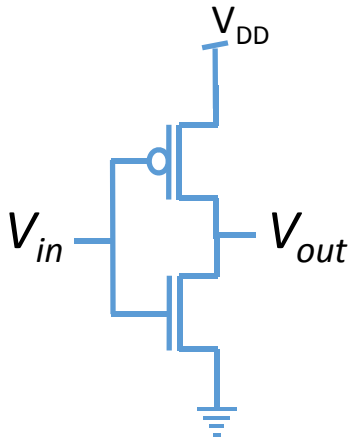
This is the Boolean Algebraic View of an inverter

V_{in}	V_{out}
0	1
1	0

This is the circuit level view of the inverter



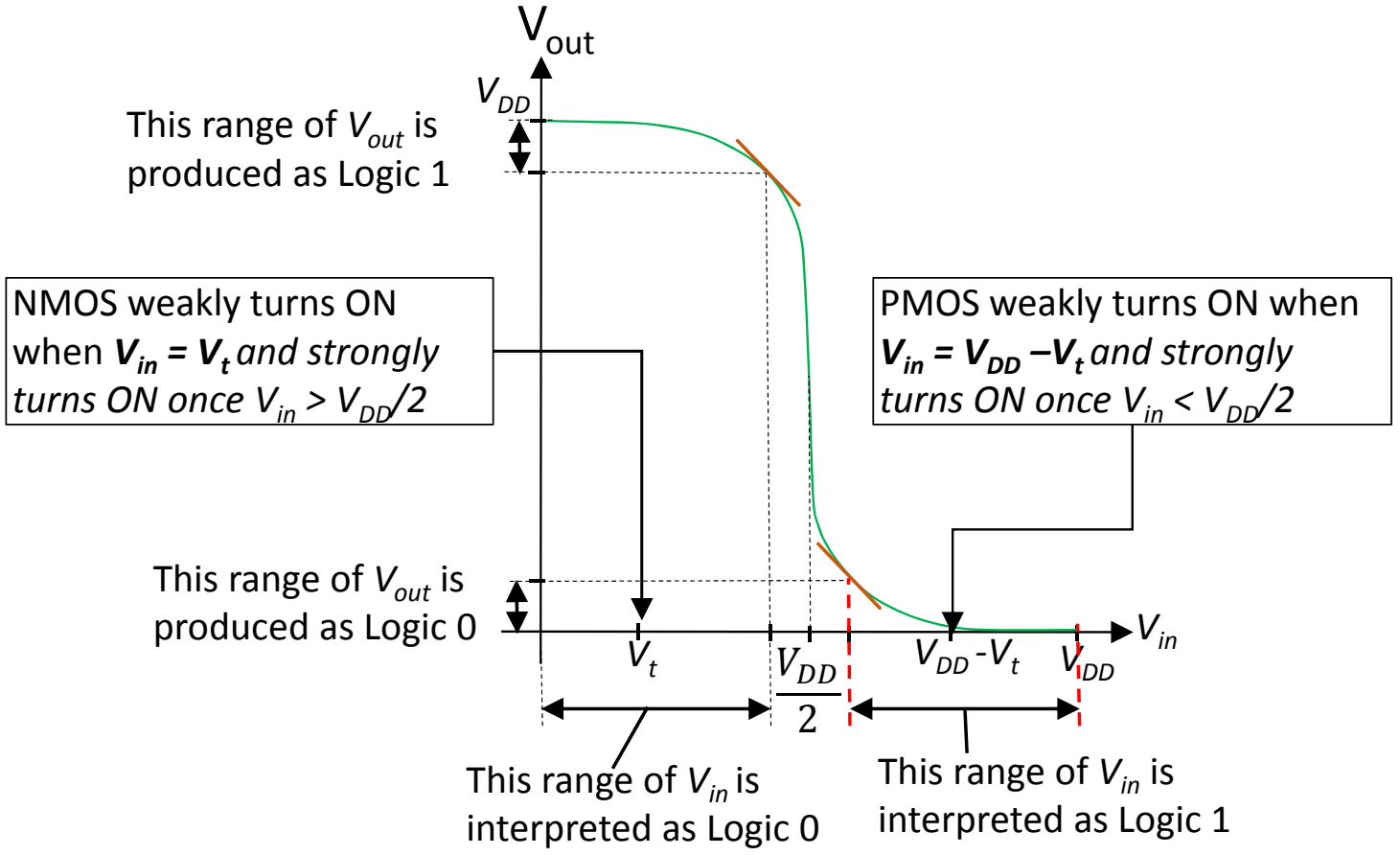
The Threshold Voltages



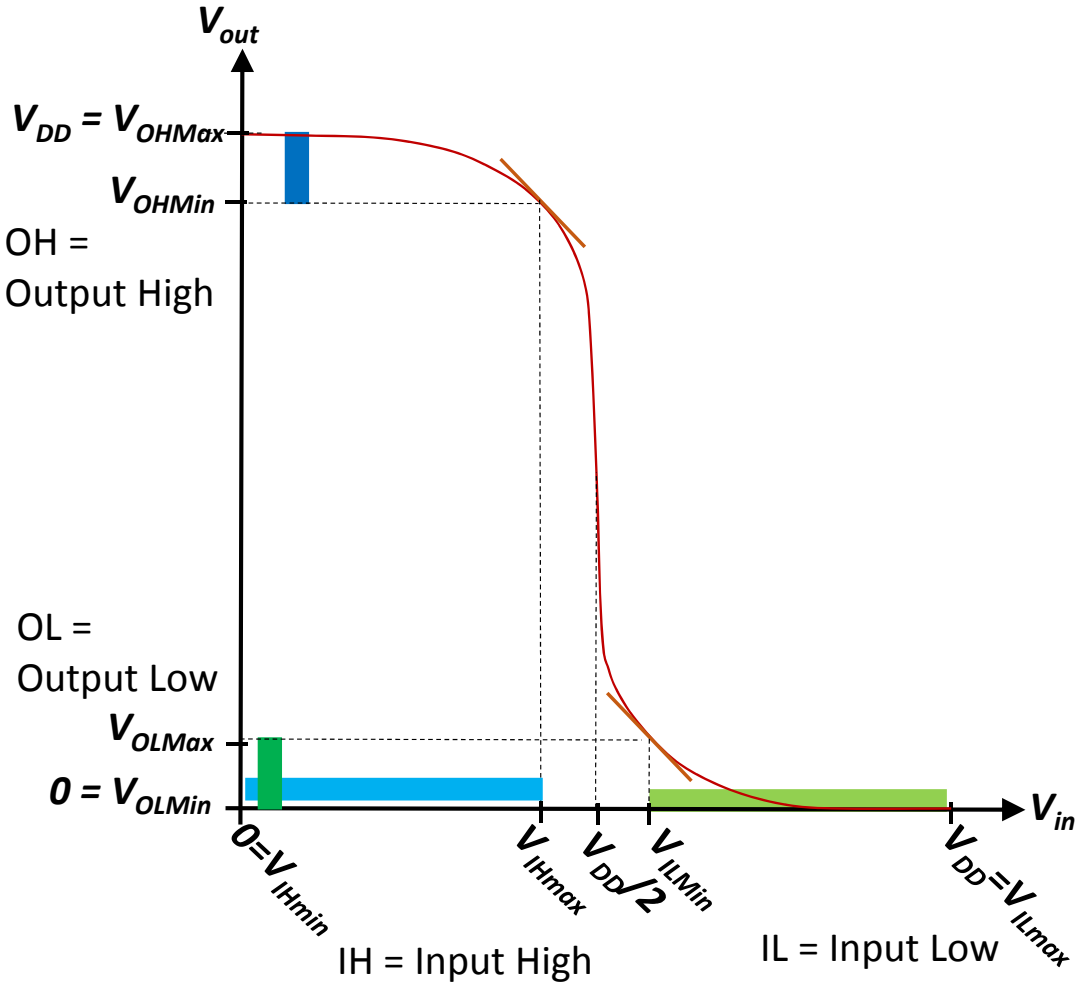
This is the Boolean Algebraic View of an inverter

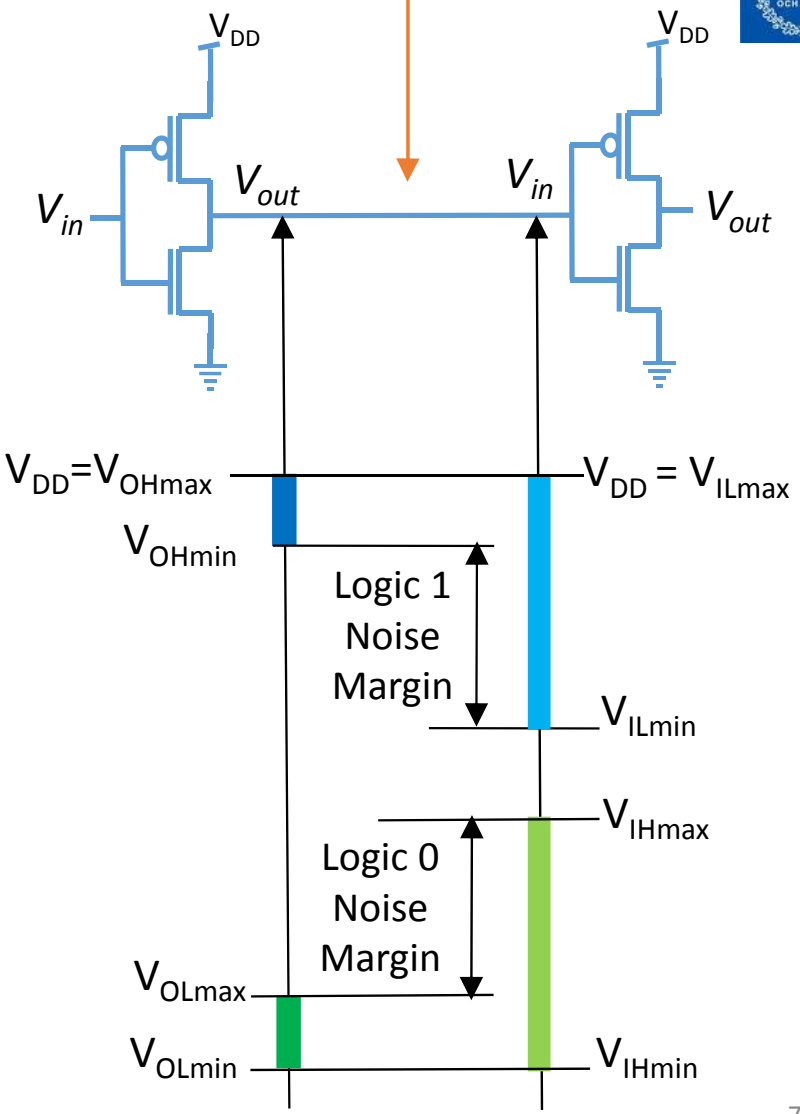
V_{in}	V_{out}
0	1
1	0

This is the circuit level view of the inverter

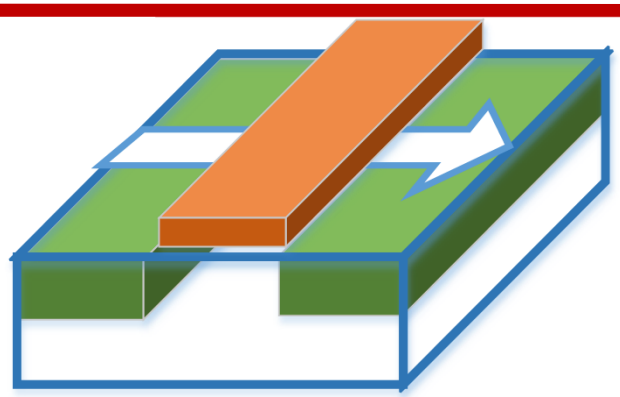


Noise Margins allow digital circuits to be robust. They allow correct interpretion 1s and 0s even if the output (V_{out}) is distorted by noise before it reaches the next inputs (V_{in})

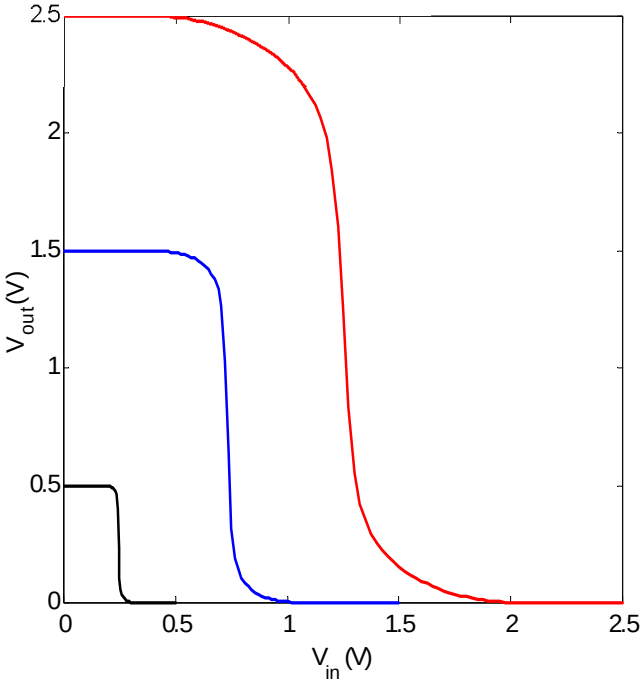




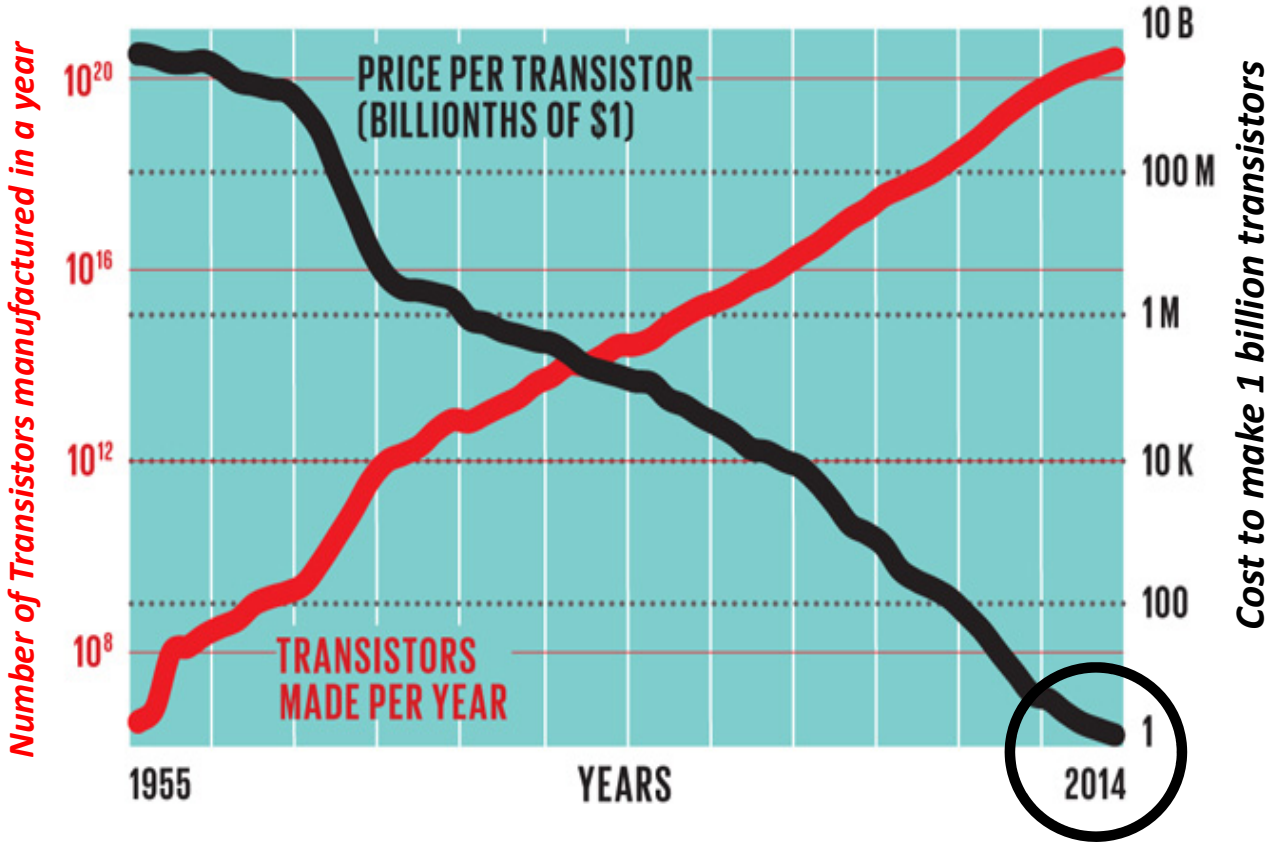
The Scaling of Transistors



1. The Scaling down of transistor geometries is done by progress in lithography
2. Smaller transistors mean
   a. Greater transistor density
   b. Faster transistors
   c. More power efficient transistors
3. Many complications also arise as a result of shrinking geometry, i.e., scaling down the size of the transistors



What has fueled the digital electronics revolution ?

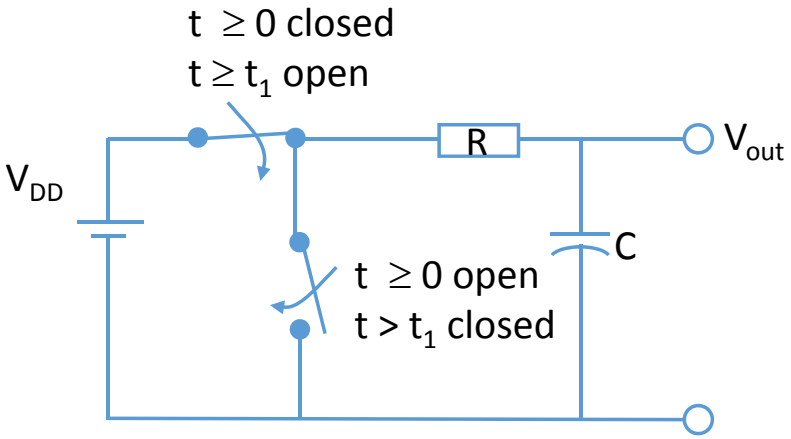


In 2014

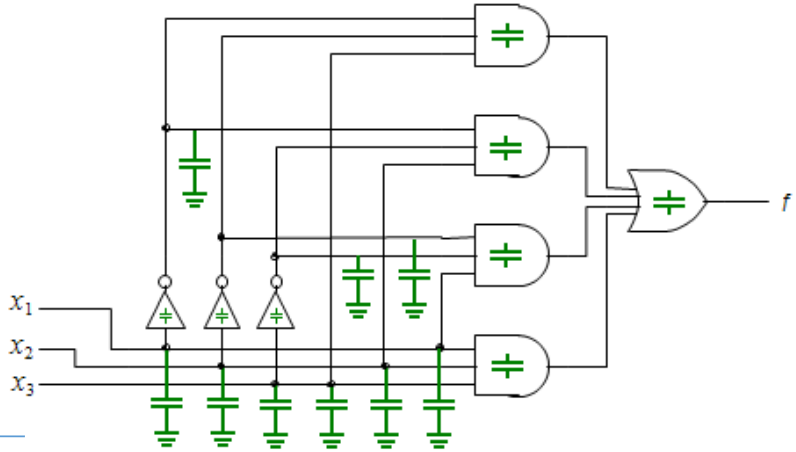
250 billion billion transistors were manufactured
8 trillion transistors/s were manufactured in 2014
25 times the number of stars in the Milky Way and
75 times the number of galaxies in the known universe

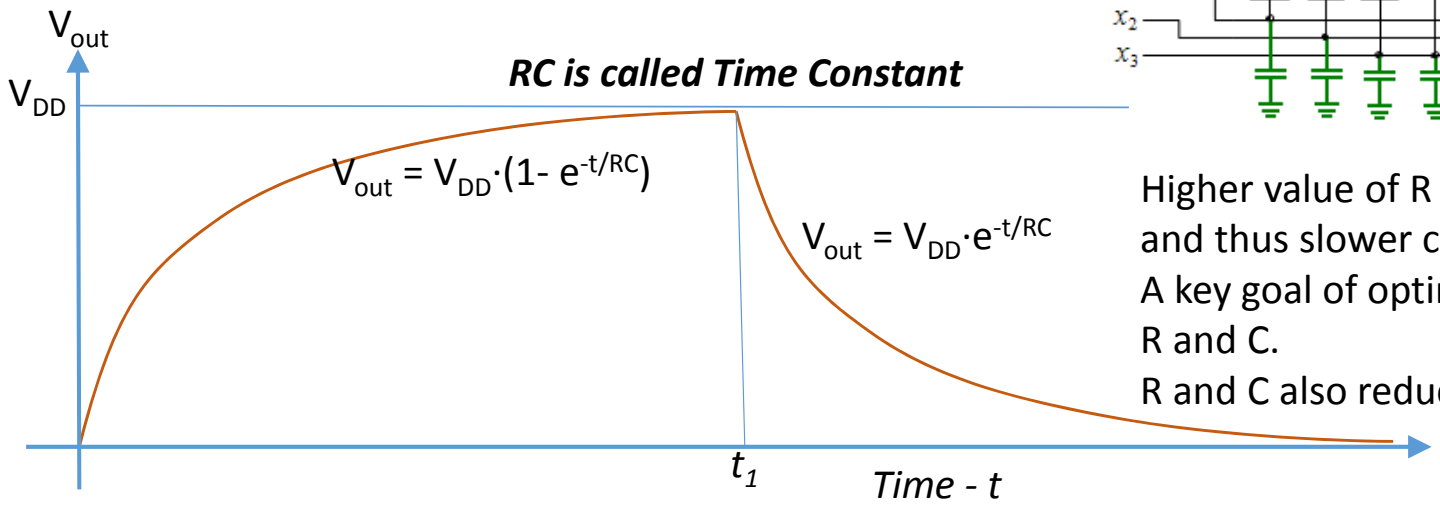
1. The robust noise margin of digital circuits
2. Technology progress enable smaller, faster, lower power consuming transistors
3. Progress in design automation

Delay in CMOS Circuits



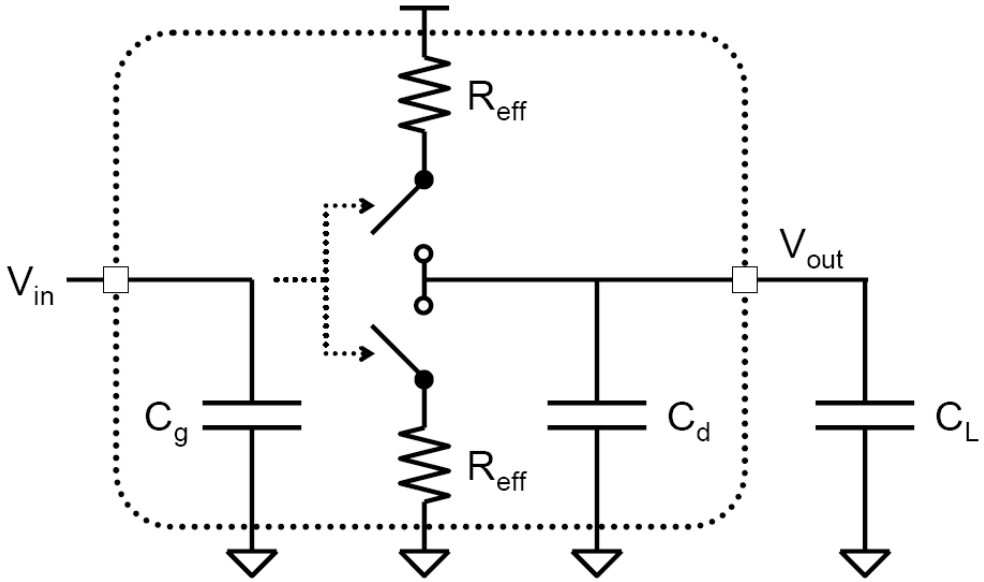
All logic elements have *R* and *C* associated with them
This results in a finite time, it takes to charge upto VDD and finite time discharge to GND

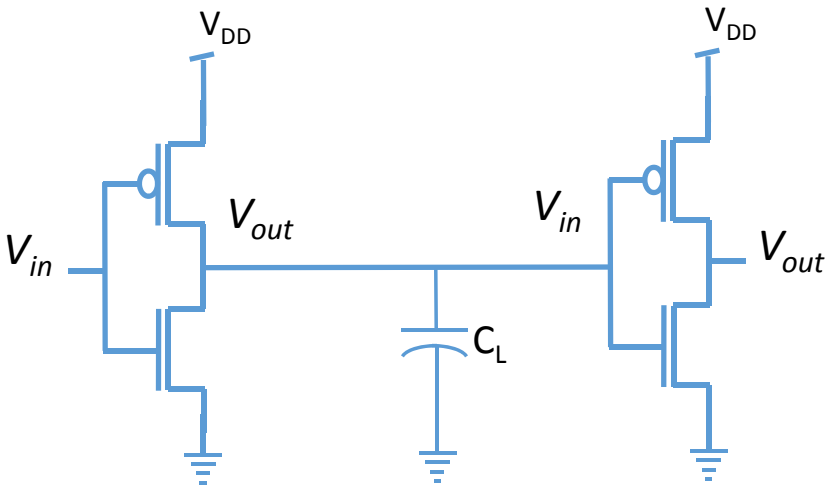




Higher value of R and C implies larger time constant and thus slower charging and discharging
A key goal of optimizing logic design is to reduce the R and C.
R and C also reduce with technology scaling

A simplified RC model of an inverter



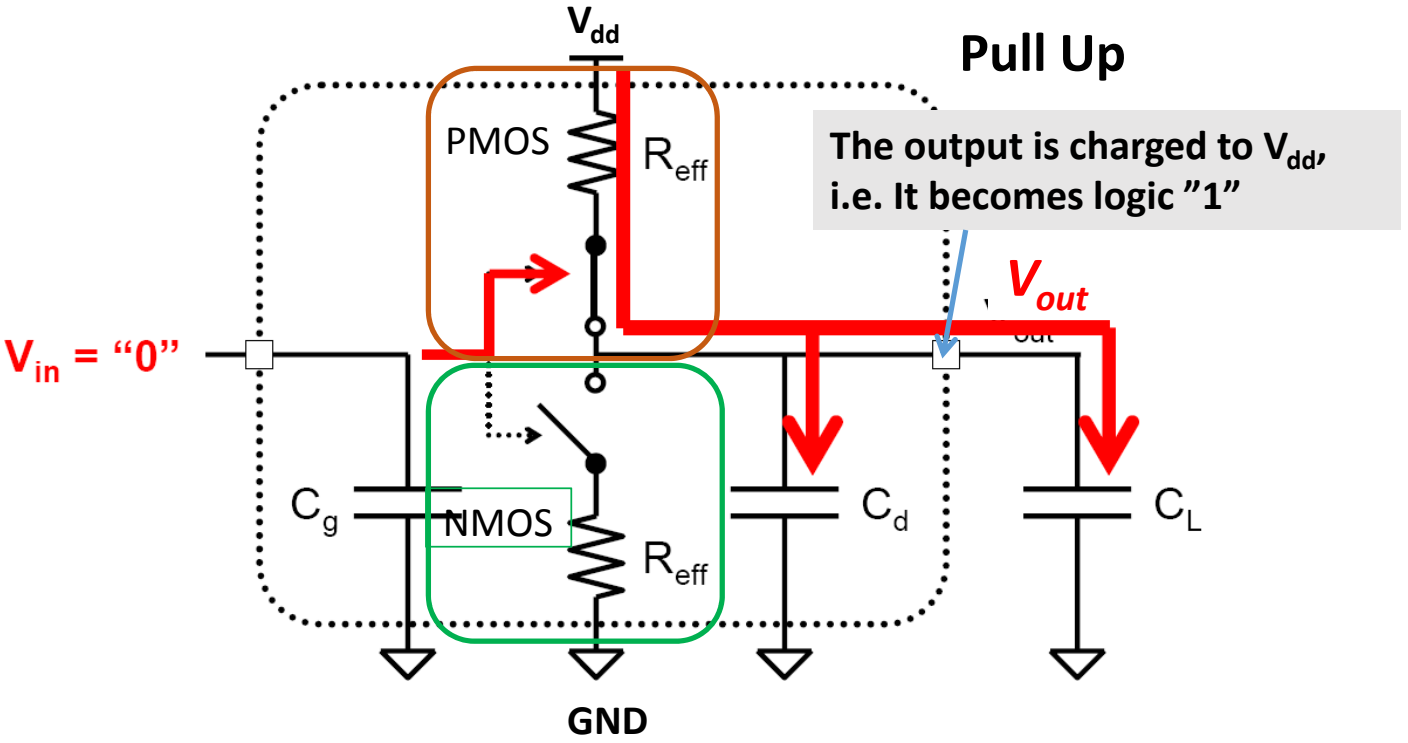


C_L comes from the wire capacitance and the gate oxide capacitance of the next gate(s) inputs.

Source: MIT. Course 6.375. Lecture L06. 2006

It takes time to charge the output (V_{out}) upto VDD

1. The R_{eff} of P transistor is more than the R_{eff} of N transistor because of lower mobility of holes that are the charge carriers in P transistors
2. The charging up to V_{DD} happens via the P transistor

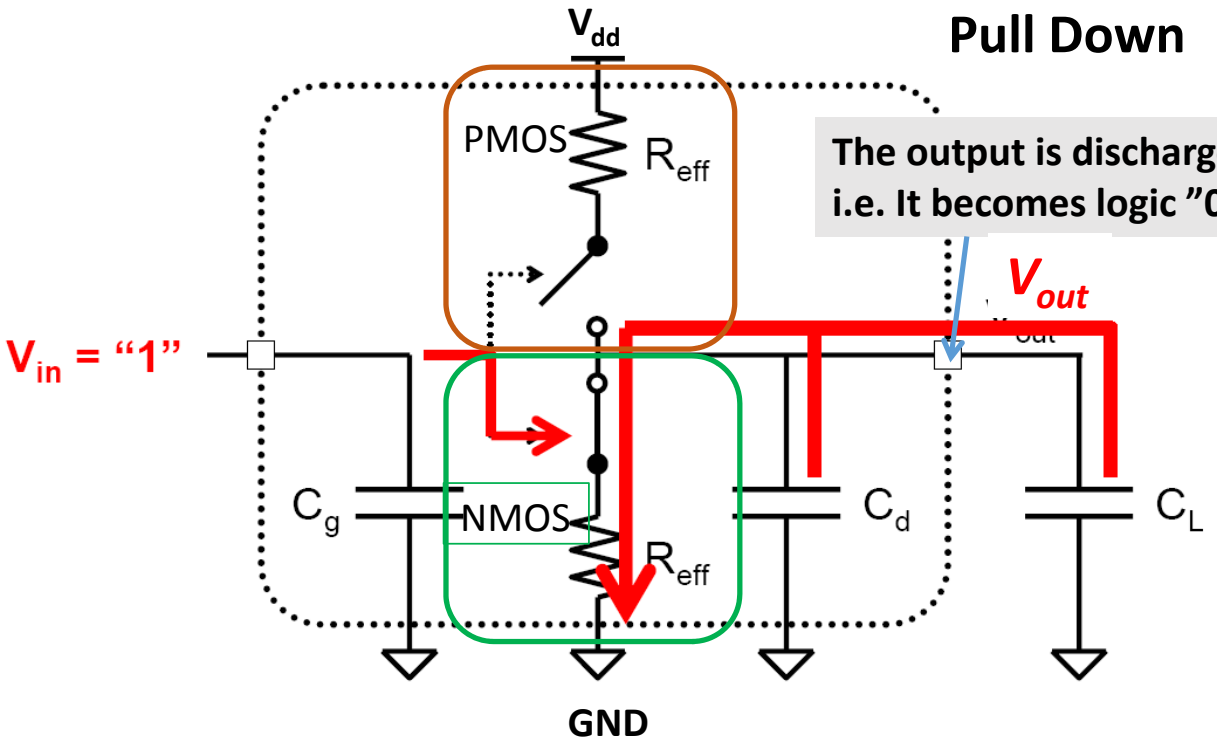


Charge RC Time Constant = R_{eff} x (C_d + C_L)

Source: MIT. Course 6.375. Lecture L06. 2006

And it takes time to discharge the output to GND

1. **The charging down to V_{DD} happens via the N transistor**
2. **This creates assymetric timing for the time it requires to rise to V_{DD} and the time it requires to fall to GND**

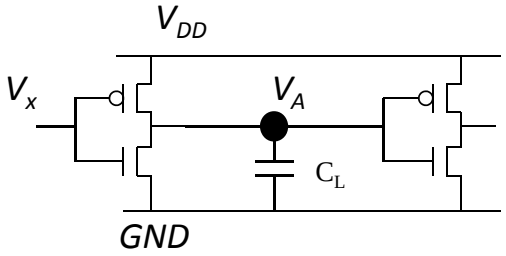


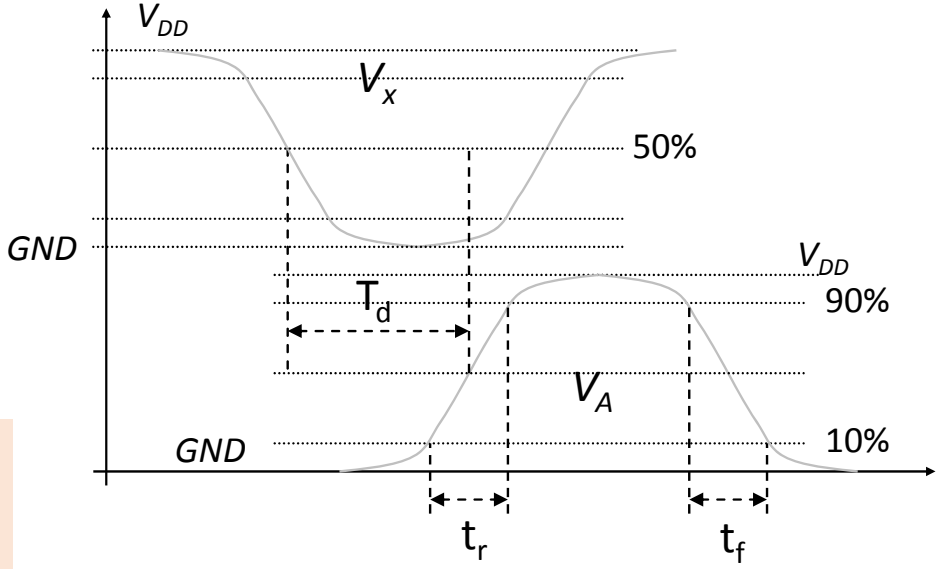
Discharge RC Time Constant = R_{eff} x (C_d + C_L)

Source: MIT. Course 6.375. Lecture L06. 2006

Rise, Fall, and Propagation Times…

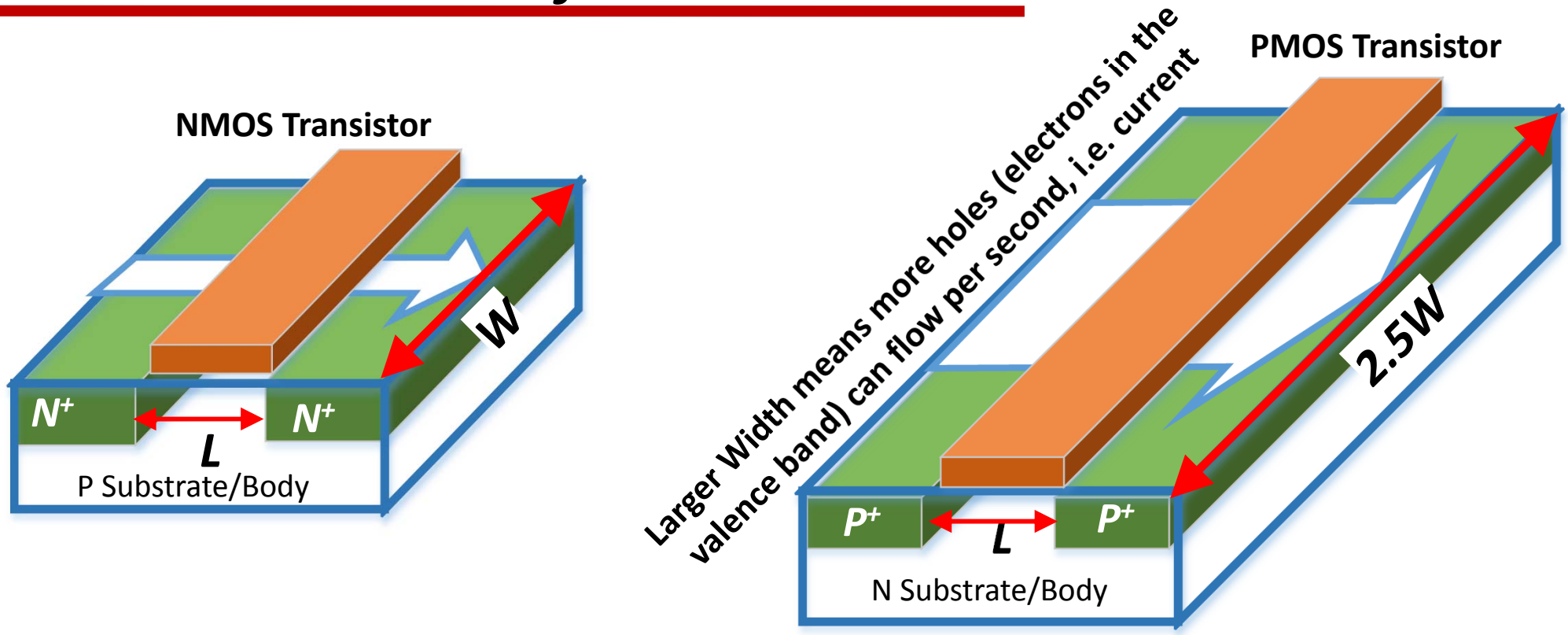
1. ***The rise and fall time shown here are symmetric, i.e. Nearly equal***
2. ***This would not be the case if the P and N transistors had the same dimension (see next slide)***
3. ***We normally make the P transistors 2 to 2.5 times wider compared to the N transistors to make sure that the P transistors have nearly equal resistance to N transistors. This makes the RC (time constant) nearly equal resulting in nearly symmetric rise and fall times.***





T_d: Propagation Delay (50% V_x - 50% - V_A)
t_r: Rise time (10% - 90%)
t_f: Fall time (90% - 10%)

In CMOS, P transistors are made wider than the N to balance the rise and fall time



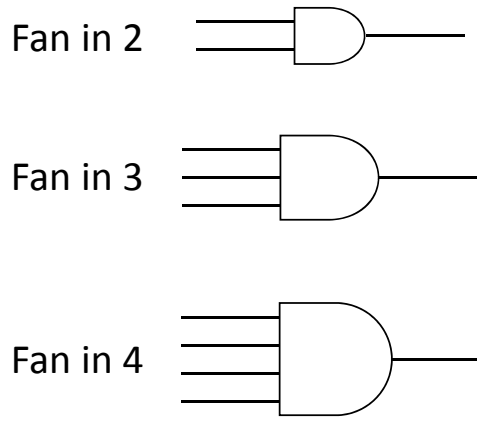
1. In CMOS, the P transistors are made wider to reduce its resistance. Reducing resistance is equivalent to increasing current for the same supply voltage (VDD) (Remember Ohm's law I =V_{DD}/ R)
2. Newer technology has reworked the material in a way that the mobility in P transistors is nearly as good as N and such assymetric widths are no longer needed

Fan In

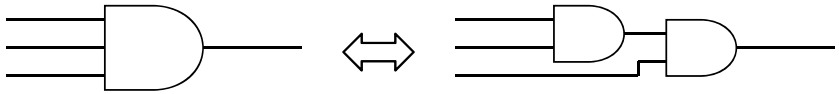
1. Fan In is the number of inputs in a logic Gate

2. What is the significance of Fan in
   Higher fan in implies higher capacitance. Higher capacitance implies slower transition to 1s and 0s. We will see later why slow transition is undesirable

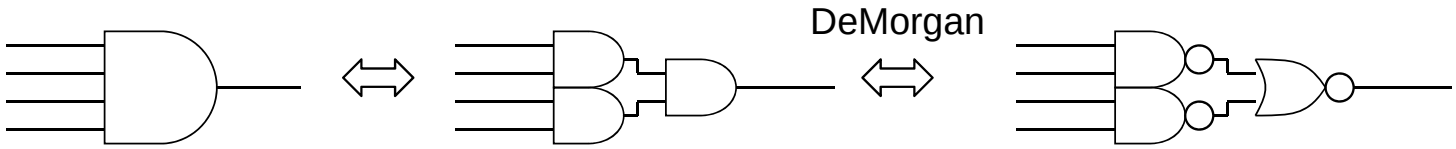
3. What can be done to reduce Fan In
   We can reduce the Fan In by using a tree structure



High Fan In is solved by using tree structures



$$a \cdot b \cdot c = a \cdot (b \cdot c)$$

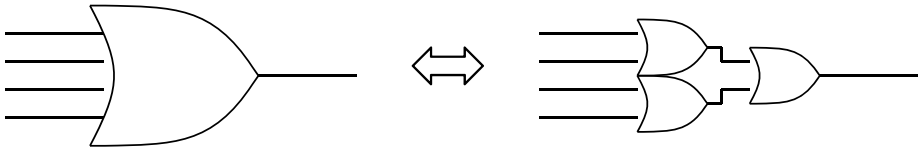


$$a \cdot b \cdot c \cdot d = (a \cdot b) \cdot (c \cdot d) \qquad a \cdot b \cdot c \cdot d = \overline{\overline{(a \cdot b)} + \overline{(c \cdot d)}}$$

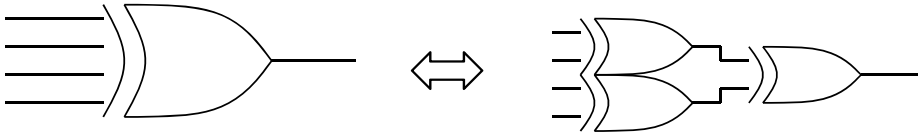
Notice that tree structure solves the problem of Fan In at the expense of greater logic depth that implies greater propagation delay

Credit: Fredrik Jonsson

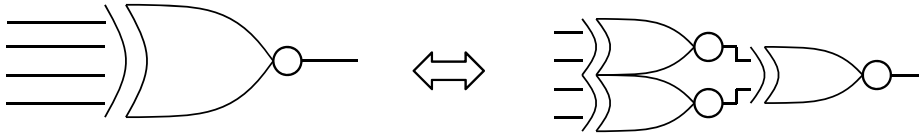
More Tree Structures



$$a+b+c+d=(a+b)+(c+d)$$



$$a \oplus b \oplus c \oplus d = (a \oplus b) \oplus (c \oplus d)$$



$$\overline{a \oplus b \oplus c \oplus d} = \overline{(a \oplus b)} \oplus \overline{(c \oplus d)}$$

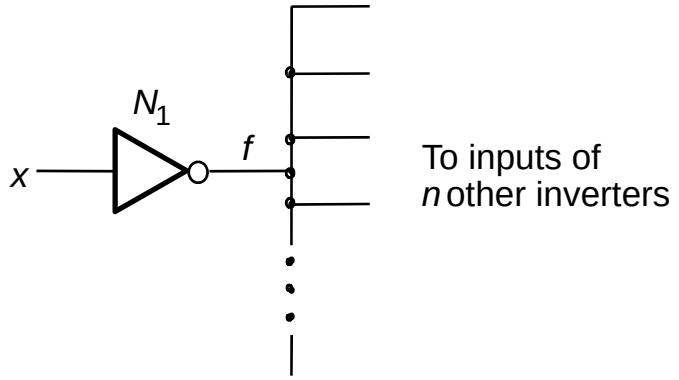
Kan you prove these equivalences

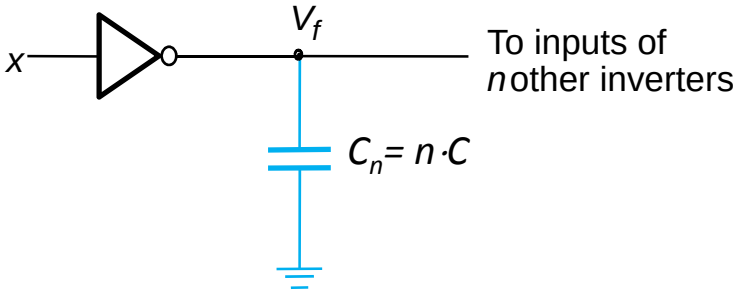
Credit: Fredrik Jonsson

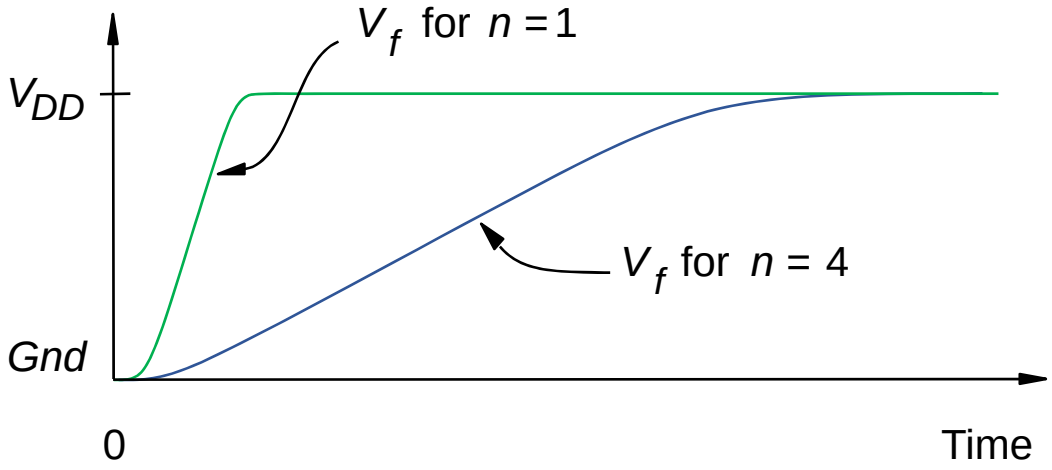
Fan out

1. Fan out is the number of inputs a logic Gate drives

2. What is the significance of Fan out
   Higher fan out implies higher capacitive load. Higher capacitive load implies slower transition to 1s and 0s.

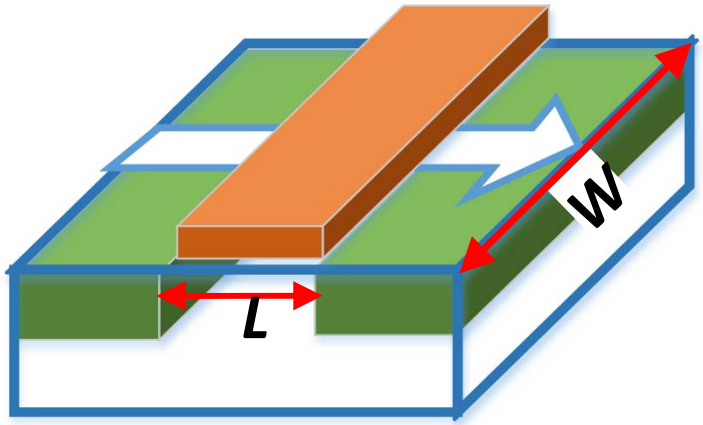


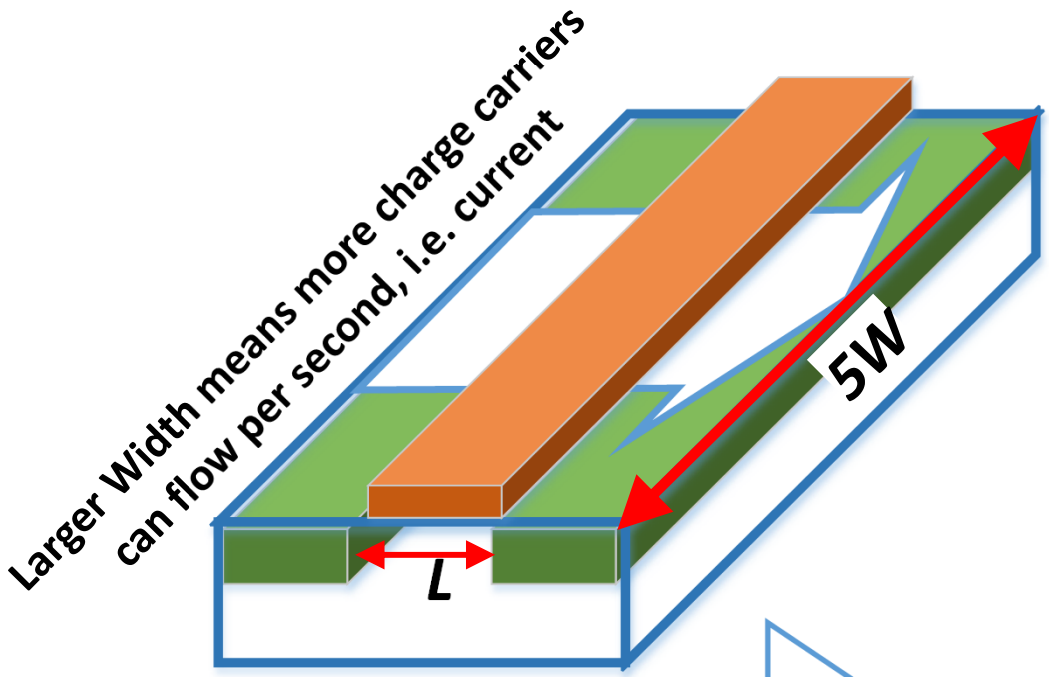




Buffers

1. Buffers are logic elements that pass the signal as it is
2. If they do not boolean algebraically transform a signal what is their use ?
3. They are used to fix two electrical issues
   a. They can restore a weak zero to strong zero and weak one to strong one
   b. Buffers can increase the amount of current available at its output to drive the logic elements driven by its output

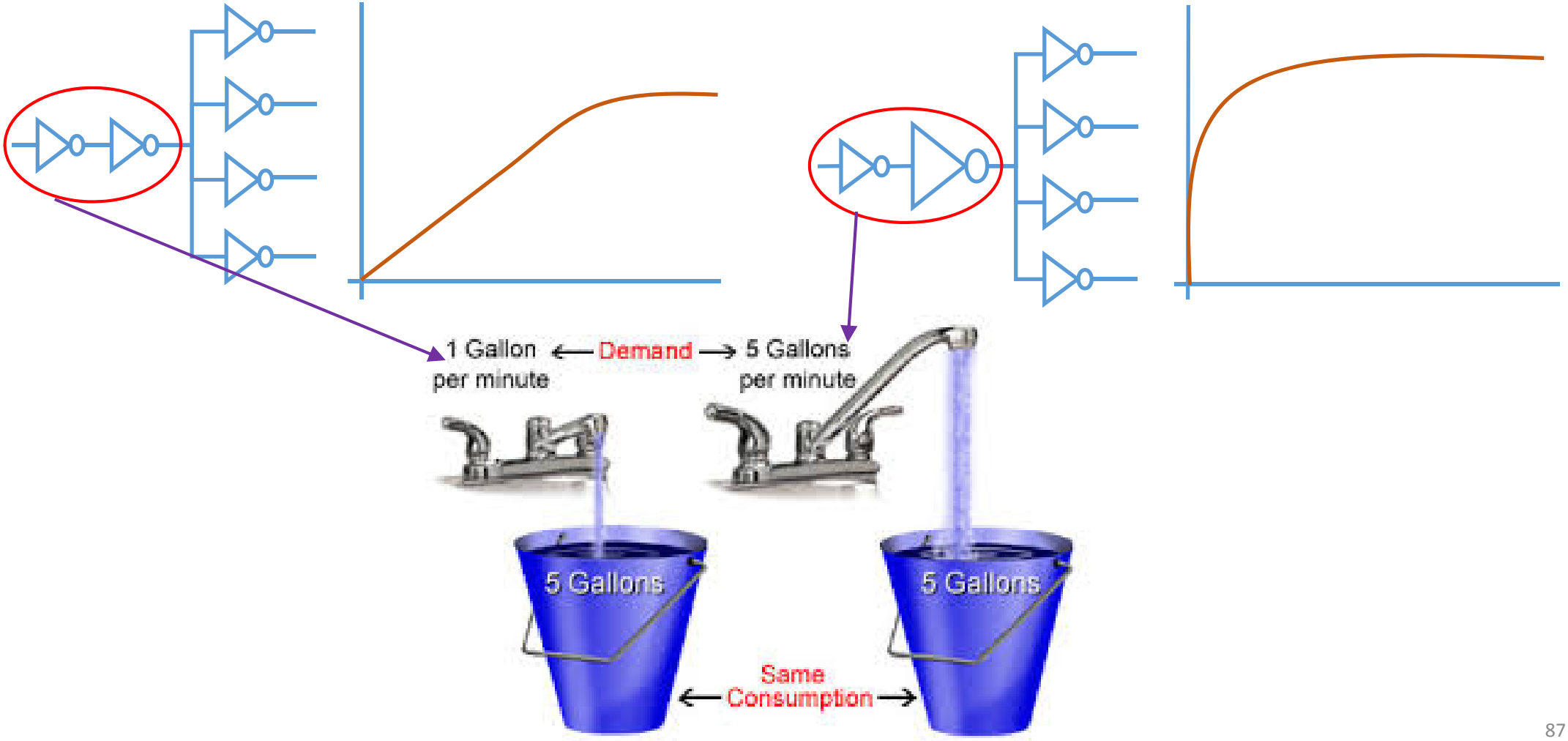




N.B: Larger CMOS buffer means both P and N transistors are made wider

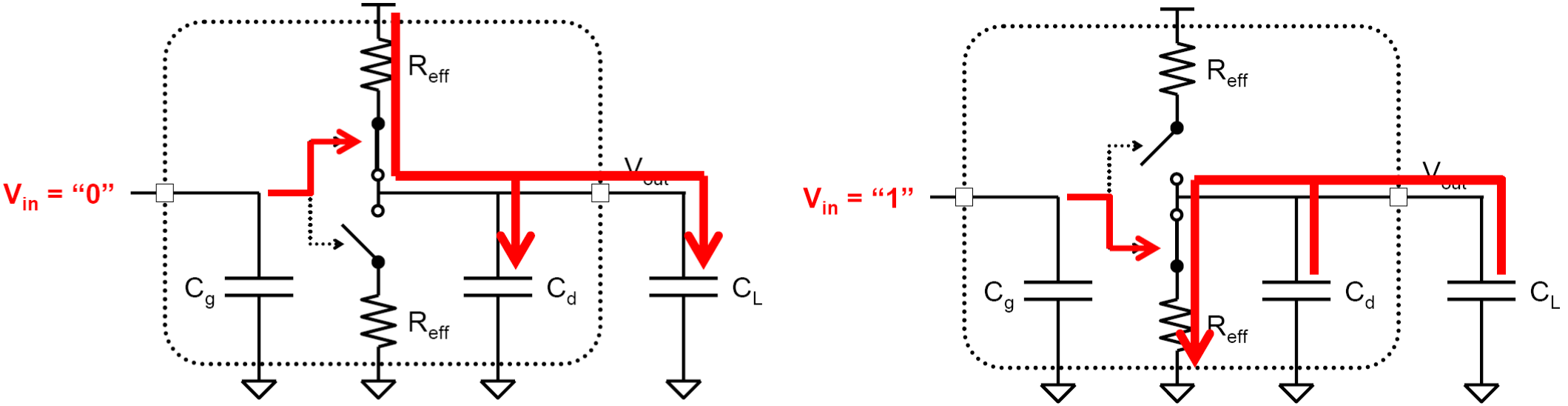
The P transistor to N transistor width ratio in small as well as large buffer will still be 2 to 2.5 times to balance the rise and fall time in each buffer.

How to deal with the high fanout problem



Power Consumption in ideal CMOS

Power in CMOS is consumed only when the inputs change that requires an output to be changed

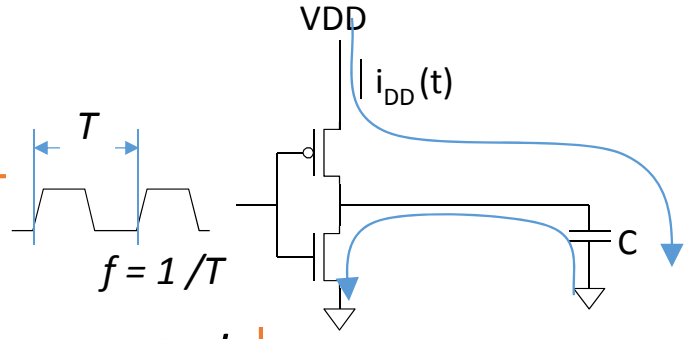


When the output is changed, power is consumed in the process of

1. ***charging the output to VDD***
2. ***or discharging the ouput to GND***

Power Consumption in CMOS

N.B. The following is a simplified derivation of power consumed in CMOS due to switching of its output state.
A more accurate calculus based derivation exists.



$$Power = Voltage \cdot Current$$
$$= V_{DD} \cdot I$$
$$= V_{DD} \cdot Q/T; \quad current\ is\ rate\ at\ which\ charge\ (Q)\ is\ transported$$
$$= V_{DD} \cdot (C \cdot V_{DD}) / T; \quad Q = C \cdot V$$
$$= C \cdot V_{DD}^2/T$$
$$= C \cdot V_{DD}^2 \cdot f$$

$(1/2)\ C \cdot V_{DD}^2 \cdot f$

Lost in resistance as heat
This constitutes the power consumption

$(1/2)\ C \cdot V_{DD}^2 \cdot f$

The other half is stored in capacitor

Power Consumption in CMOS

$$P_{switching} = (1/2)\, C\, V_{DD}^2\, f$$

1. The factor ***f*** in $P_{switching}$ equation and example above needs some elaboration

2. What the equations says is that ***C*** farads of capcitance switches, i.e., either charged to V_{DD} or discharged to GND with a frequency of ***f*.**

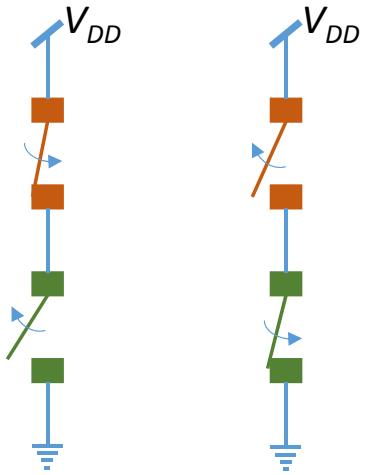
3. If ***f*** is the clock frequency and ***C*** is the total capacitance of the logic elements (gates etc.) and wires, it implies that all nodes toggle in every clock cycle.

4. This is far from true. On an average only 20% of capacitance toggles or more accurately ***k*** percent of capcitance toggles. The factor ***k*** is found by simulation

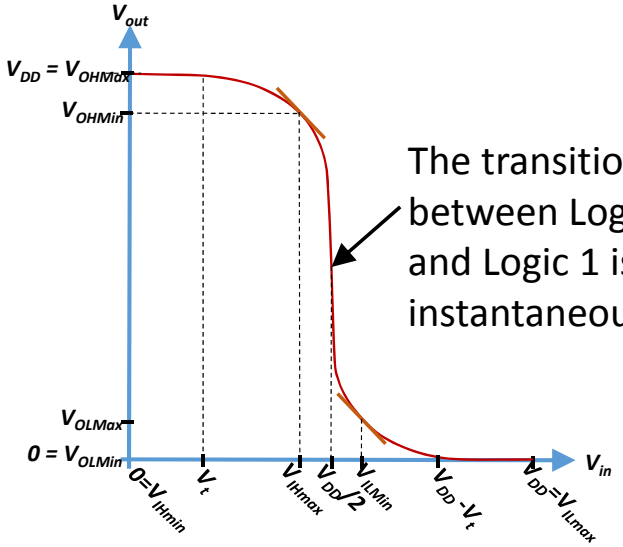
5. So it is more accurate to say that Pdynamic = $(k/100)(1/2)\, C\, V_{DD}^2\, f$;
   where k is found by simulation or estimated - ~20%

The Crowbar Current

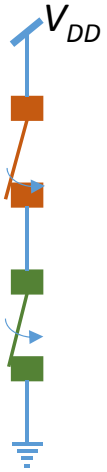
The Ideal Property of CMOS that is not True



Ideally, either Pull Up Network (PUN) composed PMOS transistors conducts or the Pull Down Network (PDN) composed on NMOS transistors conducts. Both are not supposed to conduct at the same time



The reality



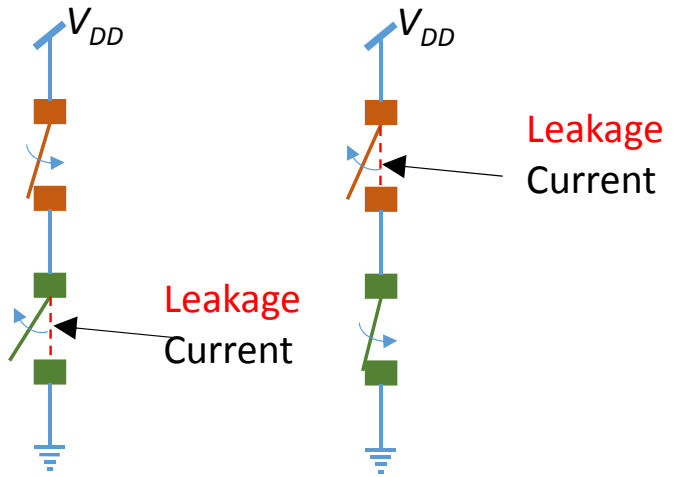
1. The PMOS and NMOS transistors take finite time to turn ON and turn OFF.
2. This creates an overlap when both the transistors are partially conducting. The creates a temporary path between VDD and GND for a current to flow that results in power consumption. This unintended current is called ***crowbar current*** and the power consumption crowbar power.
3. Crowbar current flows everytime output of a CMOS logic toggles.
4. Slow rise and fall times makes turning ON and turning OFF of the transistors slower creating a bigger overlap and thus resulting in more crowbar current

The Static Power Consumption Component

$P_{switching}$ and $P_{crowbar}$ are both consumed everytime the output of CMOS logic toggles.
These components are called ***Dynamic Power Consumption***

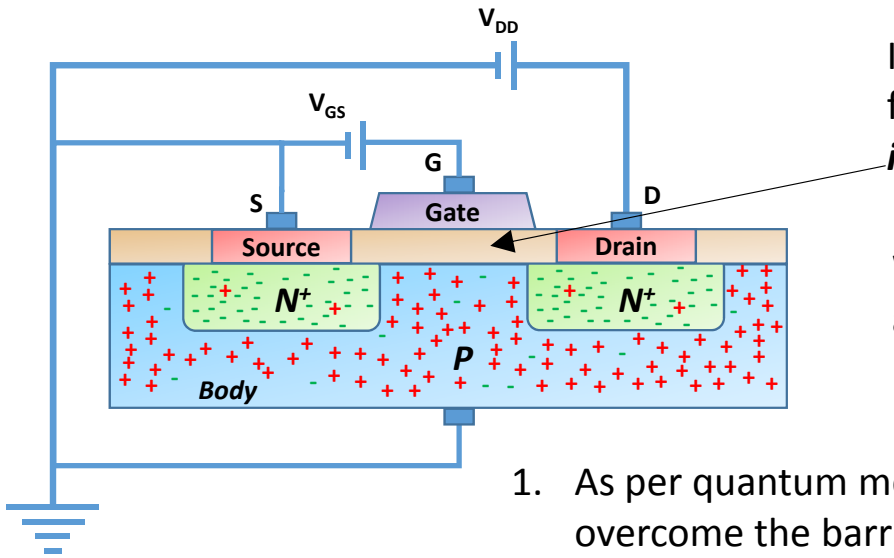
There are two more power power consumption components that consumes power even when the output does not toggle or the input changes
These components are called ***Static Power Consumption***

1. Ideally, PMOS and NMOS transistors are supposed to have infinite resistance when they are sub-threshold, i.e., when they are OFF
2. However, even when they are OFF, the resistance between Source and Drain is very large but not infinite.
3. This results in a small component of current that conducts even when the PMOS and NMOS transistors are OFF
4. Power consumption that happens because of this ***leakage*** current is called $P_{threshold}$.
5. $P_{threshold}$ is static power consumption because it is consumed even when the CMOS logic is static, i.e, even when the inputs and outputs are not changing



A small sub-threshold current flows even when the PMOS and NMOS transistors are OFF

The Static Power Consumption Components

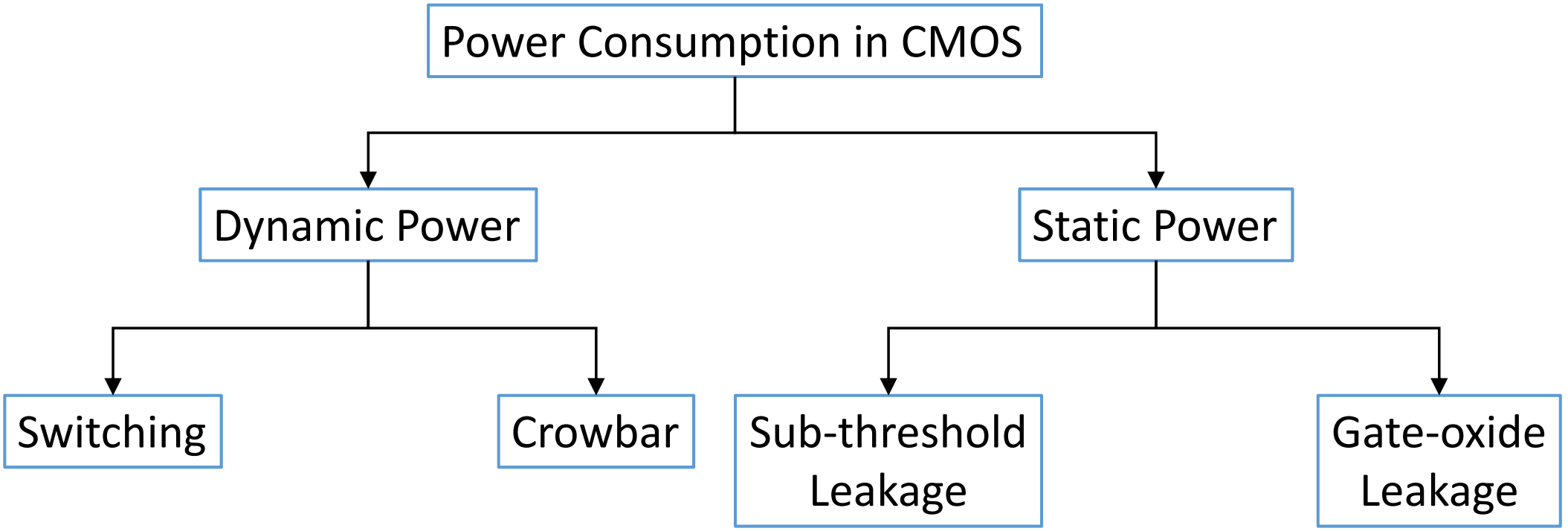


Ideally, in PMOS and NMOS transistors, there should be flow of current between Gate and Body because there is an ***insulator*** between them

With Technology scaling, the gate oxide has become thinner and thinner. Today, it is only a few atoms thick

1. As per quantum mechanics, a small amount of electrons will have sufficient energy to overcome the barrier posed by insulator and leak into body.
2. The thinner the insulator, greater the percentage of electroncs that can leak
3. This is called ***gate-oxide leakage current***
4. Power consumption due to this leakage current is called $\boldsymbol{P_{gate\text{-}oxide}}$.
5. $\boldsymbol{P_{gate\text{-}oxide}}$ is also a static power consumption component because it does not require inputs or outputs to toggle.

Power Consumption Components - Summary



The total static power consumption in all electronic devices in West Europe is 50 Tera Watt Hour
This power is sufficient to drive ALL RAILWAYs – long distance, inter-city and metro(tunnelbana) – in West Europe

CMOS - Summary

- Semiconductor Primer
- NMOS
- PMOS
- CMOS
- Realization of arbitrary boolean functions as CMOS
- Transmission Gates
- Multiplexor
- Tri-states
- Fan-in, Fan-out
- Delay in CMOS
- Power Consumption in CMOS