

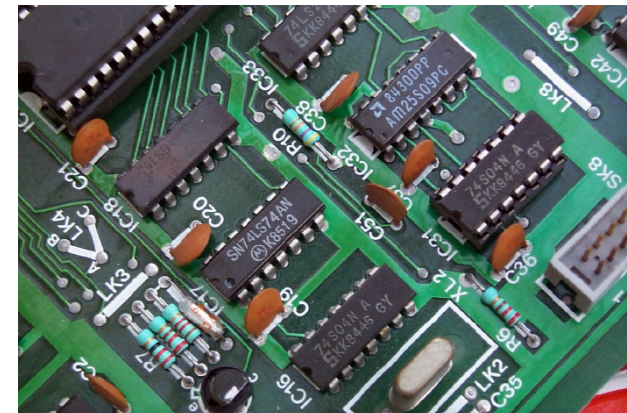
# Analog & Digital Electronics

Course No: PH-218

## Lec-28: Logic Gates & Family

Course Instructor:

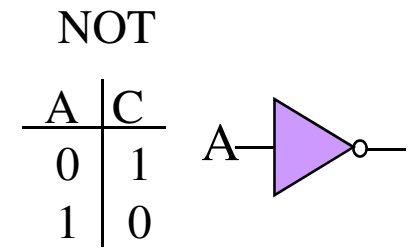
❖ Dr. A. P. VAJPEYI



Department of Physics,  
Indian Institute of Technology Guwahati, India

# Digital Logic Gates

The electrical circuits which perform logical operations are called *gates*.



All data manipulation is based on *logic*

Logic follows well defined rules, producing predictable digital output from certain input.

Main Logic gates are AND, OR, NOT, NAND, NOR and XOR

AND

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

OR

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

NAND

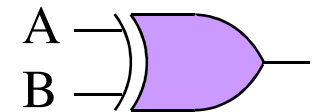
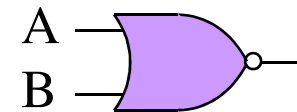
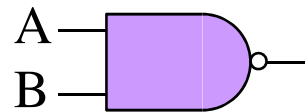
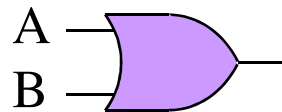
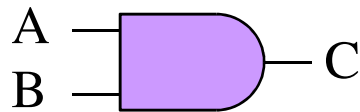
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

NOR

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

XOR

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



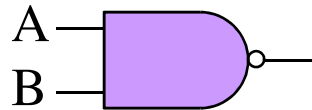
Digital logic gates NAND and NOR are called universal logic gate because we can construct all other logic gates using NAND gate or NOR gate alone.

NAND gate can be built using 4 MOSFETs ( 2NMOS and 2PMOS).

# NAND – A Universal Logic Gates

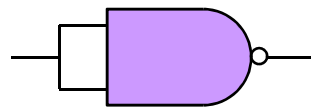
NAND

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



NOT

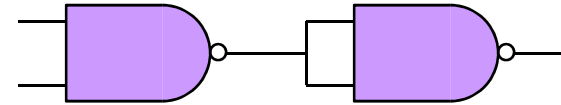
A	C
0	1
1	0



AND

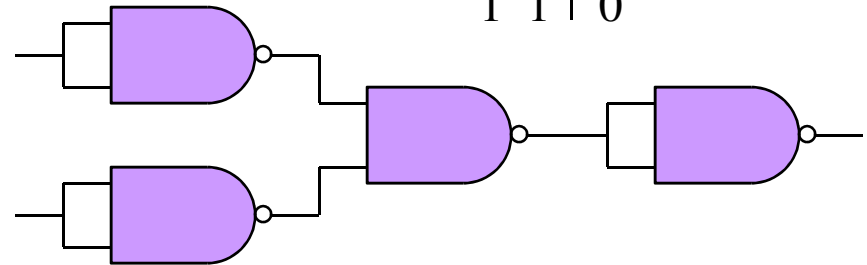
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

invert output (invert NAND)



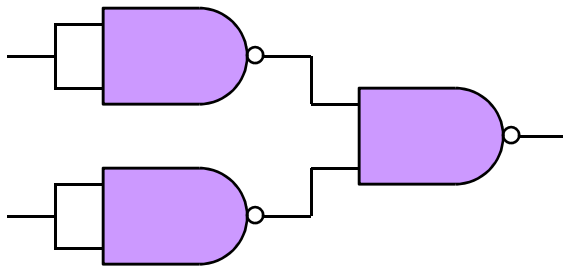
NOR

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



OR

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



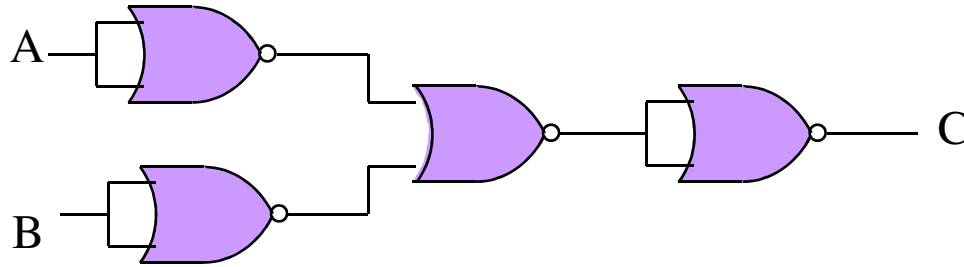
invert both inputs

invert inputs *and* output (invert OR)

# NOR – A Universal Logic Gates

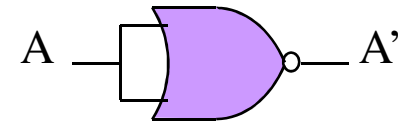
NOR

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



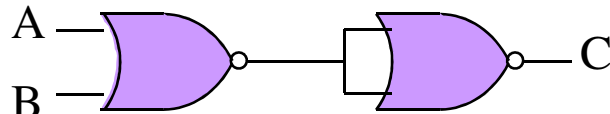
NOT

A	C
0	1
1	0



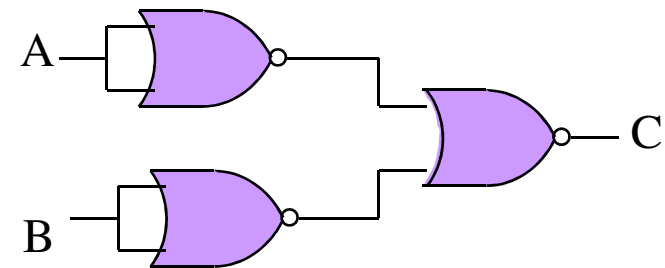
OR

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

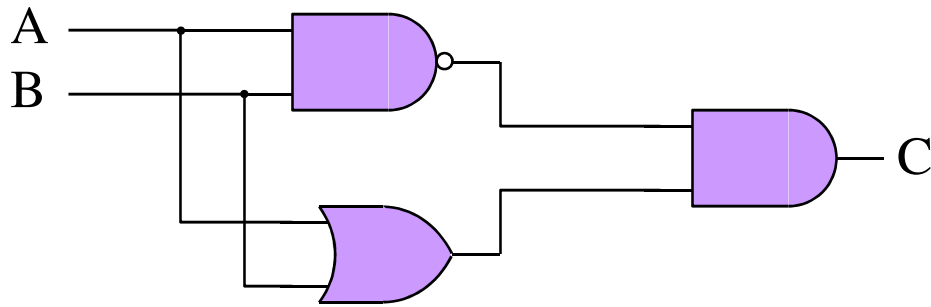


AND

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



## XOR –Logic Gates



- $XOR = (A \text{ NAND } B) \text{ AND } (A \text{ OR } B)$
- And this you already know you can make from composite NAND gates (though requiring 6 total)
- Then, obviously, XNOR is the inverse of XOR
  - so just stick an inverter on the output of XOR

### **Exclusive OR operator: operating on two variables**

*A and B is true if A or B is a 1, but not when both A and B are 1, i.e. it excludes both A and B being a 1; symbol  $\oplus$*

# Logic Family / Level Of Integration

Scheme	# gates / chip
Small Scale Integration (SSI)	<12
Medium Scale Integration (MSI)	12 - 99
Large Scale Integration (LSI)	1000
Very large Scale Integration (VLSI)	10k
Ultra large Scale Integration (ULSI)	100k
Giga Scale Integration (GSI)	1Meg

Note: Ratio gate count/transistor count is roughly 1/10

- IC logic gates fall under SSI, combinational logic circuits fall under MSI, and Microprocessor system come under LSI and VLSI.

## Digital Logic Family

- Logic families can be classified broadly according to the technologies they are built with
- There are various logic families namely –
  - ✓ Diode logic (DL)
  - ✓ Resistor-Transistor logic (RTL)
  - ✓ Diode-Transistor logic (DTL)
  - ✓ Emitter coupled logic (ECL)
  - ✓ Transistor-Transistor logic (TTL)
  - ✓ CMOS logic
- TTL and CMOS logic family is most widely used IC technologies.
- Within each family, several subfamilies of logic types are available, with different rating for speed, power consumption, temperature range, voltage level and current level.

# Nomenclature of Logic family

The different manufacturers of digital logic ICs have standardized a numbering scheme so that basic part number will be same regardless of the manufacturer.

The prefix of the part number represents the manufacturer code and the suffix at the middle denotes the subfamily of the ICs and suffix at the end denotes the packaging type.

For example: If the part number is S74F08N. The 7408 is the basic number used by all manufacturer for quad AND gate. The S prefix is the manufacture's code for Signetics, F stands for FAST TTL subfamily, and the N suffix at the end is used to specify the plastic dual in line packaging

## Suffix used for packaging:

- N - Plastic dual in line package
- W - Ceramic flat pack
- D - Surface mounted plastic package

## Suffix used for subfamily:

- 74H04 - High-speed
- 74L045- Low-Power
- 74S04 - Uses a Schottky Diode
- 74ALS04 - Advanced low power Schottky

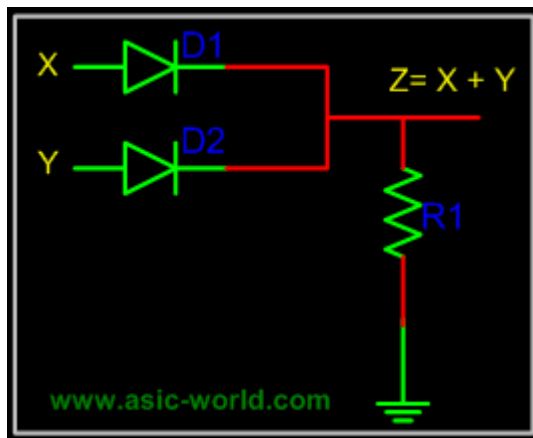
## Prefix used for manufacturers:

- S - Signetics
- SN - Texas Instruments
- DM - National Semiconductor



# Diode Logic Family

In diode logic family, all the logic is implemented using diodes and resistors.  
One basic thing about the diode, is that diode needs to be forward biased to conduct.



OR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Diode Logic suffers from voltage degradation from one stage to the next.

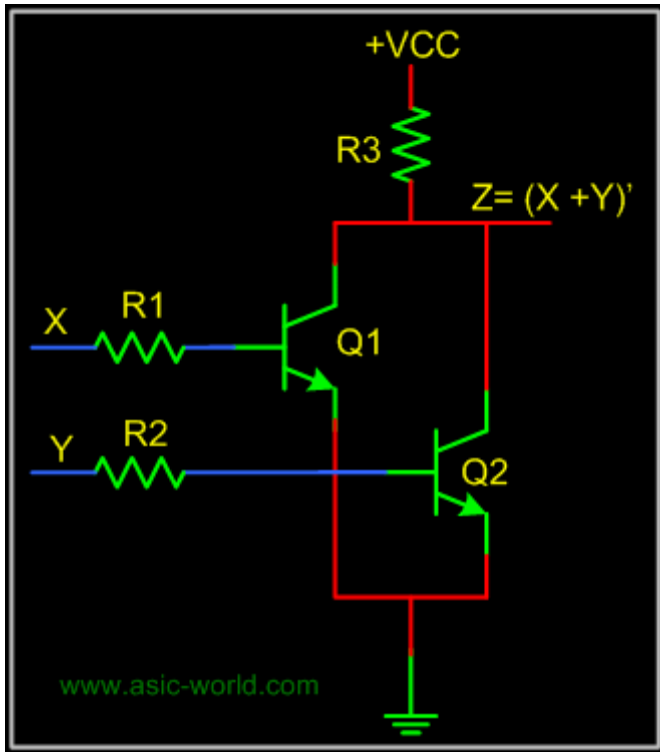
Diode Logic only permits OR and AND functions, cannot perform a NOT function.

Diode Logic is used extensively but not in integrated circuits

**How to build AND gate using Diode logic?**

# Resistor Transistor Logic (RTL) Family

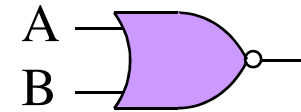
In RTL (resistor transistor logic), all the logic are implemented using resistors and transistors. One basic thing about the transistor (NPN), is that HIGH at input causes output to be LOW (i.e. like a inverter).



**NOR gate using RTL**

NOR

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0



RTL draw a significant amount of current from the power supply for each gate. Another limitation is that RTL gates cannot switch at the high speeds used by today's computers, although they are still useful in slower applications

# Transistor Transistor Logic (TTL) Family

One basic function of TTL IC is as a complimenting switch or inverter.

When  $V_{in}$  equals 1 (+5V), the transistor is turned on (saturation) and  $V_{out}$  equals 0 (0V).

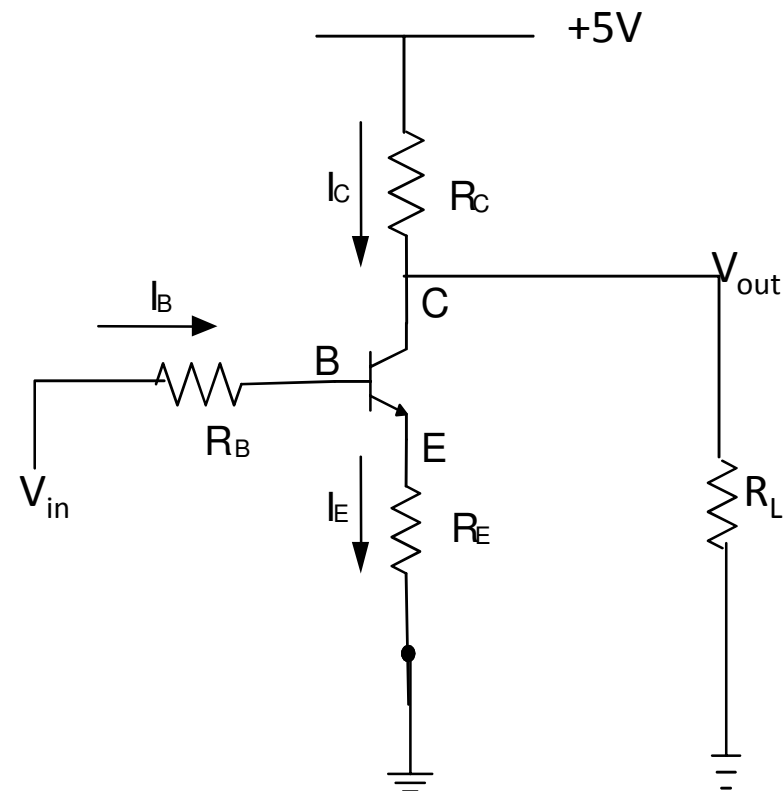
When  $V_{in}$  equals 0 (0V), the transistor is turned off and  $V_{out}$  equals 1 (5V), assuming  $R_L > R_C$

$$V_{out} = V_{CC} \frac{R_L}{(R_C + R_L)}$$

Thus level 1 of inverter output is very much dependent on  $R_L$ , which can typically vary by factor of 10. Thus we need very small  $R_C$  compared to  $R_L$  i.e.  $R_L \gg R_C$ .

But when transistor is saturated ( $V_{out} = 0V$ ),  $I_C$  will be very large if  $R_C$  is very small.

Thus we need large  $R_C$  when transistor is in saturation and small  $R_C$  when transistor is off.



The idea of variable  $R_C$  is accommodated by TTL IC.

# Transistor Transistor Logic (TTL) Family

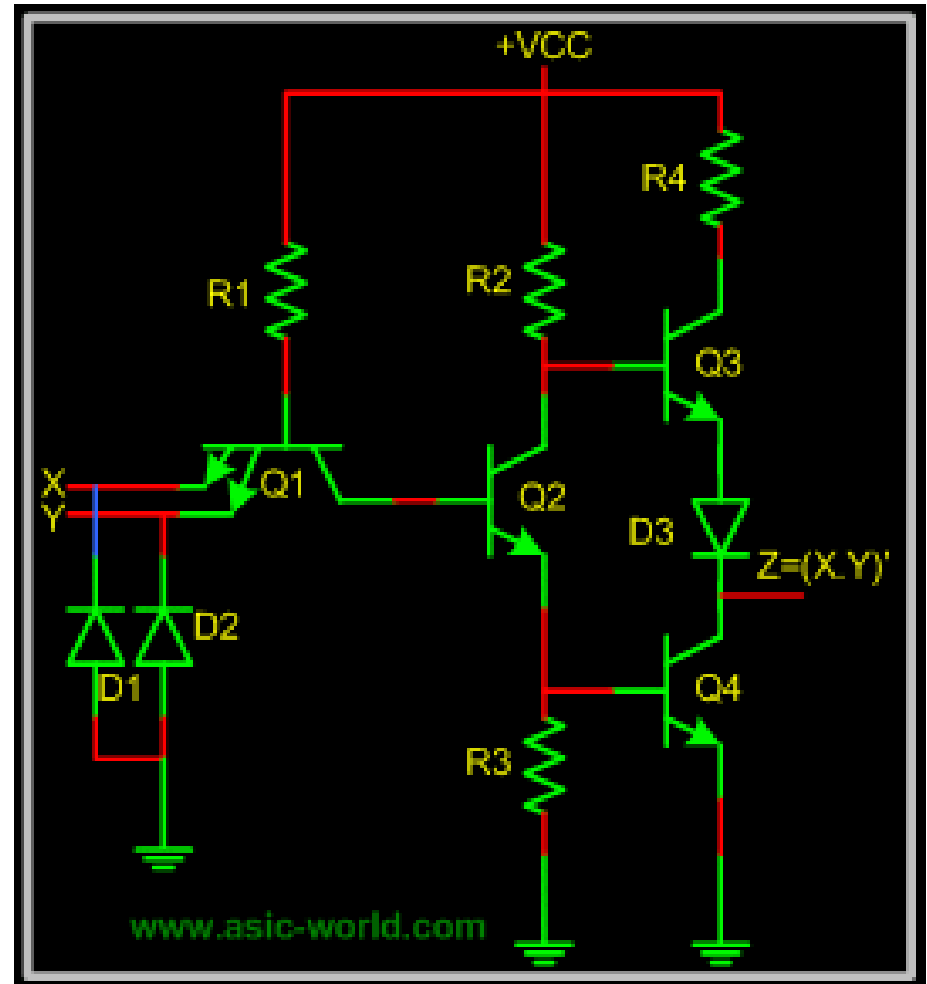
The idea of variable  $R_C$  is accommodated by TTL IC. It uses another transistor Q3 in place of  $R_C$  to act like a varying resistance.

Q3 is cutoff (act like a high  $R_C$ ) when output transistor Q4 is saturated and Q3 is saturated (act like a low  $R_C$ ) when output transistor Q4 is cutoff. Thus one transistor is ON at one time.

The combination of Q3 and Q4 is called **totem pole arrangement**.

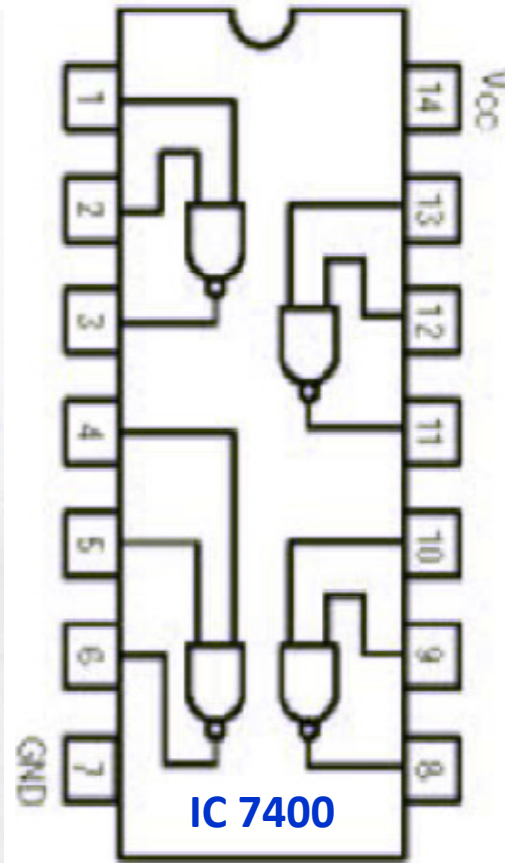
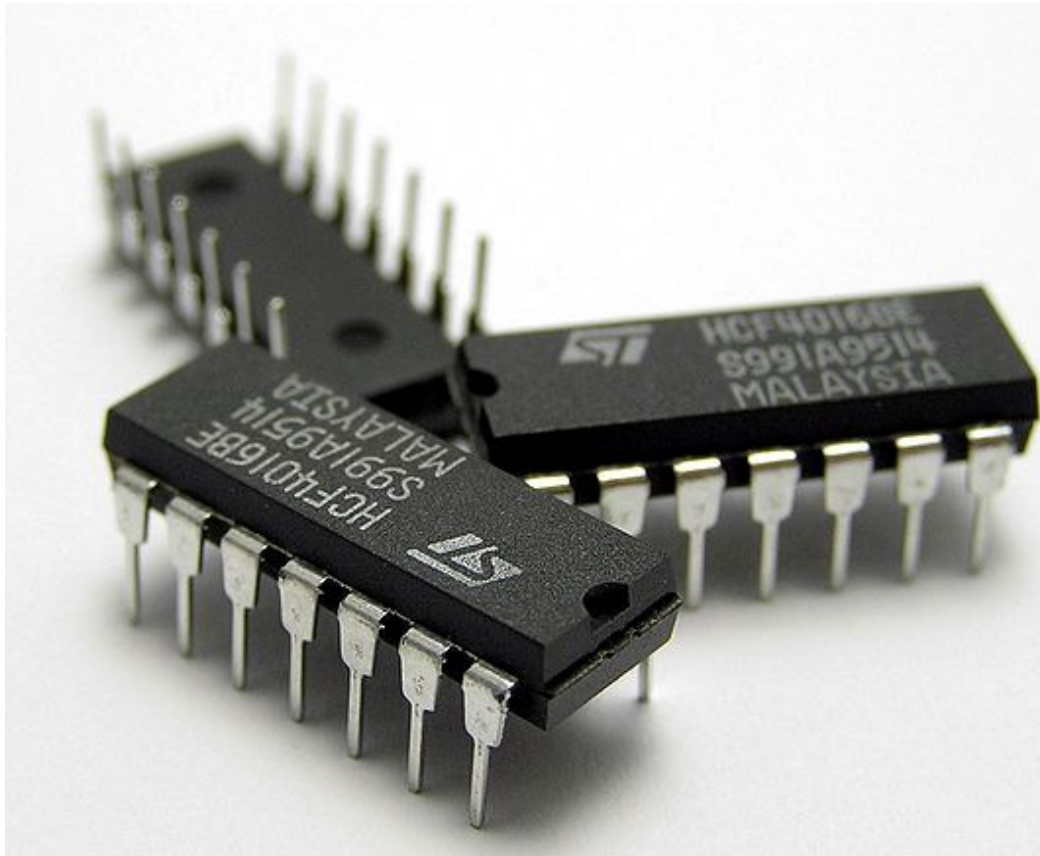
Q1 is called input transistor, which is multi-emitter transistor, that drive transistor Q2 which is used to control Q3 and Q4.

Diode D1 and D2 is used to protect Q1 from unwanted negative voltages and diode D3 ensures when Q4 is ON, Q3 is OFF.



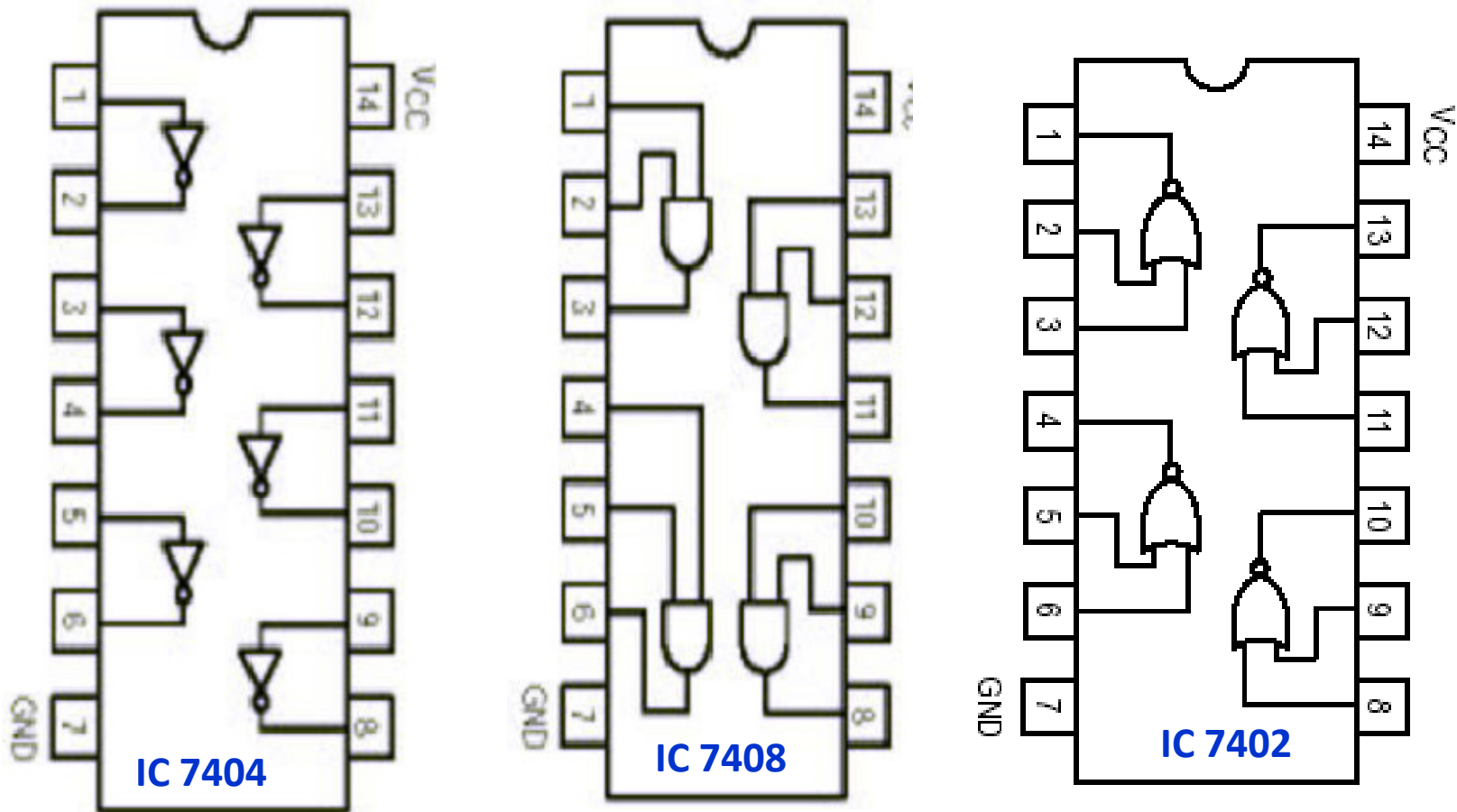
**Multi-emitter input transistor is a striking feature of TTL logic family.**

# Transistor Transistor Logic - Dual In Line Packaging (DIP)



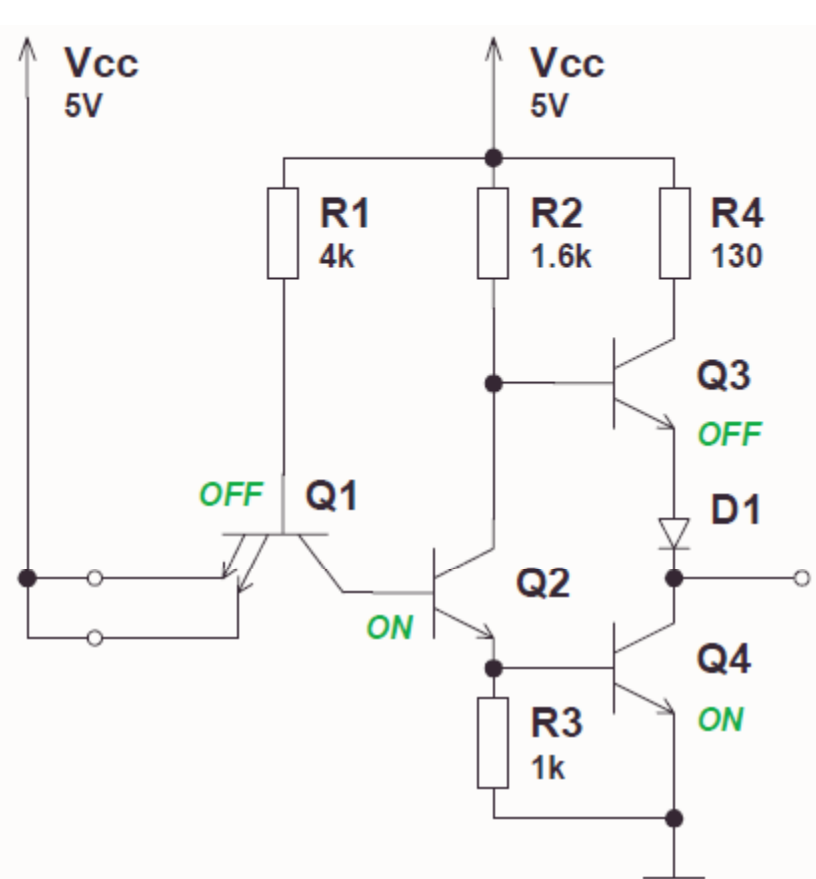
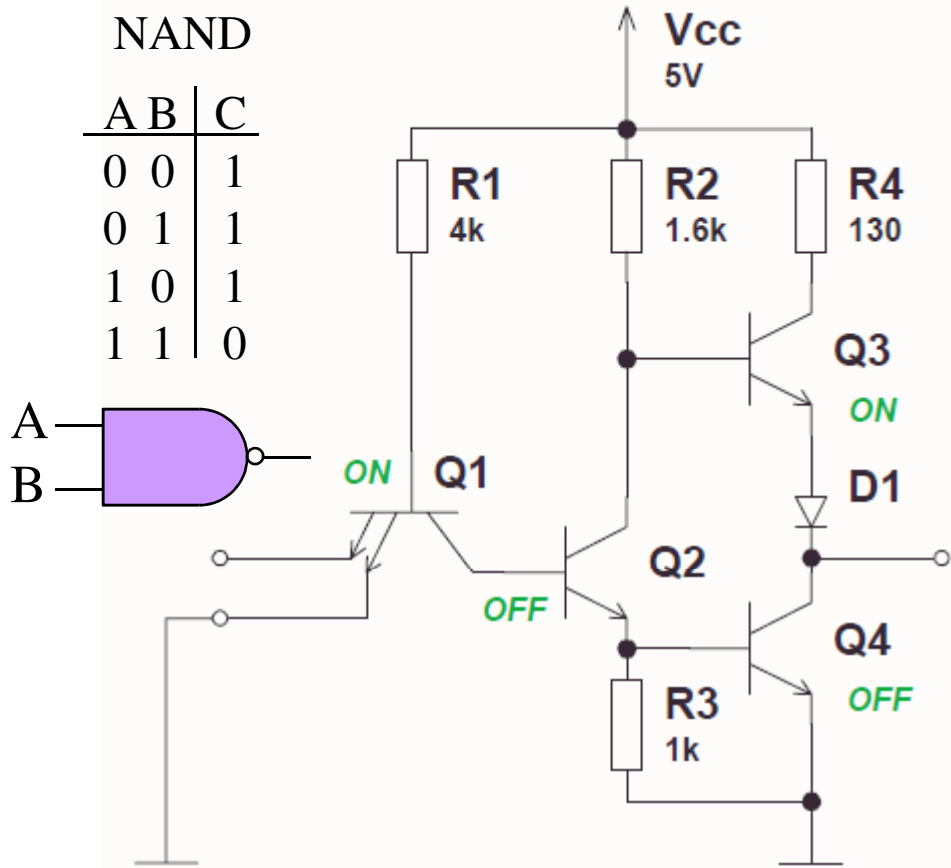
Dual In Line package (DIP) is the most common pin layout for integrated circuits. The pins are aligned in two straight lines, one on each side of the IC.

# Transistor Transistor Logic - Dual In Line Packaging (DIP)



Some common digital ICs used in labs are: IC7400 (Quad NAND gate), IC 7404 ( hex inverter), IC 7408 (quad AND gate) and IC7402 (quad NOR gates).

# NAND gate using TTL logic / Static analysis



When **one or both inputs low** (connected to GND), base-emitter junction of Q1 is forward bias so Q1 ON (saturated) and output at collector will be low making Q2 off.

- Q4 off and Q3 & D1 on making **output HIGH**.
- Power dissipation in R1, Q1, R2, R4, Q3, D1

When **inputs high**, base-emitter junction of Q1 is reverse bias so Q1 OFF and output at collector will be high making Q2 ON.

Q4 ON and Q3 & D1 OFF so **output is LOW**.

- Power dissipation in R1, Q1, R2, Q2, R3, Q4.

# Performance Parameters of logic families

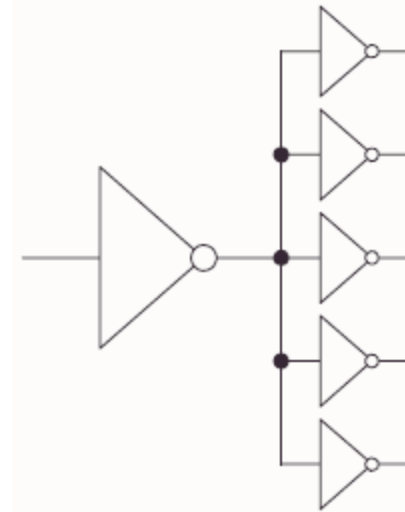
- **Fan-out**
- **Input and Output Voltage level**
- **Noise Margin**
- **Rise and Fall time, and Propagation delay**
- **Power Dissipation.**



# Input / Output Current and Fan-out

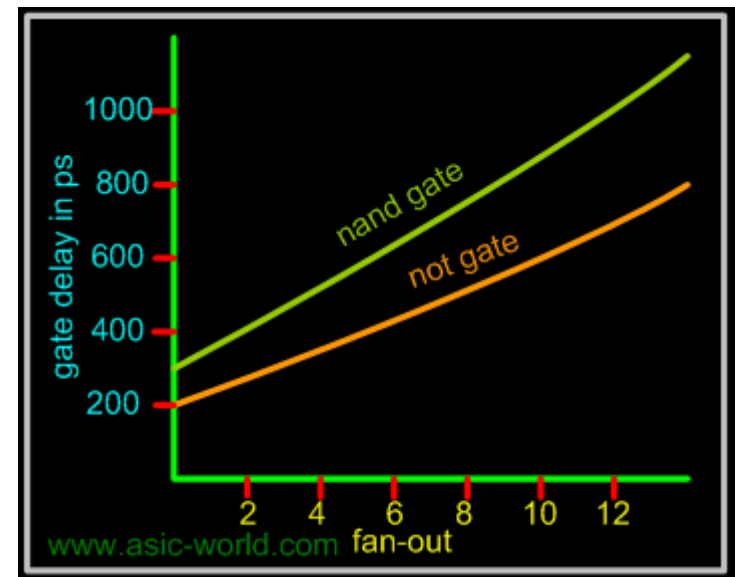
The fan-out of a subfamily is defined as the number of gate inputs of the same subfamily that can be connected to a single output without exceeding the current ratings of the gate.

A typical fan-out for most TTL subfamilies is 10.



The fan-out really depends on the amount of electric current a gate can source or sink while driving other gates. The effects of loading a logic gate output with more than its rated fan-out will degrade the performance of the circuit.

The gate delay increases with increase in fan-out.



## Input / Output Current and Fan-out

To determine fan-out, one must know how much input current gate load draws ( $I_{in}$ ) and how much output current the driving gate can supply ( $I_o$ ).

The output current capability for the HIGH condition is abbreviated  $I_{oH}$  and is called source current.  $I_{oH}$  for the 7400 is -400 $\mu$ A maximum. (- sign shows current is leaving the gate)

The input current required under HIGH condition is abbreviated  $I_{IH}$  and for 74xx subfamily is equal to 40 $\mu$ A maximum. Fan-out =  $400/40 = 10$

For the LOW condition, the maximum output current for the 74xx subfamily is 16mA, and the input requirement is -1.6mA maximum.

The fan-out is usually same for both the HIGH and LOW conditions for 74xx subfamily; if not, we use the lower of the two.

Because a LOW output level is close to 0V, the current actually flows into the output terminal and sinks down to ground. This is called sink current.

# Input / Output Voltage for TTL family

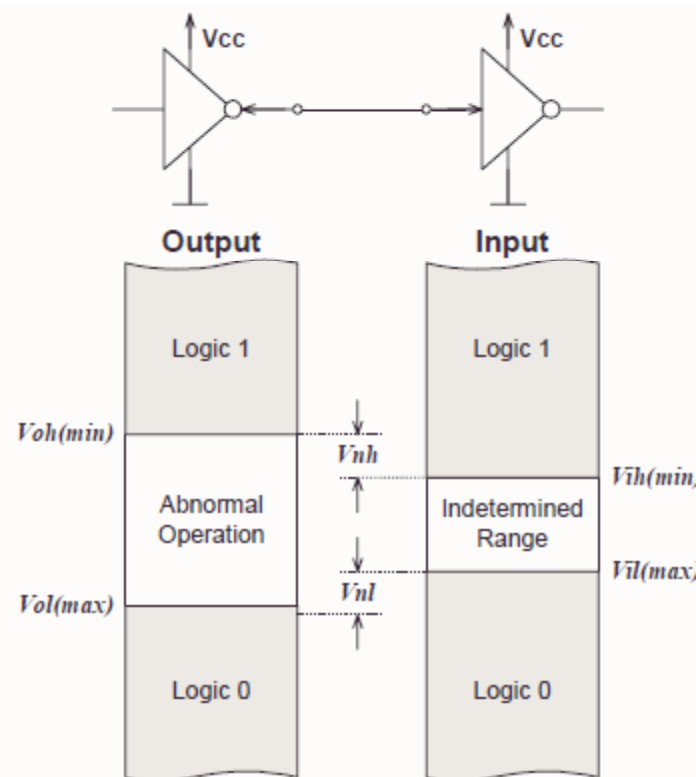
There is a limit on voltage until it is considered HIGH. As we draw more and more current out of the HIGH level output, the output voltage drops lower and lower, until finally it will not be recognized as a HIGH level anymore by the other TTL gates that it is feeding.

- $V_{OHmin}$  : The minimum output voltage in HIGH state (logic '1') = 2.4V for TTL

- $V_{OLmax}$  : The maximum output voltage in LOW state (logic '0') = 0.4V for TTL

- $V_{IHmin}$  : The minimum input voltage guaranteed to be recognised as logic 1.  
 $V_{IHmin} = 2 \text{ V}$  for TTL

- $V_{ILmax}$  : The maximum input voltage guaranteed to be recognised as logic 0.  
 $V_{ILmax} = 0.8 \text{ V}$  for TTL



# Noise Margin for TTL family

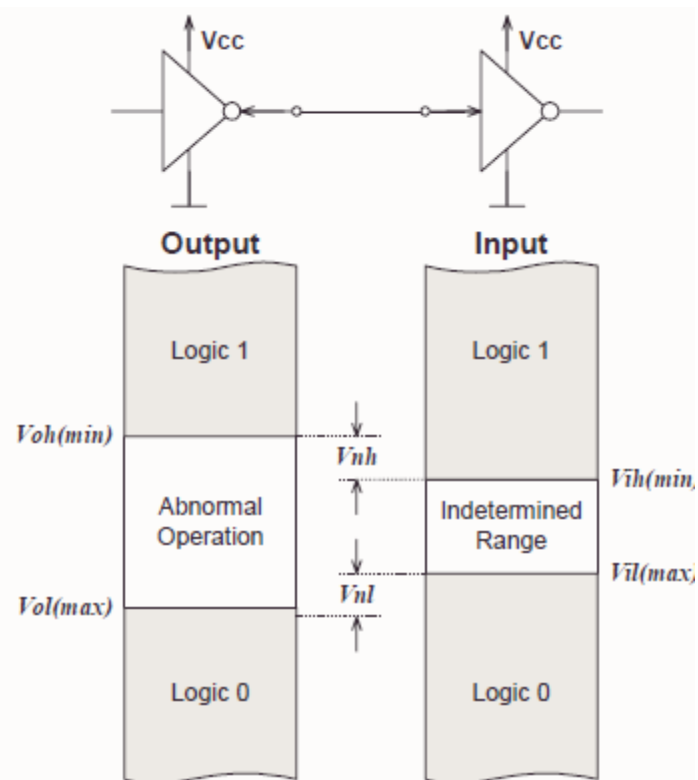
**Noise-Margin** measures how much external electrical noise a gate can withstand before producing an incorrect output. TTL will take anything below about 0.8 volt as a 0, and anything above about 2 volts as a high.

**LNM (Low noise margin):** The largest noise amplitude that is guaranteed not to change the output voltage level when superimposed on the input voltage of the logic gate (when this voltage is in the LOW).

$$LNM = V_{I_{Lmax}} - V_{O_{Lmax}} = 0.8V - 0.4V = 0.4V$$

**HNM (High noise margin):** The largest noise amplitude that is guaranteed not to change the output voltage level if superimposed on the input voltage of the logic gate (when this voltage is in the HIGH).

$$HNM = V_{O_{Hmin}} - V_{I_{Hmin}} = 2.4 - 2.0 = 0.4V$$



# Switching time and Propagation delay time

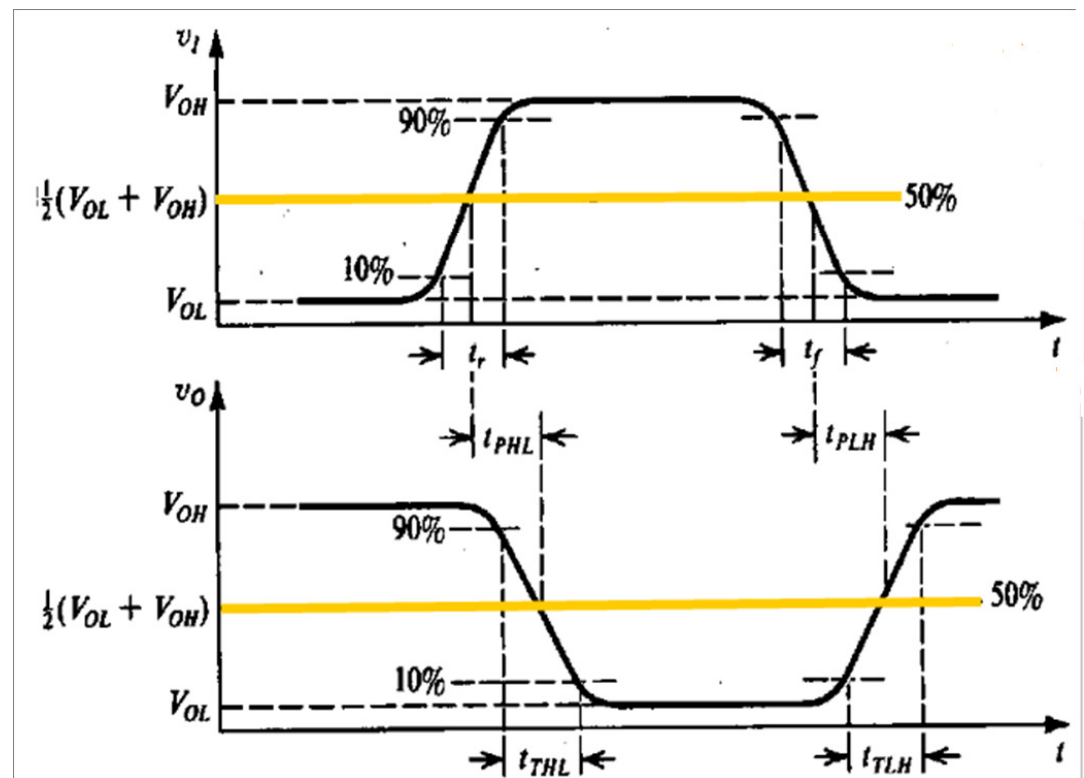
- The rise time ( $t_r$ ) is the length of time it takes for a pulse to rise from its 10% point up to its 90% point.
- The fall time ( $t_f$ ) is the length of time it takes for a pulse to fall from its 90% point to its 10% point.

The **propagation delay time** is defined as the average of low-to-high ( $t_{PLH}$ ) propagation delay time and the high-to-low ( $t_{PHL}$ ) propagation delay time.

propagation delay time

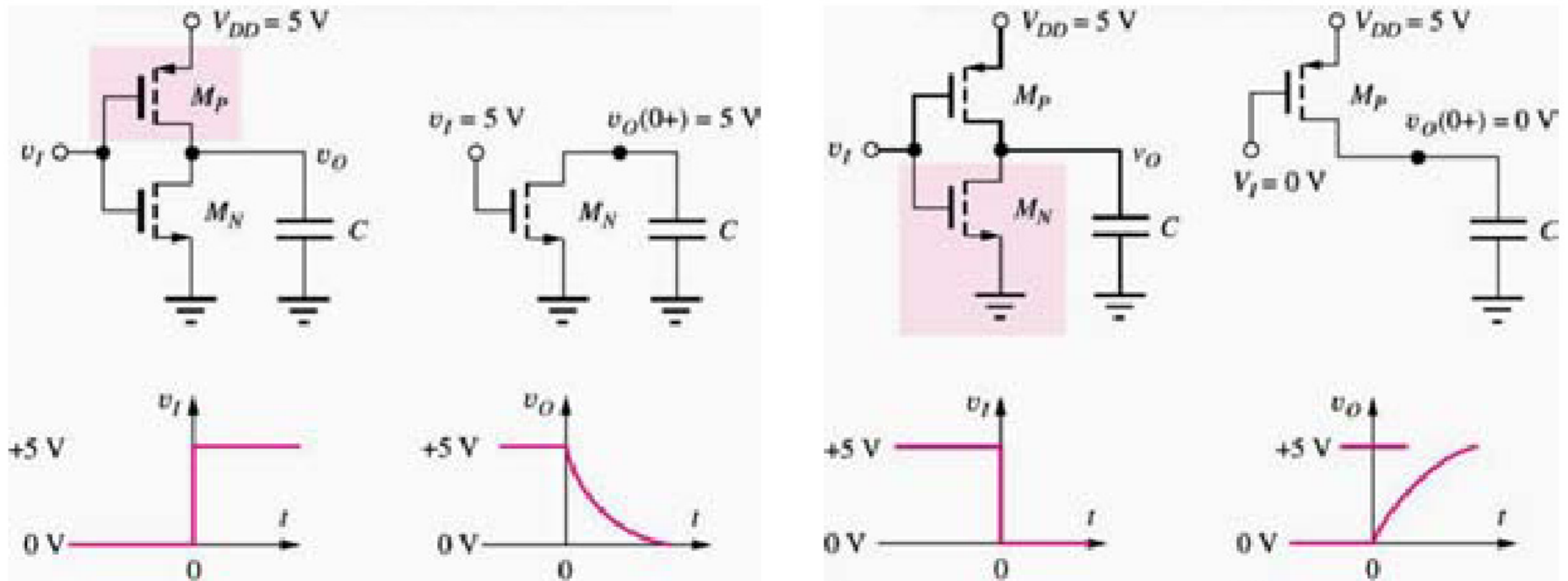
$$t_p = (t_{PLH} + t_{PHL}) / 2$$

The propagation delay time is directly proportional to the switching time and increases as the Fan-out increases.



# Estimation of propagation delay time

The propagation delay time is due to limitations in transistor switching speeds caused by **undesirable internal capacitive stored charges**.



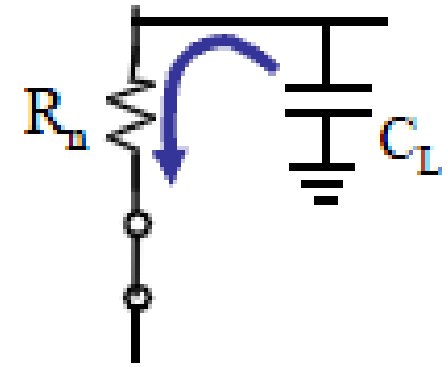
The two modes of capacitive charging/discharging that contribute to propagation delay

# Estimation of propagation delay time

To calculate delay, we consider transistor as a switch

$$I = C \frac{dV}{dt} \quad dt = C \frac{dV}{I} = RC \frac{dV}{V}$$

$$t_1 - t_0 = t_p = RC \int_{V_o}^{V_1} \frac{dV}{V} = RC \ln \left( \frac{V_1}{V_o} \right)$$



For fall delay  $t_{\text{pHL}}$ ,  $V_0 = V_{\text{cc}}$ ,  $V_1 = V_{\text{cc}}/2$

$$t_{\text{pHL}} = 0.69 R_n C_L$$

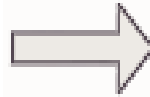
$$t_{\text{pLH}} = 0.69 R_p C_L$$

Total propagation delay time:  $t_p = (t_{\text{PLH}} + t_{\text{PHL}}) / 2$

# Evolution of TTL Logic Family

## TTL 74 Series

Standard TTL logic – saturated BJT, Obsolete now, Don't use in new designs



## TTL 74H Series

High speed TTL logic – decrease the resistance to lower the internal time constant but increase in  $P_{dis}$ . Typical  $P_{dis}$  = 22mW and  $t$  = 6ns



## TTL 74S Series

Schottky TTL logic – Deep saturation prevented by BC Schottky Diode. Reduced storage time delay. Practically obsolete. Typical  $P_{dis}$  = 20mW and  $t$  = 3ns



## TTL 74L Series

Low Power TTL logic – Increase the resistance to lower the  $P_{dis}$  but increase in internal time constant. Typical  $P_{dis}$  = 1mW and  $t$  = 35ns



## TTL 74LS Series

Low power Schottky TTL. Typical  $P_{dis}$  = 2mW and  $t$  = 10ns



## TTL 74ALS Series

Low power, high speed Schottky TTL logic-Innovations in IC design and fabrication. Improvement in speed and power dissipation. Popular.

