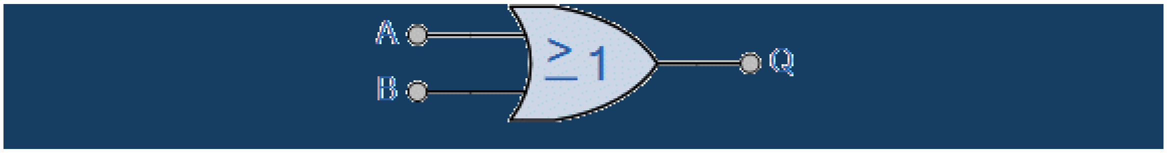


Logic OR Gate Tutorial

The Logic OR Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when one or more of its inputs are HIGH

The output, Q of a “Logic OR Gate” only returns “LOW” again when **ALL** of its inputs are at a logic level “0”. In other words for a logic OR gate, any “HIGH” input will give a “HIGH”, logic level “1” output.

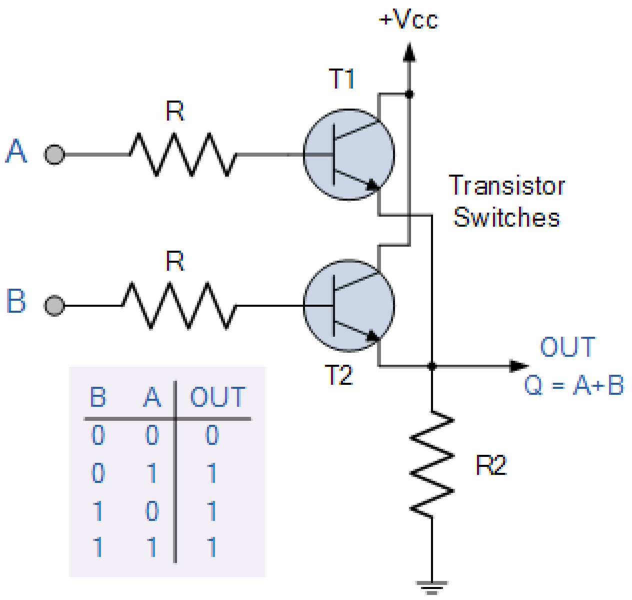
The logic or Boolean expression given for a digital logic OR gate is that for *Logical Addition* which is denoted by a plus sign, (+) giving us the Boolean expression of: A+B = Q.

Thus a logic OR gate can be correctly described as an “Inclusive OR gate” because the output is true when both of its inputs are true (HIGH). Then we can define the operation of a 2-input logic OR gate as being:

“If either A or B is true, then Q is true”

2-input Transistor OR Gate

A simple 2-input inclusive OR gate can be constructed using RTL Resistor-transistor switches connected together as shown below with the inputs connected directly to the transistor bases. Either transistor must be saturated “ON” for an output at Q.

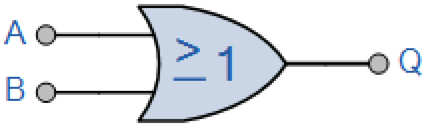


B	A	OUT
0	0	0
0	1	1
1	0	1
1	1	1

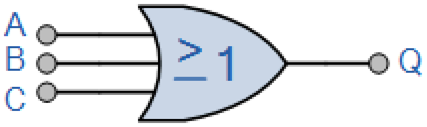
Logic OR Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape represents the logical operation of the OR gate.

Digital Logic "OR" Gate Types

The 2-input Logic OR Gate

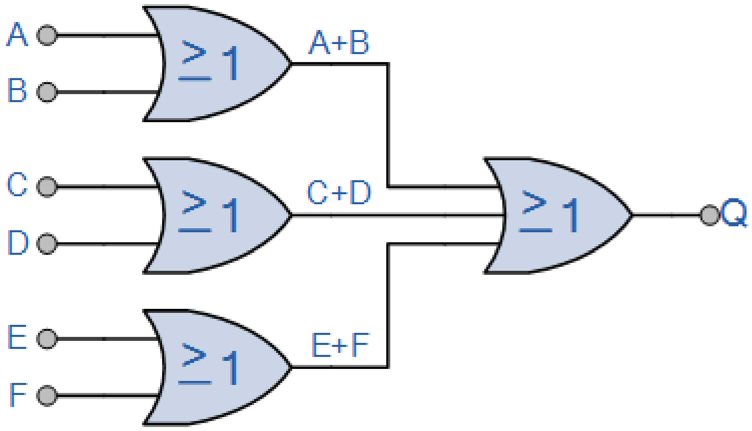
Symbol	Truth Table		
	B	A	Q
 2-input OR Gate	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression **Q = A+B**	Read as A **OR** B gives Q		

The 3-input Logic OR Gate

Symbol	Truth Table			
	C	B	A	Q
 3-input OR Gate	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	1
Boolean Expression **Q = A+B+C**	Read as A **OR** B **OR** C gives Q			

Like the AND gate, the OR function can have any number of individual inputs. However, commercial available OR gates are available in 2, 3, or 4 inputs types. Additional inputs will require gates to be cascaded together for example.

Multi-input OR Gate



6-input “OR” Function

The Boolean Expression for this 6-input OR gate will therefore be:

$$Q = (A+B)+(C+D)+(E+F)$$

In other words:

A OR B OR C OR D OR E OR F gives Q

If the number of inputs required is an odd number of inputs any "unused" inputs can be held LOW by connecting them directly to ground using suitable "Pull-down" resistors.

Commonly available digital logic OR gate IC's include:

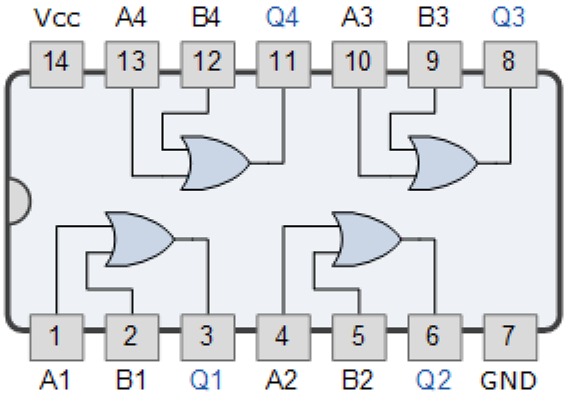
TTL Logic OR Gates

- 74LS32 Quad 2-input

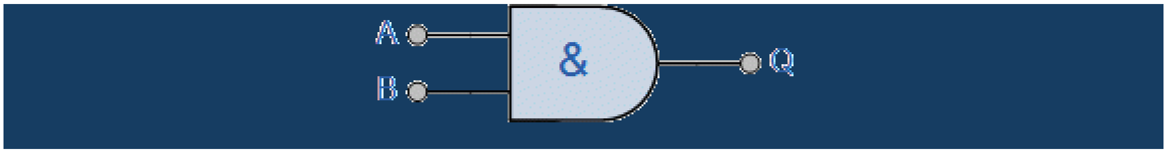
CMOS Logic OR Gates

- CD4071 Quad 2-input
- CD4075 Triple 3-input
- CD4072 Dual 4-input

7432 Quad 2-input Logic OR Gate



.



Logic AND Gate Tutorial

The Logic AND Gate is a type of digital logic circuit whose output goes HIGH to a logic level 1 only when all of its inputs are HIGH

–

The output state of a digital logic AND gate only returns "LOW" again when **ANY** of its inputs are at a logic level "0". In other words for a logic AND gate, any LOW input will give a LOW output.

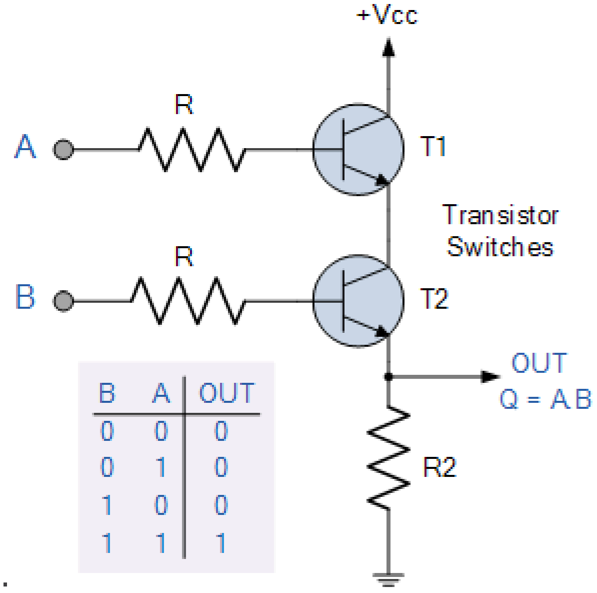
The logic or Boolean expression given for a digital logic AND gate is that for *Logical Multiplication* which is denoted by a single dot or full stop symbol, (**.**) giving us the Boolean expression of: A.B = Q.

Then we can define the operation of a digital 2-input logic AND gate as being:

"If both A and B are true, then Q is true"

2-input Transistor AND Gate

A simple 2-input logic AND gate can be constructed using RTL Resistor-transistor switches connected together as shown below with the inputs connected directly to the transistor bases. Both transistors must be saturated “ON” for an output at Q.

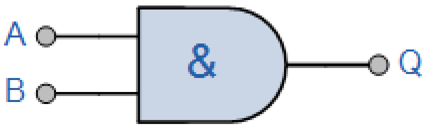


B	A	OUT
0	0	0
0	1	0
1	0	0
1	1	1

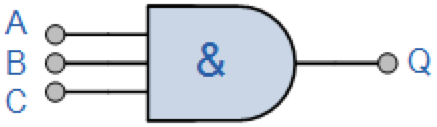
Logic AND Gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape represents the logical operation of the AND gate.

Digital Logic “AND” Gate Types

The 2-input Logic AND Gate

Symbol	Truth Table		
 2-input AND Gate	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression **Q = A.B**	Read as A **AND** B gives Q		

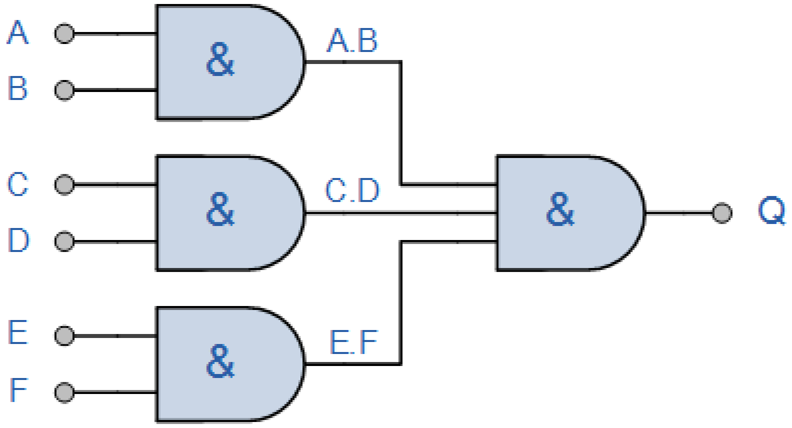
The 3-input Logic AND Gate

<table>
<tr><th>Symbol</th><th colspan="4">Truth Table</th></tr>
<tr><td rowspan="9">

3-input AND Gate</td><td>C</td><td>B</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression <b>Q = A.B.C</b></td><td colspan="4">Read as A <b>AND</b> B <b>AND</b> C gives Q</td></tr>
</table>

Because the Boolean expression for the logic AND function is defined as (**.**), which is a binary operation, AND gates can be cascaded together to form any number of individual inputs. However, commercial available AND gate IC’s are only available in standard 2, 3, or 4-input packages. If additional inputs are required, then standard AND gates will need to be cascaded together to obtain the required input value, for example.

Multi-input AND Gate



6-input "AND" Function

The Boolean Expression for this 6-input AND gate will therefore be:

$$Q = (A.B).(C.D).(E.F)$$

In other words:

A AND B AND C AND D AND E AND F gives Q

If the number of inputs required is an odd number of inputs any "unused" inputs can be held HIGH by connecting them directly to the power supply using suitable "Pull-up" resistors.

Commonly available digital logic AND gate IC's include:

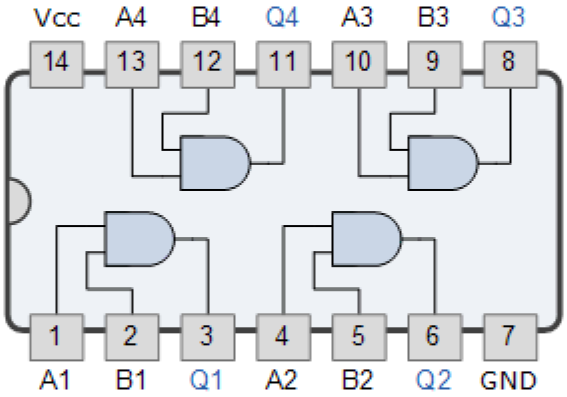
TTL Logic AND Gate

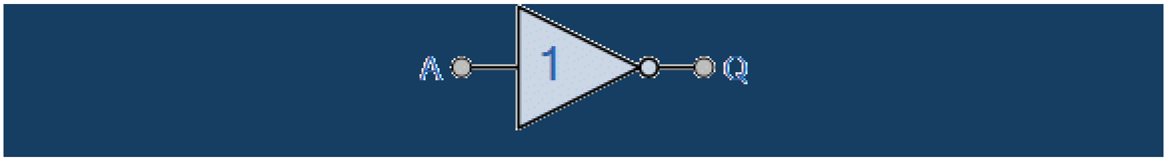
- 74LS08 Quad 2-input
- 74LS11 Triple 3-input
- 74LS21 Dual 4-input

CMOS Logic AND Gate

- CD4081 Quad 2-input
- CD4073 Triple 3-input
- CD4082 Dual 4-input

7408 Quad 2-input AND Gate





Logic NOT Gate Tutorial

The Logic NOT Gate is the most basic of all the logical gates and is often referred to as an Inverting Buffer or simply an Inverter

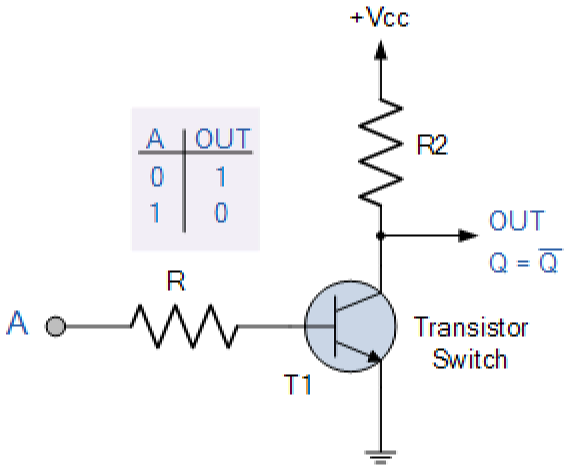
Inverting NOT gates are single input devicse which have an output level that is normally at logic level “1” and goes “LOW” to a logic level “0” when its single input is at logic level “1”, in other words it “inverts” (complements) its input signal. The output from a NOT gate only returns “HIGH” again when its input is at logic level “0” giving us the Boolean expression of: A = Q.

Then we can define the operation of a single input digital logic NOT gate as being:

“If A is NOT true, then Q is true”

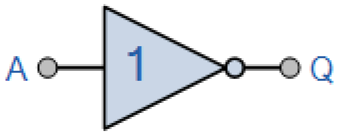
Transistor NOT Gate

A simple 2-input logic NOT gate can be constructed using a RTL Resistor-transistor switches as shown below with the input connected directly to the transistor base. The transistor must be saturated “ON” for an inverted output “OFF” at Q.



Logic NOT Gates are available using digital circuits to produce the desired logical function. The standard NOT gate is given a symbol whose shape is of a triangle pointing to the right with a circle at its end. This circle is known as an “inversion bubble” and is used in NOT, NAND and NOR symbols at their output to represent the logical operation of the NOT function. This bubble denotes a signal inversion (complementation) of the signal and can be present on either or both the output and/or the input terminals.

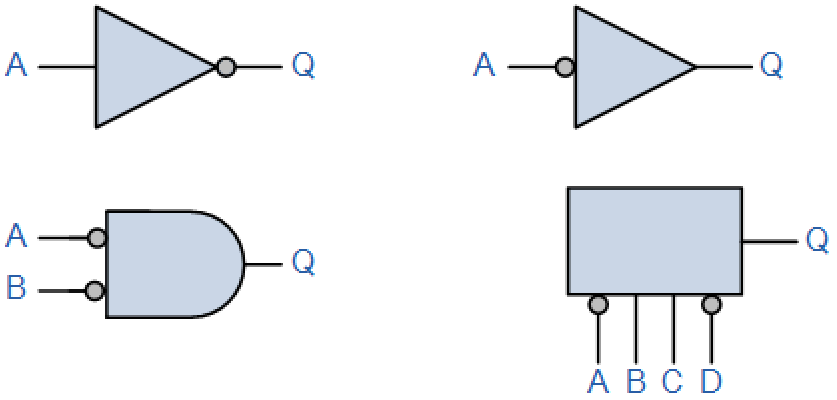
The Logic NOT Gate Truth Table

Symbol	Truth Table	
 Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression **Q = not A or A**	Read as inverse of **A** gives Q	

Logic NOT gates provide the complement of their input signal and are so called because when their input signal is “HIGH” their output state will **NOT** be “HIGH”. Likewise, when their input signal is “LOW” their output state will **NOT** be “LOW”. As they are single input devices, logic NOT gates are not normally classed as “decision” making devices or even as a gate, such as the AND or OR gates which have two or more logic inputs. Commercial available NOT gates IC’s are available in either 4 or 6 individual gates within a single IC package.

The “bubble” (o) present at the end of the NOT gate symbol above denotes a signal inversion (complementation) of the output signal. But this bubble can also be present at the gates input to indicate an *active-LOW* input. This inversion of the input signal is not restricted to the NOT gate only but can be used on any digital circuit or gate as shown with the operation of inversion being exactly the same whether on the input or output terminal. The easiest way is to think of the bubble as simply an inverter.

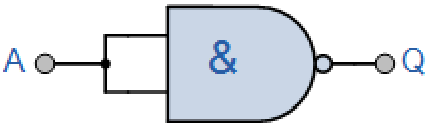
Signal Inversion using Active-low input Bubble



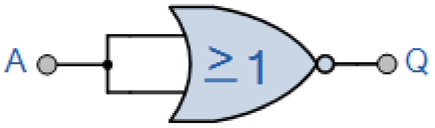
Bubble Notation for Input Inversion

NAND and NOR Gate Equivalents

An **Inverter** or logic NOT gate can also be made using standard NAND and NOR gates by connecting together **ALL** their inputs to a common input signal for example.

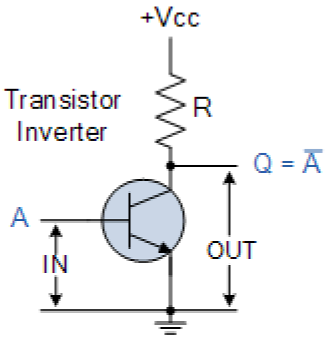


2-input NAND Gate Equivalent



2-input NOR Gate Equivalent

A very simple inverter can also be made using just a single stage transistor switching circuit as shown.



When the transistors base input at “A” is high, the transistor conducts and collector current flows producing a voltage drop across the resistor R thereby connecting the output point at “Q” to ground thus resulting in a zero voltage output at “Q“.

Likewise, when the transistors base input at “A” is low (0v), the transistor now switches “OFF” and no collector current flows through the resistor resulting in an output voltage at “Q” high at a value near to +Vcc.

Then, with an input voltage at “A” HIGH, the output at “Q” will be LOW and an input voltage at “A” LOW the resulting output voltage at “Q” is HIGH producing the complement or inversion of the input signal.

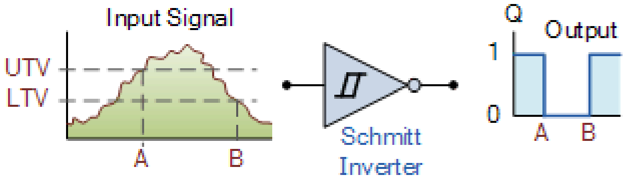
Hex Schmitt Inverters

A standard **Inverter** or **Logic NOT Gate**, is usually made up from transistor switching circuits that do not switch from one state to the next instantly, there will always be some delay in the switching action.

Also as a transistor is a basic current amplifier, it can also operate in a linear mode and any small variation to its input level will cause a variation to its output level or may even switch “ON” and “OFF” several times if there is any noise present in the circuit. One way to overcome these problems is to use a **Schmitt Inverter** or **Hex Inverter**.

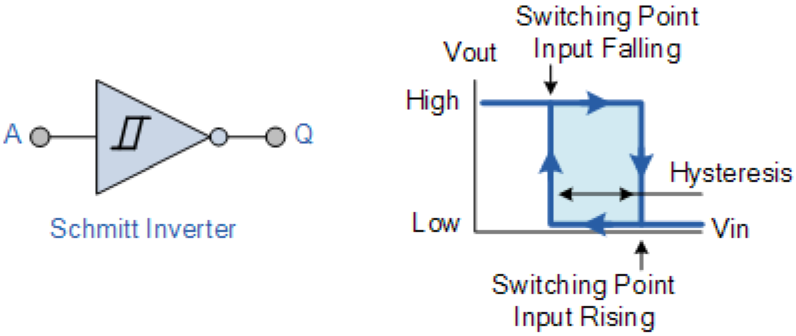
We know from the previous pages that all digital gates use only two logic voltage states and that these are generally referred to as **Logic “1”** and **Logic “0”** any TTL voltage input between 2.0v and 5v is recognised as a logic “1” and any voltage input below 0.8v is recognised as a logic “0” respectively.

A **Schmitt Inverter** is designed to operate or switch state when its input signal goes above an “Upper Threshold Voltage” or **UTV** limit in which case the output changes and goes “LOW”, and will remain in that state until the input signal falls below the “Lower Threshold Voltage” or **LTV** level in which case the output signal goes “HIGH”. In other words a Schmitt Inverter has some form of **Hysteresis** built into its switching circuit.



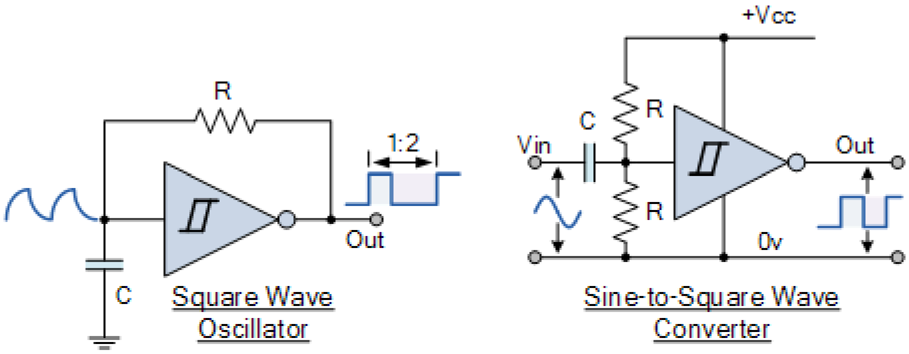
This switching action between an upper and lower threshold limit provides a much cleaner and faster “ON/OFF” switching output signal and makes the Schmitt inverter ideal for switching any slow-rising or slow-falling input signal and as such we can use a Schmitt trigger to convert these analogue signals into digital signals as shown.

Schmitt Inverter



A very useful application of Schmitt inverters is when they are used as oscillators or sine-to-square wave converters for use as square wave clock signals.

Schmitt NOT Gate Inverter Oscillator



The first circuit shows a very simple low power RC type oscillator using a Schmitt inverter to generate a square wave output waveform. Initially the capacitor C is fully discharged so the input to the inverter is “LOW” resulting in an inverted output which is “HIGH”. As the output from the inverter is fed back to its input and the capacitor via the resistor R the capacitor begins to charge up.

When the capacitors charging voltage reaches the upper threshold limit of the inverter, the inverter changes state, the output becomes “LOW” and the capacitor begins to discharge through the resistor until it reaches the lower threshold level were the inverter changes state again. This switching back and forth by the inverter produces a square wave output signal with a 33% duty cycle and whose frequency is given as: $ƒ = 680/RC$.

The second circuit converts a sine wave input (or any oscillating input for that matter) into a square wave output. The input to the inverter is connected to the junction of the potential divider network which is used to set the quiescent point of the circuit. The input capacitor blocks any DC component present in the input signal only allowing the sine wave signal to pass.

As this signal passes the upper and lower threshold points of the inverter the output also changes from “HIGH” to “LOW” and so on producing a square wave output waveform. This circuit produces an output pulse on the positive rising edge of the input waveform, but by connecting a second Schmitt inverter to the output of the first, the basic circuit can be modified to produce an output pulse on the negative falling edge of the input signal.

Commonly available logic NOT gate and Inverter IC’s include:

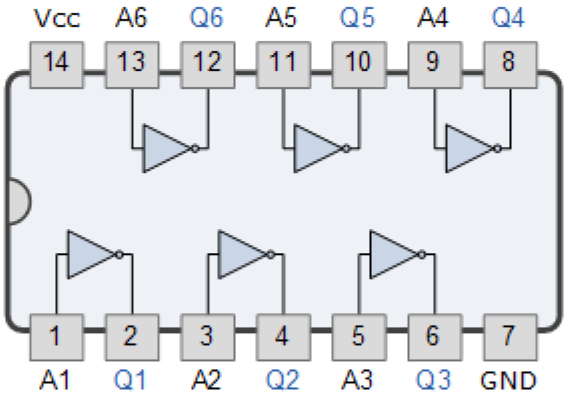
TTL Logic NOT Gates

- 74LS04 Hex Inverting NOT Gate
- 74LS14 Hex Schmitt Inverting NOT Gate
- 74LS1004 Hex Inverting Drivers

CMOS Logic NOT Gates

- CD4009 Hex Inverting NOT Gate
- CD4069 Hex Inverting NOT Gate

7404 NOT Gate or Inverter


Vcc
A6
Q6
A5
Q5
A4
Q4
14
13
12
11
10
9
8
1
2
3
4
5
6
7
A1
Q1
A2
Q2
A3
Q3
GND