Electronics Tutorials

LOGIC GATE FUNDAMENTALS

For Students, Professionals and Beyond

eBook 19

ASPENCORE

WWW.ELECTRONICS-TUTORIALS.WS

TABLE OF
CONTENTS

Our Terms of Use

This **Basic Electronics Tutorials eBook** is focused on the fundamentals of digital logic gates with the information presented within this ebook provided “as-is” for general information purposes only.

All the information and material published and presented herein including the text, graphics and images is the copyright or similar such rights of Aspencore. This represents in part or in whole the supporting website: **www.electronics-tutorials.ws**, unless otherwise expressly stated.

This free e-book is presented as general information and study reference guide for the education of its readers who wish to learn Electronics. While every effort and reasonable care has been taken with respect to the accuracy of the information given herein, the author makes no representations or warranties of any kind, expressed or implied, about the completeness, accuracy, omission of errors, reliability, or suitability with respect to the information or related graphics contained within this e-book for any purpose.

As such it is provided for personal use only and is not intended to address your particular problem or requirement. Any reliance you place on such information is therefore strictly at your own risk. We can not and do not offer any specific technical advice, troubleshooting assistance or solutions to your individual needs.

We hope you find this guide useful and enlightening. For more information about any of the topics covered herein please visit our online website at:

www.electronics-tutorials.ws

1. Digital Logic Fundamentals

Digital Electronics is a fundamental part of today's modern world and as such can be found in all kinds of applications, from mobile phones and portable devices, to TV's and music systems, to home and kitchen appliances, to cars and transportation. The list is endless with each device having some form of digital logic circuit or processor embedded within it performing a whole host of tasks.

Digital electronic circuits and systems process data by performing mathematical tasks and calculations based on their input conditions, or as data storage devices where digital logic gates are used. In large-scale digital systems, logic gates, arithmetic and memory storage, along with input, output devices or circuits constitutes a digital system.

Logic gates are devices in electronics that make up the fundamental building blocks of all digital circuits. They are manufactured as integrated circuits, commonly known as IC's for soldering onto circuit boards. Generally speaking, digital logic gates have one or more inputs along with one single output and perform a variety of Boolean operations.

While there may be only a few different Boolean algebra operations to be performed, digital logic circuits can repeat these switching operations many times over, and at a very fast speed.

> *Analogue signals represent physical properties with analogue circuits managing those signals*

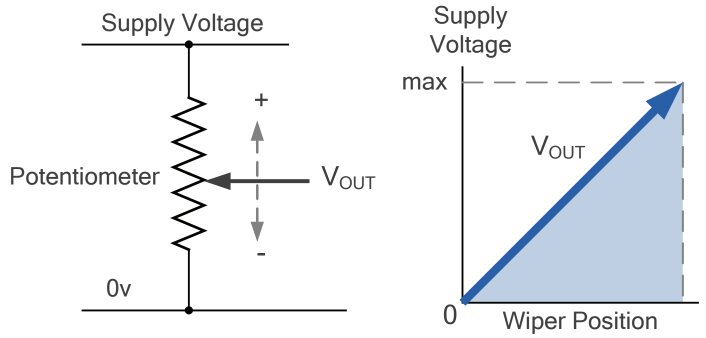
In electronics terms, an *analogue* device or system deals with one continuously variable voltage or current signal since they process physical quantities such as sound levels, light intensity, temperature, pressure, etc. Thus, all electrical circuits and systems can be viewed as a collection of analogue circuits.

Digital circuits however differ from analogue circuits in that they deal with only two, well-defined, but discrete voltage levels being **HIGH** or **LOW**, **ON** or **OFF**. That is one stable state or another. Sometimes it is required to convert these continuous analogue signals into a digital form using analogue-to-digital converters (ADC's), and vice versa, using digital-to-analogue converters (DAC's). Either way, analogue and digital circuits process their data and signals differently.

1.1 Analogue Signals

Analogue or linear circuits amplify or respond to continuously varying voltages or current levels which can alternate between a positive and negative value, or zero to some maximum value over a period of time as shown in Figure 1.

Figure 1. Analogue Voltage Representation



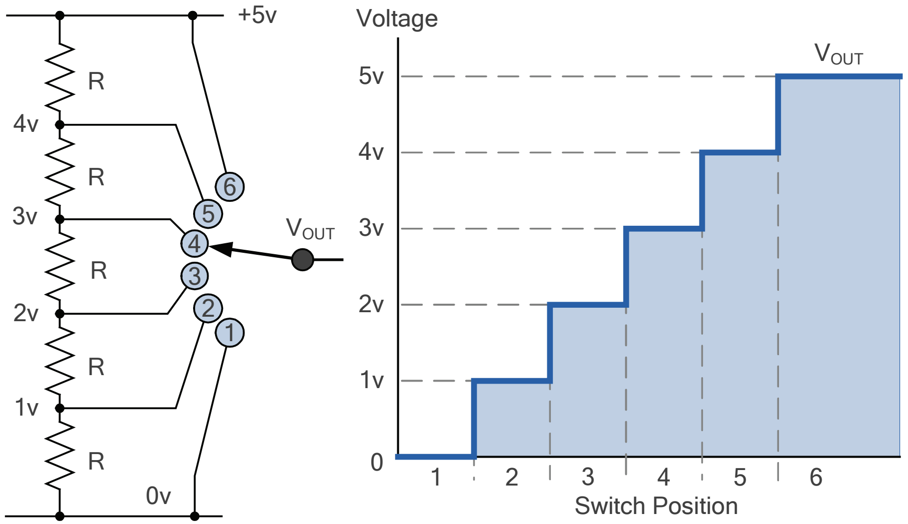
Here the output from the potentiometer (V_{OUT}) varies as the wiper terminal is rotated. This produces an infinite number of output voltage points between 0 volts and the maximum supply voltage (V_{MAX}).

This output voltage can be varied either slowly or rapidly from one value to the next. Thus there is no sudden or step change between any two voltage levels thereby producing a continuously variable output voltage. That is, analogue quantities are non-discrete with a limitless number of possibilities.

1.2 Digital Signals

Digital electronic circuits on the other hand only have two different voltage levels or states. These voltage states are generally referred to as Logic "1" and Logic "0", or HIGH (true) and LOW (false). For example, the "HIGH" and "LOW" states could be represented by 5V and 0V respectively. The exact value of these two voltage levels will depend on the technology being used such as TTL (transistor-transistor logic) or Advanced CMOS technology.

Figure 2. Digital Voltage Representation



In Figure 2. the potentiometer wiper has been replaced by a single 6-position rotary switch which is connected in turn to the junction of the series resistor chain. This forms a basic potential or voltage divider network were each resistor is of an equal value (R).

As the switch is rotated from one position (or node) to the next, the output voltage, V_{OUT} changes quickly in discrete and distinctive voltage levels or steps representing multiples of 1.0 volts on each switching action, as shown in the output graph of Figure 2.

So for example, the output voltage, V_{OUT} will be 2 volts at position 3, and 3 volts at position 4, and 5 volts at position 6, etc. Because of this switch action, V_{OUT} cannot be 2.5V, 3.1V or 4.6V, etc. Only steps of 1 volt.

Digital signals have only two discrete states with digital circuits managing those states

Finer output voltage levels could easily be produced by increasing the number of resistive elements within the potential divider network thereby increasing the number of discrete switching steps.

Then we can see that a digital output voltage can never be a continuous value but will change in discrete steps and for digital signals, only two logic levels are allowed.

Digital signals with only two voltage levels or steps are based on the binary or Base-2 numbering system. In Boolean Algebra the binary numbering system is used to describe these two voltage levels being represented by the binary digits of "1" and "0" respectively. But what do we mean by a binary "1" and a binary "0" signal.

2. Digital Logic Levels

All digital circuits handle input and output signals in which one of two possible voltage states are present at any one time. A logic-1 (**HIGH**) or a logic-0 (**LOW**). While precise DC input and output voltages are not absolutely necessary in digital electronics, each of these two voltage levels must fall within a specific DC voltage range to be classed as valid.

Digital logic gate circuits are constructed using transistors with the type of transistor used, Bipolar or MOSFET, determining the voltage characteristics of the circuit. For example, bipolar transistors create TTL (transistor-transistor logic) devices such as the 7400 series, with their supply voltage, V_{CC} being fixed at +5 volts. But lower voltages such as 3.3 V, 2.5 V, 1.8 volts are now available

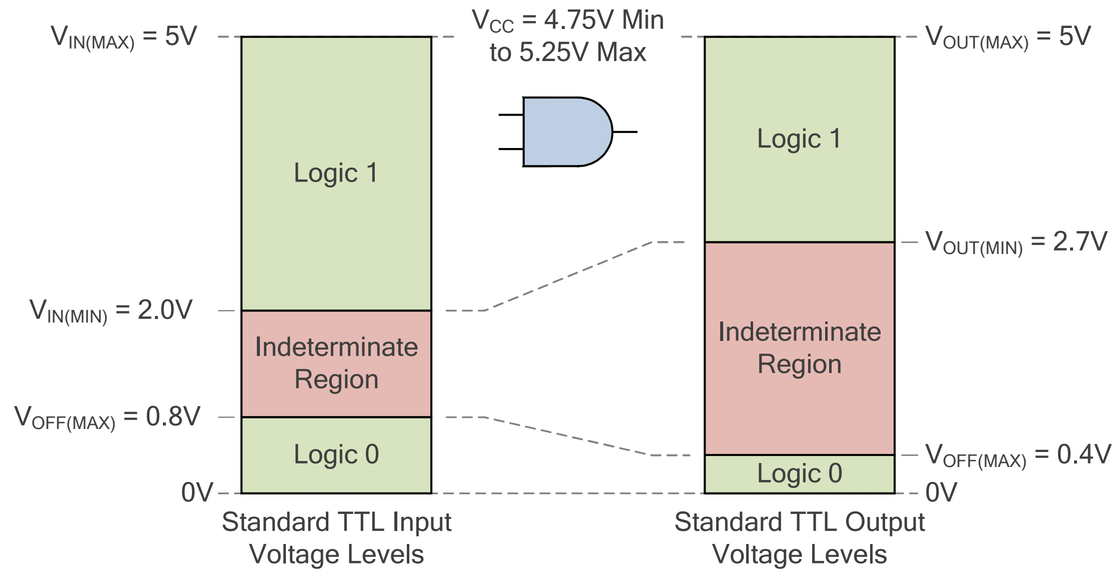
Standard Transistor-transistor logic, or TTL was the original logic gate circuit type

CMOS (complementary metal-oxide silicon) devices such as the 4000 series are constructed using complementary MOSFET or JFET type Field Effect Transistors for both their input and output circuitry so can use a supply voltage V_{DD} from between +3 to +18 volts DC. However, they commonly operate on +5 volts when used in the same circuit with TTL 74-series devices such as the HC or AC High Speed CMOS sub-family.

Digital logic gates are commonly referred to as two-state switching circuits because they switch between **HIGH** and **LOW** voltages, or **ON** and **OFF** representing binary digits of "**1**" or "**0**" respectively.

In standard TTL (transistor-transistor logic) IC's there is a pre-defined voltage range for the input and output voltage levels which defines exactly what is a logic "1" level, and what is a logic "0" level. These valid input voltage levels and output voltage levels of a digital logic gate are shown in Figure 3.

FIGURE 3. DIGITAL VOLTAGE LEVELS



Thus a logic "0" input is defined as any voltage level in the range from 0 to 0.8 volts ($V_{OFF(MAX)}$). While a logic "1" input is defined as a voltage level between 2 volts ($V_{IN(MIN)}$) and 5 volts (V_{CC}) for standard TTL. Likewise, for the output, 0 to 0.4 volts ($V_{OFF(MAX)}$) constitutes a "0" (low) level output. While 2.7 ($V_{OUT(MIN)}$) to 5 volts (V_{CC}) constitutes a "1" (high) level output.

Semiconductor manufacturers publish data sheets for each of their digital logic devices defining the recommended and absolute maximum voltage ranges for the particular sub-family used. Their data sheets commonly use the following notation to define the minimum and maximum input and output voltage levels.

V_{OH}(MIN) – Minimum Output (O) voltage for a High (H) level condition

V_{OL}(MAX) – Maximum Output (O) voltage for a Low (L) level condition

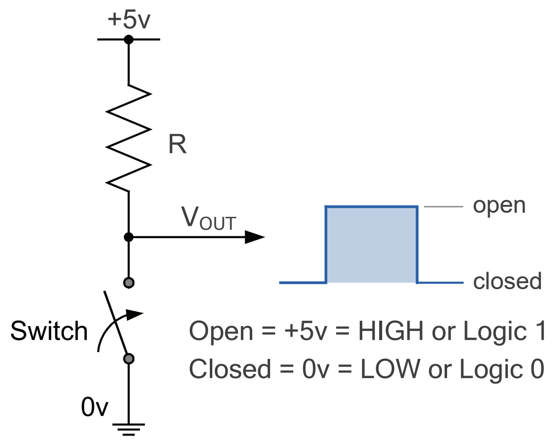
V_{IH}(MIN) – Minimum Input (I) voltage for a High (H) level condition

V_{IL}(MAX) – Maximum Input (I) voltage for a Low (L) level condition

Any input or output voltage level which does not fall within either of a logic devices ranges are said to be **indeterminate** and are therefore not valid logic levels. Floating or unconnected inputs and outputs can also produce false voltage levels if not correctly terminated.

We can also show the two-state switching action between logic "1" and logic "0" using a simple switch circuit of Figure 4.

FIGURE 4. DIGITAL VOLTAGE LEVELS



Here a resistor and single-pole switch are connected together in series across a +5 volt DC supply.

When the switch is open, 5 volts, or a logic "1" is given as the output signal. When the switch is closed the output is grounded and 0v, or a logic "0" is given as the output.

Then depending upon the position of the switch, a "HIGH" or a "LOW" output is produced.

Digital logic gates are the most basic of digital electronic devices and which can be used to produce an endless number of combinational and sequential logic circuits for use in numerous digital circuits.

They are called **Logic Gates** because their output condition is *logical* and based on the mathematical logic operation it must perform. The term *gate* is used as like the lock on a gate, only certain input conditions will unlock or open the gate to produce the required output. Usually defined as a logic "1" or HIGH output state.

The two most common and easily understood digital logic gates are the **2-input AND gate** and the **2-input OR gate**.

3. Digital Logic AND Gate

The **Logic AND Gate** is a type of digital logic circuit whose output condition can be mathematically defined as the product of two (or more) inputs, which we will denote here as inputs **A** and **B** for reference. Thus the **AND** gate symbol contains two individual inputs (**A**, **B**) and one single output (**Q**).

For the logic **AND** function to hold true, two or more events must occur together and at the same time for an output action to occur. The order in which these actions occur is unimportant as it does not affect the final result.

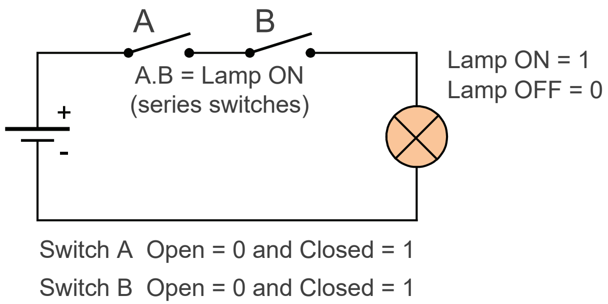
The output expression given for an **AND** gate is that for *logical multiplication* which is denoted by a single dot or full-stop symbol, (**.**). This then gives us the Boolean expression of: **A.B = Q** or **B.A = Q**. Note that we can also omit the dot and simply use AB or BA.

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

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

We can prove this AND gate statement using a two switch analogy as shown in Figure 5.

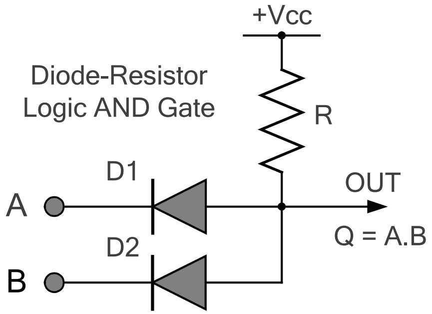
Figure 5. Switch Representation of the AND Function



The two switches, **A** and **B** are connected together in a series combination. Therefore, both switch **A AND** switch **B** must be closed (Logic-1) in order for the lamp to be ON. In other words, both switches must be closed together, or at logic "1" for the lamp to be ON.

A simple 2-input AND gate can be constructed from discrete components to form diode resistor logic, (DRL) as switching diodes themselves can be used to build logic gates for simple applications as shown in Figure 6.

Figure 6. 2-input Diode-Resistor AND Gate

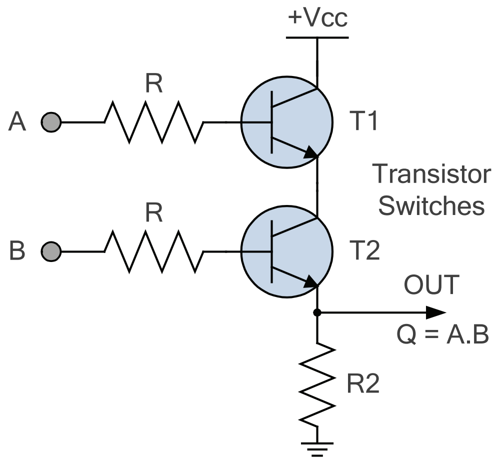


Assuming that the two diodes are ideal and the input voltages V_A and V_B are both positive logic inputs. If, either input **A** or input **B** is held LOW, the output is LOW as the diode is forward-biased. The output can only go HIGH again when both inputs go HIGH producing the required **AND** gate action.

However, for real diodes, each diode will drop 0.7 V across its pn-junction when forward-biased, resulting in 0.7 V for a LOW and 5 V for a HIGH at the output.

We can improve on this slightly and still construct an **AND** gate switch using discrete components by replacing the diodes with transistors. This create a Resistor-Transistor logic (RTL) gate as shown in Figure 7.

Figure 7. 2-input Resistor-Transistor AND Gate

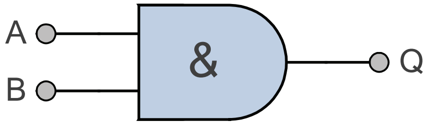


This simple 2-input **AND** gate constructed using resistor-transistor (RTL) switches has the two inputs connected directly to the transistor bases. Thus both transistors must be saturated "ON" for an output at **Q**.

Although discrete logic gate circuits are easy to understand, you wouldn't use them in practice as they have too many disadvantages. Today there are a full range of excellent logic gate IC's available to make digital circuits easier and cheaper to construct.

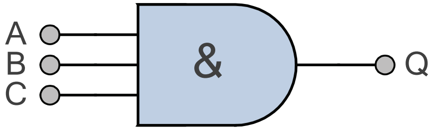
Logic AND gates are available using digital circuits to produce the desired logical function and is given a symbol whose shape represents the logical operation of the **AND** gate.

Table 1. A 2-input Logic AND Gate and Truth Table

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

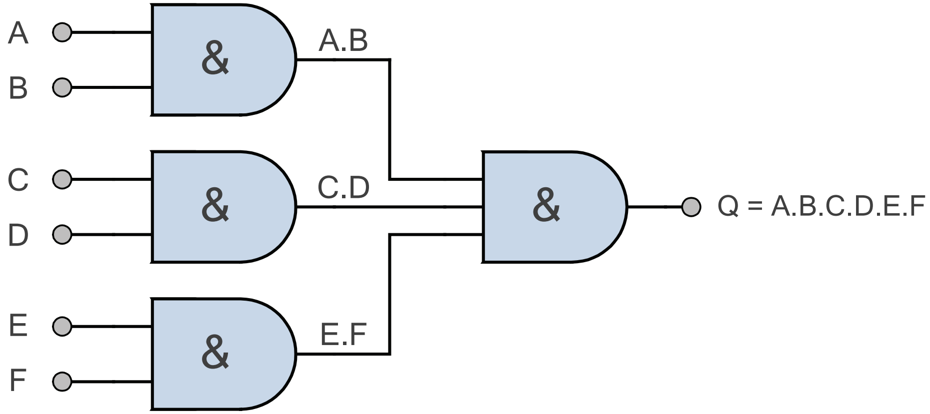
A standard 3-input digital logic **AND** gate and truth table is given in Table 2. as:

Table 2. A 3-input Logic AND Gate and Truth Table

Symbol	Truth Table			
	C	B	A	Q
	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	0
3-input AND Gate Symbol	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	**1**
Boolean Expression Q = A.B.C	A AND B AND C gives Q			

Commercially available **AND** gate IC's are commonly available in standard 2, 3, or 4-input packages. But **AND** gates can also be cascaded together to form a circuit with any number of individual inputs as demonstrated by the 6-input **AND** circuit of Figure 8.

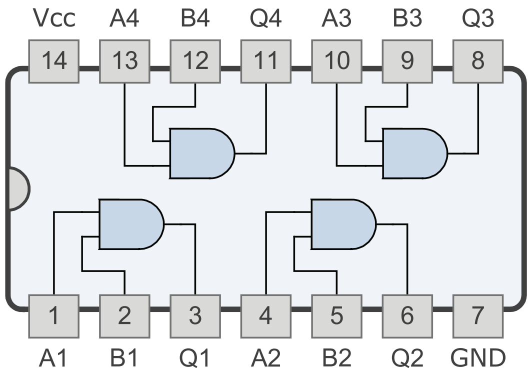
Figure 8. 6-input Digital Logic AND Gate



Thus the Boolean Expression for this 6-input **AND** gate will therefore be: **Q = A.B.C.D.E.F**

Digital logic **AND** gates are available as standard IC packages such as the common TTL **74LS08** Quadruple 2-input Positive **AND** Gate as shown in Figure 9. The TTL **74LS11** Triple 3-input Positive **AND** Gates, or the **74LS21** Dual 4-input Positive **AND** Gate.

Figure 9. Internal Arrangement of the 74LS08 Quad 2-input AND Gate IC



4. Digital Logic OR Gate

The **Logic OR Gate** is a digital logic circuit whose output condition can be mathematically defined as the sum of two (or more) inputs, which again we will denote here as inputs **A** and **B** and one single output (**Q**) for reference.

For the logic **OR** function, two or more events can occur at different times for an output action to occur. As before, the order in which these two actions occur is unimportant as it does not affect the output.

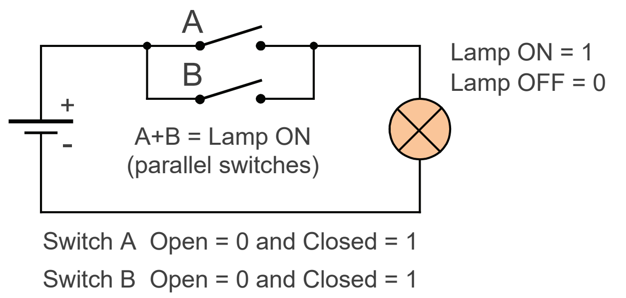
The output expression given for an **OR** gate is that for *logical addition* which is denoted by a plus sign, (**+**). Thus a 2-input (**A B**) logic **OR** Gate has an output term represented by the Boolean expression of: A+B = Q or B+A = Q.

Then we can define the operation of a simple 2-input **OR** gate as being:

"If either A and B is true, then Q is true"

We can prove this **OR** gate statement using a two switch analogy as shown in Figure 10.

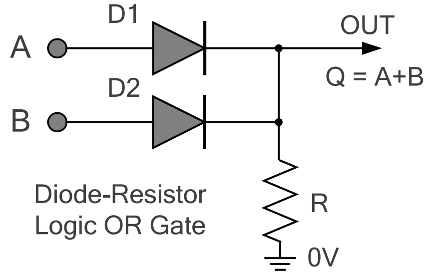
Figure 10. Switch Representation of an OR Function



The two switches, **A** and **B** are connected together in a parallel combination. Therefore, either switch **A OR** switch **B** can be closed (Logic-1) for the lamp (output) to be ON. In other words, either switch can be closed at any time for the lamp to be "ON".

As before, a simple 2-input **OR** gate can be constructed from discrete components to form diode resistor logic, (DRL) using positive logic since switching diodes can be used to build logic gates as shown in Figure 11.

Figure 11. 2-input Diode-Resistor OR Gate

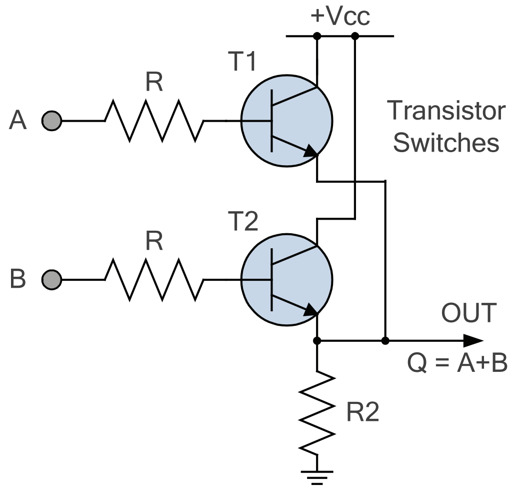


Again assume the diodes are ideal and the input voltages V_A and V_B are both positive logic inputs. If input **A** or input **B** is held LOW, the output is LOW. The output can only go HIGH only when either one of the inputs goes HIGH forward-biasing the diode producing the required **OR** gate action.

The diodes forward-biased 0.7V voltage drop 0.7 V results in a 0 V for a LOW and 4.3 V for a HIGH at the output.

We can also improve on this diode circuit by replacing the diodes with transistors to create a Resistor-Transistor logic (RTL) digital **OR** gate as shown in Figure 12.

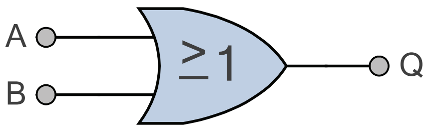
Figure 12. 2-input Resistor-Transistor OR Gate



This simple 2-input **OR** gate constructed using resistor-transistor (RTL) switches has the two inputs connected directly to the transistor bases. Thus either parallel connected transistor must be saturated fully "ON" to connect the output to $+V_{CC}$ and produce an output at **Q**.

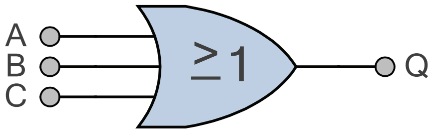
2-input *logic OR gates* are also available as standard digital circuits to produce the desired logical function. The symbols for the logical operation of the **OR** gate is that of a curved shape with the **≥1** indicating that the output is activate with any one active input.

Table 3. A 2-input Logic OR Gate and Truth Table

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

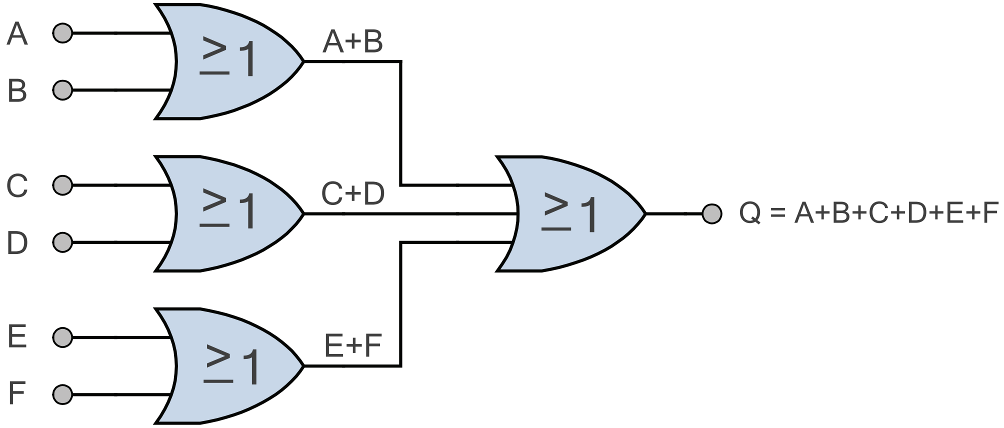
The standard 3-input digital logic **OR** gate and truth table is given in Table 4. as:

Table 4. A 3-input Logic OR Gate and Truth Table

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

Commercially available **OR** gate IC's are commonly available in standard 2, 3, or 4-input packages. But **OR** gates can also be cascaded together to form any number of individual inputs to obtain the required input value as shown in Figure 13.

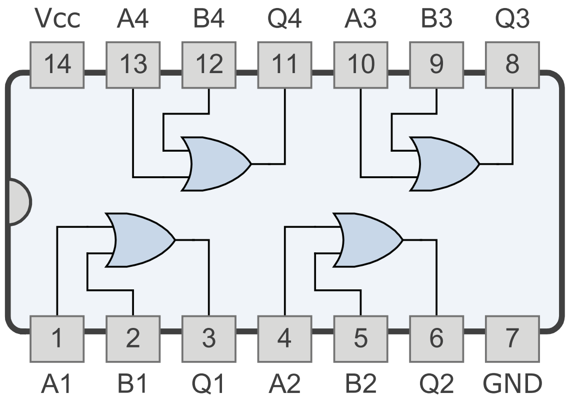
Figure 13. 6-input Digital Logic OR Gate



The Boolean Expression for this 6-input **OR** gate will therefore be: Q = A+B+C+D+E+F

Digital logic **OR** gates are available as standard IC packages such as the common TTL **74LS32** Quadruple 2-input Positive **OR** Gate.

Figure 14. Internal Arrangement of the 7432 Quad 2-input OR Gate IC



5. Digital Logic NOT Gate

The **Logic NOT Gate** is simply a single input inverter that changes the input of a logic level "1" to an output of logic level "0" and vice versa. That is the **NOT** gate "inverts" or complements its input signal.

The **NOT** gate is so called because its output state is NOT the same as its input state with its operation generally denoted by a bar or over-line (¯) above its input symbol. This over-line always denotes an inversion operation.

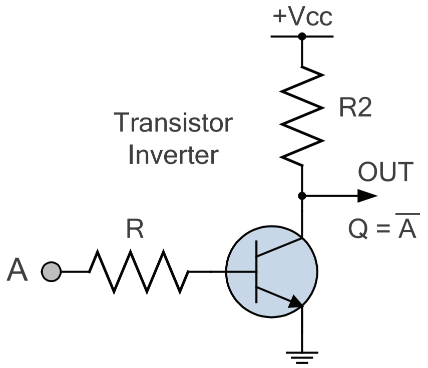
As **NOT** gates perform the logic function of *Inversion* or *Complementation* they are more commonly known as **Inverters** because they invert their input signal giving us the Boolean expression of: $Q = \overline{A}$. That is Q = not-A.

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"

A simple single input logic NOT gate can be constructed using a RTL Resistor-transistor switch as shown below with the input connected directly to the transistors base. The transistor must be fully-ON, or fully-OFF for the inverted output to be present at Q.

Figure 15. Resistor-Transistor Logic NOT Gate



The inverter of Figure 15 is constructed using just a single stage transistor switching circuit

When the transistors base input at **A** is HIGH, the transistor conducts and collector current flows producing a voltage drop across the resistor **R2**. This connects the output at **Q** to ground resulting in an inverted LOW output.

Likewise, when the transistors base input at **A** is LOW (0v), the transistor switches "OFF" so the output voltage at **Q** becomes HIGH at a value near to $+V_{CC}$.

Then, with an input voltage at **A** HIGH, the output at **Q** will be LOW and an input voltage at **A** which is LOW results in a HIGH output voltage at **Q**. Thus producing the complement or inversion of the input signal.

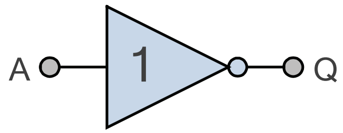
Then we can correctly say that if Q = not-A, ($\overline{A}$) and the input is at logic-0, then the output condition would be at a logic-1 as: $Q = \overline{A}$ = not-0 = 1.

The single input digital NOT gate is the most basic of logic gates performing Inversion

Likewise, if input **A** of the **NOT** gate is at a logic-1 level, then the output would be at logic-0 since: $Q = \overline{A}$ = not-1 = 0.

Logic **NOT** Gates are available as digital IC's such as the 7404 which can 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 as shown in Table 5.

Table 5. The Logic NOT Gate and Truth Table

Symbol	Truth Table	
	A	Q
 Inverter or NOT Gate Symbol	0	1
	1	0
Boolean Expression $Q = \overline{A}$ (not-A)	Inverse of A give Q	

This circle is known as an "*inversion bubble*" and is used to represent the logical operation of the **NOT** function. As the bubble represents a signal inversion or complementation, it can be present on either the output and/or the input terminals.

Commonly available digital logic **NOT** gate and Inverter IC's includes the TTL **74LS04** Hex Inverting **NOT** Gate, or the **74LS14** Hex Schmitt Inverting **NOT** Gate.

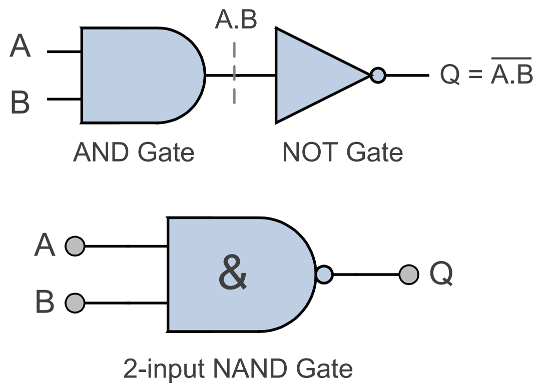
6. Digital Logic NAND Gate

If we take two separate logical gates, the AND gate and the NOT gate and connect them in series, we can produce another type of digital logic gate called a NAND (Not-AND) gate.

The **Digital Logic NAND Gate** is the reverse or "Complementary" form of the AND gate we looked at previously in section 3 with the Boolean expression given for that as *Logical Addition*, since it performs the inversion of its inputs.

The Boolean expression of the gate is denoted by the full stop symbol, (.) with a line (¯) above the output expression to signify the NOT or inversion function of the NAND gate. Then the output is expressed by the Boolean expression of: $\overline{A.B}$ as shown in Figure 16.

Figure 16. Digital NAND Gate Equivalence



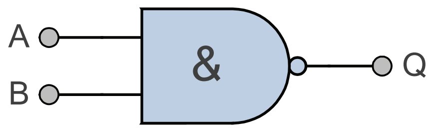
As before, the logic NAND gates is also available as a digital circuit. While the NAND gate has no real circuit symbol of its own, it uses the same shape as that for the standard AND gate but with the addition of circle (o) on its output.

This circle, sometimes called an "inversion bubble" at its output represents the NOT gate to indicate a negation (inversion) or complementary output.

> *The NAND gate functions the same as the AND gate except the output is inverted*

The truth table of Table 6. for the NAND function is the opposite of that for the previous AND gate since the NAND gate is the complement of the basic AND gate.

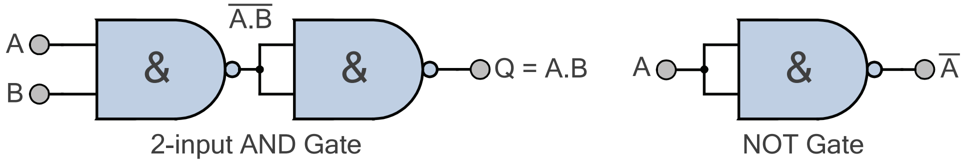
Table 6. A 2-input Logic NAND Gate and Truth Table

Symbol	Truth Table		
	B	A	Q
 2-input NAND Gate Symbol	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A.B}$	A AND B gives not-Q		

As with the AND gate seen previously, the NAND gate can also have any number of individual inputs. Commercial available NAND Gate IC's such as the TTL **74LS00**, or the **74LS10** are available in standard 2, 3, or 4 input types. If additional inputs are required, then the standard NAND gates can be cascaded together to provide more inputs. For example, the Boolean expression given for a 6-input NAND gate circuit would therefore be: $Q = \overline{A.B.C.D.E.F}$ or simply $Q = \overline{ABCDEF}$.

The NAND Gate is generally classed as a "Universal" logic gate type since they can be used on their own to produce other types of logic gate functions. By connecting them together in various combinations the basic gate types of AND and NOT function can be formed as shown in Figure 17.

Figure 17. Digital Logic Gates Using the NAND Gate



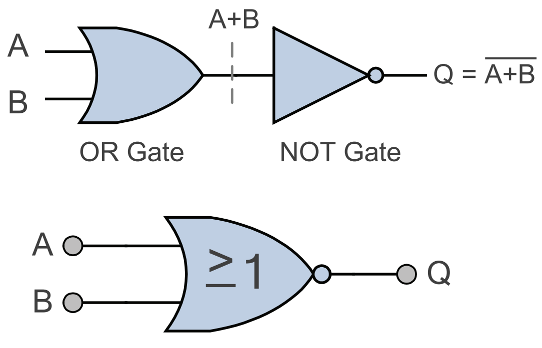
7. Digital Logic NOR Gate

Like the previous NAND gate, the **Digital Logic NOR Gate** is also a combination of two logic functions, the NOT gate and the OR gate (Not-OR) connected together in series.

The NOR gate is the reverse or "Complementary" form of the OR gate we looked at previously in Section 4, with the Boolean expression given as *Logical Multiplication*, since it performs the output inversion of its inputs.

The Boolean expression of the gate is denoted by the full stop symbol, (+) with a line (¯) above the output expression to signify the NOT or inversion function of the NOR gate. Then the output is expressed by the Boolean expression of: $\overline{A+B}$ as shown in Figure 18.

Figure 18. Digital NOR Gate Equivalence

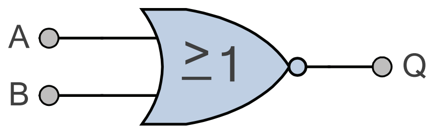


The NOR gates is also available as a digital circuit and uses the same shape as that for the standard OR gate but with the addition of an inversion bubble at its output to represent the negation (inversion) or complementary of its output.

The truth table of Table 7. for the NOR function is the opposite of that for the previous OR gate since the NOR gate is the complement of the basic OR gate.

> *The NOR gate functions the same as the OR gate except the output is inverted*

Table 7. A 2-input Logic NOR Gate and Truth Table

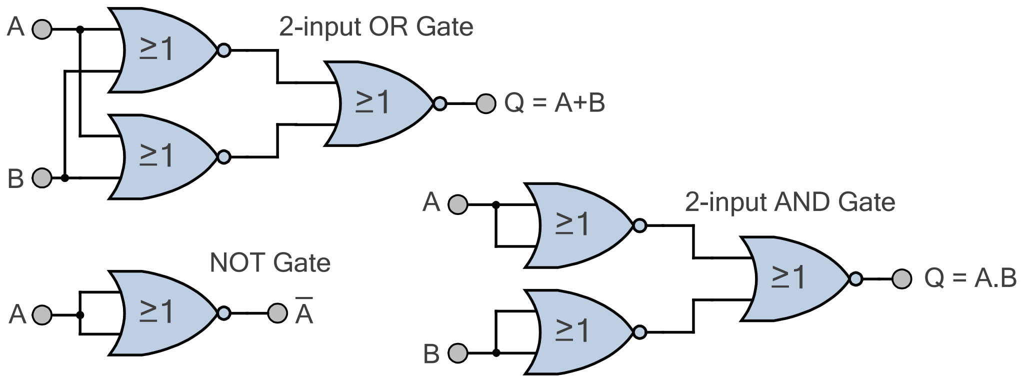
Symbol	Truth Table		
	B	A	Q
	0	0	1
	0	1	0
	1	0	0
2-input NOR Gate Symbol	1	1	0
Boolean Expression $Q = \overline{A+B}$	A OR B gives not-Q		

As with the OR gate, the NOR gate function can also have any number of individual inputs. Commercial available NOR Gate IC's such as the TTL **74LS02** Quad 2-input, the **74LS27** Triple 3-input, and the **74LS260** Dual 5-input are available as standard types.

If additional inputs are required, then the standard NOR gates can be cascaded together to provide more inputs. For example, the Boolean expression given for a 6-input NOR gate circuit would therefore be: $Q = \overline{A+B+C+D+E+F}$.

Like the NAND Gate, the NOR gate function is also classed as a "Universal" logic gate as they can be used to produce other types of logic gate functions as shown in Figure 19.

Figure 19. Digital Logic Gates Using the NOR Gate



8. The Exclusive OR Gate

While the **Exclusive-OR (Ex-OR) Gate** is not a basic logic gate in its own right, as it can be constructed by combining some of the previously discussed logic gates. But their use in arithmetic and computer logic circuits to process various binary numbers is important enough for it to be considered as complete logic gate on its own.

In Section 4 we saw that for a 2-input **OR** gate, if **A = 1**, **OR B = 1**, **OR BOTH A + B = 1** then the output from the digital gate must also be at a logic-1. Thus this type of digital logic gate is known as an *Inclusive-OR Gate*. That is, it gets its inclusive-OR name from the fact that it includes the case of **Q = 1** when both **A** and **B = 1**.

If a logic-1 output is obtained when ONLY **A = 1** or when ONLY **B = 1** but NOT both together at the same time, giving the binary inputs of **01** or **10**. Thus the output will be **1** producing an **Exclusive-OR** function, also written as: **Ex-OR** or **XOR**, for short.

> *The Exclusive-OR gate produces a HIGH output only when there is an odd number of HIGH inputs*

This is because its Boolean expression excludes the "**OR BOTH**" condition of **Q = 1** when both **A = 1** and **B = 1**. In other words the output of an Exclusive-OR gate ONLY goes HIGH when its two input terminals are at "**DIFFERENT**" logic levels with respect to each other.

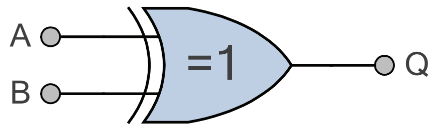
Therefore, an Exclusive-OR gate will only give an output value of logic-1 when there are an **ODD** number of 1's (HIGH's) on the inputs. If the two inputs are equal, the output is **0**.

The symbol used to denote an Exclusive-OR odd function is slightly different to that for the standard Inclusive-OR Gate. The logic or Boolean expression given for a logic OR gate is that of logical addition which is denoted by a standard plus sign.

The symbol used to describe the Boolean expression for an Exclusive-OR function is a plus sign, (**+**) within a circle (**O**) with the resulting symbol being: (⊕). This exclusive-OR symbol (⊕) represents the mathematical "direct sum of sub-objects" expression.

The truth table, logic symbol and implementation of a 2-input Exclusive-OR gate is given in Table 8.

Table 8. A 2-input Exclusive-OR Gate and Truth Table

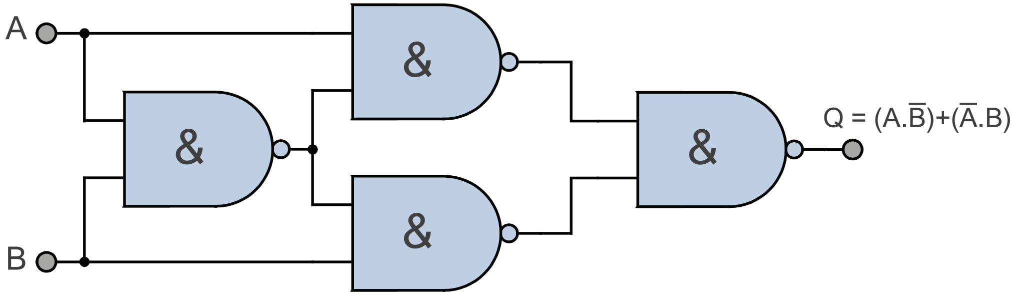
<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">

2-input Ex-OR Gate Symbol</td><td>B</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression Q = A⊕B</td><td colspan="3">A OR B but NOT BOTH gives Q</td></tr>
</table>

Note that the output is only **1** when there is an odd number of binary **1** inputs. That is only an ODD number of logic-1's on its inputs will produce a logic-1 at the output. This gives us the Boolean expression of: $Q = A.\overline{B} + \overline{A}.B = A \oplus B$

The *Exclusive-OR* gate is available as a standard TTL **74LS86** Quad 2-input gate. However, we can also construct an Ex-OR gate using a combination of **NOT**, **AND**, and **OR** gates but is much easier using our old favourite, the **NAND** gate as shown in Figure 20.

Figure 20. Exclusive-OR Gate Using Universal NAND Gates



9. The Exclusive NOR Gate

Basically, the **Exclusive-NOR (Ex-NOR) Gate** is a combination of the previous *Exclusive-OR gate* with an inverting **NOT** gate on its output. The truth table for the Exclusive-NOR gate is similar to the standard **NOR** gate in that it has an output which is at logic-1 and goes LOW to logic-0 when ANY of its inputs are at logic-1.

However, the output of a digital logic *Exclusive-NOR* gate ONLY goes HIGH when its two inputs, **A** and **B** are at the **SAME** logic level which can be either at logic-1 or at logic -0. The inverse of the Exclusive-OR gate. In other words, an even number of 1's or 0's on its inputs gives a logic 1 at the output. For example, **00** or **11**, otherwise is at logic-0.

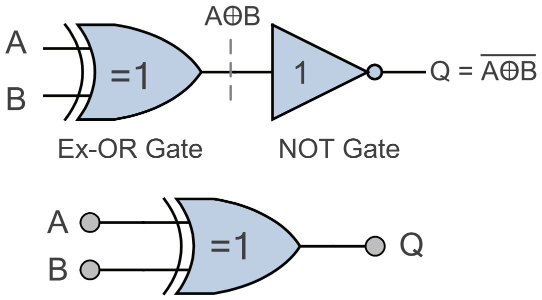
The **Ex-NOR** gate, also written as: **XNOR**, gives and output 1 when its inputs are logically equal or "equivalent" to each other. Thus an **Exclusive-NOR** gate is sometimes called an *Equivalence Gate.*

> *The Exclusive-NOR gate produces a HIGH output only when there is an even number of HIGH inputs*

The symbol used to describe the Boolean expression for an Exclusive-NOR function is the same as before with a plus sign, (**+**) within a circle (**O**) but with a bar or overline (**¯**) above the output expression to signify the **NOT** or inversion function of the gate. Thus the resulting symbol being: ($\overline{\oplus}$) indicating a coincidence function.

The logic symbol for an Exclusive-NOR gate is the same as an Exclusive-OR gate but with a circle or "inversion bubble", (**o**) at its output to represent the complementary form of the Exclusive-OR gate as shown in Figure 21.

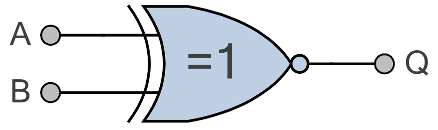
Figure 21. Exclusive-NOR Gate Equivalence



Therefore, as with the **NAND** and **NOR** gates, the **Ex-NOR** gate is simply a standard logic gate with an inverted or complementary output.

The truth table, logic symbol and implementation of a 2-input Exclusive-NOR gate is given in Table 9.

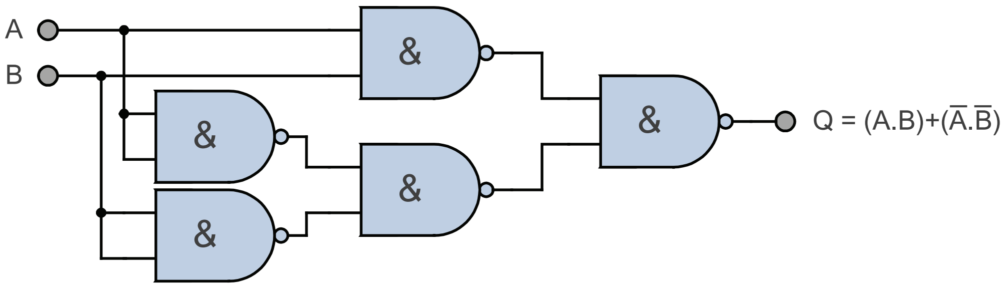
Table 9. A 2-input Exclusive-NOR Gate and Truth Table

Symbol	Truth Table		
 2-input Ex-OR Gate Symbol	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression Q = $\overline{A\oplus B}$	A AND B the SAME gives Q		

Note that the output is only **1** when there is an even number of binary **1** inputs. That is only an EVEN number of logic-1's on its inputs will produce a logic-1 at the output. This gives us the Boolean expression of: Q = $\overline{A.B}$ + A.B = $\overline{A\oplus B}$

The *Exclusive-NOR* gate is available as a standard TTL **74LS266** Quad 2-input gate. We can also construct an Ex-OR gate using a combination of **NOT**, **AND**, and **OR** gates but is much easier using the universal **NAND** gate as shown in Figure 22.

Figure 22. Exclusive-NOR Gate Using Universal NAND Gates



10. The Digital Buffer

Another digital logic device which is a complement, opposite or inverse of another standard type of logic gate is the **Digital Buffer**.

The *digital buffer* is the logical opposite of an inverter (Not Gate) we look at in section 5. However, unlike the single input, single output **NOT** gate which inverts or complements its input signal on the output, the digital buffer performs no inversion or decision making capabilities (like logic gates with two or more inputs). But instead produces an output which exactly matches that of its input. In other words, a digital buffer does nothing as its output state equals its input state.

You may be thinking, well what's the point of a non-inverting *Digital Buffer* if it does not invert or alter its input signal in any way, or make any logical decisions or operations based on its input like the **AND** or **OR** gates do.

The Digital Buffer isolates one digital device from another providing digital amplification

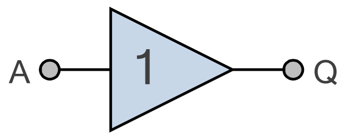
Well, sometimes in digital electronic circuits we need to isolate logic gates from each other or have them drive or switch higher than normal power loads, such as relays, solenoids and lamps. Thus the main advantages of a digital buffer is that it provides high current digital amplification. Actually, two inverting **NOT** gates cascaded together in series will operate the same as a buffer.

Digital buffers can be regarded as an Idempotent Gate since when an input passes through this device its value isn't changed in any way. The output of buffer is same as the input since it is a non-inverting device and will therefore give us the Boolean expression of: Q = A. Then we can define the logical operation of a single input digital buffer as being:

"Q is true, only when A is true"

In other words, the output (**Q**) state of a buffer is only true (logic-1) when its input **A** is true, otherwise its output is false (logic-0). Thus the logic symbol for the digital buffer is also of a triangular shape with one input and one output but with no inversion bubble at its end as shown in Table 10.

Table 10. The Digital Buffer and Truth Table

<table>
<tr><th>Symbol</th><th colspan="2">Truth Table</th></tr>
<tr><td rowspan="3">

Digital Buffer Symbol</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>Boolean Expression Q = A</td><td colspan="2">A give Q</td></tr>
</table>

Commonly available digital buffer IC's includes the TTL **74LS07** Buffer/Driver with open-drain outputs, or the **74LS34** Hex Buffer/Driver gate.

Other digital devices designed around the buffer include two-state Schmitt Trigger and Tri-state buffers which have a second enable input either positively enabled or negatively enabled.

End of this Digital Logic Gate Fundamentals eBook

With the completion of this Digital Logic Fundamentals eBook you should have gained a good and basic understanding and knowledge of standard and inverted digital logic gates. The information provided here should give you a firm foundation for continuing your study of electronics and electrical engineering as well as the study of digital logic circuits.

For more information about any of the topics covered here please visit our website at:

www.electronics-tutorials.ws