The Multiplexer

The Multiplexer (MUX)

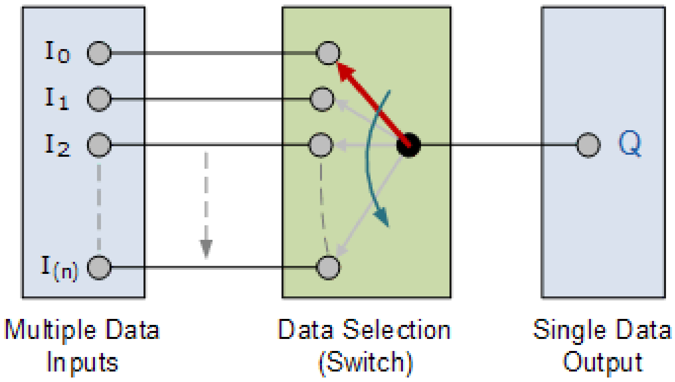
Multiplexing is the generic term used to describe the operation of sending one or more analogue or digital signals over a common transmission line at different times or speeds and as such, the device we use to do just that is called a Multiplexer.

The *multiplexer*, shortened to "MUX" or "MPX", is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal. Multiplexers operate like very fast acting multiple position rotary switches connecting or controlling multiple input lines called "channels" one at a time to the output.

Multiplexers (http://amazon.com/dp/129202562X/?tag=basicelecttut-20), or MUX's, can be either digital circuits made from high speed logic gates used to switch digital or binary data or they can be analogue types using transistors, MOSFET's or relays to switch one of the voltage or current inputs through to a single output.

The most basic type of multiplexer device is that of a one-way rotary switch as shown.

Basic Multiplexing Switch



The rotary switch, also called a wafer switch as each layer of the switch is known as a wafer, is a mechanical device whose input is selected by rotating a shaft. In other words, the rotary switch is a manual switch that you can use to select individual data or signal lines simply by turning its inputs "ON" or "OFF". So how can we select each data input automatically using a digital device.

In digital electronics, multiplexers are also known as data selectors because they can "select" each input line, are constructed from individual **Analogue Switches (http://www.electronics-tutorials.ws/combination/comb_1.html)** encased in a single IC package as opposed to the "mechanical" type selectors such as normal conventional switches and relays.

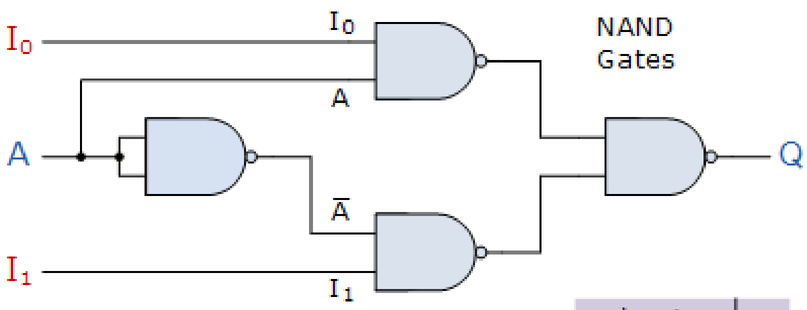
They are used as one method of reducing the number of logic gates required in a circuit design or when a single data line or data bus is required to carry two or more different digital signals. For example, a single 8-channel multiplexer.

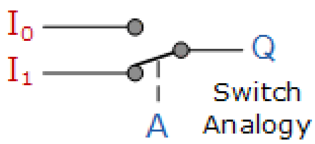
Generally, the selection of each input line in a multiplexer is controlled by an additional set of inputs called *control lines* and according to the binary condition of these control inputs, either "HIGH" or "LOW" the appropriate data input is connected directly to the output. Normally, a multiplexer has an even number of 2^N data input lines and a number of "control" inputs that correspond with the number of data inputs.

Note that multiplexers are different in operation to *Encoders*. Encoders are able to switch an n-bit input pattern to multiple output lines that represent the binary coded (BCD) output equivalent of the active input.

We can build a simple 2-line to 1-line (2-to-1) multiplexer from basic logic NAND gates as shown.

2-input Multiplexer Design





Truth Table

Inputs			
A	I_1	I_0	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

The input A of this simple 2-1 line multiplexer circuit constructed from standard NAND gates acts to control which input (I_0 or I_1) gets passed to the output at Q.

From the truth table above, we can see that when the data select input, A is LOW at logic 0, input I_1 passes its data through the NAND gate multiplexer circuit to the output, while input I_0 is blocked. When the data select A is HIGH at logic 1, the reverse happens and now input I_0 passes data to the output Q while input I_1 is blocked.

So by the application of either a logic "0" or a logic "1" at A we can select the appropriate input, I_0 or I_1 with the circuit acting a bit like a single pole double throw (SPDT) switch. Then in this simple example, the 2-input multiplexer connects one of two 1-bit sources to a common output, producing a 2-to-1-line multiplexer and we can confirm this in the following Boolean expression.

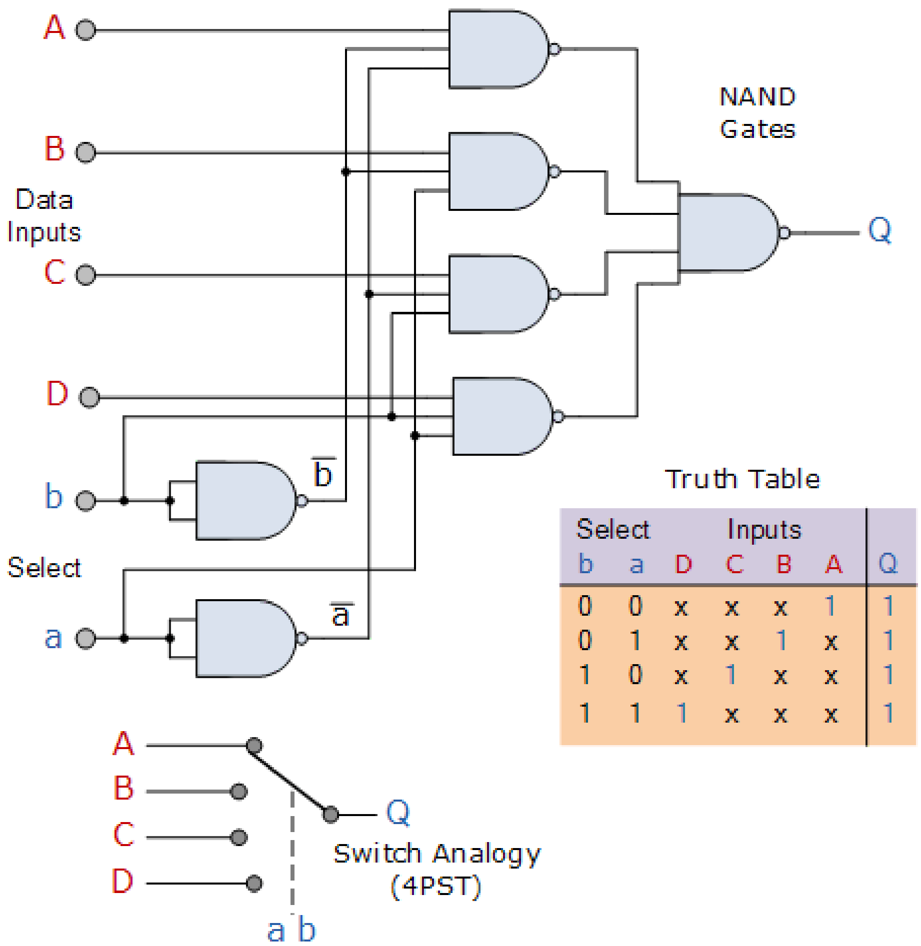
$$Q = \overline{A}.\overline{I_0}.I_1 + \overline{A}.I_0.I_1 + A.I_0.\overline{I_1} + A.I_0.I_1$$

and for our 2-input multiplexer circuit above, this can be simplified too:

$$Q = \overline{A}.I_1 + A.I_0$$

We can increase the number of data inputs to be selected further simply by following the same procedure and larger multiplexer circuits can be implemented using smaller 2-to-1 multiplexers as their basic building blocks. So for a 4-input multiplexer we would therefore require two data select lines as 4-inputs represents 2^2 data control lines give a circuit with four inputs, I_0, I_1, I_2, I_3 and two data select lines A and B as shown.

4-to-1 Channel Multiplexer



Select		Inputs				
b	a	D	C	B	A	Q
0	0	x	x	x	1	1
0	1	x	x	1	x	1
1	0	x	1	x	x	1
1	1	1	x	x	x	1

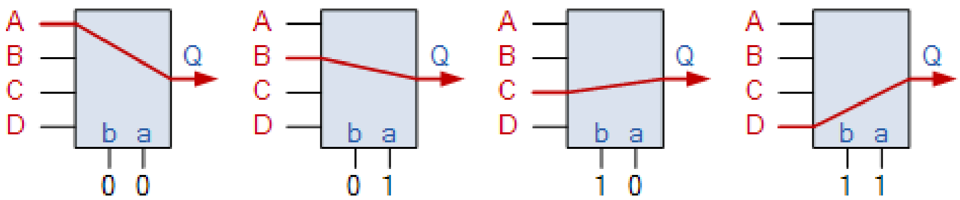
The Boolean expression for this 4-to-1 **Multiplexer** above with inputs A to D and data select lines a, b is given as:

$$Q = \overline{ab}A + \overline{a}bB + a\overline{b}C + abD$$

In this example at any one instant in time only ONE of the four analogue switches is closed, connecting only one of the input lines A to D to the single output at Q. As to which switch is closed depends upon the addressing input code on lines "a" and "b", so for this example to select input B to the output at Q, the binary input address would need to be "a" = logic "1" and "b" = logic "0".

Then we can show the selection of the data through the multiplexer as a function of the data select bits as shown.

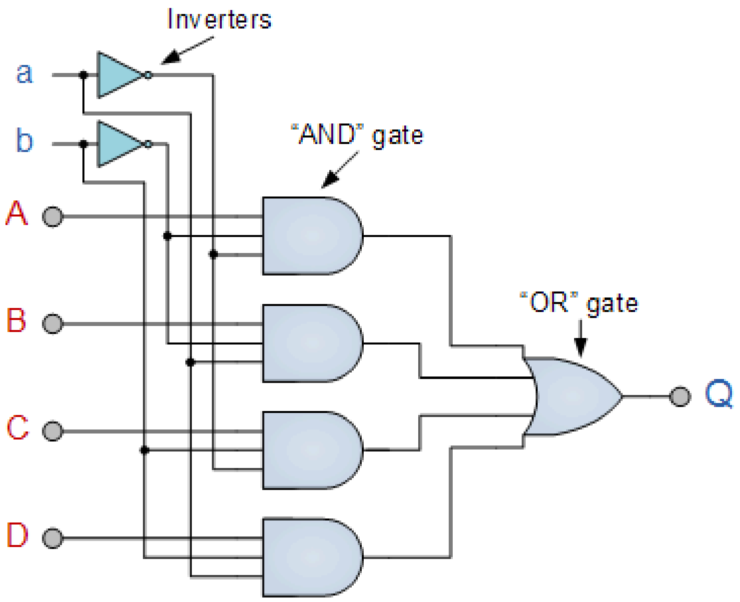
Multiplexer Input Line Selection



Adding more control address lines will allow the multiplexer to control more inputs but each control line configuration will connect only ONE input to the output.

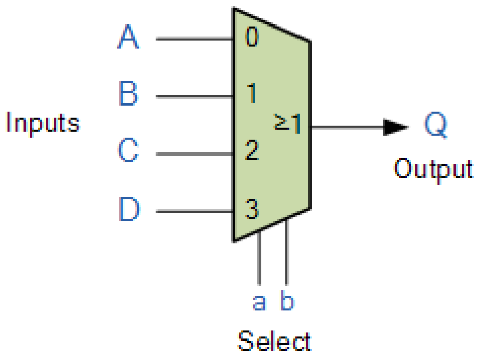
Then the implementation of the Boolean expression above using individual logic gates would require the use of seven individual gates consisting of AND, OR and NOT gates as shown.

4 Channel Multiplexer using Logic Gates



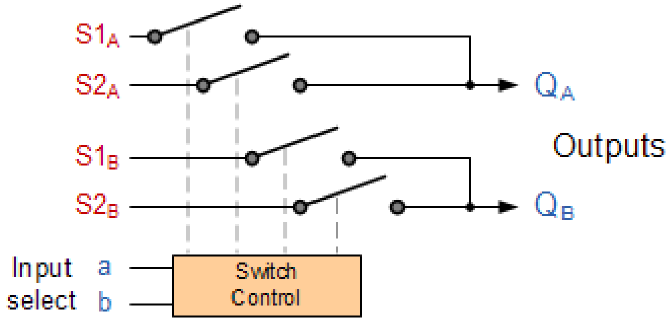
The symbol used in logic diagrams to identify a multiplexer is as follows.

Multiplexer Symbol



Multiplexers are not limited to just switching a number of different input lines or channels to one common single output. There are also types that can switch their inputs to multiple outputs and have arrangements or 4-to-2, 8-to-3 or even 16-to-4 etc configurations and an example of a simple Dual channel 4 input multiplexer (4-to-2) is given below:

4-to-2 Channel Multiplexer

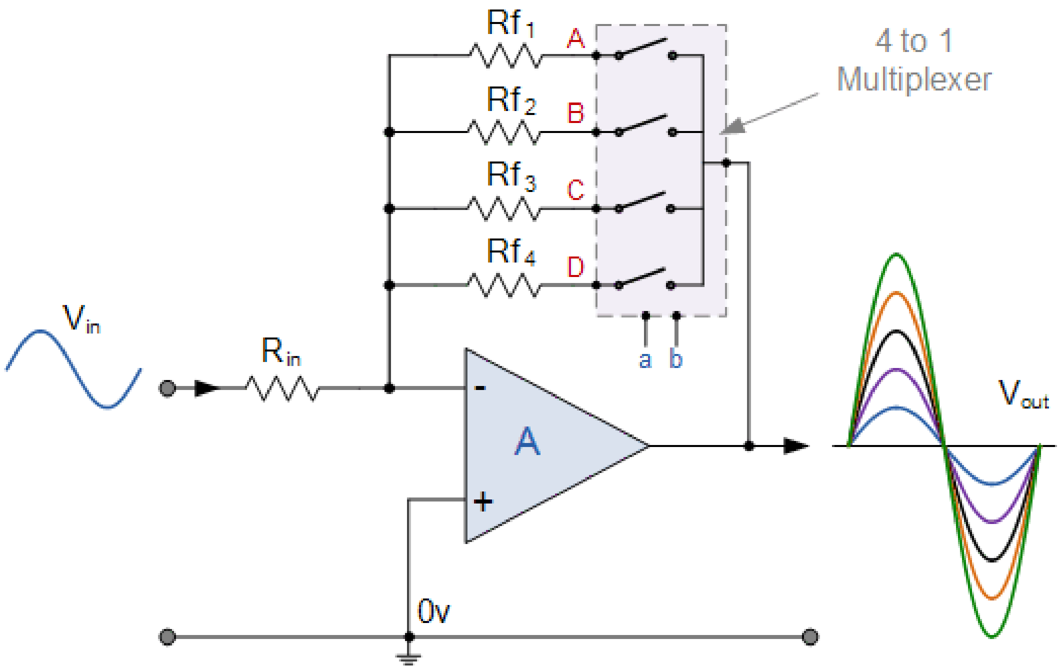


Here in this example the 4 input channels are switched to 2 individual output lines but larger arrangements are also possible. This simple 4-to-2 configuration could be used for example, to switch audio signals for stereo pre-amplifiers or mixers.

Adjustable Amplifier Gain

As well as sending parallel data in a serial format down a single transmission line or connection, another possible use of multi-channel multiplexers is in digital audio applications as mixers or where the gain of an analogue amplifier can be controlled digitally, for example.

Digitally Adjustable Amplifier Gain



Here, the voltage gain of the inverting operational amplifier is dependent upon the ratio between the input resistor, Rin and its feedback resistor, Rf as determined in the **Op-amp (http://www.electronics-tutorials.ws/opamp/opamp_2.html)** tutorials.

A single 4-channel (Quad) SPST switch configured as a 4-to-1 channel multiplexer is connected in series with the resistors to select any feedback resistor to vary the value of Rf. The combination of these resistors will determine the overall gain of the amplifier, (Av). Then the gain of the amplifier can be adjusted digitally by simply selecting the appropriate resistor combination.

Digital multiplexers are sometimes also referred to as “Data Selectors” as they select the data to be sent to the output line and are commonly used in communications or high speed network switching circuits such as LAN´s and Ethernet applications.

Some multiplexer IC´s have a single inverting buffer (NOT Gate) connected to the output to give a positive logic output (logic “1”, HIGH) on one terminal and a complimentary negative logic output (logic “0”, LOW) on another different terminal.

It is possible to make simple multiplexer circuits from standard **AND (http://www.electronics-tutorials.ws/logic/logic_2.html)** and **OR (http://www.electronics-tutorials.ws/logic/logic_3.html)** gates as we have seen above, but commonly multiplexers/data selectors are available as standard i.c. packages such as the common TTL 74LS151 8-input to 1 line multiplexer or the TTL 74LS153 Dual 4-input to 1 line multiplexer. Multiplexer circuits with much higher number of inputs can be obtained by cascading together two or more smaller devices.

Multiplexer Summary

Then we can see that **Multiplexers** are switching circuits that just switch or route signals through themselves, and being a combinational circuit they are memoryless as there is no signal feedback path. The multiplexer is a very useful electronic circuit that has uses in many different applications such as signal routing, data communications and data bus control applications.

When used with a demultiplexer, parallel data can be transmitted in serial form via a single data link such as a fibre-optic cable or telephone line and converted back into parallel data once again. The advantage is that only one serial data line is required instead of multiple parallel data lines. Therefore, multiplexers are sometimes referred to as “data selectors”.

Multiplexers can also be used to switch either analogue, digital or video signals, with the switching current in analogue power circuits limited to below 10mA to 20mA per channel in order to reduce heat dissipation.

In the next tutorial about combinational logic devices, we will look at the reverse of the **Multiplexer** called the **Demultiplexer (http://www.electronics-tutorials.ws/combination/comb_3.html)** which takes a single input line and connects it to multiple output lines.

« Combinational Logic Circuits (http://www.electronics-tutorials.ws/combination/comb_1.html) | The Demultiplexer (http://www.electronics-tutorials.ws/combination/comb_3.html) »

Other Good Tutorials in this Category

Binary Adder (http://www.electronics-tutorials.ws/combination/comb_7.html)

Binary Decoder (http://www.electronics-tutorials.ws/combination/comb_5.html)

Binary Subtractor (http://www.electronics-tutorials.ws/combination/binary-subtractor.html)

Combinational Logic Circuits (http://www.electronics-tutorials.ws/combination/comb_1.html)

Digital Comparator (http://www.electronics-tutorials.ws/combination/comb_8.html)

Display Decoder (http://www.electronics-tutorials.ws/combination/comb_6.html)

Priority Encoder (http://www.electronics-tutorials.ws/combination/comb_4.html)

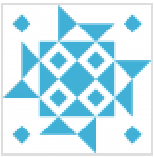
The Demultiplexer (http://www.electronics-tutorials.ws/combination/comb_3.html)

The Multiplexer (http://www.electronics-tutorials.ws/combination/comb_2.html)

Tags: **Combination Logic (http://www.electronics-tutorials.ws/tag/combination-logic)** **Digital Logic (http://www.electronics-tutorials.ws/tag/digital-logic)**

26 Responses to “The Multiplexer”

Older comments (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-1#comments)

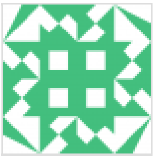


Susan Meier

do you the frequency of the tone the UCC-4 produces?

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=7833#respond)

August 19th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-7833)



David

For the Adjustable Amplifier Gain, what type of chip would be the “single 4-channel (Quad) SPST switch”? Or would any type of mux work?

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=7044#respond)

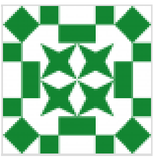
June 8th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-7044)

Wayne Storr (http://www.electronics-tutorials.ws)

A 4052 dual 4-channel or a 4051 single 8-channel analogue multiplexer/demultiplexer or any such device would work, but they must be analogue and not digital switches.

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=7045#respond)

June 8th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-7045)

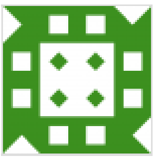


amo

that was good! well tried

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=7007#respond)

June 3rd, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-7007)



vasu

From the truth table we can see that when data select input, A is LOW (logic 0), input I1 passes its data to the output while input I0 is blocked. When data select A is HIGH (logic 1), input I0 is passed to Q while input I0 is blocked. (FALSE)

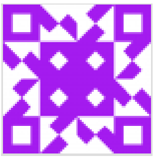
From the truth table we can see that when data select input, A is LOW (logic 0), input I0 passes its data to the output while input I1 is blocked. When data select A is HIGH (logic 1), input I1 is passed to Q while input I0 is blocked.

May 19th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-6864)
Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=6864#respond)

Wayne Storr (http://www.electronics-tutorials.ws)

Thanks Vasu 🙂

May 19th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-6866)
Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=6866#respond)



priyanka

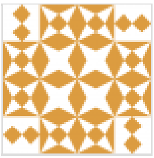
No this is wrong. The earlier statement is correct according to the given truth table. I cannot understand this. When A is low which one gets selected i0 or i1?

September 6th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-8073)
Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=8073#respond)

Wayne Storr (http://www.electronics-tutorials.ws)

The 2-bit multiplexer example above takes two input signals, i0 and i1, and “switches” one of them to the single output, Q under the control of a 1-bit data select input, A. If this data select signal is LOW (logic 0), then Q is equal to input i1. If this data select input is HIGH (logic 1), then Q is equal to input i0 as per the given truth table.

September 7th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-8078)
Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=8078#respond)



Scott Stornetta

First, these are great. Thank you.

Next, I am puzzled by one aspect of the Boolean equations you give for the 2×1 multiplexor

Q = A.I0.I1 + A.I0.I1 + A.I0.I1 + A.I0.I1
and its simplified form
Q = A.I1 + A.I0
I am familiar with basic boolean equations and the use of demorgan’s laws to simplify them. However, I have not seen the not/complement bars which you have over the subscripts 1 and 0. Does it mean conditional? Do you have a simple explanation or a link for where I could learn to understand this better?

Thank you.

February 6th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-5811)

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=5811#respond)

Wayne Storr (http://www.electronics-tutorials.ws)

No, The NOT function is basically a unary function that inverts or complement individual bits turning a 1 into a 0 and a 0 into a 1. Algebraically, NOT is represented by a short horizontal bar over the expression being complemented as (not 0) = 1. Then the complement of A is represented by not-A (or A-bar instead of not-A). Therefore, if A = 0, then not-A = 1, and if A = 1, then not-A = O. So any input with a bar over it is equal to 0

February 6th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-5815)

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=5815#respond)

wilayat shah

nice

January 29th, 2015 (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-2#comment-5736)

Reply (http://www.electronics-tutorials.ws/combination/comb_2.html?replytocom=5736#respond)

Older comments (http://www.electronics-tutorials.ws/combination/comb_2.html/comment-page-1#comments)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

What's the Answer *

+ 8 = twelve

Comment

Post Comment

80

(http://schematics.com/)

| RSS (http://feeds.feedburner.com/electronics-tutorials) | Privacy Policy & Cookies (http://www.electronics-tutorials.ws/privacypolicy) | Terms of Use (http://www.electronics-tutorials.ws/terms) | Site Map (http://www.electronics-tutorials.ws/sitemap) | Contact Us (http://www.electronics-tutorials.ws/contact) | Find us on Google+ (https://plus.google.com/+Electronics-tutorialsWs/posts) | Electronics Tutorials (http://www.electronics-tutorials.ws/) |