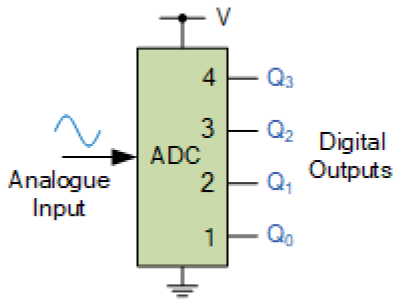


Analogue to Digital Converter

Analogue to Digital Converter, or ADC, is a data converter which allows digital circuits to interface with the real world by encoding an analogue signal into a binary code

Analogue-to-Digital Converters, (ADCs) allow micro-processor controlled circuits, Arduinos, Raspberry Pi, and other such digital logic circuits to communicate with the real world. In the real world, analogue signals have continuously changing values which come from various sources and sensors which can measure sound, light, temperature or movement, and many digital systems interact with their environment by measuring the analogue signals from such transducers.

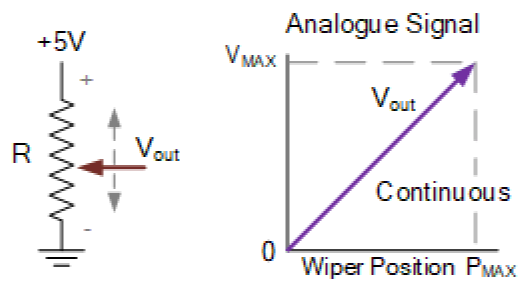
While analogue signals can be continuous and provide an infinite number different voltage values, digital circuits on the other hand work with binary signal which have only two discrete states, a logic “1” (HIGH) or a logic “0” (LOW). So it is necessary to have an electronic circuit which can convert between the two different domains of continuously changing analogue signals and discrete digital signals, and this is where *Analogue-to-Digital Converters* (A/D) come in.

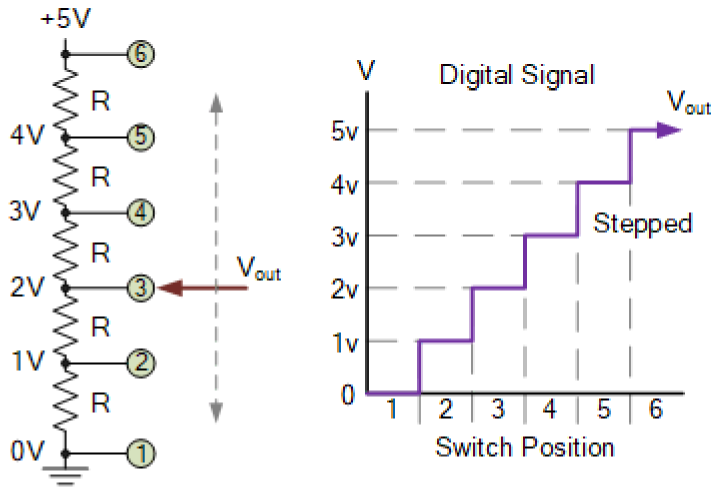
Basically an analogue to digital converter takes a snapshot of an analogue voltage at one instant in time and produces a digital output code which represents this analogue voltage. The number of binary digits, or bits used to represent this analogue voltage value depends on the resolution of an A/D converter.

For example a 4-bit ADC will have a resolution of one part in 15, (2^4 – 1) whereas an 8-bit ADC will have a resolution of one part in 255, (2^8 – 1). Thus an analogue to digital converter takes an unknown continuous analogue signal and converts it into an “n”- bit binary number of 2^n bits.

But first let us remind ourselves of the differences between an analogue (or analog) signal and a digital signal as shown:

Analogue and Digital Signals





Here we can see that as the wiper terminal of the potentiometer is rotated between between 0 volts and V_{MAX}, it produces a continuous output signal (or voltage) which has an infinite number of output values relative to the wiper position. As the potentiometers wiper is adjusted from one position to the next, there is no sudden or step change between the two voltage levels thereby producing a continuously variable output voltage. Examples of analogue signals include temperature, pressure, liquid levels and light intensity.

For a digital circuit the potentiometer wiper has been replaced by a single rotary switch which is connected in turn to each junction of the series resistor chain, forming a basic potential divider network. 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 steps representing multiples of 1.0 volts on each switching action or step as shown.

So for example, the output voltage will be 2 volts, 3 volts, 5 volts, etc. but NOT 2.5V, 3.1V or 4.6V. Finer output voltage levels could easily be produced by using a multi-positional switch and increasing the number of resistive elements within the potential divider network, therefore increasing the number of discrete switching steps.

Then we can see that the major differences between an analogue signal and a digital signal is that an “Analogue” quantity is continuously changing over time while a “Digital” quantity has discrete (step by step) values. “LOW” to “HIGH” or “HIGH” to “LOW”.

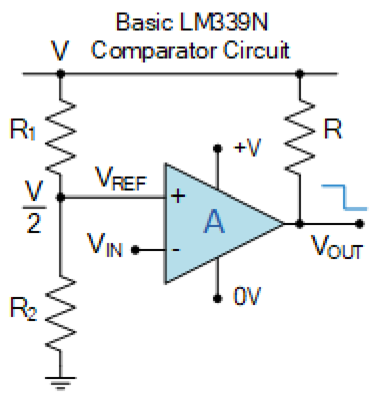
So how can we convert a continously changing signal with an infinite number of values to one which has distinct values or steps for use by a digital circuit.

Analogue-to-Digital Converter

Parallel of “Flash” A/D converters use a series of interconnected but equally spaced comparators and voltage references generated by a series network of precision resistors for generating an equivalent output code for a particular n-bit resolution.

The advantage of parallel or flash converters is that they are simple to construct and do not require any timing clocks as the instant an analogue voltage is applied to the comparator inputs, it is compared against a reference voltage. Consider the comparator circuit below.

Comparator Circuit



An analogue comparator such as the LM339N which has two analogue inputs, one positive and one negative, and which can be used to compare the magnitudes of two different voltage levels.

A voltage input, (V_{IN}) signal is applied to one input of the comparator, while a reference voltage, (V_{REF}) to the other. A comparison of the two voltage levels at the comparator’s input is made to determine the comparators digital logic output state, either a “1” or a “0”.

The reference voltage, V_{REF} is compared against the input voltage, V_{IN} applied to the other input. For an LM339 comparator, if the input voltage is less than the reference voltage, ($V_{IN} < V_{REF}$) the output is “OFF”, and if it is greater than the reference voltage, ($V_{IN} > V_{REF}$) the output will be “ON”. Thus a comparator compares two voltage levels and determines which one of the two is higher.

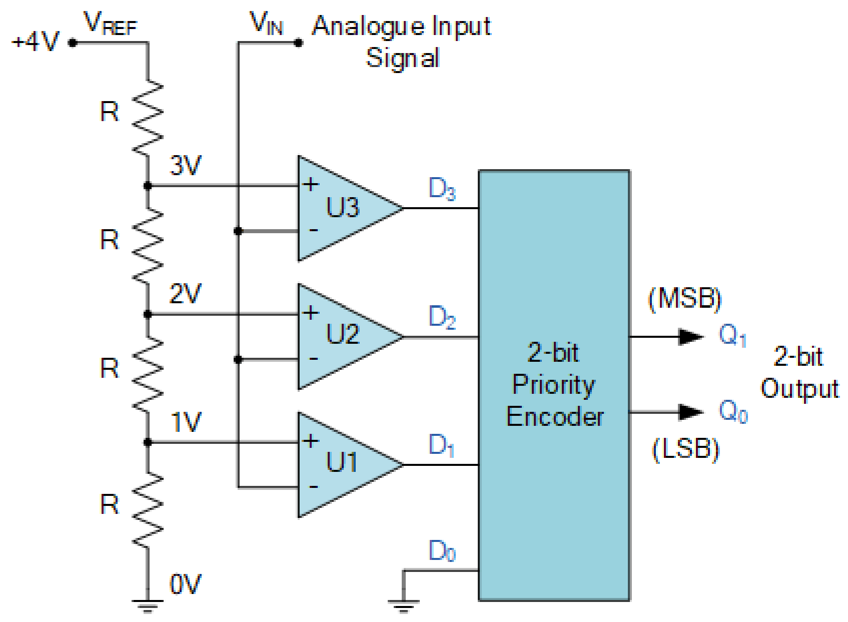
In our simple example above, V_{REF} is obtained from the voltage divider network setup by R_1 and R_2. If the two resisors are of equal values, that is $R_1 = R_2$, then clearly the reference voltage level will be equal to half the supply voltage, or V/2. So for a comparator with an open-collector output, if V_{IN} is less than V/2, the output is HIGH, and if V_{IN} is greater than V/2, the output is LOW acting as a 1-bit ADC.

But by adding more resistors to the voltage divider network we can effectively “divide” the supply voltage by an amount determined by the resistances of the resistors. However, the more resistors we use in the voltage divider network the more comparators will be required.

In general, $2^n – 1$ comparators would be required for conversion of an “n”-bit binary output, where “n” is typically in the range from 8 to 16. In our example above, the single bit ADC used $2^1 – 1$, which equals “1” comparator to determine if V_{IN} was greater or smaller than the V/2 reference voltage.

If we now create a 2-bit ADC, then we will need $2^2 – 1$ which is “3” comparators as we need four different voltage levels corresponding to the 4 digital values required for a 4-to-2 bit encoder circuit as shown.

2-bit Analogue to Digital Converter Circuit



This will give us a 2-bit output code for all four possible values of analogue input of:

2-bit A/D converter Output

Analogue Input Voltage (V_{IN})	Comparator Outputs				Digital Outputs	
	D_3	D_2	D_1	D_0	Q_1	Q_0
0 to 1 V	0	0	0	**0**	0	0
1 to 2 V	0	0	**1**	X	0	1
2 to 3 V	0	**1**	X	X	1	0
3 to 4 V	**1**	X	X	X	1	1

Where: “X” is a “don’t care”, that is either a logic “0” or a logic “1” condition.

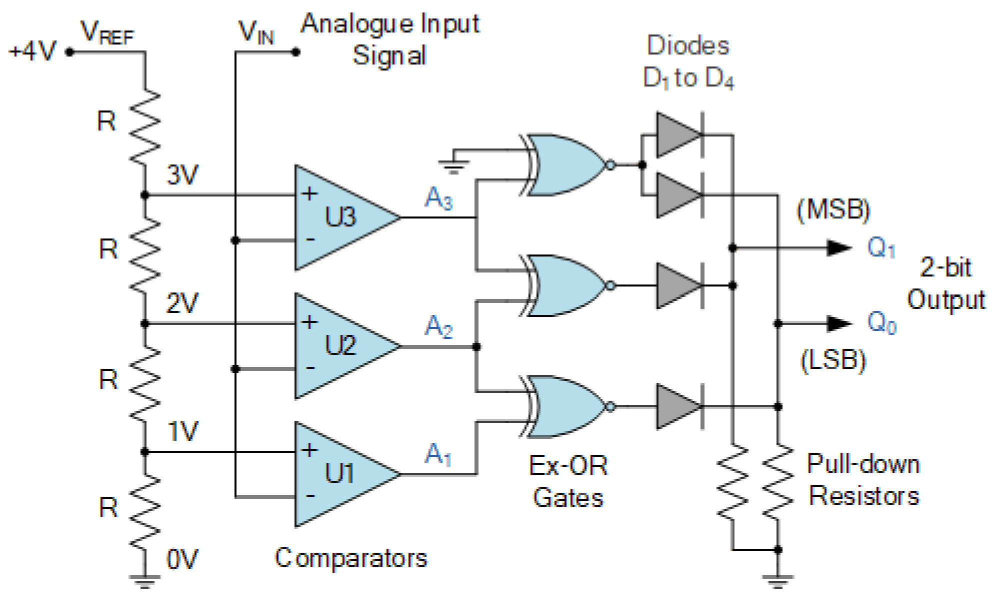
So how does this *analogue-to-digital converter* work. For an A/D converters to be useful it has to produce a meaningful digital representation of the analogue input signal. Here in this simple 2-bit ADC example we have assumed for simplicity that the input voltage V_{IN} is between 0 and 4 volts, so have set V_{REF} and the resistive voltage-divider network to drop 1 volt across each resistor.

When V_{IN} is between 0 and 1 volt, (<1V) the input on all three comparators will be less than the reference voltage, so their outputs will be LOW and the encoder will output a binary zero (00) condition on pins Q_0 and Q_1. When V_{IN} increases and exceeds 1 volt but is less than 2 volts, ($1V<V_{IN}<2V$) comparator U1 which has a reference voltage input set at 1 volt, will detect this voltage difference and produce a HIGH output. The priority encoder which is used as the 4-to-2 bit encoding detects the change of input at D_1 and produces a binary output of “1” (01).

Note that a **Priority Encoder** such as the TTL 74LS148 allocates a priority level to each individual input. The priority encoders output corresponds to the currently active input which has the highest priority. So when an input with a higher between each reference voltage level, each comparator will output either a HIGH or a LOW condition to the encoder which inturn produces a 2-bit binary code between 00 and 11 relative to V_{IN}.

This is all well and good, but priority encoders are not available as 4-to-2 bit devices, and if we use a commercially available one such as the TTL 74LS148 or its CMOS 4532 equivalent which are both 8-bit devices, then six of the binary bits would not be used. But a simple encoder circuit can be made using digital Ex-OR gates and a matrix of signal diodes as shown.

2-bit ADC Using Diodes



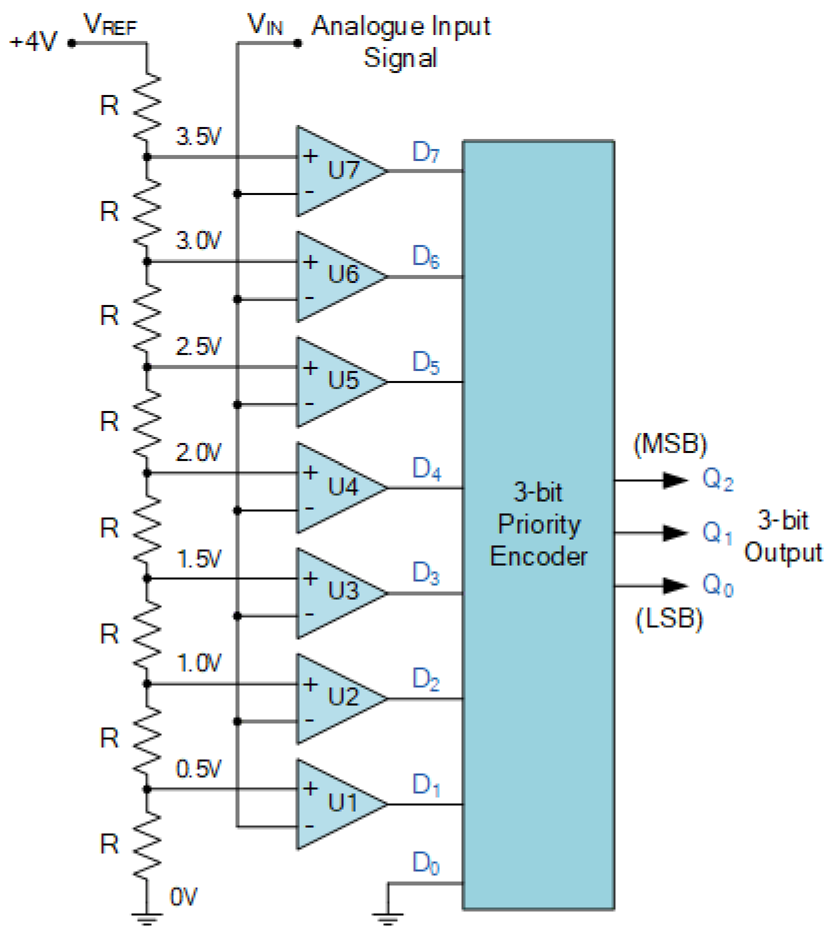
Here the outputs of the comparators are encoded using Exclusive-OR gates before being fed to the diodes. Two external pull-down resistor are used at their outputs and ground (0V) to ensure a LOW condition and stop the outputs from floating when the diodes are reverse biased.

So as with the previous circuit, depending upon the value of V_{IN} determines which comparator produces an output signal HIGH (or LOW) to the exclusive-OR gates producing a HIGH output if one input or the other input is HIGH, but not both, (Boolean expression is $Q = \overline{A}.B + A.\overline{B}$). These Ex-OR gates could also be constructed using the combinational logic AND–OR–NAND gates.

The problem here with both designs of 4-to-2 converters is that the *resolution* of this simple 2-bit A/D converter is 1 volt because as we have seen, the analogue input voltage at V_{IN} must change by one full volt in order for the encoder to change its output code. One way to improve the resolution of the output is to increase it to a 3-bit A/D converter using more comparators.

3-bit Analogue-to-Digital Converter

The parallel ADC above converts the analogue input voltage in the range from 0 to over 3 volts to produce a 2-bit binary code. Since a 3-bit digital logic system can generate $2^3 = 8$ different digital outputs, the analogue input voltage can therefore be compared against eight reference voltage levels with each voltage level being equal to one eighth (V/8) of the reference



This will give us a 3-bit output code for all eight possible values of analogue input of:

3-bit A/D converter Output

Analogue Input Voltage (V_{IN})	Comparator Outputs								Digital Outputs		
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0 to 0.5 V	0	0	0	0	0	0	0	**0**	0	0	0
0.5 to 1.0 V	0	0	0	0	0	0	**1**	X	0	0	1
1.0 to 1.5 V	0	0	0	0	0	**1**	X	X	0	1	0
1.5 to 2.0 V	0	0	0	0	**1**	X	X	X	0	1	1
2.0 to 2.5 V	0	0	0	**1**	X	X	X	X	1	0	0
2.5 to 3.0 V	0	0	**1**	X	X	X	X	X	1	0	1
3.0 to 3.5 V	0	**1**	X	X	X	X	X	X	1	1	0
3.5 to 4.0 V	**1**	X	X	X	X	X	X	X	1	1	1

Where again “X” is a “don’t care”, that is either a logic “0” or a logic “1” input condition.

However, the advantage of this type of parallel or flash A/D converter is that its real-time conversion rate is relatively fast and can be easily built as part of a project if only a few binary bits are required to produce a read out on a digital display to show the voltage value of an analogue input signal.

As well as taking an analogue input signal from a sensor or transducer and converting it using an analogue-to-digital converter into a digital binary code as part of an input interfacing circuit, we can also take a binary code and convert it into an equivalent analogue quantity using a Digital-to-Analogue Converter for output interfacing to control a motor or actuator, or commonly in audio applications.

In the next tutorial about digital circuits, we will look at the a digital-to-analogue converter, or simply DAC, which are the exact opposite of the *analogue-to-digital converters* looked at here. DAC's use op-amps and resistive divider networks to convert an "n"-bit binary number into an equivalent analogue output voltage, or current signal.

Read more Tutorials inCombinational Logic

11 Comments

Join the conversation

the non-inverting terminal.

Posted on February 08th 2022 | 9:47 am
Reply

- *Muhammad Omar Ahmad*

  Simply put, all what I wanted.
  THANK YOU

  Posted on January 21st 2022 | 1:39 am
  Reply
- *JANEMING MING*

  , the compressor looks good

  Posted on September 10th 2021 | 5:57 am
  Reply
- *cythinay*

  thanks for you sharing !!! it’s very usful !!!

  Posted on August 27th 2021 | 2:41 am
  Reply
- *cythinay*

  Thank you for the knowledge that I just need every time you post an article. It’s so weird. Thank you for your priority encoder

  Posted on August 11th 2021 | 7:20 am
  Reply
- *markee suyi*

  This is an article on high-quality analog-to-digital converters.

  Posted on August 04th 2021 | 7:24 am
  Reply
- *Marshall*

  Thanks for the post.

  One mistake there, the comparator will output 0 if Vin falls within that range instead of 1 in your output table. Maybe you want to add a “NOR” after the comparator opamp?

  Posted on July 26th 2021 | 9:54 pm
  Reply
- *linda birdseye*

  Would I find a mfg. converter any where that is a ADC 3 bit

  Posted on May 29th 2021 | 6:10 pm
  Reply

This article is really helpful. Thank you for sharing this information by order. I will look at other articles too.

Posted on January 17th 2021 | 2:07 pm
Reply

- *A Mama*

  Good approach All resistors must be high prescision and same value to minimize errors

  Posted on December 06th 2020 | 7:34 pm
  Reply