

ADC'S ANALOG TO DIGITAL CONVERTERS

The heart of the computer-based data acquisition is usually the analog to digital converter (ADC). Basically this device is a voltmeter. Its input is voltage and its output is a digital number (a collection of 0's and 1's) proportional to the input voltage. The computer is usually programmed to convert this outputted digital number into a number that represents the measured voltage.

Consider, for example, ADC designed to measure between 0 and 10 volts and convert this measurement into an 8-bit digital number. Then a 0 volt input is associated with the digital output of 0 (in binary 0000 0000) and a 9.999 volt input is associated with the digital output of 255 (in binary 1111 1111). In general the digital output of the ADC is related to the input voltage by the algorithm:

$$\text{Digital_Output} = \text{Integer part of } [256 * (\text{Volts_Input}/10)].$$

In this example if the input voltage was 4.78231415 volts then the ADC would produce the number 122 as its output. The computer program would then take this number and relate it with a voltage using the equation:

$$\text{Voltage_Reading} = (\text{Digital_Output}) * 10 / 256$$

For this example this final conversion yields 4.78. Note that because the input was converted to an 8-bit integer in the middle of the process, a great deal of potential precision has been lost. For the 8-bit device with a range of 0-10 volt, the smallest change in the output reflects a difference of 0.039 volts (10 Volts/256) on the input. Thus the error caused by the processing of a signal through this ADC would be ± 0.02 volts (or larger if the electronics are noisy).

As was shown in the above example, an 8-bit ADC can not tell the difference between two close voltage, because its resolution is $0.039 = 10/2^8$. There are ADC's with higher resolutions such as 12-bit and 16-bit devices. These would have resolutions of $0.0024 = 10/2^{12}$, and $0.00015 = 10/2^{16}$ respectively. But this higher resolution comes at a cost, in either dollars or speed and usually both. Which of these depends on the type of ADC used. An alternate method of increasing the resolution would be to reduce the input range, but this may limit the utility of the ADC.

To discuss type of ADC's, consider a 2-bit ADC with a 0 - 10 volt range. The following table relates the input voltage range to the digital output of the device:

Voltage Range	0.00-2.49	2.50-4.99	5.00-7.49	7.50-9.99
Digital Output -- binary	00	01	10	11
Digital Output -- decimal	0	1	2	3

Most ADC's use a circuit called a comparator. This circuit has two inputs and one output. The inputs (usually labeled + and -, but for our purposes we will call the + input, the **signal** input and the - input the **reference** input) are analog signals and the output is a single bit digital signal, a "yes" = 1 or "no" = 0). If the voltage applied to the **signal** input is greater than the voltage applied to the **reference** than the output is 1. Basically the comparator answers the question "Is the **signal** is greater than **reference** voltage?"

The first two types of ADC described below (the counter type and the successive approximation

type) are like a systematic approach to a number guessing game (I am thinking of a number between 0 and 16) in which after each guess the one making the guesses is told if her last guess is high or low. If a young child were playing the game, he might just make random guesses. A slightly older child would start at the lowest number and just start counting name all of the numbers until they hit the choosen number. Counting is systematic and is a fast way to cover all of the numbers This is like the counter type ADC. An clever adult will make use of the high/low information and first guess 8. If this is high, her second guess will be 4, and so forth. This gussing scheme is similar to that used by the successive approximation ADC's.

One of the simplest types of ADC is the **counter type ADC**, the input signal of the ADC is connected to the signal input of its internal comparator. The ADC then systematically increases the voltage on the reference input of the comparator until the reference becomes larger than the signal and the comparator's output goes to 0 (= no). For example consider an input signal is 4.78 volts. The initial voltage on the comparator's - input would be 2.5 volts. This is less than 4.78 so the next higher voltage (5.00 volts) is applied. This is greater than 4.78 volts so the comparator switches to 0. The digital output of the ADC is the number of times the ADC increase the voltage after starting at the initial 2.5 volts. In this case the output is 1 (= 01 in binary). This scheme is relatively simple, but as the number of ADC bits increases the time it takes to scan through all of the possible values lower than input will grow quickly.

A smarter variation of this scheme, used by the **successive approximation type ADC**, is to start with the initial reference voltage at half the maximum voltage range of the ADC. This determines which half of the ADC's range includes the input voltage and thus the first bit of the output number in binary. The next step divides this half in half, and then that half of a half in half again and so on. It take as many steps as the number of bits associated with the ADC. In our 2-bit example, with a 4.78 volt signal the reference voltage starts at 5 volts, and the comparator indicates that the signal is smaller than the reference, so the most significant (left side of a binary number) bit is set to 0. Next the reference voltage is set to 2.5 volts and the comparator indicates the signal is now larger than the signal, Thus the second bit will be set to 1. If we were using an ADC with more bits we would continue dividing the interval and assigning another bit in each step. As it is the result in our example is 1 (= 01 in binary)

For a 2-bit ADC there is no real advantage to successive approximation method over the counter method, but the successive approximation method for an 8-bit system this usually works faster. It is possible for small signals for a counter type to be faster than a successive approximation type, because the successive approximation type always take the same number of steps to get the answer while the counter type might get the answer on the first step. Often the predictability of the time to complete an measurement of the successive approximation ADC is a desirable characteristic. The successive approximation type ADC is the most commonly used type.

The fastest ADC is the **flash type ADC**, in which there is as many comparators as there are possible outputs. For example, an 8-bit ADC would have $2^8 = 256$ comparators and a 16-bit unit would have $2^{16} = 65,536$ comparators. Rather than changing the voltage to one comparator, in a flash type ADC each voltage that on might apply to the single comparator in a counter type is hard wired to its own comparator. A bunch of logic circuits determine which is the largest reference signal less than the input signal The measurement is done in one step, but these get expensive quickly with increasing resolution. For our example of a 2-bit ADC, the reference voltages would be 2.50, 5.00, 7.50 are associated with the output numbers 1,2,3. If the signal is less than all of the reference voltages, than the output number is 0. The 2.50 reference voltage is the largest reference less than the 4.78 volt signal, so in this example the digital output is once again 1.

Another type of ADC, called the **Wilkinson type ADC** measures the input voltage, by using it to charge a capacitor. The capacitor is then discharged at a constant current. The discharge time is measured and this time is proportional to the input voltage. For a 2-bit Wilkinson ADC it take 4 clock ticks for this capacitor to discharge if it is initially at the maximum voltage for the ADC.