

## Bridging the Divide: Part I - DAC Introduction

“The best of what we are and hold as true: Always it is by bridges that we live.” -Philip Larkin, British poet

*By Bill McCulley, Applications Engineer National Semiconductor*

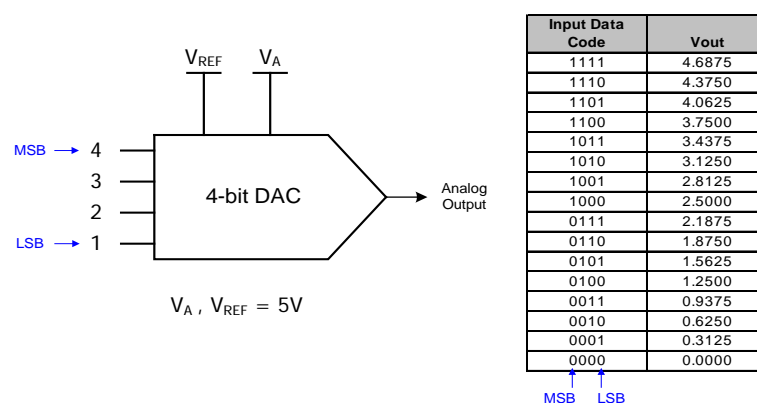
### INTRODUCTION

Our world is not a digital environment of absolutes. The signals of the real world are not made of logical highs and lows, or zeros and ones. These signals are *analog* and they meander within a range of voltages or currents. The purpose of the Digital to Analog Converter (DAC) is to convert digital data into an analog signal where the ‘real world’ exists. The digital data may originate from a microprocessor, ASIC, or FPGA, but at some point it requires conversion to an analog signal to have impact on the real world. Whether the system uses an audio amplifier, or an LED indicator, or a motor driver, the final signal will be analog in nature. The DAC serves as that bridge to transfer a digital signal into the analog domain – and hopefully ends with an accurate output signal! To design well with DACs it is good to review the fundamentals. This article covers basic DAC operation and key definitions, along with the most typical DAC topologies. The second article will discuss DAC design and implementation along with key issues such as noise. The last article will review two important DAC applications: *calibration* and *motor control*.

Since the time of the Nyquist-Shannon sampling theorem, engineers have developed and used DACs. Indeed, the earliest DACs made of discrete components predate the invention of the integrated circuit. However, only the past 25 years have monolithic DACs become widely available. According to the Nyquist-Shannon sampling theorem, any sampled data can be reconstructed perfectly - provided it meets bandwidth and Nyquist criteria.<sup>1</sup> So, with proper design, the DAC can reconstruct sampled data in your application correctly and with precision.

### DIGITAL-TO-ANALOG CONVERTER OPERATION

**Figure 1.**



**Figure 1** is a diagram of a 4-bit parallel-input DAC.<sup>2</sup> For a 4-bit DAC, there are  $2^4 = 16$  possible input data codes, as shown above. For the input data code, DACs may use straight binary or two's complementary system<sup>3</sup>, with straight binary being most common. DACs will have an analog reference ( $V_{REF}$ ), along with power supply ( $V_A$ ), and an analog output. In many cases, the reference and supply voltages may be the same, and therefore the DAC will have a single pin for both functions. Also, the reference can be a voltage or a current, depending on DAC design. The DAC multiplies the input data code by the reference to generate the output

and serves as the opposite function of ADC, which is a *divider* to convert an analog input into digital bits. Finally, the DAC can have a voltage or current output, depending on design. This article will focus on voltage-output DACs as the most common architecture available.

The *Least Significant Bit (LSB)* is the rightmost bit of the data code and represents the smallest value in a digital code and the *Most Significant Bit (MSB)* is the leftmost bit of the data code and represents the half-scale value. As you can see on the table, the LSB (0001b) represents 0.3125V. The LSB value is determined by the basic equation below:

$$LSB\ Value = G \times \frac{V_{REF}}{2^N} \quad [EQ\ 1]$$

In most DACs, the gain (G) will be 1, which reduces the equation to  $V_{REF}/2^N$ . In an ideal DAC, each additional bit will increase the output voltage by one LSB. If the value of one LSB is 0.3125V, then 0.3125V is the smallest increment the DAC can resolve. Multiplying the LSB value and the data code ( $D_{IN}$ ), the basic transfer function for a DAC is:

$$DAC\ Output = G \times D_{IN} \times \frac{V_{REF}}{2^N} \quad [EQ\ 2]$$

So, with an input data code of 1111, the DAC output is shown below. One can see that the maximum output value in this example is one LSB (0.3125V) below the voltage reference (5V).

$$DAC\ Output = 1 \times 15 \times \frac{5}{2^4} = 4.6875V$$

Returning to Equation 1, the LSB value is inversely proportional to the number of bits (N) and directly proportional to  $V_{REF}$ . Figure 2 shows a pure sinusoid (A), the output of a 4-bit DAC (B), and a 6-bit DAC (C). While an ideal DAC output is actually represented by discrete points, the output of a DAC in operation resembles a ‘stair step’ signal on an oscilloscope. As you can see on signals (B) and (C), an increase in the number of bits decreases the LSB value, and therefore improves the DAC resolution. You can also achieve a similar improvement by decreasing the reference voltage. However this will also result in a lower full-scale range for the signal, since the maximum achievable output is one LSB less than the reference.

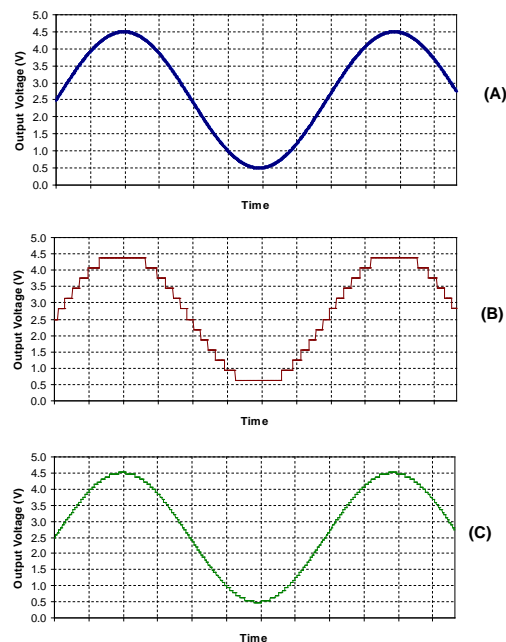
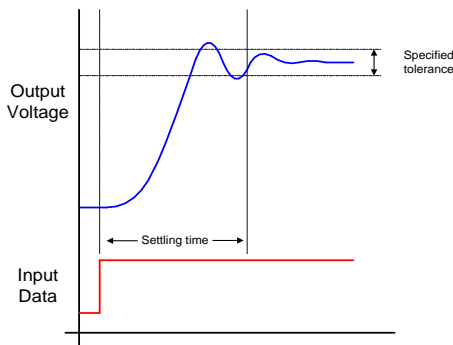
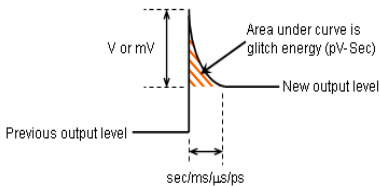


FIGURE 2 – Sinusoid and DAC resolution

## KEY TERMS AND DEFINITIONS

The basic operation of a DAC is easily understood, but the terms used among semiconductor manufacturers can appear confusing and even contradictory. So it is important for a designer to understand the meanings of key parameters as you review DAC datasheets for your application.

Term	Definition
<i>Monotocity</i>	<i>Monotocity</i> is the condition in which the DAC's transfer function's slope does not change. <i>Monotocity</i> is an indication of linearity.
<i>Resolution</i>	<i>Resolution</i> is the number of bits, which along with the analog reference, determines granularity of the signal conversion. Resolution also refers to the output value representing one LSB.
<i>Settling Time</i>	<p><i>Settling Time</i> (Figure 3) is the time from a change in the input code until the DAC output signal is generated and remains within specified output tolerance or characterization range. It is important to know that the characterization range may differ among vendors, and that can have a large impact on specified settling time!</p> 
<i>Output Glitch</i>	<p><i>Output Glitch</i> is the energy injected into the analog output when the input data code changes. The amount of glitch energy depends upon how many bits are changing from high to low or low to high. The output glitch occurs at the major carry (e.g. for example, changes from 0111b to 1000b) but it can also occur during other transitions.</p> 
<i>Offset Error</i>	<i>Offset Error</i> , also referred as <i>Zero Code Error</i> , is the difference between the actual and ideal output when the input code is zero. So, if the zero code error for a part is specified as 1.1mV, and the input data word is 0000b, then the output voltage will be 1.1mV an offset error above 0V.
<i>DNL &amp; INL</i>	<i>Differential Non-Linearity (DNL)</i> describes the error in step size and is the measure of the maximum deviation from the ideal step size of 1 LSB. <i>Integral Non-Linearity (INL)</i> is a measure of the deviation of each individual code from the straight line through the input to output transfer function. Both these terms merit a full tutorial in their relation to DACs and ADCs, so a link is available at the end of this article. <sup>4</sup>

## DAC ARCHITECTURES

*String DACs* are among the most popular DACs and have many variants available. These include the basic Kelvin Divider, binary weighted, digital potentiometer-focused, segmented-string DACs among other types. The Kelvin Divider, also referred as a string divider, is the most common and simplest DAC topology. Shown on Figure 3, this topology uses internal resistors with switches at each node and a logic block for decoding the binary inputs. An N-bit DAC will contain  $2^N$  resistors and  $2^N$  switches. This topology has a voltage output, and is monotonic by its nature and quite linear in using exactly matched resistors. A major disadvantage for this topology is relatively high output impedance. Most vendors add an internal amplifier as an output buffer to make a low impedance source to follow-on circuits. There are a large number of resistor/switches required, depending on the resolution of the DAC. A 4-bit DAC only requires 16 resistors/switches; while a medium resolution 12-bit DAC will require 4096. However, process improvements have made this topology very common in DACs up to 12-14 bits. To the right of the string divider diagram, you can see the DAC121S101, which integrates the DAC with an amp buffer and serial SPI interface for control and input data.

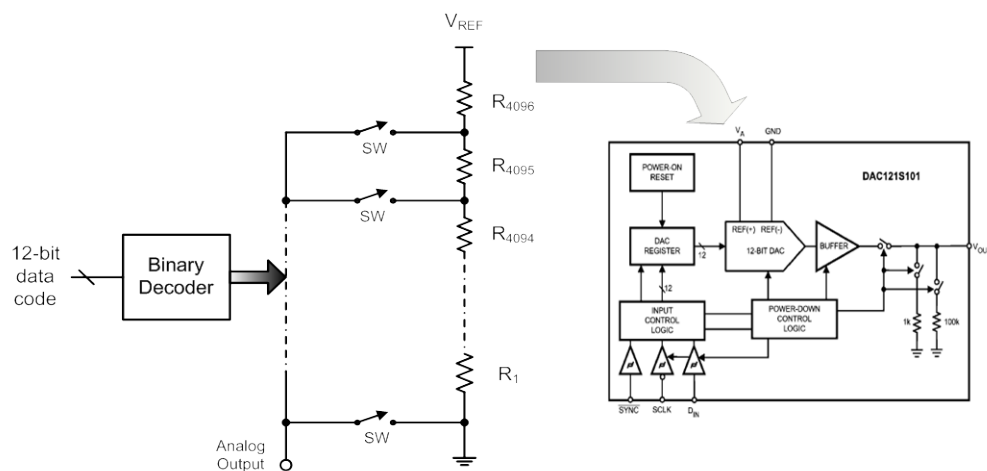
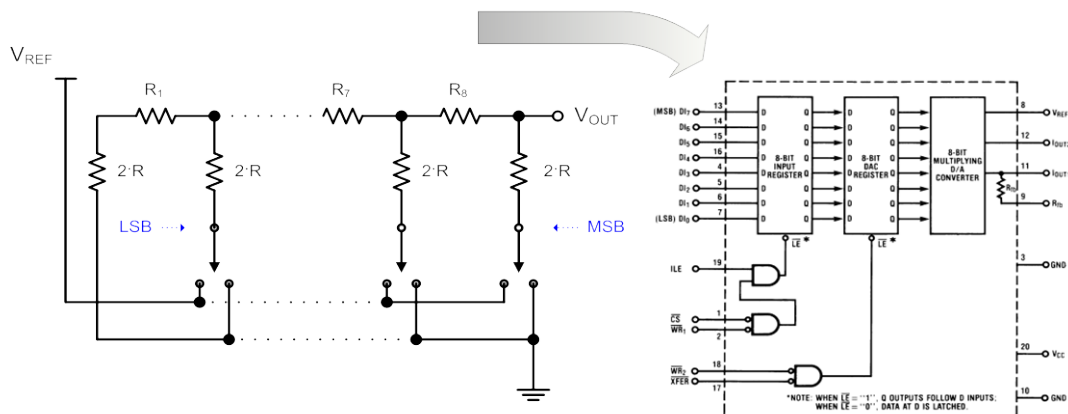


Figure 3 – Kelvin Divider and 12-bit String Divider - DAC121S101

*R-2R Ladder DACs* are another very common topology. In Figure 5, this voltage output DAC uses two values of resistors, with a ratio of 2-to-1. As seen below, the number of resistors is much reduced compared to string DACs. Any R2-R DAC needs just  $2 \cdot N$  resistors – making trimming the resistors values easier. A 4-bit DAC requires only 8 resistors, and as shown below an 8-bit DAC will require just 24 resistors. To the right of the R-2R diagram is an IC DAC which includes parallel input code register, along with support function blocks.

[Figure 5 – R-2R Ladder]



The R-2R ladder can either be designed with a voltage or current output. A key benefit of using the voltage output is the constant output impedance which makes it easier to interface with a buffer amplifier on the output. Much like string DACs, there are several variants of the R-2R ladder topology to increase performance as process technology improves.

*Multiplying DACs* (MDAC) are products based on the R-2R ladder structure. Since the R-2R switches can be done with CMOS switches, it is easy to construct a ladder to use bi-polar signals on the input. Using bipolar – positive and negative – signals on the input, the topology can be designed to enable two-quadrant and four-quadrant MDACs. These MDACs are used or even integrated in a Variable Gain Amplifier, but there may also be unique applications beyond that use.

*Segmented DACs* are a topology which mixes or cascades several other DACs architectures whether string or R-2R ladder. These specialized DACs are commonly seen in high speed video or audio systems in which the reconstruction of a signal demands performance across a wide range of voltage or frequencies.

*Sigma Delta (•-•) DACs* are a ‘relatively’ new topology and operate similar to Sigma Delta ADCs. This DAC circuit takes data at a low rate, adds zeros at a high rate into the data stream, and then filters over time at a high rate. The data stream is then passed through a •-• modulator which converts the data to a bit stream, followed by a 1-bit DAC which switches between equal and negative reference voltages. Sigma Delta DACs (and ADCs) has gained popularity due to high resolution and very good DNL. Sigma Delta DAC applications include calibration, audio, and voice band systems. It is limited in terms of bandwidth, so it is not used among high speed applications.

## SUMMARY

In the world of electronic design, the DAC plays an important role in converting digital data into an analog signal. Among a wide range of applications, the DACs can be found in calibration systems, motor control, factory test equipment, audio systems, measurement equipment, and control systems. But as important the DAC is, the design engineer should remember it is just one part of a system which can include a processor, power supplies, analog circuits, and other mixed-signal devices. The architecture is what ties these functional blocks together. Keeping that perspective in mind, the next article will cover designing with DACs.

Footnotes:

(1) H. Nyquist, "Certain topics in telegraph transmission theory", Proc IEEE, Vol. 90, Feb 2002  
(Reprint); <http://www.loe.ee.upatras.gr/Comes/Notes/Nyquist.pdf>

C. E. Shannon, "Communication in the presence of noise", Proc IEEE, Vol. 86, Feb 1998  
(Reprint); <http://www.stanford.edu/class/ee104/shannonpaper.pdf>

(2) Modern DACs use a serial interface such as I2C or SPI which allow for small packages and fewer pins. In this tutorial example, a legacy parallel input DAC is used.

(3) Tutorial on signed binary numbers –  
[http://en.wikipedia.org/wiki/Signed\\_number\\_representations](http://en.wikipedia.org/wiki/Signed_number_representations)

(4) Nick Gray's excellent tutorial on ADC & DAC errors (registration required) -  
[http://www.national.com/AU/design/0,4706,179\\_0\\_,00.html](http://www.national.com/AU/design/0,4706,179_0_,00.html)