Chapter 7: From Digital-to-Analog and Back Again

Overview

Often the information you want to capture in an experiment originates in the laboratory as an analog voltage or a current. Sometimes you want to use digital information to control a piece of laboratory equipment. In either case, you need something to interface between digital electronics and analog electronics. Depending on which of the two tasks you want to do, you will either use a *Digital to Analog Converter* (*DAC*) or an *Analog to Digital Converter* (*ADC*).

Note that, while ADC is usually referred to by its initials, DAC is usually pronounced to rhyme with "back." It is easy to get confused because a related term, *DAQ*, is a contraction for the generic term computer-based *Data Acquisition*, which refers to the computer-based collection of digital data.

This week you will explore DACs and ADCs, with the ultimate goal of creating an ADC to DAC circuit which is one the building blocks for a Digital Signal Processing (DSP) circuit. The circuit will allow you to compare an analog input signal to the digitally generated, or *synthesized*, output signal.

You will also see the idiosyncrasies of the digitization process including an effect called *aliasing*. We saw that fast analog signals can be degraded when you put them through an amplifier but effects like slew rates and frequency roll ups. In general, the fasts signals looked smoothed an attenuated. Digitized signals can show weird beat-frequency effects if the *sampling rate* is slower that the periods of the analog signal. This aliasing is actually a form of *difference- frequency* generation. In practice, the beat signal can actually be used to *lock* two frequencies together to a very high accuracy.

I. Digital to Analog Converters (DAC)

Most modern laboratory equipment can be controlled electronically. In the best case, the equipment will have a digital interface so that a computer can control the equipment by sending digital commands. When that is not possible, you can usually control the equipment with an analog (voltage or current) signal. In this case, a DAC can translate your computer's digital signals into analog signals to control the equipment. DACs are ICs that have an internal series of switches to connect a combination of divided reference values into an op-amp-based summing amplifier.

It is easy to make the divided reference voltages using a clever device called an *R*-2*R* resistor ladder, as shown in Figure 6-1. Look at this ladder from the right-hand side, and you will see two parallel resistors, each with a value of 2*R*. This has an equivalent resistance of *R*. When you add the

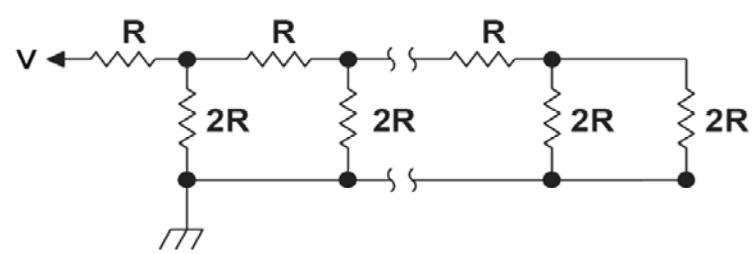

Figure 1: R-2R resistor ladder.

horizontal (series) resistor you get a total of 2*R* again. However, the next (to the left) 2*R* resistance to ground again produces an equivalent total resistance of *R*. Continuing farther to the left, we find that the effective resistance to ground is *R at every dot on the top line*!

When connected to a supply voltage, the ladder acts like a series of voltage dividers that reduces the voltage by an additional factor of 2 at each *R*-2*R* junction. V_{IN} decreases by half at each connection point along the top rail. Thus each output voltage is related to the input voltage by a power of two. We can generate an analog voltage by adding together the voltages represented by the various stages in the ladder. If we only sum outputs based on a simple a binary representation we can produce a DAC.

Note that the *current* through each 2*R* leg also reduces by a factor of 2 from left to right. This means that one can use these legs to generate a voltage signal or a current signal depending on what your particular equipment needs. Usually, current signals are faster, since they drive low impedance devices, and a lower *R* means a faster *RC* time constant. Finally, the impedance to ground at each *R*-2*R* junction always has the same value of *R*.

Voltage Mode

We will use the TLC7524 8-bit DAC, which can operate in either a *voltage* mode or a *current* mode. In either case, the DAC uses the binary input bits to control switches that force the output current through one of two output lines. Usually, one line (typically OUT2) is tied to ground. In voltage mode, the output is simply given by

$$V_{OUT} = V_{REF}(D/256),$$

Where V_{REF} is connected to OUT1, *D* is the input binary number, and the output voltage occurs at REF.

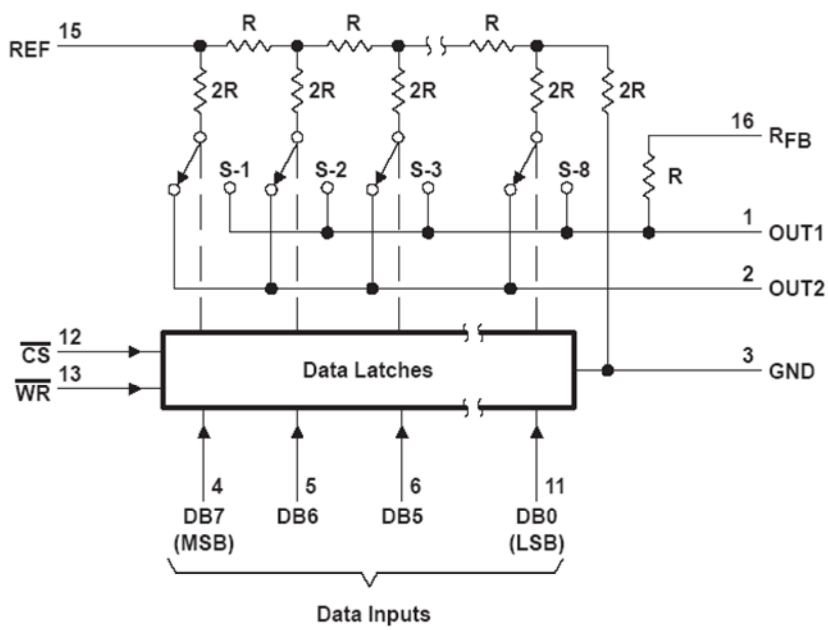

Figure 2: TLC7524 functional diagram.

This essentially uses the R-2R voltage divider with active loads to generate the digitally controlled voltage. The voltage mode is simple to connect, but the output cannot drive much current, since it has high output impedance.

Current Mode

In the current mode, one uses the output current in one of the two OUT legs. Typically, one connects these outputs to an op-amp to achieve a voltage-level output with low output impedance. The speed of the conversion process will then be limited by the response time of the op-amp. You may observe the inherent speed of the DAC by measuring the width of the current pulses along OUT1. In this configuration, the inverting amplifier changes the output sign, so that

$$V_{OUT} = -V_{REF}(D/256).$$

In order to get a positive output, you must use a negative V_{REF}. Remember that you must also supply power for the op-amp.

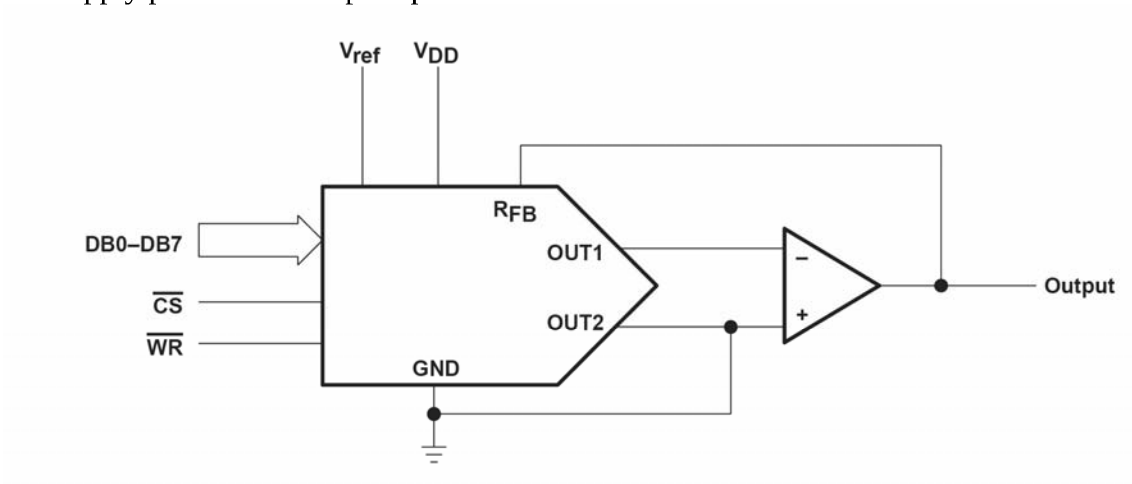

Figure 3: DAC in current mode, with an op-amp follower.

II. Analog to Digital Converters (ADC)

Even before the days of computer-controlled equipment, computers routinely processed data from experiments. Of course, this means that the data must first be translated from its natural analog condition into a digitized signal appropriate for a computer. Data acquisition systems, which are primarily Analog to Digital Converters (ADC), serve this role. Because this is such an important function these converters come in a variety of types using several different digitization techniques. The primary measures of an ADC are is speed and its accuracy (quoted in number of output bits). Often these are complementary, so that an 8-bit converter is usually faster than a 12-bit converter. When you are designing a system, you might also consider cost, complexity, output mode (parallel or serial data). Some ADCs have a Sample and Hold circuit at the front end that captures the voltage signal so that the converter can take its time to digitize.

One common ADC uses a clock to progressively approximate the input signal. Essentially, it sums the output of a DAC with the target signal. On the each clock tick, it sets increasing lower significant digits by checking the value of a voltage comparator that is fed by the target voltage and the output of the DAC. This ADC method is called successive approximation. Another technique converts the voltage to a frequency (using a voltage-controlled oscillator) and counts the frequency. Yet another method charges a capacitor with a constant current source until the capacitor's voltage is higher than the input voltage, while counting clock pulses. The fastest ADCs (parallel encoded or "flash ADCs" shown in figure 4) compare the input voltage to a set of 2^n voltages and determine the lowest voltage greater than the input. This requires lots of comparators, but it is very fast. These ADCs can digitize data at rates of several GHz. The fastest ones, of course, are the most expensive.

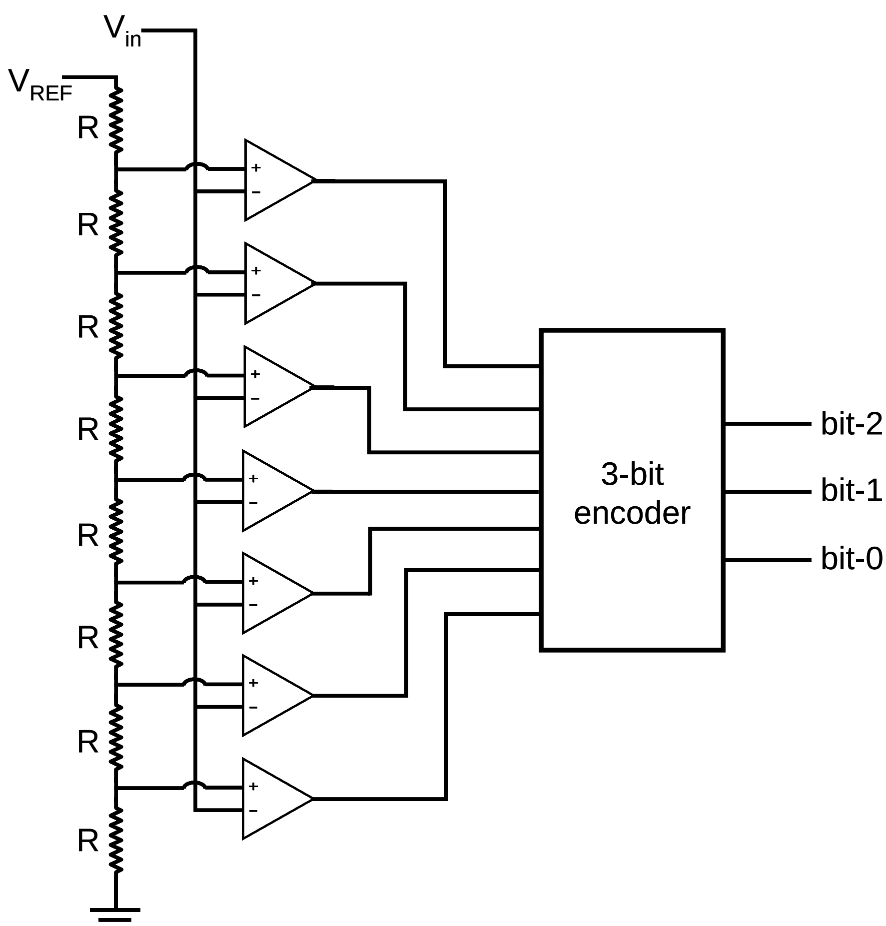

Figure 4: 3-bit flash ADC. The resistor network provides a series of reference voltages for the comparators. All the comparators above the input voltage go high, and the encoder converts the comparator outputs to a 3-bit binary number.

We will use the ADC0820, which is a flash ADC. In order to keep the number of comparators small, it holds the input voltage, and then converts it in two steps. First, it converts the upper four bits by comparing the voltage to 15 voltages from an R-2R ladder. Next, it converts the digitized value back into an analog value (reusing the R-2R ladder) and subtracts this from the input to generate the smaller, difference voltage. Finally, it uses a second set of 15 comparators to measure the result against 16 lower voltages. The entire process takes less than 800 ns, when operating off the internal timing of the ADC0820. You can run it somewhat faster with some clever timing (WR-RD) mode, but we will opt for simplicity. In the RD mode, the conversion starts when the WR# line goes low. When the conversion is complete, the INT# line goes low, and the data have been latched into the output buffers. The output buffers will be put into a high Z state when WR# goes low, until the INT# line goes low.

Design Exercise

Design Exercise 7-1: Construct a Quartus II project that will output consecutive values of a sinewave function in binary at each clock cycle, so that if the binary values were converted to an analog voltage then the output would be a sinewave. Your sinewave (one period) should include at least 100 points, and the amplitude should be constructed from 8-bit numbers.
Hint: Use a 2-dimensional register.