*Single Bit DACs in a Nut**shell***

Part I – DAC Basics

By Dave Van Ess, Principal Application Engineer, Cypress Semiconductor

Many embedded applications require generating analog outputs under digital control. It may be a DC reference voltage or an AC signal to stimulate transducers. It may be a reconstructed voice signal for some wireless application. The applications are endless. This function requires a **D**igital to **A**nalog **C**onverter or **DAC**.

DACs are cool! They allow analog signals to be generated under CPU control. The name even sounds cool. DAC, the name brings up images of a Clive Clusser adventure. DAC Bit, man of finite resolution!

The number of DACs on single chip systems is always limited, making them a dear resource. One solution is to use onboard digital resources and firmware, along with simple filters to build signal bit DACs. This article will explain the concept of single bit DACs, different techniques to construct them, and the benefits and consequences of each type.

A DAC, in its simplest form, is circuitry to generate an output that is percentage of a reference. A simple four bit implementation is shown below.

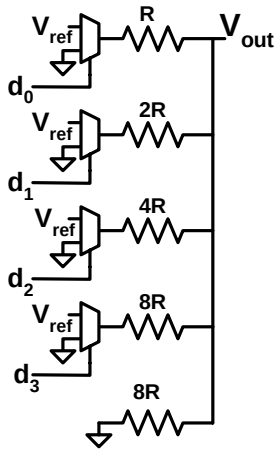

Figure 1. Four Bit DAC

The following equation defines the output voltage

$$V_{out} = \frac{d_0 V_{ref}}{2} + \frac{d_1 V_{ref}}{4} + \frac{d_2 V_{ref}}{8} + \frac{d_3 V_{ref}}{16} \quad \textbf{(1)}$$

If it is acceptable to use the supply voltage as the reference, then the input multiplexers can be replaced with digital outputs from the MCU. This simplifies the design to four digital output pins and five resistors as shown below.

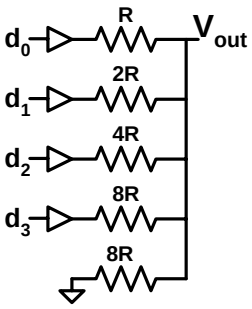

Figure 2. Ratiometric Four Bit DAC

When the supply voltage is used as the reference it is known as "**ratiometric**". This may sound cool but it is just really just clever marketing! It makes not having a real reference sound like a feature. Who would want buy a "10 Bit Serial Input, No Real Reference DAC"?

Within reason, each extra bit of resolution requires only one extra resistor and a port pin to drive it. Resistor tolerance is going to become a problem. If implemented with 1% resistors, resolution is limited to about 6bits. Each added resistor is doubles the value of the previous resistor. The tenth bit is 1024R. At some point the ratio of the resistors becomes impracticality large. This can be solved using the topology shown below.

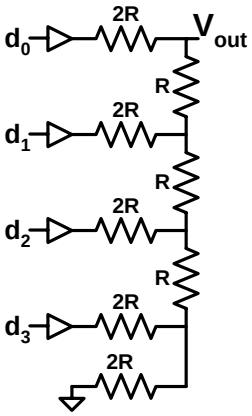

Figure 3. R - 2R Ladder Four Bit DAC

This is called an **R – 2R ladder**. It has the advantage that the resistor ratio never gets larger than two. If the **R** value is constructed with two parallel **2R** resistors, then only a single value of resistor is needed. Generally 1% resistors from the same reel hold a relative tolerance better than ½%. This increases the resolution limit to 7bits but requires three resistors per bit. To be accurate this topology really should be called a **½R – R ladder DAC**. However this was rejected, as some digital engineers could not fathom the idea of a number between zero and one.

So far the DACs discussed generate a particular ratio of the reference, 100% of the time. For example when an 8bit DAC with a 5V reference is set to 37 the output is 0.72V (5V * 37/256).

Suppose instead of supplying 14.5% of the reference 100% of the time, one could supply 100% of the reference 14.5% of the time. Just such a topology is shown below.

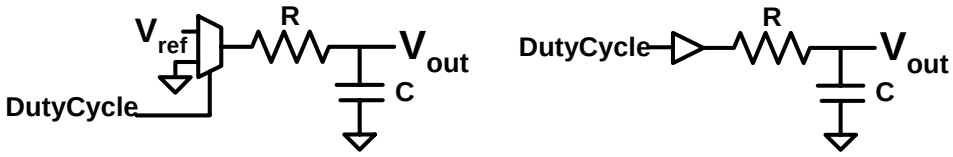

Figure 4. Single bit and Ratiometric Single bit DAC

The output voltage is $\mathbf{V_{ref}}$***DutyCycle**. Again, if it is acceptable to use the supply voltage as the reference, then the input multiplexer can be replaced by a single digital output from the MCU. This simplifies the design to a single digital output pin and a filter. These examples represent the filter as single pole R C but the filtering could be the flywheel effect of a motor or the eye viewing a pulsed LED. Any system component that has a slower response than the generated output frequency acts as the filter.

What waveform is ideal for generating a duty cycle? Well this is the stuff thesis's are made of. There are many ways to generate a duty cycle. Each has particular advantages. We'll close off this first section with the simplest method and leave the more advanced techniques for the next time.

Pulse Width Modulator (PWM)

One of the simplest ways to generate a duty cycle is to use a PWM. Many microcontrollers already are equipped with as least one and many in cases, several of them. A block diagram of a PWM is shown below.

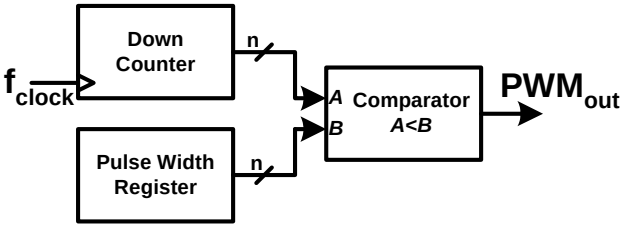

Figure 5. Pulse Width Modulator (PWM) Block Diagram

The hardware consisted of a down counter with some **Period** and a register to store a **PulseWidth** value. The comparator will go high whenever the counter's value is less the pulse width value. For a period of 256, the counter will count from 255 down to zero. If the pulse width value is 128 then the comparator output will be high when the counter output is from 127 down to zero, or 128 counts. The equations for the duty cycle and output frequency are shown below.

$$dc = \frac{PulseWidth}{Period} \qquad f_{out} = \frac{f_{clock}}{Period}$$

For PWMs, the output frequency is independent of the pulse width. The plots below are for two PWMs.

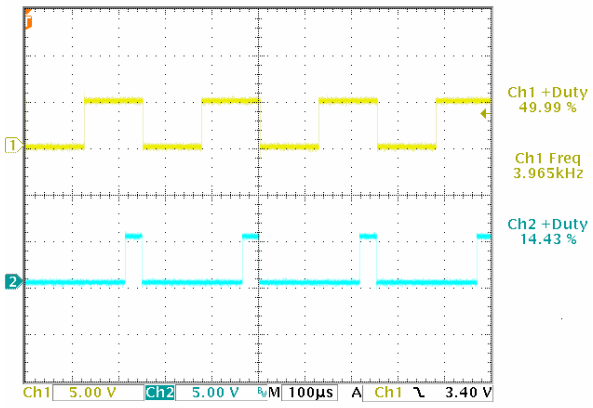

Figure 6. PWM Waveforms; 50% and 14.5% Duty Cycle

Both have a period of 256 and input clock of 1MHz, results in both having an output frequency of 3.9kHz. The top trace shows a PWM with its pulse width set to 128. This trace verifies that this is a 50% duty cycle. The bottom trace is for a PWM with a pulse width of 37. It has the same output frequency but has a duty cycle of 14.5% (37/256). Again the trace shows a signal that is high about 1/7th of the time. A spectral plot of the two signals is shown below.

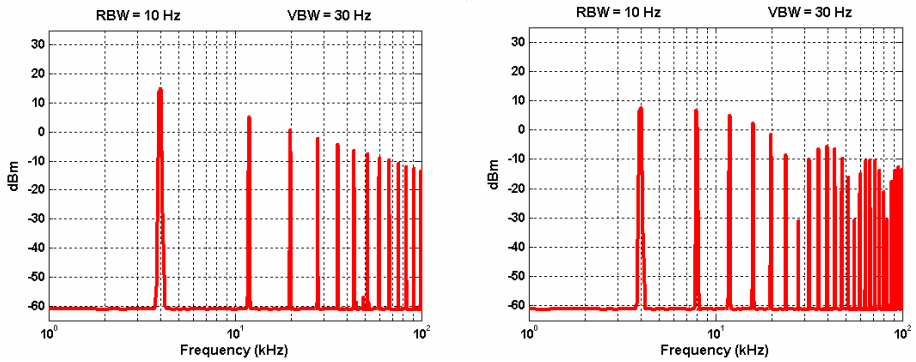

Figure 7. PWM Spectral Plots; 50% and 14.5% Duty Cycle

It is apparent that there are significant harmonics in these outputs. It will require filtering to remove them. Although relatively easy to build, PWMs suffer from significant harmonic generation that is at a relatively low frequency. Given a constant clock frequency, a PWM with finer resolution (larger period) has a lower output

frequency. The general solution is to increases the clock frequency. The maximum operating frequency for the counter and comparator limits the practical resolution.

We'll discuss several other techniques which get around these limitations in the concluding installment of this article.

END PART I

Single Bit DACs in a Nutshell

Part II – Advanced Single-Bit DAC Techniques

By Dave Van Ess, Principal Application Engineer, Cypress Semiconductor

In our first installment, "DAC Basics" (INSERT LINK TO PART I HERE), we explored how traditional resistor ladder networks can be replaced with simpler single-bit and ratiometric single bit DAC elements and how to drive them with a pulse width modulated (PWM) waveform. In this concluding chapter, we'll look at three other ways to drive these networks. Each will have example oscilloscope waveforms and spectral plots for duty cycles of 50% and 14.5%.

Delta Sigma Modulation (DSM)

PWMs reduce the number of transitions to the smallest possible value. (One high and one low per counter cycle.) DSM is a technique that increases the number of transitions to the largest possible value. For the same clock frequency, the harmonics are pushed farther out making it easier to filter them. A block diagram of a DSM is shown below

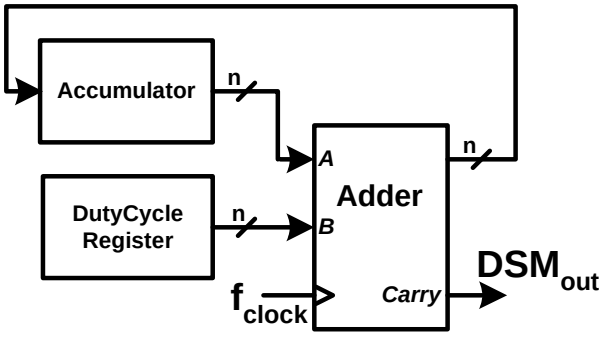

Figure 8. Delta Sigma Modulator (DSM) Block Diagram

The duty cycle value is added to an accumulated value when the adder overflows the output is high. For example, constantly adding 128 to an 8bit adder causes a carry every other time. Constantly adding 64 results in a carry one in four times. Adding 63 results in a carry one in four times, most of the time, but occasionally a carry one in five times. The equations for the duty cycle and output frequency are shown below.

$$dc = \frac{DutyCycleValue}{AccumWidth} \qquad f_{out} = \begin{cases} dc \cdot f_{clock} & : dc \le 0.5 \\ (1 - dc) \cdot f_{clock} & : dc > 0.5 \end{cases}$$

The output frequency is no longer set by the counter period. Independent of the adder width, f the duty cycle is limited to a range of 10% to 90%, the output frequency is guaranteed to be no smaller than 1/10th the clock frequency. The plots below are for two DSMs

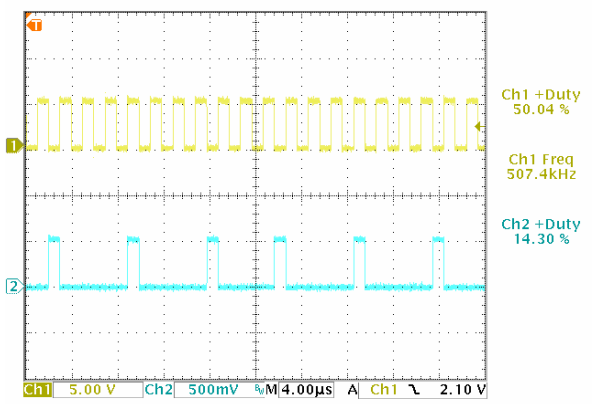

Figure 9. DSM Waveforms; 50% and 14.5% Duty Cycle

Both have an adder width of 256 and input clock of 1MHz. The top trace shows a DSM with its DutyCycleValue set to 128. This results in an output with a 50% duty cycle and an output frequency of 500kHz. The bottom trace is for a DSM with its DutyCycleValue set to 37. This results in an output high one part in seven most of the time, and occasionally one part in six, for an average output frequency is 145kHz (1MHz*37/256). The generated harmonics are pushed well past the 3.9kHz of the PWM example. A spectral plot of the two signals shown below confirms this.

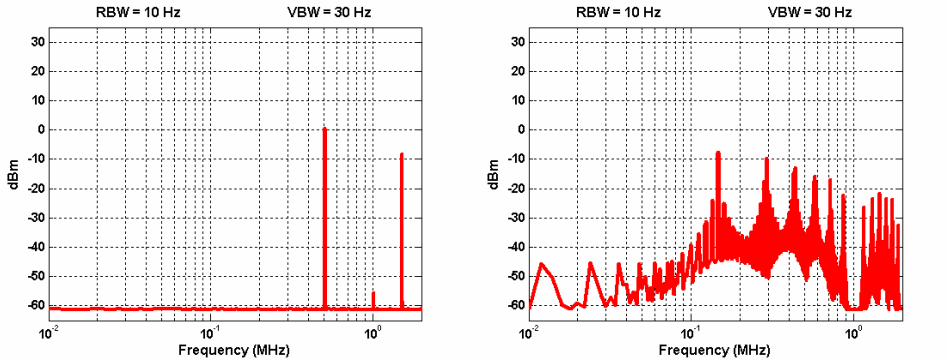

Figure 10. DSM Spectral Plots; 50% and 14.5% Duty Cycle

Note that the frequency components of the 37/256 duty cycle DMS is a mixture of 1MHz/6 and 1MHz/7. (256/37 = 6.92)

One chief problem with this solution is that no micro-controller comes with this type of hardware. It could be built with programmable logic or implemented with software. The Cypress Semiconductor CY8C27443 programmable system on a chip has eight PWMs, each capable of running with a clock frequency as high as 48MHz while using zero CPU overhead. Implementing a software DSM in the same device requires 100% of the CPU for a clock frequency of 1MHz.

Pseudo Random Modulation (PRM)

PWMS is a variation of pulse width modulation where the down counter is replaced with a pseudo random counter. A block diagram of a PRM is shown below.

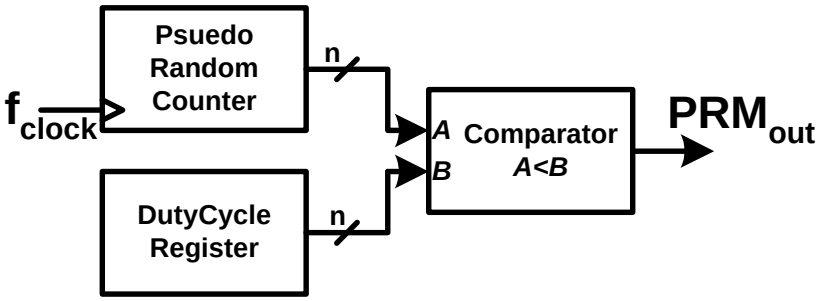

Figure 11. Pseudo Random Modulator (PRM) Block Diagram

The comparator is still high whenever the counter is below the duty cycle value It just that the counter no longer counts linearly. The output is still high the same number of counts it just that they are now (pseudo) randomly dispersed within the counter period. As with the delta sigma modulator the output frequency is not dependent on the counter period. The output frequency is higher but not as high as the DSM. The random nature of the output keeps it frequency to approximately half of what it would be for DSM. The equations for the duty cycle and output frequency are shown below.

$$dc = \frac{DutyCycleValue}{Period} \qquad f_{out} \approx \begin{cases} \frac{1}{2} dc \cdot f_{clock} & : dc \le 0.5 \\ \frac{1}{2}(1-dc) \cdot f_{clock} & : dc > 0.5 \end{cases}$$

The plots below are for two PRMs

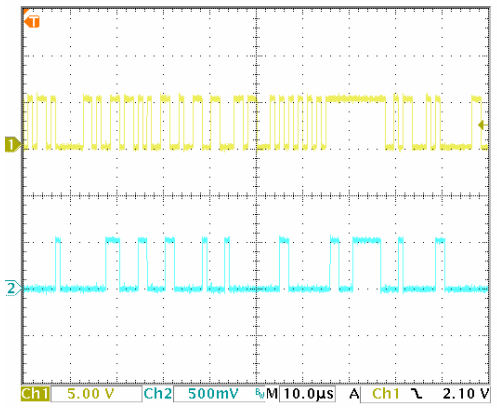

Figure 12. PRM Waveforms; 50% and 14.5% Duty Cycle

Both have an counter width of 256 and input clock of 1MHz. The top trace shows a PRM with its DutyCycleValue set to 128. This results in a very random looking output with a 50% average duty cycle and an output frequency of approximately 250kHz. The bottom trace is for a DSM with its DutyCycleValue set to 37. This results in a signal that is high 14.5% of the time. Its output frequency is around 172kHz. Being nearly random there is not much harmonic content at any particular frequency. A spectral plot of the two signals shown below confirms this.

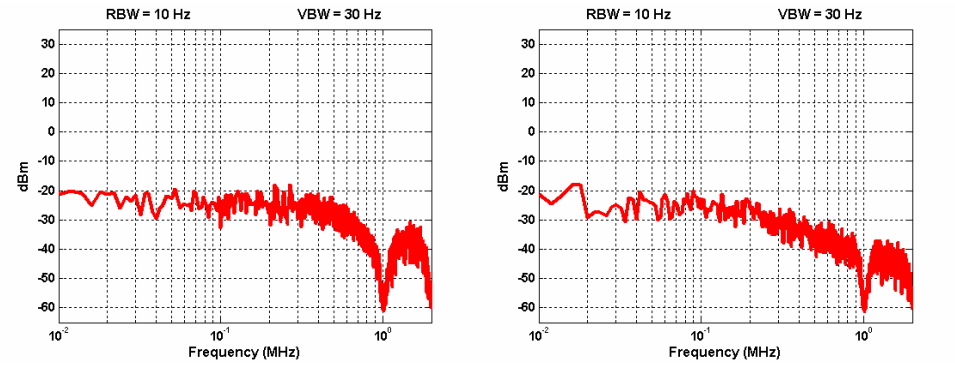

Figure 13. PRM Spectral Plots; 50% and 14.5% Duty Cycle

Although not pushing the harmonic frequencies as high as DSM there are advantages to the random nature of the output. Like the PWMs Cypress offers these types of modulators on their line of Programmable Systems on a Chip. The PSoC CY8C27443 is capable of implementing up to eight of these modulators.

Dithered Pulse Width Modulators. (DPWM)

So far three different modulation techniques have been shown. The second and third reduce the effects of harmonics by pushing tem up in frequency making them easier to remove. They do this by increasing the output frequency. This can be a problem where there is cost to high frequency switching. It may be that the components cannot switch above a certain frequency or there is a energy loss associated with switching. (Battery Chargers are a good example. There is loss every time the power FET is switched.). A fourth option is randomize the clock feeding a PWM. A block diagram of a DPWM is shown below.

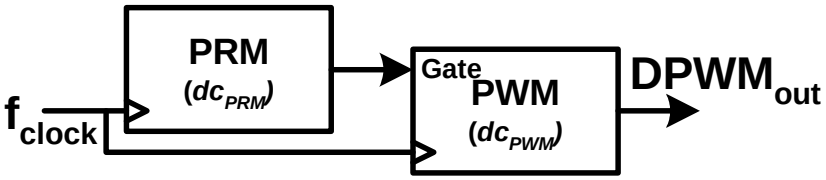

Figure 14. Dither Pulse Width Modulator (DPWM) Block Diagram

Normally the output frequency of a PWM is the clock divided by the period. 1MHz clocking an 8 bit PWM has an output frequency of 3.9kHz. Building a dither PWM requires using the gate input of the PWM. When the gate is high the PWM clock is enabled and it generates an output frequency of 3.9kHz. When the gate is low the clock is disables and the output frequency is zero. If the gate input has some duty cycle then the output is proportional to it. The equations for the duty cycle and output frequency are shown below.

$$dc_{PRM} = \frac{DutyCycleValue_{PRM}}{Period_{PRM}} \quad dc_{PWM} = \frac{PulseWidth_{PWM}}{Period_{PRM}} \quad f_{out} = dc_{PRM} \frac{f_{clock}}{Period_{PWM}}$$

For a 2MHz clock and a pseudo random duty cycle of 50%, the eight-bit PWM has an average output frequency of 3.9kHz (2MHz* ½*/256). It is average because the random nature of the PRM output causes a fluctuation frequency. For these particular parameter the change in output frequency is +/- 10%. This is shown in the waveforms below.

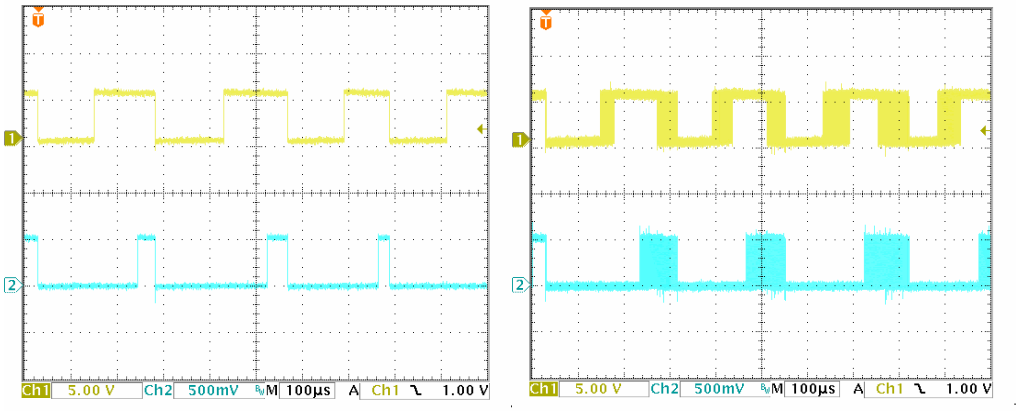

Figure 15. DPWM Waveforms; 50% and 14.5% Duty Cycle

With this much frequency shift the spectrum will no longer be harmonics at fixed frequency. This dither effect smears them out over a wider area. This is shown in the spectral plots below.

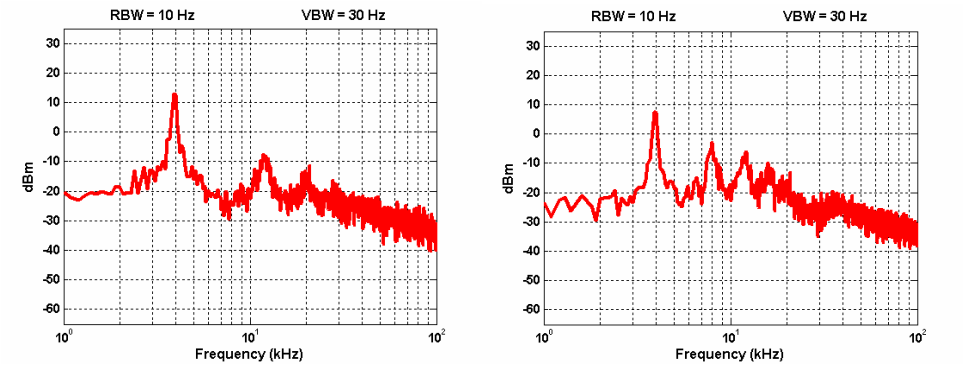

Figure 16. DPWM Spectral Plots; 50% and 14.5% Duty Cycle

Comparing these plots with those for a regular PWM (figure x) the peak harmonic noise is down maybe 3dB. And the other harmonics are significantly reduced but it comes at the expense of a overall noise florr. Another feature is that only changing the PRM duty cycle allows for fine tuning of the average output frequency. For example, the average output frequency is 3.9kHz. Changing the PRM duty cycle to 64% shifts the average output frequency to 5kHz

DACs are precious resources that are frequently completely used. It may become necessary to fabricate your own with on chip digital resources or firmware. Four examples of single bit DAC have been shown. The decision on which to use is dependent on your unique system requirements and available resources.