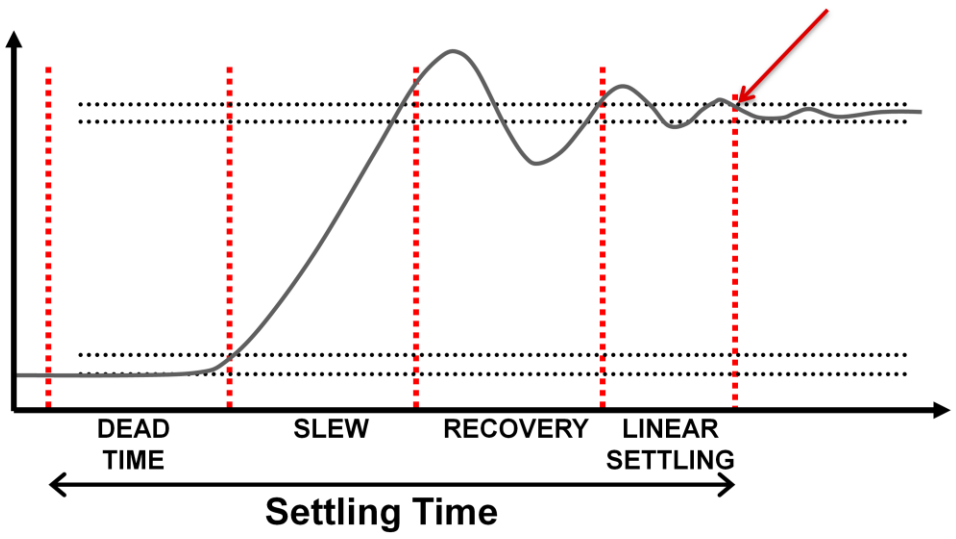
AC Specifications of Precision DACs

PRECISION Data Converters

Texas Instruments

Hello, and welcome to the Texas Instruments Precision overview of AC specifications for Precision DACs. In this presentation we will briefly cover the three most important AC specifications of DACs: settling time, glitch, and noise.

Settling time



Texas Instruments

To begin our discussion, let's look at settling time. The characterization of DAC settling time is consistent with operational amplifier settling time specifications, which should make sense as the output op-amp included in the DAC will dominate the settling time figure. Op-amp settling time is usually referred to as output settling time while a DAC specification is described as input-to-output settling time.

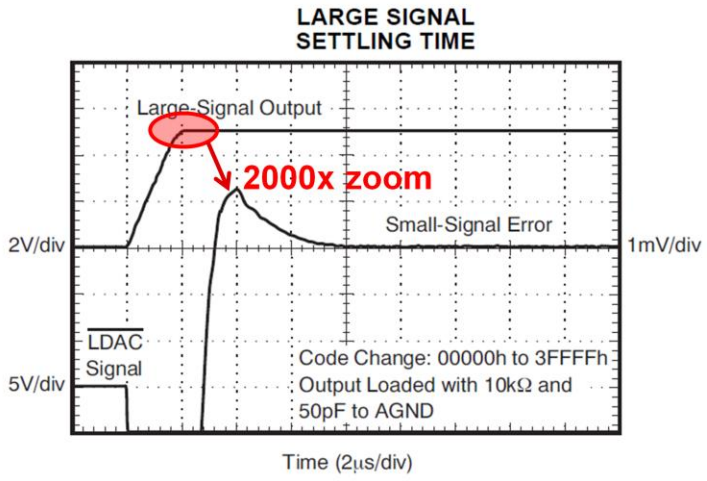
As is illustrated on this figure the difference between output settling time and input-to-output settling time is that we include a time period called 'dead time' in the input-to-output setting time figure. Dead time is specified as the time it takes for the output to rise to 10% towards the desired analog output value from the time the digital code was applied to the DAC. Dead time is caused by the time required for the DAC to latch the incoming data word, perform the required switching, and for the analog signal to propagate to the output.

Slew time takes over where dead time left off and is the time required to transition from 10% of the desired analog output to 90%. This specification, along with the remaining settling time figures, are dominated by the output amplifiers characteristics.

After slew time we observe recovery time and linear settling time - usually referred to as large signal settling time and small signal settling time respectively. At the **moment** the output passes inside of a pre-determined error band and does not leave the band again the output is said to be settled and at it's final value.

In many applications analog settling time is the dominant figure to consider for a DACs update rate, even if the serial interface is capable of running faster.

Settling time in real devices



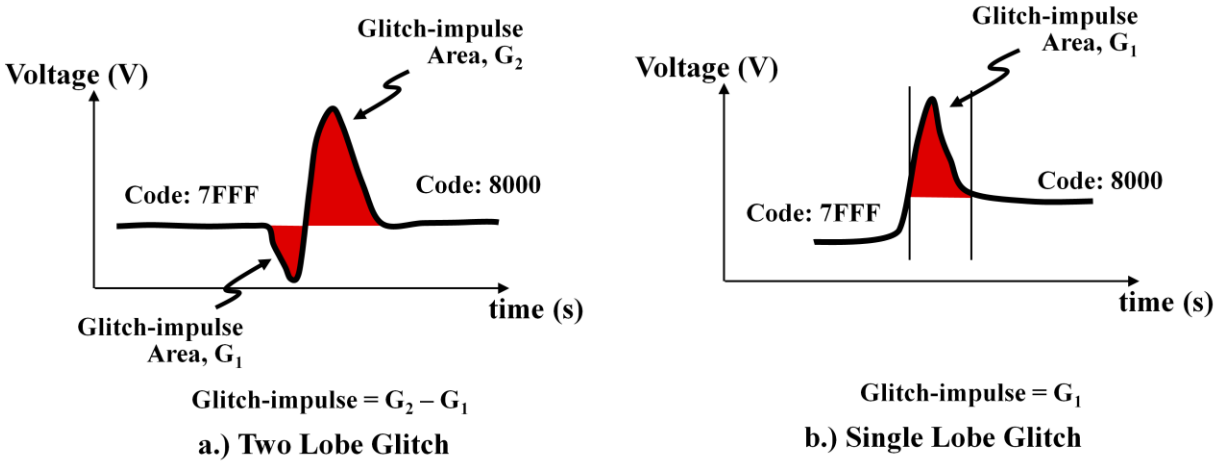
DYNAMIC PERFORMANCE[2]			
Settling time	To ±0.003% FS, R_L = 10kΩ, C_L = 50pF, code 04000h to 3C000h	5	µs

Texas Instruments

This is a capture taken from the DAC9881 datasheet. The settling time measurement is made from when the LDAC signal has fallen low and the input data word begins to propagate through the system. Notice that the dead time is very short, large signal settling time is pretty well behaved, and the settling time is primarily spent in the linear settling or small signal settling region and slew time.

Remember to pay attention to how the datasheet **electrical characteristics** table shows the settling time measurement being taken. In this case, the DAC9881 is shown to settle to 0.003% FSR with a specific load of 10kOhms in parallel with 50pF and over a transition from code 0x04000 to 0x3C000. This is a large step size and is intended to be a worst-case figure. The settling time figure will look much better across smaller code transitions as larger step sizes generally take longer to settle.

Glitch



Related to settling time is another specification called glitch or glitch area. While the settling time specification is dominated by the output buffer the glitch specification is dominated by the DAC itself and glitch will make contributions to settling time. Glitch impulse area is defined as the area associated with the overshoot or undershoot created by a code transition, and is generally quantified in Volt-seconds. There are two different types of glitches, single-lobe and double-lobe, which are mostly dependent on the DAC architecture. The DAC R-2R architecture produces the largest glitch in terms of amplitude and duration.

Glitch sources

- Two main sources
  - Switches switching – Charge injection from the switches changing states
  - Switch synchronization – the switches will not change at precisely the same time
- Major Carry Transition
  - Code changes that cause a majority of switches to transition in an R-2R or MDAC
  - 011 → 100
  - 1000 → 0111

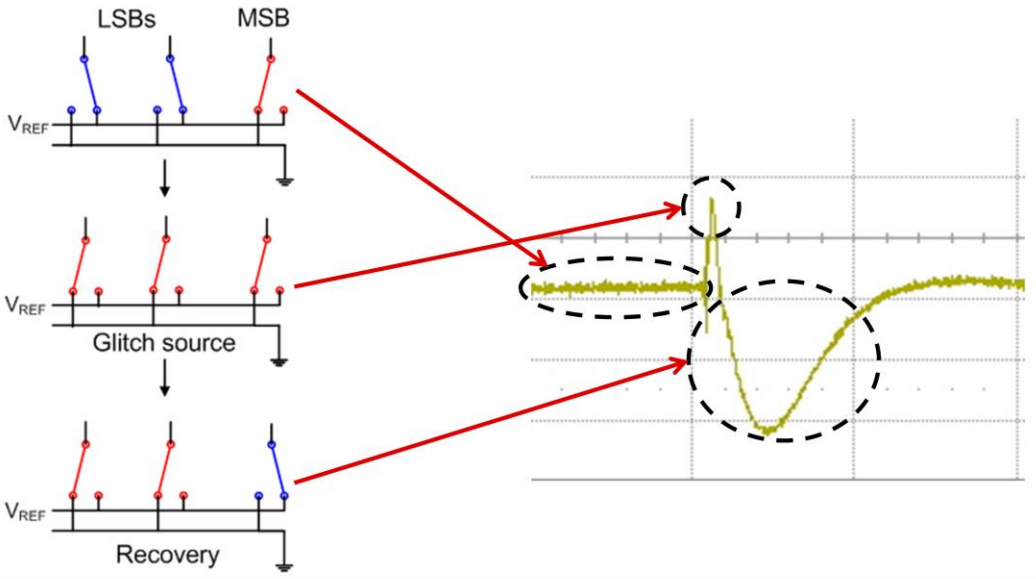
Texas Instruments

There are two primary sources of glitch, both relating to the switches used inside the DAC. First, when the switches transition the charge injection associated with the switches themselves will contribute a glitch at the output. Second, the switches will not switch at exactly the same time which will lead to larger glitch impulses.

Glitch impulse area is heavily dependent on the DAC architecture. String DAC architectures feature fewer simultaneously switching nodes and will therefore exhibit smaller glitch area R-2R DACs. R2R DACs may switch many switches in parallel during code transitions. A worst case scenario arises when every switch transitions in an R-2R DAC. The R-2R architecture displays large glitch areas at major-carry transitions - a single-code transition that causes the most significant bit (MSB) to change because of the transitioning lower bits (LSBs).

Some examples would be moving from binary 011 to 100 or 1000 to 0111. Let's look at an illustration of the first example.

Major carry transition



Texas Instruments

In order to show more detail, let's look at each switch's transition and relate it to what is observed on the output. The R-2R structure is shown in three states here.

The **first** switch state is the initial setting of the DAC. At this point the output has already settled and we have just written the new value.

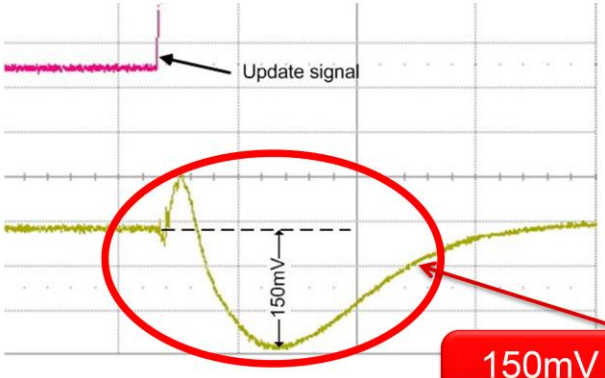
In the **second** state, the DAC enters a major carry transition where all of the switches are briefly connected to GND due to digital propagation delay of the most significant bit relative to the lease significant bits. This may be different from device to device.

Then in the **final** state, the DAC switches settle out and charge is injected from the switches. Following this is the recovery time or small signal settling time.

Code-to-code glitch

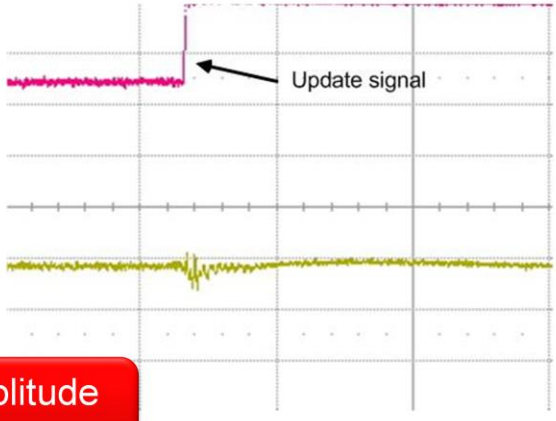
0x1FFFF to 0x20000

(major carry transition)



0x20000 to 0x20001

(non-major carry transition)



150mV amplitude glitch

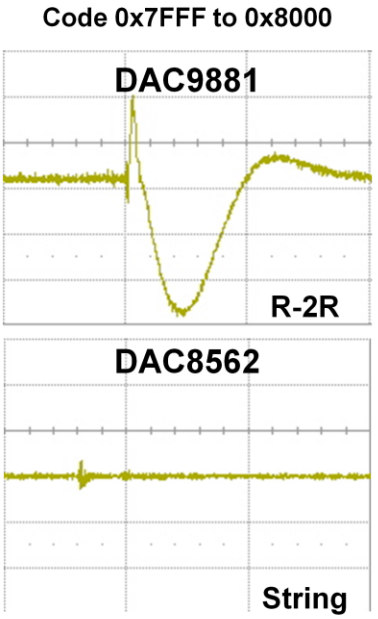
Texas Instruments

Comparing side by side a major carry transition vs a non major carry transition, we can see how large glitch is a result of a major carry transition – **on the order** of 150mV. Both pictures on this slide show the DAC9881 stepping up one code. The picture on the left is crossing over a major carry code transition while the picture on the right is not.

Glitch vs. DAC structure

- R2R DAC – DAC9881
  - Simultaneous switching
  - Synchronization
  - Large glitch amplitude
  - Settling of Internal Amplifier
- String DAC – DAC8562
  - Simple switch network
  - Tap into resistor string
  - Single Lobe glitch
  - Small glitch amplitude

Graphs: 500ns/div 50mV/div

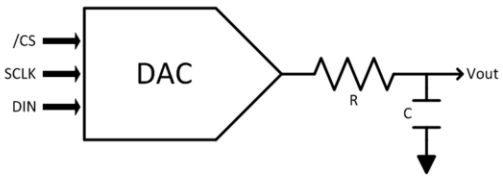


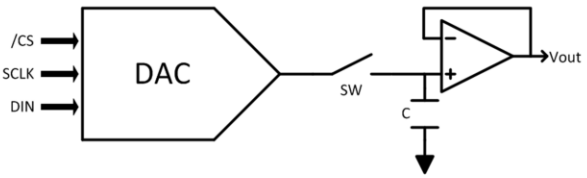
Texas Instruments

So far in our glitch discussion we've been looking at the R-2R DAC architecture because it's much easier to observe glitch area with an R-2R DAC. When we compare the R-2R glitch side by side with the string DAC we observe noticeable improvements.

Glitch reduction techniques

- RC filter
  - Easy to implement
  - Not a complete removal of the glitch
  - Decrease glitch amplitude
  - Increased settling time
- Switch cap solution
  - Removes vast majority of glitch
  - Strict timing
  - Difficult to implement
  - Slower update rates

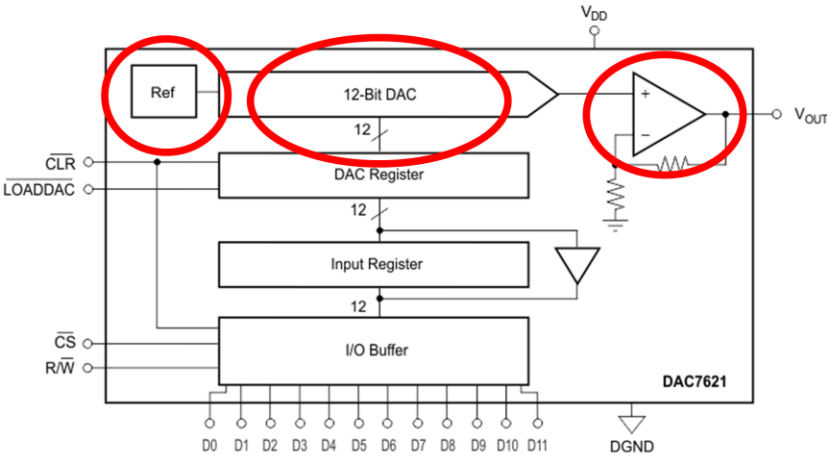




There are two solutions to glitch reduction, and they are summarized here side by side to assess the pros and cons. Depending on the application requirements, a simple RC filter may do the trick. If the system needs to maintain good settling time and the RC filer approach is therefore unacceptable, a sample and hold circuit is the best solution. Of course, the other option is to steer clear of R-2R DACs and look towards a string DAC solution to avoid large glitch impulses, but doing so may force a tradeoff in other specs.

DAC output noise sources

- Internal resistor network
- Output amplifier
- Reference voltage



Texas Instruments

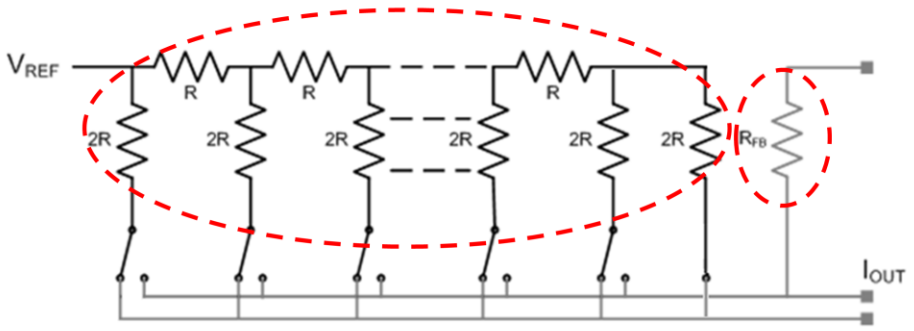
Next, we will talk about noise. As in almost any device, there is going to be noise associated with DACs. In noise conscious applications the goal is to minimize the DAC noise as much as possible to achieve optimal performance. In order to do that we need to look at the three major sources of noise contribution: the DAC's **internal** resistor network, the output **amplifier** - which may be integrated - and the **reference** voltage – also possibly integrated.

First, lets examine the resistor network.

Internal resistor network noise

- Thermal noise
  - K = Boltzman constant = 1.38e-23
  - T = Temperature = 300 K for room temperature
  - R = Equivalent resistance of resistor network
  - B = Bandwidth

$$e_n = \sqrt{4 \cdot K \cdot T \cdot R \cdot B}$$

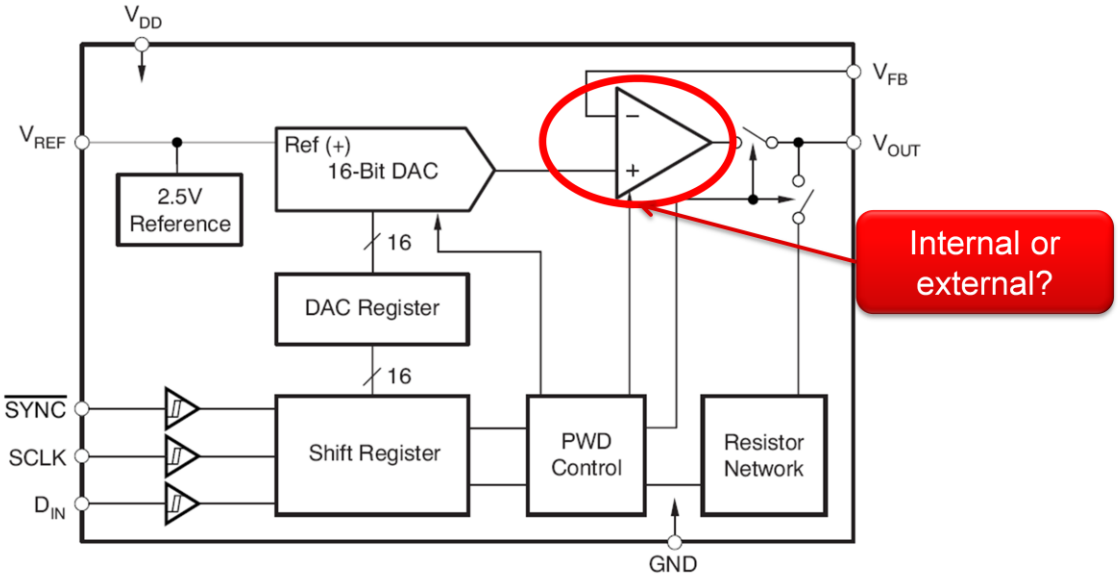


Texas Instruments

For an example calculation, let's look at an unbuffered R-2R MDAC, which is simply a resistor ladder without an output amplifier. The easiest way to think of the resistor ladder when analyzing noise is as a single equivalent resistor which changes value as you change code. At full scale code, all the switches are switched to the IOUT node, basically creating a resistor in series with the reference and IOUT. If you were to look back into the DAC from the Iout node, you will see an equivalent resistance equal to the entire resistor chain. TI datasheets often specify this impedance in the electrical characteristics table as the reference input impedance. Once you have the equivalent output impedance at IOUT, you can calculate the resistor noise, also **thermal noise**, from the network using the resistor **thermal** noise equation shown here.

One subtle but important note is that there are two resistances of interest here. First, there is the equivalent resistance of the resistor **network** which is code dependent. Second, you have the feedback **resistor** which is used in combination with an output amplifier to create an output voltage. When the DAC is in a gain of -1 The value of the feedback resistor is equal to the DAC resistor network at full scale code. Both of these resistances need to be accounted for when calculating the thermal resistor noise of the system.

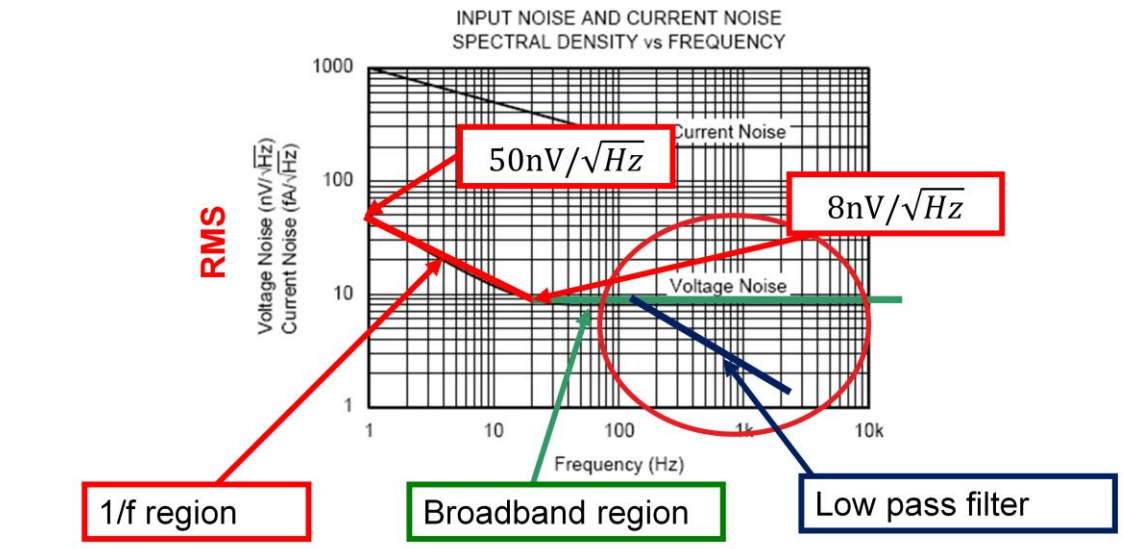
Output amplifier noise



Texas Instruments

Our second source of noise is the DAC's output amplifier. This output amplifier may be included internally in the DAC or you may have the option to choose your own external amplifier to meet your system needs.

Output amplifier noise



Source: Art Kay, Op Amp Noise 2006

Texas Instruments

Examining op amp noise has four basic components, 1/f voltage noise, broadband voltage noise, resistor voltage noise, and current noise. Current noise and resistor noise have a lot to do with implementation and less to do with properties of the amplifier itself.

For our purposes we'll only consider the 1/f and broadband noise component since these are more intrinsic values to the operational amplifier and will help us select our amplifier intelligently. The key thing to illustrate here is that in a typical application our focus will be on DC operating conditions where the bandwidth is limited and the broadband region will not contribute significant noise.

Output amplifier noise

- For external output amplifiers:
  - In general, search for amplifiers with low 1/f noise
  - See the TI Precision Labs video series on Op-Amp noise
- For internal output amplifiers:
  - Usually a number provided in the electrical characteristics table
  - Sometimes a figure to illustrate the noise spectrum

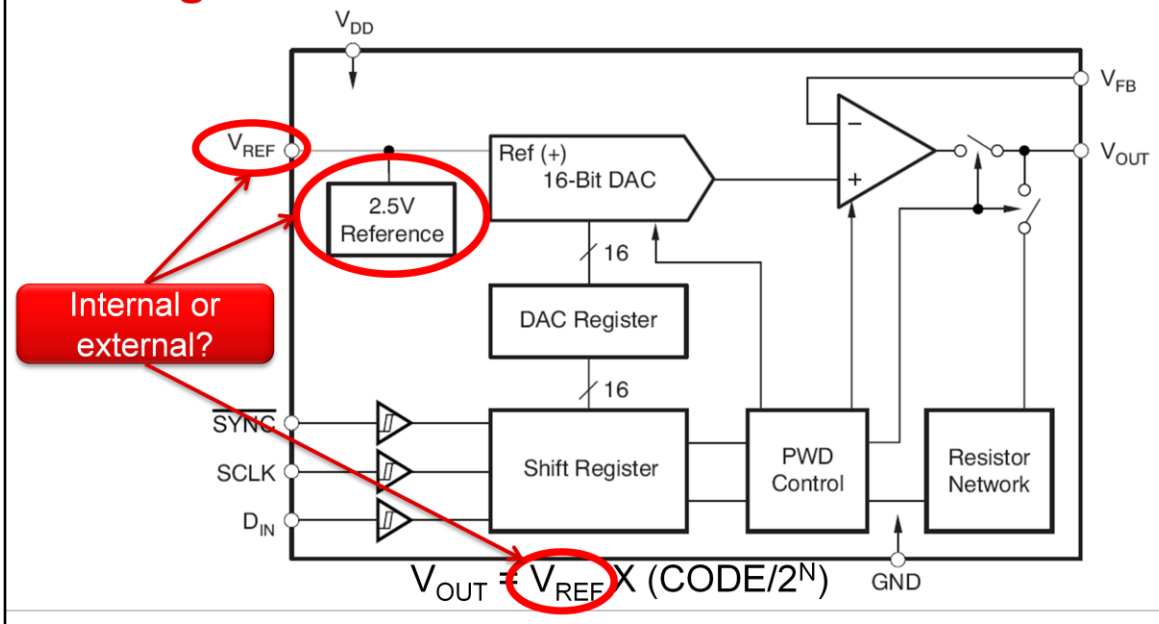
AC PERFORMANCE[2]			
DAC output noise density	T_A = 25°C, at mid-scale input, f_{OUT} = 1 kHz	90	nV/√Hz
DAC output noise	T_A = 25°C, at mid-scale input, 0.1 Hz to 10 Hz	2.6	μV_{PP}

Texas Instruments

As a best practice when your amplifier is external, search for devices with low 1/f noise. And when it comes time to do noise estimations for your design, check out the TI Precision Labs video series on Op-Amp noise for guidance on how to calculate your amplifier noise.

When the amplifier is internal to the DAC, most of the work is already done for you and the noise is characterized in the DAC datasheet. You can generally find a noise number in the electrical characteristics, and a plot of the noise spectrum with the other graphs further in the document.

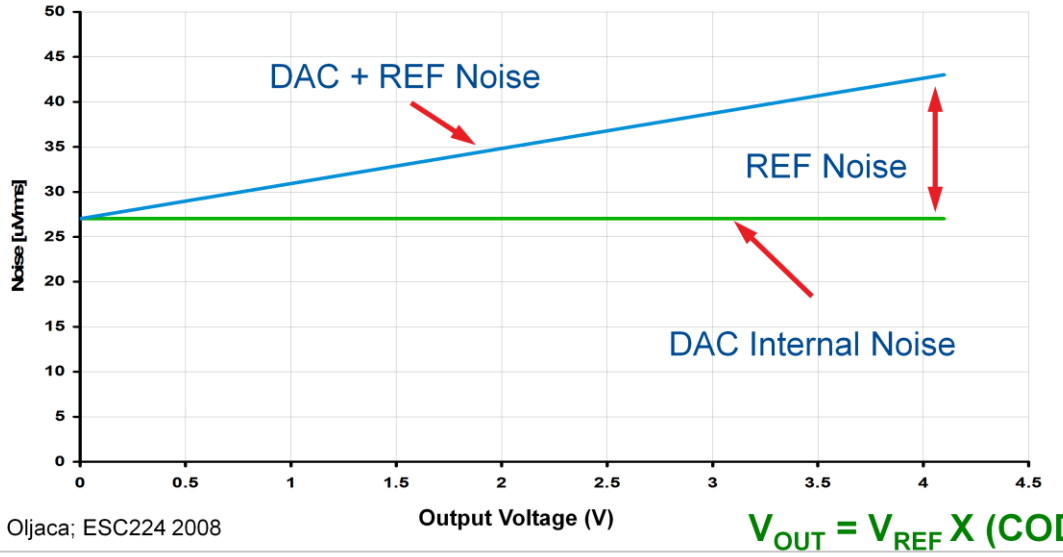
Voltage reference selection



Texas Instruments

Last, we need to look into the reference voltage. Looking back at the transfer function for the DAC, we see that the noise and gain of our DAC is directly dependent on the reference voltage. Therefore, it is very important to have a few quick design steps to optimize the design of the reference voltage.

Voltage reference noise contribution



Source: Miro Oljaca; ESC224 2008

$V_{OUT} = V_{REF} X (CODE/2^N)$

Texas Instruments

The initial accuracy and output noise level of the voltage reference is directly coupled into the DAC. The accuracy of the voltage reference is a DC phenomena, however the output noise is an AC event. As the output voltage of the DAC increases, per programmed input code, the voltage reference noise level increases in magnitude.

Keep in mind there are other sources that could also have an impact on the noise such as power supply transients and the cleanliness of the ground layer. That is why well regulated supplies and a proper layout are a major focus of every design.

In summary, we learned about the three major AC specifications that are relevant to Precision DACs: settling time, glitch, and noise. We learned about single and double lobe glitches and why they exist. And we also covered the three major noise contributors: the resistor network, output amplifiers, and reference voltages.

Thank you for watching this video on AC specifications for precision DACs. Please watch our other videos on precision DACs to learn more.