BY BONNIE BAKER

Get more accuracy from your DAC

With a DAC, you calibrate by initially determining the code-to-voltage error at one-third of the output range and again at two-thirds of the output range. The range between one-third full-scale and two-thirds full-scale avoids the output amplifier errors near the power-supply rails. The calibration of the offset and gain-error correction is a matter of simple algebra, where $V_{OUT}=a+bV_{IN}$. In this formula, a is the offset error, and b is the gain error. You can accomplish these calibrations in the digital domain with the help of an ADC that is more accurate than a DAC's target specifications. This calibration technique is effective with offset and gain errors greater than 2 LSBs (least significant bits); however, the converter's quantization errors limit this approach.

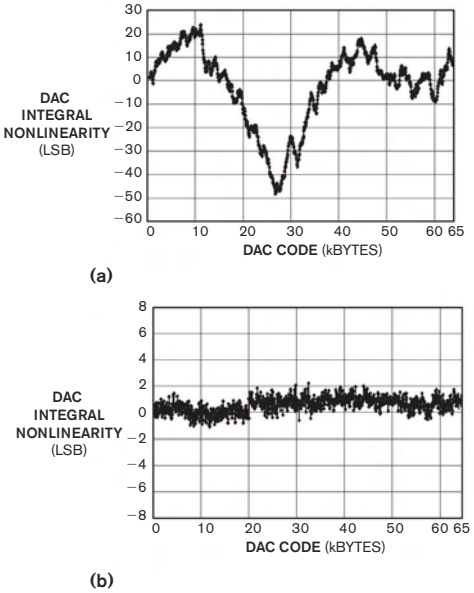

Figure 1 The INL (integral nonlinearity) of a 16-bit string DAC can vary across 10s of codes (a). A correction step of 64 LSB (1024 of the 65,536 points) reduces the INL error to less than ±3 LSB (b).

A more challenging DAC-calibration activity is to adjust the linearity of the converter's entire output range. Once again, you need an ADC that has four times the resolution of the DAC. The temptation is to calibrate every code of the DAC. This strategy may be acceptable for converters with 8, 10, 12, or 14 bits of resolution. This environment has fewer DAC codes to calibrate, the memory requirements are lower, and the accuracy of the calibrating ADC is not as demanding, allowing faster analog-to-digital conversion. For DACs with resolution of more than 14 bits, the total number of codes becomes unmanageable for processors or processor memory. Additionally, you need to use a slower ADC with higher accuracy, such as a delta-sigma converter. Higher cost and lower speeds encourage you to decide that linearizing every code for higher bit DACs is not worth the bother.

An effective alternative to linearizing every DAC code is to select several small groups of codes. The plot in **Figure 1a** shows an example of the INL (integral nonlinearity) of a 16-bit string DAC. The 16-bit string DAC comprises a string of 2^{16} resistors. The universal formula for calculating any DAC-correction code is $DAC_{COR}=INL_V+(INL_V-INL_W)(v-x)/(v-w)$, where INL_V and INL_W are the INL errors of the v and w code, and x is a code between codes v and w. If (v−w) is equal to an integer that is a power of two, you can implement the division with right shifts, reducing the processor-calculation time and complexity. The plot in **Figure 1b** illustrates the benefit of this linearization technique using 1024 code groupings with 64 codes per group.

This technique best suits DACs that are monotonic and that have INL error of more than ±8 LSB. Additionally, you must exercise care when selecting the size of the code sets. If large, sudden jumps occur from one code to the next, as may be the case with R2R architectures, this technique may prove to be counterproductive rather than an improvement of DAC performance. The string-DAC topology best suits this calibration technique because it is inherently monotonic (a requirement for this technique), and jumps from one code to the next are relatively small compared with those of other DAC topologies. EDN

REFERENCE

Anderson, Russell, and Michael Gurevich, "MSC1211/12 DAC INL Improvement," Application Report SBAA112, Texas Instruments Inc, March 2004, focus.ti.com/lit/an/sbaa112/sbaa112.pdf.

Bonnie Baker is a senior applications engineer at Texas Instruments and author of A Baker's Dozen: Real Analog Solutions for Digital Designers. *You can reach her at bonnie@ti.com.*