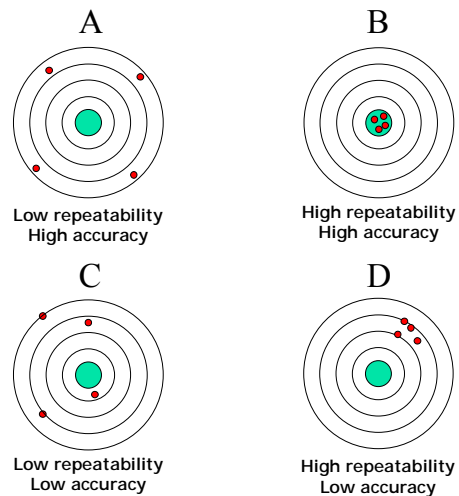


Analog-To-Digital Converters
Part 1 Of 6: Accurate And Repeatable Bits Versus Actual Bits Of ADCs
by Bonnie C Baker, Texas Instruments

Is it possible to make a quick decision about which analog-to-digital converter (ADC) you will use for your signal-level application? Does the title at the top of the data sheet have enough information to determine how many usable bits the converter has? By definition, signal-level applications operate between dc to approximately a few megahertz, depending on the converter and input signal bandwidth. With this application type several specifications come into play as you choose your converter.

The data sheet title will only take you so far: it will tell how many bits the converter will output, but it will not tell you how many of the converter bits are usable. You need to peel the onion by going into the data sheet details to get to the bottom of this *usable bit* mystery. If you are selecting a successive approximation register (SAR) ADC, you need to pay attention to dc as well as the ac specifications before you make your final determination of how many bits the ADC has. This TechNote will help you sift through these issues and quickly select the correct converter for your application.

Use Dc Specifications Of SAR Converters To Determine Accuracy



**Fig. 1: Conversion Precision From Two Perspectives:
Accurate But Not Repeatable (A); Repeatable But Not Accurate (D)**

When picking a SAR converter for your application the first step is to determine the number of usable bits of the converter in question. Usable bits are accurate in that the ADC consistently reports the correct digital output for the voltage in (see Fig. 1B). For this decision you will initially use the dc specifications, which will help you understand the accuracy of the converter (Figs. 1A and 1B). The dc specifications help determine the total-non-adjustable-error of the converter (TUE) which defines the accuracy of each bit from multiple averages of a dc input signal. From there, the ac specifications will assist in

determining the noise in the conversion process, which in-turn defines the repeatability of each bit. The signal-to-noise ratio (SNR) of the converter will assist you in defining the repeatability of each bit from conversion to conversion (Figs. 1B and 1D). By combining the dc and ac specifications you can determine the level of accuracy and repeatability of the converter, and find a converter that fits the criteria described in Fig. 1B.

You can take the TUE specification directly off the data sheet (when the manufacturer includes it) or by using the offset-error, span-error and integral non-linearity (INL) specifications. Figs. 2, 3 and 4 graphically describes these errors as they relate to the transfer function of a 3-bit converter.

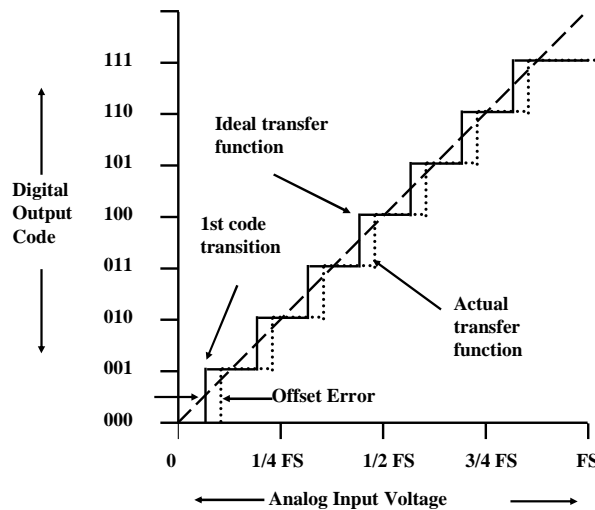


Fig. 2: Offset Error Versus An Ideal Transfer Function Of 3-Bit ADC

Offset error (see Fig. 2) is the difference between the first measured transition point and the first ideal transition point. You measure and then calculate the offset error of your converter with the formula:

$$\text{Offset error} = (V[0:1] - 0.5 (V_{\text{ILSB}})) \div V_{\text{ILSB}}$$

where,

$$V_{\text{ILSB}} = V_{\text{REF}} \div 2^n = \text{ideal LSB voltage size}$$

$$V[0:1] = \text{analog voltage of first transition}$$

$$V_{\text{REF}} = \text{full-scale voltage}$$

$$n = \text{number of converter bits}$$

You can correct the offset error of an ADC converter by adding or subtracting the offset error to every bit using the mathematics capability inside processor, or controller, software for this task.

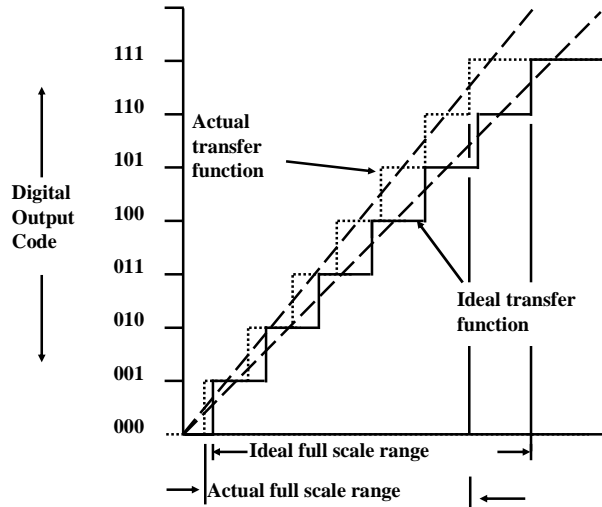


Fig. 3: Span Or Gain Error Versus Ideal Transfer Function Of 3-Bit ADC

Fig. 3 illustrates the concept of span or gain error. Span error is the difference between the ideal slope between zero and full scale (FS) and the actual slope between the measured zero point and FS. You zero out the offset errors with this error calculation and measure the last code transition at $1\frac{1}{2}$ bits below the FS input range of the converter. The calculation for span error is:

$$\text{Span error} = (V_{\text{REF}} - 2 V_{\text{ILSB}} - V[(2^n - 2):(2^n - 1)] - V[0:1]) \div V_{\text{ILSB}}$$

where,

$$V_{\text{ILSB}} = V_{\text{REF}} \div 2^n = \text{ideal LSB voltage size}$$

$$V[0:1] = \text{analog voltage of first transition}$$

$$V_{\text{REF}} = \text{full-scale voltage}$$

$$n = \text{number of converter bits}$$

You can correct the span error of an ADC by multiplying this calculated constant to every bit. You can once again use the mathematics capability inside processor or controller software for the task.

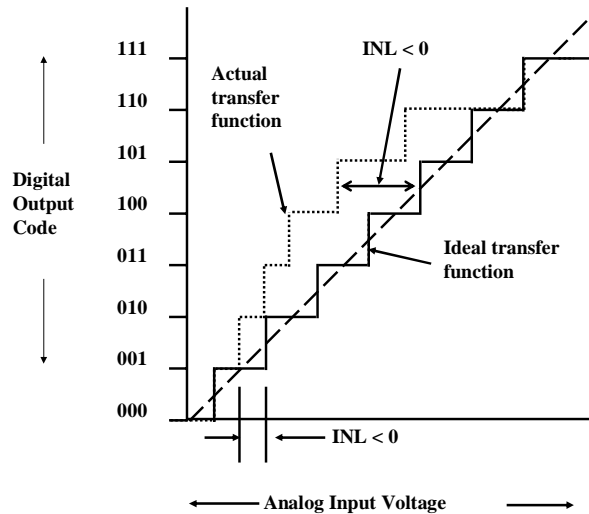


Fig. 4: INL: Maximum Deviation Between Actual And Ideal Code Transition Points With Offset And Span Errors Removed. Positive INL Signifies Transitions At Higher Voltages Than Ideal; Negative INL Means Lower Voltages Than Ideal

Fig. 4 illustrates the concept of INL, which is the maximum deviation of a transition point from the corresponding point of the ideal transfer curve, with offset and span errors zeroed. Unless you execute a total characterization of each code with each individual device, the controller or processor software cannot easily reduce INL errors. In data sheets, manufacturers define this specification using a best-fit transfer function or an end-point transfer function. You determine the best-fit transfer function with a least-squares-curve to fit to the transfer function.

If the TUE specification is not included in the product data sheet, you can calculate it as the square-root of the sum-of-the-squares of the offset error, span error and INL. For instance, with the following specifications for a specific 16-bit converter:

- Offset error = ± 2 LSB ($\equiv \pm 0.003\%$ FSR)
- Span error = $\pm 0.1\%$ FSR
- Integral non-linearity = ± 2 LSB ($\equiv \pm 0.003\%$ FSR)

The TUE of this converter is:

$$\begin{aligned} \text{TUE} &= \sqrt{(\text{offset error}^2 + \text{span error}^2 + \text{INL}^2)} \\ &= \sqrt{(0.003\%^2 + 0.1\%^2 + 0.003\%^2)} \\ &= 0.1\% \Rightarrow 9.97 \text{ bits}_{\text{rms}} \end{aligned}$$

Integral or differential non-linearity is more difficult to calibrate out of the transfer function of the converter because this error is ADC output code-dependant. If you just reduce the span error by using characterization and mathematics in the controller or processor software, you can see immediately the benefits to TUE specification:

$$\begin{aligned} \text{TUE} &= \sqrt{(\text{offset error}^2 + \text{span error}^2 + \text{INL}^2)} \\ &= \sqrt{(0.003\%^2 + 0.001\%^2 + 0.003\%^2)} \\ &= 0.0044\% \Rightarrow 14.5 \text{ bits}_{\text{rms}} \end{aligned}$$

The dominant term in the above formula was the span error. Adding or subtracting the offset error from the ADC output code will further reduce the TUE of the converter. You can see how you can improve TUE if you are willing to characterize the offset and span-error of the ADC, but in this example, reducing the span error gives a large improvement just by itself. You can reduce these errors in the software of your microcontroller or microprocessor.

The offset, span and INL specification for an ADC defines the accuracy of the device in that you can depend (on the average) on seeing the same output digital code for a given input voltage. If you want to know the accuracy of your converter, the specification or calculation of the TUE error is very useful. However, when these specifications are measured, the location of the bit-to-bit transition is an average of several measurements.

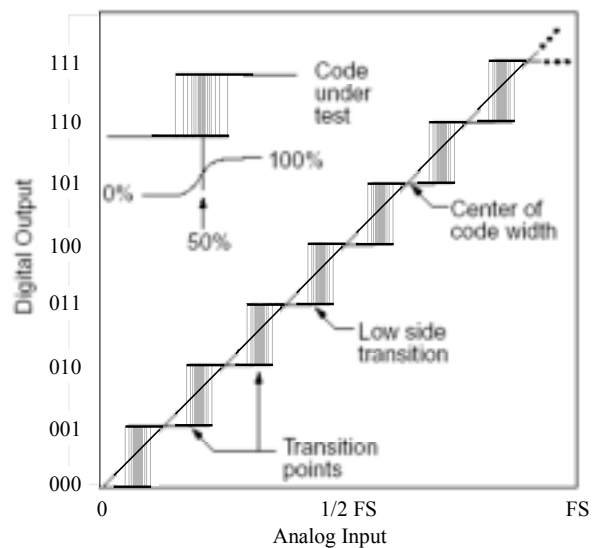


Fig. 5: Non-Ideal Transfer Function Of 3-Bit ADC Shows Transition Noise Of Every Code

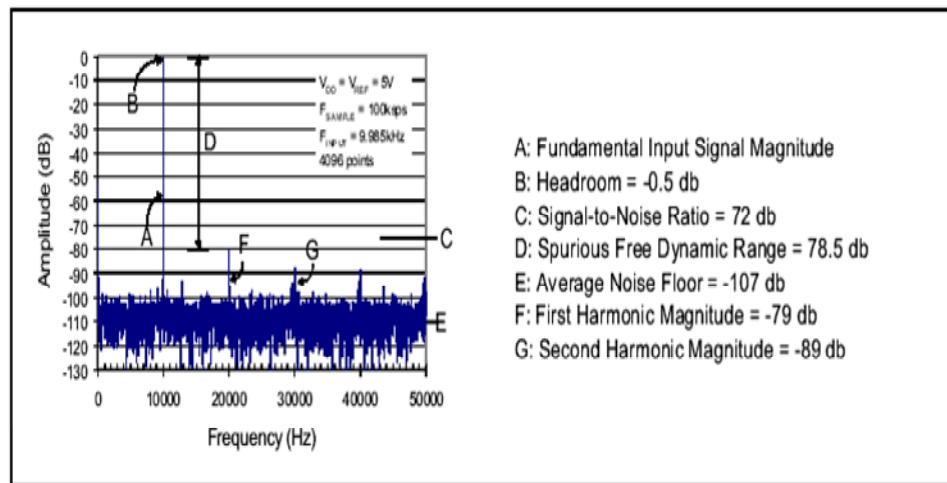
You might think that an ADC code-to-code transition is sharp, occurring at a unique, input voltage but, actually, the transition regions in the ADC-transfer function may vary widely. In fact, these regions may span across several digital-output codes. In Fig. 5 a transition point occurs when the digital output switches from one code to the next with respect to a specific analog input voltage. But, because of internal ADC noise, the transition point is typically not a single threshold from sample to sample, but rather a small region of uncertainty. Consequently you need to define the transition point as the statistical average of many conversions. Stated differently, the defined code-to-code transition is at the voltage input where the uncertainty of multiple conversions averages 50% of the time to one digital code, and 50% of the time to an adjoining digital code. Upon closer inspection the conversion transitions you collect appear to be noisy, with a Gaussian probability curve.

The offset, gain, and INL are the accuracy specifications for ADCs. Some manufacturers also call these the dc specifications. This is because these device tests use a dc-input

voltage for the conversions. But, these specifications do not tell you how repeatable the results are from conversion to conversion. They only tell you that, on the *average*, these errors will be no more or less than the minimum and maximum in your ADC manufacturer's data sheet. In order to describe the accuracy and repeatability of your converter, you need to combine the ac specifications with the dc specifications.

Use Ac Specifications Of SAR Converters For A Measure Of Repeatability

Within the ac specifications, you will find several lines of specs. The ac domain specifications, such as total-harmonic-distortion (THD), signal-to-noise ratio (SNR), spurious-free-dynamic-range (SFDR), signal-to-(noise + distortion) (SINAD), or effective number of bits (ENOB), provide information about SAR ADC repeatability. Fig. 6 shows these specifications using an FFT plot of 4096 data points taken from a 12-bit converter. The most useful specification that can provide repeatability information is the SNR. If you convert this to bits you will get $SINAD_{(bits)} = (SINAD_{(dB)} - 1.76 \text{ dB}) \div 6.02$, where $SINAD_{(bits)} = n$. This specification, combined with the dc specifications, will give you a strong feel for how accurate and repeatable the digital outputs of your converter really are.



**Fig. 6: Key Ac Characteristics Using FFT Plot:
Fundamental Input Signal (A); Signal Headroom (B);
SNR (C); SFDR (D); Average Noise Floor (E)**

Ideally, the SNR and SINAD of a converter is equal to $6.02n + 1.76 \text{ dB}$, where n is equal to the number of converter bits (this number is usually in the title of the data sheet). This theoretical noise is a consequence of the quantization noise inherent in the converter. In practice SNR is equal to $20 \log_{10} [\text{rms signal} \div \text{rms noise}]$, where rms means root-mean-square. For the FFT plot, SNR is equal to the square-root-of-the-sum-of-the-squares of all of the noise spurs in this plot, excluding dc and the harmonics of the input signal. The input analog signal in this test is sinusoidal. SINAD is equivalent to the square-root-of-the-sum-of-the-squares of the noise and harmonic spurs. In order to determine the rms

noise, many conversion results need to be collected. The FFT in Fig. 6 shows four thousand and ninety-six (4096) conversions at a data rate of 100 ksample/s.

For example, a 16-bit converter could have rms, ac specifications of:

$$\text{THD} = -90 \text{ dB}$$

$$\text{SNR} = 90 \text{ dB}$$

$$\text{SINAD} = 84 \text{ dB}$$

$$\text{SFDR} = -93 \text{ dB}$$

$$\text{* test conditions: } f_{\text{SAMPLE}} = 2 \text{ MHz, } f_{\text{IN}} = 0.5 \text{ MHz}$$

The figure of merit that determines the repeatability in terms of number of bits of this product is:

$$\text{SINAD}_{(\text{bits})} = (\text{SINAD}_{(\text{dB})} - 1.76 \text{ dB}) \div 6.02$$

$$\text{SINAD}_{(\text{bits})} = 13.66 \text{ bits}_{\text{RMS}}$$

You combine the two specifications, TUE and SINAD, using a square-root-of-the-sum-of-the-squares calculation. From such a calculation, the usable bits for the ADC in this article example are equal to 13.46 bits.

Conclusion

So the answer to the question, "How do I know that an ADC will give me an accurate, repeatable code?" can be found, with a little effort, from the converter product data sheet. For accuracy refer to the dc specifications: offset error, span error and INL. To validate the repeatability refer to the SNR ac specification. This information will arm you to tackle your accuracy and repeatability questions.

References

A Baker's Dozen: Real Analog Solutions for Digital Designers, by Bonnie Baker, Newnes-Elsevier, 2005 ISBN 0750678194

Anticipate the Accuracy of Your Converter, by Bonnie C Baker, EDN, March 18, 2004

What does the ADC SNR mean? by Bonnie C Baker, EDN, May 27, 2004

Using the Analog-to-Digital (A/D) Converter, Mitra, D'Souza, Cooper, AN546, Microchip Technology Inc.

About The Author

Bonnie Baker is a Senior Applications Engineer for Texas Instruments <http://www.ti.com> Tucson, Arizona, and has been involved with analog and digital designs and systems for nearly 20 years. In addition to her fascination with circuit design, Bonnie has a drive to share her knowledge and experience and has written over 250 articles, design notes, and

application notes. Bonnie Baker is the author of *A Baker's Dozen: Real Analog Solutions for Digital Designers* (Newnes, 2005). You can reach her at bonnie@ti.com

