Application Note, V1.1, Nov. 2010

AP32121

ADC measurement and specification

AudoNG, Audo Future and AudoMAX family

Microcontrollers

Never stop thinking

Edition 2010-12-06

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

AP32121		
Revision History:	2010-11	V1.1
Previous Version:	V1.0	

Page	Subjects (major changes since last revision)
1	Titel changed

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:
mcdocu.comments@infineon.com

Table of Contents **Page**

1 Introduction

This application note describes the measurement methods used for Analog to Digital Converters in Infineon's µ-Controllers. It covers both the definitions of static and dynamic error parameters and the test setup and algorithms used to specify the ADCs. The test conditions used during the characterization/validation of Infineon µ-Controller ADCs are also outlined.

This document is intended to help users of ADCs in µ-Controllers to understand the specification and to get the most out of the integrated ADCs.

2 Terms and Definitions

Quantization, Quantization Error	Quantization is the process where an analog signal is converted into a digital signal with a finite number of bits. The quantization error is the inevitable result of this process, even for an ideal ADC.
LSB	Least Significant Bit, expressed in voltage it is $(V_{ref+} - V_{ref-})/2^n$.
Transfer Curve	A Transfer Curve describes the input to output behavior of an ADC.
Adjusted Transfer Curve	In an Adjusted Transfer Curve, both the offset error and the gain error from a measurement are accounted for.
Transition Voltages	The Transition Voltages define the switching points from one ADC code to the next in a transfer curve.
Static Test	A test is defined to be static when the input signal to the ADC under test is slowly varied. The output results are the same as for a constant signal being input. Typically a ramp is chosen as the input signal for such a test.
Dynamic Test	A test is defined to be dynamic when the input signal varies in such a way that the dynamic errors of the ADC are revealed. Typically a sine wave is chosen as the input signal for a dynamic test.
Full Scale (FS)	The term Full Scale is used to define the highest possible digital output code of an ADC with resolution *n.*
Full Scale Range (FSR)	Full Scale Range is the range defined by the reference inputs of the ADC If the ADC has only a positive reference input and the negative reference input is tied to ground FSR = V_{Ref+}, otherwise FSR = $V_{Ref+} - V_{Ref-}$.
Resolution	The number of bits that the ADC is capable of converting an analog signal into. This is often a power of 2 and is denoted by *n* throughout this application note.

Throughout this document small letters indicate an array of data whilst capital letters are used for single values. For example, *tue* stands for the whole data array while TUEmin and TUEmax are the minimum and maximum values used for the specification.

3 Static Error Parameters

3.1 Ideal ADC Transfer Curve

There are two kinds of ideal ADC transfer curves. For the correct calculation of the error parameters of a real ADC it is important to know its corresponding ideal transfer curve. Figure 1 shows the uncompensated ideal ADC transfer curve of a 3bit AD converter. Here the code width for all codes is the same. This results in a quantization error from 0 to -1LSB.

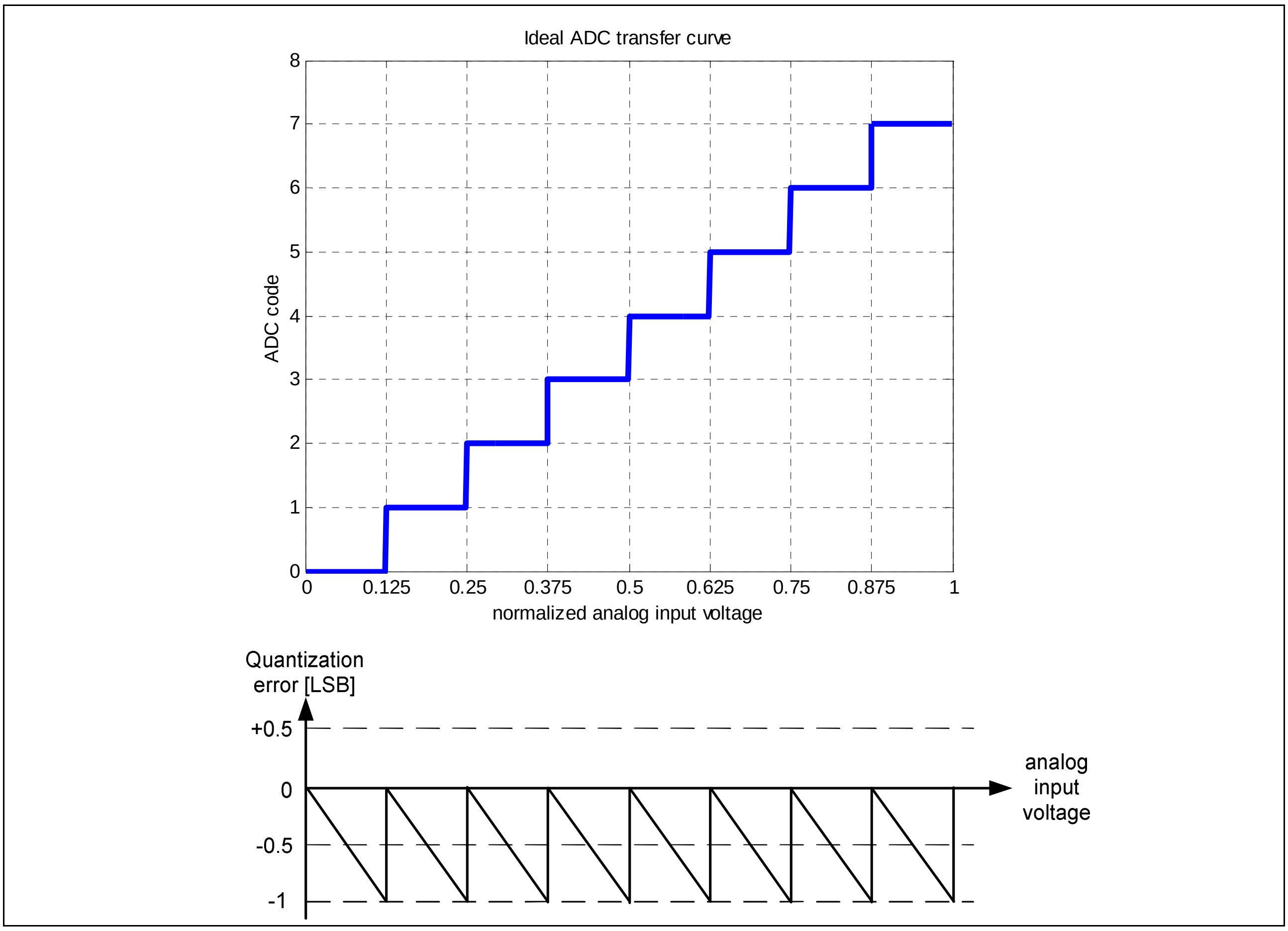

Figure 1 Uncompensated ideal transfer curve and quantization error of a 3 bit AD converter

Another way of definition is where the width of the first code is only ½LSB wide while the width of the last code is 1½ LSB wide. This type is often referred to as ½LSB compensated transfer curve and is shown in Figure 2. The resulting quantization error is now ±½ LSB.

The standard single ended ADCs in Infineon µControllers use the ½LSB compensated transfer curve while the fast differential ADCs use the compensated one.

Note: For the uncompensated transfer curve the midpoint of the analog input range occurs at code transition from $2^n/2$-1 to $2^n/2$ while for the compensated one the midpoint is located in the middle of code $2^n/2$. There are two 2^n codes but only 2^n-1 code transitions for both transfer curves.

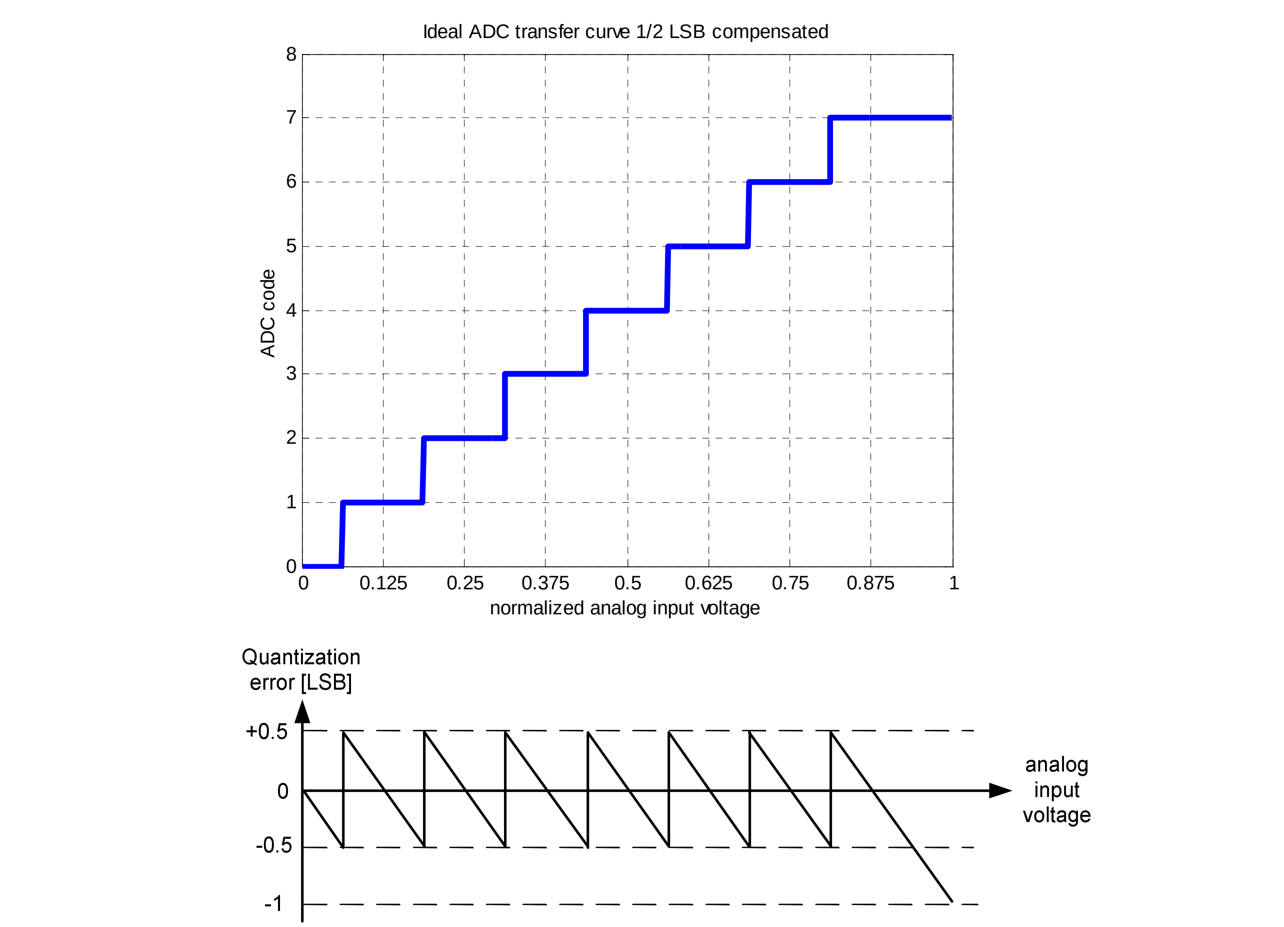

Figure 2 Compensated ideal transfer curve and quantization error of a 3 bit AD converter

3.2 Total Unadjusted Error

The transfer curve of a real ADC deviates from that of an ideal ADC. The difference between the measured transfer curve and the ideal transfer curve is the total unadjusted error (TUE). An example for a 3 bit ADC is shown in Figure 3. Note that the *tue* calculation is based on the code transitions, hence there are 2^n-1 *tue* values in a *tue* plot:

$$tue[i] = real\_code\_transition[i] - ideal\_code\_transition[i]$$

$$i = 0 \rightarrow 2^n - 1$$

The *tue* values are normally expressed in terms of LSBs of the converter. The minimum and maximum values of the *tue* array are valid for the specification of the ADC:

$$TUE\max = \max(tue)$$

$$TUE\min = \min(tue)$$

As the name implies, the TUE contains all the errors present in an ADC. These are offset error and gain error and linearity errors.

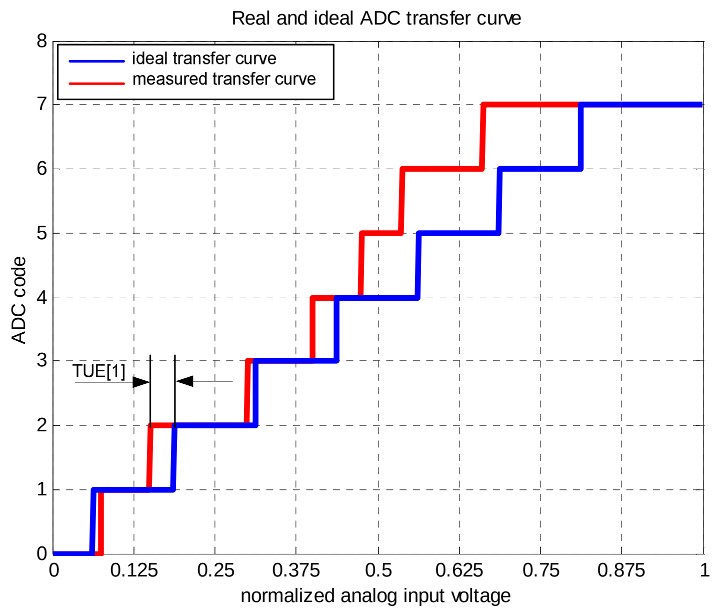

Figure 3 Total Unadjusted Error

3.3 Offset and Gain Error

For offset and gain error calculations two methods exist, the endpoint method and the best fit method.

Endpoint method

The endpoint method uses only the first and last codes for the calculations. The offset error is defined as the difference between the measured first code transition and the ideal first code transition. For the gain error a line through the first and the last code transition is built. The deviation of the slope of this line from the ideal slope defines the gain error.

Note: For the differential ADCs using the uncompensated transfer curve the offset error is defined in the middle of the transfer curve. This is because here the ADC should deliver the mid-code when its inputs are connected together. In contrast the single ended ADC should deliver code 0 when the input is connected to ground. For that reason the endpoint approach makes no sense for the offset calculation in the middle of the transfer curve.

Best fit method

The best fit method is based on a best fit line through all codes. For the compensated transfer curve the offset error is then defined as the deviation from this line to the ideal one at the lowest code. For the compensated one the deviation in the middle of the transfer curve is the offset error.

The gain error is defined as the deviation of the slope of the best fit line to the ideal slope. An example of a 3 bit ADC using the ½ LSB compensated transfer curve is shown in Figure 4.

All ADCs in Infineon µControllers are specified by the best fit method.

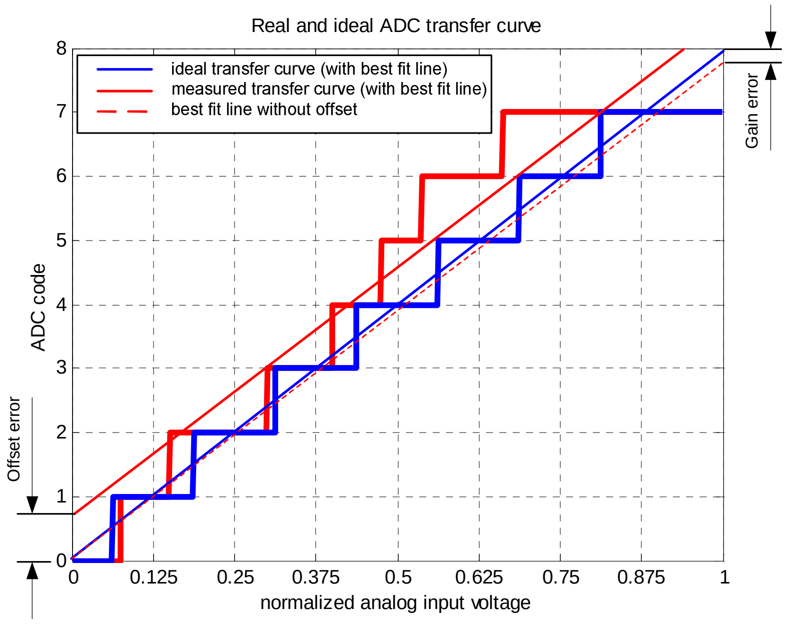

Figure 4 Offset and Gain Error

3.4 Linearity Errors

There are two kinds of linearity errors, Differential Non Linearity (DNL) and Integral Non Linearity (INL). Again, there are different definitions used in industry for both the DNL and for the INL. This can potentially lead to mis-interpretation of the specification.

Differential Non Linearity

The DNL is defined as the deviation of the real code width from the ideal code width (1LSB). The real code width is defined as real_code_transition[i] – real_code_transition[i-1]. As the first and the last codes do not have a predecessor respectively a successor, the width of these two codes is undefined and hence are not included in the DNL error. The DNL can be calculated out of the *tue*:

$$dnl[i] = tue[i] - tue[i-1]$$
$$i = 1 \rightarrow 2^n - 2$$

Note: The DNL array has only 2^n-2 values starting from 1 while the tue array has 2^n-1 values starting from 0.

The minimum and maximum of this array are valid for the specification:

$$DNL\max = \max(dnl)$$
$$DNL\min = \min(dnl)$$

The lowest code width is “0” and occurs if a code is missing. In this case the DNLmin error is -1LSB. Hence DNLmin has a lower boundary of -1LSB. DNLmax can exceed +1LSB and has no upper boundary.

Figure 5 shows an example of a 3 bit ADC where the code width of code 2 is greater then 1LSB hence having a DNL error greater then 0.

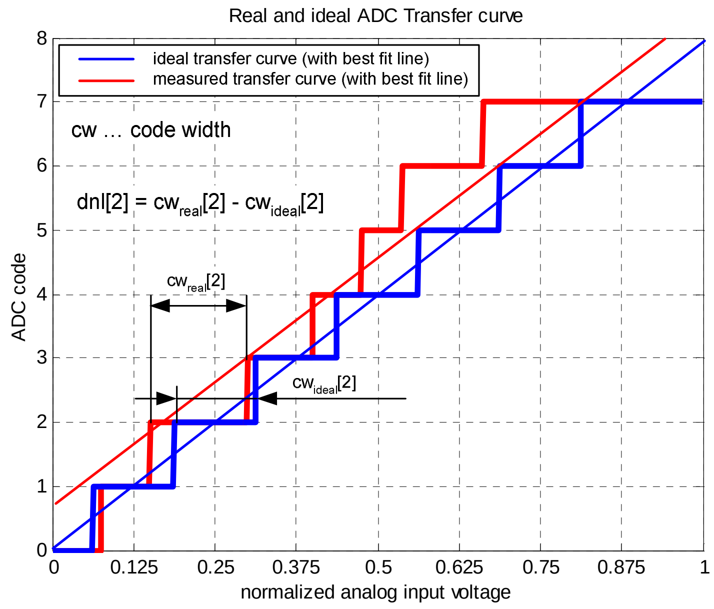

Figure 5 Differential Non Linearity, code transition testing

In the above definition the **code transitions** are used for the DNL specification.

Another method of DNL calculation is where the **code mid-points** are compared to the ideal code mid-points. This method may hide certain ADC problems. These problems can easily be seen in the example in Figure 6. In this example every second code is too wide and in between every second code is too small. However the code centers are placed exactly on the ideal positions. Hence this definition is not used for the specification of the DNL in Infineon µ-Controller ADCs.

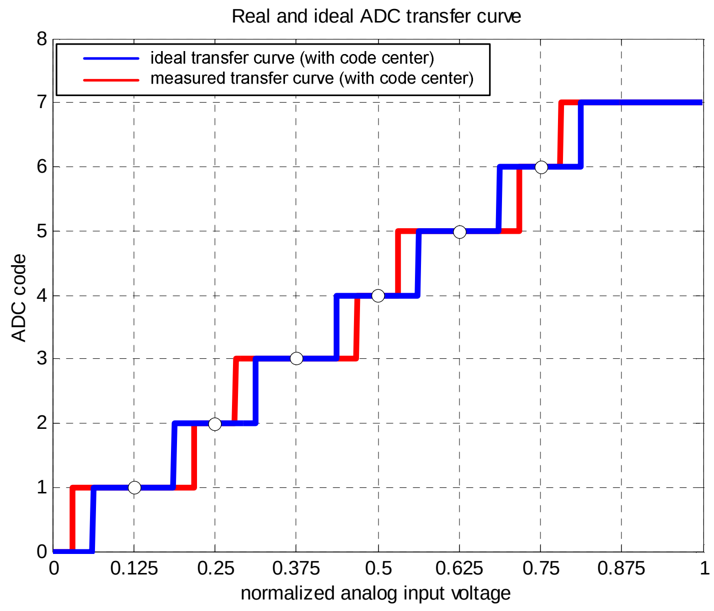

Figure 6 Differential Non Linearity, code mid point testing

Integral Non Linearity

The INL is defined as the deviation of the measured transfer curve from an adjusted transfer curve. The adjusted transfer curve is an ideal transfer curve but with the same offset and gain error as the measured one has. Hence the code width of the adjusted transfer curve is the same for all codes from 1 to 2^n -2 but is no longer 1LSB. An example of this is shown in Figure 7. In this case the adjusted code width is smaller than the ideal code width due to the gain error in the measured transfer curve.

The other way round is to correct the *tue* for offset and gain:

$$inl[i] = tue[i] - offset - gain[i]$$

$$i = 0 \rightarrow 2^n - 1$$

The minimum and maximum of the *inl* array are valid for the specification:

$$INL\max = \max(inl)$$

$$INL\min = \min(inl)$$

Note: Some INL specifications take the ideal transfer curve instead of the adjusted one. This would be the same as calculating the TUE.

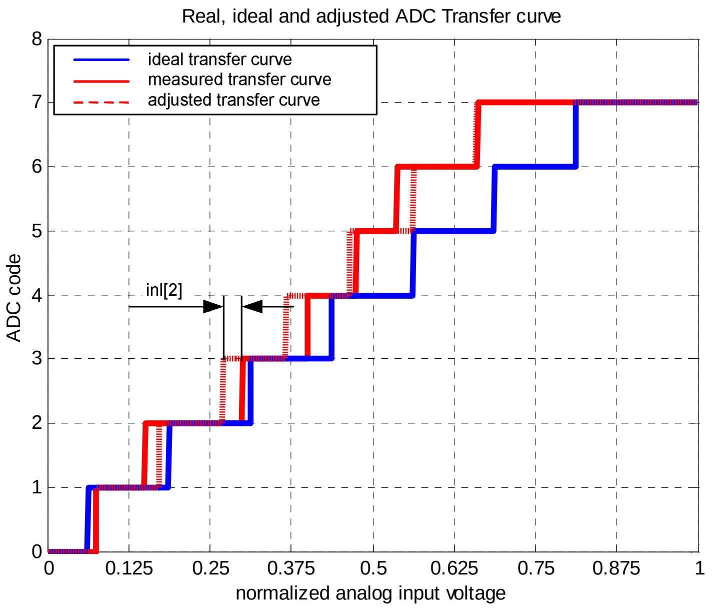

Figure 7 Integral Non Linearity

4 Static Ramp Test

This chapter describes the Static Ramp Test which is also known as the Linear Ramp Histogram Method as described in [1]. It is used for evaluating the error parameters described in Chapter 3.

4.1 Input Signal

The input signal for the static ramp test is, as the name already implies, a voltage ramp ranging either from the negative reference voltage to the positive reference voltage of the ADC or exceeding these voltages.

In the first case the ADC under test produces output codes ranging from 0 to full scale, assuming the ADC has no offset (Figure 8a).

In the case of a voltage ramp which exceeds Vref- and Vref+, the ADC under test produces more of both the 0 code and the full scale code (Figure 8b).

In order to get all codes, especially the lowest and highest codes, it is recommended to use the second approach for the input signal.

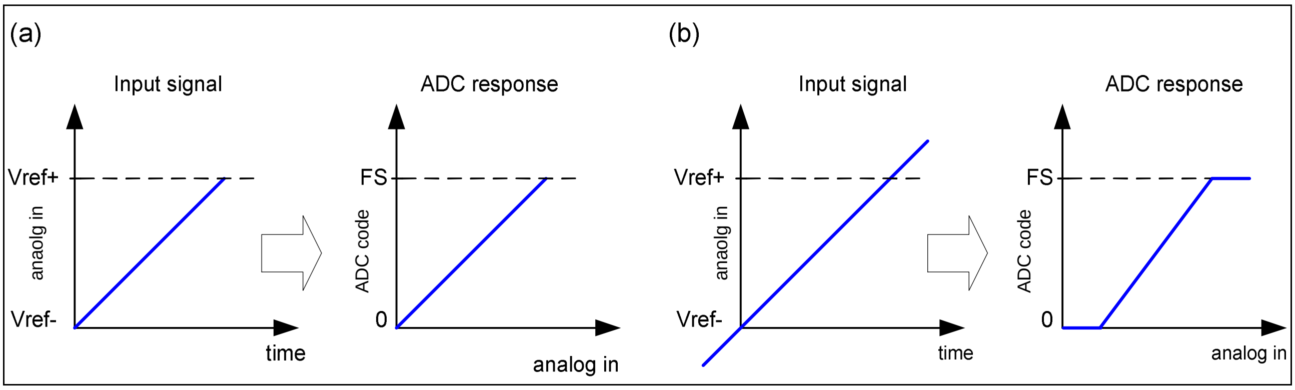

Figure 8 Input Signal and ADC Response for a Static Ramp Test

The input voltage ramp can be generated either purely analog or by using a DAC with a finite number of discrete output voltages as determined by the DAC resolution. (In reality, in most signal generators the output of the DAC will be amplified before it is output).

If an analog voltage ramp (e.g. integrator) is used, this signal must be synchronized with the clock signal which starts the ADC conversions. This signal is often referred to as the strobe signal. Furthermore, the conversions must be made in exactly equidistant time steps. If these two conditions are not met, the measurement can give poor results which are not related to the ADC under test.

Using a DAC as the signal source makes the setup easier as the only requirement here is that the ADC samples the DAC voltage after it has settled. After the conversion is finished, the DAC can be updated to the next output voltage. With an appropriate handshaking technique, any sampling jitter of the ADC has no influence on the measurement result.

In order to get a sufficient amount of samples for every ADC code, the resolution of the DAC must be greater than the resolution of the ADC under test. This is directly related to the measurement accuracy and repeatability as explained in Chapter 5.

4.2 Measurement Flow

A general measurement flow with a DAC as the signal source is shown in Figure 9. The DAC is initialized with the start voltage of the ramp before the ADC under test makes the first conversion. After the voltage on the DAC output has stabilized, the ADC converts it to a digital code which is then stored in a memory. The DAC is updated to the next voltage and the process repeats until the programmed ramp end voltage of the DAC is reached.

At the end the memory contains the ADC results corresponding to the ADC input signal but including the errors of the ADC (offset and linearity errors). This data cannot be used directly for the error calculations as the resulting transfer curve is not monotonic. In other words the ADC may toggle up and down as it approaches a code transition. This is due to the nature of all ADCs as the transfer function is a “many-to-one” mapping function. For example, all input voltages between Vref- and ½ LSB of the ADC ideally should be converted to code 0 in case of the ½ LSB compensated transfer curve.

A DAC, by contrast, converts one code into one output voltage, hence has a “one-to-one” mapping function. For more details refer to [1].

Note: For the error calculations described in Chapter 3 it is necessary to have a monotonic transfer curve to compare with an ideal ADC transfer curve.

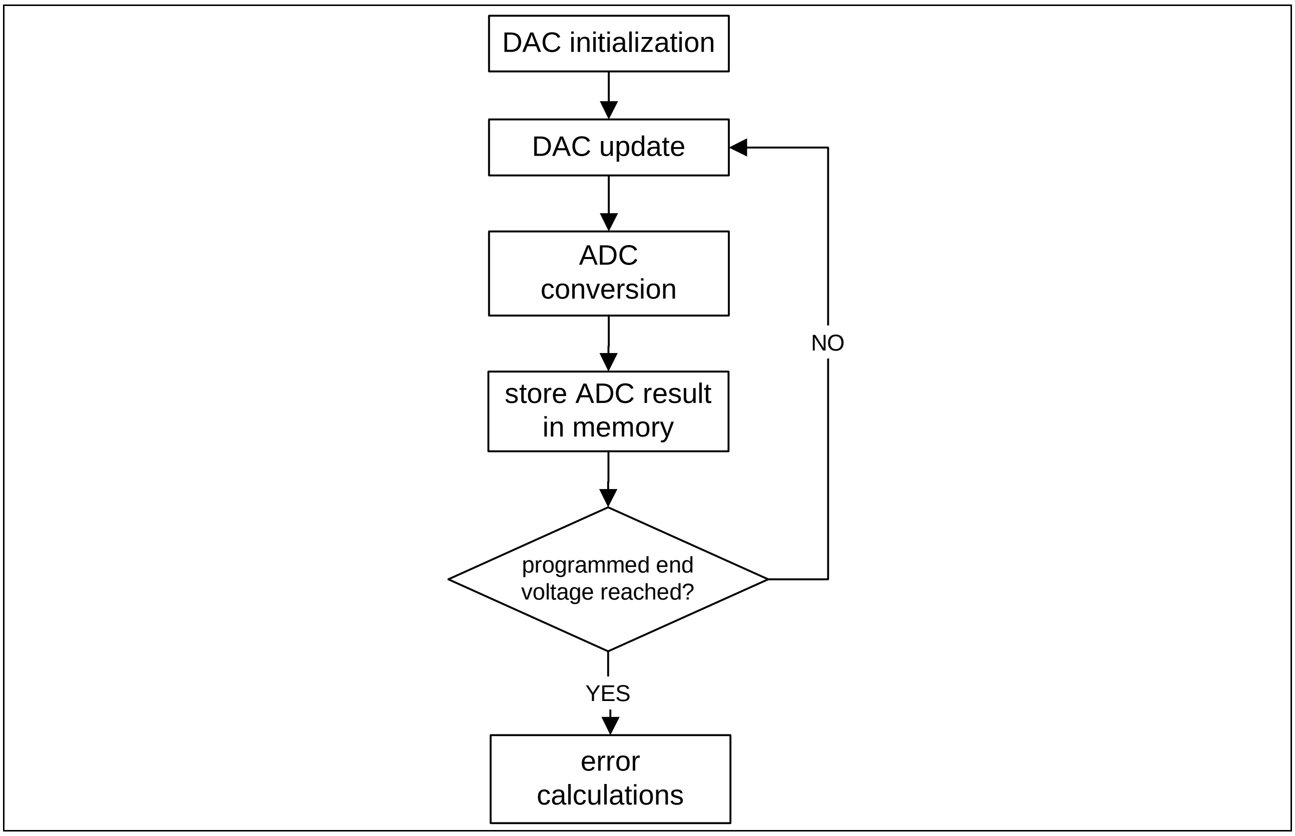

Figure 9 Static Ramp Test Measurement Flow

Figure 10 shows an example of raw measurement data for the first 4 ADC codes. Here an ADC with a resolution of n bits has been tested using a DAC with a resolution of n+3 bits. The green line is the transfer curve of an ideal ADC, the red curve represents the measured data.

With a 3 bit resolution difference between the DAC and the ADC, each ADC code width is tested for 8 different input voltages. (The LSB of the DAC is 2^3 times smaller than the LSB of the ADC). Note that for a characterization test of an ADC this would not be accurate enough, it is just for illustration of the measurement flow that this number was chosen.

To get a monotonic transfer curve out of which the error parameters can be calculated, the codes are sorted in ascending order. This process is shown for our example in Figure 11. Now this transfer function can be directly subtracted from the ideal transfer function and the *tue* plot can be built. Out of this array all errors can be calculated.

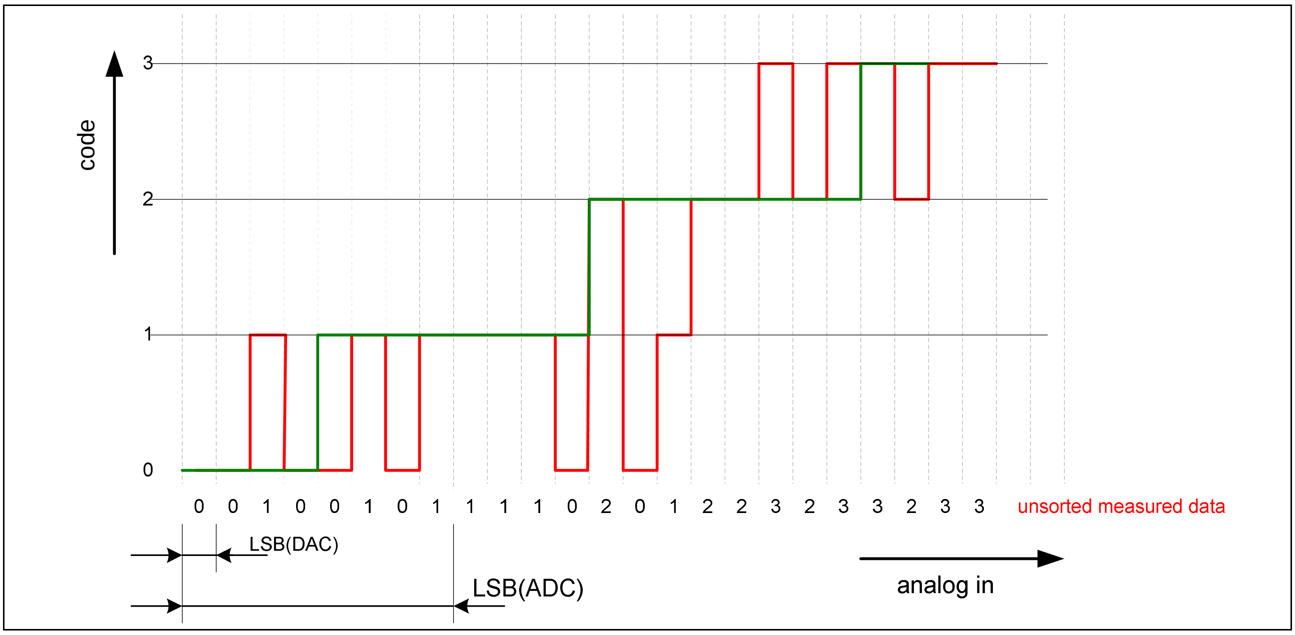

Figure 10 Example of a raw measurement data

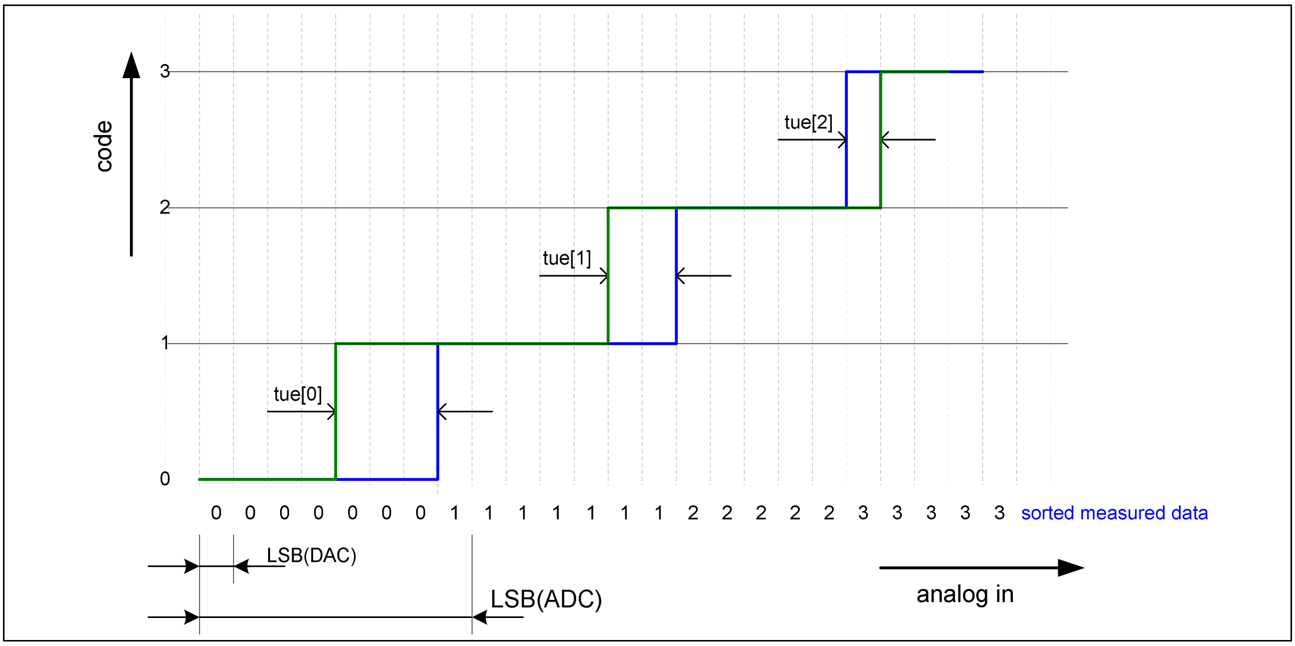

Figure 11 Example of a sorted measurement data

Once the resolution of the ADC increases, it becomes difficult to maintain the difference (ADC resolution – DAC resolution) > 4 for example. To overcome this problem we can sample every DAC output voltage more then once by the ADC. Clearly the raw data can no longer be plotted as in Figure 10 as it may happen that the ADC produces different codes for one and the same input voltage. The solution is to build a histogram which is simply a mapping of *how often every code occurs*. A histogram, denoted by H, is a 2 dimensional array with (2^n * DACresolution * Oversampling Factor) values. Every value represents the occurrence of that code. An example of a histogram for a 3 bit ADC is shown in Figure 12a. The total number of samples here is 64. This could be the result from using a 6 bit DAC as the input signal source or a 5 bit DAC as the input signal source where every voltage was sampled twice by the ADC.

Note: The process of code sorting as described previously is exactly the same as building a histogram.

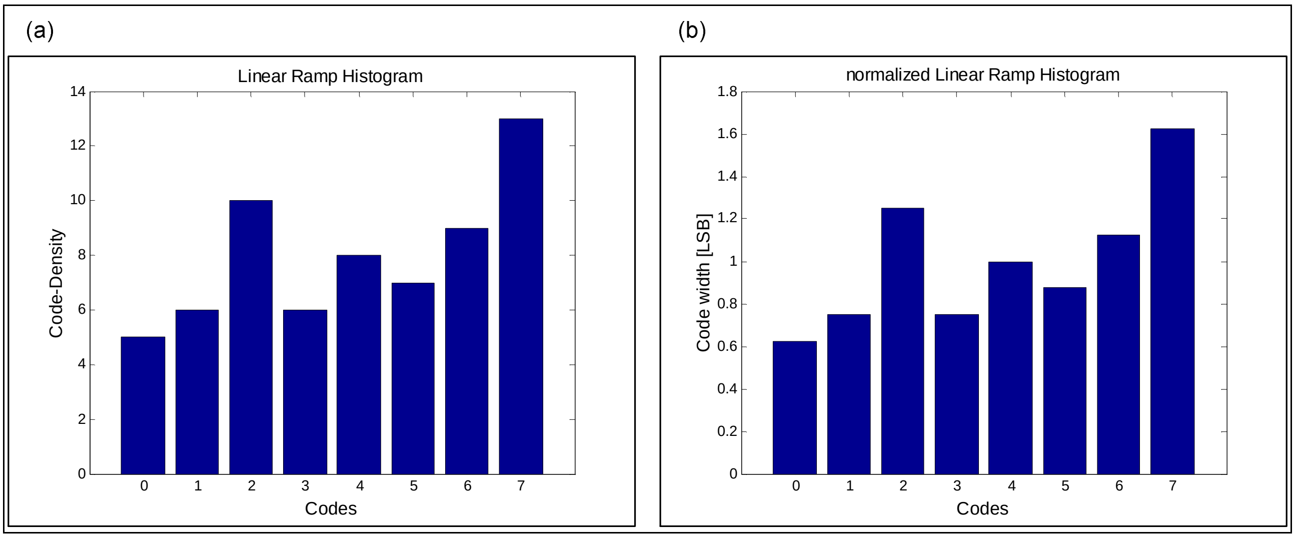

Figure 12 Example of a histogram

To convert the histogram back to a monotonic transfer curve it must be normalized to the ADC LSBs. This is done by dividing each histogram value by the average number of hits per code:

$$average\_amount\_of\_hits = sum(H)/2^n$$

$$H\_normalized = \frac{H}{average\_amount\_of\_hits}$$

Figure 12b shows the normalized histogram where every histogram value now represents the code width of the ADC.

As discussed previously, this transfer function can be subtracted directly from the corresponding ideal transfer function to get the *tue* array.

Note: In the case of an analog input ramp which exceeds the ADC reference voltages, the number of codes 0 and FS must be reduced by the same factor as the ramp is exceeded.

5 Summary and Limitations

5.1 Accuracy and Resolution

The method described above has some limitations and issues which must be considered when setting up this test. Offset and gain error are absolute measurements that require a highly accurate analog input signal. As an example, for an ADC with a reference voltage span of 3.3V and 12 bit resolution, the LSB voltage is 3.3V/2^{12} ~ 800µV.

If the generator used has the same absolute accuracy, the errors of the offset error and gain error measurement are at least ±1LSB. For the linearity errors DNL and INL, only the linearity of the supplied voltage ramp is of importance.

Furthermore, the resolution of every code width from the normalized histogram is directly related to the average amount of hits per code. If the average number of hits per code is 8, the resolution of every code width would be 1/8 of a LSB of the ADC. This is why the resolution of the DAC should be higher than the resolution of the ADC under test as mentioned before. Once this becomes difficult an oversampling factor should be used.

5.2 Limitations

Another error which has not been considered until now is monotonicity. According to [2] an ADC is nonmonotonic if, for a uniformly increasing or decreasing input signal, the output codes can increase and decrease. However a monotonicity error must not be confused with the random nature the ADC has at each decision level from when going from one code to another. Therefore [2] defines an ADC to be nonmonotonic when for any pair of input levels x_1 and x_2, with $x_1<x_2$ the following condition is met:

$$\bar{y}(x_1) - \frac{3*\sigma_y(x_1)}{\sqrt{M}} > \bar{y}(x2) - \frac{3*\sigma_y(x_2)}{\sqrt{M}} \qquad (non\ inverting\ ADC)$$

$$\bar{y}(x_2) - \frac{3*\sigma_y(x_2)}{\sqrt{M}} > \bar{y}(x_1) - \frac{3*\sigma_y(x1)}{\sqrt{M}} \qquad (inverting\ ADC)$$

Where $\bar{y}(x)$ is the mean output code value, σ is the standard deviation of a static input voltage x and M is the number of samples taken at each x value.

The linear ramp histogram method produces, in every case, a monotonic transfer function due to the sorting of the ADC results. Therefore it is not suitable for revealing monotonicity errors.

Hence a sanity check for monotonicity errors is done before an actual ADC validation using the linear ramp histogram method.

Note: Sometimes an ADC appears to be nonmonotonic if its output toggles rapidly. This may be caused by noise from the surrounding environment of the ADC and is not a monotonicity error.

6 Measurement conditions

This chapter describes the measurement conditions at which static ramp tests are performed. The specified error parameters are the result of all the ramp tests, in other words the ADC holds the specification under all the described measurement conditions. If applicable the conditions applied for the tests go beyond the specified limits, e.g. maximum ADC speed, in order to see the available margin to the specification.

Dependent on the features of the ADC there can be more specific measurement conditions which are not outlined here.

6.1 Nominal Conditions

In this test case normally a channel scan of all ADC channels of all ADCs in a µController is performed. The temperature is kept at room temperature, all supplies are at there nominal value and the ADC runs at the maximum specified speed. This test can reveal a potential channel dependency of an error parameter. The channel with the worst behavior is then preferably used for all further investigations.

6.2 Supply and temperature range

This test case covers the variation of all relevant supplies of the ADC at three different temperatures:

- Cold temperature on automotive standards: -40°C
- Room temperature
- High temperature on automotive standards: +125°C

Although the high temperature on automotive standards is defined as +125°C (which results in a junction temperature of approximately 150°C) the tests are sometimes also carried out at higher temperatures(e.g. 150°C ambient temperature) if it is required.

6.3 Reference variation

This test case covers the specified FSR of the ADC at full ADC speed and nominal supply and temperature. This is done by variation of the positive and negative reference inputs (if available) of the ADC.

Reducing the FSR means also a reduction of the absolute value of the LSB voltage:

Example: FSR = 5V, resolution=12bit, → 1LSB = FSR/4096 = 1.22mV

FSR = 3.3V, resolution = 12bit → 1LSB = FSR/4096 = 0.805mV

As the absolute accuracy (in mV) of the ADC keeps constant, the relative accuracy specified in LSB has to be adapted to the FSR decrease. Therefore the specification is relaxed for a smaller full scale range by specifying a k-factor which defines the increase of the error in LSB by decreasing the FSR.

Example: FSR = 5V, Offset error = ±4 LSB

FSR = 3V, k=1.51 → Offset error = 6.04LSB

6.4 Frequency variation

An integrated ADC in a µController may be influenced by noise and activity of the surrounding environment. This test checks for possible frequency dependent errors caused by different CPU-system frequencies. Therefore the ADC is kept at a constant frequency while the CPU and the digital interfaces operate at different selected frequencies.

Note: If port activities are involved, the ADC performance may be further reduced. But here also board effects can play a major rule, e.g.: board layout, supply blocking. For this reason this test case does not cover influences caused by port activities.

6.5 Reference exceeding

If the positive reference voltage is increased above the ADC supply a parasitic diode starts to conduct. This is also the case if the negative reference is lower then VSS. This affects the ADC performance, especially at hot temperature. This test case checks if the error parameters are within specification for the allowed limit of V_{REF+} - VDD at room temperature and hot temperature.

6.6 Speed test

This test checks if the ADC can operate within the specified minimum and maximum ADC clock speed without violation of the maximum allowed error parameters. The decisive factor is the clock at which the ADC converts, e.g. denoted with internal ADC clock f_{ADCI} or analog clock f_{ANA}. The low speed limit is a result of loss of charge during the conversion phase. The maximum speed is limited either by the sampling phase or the conversion phase or a combination of both.

This test is carried out at room temperature and nominal supplies.

6.7 Selectable reference

Some ADCs have the possibility to select the reference from an analog input channel, e.g. from channel 0. This introduces a further resistance in the reference path and limits again the maximum ADC speed.

The worst case condition for the maximum ADC speed is normally under the following conditions:

- Positive reference input selected from an analog input channel
- Supply is at the lower limit
- Hot temperature

This test checks if even under this conditions the error parameters are within specification for the maximum ADC speed.

6.8 Resolution

Beside the maximum possible ADC resolution there is often the possibility to convert at lower resolutions, e.g. 8 or 10-bit resolution instead of 12-bit. This can be used to reduce the conversion time and to increase the possible sample rate of the ADC.

This test is a pure functionality check of the possible resolution settings because the ADC simply stops the conversion earlier at lower resolution. Hence the specified error parameters can be scaled according to the following formula:

$$Error_{m-bit\,resolution} = Error_{n-bit\,resolution} * \frac{1}{2^{n-m}}$$

For example an offset error of ± 4LSB12 can be multiplied with 0,25 for conversions with 10-bit resolution.

7 References

[1] Mark Burns and Gordon W. Roberts *An Introduction to Mixed-Signal IC Test and Measurement*, Oxford University Press 2001

[2] IEEE. 2000. 1241:2000. *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*. New York: IEEE.

http://www.infineon.com

Published by Infineon Technologies AG