Application Note

AN2438/D
2/2003

ADC Definitions and Specifications

MOTOROLA intelligence everywhere™ | digital dna™

By: J. Feddeler and Bill Lucas
8/16 Bit Division Systems Engineering
Austin, Texas

Introduction

This application note will help users of analog-to-digital converters (ADC) understand common terminologies used in the electronics industry to define ADC operation and performance. There are many terms and parameters used to define the performance of ADC's. Included in this document are common definitions, numerical specifications, differences, and issues with the definitions. By understanding the terminology used to specify various ADC parameters, a systems designer can better understand how to obtain the greatest overall system performance, based on the various performance features of any given ADC system.

Terms and Definitions

The following terms are used in the electronics industry to define ADC operation.

Measurement Units

There are several terms commonly used to measure ADC performance. Improper or inconsistent use of terms may result in confusion and or misinterpretation of performance. Common measurement units in use in the industry are described here (The following examples assume a 10-bit, 5.12-V ADC with an ideal 2.56-V conversion at $200):

- **Volts (V)** — The error voltage is the difference between the input voltage that converts to a given code and the ideal input voltage for the same code. When the error is measured in volts, it is related to the actual voltages and is not normalized to or dependent on the input range or voltage supply. This measure is useful for fixed error sources such as offset but does not relate well to the observed error.
- **Least Significant Bits (LSB)** — A least significant bit (LSB) is a unit of voltage equal to the smallest resolution of the ADC. This unit of measure

approximately relates the error voltage to the observed error in conversion (code error), and is useful for systemic errors such as differential non-linearity. A 2.56-V input on an ADC with ± 3 LSB of error could read between $1FD and $203. This unit is by far the most common terminology and will be the preferred unit used for error representation.

- **Percent Full-Scale Value (%FSV)** — Percent full-scale voltage is a unit of voltage equal to $1/100^{th}$ of the input range of the ADC. This unit of error clarifies the size of the error relative to the input range, and is useful for trimmable errors such as offset or gain errors. This unit is difficult to accurately translate to observed error.
- **Counts** — A count is a unit of voltage equal to an LSB. It is a terminology unique to specifications of some Motorola ADC's and may be confusing to customers when doing cross-vendor comparisons.
- **Bits** — A bit is a unit equal to the log (base2) of the error voltage normalized to the resolution of the ADC. An error of N bits corresponds to 2^N LSB of error. This measure is easily confused with LSB and is hard to extrapolate between integer values.
- **Decibels (db)** — A decibel is a unit equal to 20 times the log (base10) of the error voltage normalized to the full-scale value (20*log(err_volts/input_range)). A 2.56 input on an ADC with an error of 50 db will convert between $1FD and $203. This figure is often used in the communications field and is infrequently used in control or monitoring applications.

ADC Transfer Curves

The ADC converts an input voltage to a corresponding digital code. The curve describing this behavior is the *Actual Transfer Function*. The *Ideal Transfer Function* represents this behavior assuming the ADC is perfectly linear, or that a given change in input voltage will create the same change in conversion code regardless of the input's initial level. The *Adjusted Transfer Function* assumes this behavior after the errors at the endpoint are accounted for.

Ideal Transfer Function

The *Ideal Straight-Line Transfer Function* of an ADC is a straight line from the minimum input voltage (voltage reference low; V_{REFL}) to the maximum input voltage (voltage reference high; V_{REFH}). The *Ideal Transfer Function* is then quantized (divided into steps) by the number of codes the ADC is capable of resolving. The input voltage range is divided into steps, each step having the same width.

This *Ideal Code Width (ICW)* — also known as 1LSB — is:

$$ICW = 1LSB = (V_{REFH} - V_{REFL})/2^N$$

Where:

N is the "width" of the ADC, in our examples, 10 bits.

The Ideal Transfer Function is then:

$$Code = (V_{IN} - V_{REFL}) / 1LSBV_{IN} = (Code*1LSB) + V_{REFL}$$

Quantization Error (E_Q) and Method

The way the Ideal transfer function is divided into steps depends on the method of quantization the ADC uses. The two possible methods are:

1. **Uncompensated Quantization —** The first step is taken at 1LSB, with each successive step taken at 1LSB intervals and the last step taken at V_{REFH} – 1LSB. The *Quantization Error* (E_Q) in this case is from 0LSB to 1LSB.
2. **½LSB Compensated Quantization** – The first step is taken at ½LSB, with each successive step taken at 1LSB intervals and the last step taken at V_{REFH} – 1½LSB. The *Quantization Error* (E_Q) in this case is ±½LSB.

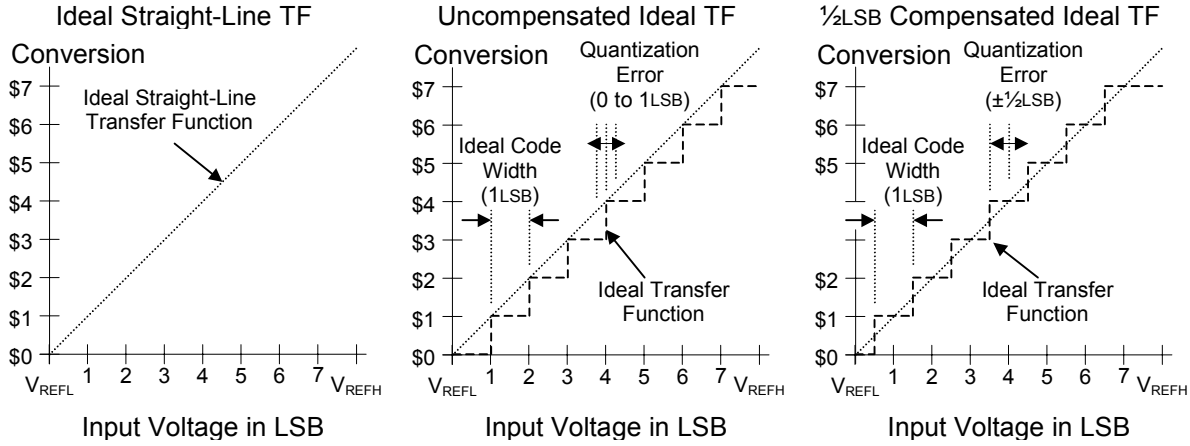
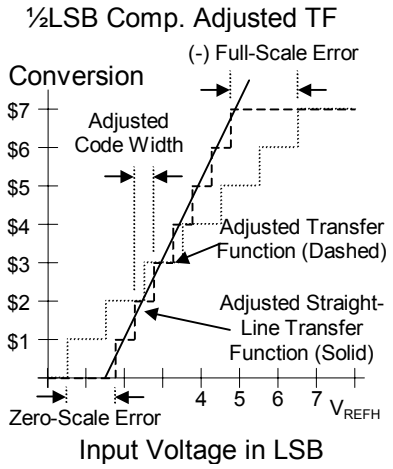


Figure 1. Quantization Graphs

Adjusted Transfer Function



Most ADC's exhibit some non-linearity at the endpoints due to the difficulty in measuring a signal that is identical to the reference. If these errors are accounted for, a more accurate portrayal of the ADC behavior in application is possible. For this reason, the *Adjusted Straight-Line Transfer Function* is drawn between the minimum input voltage plus the *Zero-Scale Error* ($V_{REFL} + E_{ZS}$) to the maximum input voltage plus the *Full-Scale Error* ($V_{REFH} + E_{FS}$).

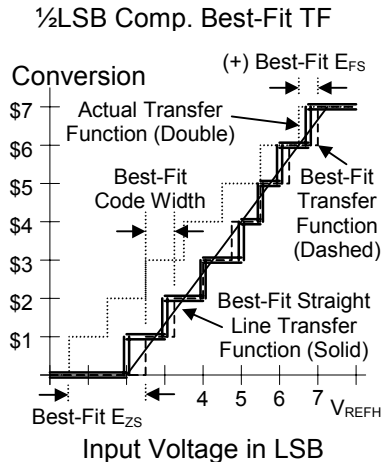
The Adjusted Transfer Function is then quantized in the same method as the Ideal Transfer Function. The *Adjusted Code Width* is therefore:

$$ACW = [(V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS})] / 2^N$$

The Adjusted Transfer Function is then:

$$Code = (V_{IN} - V_{REFL} - E_{ZS}) / ACW \qquad V_{IN} = (Code*ACW) + V_{REFL} + E_{ZS}$$

Best-Fit Transfer Function



Some ADC's exhibit low Zero- and Full-Scale Errors but still have significant non-linearities. In cases where these linearities tend to be in one direction (for example, a significantly "bowed" function) the best application results may be obtained if the errors are compared to a *Best-Fit Transfer Function*. *The Best-Fit Straight-Line Transfer Function* is the line from which the average deviation of all conversions is minimum. Computing this function requires that the entire *Actual Transfer Function* be recorded, which is impractical in most applications. Therefore, any performance parameters calculated against the *Best-Fit Transfer Function* are not useful to the user. Unfortunately, many automatic evaluation packages (software and hardware) assume this type of curve.

Zero-Scale Error and Full-Scale Error

The non-linearities at the endpoints are considered special cases due to the ease with which they are measured and corrected. The non-linearity at the beginning of the *Actual Transfer Function* is called the *Zero-Scale Error* (E_{ZS}) and the non-linearity at the top end of the function is called *Full-Scale Error* (E_{FS}). The Zero- and Full-Scale Errors have the following definitions:

- **Zero-Scale Error (E_{ZS})** is the difference between actual first transition voltage and the ideal first transition voltage (if the first transition is not from \$000 to \$001, then use the difference between the actual and ideal \$001–\$002 transition voltages, and so on).

NOTE: *The Ideal Code Width for the zero code is ½LSB for ADC's with ½LSB compensated quantization.*

Representing this error is by code widths:

E_{ZS} = CCW(0) – ICW(0)

Or, in the case where the first "x" codes are missing,

E_{ZS} = CCW(x) – sum(i=0→x)[ICW(i)]

- **Full-Scale Error (E_{FS})** is the difference between the actual last transition voltage and the ideal last transition voltage (if the last transition is not from \$3FE to \$3FF, then use the difference between the actual and ideal \$3FD–\$3FE transition voltages, and so on).

NOTE: *The Ideal Code Width for the last code is 1½LSB for ADC's with ½LSB compensated quantization.*

Representing this error by code widths:

E_{FS} = CCW(last) – ICW(last)

Or, in the case where the last "x" codes are missing,

E_{FS} = CCW(last-x) – sum(i=x→last)[ICW(i)]

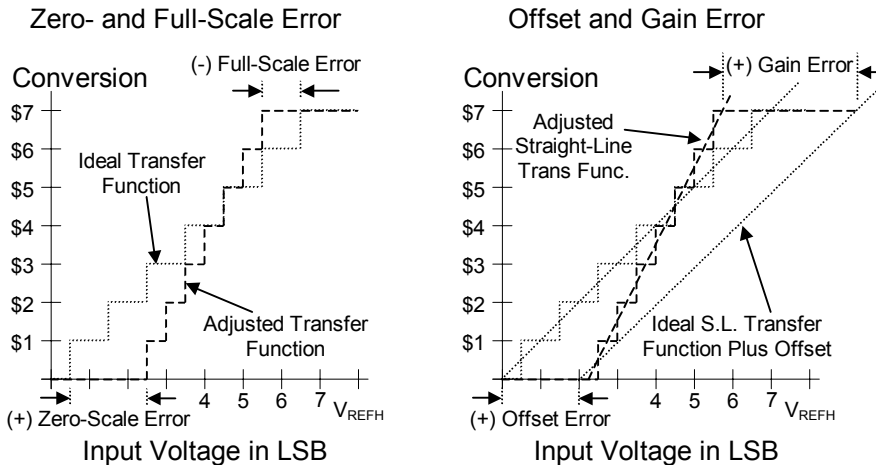


Figure 2. Endpoints Error Graph

Offset and Gain Error

Zero-Scale Errors and Full-Scale Errors can be used to calculate *Offset* and *Gain Errors*. These terms are used to define the performance on many industry-standard ADC's but the definitions used vary and can be misleading or inconsistent.

- **Offset Error (E_O), Adjusted Offset, or Zero-Scale Offset** is the difference between the actual and ideal first transition voltages. This is the same definition as *Zero-Scale Error*. The term offset; however, implies that all conversions are off by an equal amount. In the case of a strong non-linearity near the *Zero-Scale Value*, this definition may be misleading, and the less ambiguous Zero-Scale Error term is preferable.
- **Best-Fit Offset** is the difference between the *Best-Fit Straight-Line Transfer Function* and the *Ideal Straight-Line Transfer Function* at V_{REFL}. Some definitions define the offset point at the center-conversion ($(V_{REFH} - V_{REFL})$ /2) instead of at V_{REFL}. This offset is virtually impossible to measure in the application and is therefore only a laboratory curiosity. Since this yields optimistic results and is not measurable in the application, it can be misleading and will not be used.
- **Full-Scale Offset** is the difference between the actual and ideal last transition voltages. This is the same definition as *Full-Scale Error*, and is misleading for the same reason that Offset is misleading with respect to Zero-Scale Error.
- **Gain Error (E_G) or Adjusted Gain Error** is the difference in the slope of the *Actual* and the *Ideal Straight-Line Transfer Functions*. The error is not measured as a slope but rather as the difference in the total available input range from the first to the last conversions between the Ideal and Adjusted Straight-Line Transfer Functions. It is can also be expressed by:

  $$E_G = E_{ZS} - E_{FS}$$

  Gain Error is not directly measurable and the term has been inconsistently defined in the literature. Additionally, if there are strong non-linearities at the endpoints, this definition of Gain Error may be misleading, so the less ambiguous Full-Scale Error term is preferred provided a simple gain calculation (above) is possible.
- **Best-Fit Gain Error** is the difference in the slope of the *Best-Fit* and *Ideal Straight-Line Transfer Functions*. The error is not measured as a slope but rather as the difference in the total available input range from the first to the last conversions between the Ideal and Best-Fit Straight-Line Transfer Functions. Since the Best-Fit Straight-Line Transfer Function will result in an optimistic Gain Error and is virtually impossible to measure in the application, it can be misleading and will not be used.

Consistency with previous Motorola documents can be achieved by replacing references to *Offset Error* with *Zero-Scale Error* and *Gain Error* with the difference between of *Zero Scale Errors* and *Full-Scale Errors*.

Differential Non-Linearity (DNL)

Differential Non-Linearity (DNL) is the maximum of the differences in the each conversion's *Current Code Width* (CCW) and the *Ideal Code Width* (ICW). DNL is the most critical of the measures of an ADC's performance for many control applications since it represents the ADC's ability to relate a small change in input voltage to the correct change in code conversion. DNL is defined as:

Code DNL = CCW – ICW

DNL = Max (Code DNL)

Some literature defines DNL using the *Adjusted Code Width* (ACW), which means *Zero-* and *Full-Scale Error* have been adjusted for. For relatively accurate ADC's, the difference with respect to DNL is negligible, but using the ACW complicates defining and testing DNL. Additionally, this definition is only valid if the application has trim capability.

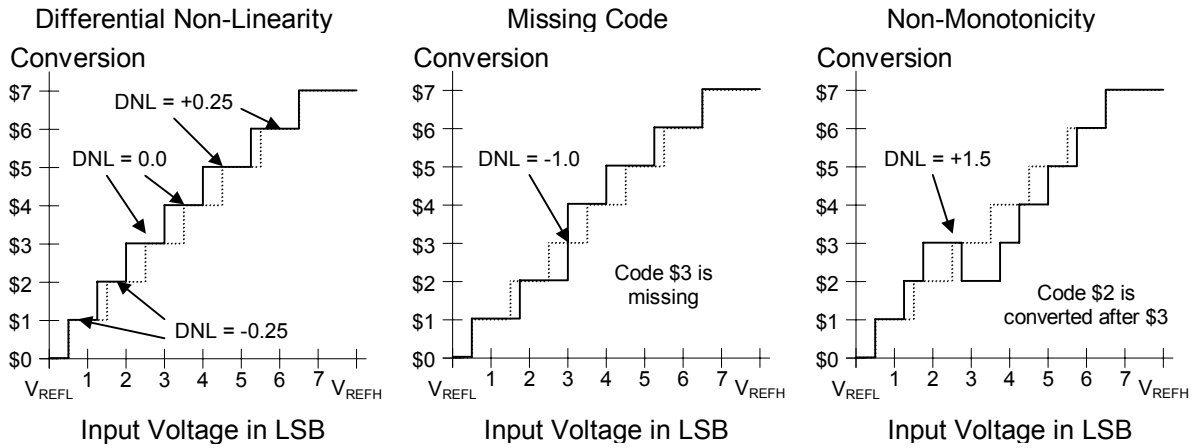


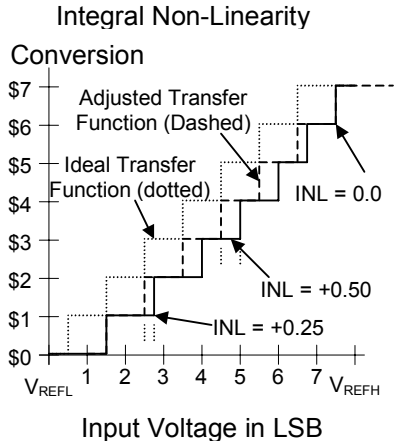
Figure 3. Differential Non-Linearity, Missing Codes, and Non-Monotonicity Graphs

Related to DNL are two critical figures of merit used in defining ADC operation. These are:

- **Missing Codes** — An ADC has missing codes if an infinitesimally small change in voltage causes a change in result of two codes, with the intermediate code never being set. A DNL of –1.0 LSB indicates the ADC has missing codes (DNL measured by this definition cannot be less than –1.0 LSB).
- **Monotonicity** — An ADC is *monotonic* if it continually increases conversion result with an increasing voltage (and vice versa). A non-monotonic ADC may give a lower conversion result for a higher input voltage, which may also mean that the same conversion may result from two separate voltage ranges. Often, the transfer function will completely miss the lower code until after the higher code is converted (on an increasing input voltage).

Some literature suggests that a DNL of greater than 1.0 LSB may indicate non-monotonicity. Non-monotonicity is usually accompanied by large, positive DNL (>1.0 LSB), although a non-monotonic situation can be coincident with a DNL of less than 1.0 LSB.

Integral Non-Linearity (INL)



Integral Non-Linearity (INL) is defined as the sum from the first to the current conversion (integral) of the non-linearity at each code (*Code DNL).* For example, if the sum of the DNL up to a particular point is 1LSB, it means the total of the code widths to that point is 1LSB greater than the sum of the ideal code widths. Therefore, the current point will convert one code lower than the ideal conversion.

In more fundamental terms, INL represents the curvature in the Actual Transfer Function relative to a baseline transfer function, or the difference between the current and the ideal transition voltages. There are three primary definitions of INL in common use. They all have the same fundamental definition except they are measured against different transfer functions. This fundamental definition is:

$$\text{Code INL} = V(\text{Current Transition}) - V(\text{Baseline Transition})$$

$$\text{INL} = \text{Max}(\text{Code INL})$$

The three primary definitions are:

1. **INL, Adjusted INL, or Endpoint INL** — The current transition voltage is compared to the corresponding transition voltage on the *Adjusted Transfer Function*. This is a useful indicator of the best the ADC can do if the endpoint non-linearities (*Zero*- and *Full-Scale Errors*) are measured and trimmed out.
2. **Unadjusted INL** — The current transition voltage is compared to the corresponding transition voltage on the *Ideal Transfer Function*. This is a measure of the total error except for *Quantization Error*.
3. **Best-Fit INL** — The current transition voltage is compared to the corresponding transition voltage on the *Best-Fit Transfer Function*. This will usually give a balanced positive and negative error across the entire curve so the results look very optimistic, but since it is difficult to obtain the Best-Fit Transfer Function in application, it is not a very useful measure. Unfortunately, this is what many evaluation packages (hardware and software) measure.

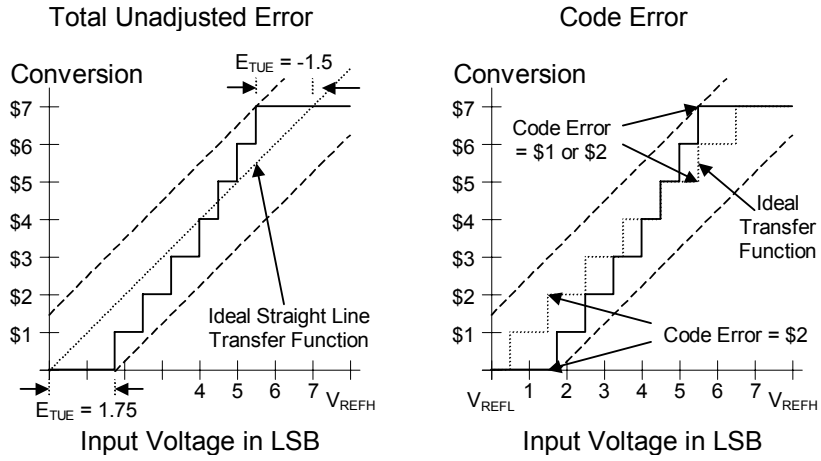


Figure 4. Total Unadjusted and Code Error Graphs

There are some related definitions to Total Unadjusted Error that vary slightly in definition. These are:

- **Total Error** is the same as Total Unadjusted Error, but the term is misused in several ADC references and is therefore misleading. The less ambiguous term Total Unadjusted Error is preferred.
- **Total Adjusted Error** is the difference between the *Actual* and *Adjusted Straight-Line Transfer Function*, accounting for INL plus *Quantization Error*. This term is redundant and potentially confusing with respect to Total Unadjusted Error and will not be used.
- **Code Error** is the error between the ideal code and the current code. This is the only figure of merit that measures by the quantized output instead of voltage. The code error is the Total Unadjusted in LSB, rounded to the nearest integer.

Power Supply Noise Error

Power Supply Noise Error (ENS) is the difference in conversion caused by power supply noise (on the V_{DDAD}; ADC power or V_{SSAD}; ADC ground pins). Power Supply Noise Errors are caused by:

- **Comparator Power Supply Rejection Ratio (PSRR)** — This ratio defines the ability of the comparator to reject power supply noise (relative to the inputs) when making a differential comparison between the inputs. PSRR is commonly measured in decibels (db). A comparator with a PSRR of 60 db will interpret 1-V power supply noise the same as 1 mV of differential input voltage.
- **Power Supply Decoupling Ratio (PSDR)** — This ratio defines how much of the power supply noise is coupled onto the inputs (either reference or input). A decoupling ratio of 24 db indicates that 16 mV of power supply noise will cause 1 mV of input or reference noise.
- **Differential Power Supply Decoupling Ratio (DPSDR)** — This ratio defines how much of the power supply noise is coupled onto the differential between the inputs. The most common way of reducing Power Supply Noise Error is by decoupling the reference and input equally (since it is impossible to reduce coupling altogether). A differential decoupling ratio of 30 db means that 32 mV of power supply noise will be interpreted as 1 mV of differential input noise.

Power Supply Noise Error is ultimately inherent in any ADC design. It is reduced in the design phase by the use of differential, cascaded circuits (increases PSRR), reducing parasitics (increase PSDR), and matching parasitics on the input and reference paths (increases DPSDR). However, these techniques are not perfect. The end user can apply the following techniques to reduce Power Supply Noise Error:

- **Analog Power Supply Bypassing** — The ADC power supply must be bypassed to its ground as close to the MCU pins as possible. The preferred method is with a high-frequency capacitor (0.01 μF or 0.1 μF) near the MCU and a larger capacitor near the power supply's source. Resistive impedance must be minimum, and chokes should never be used.
- **Digital Power Supply Bypassing** — The input channel for the ADC often is selected on a multi-purpose I/O pin. In this case, there is considerable coupling to the digital power supply. Additionally, most ADC's integrated into microcontrollers are built on the same silicon substrate as the digital circuits, which means ground noise will be coupled. For these reasons, the same bypassing techniques are recommended for the digital power supplies as the analog supplies.
- **Elimination of Noise Sources** — The best way to eliminate Power Supply Noise Error is to eliminate Power Supply Noise. Critical ADC conversions must be conducted when the rest of the system is as quiet

as possible. Most importantly, no Output Drivers on the microcontroller should be activated during the ADC conversion. The preferred mode for ADC conversions is in WAIT mode.

Input to Reference Differential Noise Error

Input to Reference Differential Noise Error, or Input Noise Error (ENIN), is the error due to noise, or short-term deviation from the DC-average, on the input (V_{IN}) or either reference (V_{REFH} or V_{REFL}) relative to the other. Input Noise Error is caused by:

- **Reference Coupling Ratio** — This defines the amount of noise that is injected onto one reference (i.e., V_{REFH}) that will be coupled onto the other reference (i.e., V_{REFL}). The higher the coupling ratio, the closer the references are to each other. If the coupling ratio is low, then between half and all of this noise will be differential to the input signal, depending on which reference the input is more closely coupled to and which reference the input voltage is closer to. The external (board-level) network of capacitors and parasitic impedances connected between the references and noise sources usually defines this ratio.
- **Differential Coupling Ratio** — This defines the amount of noise that is injected onto either the input or either reference that will be coupled onto the other. The higher the coupling ratio, the less differential noise will be seen between the input and the reference. This ratio primarily depends on the decoupling or filtering method that is used on the input.
- **Input Coupling Ratio** — This defines the amount of noise that is generated from a given noise source that is injected onto the input or reference. This depends on the layout and parasitics on the external board as well as the internal circuit.
- **EMC** — Conducted or Radiated noise can be picked up by the input or reference signals even in otherwise good printed circuit board layouts. This pick-up is usually dependent on the impedance of the input or reference source. It commonly affects the input more than the reference since the reference is usually much lower impedance.

Input to Reference Differential Noise is primarily externally controllable through good PC board layout and decoupling. Internally, shielding and isolation from switching signals and clock feed-through reduction techniques are usually employed to reduce internal sources. External methods of reducing Input Noise include:

- **Reference Bypassing** — The high and low references (V_{REFH} and V_{REFL}) should be bypassed to each other similar to the method used on V_{DDAD} and V_{SSAD} (high-frequency capacitor near pins, low frequency capacitor near source, low resistive impedance and no choke). This increases the Reference Coupling Ratio, reducing differential noise.

- **Input Bypassing** — Placement of a capacitor from the input to the reference increases the Differential Coupling Ratio, reducing the amount of differential noise. The size of the capacitor should be as large as possible, allowing for the required dV/dt of the input given the analog source impedance and the function of the signal.
- **Star Routing** — The references and analog power supplies should be star-routed so that all loads see an equal impedance to the source, and there is little coupling from one load to another that is not common to all loads.
- **Input Shielding** — Placing shields (preferably a double shield using first one of the references and then V_{SSAD}) between the inputs and all other signals (including supplies) can increase differential coupling and reduce input coupling. This also reduces EMC issues.

Noise Error Mechanism

Regardless of the type of noise (Power Supply Noise or Input to Reference Differential Noise), the error mechanism is the same. While the DC-average of most noise is zero (primarily due to the power supply which sources the reference or input), the short-term average is non-zero as shown in **Figure 5**. The amount of error depends on the magnitude and decay rate of the noise relative to the width of the sample or compare window.

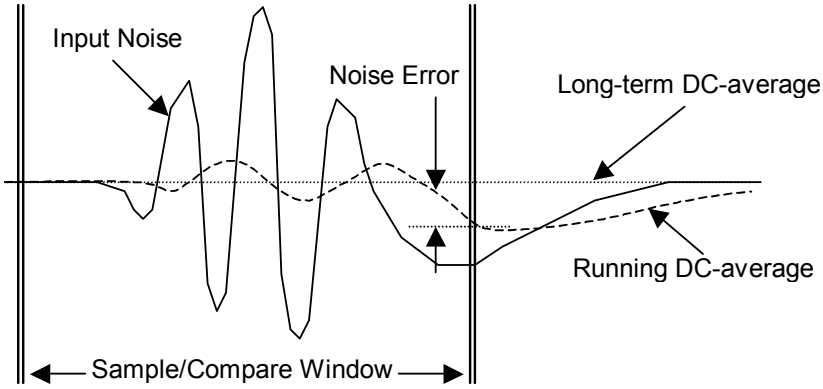


Figure 5. Noise Error Graph

Figure 5 is a simplification of the true mechanism meant for illustrative purposes. In reality, phase error due to parasitic impedances in the noise-coupling path, analog source impedance, and the gain and slew rate of the comparator will tend to shift the sensitivity either towards the beginning of the sample/compare window (in the cases of phase shift and comparator gain) or end of the sample window (in the case of analog source impedance). The general mechanism is the same in all cases.

Synchronous Noise Offset Error

The worst particular type of noise with regards to ADC performance is synchronous noise, or noise that occurs with the same signature at the same relative time in the ADC conversion process on repetitive conversions. One common example of such noise is the microcontroller's bus clock noise, if the ADC is synchronous to the bus. Other examples include I/O drivers controlling other board-level functions that in turn control the analog input.

In the case of synchronous noise, the same noise error, see **Figure 5**, occurs with each conversion. This can appear to be an offset error, instead of the variable errors typically associated with noise. In some cases, this could appear as a gain error, if the noise coupling is dependent on the voltage difference between the input and one of the supplies or references.

Changing the ADC conversion time relative to the synchronous noise can reduce Synchronous Noise Offset (after all other noise reduction techniques have been used). This is most practical if the noise source is low frequency. If the noise source is the microcontroller bus clock (or based thereupon) and the ADC is synchronous to the bus, the only option is to operate in Wait mode.

Reducing Random Noise Error

Random noise has the same signature as shown above, but has a random timing relationship with respect to the ADC conversion process. Noise of this type includes single events (very slow switching signals), EMC events, line noise, and white noise. A similar form of noise includes asynchronous noise, such as communication devices running asynchronous to the ADC clock such as SPI's. The magnitude of the noise error (as described in **Figure 5**) of this type must have a uniform distribution across ADC samples (either truly random or white) or it is at least partially synchronous.

Random noise can be divided into the following types relative to the ADC conversion cycle (the entire cycle must be considered, not just the individual sample/compare windows):

- **Single Event, Returns to Zero in Less than 1 Conversion Cycle** — Since the noise error on subsequent conversions is 0LSB, the effective noise error can be reduced to less than ¼LSB by sampling four times for every 1LSB of noise error. This is the type of noise expected from occasional I/O switching or other low-frequency switching events.
- **Single Event, Returns to Zero in X Conversion Cycles** — For an unknown return-to-zero waveform to be reduced to ¼LSB, the input must be sampled 4*X times for every 1LSB of noise error. If the waveform is predictable, the maximum noise error (therefore, the number of cycles required to average it out) can be reduced. This type

of noise includes line noise and DC-bias shifts due to regulator load changes, display drivers, etc.

- **High Frequency Events (Random Frequency)** — Random Frequency noise that is at more than ¼ the ADC conversion cycle frequency can be averaged out in the same manner as low frequency nose (four samples for every 1LSB of noise for ¼LSB error). White noise and other background noise are of this type.
- **High Frequency Event (Single Frequency)** — Any single-frequency noise at higher than ¼ the ADC conversion cycle frequency will beat with the ADC frequency, causing aliasing to occur at the beat frequency. Removing this aliasing error requires that four samples (for every 1LSB of noise) be taken at uniform intervals of at least twice the beat frequency. Higher levels of over-sampling may be necessary if the beat frequency is not known precisely. This type of noise, which includes asynchronous clock switching or higher frequency communication devices such as CAN, does not fit the uniform-distribution requirement necessary for accurate noise reduction.

Input Leakage Offset Error (E_{INO})

There is an additional source of input error that is often overlooked but can be extremely significant. Leakage on the Analog input pin or on the PC board can cause a voltage drop across the resistive portion of the Analog Input Source.

Many circuits (especially battery voltage and temperature detection circuits) use high value resistive voltage dividers to create the analog reference. Usually, a capacitor is placed on the input so the Analog AC Source Impedance (Z_{AIN}) is reduced so the ADC will be able to acquire the signal properly. However, the Analog DC Source Resistance (R_{AIN}) is still potentially very high (perhaps 100 kΩ).

Leakage on the analog inputs (I_{IN}) is usually specified at a maximum of 1 μA. Typical numbers vary from 5 nA (older process at room temperature) to 500 nA (newer process at hot temperature). This leakage on all processes increases exponentially with temperature. Added to this is PC board resistance (R_B), which can be as low as 1 MΩ but is usually around 10 MΩ. Given this, the Input Leakage Offset Error is:

$$E_{INO} = (I_{IN} + V_{IN}/R_B)*R_{AIN} / 1LSB$$

This error can be on the order of 10's of LSB if the Analog DC Source Resistance is high (example, 1 μA*100k/2.5 mV = 40 LSB). There is no way of perfectly trimming this error especially since it is temperature dependent. The user can set up a parallel input with the same Analog Source Impedance and similar, but fixed, V_{IN} to get a rough idea of the error, but since the input leakage is variable from pad to pad this method will not be perfect.

The best way to eliminate this error is to reduce the Analog DC Source Resistance and any form of leakage within the customer's control (PC Board leakage). An active circuit (op-amp) that buffers the Input Voltage (V_{IN}) can reduce Analog DC Source Resistance, but the problem may just shift to the input of the active circuit.

Acquisition Error (E_{AQ})

Acquisition Error is the error due to the inability of the acquisition circuits to fully charge the input of the ADC in the time allotted. The Acquisition Error will depend on the type of acquisition circuit the ADC uses:

- **Fixed Acquisition Time Sample and Hold Acquisition** — This type of circuit opens the input of the ADC to the Analog Input for a fixed period of time, usually some number of conversion clock cycles. In these cases, the Analog AC Source Impedance is critical since it will have to be less than a specified amount in order for the input to charge in the given time. Many ADC's from Motorola use this architecture.
- **Fixed Acquisition Time Sample and Hold with Buffered Input Acquisition** — This type of circuit is similar to the Fixed Acquisition Time architecture only for the first few cycles of the acquisition period, an op-amp buffers the input so that the ADC charges much faster. During the last few cycles of the acquisition period, the buffer is disabled and the Analog Input Source charges the ADC input the rest of the way. This architecture is nominally faster than the other for a given source impedance, but if the offset in the op-amp is high then the total charge time (from the input-offset voltage to the correct input voltage) may still be dependant on the AC source impedance. Most remaining Motorola ADC's use this architecture.
- **Variable Acquisition Time Sample and Hold Acquisition** — This type of circuit uses a variable period — either software controlled or controlled through a programmable timer — for the acquisition phase. In this manner, high Analog Source Impedance can be compatible with the ADC, at the expense of acquisition time. Since the time is variable, then in cases where there is low Analog Source Impedance the acquisition time can be as fast as possible. Many competitors use this form of acquisition.
- **Continuous Acquisition** — This type of circuit continually applies the input voltage to the ADC's comparator input. Few 10b ADC architectures are compatible with this method (full-RDAC SAR architectures can use this type of acquisition but are usually limited to about 7b of accuracy).

The Acquisition Error is related to the difference between the required Acquisition time and the allowed acquisition time.

Acquisition Time (Sample and Hold) and Error Calculations

A number of steps are required during an analog-to-digital conversion. When an analog-to-digital conversion begins, one channel select switch is closed, allowing the sample and hold capacitor (C_{AIN}) to charge. The channel select switch is then opened. The charge applied to the sample and hold capacitor is then converted into a digital representation by the successive approximation register. A conversion complete flag will be set and an optional interrupt can be generated.

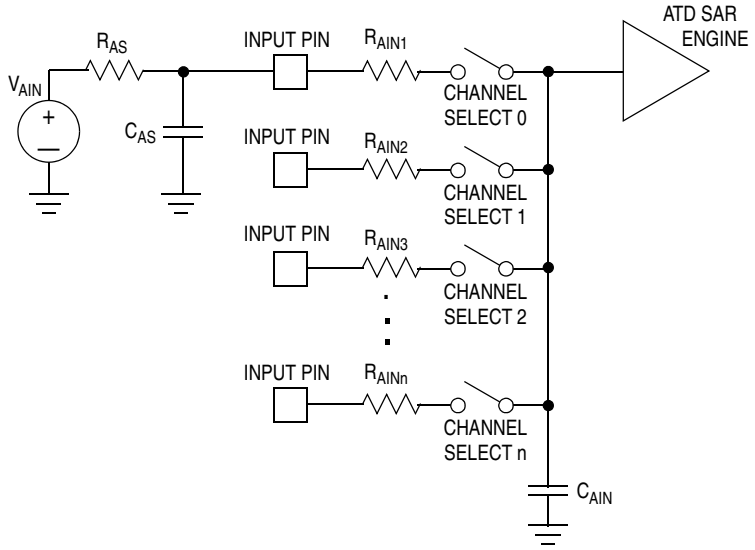


Figure 6. Resistor and Capacitor Placement

Proper sampling is dependent on the following factors:

- **Analog Source Impedance** — This is the resistive (or real, in the case of high frequencies) portion of the network driving the analog input voltage V_{AIN}.
- A**nalog Source Capacitance (C_{AS})** — This is the filtering capacitance on the analog input, which (if large enough) may help the analog source network charge the ATD input in the case of high R_{AS}.
- **ATD Input Resistance (R_{AIN})** — This is the internal resistance of the ATD circuit in the path between the external ATD input and the ATD sample and hold circuit. This resistance varies with temperature, voltage, and process variation but a worst case number is necessary to compute worst case sample error.
- **ATD Input Capacitance (C_{AIN})** — This is the internal capacitance of the ATD sample and hold circuit. This capacitance varies with temperature, voltage, and process variation but a worst case number is necessary to compute worst case sample error.

- **ATD Conversion Clock Frequency (f_{ATDCLK})** — This is the frequency of the clock input to the ATD. It clocks the counter for the successive approximation register and times the sample and hold and conversion sequencing. This clock is normally driven from the MCU's bus clock and may employ a clock prescaler. This frequency will determine the width of the sample window and the time for the conversion. In these examples, we will use 14 ATDCLK cycles for the total conversion.
- **Input Sample Frequency (f_{SAMP})** — This is the frequency that a given input channel is sampled.
- **Delta-Input Sample Voltage (ΔV_{SAMP})** — This is the difference between the current input voltage (intended for conversion) and the previously sampled voltage (which may be from a different channel).
- **Delta-Analog Input Voltage (ΔV_{AIN})** — This is the difference between the current input voltage and the input voltage during the last conversion on a given channel.

In cases where there is no external filtering capacitance (C_{AS}), the sampling error is determined by the number of time constants of charging and the change in input voltage relative to the resolution of the ATD:

$$\text{\# of time constants } (\tau) = (14 / f_{ATDCLK}) / ((R_{AS} + R_{AIN}) * C_{AIN})$$

$$\text{sampling error in LSB } (E_S) = 2^N * (\Delta V_{SAMP} / (V_{REFH} - V_{REFL})) * e^{-\tau}$$

The maximum sampling error (assuming maximum change on the input voltage) will be:

$$E_S = (3.6/3.6) * e^{-(14/((7\text{ k} + 10\text{ k}) * 50\text{ p} * 2\text{ M}))} * 1024 = 0.271 \text{ LSB}$$

In the case where an external filtering capacitance is applied, the sampling error can be reduced based on the size of the source capacitor (C_{AS}) relative to the analog input capacitance (C_{AIN}). Ignoring the analog source impedance (R_{AS}), C_{AS} will charge C_{AIN} to a value of:

$$E_S = 2^N * (\Delta V_{SAMP} / (V_{REFH} - V_{REFL})) * (C_{AIN} / (C_{AIN} + C_{AS}))$$

In the case of a 0.1 μF C_{AS}, a worst case sampling error of 0.5 LSB is achieved regardless of R_{AS}. However, in the case of repeated conversions at a rate of f_{SAMP}, R_{AS} must re-charge C_{AS}. This recharge is continuous and controlled only by R_{AS} (not R_{AIN}), and reduces the overall sampling error to:

$$E_S = 2^N * \{(\Delta V_{AIN} / (V_{REFH} - V_{REFL})) * e^{-(1 / (f_{SAMP} * R_{AS} * C_{AS}))} + (\Delta V_{SAMP} / (V_{REFH} - V_{REFL})) * \text{Min}[(C_{AIN} / (C_{AIN} + C_{AS})), e^{-(1 / (f_{ATDCLK} * (R_{AS} + R_{AIN}) * C_{AIN}))}]\}$$

This is a worst case sampling error which does not account for R_{AS} recharging the combination of C_{AS} and C_{AIN} during the sample window. It does illustrate that high values of R_{AS} (>10 kΩ) are possible if a large C_{AS} is used and sufficient time to recharge C_{AS} is provided between samples. In order to achieve accuracy specified under the worst case conditions of maximum

ΔV_{SAMP} and minimum C_{AS}, R_{AS} must be less than the maximum value of 10 kΩ. The maximum value of 10 kΩ for R_{AS} is to ensure low sampling error in the worst case condition of maximum ΔV_{SAMP} and minimum C_{AS}.

Summary of Specification Parameters

The following terms and simplified definitions will be used to describe Motorola's future ADC's:

- **Zero-Scale Voltage (V_{ZS})** — The voltage at the bottom end of the input range (equal to V_{REFL}), measured in Volts.
- **Full-Scale Voltage (V_{FS})** — The voltage at the upper end of the input range (equal to V_{REFH}), measured in Volts.
- **Least Significant Bit (LSB)** — An ideal code width, or the valid input range divided by the number of possible codes.
- **Resolution (N)** — The number of bits in the output conversion; defines the number of possible codes.
- **Quantization Error (E_Q)** — The error between the Ideal Transfer Function and the Ideal Straight Line Transfer Function as a result of the quantization of the output, measured in LSB.
- **Differential Non-Linearity (DNL)** — The error between the current code width and the ideal code width, measured in LSB.
- **Integral Non-Linearity (INL)** — The error between the actual and the corresponding Adjusted Transfer Function transition voltages, measured in LSB.
- **Zero-Scale Error (E_{ZS})** — The error between the actual and ideal first transition voltages, measured in LSB.
- **Full-Scale Error (E_{FS})** — The error between the actual and ideal last transition voltages, measured in LSB.
- ***(Alternatively to E_{FS})* Gain Error (E_G)** — The error between the voltage range the Actual Transfer Function covers from first to last transition and that of the Ideal Transfer Function, measured in LSB.
- **Total Unadjusted Error (E_{TUE})** — The error between the Actual Transfer Function and the Ideal Straight-Line Transfer Function, measured in LSB.

Common ADC Terminology

The following words are often used to describe ADC performance:

- **Absolute Error** — Total Unadjusted Error.
- **Actual Transfer Function** — The conversion code vs. input voltage curve.
- **Adjusted Code Width (ACW)** — The amount of the input range (in Volts) that the ADC would convert to each code assuming a perfectly linear transfer function between the first and last transition voltages.
- **Adjusted Gain Error** — Gain Error.
- **Adjusted Integral Non-Linearity** — Integral Non-Linearity.
- **Adjusted Offset Error** — Offset Error.
- **Adjusted Straight-Line Transfer Function** — The straight line between the minimum voltage plus Zero-Scale Error and the maximum voltage minus Full-Scale Error.
- **Adjusted Transfer Function** — The ideal conversion code vs. input voltage curve assuming perfect linearity between the first and last transition voltages.
- **Analog AC Source Impedance (Z_{AIN})** — The real portion of the Analog Input Source Impedance at the Acquisition Frequency; determines how fast the Analog Input Source can charge the ADC input.
- **Analog DC Source Resistance (RAIN) —** The real portion of the Analog Input Source Impedance at DC; determines the voltage drop from the true source to the analog input given the input leakage.
- **Best-Fit Transfer Function** — The transfer function for which the average error between it and the actual transfer function is minimum.
- **Board Resistance (R_B) —** The resistance (leakage) on the Analog Input due to the printed circuit board.
- **Code** — The digital output of the ADC.
- **Code DNL** — The DNL at a particular conversion.
- **Code Error** — The error between the current conversion and the ideal conversion.
- **Conversion —** The single point transfer function between V_{IN} and the output code.
- **Conversion Code Width (CCW)** — The difference between the voltages at which the conversion transitions to and from the specified code.
- **Differential Non-Linearity (DNL)** — The difference between the Current Code Width and the Ideal Code Width.
- **Endpoint INL** — Integral Non-Linearity.

- **Full-Scale Conversion** — The maximum code output by the ADC, typically $3FF for a 10b ADC.
- **Full-Scale Error (E_{FS})** — The difference between the ideal last transition and the actual last transition voltages.
- **Full-Scale Offset** — Full-Scale Error.
- **Full-Scale Voltage (V_{FS})** — The maximum of the input voltage range, typically equal to V_{REFH}.
- **Gain Error (E_G)** — The difference in the ideal ADC slope and the slope of the ADC transfer function, measured as the difference in the number of Ideal Code Widths from the first to last conversion between the ideal transfer function and the actual.
- **Ideal Code Width (ICW) —** The amount of the input range (in Volts) that an ideal ADC would convert to each code.
- **Ideal Straight-Line Transfer Function** — The straight line from the minimum to the maximum voltage.
- **Ideal Transfer Function** — The ideal conversion code vs. input voltage curve assuming perfect linearity across the input voltage range.
- **Input Leakage (I_{IN}) —** The leakage into the pad (not ADC), usually due to ESD circuits.
- **Integral Non-Linearity (INL)** — The sum of the individual Code DNL, except for the Zero-Scale Error, up to the current code, or the difference between the current transition and the corresponding transition on the Adjusted Transfer Function.
- **Least Significant Bit (LSB)** — A unit of voltage equal to the valid input range divided by the number of possible output codes. The bit has the smallest value.
- **Most Significant Bit (MSB)** — The bit having the largest value. Its value will be ½ of full scale.
- **Offset Error (E_O)** — The difference between the actual and ideal first transition voltages.
- **Quantization Error (E_Q)** — The error between the Straight-Line Transfer Function and corresponding Transfer Function as a result of Quantization; only included in Total Unadjusted Error.
- **Resolution** — The number of bits in the output code of the ADC, this term has no real bearing on the performance of the ADC except that all performance parameters are measured against the theoretical best ADC of equal resolution.
- **Straight-Line Transfer Function** — The code vs. input voltage that would define an ADC with an infinite number of codes.
- **Total Error** — Total Unadjusted Error.
- **Total Adjusted Error** — The difference between the Actual Transfer Function and the Adjusted Straight-Line Transfer Function; includes INL and E_Q.

- **Total Unadjusted Error (E_{TUE})** — The difference between the Actual Transfer Function and the Ideal Straight-Line Transfer Function; includes all forms of error.
- **Transfer Function** — The code vs. voltage relationship that defines the ADC output vs. input function.
- **Transition Voltage (or Point)** — The voltage at which the ADC output changes from one code to another.
- **Unadjusted INL** — The sum of the Code DNL up to the current conversion, or the difference between the current transition and the corresponding transition on the Ideal Transfer Curve.
- **Uncompensated Quantization** — Technique in which the ideal transfer function's first transition voltage is at 1LSB.
- **V_{IN}** — The input voltage to the ADC.
- **V_{REFL}** — The low-voltage reference to the ADC analog circuitry, typically (but not always) equal to V_{SSAD}.
- **V_{REFH}** — The high-voltage reference to the ADC analog circuitry, typically allowed to go lower than V_{DDAD} while maintaining a minimum voltage difference between it and V_{REFL}.
- **Zero-Scale Conversion** — The minimum code output by the ADC, typically $000 for a 10b ADC.
- **Zero-Scale Error (E_{ZS})** — The difference between the first transition and the ideal first transition voltages.
- **Zero-Scale Voltage (V_{ZS})** — The minimum of the input voltage range, typically equal to V_{REFL}.
- **½LSB Compensated Quantization** — Technique in which the ideal transfer function's first transition voltage is shifted to cause the first transition voltage at ½LSB instead of 1LSB.

Conclusion

Once a user has a good working knowledge of the various terminology used to describe the specifications and characteristics of analog-to-digital converters, he can select a converter to best suite the requirements of the system. A good understanding of the terms used to specify the converter can help the user sort out the most important design parameters for his system and enable him to get the best cost and performance trade-offs for the design.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://motorola.com/semiconductors

MOTOROLA

AN2438/D
Rev. 0
2/2003