

ADC & DAC Errors

The purpose here is to familiarize the reader **DC Errors** of DACs and ADCs. Although Full-Scale and Offset Errors are DC Errors, these will not be covered here because they have been covered in the first course.

● **Course Map/Table of Contents**

- 1. [Course Navigation](#)
 - 1. [1.1 Course Navigation](#)
- 2. [Quantization and Quantization Error](#)
 - 1. [2.1 Quantization](#)
 - 2. [2.2 The ADC Transfer Curve](#)
 - 3. [2.3 Quantization Error](#)
 - 4. [2.4 Adding ½ LSB Offset to the ADC input](#)
 - 5. [2.5 ADC Quantization Error with ½ LSB Offset](#)
 - 6. [2.6 The DAC Transfer Curve](#)
- 3. [Linearity](#)
 - 1. [3.1 Linearity - DNL \(DLE\) and INL \(ILE\)](#)
 - 2. [3.2 DNL of an ADC](#)
 - 3. [3.3 DNL of a DAC](#)
 - 4. [3.4 ADC Missing Codes](#)
 - 5. [3.5 DAC Monotonicity](#)
 - 6. [3.6 Integral Non-Linearity \(INL\)](#)
 - 7. [3.7 INL Measurement Methods](#)
 - 8. [3.8 Total Unadjusted Error](#)
- 4. [Course Summary](#)
 - 1. [4.1 ADC & DAC Errors Summary](#)


Course Navigation

1.1 Course Navigation



Course Navigation

This course is organized like a book with multiple chapters. Each chapter may have one or more pages.

- The previous and next arrows move you forward and back through the course page by page. 
- The left navigation bar takes you to any chapter. It also contains the bookmarking buttons, 'save' and 'go to.' To save your place in a course, press the 'save' button. The next time you open the course, clicking on 'go to' will take you to the page you saved or bookmarked.
- The 'Go to Final Test' button on the left navigation bar takes you back to the Analog University course listing, where you started. Take the course final test by clicking on 'Test Yourself.'
- The top services bar contains additional information such as glossary of terms, who to go to for help with this subject and an FAQ. Clicking home on this bar will take you back to the course beginning.
- Don't miss the hints, references, exercises and quizzes which appear at the bottom of some pages.



Quantization and Quantization Error

The purpose of this course is to familiarize the reader with DC errors associated with DACs (Digital-to-Analog Converters) and ADCs (Analog-to-Digital Converters). Although Full-Scale and Offset Errors are DC errors, these will not be covered here because they have been covered in

the first course.

At the end of every page there is a quiz to confirm your understanding of the material. This quiz is not your exam for the material and there are no reporting of the results. To be sure that you see all of the material, be sure to continue down the page until you see the quiz.

- 2.1 Quantization
- 2.2 The ADC Transfer Curve
- 2.3 Quantization Error
- 2.4 Adding $\frac{1}{2}$ LSB Offset to the ADC input
- 2.5 ADC Quantization Error with $\frac{1}{2}$ LSB Offset
- 2.6 The DAC Transfer Curve



Quantization

What are "**Quantization**" and "**Quantization Error**", how they are affected by resolution and how do they affect ADC performance?

- **Quantization** is the process of assigning various input ranges to discrete output codes
 - ▶ A device which does this is called a "**Quantizer**".
 - ▶ An Analog-to-Digital Converter is a **Quantizer**.
- Whenever we digitize an analog quantity (voltage, current, temperature, pressure, etc.) we assign a single digital code to a range of analog values (a "**quanta**"). That is, whatever analog values that fall between some values x_m and x_n will all be assigned the same digital code at the output of the quantizer, the ADC (also termed A/D Converter or Analog-to-Digital Converter). All analog input values between other values x_{m1} and x_{n1} will all be assigned a different digital code from those between x_m and x_n . This gives rise to the commonly seen stair step transfer function of the ADC.

Type of Device



A Quantizer is also known as

1

1. Analog-to-Digital Converter
2. Digital to Position Converter
3. Both of these are correct
4. Neither of these is correct

1 Answer: 1. Analog-to-Digital Converter



2 A range of analog values that is assigned a single digital code is known as a

1. Groupie
2. Quanta
3. Both of these are correct
4. Neither of these is correct

2 Answer: 2. Quanta



To decrease the size of the Quanta we can

3

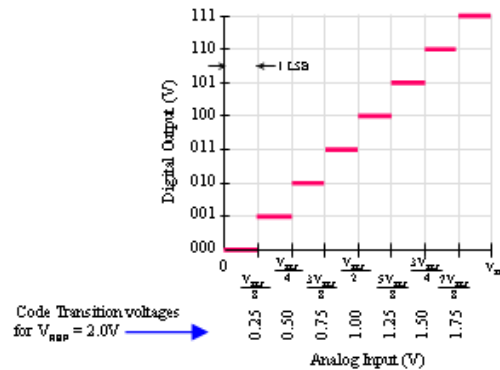
1. Lower the Reference Voltage or Current
2. Use a converter with more bits
3. Both of these are correct
4. Neither of these is correct

3 Answer: 3. Both of these are correct

The ADC Transfer Curve

The shape of the ADC Transfer Curve

- A simple 3-bit ADC will serve to illustrate the ADC **transfer curve**, which takes on the staircase characteristic seen here.



- Note that the familiar rising lines of the staircase are missing.
 - This is as it should be because the ADC output code makes jumps from code to code. However, we commonly see the transfer curve graphed as a stair step with both rising and horizontal lines.
 - A slewing from one digital code to another is not possible with a digital word.
- The code transition voltages shown here for a 2.0 Volt reference are, of course, for a 3-bit ADC.
 - Since 3 bits provide a total of $2^3 = 8$ possible codes, each code transition occurs at intervals of $V_{REF}/8$.
 - For a 2.0 Volt reference, these intervals, or **quanta**, are each $V_{REF}/8 = 2.0/8 = 0.25V$. A **quanta**, then, is the same size as one LSB.



While the familiar staircase of the ADC transfer curve that we commonly see shows both rising and falling lines, the true transfer curve is missing the vertical rising lines.

ADC Transfer Function



1 The ADC Transfer Curve is

- A stair step with horizontal and vertical lines
- A series of horizontal lines
- Answer #2 above is correct, but we commonly see #1 above being graphed
- None of these is correct

1 Answer: 3: Answer #2 above is correct, but we commonly see #1 above being graphed



2 A n-bit ADC has how many possible output codes?

- $6.02n$
- 2^n
- 10^n
- None of these is correct

2 Answer: 2: 2^n



3 Each ADC code transition occurs at intervals of

- $V_{REF}/2^n$

1. $V_{REF}/4$

2. 1 LSB

3. Both of these are correct

4. Neither of these is correct

3 Answer: 3: Both of these are correct

Q

4. A *quanta* is equivalent in size to one LSB

1. True

2. False

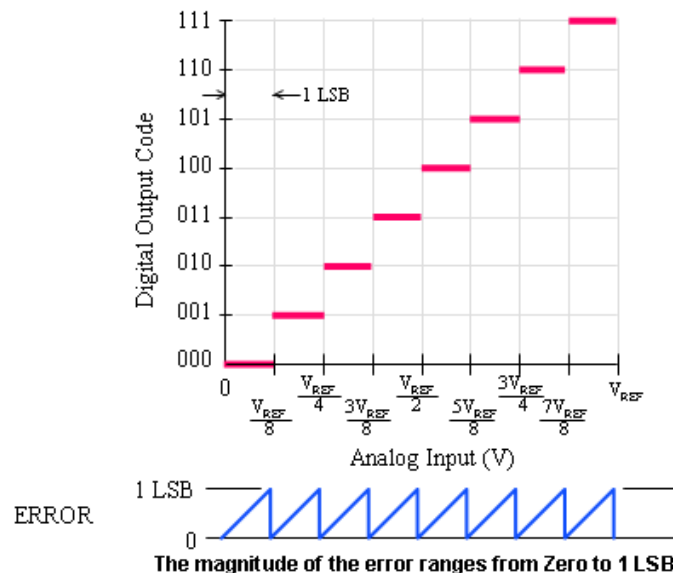
4 Answer: 1: True

3.

Quantization Error

Quantization Error defined

- Our simple 3-bit ADC can also illustrate **Quantization Error**.



- ▶ With an ADC input of zero, the output code is zero (000b) and there is no error.
- ▶ As the input voltage increases towards $V_{REF}/8$ ($V_{REF}/2^3$), the error increases because the input is no longer zero, but the output code remains at zero. This is because an input voltage *range* is represented by a single output code, as is necessary when interfacing between the analog and digital worlds.
- ▶ When the input reaches $V_{REF}/8$, the output code changes from 000b to 001b, where the output exactly represents the input voltage and the error reduces to zero.
- ▶ As the input voltage increases past $V_{REF}/8$, the error again increases until the input voltage reaches $V_{REF}/4$, where the error again drops to zero. This process continues through the entire input range and the error plot is a saw tooth.
- The fact that a *range* of input voltages, or *quanta*, is represented by and converted to a single code is what we call **quantization**. The maximum error we have here with the ideal ADC is 1 LSB.
 - ▶ This range or errors is known as **quantization uncertainty** because it is the range of analog input values that could have caused a given code and when we look at the digital word we are **uncertain** as to exactly what the input voltage was that was converted to that code.
 - ▶ Since this error results from the quantization process, another name for this maximum error due to the quantization process is "**Quantization Error**".

▶ Our simple ADC can also illustrate **Quantization Error**.

- ▶ Since the ADC can only *resolve* the input into 2^n discrete values, the converter resolution is 1 in 2^n .
- ▶ The quantization error can never be reduced below $\frac{1}{2}$ **LSB**
- ▶ For an 2 Volt reference (with a unity gain factor), a 3 -bit converter *resolves* the input into $V_{REF}/8 = 2V/8 = 0.25$ Volt steps. We can say that the converter "**resolution**" is **8 bits**, or we can say, in this example, that it is **0.25 Volt**. Quantization error is a round off error.
- An error of 0 to 1 LSB is not as desirable as would be an error of $\pm\frac{1}{2}$ **LSB** because an error range of 0 to 1 LSB means that the converted word has a maximum error of 1 LSB from the actual input value, whereas an error of $\pm\frac{1}{2}$ **LSB** means that the converted word has a maximum error is just $\frac{1}{2}$ **LSB** from the actual input value.
 - ▶ To get the $\pm\frac{1}{2}$ **LSB** error rather than the 0 to 1 LSB error, we introduce an offset into the A/D converter to force the error range to be $\pm\frac{1}{2}$ **LSB**.

Quantization error

Getting a handle on quantization error

- Q 1** The fact that a range of input voltages is represented by and converted to a single code is what we call
1. Resolution
 2. Quantization
 3. DACing
 4. None of these is correct

1 Answer: 2: Quantization

- Q 2** Quantization uncertainty arises from the fact that
1. A range of analog input values could have caused a given code
 2. The digitization process is perfect
 3. There is a 1 LSB error in the ADC output
 4. None of these is correct

2 Answer: 1: A range of analog input values that could have caused a given code

- Q 3** Quantization Error is
1. An error that can not be reduced below $1/2$ LSB
 2. The error caused by the digitization process
 3. Both of these are correct
 4. Neither of these is correct

3 Answer: 3: Both of these are correct

- Q 4** To say that a converter resolution is $V_{REF}/2^n$ is the same as saying its resolution is
1. $2/V_{REF}$
 2. $V_{REF}/2^n$
 3. Both of these are correct
 4. Neither of these are correct

4 Answer: 2: $V_{REF}/2^n$

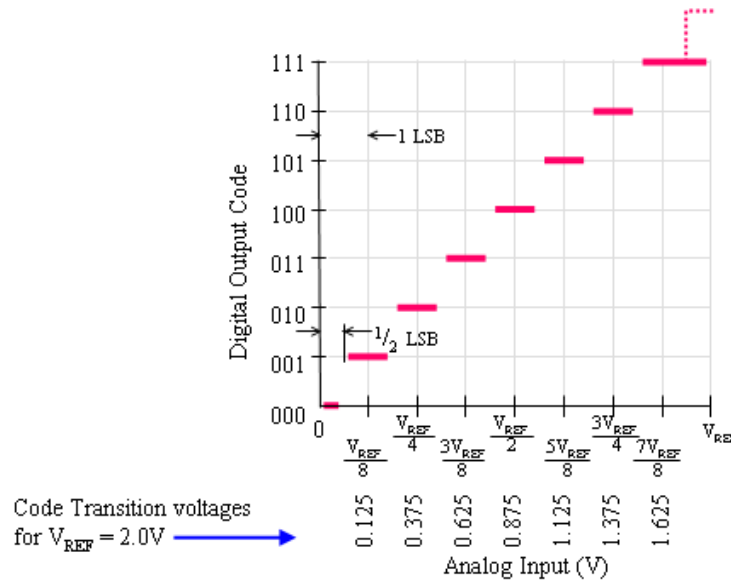
- Q 5** To get the $\pm 1/2$ LSB error rather than the 0 to 1 LSB error, we introduce an offset into the A/D converter to force the error range to be $\pm 1/2$ LSB.
1. True
 2. False

4.

Adding 1/2 LSB Offset to the ADC input

Getter a little more accuracy

- Rather than the transition from the code of zero to the first code being at 1 LSB, we add an offset of -1/2 LSB and the first code transition point is at an input level of 1/2 LSB, causing the output code to change when the input level is 1/2 LSB less than it would if there were no added offset.



- ▶ The output changes from 000b to 001b with an input value of 1/2 LSB rather than 1 LSB and the last code transition (from 110b to 111b) is at 1.5 LSB below V_{REF} .
- ▶ Note also that the center of the maximum output code (the "no error" point) corresponds to an input of one LSB less than the reference voltage. The implication here is that a higher resolution ADC will have a maximum output code that corresponds to an input level that is closer to the reference than would a lower resolution ADC, so better full-scale accuracy for higher resolution converters.
- The maximum error *range* we have here is 1 LSB. This -1/2 LSB to +1/2 LSB range or errors is still the same "quantization uncertainty" range of 1 LSB.



Adding a -1/2 LSB offset to the ADC input provides a little more accuracy because, in the ideal case, the maximum error is 1/2 LSB rather than 1 LSB.

The desire for a 1/2 LSB offset



1 The first code transition point in an ADC, in the ideal case, is at an input value of

1. -1 LSB
2. -1/2 LSB
3. +1/2 LSB
4. +1 LSB

1 Answer: 3: +1/2LSB



2 The transition to the last code, in the ideal case, is at an input value of

1. V_{REF} minus 1.5 LSB

2. V_{REF} minus $\frac{1}{2}$ LSB

3. V_{REF}

4. V_{REF} + $\frac{1}{2}$ LSB

2 Answer: 1: V_{REF} minus $1\frac{1}{2}$ LSB



3 The maximum output code with no error occurs with an input of 1 LSB below full-scale, implying

1. Greater full-scale accuracy with a higher resolution ADC
2. An input level causing the last code transition that is closer to V_{REF} for higher resolution ADCs
3. Both of these are correct
4. Neither of these is correct

3 Answer: 3: Both of these are correct



4 Whether or not the ADC has a built in offset, the quantization uncertainty is still 1 LSB.

1. True
2. False

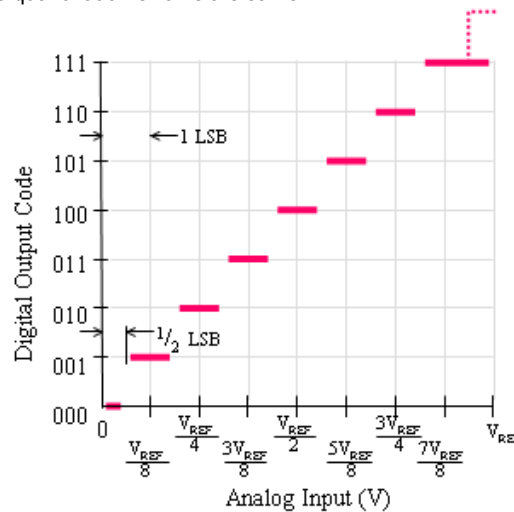
4 Answer: 1: True



ADC Quantization Error with $\frac{1}{2}$ LSB Offset

Why add a built in offset to the ADC

- The quantization error *range* with the addition of an offset at the input is the same as that error range without this added offset, but that the error is $\pm\frac{1}{2}$ LSB rather than 0 to +1 LSB. The quantization error is the same.



- ▶ With an input voltage of zero, the output code is zero (000b), as before. As the input voltage increases towards the $\pm\frac{1}{2}$ LSB level, the error increases because the input is no longer zero, but the output code remains at zero.
- ▶ When the input reaches $\frac{1}{2}$ the output code changes from 000b to 001b. The input is not yet at the 1 LSB level, but only at $\frac{1}{2}$ LSB, so the error is now $-\frac{1}{2}$ LSB.
- ▶ As the input increases past $\frac{1}{2}$ LSB, the error becomes less negative, until the input reaches 1 LSB, where the error is **zero**.
- ▶ As the input increases beyond 1 LSB, the error increases until the input reaches $1\frac{1}{2}$ LSB, where output code is incremented by one and the sign of the error again becomes negative. This process continues through the entire input range.



1 The quantization error is the same whether or not the ADC has a built in offset

1. True
2. False

1 Answer: 1: True



2 The code transition points DO NOT change when an input offset is built into the ADC

1. True
2. False

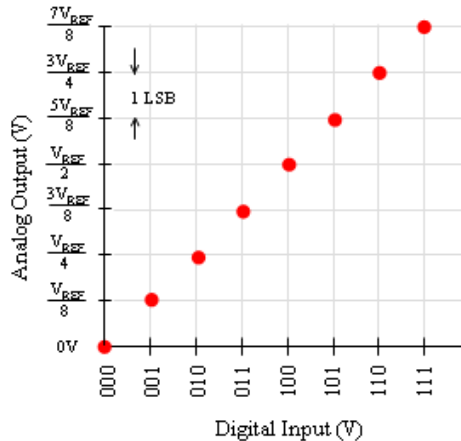
2 Answer: 2: False



The DAC Transfer Curve

The DAC transfer curve differs from that of the ADC.

- The DAC transfer curve is different from that of the ADC in that it is a series of points rather than a series of horizontal lines and the transfer curve of the DAC has no offset.
 - ▶ While the ADC input is analog and can be continuous, the DAC input is digital and discrete.
 - ▶ The continuous analog input and discrete output of the **ADC** means that its transfer function is a series of straight lines, as we have seen.
 - ▶ The **discrete nature** of both input and output of the **DAC**, however, means that both the input and the output are discrete, leading to a **series of points** on a plot of DAC output vs. input, as seen in the 3-bit DAC transfer plot here.



- As in the case of the ADC, it is common, although **incorrect**, to show this plot as a staircase.
 - ▶ If we looked at a DAC output with an oscilloscope, we would see the common staircase.
 - ▶ The horizontal lines of the staircase are caused by the time between loading of successive codes to the DAC and the vertical lines of the staircase are caused by the finite rise time of the the DAC output and/or the oscilloscope.
- The output values are shown above for a voltage output DAC, but a DAC can be designed to have either voltage or current output.
- Like the ADC, a DAC can be designed to have either a voltage or a current reference.
- Unlike the ADC, the DAC does not have a built-in offset, but a DAC does have a **quantization error** and can suffer from DNL and INL errors.

The DAC Transfer Curve



1 The DAC input, like that of the ADC, is continuous

1. True

1. true
2. False

1 Answer: 2: False



2 While the ADC transfer curve has discontinuous horizontal lines, the DAC transfer curve

1. Is a series of points
2. Is also discontinuous
3. Both of these are true
4. Neither of these is true

2 Answer: 3: Both of these are true



3 Like the ADC, the DAC has a built-in offset

1. True
2. False

3 Answer: 2: False



4 It is common, but somewhat incorrect, to plot the DAC transfer curve as a

1. Series of points
2. Staircase
3. Series of horizontal lines
4. None of these is correct

4 Answer: 2: Staircase



5 If we looked at a DAC output on an oscilloscope we would see a

1. Series of points
2. Staircase
3. Series of horizontal lines
4. None of these is correct

5 Answer: 2: Staircase

Linearity

Linearity parameters and how they affect converter performance

- 3.1 Linearity - DNL (DLE) and INL (ILE)
- 3.2 DNL of an ADC
- 3.3 DNL of a DAC
- 3.4 ADC Missing Codes
- 3.5 DAC Monotonicity
- 3.6 Integral Non-Linearity (INL)
- 3.7 INL Measurement Methods
- 3.8 Total Unadjusted Error



Linearity - DNL (DLE) and INL (ILE)

Differential and Integral Non-Linearity, while both are "Linearity" terms, are NOT the same, although they each have two different or names terms

for them.

- **DNL** (Differential Non-Linearity) and **DLE** (Differential Linearity Error) are different terms used to describe the **error in step size**.
- Similarly, **INL** (Integral Non-Linearity) and **ILE** (Integral Linearity Error) are different terms used to describe the maximum **deviation from the ideal straight line transfer function**.
- **DNL** for any given quanta is the **Actual Step Size minus the Ideal Step Size**.
 - ▶ For **positive DNL** we look at the **widest code**.
 - ▶ For **negative DNL** we look at the **narrowest code**.
- **INL** is the maximum deviation of the transfer curve from a straight line between two points along the input-output transfer curve. INL is also the sum of the DNL errors.
 - ▶ **INL** (ILE) describes the curve in the transfer function.
 - ▶ **INL** (ILE) of an ADC or DAC corresponds to the "linearity" of an amplifier.

Q

1 DNL is also known as

1. DLE
2. ILE
3. INL
4. None of these is correct

1 Answer: 1: DLE

Q

2 Positive and negative DNL are the

1. Average positive and negative DNL errors, respectively
2. The most positive and most negative or the individual DNL errors, respectively
3. The sum of all positive and the sum of all negative DNL errors, respectively
4. None of these is correct

2 Answer: 2: The most positive and most negative or the individual DNL errors, respectively

Q

3 INL is the

1. Maximum deviation of the transfer function from a straight line between two points
2. Sum of the DNL errors
3. Both of these are correct
4. Neither of these is correct

3 Answer: 3: Both of these are correct

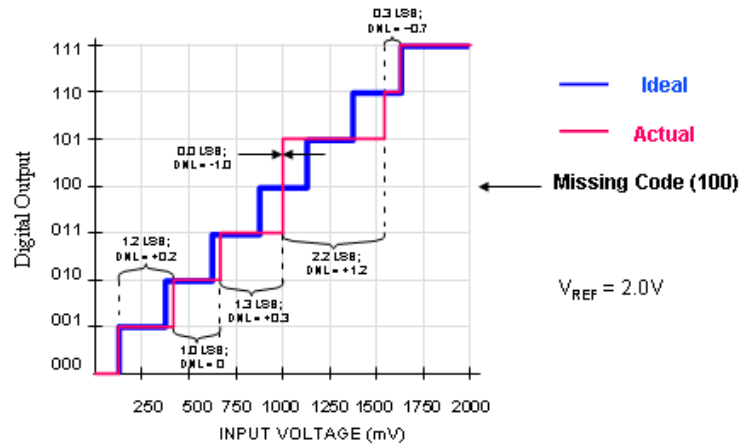
2.

DNL of an ADC

A discussion of DNL of the ADC

- In an ideal ADC, the code-to-code transition points are **exactly 1 LSB apart**.
 - ▶ These code-to-code transition points are, ideally, separated by $\text{FSR}/2^n$, where FSR is the Full Scale input Range and "n" is the ADC resolution in bits.
 - ▶ In an 8-bit ADC, for example, these changes are separated from each other by **1/256th** of full-scale.
 - ▶ The difference between the ideal 1 LSB and the worst case actual input voltage change between output code transitions is called **Differential Non-Linearity (DNL)**.

- **DNL** can be illustrated using the transfer function of a three-bit ADC, shown here. Each quanta (input voltage range for each step) should be precisely 1/8 of full-scale.



- ▶ In this example, the first code transition (from 000b to 001b) occurs at an input of $FS / 8$ (250mV for the 2 Volt reference example shown here), where FS is the full-scale input. This is exactly as it should be, so DNL for this step is **zero**.
- ▶ The second transition, from 001b to 010b, has an input change that is 1.2 LSB above the previous transition point, so this step is too large (larger than 1.0 LSB) by 0.2 LSB and we say DNL for this step is **+0.2 LSB**.
- ▶ The input change for the third transition is exactly the right size, so we say DNL for this step is **zero**.
- ▶ The digital output remains constant when the input voltage changes from 1000mV to beyond 1500mV and the code 100b *can never appear* at the output.
- ▶ The output change that causes a transition from code 011b to code 100b is so small that it does not exist, so the DNL error is **-1.0 LSB** and the code 100b is a **missing code** as it can never appear in the output.
 - ◆ To avoid missing codes in the transfer function, DNL should be **more positive than -1.0 LSB**.
- ▶ Continuing with the other steps, we see that the input range for code 101b is 2.2 LSB, or 1.2 LSB greater than the ideal 1.0 LSB, so the DNL error for this code is **+1.2 LSB**.
- ▶ Code 110b has a width of 0.3 LSB, so its DNL error is $1.0 - 0.3 = -0.7$ **LSB**.
- ▶ Since the DNL specification is the **worst positive** and **worst negative** individual errors, in this case the DNL specification would be **-1.0 LSB, +1.2 LSB**.
- ▶ Again, **DNL** indicates the **deviation** from the ideal 1 LSB step size of the analog input signal corresponding to a code-to-code increment.
- **DNL**, a static specification, **relates to SNR**, a dynamic specification. However, noise performance can **NOT** be predicted from the DNL specification, except to say that **SNR tends to become worse as DNL departs from zero**.



DNL of an ADC can never be more negative than -1.



1 In an ideal ADC, the code-to-code transition points are exactly

- 1/2 LSB apart
- 1 LSB apart
- 1/2 LSB in size
- None of these is correct

1 Answer: 2: 1 LSB apart



2

The difference between the ideal 1 LSB and the worst case actual input voltage change between output code transitions is called

- Differential Non-Linearity
- Integral Non-Linearity

- 2. Integral Non-Linearity
- 3. Code-to-Code Error
- 4. None of these is correct

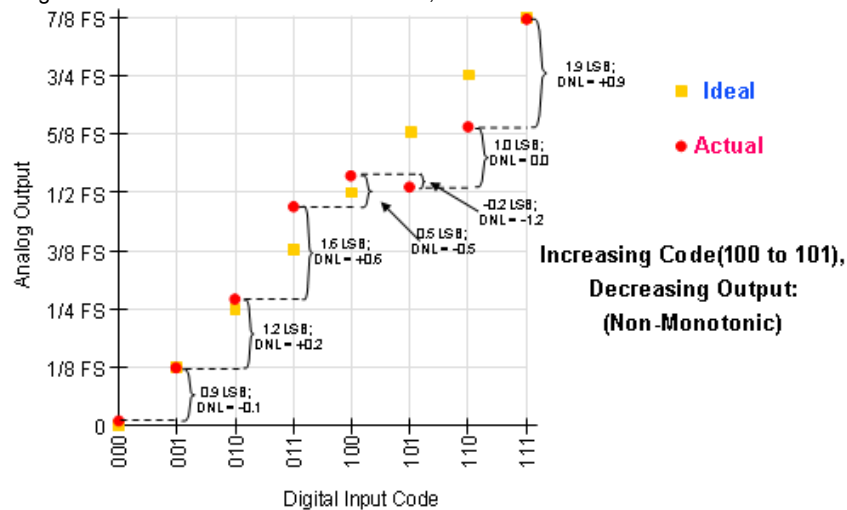
2 Answer: 1: Differential Non-Linearity

3.

DNL of a DAC

A discussion of DNL of the DAC

- In an ideal DAC, just as with the ideal ADC, the code-to-code transition points are exactly **1 LSB apart**.
 - ▶ These code-to-code transition points are, ideally, separated by $FSR/2^n$, where **FSR** is the **Full Scale Output Range** and "**n**" is the DAC resolution in bits.
 - In an 8-bit DAC, for example, these changes are separated from each other by 1/256 of full-scale.
 - ▶ The difference between the ideal 1 LSB and the worst case actual output voltage change between output code transitions is the DAC Differential Non-Linearity.
- DNL can be illustrated using the transfer function of a three-bit DAC, shown here.



- ▶ Each output (voltage or current) range for each step should be precisely 1/8 of full-scale.
- ▶ In the example here the first code transition (caused by an input code change from 000b to 001b) causes an output change of 90% of FS/8, where FS is the full-scale output.

This is **0.1 LSB less than it should be**, so DNL for this step is **-0.1 LSB**.
- ▶ The second transition, caused by an input code from 001b to 010b, has an input change that is 1.2 LSB above the previous transition point, so this step is **too large by 0.2 LSB** and we say DNL for this step is **+0.2 LSB**.
- ▶ The input change for the third transition causes a 1.6 LSB change in the output, so the step is **0.6 LSB too large** and we say DNL for this step is **+0.6 LSB**.
- ▶ Jumping ahead to the 5th code transition (from 100b to 101b) we see that the analog output (voltage or current) **decreases by 0.2 LSB**. Since we expected an **increase of 1.0 LSB**, this step is **-1.2 LSB from the expected position**, so we say the DNL for this step is **-1.2 LSB**.
- ▶ An examination of all steps shows that the **most positive DNL step is +0.9 LSB** and the **most negative is -1.2 LSB**. The DNL spec in this case would be **-1.2 LSB, +0.9 LSB**.

DAC DNL

Q 1

In an ideal DAC, as opposed to the case with the ideal ADC, the code-to-code transition points are NOT exactly 1 LSB apart.

1. True

2. False

1 Answer: 2: False

Q

2 The code transition points in the ideal DAC are how far apart?

1. One half LSB
2. One LSB
3. One and a half LSB
4. None of these is correct

2 Answer: 2: One LSB

Q

3 The difference between the ideal 1 LSB and the worst case actual output voltage change between output code transitions is the DAC

1. Integral Non-Linearity
2. Differential Non-Linearity
3. End Point Non-Linearity
4. None of these is correct

3 Answer: 2: Differential Non-Linearity

Q

4 The minimum and maximum output delta between adjacent codes of DAC is 80% of Full-Scale/ 2^n and 155% of Full-Scale/ 2^n , respectively. Which of the following is TRUE?

1. Minimum DNL is -0.3 LSB
2. Maximum DNL is +0.55 LSB
3. Minimum INL is -0.2 LSB
4. Maximum INL is +0.55 LSB

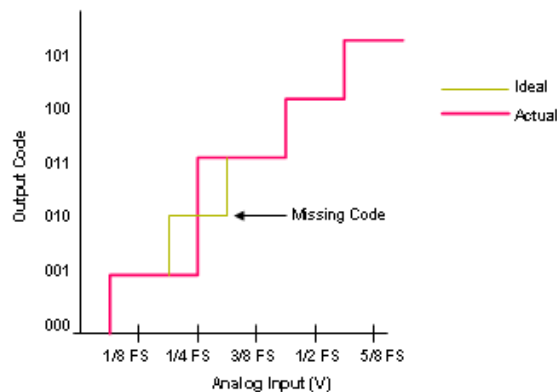
4 Answer: 2: Maximum DNL is +0.55 LSB

4.

ADC Missing Codes

A negative DNL of -1 means there is one or more missing codes in the ADC

- When no value of input voltage will produce a given output code, meaning that the code in question never appears in the output, that code is missing from the transfer function and is known as a missing code, as we have seen before.
- Again, the transfer function here is for a three-bit A/D converter. The first code transition, from 000b to 001b, takes place when the input voltage is $\frac{1}{2}$ LSB, which is correct for an A/D converter.



- ▶ The second transition takes place when the input voltage reaches $\frac{1}{4}$ FS, so the differential linearity error at that point is ± 1 LSB. The second transition has a differential linearity error of -1 LSB, causing the output code to jump from 001b

is $\pm 1/2$ LSB. The second transition has a differential linearity error of -1 LSB, causing the output code to jump from 0010 to 011b, and 010b is a Missing Code.

► Any time DNL is -1.0, there is a one or more missing codes.

- A DNL that is less than -1.0 LSB means that part of the transfer function takes on a negative slope.

This condition is virtually non-existent in modern ADCs, limiting the negative DNL to -1.0 LSB.

- Many A/D converter data sheets specify "no missing codes" as this specification can be critical in some applications, such as in servo systems.
- A "no missing codes" guarantee when the negative DNL spec is -1.0 means that the negative DNL can not really reach -1.0 , but it can come so close that, for all practical purposes, the code in question does not really exist, even with the "no missing codes" guarantee.
- A negative DNL of -0.99 LSB means that the quanta for at least one code is just 1% of what it should be and would mean that quanta is so small that it virtually does not exist.
- For example, a 10-bit ADC with a gain of one and a 2.0 Volt reference and a negative DNL of -0.99 LSB would have a quanta for at least one code of just 19.5 microvolts, which is more than likely in the noise floor, so it would be virtually impossible to obtain one or more output codes. While this code may *theoretically* exist in the transfer function, it is so narrow that it is missing *for all practical purposes*. A DNL any more negative than about -0.85 to -0.90 may as well be -1.0 .

► Specmanship! Gotta HATE it!

ADC Missing Codes



1 A missing code in an ADC means that the minimum DNL has reached a value of

1. -0.5 LSB
2. $+0.5$ LSB
3. -1.0 LSB
4. $+1.0$ LSB

1 Answer: 3: -1.0 LSB



2 For all practical purposes, an ADC has at least one missing code if DNL reaches a value of

1. $+1.0$ LSB
2. -0.5 LSB
3. -0.66 LSB
4. -0.99 LSB

2 Answer: 4: -0.99 LSB



DAC Monotonicity

The transfer function of a monotonic DAC has a slope that never changes sign (is always either positive or negative)

- Missing codes in an ADC is analogous to Non-Monotonic behavior in a DAC.
- We say a DAC is Non-Monotonic when the sign of the slope of the input-output characteristic is not constant. That is, an increase in the input code at some point causes the output voltage or current to decrease when it should be increasing (or to increase when it should be decreasing).
- Any time the DNL of a DAC is more negative than -1.0 LSB there is a change in the slope of the transfer function and we say the DAC at this code transition is non-monotonic.
- A non-monotonic condition is to a DAC what a missing code is to an ADC.
- With the ADC, a negative DNL equal or very close to -1.0 LSB indicates the presence of at least one missing code.
- With a DAC, a DNL error more negative than -1.0 indicates a non-monotonic condition.

- A non-monotonic DAC used to construct an ADC will lead to missing codes in that ADC.
- While the minimum DNL of an ADC can not be less than -1.0 LSB, the minimum DNL of a DAC can be less than -1.0 LSB.



While DNL for an ADC can never be more negative than -1 LSB, DNL in a DAC **can** be more negative than -1 LSB.

DAC Monotonicity



1 When the sign of the slope of the input-output characteristic of a DAC is not constant, we say the DAC is

1. An Offset DAC
2. Non-Monotonic
3. Both of these are correct
4. Neither of these is correct

1 Answer: 2: Non-Monotonic



2 There is a change in the slope of the DAC transfer function whenever

1. INL is more positive than +1.0 LSB
2. INL is more negative than -1.0 LSB
3. DNL is more positive than +1.0 LSB
4. DNL is more negative than -1.0 LSB

2 Answer: 4: DNL is more negative than -1.0 LSB



3 A non-monotonic condition is to a DAC what a missing code is to an ADC.

1. True
2. False

3 Answer: 1: True



4

As is the case for the ADC, DNL indicates the deviation from the ideal 1 LSB step size of the analog voltage or current corresponding to a code-to-code increment.

1. True
2. False

4 Answer: 1: True



5 Like in the ADC, DNL of a DAC can never be more negative than -1.0 LSB.

1. True
2. False

5 Answer: 2: False

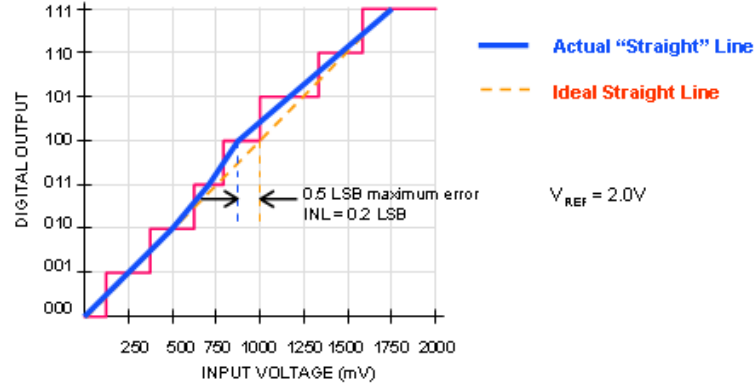


Integral Non-Linearity (INL)

INL defined and discussed

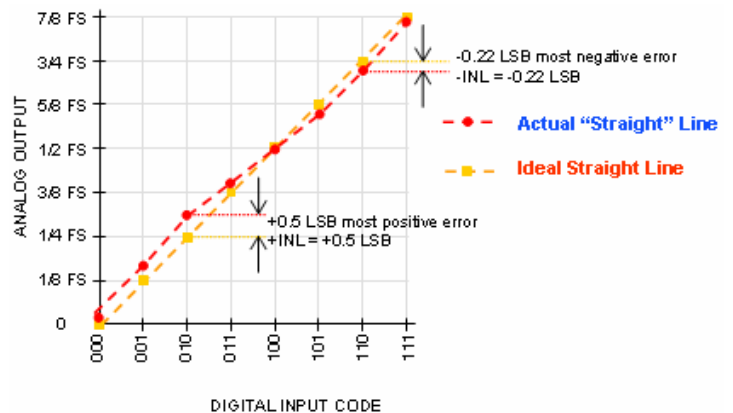
- Integral Non-linearity, INL, (also called Integral Linearity Error or ILE and Linearity Error or LE) describes the **departure from an ideal linear transfer curve** for an ADC or a DAC.

- ▶ INL does not include quantization errors, offset error, or gain error.
- ▶ INL is a measure of the straightness of the transfer function and can be greater than the differential non-linearity.
- ▶ The size and distribution of the DNL errors will determine the Integral Linearity (INL) of the converter.
- ▶ INL is measured from the ideal centers of the code ranges to the actual centers of the code ranges.



- Sometimes a converter is described as being "x bits linear."
 - ▶ For example, a converter with 10-bit resolution and 4 LSB non-linearity is sometimes described as an "8-bit linear" converter because 4 LSBs for a 10-bit device is the same as 1 LSB for an 8-bit device.
 - ▶ This arises out of the fact that, for a given reference voltage and gain factor, one LSB at 8 bits is the same code range as 4 LSB at 10 bits.
- INL is a static specification and relates to THD, a dynamic specification.
 - ▶ However, distortion performance can not be predicted from the INL specification, except to say that THD TENDS to become worse as INL departs from zero.

- Integral Non-linearity is essentially the same for DACs as it is for ADCs. It describes the departure from an ideal linear transfer curve as compared with the ideal straight line transfer curve.



- ▶ INL is measured from the ideal code points to the actual code points.

Q

1 Which of these is True?

1. INL and ILE are different names for the same thing
2. INL does not include quantization errors, offset error, or gain error
3. Both of these are true
4. Neither of these is true

1 Answer: 3: Both of these are true

Q

2 INL is a static specification and relates to the dynamic specification of

1. SNR
2. THD
3. SINAD

2 Answer: 2: THD

Q

3 THD can be calculated from

1. INL
2. DNL
3. Both INL and DNL
4. None of these is correct

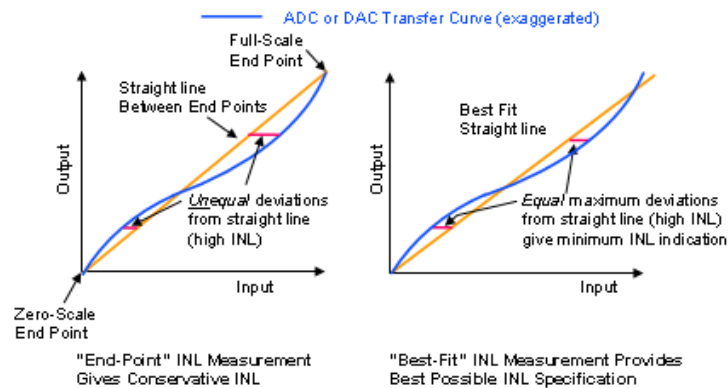
3 Answer: 4: None of these is correct

7.

INL Measurement Methods

There are two methods of measuring INL. They do not provide the same results.

- There are two methods of measuring Integral Non-Linearity (INL) and both apply equally to ADCs and DACs. These are the "Best-Fit" and the "End Point" methods.



- The Best Fit method allows for a better INL specification, while the End Point method is more conservative.
- One argument for the Best Fit method is that the customer can adjust his circuit to actually realize this low INL, achieving better overall performance.

The problem in doing this, however, is that each board must be adjusted for minimum INL for each individual converter, which is time-consuming and, therefore, expensive and not considered desirable or practical by most converter users.

- ▶ Another argument for the Best Fit method is only for dynamic applications and says that these applications are not concerned with offset and gain errors (which cause the End Point and Best-Fit INL measurements to diverge), unless the offset and gain errors are very large, and the Best Fit method is more meaningful for these applications.
- ▶ This argument does indeed have some merit for dynamic applications, except they are meaningful only in that they are a better predictor of THD performance, which is usually specified anyway for converters intended for dynamic applications.
- The End Point method tells the user the worst case INL he can expect if he simply considers the actual ADC or DAC transfer function to be a straight line between the two end points.

Hence, the End Point method is seen as more practical by many. The End Point method is certainly more conservative than is the Best Fit method.

- Comparing the INL of two competing devices is not reasonable when one device is measured using the end-point method and the other device uses the Best Fit method because there is no correlation between the two methods.
- Generally, ADCs and DACs used in d.c. applications (like most applications for general purpose ADCs) should have INL specified with the End Point method because this method more closely tests the device in the manner in which it will most likely be used.
- This author believes that it does not really matter with which INL method is specified for ADCs used in many strictly dynamic applications as these applications do not really care about INL per se. However, sometimes a dynamic application is also concerned with d.c. performance, in which case the End Point method is preferred.

The bottom line is that the Best Fit method is marginally useful, at best and serves primarily to provide a better data sheet spec than does the End Point method. Whenever INL is of major importance, it is better to have an End Point INL specification.



National Semiconductor uses the End-Point method for most of our ADCs.

INL Methods



1 How many INL test methods are there?

1. One
2. Two
3. Three
4. Four

1 Answer: 2: Two



2 Which INL test method allows the supplier to show the better INL specification?

1. End Point
2. Best-Fit
3. Quantizing
4. None of these

2 Answer: 2: Best-Fit



3 Which INL test method is the more conservative?

1. End Point
2. Best-Fit
3. Quantizing
4. None of these

3 Answer: 1: End Point



4 It is reasonable to compare the INL of two devices when they are tested with different methods

1. True
2. False

4 Answer: 2: False

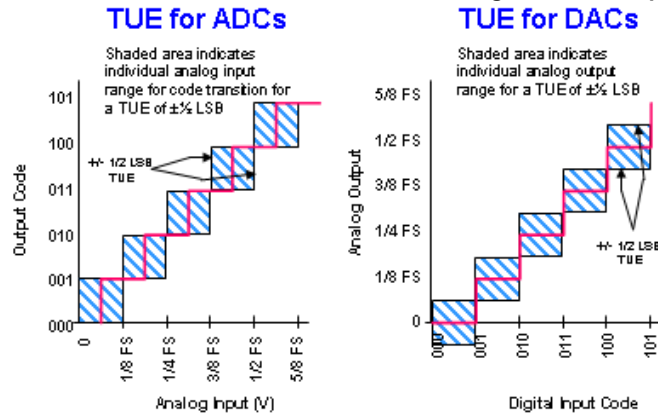


Total Unadjusted Error

Total Unadjusted Error (TUE) defined and discussed

- **When a designer wishes to meet a specific error budget, it is desirable to have a single specification that places a limit on errors from all sources.**
 - ▶ **Although gain and offset errors can be trimmed externally, trimming increases costs and sometimes reduces reliability.**
 - ▶ **When a designer wishes to meet a specific error budget, it is desirable to have a single specification that places a limit on errors from all sources.**

- Total Unadjusted Error (TUE) is a comprehensive static (DC) specification that includes linearity errors, gain error, and offset error and serves the above needs.
- TUE is the worst-case deviation from the ideal device transfer curve, assuming ideal the end points.



- TUE is a static specification. That is, it is useful for applications with DC or slowly moving input signals. Such applications include, for example, digitizing the outputs of weigh scales and of pH, temperature and pressure sensors.
- You won't find this specification on all ADC product data sheets; it is of value only when the total error specification is less than one or two LSB, so it is generally not found on data sheets of converters with higher resolution than eight-bits, or perhaps ten-bits.
 - ▶ The ADCS7478 and the ADC081S101, for example, are both specified at ± 0.3 LSB Total Unadjusted Error while the 10- and 12-bit devices in these families do not have a TUE specification.
 - ▶ The ADC10738 and the ADC12138 families of ADCs are examples 10-bit and 12-bit ADCs with a TUE specification.
- If the total unadjusted error is much larger than about 1 or 2 LSB, it makes more sense to specify separate data sheet limits for each of the component errors (Offset Error, Gain Error and INL).

Otherwise, a device with $\pm 1/2$ LSB linearity and ± 3 LSB full-scale error might be classified simply as an ADC with "3 LSB linearity" and the user wouldn't know that the device could provide excellent performance in applications that require linearity but don't need offset or full-scale accuracy.

TUE



1 It is possible to externally trim Gain and Offset Errors

1. True
2. False

1 Answer: 1: True



2 External trimming can

1. Increase manufacturing costs
2. Reduce reliability
3. Both of these are correct
4. Neither of these is correct

2 Answer: 3: Both of these are correct



3 The spec that incorporates all d.c. errors (is a comprehensive static spec)

1. INL
2. Gain Error
3. TUE
4. None of these is correct

3 Answer: 3: TUE



4 The spec that tells us the worst deviation from the ideal device transfer curve is

1. INL
2. TUE
3. DNL
4. GE

4 Answer: 2: TUE



5 Nearly all ADCs specify TUE

1. True
2. False

5 Answer: 2: False



6 The TUE spec is generally found

1. On all general purpose ADCs
2. Only on some 8-bit ADCs
3. Only on high speed ADCs
4. On DACs only, never on ADCs

6 Answer: 2: Only on some 8-bit ADCs



7 If the total unadjusted error is much larger than about 1 or 2 LSB

1. The device is most highly desirable
2. It makes more sense to specify separate data sheet limits for the component errors
3. The device becomes too expensive and no one will want to buy and use it
4. The device becomes dirt cheap

7 Answer: 2: It makes more sense to specify separate data sheet limits for the component errors

Course Summary

4.1 ADC & DAC Errors Summary



ADC & DAC Errors Summary

- Quantization is the process of digitizing.
- Quantization Error is the error that results from the quantization process.
- Quantization Noise is that noise that results from the quantization process.
- Linearity is broken down into DNL (or DLE) and INL (or ILE). These are measures of the step size error and in the departure from an ideal straight line, respectively.
- Missing Codes are those codes that can never appear in the ADC output.
- End Point and Best Fit are test methods for INL. The readings of the two test methods are not compatible with each other.
- TUE (Total Unadjusted Error) is the ADC output deviation from the ideal straight line from the ideal end points.
- Monotonicity in a DAC is the condition where the slope of the transfer curve has a constant sign. That is, the slope is always positive or always negative. Non-monotonicity in a DAC is analogous to missing codes in an ADC.

The links below, as well as the Definitions of Terms course, will be of further help.



Best Fit Method

Method for testing INL where the transfer curve is compared with a straight line set to provide the best INL performance indication. Compare this with the "End Point" method.

DNL

Differential Non-Linearity (also known as DLE - Differential Linearity Error) describes the error in step size. It is equal to the expected step size in LSB minus the actual step size in LSB. The worst positive and worst negative values are used for the DNL specification.

Dynamic applications

Circuit applications where a dynamic signal is processed, as opposed to applications where slowly changing d.c. voltages are processed.

End Point Method

Method for testing INL where the transfer curve is compared with a straight line between the end points of the transfer curve. Compare this with the "Best Fit" method.

INL

Integral Non-Linearity (also known as ILE - Integral Linearity Error) describes the deviation of the ADC or DAC transfer curve from a straight line. As such, it defines the bow in the transfer curve. INL is determined by summing the individual DNL errors. The worst positive and worst negative values are used for the INL specification.

Missing Code

A code that can never appear at the output of the ADC. One or more codes will be missing when the negative DNL is -1.0 .

Monotonicity

Monotonicity in a DAC is the condition where the slope of the transfer curve has a constant sign. That is, the slope is always positive or always negative. Non-monotonicity in a DAC is analogous to missing codes in an ADC.

Quanta

The magnitude of the range of ADC input values that is assigned a single digital code.

Quantization

The process of digitizing, or of assigning a single digital code to a range of input values (a quanta).

Quantization Error

The error that results from the quantization (digitization process) process

Quantization Noise

The ADC noise that results from the Quantization Process. This noise is the floor below which we can never reduce the noise and is a function of the ADC resolution.

Quantization Uncertainty

When looking at the digital code of an ADC output, we know that any input within a range of input values could have produced that code and we are uncertain as to the exact input value that produced that code, so we have a degree of "uncertainty" as to what the input value was that produced the output code.

Quantizer

A device which quantizes its input. An Analog-to-Digital converter.

TUE

Total Unadjusted Error. This is a static (d.c.) specification that includes linearity errors, gain error, and offset error. TUE tells us the worst-case deviation from the ideal device transfer curve.

Frequently Asked Questions

Do you have a question? We may have already answered it. Check below to see if you can find the answer to your question.

Questions

Answers

● Contact/Help Information

For additional information on getting started go to http://www.national.com/analog/training/getting_started

To contact us, and send feedback go to

<http://wwwd.national.com/feedback/newfeed.nsf/newfeedback?openform&category=pwdesignuniv>

For Frequently Asked Questions go to

<http://www.national.com/analog/training/faqs>

Thank you,
PowerWise Design University Team