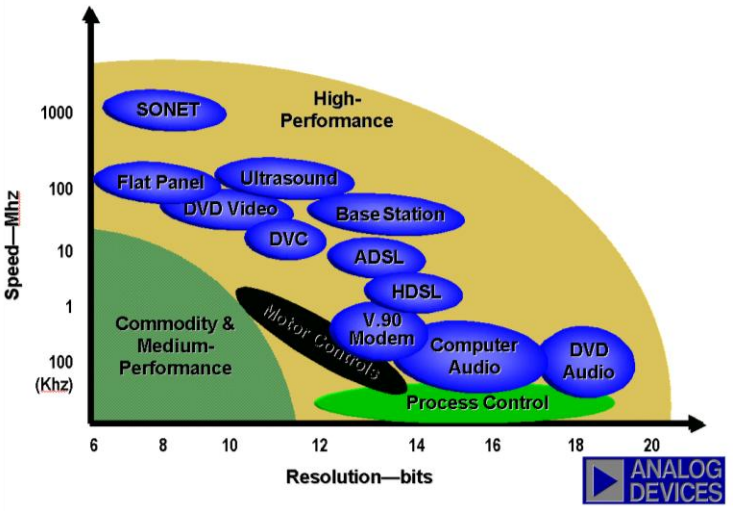
Material Covered in EE247
Where are We?

- ✓ Filters
  - Continuous-time filters
    - Biquads & ladder type filters
    - Opamp-RC, Opamp-MOSFET-C, gm-C filters
    - Automatic frequency tuning
  - Switched capacitor (SC) filters
- Data Converters
  - D/A converter architectures
  - A/D converter
    - Nyquist rate ADC- Flash, Pipeline ADCs,….
    - Self-calibration techniques
    - Oversampled converters
- Systems utilizing analog/digital interfaces

EE247
Lecture 11

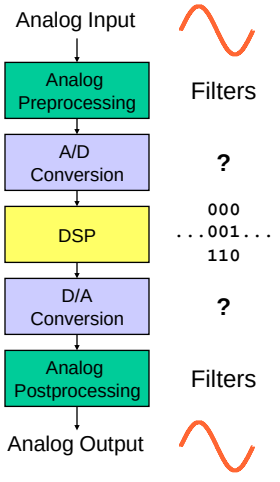
- Data converters
  - Areas of application
  - Data converter transfer characteristics
  - Sampling, aliasing, reconstruction
  - Amplitude quantization
  - Data converter performance metrics:
    - Static converter error sources
      - Offset
      - Full-scale error
      - Differential non-linearity (DNL)
      - Integral non-linearity (INL)
    - Measuring DNL & INL
      - Servo-loop
      - Code density testing (histogram testing)

Data Converter Applications



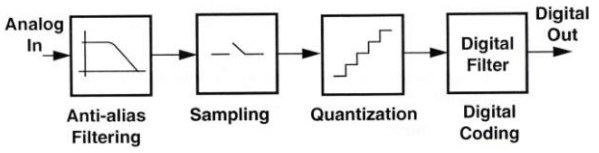
Data Converter Basics

- DSPs benefited from device scaling
- However, real world signals are still analog:
  - Continuous time
  - Continuous amplitude
- DSP can only process:
  - Discrete time
  - Discrete amplitude
    - → Need for data conversion from analog to digital and digital to analog

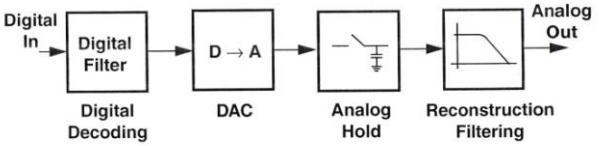


A/D & D/A Conversion

A/D Conversion



D/A Conversion



Data Converters

- Stand alone data converters
  - Used in variety of systems
  - Example: Analog Devices AD9235 12bit/ 65Ms/s ADC- Applications:
    - Ultrasound equipment
    - IF sampling in wireless receivers
    - Various hand-held measurement equipment
    - Low cost digital oscilloscopes

Data Converters

- Embedded data converters
  - Integration of data conversion interfaces along with DSPs and/or RF circuits → Cost, reliability, and performance
  - Main issues
    - Feasibility of integrating sensitive analog functions in a technology typically optimized for digital performance
    - Down scaling of supply voltage as a result of downscaling of feature sizes
    - Interference & spurious signal pick-up from on-chip digital circuitry and/or high frequency RF circuits
    - Portable applications dictate low power consumption

Embedded Converters
Example: Typical Dual-Standard Cell Phone

Contains in integrated form:

- 4 Rx filters
- 4 Tx filters
- 4 Rx ADCs
- 4 Tx DACs

} Dual Standard, I/Q

- 3 Auxiliary ADCs
- 8 Auxiliary DACs

} Audio, Tx/Rx power control, Battery charge control, display, ...

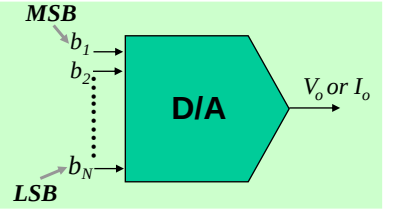
Total: Filters → 8

ADCs → 7

DACs → 12

D/A Converter Transfer Characteristics

- An ideal digital-to-analog converter:
  - Accepts digital inputs b_1-b_n
  - Produces either an analog output voltage or current
  - Assumption (will be revisited)
    - Uniform, binary digital encoding
    - Unipolar output ranging from 0 to V_{FS}



Nomenclature:

$N = \#\ of\ bits$

$V_{FS} = full\ scale\ output$

$\Delta = min.\ step\ size \rightarrow 1LSB$

$$\Delta = \frac{V_{FS}}{2^N}$$

$$or\ N = log_2 \frac{V_{FS}}{\Delta} \rightarrow resolution$$

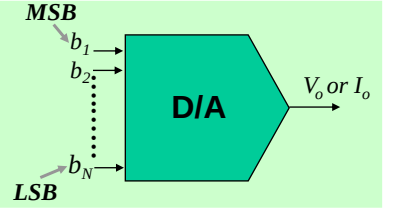
D/A Converter Transfer Characteristics

$N = \#\ of\ bits$

$V_{FS} = full\ scale\ output$

$\Delta = min.\ step\ size \rightarrow 1LSB$

$$\Delta = \frac{V_{FS}}{2^N}$$



$$V_0 = V_{FS} \sum_{i=1}^{N} \frac{bi}{2^i}$$

$$= \Delta \times \underbrace{\sum_{i=1}^{N} bi \times 2^{N-i}}_{\text{binary-weighted}} \quad , \quad bi = 0 \ or \ 1$$

$$Note: D(b_i = 1, all\ i)$$

$$\rightarrow V_o^{max} = V_{FS} - \Delta$$

$$\rightarrow V_o^{max} = V_{FS}\left(1 - \frac{1}{2^N}\right)$$

D/A Converter
Exampe: D/A with 3-bit Resolution

Example: for $N=3$ and $V_{FS}=0.8V$

input code→ *101*

Find the output value V_0

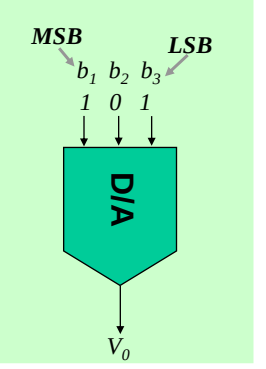
$$V_0 = \Delta\left(b_1\times 2^2 + b_2\times 2^1 + b_3\times 2^0\right)$$

$$Then: \Delta = V_{FS}/2^3 = 0.1V$$

$$\rightarrow V_0 = 0.1V\left(1\times 2^2 + 0\times 2^1 + 1\times 2^0\right) =$$

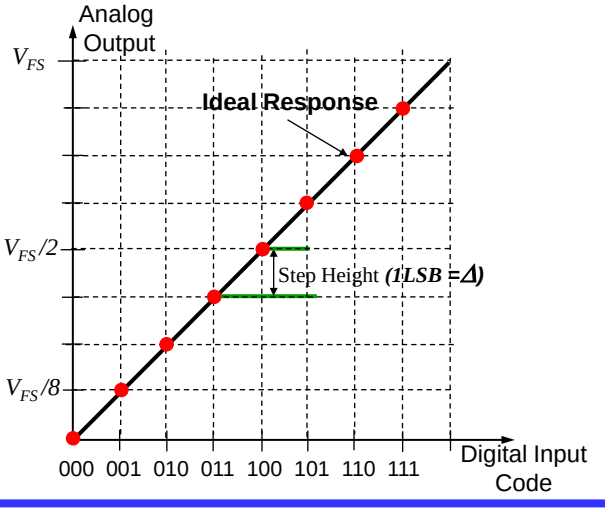
$$\rightarrow V_0 = 0.5V$$

$$Note: MSB \rightarrow V_{FS}/2 \quad \& \quad LSB \rightarrow V_{FS}/2^N$$



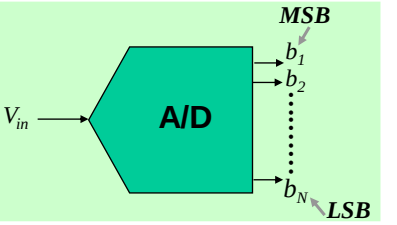
Ideal 3-Bit D/A Transfer Characteristic

- Ideal DAC introduces no error!
- One-to-one mapping from input to output



A/D Converter Transfer Characteristics

- An ideal analog-to-digital converter:
  - Accepts analog input in the form of either voltage or current
  - Produces digital output either in serial or parallel form
  - Assumption (will be revisited)
    - Unipolar input ranging from 0 to V_{FS}
    - Uniform, binary digital encoding



$N = \#\ of\ bits$

$V_{FS} = full\ scale\ output$

$\Delta = min.\ resolvable\ input \rightarrow 1LSB$

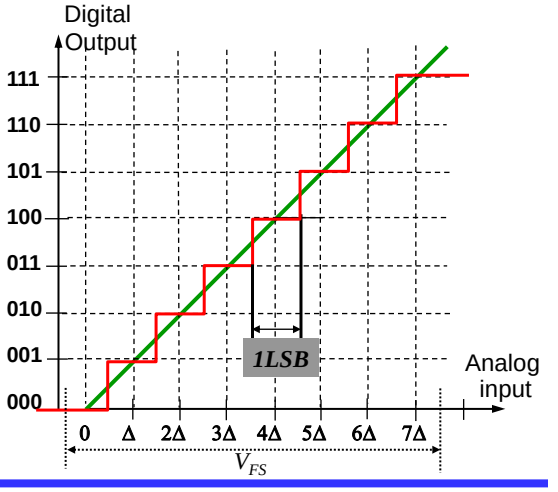
$$\Delta = \frac{V_{FS}}{2^N}$$

$$or\ N = log_2 \frac{V_{FS}}{\Delta} \rightarrow resolution$$

Ideal A/D Transfer Characteristic Example: 3Bit A/D Converter

- Ideal ADC introduces error with max peak-to-peak:
  → $(+-1/2\ \Delta)$
  $\Delta = V_{FS}/2^N$

  N= # of bits
- This error is called ``quantization error``
- For a given VFS as N increases quantization error decreases → resolution increases



Non-Linear Data Converters

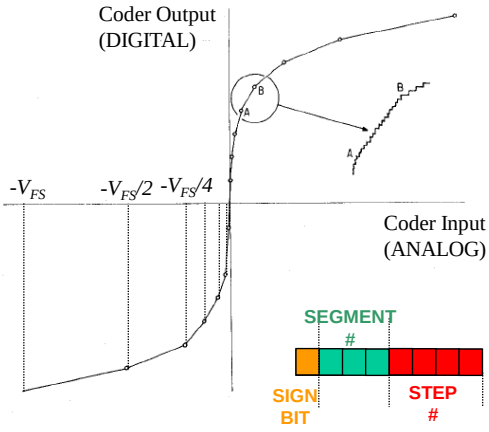
- So far data converter characterisitics studied are with uniform, binary digital encoding
- For some applications to maximize dynamic range non-linear coding is used e.g. Voice-band telephony,
  - Small signals → larger # of codes
  - Large signals → smaller # of codes

Example: Non-Linear A/D Converter For Voice-Band Telephony Applications

Non-linear ADC and DAC used in voice-band CODECs

- To maximize dynamic range without need for large # of bits
- Non-linear Coding scheme called A-law & μ-law is used
- Also called companding

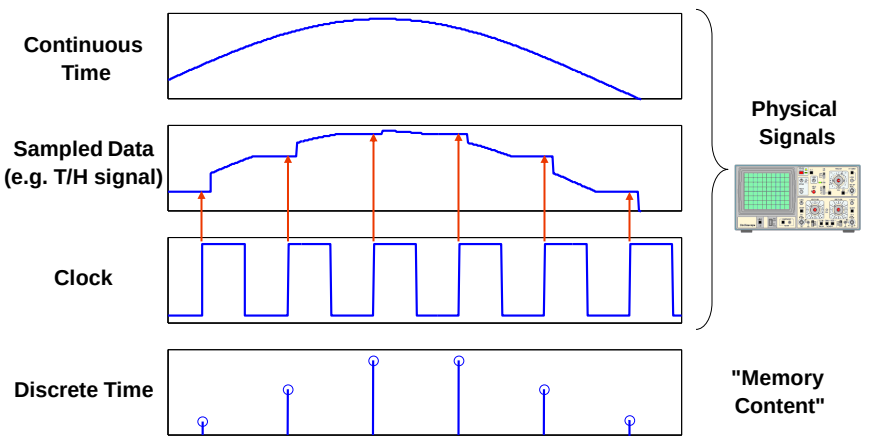
Ref: P. R. Gray, et al. "Companded pulse-code modulation voice codec using monolithic weighted capacitor arrays," *IEEE Journal of Solid-State Circuits,* vol. 10, pp. 497 - 499, December 1975.



Data Converter Performance Metrics

- Data Converters are typically characterized by static, time-domain, & frequency domain performance metrics :
  - Static
    - Offset
    - Full-scale error
    - Differential nonlinearity (DNL)
    - Integral nonlinearity (INL)
    - Monotonicity
  - Dynamic
    - Delay & settling time
    - Aperture uncertainty
    - Distortion- harmonic content
    - Signal-to-noise ratio (SNR), Signal-to-(noise+distortion) ratio (SNDR)
    - Idle channel noise
    - Dynamic range & spurious-free dynamic range (SFDR)

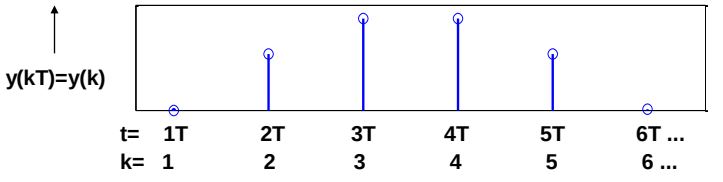
Typical Sampling Process
C.T. ⇒ S.D. ⇒ D.T.



Discrete Time Signals

- A sequence of numbers (or vector) with discrete index time instants
- Intermediate signal values not defined (not the same as equal to zero!)
- Mathematically convenient, non-physical
- We will use the term "*sampled data*" for related signals that occur in real, physical interface circuits

Uniform Sampling



- Samples spaced T seconds in time
- Sampling Period T $\Leftrightarrow$ Sampling Frequency f_s=1/T
- Problem: Multiple continuous time signals can yield exactly the same discrete time signal (aliasing)

Data Converters

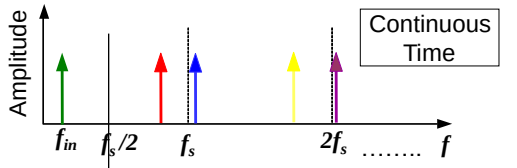
- ADC/DACs need to *sample/reconstruct* to convert from continuous-time to discrete-time signals and back
- Purely mathematical discrete-time signals are different from "sampled-data signals" that carry information in actual circuits
- Question: How do we ensure that sampling/reconstruction fully preserve information?

Aliasing

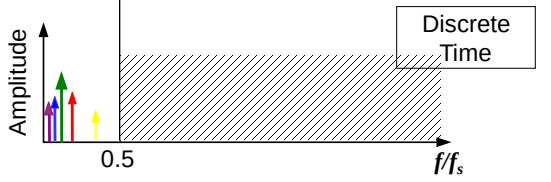
- The frequencies f_x and $nf_s \pm f_x$, n integer, are indistinguishable in the discrete time domain
- Undesired frequency interaction and translation due to sampling is called aliasing
- If aliasing occurs, no signal processing operation downstream of the sampling process can recover the original continuous time signal!

Frequency Domain Interpretation

Signal scenario before sampling

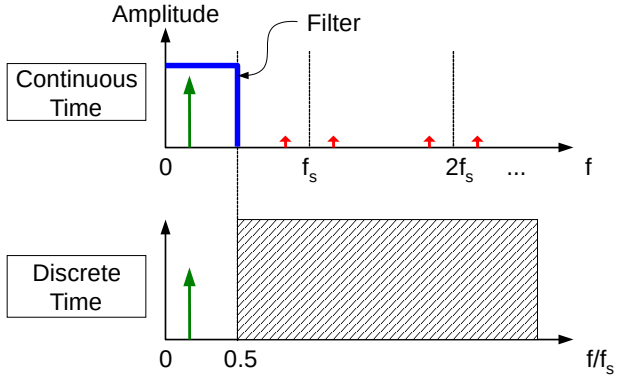


Signal scenario after sampling → DT



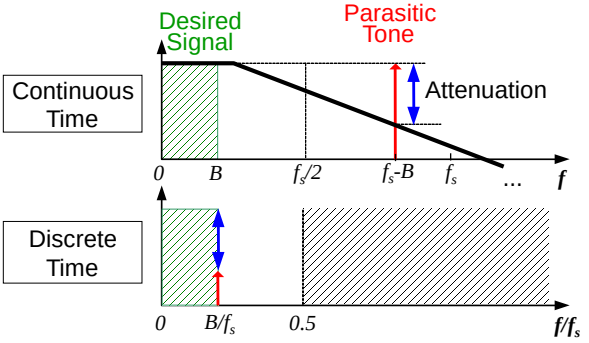
→Signals @ $nf_S \pm f_{max\_signal}$ fold back into band of interest → Aliasing

Brick Wall Anti-Aliasing Filter



Sampling at Nyquist rate ($f_s=2f_{signal}$) → required brick-wall anti-aliasing filters

Practical Anti-Aliasing Filter

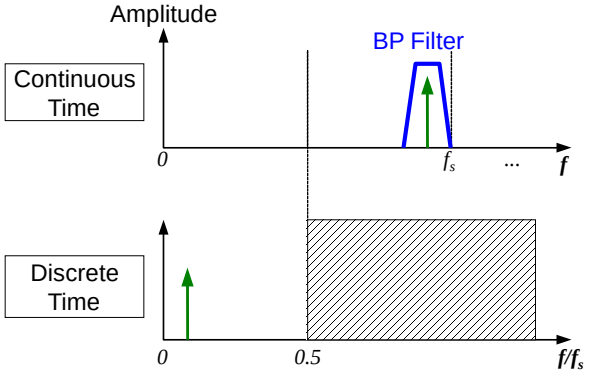


- Practical filter: Nonzero "transition band"
- In order to make this work, we need to sample faster than 2x the signal bandwidth
- "Oversampling"

Data Converter Classification

- $f_s > 2f_{max}$ Nyquist Sampling
  - "Nyquist Converters"
  - Actually always slightly oversampled (e.g. CODEC $f_{sig}^{max} = 3.4kHz$ & ADC sampling $8kHz \rightarrow f_s/f_{max} = 2.35$)
  - Requires anti-aliasing filtering prior to A-to-D conversion
- $f_s >> 2f_{max}$ Oversampling
  - "Oversampled Converters"
  - Anti-alias filtering is often trivial
  - Oversampling is also used to reduce quantization noise, see later in the course...
- $f_s < 2f_{max}$ Undersampling (sub-sampling)

Sub-Sampling



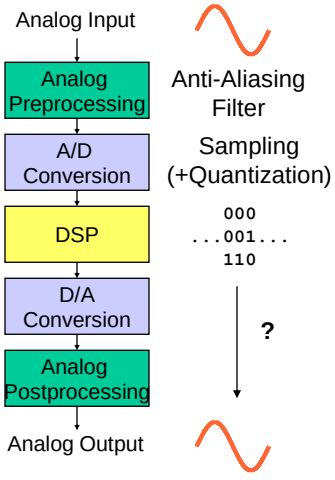
- Sub-sampling → sampling at a rate less than Nyquist rate → aliasing
- For signals centered @ an intermediate frequency → Not destructive!
- Sub-sampling can be exploited to mix a narrowband RF or IF signal down to lower frequencies

Nyquist Data Converter Topics

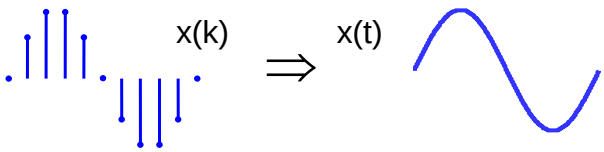
- Basic operation of data converters
  - Uniform sampling and reconstruction
  - Uniform amplitude quantization
- Characterization and testing
- Common ADC/DAC architectures
- Selected topics in converter design
  - Practical implementations
  - Compensation & calibration for analog circuit non-idealities
- Figures of merit and performance trends

Where Are We Now?

- We now know how to preserve signal information in CT→DT transition
- How do we go back from DT→ CT?



Ideal Reconstruction

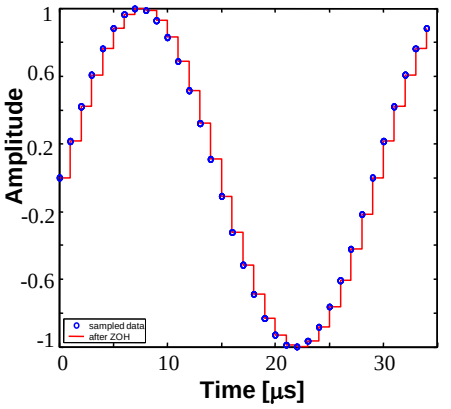


- The DSP books tell us:

$$x(t) = \sum_{k=-\infty}^{\infty} x(k) \cdot g(t - kT) \qquad g(t) = \frac{\sin(2\pi Bt)}{2\pi Bt}$$

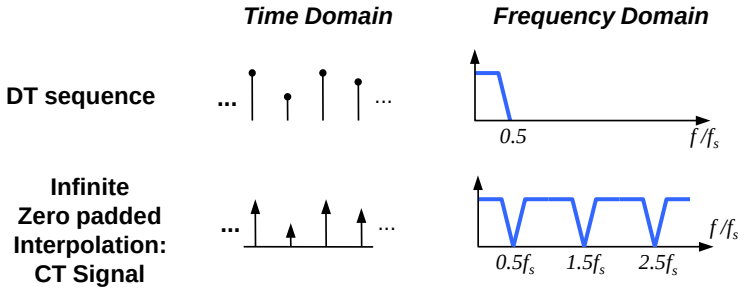
- Unfortunately not all that practical...

Zero-Order Hold Reconstruction



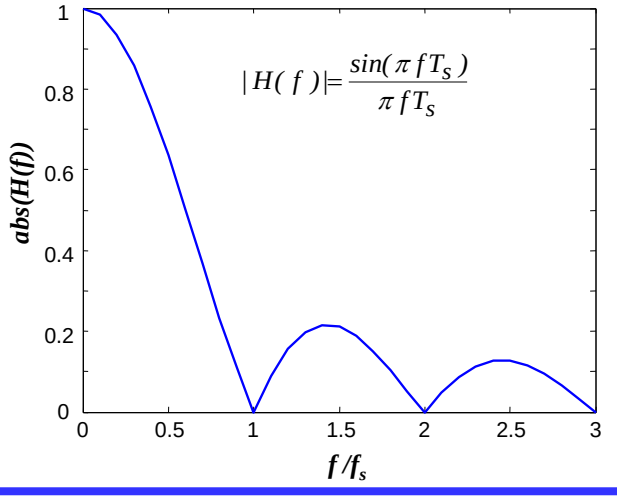
- How about just creating a staircase, i.e. hold each discrete time value until new information becomes available?
- What does this do to the frequency content of the signal?
- Let's analyze this in two steps...

DT→ CT: Infinite Zero Padding



Next step: pass the samples through a sample & hold stage (ZOH)

Hold Pulse $T_p = T_s$ Transfer Function

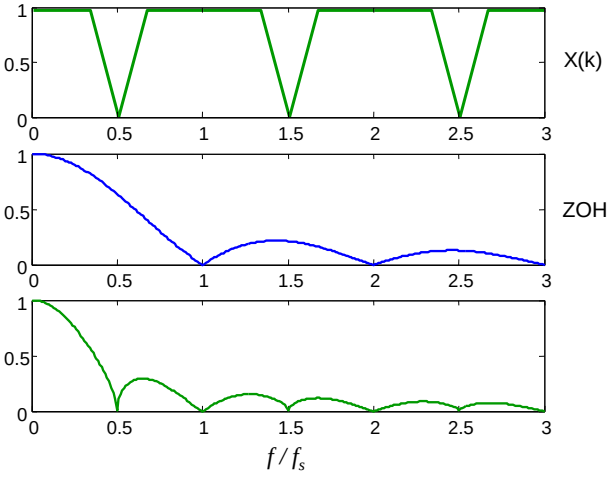


ZOH Spectral Shaping

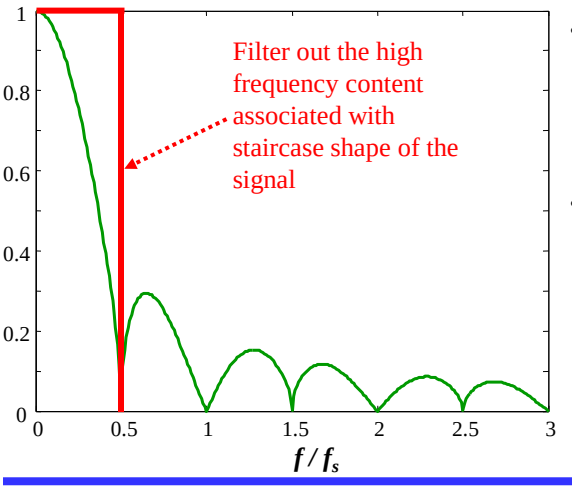
Continuous Time Pulse Train Spectrum

ZOH Transfer Function ("Sinc Shaping")

ZOH output, Spectrum of Staircase Approximation



Smoothing Filter



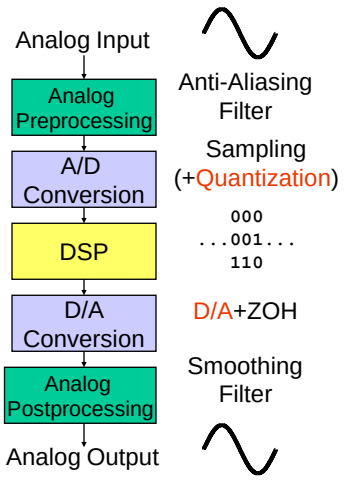
- Order of the filter required is a function of oversampling ratio
- High oversampling helps reduce filter order requirement

Summary

- Sampling theorem $f_s > 2f_{max,}$ usually dictates anti-aliasing filter
- If theorem is met, CT signal can be recovered from DT without loss of information
- ZOH and smoothing filter reconstruct CT signal from DT vector
- Oversampling helps reduce order & complexity of anti-aliasing & smoothing filters

Next Topic

- Done with "Quantization in time"
- Next: Quantization in amplitude



Data Converter Performance Metrics

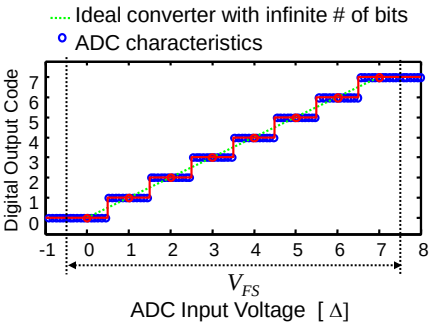
- Data Converters are typically characterized by static, time-domain, & frequency domain performance metrics :
  - Static
    - Offset
    - Full-scale error
    - Differential nonlinearity (DNL)
    - Integral nonlinearity (INL)
    - Monotonicity
  - Dynamic
    - Delay & settling time
    - Aperture uncertainty
    - Distortion- harmonic content
    - Signal-to-noise ratio (SNR), Signal-to-(noise+distortion) ratio (SNDR)
    - Idle channel noise
    - Dynamic range & spurious-free dynamic range (SFDR)

Ideal ADC ("Quantizer")

- Accepts & analog input & generates it's digital representation
- Quantization step:

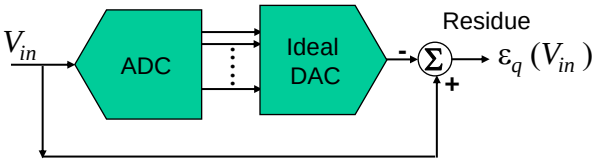
  Δ (= 1 LSB)
- Full-scale input range: $-0.5\Delta \ldots (2^N-0.5)\Delta$
- E.g. N = 3 Bits

→ V_{FS}= -0.5Δ to 7.5Δ



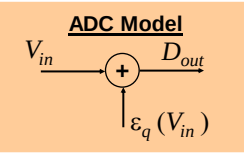
Quantization Error

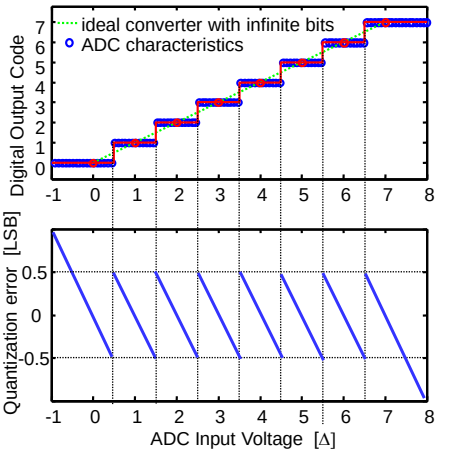
- Quantization error→ Difference between analog input and digital output of the ADC converted to analog via an ideal DAC
- Called:
  - ❑ Quantization error
  - ❑ Residue
  - ❑ Quantization noise



Quantization Error

- For an ideal ADC:
  - Quantization error is bounded by –Δ/2 … +Δ/2 for inputs within full-scale range





ADC Dynamic Range

- Assuming quantization noise is much larger compared to circuit generated noise:

$$D.R._{Maximum} = 10\log\frac{Full\ Scale\ \ Signal\ Power}{Quantization\ Noise\ Power}$$

- Crude assumption: Same peak/rms ratio for signal and quantization noise!

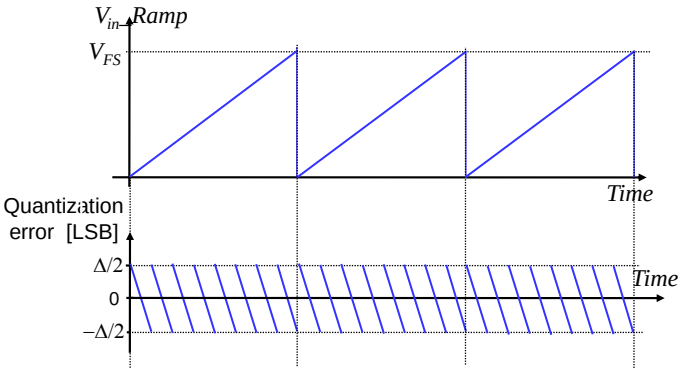
$$D.R._{Maximum} = 20\log\frac{Peak\ Full\ Scale}{Peak\ Quantization\ Noise}$$

$$= 20\log\frac{V_{FS}}{\Delta} = 20\log 2^N = 6.02\times N\ \ [dB]$$

Question: What is the quantization noise power?

Quantization Error

Assume V_{in} is a slow ramp signal with amplitude equal to ADC full-scale



Note: Ideal ADC quantization error waveform→ periodic and also ramp

Quantization Error Derivation

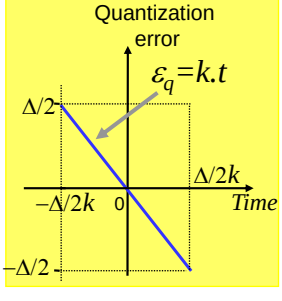
Need to find the *rms* value for quantization error waveform:

$$\overline{\varepsilon_{eq}^{\,2}} = \frac{1}{T}\int_{-T/2}^{+T/2}(k\times t)^2\,dt = \frac{k}{\Delta}\int_{-\Delta/2k}^{+\Delta/2k}(k\times t)^2\,dt$$

$$= \frac{k\times k^2}{\Delta}\int_{-\Delta/2k}^{+\Delta/2k} t^2\,dt$$

$$\rightarrow \overline{\varepsilon_{eq}^{\,2}} = \frac{\Delta^2}{12}$$ → Independent of k

$$\rightarrow \boxed{\overline{\varepsilon_{eq}} = \frac{\Delta}{\sqrt{12}}}$$



In general above equation applies if:

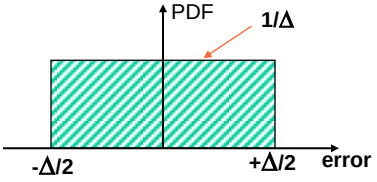
- Input signal much larger than 1LSB
- Input signal busy
- No signal clipping

Quantization Error PDF

- Probability density function (PDF) Uniformly distributed from –Δ/2 … +Δ/2 provided that:
  - Busy input
  - Amplitude is many LSBs
  - No overload
- Not Gaussian!

- Zero mean
- Variance

$$\overline{e^2} = \int_{-\Delta/2}^{+\Delta/2} \frac{e^2}{\Delta} de = \frac{\Delta^2}{12}$$



Ref: W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., vol. 27, pp. 446-72, July 1988.

B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," IRE Trans. Circuit Theory, vol. CT-3, pp. 266-76, 1956.

Signal-to-Quantization Noise Ratio

- Under certain conditions the quantization error can be viewed as "random", and is often referred to as "noise"
- In this case, we can define a maximum "signal-to-quantization noise ratio", SQNR, for sinusoidal inputs:

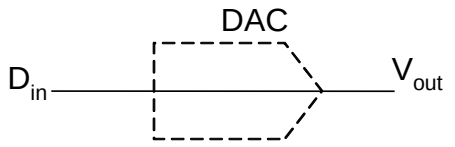
$$SQNR = \frac{\frac{1}{2}\left(\frac{2^N \Delta}{2}\right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2N}$$

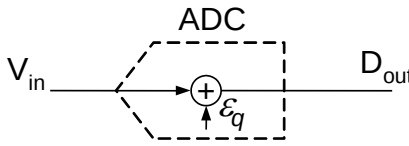
$$= 6.02N + 1.76 \ dB \qquad Accurate\ for\ N>3$$

e.g. N	SQNR
8	50 dB
12	74 dB
16	98 dB
20	122 dB

- Real converters do not quite achieve this performance due to other sources of error:
  - Electronic noise
  - Deviations from the ideal quantization levels

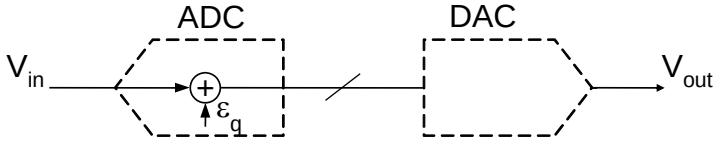
Static Ideal Macro Models

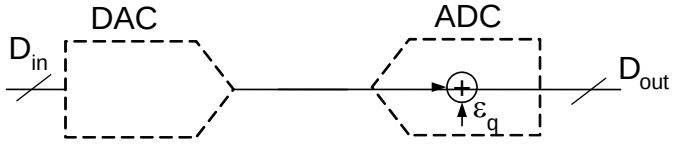




→ +-0.5LSB ambiguity

Cascade of Data Converters



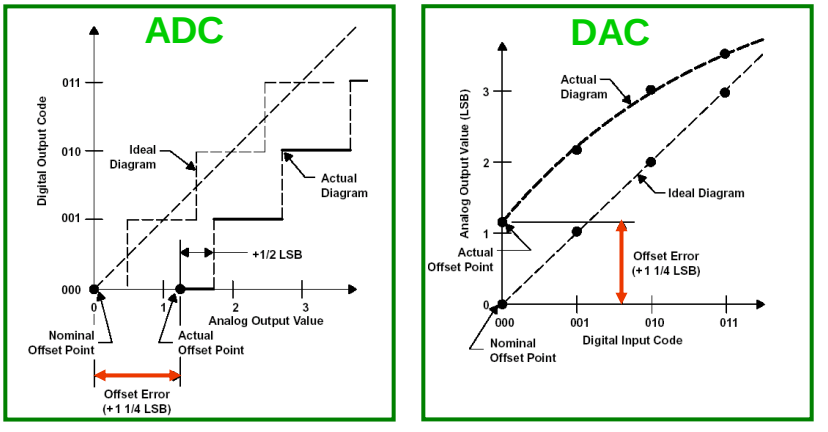


Static Converter Errors

Deviation of converter characteristics from ideal:

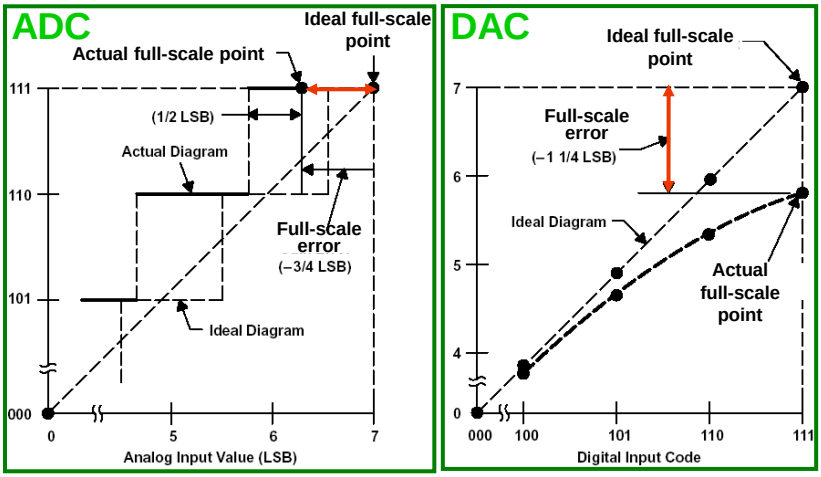
- Offset
- Full-scale error
- Differential nonlinearity → DNL
- Integral nonlinearity → INL

Offset Error



Ref: "Understanding Data Converters," Texas Instruments Application Report SLAA013, Mixed-Signal Products, 1995.

Full-Scale Error



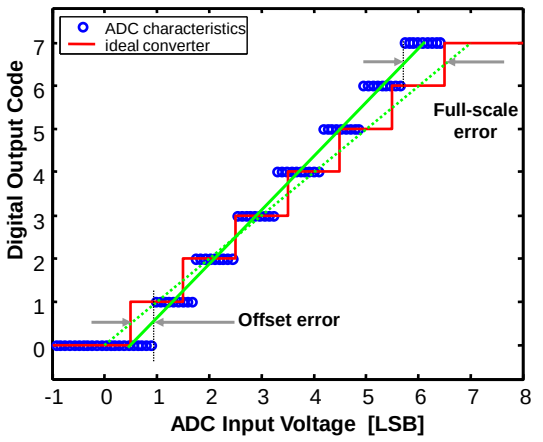
Offset and Full-Scale Errors

- Alternative specification in % Full-Scale = 100% * (# of LSB value)/ 2^N
- Gain error can be extracted from offset & full-scale error
- Non-trivial to build a converter with extremely good full-scale/offset specs
- Typically full-scale/offset error is most easily compensated by the digital pre/post-processor
- More critical: Linearity measures→DNL, INL

Offset and Full-Scale Error

Note:

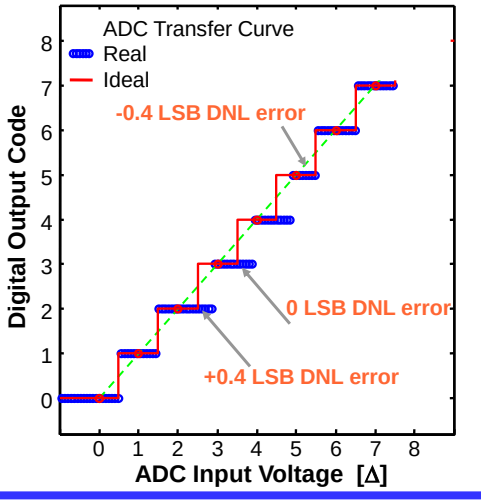
→ For further measurements (DNL, INL) connecting the endpoints & deriving ideal codes based on the non-ideal endpoints eliminates offset and full-scale error



ADC Differential Nonlinearity

DNL = deviation of code width from Δ (1LSB)

1. Endpoints connected
2. Ideal characteristics derived eliminating offset & full-scale error
3. DNL measured → code width deviation from 1LSB



ADC Differential Nonlinearity

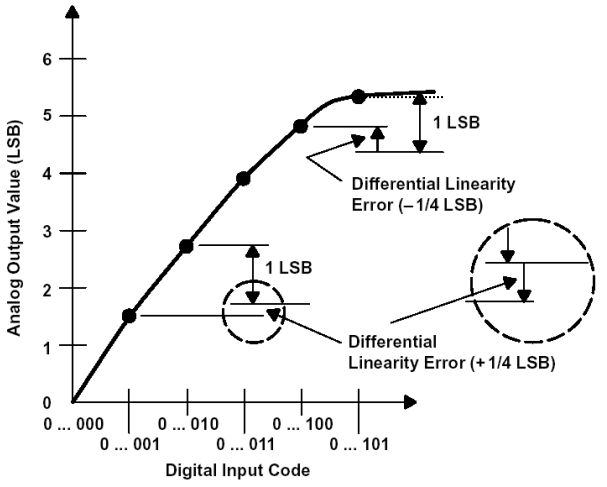
- Ideal ADC transitions point equally spaced by 1LSB
- For DNL measurement, offset and full-scale error is eliminated
- DNL [k] (a vector) measures the deviation of each code from its ideal width
- Typically, the vector for the entire code is reported
- If only one DNL # is presented that would be the worst case

ADC DNL

- DNL=-1 implies missing code
- For an ADC DNL < -1 not possible → undefined
- Can show:

$$\sum_{all\ i} DNL[i] = 0$$

DAC Differential Nonlinearity



DAC Differential Nonlinearity

- To find DNL for DAC
  - Draw end-point line from 1st point to last
  - Find ideal LSB size for the end-point corrected curve
  - Find segment sizes:

$$segment\,[m] = V[m] - V[m-1]$$

$$DNL[m] = \frac{segment[m] - V[LSB]}{V[LSB]}$$

- Unlike ADC DNL, for a DAC DNL can be <-1LSB

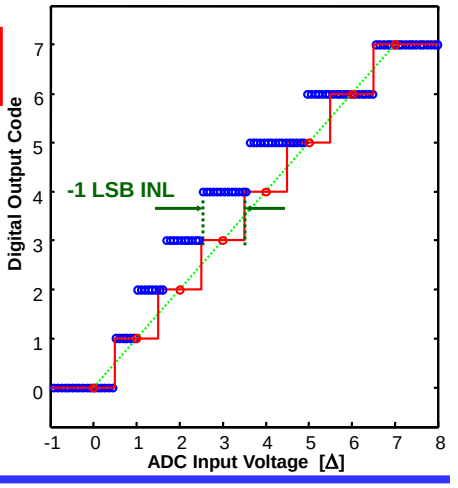
Impact of DNL on Performance

- Same as a somewhat larger quantization error, consequently degrades SQNR
- How much – later in the course...
- The term "DNL noise", usually means "additional quantization noise due to DNL"

ADC Integral Nonlinearity End-Point

INL = deviation of code transition from its ideal location

1. Endpoints connected
2. Ideal characteristics derived eliminating offset & full-scale error (same as for DNL)
3. INL→ deviation of code transition from ideal is measured



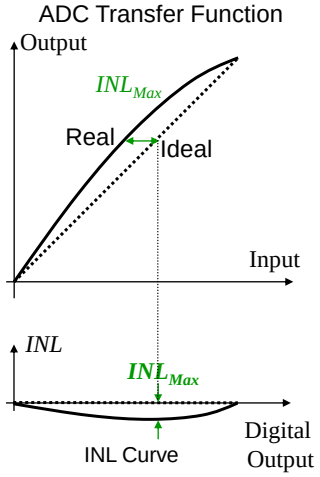
ADC Integral Nonlinearity

INL = deviation of code transition from its ideal location

INL is also a vector INL[k]
If one INL # reported
→Worst case INL

Most common→ End-point:
Straight line through the endpoints is usually used as reference,
i.e. offset and full scale errors are eliminated in INL calculation

Ideal converter steps found for the endpoint line, then INL is measured

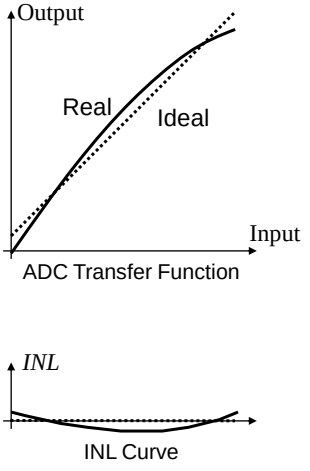


ADC Integral Nonlinearity Best-Fit

INL = deviation of code transition from its ideal location

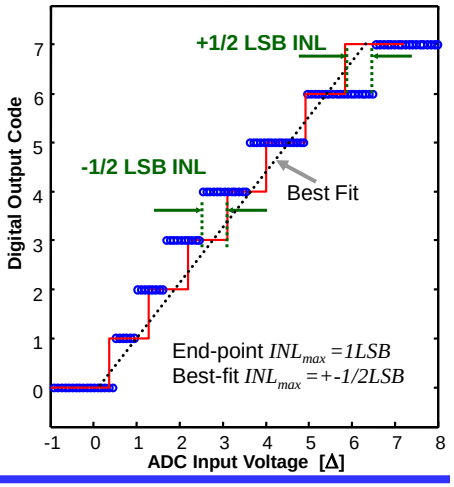
Best-Fit
- A best-fit line (in the least-mean squared sense) fitted to measured data
- Ideal converter steps found then INL measured

Note: Typically INL #s smaller for best-fit compared to end-point



ADC Integral Nonlinearity
Best Fit versus End-Point

- Best-Fit
  - A best-fit line (in the least-mean squared sense)
  - Ideal converter steps is found then INL is measured



ADC Integral Nonlinearity

Can derive INL by:

1-

- Construct uniform staircase between 1st and last transition
- INL for each code:

$$INL[m] = \frac{T[m] - T[ideal]}{W[ideal]}$$

2-

- Can show

$$INL[m] = \sum_{i=1}^{m-1} DNL[i]$$

→ INL is found by computing the cumulative sum of DNL

ADC Differential & Integral Nonlinearity Example

$$INL[m] = \sum_{i=1}^{m-1} DNL[i]$$

Notice: For end-point corrected measurement

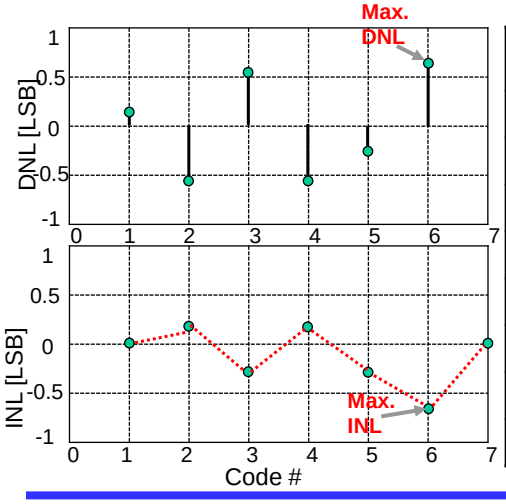
$INL[0] \rightarrow undefined$

$INL[1]=0$

$INL[2^N-1]=0$

Code #	DNL [LSB]	INL [LSB]
0	-	-
1	*0.18*	*0*
2	*-0.55*	*0.18*
3	*0.55*	*-0.37*
4	*-0.55*	*0.18*
5	*-0.27*	*-0.37*
6	*0.64*	*-0.64*
7	-	*0*

ADC Differential & Integral Nonlinearity Example



Code #	DNL [LSB]	INL [LSB]
0	-	-
1	*0.18*	*0*
2	*-0.55*	*0.18*
3	*0.55*	*-0.37*
4	*-0.55*	*0.18*
5	*-0.27*	*-0.37*
6	*0.64*	*-0.64*
7	-	*0*

DAC Integral Nonlinearity

Can derive INL by:

- Connect end points
- Find ideal output values
- INL for each code:

$$INL[m] = \frac{V[m] - V[ideal]}{V[LSB]}$$

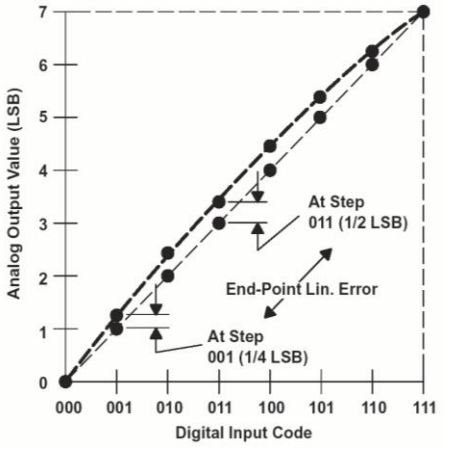
2-

- Can show

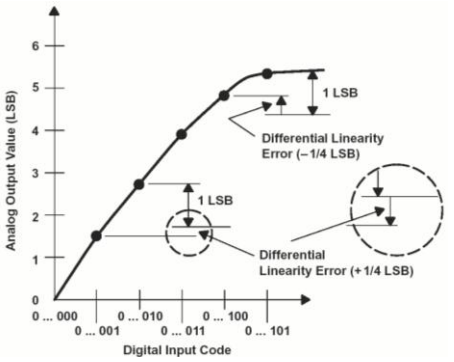
$$INL[m] = \sum_{i=1}^{m-1} DNL[i]$$

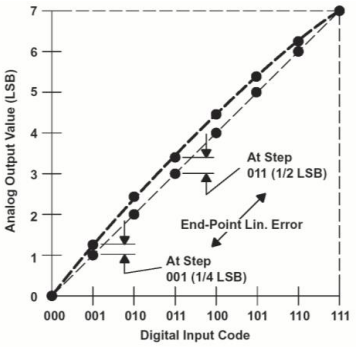
→ INL is found by computing the cumulative sum of DNL

DAC Integral Nonlinearity



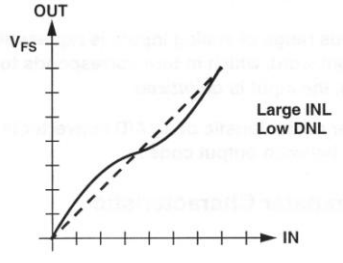
DAC DNL and INL



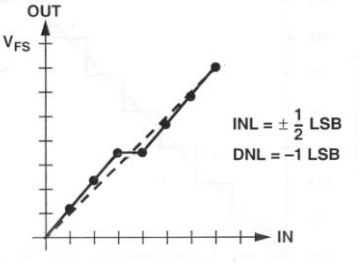


* Ref: "Understanding Data Converters," Texas Instruments Application Report SLAA013, Mixed-Signal Products, 1995.

Example: INL & DNL



Large INL & Small DNL
Smooth variations in transfer curve → Small DNL



Large DNL & Small INL
Abrupt variations in transfer curve → Large DNL

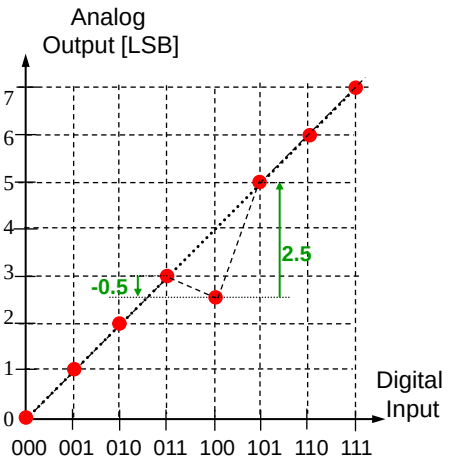
Non-Monotonic DAC

$$DNL[m] = \frac{segment[m] - V[LSB]}{V[LSB]}$$

$$DNL[4] = \frac{segment[4] - V[LSB]}{V[LSB]} = \frac{-0.5 - 1}{1} = -1.5[LSB]$$

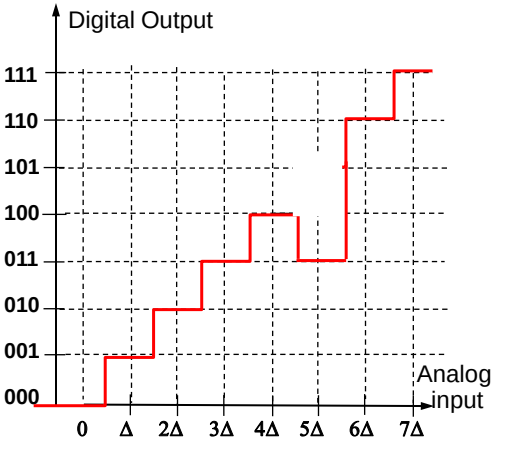
$$DNL[5] = \frac{2.5 - 1}{1} = 1.5[LSB]$$

- ***DNL< -1LSB* for a DAC**
  → Non-monotonicity
- When can non-monotonicity cause major problems?



Non-Monotonic ADC

- Code 011 associated with two transition levels !
- For non-monotonic ADC
  →DNL not defined @ non-monotonic steps

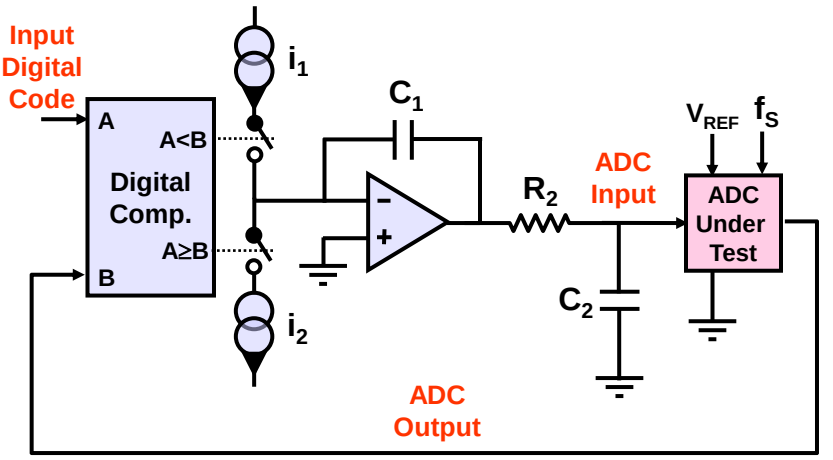


How to measure DNL/INL?

- DAC:
  - Simply apply digital codes and use a good voltmeter to measure corresponding analog output
- ADC
  - Not as simple as DAC→ need to find "decision levels", i.e. input voltages at all code boundaries
    - One way: Adjust voltage source to find exact code trip points "code boundary servo"
    - More versatile: Histogram testing

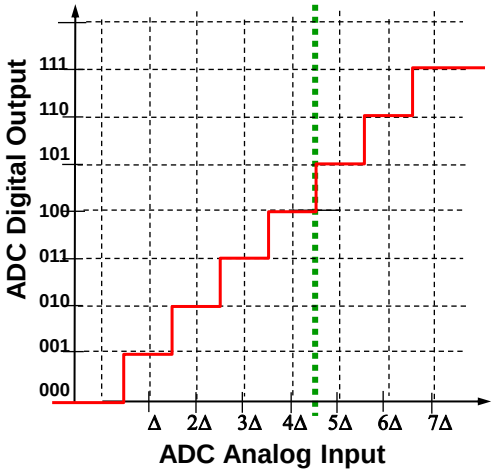
      →Apply a signal with known amplitude distribution and analyze digital code distribution at ADC output

Code Boundary Servo

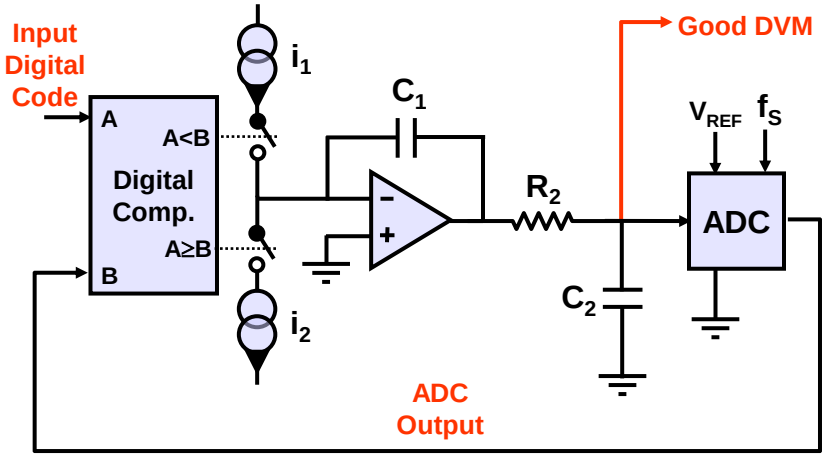


Code Boundary Servo

- i1 and i2 are small, and C1 is large ($\Delta V = it/C1$), so the ADC analog input moves a small fraction of an LSB (e.g. 0.1LSB) each sampling period
- For a code input of 101, the ADC analog input settles to the code boundary shown



Code Boundary Servo

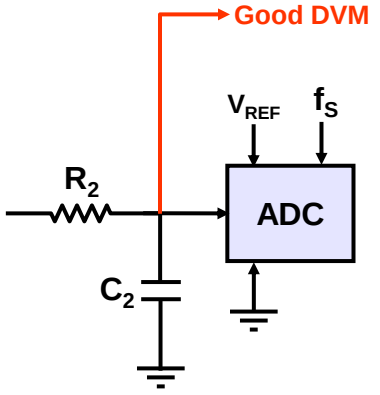


Code Boundary Servo

- A very good digital voltmeter (DVM) measures the analog input voltage corresponding to the desired code boundary
- DVMs have some interesting properties
  - They can have very high resolutions (8½ decimal digit meters are inexpensive)
  - To achieve stable readings, DVMs average voltage measurements over multiple 60Hz ac line cycles to filter out pickup in the measurement loop

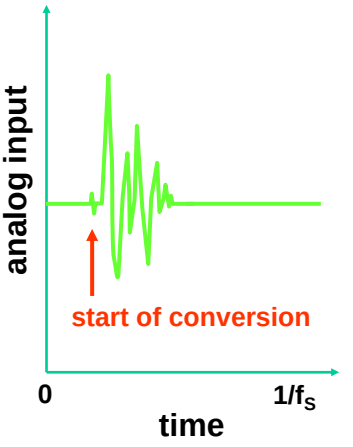
Code Boundary Servo

- ADCs of all kinds are notorious for kicking back high-frequency, signal-dependent glitches to their analog inputs
- A magnified view of an analog input glitch follows …



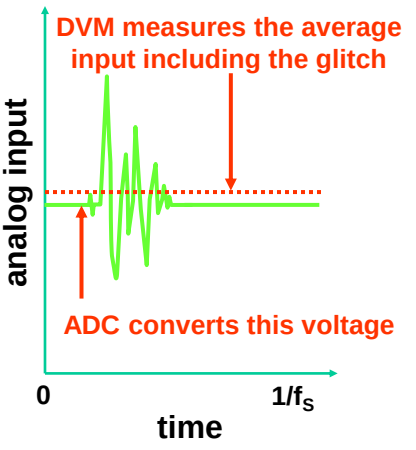
Code Boundary Servo

- Just before the input is sampled and conversion starts, the analog input is pretty quiet
- As the converter begins to quantize the signal, it kicks back charge



Code Boundary Servo

- The difference between what the ADC measures and what the DVM measures is not ADC INL, it's error in the INL measurement
- How do we control this error?



Code Boundary Servo

- A large C_2 reduces the effect of kick-back
- At the expense of longer measurement time