EE 435

Lecture 33

Absolute and Relative Accuracy
DAC Design

• The String DAC
DFT Simulation from Matlab

Rec Win  N=65536 Np =1  Nsamp = 284.93913 nres = 12  fCL/fsig = 10  fDFT/fsig = 2849.3913

![DFT Simulation Graph](image)
Summary of time and amplitude quantization assessment

Time and amplitude quantization do not introduce harmonic distortion

- Time and amplitude quantization do increase the noise floor
Quantization Noise

- DACs and ADCs generally quantize both amplitude and time.
- If converting a continuous-time signal (ADC) or generating a desired continuous-time signal (DAC) these quantizations cause a difference in time and amplitude from the desired signal.
- First a few comments about Noise.
Noise

We will define “Noise” to be the difference between the actual output and the desired output of a system.

Types of noise:

• Random noise due to movement of electrons in electronic circuits
• Interfering signals generated by other systems
• Interfering signals generated by a circuit or system itself
• Error signals associated with imperfect signal processing algorithms or circuits

Quantization noise is a significant component of this noise in ADCs and DACs and is present even if the ADC or DAC is ideal.
Quantization Noise in ADC
(same concepts apply to DACs)

Consider an Ideal ADC with first transition point at $0.5X_{\text{LSB}}$

If the input is a low frequency sawtooth waveform of period $T$ that goes from 0 to $X_{\text{REF}}$, the error signal in the time domain will be:

$$
\epsilon_Q = \begin{cases} 
0 & \text{for } t = nT_1 \\
-0.5X_{\text{LSB}} & \text{for } t = (n+1/2)T_1 \\
0.5X_{\text{LSB}} & \text{for } t = (n+3/2)T_1 \\
\text{where } T_1 = T/2^n 
\end{cases}
$$

This time-domain waveform is termed the Quantization Noise for the ADC with a sawtooth (or triangular) input.
Quantization Noise in ADC

\[ E_{\text{RMS}} = \sqrt{\frac{1}{T_1} \int_{-T_1/2}^{T_1/2} \varepsilon_Q^2(t) \, dt} \]

\[ E_{\text{RMS}} = \sqrt{\frac{1}{T_1} \int_{-T_1/2}^{T_1/2} \left(- \frac{x_{\text{LSB}}}{T_1}\right)^2 t^2 \, dt} \]

\[ E_{\text{RMS}} = x_{\text{LSB}} \sqrt{\frac{1}{T_1^3} \int_{-T_1/2}^{T_1/2} t^3 \, dt} \]

\[ E_{\text{RMS}} = \frac{x_{\text{LSB}}}{\sqrt{12}} \]

\[ \varepsilon_Q(t) = -\left(\frac{x_{\text{LSB}}}{T_1}\right)t \]
Quantization Noise in ADC

\[ E_{\text{RMS}} = \frac{X_{\text{LSB}}}{\sqrt{12}} \]

The signal to quantization noise ratio (SNR) can now be determined. Since the input signal is a sawtooth waveform of period \( T \) and amplitude \( X_{\text{REF}} \), it follows by the same analysis that it has an RMS value of

\[ X_{\text{RMS}} = \frac{X_{\text{REF}}}{\sqrt{12}} \]

Thus the SNR is given by

\[ \text{SNR} = \frac{X_{\text{RMS}}}{E_{\text{RMS}}} = \frac{X_{\text{RMS}}}{X_{\text{LSB}}} = 2^n \]

or, in dB,

\[ \text{SNR}_{\text{dB}} = 20(n \cdot \log2) = 6.02n \]

Note: dB subscript often neglected when not concerned about confusion
Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to $X_{REF}$ centered at $X_{REF}/2$?

$SNR = 20 \cdot \log_2 = 6.02 n$

SNR $= 20(n \cdot \log 2) = 6.02n$
Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{\text{REF}}$ centered at $\mathcal{X}_{\text{REF}}/2$?

Time and Amplitude Quantized Waveform
Quantization Noise in ADC

Recall:

If the random variable $f$ is uniformly distributed in the interval $[A,B]$, $f: U[A,B]$ then the mean and standard deviation of $f$ are given by

$$
\mu_f = \frac{A + B}{2} \quad \text{and} \quad \sigma_f = \frac{B - A}{\sqrt{12}}
$$

Theorem: If $n(t)$ is a random process, then for large $T$,

$$
V_{\text{RMS}} = \sqrt{\frac{1}{T} \int_{t_1}^{t_1+T} n^2(t) \, dt} = \sqrt{\sigma_n^2 + \mu_n^2}
$$
Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to \( X_{\text{REF}} \) centered at \( X_{\text{REF}}/2 \)?

\[
V_{\text{RMS}} = \frac{X_{\text{LSB}}}{\sqrt{12}}
\]

But \( V_{\text{INRMS}} = \left( \frac{X_{\text{REF}}}{2} \right) \frac{1}{\sqrt{2}} \)

Thus obtain

\[
\text{SNR} = \frac{X_{\text{REF}}}{2\sqrt{2}} = 2^n \sqrt{\frac{3}{2}}
\]

Finally, in dB,

\[
\text{SNR}_{\text{dB}} = 20\log \left( 2^n \sqrt{\frac{3}{2}} \right) = 6.02n + 1.76
\]
ENOB based upon Quantization Noise

SNR = 6.02 n + 1.76

Solving for n, obtain

\[
\text{ENOB} = \frac{\text{SNR}_{\text{dB}} - 1.76}{6.02}
\]

Note: could have used the SNR_{dB} for a triangle input and would have obtained the expression

\[
\text{ENOB} = \frac{\text{SNR}_{\text{dB}}}{6.02}
\]

But the earlier expression is more widely used when specifying the ENOB based upon the noise level present in a data converter.
ENOB based upon Quantization Noise

For very low resolution levels, the assumption that the quantization noise is uncorrelated with the signal is not valid and the ENOB expression will cause a modest error

\[ SNR_{\text{corr}} \approx \left( 2^n - 2 + \frac{4}{\pi} \right) \sqrt{\frac{3}{2}} \]

from van de Plassche (p13)

<table>
<thead>
<tr>
<th>Res (n)</th>
<th>SNR(_{\text{corr}})</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.86</td>
<td>7.78</td>
</tr>
<tr>
<td>2</td>
<td>12.06</td>
<td>13.8</td>
</tr>
<tr>
<td>3</td>
<td>19.0</td>
<td>19.82</td>
</tr>
<tr>
<td>4</td>
<td>25.44</td>
<td>25.84</td>
</tr>
<tr>
<td>5</td>
<td>31.66</td>
<td>31.86</td>
</tr>
<tr>
<td>6</td>
<td>37.79</td>
<td>37.88</td>
</tr>
<tr>
<td>8</td>
<td>49.90</td>
<td>49.92</td>
</tr>
<tr>
<td>10</td>
<td>61.95</td>
<td>61.96</td>
</tr>
</tbody>
</table>

SNR = 6.02 n + 1.76

Table values in dB

Almost no difference for n ≥ 3
Effects of quantization noise can be very significant, even at high resolution, when signals are not of maximum magnitude.

Quantization noise remains constant but signal level is reduced.

The desire to use a data converter at a small fraction of full range is one of the major reasons high resolution is required.
Quantization Noise

Effects of quantization noise can be very significant, even at high resolution, when signals are not of maximum magnitude.
Quantization Noise

Example: If a 14-bit audio output is derived from a DAC designed for providing an output of 100W but the normal listening level is at 50mW, what is the SNR due to quantization noise at maximum output and at the normal listening level? What is the ENOB of the audio system when operating at 50mW?

At 100W output, SNR = 6.02n + 1.76 = 90.6dB

\[
\frac{V^2}{R_L} = 100W \\
\frac{V_1^2}{R_L} = 50mW
\]

\[
V_1 = \frac{V}{44.7}
\]

\[
20\log_{10}V_1 = 20\log_{10}V - 20\log_{10}44.7 = 20\log_{10}V \quad -33dB
\]

At 50mW output, SNR reduced by 33dB to 57.6dB

\[
\text{ENOB} = \frac{\text{SNR}_{\text{dB}} - 1.76}{6.02} = \frac{57.6 - 1.76}{6.02} = -9.3
\]

Note the dramatic reduction in the effective resolution of the DAC when operated at only a small fraction of full-scale.
ENOB Summary

Resolution:

\[ \text{ENOB} = \log_{10} \frac{N_{\text{ACT}}}{2} = \log_2 N_{\text{ACT}} \]

INL:

\[ \text{ENOB} = n_R - \log_2 (\nu) - 1 \quad \text{n}_R \text{ specified res, } \nu \text{ INL in LSB} \]

\[ \text{ENOB} = -\log_2 \left( \text{INL}_{\text{REF}} \right) - 1 \quad \text{INL}_{\text{REF}} \text{ INL rel to } X_{\text{REF}} \]

DNL:

HW problem

Quantization noise:

\[ \text{ENOB} = \frac{\text{SNR}_{\text{dB}}}{6.02} \quad \text{rel to triangle/sawtooth} \]

\[ \text{ENOB} = \frac{\text{SNR}_{\text{dB}} - 1.76}{6.02} \quad \text{rel to sinusoid} \]
Performance Characterization of Data Converters

• Static characteristics
  – Resolution
  – Least Significant Bit (LSB)
  – Offset and Gain Errors
  Absolute Accuracy
  Relative Accuracy
  – Integral Nonlinearity (INL)
  – Differential Nonlinearity (DNL)
  – Monotonicity (DAC)
  – Missing Codes (ADC)
  – Quantization Noise
  – Low-f Spurious Free Dynamic Range (SFDR)
  – Low-f Total Harmonic Distortion (THD)
  – Effective Number of Bits (ENOB)
  – Power Dissipation
Absolute Accuracy

Absolute Accuracy is the difference between the actual output and the ideal or desired output of a data converter

The ideal or desired output is in reference to an absolute standard (often maintained by the National Bureau of Standards) and could be volts, amps, time, weight, distance, or one of a large number of other physical quantities

Absolute accuracy provides no tolerance to offset errors, gain errors, nonlinearity errors, quantization errors, or noise

In many applications, absolute accuracy is not of a major concern

but … scales, meters, etc. may be more concerned about Absolute accuracy than any other parameter
Relative Accuracy

In the context of data converters, pseudo-static Relative Accuracy is the difference between the actual output and an appropriate fit-line to overall output of the data converter.

INL is often used as a measure of the relative accuracy.

In many, if not most, applications, relative accuracy is of much more concern than absolute accuracy.

Some architectures with good relative accuracy will have very small deviations in the outputs for closely-spaced inputs whereas others may have relatively large deviations in outputs for closely-spaced inputs.

DNL provides some measure of how outputs for closely-spaced inputs compare.
DAC Architectures (Nyquist Rate)

Types

- **Voltage Scaling**
  - Resistor String DACs (string DACs)
  - Interpolating

- **Current Steering**
  - Binarily Weighted Resistors
  - R-2R Ladders
  - Current Source Steering
    - Thermometer Coded
    - Binary Weighted
    - Segmented

- **Charge Redistribution**
  - Switched Capacitor

- **Serial**
  - Algorithmic
  - Cyclic or Re-circulating
  - Pipelined

- **Integrating**

- **Resistor Switching**

- **MDACs (multiplying DACs)**
DAC Architectures

Structures

• Hybrid or Segmented
• Mode of Operation
  – Current Mode
  – Voltage Mode
  – Charge Mode
• Self-Calibrating
  – Analog Calibration
    • Foreground
    • Background
  – Digital Calibration
    • Foreground
    • Background
  – Dynamic Element Matching
• Laser of Link Trimmed
• Thermometer Coded or Binary
• Radix 2 or non-radix 2
• Inherently Monotone
DAC Architectures

- Type of Classification may not be unique nor mutually exclusive
- Structure is not mutually exclusive
- All approaches listed are used (and probably some others as well)
- Some are much more popular than others
  - Popular Architectures
    - Resistor String (interpolating)
    - Current Source Steering (with segmentation)
- Many new architectures are possible and some may be much better than the best currently available
- All have perfect performance if parasitic and matching performance are ignored!
- Major challenge is in determining appropriate architecture and managing the parasitics
Nonideal Effects of Concern

- Matching
- Parasitic Capacitances (including Charge injection)
- Loading
- Nonlinearities
- Interconnect resistors
- Noise
- Slow and plagued by jitter
- Temperature Effects
- Aging
- Package stress
Observations

• Yield Loss is the major penalty for not appropriately managing parasitics and matching and this loss can be ruthless.
• The ultimate performance limit of essentially all DACs is the yield loss associated with parasitics and matching.
• Many designers do not have or use good statistical models that accurately predict data converter performance.
• If you work of a company that does not have good statistical device models:
  – Convince model groups of the importance of developing these models.
  – (or) develop appropriate test structures to characterize your process.
• Existing nonlinear device models may not sufficiently accurately predict device nonlinearities for high-end data converter applications.
Observations

• Experienced Designers/Companies often produce superior data converter products
• Essentially all companies have access to the same literature, regularly reverse engineer successful competitors products and key benefits in successful competitors products are generally not locked up in patents
• High-end designs (speed and resolution) may get attention in the peer community but practical moderate performance converters usually make the cash flow
• Area (from a silicon cost viewpoint) is usually not the driving factor in high-end designs where attractive price/mfg cost ratios
Data Converter Design Strategies

• There are many different DAC and ADC architectures that have been proposed and that are in widespread use today

• Almost all work perfectly if all components are ideal

• Most data converter design work involves identifying the contributors to nonideal performance and finding work-arounds to these problems

• Some architectures are more difficult to find work-arounds than others

• All contributors to nonidealities that are problematic at a given resolution of speed level must be identified and mitigated

• The effects of not identifying nonidealities generally fall into one of two categories
  – Matching-critical nonidealities (degrade yield)
  – Component nonlinearities (degrade performance even if desired matching is present)
Identifying Problems/Challenges and Clever/Viable Solutions

- Many problems occur repeatedly so should recognize what they are
- Identify clever solutions to basic problems – they often are useful in many applications
- Don’t make the same mistake twice!

The problem:

The perceived solution:

The practical or clever solution:

The List Keeper!
R-String DAC

Basic R-String DAC
R-String DAC

Basic R-String DAC including Logic to Control Switches
R-String DAC

If all components are ideal, performance of the R-string DAC is that of an ideal DAC!

Key Properties of R-String DAC
• One of the simplest DAC architectures
• R-string DAC is inherently monotone

Possible Limitations or Challenges
R-String DAC

If all components are ideal, performance of the R-string DAC is that of an ideal DAC!

Key Properties of R-String DAC

- One of the simplest DAC architectures
- R-string DAC is inherently monotone

Possible Limitations or Challenges

- Binary to Thermometer Decoder (BTTD) Gets Large for \( n \) large
- Logic delays in BTTD may degrade performance
- Matching of the resistors may not be perfect
  - Local random variations
  - Gradient effects
- How can switches be made?
R-String DAC

Typical strategy for implementing the switch

- Switch is an analog MUX
- Very simple structure
- Switch array combined with the BTDD forms a $2^n:1$ analog MUX
R-String DAC

R-String DAC with MOS switches

Possible Limitations:
R-String DAC

R-String DAC with MOS switches

**Possible Limitations:**

Switch impedance is not 0

Switch may not even turn on at all if $V_{\text{REF}}$ is large

Switch impedance is input-code dependent

Time constants are input-code dependent

Transition times are previous-code dependent

$C_L$ has $2^n$ diffusion capacitances so can get very large

Mismatch of resistors
  local random variation
  gradient effects

Decoder can get very large for $n$ large

Routing of the $2n$ switch signals can become very long and consume lots of area
End of Lecture 33