Interpolating ADCs, Folding ADCs and Oversampling Converters

Tuesday 16th of March, 2009, 9:15 – 11:00

Snorre Aunet, sa@ifi.uio.no
Nanoelectronics Group, Dept. of Informatics
Office 3432
Last time – and today, Tuesday 16th of March:

Last time:
13.1 Integrating Converters
13.2 Successive-Approx. Converters
13.3 Algorithmic (or cyclic) A/D Converters
13.4 Flash (or parallel) converters
13.5 Two-Step A/D converters
13.8 Pipelined A/D Converters
13.9 Time-Interleaved A/D Converters

Today – from the following chapters:
13.6 Interpolating A/D Converters
13.7 Folding A/D Converters
14.1 Oversampled converters
Interpolating ADCs. Rightmost interpol. = 4 (1/4)

- Reduced complexity compared to Flash ADCs \( \rightarrow \) reduced input capacitance and slightly reduced power.
- In the **mathematical** subfield of **numerical analysis**, **interpolation** is a method of constructing new data points within the range of a **discrete set** of known data points.
Interpolating ADCs (2/4)

- Uses input amplifiers behaving as linear amplifiers near their threshold voltages, allowed to saturate for moderately large input signals.
- Thus “noncritical” latches need only determine the sign of the amplifier outputs.
Interpolating ADCs (3/4)

- Amplifier outputs $V_1$ and $V_2$ as well as their interpolated values are shown lowermost (fig. 13.24)
- The reference points created from interpolated values (for example $V_{2a}$, $V_{2b}$, $V_{2c}$) have latches potentially triggering in order, for increasing (or decreasing) input.
- For good linearity the interpolated signals need only cross the latch threshold at the correct points
Interpolating ADCs (4/4)

- To achieve good linearity $V_1$ and $V_2$ need to be linear between their own thresholds. In figure 13.24 this linear region corresponds to $0.25 < V_{in} < 0.5$ (horizontally).

- For fast operation the delays to each of the latches must be made to equal each other as much as possible. In fig. 13.25 this is done using resistors.
Example, based on Fig. 13.23 (1/2)

- $V_{in} = 0.4 \text{ V}$, gain of $10^{-10}$, logic levels of 0 and 5 volts.
- $V_4 = 5 \text{ V}$
- $V_3 = 5 \text{ V}$
- $V_2 = 3.5 \text{ V}$
- $V_1 = 1.0 \text{ V}$
Example – interpolating ADC (2/2)
Folding A/D Converters (13.7)

- The number of latches is reduced compared to the interpolating ADC, and even more from FLASH.
- The figure shows a 4 bit converter with folding rate of 4.
- A group of LSBs are found separately from a group of MSBs.
- The MSB converter determines whether the input signal, $V_{in}$, is in one of four voltage regions (between 0 and $\frac{1}{4}$, $\frac{1}{4}$ and $\frac{1}{2}$, $\frac{1}{2}$ and $\frac{3}{4}$, or $\frac{3}{4}$ and 1).
- $V_1$ to $V_4$ produce a thermometer code for each of the four MSB regions.
Similar to folding block responses on previous slide..

- Bipolar folder outputs
- Ex: Input 1.05:
  - F1 > threshold=0 -> "1"
  - F2 > threshold=0 -> "1"
  - F3 > threshold=0 -> "1"
  - F4 < threshold=0 -> "0"
- Thermometer code produced for each of the four MSB regions (between 0 and ¼, ¼ and ½, ½ and ¾, or ¾ and 1 for previous slide)
- (in certain respects related to interpolation in Fig 13.24)
Folding block with a folding rate of four

- Input-output response for the cross-coupled differential pair is shown lowermost

- $V_{out}$ is low if, and only if, both $V_a$ and $V_b$ are low, otherwise high

- The output from a folding block is at a much higher frequency than the input signal, limiting the practical folding rate.

- Differential solutions in practice
Folding and Interpolating ADC

- By introducing interpolation, the number of folding blocks is reduced.
- Input capacitance is reduced (if both folding and interpolating is combined).
- Folding-rate of four and interpolate-by-two.
- (Literature references on page 523)
### Interpolating and folding and interpolating ADCs

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Sampling rate</th>
<th>ENOB</th>
<th>Power dissip.</th>
<th>Supply voltage</th>
<th>architecture</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit</td>
<td>100 MHz</td>
<td>6.5 bit@5V, 7.1 bit@8V</td>
<td>1.2W@5V</td>
<td>5 or 8 V</td>
<td>interpolating</td>
<td>Steyaert, Roovers, Craninckx, CICC 1993</td>
</tr>
<tr>
<td>5 bit</td>
<td>5 GHz</td>
<td>4 bit at 5GHz</td>
<td>113 mW@1V</td>
<td>1 V</td>
<td>interpolating</td>
<td>Wang, Liu, VLSI-DAT ’2007</td>
</tr>
<tr>
<td>6 bit</td>
<td>200 MHz</td>
<td>5.35 bit</td>
<td>35 <a href="mailto:mW@3.3V">mW@3.3V</a></td>
<td>3.3V</td>
<td>folding and interpolating</td>
<td>Yin, Wang, Liu, ICSICT, 2008</td>
</tr>
<tr>
<td>6 bit</td>
<td>200 MHz</td>
<td>5.5 bit</td>
<td>78.8 <a href="mailto:mW@2.5V">mW@2.5V</a></td>
<td>2.5V</td>
<td>folding and interpolating</td>
<td>Silva, Fernandes, ISCAS, 2003</td>
</tr>
</tbody>
</table>

16. mars 2010
Oversampling converters (chapter 14 in "J & M")

- For high resolution, low-to-medium-speed applications like for example digital audio
- Relaxes requirements placed on analog circuitry, including matching tolerances and amplifier gains
- Simplify requirements placed on the analog anti-aliasing filters for A/D converters and smoothing filters for D/A converters.
- Sample-and-Hold is usually not required on the input
- Extra bits of resolution can be extracted from converters that samples much faster than the Nyquist-rate. Extra resolution can be obtained with lower oversampling rates by exploiting noise shaping
Resolution and clock cycles per sample

- Dependence of achievable resolution and required clock cycles per sample for various ADC systems.
Nyquist Sampling and Oversampling

- Figure from [Kest05]
- Straight oversampling gives an SNR improvement of 3 dB / octave
- $f_s > 2f_0$ ($2f_0 =$ Nyquist Rate)
- $\text{OSR} = \frac{f_s}{2f_0}$
- $\text{SNR}_{\text{max}} = 6.02N + 1.76 + 10\log(\text{OSR})$
Oversampled converters; High resolution and relatively low speed

Figure 1. ADC architectures, applications, resolution, and sampling rates.

**Which ADC Architecture Is Right for Your Application?**

By Walt Kester [walt.kester@analog.com]
The quantization noise is the difference between the input and output values.

\[ e(n) = y(n) - x(n) \]

This model is exact under the assumption that the quantization error is strongly related to the input signal (p. 532 in “J.A.M.”). The model becomes approximate when assumptions are made about the statistical properties of \( e(n) \), such as \( e(n) \) being an independent white-noise signal. This model leads to a simpler understanding of \( \Delta \) and with some exceptions is usually reasonably accurate.

- If \( x(n) \) is very active, \( e(n) \) can be approximated as an independent random number uniformly distributed between \( \pm \frac{\Delta}{2} \), where \( \Delta \) equals the difference between two adjacent quantization levels.

Thus, the quantization noise power equals \( \frac{\Delta^2}{12} \) (sec. 11.3) and is independent of the sampling frequency \( f_s \).

The spectral density of \( e(n) \) \( S_e(f) \) is white (constant over freq.) and all its power within \( \pm f_s/2 \), as shown in the figure:

\[
\int_{-f_s/2}^{f_s/2} S_e(f) df = \int k^2_x df = (k_x^2) f_s = \frac{\Delta^2}{12}
\]

The spectral density height is calculated by noting that the total noise is \( \Delta^2/12 \) and with a two-sided df. of power equals the area under \( S_e(f) \) within \( \pm f_s/2 \):
14.1 **Output and quantization errors for two quantizers**

**Quantizer 1:**

\[ y_1(n) = \{0.0, 0.5, 0, 1.0, 0.5, -0.5\} \]

\[ e_1(n) = \{-0.01, 0.19, -0.11, -0.36, -0.02, 0.20\} \]

\[ e_1(n) \text{ is approximated as an independent random number uniformly distributed between } \pm \frac{\Delta}{2}, \text{ where } \Delta \text{ equals the difference between two adjacent quantization levels.} \]

\[ P_e = \frac{\Delta^2}{12} \]

\[ \Delta^2 = \left(\frac{0.5}{2}\right)^2 = 0.02083 \text{ [w]} \]

**Quantizer 2:**

\[ y_2(n) = \{1.0, 1.0, -1.0, 1.0, 1.0, -1.0\} \]

\[ e_2(n) = \{0.99, 0.69, -0.69, 0.20, 0.48, -0.30\} \]

"OVERSAMPLING IS BASED ON THE ASSUMPTION THAT AN ADC'S TOTAL QUANTIZATION NOISE POWER (VARIANCE) IS THE SQUARED VALUE OF THE CONVERTER'S LEAST SIGNIFICANT BIT (LSB) VOLTAGE DIVIDED BY 12" (19NS5 05)

\[ S_{\text{eff}}(f) = 0.02083 \frac{\text{w}}{\text{rad/sample}} \]

\[ S_c(f) = 0.00033 \frac{\text{w}}{\text{rad/sample}} \]

\[ S_\Delta(f) = 0.0033 \frac{\text{w}}{\text{rad/sample}} \]
**Oversampling**

Signal of interest is bandlimited to \( f_0 \).

Nyquist sampling:

- \( f_0 \), \( f_s \), \( 2f_s \)

Oversampling:

- \( f_0 \), \( f_s > 2f_0 \), \( f_s \)

\[ \text{OSR} = \frac{f_s}{2f_0} \]

After quantization, \( y_1(n) \) is filtered by \( H(f) \) to create \( y_2(n) \) that eliminates quantization noise (together with any other signals) greater than \( f_0 \).

\[ |H(f)| \]

\[ -\frac{f_s}{2} \quad f_0 \quad f_s \quad f_{s/2} \]

**Advantage** P. 535 peak

If the input is sinusoidal, its maximum peak value without clipping is \( 2^N (\Delta/2) \).

For this case, the signal power \( P_s \) has a power equal to \( P_s = \left( \frac{\Delta^2}{2} \right)^N = \frac{\Delta^2 2^N}{8} \).

The power of the input signal within \( y_2(n) \) remains the same as before since we assumed the signal's frequency content is below \( f_0 \).

However, the quantization noise power is reduced to

\[ P_e = \int_{-f_s/2}^{f_s/2} S_e(f) |H(f)|^2 \, df = \int_{-f_0}^{f_0} h^2 \, df \]

\[ = \frac{2f_0 \cdot \Delta^2}{12} = \frac{\Delta^2}{12} \left[ \frac{1}{\text{OSR}} \right] \]

Therefore, doubling OSR decreases the quantization noise power by one-half, or equivalently, 3 dB (or equiv. 0.566).

\[ \text{SNR}_{\text{max}} = 10 \log \left( \frac{P_s}{P_e} \right) = 10 \log \left( \frac{\sqrt{2} 2^N}{\Delta^2} \right) + 10 \log (\text{OSR}) \]

\[ = 6.02N + 1.76 + 10 \log (\text{OSR}) \, \text{dB} \]

Due to n-bit quantizer due to oversampling.
\[ SNR_{\text{max}} = 10 \log \left( \frac{P_s}{P_c} \right) \textnormal{ and } P_s = \frac{\Delta^2 2^N}{8} \textnormal{ and } P_c = \frac{\Delta^2}{12 \cdot \text{OSR}} \]

\[ SNR_{\text{max}} = 10 \log \left( \frac{\Delta^2 2^N}{8} \cdot \frac{1}{\text{OSR}} \right) = 10 \log \left( \frac{\Delta^2 2^N}{12 \cdot \text{OSR}} \right) = 10 \log \left( \frac{3}{2} \cdot 2^N \cdot \text{OSR} \right) \]

\[ = 10 \log \left( \frac{3}{2} \cdot 2^N \right) + 10 \log \text{OSR} \]

\[ = 10 \log \left( \frac{3}{2} \right) + 10 \log 2^N - 10 \log \text{OSR} \]

\[ = 10 \log 2^N + 1.76 + 10 \log \text{OSR} \]

\[ = 10 \cdot 2^N \cdot \log 2 + 1.76 + 10 \log \text{OSR} \]

\[ = 10 \cdot 2^N \cdot 0.301 + 1.76 + 10 \log \text{OSR} \]

\[ = 6.02 N + 1.76 + 10 \log (\text{OSR}) \quad \text{(14.18)} \]
Oversampling (without noise shaping)

- Doubling of the sampling frequency increases the dynamic range by 3 dB = 0.5 bit.
- To get a high SNR a very high fs is needed → high power consumption.
- Oversampling usually combined with noise shaping and higher order modulators, for higher increase in dynamic range per octave ("OSR")

$$P_e = \int_{-f_0}^{f_0} S_e^2(f) df = \frac{\Delta^2}{12} \cdot \frac{1}{OSR}$$

• Total støy er gitt av:
SNR_{max} = 6.02N + 1.76 + 10\log(\text{OSR}) \ [\text{dB}]

SNR improvement 0.5 bits / octave
**Example 14.3**

Given that a 1-bit A/D converter has a 6-dB SNR, what sample rate is required using oversampling (no noise shaping) to obtain a 96-dB SNR (i.e., 16 bits) if \( f_0 = 25 \text{ kHz} \)? (Note that the input into the A/D converter has to be very active for the white-noise quantization model to be valid, a difficult arrangement when using a 1-bit quantizer with oversampling without noise shaping).

**Solution**

Oversampling (without noise shaping) gives 3 dB/octave where 1 octave implies doubling the sampling rate. We require 90 dB divided by 3 dB/octave, or 30 octaves. Thus, the required sampling rate, \( f_s \), is

\[
 f_s = 2^{30} \times f_0 \approx 54,000 \text{ GHz}
\]

This example shows why noise shaping is needed to improve the SNR faster than 3 dB/octave, since 54,000 GHz is highly impractical.
Advantages of 1-bit A/D converters (p.537 in ”J&M”)

- Oversampling improves signal-to-noise ratio, but not linearity
- Ex.: 12-bit converter with oversampling needs component accuracy to match better than 16-bit accuracy if a 16-bit linear converter is desired
- Advantage of 1-bit D/A is that it is inherently linear. Two points define a straight line, so no laser trimming or calibration is required
- Many audio converters presently use 1-bit converters for realizing 16- to 18-bit linear converters (with noise shaping).
Problems with some 1-bit converters ((?))

Why 1-Bit Sigma-Delta Conversion is Unsuitable for High-Quality Applications

by

Stanley P. Lipshitz and John Vanderkooy
Audio Research Group, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada

ABSTRACT

Single-stage, 1-bit sigma-delta converters are in principle imperfectible. We prove this fact. The reason, simply stated, is that, when properly dithered, they are in constant overload. Prevention of overload allows only partial dithering to be performed. The consequence is that distortion, limit cycles, instability, and noise modulation can never be totally avoided. We demonstrate these effects, and using coherent averaging techniques, are able to display the consequent profusion of nonlinear artefacts which are usually hidden in the noise floor. Recording, editing, storage, or conversion systems using single-stage, 1-bit sigma-delta modulators, are thus iminlical to audio of the highest quality. In contrast, multi-bit sigma-delta converters, which output linear PCM code, are in principle infinitely perfectible. (Here, multi-bit refers to at least two bits in the converter.) They can be properly dithered so as to guarantee the absence of all distortion, limit cycles, and noise modulation. The audio industry is misguided if it adopts 1-bit sigma-delta conversion as the basis for any high-quality processing, archiving, or distribution format to replace multi-bit, linear PCM.

Audio Engineering Society
Convention Paper 5395

Presented at the 110th Convention
2001 May 12-15 Amsterdam, The Netherlands
Oversampling with noise shaping (14.2)

- Oversampling combined with noise shaping can give much more dramatic improvement in dynamic range each time the sampling frequency is doubled.
- The sigma delta modulator converts the analog signal into a noise-shaped low-resolution digital signal.
- The decimator converts to a high resolution digital signal.
Multi-order sigma delta noise shapers (Sangil Park, Motorola)

- Nyquist Sampler (1 bit)
- First Order $\Sigma\Delta$ Modulator
- Second Order $\Sigma\Delta$ Modulator
- Third Order $\Sigma\Delta$ Modulator

**Note:** Higher order Noise Shaper has less baseband noise
OSR, modulator order and Dynamic Range

- 2 X increase in M → (6L+3)dB or (L+0.5) bit increase in DR.
L: sigma-delta order

Oversampling and noise shaping
Ex. 14.5

• Given that a 1-bit A/D converter has a 6 dB SNR, which sample rate is required to obtain a 96-dB SNR (or 16 bits) if $f_0 = 25$ kHz for straight oversampling as well as first- and second-order noise shaping?

• Oversampling with no noise shaping: From ex. 14.3 we know that straight oversampling requires a sampling rate of 54 THz.

• $(6.02N+1.76+10 \log_{10}(OSR)) = 96$

\[6 + 10 \log_{10} OSR = 96\]

\[10 \log_{10} OSR = 90\]
Ex. 14.5

\[ \text{SNR}_{\text{max}} = 6.02N + 1.76 - 5.17 + 30 \log(\text{OSR}) \] (14.26)

We see here that doubling the OSR gives an SNR improvement for a first-order modulator of 9 dB or, equivalently, a gain of 1.5 bits/octave. This result should be compared to the 0.5 bits/octave when oversampling with no noise shaping.

• Oversampling with 1st order noise shaping:
• \[ 6 - 5.17 + 30 \log(\text{OSR}) = 96 \]
• \[ 30 \log (\text{OSR}) = 96 - 6 + 5.17 = 95.17 \]
  A doubling of the OSR gives an SNR improvement of 9 dB / octave for a 1st order modulator;
  \[ 95.17 / 9 = 10.57 \quad 2^{10.56} \times 2 \times 25 \text{ kHz} = 75.48 \text{ MHz} \]

OR: \[ \log(\text{OSR}) = 95.17/30 = 3.17 \rightarrow \text{OSR} = 1509.6 \]
\[ 1509.6 \times (2 \times 25 \text{ kHz}) = 75.48 \text{ MHz} \]
Ex. 14.5

\[
\begin{align*}
\text{SNR}_{\text{max}} &= 6.02N + 1.76 - 12.9 + 50 \log(\text{OSR}) \\
&= 102.9 + 50 \log(\text{OSR})
\end{align*}
\]

We see here that doubling the OSR improves the SNR for a second-order modulator by 15 dB or, equivalently, a gain of 2.5 bits/octave.

The realization of the second-order modulator using switched-capacitor techniques is straightforward and is left as an exercise for the interested reader.

- Oversampling with 2nd order noise shaping:
  - \(6 - 12.9 + 50 \log(\text{OSR}) = 96\) \hspace{1cm} \(\text{OSR} = \frac{f_s}{2f_0}\)
  - \(50 \log(\text{OSR}) = 96 - 6 + 12.9 = 102.9\)

  A doubling of the OSR gives an SNR improvement of 15 dB / octave for a 2nd order modulator;

  \[
  102.9 / 15 = 6.86 \quad 2^{6.86} \times 2 \times 25 \text{ kHz} = 5.81 \text{ MHz}
  \]
Ex. 14.5 "point":

- 2X increase in M → (6L+3)dB or (L+0.5) bit increase in DR.
- L: sigma-delta order
- 6 db Quantizer, for 96 dB SNR:
- Plain oversampling: $f_s = 54$ GHz
- 1st order: $f_s = 75.48$ MHz
- 2nd order: $f_s = 5.81$ MHz

Exam problem (INF4420) below

A sampled signal is bandlimited to $f_0 = 22$ kHz. What is the sampling frequency, $f_s$, for an oversampling ratio ("OSR") of 128?

A 1-bit analog-to-digital converter ("ADC") has an inherent 6 dB SNR. Which maximum SNR is acquired by combining it with strict oversampling and an OSR of 128, if no noise shaping is used? What is the maximum SNR in the similar case exploiting 2nd order noise shaping?

If a 1-bit ADC using 3rd order noise shaping has a maximum SNR of 125 dB for an OSR of 128, what is the expected maximum SNR if the OSR is reduced to 32?
Sigma Delta converters, ISSCC 2008

ISSCC - Foremost global forum
"CT": continuous time
2nd order sigma delta modulator

Fig. 1. Block diagram of second-order $\Sigma\Delta$ modulator with decimator.

Fig. 10. Second-order $\Sigma\Delta$ modulator implementation.

Fig. 13. Measured SNR for a sampling frequency of 4 MHz and a signal frequency of 1.02 kHz.

Fig. 14. Maximum operating frequency.

15. Dynamic range as a function of the oversampling ratio for a sampling frequency of 4 MHz.

IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 23, NO. 6, DECEMBER 1988

The Design of Sigma-Delta Modulation Analog-to-Digital Converters

BERNHARD E. BOSER, STUDENT MEMBER, IEEE, AND BRUCE A. WOOLEY, FELLOW, IEEE
Additional literature

• Richard Lyons, Randy Yates: "Reducing ADC Quantization Noise", MicroWaves & RF, 2005
• Sangil Park: "Principles of sigma-delta modulators for analog to digital converters", Motorola
• Lecture Notes, University of California, Berkeley, EE247 Analog Digital Interface Integrated Circuits, Fall 07;http://inst.eecs.berkeley.edu/~ee247/fa07/
Next Time, 23/3-10:

- More from Chapter 14; Oversampling Converters (14.2, 14.3, 14.4, 14.5, 14.7)
- Beginning of chapter 16; Phase-Locked Loops (16.1)
Guide to Writing a Thesis

Department of Applied Electronics
Last updated 1997-07-25

Original manuscript written by Sven Mattsson

The Design and Implementation of a Nifty Gadget

Tidka Liz Book
April 32, 1992

Abstract
What is all this about?
Why should I read this thesis?
Is it any good?
What's new?

Preface
Have you done anything that doesn't have to do with your research?
Have you published parts of this work before?

Acknowledgement
Who is your advisor?
Did anyone help you?
Who funded this work?
What's the name of your favorite pet?

1 Introduction
What is the use of a Nifty Gadget?
What is the problem?
How can it be solved?
What are the previous approaches?
What is your approach?
Why do it this way?
What are your results?
Why is this better?
Is this a new approach?
Why haven't anyone done it before?

2 Theoretical background
What is the required background knowledge?
Where can I find it?

2.1 Various approaches to Nifty Gadgets
What is the relevant prior work?
Where can I find it?
Why should it be done differently?
Has anyone attempted your approach previously?
Where is that work reported?

2.2 Nifty Gadget my way
What is the outline of your way?
Have you published it before?

3 My implementation of a Nifty Gadget
Can you describe your implementation in detail?
Why did you use this technology?
How does the theory relate to your implementation?
What are your underlying assumptions?
What did you neglect and what simplifications have you made?
What tools and methods did you use?
Why use these tools and methods?

4 Nifty Gadget results
Did you actually build it?
How can you test it?
How did you test it?
Why did you test it this way?
Are the results satisfactory?
Why should you (not) test it more?
What compensations had to be made to interpret the results?
Why did you succeed/fail?

5 Discussion
Are your results satisfactory?
Can they be improved?
Is there a need for improvement?
Are other approaches worth trying out?
Will some restrictions be lifted?
Will you save the world with your Nifty Gadget?

6 References
What is the background reading list?
Where is the related work?
Where is the prior work?
Where can I find important material?

Appendix A
Can you outline fetal calculus or whatever complicated theory or results you are using that will obscure the text?

Appendix B
A thesis should discuss the following topics:

- Introduction
  Presentation of the problem or phenomenon to be addressed, the situation where the problem or phenomenon occurs, and references to earlier relevant research.

  Common errors
  Problem not properly specified or formulated; insufficient references to earlier work.

- Purpose
  What can be gained by more knowledge about the problem or phenomenon.

  Common errors
  The purpose is not mentioned, not connected to earlier research, or not in line with what the actual contents of the thesis.

- Problem/Hypothesis
  Questions that need to be answered to reach the goal and/or hypothesis formulated be means of underlying theories.

  Common errors
  Missing problem description, deficiencies in the connections between questions, badly formulated hypothesis.

- Method
  Choice of an adequate method with respect to the purpose and problem/hypothesis.