ADC Converters
– Techniques to reduce flash ADC complexity
  • Interpolating (continued)
  • Folding
  • Multi-Step ADCs
    – Two-Step flash
    – Pipelined ADCs

Summary Last Lecture
ADC Converters
– Comparator design (continued)
  • Comparator architecture examples

– Techniques to reduce flash ADC complexity
  • Interpolating (to be continued)
Interpolation

- Idea
  - Reduce number of preamps & instead interpolate between preamp outputs

- Reduced number of preamps
  - Reduced input capacitance
  - Reduced area, power dissipation

- Same number of latches (2^B-1)

- Important "side-benefit"
  - Decreased sensitivity to preamp offset → Improved DNL

Preamp Output

Zero crossings (to be detected by latches) at \( V_{in} = \)

\[
V_{ref1} = 1\ \Delta \\
V_{ref2} = 2\ \Delta
\]
**Differential Preamp Output**

@ $V_{in} = V_{ref1} = 1 \Delta$

$V_{ref2} = 2 \Delta$

Note: Additional crossing of $A_1\&-A_2$ (and $A_2\&-A_1$)

$A_1+A_2$ cross zero at:

$V_{ref12} = 0.5'(1+2) \Delta = 1.5 \Delta$

---

**Interpolation in Flash ADC**

Half as many reference voltages and preamps

Interpolation factor: $x2$

Example: For 10-bit straight Flash ADC need $2^B=1024$ preamps

compared $2^{B-1}=512$ for $x2$ interpolation

Possible to accomplish higher interpolation factor

$\rightarrow$ Interpolation at the output of preamps

Compare $A_2\&-A_1$

$\rightarrow$ Comparator output is sign of $A_1+A_2$
Interpolation in Flash ADC
Preamp Output Interpolation

Interpolate between two consecutive output via impedance Z

Choices of Z:
1. Resistors (Kimura)
2. Capacitors (Kusumoto)
3. Current mode (Roovers)

Vin
A1
A2
Z
Z
Vo1
Vo2
Vo1.5 = (Vo1 + Vo2)/2


Higher Order Resistive Interpolation

- Resistors produce additional levels
- With 4 resistors per side, the “interpolation factor” M=8
  (M→ ratio of latches/preamps)

DNL Improvement

- Preamp offset distributed over M resistively interpolated voltages:
  \[ \rightarrow \text{Impact on DNL divided by } M \]

- Latch offset divided by gain of preamp
  \[ \rightarrow \text{Use "large" preamp gain} \]
  \[ \rightarrow \text{Next: Investigate how large preamp gain can be} \]


Preamp Input Range

If linear region of preamp transfer curve do not overlap

\[ \rightarrow \text{Dead-zone in the interpolated transfer curve! Results in error} \]

\[ \rightarrow \text{Linear consecutive preamp input ranges must overlap} \]
\[ \text{i.e. range } > \Delta \]

Sets upper bound on preamp gain \[ < \frac{V_{DD}}{\Delta} \]
Interpolated-Parallel ADC

10-bit overall resolution:
→ 7-bit flash (127 comparators and 128 resistors) & x8 interpolation


Measured Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>10 b (7+3)</td>
</tr>
<tr>
<td>Maximum conversion frequency</td>
<td>300 MHz</td>
</tr>
<tr>
<td>Integral non-linearity</td>
<td>±1.0 LSB</td>
</tr>
<tr>
<td>Differential non-linearity</td>
<td>±0.4 LSB</td>
</tr>
<tr>
<td>SNR/THD 10MHz input</td>
<td>58/-59 dB</td>
</tr>
<tr>
<td>SNR/THD 50MHz input</td>
<td>48/-47 dB</td>
</tr>
<tr>
<td>Input capacitance</td>
<td>8 pF</td>
</tr>
<tr>
<td>Input range</td>
<td>2 V</td>
</tr>
<tr>
<td>Power supply</td>
<td>-5.2V</td>
</tr>
<tr>
<td>Power dissipation</td>
<td>4.0W</td>
</tr>
<tr>
<td>Chip size</td>
<td>9.0 × 4.2 mm²</td>
</tr>
<tr>
<td>Element count</td>
<td>36,000</td>
</tr>
<tr>
<td>Technology</td>
<td>1.0 μm bipolar:ft=25GHz</td>
</tr>
</tbody>
</table>

Interpolation Summary

- Consecutive preamp transfer curve need to have overlap → Limits gain of preamp to $V_{DD}/\Delta$

- The added impedance at the output of the preamp typically reduces the bandwidth and affects the maximum achievable frequencies

- DNL due to preamp offset reduces by interpolation factor M

- Interpolation reduces # of preamps and thus reduces input C- however, the # of required latches the same as “straight” Flash → Use folding to reduce the # of latches

Folding Converter

- Two ADCs operating in parallel
  - MSB ADC
  - Folder + LSB ADC
- Significantly fewer comparators than flash
- Fast
- Typically, nonidealities in folder limit resolution to ~10Bits
Example: Folding Factor of 4

- Folding factor: number of folds
- Folder maps input to smaller range
- MSB ADC determines which fold input is in
- LSB ADC determines position within fold
- Logic circuit combines LSB and MSB results

How are folds generated?

- Fold 1: \( V_{out} = +V_{in} \)
- Fold 2: \( V_{out} = -V_{in} + V_{FS}/2 \)
- Fold 3: \( V_{out} = +V_{in} - V_{FS}/2 \)
- Fold 4: \( V_{out} = -V_{in} + V_{FS} \)

Note: Sign change every other fold + reference shift
Generating Folds via Source-Couple Pairs

Vref1 < Vref2 < Vref3 < Vref4
As Vin changes, only one of M1, M3, M5, M7 is on depending on the input level

CMOS Folder Output

CMOS folder transfer curve max. min. portions:
  → Rounded
  → Accurate at zero-crossings

In fact, most folding ADCs do not use the folds, but only the zero-crossings!
Parallel Folders Using Only Zero-Crossings

\[ V_{in} \]

\[
\begin{align*}
\text{Folder 4} & \xrightarrow{V_{ref} + 3/4 \Delta} \text{Comparator} \\
\text{Folder 3} & \xrightarrow{V_{ref} + 2/4 \Delta} \text{Comparator} \\
\text{Folder 2} & \xrightarrow{V_{ref} + 1/4 \Delta} \text{Comparator} \\
\text{Folder 1} & \xrightarrow{V_{ref} + 0/4 \Delta} \text{Comparator} \\
\end{align*}
\]

Logic

LSB bits (to be combined with MSB bits)

---

Parallel Folder Outputs

- 4 folders with 4 folds each
- 16 zero crossings
- \( \rightarrow \) 4 LSB bits
- Higher resolution
  - More folders
    - \( \rightarrow \) Large complexity
  - Interpolation
Folding & Interpolation

Folder 4
$V_{ref} + 3/4 \Delta$

Folder 3
$V_{ref} + 2/4 \Delta$

Folder 2
$V_{ref} + 1/4 \Delta$

Folder 1
$V_{ref} + 0/4 \Delta$

Fine Flash ADC

Folder / Interpolator Output

Example: 4 Folders + 4 Resistive Interpolator per Stage

Note: Output of two folders + corresponding interpolator only shown
Folder / Interpolator Output
Example: 2 Folders + 8 Resistive Interpolator per Stage

Non-linear distortion
→ Interpolate only between closely spaced folds to avoid nonlinear distortion

A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter

Ref: B. Nauta and G. Venes, JSSC Dec 1985, pp. 1302-8
A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter

Note: Total of 40 comparators compared to $2^8-1=255$ for straight flash

Ref: B. Nauta and G. Venes, JSSC Dec 1985, pp. 1302-8
Time Interleaved Converters

- Example:
  - 4 ADCs operating in parallel at sampling frequency $f_s$
  - Each ADC converts on one of the 4 possible clock phases
  - Overall sampling frequency = $4f_s$
  - Note T/H has to operate at $4f_s$!

- Extremely fast:
  Typically, limited by speed of T/H

- Accuracy limited by mismatch in individual ADCs (timing, offset, gain, ...)

Two-Step (2+2) Example

- Using only one ADC: output contains large quantization error
- "Missing voltage" or "residue" ($\varepsilon_{q1}$)
- Idea: Use second ADC to quantize and add $-\varepsilon_{q1}$
Two Stage Example

• Use DAC to compute missing voltage
• Add quantized representation of missing voltage
• Why does this help? How about $\epsilon_{q2}$?

![Diagram showing two-stage example]

Two Step (2+2) Flash ADC

![Diagram showing two-step flash ADC]

4-bit Straight Flash ADC

Ideal 2-step Flash ADC
Two Stage Example

- Fine ADC is re-used $2^2$ times
- Fine ADC's full scale range needs to span only 1 LSB of coarse quantizer

$$\epsilon_{q2} = \frac{V_{ref_2}}{2^2} = \frac{V_{ref_1}}{2^2 \cdot 2^2}$$

Two-Stage (2+2) ADC Transfer Function
Residue or Multi-Step Type ADC

**Issues**

- **Operation:**
  - Coarse ADC determines MSBs
  - DAC converts the coarse ADC output to analog: Residue is found by subtracting \( (V_{in} - V_{DAC}) \)
  - Fine ADC converts the residue and determines the LSBs
  - Bits are combined in digital domain

- **Issue:**
  1. Fine ADC has to have precision in the order of overall ADC \( 1/2 \) LSB
  2. Speed penalty → Need at least 1 clock cycle per extra series stage to resolve one sample

---

**Solution to Issue (1)**

- Accuracy needed for fine ADC relaxed by introducing inter-stage gain
  - Example: By adding gain of \( x(G=2^{B1}=4) \) prior to fine ADC in \( (2+2) \) bit case, precision required for fine ADC is reduced to \( 2 \) bit only!
  - Additional advantage: coarse and fine ADC can be identical stages
Solution to Issue (2)

- Conversion time significantly decreased by employing T/H between stages
  - All stages busy at all times → operation concurrent
  - During one clock cycle coarse & fine ADCs operate concurrently:
    - First stage samples/converts/generates residue of input signal sample # \( n \)
    - While 2nd samples/converts residue associated with sample # \( n-1 \)

\[
D_{\text{out}} = V_n + \epsilon q_1 - \epsilon q_1 + \epsilon q_2
\]

Pipelined A/D Converters

- Ideal operation
- Errors and correction
  - Redundancy
  - Digital calibration
- Implementation
  - Practical circuits
  - Stage scaling
Pipeline ADC

Block Diagram

- Idea: Cascade several low resolution stages to obtain high overall resolution (e.g. 10bit ADC can be built with series of 10 ADCs each 1-bit only!)
- Each stage performs coarse A/D conversion and computes its quantization error, or "residue"
- All stages operate concurrently

Pipeline ADC

Characteristics

- Number of components (stages) grows linearly with resolution
- Pipelining
  - Trading latency for conversion speed
  - Latency may be an issue in e.g. control systems
  - Throughput limited by speed of one stage → Fast
- Versatile: 8...16bits, 1...200MS/s
- Many analog circuit non-idealities can be corrected digitally
Pipeline ADC
Concurrent Stage Operation

- Stages operate on the input signal like a shift register
- New output data *every* clock cycle, but each stage introduces at least ½ clock cycle latency

Digital output
\((B_1 + B_2 + \ldots + B_k)\) Bits

Note: One conversion per clock cycle & 7 clock cycle latency

[Analog Devices, AD 9226 Data Sheet]
### Pipeline ADC Data Alignment

- Digital shift register aligns sub-conversion results in time

### Cascading More Stages

- LSB of last stage becomes very small
- Impractical to generate several $V_{\text{ref}}$
- All stages need to have full precision
Pipeline ADC
Inter-Stage Gain Elements

- Practical pipelines by adding inter-stage gain → use single $V_{\text{ref}}$
- Precision requirements decrease down the pipe
  - Advantageous for noise, matching (later)

Complete Pipeline Stage

"Residue Plot"
E.g.:
B=2
G=2^2 = 4
Pipeline ADC Errors

• We cannot build perfect ADCs, DACs and gain elements
• How can we tolerate/correct errors?
• Let's first look at sub-ADC errors
• Assumptions:
  – Ideal DAC, ideal gain elements, only nonideality due to sub-ADC comparator offset

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]

\[
D_{out} = V_{in,ADC} + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \sum_{i=1}^{n} \left( I - \frac{G_i}{G_{di}} \right) + \frac{\varepsilon_q}{\prod_{j=1}^{n} G_{dj}} \]
Pipeline ADC Model

• If the "Analog" and "Digital" gain/loss is precisely matched:

\[
D_{out} = V_{in,ADC} + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_j}
\]

\[
D.R. = 20 \log \left( 2^B_n \times \prod_{j=1}^{n-1} G_j \right)
\]

\[
B_{ADC} = B_n + \sum_{j=1}^{n-1} \log_2 G_j
\]

Pipeline ADC Observations

• The aggregate **ADC resolution is independent of sub-ADC resolution**

• *Effective* stage resolution \( B_j = \log_2(G_j) \)

• Overall conversion error does not (directly) depend on sub-ADC errors!

• Only error term in \( D_{out} \) contains quantization error associated with the last stage

• So why do we care about sub-ADC errors?
  ➢ Go back to two stage example
Pipeline ADC Sub-ADC Errors

\[ D_{\text{out}} = V_{\text{in},\text{ADC}} + \frac{\varepsilon_{\text{en}}}{\prod_{j=1}^{n-1} G_j} \]

\[ D_{\text{out}} = V_{\text{in},\text{ADC}} + \frac{\varepsilon_{\text{en}}}{G_1} \]

Grows outside ½ LSB bounds

Ideal 2-Stage Pipelined ADC

2-Stage Pipelined ADC with Coarse ADC Comp. Offset
Pipeline ADC

1st-Stage Comparator Offset

Problem: \( V_{res1} \) exceeds 2nd pipeline stage overload range

First stage ADC Levels:
(Levels normalized to LSB)
Ideal comparator threshold: -1, 0, +1
Comparator threshold including offset: -1, 0.3, +1

Overall ADC Transfer Curve

Missing Code!

Pipeline ADC
Three Ways to Deal with Errors

- All involve "sub-ADC redundancy"
- Redundancy in stage that produces errors
  - Choose gain for 2nd stage < \( 2^{B_1} \)
  - Higher resolution sub-ADC
- Redundancy in succeeding stage(s)
(1) Inter-Stage Gain Following 1\textsuperscript{st} stage < 2\textsuperscript{B1}

- Choose \( G_1 \) slightly less than 2\textsuperscript{B1}
- Effective stage resolution becomes non-integer
  \[ B_{\text{eff}} = \log_2 G_1 \]

Ref: A. Karanicolas et. al., JSSC 12/1993

Correction Through Redundancy

- “enlarged” residuum still within input range of next stage

If \( G_1 = 2 \) instead of 4
  - Only 1Bit resolution from first stage (3-Bit total)
  - No overall error!
(2) Higher Resolution Sub-ADC

- Keep $G_i = 2^{B_1}$ (e.g. keep $G_i = 4$)
- Add extra decision levels in sub-ADC (e.g. add 1 extra bit to 1st stage)
- E.g. $B_1 = B_{\text{eff}} + 1$

Ref: Singer et. al., VSLI1996

Vin, ADC
ADC
B1 bits
Vref
Vref
Vref
Vin
Vref
Vin
Vref

(3) Over-Range Accommodation Through Increase in Following Stage Resolution

- No redundancy in stage with errors
- Add extra decision levels in succeeding stage

Ref: Opris et. al., JSSC 12/1998
Redundancy

- The preceding analysis applies to any stage in an n-stage pipeline.
- Can always perceive a multi-stage pipelined ADC as a single stage + backend ADC.

\[ V_{in} \rightarrow B_1 \text{ bits} \rightarrow B_2 \text{ bits} \rightarrow B_3 \text{ bits} \rightarrow B_4 \text{ bits} \]

Redundancy

- In literature, sub-ADC redundancy schemes are often called "digital correction" – a misnomer!
- No error correction takes place.
- We can tolerate sub-ADC errors as long as:
  - The residues stay "within the box", or
  - Another stage downstream "returns the residue to within the box" before it reaches the last quantizer.
- Let's calculate tolerable errors for popular "1.5 bits/stage" topology.
1.5 Bits/Stage Example

- Comparators placed strategically to minimize overhead
- \( G = 2 \)
- \( B_{\text{eff}} = \log_2 G = \log_2 2 = 1 \)
- \( B = \log_2 (2+1) = 1.589... \)

Ref: Lewis et. al., JSSC 3/1992

3-Stage 1.5-bps Pipelined ADC

- All three stages
  - Comparator with offset
- Overall transfer curve
  - No missing codes
  - Some DNL error

Inter-Stage Amplifier Offset

- Input referred converter offset – usually no problem
- Equivalent sub-ADC offset - accommodated through adequate redundancy