ADC Converters
-- Comparator design (continued)
  • Comparator architecture examples

-- Techniques to reduce flash ADC complexity
  • Interpolating
  • Folding
  • Interpolating & folding
  • Multi-Step ADCs
    -- Two-step flash
    -- Pipelined ADCs
      • Effect of sub-ADC, sub-DAC, gain stage non-idealities on overall ADC performance
Comparator with Auto-Zero


Flash ADC
Comparator with Auto-Zero

Flash ADC
Comparator with Auto-Zero

\[ V = A_{v1} \cdot A_{v2} \left( (V_{\text{in}} - V_{\text{ref}}) - (V_{\text{in}} - V_{\text{ref}}) - V_{\text{offset}} \right) \]

Substituting for \((V_{\text{in}} - V_{\text{ref}})\) from previous cycle:

\[ V = A_{v1} \cdot A_{v2} \left( (V_{\text{in}} - V_{\text{ref}}) - (V_{\text{in}} - V_{\text{ref}}) - V_{\text{offset}} \right) \]

Note: Offset is cancelled & difference between input & reference established

Bipolar Comparator Example

- Used in 8bit 400Ms/s & 6bit 2Gb/s flash ADC
- Signal amplification during φ1 high, latch operates when φ1 low
- Input buffers suppress kick-back & input current
- Separate ground and supply buses for front-end preamp → kick-back noise reduction


Auto-Zero Implementation

Comparator Example

Pipelined ADC Application

- Variation on Yukawa latch used w/o preamp
- Good for low resolution ADCs (in this case 1.5 bit/stage for a pipeline)
- Note: M1, M2, M11, M12 operate in triode mode
- M11 & M12 added to vary comparator threshold
- Conductance at node X is sum of $G_{M1}$ & $G_{M11}$


Comparator Example (continued)

- M1, M2, M11, M12 operate in triode mode with all having equal L
- Conductance of input devices:
  
  $G_1 = \frac{\mu C_{ox} W_1}{L} \left[ (V_1 - V_{th}) + W_2 (V_R - V_{th}) \right]$

  $G_2 = \frac{\mu C_{ox} W_2}{L} \left[ (V_2 - V_{th}) + W_1 (V_R - V_{th}) \right]$

  $\Delta G = \frac{\mu C_{ox} W_i}{L} \left[ (V_1 - V_{th}) - W_2 (V_R - V_{th}) \right]$

- To 1st order, for $W_1 = W_2$ & $W_11 = W_12$
  
  $V_{th, out} = -W_11/W_1 x V_x$

  where $V_x = V_{R+} - V_{R-}$

  $V_x$ fixed $W_1$, $I_2$ varied from comparator to comparator

  Eliminates need for resistive divider (DAC)

Comparator Example (Pipelined ADC)

- Used in a pipelined ADC with digital correction
  → No offset cancellation required
- Differential reference & input
- M7, M8 operate in triode region
- Preamp gain ~10
- Input buffers suppress kick-back
- $\phi_1$ high → $C_c$ charged to $V_R$ & $\phi_2$ is also high → Current diverted to latch → comparator output in hold mode
- $\phi_2$ high → $C_c$ connected to S/Hout & comparator input (VR-S/Hout), current sent to preamp → comparator in amplify mode


Reducing Flash ADC Complexity

E.g. 10-bit “straight” flash
- Input range: 0 … 1V
- LSB = $\Delta_0$ ~ 1mV
- Comparators: 1023 with offset < 1/2 LSB
- Input capacitance: $1023 \times 100fF = 102pF$
- Power: $1023 \times 3mW = 3W$
  → High power dissipation & large area & high input cap.

Techniques to reduce complexity & power dissipation:
- Interpolation
- Folding
- Folding & Interpolation
- Two-step, pipelining
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
    \(\rightarrow\) **Improved DNL**

---

**Flash ADC**

- Preamp Output

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

  \[
  \begin{align*}
  V_{ref1} &= 1 \Delta \\
  V_{ref2} &= 2 \Delta 
  \end{align*}
  \]
Simulink Model

Differential Preamp Output

Differential output crossings
@ $V_{in}$ =

$V_{ref1} = 1 \Delta$

$V_{ref2} = 2 \Delta$

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

$A_1$ & $-A_2$ $\Rightarrow$ $A_1$ & $-A_2$ = $A_1$ + $A_2$

$\Rightarrow$ cross zero at:

$V_{ref2} = 0.5 \times (1+2) \Delta = 1.5 \Delta$
Interpolation in Flash ADC

Half as many reference voltages and preamps
Interpolation factor: x2

Example: For 10bit straight Flash ADC need $2^B=1024$ preamps compared $2^{B-1}=512$ for x2 interpolation

Possible to accomplish higher interpolation factor
→ Interpolation at the output of preamps

Compare A2& -A1
→ Comparator output is sign of A1+A2

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)

Higher Order Resistive Interpolation

- Resistors produce additional levels.
- With 4 resistors per side, the "interpolation factor" $M=8$ → extra 3 bits.
- $(M \rightarrow$ ratio of latches/preamps).


Preamp Output Interpolation

DNL Improvement

- Preamp offset distributed over $M$ resistively interpolated voltages:
  → Impact on DNL divided by $M$.
- Latch offset divided by gain of preamp.
  → Use "large" preamp gain.
  → Next: Investigate how large preamp gain can be?

Preamp Input Range

If linear region of preamp transfer curve do not overlap

→ Dead-zone in the interpolated transfer curve!
Results in error

→ Consecutive preamp linear input ranges must overlap
i.e. input range > Δ

Sets upper bound on preamp gain: \( \text{Preamp}_{\text{gain}} \leq \frac{V_{\text{DD}}}{\Delta} \)

Interpolated-Parallel ADC

• 10-bit overall resolution:
• 7-bit flash (127 preamps and 128 resistors) & x8 interpolation
• Use of Gray Encoder minimizes effect of sparkle code & metastability

Measured Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</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>56/-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 x 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 $\rightarrow$ 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 $\rightarrow$ Use folding to reduce the # of latches
Folding Converter

- Two ADCs operating in parallel
  - MSB ADC
  - Folder + LSB ADC
- Significantly fewer comparators compared to 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
Example: Folding Factor of 4

• How are folds generated?

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

• Note: Sign change every other fold + reference shift

Generating Folds via Source-Coupled Pairs

Vref1 < Vref2 < Vref3 < Vref4
As Vin changes, only one of M1, M3, M5, M7 is on depending on the input level
Let us try Vref1=1.2\Delta_{\text{MSB}} Vref2=1.5\Delta_{\text{MSB}} Vref3=2.5\Delta_{\text{MSB}} Vref4=3.5\Delta_{\text{MSB}}
CMOS Folder Output

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

In fact, most folding ADCs do not use the folds, but only the midpoint-crossings!

Parallel Folders Using Only Midpoint-Crossings

- Folder 4
  - $V_{in} + \frac{3}{4} \Delta$
  - Comparator

- Folder 3
  - $V_{in} + \frac{1}{4} \Delta$
  - Comparator

- Folder 2
  - $V_{in} + \frac{1}{4} \Delta$
  - Comparator

- Folder 1
  - $V_{in} + \frac{1}{4} \Delta$
  - Comparator

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
$V_{\text{ref}} + 3/4 \cdot \Delta$

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

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

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

Fine Flash ADC

Encoder
Folder / Interpolator Output

Example: 4 Folders + 4 Resistive Interpolator per Stage

Note: Output of two folders only + 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

Note:
Total of 40 (MSB=8, LSB=32) comparators compared to $2^8-1=255$ for straight flash
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

**Two-Step Example: (2+2)Bits**

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

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

$$V_{in} \leftarrow \text{"Coarse"} \rightarrow \text{2-bit ADC} \rightarrow \varepsilon_{q1} \leftarrow \text{"Fine"} \rightarrow \text{2-bit DAC}$$

$$D_{out} = V_{in} + \varepsilon_{q1} - \varepsilon_{q1} + \varepsilon_{q2}$$

Two Step (2+2) Flash ADC

- 4-bit Straight Flash ADC
- Ideal 2-step Flash ADC

$$V_{in} \rightarrow$$
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
  \[ \varepsilon_{q^2} = \frac{V_{ref^2}}{2^2} = \frac{V_{ref^1}}{2^2 \cdot 2^2} \]

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 $\Rightarrow$ Need at least 1 clock cycle per extra series stage to resolve one sample

Solution to Issue (1)
Reducing Precision Required for Fine ADC

• Accuracy needed for fine ADC relaxed by introducing inter-stage gain
  - Example: By adding gain of $x(G=2^{81}=4)$ 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)  
Increasing ADC Throughput

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

\[
D_{out} = V_{in} + \varepsilon_{q1} - \varepsilon_{q1} + \varepsilon_{q2}
\]

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., 10-bit 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

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

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
- One important feature of pipeline ADC: 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
$(B_1 + B_2 + ... + B_k)$ Bits

**Pipeline ADC Latency**

Note: One conversion per clock cycle & 8 clock cycle latency

[Analog Devices, AD 9226 Data Sheet]
Pipeline ADC
Digital 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), power dissipation

Complete Pipeline Stage

- "Residue Plot"
  - E.g.: $B=2$
  - $G=2^2 = 4$
Pipeline ADC Single Stage Model

\[ V_{\text{res}} = Gx\varepsilon_q \]

Pipeline ADC Errors

- Non-idealities associated with sub-ADCs, sub-DACs and gain stages \( \rightarrow \) error in overall pipeline ADC performance

- Need to find means to tolerate/correct errors

- Important sources of error
  - Sub-ADC errors- comparator offset
  - Gain stage offset
  - Gain stage gain error
  - Sub-DAC error
Pipeline ADC Multi-Stage Model

\[
D_{\text{out}} = V_{\text{in,ADC}} + \sum_{j=1}^{n} \frac{\epsilon_{qj}}{G_{j}} + \sum_{j=2}^{n} \left( \prod_{j'=1}^{j-1} \frac{G_{j'}}{G_{j-1}} \right) \left( I - \frac{G_{j}}{G_{j-1}} \right) + \prod_{j=2}^{n} \frac{G_{j}}{G_{j-1}} \epsilon_{q(n-1)}
\]

Pipeline ADC Model

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

\[
D_{\text{out}} = V_{\text{in,ADC}} + \frac{\epsilon_{qn}}{\prod_{j=1}^{n} G_{j}}
\]

\[
D.R. = 20 \log_{10} \frac{\text{rms FS Signal}}{\text{rms Quant. Noise}} = 20 \log_{10} \frac{V_{\text{ref}}}{\sqrt{2} \sqrt{2}} = 20 \log_{10} \left( \frac{\sqrt{2} \times 2^{B_{n}} \prod_{j=1}^{n-1} G_{j}}{\sqrt{2} \times 2^{B_{n}} \prod_{j=1}^{n-1} G_{j}} \right)
\]

\[
B_{ADC} = \log_{2} \left( 2^{B_{n}} \times \prod_{j=1}^{n-1} G_{j} \right)
\]

\[
B_{ADC} = B_{n} + \log_{2} \prod_{j=1}^{n-1} G_{j}
\]
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