Lecture 20

Analog-to-Digital Converters (continued)

- Comparator design (continued)
  - Comparator architecture examples

- Techniques to reduce flash ADC complexity
  - Interpolating
  - Folding
  - Interpolating & folding

- Residue Type ADCs
  - Two-Step flash
  - Pipelined ADCs
    - Architecture basics
    - Effect of sub-ADC, sub-DAC, gain stage non-idealities on overall ADC performance

CMOS Comparator Example

Flash ADC

- Flash ADC: 8bits, +1/2LSB INL @ fs=15MHz (Vref=3.8V, LSB~15mV)
- No offset cancellation

Comparator with Auto-Zero


Flash ADC
Comparator with Auto-Zero


**Flash ADC**

**Comparator with Auto-Zero**

\[ V_o = A_{p1} \cdot A_{p2} \cdot \left( V_{in+} - V_{in-} \right) - V_{offset} \]

Substituting for \( V_{offset} \) from previous cycle:

\[ V_o = A_{p1} \cdot A_{p2} \cdot \left( V_{in+} - V_{in-} - (V_{ref+} - V_{ref-}) \right) \]

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


**Flash ADC**

**Using Comparator with Auto-Zero**

Auto-Zero Implementation


Comparator Example

- Variation on Yukawa latch used w/o preamp
- Good for low resolution ADCs (in this case 1.5bit/stage for a pipeline we will see later are tolerant of high offset)
- 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:
  \[ G_1 = \frac{\mu C_{ox}}{L} \left[ W_I (V_{th1} - V_{thh}) + W_{th1} (V_R - V_{thh}) \right] \]
  \[ G_2 = \frac{\mu C_{ox}}{L} \left[ W_I (V_{th2} - V_{thh}) + W_{th2} (V_R - V_{thh}) \right] \]
  \[ \Delta G = \frac{\mu C_{ox} W_I}{L} \left[ (V_{th1} - V_{th2}) - \frac{W_{th1}}{W_I} (V_R - V_{thh}) \right] \]
- To 1st order, for \( W_I = W_2 \) & \( W_I = W_{12} \)
  \[ V_{thh} = \frac{W_{th1}}{W_I} \times V_R \]
  where \( V_R = V_{thh} - V_R \).
  \[ \Rightarrow \] \( V_R \) fixed \( W_1, \) \( W_2 \) varied from comparator to comparator \( \Rightarrow \) Eliminates need for resistive divider


Comparator Example

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

Bipolar Comparator Example

- Used in 8-bit 400Ms/s & 6-bit 2Gb/s flash ADC
- Signal amplification during $\phi_1$ high, latch operates when $\phi_1$ low
- Input buffers suppress kick-back & input current
- Separate ground and supply buses for front-end preamp $\rightarrow$ kick-back noise reduction


Reducing Flash ADC Complexity

E.g. 10-bit "straight" flash
- Input range: 0 ... 1V
- $\text{LSB} = \Delta \approx 1\text{mV}$
- Comparators: 1023 with offset < 1/2 LSB
- Assuming $C_{in}$ for each comparator is 0.1pF & power 3mW
  - Total input capacitance: $1023 \times 0.1\text{pF} = 102\text{pF}$
  - Power: $1023 \times 3\text{mW} = 3\text{W}$
  $\rightarrow$ 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
  → Improved DNL

---

Flash ADC

Preamp Output

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

$V_{ref1} = 1\Delta$

$V_{ref2} = 2\Delta$
Simulink Model

\[ \text{Vin} \rightarrow \text{Preamp1} \rightarrow \text{Vref1} \frac{\Delta}{2} \rightarrow \text{Preamp2} \rightarrow \text{A1} \rightarrow \text{A2} \rightarrow \text{Vin} \]

Differential Preamp Output

Differential output crossings
\(@ V_{in} = \)
\[ V_{ref1} = 1 \Delta \]
\[ V_{ref2} = 2 \Delta \]

Note: Additional crossing of 
\[ A_1 A_2 (A_2 A_1) \]
\[ A_1 (A_2) = A_1 + A_2 \]
\[ \text{cross zero at:} \]
\[ V_{ref12} = 0.5(1+2) \Delta = 1.5 \Delta \]
Interpolation in Flash ADC

Half as many reference voltages and preamps
Interpolation factor: $x2$

Example: For 10bit 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 $A2$ - $A1$
$\rightarrow$ 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)

Interpolation in Flash ADC
Preamp Output Interpolation

Vin

A2
A2
X
A1
A1

With 2 sets of interpolation resistors at each preamp outputs \(\rightarrow\) three extra intermediate points \(\rightarrow\) 2 extra bits

Higher Order Resistive Interpolation

- Resistors produce additional levels
- With 4 resistors per side, the “interpolation factor” \(M=8\) \(\rightarrow\) 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
- Linear consecutive preamp input ranges must overlap
  - i.e. input range w/o output saturation > \Delta

Sets upper bound on preamp gain: \text{Preamp}_{\text{gain}} < \frac{V_{\text{DD}}}{\Delta}
Interpolated-Parallel ADC

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


Measured Performance

<table>
<thead>
<tr>
<th>Specification</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</td>
<td>56/59 dB 48/47 dB</td>
</tr>
<tr>
<td>10MHz input</td>
<td>56/59 dB</td>
</tr>
<tr>
<td>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 linear region need to have overlap \( \Rightarrow \) Limits gain of preamp to \( \sim V_{DD}/\Delta \)

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

• DNL due to preamp offset reduced 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
• Medium fast
• Typically, nonidealities in folder limit resolution
Example: Folding Factor of 4

- Folding factor: number of folds \(2^{BM_{MSB}}\)
- 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

\[
\begin{align*}
V_{in} & \rightarrow V_{out} = \frac{V_{FS}}{2} \\
V_{in} & \rightarrow V_{out} = -V_{in} + \frac{V_{FS}}{2} \\
V_{in} & \rightarrow V_{out} = +V_{in} - \frac{V_{FS}}{2} \\
V_{in} & \rightarrow V_{out} = -V_{in} + V_{FS}
\end{align*}
\]

- 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.

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 zero-crossings!
Parallel Folders Using Only Zero-Crossings

\[ V_{\text{in}} \]

\begin{align*}
\text{Folder 4} & \quad \text{Comparator} \\
V_{\text{ref}} + \frac{3}{4} \times \Delta \\
\text{Folder 3} & \quad \text{Comparator} \\
V_{\text{ref}} + \frac{2}{4} \times \Delta \\
\text{Folder 2} & \quad \text{Comparator} \\
V_{\text{ref}} + \frac{1}{4} \times \Delta \\
\text{Folder 1} & \quad \text{Comparator} \\
V_{\text{ref}} + \frac{0}{4} \times \Delta
\end{align*}

\text{Logic} \quad \text{LSB bits} \quad \text{(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
  - Better solution: Combine with 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 only + corresponding interpolator only shown
Folder / Interpolator Output
Example: 2 Folders + 8 Resistive Interpolator per Stage

Zero-crossing not equally spaced $\Rightarrow$ Non-linear distortion
$\Rightarrow$ 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 (MSB=8, LSB=32) comparators compared to $2^8 - 1 = 255$ for straight flash

Ref: B. Nauta and G. Venes, JSSC Dec 1985, pp. 1302-8
ADC Architectures

- Slope type converters
- Successive approximation
- Flash
- Interpolating & Folding
- Residue type ADCs
  - Two-step Flash
  - Pipelined ADCs
  - ...
- Time-interleaved / parallel converter
- Oversampled ADCs

Two-Step Example: (2+2)Bits

- Using only one ADC: output contains large quantization error
- "Missing voltage" or "residue" (ε_q1)
- Idea: Use second ADC to quantize and add -ε_q1
Two Stage Example

- Use DAC to compute missing voltage
- Add quantized representation of missing voltage
- Why does this help? How about $e_{q2}$?
- Since maximum voltage at input of the 2nd ADC is $V_{ref1}/4$ then for 2nd ADC $V_{ref2} = V_{ref1}/4$ and thus $e_{q2} = e_{q1}/4 = V_{ref1}/16 \rightarrow 4$-bit overall resolution

Two Step (2+2) Flash ADC

- 4-bit Straight Flash ADC
- Ideal 2-step Flash ADC
- Voltage quantized by 2nd 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

\[ \varepsilon_q^{2} = \frac{V_{\text{ref}2}}{2^2} = \frac{V_{\text{ref}1}}{2^2 \cdot 2^2} \]

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 1/2LSB
  2. Speed penalty → 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 $x(G=2^{B1}=4)$ 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)
Increasing ADC Throughput

• 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 stage samples/converts residue associated with sample # n-1

Residue Type ADCs

• Two-Step flash
• Pipelined ADCs
  – Basic operation
  – Effect of sub-ADC, sub-DAC, gain stage non-idealities on overall ADC performance
    • Error correction by adding redundancy
    • Digital calibration
    • Correction for inter-stage gain nonlinearity
  – Implementation
    • Practical circuits
    • Stage scaling
    • Combining the bits
    • Stage implementation
      – Circuits
      – Noise budgeting
    • How many bits per stage?
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

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
Pipeline ADC Characteristics

- Number of components (stages) grows linearly with resolution

- Pipelining
  - Trading latency for overall component count
  - Latency may be an issue in e.g. control systems
  - Throughput limited by speed of one stage → Fast

- Versatile: 8...16bits, 1...400MS/s

- One important feature of pipeline ADC: many analog circuit non-idealities can be corrected digitally
Pipeline ADC
Digital Data Alignment

- Digital shift register aligns sub-conversion results in time

Cascading More Stages

- LSB of last stage becomes very small
- All stages need to have full precision
- Impractical to generate several $V_{ref}$
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.:
- $B = 2$
- $G = 2^2 = 4$

Note: None of the blocks have ideal performance

Question: What is the effect of the non-idealities?
Pipeline ADC Errors

- Non-idealities associated with sub-ADCs, sub-DACs and gain stages → 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 Single Stage Model

\[ V_{\text{in}} - \text{Dout} - G \times e_q = \sum e_q \]

\[ V_{\text{res}} = G \times e_q \]
Pipeline ADC Multi-Stage Model

\[
D_{out} = V_{in,ADC} + \varepsilon_q + \frac{G_1}{G_d} \left( I - \frac{G_2}{G_d} \right) + \frac{G_2}{G_d} \left( I - \frac{G_3}{G_d} \right) + \ldots + \frac{G_{n-1}}{G_d} \left( I - \frac{G_n}{G_d} \right) + \varepsilon_q \frac{G_n}{G_d} \]

Pipeline ADC Model

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

\[
D_{out} = V_{in,ADC} + \varepsilon_q \frac{G_{ref}}{2^Bn} \quad \text{where} \quad \varepsilon_q = \frac{V_{ref}}{2^Bn} \quad \text{and} \quad Bn = \# \text{ of bits in final stage}
\]

\[
D.R. = 20 \log \frac{\text{rms FS Signal}}{\text{rms Quant. Noise}} = 20 \log \frac{V_{ref}}{2\sqrt{2} \sqrt{\frac{\text{rms Quant. Noise}}{V_{ref}}}} = 20 \log \left( \sqrt{\frac{3}{2}} \times 2^{Bn} \times \prod_{j=1}^{Bn} G_j \right)
\]

\[
B_{ADC} = \log_2 \left( 2^n \times \prod_{j=1}^{Bn} G_j \right)
\]

\[
B_{ADC} = B_n + \log_2 \prod_{j=1}^{Bn} G_j
\]
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 $D_{\text{out}}$ 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
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:
$-1, 0.3, +1$

Pipeline ADC
Three Ways to Deal with Sub-ADC Errors

- All involve “sub-ADC redundancy”
- Redundancy in stage that produces errors
  - Choose gain for residue to be processed by the 2nd stage < $2^{B1}$
  - Higher resolution sub-ADC & sub-DAC
- Redundancy in succeeding stage(s)
(1) Inter-Stage Gain Following 1\textsuperscript{st} Stage <2\textsuperscript{B1}

- Choose \( G_1 \) less than 2\textsuperscript{B1}
- Effective stage resolution could become non-integer
  \( B_{1\text{eff}} = \log_2 G_1 \)
- E.g. if \( G_1 = 3.8 \rightarrow B_{1\text{eff}} = 1.8\text{bit} \)

Correction Through Redundancy

"enlarged" residuum still within input range of next stage

If \( G_1 = 2 \) instead of 4
→ Only 1-bit resolution from first stage (3-bit total)
→ In spite of comparator offset: No overall error
(2) Higher Resolution Sub-ADC

- Keep $G_1$ precise power of two (e.g. $G_1=4$)
- Add extra decision levels in sub-ADC (e.g. add 1 extra bit to 1st stage)
- E.g. $B_1 = B_{1\text{eff}} + 1$

-Ref: Singer et. al., VSLI 1996

(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