Pipelined ADCs (continued)
– Effect gain stage, sub-DAC non-idealities on overall ADC performance
  • Digital calibration (continued)
  • Correction for inter-stage gain nonlinearity

– Implementation
  • Practical circuits
  • Stage scaling
  • Combining the bits
  • Stage implementation
    – Circuits
    – Noise budgeting
  • How many bits per stage?
– Algorithmic ADCs utilizing pipeline structure
– Advanced background calibration techniques

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"
**Summary So Far**

**Pipelined A/D Converters**

- Cascade of low resolution stages
  - By adding inter-stage gain $= 2^{B_{\text{eff}}}$
    - No need to scale down $V_{\text{ref}}$ for stages down the pipe
    - Reduced accuracy requirement for stages coming after stage 1
  - Addition of Track & Hold function to interstage-gain →
    - Stages can operate concurrently →
    - Throughput increased to as high as one sample per clock cycle
    - Latency function of number of stages & conversion-per-stage
  - Correction for circuit non-idealities
    - Built-in redundancy compensate for sub-ADC inaccuracies such as comparator offset (interstage gain: $G=2^{B_{\text{neff}}}$, $B_{\text{neff}} < B_n$)

---

**Pipeline ADC Error Compensation**

- 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 error
  - Sub-DAC error
Gain Stage Gain Inaccuracy

- Gain error can be compensated in digital domain – "Digital Calibration"
- Problem: Need to measure/calibrate digital correction coefficient
- Example: Calibrate 1-bit first stage
- Objective: Measure G in digital domain

\[
V_{\text{DAC}}(D = 0) = 0 \\
V_{\text{DAC}}(D = 1) = \frac{V_{\text{ref}}}{2}
\]

\[
V_{\text{res1}} = G \cdot (V_{\text{in}} - \frac{V_{\text{ref}}}{2})
\]
Gain Stage Inacurracy Calibration – Step 1

\[
V_{\text{res}1}^{(1)} = G \left( V_{\text{in}} - \frac{V_{\text{ref}}}{2} \right)
\]

\[
D_{\text{back}}^{(1)} = G \cdot \frac{\left( V_{\text{in}} - \frac{V_{\text{ref}}}{2} \right)}{V_{\text{ref}}} \rightarrow \text{store}
\]

Gain Stage Inacurracy Calibration – Step 2

\[
V_{\text{res}1}^{(2)} = G \cdot \left( V_{\text{in}} - 0 \right)
\]

\[
D_{\text{back}}^{(2)} = G \cdot \frac{\left( V_{\text{in}} - 0 \right)}{V_{\text{ref}}} \rightarrow \text{store}
\]
Gain Stage Inacurracy  
Calibration – Evaluate

\[ D_{\text{back}}^{(1)} = G \cdot \frac{(V_{\text{in}} - V_{\text{ref}} / 2)}{V_{\text{ref}}} \]

\[ -D_{\text{back}}^{(2)} = G \cdot \frac{(V_{\text{in}} - 0)}{V_{\text{ref}}} \]

\[ D_{\text{back}}^{(1)} - D_{\text{back}}^{(2)} = \frac{1}{2} G \]

- To minimize the effect of backend ADC noise → perform measurement several times and take the average

Accuracy Bootstrapping

\[ D_{\text{out}} = V_{\text{in,ADC}} + \epsilon_q \left( 1 - \frac{G_1}{G_{d1}} \right) + \epsilon_q G_2 \left( 1 - \frac{G_2}{G_{d2}} \right) + \cdots + \epsilon_q G_{n-1} \left( 1 - \frac{G_{n-1}}{G_{d(n-1)}} \right) + \epsilon_q G_n \]

- Highest sensitivity to gain errors in front-end stages
"Accuracy Bootstrapping"

![Diagram of pipeline ADC with stages and calibration direction]

Ref:
L. Singer et al., "A 12 b 65 MSample/s CMOS ADC with 82 dB SFDR at 120 MHz," ISSCC 2000, Digest of Tech. Papers., pp. 38-9 (calibration in opposite direction!)

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 error
  - Sub-DAC error
DAC Errors

- Can be corrected digitally as well
- Same calibration concept as gain errors → Vary DAC codes & measure errors via backend ADC

DAC Calibration – Step 1

- $\varepsilon_{DAC}(0)$ equivalent to offset - ignore
DAC Calibration – Step 2...2^{B_1}

- Stepping through DAC codes 1...2^{B_1}-1 yields all incremental correction values
- Measurements repeated and averages to account for variance associated with noise

Pipeline ADC
Example: Calibration Hardware

- Above block diagram may seem extensive however, in current fine-line CMOS technologies digital portion of a pipeline ADCs consume insignificant power and area compared to the analog sections
Pipelined ADC
Error Correction/Calibration Summary

<table>
<thead>
<tr>
<th>Error</th>
<th>Correction/Calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_{ADC}$, $V_{os}$</td>
<td>Redundancy either same stage or next stage</td>
</tr>
<tr>
<td>$\epsilon_{gain}$</td>
<td>Digital adjustment</td>
</tr>
<tr>
<td>$\epsilon_{DAC}$</td>
<td>Either sufficient component matching or digital calibration</td>
</tr>
<tr>
<td>Inter-stage amplifier non-linearity</td>
<td>?</td>
</tr>
</tbody>
</table>

Inter-stage Gain Nonlinearity

- Invert gain stage non-linear polynomial
- Express error as function of $V_{RES1}$
- Push error into digital domain through backend

Inter-stage Gain Nonlinearity

\[ V_{R3} = a_3 V_X^3 \]

\[ p_2 = \frac{a_3}{(2^3 + \varepsilon_{gain})^3} \]

\[ \varepsilon(D_B, p_2) = p_2 D_B^3 - 3p_2^2 D_B^5 + 12p_2^3 D_B^7 - \ldots \]

- Pre-computed table look-up
- \( p_2 \) continuously estimated & updated (account for temp. & other variations)


Inter-stage Gain Nonlinearity Compensation

Proof of Concept Evaluation Prototype

- Re-used 14-bit ADC in 0.35μm from Analog Devices [Kelly, ISSCC 2001]
- Modified only 1st stage with 3-bit open-loop amplifier built with simple diff-pair + resistive load instead of the conventional feedback around high-gain amp
- Conventional 9-bit backend, 2-bit redundancy in 1st stage
- Real-time post-processor off-chip (FPGA)

Measurement Results
12-bit ADC w Extra 2-bits for Calibration

(a) without calibration

(b) with calibration

InL [LSB]

Code

0
1000
2000
3000
4000

-1
-0.5
0
0.5
1

Combining the Bits

• Example: Three 2-bit stages, no redundancy

\[ D_{out} = D_1 + \frac{1}{2} B_{1e} D_2 + \frac{1}{2} B_{2e} D_3 \]

\[ D_{out} = D_1 + \frac{1}{4} D_2 + \frac{1}{16} D_3 \]
Combining the Bits

- Only bit shifts
- No arithmetic circuits needed

\[
\begin{array}{c|c}
D_1 & XX \\
D_2 & XX \\
D_3 & XX \\
\hline
D_{out} & DDDDDD \\
\end{array}
\]

Stage 1

Stage 2

Stage 3

\[D_{out}[5:0]\]

Combining the Bits
Including Redundancy

- Example: Three 2-bit stages, incorporating 1-bit redundancy in stages 1 and 2

\[
\begin{array}{c|c}
B_1 &= 2 \\
B_{1\text{eff}} &= 2 \\
B_2 &= 2 \\
B_{2\text{eff}} &= 2 \\
B_3 &= 2 \\
\end{array}
\]

\[D_{out}[5:0]\]
Combining the Bits

\[ D_{\text{out}} = D_1 + \frac{1}{2^{B_1 \text{eff}}} D_2 + \frac{1}{2^{B_2 \text{eff}}} D_3 \]
\[ D_{\text{out}} = D_1 + \frac{1}{4} D_2 + \frac{1}{16} D_3 \]

- Bits overlap
- Need adders

<table>
<thead>
<tr>
<th>(D_1)</th>
<th>(D_2)</th>
<th>(D_3)</th>
<th>(D_{\text{out}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX</td>
<td>XXX</td>
<td>XX</td>
<td>DDDDDDD</td>
</tr>
</tbody>
</table>

Combining the Bits

Example

\[ D_1 = 001 \]
\[ D_2 = 111 \]
\[ D_3 = 10 \]
\[ D_{\text{out}} = 011000 \]
Stage Implementation

- Each stage needs T/H hold function
- Track phase: Acquire input/residue from previous stage
- Hold phase: sub-ADC decision, compute residue

Usually no dedicated T/H amplifier in each stage (Except first stage in some cases – why?)

T/H implicitly contained in stage building blocks
Stage Implementation

- DAC-subtract-gain function can be lumped into a single switched capacitor circuit
- "MDAC"

1.5-Bit Stage Implementation Example

1.5-Bit Stage Implementation

Acquisition Cycle


Conversion Cycle

1.5 Bit Stage Implementation Example

\[ V_o = \begin{cases} 
(1 + \frac{C_i}{C_f}) V_i - \frac{C_i}{C_f} V_{ref} & \text{if } V_i > V_{ref}/4 \\
(1 + \frac{C_i}{C_f}) V_i & \text{if } -V_{ref}/4 \leq V_i \leq +V_{ref}/4 \\
(1 + \frac{C_i}{C_f}) V_i + \frac{C_i}{C_f} V_{ref} & \text{if } V_i < -V_{ref}/4 
\end{cases} \]

Note: Interstage gain set by C ratios
- Accuracy better than 0.1%
- Up to 10bit level no need for gain calibration


1.5-Bit Stage Implementation
Timing of Stages

Pipelined ADC Stage Power Dissipation & Noise

- Typically pipeline ADC noise dominated by inter-stage gain blocks
- Sub-ADC comparator noise translates into comparator threshold uncertainty and is compensated for by redundancy

\[ V_{\text{noise}}^{\text{in}} = \sqrt{\frac{V_n^2}{G1^2} + \frac{V_{n2}^2}{G1^2G2^2} + \ldots} \]

---

Pipelined ADC Stage Scaling

- Example: Pipeline using 1-bit_ref stages

\[ N_{101} \propto kT \left[ \frac{1}{C_1} + \frac{1}{G1^2C_2} + \frac{1}{G1^2G2^2C_3} + \ldots \right] \]

\[ N_{101} \propto kT \left[ \frac{1}{C_1} + \frac{1}{4C_2} + \frac{1}{16C_3} + \ldots \right] \]
If all caps made the same size, backend stages contribute very little noise.

Wasteful power-wise, because:

- Power \( \sim Gm \)
- Speed \( \sim Gm/C \)

Fixed speed \( \rightarrow Gm/C \) fixed \( \rightarrow \) Power \( \sim C \)

How about scaling caps down by \( G^2 = 2^2 = 4 \)x per stage?

- Same amount of noise from every stage
- All stages contribute significant noise
- To keep overall noise the same \( \rightarrow \) noise/stage must be reduced
- Power \( \sim Gm \) \( \sim C \) goes up!
Stage Scaling
Example: 2-bit_{eff}/stage

- Optimum capacitior scaling lies approximately midway between these two extremes

Ref: D. W. Cline, P.R. Gray "A power optimized 13-b 5MSamples/s pipelined analog-to-digital converter in 1.2um CMOS," JSSC 3/1996

Pipeline ADC
Stage Scaling

- Power minimum is "shallow"
- Near optimum solution in practice: Scale capacitors by stage gain
- E.g. for effective stage resolution of 1bit (Gain=2):

\[
\begin{align*}
V_{in} &\rightarrow \text{Gm} \rightarrow \frac{C}{2} \rightarrow \text{Gm} \rightarrow \frac{C}{4} \rightarrow \text{Gm} \rightarrow \frac{C}{8} \\
\end{align*}
\]
Stage Scaling Example

Ref: D. W. Cline, P. R. Gray "A power optimized 13-b 5 MSamples/s pipelined analog-to-digital converter in 1.2um CMOS," JSSC 3/1996

Note:
Resolution per stage:
→ 2 bits
→ A=4

How Many Bits Per Stage?

• Many possible architectures
  – E.g. B_{1eff}=3, B_{2eff}=1, ...
  vs. B_{1eff}=1, B_{2eff}=1, B_{3eff}=1, ...

• Complex optimization problem, fortunately optimum tends to be shallow...

• Qualitative answer:
  – Maximum speed for given technology
    • Use small resolution-per-stage (large feedback factor)
  – Maximum power efficiency for fixed, "low" speed
    • Try higher resolution stages
    • Can help alleviate matching & noise requirements in stages following the 1st stage
  Ref: Singer VLSI 96, Yang, JSSC 12/01
### 14 & 12-Bit State-of-the-Art Implementations

<table>
<thead>
<tr>
<th>Reference</th>
<th>Yang (JSSC 12/2001) 0.35μ/3V</th>
<th>Loloee (ESSIRC 2002) 0.18μ/3V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>Architecture</td>
<td>3-1-1-1-1-1-1-1-1-1-1-3</td>
<td>1-1-1-1-1-1-1-1-1-1-1-2</td>
</tr>
<tr>
<td>SNR/SFDR</td>
<td>~73dB/88dB</td>
<td>~66dB/75dB</td>
</tr>
<tr>
<td>Speed</td>
<td>75MS/s</td>
<td>80MS/s</td>
</tr>
<tr>
<td>Power</td>
<td>340mW</td>
<td>260mW</td>
</tr>
</tbody>
</table>

### 10 & 8-Bit State-of-the-Art Implementations

<table>
<thead>
<tr>
<th>Reference</th>
<th>Yoshioko et al (ISSCC 2005) 0.18μ/1.8V</th>
<th>Kim et al (ISSCC 2005) 0.18μ/1.8V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Architecture</td>
<td>1.5bit/stage</td>
<td>2.8 - 2.8 - 4</td>
</tr>
<tr>
<td>SNR/SFDR</td>
<td>~55dB/66dB</td>
<td>~48dB/56dB</td>
</tr>
<tr>
<td>Speed</td>
<td>125MS/s</td>
<td>200MS/s</td>
</tr>
<tr>
<td>Power</td>
<td>40mW</td>
<td>30mW</td>
</tr>
</tbody>
</table>
Algorithmic ADC

- Essentially same as pipeline, but a single stage is reused for all partial conversions
- For overall \(B_{\text{overall}}\) bits \(\Rightarrow\) need \(B_{\text{overall}}/B_{\text{stage}}\) clock cycles per conversion
  \(\Rightarrow\) Small area, slow

---

Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters

- Slow, but accurate ADC operates in parallel with pipelined (main) ADC
- Slow ADC samples input signal at a lower sampling rate \((f_s/n)\)
- Difference between corresponding samples for two ADCs \((e)\) used to correct fast ADC digital output via an adaptive digital filter (ADF) based on minimizing the Least-Mean-Squared error

Example: "A 12-bit 20-MS/s pipelined analog-to-digital converter with nested digital background calibration"

- Pipelined ADC operates at 20Ms/s @ has 1.5bit/stage
- Slow ADC → Algorithmic type operating at 20Ms/32=625ks/s
- Digital correction accounts for bit redundancy
- Digital error estimator → minimizes the mean-squared-error


Algorithmic ADC Used for Calibration of Pipelined ADC (continued from previous page)

- Uses replica of pipelined ADC stage
- Requires extra SHA in front to hold residue
- Undergoes a calibration cycle periodically prior to being used to calibrate pipelined ADC

12-bit 20-MS/s Pipelined ADC with Digital Background Calibration

- Sampling capacitors scaled:
  - Input SHA: 6pF
  - Pipelined ADC: 2pF, 0.9, 0.4, 0.2, 0.1, 0.1...
  - Algorithmic ADC: 0.2pF

- Chip area: 13.2mm²
  - Area of Algorithmic ADC <20%
  - Does not include digital calibration circuitry estimated ~1.7mm²


Measurement Results

12-bit 20-MS/s Pipelined ADC with Digital Background Calibration

<table>
<thead>
<tr>
<th>Without Calibration</th>
<th>With Calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INL</td>
</tr>
</tbody>
</table>

Measurement Results
12-bit 20-MS/s Pipelined ADC with Digital Background Calibration


Performance Summary (3.3 V, 25 °C)

<table>
<thead>
<tr>
<th>Process</th>
<th>0.35μm 2P1M CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling rate</td>
<td>20 Msample/s</td>
</tr>
<tr>
<td>Active area</td>
<td>7.5 mm²</td>
</tr>
<tr>
<td>Full Scale Input</td>
<td>1.6 Vpp</td>
</tr>
<tr>
<td></td>
<td>Without Cal.</td>
</tr>
<tr>
<td>Analog Power Diss.</td>
<td>190 mW</td>
</tr>
<tr>
<td></td>
<td>With Cal.</td>
</tr>
<tr>
<td>Total Power Diss.</td>
<td>217 mW</td>
</tr>
<tr>
<td></td>
<td>254 mW</td>
</tr>
<tr>
<td>Max. INL (Pip. ADC)*</td>
<td>4.21 LSB</td>
</tr>
<tr>
<td></td>
<td>0.47 LSB</td>
</tr>
<tr>
<td>Max. DNL (Pip. ADC)*</td>
<td>0.60 LSB</td>
</tr>
<tr>
<td></td>
<td>0.41 LSB</td>
</tr>
<tr>
<td>SVD (Alg. ADC)*</td>
<td>49.6 dB</td>
</tr>
<tr>
<td></td>
<td>59.6 dB</td>
</tr>
<tr>
<td>SFDR (Pip. ADC)*</td>
<td>59.2 dB</td>
</tr>
<tr>
<td></td>
<td>70.8 dB</td>
</tr>
<tr>
<td>THD (Pip. ADC)</td>
<td>59.4 dB</td>
</tr>
<tr>
<td></td>
<td>92.9 dB</td>
</tr>
<tr>
<td>PSRR*</td>
<td>66.0 dB</td>
</tr>
<tr>
<td></td>
<td>64.8 dB</td>
</tr>
<tr>
<td>CMRR*</td>
<td>73.6 dB</td>
</tr>
<tr>
<td></td>
<td>75.4 dB</td>
</tr>
</tbody>
</table>

* $f_{\text{dc}}=58$ kHz