Nyquist-Rate A/D Converters

David Johns and Ken Martin
University of Toronto
(johns@eecg.toronto.edu)
(martin@eecg.toronto.edu)
A/D Converter Basics

\[ V_{\text{in}} \rightarrow \text{A/D} \rightarrow B_{\text{out}} \]

\[ V_{\text{ref}} \]

\[ V_{\text{ref}}(b_1 2^{-1} + b_2 2^{-2} + \ldots + b_N 2^{-N}) = V_{\text{in}} \pm x \]

where 
\[ \left( -\frac{1}{2} V_{\text{LSB}} < x < \frac{1}{2} V_{\text{LSB}} \right) \] (1)

- **Range of valid input values** produce the **same** output signal — quantization error.

\[ \frac{V_{\text{LSB}}}{V_{\text{ref}}} = 1/4 = 1.\text{LSB} \]
# Analog to Digital Converters

<table>
<thead>
<tr>
<th>Speed, Accuracy</th>
<th>Low-to-Medium Speed, High Accuracy</th>
<th>Medium Speed, Medium Accuracy</th>
<th>High Speed, Low-to-Medium Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrating</td>
<td>Integrating</td>
<td>Successive approximation</td>
<td>Flash</td>
</tr>
<tr>
<td>Oversampling</td>
<td>Oversampling</td>
<td>Algorithmic</td>
<td>Two-step</td>
</tr>
<tr>
<td>(not Nyquist-rate)</td>
<td></td>
<td></td>
<td>Interpolating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Folding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pipelined</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time-interleaved</td>
</tr>
</tbody>
</table>
Integrating Converters

- Low offset and gain errors for low-speed applications
- Small amount of circuitry
- Conversion speed is $2^{N+1}$ times $1/T_{clk}$

(Vin is held constant during conversion.)
Integrating Converters

- Count at end of $T_2$ is digital output
- Does not depend on RC time-constant
Integrating Converters

- Notches the input frequencies which are multiples of $1/T_1$

$|H(f)|$ (dB)

Frequency (Hz) (Log scale)

-20 dB/decade slope

University of Toronto
Successive-Approximation Converters

- Makes use of binary search algorithm
- Requires $N$ steps for $N$-bit converter
- Successively “tunes” a signal until within 1 LSB of input
- Medium speed
- Moderate accuracy
DAC Based Successive-Approximation

- Adjust $V_{D/A}$ until within 1 LSB of $V_{in}$
- Start with MSB and continue until LSB found
- D/A mainly determines overall accuracy
- Input S/H required
Charge Redistribution A/D

1. Sample mode

\[ V_X = 0 \]

\[ V_{in} \approx V_{ref} \]

2. Hold mode

\[ V_X = -V_{in} \]

\[ V_{in} \approx \frac{V_{ref}}{2} \]

3. Bit cycling

\[ V_{in} \approx \frac{V_{ref}}{2} \]
Charge Redistribution A/D

- McCreary, 75
- Combines S/H, D/A converter, and difference circuit
- **Sample mode**: Caps charged to $V_{in}$, compar reset.
- **Hold mode**: Caps switched to gnd so $V_x = -V_{in}$
- **Bit cycling**: Cap switched to $V_{ref}$. If $V_x < 0$ cap left connected to $V_{ref}$ and bit=1. Otherwise, cap back to gnd and bit=0. Repeat $N$ times
- Cap **bottom plates** connected to $V_{ref}$ side to minimize parasitic capacitance at $V_x$. Parasitic cap does not cause conversion errors but it attenuates $V_x$. 

\[ V_{in} \cdot V_x = V_{ref} \]

\[ V_{ref} < V_x < 0 \]
Algorithmic (or Cyclic) A/D Converter

- Operates similar to successive-approx converter
- Successive-approx halves ref voltage each cycle
- Algorithmic doubles error each cycle (leaving ref voltage unchanged)

1. Start
2. Sample V = V_{in}, i = 1
3. If V > 0, then:
   - If b_i = 1, then V → 2(V - V_{ref}/4)
   - If b_i = 0, then V → 2(V + V_{ref}/4)
4. i → i + 1
5. If i > N, then Yes, Stop; otherwise, No
   - If No, then i > N
   - If Yes, then Stop
Ratio-Independent Algorithmic Converter

- McCharles, 77; Li, 84
- Small amount of circuitry — reuse cyclically in time
- Requires a high-precision multiply by 2 gain stage
Ratio-Independent Algorithmic Converter

1. Sample remainder and cancel input-offset voltage.

2. Transfer charge $Q_1$ from $C_1$ to $C_2$.

3. Sample input signal with $C_1$ again after storing charge $Q_1$ on $C_2$.

4. Combine $Q_1$ and $Q_2$ on $C_1$, and connect $C_1$ to output.

- Does not rely on cap matching
- Sample input twice using $C_1$; hold first charge in $C_2$ and re-combine with first charge on $C_1$
Flash (or Parallel) Converters

- Peetz, 86; Yoshii, 87; Hotta, 87; and Gendai, 91

- High-speed
- Large size and power hungry
- $2^N$ comparators
- Speed bottleneck usually large cap load at input
- Thermometer code out of comps
- Nands used for simpler decoding and/or bubble error correction
- Use comp offset cancellation

![Diagram of Flash or Parallel Converter](image)
Issues in Designing Flash A/D Converters

• **Input Capacitive Loading** — use interpolating arch.

• **Resistor-String Bowing** — Due to $I_{in}$ of bipolar comps — force center tap (or more) to be correct.

• **Signal and/or Clock Delay** — Small arrival diff in clock or input cause errors. (250MHz 8-bit A/D needs 5ps matching for 1LSB) — route clock and $V_{in}$ together with the delays matched [Gendai, 1991]. Match capacitive loads

• **Substrate and Power-Supply Noise** — $V_{ref} = 2$ V and 8-bit, 7.8 mV of noise causes 1 LSB error — shield clocks and use on-chip supply cap bypass

• **Flashback** — Glitch at input due to going from track to latch mode — use preamps in comparators and match input impedances
Flash Converters — Bubble Errors

- Thermometer code should be 1111110000
- Bubble error (noise, metastability) — 1111110100
- Usually occurs near transition point but can cause \textit{gross errors} depending on encoder

- Can allow errors in lower 2 LSB but have MSBs encoder look at every 4th comp [Gendai, 91]

\[ \text{(2N–1) to N encoder} \]

\[ \text{N digital outputs} \]

[Steyaert, 93]
Reduced Auto-Zeroing

• Tsukamoto et al, ISSCC/96
• Spalding et al, ISSCC/96
• Reduce the auto-zero portion of conversion
  — auto zero when not performing conversion
  — add one more comparator and ripple up auto-zero

Advantages

• Lower power — less current drawn from ref string
• More speed — more time for conversion

Disadvantage

• 1/f noise not rejected as much
Two-Step A/D Converters

- High-speed, medium accuracy (but 1 sample latency)
- Less area and power than flash
- Only 32 comparators in above 8-bit two-step
- Gain amp likely sets speed limit
- Without digital error correction, many blocks need at least 8-bit accuracy
Digital Error Correction

- Relaxes requirements on input A/D
- Requires a 5-bit 2nd stage since $V_q$ increased
- Example, see [Petschacher, 1990].
Interpolating A/D Converters

- Goodenough, 1989
- Steyaert, 1993
- Kusumoto, 1993

Use input amps to amplify input around reference voltages

Latch thresholds less critical

Less cap on input (faster than flash)

Match delays to latches

Often combined with folding architecture
Interpolating Converters

Latch threshold

$V_{in}$ (Volts)

$V_1, V_2, V_{2a}, V_{2b}, V_{2c}$

$I_1, I_{2a}, I_{2b}, I_2$

(Relative width sizing shown)

(All lengths same)

current interpolation
Folding A/D Converters

- Reduce number of latches using folding
- Save power and area
- Similar concept to 2-step
- Folding rate of 4 shown for 4 bit converter

Folding block responses

V_{in} = 1 \ V

V_{r} = \begin{cases} 
16, 16, 16, 16 \\
16, 16, 16, 16 \\
16, 16, 16, 16 \\
16, 16, 16, 16 \\
\end{cases}

Digital logic

Latch

b_1

b_2

V_1, V_2, V_3, V_4
Folding Circuit

\[ V_{in} \rightarrow I_b \rightarrow V_{a} \rightarrow V_{r1} \rightarrow Q_1 \rightarrow V_{CC} \rightarrow R_1 \rightarrow V_{b} \rightarrow V_{r2} \rightarrow Q_2 \rightarrow V_{out} \]

\[ V_{out} \]

\[ V_{CC} - V_{BE} \]

\[ V_{CC} - V_{BE} - I_b R_1 \]

\[ (a) \]

\[ (b) \]
Folding with Interpolation

- Folding usually used with interpolation
- Reduces input cap
- Without interp, same input cap as flash
  - [van Valburg, 1992]
  - [van de Grift, 1987]
  - [Colleran, 1993]
Pipelined A/D Converters

Analog pipeline (DAPRX - digital approximator)

V_{in} → 1-bit DAPRX → 1-bit DAPRX → ... → 1-bit DAPRX

\( V_{i-1} \) → S/H → Cmp

\( V_i \)
Time-Interleaved A/D Converters [Black, 80]

- Use parallel A/Ds and multiplex them
- Tone occurs at fs/N for N converters if mismatched
- Input S/H critical, others not — perhaps different tech for input S/H