Integrating and Algorithmic ADCs

Voltage-to-Time Converters

Thanks to Dr. Sebastian Hoyos for providing part of this material
Single-Slope Integrating ADC

- Counter keeps counting until comparator output toggles.
- Simple, inherently monotonic, but very slow ($2^N \times T_{clk}$/sample).
**Single-Slope Integrating ADC**

- Precision capacitor (C) and current source (I) are required.
- INL depends on the linearity of the ramp signal.
- Errors due to current-source output impedance and leakage current.
- Comparator must handle large common-mode input.
- Comparator’s voltage dependent offset.
- Finite Switch Resistance.

The relationship between the input and output is given by:

\[ V_i = \frac{I}{C} \cdot t_1, \quad D_o = \frac{t_1}{T_{\text{clk}}} \]

\[ \Rightarrow D_o = \frac{C}{I \cdot T_{\text{clk}}} \cdot V_i. \]
Dual-Slope Integrating ADC

- CT RC integrator replaces the current integrator.
- Input and reference voltages experience the same signal path.
- Comparator only detects zero-crossing (constant input CM).
- Clock injection and charge injection effects?
- Noise effects due to R, opamp and Vref?
- OPAMP specifications? Reset needed?
Dual-Slope Integrating ADC

![Diagram of Dual-Slope Integrating ADC]

- Exact values of R, C, and T\(_{\text{clk}}\) are not required.
- Comparator offset doesn’t matter.
- Op-amp offset introduces gain error and offset.
- Op-amp nonlinearity introduces INL error.

\[ V_m = \frac{V_{\text{in}}}{RC} \cdot t_1 = \frac{V_R}{RC} \cdot t_2 \]

\[ \Rightarrow D_0 = \frac{N_2}{N_1} = \frac{t_2}{t_1} = \frac{V_{\text{in}}}{V_R} \]

\[ t_2 = (t_1) \left( \frac{V_{\text{in}}}{V_R} \right) \]

\( t_2 \) is the time window used to count the clock pulses.
Subranging Dual-Slope ADC

If \( V_x > V_t \), then use both current sources. Slope is \(~256\) times faster.

When \( V_x < V_t \), then use the small current only, better resolution.

- Much faster conversion speed.
- Two matched current sources and two comparators are required.
Subranging Dual-Slope ADC

- Precise $V_t$ is not required if carry is propagated.
- Matching between the current sources is critical.
  
  → if $I_1 = I$, $I_2 = (1+\delta) \cdot I / 256$, then $|\delta| \leq 0.5/256$. 

\[
\frac{dV_x(1)}{dt} = \frac{I}{C}, \\
\frac{dV_x(2)}{dt} = \frac{I}{256C}.
\]

\[
D_o = N_1 \cdot W + N_2
\]
Subranging Multi-Slope ADC

Subranging Multi-Slope ADC

- Single comparator detects zero-crossing.
- Comparator response time is greatly relaxed.
- Matching between the current sources is still critical.

\[
\begin{align*}
\frac{dV_x(1)}{dt} &= \frac{I}{C}, \\
\frac{dV_x(2)}{dt} &= \frac{I}{16C}, \\
\frac{dV_x(3)}{dt} &= \frac{I}{256C}.
\end{align*}
\]

\[D_o = N_1 \cdot W_1 - N_2 \cdot W_2 + N_3\]
Successive Approximation ADC
Successive Approximation ADC

- **Binary search algorithm** → \( N^*T_{\text{clk}} \) to complete \( N \) bits.
- Conversion speed is limited by comparator, DAC, and SAR (successive approximation register)
- **Fundamental assumption:** DAC is more precise than ADC!
**Binary Search**

\[ V_i \approx \sum_{i=1}^{N} \left( b_i V_{FS} \right) / 2^i \]

- DAC output gradually approaches the input voltage.
- Comparator differential input gradually approaches zero.
- Notice that errors in each computation get accumulated!
**CHARGE RECOMBINATION**

\[ V_Y - C_1 \quad U_X \]

\[ C_2 \]

\[ i_i \]

\[ V_X \]

\[ Q = C_x U_x \]

\[ \text{i}_i \text{ is not going to change the charge} \]

\[ \text{seen at node } \ U_X \]

\[ @ \ U_X \quad \Delta Q_{C_2} = - \Delta Q_{C_1} \]

\[ Q_{C_2} = \int_{t_0}^{t_1} i_i \ dt \]

\[ \Phi_{C_1} + \Phi_{C_2} \bigg|_{t = t_1} = \Phi_{C_1} + \Phi_{C_2} \bigg|_{t = t_0} \]

\[ C_2 U_X + C_1 (U_X - U_Y) \bigg|_{t = t_1} = \Phi_{C_1}(t_0) + \Phi_{C_2}(t_0) \]

\[ V_X = \left( \frac{C_1}{C_1 + C_2} \right) V_Y + \left( \frac{Q_{C_1} + Q_{C_2}}{C_1 + C_2} \right)_{t = t_0} \]
Charge Redistribution ADC

\[ (Q_{c1} + Q_{c2} + ...)_{t=t_0} = [8 + 4 + 2 + 1 + 1] \cdot C \cdot V_i(t_0) \] \( \Phi_{1e} \)

- **4-bit binary-weighted capacitor array DAC.**
- **Capacitor array samples input when \( \Phi_1 \) is asserted (bottom-plate).**
- **Extremely power efficient (mainly passive)**
- **Medium resolution; speed is limited by switch velocity**
\[ V_x = \left( \frac{C_1}{C_1 + C_2} \right) V_y + \left( \frac{Q_{C1} + Q_{C2}}{C_1 + C_2} \right) \]

\[ V_x = V_{\text{REF}} - U_i \]

\[ U_x [16 C_0] - U_{\text{REF}} [12 C_0] = 16 C_0 \left[ \frac{V_{\text{REF}}}{2} - U_i \right] \]

\[ V_x = \frac{12}{16} V_{\text{REF}} - U_i \]

\[ U_x = \frac{3}{4} U_{\text{REF}} - U_i \]
Fundamentals on ADCs: Dual-Slope and Algorithmic

\[ V_{FS} \quad 2^n C_0 \quad V_X \]

\[ 2^n C_0 \quad U_X - 2^n C_0 U_{FS} = -2^{n+1} V_i; \]

\[ U_X = \frac{V_{FS}}{2} - U_i. \]

\[ V_{FS} \quad \frac{1}{2^n - 2^{n-1}} C_0 \quad U_X \]

\[ 2^{n+1} C_0 \quad U_X - 2^n C_0 V_{FS} = 2^{n+1} C_0 \left( \frac{V_{FS}}{2} - U_i \right) - 2^n C_0 V_{FS}; \]

\[ V_X = \frac{3}{4} U_{FS} - U_X. \]

\[ V_{FS} \quad 2^{-1} C_0 \quad U_X \]

\[ 2^{n+1} C_0 \quad U_X - 2^n C_0 U_{FS} = 2^n C_0 \left( \frac{V_{FS}}{2} - U_i \right) - 2^n C_0 U_{FS}; \]

\[ V_X = \frac{V_{FS}}{4} - U_X. \]
Charge Redistribution ADC

- **4-bit binary-weighted capacitor array DAC.**
- **Capacitor array samples input when \( \Phi_1 \) is asserted (bottom-plate).**
- **Extremely power efficient (mainly passive; *Dynamic Power (?)*)**
- **Medium resolution; speed is limited by switch velocity**
Charge Redistribution (MSB)

\[ V_i \cdot \sum_{j=0}^{4} C_j = (V_R - V_X) \cdot C_4 - V_X \cdot \sum_{j=0}^{3} C_j \Rightarrow V_X = \left( V_R \cdot C_4 - V_i \cdot \sum_{j=0}^{4} C_j \right) / \sum_{j=0}^{4} C_j = \frac{V_R}{2} - V_i \]
Comparison (MSB)

\[ V_X = \frac{V_R}{2} - V_i \]

- If \( V_X < 0 \), then \( V_i > V_R/2 \), and MSB = 1, \( C_4 \) remains connected to \( V_R \).
- If \( V_X > 0 \), then \( V_i < V_R/2 \), and MSB = 0, \( C_4 \) is switched to ground.
Charge Redistribution (MSB-1)

\[ V_i \cdot 16C = (V_R - V_X) \cdot 12C - V_X \cdot 4C \Rightarrow V_X = \frac{(V_R \cdot 12C - V_i \cdot 16C)}{16C} = \frac{3}{4} V_R - V_i \]
Comparison (MSB-1)

MSB TEST: \( V_X = \frac{3}{4} V_R - V_i \)

- If \( V_X < 0 \), then \( V_i > \frac{3V_R}{4} \), and MSB-1 = 1, \( C_3 \) remains connected to \( V_R \).
- If \( V_X > 0 \), then \( V_i < \frac{3V_R}{4} \), and MSB-1 = 0, \( C_3 \) is switched to ground.
Test completes when all four bits are determined with four charge redistributions and comparisons.
After Four Clock Cycles…

- Usually, half $T_{\text{clk}}$ is allocated for charge redistribution and half for comparison + digital logic.
- $V_X$ always converges to 0 ($V_{\text{os}}$ if comparator has nonzero offset).
Bottom-Plate Parasitics

- If $V_{os} = 0$, $C_P$ has no effect; otherwise, $C_P$ attenuates $V_X$.
- AZ can be applied to the comparator to reduce offset.
Summary on SA ADC

- Power efficiency – only comparator consumes DC power.
- DAC nonlinearity limits the INL and DNL of the SA ADC
  - N-bit precision requires N-bit matching from the cap array.
  - Calibration can be performed to remove mismatch errors (Lee, JSSC 84 at the expenses of speed).
- If $C_P=0$, comparator offset $V_{os}$ introduces an input-referred offset $V_{os}$; for nonzero $C_P$, input-referred offset is larger than $V_{os}$ ($\delta \sim \frac{C_P}{\Sigma C_j}$).
- If $V_{os}=0$, CP has no effect ($V_X \rightarrow 0$ at the end of search); otherwise, charge sharing occurs at summing node ($V_X$ is attenuated).
- Binary search is sensitive to intermediate errors made during search
  - DAC must settle into $\frac{1}{2}$ LSB within the time allowed.
  - Comparator offset must be constant (no hysteresis).
  - Nonbinary search can be used (Kuttner, ISSCC, 2002).
References

Algorithmic ADC

Re-using the building blocks
MAIN CONCEPTS:

- Comparison \( B_i = ? \) \( \Rightarrow \) conditional
- Level shifter \( \Rightarrow \times 2 \)

- **Determine MSB**

**MSB = 1**

\[ V_F S \]

\[ V_F S / 2 \]

\[ V_i \]

\[ B_1 = 1 \]

\[ B_1 = 0 \]

\[ \Rightarrow \text{ comparator} \]

\[ \Rightarrow \text{ Level shifter} \]

**MSB = 0**

\[ V_F S / 2 \]

\[ V_F S / 2 \]

\[ V_i \]

\[ B_1 = 0 \]

\[ \Rightarrow \text{ NO-OPERATION} \]

\[ \Rightarrow \times 2 \]

\[ V_F S / 2 \]

\[ V_i \]

\[ \Rightarrow \text{ 2 } \]

\[ V_F S / 2 \]

\[ \Rightarrow \text{ 0} \]

**Repeat the algorithm**
Algorithmic ADC: Basic Operations

**Fundamentals on ADCs: Dual-Slope and Algorithmic**

Jose Silva-Martinez

Texas A&M University

Spring, 2013
Algorithm

\[ V_y = V_i \]

\[ V_0 = \frac{V_{FS}}{2} \]

\[ V_y = 2 \left( V_y - \frac{V_{FS}}{2} \right) \]

\[ V_y = 2V_y \]

1) \( V_i - \frac{V_{FS}}{2} \) > 0 \( \Rightarrow \) \( y \Rightarrow B_1 = 1 \)

\[ V_y = 2 \left( V_i - \frac{V_{FS}}{2} \right) \]

UPDATE \( V_y \)

2) \( 2 \left( V_i - \frac{V_{FS}}{2} - \frac{V_{FS}}{8} \right) \) > 0 \( \Rightarrow \)

\[ 2 \left( V_i - \frac{6V_{FS}}{8} \right) \] \( \geq 0 \) \( \Rightarrow \) \( B_2 = 0 \)

\[ V_y = 4 \left( V_i - \frac{4V_{FS}}{8} \right) \]

UPDATE \( V_y \)

3) \( 4 \left( V_i - \frac{4V_{FS}}{8} - \frac{1V_{FS}}{8} \right) \) > 0 \( \Rightarrow \) \( y \Rightarrow B_3 = 1 \)

\[ V_y = 8 \left( V_i - \frac{5V_{FS}}{8} \right) \]

UPDATE \( V_y \)

4) ...
Algorithmic (Cyclic) ADC

- Input is sampled first, then circulates in the loop for N clock cycles.
- Conversion takes N cycles with one bit resolved in each $T_{\text{clk}}$. 
Modified Binary Search Algorithm

- If $V_X < V_{FS}/2$, then $b_j = 0$, and $V_o = 2*V_X$.
- If $V_X > V_{FS}/2$, then $b_j = 1$, and $V_o = 2*(V_X-V_{FS}/2)$. 
Modified Binary Search Algorithm

- Constant threshold ($V_{FS}/2$) is used for each comparison.
- 2X gain is provisioned each time residue circulates around the loop.
Fundamentals on ADCs: Dual-Slope and Algorithmic

Jose Silva-Martinez

\[
\begin{align*}
\phi_s & \quad U_{in} \\
\phi_1 & \quad U_{ref} \\
\phi_2 & \quad C_1 \\
& \quad C_2 \\
& \quad b_1, b_0 = 1 \\
\phi_3 & \quad C_1 \\
\phi_4 & \quad U_{ref} \\
& \quad U_0 = 2U_{in} - U_{ref} \\
& \quad U_{c1} = U_{c2} = 2U_{in} - U_{ref} \\
\end{align*}
\]
\[ \phi_1, \text{ next } = \begin{cases} \text{If } b_1 = 1 & \Rightarrow \frac{C_2}{C_1} V_{\text{ref}} + V_{x1} = 4 \left[ V_{\text{in}} - \frac{3}{4} V_{\text{ref}} \right] \\ \text{If } b_1 = 0 & \Rightarrow \frac{C_2}{C_1} V_{x1} = 4 \left[ V_{\text{in}} - \frac{V_{\text{ref}}}{4} \right] \end{cases} \]

\[ \phi_2 : \text{Sample } V_0 \text{ into } C_2 \]

\[ \Rightarrow \phi_1, \text{ next-next } = \text{Repeat the same procedure} \]

This circuit is offset and mismatch sensitive.

Analyze the circuit after \( \phi_2 \); assume the opamp is ideal.

\[ \phi_2 \rightarrow n \Delta C_1 = n \Delta C_2 = V_i \quad \Rightarrow \quad \phi_1 \]

\[ \frac{C_2}{C_1} V_{\text{ref}} + V_{x1} = 4 \left[ V_{\text{in}} - \frac{3}{4} V_{\text{ref}} \right] \]
\[ V_o = \frac{1}{C_1 + C_2} V_i \]

MSG is decided based on

\[ (1 + \frac{C_2}{C_1}) V_i - \frac{C_2}{C_1} V_{REF} \geq 0 \]

After this, \( C_2 \) is pre-charged

If \( b_1 = 1 \), then \( C_{next} \)

\[ V_{REF} \]

\[ C \quad V_0 (next) + \quad C_2 V_{REF} = (C_1 + C_2) V_0 \]

\[ V_0^{next} = (1 + \frac{C_2}{C_1}) V_0 - \frac{C_2}{C_1} V_{REF} \]

\[ V_0^{next} = \left(1 + \frac{C_2}{C_1}\right)\left(1 + \frac{C_2}{C_1}\right) V_i - \left[\left(1 + \frac{C_2}{C_1}\right) + 1\right] \frac{C_2}{C_1} V_{REF} \]
Fundamentals on ADCs: Dual-Slope and Algorithmic

The next decision is taken based on:

\[(1 + \frac{C_2}{C_1})^2 V_i - \left(2 + \frac{C_2}{C_1}\right) \frac{C_2}{C_1} V_{\text{REF}} \geq 0\]

For \(MSB\), \(S_i = \left[\frac{C_2/C_1}{1 + C_2/C_1}\right] V_{\text{REF}}\)

For \(MSB-1\), \(S_i = \left[\frac{(2 + \frac{C_2}{C_1}) \frac{C_2}{C_1}}{(1 + \frac{C_2}{C_1})^2}\right] V_{\text{REF}}\)

If \(\frac{C_2}{C_1} = 1 + \Delta z_i \Rightarrow\)

\[
\frac{C_2/C_1}{1 + C_2/C_1} \simeq \frac{1 + \Delta z_i}{2 + \Delta z_i} = \frac{1}{2} + \frac{\Delta z_i}{4} = \frac{1}{2} \left(1 + \frac{\Delta z_i}{2}\right)
\]

\[
\frac{(3 + \Delta z_1)(1 + \Delta z_2)}{4 + 2(\Delta z_1)} \simeq \frac{3}{4} = \frac{1 + \frac{\Delta z_1}{3}}{1 + \Delta z_1}
\]

\[
\frac{2}{3} = \frac{\Delta z_1}{3}
\]
How the $x^2$ can be implemented without this issue?

$\Rightarrow$ more complexity

\[
\phi_1 = -\frac{c_2}{c_1} u_x \\
\phi_2 = c_2 v_{o_2} + 0 = c_2 u_x + c_2 u_{o_1} \\
c_2 v_{o_2} = c_2 u_x + c_2 u_x = 2 c_2 u_x \\
\boxed{v_{o_2} = 2 u_x} \quad \text{no magic?}
\]
Start

Sample $V = V_{in}$, $i = 1$

$V > 0$

No

$V \rightarrow 2(V - V_{ref} / 4)$

$b_i = 0$

Yes

$V \rightarrow 2(V + V_{ref} / 4)$

$b_i = 1$

$i \rightarrow i + 1$

No

$i > N$

Yes

Stop

Signed input

Fig. 13.13 Flow graph for the algorithmic approach.

Chapter 13, Johns and Martin Book

Fig. 13.14 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.

Fig. 13.15 Multiply-by-two gain circuitry for an algorithmic converter that does not depend on capacitor matching.
Algorithmic ADC

- Block diagram of switched-capacitor algorithmic ADC
- Operates with two non-overlapping clock phases
- The converter reaches its final output in N cycles
- Capacitors $C_0$, $C_1$ and $C_2$ are all nominally equal
- The $C_3$ capacitor is used for sampling and canceling the amplifier offset

\[ Q_b = Q_1 V_s + Q_2 V_s \]
\[ Q_c = Q_b \]
• The input is sampled during clock phase Qs.

• During clock phase $Q_1$, the previous output is sampled onto capacitor $C_2$. The purpose of this is to multiply the output by a factor of two during the next clock phase.

• The output bit from the previous cycle determines if $V_{\text{ref}}$ or ground is connected to $C_1$. This process subtracts or adds $V_{\text{ref}}$ to the analog output of the amplifier, $V_{\text{out}}$.

• During clock phase $Q_2$, the new analog output $V_{\text{out}}$ is available and the comparator produces the next digital output bit.

• This is repeated for $N$ cycles to produce the final output word for $N$-bit resolution.
The input is sampled onto only one capacitor, reducing the input capacitance

The amplifier operates in both clock phases and there are no wasted cycles where the amplifier is idle

Switch configuration makes the amplifier insensitive to parasitics at the input

The analog output of the first cycle can be expressed as

$$V_{out}[1] = \frac{C_1}{C_0 + \frac{C_0 + C_1}{A}} V_{in} - V_{offset} \frac{C_0 + C_1 + C_2}{C_0 + \frac{C_0 + C_1}{A}}$$

The output of cycle $n$ can then be expressed as

$$V_{out}[n] = \left( \frac{C_0 + C_2 + \frac{C_0}{A}}{A} \right) V_{out}[n-1] - D_{n-1} C_1 V_{ref} - \left( C_1 + C_2 \right) V_{offset} \frac{C_0 + \frac{C_0 + C_1 + C_2}{A}}{C_0 + \frac{C_0 + C_1 + C_2}{A}}$$

Main source of error caused by capacitor mismatch and finite amplifier gain
Loop Transfer Function

- If $V_X < V_{FS}/2$, then $b_j = 0$, and $V_o = 2*V_X$.
- If $V_X > V_{FS}/2$, then $b_j = 1$, and $V_o = 2*(V_X-V_{FS}/2)$. 

\[ \begin{align*} 
V_i &\quad \text{SHA} \quad V_X \quad + \quad \text{DAC} \quad V_{FS}/2 \\
&\quad \text{2X} \quad \text{V}_o \quad 0 \quad b_j \\
&\quad V_{FS}/2 \quad b_j = 0 \quad b_j = 1 \\
\end{align*} \]
Offset Errors

\[
V_o = 2^*(V_i - b_j*V_{FS}/2) \rightarrow V_i = b_j*V_{FS}/2 + V_o/2
\]
The Multiplier DAC (MDAC)

- 2X gain + 3-level DAC + subtraction all integrated.
- A 3-level DAC is perfectly linear w/ fully-differential signals.
The 1.5-Bit Architecture

- 3 decision levels → ENOB = \( \log_2 3 = 1.58 \).
- Max tolerance on comparator offset is ±\( V_R/4 \).
- An implementation of the Sweeny-Robertson-Tocher (SRT) division principle.
- The conversion accuracy solely relies on the loop gain error, i.e., the gain error and nonlinearity.
- A 3-level DAC is required.

Architecture can be generalized to n.5-b per conversion.
Error Mechanisms of RA

- Capacitor mismatch
- Op-amp finite-gain error and nonlinearity
- Charge injection and clock feedthrough
- Finite circuit bandwidth

\[ V_o = \frac{C_1 + C_2}{C_1} \cdot V_i - (b - 1) \frac{C_2}{C_1} \cdot V_R \]

\[ V_o(t = \infty) = \frac{(C_1 + C_2) \cdot V_i - (b - 1) C_2 \cdot V_R}{C_1 + \frac{C_1 + C_2}{A(V_o)}} + \Delta V(V_i) \]
RA Gain Error and Nonlinearity

Raw accuracy is usually limited to 10-12 bits w/o error correction.
MDACs for 1 bit and 1.5 bit with and without tolerance to comparator offset

1 bit and 1.5-b with Offset Tolerance

\[ D_0 = \frac{1}{2^N} \left( b_0 + b_1 2 + b_1 2^2 + \cdots + b_{N-1} 2^{N-1} \right) \]

1.5-b without Offset Tolerance

\[ D_0 = \frac{2}{3^N} \left( b_0 + b_1 3 + b_1 3^2 + \cdots + b_{N-1} 3^{N-1} \right) \]
Effect of Offset

4-Stage Pipeline ADC with 1 and 1.5 bits per stage and $V_{\text{offset}} = V_R/4$

- SNR = 17.995 dB (1 bit)
- SNR = 24.88 dB (1.5 bits with offset tolerance)
- SNR = 20.467 dB (1.5 bits with NO offset tolerance)
Effect of Offset

8-Stage Pipeline ADC with 1 and 1.5 bits per stage and $V_{offset} = VR/4$

- SNR = 22.185 dB (1 bit)
- SNR = 49.5 dB (1.5 bits with offset tolerance)
- SNR = 21.451 dB (1.5 bits with NO offset tolerance)
References

Algorithmic ADC

- Hardware-efficient, but relatively low conversion speed.
- Binary search algorithm.
- Loopgain (2X) requires the use of a residue amplifier, but greatly simplifies the DAC – 1-bit, inherently linear.
- Residue gets amplified each time it circulates the loop; the gain makes the later conversion steps (the LSB’s) insensitive to circuit noise and distortion.
- Conversion errors (residue error due to loopgain nonidealities and comparator offset) made in the earlier conversion cycles also get amplified again and again – overall accuracy is usually limited by the MSB resolving and residue generation step.
- Digital redundancy is often used to treat comparator/loop offsets.
- Trimming/calibration/ratio-independent techniques are often used to treat loopgain error.
Precision Techniques
Ratio-Independent Multiply-By-N Circuit

- Sampling
- Charge transfer
- $C_1 - C_2$ exchanged

- Steps (1) and (2) are repeated for N times.
- $C_2$ functions as a temporary charge storage.
RA Gain Trimming

- Precise gain-of-two is achieved by adjustment of the trim array.
- Finite-gain error of op-amp is also compensated (not nonlinearity).

\[ \frac{C_1}{C_2} = 2 \text{ nominally} \]
Split-Array Trimming DAC

- Successive approximation is performed to find the correct gain setting.
- Coupling cap is slightly increased to ensure segment overlap.
Digital Calibration

- RA gain is set lower than 2X, forcing missing codes but **NOT** missing decision levels.

- Calibration is performed by measuring distance between $S_1$ and $S_2$; ($S_2 - S_1$) is later subtracted from $D_0$ (gain error and offset remain).
Digital Calibration

\[ V_o = \frac{(C_1 + C_2)V_i - (2b - 1)C_1V_R}{C_2 + C_3} \]

- \(C_1 = C_2\)
- \(C_3 = \beta C_1\)
- AZ SC amplifier

Residue Amplifier
Capacitor Error-Averaging

\[ V_{x1} = \frac{(C_1 + C_2)V_i - C_2V_R}{C_1} = 2V_i - V_R + \delta(V_i - V_R), \]

\[ V_{x2} = \frac{(C_1 + C_2)V_i - C_1V_R}{C_2} \approx 2V_i - V_R - \delta(V_i - V_R). \]

Capacitor Error-Averaging

\[ V_{x1}(C_3 - C_4) = V_o C_3 - V_{x2} C_4, \]

\[ V_o = \frac{V_{x1} + V_{x2}}{2} \approx 2V_i - V_R, \]
Capacitor Error-Averaging

\[ C_1 = C(1 + \delta_1) \quad C_2 = C(1 + \delta_2) \quad C_3 = 2C(1 + \delta_3) \quad C_4 = C(1 + \delta_4) \]

Assume \( \delta_1, \delta_2, \delta_3 \text{ and } \delta_4 \) are zero mean Gaussian with variance \( \sigma^2 \).

Find an expression for \( V_0 \) and comment on the effectiveness of the capacitor error-averaging technique.
References

Subranging ADC
Subranging ADC Architecture

- $V_{RT}$
- $V_{RB}$
- $V_i$
- Coarse Encoder
- Fine Encoder
- Coarse Flash
- Fine Flash
- MSB’s
- LSB’s
Subranging ADC

Pros
• Reduced complexity – $2^N - 1$ comparators
• Reduced $C_{\text{in}}$, area, and power consumption
• No residue amplifier required

Cons
• Typically 3 clock phases per conversion
  – Sample
  – Coarse comparison
  – Fine comparison
• THA required (two-stage S/H if the front-end SHA only holds for one phase)
• Offset tolerance on fine comparators is at N-bit level.
• Offset tolerance on coarse comparators is also at N-bit level without digital redundancy.
Redundancy in fine ADC provided by over- and under-range comparators
Digital Redundancy of Fine ADC

Search range of the fine ADC is extended on both sides.
Two-Step ADC

- Coarse-fine two-step subranging architecture
- Conversion residue is produced instead of switching reference taps.
- A DAC and a subtraction circuit are required.
- Residue gain can be provisioned to relax offset tolerance in fine ADC.
Four conversion steps can be pipelined.
Usually DAC + RA settling takes the longest time.
RA is often omitted (residue gain of one) to speed up conversion.
References