ESE 568: Mixed Signal Design and Modeling

Lec 16: November 1st, 2017
Pipeline ADCs (con’t)
Outline

- Background
  - General idea of multi-step/subranging A/D conversion

- Pipeline ADC basics
  - Ideal block diagram and operation

- Nonidealities and correction mechanisms
  - How to cope with circuit nonidealities

- CMOS implementation details
  - I/O, MDAC, comparators, OTAs, stage scaling, ...

- Architectural options
  - OTA sharing, SHA-less front-end

- Research topics
Multi-Step A/D Conversion

- Multi-step ADCs trade speed for reduced complexity and power
- Implementation example: Razavi & Wooley, JSSC 12/1992
  - 12-bit, 5MS/s, two-step ADC (6-bit/7-bit),
Concept

4-bit Flash ADC

FS
1111 (15)
1110 (14)
1101 (13)
1100 (12)
1011 (11)
1010 (10)
1001 (9)
1000 (8)
0111 (7)
0110 (6)
0101 (5)
0100 (4)
0011 (3)
0010 (2)
0001 (1)
0000 (0)

V_in

DAC

0111 (7)

Ideal 2-step (2-2) ADC

FS
11
10
FS/4

11
10
01
00

V_in

0111 (7)
Pipeline ADC Block Diagram
Uses MDAC to form a residual voltage

\[ V_{i+1} = 2[V_i + (b_i - 0.5)V_{\text{ref}}/2] \]
Building Block Analysis

- Ignore timing and use simple static model

- Each stage (except for last) has two tasks
  - Coarse quantization
  - Calculate "residue"

Penn ESE 568 Fall 2017 - Khanna adapted from Murmann
EE315B, Stanford
Stage Analysis

For algebraic convenience, treat B-bit ADC as unity gain block with quantization error.

\[ V_{\text{res}} = G \cdot [V_{\text{in}} - V_{\text{dac}}(D)] \]

\[ D = Q(V_{\text{in}}) = V_{\text{in}} + \varepsilon_q \]
Ideal Quantizer

- Full scale range defined by max. inputs that keep quantization error within ±1/2 LSB
Resulting Model (w/ Ideal DAC)

- Residue is amplified quantization error
Residue Waveform (2-bit Sub-ADC)
Stage Gain

- Can make $G$ as large as $2^B$ without overloading next stage quantizer (assuming same full-scale range)
- Sometimes advantageous to make $G$ smaller than $2^B$
- Sometimes advantageous to use larger or smaller full-scale range in following quantizer
Issue with $G = 2^B$

- Any error in sub-ADC decision levels will overload backend ADC and thereby deteriorate ADC transfer function.
Example: Comparator Offset

Problem: $V_{res1}$ exceeds 2nd pipeline stage overload range

Overall ADC Transfer Curve

First stage ADC Levels:
(Levels normalized to LSB)
Ideal comparator threshold: \(-1, 0, +1\)
Comparator threshold including offset: \(-1, 0.3, +1\)

Missing Code!
Idea #1: G slightly less than $2^B$

- Effective stage resolution can be non-integer ($R = \log_2 G$)
  - E.g. $R = \log_2 3.2 = 1.68$ bits
- See e.g. [Karanicolas 1993]
Idea #2: $G < 2^B$, but Power of 2

- Effective stage resolution is an integer
  - E.g. $R = \log_2 2 = 1 = B - 1$
  - Digital hardware requires only a few adders, no need to implement fractional weights
- See e.g. [Mehr 2000]
Example: Comparator Offset

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!
Idea #3: $G=2^B$, Extended Backend Range

- No redundancy in stage with errors ($G=2^B$)
- Extra decision levels in succeeding stage used to bring residue “back into the box”
- See e.g. [Opris 1998]
Common Variant: “1.5-bit Stage”

- Sub-ADC decision levels placed to minimize comparator count
- Can accommodate errors up to $\frac{1}{4}$ LSB
- $B = \log_2(3) = 1.589$ (sub-ADC resolution)
- $R = \log_2 2 = 1$ (effective stage resolution)
- See e.g. [Lewis 1992]
Redundancy

- The preceding analysis applies to any stage in an n-stage pipeline
- Can always look at pipeline as a single stage + backend ADC
Redundancy

- In literature, sub-ADC redundancy schemes are often called "digital correction" – a misnomer!
- No error correction takes place
- We can tolerate sub-ADC errors as long as:
  - The residues stay "within the box", or
  - Another stage downstream "returns the residue to within the box" before it reaches last quantizer
- Let's calculate tolerable errors for the popular "1.5 bits/stage" topology
Pipeline Decomposition

- Convenient to look at pipeline as a single stage plus backend ADC
Resulting Model

\[ D_{out} = V_{in} + \varepsilon_q \left( 1 - \frac{G}{G_d} \right) + \frac{\varepsilon_{qb}}{G_d} \]

With \( G_d = G \)

Backend ADC
Canonical Extension

- First stage has most stringent precision requirements
- Note that above model assumes that all stages use the same reference voltage (same full scale range)
  - True for most designs, one exception is [Limotyrakis, ‘05]

\[
D_{out} = V_{in,ADC} + \varepsilon_{q1}\left(1 - \frac{G_1}{G_{d1}}\right) + \varepsilon_{q2}\frac{1 - G_2}{G_{d1}} + \cdots + \varepsilon_{q(n-1)}\frac{1 - G_{(n-1)}}{\prod_{j=1}^{n-2} G_{d(j)}} + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}
\]
General Result – Ideal Pipeline ADC

- With ideal sub-DAC and ideal weights \( (G=G_d) \)

\[
D_{out} = V_{in} + \frac{e_{qn}}{\prod_{j=1}^{n-1} G_j} \quad \Rightarrow \quad B_{ADC} = B_n + \sum_{j=1}^{n-1} \log_2 G_j
\]

- The only error in \( D_{out} \) is that of last quantizer, divided by aggregate gain

- Aggregate ADC resolution is independent of sub-ADC resolutions in stage 1...n-1

- Makes sense to define “effective” resolution of \( j^{th} \) stage as \( R_j = \log_2 (G_j) \)
Questions

- How to pick stage gain $G$ for a given sub-ADC resolution
- Impact and compensation of non-idealities?
  - Sub-ADC errors
  - Amplifier offset
  - Amplifier gain error
  - Sub-DAC error
- We just explored these questions using simple example
  - First stage with 2-bit sub-ADC, followed by 2-bit backend
Redundancy

- In literature, sub-ADC redundancy schemes are often called "digital correction" – a misnomer!
- No error correction takes place
- We can *tolerate* sub-ADC errors as long as:
  - The residues stay "within the box", or
  - Another stage downstream "returns the residue to within the box" before it reaches last quantizer
- Let's calculate tolerable errors for the popular "1.5 bits/stage" topology
1.5-bit Stage

- $B = \log_2(3) = 1.589$ (sub-ADC resolution)
- $R = \log_22 = 1$ (effective stage resolution)
- 0.5 bit $\rightarrow$ redundancy
3-Stage Pipeline ADC w/ 1.5-bit Stage

- All three stages
  - Comparator with offset
- Overall transfer curve
  - No missing codes
  - Some DNL error


Penn ESE 568 Fall 2017 - Khanna adapted from Murmann
EE315B, Stanford
Gain Stage Offset

- Input referred converter offset – usually no problem
- Equivalent sub-ADC offset - easily accommodated through redundancy
Gain Errors

\[ D_{out} = V_{in,ADC} + \varepsilon_{q1} \left( 1 - \frac{G_1 + \Delta}{G_{d1} + \Delta} \right) + \ldots + \frac{\varepsilon_{qn}}{(G_{d1} + \Delta) \prod_{j=2}^{n-1} G_{dj}} \]
Gain Errors

- 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
Digital Gain Calibration

\[ V_{res1} = G \cdot (V_{in} - V_{DAC}) \]

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

\[ V_{res1}^{(1)} = G \cdot \left( V_{in} - V_{ref} / 2 \right) \]

\[ D_{back}^{(1)} = G \cdot \frac{V_{in} - V_{ref} / 2}{V_{ref}} \rightarrow store \]
Calibration – Step 2

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

\[ D_{\text{back}}^{(2)} = G \cdot \left( \frac{V_{\text{in}} - 0}{V_{\text{ref}}} \right) \rightarrow \text{store} \]
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
Gain Element Sensitivity

- Highest sensitivity to gain errors in front-end stages

\[
D_{\text{out}} = V_{\text{in,ADC}} + \varepsilon_{q1} \left(1 - \frac{G_1}{G_{d1}}\right) + \frac{\varepsilon_{q2}}{G_{d1}} \left(1 - \frac{G_2}{G_{d2}}\right) + \ldots + \frac{\varepsilon_{q(n-1)}}{\prod_{j=1}^{n-2} G_{dj}} \left(1 - \frac{G_{(n-1)}}{G_{d(n-1)}}\right) + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}
\]
“Accuracy Bootstrapping”

Sub-DAC Errors

- Can be corrected digitally as well
- Essentially same calibration concept as gain errors
  - Step through DAC codes and use backend ADC to measure DAC errors

Penn ESE 568 Fall 2017 - Khanna adapted from Murmann
EE315B, Stanford
Sub-DAC Errors

- Can be corrected digitally as well
- Essentially same calibration concept as gain errors
  - Step through DAC codes and use backend ADC to measure DAC errors
Calibration Hardware

- Digital is "free" and "easier" to build than precise analog circuits...

Combining the Bits

- Example: Three 2-bit stages, no redundancy

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

- Only bit shifts
- No arithmetic circuits needed

\[ D_1 \quad XX \]
\[ D_2 \quad XX \]
\[ D_3 \quad XX \]
\[ D_{out} \quad DDDDDDD \]

\[ B_1 = 2 \]
\[ R_1 = 2 \]

\[ B_2 = 2 \]
\[ R_1 = 2 \]

\[ B_3 = 2 \]

\[ V_{in} \]
Stage 1
MSB

Stage 2

Stage 3
LSB

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

- Example 2: Three 2-bit stages, one bit redundancy in stages 1 and 2 (6-bit aggregate ADC resolution)
Combining the Bits

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

- Bits overlap
- Need adders

\[ D_{out} \quad \text{DDDDDDDD} \]

Stage 1: \( B_1 = 3 \), \( R_1 = 2 \)

Stage 2: \( B_2 = 3 \), \( R_2 = 2 \)

Stage 3: \( B_3 = 2 \)
Combining the Bits

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

<table>
<thead>
<tr>
<th>D_1</th>
<th>001</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_2</td>
<td>111</td>
</tr>
<tr>
<td>D_3</td>
<td>10</td>
</tr>
</tbody>
</table>

\[ D_{out} = 011000 \]

- Bits overlap
- Need adders

B_1 = 3
R_1 = 2

B_2 = 3
R_2 = 2

B_3 = 2

Stage 1
HADD
Stage 2
FADD
Stage 3
HADD

V_{in} \rightarrow Stage 1 \rightarrow Stage 2 \rightarrow Stage 3 \rightarrow D_{out[5:0]}
Big Ideas

- Pipeline ADCs
  - Used for moderate resolution
  - Higher speed, but includes latency
- Modeled as first stage with sub-ADC
  - Stage bit resolution, Stage gain, # of stages - design parameters
- Non-idealities can be mostly digitally calibrated
- Next time, CMOS implementations
Admin

- Project
  - Design T/H
  - Due 11/17

- Start with a single ended simple track and hold
  - Lec 10 – Charge redistribution T/H
  - Get functionality and understand waveforms
  - Measure performance
  - Optimize from there
    - Sizing transistors and caps
    - Improving opamp
    - Using flip-around topology
    - Differential topology