Mixed-Signal-Electronics

PD Dr.-Ing. Stephan Henzler
# Binary Number Representation

<table>
<thead>
<tr>
<th>#</th>
<th># normalized</th>
<th>Sign magnitude</th>
<th>1’s complement</th>
<th>2’s complement</th>
<th>Offset binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>+7</td>
<td>+7/8</td>
<td>0111</td>
<td>0111</td>
<td>0111</td>
<td>1111</td>
</tr>
<tr>
<td>+6</td>
<td>+6/8</td>
<td>0110</td>
<td>0110</td>
<td>0110</td>
<td>1110</td>
</tr>
<tr>
<td>+5</td>
<td>+5/8</td>
<td>0101</td>
<td>0101</td>
<td>0101</td>
<td>1101</td>
</tr>
<tr>
<td>+4</td>
<td>+4/8</td>
<td>0100</td>
<td>0100</td>
<td>0100</td>
<td>1100</td>
</tr>
<tr>
<td>+3</td>
<td>+3/8</td>
<td>0011</td>
<td>0011</td>
<td>0011</td>
<td>1011</td>
</tr>
<tr>
<td>+2</td>
<td>+2/8</td>
<td>0010</td>
<td>0010</td>
<td>0010</td>
<td>1010</td>
</tr>
<tr>
<td>+1</td>
<td>+1/8</td>
<td>0001</td>
<td>0001</td>
<td>0001</td>
<td>1001</td>
</tr>
<tr>
<td>+0</td>
<td>0</td>
<td>0000</td>
<td>0000</td>
<td>0000</td>
<td>1000</td>
</tr>
<tr>
<td>-0</td>
<td>0</td>
<td>1000</td>
<td>1111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1/8</td>
<td>1001</td>
<td>1110</td>
<td>1111</td>
<td>0111</td>
</tr>
<tr>
<td>-2</td>
<td>-2/8</td>
<td>1010</td>
<td>1101</td>
<td>1110</td>
<td>0110</td>
</tr>
<tr>
<td>-3</td>
<td>-3/8</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>0101</td>
</tr>
<tr>
<td>-4</td>
<td>-4/8</td>
<td>1100</td>
<td>1011</td>
<td>1100</td>
<td>0100</td>
</tr>
<tr>
<td>-5</td>
<td>-5/8</td>
<td>1101</td>
<td>1010</td>
<td>1011</td>
<td>0011</td>
</tr>
<tr>
<td>-6</td>
<td>-6/8</td>
<td>1110</td>
<td>1001</td>
<td>1010</td>
<td>0010</td>
</tr>
<tr>
<td>-7</td>
<td>-7/8</td>
<td>1111</td>
<td>1000</td>
<td>1001</td>
<td>0001</td>
</tr>
<tr>
<td>-8</td>
<td>-8/8</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>0000</td>
</tr>
</tbody>
</table>
Chapter 5

Nyquist Rate
Digital-to-Analog Converters
Nyquist Rate Digital-to-Analog Converters

Basic Idea:
• Generate all possible voltages which are possible according to eq. 1
• Use switches to connect the voltage selected by $B_{in}$ to the output

$$V_{out} = V_{ref} B_{in} = V_{ref} \left( b_1 2^{-1} + b_2 2^{-2} + \ldots + b_N 2^{-N} \right)$$
3-Bit Resistor String Converter

- Generate all possible voltages with resistive voltage divider
- Switches = NMOS transistors

$$V_{ref} < \frac{1}{1 - 2^{-N}} (V_{DD} - V_{th})$$

- Transmission gates enable
  - higher voltage range
  - but higher parasitic cap, area
    (layout more complicated)
- Buffer experiences high input voltage variation
- Slow due to buffer and analog mux
- How fast does the DAC settle
Elmore Delay

Prerequisites:

– one input only
– caps between network node and ground only
– no resistive loops

There is exactly one resistive path from a network node \( i \) to the input \( s \).

The sum of all resistances along this path is the path resistance \( R_{ii} \), e.g. \( R_{44} = R_4 + R_3 + R_1 \).
The shared path resistance $R_{ik}$ is the sum of all resistances along the joint sub-path of the two paths $s \to i$ and $s \to k$.

**Example:** $R_{i4} = R_1 + R_3$
Elmore Delay (cont): Delay Approximation

Elmore delay: \[ \tau_D = \sum_{k=1}^{N} C_k R_{i,k} \]

First order approximation of the delay after which a voltage step at the input \( s \) can be observed at the output \( i \).
Elmore delay: $\tau_D = \sum_{k=1}^{N} C_k R_{ik}$

Useful for
- Estimation of wire delay
- Estimation of DAC settling time
- …
Resistor String Converter
Resistor String Converter
(with pass-gate decoder)

\[ B_{in} = b_1 2^{-1} + b_2 2^{-4} + b_3 2^{-3} \]
Delay Comparison

![Graph showing normalized delay against number of bits for string and tree structures.](image)
Folded Resistor String Converter

Combine the advantages of both converters:
(low effort for decoder, small load cap.)

Access scheme as in memories:
MSBs select row
LSBs select column

$2\sqrt{2^N}$ transistors at output bus
all bitlines are charged
Multi-Stage Resistor String Converter

- Subdivide voltage range in coarse sub-intervals first
- Copy the respective voltage interval
- Fine interpolation of the copied interval

\[ \frac{6}{8} \cdot \frac{1}{2} V_{ref} + \frac{3}{8} \cdot V_{ref} = \frac{30}{64} V_{ref} \quad 6 \text{- Bit} \]

- if opamps match the converter is monotonic
- less resistors \( 2 \cdot 2^{N/2} \)
- reduced area and power
Binary Weighted Current Mode Converters

**Until now:** All possible voltages are generated, 1 out of $2^N$ voltages is copied to the output

$$I_F = b_1 I + \frac{1}{2} b_2 I + \frac{1}{4} b_3 I + \frac{1}{8} b_4 I$$

**Now:**
- Current mode, i.e. currents are generated, superposed and then converted into the output voltage
- Input word is already binary $\rightarrow$ generated binary weighted currents and superpose them into a current that corresponds to the input word.

\[
\frac{I_{\text{max}}}{I_{\text{min}}} = 2^{N-1}
\]

scale switches
Monotonicity in Binary Weighted DACs

- Binary weighted converters are not necessarily monotonic
- Example:

\[
\begin{array}{cccc}
1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
0 & 1 & 1 & 1 \rightarrow \frac{7}{8} \\
1 & 0 & 0 & 0 \rightarrow 1
\end{array}
\quad
\begin{array}{cccc}
\frac{6}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
0 & 1 & 1 & 1 \rightarrow \frac{7}{8} \\
1 & 0 & 0 & 0 \rightarrow \frac{6}{8}
\end{array}
\]
Glitches in Binary Weighted DACs

- Different delays in the control logic of the switches causes voltage spikes, i.e. glitches
Implementation of Binary Weighted DAC

How can we generate binary weighted currents easily?

\[
I = I_1 + I_2 + I_3 + I_4
\]

\[
V_{out} = -R_F \cdot I
\]

\[
= \frac{V_{ref} R_F}{R} \left( 2^{-1} b_1 2^{-2} b_2 + 2^{-3} b_3 + 2^{-4} b_4 \right)
\]

\[
I'_1 = -V_{ref} \cdot \frac{1}{2R}
\]

\[
I'_2 = -V_{ref} \cdot \frac{1}{4R}
\]

\[
I'_3 = -V_{ref} \cdot \frac{1}{8R}
\]

\[
I'_4 = -V_{ref} \cdot \frac{1}{16R}
\]

\[
I_1 = -V_{ref} \cdot \frac{1}{2R} \cdot b_1
\]

\[
I_2 = -V_{ref} \cdot \frac{1}{4R} \cdot b_2
\]

\[
I_3 = -V_{ref} \cdot \frac{1}{8R} \cdot b_3
\]

\[
I_4 = -V_{ref} \cdot \frac{1}{16R} \cdot b_4
\]
Implementation of Binary Weighted DAC

- For $b_i = 0$ the same current flows, not to VGND but to AGND
- 30 unit resistors (in binary weighted array)
- Not necessarily monotonic
- Glitches, if switches do not switch simultaneously

$$\frac{I_{max}}{I_{min}} = \frac{R_{max}}{R_{min}} = 8 = 2^{N-1}$$
High Resolution DAC II

- High number of bits
  - large area
  - matching difficult if MSB/LSB ratio is large (currents, resistors)

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

\[ /2^N \]

MSBs

LSBs
High Resolution DAC

- High number of bits
  - large area
  - matching difficult if MSB/LSB ratio is large (currents, resistors)
Implementation of Binary Weighted DAC (with improved resistor ratio)

\[ V_{out} = V_{ref} \cdot \frac{R_F}{R} \left( 2^{-1} b_1 + 2^{-2} b_2 \right) + \frac{1}{4} V_{ref} \cdot \frac{R_F}{R} \left( 2^{-1} b_3 + 2^{-2} b_4 \right) \]

scaling the reference voltage
Implementation of Binary Weighted DAC (with improved resistor ratio)

- $\frac{R_{\text{max}}}{R_{\text{min}}} = 2$ (reduced)

- $\frac{I_{\text{max}}}{I_{\text{min}}} = 8$

- 19 unit resistors
R-2R-Ladder Network

\[ R'_{4} = 2R \]
\[ R_{4} = 2R \parallel 2R = R \]
\[ R'_{3} = 2R \]
\[ R_{3} = 2R \parallel 2R = R \]
\[ R'_{2} = 2R \]
\[ R_{2} = 2R \parallel 2R = R \]
\[ ... \]

\[ R'_{i} = 2R \]
\[ R_{i} = R \]
Implementation of Binary Weighted DAC (with R-2R-Ladder)

- Take R-2R ladder and replace AGND by a virtual ground in order to collect binary weighted currents
- Insert switches (such that node potential is not changed)
Implementation of Binary Weighted DAC (with R-2R-current divider)

- R-2R ladder as current divider
Switched Capacitor Amplifier (without output reset)

\[ v_{out}[n] = -\left(\frac{C_1}{C_2}\right)v_{in}[n] \]
SC-Amplifier with Controllable Capacitors

- Various variants possible
- Gain is altered according to binary input → multiplying DAC (M-DAC)

\[ V_{out} = k(B_{in})V_{in} \]
## Thermometer Code Converters
(method to force monotonicity)

<table>
<thead>
<tr>
<th>#</th>
<th>binary</th>
<th>thermometer code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b_1$</td>
<td>$d_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Thermometer Code Converters (method to force monotonicity)
Thermometer Code Converters (method to force monotonicity)

Top capacitors are connected to ground

Bottom capacitors are connected to $V_{\text{ref}}$
Hybrid Converter Architectures

- Thermometer coding MSBs
- Binary weighted for LSBs

[Diagram of hybrid converter architectures with annotations and symbols]

Current driver

4-bit binary LSB segment
Differential Current Steering DAC

\[
I_+ = I \left( b_1 + \frac{1}{2} b_2 + \frac{3}{4} b_3 \right)
\]
\[
I_- = I \left( \bar{b}_1 + \frac{1}{2} \bar{b}_2 + \frac{3}{4} \bar{b}_3 \right)
\]

\[
V_{OD} = \frac{1}{2} R \left( -I_- + I_+ \right)
= R I_2 - R I \left( 1 - 2^{-n} \right)
\]
Charge Scaling DAC

- Compatibel with switched capacitor circuits
- Principle: Divide charge binarily
- All caps discharged during $\phi_1$ (reset phase, no valid output)

$$C_{\text{tot}} = \sum_{i=1}^{n} C_i = C$$

Phase 1: $Q_{\text{tot}} = 0$
$V_{\text{out}} = 0$

$$\frac{C}{2} (V_X - b_1 V_{\text{ref}}) + \frac{C}{4} (V_X - b_2 V_{\text{ref}}) + ... = Q_{\text{tot}} = 0$$
$$C_{\text{tot}} \cdot V_X - \sum C \cdot 2^{-i} V_{\text{ref}} \cdot b_i = 0$$

$$V_{\text{out}}^+ = V_{\text{ref}} - \sum 2^{-i} \cdot b_i$$