Advantages: - Easier to build higher resolution DACs, complexity scales linearly with N instead of a $2^N$.

Disadvantages: - Can be subject to glitches if bits change at different times.

Consider transition from 100 to 011 in a 3-bit counter. - If $b_2$ changes before $b_3$, it will have a glitch at output until output is temporarily 000.

1. Thermometric Code Converter

Thermometric code converters solve the glitch problem of the previous type at the expense of some added complexity.

- Thermometric codes use $2^n-1$ bits to represent $2^N$ levels, as opposed to normal binary representation, which only requires $N$ levels.

<table>
<thead>
<tr>
<th></th>
<th>Decimal</th>
<th>Binary</th>
<th>Thermometric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>111</td>
<td>111</td>
</tr>
</tbody>
</table>

- Eliminates possibility of glitches in moving between adjacent values.
Example: Binary-Weighted Charge Redistribution DAC can be changed for thermometer coding.

- Not too much work in area due to passives. (7C for both).

Hybrid Converters

- Some DACs use combinations of the aforementioned techniques.

  E.g. Thermometer coding for MSBs while gliding will have the largest impact; binary weighting for LSBs to save on space & complexity.

Nyquist Rate A/D Converters

- The Nyquist rate A/D converters we will discuss fall into three categories:
  1. Low speed, high accuracy
  2. Medium speed, medium accuracy
  3. High speed, low accuracy

- Let's discuss one example of each:

  1. Low Speed: Dual-Slope Integrating Converter

- We discussed these in ECE 2110, will review it here briefly.

Operator: Phase 1 - $S_2$ resets $V_x$ to $0V$, $S_1$ connects to $-V_{in}$
- Integrator increases output $V_x$ at slope of $\frac{V_{in}}{RC}$
Phase II: After fixed # of clock cycles (determined by counter), control logic switches $S_1$ to $V_{ref}$.

- $V_x$ now discharges with a slope of $\frac{-V_{ref}}{RC}$
- Comparator triggers when $V_x$ reaches zero, time $T_2$ allows
  $V_1$ to determine input voltage

& Converse that: Proportional to $2^N$.

Assume we are in Phase II after $2^N$ clock cycles: $V_x(T_1) = 2^N \cdot T_1 \cdot \frac{V_{in}}{RC}$, where $T_1 = 2^N \cdot T_{comp}$.

Time to discharge is $T_2 = V_x(T_1)$ slope during Phase II

Counter value: $\Rightarrow \frac{T_2}{T_1} = \frac{V_{in}}{V_{ref}} = B_{out}$ (no dependence on $RC$)

(2) Medium Speed: Successive Approximation Converter

- Uses a binary search algorithm to successively determine each of the $N$ bits in the output word.

General idea:

Operator:
- Start with all $b_1 \cdots b_N = 0$, DAC output is 0 V.
- Control logic increments $b_i$ to 1, $V_{DAC} = \frac{V_{ref}}{2}$
- Comparator determining if $b_i$ would remain high or be low.
- Process repeats for each of $N$ bits.
Example: 3-bit, \( V_{ref} \)

\[ V_{in} > V_{ref} \rightarrow b_1 = 1 \]
\[ V_{in} < V_{ref} \rightarrow b_2 = 0 \]
\[ V_{in} > V_{ref} \rightarrow b_3 = 1 \]

Output: \( B_{out} = 101 \)

- Conversion time is proportional to \( N \).

\( \square \) DAC can introduce inaccuracies.

- Text gives a 5-C based implementation that does not require an explicit DAC.

3. High Speed: Flash Converter

- Requires a lot of hardware (\( 2^n \) comparators) but very fast, can do a conversion in one clock cycle.

Example 2-bit flash:

Example: \( V_{ref} = 1 \) V, taps in resistor chain are 0.125, 0.375, 0.625, 0.875.

- Let \( V_{in} = 0.1 \) V, all comp. O/Ps are "1", all hard inputs are 1
  \( \Rightarrow B_{out} = 00 \)

- Let \( V_{in} = 0.4 \) V, top 2 comp. O/Ps are "1", bottom two are "0"
  \( \Rightarrow \) Middle NAND output is 0, \( B_{out} = 10 \)

- Conversion time is proportional to a single clock cycle.
Flash converters are very fast. Their main drawback is the high hardware complexity.

- This can be reduced with little speed penalty if we can tolerate multiple clock cycles instead of just one.

**Two-Step A/D Converter**

Example: 8-bit converter.

![Diagram of 8-bit two-step A/D converter]

- Conversion now takes at least 2 clock cycles, but we have reduced hardware requirements from $2^8 = 256$ comparators to just $4 + 4 = 8$ comparators.

- The two-step converter is an example of a more general type of converter that can be used to get good accuracy and high speed at the expense of some latency.

**Pipelined A/D Converter**

- General idea: process each sample in a pipeline, determining one bit at each stage.

![Diagram of pipelined A/D converter]

- Latency of $N$ clock cycles; can do a conversion on each clock cycle.

**Oversampling Data Converters** [Ch. 14]

- We will focus on ADC implementations, but the concepts are equally applicable for DACs.

- To get the full benefit of oversampling data converters, we must employ "noise shaping," but first we will discuss O.S.C. without noise shaping.
Quantization Noise Modeling

Previously we saw that the output of a quantizer such as an A/D can be modeled as the original signal plus some added quantization noise:

\[ e(n) \leftarrow \text{quant. noise} \]

\[ X(n) \rightarrow y(n) \equiv X(n) \rightarrow \oplus \rightarrow y(n) \]

- We saw that the value of \( e(n) \) will range from \( -\frac{V_{\text{Ref}}}{2} \) to \( +\frac{V_{\text{Ref}}}{2} \).

- If \( x(n) \) is changing quickly, we can assume that \( e(n) \) will be uniformly distributed between these values.

- In this case, the power of \( e(n) \) will be \( \frac{V_{\text{Ref}}^2}{12} \) as calculated before.

- This power is independent of sampling freq. \( (f_s) \), and it's power will have a "white" distribution (Spectral density):

\[ \text{Calculate height: } k_x^2 f_s = \frac{V_{\text{Ref}}^2}{12} \quad \text{(total power)} \]

\[ \Rightarrow k_x = \frac{V_{\text{Ref}}}{\sqrt{12} \cdot f_s} \]

The Magic of Over-Sampling

- Let’s define the Over-Sampling Ratio (OSR) as \( \text{OSR} = \frac{f_s}{2f_0} \)

where \( f_0 \) is the highest freq. in the sampled signal and \( 2f_0 \) is the Nyquist rate.

- First, let's assume 2x over-sampling.

Output Spectrum:

- If we now apply a brickwall filter in the digital domain, we can eliminate \( \frac{1}{2} \) of the quantization noise power.

- so we would expect a 3 dB improvement in SNR.
Now, if we double the OSR again, we eliminate another half of the quant. noise, leading to another 3-dB improvement.

We can now modify our original SNR equation to include the effects of oversampling:

\[
SNR = 6.02 \cdot N + 1.76 + 10 \cdot \log (OSR)
\]

\[
\frac{1}{3} \text{ dB improvement for each doubling of OSR.}
\]

So, for any number of bits of resolution (N), we can get an arbitrarily high SNR by using a high enough OSR.

E.g. Can give a 4-bit converter the effective accuracy of a 5-bit converter by using an OSR of 4.