**Bandwidth Constraints for Integer-N Synthesizers**

- **PFD output has a periodicity of** $1/T$
  - $1/T =$ reference frequency
- **Loop filter must have a bandwidth $<< 1/T$**
  - PFD output pulses must be filtered out and average value extracted

**Closed loop PLL bandwidth often chosen to be a factor of ten lower than** $1/T$
**Bandwidth Versus Frequency Resolution**

- Frequency resolution set by reference frequency \((1/T)\)
  - Higher resolution achieved by lowering \(1/T\)

**Diagram:**
- PFD Loop
- Filter
- Divider
- Loop Filter Bandwidth \(<\!\!1/T\!\!\rangle

**Equations:**
- Ref(t) → out(t)
- \((1/T = 20 \text{ MHz})\)
- N[k]"
Increasing Resolution in Integer-N Synthesizers

- Use a reference divider to achieve lower $1/T$
  - Leads to a low PLL bandwidth ( < 20 kHz here )
The Issue of Noise

- Lower $1/T$ leads to higher divide value
  - Increases PFD noise at synthesizer output
Modeling PFD Noise Multiplication

- Influence of PFD noise seen in model from Lecture 16
  - PFD spectral density multiplied by $N^2$ before influencing PLL output phase noise

High divide values → high phase noise at low frequencies
**Dual-Loop Frequency Synthesizer**

- **Overall synthesizer output**

\[
\text{out}(t) = \cos(w_1 t) \cos(w_2 t) + \sin(w_1 t) \sin(w_2 t)
\]

- **From trigonometry:** \(\cos(A-B) = \cos A \cos B + \sin A \sin B\)

\[\Rightarrow \text{out}(t) = \cos((w_1 - w_2) t)\]
Advantage #1: Avoids Large Divide Values

Choose top synthesizer to provide coarse tuning and bottom synthesizer to provide fine tuning

- Choose \( w_1 \) to be high in frequency
  - Set \( \text{ref}_1 \) to be high to avoid large \( N \) low resolution
- Choose \( w_2 \) to be low in frequency
  - Allows \( \text{ref}_2 \) to be low without large \( M \) high resolution
Advantage #2: Provides Suppression of VCO Noise

- Top VCO has much more phase noise than bottom VCO due to its much higher operating frequency
  - Suppress top VCO noise by choosing a high PLL bandwidth for top synthesizer
    - High PLL bandwidth possible since ref₁ is high
Alternate Dual-Loop Architecture

- Calculation of output frequency

\[ y(t) = \cos((w_1 - w_2)t) \]

\[ \Rightarrow Nw_{ref_1} = w_1 - w_2 \]

\[ \Rightarrow \quad \text{out}(t) = \cos((Nw_{ref_1} + w_2)t) \]
Advantage of Alternate Dual-Loop Architecture

- Issue: a practical single-sideband mixer implementation will produce a spur at frequency $w_1 + w_2$
- PLL bandwidth of top synthesizer can be chosen low enough to suppress the single-sideband spur
  - Negative: lower suppression of top VCO noise
**Direct Digital Synthesis (DDS)**

- Encode sine-wave values in a ROM
- Create sine-wave output by indexing through ROM and feeding its output to a DAC and lowpass filter
  - Speed at which you index through ROM sets frequency of output sine-wave
    - Speed of indexing is set by increment value on counter (which is easily adjustable in a digital manner)
Pros and Cons of Direct Digital Synthesis

- **Advantages**
  - Very fast adjustment of frequency
  - Very high resolution can be achieved
  - Highly digital approach

- **Disadvantages**
  - Difficult to achieve high frequencies
  - Difficult to achieve low noise
  - Power hungry and complex
Hybrid Approach

- Use DDS to create a finely adjustable reference frequency
- Use integer-N synthesizer to multiply the DDS output frequency to much higher values

**Issues**
- Noise of DDS is multiplied by $N^2$
- Complex and power hungry
Fractional-N Frequency Synthesizers

- **Break constraint that divide value be integer**
  - Dither divide value dynamically to achieve fractional values
  - Frequency resolution is now arbitrary regardless of $1/T$
- **Want high $1/T$ to allow a high PLL bandwidth**

![Fractional-N Frequency Synthesizer Diagram](image)

- $N_{sd}[k]$
- $ref(t)$: reference signal
- $out(t)$: output signal
- $1/T = 20$ MHz
- Frequency resolution $<< 1/T$
Classical Fractional-N Synthesizer Architecture

- Use an accumulator to perform dithering operation
  - Fractional input value fed into accumulator
  - Carry out bit of accumulator fed into divider

\[ N_{sd}[k] = N + \text{frac}[k] \]
Accumulator Operation

- Carry out bit is asserted when accumulator residue reaches or surpasses its full scale value
  - Accumulator residue increments by input fractional value each clock cycle
Fractional-N Synthesizer Signals with $N = 4.25$

- Divide value set at $N = 4$ most of the time
  - Resulting frequency offset causes phase error to accumulate
  - Reset phase error by “swallowing” a VCO cycle
    - Achieved by dividing by 5 every 4 reference cycles
The Issue of Spurious Tones

- **PFD error is periodic**
  - Note that actual PFD waveform is series of pulses – the sawtooth waveform represents pulse width values over time

- **Periodic error signal creates spurious tones in synthesizer output**
  - Ruins noise performance of synthesizer

\[ N_{sd}[k] = N + \text{frac}[k] \]
**The Phase Interpolation Technique**

- Phase error due to fractional technique is predicted by the instantaneous residue of the accumulator
  - Cancel out phase error based on accumulator residue
The Problem With Phase Interpolation

- Gain matching between PFD error and scaled D/A output must be extremely precise
  - Any mismatch will lead to spurious tones at PLL output
Is There a Better Way?
A Better Dithering Method: Sigma-Delta Modulation

Sigma-Delta dithers in a manner such that resulting quantization noise is “shaped” to high frequencies.
Linearized Model of Sigma-Delta Modulator

- Composed of two transfer functions relating input and noise to output
  - Signal transfer function (STF)
    - Filters input (generally undesirable)
  - Noise transfer function (NTF)
    - Filters (i.e., shapes) noise that is assumed to be white
Example: Cutler Sigma-Delta Topology

- Output is quantized in a multi-level fashion
- Error signal, $e[k]$, represents the quantization error
- Filtered version of quantization error is fed back to input
  - $H(z)$ is typically a highpass filter whose first tap value is 1
    - i.e., $H(z) = 1 + a_1 z^{-1} + a_2 z^{-2} \ldots$
  - $H(z) - 1$ therefore has a first tap value of 0
    - Feedback needs to have delay to be realizable
Represent quantizer block as a summing junction in which $r[k]$ represents quantization error

- Note:

$$e[k] = y[k] - u[k] = (u[k] + r[k]) - u[k] = r[k]$$

- It is assumed that $r[k]$ has statistics similar to white noise

- This is a key assumption for modeling – often not true!
Calculation of Signal and Noise Transfer Functions

- Calculate using Z-transform of signals in linearized model

\[ Y(z) = U(z) + R(z) \]
\[ = X(z) + (H(z) - 1)E(z) + R(z) \]
\[ = X(z) + (H(z) - 1)R(z) + R(z) \]
\[ = X(z) + H(z)R(z) \]

- NTF: \( H_n(z) = H(z) \)
- STF: \( H_s(z) = 1 \)
A Common Choice for $H(z)$

\[ H(z) = (1 - z^{-1})^m \]

\[ \Rightarrow |H(e^{j2\pi f T})| = |(1 - e^{-j2\pi f T})^m| \]
**Example: First Order Sigma-Delta Modulator**

- Choose NTF to be

\[ H_n(z) = H(z) = 1 - z^{-1} \]

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
Example: Second Order Sigma-Delta Modulator

- Choose NTF to be

\[ H_n(z) = H(z) = (1 - z^{-1})^2 \]

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
Example: Third Order Sigma-Delta Modulator

- Choose NTF to be

\[ H_n(z) = H(z) = (1 - z^{-1})^3 \]

- Plot of output in time and frequency domains with input of

\[ x[k] = 0.5 + 0.25 \sin \left( \frac{2\pi}{100} k \right) \]
Observations

- Low order Sigma-Delta modulators do not appear to produce “shaped” noise very well
  - Reason: low order feedback does not properly “scramble” relationship between input and quantization noise
    - Quantization noise, $r[k]$, fails to be white
- Higher order Sigma-Delta modulators provide much better noise shaping with fewer spurs
  - Reason: higher order feedback filter provides a much more complex interaction between input and quantization noise
Warning: Higher Order Modulators May Still Have Tones

- Quantization noise, $r[k]$, is best whitened when a “sufficiently exciting” input is applied to the modulator
  - Varying input and high order helps to “scramble” interaction between input and quantization noise
- Worst input for tone generation are DC signals that are rational with a low valued denominator
  - Examples (third order modulator):

\[
x[k] = 0.1 \\
x[k] = 0.1 + \frac{1}{1024}
\]
Cascaded Sigma-Delta Modulator Topologies

- Achieve higher order shaping by cascading low order sections and properly combining their outputs
- Advantage over single loop approach
  - Allows pipelining to be applied to implementation
    - High speed or low power applications benefit
- Disadvantages
  - Relies on precise matching requirements when combining outputs (not a problem for digital implementations)
  - Requires multi-bit quantizer (single loop does not)
MASH topology

- Cascade first order sections
- Combine their outputs after they have passed through digital differentiators
Calculation of STF and NTF for MASH topology (Step 1)

- Individual output signals of each first order modulator

\[
\begin{align*}
y_1(z) &= x(z) - (1 - z^{-1})r_1(z) \\
y_2(z) &= r_1(z) - (1 - z^{-1})r_2(z) - (1 - z^{-1})r_3(z) \\
y_3(z) &= (1 - z^{-1})^2 y_2(z)
\end{align*}
\]

- Addition of filtered outputs

\[
x(z) - (1 - z^{-1})^3 r_3(z)
\]
Calculation of STF and NTF for MASH topology (Step 1)

- Overall modulator behavior

\[ y(z) = x(z) - (1 - z^{-1})^3 r_3(z) \]

- STF: \( H_s(z) = 1 \)
- NTF: \( H_n(z) = (1 - z^{-1})^3 \)
Use Sigma-Delta modulator rather than accumulator to perform dithering operation
- Achieves much better spurious performance than classical fractional-N approach

Riley et. al., JSSC, May 1993
Background: The Need for A Better PLL Model

- **Classical PLL model**
  - Predicts impact of PFD and VCO referred noise sources
  - Does not allow straightforward modeling of impact due to divide value variations
    - This is a problem when using fractional-N approach
A PLL Model Accommodating Divide Value Variations

Parameterized Version of New Model

### Noise

\[ e_{\text{spur}}(t) \]

\[ I_{\text{cpn}}(t) \]

\[ \Phi_{\text{jit}}[k] \]

\[ \Phi_{\text{npfd}}(t) \]

\[ \Phi_{\text{nvco}}(t) \]

### Divide Value Variation

\[ n[k] \]

\[ \Phi_{\text{vn}}(t) \]

\[ \Phi_{\text{out}}(t) \]

Alternate Representation

\[ 2\pi \frac{z^{-1}}{1 - z^{-1}} \]

\[ \Phi_{\text{d}}(t) \]

\[ \Phi_{\text{c}}(t) \]

\[ \Phi_{\text{vn}}(t) \]

\[ \Phi_{\text{out}}(t) \]

\[ G(f) \]

\[ F_{\text{c}}(t) \]

D/A and Filter

Freq. → Phase

\[ -20 \text{ dB/dec} \]
**Spectral Density Calculations**

- **Case (a):**
  \[ S_y(f) = |H(f)|^2 S_x(f) \]

- **Case (b):**
  \[ S_y(e^{j2\pi fT}) = |H(e^{j2\pi fT})|^2 S_x(e^{j2\pi fT}) \]

- **Case (c):**
  \[ S_y(f) = \frac{1}{T} |H(f)|^2 S_x(e^{j2\pi fT}) \]
Example: Calculate Impact of Ref/Divider Jitter (Step 1)

- Assume jitter is white
  - i.e., each jitter value independent of values at other time instants
- Calculate spectra for discrete-time input and output
  - Apply case (b) calculation

\[
S_{\Delta t_{jit}}(e^{j2\pi fT}) = \beta^2 \Rightarrow S_{\Phi_{jit}}(e^{j2\pi fT}) = \left| \frac{2\pi}{T} \right|^2 \beta^2
\]
Example: Calculate Impact of Ref/Divider Jitter (Step 2)

- **Compute impact on output phase noise of synthesizer**
  - We now apply case (c) calculation

\[
S_{\Phi_n}(f) = \frac{1}{T} \left| TN_{\text{nom}} G(f) \right|^2 S_{\Phi_{\text{jit}}} \left(e^{j2\pi fT} \right)
\]

\[
= \frac{1}{T} \left| TN_{\text{nom}} G(f) \right|^2 \left| \frac{2\pi}{T} \right|^2 \beta^2
\]

- Note that \( G(f) = 1 \) at DC
Now Consider Impact of Divide Value Variations

\[ \Phi_{\text{njt}}[k] \rightarrow \alpha \frac{T}{2\pi} \]

\[ e_{\text{spur}}(t) \]

\[ J_{\text{cpn}}(t) \rightarrow 1 \]

\[ \Phi_{\text{vn}}(t) \rightarrow 2\pi N_{\text{nom}} \cdot G(f) \]

\[ \Phi_{\text{vn}}(t) \rightarrow 1 - G(f) \]

\[ \Phi_{\text{npfd}}(t) \rightarrow \Phi_{\text{vn}}(t) \]

\[ \Phi_{\text{out}}(t) \]

\[ \Phi_{\text{n}[k]} \rightarrow \frac{2\pi}{\alpha} \frac{z^{-1}}{1 - z^{-1}} \]

\[ \Phi_{\text{d}}(t) \rightarrow \Phi_{\text{c}}(t) \]

\[ \Phi_{\text{out}}(t) \]

\[ G(f) \rightarrow F_{\text{c}}(t) \rightarrow \frac{1}{jf} \]

\[ \Phi_{\text{c}}(t) \]

\[ \Phi_{\text{n}[k]} \rightarrow \text{D/A and Filter} \rightarrow \text{Freq. \rightarrow Phase} \]

\[ \text{PFD-referred Noise} \]

\[ S_{\Phi_{\text{vn}}}(f) \]

\[ -20 \text{ dB/dec} \]
Divider Impact For Classical Vs Fractional-N Approaches

Classical Synthesizer

\[ \frac{1}{T} \]
\[ n(t) \rightarrow n[k] \rightarrow G(f) \rightarrow F_{out}(t) \]
D/A and Filter

Fractional-N Synthesizer

\[ \frac{1}{T} \]
\[ n_{sd}(t) \rightarrow n_{sd}[k] \rightarrow \text{Dithering Modulator} \rightarrow n[k] \rightarrow G(f) \rightarrow F_{out}(t) \]
D/A and Filter

- Note: 1/T block represents sampler (to go from CT to DT)
Focus on Sigma-Delta Frequency Synthesizer

- Divide value can take on fractional values
  - Virtually arbitrary resolution is possible
- PLL dynamics act like lowpass filter to remove much of the quantization noise
Quantifying the Quantization Noise Impact

- Calculate by simply attaching Sigma-Delta model
  - We see that quantization noise is integrated and then lowpass filtered before impacting PLL output
A Well Designed Sigma-Delta Synthesizer

- Order of $G(f)$ is set to equal to the Sigma-Delta order
  - Sigma-Delta noise falls at -20 dB/dec above $G(f)$ bandwidth
- Bandwidth of $G(f)$ is set low enough such that synthesizer noise is dominated by intrinsic PFD and VCO noise

$f_0 = 84$ kHz

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kHz</td>
</tr>
<tr>
<td>100 kHz</td>
</tr>
<tr>
<td>1 MHz</td>
</tr>
<tr>
<td>10 MHz</td>
</tr>
</tbody>
</table>

Spectral Density (dBc/Hz)
**Impact of Increased PLL Bandwidth**

- Allows more PFD noise to pass through
- Allows more Sigma-Delta noise to pass through
- Increases suppression of VCO noise
**Impact of Increased Sigma-Delta Order**

- PFD and VCO noise unaffected
- Sigma-Delta noise no longer attenuated by $G(f)$ such that a -20 dB/dec slope is achieved above its bandwidth