Sigma-Delta
Fractional-N Frequency Synthesis

Scott Meninger
Michael Perrott
Massachusetts Institute of Technology
June 7, 2004

Copyright © 2004 by Michael H. Perrott
All rights reserved.
Note: Much of this material is taken from MITOpenCourseWare http://ocw.mit.edu Course: 6.976
Outline

- Integer-N synthesis
  - Bandwidth constraints
- Fractional-N synthesis
  - Issue of fractional spurs
- $\Sigma\Delta$ Fractional-N Synthesis
  - Quantization noise impact on the PLL
- Recent developments for lowering the impact of quantization noise
- Conclusions
- Q&A
**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 \(1/T\)
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
Background: Classical Linearized PLL Model

- Classical PLL model
  - Predicts impact of PFD and VCO referred noise sources
  - Does not allow straightforward modeling of impact due to dynamic divide value variations
    - More on this shortly …
Parameterizing in terms of $G(f)$ helps visualize the nature (high-pass or low-pass) and gain of the noise transfer functions.
Parameterized Version of Classical Model

- G(f) represents the PLL closed loop dynamics
- G(f) is low-pass
- Nature of noise transfer very easily seen from the parameterized model
Modeling PFD Noise Multiplication

- PFD spectral density multiplied by \( N^2 \) before influencing PLL output phase noise

High divide values → high phase noise at low frequencies
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
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
The sigma-delta noise shaping analysis assumes a white quantization noise spectrum.

In order to make the input look “sufficiently exciting” a dither signal can be added to it.

Dithering methods are directly taken from sigma-delta ADC and DAC design.

- This makes sense since the synthesizer is really a DAC (digital to phase).
- Most common method is to add a random sequence to the LSB’s of the input.
**Sigma-Delta Frequency Synthesizers**

- 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
Summary: Sources of Phase Noise in $\Sigma\Delta$ Synthesis

- **Charge-pump / Phase Detector / Reference**
  - Low-pass filtered by PLL, dominant at low offset frequencies

- **VCO**
  - High-pass filtered by PLL, dominant at high offset frequencies

- $\Sigma\Delta$ dithered quantization noise
  - Low-pass filtered by PLL, noise/bandwidth tradeoff exists
A quick note on the linearized model

- Non-linearities break the assumptions of the linear model
  - The shaped noise can be “folded down” to lower frequencies due to non-linearities in the synthesizer
    - PFD/Charge-pump design
- This process is best seen through behavioral simulation
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
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
Dual Fractional-N/Integer-N Frequency Synthesizer

ADF4252

3 GHz/1.2GHz
TPC 1. Phase Noise Plot, Lowest Noise Mode, 1.7518 GHz $RF_{OUT}$, 10 MHz PFD Frequency, 200 kHz Channel Step Resolution

TPC 4. Spurious Plot, Lowest Noise Mode, 1.7518 GHz $RF_{OUT}$, 10 MHz PFD Frequency, 200 kHz Channel Step Resolution