Making Better Use of Time in Mixed-Signal Circuits

Michael H. Perrott
Massachusetts Institute of Technology
July 2007

Copyright © 2007 by Michael H. Perrott
All rights reserved.
Motivation

- The world is changing for analog designers
  - Future processes are offering degraded analog device characteristics (lower $g_m r_o$, lower supply voltage, etc.)
- There is pressing need for new mixed-signal circuit architectures
  - Leverage digital circuits to perform analog processing

- An interesting observation: representation of signals as transition times allows digital circuits to perform analog signal processing
  - Phase-locked loop circuits have been the classical benefactor of this observation

Are there new applications and new circuits that can better leverage time as a signal domain?
Just Enough PLL Background …
What is a Phase-Locked Loop (PLL)?

- VCO efficiently provides oscillating waveform with variable frequency
- PLL synchronizes VCO frequency to input reference frequency through feedback
  - Key block is phase detector
    - Realized as digital gates that create pulsed signals
Integer-N Frequency Synthesizers

- Use digital counter structure to divide VCO frequency
  - Constraint: must divide by integer values
- Use PLL to synchronize reference and divider output

Output frequency is digitally controlled

\[ F_{\text{out}} = N \cdot F_{\text{ref}} \]

Sepe and Johnston
US Patent (1968)
**Fractional-N Frequency Synthesizers**

- Dither divide value to achieve fractional divide values
  - PLL loop filter smooths the resulting variations

**Very high frequency resolution is achieved**
The Issue of Quantization Noise

- Limits PLL bandwidth
- Increases linearity requirements of phase detector
Striving for a Better PLL Implementation
Analog Phase Detection

- Pulse width is formed according to phase difference between two signals
- Average of pulsed waveform is applied to VCO input
**Tradeoffs of Analog Approach**

- **Benefit:** average of pulsed output is a continuous, linear function of phase error
- **Issue:** analog loop filter implementation is undesirable
- Digital loop filter: easily achieves long time constants without leakage
- Phase detection: use a time-to-digital converter
- Digitally controlled oscillator: many possibilities

Staszewski et. al., TCAS II, Nov 2003
Classical Time-to-Digital Converter

- Phase error is measured in increments of buffer delays from the reference edge
  - Issues: limited detection resolution, nonlinearity due to mismatch between delay stages
Impact of Limited Resolution and Nonlinearity

- Integer-N PLL: limit cycles
- Fractional-N PLL: undesired fractional spurs

Additional filtering is not effective in fixing these issues
Proposed Approach: A Better Time-to-Digital Converter

- This is a simplified view
  - We will need a few slides to properly explain this …
Consider Measurement of the Period of a Signal

- Use digital logic to count number of oscillator cycles during each input period
  - Assume that oscillator period is much smaller than that of the input
- Note: output count per period is not consistent
  - Depends on starting phase of oscillator within a given measurement period
Examine Quantization Error in Measurements

- Quantization error varies according to starting phase of the oscillator within a given measurement period
  - Leads to *scrambling* of the quantization noise
- But there is something rather special about the scrambling action …
A Closer Examination of Quantization Noise

- Calculate impact of quantization noise in time:
  \[ \text{out}[k] = x[k] + \text{error}[k] \]
  \[ = x[k] + q[k] - q[k-1] \]

- Take Z-transform:
  \[ O_{\text{ut}}(z) = X(z) + (1 - z^{-1})Q(z) \]

Quantization noise is first order noise shaped!
Measurement of phase error between two signals requires gaps between measurements.

- What is the implication of such gaps?
The Impact of Non-Consecutive Measurements

- Consider measuring input period every *other* cycle
  - Analogous to phase measurement *between* two signals
- Key observation:
  - Quantization noise is no longer first order noise shaped!

Is there a way to restore noise shaping?
Proposed Approach: Gate the Oscillator

Key requirements of approach:
- Turn off oscillator at conclusion of each measurement
- Be sure that state information within oscillator is preserved
  - Reasonably straightforward with CMOS ring oscillators

First order noise shaping is restored!
Step size in time is reduced to one inverter delay
- Quantization noise is still scrambled and first order noise shaped
- Mismatch between delay elements is *barrel-shifted*
  - Greatly improves effective linearity!
- GRO implemented as a custom 0.13u CMOS IC
- External setup consists of signal source and variable delay
  - Test issue: variable delay is nonlinear
**Measured GRO Results Confirm Noise Shaping**

Image showing a 15 Stage Gated Ring Oscillator with variable delay and logic circuitry. The diagram illustrates the input variable delay signal and the harmonics due to nonlinearity of the variable delay. The noise shaped quant. noise is also depicted in the graph.

A graph shows the frequency (MHz) on the x-axis and amplitude (dB) on the y-axis, with measurements confirming noise shaping.
PLL Applications of Proposed GRO Structure
**Multiplying DLL Concept**

- **Goal:** create a higher frequency output clock from an input reference clock signal
  - Inject the reference signal edges into a ring oscillator
  - Periodically delay reference edges through ring oscillator action
- **Key issue:** need to *precisely* tune ring oscillator frequency to avoid *deterministic jitter*
The Benefit of the MDLL Approach

Ye, Jansson, Galton, JSSC, Dec. 2002

- Phase noise of ring oscillator is suppressed by reference edge injection
- Core part is highly digital (i.e., mux and ring osc.)
  - What about $V_{\text{tune}}$ and Sel?
Our Focus

- How do we automatically adjust $V_{\text{tune}}$ to achieve minimal deterministic jitter?
  - We want $\Delta$ to go to zero!
- Can we achieve a highly digital MDLL implementation?

Let us first look at the classical implementation …
Creation of Select Signal

- Select signal (Sel) controls muxing of reference edge into ring oscillator
  - Pulse placement and width of Sel determined by:
    - Location of reference edges
    - Clock multiplication factor (N)

Creation of Sel achieved with purely digital logic
Classical Analog Approach to Adjusting $V_{tune}$

- **Key idea**: compare edges of reference to MDLL output (or other signal) to determine phase error
  - Integrate phase error to adjust $V_{tune}$
- **The problem**: phase detector and integrator have DC offsets that limit reduction of $\Delta$

Low deterministic jitter is challenging to achieve

Proposed Approach

- Compare cycle periods of MDLL output
  - Allows examination of only the MDLL output rather than comparison of its edges to the input reference
  - Deterministic jitter ($\Delta$) is directly seen as the difference between cycle periods of the MDLL output
    - Comparison of same signal allows removal of DC offset
We Can Achieve a Highly Digital Implementation!

- Use GRO to measure output cycle periods
  - Note: leverages scrambling action of GRO, not noise shaping
- Use digital version of correlated double-sampling technique to determine $\Delta$

Helal, Straayer, Wei, Perrott, VLSI 2007.
Our Prototype

- Two custom 0.13u CMOS ICs
  - GRO and core MDLL structures
- FPGA
  - Correlated double-sampling and accumulation operations
- Discrete 14-bit DAC and RC lowpass filter (5.3 MHz pole)
  - 8-bit to 10-bit DAC is adequate (RC pole at 3 MHz)
- Measured jitter:
  - 1.4 ps rms
  - 11.7 ps peak-to-peak
- Issue:
  - Oscilloscope jitter limits measurement
Jitter Estimation Based On Measured Phase Noise

- **Random jitter:**
  - 534 fs
  (1 kHz to 40 MHz)

- **Deterministic jitter:**
  - < 700 fs
  (Ref. spur: -59 dB)
What is Next?

- Apply GRO to a wide variety of PLL applications
  - Fractional-N frequency synthesizers
  - Clock and data recovery circuits
  - Phase noise and jitter measurement

- Consider other VCO-based circuits

VCO-based analog-to-digital conversion
Leveraging Time-Signaling in Analog-to-Digital Conversion

- **Input:** analog tuning of ring oscillator frequency
- **Output:** count of oscillator cycles per Ref clock period

Quantization noise is first order noise shaped!

Similar approaches: Alon, Stojanovic, Horowitz JSSC 2005
Kim, Cho, ISCAS 2006
A Better Implementation for High Speed Conversion

- Assume a high Ref clock frequency (i.e., 1 GHz)
- Increase number of stages, $N$, such that transitions never cycle through any stage more than once per Ref clock period
- Use registers and XOR gates to determine transition count
  - Avoidance of reset action improves operating speed
VCO provides quantization, register provides sampling
- Model as separate blocks for convenience
- Addition of XOR operation on current and previous samples corresponds to a first order difference operation
- Extracts VCO frequency from the sampled VCO phase signal
Corresponding Frequency Domain Model

- VCO modeled as integrator and $K_v$ nonlinearity
- Sampling of VCO phase modeled as scale factor of $1/T$
- Quantizer modeled as addition of quantization noise

Key non-idealities:
- VCO $K_v$ nonlinearity
- VCO noise
- Quantization noise
Example Design Point for Illustration

- Ref clk: $1/T = 1$ GHz
- 31 stage ring oscillator
  - Nominal delay per stage: 129 ps
- $K_{VCO} = 500$ MHz/V
  - $\pm 5\%$ linearity
- VCO noise: -100 dBC/Hz at 10 MHz offset

- Note: no front-end sampler required
SNR/SNDR Calculations with 20 MHz Bandwidth

Simulated ADC Output Spectrum

<table>
<thead>
<tr>
<th>Conditions</th>
<th>SNDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>68.2 dB</td>
</tr>
<tr>
<td>VCO Thermal Noise</td>
<td>65.4 dB</td>
</tr>
<tr>
<td>VCO Thermal + Nonlinearity</td>
<td>32.2 dB</td>
</tr>
</tbody>
</table>

VCO \( K_v \) nonlinearity is the key performance bottleneck.
Reducing the Impact of Nonlinearity using Feedback

- Place VCO-based quantizer within a continuous-time Sigma-Delta ADC structure
  - Quantizer nonlinearity suppressed by preceding gain stage
- Must achieve a highly linear DAC structure
  - Otherwise, noise folding and other bad things happen …

Iwata, Sakimura, TCAS II, 1999
Naiknaware, Tang, Fiez, TCAS II, 2000
A Closer Look at the DAC Implementation

- Consider direct connection of the quantizer output to a series of 1-bit DACs
  - Add the DAC outputs together

What is so special about doing this?
**Key Insight:** Quantizer Acts as a Barrel-Shifter

- Quantizer output rotates through 1-bit DAC elements
  - Acts to shape DAC mismatch and linearize its behavior
A Geometric View of the VCO Quantizer/DEM and DAC

More transitions with large input

Less transitions with small input

\[ V_{\text{tune}} / \overline{V_{\text{tune}}} \]

\[ I_{\text{OUT}} \]

\[ 1 - z^{-1} \text{: Differentiator} \]

31 Element DAC

Quantizing Register

First Order Difference

Current DAC

Variable Delay

Quantizing Register

973 MHz

1-bit DAC slice

\[ I_{\text{OUT}} \]
Our Prototype

- Corresponds to classic second order Sigma-Delta Candy structure *(Candy, Trans. On Comm., Mar 1985)*

- However:
  - Two integrators achieved with only one op-amp
  - *Third order* noise shaping is achieved!
    - VCO-based quantizer adds an extra order of noise shaping
Custom IC Implementing the Prototype

- 0.13u CMOS
- Power: 40 mW
- Active area: 700u X 700u
- Peak SNDR: 67 dB (20 MHz BW)
- Conversion efficiency: 0.5 pJ/conv. step

Straayer, Perrott VLSI 2007
Measured Spectrum From Prototype

Normalized FFT, $F_{IN} = 1$ MHz

<table>
<thead>
<tr>
<th>Input Bandwidth</th>
<th>10 MHz</th>
<th>20 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>76.2</td>
<td>66.4</td>
</tr>
<tr>
<td>SNDR</td>
<td>72.4</td>
<td>65.7</td>
</tr>
</tbody>
</table>

Amplitude (dB)

Frequency (MHz)
Measured SNR/SNDR Vs. Input Amplitude

SNR/SNDR vs. Amplitude, $F_{IN} = 1$ MHz

- SNR
- SNDR

Amplitude (dBFS)

SNR/SNDR (dB)
Behavioral Simulation
The Need for Fast Behavioral Simulation

- Innovation involves both architecture and circuit exploration
  - Typical design strategies closely couple these two domains
    - Simulations too slow to efficiently explore new architectures

Behavioral simulation promotes *architectural* innovation
Designer Environment for C++ Behavioral Modeling

- **Schematic**
  - Provides hierarchical description of system topology

- **Code blocks**
  - Specify module behavior using templated C++ code

- Designers *graphically* develop system based on a *library* of C++ behavioral primitives
  - Easy to create new primitives as needed

- ![Diagram of C++ behavioral modeling system](image.png)
CppSim – A Fast C++ Behavioral Simulator

Written by Michael Perrott (http://www-mdi.mit.edu/~perrott)
CppSim Allows Efficient Transfer of Knowledge

- Architectural knowledge encapsulated within detailed behavioral models and accompanying tutorials
  - Many are already available on our web site

Our next steps over the coming year:

- Detailed CppSim behavioral models and tutorials on:
  - Digital PLL Circuits
    - Multiplying DLL
    - Fractional-N Synthesizer
    - Clock and Data Recovery Circuit
  - VCO-based A/D converters
  - On-chip phase noise/jitter measurement

http://www-mtl.mit.edu/researchgroups/perrottgroup/tools.html
Conclusion

- Phase-locked loop circuits are about to go through dramatic changes as time-signaling is better utilized
  - New time-to-digital converter structures such as GRO
  - New structures/algorithms for achieving high performance (i.e., low noise, fast locking)
    - Example: MDLL circuit utilizing correlated double-sampling techniques to achieve low jitter
- It is time to apply this technology to other circuits, too
  - Example: ADC structures utilizing VCO-based quantizers
- Behavioral simulation facilitates architectural innovation
  - Example: CppSim for PLL and VCO-based ADCs

How can we further utilize time as a signal domain?