NTNU – Trondheim
Norwegian University of
Science and Technology

Ultra-low power PLL frequency sythesizer for Wake-Up Receiver applications

Cole Nielsen

Electronic Systems Design, Specialization Project

Submission date:	December 2019
Supervisor:	Trond Ytterdal, IET
Co-supervisor:	Carsten Wulff, IET

Norwegian University of Science and Technology
Department of Electronic Systems

Abstract.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Problem description.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Preface.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

List of Figures

List of Tables

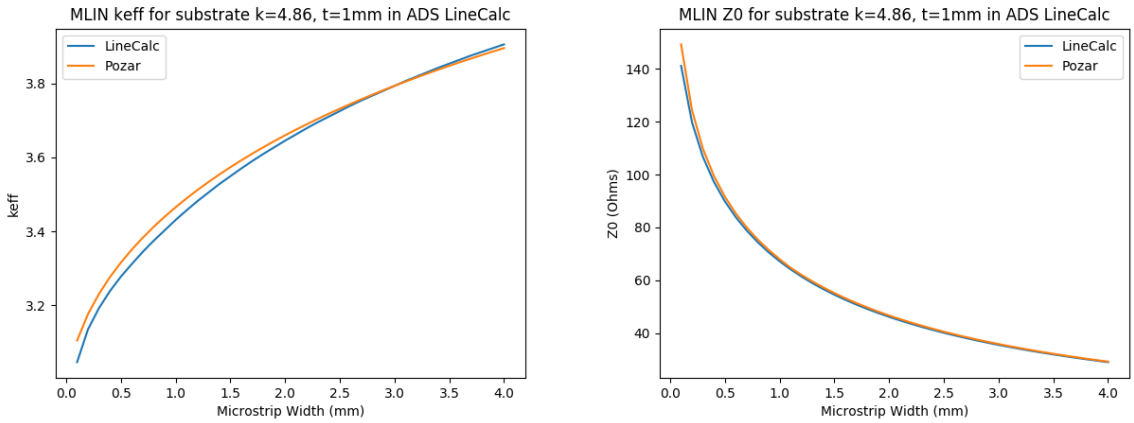
Abbreviations.

DAC	Digital-to-analog converter
DCO	Digitally controlled oscillator
FDSOI	Fully-depleted silicon on insulator
FET	Field effect transistor
FSK	Frequency-shift keying
PLL	Phase locked loop
RO	Ring-oscillator
RX	Receiver
SSB	Single side band
TDC	Time-to-digital converter
TSPC	True single-phase circuit
VCO	Voltage controlled oscillator
WURX	Wake-up receiver

1 Introduction

2 Methods

3 Results



(a) Line Width vs ϵ_e.

(b) Width vs Characteristic Impedance (Z_0).

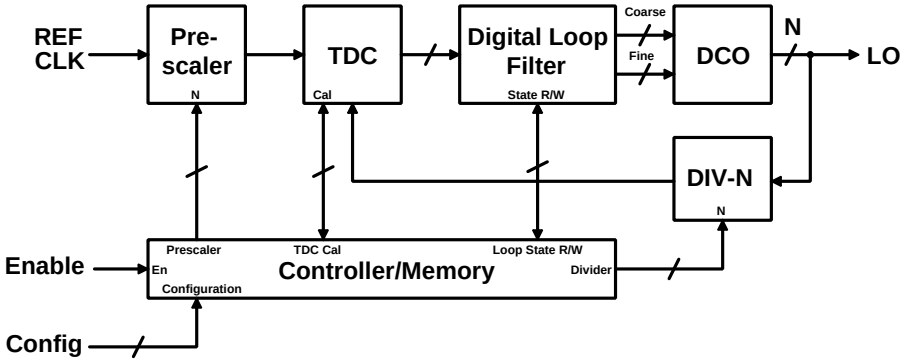
Figure 1: Comparison of LineCalc calculations and the empirical formulas.

4 Discussion

5 Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6 Block diagram



DCO	TDC	Divider	Other	SUM
70 μW	20 μW	10 μW	$<<$ 1 μW	100 μW

7 Specifications

Parameter	Value	Unit	Notes
Frequency	2.4-2.4835	GHz	2.4G ISM Band
Ref. frequency	16	MHz	Yields 6 channels
Power	$\leq$ 100	μW	
Residual FM	$\leq$ 107	kHz$_{RMS}$	BER $\leq$ 1e-2, f_{dev}=$\pm$250 KHz
Initial Lock Time	$\leq$ 50	μs	Upon cold start
Re-lock Time	$\leq$ 5	μs	Coming out of standby
Bandwidth	100	kHz	(nominally), tunable

Table 1: System-level specifications

Additionally: PLL output should support IQ sampling at LO frequency.

Parameter	Value	Unit	Notes
DCO LSB Resolution	$\leq$ 50	kHz	Determined from quantization noise.
DCO DNL	¡ 1	LSB	Ensures monotonicity
TDC Resolution	$\leq$ 3.8	ns	
TDC Resolution (bits)	$\geq$ 4.03	bits	

Table 2: Component-level specifications.

8 pres1

8.1 Motivation

- WSNs require ultra-low power circuits. A sensor should last for many years (¿5) on a coin cell battery.

- With the below P_{avg} values, a CR2032 cell with 0.6 Wh capacity will last:
  - 1 μW $\rightarrow$ 70 years
  - 10 μW $\rightarrow$ 7 years
  - 100 μW $\rightarrow$ 0.7 years
- To save power, run devices with low duty cycle; activate remotely using wake up receiver (WuRx).
  - For 5 years of life, using 10% of battery energy for the WuRx, P_{WURX} **¡ 1.4 μW** on average is required.
- Main challenge for receiver is low power LO synthesis.
  - Synthesizer-free approaches (OOK receiver) lack robustness.
  - Advanced modulation schemes are too demanding with phase noise (PN).
  - Best option is FSK, using low power synthesizer with loose PN requirements to meet power target.

8.2 State of the Art

Performance averages based on sampling of 15 works published in IEEE:

Group	Count	Power [μW]	Freq [MHz]	RF BW [MHz]	Bitrate [kbps]	BER	Sens. [dBm]
All	15	96	1400	1.2	52	3e-3	-66
800/900M	6	138	887	0.95	15	4e-3	-68
All 2.4G	6	81	2400	2	60	2.8e-3	-67
2.4G FSK	2	190	2400	1.5	98	1e-3	-69
2.4G OOK	4	27	2400	2.5	41.5	4e-3	-66

- State of art of 2.4G WuRx:
  - Active Power $\leq$ 100 μW
  - RF BW $\approx$ 1 MHz
  - Sensitivity = -70 dBm (BER 1e-3)
  - Data Rate = 100 kbps

8.3 Objectives - Synthesizer goals

- Design ultra-low-power frequency synthesizer to meet requirements for *practical* wake up receivers.
  - **$\leq$1 μW average consumption**
  - Targeting **1% duty cycle** for **$\leq$100 μW active power**.
    - Fast locking to reduce on time/energy consumption.
- Synthesis range within **2.4 GHz ISM band**.
- Enable **wake up call detection within 1s** with 1e-3 miss rate.
  - Achievable with 250 kbps data, BER $\leq$ 1e-2, 1% duty, 20% wake up call Tx density.
  - Assuming 32-symbol wake up call used for false-alarm robustness.
  - High BER inherent due to high PN, expect many misses before sucess.
  - Use large FSK modulation index (m) to ease PN and power requirements.
    - m=2 $\rightarrow$ 2π phase shift per symbol or $\pm$250 kHz frequency deviation.
    - RMS Residual frequency modulation (RFM) of PLL should be $<<$ symbol frequency deviation to achieve desired BER.
    - **RFM is derived from phase noise integration; use to constrain PLL PN.**

8.4 Architecture - concepts

- Utilize **Digital PLL**.
  - Inherent feedback helps with PVT variation (yield).
  - Calibration easy in digital design, agains helps with PVT variation.
  - Can store state when PLL idle, allowing for faster lock times coming out of standby.
- Utilize low duty cycle to achieve power target.
  - Can exploit semi-frequent calibration to improve performance.
  - Possibly can run PLL open loop when lock is achieved to save power.
- Utilize oscillator running at 1/N subharmonic of target frequency.
  - Use 2N phases in oscillator to achieve equivalent IQ sampling.
  - Avoids having to run oscillator at 2x target frequency as done typically.
  - 1/3 subharmonic operation should allow for dual mode 2.4G and 915M operation.
- Employ subsampling to reduce divider noise and TDC power?

8.5 State of Art - Sub-1 mW PLLS

Application is niche, so comparable PLLs hard to find. *Initial lock time and relock time

Type	P_{PLL} [μW]	P_{osc} [μW]	Freq [MHz]	PN@Δf [dBc/Hz]	t_{lock}* [μs]	Osc.	Ref Freq
Dig. Frac-N	650	304	2400	-110@0.5M	15/4	LC	26M
Ana. Int-N	680	510	2400	-110@1M	130/70	LC	1M
Ana. Int-N	128		500	-94@1M		Ring	31.25M
Ana. Int-N	570		800	-92.6@0.1M	200	LC	0.2M
Dig. Frac-N	250	173	2448		22/1	Ring	9M
Ana. Int-N	950		5500	-106@1M		IL-LC	

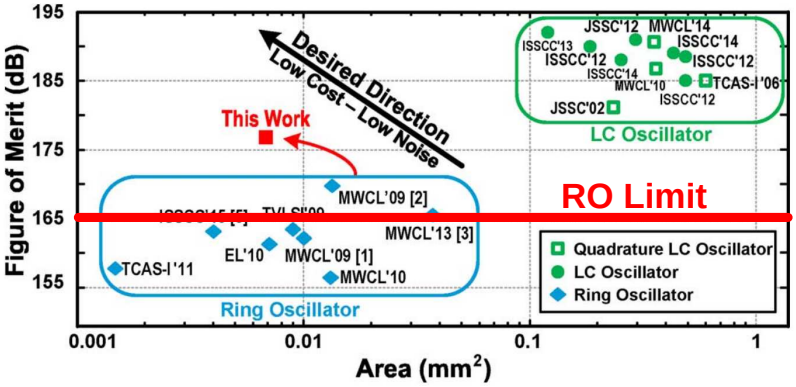
- Power limited by oscillator type. Scaling of LC-oscillator limited by gain requirements for self-starting. LC does not scale to low enough power.
- Current state of art for minimum power will be with ring oscillator, on the order of 200 μW for total PLL consumption.

8.6 Physical limits

Ring oscillator phase noise limit from "Minimum Achievable Phase Noise of RC Oscillators", Navid et al. 2005:

$$PN_{min}(\Delta f) = 10\log 10\left(\frac{7.33k_BT}{P}\left(\frac{f_0}{\Delta f}\right)^2\right) \quad (1)$$

If f_0 = 2.4 GHz, P = 50 μW, Δf= 1 MHz, T = 293 K, $\rightarrow$ **PN$_{min}$ = -84.7 dBc/Hz**
– This limit applied to the below FOM comparison (FOM PN=165 dB):



9 pres2

9.1 Initial simulation/modeling approach

- Started writing Python code to simulate digital PLL.
  - Most familiar for me (I have modelled CDRs in Python before).
  - Simple to implement digital filters with Python Control Systems Library.
    - Also - Convenient stability analysis.
- Decided starting point:
  - Model DCO based on physical phase-noise limit of ring-oscillators.
  - Model loop filter as digital IIR second-order LPF (approximate second-order type-II analog PLL).
  - Ideal divider and TDC.
- Investigate (to determine component requirements):
  - Loop-BW to achieve PN requirements. Closed-loop sensitivity analysis:
    - TDC phase noise significant at f¡BW, determine limits for this.
- Investigate (continued):
  - Quantization, linearity impact of TDC & DCO on phase noise.
    - Determine number of bits required for each, and non-linearity limits.
  - Loop filter implementations (IIR vs FIR), order & type.
    - Evaluate stability, lock time, operating region to meet requirements.
  - Digital data path requirements to implement filter.
  - Algorithm/state machine for handling continuity of fine/coarse DCO ranging.
    - Previous two lead to Verilog description.
- **Final Outcomes:**
  - Full design constraints for PLL components, sufficient to allow schematic-level implementation.

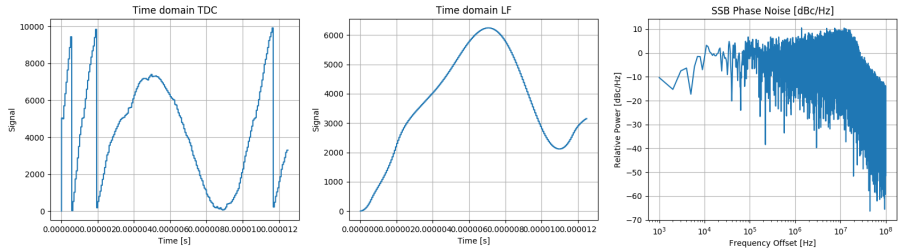
9.2 Implementation in Cadence

- When initial modeling/simulation satisfactory, translate design into a setup within Cadence.
  - Implement using Verilog-A, ahdlLib components?
- Will act as known-good test bench for testing transistor-level implementations of components in future design stages.
  - Replace element by element until fully functioning at transistor level.

10 pres4

10.1 Simulation approach

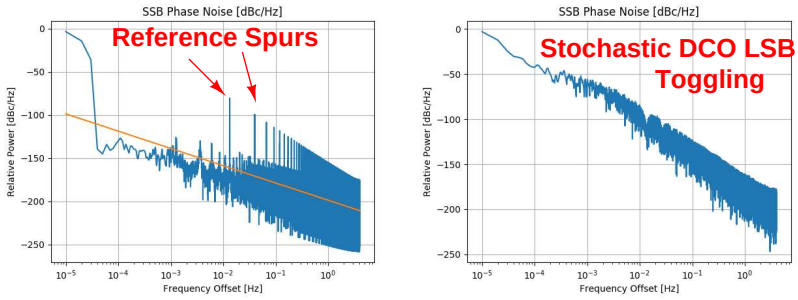
- My current full-PLL simulations are not generally stable.
  - Due to arbitrary choices for loop filter parameters not yielding a stable loop.
  - Next week plan is simulation / analysis / requirements definition of loop filter.
  - Will have to address issue of phase wrapping, fixed point resolution.



- Opted for mathematical-modelling of TDC, separate simulation of DCO.
  - Found straightforward model for TDC phase noise from Michael Perrott of MIT.
  - DCO simulation estimates quantization noise.
  - Will resume with full PLL simulation for loop filter and remainder of modeling.

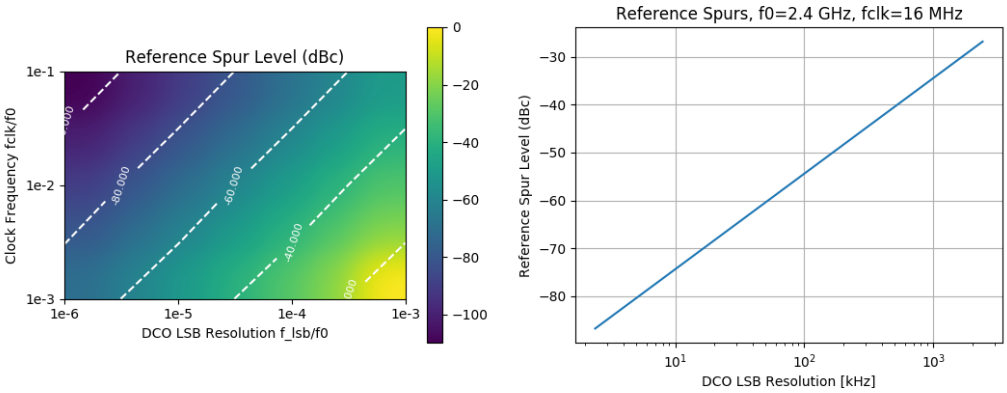
10.2 DCO performance criteria

- Resolution set by frequency accuracy requirements, quantization noise.
- Quantization noise is manifested here as:
  - Reference spurs resulting from deterministic components of signal.
  - A quasi-random noise signal when lock is achieved.
    - Results from stochastic toggling of $\sim$ 1 LSB of DCO tuning word to track low frequency variations.
    - Rolloff of -20 dB/decade at low frequency (same as ring oscillator), -40 dB/decade at high frequencies.



10.3 DCO - reference spur requirements

- Worst case reference spur level.
  - DCO tuning word toggling up/down 1 LSB every reference cycle.
- With f_0 = 2.4 GHz, f_{clk} = 16 MHz:
  - 52 kHz per LSB (i.e. K_{DCO}) is needed for a maximum -60 dBc reference spur level.

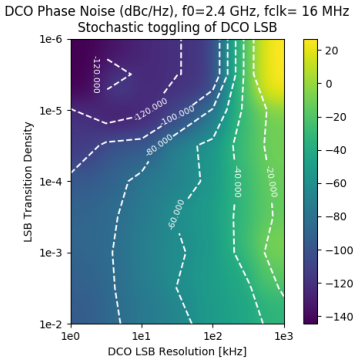


10.4 DCO - steady state cap noise

- In steady state, DCO tuning word will vary occassionaly by ∼ 1 LSB to track stochastic frequency changes.
  - This generates noise and should be less than the thermal phase noise of DCO.
  - Very pessimistic estimate here (assumes abrupt frequency change). Abrupt frequency steps contribute $1/\Delta f$ dependent component to phase noise.
- Theoretical ring oscillator phase noise limit from [2]:

$$PN_{min}(\Delta f) = 10 \log 10 \left(\frac{7.33 k_B T}{P} \left(\frac{f_0}{\Delta f} \right)^2 \right) \quad (2)$$

- If f_0 = 2.4 GHz, P = 50 μW, Δf= 1 MHz, T = 293K, → **PN ¡ -84.7 dBc/Hz** from this noise process.
  - LSB resolution of **50 kHz** seems feasible. Will have to verify with full PLL sim to account for loop dynamics.



10.5 DCO - linearity and accuracy

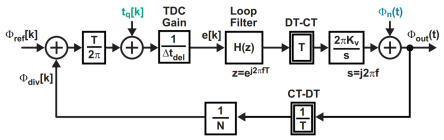
- Frequency accuracy.
  - Indeterminate IF assumed for wake up receivers, so accuracy not so critical.
  - If RF bandwidth of receiver ¿ PLL bandwidth, reasonable assumption is maximum frequency offset (accuracy) should be ¡ PLL bandwidth.
  - Current PLL bandwidth spec is 100 kHz, suggested 50 kHz LSB step from quantization noise analysis is sufficient?
    - This is assuming accuracy is tied to DCO resolution.
- Linearity:
  - Integral non-linearity over the tuning DCO range is not important, so no spec for INL is suggested. Should be locally linear (constrains DNL).
  - Monotonicity is essential, so must strictly have DNL ¡ 1 LSB.

10.6 TDC - phase noise model

- Based on a phase-domain model for PLL phase noise from Michael Perrot [1].

$$S_{\phi,out}(f) = f_{clk} \cdot |2\pi N G(f)|^2 \cdot \frac{\Delta t_{del}^2}{12} \quad (3)$$

$$G(f) = \frac{A(f)}{1 + A(f)} \quad (4)$$



- $S_{\phi,out}(f)$ is the TDC phase noise component, N is the divider modulus, G(f) is the closed loop PLL transfer function, A(f) is the open loop transfer function, Δt_{del} is the TDC time resolution.
- G(0) = 1, and G(f) ≈ 1 for f ∈ [0, f_{CBW}], where f_{CBW} is the closed loop bandwidth.
- Naive estimate for TDC time resolution:
  - Phase noise flat within closed-loop bandwidth. This component dominates power of the integrated phase noise a PLL with low TDC-resolution.
  - Use Residual frequency modulation equation and equation 2 (TDC phase noise) to estimate Δt_{del}. Integrate in f ∈ [0, f_{CBW}]

$$\Delta f_{RFM} = 2 \int_{f_a}^{f_b} f^2 * PN(f) df \quad (5)$$

- With f_{clk} = 16 MHz, N = 150 (for 2.4 GHz synthesis), Δf_{RFM} ¡ 107 kHz to meet BER requirement, and f_{CBW} = 100 kHz. (PLL presumed to have very low TDC resolution)

- Δt_{del} ¡ 3.8 ns
- Phase noise of TDC below f_{CBW} is -47.7 dBc/Hz
- Equates to minimum of 16.4 quantization steps for TDC (4.03 bits).
- Assumptions of high phase noise and low resolution appear valid.

11 pres5

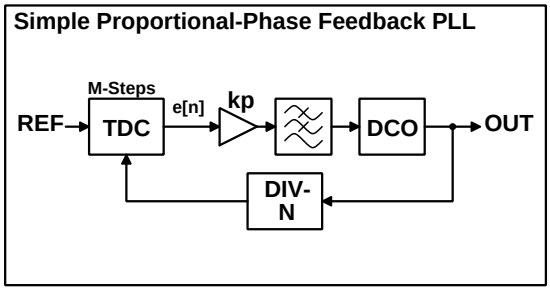
11.1 This week

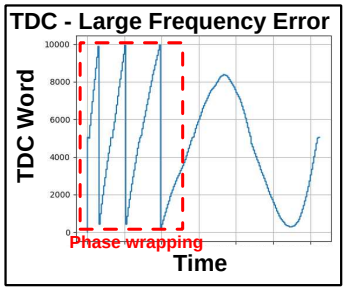
- **Primary:** Loop analysis, requirements definition.
  - Design open loop filter design to meet closed loop requirements.
    - Dependent on DCO properties (k_{DCO}, f_0, tuning range).
    - Difference equation implementing discrete time 2nd order IIR filter.
  - Datapath requirements (fixed-point resolution)

11.2 Next week

- **Primary:** Ideal component PLL implementation in Cadence, continue loop filter work.
  - Ideal component PLL implementation is not a lot of work.
  - Loop filter very critical, spend more time on this.
    - Need to make Verilog description of loop filter for simulation in Cadence.

11.3 Loop filter original attempt



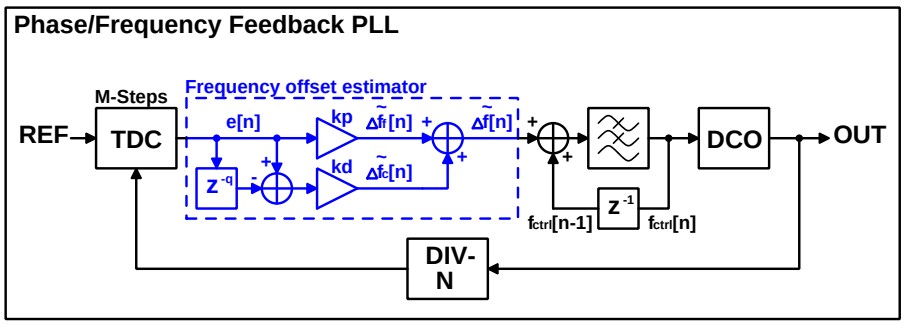


- Started with simple PLL loop with proportional-phase fed into loop filter (LF).
  - Not stable at large frequency offset, due to frequency wrapping. At the TDC:

$$\Delta\phi = \frac{2\pi\Delta f T}{N} \rightarrow T_{wrap} = \frac{N}{\Delta f} \tag{6}$$

  - Upon cold start, Δf is expected to be up to 100 MHz, N=150 $\rightarrow T_{wrap} = 1.5\ \mu$s
    - $f_{wrap} \sim$ 600 kHz, this is unstable with a loop bandwidth of 100 kHz.
  - Must opt for alternate loop structure.

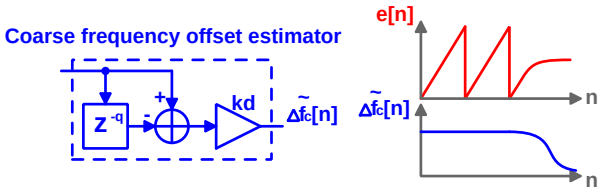
11.4 Loop filter new approach



- Two-fold approach: utilize propotional-phase and coarse frequency offset estimation in feedback.
  - Coarse frequency estimator to handle high frequency offset (e.g. cold start-up).

  - Proportional-phase for near steady state. Acts like fine frequency estimator.
- Offset estimates are summed with previous oscillator tuning word (OTW, also f_{ctrl} here), then low passed filtered to yield new OTW.
  - Low pass filter loop keeps steady state.
  - Frequency estimator updates if any changes detected.

11.5 Coarse frequency estimation



- **Coarse frequency estimation:** Given M-step TDC, outputting phase error signal $e_\phi[n]$, and a divider modulus N

$$\Delta\phi_{DCO}[n;q] = N \cdot \Delta\phi_{REF}[n] = 2\pi\frac{N}{M}\left(e_\phi[n] - e_\phi[n-q]\right), \qquad \Delta\phi_{DCO}[n;q] = \Delta\omega_{DCO}[n]qT_{ref} = 2\pi q\frac{\Delta\tilde{f}_{DCO}}{f_{ref}} \tag{7}$$

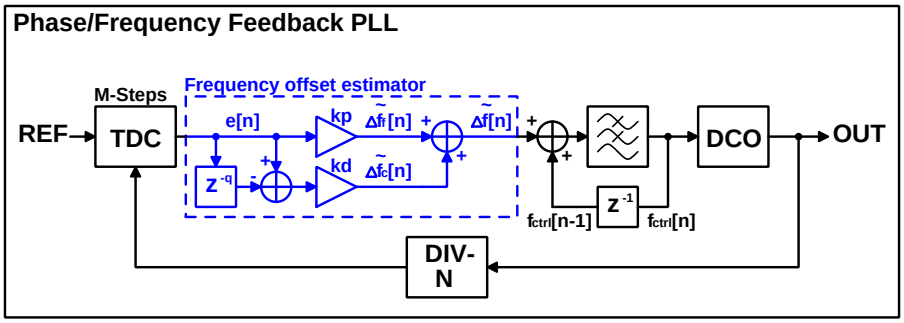
$$\Delta\tilde{f}_c = \Delta\tilde{f}_{DCO} = \frac{f_{ref}}{q}\frac{N}{M}\left(e_\phi[n] - e_\phi[n-q]\right) \tag{8}$$

- Is a discrete differentiator, with gain coeficient to convert $d\phi/dt$ to frequency.
  - Design logic to handle phase wrapping.
  - Useful in coarse frequency range calibration. Can detect fast if frequency offset too large.
  - Delay q is used to increase frequency resolution.
- Given DCO gain K_{DCO}, the required gain K_d of the filter is:

$$K_d = \frac{f_{ref}}{qK_{DCO}}\frac{N}{M}\left(e_\phi[n] - e_\phi[n-q]\right) \tag{9}$$

- Disable the coarse estimator if $e_\phi[n] - e_\phi[n-q]$ ¡ some threshold:
  - Offset small enough, allow to run as classical phase-detector mode.

11.6 Loop filter Gain Coefficients



- When running in proportional-only feedback mode, the open loop gain $A(f)$ is:

$$A(f) = K_p\frac{M}{N}\frac{K_{LPF}}{s + \omega_{LPF} - K_{LPF}}\frac{2\pi K_{DCO}}{s} \tag{10}$$

- The closed loop gain is therefore (continuous):

$$G(f) = \frac{A(f)}{1 + A(f)} = \frac{2\pi M K_p K_{LPF} K_{DCO}/N}{s^2 + s(\omega_{LPF} - K_{LPF})/N + 2\pi M K_p K_{LPF} K_{DCO}/N} \tag{11}$$

- The form of a second order low pass filter is, with natural frequency ω_n and damping coefficient ζ:

$$H_{LPF}(f) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{12}$$

- Setting $H_{LPF}(f) = G(f)$, equivalencies for ω_n and ζ are found:

$$\omega_n = \sqrt{2\pi M K_p K_{LPF} K_{DCO}/N} \tag{13}$$

$$\zeta = \frac{\omega_{LPF} - K_{LPF}}{2\sqrt{2\pi M N K_p K_{LPF} K_{DCO}}} \tag{14}$$

- The Butterworth closed loop response with 100 kHz bandwidth, $\omega_n \sim 2\pi \cdot 100Khz$, $\zeta = 0.707$.
- Coefficients K_{LPF}, K_p, ω_n can be solved computationally.
- Need to reformulate in Z-domain.

11.7 Coarse frequency calibration

- Ring oscillator DCOs will have large variation in frequency due to PVT variation.
- Use bank of coarse capacitance values to correct range of oscillator.
- Coarse frequency estimator used in this calibration. DCO is has a fine range of Δf_{fine}.

Coarse frequency tuning state machine (as pseudo-code)

1. C_{tune} = C_{opt} = 0; LE = Δf_{fine}
2. Reset PLL
3. Estimate Frequency offset $\rightarrow \tilde{f}_{offset}$
4. **If** abs($\tilde{f}_{offset}$ - $\Delta f_{fine}/2$) **;** LE: // Centers fine tuning range around target frequency
   - LE = abs($\tilde{f}_{offset}$ - $\Delta f_{fine}/2$)
   - C_{opt} = C_{tune}
5. **If** C_{tune} == C_{max}: **Goto** 8
6. C_{tune} += 1
7. **Goto** 2
8. C_{tune} = C_{opt}; **End**

References

Week Number	Dates	Tasks	Outcomes
36	2.9 - 8.9	Review PLL Design	Refreshed Knowledge
37	9.9 - 15.9	Modeling/simulation (set up)	–
38	16.9 - 22.9	Modeling/simulation	TDC/DCO Requirements
39	23.9 - 29.9	Modeling/simulation	Loop Filter/Digital Algorithms
40	30.9 - 6.10	Modeling/simulation	**Loop filter,** Ideal implementation in Cadence
41	7.10 - 13.10	Circuit Research	DCO/Divider topologies
42	14.10 - 20.10	Circuit Research	TDC/other topologies
43	21.10 - 27.10	Circuit Implementation	Digital logic (schematic)
44	28.10 - 3.11	Circuit Implementation	DCO (schematic)
45	4.11 - 10.11	Circuit Implementation	Divider/other (schematic)
46	11.11 - 17.11	Circuit Implementation (TDC)	
47	18.11 - 24.11	Circuit Implementation (TDC)	TDC (schematic)
48	25.11 - 1.12	Full Circuit testing	Testbenches, find bugs, design fixes
49	2.12 - 8.12	Full Circuit testing	Design Fixes/iteration
50	9.12 - 15.12	–	–

Legend: **Done** **Current** **Revised**

A Appendix - Schedule

B Appendix - Code

```
######################################################################
# Simulation loop
######################################################################

t0 = time.clock()
for n in range(SAMPLES)[1:]:
    clk_out[n] = clk.update()
    tdc_out[n] = TDC_SCALE*((TDC_OFFSET+tdc.update(clk=clk_out[n-1], xin=
        div_out[n-1]))%TDC_STEPS)
    lf_out[n]  = lf.update(xin=tdc_out[n-1], clk=clk_out[n-1])
    osc_out[n] = dco.update(lf_out[n-1])
    div_out[n] = div.update(osc_out[n-1], DIV_N)
tdelta = time.clock() - t0
print("\nSimulation completed in%f s"%tdelta)
```

Listing 1: excode