Ultra-Low Power PLL for Wake-up Receiver Applications

Specialization Project Progress - 6th Week

Cole Nielsen
Department of Electronic Systems, NTNU

4 October 2019 (Calendar week 40)

Autumn Timeline

Week	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, DCO, TDC, calibration**
41	7.10 - 13.10	Circuit Research	DCO/Divider topologies, **Ideal Virtuoso implementation**
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**

Timeline Tasks

This week

— Originally planned on implementing ideal simulation in Virtuoso using ideal library components and some Verilog.

— *What I actually did*:

- Found issue with how I defined phase noise requirements
  — Had to redefine this, the TDC and loop bandwidth specification.
- Had to rework DCO simulation model to have accurate phase noise.
- Analyzed sensitivity/variation of DCO gain for loop filter analysis.
- Analyzed TDC to determine accuracy/linearity (tied to calibration)
- Decided to change loop filter to use PI-based control.

Requirements redefinition

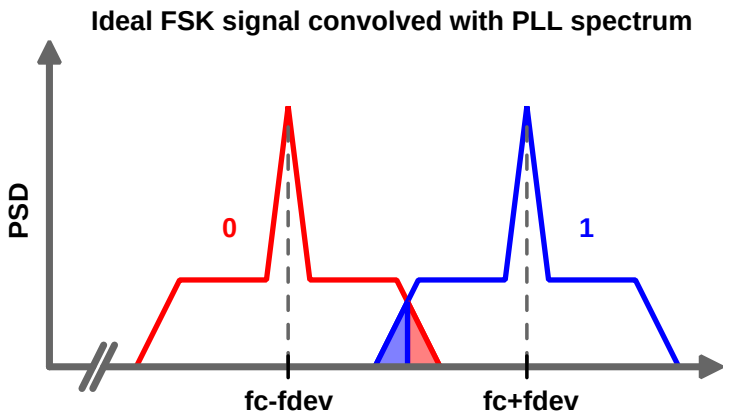
Redefining phase noise and loop bandwidth requirements

— *Made bad assumption with original requirements*, i.e. residual frequency modulation is predicated on a Gaussian phase noise distribution.

— Now redefined in terms of the actual spectum of the PLL for use with 2FSK. Phase noise is derived from the closed loop PLL transfer function G(f).

— Bandwidth is now selected to minimize amount of power that overlaps in 2FSK tones.

- (Integrated PSD in overlap)/(total power) = bit error probability.



Requirements redefinition

Loop definition and phase noise components.

— Phase noise and closed loop response G(f) are simple to define. Second order butterworth G(f) is desired. Loop bandwidth = ω_N, ζ = 0.707:

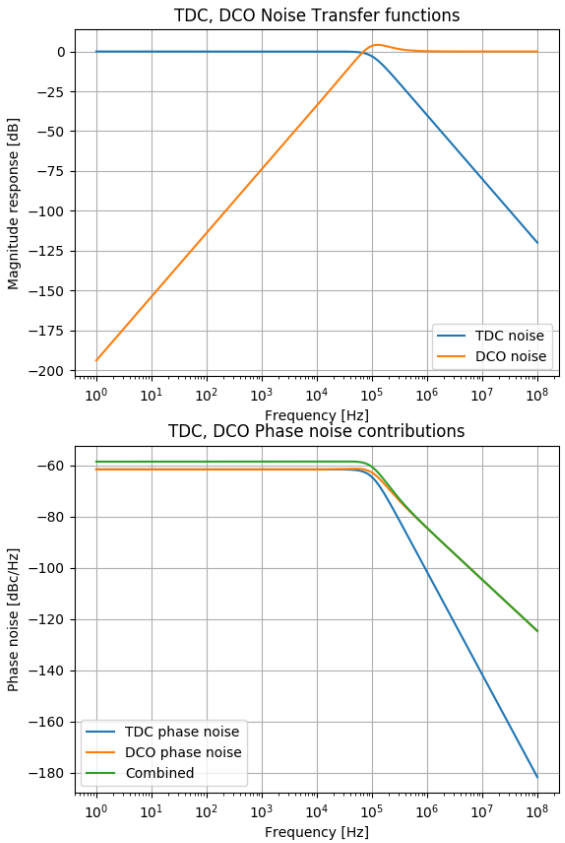
$$G(f) = \frac{\omega_N^2}{s^2 + 2\zeta\omega_n s + \omega_N^2} \quad (1)$$

— The TDC quantization phase noise is given by:

$$PN_{TDC}(f) = f_{ref} \cdot |2\pi NG(f)|^2 \cdot \frac{\Delta t_{del}^2}{12} \quad (2)$$

— The oscillator phase noise (based on ring oscillator theoretical limit) is:

$$PN_{osc}(f) = |1 - G(f)|^2 \cdot \frac{7.33 k_B T}{P_{osc}} \cdot \left(\frac{f_0}{f}\right)^2 \quad (3)$$



Requirements redefinition

Connecting BER, loop bandwidth, frequency deviation.

— Need to calculate power of oscillator within Δf of the carrier. If $V_{osc} = \Re\left(e^{j\omega_0 t - j\phi_{noise}(t)}\right)$, and $\phi_{noise}(t)$ is small:

$$\Re\left(e^{j\omega_0 t - j\phi_{noise}(t)}\right) \approx \Re\left(e^{j\omega_0 t}(1 - j\phi_{noise}(t))\right) = \cos(\omega_0 t) + \phi_{noise}(t)\sin(\omega_0 t) \tag{4}$$

— The power within Δf of the carrier is (if the carrier is normalized to a power of 1):

$$P_{out}(\Delta f) = 1 + 2\int_0^{\Delta f} \phi_{noise}^2(t)df \tag{5}$$

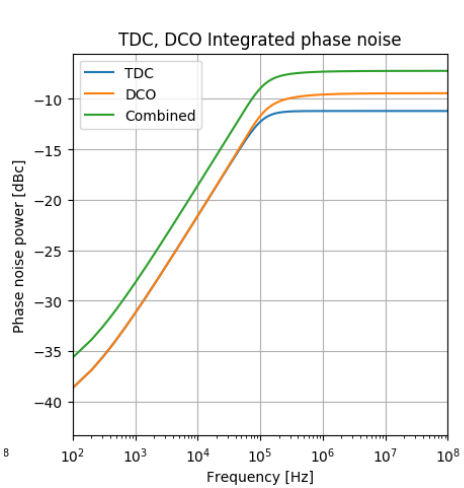
— To find the minimum FSK frequency deviation for a given BER, the following is solved computationally. This is based on the overlap principle. ($f_{dev} = \Delta f$)

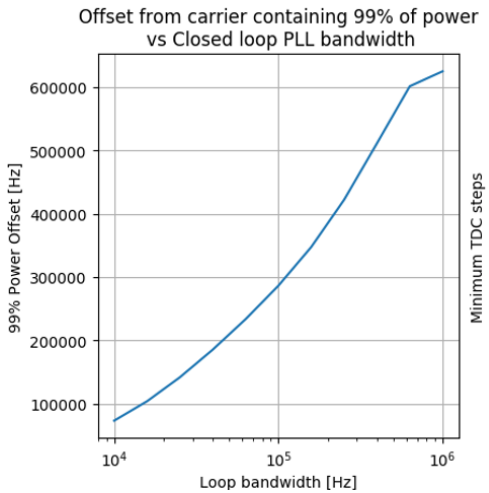
$$P_{out}(\Delta f = \infty) \cdot (1 - 2BER) = 1 + 2\int_0^{f_{dev}} \phi_{noise}^2(t)df \tag{6}$$

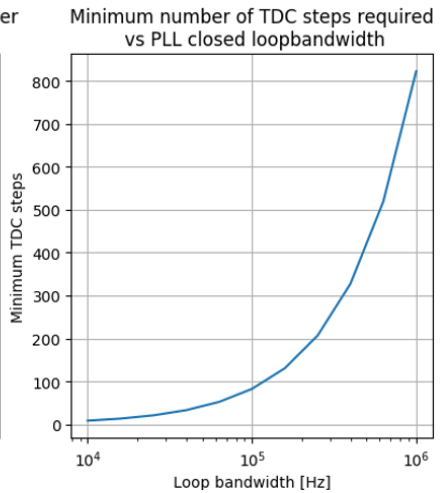
Requirements redefinition

Solving for loop bandwidth computationally

- Most power from phase noise is accumulated near frequency of loop bandwidth.
- TDC and DCO noise were optimized to provide approximately the same contribution to phase noise.
- For 250 kHz of frequency deviation, $\leq$ 70 kHz of closed loop bandwidth is required.
- Selecting 50 kHz loop bandwidth, with 64 TDC steps to allow for 50 kHz margin on BER=1e-2 with f_{dev} = 250 kHz.







Loop Filter

Resolving unknowns

— Thoughtful loop filter design requires consideration of some physical parameters

- **TDC accuracy**
  - — Affects PLL transfer function, so we want to minimize variation.
  - — Innately tied to calibration process.
  - — Important for accurate calibration of frequency, to hit PLL lock-in range.
  - — Important for INL; also DNL associated with TDC phase wrap-over.
- K_{DCO} **and variance of** K_{DCO}
  - — Very important in determining gain coeficients in loop filter.
  - — Variance of K_{DCO} will be **the most** significant factor in overall PLL loop perfomance variation, and will be hard to calibrate.
  - —

— Consequently, I analyzed the real-hardware considerations for these parameters to provide some inside into the loop design.

TDC accuracy

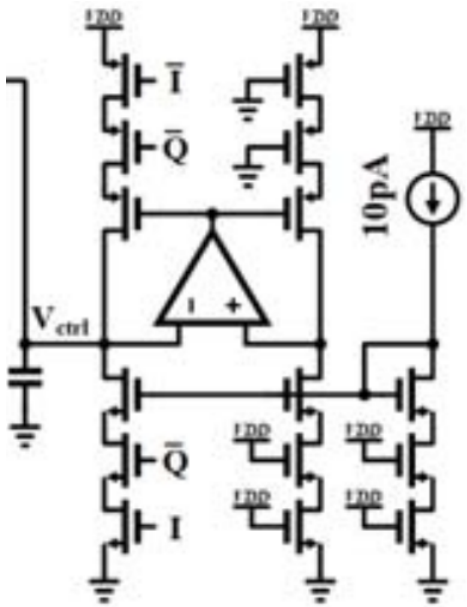
Resolving unknowns

- Had to look into TDC circuit architectures to consider question accuracy and calibration.
- **Analog or digital??**
  - **Analog**
    - (+) Better accuracy (always running)?
    - (-) Constant Power draw from CP/loop filter
    - (-) Have to wait for lock everytime
  - **Digital**
    - Calibrate once, run for a long time before recalibrating.
    - Lower overall power.
    - Inherent drift.
    - Possibility - Counter based design requiring no calibration???
- Currently believe digital will have power supremacy.
- Targeting $\pm$ 50 ppm accuracy to correspond to $\pm$ 120 kHz at 2.4 GHz.

Analog TDC

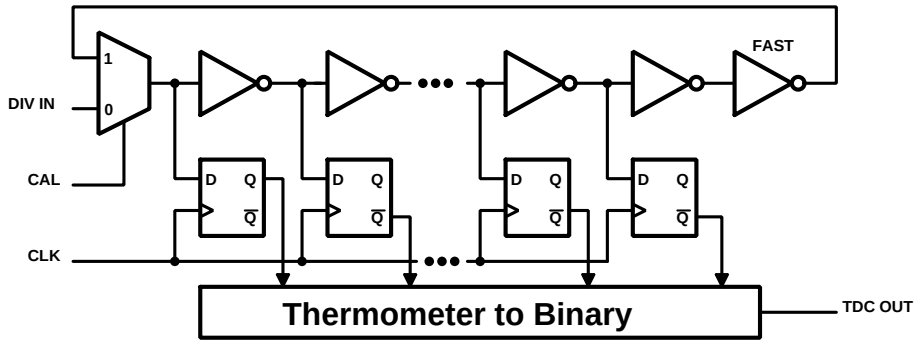
Traditional Analog DLL

- Simple analog DLL with M inverter stagers, first order filter.
- Lock time on the order of 10 μs. With digital, amortized calibration time **should** be much lower than this.
- With first order loop filter, 9.9$\cdot\tau$ is required for $\pm$ 50 ppm settling.
  - For 10 μs settling time, given $t_{settled} = 9.2/(2\pi f_p) \rightarrow f_p = 157$ kHz.
- with 2^N=M stages, tap stages M/2 ($\bar{I}$), 3M/4($\bar{Q}$) and M (I) to use simple IQ based charge pump/phase detector circuit on the right.



Digital TDC

Calibrated Delay line



- Standard M-stage inverter-based delay line.
- Use mux to allow for delay line to operate as ring oscillator
- Measure phase with TDC of free-running oscillator.
  - Change in phase between measurements is related to error of the delay line.
- If Δt_{line} is the delay error of the delay line:

$$\frac{\Delta t_{line}}{T_{ref}} = \frac{\Delta TDC(N_{CYCLES})}{M \cdot N_{CYCLES}} \tag{7}$$

Digital TDC - Algorithm

Calibrated Delay line

— For $\pm$ 50 ppm accuracy, 10% tuning range and 10-bit DAC is sufficient.

— Use binary search algorithm to find optimal tuning word.

- Frequency measurement granularity increases with measurement period, start coarse and increase double time with step of binary search

— With M=64, and 10 bits, search will take 128 μs.

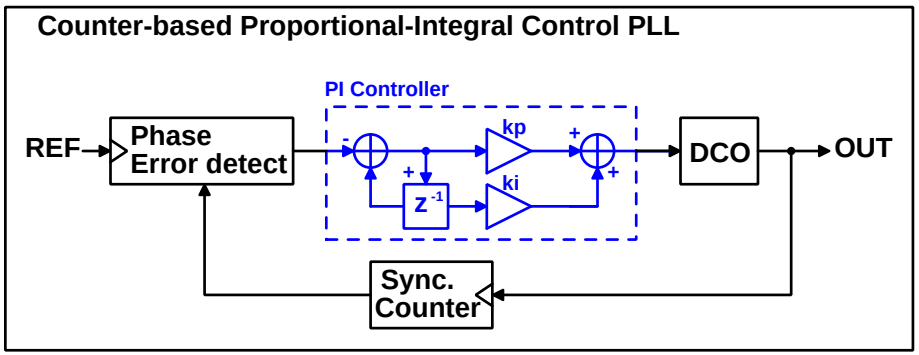
— (May also require some coarse calibration, this is expected to be fast)

— Better than analog if we calibrate less than 1 in 13 power ups.

- 40 power-ups a second with 10 sec between calibrations this is amortizes to 0.32 μs calibration time per wake up.

Digital TDC

Counter-based TDC



- Use synchronous counter clocked by oscillator to keep track of phase.
- Loop filter samples the counter phase every reference clock cycle.
- **Requires no calibration.** Linearity good.
- Resolution is $\log_2(2.4G/16M)$ = 7.2 bits. Good enough.
- Instant start up.
- **Power will be an issue???**
- Will have to test in simulation...

DCO frequency tuning and K_{DCO}

Resolving unknowns

— The fractional tuning range of the oscillator is:

$$\frac{\Delta f}{f_c} = \frac{1}{2} \cdot \frac{\gamma V_{DD} (1 - \ln 2)}{V_{DD} \left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2} \ln 2 \right) - V_{t0} (1 - \ln 2)} \tag{8}$$

— If a N-bit DAC is used to control the oscillator, the resulting DCO gain is therefore:

$$K_{DCO} = \frac{\Delta f}{2^{N_{DAC}}} = \frac{f_c}{2^{N_{DAC}+1}} \cdot \frac{\gamma V_{DD} (1 - \ln 2)}{V_{DD} \left(\frac{7}{8} - \ln 2 + \frac{\gamma}{2} - \frac{\gamma}{2} \ln 2 \right) - V_{t0} (1 - \ln 2)} \tag{9}$$

— The variation of the DCO gain, given the parameters it depends on is:

$$\sigma_{KDCO} = \sqrt{\left(\frac{\partial K_{DCO}}{\partial V_{DD}} \cdot \frac{\sigma_{VDD}}{K_{DCO}} \right)^2 + \left(\frac{\partial K_{DCO}}{\partial V_{t0}} \cdot \frac{\sigma_{Vt0}}{K_{DCO}} \right)^2 + \left(\frac{\partial K_{DCO}}{\partial \gamma} \cdot \frac{\sigma_{\gamma}}{K_{DCO}} \right)^2} \tag{10}$$

— For example: 1% variation in all parameters yields ca. 6% variation in tuning.

DCO fix

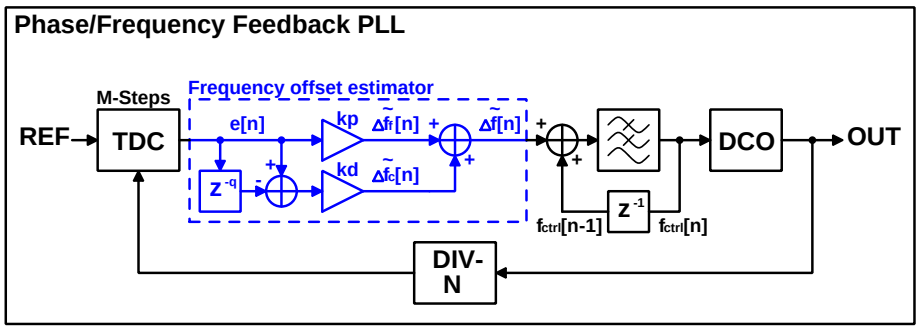
Updating phase noise model

- Fixed phase noise model in time domain simulation of PLL.
- Before phase noise was not accurately set, just loosely.
- Derived a Z-domain model for discrete random phase walk, am able now to convert continuous time phase noise spec to a discrete time model with accurate phase noise.

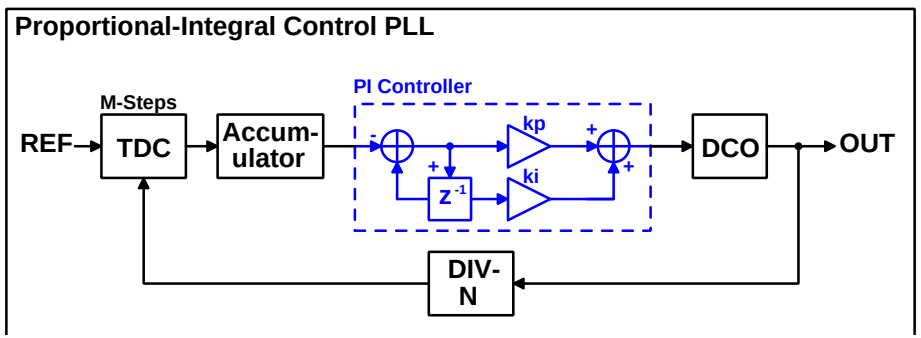
Loop Filter Change

Change to PID based loop filter

— **Old:** Complicated.

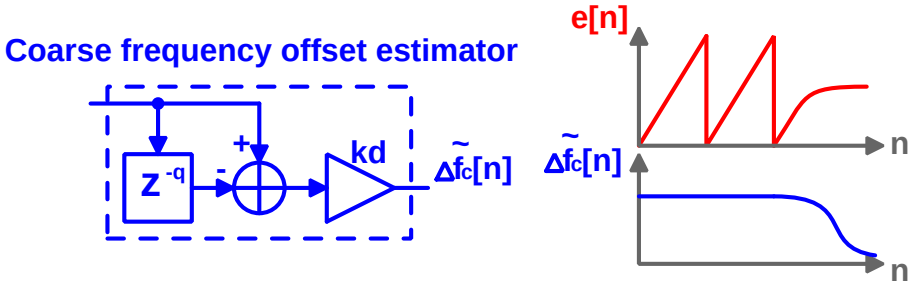


— **New:** Much simpler PID-based, added phase accumulator to handle wrap-over issues, allows for more stable control. Calibration should prevent accumulator overflow. Derivative path only used for calibration (not shown).



Loop Filter

Coarse frequency offset estimation



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

$$\Delta\phi_{DCO}[n;q] = N \cdot \Delta\phi_{REF}[n] = 2\pi \frac{N}{M} \left(e_\phi[n] - e_\phi[n-q]\right), \qquad \Delta\phi_{DCO}[n;q] = \Delta\omega_{DCO}[n] q T_{ref} = 2\pi q \frac{\Delta\tilde{f}_{DCO}}{f_{ref}} \tag{11}$$

$$\Delta\tilde{f}_c = \Delta\tilde{f}_{DCO} = \frac{f_{ref}}{q} \frac{N}{M} \left(e_\phi[n] - e_\phi[n-q]\right) \tag{12}$$

— 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.

Frequency Calibration (New)

Coarse frequency calibration algorithm

- Use binary search algorithm to tune capacitor bank for frequency range.
- With 75 cycles measurement time, 2.4 GHz RF frequency, 16 MHz reference, and 64 TDC steps, within 0.5 MHz can be measured in 5 μs
- A 4 bit binary seach would take circa 20 μs to execute.
- Should hopefully put initial frequency of PLL in lock range, and avoid issues with phase wrapping in the phase error accumulator.

Specification (changed)

System Performance Targets

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~~ 50	kHz	(nominally), tunable

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

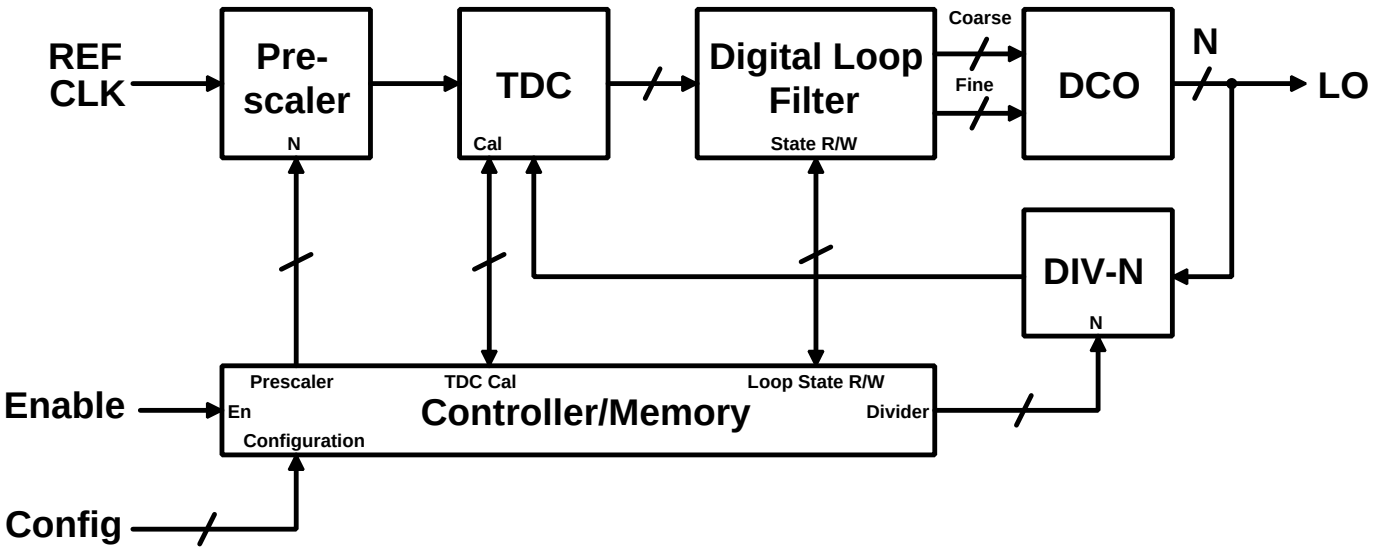
Specification (changed)

PLL Component Performance Targets

Parameter	Value	Unit	Notes
DCO LSB Resolution	≤ 50	kHz	Determined from quantization noise.
DCO DNL	< 1	LSB	Ensures monotonicity
TDC Resolution	~~3.8~~ 0.95	ns	
TDC Resolution (bits)	~~4.03~~ 6	bits	

Architecture (unchanged)

Block Diagram



Power Targets

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

Project Phases

Autumn 2019

- System modeling and simulation.
  - Learn PLL theory in detail
  - Evaluate feasability of PLL architectures (counter, TDC-based)
  - Determine requirements for TDC/DCO/Divider/logic (bits of resolution, accuracy etc) to meet PLL performance specifications.
  - Determine digital logic for loop filter, validate stability and lock time performance.
- Research ultra-low power circuit topologies to implement system components that will meet determined requirements.
- Translate component-level specifications into schematic-level circuit designs.
  - Try, fail, try again until functional at schematic level.
    - I expect the TDC to be difficult.

Project Phases (continued)

Spring 2020

- Finalize schematic-level design.
- Establish thorough tests for PLL performance (automated?) to help in layout.
- Layout of PLL.
  - Design iteration until design specs met.
  - Probably very time consuming.
- Full characterization/validation of design performance.
  - Comprehensive Corners/Monte-Carlo testing (time consuming??)
  - More design iteration if new issues crop up...
- Thesis paper writing.