Python Framework for Design and Analysis of Integer-N ADPLLs

Specialization Project Progress - 13th Week

Cole Nielsen

Department of Electronic Systems, NTNU

22 November 2019 (Calendar week 47)

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
42	14.10 - 20.10	Circuit Research	TDC/other topologies
43	21.10 - 27.10	Spur analysis, filter automation	
44	28.10 - 3.11	Filter automation, SNR estimation	
45	4.11 - 10.11	Variation analysis, flicker noise	Histograms/yield estimates
46	11.11 - 17.11	Real DCO sensitivity, TDC/divider jitter	Simlate ring-DCO in Virtuoso
47	18.11 - 24.11	~~PLL + Radio simulation~~ **Report**	~~BER estimate~~
48	25.11 - 1.12	Organize code, Report	(I have an Exam on 30.11)
49	2.12 - 8.12	**PLL/Radio BER simulation**, Report	
50	9.12 - 15.12	Report writing	Complete by 14.12

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

Timeline Tasks

This week

— **Report writing**

- Redirected focus this week to writing as it is high priority.

— **PLL/Radio BER simulation**

- Original plan for week. Less of a priority, moved to after 30 November (post-exams for me).

— **Ring oscillator simulation in 22FDX:**

- Fit simulation data to model, reasonably accurate.
- Not sure this fits in well with the report.

Report writing

Format

- Abstract
- Problem/project description
- Introduction
- Theory
  - Discuss theory relevant to understand the simulation engine and optimizer.
- Methods
  - Describes how the simulator is implemented, how the filter designer/optimizer is implemented.
- Discussion (combined with results)
  - Discuss choices made in simulator/optimizer design, compare to existing solutions, design examples
- Concusion

Report writing

Theory section

- Introduce what a PLL is
- Develop basic fully continuous PLL model
  - Use continuous model to develop basic PI-continuous loop filter.
- Develop discrete-time, digital PLL model from continuous model
  - Both the discrete transfer-function and continous approximations developed.
- Pll phase noise
  - Derive/explain basic phase noise models for all PLL components
  - Derive PLL noise PSD from PLL phase signal
  - Derive PLL noise sensitivity functions

Report writing

Methods

— Simulator implementation

- Describe how to implement a accurate oscillator phase noise in discrete time simulation
- Describe discrete phase noise/spur measurement simulation requirements and calculation method
- Describe Monte-Carlo sampling engine

— Optimizer implementation

- Describe implementation of fast settling time and phase noise estimation heuristics
- Describe loop filter optimizer algorithm to perform phase noise minimization with constrained settling time
- Describe second order optimization of loop filter implementation resolution.
- Describe BER simulation implementation.

Report writing

Discussion

- Simulator/optimizer implementation
  - Discuss choices made in simulator/optimizer approach
- Comparison to existing solutions
  - Discuss what has been done before, what is new with this approach, compare performance?
- Design example
  - Show design process and results (simulation) with engine
  - Mention any considerations that must be made using the engine (i.e. pitfalls)

Ring oscillator in 22FDX

Ring oscillator model fit

— Last week discussed simulation of ring oscillator

— Curve fitted the below equation based on the following parameter sweeps: RVT/SLVT devices, varied temp in {-40, 25, 85}, L in {100n, 500n}, W/L in {1, 10}, C_{load} in {0, 1f, 10f}, V_{BB} in {0, 0.8}

— Now have estimates for model of μ_n, $C_o x$, body effect coefficient γ and V_{t0}

$$f_{osc} = \frac{\mu_n C_{ox}}{4 \ln 2 NC} \left(\frac{W}{L} \right)_n \left[V_{DD} \left(\frac{7}{8 \ln 2} - 1 \right) - V_{t0} \left(\frac{1}{\ln 2} - 1 \right) + \gamma V_{BG} \left(\frac{1}{\ln 2} - 1 \right) \right] \quad (1)$$

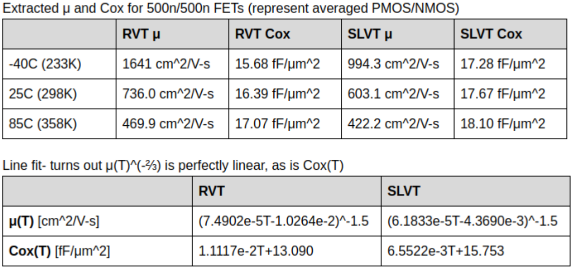
Via the Python simulation, can apply Monte-Carlo sampling to these model parameters to capture effects of variation of DCO gain:

$$\frac{\partial f_{osc}}{\partial V_{BG}} = \gamma V_{BG} \frac{\mu_n C_{ox}}{4 \ln 2 NC} \left(\frac{W}{L} \right)_n \left[\frac{1}{\ln 2} - 1 \right] \quad (2)$$

Ring oscillator in 22FDX

Ring oscillator model - fitted parameters

Parameter	NFET	SLVTNFET	PFET	SLVTPFET
V_{th} [mV]	390-0.69T	340-0.67T	-387+0.88T	-312+0.77T
γ [V/V]	-0.072	-0.083	0.082	0.069



Extracted μ and Cox for 500n/500n FETs (represent averaged PMOS/NMOS)

	RVT μ	RVT Cox	SLVT μ	SLVT Cox
-40C (233K)	1641 cm^2/V-s	15.68 fF/μm^2	994.3 cm^2/V-s	17.28 fF/μm^2
25C (298K)	736.0 cm^2/V-s	16.39 fF/μm^2	603.1 cm^2/V-s	17.67 fF/μm^2
85C (358K)	469.9 cm^2/V-s	17.07 fF/μm^2	422.2 cm^2/V-s	18.10 fF/μm^2

Line fit- turns out μ(T)^(-⅔) is perfectly linear, as is Cox(T)

	RVT	SLVT
μ(T) [cm^2/V-s]	(7.4902e-5T-1.0264e-2)^-1.5	(6.1833e-5T-4.3690e-3)^-1.5
Cox(T) [fF/μm^2]	1.1117e-2T+13.090	6.5522e-3T+15.753

Ring oscillator in 22FDX

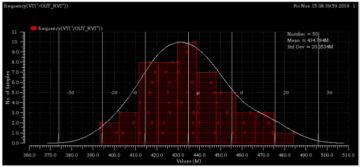
Include in report???

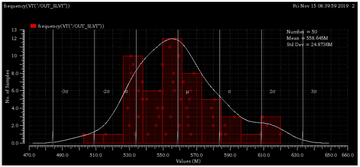
- Current report is relatively agnostic to implementation. Should this be included as it is a very implementation specific thing?
- Perhaps can reserve for later portion of project.

Ring oscillator in 22FDX

Ring oscillator simulation - variation

- Simulated simple 5 stage ring oscillator with V_{DD} = 0.8.
- Used RVT/SLVT devices, varied temp in {-40, 25, 85}, L in {100n, 500n}, W/L in: {1, 10}, C_{load} in {0, 1f, 10f}, V_{BB} in {0, 0.8}
- Also perfomed Monte-Carlo at 25C with L=500nm and W/L=1:
- Standard deviation for RVT is 4.7% of nominal value, for SLVT it is 4.5% of nominal.





Specification (unchanged)

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	
FSK BER	$\leq$ 1e-2		2FSK with 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	50	kHz	(nominally), tunable

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

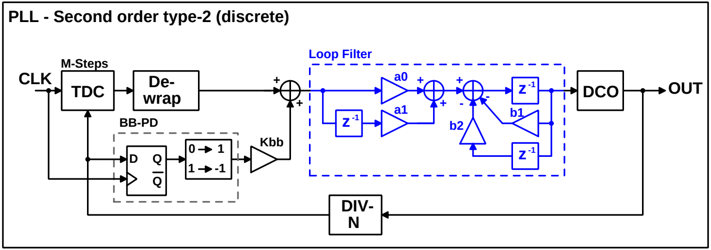
Specification (unchanged)

PLL Component Performance Targets

Parameter	Value	Unit	Notes
DCO LSB Resolution	$\leq$ 50	kHz	Determined from quantization noise.
DCO DNL	< 1	LSB	Ensures monotonicity
TDC Resolution	0.95	ns	
TDC Resolution (bits)	6	bits	

Architecture (updated)

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.
- Estabilish 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.

References

[1] "Ultra-Low Power Wake-Up Receivers for Wireless Sensor Networks", N. Pletcher, J.M Rabaey, 2008.
`http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-59.html`