



DESIGN **AUTOMATION** CONFERENCE

Modeling and Simulation of an All Digital Phase Locked Loop

Russell Mohn
Epoch Microelectronics Inc., Tarrytown, NY

Motivation

- Explain how Matlab/Simulink can be used to model and simulate all-digital PLLs
 - Construct 2 useful models
 - phase domain
 - time domain
- Explain how function block specifications can be verified from top level PLL specifications

Overview

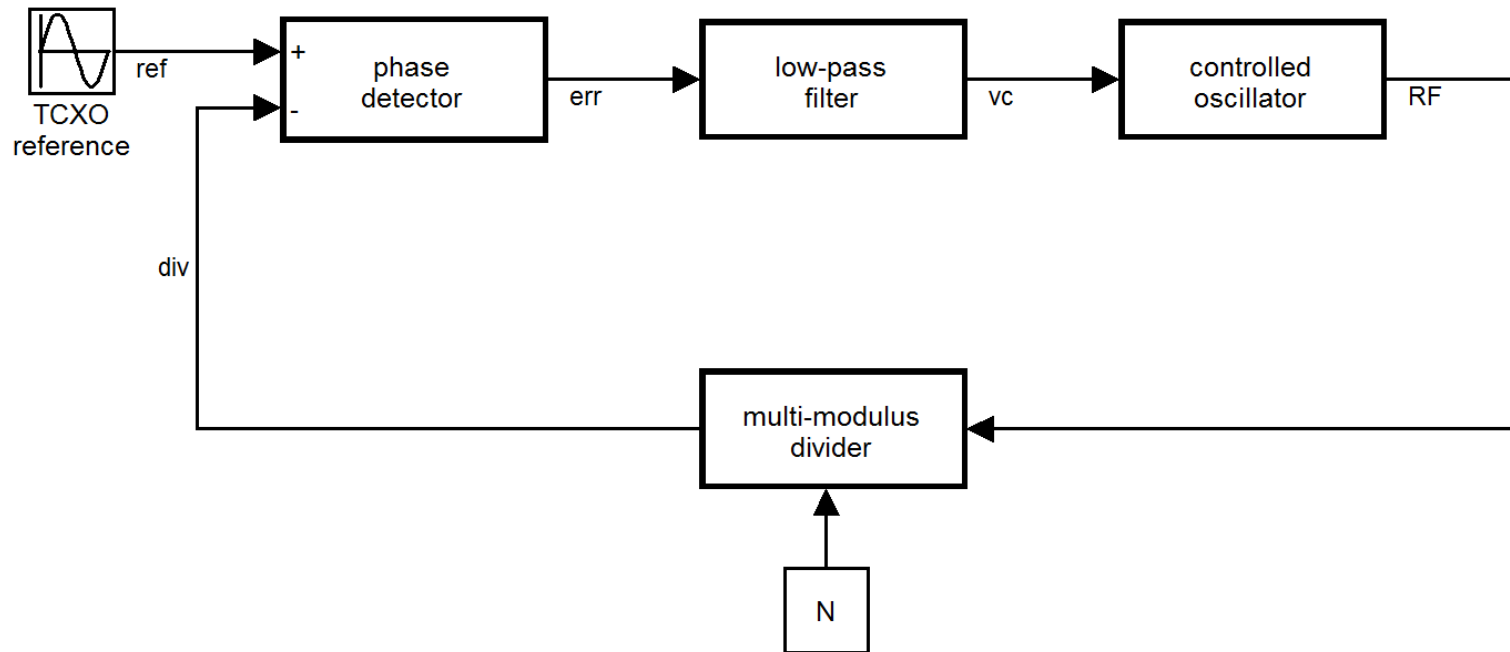
- Brief background on PLLs
- Challenges of PLL Simulation
- PLL Modeling (solution to the challenges)

Background

- What is a PLL and how does it work?
- Brief history of PLL development
- PLL applications

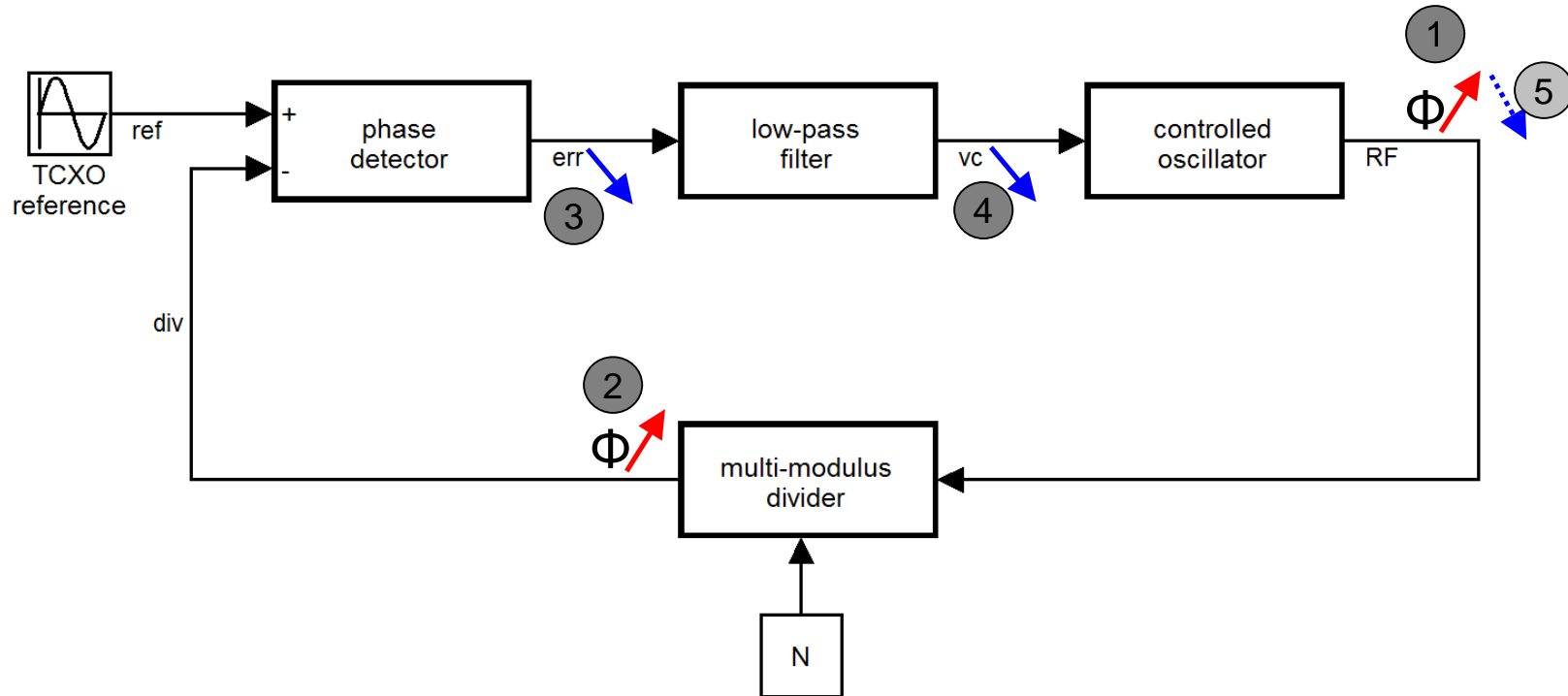
PLL Qualitative Description

- Function blocks of a phase-locked loop (PLL)



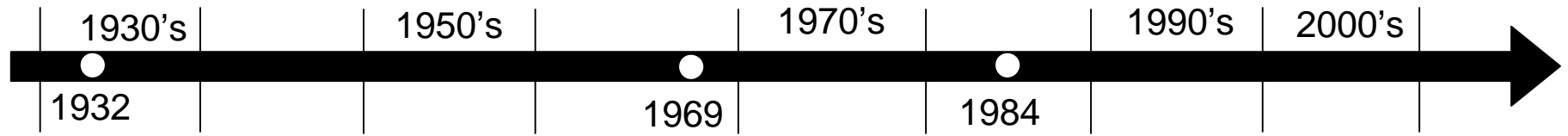
- Closed loop system using negative feedback
- Forces $RF = N \cdot ref$, and a fixed phase relationship between 'ref' and 'RF'

PLL Qualitative Description



- If RF phase drifts ahead of $N \cdot \text{ref}$'s phase
 - Phase detector 'err' signal forces controlled oscillator to slow down
- If RF phase drifts behind $N \cdot \text{ref}$'s phase
 - Phase detector 'err' signal forces controlled oscillator to speed up

Brief History of PLL Development



- 1932: Henri de Bellescize describes the technique in the French journal *L'Onde Électrique*
British scientists build the first PLL to counteract LO drift in homodyne receivers
- 1930's: Analog television receivers use PLLs to synchronize local sweep signal timing to transmitted pulses
- 1969: Signetics introduces a PLL on a chip - applications for the PLL multiply
- 1970's: - RCA introduces the CD4046, a CMOS PLL which became popular
- analog-PLLs → digital PFD's → **all-digital PLLs**
- 1984: John Well's (Marconi Instruments) fractional-N synthesis using noise-shaping
- 2009: Continued innovation and development is evidence of PLL's usefulness

PLL Applications

- Radio
- Television
- Telecommunications
- Computers



Challenges of PLL Simulation

- PLL specifications
- PLL design challenges
- Existing simulation software for PLLs

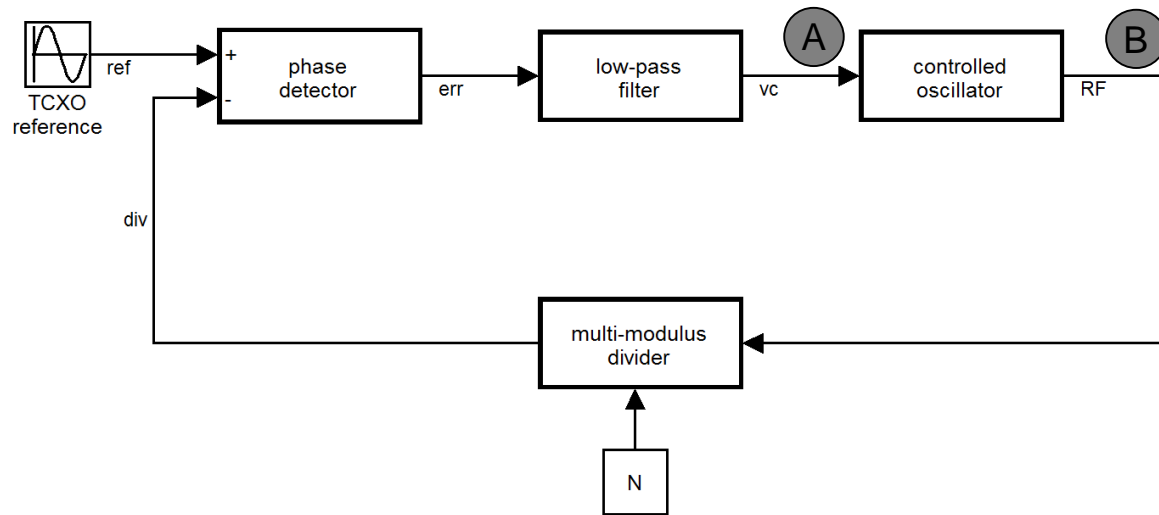
Typical PLL Specifications

	Spec.	[unit]
Reference	50	MHz
Frequency range	2000 - 3400	MHz
Frequency spacing	500	kHz
Lock-up time	200	us
Integrated phase error	3	deg-RMS
Phase noise	-120	dBc/Hz @1MHz
Phase noise	-142	dBc/Hz @10MHz
Current consumption	20	mA

How to predict if a PLL design will meet specifications?

1. Build and measure → costly, time-consuming
2. Simulate → performance in frequency and time domains

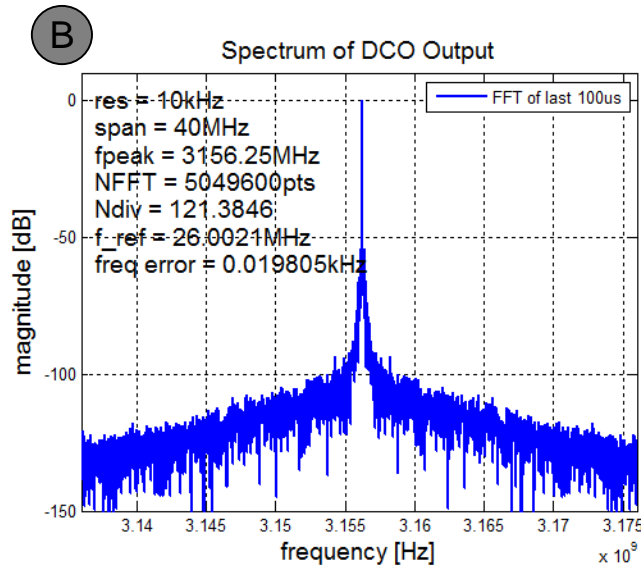
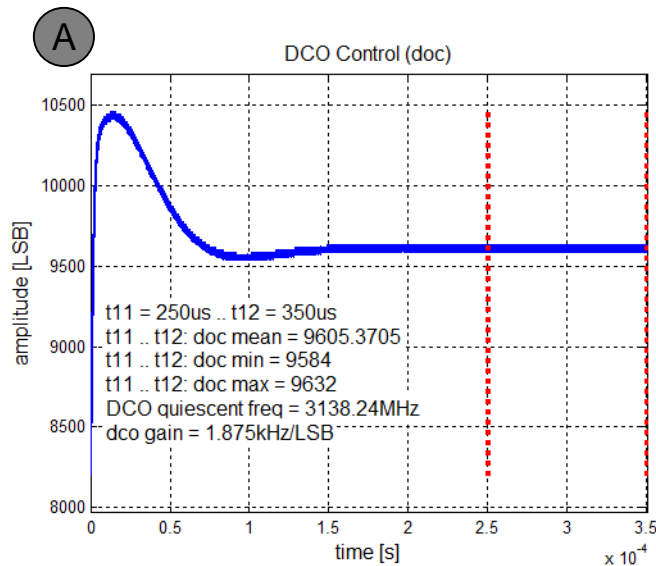
PLL Design Challenges



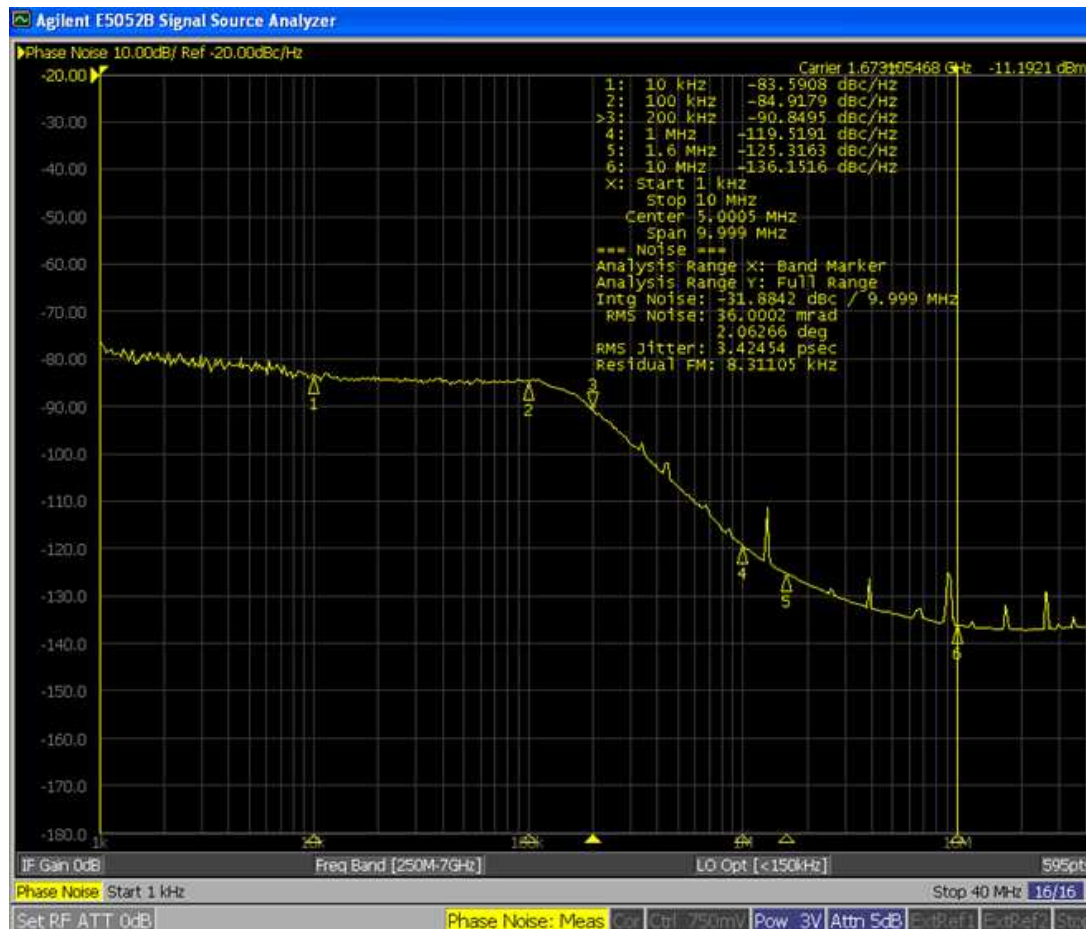
(B): femtoseconds of resolution
- phase noise

(A): microseconds of data
- settling, lock-up time

- 9 orders of magnitude in time-scale
- Simulation time potentially impossibly long



PLL Design Challenges



PLL phase noise measurement

→ Automatically taken care of by Agilent E5052B Signal Source Analyzer

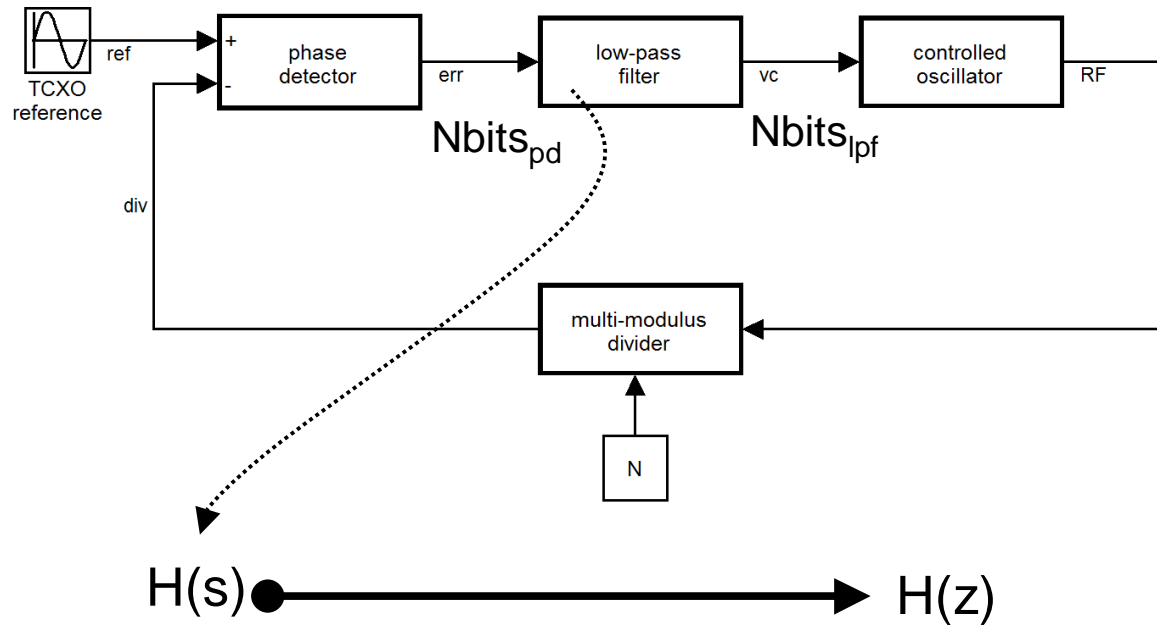
How to re-create and analyze in simulation?

$$\bar{\phi}^2 = \int_{f1}^{f2} S_{\phi}(f) df$$

$$rmsNoise = \sqrt{2 * \bar{\phi}^2} \text{ (rad)}$$

Need to determine $S_{\phi}(f)$, the power spectral density of the phase noise

Digital PLL Design Challenges



'Analog' PLL designer not necessarily comfortable with digital design concepts

- discrete time
- discrete amplitude

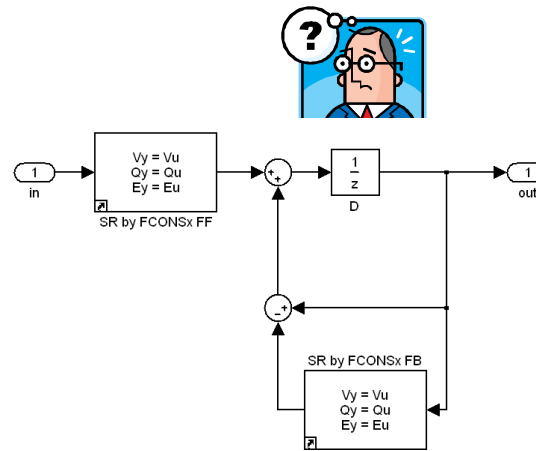
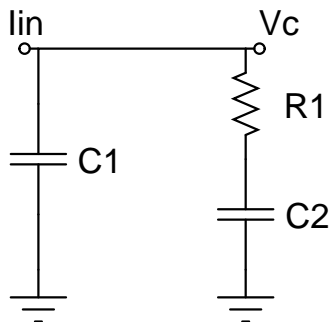
How many bits?

- $N_{bits_{pd}} = ?$
- $N_{bits_{lpf}} = ?$


How to build a phase detector with digital output?

Quantization noise

Overflow/Saturation

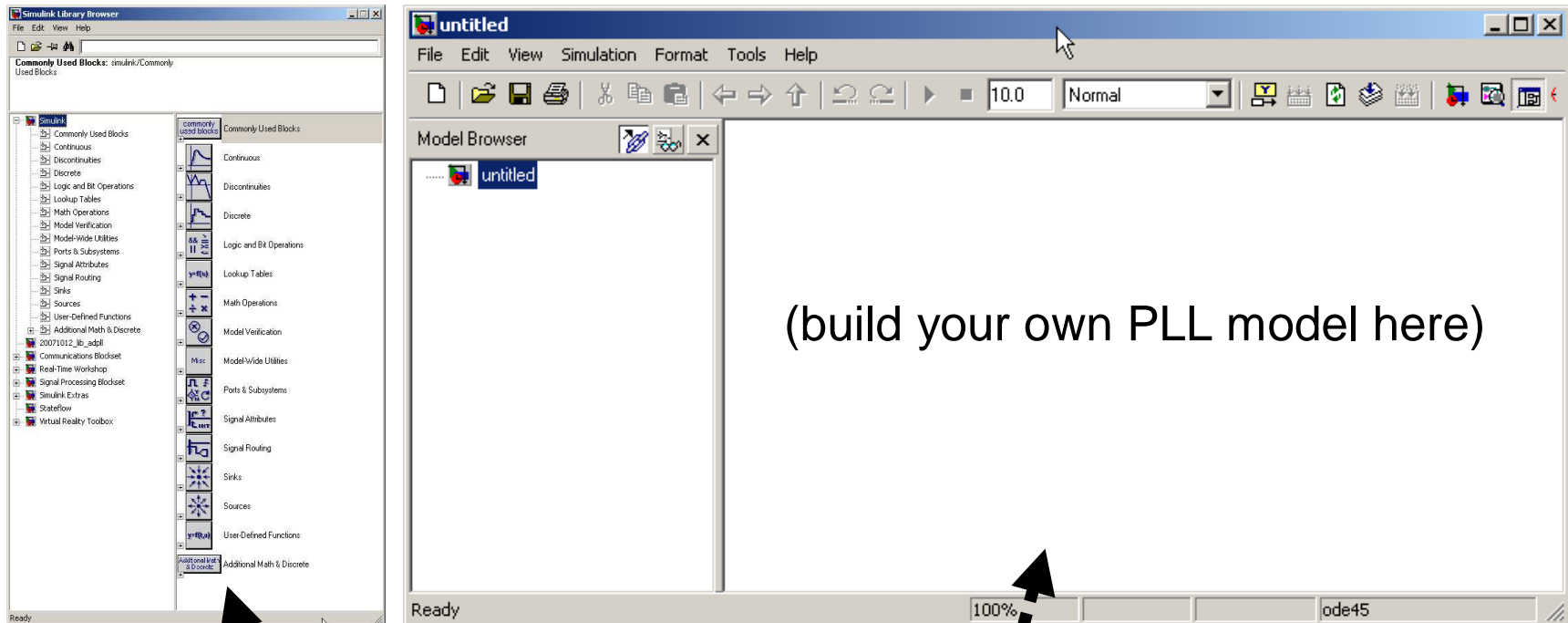


Existing PLL Simulation Software

Tool	Pros	Cons
CppSim (developed by M. Perrott at MIT)	- Fast simulation speed	- C++ based - learning curve
Verilog-A (full treatment on www.designers-guide.org)	- Used in SPICE tool environment	- Slower simulation speed
ADS (Agilent's Advanced Design System)	- display and analysis	- cost
SimPLL (Applied Radio Labs)	- integrates with existing PLL ICs	- limited flexibility
Matlab/Simulink 	- support for fixed point simulation - build behavioral and functional models - display and analysis	- learning curve

Focus of this talk

Matlab/Simulink



Building blocks

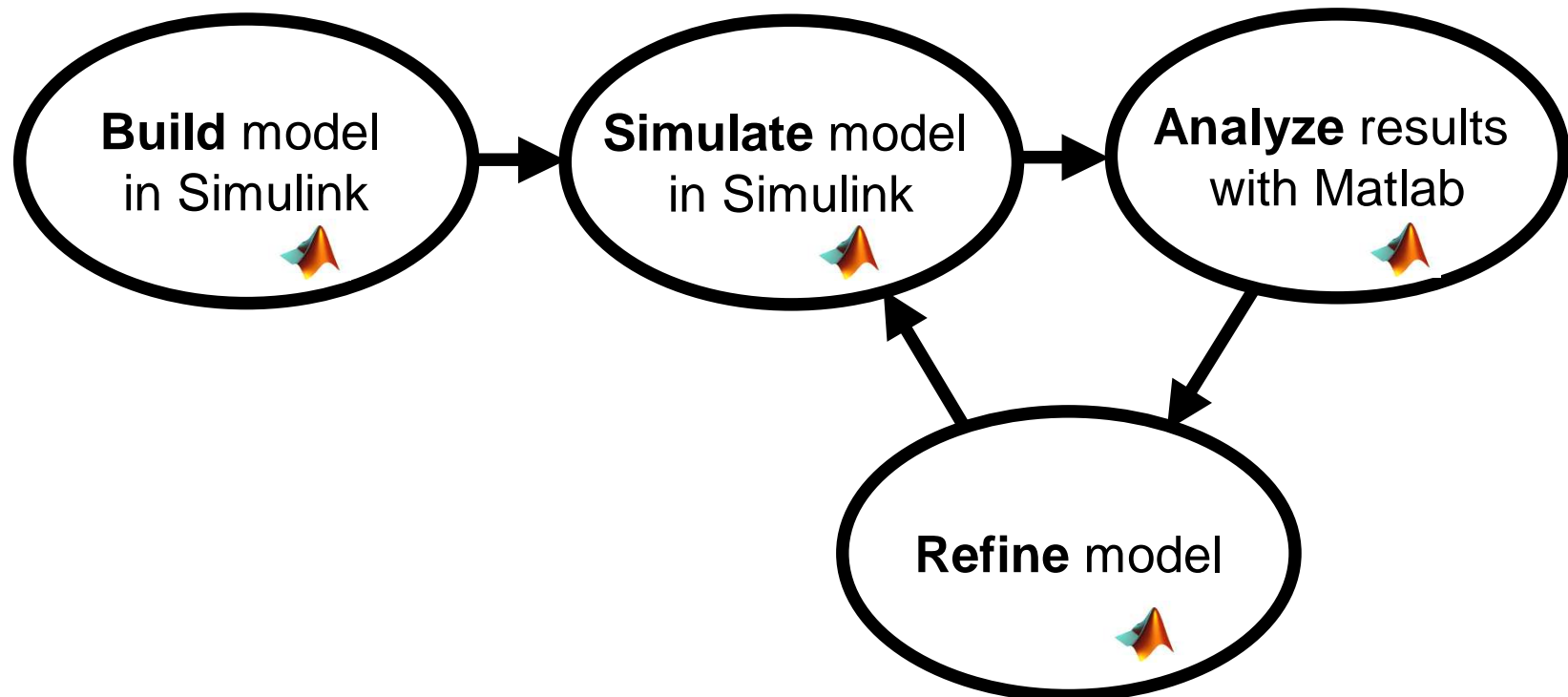
- mathematical operations
- logic operations
- user-defined functions
- memory elements
- pre-existing library blocks

Model is versatile

- behavioral
- functional
- floating point
- fixed point

PLL Modeling

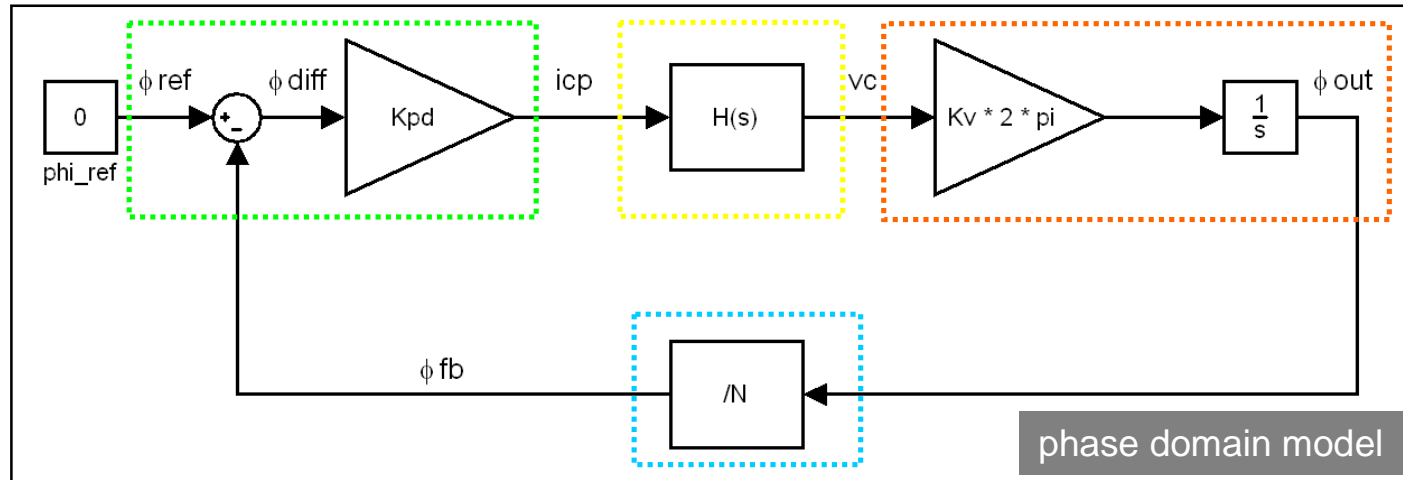
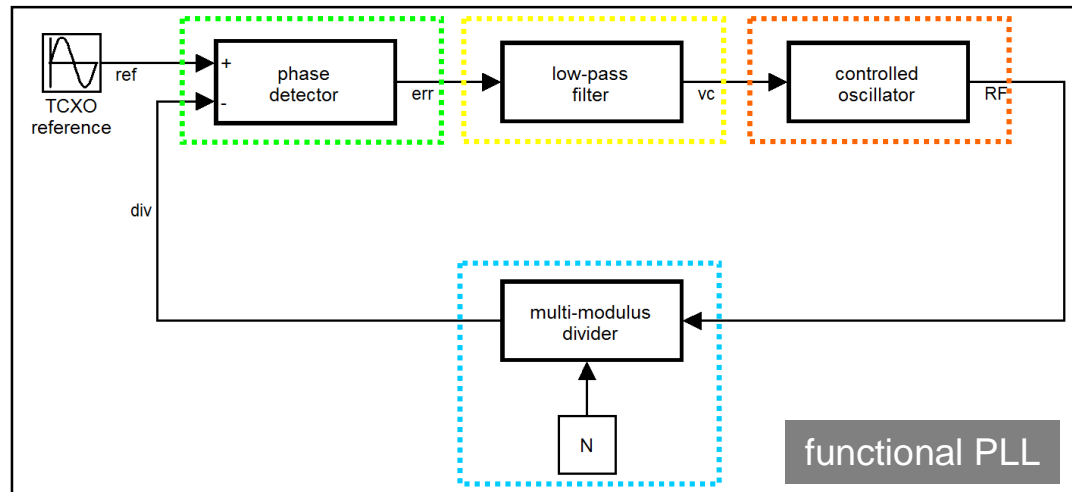
- Phase domain ADPLL model
- Time domain ADPLL model



Phase Domain Model (PDM)

Assume:

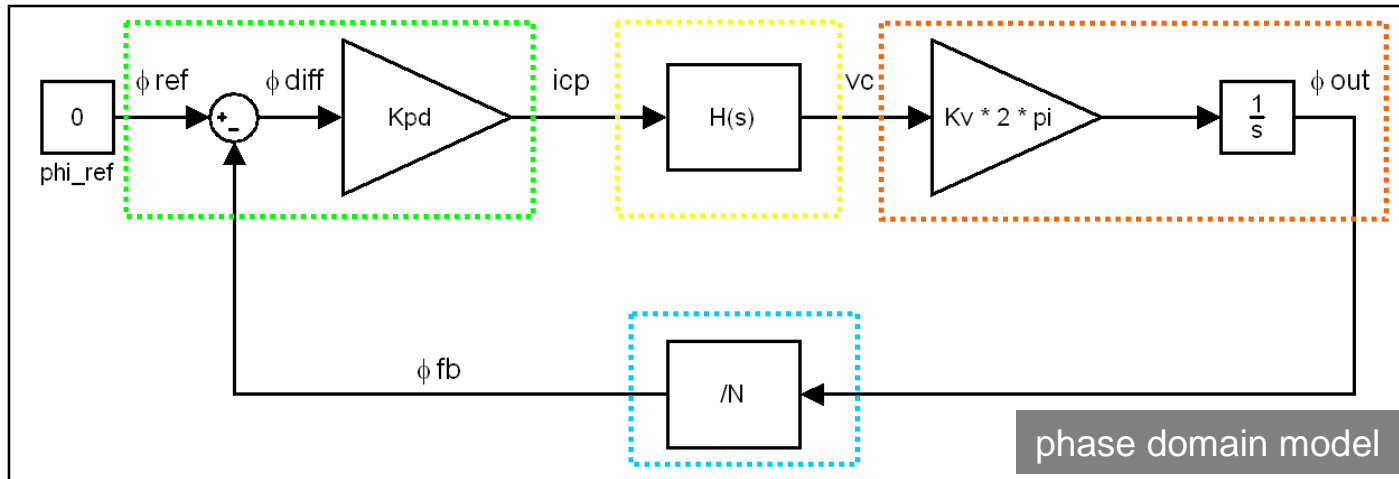
1. Variables represent the phase of the signal
2. Loop is locked
3. Noise is a small signal in a linear system



Explained in literature, eg:

1. Johns, David and Martin, Ken. *Analog Integrated Circuit Design*. Wiley & Sons, 1997.
2. Kundert, Ken. *Predicting the Phase Noise of PLL-Based Frequency Synthesizers*. www.designers-guide.org, 2005.

PDM - Transfer functions



Phase detector

$$icp = Kpd * \phi_{diff}$$

$$icp = Kpd * (\phi_{ref} - \phi_{fb})$$

Controlled oscillator

$$\phi_{out}(s) = \frac{2\pi K_v}{s} * vc(s)$$

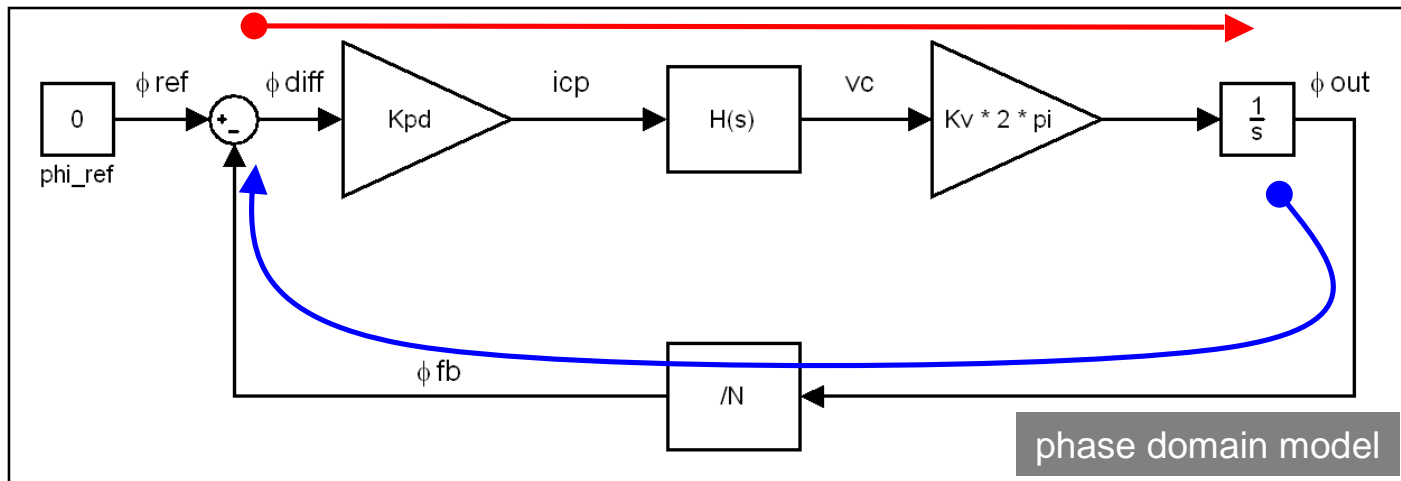
Loop filter

$$H(s) = \frac{vc(s)}{icp(s)}$$

Divider

$$\phi_{fb}(s) = \phi_{out}(s) / N$$

PDM – Check Stability



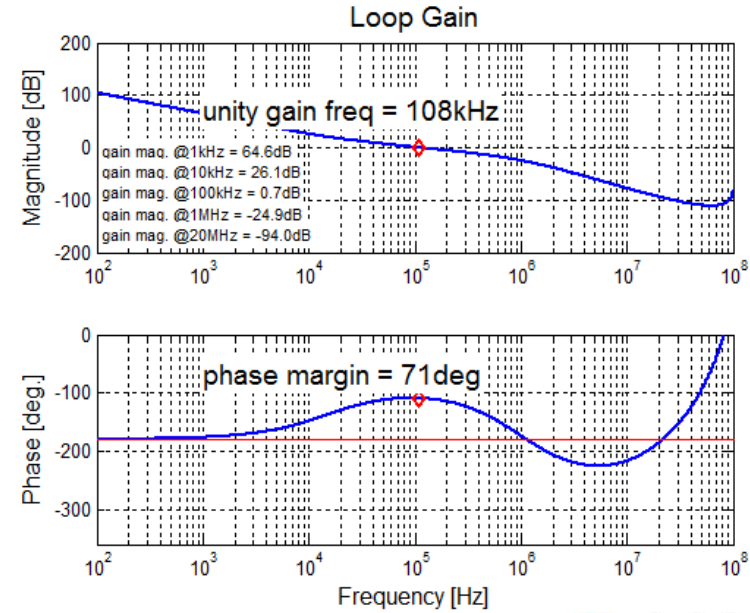
Forward gain (A)

$$A = Kpd * H(s) * 2\pi * Kv / s$$

Feedback gain (B)

$$B = 1 / N$$

Loop Gain = AB

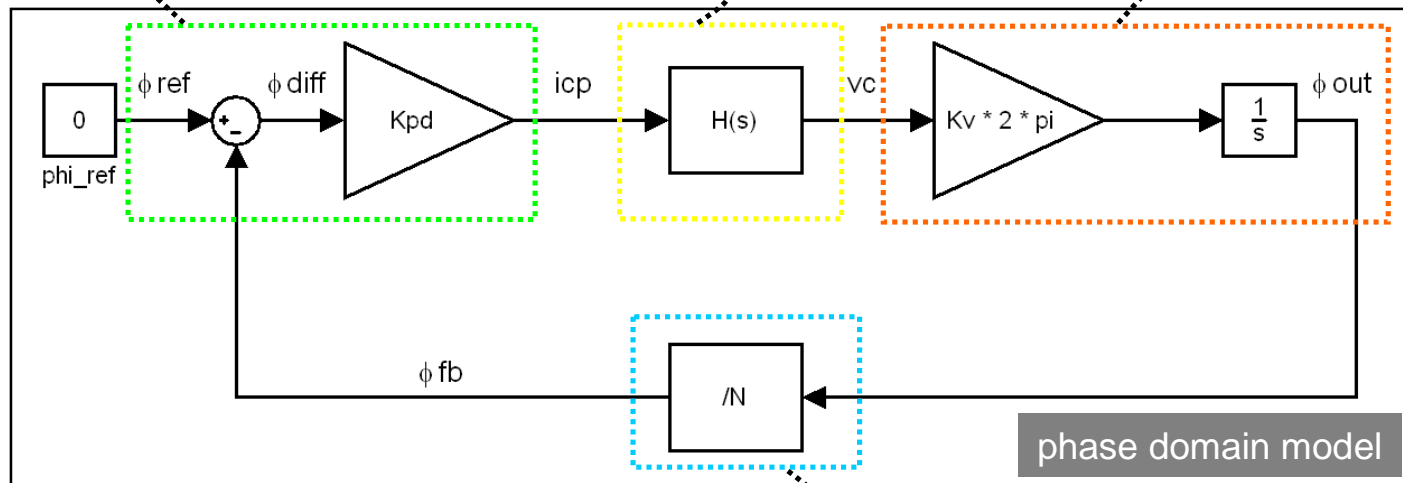


Digital model based on analog

PFD \rightarrow *DPFD*
quantized phase difference

LPF \rightarrow *DLPF*
 $H(s) \rightarrow H(z)$
discrete time filter

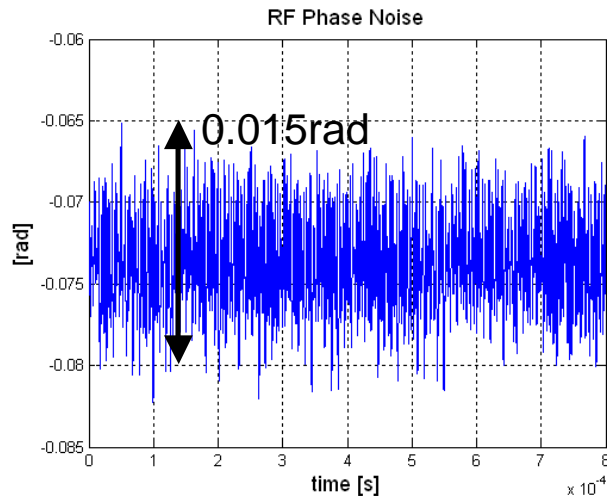
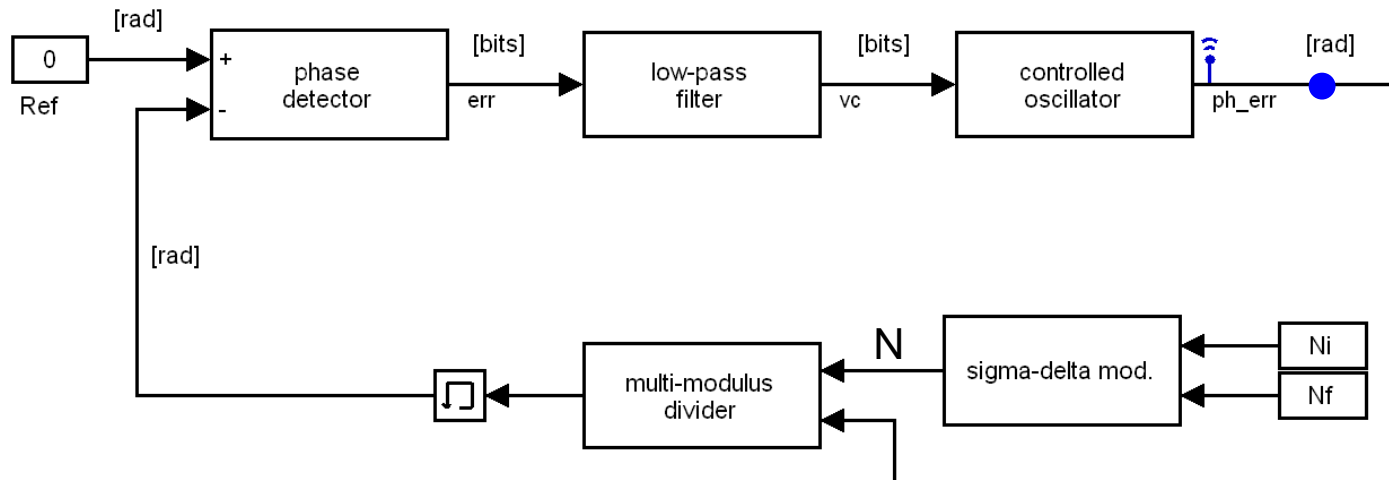
VCO \rightarrow *DCO*
quantized frequency step



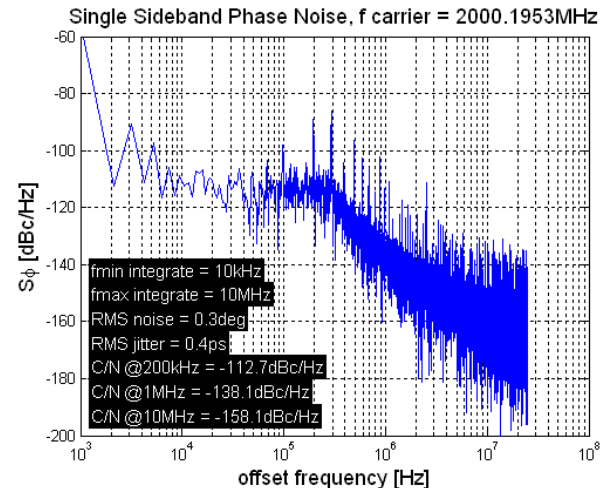
divider \rightarrow similar to analog
use sigma-delta modulator for
fractional synthesis

time & amplitude:
continuous (analog) \rightarrow discrete (digital)

ADPLL Phase Domain Model



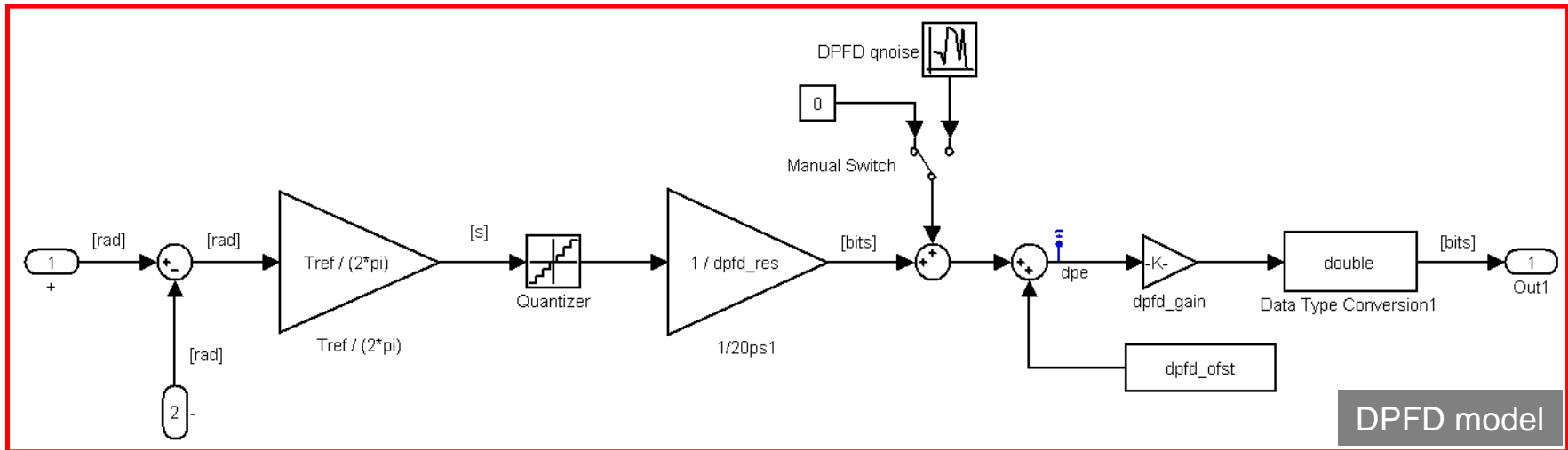
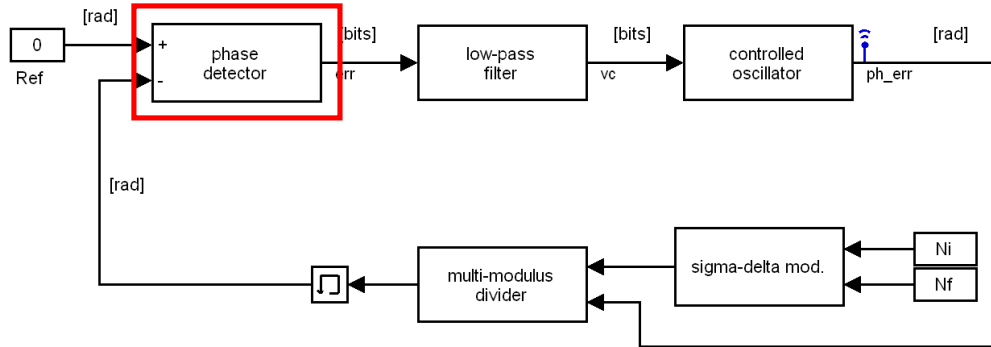
PSD



Each function block

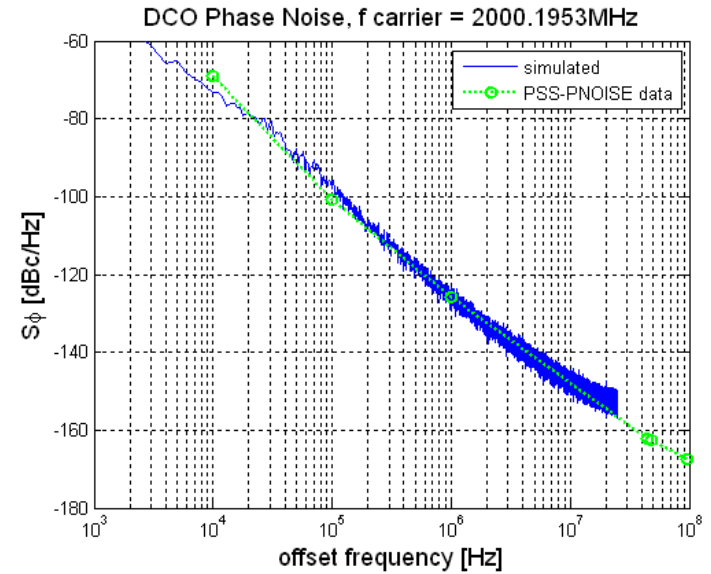
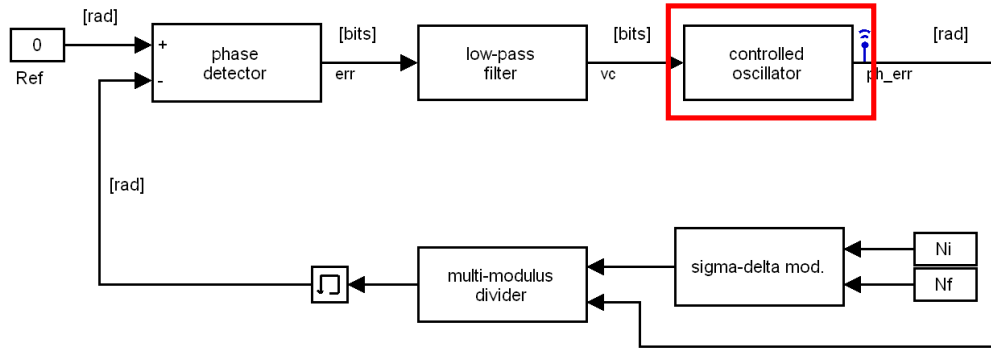
- performs its phase domain function
- adds noise associated with its function

PDM model for DPDFD



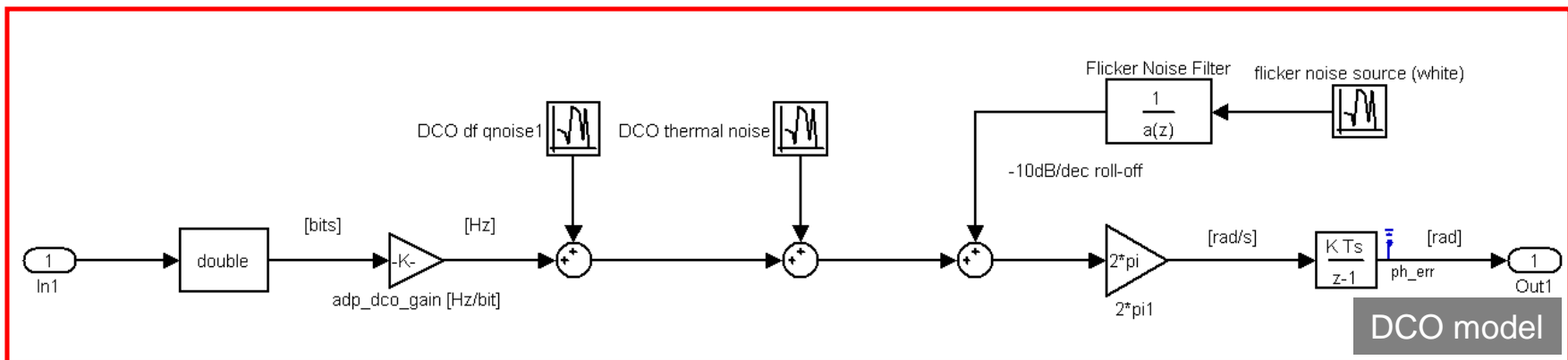
Adds quantization noise

PDM model for DCO



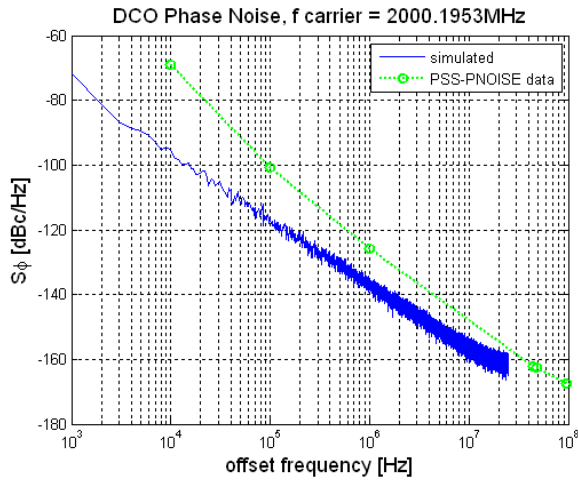
Target C/N from circuit-level simulation

Determine noise source variances through trial-and-error

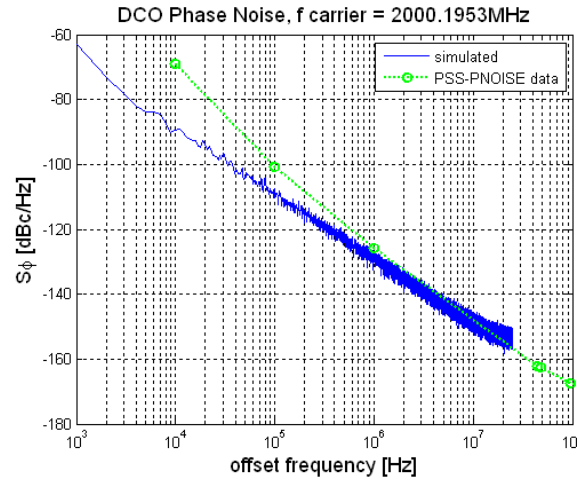


DCO Noise Contributors

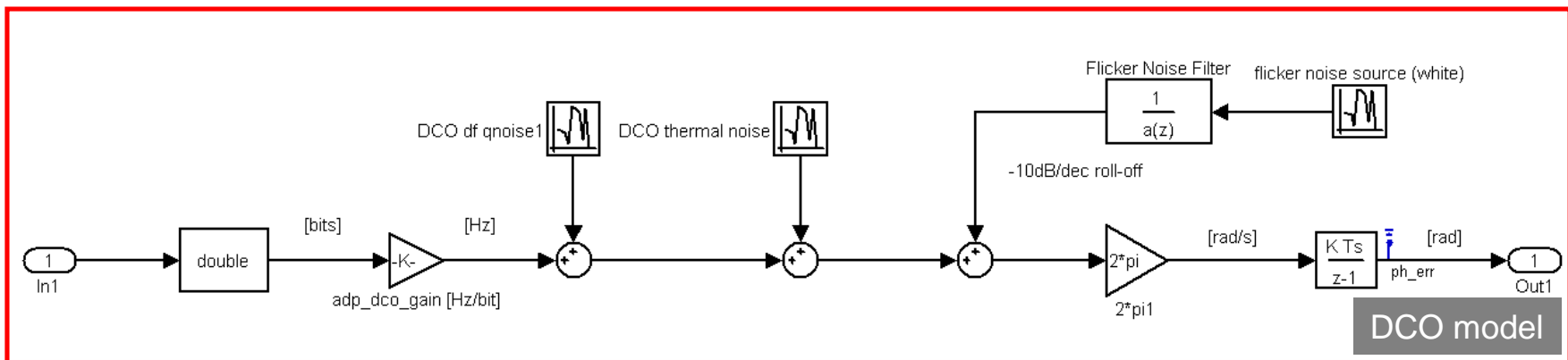
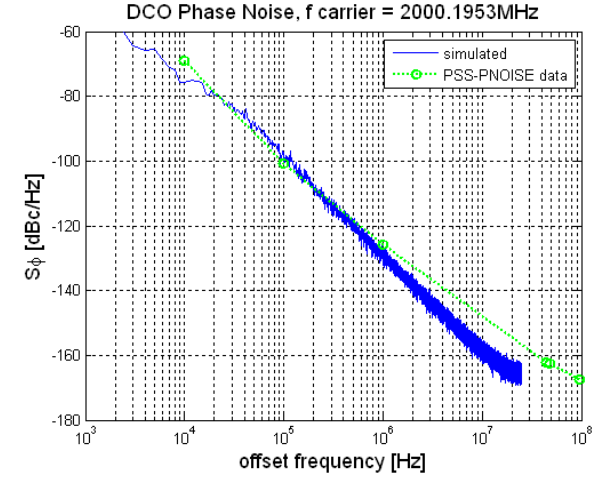
Quantization



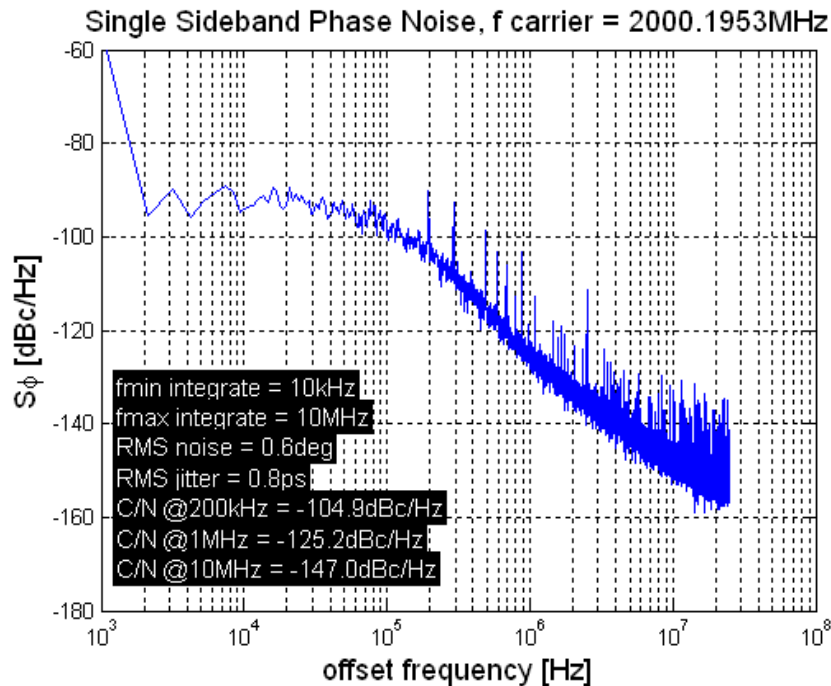
Thermal



Flicker



PDM – ADPLL Phase Noise



- Simulation time ~ 1min.
 PDM can accurately simulate
- integrated phase error
 - far-away phase noise

	Spec.	[unit]
Reference	50	MHz
Frequency range	2000 - 3400	MHz
Frequency spacing	500	kHz
Lock-up time	200	us
Integrated phase error	3	deg-RMS
Phase noise	-120	dBc/Hz @1MHz
Phase noise	-142	dBc/Hz @10MHz
Current consumption	20	mA



Sub-block specs. → ADPLL specs.

DPFD

resolution = 39ps/b

DCO

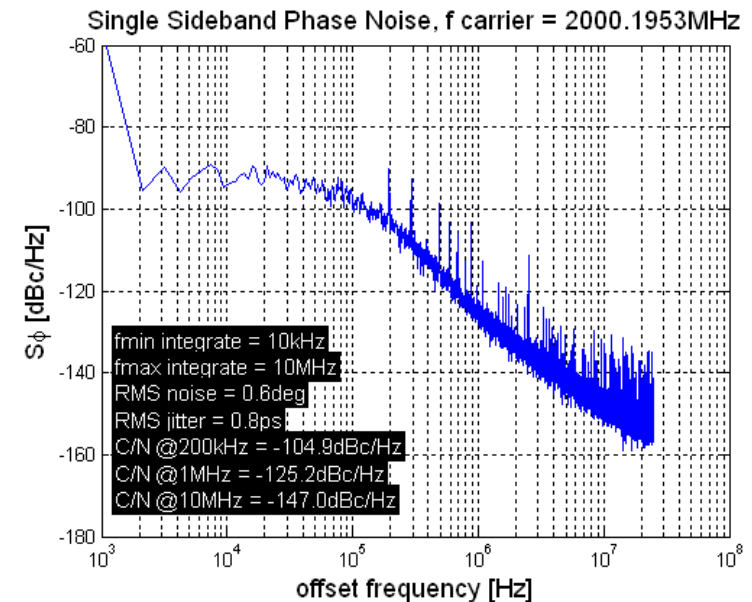
C/N = -70dBc/Hz @ 10kHz

C/N = -125dBc/Hz @ 1MHz

Gain = 2.5kHz/b

Can I relax the performance of the sub-blocks?

	Spec.	[unit]
Reference	50	MHz
Frequency range	2000 - 3400	MHz
Frequency spacing	500	kHz
Lock-up time	200	us
Integrated phase error	3	deg-RMS
Phase noise	-120	dBc/Hz @ 1MHz
Phase noise	-142	dBc/Hz @ 10MHz
Current consumption	20	mA



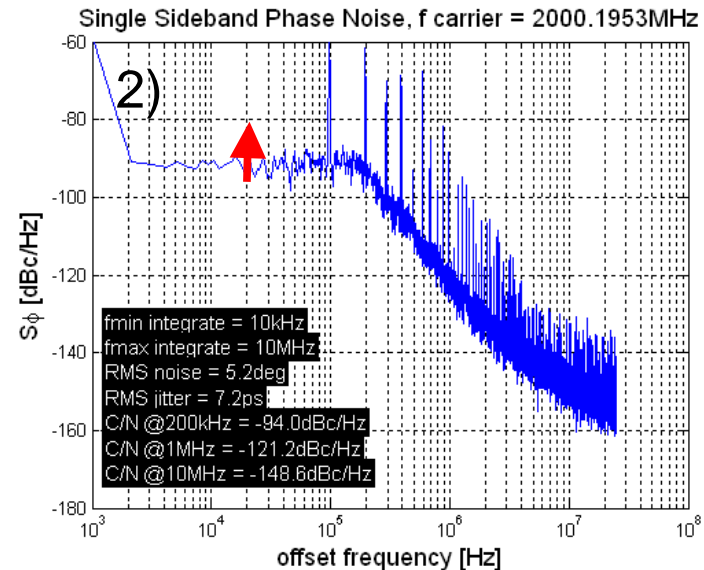
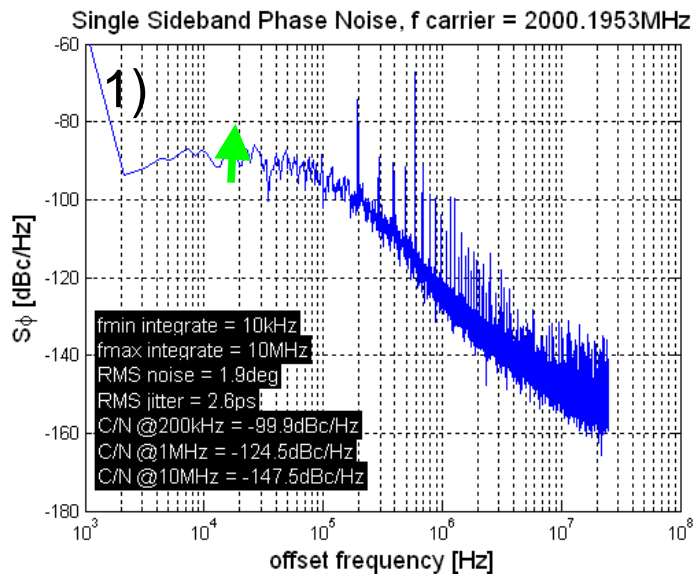
Choosing DPDFD Gain

DPDFD

- 1) Gain = 156ps/b → integrated phase error in spec.
- 2) Gain = 312ps/b → integrated phase error out of spec.

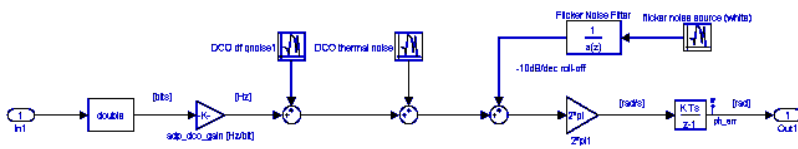
Choose DPDFD gain based on

- desired margin from phase error spec.
- DPDFD dynamic range requirement

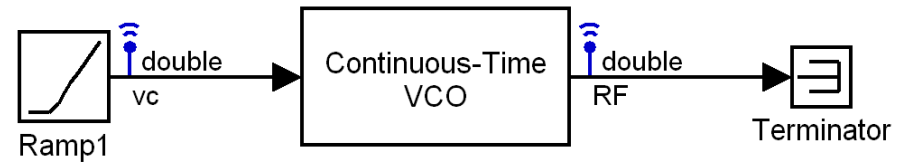


Time Domain Model (TDM)

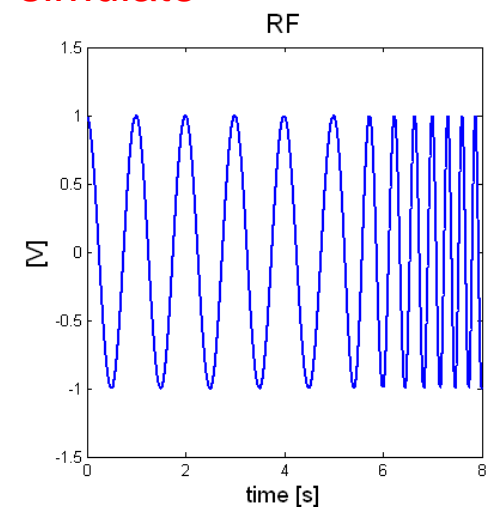
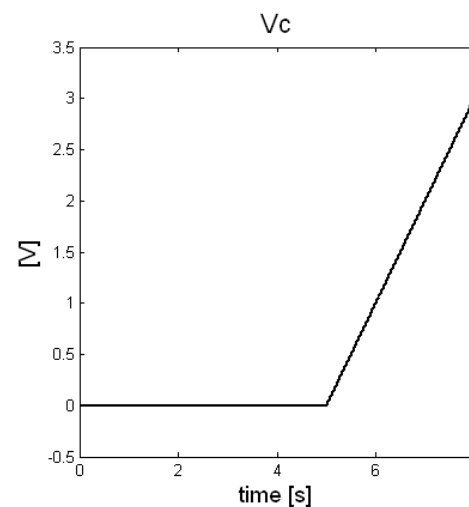
For example: the oscillator



simulate



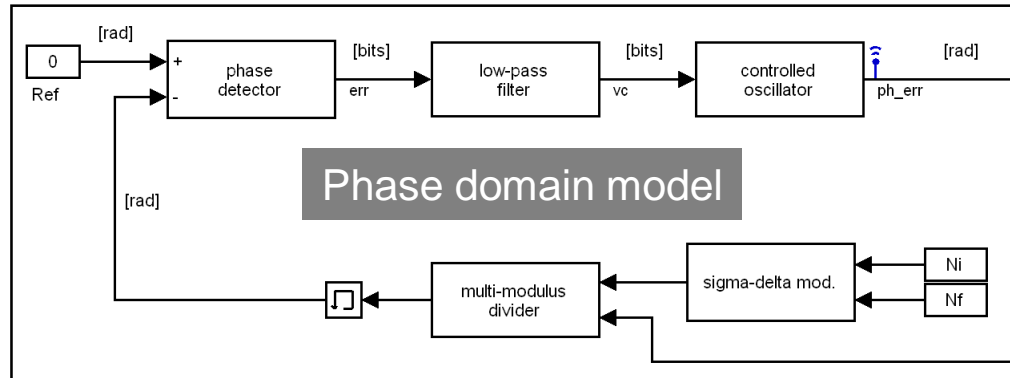
simulate



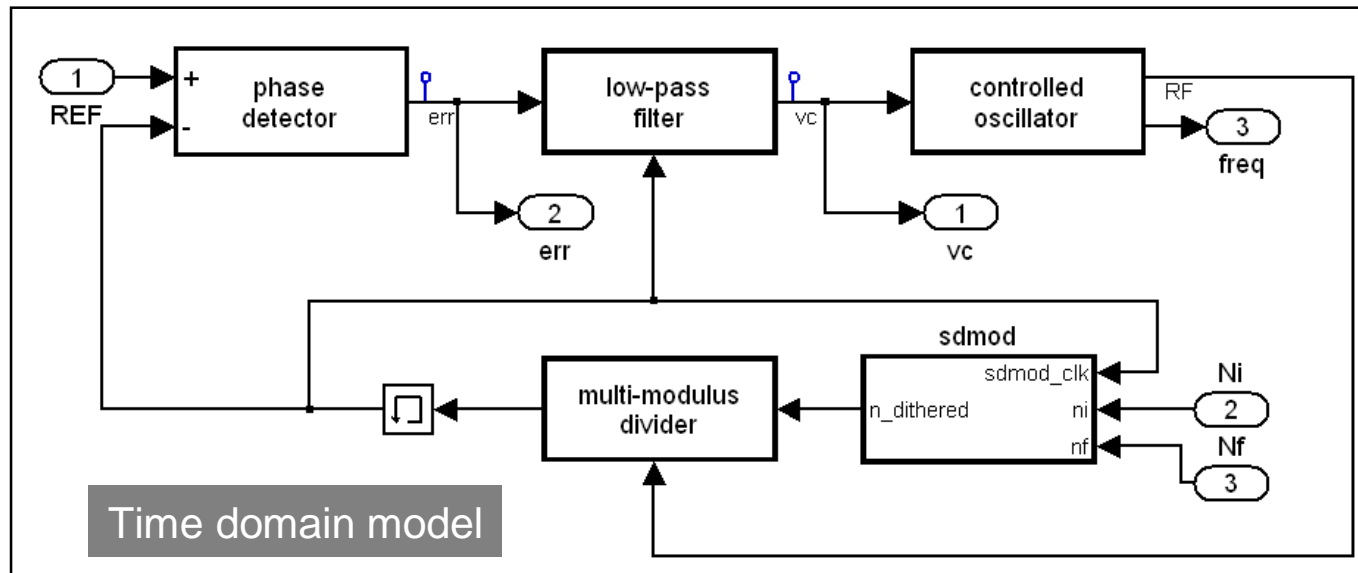
Motivation

- more intuitive
- closer to implementation
- large signal behavior

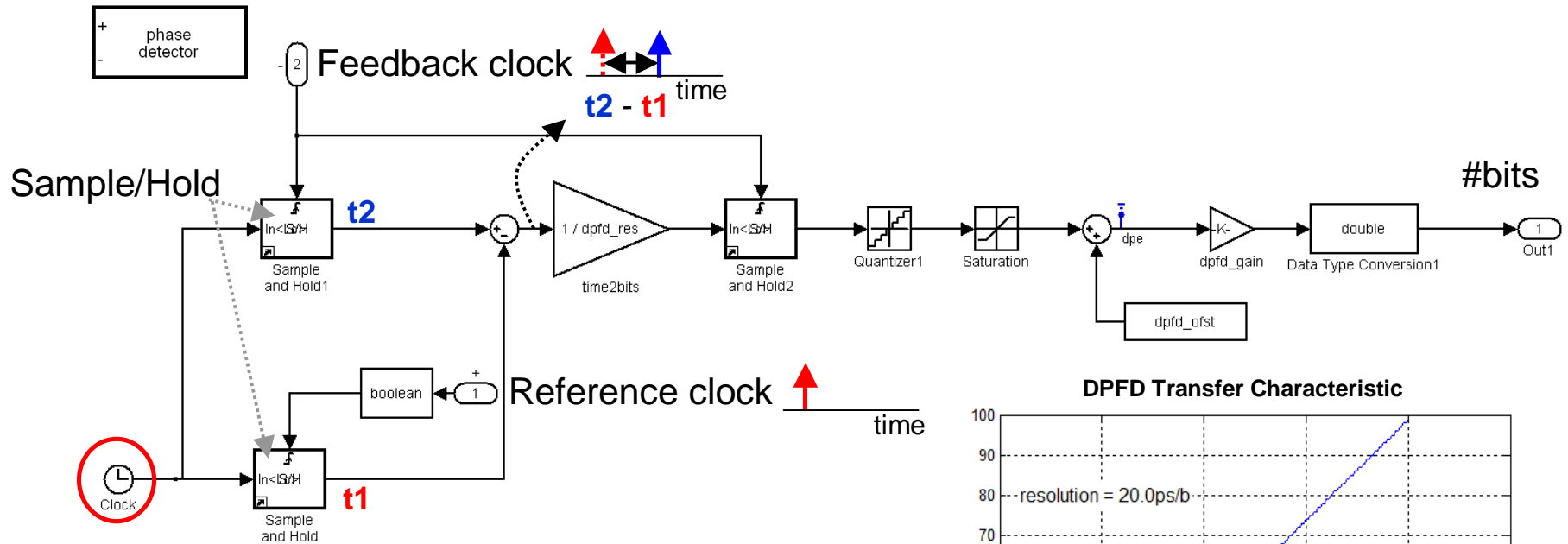
Time Domain Model (TDM)



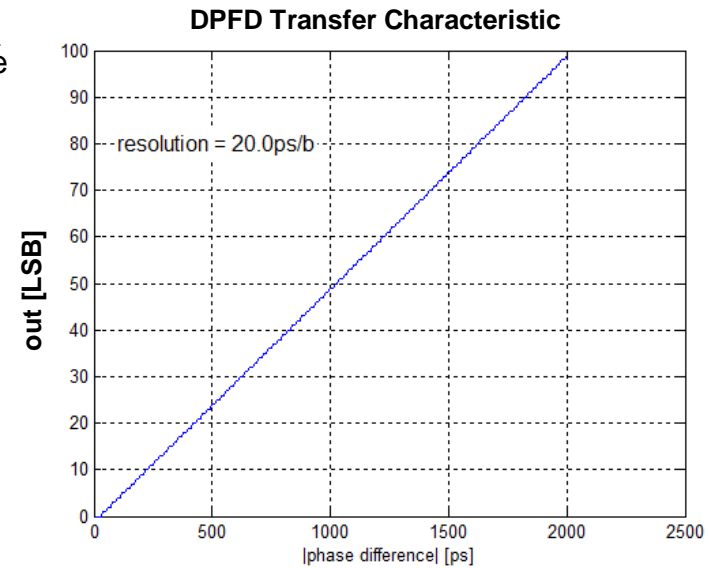
Same ADPLL structure
Different sub-blocks



Digital PFD Model

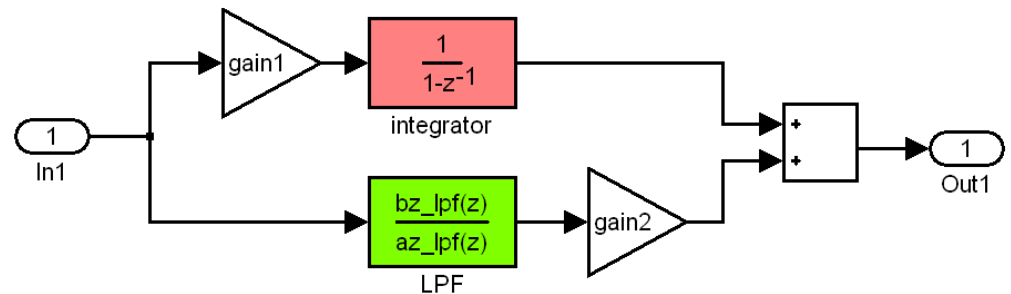
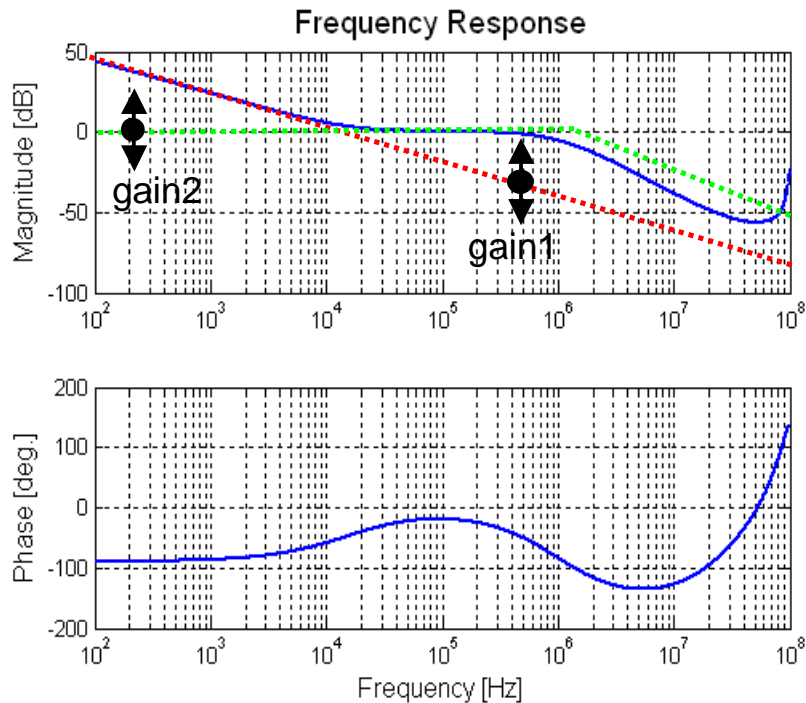


0, ..., **t1**, ..., **t2**, ...
 Clock = simulation time



Behavioral model
 Resolution limited by simulation step
 Phase difference is quantized → 'digital' PFD

Filter Design – Where to begin



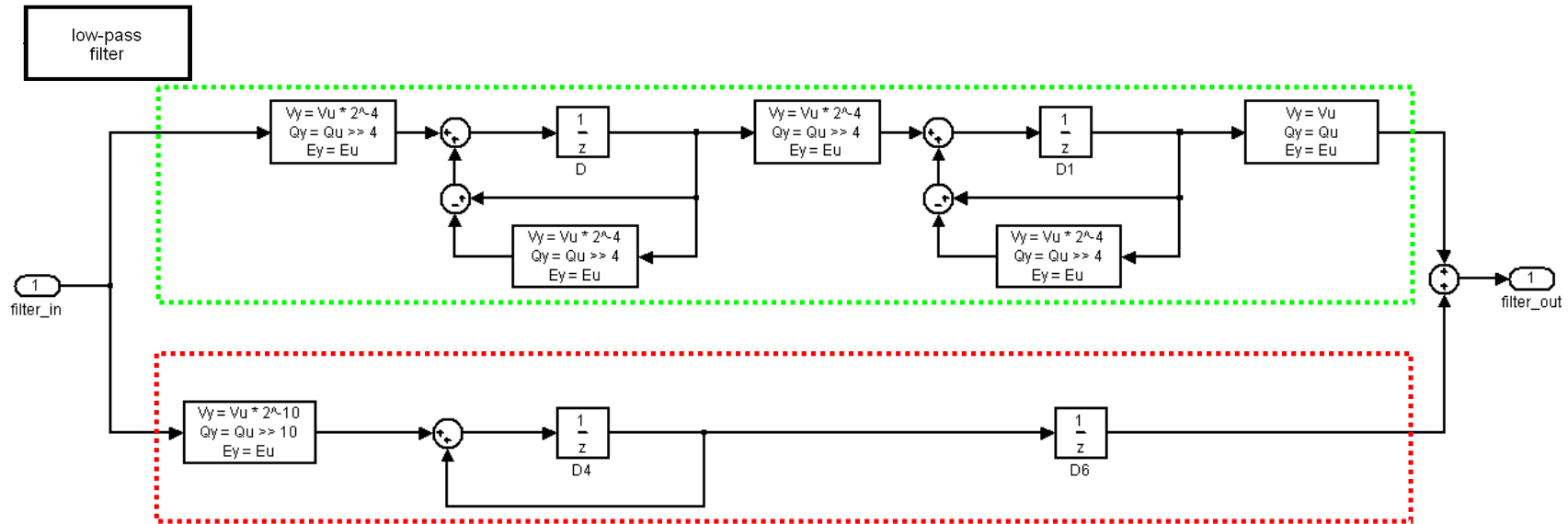
Design DLPF such that phase margin is reasonable, eg $> 60^\circ$

Lock-up time $\sim [1..5] / (\text{unity gain frequency})$

Lock-up time = 200us \rightarrow unity gain $> 25\text{kHz}$

- ➔ 1. desired response \rightarrow behavioral model
- 2. functional model

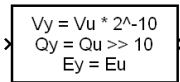
Filter Design – Functional model



D flip-flop

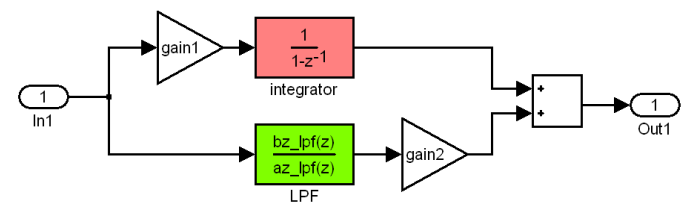


adder



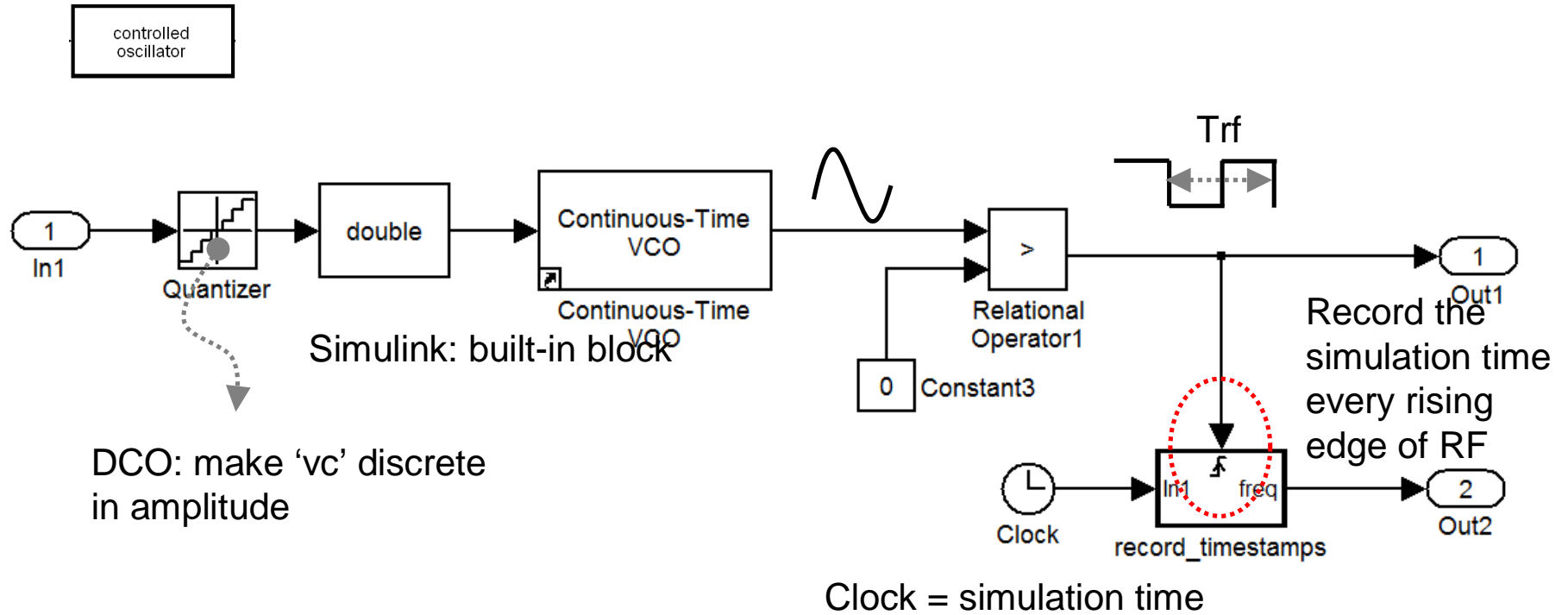
bit shift

is equivalent to:



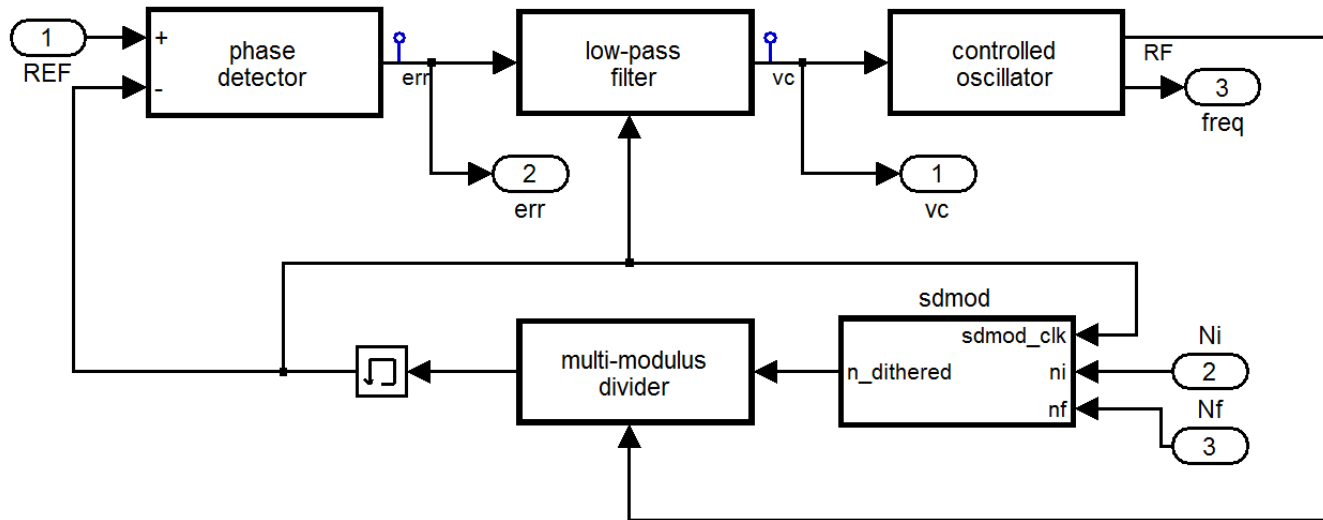
1. desired response → behavioral model
- ➔ 2. functional model

Oscillator Model

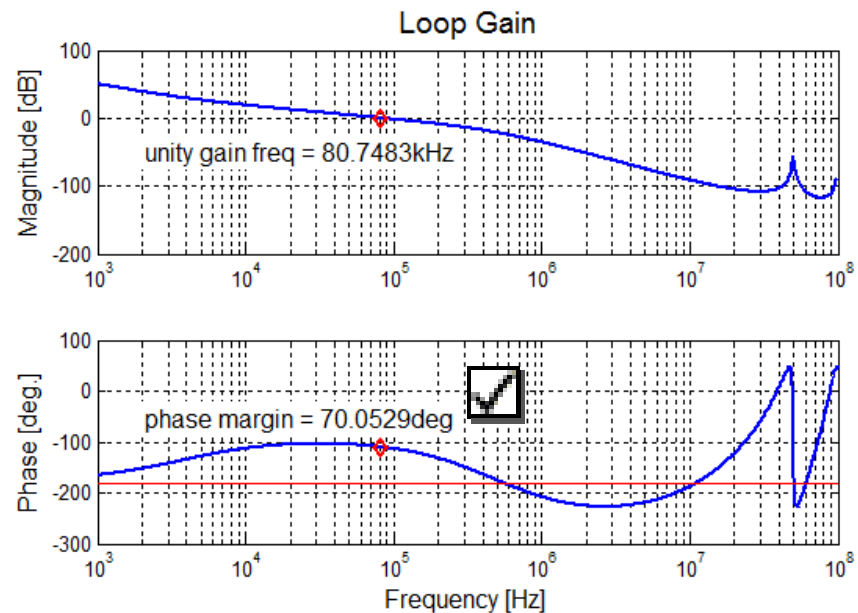


Use Simulink's variable-step solver
- 1.2ms of PLL operation in ~10 minutes

ADPLL Model Setup



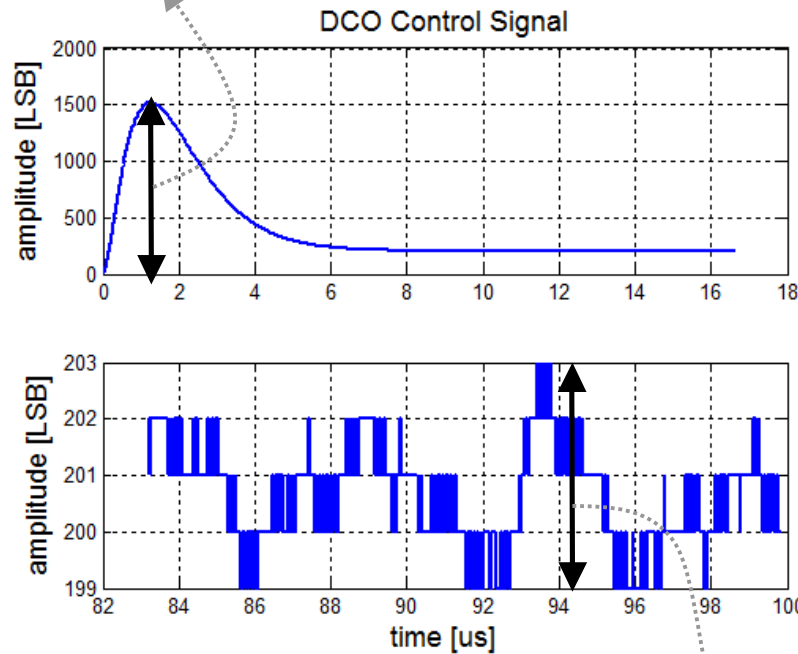
1. Set parameters:
 - DPFD gain
 - DLPF coefficients
 - Oscillator gain
 - N value
2. Check stability
3. Simulate with Simulink
4. Analyze data



ADPLL Transient Response

1500LSB

Loop is not settled yet



?
?
?

	Spec.	[unit]
Reference	50	MHz
Frequency range	2000 - 3400	MHz
Frequency spacing	500	kHz
Lock-up time	200	us
Integrated phase error	3	deg-RMS
Phase noise	-120	dBc/Hz @1MHz
Phase noise	-142	dBc/Hz @10MHz
Current consumption	20	mA

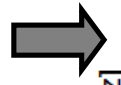
4LSB

$$2.5\text{kHz/b} * 4\text{LSB} = 10\text{kHz}$$

Loop settles in < 100us
(within $\pm 5\text{kHz}$)

ADPLL Phase Noise

1. calculate timestep errors → jitter
2. calculate PSD of jitter
3. calculate integrated phase error from PSD

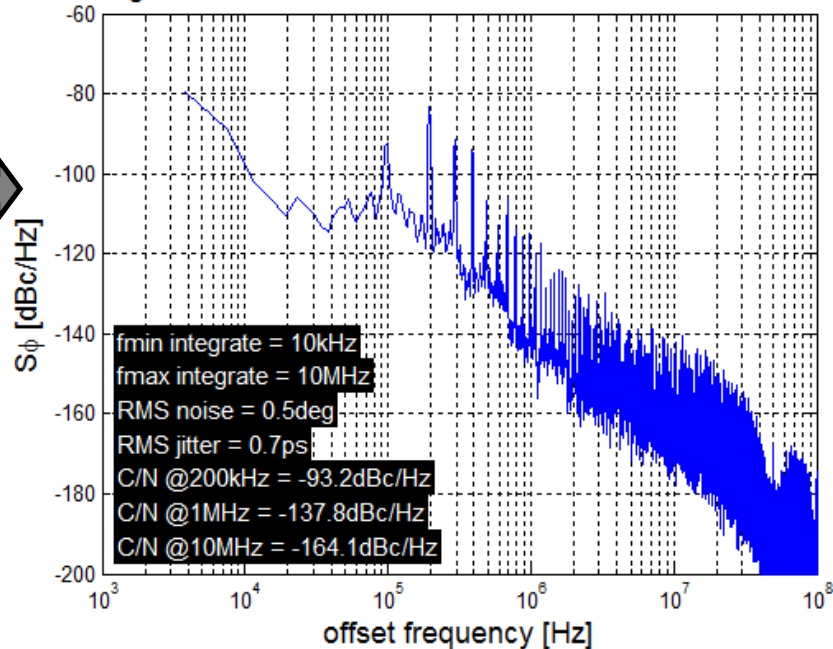


Time domain model is useful
Reasonable simulation time

→ Shows transient performance and gives some insight into frequency performance with one, intuitive model

Limitation: DCO thermal & flicker noise not modeled

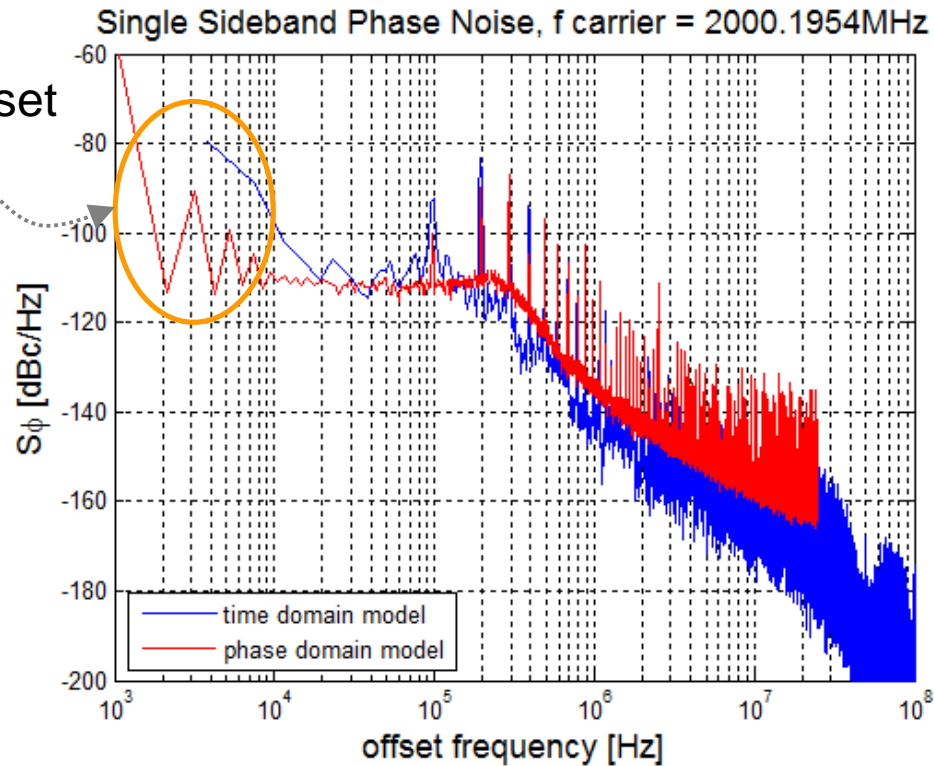
Single Sideband Phase Noise, f carrier = 2000.1954MHz



	Spec.	[unit]
Reference	50	MHz
Frequency range	2000 - 3400	MHz
Frequency spacing	500	kHz
Lock-up time	200	us
Integrated phase error	3	deg-RMS
Phase noise	-120	dBc/Hz @1MHz
Phase noise	-142	dBc/Hz @10MHz
Current consumption	20	mA

Do both models agree?

Need longer data set
for better low-
frequency
resolution



Both models have DCO quantization noise $\neq 0\text{W}$
thermal noise = 0W
flicker noise = 0W

The models are in fairly close agreement.

Conclusion

- Matlab/Simulink can be used to model and simulate ADPLLs
- Developed 2 models for ADPLLs
- From the phase domain model we can verify
 - Phase noise, phase error
- From the time domain model we can verify
 - Functionality
 - Lock-up time
 - phase noise (from quantization)
 - fixed point effects
- Models are used to derive sub-block specifications from ADPLL specifications

Please stop by our booth to view an ADPLL simulation demonstration.
Thank you!



DESIGN **AUTOMATION** CONFERENCE