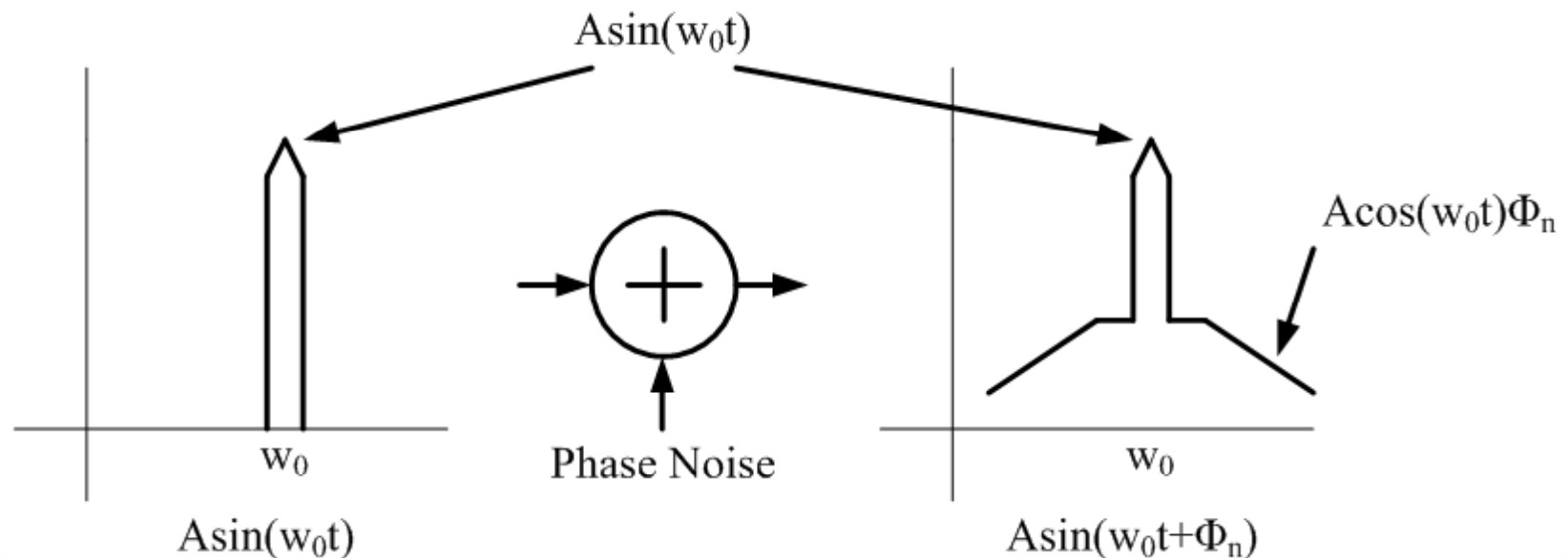


ELEN 6901
PLL Phase Noise/Jitter Modeling

Chun-Wei Hsu
Karthik Tripurari

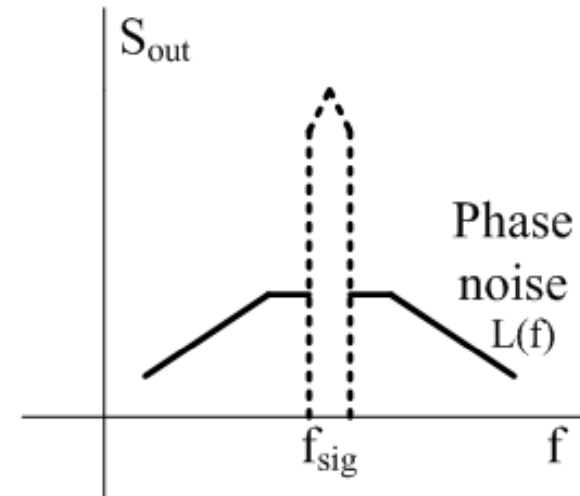
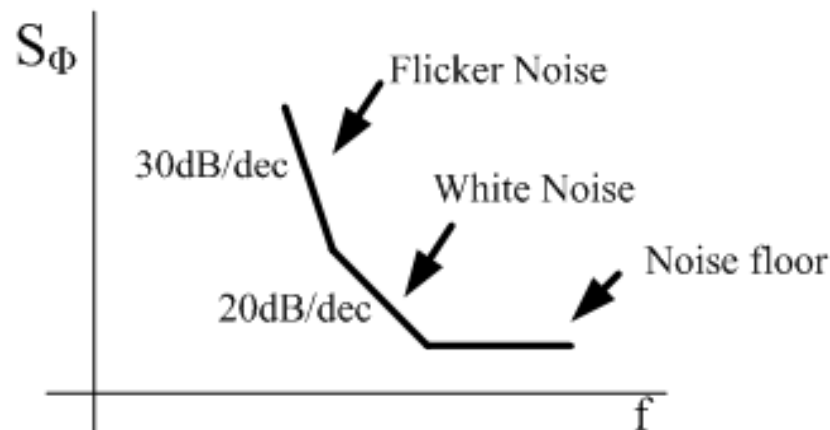
Introduction of Phase Noise

- The oscillator output V_o with the phase noise Φ_n
$$V_o = A \sin(\omega_0 t + \phi_n) = A \sin(\omega_0 t) \cos(\phi_n) + A \cos(\omega_0 t) \sin(\phi_n)$$
where A is the amplitude and ω_0 is the frequency
- For small noise term
$$V_o \cong A \sin(\omega_0 t) + A \cos(\omega_0 t) \phi_n$$



Introduction of Phase Noise

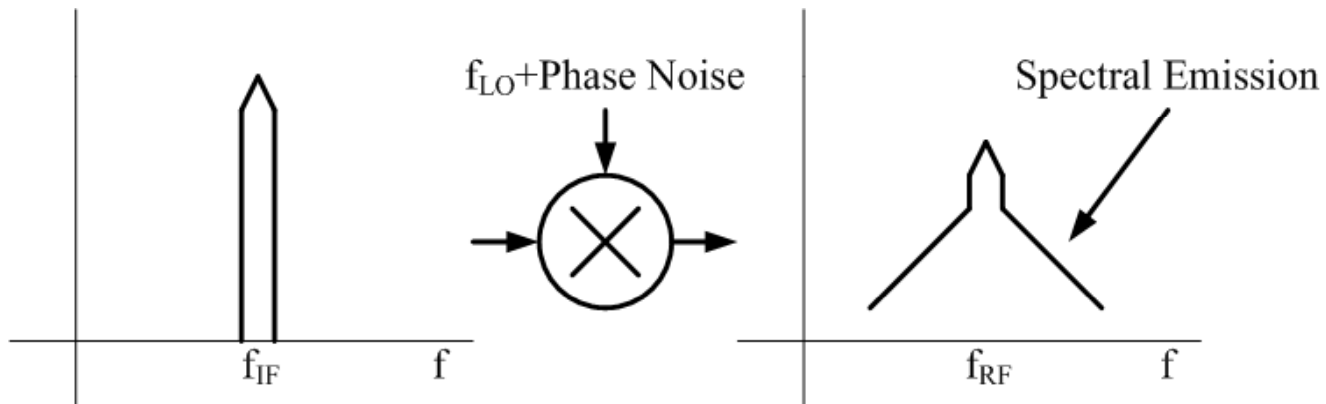
- Phase perturbation from device noise includes white and flicker noise
- Power spectral density S_{Φ} of phase noise Φ_n



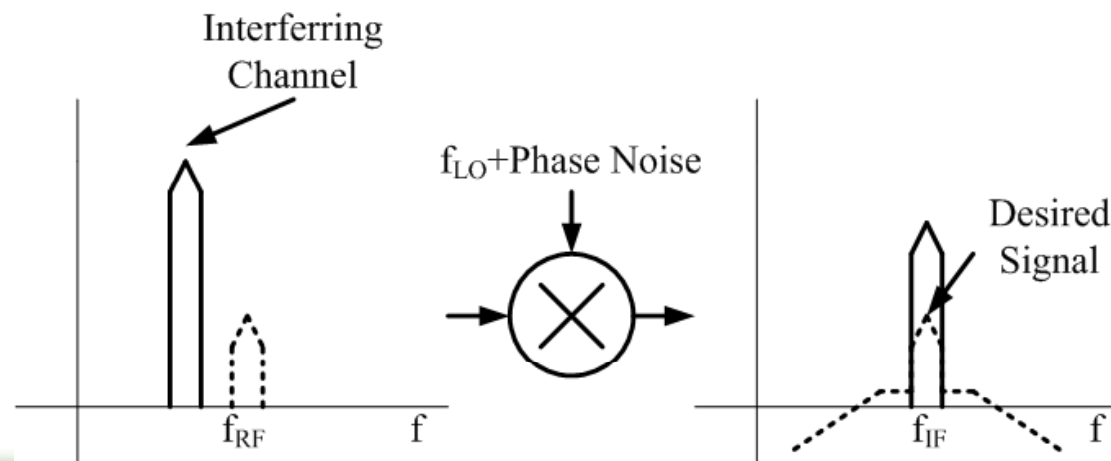
- Extracting noise parameters
 - Single-sided PSD $L(f)$
 - $L(\Delta f) = (1/2)S_{\Phi}(\Delta f)$

Impact of Phase Noise

- Phase Noise
 - Reduce SNR of the signal
 - Tx: Out-of band emission



- Rx: Reciprocal mixing



Phase Jitter

- Jitter issues
 - Sampling cases
 - Degrade SNR of ADC/DAC
 - Data recovery considerations
 - Reduce the eye-opening
 - Decrease BER
- Definition of phase jitter
 - Difference between the measured time and the ideal bit period

$$\sigma^2_{phase_jitter} = \left(\frac{1}{2\pi f_{sig}}\right)^2 \int S_{\Phi} df$$

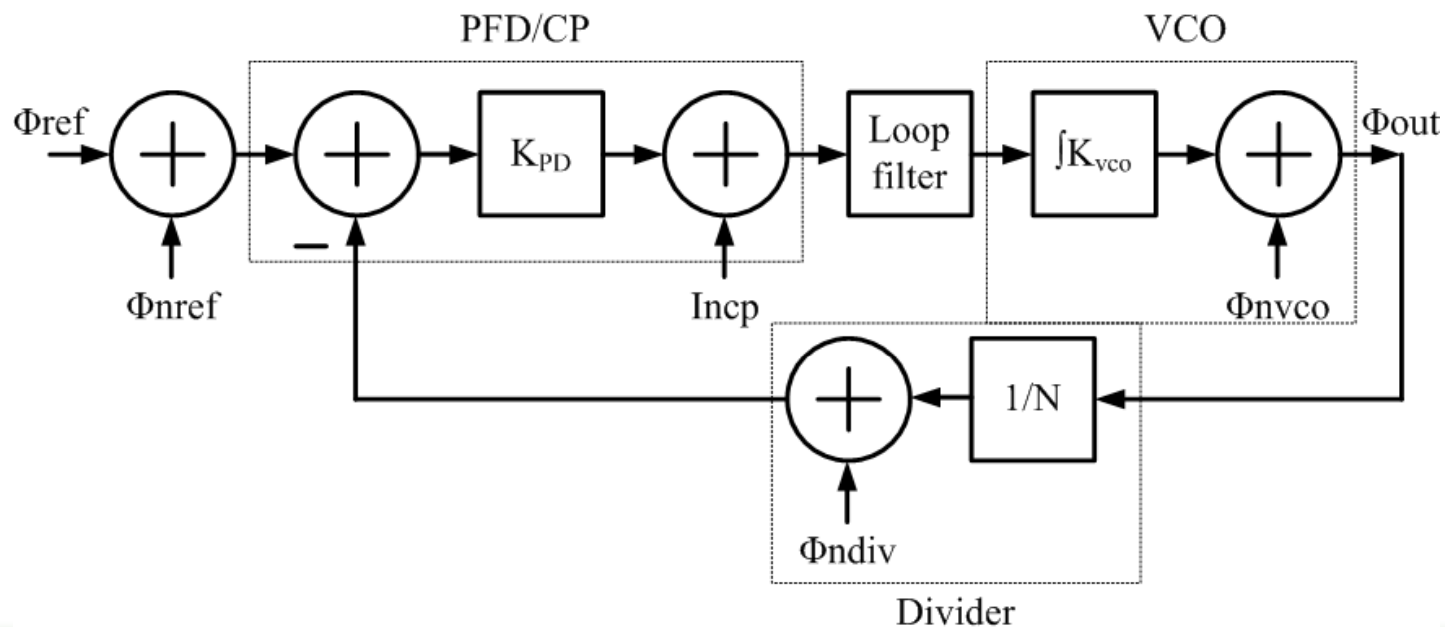
PLL Noise Modeling

- Phase domain model
 - Simple and Linear model
 - Sampling nature of PLL ignored
 - Efficient for the noise analysis when the PLL is in locked state

- Voltage domain model
 - A complete but complex model
 - No quiescent operating points and only periodic operating points
 - Describes phenomena like cycle slipping, false locking and lock capture
 - Long simulation time

Phase Domain Model

- Noise Sources
 - Reference
 - PFD/CP
 - Loop filter
 - VCO
 - Divider



CPPLL Type II 3rd Order

- Transfer function of loop filter (R-C//C_p)

$$Z(s) = \frac{(s/w_z + 1)}{s(C + C_p)(s/w_p + 1)}$$

$$w_z = \frac{1}{RC}, \quad w_p = \frac{1}{RCC_p/(C + C_p)}$$

- Transfer function of loop gain (two poles at s=0)

$$L(s) = \frac{Icp}{2\pi} Z(s) \frac{Kvco}{s} \frac{1}{N} = \frac{Icp}{2\pi} \frac{Kvco(s/w_z + 1)}{s^2 (C + C_p)(s/w_p + 1)} \frac{1}{N}$$

Noise Transfer Function of Type II CPPLL

- Noise from the reference

$$H_{ref} = \frac{\Phi_{out}}{\Phi_{nref}} = \frac{NL(s)}{1+L(s)} = H(s)$$

where $H(s)$ is the closed-loop transfer function of PLL

- Noise from PFD/CP

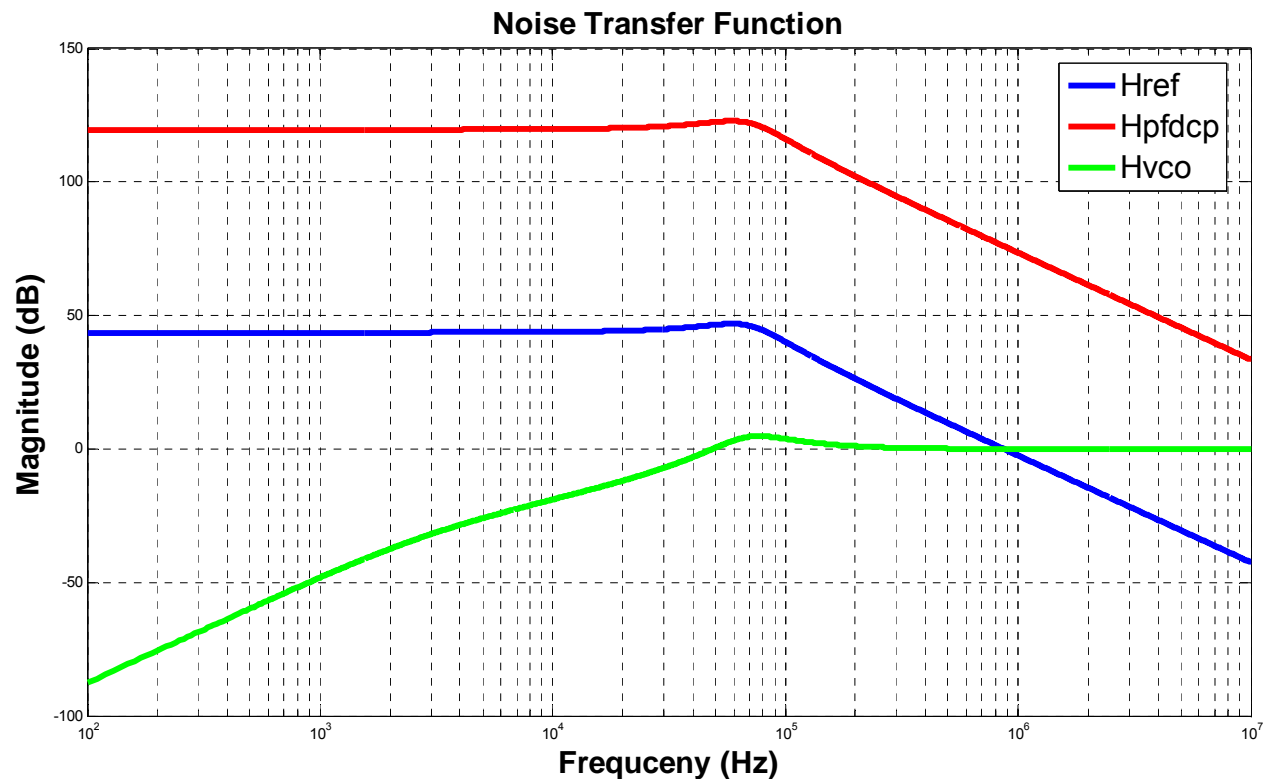
$$H_{pfhcp} = \frac{\Phi_{out}}{I_{ncp}} = \frac{Z(s)}{1+L(s)} \frac{K_{vco}}{s}$$

- Noise from VCO

$$H_{vco} = \frac{\Phi_{out}}{\Phi_{nvco}} = \frac{1}{1+L(s)}$$

Noise Transfer Function

- PLL parameters
 - $F_{out}=2.4\text{GHz}$, $F_{ref}=16\text{MHz}$
 - $I_{cp}=1\text{mA}$, $K_{vco}=2\pi*300\text{MHz}$
 - $R=300\Omega$, $C=200\text{nF}$, $C_p=10\text{nF}$



VCO Model

- Fundamental function

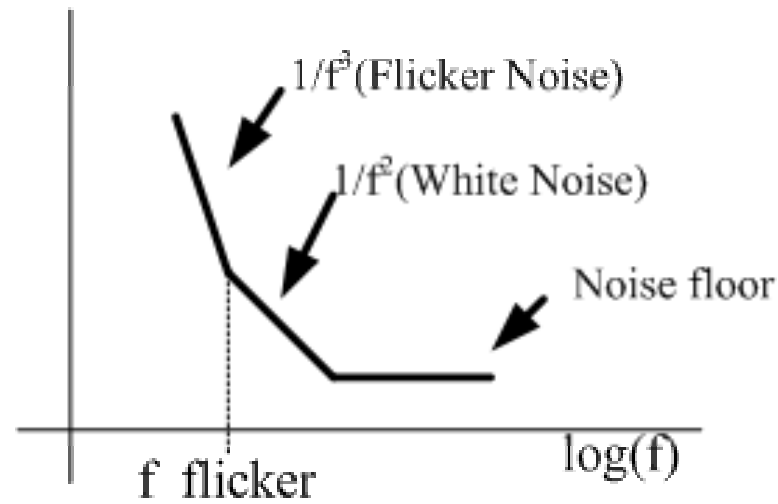
$$\Phi = \int f dt = \int (K_{vco} \times V_c) dt$$

- Noise model

- Flicker/White noise added to the frequency

$$f = K_{vco} \times V_c + v_n(t)$$

- $1/f^3$ and $1/f^2$ regions of the phase noise plot



White Noise Generating in Verilog-A

- `white_noise(PSD,"name")`
 - PSD is Power Spectral Density in units of V^2/Hz or A^2/Hz
 - “name” identifies the source while analyzing the noise simulation results

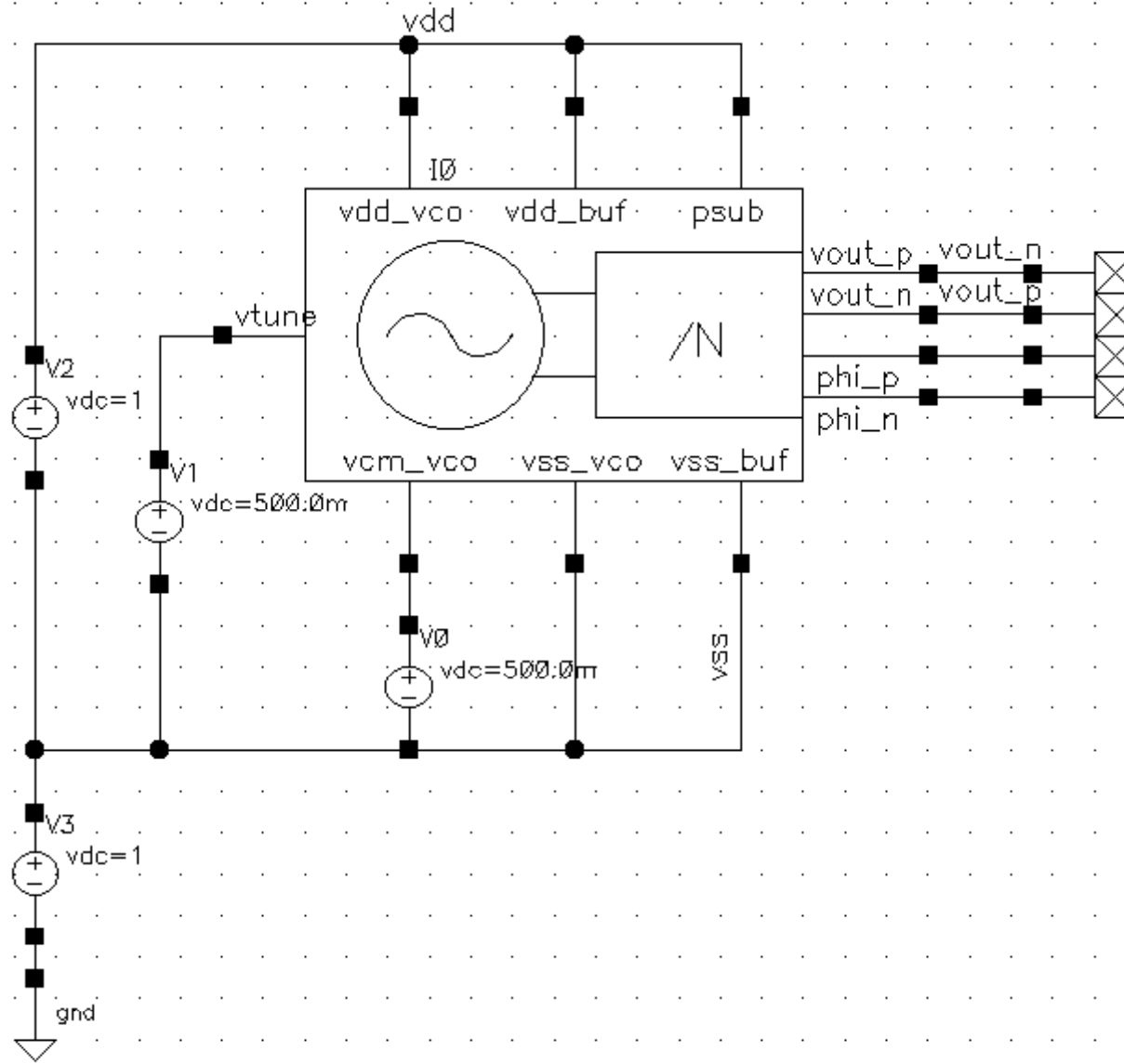
- Examples
 - `V(res) <+ white_noise(4*`P_K*$temperature*Rs,"thermal")`
 - `I(diode) <+ white_noise(2*`P_Q*Id,"shot")`

Flicker Noise Generating in Verilog-A

- flicker_noise(pow,n,"name")
 - pow is Power Spectral Density of the source at 1Hz in units of V^2/Hz or A^2/Hz
 - n is the order of the frequency i.e. $1/f^n$ is modeled
 - "name" is to identify the source while analyzing the noise simulation results

- Example of $1/f$ roll-off noise
 - f_flicker: corner frequency
 - wn: Desired white noise level
 - $V(out) <+ flicker_noise(wn*f_flicker, 1, "fn")$

VCO/Divider Test Bench



VCO PSS Simulation Setup

OK Cancel Defaults Apply

Analysis tran dc ac noise
 xf sens dcmatch stb
 pz sp envlp pss
 pac pstb pnoise pxf
 psp qpss qpac qpnoise
 qpxf qpsp

Periodic Steady State Analysis

Engine Shooting Flexible Balance

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
---	------	------	-------	--------	-------

Clear/Add Delete Update From Schematic

Beat Frequency Beat Period Auto Calculate

Output harmonics

Number of harmonics

Accuracy Defaults (empreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator Oscillator node Select
Reference node Select
Osc initial condition default linear

Sweep

Enabled

Check this for any autonomous osciillator

Fundamental Oscillation
Frequency of the oscillator

VCO Pnoise Simulation Setup

OK Cancel Defaults Apply

Analysis tran dc ac noise
 xf sens dcmatch stb
 pz sp envlp pss
 pac pstb pnoise pxf
 psp qpss qpac qpnoise
 qpxf qpasp

Periodic Noise Analysis

PSS Beat Frequency (Hz) 10G

Sweeptype absolute

Output Frequency Sweep Range (Hz)

Start-Stop Start 1k Stop 10M

Sweep Type Automatic

Add Specific Points

Sidebands
Maximum sideband 20

Output
voltage Positive Output Node /net2 Select
Negative Output Node /vss Select

Input Source none

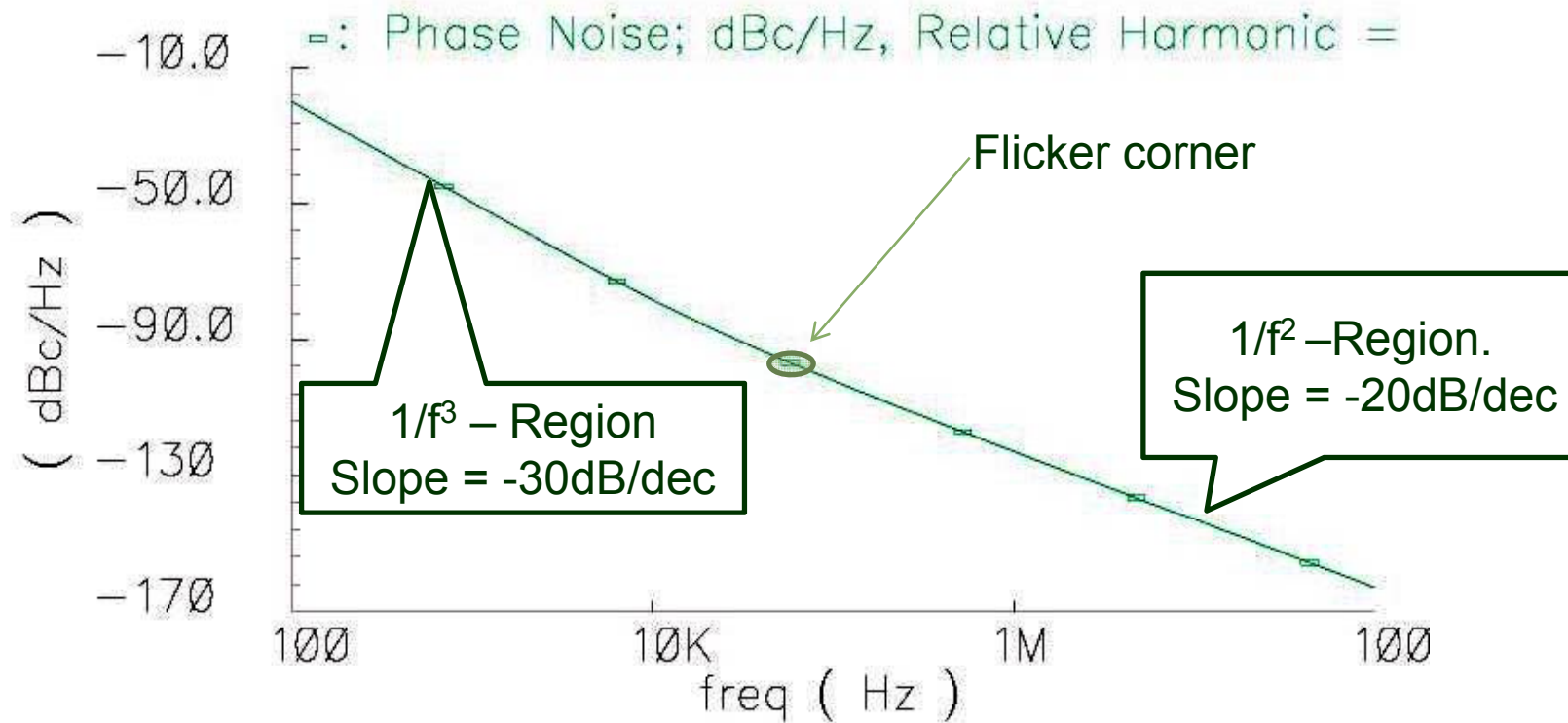
Noise Type sources
sources: single sideband (SSB) noise analysis
Noise Separation yes no
separate noise into source and gain

Enabled Options...

Noise Folding

How to choose this?

VCO Simulation Results



Alternate VCO Model for Phase Noise

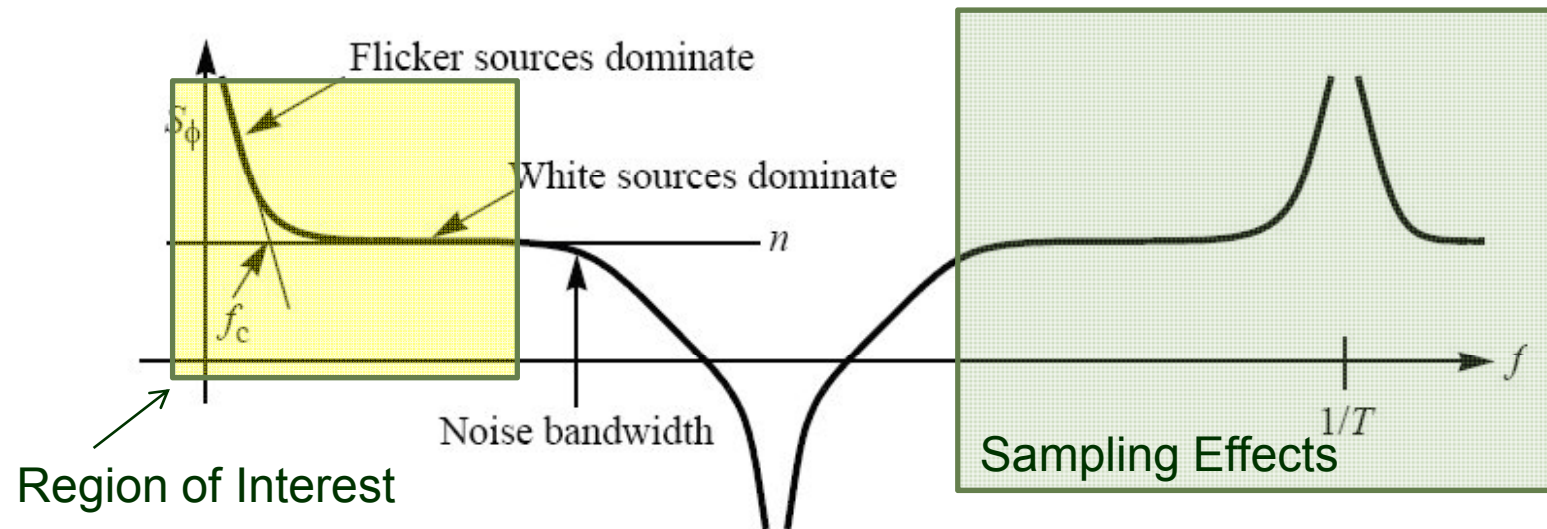
```
module vco(in, out);  
input in; output out;  
voltage in;  
phase out;  
parameter real gain = 1 from (0:inf); // transfer gain, Kvco (Hz/V)  
parameter real n = 0 from [0:inf]; // white output phase noise at 1 Hz (rad2/Hz)  
parameter real fc = 0 from [0:inf]; // flicker noise corner frequency (Hz)  
  
analog begin  
    Theta(out) <+ 2*`M_PI*gain*idt(V(in));  
    Theta(out) <+ flicker_noise(n, 2, "wpn") + flicker_noise(n*fc, 3, "fpn");  
end  
endmodule
```

White noise ($1/f^2$)

Flicker noise ($1/f^3$)

V(out) instead of Theta(out).
Discipline of "out" should be
electrical/ voltage

Divider Verilog-A Model



```

module divider(in, out);
input in; output out;
phase in, out;
parameter real ratio = 1 from (0:inf); // divide ratio
parameter real n = 0 from [0:inf]; // white output phase noise (rads2/Hz)
parameter real fc = 0 from [0:inf]; // flicker noise corner frequency (Hz)

analog begin
    Theta(out) <+ Theta(in) / ratio;
    Theta(out) <+ white_noise(n, "wpn") + flicker_noise(n*fc, 1, "fpn");
end
endmodule

```

Divider Noise Simulation Setup

OK	Cancel	Defaults	Apply	Help	
----	--------	----------	-------	------	--

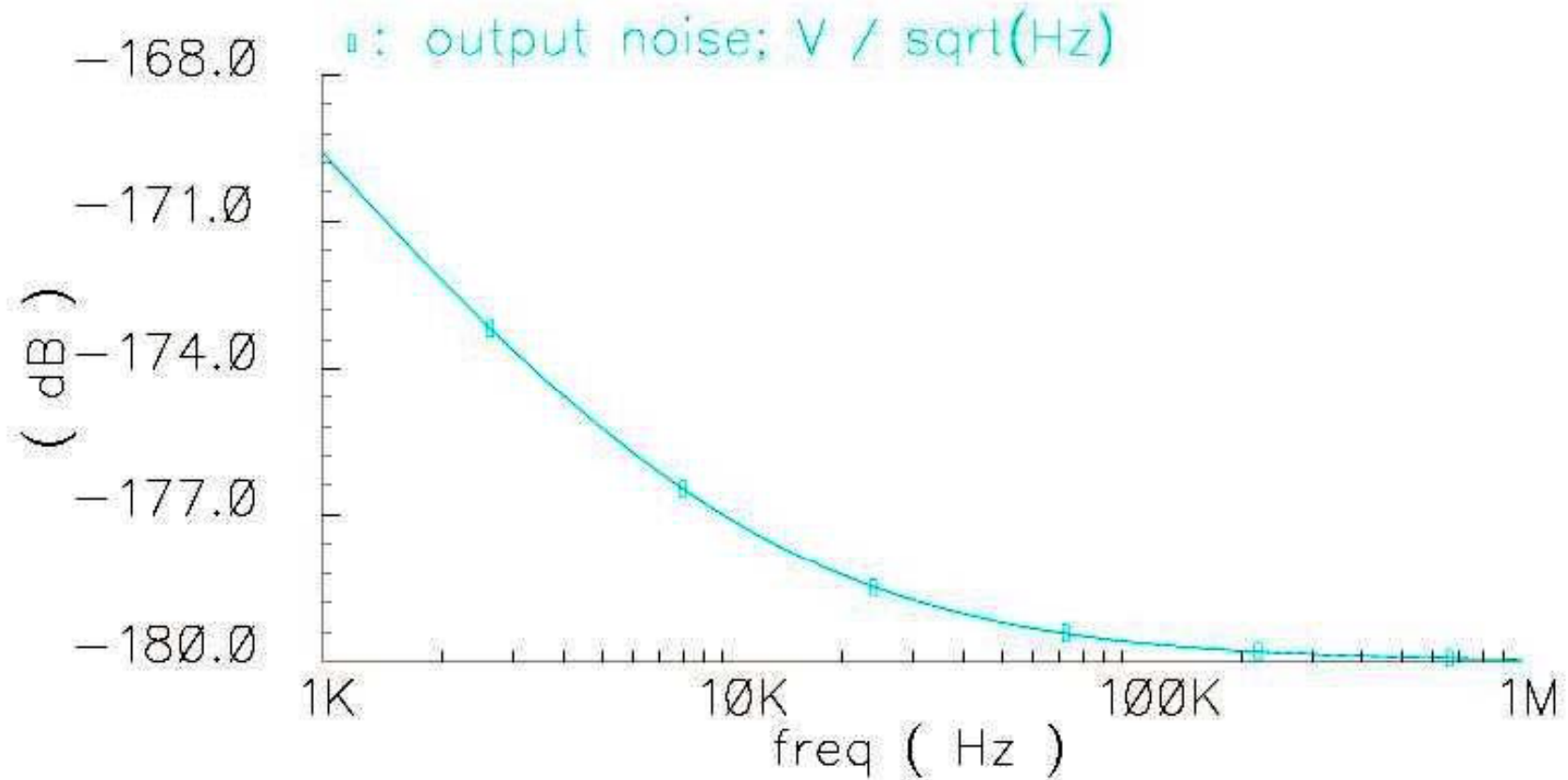
Analysis	<input type="checkbox"/> tran	<input type="checkbox"/> dc	<input type="checkbox"/> ac	<input checked="" type="checkbox"/> noise
	<input type="checkbox"/> xf	<input type="checkbox"/> sens	<input type="checkbox"/> dcmatch	<input type="checkbox"/> stb
	<input type="checkbox"/> pz	<input type="checkbox"/> sp	<input type="checkbox"/> envlp	<input type="checkbox"/> pss
	<input type="checkbox"/> pac	<input type="checkbox"/> pstb	<input type="checkbox"/> pnoise	<input type="checkbox"/> pxf
	<input type="checkbox"/> psp	<input type="checkbox"/> qpss	<input type="checkbox"/> qpac	<input type="checkbox"/> qpnoise
	<input type="checkbox"/> qpxf	<input type="checkbox"/> qpssp		

Noise Analysis	
Sweep Variable	
<input checked="" type="checkbox"/> Frequency	
<input type="checkbox"/> Design Variable	
<input type="checkbox"/> Temperature	
<input type="checkbox"/> Component Parameter	
<input type="checkbox"/> Model Parameter	

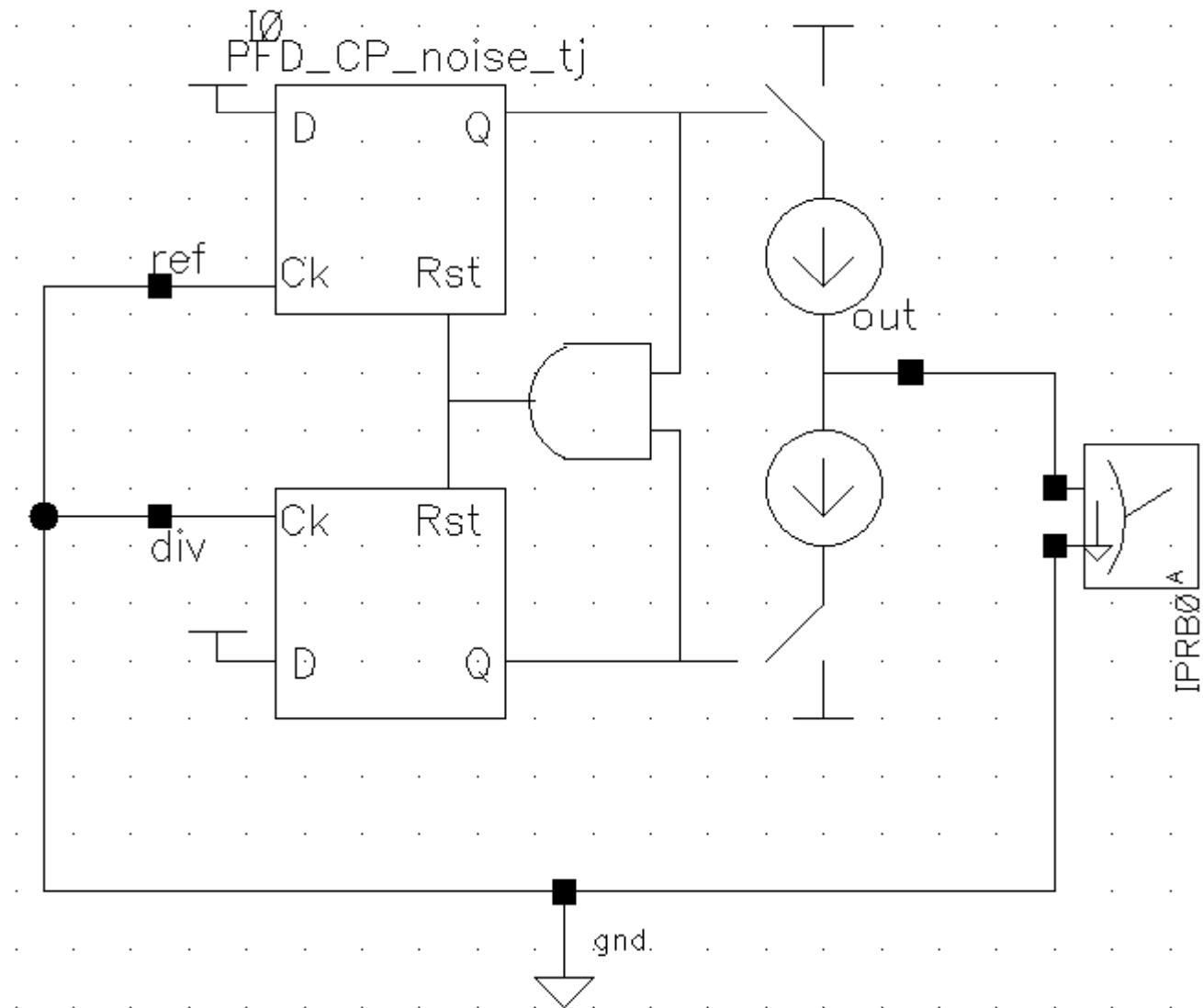
Sweep Range	
<input checked="" type="checkbox"/> Start-Stop	Start <input type="text" value="1k"/> Stop <input type="text" value="1M"/>
<input type="checkbox"/> Center-Span	
Sweep Type	
<input type="text" value="Automatic"/>	
Add Specific Points <input type="checkbox"/>	

Output Noise	
<input type="text" value="voltage"/>	Positive Output Node <input type="text" value="/out"/> <input type="button" value="Select"/>
	Negative Output Node <input type="text" value="/gnd"/> <input type="button" value="Select"/>
Input Noise	
<input type="text" value="none"/>	
Enabled <input checked="" type="checkbox"/>	<input type="button" value="Options..."/>

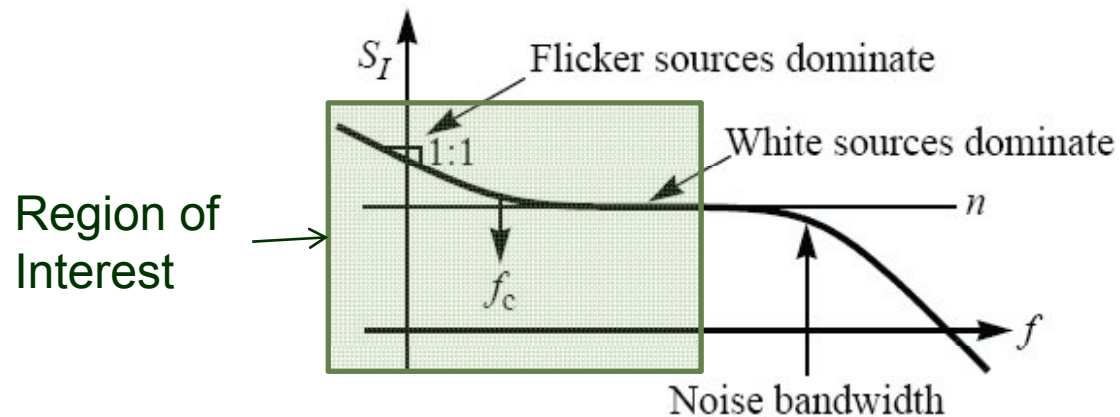
Divider Simulation Result



PFD/CP Test Bench

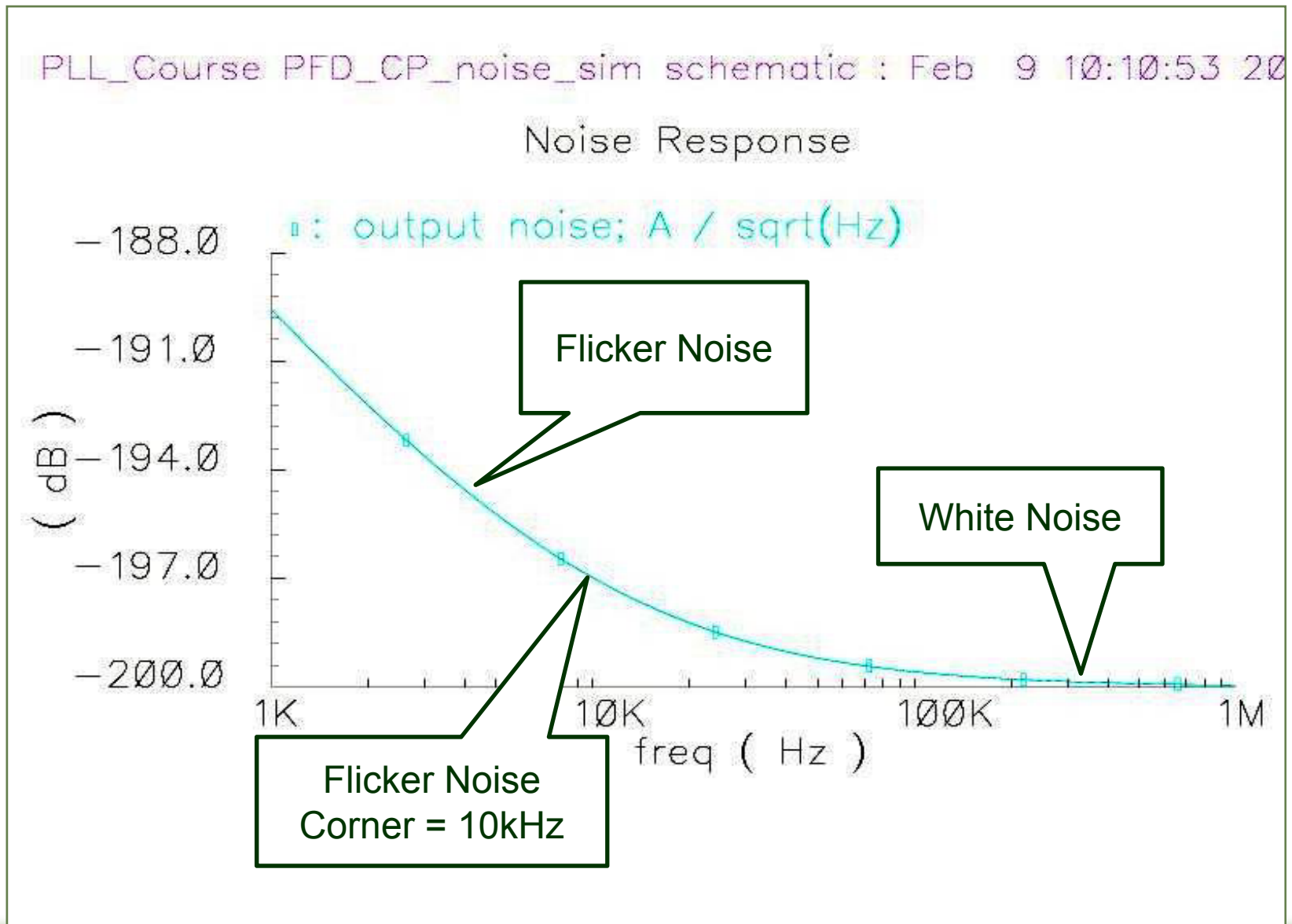


PFD/CP Verilog-A Model



```
module phaseDetector(pin, nin, out);  
input pin, nin; output out;  
phase pin, nin;  
electrical out;  
parameter real gain = 1 from (0:inf); // transfer gain (A/cycle)  
parameter real n = 0 from [0:inf]; // white output current noise (A2/Hz)  
parameter real fc = 0 from [0:inf]; // flicker noise corner frequency (Hz)  
analog begin  
    I(out) <+ -gain * Theta(pin,nin) / (2*`M_PI);  
    I(out) <+ white_noise(n, "wpn") + flicker_noise(n*fc, 1, "fpn");  
end  
endmodule
```

PFD Phase Noise Simulation Result



PLL PSS Analysis Setup

OK Cancel Defaults Apply Help

Analysis tran dc ac noise
 xf sens dcmatch stb
 pz sp envlp pss
 pac pstb pnoise pxf
 psp qpss qpac qpnoise
 qpxf qpsp

Periodic Steady State Analysis

Engine Shooting Flexible Balance

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
---	------	------	-------	--------	-------

Clear/Add Delete Update From Schematic

Beat Frequency 16M Auto Calculate
 Beat Period

Output harmonics
Number of harmonics 50

Accuracy Defaults (errpreset)
 conservative moderate liberal

Additional Time for Stabilization (tstab) 100
Save Initial Transient Results (saveinit) no yes

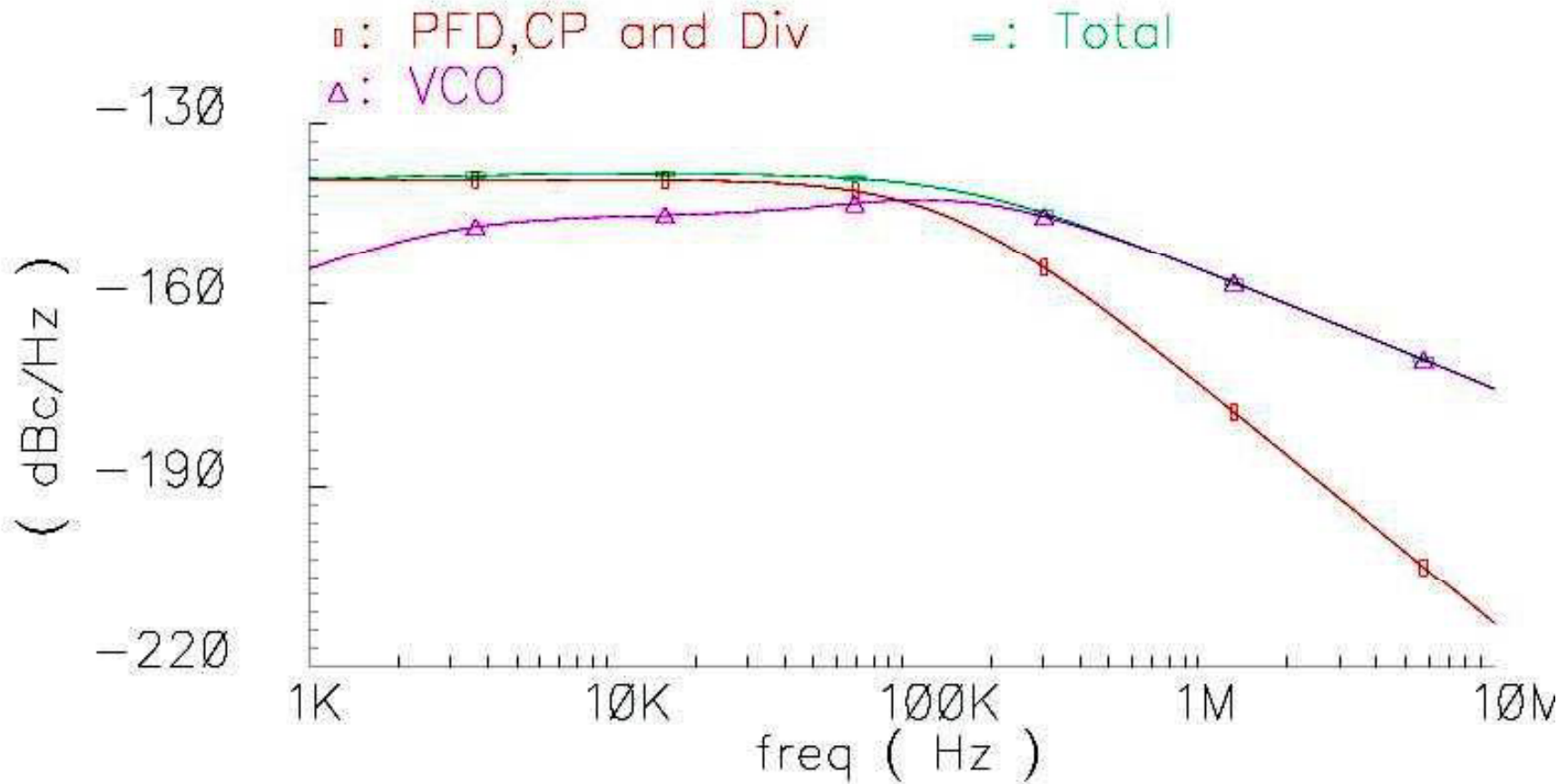
Oscillator

Sweep

Enabled Options...

- PLL is a driven system so the oscillator option should be turned OFF

PLL Simulation Results



Summary

- Phase noise and its effects on RF transceiver systems were introduced
- Phase domain and voltage domain models were compared
- Verilog-A models for the different building blocks of a PLL were discussed
- Simulation setup for running phase noise simulation was demonstrated

Acknowledgements

- We would like to thank Prof. Peter Kinget for the helpful discussions on the content of this presentation and Shih-An for teaching us techniques to improve the simulation speed
- The Verilog-A code used for this presentation and those we have ever written are derived from the ones in www.designers-guide.org