Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
PLL DesignGuide

© Agilent Technologies, Inc. 2000-2011
5301 Stevens Creek Blvd., Santa Clara, CA 95052 USA
No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Mentor products and processes are registered trademarks of Mentor Graphics Corporation. * Calibre is a trademark of Mentor Graphics Corporation in the US and other countries. "Microsoft®, Windows®, MS Windows®, Windows NT®, Windows 2000® and Windows Internet Explorer® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Oracle and Java and registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC. FLEXlm is a trademark of Globetrotter Software, Incorporated. Layout Boolean Engine by Klaas Holwerda, v1.7 http://www.xs4all.nl/~kholwerd/bool.html . FreeType Project, Copyright (c) 1996-1999 by David Turner, Robert Wilhelm, and Werner Lemberg. QuestAgent search engine (c) 2000-2002, JObjects. Motif is a trademark of the Open Software Foundation. Netscape is a trademark of Netscape Communications Corporation. Netscape Portable Runtime (NSPR), Copyright (c) 1998-2003 The Mozilla Organization. A copy of the Mozilla Public License is at http://www.mozilla.org/MPL/ . FFTW, The Fastest Fourier Transform in the West, Copyright (c) 1997-1999 Massachusetts Institute of Technology. All rights reserved.

The following third-party libraries are used by the NlogN Momentum solver:

"This program includes Metis 4.0, Copyright © 1998, Regents of the University of Minnesota", http://www.cs.umn.edu/~metis , METIS was written by George Karypis (karypis@cs.umn.edu).

Intel@ Math Kernel Library, http://www.intel.com/software/products/mkl

SuperLU_MT version 2.0 - Copyright © 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy). All rights reserved. SuperLU Disclaimer: THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: http://www.qtsoftware.com/downloads Patches Applied to Qt can be found in the installation at: $HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this documentation and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at http://systemc.org/ . This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

PLL QuickStart Guide

This PLL QuickStart Guide is intended to help you get started using the Phase-Locked Loop Design Guide effectively. For detailed reference information, refer to *PLL DesignGuide Reference* (dgpll).

> **Note**
> This document is written describing and showing access through the selection dialog box method. If you are running the program through the cascading menu preference, the appearance and interface will be slightly different.

The PLL DesignGuide has many simulation setups and data displays that are very useful for designing a phase-locked loop. The simulation set-ups are categorized by the PLL configuration, simulation technique, and type of phase detector and low-pass filter. The simulation set-ups are for analysis.

> **Note**
> This DesignGuide is not a complete solution for all phase-locked loop techniques, but covers the most common approaches. Subsequent releases of this DesignGuide will include an expanded range of features.

Using DesignGuides

All DesignGuides can be accessed in the Schematic window through either cascading menus or dialog boxes. You can configure your preferred method in the Advanced Design System Main window.

Select the **DesignGuide** menu from Advanced Design System Main window.

The commands in this menu are as follows:

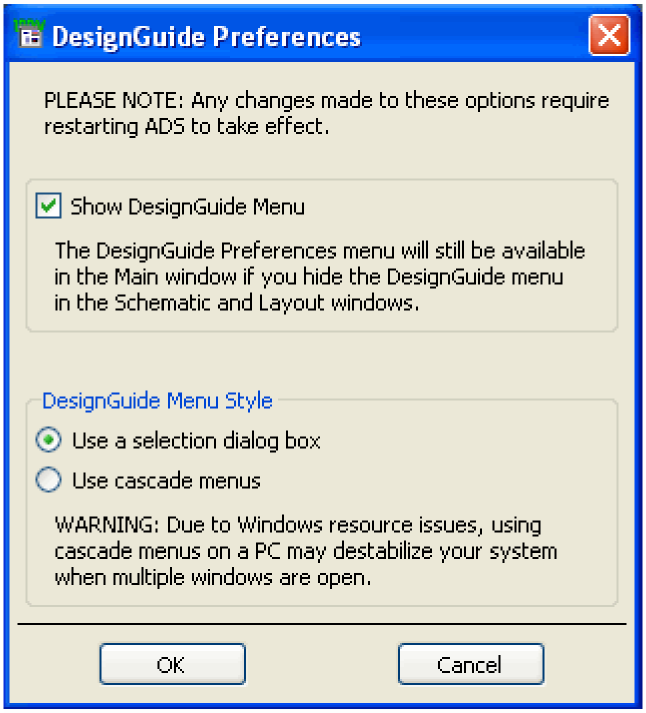
- **DesignGuide Developer Studio** > **Developer Studio Documentation** is only available on this menu if you have installed the DesignGuide Developer Studio. It brings up the DesignGuide Developer Studio documentation. Another way to access the Developer Studio documentation is by selecting **Help > Topics and Index > DesignGuides > DesignGuide Developer Studio** (from any ADS program window).

- **DesignGuide Developer Studio** > **Start DesignGuide Studio** is only available on this menu if you have installed the DesignGuide Developer Studio. It launches the initial Developer Studio dialog box.

- **Add DesignGuide** brings up a directory browser in which you can add a DesignGuide to your installation. This is primarily intended for use with DesignGuides that are custom-built through the Developer Studio.

- **List/Remove DesignGuide** brings up a list of your installed DesignGuides. Select any that you would like to uninstall and choose the **Remove** button.

- **Preferences** brings up a dialog box that allows you to:
  - Disable the DesignGuide menu commands (all except Preferences) in the Main window by unchecking this box. In the Schematic and Layout windows, the complete **DesignGuide** menu and all of its commands will be removed if this box is unchecked.
  - Select your preferred interface method (cascading menus vs. dialog boxes).



DesignGuide Preferences

PLEASE NOTE: Any changes made to these options require restarting ADS to take effect.

☑ Show DesignGuide Menu

The DesignGuide Preferences menu will still be available in the Main window if you hide the DesignGuide menu in the Schematic and Layout windows.

DesignGuide Menu Style

● Use a selection dialog box

○ Use cascade menus

WARNING: Due to Windows resource issues, using cascade menus on a PC may destabilize your system when multiple windows are open.

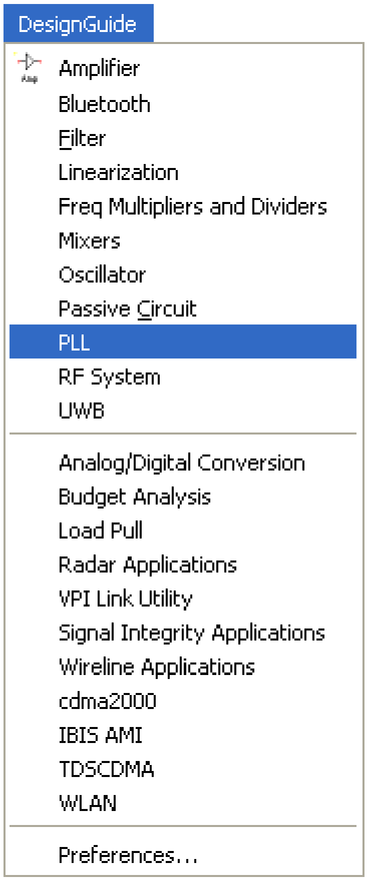
Close and restart the program for your preference changes to take effect.

Note
On PC systems, Windows resource issues might limit the use of cascading menus. When multiple windows are open, your system could become destabilized. Thus the dialog box menu style might be best for these situations.

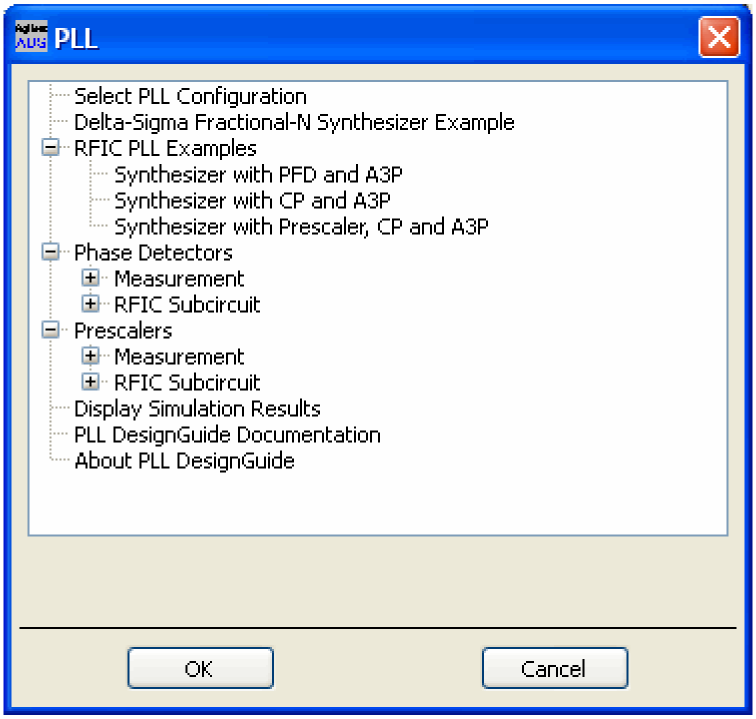
Basic Procedures

The features and content of the PLL DesignGuide are accessible from the DesignGuide menu found in the ADS Schematic window.
To access the documentation for the DesignGuide, select either of the following:

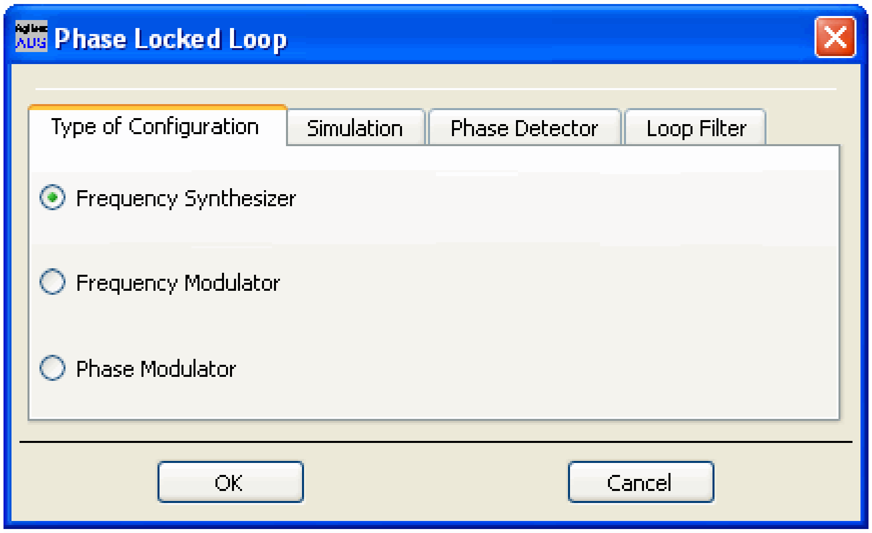
- **DesignGuide** > **PLL** > **PLL DesignGuide Documentation** (from ADS Schematic window)
- **Help** > **Topics and Index** > **DesignGuides** > **PLL** (from any ADS program window)



You have the option of selecting a PLL Configuration or choosing to examine one of the various RFIC PLL examples. The RFIC examples are subsets of the various PLL configurations, whereby device level components replace the phase detector and prescalar model components.


PLL
Select PLL Configuration
Delta-Sigma Fractional-N Synthesizer Example
RFIC PLL Examples
Synthesizer with PFD and A3P
Synthesizer with CP and A3P
Synthesizer with Prescaler, CP and A3P
Phase Detectors
Measurement
RFIC Subcircuit
Prescalers
Measurement
RFIC Subcircuit
Display Simulation Results
PLL DesignGuide Documentation
About PLL DesignGuide
OK
Cancel

You have the **Select PLL Configuration** option to select a PLL Configuration, as shown below.

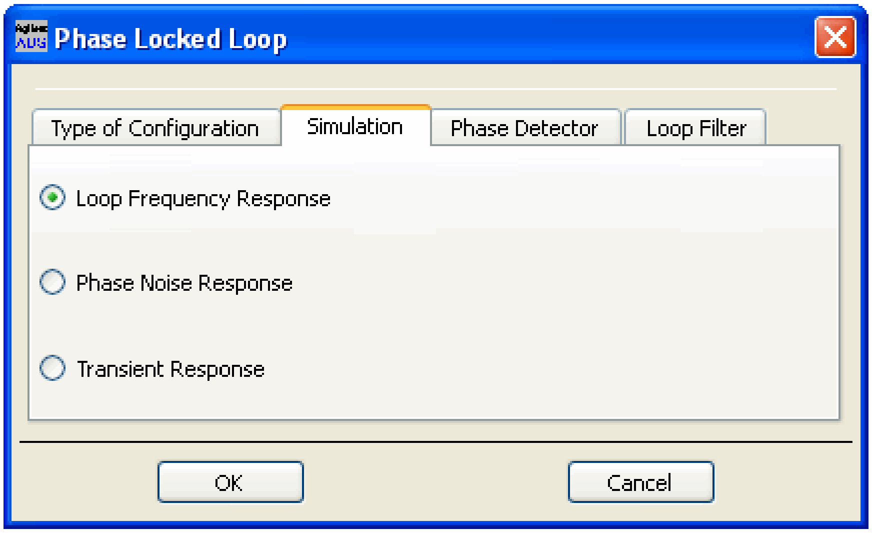


● Frequency Synthesizer

○ Frequency Modulator

○ Phase Modulator

You can select one of the available PLL configurations as shown below.
Having identified the type of PLL structure, you then select one of the three simulations available from the **Simulation** tab, as shown here.
The simulations include *Loop frequency Response*, *Phase Noise Response*, and *Transient response*.

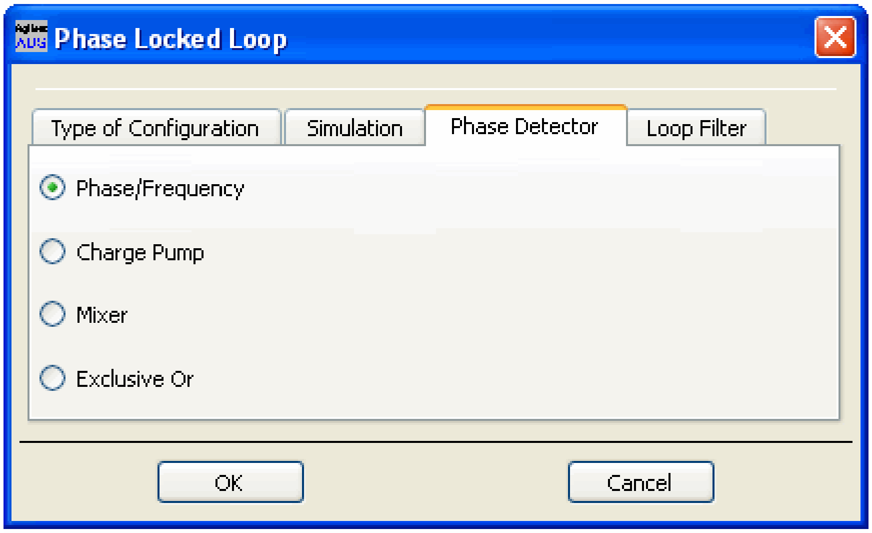


● Loop Frequency Response

○ Phase Noise Response

○ Transient Response

You then need to identify the phase detector and low-pass filter used in your design. Some combination are unavailable at this time but are expected to be available in future upgrades.
The selection box for phase detectors is shown here.



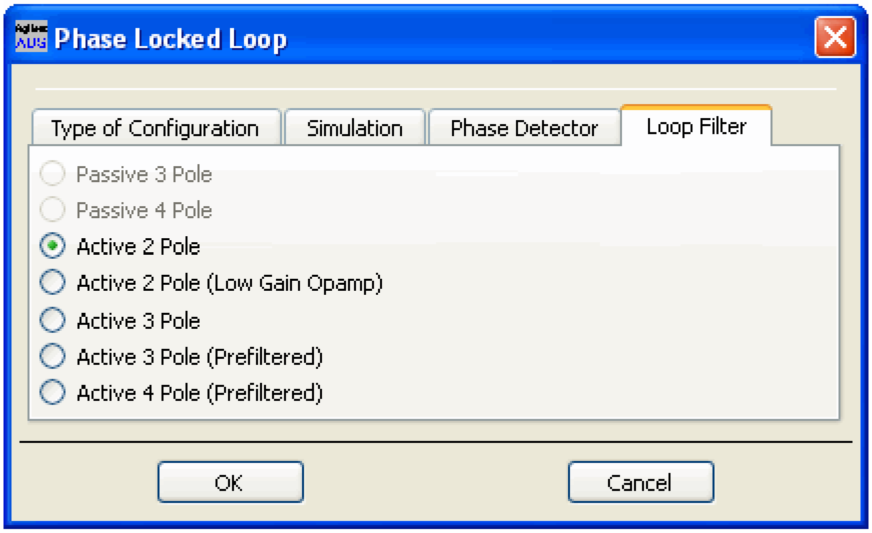
● Phase/Frequency

○ Charge Pump

○ Mixer

○ Exclusive Or

Below is the selection box for loop filters. The grayed-out selections are not available at this time. Right-click one of the available selections. For a detailed description of the loop filter selections, refer to the *PLL DesignGuide Reference* (dgpll).



○ Passive 3 Pole
○ Passive 4 Pole
● Active 2 Pole
○ Active 2 Pole (Low Gain Opamp)
○ Active 3 Pole
○ Active 3 Pole (Prefiltered)
○ Active 4 Pole (Prefiltered)

Selecting Appropriate Configurations

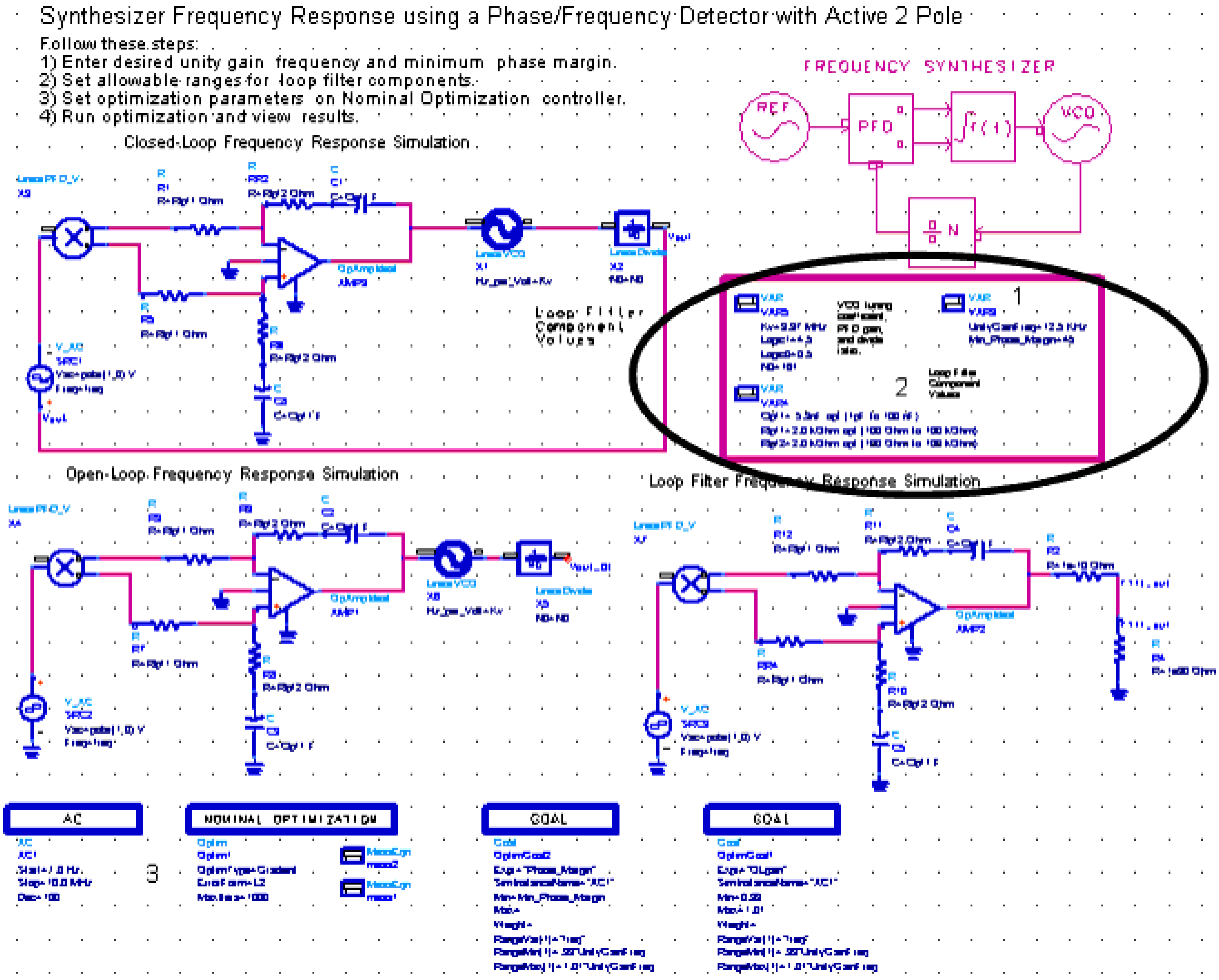
The Phase-Locked Loop DesignGuide is broken up into different sub-categories, as shown in the previous section. The specifications that you select depend on your desired simulation and the type of PLL structure that your system can utilize.

If, for example, you are designing a synthesizer, you can start with the loop frequency response configurations shown in the section Phase Margin and Unity Gain Bandwidth. The output parameters will be used for evaluating the phase noise and transient responses.

Most information on the data display for this design simulation and others is in a format that engineers can easily understand. The visibility of equation syntax is minimized. Information about items on a data display that you will want to modify is enclosed in red boxes.

Phase Margin and Unity Gain Bandwidth

The optimization procedure based on achieving a desired Phase margin and Unity Gain Bandwidth is shown here. Enter the desired values, as well as your VCO tuning parameter, the divide ratio, and the Phase Detector characteristics. (Enter this data in the area of the schematic encircled in the following illustration.)

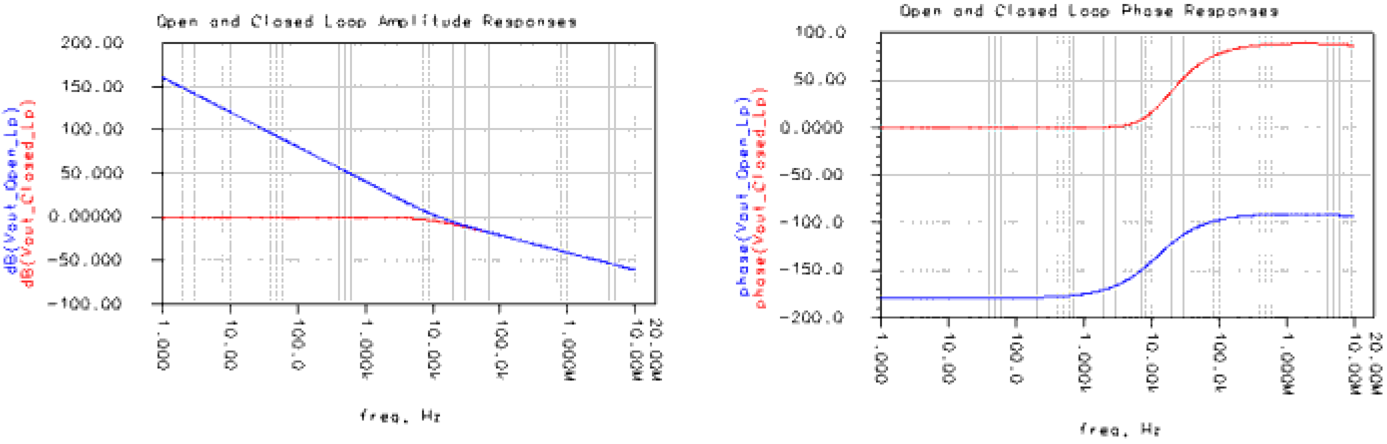


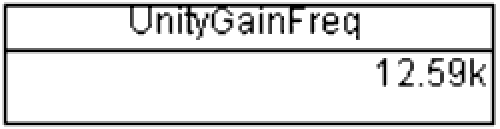
Synthesizer Frequency Response using a Phase/Frequency Detector with Active 2 Pole

Follow these steps:
1) Enter desired unity gain frequency and minimum phase margin.
2) Set allowable ranges for loop filter components.
3) Set optimization parameters on Nominal Optimization controller.
4) Run optimization and view results.

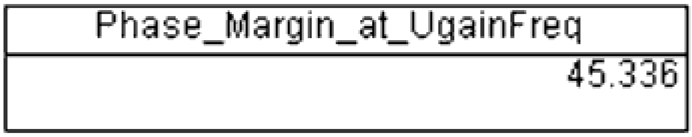
In the data display results shown here, the resultant Phase Margin and Unity Gain Frequency are displayed, along with the optimized loop filter parameters. If the objectives have not been met, you should adjust the loop filter parameters to alter the initial conditions of the optimization and re-run the simulation.

Synthesizer Frequency Response with a Phase/Frequency Detector and Active 2 Pole





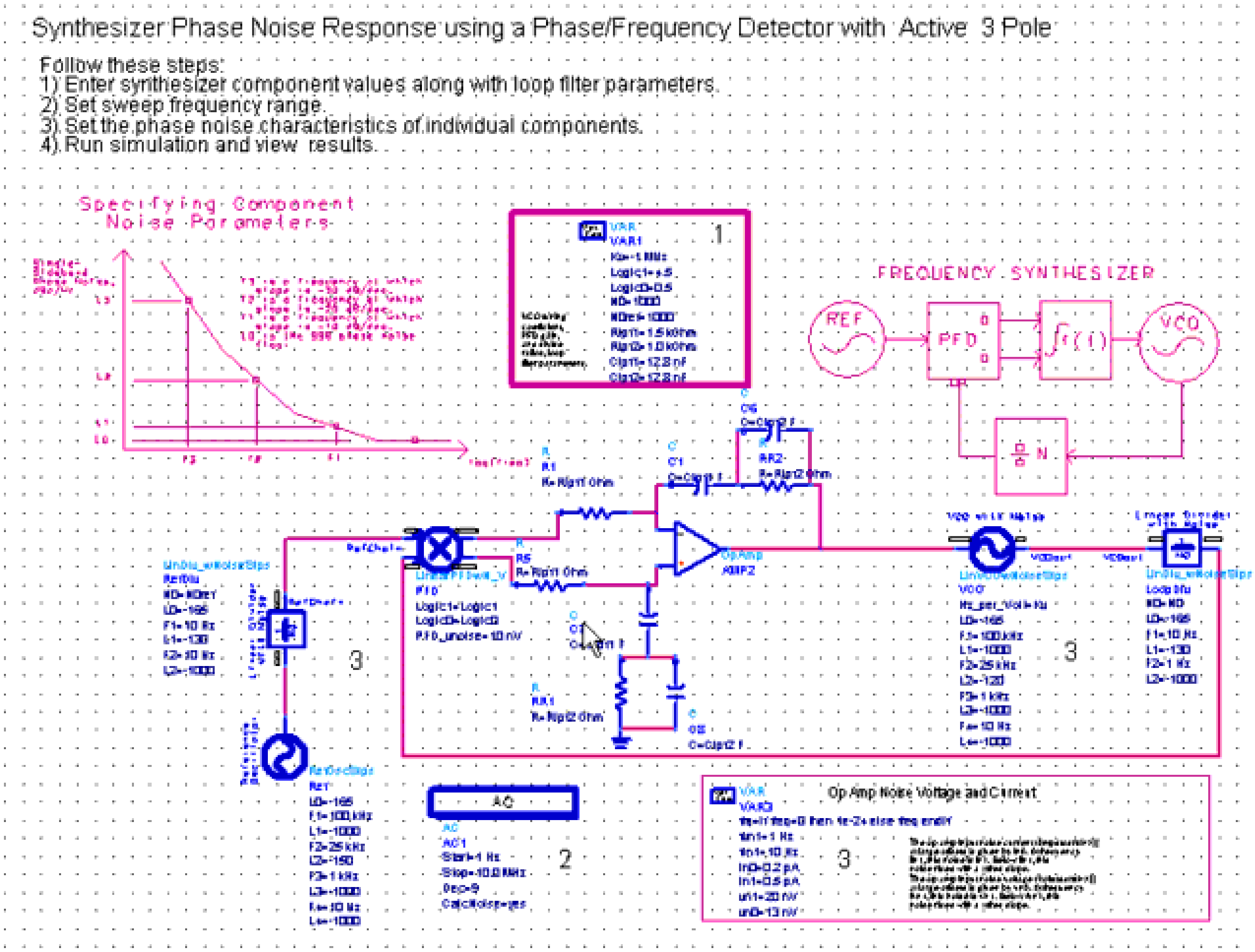
UnityGainFreq
12.59k



Phase_Margin_at_UgainFreq
45.336

Phase Noise Response

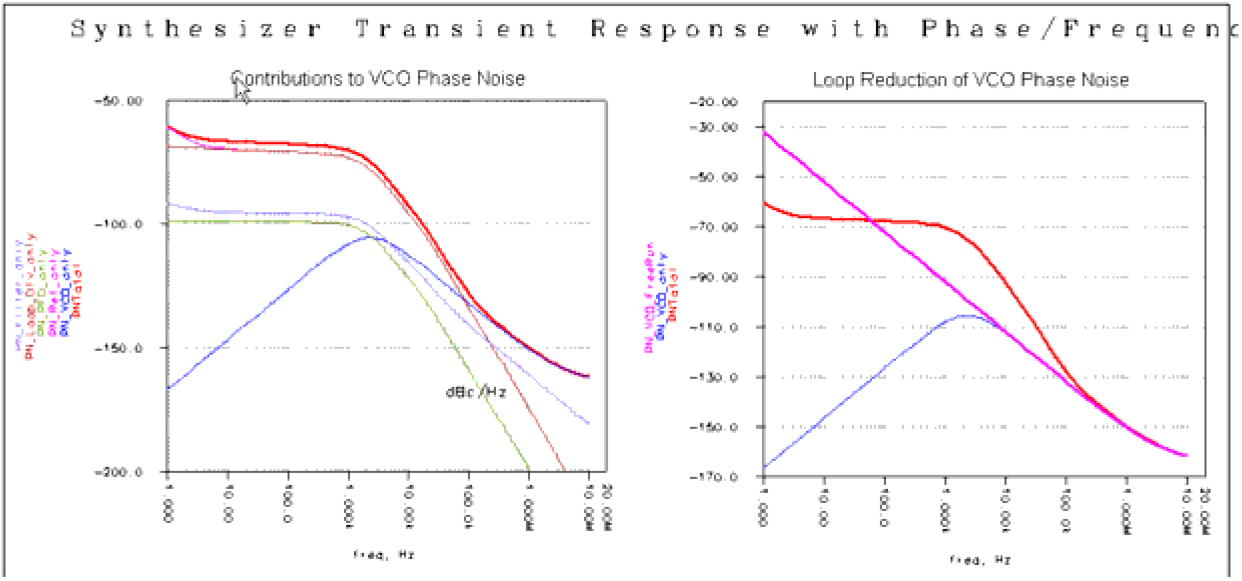
The parameters derived from the Loop Frequency Response schematic should be entered into the Phase Noise Response schematic.



Synthesizer Phase Noise Response using a Phase/Frequency Detector with Active 3 Pole

Follow these steps:
1) Enter synthesizer component values along with loop filter parameters.
2) Set sweep frequency range.
3) Set the phase noise characteristics of individual components.
4) Run simulation and view results.

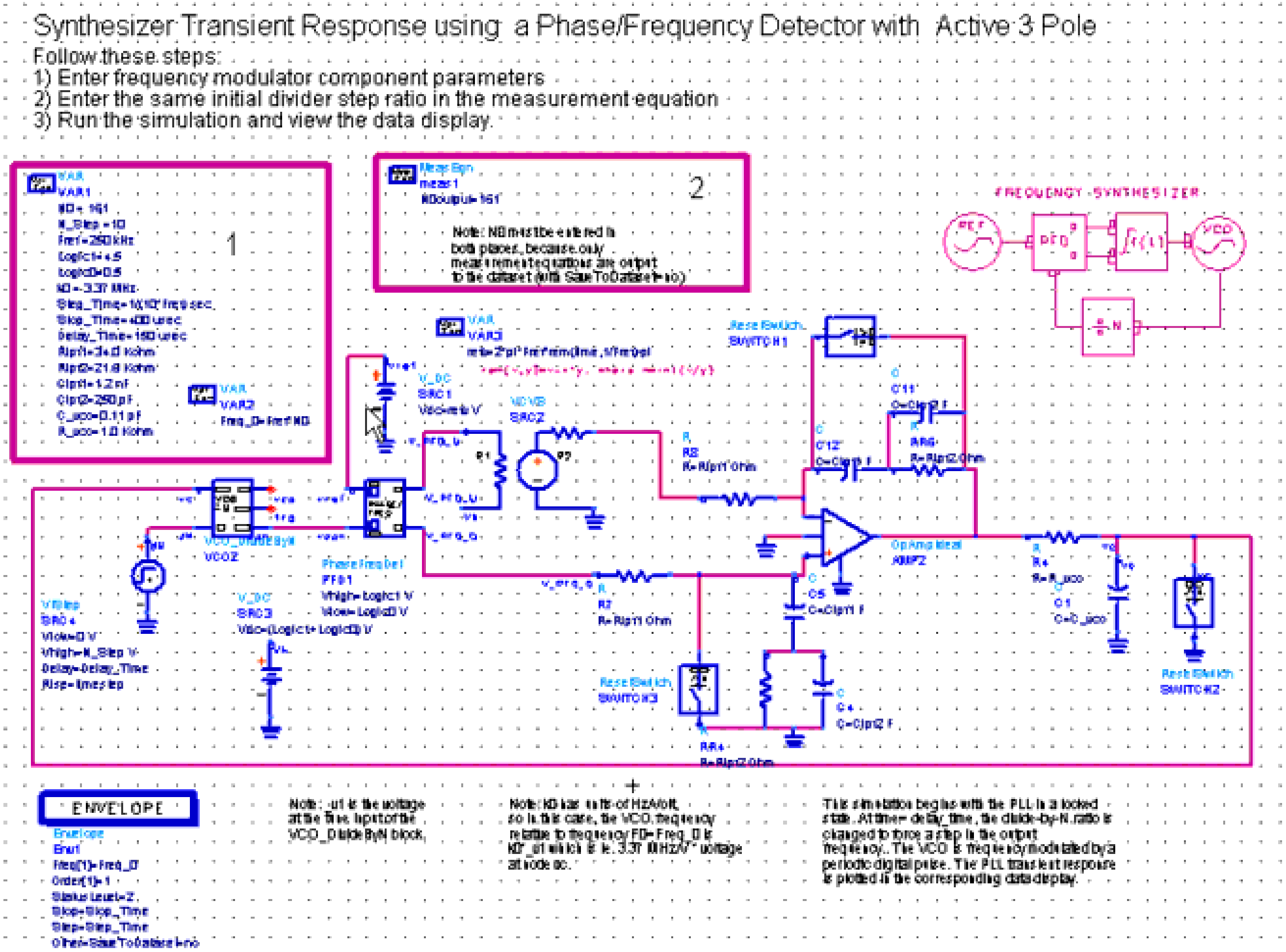
The phase noise characteristics of each component should be set on each subcircuit block. The F and L parameters that describe the noise versus frequency characteristics are depicted in the schematic.
The data display corresponding to the Phase Noise schematic is shown here.



The graph on the left displays the individual noise source's contributions and the graph on the right shows the overall noise performance of the PLL.

Transient Time Response

Shown here is the schematic for evaluating the transient time response of a synthesizer.



Synthesizer Transient Response using a Phase/Frequency Detector with Active 3 Pole

Follow these steps:

1) Enter frequency modulator component parameters
2) Enter the same initial divider step ratio in the measurement equation
3) Run the simulation and view the data display.

VAR
VAR1
N0= 161
N_Step = 10
Fref=250 kHz
Logic1=4.5
Logic0=0.5
K0=-3.37 MHz
Step_Time=1/(10*Fref) sec
Stop_Time=400 usec
Delay_Time=150 usec
Rlpf1=34.0 Kohm
Rlpf2=21.8 Kohm
Clpf1=1.2 nF
Clpf2=250 pF
C_vco=0.11 pF
R_vco=1.0 Kohm

VAR
VAR2
Freq_0=Fref*N0

MeasEqn
meas1
N0output=161

Note: N0 must be entered in both places, because only measurement equations are output to the dataset (with SaveToDataset=no)

VAR
VAR3
init=2*pi*Fref*rem(time,1/Fref)/pi

Note: -u1 is the voltage at the fine input of the VCO_DivideByN block.

Note: K0 has units of Hz/volt, so in this case, the VCO frequency relative to frequency F0=Freq_0 is K0*_u1 which is i.e. 3.37 MHz/V * voltage at node uc.

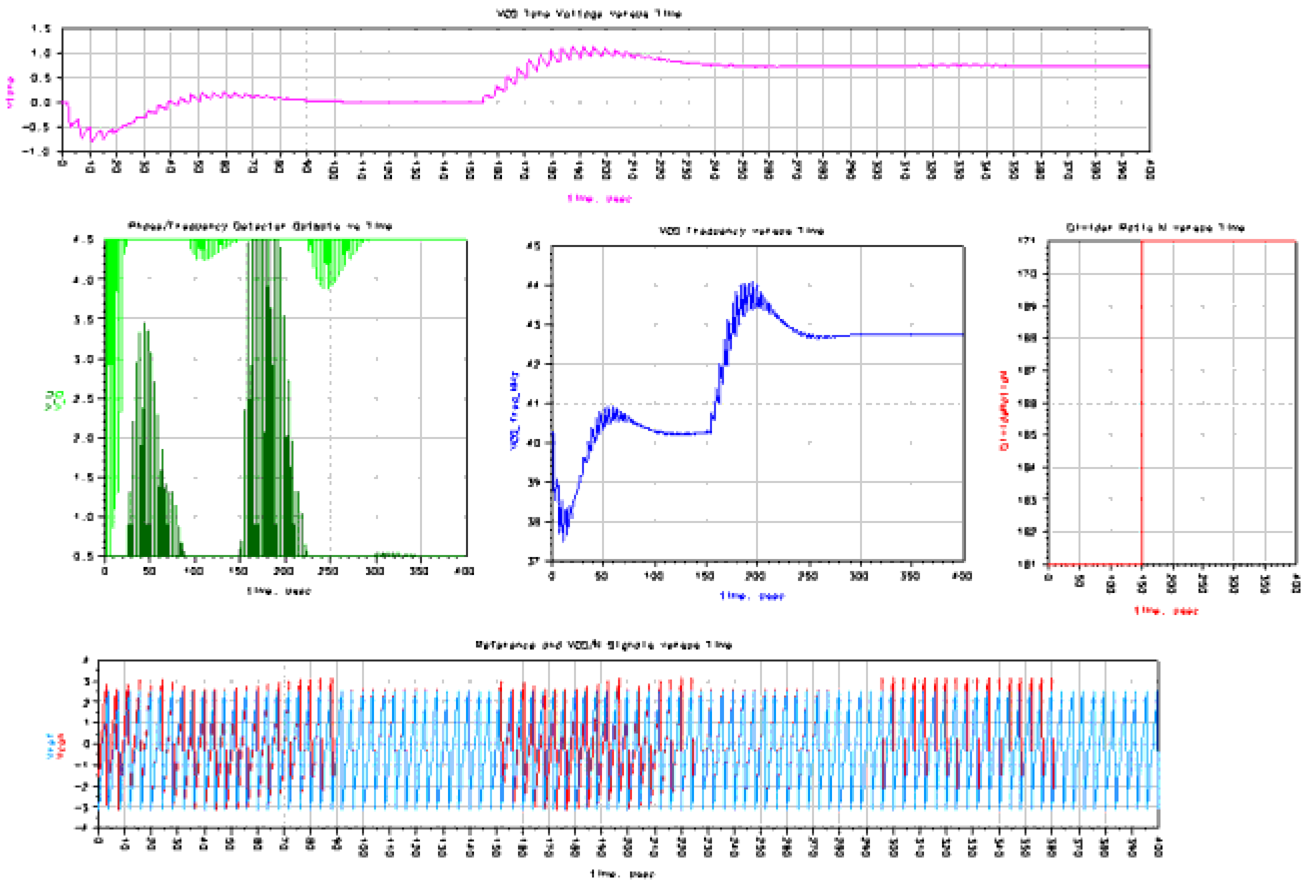
This simulation begins with the PLL in a locked state. At time= delay_time, the divide-by-N ratio is changed to force a step in the output frequency. The VCO is frequency modulated by a periodic digital pulse. The PLL transient response is plotted in the corresponding data display.

ENVELOPE
Envelope
Env1
Freq[1]=Freq_0
Order[1]=1
Status Level=2
Stop=Stop_Time
Step=Step_Time
Other=SaveToDataset=no

The loop filter parameters derived from the Loop Frequency response need to be entered into this schematic, along with the VCO and Phase Detector constants. The transient response requires additional parameters such as the Reference Frequency and the stop and delay time, as well as the Divider Ratio step change.

The Transient response data display has several figures that will describe the PLL performance as a function of time. From this display, you can evaluate the settling time and use the results to debug the phase-locked loop.

Synthesizer Transient Response with Phase/Frequency Detector and Active 3 Pole



PLL DesignGuide Reference

The sections that follow provide a basic reference on the use of the PLL DesignGuide.

Using the PLL DesignGuide

The Phase-Locked Loop (PLL) DesignGuide is integrated into Agilent EEsof's Advanced Design System environment, working as an interactive handbook for the creation of useful designs. The Guide contains many templates to be used in the ADS software environment. These templates can assist the PLL developer in designing a phase-locked loop to meet performance specifications.

You can use the optimization templates to define the loop performance, then proceed to evaluate the phase noise response and transient response. The DesignGuide provides a complete tool kit to interactively explore dynamic PLL systems at the top level as part of an integrated design process.

In addition to the requirements of the ADS EEsof software, the PLL DesignGuide requires approximately 10 MB of additional storage space.

> **Note**
> This documentation assumes that you are familiar with all of the basic ADS program operations. For additional information, refer to *Schematic Capture and Layout* (usrguide).

The primary features of the PLL DesignGuide are:

- Complete PLL synthesis capability
- Frequency synthesizer design
- Phase modulator design
- Frequency modulator design
- Phase demodulator design
- Frequency demodulator design
- RFIC PLL examples
- Open and closed loop frequency response
- Phase noise simulation
- Time domain transient simulation
- Four distinct phase detectors
- Seven loop filter configurations
- Opamp, VCO, phase detector, reference characterization
- Easy modification to user-defined configurations

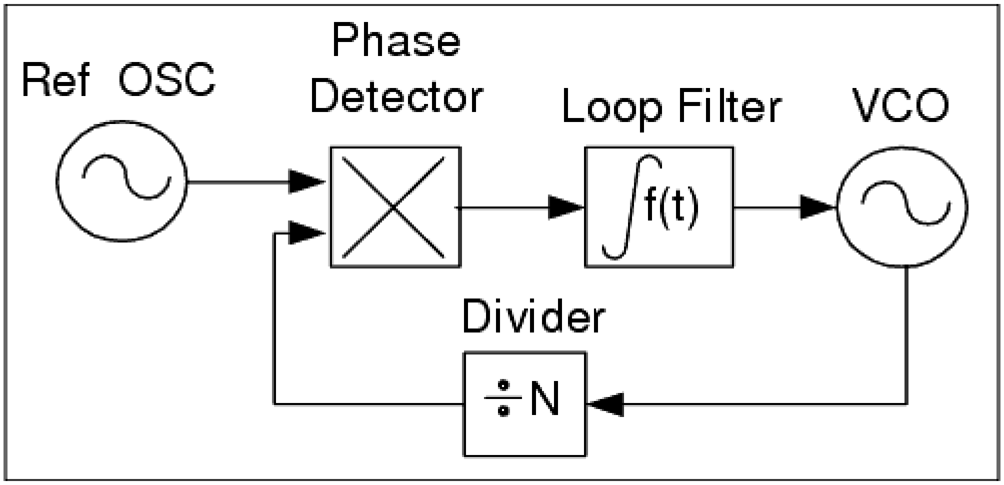
PLL Configurations

Following are diagrams and basic descriptions of the PLL configurations included in this

DesignGuide. To access these tools, select *DesignGuide* > *PLL DesignGuide* > *Select PLL Configuration* from the ADS Schematic window, and make appropriate selections in the tabs of the dialog box.

Frequency Synthesizer (SYN)

A block diagram of the basic phase-locked synthesizer is shown in Frequency Synthesizer.

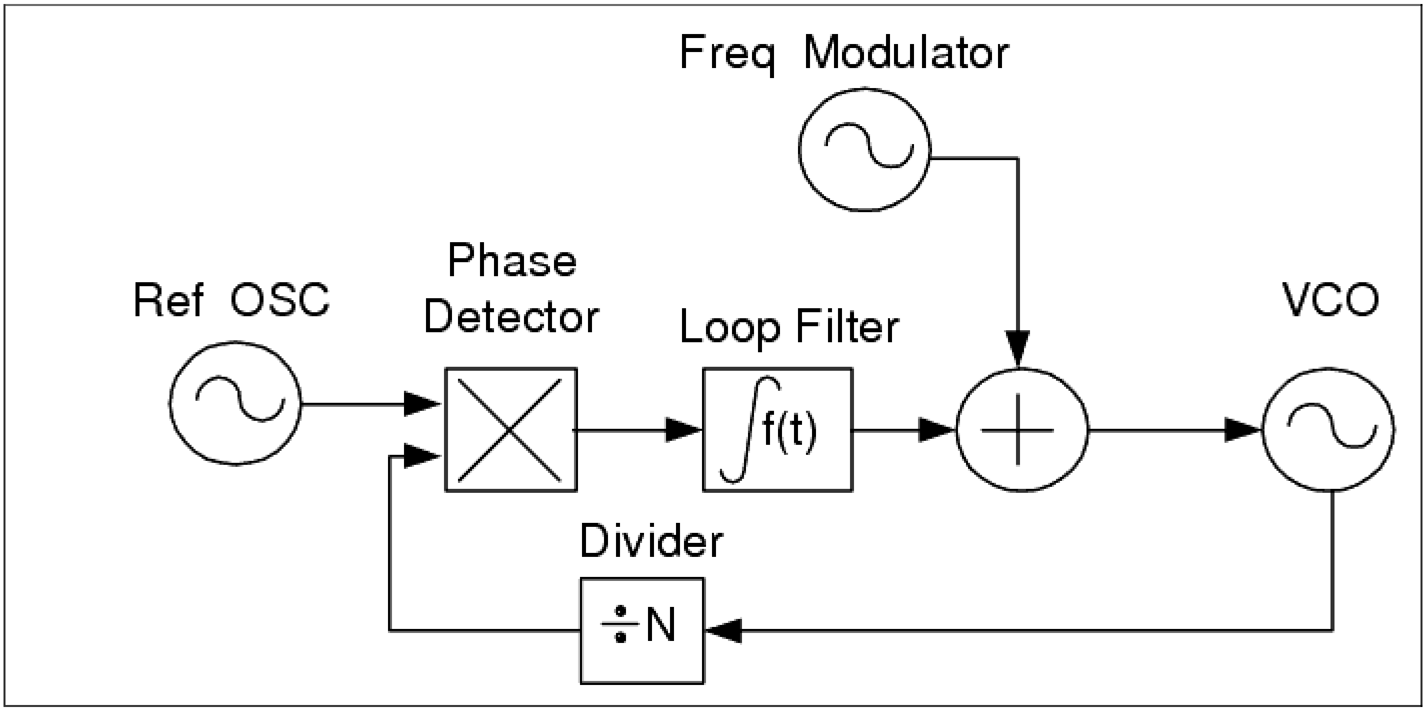


Frequency Synthesizer

The voltage controlled oscillator frequency is divided by N and then compared in a phase detector with the reference oscillator. The accuracy and long-term stability of the output frequency are controlled by the reference oscillator. The short-term stability is N times the reference inside the loop bandwidth and that of the VCO outside of the loop bandwidth. This allows a means of generating several highly accurate output frequencies. Frequency selection is performed by changing the divider ratio N.

Frequency Modulator (FMD)

Frequency modulation of the phase-locked loop is produced by adding a baseband voltage into the VCO tuning terminal along with the output of the loop filter, as shown in Frequency Modulation.

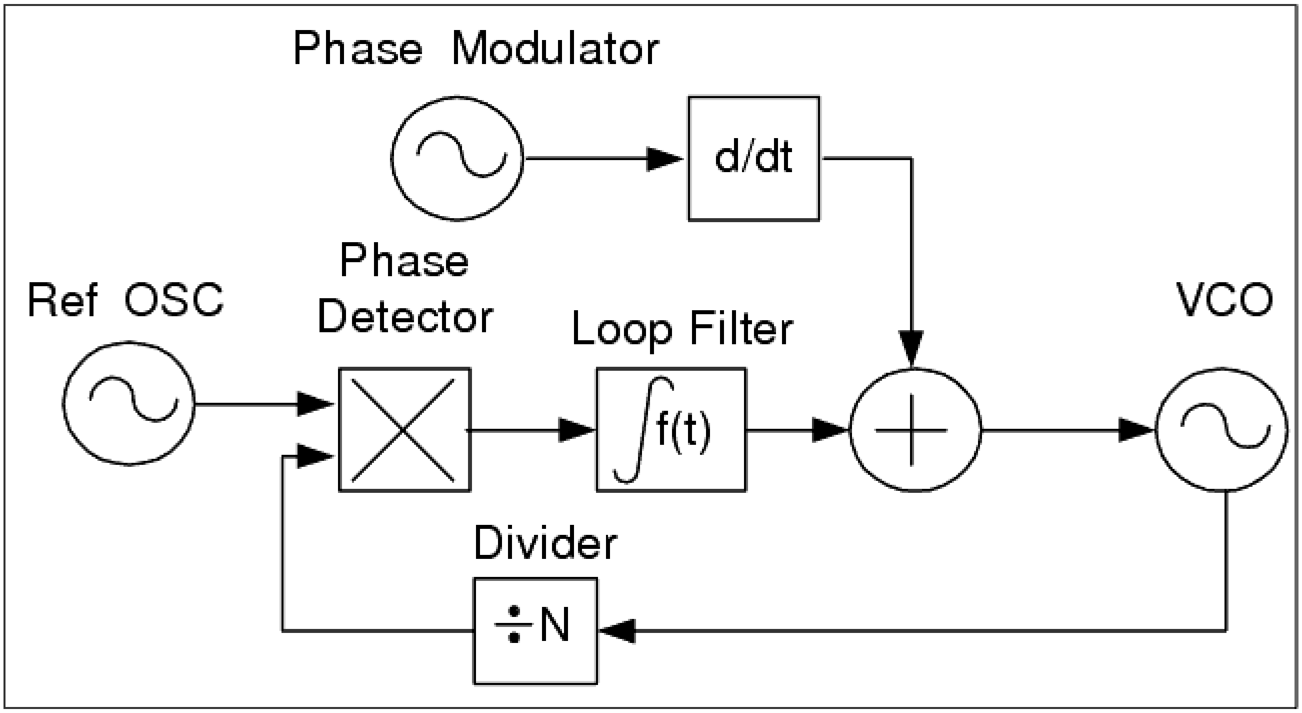


Frequency Modulation

The loop bandwidth must be smaller than the smallest modulation frequency to avoid linear distortion. The VCO characteristics must be linear to avoid nonlinear distortion of the modulation.

Phase Modulator (PMD)

Phase modulation of the phase-locked loop is produced by adding a baseband voltage into the VCO tuning terminal along with the output of the loop filter, as shown in Phase Modulator.



Phase Modulator

Another alternative is to add the phase modulator input before the loop filter, thereby, eliminating the need for the differentiator. The loop bandwidth must be smaller than the smallest modulation frequency to avoid linear distortion. The VCO and phase detector characteristics must be linear to avoid nonlinear distortion of the modulation. Phase modulation of the phase-locked loop is produced by adding a baseband voltage into the VCO tuning terminal along with the output of the loop filter.

Phase Detectors

This section provides detailed information on the phase detectors used in the PLL DesignGuide.

Detector Types

- Phase/Frequency Detector
- Charge Pumped Detector
- Mixer
- Exclusive OR

Phase/Frequency Detector

The digital behavior is modelled as a common D flip-flop. Phase correction is provided by pulse width modulation of the output.

$$|K_d| = \frac{Logic1 - Logic0}{2\pi} \ldots\ldots \left(\frac{V}{radian}\right)$$

Charge Pumped Detector

The charge pumped detector is identical to the Phase/Frequency detector except that the output is a single-ended current source.

$$K_d = \frac{Id}{2\pi} \ldots\ldots \left(\frac{Amps}{radian}\right)$$

Mixer

Mixers having wide bandwidths of operation but also have a limited locking range and therefore tend to require help during start-up.

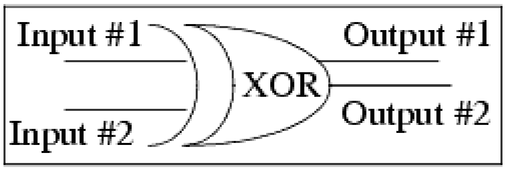
$$K_d = MixerGain \ldots\ldots \left(\frac{V}{V}\right)$$

Exclusive OR

An exclusive OR consists of basic logic components. When combined, they obey the Truth Table shown here. They provide only phase-error information.

$$K_d = \frac{Logic1 - Logic0}{\frac{\pi}{2}} \ldots\ldots \left(\frac{V}{radian}\right)$$

Input #1	Input #2	Output #1	Output #2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1



RFIC PLL Examples

This section provides a few examples of RFIC configurations that are used in the PLL DesignGuide.

Synthesizers

Three RFIC Synthesizer examples are demonstrated

- Active 3 Pole PLL configuration using a RFIC MOSFET Phase Frequency Detector
- Active 3 Pole PLL configuration using a RFIC MOSFET Charge Pumped Detector
- Active 3 Pole PLL configuration using a RFIC MOSFET Prescalar and Phase Detector

Phase Detectors

Four RFIC phase detector configurations are examined:

- MOSFET Phase/Frequency Detector
- MOSFET Charge Pumped Detector
- BJT Mixer Detector
- BJT Exclusive OR Detector.

Each configuration is measured to determine its Phase Detector Sensitivity (amps/radian or volts/radian). Also included are selectable subcircuit components.

Prescalars

Two RFIC prescalar configurations are examined:

- MOSFET Divide by 2
- BJT Divide by 2

Each configuration demonstrates the divider ratio with its transient response. Also included are selectable subcircuit components.

Reference

This section provides some useful reference information on the use of the PLL DesignGuide.

Template Reference Guide

To access the templates listed here, select *DesignGuide* > *PLL DesignGuide* > *Select PLL Configuration* from the ADS Schematic window, and make appropriate selections in the tabs of the dialog box.

Circuit Types

SYN (Synthesizer)
FMD (Frequency Modulator)
PMD (Phase Modulator)
PDM (Phase Demodulator)
FDM (Frequency Demodulator)

Simulations

FQ (Loop Frequency Response)
PN (Phase Noise Response)
TN (Transient Time Domain Response)

Phase Detectors

PF (Phase/Frequency)
CP (Charge Pump)
MX (Mixer)
XR (Exclusive OR)

Loop Filters

P3P (Passive 3-pole PLL integrator)
P4P (Passive 4-pole PLL integrator)
A2P (Active 2-pole PLL integrator)

A2PLG (Active 2-pole PLL integrator for Low Gain Opamps)
A3P (Active 3-pole PLL integrator)
A3PPF (Active 3-pole PLL integrator with passive pre-filtering)
A4PPF (Active 4-pole PLL integrator with passive pre-filtering)

Possible Template Configurations

Following are two possible template configurations. The sections that follow provide more detailed template examples.

SYN_CP_FQ_P4P

Closed and Open Loop Response of Frequency Synthesizer with charge pump detector and using a 4-pole passive PLL integrator.

PMD_PF_TN_A3PPF

Transient Response of a Phase Modulator with a Phase/Frequency Detector using an active 3-pole PLL integrator with a pre-filter.

Template Example: SYN_CP_FQ_A3P

This example is of a phase-locked loop frequency synthesizer that uses a charge-pumped phase detector and has an active 3-pole PLL integrator. The template SYN_CP_FQ_A3P identifies the fact that we are interested in the closed- and open-loop frequency response. This template also contains an optimization for determining the best resistor and capacitor values in the integrator based on the desired loop bandwidth and phase margin. There are three distinct circuits in this template:

- Closed Loop Response
- Open Loop Response
- Loop Filter Response

Active 3-Pole Integrator

The active 3-pole PLL integrator is a second-order filter.This combines with the VCO's pole to create a 3-pole PLL. The loop bandwidth must be significantly lower than the reference frequency to ensure proper sideband suppression. The loop filter consists of two capacitors (*Clpf1* and *Clpf2)* as well a resistor (*Rlpf1*). The operational amplifier used has ideal characteristics.

PLL Input Parameters

You need to identify various parameters before simulating. The VCO gain constant or tuning sensitivity parameter is identified as Kv (MHz/volt). The-charge pumped phase detector uses current Id (amps). The divider ratio N0 is the ratio between the VCO center frequency and the reference frequency. The other parameters are the Loop filter component values. The filter parameters are optimized from an initial guess value.

Optimization Parameters

The optimization criteria are the desired PLL loop filter bandwidth and the corresponding phase margin. The goals of the optimization process are to vary the loop filter component values until the loop filter bandwidth and the phase margin are within the error bounds specified in the *Goal* item. The frequency range and number of data points for the simulation are set in AC. The type of optimization and considerations are identified in the *Nominal Optimization* item. The measurement equations assist in the collection and plotting of the data results.

Closed Loop Response

The closed loop frequency response is simulated based on the optimized loop filter component values.

Simulation Results

The initial guess values of the loop filter components can be altered if the optimization results do not meet the desired design constraints. Once the *Simulate* button is chosen, the optimizer begins to adjust the loop filter components to obtain the desired loop filter bandwidth and phase margin. The *New Data Display Window* button corresponding to the design schematic is then chosen. Open the data set corresponding to the template name. In this example, open *SYN_CP_FQ_A3P.dds* . The Optimized filter bandwidth and Phase Margin are identified, as well as the corresponding loop filter component values. The plots of the open and closed loop frequency responses are displayed.

Template Example: SYN_CP_PN_A3P

This example is of a phase-locked loop frequency synthesizer that uses a charge pumped phase detector and has an active 3-pole PLL integrator. The template SYN_CP_PN_A3P identifies the fact that we are interested in the phase noise response. The optimized loop filter parameters generated from the frequency response template SYN_CP_FQ_A3P can be used in this template. The PLL parameters and the desired AC frequency sweep range need to be specified. The opamp noise characteristics can be altered to reflect your opamp.

PLL Parameters

The PLL parameters consist of the VCO tuning sensitivity Kv (MHz/volt), phase detector current Id (amps), inner loop frequency divider N0, reference frequency divider N0ref (if applicable), and the loop filter components.

Simulation Frequency Sweep

The sweep range of the AC simulator is set by the start and stop frequency, as well as the grid on the logarithmic plot.

Phase Noise Characteristics

In modeling the phase noise of the various phase-locked loop components, three distinct frequencies (F3,F2,F1) are defined at which the phase noise characteristics exhibit single sideband slopes of (-30,-20,-10 dBc/Hz), respectively. These frequencies (F3,F2,F1) correspond to the phase noise values of (L3,L2,L1), respectively. L0 defines the broadband noise floor.

Simulation Results

The *Simulate* button is then chosen and once the simulation is complete, the *New Data Display Window* button corresponding to the design schematic is chosen. The data set corresponds to the template name. In this example, open *SYN_CP_PN_A3P.dds* . The plot on the left depicts the phase noise contribution versus frequency of the various components of the PLL in the locked state. The plot on the right shows the overall PLL phase noise performance, where we expect to see system phase noise characteristics to track the reference oscillator inside the loop bandwidth, then track the phase noise of the VCO outside the loop bandwidth. The table demonstrates the PLL phase noise at different frequencies.

Template Example: SYN_CP_TN_A3P

This example is of a phase-locked loop frequency synthesizer that uses a charge-pumped phase detector and has an active 3-pole PLL integrator. The template SYN_CP_TN_A3P identifies the fact that we are interested in the transient time domain response. The optimized loop filter parameters generated from the frequency response template SYN_CP_FQ_A3P can be used in this template. The PLL parameters need to be set up. An Envelope Simulation is performed, where the fundamental frequency is that of the reference oscillator.

PLL Parameters

The PLL parameters consist of the individual loop filter component resistor and capacitor values. These values are typically derived form the optimized frequency response simulation template. In addition, the parasitic capacitance (_C_vco_) and resistance (_R_vco_) can be included in the transient simulation. The PLL parameters are specified VCO tuning sensitivity *Kv* , initial divider ratio *N0* , reference frequency *Fref,* and charge pump maximum current *Id* . The transient parameters are then specified: the loop divider step change *N_Step* , the delay time before the step occurs *Delay_Time* , the step time of the simulation *Step_Time,* and the stop time of the simulation *Stop_Time* . The delay time is used to allow the simulation conditions to stabilize before the divider step change occurs. The step time refers to the resolution accuracy of the simulation. The stop time identifies the length of time the simulation results progress, this time should be long enough to observe the step change stabilizing. The initial divider ratio *N0* needs to be entered in two places: the variable equation and the measurement equation.

Simulation Schematic

The individual components of the PLL transient simulation are identified. Note that the reference oscillator is a sawtooth waveform, allowing for better accuracy in the phase detector.

Simulation Results

The *Simulate* button is then chosen. Upon completion of the simulation, the *New Data Display Window* button corresponding to the design schematic is chosen. The data set corresponding to the template name. In this example, open *SYN_CP_TN_A3P.dd* s. The upper left plot depicts the tuning voltage that controls the VCO. The corresponding VCO frequency tracks the tuning voltage in the lower left plot. The upper right plot shows the charge pump current as a function of time. The lower right plot is the step function for the divider ratio.

Parameter Definitions

[UnityGainFreq] Loop Bandwidth: Loop bandwidth in Hertz. The loop bandwidth is the frequency at which the PLL's open loop gain equals unity (0 dB).
[Min_Phase_Margin] Phase Margin: Loop phase margin in degrees. The phase margin is equal to 180 degrees minus the open loop phase at the loop bandwidth frequency.
[Clpf1] Capacitor #1: First loop filter capacitor.
[Clpf2] Capacitor #2: Second loop filter capacitor.
[Clpf3] Capacitor #3: Third loop filter capacitor.
[Rlpf1] Resistor #1: First loop filter resistor.
[Rlpf2] Resistor #2: Second loop filter resistor.
[Rlpf3] Resistor #3: Third loop filter resistor.
[C_vco] Parasitic VCO Capacitor: VCO tuning line can be modelled as having a shunt capacitor.
[R_vco] Parasitic VCO Resistor: VCO tuning line can be modelled as having a series resistance.
[BStopN] Bandstop Filter Order: The bandstop filter is used to reduce the reference sideband level. This filter can significantly alter the loop performance. The phase margin can be degraded, which will introduce instability to the loop.
[Id] Charge Pumped Phase Detector: Maximum output current for the charge pump detector. This parameter sets the sensitivity of the detector by Kd=Id/2*pi.
[Logic1] Phase Frequency Detector Upper Voltage: Upper logic level in the digital circuit.
[Logic0] Phase Frequency Detector Lower Voltage: Lower logic level in the digital circuit.
[Mixer_Gain] Mixer voltage gain: Forward gain of the mixer. Typically it is set to 1 for a passive ring diode mixer.
[N0] PLL loop divider: Divider value in the loop.
[N0ref] Reference oscillator divider: Divider value for the reference oscillator.
[N_Step] PLL loop divider step change: Sstep change in the divider value of the loop.
[Kv] VCO tuning sensitivity in MHz/volt: The VCO tuning sensitivity is assumed to be linear across the tuning bandwidth.
[freq] frequency: Generic name for the simulation frequency.
[timestep] time step: Generic name for the simulation time steps.
[Freq_0] initial VCO frequency: Defines the initial VCO frequency before the divider step function is applied.
[Fref] reference oscillator frequency: Defines the reference oscillator frequency.
[K0] VCO tuning sensitivity (MHz/volt): The VCO tuning sensitivity is assumed to be linear across the tuning bandwidth. This value is used in the transient simulation.
[Step_Time] transient sampling period: The transient simulation time step controls the accuracy of the simulation.
[Stop_Time] transient stop time: The transient simulation stop time sets the duration of the simulation.
[Delay_Time] transient delay time: The transient simulation delay time determines the start of the change in the divider ratio value.
[Switch_Time] transient switch time: The transient simulation switch time determines the rate of change of the modulation.
[Vp_dev] modulation voltage (volts): The PLL is modulated by this peak-to-peak voltage in the transient simulation.
[Rout_mod] modulator output resistance (kohms): The PLL is modulated by a source with this output resistance.
[N_Step] divider step change: The transient simulation divider ratio is changed by this

value.
[L0] broadband phase noise parameter: The component broadband phase noise is set by this value.
[F1] frequency #1 phase noise parameter: The component frequency corresponding to the phase noise L1 is set by this value.
[L1] phase noise #1: The component phase noise corresponding to the frequency F1 is set by this value.
[F2] frequency #2 phase noise parameter: The component frequency corresponding to the phase noise L2 is set by this value.
*[L2] * phase noise #2: The component phase noise corresponding to the frequency F2 is set by this value.
[F3] frequency #3 phase noise parameter: The component frequency corresponding to the phase noise L3 is set by this value.
[L3] phase noise #3: The component phase noise corresponding to the frequency F3 is set by this value.
[F4] frequency #4 phase noise parameter: The component frequency corresponding to the phase noise L4 is set by this value.
[L4] phase noise #4: The component phase noise corresponding to the frequency F4 is set by this value.
[in0] spectral noise current parameter #0 of opamp (amps/sqrt(Hz)): Defines one of the noise current parameters that model the opamp characteristics.
[vn0] spectral noise voltage parameter #0 of opamp (Volts/sqrt(Hz)): Defines one of the noise voltage parameters that model the opamp characteristics.
[in1] spectral noise current parameter #1 of opamp (amps/sqrt(Hz)): Defines one of the noise current parameters that model the opamp characteristics.
[vn1] spectral noise voltage parameter #1 of opamp (Volts/sqrt(Hz)): Defines one of the noise voltage parameters that model the opamp characteristics.
[fvn1] spectral noise frequency corresponding to vn1 Defines one of the noise voltage parameters that model the opamp characteristics.
[fin1] spectral noise frequency corresponding to in1 Defines one of the noise current parameters that model the opamp characteristics.
[Filt_out] loop filter frequency response: Loop filter frequency response output value.
[Vout_OL] open loop frequency response: Open loop frequency response output value.
[Vout] closed loop frequency response: Closed loop frequency response output value.
[RefChain] phase noise of reference chain: Amount of phase noise in the reference chain.
[PFD_vnoise] phase/frequency detector noise: Value of noise generated by the phase detector.
[VCOout] overall phase noise of PLL: The overall noise voltage generated by the PLL.
[phi_rms] vco phase noise: The phase noise contributed by the VCO.
[refv] reference oscillator: The reference oscillator is modelled as a sawtooth waveform.
[VCO_FR] output at vco frequency: Output of the PLL at the VCO frequency.
[N0output] divider ratio function: Step function that describes the divider ratio defines this value.
[vtune] tuning voltage: Control voltage that drives the VCO defines this value.
[VCO_freq_MHz] VCO frequency (MHz): VCO output frequency.
[ChargePumpI] charge pump current (amps): The charge pump current defines this value.
[DivideRatioN] frequency divider ratio: Frequency divider ratio.
[Vcon] VCO divided output: VCO output after it has passes through the frequency divider.

[Vref] reference voltage: The reference oscillator voltage defines this value.
[V_U] upper phase/frequency detector: The upper phase/frequency detector output.
[V_D] lower phase/frequency detector: The lower phase/frequency detector output.

Encoded Subcircuits

The following section provides useful reference information for the encoded subcircuits in the Phase Locked Loop DesignGuide.

LinearMIX_V_plllib

Used in the Mixer PFD configurations, which operate in AC mode simulations. This component is an ideal mixer phase detector implementation. The input parameter is the MIXER_GAIN. This parameter is defined as the linear voltage gain of the mixer, see properties in phase detector section.

LinearPFD_plllib

Used in the Charge Pumped detector configurations, which operate in AC mode simulations. This component is an ideal charge pumped detector implementation. The input parameter is the "Id". This parameter is defined as the charge pump current (amps/radian), see properties in phase detector section.

LinearPFD_V_plllib

Used in the Phase/Frequency detector configurations, which operate in AC mode simulations. This component is an ideal phase/frequency detector implementation. The input parameters are LOGIC1 and LOGIC0. These parameter define the detectors output states ON and OFF voltages. The detector sensitivity is measured as (volts/radian), see properties in phase detector section.

LinearPFDwN_V_plllib

Used in the Phase/Frequency detector configurations, which operate in AC mode simulations. This component is an ideal phase/frequency detector implementation, including noise floor parameter. The input parameters are LOGIC1,LOGIC0 and PFD_Vnoise. These parameter define the detectors output states ON and OFF voltages, as well as the detector's output noise floor (nV). The detector sensitivity is measured as (volts/radian), see properties in phase detector section.

LinearPFDwNoise_plllib

Used in the charge pumped detector configurations, which operate in AC mode simulations. This component is an ideal charge pumped detector implementation, including noise floor parameter. The input parameter is the Id and PFD_inoise. These parameters define the charge pump's current sensitivity (amps/radian) as well as the detector's noise floor (pA). See properties in phase detector section.

LinearVCO_plllib

Used with all the phase detector configurations, which operate in AC mode simulations. This component is an ideal voltage controlled oscillator implementation. The input parameter is Hz_per_Volt, which corresponds to Kv. This parameter defines the voltage controlled oscillator's frequency sensitivity (Hz/volt).

LinearXOR_V_plllib

Used in the Exclusive OR detector configurations, which operate in AC mode simulations. This component is an ideal Exclusive OR detector implementation. The input parameters are LOGIC1 and LOGIC0. These parameter define the detectors output states ON and OFF voltages. The detector sensitivity is measured as (volts/radian), see properties in phase detector section.

Bibliography

1. Ulrich L. Rhode, *Microwave and Wireless Synthesizers* , John Wiley & Sons, New York, 1997.
2. Vadim Manassewitsch, *Frequency Synthesizers* , John Wiley & Sons, New York, 1987.
3. Floyd M. Gardner, *Phase-lock Techniques* , 2nd edition, John Wiley & Sons, New York, 1979.
4. James A. Crawford, *Frequency Synthesizer Design Handbook* , Artech House, Boston, 1994.
5. Alain Blanchard, *Phase-Locked Loops* , John Wiley & Sons, New York, 1976.
6. P.V. Brennan, *Phase-Locked Loops: Principles and Practice* , McGraw-Hill, London, 1996.
7. Dan H. Wolaver, *Phase-Locked Loop Circuit Design,* Prentice Hall, New Jersey, 1991